# NEDA

Dedicated to Packet Networking

# 1994 Annual

**Volume 5** **Revised April 1994**

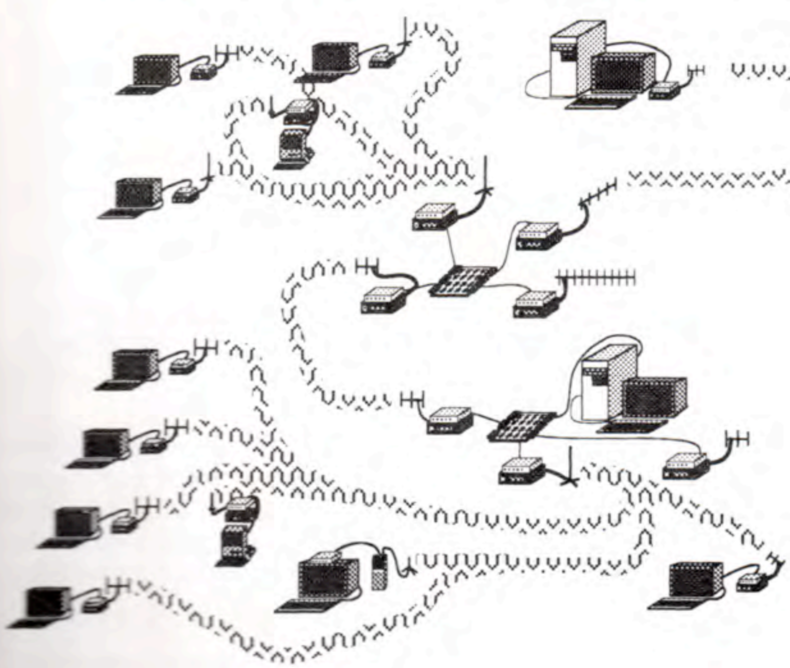
## Amateur Radio Packet is Packet Networking

**This book includes all of the recorded knowledge of the North East Digital Association, a group that has successfully promoted into existence the Western hemisphere's *largest* Amateur Radio packet network.**

***Included in this volume*:**

- **Introduction to packet radio**
- **How to buy and use a TNC**
- **What is a network and how does it work.**
- **Dozens of technical articles.**
- **The best from four years of NEDA Quarterlies!!!**
- **More than complete TheNET X-1J networking and user manual including background info.**
- **Command information for BBS programs and DxCluster.**
- **Sample BBS, CROWD and keyboard sessions.**
- **Monster glossary of packet radio terms.**

There are dozens of ways to use existing packet radio hardware to enhance Amateur Radio communications. Some work. Most, however, suffer from short life, because of a poor upgradability or a lack of growth. The North East Digital Association has, since 1989, promoted one scheme that insures a usable, functional network with off-the-shelf hardware—a network able to handle BBS and DxCluster use/traffic, TCP/IP, and live keyboarding with ease. A network established by hams involved in NEDA proves these principles over an area that spans the Great Lakes to Maine, Pennsylvania into Canada—hundreds of sites. The system is self propagating with the participants quickly gaining the expertise to promote the system into new areas.



Please join us in learning about packet radio networking and creating the network that Amateur Radio needs to perform it's educational and public service mission into the next century—it really works.

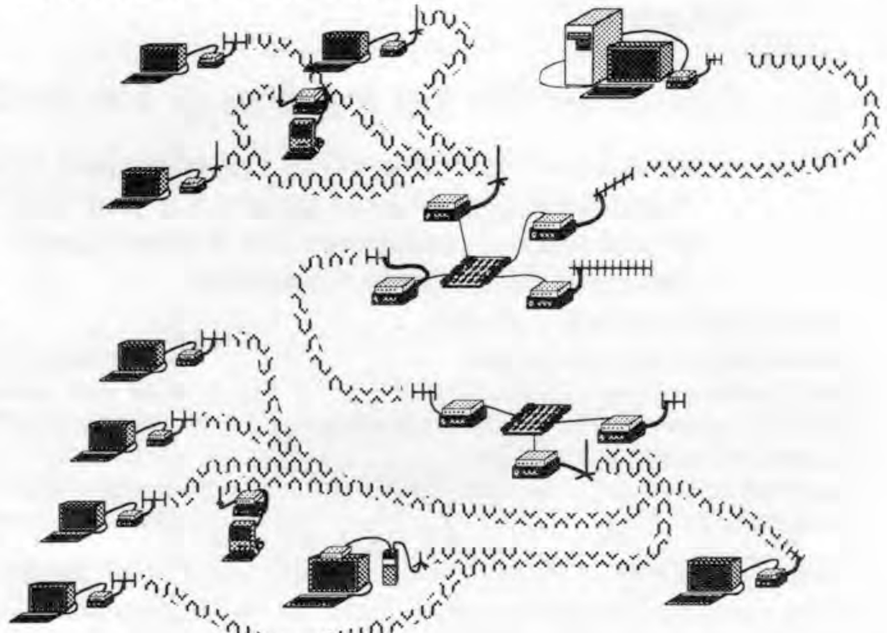
# Reading the Annual

If you've never seen this book before you'll want to page through it and see what's there. If you are totally new to packet radio you'll have to skip about a bit to find the good stuff. Since this edition of the Annual is being first read by members of NEDA and second by people who are already packeteers and are looking at joining NEDA, this book is organized with the information that experienced packeteers will want to see earlier in the book. The sections: *What is Packet*?; *Operating a Packet Station; How to use the Network, basic*; *Beginners Guide to Understanding Servers* are probably the first ones to read. Also there is a glossary. See the table of contents.

On the other hand, if you are an experienced TheNET node builder you are going to want to go directly to the goods and browse through the TheNET Resource Manual. I highly recommend that you also look at the sections on Hidden Transmitter Syndrome and the technical articles in the back half of the book.

Since this is a rather large publication it may be rather hard to read from cover to cover. This is especially true when this is the second edition of the book that you've received. I've found one of the least painful ways to go through it a second or third time is to use the book to explain packet networking to a newcomer. (see how I get you to volunteer? hi). Another way is to critique it. Pick on anything you can pick on. Look for things that are *not* said that you think should be. Send your comments to the address on the front cover or give your editor (or assistant editor) a phone call.

*Are you part of a packet network promotional club besides NEDA? Could you publish this book as part of your club's documentation? Please get in touch!* **—NEDA**



# Network!

NEDA promotes operation on a multi-application packet network. The network is mostly TheNET based. Users may connect into the network via 2 meter access points using normal TNCs and access other users and servers over a range of four or five hundred miles. There are several dozen servers of many kinds available to any user anywhere in the system. There are many round table conference nodes called CROWD which the users meet on every evening. See the maps in the Quarterly.

Use of and growth of the network is encouraged. Users may connect to network nodes and observe how the system is configured. Membership is encouraged but is not necessary. Members have the advantage of being listed in and receiving the Quarterly. Non members have access to this material only through privately photocopied copies. Generally members have more fun. This plus the fact of support of this process has been shown to be worth the membership dues.

The network is entirely privately owned or owned by radio clubs. NEDA does not own any hardware in the network. Network participants have agreed to certain principals that have been instrumental in seeing the system grow from a few sites to more than 70 sites. The following is a brief of those principals.

- The network is open for use by all packeteers;
- The network carries traffic for keyboard users, DxClusters, DxCluster users, packet mailboxes and mailbox users, playing of games, TCP/IP hosts, transferring of data or programs and many other kinds of operations;
- No user or server is more important than any other and must share the network equally (except for in emergency situations or in officially sponsored emergency drills);
- Network operators agree to a standard set of parameters, within the limitations of the software used, such that equal access to all is assured;
- Emergency operation and public service are of major importance, both because of the purpose of amateur radio and because by public service we get publicity and thus growth.

# NEDA Officers - A History

The following is a list of the hams who were founders or who have served as board members and board appointed officers since the club started.

**Herb Belin - WA1TPP**
founder
board member 1990
nrs 1990

**Kevin Wright - WA2VAM**
founder
board member 1990, 1991
nrs 1990, 1991

**Dana Jonas - WA2WNI**
founder
board member 1990, 91, 93, 94
alternate 1992
nrs 1990, 1991
secretary 1990-1993
editor 1992
co-chair NESAC 1990-1994

**Rob Marzili - WZ3T** (ex kc3bq)
founder

**Jim Wzorek, K1MEA**
founder
board member 1990-1994
nrs 1991
nbbsc 1990, 1991, 1992

**Bob Lafleur, NQ1C**
founder
board member 1990, 1992
alternate 1991
information manager 1990-1992

**Tadd Torborg, KA2DEW**
founder
board member 1990, 1994
alternate 1991, 1993
chairman ntech 1990, 1993, 1994
editor 1990, 1991, 1993

**Herb Salls, WB1DSW**
treasurer 1990-1993
membership 1990-1993
office manager 1990-1993

**Cal Calvito, WA1WOK**
board member 1992, 93
alternate 1991
co-chair nesac 1990-1994

**Cal Stiles, W1JFP**
board member 1991-1994
board chair 1991-1994
tcp/ip 1992-1994

**Linds Collins, NR1N**
board member 1991
alternate 1992
nrs 1990, 1991

**Rich Place, WB2JLR**
board member 1991-1993
chairman ntech 1991, 1992
nrs 1991

**Howie Cohen, WA2TVE**
Chairman HexiPus 1991-1993

**Bob Seger, WB2QBQ**
alternate 1991
board member 1992-1994
treasurer 1993-1994
information manager 1993-1994

**Don Russ, N2CZL**
alternate 1991

**Russ McAllister, WA1TLN**
alternate 1992-1994

**Chris Piggot, WZ2B**
alternate 1992
tcp/ip 1992

**Burt Lang, VE2BMQ**
alternate 1992
board member 1993, 1994

**Dave Packard, K1YHR**
alternate 1993, 1994

**Paul Straney, N2LSS**
nbbsc 1993, 1994

**John Burningham, WB8PUF**
alternate 1993, 1994
membership 1993, 1994
editor 1994

**Don Rotolo, N2IRZ**
alternate 1993, 1994
office manager 1993, 1994

**Bill Slack, NX2P**
alternate 1993, 1994

# 1994 NEDA Annual Table of Contents

# What is Packet?

Packet is a method of communicating digitally. The word packet describes the manner that blocks of information are transmitted and received. Text information, either hand typed or computer generated, is transmitted from station to station in blocks of between 1 and 256 bytes. Each block is acknowledged by the destination station. Lost blocks are re-sent. This means that every block that a user would see on the receive end is guaranteed to be error free. This does not guarantee that all blocks will make it. It is up to the sending station to resend lost blocks. How and when the sending station transmits we'll get into later.

## Why do we packet?

Of all of the modes of communications used in ham radio, packet is the only mode which inherently allows several conversations to occur in the same piece of spectrum over the same path. This means that on one frequency in the 2m band several pairs of stations can carry on conversations at the same time. A packeteer can start a conversation on a frequency that may already be in use, without fear of hindering the other conversation or conversations. We can take advantage of a linked network of packet stations without worrying about keeping other stations from doing the same. We can connect our station or computer to other stations to run operations that might last for hours or even years!

One practical application of long duration packet operation is Dx spotting. All over the US and Canada DxCluster servers are connected together to share Dx spotting information. The way you use this is to packet with a local contact point (there are many) all the time while you are operating your HF station. Whenever you work a rare one on HF you can type a note to all of the other hams who are currently *on the net*. This can include 200 or more stations at once. Each message you type can be routed to all of the other packet stations (that are checked in) at once or you can select to type to an individual station. You in turn will see all of the Dx spotting reports typed by the other hams that are tied in.

Packet is a computer based communications method. This means that your communications can take some advantage of the *power of the computer*. For instance your packet station could be used as an excellent selective call device. You can leave your station on all of the time and when another ham calls you your station can inform you. Thus you take notice of only that activity which is directed at you. (You can also set it up to monitor other local activity.) Using the latest TNCs that have built in message storage called Personal Message Systems you can have your friends leave messages for you when you are not in. Your TNC will tell you that there is mail waiting with a signal lamp.

## What do we need to packet?

There are 3 basic parts to a packet radio station: A radio system, a display/entry device, and a packet radio TNC. Lets cover each in turn.

***Radio***

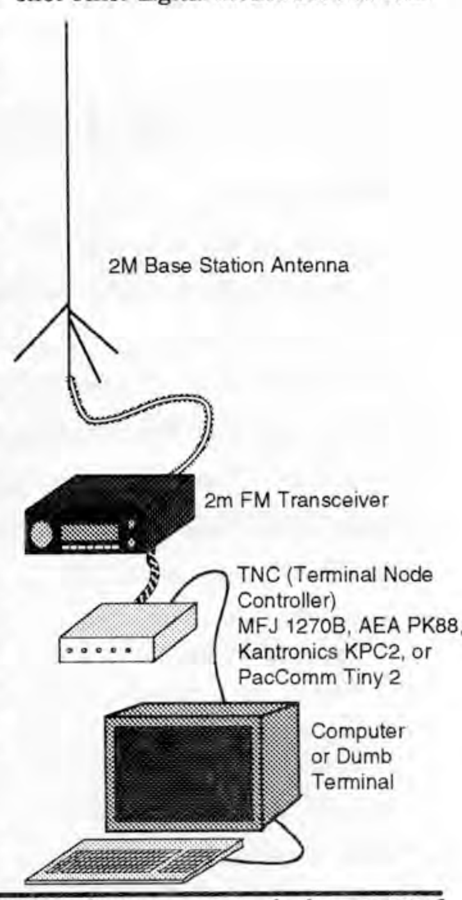
The radio system looks a lot like a base station 2 meter FM setup. The only real difference is that you don't need the microphone to do packet. Note that VHF packet can be done on 220 or 440 in some areas as well and we'll mention HF packet in another article. Like any other aspect of ham radio QRP is not as easy to use for a beginner. The ideal station would be a low power 2 meter station (1 watt) with a small beam but until you get used to packet and what is out there I recommend that if you can arrange it that you start out with a 25 watt station and a good base station antenna. In some areas a handie talkie will perform perfectly.

***Computer or dumb terminal***

The display/entry device can be anything from a simple computer system like the Commodore 64 to a "dumb" CRT terminal to a more elaborate computer system like a PC clone or Macintosh. The computer must have a TTL serial or RS-232 interface and you must have a communications program to run on it. A computer with a disk drive will allow you to store your conversations or received text and may also let you use some of the more powerful or sophisticated packet modes. A "dumb" terminal may be found in the surplus market for $30 or so (for instance flea market specials).

***TNC***

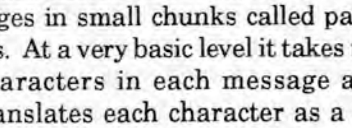
The packet radio TNC is the real key to the operation. This "TNC", which means Terminal Node Controller, is a computer device itself that takes care of all of the dirty work involved with packet communications. The TNCs range in price from about $115 to about $400. The cheaper TNCs are just as good for VHF packet radio as the more expensive ones. The more expensive TNCs offer other digital modes such as com-



puter operated Morse Code, AMTOR, digital reception of rebroadcasted satellite pictures, RTTY (radio Teletype), Facsimile, and even slow scan television. Consult the packeteer of your choice or ham magazine for advice on which of the models is the better choice! If you will settle for VHF packet operation only you should not spend more than $130 for your TNC.

## How does it work?

Packet radio allows the digital transmission and reception of messages in small chunks called packets. At a very basic level it takes the characters in each message and translates each character as a sequence of high tones and low tones. Each letter consists of 8 bits, each bit can be a high tone or a low tone. The letter is preceded by a high tone and followed by a high tone (Start bit and Stop bit). A letter "C" sounds like: high lo high lo lo lo lo high high high.



The tones all run together with no silence in between and on amateur packet radio the total length of a single character is about 10/1200ths of a second. That's ten bits at 1200 bits per second. This means that a message with one hundred and twenty characters could be sent in one second.

***Addresses***

Now comes the nifty part. Each packet includes, at the start of the packet *burst,* the callsign of your station and the callsign of the destination station! That means that if you chose, your station can reject any packets that are not to you. Secondly each packet station only transmits for long enough to get across its short message. Thus several hams can use a single frequency for conversations without having to *listen* to each of the other conversations.

In normal packet operation you would type a carriage return after each line of text that you are sending to another station. After you type the carriage return your packet station will wait for a quiet moment on the frequency and then send its message. If you have specified an intermediate station in the *path* to your friend then the intermediate station will hear its call in your message and will retransmit your message, only if the message is received perfectly and after the intermediate station sees the frequency quiet. Then your friend's station will hear the message and send back an *acknowledgment* which is picked up and echoed by the intermediate station. When your station gets the acknowledgment it will go on and send your next line of text when you hit the next carriage return. If you have already hit the next return then your station will immediately start looking for the frequency to get quiet and will then transmit the next line. If your station waits for a preset amount of time and doesn't get an acknowledgment for its packet it will send another one. This will repeat until the message gets through or your station sends RETRY amount of times, (usually 10). The form of communications where your station waits for a quiet moment and then transmits its message is called "Carrier Sense - Multiple Access" or CSMA.

***Digipeaters***

By the way, you can specify up to 8 stations as intermediates and your message will be echoed by each in turn all the way to the destination station. Each intermediate station is called a "digipeater". Any station, including yours, may be used as a digipeater by another station, merely by specifying your station's call as an intermediate. Except for emergencies or when no other resource is available you should not use digipeating. See *Hidden Transmitter Syndrome* and *Using a TheNET Network*.

## How do we use it?

Your packet TNC operates in 2 modes: Command mode and Converse mode. In Command mode you can instruct your TNC about its operation, its callsign, RETRY value or whether it MONITORs the channel or listens only to messages with its callsign. Additionally you can command your TNC to *connect* to another station. It is in the connect command that you specify the destination callsign and the callsigns of any intermediate stations.

In Converse mode anything you type will be sent over the air when you type a carriage return. If you are connected to another station the TNC will send the message and wait for an acknowledgment or retry as described above. If you are *not* connected to another station your TNC will send the message as soon as the frequency clears and will not wait for an acknowledgment. This is called *unproto* or *non-connected* mode and is useful if there are other stations MONITORing the frequency. This is how you may call for any contacts (for instance calling CQ). From command mode you can tell your TNC to use a digipeater during unproto transmissions using the UNPROTO command. More detail on this process is covered in *Operating a Packet Station*.

## Anything Else?

Glad you asked. There are many ways to play with packet. Some hams set up automated stations (called servers) which allow connection by other hams. You may then connect to an automated station and command it to perform many fun and useful functions. Among the most common sort of automated station is the packet bulletin board or PBBS, also known as a "mail box". These stations allow a packeteer to connect up and send and read messages. Each message is usually up to about 2000 characters long but sometimes 10,000 or more. The mail box lets you look at any messages that are listed as bulletins. You can send messages to other stations and you can read messages that are to you. You can also send bulletins. (This is covered under *A Beginner's Guide to Servers*)

**—KA2DEW**

# Operating a Packet Station

At present there are several different brands of packet controllers and many kinds of software, both in the packet controllers themselves and that will run on personal computers. This article can not attempt to cover all forms of packet software and hardware combinations. Instead I'm going to address the specific software that the Tucson Amateur Packet Radio club produced. This software is identical to that which PacComm and MFJ ship although both of those companies offer enhanced versions. AEA and Kantronics use a similar (but not exact) software in their simpler packet controllers. Please consult your owner's manual for the details.

Hook up the packet station. You'll need a 2m radio and antenna. To find a frequency to operate on you should consult the packet maps, ask somebody or if all else fails dial your radio around in the range of 144.91 through 145.09. If you can get a S9 or better signal on one of those frequencies you are probably all right. Figure out what frequency you want to try first and note it. We'll get back to the radio in a few inches.

If you can get the callsign of somebody that is within range and the frequency they'll be on, that would be advisable.

You'll need a cable to plug your TNC into your radio. For your first attempt it is advisable to connect the TNC to the earphone/external speaker jack on the radio, as well as to the microphone connection. Some radios have audio available on the microphone jack and others have it on an accessory jack on the back of the radio. The reason that you'll want to start with the earphone/external speaker jack is because you definitely have volume control of that audio and it is definitely going to have the range (both loud and quiet) that you'll need. Most TNCs come with enough information to make that cable. You will also need the pin-out of the microphone connector on your radio. If you don't have the documentation for the radio you may have to take the microphone apart to figure it out. The connections in the mike connector you'll need are PTT (ground = active), Microphone audio, and Ground. Most TNCs have a receive squelch signal input but you won't be using that now. Make sure you have ground connected to the earphone plug as well as to the microphone plug.

Plug the TNC into power with the TNC to radio cable disconnected on both ends. Dial the radio to a known quiet frequency or disconnect the antenna. Turn off the radio. Turn on the TNC. The TNC should power up and indicate so via it's LEDs. The PacComm Tiny 2 and the MFJ 1270B will light up three lights as soon as the power switch is pushed in. Within 5 seconds two of the three lights will simultaneously go off. This means that the TNC is working. Now turn on the radio. Since the radio is not hearing anything the S meter should read zero. Open the squelch so that you hear the FM rushing noise. Turn the volume all the way down while leaving the squelch open. Now plug the TNC cable into the TNC and into the earphone jack of the radio. At this point only one LED should be lit on the TNC (more LEDs on the PK232, KAM, PK88). Adjust the volume on the radio upwards until the DCD LED lights up. That means your Receive Audio and Ground leads are correctly connected. Adjust the volume such that it is right at the point where the DCD LED just turns on. It should either be flickering or solid on. There are some radios where this procedure does not work, the Midland 13-500 and 13-509 are two that I've had this problem with. If that is the case you'll have to guess at the volume setting. It's lower that it would seem to need to be. The volume and DCD flicker alignment procedure is actually the best we can do without an oscilloscope and taking apart the TNC. You don't have to be that picky.

Now close the squelch until the DCD light goes off. You'll want the squelch open enough that you can definitely copy the station you'll be talking to. In general the only station you need to be able to talk to direct is the packet node you'll be accessing the network with. In some cases, however, you'll want the squelch fairly loose so get used to setting it right at the point where the DCD light goes off.

Now hook up the radio's antenna and set the frequency to where you'll be communicating. Your RF side is now ready.

Hook up the computer or dumb terminal to the TNC. With PacComm TNCs the default baud rate is 1200 bauds. The MFJ 1270B is set with dip switches on the back. Be careful moving the dip switches, it's pretty easy to rip the switch from the PC board, it's only soldered down and is not attached to the case. Hold it with one finger while moving the switches with a pen or small tool. One safe way is to put the face of the TNC on your knee and hold just the switch with one finger, then make adjustments. It's not that it is *really* that touchy but it's a real bear to replace or even to spot that that's the problem. *Note: If you have an aging MFJ TNC that seems to work on the RF side but doesn't talk to your CRT terminal, check the switch. They break.*

So, set your terminal program or CRT terminal to 1200 bauds or what you have the TNC preset for. Now power the TNC off and back on. Just as the two lights on the front panel wink off, the TNC should send your screen about five lines of text indicating the revision of TNC software and the manufacturer.

Type a single carriage return. The TNC should echo with a cmd:. If you type another carriage return you should get another cmd: on the next line. If an extra line is used or if the cmd: is on the same line as the previous one then you'll want to play around with your CRT terminal or terminal program on your computer.

Now type **my KA2DEW** (use your own call!) and then a carriage return. The TNC should come back with some kind of acknowledgment like: **was N0CALL** or something like that. You should have another cmd: prompt. Now type **M** and a return. The TNC will come back with **MONITOR is OFF** or **MONITOR is ON**. Monitor is a feature by which you can see what is happening on the frequency you have selected. You'll want to read up on monitor in your TNC manual. If you type DISP M and a return the TNC will tell you the status of all of the monitor options. MH is another useful command. Try it. If it displays a list of callsigns then you have selected a busy frequency. Each time you see a new callsign on the screen it should be added to the MH (mheard) list.

Now we're going to try connecting to the station of your choice. If you have a friend that is within range, and can get on packet while on the phone with you then call them now.

The command to start a conversation is **C CALLSIGN** where CALLSIGN is the call of the friend or of another station you can hear. Let's assume that you are within range of a station whose callsign is KA2EIA. We'll have a conversation with him and then go back to monitoring the channel.

| | |
|---|---|
| cmd:**m n** <return> | This turns off monitor mode. n means no. |
| MONITOR was ON | |
| cmd:**my** <return> | Just to be sure we did this before |
| MYCALL is KA2DEW | Good. It's correct. |
| cmd:**c ka2eia** <return> | Here's where we start the connect sequence. |

*** Connected to KA2EIA

Great. That means that you can have a conversation with KA2EIA. What *Connected to* means is that your packet station now has KA2EIA's callsign in memory. Let me back up a short way and give you some more basics.

The TNC has several *modes* of operation. Some of the modes are independent of others. I'll list them here and try to explain what they mean.

**• *Command mode***

In Command mode when you type something and follow it with a return the TNC will inspect what you typed and try to interpret it as a command. M, MH, MY, C are all commands, meaning Monitor, MHeard, MYcall and Connect. Each time you type a return while in command mode the TNC will respond with a cmd: and sometimes a message in answer to your command. The only exception to this is when you use a command to go to another mode.

**• *Converse mode***

In Converse mode what you type is intended to go to another station. You can get into Converse mode from command mode by typing **conv *return*** at the cmd: prompt. Most TNCs also let you type **k *return***. To get back to command mode from Converse mode you can type control C (also written as ^C) which means that you hold down your Control key on your keyboard and type a C. It's almost the same as shift C but it's control C.

**• *Monitor mode***

Monitor mode means that you have Monitor turned on in your TNC. Monitoring can be done in both Converse and Command. There are other TNC commands that control whether monitoring is done when connected..

In addition to these 'modes' there is also a *Connect status*. You can either be Connected or Disconnected or in the process of getting connected or getting disconnected.

**• *Connected***

This means that you are connected to another station. If you go into Converse mode while Connected anything you type on the keyboard, followed by a return, will be sent to the connected station. The connected station is expected to acknowledge a connected mode packet. If no acknowledgment is sent then your TNC will retry.

**• *Disconnected***

When disconnected if you go into converse mode and type something it will be sent out over the radio, just as in connected mode. The difference is that the TNC does not expect an acknowledgment for each transmitted packet. The callsign used for disconnected or 'Unprotocalled' transmissions is set by the UNPROTO command.

Back to where we were. Since we got the *** Connected that means we are now connected to KA2EIA. We are also now in Converse mode. When we type text in converse mode while connected it will be sent to KA2EIA. We can temporarily go back to Command mode, even while connected, by typing a control C. To disconnect from KA2EIA we must go to Command mode and then use a D command. D means Disconnect. Using a C command with no callsign will ask the TNC what the current connect status is. The answer may be Connected to KA2EIA, Disconnected, Connect in progress, Disconnect in progress.

So, from where we left off:

*** Connected to KA2EIA
Hi Tadd, I'm talking to Fred on the 220 repeater.
**OK Steve. I'll type, you can read when you're free. I just got home from work and found the antenna on my front porch. I looked through the junk and can only find 4 of the boom to element clamps. I think I'm missing the one for the driven element. Was that part of the package?**
OK, I'm clear on 220. The driven element clamp is part of the T-match assembly. That's part of the mess that is in the paper bag. Did you get that

or did your St. Bernard get that? hi
**Ah. Paper bag. Under the pulley and rope?**
Yep. You got it.
**Here it is. What a mess.**
Yep. Last year at field day it got away from us. It might have been wise to remove the coax before tipping the tower over eh?
**Sure thing. I'll get to work on it.**
Brap at you later.
73.
**^C**
cmd:**D**
*** Disconnected

Notice how at the end of the conversation we did a control C and then a D for disconnect? That's the normal way of breaking communications.

Now let me give you an example of a bad connection under noisy conditions.

cmd:**c ka2hcl**
*** Connected to KA2HCL
**Hello Ken?**
*** Retry Count Exceeded.
**^C**
cmd:

What happened here is that we got the connect to KA2HCL but never got an acknowledgment for the "Hello Ken" text. What Ken probably saw was a *** Connected to KA2DEW and then nothing after that. If Ken typed anything back eventually he also would get *** Retry Count Exceeded.

By watching the LEDs on your TNC you can make a pretty good guess as to how communications is going. The STA LED is lit when you have typed something and it hasn't gotten acknowledged yet. If this is the case it doesn't do any good to type anything more. Your TNC will buffer everything you send but only while you are connected. If you send a long diatribe at the station you are talking to and you get disconnected only after typing a lot you may have wasted words. This can be disappointing.

In the sections on servers and use of the network we'll explore other things that we can do with packet. Key words: Don't get frustrated. Call somebody and talk about it. If you learn something that you think is critical to the newcomer experience but that isn't in this book, write it down. By all means talk to the editor and see that it gets in here. 73 and see you on packet!

**—Tadd, KA2DEW**

# Beginners Guide to Understanding Servers

A server is a device that serves remote users, in this case - you. That's pretty vague I know. That's because servers can be so many different things. In a vague way your TNC is a server to you. It all depends on what you call remote. For our purposes a server is something that you will access via packet radio from your station.

### *Command Line*

All servers on packet radio use a device called a *command line*. A command line is a mechanism where a person can use a keyboard to type a *one line command*. When a computer is expecting a command line command it will wait until it sees a *carriage return* character and then will look at what characters preceded the carriage return. A command line is usually associated with a *prompt*. A prompt is a text message that tells you that it is time for you to type. The **cmd:** prompt on your TNC is a good example. Each time you see a **cmd:** prompt you may type a command. That is called a command line. The process by which your TNC analyses your command is called a *command line interpreter*.

When you tell your TNC to connect to a server it makes your keyboard the control for the server's command line. That means that if you type something and hit a return what you type will be sent to the server and the server will try to figure out what you mean. Figuring out what you mean is actually pretty simple because servers have a very limited list of things they will expect. If you type something that the server doesn't expect it will usually give you a simple *error* message to tell you that what you typed was meaningless to it. It may tell you that what you typed was meaningless even if you only missed by one character. When you type something that is one of the things that the server expects it is called a command. You are commanding the server to do something.

Many of the commands that a server expects are things that the server can do immediately. For instance you can tell the server good-bye usually with a **B** command. Just type **B** and a return. When the server gets the B command it will disconnect from you leaving you with a *** DISCONNECTED on your display.

### *Disk Drives*

A disk drive is a computer accessory that stores data in the form of magnetic impulses on a flat media that is much like an extremely high quality magnetic recording tape. The density of data is high enough that one thousand billion characters could be stored on a surface that is measured in tens of square inches. Access time to that data can be as fast as a hundredth of a second and the data can be read off the disk at millions of characters per second. The particular type of disk drive I'm describing here is called a hard disk because in order to get that much data on so small an area the recording surface has to be very flat and in order to get the data on and off the disk so fast the disk has to spin very fast.

A disk drive is an accessory to many computers, even cheap ones. The disk drive I was describing above costs about $1000 but drives which are somewhat slower and which store less data can cost as little as $200. Amateur radio operators can have a computer system with a hard drive for around $450.

There are many things that hams can do with such a computer. One of the things we can do with a computer that has a hard drive is make it possible for packet users to store and retrieve data on the drive from over the radio. Such a computer system is called a *server*.

**Servers**

Most servers are computer systems that have a hard drive and which allow hams to connect in via packet. Once connected the ham gets a prompt and can type commands on a command line. The server interprets the commands and responds. Many of the commands allow the ham to write text to the server's disk drive or read text from the disk drive. Here is a brief run down on what some of the servers do:

***Bulletin Board Systems (BBS)***

These servers allow a user to send messages to other users, to read messages and to send and receive bulletins. Messages and bulletins may be sent to users that use the same BBS and to those who use other BBSs.

***DxCluster***

Users connect and stay connected. When a user of any DxCluster hear a rare Dx station they post it to the DxCluster which copies it to all of the other DxClusters.

***DOSgate***

Users connect and then use the programs that the DOSgate operator have available. These include satellite tracking, VE exam simulation, repeater directories, games.

***Callbook Server***

Lets a user reference a computerized amateur radio callbook.

***CROWD converse node***

Several users connect and have a multi-way conversation.

***NOS***

Users access a NOS server and then can utilize TCP/IP to access other TCP/IP systems.

**Bulletin Board Systems**

The most common server used in amateur radio today is called a Bulletin Board System, or BBS. A BBS is one of many servers that is commonly made out of an IBM PC compatible computer. A BBS has an external connection to talk over packet radio to other BBSs and to users. The BBS also has a disk drive which allows the storage of messages and text files. Users connect into the server over packet radio and with the command line prompt may read messages and text files. They may also send messages to other packeteers and create text files.

The first of the packet BBSs was created by Hank Orenson, W0RLI. The program that Hank wrote started with a simple command line and a few functions. The functions were as follows:

S *callsign @ otherbbs* Send a message to station *callsign* and deliver it to BBS *otherbbs*

L list messages

R *msgnum* Read message *msgnum*

I Read the Info file

B disconnect from BBS.

When the user connects to the BBS they get a welcome message that is created by the BBS operator. It might look something like this:

cmd: **c w3iwi**
*** Connected to W3IWI
Welcome to Tom's BBS in Baltimore MD.
Use the I command to get information about the BBS.
KA2DEW de W3IWI B,I,L,R,S>
To send a message to somebody else the user would use the S command:
**s ka2eia**
Enter a title for your message:
**Trying new BBS**
Enter your message. Type control Z on a blank line to end your message:
**Steve,**
**You finally talked me into getting on packet radio. I hope this works.**
**How often do you check into Tom's BBS to get mail?**
**Tadd**
**^Z**
message stored as #215
KA2DEW de W3IWI B,I,L,R,S>

To read the message Steve would connect to the BBS later and do an L command which showed all of the messages on the board.

One of the features that made Hank's software popular was that a ham could connect to one BBS and send a message to a ham at another BBS. The BBSs had text files on disk called forward files. This file had a list of the other BBSs and a simple script that told the BBS computer what it had to do to get a message to the other BBS. Basically what the BBS did was the same thing that a user would do. It would do a connect to the other BBS and then use the S command to send the message.

As experimentation allowed and as the program got popular Hank added commands and features to the program and tried new things. The list of commands expanded from the simple few to hundreds. The user interface (the list of commands and the way the command line works) stayed very similar.

Originally what Hank wrote was a program for a computer that Xerox produced as a work processing system and which Xerox had discontinued. There were scads of these computers available for a few hundred dollars which was, at the time, considerably cheaper than most personal computers. Once clones of IBM's PC started showing up on the market for around $1000 other writers of BBS programs published their contributions. The first of these was WB8MBL. Later W0RLI converted to IBM compatible PCs. In the seven or so years since then many more authors published their contributions. One very nice thing about the BBS servers, which was started by Hank, is that all of the software has been free. The software authors have had to take public approval and user feedback as their only reward (except for infrequent donations). Actually, except for very few exceptions, all software for amateur packet radio has remained free.

If you are interested in obtaining a copy of the current run of BBS software you have only to telephone one of the many telephone BBSs. Look for articles announcing the release of new software elsewhere in this *Annual* and in future issues of the *Quarterly*. For user information on just about all of the BBSs you need only connect to the BBS and make use of the extensive HELP facilities on any of the BBSs. Some also have Downloadable files for users.

### *DxCluster*

Around about 1987 AK1A, Richard Newell, created the PacketCluster™. This is a computer software package which runs on a PC to implement the DxCluster.

The primary purpose of a DxCluster is to relay immediate rare Dx spotting information between stations. Packet stations connect to the DxCluster and stay connected for long periods while they are at their HF station. When they hear a rare country on CW or SSB they type a single command into their packet station which is relayed to all of the other connected packet stations. A ham operating HF may see a Dx spot come through from the DxCluster and will tune to the specified frequency on the HF radio and work the rare station.

The secondary purpose of a DxCluster is to be a resource of HF operating information, including callbook information, propagation forecasts and other HF DXing data.

The neat thing about DxClusters is that a DxCluster may make a full time connection to other DxClusters and share information. When a Dx spot is entered at one DxCluster it is processed and passed to all of the other connected DxClusters. In a system of dozens of DxClusters over several states, hooked together via quality network links like those specified elsewhere in this book, a Dx spot may be entered and will be passed to the far end of the DxCluster network in a matter of less than a minute or so. This means that the information is still current when it gets out to as many as 500 or so HF stations.

## Abbreviated DxCluster Command List

***(Thanks to K6LLK)***

| Command | Description | Syntax |
|---|---|---|
| Announce | Make a general announcement to all nodes | A/F |
| | Make a general announcement to local node | A/L |
| Bye | Bye, disconnect from the PacketCluster | BYE |
| Conference | Enter network conference mode | CONFER |
| DELETE | Delete mail message | DE msg # |
| Directory | Show active mail messages | DI |
| | Show All active mail messages | DI/A |
| | Show mail to or from yourself | DI/O |
| Dx | Make a DX spotting info announcement | DX *freq call* |
| Show Dx | Show a Dx spotting announcement | SH/DX |
| Help or ? | Help (displays this listing) | H |
| | Display help for a particular command | HELP *command* |
| Read | Read a mail message | R *msg#* |
| Reply | Reply to the last-read mail message | REP *msg#* |
| Send | Send a private mail message | S *call* or S/P *call* |
| Set | Set user specific parameters | Example: SET/Name Tim |
| Show | Display various DxCluster Databases | SH/*commands* |
| Talk | Talk to a specific station | T *call* |
| Type | Display a particular file | Example: TY/BULLETIN CMND.TXT |
| Update | Update a database | UPD/Data |
| Upload | Upload a general file | UPD/File |
| | Upload a bulletin file | UPD/Bull |
| WWV | Make a WWV announcement | WWV SF=*xxx*, A=*xx*, K=*xx*,*forcast* |
| | Show WWV announcements | SH/WWV |

***This material was copied from the Fall 1990 edition of the NCPA Downlink***

Since many DxCluster systems may be connected in a network the facilities of all of the DxClusters are available from any one DxCluster. The commands available on the DxCluster allow easy access to all of these facilities without burdening the HF operator with having to learn much about the network.

See the side-bar for a short table of Pavilion Software PacketCluster™ commands.

DxCluster operation is currently available with two software packages. MSYS, which is free and which is available on many of the telephone BBS systems and AK1A's version which is marketed as PacketCluster™ by Pavilion Software.

PacketCluster™ is about $150. Updates are available for around $100 when they become available. The current version, by the way, is 5.4-47. PacketCluster™ from Pavilion Software is not cheap but the support is excellent.

Pavilion Software
PO Box 803
Hudson MA 01749
508-779-6527

There is a telephone BBS in California available to support PacketCluster™ owners: DxBBS at 916-992-0923

When a user connects to a PacketCluster™ the user gets a message that the system operator has programmed and then gets a prompt just like in the BBS software above. The command set is different of course but you can play with it and it will be fairly obvious.

MSYS is available from most telephone BBS systems that support amateur radio products (See the list published in the *Quarterly*). The commands for MSYS are published elsewhere in this document.

**DOSgate**

Another useful automated station is called DOSgate. This is a program written by NM1D, Rich Bono. This program is run on a IBM PC clone and allows a packeteer to connect up and use the PC as if it were his own station. You can run programs and even create files. It also serves as a packet mail box. Some of the programs circulating that are commonly found on DOSgates include

- automatic FCC testing sessions that use the real FCC test elements to let you practice a license exam. This is great for getting new licensees used to the idea of passing their first test.
- satellite tracking program to let you see when each of the current satellites will be above the horizon and what beam headings to use.
- games like Zork and Wumpus
- amateur radio callbook. This may include your local call district or the entire FCC and DOC database.

When a user connects into DOSgate he gets a message that the system operator has created, followed by a prompt which is very similar to a PC DOS prompt:

C:\HAMRADIO>

The user can type a PC DOS command, like CD or TYPE, or the user can type the name of a program that the system operator has put onto the DOSgate system. The program can be any PC DOS program that generates straight text. No graphics and no screen formatting is allowed.

I have had conflicting reports about the availability of the DOSgate program. When I first heard of DOSgate, it was free. I've seen it downloadable from various telephone BBSs. I've heard that it was an inexpensive for-sale program, however. The best bet is to contact NM1D directly using NM1D @ WB1DSW.nh. Please get back to your editor if you have better information.

**CROWD Nodes**

These are covered in another section.

**NOS**

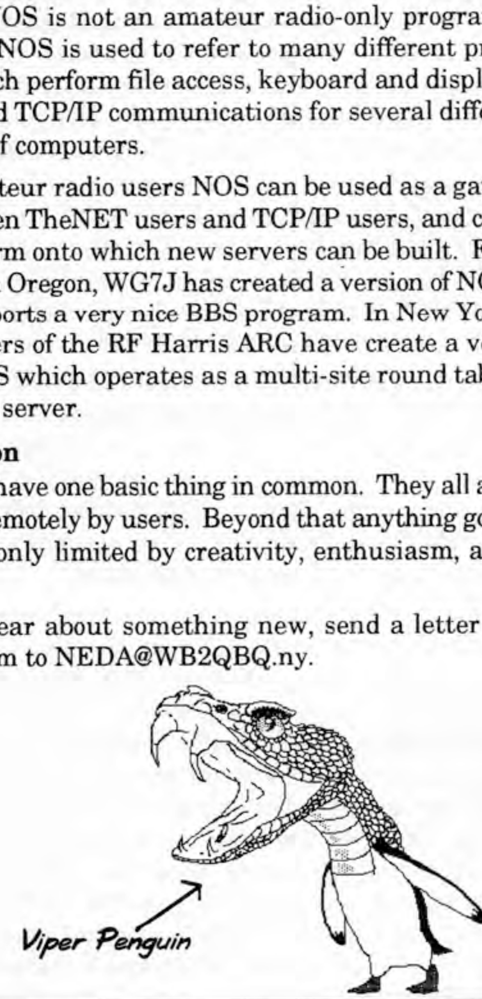
NOS means Network Operating System. NOS is a program which operates as a server to other packeteers and as an operating console for the local user. The primary purpose of NOS is to be the operating program for a local user to communicate with other stations using TCP/IP. NOS is not an amateur radio-only program. The name NOS is used to refer to many different programs which perform file access, keyboard and display control, and TCP/IP communications for several different kinds of computers.

For amateur radio users NOS can be used as a gateway between TheNET users and TCP/IP users, and can be a platform onto which new servers can be built. For instance, in Oregon, WG7J has created a version of NOS which supports a very nice BBS program. In New York the members of the RF Harris ARC have create a version of NOS which operates as a multi-site round table conference server.

**Conclusion**

Servers have one basic thing in common. They all are operated remotely by users. Beyond that anything goes and we're only limited by creativity, enthusiasm, and time.

If you hear about something new, send a letter or packet gram to NEDA@WB2QBQ.ny.



# TNC Buying Guide

Terminal node controllers come in several forms. The evolution of these controllers may be of interest as it simplifies the mystery as to why some controllers look the same even though they are from different manufacturers and as to why other controllers are so wildly different.

The furthest back we have to go into TNC development in order to make sense of what exists today is the beginning of the Tucson Amateur Packet Radio club. TAPR produced a TNC model based on a Motorola 6800 Microprocessor, in 1983. The unit cost $369 as a kit, not including the aluminum chassis. The PC board that TAPR had designed included an AC power supply which was required in order to get the plus and minus 12 volt levels need for the serial port. TAPR would sell the rights to manufacture the unit for a small sum to any and all who wanted it. TAPR's purpose was to get companies competing against each other in the manufacture of these units. AEA and Heathkit took them up on it and started marketing TNCs which looked almost identical to the TAPR TNC. The price tag for the Heathkit TNC kit was comparable to the TAPR price. AEA's was a hundred or so more, as a finished product. All three TNCs included a complex front panel LED display and features like Morse ID and Vancouver protocol support. A year later Kantronics appeared on the scene with a much cheaper model which was also based on the TAPR but using more modern, and fewer, components. The Kantronics KPC-1 cost around $250 as a finished product and operated with DC or using the included wall pack. The KPC-1 had only three LEDs on the front panel and used a simpler, but as powerful, microcomputer. The KPC-1 could run on a single voltage DC supply. The commands that the 4 TNCs understood and the way they were operated was almost identical.

In April of 1985 TAPR announced a new TNC called the TNC 2. TAPR wanted to get the price of a TNC down much lower. The TNC2 was a much simpler unit in terms of components but more powerful in that it took advantage of ICs which, in 1985, cost five dollars or more. The TNC 1 (as TAPR's first TNC was now called) was based on more, cheaper, components.

The TNC2 looked very much like the Kantronics KPC-1 but went in a totally different direction with it's microcomputer, being based on a Z80 chip instead of the Motorola family of micros as had the four previous TNCs. In addition the TNC2 used a cute charge-pump power supply circuit for the serial port so it could run on 12 volts and not have the plus and minus supply requirement. The TNC 2 was introduced at $249 in kit form. TAPR did an interesting thing here. Unlike the TNC 1 which was sold for a few years by TAPR, when the TNC 2 was introduced TAPR announced that they would only make 1000 of them. The phone would open at noon and "have your credit card ready!" For three days the Tucson telephone system was brought to it's knees!! Delivery of the TNCs would occur over a 5 month period at which time TAPR would stop production of the kit. Again TAPR was pushing for manufacturers to pick up the product. This time the manufacturers did so, and with a vengeance.

AEA, MFJ (existing amateur radio accessories manufacturers) and GLB (existing amateur radio RF kits manufacturer) all introduced TNCs which were very close to the TAPR model. All were priced about the same as the TAPR. PacComm was founded in Florida to manufacture TNC 2 clones. The price war was on. Within 2 years (1988) TNCs were down to $130. Now the war moved to bells and whistles. Multimode TNCs and built in mailboxes started appearing. Next came reduced power requirements and small size. TNCs which were built on PC compatible plug-in cards came out.

More recently hobbyists have cost reduced packet radio to around $50 for hams who already own PC compatibles by creating TNCs who's digital section would use the PC compatible. These TNCs are just a modem chip on a board that plugs into the PC's serial or parallel ports. They are called Baycom and PMP. (Poor Man's Packet).

One feature which appeared on the scene in the last couple of years is the internal mailbox mentioned above. This allows the TNC owner to leave messages in the TNC memory for friends to get when connecting in from over the radio. The friends can then leave messages for the TNC owner. It is also possible for a BBS system to forward messages to the TNC and to pick up messages from the TNC that are out bound. This means that the TNC owner need not connect to the BBS to send and receive forwarded mail. The TNC will flash an LED to indicate that mail has been received.

As of 1992 we have quite a collection of TNCs to pick from. I can make recommendations but in truth I don't have all of the information. What I'll do instead is to tell you what I know and hope that when you learn better that you'll get back to me.

There are five categories of TNCs that deserve a look:

- Stand-alone VHF-only
- Stand-alone VHF/HF deluxe multimode
- PC compatible internal (plugs inside computer)
- Modem only, Software TNC (requires computer)
- Stand-alone Micro-size and Micro-power

Unfortunately this book needs to get done and documenting all of the TNCs on Earth is a good way to see that it doesn't. So, for most of the existing TNCs I'm only going to be able to mention their existence (if even that). I have only owned and operated a few of these and am dependent on contributors for the information. Thus the coverage of bugs and features isn't consistent. Please, if you have experience with TNCs, send me info. Also make corrections and criticisms!

## Stand-alone VHF-only

- TNCs can all be operated on 12V regulated DC.
- Directly plug into the microphone and earphone of a single VHF radio
- Front panel LEDs indicate connect, PTT, DCD, power
- Comparable size, 4"x1" face x 7" deep
- Comparable power requirements, about 200mA
- All can be operated easily with dumb terminal or computer with terminal emulator
- Use standard cmd: prompt
- Allow multiple connections at the same time
- Internal mailbox connectable by other stations
- Have internal connections for new modems
- Cost about $130

**MFJ 1270B**

The MFJ TNC is available off the shelf from many of the amateur radio dealers. Ham Radio Outlet and Amateur Electronics Supply are two that sell this unit at reasonable prices and with very quick delivery time.

The 1270B uses the TAPR specified connections for power (coaxial bullet connector), serial port (DB25) and radio connector (5-pin DIN). The 1270B also has a TTL connector for it's serial port to allow compatibility with the older computers including the C64 and TI99.

The 1270B is software compatible with the TAPR model so it can be used with any EPROM software that is available for that unit, including TheNET, ROSE, KISS and DED host mode. With the included software the user can, of course, connect into any of the AX.25 networks. Software compatibility is only an issue if you plan on going inside the unit and *play* with packet.

Serial port baud rate is set via a dip-switch on the back panel and is adjustable from 300 to 9600 bauds. Radio baud rate (only useful with external modem or if used without a modem) may be set on the same dip-switch to 300, 1200 or 9600 bauds.

Power Supply:
- 12V unregulated,
- 180mA.
- includes wall pack.

Bugs:
- the case is thin sheet metal
- the baud rate adjustment switch is fragile and will break if abused.
- Once a station is connected to the PMS (Personal Message System) the operator of the TNC can't talk to that station. You can disconnect to PMS user but can't patch over to him.

Features:
- TAPR2 compatible so you can change out the software to be TheNET or ROSE networking software, or other new softwares.
- Off-the-shelf at many ham radio stores.

Where to get one:
Ham Radio Outlet - 1-800-854-6046

**Stand-alone TNC?**

Stand-alone means that the TNC can operate without a personal computer. A dumb CRT terminal is all that is needed to command or operate them. Other TNCs need to be plugged into a computer in order to be operated.

***AEA PK88***

The PK88 is available off the shelf from many of the amateur radio dealers. Ham Radio Outlet and Amateur Electronics Supply are two that sell this unit at reasonable prices and with very quick delivery time.

The AEA PK88 uses the TAPR specified connections for power (coaxial bullet connector) and RS-232 (DB25). The radio connection is via an 8 pin mike jack (same kind as used on ICOM and Kenwood mobile and base rigs) and/or via a mini phone plug. A phone plug to phone plug jumper is including allowing immediate connection to a radios earphone/external speaker port. Out of box to listening to packets is about one minute.

Serial port baud rate is set in software. When you first hook up to the TNC you type the * key over and over until the unit answers. That sets the baud rate.

Power Supply:
- 12V unregulated,
- 180mA.
- includes wall pack.

Bugs:
- Not TAPR2 compatible software.

Features:
- Baud rate adjustment is in software, no switches.
- Extra LEDs for more status indication.
- Maildrop may be interrupted so the TNC owner can directly talk to someone who connected to the maildrop.
- The case is very heavy.

Where to get one:
Ham Radio Outlet - 1-800-854-6046

**TAPR TNC2 Compatible?**

TAPR2 or TNC2 compatable means that the computer hardware in the TNC is identical to that created by TAPR (the creators of TNC2). This is important because that means that any public domain software written for TNC2 will work in your TNC. TheNET, which is a networking software, is used for network sites not for user stations but you never know when in an emergency you could be the one who might have the one piece of emergency equipment (your TNC) necessary to get things working again. Other public domain software you might use include programs which do new kinds of packet operation.

It is quite possible that makers of non-TNC2 compatible TNCs might incorporate any new features that come out in their proprietary software but most developments for inexpensive TNCs will be available for TNC2 compatibles first.

### *PacComm Tiny 2 Mark2*

The Tiny-2 is available at PacComm independent distributors (not ham radio stores) and from PacComm direct. Delivery time is 2 to 4 weeks.

The PacComm Tiny 2 uses the TAPR specified connections for power (coaxial bullet connector) and radio connector (5-pin DIN). The Tiny 2 also has a TTL connector for it's serial port to allow compatibility with the older computers including the C64 and TI99. The Tiny 2's RS-232 connector is the same connector and pinout as a PC compatible 9-pin.

The Tiny 2 is software compatible with the TAPR model so it can be used with any EPROM software that is available for that unit, including TheNET, ROSE, KISS and DED host mode. With the included software the user can, of course, connect into any of the AX.25 networks. Software compatibility is only an issue if you plan on going inside the unit and *play* with packet.

Serial port and radio baud rates are set via a set of jumpers inside the box to 300, 1200, 2400, 4800, 9600, 19,200, 38.4Kbauds.

Power Supply:
- 12V unregulated,
- 50mA.
- Wall pack not included.

Bugs:
- the baud rate adjustment is inside the box.
- Like the other TNCs of this one's class the case closures are non-threaded and the metal screws will strip if removed and tightened too many times. Unfortunately the Tiny 2 is clumsy to close once the screws are stripped.
- Once a station is connected to the PMS (Personal Message System) the operator of the TNC can't talk to that station. You can disconnect the PMS user but can't patch over to him.

Features:
- TAPR2 compatible so you can change out the software to be TheNET or ROSE networking software, or other new softwares.
- baud rate may be set to 38.4Kbaud.
- available without EPROM and manual for a $20 cost savings making this the most economical unit for network building.

Where to get one:
NX2P Electronics - 201-729-6927

### *Kantronics KPC-2*

The KPC-2 is a VHF/FM 1200 baud packet TNC with additional packet features.

The KPC-2 uses the TAPR specified connections for power (coaxial bullet connector) and RS-232 (DB25). The radio connection is via an DB9 which is standard only to Kantronics.

Serial port baud rate is set in software. When you first hook up to the TNC you type the * key over and over until the unit answers. That sets the baud rate.

Power Supply:
- 9 to 14V unregulated,
- 250mA.
- includes wall pack.

Bugs:
- Not TAPR2 compatible software.
- Includes KA node software whose major impact on packet radio is to cause hidden transmitters.

Features:
- Baud rate adjustment is in software, no switches.
- Extra LEDs for more status indication.
- Includes KA-node software which is very useful in emergencies when traffic level is low.
- PMS uses a separate callsign and may use an alias as well. Thus the PMS could be accessed by KA2EIA-2 or by STEVE.

Where to get one:
Ham Radio Outlet - 1-800-854-6046

### *DRSI DPK-2*

The DPK-2 uses the TAPR specified connections for power (coaxial bullet connector) and radio connector (5-pin DIN). The DPK-2 also has a TTL connector for it's serial port to allow compatibility with the older computers including the C64 and TI99. The DPK-2's RS-232 connector is the same connector and pinout as a PC compatible 9-pin.

The DPK-2 is software compatible with the TAPR model so it can be used with any EPROM software that is available for that unit, including TheNET, ROSE, KISS and DED host mode. With the included software the user can, of course, connect into any of the AX.25 networks. Software compatibility is only an issue if you plan on going inside the unit and *play* with packet.

Serial port and radio baud rates are set via a set of jumpers inside the box to 300, 1200, 2400, 4800, 9600. The Serial port rate may also be set to 19.2Kbauds.

Power Supply:

- 9 to 12V unregulated, (9v battery)
- 50mA,
- Wall pack not included.

Bugs:

- the baud rate adjustment is inside the box.
- since I haven't had my hands on one I can't compare it to the other TNCs of this class but it is very similar.

Features:

- TAPR2 compatible so you can change out the software to be TheNET or ROSE networking software, or other new softwares.
- serial baud rate may be set to 19.2Kbaud.
- LEDs may be disconnected inside the unit for more power savings.

Where to get one:

DRSI - 813-461-0204

*Some information for the DRSI TNC was obtained from CQ Magazine Feb 93. See the review on page 65 of that issue.*

## Stand-alone deluxe multiport or multimode

***Kantronics KAM***

The KAM is a multimode VHF/HF digital communications controller including VHF packet TNC mode. The KAM can operate VHF packet at the same time as operating one of it's HF modes, including CW, RTTY (baudot and ASCII), FEC, ARQ, and NAVTEX.

This unit is normally used with a Kantronics program running on a PC compatible.

No price or feature information available at this time.

***MFJ1278***

Multi-mode controller with AMTOR, RTTY (baudot and ASCII), CW, FAX, SSTV (MFJ compatible), and NAVTEX.

Connections include RS-232 and TTL serial ports, peripheral I/O port, radio port (probably a TAPR standard 5 pin DIN).

Price: $280.

Power Supply:

- included AC supply.

Bugs:

- Not TAPR2 compatible software.
- Operating mode not displayed on front panel.

Features:

- Baud rate software adjustable.
- Lots of LEDs for status indication.
- Includes personal mailbox function.

Where to get one:

Ham Radio Outlet - 1-800-854-6046

***AEA PK232***

The PK232 is a multimode VHF/HF digital communications controller including VHF packet TNC mode.

The PK232 uses the TAPR specified connections for power (coaxial bullet connector) and RS-232 (DB25). The radio connection is via a 5-pin 0.1 center in-line connector. A wired cable with one of these connectors comes with the unit. All you have to do is add the connector to fit your radios mic connection. An audio in-line is provided through this connector, as well as a 1/8 inch mini phone jack for connecting to your radios earphone/external speaker jack.

Serial port baud rate is set in software. All you do is send a series of ***, and the PK232 will match the terminal's speed.

In addition to packet, the PK232 will run RTTY at 5 bit baudot code, or 7 or 8 bit ASCII at baud rates from 45 baud up to 300 baud, with settings for 60 and 100 words per minute. It will run AMTOR and machine Morse Code both send and receive at up to 100 words per minute. (To receive the code needs to have near textbook spacing). With the right cables it will also do weather FAX.

Price: $320

Power Supply:

- 12V, 700mA regulated (not included)

Bugs:

- Not TAPR2 compatible software
- Loss of power before unit is turned off may totally reset the unit
- Internal 14 MHz clock will cause birdy on one frequency on 20 Meters. If it falls on your favorite frequency, the unit can be modified to move it off frequency, but I have not heard of any permanent fix.
- Operation of the unit can be so complicated that it may be totally baffling for a beginner. A program for your computer, to operate the PK232, is a must. Such a program is available from AEA at an additional cost.

Features:

- Baud rate software adjustable.
- Lots of LEDs for status indication.
- Tuning indicator for HF modes.
- Includes personal mailbox function.

Where to get one:

Ham Radio Outlet - 1-800-854-6046

## PC compatible Internal

These are cards that plug into a PC compatible. Unless noted they require software running in the PC to make them operate. Sorry for the appearance of PacComm favoritism but I got most of this information from a chance telephone call. More will flow in the next edition of this volume.

***AEA PCB88***

Same features of the PK88 but requires a PC compatible computer. PC compatible plug in internal PC card. Has on board processor and external power connection (12V) so the TNC can operate when the PC is turned off.

Price: not available

Contact AEA at 800-432-8873 or 206-775-7373 for more information.

***PacComm PC320***

Complete TNC on a PC compatible plug in card. Allows two radio connections but only one at a time. Includes HF (300 baud) and VHF (1200 baud) modems. Standard modem disconnect header. The PC320 is available at PacComm independent distributors (not ham radio stores) and from PacComm direct. Delivery time is 2 to 4 weeks.

Both radios connect to a single DB-9, although the connections exist for both radios at the same time. A shareware terminal emulator program for a PC compatible is included. A Terminate Stay Resident program (run in your autoexec.bat) is included that supplies the tuning indictor for HF modem operation. A command at the cmd: prompt changes which port is in use.

The PC320 is powered by an external 12V supply and may remain on when the PC is shut down.

Price: $210<br>
Where to get one:<br>
NX2P Electronics - 201-729-6927

***PacComm PC105D***

A modem on a PC compatible plug in card. Allows one radio connection on VHF (1200 baud). The PC105D is available at PacComm independent distributors (not ham radio stores) and from PacComm direct. Delivery time is 2 to 4 weeks.

The radio port connector is a DB-9. The unit comes with G8BPQ node software and includes the G8BPQ terminal emulator. It also comes with "node manager software" written by AC4X which operates as a terminal emulator, as well, using the G8BPQ node software.

Includes open squelch DCD adapter. (that's what the D suffix means)

There is a 2nd modem port on the PC105D board which might be used for a TTL port. This feature is not well supported by the documentation. This unit is specifically designed to allow you to not use the COM1 and COM2 interrupts.

Price: $120<br>
Where to get one:<br>
NX2P Electronics - 201-729-6927

***PacComm PC110D***

A modem on a PC compatible plug in card. Allows one radio connection for either HF (300 baud) or VHF (1200 baud). The PC110D is available at PacComm independent distributors (not ham radio stores) and from PacComm direct. Delivery time is 2 to 4 weeks.

The radio port connector is a DB9. The unit comes with G8BPQ node software and includes the G8BPQ terminal emulator. It also comes with "node manager software" written by AC4X which operates as a terminal emulator, as well, using the G8BPQ node software.

Includes open squelch DCD adapter. (that's what the D suffix means)

There is a 2nd modem port on the PC110D board which might be used for a TTL port. This feature is not well supported by the documentation. This unit is specifically designed to allow you to not use the COM1 and COM2 interrupts.

Price: $130<br>
Where to get one:<br>
NX2P Electronics - 201-729-6927

***PacComm PC120D***

Two complete modems on a PC compatible plug in card. Allows two radio connections at a time. One radio connection can be on either HF (300 baud) or VHF (1200 baud). The other must be on VHF (1200). The PC120D is available at PacComm independent distributors (not ham radio stores) and from PacComm direct. Delivery time is 2 to 4 weeks.

Both ports are on one DB9. The unit comes with G8BPQ node software and includes the G8BPQ terminal emulator. It also comes with "node manager software" written by AC4X which operates as a terminal emulator as well, using the G8BPQ node software.

Both ports have open squelch DCD adapters. (that's what the D suffix means)

This unit is specifically designed to allow you to not use the COM1 and COM2 interrupts.

Price: $160 (call for latest price)
Where to get one:
NX2P Electronics - 201-729-6927

***DRSI PC*PA***

Available in dual port and single port models, this is a PC compatible plug in internal PC card.

## Modem-only Computer-required

These units require a personal computer to operate. Some use Commodore 64 computers, others use PC compatible computers. These are generally kits or inexpensive assembled units costing around $50. The software is usually free-ware (no cost) or shareware (pay if you are honest) and requires that the computer is on and running the program in order for packet to function.

***Tigertronics BayPac BP-1***

This is a PC compatible computer based modem/TNC. The TNC plugs into an external port on the PC (not sure if it's the printer port or COM/serial port) and operates using supplied software. The entire unit is the size of the connector that plugs into the PC. Call Tigertronics for more details.

Price: $49.95 for the modem. Software included is shareware (I think) and has to be paid for after purchase. Check with Tigertronics on this.
Where to get one:
Tigertronics Inc. - 1-800-822-9722 or 503-474-6700

***MFJ 1271***

This is a Commodore 64 computer based modem/TNC. The MFJ modem unit plugs into the computer's cassette port and operates both 1200 and 300 baud. The unit may be driven by 'Digicom' software (share ware) or by the MFJ purchased software. Contact MFJ at 1-800-647-1800 for more information.

Price: $49.95 for the modem, $4.95 for the MFJ 1293 software.
Where to get one
Ham Radio Outlet - 1-800-854-6046

***PMP***

N8KEI and crew developed this one, called Poor Man's Packet, in the 1989 through 1990 time frame. This is a device which plugs into the parallel port (I think) of a PC compatible.

## Micro-size and Micro-power

***Kantronics KPC-3***

The KPC-3 is the successor of the KPC-2 described earlier. The software appears to be slightly enhanced. The TNC is not *micro* sized although it is a bit smaller than most of the normal stand alone units.

Serial port baud rate is set in software. When you first hook up to the TNC you type the * key over and over until the unit answers. That sets the baud rate.

Price: $110
Power Supply:
- 6 to 25V unregulated.
- 14mA with LEDs off.
- operates off of internal 9V battery if desired.

Bugs:
- Not TAPR2 compatible software.

Features:
- Baud rate adjustment is in software, no switches.
- Has "new mail" indicator LED.
- Has LED indicating someone connected to mail box.
- Some parameters are remotely accessible.

Where to get one:
Ham Radio Outlet - 1-800-854-6046

***PacComm HandiPacket***

This is a non-TNC2 compatible TNC. It is low power consumption and just slightly bigger than a comfortable shirt pocket size. Contact the supplier for more information. Includes internal rechargeable NICAD battery which runs it for 9 hours.

Price $209
Where to get one:
NX2P Electronics - 201-729-6927

# MSYS BBS User Command Summary

### General Commands

| | |
|---|---|
| B | Log off the PBBS. |
| I | Display hardware configuration of PBBS. |
| ID | Display list of the ports, digipeaters, gateways, and nodes available on the system. |
| J # | Display callsigns of stations recently heard by or connected to the PBBS. [optional, port #]. |
| Jx # | Display callsigns of other systems recently heard by the PBBS. [optional, port #] {may be disabled by Sysop} |
| JB | PBBS Systems |
| JD | Digipeaters |
| JG | Gateways |
| JK | KaNodes (i.e.; Kantronics) |
| JN | NetRom/TheNet Nodes |
| JM | MSYS PBBS's |
| JT | TCP/IP users. |
| M | Display system Message of the Day. |
| N x | Enter your name (x) in the system. (10 characters maximum). |
| NQ x | Enter your QTH (x). |
| NZ x | Enter your Zip Code (x). |
| NH x | Enter the callsign (x) of the PBBS where you normally receive mail. |
| P x | Path last used by a station (x) to connect to the PBBS. |
| PF x | Path used by the PBBS to forward messages to another PBBS (x). |
| T | Page the System Operator so that you may talk to him. |
| U | Display list of stations currently connected to the PBBS and their current activity. |
| V | Display PBBS software version number. |

### Message Commands

| | |
|---|---|
| CC # x | Send a copy of a message (#) to a station (x). |
| K # | Kill message number #. |
| KM | Kill all messages to you that you have read. |
| KT # | Kill NTS Traffic message numbered #. |
| L | List new messages since the last time you listed messages and then logged off the PBBS using the B command. |
| LB | List bulletins (ALL OF THEM!!!) |
| LM | List messages to or from you (List Mine). |
| LT | List NTS Traffic messages. |
| LL # | List the last # messages. |
| LU | List messages to you which you have not read. |
| L< x | List message from a station (x). |
| L> x | List message to a station (x). |
| L@ x | List messages with a specific coverage area or specific PBBS (x). |
| L # | List messages numbered greater than or equal to #. |
| L #1 #2 | List messages numbered greater than or equal to #1 and less than or equal to #2. |
| L"xyz" | List messages with the specified string (xyz) in title. {case insensitive} |
| L'xyz' | List messages with the specified string (xyz) in title. {case insensitive} |
| R # | Read message number #. |
| RH # | Read message number # displaying full message header and forwarding info. |
| RM | Read all messages to you which you have not already read. |
| RN # | Read message number # displaying no message header information. |
| REPly # | Send a reply to the author of message number #. |
| SB x @ y | Send a bulletin message to a target audience (x) within a specified coverage area (y). |
| SP x @ y | Send a private message to a station (x) at a specific PBBS (y). |
| ST # | Send an NTS Traffic message to the specified Zip Code (#). |

### Help Commands

| | |
|---|---|
| A | Abort current process and return to PBBS command prompt. |
| H | Display a short summary of all PBBS commands. |
| ?x | Display extended help information for a command (x). |
| X | Toggle between short and extended command menu (expert mode). |
| XF | Enable "Fast" mode (>1 line/packet). |
| XS | Enable "Slow" mode (only 1 line/packet). |
| X # | Set number of lines (#) to display before pausing. |
| * xxx | Enter a comment line (xxx). Useful for replying to the Sysop when you get a "Message from Sysop...". |

### MSYS DX Node

| | |
|---|---|
| C | Enter the MSYS DX Node. Note: May not be available on all systems/ports. |

### MSYS DX Node Commands

| | |
|---|---|
| B | Disconnect from DX node. |
| BBS | Return to BBS. |
| C | Enter the local conference. |
| DX # x z | Enter DX spotting data. An HF frequency (#) for a DX callsign (x) with any additional information (z). |
| H | List available user commands. |
| SH/CL | List basic stats about DX network. |
| SH/CO | Lists DX nodes in the network and users on each node. |
| SH/Dx/# | List last # of DX spots. Default = 5. |
| SH/U | List local DX node users. |
| SH/WWV | Lists WWV reports (max of 5). |
| T x | Send your following lines to a specific user (x). |
| T x zzz | Send a single line message (zzz) to a specific user (x). |

### File Transfer/Manipulation Commands

| | |
|---|---|
| D x | Download a file (x). |
| D y/x | Download a file (x) which is contained in a directory (y). |
| G | Initiate file search routine. |
| UP | Upload a file. {You must be authorized by the Sysop to upload files} |
| W | List what files and/or directories are available for downloading files from. |
| W y | List what files are available for downloading in a specified directory (y). |

### MSYS Network Node Commands

Note: May not be available on all systems/ports.

| | |
|---|---|
| BBS | Connect to Bulletin Board. |
| B | Disconnect from node. |
| C# x | Issue a connect request to a node or station (x). # is required when connecting to a user station to identify the port number to be used. |
| H | List available user commands. |
| I | Display basic information about the node. |
| J # | Display callsigns of stations recently heard by the node. [optional, port #] |
| K # | Display KaNodes (i.e.: Kantronics) heard by the node. [optional, port #] |
| N | List network nodes heard |
| P | Display description of each port. |
| R | Display routes used to get to nodes. |
| T | Page the System Operator. |
| U | Display list of stations connected to the system and their current activity. |

# W0RLI BBS User Command Summary

### General Commands

| | |
|---|---|
| B | Log off the PBBS. |
| F | Displays list of destinations for messages waiting to be forwarded. |
| I | Display hardware configuration of PBBS. |
| N x | Enter your name (x) in the system. |
| NE | Toggle Expert User status. |
| NQ x | Enter your QTH (x). |
| NZ x | Enter your Zip Code (x). |
| NH x | Enter the callsign (x) of the PBBS where you normally receive mail. |
| S | Display PBBS status including what connects exist and what they are for. |
| T | Page the System Operator so that you may talk to him. |
| V | Display PBBS software version number. |

### Message Commands

| | |
|---|---|
| CM # x | Send a copy of a message (#) to a station (x). |
| E # | Edit message header for a message (#). |
| K # | Kill message number #. |
| KM | Kill all messages to you that you have read. |
| L | List new messages since the last time you listed messages and then logged off the PBBS using the B command. |
| LA # | List oldest # messages. |
| LB | List bulletins (ALL OF THEM!!!) |
| LC | List bulletin categories. |
| LM | List messages to or from you (List Mine). |
| LP | List personal messages. |
| LT | List NTS Traffic messages. |
| LL # | List the last # messages. |
| LU | List messages to you which you have not read. |
| L< x | List message from a station (x). |
| L> x | List message to a station (x). |
| L@ x | List messages with a specific coverage area or specific PBBS (x). |
| L # | List messages numbered geater than or equal to #. |
| L #1 #2 | List messages numbered greater than or equal to #1 and less than or equal to #2. |
| L"xyz" | List messages with the specified string (xyz) in title. {case insensitive} |
| L'xyz' | List messages with the specified string (xyz) in title. {case insensitive} |
| R # | Read message number #. |
| RH # | Read message number # displaying full message header and forwarding info. |
| RM | Read all messages to you which you have not already read. |
| SB x @ y | Send a bulletin message to a target audience (x) within a specified coverage area (y). |
| SP x @ y | Send a private message to a station (x) at a specific PBBS (y). |
| SR # | Send a reply to the author of message number #. |
| ST # @ | NTSxx Send an NTS Traffic message to the specified Zip Code (#) in a specified state (xx). |

### Help Commands

| | |
|---|---|
| ? | Display a short summary of all PBBS commands. |
| H x | Display extended help information for a command (x). |
| H * | Display COMPLETE Help Documentation. |

### File Transfer Commands

| | |
|---|---|
| Dy x | Download a file (x) which is contained in a directory (y). |
| Uy x | Upload a file (x) to a directory (y). |
| W | List what directories are available for downloading files from. |
| Wy | List what files are available for downloading in a specified directory (y). |

# AA4RE BBS User Command Summary

## General Commands

| | |
|---|---|
| B | Log off the PBBS. |
| DU | Display user status |
| J | List TNC ports - Lists what tnc ports are in use by the PBBS you are connected to. |
| JL | List Connected - Calls of users that have connected lately to the PBBS. |
| JN | List Presently Connected - List calls presently connected to the PBBS by port "A,B,C....and stream 1,2,3....". |
| JP | Calls Heard - Where "p" is the tnc port identifier. Gives a short list of stations recently heard on that port. Typing "JA" will list those stations monitored on port A. Typing "JB" will list those stations monitored on port B, etc.. Use the J command to list what ports are active. |
| NN | Name - Use as NN *yourname* to enter or change your name in the database. |
| NH | Set Home BBS - Use as NH homebbs to enter or change your home BBS. |
| NE | Toggle *Expert* mode to enable short prompts and messages. |
| NZ | Zipcode - Us as NZ *zipcode* to enter your zipcode. |
| T | Talk to Sysop |
| V | Version - Shows version of BBS software |

## Message Commands

| | |
|---|---|
| K x | Kill msg #x. Also K x x x x to kill multiple messages. |
| KM | Kill Mine - Kills all messages addressed to your callsign that have been read. |
| KT # | Kill Traffic - Kills NTS traffic, even if not addressed to you. |
| L | List any messages that have been posted to the BBS since the last time you used the L command. |
| LM | List Mine - List message to you |
| LN | List New - list any messages that haven't been read. |
| LU | List Unread - list messages to you that you haven't read. |
| LL x | List Last *x* messages. |
| L x | List only messages above number *x*. |
| LA | List type A bulletins |
| LB | List type B bulletins |
| LT | List Traffic |
| LF | List Forwarded mail |
| LY | List All messages |
| L@ call | List AT - List messages addressed to BBS *call*. |
| L> call | List all messages sent to *call*. |
| L< call | List From - List all messages sent by *call*. |
| LD > x | List by Date - List all messages newer than date *x*. Use yymmdd. |
| LD < x | List by Date - List all messages older than date *x*. Use yymmdd. |
| L$ x | List BID - Lists messages whose BID is *x*. |
| LS x | List/Search - Lists messages with subjects that match pattern *x*. For information on patterns, Type H ! at the BBS command prompt. |
| R x | Read Message #x. Also R x x x x to read several messages. |
| RA | Read All Type A Bulletins. |
| RB | Read All Type B Bulletins. |
| RT | Read All Traffic (NTS). |
| RH x | Read with Headers. |
| RM | Read My Messages. |
| R$ x | Read Bid# - Read messages whose BIDS match *x*. |
| RS x | Read Search - Read messages whose subject have pattern *x* |
| R> call | Read To - Read messages to *call*. |
| R< call | Read From - Read message from *call*. |
| R@ call | Read At - Read messages addressed to BBS *call*. |
| RD > x | Read msgs newer than date *x*. Use yymmdd. |
| RD < x | Read msgs older than date *x*. Use yymmdd. |
| REPLY msg# | Generate a reply to the originator of msg#. Same as SR command. |
| SP x @ y | Send a personal message to *x* with optional destination BBS *y*. |
| SB x @ y | Send a bulletin to target audience *x* with specified coverage area *y*. |
| SR x | Send Reply to message #*x*. Same as REPLY. |
| ST x | Send an NTS traffic message to the specified zip code *x*. |

## Help Commands

| | |
|---|---|
| H | Gives a short page of help including info on how to get more help. |
| H*L* | Help for Command *L*- Detailed HELP with individual system commands. For example, type "H U" for HELP with uploading. The descriptions in this guide are similar to what you would get with this command. |
| I | Gives short page of information about the particular BBS |

## File Commands

| | |
|---|---|
| W | What files - list all of the file directors on the BBS. Also shows space remaining. Don't overflow by uploading too much! |
| WD dir | What w/Date - List files in specified directory *dir* along with the time stamp. |
| WX dir | What Expanded - List files in specified directory dir along with timestamp and size. |
| D d f | Download file *f* from directory *d*. Requests download of a file. |
| DB p d f | Download binary file specifying protocol *p*, directory *d* and file *f*. |
| U d f | Upload file *f* to directory *d*. Upload a file, end with ^Z or /ex. |
| UB p d f | Upload binary file specifying protocol *p*, directory *d* and file *f*. |

# Sample BBS Sessions

This is a couple of sample BBS operating sessions using a MSYS BBS. In the first session I'm checking in as a new user. In the second session I'll already be registered so I won't have to do it again.

```
cmd:my ka2dew
MYCALL   was KA2DEW
cmd:c potsdm
cmd:*** CONNECTED to POTSDM [03/29/91  11:50:25]
c bbsjxi
POTSDM:K2CC-1} Connected to BBSJXI:KA2JXI
[MSYS-1.11-H$]
Hello ?, Welcome to KA2JXI's MSYS BBS in Ogdensburg, NY.
Welcome! As a new user you will see this information. Next time you connect
you will go directly to the BBS command prompt. Please register as requested
below. The home bbs you give must NOT be a personal (built into a TNC) BBS
but rather a well known full service bbs that does mail forwarding. If you
haven't picked a home bbs yet, you are welcome to use this one. Just enter
its call with the NH command.
Please use N command to enter your name
Please use NQ command to enter your QTH (City and state)
Please use NZ command to enter your Zip or Postal Code
Please use NH command to enter call of BBS you want your mail sent to
Enter command: A,B,C,D,G,H,I,J,K,L,M,N,P,R,S,T,U,V,W,X,?,* >
n Tadd
Enter command: A,B,C,D,G,H,I,J,K,L,M,N,P,R,S,T,U,V,W,X,?,* >
nq Colton NY
Enter command: A,B,C,D,G,H,I,J,K,L,M,N,P,R,S,T,U,V,W,X,?,* >
nz 13625
Enter command: A,B,C,D,G,H,I,J,K,L,M,N,P,R,S,T,U,V,W,X,?,* >
nh ka2jxi
Enter command: A,B,C,D,G,H,I,J,K,L,M,N,P,R,S,T,U,V,W,X,?,* >
ll 4
MSG # TR  SIZE TO     FROM   @BBS   DATE   TITLE
2999 B$  2120 SAREX  KD2BD  AMSAT  910328 STS-37 SAREX Introduction
2998 B#  1832 CARF   VE7VCA ALLCAN 910328 INTERNATION CONTEST CALENDAR
2997 B#  1015 ALL    VE7EMD ALLCAN 910328 Help Find Sin-Cosine Pot
2978 B$  2127 ALL    N2KPK  ALLBBS 910327 RE:PK232/TS-440S
Enter command: A,B,C,D,G,H,I,J,K,L,M,N,P,R,S,T,U,V,W,X,?,* >
sp wa2wni
Enter title if local, CITY STATE & POSTAL CODE if not local:
on the air from upstate
Enter text, end with ^Z or /EX, ^A to abort
Dana,
  It's been a while since I've been on packet.  I had to re-register with JXI.
I'll be home all weekend so perhaps you can connect to my station.  I'll leave
it on the POTSDM node's frequency.  The route is still via PLB, PAN, OGDENB
though.  Good luck on the 145.01 stuff!
    Tadd
/ex
Enter command: A,B,C,D,G,H,I,J,K,L,M,N,P,R,S,T,U,V,W,X,?,* >
b
KA2DEW de KA2JXI: 73 --- Roger
*** DISCONNECTED [03/29/91  11:50:25]
```

That was an operating session last spring. When I did a ll 4 I was asking the BBS to *list last 4* messages. I could have asked for last 100 although that would have taken longer.

Now here is when I got on earlier this week. This time you'll see that I got some new mail, read it, deleted it and then disconnected.

```
cmd:c potsdm
cmd:*** CONNECTED to POTSDM [10/21/91  11:58:03]
c bbsjxi
POTSDM:K2CC-1} Connected to BBSJXI:KA2JXI
[MSYS-1.11-H$]
You have unread mail, please kill when read:
MSG # TR  SIZE TO      FROM   @BBS   DATE   TITLE
10501 PN   289 KA2DEW N2CGY   ---   911020 Vacation
Enter command: A,B,C,D,G,H,I,J,K,L,M,N,P,R,S,T,U,V,W,X,?,* >
r 10501
MSG # TR  SIZE TO      FROM   @BBS   DATE   TITLE
10501 PN   289 KA2DEW N2CGY   ---   911020 Vacation
Tadd,
  We had a wonderful time on the trip and are back safely.  Pictures should
be arriving at your address in a week or so.  Drop me a note with when you
will be back down next time.
  73 from all back home
*** END OF MSG # 10501 from N2CGY @ WA2PUU.#ENY.NY.USA.NA
Enter command: A,B,C,D,G,H,I,J,K,L,M,N,P,R,S,T,U,V,W,X,?,* >
k 10501
Msg 10501 Killed
Enter command: A,B,C,D,G,H,I,J,K,L,M,N,P,R,S,T,U,V,W,X,?,* >
b
KA2DEW de KA2JXI: 73 --- Roger
*** DISCONNECTED [10/22/91  02:00:29]
```

It is easy to read mail. Each time you check into the board you should do a list last 20 to see if there are any new messages that apply to you that might not have been directed to your callsign. To read any message you just specify the message number in the r command. Don't kill other people's messages though. Actually if you try the BBS will simply tell you that it isn't your message. There is very little you can do to mess up a BBS without *really* trying.

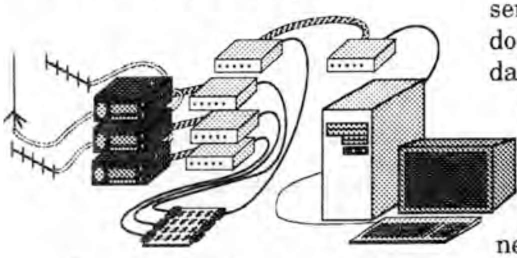
# NEDA: What's a node?

A node is an active location in a network. A network is a collection of nodes which allow data to be carried from place to place. Each node consists of 1 or more ports.

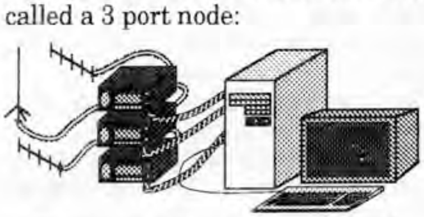
For this discussion I'll first break down types of ports and then try to give a brief description of the different kinds of nodes.

## Ports

In most cases a port is where a radio hooks up to a node. If a node has two ports then it usually has two radios. Sometimes a node has a port that isn't hooked to a radio. The most common case is where the node is at the same site as another computer that is used on packet. The computer talks to the node by a wire link instead of by a radio link. In this case there is a port on the node which has no radio hooked up to it. The pictured node would be a four port node:



In many cases a node is constructed out of a PC. The radios may be connected to the PC via TNCs or may be connected directly to modem cards in the PC such as the case with DRSI cards. In these cases there is almost always an application running on the PC, such as a BBS or DxCluster. The application is not considered to be a port, even though that may be a destination of data. The pictured node would then be called a 3 port node:



How a port is seen by a user depends entirely on the type of software used for the node. See *Types of Nodes* for more on user interface.

### What a radio talks to

Ports are described by what they talk to. What they talk to is described as users, servers and nodes.

#### *Users*

A user is a station which mostly accesses information from the network and sends short packets into the network. Personal home stations and EOC (Emergency Operations Center) stations qualify. User stations predominantly connect into the network and access information. If a user is to send information into the network the information is sent slowly, as with a keyboard, or infrequently, as with a file or mail message transmission. Very few users send more than one file per day into the network. Most will send about one long packet every minute when they are very busy. Personal Message Systems (PMS) usually qualify as the PMS usually is sent to by a server *from* the network. The PMS doesn't send more than one file per day, usually.

#### *Servers*

A server is a station that offers a service to more than one individual. Servers are connected to by users and by other servers. Some of the servers on packet radio today are *Bulletin Board Systems* and *Mail Boxes*. Both serve similar functions as message stores and file stores. Users connect to these servers and read bulletin messages, download information files, or send and receive messages with friends. Other kinds of servers are conference bridges which allow many users to communicate in a round table, and real time information resources which allow users to participate in the acquisition and dispersal of data. One common use for this kind of server is Dx spotting.

#### *Nodes*

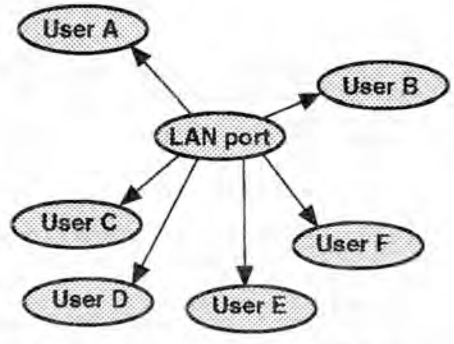
A node is a server that is used as a real time switch by many users and servers. As a real time switch any messages that are stored in a node are only there for a matter of seconds until they can be passed to a destination user or server, or to another node on the way to the destination.

#### *Describing Ports*

So, ports can be described as user ports, server ports or node ports, or a combination of the three. Good networking practice has it that ports should be configured based on the access requirements of the stations that it talks to, not on the type of stations they are. The ports are divided into two classifications: Ports where stations only receive data from the network (user LAN port), and ports where stations both send and receive data from the network (dedicated point to point link port). If best network design practices are followed, then for any kind of network node or server, these two port types are the only port configurations that need to be considered.

#### *Local Coverage User Port*

A user port is where a user connects to access the network. Local coverage user ports are used exclusively by stations that are either keyboard operated or that are receive-only stations, i.e. *users*. Local coverage user ports are on frequencies chosen to avoid co-channel occupancy with servers and other node ports. The local coverage user port is very efficient in that there are very few incidences of collision. The reason that this is true is that if all users can hear all transmissions made by the local coverage user port, and




Even though *user A* can't hear *user E* they aren't likely to collide as they spend most of their time listening to *LAN port*.

users very rarely send data, except as an acknowledgment of data transmitted by the local coverage user port, then there will be very few transmissions that *could* collide. See *Hidden Transmitter Syndrome*.

### *Flat Network Backbone*

There are many ways to make a backbone. The most compromising way (and the most popular) is to put a radio on a frequency and label it a backbone. Other nodes and servers will have radios on the same frequency. In most cases there are radios using the frequency that can only hear some, and not all, of the other radios. This kind of networking is called a flat network. The backbone links will perform well only when the data throughput required is very low. When the data throughput on the channel creeps up to a value that is somewhere around 20% of the potential throughput the performance on the channel drops sharply. Eventually the stations that have transmit data will give up due to retries or time-outs. This is called catastrophic throughput failure. The 20% figure depends on what stations are doing the talking and whether they are heard by other stations that need to talk but is a good ball-park value. One of the worst results of catastrophic throughput failure is that it is very frustrating to keyboard/real-time users.

### *Hidden Transmitter Free Backbone*

A better way of making a backbone is to make sure that all of the radios on the frequency can hear each other. In this case no radio will transmit when another radio is already transmitting. This provides a performance increase of better than a magnitude over the previous method. Most systems that are set up this way use a repeater to assure that all backbone radios can hear all of the other backbone radios on the channel. All but one of the sites would be a standard transceiver. The one site would be the repeater. The major flaws in this kind of networking are that collisions may result when some of the stations are aggressive and key up at the same time. Also the throughput on the channel is shared amongst all of the stations. If the backbone is optimized for performance under heavy load it will not perform as well under light load unless some sort of compensating back-off is utilized. This is a feature that TheNET does not have, unfortunately.

### *Dedicated Point to Point Backbone*

The best way of making a backbone with standard transceivers is to set up dedicated *point to point* links such that each backbone port talks to one other backbone port. If only two backbone ports are on a frequency in a given area the Tx/Rx cycles of the two nodes will toggle gracefully and throughput is maximized during any kind of loading situation. The backbone port digital hardware is optimized for maximum transmit and receive response based on the other radio being used on the frequency. The performance increase seen using this approach is easily worth the increased investment of having a separate port and radio for each node to node link, over either of the two compromise methods discussed in this section.

### *Server Multi-access Port TCP/IP*

In some networking situations it is uneconomical or unfair to designate stations which are part time as servers and force them to provide point to point links. This is the case where a station wants to operate as a TCP/IP host. A TCP/IP host is not likely to be happy on a local access user port, forced to be a receive only station. TCP/IP is just too powerful and neat to be under that kind of restriction. On the other hand it is rather expensive to have to fund both ends of a dedicated link for what might very well be a short term toy. Many would choose not to toy with TCP/IP *at all* if this was a requirement. Thus the Server Multi-access Port. This kind of port is operated in a hidden transmitter free fashion if it's going to work credibly. It is on 220MHz, or 420 and above. Because of the range limitations that are imposed by a simplex HTS free LAN, builders tend to opt for a repeater environment as described above. There are now many TCP/IP environments successfully using a repeater. One of the TCP/IP stations on the LAN is designated to be the Network <-> TCP gateway if more than a few TCP/IP stations share the LAN. Only that gateway station is propagated using the TheNET protocol. That usually doesn't cause a major problem for the TCPers and it keeps the node routing tables short on the network. On a sparse network where there are few nodes in the node routing tables all TCPers on the repeater would use TheNET protocol to talk into the network.

### *Wide Coverage General Access Port*

In some areas there is so little population that local coverage user access ports are not justifiable or affordable. Also there are areas where packet equipment was placed on high mountains and hasn't been upgraded to local coverage equipment due to lack of interest, lack of financing or because it is just impossible to subdivide the coverage because of very rugged terrain. In some existing packet systems there are systems that are entirely based on single frequency 2meter nodes. A port that faces such an environment but that is part of a node stack such as we are describing in the Annual is called a Wide Coverage General Access Port. This kind of port should be avoided in all new systems and should be made redundant as soon as possible in existing systems.

## Types of nodes

**Digipeater:**

This is not referred to as a node by packeteers although technically it is a node. This kind of node accepts traffic from a packet station and relays the traffic to another packet station. Only traffic from one station to one other may be stored in the digipeater at a time. The digipeater will ignore other traffic until it can deliver the stored traffic. The digipeater does not acknowledge traffic handed to it, rather it is up to the destination station to take care of this. Thus if the digipeater is unable to deliver the traffic (the destination station gets QRM or signal too weak) it is up to the sending station to regenerate the data (retry).

### Single port TheNET, NET/ROM, G8BPQ or MSYS:

This kind of node will accept a connect from a user station and allow the user to request a connect to another node or to another user. Data from a user is acknowledged by the node and then delivered to the next node or destination station. If the destination station or next node doesn't acknowledge the packet it is resent by the node.

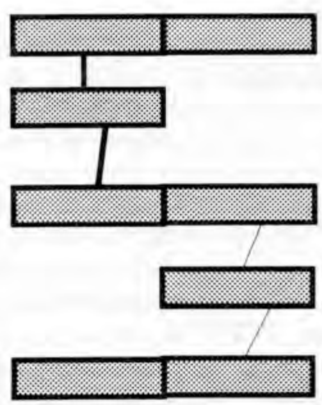
Single port nodes broadcast lists of known nodes so that each node knows of all of the surrounding nodes. Taking advantage of this the user can connect to a local node and then request a connect to a 'next node' that may be several nodes away. The node will recognize the connect station as another node and will attempt the connection via whatever route is required to make the path. The path is usually selected based on the automated nodes broadcasts so sometimes the single port node may be tricked into using a path that is unreliable or might not work at all! (Covered in more detail later in this document) Most single port nodes are individual efforts and no system wide design philosophy is used so that many if not most paths between nodes are unreliable. It is always true, however, that a multi-node path will require traffic between the nodes. This traffic will be on the user's frequency, thus causing all of the users within range of the multiple nodes to be delayed. This also leads to tremendous occurrences of hidden transmitter syndrome (described later).

### Single port KAnode:

This node is similar in operation to the single port TheNET and NET/ROM nodes except that automatic generation of node lists is limited to the adjacent nodes. This means that any connect to a 'next node' will not be via any intermediate node. KA-node does support automatic use of digipeaters between nodes however it is not compatible with TheNET, G8BPQ etc..

### Multiple port TheNET, NET/ROM, G8BPQ, MSYS:

Like the TheNET and NET/ROM nodes described above this node allows the user to connect and then request a 'next node'. An important difference is that the 'next node' may be on a different frequency! These nodes consist of 2 or more independent ports, each port being a separate digital section and radio. The ports communicate via wires and are located at the same site. Connection between the ports is usually at 9600 or 19200 bauds (as compared to the usual speed of radio communications at 1200 bauds). Thus operation from one port to another at the same node site is nearly instantaneous.



Using multiple port nodes it has been possible to have packet users connect between frequencies transparently to access other users and automated packet stations (like mailboxes and bulletin boards). One idea that sprang out of this capability is the concept of the backbone.

### Early Backbones

A backbone was taken to be a frequency on which two or more nodes communicated. This frequency would be other than two meters and would exclude keyboard users. This was so that there would be fewer 'hidden' sources of data. It was presumed that this would improve the performance of the remaining stations and nodes.

Lack of performance on backbone circuits is proportional to the volume of data produced by hidden stations not the number of hidden stations.

### Backbones: What happened?

What has happened is that each region would set up a backbone channel such that long haul traffic could be moved off of 2 meters. Some of these backbone channels were set up by bulletin boards (before TheNET and NET/ROM) so that traffic could be sent to other BBSs without interference from individual non-bulletin board stations. A popular example of such a backbone implementation was the common use of 221.11 and 441.0 in the northeastern states.

Since the advent of multiport nodes it has been possible for traffic to pass from one regional backbone to another in a single point to point connect. I.E. a station from Connecticut could connect to a station in Maine by connecting from his 2 meter radio to a node that had 220 capability which would talk across 220 to a node that had both 220 and 440, and then across 440 to a node that had both 440 and 2 meters, and then back to 2 meters. As time went on each of the backbone frequencies (and there were only several) has gained in quantities of nodes and quantities of data. Originally there was no attempt to control hidden transmitters and precious little to control erroneous path generation due to unbalanced transmitters verses receivers at each site. (Automated node broadcasts, remember?).

## NEDA network node

A NEDA node consists of all of the interconnected TNCs, computers, radios and associated hardware at the single site, which performs the switching functions for it's piece of the network. The definition holds true for whatever type of networking software in use. i.e. NOS, MSYS, TheNET, NET/ROM, ROSE, TEXNET, G8BPQ etc..

These nodes have multiple ports, at least one of which is a backbone port. The backbone ports talk to other backbone ports such that packet data can travel from node to node on non user frequencies. (This way users are the only stations that have to share user frequencies) The important difference between this network node and most other multiport nodes is that the backbones in this system are maintained as hidden transmitter free point to point links (See page on hidden transmitters later in this package). This is done simply by supplying a separate set of radios on an independent frequency for each backbone. In concept this is extremely simple and obvious. This incurs several disadvantages however.

### NEDA node: Disadvantages

The first is obviously cost. Each of the sites houses at least 2 sets of radios and TNCs, most sites must have 3 or more. Most have 4 or more sets. That's a lot of radios and TNCs.

The second disadvantage is that each frequency must be chosen to not have interference from any other station, at a different site or at the same site. It's difficult to have 2 backbone frequencies coming into a single site on frequencies that are near each other. There are really only 2 bands on which backbone links are conveniently constructed, 220 and 440. If a site has to have 3 or more backbones it is important to maintain frequencies that are separated when in the same band and radios and antennas that will not interfere. This becomes a technical challenge. Also there should be a different frequency for each backbone in a given region. That becomes an administrative challenge. In order for the system to work well there cannot be other stations operating on the backbone frequencies. That becomes a public relations challenge.

### NEDA node: Advantages

So, why bother? The first answer is performance! There is at least a 5 times performance increase using the same baud rates and hardware as a flat network. If only two stations are talking to each other across the backbone the timing values in a TheNET or equivalent node may be maximized for that situation. That can lead to a 20 times performance increase, or more, over a non hidden transmitter free backbone. That's a 20 times improvement (at least!) with a 1.5 times increase in site cost. Not too bad a trade off.

The second answer is marketability. Once built in the way described in this document a network is usable by keyboarders, Dx spotting networks, BBS stations for forwarding, users for access to BBSs, TCP/IP hosts, etc. Because network site to network site transport is moved off of 2 meters that band may be used for low power personal keyboard stations. The amount of fun and interested had by all of the network users and participants goes way up. A 60% of all hams involvement rate is achievable. This will allow a very successful packet program to exist.

The third answer is adaptability. It is easy to replace or reconfigure all of the ports on a single backbone channel with many fewer complications because there are only two sets of equipment involved. A performance increase on all ports involved could be achieved by adding 4800 baud modems at both of the sites.

It's not really that hard to add additional ports to a single or dual port node. For example at sites with limited antenna space it may be possible to use dual band and diplexed antenna systems to achieve multiband operation. Most backbone links run relatively low power and directional antennas which make it easy to keep the transmitters out of the receivers.

### Backbones using repeaters

Instead of supplying separate sets of hardware for several backbone channels at a site it is possible to do backbone linking through a repeater. The repeater would be placed at one site and then several others would access it. There are several advantages and disadvantages to this.

#### *Repeaters: Disadvantages*

• Each site that links to the repeater must have a dedicated radio on the same frequency pair as the repeater. This ties up two frequencies and the same pair must be used at each site. This is instead of being able to choose a different link frequency from the 'main' site (that would have the repeater) to the other sites based on the spectrum availability at each of the sites.

• The total data throughput (added for all site/rptr links) is going to be less than the baud rate (which must be the same on all links to that repeater). If the N links were on independent frequencies the throughput is theoretically N *times* the baud rate for a system of N sites. This is because a different set of data can be traveling on each of the links at one time instead of only one set of data as in the case of the repeater.

• Collisions can occur between stations accessing the repeater if they both key up at the same time.

• One station on the repeater can set his timing values such that the one station exceeds the theoretical throughput for the average station. This will improve that station's timing but drastically reduce the total throughput.

• The repeater turn-on time is added to TXDelay for the remote stations in many repeater arrangements.

• If the repeater should fail all sites that depend on it loose connectivity.

• Throughput on a repeater is only going to be good enough for a backbone between expandable node sites if the repeater channel baud rate is much higher than the baud rate needed on point to point links. High baud rate repeaters are much more difficult to build and maintain than low baud rate point to point links.

***Repeaters: Advantages***

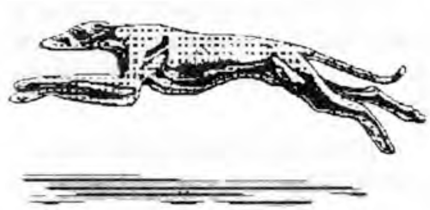
• The total packet travel time on the repeater, if not overloaded, would be half that of a separate link system.

• The cost of the system is one full duplex radio and 1 normal radio per site as opposed to 2 radios times number of sites.

• If a central link site is necessary and that site is a commercial radio installation it might not be practical to have several link radios. A repeater only needs one antenna.

Note on repeater usage: All stations involved in the repeater operation must be using modem tone DCD, not noise level DCD unless the repeater has a very short keyup and unkey delay.

Repeaters should be used on user ports ***or*** for tying part time or low usage servers into the network where the servers aren't feeding other servers. In other words, if a server has only one port then it could be tying into the network via a repeater. If a server has two ports then it should be using a point to point link to get into the network. For user access to the network, repeaters are excellent but only if the repeater is exclusively for low duty cycle users. Users sharing a repeater with a server is not a good idea.

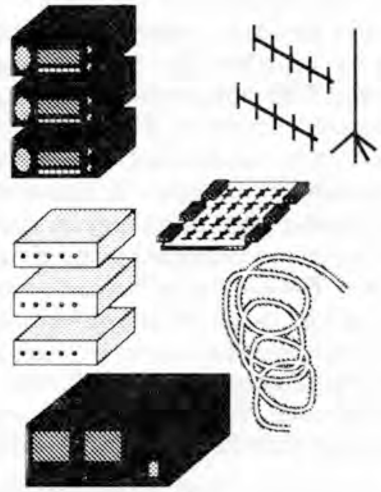


## What does it take to make a NEDA node?

The most common node configuration consists of:

- two UHF or 220 FM radios,
- a 2 meter radio,
- two yagis,
- an omni,
- three TNCs,
- three runs of coax,
- a power supply,
- a diode matrix.
- three ROM chips with TheNET,
- miscellaneous serial cables, mike cables and power cables.

The omni is 6dB gain or less for the 2 meter user port. Depending on the population density in your area your user port may be very low power, even to the point of having attenuators in the receive path.



***User Port Recommendations***

Most sites have a user access simplex 2 meter user port. The user port is geared to cover about 50 packeteers. If the normal ratio of hams to non hams is 1:600 and the normal ratio of packeteers to hams is 1:2 then the population area that your user port should cover should be less than 60,000. If there are 5 frequencies available on 2 meters then a city of 300,000 could be covered with very little concern for over-coverage. Be sure to take rural areas into account when calculating coverage. It is important that the number of packeteers (stations that receive-mostly) that must use a single user port is 50 or less. If more than 50 need to access the user port then attrition of packeteers will result. Thus it is important to watch for over coverage. It is also important that some frequencies are left available for expansion and experimental operation. It would be real silly of us to assume that we're using the *best* possible networking tools. Some frequencies must be left on 2 meters for high tech LANs. It should be stressed that 'high tech LAN' does not include a wide area coverage *mess*.

If a situation exists where there are more than ƒ× 60,000 population a directional antenna might be used, or attenuators for reduced range. Also keep in mind that the more LANs in the area, the better.

***Politics of User ports***

Radio hams accumulate in communities. A community is a group where the hams know, or know of, most of the other hams. If a 2m user port covers more than your radio community, packet growth will be stifled.

If someone wants to put up a node where there is already LAN coverage then space *must* be made, by dividing coverage with the aforementioned directional antennas. Telling someone to not put up hardware when they offer is as good as telling the FCC that we don't need an amateur service after all. The thing to do is to tell the newcomer *how* to use the hardware, not *whether* to. Think up some good way to use the equipment without compromising the network and without hurting the other users. Show them this book. Get it together!

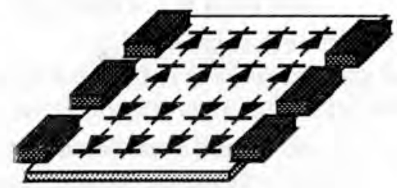
***Backbone Recommendations***

Each site has at least one dedicated point to point backbone port that talks to a single other node. The backbone port is on 50MHz, or 220MHz and above. Most of the backbone ports use yagi antennas pointed at the neighbor node to which the backbone runs. This way we have some hope of reusing backbone frequencies within the network. It is also desirable for practical and financial reasons to put more into the antenna so that sensitivity and output power may be minimized. The backbone port must be running software that is protected against unauthorized operation. The software we've been using mostly in the network is TheNET Plus v2.10st by Bill Beech and TheNET X1 by G8KBB. The parameters used in a backbone node are optimized for each end of the link. This allows us to often have less than 300ms transmit/receive overhead on a backbone link.

The most successful nodes have all ports based on TheNET software running in a PAC COM Tiny 2, MFJ-1270B or other TAPR2 compatible TNC. It is quite possible that in the future nodes will be based wholly on other kinds of equipment.

### Networking Resources

Companies have sprung into existance to sell hardware for network building. Amateur Networking Supply and NX2P Electronics sell diode matrix kits. A.N.S. also offers a wireline linking system. More on this later in the book (See *Building a Node*)



Recent vintage TNCs may be tied together at 9600 or 19.2K baud (only the PacComm will do 19.2K baud) across the matrix. The matrix is unnecessary for a 2 port node.

So, to make a 3 port node (user port and 2 backbone ports) you'll need the following:

2m rig, gain omni antenna, coax.

440 backbone rig, end mount beam or dipole, coax

220 backbone rig, end mount beam or dipole, coax

3 TNCs, 3 node chips,

Diode matrix, radio to TNC cable hardware (connectors included with most TNCs)

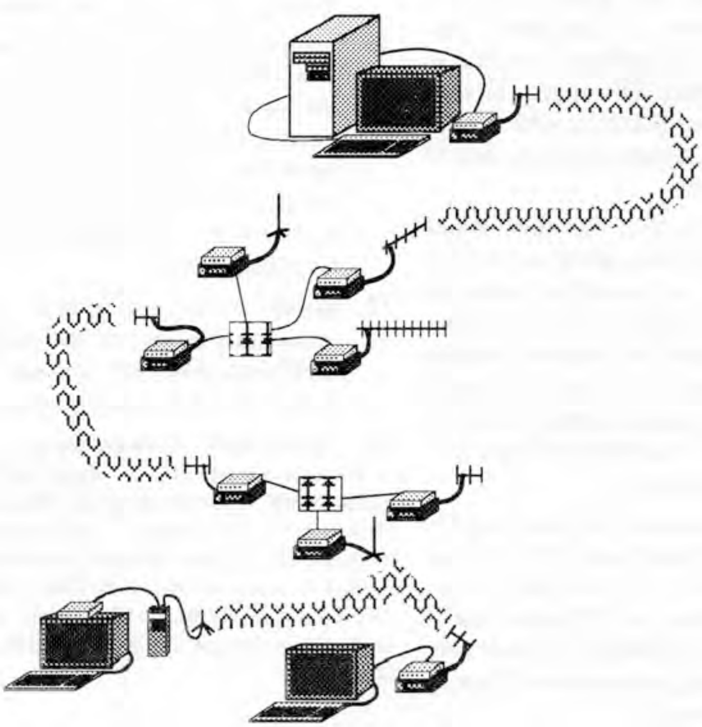
Power supply and control point mechanism.

#### *Other Comments*

Node sites don't necessarily need to be located on a big hill. Although some times backbone linking requirements dictated high sites, it is important that a user port can serve a specific geographic region of *users* with its coverage effectively and *not* interfere with adjacent packet systems using the same frequency. Some systems that are at high elevations tend to "hog" a given user channel over a larger territory than necessary. A better approach is to create small local area networks (LAN) with well defined areas of coverage. This also allows for the reuse of 2 meter user channels at nodes that are closer together without interference. Further efficiency is achieved by the fact that fewer users will access each user port. This is called the cellular approach to user ports.

Node site accessibility is also an important consideration for constructing a node. There are some serious advantages to putting nodes at the node owner's house:

- The node can be serviced in short notice and with little hassle.
- The operator can observe problems that might not be apparent from a remote station.
- Radio equipment that is not hardened for an outdoor environment will work fine at a home node location.
- The site manager may attach other systems to the node via a wire link (as opposed to radio link) such as a BBS, TCP/IP station etc.
- Diagnostics may be done to the station using equipment that one might not want to haul or leave at a mountain site.
- Node reconfiguration for experimental or emergency linking is possible.
- Christmas lights are unnecessary as the TNC light show is fantastic
- The node is available for demo for curious visitors.



## Network Concept

Now that you've heard what the nodes look like here's what led to this design.

Important considerations when first specifying network hardware and direction included:

- Network capacity;

*Can the network handle the capacity that will be imposed on it? This is partly determined by the existing network as the users will generally only expect what they have already gotten used to.*

- Network promotability;

*A good network is of little use if the target audience can't understand it.*

- Operator and technician availability to fit the chosen hardware and software;

*Using equipment or software that is obtained or built only with special talents or connections to build the prototype network will not lend to a successful network. So, either choose off-the-shelf components or start your network development by creating sources for the required materials.*

- Politics;

*Unfortunately this is an important consideration. In order for your budding network to survive it must allow for the participation by people who are greedy and egotistical. The network design must allow avenues for those people to help the network, without destroying it. In the same token it's important that no design rules are made which give special privilege, especially where this may be construed to be negative by any party.*

## Basic Networking Guidelines For An Amateur Radio Packet Network

If these rules are followed the participants will create a fun, expandable and upgradable packet network that will be the equal of any in the world. Compromising on these in any way will lead to limitations and eventual dead ends. These rules apply to amateur radio in the United States and may see some modification in other countries merely due to spectrum changes and government regulations. If a system is going to be created and used by hams, and depend entirely on ham radio, then these are good rules:

1. All backbones are dedicated point to point links. Backbones are on 220MHz and up. Backbones are *never* on 144MHz. See *Hidden Transmitter Syndrome.*
2. LANs have only a network connect. Just one node for access to the network and no more than 10 active users at a time during a standard operating period. Users connect to the node and then away from the node, either via the network or direct from the node to another users. No digipeating.
3. Servers connect to the network via dedicated links if they are high volume data generators or via hidden transmitter free shared links if they are not high volume data generators. Server links are on 220MHz and up. Server links are *never* on 144MHz
4. User access to servers is via the network. Servers shouldn't have any 2m hardware as they are connected to via the network. An exception is where users can gain network access via the server in which case: the server would be also be considered a user access port; the server frequency is not on the same frequency as any other user access port; users can get to the rest of the network from the server's user port. In any case servers don't need 2m hardware on the same frequency as another user access port within RF range. (Except for redundancy in which case the redundant radio is normally *disabled*)
5. Corollary of 2. Local coverage user LANs are designed so that they do not see other nodes or users of other LANs. LANs need to be low power so that this can work. No node to node communications may exist on 2 meters. See *Hidden Transmitter Syndrome.*
6. Locations and mechanical design of node housings are not important. Nodes may be on mountains or in homes. Backbone and dedicated links may be of high or low power depending on need.
7. Length of backbone hops is not important. Reliability and signal to noise ratio are important.
8. All nodes/users in the network must have standardized window sizes. They don't have to be the same but they must be agreed upon by all network level 4 data sources. This is important because otherwise network users will not have the same priority.
9. Any station that will transmit data at greater than 300 chars/minute is a server. Any station that provides services to stations over the radio is a server.
10. Link and backbone throughput should be improved as loading increases. 1200 baud is pretty good for backbones in a new network, if links are point to point, dedicated, and with no interference. See *Node Radio Considerations.*
11. Radio interference from on site equipment is just as bad as interference from off site equipment. It must not be tolerated.
12. Redundancy is important.

These are good rules for any amateur radio packet network using AX.25 based link layer protocols. Additional restrictions may be imposed by network software. Any comments on these rules should definitely be aired. Times and technology change. This is a good start though.

## Use 1200 baud first!

Any new network expansion should use 1200 baud point to point links for the following reasons:

- 1200 baud is misconstrued as being slow, especially by those who are only used to non-point to point links. A 1200 baud point to point backbone with 250ms keyup delays can pass 8 Megabytes per day.
- Any network linking equipment that is installed and working is worth substantially more than just the parts. That means that you can turn around and reuse the equipment somewhere else if you upgrade an existing link. Waiting to get faster equipment before putting in a link is silly. Put it in slow, first, rather than wait. Then upgrade as resources become available. A 1200 baud point to point link is far better than a HTS infested link channel.
- If you have a network that is promoted to all hams, you will find that some of them will desire to get involved and expand your network. This will lead to more network facilities and redun dancy. The more people buildin on the network the better it wil be for all.
- 1200 baud equipment is easy t work with. Once your system i up and running it is trivial t promote others in your area an in adjacent areas to add on to th system.
- Higher baud rates are not as goo as they may seem because factor like TXDelay are not changed jus because you up the baud rate They just become more obvious

## TheNET Network Development Guidelines

1. All backbones are hidden transmitter free and have backoff timing set so that consistant throughput results are obtained under heavy load. Dedicated Point to Point links are *highly* recommended (some network builders will not tollerate more than two radios on a link)
2. Time-to-live should be established network wide. Link qualities should be set so that the furthest propagating node is not as far as time-to-live. See *Modes and Parameters* in the *TheNET Guide.*
3. Windowsize parameters should be the same and should be a low number. The value should be such that during 50% loading the number of packets outstanding for a given connect is about 1 per 10 seconds of planned latency. This allows for approximately 23 characters per second through put for each connect. With 100cps throughput rate on backbones and a time-to-live such that three 1200 baud backbones are the maximum traversed in a L4 hop a windowsize of 2 is appropriate.
4. Nodes broadcasts and node broadcast acceptance should be turned off on local coverage user ports to discourage node to node communications on 2 meters. This makes it so that a node on your local coverage user frequency won't see a nodes broadcast from your node. The users will be able to get nodes lists etc.. Nodes propagation in and out of the node via the RS-232 port still works. This makes it so that if somebody decides that they want to put up a node and link into the network by your LAN port that they are discouraged from doing this, immediately. Nothing is so frustrating as to be informed of a rule only after breaking it. By setting up the node in this manner a potential abuse is stopped before it begins.
5. Retry rate/level 4 time-out should be the same for all nodes and should be greater than the worse time it takes to transmit the maximum length message the maximum number of nodes allowed by time-to-live and over the worst path in the network.
6. Nodes whose L4 retry rate, L windowsize and time-to-live ar not predictable must not be al lowed to take part in networl level 4 operation. All L4 networl hardware operators must agre on these figures. This make sure that all network users ge as close to an equal share of th network capacity as possible.

Level 4 operation means that th node can pass data to another node multiple hops away. If a station i denied level 4 access then it must d a simple connect to the adjacent nod and then connect on into the network This might be necessary in the cas of a non-compliant TheNET, MSYS NOS, G8BPQ.

7. Nodes broadcast limitation an nodes broadcast reception con trol are used to *limit* the level access of nodes that are not net work participants (i.e. who don' abide by the above rules, in cluding rule 6).

# How To Use The Network

Let us take as an example a new user. The user wants to get to a friend in Poughkeepsie New York whose call is K2QRM. Our user is located in Exeter NH. Assume for the sake of the example, that both are using 2 meter stations and have base station antennas. The station in Exeter is looking at his NEDA user port map and finds that KNGSTN:WB1DSW-5 is on 145.05. He dials his radio over to that frequency. Next he tells his TNC to connect to WB1DSW-5. He gets a message on his display that says:

```
*** Connected to WB1DSW-5
```

At this time, out of curiosity he asks the node for its INFO message and a list of nodes by doing the following:

```
I
KNGSTN:WB1DSW-5}
sysop WB1DSW & NR1N
  QTH East Kingston NH
 freq 145.05
 info NR1N@WB1DSW

download file NETWORK in the N directory
( DN NETWORK ) on WB1DSW BBS for more information about local nodes
```

then he types:

```
N
KNGSTN:WB1DSW-5} Nodes
```

| | | | |
|---|---|---|---|
| ARLNG1:K1CF-1 | ARLNG2:K1CF-2 | ARLNG4:K1CF-4 | ASH:KB4N-5 |
| BBSASH:KB4N | BBSDSW:WB1DSW-5 | BBSEDY:KA1EDY-1 | BBSGOZ:KA1GOZ-1 |
| BBSN:NS1N | BBSPHY:WA1PHY-1 | BBSWOK:WA1WOK-2 | BED:WA1PHY-3 |
| BELNAP:N1DCT-3 | BERK:WA1TPP-3 | BERK2:WA1TPP-13 | BERK3:WA1TPP-14 |
| BERK4:WA1TPP-9 | CENTNH:K1BKE | CHSTR:K1MEA-2 | CROWD:WA1WOK-7 |
| DENNIS:KQ1K-7 | NASHUA:KA1GOZ-9 | NBY:KA1EDY-5 | NHOEM1:WA1WOK-1 |
| NHOEM3:WA1WOK-3 | NSHORE:KC1PK-5 | NSHR22:KC1PK-3 | SALT:KQ1K |
| SCIT:NS1N-5 | SWNH:KA1BBG-1 | STMFRD:KB2CS-1 | SWNHU:KA1BBG-4 |
| WNDHM1:K1TR-1 | WNDHM2:K1TR-2 | VNH:WA1TLN-1 | YCCCDX:K1TR-3 |

The Info message can be up to 240 characters long and is set by the node manager. All of the nodes should have useful info messages. The nodes list is a table of node names and callsigns of the nodes that are available via the backbone within 3 hops and in some cases the nodes that are heard by the user port on the user port frequency. Referring to the map our user sees that CLV:N2CJ-1 is a node near Poughkeepsie NY. The route from KNGSTN to CLV is KNGSTN -> WNDHM -> CHSTR -> STMFRD -> KNDRHK -> CLV. Each step is via HTS free backbone links. Because the system only shows nodes three hops away the user must step to a mid-point node. STMFRD shows in KNGSTN's nodes list. He then types:

```
C STMFRD
```

if all goes well about 15 seconds later he gets the message

```
KNGSTN:WB1DSW-5} Connected to STMFRD:KB2CS-1
C CLV
```

and after 30 seconds gets the message

```
STMFRD:KB2CS-1} Connected to CLV:N2CJ-1
```

At this point our user can type

```
C K2QRM
```

and after about 35 seconds gets

```
CLV:N2CJ-1} Connected to K2QRM
```

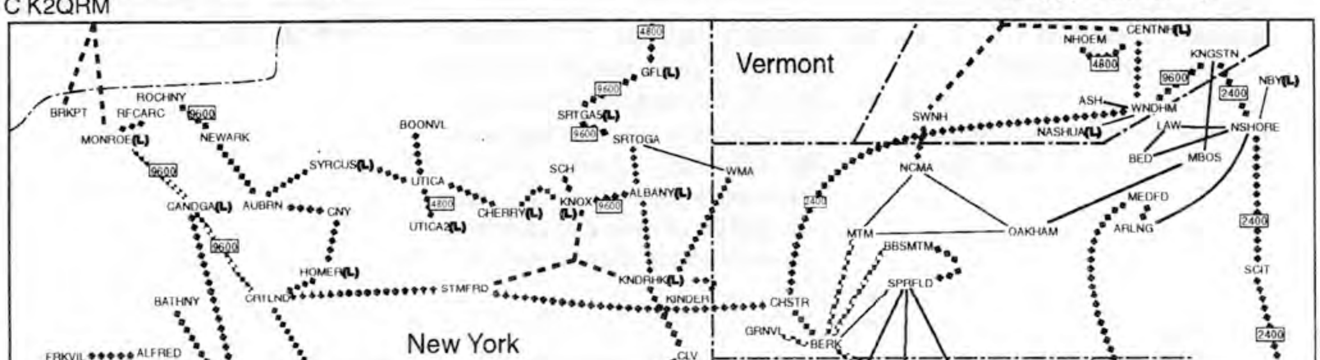
Now any text typed by our user will go to K2QRM and any text from K2QRM will come back to our user. To terminate, either party simply exits back to command mode using control C and types D as usual.

If our user desires he may connect back to KNGSTN node and then to STMFRD and type an N command. This will show the nodes that STMFRD knows about from it's point in the network. Note that the user ports on KNGSTN and CLV (as well as STMFRD) may be on different frequencies. It is only important that the frequency of the user port is the same as the users that are talking to it. The network is responsible for connecting user port to user port. In this case KNGSTN is on 145.05 and CLV is on 145.09. Even if they were both on 145.05 the network would still be used to pass the traffic. Also in this case KNGSTN and CLV are about 200 miles apart. At each hop the data travels from node to node via the backbones. Each backbone is on a separate frequency. All of this is transparent to the user.

When you establish a frequency for your station you should make note what network user port serves your station best. You can then tell your friends how to get to you from the network.

This document's purpose is to provide the information necessary to build a network of this type. Also included is more information on using the network, both as a user and as a server.

Happy Packeting!



# Hidden Transmitter Syndrome

This is the bane of most earlier packet networks. A system with 3 sites: Station A and Station C are far enough apart that they don't hear each other at all. Station A has traffic to go to station C and station C has traffic to go to A or B. Station A will transmit when it doesn't hear anything. Station C will do the same. Site B hears both A and C. If C is transmitting and A decides to transmit, both messages are lost. If A is waiting for a reply from B and C is talking, then A has to wait. If C is talking for too long, A will retry, thus trashing the message C is sending to B.

If the A to B link was on a different frequency than the B to C link, the observed performance increase is greater than 5 times, regardless of the baud rate! *A hidden transmitter is a station that can be heard by one or more stations on a frequency but can not hear ALL of the stations on the frequency.* It is the stern recommendation of NEDA to not allow hidden transmitters on backbones.

### Hidden Transmitters on User Ports

Clearly it is not possible to eliminate HTS on simplex user ports. It is certainly possible to eliminate HTS on a user port if the user port could take advantage of a duplex repeater. However, given that most user ports are simplex systems the effects of the hidden transmitter problem must be taken into account

If the user port node is the only station on frequency that can hear everybody and if all stations on frequency have parameters set to take advantage of this, the following is true:

Given that a user port can be heard by all stations on a LAN: *Time used for data transmission by the node is about 80% efficient. Time used for data transmission by user stations is about 80% efficient, under minimum load. Under maximum load the time used for data transmission by the node is still 80% efficient but time used by the user stations drops down to near 0% efficient. Thus it is useful to have sources of lots of data (i.e. automated stations like BBSs) to be sourced through the network and not from a normal user site on a 2 meter user frequency.*

For this reason NEDA requests that stations which source a lot of data (i.e. servers, hosts and BBSs) use dedicated link channels on 50, 220, 440 or above. This allows for the best of all worlds for the servers and for the users. If we create our network in this fashion we'll create a system which will grow and in which fun and learning are maximized.

### Minimizing HTS

***1. On Backbones:***

Keep each link hidden transmitter free. Try to configure backbones as dedicated end to end links using only two radios per frequency/link if on a high traffic path. High volume means that there is channel activity as much as 1/4 of the time during peak loading periods.

Set persistence values at each port on a backbone to 256/(N-1) where N is the number of nodes on your backbone frequency. Example: If 4 nodes on backbone frequency then use Persistence of 85 at each port.

Set slot time at each port on a given backbone equal to TXDelay for that port.

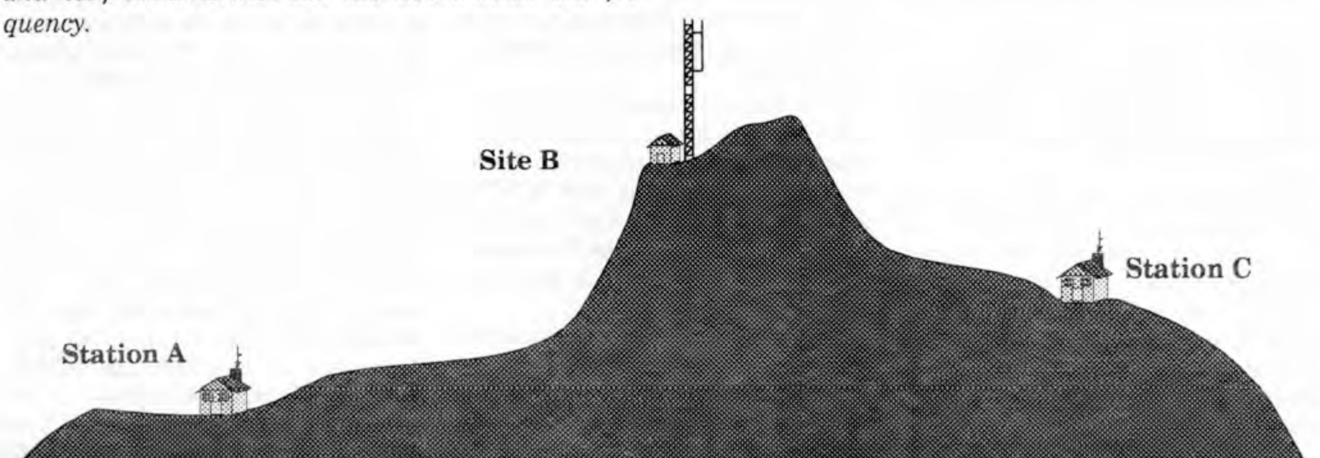
***2. Users that are high data volume sources:***

High data volume sources should minimize collisions with low volume users by setting up dedicated uplink channels to one or more local nodes. Arrangements might be made with other local high data volume sources to share a dedicated link port. This should not be on 2 meters.

***3. Users that are not high data volume sources:***

These users should arrange to use minimum power to assure full quieting signal at a NEDA node user port which has few or no co-channel nodes or servers. Observe the node in monitor mode and see what/who is uplinking to it. What/who is downloading is much less important. On channel DXing of user ports is seriously not a good thing.

Keep MAXframe low (1 or 2), and DWait at 16. CROWD conference nodes let you use PACLEN up to 240 so set it at 240.



# TCP/IP and Packet Networking

TCP/IP is one of the protocols that run across the TheNET network. The way this works is that TCP/IP hosts have dedicated links into TheNET node sites on 220, 440 or up. Then the TCP/IP host transmits a TheNET node ident into the network. The Idents all start with the two characters 'IP'. Thus IPWMA, IPQRP etc.. That node ident propagates up to 7 TheNETs away to all other similarly linked TCP/IP hosts. TCPers that want to access hosts across the network must route through TCP/IP hosts along the way. No end to end TCPing is supported as the node broadcast qualities in the TheNET network are only 7 hops. For new TCP systems where this would be a problem, compromises, either involving software hacks or additional appropriately located TCP hosts may be constructed. Check the user port maps for the locations of TCP hosts in the network. Check in the Quarterly for your local TCP/IP address assignment coordinator.

Some TCP stations access the network via TCP-only links into hosts which are already linked into the TheNET network. Additionally it is possible for non-TCP stations to access TCP hardware over the TCP-only network by doing TheNET connects into the TCP hosts that *are* listed in the nodes lists. If you use this method, after you connect to the TCP host, hit an extra carriage return. Then the TCP host will respond with a prompt.

The recommended network subnet for TCP/IP usage is that one or more TCP/IP hosts would have point to point links into TheNET nodes that are in the network. Those TCP/IP hosts would support a TCP/IP-only repeater. What this does is set up a subnet for TCPers which would be entirely TCP/IP. That allows the TCP stations to use backoff and P-persist without competing with AX.25 non-TCP stations. Additionally it allows the community of TCP stations to interact so as to further build packet activity. We should stress that backbones should be a cooperative venture between all packet users. It is probably true that there are other schemes which would work better on the long haul than TheNET. However, TheNET is the only protocol currently available that supports all kinds of packet users. If there is new information on this please present it to the editor.

Note to computer hackers: UHF link radios need not be expensive. All you need to do is find a RF hacker to help if you don't want to mess with it. There have been two separate UHF radio notices in the NEDA Quarterly for under $80 per rig. (published in this *Annual*) NX2P Electronics also sells a UHF link radio (from PacComm) that would do very well. It's in the $200 price class per rig.

**—End of File**

# What is TCP/IP?

Transmission Control Protocol/ Internet Protocol. TCP/IP defines a protocol suite. TCP/IP is a system of messages sent between computers, via radio (or telephone, or wire) that enable the computers to exchange data meaningfully. Where AX.25 is a protocol that defines how two TNCs can communicate, either directly or via digipeaters, TCP/IP is a protocol suite that defines how two computers can communicate, over wire line, telephone with modems, two or more TNCs, NET/ROM, TheNET, ROSE, etc. TCP/IP is called a suite of protocols because it actually includes hundreds of different message types and response procedures for dozens of different purposes.

Defined in TCP/IP as commands (and separate protocols) there are TELNET, FTP, SMTP, FINGER, PING and others that are of direct use to the user.

TELNET establishes a real time two way interactive connection between a user at his own computer and another remote computer. This lets the user command the remote machine as if he were sitting at the keyboard of the remote machine. This is similar in effect to how an AX.25 user perceives TheNET and BBS operation.

FTP or File Transfer Protocol is a customized command set for getting or putting files on a remote computer from the user's computer. Files may contain non-text information. This is a key feature of TCP/IP for amateur packet radio.

SMTP or Simple Mail Transfer Protocol is a system for automatically routing multi-line messages from one computer to another over any number of intermediate computers. Unlike FTP and TELNET which require that an end to end path must be established in real time SMTP allows messages to traverse the computers that are available and then wait for computers that are unavailable and then proceed when they come on line.

FINGER is a command whereby a user can ask a remote machine for information about another user. Thus I could do a finger on the user NQ1C on the machine NQ1C and get back that NQ1C is Bob Lafleur and whatever other information that Bob wants his finger file to contain.

PING is a command to send a packet to a remote computer to find out if it is connected to the network and if so how long it takes to get a packet there and back.

There are many other useful protocols built into TCP/IP that allow such things as data sharing between programs running on two different computers, identifying what hosts are available, finding out the time at a remote machine, authenticating passwords and even passing silly quotes.

Message routing with TCP/IP is based on a 32 bit address and aliasing. Each host computer is given it's own specific 32 bit network address and a text alias. The text alias for amateur TCP/IP is usually callsign.ampr.org. The .ampr.org is used to differentiate the amateur network with the commercial networks in cases where there are tie ins between the two. The 32 bit address if of the form 255.255.255.255 where each of the numbers is called an octet signifying that it uses 8 bits. Each of the four octets from left to right decreases in priority. The first octet is used to determine whether the destination address is ham, military, commercial, educational, etc.. The second octet might indicated which state the destination machine is in. The third, depending on how the ham TCP/IP addressing committee decides to run things, might determine a network node output or a county or city. The last octet determines which individual machine that the message goes to. So, given that the network extends across the country, it should be possible to address a message from any TCP computer to any other. The addressing system also allows for more than one user at each machine. Thus I can be ka2dew@k1tr.ampr.org which is different than k1tr@k1tr.ampr.org.

The process in the TCP program which sends messages from the host and waits for acknowledgment from the destination station is more sophisticated than TheNET. With TheNET up to four messages are sent out of the originating node and then when acknowledgments come back for those messages new messages can be sent out again. Four messages may be outstanding at a given time. If an acknowledgment for a message doesn't come back for 5 minutes the originating node will regenerate the message. With TCP/IP what is a 5 minute timer in TheNET is automatically adjusted depending on previous performance of the link. This is called 'backoff'. TCP/IP is loaded with this kind of intelligent networking features.

TCP/IP using the KA9Q software package is very easy to modify. NET.EXE which is the original KA9Q package has been added to by dozens of other amateurs. NOS, Network Operating System, is updated and customized by many hams for many purposes and is entirely public domain. This is in contrast to TheNET which is only modifiable with great difficulty.

TCP/IP is a mature protocol system due to the vast number of people working with it. TCP software is available for most computers. It can be run over a huge number of different kinds of data links. It is extremely powerful. It is in use on many, if not most, commercial workstation systems in the world. Sun Microsystems, Apollo, HP, Xerox, DEC, Apple, Next, Wang, and most other computer companies either use TCP/IP exclusively or at least offer support for it as a standard feature of their computers.

For more good info on TCP/IP I recommend *Internetworking with TCP/IP* by Douglas Comer and published by Prentice Hall.

### *Why Aren't All Packeteers Using It?*

The first reason is that it won't run in a simple TNC. It is quite possible to have lots of fun and take advantage of many of packet radio's capabilities without TCP/IP. Truly there are things that we can do *with* TCP/IP that we can't do without TCP/IP. Many things we're already doing without TCP/IP could be done better *with* TCP/IP. That won't convince all packeteers to use TCP/IP from their homes. At this time TCP/IP still requires a computer somewhat more powerful than a TNC.

TCP/IP hosts that send into the network should access it via dedicated point to point links or via an HTS free LAN. TCP/IP transmission into 2 meter user ports is bad. They must also use parameters that are no more aggressive than those recommended for the TheNET nodes. The reason is that TCP stations are themselves networking hosts. They can source high volumes of data to the network at will and can be remotely controlled from other TCP stations to do this. In order to function across the network TCP switches, by there very nature, must be independent of the controls that TheNET places on network traffic through the TheNET parameter controls. Specifically, retry rate and window size are controlled for NEDA AX.25 users but couldn't be controlled for TCP/IP users. In addition the NEDA network user access policy is designed to permit *equal* access to network services by all stations. Because of the way packet works in regards to hidden transmitter syndrome, stations transmitting into a user port cause much more loading than stations receiving *from* a user port so having TCP/IP stations on shared 2m channels would not be fair. This makes a TCP/IP station cost about $800 (in radios and TNCs) in addition to the cost of a PC, instead of about $400 and the cost of a dumb terminal for normal packet.

### *NOS - Network Operating System*

NOS is a software package created in the past several years which implements TCP/IP in an amateur radio environment. It runs on a PC, Amiga, Macintosh and several different business computers. An intriguing capability of NOS is the ability for a TNC-only user to connect into the NOS node and then use it's TCP features. This opens up the possibility of building a network using NOS nodes instead of TheNET. Users who do not operate TCP/IP hosts would be able to connect into a local node and then use TELNET to navigate the network and get to other hams and servers. This gives us the ability to use the better features of TCP/IP for networking while still leaving the network open to AX.25 users.

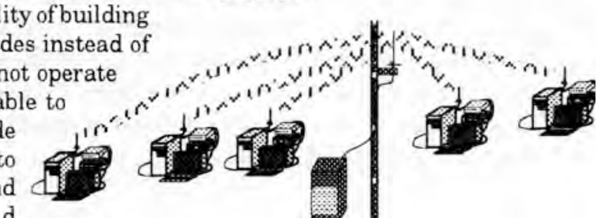
### *Why not NOS everywhere?*

When building a network of many sites one of the driving factors in new amateur participation is that the newcomers can have fun with the existing equipment. One thing that amateur radio operators find fun about packet is hacking around the network and seeing how the radios are used. In a NOS network this is not necessarily possible. It takes a deliberate effort of the sysop to make this information available. That plus the fact that NOS is complex might dissuade a newcomer from adding to the network. That alone is a major contributing factor to why TheNET is so much more common in amateur radio network building than TCP/IP.

Another feature of NOS which may be causing a problem with amateur radio network builders is that when a user establishes a TELNET session with a remote NOS host the user is asked for his callsign. Each time the user steps to a new node (while navigating or playing) the user is asked for his callsign. There is no sure way for a concerned observer to trace back through the network to find the original callsign or origin NOS site for an observed user. Since amateur radio is a self-policed body it behooves us to provide a mechanism for such traceability. This is not a built in feature of any TCP/IP software at this time.

Other things that need to be improved with NOS are it's documentation, ease of set-up, and the availability of off-the-shelf single board solid state machines that perform the duties of a NOS network node. A program for TNCs which does TELNET wouldn't hurt either. There is a light down the tunnel (hope it isn't an oncoming train hi!) There are now two solid state NOS hardware platforms. The Kantronics Data Engine and the Gracilus PacketTen. One or both of these may alleviate these objections. Then look out TheNET!

**—KA2DEW**



# CROWD: Conference Nodes

One of the TheNET software versions available allows for a roundtable or conversational type of QSO between multiple stations. NEDA has made the use of node sites equipped with this software easy to recognize by giving these special feature ports a distinctive name. NEDA converse ports are all called CROWD regardless of their callsigns or location in the network. This makes it extremely easy for keyboard users to find and access a particular CROWD port for a fun time keyboarding with other packeteers. All a user has to do is connect to any node that hosts a CROWD port and type C CROWD to be automatically connected to the local **CROWD** port.

The usage and commands for these ports is extremely easy to learn. The main commands are all proceeded by a forward slash line "/" and are entered on a line by themselves in order to be accepted as a valid command.

The commands and what they do are as follows:

***/? Help***

This is the HELP command. It will give you a page full of commands and what they do. You may also enter a /H to get the same thing.

***/W Who***

This stands for WHO and will give you a listing of who is logged onto the CROWD port and what channels they are on. There are 255 channels on each crowd and each channel can theoretically have 255 users.

***/M Message***

This is the command to send a *PRIVATE* message to another station who is logged onto the same channel that you are. Usage syntax is:

```
/M WA2TVE Hi Howie, nice to see you here!
```

Where WA2TVE is the station to whom private text is sent and the line of text that follows will be the private message. The sent text will only be seen by WA2TVE.

***/C Channel***

In order to change channels you would use this command.

When 1st logging onto a CROWD you will be on channel zero. To change to another channel you send the slash C followed by a space and a number between 1 and 255. See the helpful hints for suggestions on gentlemen's agreements regarding channel usage.

***/I Invite***

If you wanted to invite someone to join you on whatever channel you are on use this command. Syntax is slash I followed by a space and the callsign of the station you want to invite. That station will get a one line blurb asking him to join you on whatever channel you sent the invite command from.

**/B Bye**

Or if you prefer a /Q for QUIT. This logs you off the CROWD port and disconnects your stream into the port.

One last operational point on CROWD usage is that you must send at least a <return> command after connecting to CROWD for it to actually enter you into its list of users and show you as logged on. For convenience sake the best thing to send after getting the initial connection back from the port you are accessing is the /W command. This will not only get you initialized on the CROWD but also show you just who is logged on and already there.

**Helpful Hints about CROWD nodes**

The following usage conventions are suggested as good operating practices when using a CROWD node. The conventions are designed to keep loading to a minimum while allowing the most effective use of any CROWD port.

1> When you have more than *three* users on a CROWD, move the group to any channel except channel zero. This makes it possible for a new user to log in and see the /H or /W command without being bombarded with traffic from your existing conversation.

2> At some times during your operation on CROWD you might find that other people's response to your comments are delayed so long that conversation becomes unfriendly. When this happens *do not* send lines of remarks regarding this phenomena. At least, don't send it in an open message. Take advantage of the /m feature when the CROWD seems to be terribly slow. Otherwise it'll just get slower.

3> Keep an eye out for *new users*, who are easily confused and haven't learned how to use the CROWD node yet. It is best if someone instructs a new user by chatting with them on channel 0 before dragging them up to an active QSO channel. This concept alone is reason enough to leave channel 0 clear most of the time when the crowd is in use.

4> <*PRIVATE*> messages can *still* be seen by other people, particularly from the various USER ports it finally comes out on. Be discrete and use operating and conversation habits that act like *everybody is listening*, as if you were operating any other amateur mode.

5> The node ops of the network nodes are responsible for making things run right. Don't be afraid to send packet mail to the sysops of a particular site to report a problem or irregularity. Use the INFO command at the user port of the site hosting the CROWD to ascertain the site operator. Node ops would rather get several complaints about system problems than not know that anything is wrong.

9> Extreme "upper" channels have been suggested as places to hang out when waiting for a sched (IE: above channel 200). If it seems that stations are hiding up in the 200 range it is probably because they aren't able to pay as much attention to the conversation as they would have to if they stayed on the active channel while waiting for their sched (specific stations).

10> If you log into a CROWD, stay there for at least several minutes! Sometimes when the network is loaded, it takes several minutes for traffic to travel through the network and then another couple of moments for the station monitoring the CROWD to get back to you.

11> If a CROWD is inactive, feel free to leave a stream from your TNC on channel 0 waiting for someone else to log in while you tinker in the network on another stream from your TNC. This way someone else might check in and find you on the CROWD. This is a great way to meet people and to promote keyboard to keyboard conversation. By the way the current 'no activity time-out' on CROWD nodes in the network should be 2 hours. The good news is that you can leave your terminal on in your shack and work on something else. The bad news is that if you walk away people will be saying "Fred? You there? Fred? helllloooo? Fred??" until your two hours is up!!!

12> Don't get stuck in a rut. There are CROWD nodes at many of the network sites (and some outside of our network). Try them all, what the heck, try them all at the same time! (and hope that you are well practiced at keeping your TNC streams straight!) See NEDA user port maps for CROWD sites.

13> If an emergency net starts up on the CROWD port that you are using the net control station may ask stations to either log off or move to certain channels for various purposes. Please comply as all of the capacity of the network may be required for emergency traffic. The CROWD nodes (and amateur radio) are here primarily here as an emergency resource. We just have fun with it in mean time and hope that we never have a real emergency.

**Running your own CROWD**

Once you have a multiport node set up with at least three user ports able to access it via backbone links a CROWD node may be added. If a CROWD node is added to a lesser network it only adds to the frustration of the users. The CROWD node is a TheNET chip similar to that used in network nodes. It runs in a TNC2 and hooks up to the diode matrix in your multiport node. The radio port is also active and can be used as a backbone port although neither the routing info or parameters are remote programmable so this is not recommended except in emergencies.

The CROWD software is available from most of the packet Elmers and is normally handled on PC compatible floppies in binary data form. The Elmer can tell you who can burn an EPROM for you if supply a 27256 or compatible.

Note that too many CROWDs on a network spread the users thinly to the point where they don't run into each other anymore. Only if the concentration of users is good do CROWD round tables generally start.

**What does it look like?**

Here is an example conversation on CROWD. In this conversation Doc, WB2JAB, Tadd, KA2DEW and Pete, N2IJW were already talking when KA0PGQ signed on. Tadd is recording this so it is seen from his perspective. Tadd's typing is in italics.

```
cmd:
c potsdm-5
cmd:*** CONNECTED to POTSDM-5
c crowd
POTSDM:K2CC-1} Connected to CROWD:KA2DEW-7
/w
CROWD:KA2DEW-7> POTSDM CROWD /w->who /h->help|1091
User       Circuit            Channel
WB2JAB     CANTON:WA2MZF-5    0
N2IJW      CANTON:WA2MZF-5    0
KA2DEW     POTSDM:K2CC-1      0
***
Now I'm back on.
<N2IJW>:   YUP, STILL HERE. THOUGHT ABOUT
           GOING TO GREEN N GOLD GAME, BUT
           NOT MUCH DIFFERENT THAN GOING TO
           THAT MIDNITE (12:01 PRACTICE ON
           OCT.5TH.
<WB2JAB>:  OK WELL EVEN I CAN UNDERSTAND
           THAT !
<N2IJW>:   DOC, ARE YOU DIGIPEATING THRU ME? SEEMS
           MY RIG IS XMITTING A LOT!
<WB2JAB>:  HOWS THE JOB GOING PETE ?????<<
<WB2JAB>:  NO PETE I AM TO 'CANTON' V MNY THEN
<WB2JAB>:  CROWD..???????????
<N2IJW>:   IT'S GOING WELL. FEW CHANGES SINCE I
           BEGAN, BUT ALWAYS LOOKING UP. AM SURE
           GLAD THAT I MADE THE CHANGE IN JOBS.
<N2IJW>:   OK, SOUNDS LIKE GOOD PATH.
I think that that old job sounded pretty
bad Pete. You can WALK to work now.
<N2IJW>:   YES, I WALK ALMOST EVERY DAY. EXCEPT THE
           DAYS WHEN THE WEATHER SERVICE REALLY
           BLOWS IT. THEY CALL FOR RAIN, I DRIVE,
           AND THE DAY IS SUNNY!!
<WB2JAB>:  YOU STILL GOING TO WORK DAILY PLANET AT
           THE GAMES PETE ???
<N2IJW>:   DON'T THINK SO, DOC. CAN ONLY FLY FOR
           FREE FOR THAT ONE YEAR.  ALTHOUGH I
           HAVEN'T ASKED GARY MIKEL ABOUT IT.
<WB2JAB>:  TADD HOW MANY CHANNELS IS ON CROWD ?????
<N2IJW>:   RIGHT NOW, THERE'S A DUMB SHOW ON TV...
           WILL HAVE TO CHANGE THAT!
It's got 255 channels but it can only
handle 32 users.  Pretty dumb eh?
*** KA0PGQ signed on
Howdy PGQ! Tadd here.
<N2IJW>:   HOWDY...PETE HERE.
<WB2JAB>:  HELLO KA0PGQ HOW'S BY YOU, & WHAT'S NEW
           IN THE "ZOO" ??
<WB2JAB>:  MY NAME IS "DOC" QTH IS WINTHROP, N.Y.
```

<KAOPGQ>: zoo??? the only zoo i know is the one i left in germany. Am now at Drum.
*How long were you in Germany?*
*Also how long are you going to be at Drum?*
<N2IJW>: PGQ, WHAT'S YOUR NAME? GERMANY, DID YOU LIVE THERE?
<KAOPGQ>: ok there is tod,pete,doc..... sounds like we are missing snow white
<KAOPGQ>: name here is rob
<KAOPGQ>: i will be here for a while
*Tadd, you cretin!*
<KAOPGQ>: and i am not shure how long i will be here
<WB2JAB>: HI ROB NICE MEETING YOU OM.
<KAOPGQ>: i wuz in da land for 3.5
<N2IJW>: ROB, DO YOU SPEAK GERMAN?
<KAOPGQ>: call there was/is da2xl
<KAOPGQ>: i was in giessen which is north of frasnkfert
<KAOPGQ>: sprect bissen doitch
<WB2JAB>: HAY TADD CAN LINDA COME TO THAT MEETING ??????<<
<N2IJW>: SEHR GUT! ICH SPRECH SEHR GUTES DEUTSCH, ABER NUHR NICH MIT SCHRIFT!!
<KAOPGQ>: ??????? cat got ur tung????
*Sure thing. It's an open meeting. It'll probably be > 2hrs long tho.*
<KAOPGQ>: youbetcha
*We can send out for pizza if it gets too late.*
*We've got the room until 9PM.*
<N2IJW>: ROB, MY GERMAN IS FLUENT, BUT NOT IN WRITTEN-FORM!
<WB2JAB>: BY THE WAY ROB LINDA IS 3, & WEIGHS .ABOUT 110#.
*3? 110#?? That's hard to handle.*
<KAOPGQ>: what are you all runnin? i have a KAM and a tandy laptop PC runnin pacfile
<N2IJW>: <<<<<<<<<< AND LOVES MILKBONES!!!!!!!!!!
*I'm using a Mac with a PacComm Tiny 2 and a Kenwood into a dipole on the roof.*
<WB2JAB>: WELL SHE DOES HAVE A MIND OF HET OWN !
<N2IJW>: HAVE A TELEVIDEO-950 DUMB-TERMINAL, WITH A PAC-COMM TINY HERE.
<KAOPGQ>: 110 is that kilos ore carrats
<WB2JAB>: RIGS ARE: TANDY 1000, MFJ-1278, KENWOOD TR 7730 , & A RINGO RANGER 2.
<WB2JAB>: LBS. ROB GOT SOME TO GROW YET.
*Rob, you going to make the packet meeting tomorrow?*
<KAOPGQ>: i have a amcomm radio .... very old fm mobile
<KAOPGQ>: how tall???
<N2IJW>: RIG HERE ALSO OLD...MIDLAND 13-500 XTAL-RIG.
<WB2JAB>: ABOUT 22", SHE HAS A BIT TO GROW YET, BUT NOT MUCH.
<KAOPGQ>: i must work tomorow ... and i have no transportation any way
<N2IJW>: TADD, MAYBE ROB CAN CATCH A RIDE WITH KA2QJO, KRISS...IF HE'S COMING.
*Yup. It is too late at night to arrange that though. Sigh.*
<WB2JAB>: BY THE WAY ROB LINDA IS A BLOODHOUND..
<KAOPGQ>: 22" 110 lb 3? what gives????
<KAOPGQ>: i must work tomorow i
<KAOPGQ>: rrr wooofff
<WB2JAB>: ITS STANDARD SIZE FOR A BLOODHOUND ROB..
*WOOOOOF!*
*WOOF hahahaha. Funny joke Doc*
<N2IJW>: ROB, SEE IF YOU CAN REACH KRISS (KA2QJO) ON THE 147.255+ REPEATER, IF YOU CAN MAKE IT...AND HOPEFULLY KRISS IS COMING.
<N2IJW>: TADD, SEE ROB KNOWS WOOF!
<KAOPGQ>: i like cats
<KAOPGQ>: i wont be able to do anything tomorow i have 24 hr phone watch
<KAOPGQ>: here kitty kitty kitty
<WB2JAB>: WELL LINDA HAS BEEN AT K2CC BEFORE TADD.
<N2IJW>: TADD, LINDA MIGHT BE ABLE TO ACT AS A GUARD-DOG....
*Might?*
*110# and you say MIGHT?*
*AAAAAAAAAAAAAAAAAH*
*/w*

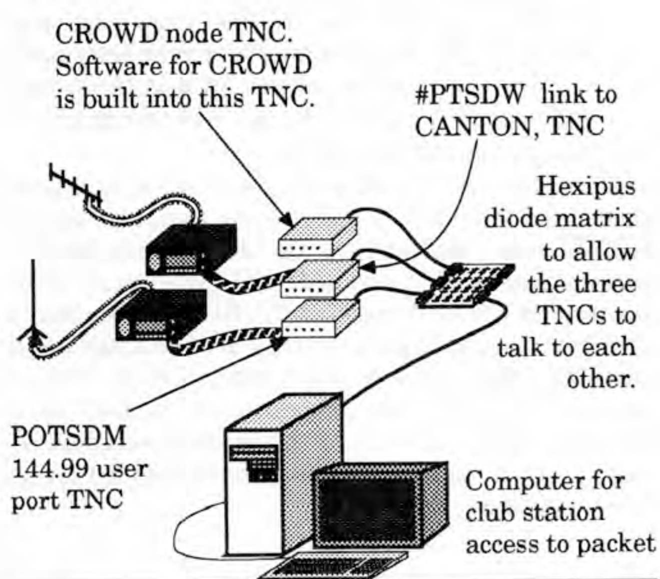
| User | Circuit | Channel |
|---|---|---|
| WB2JAB | CANTON:WA2MZF-5 | 0 |
| N2IJW | CANTON:WA2MZF-5 | 0 |
| KA2DEW | POTSDM:K2CC-1 | 0 |
| KAOPGQ | WATERT:KA2QJO-5 | 0 |

***

Well enough of that. Conversation on the CROWD that night ranged from talk about race cars, to guns, saxophones, radios, packet, dogs, food, you name it and the conversation lasted for about 3 hours. It's a lot of fun. Join the CROWD.. If nobody is on when you get on just hang around. Leave CROWD on when you are working around the shack. If you do that every evening and anybody else does too they you've started a CROWD! Here's a copy of the help file from CROWD:

| NEDA *CROWD* port | Commands are: |
|---|---|
| /? | Print help information |
| /BYE | Terminate the convers session |
| /CHANNEL n | Switch to channel n |
| /EXIT | Terminate the convers session |
| /HELP | Print help information |
| /INVITE user | Invite user to join your channel |
| /MSG user text | Send a private message to user |
| /QUIT | Terminate the convers session |
| /WHO | List all users and their channel numbers |
| /WRITE user text | Send a private message to user |

***



# TheNET Network Node Guide

**This document attempts to explain the theory and operation of a generic TheNET node. This document applies to TheNET X-1J.**

TheNET software by
Hans Georg Giese DF2AU & NORD><LINK
and by Dave Roberts, G8KBB,
William Beech, NJ7P with Jack Taylor, N7OO.
Documentation by G8KBB, KA2DEW and N7OO.



**TheNET and TheNET Plus**
**Portable. Compatible. Public Domain.**
**NORD><LINK**


## Table Of Contents

# Introduction

TheNET is a software package that resides in an EPROM in a TNC2 compatible TNC. An entire network is built up out of TNCs running TheNET. Keyboard operators and servers use the network to communicate with each other. This manual is a users's manual, installers manual and network manager's manual for TheNET. See the section "NEDA: What's a Node" for background and networking information.

# TheNET Versions

The original TheNET that was common in the US was version 1.0 by DF2AU. At that time source software in well documented, but German language, C code was made available. Since then, DF2AU and friends produced versions 1.01, 1.1 and 1.16. 1.0 is extinct and 1.1 is almost extinct by this time. 1.16 is in common use and supports features not found in any other TheNET versions. 1.01 was the last DF2AU for TNCs (there's a version for PCs) source code that I have seen.

NJ7P used the TheNET 1.01 source code and created TheNET Plus verions 2.00 through 2.11. NJ7P source code is not available as of this writing.

G8KBB used the TheNET 1.01 source code and created TheNET X1, versions X1-A through X-1J. G8KBB will send source code on demand. It's still mostly in German though.

At this time G8KBB is still creating software. NJ7P and DF2AU are both known to be working on packet hardware/software projects at this time.

***NJ7P TheNET Plus versions***

**v2.00** was the prototype test node. The maximum number of calls listed in the Heard list was 10 over a 10 minute period.

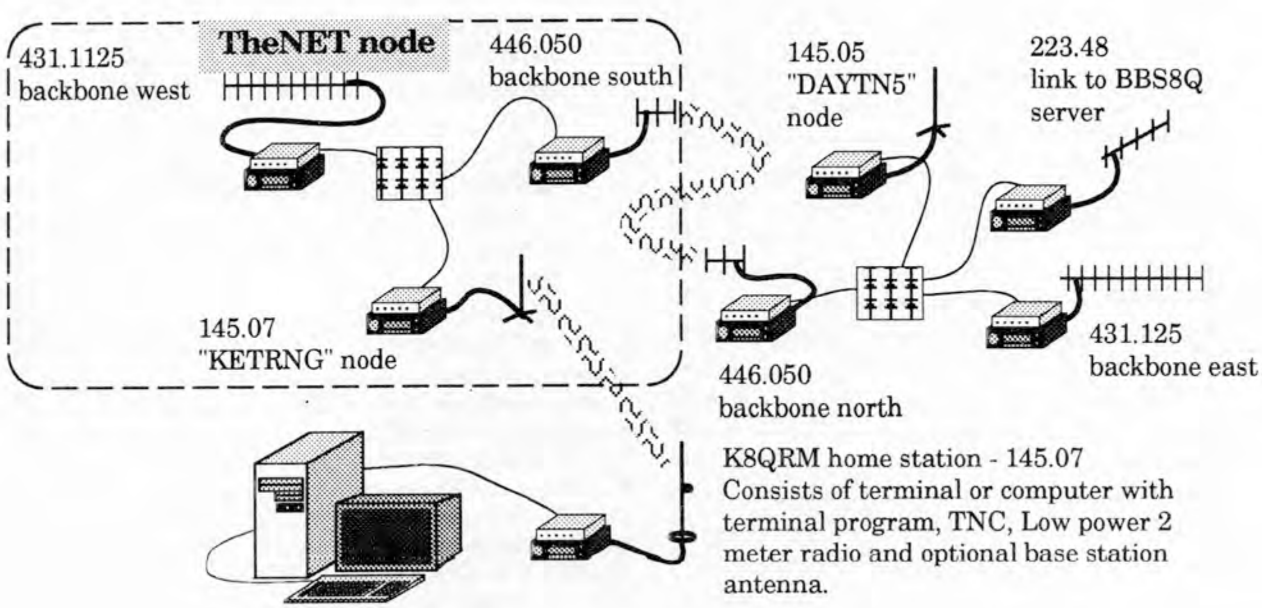
**v2.01** was the first "official release" of TheNET Plus. The Heard list was changed to a maximum of 20 calls listed over a 15 minute period. A parameter was added which allowed the NodeOp to set the maximum number in the Heard list to a value less than 20, if desired.

**v2.02** was not released.

**v2.03** was identical to v2.01 with the exception of a bug fix associated with Parameter 24. In v2.01 if callsign verification was turned off, it also disabled the N * function. v2.03 corrects this situation.

**v2.04** corrected a problem caused by the fix in version 2.03 and changed Parameters 28, 29, and 30 to agree with the SETPLUS utility and the current documentation.

*<< This information copied from a document by N7OO >>*



**v2.05** added several new NodeOp convenience features. One was to have the STA LED light when someone connected to the node. It was felt this would assist the NodeOp in servicing his equipment while at the site. Another new feature was to add three sysop KEY commands, MARK, SPACE and DIDDLE which keys the transmitter and turns on appropriate alignment tones. Also added was an ON - OFF remote control capability. One of the NODE command responses was changed to **Host busy**, instead of **Host table full** when a user attempted to connect to the host (a non-allowable function when the host port is active). The **Not Found** response to an unknown node now indicates **Not Found: <node alias>** to assist those running multiple streams in identifying which stream is which.

**v2.06** corrects a long standing routing display anomaly, which takes care of the last known problem.

**v2.07** was not released.

**v2.08** adds connect command disable, #node propagate disable option, adds broadcast via port options and reorders parm list. Only 16 parameters are read/write.

**v2.08B** makes ROUTES command response show both mnemonic and callsign.

**v2.09** is a debug, beta test only version

**v2.10st** adds a disconnect back to last node feature; nodes broadcast on power up; shorter PARM command (P 6 200 instead of P * * * * * 200); transmission 1s and 0s alternate instead of just flags during key-up delay tx; adds parameters for telemetry option; increases number of read/write parms to 23; USER response is modified to add level 3 usage display, digipeating in and out of the node strictly prohibited.

**v2.10dx** doubled the available circuit table space from previous versions of TheNET and TheNET Plus.

**v2.11** fixes a few bugs in 2.10. Also released with st and dx versions

## X code versions by G8KBB et al.

X-1 was the first release of this code.

X-1A added the escape-N command and the change to the connect, nodes and reset commands. The timers were also added to the stats command.

X-1B removed all the escape commands apart from C,D and P. It also added the MODE command and extended the + and - command qualifiers to all commands.

X-1C added TALK, MANAGER and AUDIT. The SYSOP command was enhanced and the INFO command was altered to limit the length of a message (a bug in the original version of TheNet). The help screen was changed to display commands in a combination of upper and lower case.

X-1D extended the auditing and statistics to cover auditing everything but level 3, and statistics of the CPU, Level 1, Level 2 and timers.

X-1E added beacon timer control, the connect redirector, the nodes dump facility, level 3 & 4 statistics and the LINKS and CALIBRATE commands.

X-1F added the CLOSEDOWN, ACL, CTEXT, DXCLUSTER, HELP and BTEXT commands. Another parameter was added to the MODE command to control textual messages. The mod suggested by DF2AU to correct the DCD latchup was included. Additional statistics were added covering CRC errors, receiver overrun, transmitter underrun and framing errors.

X-1G added mainly the IP router, with the following commands to control it - IPROUTE, ARP, IPSTATS, IPADDRESS, IPBROADCAST. In addition, the ALIAS, BBSALIAS, HOSTALIAS and DXCALIAS commands crept in, as did QUIT as an alternative to BYE. The help messages extended to enable nodes in the routes list to appear as alias:callsign, and an extra byte on the MODE command allowed '#' nodes to be selectively NOT broadcast. The order of HELP and HOST commands changed so that 'h' on its own gave help not host. The code was optimized with some time critical parts being recoded in assembler and a peephole optimizer being used for additional improvements.

X-1H fixed 3 bugs in X-1G.

X-1J added the deviation meter support with the Meter command and Mheard changes. In addition, parameters were added to the MODE command for slime trail control, control of digipeating and reconnection to node. The command syntax of Info, Btext and Ctext was changed to support multiple lines and the Info message space was doubled to 160 bytes. Nodes broadcasts now occur 60 seconds after power up and the ARP Digi bug fix was included. The level 4 minimum retries was dropped to 1 and the PARM, MODE, IPSTATS, METER and MTU command syntax was extended to support 'offset & value' type operation. An MTU command was added to allow IP MTU limits to be changed under software control. The node alias case sensitivity bit and TALK 8 bit data bits were added. Last but not least, X-1H and before had a problem where sometimes the node would crash when reset or it would scramble it's memory when reset. This is now fixed.

Future As of this writing NJ7P and G8KBB are working on a combined effort.

# Why TheNET?

TheNET is a good choice for network building. It has several key qualities which make it a very good choice for an amateur radio packet network building block.

- It allows for sites with multiple radios and backbone links;
- It supports DxCluster, TCP/IP, keyboard to keyboard, BBS forwarding, etc.;
- TheNET is used in TNC2s which can handle relatively harsh environments;
- It's software is mostly bug free;
- New versions with new features are frequently available;
- There are several groups working on the software so future viability is probable;
- Network architecture is verifiable remotely by all users;
- Other compatible software that is not TNC2 based exists;
- Operating parameters, routing, and other sysop configurations are visible to all users;
- Backbone routes are easily protected against miscreant abuse while still being monitorable by any ham with a receiver and TNC (and appropriate modem)
- Network sites using TheNET are easily upgradable in single radio/port increments;
- It's fun for a non-invested user;
- Some level of automatic route determination and information is created automatically by TheNET and is available for both the network to use and for the network user to inspect.
- It's cheap (less than $140 per port for digital hardware);
- Users of a TheNET network are quickly exposed to the tools that are available to debug and analyze the network.

It is important in ham radio packet network implementation that good design practice is followed. Elsewhere in this *Annual* is information which is required to build a successful TheNET based network.

### Other Packages

There are several other network software packages available. Each has advantages and disadvantages. The good news is each has its champions and therefore will compete with each other even as one package pulls ahead in popularity for a while. Items to look for include:

***Easy learning curve***

The common user can go out and learn about packet networking while using the networking software for practical purposes. The network software should be obvious enough that a beginner can play, while not so simple as to restrict an outsider from learning how the network operates. A newcomer shouldn't be required to study nuclear physics, brain surgery and VLSI chip design in order to use the network, but neither should the user be given a set of instructions on how to use particular aspects of the network with no hope of learning anything more without becoming an insider first.

***Built by hams***

The basic amateur radio operator can be considered to be a person who has a minor favoritism towards kit assembly (antenna construction, coax cable assembly, plug this into there), technogadgetry, communications, community service, and emergency preparedness. Any construction project that is going to be successful with a wide cross section of hams is going to have to have that common denominator. If the project requires a ham to be a programmer, a PC computer technician, a radio technician, and wealthy to boot, it's not going to grow.

# Using a TheNET Network, brief

This is a short description of what a TheNET network looks like to the casual user. The following chapters are more in depth but assume the reader understands some basics are elaborated on this page. This page assumes the reader understands what packet is, what it means to connect to something, and what a command processor is.

The user connects from his TNC to a TheNET node. Choosing the best node for access is discussed later. The connect is done using the connect command from the user TNC. The node may be addressed using either the node's callsign and specific ssid or using the node's mnemonic and any of the 16 SSIDs. Let's assume our user is N2MGI and the local node is POTSDM:K2CC-1. The following are all valid connect commands:

c POTSDM
c K2CC-1
c POTSDM-4

The POTSDM node will answer any of these. All are perfectly reasonable ways to connect to the node. The reason the node allows this versatility is if the user can do multiple streams he can connect to the same node up to 17 times, or however many streams the TNC allows if less than 17.

In this example the node would answer and a *** **Connected to** message would show on N2MGI's screen. A monitoring station would observe:

**N2MGI>POTSDM (c)**
**POTSDM>N2MGI (ua)**

or

**N2MGI>K2CC-1 (c)**
**K2CC-1>N2MGI (ua)**

or

**N2MGI>POTSDM-4 (c)**
**POTSDM-4>N2MGI (ua)**

The node will assume any of the 17 identities for the purpose of maintaining the connection. N2MGI could, on three different streams, connect to all three of these identifiers.

### *Network Use*

After the user has gained access to the node he can request a list of destination network sites or he can issue the Connect command to select one. Then he can use the Connect command from the destination node site to connect out of the network. The following is an example procedure where N2MGI might connect to KA2DEW using the network. From the commanc prompt N2MGI types:

**cmd:**C POTSDM

Shortly his TNC answers:

***** Connected to POTSDM**

N2MGI is now connected to the TheNET nod( POTSDM. All text typed now will be processed by th( node as commands. There are several commands avail able at this time. For MGI's purposes of connecting t( KA2DEW from CANTON node he then types:

C CANTON

This tells the POTSDM node to pass MGI off to th( CANTON node. POTSDM returns:

**POTSDM:K2CC-1} Connected to CANTON:WB2MZF-2**

Once again MGI can enter any of the TheNET use commands. To get to KA2DEW he types:

C KA2DEW

POTSDM node receives the "C KA2DEW' from MGI and passes it off to CANTON. CANTON makes the connection to KA2DEW and then sends it's response back through the network to POTSDM which then sends:

**CANTON:WB2MZF-2} Connected to KA2DEW**

KA2DEW's station, which has gotten a connection from CANTON under the callsign of N2MGI-15 responds tc CANTON with a connect text. That text is sent through the network to POTSDM which then sends it to N2MGI. So:

**Tadd's station. Use KA2DEW-4 for my PMS**
**or beep several times to get my attention!!**

Now any traffic N2MGI or KA2DEW type will be routed back through the network to the other station. From either station's point of view the network connection is direct.

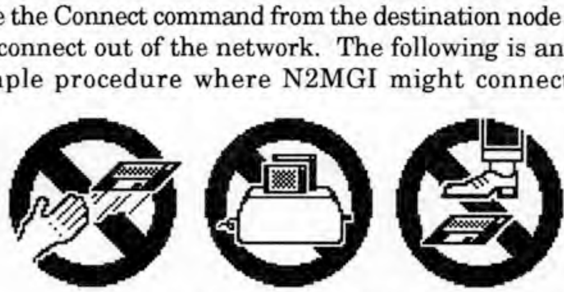
When the conversation is complete, either station may disconnect. Let us assume KA2DEW does the disconnect by typing ^C and then D. N2MGI, who initiated the connect, will be reconnected to CANTON. N2MGI will receive:

**CANTON:WB2MZF-2} Welcome Back**

This means the next thing that N2MGI sends will be interpreted by the CANTON node. N2MGI could now go and connect to something else. If MGI issues the B command, the CANTON node will disconnect and MGI will receive:

**POTSDM:K2CC} Welcome Back**

At any time, including before KA2DEW had disconnected, MGI could do a ^C and D to break the circuit. There is no advantage gained by asking the network to disconnect itself.



ILL-ADVISED DISKETTE CARE

# Operating a TheNET Node

**TheNET Commands**

A TheNET TNC accepts commands from any connected station. One of these is the CONNECT command, used to contact another station or to move through the network. Other commands include requests for information, more complex communication requests (different than the CONNECT command), and programming operations that would be used to configure the node. The coverage of the commands and configurables fall under several sections, notably *Non-Sysop Commands*, *Sysop Commands*, and *Modes and Parameters*. Sysop Commands require the completion of a sysop password exchange. The sysop password is set when the TheNET EPROM is burned (see *Constructing a TheNET Node*). This is a list of the commands:

| ***Command*** | ***Description*** | ***Page # Non-Sysop*** | ***Page # Sysop*** |
|---|---|---|---|
| ACL | Access Control List | 44 | 53 |
| ALIAS | Set node name | 44 | 54 |
| ARP | Add to, erase, or read IP addr trans tbl | 44 | 54 |
| AUDIT | Configure or check Audit mode | 46 | 54 |
| BBS | Set or read local bbs setting | 45 | 55 |
| BBSALIAS | Set or read over-the-air alias for BBS | 45 | 55 |
| BTEXT | Set or read Beacon text | 46 | 55 |
| BYE | Tell node: disconnect from it's end | 46 | - |
| CALIBRATE | Turn node Tx on, send test signal | - | 55 |
| CLOSEDOWN | Shutdown node until power removed | - | 56 |
| CONNECT | Connect to other node or local station | 46 | 56 |
| CQ | Send text over radio, wait for connect | 46 | |
| CTEXT | Set or read connect text | 46 | 56 |
| DXCALIAS | Set or read over-the-air alias for DXC... | 46 | 55 |
| DXCLUSTER | Set or read local dxcluster setting | 46 | 56 |
| HELP | Display help message | 46 | - |
| HOST | Set or display local host setting | 46 | 56 |
| HOSTALIAS | Set or read over-the-air alias for HOST | 46 | 55 |
| INFO | Set or read info text | 46 | 56 |
| IPADDRESS | Set or read node's IP address | 47 | 57 |
| IPBROADCAST | Set or read IP broadcast address | 47 | 57 |
| IPROUTE | Set or read IP route table | 47 | 57 |
| IPSTATS | Set and clear IP MIB stats + 1st config | 47 | 58 |
| LINKS | Display L2 "links" currently set-up | 48 | 58 |
| MANAGER | Request 'manager' privilages | - | 58 |
| MHEARD | Display last heard, or set heard length | 48 | 59 |
| MODE | Set or read 2nd of 3 sets of node configs | 48 | 59 |
| MTU | Change packet lengths for use in TCP/IP | - | 59 |
| NODES | Access list of known network sites | 48 | 60 |
| PARMS | Set or read 3rd of 3 sets of node configs | 49 | 61 |
| QUIT | Tell node: disconnect from it's end | 49 | - |
| RESET | Selectable code or warm start | - | 61 |
| ROUTES | Access list of network neighbors | 49 | 61 |
| STATS | Read operating statistics | 50 | - |
| SYSOP | Request "sysop" privileges | 52 | 62 |
| TALK | Enter round table conference | 52 | - |
| UI | Send immediate UI frame | 52 | - |
| Disabling of commands | | - | 62 |

## TheNET Configurations

A TheNET node may be configured to customize operation, to make the node friendly to local and distant users, and to provide network management. This table lists the configurable by description and by the name it is given in the PATCH.EXE program, the command needed to change the item and then the page on which the configurable is described.

| *English Description or PATCH description* | *PATCH.EXE name* | *sysop command* | *page #* |
|---|---|---|---|
| Accept BBS, DXC and HOST aliases enable/disable | Extra aliases | MODE /14 | 65 |
| Access control (select callsigns to allow/restrict) | not available in PATCH.EXE | ACL | 53 |
| Acknowledge delay, AX.25 | Ack (T2) time | PARM /21 | 70 |
| Acknowledge delay, transport | Transport acknowledge delay | PARM /11 | 69 |
| All commands show in sysop mode enable/disable | Show all available commands to sysop | MODE /12 | 65 |
| Audit (node operations reports) | not available in PATCH.EXE | AUDIT | 54 |
| AX.25/L2 Frame acknowledge timeout (FRACK) | FRACK T1 time | PARM /18 | 70 |
| AX.25/L2 Link check timer (CHECK) | Active check (T3) time | PARM /22 | 71 |
| AX.25/L2 Response delay (RESPTIME) | Ack (T2) time | PARM /21 | 70 |
| AX.25/L2 Retries | AX.25 retries | PARM /20 | 70 |
| AX.25/L2 Windowsize (MAXFRAME) | AX.25 window size | PARM /19 | 70 |
| BBS callsign and alias | not available in PATCH.EXE | BBSALIAS | 55 |
| Beacon control enable/disable or only when in use | Beacon mode control | PARM /25 | 71 |
| Beacon Interval | Beacon period | MODE /10 | 64 |
| Beacon text message | not available in PATCH.EXE | BTEXT | 55 |
| Broadcasting # nodes on radio enable/disable | Selective nodes broadcast on ports | MODE /4 | 63 |
| Broadcasting # nodes on RS-232 enable/disable | Selective nodes broadcast on ports | MODE /4 | 63 |
| Broadcast interval, radio/obsolescence counter | Nodes broadcast interval | PARM /7 | 68 |
| Broadcast interval, RS-232 | Crosslink port node b'cast interval | MODE /8 | 64 |
| C command redirection | Connect redirector | MODE /11 | 64 |
| Callsign validation enable/disable | Callsign Validation | PARM /24 | 71 |
| CQ broadcast enable/disable | CQ broadcasts | PARM /26 | 71 |
| Crosslink protocol selection | RS-232 crosslink protocol | MODE /5 | 63 |
| Connect text enable/disable | User message control flags | MODE /12 | 65 |
| Connect text message | not available in PATCH.EXE | CTEXT | 56 |
| CWID repeat period | CWID repeat period | MODE /2 | 63 |
| CWID speed | CWID keyer speed | MODE /3 | 63 |
| Digipeat from node enable/disable | Digipeat up/downlink control | MODE /17 | 66 |
| Digipeat into node enable/disable | Digipeat up/downlink control | MODE /17 | 66 |
| Digipeat thru node enable/disable | Digipeat | PARM /23 | 71 |
| DxCluster callsign and alias | not available in PATCH.EXE | BBSALIAS | 55 |
| Full vs Half duplex on radio | Full Duplex | MODE /7 | 64 |
| Hide slime trails enable/disable | Control of slime trails | MODE /16 | 66 |
| HOST redirection and alias | not available in PATCH.EXE | BBSALIAS | 55 |
| Inactivity timeout | No-activity time-out (seconds) | PARM /15 | 70 |
| INFO text | Information message | INFO | 56 |
| IP Datagram mode | AX.25 ports mode control (DG/VC) | IPSTATS /1 | 58 |
| IP Default time to live | The default Time To Live (TTL) | IPSTATS /3 | 56 |
| IP routing enable/disable | Enable or Disable the IP router | IPSTATS /2 | 56 |
| IP Virtual Circuit mode | AX.25 ports mode control (DG/VC) | IPSTATS /1 | 58 |
| Maximum Transmission Unit (MTU) | Set MTU parameters menu | MTU | 59 |
| Min quality to be added to nodes list automatically | Minimum auto update quality | PARM /2 | 66 |
| Node broadcast interval (obsolescence interval) | Nodes broadcast interval | PARM /7 | 68 |
| Node broadcast on RS-232 | Crosslink port node b'cast interval | MODE /8 | 64 |
| Node broadcast on RS-232 enable/disable | Selective nodes broadcast on ports | MODE /4 | 63 |
| Node broadcast on radio enable/disable | Selective nodes broadcast on ports | MODE /4 | 63 |
| Node list length (maximum number) | Size of destination node table | PARM /1 | 66 |
| Obsolescence init value | Initial obsolescence count | PARM /5 | 67 |
| Obsolescence minimum to broadcast | Minimum obsolescence count for broadcast | PARM /6 | 68 |
| P-Persistance | Persistence | PARM /16 | 70 |
| "Please wait, trying xxxx" message enable/disable | User message control flags | MODE /12 | 65 |
| Radio default quality | HDLC default quality | PARM /3 | 67 |
| Reconnect to switch on disconnect enable/disable | Auto reconnect to node | MODE /15 | 65 |
| Routes show alias:callsign vs callsign only | User message control flags | MODE /12 | 65 |
| RS-232 default quality | RS-232 (crosslink) default quality | PARM /4 | 67 |
| Slime trail feature enable/disable | Control of slime trails | MODE /16 | 66 |
| Slot Time | Slot time | PARM /17 | 70 |
| Time to live Initializer | Initial time to live (hops) | PARM /8 | 68 |
| Transport layer (L4) # of locally buffered packets | Transport overfill limit (frames) | PARM /14 | 69 |
| Transport layer (L4) # of Retries | Transport retry counter | PARM /10 | 69 |
| Transport layer (L4) Acknowledge delay | Transport acknowledge delay | PARM /11 | 69 |
| Transport layer (L4) Frame Acknowledge timeout | Transport FRACK timeout (seconds) | PARM /9 | 68 |
| Transport layer (L4) window size | Transport window size | PARM /13 | 69 |
| TxDelay | Transmit keyup delay | MODE /6 | 64 |
| Windowsize (L4, network, transport) | Transport window size | PARM /13 | 69 |

# Non-Sysop Command List

This chapter describes commands and operations stations might do if they did not have system operator capability (or responsibility).

Once a station is connected into a node, everything the user sends to the node will be processed as a command until the user disconnects or until a command is entered which connects the user through the node to something else.

Some of the commands may be disabled by a node op. When the command is typed the response will come back as "Sorry, 'ACL' is not enabled". Before jumping to conclusions about the node or the node-op's ancestors, consider why a command is disabled and consider what convenience might be gained (or what information withheld) by a certain command being made unavailable. Be sure that you are working with a version of software that is compatible with your expectations for available commands. (Use the USERS command to verify software version). These are the commands available to the casual operator. ACL, ALIAS, ARP, AUDIT, BBS, BBSALIAS, BTEXT, BYE, CONNECT, CQ, CTEXT, DXCALIAS, DXCLUSTER, HOST, HOSTALIAS, INFO, IPADDRESS, IPBROADCAST, IPROUTE, IPSTATS, LINKS, MANAGER, MHEARD, MODE, PARMS, QUIT, RESET, STATS, SYSOP, TALK, UI

## ACL

Access Control List: This function is used to limit access of a node to a selected list of callsigns, or to restrict access for a particular callsign or list of callsigns. Access is controlled for all callsigns by the default value, then separately assigned for specific callsigns listed below the default. Access applies to both over the radio and between node TNCs at a single site. The access granted the default or a particular callsign is specified by a decimal number that is the sum of a series of binary bits. The binary bits have values of:

| | |
|---|---|
| bit 0 = decimal 1 | Bar on-channel L2 connect to the node |
| bit 1 = decimal 2 | Bar on-channel L2 connect from the node |
| bit 2 = decimal 4 | Ignore nodes broadcast from node with this callsign |
| bit 3 = decimal 8 | Bar gatewaying at L3 from node with this callsign |
| bit 4 = decimal 16 | Bar incoming L4 connects from node with this callsign |
| bit 5 = decimal 32 | Bar outgoing L4 connects to node with this callsign |
| bit 6 = decimal 64 | Ignore SSID of this callsign when checking bits. |
| bit 7 = decimal 128 | unused bit. |

The decimal number is the sum of the bits. So, value = 67 would be bit 0 + bit 1 + bit 6 which would mean "don't allow on-channel access by this callsign, regardless of ssid". Here are example ACLs taken from a user port, ALBANY:

```
ALBANY:WA2WNI-1} ACL: default 0
K2QRM    127
```

In this case the default is to allow all forms of access for all callsigns. The specific callsign, K2QRM, has been disallowed for all access. Note any station may verify this is true. There need not be any suspicions of being 'locked out' because any station may verify this is or isn't true. Thus a sysop will be more circumspect before specifying access. Note some nodes exist for specific access (like emergency or backup) and access is restricted so the node isn't used during 'normal' operation. In that case the INFO text should be pretty clear on the matter and at any rate there is no need for an individual to be specifically excluded in this list. The normal method of access control in this case would be for individuals to be specifically included and the default would be set to exclude everybody else. If you find the ACL function is not enabled, so you can't verify this list, a friendly note to the sysop asking why might be in order.

Here is an example taken from a backbone port, #KNDS. In the case of a backbone port all stations are disallowed except for the network neighbors. In the case of #KNDS there are four neighbors. Here is the Routes list taken from #KNDS:

```
#KNDN:WA2WNI-13: Routes:
  1  KNDRHK:WA2WNI-1      203 23
> 1  CROWD:KA2JWJ-11      203 1
> 0  #ALKND:KB2CS-14      203 14
> 1  #KNOXW:WA2WNI-12     203 32
```

This routes list indicates there are level 2 routes between this node (wni-13) and WA2WNI-1, KA2JWJ-11, KB2CS-14 and WA2WNI-14. To include the neighbors but exclude all others, this is what the ACL will look like.

```
#KNDS:WA2WNI-13} ACL: default 7
WA2WNI   64
KA2JWJ   64
KB2CS    64
```

The default in this case disables L2 access for all stations and ignores node broadcasts. Next the callsigns WA2WNI, KA2JWJ and KB2CS are enabled for all access, for all SSIDs. This is necessary in order for nodes with those three callsigns to be able to pass data. It is not possible to stop a random station from passing through a backbone by any manipulation of the ACL list on a hidden node. Only at the random station's entry or exit from the network would it be possible to limit that station using any operation of TheNET X-1J software.

## ALIAS

This command allows a sysop to change the node's alias. A non-sysop can only retrieve the current alias. Here are examples applied to ALBANY and #KNDS:

```
ALBANY:WA2WNI-1} Alias is ALBANY
#KNDS:WA2WNI-13} Alias is #KNDS
```

## ARP

The ARP table maps a destination IP address or gateway IP address to a Net/Rom callsign (TheNET routing information). It may also map the IP address to a direct L2 neighbor using either VC (virtual circuit) or DG (datagram) modes.

In the following example all of the destination stations are via Net/Rom (TheNET) routes. The STMFRD node is being used as a relay of TheNET node names so each of the stations on the Destination list need only have STMFRD on their node lists instead the node names for all of the IP stations.

STMFRD:KB2CS-1}

| Destination | P | hw-type | Callsign | Mode | Timer |
|---|---|---|---|---|---|
| 44.68.67.254 | | Net/Rom | WB2QBQ-1 | | |
| 44.69.3.42 | | Net/Rom | N2NQH-1 | | |
| 44.69.3.41 | | Net/Rom | W2RGI-1 | | |
| 44.44.2.254 | | Net/Rom | K1MEA-2 | | |
| 44.44.3.254 | | Net/Rom | K1MEA-5 | | |
| 44.52.8.126 | | Net/Rom | W1ET | | |
| 44.52.5.129 | | Net/Rom | NR1N-2 | | |
| 44.52.7.14 | | AX.25 | KA1PQE | VC | |
| 44.52.7.13 | | AX.25 | N1QYO | VC | |
| 44.52.6.1 | | Net/Rom | KA1BBG-4 | | |
| 44.69.1.128 | | Net/Rom | WB2PSI-9 | | |

**Destination** This is the destination IP address.

**P** If P is entered the node will answer an *ARP request* from a TCP/IP station for the given destination address, over ax.25. This isn't used at the STMFRD node because no ax.25 routings exist.

**hw-type** is Net/Rom or ax.25

**Callsign** is the ax.25 or Net/Rom callsign to route messages for *destination* TCP/IP address.

**Mode** indicates DG (datagram) or VC (virtual circuit). Datagram means an acknowledge is generated by the distant end of the network where virtual circuit indicates that an acknowledge is given by the next neighbor as well by the station at the distant end of the network. Mode being blank means the node uses the default in the IPSTATS table.

## AUDIT

This command is used by the sysop to set the level of audit. Auditing allows the sysop to observe some functions of the network and its interaction with stations using the network. Some of the options enable the node to send 10 minute updates. Others send messages to the sysop when events occur at the node. The audit command is set via enable bits, as in the ACL list. The bits are defined as follows:

| | |
|---|---|
| bit 0 = decimal 1 | 10 minute updates of % of time DCD and PTT. |
| bit 1 = decimal 2 | msg when L2 connect or disconnect made |
| bit 2 = decimal 4 | reserved for future use |
| bit 3 = decimal 8 | msg when L4 connect or disconnect made |
| bit 4 = decimal 16 | msg giving sysop command used. |
| bit 5 = decimal 32 | msg giving any command entered on the node |
| bit 6 = decimal 64 | 10 minute updates of CPU loops and minimum number of buffers available during the period. |
| bit 7 = decimal 128 | unused bit. |

Since auditing may only be turned on by a sysop, an since that is covered in the *sysop* section, the only r maining information a non-sysop might want is how t tell when a sysop is auditing and how to listen in as non-sysop.

If a station who is suspected to be a sysop is audi ing, two things will be evident. One is they show u on the user's list as a station uplinked or circuited int the node and not connected out. Second is if you typ audit, you'll get back a number that is non-0. The r sponse to an audit command will look like this:

ALBANY:WA2WNI-1} 63

or something similar. This does not mean the node i audited. It's just a clue. Note any station terminated a the same node you are may be 'talked' to. See the TAL command later in this section.

To listen to an auditing sysop you'll need to be abl to get on the sysop's downlink channel or to monitor on of the backbones the sysop is connected via. The result of an audit command are shown under the sysop com mands section.

## BBS

This command looks at one of the three *redirectio aliases*. If executed as BBS ? it will respond with th callsign or mnemonic of the station specified for the BB command. If executed as BBS, it will attempt to con nect to the station specified by the sysop. See *BBSALIAS* below.

## BBSALIAS, DXCALIAS, HOSTALIAS

BBSALIAS responds with the alias associated wit the BBS command. If a L2 uplink is attempted on th node's RF frequency to the mnemonic specified in th BBSALIAS command, the node will accept the comman and attempt to connect to the node specified in the BB command. In addition to the BBS/BBSALIAS pair ther also exists the DXC/DXCALIAS and HOST/HOSTALIA pairs. Each may be used in a similar manner. The in tent is to set up several aliases so users of the node, o frequency, may have simplified access to several serv ers. The node sysop may set up these three alias pair in any practical fashion. All three may point to DxClus ters, or to other nodes. C STMFRD might be a possibl command, on the frequency of the CHSTR node, hav ing CHSTR relay the command via the redirection alia to the STMFRD node. Wonderfully confusing!

Note the BBS, DXC and HOST commands will re spond and make a connection if the parameter of th particular command has been set (test by typin BBS ? ). In order for the BBSALIAS to be recognize on-channel by the node, not only does the BBSALIA and BBS have to be set, but the *mode* option *Extra aliase* (mode number 14) must be set to 1.

**BTEXT**

Views the Beacon Text. The beacon text may be sent out by the node at the beacon interval. The interval is set and viewed via the MODE command (*Beacon period*).

**BYE**

Instructs the node to disconnect. The node may give a "good-bye" message if so configured. This command is the same as the *quit* command.

**CONNECT**

This command instructs the node to connect to another station. The command is entered as **C STATION** where station may be a six character or less text string or a valid amateur callsign. First the node searches it's own database for a match between *station* and a known node name or a callsign associated with a known node. If a node name match is found the callsign associated with that node is used. If that is the case or if *station* matches a node's callsign, a network connect is attempted to the requested node.

If no match is found the node will process STATION and determine if it is a valid callsign. If not the node will send an error message to the user. If it was a valid callsign a connect attempt is made via the modem port of the node. If successful the user will be sent nodecall:nodename} Connected to STATION. If unsuccessful the user will be sent an error message.

**CQ**

The CQ command is used by a station who wants to make a random connection from a node. The node may either be the local node, or a remote node. The CQ command is sent with a parameter of up to 77 characters. The node will send a message with the calling station's callsign (-15), to CQ, with the parameter text in the info frame. Thus if user NK1M types:

cq Bill calling from Nashua

The node will send over the air:

NK1M-15>CQ: Bill calling from Nashua

The node only transmits the text once. If Bill wanted the text sent several times he'd have to type it several times. The node puts NK1M into CQ mode. That means that if it hears anybody trying to connect to NK1M-15 it will complete the patch, connecting the new station back through the network to NK1M. Additionally while NK1M is in CQ mode the USERS list will show NK1M-15 as calling CQ:

Circuit (MONROE:WB2GNR-1 NK1M) <--> CQ (NK1M-15)

If a station connects to the node NK1M is calling from and connects to NK1M-15 from the node, the node will make the patch. CQ mode times out and disconnects NK1M after "No-activity timer" runs out (usually 2 hours). See *PARM command.*

**CTEXT**

If a CTEXT is set and if it is enabled in the MODES settings a connect text is sent to any station that uplinks to the node as they connect. The CTEXT command will display the message to any connected station. See MODE command for *Connect text enable/disable* below, or see *Sysop Command list* section.

**DXCALIAS**

See BBSALIAS command.

**DXC or DXCLUSTER**

See BBSALIAS command

**HELP**

This command tells the node to send it's HELP text. This is usually a page or so of data. The HELP text may be changed at EPROM burn time. Here is the default help text:

This is a Z80 based node running TheNet + an IP router.
The basic commands are the same as the standard TheNet code,
but with many additions. Brief details of the main extras are:

| | |
|---|---|
| BBS, DXcluster, Host | - These commands connect to nominated stations |
| Links, Stats, IPstat | - These give info on the node state |
| Mode | - Like parms, this gives info on extra parameters |
| Bye, Quit | - Bye |
| MHeard | - The most recent stations heard & rx deviation |
| Talk | - Conferencing. |

The Talk command allows stations to hold a multi-way conference. When in Talk, all lines sent by a station get copied to all other stations in the conference. To find out who is in a conference use the User command.

**HOST**

See BBSALIAS command

**HOSTALIAS**

See BBSALIAS command

**INFO**

The INFO command instructs the node to send it's info text. The info text consists of up to 240 characters of data. The data is up to 80 characters of information stored in the EPROM and up to 160 characters loaded over the air by the sysop.

Normally the block of text will describe the node's purpose, location, frequency, who to contact, servers accessible etcetera. Examples of info messages are:

WB2QBQ-1:KNOX}

| | |
|---|---|
| port | 144.91 USER |
| QTH | Knox, N.Y. |
| sysop | WB2QBQ @ WA2PVV |
| phone | 518-555-1212 |
| maps | NEDA Box 563 Manchester NH 03105 |

K2TR-10:DXKNOX}

| | |
|---|---|
| Port | Dedicated DxCluster Link |
| QTH | Knox, N.Y. |
| info | WB2QBQ @ WA2PVV |

Enter C K2TR for connect to YCCC/AARA DxCluster System

N4GAA-1:MOOSBG}
Port 145.09 MHz USER access channel
QTH Mooseburg S.C.
spnsr N4GAA @ W3IWI
servers pls use 441.1 for network access!

For server info C KA2EIA then DN SERVERS.INF

## IPADDRESS

reads the node's IP address. Example:
STMFRD:KB2CS-1} My IP address : 44.69.3.48

## IPBROADCAST

reads the node's IP broadcast address.
STMFRD:KB2CS-1} Broadcast IP address : 44.255.255.255

## IPROUTE

reads the IP routing table. This is a list of destionation subnets and stations and direction (method) and address of the gateway station that will handle TCP/IP traffic to the specified destination.

STMFRD:KB2CS-1}

| Destination | Len | I/F | Gateway | Metric |
|---|---|---|---|---|
| 44.69.1.40 | 32 | Net/Rom | 44.69.3.41 | |
| 44.69.2.42 | 32 | Net/Rom | 44.69.3.41 | |
| 44.69.3.41 | 32 | Net/Rom | | |
| 44.69.3.42 | 32 | Net/Rom | | |
| 44.56.0.150 | 32 | Port 0 | 44.56.0.150 | |
| 44.52.0.16 | 30 | Port 0 | 44.56.0.150 | |
| 44.52.8.0 | 24 | Net/Rom | 44.52.8.126 | |
| 44.69.40.0 | 22 | Net/Rom | 44.69.3.41 | |
| 44.69.0.0 | 16 | Net/Rom | 44.69.1.128 | |
| 44.44.0.0 | 16 | Net/Rom | 44.44.2.254 | |

**Destination** is the TCP/IP address of the subnet or station this route is directed to.

**Len** is the number of significant bits in the address. This lets the IPers set up subnets which are whole regions of IP addresses that may be routed through a single machine (or several machines for redundancy's sake). The way this works is the number of bits from the left edge of the address is specified. Since 8 bits are used for each number, 16 bits means only the left two numbers are significant. So, in the case of 44.52.8.0, at 24 bits, the addresses 44.52.8.0 through 44.52.8.255 are all routed through 44.52.8.126. See the ARP table for what Net/Rom node is used for messages to 44.52.8.126 (answer = W1ET).

**I/F** indicates Net/Rom or *port 0* or *port 1*. *port 0* indicates ax.25 but over the radio. *port 1* would indicate that you had a TCP/IP route to a NOS station over the serial port but in practice this setup is not used.

**Gateway** is the IP address of the gateway station to get us into the subnet shown as destination. Gateway stations must show in the ARP table.

**Metric** is not used yet.

## IPSTATS

STMFRD:KB2CS-1} 1 1 9 112 12 0 14497 123 0 0 254 0 7 30 0 0 0 0 0 0

The IPstats command has the same basic syntax as the PARMS and MODE commands. When invoked without parameters, it displays the current stats. Each statistic may also be altered by sysop.

In addition to the standard IP MIB, there is an additional parameter used to set the level 2 default modes, and the first entry in the MIB is used to enable or disable the router.

The complete set of IP MIB stats are included for compatibility with other IP systems, but several are not used. Also, the stats are 16 bit counters not 32 bit counters as in NOS. Like NOS however, the stats do not reset every hour, they must be cleared by the sysop. They will however wrap around at zero.

The entries are:

1 Port default modes, 0, 1, 2 or 3.
2 Enable / Disable IP router
3 Default IP Time To Live
4 IP Received frames
5 IP Header Errors
6 IP Input Address Errors
7 IP Forwarded Datagrams
8 IP Unknown Protocols
9 IP input frames Discarded
10 IP Input frames Delivered
11 IP Output Requests
12 IP Output Discards
13 IP Output No Routes errors
14 IP Reassembly Timeout errors
15 IP Reassembly Required errors
16 IP Reassembly OKs
17 IP Reassembly Fails
18 IP Fragmentations completed
19 IP Fragmentation Failures
20 IP Fragmentation Creates

The default mode word may be set to 0, 1, 2 or 3. Each bit controls a port, with bit 0 controlling port 0 (radio port) and bit 1 controlling port 1 (RS-232 port). When set to 1, the default mode for that port when sending on a level 2 connection will be Datagram. When set to 0 it will be by Virtual Circuit. The default mode is used when no other information is given, either by the ARP table or the TOS bits in the IP header.

The enable/disable word may be set to 0 or 1. When set to 0, the operation of the router is stopped, when set to 1 the router functions.

The IP Time To Live (TTL) word is used to set the number of routers through which an IP frame may pass before it is discarded. It is similar to the node layer 3 TTL word. It may be set to any value up to 255, but values below 2 make no sense and are therefore not permitted.

The IP fragmentation reassembly timeout counter is not used as the node is just a router. It is left set to 30.

The rest are just statistics. Some are not functional.

## LINKS

This command shows the current level 2 links to the node. Displayed one per line: callsign the node used, callsign the other station (may be a neighbor node) used, link state, port number, and current retry count. [*note: i have no idea what these numbers mean*]

```
STMFRD:KB2CS-1} Links:
STMFRD WB2EFB 4 0 0
KB2CS-1 WA2WNI-12 4 1 0
KB2CS-1 KA2IPF 4 0 0
KB2CS-1 KB2CS-13 4 1 0
KB2CS-1 WA2VAM-14 4 1 0
```

## MHEARD

The heard list shows the last few stations heard. For each station the following information is displayed:

***Pkts*** # of packets since the station first appeared on the list. The count wraps around to 0 when it reaches 65535.

***Port*** Either 0 or 1. 0 indicates that the station is over the radio from the node. A 1 indicates that the station is another node on the same node stack.

***Time*** Time accumulated since the station was last heard. Measured in hours:minutes:seconds.

***Type*** If blank the station is probably a normal TNC. If *Node* the station has been heard sending TheNET/MSYS/G8BPQ or NET/ROM packets. If TCP/IP then the station was heard sending AX.25 TCP/IP frames.

If more calls get heard than there is room for on the heard list the older ones fall off the end. Callsigns that are not heard for 12 hours are removed from the list.

STMFRD:KB2CS-1}

| Callsign | Pkts | Port | Time | Type |
|---|---|---|---|---|
| WA2WNI-12 | 58522 | 1 | 0:00:00 | Node |
| W2RGI-14 | 1291 | 1 | 0:00:01 | Node |
| WA2VAM-14 | 48625 | 1 | 0:00:01 | Node |
| KA2IPF | 733 | 0 | 0:00:09 | |
| KB2CS-13 | 41140 | 1 | 0:00:12 | Node |
| WB2EFB | 571 | 0 | 0:01:45 | |
| KB2CS-14 | 20844 | 1 | 0:07:25 | Node |
| WB2EFB-1 | 35 | 0 | 0:45:11 | |
| KC2IV-4 | 48 | 0 | 1:11:20 | |
| N2NDP | 315 | 0 | 1:16:20 | |
| KA2QHL | 199 | 0 | 1:28:24 | |
| N2QNI | 38 | 0 | 2:30:17 | |
| WA2UXE | 2 | 0 | 2:38:53 | |
| N2EIT | 92 | 0 | 4:19:32 | |
| KB2KNX | 83 | 0 | 4:49:29 | |
| N2LWC | 93 | 0 | 5:08:25 | |
| N2GWT | 559 | 0 | 5:39:10 | |
| WA2EYH | 13 | 0 | 5:56:33 | |
| KB2FIO | 1583 | 0 | 6:41:56 | |
| KA2SIW | 6 | 0 | 8:02:02 | |

## MANAGER

This command is much like SYSOP, described later in this section.

## MODE

This command tells the node to read back it's current MODE settings. The MODE settings are 17 of the many configurables in the X-1J node. The list looks like this:

FRED:KA3QSY-4} 0 0 6 2 0 30 0 450 3 600 1 19 3 1 1 3 3

Follows is a list of the *mode* parameters and what they mean:

| *#* | *Min* | *Max* | *Function* |
|---|---|---|---|
| 1 | 0 | 1 | RS-232 host mode (normal/DCD mode) |
| 2 | 0 | 3600 | CWID repeat period ( seconds ) |
| 3 | 4 | 10 | CWID keyer speed ( 10's msecs per dot ) |
| 4 | 0 | 3 | Selective nodes broadcast on ports: 0=none, 1=HDLC only, 2=RS232 only, 3=Both ports |
| 5 | 0 | 3 | RS-232 crossling protocol: 0=crosslink, 1=KISS, 2=KISS+selcopy, 3=KISS+allcopy |
| 6 | 0 | 255 | Transmit keyup delay ( 10's of milliseconds ) |
| 7 | 0 | 1 | Full duplex |
| 8 | 0 | 65535 | Crosslink port node b'cast interval ( secs ) |
| 9 | 0 | 3 | Node broadcast algorithm port control: 0=off, 1=HDLC, 2=RS232, 3=both ports |
| 10 | 600 | 3600 | Beacon period( seconds ) |
| 11 | 0 | 2 | Connect redirector |
| 12 | 0 | 127 | User message control flags |
| 13 | 0 | 3 | Hash node broadcast port control |
| 14 | 0 | 1 | Extra aliases |
| 15 | 0 | 1 | Auto reconnect to node |
| 16 | 0 | 3 | Control of slime trails. Each bit controls a function: Bit 0 if set hides slime trails in nodes listing Bit 1 if set causes slime trails to be ignored |
| 17 | 0 | 3 | Digipeat up/downlink control: Bit 0 set causes node to refuse digi'd L2 uplinks Bit 1 set, node refuses to allow digi downlinks |

Complete descriptions of the function of each of the MODE options is presented in the *Parameters & Modes* section.

## NODES

This command gives the user access to the routing table in each node. After connecting to a node a user can use N, N * and N <alias or callsign> to get information from the node about it's routing table.

The **NODES** command (abbreviated **N**) returns a listing of the user port nodes contained in the routing table. It gives a user a listing of possible destination nodes for him to connect to.

**N**

```
SRTGA1:WA2UMX-1} Nodes:
WA2PVV              WA2UMX-4             BBSUMX:WA2UMX
GFL:N2AYY-1         KINDER:WA2PVV-7      NEDALB:WA2WNI-3
OTSEGO:W2SEU-1      SCROWD:WA2UMX-7      SRTGA2:WA2UMX-2
SRTGA5:WA2UMX-5     SRTOGA:WA2UMX-14     WMA220:K1FFK-2
```

The listings which include a six character (or less) mnemonic followed by a callsign are nodes whose information was received via a node routing broadcast. The listings which include just callsigns are nodes whose information was received in order to create a return path to a otherwise unknown node. This function is called *Slime Trailing* and is described in the *Theory of Operations*.

The NODES * command (abbreviated N *) returns a listing of all of the nodes contained in the routing table. This includes the # nodes. (See *Selecting Mnemonics*)

N *

SRTGA1:WA2UMX-1} Nodes:

| | | |
|---|---|---|
| WA2PVV | WA2UMX-4 | #GFL10:N2AYY-10 |
| #SCR10:WA2UMX-10 | #SCR11:WA2UMX-11 | BBSUMX:WA2UMX |
| GFL:N2AYY-1 | KINDER:WA2PVV-7 | NEDALB:WA2WNI-3 |
| OTSEGO:W2SEU-1 | SCROWD:WA2UMX-7 | SRTGA2:WA2UMX-2 |
| SRTGA5:WA2UMX-5 | SRTOGA:WA2UMX-4 | WMA220:K1FFK-2 |

The N command can be used to determine the neighbor node and quality for a particular node. The syntax is N <alias or callsign>. An example:

We are at the CANTON node and wish to know the route to POTSDM. We issue a N POTSDM command and receive back this response:

CANTON:WA2MZF-5} Routes to: POTSDM:K2CC-1

```
> 128 3 1 #CNTPO:WA2MZF-11
  81  3 1 #CNTWA:WA2MZF-10
```

This tells us there is a route to POTSDM and it is an RS-232 path (the 1) via WA2MZF-11 which is a backbone node and this route is currently in use. The > symbol indicates the route is in use. The 128 is a quality value. Quality is actually a terrible word for this but it's the word we're stuck with. Quality values indicate how far away a node is through the network. The value starts at 256 and is degraded across every link or diode matrix. The smaller the number the further away from the start you are. The next number, 3, is the obsolescence of POTSDM in CANTON's node table. The obsolescence is decremented each node broadcast interval. This is much better explained in the *theory of operations* section and in the *Modes and Parameters* section. The next number, 1, indicates the route to POTSDM is via the port 1 which is the RS-232 port. Port 0 is over the radio. Finally the next neighbor on the way to POTSDM is listed. The next line (there can be up to three listings) is an alternate path. It's quality, obsolescence, port number and neighbor are shown. In many (if not most) network arrangements the alternate route is useless.

The N <alias> command is extremely valuable for mapping and hacking networks.

## PARMS

This command is very similar to MODES. The PARMS (parameters) are only different because of historical reasons. In earlier versions of TheNET the PARMS existed. MODE is a creation of G8KBB (author of TheNET X) and all of the PARM-type adjustables that G8KBB created are in the MODE command. This is an example PARMS response:

HCKNSK:KA2DEW-1} 100 50 0 203 7 5 900 13 180 1 1 120 2 4 7200 255 10 5 1 10 50 32000 0 1 2 1

Follows is a table of PARMS:

| # | Min | Max | Function |
|---|---|---|---|
| 1 | 1 | 400 | Size of destination node table |
| 2 | 0 | 255 | Minimum auto update quality |
| 3 | 0 | 255 | HDLC (radio port) default quality |
| 4 | 0 | 255 | RS-232 (crosslink) default quality |
| 5 | 0 | 255 | Initial obsolescence count |
| 6 | 1 | 255 | Minimum obsolescence count for broadcast |
| 7 | 0 | 65535 | Nodes broadcast interval (seconds) |
| 8 | 0 | 255 | Initial time to live (hops) |
| 9 | 5 | 600 | Transport (FRACK) timeout (seconds) |
| 10 | 1 | 127 | Transport retry counter |
| 11 | 1 | 60 | Transport acknowledge delay (seconds) |
| 12 | 1 | 1000 | Tranpsport busy delay |
| 13 | 1 | 127 | Transport window size |
| 14 | 1 | 127 | Transport overfill limit (frames) |
| 15 | 0 | 65535 | No-activity time-out (seconds) |
| 16 | 0 | 255 | Persistence (n/256) |
| 17 | 0 | 127 | Slot time |
| 18 | 1 | 15 | FRACK (T1) time |
| 19 | 1 | 7 | AX/25 window size (MAXFrame) |
| 20 | 0 | 127 | AX.25 retries |
| 21 | 0 | 6000 | Ack (T2) time (link layer RESPTIME) |
| 22 | 0 | 65535 | Active check (T3) time |
| 23 | 0 | 1 | Digipeat |
| 24 | 0 | 1 | Callsign validation |
| 25 | 0 | 2 | Beacon mode control (0=off, 1=only if active, 2=always) |
| 26 | 0 | 1 | CQ broadcasts |

Complete descriptions of the function of each of th MODE options is presented in the *Modes and Param eters* section.

## QUIT

Instructs the node to disconnect. The node may giv a "good-bye" message if so configured. This comman is the same as the *bye* command.

## ROUTES

This command yields a listing of all radio line of sigh or wire connected nodes directly worked (at level 2) b the node. These nodes are called *neighbors*. The listin will also show nodes and digi routes set by the syso locking commands. Due to the different protocols in volved, TheNET does not recognize KA-Nodes, ROSI nodes (at least in the current version of ROSE), o TEXNET nodes in it's routes list. It will recogniz G8BPQ, MSYS and compatible TCP/IP nodes. A typica routes display may look like this:

#LFHCK:WA2IKL-14} Routes:

```
> 1 LTLFRY:WA2IKL-1     203 1  !
> 1 #LFKIN:WA2IKL-12    203 23 !
> 1 #WIRE1:WA2IKL-7     203 14 !
  1 #LFPAT:WA2IKL-10    203 11 !
> 0 #HCKLF:KA2DEW-11    203 7  !
```

In column one we see a 1 for all paths through th *matrix* and a 0 indicating a *radio* path to the #HCKLI node. The right arrow (>) indicator tells us four of th paths are either in use or had activity within the past 1 minutes. All radio paths show a path quality value o 203. All RS-232 paths show a path quality value of 203 The last column indicates the number of nodes source from this route. None of these values indicate th expected or real performance of the link except if the # of nodes sourced from a route is zero we can make a goo guess the route is not on the air, sysopped incorrectly, o disabled.

## STATS

The STATS command has no parameters. It prints a number of internal TNC statistics. This command is astoundingly useful. It lets the sysop fine tune his system by knowing when radio adjustments are wrong or when a hidden station is colliding with his link. It lets the user or hacker determine how well a link actually works. By observing the stats results on different links the traffic flow on a system can be analyzed.

Basics: L1 refers to the actual electrical operation of the radios or RS-232 ports on the node TNC. L2 refers to AX.25 links between the node and users servers or other nodes whether over the radio or over the RS-232 port. L3 refers to traffic between two other nodes that passes through the node you're looking at. L4 refers to node to node connects that start or end at the node you're looking at. Here is a sample stats message→→→

```
#SRKL1:N2CGY-12} Statistics
L1 Tx % :     16    18    13    19    21    23
L1 DCD% :     14    12    9     14    17    18
L1 RxOvr:     0     0     0     0
L1 TxUnd:     0     0     0     0
L2 RxCRC:     7     0     12    0
L2 heard:     264   846   681   1070
L2 recvd:     260   343   678   799
L2 sent :     404   344   972   764
L2 RxRNR:     0     0     0     0
L2 RxREJ:     0     0     0     0
L2 TxRNR:     0     0     0     0
L2 TxREJ:     0     0     0     0
L2 fails:     0     0     0     0
L3 g'wyd:     340   468
L4 recvd:     6     0
L4 sent :     4     0
Buffers :     665   659   659   661   659   634
CPU loop:     415   413   418   398   386   388
Timers  :     732   2452
```

Here is a breakdown of what the *stat* dump means:

First we have the L1Tx%. That's the percentage of time the TNC has the PTT asserted. This does include Txdelays. L1DCD% similarly counts all the time the DCD is detecting Rx signal.

```
           ____________________________________ 0 to 10 minutes ago (most recent 10 minute period)
           |     ______________________________ 10 to 20 minutes ago
           |     |     ________________________ 20 to 30 minutes ago
           |     |     |     __________________ 30 to 40 minutes ago
           |     |     |     |     ____________ 40 to 50 minutes ago
           |     |     |     |     |     ______ 50 to 60 minutes ago
           |     |     |     |     |     |
L1 Tx % :  9     20    25    26    28    21     <— Percent time transmitter was on the air
L1 DCD% :  24    36    46    42    38    41     <— Percent time TNC detected data carrier
```

```
           ____________________ Port 0 (radio) Overruns & framing errors current hour (shown as it accumulates)
           |     ______________ Port 1 (RS-232) Overruns & framing errors current hour (shown as it accumulates)
           |     |     ________ Port 0 (radio) Overruns & framing errors previous hour
           |     |     |     __ Port 1 (RS-232) Overruns & framing errors previous hour
           |     |     |     |
L1 RxOvr:  0     99    0     45   <— Receive
```

```
           ____________________ Port 0 (radio) Underruns current hour (shown as it accumulates)
           |     ______________ Port 1 (RS-232) Framing errors current hour (shown as it accumulates)
           |     |     ________ Port 0 (radio) Underruns previous hour
           |     |     |     __ Port 1 (RS-232) Framing errors previous hour
           |     |     |     |
L1 TxUnd:  0     0     0     0    <— Transmit
```

The following are error reports from Level 2. Level 2 is the software in the TNC that talks to immediate neighbor TNCs, both directly over the radio and directly via the RS-232 port (usually over a diode matrix)

```
           ____________________ Port 0 (radio) Current hour (shown as it accumulates)
           |     ______________ Port 1 (RS-232) Current hour (shown as it accumulates)
           |     |     ________ Port 0 (radio) Previous hour
           |     |     |     __ Port 1 (RS-232) Previous hour
           |     |     |     |
L2 RxCRC:  309   5     594   30     <—Frame checksum errors
L2 heard:  401   367   3401  2095   <—Packets heard
L2 recvd:  372   313   1472  833    <—Packets received by node
L2 sent:   385   363   1702  913    <—Packets sent by node
L2 RxRNR:  0     0     0     0      <—Receive Not Ready packets received
L2 RxREJ:  0     0     1     0      <—Reject packets received
L2 TxRNR:  0     0     0     0      <—Receive Not Ready packets sent
L2 TxREJ:  0     0     0     0      <—Reject packets sent
L2 fails:  1     0     0     1      <—Number of link time-outs
```

```
                ___________ Frames Current hour (increments as you watch)
                |      _____ Frames Previous hour (increments as you watch)
                |      |
L3 g'wyd:      116    606   <—Number of level 4 frames gatewayed between nodes
L4 recvd:      140    193   <—Number of transport frames received by the node
L4 sent:       164    190   <—Number of transport frames sent by the node

                ______________________________________ 0 to 10 minutes ago
                |     ________________________________ 10 to 20 minutes ago
                |     |     __________________________ 20 to 30 minutes ago
                |     |     |     ____________________ 30 to 40 minutes ago
                |     |     |     |     ______________ 40 to 50 minutes ago
                |     |     |     |     |     ________ 50 to 60 minutes ago
                |     |     |     |     |     |
Buffers :      601   547   524   541   551   592   <— Free Buffers
CPU loop:      298   274   276   286   297   290   <— CPU loading - Number of times divided by 100 that the CPU
                                                     makes it around the internal scheduler. If this drops below 100th
                                                     CPU is running out of steam.

                ___________ Hours since last warm start
                |      _____ Hours since last cold start
                |      |
Timers:        38     1296
```

For level 1, six pairs of numbers are printed, corresponding to the percentage of time the transmitter was on followed by the percentage of time the receiver DCD was on, for each of the last six 10 minute periods. The data is presented most recent period first. Two pairs of numbers are then displayed showing the transmitter underrun and receiver overrun. These are formatted with port 0 followed by port 1 for the current hour followed by the totals for the previous hour. In the case of the RS-232 port, underruns are not possible, and an additional error (framing) is included. The Rx overrun includes overruns and framing errors.

For level 2, the following are displayed :

Frame checksum errors Total packets heard
Total packets received by the node (i.e. sent to it)
Total packets sent by the node
Total receiver not ready packets sent
Total reject packets sent
Total receiver not ready packets received
Total reject packets received
Total number of link time-outs

For each of the level 2 statistics, four numbers are shown. The first two are cumulative totals over the period of one hour, incrementing in real time. The last two are the totals for the previous hour. Each pair of numbers is the total for the radio port followed by the total for the RS-232 (cross-link) port.

For checksum errors, port 0 shows CRC errors and port 1 shows (when in 'cross-link'/matrix protocol mode only), checksum errors. As HDLC errors can be triggered by noise, acceptance of CRC errors is conditioned by the state of the DCD line. If DCD is on and an error is signalled, it will be added to the count. This reduces the false counts, but does not eliminate them. Distant stations that keep the squelch open (just) without being properly heard will result in lots of apparent errors.

For level 3, the number of level 4 frames gatewayed between nodes is displayed.

For level 4, the number of transport frames sent and received by the node are shown.

For level 3 and 4 statistics, two numbers are shown. The first is the number of frames accumulating for this hour, and the second number is the total number of frames for the previous hour.

For CPU health checking, two statistics are shown, the CPU loading and the buffer usage. Each looks like the level 1 stats with 6 numbers corresponding to the last six 10 minute periods.

The CPU loading shows the number of times, divided by 100, that the CPU makes it around its basic internal scheduler. For a node just switched on, receiving nothing, this will be about 470ish for a 4.9 MHz clock. With lots of nodes, a heard list of 20 stations and 70-80% activity on the radio channel for it to listen to, this can drop to about 350ish. If it drops to double figures, worry, as the CPU is beginning to thrash. At low double figures, the CPU is pretty much working flat out. At that point you should consider: increasing the TNC clock rate; breaking up your diode matrix/cluster into two or more diode matrices/clusters; reducing the speed of the diode matrix communications.

The BUFFERS statistic shows the minimum number of free buffers the software had available to it during the last six 10 minute period. This indicates whether the TNC is failing to deliver data passed to it for onwards transmission, as well as how much data is backed up waiting.

The last line shows number of hours since the last warm start and hours since last cold start.

## SYSOP

This command is used to request a verification string to become node sysop. Sysop mode lets the authorized person or people make adjustments to the node. The node will respond with five numbers of range 1 -> 79. The user then types a five character response. The node will not acknowledge the response. If the response was correct the node will set a flag indicating for the duration of your connect you are able to make changes as per the sysop commands list.

## TALK

Talk is a conferencing command. It allows a number of stations to hold a simultaneous conference (a bit like the CROWD node). There is only one conference, and stations may connect to it by connecting to the node and issuing the TALK command. It may be exited by disconnecting or issuing the command '/EXIT' at the start of a line. (/EXIT may be abbreviated /EX, and is not case sensitive).

Each line sent by a user is copied to all other users in the conference, preceded by the callsign of the user.

Whenever a new station enters the conference, or a station leaves the conference using the '/EXIT' command, the other conference users get a message informing them of the event. These status messages are sent with the callsign of the node rather than the user.

Finally; when entering the TALK command, a message may be sent to all those users who are connected to the node but not otherwise doing anything. For example if GxABC enters the line

`TALK Hello fred, can I have a chat, type 'TALK'`

Then all other stations connected to the node, present in the USER list but idle, get the message

`K1QRM>> Hello fred, can I have a chat, type 'TALK'`

displayed on their terminal.

Note that merely connecting to the node does not consitute being listed in the users list. A station must have excecuted at least one command on the node before he becomes listed. A TheNET node will assume any new connectee is another TheNET node until the first exchange of information. If you type a command, the node realizes you aren't a TheNET, and that's when you get listed in the user list.

## UI

The UI command allows a string to be sent as a Level 2 UI frame. The syntax is

`UI dest string_of_text_ending_in_return`

***Dest*** is a callsign like destination such as 'MAIL'.

***string_of_text..*** is up to 230 characters of text.

What will happen is a single UI frame will be sent with a source callsign of the user who entered the command,a destination callsign of dest, and the rest of the string as text.

This is similar to the CQ command but the user is not listed on the user list as calling CQ. It is designed to be used in situations where a local BBS does not have access to a common channel and wishes to send mail notification packets. Not surprisingly, the ability to do this is BBS specific.

# Sysop Command List

This chapter describes the special system operator commands and system operator options on normal commands that are available to manage a TheNET X1 node. See the User Commands section for more detail on the commands. In order for a sysop to use these commands the sysop password exchange must first be performed. See *SYSOP* below.

## ACL

Each TheNET node TNC has an Access Control List. This list allows the sysop to selectively disable certain node functions based on the callsign of a station. ACL can:

- eliminate an abusive station that is uplinking or downlinking into the network via a network port that you sysop (ACL callsign + 67);
- protect a dedicated link from on-frequency encroachment by stations other than the two who are supposed to be on frequency (see example below);
- stop stations from a particular node from connecting to or from your node. (ACL nodecallsign + 48). This could get *very* confusing;
- stop stations from a particular node from passing through your backbone node (ACL callsign + 8);

Each time a callsign is heard over the air, over the RS-232 or as part of a network packet (L3/L4) the list is checked. The first entry of a callsign that matches is used. If no entry is found the default value is used. If restriction is found the packet is dumped. If appropriate, an error message may be sent to the originating station to indicate the connect is not possible.

The syntax of a list entry is:

```
callsign   value
```

where value is the sum of the enable/restrict bits. The bits are:

| | *add* | |
|---|---|---|
| bit 0 = | 1 | bar incoming level 2 connection from a station with this callsign. This affects on-channel stations and RS-232 stations only. If callsign is a neighbor network node, this will disrupt node to node traffic through the neighbor 'callsign'. |
| bit 1 = | 2 | bar outgoing level 2 connection (downlink) to a station with this callsign. This affects a station that matches callsign on the radio port or RS-232 only. If callsign is a neighbor network node, this will disrupt node to node traffic through the neighbor 'callsign'. |
| bit 2 = | 4 | ignore nodes broadcasts from this station (only if callsign is a neighbor) |
| bit 3 = | 8 | bar gatewaying at level 3 from node with this callsign. |
| bit 4 = | 16 | bar incoming level 4 connections from a node with this callsign |
| bit 5 = | 32 | bar outgoing level 4 connections to a node with this callsign |
| bit 6 = | 64 | ignore SSID in matching an entry |

The enable/restrict bits are added together to form th value. For instance a station listed as:

```
KA2DEW     1
```

is not allowed to connect to the node. KA2DEW- would be permitted to connect and anybody could cor nect to KA2DEW.

This is the syntax of the ACL command.

| | |
|---|---|
| ACL * value | set default value |
| ACL callsign + value | lock in a callsign entry |
| ACL callsign - | remove a callsign entry |

If you are not sysop, or if ACL is given on its own, th current contents of the ACL are shown. The first forı of the command changes the default value, the secon form makes an entry in the list, the last form remove an entry from the list. It complains about syntax eı rors but does not respond at all to valid ACL changes

A few moments thought will show the sequence of commands to the right is quite catastrophic. You will not be able to get back in again apart from via the host port and no-one will be able to connect to or from the node. If you intend to experiment with the command you should only do so on a node that you hav host-port access to, or have somebody on site who ca conveniently do so.

```
C NODE
SYSOP
RfFle
ACL * 127
^C
cmd:D
*** Disconnecte
```

The list can be used as an 'accept' or 'reject' list b judicious use of the default. To create a list that exclude specific calls, put them into the list with the require bits set in the value. The default should be zero. T create an 'accept' list, put entries in with the require bits zero and set the corresponding bits in the defaul Individual bits may be used to create accept or reje lists for each function.

The command steals buffers at a rate of one buffe per four entries in the ACL. Also, a long ACL will slo the node down nicely - so think before you enter a lor list.

***Example:***

To protect a dedicated link backbone between th HCKNSK:KA2DEW-1 node and the LTLFRY:WA2IKI 1 node sites we go to each of the two # nodes and do th following commands:

| | |
|---|---|
| ACL KA2DEW + 64 | allow all KA2DEW callsigns to perform all functions. |
| ACL WA2IKL + 64 | allow all WA2IKL callsigns to perform all functions. |
| ACL * 7 | ignore on-channel connect in or out ignore node-broadcasts. |

## ALIAS

The ALIAS command allows the node's alias to be changed. The syntax is :

| | |
|---|---|
| ALIAS ? | ask for the current alias. |
| ALIAS * | clear the alias |
| ALIAS new-alias | specify a new alias. |

If new-alias is deemed to be a valid alias, the node's alias is changed to the new one entered. [*Note the algorithm that checks for the alias structure is a bit queer. It is however, the original algorithm of TheNet and I am loathe to change it for fear of side effects. Note too the companion CALLSIGN command is not included - chaos is not something I crave.-g8kbb*] If the sysop gives the parameter of '*', the node's alias is cleared.

## ARP

Add or modify the ARP table. The syntax of the command is:

| | |
|---|---|
| ARP | print the current ARP list |
| ARP addr | print the ARP listings that match *addr*. If an illegal addr is entered the node will return the entire ARP list. |
| ARP addr + [ P ] protocol call [ mode ] | adds new entry |
| ARP addr - protocol | removes entry |

In the first form an entry is made in the table, in the second an entry is deleted. This is only permitted for sysop or manager.

The parameters are :

| | |
|---|---|
| addr | An address of the form nnn.nnn.nnn.nnn |
| P | If present, marks the entry as 'published' |
| protocol | AX25 or NETROM |
| call | A valid amateur callsign, e.g. G8KBB-5 |
| mode | DG or VC |

*See User Commands section for more information on ARP.*

## AUDIT

As described under *Non-Sysop Commands*, the audit function allows a sysop to monitor the operation of the node. In order for this command to operate the sysop must answer the *Manager* password exchange. The *Manager* exchange is identical to the Sysop exchange.

The syntax of the audit command is :

AUDIT new-value

where new value is an integer value. If no value is given, or the user does not have SYSOP status, the current mask value is displayed. Otherwise, the mask is updated and the new value displayed.

The mask controls the auditing of various events in the node. Not all values are used yet, but those that are, include:

| bit | function |
|---|---|
| 0 | Level 1 statistics on 10 minute updates |
| 1 | Level 2 connects & disconnects |
| 2 | reserved for future use |
| 3 | Level 4 connects & disconnects |
| 4 | Level 7 limited events (use of sysop) |
| 5 | Full level 7 auditing 6 CPU auditing messages (10 minute updates) |

It is suggested that the usual settings can simply be 0 or 63.

For level 1, messages are sent every 10 minutes showing the percentage of time the receiver detected carrier and the percentage of time the transmitter was on.

At level 2 & 4, the messages are of all connects and disconnects, shown in 4 different ways :

| | |
|---|---|
| C | Connect message received by node |
| CA | Connect message sent / Acknowledge received |
| D | Disconnect message received by node |
| DA | Disconnect message sent / Acknowledge received |

In each case, 2 callsigns are shown. At level 2 these are the source and destination of the AX.25 link. At level 4, it is the remote node callsign and user callsign. Each message is preceded by an indication of the source of the message, such as "L2" or "L4".

At level 7, with bit 4 set and bit 5 clear, the only event currently audited is the use of the Sysop command, either directly or via the manager command. If bit 5 is set, all commands given to the switch are audited, preceded by the callsign of the user who entered the command.

Bit 6 controls CPU health check auditing. If set, whenever the internal CPU statistics are updated, messages are sent showing the CPU processor loading total and the minimum buffers level (see STATS for more information).

The audit mask value should be set to 0 when not actually being used. Do not leave it set to another value as this wastes processor time. Note also that full auditing on a busy node makes things worse. Treat it as a debugging feature!.

## BBS

Set the *charstring* the node will attempt to connect to when; the BBS command is typed on the node, or; when a station attempts to uplink to the bbs-alias on the node's radio port.

The syntax of the command is :

| | |
|---|---|
| BBS | connect to an already assigned station |
| BBS * | erases the BBS string |
| BBS ? | requests the current setting |
| BBS <charstring> | sets the string the node will connect to. |

## BBSALIAS
## HOSTALIAS
## DXCALIAS

These commands are used to enable the node to respond to up to three additional aliases. The syntax of each is the same, and by way of example the BBSALIAS syntax is :

BBSALIAS *
BBSALIAS new-alias

If not sysop, if no new alias is specified, or if it does not pass the weird alias syntax checker (*see ALIAS*) the current alias is displayed. If not, the alias is changed. If '*' is given, the alias is cleared.

The aliases so entered have nothing to do with the node's identity. If a BBS alias is set, for example to MXMBBS, the node will listen for level 2 connects to that alias. It will respond to them and will automatically invoke the BBS command. The use will also get the optional welcome (ctext) message and 'trying to connect to ....' messages if enabled by the appropriate 'mode' parameter.

The idea is where a node sits on a channel that does not have access to the local host, BBS or cluster, the normal aliases of those stations may be enabled in the node to allow consistent access to the local services. Note the three stations do not have to be a BBS, Host and cluster, it could be three BBSes or any other combination.

## BTEXT

The BTEXT command sets or displays the additiona beacon text sent along with the beacon packets. Set the beacon interval in the PARMS command.

The syntax of the command is :

BTEXT message
or
BTEXT *

The BTEXT storage area has room for 160 characters The message may be multiple lines long, so long as it doesn't exceed 160 characters. To change the BTEXT message do the following:

Type BTEXT * to clear the BTEXT.

Type BTEXT *texttextext* to write the first line.

Type BTEXT *texttexttext* to write the second line.

You can keep sending BTEXT lines until you use all of the 160 characters. Each time you send a BTEXT message the complete BTEXT is echoed back. The last line will be truncated if you type too many characters.

An example BTEXT:

MOSSVL:K1QSY-5}
low coverage network access port
C MOSSVL for network access
C BBSQSY for BBS

## CALIBRATE

This command allows remote calibration checks of the transmitter deviation. Its syntax is

CALIBRATE period [ toggle ]

The period (1 to 60 seconds), is the time which the transmitter will key up for with constant tone. It is undefined as to which tone will be sent. If the second parameter is given, the node will toggle between the tones every [toggle] seconds. The toggle must be between 1 and [period] seconds. If a period is not given, the user is not sysop or manager, or if it is out of range, the command is ignored. If the tone generator is busy because it is about to send a CWID sequence, a 'busy' message is returned. Note - quite often it can appear the node has locked up having failed to transmit the full calibrate period. In fact, this is usually the hardware PTT watchdog in the TNC. The node thinks it is still sending but the hardware timer has removed the PTT signal.

### CLOSEDOWN

The closedown command is used to shut down the node remotely. If successfully executed, the node will effectively stop operating until it is reset (e.g. by a power up). The node's configuration (routes, messages etc.) are not destroyed - the node simply hits a HALT instruction. You must be sysop to execute the command.

The syntax of the command is:

CLOSEDOWN A

The node will respond with 5 numbers just as when the *sysop* or *manager* command was executed. Yes, you guessed, the node expects another password. Give it correctly and the node closes down completely. Get it wrong and you lose your sysop status. This obtuse and awkward syntax is designed to make sure it is not accidentally executed.

### CTEXT

The CTEXT command sets a message sent to a user who connects to the node by uplinking to the node's alias. CTEXT is enabled in the MODES command under Help Messages Enable Flags. The CTEXT is set in the same manner as the BTEXT and INFO text commands.

Note: For œsthetic reasons you may want to set the first line of the CTEXT to "}" or something similarly simple. Thus the CTEXT might look like:

HCKNSK:KA2DEW-1} }
Please run low power
Type I for info or ? for command list

### DXCALIAS

See BBSALIAS

### DXCLUSTER

See BBS

### HOST

The syntax of the command is :

| | |
|---|---|
| HOST | connect to the already specified host |
| HOST * | erase the host specification |
| HOST ? | read back the current specification |
| HOST callsign | specify a new host |

This command is very similar to the 'BBS' command. It allows connection to a local host, BBS or other server. The difference however, is as long as the TNC is not in 'crosslink' mode (i.e. pin 23 on the RS-232 port is high), and if a callsign is not set, the 'host' command connects to the local port.

The idea of this command is, like with the 'bbs' command of the 'BPQ software, a user may connect to the local BBS, another node or server from this node. For example, if a print server were connected to the node in 'host' mode, this command would allow connection to it (like the 'connect' command with no other parameter). In KISS mode, setting a callsign or node alias allows connection to that system.

### HOSTALIAS

See BBSALIAS

### INFO

This command changes the RAM portion of the INFO text the user can request from your node. The first 80 characters are in EPROM and cannot be changed.

Syntax is:

INFO texttext text text text text ....
INFO *

INFO followed by a * will delete all of the RAM portion of the info message. INFO followed by text will appended to the message. If you run out of your 160 character RAM space the message will be cut off in midline. You will have to clear the text and start over.

An example for setting INFO:

Sysop action: **I * <return>**

Node response
HCKNSK:KA2DEW-1}
Hackensack NJ
Sponsored by KA2DEW

This information is the EPROM default and cannot be erased after the EPROM is burned.

Sysop action: **I port 144.93 USER**

Node response:
HCKNSK:KA2DEW-1}
Hackensack NJ
Sponsored by KA2DEW
port 144.93 USER

Sysop action: **I cnty Bergen**

Node response:
HCKNSK:KA2DEW-1}
Hackensack NJ
Sponsored by KA2DEW
port 144.93 USER
cnty Bergen

Sysop action: **I Info KA2DEW @ KA2DEW**

Node response:
HCKNSK:KA2DEW-1}
Hackensack NJ
Sponsored by KA2DEW
port 144.93 USER
cnty Bergen
info KA2DEW @ KA2DEW

Sysop action: **I This port for keyboard access to the network**

Node response:
HCKNSK:KA2DEW-1}
Hackensack NJ
Sponsored by KA2DEW
port 144.93 USER
cnty Bergen
info KA2DEW @ KA2DEW
This port for keyboard access to the network

Sysop action: **I from here C BBSDEW and type H NETWORK for information**

Node response:
HCKNSK:KA2DEW-1}
Hackensack NJ
Sponsored by KA2DEW
port 144.93 USER
cnty Bergen
info KA2DEW @ KA2DEW
This port for keyboard access to the network
From here C BBSDEW and type H NETWORK for information

This process can be repeated until the maximum of 160 characters, including non-typing and punctuation, has been reached.

The key things to have in a user port's info are:

*What frequency the port is on;*
*What type of port it is;*
*Where it is located;*
*How further information can be obtained about the network or the club;*
*any special restrictions for the port;*
*Callsign and BBS for the sysop and / or sponsor of the node.*

The key things to have in a backbone port's info are:

*What sites are involved in that link freq;*
*Who sponsored the node and / or port;*
*The band, power, and distance involved;*
*Frequency if useful and reasonable.*

If all of this information is available someone in your area who wants to link to your node can easily get in touch with you and understand what has been done so far. This way network growth can be facilitated.

## IPADDRESS & IPBROADCAST

These commands are used to set or display the IP addresses used by the node. The syntax of each is (by way of example):

IPADDRESS [ ipaddress ]

where ipaddress is in the form

nnn.nnn.nnn.nnn

where nnn is an integer in the range 0..255

So to set the node IP broadcast address to that used over here, the command would be :

IPBROADCAST 44.131.0.0

The IPADDRESS is the address the node will respond to. It is used only as detailed in section 7. The IP broadcast address is the one used to denote broadcast packets that will be largely ignored. [*Note that port addressing is NOT currently supported. Anyone who finds this limiting, drop me a line and I'll see if I can change it. -g8kbb*]

## IPROUTE

This is one of the two main databases used by the node. The IP Route table is used to tell the router where to send a frame for a specific destination. It maps addresses or address ranges to a gateway IP address and to sub-network ports. The ARP database then tells the node what station corresponds to that address and protocol. The node supports two subnet protocols, AX25 and Netrom.

The database is stored in an ordered list, in decreasing order of the number of relevant bits. This is to permit searching of the database when trying to find a specific destination. Given an address, it scans addresses with decreasing numbers of bits until it finds a match.

The syntax of the command is as follows :

IPROUTE address [ /bits ] + port [gateway [metric]]

or

IPROUTE address [ /bits ] -

In the first form, it makes an entry in the table, in the second it deletes one. Only sysop or manager may effect such a change. The parameters are as follows :

| | |
|---|---|
| ***address*** | The amprnet address in the form nnn.nnn.nnn.nnn |
| ***bits*** | The number of significant bits (eg 44.131.0.0 / 16) |
| ***port*** | The port, either 0 or 1 for AX25 or n for NETROM |
| ***gateway*** | The optional gateway for this dest. nnn.nnn.nnn.nnn |
| ***metric*** | Currently not used, a numeric value |

When an entry is made with a specific number of bits, the address is 'masked off' to that many bits, so enter an address of 44.131.16.31 / 24 and it will get entered as 44.131.16.0. The valid range for the number of bits is 1 - 32.

**IPSTATS**

The IPstats command has the same basic syntax as the PARMS and MODE commands. When invoked without parameters, it displays the current stats. Each statistic may also be altered by sysop.

In addition to the standard IP MIB, there is an additional parameter used to set the level 2 default modes, and the first entry in the MIB is used to enable or disable the router.

The complete set of IP MIB stats are included for compatibility with other IP systems, but several are not used. Also, the stats are 16 bit counters not 32 bit counters as in NOS. Like NOS however, the stats do not reset every hour, they must be cleared by the sysop. They will however wrap around at zero.

The entries are:

| | |
|---|---|
| 1 Port default modes | 11 IP Output Requests |
| 2 Enable / Disable IP router | 12 IP Output Discards |
| 3 Default IP Time To Live | 13 IP Output No Routes errors |
| 4 IP Received frames | 14 IP Reassembly Time-out errors |
| 5 IP Header Errors | 15 IP Reassembly Required errors |
| 6 IP Input Address Errors | 16 IP Reassembly OKs |
| 7 IP Forwarded Datagrams | 17 IP Reassembly Fails |
| 8 IP Unknown Protocols | 18 IP Fragmentations completed |
| 9 IP input frames Discarded | 19 IP Fragmentation Failures |
| 10 IP Input frames Delivered | 20 IP Fragmentation Creates |

The default mode word may be set to 0, 1, 2 or 3. Each bit controls a port, with bit 0 controlling port 0 (radio port) and bit 1 controlling port 1 (RS-232 port). When set to 1, the default mode for that port when sending on a level 2 connection will be Datagram. When set to 0 it will be by Virtual Circuit. The default mode is used when no other information is given, either by the ARP table or by the TOS bits in the IP header.

The enable / disable word may be set to 0 or 1. When set to 0, the operation of the router is stopped, when set to 1 the router functions.

The IP Time To Live (TTL) word is used to set the number of routers through which an IP frame may pass before it is discarded. It is similar to the node layer 3 TTL word. It may be set to any value up to 255, but values below 2 make no sense and are therefore not permitted.

The IP fragmentation reassembly time-out counter is not used as the node is just a router. It is left set to 30 seconds just to show which one it is !

The rest are just statistics. The patient user can have hours of fun working out which ones are not used (or just think about it for a second or two).

**MANAGER**

The MANAGER command gives the user extra privileges. In this version, this amounts to the ability to receive audit messages from the node. The level of auditing is set by the AUDIT command.

The privilege remains in force until cleared by a command that affects the user state. These are: entering the TALK state, executing the SYSOP command, entering the MANAGER command and getting the password wrong, or disconnecting from the node. Failing to get the password right when using the closedown command will also remove the manager privilege.

A user with MANAGER privilege also has SYSOP privilege.

**METERS**

The Meter command is used to control the ADC functions of the software. In this version, this is limited to the Deviation meter, but future releases may extend this, for example to configure a signal level meter.

The syntax is as for the PARMS commands. It currently has only one parameter. When set to 0, the deviation meter is disabled. When set to a value in the range 1-255, the meter is enabled and the value is used as a scaling factor. The ADC is an 8 bit device, so it will give a response in the range 0-255, corresponding to an ADC input voltage in the range 0-3 volts DC. If optimally configured, this corresponds to the maximum audio level possible for the given receiver discriminator.

The ADC reading (0-255) is multiplied by the meter parameter value (1-255) to give an answer in the range 0 to 65 kHz (approx). This is the value displayed in the mheard list. Hence, if, for example, a DC voltage of 2 volts at the input to the ADC corresponds to 3.4 kHz deviation, the ADC reading will be 171 (± a few ) and the Meter parameter will need setting to 20 (ie to 3400/171 ). If the ADC reading is 254 or higher, in order to indicate an overrange, the symbol '>' will precede the corresponding deviation entry in the heard list.

## MHEARD

This command specifies the length of the heard list. The command is:

MHEARD #

# specifies the maximum length. Setting it to 0 disables the MHEARD list operation.

The heard list uses free buffers for the list, so a large setting means less RAM for the node software.

The list is maintained as linked list, with the most recently heard station first. The display shows the number of packets heard from that station and the time since it was last heard, in hours minutes and seconds. In addition, it shows the port on which the station was heard together with an indication as to whether the station is a node and / or a TCP/IP station. It does this by examining the PID byte.

Every hour the list is checked for stations that have not been heard for 12 hours, and any such stations are removed from the list.

To disable the internal updating of the list (and thereby stop the CPU expending effort on the function), set the size to zero rather than just disabling the command. Note though, the node will not clear the list as updates have been disabled, so it will be up to 12 hours before the buffers used are freed. To accelerate this process, set the size to 1, wait until it has heard a station (any one will do) then set it to zero. This will free up all but one buffer immediately.

## MODE

The MODE command is used to set many of the adjustables in the X-1J node software. There are two ways to make a change. First lets review the MODE string.

LTLFRY:WA2IKL-1} 0 0 6 2 0 30 0 450 0 600 1 19 3 0 1 3 3

There are 17 numbers that are referred to as Modes. The Modes actually control more than 17 different functions. A complete description of each of the MODE numbers is in the section of this document titled *Parameters & Modes*. The MODE command allows a sysop to change these numbers.

There are two ways to change a mode number. The command may be specified as:

MODE /# value

This will change the number # to the value *value*. For instance, MODE /4 1 will change the above string to:

LTLFRY:WA2IKL-1} 0 0 6 1 0 30 0 450 0 600 1 19 3 0 1 3 3

The other method allows several, or all, of the mode numbers to be changed at once. It is specified as a string of numbers. Numbers that are to stay the same may be specified with an asterisk. Thus to change numbers 5, 6 and 7 to 1, 40, 1 the following command may be used:

MODE * * * * 1 40 1

The result would be

LTLFRY:WA2IKL-1} 0 0 6 2 1 40 1 450 0 600 1 19 3 0 1 3 3

## MTU

The MTU command is used to configure the node Maximum Transmission Unit figures (primarily for TCP IP support). [*In general, they should be left at the default value Do not experiment unless you are sure you understand the significance of what you are doing!!!!-g8kbb*] The syntax of th command is identical to the syntax of the PARMS command. There are 5 values configured by the command These are :

| # | Dflt | Function |
|---|---|---|
| 1 | 256 | Sets MTU for IP router, AX.25 L2, Radio por (port 0) |
| 2 | 256 | Sets MTU for IP router, AX.25 L2, RS-232 por ( port 1 ) |
| 3 | 236 | Sets the MTU for the IP router Net/Rom interface |
| 4 | 257 | Sets the maximum number of data bytes permitted in an AX.25 level 2 frame before an erro response is returned to the sender (frame too long) |
| 5 | 328 | Sets the maximum number of bytes permitted in a level 2 packet. Above this, frames are discarded. If a packet may contain 256 data bytes and a maximum length address of sender, recipient and 8 digis comprises 70 bytes, and 2 bytes are used for control bits, the total (256+70+2) is the setting of this parameter. |

The minimum that an MTU may be set to is 64, the maximum is 1024, but large packets increase the probability of crashing the node. Beware!!!!!!.

The MTU for the Net/Rom port should not in general be set higher than 236 or it will not be compatible with Net/Rom.

The limits on the other two correspond to those necessary to support frames in the range 256-1024 data bytes long.

## NODES

This command allows the sysop to add, delete or lock nodes in the nodes list. In addition a sysop may request a dump of the entire nodes table in a matrix, showing all routes for all nodes. The syntax for the node matrix dump is:

```
N * *
```

Note this command is capable of crashing the node. Since the TheNET software will attempt to package the entire message as a single fragmented message the Windowsize and Congestion Control buffer space settings will not affect the amount of memory used to buffer the response from this command. If all of the available buffers are used to store the result of the N * * message the system will crash. It is unlikely you will crash a node that has < 30 nodes listed in it's table. The crash will take the form of a soft reset. All of the users will be dumped but settings should not be changed. (no guarantees).

The other form of the NODE command sysops will use is the add/delete/lock form. This is the syntax:

```
N callsign +/- nodename qual obs port neighbor
```

where:

| | |
|---|---|
| callsign | callsign of the node being changed. |
| +/- | + to add or lock, - to delete |
| nodename | nodename of node being changed, also called the mnemonic of a node. |
| qual | quality node is listed or will be listed |
| obs | # of node broadcasts before listing goes away, 0 = never goes away |
| port | current port or new port for listing |
| neighbor | neighbor node's callsign. |

This form of the command is used to modify the node list. Specifically you may change one neighbor entry for one node with each N command. Each node entry has up to three possible destinations. If a new destination is presented either through an automatic nodes broadcast or via sysop command the three destinations with the highest quality values will remain. This is very important. See the examples below.

For all explanations below assume this result for the command N GOZER:

```
#KRGOZ:KA2DEW-6} Routes to: GOZER:N2QRM-1
> 191 4 0 #GOZKR:N2QRM-4
  61  5 1 #KRMOG:KA2DEW-11
  61  5 1 #KRCHE:KA2DEW-12
```

This listing shows us there is some distant node called GOZER:K2QRM-1. The best route to GOZER is over the air (port 0) to neighbor K2QRM-4 which is a backbone port named #KRMOG. The obsolescence of the link is 4 which means if no update is heard within four node broadcasts from N2QRM-4 this route will be forgotten. The second best route and third best route are to neighbors over the RS-232 port and are both at quality 61.

### *changing the nodename of a remote node*

The nodename for a node is only present for the convenience of the users. It is never used by the nodes for routing. The callsign is the pointer into the node database. We can change the nodename associated with a callsign in the database. The nodename will change to whatever the next node broadcast has for that callsign, so this effect is temporary (unless of course you make it so your node will never hear about the remote node via a broadcast).

```
n n2qrm-1 + gatekp 191 5 0 n2qrm-4
```

will change the nodename from GOZER to GATEKP.

```
#KRGOZ:KA2DEW-6} Routes to: GATEKP:N2QRM-1
  191 5 0 #GOZKR:N2QRM-4
  61  6 1 #KRMOG:KA2DEW-11
  61  5 1 #KRCHE:KA2DEW-12
```

### *deleting a node entry*

To delete the listing of one neighbor-route for a certain node the command is used with the minus sign. The callsign, nodename, port number and neighbor are critical. If the callsign, port number or neighbor are specified incorrectly this command will have no effect. If the nodename is specified incorrectly this command will work but will also change the nodename. The quality and obsolescence specifications are not important but must be included. Thus, to remove the 2nd neighbor entry for GOZER the command is as follows:

```
n n2qrm-1 - gozer 1 1 1 ka2dew-11
```

The response will be:

```
#KRGOZ:KA2DEW-6} Routes to: GOZER:N2QRM-1
  191 4 0 #GOZKR:N2QRM-4
  61  4 1 #KRCHE:KA2DEW-12
```

If the sysop wanted to remove the entire K2QRM-1 node entry the N command would have to be used 3 times, one for each neighbor. Note that node broadcasts from neighbors may put the node entry back in! In order for a node deletion to stay, all sources of node broadcasts would have to be silenced. One way to do this is to go to all of the # ports at a given site and lock in nodes with quality less than min-to-broadcast.

### *adding a node entry*

The N command specified with the plus sign is used to add an entry.

```
n n2qrm-2 + foo 50 4 1 ka2dew-3
```

will result in a node being added:

```
#KRGOZ:KA2DEW-6} Routes to: FOO:N2QRM-2
  50 5 1 KA2DEW-3
```

Note that since KA2DEW-3 is an unknown node to KA2DEW-6 the nodename for KA2DEW-3 is not shown.

This command may be used to change the value of an existing route-to-a-node by using the command with the existing node's call, nodename, port and neighbor. For instance, in the example shown above if we wanted to make #KRMOG the best route to take we could either reduce the quality of the #GOZKR route or increase the quality of the #KRMOG route.

***locking in a node***

If the sysop wants to lock a node in that is not heard via a node broadcast, or if the sysop wants to lock a node at a value different from that heard via a node broadcast, the command is used much like in *adding a node*, above, except that the obsolescence is specified at 0, which equates to infinity. Thus:

n n2qrm-2 + foo 50 0 1 ka2dew-3

will result in a node being added and *locked*:

#KRGOZ:KA2DEW-6} Routes to: FOO:N2QRM-2
50 0 1 KA2DEW-3

This node listing will only go away if it is removed by a sysop, if the node is cold-reset, or if three other entries get into the node database at higher quality values (i.e. > 50 for this example).

***Notes on N command***

When a node entry is made by the sysop, callsign checking is forced *on* rather than being determined by the callsign checking parameter.

## PARMS

The PARMS command is used to set many of the adjustables in the X-1J node software. The adjustables are listed in the *Parameters & Modes* section. This command is used the same way as the MODE command.

## RESET

The syntax of the command is

RESET warm-start - dump users, don't change anything else, like cycling power.

RESET A cold-start-go back to EPROM values, like erasing the RAM and cycling power.

In either of these cases the node performs the reset immediately. You will not be cleanly disconnected and will appear to be hung. You'll have to disconnect from your end.

## ROUTES

This command allows routes to specific neighbor nodes to be locked in or changed. To permanently add a route to the neighbor route table the command is:

R <port> <neighborcall> + <pathquality>

*port* is either 0 or 1.

0 means *route over radio port*

1 means *route over RS-232 port*

*nodecall* is the callsign of the neighbor node whose route you are adding

*pathquality* is the value that will be used in the node routing broadcast interpretation. See *Nodes Broadcasts* in the *Theory of Operation* section.

To unlock the routes or to change the quality value for an non-locked route use:

R <port> <neighborcall> - <pathquality>

When unlocking a route all of the data in the command must exactly match the locked data. To change the quality value for a non-locked route simply specify the new PATHQUALITY.

An example:

R 0 N2CGY-3 + 143 WA2JWJ-1

The result of this operation might look like:

MAL:W2RRY-1 Routes}
0 DKC:W5YI 50 9
0 CLOUD:W2VXY 192 43
0 WMASS:N2CGY-3 143 1

Here we see the exclamation mark which indicates a locked entry.

The most common example of locked routes is in a backbone link which is supposed to be protected and dual ended. You may lock in the neighbor route and set the radio channel 0 path quality (Parm 3) to zero. This protects against unauthorized backbone use or misrouting caused by propagation or DX. The wanted routes would then be locked in at quality 203. This means all nodes sourced from the neighbor will have routing qualities based on the 203. See *Theory of Operation* for more information on quality calculations.

0 DKC:W5YI 50 9

In the routes list the second value after the neighbor callsign (in this case =9) is the number of nodes sourced from the listed route. If a route is locked this value may be 0, indicating that no nodes are sourced from the neighbor.

Changing the value of an established (but not locked) route may also be done with the routes *minus* command. Note attempts to remove a route which is sourcing nodes will not be effective. The best you can achieve is to set the route quality to 0. If a node is locked using a route you want to remove you must first unlock the node. If a node is locked to a neighbor for which there is no route, a route will be created automatically at the quality with which the locked node is set.

### SYSOP

In order for any sysop command to be executed the connecting station must be sysop. This command is used to obtain that status.

The command is typed with no parameters. The node will respond with five numbers separated by spaces. Each number refers to a single character in the password string. 1 refers to the first character. 22 refers to the 22nd character. All you do is type the five characters indicated and hit <return>. The node will *not* tell you that you have been successful. This would be too obvious to a listening station. You can actually run the SYSOP command several times, correctly answering only one password and falsely answering the rest. In addition you may type extra characters before the valid five and extra characters after the valid five characters (on the same line). The five correct characters will be lost in the string of characters you type. A listening station won't be able to figure out which five characters you answered correctly or which number string you answered correctly. .

To test the success of your SYSOP command, type **P**. This will give you a string of numbers, representing the default values for the various node parameters. Note the value of the first number (typically 100).

Now type P 99. If successful, the first parameter should have the new value 99. Again type P space and insert the original number back in the parameter listing (P 100).

Although password strings may be up to 80 characters long it is common practice to keep them as short as 40 characters. Sample password string:

FRED WAS A BIG HERO AROUND HERE UNTIL TH

I usually write out my password strings in a matrix so they are easy to translate:

| | x0 | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x | !! | F | R | E | D | | W | A | S | |
| 1x | A | | B | I | G | | H | E | R | O |
| 2x | | A | R | O | U | N | D | | H | E |
| 3x | R | E | | U | N | T | I | L | | T |
| 4x | H | | | | | | | | | |

Remember that the first valid digit is 01. So, the following exchange would properly sysop the LYNNWD node:

SYSOP

K7QRM:LYNNWD} 31 13 40 01 08

The 31 indicates the 31st character in the password string. From the matrix we get the second character from the 3x line. The first character is character 30 and is a R. The second character is character 31 and is an E. Next we get character 13 from the 1x line which is an I. Continue for all five characters. Note that the node will not return a number that represents a space character. So, I type:

EIHFS

The node won't respond to this so in order to verify that I got in, I type:

**P**

K7QRM-1:LYNNWD} 100 50 0 203 7 8 900 7 180 1 1 120 2 2 7200 255 10 5 1 10 50 32000 0 1 2 1

**P 99**

K7QRM-1:LYNNWD} 99 50 0 203 7 8 900 7 180 1 1 120 2 2 7200 255 10 5 1 10 50 32000 0 1 2 1

**P 100**

K7QRM-1:LYNNWD} 100 50 0 203 7 8 900 7 180 1 1 120 2 2 7200 255 10 5 1 10 50 32000 0 1 2 1

Note the first parameter changes from 100 to 99 and back to 100. If it doesn't change you have the wrong password or you didn't respond correctly. It is very important the parameter is changed back! So, if you do this procedure don't change the first parameter by too much. If you were to change it to 5 and then weren't able to change it back (Your tower just blew down) the node would soon become useless.

A suggested alternate way to record password strings is with the character numbers on the line above the password. This lets the sysop record several passwords on one page. Example:

```
0    5    10   15   20   25   30   35   40
|    |    |    |    |    |    |    |    |
 FRED WAS A BIG HERO AROUND HERE UNTIL TH
```

### Note on All Sysop Commands:

There is often a requirement to be able to disable the connect command while allowing level 3 relaying. This is achieved by the use of a command qualifier, the syntax of which is :

CONNECT -

The opposite command is

CONNECT +

This has no effect on layer 3 relaying. Also, the BBS and HOST commands will still allow connections to be made if they are enabled and set.

Further, the syntax is valid for *all* commands, for example the CQ command can also be disabled in the same way. Be careful though. The command is only accepted from the sysop, so if you disable the sysop and manager commands you will lock out remote management!

*Do not disable commands that report routes, nodes, parms, stats, users, modes, etc. These are incredibly valuable for network debugging and educational purposes. You might disable the IP functions if IP is not being used. Disabling functions that have use for education is counter productive!*

# Modes and Parameters

### RS-232 host mode (normal/DCD mode) (mode /1)

This parameter controls the 'host' mode. This is the mode of operation of the RS-232 port when pin 23 is 'high'

The valid values are 0 or 1.

In mode 0, the port operates as per the standard node specification. Mode 1 is designed to allow connection to hosts or modems or similar equipment that expects a 'CD' type of signal to signify an incoming/outgoing connection is called for.

In mode 1, the <escape> C and <escape> D commands are disabled and the other <escape> commands do not operate when connected. Instead, hardware handshakes are used to control connections to and from the TNC.

The TNC monitors pin 20 to determine the state of the host, and signals state changes to the host with pin 5. When an incoming connect request is received (by the 'c' command with no parameters or by the 'host' command), the TNC raises pin 5 to signal the connection and expects pin 20 to change state in response.

When the host wishes to connect to the TNC, it signals on pin 20 and the TNC responds by changing the state of pin 5.

It handles disconnects in a similar manner. Either the node or the host may initiate disconnects.

This mode is experimental, changes may be made to its operation. It is designed for modems, print servers or hosts such as UNIX system TTY login drivers.

### CWID repeat period (mode /2)

This parameter is the CWID repeat period in seconds. 0 disables it but do not set it below 120 apart from to disable it.

(Range 0 - 3600) in seconds

### CWID keyer speed (mode /3)

This parameter controls the keyer speed. Specifically, it sets the number of 10 millisecond periods per dot and per inter symbol delay.

The speed of sending is 120/n, so setting n to 6 gives 20 wpm. Valid values are 4 to 10, corresponding to speeds of 30 and 12 wpm respectively. Setting this number to zero is the same as asking the universe to disintegrate at above the speed of light.

(Range 4 - 10)

### Selective nodes broadcast on ports (mode /4)

This parameter allows control to be exercised over which ports nodes broadcasts are sent. Valid settings are 0-3.

0 = no nodes broadcasts (the node broadcast interval still occurs and obsolescence values are still decremented)

1 = broadcast on HDLC port 0 only.

2 = broadcast on RS-232 port 1 only.

3 = broadcast on both ports.

The reasons this might be done is to

- discourage node to node communications on a user channel
- reduce clutter on a user channel
- hide the node (if held for a backup route) or
- in concert with locking the node in at another location, this can be used to create a gateway or dedicated use link.

This function does not affect a user's ability to ask fc and receive Node lists.

### RS-232 crosslink protocol (mode /5)

This parameter is used to set the communication protocol used on the crosslink port when pin 23 is tie low. The valid values are 0, 1, 2 or 3

Mode 0 - standard crosslink protocol enabled

Mode 1,2,3 - use KISS instead of crosslink.

In mode 1, KISS simply replaces the crosslink prc tocol.

In mode 2, packets received from the radio part no intended for the node are copied to the RS-232 port i KISS mode. Similarly packets received on the RS-23 port not intended for the node are sent to the radio port

In mode 3, all packets received on one port are cop ied to the other port as well as being analyzed by th node.

These modes are not simply KISS implementation that replace the node, they run with the node.

Mode 2 is designed to allow a KISS application an a node to share a radio without interference with eacl other. The point is, the PC TCP/IP system can b switched off while leaving the node running to allov others to use it.

Mode 3 is a debugging mode. One problem when faul finding on a node is it is impossible to see what the nod is seeing on the channel without replacing the ROM. B setting this mode, it is possible to connect a KISS ap plication to the RS-232 port and observe what the nod is seeing. Mode 3 is also designed to allow a PC run ning AXSTATS to be connected to the RS-232 port t allow logging and analysis of channel performance fron the node itself. Note packets initiated by the node fo one port will not get copied to the other.

**Transmit keyup delay (mode /6)**

TXDelay in a TheNET node is the same as TXDelay in a TNC2. This adjusts the period of time between keying the transmitter and when it actually starts sending data. If this value is too short the receiving station will not hear the start of the packet and a failure will result. If this value is too long data throughput will be less than optimal.

User ports should not be shorter than 35 or you will exclude stations with slower switching radios from using your user port. Backbone ports should be optimized to find the absolute lowest value that will work reliably with all other radios on its backbone link, then bump the number up a few notches so switching delay drift and squelch performance (due to temperature changes) don't interfere with reliable Tx/Rx switching.

(Range: 0 - 255) in 10s of milliseconds.

**Full Duplex (mode /7)**

This parameter controls whether full duplex is used on the HDLC port. If 1 then use full duplex. This means the node will transmit when it needs to, regardless of the condition of the RF DCD signal or whether it is currently receiving a packet on it's HDLC receiver.

**Crosslink port node b'cast interval (mode /8)**

This sets the rate node broadcasts on the RS-232 port will occur. This does not set the *node broadcast interval* so node obsolescence will still occur at the rate set by PARM #6 (broadcast interval). If this parameter is set to zero (0), RS-232 node broadcasts will occur at the same rate as *Nodes broadcast interval*.

The purpose for setting the RS-232 rate faster is nodes that fail or are newly recognized will propagate across the node stack much quicker while not making a noticeable difference on network performance. RS-232 communications is very quick compared to even 9600 baud radio communications (because of the lack of TXDelay on RS-232).

Note broadcasting of nodes on the RS-232 (crosslink) port can be disabled entirely by *mode #4*.

(Range 0 - 65535) in seconds

**Node broadcast algorithm port control (mode/9)**

This option allows the sysop to control what nodes are included in a node broadcast. The normal/old method was to broadcast all nodes that have a high enough quality and obsolescence that are stored in the node table. The quality value broadcast was always the quality that is the highest of the (up to 3) qualities stored for a node. TheNET X1 now includes an optional broadcast algorithm that on each of the two ports (RS-232 and HDLC) only broadcasts nodes heard on the other port and at the quality value stored for the node via a neighbor over the other port. This means if a node is only heard of via a broadcast over the matrix this node won't include that node in it's matrix broadcast. This feature can be enabled/disabled separately on the RS-232 or HDLC ports.

If this feature is enabled for all TNCs that share a diode matrix, there will never be false routing, routing loops, and it allows multiple HDLC routes to be usable from the same node stack for a particular distant destination node.

In a network that uses dedicated point to point links, enabling the new algorithm reduces broadcast time and allows secondary routes to propagate.

Legal values for this parameter are 0, 1, 2 or 3.
0 = use old algorithm on both ports.
1 = use new algorithm on HDLC.
2 = use new algorithm on RS-232.
3 = use new algorithm on both.

**Beacon period (mode /10)**

This parameter sets the beacon interval in seconds. The node will send a message callsign>ID: every beacon interval. If the BTEXT is set it will be sent in this message.

(Range 600 - 3600) in seconds

**Connect redirector (mode /11)**

This mode parameter allows the sysop to determine what happens when a C command is sent with no parameter. In TheNet 1.01, when 'connect' is given with no destination callsign, the node attempted to connect to the local host port. In a crosslinked system, this vanished down a black hole. In previous versions of this code, the node attempted to connect to the station set by the HOST command, only trying the local host port if no destination was set by HOST. With this version, the node may be configured to connect to the station set by the BBS, DXCLUSTER or the HOST command depending on this parameter. When zero, connect attempts will go to the HOST station, when set to '1', it will attempt to connect to the BBS callsign. When set to 2 it will attempt to connect to the DXCLUSTER callsign.

**User message control flags (mode /12)**

This mode entry controls several options. Each is an on/off switch and is controlled by a binary bit. The bits are added together to obtain the proper value for this mode entry.

| *bit* | *value* | *function* |
|---|---|---|
| 0 | 1 | Issue a message when trying BBS etc. |
| 1 | 2 | Show all available commands to sysop |
| 2 | 4 | Say 'goodbye' nicely |
| 3 | 8 | Give connect text on uplink to alias (CTEXT) |
| 4 | 16 | Show nodes in routes as alias:callsign |
| 5 | 32 | Pass 8 bit data in TALK command |
| 6 | 64 | Make node alias handling case sensitive |

If option 1 and 4 are to be enabled, with the rest disabled, add 2 and 16. This value would then be 18.

***bit 0***

When bit 0 is set, and the BBS, HOST or DXcluster commands are given, a message is sent from the node telling the user a connect attempt is being made.

*Example with mode enabled:*

```
cmd:c qsy
*** CONNECTED to QSY
CHESTER:K4QRP-6} Please wait, trying K2QSY-4
CHESTER:K4QRP-6} Connected to BBSQSY:K2QSY-4
```

In this example BBS was set to K2QSY-4, BBSALIAS is set to QSY and bit 0 of mode /12 is set to on.

*Example with mode disabled:*

```
cmd:c DXBARA
*** CONNECTED to DXBARA
BLTMOR:KA2HCL-2} Connected to DXBARA:N0NDO-2
```

***bit 1***

When bit 1 is set, and a sysop gives an incorrect command, the help screen shows all commands possible, including those currently disabled (as by definition they are not disabled for the sysop!).

***bit 2***

When bit 2 is set, then the use of the *bye* command will solicit a *goodbye* message from the node.

***bit 3***

Bit 3 switches on and off the CTEXT message. When enabled, and a CTEXT message is set, whenever someone uplinks to the node alias the CTEXT message is sent immediately on connect.

***bit 4***

Bit 4 switches the way in which nodes are shown when the ROUTES command is used. When set to 1, nodes are shown as *alias:callsign* . When set to 0, they are shown only as *callsign*.

***bit 5***

Bit 5 controls only the passing of data in TALK mode. Normally, all data sent to the node has its most significant bit cleared, to eliminate parity or similar problems. This is not ideal for those countries that use the extended character set. When this bit is set, and only when in TALK, data is passed as 8 bit data. Note this does not apply to an initial message sent on the same line as the TALK command.

***bit 6***

Bit 6 makes node handing case sensitive. Normally, node aliases are forced to upper case for searching in the table and for user *connect requests*. If this bit is set, these operations will become case sensitive. This could be very confusing for users unless they are aware of it and expect it. It allows node aliases to be entered as lower case, for example in setting the node alias and in forcing routes. Don't set this bit unless it is actually needed!.

**Hash node broadcast port control (mode /13)**

The node TNCs used as backbone ports are not visible and use # nodenames so users of the network will not have to sort through many useless nodenames to find the useful ones. The NODE command will, by default, only show the non-# nodes. Since the # nodes aren't used to navigate or to route packets they can be eliminated from the node broadcasts. In a node broadcast, the broadcasting station will always include itself even if it is a # node. This mode option does allow all other # nodes to be restricted from being broadcast. The sysop may specify what ports will broadcast other # nodes.

0 = broadcast # nodes on both ports
1 = don't broadcast # nodes on HDLC
2 = don't broadcast # nodes on RS-232
3 = don't broadcast # nodes on either port

**Extra aliases (mode /14)**

If this is set to 1, the node will listen for (and accept uplinks to) the aliases set in HOSTALIAS, DXCALIAS and BBSALIAS if they are set. If this parameter is set to 0, or if the respective aliases are not set, it will do nothing. If you do not use the aliases, set this option to 0 to avoid wasting processor time. Note: this option cannot be set in with the PATCH program.

### Auto reconnect to node (mode /15)

If a node to node to node connection has been made by a network user and the user is disconnected from the distant end the user may be reconnected to a intermediate node if this option is enabled.

Example. This node is FRED and has this option set to 1. The user connects into FRED and then connects on to JOE. The user is disconnected at JOE or does a BYE command. JOE sends a disconnect back to FRED. The user gets a message like:

FRED:K1QSY-1} Welcome Back

If FRED has this option set to 0 the user would get

*** DISCONNECTED

### Control of slime trails (mode /16)

If a node receives a packet traveling through the network from a unknown distant node, the node will create an entry in it's node table containing the information required to pass a response back to the distant node. If a new node (that has not propagated through the network) is used to connect through a network to a distant node, entries in all of the involved node TNC's node tables will be created back to the new node. This trail of temporary entries is called a slime trail. (the trail left by a slimy worm?). Slime trails are temporary and are asynchronously deleted by each node, sometimes stranding packets that have not been acknowledged. The are created with a lifetime of *broadcast interval* times *init-obsolescence*. They will expire when the obsolescence count for the entry reaches zero, stranding any outstanding packets. This can cause undue frustration on the part of real-time keyboard users and for this reason slime trails are considered an undesirable side effect of unbalanced networks.

The node entry for a slime trail looks like a node listing but with a callsign and no alias. Example:

WIRELN:KA2DEW-6} Nodes:

| | | |
|---|---|---|
| KA2DEW-4 | CROWD:KB2BLX-9 | KINNLN:WA2IKL-5 |
| KINZOO:WA2IKL-8 | LTLFRY:WA2IKL-1 | PTRSON:KA2DEW-3 |
| RKL:K2SK-1 | RKLAN:K2SK-2 | |

KA2DEW-4, as shown in the example, is a slime trail entry.

Each bit in this option controls a different function. Bit 0, if set, causes any stations without aliases to be *hidden* when a nodes command is given. Bit 1, if set, causes the node to refuse to make slime trail entries in the node table. Before you use this feature, be careful to make sure that you understand the implications of doing so, as without fixed entries the node will refuse to accept level 4 connections from a station until it has heard their node broadcast.

### Digipeat up/downlink control (mode /17)

This option controls whether a station can digipeat into and/or out of the node.

0 = digipeating is allowed
1 = no digipeating into the node
2 = no digipeating out of the node
3 = no digipeating into or out of the node.

Note: This option does not affect digipeating via the node.

### Size of destination node table (parm /1)

This parameter sets the number of hidden and visible nodes that may be stored in the nodes table for a TheNET TNC. This value sets the amount of TNC memory used to store the node information. Each entry takes memory space so setting this value much higher than you'll likely use is a waste of buffer space that could otherwise be used by the node to improve performance and increase the number of user connects available.

Ideally the number of nodes you'd want to store in the nodes table is somewhat less than 100 if only to keep the responses from the NODE command short. Even a 100 node long NODES response would be sent in seven 240 character long packets. Node broadcasts (passing routing information) are longer for more nodes listed. What is more important is all of the nodes in the node table are available for passing traffic even under the most loaded conditions. This is covered under *Initial Time To Live* (parm /8) and in other sections of the *Annual*.

(Range: 1 - 400) measured in # of nodes

### Minimum auto update quality (parm /2)

This parameter sets the minimum value for quality for a node to be saved into the routes table. This also may be called a filter value. This restricts the number of nodes saved into the nodes table based on their quality. Making this value higher reduces the number of nodes, making this value lower increases the number of nodes. When a node hears a nodes broadcast from a neighbor node it processes that broadcast in terms of the quality value associated with that neighbor node. Any nodes learned about whose resultant quality is less than *minimum quality* are ignored. If the quality to all backbone nodes is the same, regardless of path type or port number the length of the network can be predicted based on the path quality and *minimum quality*.

Since rounding occurs in the calculation of node qualities, there is a critical value for this parameter at which point the node quality is not reduced any further. If the quality used between TNCs is 255 (even for RS-232 value) the critical value for this parameter is 128 and propagation reaches that in 129 hops. That means if the value for this parameter is less than 128 and if qualities of 255 are used for node to node propagation a node would be propagated over a path of nodes that was infinitely long.

This also means if there were three nodes that were all neighbors (like in a node stack), any node heard by those three nodes will stay in the node tables forever, even if it was never heard from again. If the quality used between TNCs is 64 the critical value for this parameter is 0 and the quality is reduced to 0 in 6 hops.

(Range: 0 - 255) no units

**HDLC (radio port) default quality (parm /3)**

This parameter is the default quality for port 0 which is the HDLC or *radio* port. This number is used to set the quality of the path to a newly detected neighbor node heard over the radio port. Later the value given to that neighbor node is used to process received node quality information.

This number, used as a fraction over 256 (i.e. N÷256) would be applied to the qualities broadcast by a neighbor node. If a neighbor node FRED broadcasts information for a distant node GEORGE such that FRED knows of GEORGE at quality 120, and if this node (TOM) has a route quality of 128 to FRED then this node will store GEORGE at 120 x (128/256) = quality 60.

Usage: In a system where *time to live* (parm /8) is used to determine the number of backbone hops away from which this node will be available, route quality to all neighbor nodes should be the same on all highest quality backbone ports, both RS-232 and radio.

For backbone nodes used in a hidden transmitter free backbone or dedicated point to point backbone (i.e. a designed and agreed upon convention between all stations using the frequency in the local area), the *route* quality to each neighbor should be fixed/locked at a number such that the node listings will be propagated, and the value for parm 3, *HDLC default quality*, should be set to 0 so no non-sysop added routes appear. This makes the network ignore a violation of the convention (by a third party). Notice should be sent to all node sysops involved before changes are made so route lists may be updated.

If it is agreed a convention between participating node owners is necessary in order to establish a dedicated point to point link or hidden transmitter free link there is no reason not to take precautions against a breach of that convention. All backbone TheNET TNCs that operate as other than agreement supported hidden transmitter free links would have all HDLC port 0 route qualities set to a much lower value. This kind of radio port is also called a *gateway* port. In this case the user port at each node site that shares the frequency would show at the next site across the backbone/gateway but the nodes on either side would not be seen or propagated.

Setting this parameter to 0 does not affect nodes know via the RS-232 interface (matrix connect), nor route which are already existing or locked in.

A common error in using TheNET is to assume qualit values are effected by or affect the performance of con munications to and from a neighbor station, either vi port 1 or port 0. This is incorrect. Quality values ar used for node propagation control, i.e. how far throug the network a node will be listed, and to control th deterioration of a node listing's lifetime if the source ( the node is cut off. If a link goes down, quality value affect how long it will be before a node listing vanishe from the network. A shorter time is better as false nod listings promote incorrect routing, user frustration, an unnecessary network congestion.

Note: Changing the *HDLC or RS-232 path quality* wi not change existing route qualities. This will only af fect new routes. To change existing routes you can:

- manually use the R command in sysop mode;
- decrease broadcast interval for a short time so th current routes expire;
- disconnect the node from the radio so current route expire

(Range: 0 - 255) no units

**RS-232 (crosslink) default quality (parm /4)**

This parameter is the default quality for port 0 whic is the HDLC or *radio* port. This number is used to se the quality of the path to a newly detected neighbor nod heard over the radio port. Later the value given to tha neighbor node is used to process received node qualit information. See parm #3 for a complete description o quality and default path quality.

Note: Changing this value will not change existin route qualities. This will only affect new routes. T change existing routes you can:

- manually use the R command in sysop mode;
- decrease broadcast interval for a short time so th current routes expire;
- disconnect the node from the radio so current route expire

(Range: 0 - 255) no units

### Initial obsolescence count (parm /5)

This value defines how long the node table will keep information on another node. Every node listing includes an obsolescence count.

Each time a nodes broadcast is received, all nodes included in the broadcast are updated. The routing information for each node, as stored in memory, includes the obsolescence for each of three routes. The obsolescence count for the route to the broadcasting neighbor, for every node in the node broadcast, is updated to the obsolescence counter initialization value if the routing exists for the node in the table and to the broadcasting neighbor. If the routing does not exist for the node in the table, and if the quality value associated with the routing is higher than the lowest of the three listings, the lowest listing will be bumped, the new routing will be installed. That routing will then have an obsolescence count equal to the initialization value.

Each time this node makes a nodes broadcast the obsolescence values for all stored node listings are decremented by 1. Each node listing whose obsolescence value is decremented from 1 to 0 is removed from the routing table.

(Range: 0 - 255) no units

### Minimum obsolescence count for broadcast (parm /6)

This is a filter which determines how old node information can be and yet still be broadcast.

This sets the limit on the minimum obsolescence value associated with each node for it to be included in the nodes broadcast. The node doing the broadcasting is always included in the broadcast. This value used in concert with *Obsolescence counter initialization value* can be used to force a node to only broadcast itself by simply making this parm bigger than the initialization value. This is a desirous effect for ports facing nodes which don't participate in the *dedicated link* backbone system, on gateways for instance.

(Range: 1 - 255) no units

### Nodes broadcast interval (seconds) (parm /7)

A TheNET node TNC has an internal broadcast interval timer. This value sets that timer. When the timer runs out the node decrements the obsolescence counts for all of the nodes in it's nodes table and does a node broadcast. The nodes broadcast is a formatted list of all nodes in the routing table whose obsolescence counts are greater than the minimum to broadcast (see *Obsolescence counter, minimum for broadcast*)

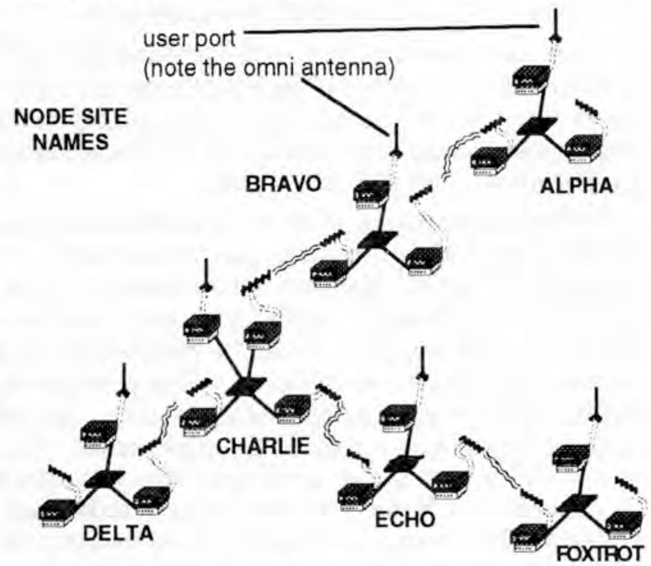
Whatever value is set, it should be compatible with that of the neighbor nodes. If this node broadcasts more frequently than the neighbor node it might forget about listings the neighbors tell it about. This is because the obsolescence counts may be decremented past 1 before the neighbors rebroadcast. The opposite is true if this node doesn't broadcast often enough. Incompatible node broadcast intervals may be overcome by increasing or decreasing the *Initial obsolescence count*.

(Range: 0 - 65535) in seconds

### Initial time to live (hops) (parm /8)

This is the number of hops between user ports that your message will travel. For instance, if you have a network that looks like this:

If you were to connect to the FOXTROT user port and then do a C ALPHA your message would travel 9 hops. If the *time to live* on FOXTROT, or on ALPHA, was less than 9 you would not get the connect. Eventually you'd get a *failure with ALPHA* which would be caused by a transport layer time-out.




This is the number of TNC hops a message from this node can go before it is killed. Each message transmitted through the network by a node has an associated time to live. Each time the message is received and retransmitted by any TheNET node TNC the time to live for that packet message is decremented. If the time to live reaches 0 the message is thrown out. This parameter sets the time to live start value for each message originated by this node.

If *time to live* is set too high and if a routing loop occurs much network time may be wasted. Additionally time to live protects the network from long distance abuse. If time to live is kept relatively small a poorly sysopped node will not cause much damage. For proper network function a network node must not propagate further than *time to live* # of TNC hops.

(Range: 0 - 255) measured in # of TNCs

### Transport FRACK timeout (seconds) (parm /9)

Sets the number of seconds your local user port will wait before retrying a packet across the network. In this time the destination user port must acknowledge the packet and that acknowledgment must make it back to the origin user port. If the packet is retried there will be a second redundant copy of the message heading across the network even if the first copy successfully arrived. This value must be set to the maximum amount of time it will take for a packet to travel the number of hops set by *time to live* (and *minimum quality* with both *default path qualities*) and the acknowledgment to return to the node of origin.

This parameter has to be reasonably small as this determines how long a user will wait when a backbone fails or a node is off the air. This number has to be large enough however, so even under the worst loading conditions a user will not get disconnected if the links do work. The value chosen for this parameter is dependent on the values chosen for node to node quality and minimum quality for auto update. Think of this value in terms of number of seconds allowed for each TNC traversed in a node to node connect. A value of 200 and a maximum number of TNC hops of 7 gives us about 7.25 seconds per TNC including the end to end acknowledge.

(Range: 5 - 600) measured in seconds.

### Transport retry counter (parm /10)

Sets the number of copies of a given packet the origin user port will send into the network to the destination user port, timed by *transport FRACK time-out*, before declaring the path disconnected. This value may be set to 1 which would create the least loading by an individual network user but will not allow trying alternate paths. Network developers should weigh the benefit of having alternate routes tried (consider where alternate routes exist) against the probability of excess network loading caused by retries due to existing network loading.

(Range: 1 - 127) measured in tries.

### Transport acknowledge delay (seconds) (parm /11)

This is the amount of time a port waits before acknowledging a transport layer packet that was received.

If there are two nodes passing information via L4 packets, back and forth, and not sharing data with any other nodes, setting this value to > 1 will allow acknowledgment frames to be piggy backed with information frames, thus saving transmission time. In the condition where most nodes pass L4 data with multiple different nodes at the same time, rather than with one single other node this feature would not be of value. If this value is set to it's minimum the performance for random node to node communications would be enhanced, at the expense of repetitive same node to same node communications.

(Range: 1 - 60) measured in seconds.

### Transport busy delay (seconds) (parm /12)

In the event a transport layer (L4) circuit cannot handle more data (*transport layer window size*) a busy flag is sent back to the originating node. This parameter is the number of seconds the origin node waits before retrying a message lost due to the busy condition. When the busy node clears it also generates a packet back to the origin node announcing it is clear.

(Range: 1 - 1000) measured in seconds.

### Transport window size (parm /13)

This is the number of unacknowledged packets that can be outstanding for a given circuit (each user connect). If this number is kept small the throughput on a very lightly loaded network will be worse than it might be with a large window size. If this number is large on a heavily loaded network stations with the most amount of traffic would hog the network.

This number should also be set in concert with the Transport FRACK Time-out. The Transport FRACK Time-out must be calculated so there is plenty of time for window-size # of packets to traverse the network, with acknowledgments coming back, and under heavy network loading, before the Transport FRACK Time-out occurs.

(Range: 1 - 127) measured in packets

### Transport overfill limit (frames) (parm /14)

This is the maximum number of packets buffered in a node TNC for a given circuit. This buffering is done only at a destination node. Those nodes passed through transparently are not affected by *Transport* values.

When a station sends text to a node the node will try to send the text off to the destination across the network. If the network gets slow or if the destination can't deliver the text due to problems on the other end the local node will get backed up. The number of packets that will get stacked at each end is the *transport overfill limit*. (also called *congestion control threshold*)

Setting this value higher makes for smoother operation for a smaller number of high volume users. Setting this value lower makes for smoother operation for a larger number of low volume users. Generally, with lower values, the low volume users are able to run smooth while the high volume users are caused to be delayed. An exception to this is when this value is set high at a node local to the high-volume sending station where the sending station will have completely sent a large message and will start a time-out timer. Since the message is buffered at the local node and must then filter through a network, if the network latency is high the high-volume sending station might decide the link had failed. If this value were lower, the sending station would not be able to unload it's message as fast and would not start it's time-out timer until more of the message had actually traversed the network.

(Range: 1 - 127) measured in packets.

**No-activity time-out (seconds) (parms /15)**

This is how long a station may stay connected to a node if no traffic flows via the station's circuit to or from the node.

When choosing a value for a node/port a user will be connecting to, keep in mind users will be connecting to CROWD nodes and DxClusters which may go for extended periods of time without activity.

(Range: 0 - 65535) measured in seconds.

**Persistence (n/256) (parm /16)**

This figure determines the aggressiveness of the TNC's transmit function. A high value of P-persistence will cause the TNC to be very aggressive. If there are more than 2 TNCs waiting to transmit on a single channel and their P-persistence is set incorrectly the data throughput will suffer. A formula used to calculate ideal P-persistence is Pp = (256/N-1)-1 where N is the number of TNCs on the channel that could have data to go out at the same time. So, if the channel only has two TNCs the P-persist could be set to 255. If the number of TNCs is 3 then P-persist could be set to 127.

(Range: 0 - 255) measured in fractions of time N/256.

**Slot time (parm /17)**

This is the amount of time the node will want the channel to be clear before attempting the P-persistence calculation again.

This value should equal the TXDelay for the node plus the worst response delay for other stations on the frequency. For dedicated point to point links (2 radios on a frequency) this value is unimportant as P-persistence when set to 255 overrides the value of slot time. As with P-persist, this value depends on the type of application. P-persist and Slot time work together to set up a random delay determining when the node will key up following a DCD decision that the channel is clear. This is an anti-collision technique. When the node is ready to transmit, a number between 0 and 255 is internally generated. If the random number is equal or less than the value set by P-persist, the node keys immediately upon sensing a clear channel. Otherwise the node waits for a period of time equal to the slot time and then internally generates a new number, etc. A value of 63(+1) is 25% of 255(+1) and thus sets the percentage of time the node will immediately keyup. Protected trunking nodes (those with only one transmitter on their receive frequency) would have faster throughput if there were no node keyup delay. Setting P-persist to a value of 255 will accomplish this.

(Range 0 - 127) measured in 10s of milliseconds.

**FRACK (T1) time (parm /18)**

This sets the time delay after a packet is sent to a user or neighbor node before the node will attempt a retry. For double ended hidden transmitter free backbones this should be set to a minimum value, 1. For *user* ports, setting this value higher gives priority to those users that are most consistent into the node. Higher values (8 - 12) might be used on crowded user ports (> 5 users at a time) to give users with more reliable signals higher priority. (users with retries can take up a lot of channel time otherwise)

(Range: 1 - 15) in seconds

**AX.25 window size (MAXframe) (parm /19)**

This parameter is the same as the MAXframe command available in most user TNCs.

This sets the number of packets, of those available in memory to send to a user or adjacent node, that will be sent in one transmission. This parameter affects how access to both backbones and user ports is distributed to the various users. It appears as this number is increased, the higher volume users will intermittently get larger shares of the network loading. Users who send well spaced packets or fewer packets gain no benefit through having this value set higher. [*More research is definitely needed into how this parameter affects the performance of a network*]

(Range: 1 - 7) in # of packets

**AX.25 retries (parm /20)**

This value sets the number of times a TheNET node will retry on the radio port. This value is the same as for a regular user TNC. TheNET nodes default to using AX.25 lvl 2. If a node to node link retries out the specific destination node (transport destination that is) is marked as bad for the one specific neighbor. This takes place in the nodes table. See *Theory of Operation: Routing*. Note no notice of a link failure is sent back to either origin or destination nodes in the case of a retry time-out on a backbone link. The only method of failure in this case is that of *transport layer time-out* as defined in one of these parameters.

(Range: 0 - 127) in # of tries.

**Ack (T2) time (link layer RESPTIME) (parm /21)**

10s of milliseconds between receiving a packet from a neighbor node or user before the node will acknowledge a packet. This is actually the *response* time in Ms. Setting this value too low on a user port will prevent some users from being able to access the port as older radios and some newer rigs with very slow locking synthesizers will not recover from transmit fast enough. Making this value larger makes the node less aggressive which might be useful on a crowded frequency.

*This value is dependent on the radios this TNC is talking to. Check this value if problems occur.*

(Range: 0 - 6000) measured in 10s of milliseconds.

**Active check (T3) time (parm /22)**

This parameter sets an idle link timer. If a link is inactive for this amount of time a check packet is sent to make sure the other end is still there. This value is set rather large as making it smaller will start to waste time. Making this value 0 will also work but if link no-activity time-out is set to a large number a user could turn off his station and still be listed as a user on a node for the entire duration of the no-activity time-out.

(Range: 0 - 65535)

**Digipeat (parm /23)**

This option enables or disables digipeating via the node. If this option is on (set to 1), stations could use the nodename or callsign as a digipeater path. In this case a station may also digipeat through the node to another node on the diode matrix. In addition a station could digipeat through two nodes on the diode matrix (if both had digipeating turned on).

If this option is turned off (set to 0) then digipeating attempts via the node will be ignored.

**Callsign Validation (parms /24)**

This option allows the sysop to enable or disable callsign validation. This affects the C command in the nodes command processor. This also affects the valid callsigns that may be used to connect *to* a node. If this feature is enabled a user would not be able to connect to the node with the callsign of N0CALL.

If validate callsigns is turned on (set to 0) the C command in the node will only accept valid callsigns and existing node names. This feature should be enabled on most all nodes as this eliminates the delay a user might have to wait to find out that he/she specified a node name incorrectly. If your gateway port sees KA-nodes this feature will need to be disabled so that users can connect from this port to a node name that won't be listed in the TheNET node.

0=allow all callsigns;

1=only text strings with an embedded number.

**Beacon mode control (parm /25)**

Configures automatic identification of the node. Legally all amateur stations need to identify with an authorized callsign. That means if someone connects to a node using it's nodename it must still identify with it's callsign. Since point to point backbone ports are always connected to using networking protocol and not AX.25 we know they will always communicate using their callsigns. Thus turning on an additional identification is unnecessary. On the other hand a user port must be identified since it allows users to connect using the nodename. By setting this parameter to conditional the node will only ID when it is in use. By definition a wide coverage user port shares the channel with other servers and nodes. Having the ID beaconed every 10 minutes would be greedy of channel time. On the other hand a Low coverage user port doesn't share the channel and can arbitrate for itself when to make it's 10 minute ID transmission.

0 = do not send beacons;

1 = if active;

2 = send beacon packets regularly

**CQ broadcasts (parm /26)**

This parameter disables or enables the CQ broadcasting feature. (see User Command List). The CQing user still shows up on the USERS list but the CQ unproto message is not sent.

0=no CQ message,

1=CQ message enabled

# Theory of Operation

TheNET is a software package that runs in a TNC2. The purpose of the software is to control a TNC in a system of TNCs called a network. The network is usable by hams running any mode of amateur radio packet based on AX.25 protocol. The hams can utilize the network to chat in real time, access remote computers, pass traffic or perform paging and remote control. Follows is a technical description of the TheNET software.

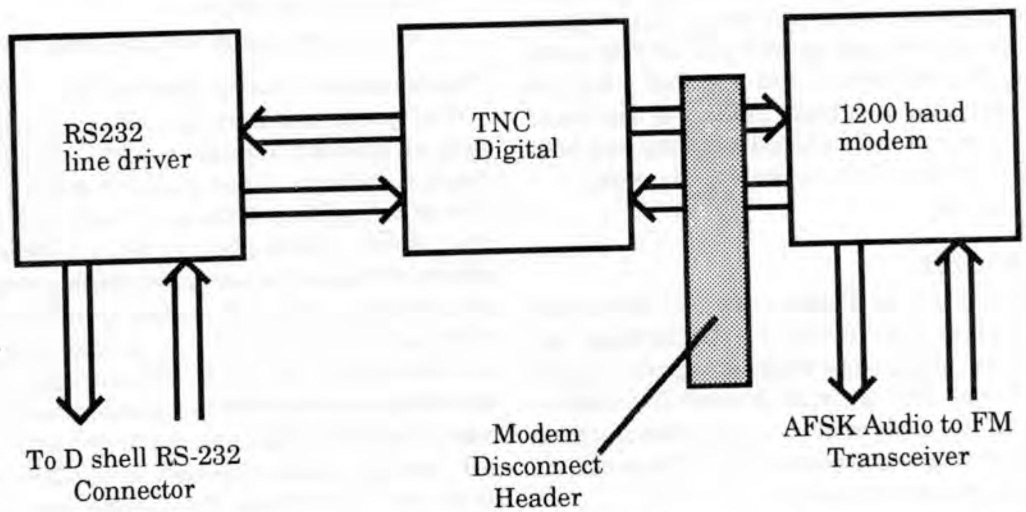
### Hardware - L1

The TNC2 is a dual port device. That is, it has two serial i/o channels. One of these i/o channels is hooked to an RS-232 driver and a D shell connector. The other i/o channel runs to the *modem disconnect header* and then to a 1200 baud modem.

*L1:I/O control and hardware;*
*L2:AX.25 linking;*
*L3:network routing;*
*L4:transport processor; and*
*L7:command processing.*

### Link Controller - L2

The TheNET node's link controller will accept and make AX.25 connects on either the modem or RS-232 ports. If a station connects to the TheNET node on either port the node will remember a connection is made, the callsign of the connecting station, and the callsign used to connect to the node. These are saved as the address field. The node can accept the connect using the preset callsign and ssid in the EPROM or using the nodename




with any of 16 SSIDs. Connects may be accepted by the node from the same callsign on all 17 *callsign - nodename - ssid* combinations at the same time. The next time a packet is received that matches that address field the node will classify the connecting station as either a user or as another node.

If the connection is a user the user is added to the users list and any further communications is passed to the command processor. The node will determine that the connection is a user the first time the connecting station does something a node wouldn't do. Any text command, for instance. The user may interrogate the node for information it has (see user commands) or may command the node using the sysop commands or using CONNECT or CQ. If the user uses the CONNECT command he may establish a connection to another node or to a user from this node. This is covered later under *"Circuit"*.

The software that runs in the TNC2 is installed in a 64K EPROM and is mostly compiled C code. Some small sections are; written in assembly language. Also stored in the EPROM are default parameters and text strings. Generally the text strings are not user programmable but a bit hacker could find and change them. Text strings that exist include "`Connected to`" message, command names, "USERS" response string, beacon text and error messages. The default parameters, callsign, node name and password are programmable. Most installers of TheNET X-1J will be using the PATCH.EXE program with a PC.

### International Systems Organization (ISO)

The function of a TheNET node is to act as an active data *store and forward* device as well as a remote *command interpreter* for users. Communications can occur both on the modem port and on the RS-232 port at the same time. Communications is AX.25 with networking and routing operations which are written within the bounds of the ISO 7 level model. That is, the processes in the TheNET software are modularized in the following functions:

### Routing Processor - L3

If the connection is another node the next message that follows will contain TheNETese. TheNETese is a slang term that means the communications has non printable characters TheNETs understand. More on that will be covered under *Protocol* [*not yet written, sorry-ka2dew*]. If the node's link controller gets the TheNETese it marks the station as a neighbor TheNET node and passes the connection information up to the routing controller. If the traffic received is destined to another node the routing processor passes it back to the link controller to go out to the next neighbor node in the chain. A neighbor node is a node this node can talk to directly, either over the RS-232 port or over the modem port.

### Transport Processor - L4

If a neighbor node passes traffic to me (I'm a TheNET TNC) destined for me my routing processor passes the message to my transport processor. My transport processor is responsible for making sure all data that originates here or is destined to me makes its way across the entire path between circuited nodes. So, if I connect from here to another node many sites away it is my transport processor that is responsible for seeing the message gets there.

The transport processor gets messages from the network processor and from the command processor. The command processor is hooked to the user. Users can connect to a node and tell it's command processor to connect to another node. Users can tell the command processor to connect to another user or server station.

### Command Processor - L7

The command processor may be instructed with ASCII text commands from a user station. Much of the remainder of this manual deals with command processor functionality. The important functions needed for understanding of the remainder of *"Theory of Operations"* is the command processor allows the user to connect to other nodes via the network over either the modem port or the RS-232 port and to stations that are not nodes over the modem port. In addition the user can request lists of:

*all nodes in the routing table;*
*neighbor nodes;*
*the best three neighbor nodes for a particular node;*
*all L2 connected stations known to be users.*

### Program Start-up

The program starts up when power is applied. It lights the STA and CON LEDs for a second and then turns them off. It initializes its memory, copying default parameters unless it has what it thinks are valid parameters and INFO message in RAM already. Then it sends a beacon message on both the modem and RS-232 ports. The node broadcast timer is started.

### Routing

When the routing processor gets a packet to send out it looks at the destination address provided by the transport processor. The destination address is the callsign of the requested destination node. The routing processor looks up the node in it's node database (routing table). It will find up to three neighbor node callsigns which are in the direction of the destination. These neighbor nodes are routes to the requested node. If the requested node is unknown the message is thrown out. The information supplied with each route is the callsign of the neighbor node, the port number of the route, the quality value associated with the path and a flag indicating a route is already in use. If no flag is set the router selects the highest quality route and sets its flag. The port number describes whether it's an over the radio shot or an over the RS-232 shot. This information is passed to the link manager.

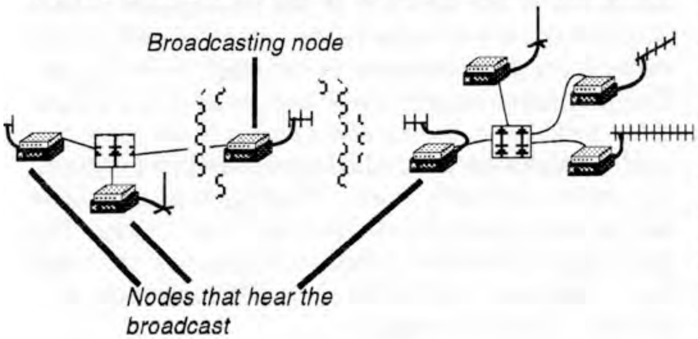
#### *Slime Trails*

The router knows of up to 100 nodes (adjustable) and knows of up to 3 routes per node. If a message, whose origin node is not in the routing table, is passed to the router, the router notes what neighbor node and port sourced the message and installs the origin node and route into the routing table. This way when an answer to that message comes back through the node the node will know what to do with it. This function is called slime trailing and only happens in the event that the origin node knows of destinations within the network, and where the network doesn't know of the origin node. *Important: If the routing table already has 100 nodes in it a slime trail cannot be added.*

The reason this function is called a slime trail is that when a user requests a copy of the routing table (nodes list) the slime trail shows up on the nodes list with just the callsign. If the user traces down the origin node by using the N command with the callsign as the parameter the user will step through nodes until he reaches the origin. At each step there will be a node listed with just the callsign. At each node along the route the slime trail route will time out randomly based on internal TNC timers.

### *Nodes Broadcasts*

Every 15 minutes (parm adjustable) the node will send a nodes broadcast via both it's RS-232 and modem ports. This broadcast allows the neighbor nodes to maintain their databases of nodes this node is sourcing. The broadcast consists of all of the nodes whose obsolescence counts are equal or greater than the "Minimum obsolescence count for broadcast" parameter, all of the nodes whose obsolescence counts are 0 (locked nodes), and only those nodes whose quality is greater than minimum quality to broadcast. The format and order of the nodes broadcast is basically:




This node, PQ=256
knownnode, PQ=node-qual
knownnode, PQ=node-qual
knownnode, PQ=node-qual
...

node-qual is the highest of the 3 values the node has stored for knownnode.

When a node receives the nodes broadcast it goes through the following program sequence:

Check if route to the neighbor we are hearing is in place. No? Then use parameter 2 or 3 depending on if this is RS-232 or HDLC and put the neighbor node in the route list.
Apply the route quality for the neighbor we are hearing, to the received nodes broadcast.
As each node is clocked in store it in the routing table under the node's callsign with the route indicating the neighbor we are hearing.
As each route entry is being stored in the routing table, set the obsolescence counter to the initialization value in the parameters. If the route to the node we're storing is locked in, then ignore the incoming information.

### *Circuits*

The process of connecting into a network node, across the network, and out of the network requires three operations. The first is called an uplink. The second is called a circuit and the third is called a downlink. The uplink and downlink stages are AX.25 connections to the link processor of each of the two end nodes. The link processors pipe directly to the command processors. Circuiting means that the command processors pass the traffic to the transport processor which then passes the traffic to the router etc..

After the router in the origin node gets traffic to go to some distant destination node the traffic can hop from node to node over many TNCs before again reaching a transport processor which will be at the destination node. The destination transport processor will then acknowledge the packet and the process is repeated in reverse. At the same time the destination transport processor acknowledges the message it sends the message to the destination command processor. If the origin transport processor hasn't gotten an acknowledgment before a timeout timer expires (transport time-out) it can resend the message. If the origin transport processor gets another message from the origin command processor it can send that one into the system as well. It can have up to 2 messages in transit without having acknowledgments. When the command processor attempts to send the 3rd message the transport processor will respond with a 'choke' flag. The origin node will wait until one of the messages gets acknowledged and then sends the next. This is called a 'sliding window' (the window slides across the list of packets in the entire message).

# Setting up a TheNET X-1J TNC

This section covers making EPROMs, modifying a TNC to run TheNET X-1J, and testing the TNC.

***What you'll need***

In order to burn EPROMs you'll need the TheNET X-1J software, a PC and an EPROM programmer. To test the EPROMs you'll need a terminal emulator program or a dumb terminal, the cable hooking up the TNC to the computer/terminal, and power for the TNC.

EPROMs cost about $6.00 each from mail order places. The device you'll want to get is a 27C512-20 32Kx8 EPROM. That means it has a capacity of 512K bits (64Kbytes) and a 200 nanosecond access time and is CMOS. You can also use any of the 27512 series that is at least as fast as 200ms. JDR Microdevices @ 800-538-5000 sells the 27512-200 for $5.95. The 27C512-200 is $6.95. The difference is a *minor* amount of current consumption and RF noise emission. Surplus EPROMs have been seen at flea markets for as low as 50¢ in the north east. They've been seen for free at other places. The reason that they get so cheap is that some companies have rules that EPROMs can only be used once. After that they get scrapped.

EPROMs can be reprogrammed hundreds of times, supposedly. The very fact that some companies throw them out after one burning is a dead giveaway that something is afoot. Read on. An EPROM is a memory device that is programmed with a special tool called a PROM burner or EPROM programmer. The appropriate one for our purposes can be found from lots of mail order houses. JDR Microdevices @ 800-538-5000 sells one for PC compatibles that comes as two pieces, a burner and a plug in card. I like this one because it uses a high quality DB25 to DB25 cable between the two pieces, instead of a ribbon cable, programs bigger EPROMs which may be useful down the road, and I know that it works on a higher speed computer, if you need. Mine is running on a 33Mhz 80386 and it's compatible with 8088 PCs as well. It costs $149.90 and is ordered as a MOD-MEP and a MOD-MAC. Another good model is sold by Intronics @ (913)-422-2094. It costs $129.95 and plugs into the parallel printer port of a PC compatible computer. The thing I like about this one, besides the low price, is that it can be transfered to another computer in seconds.

You'll also need a UV eraser (ultraviolet light). This is a device that exposes the EPROM's little erase window to UV light for a timed interval. The cheaper erasers require the user to time them. The erase interval is usually not critical. A good eraser will take from 20 minutes to an hour to erase EPROMs. JDR, Intronics and Digikey (1-800-DIGIKEY) sell one for $39.95 which is a tiny four EPROM eraser called DATARASE II. I prefer the $68.50 Logical Devices model #QUV-T8/N from Active Electronics @ 206-881-8191 which erases 15 or so at a time.

TheNET software is available from land-line BBSs. COYNET BBS @ 914-485-3393 in NY, SALT BBS @ 508-385-3427, AA6ED BBS @ 206-271-4657, Vectorboard BBS @ &16-544-1863 or 716-544-2645, Highland BBS @ 716-761-6460 and Compuserve are likely places. It comes in a ZIP compressed or self-extracting PC DOS compatible file. The compression program is also available on the land line BBS. You'll need a phone modem to receive this data. The other way is to get it in floppy or EPROM format from somebody else who has it. If all else fails send a message to one of the officers of NEDA or ask on your local BBS. Since I have my JDR catalog out I'll mention that they have a PC compatible 2400 baud modem for $49.

***Before you start***

Hook up your TNC to your computer or terminal and to power. Test the TNC. Make sure it does all that it is supposed to do before you start modifying it.

***Working with the files***

Once you uncompress the software you'll have several executable programs, a plethora of .DOC and .TXT files, and a couple of files called THENET1.X1J and THENET2.X1J. The two THENET files are the upper and lower half of the TheNET software that will eventually go into the EPROM. These are *binary files*. That means that, like executables, they are programs and not readable text files. The THENET files are burned into EPROMs and used to run the TheNET program in the TNCs. The process by which you'll program the EPROMs will create a custom pair of binary files for each node TNC you are going to set up.

Make a copy of each of the two THENET files. Name them with a nodename you intend to use. If your node is named FRED, create a file called FRED.1 and FRED.2 by using the COPY command on THENET1.X1J and THENET2.X1J respectively. Now type

```
PATCH FRED.1 FRED.2
```

The PATCH editor has help using the F1, F2 and F3 keys. Go to the level 2 parameter menu and change the callsign of the node to the one you'll be using for your FRED node. Go back to the main menu and then to the Level 7 Parameters and then select the Password menu. Use the down arrow and a return to go into the password set screen. Type in your password. I recommend using all capitals with a single space between correctly spelled words or proper common names (You may have to read this password over the phone to somebody someday). I suggest you keep it to six or so words or less than

40 characters. Hit ESCAPE twice to save and exit. Now go to the Information message menu and "Set a new message". Do this exactly:

*Hit a return*

*Type the town, state that the node will be in followed by a return.*

*Type "Sponsored by yourcall @ bbscall" and do not hit a return. Now hit Escape twice to save and exit this menu.*

The reason you have done this is that the first line will be the line following the callsign:nodename). It looks tacky if you have something following that so don't. The reason you didn't put any other information, like frequency, purpose of port, etc.. is that you can put all of that in the sysop programmed portion of the INFO message. The EPROM part can't be changed by remote control and will require reburning the EPROM. Also, now that you have the info text set the way you want it, you don't have to change it again. (feel free to do it the way you want to but I recommend you don't get too complicated or specific). You can put up to 80 characters in the internal space and 160 in the sysop programmed portion.

Now you are done with all of the non-sysop changeable things. You may want to go and program the node name in the level 3 menu but the most important thing is to get you up and running with your first TheNET chip as soon as possible so you can get the process of modifying TNC's over with.

To exit out of the program and save your changes use Escape to get back to the main menu (if you're not already there) and then use H to save changes to image on disk. It'll save into the image you started with. Next hit escape to exit the program.

Now burn the EPROM. The way I do it is to run the EPROM program and then load the FRED.1 image in starting at EPROM location 0. Then burn the prom. Now load the FRED 2 image in starting at EPROM location 8000h. Burn the prom a second time. If you need INTEL or MOTOROLA format data for your eprom programmer (your EPROM burner manual will tell you that) you can use the utilities that came on the X-1J disk. See the file MOTOROLA.DOC or INTEL.DOC for more.

Now we'll modify your TNC to run TheNET X-1J.

### Modifying the TNC

The TNC you have chosen must be a TNC2 compatible. There are modifications necessary for the MFJ 1270C that are not covered here. You'll have to check elsewhere (there is an article in the NEDA Quarterly published in April of 1994 on the subject).

If you are using a MFJ 1270B, PacComm Tiny 2 or Tiny 2 Mark 2, DRSI DPK-2 or DPK-9600 then you are all set.

Locate a static free and electrically non-conductive work surface. Open the TNC and remove the PC board.

### *Top of the SIO/paging jumper mod*

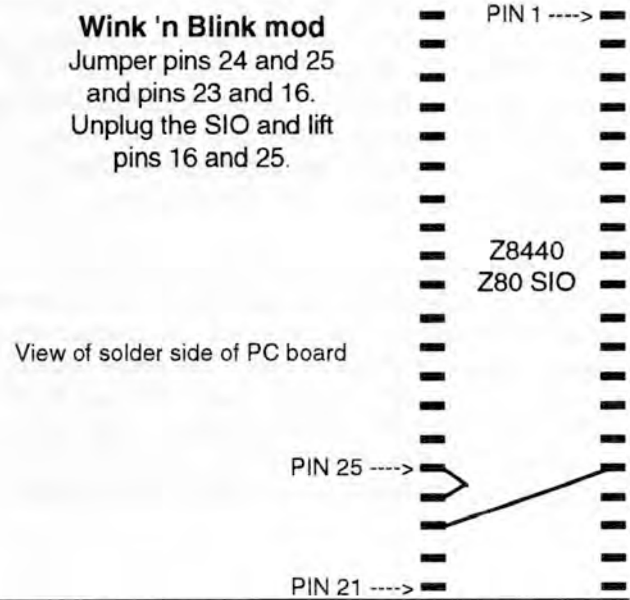
Unplug any existing EPROM. Identify and unplug the Z-80SIO chip. There should be both a Z-80CPU and a Z-80SIO. They're the two 40 pin chips. Starting from the end of the chip with the notch in it count 16 pins down the left and right side of the SIO chip. Carefully bend pin 16 and pin 25 slightly out of true. The idea is to not damage the pins but make it so they don't go into the socket when the chip is re-inserted. If you over-bend the pin you will break it off and that would be bad. (Radio Shack doesn't carry them although JameCo, JDR and others do) Carefully plug the chip back into the socket, watching those two pins. Now solder a 4" or so wire to pin 16 of the SIO. This is the *paging wire*, used by the X-1J program to control A15 (address 15) on the 27512 EPROM.

### *Installing the EPROM*

Carefully bend pin 1 slightly up. Plug the EPROM into the socket. Pin 1 should just clear the edge of the socket. Solder the paging wire to pin 1 of the EPROM.

### *First power-up test*

If this TNC has been used for TheNET before, remove the RAM battery jumper and wait a couple of minutes. Check the RS-232 baud rate on the TNC and on your terminal. Set them both to 9600 baud. Plug in the RS-232 cable to your terminal/computer. Plug in power and turn it on. You should get a short message on the screen telling you the software version and the callsign of the TNC. If you do, hit the ESC button and a C followed by a return. You should get a Connected message. If you don't, or if you didn't get the startup message, debug the situation. Check if your terminal is working by unplugging the RS-232 from the TNC and then shorting pins 2 and 3 of your cable. You should get local echo. Check the baud rates again. Make sure you're using the right COM port on your terminal or computer.



***Now that you have the TheNET TNC working***

Type I <return> and verify that your info is set correctly. Check the callsign. It should be correct when it comes back with the info message.

***Wink-N-Blink mod***

The wink-n-blink mod is necessary to verify proper operation of the diode matrix and of the TNCs. Unplug power and RS-232 from the board. Check which chip is the SIO and verify it's orientation (where is the pin 1 end?). Now flip the board over. Solder a jumper from pin 16 to pin 23 of the SIO and another from pin 24 to pin 25 of the SIO.

That's all there is to it. Flip the board back over and power it up again. You should see the CON light come on with the PWR light. After about one second the CON light goes off.

The CON light now indicates when your TNC is sending a TheNET crosslink PTT (over the RS-232 port). The STA light indicates when the TNC is receiving a TheNET crosslink DCD. This is important once you get your matrix together.

You now have one working TheNET TNC. We haven't tested the RS-232 PTT/DCD yet but that will have to wait until you get a second TNC set up and get the matrix together. For your next chip, try to get more of the defaults set up the way you want them. NEDA has published a list of 'standard' parameters elsewhere in this document.

After spending some time experimenting with the TheNET sysop commands you should go to the section "*Building Nodes*". Note that since you are plugging directly into the RS-232 connector (Host Port mode) that you don't need to do SYSOP to take control of your TheNET TNC. The following commands are available via the Host Port only.

Esc C  Connect to the Host port.
Esc D  Disconnect
Esc P  change password (until next time the RAM is cleared or a cold reset occurs)

# ROM Burning

These hams have volunteered to burn EPROMs f networking purposes. Give them a call or send a lette If you have any information for this list please send packet to NEDA @ WB2QBQ with Attn: editor in th title field.

Maine - John, NS1Z @ NS1Z.ME volunteers to bur TheNET EPROMs for those that can supply nodenam callsign, password and EPROM. John has a supply EPROMs and will accept replacement (mail order hou of your choice if you like) or EPROMs in advance. 20 364-2246

New York - Paul, N2LSS volunteers to bur EPROMs. If you want ROSE you should supply bina images and chips. If you want TheNET you can supp binary images or can get in touch with Paul and tell hi what callsign, info, password, nodename, parameter you want and what version of software. He ha TheNET Plus and X1. Paul wants EPROMs or fun (mail order catalog of your choice) in return for burne EPROMs but he has stock to get things going in hurry. Contact Paul via N2LSS@N2LSS.NY or 60 965-8327.

New Jersey - Don, N2IRZ will supply the EPRO and burn it with information of your choice. He has X J, TNplus2.11, ROSE 3.4 (and will keep on top of ne versions). Supply callsigns, node names, addresse passwords as appropriate. Include floppy data if po sible or Don will figure out what you need. Send SASE to Don for an info/configuration sheet if you nee to. Don asks for $7/EPROM and he'll supply the part mailers etc. or if you can supply postage, maile EPROM and binary image, Don will send it back as voluntary (free) service. Don Rotolo Box 219 Montva NJ 07645.

Washington (state) - John, KA7TTY will bur EPROMs for TheNET X1-J. Send EPROMs and co figuration information to John MacDuff 620 South Ea Bush St. Issaquah WA 98027.

Quebec - Burt, VE2BMQ will burn EPROMs for TheNE 2.08, 2.10 or X1-J. Contact Burt via
Burton Lang 1153 Hwy 203 Howick PQ, J0S 1G0 or
VE2BMQ @ VE2FKB.#MTL.PQ.CAN.NA

*Anyone interested in a copy of the program drop me a message at* G8KBB@GB7MXM.#36.GBR.EU
*Also, any suggestions for change gratefully received.*

*Dave G8KBB*
*7 Rowanhayes Close*
*Ipswich IP2 9SX*
*England*

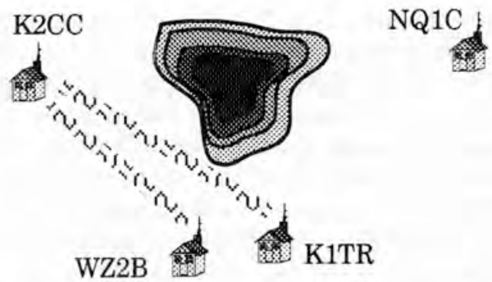
# Networking Around HTS

AX.25 is a Carrier Sense Multiple Access system. That's what CSMA stands for. CSMA means that each station depends on it's own receiver to determine when it's OK to go into transmit mode. In many commercial packet systems which use CSMA type protocols it is given that all of the transceivers can hear each other.

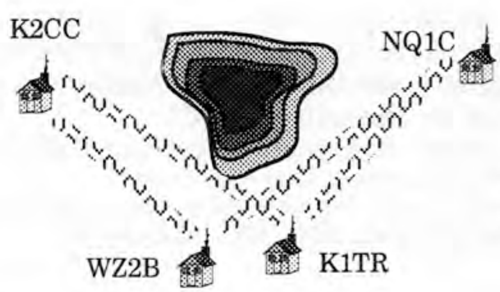
In amateur radio packet that is almost never the case. What that means is that sometimes a station can be talking and another will just come onto the frequency and start talking. It sort of sounds like twenty meters on a winter Sunday evening. Usually on twenty meters when that happens, an operator who can hear both stations that were transmitting will say something like "The frequency is already in use".

In AX.25 packet what does happen is that the two stations who were talking don't get any answer so they try again. Often times the timing can work out so that both stations will once again transmit at the same time (collide) and will waste even more time. What is worse is that in many areas on two meters there will be more than two stations trying to transmit at a time. There may be dozens. This means that the number of bad transmissions per successful exchange can be very large. Each time there is a bad transmission the stations have to wait a certain amount of time before retrying also. This wastes time.

Most amateur radio packet on two meters is done at 1200 bauds. This means about 150 characters per second. If there are only two stations in the local area and they can only hear each other, and they have reasonably fast radios the number of bytes that they can transfer per second is around 80. In a situation with two stations who can't hear each other, trying to pass data to another two stations who hear both equally well, the rate will probably be closer to 5 bytes per second per station. 80 bytes per second is pretty fast for a person to read. 5 is very slow. If the two stations could hear each other the rate might be up to 36 bytes per second for two stations. That's assuming that they are not both 'greedy'. If they are both 'greedy' it's possible that no data would be passed at all! The process by which AX.25 (CSMA) stations jam each other because they can't hear each other is called Hidden Transmitter Syndrome or HTS.




*In this illustration we have K2CC talking to WZ2B. K1TR is listening but is not involved in the conversation. The throughput between K2CC and WZ2B is about 80 characters per second.*




*In this illustration we have K2CC talking to WZ2B. NQ1C is talking to K1TR. Because K2CC and NQ1C can't hear each other they frequently go into transmit mode at the same time. K1TR and WZ2B both get garbage. Throughput is drastically reduced. NQ1C and K2CC are called Hidden Transmitters because they can't hear all of the transmitters that other stations they talk to can.*

## Network Implications of HTS

100% of theoretical throughput could only be arranged if there was some overriding control that made sure there were never any collisions and that the transmitters came on just at the right times. This is not possible if the only means of synchronization is via the CSMA channel and random timers. If all things are arranged as best they can be the best performance that can be achieved with a hidden transmitter environment is 20% of theoretical throughput. Two notes. 1. Only if all participants on the channel are cooperating will this occur and; 2. Only if the participants don't try to push the throughput on the channel. If the throughput exceeds a threshold (determined by the aggressiveness of the users) then collisions and retries on the channel will force the throughput to near 0%. In order to prevent the throughput collapse, means of backoff must be implemented. Because backoff is not a feature incorporated into most AX.25 based devices the only way to prevent throughput collapse is to avoid having hidden transmitters that source data.

## Workable Methods of Avoiding HTS

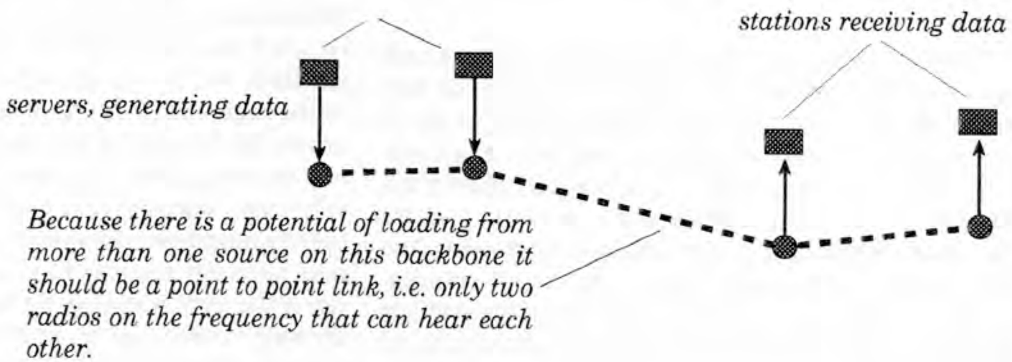
For backbones the best method of avoiding HTS is to have no hidden transmitters. If there will ever be a situation where two stations were sourcing a data into a particular link, even from further down the network, then that link should be a double ended point to point link.

For network access points the same rules apply. If there will be more than one source of data cranking out at full speed then there should be dedicated links to each source of data. It is important to note the difference between data generators and data receivers. A data receiver isn't a potential part of HTS. You can create a situation where there will be a dozen data receivers, hungrily accepting data and only generating the briefest of acknowledgment transmissions, on the same frequency. If, however, there is even one hidden source of data on the frequency, the data receivers will not be able to get even the short acknowledgment through.

Most packet users spend most of their time receivi data. This is great because it fits the model of *data ceivers only* just fine. Even when the average pack user is in data generate mode the frequency of pack data transmission is rather low. Usually the user is ha typing messages. This fits well within our 20% or le loading theory.

### AX.25 in the Network

So what we have decided is there are a few differe ways TNCs and CSMA are used in the network. The are point to point backbones, low usage hidden tran mitter free backbones, hidden transmitter free input por for data generators, and user channels for data recei ers only.




# X-1J TCP/IP Router Information

The IP router co-exists in the node with the other software. It is connected to the L2 and L3(Net/Rom) protocol machines, and is managed from the L7 switch. It will accept data from L2 Datagrams, L2 Virtual Circuits or NOS protocol extended Net/Rom frames. It will output to these 3 depending on the setting of the IProute and ARP tables.

The router supports the IP option of NOS and also does IP fragmentation. Level 2 segmentation is not supported. In addition, ICMP is implemented in so far as it is needed to respond to errors or PINGs.

No higher layer support is provided, i.e. TCP is not implemented, ip_send() and ip_receive() are only implemented in so far as they are needed for ICMP. You can therefore PING it but anything else will solicit an ICMP error message.

It will respond to ARP & REV_ARP requests but will never initiate them. The default MTU is 256 for AX.25 and 236 for Net/Rom. It will accept longer datagrams than this and fragment the output but it is not recommended as it merely wastes RAM. The MTU command may be used to change this.

It is possible to be creative in the use of L2 datagra and virtual circuits by use of the port default settin and the ARP table. The algorithm used is :

When a frame is to be sent, the ARP table is scanne for the appropriate entry. The entry tells it what cal sign to use. For Net/Rom encapsulation, it is send t the Net/Rom protocol handler. For AX.25 encapsul tion the following applies. The ARP table may indica DG or VC. In this case, that mode is taken. If there no DG or VC entry, the TOS bits are examined. If th delay bit is set, a datagram mode is selected. If not, an the reliability bit is set a virtual circuit is selected. neither bit is set, the default mode for that port is use to select a mode (see IPstats command, first parameter

Port addressing is not supported at the moment.

The IP router is manually controlled - no rspf or rip or even ARP requests. This is because 32K of RAM doe not allow such niceties as queuing frames whilst wai ing for address resolution.

# Node Sites and Hardware

The best place to put a node is where it is most convenient. If it's easy to build a node it is more likely to get built. If a node needs to be constructed to serve a particular link and only one site will do, then built it for that site. There is a lot to be said however, for having nodes situated where they can be observed, serviced, and played with.

## Higher is not better

There may be several reasons for putting up a node. One of those reasons is to allow a group of packeteers to access services or other stations that are available through a network of nodes. Perhaps you already have a network to link into, or perhaps you have hopes of building one. At any rate, user access to the network is important as the users are the best candidates for creating new services and new network installations in the future.

The best user access to a network would be where each user has a dedicated point to point link from the network node to the user station. This is extravagant to say the least. A good compromise would be a low coverage user port that serves only a small number of users. The limitations and design goals for a user port are that there should be no more than twenty stations (ten may be a more reasonable figure) on the air simultaneously accessing the user port and none of the stations should be major sources of data. Very infrequently should the users of your user port generate data faster than typing speed. Most users spend most of their on air type receiving data from BBSs, DxClusters, CROWD nodes or databases so this isn't much of a limitation.

Because your user port can't be allowed to hear any other node sites (or servers) your coverage is going to have to be strictly controlled. If your node site is on top of a high mountain or tower this may be difficult. Use of directional arrays or low gain antennas may be required. Perhaps an attenuator or tight squelch could solve the problem. Keep in mind that your node's user port shouldn't be allowed to be heard by the other nodes on the frequency either. Sometimes a node site that is designed to serve a distant city or long river valley can take advantage of tight patterned yagis. Beware of reflections off of mountains. If you aim a beam away from the mountain you reduce the reflections. Keep an open mind and don't use high power. Remember that amateur radio spectrum is a limited resource. Use it wisely.

Node sites in homes have particular advantages. Whenever a ham is involved certain characteristics may be assumed. One is that if three radios is fun there is bound to be five or six in the near future. Putting a three port node in a ham house situation is a good way to make sure that more expansion occurs. These things are darn fun to watch. It is also particularly easy to add local computer access to the node with minimum expense. This means that a server can be added to the network very easily.

Nodes in commercial sites have advantages as well. One of these is that backbone paths can usually be quite long. Often commercial sites offer fairly high towers so separation between antennas on the same band may be achieved. It is quite possible to run as many as a half dozen backbone links in the UHF ham band at a single site. The way this is done is by running the links in half duplex mode. The receive and transmit frequencies may be split by as much as fifteen or twenty megahertz. Then the links can be set up so that all of the receivers are in the high end of the band and all of the transmitters are in the low end of the band. So long as the antennas are reasonably separated vertically this should be very easy. Because your radio's transmitters are about twenty megahertz away from the commercial band this may be easily approved by the site managers. This is one of the more wild ideas for node to node linking. Using 25 watt commercial or amateur mobile radios on simplex you should be able to get two or three UHF links at the same commercial site, given that you can get the antennas and coax runs approved.

One problem associated with commercial sites in some metropolitan areas is the coverage for the user port may be higher than desired. The easy solution to this problem is to not have a user port at the high node side. Perhaps one of the pre-existing servers would house a user port. Perhaps you can set up several by using dedicated links to each of the local servers in the metropolitan area and maybe adding a couple of node sites just for the sake of having low coverage user ports. If your commercial site has good enough coverage of the city your cellular user port/nodes can be made using low power handy talkies. One watt commercial UHF handy talkies can be readily had for less than $100. Used two meter ham gear is pretty cheap.A simple UHF antenna, a two meter vertical, two feedlines, two TNCs and a power supply is all that is required to make a cellular user port. Now that you've got all of these simple repeater sites located in peoples homes, how long will it be before some more backbones start showing up into these sites? Your system will expand quickly as the ham radio public realizes how much fun it is to play with a real packet network.

### *Radios for Nodes*

Radios selected for node use should be capable of heavy duty use. The Tx/Rx switch circuitry should be able to handle virtually millions of operations without failure. This means PIN diode Tx/Rx switching as a first choice followed by high quality reed relay switching. Receiver front-end filtering should be quite sharp if your node is to coexist with other radio services. In that case consider using one or two tuned cavities to cut down on front end overload and desensitization. If your radio is operating on a simplex frequency, the cavities will also aid in reducing the effects of "white noise" being generated by the transmitter. At congested sites, a circulator may be required.

Some amateur class VHF radio's employing PLL frequency synthesizer technology should be avoided. Two reasons: PLL settling time between transmit and receive is too slow for optimum packet throughput. Consider the following table.

This is a table of maximum throughput in bytes per second assuming 230 bytes of data per 256 byte transmission with a 16 byte acknowledgment, for each popular data rate and with different TXDelay settings which would be the same on both ends of a data link:

| baud | byte | Throughput given TXD of | | | | |
|---|---|---|---|---|---|---|
| rate | time | 0ms | 40ms | 250ms | 350ms | 500ms |
| 1200 | 6.67ms | 127 | 121 | 99 | 91 | 81 |
| 2400 | 3.33ms | 254 | 233 | 163 | 143 | 121 |
| 4800 | 1.67ms | 506 | 431 | 241 | 199 | 158 |
| 9600 | .833ms | 1015 | 750 | 316 | 248 | 187 |
| 56K | .104ms | 8131 | 2124 | 435 | 316 | 223 |

Note that the actual TXDelay setting in the TNC Parms is in tens of milliseconds. Therefore the 500ms values in this chart would be achieved by setting the TXDelay to 50; 40ms values would be TXDelay of 4.

The length of a single byte BYTELENGTH = 8/baudrate
The length of one byte of data, including inefficiencies is
LOADEDBYTE = [(TXD x 2transmissions) + (BYTELENGTH x 272bytes)] / 230

Throughput per second = 1 / LOADEDBYTE

This means the speed of the radio's transmit to receive and receive to transmit switching is vitally important. Also, the transmitter may be keyed before stabilizing on frequency. This latter situation could cause interference to other receivers on different frequencies. This may be a serious concern if you choose a commercial radio environment for your node. If your candidate radio uses PLLs, solicit the manufacturers advice on suitability for packet node use.

In general, retired commercial service FM radios, such as the Motorola MICOR and GE MASTR II, or later, make excellent node radio choices. The commercial radios are designed to operate in moderate to high intensity RF environments, are physically rugged, and fairly reasonably priced on the used market. These radios typically come in a variety of power levels up to 110 watts (suitable for long haul dedicated UHF/6m backbone links; **User** ports should generally run less than 25 watts ERP.)

If this information is daunting to you then please just keep it in the back of your mind. If you are running your node out of a non-commercial radio site, like your home, then you can worry about this after you have your multiport node up and running. Using ham radio HTs and mobile rigs you can get things going and then swap out critical components later. The most important thing here is that you get your multiport node up and running with dedicated point to point backbones. Then you can worry about radio and baud rate improvements.

### TNCs

MFJ 1270B and PacComm Tiny 2 are the current models of the chosen TheNET TNC. Neither TNC needs modifications to work with TheNET. However, there is a bug with the MFJ1270B in that some models are shipped with the RS-232's control lines messed up. The general fix for this is to jumper pins 20 and 4 on the RS-232 connector for that model. If you use a Tiny 2 you can operate at 19.2Kbaud with the HexiPus™. Also the HexiPus™ pinout is identical to the Tiny 2 so you can use a straight through cable.

### Finally

Note: Don't compromise on anything. Be as high class in your system design as you can and still get results. This way your system will expand gracefully. If you compromise on your backbones and don't use point to point links you'll hurt your network very badly later. Do not, however, wait on high tech solutions when low tech solutions will work sooner. Many a packet system has been held up until interest was lost because 9600 baud equipment was going to be *working soon*. Put in the 1200 baud point to point link, then upgrade after you have the 9600 baud stuff working.

Word to the wise: **Never tell somebody that they can compromise temporarily. Compromise and temporarily should never be in the same sentence!**

# TheNET Node Mnemonics

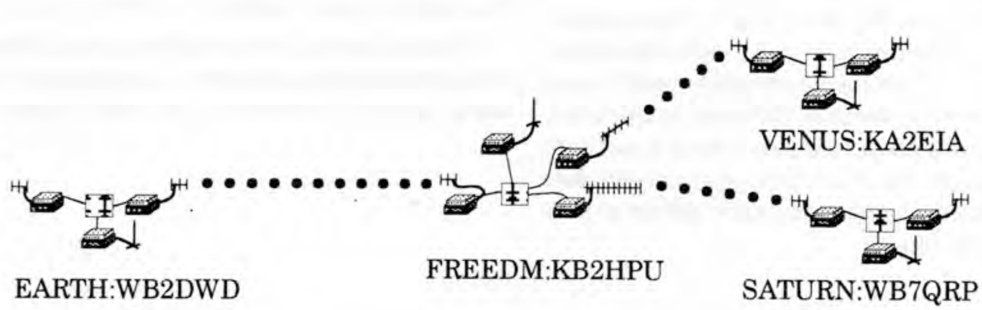
TheNET nodes incorporate a translation table that allows each TheNET TNC to be referred to by a six character mnemonic. This mnemonic is called the *node name*. Node names exist for the convenience of the human users. When ever the nodes talk to each other they use the callsigns automatically, even if the user uses the node name. So, whenever traffic is monitored between two nodes on a backbone callsigns will be seen.

TheNET nodes run in TNC 2s. Thus there is a limit to the amount of memory available both for program and data. The number of nodes that can be listed in memory is limited. Normally this is limited to 100 nodes. A feature of TheNET is that nodes used for backbone links can be made to not propagate automatically. That way the only nodes stored in each TheNET TNC's routing table are the user connect ports or node ports that must be visible for other reasons. The way that backbone nodes are set to not propagate is to name the node with a "#" character as the first letter in the node's name. A second feature of TheNET is that when a user requests a nodes list with the *NODES* command the node does not list the # nodes. Thus they are called hidden nodes.

### Backbone Ports

Backbone ports are not visible unless looked for but looking for them and navigating around the network is one of the most fun and educational parts of playing with TheNET. For this reason it is a very good idea to provide as much information as possible through the naming of the backbone nodes. Limitations on the names include that the first character must be a #. Five characters, all capitals, can still provide a good bit of info.

One choise for naming # nodes is to include the name of the site the node is located at, abbreviated to two or three characters, and then the name of the destination of the backbone link as the other two or three characters. Thus if we have a network as shown:




If we looked at the Routes command response from FREEDM we might see:

```
FREEDM:KB2HPU-1} Routes:
> 1 #FREAR:KB2HPU-12 203 38
  1 #FRVEN:KB2HPU-10 203 37
> 1 #FRSAT:KB2HPU-11 203 29
```

This makes it pretty clear where one would have to connect to look at the actual backbone TNC to go to, say, VENUS. If you went to #FRVEN:KB2HPU-10 and did a routes command you'd see:

```
#FRVEN:KB2HPU-10} Routes:
> 1 #FREAR:KB2HPU-12 203 38
  1 FREEDM:KB2HPU-1  203 37
> 1 #FRSAT:KB2HPU-11 203 29
> 0 #VNEAR:KA2EIA-13 203 28 !
```

Another method, and perhaps the as useful, is to use a site designator for three characters, followed by the compass heading, such as #LYNWS, #LYNEA, #LYNSO as backbone nodes for the LYNNWD node. What you'll see in the routes list at LYNNWD is:

```
LYNNWD:KA2DEW-1} Routes:
> 1 #LYNEA:KA2DEW-12 203 38
  1 #LYNWE:KA2DEW-8  203 37
> 1 #LYNSO:KA2DEW-11 203 29
> 1 CROWD:KA2DEW-7 203 1 !
```

What you'll see from the adjacent backbone node in Edmonds (west of Lynnwood) is:

```
#EDMEA:N0NDO-6} Routes
> 1 EDMOND:N0NDO-1  203 1
  0 #LYNWE:KA2DEW-8 203 22 !
```

It is pretty obvious from the routes list at Edmonds that we're seeing the LYNNWD node as a route. What's more important though is that from the LYNNWD user port it is pretty easy to figure out which # node to connect to if we want to look at the route to the west.

### User Ports

User ports should be labeled with a town or mountain, or in bigger cities, the neighborhood. Identifying a user port by club, region or airport is not as good. Same with using a name that is only known by the locals. The purpose for a node name is to identify it's location in the network for the outsider. Use the info text to mention the sponsor. Clubs are generally not known outside the area of it's influence and if it's area of influence is large then the location of the 'club' node is not obvious.

Naming a node by region, like SNJ or WNY or COHIO is a problem because if the node really does cover an area that large then it is probably useless due to hidden transmitter problems. Also the very existence of a node with such an all encompassing sounding name is that other people who might put up a node in the same region might feel that they are stepping on toes or that redundancy isn't desired.

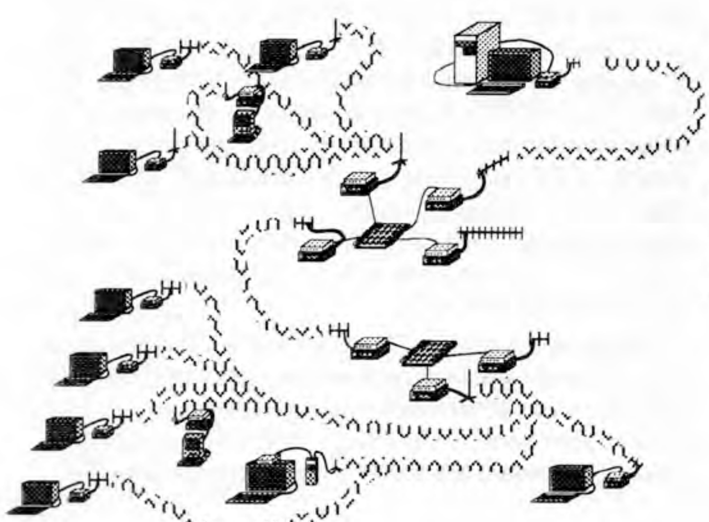
Airport identifiers may be sufficient for naming nodes as far as locating them on a city by city basis but there will no doubt be cases where the airport isn't near the city that it is named after, there are many more than one node per airport, or there is no airport nearby. Also many hams would have a hard time figuring out where a node is by it's airport identifier.

By all means name a node by it's city rather than by some cute code that's useless to all but the naming party.

The user port need not have the frequency in it's title. A node named ALB144 is not as obvious as one named ALBANY or ALBNY1. Again the info text, or a map, will fulfill the need for a frequency designator.

Here is a system for compressing a long city name into six characters which could also be used for compressing into five characters. This method was submitted by VE1YZ. VE1YZ gives credit to Boeing Corp. for documenting this standard for the aircraft industry.

Names with the desired number of letters or less are left alone.



Names with more than the desired number of characters are abbreviated using the following rules sequentially until the desired number of letters remain.

- Delete double letters.
- Keep the first letter, first vowel and last letter. Delete other vowels starting from right to left.
- Keep the last letter, then delete consonants from right to left.

Fixes for locations with multi-word name

- Use the first letter of the first word and abbreviate the last word using the above rules sequentially until only the desired number of characters remains.

Examples for six character node names:

```
Albany -> ALBANY
Lake George -> LGERGE
Seattle -> SEATLE
Manchester -> MNCHSR
Syracuse -> SYRCSE
```

It's quite possible to come up with better node names than that which this algorithm generates. For instance SYRCUS might be better than SYRCSE or LKGRGE might be better than LGERGE. It's obvious however that this is hands down better than LKG for Lake George or MAN for Manchester.

### Specialized Ports

For dedicated links and server ports, it is a good idea to pick a label that fits either the application, or to use a abbreviated version of the site's main user port with a number after it. *Make sure the info text clearly spells out what the port is for.*

Examples of specialized ports in our network are DXCLUS at UTICA node which is the DxCluster uplink, and DXKNOX at KNOX which is near Albany NY and serves the YCCC/AARA DxCluster run by K2TR. Give consideration to where your label will show on the list. This was one of the reasons that BBSxxx was chosen for G8BPQ and MSYS BBS ops, so that the BBS's would all be listed in the same part of the nodes table!

### Local Use Ports

In some node stacks it may be useful to create a visible node that only talks to the node owner. One way of doing this is to create a node named LOCAL. When burning the eprom for LOCAL tell it to turn off nodes broadcasts entirely and set the default RS-232 quality to 50. Now lock LOCAL in at the user port(s) at your node stack at a quality of 50. A more complete description of this is in the *Building Nodes* section.

# Building Nodes

This section discusses EPROMs and files, hooking up several TNCs into a node, testing the node, building wireline links, figuring out what's going on from the wink-n-blinks, setting the TxDelay. By this time you should have completed the modifications specified in the section *Setting up a TheNET X-1J TNC* to two TNCs. Please use two different callsigns for the two node EPROMs. Having the same callsign will cause problems.

When burning your EPROMs you'll want to make binary images on the PC. One easy way to remember which image is for which EPROM is to name the images with the nodename followed by a dot one for the low image or dot two for the high image. Thus a directory might look like this:

```
Volume in drive C is DOS5
Directory of C:\X1J\SPARTA

.            <DIR>     02-10-94 11:25p
..           <DIR>     02-10-94 11:25p
PATCH    EXE     56545 08-18-93  8:40p
#SPRFR   1       32766 08-12-93  8:21p
#SPRFR   2       32766 08-12-93  8:21p
#SPRNK   1       32766 08-12-93  8:21p
#SPRNK   2       32766 08-12-93  8:21p
SPARTA   1       32766 02-10-94 11:49p
SPARTA   2       32766 02-10-94 11:49p
         9 file(s) 253141 bytes
         38989824 bytes free
C:\X1J\SPARTA>
```

### Coupling TNCs together

A node site consists of two or more TheNET TNCs. The TNCs talk to each other via their RS-232 ports. A simple cable may be made to connect two TNCs for a 2 radio node (also known as a 2 port node). The two-port-node-cable is shown on a side bar. For more than two TNCs you'll want to use a diode matrix. The matrix allows the TNCs to talk to each other in groups of more than 2. There are two manufacturers of diode matrices at this time. They are A.N.S. (Amateur Network Supply) POBox 219 Montvale NJ 07645, and NX2P Electronics 201-729-6927. Both cost around $35. The NX2P unit is five ports while the A.N.S. unit is six ports. The NX2P unit may a higher quality PC board and is more conventional in the way it uses a cable to each TNC. The A.N.S. unit is commonly used with a stack of TNCs on edge (see photo) and no cables. The A.N.S. unit is used with cables when different brands of TNCs are mixed. Cables for either matrix and any of the TNCs is available from computer accessory vendors and electronics supply companies. Both matrices use DE-9 connectors. When used with 9-pin TNCs (DRSI and PacComm) a straight-through 9-conductor cable may be used (or no cable with the A.N.S.). When used with a 25 pin TNCs (MFJ), a jumper must be added, either in the 25-pin connector itself, or in the TNC. See side-bar table. The instructions that come with both products describe the cabling required for 25 pin TNCs.

#### *How TheNETs talk on RS-232*

TheNET software has two different modes of communications over it's RS-232 port. It can talk in ASCII, as in to a CRT terminal, or it can talk in TheNETese, as in to another TNC or PC compatible. In order to use a PC compatible you'll need to be running one of several programs on the PC compatible that understands TheNETese. NOS, MSYS and G8BPQ are all capable of this. (See articles in the *Quarterly Compendium* later in this Annual for more information on hooking up these products to the matrix). ***A TheNET TNC will talk TheNETese to another TheNET or PC program only if the TheNET jumper is installed.*** The jumper is between two pins on the RS-232 connector of the TNC. On the PacComm Tiny 2 the jumper is between pins 5 and 9. On the MFJ 1270B the jumper is between pins 10 and 23. If you are constructing your own diode matrix or hooking up a TNC to a PC you'll need to make this jumper. If you use a 3 wire (Rx, Tx and Ground) RS-232 interface to a dumb terminal the jumper will not be in place and the TNC will talk ASCII.

Note that in some older TAPR2 compatible TNCs and MFJ 1270 TNCs you'll need to run a jumper inside the TNC box (under the PC board) between the ground side of JMP9 to pin 23 on the DB25. The ground side of JMP 9, which is a six pin jumper, is the side of the set of pins that has three pins connected together. This should be on the *front* side of that jumper set.

#### *Test your 2 port node*

Plug power into two TNCs and attach them to the matrix or to each other via the dual-port-node-cable. Power on one TNC. You should see the CON LED go on and then off. Now power up the other TNC. If the matrix/cable is working correctly the STA LED of the first TNC should come on as you power up the 2nd TNC. The CON LED on each TNC is its RS-232 PTT signal. The STA LED is the TNC's RS-232 carrier detect signal. If you toggle the first TNC on and off, the STA light of the 2nd TNC should follow the CON light of the 1st. Each time a TNC sends a message over the matrix the LEDs should flash. One minute after the power is applied to a TNC it will do a nodes broadcast. On TNCs where broadcasting is enabled, the broadcast will be made on the enabled ports. You should see this on the LEDs. Now you can hook up some radios and see if you can connect to your node.

If you had a problem, like the CON LED didn't light (even at power-up) or stays on all the time, or if the STA light stays on all the time, you probably have a wiring error in your wink-n-blink mod. Perform visual inspection and go back to the article on setting up an X-1J TNC.

## Pin #s for 9-pin matrix to Tiny-2 and 9-pin matrix to MFJ 1270B cables

| *Pin name* | *matrix DB9* | *Tiny 2 DB9 female* | *MFJ 1270B DB25 male* |
|---|---|---|---|
| unused | pin 1 | pin 1 | |
| RxData | pin 2 | pin 2 | pin 3 |
| TxData | pin 3 | pin 3 | pin 2 |
| unused | pin 4 | pin 4 | |
| sig ground | pin 5 | pin 5 | pin 7&1&shell |
| unused | pin 6 | pin 6 | |
| DTR | pin 7 | pin 7 | pin 4 & 20 |
| CTS | pin 8 | pin 8 | pin 5 |
| TheNET enable | pin 9 | pin 9 | |
| DB25 TheNET enable jumper 10 -> 23 | | | pin 10 |
| DB25 TheNET enable jumper 10 -> 23 | | | pin 23 |

If you are making up your own 9-pin to 9-pin cables you do not need to wire the unused pins.

Having the unused pins wired on 9-pin to 9-pin cables will not make any difference.

If you are using a stock 9->25pin cable or adapter you will need to jumper pin 22 to pin 23 inside the TNC. See the instructions that come with the matrix.

## 2 port node cable

***For MFJ 1270B***

| 1st DB25 | 2nd DB25 |
|---|---|
| pin 1&7 | pin 1&7 |
| pin 2 | pin 3 |
| pin 3 | pin 2 |
| pin 4&20 | pin 5 |
| pin 5 | pin 4&20 |
| pin 10&23 | |

You should have 9 pins in use on each connector.

***For PacComm Tiny 2***

Make these connections

| 1st DB9 | 2nd DB9 |
|---|---|
| pin 2 | pin 3 |
| pin 3 | pin 2 |
| pin 5&9 | pin 5&9 |
| pin 7 | pin 8 |
| pin 8 | pin 7 |

You should have 6 pins in use on each connector.

### *Talking to your node over the radio*

The first thing you should do is connect to the node. Now do a ROUTES command. If it has been greater than a minute the neighbor node should show up. I'm going to use K1QRM-1:FRED and K1QRM-12:#FRJOE as examples for your two nodes. Having connected to FRED and done a routes list you should have something like this:

```
FRED:K1QRM-1} Routes
  1  #FRJOE:K1QRM-12  203  1
```

Since you may not have modified the parameters from the defaults (you'll have to do that next time you burn these EPROMs), the number 203 might be different. If you do not see an entry for K1QRM-12 you'll have to add one. The simplest way to sysop (make changes to) your node is disconnected it from it's node cable or matrix and plug it into the computer/terminal. Having done so, type Esc C and return.

The node should come back with:

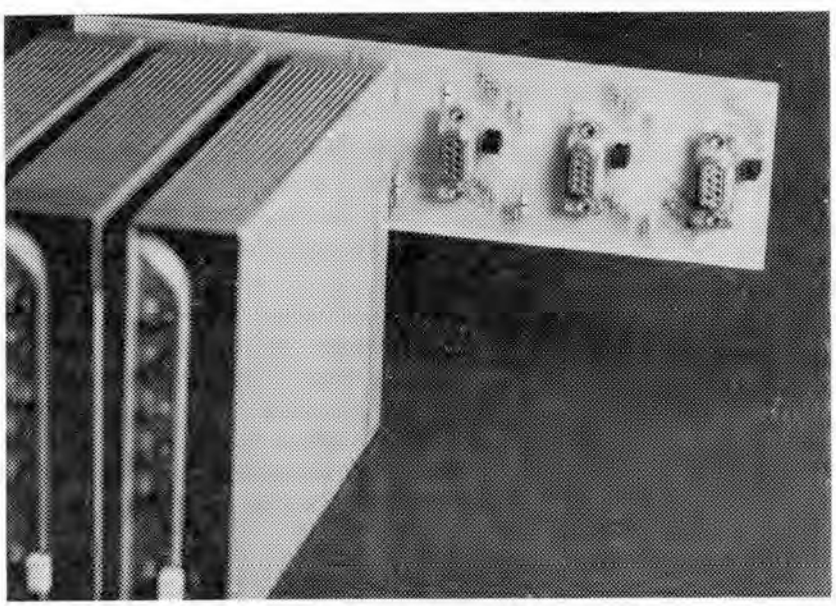
```
THENET X-1J K1QRM-1
By DF2AU/DC4OX/G8KBB
```

Type R <return> and N * <return>. If the lists are empty (They should still be if they were before) you can manually add the neighbor with the command:

```
N K1QRM-12 + #FRJOE 203 0 1 K1QRM-12
```

The node should respond with:

```
FRED:K1QRM-1} Routes to: #FRJOE:K1QRM-12
  203 0  1  #FRJOE:K1QRM-12
```



*Amateur Networking Supply's 6 port "Netrix" diode matrix board*

Check the parms and mode lists. Look at modes 5 and 8 and parms 1 thru 8. Make sure that they match up with a user port TNC (since you are connected into FRED, your user port). It is must important that you are accepting and making RS-232 node broadcasts.

Now unplug from the TNC, connect it back to the other node via the cable or matrix and connect back up to the node via your radio. Now try connecting to K1QRM-12. You should see the CON light on FRED (Tx) and the STA light on #FRJOE (Rx) flicker at the same time. Immediately after you should see #FRJOE PTT and FRED Tx and #FRJOE Rx. You should then get a connect message on the screen.

If FRED did Tx and #FRJOE didn't acknowledge with it's own Tx, (and you didn't get connected), disassemble the TNCs and check the baud rate. Also ohm out the connections between the two TNCs. Inspect the Matrix. If the STA light stays on all of the time while the TNC is connected to the matrix or cable, then disconnect the TNC from the matrix and check it carefully for shorts. Also consider opening up the TNC and checking the wink-n-blink mod. A short in one TNC could cause a problem in the other TNC.

We'll get back to working with the node in a few paragraphs.

### *How to maintain the node stack*

In the future you'll want to have a reliable way to get into your node stack. If you are a visitor to the node site when you are servicing it, a portable radio and computer might be ideal. You might leave a dumb terminal on site and bring a TNC and HT with you when you visit.

If you live at the node site or have a station there that is used often you should consider a wireline link. This is a system whereby your user station is connected into the node stack without going over radios. This is also the great method to tie equipment into the node stack or to link two node stacks together if you have more than six radios to tie in. There an article in the compendium called Multi-way Wireline Node which describes a method for constructing wireline links. There is a kit for this from A.N.S. which sells for a few dollars and includes little PC boards for this project.

Another scheme for getting a port into the node stack is to build the *Two Radios On One TNC* project from the *Compendium* in this *Annual*.

You could use a G8BPQ, MSYS or NOS program on a PC wired into the matrix appropriately. Check out articles on this subject in the *Compendium*. I strongly recommend the wireline link.

### *Wireline link user port*

Burn an EPROM for a node called LOCAL or WIRELN (or something similar). We'll assume that you are using "LOCAL" for this discussion. Use the parameters and modes shown in the side bar. Configure a TNC as an X-1J node and test it with the LOCAL eprom. Configure the second TNC as a regular user port with the supplied EPROM. Set the baud rate on both the X-1J and user TNC at 9600 baud.

Build the circuit shown in the article or in the instructions that come with the A.N.S. kit or from the article. Make sure you cut the two traces on the bottom of the TNC boards. Plug your terminal/computer into the user TNC and power up. Connect to LOCAL. Observe the PTT lights on the two TNCs. If you are using PacComm TNCs the DCD lights will track the alternate PTT light (as you would expect). On MFJ TNCs this will not be the case because the MFJ's DCD signal is detected on the modem side of the Disconnect Header, not on the computer as is the case with the PacComm.

Hit a few carriage returns and observe the light patterns. The STA light on the user TNC should not stay on for very long.

Now go back to command mode on the user TNC and set TXDelay to 1, DWait to 0, RESPTIME to 0. That should make things nice and fast.

Sysop LOCAL and see that your password works. Set LOCAL's info text to something like this:

```
LOCAL:K1QRM-10} }
wire link TNC connected to K1QRM's user station
C K1QRM-4 to get Tadd's mail drop and more information
then do L to list messages
```

### *Notes on the use of LOCAL:*

After you get through the testing stage listed below, and if LOCAL will be a permanent part of your node stack, you'll want to go to each of your TheNET TNCs and lock in LOCAL as a node. LOCAL will not broadcast if the parameters were set as in the side bar because if it did, each time you brought local with you to test a site, LOCAL would show up all over the network. Instead it is left to the node op to lock in LOCAL using the sysop's N command. When you do lock it in you should set it to a quality of 50 and a lifetime of 0 (permanent). That way it will only be visible at your node stack. Other node ops may also have a LOCAL node. You really only need to lock in LOCAL at your user ports.

### *Final Set-up of the network node*

After you have your node talking within it's RS-232 LAN it's time to make it talk to the neighbor nodes. You should sysop the #nodes and lock in routes to the neighbor nodes. Also make sure the parameters match those in the back of this Annual. Once this is done, the neighbor's broadcasts will be picked up and passed around in your node stack. You should also use the ACL command as recommended in the sysop command section to enable the neighbor callsigns and prohibit all other L2 activity. Set the INFO text messages to reflect what the TNCs will actually be used for.

## Parameters for Wireline link node running X-1J

| # | Function | |
|---|---|---|
| 1 | Size of destination node table | 100 |
| 2 | Minimum auto update quality | 50 |
| 3 | HDLC (radio port) default quality | 50 |
| 4 | RS-232 (crosslink) default quality | 50 |
| 5 | Initial obsolescence count | 3 |
| 6 | Min Obs to broadcast | 2 |
| 7 | Nodes broadcast interval | 900 |
| 8 | Initial time-to-live | 5 |
| 9 | Transport FRACK timeout (seconds) | 20 |
| 10 | Transport RETRY counter | 1 |
| 11 | Transport acknowledgement delay | 1 |
| 12 | Transport busy delay | 20 |
| 13 | Transport window size | 2 |
| 14 | Transport overfill limit (frames) | 4 |
| 15 | No-Activity time-out (seconds) | 7200 |
| 16 | Persistence (n/256) | 255 |
| 17 | Slot time | 1 |
| 18 | FRACK (T1) time | 1 |
| 19 | AX.25 windowsize (MAXframe) | 1 |
| 20 | AX.25 retries | 10 |
| 21 | ACK (T2) time (link layer RESPTIME) | 50 |
| 22 | Active check (T3) | 32000 |
| 23 | Digipeat | 0 |
| 24 | Callsign validation | 0 |
| 25 | Beacon mode control | 0 |
| 26 | CQ broadcasts | 1 |

**MODES**

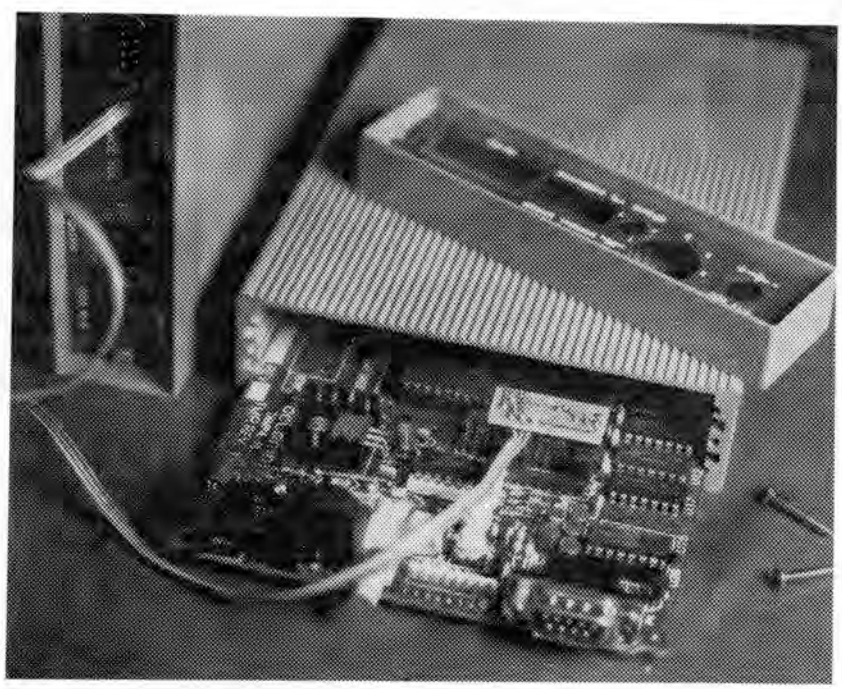
| # | *Function* | |
|---|---|---|
| 1 | RS-232 host mode | 0 |
| 2 | CWID repeat period ( seconds ) | 0 |
| 3 | CWID keyer speed | 6 |
| 4 | Selective nodes broadcast on ports: | 2 |
| 5 | RS-232 crosslink protocol | 0 |
| 6 | Transmit keyup delay | 3 |
| 7 | Full duplex | 0 |
| 8 | Crosslink port node b'cast interval | 450 |
| 9 | Node broadcast algorithm port control: | 3 |
| 10 | Beacon period( seconds ) | 600 |
| 11 | Connect redirector | 0 |
| 12 | User message control flags | 27 |
| 13 | Hash (#) node broadcast port control | 3 |
| 14 | Extra aliases | 1 |
| 15 | Auto reconnect to node | 0 |
| 16 | Control of slime trails. | 0 |
| 17 | Digipeat up/downlink control: | 0 |

### In summary

If you can think of anything that I've left out in regards to getting a node started please drop a packet to NEDA @ WB2QBQ.ny, ATTN: Annual Editor.

One rule to keep in mind when making a node or network, don't compromise. Don't do anything temporarily. Assume that any temporary fix that you are contemplating is permanent. If you compromise on something once and loose a bit of potential performance, you will send a message to the users of your system and to other potential system builders that this is what they can expect and this is OK. Wrong!

Make sure you have fun and spread the word. 73!



*Amateur Networking Supply's WireModem board installed in a Tiny-2*

# TheNET X-1J Quick Reference Guide

***Switch Commands***

ACL [ { callsign + value } | { callsign - value } | { * value } ]
ALIAS [ * | new_alias ]
ARP [ ipaddr [ { - ptcl } | [ + [ P ] ptcl callsign [ DG | VC ] ] ]
AUDIT [ number_from_0_to_255 ]
BBS [ callsign | * | ? ]
BBSALIAS [ * | new_alias ]
BTEXT [ * | beacon_message_text ]
BYE
CALIBRATE [ period_value_from_1_to_60 [ toggle_value_1_to_period ] ]
CLOSEDOWN a
CONNECT [ callsign [ [V] digilist... ]
CQ [ message_for_CQ_packet ]
CTEXT [ * | connect_message_text ]
DXCALIAS [ * | new_alias ]
DXCLUSTER [ callsign | * | ? ]
HELP
HOST [ callsign | * | ? ]
HOSTALIAS [ * | new_alias ]
INFO [ * | text ]
IPADDRESS [ new_IP_address ]
IPBROADCAST [ new_IP_address ]
IPROUTE [ ipaddr [ /bits ] [ - | { + port [ ipgateway [ metric ] ] ] ]
IPSTATS [ { NEWPARAM | * } { NEWPARAM | * } ..... ] (see note 5)
LINKS
MANAGER
MHEARD [ number_from_1_to_100 ]
MODE [ { new_param | * } { new_param | * } ..... ] (see note 5)
MTU [ list_of_parameters ] (see note 5)
NODES [* [*] | call {+|-} ident qual count port neighbor [digis]]
PARMS [ { newparam | * } { newparam | * } ..... ] (see note 5)
QUIT
RESET [ any_character ]
ROUTES [ port nodecall [ digilist ... ] { + | - } pathquality ]
STATS
SYSOP
TALK [callsign text_to_send ]
UI dest string_of_text_to_be_sent_in_UI_frame users

*notes*

1 Any command may be enabled or disabled by the use of the '+' or '-' modifier, as shown below:
ANY_COMMAND [ + | - | that_command's_parameters ]
2 IP addresses form is nnn.nnn.nnn.nnn where nnn is 0..255
3 IProute port parameter takes the form 0 or 1 for radio or rs232 AX.25 or N for Net/Rom
4 ARP ptcl parameter is AX.25 or Net/Rom ( may be abbreviated )
5 PARMS/MODE/IPSTAT commands may be specified using the old syntax or with the new syntax where PARMS /5 20 sets parameter 5 to 20.
6 BTEXT, CTEXT and INFO may be multiple lines. Refer to manual.

HCKNSK:KA2DEW-1} 100 50 0 203 7 5 900 13 180 1 1 120 2 4 7200 255 10 5 1 10 50 32000 0 1 2 1

| # | Typical | Function |
|---|---|---|
| 1 | 100 | Size of destination node table |
| 2 | 50 | Minimum auto update quality |
| 3 | 0 | HDLC (radio port) default quality |
| 4 | 203 | RS-232 (crosslink) default quality |
| 5 | 5 | Initial obsolescence count |
| 6 | * | Minimum obsolescence count for broadcast |
| 7 | 900 | Nodes broadcast interval (seconds) |
| 8 | 13 | Initial time to live (hops) |
| 9 | 200 | Transport (FRACK) timeout (seconds) |
| 10 | 1 | Transport retry counter |
| 11 | 1 | Transport acknowledge delay (seconds) |
| 12 | 180 | Tranpsport busy delay |
| 13 | 2 | Transport window size |
| 14 | 4 | Transport overfill limit (frames) |
| 15 | 7200 | No-activity time-out (seconds) |
| 16 | * | Persistence (n/256) |
| 17 | * | Slot time |
| 18 | * | FRACK (T1) time |
| 19 | 1 | AX/25 window size (MAXFrame) |
| 20 | 10 | AX.25 retries |
| 21 | * | Ack (T2) time (link layer RESPTIME) |
| 22 | 32000 | Active check (T3) time |
| 23 | 0 | Digipeat |
| 24 | * | Callsign validation |
| 25 | * | Beacon mode control (0=off, 1=only if active, 2=always) |
| 26 | * | CQ broadcasts |

* indicates value depends on application.

### *IPSTAT Parameters*

| # | Min | Max | Function |
|---|---|---|---|
| 1 | 0 | 3 | ip L2 AX.25 Modes ( 1 bit per port, 1=DG ) |
| 2 | 0 | 1 | ip Forwarding, 1=enable router, 0=disable |
| 3 | 2 | 255 | ip Default TTL |
| 4 | 0 | 0 | ip In Receives |
| 5 | 0 | 0 | ip In Header Errors |
| 6* | 0 | 0 | ip In Address Errors |
| 7 | 0 | 0 | ip Forwarded Datagrams |
| 8 | 0 | 0 | ip In Unknown Protocols |
| 9* | 0 | 0 | ip In Discards ( TTL exceeded ) |
| 10* | 0 | 0 | ip In Delivers |
| 11 | 0 | 0 | ip Output Requests |
| 12* | 0 | 0 | ip Output Discards |
| 13 | 0 | 0 | ip Output No Routes |
| 14* | 1 | 30 | ip Reasm Timeout |
| 15* | 0 | 0 | ip Reasm Requireds |
| 16* | 0 | 0 | ip Reasm OKs |
| 17* | 0 | 0 | ip Reasm Fails |
| 18 | 0 | 0 | ip Frag OKs |
| 19* | 0 | 0 | ip Frag Fails |
| 20 | 0 | 0 | ip Frag Creates |

Those marked '*' are not used in this version

### *Mode Parameters*

| # | Min | Function |
|---|---|---|
| 1 | 0 | Hardware handshake in host mode |
| 2 | 0 | CWID repeat period ( seconds ) |
| 3 | 6 | CWID speed ( 10's msecs per dot ) |
| 4 | * | Node broadcast channel enable flags<br>bit 0 = enable on HDLC<br>bit 1 = enable on RS232 |
| 5 | 0 | Crosslink protocol selection 0=crosslink,<br>1=KISS, 2=KISS+selcopy,<br>3=KISS+allcopy |
| 6 | * | TxDelay in 10s of mS. |
| 7 | 0 | Full duplex enable flag |
| 8 | 450 | RS232 node broadcast period in secs. |
| 9 | 3 | Node broadcast algorithm control flags<br>0=off, 1=HDLC port<br>2=RS232 port, 3=both ports |
| 10 | 600 | Beacon interval (seconds) |
| 11 | * | Connect redirection to BBS flag |
| 12 | * | Help messages enable flags |
| 13 | 3 | # node broadcast disable flags<br>bit 0 = no # nodes on HDLC broadcast<br>bit 1 = no # nodes on RS232 broadcast |
| 14 | * | Enable extra aliases monitoring if set |
| 15 | * | Enable auto reconnect to node |
| 16 | * | Slime trail control.<br>bit 0 = hides slime trails in nodes listing<br>bit 1 = Do not allow slime trailing |
| 17 | * | Digipeat control.<br>bit 0 = refuse digi'd L2 uplinks<br>bit 1 = refuse digi downlinks |

### *Audit Bits*

| Bit | Function |
|---|---|
| 0 | Issue L1 stats every 10 minutes |
| 1 | Audit L2 connects & disconnects |
| 2 | unused |
| 3 | Audit L4 connects & disconnects |
| 4 | Audit L7 use of sysop command |
| 5 | Audit all L7 switch commands |
| 6 | Issue CPU stats every 10 minutes |
| 7 | unused |

### *ACL Bits*

| Bit | Function |
|---|---|
| 0 | Bar all incoming L2 connects |
| 1 | Bar outgoing L2 downlinks |
| 2 | Ignore nodes broadcasts |
| 3 | Bar gatewaying at level 3 |
| 4 | Bar incoming L4 connects |
| 5 | Bar outgoing L4 connects |
| 6 | ignore SSID in searching |
| 7 | unused |

### Host *Escape* Commands

| | |
|---|---|
| <escape> C | Connect to TheNET node |
| <escape> D | Disconnected |
| <escape> P [ new_password ] | |

### *User Message Control Flags (for parm /12)*

| Bit | Function |
|---|---|
| 0 | Issue a message when trying BBS etc. |
| 1 | Show all available commands to sysop |
| 2 | Say 'goodbye' nicely |
| 3 | Give connect text on uplink to alias (CTEXT) |
| 4 | Show nodes in routes as alias:callsign |
| 5 | Pass 8 bit data in TALK command |
| 6 | Make node alias handling case sensitive |

# Common Problems

•••The TNC seems to be operating at an extremely low baud rate or it takes forever to transmit.

The problem may be in the baud rate generator or the baud rate switch. Check the jumpers if it's a PacComm. If it's an MFJ it may be a broken switch or the switch may be set to an improper value.

•••If the node is doing funny things like:

- only working over the radio or only working over RS-232
- not responding at all
- using the wrong callsign or node name
- transmitting all the time
- not handling the LEDs right
- disconnecting everybody once in a while
- never nodes broadcasting or ignoring incoming broadcasts entirely

Try sysopping the node and using RESET or if you have local access you can turn off the TNC, remove the battery jumper for more than 30 seconds, replace the jumper and then turn the TNC back on.

•••If your new node stack doesn't allow you to connect across the matrix, make sure that you have waited long enough for nodes broadcasts to work. The nodes broadcasts over the matrix happen at the same time they do over the radio. Just because the TNCs are stacked they don't necessarily know about each other. It is possible using the Sysop:NODE command to lock in each port in your node stack to each other port so they begin communications immediately. This is a good way to get used to doing sysop commands.

Also if you are using TheNET 2.09 or later you can reboot the TNCs. Turn off one, turn it back on and wait 5 seconds. Now do it to the next one. That will make sure that all of the TNCs have node listings for each other.

•••You've changed the parameters so that the default route quality is 200 but the routes are still at the old value.

The route qualities are set by the parms when the route first comes into existence. If you want the route quality to change you must either change them manually using the sysop route command or by making the routes go away and then come back. This can be done by increasing the nodes broadcast rate (parameter 6) temporarily or by disconnecting the radio or matrix for several nodes broadcast intervals.

When a neighbor node is first heard the route quality is set based on the parameters. Simply changing the parameters does not change the route quality. However, if you change a route quality to a neighbor node, the nodes sourced from that neighbor node will change in value as soon as the next nodes broadcast is heard from that neighbor.

•••Traffic across the node stack seems to get delayed by several seconds even though the backbones in and out are not particularly busy

This may be caused by incorrect wiring of the matrix, connectors to the TNCs or problems with the TNCs themselves. Make sure that the TNCs respond correctly. If you haven't done the wink and blink mod you should do this now. What may be happening is that the CTS signal isn't getting asserted onto the diode matrix when a TNC goes to send data over the matrix. The other TNCs may be transmitting at the same time.

•••One or more nodes show on your NODE list but you can't connect to it.

This may be caused by the quality levels being set too high on one of the intervening nodes for the time-to-live settings on one of the two end nodes. This is common when interfaces between non-compliant networks are incorrectly constructed. Ideally a gateway would be set up which would be used as a stepping point between the networks. If no such gateway exists and the nodes are allowed to propagate across the interface, such non-connectability problems are bound to happen. Your best bet is to find that interface and step across it manually.

It could also be caused by the quality levels being uneven going on the reverse path back to your node from the destination node. Normally this second problem is fixed by what is called slime-trailing. Slime trails don't work if one of the TNCs along the way has a full NODE list.

•••If you have any other problems or figure out a problem for yourselves make sure you send a message about this to

**NEDA @ WB2QBQ.ny.usa.na**
**subj: ATTN. tech documentation.**

# Glossary of Packet Networking Terms

*See the end of the glossary for credits, comments on content and copyright notice.*

ACK
acknowledgment
AEA
AFSK
AFT
algorithm
alias
Aloha net
Amiga
AMRAD
AMTEX
AMTOR
analog signal
ANSI
APLink
ARPA Suite
ARQ
ASCII
ASLIP
asynchronous
audit trail
autoforward
autorouting
AX.25
B2A/A2B
backbone
backoff
back-up battery
baseband
baud
baudot
Baycom
Baypac
BBS
beam
BER
BERT
beta test
BID
bit-stuffing
BPQ
bps
breakout node
BSQ
BTW
callbook server
carriage return
CCIR
CCITT
cellular
chat node
choke/unchoke
circuit
CLNS
CLOVER
cluster
CMOS
collision
collision avoidance
Commodore 64
compressed forwarding
conference node
connectionless
connection-oriented
CONS
contention
control character
converse mode
converse node
COR
coverage
CPU
CRC
CROWD
CSMA
CSMA/CA
CSMA/CD
CTS
data
data collision
DataEngine
data rate
data set
datagram
DB-9 connector
DB-25 connector
DCD
DCD hold-off
DCE
DE-9 connector
dedicated link
dedicated port
defined neighbor
deviation
DFD-PBBS
diddle
digi
Digicom>64
digipeater
digital signal
diode matrix
discriminator
disk drive
DOERS
dogbone
DOVE
downlink
DRSI
DSP
DTE
DTR
dumb terminal
duplex
duplex digipeater
DWAIT
DxCluster
dynamic rerouting
envelope delay distortion
EOC
EPROM
EPROM
ERS
eye pattern
false route
FBBS also FBB BBS
FCS
FEC
F.E.M.A.
firmware
flag
flat network
floppy disk
See disk drive
flow control
FM
forwarding
forward file
forwarding frequency
FRACK
frame
FTP
FSK
full-duplex
G3RUH modem
G8BPQ Code
gateway
group delay distortion
half-duplex
handshake
HAPN
HAPN-T modem
hard disk
HDLC
header
heard list
hex
HEX9
HexiPus™
hidden transmitter syndrome (HTS)
hierarchical address
hierarchical routing
hold-off
host
host mode
HTF
HTS
HTS free
IF
IF Bandwidth
image
Internet
IP
IS
ISO
jitter
JNOS
K9NG
KA9Q Internet
Kantronics
keyboard-to-keyboard
kill
KISS
LAN
LAPB
Lan-Link
link layer
local area network
locked node
locked route
loopback
Macintosh
mail box
mail drop
mark
matrix
matrix monitor
MBL BBS
MFJ
MID
MIR
mnemonic
MP-Net
modem
modem header
MSYS
MTL/WQC Network
multistreaming
NAK
NAPRA
NBOD
NEDA
Neighbor
NEPRA
NET/ROM
network
network layer
node
node broadcasts
node cluster
node hopping
node-op
node stack
NOS
NRZ
NRZI
NTECH
null modem
obsolescence count
Octopus
OEM
OPEN
OSI
overhead
PacComm
packet
packet switch
PacketCluster
PacketTen
PACSAT
PACTOR
PAD
Paket
PakMail
Pakratt
parameters
parity
PARMS
path
PBBS
PC
PC*PA
perming
persistence
personal BBS
PID
PLL
PM
PMP
PMS
point to point link
polling
poll packet
port
PROM
protected backbone
protocol
pseudo-digital repeater
PSK
PSR
QRM
QRN
quality
R: header line
R95
RAM
RATS
real time
REBBS
redundancy path
regenerating repeater
repeater, full duplex real time regenerating
response time
retry
reverse forward
RLI BBS
RM
ROM
ROM image
ROSE
ROSErver/PRMBS
Route Stepping
router
routing loop
RS-232
RTS
RTTY
RUDAK
RXData
SAREX
selective neighbor routing
semi-duplex
serial port & serial communications
server
shared, HTS free backbone circuit
SID
simplex
simplex digipeater
site hardening
site manager
site sponsor
site sysop
SITOR
slime trail
slottime
SMTP
space
SSID
STA
store-and-forward
stream
switch
synchronous
sysop, BBS sysop
TAPR
TCP
TCP/IP
TELNET
terminal
TEXNET
TheNET
TheNET PARMS
TheNET Plus
TheNET X
three way wireline link
throughput
time to live
throughput
Tigertronics
time-to-live
TINK
TNC
topology (network)
TPG
TPK
transfer rate
transparent mode
transport layer
TTL
turnaround time
TXData
TX Delay
UA
UART
UI
unproto
uplink
user channel
VADCG
Vancouver protocol
VC
virtual circuit
WAN
weather node
wide area network
wide band packet
Wink 'N Blink Mod
wireline link
wirelink
WORLI
wormhole
X.25
XON/XOFF
YAPP
zoo channel
4RE BBS
7-Plus

## ACK
Acknowledgment. A packet sent by a receiving station to tell the sending station the packet was received correctly and the sequence number of the next packet that it would expect. The sending station then knows whether to send the next packet or to resend a missing packet. A form of handshaking. Also the 1 character handshaking response used in AMTOR mode A (ARQ mode).

## acknowledgment
See ACK.

## AEA
Advanced Electronics Applications Inc. AEA designs, manufactures and markets a wide range of amateur packet products as well as other amateur related items.

## AFSK
Audio Frequency Shift Keying. A method of digital modulation. This is a mechanism for sending digital information over a radio. A signal 0 is sent using one tone while the signal 1 is sent using a different tone. This is the mechanism used by telephone modems and packet radio modems.

## AFT
Amateur Framing Technique. This is the protocol used by ROSE switches on an RS-232 LAN. First proposed by Toby Nixon of Hayes, it is pending CCITT adoption as the accepted method for sending X.25 over asynchronous links.

## algorithm
An algorithm is a predetermined step by step procedure that solves a specific problem.

## alias
An alternate ID for a packet station. Usually an alias is used for user connects into a server station. The server station uses an amateur callsign for communication with other servers and also beacons it's callsign, with it's alias to reduce confusion and to handle legal identification requirements. In the case of TheNET compatible nodes the alias is also called a mnemonic.

## Aloha net
An early (1975) packet radio experiment conducted by the University of Hawaii. The Aloha net is known for it's performance definitions for packet systems which have hidden transmitters.

## Amiga
Perhaps the best of the personal computers available in the hobbyist price class the Amiga has seen terrible sales and lousy acceptance by the business and hobby computer community. It is most likely this is because of Commodore's poor marketing and the excellent marketing efforts of Apple Computer and IBM. The Amiga does not have very high acceptance in the amateur radio community either. If you are shopping for a computer you will want to check these out, however. Most varieties of amateur radio programming are available on the Amiga. Hams who own Amigas tend to be extremely supportive of new owners of Amigas. Programs available for the Amiga include Satellite tracking, TCP/IP, AX.25 terminal programs, educational and technical programming. (See *Macintosh, PC*)

## AMRAD
AMateur radio Research And Development corp. A nonprofit group based in Virginia devoted to advancing packet radio.

## AMTEX
AMtor TEXt. A bulletin broadcasting system used by AMTOR bulletin stations like W1AW. Similar to the US Coast Guard NAVTEX system of naval advisories.

## AMTOR
AMateur Teleprinting Over Radio. An improved method of RTTY that uses some forms of error recognition and correction to improve copy. Sort of a very simple basic form of packet using 3 character groups. AMTOR uses a limited character set (Capitol letters, numbers and a couple of controls like CR/LF - similar to the Baudot set). AMTOR is normally used on HF frequencies. AMTOR is modeled on a commercial protocol called SITOR. (See *ARQ, FEC, SITOR, PACTOR, RTTY, Baudot*)

## analog signal
An electrical signal that changes in a smooth continuous manner and whose voltage or current may represent a numerical value or other physical property. Compare with: digital signal which is a signal that changes in discrete unambiguous steps.

## ANSI
American National Standards Institute. A US organization that sets standards on all sorts of things from nuts to computers. (See *ISO*)

## APLink
Amtor Packet Link. An AMTOR BBS program written by Vic W5SMM and used on an IBM PC to operate an AMTOR mailbox and gateway between the packet BBS network and international AMTOR links.

## ARPA Suite
The set of protocols standardized by the Advanced Research Projects Agency of the US Dept. of Defense. Includes TCP and IP as elements, but leaves the lower levels (subnetwork and down) deliberately unspecified. The ARPA suite can be run on top of multiple subnetworks, unifying them into a single Internet.

## ARQ

Automatic Repeat reQuest. An error correction technique in AMTOR where the receiving station sends a 1 character ACK/NAK response to each AMTOR group sent. (See *AMTOR, ACK, NAK, FEC, handshaking*)

## ASCII

American Standard Code for Information Interchange. Also called USASCII. The standard code that defines text characters (and control characters) in terms of binary or hexidecimal numbers. It is used for the transfer of text between computing devices or input/output terminals.

## ASLIP

Asynchronous Serial Line Protocol (usually just called SLIP). A technique for encoding IP datagrams so they can be sent across ordinary asynchronous modems and communications hardware.

## asynchronous

A method of serial communication which uses start and stop bits to delimit each character and to synchronize the receiver to the data stream. The time between characters is undefined in async mode. Compare with: synchronous serial communication where a clock signal is encoded into the data stream in some manner and which does not need start and stop bits. RS-232 is an asynchronous mode of communication.

## audit trail

The list of headers (R: lines) attached to every BBS message that is used to trace the path the message has taken from its originating BBS. (See *header, path, R: header, BBS*)

## autoforward

Many of the bulletin board and mail server programs (BBS) are capable of passing messages to each other. The process of a bulletin board recognizing that it has mail to go to another bulletin board, connecting to another board and then sending the traffic is called Autoforwarding. This allows packet users to send mail in a non real time fashion anywhere on the planet where compatible BBSs exist. (See *Forward File*)

## autorouting

This is a process by which a network node can pass traffic to another node via one or more intermediate nodes.

## AX.25

This is the designation for the protocol used by TNCs to talk to one another. A book from the ARRL includes a [cryptic but] complete specification for AX.25. (See *Protocol*)

## B2A/A2B

Binary-to-ASCII/ASCII-to-Binary. This is a progran written by Brian Riley KA2BQE that takes a Binar computer file and converts it to ASCII characters, so i can be transmitted via packet. The same program i used to convert the ASCII characters back to the original Binary file.

## backbone

A backbone is a system of links where nodes may com municate without interfering with or being interfere with by local access, and where data may be passed i a fashion and with hardware that is optimized for passin data, rather than optimized for inexpensive user station

Example:

- Most user stations operate at 1200 baud on meters. A backbone would be more efficient an less susceptible to interference if it were on UH or 220. Also a backbone might be optimized b taking advantage of the knowledge of all radio on each link. Such optimization might include setting acknowledge delay (RESPTIME), Trans mit lead time (TxDELAY) or persistenc (PPersist) to values that work best for the radio on the backbone frequency. Such settings migh be impossible if any average user stations wer to be able to access the link radios. In additio baud rates might be increased if only a few radios/TNCs need be affected.

## backoff

When a packet is sent and not responded to, the sending station will wait a specified 'backoff' before retrying In simpler systems this is called "FRACK" or FRam ACKnowledge delay. In more complex systems, like TCP IP, the backoff time can be calculated based on previous performance of the link. One such backoff proce dure is called exponential backoff. In this system th amount of time delayed between resending a misse packet increases by a stable factor each time the packe is tried, until some maximum backoff time is reached (See *collision avoidance, DWAIT, FRACK, Persistence TCP/IP, Retry*)

## back-up battery

A lithium cell used in TNCs and other computing de vices to maintain the data stored in the RAM durin times when the device is powered off. (See *RAM EPROM*)

## baseband

The original data signal before any encoding or modu lation operations are performed on it is called the digi tal baseband signal.

## baud

The "signaling" rate on a data channel expressed as signal elements or symbols transmitted per second. A signal element or symbol may be a pulse or a burst of tone or any signal change that can be measured. A signal bit or symbol may contain more than 1 data bit so therefore baud is not equal to bps. A telephone 9600 bps modem usually operates 2400 baud with 4 possible data states for each signaling element. In packet radio one character of text equals 8 bits of information. 8 bits of information requires 8 transitions of the modem tones. That means that there will be about 150 characters of data in one second of 1200 baud transmission except for the fact that packet transmissions include key up time and callsign information which take up many characters of time. (See *bps*, *data rate*, *transfer rate*)

## baudot

A 5 bit code used on RTTY communications. Named after J. Baudot, an early French inventor of telegraphic instruments. Compare with: ASCII which is an 8 bit code capable of coding 256 characters instead of 32 for baudot. (See *ASCII*, *RTTY*, *AMTOR*)

## Baycom

A simple modem and software package designed and supported by DL8MBT, DG3RBU and the Baycom group in Germany. Early versions were shareware but the latest version (v1.5) is a commercial product marketed in North America by PacComm and Tigertronics.

## Baypac

A Baycom modem marketed by Tigertronics Inc.

## BBS

Bulletin Board System. This is a server which is accessed by packet stations to be a repository for messages and files. Those messages and files can be accessed by all packeteers who connect to the BBS, if desired. BBSs also have a capability called Forwarding which may be used to send files between BBSs. (See *autoforward*)

## beam

A directional antenna. Beams are usually made of aluminum and are constructed from a horizontal length of tubing and between 4 and 16 1/2 wave length aluminum elements. Beams for packet radio usually cost between $50 and $100. The key features of a beam are 1. Directionality and 2. gain. In this case gain means that because less power and signal is wasted in the direction the beam isn't aimed at, it is used in the direction the beam is aimed at. See the ARRL Handbook for more information on antennas.

## BER

Bit Error Rate. The average number of errors per given quantity of data bits transmitted on a communications system. Usually expressed as errors per thousand or million or some such figure.

## BERT

Bit Error Rate Test. Any error test on a communications system or component (such as a modem) to determine the average rate of occurrence of errors in the data stream being passed. (See *BER*)

## beta test

Beta test is the pre-release testing of hardware or software with selected typical customers to find out if there are any bugs or problems before releasing it to the general public.

## BID

Bulletin ID. A number given to each bulletin sent out in the BBS forwarding system for identification purposes and to prevent duplicates being created in the system. Compare with: MID which is a number given to all messages on the BBS system, personal as well as bulletins. The MID and the BID of a bulletin are the same. Personal messages do not have BIDs unless they are addressed to a distribution list like SYSOP.

## bit-stuffing

A technique used to prevent confusion between any 111111 bit pattern in the data and the flag character (01111110) used to delimit the start and end of each frame in the packet.

## BPQ

A node or packet switch software written by John G8BPQ that creates a multiport node on an IBM PC or clone. Very popular with BBS operators to provide multiconnect BBS services with several ports i.e. LAN and Backbone ports.

## bps

Bits per second. The rate at which binary data is transferred on a circuit. See also baud, data rate, transfer rate. Compare with: baud, the "signaling" rate on the data channel.

## breakout node

This is a node that is capable of handling many links. In many cases packet nodes have been installed in places where many radios or backbone links are not allowed, such as on high mountains of great commercial value. A breakout node holds no special meaning except it is a node that has proven to be very expandable and at which the node owner sees little or no limitations on reconfiguration.

## BSQ

A communications protocol for sending binary files over packet radio. (See *7-Plus*, *R95*)

## BTW

By The Way. An abbreviation sometimes seen in packet messages.

## callbook server

This is a network server whose function is to allow stations to access, in real time, callbook information. These servers are operated both stand-alone and as part of DxClusters. See the server listing in your Quarterly.

## carriage return

On a mechanical typewriter the motion of pushing the carriage, which held the paper, to the left all the way until it stops is called a carriage return. When electric typewriters came into existence there was a particular button on the keyboard, at the right hand end of the typewriter keys marked J, K, L, ;, ', which was called the return key or carriage return key. On modern computer keyboards that key still exists and is either left unmarked or is labeled enter, return, carriage return, or ø. The function of the carriage return key is to instruct the computer or remote device that the end of a line has been reached. On a terminal or computer emulating a terminal when the carriage return is pressed the binary number 00001101 is generated. The symbol used to describe a carriage return is sometimes ^M which means control M. This number is also represented as 0D hex or 13 decimal. (See *control character*)

## CCIR

Consultative Committee for International Radio. An international organization that sets standards for radio communication.

## CCITT

Consultative Committee for International Telephony and Telegraphy. A United Nations committee for telecommunications standards.

## cellular

Broken up into small areas, or cells. If a packet network is broken up into two kinds of links, user and backbone, only user links would be omni-directional. The coverage of a single user port might be thought of as a cell. The area covered by a cell would be best if there were a limited number of users in the cell. A suggested area of coverage is 50 total packet users or ten maximum on line at once. Thus the size of a cell varies depending on what area the cell is in. In rural northern Ontario a cell might be hundreds of miles across (if such coverage could be achieved) where as in Brooklyn a cell might be measured in hundreds of feet.

## chat node

See conference node.

## choke/unchoke

When a computer is unable to process data as fast a another computer is sending it the receiving compute may instruct the sending computer to stop sending th data. This condition is often referred to in the packe world as 'choke'. 'Unchoke' refers to the re-enabling o the sending computer.

Example:

- A TNC running TheNET is capable of storing a fixed number of packet frames in memory. If thi number gets exceeded data might be lost. Because each TheNET node is capable of supporting many users and some other network management functions simultaneously the memory is partitioned to smaller blocks called buffers. Each user on the node is allowed two buffers (in the recommended network). When those two buffers are used up the TheNET TNC attempts to choke the user. If the TheNET TNC is at the far end of a network circuit the choking and unchoking process takes somewhat longer. What actually happens is that when a message is passed across the network to the destination user port that destination user port may respond with a 'choke' message. The destination user port will attempt to deliver the data that is has and when it is ready will send an unchoke message back to the originating user port. Based on a time-out timer the originating user port might resend it's delayed packet even though it has not received the unchoke message.

## circuit

In a TheNET network a circuit is an assigned connection between two nodes. Each of the two nodes has information to the effect that the circuit exists. The two nodes also have a routing table from which the first element on the path to the other node may be realized but the two nodes do not know all of the intervening nodes. The circuit exists until the destination user or server or the originating user or server disconnects, or until one of the two nodes decides that data cannot be sent any more (due to L4 retry time-out or unchoke failure) or if no data is passed across the circuit during the time set by the no activity time-out. In the recommended network the L4 retry time-out is 5 minutes times 2 retries and the no activity time-out is two hours. (See *neighbor*, *choke/unchoke*)

## CLNS

Connectionless Network Service (See *connectionless*, *datagram*).

## CLOVER

An improved technique for packet communications on HF frequencies. Clover uses 4 audio data channels (like a four-leafed clover) and automatically adapts to changing propagation conditions. It uses a form of forward error correction (FEC) to improve efficiency. (See *FEC*, *AMTOR*, *PACTOR*)

## cluster

See PacketCluster, node stack.

## CMOS

Complementary Metal Oxide Semiconductor. A type of digital circuit made up of MOSFET transistors widely used in computers and other digital devices. CMOS circuitry generally uses very low power and can operate over a wide voltage range (typ 3-15v). Some versions are compatible with TTL circuitry. CMOS is also the name given to the memory that stores set-up configuration information in an IBM AT computer. (See *TTL*)

## collision

This is an event where a receiving station doesn't receive it's desired packet because another packet was generated by a different packet station in the same time frame as the desired packet and interference occurs. In this case the sending station will wait a backoff time and the packet is retried or a special poll packet is generated. (See *backoff*, *poll-packet*, *retry*)

## collision avoidance

Any technique that reduces the possibility of another collision on a retry transmission of a packet. (See *persistence*, *slottime*, *dwait*, *retry*)

## Commodore 64

Back in the late 70s, before everybody had their own computer, Commodore was one of the companies that cranked out a real cheapy for the purpose of grabbing a piece of the very lucrative extremely low budget computer market. They were successful. The Commodore 64 is the only one of those 1970s low budget computers that is still popular. Others include the TI99 and ZX81.

The Commodore 64 is a computer in a keyboard. The entire computer guts are located inside the plastic case which looks like an ordinary keyboard. Enclosed in the box are 64K of memory (equivalent to 64000 characters of text) which is relatively small compared to the 1000s of K of memory found in modern PC compatibles, Macintosh and Amiga models. The software for the Commodore 64 is usually built into accessories that plug into an expansion slot, or available on cassette tape.

The Commodore 64, unlike the Macintosh, Apple 2, and PC compatible, often uses a normal television set for a video display. Even with the optional (and rather expensive) computer monitor display the Commodore only gets about 40 characters of text across the screen and about 16 lines of text. This is as compared to 80 characters by 25 lines for the cheapest of PC compatibles, or to about 128 characters by 43 lines for more modern models. For packet radio 80 characters is the standard line length so 40 characters is a handicap.

There is a disk drive made for the Commodore 64. It actually sold for several times the cost of the computer! Since the Commodore 64 is often found as a throw-away or dug out of someone's basement the disk drives can be found for free as well. The highest reasonable price paid for a Commodore 64 including the disk drive is probably in the vicinity of $100. This is one of the things that makes the Commodore 64 an astoundingly popular amateur radio accessory.

## compressed forwarding

Forwarding between BBSs using a data compression technique to reduce the amount of data to be exchanged. This is becoming common between BBS servers.

## conference node

A specialized server that allows many users to connect and communicate with each other. Resembles a conference call on the telephone. Some node software like TheNET X-1J and NOS have built-in conference modes. Also called crowd or chat nodes. Conferences exist in single site and multi-site varieties. The CROWD node covered on page 34 of the Annual and the TALK feature of TheNET X-1J are of the single site variety. The NOS-chat that is growing out of Rochester NY is of the multi-site variety. DxClusters sport a chat feature that is multi-site, over multiple DxCluster sites.

Recently a system similar to the NOS-chat system has been springing up using commercial data-links between the sites. These cover areas that exceed many of the terrestrial amateur radio networks. There is a sensationalist aspect to the commercial data-link conferences because the people are at distances not found with true Amateur Radio networks.

## connectionless

refers to a packet protocol or service that does not have the concept of a "connection". Packets may be sent at will, without prior arrangement or need for connection setup/teardown procedures. An analogy would be the postal service where letters (datagrams) can be sent anywhere without prior arrangement.

## connection-oriented

refers to a protocol or service that requires that a logical or virtual "connection" first be established with a special procedure before data can be sent. Another procedure is used to "tear down " the connection when it is no longer needed. The amateur packet AX.25 level 2 protocol is connection oriented. An analogy would be the telephone system where you must callup (connect to) another phone before talking and hang up (tear down) the connection when you are finished.

## CONS

Connection Oriented Network Service (See *connection-oriented, virtual circuit*).

## contention

Contention is when 2 stations on a channel want to transmit at the same time. If they do, it may produce a collision. (See *collision, HTS*)

## control character

On all keyboards which meet the American Standard Code for Information Interchange (ASCII) standard there is a key marked as Ctl, Cntrl, Ctrl or Control. This key is a shift key. That means that when that key is pressed the values associated with other keys are changed. Now when a normal alphabet key is pressed the control value of that key is sent, instead of the normal alphabet value. The control value is called the control character. Example: If the control key is pressed and an A is pressed the keyboard (or computer) will generate a control A. This is also described in text as a ^A. There are actually 32 control characters, ^A through ^Z and six others. The control characters are interpreted by computers as ASCII defined functions. For instance, ^M is the same as a carriage return. ^C is used by most computers to mean stop, or go to command override mode. (See *carriage return*)

## converse mode

Conversation made. The normal connected mode of operation of a packet TNC from the keyboard. Certain characters are used for control purposes and must be avoided in converse mode e.g. CR (carriage return or enter) will send a packet. Compare with: Transparent mode where all binary or hex characters are sent without regard for commands.

## converse node

See CROWD, conference node

## COR

Carrier Operated Relay. A circuit that closes a relay when a radio carrier is detected. This is how a voice repeater knows to turn on its transmitter when it receives a signal on its input frequency.

## coverage

The area in which a signal can be heard is that signal's coverage. Usually coverage refers to both transmit and receive. If a station can both talk to and receive from a station in a particular area that station has coverage in that particular area. Wide area coverage means that the station can be heard and can hear a wide area. If a station can be heard in an area that it can't hear from then there are problems. Certain derogatory words have been used to describe those problems, like alligator (mouth, no ears).

## CPU

Central Processing Unit. The "smart" IC that does all the calculations and data manipulations in a computer or computing device.

## CRC

Cyclic Redundancy Check. The error checking procedure that verifies a packet. (See *FCS*)

## CROWD

This is the name given by the NEDA founders for a piece of software written by NORD><LINK to run in a TheNET network. Most NORD><LINK documentation refers to this is a mini-conf (conference) node. The CROWD software is installed at a TNC. Access is over the network only, through the serial port in the CROWD TNC from other TNCs at the same node site. (See *conference node*)

## CSMA

Carrier Sense, Multiple Access: This is a system of packet operation that requires that all stations wait for the channel to be quite before transmitting. Most amateur radio packet uses CSMA.

## CSMA/CA

Carrier Sense, Multiple Access with Collision Avoidance A CSMA system with a method to reduce repeated collisions between two stations trying to transmit at the same time. One example of collision avoidance is the FRACK/DWAIT method where a random time delay passes before a packet transmission is retried. (See *persistence, polling*)

## CSMA/CD

Carrier Sense, Multiple Access with Collision Detection This is reference to the system used in AX.25 amateur packet system that detects collisions by using the sequence numbers that accompany each packet and ACK to detect missing frames.

## CTS

Clear to Send. A control line on a RS-232 port (pin 5 on DB-25) which indicates that the device is turned on and ready to receive data. Compare with: RTS or DTR which is the line that indicates that a device is got data ready to send. (See *RS-232, RTS, DTR*)

## data

The numerical representation of information. Usually expressed as binary bits or hexadecimal bytes.

## data collision

See Collision.

## DataEngine

A TNC manufactured and marketed by Kantronics. The DataEngine has two HDLC radio ports and one serial port. The modem are plugged in and are available up to 19.2 Kbps.

## data rate

The basic rate at which data is transferred on a circuit. Often referred to as "baud rate" (which is like saying speed speed) but more correctly should be bits per second or bps. (See *baud, bps*)

## data set

A telephone industry name for a modem.

## datagram

Information packets in a connectionless environment. Datagrams are completely self-contained as far as the network is concerned. The information needed to get each datagram to its destination (including, but not limited to, full source and destination addresses) is carried in each datagram. An analogy would be a letter sent in the postal system. The envelope contains all of the information (name and address) necessary to deliver the letter (datagram).

## DB-9 connector

A common but incorrect name for a DE-9 connector. The second letter refers to the size of the connector shell - a B size is used for 25 pin connectors.

## DB-25 connector

The 25 pin subminiature D shaped connector used on many digital devices as a RS-232 (serial) port. (See *DE-9 connector*)

## DCD

Data Carrier Detect. DCD is a signal indicating the presence of data on the communications channel. The TNC uses the DCD signal to hold-off the transmitter when the channel is occupied. Some TNCs and modems employ a "DCD" that simply indicates the presence of a RF carrier or noise energy on the channel rather than true Data Carrier Detect. (See *hold-off, modem, TNC*)

## DCD hold-off

See hold-off.

## DCE

Data Communications Equipment. Usually refers to a modem or any equipment that receives data from a DTE (terminal or computer).

## DE-9 connector

The 9 pin sub-miniature D shaped connector used on IBM AT computers and many other digital devices as a serial port connector. Often called (wrongly) a DB-9 connector. (See *DB-25 connector*)

## dedicated link

A point-to-point link between two dedicated ports for the exclusive use of those ports or nodes or for the use of stations passing through the network, using those two nodes. (See *HTS, backbone, point to point link*)

## dedicated port

This is a port designated for a specific purpose with only one other station on the frequency, usually a tie-in to a server or other network hardware. Other stations who would connect through the network might pass across the link that uses the dedicated port. No third station would access the dedicated port on the dedicated port's frequency. (See *point to point link, HTS*)

## defined neighbor

Another name for NETROM/TheNET's locked route. (See *selective neighbor routing, point to point link, HTS*)

## deviation

The deviation of a FM radio is the maximum change or shift in the carrier frequency during modulation. It is usually expressed as peak deviation in kilohertz.

## DFD-PBBS

An amateur packet BBS program written by Joe N3DFD.

## diddle

slang for the shifting of an AFSK signal back and forth.

## digi

See digipeater.

## Digicom>64

A software and modem package designed to emulate a TNC on a Commodore 64 computer. The software was written by DL2MDL and others while the modem was designed by Willy YV1AQE. The software and modem info is distributed in North America by Barry W2UP and is described in 73 Magazine Aug 88.

## digipeater

A TNC used for relaying messages on a single frequency. Digipeater functionality is built into all user TNC software. A digipeater is used for sending a message beyond the range of a user station. Under conditions where networking is not available two stations that want to communicate beyond their own range may use a digipeater in between. A digipeaters does not recognize when a message it has relayed does not get through. It is up to the sending station to retransmit. Digipeaters are inherently susceptible to hidden transmitter syndrome. NEDA recommends against digipeating in any form except in emergencies. (See *simplex digipeater, duplex digipeater, store-and-forward, real time, repeater*)

## digital signal

An electrical signal that changes in discrete unambiguous steps each representing a numerical data value, or logic state.

## diode matrix

The TNCs running ROSE or TheNET network software can communicate to each other over their RS-232 ports. If two TNCs are used at a node site the connections are simple connector to connector wire connections. If more than two TNCs are used a diode matrix is required. (See *HexiPus*™)

## discriminator

In an FM radio the discriminator is the circuit that derives audio from the IF signal. After the discriminator the FM receiver will change the audio to remove hiss. One of the techniques for improving the bandwidth performance of a typical two way voice radio, in order to make 4800 or 9600 baud modems work with radios that weren't designed for it, is to connect the modem's receive audio line directly into the discriminator circuit. How this is done depends on the radio.

## disk drive

A disk drive is a computer accessory that stores data in the form of magnetic impulses on a flat media that is much like an extremely high quality magnetic recording tape. The media is on a spinning platter, like a phonograph, and is read and written by a tape head that moves across the media, like a phonograph. The media is of two basic qualities and is called floppy or hard. Floppy media is low tolerance, meaning that if the media is dented or dirty it should still work. Floppy media can store about 200,000 characters of data per square inch. Most floppies store less than that. The rate at which data can be read or written to a floppy drive is usually less than 30,000 characters of data per second. Floppy media is always exchangeable. You can change out the floppy in the disk drive quickly. Floppy media is commonly used for transfer of information between computers and for storage of seldom used programs. On really low budget computer systems floppy media may be the only non-volatile (not erased when power is removed) memory that the computer has. Floppies are commonly seen in two different sizes, 3.5" and 5.25". 3.5" floppy media is sealed inside a plastic cartridge which is flat and square. There is a little access door which covers the media when it is not inserted into the floppy drive. This is commonly called a 3.5" floppy disk. Media sizes depend on the way the data is stored on the floppy disk and usually range from 360K (360,000) to 1.44M (1,440,000) characters. 5.25" floppies are jacketed by a thin plastic cover which doesn't entirely cover the magnetic media. They are commonly stored with a paper slip over the exposed parts. Media sizes range from 80K to 1.2Mbytes.

Hard media is high tolerance. Getting dirt on the media will destroy it. Normally (although not 100% the case) hard media is sealed into the disk drive with the read/write tape head. Hard media is commonly used for storage of a computer's operating system, applications programs, and temporary data. Hard media can store as much as 1,000,000,000 characters of data per square inch. Access time to that data can be as fast as a hundredth of a second and the data can be read off the disk at millions of characters per second. In order to get that much data on so small an area the recording surface has to be very flat and in order to get data on and off the disk so fast the disk has to spin very fast.

## DOERS

Digital Operators Emergency Radio Service. An active packet network group in Plattsburgh, N.Y.

## dogbone

Two diode matrices connected together with a wireline link. (See *diode matrix*, *wireline link*)

## DOVE

An OSCAR satellite (OSCAR 17) whose full name is Digital Orbiting Voice Encoder.

## downlink

A circuit from a node to a user, initiated by the node on command from a distant user.

## DRSI

Digital Radio Systems Inc. Best known for a line of PC plug-in TNC cards.

## DSP

Digital Signal Processing. A modern technique of analyzing analog signals by converting the analog signal to a digital form and processing it with a specialized computer circuit.

## DTE

Data Terminal Equipment. Usually refers to a terminal or computer or any equipment that generates data.

## DTR

Data Terminal Ready. One of the RS-232 signals (pin 20 on DB-25) that indicates that the computer or terminal is ready to send data. It is similar and sometimes interchangeable with RTS (ready to send).

## dumb terminal

An ASCII terminal with video display and keyboard that can send and receive ASCII text but cannot do any computational operations.

## duplex

Duplex means two channel. A full duplex signal consists of two separate channels. Both ends of the radio circuit need to have a separate receiver and transmitter such that the receiver on each end can here the opposite station's transmitter regardless of the state of it's own transmitter. A half duplex signal consists of two separate channels also, but one end of the circuit does not receive when in transmit mode. A voice user operating a repeater is usually in half duplex. (See *repeater*)

## duplex digipeater

like a simplex digipeater, except that different receive and transmit frequencies are used. Compare with: Full duplex real-time repeater which repeats received data at exactly the same time. (See *digipeater*, *store-and-forward*, *real-time*)

## DWAIT

Digipeat WAIT. A delay in sending a packet automatically inserted by a TNC when originating a packet. The delay starts when a packet is ready to be sent, after the channel becomes clear. A digipeated packet is sent without waiting this delay. Used as a collision avoidance system when digipeaters are in use. (See *persistence*)

## DxCluster

A server used by HF operators to pass information about contacts. This software, originally written by AK1A, also operates as a database of HF related information. One key feature of the DxCluster software is that DxClusters may share contact information (called Dx Spots) in real time. That is that one station connected to one DxCluster may introduce a Dx Spot report which will then be shared by all of the stations connected to all of the DxClusters which are networked together. See the server listing in your NEDA Quarterly. (See *PacketCluster*)

## dynamic rerouting

In a network where redundancy exists in the backbones from one city to another some types of network software allow for the network to recover automatically from a backbone hardware failure by rerouting traffic through the redundant link. This is called 'dynamic rerouting' as it can adjust dynamically to a changing network. (See *ROSE*, *TCP/IP*)

## envelope delay distortion

See group delay distortion.

## EOC

Emergency Operations Center. This is a term used by some state governments for a state or county government owned facility at which emergency services radio equipment and other gear is clustered. Some E.O.C. facilities are located in bomb shelters, others in buildings as part of a state office complex. A good packet network would include an on line TNC or BBS at every E.O.C. in the network's region. (See *OEM*)

## EPROM

Erasable Programmable Read Only Memory. This is an integrated circuit (IC) which is used in computers, including TNCs, to permanently hold a computer program. In PC compatibles and Macintoshes EPROMs are used to hold the boot program. That's the program which is responsible for loading the operating system into the computer from a hard disk or floppy disk. In TNCs all of the program is located in one EPROM. EPROMs are erasable using ultraviolet light for between 2 and 40 minutes. Thus EPROMs have a small lens in the middle of the top which exposes the internal electronics. During long term usage EPROMs are covered with a piece of opaque tape. EPROMs cost about 50¢ surplus and about $5.00 new. The EPROM used for TheNET is called a 27C256 or a 27256 (either is OK). EPROMs can be programmed using a peripheral to a PC called an EPROM programmer which costs about $150 from JDR Microdevices @1-800-538-5000.

## ERS

Exposed Receiver Syndrome: This is a condition where a packet station, be it node or user, is unable to transmit due to the fact that it perceives the channel as being active continuously. This can be caused by Hidden Transmitter Syndrome and is often the case when a node is located on a high hill with surrounding metro areas. (See *HTS*, *CSMA*) Also see an article in the *Compendium* called *The Exposed Receiver Syndrome*.

## eye pattern

A pattern produced on an oscilloscope to show jitter and phase distortion of a data signal transmitted on a communication channel or through a particular component such as a modem. The eye pattern is produced by triggering the scope with the original untransmitted data signal and displaying the received data stream on the horizontal trace.

## false route

In a network using TheNET software the node routing is generated automatically by the nodes themselves. If improperly managed it is quite possible for routing to be discovered and used by the nodes such that Dx propagation paths are treated as real paths. In this case a route may be created in the routing table that depends on a 'lift' (propagation enhancement) condition. When the lift goes away the nodes will be helplessly trying the 'false route'. This condition is preventable in a TheNET system by manually controlling the route tables to specify valid routes to neighbor nodes. This situation cannot occur with ROSE or TCP/IP software as all neighbor nodes and routing information is created manually. (See *TheNET*, *locked route*, *ROSE*, *TCP/IP*, *neighbor*).

## FBBS also FBB BBS

An increasingly popular amateur BBS software written by Jean-Paul F6FBB and others.

## FCS

Frame Check Sequence. A 16 bit (2 byte) number included with each frame in the packet used for error checking.

## FEC

Forward Error Correction. A technique of error correction in which packets or AMTOR groups combine the data from two or more transmissions to yield less errors. In AMTOR FEC mode, the data is sent twice and the receiving station(s) record all known characters without resorting to an ARQ ACK/NAK transmission. (See *AMTOR, ACK, ARQ*)

## F.E.M.A.

Federal Emergency Management Agency. These are the guys with megabucks to spend on your networking project. FEMA is very interested in seeing a non-government owned and maintained network of interstate packet equipment. Don't wait for F.E.M.A. money but F.E.M.A. can make life easier if they are impressed with what we can do with networks. They've been known to drop words on other government agencies like forest departments etc.

## firmware

Software stored permanently in a ROM or EPROM IC.

## flag

A data character (01111110) used to delimit packets (beginning and end) and to separate multiple frames in one packet transmission. The same character is often used during the TxDELAY to help synchronize the TNC receiver circuits at the beginning of packets. Should 6 binary ones occur together in the text of the packet, a process called bit-stuffing is used to modify the character so that it can not be confused with a flag character. See also bit-stuffing, TxDELAY. In computer terminology, a flag is a single status bit included in an information field. For example, in a BBS program a number of flags are used to indicate whether a message has been read, forwarded, killed, old, etc.

## flat network

A flat network is a system of dual port nodes where one of the ports is a backbone port. All backbone ports in the network are on the same frequency and may not be hidden transmitter free. It is said to be flat because the network topology is flat. Flat backbone ports are defined here so a node builder constructing a system per these specifications can have an access to or from an existing flat network. A flat backbone port is not on 2 meters or HF. This port specification is also used where the other stations on frequency are not able to operate with respect given to Hidden Transmitter Syndrome or locked routes and connect disable are not used on a adjacent nodes. (See selective neighbor routing in Ne working Around HTS in the *TheNET Network Noc Guide* )

## floppy disk

See disk drive

## flow control

The control of data flow on a communications circuit b software (XON/XOFF) or hardware means (RTS/CTS (See *handshaking, RTS, CTS, XON/XOFF*).

## FM

Frequency Modulation. This is a method of transfe ring data or voice information over a carrier signal. I packet radio the carrier signal is commonly a radio wav FM is done by changing the frequency of the carrier i proportion to the wave form (audio or digital). Th amount of change is called deviation and is usuall around 3 to 5 KHz for a typical voice radio.

## forwarding

The transferring of messages between BBSs. (See *au toforward*)

## forward file

This is the disk file on a packet bulletin board syster (PBBS) that is responsible for directing the autoforwarc ing operation. By making entries in this file the PBB system may select what packet paths are used to eac PBBS that is forwarded to, when each operation is pe formed and what traffic is sent during each piece of th forwarding operation. (See *autoforward*)

## forwarding frequency

A channel or frequency used exclusively for transferrin messages between BBSs.

## FRACK

FRame ACKnowledge delay: This is the time after packet is transmitted by a TNC before the TNC decide that a frame acknowledge is not going to occur. At tha point the TNC performs backoff (some TNCs + TCP/IF and a retry. FRACK is calculated based on the num ber of digipeaters that you specify in your connect com mand. (See *backoff, retry*).

## frame

A frame, or packet frame, is a single packet. Severa frames may be sent in one transmission but each ha it's own address, start and stop.

## FTP

File Transport Protocol. This is a part of TCP/IP whicl allows a user of a TCP/IP host to request or send file from other TCP/IP hosts.

## FSK

Frequency Shift Keying. A method of digital modulation where the carrier is switched between two distinct frequencies. This is the technique used on HF packet.

## full-duplex

Communications in which reception and transmission take place at the same time. In radio this means that transmission and reception are on two separate channels. (See *duplex, half-duplex, repeater*)

## G3RUH modem

A 9600 bps plug-in modem for TNC-2s and other amateur TNCs. Circuitry contains adaptable filters to adjust for bandwidth limitations in commercial radios and a "randomizer" circuit to prevent DC offsets on modulated data. Similar to but may not be totally compatible with K9NG modem. Believed to be compatible with the new TAPR 9600 bps modem. (See *modem, K9NG modem, TAPR, HAPN*)

## G8BPQ Code

John Wiseman, G8BPQ, developed a TSR (terminate stay resident) program for the IBM PC and compatibles that would imitate TheNET and allow node access for a program that runs on the PC. This program simulates the TheNET node functionality and allows routing from a TheNET system directly to the PBBS or other program running on the PC. Unlike a TheNET node which can only handle one radio per TNC, the G8BPQ program may direct traffic in and out of several radios by using KISS TNCs or other TNC/modem cards. The normal use for this program is to allow a PC running a server to communicate via a wire to an on-site packet node by connecting the RS232 port of the PC to the matrix. This connection involves the use of a few components to buffer the RS232 lines before they connect to the matrix. See G8BPQ v4.06 documentation or the article in the NEDA Quarterly Compendium section of this Annual. (See *KISS, dedicated port, locked node, wireline, NET/ROM, TheNET*)

## gateway

A configuration of nodes where connectability is available by deliberate manipulation but where automatic end-to-end routing is not possible. This is useful for connecting two networks together such that users and servers on one network can access users and servers on the other network without compromising networking practices on either network.

Examples:

- To access packet radio from Fred's telephone packet gateway I can phone up and use a password. After Fred's machine accepts the password I can use my callsign on Fred's PC.
- To gateway into TCP/IP in Seattle from Portland I can use the TheNET network by connecting into OARS which is a PC running NOS. Once I get connected I can use the TELNET program to access another TCPer. OARS is a gateway.

A very general term for anything that connects two networks together especially two networks with different protocols. For example between a VHF 1200 bps network and an HF APLINK station. In the ARPA world, "gateway" has a much more specific meaning: a packet switch that handles IP datagrams.

The name of a packet newsletter published by ARRL and now included in QEX magazine.

## group delay distortion

Also called envelope delay distortion. A distortion of the data signal produced when the different frequency spectral components of the digital signal are phase shifted by different amounts resulting in a distorted pulse shape. For best results, it is important that radio filters, amplifiers and other components in the communication system have a constant phase shift across their bandwidth. This is called "flat group delay" characteristic.

## half-duplex

Communications in a duplex system arranged to permit operation in either direction but not in both directions at the same time. (See *duplex, full-duplex, digipeater, simplex, simplex digipeater, semi-duplex*)

## handshake

The exchange of data or signals between a sending and receiving devices that ensure that the circuit is ready for communications. On RS-232 linked devices, the RTS (or DTR) and CTS lines are used for hardware handshaking or Control-S/Control-Q characters for software handshaking. On packet radio circuits, supervisory bytes in the packet and ACK and other control frames are used for handshaking. (See *RTS, CTS, DTR, RS-232*)

## HAPN

Hamilton Area Packet Network. An active amateur packet group in Hamilton, ON best known for the development and marketing of a 4800 bps modem (for installation in TNC-2 or clones) and plug-in TNC cards for PCs. For more information, contact: HAPN, 5193 White Church Road, Mount Hope, ONT L0R 1W0. (See *HAPN-T modem*)

## HAPN-T modem

A 4800 bps add-on modem card for TNC-2 (and clone) TNCs which uses bipolar pulse modulation and can be used with most any radio transceiver. (See HAPN)

## hard disk

See disk drive

## HDLC

High-level Data Link Control. The ISO level 2 link level protocol on which AX.25 was based.

## header

The R: lines attached to a BBS message that indicate the forwarding path that the message has taken. The header line lines contain the date and time, message number, and hierarchical address of each BBS that handled the message. Header may also describe the TO, From and @BBS address fields of the message as well as the subject line information of the message.

## heard list

On several different packet devices, including user TNCs, BBSs, nodes etc.. there exists a feature whereby a list of stations heard are recorded. That list is called a heard list. Access to the list is different depending on the application. You can look for it though, now that you know that it exists. Often the list is called MHEARD or JHEARD (no clue as to why the J is used. M means monitor).

## hex

Hexadecimal. Numbers to the base 16 (0-9,A-F).

## HEX9

An Amateur packet group based north of Toronto in the Barrie, ON area.

## HexiPus™

Six way diode matrix card: This was a product of the North East Digital Association (developed by WB2JLR) that allows up to six TNCs to communicate via RS-232. This was used in TheNET multiport nodes such that up to 6 radios may be installed at a single network node site. Diode matrix boards are now sold (cheaper) by NX2P Electronics and A.N.S. (See *wireline link, diode matrix*)

## hidden transmitter syndrome (HTS)

A condition in a CSMA packet system where several stations who cannot hear each other (they are HIDDEN TRANSMITTERS) are allowed to transmit at the same time and thus cause collisions thereby reducing throughput on the channel. Compare with: ERS exposed receiver syndrome, which is the inhibiting of transmit at a high node site due to other nodes on the same channel at a distance. They may not be strong enough to trash the local packet but reduce the throughput due to the delayed response. (See *CSMA, collision, throughput*)

## hierarchical address

A geographical addressing system for stations involved in the amateur BBS system. The hierarchical address is a series of abbreviations of increasingly larger geographical areas (each abbreviation is separated by periods) within which the station resides. For example, the hierarchical address of a major Montreal BBS is VE2FKB. #MTL.PQ.CAN.NA which in effect says that VE2FKB is at Montreal which lies within Province of Quebec which lies within Canada which lies within North America. The state/province abbreviations used are those normally accepted by the postal system. National abbreviations are set by international agreement.

## hierarchical routing

A BBS forwarding technique using the hierarchical address as guidance.

## hold-off

The process by which a TNC delays transmitting until the DCD indicates the channel is clear. (See *DCD, collision avoidance*)

## host

The computer or terminal attached to a TheNET node when operating in host mode for sysop entry to the serial port. Host is also the name given the computer that controls a TCP/IP or Internet node. (See *host mode*)

## host mode

WB8DED created a software package for the TAPR TNC 1 that was called Host Mode. This package was later created for TNC 2. Some BBS programs took advantage of the command language in Host Mode to control the TNC and to allow multiple users to connect to the BBS at the same time. AA4RE BBS may have been the first software to use this feature. TheNET incorporates a very small subset of the Host Mode command set.

Host Mode is used to refer to the condition where a node has a CRT terminal or computer plugged into it that will be used in ASCII mode (not using networking protocol).

## HTF

Hidden transmitter free. (See *HTS free*)

## HTS

Hidden Transmitter Syndrome: This describes a condition where throughput is drastically reduced to well below the specified baud rate because a single station is able to hear two or more stations that can't hear each other. (See *throughput, hidden transmitter syndrome*)

## HTS free

By making sure that every radio/TNC on a frequency can hear every other radio/TNC most of the collision problems and inherent loss of throughput may be removed. At this point backoff becomes effective and the performance of the system of radio/TNCs may be predicted more accurately. The only remaining problems occur when radio dead time due to slow transmit receive switching is excessive. Also backoff must be used if there are more than two radio/TNC sets on frequency. (See *backoff*)

## IF

Intermediate Frequency. The function of a radio receiver is to convert a radio wave, which has audio on it, to audio information to be played into a speaker or into a packet modem. The process is often done in two steps. First the radio signal (RF or radio frequency) is converted from the tuned frequency to a known constant frequency. This is done by mixing the incoming signal with the VFO or synthesizer. This known constant frequency is the IF. Next the IF is converted into audio via a discriminator (in the case of FM)

## IF Bandwidth

Intermediate Frequency bandwidth. The width of the band of signals that can pass easily through the intermediate frequency stage of a superheterodyne receiver. Signals wider than this bandwidth become severely distorted.

## image

See ROM image.

## Internet

The Internet is a public system of computers which communicate over commercial lines (usually telephone or leased telephone lines) using TCP/IP. Usage of the Internet network is free. Usage of the computers that are hooked to the Internet is not necessarily free. Most people who have access to the Internet either pay a fee or have connection to the network from work or school. There is a book on the subject called Internetworking with TCP/IP by Douglas Comer and published by Prentice Hall. You should read this if you are interested in the details.

## IP

Internet Protocol. The core protocol of the ARPA suite. IP is a simple connectionless (datagram) protocol that handles addressing, fragmentation and type-of-service routing in the heterogeneous internetwork environment. (See *TCP/IP*, *KA9G*, *ARPA Suite*, *connectionless*)

## IS

Intermediate System. ISO's term for a packet switch.

## ISO

International Standards Organization. ISO formulates and publishes specifications for everything from screw threads to computer communication protocols. (See *OSI*)

## jitter

variations in the phase or amplitude of a data modulated signal having no relationship to the data. On amateur packet signals, phase jitter may cause errors in decoding the data.

## JNOS

A version of KA9Q NOS written by WG7J that combines a BBS, node, and conference server.

## K9NG

A 9600 bps modem designed by Steve K9NG and marketed by TAPR. It uses the same modulation technique as the G3RUH modem but does not have the adaptive filters. The "randomizer" circuit from the K9NG modem is used in the G3RUH. It may be compatible with the G3RUH in some circumstances but not guaranteed. It has been replaced by the new 9600 bps modem from TAPR. (See *modem*, *G3RUH modem*, *TAPR*, *HAPN-T modem*)

## KA9Q Internet

Original name for TCP/IP or NOS amateur packet software. A C software package developed by Phil Karn KA9Q and others. Implements the major elements of the ARPA protocol suite: IP, ICMP, TCP, UDP, Telnet, FTP, SMTP and ARP. Also implements subnetwork drivers for SLIP, KISS, AX.25, Ethernet and Appletalk. Primary environment is the IBM PC (and clones), but has been rewritten for 68K-based machines like the Commodore Amiga and Apple Macintosh, also to UNIX system 5 environments. (See *NOS*, *TCP/IP*, *Internet*, *ARPA Suite*)

## Kantronics

Kantronics designs, manufactures and markets a range of amateur packet products including the popular KPC TNCs and KAM multimode controller.

## keyboard-to-keyboard

Communications between two packet user stations in real time. Tends to be rather slow unless both operators are fast typists.

## kill

The process of deleting a message from a BBS after reading it or for other reasons.

## KISS

Keep It Simple, Stupid. A TNC operating mode where the TNC merely translates packets between half duplex, synchronous HDLC on the radio port and full duplex asynchronous SLIP framing on the host port; the host computer must implement all higher level protocols, including AX.25 if it is used. Gives the host computer full access to and control over all fields in each packet. Compensates for the lack of a HDLC hardware controller on many computers. The KISS TNC is only responsible for TX delay and DCD hold-off. Kiss is used with TCP/IP hosts and also often used with BPQ and other such PC based packet switches. (See *TCP/IP*)

## LAN

Local Area Network: NEDA defines a LAN as a user access node and a group of users. Servers do not communicate with the network on the LAN frequency but use dedicated access frequencies. LAN users which are home stations running minimum antenna and power configurations to access the node may access multiple servers through the network via the local access node.

Example:

- A sample network node setup would include a user port that can hear twenty or less active packeteers. No other user ports and no servers share the frequency within the range of the user port radio. The node setup must included one or more backbone links to other nodes. If any servers exist in the area that need network access then dedicated link radios and TNCs are added to the node stack. Users may access those servers (if any user services are supplied) from the LAN frequency by using TheNET, ROSE or other networking technology.

(See *local area network*)

## LAPB

Link Access Procedure, Balanced. AX.25 is based on the Balanced Link Access Procedure of the CCITT X.25 standard. LAPB in turn conforms to the HDLC standard.

## Lan-Link

A shareware terminal program written and supported by Joe, G3ZCZ. (See also *YAPP*, *Paket*)

## link layer

Level 2 in the 7 layer OSI computer communications protocol set. AX.25 is the amateur packet level 2 protocol.

## local area network

A digital network channel covering a small geographical area or serving a limited number of users. In the case of amateur packet, some suggest that a LAN should not cover an amateur population of more than 100-200 amateurs and should not have more than one high volume data generator (node or server) on the same channel to control congestion. A properly configured LAN will have a minimum of hidden transmitter syndrome problems. Compare with: wide area network, which is a network covering a large geographical area and often directly linked to other networks on the same channel. (See *LAN*, *wide area network*, *HTS*, *ERS*).

## locked node

TheNET nodes have the capacity to generate routing lists automatically based on parameters set in the node's RAM. The parameters specify default quality values to be assigned to routes to each neighbor, separately defined for radio port neighbors and RS-232 port neighbors. If a neighbor tells me (I'm a node now) about a certain node, by callsign and node name and quality, I'll remember it for a duration that is also setable by the parameters in RAM. If that duration ends and I haven't gotten a refresh on that information, I'll forget about the node. A locked node is where an individual node is given a specific quality and nearest neighbor route, but for which the duration is set to infinite. This locked node must be manually entered by a sysop but is visible to anybody who wants to look for it by doing a N NODE NAME for any nodes that are suspected as being locked.

A locked node may be used to make sure that a node listing doesn't go away, to set a node in with a route that is different than which may be automatically generated, to set a node in with a quality which is less than that which is automatically generated, or to make a node not show up at all. The last is difficult to do and no longer necessary with the advent of the ROUTE 2 command. (See Sysop Commands, part of the *TheNET Network Node Guide* in this Document)

## locked route

TheNET nodes have the capacity to generate routing lists automatically based on parameters set in the node's RAM. The parameters specify default quality values to be assigned to routes to each neighbor, separately defined for radio port neighbors and RS-232 port neighbors. It is possible using the sysop's ROUTE command to manually set a route at a specified quality. This may be done so that the routes never change due to weather induced radio performance effects, due to accident or due to malicious intent by another party.

## loopback

A test in which the output of a modem modulator or other full-duplex digital device is looped back to the input of the demodulator or device. The looped back signal may be either analog or digital.

## Macintosh

A personal computer which is a product of Apple Corporation. A Macintosh is a windows oriented computer which is not compatible with IBM's PC. The Macintosh is best known for being simple to learn and very visual oriented. Most programs available for amateur radio are written for IBM PCs and so the Macintosh is not as popular for hams as the IBM PC compatible. On the other hand the programs that are available for the Macintosh tend to be so much easier for a non computer person to learn that Macintosh computers have a larger user base in ham radio than would seem likely by mere utility. (See *PC*).

## mail box

A personal BBS in a TNC. Sometimes also refers to any BBS system handling personal mail. (See *Mail Drop*, *PMS*, *Personal BBS*)

### mail drop

A part of a TNC program that allows messages to be loaded into the TNC and then retrieved from over the air or from the terminal at the TNC. Mail Drop is what AEA calls this function. It is also called PMS or Personal Message System, by PacComm.

### mark

One of the two possible binary states in a data communications system. The mark is the resting state in an asynchronous serial system. The negative voltage state on a RS-232 port is called mark. One of the two tones in an AFSK modulation is mark. Compare with: space, which is the opposite state.

### matrix

Matrix = diode matrix: The TNCs running ROSE or TheNET network software can communicate to each other over their RS-232 ports. If two TNCs are used at a node site the connections are simple connector to connector wire connections. If more than two TNCs are used a diode matrix is required. (See *HexiPus™*, *diode matrix*)

### matrix monitor

Communications between TheNET TNCs via the RS-232 port or over a matrix is not in a textual format that is readable by a dumb terminal or protocol analyzer. A Matrix Monitor is a hardware or software device that can display the data passing across the matrix in a form that is legible and informative. G8BPQ code includes a program that can observe TheNET communications over the matrix. KA2DEW developed a crude single board computer with this capability also but the product was never made reproducible. (Anybody want a good project?)

### MBL BBS

An amateur BBS program written by Jeff WA7MBL.

### MFJ

MFJ Enterprises Inc. MFJ designs, manufactures and markets a range of amateur packet products such as the popular MFJ-1270 TNC (TAPR TNC-2 clone) used for network building as well as other amateur related items.

### MID

Message ID. An identification number given to all messages entered on a BBS system. For bulletins it is the same as the BID. (See *BID*)

### MIR

Russian (Soviet) space station whose cosmonauts regularly use packet radio to communicate with amateurs around the world. The callsigns used all end in MIR.

### mnemonic

See alias.

### MP-Net

Montreal Packet Net. A group of amateurs from Montreal, Quebec who were responsible for the very first amateur packet radio broadcast on May 31, 1978.

### modem

MOdulator/DEModulator. A device which takes the data modulated signal (RF, audio or pulse) received from the communications channel and restores the data to a form that can be used by the CPU in a computing device (or that can be read on a terminal). At the same time it converts data from the computing device to a form that can be transmitted out on the communications channel. Most TNCs contain an internal 1200 bps modem and have space to plug-in another modem for a different radio data speed. (See *TNC*, *G3RUH modem*, *K9NG modem*, *HAPN-T modem*)

### modem header

The connector inside a TNC used to connect an external modem to the TNC for higher speed or different mode of communication. All the data and control lines from the CPU/SIO to the modem pass through this connector.

### MSYS

An amateur packet BBS software written by WA8BXN. Versions also contain DxCluster and conference node modes. MSYS has support for TheNET routing and emulates a TheNET nodes. Note that MSYS's simulation of TheNET is limited by the performance of the PC and may be very poor when the PC is doing BBS operations. The MSYS TheNET emulation features are best utilized as a way to pipe user traffic in and out of the board via the TheNET style node mnemonic.

### MTL/WQC Network

A network of nodes in Montreal Area dedicated to linking Montreal, south eastern Ontario and northern NY/VT.

### multistreaming

This is the process by which a user can connects to several stations at once. (See *stream*)

### NAK

Negative AcKnowledgement. A packet or AMTOR ACK response that indicates that the data was NOT received correctly. (See *AMTOR*, *ACK*, *handshake*)

### NAPRA

Northwest Amateur Packet Radio Association: This is an Association founded in 1983 to promote packet radio in the north west region of the United States.

### NBOD

NEDA Board Of Directors: This is the term used to refer to the board of directors and appointees of NEDA.

## NEDA

The North East Digital Association. This is an Association that was formed in the fall of 1989 to support and instigate packet network development in the north east region of the US and Canada. NEDA is a club that was formed in 1989 to support and promote efficient packet network development. Development of network concepts and standards as well as the writing and publication of up-to-date network information (at all levels of access, operation, creation, installation, and maintenance) and general packet information are important goals of NEDA. Membership is open to all interested persons. The contact address is Box 563 Manchester NH 03105, or via packet is NEDA @ WB2QBQ.ny.

## Neighbor

In a network of nodes the neighbor of a node is any node that is talked to directly on either local wire (RS-232 for instance) or via the radio through a backbone.

Example:

- If a linked system consists of FRED <-> BOB <-> ED <-> LEFT <-> RIGHT then the neighbors of ED are BOB and LEFT

## NEPRA

New England Packet Radio Association. An extinct amateur packet group based in the New England/Boston area.

## NET/ROM

A proprietary product of Software 2000, Inc (WA8DED and W6IXU). Consists of ROM firmware for the TNC-2. Implements AX-25 at the link layer (L2), with ad-hoc protocols at the network (L3) and transport layer (L4). Also provides a command interpreter and "transport level bridge" that patches incoming or outgoing regular AX.25 level 2 connections to internal transport layer connections. Uses datagrams at the network layer, virtual circuit at the transport layer. Provides automatic routing between NET/ROM nodes but the user is still responsible for "source routing" between the end NET/ROM nodes and the ultimate source and destination. Compare with: ROSE, a network protocol which is similar in function but uses manual management of routing tables rather than automatic dynamic routing.

There is some suspicion that the original Software 2000 version of NET/ROM has it's roots in software written by DF2AU of NORD><LINK. No court case was held and in popular circles this discussion has been dropped. Certain organizations and magazine companies have been ignoring developments in NET/ROM networking because of this suspicion.

(See *TheNET*, *BPQ*, *ROSE*)

## network

In packet terms, a network is a system of nodes interconnected in such a way that any node can communicate with any other node in the system in an efficient and speedy manner. An example would be a network of user ports wire-linked to backbone nodes that are in turn connected to each other by UHF point-to-point links.

## network layer

Level 3 of the seven layer OSI communications protocol set. The network layer specifies the communications between adjacent nodes or networks and interfaces with the User at the level 2 link layer and with distant nodes at the level 4 transport layer protocol.

## node

A node is an active element in a network. This can mean anything from a user station to a bulletin board. Traditionally a node in packet radio is an intelligent router of real time data, somewhat more intelligent than a digipeater but faster than a store and forward BBS. (See *TheNET*, *NET/ROM*, *gateway*, *network*, *digi*, *repeater*, *server node*)

## node broadcasts

Each node sends a one way message out every half hour (setable) that tells it's neighbors what nodes are listed in the nodes table. The neighbor TheNET nodes interpret this information based on a factor called quality. (See *Nodes Broadcasts* which is part of *Theory of Operation* in the *TheNET Network Node Guide* part of this document)

## node cluster

See node stack.

## node hopping

node hopping, or route stepping, is when the user explores the network by the process of connecting from one node to the next along a path and checking the routes available by reading the nodes table, routes list and INFO response. One of the features of TheNET is that an individual may hunt through a TheNET networking and take advantage of local Routes commands to determine what all of the neighbors of a particular node are. With this knowledge the user may then connect to a neighbor node and repeat the process. In this way an individual user may entirely map the existing network or collection of networks.

One advantage of route stepping is that if there is a site that is accessible from one end of a network but that is not known on the other side of the TheNET network the user may simply connect from his/her origin user port to a node that knows of the desired site and then connect to that desired site. This may be done repetitively to get to a very distant node.

Another advantage of route stepping is that there are timers internal to the TheNET node that specify how long a packet may take to get from one end of the network to the other. If the packet takes longer than specified by the timers then network-end-to-end retries are performed. This results in unnecessary network load. Furthermore if the retries ever fail then the user is disconnected. By making smaller steps the user may create a more robust path. Smaller hops, when taken to extremes, will result in excess network loading though. Node stepping has been taken as a disadvantage by some network developers due to the (apparently) vast amount of network traffic that is generated by this user-play process. It is only bad for a station to do node hopping if the process is abused for normal traffic. It is best for normal, non-exploratory traffic for a station to connect across three backbone links at a time in a direct path to the destination. (See *TheNET*)

## node-op

The person(s) responsible for the software and parameters of a node stack. Also called site sysop.

## node stack

Two or more nodes on one site interconnected by a diode matrix. Also called a node cluster.

## NOS

Network Operating System. NOS is a program which is generally used to communicate using the TCP/IP protocols but which actually is much more than just a program that does TCP/IP. NOS runs on a personal computer and. NOS is the name used to describe dozens of different programs which are all similar and which all are used for TCP/IP but which have huge differences in look and feel and features supported. NOS exists for most hard-disk based personal computers. (See *KA9G*, *TCP/IP*)

## NRZ

Non Return to Zero. A binary code format in which binary ones and zeros are represented by two discrete voltage levels and the voltage remains at the indicated level for the duration of the code bit. Compare with RZ or bipolar pulse modulation in which the signal would return to an average level between bits. The name NRZ is somewhat confusing but it may come from the magnetic recording industry where zero indicated an unmagnetized state on the tape and binary ones and zeros were indicated by positive or negative magnetization. NRZ is the form that most binary signals take within computer circuitry. (See *NRZI*)

## NRZI

Non Return to Zero Inverted. A binary code format in which a data 0 produces a transition (either from 0 to 1 or from 1 to 0) in the code and a 1 in the data produces no change in the code. It does not mean that the NRZ code is merely inverted. The main advantage of NRZI is that it does not matter at what point in the transmission that one starts to decode, the subsequent data will be the same. The code signal actually sent to the modem and transmitted by an amateur packet TNC, is in NRZI format. On receive the TNC converts it back to NRZ format for the CPU to process. NRZI is also known as NRZ-S (space).

## NTECH

NEDA technical committee. A committee operating under NEDA that formulates technical standards and resolves problems in the NEDA network or on networks in general.

## null modem

A interconnect device or data cable used to connect together two DCE or two DTE digital devices. The RS-232 TXData/RXData and RTS/CTS lines are swapped.

## obsolescence count

In a NETROM/TheNET system, each node entry in the nodes table is given an "initial obsolescence count" each time the route is confirmed by a neighbor's node broadcast. The obsolescence count is reduced at regular intervals. When the obsolescence count reaches a predetermined value, the node entry is considered obsolete and is no longer broadcast to its neighbors.

## Octopus

This is a product of John Painter-N0NDO (rights now owned by NEDA) that is an 8 port diode matrix card which couples TNC2 compatible TNCs to make a multiport node stack. This product has been outdated by the HexiPus™. (See *HexiPus™*, *Diode matrix*)

## OEM

Office of Emergency Management. This is a term used by some state governments for a state or county government owned facility at which emergency services radio equipment and other gear is clustered. Some O.E.M. facilities are located in bomb shelters, others in buildings as part of a state office complex. A good packet network would include an on line TNC or BBS at every O.E.M. in the network's region. (See *EOC*)

## OPEN

Ontario Packet Experimenters Network. For more information see TPG or HEX9. Also Ohio packet experimenters network.

## OSI

Open Systems Interconnect. The OSI is a project of the International Standards Organization to develop a set of computer communication protocols. The OSI is a framework with which communications protocols are described. Once a protocol can be described within the OSI framework it is easy to see how the protocol can be used in concert with other protocols. An example of how the OSI framework is used can be seen in the *Theory of Operations* section of the *TheNET Network Node Guide* in this document.

### overhead

The non-information data that is sent on a channel to control routing, addresses and supervisory bytes sent with the information data in the packet and any other transmissions that do not convey actual information but still take up time and capacity on the channel. Some node protocols (like ROSE) have less overhead than others (like TheNET).

### PacComm

PacComm Packet Radio Systems Inc. PacComm designs, manufactures and markets a range of packet radio products, both amateur and commercial, including the popular Tiny-2 TNC used in many network nodes.

### packet

A packet is a block of many characters (or bytes) which are sent together along with a few extra characters (checksum) used to guarantee that the data is completely error free. The packet includes addressing information so that the receiving station knows that the packet is for it as well as who sent the packet. (See *datagram*)

### packet switch

Another name for a L3/L4 node. (See *node, ROSE, NET/ROM, TheNET, BPQ*)

### PacketCluster

A proprietary software from Pavilion Software. It creates a specialized BBS for DXers and operates with all users connected in such a way that dx information can be distributed in "real time".

### PacketTen

An advanced high speed network node controller designed and marketed by Gracilus Inc. The PacketTen controller will handle up to 10 ports, some of which can be up to 1 Mbps (that one million bits per second) and runs TCP/IP NOS code.

### PACSAT

An amateur radio satellite carrying a packet store-and-forward node. When launched became OSCAR 16. OSCAR 19 (LUSAT built by AMSAT Argentina) is almost identical.

### PACTOR

PACket Teleprinting On Radio. An HF digital communications protocol developed in Germany. PACTOR combines the good features of both AMTOR and packet for improved, more efficient HF data communications.

### PAD

Packet Assembler/Disassembler. A device that interfaces an ordinary "dumb" terminal to an X.25 packet network. It gathers typed characters into outgoing packets and translates incoming packets back into serial asynchronous data streams. Also provides a simple command interpreter for setting up and tearing down connections, controlling parameters, etc. The amateur packet radio TNC was heavily modeled on the PAD.

### Paket

A terminal program written by Tony VK2DHU. (See *Lan-link, Yapp*)

### PakMail

AEA's name for their personal BBS in a TNC.

### Pakratt

AEA's PK-232 multi-mode TNC.

### parameters

In TheNET software there is a list of values used by a TheNET node to configure options. These values are called parameters or PARMS. (See Parameters, in the *TheNET Network Node Guide*)

### parity

A bit added to a binary word for error checking purposes. For odd parity, a 1 or 0 bit is added to 7 data bits so that the total bit count is an odd number. For even parity, the total bit count is made even with the parity bit. Parity words can similarly used with groups of binary words.

### PARMS

Abbreviation for parameters. In TheNET software there is a list of values used by a TheNET node to configure options. These values are called parameters or PARMS. (See Parameters, in the *TheNET Network Node Guide*)

### path

This word is used to mean the nodes, digis and servers that must be used to pass data from one point to another. Sometimes the path may be specified without including some intermediate nodes if the knowledge of those nodes is not necessary to pass the data or make a connection.

### PBBS

Either Personal Bulletin Board System or Packet Bulletin Board System. The former is called personal mail drop or personal mail system (PMS) to avoid confusion. PMS indicate a mail box that is contained inside a normal user TNC. Packet Bulletin Board Systems are referred to as BBS usually. (See *BBS*.)

## PC

Personal Computer. Usually refers to a computer that is identical in function to a product by IBM that was marketed as an IBM PC. They are more correctly referred to as IBM compatible PC. PC could mean any kind of computer that is used by an individual for general purposes (i.e. not a microwave oven control panel). It is sometimes hard to determine if a person who mentions PC is referring to a generic personal computer (i.e. Macintosh, Amiga, IBM PC, Atari etc..) or specifically means an IBM PC compatible. In the Annual we'll try to be real careful and refer to all IBM compatible PCs as PC compatibles or IBM clones.

The PC compatible is the hardest of the popular personal computers to learn, the least expensive for the processor and display performance you get, and the most supported with technical software. There are dozens of choices for packet radio software available for the PC compatible. There are even several different custom hardware products that let you have a TNC built into the PC compatible. The cheapest method of creating your own TheNET proms is also based on a PC compatible and a plug in hardware card. (See *Macintosh*, *Amiga*)

## PC*PA

Personal Computer Packet Adapter. A PC plug-in card made and marketed by DRSI that operates as a TNC.

## perming

In a TheNET system, the technique of network management by "locking routes" and "locking nodes" is called perming.

## persistence

A collision avoidance technique where the decision to transmit a packet is made by generating a random number between 0 and 255, comparing it to a parameter called p-persistence and if the random number is less than the p-persistence parameter, the packet is sent. If the number is greater, the TNC waits a period of time called slot time and then repeats the process. The p-persistence parameter is usually set to a value equal to 256 divided by one less than the maximum number of stations expected on the channel. (See *Frack*, *Dwait*, slot *time*, *collision avoidance*, *back-off*)

## personal BBS

A limited function BBS contained within a users TNC firmware with which the user can enter or receive his/her own personal messages from other users or from the nearest full service BBS. Usually referred to by one of the commercial trade names such as PMS, Mail Drop, etc.

## PID

Protocol ID. The first byte of the packet frame which identifies which protocol is used for the packet frame. AX.25 PID is $F0 while TheNET and other higher level protocols have other PIDs.

## PLL

Phase-Locked Loop. A circuit using feedback methods to control the frequency of an oscillator. Usually used as frequency control in synthesized radios.

## PM

Phase Modulation. A modulation technique in which the phase of the carrier is changed in relation to the modulating signal. Similar to FM, but not the same. (See *FM*)

## PMP

Poor Man's Packet. A simple modem and software package to be used with a personal computer to emulate a TNC. Developed by and published in 73 Magazine. (See *Baycom*)

## PMS

Personal Mail System. PacComm's version of a limited function BBS contained within a user's TNC. It usually is included with the TNC as a standard feature when it is bought. The program allows the user of the station, or hams connecting over the radio to leave mail that can be picked up either locally or remotely. Some incorporate the ability to reverse forward. (See *forward*, *reverse forward*)

## point to point link

This is a radio path between two sites on a frequency where there will only be two radios within range. Neither radio may hear any other radios except the other end of the link. If these conditions are met packet data may travel across the link at the absolute highest throughput possible for a packet link using the provided equipment. A point to point link is much faster (by several times) than any other kind of link architecture using the same class equipment. This is because there will be no delays caused by collisions. A point to point link is much more efficient, in terms of spectrum usage, than any other link architecture. This is because directional antennas may be used and power may be minimized at each end of the link.

Note that the fact that a link is a point to point link does not specify what kind of data is being passed or what baud rate the link is running. Point to point links may be used for server to server, server to network, network to network or user to network communications.

In a TheNET network point to point links should be used for all node to node backbones. For mapping purposes the specification Point to Point Link is only used if both stations indicate that they will be the only two stations on that frequency. This specification implies the use of selective neighbor routing.

(See also the section titled *Networking Around HTS* in the *TheNET Network Node Guide* in this document)

## polling

In packet terms, polling is a collision avoidance method in which one master station queries each of the users on the channel if they have a packet to transmit. The slave stations will not transmit until they have been "polled" by the master station. In this way no two stations will transmit at the same time thus avoiding collisions. No amateur systems exist which use this method, at this time.

## poll packet

In the latest version of AX.25 packet protocol if a transmitted information packet is not acknowledged the transmitting TNC will generate a poll packet to see if the destination TNC is still around. If the poll packet is acknowledged then the transmitting TNC will once again attempt to send the information packet. Note that if there is a periodic noise at the receive TNC that the poll packets might be received but that a particularly long information packet might never get through. In that case the retry process might take place until manual intervention occurs. (See *Retry*)

## port

An input/output channel or connector on a node or TNC. A TNC normally has one or more radio ports hooked to a radio transceiver(s) and a RS-232 or serial port that may be connected to the users terminal (computer) or another node serial port in the case of a TheNET (or other type of higher level node) node stack. A port may also refer to a special purpose node such as a user-port, IP-port, backbone port, etc. or any such access node to a network.

## PROM

Programmable Read Only Memory. (See EPROM)

## protected backbone

A protected backbone is a backbone where none of the known devices involved in the backbone will accept traffic from any unknown device. (See *Backbone*)

## protocol

A communications protocol is the set of rules and procedures used to implement a technique or method of communications. There are many different protocols for many different purposes. AX.25 is a protocol which describes how small computers will talk to each other. TCP/IP is a protocol which describes how computers of all sizes will talk to each other, using more computers as mid stations. TheNET protocol describes how nodes in a network will talk to one another.

## pseudo-digital repeater

A full duplex node or repeater which transmits a sig nal (such as a tone, flags or unmodulated rf carrier) t indicate to the other stations on the channel that th channel is in use. An example would be a HAPN 480 bps node that transmitted a carrier whenever an rf signa was heard on the input channel. As the HAPN DCD i actually a noise squelch system, the presence of such carrier on the input of the node is "repeated" to the outpu channel. This can be very useful on split frequenc multiuser shared channels. It is also simpler than a ful regenerating repeater but at the cost of half th throughput. An audio repeater used in packet servic would be a pseudo-digital repeater if the repeated tone were unsuitable for decoded data (say due to phase shif or noise) but served to prevent collisions.

## PSK

Phase Shift Keying. A data modulation method in whicl binary data is encoded as discrete changes in the phas of the carrier signal. In amateur packet, PSK is use mainly on OSCAR satellite data communications PacComm markets a PSK modem to be used with TNCs

## PSR

Packet Status Register, a newsletter published by Tucso Amateur Packet Radio Corp. (See *TAPR*)

## QRM

Man-made interference on a radio frequency, intentiona or not.

## QRN

Natural interference on a radio frequency. Lightning solar noise (very week signal work) are examples o natural interference.

## quality

TheNET software allows for a factor called quality. Th quality factor is used to determine the path length fo a network connect.

Quality factors are determined as part of the node broad cast sequence. Each node sends a one way message ou every half hour (setable) that tells it's neighbors wha nodes are listed in the nodes table. The neighbors hav a quality factor that associates incoming node listing to new quality values. TheNET quality values for a give link are not calculated by the nodes themselves but ar set by the sysop. Thus the fact that a link has a highe quality than another link does not reflect the perfor mance. Quality values are used to limit the spread c nodes over selected backbones and wire connections. (Se Nodes Broadcasts which is part of *Theory of Operatio* in the *TheNET Network Node Guide* in this document

## R: header line

See header.

## R95

Radix 95 is a shareware program written by Greg WD5IVD and distributed by Texas Packet Software. A communications protocol for sending binary data on packet by converting it to ASCII by a known algorithm, splitting the file into small chunks and sending them as regular messages. (See *7-Plus*)

## RAM

Random Access Memory. This is an IC in a computer that holds data only so long as power is applied. This is usually used only for storage during the execution of a program. TNCs use RAM for temporary storage, messages and parameters. Normally TNC RAMs are powered all the time using a lithium battery in the TNC.

## RATS

The Radio Amateur Telecommunications Society. RATS is an amateur radio association in New Jersey that is dedicated to the improvement of communications systems in the Amateur Radio Service. RATS is best known in the packet field for ROSE, a networking protocol software written by Tom W2VY and others. For more information on RATS contact: The Radio Amateur Telecommunications Society, PO Box 93, Park Ridge, NJ 07656-0093. (See *ROSE*, *ROServer*, *PRMBS*)

## real time

When a signal is sent and a result is expected back within a short enough time to fall within a person's attention span the operation is said to be in Real Time. Keyboard to keyboard operation is real time. Keyboard to server is real time. Reading your mail from a remote BBS is real time. Sending a message to a friend via a packet BBS is not real time because the sender doesn't know how long it will be before the friend's answer comes back.

## REBBS

An amateur BBS software written by Roy AA4RE. Also 4RE BBS and AA4RE BBS. REBBS is used as a @BBS destination for bulletins directed to users and/or sysops of AA4RE BBSs.

## redundancy path

In a mature packet network more than one path would exist between every two points. If one of the two paths is preferred due to load handling capability or number of node hops then that path would be called the primary path. The other path would be called the redundancy path. (See *Path*)

## regenerating repeater

See repeater, full duplex real-time regenerating.

## repeater, full duplex real time regenerating

A digital repeater that receives and transmits at the same time (on 2 separate frequencies). The received and demodulated data is fed to the transmitter modulator so that an exact copy of the received data (but cleaned of all noise and distortion from the receiver) is retransmitted at the same time it was received. Such repeaters are useful to create a hidden transmitter free environment on a wide area network. (See *digipeater*, *real-time*, *pseudo-repeater*)

## response time

This is the time between sending data to a remote device before an expected response returns to the originating station.

## retry

Retry is the process by which a packet that is sent and not acknowledged will be resent by the sending station. This retry is repeated until the acknowledgment is received or until a "retry counter" reaches its limit and the circuit is terminated. (See *back-off*, *dwait*, *frack*, *persistence*, *collision avoidance*)

## reverse forward

This is a feature of all modern BBSs and some PMSs where a connecting BBS may request if any mail is available to be taken by the connecting BBS. The prompt and answer exchange that takes place is in plain text and may be monitored if you are curious. (See *forward*)

## RLI BBS

An amateur BBS software package written by Hank W0RLI.

## RM

Reference Model. Another name for the OSI 7 level set of data communication protocols. (See *OSI*, *ISO*)

## ROM

Read Only Memory. A non-volatile memory IC used to permanently store operating programs in computers and other digital devices. ROMs come in many forms such as PROM (field Programmable ROM), EPROM (Erasable Programmable ROM), EEPROM (Electrically Erasable Programmable ROM), OTP (One Time Programmable ROM), etc.

## ROM image

The set of binary data that is programmed into an EPROM.

## ROSE

RATS Open System Environment: This is a networking software package created by Tom Moulton, W2VY, in concert with the Radio Amateur Telecommunications Society in New Jersey which implements a multiport, multistation packet radio network. The software consists of several parts. The most major part is in the form of an EPROM that resides in a TNC2 compatible TNC. Other parts include network management and system operation tools that run on a PC compatible but which are not integral to the network's day to day operation.

ROSE operation is done with the use of site callsigns and numeric designations that are traditionally in the form of a telephone area code and local code. The user treats the local ROSE 'switch' as a digipeater with the destination switch's numeric code as a second digipeater in the user connect sequence.

Example:

- To connect from WB2DWD's station in Long Valley NJ to KA2DEW's station in Potsdam NY a user would do:

  C KA2DEW v NX2P-3,315268

  where KA2DEW is the destination user's callsign, NX2P-3 is the user's local switch, and 315268 is the designation for the switch in KA2DEW's area and on the frequency that KA2DEW will be operating. Note that KA2DEW will get a connect message that looks like:

  *** Connected to WB2DWD via K2CC-3,201876

  call being setup

  where WB2DWD is the originating station, K2CC-3 is KA2DEW's local switch and 201876 is the numeric designation for WB2DWD's local switch. 201876 is the same switch as NX2P-3 and 315268 is the same switch as K2CC-3.

  ROSE will acknowledge the connect attempt immediately, even though the packet hasn't had time to traverse to your destination. Once the connect is complete a message will be sent to the originating station (you) that says:

  call complete to KA2DEW @ 3100201876

  3100 is the DNIC (Data Network Identification Code) for the United States.

The linking methodology between ROSE switches is very open to the design requirements of the implementation. ROSE switches may be linked with a common trunk frequency, with hidden transmitter free backbones, or on the user access frequencies. ROSE switches may be built with from one to many TNCs on many frequencies.

The routing methodology used in ROSE is based on a fixed table stored in RAM in each switch and downloaded by the designated sysop. This may be done locally or from the far end of the network. The routing may list individual switches and include the neighbor for each individual switch or the switches may be listed by class allowing whole 'area codes' to be listed with the same neighbor node. Alternate routes may also be specified. If a link fails completely or is taken off the air the system would adapt quickly.

This software has been in Beta test station for several years and as of August of 90 has been used successfully for building multiport nodes.

A notable difference between ROSE and TheNET is that with ROSE a user doesn't have to know about any intervening hardware between his entry and exit ports in the network. On the other hand the user may be unable to find out anything about the intervening hardware. There is an option with ROSE for the sysop to download applications to a ROSE switch (must be renewed each time power is removed from the switch) which may provide information as to the network connectivity and services.

For more information on the ROSE protocol, see the downloadable user information files on many BBSs or contact RATS directly at POBox 93, Park Ridge NJ 07656. Or send to ASKRAT @ KB4CYC.nj

## ROSErver/PRMBS

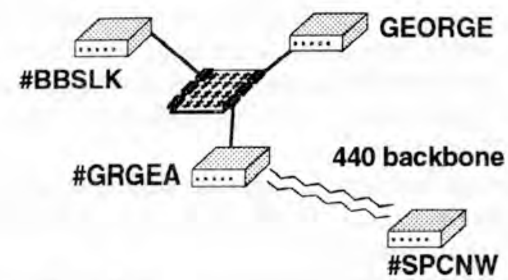
An amateur packet BBS software written by Brian KA2BQE. Although PRMBS was written originally for use on the ROSE network, it is now being used on many non-ROSE systems. (See *BBS*, *ROSE*, *RATS*)

## Route Stepping

See Node Hopping.

### router

Any device that can control and direct the routing of datagrams by a node. For example the internal protocol in TheNET X-1H firmware that directs IP datagrams to their proper address according to a manually entered routing table would be an "IP router". (See *dynamic routing*, *TheNET*)




## routing loop

This is a condition where a packet is sent through a node more than once due to routing errors. Routing errors can occur when a node is goes off the air or if a backbone link is lost. Here is an example of a routing loop. If #SPCNW tells the #GRGEA about SPCNDL node and then the 440 backbone goes down #GRGEA will still tell GEORGE and #BBSLK about SPCNDL. In it's next broadcast GEORGE will tell #BBSLK about SPCNDL. #GRGWE will then tell #GRGEA about SPCNDL. After three node broadcasts #GRGEA will forget about the 440 backbone to #SPCNW but will still know about SPCNDL because #BBSLK will have sourced it.

Now, if a packet comes into #GRGEA destined for SPCNDL it will pass it to #BBSLK who will then pass it to GEORGE who will then pass it to #GRGEA and the loop will continue until time to live runs out.

## RS-232

RS-232-C is the Electronics Industry Association (EIA) standard number for the most common interface used between computers. This is usually called RS-232. A signal which uses the RS-232 standard is often said to be at RS-232. The computer to TNC connection is at RS-232. Normal computer internal data signals use ground and +5 volts to indicate a zero or a one. RS-232 generates -12 volts and +12 volts to indicate a Mark or a Space which are analogs to zero and one. A RS-232 receiver detects a Space (a bit = 1) as anything greater than +3 and a Mark (a bit = 0) as anything less than +3. The reason that this method is used for computer to computer signaling is that TTL (0v -> +5v) is more subject to line noise, capacitance and non-true grounds than is RS-232. Also called serial communications.

## RTS

Ready To Send. A control on a RS-232 port that indicated that the device has data ready to send. On some devices the DTR line is used instead of RTS. On a packet TNC modem header, the RTS line is used as PTT signal for the transmitter. (See *CTS, DTR, RS-232, handshaking*)

## RTTY

Radio TeleTYpewriting. An early mechanical based (now supplanted by computers) method of data communication on radio using the baudot code. In 1980 ASCII was also permitted on RTTY in the US. TELEX and TWX are commercial telephone systems using the same techniques as RTTY.

## RUDAK

Regenerativer Umsetzer fur Digitale Amateurfunk Kommunikation (English = Regenerating Transponder for Digital Amateur Communications). A packet transponder project flown on board OSCAR 13 satellite. Developed by AMSAT-DL group in Germany.

## RXData

Received Data stream produced by a modem demodulator. (See *TXData*)

## SAREX

Shuttle Amateur Radio EXperiment. An educational program in which U.S. Shuttle astronauts communicate with schools using voice and packet.

## selective neighbor routing

This is the same as locked neighbors. This means that each packet station has a list of neighbor stations which may be talked to. Other neighbor stations would be ignored. This does not presuppose that unlisted other stations might network in from beyond the neighbor stations. The reason that this would be done is to discourage the trashing of point to point links. (See *point to point link, HTS*)

## semi-duplex

A communications system that is full duplex at one end and simplex at the other end. Packet operation through a real-time full-duplex repeater would be semi-duplex operation.

## serial port & serial communications

The part of a computer responsible for sending binary data in a serial fashion. Normally computers talk internally with parallel data signals, that is that all of the important bits for a block of information are sent at once. A serial communications uses only one wire which is toggled many times for a single block of information. Thus a letter A might be sent in parallel all at once when it must be sent as a string of ones and zeros in sequence in serial. The serial port usually consists of a single chip called a UART, a RS-232 driver chip and a connector.

## server

A server is any stations that is operating as a service to other users than the owner. This may included BBSs, DxClusters, DOSgates, TheNET nodes, ROSE nodes, TCP/IP hosts etc.

## shared, HTS free backbone circuit

A backbone channel shared by 3 or more nodes, all of whom can hear each other either directly on a simplex frequency or with the help of a digital repeater (regenerating or pseudo- repeater) on a full duplex split frequency. The available throughput on the channel is shared between the users and appropriate backoff delay must be used. (See *backbone, HTS, repeater, collision avoidance*)

## SID

System ID. A block of characters exchanged between BBSs when they connect to each other. The SID is used to indicate which software system the BBS is running and what features are available for forwarding.

## simplex

A communication method in which communication between two stations takes place one direction at a time regardless of whether the receiving and transmitting is on the same frequency or on split frequency. However in amateur radio terminology, simplex usually means receiving and transmitting on the same frequency. This can lead to confusion, so one should be very precise as to what they mean when referring to simplex (either single frequency simplex or split frequency simplex). Amateur packet operation is normally simplex but most modern TNCs are capable of operating full-duplex if connected to the proper radio system. (See *half-duplex, full duplex*)

## simplex digipeater

a regenerative digital repeater that receives a packet, verifies that it was received correctly, and if appropriate retransmits it on the same frequency it was received on as soon as the channel is clear. Also called a Store-and-forward repeater. Compare with: Repeater, Full duplex real time regenerative, which retransmits the data at exactly the same time as it was received but which does not check for errors. (See *digi, duplex digipeater, repeater, store-and-forward*)

## site hardening

Term for ruggedizing a site by adding backup power, shielding or lightning protection. This includes weather protection and protection against RF problems or nuclear attack. In amateur packet networking, usually only protection against heavy ice, high winds and power outages are considered.

## site manager

This is the person or persons that are responsible for node hardware and site access.

## site sponsor

This is the person or persons who are financially involved in acquiring and maintaining node hardware.

## site sysop

This is the person or persons who have software control over a node site.

## SITOR

A commercial communications system very similar to AMTOR and used mainly for marine communications.

## slime trail

On a NETROM/TheNET node, the node table will temporarily show distant nodes that connect through it. The temporary node will be listed at the beginning of the nodes list and will show callsigns only, no alias. This node list entry is called a slime trail because you can trace back to see the origin and route of the distant node.

## slottime

In the persistence method of collision avoidance, slottime is the time delay before repeating the random number persistence calculation. (See *persistence, Dwait, CSMA / CA, collision avoidance*)

## SMTP

Simple Mail Transfer Protocol. This is the part of the TCP/IP system which is responsible for sending mail between TCP/IP hosts. This is a non real time service which operates in a way very similar to normal packet BBSs.

## space

Space, like mark, is one of the two possible states in a binary communications system. On an asynchronous serial system, the start bit is space. On a RS-232 port, the positive voltage level is space.

## SSID

Secondary Station IDentification. A callsign is normally used as an address. In a case where a ham needs to have more than one address on the air at the same time the callsign may be used with an ssid. There are 16 different possible SSIDs. NK1M-2 is an example of an address using a callsign and an ssid of 2. NK1M is the same as NK1M-0. -15 is the highest ssid that can be used.

A function of TheNET, G8BPQ, MSYS and other node software is to change the ssid of a callsign that passes through the node or network of nodes. If a station connects to a node with an ssid of 0, when the station connects out of the node with an ssid of 15. The formula used is 15 - ssid = exit ssid. Thus a station using an ssid of 4 will emerge from the node or network of nodes with an ssid of -11.

## STA

Special Temporary Authorization. In the USA, a special permit granted by the FCC to operate in a manner not normally permitted. ALL AUTOMATIC HF packet forwarding up to 1993 was permitted under a small number of STAs sponsored by ARRL even though unattended HF operation was not normally permitted.

## store-and-forward

The process used in nodes and digipeaters where a packet is received, processed for errors, etc., and retransmitted toward its destination at a later time (seconds later). This is compared with a full duplex digital repeater that retransmits the data at virtually the same time. The term may also describe the BBS forwarding of messages hours or even days after they are entered or received.

## stream

AX.25 allows many connections to be made from one station at the same time. Each connection is called a stream. The origin and address callsign pair for the connections must be different for each stream. That is: If I am KA2EIA and I connect to three other stations, I can connect to NK1M, K1MEA and NQ1C but I cannot connect to NQ1C, NQ1C and K1MEA. This process is called multistreaming and is available on most modern TNCs. Look at the **USERS** command in your TNC's manual.

## switch

Another name for a node. (See packet switch, node, ROSE, BPQ, NET/ROM, TheNET)

## synchronous

A serial communications mode in which the data bits are sent continuously without character start and stop bits and with an embedded clocking signal for synchronization of the receive circuits. Amateur AX.25 packet radio communications use synchronous transmission of data.

## sysop, BBS sysop

The person(s) responsible for the smooth operation of a BBS, including maintaining forwarding routes, redirecting misaddressed messages, checking for illegal or improper messages, etc. (See site sysop, BBS)

## TAPR

Tucson Amateur Packet Radio Corp. TAPR is a non-profit organization that develops and promotes amateur packet radio including hardware, standards and publications. TAPR are probably best known for the TNC-2 TNCs (and their widely marketed clones) that were the impetus for the major growth of amateur packet radio. They publish a regular newsletter called Packet Status Register (PSR). For more information on TAPR please contact: Tucson Amateur Packet Radio, PO Box 12925, Tucson AZ 85732-2925.

## TCP

Transmission Control Protocol. A major element of the ARPA suite. Provides reliable, connection-oriented byte stream service on an end-to-end basis. Runs atop IP and sits at the transport and session layers. (See KA9G, TCP/IP)

## TCP/IP

Transmission Control Protocol/Internet Protocol. This is a suite of protocols that define the operation over the 'Internet'. This package of protocols was used by Phil Karn, KA9Q, for the creation of a packet radio version of TCP/IP. As this is a fairly mature networking system it supports many features not available in the current 'made for ham radio' protocols. It also has features that would take much better advantage of networking resources for the transmission of volume data than do TheNET and ROSE. One problem that TCP/IP for ham radio currently has, however, is that it requires a more sophisticated computer and a more sophisticated operator than are required to operate ROSE and TheNET. (See Internet)

## TELNET

A presentation/applications protocol in the ARPA suite (included as part of most TCP/IP packages) used for terminal to terminal and terminal to host communications (e.g., remote login).

## terminal

A terminal is a display entry device. A CRT Terminal, which means Cathode Ray Tube terminal is normally just called a terminal. They are also referred to as dumb terminals. Sometimes a computer is used as a terminal.

A terminal is usually a display screen and keyboard hooked to an RS-232 port. When you type on the keyboard data is sent out of the TxData pin of the RS-232 connector on the terminal. When RxData signals are detected on the RS-232 connector the text is displayed on the screen.

## TEXNET

A networking node protocol developed by the Texas Packet Radio Society Inc. and used primarily in Texas and the southwest. TEXNET uses a custom three port node which supports 1200 or 9600 baud modems, as daughter cards. An interesting feature of the TEXNET board is that it can support a local hard drive using the TEXNET board's on-board disk controller.

## TheNET

This is a networking software package created by Hans Giese, DF2AU, in concert with NORD><LINK in Germany which implements a multiport, multistation packet radio network. The software consists of several parts. The most major part is in the form of an EPROM that resides in a TNC2 compatible TNC. Other parts include node configuration tools that run on a PC compatible but which are not required after initial system setup.

TheNET operation is done with the use of site callsigns and mnemonic designations that are traditionally in the form of a city name. The user treats the local TheNET node as a remote command processor by first connecting to it, then away from it.

Example:

- To connect from WB2DWD's station in Long Valley NJ to N2MGI's station in Potsdam NY a user would do:

  C SUSSEX, then when connected,

  C POTSDM, then when connected,

  C N2MGI where N2MGI is the destination user's callsign, SUSSEX is the user's local switch, and POTSDM is the designation for the switch in KA2DEW's area and on the frequency that KA2DEW will be operating. This would only work if POTSDM showed at SUSSEX's nodes list. In practice with TheNET each connect step can only be a few node steps.

  Note that N2MGI will get a connect message that looks like:

  *** Connected to WB2DWD-15

  where WB2DWD-0 is the originating station

The linking methodology between TheNET nodes is very open to the design requirements of the implementation. TheNET switches may be linked with a common trunk frequency, with hidden transmitter free backbones, or on the user access frequencies. TheNET switches may be built with from one to many TNCs on many frequencies.

The routing methodology used in TheNET is based on a dynamic table stored in RAM in each switch and automatically generated by periodic information transfers between nodes and within restrictions placed on each TNC by the designated sysop. This may be done locally or from the far end of the network. The routing lists individual nodes and includes the neighbor for each individual node. Alternate routes are automatically generated but in practice are not used. If a link fails completely or is taken off the air the system will adapt to the lack of that link after a number of hours.

TheNET has been in use now for several years in the US. NJ7P, Bill Beech in Arizona has been releasing heavily modified versions of TheNET 1.0 under the name of TheNET Plus. G8KBB has been releasing a different heavily modified version of TheNET 1.0 under the name of TheNET X.

There is some suspicion that the original German version of TheNET has it's roots in software stolen from Software 2000. No court case was held and in popular circles this discussion has been dropped. Certain organizations and magazine companies have been ignoring developments in TheNET networking because of this suspicion.

A notable difference between TheNET and ROSE is that with TheNET a user can delve into the routing tables of each of the nodes and find out how the network is put together. The user can also determine from available information at each node TNC how well the node is managed and how well it is integrated into the surrounding network equipment. On the other hand the user is required to know at least some information about the network's architecture, at the minimum, and in some areas the user needs to have a very complete knowledge of the architecture in order to use the TheNET nodes effectively.

(See ssid, G8BPQ, NET/ROM, MSYS)

## TheNET PARMS

TheNET nodes, which run in TNCs, operate using timers and other parameters that are initialized in the EPROM when it is burned. Some of these timers and parameters may be modified over the air by the site sysop. A complete description of these parms is published in this document. (See TheNET)

## TheNET Plus

TheNET 2.xx versions written by Bill NJ7P.

## TheNET X

A version of TheNET written by G8KBB.

## three way wireline link

This is a circuit that allows up to 3 TNCs to be connected together as if they were over the air to each other. This circuit bypasses the modems in each TNC so that the three TNCs may communicate at high speed. The three way wireline link circuit was presented in the NEDA Quarterly 1.4 and has been out-dated by the Multi-way Wireline Node Switch Coupler (Quarterly July 93). (See wireline link)

## throughput

This term specifies how many bytes sent by an origin station actually reach the destination station in a give period of time. Throughput is a much better number to describe network performance than baud rate. Baud rate only describes the number of bit transitions that may possibly leave a transmitter in a second. Throughput is a statistic that actually means something to an end user. Throughput is calculated either by observation or by taking the original baud rate, given in bytes per second, subtracting out all of the wasted time and overhead due to network protocols, the lost time due to choking and due to collisions. (See Choke)

### time to live

When a packet is sent from one TheNET node to another TheNET node the packet contains several bytes of information which are useful for the intervening TheNET nodes. One of those bytes of information is the time-to-live. Each time a node relays the packet one hop further the time-to-live is decremented. When it decrements to zero the message is discarded. Thus if the number of hops that the packet has to go to reach it's specified destination is greater than the initial time-to-live the packet won't get there. In addition if the time-to-live on the return trip is not high enough an acknowledgment will not be returned.

### throughput

The amount of data sent by an originating station that actually reaches its destination in a given period of time. This must not be confused with the channel data rate in baud or bps. Throughput can be calculated either by observation or by taking the original data rate and subtracting out all the wasted time and overhead due to network protocols and TX delays, lost time due to choking and that due to collisions. See also choke, overhead, retries, response time.

### Tigertronics

Tigertronics Inc. markets a Baycom modem called Baypac and is an official distributor of Baycom ver 1.5 software. (See Baycom)

### time-to-live

In a NETROM/TheNET network, all frames are assigned a Time-to-Live number which specifies the maximum number of nodes the frame can be passed to before being canceled. It is used as a protection against looping endlessly or to control the propagation of L4 routes.

### TINK

Slang for TNC.

### TNC

Terminal Node Controller. This is the brains that connects the user's terminal to his radio so that he can communicate to other stations. The TNC's job is to take text typed on the terminal or computer and store it until the user hits a return. At that time the text is sent to the destination station. Each line of text (ended with a carriage return) becomes a packet and is stored in the TNC until it can be sent to the destination station.

The TNC also has commands that let the user set the callsign of the station and set up communications with another station or stations (Connect command).

The TNC is like a home phone modem in that it converts digital character data to tones. The big difference between a TNC and a phone modem is that the TNC has the intelligence to direct your traffic to a specific destination and to receive connects using it's own microprocessor and internal software. A phone modem is relatively stupid in that it can only work on a channel where there is only one destination, i.e. a telephone.

By changing the internal software TNCs may also be used for other purposes besides home station use. This includes running as part of a set of TNCs in a network node.

### topology (network)

The interconnecting hardware portion of the network and how it is planned and implemented. How well a network functions is more related to its topology (the quality of its links and paths) than to the software used to form a network. Topology may be described without reference to geography. Geography need not be explained to describe what services exist and what form interconnections take in a network.

### TPG

Toronto Packet Group. An active packet radio group in Toronto, ON. For more information, contact Keith Gooby VE3OY @ VE3OY.

### TPK

An amateur packet terminal program for personal computers written by Gerard FC1EBN and others. TPK is designed to run in conjunction with a complementary program on a FBB BBS to give the user an up-to-date listing of BBS messages without having to connect to the BBS. The FBB BBS sends out a UI frame with the header information each time it receives a message or bulletin. The user station monitors the channel in an unconnected mode and records the UI frames so as to prepare a listing of the bulletins available. Any personal messages and any desired bulletins can then be downloaded by the user terminal automatically (often using compressed forwarding). This technique is designed to reduce congestion on the channel where normally many users would connect and download the same long list. (See FBB BBS)

### tranfer rate

See data rate.

### transparent mode

A mode of operation of a packet TNC that allows the sending of all possible binary sequences without fear of actuating commands in the TNC. Designed for the transfer of binary data files. (See converse mode)

### transport layer

Level 4 protocol in the seven level OSI computer communications protocol set. It controls the transfer of datagrams between two level 3 nodes via a number of intervening L3 nodes.

## TTL

Transistor Transistor Logic. This is the name for logic circuits which operate using 0V for a zero and 5V for a one to do binary operations.

## turnaround time

The total time that it takes for a radio and TNC to switch between transmit or receive so as to properly communicate with the other end of the circuit. (See TX Delay)

## TXData

Transmit Data stream fed to a modem modulator. (See RXData)

## TX Delay

The delay between the time the TNC issues a transmit command and the actual packet data starts. The TNC usually sends flags during this period although some node software may send an alternating 0101 bit stream that sounds to the ear like a pure tone.

## UA

Unnumbered Acknowledgment frame. A packet frame sent in an unconnected mode to acknowledge a connect or disconnect request.

## UART

Universal Asynchronous Receiver/Transmitter. This is an IC which is used in a computer to construct a serial port.

## UI

Unnumbered information frame. An Information frame without a frame number that is sent as a broadcast during a beacon, nodes broadcast, CQ, TPK item, and other similar frames. It is not acknowledged and there is no guarantee that it will be received.

## unproto

An unproto packet is a packet transmitted without expecting a response. Technically it is called a UI frame which means Unnumbered Information frame (frame = packet). If a packet station were to send a beacon for all to see or were to call CQ, the station would use a Unproto packet.

## uplink

A circuit from a user to a node or BBS initiated by the user. (See downlink)

## user channel

The frequency designated for users to access the network. This frequency would be devoid of servers and other nodes aside from the one designated. (See LAN, WAN)

## VADCG

Vancouver Amateur Digital Communications Group. An amateur packet group in Vancouver B.C. responsible for developing one of the earliest widely used packet protocols called the Vancouver Protocol.

## vancouver protocol

An amateur packet level 2 protocol developed in 1979 by Doug VE7APU and VADCG in Vancouver BC. Also known as the VADCG protocol.

## VC

See virtual circuit.

## virtual circuit

The service provided by a connection-oriented network. Virtual circuit data packets generally carry less header information than datagrams, since addresses have been specified at connection setup time. Amateur packet AX.25 level 2 uses virtual circuits. (See connection-oriented)

## WAN

Wide Area Network: This is a system where many servers and nodes may talk to each other. This kind of system is rugged in that communications would probably not be compromised if a single site went off the air. The major problem with this methodology is that if the only packet systems available are of this type then users, which present transient loading, will find that the WAN is unable to support massive intermittent loads during peak usage times. Since some users will be servers and some will be live keyboard stations the keyboard stations will probably not be the source of the truly massive loads. It would seem to the keyboard stations that the network was being intermittent for no particular reason. This causes frustration and leads to non-utilization of packet. (See *zoo channel*, *LAN*)

## weather node

A weather station linked to a packet radio node for remote monitoring of weather conditions by packet radio.

## wide area network

A data network covering a large geographical area often linking together many LANs on the same channel. A good (or poor depending on your viewpoint) example of a wide area network is the 145.01 MHz channel used on 2m 1200 bps packet. Wide area networks are also chronically affected by severe hidden transmitter syndrome (HTS) and exposed receiver syndrome (ERS). See also HTS, LAN, ERS. Compare with: LAN which covers a small area and which should have only one node or server on the channel.

## wide band packet

Anything faster than 1200 bps (baud). Some would suggest that anything slower than 56Kbps is not wide band or that wide band would be any speed that requires more radio bandwidth than the "normal" vhf fm transceiver used on amateur packet.

## Wink 'N Blink Mod

This is a modification to a TNC2 that allows that the CON and STA lights are used to monitor the RS-232 port's DCD and CTS signals. These signals act as PTT and Carrier Detect on the RS-232 so making this mod allows an observer to watch activity between the several TNCs at a single node site. This is an excellent diagnostic tool and is fun to watch. This circuit was described in the Fall of 1990 NEDA Quarterly and is also described under *Setting up a TheNET* in the *TheNET Network Node Guide* part of this Annual.

## wireline link

This is a connection between the modem headers of a pair or more of TNCs such that the TNCs communicate via their radio ports but without an intervening pair of radios. Because the modems are bypassed the TNCs may talk at a higher data rate than 1200. This circuit is described under *Building Nodes*, part of the *TheNET Network Node Guide* in this document and in an article in the Compendium.

## wirelink

Another term to describe a node to node or TNC-node to PC server connection via the serial ports, diode matrix or modem headers.

## W0RLI

Hank Orenson, W0RLI, author of a widely used packet bulletin board.

## wormhole

An amateur packet circuit between two distant points using commercial communication circuits such as telephone, satellite or microwave links.

## X.25

A CCITT standard protocol for the subscriber interface to a public packet switched network. Consists of two layers, link (level 2) and packet (level 3). The amateur AX.25 protocol is a highly modified version of just the link later of X.25; it does not have a packet layer.

## XON/XOFF

Software handshaking using characters such as Crtl-S/Crtl-Q to turn on and off a communications channel Compare with: RTS/CTS handshaking which uses hardware control lines on the RS-232 port to control data flow.

## YAPP

Yet Another Packet Program. A shareware terminal software package to interface a personal computer to a TNC. Contains scrolling, message handling, editing, and other utilities to aid the user on packet. Written and supported by Jeff WA7MBL. (See *Lan-Link, Paket*)

## zoo channel

A frequency where servers transmit to other servers (including, possibly, nodes talking to nodes) where hidden transmitter syndrome is not considered. (See *HTS, dedicated link, server*)

## 4RE BBS

See RE BBS, AA4RE.

## 7-Plus

A communications protocol written by DG1BBG for sending binary files over packet radio. (See *R95*)

Do not hesitate to verify any of this information and (as usual) send me corrections, missing words, better definitions etc. etc.

Notes: This Glossary of Packet Terms has been compiled from several sources in the public domain in addition to my own personal knowledge and experience.

The following are listed as authors of this glossary:
VE2BMQ, KA2DEW, N2IRZ,
Toronto Packet Group,
North East Digital Association Box 563 Manchester NH 03105,
Northwest Amateur Packet Radio Association Box 70405 Bellevue WA 98007
TPG Packet Users Guide
NEDA Annual
NAPRA Notebook.



—Tadd, KA2DEW care of NEDA

# NEDA Quarterly
# Compendium of past issues

The following pages have all of the technical articles printed in past Quarterlies that are not specifically ou dated. Some of the information is redundant with what is published earlier in this Annual. They are still ir cluded so as to show differing points of view. Some also deliver the information in different ways. This ma help with explanations for topics which are not altogether clear.

Also included are graphical articles for packet node sites that have been in past NEDA Quarterlies. Althoug the sites involved may have changed and the information is no longer current they still stand as decent exampl of network construction.

Disclaimer: The articles presented in this section are from the point of view of the authors. They do not ne essarily reflect NEDA policy. Please keep an open mind. If you would like to submit an article to the Quarterl or enter a letter tot he editor, send to the editor's address, the PO Box or NEDA @ WB2QBQ.ny.

## Table of Contents

# Death By Competition

*"I know of no safer depository of the ultimate powers of society but the people themselves. And if we think them not enlightened enough to exercise their control with a wholesome discretion, the remedy is not to take it from them but to inform their discretion by education."*

Thomas Jefferson
Co founder, United States of America

Before reading the rest of this editorial comment I would like readers to re-read the above statement, and forgetting about ourselves as amateurs consider how our country would have developed if not for the framework of our government which was forged by men like Thomas Jefferson.

It is no wonder that our country developed as fast as it did for information on what we had, how good it was and what made things tick was readily available to anyone who cared to know. Furthermore anyone who wanted to be a part of it all had only to express sufficient interest, intellect and time for involvement to be immediately planted firmly in some aspect of the grand scheme of things. No aspect was truly closed to those desiring involvement. Politics, society, industry, administration, agriculture, exploration and dozens of other areas were attractions for anyone with such leanings.

I would like everyone to consider that Amateur Radio is just as open and just as ripe for development and involvement as the example above. The problem is that without this understanding and an exchange of free information it cannot continue to attract involvement by those who are best capable of helping us grow. If this happens we will stagnate and wither in our knowledge base thus creating our own demise.

So what is the bottom line here?

**Involvement**
**Information**
**Cooperation**

It is hard to believe that anyone reading this who is a NEDA member is *not* already involved in packet to the point that they can actively promote our cause. And *What* you ask is our cause? Very simply it is to implement, promote and widely share those technical advances in networking and implementation philosophy that ultimately improve interconnectability of *all* packet services. Not just users. Not just Bulletin Boards. *Not* just special services like a DOSgate or DxCluster or CROWD port and *certainly not* just any *one network* containing such things within it! *All packet services* - inclusive - in its entirety - everywhere - globally. We must openly encourage the *wholesome discretion* referred to by Mr. Jefferson but also make sure it is an *'informed'* discretion. There is much 'power' to implement our technology with mostly just the use of the correct information, but there is no better way to efficiently use technical resources and hardware than to universally share whatever is available for all applications. This does require up front planning when doing things however as retrofits and add-ons rarely (if ever) achieve such efficiency.

***Death by competition***

The real killer of efficiency is non compatible independent implementation of redundant applications. Yes, we see them all the time in the private sector because in the commercial world most communications users are in *competition* with each other. They wish to have *independent* paths of communications that each individual group controls totally to their own financial (or image) gain. Certainly this is all right if such individuals can afford it all and it doesn't detract from the capacity of another group to reproduce their own services, but increasingly this is proving not to be the case.

A very recognizable example of this effect in the non-ham world recently has been brought to light by the needs of state governments in their statewide communications networks. The problem was the lack of coordinated government and public safety communications services caused enormous drain on statewide budgets. The statewide law enforcement agencies had a statewide network; the departments of education, transportation, health, social services, fire services, civil defense, administration, etc. ad nauseam had their *own* statewide communications network! The cost and maintenance of each system was variable because they were all just slightly different, but the overall cost of both implementation and operation was *enormous* not to mention the fact that these systems rarely had capacity to cross communicate! Budget administrators, upon uncovering this "turf protecting" policy creating essentially private (to each agency) networks for each agency, rioted at the misuse and inefficient application of taxpayer's bucks.

Centralized Telecommunications Agencies with the directive to create a single statewide network with more than enough capacity to handle *all* government agency needs were quickly created by state governments capable of fast response and dire need for financial recovery. Previous independent networks were rapidly phased out and integrated network use mandated. Systems created with long term objectives of integrating services quickly proved to be effective *and* drastically reduced the expenses involved with keeping themselves going. Public service agencies discovered the real value of "interoperability". Indeed, several state systems paid for themselves in record time!

How was this done? Cooperation and joint involvement was the key. It is interesting to note that in many states where such cooperative efforts were not mandated by some high level authority the wallowing in financial mire and inefficiency still goes on. This sure says something for cooperation in government eh?

There are really not many of differences in amateur packet networking and government communications services networking. We both work on limited funds, resources, manpower, support, time and a few other things as well (recognition and respect for personal sacrifice for others in the performance of public services for example?)

There are now in existence some real good examples of large scale amateur cooperative efforts that have achieved significant, even historical, events. Some of them have really opened the eyes of govern ment and commercial observers wh then copy our examples. New mod developments, super inexpensiv communications satellites in orbi serving globally and now wide scal data networking on a flea powe budget. Lets get on with things a innovators, supporters and educa tors of our fellow amateurs. Bu most of all, lets do it *together*!

**—Dana Jonas, WA2WNI**

*Copied from NEDA Quarterly aug 90*

---

## Speedup for Relay Keyed Radios

*Copied from NEDA Quarterly jun 90*

Here's a simple trick to improve those radios of yours with relay keying. While modifying for PIN diode switching would be great, some radios switch so many different voltages that it's not feasible to go to solid state switching. I tested several relays and found that while most will turn on in under 10 msec, the drop out time was at least 3 times that long. When the relay is turned off, current continues to flow through it via the snubber diode. The current dies out exponentially depending upon the relay coil's resistance and inductance, but in the meantime it keeps the relay energized. By adding resistance in series with the snubber diode you can decrease that time constant and speed things up. Adding this resistance will increase the voltage transient that the keying transistor will see, so you must be careful not to use too big of a resistor. The procedure to follow is this:

- Look up the Collector - Emitter voltage rating of the keying transistor hooked to the relay.
- Divide the voltage rating by the voltage applied to the relay and write down that ratio.
- Measure the resistance of the relay (measure it both directions and use the higher reading so that you aren't just reading the diode across it.)
- Multiply the resistance you mea sured by the ratio calculatec above, and round that numbe down to the nearest available re sistor value.
- Add this resistor in series witl the snubber diode that is acros the relay.

When you are done, the voltage transient that occurs at the collecto of the transistor when it unkeys wil be just under the voltage rating o the transistor. The time for the relay to unkey will be reduced by a factor equal to the ratio calculatec above.

**—Rich Place, WB2JLR**

---

## Pre/de-emphasis

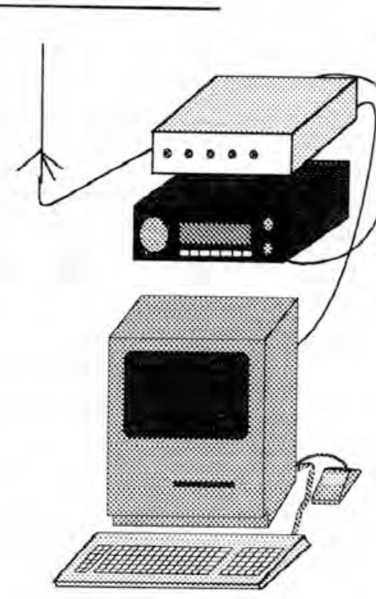
*Copied from NEDA Quarterly feb 91*

Using 1200 baud modems and FM radios there is a theoretical advantage of not using pre-emphasis and deemphasis. Tests run show the advantage to be in the neighborhood of 2 dB.

Knowing this, and wanting my User port to work its very best, I made a serious mistake; I disabled the pre-emphasis and deemphasis. To get the 2 dB advantage you must disable the emphasis at both ends of the link. Defeating it at one end of the link but not the other results in delay distortion of the data, which can be disastrous. Depending upon the modem chip used, the radios will either work marginally, or not at all. Since most users connect to their TNCs via the microphone and speaker jack, they have pre-emphasis and deemphasis, so the User port radio in the node needs it too.

On the other hand, if you have a long haul backbone link and think that another 2dB will make a difference, this would be an easy way to get it. Just be sure both ends of the link get the changes.

**—Rich Place, WB2JLR**



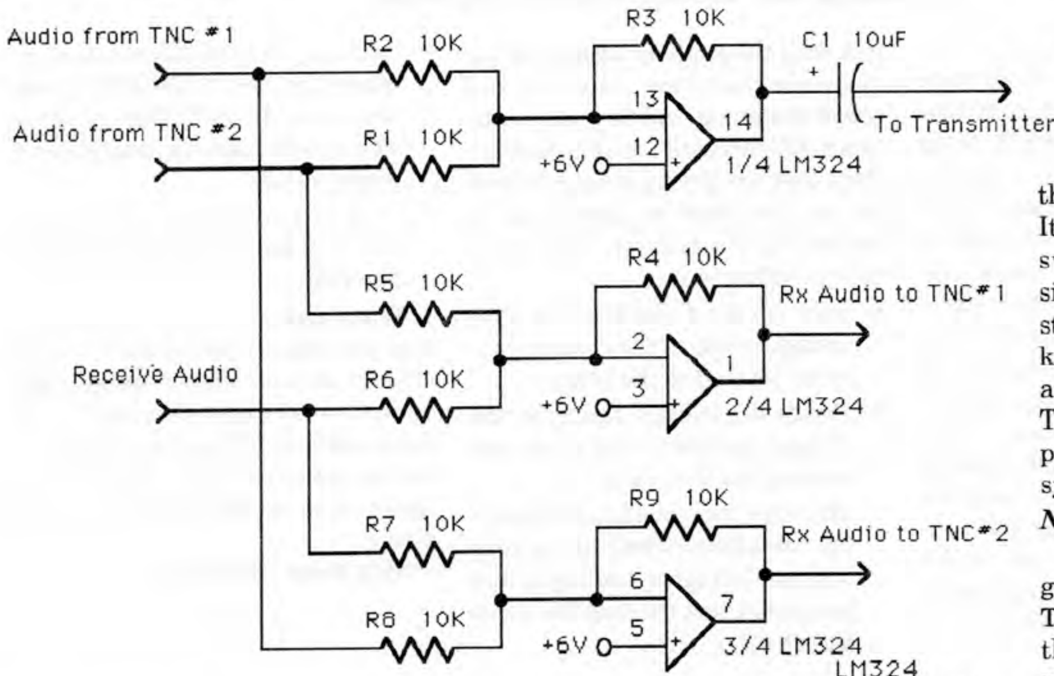
# Two TNCs on One Radio

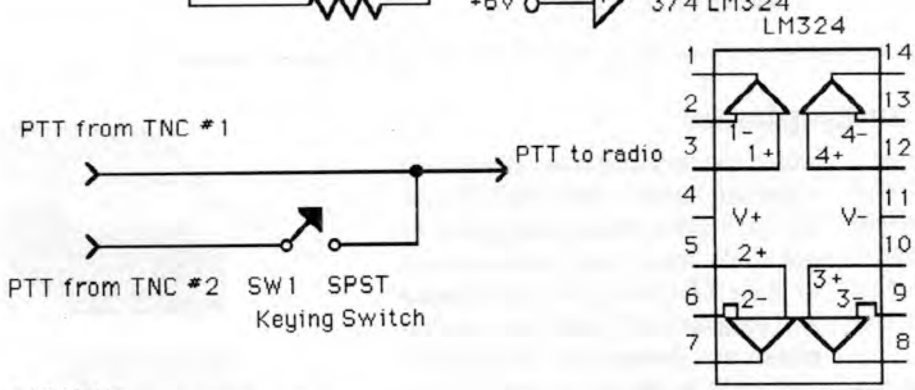
*Copied from NEDA Quarterly June 92*
4-30-92 Rich Place WB2JLR

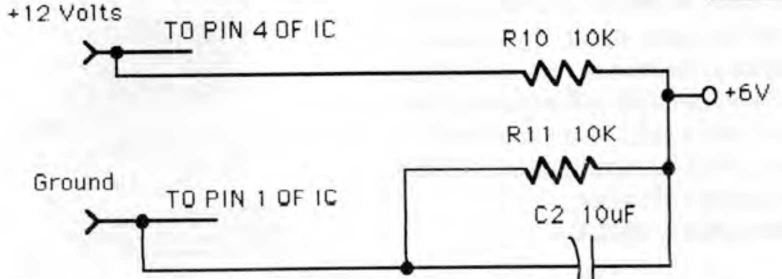
Here is a simple circuit that will let you interface your own personal packet station to the node in your home. I use it for interfacing my packet station, WB2JLR and WB2JLR-4 (PMS) to the CANDGA node. It lets me monitor 144.99, the *user port* frequency, without requiring a second radio. I can also *connect* directly to anyone on 144.99, as well as connecting to the node.

***How it works***

The circuit uses a quad op-amp, which could be any garden variety part such as an LM324 or TL084. Three sections of the op-amp are used a summing circuits to combine two audio sources into one. TNC#1 is the *user port* for the node, and TNC#2 is another TNC for your personal use and possibly PMS mailbox. The transmit audio from each of the TNCs is summed by the first section of the op-amp to provide a combined audio signal to the 2 meter radio. In this way either TNC can transmit out on the *user port* frequency. The second op-amp section combines the receiver audio with the transmit audio from TNC#2, and feeds it to TNC#1. This allows TNC#1 (the *user port*) to hear both the 2 meter signal and the signal from your own TNC. The third op-amp section combines the receiver audio with the transmit audio from TNC#1, and feeds it to TNC#2. This makes it so TNC#2 (your station) can hear both signals coming in on 2 meters, and the signals from the node.




An added feature is the ability to control the key line. Normally a switch connects the PTT line from the two TNCs in parallel so either of them can key the radio. It is possible to open the switch so transmitted signals from the personal station to the node do not key the 2 meter radio and go out over the air. This may be desired to provide privacy when sysopping the node.

***Notes:***

Each circuit has unity gain in all directions. The feedback resistors on the op-amps could be made variable if gain adjustment was desired.

It may be necessary to add series capacitors in all of the input and output lines, depending upon the TNC and radio. If the units are already AC coupled then no additional caps are needed.

*[Another approach (more costly) to getting the node owner a connect into the network for his own TNC is shown in the TheNET Resource Manual under the chapter Burning EPROMs and Putting It All Together.-ed]*

—**Rich Place, WB2JLR**

# LAN Architecture or Should I use a Beam or an Omni?

*Copied from Quarterly December 1991*

There has been controversy since the early days of amateur packet radio as to whether a packeteer should use an omni or a beam. I'll try to resolve that in this article. I think I can show that in modern metropolitan packet radio a user station should utilize a beam if possible.

Most packet radio operations in the U.S. occur on 2 meters. In most situations when a packet user turns on the radio and TNC the station will hear other sites. Some of those other sites will hear yet other sites and so on. In most cases there will be more than one server, node, digipeater etc. on the frequency. This is far from ideal. In this case planning either has not taken place or has not been effective. For the purposes of this article I'm going to focus on LAN channels where planning has taken place and where we're now trying to make it effective.

***Fixed # of stations, all stations hear each other***

There are two LAN architectures available to users of current day off the shelf TNCs. The first is the same architecture used in commercial CSMA ethernet systems. In this all stations can hear each other. All are basically omni. All have equal priority and may make a local decision on when to transmit and be pretty sure of not colliding with another station.

This kind of LAN is possible on Amateur Radio only where spectrum space is not a premium and all of the packeteers are in a planned region. This may be the case in a small community, not a major metropolitan area.

***One server, stations don't all hear each other***

The second architecture is one in which it is not possible to predict how many active stations can hear each other. Using standard TNCs the only form this LAN can take and still function with better than 20% efficiency is the form in which

- one station on the LAN can hear and talk to all of the other stations
- that one station is the only server on the channel.

These two points are usually the case on *designed* LANs because all of the users access one node or one server on a given frequency.

*It can be proven experimentally or via simulation the only way to efficiently use a CSMA system with hidden transmitters requires the total utilized channel time must be less than 20% of the available time. If the server is not a hidden transmitter to anybody then it may use as much time as it wants. Only the user stations need divide the remaining time by 5.*

In this scenario a beam should be used for all user stations if possible because

- it won't affect the channel utilization calculations at all whether the user stations can hear each other or not, so long as the user stations don't transmit very much
- and the geographic coverage of the LAN may only be controlled if the individual user stations cooperate by using beams.

The two drawings show geographic area used by a LAN where the users have beam antennas and by a LAN where the users have omnidirectional antennas.

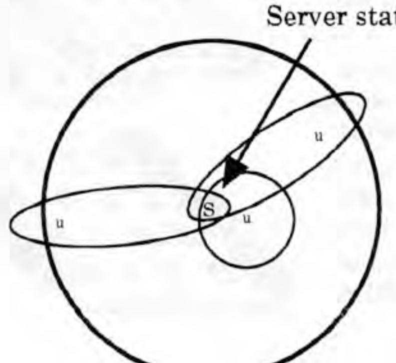
It can be seen from the drawings that if a *large* group of people operated to a server station, using omnis, the size of the LAN would be around twice as wide as the LAN where user stations are operating with beams. This would lead to one quarter the number of possible LANs on each 2m frequency in the metro area.

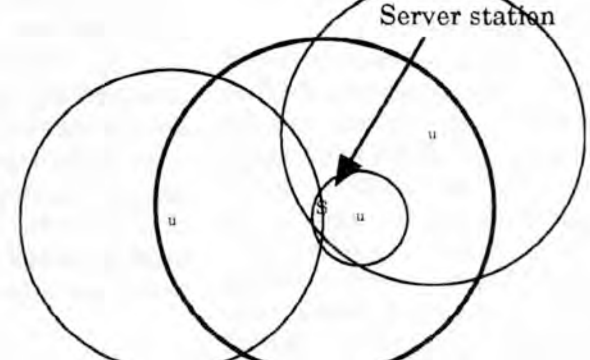
Since the most efficient (highest data bandwidth) utilization of a frequency, with a given baud rate, is with a single-server type of LAN, the amount of data bandwidth available on a frequency would go up by a factor of 4 if all user stations used beams. (Area of a circle is $\pi r^2$)

Since most 2 meter LANs are not planned there is an immeasurable improvement to be gained by running cellular LANs, with users having beams.

Your next phase, after you prove out the cellular concept is to start reducing the size of your LANs. By reducing the max number of stations on a LAN you increase the performance you give each station. Baud rate is secondary. Obviously having a digipeater on the highest building in the city is right out (hi).

**—Tadd, KA2DEW**



# The HexiPus Story

*Copied from NEDA Quarterly feb 91 - updated feb 94*

By the summer of 1988 KA2DEW and WA2WNI had been conversing regularly via packet for about 2 years. KA2DEW lived in a rented house with John Painter in Nashua NH. WA2WNI lived with his wife and two kids in Valatie NY near Albany. The path they used was often digipeating through two very well placed digipeaters over the distance of 150 or so miles. Other times they would maneuver their way through the 221.11 backbone and still at other times they would rely on the PBBS systems to deliver their mail overnight. Both WA2WNI and DEW had W0RLI PBBSs of their own and both ran those PBBSs with a separate 2m user channel and 440 or 220 'backbone' channel for forwarding and whatnot.

The reason the summer of 1988 is important is that it was around that time the level of frustration from the difficulties of passing mail and from trying real time packet from Albany to Nashua reached a threshold where the two decided to do something about it.

John Painter is important to this story because as he was sharing a house with DEW he had observed these goings on. John, or Tjp (The john painter) as he likes to be called, is a technical person. He makes his living consulting to various companies that need custom VAX/VMS software to do tiny little nefarious tasks like graphics or dual ported diskdrive support etc.. John had observed Tadd, KA2DEW, on the phone with Dana, WA2WNI, for all hours of the night trying to find packet routes that work.

Now, Tadd and Dana were avid followers of the goings on in the New York metro area and had seen the application of TheNET in multiport nodes under the auspices of the Eastnet Backbone Network. Doug, WB2KMY (Kiss My Yagi) had taken Tadd and Dana under his wing to educate them as to technical solutions using the multiport TheNET design. Tadd and Dana learned well. One thing they learned quickly is that building those icky diode matrices is a pain in the derriere.

Tjp was in on one of these construction projects (it was unavoidable) and decided that this was a perfect application for his Macintosh and McCAD program. Presto chango cherio and the Octopus was born. John made 200 of the boards and he made them 8 port figuring that if 4 ports was good, 8 would sell like hot cakes. This was all based on Tadd and Dana's prediction the Octopus would make node manufacture easy enough that people would use them. [They were lying. They just wanted the boards hi]

Well.. 2 years later the Octopus boards are all sold out. It was time to make more. Many things have happened in the last 2 years. For one thing, NEDA was born out of the incredible growth in multiport nodes that occurred in the vicinity of Albany and Nashua [hmm...]. John has moved to Kansas City, Tadd recently to Potsdam NY. So.. the NEDA board of directors made a general statement that a replacement Octopus was required.

Rich, WB2JLR, took the project and ran with it, creating the HexiPus. The reason NEDA wanted a new board were several fold:

1> 8 ports was deemed electrically unsound due to loading problems when one TNC is driving 7 others. 6 ports was deemed more appropriate.

2> The Tjp Octopus didn't say anything about NEDA.

3> The Tjp Octopus was made small to keep costs low. The size is uncomfortable for some to construct.

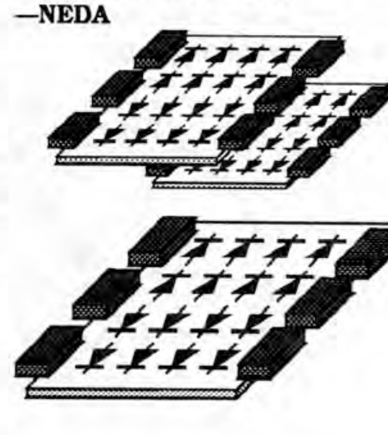
4> John Painter (Tjp) wasn't around when the time came to make these boards. He has since been back in contact and is planning on working on other projects for NEDA.

So, NEDA acquired 200 HexiPus boards. They say NEDA all over them. The club formed a committee headed by WA2TVE, Howie, to mail the boards. The price for the boards with diodes as a kit was $22.95 plus $4 for shipping. The price for the boards with diodes and 9-pin Dshell connectors was $29.95 plus $4 shipping.

The Dshell connectors are female and required a custom cable to plug between the Hexipus and a Tiny 2. If you were using a MFJ 1270B or AEA PK88 (for ROSE) you could get a standard PCAT to modem cable and modify it to put in the missing pin connections.

The HexiPus sold very well and in two years were totally gone. Shortly after the Hexipus was introduced NX2P electronics introduced a board that supported five TheNET TNCs or five ROSE TNCs. This board sold well via PacComm and from NX2P direct. (had JLR and DEW known that NX2P was coming out with his board they probably wouldn't have made the Hexipus). Shortly before the HexiPus stock ran out Amateur Networking Supply started selling a six port matrix that, unlike its predecessors, plugs directly into the back of a stack of TNCs, using no wires. It looks like NEDA is no longer required to supply node builders with the needed diode matrix hardware. There are also rumors that a TTL-only version of a TNC interconnect might be available in the future.

**—NEDA**



## An Improvement to the Wink-N-Blink Mod

*Copied from NEDA Quarterly feb 91*

In the last Quarterly (See Resource Manual) KA2DEW provided a simple TNC conversion that modified the STA and CON LEDs to show RX and TX data being passed over the RS-232 port. This was part of his 'Wireline Linking' article. As you may know a TNC running TheNET code does not normally use the STA and CON. The WNBM (Wink-N-Blink Mod) can be very useful as a diagnostic tool to show matrix activity such as which port is sourcing matrix data and when. [See *"Burning EPROMs and Putting It All Together"* in the TheNET Resource Manual in this document]

I did find it aesthetically more pleasing however to remove the right hand 3 LEDs in my Tiny 2s and rearrange them so the LED color order from left to right across the front of the TNC was Green Red Green Red and the power LED became the remaining Yellow. The 1st pair of Green/Red show RF port Rx/Tx data and the 2nd pair then show the Rx/Tx data appearing on the RS-232 port. When the TNCs from your node are physically stacked on top of one another it creates and easy to visualize pattern so you can quickly familiarize yourself with what normal data throughput should look like. (Not to mention the fantastic light show that will entertain visitors to your node setup when the node matrix is extremely busy!)

Several useful things cropped up almost immediately after I had put the WNBM in place. First was I noticed that one port was sending many Tx bursts without responses from anything. This could not possibly be caused by **that** many matrix collisions in a row! Indeed it wasn't as I quickly found there was an open connection across the matrix between two TNCs. The receiving TNC wasn't getting the data at all so the sending TNC kept trying until the alternate default path took it a different way around the matrix.

The second item of interest was to actually observe 3 TNCs passing data that should only have gone through 2 TNCs. Alas there was a routing glitch caused by a locked value. Data in one direction was going through TNCs A and B while the return data was going through TNCs A to C to B. Of course one can observe by looking at parm and route tables all this anyway but it takes a bit of looking to see what paths are set up to do what.

One of the observations I am keeping an eye on now is the effect of circuit choke on the matrix. I have noted at times the RF ports doing all sorts of activity, and the matrix doing nothing at all.

**—Dana Jonas, WA2WNI**

## KISS: What Is It and How Do I Get Out of It?

*Copied from Dedicated Link Sept 92*

Over the last several years there have been a number of BBS messages along this line: *I put my TNC into KISS mode and can't get it out. What do I do?*

KISS is an acronym for "Keep It Simple Stupid!" It is the name of a protocol designed for TNC that allows the TNC to be totally controlled by a computer (like a PC). It is used for TCP/IP as well as for some BBS programs. KISS messages always start with the character $C0 (not a standard ASCII character) and end with a $C0. The second character is a *command byte* which indicates the kind of message. Some messages set things like TXDelay. Others indicate that data (i.e., a packet frame) is to be carried. A TNC operating this way relies on the host computer to construct the packet (in all of its gory details). The TNC simply converts the packet from asynchronous serial to synchronous.

It is not possible to generate KISS messages from a 7 bit terminal. There are a number of programs which interface with TNCs using KISS including TCP/IP, NOS, MSYS and G8BPQ.

Most TNC2 clones now come with KISS resident in the EPROM. For MFJ1270/1274 TNCs, the command is KISS ON. The second step is a TNC reset. This is done either with a *RESET* command or by turning off, then back on. The TNC comes back on in KISS mode. It appears to come up at the same (terminal) baud rate that it was at prior to the reset. Note that once KISS is on, it no longer recognizes your normal computer and *KISS OFF* has no effect!

I do know of three ways to get your TNC back into normal operation:

- use a KISS computer interface; This involves running a computer program on your personal computer that talks KISS and asking your computer to take the TNC out of KISS mode.
- Allow the battery-backed RAM to discharge. This can be done by disconnecting the battery with the TNC off. On MFJ TNCs, it may be easiest to slide a piece of paper between the battery and the battery contact then letting it sit for 10 minutes.
- On a PC Compatible, hold down the ALT key and type 192 let up on the ALT key. Press the ALT key, type 255, let up on the ALT key. In other words: ALT-192, ALT-255. *You must use the numeric keypad numbers on an IBM compatible.* This is an old trick. It's sending a two byte character sequence equivalent to decimal 192 and 255.

Thanks to Stu, WD4ECK for the ALT key sequence tip. He heard it from a friend on the AMSAT net that meets on 80 meters every Tuesday night...

**—Jim Wagner - KA7EHK @ WG7J**

# Kantronics D4-10 UHF Radio

*Copied from Quarterly nov 91*

These are some notes based on our experiences (over the last 10 days) of making the new Kantronics D4-10 radios work at 19.2 KB. Our group is using these radios to build a metropolitan area network that includes a full duplex UHF digital repeater, a G8BPQ switch, and high-speed links to other areas.

### *The Hardware*

The Kantronics D4-10 radio (not to be confused with the 2M DVR2-2) is a UHF radio designed for data transmission. Kantronics has optimized the D4 to move data at 19.2 kilobaud within a 100KHz channel with a 60KHz receiver bandwidth. It's crystal controlled on two channels nominally in the 430 MHz range and is rated at 10W output, although my Bird says more like 15. It has a <much> better receiver than the 2M DataRadio.

The interesting feature of the radio is it has a TTL level I/O port designed for direct FSK. TXD will modulate +/- 10KHz around the center frequency, and RXD is derived from a data slicer. The squelch circuit is very fast (~2ms) and is available as DCD on the connector. And not to worry — the TXD line is shaped, so the FSK isn't based on square waves. The bandwidth is within FCC limits (100KHz) for the 70cm band.

### *Our Approach*

Since the radio is designed with digital levels in mind, my first testing with two of the beta models last March focussed on the simple approach — using an 8530 SCC chip to generate HDLC frames and shoving those frames directly into the D4 TTL port. To my surprise, it worked!

Since then, we've decided to base our network, at least for now, on that approach. If and when modems arrive that can do a better job, we'll probably use them, but for now the savings of $100 per radio by not buying 19.2K modems outweighs the relatively small advantages the modems offer (mainly in more reliable DCD, but even that's open to question).

### *Using the Ottawa PI Card*

Our first experiments used the Ottawa packet group's PI card (a DMA driven, 8530 plug-in card for the PC bus). Interfacing them to the D4 is a snap — just wire up a five conductor cable between the two, set up NOS, and you're in business.

### *Interfacing the Data Engine*

However, the PI card only works in PCs, and (at present) only works with the KA9Q NOS software. We wanted to have an alternative packet generator available, so we focused on interfacing the DataEngine to the D4, sans modem. That also proved easy to do.

Kantronics makes a small jumper board (for about $25) designed to allow the DataEngine to work with an external modem. Just get one of those, jumper it as a type "A" modem, and add a CMOS chip to divide the RXClock signal by 32 to feed back as TXClock.

More specifically, we used a CD4020 with the clock connected to pin 5 of the internal modem header and the divided output connected to pin 6. 12 volts is available on the jumper board; we used a 1K resistor and 5.1 volt zener diode to power the 4020 chip with the necessary TTL level. The chip can be mounted "dead bug" style on the jumper board; the whole thing makes a very nice package.

### *Software Speed Selection*

With either of these approaches, the actual data rate on the radio link is totally software-driven. It's just a matter of what speed you program the baud rate generators to. We've moved packets at every supported rate from 110 baud to 28.8kb (28.8 doesn't work very well, but it does work), simply by using the appropriate "attach" command with NOS, or "modem" command with the DataEngine.

### *Results*

First, these radios are as fast as Kantronics says they are. The PI card driver allows TXDelay to be set in 1ms increments, and we've found that a TXD of 4ms works. We're using 5ms to provide a bit of margin. Remember, this is <milliseconds>, not the (milliseconds times ten) that most TXD values represent.

Our initial testing shows very respectable throughputs are easy to achieve, at least across the room. Using NOS on 286 or better machines, and a RAM disk to avoid mechanical slowdowns, we've consistently seen FTP file transfers of binary files go at 1600 or more characters per second between two PI cards. Note, though that this is on a totally clear channel, with all parameters set wide open. In the real world, neighborliness will require backing things off a bit.

We do see some packets that don't get acknowledged; the resultant retries and backoff can slow things down a bit. We're investigating the problem, but at the moment don't have any clues.

We only began testing the combination of a PI card station talking to a DataEngine station last night. The throughput there has been more like 650-700 characters per second. We're not sure why this great a difference exists. Possibly the problem is that the DataEngine-to-host link on the serial port is running at the same speed as the radio link, that the computer just can't keep up with 19.2 serial data (we're not using 16550s, so even though the machine is a 386, this is quite possible), or that the asy code in NOS is be less efficient than the PI driver. We're going to continue looking at this.

### *Digital Repeater*

We're turning two of the D4s into a digital repeater. Our input frequency is in the 420 MHz range, with output on 430 (a 10MHz split).

The interface is actually very easy but it took a <lot> of trial and error to get things working.

The problem in a nutshell is that although the digital port is advertised as "TTL", it really isn't. The PTT line is fairly standard — to key the radio, bring the line to ground and sink about 5ma.

However, the DCD, TXD, and RXD lines are all tied to op amp stages set up as comparators. Although they are biased to switch with TTL levels, we found that using 13.8 volts is much more reliable.

Also, it's not obvious from the documentation but the FSK keying circuit actually has <three> states, not two. Grounding TXD shifts 10KHz down, pulling it high shifts 10KHz up, and something between will put out the nominal frequency. This cost us a lot of time — our first interface <seemed> to be modulating the radio, and we could hear the data on the receiver's speaker, but there was no RXD. It turned out we were shifting between -10 and center — enough deviation to make noise, but not enough to trigger the data slicer.

Anyway, the answer was simple once we figured it out. We used a CD4049 hex inverter chip. Two cascaded sections provide PTT from the DCD input. Two more sections interface RXD to TXD. The chip is powered from the same 13.8 volt supply as the radios.

The critical thing it took a while to figure out is that RXD <must> be tied high at the input to the inverter. Not doing this is what caused our indeterminate keying state. 22K between Vcc and RXD worked fine for us. DCD would probably also benefit from a pull-up resistor, but seems to work OK without one.

Of course, you'll need extra circuitry for control and time-out timers. We're also looking at ways to come up with a more reliable keyin scheme; if the repeater is brought u by a rogue carrier, that will shut th whole network down. A circuit tha detects a packet's opening flags an trips a short timer (maybe 1 seconc AND'ed with the squelch-derive DCD is probably the simplest an swer.

The repeater turn-around i pretty quick. We've been able to re liably send packets through it wit a TXDelay of 10ms. Obviously, hang-timer won't work in a systen based on carrier-derived squelch, s the repeater output is indistinguish able from any other packet station

Repeater identification will b handled by the G8BPQ node that wil be interfaced with the repeater.

**—John Ackermann, AG9V, and th Miami Valley FM Association Dayton, Ohio". Republication an distribution is no problem so lon as credit is given.**

---

## New Deal Available on UHF HTs (and still available 12/93)

*Copied from Quarterly may 91*

About a year or so ago some NEDA member may recall how we were frantically buying up Wilson MH400 UHF HT's we then put into links in a number of places. WA1TPP did an article for the Quarterly that showed us how to speed up and utilize the rig for packet as efficiently as possible and thus a number of links and special ports were put on the air as a result of these inexpensive little 2 watt hand held rigs. (The rigs only cost us about $80 each!)

Well, it appears that the original vendor has struck yet another bargain with the folks at the Wilson/ Regency warehouse and managed to make another incredible bulk buy out. The deal this time is for UHF Regency MCPU-404 handhelds. These rigs are new closeouts that are 4 channel, 4 watt crystal controlled handie talkies. While they don't come with antennas or batteries, the vendor is selling them to us for the incredible price of $49.95 each plus shipping. If you should choose to use the rig as a portable and not cannibalize it for a packet link the vendor will sell you a drop in charger for $29.95 and batteries for $20 each. The batteries, by the way, are the same ones used by the Yaesu FT 203 and 209 series radios, known commercially as a BP-4. The crystals it takes are HC-18u size with wire leads and the unit also has a factory installed jack that will take a plug in CTCSS board.

Please contact the vendor directly as we will not have sufficient time to put together a NEDA bulk buy like in the past. He was kind enough to let us in on the lower pricing because of being on his preferred buyers list from the previous 30 or 40 odd units we bought before. Please move quickly on this as the vendor might do something like raise the price or sell the whole lot to someone. The address:

Torg's Electronics
9280 W. 360N
Shipshewana, IN 46565
Phone: (219) 768-4406

The proprietor is Mr. Torgeson so should you give him a call to get in on this, make sure to pass on the regards and best wishes for all the NEDA members already taking advantage of the bargains he has provided us in the past. Who knows? With a little effort maybe we can get Mr. Torgeson to get his ***own ham license***!?!

Oh, one more thing. Manuals for the unit should be readily available from *Regency Electronics*, or Torg's can most likely provide you with a copy for a nominal extra cost.

**—Dana Jonas, WA2WNI**

# KNOX:WB2QBQ Node

*Copied from Quarterly aug 91*

Bob, WB2QBQ owns and operates the KNOX node in Knox New York. The node is located in and around his house. He is currently operating seven radios and antennas for the node itself and other radios and antennas for packet, FM and HF. The system has sprung up in a very short time, from a digipeater in 1989 to a seven port node in the summer of 1991. Bob spends a great deal of his ham radio time tinkering with the radio and antenna systems as well as playing on packet. He also derives much enjoyment from having his system used by others as is evidenced by the amazing light display on the TNCs as the node is operating. Visitors to his site have remarked at how the action never seems to stop.

Servers that are almost always using the KNOX node as a through path include the K2TR DxCluster, WA2TVE BBS, WA2PVV BBS, and WA2UMX BBS.

Recent additions to the node include a 900MHz 9600 baud link to Albany. The pre-existing 440MHz 1200 baud link is still in place but will be reallocated once testing on the 900MHz link is competed. Bob expects to link to another site using the already in place 440 gear.

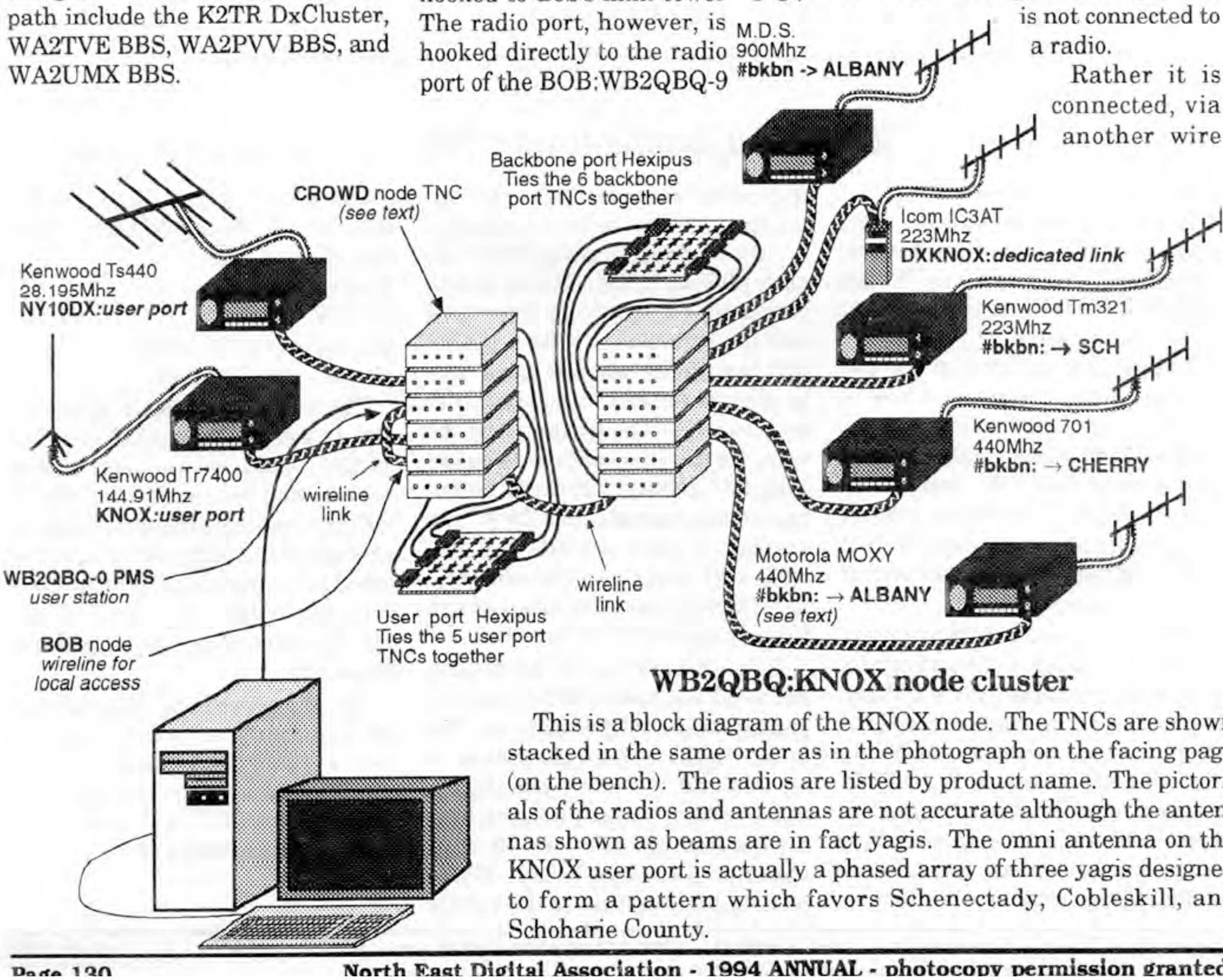
A recent problem that Bob had is that his TS-440 HF radio has failed. As of this writing that radio is on the way to Kenwood but will hopefully be back on at or shortly after publication date.

***Wireline Links***

KNOX node has two separate wireline links. The first connects the home station with the network node. Bob's home station is the third TNCs down in the left stack as seen in both the photograph and the diagram. The TNC has regular PacComm PMS software and is hooked to Bob's mini-tower PC. The radio port, however, is hooked directly to the radio port of the BOB:WB2QBQ-9 TheNET TNC. Stations who connect to KNOX and do a nodes list will see the BOB node. If you connect to the BOB node and then connect to WB2QBQ you will be connected to Bob's home station and the "*** Connected to" message will appear on the PC's screen. However, there are no radios in between WB2QBQ-0 and BOB: WB2QBQ-9. The circuit is made with a few 12 inch pieces of wire connected to the modem headers of the two TNCs. That connection is described elsewhere in this issue. (See *Wireline Link*).

The other wireline link at the KNOX node is between the two separate node stacks. The right hand node stack consists only of backbone links and the DXKNOX node. The bottom TNC in the right hand stack is connected to the other five TNCs through the Hexipus but is not connected to a radio.

Rather it is connected, via another wire




## WB2QBQ:KNOX node cluster

This is a block diagram of the KNOX node. The TNCs are shown stacked in the same order as in the photograph on the facing page (on the bench). The radios are listed by product name. The pictorials of the radios and antennas are not accurate although the antennas shown as beams are in fact yagis. The omni antenna on the KNOX user port is actually a phased array of three yagis designed to form a pattern which favors Schenectady, Cobleskill, and Schoharie County.

link, to the bottom TNC on the left hand stack. Traffic that travels across the backbones through Bob's house, but that is not destined for the KNOX node, travels from radio to TNC through the matrix, to TNC and back out the radio, without ever crossing that wireline link. The only traffic that goes through the wireline link is traffic that goes to the BOB node, the KNOX node, the NY10DX node or the CROWD node. This isolation between the node stacks allows the TNCs on each matrix to have less crowding in their communications between each other. More importantly the wireline link concept allows very large node complexes to be built without regard to RS-232 electrical considerations. A TNC is designed to hook up to one computer. Hooking it up to five other computers (TNCs) through the matrix is wonderful for building networks but stretches the capabilities of the TNC's RS-232 port. Because Bob wanted to have four backbone links, a DxCluster port, a CROWD, a user port on two meters and an HF gateway, *plus* the BOB node he had to break up the node cluster into two separate nodes.

There is another advantage to having the wireline link between the TNCs: It makes a *real* nice light show!!

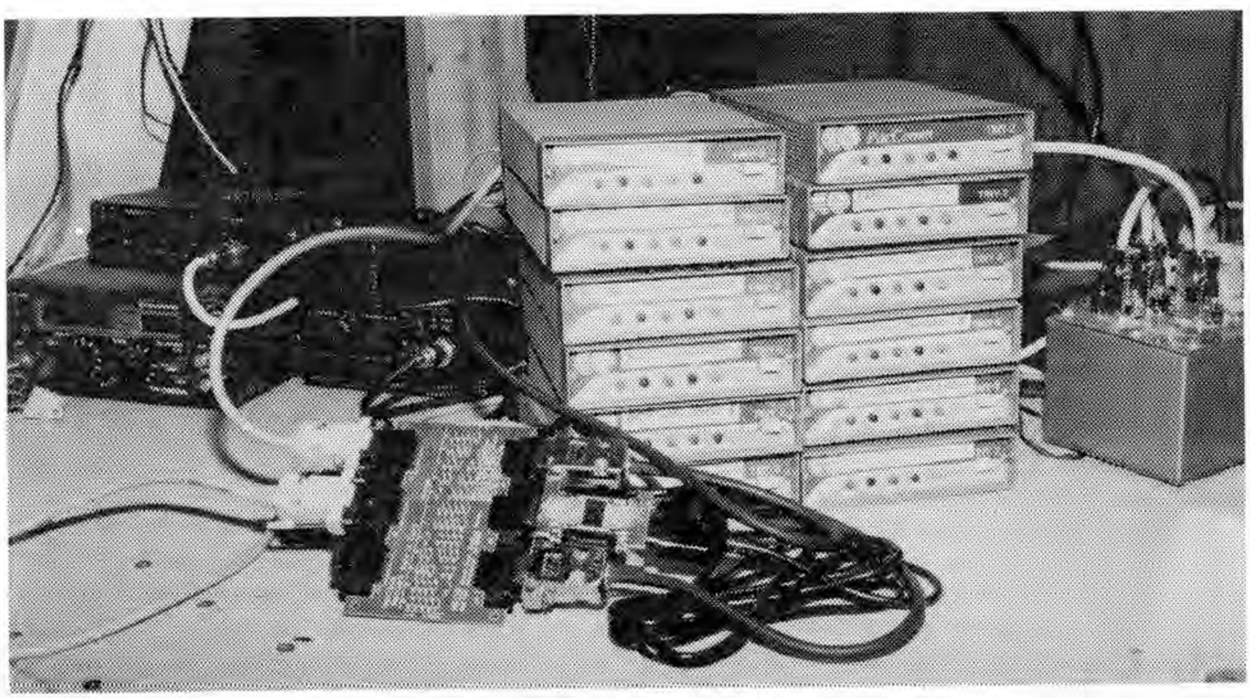
### *CROWD node*

The first TNC on the left hand stack is the CROWD node. This TNC is plugged into the Hexipus matrix with four other TNCs but is not running TheNET software. Instead it's running NORD><LINK mini-conf software. It is fully TheNET compatible but instead of handling network traffic, the CROWD node's purpose is to allow stations who connect a round table conference capability. The name CROWD was coined by WA2WNI in 1988 and has since been widely used for this function. There are CROWD nodes at many sites in the north east. Each CROWD node has a different callsign and usually only one CROWD node will show at any node site. If you and a few friends want to have a conversation as a group you can all connect to CROWD at KNOX and each ham will see all of the text typed by each of the others. It's great fun and an excellent utility for emergency traffic handling.

Simply connect to the KNOX node, then connect to CROWD. Now type slash W "/w" and the CROWD will give you a list of the other stations connected in and will announce your presence on the CROWD. If there is nobody on, hang in there and hopefully somebody else will check in during the two hours before you time out. If you type anything you reset the timer for another two hours. If you check in regularly you'll eventually start something and before you know it you'll have a nightly CROWD crowd. Also check out the CROWD nodes at STMFRD and CANDGA nodes.

Connect over to KNOX and play around with the R and N commands. For novices at TheNET network hacking you'll find the diagram very helpful for learning how those commands work and how TheNET networks are put together. Have fun!

**—Dana, WA2WNI; Tadd, KA2DEW; Bob, WB2QBQ; and Bob, NQ1C (took the picture)**



**Close-up view of the KNOX node TNC stacks**

## Suggested Packet Settings

*Copied from NEDA Quarterly feb 91*

Are you having trouble with more retries than you should when the frequency is busy? Or maybe one station is hogging it all and you can't figure out why there is no room for your packets. Here are some ideas that may help the situation. Get together with your fellow packeteers that are operating on the local user port and review the settings of the timers in your TNC's. Much of the following information was contained in a couple of articles by WB6RQN in 73 magazine some time ago, and we have found the suggestions to be very helpful on the VNH user port. It does require a cooperative effort of all the users.

***TXDelay***

Run it as low as you can and still work the stations you normally converse with. I can run my TXDelay at 8 quite reliably when working VNH. VNH has a very fast receiver, however a setting of 20 to 30 is required to accommodate some of the local stations (I'm working on them, hi) that connect to me in the occasions that they can work me direct and not through VNH.

Note: True DCD in the TNC will drastically improve your receive response time. True DCD boards are available from PacComm for about $26.00 and these will work with most TNC's that have the 3105 modem chip. It's very easy to install, you just pull out the 3105, plug in the board, then plug the 3105 back into the new board. Open up your squelch, and away you go. The MFJ TNC's are represented to have true DCD already installed, but I have found it does not work effectively with some receivers. The only way you can tell is to try running your squelch open, and see if the DCD light shows on receiver noise. If it doesn't light with your audio gain control set properly you've got it made. Running with the squelch open makes for a faster receive system.

***Non-Persistent CSMA***

If your TNC has 'non-persistent CSMA' (Carrier-Sensed Multiple Access), use the following settings. You can determine this by looking at your commands list. If you have it, the commands PErsist, PPersist and SLottime will be in your DISPlay.

SLottime: Set to equal TXDelay

PErsist: Set to equal 255 x (1/n) where n = the number of stations using the channel other than you.

PPersist: set to ON

DWait: Set to 0

If you do not have non-persistent CSMA then DWait is set to twice the value of the highest TXDelay setting of all the stations in the area. This will do a similar job to what non-persistent CSMA does.

***FRack***

Set to 5 or greater, 10 works pretty well if it's busy.

***MAXframe***

Set to 1. This gives everybody's packet a chance at the frequency, in its respective order. Plus it cuts down on retries if the frequency is loaded with hidden transmitters. I have found this one to be a highly controversial issue. 6 people will give you 6 different answers on what is best for MAXframe, plus the square of that number of reasons why one is better than the other. All I can say is 1 works best for me. If the frequency isn't busy and there are no hidden transmitters around hitting your packets with big blasts of data then higher numbers will work fine.

**—Cal Stiles, W1JFP**
**Mount Ascutney Packet Radio Association, (MAPRA), The VNH folks.**

---

## New Macintosh Packet Program Released

*Copied from NEDA Quarterly June 92*

*Virtuoso* is a Macintosh communications program written specifically for packet radio. it has features that packet radio operators need, and also packs in a lot of bells and whistles to make packet radio communications smooth and effortless.

The program was written by James E. Van Peursem KE0PH, who started his Macintosh packet radio career using programs written for general communications (you know the ones). This was okay for starting but it turned out to be quite a bother and he found himself spending more time trying to get the program to do what he wanted it to do than actually communicating on packet. He also has seen literature for other packet radio communications programs, but the seemed to lack the power and ease of use that he had grown to love in a Macintosh program.

James says that *Virtuoso* is his solution. It packs all of the power of the best of the programs and is written specifically for packet radio, so operating has never been easier. Some of the features now implemented are:

- Powerful scripting to automate routine tasks,
- Startup and exit scripts,
- Save incoming text to disk file,
- Append a selection of text to an existing file,
- Print a selection of text,
- Find the last time you heard someone,
- Spelling checker checks words as you type them,
- Windows can be scrolled to see previous text,
- Supports full font, size, style and justification,
- Supports 300 to 9600 baud and
- Automatically puts your TNC in and out of KISS mode.

**Continued on next page →**

# MSYS To NEDA HexiPus Interface

*Copied from Quarterly feb 92*

This information is for those who wish to connect a computer directly from a serial port into a TJP Octopus or a NEDA Hexipus diode matrix. This computer could be running MSYS BBS/node software, for example.

There was an article in the Spring 1991 issue of the NEDA Quarterly that allows you to do this with a computer running G8BPQ network software into a matrix. That scheme does not work with MSYS. This information does work and has been in service at KA2JXI BBS for more than 3 weeks without a hitch.

This mod is performed on your computer serial card, and involves adding two 10,000 ohm, 1/4 or 1/8 watt resistors.

### *For use with a serial port using a DB-9 connector*

Locate pin 2 on the DB-9. Trace this back to a pin on it's 1489 line receiver integrated circuit (could be pin 1, 4, 10 or 12). Solder a 10K resiator from this pin to a grounud trace on the PC board (could use pin 7 of the 1489). Next, locate pin 8 on the DB-9 and trace it back to it's 1489 line receiver IC (could be pin 1, 4, 10 or 12). Solder another 10K resistor from this pin to a +12 volt source, such as pin 14 of a nearby 1488 line transmitter IC. This completes the mod to the serial board for a DB-9 connector.

### *For use with a serial port using a DB-25 connector*

Locate pin 3 on the DB-25. Trace this back to a pin on it's 1489 line receiver integrated circuit (could be pin 1, 4, 10 or 12). Solder a10K resistor from this pin to a grounded trace on the PC board (could use pin 7 of the 1489). Next, locate pin 5 on the DB-25 and trace it back to it's 1489 line receiver IC (could be pin 1, 4, 10 or 12). Solder another 10K resistor from this IC pin to a +12 volt source, such as pin 14 or a nearby 1488 line transmitter IC. This completes the mod to the serial board for a DB-25 connector.

*Note that these modifications do not affect the serial port for use in other applications, so the mod may be done and forgotten. They do not have to be removed.*

### *Connecting the modified computer port to the diode matrix:*

Serial port with DB-9 connector:

| Serial | TJP Octopus | Hexipus |
|---|---|---|
| Pin 1 | Connect to pin 4 & 6 | |
| Pin 2 | TXD | pin 3 |
| Pin 3 | RXD | pin2 |
| Pin 4 | See above | |
| Pin 5 | SG | pin 5 |
| Pin 6 | See above | |
| Pin 7 | DTR | pin 8 |
| Pin 8 | CTS | pin 7 |

Serial port with DB-25 connector:

| Serial | TJP Octopus | Hexipus |
|---|---|---|
| Pin 2 | RXD | pin 2 |
| Pin 3 | TXD | pin 3 |
| Pin 4 | CTS | pin 7 |
| Pin 5 | DTR | pin 8 |
| Pin 6 | Connect to pin 8 & 20 | |
| Pin 7 | SG | pin 5 |
| Pin 8 & 20 | See above | |

*NOTE: Some serial ports actually control the computer serial port DTR line in such a manner that it is switched to NOT TRUE state when the port sends data. If this is case with your's, and it interferes with the computer's ability to send data to the matrix, you may have to cut the PC board trace to the DB connector's DTR pin and connect a 1,000 ohm resistor from the connector end of the cut trace to a +12 volt source (such as a 1488 IC pin 14).*

**—Roger, KA2JXI @ KA2JXI.ny**
**—bbs sysop, Potsdam NY (OGDENB node)**

---

← **Continued from previous page**

A channel window has two panes. The top pane contains the incoming information from your TNC. The bottom pane contains what you type and send to your TNC. The size and location of the channel window can be changed easily as in any other Macintosh program. Both panes can be scrolled up and down to see items that scrolled of the screen.

A keyboard buffer window allows you to type long messages before they are transmitted. This window supports the cut, pasted, clear and undo functions (like any good text editor).

Users may use the control key or the option key (if they don't have the control key) to send control characters to the TNC. USers may optionally strip received line-feeds or all control characters before displaying and saving received dat to disk. CTRL-Gs can be passed to beep your computer if desired.

*Virtuoso* is shareware, that is, it is distributed freely, however, if you decide that you like the program and keep on using it, you must pay the shareware fee of $20 US. In order to check your spelling, *Virtuoso* nees a dictionary. This is available for $10. To register *Virtuoso* ($20) or to register *Virtuoso* and receive the dictionary ($30) write to:

*James E. Van Peursem, KE0PH*
*RR #2, Box 23*
*Orange City, IA 51041*

**—from May 1992 Gateway/ARRL**

# Gracilus PacketTen Network Node

*Copied from Quarterly aug 91*

Hi, I'm a nodeop of a Gracilus PacketTen based node here in the Chicago area. Yes, we are aware of NEDA. We've talked with various NEDA guys at the Dayton Packet Dinner, and know a little about how you do networks. We basically agree with the provisions you make to rid your network of what you call HTS stations, so the network can perform reasonably. Our attempts to do that here have resulted mostly in Chicago style politics by people from outside the Chicago area, rather than an acceptance of the work that has to be done to make things work and a commitment to get on with it. Congratulations on your organizational success.

As you may know, K9NG is a local here. The local group in its early days recognized the need for development, and supported these efforts. The result was a 9600 baud network, using NET/ROM, a few months after NET/ROM was shipping. I developed modifications that matched the K9NG modem to Midland 13-509s, and we installed a 9600 baud NET/ROM at the K9VXW-1 site and downtown Chicago. The downtown Chicago site uses, and continues to use, one of K9NG's original prototype modems interfaced to the Hamtronics 220 Rx and Tx boards, he used for the tests published in his paper along with the FM-5. It's been in continuous operation since 1987 at that site. Reports of G3RUH's early efforts to do 9600 baud, as reported in the British Packet Newsletter electronic edition, and circulated around the Midwest by N8XX, were actually transported thru the operational K9NG Modem based NET/ROM network in the Chicago area.

We came to the conclusions regarding dedicated link network structure, about the same time the *staggered link* idea was published in GATEWAY, by the Florida guys. Considering it further, and the practical limitations of Midwestern access to high performance RF sites, we did the system design for an idea which I've dubbed ***Cellnet***. It's relatively easy to do networking in mountainous areas, where there are good RF paths at people's summer cottages, with minimal tower and feedline. It's a little tougher in the Midwest where the 50 Kbuck tower only gets you 30 miles, reliably, and donated tower space is not always forth-coming for 3 or 4 antennas to be able to build networks like NEDA does. Cellnet solves these problems.

### *Cellnet*

With a single dual band antenna, or an up/down mount 430 and 2 meter antenna, 3 links and a LAN station can be operated. The links will be full duplex, dedicated point to point, which is about 6 times better throughput than the HTS protected, simplex CSMA technique used in the NEDA network, and for about the same cost. The PacketTen was developed in response to the Cellnet concept, but its application has grown beyond that. See my paper in the 7th ARRL Computer Networking Conference notes.

### *Recent History*

Anyway, based on that prehistory, here's what happened about the time we finished up the main 220/9600 baud stuff in the area: N4PCR moved to town. Additionally, KA9Q's code was first being tried and the fact it could handle so many ports, and simultaneous links made me think that it was time to start getting the Cellnet ideas I had down on paper. At Dayton that year, to my surprise, Karn and Dr. Death did a dog and pony show about a system very similar to the Cellnet ideas I'd had. So that really got me motivated to write it up and finish the system calculations. The excitement was contagious and N4PCR started working on a controller that would work with NET. His first attempt was good but just as he got it to work the 68302 chip became available. The 68302 is what the PacketTen controller is based on. He dropped the old project and began a 68302 project, formed a company, got two other guys to help him with it, and just started *really* working hard on it all.

### *Configuration*

The PacketTen system has 4 building blocks:

- 2 versions of the processor card;
- and 2 versions of the interface card.

### *Processor Card*

There is a PC plug-in processor and another version that can stand alone or be plugged into the PC plug-in version. Each processor card has the 68302 on it which has three DMA ports, a 68000 CPU and a RISC coprocessor for fast port operation, all in one chip. Additionally each PacketTen processor card has an SCC chip on it for two slower speed ports (up to 19.2Kbaud). The 68302 ports can do up to a megabaud. The maximum throughput per processor is around 2 Megabaud.

### *Interface Card*

The original interface board is a straight RS-232 port board. The next interface board they did is one that can handle the Kantronics versions of the standard 1200 baud and G3RUH modems right on the board. I think they might be coming up with a RS-232 interface board built as a plug in, with the same mechanical layout as the Kantronics modem, so they can do away with the original interface board altogether.

### *PC based 10 port switch*

As I said, above, the two versions of the processor board can be connected. This makes a 10 port switch. The two processors, in such a switch, and the resident PC communicate via triple ported memory, for full performance between the three.

### *Software*

The PacketTen comes with NOS software, ported to the 68000 in EPROM. A stand-alone PacketTen is the only NOS-in-a-box system available today. There is an

EEPROM for configuration memory. The KA9Q NOS has been expanded upon so a NET/ROM-like user interface is there with full locked routes capabilities. The commands are different, being built into NOS, but the capability is there. I should know, I've been on his case to put them there, HI. The latest version of the code now has sorted NET/ROM nodes list too.

### *Defined Neighbors*

By the way, we call locked routes *defined neighbors* in this part of the world so the NOS command is used for that is the NET/ROM neighbor command. It is actually much easier to understand and communicate to others once you are up to speed with the Gracilus/NOS terminology.

### *Memory*

The PacketTen has large memory for network node routing. It's much larger than the 32K of RAM that NET/ROM can use on a TNC2.

### *TCP/IP*

Since the PacketTen is running NOS, TCP/IP goes right through it. There's no need for NET/ROM-to-IP gateway stations if two IP LANs are connected together through links made up of PacketTen stations. IP is truly the future for high performance packet. It will eventually have the capability of automatic routing through hierarchal and determined routes. This combined system is powerful enough that world wide real time routing would be possible since beyond the 'determined' routing horizon packets would be switched by hierarchies kind of like switching real time traffic like BBSs.

Now, for *non* real-time traffic (zzzz)

**—Don, WB9MJN@W9IUP**

*Full info can be gotten on the PacketTen from:*
*Gracilus*
*623 Palace St.*
*Aurora, IL 60506*

---

# Use of GE Phoenix VHF Radios for Node User Port Transceivers

*Copied from Quarterly aug 91*

Recently I have received a couple of GE Phoenix VHF transceivers I considered for use at the CLV site. What a deal! A generous ham friend asked if they might be useful in our cluster. I saw an opportunity to reclaim a couple of expensive dual-banders and return them to normal use. These Kenwood stalwarts have given meritorious service for two years without a hitch. However, now was the time to replace them with rugged and durable radios.

The Phoenix is a 25 watt crystal controlled mobile unit whose small package lends itself well to stacking in tight quarters. It's tight front end and handy interface points make it a breeze to interface with TAPR clone TNCs.

A service manual was obtained and crystal data was determined. I ordered them from International Crystal Manufacturing Co. for about $38.00 a pair. Once inserted they tuned up on frequency without any trouble. The Phoenix seems to have a bottom frequency limit somewhere around 145.00 MHz. All coils were near the limits of their travel after tuning, however adjustments were positive with definite nulls and peaks where required. After about 1 hour tuning, the radio sprang to life with 23 watts output and a receiver sensitivity of about .35µV. Absolutely perfect for a rig used as user port with a low gain omni antenna.

I learned long ago speaker audio is not necessarily the best for use with TNCs of the TAPR variety (or any other for that matter). I usually tap audio at the hot side of the receiver's volume control. This provides an easily found tap point and results in great audio with minimum distortion and filtering. Lo and behold, after studying the service manual I found an excellent audio tap right there on the Phoenix's rear interface connector. It is labeled 'FLTRD VOL/SQ HI'. As it turns out, this is audio after processing by a low pass filter that removes any CTCSS tones on the received signal. No problem for the TNC, as packet tones are not within this range. The filtered audio makes the job of the PLL much easier as it doesn't have to discriminated low frequencies and their resulting harmonics. An added bonus is the volume control no longer has an effect on the desired packet tones and can be turned down so as to keep the site quiet and turned up when making antenna adjustments or to check the path.

The transmitter interfacing was straight forward. I simply connected the TNC transmit, audio and PTT lines to the radio's rear connector. Two audio inputs are available. I used the microphone input. I adjusted the TNC level and the radio's deviation control for 3KHz maximum deviation. This resulted in a sweet, easily decoded signal with no distortion.

The GE Phoenix appears to be an excellent radio for our purpose and could be for you too. They are easily found on the surplus market and require very little effort to retune for packet service. They are well built, durable and should provide many hours of dependable service.

**—Charlie, N2CJ@WA2YSM.ny**

# NOS PARMS Table

*Copied from Quarterly aug 91.* This file roughly indicates the NOS equivalents of various TheNET PARMS. Please pay careful attention to the footnotes, some of them are important.

| TN2.08 | | Description NOS |
|---|---|---|
| 1 | min quality for update | "netrom interface ax0 IPROCH 230" note 1] |
| 2 | HDLC channel quality | N/A (fits in with #1) |
| 3 | RS-232 channel quality | N/A (fits in with #1) |
| 4 | obso count initial value | fixed at "5" [note 2] |
| 5 | min obsolescence for broadcast | node is broadcast until it drops off |
| 6 | nodes broadcast interval | "netrom nodetimer" |
| 7 | FRACK | not sure |
| 8 | MAXframe (layer 2) | "ax25 MAXframe" |
| 9 | link-layer retries | "ax25 retries" |
| 10 | digipeat | "ax25 digipeat" [note 3] |
| 11 | validate callsigns | N/A |
| 12 | host mode connects | N/A |
| 13 | TxDELAY | "param ax0 1 15" [note 4] |
| 14 | broadcasts on/off | "netrom bcnodes ax0" turns on broadcasts for the interface named ax0 |
| 15 | *pound* node propagate | what does this mean? |
| 16 | connect command enabled | set through password file /ftpusers" |
| 17 | destination list max size | dynamically allocated |
| 18 | Time-to-Live initializer | "netrom ttl" |
| 19 | transport layer timeout | "netrom irtt" [note 5 - neato!] |
| 20 | transport layer retries | "netrom retry" |
| 21 | transport ack delay | "netrom acktime" |
| 22 | transport busy delay | "netrom choketime" (maybe? not sure) |
| 23 | netrom window size | "netrom window" |
| 24 | congestion control | (not sure) |
| 25 | non-activity timeout | n/a |
| 26 | P-persistence | "param ax0 2 128" [note 6] |
| 27 | slot time | "param ax0 3 xxx" xxx = slot ime |
| 28 | t2 | (got to look this one up, book t work) |
| 29 | t3 timer | "ax25 t3 xxx" (zero means never") |
| 30 | N/A | |
| 31 | N/A | |
| 32 | N/A | |
| 33 | duplex | "param ax0 5 xxx" |

One thing that NOS has that TheNet doesn't have:
netrom verbose [ON | OFF] if "off", on a broadcast the switch will only broadcast itself; it will not repeat its nodes list, only advertise its presence.

***Disclaimers***

(1) I didn't do it.
(2) This list isn't totally complete, so if you have questions **ASK**
(3) **Watch out for units**. For example, PARM 21 (ACKTIME) is in seconds, but in NOS "netrom acktime" is in milliseconds. This can be a serious problem if you've got retry timer units out of whack. Look it up.
(4) If you've got sysop privileges, you can remotely sysop these parameters, with the exception of those that can only occur once on initialization (like setting the incoming route quality)

***Notes:***

(1) This can only be done once, in autoexec.net. In the example, it sets interface ax0 as the node mnemonic "IPROCH" with route quality 230.
(2) In NOS, the obsolesence counters are decremented on a totally different timer than the nodes broadcast timer. Since this number is fixed at 5, if you want it to drop a node after 2 hours you would set "netrom obsotimer" to 2 hours divided by five, or 1440 secs)
(3) You may never digipeat "through" level three
(4) ax0 is the name of the NET/ROM interface. The example is 150ms.
(5) This is a really neat feature of NOS. The initial round-trip-time (netrom irtt) is the "default" time in which you expect an ack back. After a few packets, NOS starts remembering what the *real* time is. Thus, after it has been running for a few hours, it will learn to expect a response from a close node in a much shorter time than a far away node. A fudge factor is added to this round-trip time internally. Because of this feature, setting the transport retry limit to 1 is not practical; I have been using 3 and think that is fair.
(6) ax0 is the name of the interface we're programming. The '2' is the KISS parameter code for persistence.

**—Chris Pigot, WZ2B**

# Installation of the MFJ 2400 Baud Modem in the Tiny-2/Micropower-2

*Copied from Quarterly aug 91*

While performing this installation, normal anti-static procedures should be followed. Avoid carpets and plastic chairs (plastic chairs are really bad) as a start.

1 Remove the nylon standoffs from the modem so it will fit on the TNC.

2 The MFJ2400 has provisions for an on board negative voltage regulator. A 79C05 (-5V, 3 terminal voltage regulator in a TO-92 package) should be installed at VR1 and the trace between pins 2 and 3 of CN2 should be cut and a jumper should be installed between pins 1 and 2 of CN2. This now allows the MFJ2400 to be powered from -12V rather than -5V.

Steps 3 thru 14 will have to be performed to get the TNC and modem to fit into the case, however I recommend getting it to work first by skipping to step 15 and simply plugging the modem directly onto the TNC's modem disconnect header (MFJ2400-CN1 to TNC-J5). Pin one at CN1 on the modem should be aligned with pin one at J5 on the TNC.

3 Remove the Connector at CN1 on the MFJ2400. This is necessary since the shape of the PCB does not allow it to be plugged directly into the TNC and still fit in its case. Be careful not to break any traces. Clear out the holes so wires can be soldered in place later.

To remove this connector, it may be easiest to break its plastic housing into small pieces with a wire cutter. The plastic housing usually can be pulled away from the PCB leaving only its contacts soldered in the PCB. These contacts can then be pulled out one at a time while applying heat where they are soldered.

4 Remove the connector at J5 on the TNC for the same reason as item #1 above. Clear out the holes so wires can be soldered in place later. To remove this connector, it may be easiest to apply heat to the solder side of the PCB at one of the pins. After the solder softens, pull lightly on the pin from the other side, it should pull out from the PCB and plastic. Repeat for all the pins.

For steps 5 thru 14, #24 gauge solid conductor wire is recommended. It is best to solder the wires in place on the modem first, with .75 inch of insulation from the bottom of the modem. Long stripped portions (2 or 3 inches) of the wire past the insulation will make insertion into the TNC easier.

5 Wire pad at MFJ2400 CN1 pin 6 to pad at TNC J5 pin 6.
6 Wire pad at MFJ2400 CN1 pin 11 to pad at TNC J5 pin 11.
7 Wire pad at MFJ2400 CN1 pin 12 to pad at TNC J5 pin 12.
8 Wire pad at MFJ2400 CN1 pin 13 to pad at TNC J5 pin 13.
9 Wire pad at MFJ2400 CN1 pin 14 to pad at TNC J5 pin 14.
10 Wire pad at MFJ2400 CN1 pin 15 to pad at TNC J5 pin 15.
11 Wire pad at MFJ2400 CN1 pin 16 to pad at TNC J5 pin 16.
12 Wire pad at MFJ2400 CN1 pin 17 to pad at TNC J5 pin 17.
13 Wire pad at MFJ2400 CN1 pin 18 to pad at TNC J5 pin 18.
14 Wire pad at MFJ2400 CN1 pin 20 to pad at TNC J5 pin 20.
15 Cut the trace between pins 11 and 12 of J5 on the TNC.
16 Cut the trace between pins 13 and 14 of J5 on the TNC.
17 Cut the trace between pins 17 and 18 of J5 on the TNC.

*The following traces should* ***not*** *have been cut on J5 of the TNC:*
*Traces between pins: 1 and 2; 3 and 4; 5 and 6; 9 and 10; 19 and 20*

Using the supplied cable which plugs into CN7 on the MFJ2400 perform the following:

18 Cut one of the connectors off the supplied cable, strip and tin the ends of the unterminated leads.
19 Solder the Orange wire to the negative (-) side of C24 (CN7-3 Ground)
20 Solder the Yellow wire to the 5V output (pin 3) of U5 (CN7-4 +5)
21 Solder the Green wire to the side of R6 which is connected to U15. (CN7-5 - voltage)
22 Solder the Red wire to pin 4 on the radio connector.(CN7-2 RX Audio)
23 Remove C27 from the TNC and solder the Brown wire to the C27 pad which is connected to R12. (CN7-1 TX Audio)
24 Connect a wire from R37 where it connects to T5 on the MFJ2400 (the end closest to IC6) to Ground (Modem enable wire). Grounding this wire enables the 2400 baud modem.
25 The Radio baud rate setting on the Tiny/Micropower should be set to the 2400 baud position
26 Place some insulating material between the Modem and the TNC to prevent shorts. (note - anti-static foam is not an insulator. I have seen people use it as such which causes some real interesting problems)
27 Clip approx 1/8 inch off the ends on the pins at CN6 on the modem. They are a little long and will short to the case unless cut.
28 T3 should be bent over so its tab does not short to the case.
29 To hold the modem in place, a couple of pieces of buss wire can be wired from the ground plane on the component side of the modem along the edges (scrape away the solder mask) to the ground traces along the edges of the Tiny/Micropower.

This completes the installation.

***NOTES:***

To restore 1200 baud operation, first the Modem enable wire connected to R37 on the MFJ2400 should be disconnected from ground and be left to float (Item 24). Second C27 needs to be reinstalled (Item 23), but the Brown wire need not be disconnected since the 2400 modem goes to a high impedance output. Finally switch the radio baud rate setting back to 2400 baud. Proper wiring of a 3PDT switch to perform the above would allow switch selecting 1200 or 2400 baud operation.

For proper 2400 baud operation thru the radios microphone jack proper setting of the transmit deviation is necessary. 3.5 kHz is recommended. The transmit audio level can be adjusted by using the R12 on the TNC and if the output range is not sufficient or touchy the jumper can be moved on CN6 of the MFJ2400 or change the adjustment range (see MFJ2400 baud manual) The potentiometer on the MFJ2400 can also be used to adjust the transmit audio level as well.

The TNC which originally would function with input voltages from 9 to 14 volts will now only work with 11 to 14 volts input. This is because the negative voltage picked off of U15 on the TNC will not be enough for the -5V regulator on the modem. When this happens the modem stops functioning although the TNC will still function normally over the RS-232 port.

**—Bill Slack, NX2P**

# NEDA and Servers On 2 Meters

*Copied from NEDA Quarterly feb 91*

This article addresses the question of "What is NEDA's stand on servers using 2 meter user ports to access the NEDA network". A server is any station that is on the air as a service to users other than it's owner. This includes PBBSs, DxClusters, TCP/IP gateways, DOSgates, CD-ROM callbook servers and any station that sources large volumes of data to other stations across the network.

NEDA network participants voluntarily agree to a consistent set of technical guidelines. These guidelines only specify the software running in the node TNCs at each NEDA node site and the interlinking methodology between nodes. However, at least one of the club's technical documents describes how detrimental a station sourcing high volume data to a user port can be on users. Let me restate:

On a given node if the server has it's own uplink on 440 and the users access the server via the 2meter user port there will be very few collisions even when loading is at maximum.

If the server is accessed via the 2meter port (the server is on the same 2meter freq.) then there will inevitably be collisions. If the user port is fully loaded by users accessing the server or by other server activity then there will be lots of collisions and efficiency will be no more than 19% and sometimes as bad as 0%. [1]

Server ops who do tie into their local network on 2 meters will definitely degrade the performance of users at that 2 meter port. The users will also hamper communications of the server with the network. If the server were to find a dedicated access into the network the server and users would both benefit in efficiency, and more fun would be had by all. While servers may share a frequency with users, functionality will be far below that of servers on dedicated links to the network. If the server is offering a valid service there is no reason in the world those benefiting from same couldn't help fund the dedicated link for the server!

It is up to the node sponsors to determine their own policy to approve or disapprove of server activity on the frequency of the node's user access port. It is up to the potential server's operator to respect the wishes of the sponsor of the node. This policy is no different that the manner in which voice repeaters are operated.

The more fun that is had and the more efficiency with which the network is run, the more the network will grow and the more different kinds of services will be available. The bigger the network is, the more good people there are working together for a common goal. That is NEDA's position.

*[1]Binder R. Abramson, N, Kuo, F., Okinaka, A., Wax D. (1975), 'ALOHA packet broadcasting - a retrospect', AFIPS Conference Proceedings, 44, 1975 NCC, 203-215. ** reference and information source for the Quarterly was taken from "The Data Ring Main" by Flint. Published by Wiley.*

**—The NEDA Board of Directors**

# Notes about the 9600 G3RUH modems
## (including the NB96)

*Copied from Quarterly aug 91*

You may or may not have known that voltage variations on the 12V line affect the modulation level out of the NB96. This has caused some problems in some of the stuff I have done. A voltage change from 12 to 13.8V can easily change the transmit deviation (when fully interfaced to radio) from 3KHz to 5KHz. This is something to be aware of when setting up the 9600 BPS modems

I heard a report about a fix, which has made a big difference for some UoSAT ops, but with no cause. I looked into it, and I can see why. Here it goes:

On the NB96 card (internal or external) a reference voltage is derived off the 12V power supply by a resistor divider. This divider consists of two 100K with a .1uF capacitor from the junction of the resistors to ground. The resistors in question are RS1-3 and RS1-4. This provides a 6V reference. The capacitor and resistor values set a time constant of .005 seconds which filters out high frequency noise (above 200 Hz) which may be present on the 12V line. The 200 Hz significantly below the modulation rate so that is good, but it is too high to provide isolation from 60 or 120 cycle ripple from a poorly regulated power supply, and it provides no isolation from the low frequency voltage changes which occur when the transmitter keys up. I have noted that I have had to set TxDELAY much longer than I would have expected in some installations. This may be the root cause of that.

That resistor divider is buffered thru an op-amp follower to convert it to a low impedance suitable for use by the remainder of the circuitry. This is good because otherwise the problem would be much worse, but still any fluctuations in the reference on the high impedance side of the op-amp will be faithfully duplicated on the low impedance side since that reference is used by all the analog circuitry. Noise there degrades the overall system performance.

The fix to this problem, to replace RS1-3 with a 6.2V zener diode (anode goes to gnd). I would recommend replacing RS1-4 with a lower value resistor to increase the idling current of the zener, minimizing voltage swings. Something between

**Continued ➠**

# Tiny-2 TNCs at 38.4Kbaud

*Copied from Quarterly aug 91*

I did some tests upping the clock speed on the Tiny-2 TNC. I successfully got them to work with a 9.825 MHz clock which is 2x the normal clock. Since the baud rate clocks are also derived off of the main system clock, the 19.2 position now generates 38.4Kbaud. As higher speed links are more common, and the number of these links at one site increasing, the normal capacity of the 9600 or 19.2K RS-232 link between the TNCs is becoming increasingly taxed. 38.4 may be a viable solution.

One of the driving factors behind the test is the poor high speed performance of the standard TNC code. Doubling the clock speed should go a long way to helping. Of course all the timing parameters are now only one half their original value.

A TNC modified as such does not have the same computing power of a "data engine" or PC, however using it for high speed packet does become viable. Can you call 38.4 high speed? Some people seem to thing 56K is a magic number. Sort of like 9600 is magic compared to 4800 although 4800 is quite respectable. Of course some people would say anything less than a 1 megabit is low speed. In my mind, 300-2400 is standard (slow) speed packet. 4800-19.2 medium speed, and I would call 38.4 to 1 Mbaud high speed. Above that can be very high. All is relative.

Here is an idea. The G3RUH modems can operate above 9600 by changing some of the filter components. In fact I am told they would work much higher including 38.4K. So now we have a TNC with a 2x clock, and set the radio baud rate to the 19.2 position so it is actually operating at 38.4K. We modify the G3RUH modems filtering so it will do the 38.4K. So far so good, the proceeding is pretty straight forward. Next we take one of the Tekk KS-900 data radios. We change some of the front end filtering on the transmit side so it will pass higher frequencies than it was originally designed for. Now the only limiting factor is the 20KHz IF filter of the radio. We need more receive bandwidth. It is relatively easy on the transmit side, but we need to find a wider IF filter. Well I was thinking, what about FM broadcast receivers? They must use a wider IF and in fact, it should be about what we need. Perhaps a bit wide, but it should do it. What do you think? Basically a 38.4K link for the cost of a 9600 baud link plus a few extra parts and some work. Of course with the wider bandwidth range would be reduced. I am making some assumptions about the ability to modify the radio which is where I am not sure about. Any comments?

> *All the new Tiny-2s use 6MHz parts, but older units may use slower speed parts which means they could not be modified with the higher speed clock. In fact there is no guarantee that new units will work at the high speed. I only made the test on two units both of which worked fine. Also after upping the clock speed, tests should be made with the TNC at various temperatures to make sure it will function reliably.*

**—Bill Slack, NX2P**

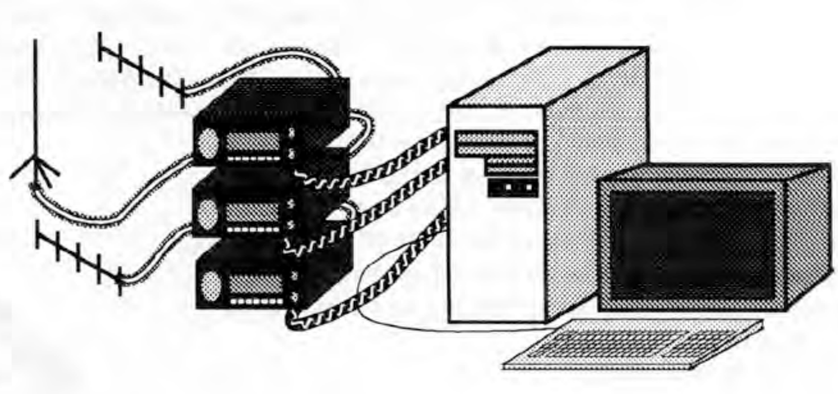
*[editor 1/4/1993: PacComm now sells a Tiny-2 Mark 2 for the same old price but it comes standard with 38.4Kbaud and 10Mhz selectable oscillator. It's also using low power CMOS gates now so it's only 50mA current @12V]*

**Continued from last column**

1K and 5.1K should do nicely. Also adding a larger capacitor in parallel with C24 would be a good idea. A 5.0µF capacitor would maintain the same time constant if a 1K resistor were used instead of a 100K. 25µF would provide some immunity to 60 and 120 cycle stuff, but 25µF is starting to get big.. The Zener works well on the lower frequency stuff so the time constant between the resistors and capacitor is much less important so it is not worth getting too hung up on the capacitor. C24 should be left in the circuit as a .1uF. That provides isolation from the high frequency stuff which the Zener is not good on. RS1-3 and RS1-4 are parts of a resistor SIP network so it is not simple to just remove these without affecting the other resistors. Since the impedance of the reference circuit is being dropped so low with the above changes, these resistors can be left in place with little effect.

Note: the above note/modification not only applies to the PacComm units, but probably all the G3RUH based modems since the above problem is in his original design.

**—Bill Slack, NX2P**



# The Exposed Receiver Syndrome

*Copied from Quarterly aug 91*

By now you have probably heard of the Hidden Transmitter Syndrome (HTS), and NEDA's policy of insisting all backbone links be HTS free. The hidden transmitter problem is the scourge of low cost backbone attempts. A related but less well known problem, which can have a detrimental effect on hilltop User Ports, is the Exposed Receiver Syndrome (ERS). The Exposed Receiver is the receiver at a *user* port that hears much more than it needs to; it is "exposed" to signals working other nodes in distant communities.

The problem with nodes experiencing ERS is, *user* port TNC defers to the distant signals, waiting unnecessarily before sending. The TNC does not realize these other signals it hears are so distant it could safely transmit to its local users without causing interference to the distant stations. In the meantime, the local users' TNCs may retransmit because of the unexpected long wait for an acknowledgment. This further adds to congestion on the channel and slows things down.

What does this mean for you, the operator (or future operator) of a NEDA node? First, this shows the need for small, local coverage nodes ("nodes in homes" as Tadd calls it). As activity grows, the packet network will benefit from a "cellular" approach, where several local coverage nodes are linked together via a HTS backbone, rather than trying to cover the same area with one massive coverage mountaintop node. Second, the *user* port frequency should be chosen to reduce ERS from other nodes on the frequency. This means wide coverage area nodes should be on less popular frequencies, as opposed to say 145.01. High power output and high gain antennas may actually hamper the usefulness of your node rather than enhance it. If the node is not centered on the intended coverage area, then use a directional antenna to target your audience and limit your exposure.

Phil Karn, KA9Q, proposed a software solution to this problem in a paper at the last ARRL Computer Networking Conference. This may be a long term solution, but it would not be compatible with existing systems now.

Consider adding a CTCSS encoder/decoder to the node. If 2 nodes sharing the same frequency each had CTCSS encoders, they could each sense the presents of the subaudible tone from the other node and then squelch the receiver. (The sense of the squelching is the reverse of what is usually done with CTCSS). When a local user started transmitting he would capture the receiver at the local node, thus ending the sub-audible tone and allowing the receiver to open. Since each nodes' receiver would stay squelched whenever the other node transmitted, the nodes would never end up waiting for each other before starting to send.

**—Rich Place WB2JLR**

---

# Split UHF Frequencies for Maximum Band Utilization

*Copied from Quarterly nov 91*

At the VE2RM:WQC node site I have installed a packet repeater. The repeater site is operated by the VE2RM radio club. Because of the split frequency (5MHz) nature of the repeater I was limited in installation of UHF links. I proposed instead of running point to point backbone links on UHF simplex frequencies, which are scarce at the site because of the repeater, that we run our point to point links on half-duplex channels whose pairs are adjacent to one another. Thus as the repeater transmits on 441.025 and receives on 446.025, the links would all transmit in the region of 441 and receive in the region of 446. This implies the sites we're linking at must also be using a split, half duplex, method of UHF channalization. This plan allows for many UHF links in and out of the same site.

The idea has been implemented now in both the Montreal and Quebec metro regions and seems to be without flaw.

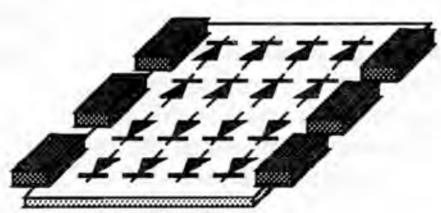
**—Burt Lang, VE2BMQ**

# Bug! MFJ's in Node Operation

*Copied from Quarterly aug 91*

MFJ TNCs wired incorrectly may be causing massive slow-ups in multi-port nodes. A miss-documented pin on the 25 pin connector could be causing collisions on the matrix of nodes with more than two TNCs. The Octopus, Hexipus documents (and MFJ manual) describe pin 20 as the RTS line when actually pin 4 is the RTS line. This causes some TNCs to not know another TNC is already talking on the Matrix. The fix is to simply jumper pin 4 and 20 on all MFJ TNCs' DB25. Also do the Wink-N-Blink mod described in the Annual.

**—Tadd Torborg, KA2DEW**



# Screamers!

**or**

# The Network ... What is it?

*Copied from Quarterly feb 91*

How many times have you tried to explain to a newcomer what the Network is all about and how NEDA's version is superior to a string of digi's or a single frequency arrangement of nodes only to have that person walk away from you after 5 minutes or go glassy eyed and mumbling to themselves. We all have at one time or another and it hurts to realize that many packeteers have little or no idea of what goes on in the network. If you think about it for a moment it isn't that much different than people's approach to the phone system in that everyone uses it and can detect the slightest decrease in performance but not many understand how it works. Next time you try to explain the system's workings to someone think about how NEDA evolved and your explanation will make far more sense.

Going back to the earliest recorded history of packet we find reference to a Neanderthal race of pseudo techs that developed a method of communications which consisted of placing individuals, one after another, on a string of mountaintops. Each was within earshot of the next and each individual was equipped with a magaphonelike gourd used to enhance his directivity and to increase his unidirectional gain. Each person was capable of repeating what he heard but only the first and last had a check back memory, intelligence was a rare commodity in those days. Messages would move from one end of the mountain chain to the other through a series of shouts and grunts one to the next ending hopefully with a final shout to the end mountain. What with screaming beasts in the jungle below, erupting volcanos and violent electrical storms the message was often lost somewhere between start and finish. This would result in a series of screamed "What did You say?" — "Beats me, let me check with the next guy "exchanges that would eventually return to the beginning Mountain top and then the process would repeat itself. If one was patient eventually all pieces of the message would arrive at the end mountain and communications was deemed to have taken place. The age of the DIGIPEATER had come and communications limped along for several years in this fashion.

For some this approach seemed to be a little clumsy and finally the question was asked "What if each person on each mountain top had a check back memory?" The mountain tops quivered as this flash of brilliance reverberated through the system. "With individual memory we wouldn't have to restart missed message segments from the first mountain but rather we could correct mistakes or repeat portions missed between any two mountains at the point where the problem occurred." It didn't take long for people to realize that this was a vast improvement over the aging DIGIPEATER approach. One local leader of an especially primitive tribe of people in what is now NY State was heard to say "I Node There was a better way to get this dun" and thus the Age of the Node was born.

The new system grew in leaps and bounds and soon there were nodes all over the place, all screaming and grunting from there respective mountain tops at the top of their lungs and remembering things between screams and passing traffic hither and yon. As time went on, more and more nodes appeared and people started to notice messages were taking longer and longer to get from one end of the system to the other. Once again people started to wonder if there was a better way to get things done. A young chief with long flowing golden tresses journeyed to the top of one of the mountains and after listening for a few moments to the cacophony of grunts, screams, hiccups and burps exclaimed "This noise is awful, I can't tell one message from the next! We NEDA better way of doing things". Yup, you guessed it. This was the start of the NEDA Network and it represented another great leap forward for communications.

The young chief reasoned that the problem with the Old Node system was that with so many Nodes on so many mountain tops all screaming at the same time individual messages were being lost in the roar. He returned to his village and in a stroke of genius discovered that certain member of his tribe could only hear low tones while others could only hear middle tones and some could only hear high pitched tones. He then selected other members of his tribe with low, medium and high pitched screams and he started to build his new network. He placed pairs of screamers on mountain tops with different pitched voices and along side of them placed pairs of tribe members with different hearing responses. He found by using this approach they could be receiving a message from a distant mountain and at the same time be sending one to the next mountaintop without the two processes interfering with one another. Once again the communication rate bounded upward.

Well many years have since passed and the young chief is now old but his work goes on. Through careful selection some mountain tops have 6 or more pairs of screamers and listeners. It is rumored some mountain tops are populated with *faster* screamers and messages are moving more rapidly than ever....

I hope this historical perspective helps you the next time someone says "Tell me about NEDA".

*73's*

*The Old Packeteer*

# Kantronics D4-10 UHF Radio Digital Repeater

*Copied from Quarterly feb 92*

*[Here's some followup information on the digital repeater project reported on in the last article: Using the Kantronics D4-10. -Ed]*

We now have a 19.2KB full duplex digital repeater on the air here in Dayton. It's built from two Kantronics D4-10 DataRadios wired back-to-back with minimal glue and control circuitry.

The frequencies are 420.950 input and 430.950 output. The antenna is an 11.5db gain Diamond Omni (specially cut for 430MHz; they <are> available if you whine enough). We're using a 4-can TX/RX duplexer.

The design is really simple... essentially, RXD from the receive radio is passed to TXD of the transmitter and DCD is passed to PTT. The minimal glue is a single 4049 hex inverter. Of course, interfacing to a TNC to tie the repeater to a node, and the control stuff to keep it legal, adds some complexity, but not much.

***A few things we've learned:***

1) The UHF band is not an ideal home for even moderately wide band packet. Our first channel choice — 420.75/430.75 — was scotched because it turned out the local ATV repeater had its audio output on 430.75. Then, it turned out that a local repeater had audio links every 100KHz from 420.075 to 420.975.

These problems were solved by moving to 420.95in/430.95 out, and working a deal with the repeater group to free up the 420.5-430.0 range.

Now, we're still hearing some sort of link (or possibly a spur) on 430.95 occasionally. We're pretty sure we'll be able to either move it or live with it, though.

But ATV is the real bugger in this range. Assuming the common 426.75 video carrier frequency, operation from 430.05-430.55 is likely to cause some interference to the ATVers, though not much to our operations. Above 430.55 is probably OK, but audio on 430.75 may screw up that channel (as it does for us).

Since the 425-431 ATV channel is pretty universally accepted, and the 430-431 range is also allocated for wide-band packet in many areas (at least in theory), the potential for trouble is great. Repeaters and links aren't allowed from 431 to 433, there's weak-signal and satellite stuff above that, and there's likely to be another ATV channel (in our area, the repeater input) from 437-442.

Anyway, be prepared to do some dancing if you plan to use a bunch of 100KHz channels on UHF...

2) The repeater runs with no squelch tail. In our case, this was necessary because we're using squelch for DCD (refer to my earlier postings for endless ramblings on this subject). It works pretty well, though; even through the repeater we're using TXD of 20ms with good results.

3) Since we're not regenerating clock at the repeater, the system will actually pass any data speed up to 19.2 (or beyond?) that deviates +/- 10KHz from center. That makes things interesting for experimenters...

4) The wider (60KHz) bandwidth of the 19.2 signal has its effect; a path that's solid on 2M packet with omni antennas doesn't cut it on 430 even with a 22 element beam at one end and an 11.5db omni at the other. I'd be a lot more comfortable with radios that ran 25-30 watts than 10-12, but one takes what one can get...

(Note, though, that on a good path, 30-40 miles is no sweat with these radios.)

Anyway, we're having fun (I think...)

**John Ackermann, AG9V**
**Miami Valley FM Association, Dayton, Ohio.**

# Hardline Connector Help

*Copied from Quarterly feb 92*

While shopping around last fall at the ELMIRA hamfest, I met Walter Obenhofer, NQ2O who was selling various types of hardline connectors. I believe that Walter machines these items himself, and a beautiful job he does! The connectors he machines in his hobby time look and work just as well as commercially produced connectors costing up to two or three times as much.

I decided to give it a shot and purchased several of his "economy" cable connectors and installed them. The connectors, which came complete with O rings to keep moisture out, worked great. Walter also provides excellent service on telephone orders. I had to order some additional connectors to finish a project on my new tower, and the special 7/8" N fittings were delivered very promptly. Even when I had to "special order" a fitting for solid core cable, Walter was able to provide this part and he even sent extra center pins for my convenience.

Hats off to NQ2O for his fine service to the amateur community. Should anyone need any type of hardline connector, give him a call. Walter either stocks or can create connectors for just about any type of feedline you might have at very reasonable prices! His address and phone number are:

*Walter Obenhofer NQ2O*
*159 Lighthouse Road*
*Hilton, NY 14468*
*Phone 716 392-4231 after 6:00 pm*

**—Bob WB2QBQ**

# Interfacing PacComm NB-9600/G3RUH modem to Kantronics D4-10

*Copied from NEDA Quarterly June 92*

Due to numerous requests, I am posting modification instructions for the PacComm NB-9600 G3RUH modem to convert it to TTL input/output to drive a Kantronics D4-10 at 19.2Kbaud. Don't forget to change the radio speed jumpers in whatever TNC you are using to 19.2K after these mods are done.

I mention this only because I spent one hour trying to figure out why my mod wouldn't work, only to finally realize I never changed the TNC *speed* to 19.2K to drive the blasted modem! (what an idiot huh?) It's the simple things that getcha! :-)

5 Feb. 92

All part numbers given are for the PacComm NB-96 modem.

***Receive audio in to receive data in (conversion to TTL input) mod.***

Locate U10 pin 2, and circuit board trace going to U5 pin 2.

1 Disconnect output of U10 pin 2 by either lifting the lead leg on the integrated circuit or by cutting trace. Lifting the leg on the IC is the easiest way to go about it.

2 Connect TTL receive data input from the D4-10 to trace going to U5 pin 2.

Note: On Kantronics 19K2/9K6 modem, they actually put in a jumper that lets you either drive U5 pin 2 directly from a TTL source (such as the D4-10) or when the jumper is installed, the input to this stage reverts to normal G3RUH operation with U16C acting as the Rx filter for the analog input.

***Transmit audio out to Transmit Data out (TTL output) modification.***

Locate U18 (74HC164) pin 3 and trace going to Jumper-1 pin 1. Please note that Jumper-1 normally is jumpered from pins 2 to 3 with a header jumper.

1. Connect wire from Jumper-1 pin 1 to D4-10 transmit data input.

Note: On Kantronics modem, they do the same thing. They disconnect the output of U18A (after C34) with a header, and allow you to drive the output with U18 pin 3 or the analog output from U18A determined by the position of a header jumper. If it is desired to disable the audio output of the G3RUH modem, you can cut the trace after C34 or remove C34. I did neither and just let it run since I was no longer using the audio output, and I saw no harm in just leaving it run.

***Lock Detector modification.***

Locate resistor pack RS-2. It is a 100K resistor pack and its only purpose is to give 100K of resistance between pins 1 & 2 and 3 & 4. The other resistors in the pack are unused. The goal here is to change this resistance to approx. 50K, and there are a couple of ways to achieve this as listed below:

1 Cut traces to RS-2 and install two 50K resistors. One between pins 1 & 2, and another between pins 3 & 4.

2 Leave everything intact and place a 100K resistor across pins 1 & 2, and another across pins 3 & 4, this will form two parallel resistor networks with each one offering 50K to the circuit. I used this method. Not much room to work on the board for this step and soldering is in very tight quarters.

3 Use the unused portions of RS_2 to do the same thing as #2 above using shorting wires.

Note: The value of 50K for the lock detector was chosen from notes given in the PacComm NB-96 manual. I have been experimenting with these values and have achieved what appears to be better results with values different from those reported. However since I can not document *why* that is happening, and since it might be limited to my application, I will not go into details on what values I am using. I merely point this out so that if you happen to notice what appears to be poor DCD detection on what appear to be good signals, this might be a good first place to look.

Note: There were some suggestions given in the past to modify a Kantronics DVR2-2 to work with the NB-96 modem at 9600 baud. These suggestions consisted of bypassing one resistor and changing the value of another *in* the radio. The net result was to increase the audio output level of the radio to the modem, and to lower the transmit audio input level from the modem to the radio. I have found that increasing the audio output of the radio to the modem often-times results in *worse* performance and would recommend keeping it stock.

However, lowering the transmit drive level is required but it seems to make more sense to do this in the modem rather than in the radio. If a 47K resistor is placed in series with transmit audio output (*internal* to the modem) then it will duplicate Kantronics design, and will offer easy adjustment of transmit deviation whether you are using a DVR2-2 or a D4-10. Of course this is only necessary at 9600 baud, as when the above modifications are done, the output of the modem is TTL and not analog.

**Mark Bitterlich wa3jpy @ wb4uou .nc**
**mgb@tecnet1.jcte.jcs.mil (internet)**

# Modifying the Motorola Mocom 35 For 9600 Baud Packet

*copied from a Great Lakes International Digital Association technical bulletin as published in Quarterly feb 92*

## NOTE.. Start with a operational tuned up radio on the frequency you are going to be using it on.

1 Remove DC power cord, Antenna cable, and take the radio out of the case.

2 From the component side of the radio, remove jumper from pin #40 and #41 located with the front of the radio facing you just to the left of the center partition, about an inch and a half from the back. NOTE.. keep this jumper.

3 Remove the ends from the jumper in previous step. Obtain a 1.1 uH RFC and place insulating covers on leads, (note do not cut the leads on the rfc use full length), solder the jumper ends to the ends of the RFC.

4 Install this RFC on Pins #40 and #3, note #3 is located to the right of T101 and the left of Variable resistor. on the left side of the radio.

5 Remove capacitor C119 to the right of T102.

6 Remove capacitor C117 to the right of T101.

7 Turn radio over front facing you solder side up.

8 On the right hand side locate T101 and T102 near the rear of the chassis.

9 Cut trace to the right of T101 alignment hole, between the first and second component holes on trace going from front to back of set, (this trace has three component holes) see figure 1 .

Add a 4 inch length of hookup wire here on top side of board to xtal

Add a 47pf mica from point A to ground. Add a 36 pf mica from point B to ground.

10 Solder a 47 pf Mica capacitor from hole A of foil you cut in step 9, to ground. See figure 1.

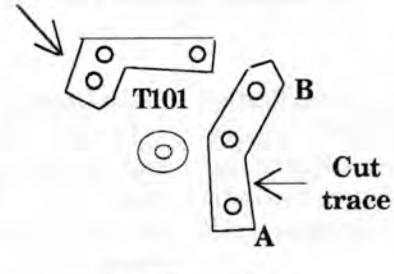



Figure # 1

11 Solder a 36 pf Mica capacitor from hole B of foil you cut in step 9, to ground. See figure 1.

12 See Figure 2 and cut trace to front right of T102 between point A and middle hole of trace.

T102
alignment hole -> ( o )

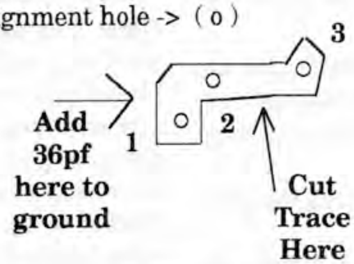



Figure 2

13 Connect power cord, antenna and turn on and retune T101 and T102 before continuing with modification, once tuned remove power cord and antenna.

14 Locate the Transmit Crystal channel element and remove from board.

Open up the element and remove the fixed capacitor approximately 25 to 32 pf in parallel with the variable capacitor. Solder a 4 inch length of small hookup wire to the junction of the variable capacitor, crystal and thermistors, on the foil side of this pc board. Re-install the crystal circuit in the plastic element case and file a notch on the side of the plastic case for the jumper wire to exit between the plastic cover and the main chassis when reinstalled.

15 Reinstall the channel element on the main chassis. Solder the jumper wire from the crystal element to the rear hole where C117 was removed next to T101, (see figure 1). This is the is the junction of CR102 and R112.

16 Locate the mic jack on the left side of the chassis component side front facing you. There is a 10 pin terminal strip running along the left side rail, number these terminals from front to back (1 to 10).

17 Disconnect the wire running from the mic connector to terminal strip pin 5. Solder this wire to blank terminal pin 7 on this strip.

18 Using a length of RG174 or teflon equiv. coax, prepare one end, and solder the center conductor to pin 7 of terminal strip on left side. Solder the shield to terminal 9 of same strip.

19 Feed this coax between the face plate and the pc board near the speaker from the top to the bottom of the chassis.

20 Turn the radio over with the face toward you and the solder side up.

Locate the metal shield in the shape of a trapezoid, near the front right side of the pc board (this shield has a 3/8 inch hole in it for alignment of the Discriminator).

Along the front left angle side of this shield is a jumper wire connecting the shield to the pc board. Just to the rear along this side the very next foil trace is the discriminator output. There is possibly a printed triangle with the number 15 pointing at this foil pattern. This is the junction of CR2, R35, C48, R37, R38.

Using the *free* end of the RG174 you just installed in step 18, measure and cut to length and prepare the end so that it can be soldered to the above described discriminator trace.

Solder the center conductor to the trace described above and solder the shield of the cable to the metal shield.

21 Connect the DC power cable and antenna to the radio, align the transmit crystal to frequency.

22 Pin pinout of terminal strip on top left hand side is:

***front***

**pin 1** PTT

**pin 2** Transmit audio

**pin 3** ground

**pin 4** ground

**pin 5** wire from speaker

**pin 6** unknown

**pin 7** new discriminator output pin to mic connector

**pin 8** jumpered to pin 10

**pin 9** ground

**pin 10** +12 volt DC input to radio

23 Microphone connector 6 pin Din plug

**pin 1** Rx discriminator audio output to TNC

**pin 2** ground

**pin 3** ground

**pin 4** 9600 baud TNC audio input to radio

**pin 5** PTT line

**TNC,** set the tx audio level in the 9600 baud modem as per instructions from PacComm, and you're away to Happy Packeting.

I hope this bulletin is of some help to other's working on system development, more in this series will follow as modifications are made. Next will be a writeup on the Mocom 70 series conversion which is very simple because it is true FM to begin with, Also the Motorola Moxy, and the Motorola Maxar will be modified, these are also true FM.

**—Ron, VE3MX**

**—sysop of Port Colborne Ontario node**

**—asst sysop VE3SNP BBS**

# Students Participate in Amateur Packet

*Copied from MAPRA Digitell, as printed in Quarterly feb 92*

NEW HAMPSHIRE EDUCATION
BULLETIN # 4 DE WB1GXM
ASSISTANT SECTION MANAGER
NEW HAMPSHIRE-EDUCATION
G.E.A.R.S. WEATHER PROJECT

Increasingly, more school groups, licensed educators, and students, are using Packet Radio. School traffic between schools across town and across the world can be seen daily moving from node to node or via Aplink. Much of it is packet pen pal traffic, school contacts, and teacher to teacher contacts.

The nature of this traffic is starting to change. KB0CUS, Chuck Bryant, Marlbourogh Elementary School, Kansas City, Missouri, uses this medium for the collection of material for his student newspaper, Bacon Bits. Recently, he was the national collection point and Packet/ landline interface for Geo Quiz, a geography game involving over 29 schools across the nation.

Closer to home, the Goshen-Lempster Educational Amateur Radio Society, or G.E.A.R.S. has been using packet as a way of collecting weather data from individual stations and schools for a science unit on weather. It will be used for graphing and comparing both in science and math.

Both licensed and unlicensed kid participated in intergrating Amateu Radio into the science project, show ing what Amateur Radio can do, an motivating other students in becom ing licensed.

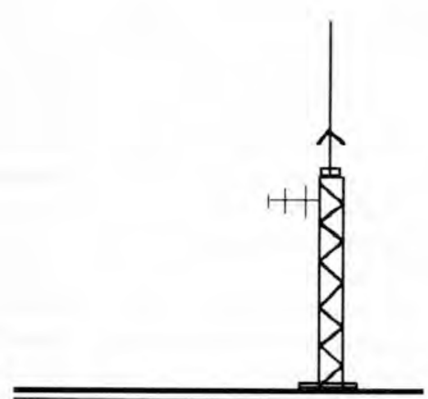
The project was initiated with a announcment at a meeting of the M Ascutney Packet Radio Associatio (MAPRA) and by sending a packe bulletin addressed to NEBBS.

The GEARS is using an Advance Electronic Applications PK-88 termi nal node controller, with AEA's lat est firmware, at the school i Lempster, NH, WB1GXM-4 Mbx and in Claremont,NH, WB1GXM- Mbx. By using two tncs, stations i and out of the area were able to aut forward via WA1YTW.NH.USA.N to WB1GXM-1.

The kids, with an assist from me download the data at classtime. Thi was their first exposure t telecomputing and Amateur Radic Several are now in GEARS's fift Novice class.

Looking into the future, the con tinued technological advancement c Packet can assist other schoo telecomputing projects. If you hav an idea for using packet, please con tact me at the address above. Th kids and I would like to hear fron you.

**—Conrad WB1GXM**



# ROSE X.25 Packet Switch Development

*Copied from NEDA Quarterly June 92 updated 2/15/94*

*The following is a development history of ROSE that I had asked Andy to compile for Quarterly readers some time ago. Andy graciously complied and so now we have a better view of this alternate packet networking method. It is my hope that by exploring the similarities and operating environments here in print, that some will be encouraged to experiment with both ROSE and ROSE-TheNET interoperability. As Experimenters hams should leave no stone unturned!*

*-NEDA editor*

Contrary to pronouncements to the contrary, Tom Moulton, W2VY, continues to actively develop the ROSE X.25 Packet Switch. Users of the RATS ROSE Network can attest to this — Tom uses our network to test new code!

ROSE Switch version history, and future plans, is included as a "read.me" file in the ROSE distribution ZIP file. That file, taken from ROSE X.25 Packet Switch Version 2.9 (release date 920501) follows.

***28 June 1990***

Switch was not setting a couple of Reserved bits in the numeric digi field, this was causing one vendor's code to crash. Had a bug where the T3 (Check) timer was not being started if the user was running AX25L2V2 OFF (if Ver 1).

Note that L2T2 and L3T2 should be set to 4 if users are using 256 byte frames. If they are using 128 byte frames then the defaults should be OK.

The switch will now allow calls on LCN's that are out of range. What this means is now when connecting to an unconfigured switch the MaxVC 5 is no longer needed. It will not allow more than 5 connections until it is configured and the link is brought down. (Most easily done with the power switch or by using BOOTER.LOD)

******* *Major Bug Fixed* ******

There was a bug from since before 060289 where when Switch A was connecting to Switch B and Switch B did not know about Switch A when the Level 3 was Restarted it might crash based on what was in high RAM. This would also vary from machine model (if TNC2 vs PK88 or DR-200, etc.) which explains some other oddities that have been seen.

***Independence Day 1990***

Fixed a bug that would add garbage entries in the users display if a user tried to connect through two (or more) digi's using the switch as the first. (Did not specify a network address) I don't think this would really cause any problems but it might have, since the USER Linked to had a pointer that was pointing off into memory somewhere - but i don't think it would ever modify anything out there...

***13 July 1990***

Have invalid connect requests (except to self) be ignored instead of returning a DM (if *** BUSY). Allow for L in X.121 Address to be translated to the digit 1. (Should have been this all along, I did have a typing class once!)

Repaired the PK87/88 overlay. Renamed it to be PK88.OVR.

*All references to PK87 should be changed to PK88!*

***10/26/90***

The 900713 version has been performing very well. This version only has three changes.

1) Password protection to LOADER

2) The RS232 port now supports the diode matrix used for Net/ROM as well as standard RS-232.

3) Clean up for CONFIG and LOADER and fixed a bug in USERS. Config and Loader would not properly handle the case of someone attempting to connect to them while someone else was using them. And Users would not allow more than one connection at a time (fixed).

Added two new applications: MHEARD.LOD gives a shorter heard list when it is loaded, users still connect to HEARD.

INFOSP.LOD has Spanish messages on the disconnects.

*If using an RS-232 LAN you need to cut the jumper between 10 and 20!*

**** SEE MANUAL.UPD for changes to the manual!!!!

***11/11/90***

The EPROM defaults to having No Password, can be changed via CONFIG. Added some space after the COPYRIGHT NOTICE for local LAN info. 60 Bytes (My copyright message is "ROSE X.25 Packet Switch Version xxxxxx by Thomas A. Moulton, W2VY")

To make the text get sent to users you will need to modify two pointers, these are l2wdta and l2eob, see ROSEZSW.LST. Note that l2eob points to the Last char of the string and l2wdta points to the next location. (if l2wdta = l2eob+1, Always!)

The password can be burned in EPROM as well, location defpwd. The first byte to the number of characters the reply needs to be, the second byte is the length of the key (40 max) and the next 40 bytes are the key.

Both of these will be modifiable via CONFIGUR.EXE in the near future, and I will send a message to ALL@ROSE when it is ready for download.

***8/10/91***

One major change - All Messages out of the switch that BBS's or other programs might look at now start with "Call." Here is the list:

Call Being setup Call Completed to

*** Call Clearing ***

*** Call Reset ***

I did some space conservation. Also wrote an application (aka .LOD) that will print out the contents of the LINK (Level 2) data structures. This is really only useful to help debug the NO PEER problems. Do the following: Upload UDUMP.LOD (You may have to BOOT the switch first, it is BIG!) then when you have some NO PEER's connect to UDUMP and hit return (With capture file open) - It will print a lot of information. Send me the file either by mail, packet or MODEM (The latter is preferred)

***30 Aug 1991***

Some space conservation, changed the memory usage limits to send RNR's a little sooner (So there is more memory to work in) to help the RR/RNR loops. Also made a change upon when the timers are reset, to help avoid the RR/RNR loops. This is an attempt to avoid the problems.

Also made a change to application connect logic to allow a single application to accept connects for many destinations. This is to be able to start working on CONFERENCE Bridge and Broadcast Applications.

***10 Sep 91***

Added some debug code to detect where the NO PEER vc's were cleared from within the code.

***25 Nov 91***

Added better flow control for VC's and L2 Links. We hope that this will reduce/eliminate the NO PEER problem. Will also reduce the amount of data taken by a single VC.

***30 Mar 92***

Moved, Son was born, bought new computer,... Updated Applications to not respond to CR and to send all data when you first connect to them (Like INFO). To keep various people happy am calling this version Version #2.8, many things will still be based upon the date.

The file names will be changing to have the version number instead of date!

The idea is to be able to say that we will have PID (aka TCP/IP) support in version 3.0

Also now support updating TXDelay from CONFIG!

:020E000000 Read the value and :010E0100ttcc TT is TXDelay value (HEX 10ths of milliseconds), CC is checksum 28D8 is default if you increase TT by 1 decrease CC by 1 (28h = 40 is 400 MS)

Also removed MHEARD.LOD and modified SHEARD.LOD to only allow one Heard List per connection. (Single-HEARD).

***1 May 1992 Version 2.9***

Added new parameter to clear packets, the switch can now indicate where a clear originated. This means if you are told 0D00 you will also know Which Switch has the problem! The required many updated to the handling of call setup and clearing, so the information could be relayed back. It will only be displayed to the user when INFO.LOD is loaded, and will not be displayed when in Reliable mode. Also pass Q, D and M bits correctly. And found a bug is the passing of Call User Data.

***11 July 1992 Version 3.0 Beta***

Full Support for Protocol ID's (PIDs) Other than F0 (User Data) - This means that TCP/IP and any other Protocol will be transported by ROSE Switches. Alpha testing will be in June and it should be released in July 1992.

Also added support for a LID list, and a way to redirect a call@address to any other call@address. These items have not been tested completely and are not yet supported in CONFIGUR.EXE

***30 July 1992 Version 3.0***

Minor bug fix to clearing cause/ address. TCP/IP support is tested and works! MAKEPROM.EXE now supports BANNER text - a single line of up to 60 char that follows the "ROSE X.25 Packet Switch..., W2VY". Also now supports single depth INCLUDE files. Command Prompt (INCLUDE file) or command line /INCLUDE=file. I have also included sample files for various radios, speeds for user and point to point links.

The .CNF files have changed the word NODE to SWITCH in all instances. (node is still supported, but will be phased out.)

***11 September 1992 Version 3.1***

Had a major bug with eating memory in 3.0 took a while to fix, the answer was that the switch would drop about 60-70 bytes each time there was a "routing loop" detected, which in a well configured site will happen often if one of the main links is Out of Order.

***3 March 1993 Version 3.2***

The LIDLIST now checks for a ssid's (Do not specify one when con figuring). The Switch callsign use for outbound connect requests wi be the Digi Call when the connec tion is in reliable mode. (For IP Us ers) There was a case where the ad dress on a Call Clearing would b incorrect if the call had been com pleted, it is now fixed. I have als added structures to include the Ad dresses of neighboring switches s that later USERS can be updated t display this information, usefull fo Switch Hopping.

***12 April 1993 Version 3.3***

Completed updates to includ neighbor switch address in USER display, also reformatted USER display to present informatio clearer. Fixed a bug in the 4 digi addresses are handled, they used t not work since they looked like DNIC. Now the DNIC is the defaul DNIC (same as entry switch) whe there is only 1 digi field of number and if there are two digi fields c numbers then the DNIC must b provided. This is for future expan sion and allows for addresses up t 14 digits long.

***Currently working on***

New version with 27512 eprom to allow applications to be stored i EPROM instead of requring a re config each time power is cycled.

*Thomas A. Moulton, W2VY*
*W2VY@WB2GTX.nj*
*100 Overlook Terrace Bloomfield, NJ 07003 (201) 338-W2VY*

# 2400 Baud Applications: Effective Throughput Analysis

*Copied from NEDA Quarterly June 92*

When I first heard about the availability of 2400 baud modems for packet, I was very interested. Twice the speed, although not terrific, was still better, and the 9600 baud alternative required special radio characteristics and there was little knowledge about making 9600 baud work thru the available radios. The "thru the mic" interface 2400 baud allows was at least a partial solution.

Then over the next couple of days and conversations with several individuals, reality started to set in. It was not twice as fast. I realized at 2400 baud an acknowledgment packet takes virtually the same time as at 1200 because of the radio key up times. Add to that the key-up time when starting to send the next burst of data packets and other bits of dead time and very quickly the increase in speed gets dissipated. Under optimum conditions with short Tx delays and large frame sizes, a 50%+ increase in speed could be achieved, but more typically only a 30% increase in speed might be experienced. That is assuming there is no other activity on the channel. If there is some other activity, or a simplex hop is used, the send\ack cycle gets lengthened and any speed benefits gets virtually lost. One of the few benefits that helps is that with shorter bursts the probably of getting hit by a hidden transmitter is reduced. All in all it did not appear that the cost of the 2400 baud modems was worth the performance increase. Although I did experiment, I like many other of the system operators had pretty much written off 2400 baud.

Over the past couple of months, I detected a flaw in my logic. The flaw is I had assumed as channel congestion increased, the speed advantage would be further diluted. This is not entirely true however.

I find prevailing operation conditions differ now from the past when I had made my original conclusion about 2400 baud. Many sites are now backboned to servers and simplex hops are much less common, while at the same time other users on the channel at the same time are more common. Typically what may be experienced when working a server directly or a server thru a node is burst of data separated by significant pauses. This is due to activity causing delays while the server or node waits for the channel to clear to send more data. Sometimes it is because of several users of the same port and air time gets divided between the users. A typical user may see a 2 second burst of data every 15 seconds or so. This can put the average throughput down around 200 baud. Under these circumstances a 1200 baud HTS free link has no trouble supplying data to a user port.

In my logic, I only made comparisons between what 1200 baud can do on a clear channel compared to 2400 under the same. I did not consider busy channels like the above. When the channel is busy the send/ACK turnaround times become insignificant compared to the time waiting for other data to pass. As mentioned above, at 1200 baud a 2 second burst of data with a 10 second pause will result in a average throughput down around 200 baud. At 2400 baud with the same TNC parameters, the user may see a 1 second (same amount of data sent in half the time) burst with still 10 second pause (since other users may still be using 1200 baud). This results in only a increase to an average of 218 baud from 200 baud. Again a minimal increase and my logic appears to hold.

In my earlier logic, I did look at the effects of increasing frame size and number of frames sent in one burst and found the send/ACK turn around time still significantly cut into speed improvements at 2400 baud. Now lets assume on our busy channel, we configure the station running 2400 baud to send twice as much data in one burst (i.e. maintaining the same length burst). Then at 2400 baud 2 seconds of data are received with a 10 second pause which yields an average of 400 baud. There it is 400 baud instead of 200 baud! It is truly twice the throughput a station running 1200 baud would see. The reason we now experience this increase is that the send/ACK turnaround time is no longer a factor.

Now lets assume we set up a dual 1200/2400 baud userport. Users then would be able to upgrade their TNCs and see this performance gain. It also should be noted that there has not been significant advancement in basic TNC2 hardware for a long time and many of the users may be willing to make the $60 to $70 upgrade since it has been years since they purchased their TNC. If we further assume that a significant number of the users make an upgrade to 2400 baud, that 2 seconds of data with a 10 second pause may change to a 2 seconds of data with a 7 second pause because another station using the channel you would be waiting for, could now be running 2400 baud reducing the delay time between data bursts. This further increases the average throughput to 533 baud which is more than 2.5 times as fast as things were when everyone was on 1200 baud. Who would have guessed that a 2:1 increase in data rate in fact increase average throughput greater then 2:1. I am not pulling your leg. Keep in mind that I am talking about average throughputs that are down around one quarter of the actual data rate because of congestion.

Through the above logic, I have concluded it is worth investing in 2400 baud userports where the prevailing conditions are as I described. Whenever experienced average throughput is one third or less of the maximum obtainable average throughput, 2400 baud properly configured will provide a 2:1 increase. If enough stations operate at 2400 baud greater then 2:1 can be experienced. A special 2400 baud only LAN would provide even better performance to those stations that upgrade.

However it should be noted operation on a uncongested channel at 1200 baud will perform nearly as good as 2400 baud under the same conditions. There are some other issues that should also be considered such as what about all those TNCs with open squelch DCD capability that may be on the channel you are trying to operate 2400 baud? It is true they will not detect the 2400 baud. I submit it is not as big a problem as you may think. The additional collisions relative to the hidden transmitter problems already experienced on most user LANs will be minimal. That is another long discussion to which I will not go into here. For a dual 1200/2400 baud userport I recommend a open squelch detection circuit which detects both 1200 and 2400 baud rather than squelched operation. This is not difficult to implement, I'll try to get together something for the next *Quarterly*.

**—Bill NX2P**

## Making K9NG Modem Work With More Radios

*Copied from Quarterly feb 92*

The following mods should be useful in making the k9ng work better with more radios:

Data-derived Rx Clock may be derived from U4 pin 9.

Half an LS02 can be used to buffer and gate RXD and RXC with DCD/, which will really cut down on the number of frame aborts in the switch or TNC.

R31 should be changed to 0 ohms, and C18 should be increased to .1 uF to improve the DCD circuit. Cuts down on chattering.

If you don't have a 16x or 32x clock available, a 4060 and a 4.9152MHz crystal will get you one for about $3 total.

At 4800 bps, no changes are needed to the input RCV Filter, but at 9600 bps, the capacitor values are quite critical because the low-pass filter cutoff is too close to 4800 Hz. If your radio has a decent IF, you can just cut the values down and you'll get fewer damaged received packets. (In fact, what I do is reduce C13/14 from a parallel combination of 2700 and 1800 pf to just the 2700 pf, and use the single 1800 pf left over instead of C15/16, which used to be 2200pf + 56pf.)

Change C19, a 220uF capacitor, to something like 6uF to avoid frying the regulator.

For a TNC-2, jumper the modem header pin 3 to pin 4 so TheNET can find itself. Don't cut the jumper on 9-10, so that the keying circuit inside the TNC will continue to work. Do cut the jumper on 7-8. Putting a jumper across 1-2 will let the DCD light on the TNC work.

Hanging a diode from U2 pin 6 to the RTS/ line (solder it from the right-hand side of R26 to the right-hand side of D3) will inhibit the DCD/ output while you're transmitting, which might confuse things.

Note because of the hang time of the DCD circuit, you WILL get a brief burst of DCD after you unkey, but that's unlikely to cause any problems even in a critical device driver, since you will have already told the interface to deassert RTS/, and presumably you're therefore ready for incoming carriers.

Oh, and of course you can leave out all the stuff after U6 pin 10 and U2 pin 14. That's all just DC switching stuff you won't need if you're using the TNC's PTT or some external PTT. If you are using the modem without a TNC, you can use U6 pin 10 to drive a transistor, or a 555 and a transistor if you need a blab-off.

RTS/ is not asserted.

**—Brian Kantor wb6cyt**

## Radios Suitable for 9600 Baud Packet Use

*Copied from Quarterly feb 92*

Motorola MAXAR and MITREK have both been modified for 9600 by Brian wb6cyt, for the k9ng; these will work with the g3ruh as well.

There are some 9600 baud ready radios, like the TEKK KS-900, Kantronics DVR 2-2 (not recommended), and D4-10, but most of us will be using our existing radios for 9600 baud packet. Here are some specific "mods" and tips.

From James Miller g3ruh:

Radios known to be used at 9600: Alinco: DR-1200 DataRadio, ALR-72, ALR 709; Kantronics DVR 2-2 Data Radio; Icom IC:22, 25, 38, 228, 271, 290, 471; Kenwood TR: 7500, 7700; TM: 211, 212, 221, 231,431, TS: 700 and 770; Standard C58, C140; Yaesu FT series: 212, 221, 230; (We may assume this is with the g3ruh modem)

Icom IC290H/V: RXA may be obtained at IC12, pin 9, on the main board; TXA may be injected at D-3 cathode on the main board, through a 680 ohm resistor. This one is my personal 2 meter 9600 rig, and works great with either modem.

*This material was passed on to me by WA2ZKD who copied it from a note sent out by Mike Curtis, WD6EHR who had provided the info in one of his postings to a 9600 Baud discussion group on Internet -Ed*

# Converting the ICOM IC471A for 9600 Baud Operation

*Copied from Quarterly June 1992*
**Don Lemke, WB9MJN**
**24 March, 1992**

This document details the modifications necessary to operate 9600 baud .5GFSK data modulation thru an Icom IC-471a transceiver. This is the modulation that was originally developed for Ham Radio usage by Steve Goode, K9NG, and also employed in the G3RUH and TEXNET modems.

### *Modulator*

The IC-471a has a "Direct FM" Modulator. This modulator uses a varactor, D4 on the "MAIN" C.B. (Circuit Board). This modification patches the 9600 baud FSK Transmit Audio into the modulator circuit, in a way that transmit audio, and the FSK audio have minimal distortion. As with any 9600 baud modification to a "Direct FM" modulator, readjustment of the audio deviation is recommended, after the modification.

Step 1: Replace C17 with a .47 uF, 25 w.v. non-polarized capacitor. The capacitor should be physically appropriate in size. I use one that had a .3 inch lead spacing, was about .4 inch square, and one eighth of an inch thick.

Step 2: Attach on the bottom of the C.B., a 3.3 KOhm, 1/4 watt resistor to a C.B. trace between R30 and C17. Leave the far end free.

Step 3: Attach a .001 uF, 50 w.v. disc ceramic capacitor to a ground C.B. trace near the free end of the 3.3 KOhm resistor. Twist the free end of the resistor and this capacitor together.

Step 4: Attach a piece of small coax to ground, and the junction of the resistor and capacitor. I used RG-178 coax. The center conductor gets attached to the components. Insulate the connections with Fiberglass tape, so when the C.B. is reinstalled the connections will not short out against the chassis, or the C.B.

Step 5: Solder the other end of the coax to MOLEX socket terminals (#02-06-1103). Insert the center conductor terminal into the "Auxiliary" connector position number 9, and the ground into "Auxiliary" connector position number 17. This completes the Modulator modification.

### *TX/RX Turnaround Time Improvement*

The stock IC-471a applies power to its TX'er audio stages only during transmit. Thus, when the transmitter comes on, RC time constants in these circuits cause the transmitter to drift for approximately 1 second. To cure this problem, this modification reroutes the DC power for Q1 to a filtered +8V point, and enables the IC1 stages continuously. This modification is to the IC-471a "MAIN" C.B..

Step 1: Cut free the long lead of R26, which should be towards D3.

Step 2: Solder a piece of jumper wire onto the free lead of R26, and insulate connection with heat shrink tubing.

Step 3: Solder the far side of the jumper wire to IC1, pin 8.

Step 4: Solder another piece of Jumper wire to IC1, pin 8 and far end of this wire to the junction of D1 and R4. This completes the TX/RX Turnaround Time Improvement modification.

### *Receiver FSK Audio*

The IC-471a uses a MC3357P Receiver IF chip. The discriminator output of this chip has a low enough impedance that it can directly drive a shielded wire without the signal being distorted. In addition, this audio signal is provided an unused C.B. header, J14 on the "MAIN" C.B. which is easily used to connect the signal to the rear panel "Auxiliary" connector.

Step 1: Obtain a .1 inch spacing, 2 position, socket which will fit onto J14 and solder subminiature coax to the socket, so that the center conductor socket will mate with the J14 pin on the "R214" side of J14. The ground lead is soldered to the other position of the socket. The MOLEX catalog shows a part number 22-01-2026 for the connector housing, and it requires 2 type 4809c terminal inserts.

Step 2: Solder a MOLEX socket terminal (#02-06-1103) onto the other end of the coax, ground lead.

Step 3: Solder a .47 uF, 25 w.v. capacitor to another MOLEX socket terminal. Connect the coax center conductor to the other lead of the .47 uF capacitor, and insulate this connection with heat shrink tubing.

Step 4: Insert the coax ground conductor terminal into the "Auxiliary" connector pin 15, and the capacitor terminal into pin 11.

This completes the receiver FSK audio modification. Bringing it all together: The MOLEX part number for the "Auxiliary" connector mate is 03-06-2241, and the pin terminal part number is 02-06-2103. The connector has 24 pin terminal positions, so if you plan on using all these positions someday, order 24 of the 02-06-2103 part number.

### *Comments on filters*

The IC471A has the ideal Receiver IF BW for 9600 baud operation. As part of this project I measured the Receiver IF BW to be 14 KC at -6 dB. The 455 KC Filter part number spec is 15 KC BW at -6 dB. The IC471a also has a crystal filter for the transmitted signal. On the RF.YGR C.B., FL2, a part number 70M15A, is part of the transmitter. The part number indicates the filter is 15 KC wide at 6 dB down, I believe. The effect of this filter is to reduce the single bit transitions of the K9NG modulation, by a small amount and time delay distort the signal a small amount. All I can report, is this is not causing a big problem here on the terrestrial 9600 baud LAN, where other users have TEKK and IC475A radios. The IC475A has this same filtering. I am looking to replace this filter with a 20 or 25 KC wide at 6 dB down filter.

In a complete transmitter to receiver path, the signal passes two time delay distorting filters, of the best BW for detection, for these type of filters. This de-optimizes things some. This report will be sent thru the IC471A, 9600 baud station here, to the network, however, illustrating the effect is small.

### *IC-471a problems*

This IC-471a was used when I obtained it. Looking at the manual spec for receiver sensitivity, of .5 uV for 20 dB Quieting, many of you might cringe. After looking at the receiver line up, and the parts used, this performance seemed somewhat odd. This is not a radio with a 6 stage helical resonator front end filter, and J-fet mixer! That type of radio would typically give the IC471A, specified performance. The IC-471a is designed to have considerably better sensitivity than this, although obviously at the expense of front end overload handling. Something is wrong.

Measurement of my used IC-471a showed besides the production and/or design problem I suspected, something was really wrong with my particular radio. The sensitivity varied dramatically with temperature. It would start out on spec, then slowly get as bad as -90 dBM for 20 dB quieting. This is BAD!

I tracked the variable sensitivity to a problem in the PLL box. As the unit warmed up, its output would vary from around .75 vrms to less than .1, on center band. I never "fixed" the PLL, but by removing the cover, and putting 1/4 inch spacers between the PLL enclosure and the module that is mounted over the PLL box, the PLL stabilized with enough output, that near spec sensitivity was maintained.

Next I went after the poor spec. While finger testing L29, the sensitivity improved, greatly. After replacing L29 with a new coil, with an additional turn (3 instead of 2), and the sensitivity improved about 20 dB. A 20 dB quieting signal needed less than -110 dBM, which is about .2 uV. This is the typical range for UHF receivers of this design! L29 is part of a PI circuit used to transform the PLL output impedance of 50 ohms to a very high voltage and impedance needed by the first receiver mixer stage, Q12. At this sensitivity, with a 19 element F9FT Yagi, UO14 9600 baud was very easy to copy for most of the pass, even without a preamp.

I include these experiences in this article on how to do 9600 baud packet, to hopefully make it clear to people even a pretty looking radio can perform like garbage, and if the radio isn't working in the first place, the 9600 baud packet radio is not going to either. Nobody should expect they will be able to do 9600 baud Packet radio, with existing rigs, like they would do 9600 baud telephone line communication. Yes, you may actually need to learn and do some hardware electronics to do this.

### *Concluding remarks:*

With these modifications, the IC-471a makes a nearly ideal 440 band 9600 baud packet radio. I've used mine on UO14, as a receiver, as well as over our 9600 baud terrestrial LAN. A PING test thru N4PCR-0/1, which uses a TEKK radio, and preamp, and one of the CELLNET 56 KB, FDX radios described in the file CELLNET.UPT, to the K9VXW-1 node, where the other CELLNET prototype FDX radio is, passed 98 of 100 kilobyte packets with the NOS Window = 2048, MSS = 216. I hope this article is helpful, and will encourage some of you to take the plunge and try out 9600 baud packet radio.

All commercial use of the information developed by the author in this article is reserved. Like somebody said, "I'm not making money at this, and I don't want you to make money off me either". In other words, you are free to do these modifications for yourself, or have somebody do them for you, if he or she gets nothing in return for it.

**—73, Don WB9MJN @ N9HSI.il**

---

## NEDA Makes Front Cover of CQ Magazine

*Copied from NEDA Quarterly June 92*

For those alert members who are readers of CQ magazine the June '92 issue should have caught your eye. NEDA member Peter Brayman, KB2HPU is featured at the controls of his school's club station twiddling the knobs of a Kenwood HF rig. Prominently visible on the desk in front of Pete are a set of NEDA Network Wide and Eastern NY Regional Maps.

Pete, who is no novice to keyboarding about the network, frequently assists your Quarterly Editors in verifying node configurations for map updates. He also is one of the remote sysops for the Oneonta Amateur Radio Club's MSYS BBS, W2RGI. OARC's members are the primary support group for New York State's Emergency Management Office in Oneonta, NY. The EOC in turn is used as the home location for the W2RGI BBS.

Hats off to Paul Agoglia, WN2K who is the Unatego (NY) High School teacher responsible for getting the School Amateur Radio Club going, and to Pete Brayman network hacker extraordinare, for the fine promotion job in their area!

**–Dana WA2WNI**

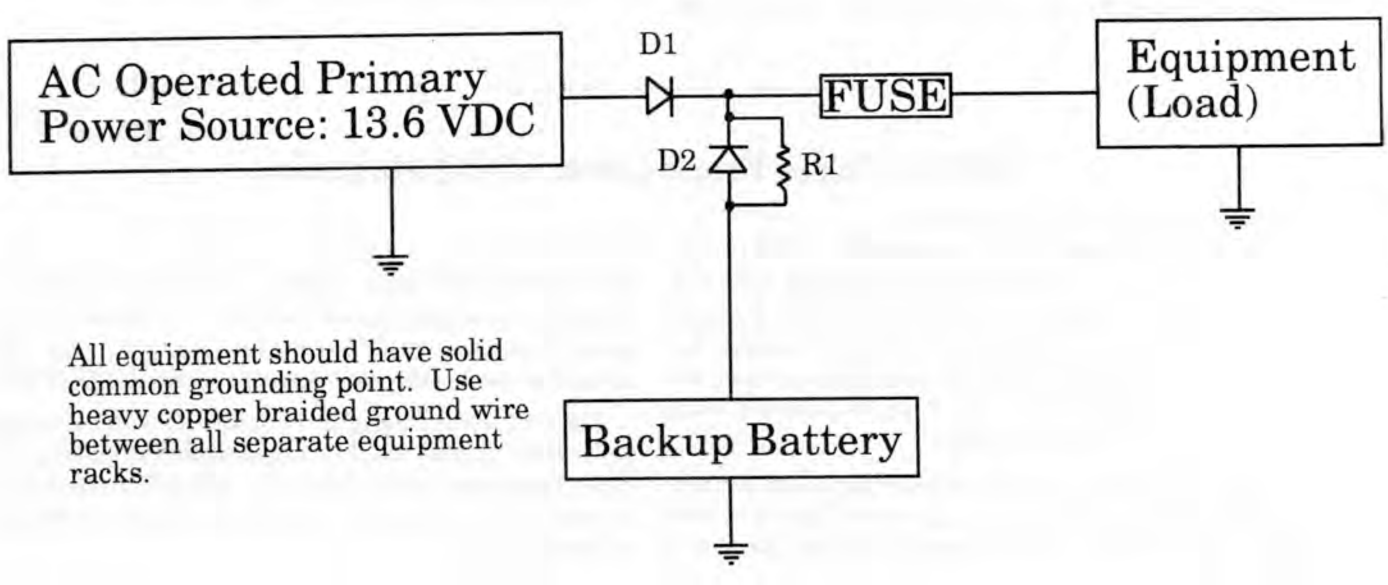
# Network Station Hardening - Preparing for the Worst of Times

*Copied from NEDA Quarterly June 92*

This is an article about how to "Harden" your packet station or network site for emergency applications. Site hardening is a term that comes from military specifications for equipment intended to operate under very extreme conditions. While it is true that what we do is mostly hobby related, there is no reason why in some cases we cannot take reasonable measures to insure that our equipment does not become useless at the drop of a hat, or as in this month's column, the drop of your local AC power line during a storm. Hopefully over time the *Quarterly* can cover some of the more common conventions for what is good engineering practice to protect your station or node from various forms of hostile environment. For this month though I am only going to provide a simplified way to keep your equipment powered up should your power source die or bite the dust.

There are many forms of backup power systems, some are good, some are only mediocre when considering cost, ease of installation and overall effectiveness. The system shown in the diagram here is profoundly simple and I have been successfully using this configuration for a number of applications for at least 12 years. As you can see there are no switches, relays, sensors or other gizmos to clutter up its effectiveness. The entire trick is done by isolating 2 power sources, one AC powered and one from a standby battery, using a simple set of blocking diodes. The battery should be of sufficient size and amp/hour rating to run whatever equipment it is connected to for the amount of time you wish as an emergency operating period. Ideally this should be 24 to 48 hours of run time if your site is not easily accessible and is required for emergency use from time to time. More capacity is desirable, but not always feasible due to power/cost restraints. The diodes should be whatever new or surplus monster diodes you can muster that will easily handle the continuous loading of your equipment PLUS your battery bank if it is being recharged. The main AC power source should be capable of operating all your equipment at maximum load AND be able to recharge your battery bank thru charging resistor R1 at the same time. The fuse is to protect the equipment from damage if anything goes wrong with your setup and allows obvious protection for the power supply and battery if the equipment is the thing to go west. The really neat thing about this setup is that you can make it as BIG or as little as you want, merely scaling the sizes and ratings of the components to match your application. My smallest application is a 12V 5 A/h cell on 1/2 of a Radio Shack 2 amp bridge rectifier being fed by a 500 ma 13vdc wall transformer , bridged by a flashlight bulb for R1 and the whole array keeps a TINY 2 alive for an EOC mini-BBS with a low power radio. The largest application so far uses 2 large stud 50 amp rectifiers and floats a set of 2, 12vdc 40 A/h wet cells. This setup uses an automotive headlight for R1 to limit return current to the cells and the array feeds the main DC power buss to hold up all the radios on the backbones of a major node site. In such setups it is important to set the float voltage getting to the cells at full charge such that the batteries are not being constantly overcharged. This can lead to dangerous outgassing in some cases, or overheating of the cells. It is important to take precautions in such cases to make sure that adequate ventilation is present and that the cells are not charged or floated at a rate beyond specs.

If you wish it is possible to add all sorts of bells and remote whistles to this setup. For example, if you put a continuous duty relay on the AC source side of D1, and route all non-essential equipment power thru its contacts, then when the power goes out, all the non essential equipment goes instantly off line, leaving the more




All equipment should have solid common grounding point. Use heavy copper braided ground wire between all separate equipment racks.

essential stuff on line. When commercial power is restored, the powered down stuff comes back up. Simple yes? Remote control of TNC's and peripheral hardware can also be achieved by using the LED ON / LED OFF feature of some TheNET code. By using a suitable switching method a node manager can remotely control the power on equipment that might best be shut off when the node is needed for some emergency application.

This is just a primer to get your brain in gear. Hardening a station is sometimes no more than doing some simple things to make it less susceptible to being interrupted. Power is just one of a number of things to take into consideration. Next time we will look into some simple steps to effective lightning /surge protection. Meanwhile, just make sure your AC powered supplies are plugged into readily available commercial surge protectors. At a few bucks a piece it isn't worth NOT having one in line with your gear! You may also be amazed to learn how cheaply you can DC ground your antenna systems! Remember that in the packet network, the total throughput effectiveness is only as good as the weakest site. Don't let the weak link when we need the network be yours!

**—DANA WA2WNI**
**NEDA Emergency Services Advisory Committee**


## DOERS and the RAND Node

*Copied from NEDA Quarterly June 92*

Realizing the need to extend the NY State amateur packet backbone network to the north-eastern part of the state, the DOERS (Digital Operator's Emergency Radio Service) was established by KC2JO (John Taylor) and KD2AJ (Chuck Orem) in the fall of 1990. In December of 1990 a coordination meeting was held in Plattsburg to determine packet priorities and to discuss with packet groups in the adjoining areas the best ways to serve the region and remain compatible with their needs. As a result of that meeting the DOERS became determined to not only complete the 4800 baud backbone north from the Saratoga region, but to also plan paths to Quebec to the north, Vermont to the east, Malone/Massena to the west, and Tupper Lake to the southwest.

We are proud to announce that after a good deal of time, money, and hard work the 4800 baud link is complete to RAND:W2UXC-4 (on Rand hill, 8 miles northwest of Plattsburg), and on to Alberg, Vermont (WA2SPL).

RAND:W2UXC-4 was set up as a repeater node in order to deal with the hidden transmitter syndrome. The local userport, NYPLB:W2UXC-9, is on 144.91 MHz. Those connecting to RAND via the backbone (from the south for example) at present need to first C KT2M-1 to get RAND on the nodes list.

Special recognition should go to Carl Smith (N2JKG) DOERS President and our technical wizard, Burt Lang (VE2BMQ) site manager for the VE2RM site in Quebec, Dick Jenkins (N2AYY) site manager for the WA2UMX system, and Jim Mcknight (WA2UMX) sysop for WA2UMX BBS.

In addition to the 4800 baud backbone link DOERS maintains the 145.01 PLB:W2UXC-1 node at Lyon Mt., NY and the 145.61 KD2AJ-1 Digi at Rand Hill.

DOERS would be glad to share any additional information regarding the 4800 Baud Repeater.
Send packet to:
N2JKG @ KD2AJ.#NNY.NY.USA.NA.

**—N2IXL - v.p. DOERS**

---

## Baycom Software

*Copied from NEDA Quarterly June 92*

I have seen several bulletins over the past couple of months asking about Baycom version 1.50 software. Due to my commercial connections, I have not been able to answer these questions over the air.

For you who have not heard, version 1.50a is now available. One reason for the confusion is that this version unlike earlier versions is not shareware. Therefore it has not been available for download and was not announced "formally" thru packet channels. PacComm has contracted with Baycom to represent their interests in the United States. The new 1.50a version of the Baycom software comes with a new printed manual and is available from PacComm, myself and other licensed agents. This software goes for $20.00 with the manual. PacComm also offers licensed Baycom hardware manufactured here in the US. The total cost of a package for hardware and software starts around $65.00.

**-Bill NX2P**

# Quality Values vs TheNET Node Propagation Distance

*Copied from Dedicated Link Sept 92*

NAPRA specifies a node propagation distance of 7 hops. That means that 7 TNCs away from a node the node will be visible but 8 TNCs away it will not. Assuming that each TNC to TNC over the radio hop is a point to point link and each TNC to TNC local hop is over a Hexipus each node site will be visible three node sites away. The reason for this number is :

- If the network is built well and if networking is at all popular the number of nodes that will be three node sites away starts approaching 100. Four node sites away will reach a number correspondingly higher. Keeping node lists short, is for TheNET a good idea for many reasons.
- By reducing the number of sites each connect command can traverse we improve the reliability of each connection.
- Because the quality value of a node is reduced drastically with each hop if a link goes down the node's which are no longer available disappear fairly quickly. This also goes for nodes which are placed on the air temporarily.

NAPRA makes the recommendation of 203 for a broadcast quality for both RS-232 and radio links for all TheNET nodes and 161 for all single CPU nodes (like G8BPQ, MSYS, NOS).

The value used for Time-To-Live is used to limit the number of hops that a transmission may travel across the network. This is useful for reducing the effect of bad routing which occurs when a link fails. If Time-To-Live is set to a value less than the number of hops that a node may propagate (set by the quality values) then users may attempt to connect to nodes that can not be gotten to. This causes immense frustration. Please do not allow nodes which are unreliably connectable to show on the lists of nodes that are in your control. By keeping your node lists down to what is usable you do a great service for packet radio. Users are our future.

Use this program to determine how far your nodes will propagate. If Time-To-Live is set too low, either set your quality values lower, set Time-To-Live higher, or re-think how things are working.

**—KA2DEW**

```
10    msgqual = 256:  rem initial value for a node
20    hopnum = 0:     rem we haven't started hopping yet
30    minqual = 50:   rem minimum quality for all nodes
40    PRINT ""   :    rem blank line
50    PRINT "at start node"
100   REM start of loop
110   PRINT "message qual at hop #";hopnum;"=";msgqual
120   INPUT "Enter quality for this hop ->";hopqual
130   msgqual = msgqual * (hopqual / 256)
140   msgqual = INT(.5 + msgqual)
150   hopnum = hopnum + 1
160   IF msgqual < 50 THEN 2000
170   GOTO 100
2000  REM end sequence
2010  PRINT "message quality is below min-qual."
3000  END
```

# New TAPR kits introduced

*Copied from Dedicated Link Sept 92*

Tucson Amateur Packet Radio (TAPR) introduced two new kits at it's annual meeting: a 9600-bit/s modem and a satellite tracking antenna controller

***9600-bit/s Modem***

The new 9600-bit/s modem kit incorporates many enhancements over the K9NG modem kit which has been available at TAPR for several years. The new kit offers full-duplex operation, improved transmit spectrum, improved clock recovery, DCD and an optional bit regenerator for use as a full-duplex 9600-bit/s repeater. An optional clock is available for stand-alone bit regenerator usage or elsewhere where not provided by the TNC. The modem connects to the standard TAPR modem disconnect header and fits inside a TNC 2, PK-232 and many other TNCs. The kit costs $70 including shipping and handling in the US. The bit regenerator option is $10 and the clock option is $5.

***Satellite Tracking Antenna Controller***

By arrangement with JAMSAT, TAPR is offering in KIT form the TrakBox developed by SM0TER, JA6FTL and others. This requires a computer or terminal to enter the Keplarian data. It has an LCD display and runs stand-alone after loading the data. It directly controls the Kenpro/Yaesu rotators and can be adapted for other types. It also provides Doppler correction for the Kenwood, ICOM and Yaesu radios that are used for satellite communications. The kit costs $185 including shipping in the US.

*Both kits are available now from*
*TAPR*
*PO Box 12925*
*Tucson, AZ 85732*
*602 749-9479 Tues.-Fri, 10AM -3PM MST*

**Bob Neilsen, W6SWE,**
**via Compuserve's HamNet**
**from May 1992 Gateway/QEX/ARRL**

# Notes on FT736 & 9600 Baud Operation

*by James Miller G3RUH*
*Copied from Dedicated Link Sept 92*

These notes tell you where to get FM RX audio direct from the discriminator, and where to modulate the FM TX varactor directly. These mods are non-destructive and take no more than a few minutes. The signal by-pass the "DATA SOCKET" for high grade FM operations.

***The RX mod is suitable for:***

- UOSAT-D 9600 baud downlink and terrestrial links
- 1200 baud AFSK/FM Standard Packet - BUT IT'S UNSQUELCHED.

***The TX mod is suitable for:***

- FO-20/PACSAT uplink (1200 bps Manchester FM)
- UOSAT-D 9600 baud uplink direct FSK and terrestrial links
- 1200 baud AFSK/FM Standard Packet.

### FT736 - FM Direct from Discriminator

Detected FM direct from the receiver discriminator is available from the RX UNIT at the junction of R91 and C83. These components are shown in the top right-hand corner of the schematic.

Proceed thus:

1 Disconnect FT736 from the mains electricity. (Safety).

2 Remove top cover only.

3 RX Unit is the vertical module on the left.

4 Locate R91 which is about 25mm from the top, 50mm from the radio rear. The resistor is "on-end", and near a couple of glass diodes.

5 Scrape any paint off R91's free end and wet with solder.

6 Your RXaudio lead should be a fine screened cable; connect the inner to R91, and the outer braid to a ground point (e.g. can of TO09)

7 Route the cable out though any convenient aperture in the case.

8 The discriminator sensitivity (FM Normal) as about 6 kHz/volt.

### Important note on 9600 Baud Use

Some FT736 receivers are fitted with an LFH12-S IF filter for FM. (CF01 at the top front of the RX Unit). This is a 12 kHz bandwidth filter which is a little too narrow for 9600 bps FSK operation. It is recommended you change this to 15 kHz or better still for UOSAT-D use, 20 kHz bandwidth which will allow more tolerance for doppler shift, and give a far better "eye". Suitable filters are: LFH-15S or CFW455E, and LFH-20S or CFW455D. The first of these is a Yaesu spare part, and is often already fitted.

### FT736 Direct Varactor FM Modulation

Refer to the circuit diagram; inject your TXaudio at the junction of R32/C29 on the TX Unit. The signal level at this point should be 800 mV peak-peak, and will give +/- 3 kHz deviation. ***Do not exceed this level***.

Set Mic Gain to min.

Modulating the FM transmitter this way you get an LF response down to 18 Hz (at which point the associated synthesizer PLL begins to track the modulation), and an HF response which is flat to some 10 kHz.

Proceed thus:

1 Disconnect FT736 from the mains electricity. (Safety).

2 Remove top cover only.

3 TX Unit is the module flat on the left (not the one tucked down the side vertically).

4 R32 is just to the left of the rectangular shielded enclosure. The resistor is "on end". Scrape any paint off the free leg.

5 Your TXaudio lead should be a fine screened cable; connect the inner to R32, and the outer braid to the adjacent enclosure.

6 Route the cable out though any convenient aperture in the case.

7a 1200 BAUD PSK MODEM: TXaudio of 800 mV pk-pk can be obtained by adjusting the components C9= 1uf, R3=47k, R5=infinity (i.e. remove). C10 stays at 10nf (0.01uf).

7b 9600 BAUD FSK MODEM: Adjust TXaudio level with VR1

*Notes compiled by G3RUH @ GB7SPV 1990 Mar 16*

### Modification to the Yaesu FT-736R

by G4WFQ 12/1/91.

This modification was given to me by Zeno Wahl, G0NJC/VE3LMX (U.O.S)

The modification lowers the frequency response to 3 HZ, and gives a far better "eye" by reducing L.F flutter.

Proceed thus.

1 Locate "TXPLL UNIT" (Vertical board on Tx unit).

2 Locate R01 (Scrape any paint off. Wet component with FINE solder.

3 Solder 560ohm Resistor on R01 (end nearest to pll board) Solder 47microfarad tantalum in series with 560R. Take (-) negative leg of Cap to Gnd, eg case of Txpll unit.

**—G3RUH & G4WFQ**

# TheNET Plus 2.11 released

*Copied from NEDA Quarterly of November 1993*

NJ7P has released a new version of TheNET Plus. This appears to be a bug fix for 2.10. There are no noticable additional 'features' over 2.10. Here is the official release blurb that was included with the 2.11 distribution.

*To summarize briefly, version 2.11 has several new features over those found in version 2.08:*

1 *Automatic reconnect. If one is "BYED" out of a distant connect, the circuit will automatically be reconnected at the first 2.09, 2.10 or 2.11 node back from the "BYED" point.*

2 *Improved SYSOP parameter setting feature.*

3 *Transport layer parameters are now remote SYSOP settable.*

4 *A specialized version 2.11dx for DXCluster support.*

5 *A specialized version 2.11tm (telemetry) is available upon request.*

6 *USERS display has been revised to show level 3 thrulink source and destination nodes. (Helps users and NodeOps alike to keep tabs on system activity.)*

7 *The diddle feature of 2.10 was removed. It was unreliable and required setting TXD to an odd value.*

*8 *A completely revised and expanded documentation and support package. The Network.exe support package includes a nice VHF/UHF/microwave radio path analysis program as well as network design info.*

*Be advised the node DIGI function had to be removed from version 2.11 in order to make room for the new features. This occured with the version 2.09 developmental nodeware as well. If one needs the digi function for their particular application, best to stay with version 2.08.*

If you are unable to find a copy of 2.11 on a BBS you can send a 3.5" 720K or 1.44MB floppy and a SASE (capable of handling the floppy) to NEDA's POBox and your floppy will find its way back to you complete with software.

**—EOF**

# JNOS BBS

*Copied from NEDA Quarterly Summer/Fall 1992*

Most NOS implementations include a BBS function which looks a lot like a W0RLI or AA4RE style BBS. Many packeteers and most BBS ops have a rather negative view of BBSs based on KA9Q's TCP/IP implementation (which all NOS versions are). That image should be undergoing a major change with the BBS addition by WG7J. I have been watching its development in Corvallis where it is the BBS accessible from the CRV node.

First, perhaps, we might ask, "why, in heaven's sake, *another* BBS program?" The reason is that BBS programs which offer TCP/IP connectivity are few. MSYS, which attempts to do this, has flaws when viewed from a TCP/IPer perspective. WG7J has also added enhancements which will make his program stand on it's own in the BBS arena.

Johan (WG7J)'s version of TCP/IP operates for a long period of time without human intervention for restarts. Periodic crashes due to memory usage bugs has been known to be standard fair for MSYS and NOS in the past. Not so with WG7J's version. WG7J's NOS behaves as a node if desired (somewhat like MSYS). Johan has made the node interface more *normal* to users and improved the BBS so that it provides standard forwarding. The result is a program which occupies much less memory space than MSYS, which implements TCP/IP properly, and is a BBS with some differences.

The differences will be quite noticeable to anyone who uses any of the *standard* packet BBS programs. The most obvious is the *area* concept which is standard with land-line BBS operations including CompuServe. This saves you from going through all of those for-sale bulletins when all you want to see is ARRL or RACES. The BBS places bulletins into these areas (which are specified by the SYSOP). The BBS will tell you what the predefined areas are if you send it an ***a*** or ***areas***. Doing this on CRVBBS gives me:

```
a
Current message area is: ka7ehk

Available areas are:
ka7ehk              Your private mail area
allor       -       All of Oregon stuff
allusa      -       ALLUSA bulletins
allusw      -       ALLUS-WEST bulletins
amsat       -       AMSAT bulletins
ares        -       A.R.E.S. bulletins
arrl        -       ARRL bulletins
dx          -       DX bulletins
fcc         -       F.C.C. related issues
help        -       help about this system
humor       -       humorous stuff (whatever that
means :-) )
ntslocal    -       Corvallis area NTS traffic to be
delivered
osuarc      -       Oregon State University ARC stuff
pnw         -       Pacific North West stuff
races       -       R.A.C.E.S. bulletins
sale        -       'for sale' bulletins
tcpip       -       TCP/IP related stuff
wanted      -       'wanted' bulletins
```

Note that you have an area all of your own; each time you log on, you start in your personal area. To call up any other area, just send, for example "a allusa"; in this case, the response is:

```
a allusa
allusa: 131 messages   -  0 new.
>
```

**Continued on next column ➡**

## Ramsey FX-146 2m rig

*Copied from Dedicated Link Sept 92*

This article is intended for anyone planning to build or having already built, the Ramsey FX-146 radio kit.

The front end of the radio is very wide, from 130-160 MHz, which for some seems to be a blessing, but personally, I have had more than my share of intermod products and image frequencies wiping out the data/information/QSO I want to pursue, and so, finally called Ramsey with a question about the radio's bandpass filter. As it turns out, there is a filter circuit that is available from them (it was free at the time, so ask them about it) that will replace the existing front end filter circuits. You remove several parts and mount the small board in place of those parts. The change was immediate and *immense*! I no longer pick up the local police, fire, paging systems and general taxi-cab calls found within the business band (bleeding over into my *ham* radio!) My packet system is working 1000% better, and I can now *hear* stations that are a total of five miles away (Hi Hi!) Not to mention stations I didn't *know* existed, simply because the interference wiped them out. Anyway, this mod works *great*, and if you have one of these rigs you owe it to yourself to call them and request the mod, even if you don't need it in your area, you *might* someday. By the way, I live in Colorado Springs, not really known as a high density radio area, but, apparently it *is*. I hope this helps someone else.

**—Rick NONJY @ WOLKD**

---

**Continued from previous column**

In the event an area is empty, there is only the ">" response. An "L", gives you all of the new messages in that area. The BBS keeps track of the "new" messages for each user in each area. Any (new or old) can be read with R #. You can S, SP, SB, etc. to your heart's content in any area. "LM" (list mine) and "RM" (read mine) work only in your personal area. One of the quirks of this BBS is that the message numbers change as old ones are deleted; the oldest message in a given area is *always* # 1! Sysops might like the fact that version 1.01 now includes automatic message and old bid expiry.

If you want more information about this version of KA9Q-NOS, contact WG7J@ WG7J.OR.USA.NOAM. It is also available on Internet and several land-line BBS's. I can recommend it quite highly. The node "shell" is also well thought out, and, when used, is very straight forward. Sorry, but this is PC compatible only!

**—Jim Wagner, KA7EHK**
**copy from the Operator via packet radio**

## What's A DataEngine?

*Copied from Dedicated Link Sept 92*

Several of our Northwest Oregon nodes are now us ing DataEngines. At this moment, they include PDX7 CREST, SALEM, WLYNN. NPT has one expected t go on line soon. SNOPK has another. What are thes machines?

Lets first review what is a TNC. Though perhaps no obvious, a TNC has two parts. One is a "modem" whicl converts digital signals to/from a form suitable for ra dio transmission. In the case of the 1200 baud whicl most of us use, the signal suitable for radio transmis sion is AFSK (two audio tones switched back and forth The other part is a little harder to explain; it might b called "serial-HDLC converter". It converts between non synchronous RS-232 serial & HDLC. Non-synchronou serial means that characters can come at any time HDLC refers to characters which are formatted into pack ets and are sent in a synchronous stream for at least whole packet. We need not go into the formatting of pack ets; that is another issue and lots of references explai it in some detail for those so inclined. The standard TN( has one RS-232 port (the terminal interface) and on HDLC port (the RF port).

OK, so if that is a TNC, what is a DataEngine? First a DataEngine also has an RS232-HDLC converter Unlike a standard TNC, instead of 1 HDLC port, ther are 2. The modem part of a DataEngine uses plug-i modems which are designed for the baud rate of the port There are locations for 2 modems on the PC board. I is possible to have each port run at a unique baud rate That gets the node to "dual-port". A third port can b obtained by attaching a standard TNC to the serial port The TNC is set up to operate in "KISS mode" (more abou KISS in another article).

To get 4 or more ports, there are several ways one ma go. Two DataEngines may be coupled together at thei serial ports, just like 2 TNCs in a node. This gives ( rather expensive) 4 port node. An alternative is to connec several TNCs on a "party-line" RS-232 buss at th DataEngine's serial port. To do this, the TNCs must us "multi-drop KISS" (an EPROM-only KISS version).

Well, then, why the DataEngine when TNCs are use for radio ports, anyway? Where a DataEngine shine is interfacing between networks running at different bau rates. The node problem is more than "Do I have th right modem?" There also needs to be a lot of memor (buffer) because things will come in on the highest spee port faster than it can go out other ports. Thus, dat needs to be saved until it moves out. This is more effi ciently handled than in standard TNCs.

### *Other uses for DataEngines*

A DataEngine can work in many modes. It can "lool like" a standard TNC2, a KISS-mode TNC, a TheNE node, a WA8DED "host mode TNC", etc.

**—KA7EHK, Jim Wagner**

# Transmit Deviation and Packet

*Copied from NEDA Quarterly of Summer / Fall 92*

The importance of properly setting your transmit audio level, which directly affects your transmit deviation, cannot be understated. The graph at the end of this article shows the relationship between the signal level required for 99% copy and the transmit deviation. On the right side, this relationship is shown in relative terms: performance in decibels, and power required at the transmit end. It is easily seen that a transmit audio level even slightly higher than ideal affects the path quality tremendously. Note that lower deviations have a less severe effect.

Two factors cause the degradation in performance at deviations above 3.5 kHz: First, some of the transmitted signal's energy falls outside of the receiver's passband, resulting in a lower overall signal and waveform distortion. Second, the squelch circuit response time becomes slower and is more likely to "false", or operate improperly. With an excessive deviation, the squelch circuit will close for a few milliseconds in the middle of receiving data, causing a total loss of that packet. While both of these factors occur in voice communications, small amounts of distortion and signal loss are hardly noticeable.

Nearly all TNCs have the transmit audio set too high when new. While the deviation limiter circuit in your radio will take care of most of the problem, this circuit usually permits about 5 kHz of deviation. Some radios do not have a limiter circuit, or it doesn't work very well, or is mis- adjusted. Signals with 7 kHz deviation have been monitored.

It is recommended that the transmit deviation be set to 3.0 kHz using a deviation meter or service monitor. If you don't have this test equipment, first try sending out a packet message in your local area, or asking your local network sysop - chances are someone will be happy to help. If not, the following procedure will get you close.

The transmit deviation adjustment is made by varying the audio output level of the TNC. Most TNCs have an adjustment potentiometer accessible without opening the case, but you may have to go inside. To find out for sure, read the manual. It is not recommended that you adjust the deviation limiter on the radio, since distortion will result. The deviation limiters on older rigs also don't work very well, either.

1. Obtain another receiver to listen to your transmitted signal.
2. Locate the audio output adjustment potentiometer on your TNC. (Consult the manual).
3. Put your TNC into "CALibrate" mode. (Check your manual on how to do this.

With most TNCs you can type CAL at the "cmd:" prompt and then press the letter K to key the radio (transmit). Pressing D makes it "dither" (oscillate) between transmit tones rapidly, which is a better way of setting deviation than with a single tone. If necessary, use only the lower tone to set deviation. Pressing Q shuts off the transmitter and returns you to the "cmd:" prompt)

4. While listening on the receiver, adjust the TNC's audio output upwards until the audio on the receiver does not get any louder. Find this point carefully by repeating the adjustment a few times. This is where the radio's limiter begins limiting the deviation, usually 5 to 7 kHz.
5. Now adjust the TNC's audio output level downwards until the audio on the receiver is about one-half as loud as it was. (Note: most TNC manuals say to leave the audio level at maximum - but this is often 5 kHz deviation, much too high). The audio on the receiver should sound soft but not faint.

At this point the transmit deviation should be between 2 and 4 kHz, which is much better than 5 to 7 kHz.

Some combinations of TNC and radio are very difficult to adjust, because the adjustment is very sensitive, giving you either too much audio or not enough audio. The fix for this is to add a resistor in series with the transmit audio line to the radio. Something in the range of 10k or 20k Ohms should work fine.

**—Bill Slack, NX2P**

---

# G8BPQ Interface to HexiPus

*The following text borrowed from BPQ v4.06 release docs*

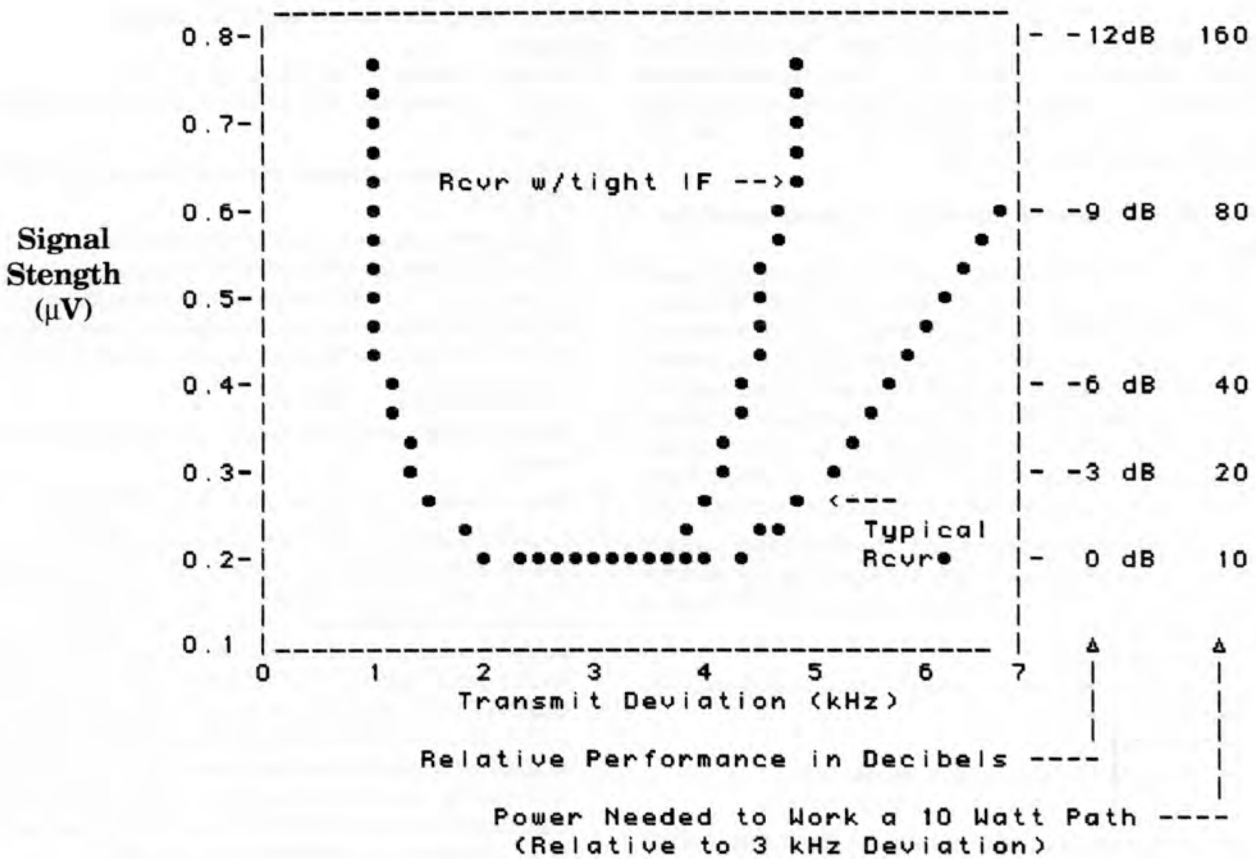
The driver code for linking to a NET/ROM diode matrix has been fixed. I had misunderstood the system, and used the wrong polarity of signals. So inverters were required to connect the PC to the matrix. Now a simple 5 wire cable with one pull-up resistor can be used to link a diode matrix port intended for a TNC2 to the PC. Remember that to operate with a matrix, you must set FULLDUP=0 in BPQCFG.TXT. This time I have tested it pretty thoroughly! **—John, G8BPQ**

*The following text converted by KA2DEW from information supplied in BPQ v4.06 release docs*

```
 PC25 pin    PC9 pin       Hexipus
        2  —  3  ———————  2
        3  —  2  ———————  3
        4  —  7  ———————  8
     —  5  —  8  ———————  7
     |
10K  R  7  —  5  ———————  5
     |
     —  20
```

*Copied from NEDA Quarterly of July 1993*

*Received Signal Strength for 99% Copy vs Transmit Deviation*




## Care and Feeding of TNC Batteries

*Copied from Dedicated Link Sept 92*

Do you ever really look at your TNC? It probably just sits there in the corner under some magazines on top of your power supply. And it just keeps working and working, right? Well, usually but not always!

W5NUI recently pointed out a potential trouble spot with many TNCs. He and KA7UPD use "early" PK232s which have the RAM backup battery mounted on the top cover of the TNC, above the circuit board. What do batteries do when they get old? They leak! And that leaking battery acid drips down on the circuit board. It does a great job on a circuit board.

This brings us to the discussion point: If your TNC is more than a few years old, it would be smart to replace the backup battery. The battery cost should be no more than a few dollars. I cannot find any reference to battery replacement in my MFJ-1270/1274 manual.

If you don't replace the battery, at least look at it every year or so. Watch for corrosion, liquid, or crystal growth. By the way, the same goes for the backup batteries in your computer and other electronic equipment including HTs, computers and smoke alarms! W5NUI suggests replacing such batteries on a regular annual event (your birthday, a holiday, etc); he also suggest taping a small tag with the date of most recent battery replacement to the outside of equipment

You should plan on loosing all of the TNC parameters when you replace the battery. I keep a list of all the non-default parameters on a card near my TNC. This helps me to remember what needs to be set and to what value.

**—KA7EHK**

## Tekk Link Radios

For a couple of years Tekk has been manufacturing digital link radios. PacComm has been selling them with their TNC and G3RUH modem in a one-piece configuration. The prices that the radios are available for varies, it seems, from $130 to $200 for the 9600 baud capable 1 watt units. I have heard recently that they were as low as $90 factory direct. Here is the information for you to contact the factory direct. Here's the information

TEKK Data Division
224 N.W. Platte Valley Dr.
Kansas City, MO 64150
phone: 800-521-8355
FAX: 816-746-1093
Service: 816-746-1098

Thanks to Linds, NR1N for this info.

**—Editor**

# HAPN-T 4800 BPS Modem in PacComm Tiny-2 MK-2 Installation Instructions

The following instructions detail the installation of a HAPN-T 4800 bps modem into the new PacComm Tiny-2 MK2 TNC to give a package that is clean (with no hanging wires out the back) and able to switch speeds between 1200 and 4800 bps by means of a front or rear panel switch or in the case of a TheNET node by remote control.

***On the HAPN-T board, make the following modifications:***

1. Drill the extra 6 holes for the new 26 pin header socket according to the directions in the revised HAPN-T instruction sheets section 11 on page 5 (If you do not have a section 11 on page 5 then you have the older instruction sheets - just drill the extra 2 rows of 3 holes using a piece of perf board as a drill guide if necessary to maintain the proper spacing). A trace from J103 pin 5 will be cut in this process - do not worry, it is not used in the modified board and should be scraped away in the area of the header pin.
2. Install the 26 pin header socket (or a short2x3 pin socket if the 20 pin socket is already installed) and the wire the extra pins as follows using short pieces of small hookup or wirewrap wire:
   Pin 21 to J102 pin 1
   pin 22 to the via hole between U1 and C28 that was connected to J103 pin 5.
   pin 24 to J103 pin 4.
   pin 25 to pin 15 of the modem header (J4).
   pin 26 to the +12 terminal hole in the corner.
3. Put a shorting jumper wire between the 2 pins at J105 (speed select). Leave the header in place and jumper it under the board. (note: the header on J105 should be a right angle type to keep the board profile low).
4. Jumper U6 (74HC132) pin 2 to pin 13. This makes DCD work at both speeds.
5. Solder a wire between J102 pin 5 and J4 (modem header) pin 1. This makes HAPN DCD work in place of the TCM3105 DCD on the Tiny-2.
6. Solder loops of bare wire as test points on the end of R9 nearest U3 (TP4) and on pin 6 of U6 (data out test point).
7. Install a 2N3904 transistor in the 3 holes in-line along the edge of the board near J105. (collector end towards J105 - case flat side towards center of board).
8. Install 47K resistor in the position beside R31.
9. Install a 5 inch length of hook-up wire in the hole connected to the end of the 47K resistor nearest the 1N5231B. These last 3 mods allow the speed to be remotely switched by applying a logic high to this wire. It can be connected to pin 1 of P4 (DE9P serial connector) or pin 17/18 of J3 immediately behind P4 on the Tiny-2 MK2 to implement this feature in a TheNET node with the RELAY ON/OFF feature. Otherwise this wire is grounded (to J3 pin 1 or wherever convenient).

That completes the changes to the modem board.

***On the Tiny-2 Mk2 board make the following changes:***

A. Solder a fine wire (like 30 gauge wirewrap wire for example) between the 1200 baud pin on JPT (or JPR) and J5 pin 12.

B. Solder a fine wire between the 4800 baud pin on JPT and J5 pin 16.

C. Locate the trace on the top of the board that runs from J5 pin 12 under the end lead of R6 and goes between pins 15 and 16 of U14 (SIO chip). Scrap the solder mask off the trace and check with an ohmmeter for continuity with J5 pin 12 if unsure. With a sharp knife, cut the trace.

D. Jumper U10 pin 1 to pin 13.

E. Remove and do not use the jumper plug on JPR (radio baud select).

F. Remove and do not use jumper plug on JPD (DCD select).

G. Locate the small via hole between Q3 and R26 trimmer pot. (note it may be covered by the 100K white printing). The trace from this via hole is an undocumented connection from the EPROM switch on the back panel to pin 16 on J5. It must be cut to avoid interference with the 4800 baud clock signal that the HAPN board uses on pin 16. Cut this trace in a spot that would be easy to bridge should you want to restore this feature. A wire from the via hole to J105 on the HAPN modem board allows the EPROM switch to be used to switch speed. If you choose this option, be sure to jumper JPROM (near lithium backup battery) to disable the EPROM switch function.

H. Otherwise install a switch on the front panel just above the *connect* LED. A miniature SPST switch is sufficient. Wire one terminal to J105 on the HAPN modem and the other terminal to a nearby ground hole. Grounding J105 will enable the 1200 baud speed setting (and disable the remote speed switch option). Switch open enables 4800 baud speed (and allows remote speed switch to select 1200 baud if desired).

I. Some radios may need more audio than is normally available from the Tiny-2. R11 (currently 22K) may be reduced to 2K or even 1K to substantially increase the TX audio drive available.

J. Solder a small capacitor between J2 (DIN socket to radio) pin 1 and ground. The value of the capacitor should be .001 uf as long as R11 is 22K. The RC low pass filter of this capacitor with R11 helps prevent high frequency noise from the CPU clock getting into sensitive modulator circuits. If R11 is substantially reduced, increase the value of the capacitor in proportion unless there is a shortage of audio drive. If this is the case , the radio will be much less affected by the noise anyway. Some extremely sensitive modulator circuits in radios have been known to generate noise sidebands up to 5 MHz wide without this RC low pass filter, so use it if possible.

K. Locate the double row of holes behind the DE9 connector (marked J3). Clear the solder from pin 1 ( the square pad nearest Q4) and pin 17 or 18 (holes second from the

**Continued on next page**

# Bank Switching Modification for TheNET X-1J on Tiny-2 MK-2 TNCs

*Copied from NEDA Quarterly of November 1993*

Remove the Z80 SIO chip. If you have not made the Wink&Blink modification, do it first, soldering a jumper between pins 16 & 23, and a second jumper between pins 24 & 25 on the Z80 SIO socket on the solder side of the board (the side without the parts).

Bend pins 16 & 25 of the Z80 SIO chip outwards so when you plug the chip back in, the pins end up outside the socket.

Cut a 6" piece of insulated wire (I use the wire that is sold for wire wrapping in Radio Shack), strip 1/4" insulation off one end of the wire and wrap it around pin 16 of the Z80 SIO chip (one of the pins you bend outward), solder the wire to the pin to ensure a good contact.

Route the wire under the Z80 SIO chip so it exits the at the end of the chip between pins 1&40, plug the chip back into it's socket. Route the wire through one of the DCD State Opt. connector (J10) holes to the bottom of the board (if you have a connector in J10, you will have to use a longer piece of wire and find another route. Cut and strip the end of the wire and solder to pin 1 of the EPROM socket.

Locate R41 (10K) resister, it is located near pin 1 of the EPROM on the component side of the board, unsolder the resister and discard.

The Option/Normal EPROM switch on the rear of the TNC must be in the Option (out) position. After you set the switch you can pull the button's cover out, making it harder (but not impossible) to press.

**—John, WB8PUF**

# 19.2Kbaud RS-232 for the MFJ-1270 TNC

*Copied from NEDA Quarterly of Summer/Fall 1992*

This modification allows you to use 19.2Kbaud from the TNC's RS232 port. This mod was provided by KB9AQZ, Hanspeter. The diagram is by KB8HAV, Charlie.

1 replace U3 (LM324 or LM348) with a TL074, TL084 or MC33079P
2 Cut trace from U1 pin 2 to SW2 pin 1 (300 bps position!) see ** below .
3 Wire U1 pin 10 to SW2 pin 1 , see *** below .

When you set SW2 position 1 to on, you get now 19200 bps terminal rate, instead of 300 bps. (300 bps is no longer available!)

I placed miniature relays on the printed circuit board near SW2 and J-4 to switch the radio and terminal rates between 1200-9600 for packet radio and 9600-19200 for UO-22, otherwise you must change the switches in the back every time you change the mode of operation!

pcb = pc board connections (pin 9 to 16 are shown for reference only)

sw2 > = Has the numbered switch positions for setting the baud rate.

These numbers can be seen from the rear of the tnc .

x————x = New wire that is added from sw2 pin1 to U1 pin 10 . This is the only wire added .

//////////// = Original trace that must be cut . It ran from sw2 pin1 to U1 pin 2 . This is the only trace that must be cut .

U1 is a MC14040b in my 1270 .

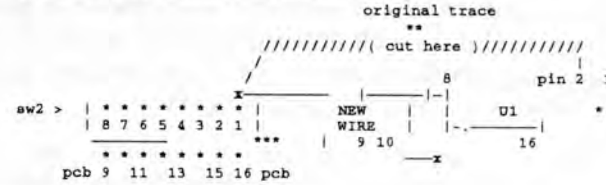
The only mod to U3 is to replace it. (no trace to cut here)

The TL084CN was only $1.50 here in Kentucky.

NOTE : This drawing is not to scale. You will find sw2 and u1 on back edge of the pc board next to 12 volt input jack. All pin positions are shown from the *top view* of the pc board and the tnc LED's positioned at your chest. All cuts and connections are made on the bottom side of pc board, keep this in mind when you are identifying the needed pins.

I would like to thank Hanspeter HB9AQZ for this mod. It has make a hugh difference in the down load bps from UO-22 at my station.

**—Charlie, KB8HAV, Louisville, KY**




**Continued from previous page**

other end) and solder single header pins into these holes. Pin 1 (the square pad) is ground and pin 17/18 is connected to pin 1 of P4 (the DE9P connector). P4 pin 1 goes high when the connect LED is on and is a suitable drive source to control the remote speed select on a TheNET node with ON/OFF sysop command. To implement this option, connect the wire from the 47K resistor on the HAPN board to J3 pin 17/18. If this option is not desired connect the remote switch wire to J3 pin 1 (ground).

L. Cut the traces between J5 pins 11 and 12, and between pins 17 and 18 on the bottom of the board.

That should complete the mods necessary to adapt the HAPN-T modem to the Tiny-2 Mk2. When you plug the modem into the modem header be sure that the pins line up properly. Tune up the modem according to the HAPN instructions and enjoy high speed packet.

**—VE2BMQ**

# Diode Matrix to NOS Wiring (as used at wb2psi)

The hexipus and other diode matrixes for connecting TheNET-style nodes have always been a source of confusion. This is an explanation of how and why it works.

The hexipus was designed to connect two Tiny-2 TNCs with straight-thru cables. The Tiny-2 was designed to connect to a PC (9-pin connector) with a straight- thru cable, just like a modem. That means the Tiny-2 is DCE: it receives data on the TXD pin, and sends data on the RXD pin - exactly the opposite of what seems right.

The hexipus is wired like a null-modem, except with diodes. That much makes sense - but you have to understand that the hexipus isn't a null-modem. It is the opposite: a null-terminal, null-computer, or null-DTE. In other words, the diode orientations are exactly the opposite of what they would be if the hexipus were a null-modem.

The other important thing to understand is that the data signals go thru the diodes in a different direction from the flow control signals. This is easiest to explain in terms of voltage levels and idle states. The purpose of the diodes is to allow a single line with non-idle state to override all the other lines with idle states.

Data lines: the idle state is a STOP bit = 1 = MARK = V- the data state is a START bit = 0 = SPACE = V+ One V+ must be able to override multiple V-

Flow lines: the idle state is V+
the flow state is V-
One V- must be able to override multiple V+

Note: on most Breakout Boxes, GREEN=SPACE and RED=MARK

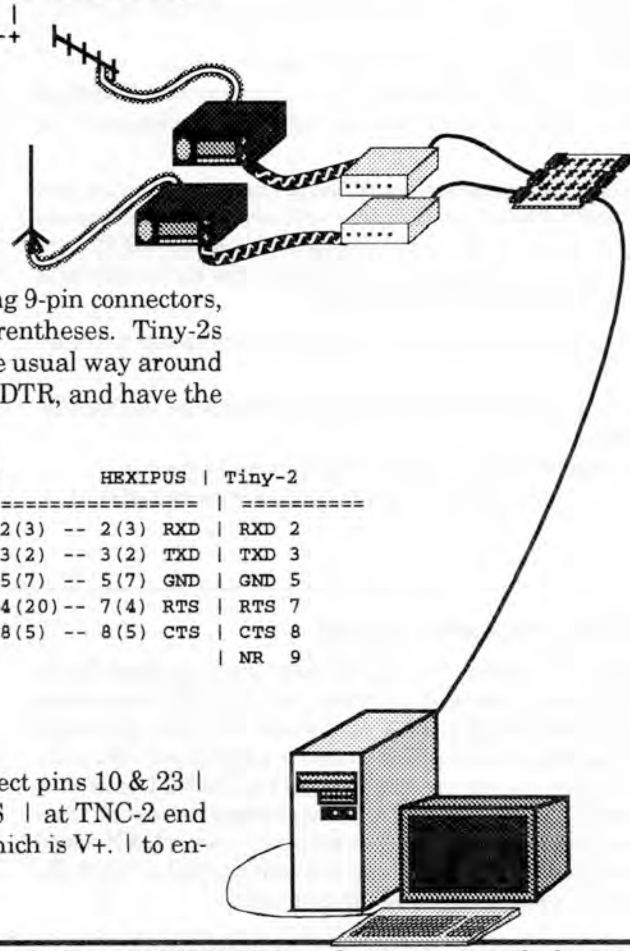
*Copied from NEDA Quarterly November 1993*

So, this is what the matrix must look like:

```
-------------------------------------------------------+
         -                |            -               |
Idle RXD --->|--   ~      | Idle RXD --->|--   +       |
         -       |--- TXD |            +       |--- TXD |
Idle RXD --->|--          | Send RXD --->|--           |
-------------------------------------------------------+
         +                |            +               |
Idle CTS ---|<--   ~      | Idle CTS ---|<--   -       |
         +       |--- RTS |            -       |--- RTS |
Idle CTS ---|<--        ^ | Send CTS ---|<--           |
            pullup|                   +-----------+
            resistor to V+            |
--------------------------------------+
```

***Symbols:***
- \+ Positive Voltage
- \- Negative Voltage
- ~ No current flow

All signals flow from left to right.
Resistor = 4.7 ohms



So, the simplified 2-port diode matrix would look like this (9 pin sense):

2 RXD—>|—TXD 3
3 TXD—|<—RXD 2
5 GND———GND 5
7 RTS—>|—CTS 8
8 CTS—|<—RTS 7

The RXD diode arrow always points to TXD
The RTS diode arrow always points to CTS

The more common cables are shown below, assuming 9-pin connectors, but showing the 25-pin connector assignments in parentheses. Tiny-2s have the pullup resistor internally, but PCs don't. The usual way around this is to solder the resistor to an unused pin such as DTR, and have the PC software assert DTR t supply the V+.

| PC NOS HEXIPUS | PC MSYS HEXIPUS | TNC-2 HEXIPUS | Tiny-2 |
|---|---|---|---|
| TXD 3(2) -- 2(3) RXD | TXD 3(2) -- 2(3) RXD | RXD 2(3) -- 2(3) RXD | RXD 2 |
| RXD 2(3) -- 3(2) TXD | RXD 2(3) -- 3(2) TXD | TXD 3(2) -- 3(2) TXD | TXD 3 |
| GND 5(7) -- 5(7) GND | GND 5(7) -- 5(7) GND | GND 5(7) -- 5(7) GND | GND 5 |
| RTS 7(4) -- 8(5) CTS | DTR 4(20)-- 8(5) CTS | DTR 4(20)-- 7(4) RTS | RTS 7 |
| CTS 8(5) -- 7(4) RTS | CTS 8(5) -- 7(4) RTS | CTS 8(5) -- 8(5) CTS | CTS 8 |
| DTR 4(20) asserted | RTS 7(4) asserted | | NR 9 |

Solder resistor at PC | Solder resistor at PC | Connect pins 10 & 23 | Connected end of cable to CTS | end of cable to CTS | at TNC-2 end of cable | straight and DTR which is V+. | and RTS which is V+. | to enable NetROM mode | through.

**—Jim Lill, wa2zkd**

# TheNET Parameters, Why 3 Hops?

*Copied from NEDA Quarterly November 1993*

Everybody in the network should be using NEDA specified parameters in the TheNET nodes or should be sending mail to the club POBox suggesting alternatives so that we can all open communications on the subject. Tell the board what should be changed in the specification that the club is publishing. This is very important if the network is to grow to interconnect the entire region.

Rumor has it that most who don't love the NEDA parameters believe that 3 hops isn't enough. Observation of non-compliant nodes seems to bear that out. There is empirical evidence and a bit of common sense that led to the idea of limiting the propagation of nodes. It has nothing at all to do with the idea of limiting users or restricting operation. The purpose of it is to make the network work better, get used more, and be more fun.

The object of the game is to tie as many packeteers together as possible and still have our system work, in real time. It is *not* the object to have as many stations listed on our nodes lists as possible. If it makes our system more efficient, or more promotable, to have fewer nodes on the nodes lists then we should have fewer nodes. One thing that is for sure, is that it is not possible to have all of the nodes on any single nodes list. A line must be drawn someplace. The current board of NEDA seems to think that 3 dedicated point to point backbone hops is a good place to do it. Then, given that 3 hops is the chosen number, the other parameters were calculated. .

**The reasons for having fewer hops of node propagation, rather than more include:**

- ***To restrict false routing***

By false routing I mean routing over the wrong path via automatic TheNET routing. Consider this example: We have a large network that has many redundant links, but is one link is bad, yet is still propagating nodes. How do you route around the bad link? You'd practically have to single step all the way around the network just to make sure that your connect went the route you are intending. Note that it is quite likely that the network of our future is exactly what this example suggests.

- ***To keep the network small enough that an individual user can observe the exceptional performance of properly configured nodes and links***

It is very important that newcomers can figure out what is good and what isn't through personal exploration. They also must be able to map their own way though the network until they get enough interest to acquire maps. Also it is very important that every amateur can figure out the network configuration, whenever they want. Having node propagation limited to fewer link hops helps this, a lot. Imagine a newcomer looking at an enormous nodes list and finding that the first bunch of nodes he/she tries don't connect, either not quickly or not ever. Does this sound familiar? Not in any network that follows NEDA standards!

- ***To improve network stability***

By having shorter node tables and shorter propagation the time-to-stability after a route failure or change is much faster. Since a node may only be broadcast 6 times (3 radio hops, 3 matrix hops) before its quality has degraded too low a change or disruption is stabilized in only 6 broadcast intervals. This is 90 minutes by the current NEDA recommendations.

- ***To increase robustness of node to node connects***

By making the steps required shorter we make the circuits more robust. If a station tries to go across many hops in one connect, the chance of failure is much higher than if a station tries to go only a few hops. We can reduce frustration by making the hops shorter. This makes the network seem more robust too.

- ***To limit the impact of wildcat disruption of the network, caused by incorrect node management or hostile intent***

A station that is running network software, either TheNET, G8BPQ, MSYS, NOS, etc.. may set parameters that interrupt service across **all of** a backbone, to as far away as the furthest node that the station's equipment has in its node's list. An example: If a station were to set its L4 time-out to 10 seconds, and its L4 retry counter to 20, then sent several packets to the furthest node it could reach, none of the packets would reach the destination and be acknowledged before the L4 time-out caused retries. Soon there would be 20 x Windowsize packets going across the backbone, just from that one stream, from that one station! There are worse things that could happen. Reducing the number of hops that two stations could do L4 traffic increases the debugability of such disruptions. It also requires more cooperation to do long distance TCP hops. I guess we'll all just have to cooperate.

- ***To limit the effective area of network software bugs***

Several times in the history of TheNET there have been bugs which caused network failures. The first was a bug where control characters in a node name would crash any node in which a Node list was requested. The bad nodename would propagate (it was originated by a defective network node) throughout the limit of the node propagation range (as defined by the Quality factors) and caused massive disruption throughout the network. Tracking it down was incredibly difficult because the

**Continued on next page →**

# Table of Node List Lengths vs Number of Hops

*Copied from NEDA Quarterly of November 1993*

This table shows the number of nodes that will be on the nodes list of a node in the middle of a network whose nodes all have the same number of visible ports and links. This is not to assume that we'll ever have a network like that. Our standards have to take into account that some nodes will be very complicated while some will not. If we look at any successful network or successful portion of a network we'll see that some nodes have four or five links while some have two or so. Most have two or three with the average being someplace in between. The average isn't what we are concerned with. We're more interested in what the worst-case situation is. Our parameter standard has to handle all circumstances without compromise. Even if you disagree with that last statement you'll want to believe that we should handle a lot more than 50% of the situations!

**Links** is how many links come out of each site.

**Visible** is how many visible nodes will be propagated from each site

**Hops** is how many hops the nodes will propagate.

**Nodes** is how many nodes will be on a nodes list.

| links | visible | hops | Nodes |
|---|---|---|---|
| 2 | 1 | 1 | 3 |
| 2 | 1 | 2 | 5 |
| 2 | 1 | 3 | 7 |
| 2 | 1 | 4 | 9 |
| 2 | 1 | 5 | 11 |
| 2 | 2 | 1 | 6 |
| 2 | 2 | 2 | 10 |
| 2 | 2 | 3 | 14 |
| 2 | 2 | 4 | 18 |
| 2 | 2 | 5 | 22 |
| 2 | 2 | 6 | 26 |
| 3 | 1 | 1 | 4 |
| 3 | 1 | 2 | 10 |
| 3 | 1 | 3 | 22 |
| 3 | 1 | 4 | 46 |
| 3 | 1 | 5 | 94 |
| 3 | 1 | 6 | 190 |
| 3 | 2 | 1 | 8 |
| 3 | 2 | 2 | 20 |
| 3 | 2 | 3 | 44 |
| 3 | 2 | 4 | 92 |
| 3 | 2 | 5 | 188 |
| 3 | 2 | 6 | 380 |
| 3 | 3 | 1 | 12 |
| 3 | 3 | 2 | 30 |
| 3 | 3 | 3 | 66 |
| 3 | 3 | 4 | 138 |
| 3 | 4 | 1 | 16 |
| 3 | 4 | 2 | 40 |
| 3 | 4 | 3 | 88 |
| 3 | 4 | 4 | 184 |
| 3 | 4 | 5 | 376 |

| links | visible | hops | Nodes |
|---|---|---|---|
| 4 | 1 | 1 | 5 |
| 4 | 1 | 2 | 17 |
| 4 | 1 | 3 | 53 |
| 4 | 1 | 4 | 161 |
| 4 | 1 | 5 | 485 |
| 4 | 1 | 6 | 1457 |
| 4 | 2 | 1 | 10 |
| 4 | 2 | 2 | 34 |
| 4 | 2 | 3 | 106 |
| 4 | 2 | 4 | 322 |
| 4 | 2 | 5 | 970 |
| 4 | 3 | 1 | 15 |
| 4 | 3 | 2 | 51 |
| 4 | 3 | 3 | 159 |
| 4 | 3 | 4 | 483 |
| 5 | 1 | 1 | 6 |
| 5 | 1 | 2 | 26 |
| 5 | 1 | 3 | 106 |
| 5 | 1 | 4 | 426 |
| 5 | 2 | 1 | 12 |
| 5 | 2 | 2 | 52 |
| 5 | 2 | 3 | 212 |
| 5 | 2 | 4 | 852 |

**Continued from previous page**

quality factors we had, up to 240 and 255, caused the node to echo across matrices virtually forever, and to echo across backbones dozens of times.

Another bug that has occurred is when a piece of network software didn't handle an error condition properly and locked up. The error condition occurred because one of the other software writers had implemented only a subset of the TheNET protocol. Only those stations that did L4 node to node connects together were in danger.

- ***To reduce the impact of emergency operations.***

In the case of an emergency operation it might be necessary to make a temporary change to create a new path between nodes, to add temporary equipment, or to block traffic flow. 90 minute *time-to-stability* that the NEDA parameters specify is very nice for both the setup of an emergency operation and for the fast recovery once the operation is discontinued.

- ***To keep nodes lists to a reasonable size i.e. < 100***

Large nodes lists baffle users, reduce manageability and tie up network resources. If we are successful in promoting an open technology that any ham can participate in and expand on, then we will necessarily have many many nodes and many many links. In other networks that have used this design philosophy there are many nodes that have many links. Our survival depends on our adopting standards that we can stick to for a long time. If we make assumptions about the scarcity of links and services in our network we are doomed to failure. By adopting parameters for 3 backbone hops maximum propagation we are planning for a network that can grow, without needing exceptions at every turn.

**Continued on next column→**

# Program for Calculating Node List Size

*Copied from NEDA Quarterly of November 1993*

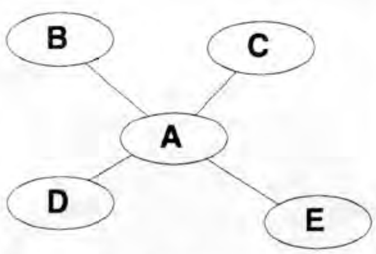
This program was written on an HP handheld calculator. It assumes a homogeneous network where all of the nodes are the same. This is for simplicity. The object of the program is to show how the node list will look during worst case conditions.

Register L = # of Links from each site

Register V = # of Visible nodenames (non pound nodes)

Register H = # of Hops that each node listing will propagate.

The program simply looks for one less than the number of hops. The last hop is a special case. Each hop generates V + PrevHop*(Links-1) The reason that we use links minus 1 is that one of the links from the node being considered comes from the node that we're broadcasting to. Example: If node A has four links and is broadcasting to node E, node E doesn't care how many nodes A has that E told it about. Thus we only count the nodes that node A heard from B, C, and D. The reason for the end-condition is that the last node wants to have a count of all of the nodes coming from all of its links.




```
LBL A   start of program

STP     ask for # of links
STO L   save # of links
STP     ask for # of Visible TNCs
STO V
STP     ask for # of Hops broadcasts will go
STO H

—— start
RCL V   get visible TNCs at first node
STO U   Save in temp register                    Reg U is PrevHop
RCL H   get # of hops
1
-       one less because we handle end conditions separately
STO T   save temp register.                      Reg T is loop counter

—— inner loop
LBL B
RCL U   get total # of nodes we hear about over each link
RCL L   get total # of links
1
-       Xreg now has # of links we're hearing from that we'll talk about
*       Xreg now has # of nodes we're hearing about, from links we care about
RCL V   get # of nodes at THIS site
+
STO U   save in temp register

RCL T   get loop counter register
1
-       decrement loop counter
STO T
X>0?    are we ready for finish up
GTO B   no? Then loop again

—— finish up
RCL U   get total # of nodes we hear about over each link
RCL L   get total # of links
*
RCL V   get # of nodes that I have
+
GTO A   back to start
```

**←Continued from previous column**

Consider the table on the previous page. We'd have to have a pretty simple network to not run into trouble with five hops. It is unlikely that we'll run into problems with 3 hops. With four hops we are in danger of running into trouble in 30% of the network.

***The rest of the numbers:***

The parameters that depend directly on the idea of having three backbone hops as a limit of node propagation are:

Minimum Quality For Auto Update
HDLC Channel Quality
RS-232 Channel Quality

Time To Live Initializer - number of hops before a message between nodes will give up. This saves us from wasting network time due to routing loops.

The parameters that may be tailored for best performance after knowing the performance of the backbone links and the number of hops a station connect are:

Transport Time-out
Transport Busy Delay

The more hops you have the longer your time-outs will have to be.

All of these parameters are explained in laborious detail in the Annual. Please read the descriptions and get in touch. We can change things and improve explanations. Your proposals will be printed in the Quarterly if you want.

Please work with the club ..and..

Failing to plan, is planning to fail.

**—N0NDO, WA2WNI and KA2DEW**

# Multi-way Wireline Node/Switch Coupler

*Copied from NEDA Quarterly for July 1993*

The NX2P wireline adapter allows two or more co-located TNCs to communicate with each other via their radio ports at high speed, by bypassing their modems. This is useful for connecting one or more user station directly to a node or switch TNC without going over the radio. This may also be used to couple multiple diode matrix boards together or to tie a BBS (that uses a KISS TNC for instance) into a matrix, via a dedicated node or switch TNC on the matrix. This wireline adapter method will work with 2, 3 or even up to 6 TNCs. You could have a BBS using KISS, a NOS node, using KISS, a user TNC and three diode matrices, all wired together and all running at wireline speeds, using very few components. This far outshines the circuit that KA2DEW came up with in 1989.

Construct the circuit shown in the diagram.. It is connected to the 20 pin Modem disconnect header inside the TNC. The transistor can be just about any PNP type, the resistor is a 1/4 watt with the value not critical, and the diode is a plain-vanilla 1N914 or equivalent. Build the circuit either on a 20 position modem disconnect socket or using a 10-pin SIP socket. There's not much room (about 1/2") between the PC board and the TNC's case so the transistor will have to be bent over.

The three wires (the third is TNC ground, available from pin 15 of the modem disconnect header in the Tiny-2) should be long enough to exit the TNC case via the TTL connector hole. Put the connector of your choice on the end, so you can remove the TNC from the wireline if necessary.

The wireline itself is simple: just connect all of the wires from pin 17 together, all of the wires from pin 1 together, and all of the grounds together. Naturally, you'll use a connector that mates with whatever you used on the TNC.

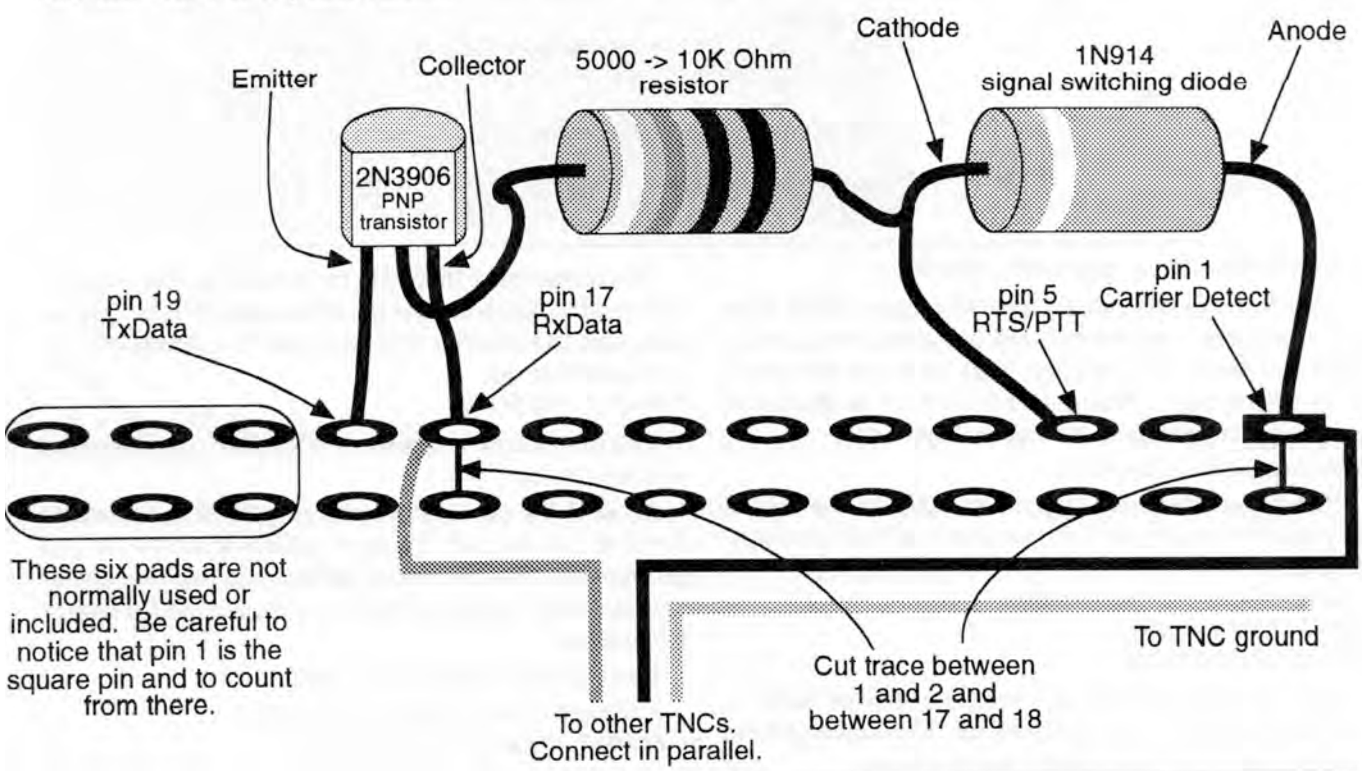
***The way this works:***

Pin 17 is RxData and Pin 19 is TxData. The TNC is cranking out flags (on/off/on/off...) on the TxData line all of the time. The transistor, when off, keeps the TxData from getting to the RxData. Pin 1 is carrier detect. Normally it floats high, around 5 volts. Pin 5 is PTT and it normally is also at 5v. When the TNC goes to transmit it pulls PTT low, dragging it's own pin 1 low through the diode, and thus dragging all of the other TNCs' pin 1 low. This TNC's PTT light will turn on, as well as all of the DCD lights. Since pin 5 is low, only on this TNC, the base of the transistor is pulled down, through the resistor, opening a path for TxData to get out to pin 17 which is then connected to all of the other TNCs. (actually it's magic, we made all this up).

**—Article by N2IRZ and KA2DEW.**
**Idea and design by NX2P.**
**First published in "By Any Other Name", the magazine of the Radio Amateur Telecommunications Society (RATS)**

*Board kits with all parts, including plugs which attach to the top of an existing modem disconnect header (included with TNC or you supply) are available for $3.00 per TNC and $1.00 for a gizmo that serves as the common junction point between up to 6 TNCs.*

*Amateur Networking Supply*
*POBox 219*
*Montvale NJ 07645*



# HROUTE Forwarding -- An Alternate Approach

*Copied from NEDA Quarterly for November 1993*

### *Purpose*

The purpose of this paper is to propose a different approach to BBS personal mail forwarding using the six character *grid square* locator.

### *Background*

This method was developed to solve an ongoing problem encountered while forwarding using the R header within NY State.

### *History*

Since the beginning of message forwarding, political boundaries have been used. Forwarding by R header became prevalent in the late 1980's the accepted format being callsign.XX.YYY.ZZ where XX is the two character state code, YYY is the three character country code and ZZ the two character continent code, giving a string length of 10 (including the periods). Parsing could be done on either callsign, state, country or continent, in that order. So, the typical R string would be:

KA2MSL.NY.USA.NA

As the number of bulletin board systems grew, it was found that the state designator was too large to be useful and so an *area* field was added to the string in the format #WWWW. This field would contain a geographical location within the state and be used for intrastate forwarding. A typical format could be #MHV (Mid Hudson Valley). This describes the area of NY state north of the Westchester/Rockland counties and south of the greater Capitol District. This adds another five to six characters to the string. The string now is:

KA2MSL.#MHV.NY.USA.NA

There were now five sub strings to use to determine location. This has become an accepted standard by many BBS systems in the United States. There are those sysops who felt that the two character continent field was inadequate and expanded it to 4 characters. i.e. NOAM vs NA for North America. This could give us a 25 character string.

While useful in routing, this added information does increase the overhead in sending a message or bulletin over a long distance. Multiply this by the number of stations that handle this message and this will add kilobytes to the total send length. In fact, the R header information exceeds in size the total length of the message in many instances.

### A Different Approach

For years, contesters have been using the *grid square* as an exchange for location of the station worked. A grid square is a 2° x 1° box based upon the latitude and longitude of a station. This information is not only used to determine location but is also widely used to determine distance and beam heading between two points. The format is AAxx (two letters two numbers). i.e. FN21

Because of the size of the 2° x 1° box is quite large in the temperate zone, the box has been further divided into 5 x 2.5 sub squares in the format aa-zz. So, the format becomes AAxxBB I.E.: My QTH is FN21XM. This gives a very precise location of a station. In fact the remote sysop for my system lives only a few miles away and is in the FN21XN box.

My proposal is to use the callsign and grid square as the R header information in the format:

Callsign.AAxxBB a maximum of 13 characters. This provides each BBS with an identification string which not only provides location, but also direction from the originating BBS

Since most BBS systems allow wild cards and exceptions, parsing becomes a simple matter on a local and regional basis. Forwarding by callsign is unaffected.

*Wotan*

### *Implementation*

To avoid confusion and make the transition easier, the implementation of this plan might be in three Phases.

### *Phase One*

Replace the #WWWW regional subdesignator with the 6 Character Grid square.

Pre Phase 1 KA2MSL.#MHV.NY.USA.NA
Post Phase 1 KA2MSL.#FN21XM.NY.USA.NA

### *Phase Two:*

Following a sufficient period of time, the State designator could be removed along with the leading #, yielding the following:

KA2MSL.FN21XM.USA.NA

### *Phase Three*

The final phase could occur once this plan is implemented on a global basis, and the Country and Continental designators become redundant. We would arrive at the final form:

KA2MSL.FN21XM

### Conclusion:

Using an existing standard we can:

- Define the Global location of an individual BBS
- Reduce the size of R Headers and forwarding overhead
- Simplify forwarding scripts by eliminating political (state) boundaries.

This method is being tested in the upper NY state area. So as not to upset those stations not participating in the study, the grid locator info was inserted before the regional sub string and preceded by a .# and followed by a period.

KA2MSL.#FN21XM.#MHV.NY.USA.NA

It was found that two 'Regional strings' caused some concern on the part of several sysops, therefore those of us in the test group have substituted the Grid Locator for the regional string, effectively implementing Phase One, on a test basis.

**—Mike, KA2MSL**

*[With the release of FBB v5.15b there was a limitation of six characters in each H-address sub-field. This required the removal of the # character preceeding the grid locator - ed]*

# NEDA Constitution

## 1. Purpose of this Article

a. This article lays down the rules for operation of the North East Digital Association. No other N.E.D.A. document may change or replace the rules set down in the Constitution. The Constitution may only be modified by the procedures described herein.

## 2. Officers

a. There are six Board of Directors positions plus appointments and alternates. The board of directors are elected for two year terms. Three of the directors are elected annually.

## 3. Appointments

a. Appointed positions include Treasurer, Chairman of the General Meeting, Membership Director, Board Member Alternates, Chairman of the Technical Committee and Network Regional Sysops. The Network Regional Sysops report to the Chairman of the Technical Committee and are considered members of the Technical Committee.

b. Other appointments may be made at the direction of the board of directors. These appointments are made by the board of directors. Only voting members may be appointed to a committee chairmanship, board member alternate or office position. Board members may also serve other appointed positions and appointees may serve multiple appointments.

## 4. Board Member Alternates

a. Each board member may appoint an alternate to represent him or her at board meetings in the event that the board member is unable to attend.

b. The alternate must be approved in advance by the board during a board meeting in which the board member presenting the candidate for alternate is present. The candidate must also be present and agree, or furnish written consent to serve.

c. Appointment of an Alternate may be terminated at any board meeting under any one of the following conditions:

- At the request of the board member the alternate represents.
- At the request of the alternate.
- Using the same procedures as removal of a board member, (Article 5).

d. The alternate appointment is automatically cancelled when:

- The alternate is elected to a board position;
- The alternate is no longer a voting member;
- The member the alternate represents is no longer on the board.

e. The alternate has full voting rights at board meetings in the absence of the aboard member which he or she represents.

f. It is the responsibility of the board member and his or her alternate to maintain reasonable communication so that the alternate my act on behalf of the board member in an informed manner.

g. Any alternate may act on behalf of any absent board member, who's alternate is also absent, if necessary to provide a quorum. The member he or she is originally designated to represent must also be present. The alternate would have the same voting rights as in (e.) above for the member he or she is representing at the meeting.

## 5. Removal of a Person From Office or Revocation of Membership Privileges

a. A petition for removal of a person from office or membership must be submitted in writing to the board of directors with a minimum of four signatures of voting members. The petition must be presented at least two weeks before a quarterly board meeting in which it is to be acted upon. The board of directors must vote on the petition at a quarterly board meeting. The document will be kept in the club archives unless removed and expunged at a later board meeting.

b. This person being removed is held as a removal-pending member for one quarter and then is reviewed at the following quarterly board meeting. This issue is then presented in the minutes in the Quarterly so that it may be reviewed by all the membership and commented on before the following quarterly board meeting.

c. A person removed from membership is not eligible for voting membership unless the privilege is restored by an act of the board of directors at a later date.

## 6. Membership

a. Membership is open to all. Dues are at least 2 levels for individuals. One of these levels is called Voting Membership. Voting membership is open to all except as defined under 'Removal' above.

## 7. Dues

a. Dues are paid to the Membership Director or his designee who then forwards the funds to the Treasurer. Dollar values of dues is set in the NEDA bylaws but the dues level for a Voting member is $25 or greater. Dues are used to fund:

- operating expenses for the club;
- development costs for club products that facilitate network growth.
- documentation in the form of an Annual and Quarterly
- documentation in the form of free technical documentation distributed for the benefit of packet networking.
- documentation in the form of free promotional literature on NEDA and on packet networking.

## 8. Membership Privileges

a. Voting Members receive the 4 copies of the NEDA Quarterly per year and a copy of the Annual each year. The Annual is delivered to the member at renewal time (after renewal) or at the anniversary of the member's membership.

b. Voting members are invited to attend the Board of Directors meetings, run for office annually and vote for officers by mailed ballot.

c. Additional privileges are defined in the bylaws.

## 9. Board Meetings

a. A Board of Directors Meeting is a physical gathering of the board members.

b. A minimum of 4 directors must be present to open a board meeting. The board meetings must be announced via the NEDA Quarterly or via packet mail to every voting member at least two weeks before the meeting. If a quorum of board members is not available to start the meeting a new meeting must be scheduled and new announcements must be sent.

c. Board meetings must be held in different cities each time to make it possible for all voting members to have equal access to the proceedings of the board of directors.

d. Board meetings may be attended by voting members or those given special dispensation by the board of directors or any approved by the bylaws.

e. Board meetings must be held 4 times per year. The 4 quarterly board meetings are held as close as possible to the first week of January, April, July and October. Additional board meetings may be called by the board of directors with a vote of 4 board members. A board meeting is required in order to:

- spend club funds.
- discipline a member;
- change the appointment for network sysop or chairman of any committee.
- re-assignment or assignment of a board member alternate;
- change the constitution or bylaws
- appoint the chairman of the annual meeting or change that appointment.
- form or disband any committee.

f. Actions which must occur at the board meeting include the reading of a current NEDA treasury report. This will be recorded in the minutes and printed in the subsequent NEDA Quarterly.

## 10 <removed>

## 11. Elections

a. Elections are held by mailed ballot after the October Board of Directors Meeting. Immediately after the October Board of Directors Meeting attendance of each member, over the previous year's board meetings, are tallied. Any voting member who is paid up for two years from the end of October of the current year, who has attended half of the year's board meetings, and who are not already in the middle of a two year term are automatically nominated and are listed on the ballot.

b. This ballot is sent to all NEDA voting members complete with a self addressed stampled envelope. The envelope also has a return address label with a note stating that the return address must be filled in for the ballot to be counted. The ballot includes instructions that the voting member should order all of the listed people in ascending order, 1 for first choice, 2 for second choice. This way the results will still be meaningful if one or more nominated members are unavailable to fill the positions. The ballots are mailed to the club POBox and then counted by one of the board members whose term is not expiring this year. The balloting process, and the counting process must be operated with a process which maintains confidentiality of the ballots.

c. The ballots must be mailed out to all NEDA voting members within two weeks of the board meeting. They must be returned to the club POBox within five weeks of the board meeting. Results are included in the Quarterly or are mailed out separately to all members to arrive at least a week before the winter board meeting.

d. The results include the following statistics:

- total number of ballots sent;
- total number of ballots returned.:
- list of all nominees;
- list of the three new board members;
- and a list of nominees who abstained but who had a higher vote than the selected board members.

e. If three new board members are not chosen by this process then a board member may be chosen by consensus of the founders and the existing board from those voting members who were previously board members and who ended their term as board member in good standing. If there still are not three eligible new board members then the club must be dissolved.

## 12. Board Member Responsibilities

a. Board members or their alternates must attend the quarterly board meetings or obtain an alternate to handle meetings the board member cannot attend. Failing to do so twice in a single year is grounds for removal from office. Board members or their alternates are also obligated to attend additional board meetings called by verbal agreement by any four of the board members.

b. Board members represent NEDA and are obligated to carry out the NEDA Charter in regards to dealings with other members and non-members.

c. The board of directors as a body are responsible for seeing that the NEDA Quarterly and the NEDA Annual are published on time. As these are the instruments of the club and as the NEDA Quarterly is the means by which the financial operations of the club are published to the membership, the paying membership has the right to expect these documents.

## 13. Filling Spots on the Board Due to Board Member Resignation

a. If a board member resigns or is otherwise no longer available to fulfill the remainder of his or her term a new board member is selected to serve until the next annual meeting. The new board member is selected from those voting members who were previously board members and who ended their term as board member in good standing.

## 14. Network Maps

a. Network maps must be maintained and are presented in the Quarterly. The maps must consist of at least the callsign, nodename, location (at least relative), user access frequencies for AX.25 (if any) and backbone connectivity for all NEDA network nodes.

## 15. NEDA Quarterly

a. The NEDA Quarterly is published within 60 days after the quarterly board meeting. The Quarterly is fully described in the bylaws but as a minimum must include the minutes of the board meeting (including the treasurer's report), the network maps, and membership roster.

b. The board may delegate the task of production and mailing of the Quarterly but maintain the responsibility.

c. In the Fall edition of the Quarterly whatever results that are available from the annual elections are printed. This may include the nominees or the final results.

## 16. NEDA Annual

a. The NEDA Annual is the current statement of NEDA packet network involvement. This includes user information for usage of the NEDA network as well as lessons in the technology needed to fulfill the goals of NEDA as stated in the charter.

b. This document is delivered annually to each and every paid member of the club. This document should be updated at least once annually to reflect the current state of networking technology in use by NEDA.

c. The Annual is the responsibility of all of the board members. The board may delegate the task of production and mailing of the Annual but maintain the responsibility.

## 17. Changes to the Constitution

a. Changes to the Constitution may only be made by the following process:

b. At a regularly scheduled quarterly Board of Directors meeting a proposal for a change is submitted in printed or typed form (8 copies) to each of the Directors, to the editor and to the secretary. The item must be presented in person by a NEDA voting member.

c. The format of the submission is in bulleted sections. The following sections must be included: TITLE, PRESENTED, BY, BRIEF, SPECIFICS, PURPOSE. The page is headed with "Constitutional Change Request". TOPIC is followed by one line which identifies the change request. PRESENTED is followed by the date of the board meeting. BY is followed by the name and callsign of the author. BRIEF is followed by a single paragraph description of the change. SPECIFICS is followed by a paragraph by paragraph description of the changes including reference section and paragraph numbers. PURPOSE is followed by a justification for the change. A sample change is available from the club.

d. The proposed change is entered into the minutes of the Board of Directors meeting at which it is presented. Discussion may follow. No vote is taken at this time.

e. At the following board meeting the change is brought up as old business and after discussion is either ratified or not. No change is made if a tie occurs.

f. If a change is ratified then the new copy of the Constitution is printed in the following Quarterly in its entirety.

## 18. Changes to Bylaws

a. Changes to the bylaws may be made at a single board meeting with the vote of a majority of the board members present. If a tie occurs then no action is taken.

## 19. Grounds for Dissolution

a. If the board of directors doesn't hold 4 board meetings during the year or if the club is unable to hold elections or there were not three eligible and willing candidates or if the Quarterly in at least it's minimum form) isn't delivered on time then the club must be dissolved.

## 20. Dissolution of the Club

a. After paying out any pending bills the treasurer is directed to write a check for the remainder of the club treasury to the American Cancer Society and to close the all club bank accounts. The name of the club (i.e. North East Digital Association) and it's logo NEDA become the property of the founders of the club, WA2WNI, WA1TPP, KA2DEW, K1MEA, NQ1C, WA2VAM, KC3BQ, to do with as they wish. All paperwork pertaining to software management of individual nodes is delivered to the node/site managers.

*This page left blank so as to not mess up the paperwork at NEDA's office. This has been a complaint in the past. Please have your 4 year old daughter (or son) scribble all over this side of the page just to keep John's (the NEDA membership director) life interesting. If you don't have a 4 year old daughter (or son), ask your neighbor (they'll lend you one). Thanks.*

Name, age

# North East Digital Association Membership Application

Welcome to NEDA and Packet Radio. This is the official membership form for NEDA.

**Some general information about NEDA:**

NEDA is a club formed in 1989 to promote packet radio and to lead the development of a general purpose, user accessible wide area packet radio network.

NEDA's area of interest includes the north eastern United States, Quebec, Ontario and the Maritime Provinces..

NEDA publishes the *NEDA Quarterly* as a periodical four times a year. The *NEDA Annual* is published each year as well. Associate w/Quarterly and Voting Members receive these documents by mail. The club distributes the magazine at flea markets as well.

NEDA's administration is based on 6 directors, 6 director alternates and several appointees. The six directors of the board are elected by the membership for two year terms. The six alternates are appointed by the elected board. Three of the directors are elected each year. The appointees include recording secretary, membership secretary, treasurer, editor, office manager, mapmaker. The board meets four times a year in various locations within the club's area of interest. Those meetings are open to the voting membership and are fully documented in the minutes which are published in *The NEDA Quarterly.*

NEDA members sponsor general interest and specific interest packet meetings throughout the region of interest of the club. Those meetings may be announced in *The NEDA Quarterly* and meeting results may also be published. Other packet radio clubs can request space in *The NEDA Quarterly*. NEDA's focus is to publish information on packet radio and packet radio networking.

**The membership/dues structure of NEDA is as follows:**

| | |
|---|---|
| Associate/Subscription with US address | $15 |
| Voting with US address | $25 |
| Associate/Subscription with Canadian address | $20 |
| Voting with Canadian address | $30 |
| Associate/Subscription with non-Canada/non-US addr. | $30 |
| Voting with non-Canadian/non-US address | $40 |
| Upgrade to voting, all countries | $10 |

All membership rates are US funds only. Canadian applicants should send funds in a Postal or Bank Money-Order in US funds. Thank you. This form is dated Q51-021894.

**Name:** ____

**Address:** ____

**City:** ____ **State or Province:** ____

**Country:** ____ **postal code:** ____

**Membership desired - Check one box:**

| | |
|---|---|
| US Associate/Subscription Membership, $15/year ☐ | US Full/Voting Membership, $25/year ☐ |
| Canada Associate/Subscription Membership. $20/year ☐ | Canada Full/Voting Membership $30/year ☐ |
| Non-US/Can Associate/Sub. Membership. $30/year ☐ | Non-US/Canada Full/Voting $40/year ☐ |
| Upgrade to Full/Voting Membership, $10/year ☐ | Information Correction Only ☐ |

Check here if this is a RENEWAL or if you've ever been a NEDA member before. ☐

Enter # of years you wish to pay for: ____ Amount Enclosed: ____

**Callsign:** **Full Service BBS at which you get your packet mail**

@ .# .

Example: N3EIC @ KA2EIA.#EPA.pa

**Home Phone:**

( )

**Other Computer addresses we can contact you at: (TCP/IP, FIDO Net, Internet, CompuServe etc.)**

If a NEDA member gave you this form, what is his or her callsign? ____

**County:** ____

**Make Checks to NEDA.**
**Address this form and all other correspondence to:**
**NEDA**
**PO Box 563**
**Manchester NH 03105**

| NEDA representative | Office Manager | Membership Secretary |
|---|---|---|
| Receiving Officer: | Date stamp: | Saved in DB: |
| Amount: | Amount: | |
| Intro Package delivered: ☐ | Check #: | Intro Package Mailed: |
| Date: | Tracking #: | |
| Check #: | | |

# Sysop's Help Sheet for TheNET Plus 2.08/2.10

### *TN2.08*

| PARMS # | Function | user port | #DPPL | #HTSF Multiway | gateway | non ideal |
|---|---|---|---|---|---|---|
| 1 | Minimum auto update quality | 50 | 50 | 50 | 50 | 50 |
| 2 | HDLC (radio port) default quality | 0 | 0, A | 0, A | B | 0 |
| 3 | RS-232 (crosslink) default quality | 203 | 203 | 203 | 203 | 203 |
| 4 | Initial obsolescence count | 5 | 5 | 5 | 5 | 5 |
| 5 | Min Obs to broadcast | 6 | 3 | 2 | 255 | 3 |
| 6 | Nodes broadcast interval | 900 | 900 | 900 | 900 | 900 |
| 7 | FRACK (T1) time | 4 | 1 | 1 | D | 9 |
| 8 | AX.25 windowsize (MAXframe) | 1 | 1 | 1 | 1 | 1 |
| 9 | AX.25 retries | 10 | 10 | 10 | 10 | 10 |
| 10 | Digipeat enable | 0 | 0 | 0 | 0 | 0 |
| 11 | Callsign validation | 1 | 1 | 1 | 0 | 1 |
| 12 | Host Mode Connects (not in X-1J) | 0 | 0 | 0 | 0 | 0 |
| 13 | Transmit keyup delay | 35 | E | E | E | 35 |
| 14 | Selective nodes broadcast on ports: | 2 | 3 | 3 | 3 | 2 |
| 15 | Hash (#) node broadcast port control | 1 | 1 | 1 | 1 | 1 |
| 16 | Connect command enable | 1 | 0 | 0 | 1 | 1 |
| **EPROM Parameters** | | | | | | |
| 17 | Size of destination node table | 100 | 100 | 100 | 100 | 100 |
| 18 | Initial time-to-live | 13 | 13 | 13 | 30 | 13 |
| 19 | Transport FRACK timeout (seconds) | 200 | 200 | 200 | 600 | 200 |
| 20 | Transport RETRY counter | 1 | 1 | 1 | 1 | 1 |
| 21 | Transport acknowledgement delay | 1 | 1 | 1 | 1 | 1 |
| 22 | Transport busy delay | 180 | 180 | 180 | 450 | 180 |
| 23 | Transport window size | 2 | 2 | 2 | 2 | 2 |
| 24 | Transport overfill limit (frames) | 4 | 4 | 4 | 4 | 4 |
| 25 | No-Activity time-out (seconds) | 7200 | 7200 | 7200 | 7200 | 7200 |
| 26 | Persistence (n/256) | 255 | 255 | C | C | C |
| 27 | Slot time | 1 | 1 | C | C | C |
| 28 | ACK (T2) time (link layer RESPTIME) | 50 | 20 | 20 | 50 | 50 |
| 29 | Active check (T3) | 32000 | 32000 | 32000 | 32000 | 32000 |
| 30 | Beacon mode control | 2 | 0 | 0 | 1 | 1 |
| 31 | CQ broadcasts | 1 | 0 | 0 | 0 | 1 |
| 32 | Heard List Length (not an X-1J parm) | 20 | 20 | 20 | 20 | 20 |
| 33 | Full duplex | 0 | 0 | 0 | 0 | 0 |

### *TN2.10*

| PARMS # | Function | user port | #DPPL | #HTSF Multiway | gateway | non ideal |
|---|---|---|---|---|---|---|
| 1 | Minimum auto update quality | 50 | 50 | 50 | 50 | 50 |
| 2 | HDLC (radio port) default quality | 0 | 0, A | 0, A | B | 0 |
| 3 | RS-232 (crosslink) default quality | 203 | 203 | 203 | 203 | 203 |
| 4 | Initial obsolescence count | 5 | 5 | 5 | 5 | 5 |
| 5 | Min Obs to broadcast | 6 | 3 | 2 | 255 | 3 |
| 6 | Nodes broadcast interval | 900 | 900 | 900 | 900 | 900 |
| 7 | FRACK (T1) time | 4 | 1 | 1 | D | 9 |
| 8 | AX.25 windowsize (MAXframe) | 1 | 1 | 1 | 1 | 1 |
| 9 | AX.25 retries | 10 | 10 | 10 | 10 | 10 |
| 10 | Callsign validation | 1 | 1 | 1 | 0 | 1 |
| 11 | Host Mode Connects (not in X-1J) | 0 | 0 | 0 | 0 | 0 |
| 12 | Transmit keyup delay | 35 | E | E | E | 35 |
| 13 | Selective nodes broadcast on ports: | 2 | 3 | 3 | 3 | 2 |
| 14 | Hash (#) node broadcast port control | 1 | 1 | 1 | 1 | 1 |
| 15 | Connect command enable | 1 | 0 | 0 | 1 | 1 |
| 16 | Size of destination node table | 100 | 100 | 100 | 100 | 100 |
| 17 | Initial time-to-live | 13 | 13 | 13 | 30 | 13 |
| 18 | Transport FRACK timeout (seconds) | 200 | 200 | 200 | 600 | 200 |
| 19 | Transport RETRY counter | 1 | 1 | 1 | 1 | 1 |
| 20 | Transport acknowledgement delay | 1 | 1 | 1 | 1 | 1 |
| 21 | Transport busy delay | 180 | 180 | 180 | 450 | 180 |
| 22 | Transport window size | 2 | 2 | 2 | 2 | 2 |
| 23 | Transport overfill limit (frames) | 4 | 4 | 4 | 4 | 4 |
| **EPROM Parameters** | | | | | | |
| 24 | No-Activity time-out (seconds) | 7200 | 7200 | 7200 | 7200 | 7200 |
| 25 | Persistence (n/256) | 255 | 255 | C | C | C |
| 26 | Slot time | 1 | 1 | C | C | C |
| 27 | ACK (T2) time (link layer RESPTIME) | 50 | 20 | 20 | 50 | 50 |
| 28 | Active check (T3) | 32000 | 32000 | 32000 | 32000 | 32000 |
| 29 | Beacon mode control | 2 | 0 | 0 | 1 | 1 |
| 30 | CQ broadcasts | 1 | 0 | 0 | 0 | 1 |
| 31 | Full duplex | 0 | 0 | 0 | 0 | 0 |
| 32 | Telemetry Unit Only | | | | | |
| 33 | Telemetry Unit Only | | | | | |

**Note: NEDA recommends TheNET X-1J as of Feb 1994**
**See page 174 for definitions.**

### *Notes*

**A.** HDLC default quality for Dedicated Point to Point Links and HTS free multiway backbones. This value is burned into the EPROM at quality 0 and left at 0 in the Parms but the *routes* to the neighbors are locked in at quality 203.
*R 0 callsign + 203*

**B.** HDLC default quality for gateway ports This value must be determined by the sysop such that the gateway port can fit into the neighboring network as well as being accessible to the NEDA-compliant network. 82 is a good value to burn into the EPROM. No matter what you have the HDLC quality set to the nodes that the gateway hears over the radio will not be broadcast over the RS-232 port because the minimum obsolescence to broadcast (PARM /6) is set to higher than the initial obsolescence. Do not lock in any nodes at > 50 quality.

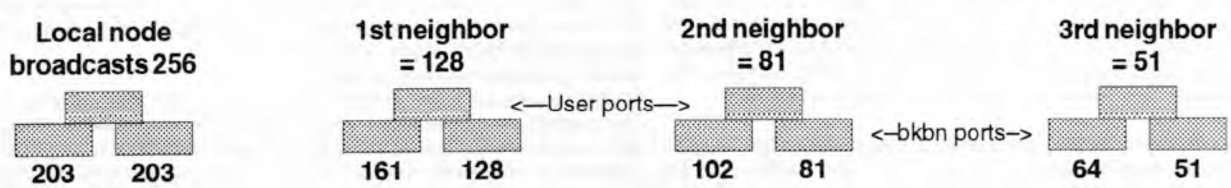
**C.** Persistance and slot time for Hidden Transmitter Free, Gateway and Non-Ideal This value must be calculated based on the intended agressiveness of the port and the number of transmitters on the frequency. For HTS Free multiway backbones this should be equal to 256/(N-1).

**D.** FRACK time for a gateway node. This needs to correspond to whatever standards are being used by the rest of the stations on the radio channel. See *Modes & Parameters.*

**E.** Transmit keyup delay for Dedicated point to point link, HTS free backbone, and gateway. This number should be imperically tested. It should be set to 1.25 times the lowest value that works.

**F.** Connect redirector
This parameter has no real affect on the network. Read the Modes & Parameters

***This drawing represents the node quality value for a single node as it propagates through several node hops.***



# NEDA TheNET Node Spec

**This is the Parameter and Mode Specifications for a NEDA Compliant Node - Feb 1994**

| X-1J PARMS # | Function | user port | #DPPL | #HTSF Multiway | gateway | non ideal |
|---|---|---|---|---|---|---|
| 1 | Size of destination node table | 100 | 100 | 100 | 100 | 100 |
| 2 | Minimum auto update quality | 50 | 50 | 50 | 50 | 50 |
| 3 | HDLC (radio port) default quality | 0 | 0, A | 0, A | B | 0 |
| 4 | RS-232 (crosslink) default quality | 203 | 203 | 203 | 203 | 203 |
| 5 | Initial obsolescence count | 5 | 5 | 5 | 5 | 5 |
| 6 | Min Obs to broadcast | 6 | 3 | 2 | 255 | 3 |
| 7 | Nodes broadcast interval | 900 | 900 | 900 | 900 | 900 |
| 8 | Initial time-to-live | 13 | 13 | 13 | 30 | 13 |
| 9 | Transport FRACK timeout (seconds) | 200 | 200 | 200 | 600 | 200 |
| 10 | Transport RETRY counter | 1 | 1 | 1 | 1 | 1 |
| 11 | Transport acknowledgement delay | 1 | 1 | 1 | 1 | 1 |
| 12 | Transport busy delay | 180 | 180 | 180 | 450 | 180 |
| 13 | Transport window size | 2 | 2 | 2 | 2 | 2 |
| 14 | Transport overfill limit (frames) | 4 | 4 | 4 | 4 | 4 |
| 15 | No-Activity time-out (seconds) | 7200 | 7200 | 7200 | 7200 | 7200 |
| 16 | Persistence (n/256) | 255 | 255 | C | C | C |
| 17 | Slot time | 1 | 1 | C | C | C |
| 18 | FRACK (T1) time | 4 | 1 | 1 | D | 9 |
| 19 | AX.25 windowsize (MAXframe) | 1 | 1 | 1 | 1 | 1 |
| 20 | AX.25 retries | 10 | 10 | 10 | 10 | 10 |
| 21 | ACK (T2) time (link layer RESPTIME) | 50 | 20 | 20 | 50 | 50 |
| 22 | Active check (T3) | 32000 | 32000 | 32000 | 32000 | 32000 |
| 23 | Digipeat | 0 | 0 | 0 | 0 | 0 |
| 24 | Callsign validation | 1 | 1 | 1 | 0 | 1 |
| 25 | Beacon mode control | 2 | 0 | 0 | 1 | 1 |
| 26 | CQ broadcasts | 1 | 0 | 0 | 0 | 1 |
| **MODES** | | | | | | |
| *#* | *Function* | | | | | |
| 1 | RS-232 host mode | 0 | 0 | 0 | 0 | 0 |
| 2 | CWID repeat period ( seconds ) | 0 | 0 | 0 | 0 | 0 |
| 3 | CWID keyer speed | 6 | 6 | 6 | 6 | 6 |
| 4 | Selective nodes broadcast on ports: | 2 | 3 | 3 | 3 | 2 |
| 5 | RS-232 crosslink protocol | 0 | 0 | 0 | 0 | 0 |
| 6 | Transmit keyup delay | 35 | E | E | E | 35 |
| 7 | Full duplex | 0 | 0 | 0 | 0 | 0 |
| 8 | Crosslink port node b'cast interval | 450 | 450 | 450 | 450 | 450 |
| 9 | Node broadcast algorithm port control: | 3 | 3 | 3 | 3 | 3 |
| 10 | Beacon period( seconds ) | 600 | 600 | 600 | 600 | 600 |
| 11 | Connect redirector | 0, F | 0 | 0 | 0,F | 0, F |
| 12 | User message control flags | 27 | 19 | 19 | 19 | 19 |
| 13 | Hash (#) node broadcast port control | 3 | 3 | 3 | 3 | 3 |
| 14 | Extra aliases | 1 | 0 | 0 | 0 | 0 |
| 15 | Auto reconnect to node | 1 | 0 | 0 | 0 | 0 |
| 16 | Control of slime trails. | 3 | 3 | 3 | 0 | 3 |
| 17 | Digipeat up/downlink control: | 3 | 3 | 3 | 3 | 0 |

### *Notes*

**A.** HDLC default quality for Dedicated Point to Point Links and HTS free multiway backbones. This value is burned into the EPROM at quality 0 and left at 0 in the Parms but the *routes* to the neighbors are locked in at quality 203.
*R 0 callsign + 203*

**B.** HDLC default quality for gateway ports This value must be determined by the sysop such that the gateway port can fit into the neighboring network as well as being accessible to the NEDA-compliant network. 82 is a good value to burn into the EPROM. No matter what you have the HDLC quality set to the nodes that the gateway hears over the radio will not be broadcast over the RS-232 port because the minimum obsolescence to broadcast (PARM /6) is set to higher than the initial obsolescence. Do not lock in any nodes at > 50 quality.

**C.** Persistance and slot time for Hidden Transmitter Free, Gateway and Non-Ideal This value must be calculated based on the intended agressiveness of the port and the number of transmitters on the frequency. For HTS Free multiway backbones this should be equal to 256/(N-1).

**D.** FRACK time for a gateway node. This needs to correspond to whatever standards are being used by the rest of the stations on the radio channel. See *Modes & Parameters.*

**E.** Transmit keyup delay for Dedicated point to point link, HTS free backbone, and gateway. This number should be imperically tested. It should be set to 1.25 times the lowest value that works.

**F.** Connect redirector
This parameter has no real affect on the network. Read the Modes & Parameters section. Set to 0 usually.

***User port***

Visible node port where live keyboard stations access the network and access resources over the network. No station is heard on-channel by the user port that transmits > 3K/hour or 300K/month. No node to node communications on HDLC side. No nodes broadcasts on HDLC side. All node broadcasts on HDLC side ignored.

***Non-Ideal User Port:***

Visible node port where we don't want to do node to node communications but where there are other nodes or servers that this node can hear, on frequency, that are causing HTS effects. This might be a 2m user port that has gone to seed and sprouted automated stations that transmit more than 3K/hour: such as BBSs, NOS, KAnodes, digipeaters, HF gateways, APRS stations, DxClusters, weather nodes, TheNET nodes, switches, etc.. This port doesn't broadcast nodes on HDLC, or accept broadcasts.

***Gateway:***

Visible non 2meter node port that acts as a hopping point between networks. Accepts node broadcasts from both HDLC and RS-232 ports but only broadcasts itself The purpose of this is to allow two networks that have conflicting parameters to meet, allowing users and services to cross over, but without having the node lists intermingle. We're never going to get a gateway defined exactly because they will be customised for the various circumstances every time but this is the general idea.

***#HTSF HTS Free multiway***

Hidden non 2meter node (#node) for a protected HTS free backbone. All stations on frequency adhere NEDA parameters and all agree to not have more participants on frequency without further agreement. All HDLC routes are locked in. The default HDLC quality is set to 0. Persistance is set to 256/(N-1) where N is # of transmitters. ACL is used to selecively enable participating neighbor nodes. Default ACL=7.

***#DPPL Dedicated point to point link***

Hidden non 2meter node (#node) for a protected HTS free backbone with only 2 radios on channel within range of either site. Both stations adhere to NEDA parameters. The HDLC neighbor station is locked in with ROUTE command at both sites. Default HDLC = 0. ACL is used to selecively enable participating neighbors. Default ACL = 7.