HAM WEB NODE

Dale Puckett, KOHYD, and John Bandy, W0UT
23440 W. Highway 54, Goddard, Kansas 67052 U.S.A.
k0hyd@feist.com; john.bandy@twsubbs.twsu.edu

Abstract

This paper describes an existing amateur packet node that provides many graphic and text services to other amateur packet stations in the amateur radio 2 meter band.

Keywords

Amateur radio, packet, http, web node, graphics, digital, Linux, VHF.

Introduction

Packet radio activity in Kansas has declined dramatically in the past several years. This trend which has also shown up in the rest of the nation, can be partially contributed to the immense popularity of the internet and the migration of digital operators to the World Wide Web. The goal of this project is to reverse this trend and bring amateur radio operators interested in the digital domain back to the packet radio fold.

Because of the easy access to commercial web browsers and the popularity of the http medium, we reasoned that an amateur packet node running at 9600 baud would provide an attractive entrance into the http world, if we kept the graphics load relatively low. This concept was proved in practice and we have been able to download a simple html page with several graphics (including a 24K image of a QSL card) in 26 seconds. We have also used the ftp protocol to transfer several large documents each day.

To test the theory, we initially set up two stations -- a Linux based node operating on three frequencies and a client operating on the 9600 baud LAN (145.63 MHz). Both operators use a graphic user interface (GUI) running under Linux. The Web Node operator uses the KDE desktop and the client operator uses X Windows. The long term goal is to let amateur operators running Windows 98 use the TCP/IP protocol over ax25 to communicate with the Web Node. They can then access the ham web node by requesting the nodes URL in their favorite browser. For example, the client station now connects to the node by entering "http://k0hyd.ampr.org/" in his browser address field.

Infrastructure

The KOHYD node operates on three frequencies. The 9600 baud LAN where this experimentation is taking place is on 145.63 MHz. A 1200 baud, 145.01 node provides a link to Kansas packet operators to the West and South of Wichita. A 9600 baud node on 430.55 MHz provides operators on the 9600 LAN access with a high speed connection to Topeka and Kansas City on Kansas Packet Radio backbone. While most TCP/IP operation is conducted on 145.63 MHz, TCP/IP routing has been set up over NetROM on the backbone with BNSBB, NORLR, in Burlington, KS. This allows routing to both Northeast and Southeast Kansas.

NetROM nodes are available on all three frequencies. However, the full node list is only transmitted on the local 9600 baud LAN. In addition to the NetROM services, a full service F6FBB bulletin board is available to operators on all three frequencies. A simple DX Cluster, presently DXNet, is also available on all three nodes. Eventually, the cluster will operate on 145.97 MHz with a tie to neighboring clusters on 430.55 MHz. We are working with DX operators and officers of the Wichita Amateur Radio Club (WARC) to determine the interest level of this service. If interest dictates, we also hope to provide a gateway to the intemet in the future.

The remainder of this paper will detail the services provided and list the configuration files needed by a Linux AX25 operator.

Services

Web pages: HTTP via TCP/IP. /http://k0hyd.ampr.org/ [44.122.0.4].
FTP using TCP/IP: 9600 bps download/upload of graphic files, etc.
Telnet using TCP/IP: Amateurs may run software on host server.
Mail with SMTP and POP3 servers using TCP/IP.
News using TCP/IP.
NetROM Nodes via AX25.
DX Cluster via AX25.
Public Bulletin Board System (F6FBB) via AX25.
All Standard AX25 services.

Hardware

Cyrix P133+ based computer
Four (04) Serial Ports
Kantronics 9612 TNC
Kantronics 9612 Plus TNC
Motorola Mitrek running 50 watts on 145.63 MHz
Yaseu FT-5100 running 50 watts on 145.01 MHz
Tekk KS-1000 on 430.55 MHz
Mirage BD-35 dual band amplifier with 32+ watts out
Comet GP-9 dual band antenna (02)
Duplexer
FLEXI 4XL and RG-213 Coax

Software

Caldera OpenLinux 2.2 operating system
KDE 1.1.1 desktop for Linux
Apache Web Server
F5MZN DxCluster
XFBB F6FBB X Window based Bulletin Board System (PBBS)
LINUX AX25 Kernel Support
AX25 Utilities
sunsite.unc.edu/pub/Linux/.
www.us.kemel.org
www.kde.org

Technical Support

Nate Bargmann, NONB
Caldera Systems www.calderasystems.com
Majordomo@vger.rutgers.edu
  subscribe linux-hams
  subscribe linux-newbie

Helpful books and journals.

Linux AX25-HOWTO Amateur Radio, Terry Dawson, VK2KTJ
OpenLinux User's Guide
ARRL Handbook
Linux Journal
QST Journal
TAPR Packet Status Register

Software Configuration Details

Configuring a full service Linux-hosted node is not for the faint of heart. There are what seems like dozens of configuration files, used by dozens of applications, many of them running in the background as daemons. Yet, if you approach the problem in a logical fashion and do a lot of reading, the plot thickens and everything comes together. The secret is to gather all the information you need to do the job.

In this section we'll attempt to show you the information we gathered together to assemble the full service node described above. By following the examples below and changing the names, callsigns and ip addresses to protect the innocent, you will find it relatively easy to get a Linux station on the air.

We'll take a look at the **axports** file first.

```
# /etc/ax25/axports
#
# The format of this file is:
#
# name callsign speed pacPen  window  description
#
01     KOHYD-13      19200  255    4      145.01 MHz (1200  bps)
63     KOHYD-12      19200  255    4      145.63 MHz (9600 bps)
55     KOHYD-9       19200  255    4      430.55 (9600 KS State Backbone )
pl     KOHYD-11      19200  255    4      Pipe from Node
P2     KOHYD-10      19200  255    4      Pipe to BBS
```

The first thing you'll notice is the fact that each physical port on a Linux AX25 system has its own callsign. If I call another station on 145.01, the system uses the callsign, KOHYD-13. The other callsigns in the table above are used when a station is called on the frequency described in their row. The 19200 is the speed of the rs-232 connection between the computer and the tnc. The ports pl and p2 are special. They represent internal pipes within Linux. pl takes the data from the node. p2 receives that data for the F6FBB BBS.

We used the high SSID numbers for these internal ports because they are not generally used by the end user. They are only used by the Linux system to communicate with other stations. End users will use the NetROM nodes which generally have lower SSID numbers.

Speaking of NetROM, we‘ll look at the nrports file next.

```
# /etc/ax25/nrports
#
# The format of this file is:
#
# name callsign alias paclen description
#
NetROM KOHYD-14      #ICTSW 235    NetROM Switch
netnod KOHYD-1       KSICT 235     Node Console
netbbs KOHYD-3       ICTBBS235     FBB  Full  Service  BBS
netdx        KOHYD-5       ICTDX 235    Experimental  DX  Cluster
```

The names in the first column of this table are the internal names of the four NetROM nodes used at KOHYD. They each use the callsigns in the second column of their row. The name of the nodes broadcast to neighboring NetROM nodes are listed in the third column. The fourth column contains the number of bytes in each packet and the last column contains a short description of each node. Notice here that the **SSIDs** on these callsigns is in the lower range most commonly expected by end users.

Notice here that the node named NetROM is defined as a hidden node by using a pound sign at the beginning of the name. While end users can connect to **#ICTSW,** they can’t do anything else once connected. For this reason, this node is hidden from view. To connect to the node and travel on up the backbone, end users must connect to the Node Console, KSICT. The nrports file works with the **nrbroadcast** file to set up your NetROM nodes.

```
# /etc/ax25/nrbroadcast
#
# The format of this file is:
#
# ax25_name min_obs def_qual worst_qual verbose
#
01     5      190    100    0
63     5      192    70     1
55     5      192    100    0
```

Notice here that the quality for each physical port defined in the exports file may be assigned individually. By placing the def qual of the 55 port higher than the 01 port, the operator forces the station to attempt a connectionon the high speed port before trying on the low speed port. This strategy is required when the same node is available on more than one frequency. If the first connection fails, the node will attempt a connection on the other frequency. Take special notice also of the 0 and 1 in the last column. This column is telling us that the node broadcasts on the 145.63 MHz 9600 baud LAN are verbose. All the nodes available to KSICT are made available to the end users on this frequency. Conversely, the other nodes on the 1200 and 9600 baud backbones do not need this information so they are not set to verbose. Using the non-verbose mode prevents the broadcast of weak, almost non-existent links that are created when a short band opening delivers a node broadcast to the local node.

How does the Linux system know what program to run when another station connects? By using the **ax25d.conf** file. We’ve abbreviated the listing to save space.

```
# /etc/ax25/ax25d.conf
#
```

```
# ax25d Configuration File.
#
# AX.25 Ports begin with a '['.
#
#
[KOHYD-O VIA 01]
NOCALL   * * * * * *  L
default  * * * * * *  -         root  /usr/local/sbin/ttylinkd   ttylinkd
#
[KOHYD-0 VIA 63]
NOCALL   * * * * * *  L
default  * * * * * *  -         root  /usr/local/sbin/ttylinkd   ttylinkd
#
[KOHYD-O VIA 55]
NOCALL   * * * * * *  L
default  * * * * * *  -         root  /usr/local/sbin/ttylinkd   ttylinkd
#
[KOHYD-1 VIA 01]
NOCALL   * * * * * *  L
default  * * * * * *  -         root  /usr/local/sbin/node node
#
[KOHYD-7 VIA 63]
NOCALL   * * * * * *  L
default  * * * * * *  -         root  /usr/local/sbin/axspawn    axspawn %u +
#
[KSICT VIA 55]
NOCALL   * * * * * *  L
default  * * * * * *  -   root  /usr/sbin/node      node
#
# NET/ROM Ports begin with a '<'.
#
<netnod>
NOCALL   * * * * * *  L
default  * * * * * *  -   root  /usr/sbin/node      node
#
#<NetROM>
#NOCALL  * * * * * *  L
#default * * * * * *  -   root  /usr/sbin/node      node
```

The first three entries above tell the Linux software that any station connecting to KOHYD-0 on any one of the three ports will be connected to the ttylinkd daemon which provides a method for chatting keyboard to keyboard. Actually, you repeat each callsign for each port you want to listen on. We have deleted the duplicate entries here to save space.

The next entry shows that any station connecting to K0HYD-7 on the 145.63 MHz port will be connected to the axspawn program. This AX25 Linux Application program spawns a Linux shell for the connected station and lets the operator drive the Linux computer.

The next line shows that the system will listen for the node name KSICT on the 430.55 MHz port. When it hears a station, it will connect it to the node program. This program emulates the familiar interface users expect when the connect to a NetROM or theNet node.

Notice that only the netnod and NetROM ports are used in the ax25d.conf file. The others are left out of this file to prevent the ambiguity caused by the fact that the XFBB BBS software is also listening for these nodes on the same physical hardware port. The XFBB and the DxNet software use the nodes named ICTBBS and ICTDX. Notice also that the callsign SSID used by the FBB BBS software, K0HYD-3, cannot be used by any other software. For this reason there are no listeners in the ax25d.conf file for KOHYD-3. Here is the operative part of the FBB port.sys file.

```
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
 0   0    0   0      0     0     0     0      00/01   ----  File-fwd.
 1   8    2   63     250   2     1     10     30/60   XUWY  145.63
 2   2    2   01     250   4     1     10     00/60   XUWY  145.01
 3   6    2 netbbs   250   4     1     10     15/60   XUWY  NetROM
 4   6    2   p2     250   4     1     10     15/60   XUWY  ViaNode
#
```

in the fourth row of this table the FBB software is told to listen to the netbbs device. This NetROM device goes by the name of ICTBBS.

Here is the corresponding **DxNet config** file.

```
SET LANGuage english
SET/QTH Goddard, KS
SET/HOME ICTDX
SET/LOCAtor EM17EN
SET/call kOhyd-5
SET/ssid +5
SET/port 01 55 63 netdx
```

Notice here that this software presently listens on all three ax25 ports and on the netdx port which goes by the node name of ICTDX.

To connect to the BBS software when connected to the KSICT node, the user issues the BBS command. It is set up in the **node.conf** file. A few of the key lines are listed here.

```
HostName      kOhyd.ampr.org
LocalNet      44.122.0.0/24
Alias           BBs      "c pl kOhyd-3"
NodeId        #ICTSW:KOHYD-14
NrPort        NetROM
```

The third line above shows that when a node user types "bb" the Linux system will use the pl port, a Linux pipe defined in **axports,** to connect him to KOHYD-3 which is the only caiisign the BBS uses.

Getting it all started

If you started up your Linux based amateur station manually, it could turn into quite a job, require a great memory and take a lot of time. The solution to this dilemma is to use a Linux shell script. We'll highlight the command needed to bring your station on the air in the listing below.

```
#!/bin/sh
# /etc/ax25/start
# This file starts the ax25 Programs
# First, the NetROM device must be brought down so they may be created again
echo "Beginning the ax25 setup ..."
echo "Erase the ifconfig nrO, nrl, nr2 and nr3 entries ..."
ifconfig nr0 down
ifconfig nr1 down
ifconfig nr2 down
ifconfig nr3 down
# Next, you must attach the ax.25 KISS interface to ttyS1
# We areusingtwo, dual port TNCs at this station.
# Two pseudo ports, one for each TNC channel are created first.
# The first TNC is on ttyS1 (COM2)
```

```
echo "Attaching ax.25 interfaces ... KPC-9612"
mkiss -s 19200 /dev/ttyS1 /dev/ptyq0 /dev/ptyq1
sleep 2
# Then, we attach each port.
kissattach -i 44.122.0.4 -m 512 /dev/ttyq0 01
kissattach -i 44.122.0.4 -m 512 /dev/ttyq1 63
# And, set the desired operating parameters.
kissparms -p 01 -t 250 -s 200 -r 25
kissparms -p 63 -t 90 -s 90 -r 25
# The second TNC is on ttyS0 (COM1)
# Attach the ax.25 MISS interface to ttyS0
echo "Attaching ax.25 interfaces . . . KPC-9612 Plus"
mkiss -s 19200 /dev/ttyS0 /dev/ptyq2 /dev/ptyq3
sleep 2
# kissattach -i 44.122.0.4 -m 512 /dev/ttyq2 XX
kissattach -i 44.122.0.4 -m 512 /dev/ttyq3 55
# kissparms -p XX -t 250 -s 200 -r 25
kissparms -p 55 -t 90 -s 90 -r 25
# Now set up Alias from BBS through Node
# We use a Linux pipe to accomplish this.
sleep 3
kissnetd /dev/ptys1 /dev/ptys2 &
sleep 3
kissattach -i 44.122.0.4 -m 512 /dev/ttys1 pl
kissattach -i 44.122.0.4 -m 512 /dev/ttys2 p2
# Configure the IP Routing over AX25
echo "Now, configure the IP Routing ..."
route add -net 44.122.0.0 netmask 255.255.255.0 ax0
route add default ax0
route add pc.k0hyd eth0
ifconfig ax0 44.122.0.4 hw ax25 k0hyd-13
ifconfig ax0 broadcast 44.122.0.255 netmask 255.255.255.0
ifconfig ax1 44.122.0.4 hw ax25 k0hyd-12
ifconfig ax1 broadcast 44.122.0.255 netmask 255.255.255.0
ifconfig ax2 44.122.0.4 hw ax25 k0hyd-9
ifconfig ax2 broadcast 44.122.0.255 netmask 255.255.255.0
ifconfig eth0 broadcast 44.122.0.255 netmask 255.255.255.0
echo "Start All Services ..."
sleep 3
# Configure NetROM with TCPIP
nrattach -i 44.122.0.4    NetROM
nrattach -i 44.122.0.4    netnod
nrattach -i 44.122.0.4    netbbs
nrattach -i 44.122.0.4 netdx
ifconfig nr0 broadcast 44.122.0.255
ifconfig nr1 broadcast 44.122.0.255
ifconfig nr2 broadcast 44.122.0.255
ifconfig nr3 broadcast 44.122.0.255
# We have one preferred neighbor on the backbone
# The route to him is locked.
echo "Form a locked route table ..."
nrparms -routes 55 n0kta-1 + 198
sleep 3
# Configure Routes to Individual TCPIP Stations
echo "Configure routes to TCPIP stations ..."
arp -H ax25 -s 44.122.0.2 wf0a-10
arp -H ax25 -s linux.n0nb n0nb-8
arp -t NetROM -s n0rlr ka0oxh-7
nrparms -nodes n0rlr-2 + BNSBB 192 6 55 n0kta-1
route add n0rlr nr0
route add wf0a ax0
route add w0ut ax1
```

```
sleep 4
# Now that most of the configuration has been initialized,
# we must start all the daemons that run in the background
ax25d &
sleep3
netromd &
sleep 3
mheardd &
sleep 2
echo "Load the old Node Table"
/root/nodes.save
echo "Associating Users to callsigns ..."
axparms -assoc w0ut w0ut
axparms -assoc n0kta n0kta
echo "Now issue beacon to tell them we're here ..."
beacon 01 -d CQ -t 15 "KSICT [KOHYD-1], Full service Linux NetROM/TCPIP node
in Goddard"& beacon 63 -d CQ -t 15 "KSICT [KOHYD-1], Full service Linux
NetROM/TCPIP node in Goddard%
beacon 63 -d CQ -t 15 "KSICT [KOHYD-1], Full service Linux NetROM/TCPIP node
in Goddard% beacon 63 -d CQ -t 15 "KSICT [KOHYD-1], Full service Linux
NetROM/TCPIP node in Goddard"&
sleep 5
echo "All AX25 services should now be configured."
```

All of the configuration files used at KOHYD.AMPR.ORG are being made available for http download at http://www.fn.net/~k0hyd

Acknowledgements.

The authors are grateful to Nate Bargmann, N0NB, who built the first Linux AX25 Node in our community and converted the surplus Motorola Mitrek VHF mobile transceiver to the 9600 bps data radio.

We were also assisted by Joe Johnson, NOKTA, Richard Rose, WFOA, and others.

Reference.

[I] Jeff Tranter, VE3ICH, “Packet Radio Under Linux”, Linux Journal, SSC, Seattle, WA, Issue #41, pp.44-47, September 1997, phone 206-782-7733.

[2] Fred Treasure, "NF/Observatory Networking with Linux OS”, Linux Journal, SSC, Seattle, WA, Issue #34, pp. 14-17, February 1997, phone 206-782-7733.

[3] Dale Puckett, KOHYD and John Bandy, WOUT, Ham Web Station, Submitted to 18th ARRL and TAPR Digital Communications Conference, Phoenix, AZ.

(4) MVFMA AX.25 Configuration Files at http://www.febo.com/packetnet/mvfma.html

Trademarks.

Linux is not a trademark, and is not connected to UNIX or X/Open. The X Window System is a registered trademark of the Massachusetts Institute of Technology.

No Warranty.

This material is provided “as is” and without any expressed or implied warranties, including, without limitations, the implied warranties of merchantability and fitness for a particular purpose.

Downloadable from http://www.fn.net/~k0hyd/ax25server.tar.gz