

XNET

—

AN X-WINDOWS BASED AX.25 PACKET NETWORK ANALYZER

File Name : ps.main24hours

Title : ps.main24hours

Creator : XV Version 3.00 Rev: 3/30/93 - by Jc

Pages : 1

OPERATING MANUAL

RICHARD PARRY, W9IF

TABLE OF CONTENTS

1. INTRODUCTION.....	90
1.1. ACKNOWLEDGEMENTS.....	90
1.2. DISTRIBUTION POLICY.....	90
1.3. REQUIREMENTS.....	90
1.4. WHAT XNET DOES.....	91
1.5. WHAT XNET DOESN'T DO.....	91
1.6. TNC CONFIGURATION	91
1.7. QUICK START.....	92
1.8. PRINTING	93
2. DETAILED INSTALLATION.....	94
2.1. SOFTWARE.....	94
2.2. PORT SETTINGS.....	95
3. MAIN CONSOLE WINDOW.....	96
3.1. START TIME.....	96
3.2. ELAPSED TIME.....	97
3.3. TOTAL PACKETS	97
3.4. ACTIVE PACKETS.....	97
3.5. TOTAL NODES.....	97
3.6. ACTIVE NODES.....	97
3.7. CONNECT NODES.....	97
3.8. UPDATE PERIOD.....	97
4. MAP WINDOW	98
4.1. HEARD STATIONS.....	99
4.2. DESTINATION STATIONS.....	99
4.3. UNHEARD STATIONS.....	99
4.4. LINES.....	99
4.5. UPDATE PERIOD.....	100
5. NODES WINDOW.....	101
5.1. STATION.....	102
5.2. PACKETS (PKTS).....	102
5.3. PERCENT (%).....	102
5.4. LAST COMMAND (FRAME TYPE).....	102
5.5. AGE.....	102
5.6. MESSAGE (LAST INFO).....	103
5.7. COLORS.....	103
5.8. UPDATE PERIOD.....	103
6. GRAPH WINDOW	103
6.1. GRAPH 1 HOUR.....	103
6.2. GRAPH 5 HOUR.....	103
6.3. GRAPH 25 HOUR	104
6.4. UPDATE PERIOD.....	104
7. TERM WINDOW.....	105
7.1. CLEAR SCREEN (CLS).....	106
7.2. UPDATE PERIOD.....	106

8. SIMULATION PREFERENCES (SIMUL).....	106
8.1. SERIAL PORT.....	107
8.2. SIMULATION	107
9. SERIAL PORT PREFERENCES (PORT).....	108
9.1. PORT.....	108
9.2. STOP BITS.....	109
9.3. FRAME SIZE.....	109
9.4. PARITY BITS.....	109
10. PREFERENCES (PREFS)	109
10.1. RADIO BAUD RATE.....	110
10.2. HEARD STATION TIME-OUT.....	110
10.3. UNHEARD STATION TIME-OUT.....	110
10.4. CONNECTION TIME-OUT.....	110

LIST OF TABLES

Table 1.3-1. XNET System Requirements	90
Table 1.4-1. XNET Features.....	91
Table 1.6-1. TNC Commands to Set Manually	92
Table 1.6-2. TNC Commands Automatically Set by XNET.....	92
Table 1.7.3-1. Quick Start, Simulation.....	92
Table 1.7.5-1. Quick Start, Real Network Connection.....	93
Table 2.1-1. XNET Files	94
Table 2.2-1. Typical UNIX Port Parameters.....	95
Table 2.2-2. XNET Embedded Port Initialization.....	95
Table 5.4-1. Frame Types	102

LIST OF FIGURES

Figure 3-1. XNET Main Window	96
Figure 4-1. MAP Window.....	98
Figure 4.4-1. Connected Nodes.....	100
Figure 4.4-2. Mutually Connected Nodes	100
Figure 5-1. NODES Window	101
Figure 6-1. GRAPH Window.....	104
Figure 7-1. TERM Window.....	105
Figure 8-1. SIMUL Prefs Window.....	106
Figure 9-1. Serial PORT Window.....	108
Figure 10-1. PREFS Window.....	109

1. INTRODUCTION

XNET is a 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.

XNET is written in Tcl/Tk and can operate on any computer system supporting the language and UNIX. The hardware necessary to operate XNET is a radio receiver and a Terminal Node Controller (TNC) which connects to the computer system through an RS-232 port.

This operating manual provides the information necessary to allow the user to properly use XNET. Section 1 provides a quick tour of XNET and fundamental information to get started. Section 2 is provided for those who wish more detailed information regarding the program, and/or, have trouble and need additional information to properly configure the system. Sections 3 through 10 describe each of the major features of XNET.

1.1. ACKNOWLEDGEMENTS

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

1.2. DISTRIBUTION POLICY

XNET is a freeware program. It may be freely distributed on BBS's and the Internet. The distribution consists of a *tar* file, *xNet.tar*. See the software section of this manual for the exact files included in the distribution.

1.3. REQUIREMENTS

Other than packet radio equipment such as the receiver and TNC, the following additional computer components are required.

Table 1.3-1. XNET System Requirements

- Any computer system supporting Tcl/Tk (typically a UNIX workstation)
- Tcl version 7.3
- Tk version 3.6 with *addinput* patch

IMPORTANT: XNET was developed using Tk version 3.6. This version does not support the ability to spawn a process to accept serial data as a background process. Therefore a widely used patch, *addinput*, was developed to fulfill this requirement. Version 4.0 of Tk will permanently solve this deficiency with the *fileevent* command. The new version of Tk is presently in beta testing (circa May 95). There is no projected date for updating XNET to support the new version of Tk.

1.4. WHAT XNET DOES

XNET provides many features that are useful to both the casual packet user as well as a network administrator and the packet radio BBS sysop wishing to better understand a packet network.

Table 1.4-1. XNET Features

- Counts packet traffic
- Collects node statistics
- Graphical representation of network utilization
- Displays raw network traffic
- Includes example simulations of network traffic
- Visual display of network connections
- Extensive use of color and GUI to display information

1.5. WHAT XNET DOESN'T DO

Since data coming to XNET comes from the TNC, some information is lost and/or filtered by the TNC. For example, 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 way of keeping statistics regarding collisions, communication errors, and other similar parameters that the TNC filters out.

Limitations such as those described can be cured by placing the TNC in the KISS mode thereby causing 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.

1.6. TNC CONFIGURATION

The number and variety of TNCs currently available on the market makes a definitive list of TNC parameters to set for XNET operation difficult. Tables 1.6-1 and 1.6-2 show a minimal list of commands that must be issued to insure that the TNC is properly configured. The user should consult the TNC manufacturer's manual for detailed information regarding these commands and configuring the TNC accordingly.

The following commands must be set by the user manually before using XNET. KISS OFF is necessary since we would like the TNC to perform the task of disassembling the packet and providing XNET with uncorrupted ASCII packet strings. HOST OFF is necessary on those systems such as the AEA PK-232 88 which supports a HOST mode.

Table 1.6-1. TNC Commands to Set Manually

- KISS OFF
- HOST OFF

Each time XNET is STARTed, the commands below are automatically sent to the TNC. ECHO OFF prevents characters sent to the TNC from being echoed back. HEADERL OFF prevents the TNC from separating the *header* from the *body* of the packet by printing them on separate lines. MCOM is not supported on all TNCs, however, in those that do support this command, the function should be switched ON. MPROTO is set ON to help insure that all packets are monitored and that none are filtered by the TNC. The MONITOR 6 command is also provided to insure that the TNC accepts and passes all packets to XNET.

Table 1.6-2. TNC Commands Automatically Set by XNET

- ECHO OFF
- HEADERLN OFF
- MCOM ON
- MPROTO ON
- MONITOR 6

1.7. QUICK START

This section is provided to give the user sufficient information to get started quickly. If difficulties are experienced, section 2 should be consulted for additional information.

Ideally, a program should be so well written and intuitive that no manual is required. Hopefully, XNET falls into that category and only the more advanced features of the program require the manual. However, this manual does include information that the user will find useful for both packet radio networks in general and XNET in particular.

- 1.7.1. *tar* the file `Xnet.tar` as shown below. A new directory will be created called `Xnet`. The file `Xnet.tar` may be deleted at this point.

```
Type: tar -xvf Xnet.tar
```

- 1.7.2. Start XNET

```
Type: wish -f xNet
```

- 1.7.3. To help divorce software problems from hardware problems and to quickly gain XNET operating expertise, it is best to first accustom yourself to the program by selecting one of the *simulations* that are provided.

Table 1.7.3-1. Quick Start, Simulation

```
Select: STOP if the program is running
Select: SIMUL button
Select: Chicago 147.090 example replay of actual traffic
Select: SAVE to quit SIMUL and install settings
Select: TERM to open terminal window
Select: NODES to open nodes window
```

Select: **GRAPH** to open graph window
Select: **MAP** to open map window
Select: **START** to begin XNET

- 1.7.4. Trying to select other simulations from the **SIMUL** window will accustom you to the program. Note that before a new simulation is begun, you must first **STOP** the program. XNET provides a reminder should you forget.
- 1.7.5. Assuming the TNC is properly connected and the software is functioning based on the quick simulation test above, you are now ready to monitor network traffic from the TNC through the serial port. A UNIX serial port is a relatively sophisticated device with numerous settings. See the **PORT SETTINGS** section of this manual if you experience difficulties.

Table 1.7.5-1. Quick Start, Real Network Connection

Select: **STOP** if the program is running
Select: **SIMUL** button
Select: **serial Port** to de-select simulation and open serial port
Select: **SAVE** to quit **SIMUL** and install settings
Select: **PORT** button
Select: **/dev/cua3** or whichever port your TNC is connected to
Select: **7** bit frame size or compatible frame size for your TNC
Select: **1** stop bit or compatible stop bits for your TNC
Select: **NONE** parity or compatible parity for your TNC
Select: **SAVE** to quit **PORT** and install settings
Select: **START** to begin XNET

1.8. PRINTING

XNET does not support printing. This was by design, as most, if not all, UNIX systems have excellent printing utilities. As an example, John Bradley's shareware program **xv** allows the user to select any X-window and save the window in a plethora of formats appropriate for printing or including in documentation. All screen shots included in this manual were taken using **xv**.

2. DETAILED INSTALLATION

This section is provided for the user wishing more detailed information regarding XNET. It also provides information that may be useful in the event the user is experiencing installation difficulties.

2.1. SOFTWARE

The XNET directory contains the files listed in Table 2.1-1. Although there are numerous files, the user need only be concerned with the main program, xNet. The following is a brief description of the other files for those wishing a more in-depth understanding of XNET.

The group of files ending with the `.tcl` extension are modules of the main program and were created to ease development and provide a more structured overall program. For example, `xGraf.tcl` is the portion of the program that displays XNET graphs. In a similar vein, `xMap.tcl` displays the graphical map of the network.

The file `prefs` is a small text file that contains the preferences saved by the user. For example, the baud rate, stop bits, etc. are all saved here.

The files beginning with `sim` are the simulation files that were taken from actual off-the-air AX.25 packet traffic. These are UNIX command scripts that are called by XNET rather than using the serial port of the computer for data.

Table 2.1-1. XNET Files¹

-rwxr--r--	1	user1	users	44	May 24	11:16	prefs*
-rwxr--r--	1	user1	users	191	May 24	11:16	sim1.sh*
-rwxr--r--	1	user1	users	58312	May 24	11:16	sim10.sh*
-rwxr--r--	1	user1	users	235	May 24	11:16	sim2.sh*
-rwxr--r--	1	user1	users	503	May 24	11:16	sim3.sh*
-rwxr--r--	1	user1	users	713	May 24	11:16	sim4.sh*
-rwxr--r--	1	user1	users	10760	May 24	11:16	sim5.sh*
-rwxr--r--	1	user1	users	10760	May 24	11:16	sim6.sh*
-rwxr--r--	1	user1	users	41953	May 24	11:16	sim7.sh*
-rwxr--r--	1	user1	users	14221	May 24	11:16	sim8.sh*
-rwxr--r--	1	user1	users	39118	May 24	11:16	sim9.sh*
-rwxr--r--	1	user1	users	6434	May 24	11:16	xGraf.tcl*
-rwxr--r--	1	user1	users	4121	May 24	11:16	xMap.tcl*
-rwxr--r--	1	user1	users	28372	May 24	11:21	xNet*
-rwxr--r--	1	user1	users	3695	May 24	11:16	xNodes.tcl*
-rwxr--r--	1	user1	users	8940	May 24	11:16	xPort.tcl*
-rwxr--r--	1	user1	users	8270	May 24	11:16	xPrefs.tcl*
-rwxr--r--	1	user1	users	4895	May 24	11:16	xSimul.tcl*
-rwxr--r--	1	user1	users	4633	May 24	11:16	xTerm.tcl*

IMPORTANT: All files must be located in the same directory as xNet. Files are loaded and executed only as needed to reduce initial execution response time and to save memory.

¹ The dates and sizes of files are subject to change as versions of XNET change.

2.2. PORT SETTINGS

A UNIX serial port is very flexible and therefore fairly complex. Each of the parameters must be properly set to assure XNET compatibility. The following is a list of parameters for a typical UNIX serial port.

Table 2.2-1. Typical UNIX Port Parameters

```
speed 9600 baud; rows 24; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 -hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
```

XNET provides the user the ability to change only four parameters using the **PORT** preferences window: baud rate, number of stop bits, frame size, and parity. These parameters represent the most common parameters that may need to be changed to assure compatibility with a variety of TNCs.

Four additional parameters are automatically configured by XNET (*echo*, *inlcr*, *igncr*, and *icrnl*). These are important parameters that need to be set as shown in Table 2.2-2. The code is embedded in the file *xNet* and should not need to be changed. Note, when XNET terminates, it does not reset the port to its original values.

Table 2.2-2. XNET Embedded Port Initialization

```
# turn off character echo, receive only.
exec stty -echo < $serial_port

# specify CR and NL characters to accept
exec stty -inlcr igncr -icrnl < $serial_port
```

If the user finds the need to modify the port further, configuration may be accomplished by embedding the changes within XNET as shown above, or by invoking the appropriate UNIX command *stty* before XNET is run.

3. MAIN CONSOLE WINDOW

When XNET first begins, a single window called the *console* is displayed. This console provides the primary interface for the operation of XNET. Assuming that the preferences have been configured correctly for the system, the user may hit START to begin monitoring the network. The main console provides minimal information. Its primary purpose is to allow the user to select additional desired features (i.e., MAP, TERM, NODES, and GRAPH), as well as setting preferences (i.e., SIMUL, PORT, and PREFS).

It is important to note that most of the preferences are frozen once XNET is started (i.e., when the START button is hit). There are important reasons for this and we will revisit this subject later when discussing preferences. XNET will remind the user when a restart (STOP followed by START) is necessary.

File Name : ps.main24hours
Title : ps.main24hours
Creator : XV Version 3.00 Rev: 3/30/93 - by Jc
Pages : 1

Figure 3-1. XNET Main Window

3.1. START TIME

The *real* time and date when XNET was started is displayed here. This time is used for computing *Total Packets* which represents the total number of packets received from the starting time and date.

3.2. ELAPSED TIME

This value represents the time in hours, minutes, and seconds since XNET was started.

3.3. TOTAL PACKETS

The value displayed is the total packets received by XNET from the initial START. This is always higher than the active packet value described next.

3.4. ACTIVE PACKETS

Packets from stations that are actively displayed in the NODES window are displayed here. The difference between ACTIVE and TOTAL PACKETS arises from the fact that stations are removed from the NODES window when they have timed-out resulting in packet traffic being removed from the active packet count. Packets sent by both currently active nodes and past timed-out nodes are represented in *Total Packets*.

3.5. TOTAL NODES

The total nodes that have been monitored from initial start of XNET is displayed here.

3.6. ACTIVE NODES

The currently active nodes displayed in the NODES list window are displayed here. 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 resulting in these nodes being removed from the active node count, while still being accounted for in the total active nodes.

3.7. CONNECT NODES

This value 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.

3.8. UPDATE PERIOD

The clock (elapsed time) is updated every second. The remaining information is updated when a new packet arrives.

4. MAP WINDOW

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. An example display is shown in Figure 4-1. It is important to emphasize that the map displays only heard stations and stations that have been sent a packet, not all nodes are displayed. A further explanation is provided below.

File Name : ps.map3connections.v2

Title : ps.map3connections.v2

Creator : XV Version 3.00 Rev: 3/30/93 - by John Bradley

Pages : 1

Figure 4-1. MAP Window

4.1. HEARD STATIONS

Heard nodes are stations that have actually sent a packet. They are displayed on the map in blue.

4.2. DESTINATION STATIONS

Destination stations are those stations that have been sent a packet. They may or may not be on the network. However, since a heard station has transmitted a packet to that station, it is displayed. These stations are displayed in brown. If the destination station should return a packet, it is promoted to heard station status and the color is changed.

4.3. UNHEARD STATIONS

Other nodes which are on the network, but not heard directly, are not displayed. An example will help clarify this point. In the example packet below, W9IF sends a packet to KC5DSI using two digipeaters (W2IY and WB2BIJ). The heard node in this example is WB2BIJ who actually relays the packet on the local network.

```
W9IF>W2IY>WB2BIJ*>KC5DSI [I;7,6] How are you Christopher?
```

Therefore, the MAP window will show only WB2BIJ and KC5DSI with a line drawn from the former to the latter. The stations W9IF and W2IY will not appear on the MAP at all. The NODES window is used to display that information. One last point, technically speaking KC5DSI is an unheard node (no packets have been sent from it). However, KC5DSI is first and foremost a *destination node* and that status requires it to be displayed on the map to show that a packet was transmitted to it.

4.4. LINES

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. XNET loosely interprets a connection. A more detailed definition follows later in this section.

If a node sends a packet to another node, a line is drawn from the source (left side of MAP window) to the destination node (right side of the MAP window). As Figure 4.4-1 shows, each node is drawn twice, once as a source (left side) and once as a destination (right side). The number next to the node name is the number of connections (i.e., lines) for this node.

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). Take the example, if NODE2 sends a packet to NODE1, a line is drawn from NODE2 on the left (source) to the NODE1 on the right (destination). If NODE1 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 connections. The foregoing scenario is depicted in Figure 4.4-1.

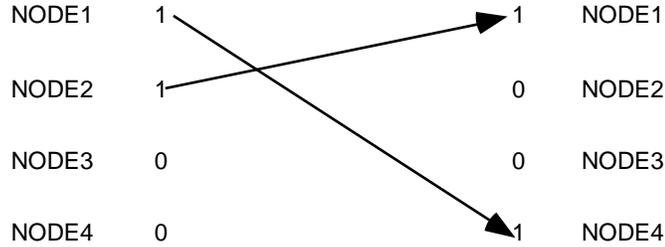


Figure 4.4-1. Connected Nodes

Let's look at another example. If NODE1 sends a packet to NODE2, a line is drawn from NODE1 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 net affect is a connection represented as in Figure 4.4-2. Note that the lines are thicker and are black rather than brown when displayed by XNET.

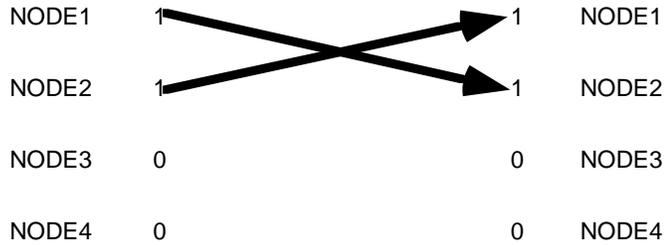


Figure 4.4-2. Mutually Connected Nodes

The case of a connection is vividly shown in the MAP window, Figure 4-1. In that window, N5AUX sent a packet to ESY. ESY also sent a packet to N5AUX, therefore two thick black lines are drawn by XNET to indicate a connection. N5AUX is also connected N5AUX-11, again thick lines are drawn between the two nodes. The MAP shows that WT5Q and N5LDD-1 are yet a third pair of connected nodes. The remaining nodes are not connected. Packets have been sent to several nodes but the destination nodes did not return a packet to the sender, these are shown as thin brown lines.

4.5. UPDATE PERIOD

The information in this window is updated whenever a new packet arrives and at 10 second intervals.

5. 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.

File Name : ps.nodes

Title : ps.nodes

Creator : XV Version 3.00 Rev: 3/30/93 - by John Bradley

Pages : 1

Figure 5-1. NODES Window

This window shows all currently active nodes. Other network statistics include the number of packets transmitted by that station, the network utilization in percent, and the elapsed time since a packet was last received from that station.

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 mechanism needs to be used by XNET to ascertain connect status. That mechanism is to *age* the packets and remove nodes

that have exceeded the time-out period. The PREFS menu is provided to allow the user to specify the *time-out* period, if exceeded, the node is assumed to be disconnected.

5.1. 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 (exceeded the time-out period specified by the preferences).

5.2. 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.

5.3. PERCENT (%)

This statistic represents the percentage of total packets transmitted by the currently active node with respect to the other currently active nodes. This value is very useful as it quickly shows which of the nodes is responsible for most of the traffic.

The value should not be confused with network utilization. That information is best gleaned from the graph window.

5.4. 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. A TNC such as the AEA-232, categorizes the frames as shown in Table 5.4-1.

Table 5.4-1. Frame Types

C.....	Connect Command (also known as a SABM frame)
D.....	Disconnect Command
I.....	Information Command and Response
UA.....	Unnumbered Acknowledgment Response
UI.....	Unnumbered Information Frame
RR.....	Receive Ready Command and Response
RJ.....	Reject Command and Response
RNR.....	Receive Not Ready Command and Response
FRMR.....	Frame Reject Response
DM.....	Disconnected Mode Response

5.5. AGE

The age indicates the time in hours, minutes, and seconds, since the last packet was sent.

5.6. MESSAGE (LAST INFO)

Information packets contain the data that is to be transmitted to the receiving station. 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.

5.7. COLORS

Heard stations are shown in blue and unheard stations are shown in brown.

5.8. UPDATE PERIOD

The information in this window is updated whenever a new packet arrives, and at 10 second intervals.

6. GRAPH WINDOW

The GRAPH window provides channel utilization information. This window shows 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.

XNET computes channel utilization based on the number of characters received during a one minute sampling period. The 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).

6.1. GRAPH 1 HOUR

This graph displays relative network utilization for a one hour period. At the end of the period, data drops off the edge of the graph. Therefore the graph always shows the most recent 60 minute period.

6.2. GRAPH 5 HOUR

This graph displays relative network utilization for a five hour period. At the end of the period, data drops off the edge of the graph. The graph always shows the most recent 5 hour period.

File Name : ps.graf5

Title : ps.graf5

Creator : XV Version 3.00 Rev: 3/30/93 - by John Bradley

Pages : 1

Figure 6-1. GRAPH Window

6.3. GRAPH 25 HOUR

This graph displays relative network utilization for a twenty five hour period. At the end of the period, data drops off the edge of the graph. The graph shows the most recent 25 hour period.

6.4. UPDATE PERIOD

The information in this window is updated at 1 minute intervals.

7. TERM WINDOW

The TERM window displays raw network traffic. However, the packet is truncated if it exceeds the width of the window. This window was intended to allow the user to see packets as they are received by XNET. Truncating the packets was performed to allow the user to easily see the header of each packet, since that is the information of most interest.

File Name : ps.term

Title : ps.term

Creator : XV Version 3.00 Rev: 3/30/93 - by John Bra

Pages : 1

Figure 7-1. TERM Window

7.1. CLEAR SCREEN (CLS)

Hitting this button will cause the display to clear.

7.2. UPDATE PERIOD

The information in this window is updated when serial data enters the port.

8. SIMULATION PREFERENCES (SIMUL)

The simulation preferences is actually one of three preference windows. The windows are: 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.

File Name : ps.simul
Title : ps.simul
Creator : XV Version 3.00 Rev
Pages : 1

Figure 8-1. SIMUL Prefs Window

8.1. SERIAL PORT

This SIMUL window allows the user to select between the serial port or one of the simulations. If a serial port is selected, the PORT parameters preference's window should 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).

You may look upon this preferences window as a way of selecting between a real time network or a pre-recorded network simulation.

8.2. SIMULATION

Selecting a simulation is ideal for testing and educational purposes. Most of 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).

NOTE: An examination of the simulations files will show that these are UNIX script programs that may be easily modified.

9. SERIAL PORT PREFERENCES (PORT)

The serial PORT configuration window is provided to allow the user to specify port parameters. All information that XNET uses comes from the TNC which provides the information over an RS-232 link. It is therefore imperative that the two devices use the same transfer protocol.

File Name : ps.port
Title : ps.port
Creator : XV Version 3.00 Rev: 3/30/93
Pages : 1

Figure 9-1. Serial PORT Window

9.1. PORT

The user is allowed to select from one of the serial ports using UNIX nomenclature for the serial port device.

9.2. STOP BITS

The number of stop bits used by the TNC is specified here, typically 1 stop bit.

9.3. FRAME SIZE

The frame size, 7 or 8 bits is specified here. This is typically 7 bits for most TNCs.

9.4. PARITY BITS

Here the user may select parity: odd, even, or none. In most applications, NONE is the preferred selection.

10. PREFERENCES (PREFS)

This window is provided to allow the user to specify miscellaneous XNET preferences.

File Name : ps.prefs

Title : ps.prefs

Creator : XV Version 3.00 Rev: 3/30/93 - by John Brz

Pages : 1

Figure 10-1. PREFS Window

10.1. RADIO BAUD RATE

The radio baud rate is the transfer rate of data over the radio link. This value does not affect the physical rate of the serial port. The value is used only in the computation of network utilization as displayed in the GRAPH window.

10.2. HEARD STATION TIME-OUT

The NODES window displays all active nodes. These nodes are defined as being either heard (blue) or unheard (brown). The time-out specified here is used to determine when the heard station is assumed to have left the network and therefore is dropped from the list.

10.3. UNHEARD STATION TIME-OUT

This time-out period performs a similar function as the *heard* time-out, but the period is for the *unheard* stations.

10.4. CONNECTION TIME-OUT

Like the two time-out periods discussed, the *connection* time-out determines when nodes are removed. However, this time-out is used to update the MAP window.