XNET

A GRAPHICAL LOOK AT PACKET RADIO NETWORKS

Richard Parry, W9IF
rparry@qualcomm.com
http://www.qualcomm.com/~rparry/xnet

ABSTRACT

This paper describes AX.25 packet networks from a graphical standpoint using XNET, a software program specifically designed for network analysis. Networks are complex entities most easily explained visually. Through the graphical displays, one can more easily gain an appreciation and understanding of a network. It also allows one to see problems and the general behavior of the network. *XNET* runs on UNIX/LINUX systems supporting the Tcl/Tk language.

KEYWORDS
Packet Radio, Tcl/Tk, Linux, UNIX, AX.25

INTRODUCTION

Amateur packet radio networks can be an enigma. The average packet user sits in the ham shack enjoying error-free communication while sporadically hearing the Terminal Node Controller (TNC) toggle the transmitter on and off for no apparent rhyme or reason. Somehow messages are sent and received, files are uploaded and downloaded, countless messages fly through the ether without error, all despite the whims of both atmospheric and man-made interference. What is most miraculous, is that the users operating in this manner share the same frequency, an impossibility on virtually all other standard analog modes of communication such as SSB, AM, or FM; spread spectrum being one of the few exceptions.

Understanding and studying a network is a complex matter. Fortunately, network analyzers are available to aid the packet user in understanding how a network operates. Two such programs are *XNET* for UNIX/LINUX systems, and *Packet Tracker* for Apple Macintosh computers. In this paper, XNET was used for all graphic network displays. Even for those who have no intention of using a network analyzer program, the diagrams in this paper and the XNET Home Page on the World Wide Web should prove to **be** of interest as they clearly show how a packet network functions thereby providing greater insight into networks.

64

XNET

XNET is an X-'Windows based network analyzer designed specifically, to monitor $AX.25$ packet radio networks. It will collect and display network data allowing the user to understand network traffic and channel utilization. The program is written in Tcl/Tk (Tool Command Language / Toolkit) and can operate on any computer system supporting Tcl/Tk and the UNIX operating system.   The only additional hardware necessary to use XNET is a radio receiver and a Terminal Node Controller.

XNET provides many features that are useful to both the casual packet user as well as a packet radio BBS or DX packet cluster system operator wishing to better understand a packet network.   Functions performed by XNET include:

- Packet counting
- Displaying network nodes and statistics
- Graphical representation of network utilization
- Displaying raw network traffic
- Playback of network. traffic
- Visual display of network connections
- Extensive use of color and GUI to display information

It is reasonable to mention what XNET does not do, since there are some limitations.   The limitations are not necessarily in the program, but in the system.   For example, data received by XNET from the TNC is filtered, resulting in lost information that cannot be used for network analysis.. Specifically, all packets sent to XNET from the TNC have a valid FCS (frame checksum), since the TNC will not pass packets to XNET with an invalid FCS. Therefore XNET has no w-ay of keeping statistics regarding collisions, communication errors, and other similar parameters that the TNC filters out.

These limitations can be cured by placing the TNC in the KISS mode which causes the TNC to pass raw bit level data. XNET would then be responsible for parsing the packets, error checking, and numerous other functions. The resulting program would provide additional useful information, however, it would be a significantly different program.

MAIN CONSOLE

When XNET first begins, a single window called the *console* is displayed as shown in Figure I.   This console provides the primary user interface for the operation of XNET.   The main console's primary purpose is to allow the user to select desired features such as the MAP, TERM, NODES, and GRAPH windows, as well as setting preferences using the SIMUL, PORT, and PREFS windows to configure the program.   The START and STOP buttons perform the function of beginning and ending the program. and the ABOUT button provides a short description of the program, author, and version.

The *Start* Time as shown in the console, is the actual time and date when XNET was started by the user. This time is used for computing *Total Packets* which represents the total number of packets received from the starting time and date.

Elapsed Time represents the duration in hours, minutes, and seconds since XNET was started.   In the example, XNET had been running nearly two hours.

Total Packets provides the grand total number of all packets received by XNET from the initial start of the program. This value is always equal to or larger than the number *of active packets.*

Active Packets are packets from stations that are currently on the network.   The difference between *active* and *total packets* arises from the fact that stations leave the network and thus have timed-out., resulting in packet traffic being removed from the active packet count. In this example, during the nearly

two hour duration that the program has been running and collecting statistics, a total of 1,442 packets have been received, and 363 packets have been received from nodes that are currently on the network .

Total Nodes represents the total nodes that have been monitored from the initial *Start Time.* This value is similar to the *Total Packets* since it represents all nodes that are presently on the network as well as those that have disconnected from the network. Therefore, *Total Nodes* is always larger than or equal to the *Active Nodes.*

Active Nodes, as its name indicates, represents all nodes that are currently active on the network. As we shall see when we discuss the NODES window, details of these active nodes are displayed there. The difference between the *Active* and *Total Nodes* arises from the fact that some stations are removed from the active node list when they have timed-out. This results in the nodes being removed from the active node count, while still being accounted for in *Total Nodes.*
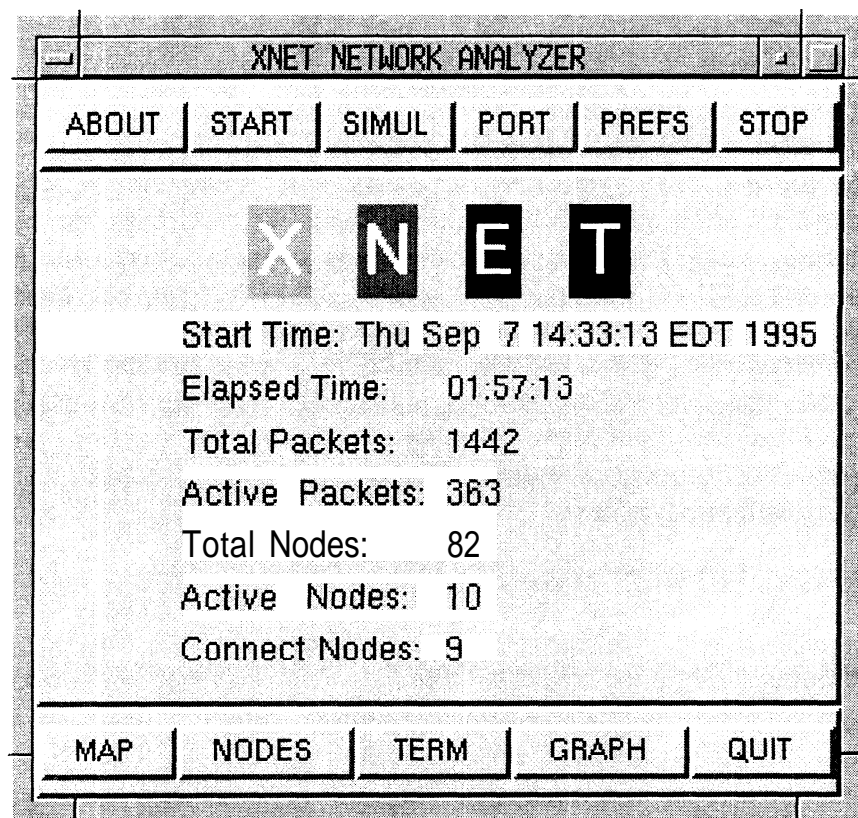

Figure 1. Main Console

Connect Nodes represents the number of nodes displayed in the MAP window. All the nodes listed on the MAP and therefore counted, are nodes which have transmitted packets or have been sent packets. Therefore, in the example, during the nearly two hour running period, 82 nodes were active on the network, only 10 nodes are currently active, and 9 of the nodes have sent and received packets.

MAP WINDOW DEFINITIONS

The *map window* gives the "big picture" of the network. It is intended to be a graphical representation of a heard node on the network sending packets to a destination node. Before examining

an actual map window. let's take a moment to understand how to read the maps and define a few key terms.

Heard Nodes are stations that have actively sent a packet: they are displayed on the map in blue. Nodes that have been sent a packet are referred to as *Destination Nodes*;they are displayed in brown. It is important to note that a destination node may or may not be on the network, Just because a node sends another node a packet, we have no way of knowing if that destination node is on the network until that destination node sends a packet. Then and only then is it promoted to heard station status. Other nodes which are on the network, but not heard directly. are referred to as *unheard stations* and are <u>not displayed</u> on the map.

Referring to Figure 2, first note that each node is drawn twice, once as a source (left side) and once as a destination (right side). In addition, although no arrowheads are attached to the line by XNET. the line always represents packet traffic from source to destination (i.e., left to right). It' NODE2 sends a packet to NODE1, a line is drawn from NODE2 on the left (source) to the NODE 1 on the right (destination'). Similarly, if NODE 1 sends a packet to NODE4, a line is drawn from NODE1 on the left (source) to NODE4 on the right (destination). Lines are drawn indicating that a packet was sent, however. there is no *connection.* We will shortly discuss exactly what *we* mean by a *connection.*
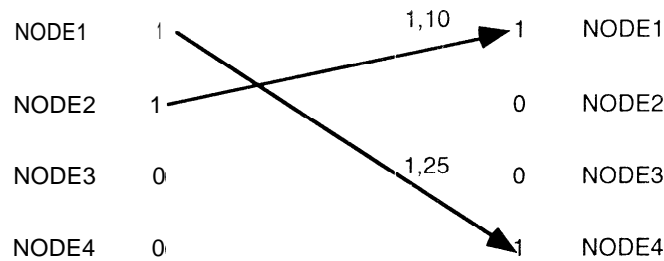

Figure 2. Unconnected Nodes

Let's look at the example shown in Figure 3. If NODE1 sends a packet to NODE2, a line is drawn from NODE 1 on the left (source) to NODE2 on the right (destination). If NODE2 returns a packet to NODE1 , a line is drawn from NODE2 on the left (source) to NODE1 on the right (destination?. The result is a connection. XNET displays a connection using blue lines rather than brown.

A further explanation of connected nodes is in order. AX.25 is a *connection oriented protocol.* To a telecommunication's engineer this has a very specific meaning. If a node sends a packet to another node. a line is drawn from the source to the destination node. If the destination node returns a packet to the source node, XNET assumes that a connection between the nodes is in place. Admittedly. XNET loosely interprets a connection. One could argue that each node is sending packets to each other but neither is listening, thus a connection is not in place. Although possible, this scenario is assumed to be unlikely.
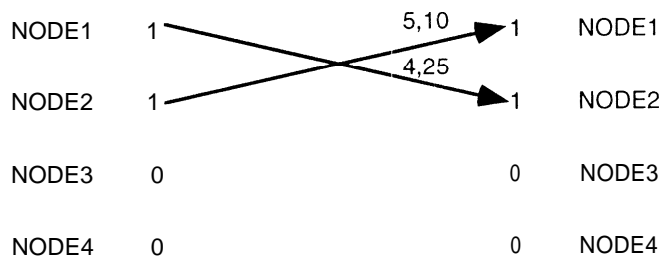

Figure 3. Mutually Connected Nodes

The XNET maps display additional information. Near the destination node column on the right of the map, there is a pair of numbers separated by a comma. The first number represents the number of packets sent to the destination node by the source node. The second value is the time in seconds from when the source node sent a packet to the destination node. Returning to Figure 3, note NODE1 has sent 4 packets to NODE2 and 25 seconds has passed since that packet was sent. Similarly, NODE2 has sent 5 packets, and 10 seconds has elapsed since that last packet was sent.

MAP WINDOW EXAMPLES

We are now ready to examine a few real network maps. These maps were developed from monitoring actual on the air packet networks in the Dallas, Texas area.
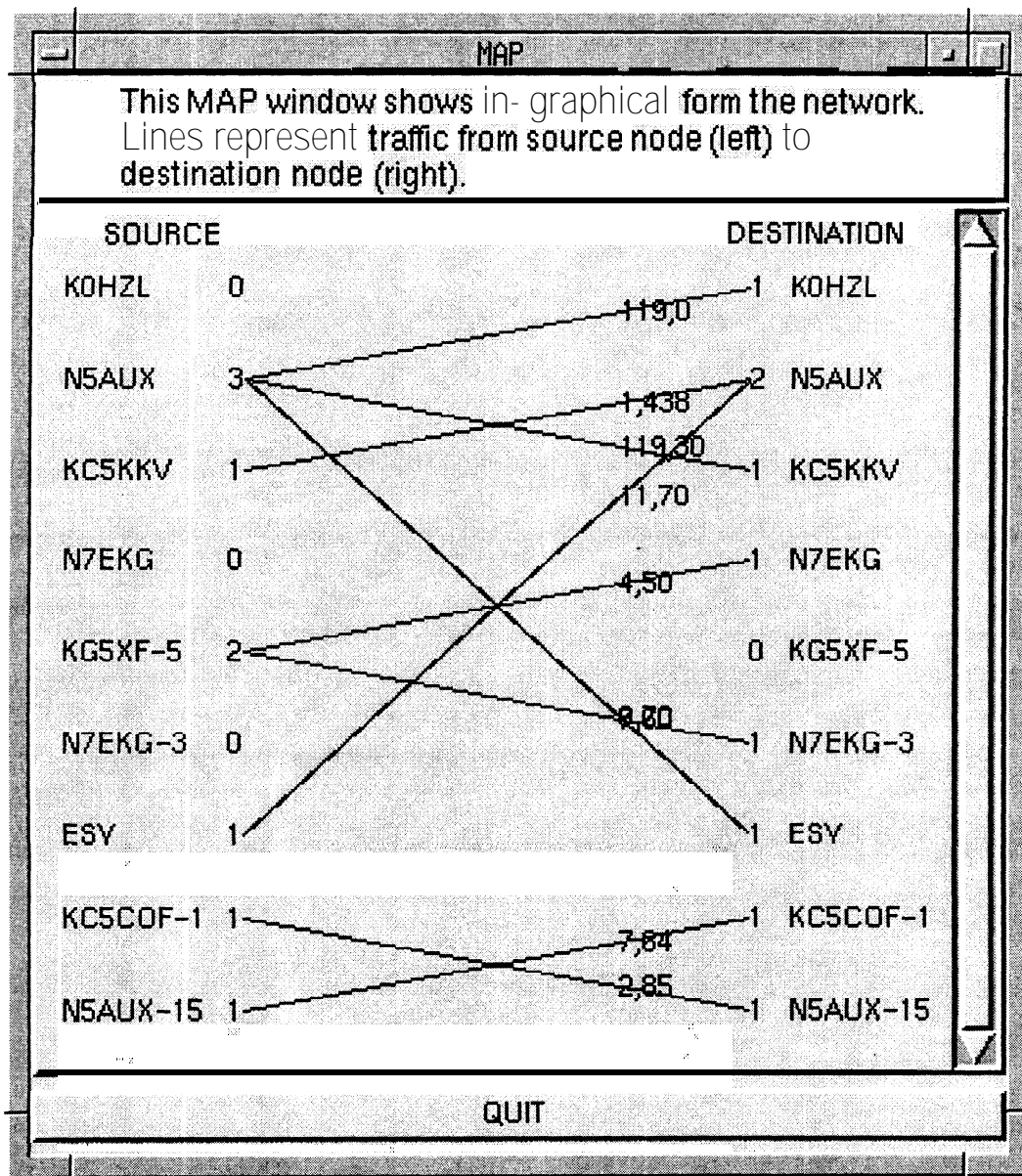

Figure 4. Simple Network

The figures shown below represent packet networks from two different types of networks. Specifically, Figures 4 and 5 depict a normal simple channel network and a APRS packet network respectively. It is important to emphasize that not all nodes are displayed: the map displays only heard stations and stations that have been sent a packet.

In Figure 4. N5AUX sends a packet to node ESY. ESY returned a packet to N5AUX resulting in a connection as previously defined. A pair of blue lines are drawn by XNET to indicate the connection. N5AUX is also connected KC5KKV, shown again by the pair of blue lines. KC5COF and N5AUX-15 are yet a third pair of connected nodes. The remaining nodes are not connected. meaning that packets have been sent, but the destination nodes did not return a packet to the sender.
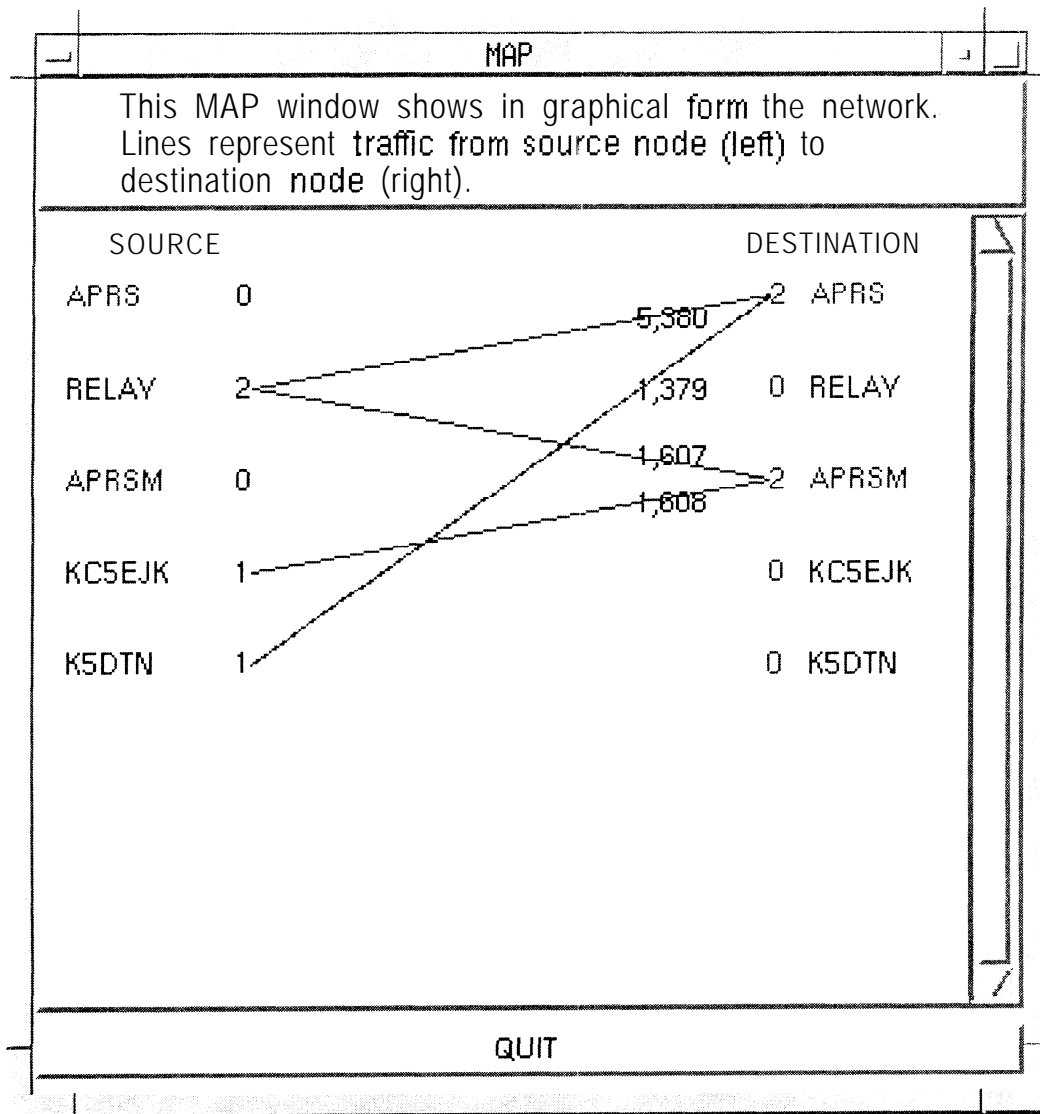

Figure 5. APRS Packet Network

Figure 5 shows an APRS (Automatic Packet Reporting System) Packet Network. As one might expect, there are no AX.25 connections. XNET displays this by showing only brown lines. This is not surprising since all stations in an APRS merely transmit UI (Unnumbered Information) frames at

intervals. The source node does not listen, it only transmits packets which include the nodes present latitude and longitude. They do not request or expect a response. Typically intervals for transmission is 1 minute if the station is mobile, and 60 minutes for fixed station transmitters. These UI frames contain the latitude and longitude of the station.

Those familiar with APRS networks will note the existence of the RELAY and APRS nodes which are used to "repeat" the packet to provide wider coverage for both fixed and mobile nodes on the network.

The reader is encouraged to browse the XNET web page for further examples and for colored map displays which greatly enhances the ease of understanding the network.

NODES WINDOW

The NODES window is intended to provide additional information that is not conducive to displaying in the graphical MAP window. It also displays all nodes, both heard and unheard, while the MAP window is limited to displaying heard and destination nodes only.

Figure 6 shows an example of a NODES window, which shows all currently active nodes. Other network statistics include the number of packets transmitted by that station, network utilization in percent, and the elapsed time since a packet was last sent by that station.

Remember that XNET does not monitor disconnect packets to detect a node disconnect. It determines if a node is active from the age of the last packet transmitted by that node. Detecting disconnect packets could be useful, however, many nodes leave the network, without sending a proper disconnect frame. Therefore another method needs to be used by XNET to ascertain connect status. That method is to *age* the packets and remove nodes that have exceeded the time-out period. The PREFS menu, which we will discuss subsequently, is provided to allow the user to specify the *time-out* period. If the time out period is exceeded, the node is assumed to be disconnected. For example, if the time-out is set to 1 minute and XNET has not heard from that node for 1 minute, the node is assumed to have disconnected from the network and is removed from the window.

STATION This column is reserved for displaying the name of the active stations. Active stations are those heard and unheard stations that have not timed out (e.g., they have not exceeded the time-out period specified by the preferences).

PACKETS (PKTS) The total number of packets transmitted by the specified station is displayed here. Note that the unheard station packet count is always zero since only heard stations transmit packets.

PERCENT (%) This metric represents the percentage of total packets transmitted by the currently active node with respect to other currently active nodes. This value is very useful as it quickly shows which of the nodes is responsible for most of the traffic. This value should not be confused with network utilization. That information is best gleaned from the graph window. In the example, W5XJ has sent 172 packets which represents 97% of the total traffic.

LAST COMMAND (FRAME TYPE) The AX.25 data link layer protocol specifies a type for all frames. Frames fall into three main categories, Information, Unnumbered, and Supervisory. For W5XJ, the last command sent was RR which is a Receiver Ready co&and.

AGE The age indicates the time in hours, minutes, and seconds, since the last packet was sent. For example, W5XJ has not sent a packet for 3 minutes 10 seconds.

MESSAGE (LAST INFO) Information packets contain the useful information that is to be transmitted to the receiving station. The term useful is used here to indicate that this frame contains the information that is of primary interest to the user. Most of the other frames are overhead frames

70

required by the protocol. Only the first few characters of the information string are displayed in this window when it is initially opened. The window may be re-sized to show the entire field if the user wishes.

COLORS Heard stations are shown in blue and unheard stations are shown in brown.
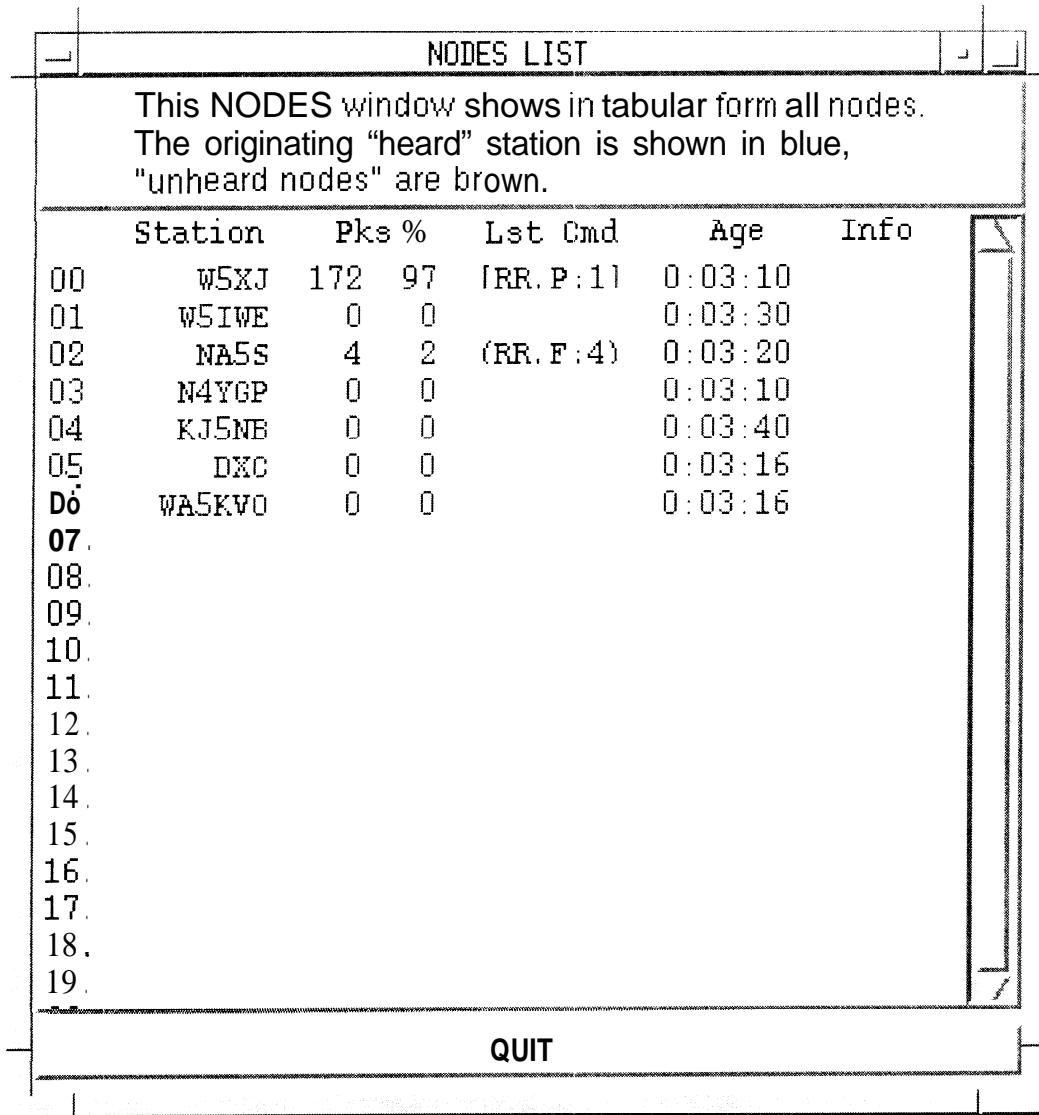
```
┌────────────────────────────────────────────────────────────┐
│ ─                      NODES LIST                      ┘  ┐ │
├────────────────────────────────────────────────────────────┤
│      This NODES window shows in tabular form all nodes.      │
│      The originating "heard" station is shown in blue,       │
│      "unheard nodes" are brown.                              │
├────────────────────────────────────────────────────────────┤
│         Station   Pks %   Lst Cmd      Age     Info    ◥     │
│   00       W5XJ   172  97  [RR.P:1]   0:03:10                 │
│   01      W5IWE     0   0              0:03:30                │
│   02       NA5S     4   2  (RR.F:4)   0:03:20                 │
│   03      N4YGP     0   0              0:03:10                │
│   04      KJ5NB     0   0              0:03:40                │
│   05       DXC      0   0              0:03:16                │
│   D0      WA5KVO    0   0              0:03:16                │
│   07.                                                        │
│   08.                                                        │
│   09.                                                        │
│   10.                                                        │
│   11.                                                        │
│   12.                                                        │
│   13.                                                        │
│   14.                                                        │
│   15.                                                        │
│   16.                                                        │
│   17.                                                        │
│   18.                                                   ◣     │
│   19.                                                   ╱     │
├────────────────────────────────────────────────────────────┤
│                          QUIT                                │
└────────────────────────────────────────────────────────────┘
```

Figure 6. Nodes Window

GRAPH **WINDOW**

The GRAPH window shown in Figure 7, provides channel utilization information. This window- can display channel utilization for 1, 5, and 25 hour periods. The user may freely move from graph to graph to obtain more or less resolution as desired.

The graph should be used as a *relative* rather than *absolute* value of network utilization. Specifically, XNET computes channel utilization based on the number of characters received during a one minute sampling period. The magnitude (height) of the vertical line is also based on the RADIO BAUD rate selected in the PREFS menu. Note that this value is used for computational purposes only and represents the baud rate of transmission over the radio link (not the data rate between the computer and the TNC).
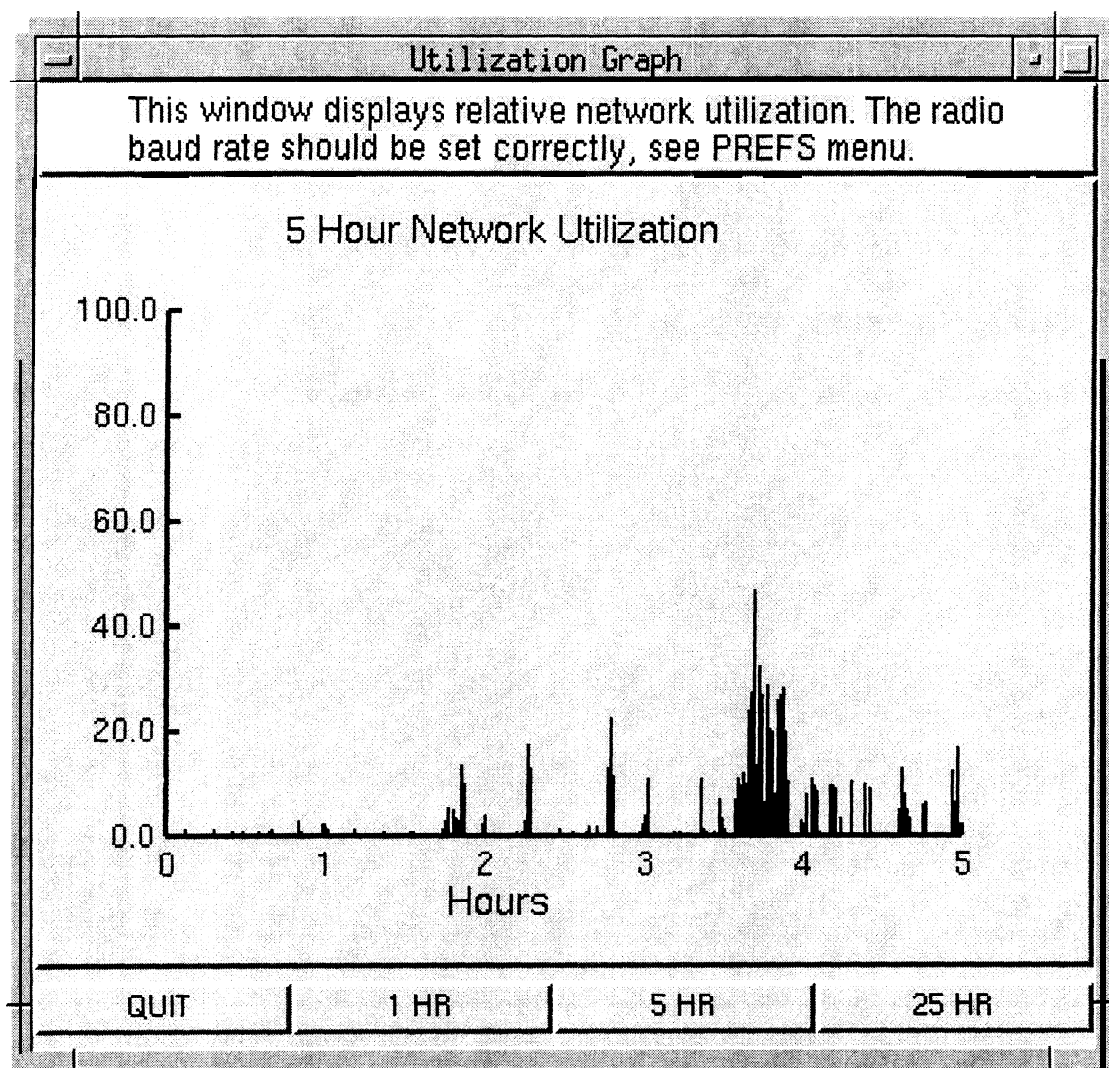

Figure 7. Graph Window (5 Hour Logging)

PREFERENCES

The PREFS window shown in Figure 8, is provided to allow the user to specify miscellaneous XNET preferences. These include the baud rate of the radio link, and various time out periods. The Radio Baud Rate represents the bit rate of the radio link and is used exclusively for the computation of network utilization as displayed in the GRAPH window.

Recall that the program has no certain way of knowing if a node has left the network. Therefore, time-out periods are provided. After this period is exceeded, the node is assumed to no longer exist and is dropped from all displays except the main console which retains the existence of this node in the Total Nodes and Total Packets count. A careful examination of the window shows that XNET provides for three types of time--outs: Heard station, Unheard station, and Connection. These time-out periods may be specified separately. The wide range of time-out periods, 30 seconds to 60 minutes, is provided to allow the user to adjust XNET for the various types of networks that might be analyzed. For example, in normal operation, time-out periods of 1 to 5 minutes are preferred. However, in an APRS network, many of the nodes transmit only once an hour, in which case the user would select a longer time-out period.
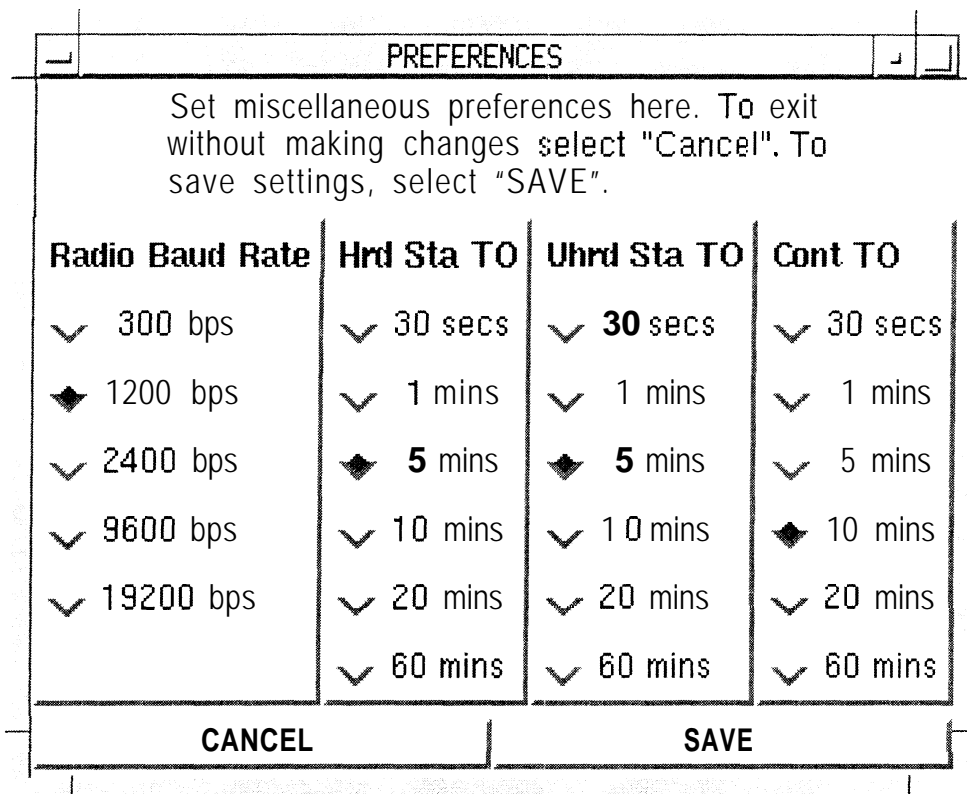

Figure 8. Preferences Window

SIMULATION WINDOW

Although XNET is intended as a graphical program, there are several other windows that may be selected. They are mentioned here briefly to allow the reader to understand some of the program's features.

The simulation preferences window (not shown) is one of three preference windows: SIMUL, PORT, and PREFS. In each of these windows the user is allowed to configure and select preferences. These preferences are saved in a text file (prefs) which is used each time XNET is started.

The SIMUL window allows the user to select between the serial port or one of the pre-recorded simulations. If the serial port is selected, the PORT parameters preference's window needs to be

configured for the desired port parameters such as the baud rate, stop bits, parity, etc. Note that if a simulation is selected, the port parameters have no affect (i.e., they are not needed).

Selecting a simulation is ideal for testing and educational purposes. The simulations are actual prerecorded off the air network traffic. However, the traffic is not played back in real time. The traffic has been designed to send information at a pre-defined interval (1 second in most cases).

TERM WINDOW

The TERM window (not shown) displays raw network traffic. However, the packet is truncated if it exceeds the width of the window. This window is intended to allow the user to see packets as they are received by XNET. Truncating the packets is performed to allow the user to easily see the header of each packet, since that is of most interest.

SERIAL PORT PREFERENCES

The serial PORT configuration window (not shown) is provided to allow the user to specify port parameters. Here the user specifies the port, stop bits, frame size, and parity. Since all information that XNET uses comes from the TNC which provides the information over an RS-232 link, it is necessary that the two devices use the same transfer protocol.

CONCLUSION

XNET is intended to provide information about AX.25 packet networks that is difficult and perhaps impossible to ascertain by other means. The program displays network statistics in both tabular form and an easily understood graphical form. Through the use of graphical displays, color, and an easy to use interface, the program provides an excellent tool for the analysis of amateur radio AX.25 networks.

ACKNOWLEDGMENT

I would like to thank Mark Sproul, KB2ICI, for *Packet Tracker,* a program similar to XNET for the Apple Macintosh. Packet Tracker provided the inspiration for XNET and many of its features have been included.

DISTRIBUTION

XNET is a freeware program available under the GNU General Public License, Version 2, June 1991, Free Software Foundation, Inc. 625 Massachusetts Avenue, Cambridge, MA 02139. It may be downloaded from any of the following sources. Further information regarding the necessary files to download and the system requirements are included there. The present version as of this writing is 1.1. The files are tarred under than name `xnet-1.1.tar` and need to be extracted. The method is described in more detail in the readme file, `xnet-1.1.README`.

The author's home page is the primary download site and source for information.

```
http://www.qualcomm.com/~rparry/xnet
```

The following is another possible download site for XNET and other amateur radio software programs.

```
ftp.ucsd.edu/hamradio/packet
```

REFERENCES

1. Fox, Terry L, *"AX. 25 Amateur Packet-Rudio Link-Layer Protocol,"* Version 2.0 October 1984, American Radio Relay league, Inc. Newington, Connecticut, October 1984.

2. Ousterhout, John K, *"Tcl and the Tk Toolkit,"* Addition-Wesley, Reading. Massachusetts. 1994

3. Sproul, Mark, *"Packet Tracker: A Graphical Packet Tracking Program,"* 12th ARRL Digital Communications Conference, Tampa Florida, September 1993, pp 76-82.

4. Welch, Brent B, *"Practical Programming in Tcl and Tk,"* Prentice Hall, Upper Saddle River, NJ 1995

5. Key Tcl/Tk Web page URLs
   ```
   http://www.sunlabs.com:80/research/tcl/
   http://www.sunlabs.com:80/~rjohnson/tcl/
   http://wwwtios.cs.utwente.nl/~binkhors/tcl.html
   ```

6. Key Linux Web page URLs.
   ```
   http://sunsite.unc.edu/mdw/linux.html/
   http://www.cris.com/~hallow/linux/2.0.html/
   http://www.ddj.com/ddj/1995/1995.05/welsh.htm/
   http://www.geog.ubc.ca/sparclinux.html/
   http://www.redhat.com/linux-info/
   ```

7. Amateur Radio Specific Linux Web page URL.
   ```
   http://www.rahul.net/perens/LinuxForHams/
   ```