Using the Wisconsin Packet Network

by Andy Nemec, KB9ALN

This series of articles originally appeared in the Badger State Smoke Signals between 1995-2003, and is a tutorial designed to help newer packet operators navigate the Wisconsin Network and use common packet radio facilities. Packet Radio operators who have been active for some time may also find some of this information useful, and we hope you do, too.

- Part - 1 - Explains what a node is in it's basic form.
- Part - 2 - Explores how nodes are connected together in a network.
- Part - 3 - How nodes exchange routing information.
- Part - 4 - Making the best of bad paths.
- Part - 5 - A network journey - Green Bay to Milwaukee.
- Part - 6 - The journey continues.
- Part - 7 - Segmenting your connection circuit - how and why.
- Part - 8 - Q&A about nodes.
- Part - 9 - A look at the X-1J node.
- Part - 10 - An introduction to TCP/IP packet stations.
- Part - 11 - Using TCP/IP stations as network nodes.
- Part - 12 - Your station as part of the packet network.
- Part - 13 - Optimizing your radio and TNC for packet use.
- Part - 14 - BBS Basics with emphasis on the MSYS BBS.

- Part - 15 - Sending messages from an MSYS BBS.
- Part - 16 - Using an MSYS BBS as a Node.
- Part - 17 - Q&A on the MSYS BBS.
- Part - 18 - Using the Kantronics KA-Node
- Part - 19 - Using the REQDIR and REQFILE on BBSs.
- Part - 20 - 9600 bps TNC and radio information.
- Part - 21 - Handling NTS traffic via Packet Radio.
- Part - 22 - How to avoid "Flame" messages.
- Part - 23 - Part 1 of "The New Age" of Packet Radio.
- Part - 24 - Part 2 - Automated packet and "new" protocols.
- Part - 25 - TCP/IP and Future Packet.
- Part - 26 - More TCP/IP and future packet - network needs.
- Part - 27 - General Info on mail forwarding to your TNC.
- Part - 28 - Setting a Tiny-2 TNC up for mail forwarding.
- Part - 29 - Setting a KPC-3 TNC up for mail forwarding.
- Part - 30 - Setting up an AEA TNC for mail forwarding.
- Part - 31 - Emergency operations and packet radio.
- Part - 32 - More info on emergency packet operating.

- Part - 33 - Planning your emergency packet system.

- Part - 34 - An APRS Primer.

- Part - 35 - Internet Gateways - Part One.

- Part - 36 - Internet Gateways - Part Two.

- Part - 37 - Internet Gateways - Part Three.

- Part - 38 - Internet Gateway Q&A.

- Part - 39 - A look at Flexnet.

- Part - 40 - What is TNOS/Linux, do I need it?

- --Return- - Back to the Main Index

Using the Wisconsin Network - Part One

by Andy Nemec, KB9ALN

This is part one of a series of articles designed to help packeteers understand and use the packet radio node network in Wisconsin. In Part one, we start off with a general discussion of nodes and networking, in basic terms. It is oriented toward the newer packet operators, but even experienced operators may find some useful information in here. We hope that this series adds to your enjoyment of the packet radio, and you find it good reading.

What is a node?

A node in it's basic form can be called an intellegent digipeater. Those of you who have operated your packet stations any time at all know that every TNC can be used as a Digital Repeater. This is done in order to extend the distance a packet station can send and receive packets. A digital repeater just simply repeats packets that are addressed to it. An intellegent digipeater has the ability to make sure that the repeated packets reach where they are supposed to go. Nodes are sophisticated, intellegent digipeaters, and some types are capable of much more than just digipeating. For our purposes, we will lump these nodes into 3 catagories, Non-Network Nodes, Special Purpose Nodes, and Network Nodes.

A Non-Network node is one that basically is an intellegent Digipeater. It recieves a packet from one station, And sends it to another, checking to see if it arrived. Alone it serves a useful function. It will also tell you what stations it has heard, and when. A good example of a non-network node is a Kantronics KA-Node, sometimes called a "Wild Node". This KA-Node is part of most Kantronics TNC's. Some nodes, including fancier Kantronics nodes, perform special purposes, like Gateway Service.

This brings us to the Special Purpose Nodes. These may connect 2-Meter packet radio activity to 20 Meters, connect Packet radio to a satellite link, access to a Public Bulletin Board System, or a DX Spotting system. Special purpose nodes may be linked to other nodes in a Network, but not always are. Some newer nodes provide so many features that they almost can be classified as Multi-Purpose.

Then we come to Network Nodes. The Network Node is connected to other Network Nodes, and together they offer one attractive feature. That is the ability to send and receive packets over great distances with great speeds. These nodes carry digipeating on to a science, and even speak their own language to make sure the packets reach their destination. There are several different kinds of Network Nodes out there, and we will not devote the time and space to something you aren't likely to encounter. In Wisconsin, the over- whelming majority of Network Nodes in use are called "**TheNet**". This is the type of firmware (operating system) that the node uses to route packets to the right place. Our discussion of Network Nodes will be oriented toward "TheNet" type nodes found in Wisonsin.

Note that the Network Nodes know how to get "route" a packet based on where you want to tell it to go. Each node has talked to other network nodes, telling it what it can hear and talk to. They exchange information on what other nodes it can connect to, and the signal quality of these nodes. We won't get into the details of how the process works here, but this fact is useful to know.

Each of these nodes has a callsign, like any other Ham Radio Station. They are sponsored by a club or an individual, like a voice repeater. They also have another name they are known by, an alias. This alias generally starts with the state name, and 3 to 5 letters that give some clue to its location. Usually these are chosen from the local airport designator. For example, here in Green Bay our local network node is known as WIGRB, WI for Wisconsin, with GRB the airport designator. Other nodes on the network know that WIGRB exists, the best way to get to it, and where else it can go.

This is valuable information for packet operators, because it helps to get them where they want to go. In Part-2, we will discuss how a Network is put together, so you understand how to use it in an effective way.

Using the Wisconsin Network - Part 2

by Andy Nemec, KB9ALN

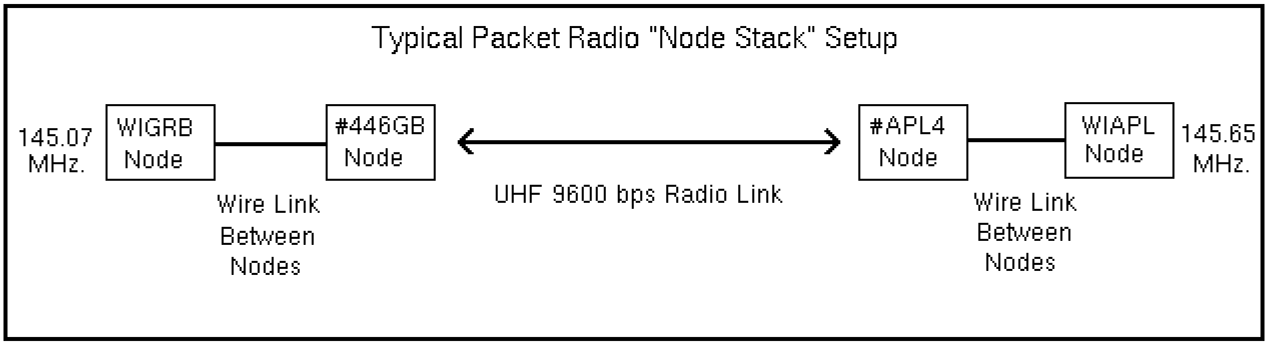
In Part one of our series, we discussed what digipeaters and nodes are, the different kinds of nodes, and what makes them so special. In Part Two, we will continue our discussion. Let's review.

A digipeater is a simple repeater of packets. A node is a more sophisticated repeater of packets, it knows if a packet was received, and can often allow you to send your packets over a very long distance. A network node is linked to other nodes and is often capable of doing more than just repeating your packets. We've talked about using nodes to pass our packets a longer distance. However, the need for a network of nodes grew out of another problem closely related to this need for distance.

You may have found that not everybody in the packet community is on the same frequency in different parts of the state. This was done on purpose, so that people would not crowd up on one frequency, making it useless. Different areas of the state are divided up into seperate Local Areas. This is usually done by region, county, or even parts of a city. Each division is called a "LAN", meaning "Local Area Network", having a separate LAN frequency. Some way had to be devised so that different areas could communicate with each other, and that is where the network of nodes comes in.

The Nuts and Bolts of a Node system.

A node in it's physical form is a TNC, a Radio and the antennas it needs to work, similar to other packet stations. The TNC is specially modified with a different set of operating instructions. Some garden variety TNC's have a node built in, like a Kantronics TNC. Network nodes are most often dedicated TNC/Radio Combinations set up specially for the purpose. Network nodes are also connected, by Radio, by wire, or by both. Nodes connected by wire at one location form a "Node Stack". The node radios in a particular stack operate on different frequencies and all "Talk" to each other, and with other node stacks. They exchange information concerning their ability to contact other network nodes. Network Nodes that contact each other over the air usually do so on dedicated frequencies reserved for Node-to-Node commun- ication. This is called a "Backbone", and is devoted to connecting different nodes together so that your packets can be passed a greater distance and to other LAN frequencies. The graphic below will demonstrate a pair of network node stacks that carry packets between 2 LANS by way of a backbone.



If you guessed that the people operating on the Green Bay LAN can connect to the people operating on the Appleton LAN even though they are on different frequencies, you are right. Other LANS can be arranged so that they, too are on the backbone. This is the basis of a packet radio network. In the next part of this series, we will put this knowlege to work. We'll use information you can get from a node in order to explore the network.

Using the Wisconsin Network - Part 3

by Andy Nemec, KB9ALN

In parts One and Two of this series, we explored what a network node is, how they are superior to Digipeaters, what they can do, and a little about how they do it. We learned about the concept of a Local Area Network (LAN), and how the network nodes link them together. Now we will explore more of what a network node does, and a little bit of how to effectively use them.

First, consider what might be needed to do the job of connecting one LAN to another. We talked about the fact that nodes "Talk" to each other, and that they know that other nodes exist, and what other nodes they can "talk" to. How is this done?

Well, nodes periodically "poll" each other to determine if a node is present, and how good of a communication path exists between them. They exchange a series of packets, and take note of the time that it takes a given node to respond in this exchange. A formula is used to calculate what is known as a "Route Quality" number. The higher the number, the better the "Route Quality".

Different route quality standards are used at different node baud rates. For example, a 1200 baud 2-Meter node may have a route quality of 192, and this is considered good. A UHF Node operating at high speed (9600 baud) may have a good route quality of 225. Nodes linked by wire on the same node stack will usually have the maximum quality possible of 255. Why is this important? Nodes use this information as a means to know how to route your packets through the network. How? Each node sends out a "Broadcast" to other nodes letting them know of other nodes it can reach, and of the route quality. A typical "Nodes Broadcast" may look like this:

```
KE9LZ-5>NODES UI Pid = CF
 Nodes Broadcast de WIGRB:
#446GB N9CFN-4 via N9CFN-4 255
#APL4 KB9BYQ-6 via N9CFN-4 222
#446DC W9AIQ-9 via N9CFN-4 222
#446CR KE9LK-8 via N9CFN-4 222
WIMAR KE9LK-7 via N9CFN-4 200
WIDC W9AIQ-1 via N9CFN-4 200
#GRB4 via KE9PW-5 255
```

Now a look at the broadcast in a little detail. The top line looks pretty much like most packet "headers" on your monitor screen, with one exception. Notice that it says "**PID = CF**". This is the "**Protocol Identifier**". Remember when we talked about nodes speaking to each other in their own language? This is what is known as the "**Net/Rom Protocol**". This explains why most TNC's cannot interpret these broadcasts, and why you may have never seen them. The above sample came from the Green Bay node with an alias of WIGRB.

Each route that is broadcast consists of a node alias, followed by that node's call-sign, the route it uses to connect to it, and the route quality number. The first entry is a node that is connected by wire to the node, because it's route quality is 255 (the highest it can be). The second, third and fourth entries are nodes that are reachable through a high-speed UHF radio link, of good quality, 222. The fifth and sixth listings are of good quality also, reachable through a UHF link as well, but are wire linked to the high-speed nodes #446DC and #446CR. The last listing is - you guessed it - wire linked to this node.

. By now you may be wondering why we have bothered to learn this information, because you can't normally see the broadcasts. The answer to this is simple: You can see a portion of this broadcast by issuing the ROUTES or R command to a node you have connected to. You will not see all of the routes on the route table, only the ones that can be accessed directly by the node.

This is useful information. It not only tells you that there is a path to a node, but how good it is. Like the rest of radio, not all routes are reliable 100% of the time, and this allows you to determine how to make a long-distance connection. The "Routes" command is one of the more useful that can be givien to a node.

In the next installment, we will explore more of how a node works, and what commands you will find useful when you "Surf the Network".

Using the Wisconsin Network - Part 4

by Andy Nemec, KB9ALN

In Part 3 of our series, we explored how a node broadcasts it's "**Routes**", and why this is important to the node. We also talked a little about how this information can be gotten from the node, and how it can be useful to you. We will continue here, and go further into how you can more effectively use network nodes.

We have talked about how radio paths are not always 100% perfect, and how the route quality numbers can help you determine how good a path to a given node is. But how do you determine where you can go, and more importantly, how to get there?

Remember that the node keeps a "Route Table" of nodes that it can reach. Each is assigned a quality number. This number is recalculated every time a route is rebroadcast through a node. For example, lets say that we have a Node called WAPR1 hearing a node with the alias of WAPR2. The path quality to WAPR2 is 160. Another node with an alias of WAPR3 hears the node WAPR1 with a route quality of 160. Node WAPR3 calculates the route to WAPR2 as being 140. This is because there is a little bit of route quality loss evey time a route is passed through a node. Earlier we mentioned that a route quality of 150 and higher will probably be a fair-quality route. Not excellent, but usable under most circumstances. If a route is lower than 150, it will not be reliable. So, most nodes are set to ignore routes with a quality lower than 150.

Routes with a quality of higher than 150 are put on a "Nodes List". You may have already seen a "Nodes List". You can easily see this by giving a node the "**Nodes**" or "**N**" command. This will tell you every other node that can be reached by the node that you are connected to. It lists both the Alias and the Call-Sign of these nodes. This Nodes List is updated automatically every hour. New routes are added when the nodes hears other nodes broadcasts, and old ones that are no longer heard are gradually deleted from the nodes list.

Notice the word "Gradually". This does not happen right away, and an "Obsolete" route may remain on the nodes list for a few hours. This is a source of frustration for packeteers, a node may be on the list, but they can't connect to it. The node thinks this obsolete listing is there, but it may have faded out due to band conditions. How do we get deal with marginal or obsolete routes?

First, use the "**N**" (**Nodes**) command sparingly. Sometimes, a rather sizable list of nodes will appear when you invoke this command. This will not only give you a few "Obsolete" node listings, but it tends to congest the frequency you are on, not to metion other frequencies that a distant node or nodes may be on. When connected to a distant node, the R (routes) command may be a much better indicator of nodes that you can actually connect to. It is also helpful to know where a particular node is, and where it goes.

Another node command, the **I** (**info**) command, will be most helpful. This will usually return one or two sentences telling you where the node is, perhaps who sponsors it, and sometimes what function it provides. This can help you pick your path by eliminating useless connections, ones that you do not need to make to get from point A to point B.

Another thing that will help you navigate the network is careful observation. Keep an eye on your local Nodes List. Keep track of what nodes reliably appear on the list, and when they appear. Nodes that appear only in the early morning hours are probably not reachable at about 2 P.M. It is wise not to try connecting to them in the late morning hours, even if they are still on the nodes list. Remember, it takes a few hours for a route to become obsolete and disappear from the nodes list.

Perhaps the most helpful suggestion that can be made about navigating the network with success is to use it carefully. If you like to log onto distant BBS's to check them out, use the features of the BBS to keep the connection quick, and more reliable. This is done by keeping your packets short. This will also be appreciated by your fellow packeteers. Large packets may disrupt the connections of others using the network. This is an ideal time to talk about using distant BBS's with care.

A large majority of BBS's in Wisconsin are similar. Note that they are networked together, and messages from one are forwarded to another, if they are for **ALLUSA**, **ALLWI**, or **DIST9** distribution. So it makes no sense to list all of the messages with the L command, because you will probably see the same messages on your local BBS. A much better option would be to use the L@ command.

Let's say you are on a BBS with the call-sign if KB9ALN. Send the command **L@KB9ALN** and you will get messages that originate from that BBS only, instead of the same messages that can be read at your local BBS.

Other "DX-BBS" hints:

1) Send the BBS the "**XS**" command. This sends one line at a time, and makes for shorter, more reliable packets.

2) Use the "**X**" command if you are very familiar with the style of BBS that you are on. This will give you a shorter command line.

3) Use a variation of the **X** command to limit the number of lines that the BBS will send before pausing to ask you if you want more. **X10** will tell the BBS to send 10 lines before asking you if you want more.

NEVER tell a distant BBS to send a listing Continuously. This will probably kill your connection, and it may disrupt other traffic on the network. Keep this information in mind when you use the network. We will show practical examples of how to use this information in our next installment of "Using the Wisconsin Network".

Using the Wisconsin Network - Part 5

By Andy Nemec, KB9ALN

In Part 4 of our series, we discussed the most important of the network node commands, **Routes**, **Nodes**, and **Info**. We also talked about careful use of the network, and efficient use of distant BBS's. In this installment, we will put this information to use. We will begin a journey through a network node path. We will discover some standardization that will help you navigate to a distant destination.

We have discussed the construction of a typical node stack and how they communicate. You'll need to remember that discussion as it does relate to navigating the Wisconsin Network.

You already know that Local Area Networks (LANs) are linked together by Backbone Nodes. These backbone nodes are the traffic carriers, and may often carry traffic between several LAN nodes. We might use an analogy here to better understand this system. Think of this all as a highway system. The LAN nodes are on-ramps to a freeway, the Node Stack is an interchange, and the Backbone is the "freeway". Sometimes it is useful to know the difference between a backbone node and a LAN node when you connect to one.

For example, it is not logical to connect to a distant LAN node and tell it to connect to another, more distant LAN node. Using our analogy, you would not want to exit a freeway and get tied up in an interchange if you wish to make an efficient, rapid trip on a freeway. So, backbone nodes will be carrying your packets, and your connection to LAN nodes will only slow your communication.

However, backbone nodes are generally designated as "hidden" nodes. If you connect to your LAN node and ask for a node list with the "**N**" command, you will see no backbone nodes listed. There are good reasons for this, but we will discuss them later. If we don't see the backbone node listed with the "**N**" command, how do we get to see that it is there? We use the "**Routes**" command.

The "**R**" command will show all nodes connected on the same stack, whether they are "Hidden" backbone nodes or not. Backbone nodes almost always have a **#** in front of their alias. Like the LAN node alias, there is usually a clue to a backbone node's location in the alias. For example, the Green Bay UHF 9600 baud backbone node has an alias of #446GB. It tells you that it is a "Hidden" backbone node, it is UHF, and it is in Green Bay. That is just a little cryptic, but the I command will tell you that is is backbone node that links Green Bay, Algoma, and Appleton, Wisconsin.

Now, let's start an imaginary journey through the network from Green Bay to Milwaukee. We will assume that you know it is possible, and that you may know that what your destination node is, but are not quite sure. We may have to take a few off-ramps to get there, but once we find the way there, we will know better than to take these exits.

The first thing you may have done is to get the list of nodes with the "**N**" command. There is nothing listed that resembles an alias that would indicate a Milwaukee node. A little bit of reasoning will let you in on what would be the most likely path to take to get there, though. The "**N**" command yields the following:

```
WIGRB:KE9LZ-5} Nodes:

GRBBBS:KB9ALN-5    MTWBBS:N9GHE      MTWDX:N9GHE-7     NEEBBS:KA9JAC
SHEBBS:NF9R        WIALG:KE9LZ-8     WIAPL:KB9BYQ-5    WICRIV:KE9LK-7
WIDC:W9AIQ-1       WIGLK:KB9WC-7     WIMTW:N9GHE-8     WINEE:KA9JAC-5
WISHEL:NF9R-8      WISTB:W9AIQ-7
```

The first nodes to rule out are ones that contain "BBS", for obvious reasons. Now we think of geography and airport designators. We want to look for a node that is part way toward our destination. If you want to take it one step further, you could look in the callbook would let you know for sure. There are a couple of possibilities, if you have a map in front of you.

WIMTW looks like a possibility, that looks suspiciously like an airport designator for Manitowoc. WISHEL (and SHEBBS) seem to indicate Sheboygan. A quick check of the call-book will confirm these deductions. Let's start by connecting to WIMTW, as that is about 1/3 of the way to Milwaukee. Once connected, send the "**R**" command. This will return:

```
WIMTW:N9GHE-8} Routes:
>1 #446MT:N9GHE-9 255 19
>1 #WIRED:N9GHE-6 255 2 1
 0  MTWDX:N9GHE-7 255 18
```

Well, now we can look at this and determine that there are 2 backbone nodes connected to this node stack. Notice the > symbol in front of #446MT. This indicates a "Route in Use". It also can point you in the right direction in your travels. Let's investigate the first one, #446MT, and send the "R" command. We then see this:

```
#446MT:N9GHE-9} Routes:
>1 WIMTW:N9GHE-8 255 20
>1 #WIRED:N9GHE-6 255 2 1
  0 MTWDX:N9GHE-7 255 18 0
  0 #446AG:KE9LZ-7 224 8 0
  0 #446SH:NF9R-9 224 18
```

Then we send "**I**" for **Info** and see this:

```
#446MT:N9GHE-9} Backbone Node 446.100 Manitowoc, Wi [44.92.22.2] 9600 Baud Backbone to Sheboygan and Algoma
```

This is information that you need to know. This tells you what this node does, and where it goes. A helpful hint: Take notes, or turn on your printer while you journey so that you know where you have been. When we continue in part 6, we will use this information to explore the the next stop on our journey, Sheboygan.

Using the Wisconsin Network - Part 6

by Andy Nemec, KB9ALN

In Part 5 of our series, we started a journey from Green Bay to Milwaukee, and stopped our travels in Manitowoc. We saw the "**Routes**", "**Nodes**" and "**Info**" commands in action, and started to decipher the seemingly mysterious and cryptic node aliases. We found that the "**Routes**" is often a much better indicator of where a node goes than the "**Nodes**" command, and we also found it useful in avoiding unintended journeys onto LANs.

Last month we stopped our journey in Manitowoc. Some folks felt they were "stuck" there. Now really, if you looked around a little, Manitowoc is not such a bad place to be, is it? This part will find us moving on to Sheboygan. In addition to being a nice town, it also has an interesting nodestack.

And when we were in Manitowoc, our use of the "**Routes**" and "I**nfo**" commands told us that the 9600 baud backbone node in Sheboygan is #446SH:NF9R-9. So now we will connect to that node. Once connected to this node, which we know as a backbone node, we send the "**R**", or "**Routes**" command. Here is what we see (in abbreviated form):

```
#446SH:NF9R-9} Routes

  1 WISHEL:NF9R-8
  1 #SHEX:NF9R-6 1
  0 SHEBBS:NF9R 1
  0 #SHE2:NF9R-2 1
  0 #SHEB:NF9R-3
> 0 #446MT:N9GHE-9
```

Now let's look at the possibilities for continuing our journey via the "Backbone" nodes. Remember that backbone nodes generally have aliases starting with the # symbol. The SHEBBS is just what you'd think - The Sheboygan County Amateur Radio Club BBS. #SHEX does not seem to indicate much to us (but really does - it is a "Cross-Stack" Node). There appear to be 2 possibilities here. Let's try connecting to them in order - first, #SHE2.

Once connected, sending the "**I**" (Info) command tells us that this is a backbone node intended to connect us to the KR9S DX Cluster. DX'ers may want to stop right there, but we are headed to Milwaukee and this is clearly not the right route. So, we send the **"B**" (Bye) command and are returned to #446SH with the "Welcome Back" greeting. Now we try the other possibility, #SHEB and connect to it.

The "**I**" command will tell us that it is a 4800 baud link between Plymouth (near Sheboygan), and a node located in Slinger (near Milwaukee), #SLGB. If you have been looking at a state map, you have a good idea of the locations. So now we connect to #SLGB. The "**R**" Command will show (again, in abbreviated form):

```
#SLGB:WB9TYT-8} Routes

1 WIMKE:WB9TYT-9
1 WICONV:WB9TYT-6
1 WISLG:WB9TYT-7 1
1  WINOT:WB9TYT-3
1 #SLGH:WB9TYT-5
1 WB9TYT:WB9TYT
```

It appears that we have neared the end of our journey. If you connect up to each node and send the "Info" command, you will find out which ones are LAN nodes, and what frequency and area they serve. If you have been keeping track of the journey, you will have a record of what nodes have been used. The correct sequence of nodes thus far is:

WIGRB, #446GB, #446AG, #446MT, #446SH, #SHEB, #SLGB, and then the LAN Node (Either WISLG or WIMKE).

We have finally made the journey, and have used Backbone nodes between the LAN nodes. In our next installment, we will discover a few techniques that will insure a good connection without as much typing. We will also see how to deal with "Marginal" paths.

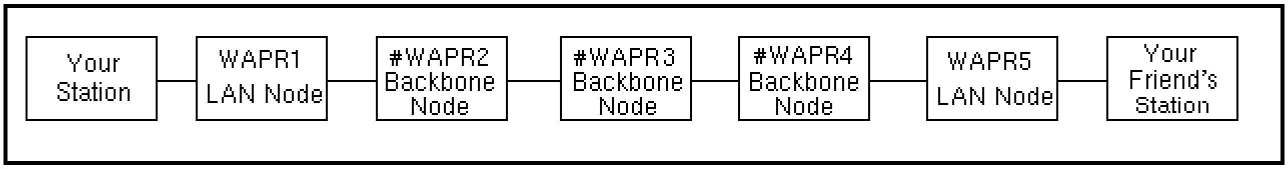
Using the Wisconsin Network - Part 7

by Andy Nemec, KB9ALN

Part 6 of our series found us completing our journey from Green Bay to Milwaukee. We made a brief layover in Manitowoc, and discovered just how to plod along via a backbone. In the course of our travels, we have found how to identify these backbones, and have discovered just how to determine where they lead. Now, we find out how to make our connections both faster and more reliable.

We have learned how the "**Routes**" command can give us the general layout of a nodestack, and how this is important in determining just how successful a link will be. Now we will discover a shortcut. First, let's remind ourselves of the nature of a network, and the basis of the network node.

Remember that a network is esentially a collection of intellegent digipeaters. They try to get a packet to a particular destination, and try to verify that it has arrived. But a long path of networked nodes leads to an interesting situation. Consider the way in which a network of nodes actually operates in the example below. You are traveling through 5 nodes to get to your friend's station a long distance away. You probably realize that a network of backbone nodes is used in your connection, and they are. Because the Backbone Node aliases are prefaced by a # sign, you will not see them with the "**N**" command. Assume that his LAN node shows up on your LAN node's Node List. Here is the path:



Your logical course would be to tell your LAN node, WAPR1, to connect to your friend's LAN node, WAPR5. Then you would tell it to connect to your friend's station. The problem is, you are going through 5 nodes over a long distance. Worse yet, you really don't know that the Backbone nodes exist until you plod along with the Routes command, because they are "Hidden" backbone nodes. Normally, you would not worry about the backbone nodes, because most of the time, the connections are all "automatic". You have noticed, however, that your packets seem to take forever to reach your friend. All of the sudden, you have a keen interest in the path used. You want to see if there is a better path, or a more reliable way, to make the connection.

Nodes try to make a connection over the best possible path, but are subject to all sorts of problems when they do. One of these problems is related to how a node makes a connection. Remember, a network of nodes is really a network of sophisticated Digipeaters, and that is how they maintain a connection over a long distance. When you connect up to your LAN Node, WAPR1 and tell it to connect to your friend's LAN Node, WAPR5, a lot of behind the scenes stuff goes on.

First, your LAN node knows that the way to WAPR5 is through #WAPR2. It connects up to #WAPR2 and, in effect, says "I have a connect request for WAPR5 from a user at WAPR1". #WAPR2 knows the route to WAPR5 is through #WAPR3. #WAPR2 says "I have a connect request for WAPR5 from a user at WAPR1". #WAPR3 knows that the route to WAPR5 is through #WAPR4. It says to #WAPR4, "I have a connect request for WAPR5 from a user at WAPR1." #WAPR4 knows how to get to WAPR5, and says that is has the connect request from a user at WAPR1. WAPR5 acknowleges, and sends the acknowlegement to #WAPR4. This acknowlegement is sent, "bucket-brigade" style, back to WAPR1. A "Circuit" is established, and each packet is sent in a numbered sequence to help keep track of it. This will make certain that each packet will arrive, theoretically.

So far, so good. But there is a problem with the way that nodes are commonly networked. Each packet requires an "End-to-End Acknowlegement". Remember, in an effort to make sure that packet radio is "Error-Free", each packet must be "Acknowleged", verified that it has been received correctly. "End-to-End Acknowlegement" means that if one station does not hear a packet, it must be repeated from the originating station. This is very similar to raw digipeating.

One end of the connection sends a packet, this packet is sent along through each node, and arrives at the other end. The acknowlegement packet is sent back to the originating node through each node that passed the original packet.

To use our example, if #WAPR3 does not hear a packet from #WAPR2, WAPR1 will have to send the packet again if it does not get an acknowlegement within a certain amount of time. If one of the backbone-to-backbone node links is marginal, or perhaps subject to some interference, then the packets seem to take forever to get to their intended destination. This is because each packet has to be re-sent along the weak part of the path. It also has to be acknowleged along the entire path. Even if the packet arrives, safe and sound, the originating node must know that it has arrived. Re-sending the acknowlegement wastes time, too.

Now that we know why a particular node path might be bad, we can do something about it. By now you are familiar with the "Nodes" command. But you may not know you can use a variation of it to find out just how good of a connection is possible to a particular node "destination". The variation is simple. If you need to know the how WAPR1 tries to get to WAPR5, just ask it by typing:

```
N WAPR5
```

You may see something like this:

```
WAPR1:W9XYZ-5} Routes to WAPR5:W9XYZ-9
#WAPR2:W9XYZ-6 232 2 1
#WAPR3:W9XYZ-7 152 5 1
```

What does it all mean? The first column of numbers (232 and 152) indicate the route quality number - the higher the better. The second column is the "Obsolescence Count". This is a measure of how long ago a node broadcast was heard, and when it will be dropped from the node list. The third column is the "**port**" number. Port 0 is the direct radio port, port 1 is a wireline to another TNC on a node stack. You can find out, node by node, how the route quality is for each step of the journey from WAPR1 to WAPR5. You may wish to use the "**Info**" command to find out where each node is, and then plot your progress on a map.

You may find, for some reason, that a route between two nodes is much weaker than the rest of the path. Here is where you can "**Segment**" the circuit. When you "segment" the circuit, you make a manual connection between 2 backbone nodes. Why?

Because acknowledgement is made directly between nodes, along a node-to-node circuit. As the proverb goes, a chain is only as strong as it's weakest link. If you let the nodes at the weakest link acknowlege to each other, the weak acknowledgement is concentrated in one spot - the weak link is made stronger. Rather than having an acknowlegement repeat through 5 nodes, it is only repeated between 2. Fewer repeated packets means a quicker, and a more stable connection. Circuit segmenting is not a revoloutionary technique, but it will make your network connections better.

Using the Wisconsin Network - Part 8

By Andy Nemec, KB9ALN

In the last part, we talked about using a variation of the "**Nodes**" command in order to determine the route to another node. We talked about "**segmenting**" the packet circuit so that the weakest link in the radio path is made stronger. This makes the overall circuit stronger, and faster. This month we will take a little time to answer a few of the most frequently asked questions about travelling the network. Names and Node aliases have been changed to protect the innocent and guilty!

Q. We have a network node called "WIBRAT" here in Bratsville, and a friend of mine operates a Kantronics node in his TNC called "WISAUS". I can connect to the KA-Node WISAUS from my station, and can connect to the "WIBRAT" node from his KA-Node. However, I cannot connect to WISAUS from WIBRAT. Niether can anyone else, so I know it is not my station. The WIBRAT node won't even try, it just says "Invalid Call".

A. This is a peculiar situation, but easily explained. You see, the WIBRAT node is a network node, and only recognizes other networked nodes as being able to use aliases. KA-Nodes, even if they have an alias, are not networked nodes. They do not know how to speak the same language. There is hope, however. If your KA-Node operator changes the call of the KA-Node to his standard call-sign with a different SSID, you will be able to make a connection. Most KA-Nodes are factory set to have the node as -7. For example, if I had a Kantronics TNC and I elected to run a KA-Node, it would be KB9ALN-7. I suggest that people stick with the default here. It brings uniformity. While you can't make a network out of KA-Nodes, they can be a help in remote areas that do not have a convenient network note.

Q. What is an **X-1J4**? We recently had a change in our local node, and now the **H** command does not give me the "Heard" list.

A. TheNet X-1J4 is a slightly different version of the familiar "**TheNet**" nodes you are used to. Why have people changed? Because it offers a whole lot more features and capabilities, and a whole lot more network control. By the way, your heard list is still there. It is now the "**M**" command (or **MHeard**). It is a better heard list, that tells you how long ago a station was heard. The "**H**" command, if you haven't noticed, gives a "**Help**" listing. In future editions of this series, we will explore TheNet X-1J and newer nodes and modes in depth.

Q. When I connect to my friend in Beavisville, about 100 miles away, it seems to take forever to get my packets through. Funny thing is, it shows that I am connected to him fairly quickly. Why does this happen?

A. This is indeed peculiar, but easily explained. You have a weak link somewhere in your circuit. The reason why you seem to be connected right away is because a connect request is a very short packet. Remember, short packets go through the network much faster than long ones. Try "Segmenting" the circuit, and see if that helps. See part 7 for info on this.

Using the Wisconsin Network - Part 9

By Andy Nemec, KB9ALN

In our last part of the series, we took a breather and answered a few of the most commonly encountered questions concerning network nodes. In this part, we will expand on one of the questions concerning the "**X-1J**" nodes.

Why? The X-1J series of nodes is compatible with and is very similar to familiar "**TheNet**" nodes we have been working with all along. The X-1J will accomodate newer operating modes, it is capable of doing a great deal more. It also makes conventional operating modes easier and more convenient to use. That is why it is rapidly becoming the new node firmware of choice in Wisconsin, and many other areas. In addition to providing all of the usual node services, it also has the ability to route TCP/IP Packets as well.

For those of you who are unfamiliar with **TCP/IP**, one can safely say that it is an amateur radio version of the Internet, and can be connected to the Internet. TCP/IP uses a different method to hook our computers together than the standard "AX.25" method we are all used to. TheNet X-1J allows us the opportunity to interface to the rest of the computer world, yet still retain "backward compatability" so that we can use AX.25 as well. In order to accomodate the expanded features, some commands needed to be changed from the older "TheNet" Node Firmware.

And that is the subject of this article, to compare the two and show the command differences. Question is, how do you tell what kind you are connected to? Luckily, there is one command that will tell you very quickly - the **U** command. Once connected to the node, it will not only tell you who is using the node, but what type it is. This is usually the first line you get when you ask for **users**. You will see one of the these 3, in all likelihood:

TheNet 2.08
TheNet 2.10st
TheNet X-1J4

Once you know what one you are using, you can use the appropriate command. You can find out which commands the node will take by sending a ?. **TheNet 2.08** and **2.10st** will show:

Bye Connect CQ Heard Info Nodes Routes Users

while TheNet X-1J will show:

Invalid Command - Choose from : Connect CQ Bye Help Info Nodes Routes Talk Stats Host BBS DXcluster MHeard Users Quit IProute ARP Adc

Quite a difference between the two! They have more in common than this would lead you to beleive, however. You can still use the X-1J to do what you have always done, you can just do a little more. First, we start with the similarities.

Bye, **Connect**, **CQ**, **Info**, **Nodes**, **Routes**, and **Users** all do the same in both versions. By now, you probably know what all of these do with the possible exception of **CQ**. But that it easy to understand, it just allows you to call CQ.

Now to the different use of **H**. In the **2.08** and **2.10st** versions of TheNet, it gives you a list of the most recently heard stations on the TNC in which it was installed. It the **X-1J**, it gives you a small **Help** file. This covers the most common commands and the command differences between this version and the conventional TheNet nodes.

The **Heard** list is still in the X-1J. It is called up with the **M** (or **MH**) command. An easy way to remember this is "**Most recently Heard**". This heard list is better than the older style. It tells us what "**port**" a station was heard on (Radio port 0 or wireline Port 1 to the rest of the stack). It will also tell you the elapsed time since a station was last heard, how many packets that station has sent, and may also show that station's deviation or signal strength. This depends on whether an accessory board has been installed and hooked to the Node radio. Not essential, but really handy if the node is equipped with it!

There are also some added commands. Here is a quick rundown and just what they do:

ARP - A TCP/IP Function. Gives the "Address Resoloution" table from the node.
ADC - Will give information from accessories hooked to the node, like a temporeature sensor.
IPA - Will give the TCP/IP Address of the node.
IPRoute - Gives the TCP/IP Routing table for this node.
BBS, DXcluster, Host - These commands will hook you up to the local BBS, DX Cluster, or TCP/IP Host Computer. Easy enough.
Quit - Same as Bye.
Stats - Not of too much use to the average user. Will show statistics on Node (CPU) useage.
Talk - Conference mode. To enter, type **T** when you first connect up to the node. To exit, type **/EX** on a line all by itself.

And that is a quick overview of the differences between the newer "X-1J" series nodes and the older "TheNet" nodes. Not all of the features are enabled on every node, but this sumarry shows what this powerful new firmware is capable of. If you would like more detailed information, look in the nodes section here on this site.

Using the Wisconsin Network - Part 10

by Andy Nemec, KB9ALN

In the past editions of "Using the Wisconsin Network", we have devoted a fair amount of space to the operation of "TheNet" Nodes, and how to use them. While this is important to knowing the network, there are other types of stations you may encounter on your travels. We will try to cover most anything you are likely to find, and attempt to give some practical background information for you. We will start with an introduction to one of the more mysterious types of stations you may encounter, the TCP/IP station.

An ever-increasing number of hams are exploring this mode, and chances are you may encounter a station using it. Once we get familiar with it, we will move on to examples of TCP/IP Amateur Radio stations and show you how to use them.

What the heck is TCP/IP, anyway?

TCP/IP stands for **Transport Control Protocol / Internet Protocol.** I know, that tells you what the letters mean, but just what is it? Well, surprisingly enough, you really don't need to know too much about the terminology to get the idea of what it is basically about. You might remember from one of the earlier editions of this series a discussion of "**Protocols**". In case you might have forgotten about this, we'll review.

A protocol, in computer terms, is the method or language that computers use to pass information back and forth. A computer can use one or several methods to speak to another computer. The important thing is that the computers must both be speaking the same language at the same time and speed. Most Amateur packet radio stations use a protocol, or specific method to communicate with each other called **AX.25**. This method of communication was adapted from Ma Bell (AT&T) for use on Amateur radio.

"**TheNet**" Nodes use a variation of this protocol called "**Net/Rom**" to exchange information. Although they work OK in Amateur radio, these protocols are not widely used in the rest of the computer world. For example, the Internet, a global network of connected computers. These computers all need a standard set of protocols in order to communicate. The standard set of protocols used is referred to as "TCP/IP". Amateur Radio has adapted TCP/IP protocols, and others, to communicate via packet radio. Amateur Radio TCP/IP Stations are also capable of communicating with the AX.25 protocol. This is the legally recognized protocol for Amateur Radio, so all of the TCP/IP Packets are "Buried" in an AX.25 packet. This also means that someone who only uses AX.25 can communicate with a TCP/IP station (though they may not be able to use all of it's features).

Why TCP/IP?

One of the main attractions of using TCP/IP might be already apparent. Because it is possible for the computer to communicate with a world-wide network using TCP/IP, the opportunities become seemingly endless. Stations that are part of a Amateur Radio TCP/IP network have greater flexibility of operating, providing the operator with more than the usual mailbox and keyboard services. The computer becomes an integral part of a packet station with TCP/IP, because the computer is "connected" to another computer. With standard AX.25, TNC's are connected, typing is done, and TNC's are then disconnected. Your computer operates as little more than a "dumb terminal".

Consider what is possible amongst computers using TCP/IP:
-Standard keyboard chats
-Standard "PBBS" type mailbox for AX.25 users
-Automated Mail delivery to other TCP/IP stations, AX.25 mailboxes, and BBS's
-Automated message forwarding
-Access (limited) to a computer's hard drive
-Ability to transfer text and binary files simply and easily (including Wave, GIF, JPG, COM, and EXE

files, to name a few)
-Ability to test radio paths before using them
-Possibility of receiving special-interest "Newsgroups" by automated mail
-Extensive Remote Sysop-ing of the station
-Possibility of connection to the Internet
-Ability to carry on a multi-station conference discussion
-Possibility of networking computers in your own home, so that you may operate your packet station from outside of the shack
-The ability to have a TCP/IP station operate as a Network Node

And that is where most of us will get our first exposure to a TCP/IP station, as a network node. Now that you know a little bit about them, we can devote some space to a discussion of using one of these stations. Look for that in the next edition of "Using the Wisconsin Network".

Using the Wisconsin Network - Part 11

by Andy Nemec, KB9ALN

Last time we turned in a slightly different direction in our exploration of the Wisconsin Network. We started to explore the somewhat mysterious world of the **TCP/IP** station. We learned just what "TCP/IP" is, and what makes them different from the garden-variety "AX.25" station. We also learned that they are capable of the "standard" AX.25 protocol so that they can communicate with "standard" stations. And we learned that these stations can also talk to Network Nodes, and are part of the network.

And this is why we are learning about them; you may encounter a TCP/IP station that functions as part of the network and we should learn how to use it as such. Most IP stations you will come into contact with use the network in a different way, to support the TCP/IP functions of the system. Because of their inhernet versatility, they are gaining wide usage as "Packet Switches". They can handle various different protocols, and therefore, can do many different jobs. Most likely, when you connect up to one, you will be connecting to what looks like a BBS. But a TCP/IP station can and is often used as a node. When you connect, you may get a greeting screen that looks something like this:

```
[WNOS-4B0-HMI$] Currently 1 user.

Welcome to KB9ALN's Mailbox and Node in Green Bay, Wisconsin.

To chat with me, Type C . Mail area (WX9APR): 0 Messages
Enter Command: (a,b,c,co,conv,d,du,e,f,h,i,k,l,m,n,n c,p,q,r,s,t,u,v,w,?)>
```

The first line you see identifies the program in use by this station, which is called "**WNOS**". WNOS is an abbreviation meaning "**WAMPES Network Operating System**". WAMPES is a group of Amateurs from Germany who modified a program written by **Phil Karn**, **KA9Q** called **NOS**. Virtually every flavor of TCP/IP has NOS in it's name.

The "**HMI$**" part of the first line lets other BBS's know that this station will accept forwarded messages, among other things. The second line is self-explanatory, the users currently connected. The third and fourth lines of text are just a greeting and an instruction as to how to do a keyboard chat with the operator of the station.

Notice that the fifth line mentions a "**Mail Area**". Every one who connects up to the station has a personal mail area where their messages are stored. You log onto your own personal area. The user connected here is WX9APR (a call I pulled out of thin air for example), and has no messages. If you were to connect up to this station, you would see your call sign in place of this one.

Now, to the command line. It does look similar to a BBS, doesn't it? There are a fair number of BBS function in there, and they function much like any other BBS. We'll cover each command specifically below:

A - area

This is used to find out about the various *mail areas* available, and allows you to change to a different one. When you log onto a TCP/IP station, you are in your *personal mail area.* Those of you who have used the **MSYS** BBS systems may be familiar with the concept of *message catagories*. The mail area is very similar.

B - bye

This will disconnect you from the station.

C - chat

Allows you to talk to the Sysop, if available. Some TCP/IP stations use the "**O**" command (for "**Operator**").

CO - connect

Makes a connection as a node to another station. Sometimes, this will make a network node connection for you. Most of the time it will connect you to another station much like a "KA-Node" does.

CONV - "converse"

This allows you to join in a "round table". More than one person can join in a conversation with you and all stations can see what each participant is sending. Their call-signs are inserted in front of each sentence they send.

D - download

Allows you to download a text file. The format is **d (filename)**

DU- download

This allows you to download binary files that are "**UU**" encoded. The format is **du (filename)**

E - escape

Allows you to set an "**escape character**" This is a key combination (like control-x) that is used if you want to abort a process, but wish to remain connected to this station.

F - finger

This retrieves a short info file about a user of this system.

H - help

Just what you would expect. In addition, you can use this in conjunction with a particular command. **Help Finger** will give you specific information about the finger command.

I - info

Gives you a short info file about this TCP/IP station.

K - kill

Deletes a message, just like any other mailbox or BBS.

L - list

Lists messages. Again, just like any other BBS.

M - mheard

Returns this station's heard list.

N - nodes

Just like any other node, this is a list of network nodes that this mailbox and node can connect to.

N C - "Netrom" or "Network" Connect

Some TCP/IP stations need to know that you wish to connect to a network node. If you wish to connect to the Green Bay LAN node WIGRB, you would send: **n c wigrb** .

P - path

Gives a list of nodes that this node can directly connect to, without going through the Network. Similar to "**routes**".

Q - quit

Same as "**bye**".

R - read

Read a message, just like a conventional BBS.

S - send

Send a message. Again, this is the same as a regular BBS or Mailbox, and the same process is used.

T - telnet

This is TCP/IP lingo telling this computer to connect up with specified computer running TCP/IP.

U - users

Shows who is currently connected to this station.

W - what

Sends you a list of What files are available for downloading.

Now you may have a better idea of how to use a TCP/IP mailbox/node. There is much more that you can learn about these stations. Space permits us to only go so far. But operation as a network node is remarkably similar to any other node you may have used. If you have questions about a TCP/IP station, the best way to find out more is to contact someone who operates this mode. Most folks who operate these stations are more than happy to help you use their station.

Using the Wisconsin Network - Part 12

by Andy Nemec, KB9ALN

In the past editions of "Using the Wisconsin Network", we have devoted a lot of discussion to using the various network stations. We have also discussed, in basic terms, just how these pieces work together. In this installment, we will go backwards. Yes, we will look at the start of the network by looking at your station. You see, the network starts at your station, from your point of view, and from others as well.

Why do we say this? Go back to what a node is (and what it does) for your answer. A node is a sophisticated repeater of packets that you, and the station you are connected to, originate. If any of the stations using the network are not set up correctly, a node gets busy, and in some cases can sever a connection altogether. Looking at packet radio basics will tell show us why this can happen.

Remember what the basis of packet radio is - the sending of digital data by way of a shared radio channel. The fact that packet radio data is sent in short bursts allows several stations to use one frequency in the same time frame. In fact, packet radio stations share time as much as they share a frequency. Now let's review how packet becomes "error free" (or nearly so).

First, a line (or character) of text is sent from your computer to the TNC. The TNC counts how many bytes of data are being held in it's memory, ready to be sent out as a packet. When the number of bytes gets to a certain value (called the **PACLEN**), the TNC assembles a packet. When a packet is assembled, it calculates a number, called a **checksum.** This sending TNC assigns a sequence number to this packet, and addresses it.

Then, the TNC checks to see if the any other stations are sending any packets on the frequency. It waits for what it thinks is a good amount of time before transmitting. Finally, the TNC keys the transmitter and sends the data to the transmitter in the form of audio tones. When the complete packet is sent, it unkeys the transmitter, and waits.

If receiving TNC can copy the packet, it converts the audio tones back into data that the computer inside of the TNC can use. It disassembles the packet, determines if the packet meets the checksum, and if it is of the correct protocol. Remember, most user stations use the **AX.25** protocol, Nodes use **Net/Rom**, and TCP/IP stations use **IP**. Then it replies to let the sender know it's status.

The receiving station replies in one of a few ways, if it hears the packet. If the packet meets the checksum that is sent with the packet, and the sequence number matches what it is expecting, it checks to see if the frequency is clear. It waits for what it considers a good amount of time, and responds with a "**Receiver Ready**" packet. It also tells the sender what the next expected packet will be numbered as. All is well so far.

If however, the packet is corrupted for some reason, or it is a duplicate packet, the receiver will send a "**Reject**" packet. If it fails the checksum, it tells the sender this. Again, it tells the sender what sequence number of packet it is expecting. If the packet is not of the expected protocol (generally AX.25), then it sends a "**Frame Reject**". If it is a serious protocol error, it will break the connection and make you start over. Protocol errors between two AX.25 packet stations are rare, however, and are usually due to incorrectly decoded packets.

If the receiving station does not hear the packet, or if the packet is so badly distorted that it cannot recognize it, it does not reply. The sender is waiting for a reply, however, and a timer determines just how long it will wait before it asks "Did you get the last one?". If no reply is heard, and the timer expires, it does just that. This is called "**polling**".

While all of this might be "more than you really wanted to know", it is important to understand in these basic terms. Why? Look at all of the timing at work here. First, we divide up time to allow several stations to use a given frequency. The sender waits to send data. The receiver waits to acknowlege the receipt of data. The sender waits a certain amount of time before checking to see if the packet has arrived. As with the rest of life, "Timing is Everything!".

Consider what happens if a station sends a packet out, and does not wait long enough for the receiver to reply. That is a wasted packet. What happens if a receiver does not wait long enough before acknowleging a packet? Some other station will not get a chance to transmit a packet. What happens if both stations do not wait long enough between transmissions, or between polling? Then the frequency is virtually "locked up" as long as these stations are connected! And that is where the network comes in.

The network starts with you, and your fellow packet operators. Remember, we are all "time sharing", as well as frequency sharing. All of these timing parameters are configurable through simple TNC commands. The sad fact is, the defaults that are factory set are totally unrealistic in today's packet radio environment. Most of these were decided on in the early 1980's when packet radio was not nearly as well-populated as it is now. Add to that the fact that a lot of TNC instruction manuals are written in Techno-Babble, and it is no wonder that we are all confounded and prefer to leave them as they are. We could all be operating a lot more efficiently than we are now. And that is what the next installments of this series will be about. We will discuss TNC parameters, what they mean, and how to set them up in a cooperative manner.

Using the Wisconsin Network - Part 13

by Andy Nemec, KB9ALN

In the last edition of "Using the Wisconsin Network", we started looking at the network from it's beginnings, your station. We discussed in basic terms, just how these pieces work together. In this installment, we will continue this discussion and look at setting up your radio and TNC so that your station becomes more "friendly" toward the network, and to other stations operating on your LAN frequency. And we will start in that order, with the radio.

Many people forget that packet radio is radio. The computer and associated boxes you have may all be working just fine, but useless if your radio is not properly set up. Remember that your data ultimately gets transmitted and received as audio tones, so the first place you have to look at packet performance is at your radio. First, let's look at transmitting.

Your transmitter has to be on frequency, have the proper amount of deviation, and has to have *clean* audio. No Buzzes, no P.L. tones, and no RF getting into the microphone audio circuitry. Frequency error is often overlooked when people are trying to diagnose a bad packet connection. If you have any doubt, try checking it with a frequency counter. You will need absolutely NO audio into the mike circuitry when you do this, and don't forget to turn off your P.L. Assuming that you find your radio on frequency, you can go on to the next step, checking out your transmitter deviation.

Your audio level from your TNC directly effects this, so you will need to hook your TNC back up and send some packets. To accurately measure deviation, you need to look at the signal on a Service Monitor. This handy test gear is out of the range of most Amateurs, but ocassionally you will find someone who has one. If you have access to one of these, your peak deviation should be approximately **+/4 KHz**. If you are like most of us and don't have access to one, you can come close with an oscilloscope and another radio to listen on. If you don't have a 'scope, then your ears will have to do. The trick to these two methods is to have a reliable reference to compare your signal to.

This would probably be the local node for most people. You adjust your transmit audio level (On the TNC) so that it matches the volume level of the station you are using for a reference. That is why you will need to listen on another radio. It is easier and more accurate if you use an oscilloscope hooked to the speaker terminals of the other radio. It is even more accurate if you hook it to the discriminator of the other radio, and have it calibrated. This is not usually the case with most people, so you will probably be hooking it up to the speaker.

Once you have your transmit audio level set, you may wish to look at the cabling to the TNC. Use high-quality shielded cable for your connections to the microphone circuit (or Data connector, if your radio has one). There is also one final part of the transmit system you should pay attention to. That is what kind of antenna you are using, and where you have placed it in relation to your computer and TNC.

First, using a rubber-duck antenna is not a preferred setup for packet. Not only does it not work very well when receiving or transmitting, but it also hears stuff you don't want it to hear. Keeping a rubber duck in close proximity to the computer and the TNC will insure the reception of computer hash from both of them. It also has the nasty property of radiating a signal into your electronics, which can distort your transmit audio and possibly cause damage to your system. Best to have an antenna on the roof, or at least a good distance away from the rest of the equipment.

Now on to the receiver. There are 2 simple adjustments here. You don't need too much volume to get the TNC to decode properly, so setting the volume at 10 to 11 O'Clock is generally quite enough. More is not better here! The squelch may require a bit of tinkering with. Generally speaking, the squelch should be set about 1/8th turn past the threshold of quiet. Sometimes lower noise levels will open up your squelch. This is not good, because in most cases this will cause your DCD light on your TNC to come on, and will prevent your station from transmitting (it thinks someone else is transmitting). Don't overlook this if your station does not want to transmit!

Now that we have looked at the radio end of your station, we will move on to the TNC. As was said earlier, this is important as it affects the way that your station interacts with others using the network and the LAN. The **North East**

Digital Association (**NEDA**) has set up guidelines for the real world of packet radio that make sense in today's packet radio environment. They make a great deal of sense when you look at it, and we will use their guidelines as a basis for setting up our TNC's. Below is a list of the more important parameters, what they do, and what is a realistic setting for them. Remember, most factory defaults were set in a day when packet radio was not as well populated as today, so it makes sense to change them to match today's packet radio world.

BEACON = 0

Turn off your beacon. There is no need to have a beacon unless you have important information to disseminate. Your idea of a good joke of a clever saying is not a good reason. If people need to know that you are out there, they can look at their "Heard" list. "Mail For" and ID broadcasts are legitimate beacons. Every Amateur radio publication relating to Packet Radio tells us that Beacons are a waste of spectrum. Let's end this practice!

DIGIPEAT = OFF

Turn off your digipeat feature, unless you have no node in the area, or it is absoloutely necessary for some reason. Nodes do a much better job of repeating your packets, and keeping the Digi on will often override other features of your TNC that govern timing.

FRACK = 6 to 12 (seconds)

This is the "**FR**ame **ACK**nowlegement" timer. It governs how long your TNC waits for an acknowledgement from the other station you are connected to. In croded areas like Milwaukee, for instance, you may want to use 10 to 12.

MAXFRAME = 1

This governs how many Data frames are sent with each transmission. Too many frames will keep your transmitter up too long, and not allow others to share the channel. Also note that the more frames sent, the longer the packet. The longer the packet, the more chance of errors, and this means poor throughput of data.

PPERSIST = ON

This allows the TNC to wait a random amount of time before transmitting. You must have the Digipeater turned off for this to function. This gives other stations a chance to transmit, too.

PERSIST = 64

This generates a random number to be used in conjunction with **SLOTTIME** to calculate the time it waits before transmitting. The lower the **PERSIST**, the longer it waits. Though 64 is suggested, you may use 128 if yours is not a busy packet LAN frequency.

SLOTTIME = 20 (in ms)

This is used with the **PERSIST** setting and again, governs the time the TNC delays before transmitting a data packet to the receiving station. Note: MFJ calls this "deadtime".

RESPTIME = 10 to 15

This is a timing parameter that controls how long the TNC waits to acknowledge a received packet. This differs from the above **PERSIST** in that **PERSIST** is involved in sending a data packet. **RESPTIME** is only effective in the acknowlegement of a received packet.

RETRY = 10

This is the maximum number of retries allowed before a station sends a disconnect. You may also consider setting this to 6 or so. If you can't get to somebody in 6 tries, someone has problems!

Note that some TNC manufacturers may use slightly different command names than we have used here (especially Multi-Mode controllers). Take a good look at the TNC manual and see if any of the commands have a similar description. While this all may seem to be more trouble than it is worth, it will make a pronounced difference in throughput if you take the time to change a few settings. And anytime you can improve efficiency and channel usage, it is certainly worth the time.

If everyone on a LAN can cooperate and use the same settings, life will be easier for everyone. And if life is easier on the LAN, then it will be easier on the network. And that will help you use the network more effectively.

Using the Wisconsin Network - Part 14

by Andy Nemec, KB9ALN

Last time we finished our look at your station and how it works with the network. We covered reccomended TNC parameter settings, and talked about how these affect network performance. But no matter how well your TNC and radio are "tweaked", your operating habits can also affect network and LAN performance. They can also affect your enjoyment of this mode. Many people are not quite clear on some aspects of packet operation, and this can frustrate them and keep them from enjoying the potential of this mode. One of the more popular Packet Radio activities to be found is the daily check-in to the local BBS. As simple as this ritual may be, there is a great potential that is largely unused in most of the BBS's that are in use in Wisconsin.

Most of the BBS's in this state are called MSYS. It is a particular kind of program written by Mike Pechura, WA8BXN. Simple functions are easy enough to use, if you are familiar with them. Some of the more complex functions are often ignored; some folks simply have not explored them. We will not cover all of these, but will tell you how to more efficiently utilize one of these BBS's, so you can get the maximum enjoyment out of it, and be nicer to your packet neighbors. In this installment, we will cover the "receiving" of packet messages. Next month, we will cover "sending". In the following installment , we will look at the MSYS node operation. First, there are some things you should know about MSYS.

Although it is a capable program, it can be a "channel hog". It has a tendency to be very aggressive, and will do a very thorough job of dominating the LAN frequency. This affects your use of it, you can aggravate the situation or help to alleviate it. There are also some MSYS BBS's that perform a dual function; they are not only BBS's, but Network Nodes as well. Lastly, there are certain things you should know before you start sending "flood" messages (commonly called bulletins), no matter what type of BBS that you send them from. First, a few BBS basics.

Those messages you read that come from all over the country (and the world) are passed from one BBS to another, bucket-brigade style. What appears on your local BBS will very likely appear on any other BBS you visit in the state. Each BBS has what is called a "hierachial address", and that is how your personal messages make their way to you, your home BBS has a unique address. It is best to choose a home BBS that is close to you. It will save a lot of time, and it very likely is no different than the one that you have in your town.

Now, about your "home" BBS. Everyone who elects to send and receive packet mail must choose one (and *only* one) home BBS. You choose your home BBS when you register at a given BBS. Once you register a BBS as your home BBS, you have a hierarchial address at that BBS. If you have any doubt as to what your hierarchical address is, or how to register at a BBS, leave a message to the sysop of that BBS and he or she will be happy to help you out. The Hierarchial address is most significant if you expect to receive a reply to any message.

There are 3 basic types of messages; personal, bulletin, and NTS traffic. Personal messages are just that - intended to a particular person and no one else. When you log into an MSYS BBS, you will be notified if you have mail waiting for you. On the current versions of the MSYS BBS software, bulletins are listed by category. When you log in, you will see a listing of such catagories, which is drawn from the "To:" address of the message. NTS traffic is found by using the **LT** command. Many people do not realize that this command exists. It is easy enough to use and can be helpful to pass local NTS traffic.

Recent versions of MSYS BBS software are oriented toward simple operation. As was said earlier, you will get a listing of your personal mail when you log in. For example, I may see something like this if I have personal mail waiting for me when I log in:

```
Welcome Andy, to KB9ALN-5's MSYS BBS in Green Bay, Wisconsin.

You have unread mail, please kill when read:
MSG #  TR   SIZE   TO        FROM      @BBS      DATE     TITLE
33989  PN   2112   KB9ALN    N9PAY     KB9ALN    960401   April Fools!
```

Enter right 1 digits of msg #'s to read or enter for none

To read the message, I would send the last right-hand digit (in this case, 9) to the BBS and it will display the message. It's that simple. If I did not want to read this message just yet, I would hit the Enter key. I could later list this message with the **LM** command, and read it as well.

Bulletins are just as easy to read, but can often cause a great flood of packets that can greatly disrupt the LAN frequency and the network, if done in a sloppy manner. Most people are familiar with the **L** command. This simply lists new messages that are on the BBS. The problem with this command is that it is not selective at all - it will list all the messages the BBS has received since you last logged on. You will see a great many listings for things that you may not have an interest in. This is a waste of the frequency.

As I said before, MSYS is a very aggressive program. A **LIST** command will make it very hard to share the channel with others. There is a much better alternative, though. When you log onto the BBS, a list of message categories is nicely displayed for you. If you have an interest in For Sale items, look at the catagory called SALE. You can list only the messages in this category by typing **L SALE**. This way, you get a chance to read the messages you want without jamming up the frequency looking at a list of messages you don't want.

Periodically, the BBS will pause and ask you:

"More? [Y]es, No, Continuous?"

To receive another screenful, hit the return key. Sending an **N** will stop the listing. Sending a **C** is Rude! This means that the BBS will send a listing of messages that will not pause until it is done. *NEVER* do this, you will tie up the LAN or Network for a significant amount of time. This is one "feature" that should immediately be written out of the MSYS program; it is pointless and causes more problems than it is worth.

Sending the BBS **LC** will give you a listing of the message catagories, should you list one category and forget what other message categories are available.

XC toggles between an automatic listing of catagories when you log in, and no listing of categories at log-in Reading a bulletin is much the same as reading a personal message. As before, the BBS will periodically pause and ask you if you want more, or if you would like to read one of the messages. It will prompt you for a certain number of digits in the message number. You would type in those digits, and it will display the message. ***Hint:*** You may read more than one message in succession, and you must give the proper number of digits. For example, if the BBS asked you for the right 3 digits of the message number, and the message number is 26095, then you will send 095 . You will need to send the 0 in order for the BBS to correctly interpret the message number. This covers the basics of reading messages. Next time, we will cover the correct way to send a message or a bulletin.

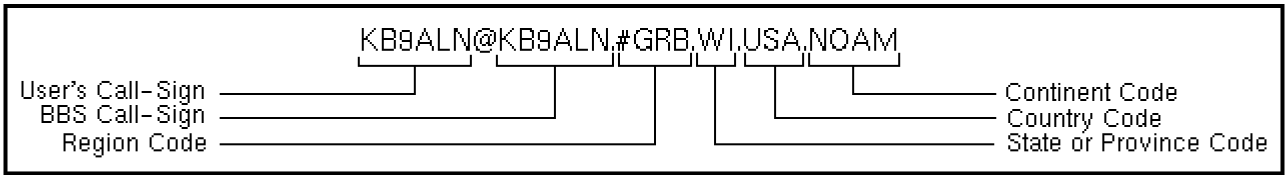
Using the Wisconsin Network - Part 15

by Andy Nemec, KB9ALN

In the last edition of our series, we talked about using your local BBS, specifically listing and reading messages. We noted that the great majority of the BBS's in Wisconsin are "MSYS", and that the instructions that you get here will probably apply when using your local BBS. Even if your BBS is of a different type, they all operate in much the same manner.

The same can be said of sending a message. No matter what type of BBS you may encounter, the procedures regarding the sending of messages are even more universal than the special features encountered in the listing and reading of messages. One important item we touched on in the last installment deserves review and elaboration. That would be your hierarchial address.

Remember, everyone who uses packet radio and elects to send and receive packet mail has one. It is derived from your call-sign, the call-sign of your home BBS, a regional descriptor (optional), as well as state, country and continent codes. For example, we will use my hierarchail address to dissect and analyze. Here it is:



Note the structure of the address, how it is separated by periods, and somewhat resembles an internet E-Mail address. This is important to remember when you decide to send packet mail to your buddy who might spend his winters in Florida. You need to know his hierarchical address to send mail to him, and he needs to know yours, and they both have to be correct. Once you do, it is a simple matter to send him a message.

As with any other mailbox, you use the **S** command, along with the hierarchial address. We will, for the purpose of illustration, make up the following address - **W2XBY@W2XBX.#LI.NY.USA.NOAM**. which belongs to a mythical friend Joe-Bob, who uses W2XBX as his home BBS in New York. So when you log onto the BBS and decide to send a message, you would send this to the BBS:

S W2XBY@W2XBX.#LI.NY.USA.NOAM

The BBS Would respond with

Enter Title or City, State, and Postal Code:

Where upon you might send "Hi, Joe-Bob!", or something similar. The **City**, **State** and **Postal code** reference is for sending **NTS traffic**. It is not necessary for sending personal messages when you know the hierarchical address of the recipient. Adding the postal information may occasionally be helpful if the BBS is a new one, or if you have an incomplete hierarchical address.

Once you have sent a title, type in the body of the message and end it with a Control-Z key combination, or /EX on a new line.

That is all there is to sending a personal packet message. Sending a Flood Distribution ("bulletin") message is not very hard, but requires a little more thought. It is possible to send a message to every BBS in a state, region, country or even world-wide. One must use wisdom and logic when deciding just how far you want to "flood" with a bulletin.

Most people use a bulletin to advertise some piece of amateur radio equipment for sale. Some play Chess by packet radio, some develop special interest groups via packet, and still others look for information or equipment. Logic and your knowlege of the rules should govern your use of this powerful capability of the packet system. For example, one would never try and sell a $5 item via a packet message sent to every BBS in the country. Shipping cost may make it seem like a silly effort, as sometimes it would double the price of your $5 item. This also applies when it comes to large or heavy items. Legalities also apply with Sale and Wanted-to-Buy items. The same rules apply to packet sale ads as swap nets. It has to be a piece of equipment normally used in amateur radio operations. These are valid considerations that need to be addressed when you start to send a "bulletin".

That being said, we can now explore the options you have at your disposal, and how to use them. There are a number of "flood designators", and in Wisconsin we use **ALLWI, DIST9, ALLUSA, ALLCAN,** or **WW. ALLWI** is a designator for "All Wisconsin". **DIST9** distributes your message throughout the 9 call area (Wisconsin, Illinois and Indiana). **ALLUSA** distributes your message to every BBS in the USA, **ALLCAN** to every BBS in Canada, and **WW** to all networked BBS's in the world. So, we do need to be selective when we route a "flood message". Now, to actually send a message in this manner.

Let's say we want to send a message throughout Wisconsin to everyone interested in WAPR. When we get the BBS's command prompt, we would type the following:

```
S WAPR@ALLWI
```

and then wait for the "**Title..**" prompt, similar to the way we would send a personal message. The difference is that we have replaced the customary hierarchial address with a topic and a "flood route designator". In this case, the subject is WAPR and the flood route is **ALLWI**. You could just as easily send a message to everyone interested in **PACKET@DIST9**, meaning that your message to packet would be distributed throughout the 9 call-area. Once you send the body of the message, you would close it in the same manner that you close a personal message.

Choose your topics (and their abbreviations) with care. You are allowed a mere 6 characters for a topic. Avoid starting it with a number (like 4SALE). Try to make it easily recognizable, and pertitent to the subject at hand. This will make your topic more readily understood, and the message more likely to be read. This covers the very basic information you will need to send packet mail of the most popular types. Next month, we will cover the MSYS Node operation, and the following month we will have a "catch-all" session of odds and ends concerning the finer points of BBS operation.

Using the Wisconsin Network - Part 16

by Andy Nemec, KB9ALN

Last time we traveled our Wisconsin network, our discussion centered on sending a message from a typical BBS you would find on our packet network - the "MSYS" BBS. While this type of BBS provides the usual services associated with any other BBS, it also can be used as a node in many instances. In fact, it is pressed into that service in many parts of our state.

MSYS nodes use the very same computer to operate the BBS as the Node. The call-sign will have a different SSID than the BBS (commonly it will be -5).There is also a bit of a price to be paid for operating a sophisticated BBS and Node on one computer. MSYS Nodes operate a little more slowly than a dedicated network nodes, but they do get the job done.

MSYS BBS's will function as a node in a couple of ways, as non-networked or networked. As a non-networked node, you will find that it operates in much the same way that a **K-Node** operates. If you have a Kantronics TNC, or use one as a local "Wild Node", you pretty much know how it operates. The basic commands of the K-Node style MSYS node will look something like this:

Bye BBS Connect Help Info JustHeard Knode Ports Talk Users

Depending on the way that the BBS and node are configured, you may see more of these commands, or perhaps fewer. This is a typical arrangement. The most obvious differences are shown in the **JustHeard**, **Knode**, **Ports** and **Talk** commands.

The **JustHeard** command will give you a heard list, broken down by port number. For example, a typical JustHeard list might look something like this:

```
On BBS     Port 0     Port 1     Post 2    Port 3     Port 4
W9XBD      KB9XTL     NU9UR      W2XBS     KX9TC      DL0MEN
AZ9ZA
W9XFN
```

Note that there are a couple of odd things going on there. Just as normal network nodes have more than one port, an MSYS node can, too. Notice, too that the BBS is counted as a port. You can usually connect to the BBS from the node by typing BBS when you connect to the node.

Also notice DL0MEN on port 4. Yes, this type of node may have several radios hooked to it, and one of them might very well be an HF radio.

The **Ports** command will give you a listing of just what port does what. For example, you may see a listing like this when you invoke this command:

```
Port 0 is 145.090
Port 1 is 446.375
Port 2 is 145.030
Port 3 is 446.200
Port 4 is 14.105
```

The **Knode** command is similar to the **Nodes** command you will find on networked nodes, but only lists known **Kantronics Knodes.** No network nodes will be shown.

The **Talk** command will connect you to the BBS operator's keyboard, so you can talk to him or her. When you issue this command, you will see this:

Paging SYSOP for 60 seconds.

If the sysop is prepared to answer your call, he or she will break in and converse. If not, you will be returned to the node and will see the prompt:

Node Cmd?

The remainder of the commands are the same as a standard network node, with one exception. The **Connect** command is different because of the number of ports involved. In the example above, we know from issuing the Ports command that Port 2 is 145.030 Mhz. If you wish to connect to W2XBS on that frequency, you would alter the standard connect command to include the port number. It would appear as:

C2 W2XBS

If your station was on port 1, you would use **C1** for the connect command. Networked MSYS nodes operate in the same way as the usual "TheNet" style node. The **Help** command will return the same information as the K-node version, with a couple of additions.

These are **Nodes** and **Routes**. They will give you the same results as "TheNet" type nodes. The **Connect** command shares some of the restrictions of the Knode, unless you are making a connection to another Network Node.

For example, if you find a node that you wish to connect to by using the **Nodes** command, you do not need to know what port it is on - it's automatic. If you intend to connect to a station that is not a node, you still need to add the port number after the **C** (or **Connect**). If you try to make a connection to a network node by using the port number along with the **C**, the MSYS node will make that connection as a standard "AX.25" connection. In other words, the node will behave as though it were a Knode, and you will loose the advantages of the network connection.

When in doubt as to the type of node, you can always issue the **Help** command. MSYS nodes will have a long and thorough help file that will list the valid commands and a short explanation of each. That wraps up our discussion of the node function of the MSYS BBS. Next time, we will do a "wrap-up" of loose ends concerning the BBS and Node operation of the MSYS BBS program.

Using the Wisconsin Network - Part 17

by Andy Nemec, KB9ALN

For the last 3 parts, we have been investigating the use of the MSYS BBS as both a node and a BBS. This time, we will use a Question-and-Answer format to tie up some "loose ends", using the most common questions and the most common-seen errors made by operators using **MSYS**.

Q - Why, when I end a message with a /EX, does it sometimes take 2 tries before I get a "Message Stored" response?

A - This is an easy one. The /EX (or control-Z) has to be sent to the BBS on a fresh line. You may think that every line you type is a "fresh line". Not so. Just because your program looks like it starts a new line after you have typed 80 characters does not mean it adds a carriage return (which is how a computer knows to start a new line of text). Most of the common packet programs (Hostmaster, PackRatt, etc) do not. If your packet program has "word wrap" for your transmitted text, this is no problem. The soloution for those who do not is to hit the "return" or "enter" key just before you send the /EX or Control-Z.

Q - I log onto our local BBS every day. If I connect up today and List the messages, I will see a certain amount. When I connect up tomorrow, the BBS will not show me the messages I saw listed today. Why?

A - The BBS keeps track of every user. Not only does it remember your home BBS and when you last checked in, but it remembers the number of the last message you have seen listed when you last connected and said Bye. The messages are still there, you just have to look for them. If there was a particular message you were interested in and don't remember it's message number, there are a couple of ways to locate it. If you know what catagory the message was under, like "SALE", then you can list the all of the SALE messages. If you remember that the message concerned a TS-450S for sale, then you can have the BBS do a search for you. Just type: L"TS-450" and the BBS will list every message that contains the phrase "TS-450". You may also see unrelated messages that contain this string of letters, so you may have more than one message to choose from.

Q - I try to mail a FOR-SALE bulletin out, but the BBS keeps telling me that the message is undeliverable or something like that.

A - Chances are that you tried to enter the message as "FOR-SALE", which is too many letters for the "@BBS" field. Try typing it as SALE, it should go through then.

Q - When I enter a bulletin into the BBS, the BBS tells me that my message is being held and made invisible for Sysop review. Why?

A - FCC rules state that the originator and the first forwarding station are held responsible for the content of any messages. Just about all Sysops hold any outgoing bulletins so that they can check them for content. Even though your message may be innocent enough, there are some that are not. So, all outgoing bulletins (and sometimes personal mail) are held so that the Sysop can check them. It is nothing personal, it is simply the Sysop protecting his or her license.

Q - I am trying to mail a friend in Texas, but the BBS keeps telling me that that the message cannot be delivered because of a missing or incorrect heirarchial address. I am sure I typed it in right. What gives?

A - Chances are, your friend is at a new BBS, or one that is unknown to your BBS. Send a message to your Sysop and explain the situation to him or her. All that is usually required is to add your friend's BBS into the list of BBS's that yours knows about.

Q - I am not getting mail from my friend in New Mexico. He gets my mail, but I get nothing from him.

A - Contact your Sysop. Sysops are always interested when something like this happens. Chances are, there is a BBS somewhere down the line that does not know how to route some of the mail it receives, yours among them. Sysops can trace where something is getting lost and generally fix it without too many problems. But, they have to know about it first, and are probably not aware of the situation.

Q - When using an MSYS node, I can never connect to a friend of mine. I know his station is up. What is happening here?

A - Chances are, you are not telling the node what port to look for your friend on. The MSYS node defaults to port 0 when it is trying to make a connection for you. The best way to find out what port your friend is on is to check the JustHeard list. If he is on port 2, (let's say that his call-sign is W9XBY) then you use the command: C2 W9XBY (or whatever your friend's call is).

Using the Wisconsin Network - Part 18

by Andy Nemec, KB9ALN

In past editions of Using the Wisconsin Network, we have dealt with all sorts of networked packet radio node stations. This time, we will deal with a station that is not strictly a Network Node, but becomes part of a network in very many areas. This type of station also is capable of doing some nifty stuff, and you, or someone you know may even own one. It is a **KA-Node**, and it is inside of a great many of the Kantronics TNC's that are sold these days. A KA-Node is not a networked node in the strictest sense, but you can make a long-distance connection through a series of them to form a loose network, of sorts. They do not utilize the "Net/Rom" protocol, as Network Nodes do. They are strictly AX.25 nodes.

They are a sophisticated digipeater that operates on the same principle as a node. They repeat your packets, acknowledge the receipt of them, and acknowledge the receipt of a packet intended for your station before the packet is sent to you. In other words, they do not rely on "End-to-End" acknowledgment the way that digipeating does. Therefore, it is much more efficient when used as a node. The KA-Node can be used as a node on your usual 2-meter LAN frequency, and some can even act as a gateway to other frequencies or LANs. Most KA-Nodes have a Secondary Station I.D. of -7, although this can be changed to a different SSID.

They can also have an alias, but this alias is not recognized by network nodes (they don't speak the same "Net/Rom" language). There are two versions of the KA-Node. The single-port version is found on TNC's like the KPC-3, while the dual-port version can be seen in the KAM and 9612 TNC's. The single-port version is easy enough to get familiar with, it resembles a simple network node in many ways. When you connect to one, you will get a greeting and a prompt that looks something like this:

```
###CONNECTED TO WILD NODE WX9APR-7
Welcome to my node, feel free to use it. PBBS is WX9APR-1
ENTER COMMAND: B,C,J,N, or Help ?
```

The **B** command is just like any other node or BBS - use it to do a polite disconnect.

The **C** command tells the node to connect to another station. **C WZ9APR** will tell the node to connect to station WZ9APR. If it is unable to connect to this station, the node will respond with:

```
###RETRIED OUT AT NODE WX9APR###
ENTER COMMAND: B,C,J,N, or Help ?
```

The **J** command is the Heard List, and can be seen in three forms, **J**, **JS**, and **JL**. **JS** is a shorter form heard list, without the extras. **JL** will give a longer heard list, with the time and date the station was heard, along with digipeaters and the type of station it has heard.

The **N** command is also much the same as what one would encounter on a network node. It will list nodes, and it gathers a node list based upon the station ID that it hears from another station. Other KA-Nodes are shown on this node list, as well as any network nodes it can directly monitor on the frequency. KA-Nodes don't show nodes linked by backbone or wire link.

The **Help** command will show all of the valid node commands and their variations. The Dual-Port KA-Node offers one other important feature that makes it all the more useful. There is one addition to the command prompt, and it looks like this:

```
###CONNECTED TO WILD NODE WX9APR-7 (CHANNEL A)
Welcome to my node, feel free to use. PBBS is WX9APR-1
ENTER COMMAND: B,C,J,N,X, or Help ?
```

The **X** command is added, and the **J** Heard list is expanded a bit. And notice that it now tells you that you are connected on "Channel A". This node has two ports, or channels that connections can be made on. They can be configured to be "Channel A" and "Channel B", or "VHF" and "HF", to name a couple. When they are configured in this manner, the node becomes a "Gateway" to another frequency, and sometimes another mode. Using one of these nodes configured as a gateway is fairly easy. If the operator has the gateway enabled, you simply connect to the node and issue the **J** (heard) command. The heard list will indicate whether a given call-sign was heard on channel A, B, VHF or HF with a forward slash and an abbreviation after the call-sign. For example, this heard list shows:

WZ9APR/V 07/04/96 07:04
WY9APR/H 07/04/96 07:03

WZ9APR was heard at 7:04 on July 4th, on VHF. WY9APR was heard on the same date, one minute earlier, on the HF port. This is where the X command comes in. If you wished to connect to WZ9APR, you would use the **C** command. If you wished to connect to WY9APR, you would use the **X** command (Cross-Channel Connect) to make the connection. This tells the TNC to look for the station on the other port (channel). This presents some interesting possibilities.

The other port can be HF or VHF, or can be a different mode - like AMTOR, GTOR, or RTTY. If one of the ports is HF packet on say, 20 Meters, one could carry on a conversation with someone on the other side of the world, DX permitting. This node can also provide a VHF gateway to another VHF LAN. Usually the short greeting you receive when you connect to the node will tell you what port does what job.

Though KA-Nodes are not really part of the Wisconsin Network, there certainly are enough of them. Besides the gateway service they can provide, they are far superior to the practice of digipeating to a network node, should you have difficulty making a direct connection to a node on your LAN frequency. In this manner, they help you to better navigate through the Wisconsin Network.

Using the Wisconsin Network - Part 19

by Andy Nemec, KB9ALN

Did you know that a distant BBS takes Requests? Most BBS's can furnish you with a variety of information, all you have to do is ask for it. Best of all, you don't even have to navigate the network to do it. It can all be done by packet mail from your local BBS, saving you time and giving you the benefit of a local connection when retrieving files and other information. There are 3 information servers that most BBS's in Wisconsin have available.

You can request the contents of a directory, request a file, and on some BBS's, request a callbook look-up. The reply comes back to you at your local BBS as a mail message. The **REQDIR** is a function that will return the contents of a remote BBS's directory. This means that you do not have to connect long-distance to a BBS and go through a long process of manually looking at the directory. It can be done with a simple mail message. All you have to know is the call-sign and heirarchial address of the remote BBS (as well as your own, naturally). Let's say that you want to know what is in the directory of our KA9JAC's BBS in Neenah. Start the message with:

SP REQDIR@KA9JAC.WI.USA.NA @(Your BBS)

The @YourBBS tells the remote BBS to mail the reply to you at your home BBS. Substitute your actual BBS call-sign with this. The BBS will respond with:

"Enter Title or City, State, and Postal Code:"

Keep this line blank. Simply hit the Enter key on your computer. The BBS will respond with:

"Enter Message (^Z or /EX to End, ^A Aborts)"

Leave this blank by sending a Control-Z (or /EX) to end the message. That will start the process. You will get a message by return mail listing the contents of the directory. I did this with the KA9JAC BBS in Neenah, and my reply looked like this:

********** Main File Directory **********

```
ALIGN.MFJ    1603   ARES            BART        2751 CTYHUNT
DX-NOTES           DXDATA.A-E       DXDATA.F-L      DXDATA.M-R
DXDATA.S-Z         FAIL.386     942 FCC            FIELDDAY
FILE1         846  FREQ-STS.74 2225 FREQCORD.FRM 3832 FREQS
GEOGRAPH            HUMOR           ICOM1.MOD        COMSERV
INTERNET.GAT   1351  KDK            KENWOOD1.MOD      KENWOOD2.MOD
KENWOOD3.MOD         KEPS           KNWDSERV         MFJ
MODS               MSYS           NASA.BBS  627   NEWDXA       1148
NODES              PACKET          PACKET.10  408  PACKET.WIS
REQFIL.HLP     602  REQSAT.DOC   1994 SATELITE        USDATA
VIRUS        4235  WALL.DOC     1792 WAPR           WAPRFREQ.APP  3730
WARNING        881  XMAS        1301
```

Along with the list of files, you have a list of subdirectories that also contain files. File names follwed by a number (in the above example, ALIGN.MFJ, BART and FAIL.386 and others like them) are indeed files. Names appearing

without numbers are indeed subdirectories.
Yes, you can also send another message and find the contents of those directories, too. The procedure is almost exactly the same as in the first example, except for the title. Let's say that you want to find the contents of the **WAPR** directory. Use the same format to send a message to **REQDIR**. When the BBS prompts you for a title, send:

/WAPR

After you get the prompt for the body of the message, just send a Control-Z (or /EX) to close the message. You will get, by return mail, a list of the files in the WAPR subdirectory.

Okay, now you have a list of files in front of you. This is nice if you collect lists of directories, but not much good unless you can get one of these files. Of course, you can. That is what the **REQFIL** function does. Let's say that you found a file called WAPR-BL.DOC that interests you. This would be a copy of the WAPR By-Laws, and this file can be mailed to you just as your request for the directory listing was. Using our above example at KA9JAC, you would use this format to send a message:

SP REQFIL@KA9JAC.WI.USA.NA @(YourBBS)

The BBS responds with:

Enter Title, or City, State, and Postal Code:

You now send:

WAPR/WAPR-BL.DOC

Notice that we did not include a / in front of the WAPR. It is not necessary. Just remember that you use the / in front of the directory name to request the contents of a directory, but not when requesting a file. Use a / between the directory name and the actual file name.
The BBS will now ask you for the message. Again, you will leave this blank and simply close the message with a Control-Z or /EX. You will get the file by return mail for you to retrieve at your convenience. A word of caution is in order, though.

Directory listings will show the size of the file that is listed. Keep this in mind when you are retrieving the file. A very large file downloaded during peak hours will almost certainly wreak havoc on your fellow packet operators. If you get a large file returned, take note of the traffic on the frequency and try to avoid congestion. Download it during off-peak hours to keep your LAN a happy place.

The last nifty feature can help you out with your QSL'ing chores. Some BBS's are equipped with a call-book server, and you can remotely take advantage of this. A message to **REQQTH** will do the job quite nicely. Just send:

SP REQQTH@RemoteBBS.Address.WI.USA.NA @(YourBBS)

The BBS responds looking for the title, or City, State and Zip. You simply enter in the call-signs of the stations you wish to look up. Separate each with a space, and due to space limits on the title line, you can't do any more than 5 per **REQQTH** message. An example:

N9BQM KA9JAC KE9LZ

The BBS will again respond asking you for the body of the message. Once again, leave this blank and close the message with a Control-Z or a /EX. You will get a reply with the callbook information for those call-signs, or a message saying that the call was not found. Naturally, not all of the BBS's you encounter may have any or all of these features available. If a particular BBS does not have a particular feature enabled, you will receive a reply informing you of this fact. It's easy enough to use, and just another way you can enjoy Using the Wisconsin Network.

Using the Wisconsin Network - Part 20

by Andy Nemec, KB9ALN

In the last couple of years, packet operators have been able to take advantage of higher speed packet operation at a more reasonable cost than in the past. TNC manufacturers are now accomadating the amateur radio market and have pre-packaged high-speed TNC's available. Although they have become much easier to set up than in the past, there are some things that you should know before you take the plunge into the magic land of 9600 baud operation and buy that new TNC.

First of all, how much good will it really do you? Under ideal conditions, 9600 baud data rates are 8 times faster than the 1200 baud operation that we have become accustomed to. In reality, a lot of factors affect data throughput and your results may vary considerably from the ideal. No question about it, 9600 baud packet operation is faster. In order to see a substantial difference, there are important things you have to deal with. Your radio is the first thing to look at.

What kind of radio do you plan on using? Most garden-variety FM Amateur transceivers are not adequate for the purpose, even the ones marked "9600 Baud Ready". Why? Modern radios use a method of modulation that is incompatible with 9600 baud operation. For narrow-band voice signals, this works acceptably. For the wide-band 9600 baud data signal, this modulation scheme will cause serious distortion of the data signal, making it impossible to decode. Not only that, a lot of these radios have a significant frequency error brand new, "out of the box". Another common affliction of these radios is the time it takes a transmitter to "settle" on frequency, and the time a receiver takes to recover from the transmit mode. These radios are often slow on both counts, causing retries and lost data.

Even if your station can send data fast, receiving it is quite another matter. What can be done? There are 3 alternatives. The first is a multi-mode radio. Why? These radios rely on a different modulation scheme, and they often have faster transmit-to-receive "turnaround time". Therefore, they not only transmit a cleaner data signal, the receiver recovers quickly, and can hear a response from a sent packet. They also feature a more precise tuning readout, which means that it is easier to get them on the proper frequency. The drawback is that these radios are often very expensive.

Another alternative is to buy a radio expressly designed for this purpose. Older Kantronics Data Radios are best avoided - I have heard little good about these radios. One radio that I hear consistently good reports about are the radios made by Tekk. There may be others as well. The important thing to look for is a crystal-controlled radio, or one that has a synthesizer optimized for 9600 baud operation.

The third alternative is to obtain a surplus commercial radio, such as a Motorola or G.E. We have had excellent success with Motorola Mitrek radios, they are crystal controlled, inexpensive on the used market, and very high quality. The disadvantage with getting any second-hand radio is the adjustment required to get it to operate correctly. it takes a little bit of technical skill and equipment to get everything set right. Unless you know a qualified technician, this may not be an option. Even some commercial equipment requires a small amount of modification to get it "just right" for 9600 baud operation.

Now that we have talked about the RF end of things, there is yet another thing to consider. Even if your station is 9600 baud ready, is your area ready for it? In other words, do you have access to a 9600 baud end-user node? In their eagerness to utilize this mode, some operators who do not have access to such a node rush out and buy a TNC, and get it almost working with some kind of radio. They neglect to find out if there is a 9600 baud end-user node and find themselves on a 9600 baud backbone node frequency. I can't stress this point enough to you - *DON'T DO THIS!*.

Why? A lot of these radio/TNC combos are not optimized for the backbone frequency. Remember, the purpose of the backbone is to handle a high volume of packet traffic, and these nodes are optimized for this purpose. They often have very fast response times, and communicate with nodes quite a distance away. This can disrupt the network in two ways.

The first disruption occurs from retries between a poorly set up end user and the local backbone node. The backbone node answers the end user quicker than it can deal with, and a packet has to be repeated very often. This slows data throughput between the end user and the local backbone node, as well as between the backbone nodes. The second problem occurs because of the "**hidden transmitter syndrome**".

Remember that backbone nodes may well be a considerable distance away, and are often located in high places with good paths to the remote node. The end-user station may not be so well blessed; it may be using a marginal antenna, or may not be in a good location. This means that it cannot hear the remote node. Consequently, this end user will transmit at the same time the remote node does. The local backbone node hears both of them, and hears a "double". Neither packet is decoded, and they have to be re-tried. Enough of these can seriously disrupt a network conection, and frustrate the end user. This will spoil it for everybody!

All of this may sound quite discouraging. Actually, it is intended as a guide to help you think carefully about your impending purchase. No doubt, 9600 baud packet operation will become the norm in a very few years. However, a lot of people may get discouraged if they make the wrong choice. What to do? If you don't have a 9600 baud LAN end-user node, thnk about what it would take to have one in your area. It can be done for less than $500, and if enough interested people in your area kick in a few bucks, that is not such a big pile of cash to generate. Node Operators often have enough cash stuck into their systems, and they often pull it out of their own pockets. Making it a community effort will not only make it a whole lot easier, it can help pull a radio club together and give it a project.

If you don't have a 9600 baud LAN frequency, contact the WAPR frequency coordinator and ask for one. He can assign one that is compatible with the bandplan in effect in your area. If you can't put an end-user node on the air quickly, at least you will have a place to park until you can. And it will keep you from interfering with the backbone operation. The bottom line in all of this is "do it right". Not only will you get more enjoyment from this mode of operation, you will be operating in concert with the rest of the Wisconsin Network.

Using the Wisconsin Network - Part 21

by Andy Nemec, KB9ALN

One of Amateur Radio's primary reasons for living is to handle messages on behalf of other people. Sometimes these are mere demonstration messages of the routine variety, some are actually important emergency traffic. The need for relaying messages in an orderly fashion was found by none other than Hiram Percy Maxim himself in 1913. The ARRL and the National Traffic System were born of this need.

As with virtually any mode of Amateur Radio operation, packet can and is used for message handling in the National Traffic System (NTS). Like anything else, there is a right way to do this, and a wrong way. In this edition of our series, we will explore just how to use the Wisconsin Network of BBS's to handle NTS Traffic.

If you know anything about the NTS, you know it uses a specialized format. There are reasons for this - NTS operators want to make certain that the message gets through accurately, and want to know what when wrong when it doesn't. For this reason, it is important that you know how to do this properly, and follow a standard format. Knowing how to properly handle a routine message now will be invaluable if a disaster strikes and you have to handle disaster traffic. All that being said, let's look at how to do it.

First, the usual Send command is now changed a little bit. Use the ST command when sending traffic. The line sent to the BBS includes the Zip code, the @ symbol, then the letters NTS followed by the destination's two-letter state abbreviation.

For example, suppose you are asked to send traffic to me. The person asking you to send the traffic should have a name, address with Zip code, and hopefully a telephone number. In my case, the line you would send to the BBS would look like this:

ST 54304@NTSWI

the BBS responds with

Enter Title or Postal Code

At this point, you send the destination city and recipient's phone number. In my case, this would be:

Green Bay, Wi. 920-555-1212

It is important that you limit the number of carachters to 37 or less. At this point, the BBS will respond with the familiar

Enter Message, End with ^Z, /EX (^A Aborts):

Now, you enter the preamble, the body of the message, and the signature.

The preamble appears on the first line, and includes the number of message that your station has handled from the start of the year, the handling instructions, your call sign, the word check, the town of origination as well as the date and time of origination. Let's say your call-sign is AX9XX, and this is the first message you have handled this year. You sent this on October 31st at 12:01 A.M., and it is a routine message of 10 words. The preamble would appear like this:

NR 1 R HXG AX9XX 10 Anytown, WI 10-31-96 0601Z

The NR 1 is Message number 1 from you station. R is the type of message, in this case Routine. HXG indicates that you should deliver this message without making a toll-call or mailing. 10 is the word check, and you are located in Anytown, Wisconsin. You sent the message at 12:01 A.M. local time on the 10-31-96. Of course, you use UTC time when handling traffic of any kind. This is why our 12:01 A.M. became 0601Z.

This article is not intended to give you a complete tutorial on traffic handling, just how to enter and handle the message via packet radio. If you are not familiar with the contents of a preamble, consult the Radio Amateur's Handbook. The chapter on "Operating a Station" covers this material well. The next line to send to the BBS separates the preamble from the body of the message, it is simply:

-------------------bt--------------------------

Then comes the body of the message. After this is sent, you once again send a "break" like the one that separates the preamble from the body. Then we have the signature and reply instructions, if any. It may appear something like this:

John Waprmember, AX9XX, Anytown, Wi. Reply to:AX9XX@WB9TYT.EN63EB.WI.USA.NA

After this, you send Control-Z (or /EX), and the traffic will be on it's way.

Of course, originating the message is only part of traffic handling. While the BBS forwarding system will handle relaying the message, some person has to deliver it. Delivering traffic can be a lot of fun, especially if the person you are delivering it to is not a ham and you have a chance to educate them a little about ham radio.

It is best to deliver the message without all of the preamble stuff - most non-hams have no idea of what it means. If ARL numbered messages are in the body of the message, don't read them to the recipient as "ARL 63" (or whatever), just interpret the number as text.

Naturally, you have to know if there is any traffic on your BBS if you want to deliver some. Use the LT command to determine if there is. You will get a listing of NTS traffic, or "***None Found***" as a reply. If you do find a piece of traffic and do deliver it, be sure to kill it from the BBS. Otherwise, some other well-meaning ham may find it and deliver it again. Use the format:

KT (message numer)

to delete the message. If the message number was 14523,

KT 14523

would erase the message from the BBS. Handling traffic is fairly easy with packet radio, if you know how. If you need more information, contact your local ARES Emergency Coordinator. They have a lot of good information and can give you some tips on how to handle traffic.

Using the Wisconsin Network - Part 22

by Andy Nemec, KB9ALN

We've all gone through it. Getting Comm-Port settings right, adjusting to the new packet program, and finally, understanding what is truly going on when we communicate by packet radio. It takes time to learn how to navigate through a packet radio network, and still more time to deal with the ins-and-outs of using a BBS. When looked at from a beginner's point of view, it seems like a rather daunting task. As if there wasn't enough complication for new packet operators, there is always the risk of sending out a flood message, only to receive a "flame" message in response.

"Flames" are unkind messages (sometimes downright hostile!) that may alert the originator of some improper or disagreeable content of a message. Ocassionally, they will be misdirected attempts to educate a newer packet radio operator who unknowingly makes a mistake, or violates a regulation. Other times, the sender of the "flame" will misinterpret the content of a message. Ocassionally it will just be an inability to express an opposing opinion in a civil manner. In this part of our series, we will offer some clues as to how to avoid the dreaded "flame" message.

The first step in avoiding a flame message is to try and keep your message simple, legal, and conforming to "good Amateur practice". "Good Amateur practice" is a rather ambiguous term, but perhaps these hints will help you meet this standard, whether you are responding to a message or generating a flood message. Not to mention avoiding the dreaded Flame!

1) Only offer items for sale that have genuine amateur radio use. Avoid CB radios, cars (with or without 2-meter rigs installed in them), or any item that cannot be legally used for Amateur radio. Treat packet radio "For Sale" ads as a big swap net. A net control on a swap net would not allow you to list an automobile on the net, so why would this be permitted on packet radio?

2) Like debating? Keep your contributions civil, thoughtful, and to the point. Don't insult someone for their point of view. Don't say "I think all people who vote for candidate John Q. Politician have holes in their heads!". This is only asking for a flamed response. The surest way to turn your audience off is to insult them! This may sound like common sense (and it really is), but is so easy to forget.

3) Don't inadvertently advertise for anybody, this includes yourself. Although it is perfectly legal and commonplace to have your company name and position on Internet E-Mail, it may be construed as a subtle advertisement when done on packet radio.

4) Keep your signature file short. A long signature file with cute pictures, American flags, and other unnecessary information will only serve to alienate someone. If you do send a signature file, make sure you do not have an excess number of blank lines in it. A good signature file will have useful information in it - such as how to respond via packet or Internet E-Mail. A bad signature file might have a large American Flag with the words "Vote Straight --------" (fill in the party of your choice!) in it.

5) Avoid using Upper-Case lettered words to emphasize a point. **THIS** is considered **SHOUTING** and **IMPOLITE**. (See how easy it is to be irritated by this?)

6) Consider whether your message will be of interest to others. If you are interested in say, astronomy, you may find an interested party or two to converse with. If you are looking to correspond with someone regarding 13th century romance languages, you may have a much tougher time. Consider sending a CQ message first - Don't send a bunch of messages that few are likely to respond to.

7) On the subject of CQ messages, it makes no sense to send a CQ@WW message when you want to make a packet contact to Arizona. If you are targeting a particular state, use the @ symbol followed by the two-letter state abbreviation. Most BBS's are set up to forward based on the state abbreviation, and you will be able to target an area rather than needlessly flood the country (or the world) with a message intended for limited distribution.

8) Keep your message content in good taste. Long, graphic descriptions of invasive medical procedures are not only unnecessary, but in poor taste. Remember that you cannot always target your audience - someone may read such a message out of curiousity. And not all messages (or conversation) are suitable for all people.

These are a few hints to help you avoid the dreaded flame. Believe it or not, I have seen messages that are contrary to one or more of these suggestions, and recently.

There is one more that you should take to heart, though. Use the old "tincture of time" prescription to help you avoid flame messages. When you read a message that you disagree with, or find really offensive, save it to a file while you are reading it. Then let it sit for a day. Re-read it and try to determine if the words used actually mean what they seem to mean. Remember, humans make mistakes, hit the wrong keys, and choose the wrong words. Someone may have had a bad day, and may have taken it out on their keyboard.

Once you have saved this message, re-read it, and decided to respond to it, it will be easy to intelligently debate. Simply edit the file with a text editor and remove the lines in the original message. Add your own, and you have a reply. If you need to quote the originator, you can add a > sign in front of each line. Some packet programs include this feature, so this may not be necessary. Once you see the original text, you are not caught trying to remember what was said by the originator. Once you have typed a reply, let it sit for a day or two. Nothing is so important that it can't wait a day. Some packet debates take place over the course of weeks, or even months. Then look at what you have typed to see if it is really what you meant to say. This will pave the way for a civil response. Once you are satisfied with the way it is worded, use the upload feature of your packet program to send it to the BBS.

After all is said and done, you may still get a flame message. Some folks do not always interpret what you say in the same way that you might. Some folks just like to flame others. If this is the case, simply follow the above steps (saving the message to a file, re-reading it, and then editing a response) to try and keep from falling into the same trap. These suggestions will not guarantee a "Zero-Flame-Zone" on your local BBS, but it sure can help to make it a little better. And the fewer flames that are forwarded by the BBS network, the better Amateur Radio is for all of us.

Using the Wisconsin Network - Part 23

by Andy Nemec, KB9ALN

All of us know that technology changes, just like the rest of life. After all, it was not all that long ago that packet radio was a brand-new mode, and more of a curiosity than useful. Not only have various aspects of packet radio operation been improved, there are promising innovations just waiting for the right opportunity to become practical and useful. So far, we have only scratched the surface and have seen a more of an improvement of packet radio rather than a revoloution. This improvement is reflected by the changes we see in the network here in Wisconsin.

A few years ago for example, TCP/IP operation was limited to one experimental frequency. Improvements in the programs used and the methods used to route TCP/IP "packets" made it possible to "mainstream" this mode. A good many of the network nodes in Wisconsin are now capable of passing TCP/IP traffic. This only makes sense considering that it is the protocol set used for the Internet. Although our Amateur packet network differs from the Internet, there is a great deal we have in common. It all boils down to computers exchanging information. Whether it is a keyboard chat or sending files to and fro, the common element is this exchange of information.

Right now our network is evolving from the use of two protocols - methods of transporting information. Why would we need other protocols? Think of a protocol as a vehicle used to transport information. The information comes in various types. Think of this information as items that need to be transported. You would never try to transport firewood through the Northwoods with a Cadillac Eldorado, would you? It is not suitable for the purpose. Likewise, you probably wouldn't want to take a cross-country trip in a logging truck. The Cadillac is better suited for this purpose.

The same can be said of protocols. Using AX.25 and Net/Rom will not fulfill the potential that packet radio has - they are not suitable for performing the advanced work we would like to do. The bulk of the information transported from one Amateur packet radio station to another is passed in much the same way it was 5 years ago (a long time in the computer world!). If we really want to bring packet radio into the "information age" and find new uses for it, we will have to look at the technology used by the Internet for inspiriation. While we don't want to mimic every aspect of the Internet, we can adapt, rework, and refine the technology so that we can use some of it with radio. In the process, we will can come upon innovations unique to Amateur digital radio networking.

Innovation is, of course, a natural part of Amateur Radio. Change does not come easy to our packet radio system. Let's stop and think of why things have not changed significantly in the last 5 years. One reason may be that some TNC manufacturers and Amateur Radio equipment vendors find the status quo a comfortable place to be. It makes them money (not a bad thing in and of itself) by rehashing the same old technology. Not to mention the fact that we have not pushed for advanced technology. Until recently, our eyes have not really been opened to new, exciting possibilities. There has been no percieved need to change things, so there has been no effort expended toward innovation. If we are continue to keep packet radio exciting, we will need increased capability in this mode.

Presently, Amateur packet radio is limited to being a mail service, an occasional way to chat, and a way to transfer small text files. Binary files can be difficult because there is no standard transfer protocol that everyone uses. One does not get a graphical user interface and other conveniences like search "engines" that we find on the Internet. If we want to expand the capabilities of packet radio, we may very well have to adjust to new ways of networking - and using the network.

We are doing well getting the Wisconsin network constructed, better than a lot of states. 9600 baud nodes are fast becoming part of the network, and some are even cropping up as end-user LAN nodes. One of the more important steps that WAPR took was recommending the installation of TheNet X-1J node firmware. It's expanded AX.25 and Net/Rom capabilities, in addition to the ability to route TCP/IP, certainly are a great step forward. While we are in a fairly good position to step closer to the "New Age" of packet radio, there is considerable that remains to be done, and new possibilities to explore.

And that is what the next few parts of this series concerns, opening our eyes to new possibilities. We will see new uses for packet radio, and new methods to transport information through a network. We will see how other parts of this country are constructing their packet radio networks and what hams in other countries are doing with their systems. We will discover just what "more" is, how we may get there, and how to utilize it when it does arrive. After we see what is possible, we will explore a few ways to make your current packet station a little close to the "new age". Hope you find this direction interesting.

Using the Wisconsin Network - Part 24

by Andy Nemec, KB9ALN

In the last installment of the series, we talked about how our packet radio network has been evolving, and talked about the need to look toward the future. Many of us have tasted the advanced nature of the internet, and have wondered if we could do some of what we see there in Amateur Packet Radio. This time, we will look at the elements of a network and why we might consider not only other methods of networking, but of operating as well. But in order to do that, we have to analyze the whole networking process and look at what we do with it.

We all know that a network allows us connectivity with other stations. To aid our analysis, consider what happens when you want to leave a message to a friend of yours who lives some distance away from you. If you are in a typical packet radio situation, you may have 2 or 3 ways to do this:

1) You can connect to your local BBS and type in a message (if you both have access to a BBS). While other stations do the work, you have to remember his BBS address, you have no time to correct spelling errors and word your message properly.

2) You can connect up to his mailbox and leave a message there. Of course, you would have to navigate through a series of nodes to get to his mailbox. And if his station is not active, you have wasted your time. If you have a shaky path to any node in the circuit, you may suffer sudden disconnection and your message is lost in the ether. You will have to try another time. Again, you have no editing capabilities.

3) You could also leave a message in your mailbox, or another friend's mailbox. Of course, your mail recipient must remember to check for the mail (just like the BBS example above). If you choose to leave a message in your friend's mailbox, you also have to remember node aliases, the best route to take, and might have to wait for the right time of day to make a connection.

To sum it up, the process looks like this:

1) Connect to your local LAN node.

2) Gain access to the network.

3) Know the alias of his LAN node.

4) Deal with any nodes on the backbone making shaky connections.

5) Segment your connection circuit if there are any shaky paths.

6) Connect to his LAN node.

7) Connect to his mailbox and leave a message.

This is a heck of a lot of thought and human intervention required for a system that is run by computers! In contrast, consider what happens if you E-Mail one of your friends on the Internet:

1) You dial-in to an Internet service provider, and leave a message.

2) Your friend dials into his internet service provider and picks up his message.

It can be made even simpler. Some folks have their computers set up to automatically connect up to the service provider to send and collect E-Mail. Wouldn't it be nice to do that on Amateur Packet Radio?

Well, it is feasible, and many people do it right now with packet radio E-Mail. The secret is to have an advanced mail system and an open, transparent network at your disposal, as the Internet is. What do we mean by an open, transparent network?

Consider all of the stuff that goes on behind-the-scenes when making any kind of connection on the Internet, for any purpose. An excellent example of this is what happens when a person is "Web Surfing". When a person connects up to a "Web Site", he or she has no knowledge of how the data gets to or from this site. All they have to know is "//www.bogus.com", an internet address. This is a transparent connection, computers that do the routing and relay of the data are not seen by the user.

In a global computer network, there is likely to be a significant amount of network activity just to allow someone to connect up to this web site. This network is also open to a number of different protocols. The protocol used in this case is http, used for transferring the hypertext present in most web pages.

Contrast this with what you had to go through to get a message to your friend. There is a lot to know, and a lot to do. You have to know details about your friend's station and the network that the average Internet user doesn't ever worry about. So in order to make our mail system more "automatic", we need the following :

1) A way to send mail with maximum convenience, preferably without ever "leaving our own station".

2) A mail system that knows who the recipient of the mail is.

3) Mail systems that know how to work with a network.

4) An open network - capable of handling a protocol used by a "smart" mail system that works with the network.

5) Network transparency - packets of any kind are routed to their destination without human intervention.

6) Some method of receiving the mail and possibly storing it.

7) Ability to collect the mail in a convenient fashion.

Amateur Packet Radio has tried to accomplish this in a variety of ways. It has always been the vision of packeteers to have "door-to-door" mail delivery. While we will explore ways to get closer to this with our existing situation, we will not see mail fully automated with what we are using now. Although we would like to think of our current network node system as "auto-routing", it really is not always very efficient when we ask it to do this. In addition, some of our network nodes will not pass data that is sent using protocols other than Net/Rom and AX.25 (remember our discussion of how some protocols are not well suited for some tasks, but prefectly to others).

I have used mail as an example of one process that might be able to use a different protocol. There are other packet radio functions that may well be able to use protocols different that what we are using now. When discussing a digital radio network, we can further break a network system down to a few basic functions we will discuss in the next installment. These can be described as:

1) A originator of data.

2) A Network Interface Host Computer (comparable to a network node we now use, but more functional and automatic).

3) A transparent Auto-Routing network requiring no user knowlege or intervention.

4) Another Network Interface Host Computer at the destination.

5) A receiver of data. Actually, origination and reception of data can be combined into the function of Network Interface Host Computers, further eliminating two of these functions.

Next time we will dissect the functions of each of these parts of our radio network.

Using the Wisconsin Network - Part 25

by Andy Nemec, KB9ALN

In the last part of this series, we took a look at the future of Amateur Packet Radio, and what we can do with it. We took note of the evolution of the packet network, and what other changes we might want to see in it's continuing development.

We drew some parallels with the Internet, and how we would draw on some of the networking advances that have been made there, perhaps applying them to Amateur Radio. When we left the last part of our series, we broke the network system down to a few basic functions like these:

1) An originator of data.

2) A Network Interface Host computer (Similar to what we use now, only more functional and automatic).

3) An transparent Auto-Routing network requiring no user knowlege or intervention.

4) Another Network Interface Host Computer at the destination.

5) A receiver of data.

Of course, the originator of data would be your, or any other Amateur Packet Radio station. We mentioned that this station could be combined with a Network Interface Host Computer, more on that later.

The Network Interface Host Computer can be compared to a current node in some ways, but capable of being more "network friendly" than our current nodes. This, as well as the network itself, deserves the most study and contemplation. Just what would this computer do? Aside from storing and auto-delivering your personal mail, it would route packets properly, maintain a table of reachable destinations, and know the best way to reach a destination. It could act as a gateway to the Internet, it would allow use of different protocols to accomplish different tasks. In effect, it could replace both the BBS and the nodestack you are currently using, and allow you to access the "World-Wide Web" with a text browser (if data rates can be increased sufficiently).

The network itself would consist of interconnected host computers. They would exchange data concerning routing, as well as acting as relay stations and routers to make certain any data it handles is effectively sent to it's destination. While some of this sounds familiar, the methods that these computers use to communicate would be different and far more varied than they are now.

Currently, only a portion of the network will allow anything other than AX.25 and the Net/Rom (node) protocols to be passed. This is the primary difference between what we are using now and what we could use in the future. In the past 3 years or so, we in Green Bay have been doing considerable experimentaion with TCP/IP protocols - the protocols used on the internet. Currently, we have 10 TCP/IP stations in full-time operation, with one or two ocasionally operating.

In addition, we have a node stack that is equipped with the X-1J4 node firmware, which allows both conventional and TCP/IP protocols to be passed through it. While the X-1J node does not have all of the features of the Network Host Interface Computer described earlier, it is a mid-way point that allows some of the advanced features we may be looking for. Currently, the only real network host computers we have are the TCP/IP stations themselves. These stations access the network through the nodes, and work with the network.

Here is a sampling of what we have been able to do with these stations:

1) Mail a freind without ever having to manually connect up to a node or a mailbox. This is done with the use of "SMTP", the "Simple Mail Transfer Protocol". While SMTP can directly deliver mail to a recipient, it's operation can

be modified to store it at a different computer - a different station.

2) If mail is left at another station, it can be automatically collected from that station using the "Post Office Protocol" (POP). Stations are configured to automatically "POP" the nominated "POP Mail Server" and collect mail. These operators don't have to remember to check into their local BBS - their station automatically remembers for them.

3) Transfer text and binary files to other TCP/IP stations with ease. This is the "File Transfer Protocol" (FTP) at work. There is no need to fool with YAPP to transfer binary files, binaries are simply identified as such and the system automatically accomodates this.

4) Test a radio path by requesting an Echo. This is called "Pinging" a station. The Amateur Radio implemetation will tell you how long it takes to get an echo return, letting you judge how busy the network is.

5) Find out who somebody is without taking a lot of time to do so. This is the "Finger", kind of like tapping someone on the shoulder and asking "Who are you?" A short custom information file is returned with the answer.

There are even more possibilities that we can talk about, and more yet to explore. We have talked about this before in an earlier installment of this series, but needed to repeat this here to show the versatility of the TCP/IP protocol set.

So now we come to the next step in the evoloution of the network as it pertains to Amateur Packet Radio. The Network Interface Host Computer would perform a lot of the functions that current TCP/IP stations would perform, freeing their resources for other things. POP Mail and the SMTP system - automatic mail - could free things up additionally. And now with the intense interest in the World-Wide Web, we could also utilize that for Amateur Radio.

Now let's talk about the other things we will have to consider for the Amateur Radio to effectively use this protocol set. First we need more user-friendly for the occasional user of this powerful system. Programs could be stand-alone DOS versions, or based on the Windows operating platform. Secondly, we need to resolve one issue that is important to making this sytem widespread.

Currently, all TCP/IP stations are numbered, kind of like a serial number. For example, my IP address 44.92.20.9. All Amateur Radio network stations begin with the 44 series of numbers. Obviously, there are a limited number of TCP/IP addresses available, and assignment of the number varies according to location. Therefore, widespread use and mobility becomes another complication.

Then there is overcoming the wide-area reliable pathing and routing of data. Right now we rely on the Net/Rom system to do this. Although we have maybe gotten used to some of it's shortcomings, it is not exactly reliable or automatic. TCP/IP routing principles are borne of a wired network that is not subject to propagation variables. So simply adapting all of the routing techniques the Internet uses are not effective. There have been some attempts at doing this, and progress is being made.

Next time we will further explore how a Network Interface Host Computer system can overcome some of these problems, and what to look for in the future.

Using the Wisconsin Network - Part 26

by Andy Nemec, KB9ALN

In recent installments of this series, we have been looking toward the future, and investigating what the networks of the future might look like. We broke the whole process of leaving a message to a friend down to a step-by-step procedure, and talked about how each part of the network and you interact. One thing we discovered was that packet radio messaging has one comparison - the internet.

We also discovered that the protocol set used by the internet, TCP/IP is in wide use by a number of amateur stations. While the TCP/IP amateur implementation does not exactly duplicate the internet, it is very close, with some accommodation of our unique radio environment. The main attraction of the internet is the simplicity, and the capabilities presented by an open, transparent network. We also talked of the Network Interface Host Computer as being a gateway for users to the network. Now some of you might be thinking "Geez, now he wants me to learn some other program and run TCP/IP". Well, not exactly.

However, the network of the future may very well include many elements of the internet, adapted for Amateur Radio use. Your station may become a vital part of a TCP/IP network, if you choose, or you could remain with your current station and still utilize the services of the Network Interface Host Computer. This is similar to what we now use, the node stack. Instead of a conventional node stack, the host computer would perform routing and the conversion of the TCP/IP protocols to/from your current AX.25 protocol. Even though you use AX.25, you would still be able to take advantage of some advanced features of this type of arrangement. In the final analysis, you could have a greatly enhanced packet station with little changes in your actual hardware or software, if you so desire. The main difference would be in learning new terminology and new functionality.

We have covered most of the differences between what we now use and what we could use. And we are not quite ready to make the big jump into radio "cyberspace". But we are close. I mentioned some of the things that need to be added or improved to make these advances a reality. Here is a rundown of what is needed.

First we need to improve the AX.25 protocol if we are going to continue to use it. There are "spaces" in the AX.25 protocol that allow for expansion without scrapping the entire system. One such extension needs to be the addition of forward error correction. Remember that our current protocol will not correct errors, only detect them and allow for a corrupted packet to be resent. Some packet operators have used AMTOR in the past, and recall that it uses one of two modes. One is "FEC" (Forward Error Correction) and the other is "ARQ" (Automatic Repeat Request). Right now, packet "gurus" are working on this very problem, improving the protocol so that AX.25 can be made more reliable. AX.25 is really an adaptation of the X.25 protocol originated by the telephone industry, and is not well suited for the kinds of things we are trying to do. At the same time, we do need to have the capability to use the existing protocol so that all the current TNC's will still be useable. This is what we know as "Backward Compatibility".

The next need is the ability to maintain reliable routing tables amongst network hosts using TCP/IP. There is one promising protocol called "**RSPF**" (Radio Shortest Path First). It needs to be refined and perfected before it can see wide implemetation. Once done, it has the prospect of becoming far more effective than the current **Net/Rom** system. RSPF actually measures the time it takes to get to a given destination, rather than assuming that a route broadcast is good.

One other thing we touched on was the need for programs that are "User Friendly" and interact with the network in a transparent manner. Microsoft's Windows seems to be a very popular user interface to a computer, and it also has some TCP/IP capability available to it. This is, potentially, a good situation for the software writer - adapt what is there without starting from "square one". This is a good way to incorporate new AX.25 protocol extensions too, with the ability closely control the TNC. Most TNC's have the capability to operate in the "KISS" mode. This allows the computer connected to it to perform most of the processing of packets, making it easier to utilize protocol extensions.

Of course we would not want to leave DOS users in the dust, work would have to be done there as well. It should be possible to use a good many of these advanced function on a relatively simple computer. I (and others) have run

TCP/IP programs on 8 MHz 8088 computers with V-20 processors installed. All of this was done with MS-DOS 3.3 as an operating system.

To sum it all up, we can run advanced systems without spending a huge amount of money for hardware upgrades.

Working hand-in-hand with the software approach is the ability to assign temporary IP addresses. You recall from our last part of the series that there are a limited number of IP addresses availble. Anyone who has used a **PPP** Internet service has used a form of this. PPP is the internet **Point-to-Point Protocol.**
When you use Netscape, you are very likely using this protocol and may not even know it! Therefore, if one were to decide to take advantage of the more exotic aspects of a new network, the Network Interface Host Computer should be able to accommodate this. Currently, this is being talked about as a solution to Mobile/Portable TCP/IP packet radio operation. And it would allow for easier setup of the end-user program.

We also need to maintain a good, high-speed radio network. Maintenance not only involves constructing a network and making sure all the radios and devices function. It also keeping the databases and host configurations up to date. WAPR helps to build and maintain the Wisconsin network, but it always is a struggle in a volunteer network such as we have. Skilled people who understand networking are in demand for both the radio and "software" types of work.

The last thing we need to move to the "next step" is the willingness to try new things. Obviously, we can't experiment with a whole statewide network all at once, but we can experiment locally and regionally. And we can spend more time reading about and learning new ways of connecting our computers to share information. At the same time, we can't make what is in place obsolete. Many people have too much invested in their current hardware and do not wish to change from their current operating setup. It suits their needs, and they are comfortable with it. So we need to improve the system without "throwing the baby out with the bathwater".

And that is the next direction we will take in this series. Next time, we will look at what you can do to make your packet life a little easier - automated - with little or no change in hardware or software, just configuration.

Using the Wisconsin Network - Part 27

by Andy Nemec, KB9ALN

Lately we've been looking at several enticing possibilities of automating packet radio. Most operators don't realize it, but there is a fair amount of automation of the mail system built right into most every TNC these days. It was the vision of the first packeteers to have "door-to-door" mail service with packet radio in the very beginning. In this installment, we will explain how your local BBS and your station can work together to make this door-to-door delivery a reality. Yes, we want your TNC to become part of the BBS Network and interact automatically with your local BBS so that you can enjoy a more automatic packet E-mail service.

The concepts used in BBS mail forwarding might be a little foreign to most people, and one can easily "upturn the apple cart" if you do not know how the system works. For that reason, we will explore a bit of how BBS mail gets from point A to point B. This will help you to understand how your TNC will interact with the BBS network, and help stop problems before they start.

First, we'll talk about how a BBS gets packet mail and flood messages, and what it does with these messages once it has them. Notice that you may see messages from all over the country, maybe even all over the world on your local BBS. They arrive through the **store-and-forward** system. This system is simple enough in concept. Messages are stored on a BBS, and sent to the next BBS down the line in a large group or batch. One BBS will automatically connect to another to establish a **forwarding session** when messages are exchanged. When the sending BBS is through with it's work, the it asks "Do you have anything for me?". If so, messages are **reverse-forwarded** to the originator of the forwarding session.

Messages are stored on the BBS, and the next BBS down the line does the same thing until it gets to the last BBS in the line. While this seems simple enough, there are a large number of packet radio BBS's present world-wide and a large number of paths to get to a BBS. This means that there is a possibility of a message reaching a BBS through mulitiple forwarding routes. How does your local BBS know not to accept a duplicate message?

Each message has a unique **Message Identification Number** called a **MID**, or a **Bulletin Identifier** called a **BID**. This is sort of a unique serial number that accompanies a message. It is generated by the originating BBS or TNC when the message is created. Your local BBS keeps a list of these BIDs and compares what the BBS is proposed to receive during a forwarding session to what it already has. If it sees that unique BID on a message it already has, it refuses that message and asks for the next one in the batch to be forwarded.

If you've been a regular reader of this series, you already know about the hierarchical addressing scheme that BBS's use world-wide. If you do not, please consult **Part 15** of "Using the Wisconsin Network" before proceeding. This is very important, and cannot be overlooked. If you're not familiar with the casual operation of your TNC's mailbox, and how to command your TNC, stop right here and either read up on this, or obtain the help of an experienced operator. Mistakes made here can show up world-wide, and may cause you to be the recipient of some unfriendly advice.

Once you are at the stage where you are comfortable with these concepts, you can proceed. You will also need the following to properly set up your TNC:

1) The permission and participation of your local BBS Sysop.

2) To know if your TNC's mailbox supports forwarding and reverse forwarding.

3) Knowledge of the size of your TNC's mailbox. How much can it hold?

4) Your packet station to be "on the air" when you expect to participate in a forwarding session.

5) Knowlege of the times that your LAN frequency is least busy.

6) The call-sign of the BBS you intend to exchange mail with. This should be your "home" BBS.

7) The path you use to connect to it, and the command you use to connect to your BBS through a node, if you use one when accessing your local BBS.

8) Knowledge of your home BBS's hierarchical address.

Now the "why" of all of this.

You need the permission and participation of your local BBS Sysop because he or she has to set up the BBS both originate a forwarding session and to accept a proposed forwarding session from your station. Keep in mind that some BBS's are set up with limited capabilities and a big upgrade may be in order to accommodate your request.

Obviously, you need to know if your packet TNC can forward. Most will accept a reverse forward. You can often find information concerning this in the "PBBS Sysop" section of your TNC's commands manual.

You need to know how big your mailbox size is in bytes because this has a direct bearing on how much incoming mail you can receive. You can usually find this out by logging onto your mailbox when it is empty. It will tell you how many bytes are "free". Keep in mind that a typical typewritten page formatted at 80 columns by 54 lines is between 2000 and 3000 bytes. If you expect to accept mail in any quantity, make sure your TNC's mailbox is sufficiently large to accept what you want to get. Remember to keep your mailbox clean of old messages. Otherwise it will take multiple forwarding sessions to get your messages. This also has a direct bearing on whether you will tell your TNC to kill off any messages that have been forwarded.

Likewise, you would expect to keep your packet station available for BBS connects in order to receive your mail. If yours is a part-time packet station, perhaps you should consider leaving the messages on the BBS for manual retrieval.

You also would want to keep mail forwarding restricted to the times when your LAN frequency is least congested. Involved forwarding sessions can reduce someone else's packet throughput. Your BBS Sysop may also have forwarding to other BBS's scheduled for a particular time. Consult your BBS Sysop when deciding what time to forward.

Obviously, you need to know the call-sign of the BBS and how to get to it. If you use a node to get to the BBS, you have to know just what text to send to the node in order to get to the BBS. Most TNC's that accommodate node connections to a BBS for forwarding can only utilize one node connection. If your BBS is some distance away from you and requires multiple node connections to reach it, you might best use a BBS closer to you. If you connect to your local BBS directly or through just one node, then you are a good candidate to interact with the BBS network.

It is best to have your home BBS the same as the one you intend to forward with. Otherwise, things may get confusing for respondents of your messages. Therefore, you may get your messages out, but not back. This may also leave some poor BBS Sysop down the line wondering what to do with your mail. He may not know how to get it to you.

The last item you will need to know is the correct, complete hierarchical address of your home BBS. This is entered into your outgoing messages so that people may respond to your messages correctly. The BBS network may know very little about your mailbox-in-a-TNC, but it will know how to forward to your home BBS. It is your home BBS that knows enough to pass a message along to you.

In the next few installments, we will look at some of the most common TNC's and how to specifically set them up for door-to-door mail service.

Using the Wisconsin Network - Part 28

by Andy Nemec, KB9ALN

Last time we started a general discussion of "door-to-door" mail delivery. It is possible, with a little configuration, to have your local BBS deliver mail right to your TNC, without having to check into the BBS itself. And it is possible, with a lot of TNC's, to send outgoing mail directly into the BBS. This time, we will talk about configuring one specific TNC, the Paccomm Tiny-2.

While the command structure for this TNC is identical to other **TAPR-2** clones, there are subtle differences seen when it comes to setting up the mailbox so that it does what we want it to do. Just a word of caution before we begin. You will need the cooperation of your BBS's Sysop in order to do this. You should also re-read the cautions that appear in part 27. This is very important - improper operating can cause problems on your LAN. We will presume you have done this by now, and we can investigate what this TNC can do.

In addition to accepting your incoming mail, the Tiny-2 can forward personal messages to your local BBS so that it can pass them through the network to their destination. It will only forward messages that are specifically designated (**marked**) for forwarding. There is a provision for automatically marking messages for forwarding, though. In the Tny-2, forwarding is supposed to only take place during a "reverse forward" request. In order for your TNC to get this request, it must first be the recipient of a forwarding session from your BBS. In other words, your outgoing mail will only go out to your BBS if you have received mail from your BBS.

There is a provision for forwarding to one specific mailbox/BBS, so you can "beat the system" on this. We will cover all of the pertinent commands alphabetically, so that you can follow along in your manual easily.

AUTOFWD ON

This automatically marks messages for forwarding.

CLKSET OFF

This will not allow a BBS to update your TNC clock. You may use local time, the BBS UTC.

FORWARD

This is used to manually mark messages for forwarding. No need for this, **AUTOFWD** is **ON**.

FNPMS (call-sign)

The call-sign refers to the call-sign (not alias) of the node you will use to connect to your BBS. Only one is allowed. You may also use a digipeater (not recommended) after the node call.

FPMS (BBS Call-sign)

This will be the call-sign of the BBS you will exchange mail with.

HOMEBBS (BBS Call with hierachial address)

This is your home BBS. Although there is a provision for an SSID, SSID's are not used in the hierarchical BBS address scheme and are not used here.

KILLONFWD ON

Will delete any message from your mailbox that has been sent out of your mailbox and into the BBS.This will help keep the mailbox clear so that it can accept new messages.

NODETEXT (Text String)

This is the command you send to any node used to connect to your BBS. The TNC will use this the same way you do. If you connect to node WZ9APR-1 to get to the BBS WZ9APR-5, you send the node this command:

C WZ9APR-5

Your TNC has to use the same path, so the text string is **C WZ9APR-5.**

In addition to these commands, there are others that are optional and may want to change. These are:

3RDPARTY (ON) or (OFF)

This allows other people to use (and possibly abuse) your mailbox. If it is off, only select people are allowed use of your mailbox.

MSGHDR (Text String)

This allows a line with "**From (Some call-sign)**" to appear at the top of outgoig messages. This is a preference matter, but note that BBS messages normally have this data already.

MSGROUTE (ON) or (OFF)

This will allow the forwarding path to be displayed in the message.

RFNPMS (Node call-sign)

This is the node call-sign used to get to the BBS. It is similar to **FNPMS**, but is used only during forced (manual) "reverse forwarding".

RFPMS (ON) or (OFF)

This forces a **reverse forward**. It is an immediate-mode operator command, and asks your TNC to connect to a BBS to see if there is any mail waiting for it.

TKILLOK (ON) or (OFF)

Allows any **NTS** traffic originating at your station to be killed by anyone connecting to your mailbox. Of course, you should be familiar with the usual operating aspects of your TNC's mailbox before getting into this seriously.

And by all means, be sure to follow your BBS sysop's advice when setting these parameters (even if they differ from mine!) because he or she knows their system best. And after all, this is a cooperative effort between you and the sysop. And that is what it takes to have a truly automatic mail system - human cooperation.

Using the Wisconsin Network - Part 29

by Andy Nemec, KB9ALN

In the last two installments of our series, we have been discussing how to automate our packet mail system. You will recall that we discussed the general concepts at first, then looked at specific commands used in the set up a of a Paccomm Tiny-2 for this service. In this installment, we discuss how to set up another popular TNC model, the Kantronics KPC-3, for automated mail delivery and pick-up.

Users of other Kantronics TNCs may find that a lot of these commands are similar, if not the same. Do note that this TNC has been around a while, and there are a number of firmware revisions since the first one rolled out of the factory doors. For that reason, you should also double-check these commands against the ones in your manual. The commands I used are from a fairly recent firmware version of this TNC, version 6.0. Some older models may have a slightly different command set, and not all are capable of every feature. Of course, if you have read the last two parts of this series, you no doubt remember that you will have to consult your BBS sysop when configuring your station. Also read the other cautions in part 27 of this series. Now on to the meat of things.

You found out that setting up the Tiny-2 is fairly easy when it comes to your forwarding parameters. The KPC-3 on the other hand, requires a little more time and effort to set up. In exchange for this effort you get a few extra features (and some of them you may very well use). One special word of warning is required for KPC-3 users. This TNC has precious little memory available for the PBBS to start with, and certain features enabled, even less. You will have to perform a delicate balancing act with TNC memory management or live without certain features.

Perhaps the most prominent, and sometimes unused feature of this TNC is the **KA-Node**. In areas not served by a local network node, or distant from it may find this feature valuable. If you find that you are maintaining a KA-node for a user group of any size, you may want to re-evaluate your situation and change your strategy. Each KA-Node connection circuit requires 4K of memory, with a maximum of 6 circuits. If you use all 6, you have no mailbox to deal with at all.

Another feature requiring memory usage is the Remote access feature. This allows you to remotely manage your mailbox and TNC from another station. If you clear the **MYREMOTE** call-sign and **RTEXT**, you will not have to worry about this feature using memory. Of course, then you have no remote access.

All of this changes if you have additional memory (up to 512K) installed in the TNC. That is one option you have if you find that you are providing a lot of users with PBBS and KA-Node services. With all of that being said, let's look at the commands you will become familiar with:

HTEXT (text string)

This is your hierarchical address. Kantronics recommends that you set this to only part of your hierarchial address. For example, if your home BBS is **WX9APR.#CWI.WI.USA.NOAM** then enter **#CWI.WI.USA.NOAM**.

MYPbbs (call-sign with SSID)

This is the call-sign of your mailbox, usually your call with a -1 after it.

PBbs (5 or more)

This allocates 1K blocks of memory to the Mailbox in the TNC. If you have a memory expansion kit in your TNC, you may set this much higher. The default is 5, meaning 5K, which is about 2 printed pages' worth.

PBForward (call-sign) (Every) or (After) (time interval)

Usage requires you specify a call-sign of the BBS used in forwarding, and a the word **EVERY** or **AFTER** and a time in hours. This tells the TNC to try a forward session to the BBS at a certain time interval. You might wish to keep this set to every 3 hours or so to avoid tying up your LAN frequency.

PBHOld ON

Now you get a taste of BBS Sysop-ing. This sets the TNC to hold all messages so that you can review them before they are forwarded. FCC Rules hold the originator and first forwarder (this is ***YOU!***) responsible for message content.

PBRevers ON

This allows your TNC to request a collection of your mail right after it has finished forwarding your outgoing mail.

The following commands can be set differently than the recommendations with care. They will depend on your operating habits, but you will still have to look at them.

MAXUSERS 6

Sets the maximum number of users of your station to 6 (5 connections plus a Mailbox user). If you find you rarely type to more than one person at a time, you might try 3.

USERS 5

This is the number of converse connections available. Again, you may set this lower to conserve memory.

The next few commands deal with the KA-Node. If you have no use for it, or seldom use it, then you can turn it off with these commands, also to save memory (see the above discussion on memory).

NDWild OFF

Turns the "wild node" function off.

NText (text string)

Here, **(text string)** is the connection text that people receive when connecting to your node, if it is enabled. I recommend turning it off unless it is needed. Therefore, clear this text out.

NUMnodes 0

This shuts off the KA-node function of the TNC. Again, it is best to turn this off unless it provides a regular service.

The next 3 commands can be set to your preference, but there are some important considerations in setting them.

PBKillfwd (ON) or (OFF)

When you turn this off, all personal and NTS traffic messages will remain in you mailbox even after they have been forwarded. It is best to keep this ON to save memory space.

PBPERSON (ON) or (OFF)

When this function is turned on, it prevents a message from being addressed to anyone other than you.

PBHeader (ON) or (OFF)

When you turn this to ON, the mailbox will also store the routing headers along with the message. This takes memory, and some messages have a half-page of lines with this information. If you have the memory, it is nice. Turn it off if you need memory.

Naturally, if you are not familiar with the operation of the mailbox, you might wish to "practice" with it. And always, follow the recommendations of your local BBS sysop when setting any parameters.

Using the Wisconsin Network - Part 30

by Andy Nemec, KB9ALN

The last three installments of our series have found us setting up a few of the more popular TNC's for mail forwarding. This is one way to get "Door-to-Door" mail delivery service in packet radio. This automation of the packet radio system is not too hard to do, but does require a little more than the usual care in setting up. In this installment, we will conclude our sub-series with a look at configuring a couple of the "PK" series of TNC's made by AEA.

There are a number of TNC's with the PK model series designation, and their ability to do any amount of mail forwarding depends on the model and firmware revision of a given unit. We will discuss three models that are similar in their command structure with regard to packet, the PK-88 and companion model PCB-88, along with at least one version of the PK-232 MBX. If you have a PK-64 that was intended to be used with a Commodore 64, sorry. This model has a rudimentary mail drop that cannot be set up for mail forwarding without some major changes. Therefore, we will not discuss that TNC. There were more than a few versions out there of both of these TNC's, and one could write an entire article about identifying the various firmware revisions and their capabilities! We will concentrate on the most popular and readily available used PK's of the 3 or 4 year-old variety.

There were many firmware upgrades available for both of these TNC's, and some of the expanded capabilities are worth investigating. AEA, despite it's troubles in 1996 and 1997, is still in business under different ownership. Parts, upgrades, and all of the other customary support are back in place. That being said, we'll talk about general forwarding considerations.

First, note that most versions of the PK series do not initiate a connect and forward mail on their own. Your local BBS has to connect up and forward to it, and request a reverse forward. In other words, your TNC may have mail stored, waiting to be forwarded until you get incoming mail. Your BBS sysop may be able to provide a workaround, depending on what the BBS is capable of doing.

Next, the message has to be addressed to a recipient at another BBS. This is done a little differently than some mailboxes and just about all BBS's. Normally when you enter a message into a mailbox or BBS for distant delivery, you include the recipient's home BBS and the hierarchical address of that BBS. Not so in some PK's. The usual BBS send command for such a message might look something like this:

S WX9APR@WZ9APR.#CWI.WI.USA.NOAM

But a your PK might only recognize

S WX9APR @ WZ9APR

If you try to add the usual hierarchical address components, it will ignore it and the rest of your message! The PK will still give you the appropriate mailbox prompts, but the message will hit the bit-bucket. The lack of a complete hierarchical address may be a hindrance to automatic mail delivery by some BBS's. Some BBS's have a list of "known BBS's" and this is no problem - they recognize the BBS call-sign and "fill in the blanks". Others may not do this. You must discuss this with your BBS sysop and see if this will cause a problem for him or her.

Another accommodation you must make is the addition of spaces between the recipient call-sign, the @ symbol, and the recipient's home BBS call-sign. Notice that this appears in the above example. A Send command formatted like this

S WX9APR@WZ9APR

will not work! Again, the usual mailbox prompts will appear, but the message will wind up in digital "La-La Land". You and your users need to be aware of this fact and will need to adjust your usual habits to accommodate the PK.

Aside from these accommodations, one more will have to be made. Most PK's will not automatically forward a message, even if it is properly addressed. A message intended to be forwarded first has to be "flagged" for forwarding by the operator of the mailbox. This is done with one of the "Sysop Only" commands available only at the console while the operator is locally logged on, "Edit". This command takes the form of the command, followed by the message number, and the letter **F** (for forward). If message number 5 is to go to a distant BBS, then the command would look like this from the command prompt

Edit 5 F

Simple enough, but it does require that the mailbox Sysop know that the message is there and manually edit the message. If the message is not marked for forwarding, it will just sit there in the mailbox.

There is also another Sysop-only mailbox command, Free. This shows you how much free memory is left in the TNC for mailbox use. This is handy to help keep track of how much space is left when you find your mailbox is heavily used.

Here are the specific PK Commands that must be set for forwarding to take place:

BBSMSGS OFF

This sets the responses the TNC sends to look like "TAPR" TNC command responses. AEA recommends this for compatibility with some BBS software.

HOMEBBS (call-sign)

This designates the BBS that will be forwarding to your mailbox. Do not specify an SSID - the TNC will ignore it.

LITE OFF

This is present on PK-88's, among others. It is "Packet Lite", a shortened version of the AX.25 protocol. Only some TNC's and almost no BBS's recognize this shortened protocol, it should be shut off.

MAILDROP ON

Naturally, your mailbox has to be on before anybody can use it, and this includes the BBS that you intend to exchange mail with.

MYMAIL (Call-sign with SSID)

This is the call-sign you wish to assign to your mailbox. By convention, it is your call-sign with an SSID of -1. While you can use another version of your call-sign, it should be different than **MYCALL** (the default).

PK Commands that are optional:

3RDPARTY (ON) or (OFF)

This can be set to on or off, and determines if call-signs other than yourself can send or receive messages with the PK mailbox. **3RDPARTY ON** allows anyone to send messages to any call-sign.

KILLONFWD (ON) or (OFF)

Setting this to **ON** will kill any messages that have been forwarded. If you find that the mailbox is frequently running out of free memory, turn this on. Forwarded messages will be deleted, making room for new messages.

USERS (2 or more)

If you plan on letting the rest of the world use the mailbox (this includes the BBS that is forwarding to you), then this is best set for more than one user. If you are conversing with someone while the BBS tries to start a forwarding session, the BBS will get a "busy" from it's connect request.

Also to be considered are **MFROM**, **MTO** and any **BUDCALLS** that might be in your TNC. These control the manner in which the TNC listens, and of course it has to be able to listen to hear the BBS make a connection. So do keep this in mind.

This will end our sub-series on mail BBS-to-TNC mail forwarding. Please remember that this is something that requires close cooperation with your local BBS Sysop, and do follow his or her advice. While we can't cover all makes and models of TNC's, your Sysop usually can assist with command parameters and other settings. Good luck with your door-to-door delivery of mail!

Using the Wisconsin Network - Part 31

by Andy Nemec, KB9ALN

Someone once said, in describing packet radio, that it is a mode that is looking for it's "niche". One may easily draw that conclusion these days from simple observation. Not all packeteers use the mode for the same thing. Some use it for information, reading everything that comes off the BBS. Some use the Sale ads on the BBS to keep track of pricing on used gear. Some use it to communicate with their buddies, and still others use it to chase rare DX. In fact, one can almost call packet radio a tool rather than a particular mode in the traditional sense. While it might not be obvious to the casual observer, this is what packet radio is becoming - a resource more than a preoccupation. At least to some members of the packet community, it is regarded more as a utility than a passion. Which is not a bad way to look at it when you pursue emergency communications as part of the hobby we call Amateur Radio.

Packet radio can become a valuable tool in emergency communications, rather than a replacement for a particular way of doing things. While I have not experienced every kind of disaster operation, there are some that I and other amateurs have experienced. In the process of dealing with these disaster scenarios, packet radio has revealed itself as a rather helpful resource. What can you use packet radio for in an emergency?

Well, there are things that packet radio is simply not cut out to do. If everyone used packet exclusively as a means to report storm activity, the Skywarn program would be of questionable value. However, there are some areas where packet has found it's "niche". In the case of the Skywarn system, it can certainly aid meteorologists in getting information before a net control operator is able to get to the NWS office. Many such operations configure the packet station to print out incoming messages so that meteorologists can read storm reports before the net control operator arrives. While voice is often the quickest and most efficient method to convey severe weather reports, it does no good when there is nobody on the other end of the radio circuit. This is one tool that the Skywarn people can find useful.

And we can't neglect APRS. The Automated Position Reporting System can provide "pinpoint" coordinates eliminating any confusion as to a spotter's location. While these packets do not travel through a network well, they certainly can be used locally, with the coordinates relayed via a conventional method. After the storm, many ham radio operators accompany the Red Cross on Damage Assessment rides. An operator with voice capabilities and a laptop can cover both a quantity of information and satisfy a need for immediate communications. Nothing beats packet for passing a quantity of information. What can take a long time conveying via voice can be sent in minutes via packet. Damage assessment data can be saved on the laptop and uploaded later, calls for immediate needs can be handled by voice.

I have also found that, when working with Kewaunee County RACES, packet is a helpful tool there. When long lists of any nature need to be transferred, packet provides an easy way to do this. We had a drill "casualty" list that would have taken 10 minutes to send and verify by voice. With a good typist on the other end, we were able to send it easily in seconds to our Public Information Officer in Green Bay.

When working the Weyauwega train derailment, we found two good uses for packet. First, it was a good backup to our voice radio path to the Division of Emergency Management offices in Madison. The second was to have a method of conferring with the folks in Madison without anyone easily overhearing our conferences. Emergency coordinators often have fears of someone (from the press or general public) overhearing part of a conversation and misinterpreting what they hear. Packet is a little tougher to listen in on, partly because of the equipment, and partly because of the way it comes across the screen when you are monitoring. This, and the ability to sit back and carefully compose a note made our jobs easier - we could think of what we needed to talk about rather than trying to remember it on the fly. And we did not have to be quite as diligent in our choice of wording. Of course, we never did forget that someone could listen in on our conversation. That comes with the territory.

As you can see, packet can be a useful tool when working in emergencies. In the next few installments, we will explore the mechanics of incorporating Packet Radio into your emergency operations.

Using the Wisconsin Network - Part 32

by Andy Nemec, KB9ALN

In the last installment of this series, we took a look at packet radio and how it might relate to emergency communications. We noted that a well thought out system and a properly configured station can be most helpful when performing emergency work. In this installment, we will discuss how to determine if packet radio can be helpful to you, and what to about it if you think it can be.

"Where do you want to go today?"

With apologies to the software folks in Washington, we can modify this question a bit to read "What do you want to do today?". Will you be trying to correct a communications deficiency, or are there any particular special needs you will be attempting to fill? In order to answer this question, you will have to rely on your past experience and by thinking your emergency communications plan through.

While we are trained to be prepared for any emergency, readiness for hurricanes is Wisconsin is a bit unnecessary. Your time as an emergency communicator should be devoted to preparing for and thinking about the kinds of emergencies you are more likely to encounter. Start by thinking about your local situation. Is there any industry in your county that deals with hazardous material? Are you near a major airport? Are there any environmental risks you may have to deal with, such as dams, nuclear power plants, or any other unique risks?

All of these disaster scenarios can necessitate an evacuation, or worse yet, mass casualties. In these scenarios, packet can be a positive supplement to your other communications capability. Packet radio is excellent for relaying large lists of evacuees to the Red Cross or other disaster management agencies, for example. Packet can also be used to transmit other long, involved instructions to and from field workers. It can also, when properly set up, allow one to access the internet for information on dealing with chemical spills or other hazardous material incidents. If you have a large quantity of hazardous materials being used or shipped through your county, you may find this capability interesting.

One seasonal threat we deal with in Wisconsin is severe weather of various kinds. If you often have trouble staffing a weather service office promptly, a packet station can help. Setting up a printer to print out storm data will certainly help meteorologists get data without trying to listen to a radio. In the winter time, snowfall amounts can be reported without making phone calls. Do you routinely have a problem communicating with a particular part of your county, or with a particular place outside of your county? Packet can be a big help if a node is in place to relay your signal.

We have found it very useful to communicate with the Wisconsin Department of Emergency Management via packet. Voice contact is marginal and inconsistent. There are enough network nodes to complete the circuit and communicate with Madison from Northeastern Wisconsin. There are nodes that we use to conference with stations not only from Madison, but from neighboring counties as well.

Speaking of conferencing, packet is particularly useful for this purpose. 4 to 6 stations can conference on a typical conference node, and this can be very helpful for coordinating activities with other agencies

. By this time (and with knowledge of your past experience), you should have a pretty good idea of where you might put packet radio to use. Now let's talk about implementing it. You can take this process step-by-step:

1) Test your idea with existing equipment before investing in the necessary equipment. If you know people with laptops computers and TNC's, go on-site and see how it actually works. Check paths and access to nodes and such.

2) Plan your system. Do you plan to have amateur radio access to the internet through a gateway station? Where would it be, who would control it? Obviously, most laptop computers we will be using will be older and less capable than newer units. If you plan on accessing large amounts of data on a frequent basis, perhaps your station at "headquarters" (the county E.O.C.) can be set up as a server. This would, of course, require software with expanded capability.

3) Select software. This is a function of what you need to do. There are many choices out there, but the rule of thumb is this: simplest is best, unless you have a definite need to complicate things. Field units should have relatively simple software that is easy to learn and use.

4) Once you have determined your plans you can start to acquire equipment. When possible, you should look for the highest-possible performance units. If you have access to a 9600 bps user LAN, and can get a radio capable of operating there with good performance, it would be wise to think carefully about investing in this equipment. Remember, you wish to keep this equipment around for a long time. At some point, 1200 bps will be more of a historical mode than a common operating speed. Many TNC's can be upgraded with a 9600 bps modem, consider this as an alternative. How easy is it to install? I highly recommend a TNC that is truly TNC-2 compatible. Alternate firmware for these TNC's is available from various sources, and plugs right in. In the event that someone creates fancy new firmware capable of something really neat, you will find it is much easier to use this new firmware in a Paccomm Tiny-2 than a Kantronics KPC-3. The Paccomm is truly TNC-2 compatible. The KPC is not.

5) Provide for training. A lot of hams have not been exposed to packet radio, or tried it once many years ago. They may not be familiar with operating the equipment. Even if an operator can make the equipment play, they may not know how to get from point A to point B through the network. In addition to regular training, you should have a short "cheat sheet" to help jog the memory of an infrequent packet operator.

6) Once you have a system in place, designate a person to be in charge of the packet operations and maintenance. You will need someone to cycle the batteries in laptops, radios, and TNC's. This person can also keep operating systems up-to-date, along with keeping any instructions current (like the "cheat sheet" mentioned above).

7) Use packet in your exercises just like any other communications system. You probably already know that people get better at doing something when they practice. Make it a point to keep people in practice.

In the next installment, we will look at the specifics of planning a system.

Using the Wisconsin Network - Part 33

by Andy Nemec, KB9ALN

Last time, we started a discussion about the types of things an Emergency Coordinator or Radio Officer has to think about before planning to use packet in their operations. In this part of the series, we will talk about the actual planning, once you find a niche for it in your ARES and/or RACES system.

Last time, we talked about the types of emergencies, and the types of operations you are likely to encounter. An EC probably already has a very good idea of what is expected of his or her organization. We will assume, for the moment, that we are dealing with a fairly common scenario in Wisconsin. This county is mostly rural, with one smaller city and a number of smaller towns and villages. While Milwaukee county may be the exception to this usual scenario, some of the same basic rules apply, the system is bigger and more complex there.

From what we know of our county, we can prepare for flooding, hazardous chemical spills from either rail or trucking, possible severe weather, and potential of a large fire, perhaps. In the right circumstances, any one of these can mean mass evacuation, mass casualties, or both. Anytime we are dealing with this type of scenario, the Red Cross is likely to become involved. Here is a very good place for packet radio.

In addition to communications between the on-site Red Cross staff and other agencies, there could be utility in having an operator at the Red Cross Chapter office. A Red Cross shelter would be a great place for a portable station. If there is a possibility of a large-scale disaster and multiple shelters, then more portable stations would be handy. You also know you will need communications between the on-site Incident Command Post and the Emergency Operating Center. I

f your county has a mobile communications center, a permanent installation is ideal. An older laptop (8088 even) will do fine in such an installation. In fact, none of the computers need be exotic if you are doing basic packet radio. When starting an emergency packet setup from scratch, it is best to "Keep it Simple,..".

The RF portion will require some testing and perhaps even a little experimentation. How far away will you be from a network node (or something suitably similar) is a big determining factor when you start to look at radios and power levels. The installation itself will also dictate what you use for equipment. For example, you would not want to use a high-powered radio in a mobile communications van. This radio will also be subject to interference from other radios. You may wish to investigate using a surplus commercial radio (Motorola or G.E.). They are generally far better performers in a bad radio environment and are not too expensive. In any case, do make certain the radio is easy to operate.

The TNC is something you can be a little flexible on. There are two considerations here. One is to look for ease of future upgrading. Just how exotic are you likely to get with the system? If you need expandability (for adding APRS, for example), stick to a TNC that is TNC-2 compatible. You might also think about what might be the most popular TNC in your area. It makes sense to select one in wide use. It would be more likely to be familiar to the packet operators. Less time trying to figure out a given TNC is more time spent operating.

The same can be said of software. Field units should use the simplest possible software. Again, familiarity works in your favor. If everybody in town uses Hostmaster, then you should consider it. In general, it is not a good practice to use a complex program like PaKet on a remote or portable station, there is too much to go wrong for the novice user.

Exotic services provided by a station at the EOC or elsewhere need to be carefully thought out. How likely is a whiz-bang service to be used? If it is absolutely necessary to make your job easier, do try to use and implement it in the simplest way possible.

In addition to all of the planning mentioned before, these things need to be mentioned:

1) Keep it as simple as you can.

2) Stay in regular contact with your local Node Operator. Find out what his plans are for the future. Does he have emergency power? Are there any plans he or she is making that would affect your equipment purchases?

3) Try and utilize what you already have, rather than building an entire network around your needs. If you have trouble accessing the network in any of your county, consider looking for a packet operator in that area who can activate his KA-Node (or digi as a last resort). See if he is capable of operating on emergency power. If not, see what you can do to help him with this.

4) While a lot of us are quick to offer help, not many of us are good at asking for it. If this packet game is kind of over your head right now, delegate! Find a practical minded person to head up this project, and get acquainted with the project and packet.

5) Once you have a system in place, use it. Incorporate it into your routine operations as well as your drills, if you can.

6) Make it a point to find a path to the Wisconsin Division of Emergency Management ham station through the network. It is a BBS with the call-sign of WC9AAG, alias of WIDEM. Once you get registered, leave a message with Mack, N9NTB telling him you are packet-ready. Introduce yourself, and leave instructions on the path used to get to your local LAN, as well as a BBS (or mailbox) where mail can get your mail. This will help the packet operators in Madison should they need to get a hold of you, and they can put that BBS in their forwarding files.

7) Support packet radio networking whenever possible. Remember, the more developed a network becomes, the more useful it will be to you.

Hope this helps as a guide to planning the packet-based part of your emergency communications system. Next time, we will look at incorporating APRS into the system. We'll discuss what it is and if it can be useful to you.

Using the Wisconsin Network - Part 34

by Andy Nemec, KB9ALN

Over the course of the last few months, we have been discussing the integration of packet radio into your ARES/RACES operations. This time out, we will go a little further with a discussion of APRS.

Just what is APRS?

It is an acronym meaning "Automated Position Reporting System". The APRS software allows a station to broadcast it's position periodically, to receive position reports from others, and mark it all on a map appearing on your computer screen. It is all done via packet radio's AX.25 protocol, and most operations take place on 144.930 MHz. In addition to the mapped position appearing on your screen, APRS software can also broadcast data from some selected home weather stations. You still can conduct conversations with APRS, though the process is slightly different and prone to errors.

How does it work?

The operator can enter the position coordinates manually, or a Global Positioning Receiver (GPS) unit can be added for truly automatic operation. The GPS receiver listens to signals from several satellites and is able to calculate it's position relative to the satellites. This data is displayed, as well as coded and sent to a data output port. Data from the GPS unit is sent to the TNC, where it is formatted and broadcasted as a "UI" (connectionless) packets.

Digipeaters are used to relay the signal as required. Some digis are devoted to localized areas for weak-signal stations, other digis provide wide-area digipeating, and others act as HF-to-VHF gateways. They all have specific designations in the APRS system, and specific functions.

The software used to interact with the GPS unit and TNC is specialized and responsible for much of the functionality of the system. You cannot use a standard terminal program to operate APRS, it just won't work. The good news is that the APRS software for DOS, Windows, and MAC formats is widely available free of charge. You will be pleased to know that the software does not require a powerhouse of a computer to run, an 8088 or better is sufficient. The software is also designed to work with all models of TNC's equipped with APRS.

What to do with it?

Well, the possibilities are plentiful if you are in the right situation. Think about these for a moment: - Precise position reporting for weather spotting - Following a parade route - Radio Direction Finding - Shadowing Emergency Government personnel Any or all of these can help you to do the job better and faster, not to mention doing things you have never done before. In short, anytime you need to know where someone or something is, you can use APRS to report it's position via packet radio.

Perhaps by now you have concluded that you can use APRS and wish to incorporate it into your system. Now you have to consider what to do in the way of equipment. There are two approaches here, a traditional and a new method that looks to be very cost-effective. First, the traditional approach.

Naturally, you will need all of the normal elements that comprise a packet station. The appropriate radio, antenna, and feedline all have to be there just as in a "normal" packet station. The TNC must be APRS compatible, a lot of TNC's of recent manufacture have APRS support. It is not economically feasible to "retrofit" an old TNC with current APRS support unless a manufacturer offers a low-cost kit for a specific model.

Now you need to think of the GPS unit you will attach to it. Handheld units are popular for roaming, but suffer from a little inflexibility. The antennas are built into these units, and they must have a direct view of the satellites. Permanent locations can take advantage of a remote antenna installations, which removes that problem. Most operators select a hand-held unit for it's portability.

Once you have decided on the style, you have to do a little bit of checking before you can determine if it will work with your TNC. The GPS unit has to send it's data to the TNC, of course. And it has to have the proper port and data format for your TNC to understand it. Luckily, this seems to be somewhat standardized. Most GPS units have the required NEMA data port with the proper format. If there is any doubt in your mind as to compatibility, refer to your TNC's owner's manuals or call the TNC manufacturer. Best to spend the time to check on this very important matter rather than buy a unit you can't use!

This traditional APRS approach requires a radio, antenna, TNC, and GPS receiver dedicated to the task. There is another method of utilizing APRS that may be especially appealing to Emergency Coordinators for more than one reason. This method comes to us in the form of a soon-to-be- released "semi-kit" from the **Tuscon Amateur Packet Radio Association** (**TAPR**). It is called the "MIC-E" and it allows you to operate conventionally, yet relay APRS data with your existing radio. It takes the GPS receiver data, formats it into a packet, and tacks it onto the end of your voice transmission. This is a transmit-only system, and does not require a TNC. It is intended to plug into the microphone jack of the radio, with the regular mike plugging into it. This is quite an advantage in a number of situations. There is a much lower cost of equipment, a less complex system, and the operators have fewer pieces of equipment to fiddle with. Imagine getting an updated position report from a weather spotter every time he unkeys his mike - and you can still utilize existing voice repeaters.

Those are two ways to implement APRS in your emergency communications system. If you would like more information on APRS, try this URL:
http://www.macatawa.org/~ares/
It is especially helpful as this is an ARES organization and use APRS effectively. They also have the current APRS software for you to download, if you wish.

There is also the **Tuscon Amateur Packet Radio Association**'s web site. This is where you go for information on the "MIC-E", as well as general information on APRS.

Using the Wisconsin Network - Part 35

by Andy Nemec, KB9ALN

Packet Radio, since it's inception, has undergone a considerable amount of change. The first big leap after Digipeating was the BBS. Net/Rom (TheNet), TCP/IP, APRS, and promises of more yet have been slowly appearing. Now we are starting to see even more change, and the latest evolution of packet radio is not without controversy. In the next few series of articles, we will explore the Internet Gateway to discover it's draw and see why it has been the target of so much controversy.

Defining an internet gateway can be confusing to some folks - there is some new terminology, and before we go further, we had best define what a gateway is and it's close cousins.

An interactive gateway allows users to connect to the internet through packet radio and utilize the many services found there. In some cases, access is limited to "Amateur Only" sites, still others will allow full access (well, all except the sites that are not appropriate for the air). This gateway also will go "the other way". Users can log in via the Internet, and have access to the radio network. Potential packeteers can get a taste of this mode before they lay out money for this equipment. This type of gateway usually greets the logged-on user with a conventional BBS interface. Most of these gateways use TNOS or JNOS to perform the linkage to the Internet.

Aside from the BBS commands you are used to seeing (like Send, Read, Kill, List, etc.), there will be a few unfamiliar ones (like **telnet**). Most interactive gateways will support AX.25, Net/Rom (TheNet), TCP/IP and round-table discussions, similar to the "Internet Relay Chat". You may send and receive E-Mail through the internet, through the BBS network, connect up to distant Amateur sites on the internet, and use it as a conventional Network Node on radio or the internet. In that sense, the commands mirror the nodes that you are used to.

Some of these interactive gateways also provide this "wormhole" service that treats a distant node as just another node on the network. New York becomes just as local to the network as "the next town over" when using a wormhole in this manner.

Another type of gateway is the dedicated point-to-point wormhole gateway. Unlike a fully interactive gateway, it is limited to providing this "wormhole" service only and is not intended to provide users with full access to the internet. It becomes a bridge to complete a radio network. For example, there could be a wormhole between a system in Milwaukee and one in Florida. The goal is a seamless linkage between two user LANs in different parts of the country. This has generated a bit of controversy, as the Tuscon Amateur Packet Radio Association (TAPR) has been promoting the idea of using the internet to complete the national packet radio network in this manner.

Mail Forwarding Gateways do not provide real-time access to the internet, and it does not complete a radio network on a real-time basis. Rather, it is used by two BBS's to exchange mail via the internet. Most often, this is used when radio paths are unreliable or unavailable. This practice has been the subject of controversy as well. While we will not attempt to enter the controversy in this column, the explanation should help you understand what each service is and may help you understand the debate better.

Now that we know what types of gateways have found use, we can talk about how they are constructed. It is not too hard to understand, but it is not your typical packet station. Let's talk about the first case, a fully interactive internet gateway.

Any type of gateway is constructed with two or more "ports". One will go to the internet and may well be a phone modem, or it could be a network "ethernet" card hooked up to some kind of network that eventually reaches the internet. In any case, this port allows an interface with the internet. The other ports are connected to a node stack, or become a node stack.

It is possible for one of these computers to act as a full node and network router. In the case of the machine being tied to a node stack, the serial port of the computer is "multiplexed" into the serial ports of the TNC's. This is the same situation as a normal node stack - the computer is just treated as another TNC. In the case of the computer performing the task of a node stack, the TNC's are connected to serial ports of the computer. The TNCs operate in KISS mode and the computer software does all of the work normally reserved for the node firmware. From there, the rest of the node stack - radios, antennas, etc - are the same as in a conventional node stack. So the internet gateway can either replace the normal node firmware, or augment it.

Now that we have our look at gateways, we will further explore how to use them in the next installment of "Using the Wisconsin Network".

Using the Wisconsin Network - Part 36

by Andy Nemec, KB9ALN

In our last installment of this series, we started a discussion of Amateur Radio-to-Internet Gateways. Non-interactive gateways (used for mail forwarding and to complete the network) are largely "invisible" to you. So we will devote our time to the discussion of using interactive gateways. These packet radio stations can perform most, if not all, of the following functions:

1) Handling Internet E-Mail (Outgoing and Incoming).

2) Provide access to other Amateur Gateways.

3) Allow connections to Internet sites of Amateur Interest.

4) Provide local node service.

5) Provide access to Amateur "Converse Servers" (more on that later).

6*)* Provide access to a few Amateur-interest "mailing lists".

7) Make dedicated node circuits available to other packet operators in other parts of the country.

By far the most popular use of a gateway is for Internet Electronic Mail. You could use a gateway to E-Mail anybody who has an internet E-Mail address. Of course, there are some limitations as to who you can E-Mail because we are using Amateur Radio. Legality, privacy, and other message content are important considerations here. That being said, how do you do it?

Surprisingly enough, there is not too much difference between gateway E-Mail messaging and "regular" packet messaging. The procedure is the same as using any BBS, in fact, you are greeted with a BBS-style prompt when you log-in. Consider the prompt you get when logging into the KB9BYQ gateway in Appleton:

**CALL,BBS,WX,CONV,NODE,NWSGRB,SYSOP,GRB4,A,B,C,CO,D,E,FI,G,H,I,J,KM,L,M,N,P,PI,PO,Q,
R,S,T,U,V,W,X,Y,Z,?>**

While this seems to be quite a lengthy prompt, some of it will look familiar. Notice the **K, KM, L**, **R** and **S** in there. They do the same thing as on any other mailbox or BBS. The procedure for sending Internet E-Mail is the same as you are used to, assuming you have used a packet mailbox or BBS of some sort.

Assume you want to send a message to your friend, who has an E-Mail address of **bigbob@aol.com**. After you get the prompt, type:

s bigbob@aol.com

You will get a return prompt of:

Subject:

As usual, you will type in a subject, and press the Enter key. The next prompt says:

Enter Message (^A Aborts, end with ^Z or /EX on a new line)

Then, you type your message. When you are through, press the Enter key and send the key combination Control and Z, or send /EX to close the message. The gateway will respond with:

Message forwarding to bigbob@aol.com

and return the BBS-style prompt.

Reading mail is exactly the same as you normally do - **R 1** will let you read your message #1 on the computer. **KM** will kill all of your messages, and **K** (followed by the **message number**) will kill a selected message.

While this is simple enough to do, there are a couple of things you need to be aware of before you send off your first message.

1) Internet E-Mail addresses are case-sensitive. You cannot expect to leave your "Caps-Lock" on and have an address interpreted correctly. **BIGBOB@AOL.COM** is not the same as **bigbob@aol.com**, nor is it the same as **bigbob@AOL.com**.

2) While we are on the subject, it is not considered good etiquette to type your message out entirely in upper-case. Internet-savvy folks refer to this as "Shouting".

3) What's legal? In general, you follow the same rules regarding voice radio operation. While the FCC does permit us to carry on a limited amount of business over the air, it does not allow us to run a business over the air. It is best to be safe with this. Do not make any business inquiries via E-Mail and make certain the language in any message (sent or received) is within the limits of Part 97.

4) What's permitted? The gateway operator becomes the "first forwarder" of any traffic you may receive. Because of the his license is on the line, he sets any restrictions on it's use. Consult the gateway Sysop about his usage policy.

5) Disconnect if you start to receive an inappropriate message. Reconnect to the gateway, list the message, and kill it. If it is unwanted E-Mail from a commercial promoter ("SPAM"), notify the gateway Sysop. If the offending mail is from an acquaintance, perhaps a phone call to him or her is in order.

6) Incorrect addressing will result in an error message appearing the next time you log in to the gateway. If you get one of these, simply kill off this message, along with the original "bounced" message. Re-check your addressing, and try sending the message again.

That's all for this time. In the next installment, we'll discover more things you can do with a gateway.

Using the Wisconsin Network - Part 37

by Andy Nemec, KB9ALN

In our last discussion concerning Internet Gateways, we started a discussion of mail service. This time out, we will continue our discussion as there is a lot more to internet E-Mail than there is to most packet mail.
In our examples, we have been using a typical gateway setup, KB9BYQ's TNOS gateway in Appleton.

Internet E-Mail, as most often seen, is "plain text". That is, a letter typewritten and easily readable as ASCII text. Of course, as soon as people found out how wonderful electronic mail was, they wondered if it was possible to mail programs, pictures, and other binary format files. Of course it is, through the use of encoding and the E-Mail attachment. You can send someone almost any kind of file in this manner, a binary executable program, a JPG picture, and even video and "movies". You can also send word processor and other files ("rich text") that are stored in a binary format. These are exactly the same as any other binary file, therefore, they must be specially encoded to be sent in a "plain text" mode.

If you operate a TCP/IP station, you probably already know that the mailer you use will take care of any special file types (at least most often). The information provided below is not so much for you, but for users who connect via AX.25 text mode to collect their mail from the gateway. Most TCP/IP users will have their mail automatically collected by your TCP/IP program. However, you may find the following information helpful, especially the information concerning file sizes.

There are various methods used to encode programs so that they can be mailed, among them UU, Base 64 and Bin2Hex. While this is no problem for the internet, where data rates reach astronomical speeds, they can represent a considerable challenge when collected by packet radio. There are a few things to consider if you decide to mail or receive a program encoded in this manner. The first thing to consider is the size of the file. It would be a lifelong endeavor to collect a 1-megabyte file at 1200 bps! You best recommend to anyone mailing you at a gateway to limit file sizes. Even a 10K file can try your patience at a typical LAN speed.

Also remember that sometimes an encoding program will double the size of the file! Before you send anything this way, encode the file and look at the file size. Anything that "bloats" a file is to be avoided (unless the file is a small one). In order for a lot of word-processor documents to be successfully transferred, you may need to convert these as well. Most of these kinds of files are a combination of plain text as well as odd control characters. In order for them to be safely transferred, they should be encoded prior to being sent.

HTML documents, on the other hand, do well via standard packet mail. HTML is a text-based programming language that is used extensively on the World-Wide Web. Of course any such document must be viewed with a Web Browser, but they can be transferred via packet mail, with no encoding necessary.

How to tell if you need to encode a file

That's pretty easy. If it has a file extension of .TXT, then it is usually safe to send without encoding. Don't be fooled by a .DOC extension - most of these are plain text, but some are word processor documents containing formatting and control characters. When in doubt, look at it with the DOS "TYPE" command (or a program like LIST.COM) and look for odd characters on the screen, or beeps coming from your PC's speaker. If you see or hear anything unusual, then it should be encoded. Any document created with a word processor and not saved as "ASCII plain text" will need to be encoded.

That being said, here's a special warning to users of Windows "Write!". Even if you tell this program to save the file in "plain ASCII text", it will add extraneous control characters to the file. Take note of this and either use a plain text editor, or use a different word processor that you are certain saves files as plain text.

Encoding and Decoding Methods

Now that we have a pretty good idea of what might need to be encoded, we should explore what kinds of encoding and decoding program can be used. One of the more popular ones used by Internet mailing programs is base 64. The procedure in a case like this is to capture the file as you would any other text file you are saving. Then copy the file to the directory that your internet mailer uses, and tell it to decode it. Then you look at it with the mailer, or save the executable file you have been sent.

Another method of file transfer via mailing is to use the UU encode and decode programs. These are stand-alone DOS programs that will encode or decode a binary-format file into a format that looks like plain text to the mail programs. When receiving a file like this, simply tell the UUDECODE program to decode it. You can then use the executable, or view a "rich text" format message or word processor document.

Either of these file conversion methods are likely to be decodable by mailers, or at least easily decodable with a stand-alone program. When in doubt about any such file exchange, think about the following:

- Will your correspondent be able to decode the message?

- Will you be able to decode the message?

- How big is the file?

Common sense says to keep any file under about 10K. - What is the content of the file? Remember, a picture file of inappropriate subject matter is just as illegal as any illegal wording. And one more thing to discuss.

Many of you have seen me use words like "Encode" and "Decode". Many astute observers will wonder if encoding a message sent via packet is legal. The phrase "Codes and Ciphers" in part 97 rings a bell with most people, but they often forget the "rest of the story". The full text of the rule in question should help to reassure you on this matter. This rule points to intent of the encoding more than anything else. Remember, it is illegal to code a message to hide it's meaning. It is OK to compress or encode a message if the intent of the encoding is to facilitate message exchange.

In other words, if you are using encoding to make a message transfer more efficient or even just plain possible, you are following the law. If you are using encoding simply to make it harder for someone to see your message, then you are outside of the law.

That's all for this time. In the next part of our series, we will continue to discover more of the features of the gateway.

Using the Wisconsin Network - Part 38

by Andy Nemec, KB9ALN

The last two parts of our series have found us exploring the Internet Gateway. We will conclude with a catch-up of miscellaneous topics in a "Q&A" format.

Q - I saw in part 37 of this series that one can send binary files if they are encoded. However, the sysop of the gateway I use frowns on this practice. Why?

A - The reason is that he does not want to spend his time trying to decode and view every encoded message that goes through his system. The same rules regarding regular packet mail forwarding apply to the internet when Amateur Radio is involved. The originator of the message, as well as the first forwarding station are legally responsible for the content of the message. Therefore, a good gateway operator will want to keep a close eye on things, it is his license on the line. Obviously, you are using the gateway at the pleasure of the Sysop, so it is reasonable for him to limit encoding mail messages - it does take a lot of time to make sure their content is legal.

Q - What should I do if I get SPAM mail at a gateway?

A - If you are in the process of reading it, stop immediately. If you must do a hard disconnect, so be it. Don't kill the message yet! Then contact the gateway Sysop and inform him or her of the situation. The Sysop will probably want to look at the message to see if the point of origin can be determined. If this can be done, a Sysop can generally set a trap for that originator and block out future mail delivery attempts from that internet host.

Q - I have been trying to mail a friend of mine, and every time I try, the gateway says "Bad host" and won't let me leave a message. What is happening here?

A - Chances are, you have something wrong with the way you typed the address. E-mail addresses are structured like this:

username@computername.networkname.domain.optional-country-code.

The username is your friend's internet mail name. The optional computername is followed by a network name, then the domain. Valid domains are org, net, com, gov, mil, tv, biz and a few others you will be seeing shortly.. If your friend has an internet provider in a country other than the US, you will probably see a country code in it. Internet addresses are in most cases entirely lower case. Occasionally you will see a username in Capital letters. If this is the case, then type ONLY that portion in capital letters. Pay special attention to the rest of the address - That is what the gateway says is wrong with the address. What it is saying is "There appears to be no such computer known by that name on the internet". So you need to triple check the address, and correct it.

Q - I sent a message to a friend and got another message later titled "Failed Mail" from "Delivery Subs". I did not ask to have subs delivered, what is this?

A - Nobody called up a sub shop as a prank - this is a message from the mail delivery **subs**ystem telling you that your message was refused by the destination computer. The most frequent reason is an incorrect user name in the To: address. Another reason could be that the destination computer was not able to be reached. It may be because of a network failure, or the destination computer was temporarily off-line. Included in the message will be a notice that the mail system keep trying for 3 to 5 days. If this is the case, wait and see if you get another message telling you it had failed and was deleted. If you do not get this message, then you should try mailing again.

Hint: If you get one of these messages, save it to a file on your computer. That way, if it fails completely, you can re-edit it and upload it later in another message. That is, if you do find out where the problem is.

Q - What is an AXIP link? What do I have to do to use it?

A - AXIP is a way to send the AX.25 protocol that we all use over the internet. An AXIP link will operate as though you are in direct communication with another AX.25 station. This allows network nodes to function over the internet in the same manner that they operate on radio. While not particularly efficient, it does work and provides a lot of AX.25 users with an easy way to access remote Net/Rom nodes over the internet. If you connect up to a gateway and connect to a remote node, you are using an AXIP link. There is nothing special to do, an AXIP link generally appears transparent to the user. If you know how to use a regular networked node through the internet, you know what to do. If you have not ever used a node, check out the first few parts of this series to familiarize yourself with one.

That's all we have space for this time.

Using the Wisconsin Network - Part 39

by Any Nemec, KB9ALN

In this part of our series, we will take a look at a different way to network in Packet Radio - Flexnet.

Flexnet is a digipeater-based method of networking nodes, somewhat similar to "Net/Rom", which is used extensively in Wisconsin. Although I am not aware of any Flexnet digipeaters operating in Wisconsin, it is interesting to look at other networking options. Flexnet comes to us from Germany, and was developed nearly 10 years ago. It enjoys wide popularity in Europe. but has not caught on much here in the U.S.

Some may be wondering why Amateur Packet radio should use digipeaters - after all, they are inefficient and one missed packet can translate into several retries on a marginal path. With normal AX.25 digipeaters, this is indeed a probability. Flexnet, however, operates differently than digipeaters that you may be accustomed to. Packets are acknowledged over every step of a multiple-hop link. This makes for a reliable connection over several digipeating "nodes". The mechanism used to acknowledge packets is different than Net/Rom, and far more reliable (according to it's proponents). The end-to-end acknowledgment of basic, unenhanced digipeating is gone, along with it's problems.

Flexnet is AX.25 based, and has the ability to digipeat TCP/IP frames that are encapsulated in AX.25. Net/Rom does not have the same ability to carry TCP/IP - when forced to, it will, but very poorly. A thought may have occurred when comparing Flexnet with Net/Rom. Does a packet operator have to know the call-signs of every digipeater in the connection path? No, Flexnet is autorouting. In other words, it knows if it can get to a distant digipeating node, and which route to take in order to get there. All you need to know is the call-sign of the station you wish to connect to, the call-sign of the local digipeater (node), and the call-sign of the distant digipeater. From there, routing is automatic - the network knows the way.

The down side is that operators will not have the easy node aliases to remember - Flexnet seems to operate with call-signs only. It may be possible to use an alias as a call-sign, but the authors do not mention this in their introductory documentation.

Features

One very attractive feature of Flexnet is compression - all packets sent between Flexnet Digipeaters are compressed to save bandwidth. This helps with speed - Flexnet spends a lot of time acknowledging packets between it's digipeaters. That acknowledgment makes for it's good reliability, and the compression is needed to make this operation speedier than it would normally be.

As was mentioned before, it acknowledgment method is different than Net/Rom. This means that "hung" Net/Rom connections would be a thing of the past. Connections across state lines with multiple digipeating "nodes" is theoretically possible (if enough nodes exist to make a path). It's authors say it is possible to make a connection across Germany and into other countries solely with RF based nodes. It should be noted though, that Germany and most of Europe enjoy an abundance of network nodes, so this more than anything makes this possible. However, it does effectively show that we can overcome the Net/Rom restriction on the number of links. And it also shows us that Flexnet is a workable networking system in a crowded packet radio environment.

Flexnet is not limited in it's architecture in most ways. It is capable of operation at both 1200 bps and 9600 bps. It is capable of multi-port operation, so it is possible to incorporate it into a LAN/backbone "node stack" setup. It should be noted that there are some serious hardware considerations that we will explore later on.

And one appealing feature certainly is cost - there is none. It is available through the Internet on the author's home page.

Drawbacks

There are a certain number of drawbacks to Flexnet, and they would certainly have a bearing on it's implementation. While the software is free, a node operator may well have to make it all up with hardware.

The most common implementation involves a number of PC Packet cards, installed in a personal computer devoted to the task. It is possible to use a KISS TNC with Flexnet, but all of the Flexnet capabilities are generated in software. As of this writing (and this may be subject to change soon), there is no plug-in TNC-2 firmware chip that has Flexnet capabilities. So, a computer has to be used to perform the actual Flexnet operations. And based on experience, an 8088 may not be fast enough to do the trick in a multi-port "node-stack" type of setup. Now that 486 computers are getting fairly inexpensive on the used market, this may be less of problem than in years past. However, most node sites are remote and "unattended". One may go through more than a few 486 computers in a good lighting season!

TCP/IP operators may have an interesting time with Flexnet. The main difference between it, and say, an X-1J node is that it reverts to a "**Virtual Connection**" mode rather than a "**Datagram mode"**. This means you will have a particularly difficult time with timing parameters like IRTT, for one.

The most significant drawback to Flexnet may be the very thing that may make it desirable - it is different. This means that any part of the network using it would not easily interface to a different network. Even if a large commitment to "change over" network software was agreed upon, there would be gaps in coverage while the changeover was taking place. Some may consider it a nuisance, others (like Emergency Coordinators) may look at changeover coverage gaps as a real handicap, temporary or not. In short, a changeover of this nature (Flexnet or some other system) would not be entirely painless.

Is Flexnet a workable networking option in Wisconsin? Only real working experience will truly tell if it is. It may appear not, simply because our neighbors would certainly have a tough time interfacing with us. Add to that the fact that there is no support for it in **MSYS** (the BBS software of choice in Wisconsin) as well as other popular Host TNC programs, and the deck is somewhat stacked against it.

With the continuing incorporation of the Internet into packet radio, it would seem most likely to use TCP/IP as a networking protocol. However, there are some elements of Flexnet that certainly can be incorporated in some kind of radio adaptation of TCP/IP. So it is certainly worth testing, for those of you who are interested in digital networking. So if you are hardy digital experimenter, research it via the World-Wide-Web and check it out yourself. Find out what others have to say about it and try it.

Using the Wisconsin Network - Part 40

by Andy Nemec, KB9ALN

In this last part of our serious, I will answer a question asked of me quite some time ago. It is one that may not arouse much more than idle curiosity of you, but it may be of interest. The question: "What is **TNOS/LINUX** and do I need it?"

Actually, "**TNOS/LINUX**" indicates two different things. Regular readers of this series may remember that **TNOS** stands for "**Tampa Network Operating System**". This is a spruced-up version of the venerable **KA9Q NOS** that has powered **TCP/IP** stations for quite a while. It has specialized features and enhancements above the KA9Q NOS and is widely used in Amateur Radio for gateways and BBS's.

The second half of that, **Linux**, is another thing altogether. This is a different kind of computer operating system - different from **DOS** and **Windows**. In fact, it is not a Microsoft product, and is quite different from these products.

Linux transforms a standard modern PC into a "**UNIX** computer". **UNIX** is a computer operating system that has it's origins in the AT&T corporation. It is designed to be a multi-user, multi-tasking operating system, unlike DOS (and some would argue Windows). Linux slightly differs from UNIX in some of it's commands, but basically is a special version of the UNIX operating system intended to be run on the Intel 80x86 processor that is the heart of most personal computers. There is also a special version for Mac Intosh computers. What does this have to do with Packet Radio, you might ask?

Simple. Because Linux is Multi-user and Multi-tasking, it allows a BBS or gateway operator a great deal of flexibility with his system. No longer is a computer strictly devoted to BBS use. The multi-tasking nature of Linux makes it possible to not only run the BBS program, but also run one or more other application programs for one or more users. While the BBS runs quietly on one "virtual console", the Sysop can be editing files, playing games, or actually getting some work done! Linux is an extremely efficient operating system, and because of this efficiency, it can do a whole lot more than a computer using DOS and Windows at one time. It also manages memory by itself - no external memory managers are used to overcome the DOS 640K memory limit. This is a real problem in BBS's - the more users that are logged on, and the more forwarding jobs the BBS is engaging in, the more memory is needed to process all of this activity. Linux overcomes this barrier.

Linux also has networking abilities built right in - part of the operating system "**kernel**". This is particularly handy for gateway use - a computer running Linux will not require yet another computer to interface it to an internet provider. Some BBS programs will not work very well for this while operating under a DOS operating system, others cannot do it at all. Because networking is built right into Linux (and because the memory management is so good), this type of operation comes naturally to it.

Do you need Linux?

If you are an average packet user, no. Most packet users are content to run their host or terminal program under DOS or Windows (or maybe even **OS/2**). They may only be interested in doing one thing at a time, and do not run a BBS or a gateway. However, there are several packet programs available for Linux, both simple and complex.

Linux seems to be tailor-made for "power users" mostly, or anyone who has something they want done with a computer that is difficult or impossible to do with DOS.

Another consideration with Linux is that it is not an MS-DOS system, so the file system and the very nature of the operating system is different. You cannot pop a DOS disk into a machine running Linux and expect it to work. You must run DOS programs on it's DOS emulator, (or a Windows emulator for Windows programs) While some DOS and Windows programs run well with these emulators, not all such programs will run.

Because Linux is not nearly as popular as DOS and Windows is, you will not find a vast quantity of ready-to-run software for it - somewhat of a disadvantage.

Can you use it?

Of course you can, if you are up for a challenge. I only say that because you will have to learn the commands associated with an entirely new operating system. Remember the first time you used DOS? That memory will return to you if you investigate Linux. Although there are some familiar commands (some DOS commands have their origins in UNIX), the syntax is a little different and there are a whole lot more of them. Linux also has a freeware Windowing system called **X-Windows**.

So there is plenty to learn with this operating system. Do you need anything special to operate Linux? Older versions of Linux (like Red Hat 4.0) will run on a computer with a 386 or better processor. Memory is helpful, 8 megs is recommended, but 4 will work fine. I have run it on a 16 MHz 386 computer with 2 megs of RAM, and was surprised how fast that old 386 ran. Newer versions require much more in the way of memory and utilize the newer Pentium processors. The newer versions (such as Red Hat 7.2 and up) are geared toward power users who do a lot of intensive graphic and office work. As a result, a Pentium processor and 32 Megabytes of memory are required to take full advantage of their capabilities.

You won't need to completely trash your existing operating system in order to try out Linux. One can get a "boot manager" that will allow you to boot your computer into one of a number of operating systems, if you desire.

So, if you are bored with your computer and want an interesting challenge that will reward you with some amazing capabilities, you may wish to investigate Linux. If you have Internet access, browse the Web page for the official Linux site: **www.linux.org**. If you are the slightest bit interested, you will find hours of reading material there.

There are also several CD vendors that sell Linux "kits" for a nominal price. One vendor, Slackware, targets it's package toward amateurs and includes a lot of ham-related programs in it's package.You can also find amateur radio packet programs to use with Linux from various Web sites. **www.lantz.com** is the official site for TNOS, for example. The **Tuscon Amateur Packet Radio Association** (TAPR) software library also has a section devoted to Linux.

So if you are up to a challenge and want to learn more about networking, look to Linux. If you are content with DOS and Windows, and you find your packet program meets your needs, then you may find your questions answered.

End of the series