

Linux FPAC mini-HOWTO

Bernard Pidoux, F6BVP, f6bvp at amsat dot org

v2.0, 2012-October-9 [**This version starts a new documentation cycle taking progressively into account Mageia distro, RaspBerry Pi project and Linux kernel 3.x**]

Packet radio is a way of connecting computers via Amateur Radio. This mini-HOWTO covers the installation and configuration of an Amateur Radio packet switch FPAC under Linux operating system.

The reader will be guided step by step through the process of configuring a FPAC node capable of automatically reach a destination via a network of interconnected node stations (either via radio or Internet).

To illustrate the power of FPAC combined with Linux ROSE kernel module I give here an example of a connection to a remote station through [three different routes and protocols](#) depending on destination callsign.

Another example explains how Linux kernels achieves frame routing via [ROSE module through rose route and rose nodes tables set up by FPAC](#).

The user will be responsible for the HF radio interfaces configuration **not described extensively** here. However some examples and links will be provided.

If your Linux box is setup with a radio device, FPAC can provide inter connections between local packet-radio networks and remote nodes using [AX.25](#) Packet Layer Protocol (AX.25 PLP) RATS Open Systems Environment, [ROSE](#).

A network of FPAC nodes can transparently interconnect many applications such as BBS stations for bulletin or message forwarding, DX spots, and Terminal Chat user connections, etc..

For a complete description of FPAC software, please read this [FPAC HOWTO](#) (or from this [mirror](#)).

This Linux FPAC mini-HOWTO document is also available in PDF format here :
<http://rose.fpac.free.fr/MINI-HOWTO/FPAC-MINI-HOWTO-2.00.pdf>

1. [Introduction](#)
 2. [How to configure a Linux 2.6 kernel with AX25 modules](#)
 3. [How to install AX25 libraries](#)
 4. [How to install AX25 tools](#)
 5. [How to install AX25 apps](#)
 6. [A script for various directories and file rights setup](#)
 7. [How to configure ax25ipd](#)
 8. [How to install and configure FPAC 3.27](#)
 9. [AX25 startup scripts](#)
 - 10.1 [Router and Firewall configurations](#)
 - 10.2 [Interfacing FPAC with application programs](#)
 - 10.3 [Other Network Utility programs](#)
 11. [More HOWTOs](#)
 12. [Getting Help](#)
-

1. Introduction

FPAC for Linux was written by Jean-Paul, F6FBB. An earlier FPAC version , written for DOS, was coded in assembly language by Antoine, F6DWJ, in collaboration with Rémy, F6ABJ. FPAC is based on [Thomas A. Moulton W2VY](#) implementation of the [Radio Amateur Telecommunications Society](#) (RATS) Open Systems Environment (ROSE), based on X.25 CCITT standard, in assembly language and "C" for packet radio Terminal Node controller TNC2. Jean-Paul also wrote the [FPAC HOWTO](#) (or from this [mirror](#)) which covers the operation and setup of the FPAC node. An [FPAC sysop's manual](#) has also been written by Charles Schumann, K4GBB.

The purpose of this document is to assist the reader in the setup of an Amateur Radio Packet node from scratch. The process is based on my own experience using a Mandriva Linux distro. Before starting here are a few recommendations. Avoid using a brand new hardware with the most recent high performance graphic adapter or sophisticated Ethernet interfaces included on the mother board. Linux drivers like nVidia are proprietary and may not be included in Linux kernel. You must know that Linux is very performant and does not need high frequency microprocessors as opposed to a well known commercial operating system. Take the opportunity to recycle your old computer rejected by your children for low performance reasons. You will be surprised how fast Linux boots on this "old" machine.

With Linux 2.2 kernels, the last available being 2.2.26, AX25 applications were quite stable. It managed a packet radio system for weeks without needing intervention. However, you may want to use more recent Linux kernels which contain support for USB interfaces, serial ATA disks, Blue Tooth, and wireless LAN. Due to extensive changes in the kernel code, AX25 was not as well supported on 2.4 and early 2.6 kernels. Although you can run AX25 with kernel 2.4, Linux system often becomes unstable and will lock-up. Ralf DL5RB and other Linux network developers worked intensively on AX25 code adaptation to 2.6 kernel. Versions number higher than 2.6.30 are fine to run HamRadio AX25 applications. With Mandriva distro you have a Linux system ready to run AX.25 applications in a few steps.

Previous versions (1.77) described step by step how to set up a packet radio FPAC node switch using Mandriva Linux distro considered as one of the most user friendly as well as being highly professional. This is the one I have been using for more than ten years. However, Mandriva has now been forked and a lot of users have adopted new distro called [Mageia](#). I will progressively update this page in order to take this into account. There has been another major event this time about hardware, since RapsBerry Pi project has produced an ultra low cost Linux computer (35 US\$) that runs under Linux and on which I could install and run an FPAC node. Charley K4GBB came with a very complete documentation file here : <http://k4gbb.no-ip.info/docs/Raspberry.html>

A step by step procedure to set up an FPAC node, based on [Mageia Linux distro](#) with 3.x Linux kernel, will cover :

1. Installation of Linux 2.6 kernel with AX25 modules
 2. Installation of AX25 libraries
 3. Installation of AX25 tools
 4. Installation of AX25 applications
 5. Configuration of the ax25ipd daemon
 6. Installation of the FPAC software
 7. Installation of AX25 start-up scripts
 8. Setting up the FPAC switch.
-

2. How to configure a Linux 3.x kernel with AX25 modules

- Installing Mandriva Linux is out of the scope of this MINI HOWTO. In order to familiarize yourself with Linux, there are two Live version. One is available for free on a bootable CD. [Mandriva Linux live distro is called "One"](#). There are different medias (CD ISO images) for the 2010 "One" edition for different CPU architecture (i586 for 32 bits CPUs or x86_64), including various selection of languages, with either a KDE or GNOME desktop. Download the version you need from an [FTP mirror site](#), most popular version is probably the Mandriva One Free KDE iso. Burn the CD and boot it. This one of the fastest and simplest way to get a working Linux with a recent kernel including AX25 capability. See Mandriva wiki pages for docs about [installing Mandriva Linux](#).
- **Mandriva One live version** is not a distro to run FPAC application. Use it to familiarize yourself with Linux. Due to the lack of compiler or other development tools, you could not install FPAC.
- Mandriva One is also available on a **2 GB USB pendrive** with either 32 bits Linux with KDE or Gnome desktop. This key is easy to carry and allows you to install Mandriva linux on a Netbook or a PC with no CD drive. Price is **29 US\$** or 19 Euros. Mandriva Linux One is a full Linux operating system on a simple key for both new and experienced Linux users, it is fast to install and also safe to try in live mode.
- Mandriva is also selling 59 US\$ or 49 Euros an **8 GB memory USB key bootable with a live Linux distro called Mandriva Flash**. You can find a presentation of [Mandriva Flash product on this Mandriva page](#). This is also a **live Linux distro**, meaning that it is fully installed and bootable from the USB key. It does not need a hard disk and uses your computer's RAM memory. There is a large space left on a partition for saving personal data. I wrote a special [FPAC Linux Flash Mini Howto](#) (outdated) to describe installation of FPAC AX25 packet router, FBB BBS and other AX25 tools and applications on this Flash key. In conjunction with a Laptop computer, this is a very nice mean to obtain a highly portable AX25 system configuration.
- Or if you already know Linux a bit, and want a more complete system with development and server options (nfs, ftp etc..) including compile facilities, you can install a full Mandriva Linux free distro. In my experience this is one of the easiest Linux distro to install. It can be freely downloaded from a [Mandriva mirror site](#). It is available as a 4 GB DVD ISO image to be downloaded via FTP or BitTorrent. There are two versions depending on the Linux environment you want : either a 32 bits or a 64 bits Linux. If your Linux machine has a 32 bits processor (like Intel 586 or 686) you cannot install x86-64 version, while if your CPU is a 64 bits processor, you can install any version. However, 64 bits Linux will probably be more performant (faster) than 32 bits Linux.
- To get **Mandriva Linux**, download the corresponding ISO file and burn the DVD. Then boot your computer from the DVD. The installation is straightforward. You should select **advanced** or **personalized** install options, when being in a menu proposing KDE or Gnome desktop. Then from this advanced menu select at least : **development, network servers and network client** and other options of your choice. Near the installation end you will be asked for **root password**, a **user name** and **password name**. Later it will be useful to create another user account with the name **'fpacuser'** , password **'fpac'** and shell **/usr/local/sbin/fpacnode**. We shall see why.

During the installation, there are two important things to complete: graphic configuration and network configuration. Be sure that graphics are well configured by performing the proposed test. It is not a good idea to use DHCP protocole to configure your local network address. Set preferably a fixed IP address to your Ethernet interface. Example **192.168.0.200**. Create a user account for daily use. However, AX25 applications need to be started and run under Supervisor User account. Login as **root** is thus necessary to run AX25 applications. However it is not possible to run in Graphic User Interface (GUI) mode like KDE or Gnome under a root login, but as a

user. Then you need to open a Terminal and log in as **SU**.

- Hamradio devices are most often connected through serial interfaces to your computer. Data and software device control are using serial ports like ttyS0 or pseudo TTY devices. **Legacy pseudo TTY system is now obsolete and replaced by Unix98 standard.** Recently AX25 libraries, AX25 applications and AX25 tools has been converted to new standard and will be used in this MINI HOWTO. If you still want to use legacy pseudo TTY it is necessary to instruct Linux kernel to create them at boot time. However the new standard is better for it is using much less memory and you can skip the next two paragraphs.

1. If you have good reasons to keep using old pseudo TTY standard follow these instructions. Be aware that some kernels do not support it. Edit **/boot/grub/menu.lst** file with vi editor in a terminal console. Command is : **vi /boot/grub/menu.lst**
2. Duplicate paragraph beginning with **title linux**. vi editor syntax is the following : move the cursor to the **title** line. Then type **4** and **yy**. Then move the cursor to the empty line created after the new paragraph. Type **p** and edit the new paragraph entry name (command **i** for insert). For example, replace '**title xx**', by **title xx-custom**. At the end of line beginning with kernel, add **pty.legacy_count=10** then save the file (<escape> : **wq**) and reboot. Be carefull not to write on next line with **initrd** or you would get a boot failure with a kernel panic. Believe me, I know what I am talking about.

- Here I strongly recommand to add **maxcpus=1** option into grub boot menu to instruct Linux kernel not to run on uniprocessor mode even if the kernel was compiled for multi core and your CPU is a dual core or more. **Runing only one CPU is needed until all AX25 libraries and application programs are ready to run on SMP kernels.** This should replace **nosmp** argument, that also instructs kernel to run on uniprocessor mode but triggers at the same time an instruction for kernel not using ACPI, that may lead to bad interrupts handling and system instability.

Here is an exemple of */boot/grub/menu.lst* entry with **maxcpus=1** and **pty.legacy_count=10** boot options

```
title desktop 2.6.34-maxcpus1
kernel (hd1,4)/boot/vmlinuz-2.6.34-desktop-2mnb BOOT_IMAGE=desktop_2.6.34 root=UUID=7d268bb1-5045-4ccc-92b7-28a228f7a484 maxcpus=1 pty.legacy_count=10 resume=UUID=4ee3102a-966b-4bb8-8bd5-53d37cf5a2e8
splash=silent vga=788
initrd (hd1,4)/boot/initrd-2.6.
```

- After rebooting using **linux-custom** entry, if you have chosen to create Legacy PTYs you may check for the existing pseudo TTYs by listing **ls /dev/ptyp*** or **/dev/ttyp***. Ten PTY (master and slave) pairs were created and will appear under names **/dev/ttyp0** to **9** and **/dev/ptyp0** to **pty9**.
- Linux operating system with 2.6 kernel is well suited for AX25. It has all AX25 already enabled with AX25 modules and drivers (ax25, rose, mkiss, 6pack, netrom, etc..already compiled). Before running AX25 applications, you need to install AX25 libraries, tools and apps.
- If you have serious reasons to build a custom kernel, you need to download a kernel source package from kernel.org and compile the kernel after selecting the modules needed by your computer configuration. Instructions for compiling a kernel are documented in [the Linux Kernel HOWTO](#).
- Latest stable kernel source may be found at [kernel.org site](http://kernel.org) or at <http://www.eu.kernel.org/> from Europe. From there find the latest kernel file on line. For example Latest Stable kernel 2.6.37.1
- Click onto 2.6.37.1 field to download complete source package. Or click on [Patch] to download the latest patch, or [View Inc.] for incremental patch. If you click onto [Full source] you get for

example a link to: <http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.37.1.tar.bz2>

- Open a console Terminal (Konsol window if you are under KDE) on your Desktop. Switch to a root session, by entering command **su** and enter the **root password** you provided when you installed Mandriva Linux.

Move downloaded file into **/usr/src/** directory and enter this directory with **cd /usr/src** command

Unpack the archive :

tar xf linux-2.6.37.1.tar.bz2 this will create linux-2.6.37.1 directory.

Create a symbolic link **linux**

ln -s linux-2.6.37.1 linux

Create the two following symbolic links to linux that you can use later when you download and apply a kernel patch.

ln -s linux a

ln -s linux b

- You are now ready to configure your custom kernel. For more help consult the [LINUX KERNEL HOWTO](#). Open a console terminal and switch to a root session by typing the command **su** to log in as root and enter the root password. Next, change directories, **cd /usr/src/linux**. If this is the very first time you compile the kernel source since you installed your Mandriva Linux distro, just type **make oldconfig**. This will copy the present hidden **.config** kernel configuration file from **/boot** into linux directory. This file is only visible via command **ls -la**. By the way, it is recommended to **make oldconfig** whenever you update Linux kernel using a kernel patch, or when you download a new kernel version source and want to use **.config** from a previous version. In that case, copy hidden **.config** file from old Linux directory into new linux directory. Doing that followed by **make oldconfig** you will avoid a fastidious complete custom configuration each time you apply a kernel patch or download a new kernel source. You will be eventually asked to validate items that appeared or changed between the old and the new Linux kernel version. In most cases the default option may be accepted by pressing <Enter> key. At the end of the **oldconfig** process jump to start kernel compilation paragraph below.
- Or you may want to build a customized kernel for the first time or change previous options, then enter **make menuconfig** command to start kernel configuration menu that will allow you to select or deselect some items for a custom kernel. You could also build a graphical menu with command **make xconfig** on the command line in a console terminal from **/usr/src/linux** directory. A menu will pop-up and you will be able to select or de-select the different items, depending on the hardware present on your computer. Some items are selected by default. This **xconfig** command needs ncurses package, QT3 and QT3 development to be installed (run command "**urpmi qt3 qt3-devel**" from a Konsol terminal or use Mandriva Control Center MCC to install QT3 development package). Compiling the kernel plus all modules can take a lot of time on a slow computer. For example, it can take a few hours on a 500 MHz Pentium III. You will start the script for building a kernel configuration menu with the command **make menuconfig** or **make xconfig**. In **processor type and features** menu it is recommended to unselect [] Tickless System (Dynamic Ticks). For Linux an FPAC server you want to select **timer frequency 250 Hz** option. If you compile more modules than actually needed, Linux will find the necessary drivers and load them at boot time. To compile AX25 modules, you need to select the sub menu Networking and enable **Amateur Radio Support**. Enabled modules are indicated by a letter **M** (menuconfig) or by a dot (xconfig). To disable an item, type **N** (menuconfig) or click on the item box (xconfig). Select sub menu **Amateur Radio Support** and enable **all the protocols**. Select the **AX25 Network Device Drivers** menu and enable **All items** : serial port KISS driver, serial port 6PACK driver, BPQ Ethernet driver, Highspeed (DMA) SCC driver for AX.25, Z8530 SCC driver, all the BAYCOMs, YAM driver for AX.25.
- Once again, if you prefer to use old legacy pseudo ttys method in menuconfig enter submenus

Device drivers and Character devices. Localize line **Legacy (BSD) PTY support. On next line there is (0) Maximum number of legacy PTY in use. Press <Enter> and replace the 0 number by 10. That should be sufficient as it will create ready to use 10 ptys pairs.**

- Save the configuration by answering **yes** at the end of the config menu.
- Start kernel compilation with the following command from linux directory :

make

- The compilation will take a while depending on the number of files to be compiled and the speed of your CPU. When compilation is done type the following commands :

make modules_install

make firmware_install (with kernel 3.x)

make install

- If everything went well you can now reboot your system. The choice for the new kernel will appear on the GRUB boot menu after the line Linux 2010.2 (default boot) on a line showing **Linux with 2.6.36 kernel** (or the name of the kernel you just compiled). If the system reboots correctly you will have now a Linux system with AX25 and ROSE modules up to date. You can verify the availability of AX25 modules by loading some of them from a terminal console :

modprobe ax25

modprobe rose

modprobe mkiss

- Command **lsmod** will display the driver list.
- When you are satisfied that the new kernel is working you can make it the default boot kernel. Using in the *Mandriva Control Center (MCC)*, the system configuration tool.
- For the next steps, use a character mode console. It is more convenient than using the Graphic User Interface (GUI). To open a graphic console press simultaneously the three **CONTROL**, **ALT** and **F1** keys. Log in as root and provide the root password. You can open up to 6 console sessions and switch between them by pressing **ALT** and **Fn** keys, with 'n' being between 1 and 6 (**ALT+F1**,..., **ALT+F6**). You may also return to graphic console by typing both **ALT** and **F7** keys (or **ALT** and **F8** keys) simultaneously.
- We are now ready to install AX25 libraries.

3. How to install AX25 libraries

Like many other AX25 applications FPAC needs the AX25 libraries. When running various AX25 applications we call common subroutines that are included into ax25 libraries.

*I have patched [official package source](#) of **libax25-0.0.12-rc2** package to remove some bugs.*

3.1 Using RPM package to install AX25 libraries

The easiest way to install AX25 libraries is by using an RPM archive package. I have build two packages depending on your Linux installation. If you installed a 32 bits Linux (usually for i586 or

i686 CPU) you should download [libax25-0.0.12-1mdv2010.1.i586.rpm](#) archive file, while you should download [libax25-0.0.12-1mdv2010.1.x86_64.rpm](#) archive for a 64 bits Linux environment.

Here are the steps to follow once you have downloaded either of the above archives. The principles are the same for both archives install. First, you need to be root in order to perform archive install.

cd into the libax25 package directory and enter the command :

rpm -ivh libax25-0.0.12-1mdv2010.1.i586.rpm

or enter the command **rpm -ivh libax250.0.121mdv2010.1.x86_64.rpm**

You should get a message like :

```
Preparation... ##### [100%]
 1:liblibax25_0 ##### [100%]
```

Due to my quite new experience with RPM packaging, AX25 library will not be installed in the final working directory :-(

Enter the following commands : **mv /usr/lib/libax25* /usr/local/lib/**

You can then list the AX25 library files : **ls -la /usr/local/lib/**

Ten lines will be displayed, with the same file names as below, but with probably different sizes.

In case of **x86-64.rpm** package the location of libraries will be slightly different and the command for moving the files to

/usr/local/lib also different : **mv /usr/lib64/libax25* /usr/local/lib/**

Some other files need to be renamed or moved also with the commands :

mv /usr/include/netax25 /usr/local/include

mv /usr/local/include/netax25/ax25.h /usr/include (this is to put back ax25.h in the right place !)

mv /usr/share/doc/liblibax25* /usr/share/doc/libax25*

You will be pleased to see that you can read manuals for axports, nrports and rports respectively via command like : **man axports**

Suppose you need to upgrade AX25 libraries, rpm manual will tell you how to do it.

And finally, in case you want to uninstall the AX25 libraries, you can use the command : **urpme liblibax25_0.**

3.2 Ubuntu Linux users may use AX.25 libraries DEB package

Charlie K4GBB has built an AX25 libraries package for Ubuntu Linux system (Debian) that you may try if you installed that distro.

Please report your observations as this is K4GBB first attempt with deb package.

http://f6bvp.free.fr/logiciels/ax25/libax25-0.0.12_rc2-1_i386.deb

3.3 An AX.25 libraries RPM package has been prepared by Lee VE7FET (ve7fet.at.gmail.com) and can be downloaded from this site <http://code.google.com/p/linuxax25/>

<http://code.google.com/p/linuxax25/downloads/detail?name=libax25-1.0.1-2.i386.rpm>

3.4 Option to install AX25 libraries from sources

If you prefer to install AX25 libraries from source then follow the following instructions.

- Click on the following link to download the **latest** libax25 source library package version : [libax25-0.0.12-rc2.patched_f6bvp.tar.bz2](#)
- Download the package into the directory of your choice.

Make directory :

```
mkdir /usr/local/src/ax25 and change directory
```

```
cd /usr/local/src/ax25
```

Move the archive file into this directory then unpack it :

```
tar xf libax25-0.0.12-rc2.patched_f6bvp.tar.bz2 then change the directory
```

```
cd libax25-0.0.12-rc2.patched_f6bvp and type the following set of commands :
```

```
./configure
```

```
make
```

```
make install
```

```
make installconf (this command, only if no previous configuration files exist. Otherwise they would be overwritten)
```

This will copy the following configuration sample files into `/usr/local/etc/ax25` directory : **axports, rsports and nrports**, include files into `/usr/local/include`, documentation manual pages accessible with **man** program into `/usr/local/man` and it will install AX25 libraries into the `/usr/local/lib/` directory.

ls /usr/local/lib should display the following files and symbolic links :

```
-rw-r--r--  1 root root 101464 déc  4 19:59 libax25.a
-rw-r--r--  1 root root  20816 déc  4 19:59 libax25io.a
-rwxr-xr-x  1 root root    720 déc  4 19:59 libax25io.la*
lrwxrwxrwx  1 root root    18 déc  4 19:59 libax25io.so ->
libax25io.so.0.0.0*
lrwxrwxrwx  1 root root    18 déc  4 19:59 libax25io.so.0 ->
libax25io.so.0.0.0*
-rwxr-xr-x  1 root root  23090 déc  4 19:59 libax25io.so.0.0.0*
-rwxr-xr-x  1 root root    706 déc  4 19:59 libax25.la*
lrwxrwxrwx  1 root root    16 déc  4 19:59 libax25.so ->
libax25.so.0.0.0*
lrwxrwxrwx  1 root root    16 déc  4 19:59 libax25.so.0 ->
libax25.so.0.0.0*
-rwxr-xr-x  1 root root  76967 déc  4 19:59 libax25.so.0.0.0*
```

- **VERY IMPORTANT** : check that there are no other libax25 previously installed on your Linux system in `/usr/lib/` directory. **In that case you should remove all corresponding**

/usr/lib/libax25* files !

- Then you should add a line with **/usr/local/lib** at the end of **/etc/ld.so.conf** file and run the following command **/sbin/ldconfig**
- *Note : the above change in ld.so.conf and the call to ldconfig is performed automatically when running the script in chapter 6.*

4. How to install AX25 tools

I patched [official ax25-tools ax25tools-0.0.10-rc2](#) package in order to update it and remove some bugs. You may choose to use the RPM (or DEB) packages for easy install or the new source files and compile them.

To get the source files archive, right click on the following link and download the package into the directory of your choice

http://f6bvp.free.fr/logiciels/ax25/ax25-tools-0.0.10-rc2.patched_f6bvp.tar.bz2

Move the file to **/usr/local/src/ax25/** then change to that directory.

Unpack the downloaded package

tar xf ax25-tools-0.0.10-rc2.patched_f6bvp.tar.bz2 then change the directory
cd ax25-tools-0.0.10-rc2.patched_f6bvp and type the following set of commands :

```
./configure
make
make install
make installconf    (this command only if no previous
configuration file exist, otherwise they would be overwritten)
```

This will copy sample configuration files **ax25d.conf**, **ax25.profile**, **ax25spawn**, **nrbroadcast.conf**, **rip98d.conf**, **rxecho.conf**, **ttylinkd.conf** into **/usr/local/etc/ax25**, documentation manual into **/usr/local/man**, application tools will be placed into **/usr/local/sbin** and **/usr/local/bin**

4.1 Ubuntu Linux users may want to use AX.25 tools DEB package

Charlie K4GBB has built an AX25 tools package for Ubuntu Linux system (Debian) that you may try if you installed that distro.

Please report your observations as this is K4GBB first attempt with deb package.

http://f6bvp.free.fr/logiciels/ax25/ax25-tools-0.0.10_rc2-1_i386.deb

4.2 Another option is from AX.25 tools RPM package prepared by *Lee VE7FET* (ve7fet.at.gmail.com) that can be downloaded from this site
<http://code.google.com/p/linuxax25/>

<http://code.google.com/p/linuxax25/downloads/detail?name=ax25tools-1.0.1-2.i386.rpm>

5. How to install AX25 applications

Unlike the previous two packages, AX25 applications have evolved recently. Steve VK5ASF has provided an `ax25ipd` version with dynamic IP capability. As this version is not included into [official ax25-apps release](#) at the present time, I have provided the source of patched `ax25-apps-0.0.8-rc2` package. You may choose to use the RPM (or DEB) packages for easy install or get the new source files and compile them. It is always preferable to compile software on your own system.

To get the source files archive, right click on the following link to download the newest version package into the directory of your choice : http://f6bvp.free.fr/logiciels/ax25/ax25-apps-0.0.8-rc2.patched_f6bvp.tar.bz2

Move the file into a directory, for example, `/usr/local/src/ax25` then change to that directory.

Unpack the package :

```
tar xf ax25-apps-0.0.8-rc2.patched_f6bvp.tar.bz2 – Change directory
```

```
cd ax25-apps-0.0.8-rc2.patched_f6bvp and type the following set of commands :
```

```
./configure
make
make install
make installconf (this command only if no previous configuration
file exist, otherwise they would be overwritten)
```

This will copy sample configuration files `ax25ipd.conf`, `ax25mond.conf`, `ax25rtd.conf` into `/usr/local/etc/ax25`, documentation manual into `/usr/local/man`, application tools will be placed into `/usr/local/sbin` and `/usr/local/bin`

5.1 Ubuntu Linux users may use AX.25 applications DEB package

Charlie K4GBB has built an AX25 applications package for Ubuntu Linux system (Debian) that you may try if you installed that distro.

Please report your observations as this is K4GBB first attempt with deb package.

http://f6bvp.free.fr/logiciels/ax25/ax25-apps-0.0.8_rc2-1_i386.deb

5.2 Another option is from AX.25 tools RPM package prepared by Lee VE7FET (ve7fet.at@gmail.com) that can be downloaded from this site

<http://code.google.com/p/linuxax25/>

<http://code.google.com/p/linuxax25/downloads/detail?name=ax25apps-1.0.1-2.i386.rpm>

- We now need to perform some housekeeping.

6. A script for various directories and file setup

To run an FPAC node on a 2.6 Linux kernel we need to setup some directories and file rights. I wrote the following script to facilitate the job.

- You can cut and past **rc.init.script** into a directory of your choice
- modify the file permissions **chmod a+x rc.init.script** and run it with the command **./rc.init.script**
- Or [download the script from here](http://rose.fpac.free.fr/MINI-HOWTO/) into your chosen directory and use the same commands as above.

```
#!/bin/sh
# rc.init.script
# Part of http://rose.fpac.free.fr/MINI-HOWTO/
# by f6bvp at amsat dot org
#
# This script should be run only once after installation of
#
# - libax25
# - ax25-apps
# - ax25-tools
#
# and before installing FPAC fpac327 package
#
# AX25 libraries declaration (into ld.so.conf)
#
echo "/usr/local/lib" >> /etc/ld.so.conf
/sbin/ldconfig
#
# Reserving UDP port 10093 for FPAC service
#
echo "fpad 10093/tcp # FPAC" >> /etc/services
echo "fpad 10093/udp # FPAC" >> /etc/services
#
# Making necessary AX25 directories
#
mkdir /usr/local/var
mkdir /usr/local/var/ax25
cd /var
ln -s /usr/local/var/ax25 ax25
mkdir /usr/local/etc
mkdir /usr/local/etc/ax25
cd /etc
ln -s /usr/local/etc/ax25 ax25
#
# Add some FPAC directories and files
#
mkdir /usr/local/var/ax25/fpac
/bin/chmod a+x /usr/local/var
/bin/chmod a+x /usr/local/var/ax25
/bin/chmod a+x /usr/local/var/ax25/fpac
/bin/touch /usr/local/var/ax25/fpac/loggedin
/bin/chmod a+w /usr/local/var/ax25/fpac/loggedin
#
mkdir /usr/local/var/ax25/mheard
/bin/chmod a+x /usr/local/var/ax25/mheard
/bin/touch /usr/local/var/ax25/mheard/mheard.dat
#
/bin/touch /usr/local/etc/ax25/axports
/bin/touch /usr/local/etc/ax25/nrports
/bin/touch /usr/local/etc/ax25/flexd.conf
/bin/touch /usr/local/etc/ax25/flexd.routes
#
echo "Do not forget to edit 'SYSOP=' and 'CONNECT='"
echo "in file /usr/local/sbin/fpac.sh"
echo "according to your own BBS callsign and sysop name"
#
```

- Next we need to edit **/usr/local/etc/ax25/axports**, a very important file for the declaration of AX25 port names. Replace callsigns by your own callsign and SSID.

Port **axudp** will be attached by kissattach to a pseudo tty. I declared a speed of 115200 baud, but I am not sure it is used.

Paclen = 256 is the upper limit allowed in standard AX.25 mode.

ROSE routing protocol needs three bytes, so applications that use ROSE protocols should declare a Paclen of no more than $256-3 = 253$ Bytes for calls through a rose port (default port is rose0 created by FPAC).

For example in the **xbbb** port.sys file I set paclen at 250 (see paragraph 10.2.1) to get 253 bytes frames when sent via ROSE protocols.

- It is critical that stations talking together use the same Maxframe parameter. AX25 protocol will allow up to 7 frames but, I am convinced that a small Maxframe is better, especially on radio links. For AXIP (axudp) I suggest that we set Maxframes to 4. In the example I set Maxframe = 2.
- In the following example of </usr/local/etc/ax25/axports> port axudp will be used by FPAC for Internet access. Other ports are used to connect dual ports TNCs or single port TNCs using mkiss driver. Callsign SSID must be unique. You cannot use twice the same SSID number. However you can use a dummy call for axudp port (like AXLINK-0).

```
# /usr/local/etc/ax25/axports
#
# The format of this file is:
# name  callsign speed paclen maxframes description
#
aprs    F6BVP-12  38400    256  2  144,800 (APRS)
uhf-1   F6BVP-13  38400    256  2  RX:430,325/TX:439,725 9600 Bauds
axudp   F6BVP-10  115200   256  2  Linux (AX25<->UDP)
#
```

- In the above example, port names aprs, uhf-1, axudp, that are more explicit than usual port 0, 1, 2 etc... Other names could be **vhf, 40m** etc... The same port names should be used in configuration files that depend on axports port names.
- If you plan to use KPC9612 TNCs, you may [download from here](#) a sample initialization script for two dual radio ports KPC9612 (1200 & 9600 baud). The script should be run after **rc.ax25** script presented in chapter 9. This script is for old style pseudo tty. **You can download from her other scripts examples using [Unix98 pseudo tty](#) provided by Charlie K4GBB.**

7. How to configure ax25ipd

- When installing **ax25-apps** we have compiled **ax25ipd**. This application runs as a daemon in the background. The configuration file is **/usr/local/etc/ax25/ax25ipd.conf**. It tells **ax25ipd** the name of the port where to get or put AX25 packet frames, the name of the port toward the Internet world, and the Internet Protocol address (IPv4) plus UDP port numbers of the remote stations. Given this, **ax25ipd** daemon routes frames in both inward and outward direction between UDP port 10093 and master pseudo tty device. Master pseudo tty device **ptmx** and slave pseudo tty (in the form of **/dev/pts/n** with **n** from 0 to 256) are the two extremities of a virtual tunnel. When a character is written at one extremity it can be read at the other and vice versa. Remember that kissattach links pseudo tty device to AX25 port **axudp** in **/usr/local/etc/ax25/axports** (ax0) in our example. Thus AX25 ax0 port is connected to Internet via this complicated chain. **Ax25ipd** does another important work. It performs an encapsulation of AX25 frames into Internet Protocol frames (IP). AX25 frames are embedded into standard IP frames that Internet can handle. On the other direction, IP frames arriving on UDP port 10093 are stripped by **ax25ipd** and finally AX25 frames are presented to ax0 device. When ax25ipd daemon is running, any local AX25 application can speak transparently via Internet with another remote application by writing or reading on ax0 port without knowing the operations performed after that step.

- Our sample *ax25ipd.conf* listing follows. It can be [downloaded from here](#). There is also another [example of ax25ipd conf file for legacy PTY here](#).

```
# This file is in /usr/local/etc/ax25/ax25ipd.conf
# Configuration file for ax25ipd
#
socket udp 10093
#
# Operation mode (digi or tnc)
#
mode tnc
#
# Serial port or tunnel connected to kissattach
# slave pseudo tty /dev/pts/0 is attached to port ax0 (name axip in
/etc/ax25/axports)
device /dev/ptmx
#
# Initialiser la vitesse du port pseudo tty
#
speed 115200
#
# loglevel 0 - pas de sortie
# loglevel 1 - info de config seulement
# loglevel 2 - evenement majeurs et erreurs
# loglevel 3 - evenements majeurs, erreurs et trace des trames AX25
# loglevel 4 - lister tous les evenements
#
loglevel 3
#
# Broadcast address. All listed address will be sent to routes with "b" flag
#
broadcast QST-0 NODES-0 FBB-0
#
# AX.25 routes declaration
# Format is route (callsign-ssid ) (destination host ip address or alias name)
# a 0 ssid will route all ssids (joker).
#
# route <dest_callsign> <des_address> [flags]
#
# Valid flags are :
# b - broadcast validated via this route
# d - default route
#
route f5kbw-0 ns1.f5kbw.org udp 10093 b
#
route vk2tv-0 115.70.139.149 udp 10093 b
route vk2xb-0 122.252.16.174 udp 10093 b
route vk7hdm-0 115.69.164.134 udp 10093 b
# Palm River KP4DJT
route kp4djt-0 w4bgh.no-ip.org udp 10095 b
#
route flmvp-0 flmvp.no-ip.org udp 10093 b
route f3kt-0 f3kt.dyndns.org udp 10093 b
#
route w4akh-0 198.172.211.163 udp 10093 b
#
route f6ggy-0 f6ggy.dyndns.org udp 10093 b
# Route to Citrus
route k4gbb-9 k4gbb.serveftp.com udp 10093 d
route k4gbb-14 k4gbb.serveftp.com udp 10093 b
# Route to LinuxLab
route k4gbb-12 k4gbb.serveftp.com udp 10094
#
route f6bvp-9 192.168.0.64 udp 10094 b
route f6bvp-8 192.168.0.64 udp 10094
route f6bvp-14 192.168.0.64 udp 10094
# F6BVP-7/6 Machine virtuelle 192.168.0.103
route f6bvp-7 192.168.0.103 udp 10097 b
```

```

route f6bvp-6 192.168.0.103 udp 10097
# F4BWT-0 = joker
route f4bwt-0 62.147.157.243 udp 10093 b
route kd4yal-0 kd4yal.servebbs.org udp 10093 b
route f6gov-0 f6gov.no-ip.org udp 10093 b
#
route va2bbs-0 va2bbs.homeip.net udp 10093 b
# route to Costa Rica
route ti2has-0 201.200.85.62 udp 10093 b
#
route f8coj-0 f8coj.no-ip.org udp 10093 b
# route to Aruba
route p43l-0 p43lfpac.no-ip.org udp 10093 b
# Route to Nicaragua
route ynlbbs-0 ynlbbs.no-ip.org udp 10093
#

```

- As can be seen on the above route lines, with the present version, **ax25ipd** can handle permanent symbolic address, digital IP numbers as well as dynamic IPs, thanks to Steve VK5ASF.

8. How to install and configure FPAC 3.27

Jean-Paul F6FBB, the author of Linux FPAC, stopped its development with version 3.25 for Linux kernel 2.2, but released a compiled working version (3.26) on a floppy image that can be downloaded from [F6FBB FTP site](#) or from a [mirror site](#). This [Windows program](#), Rawrite, will help you to make a bootable floppy that will run Fpac. FPAC 3.26 was associated with Linux system glibc6 library compatible with 2.4 kernels. For our purpose we need to compile FPAC sources.

8.1 First install option from FPAC source package

- Compiling FPAC from sources and using recent GCC compiler and kernels gave errors. With the help of Jean-Paul F6FBB, I included changes to enable you to compile FPAC sources. The new FPAC release presented here is compatible with 2.4 and 2.6 kernels. **With 2.6 kernels, FPAC can now make connections through up to three alternate routes.** Connection will be performed transparently using available opened route via one of three possible neighbour nodes for each route toward any destination ROSE address. First opened route is automatically selected.
- Find the latest FPAC327 source package from here <http://code.google.com/p/fpac/downloads/list> (for example **fpac-3.27.15.tar.bz2**)
- Download the archive file. For example : **fpac-327.15.tar.bz2**
- Move the archive file into **/usr/local/src/ax25** then **cd /usr/local/src/ax25**
then **tar xf fpac-327.15.tar.bz2**

then **cd fpac-327.15**
- Then run the commands :
make
make install
make installconf (only if no previous FPAC configuration file exist, as they would be overwritten)
- **make installconf** command will create subdirectories and files necessary for FPAC into **/usr/local/var/ax25** and **/usr/local/etc/ax25**. FPAC application programs files are installed into **/usr/local/sbin**

8.2 Second install option from FPAC package

- Another method to get latest FPAC (development version) is using the following **svn** command :

```
svn checkout https://fpac.googlecode.com/svn/branches/3.27/ fpac-3.27
```

Then enter into *fpac3.27* new directory

```
cd fpac-3.27
```

Run the following script (this is needed only the first time)

```
./autogen.sh
```

Enter

```
./configure
```

Then

```
make
```

and finally

```
make install
```

- In order to keep your fpac source up to date you will only want to enter the following command from inside *fpac3.27* directory

```
svn update
```

If any file has been downloaded, then perform **make** and **make install** and reboot your system. Really cool isn't it ?

8.3 FPAC configuration

- Former unique FPAC configuration file is now divided into a main FPAC configuration file, a FPAC nodes configuration file and a FPAC routes configuration file.
- Main FPAC configuration file is */usr/local/etc/ax25/fpac.conf* and [FPAC HOWTO](#) explains how to configure it.

The following examples *fpac.conf*, *fpac.nodes* and *fpac.routes* configuration file are setup for a FPAC station connected via Internet and radio ports.

[You may download the example from here.](#)

```
# FPAC AX.25 packet node switch configuration file
# Fichier de configuration du node FPAC
# This file is in /usr/local/etc/ax25/fpac.conf
# Ce fichier doit etre dans /usr/local/etc/ax25/fpac.conf
# Node information
# Informations concernant le node
#
L2call      = F6BVP-10
L3call      = F6BVP-11
Trcall      = F6BVP-15
DNIC       = 2080
Address    = 175502
```



```

#Coverage = 175202 175302 175402
InetPort = 10093
InetAddr = 44.151.75.15
Password = xxxxxxxxxxxx
City = Paris
State = 75017
Country = France
locator = JN18DV
UserPort = *
DefPort = axudp
#
# Additional commands
# Commandes supplementaires
#
Command
    APRS = /usr/local/sbin/monitor -rt -p aprs
    HRD = /usr/local/sbin/mheard -d m
    BBS = connect fe6bvp-1 2080175502
#    BBS = /usr/local/sbin/call_tcp f6bvp 1022
    CLUster = /usr/bin/telnet 192.168.0.64 9000
    CLSpider = /usr/bin/telnet 192.168.0.64 7300
    CONVers = /usr/bin/telnet f4bwt.no-ip.org 3600
    STHeure = /bin/cat /usr/local/var/ax25/fpac/fpacstat.dat
    STJour = /bin/cat /usr/local/var/ax25/fpac/fpacstat.day
#    Telnet =
    IPLinks = /bin/netstat --ax25 --netrom
    IPRoutes = /bin/netstat -nr
    ROse = /bin/cat /proc/net/rose
    RS_NEigh = /bin/cat /proc/net/rose_neigh
    RS_NOdes = /bin/cat /proc/net/rose_nodes
    RS_Routes = /bin/cat /proc/net/rose_routes
#    NR_netrom = /bin/cat /proc/net/nr
#    NR_NEigh = /bin/cat /proc/net/nr_neigh
#    NR_NOdes = /bin/cat /proc/net/nr_nodes
    MANuals = /usr/bin/man %1
    TRace = connect f6bvp-15 2080175502
#    KCK = connect f5kck-10
#    KBW = connect f5kbw-8 2080833501
    USA = connect kd4yal-8 3100727489
    VERsions = /usr/local/bin/versions
End
# La commande BBS connectera l'utilisateur au BBS colocalise sur la meme
# machine via la couche rose.
# La commande CLu idem a BBS
# La commande STat affichera les statistiques des connexions
# La commande telnet etant vide elle ne sera pas affichee aux users
# Commandes reservees aux "sysops"
# Sysop commands
#
Sysop
    Disk = /bin/df -k
    SYSop = /usr/local/sbin/fpacshell
    YGET = /usr/local/sbin/yapp -u /usr/local/etc/ax25/%1
    YPUT = /usr/local/sbin/yapp -d /usr/local/etc/ax25/%1
    RM = /bin/rm /usr/local/etc/ax25/%1
    LS = /bin/ls -l /usr/local/etc/ax25/%1
    CP = /bin/cp /usr/local/etc/ax25/%1 /usr/local/etc/ax25/%2
    CAT = /bin/cat /usr/local/etc/ax25/%1
    MV = /bin/mv /usr/local/etc/ax25/%1 /usr/local/etc/ax25/%2
    REBoot = /sbin/reboot
    WEdit = /usr/local/sbin/wpedit %*
end
#
#
# Other applications
#
# TEST
#Application
#F6BVP-13 = /usr/local/bin/testcommande /home "-P 0" -a -d95

```

```

#End
#
# on donne au port d'accès utilisateur VHF l'adresse 175502
# will add a rose device like rosel
# AddPort = VHF
# Address = 175602
# Port = vhf-1
# End
# on donne au port d'accès utilisateur UHF l'adresse 175302
AddPort = UHF
    Address = 175302
    Port = uhf-1
End
# Port UHF 9600 via carte SCC4 et T7F (experimental)
#AddPort = UHF-9600
#    Address = 175402
#    Port = uhf-2
#End
#
# List of predefined users (to specify a non-default port)
#
# La commande USER est ici inutilisée car les trames sont acheminées
# par la couche rose. Elle était utile auparavant lorsqu'on utilisait
# le Kissnetd pour créer un réseau interne pour relier les applications
# entr'elles.
#User = BBS
#    Path = F6KDS-1
#    port =
#End
#
# Liste d'alias. Permet une connexion facile à certains indicatifs
#
# Alias = F6BVP-2
#    Path = F6BVP-2,175520
# End
#
# Alias = F6BVP-3
#    Path = F6BVP-2,175520
# End
#
# All conf files in /usr/local/etc/ax25/
# FPAC nodes définis dans fpac.nodes
# FPAC nodes are defined in fpac.nodes
# ROSE routes are defined in fpac.routes
# ROSE routes définies dans fpac.routes
# AXIP routes are defined in ax25ipd.conf
# AXIP routes définies dans ax25ipd.conf
#
#

```

FPAC nodes configuration file is ***/usr/local/etc/ax25/fpac.nodes***

The following example ***fpac.nodes*** [may be downloaded from here](#)

```

# FPAC ROSE adjacent nodes configuration file
# Fichier de configuration des Nodes adjacent ROSE FPAC
#
# This file is in /usr/local/etc/ax25/fpac.nodes
# Ce fichier doit être dans /usr/local/etc/ax25/fpac.nodes
#
# List of adjacent nodes
#
Node = VK7HDM
    Path    = VK7HDM-5
    DNIC    = 5050
    Address = 626300
    Port    = axudp
    NoWP    = 0
End

```

Node = TI2HAS
Path = TI2HAS-9
DNIC = 7120
Address = 282700
Port = axudp
NoWP = 0

End

Node = ARUBA
Path = P43L-4
DNIC = 3630
Address = 297585
Port = axudp
NoWP = 0

End

Node = VK2XB
Path = VK2XB-2
DNIC = 5050
Address = 699300
Port = axudp
NoWP = 0

End

Node = VK2TV
Path = VK2TV-2
DNIC = 5050
Address = 656200
Port = axudp
NoWP = 0

End

Node = KP4DJT
Path = KP4DJT-9
DNIC = 3100
Address = 813626
Port = axudp
NoWp = 0

End

Node = K4GBB
Path = K4GBB-9
DNIC = 3100
Address = 352726
Port = axudp
NoWp = 0

End

Node = MVP
Path = F1MVP-5
DNIC = 2080
Address = 886801
Port = axudp
NoWp = 0

End

Node = YAL
Path = KD4YAL-9
DNIC = 3100
Address = 727489
Port = axudp
NoWp = 0

End

Node = F5KBW
Path = F5KBW-9
DNIC = 2080
Address = 833501
Port = axudp
NoWp = 0

End

Node = F4BWT
Path = F4BWT-11
DNIC = 2080
Address = 428501
Port = axudp
NoWp = 0

```

End
Node = LAN520
  Path = F6BVP-9
  DNIC = 2080
  Address = 175520
  Port = axudp
  NoWp = 0
End
Node = LAN521
  Path = F6BVP-7
  DNIC = 2080
  Address = 175521
  Port = axudp
  NoWp = 0
End
Node = LAN522
  Path = F6BVP-5
  DNIC = 2080
  Address = 175522
  Port = axudp
  NoWP = 0
End
Node = GGY
  Path = F6GGY-9
  DNIC = 2080
  Address = 847501
  Port = axudp
  NoWP = 0
End
Node = F3KT
  Path = F3KT-11
  DNIC = 2080
  Address = 444501
  Port = axudp
  NoWp = 0
End
Node = COJ
  path = F8COJ-11
  DNIC = 2080
  Address = 854509
  Port = axudp
  NoWp = 0
End
Node = BBS
  Path = YN1BBS-9
  DNIC = 7100
  Address = 505522
  Port = axudp
  NoWp = 0
End
#

```

FPAC routes configuration file is ***`/usr/local/etc/ax25/fpac.routes`***

The following example ***`fpac.routes`*** [may be downloaded from here](#)

```

#
# FPAC ROSE route configuration file
# Fichier de configuration des routes ROSE FPAC
#
# This file is in /usr/local/etc/ax25/fpac.routes
# Ce fichier doit etre dans /usr/local/etc/ax25/fpac.routes
#
# Routes to adjacent nodes
#
# DNIC specify a default DNIC for the following addresses
#
# Address is for 1 to 6 digits, using hierarchy

```

```
#
Routes
DNIC      = 0
2080     = F4BWT
2080     = F3KT
2080     = F5KBW
3020     = K4GBB
3100     = KP4DJT
3100     = YAL
3100     = K4GBB
3620     = TI2HAS
3620     = BBS
3620     = ARUBA
3630     = TI2HAS
3630     = BBS
3630     = ARUBA
5050     = VK2TV
5050     = VK2XB
5050     = VK7HDM
7120     = ARUBA
7120     = BBS
7120     = TI2HAS
7100     = ARUBA
7100     = TI2HAS
7100     = BBS
#
DNIC      = 2080
# 175505 = KVP
  175522 = LAN522
  175521 = LAN521
  175520 = LAN520
#
  854509 = COJ
#   175 = LAN520
#   178 = LAN520
#   195 = LAN520
     1 = LAN521
     1 = LAN520
    444 = MVP
    444 = GGY
    444 = F3KT
  444501 = F3KT
     7 = F5KBW
     8 = F5KBW
     8 = COJ
    8335 = F5KBW
  833501 = F5KBW
     847 = F4BWT
     847 = GGY
  847501 = GGY
     428 = F4BWT
  428501 = F4BWT
  886801 = MVP
     9 = F5KBW
DNIC      = 3100
  352726 = K4GBB
  727489 = YAL
  813626 = KP4DJT
DNIC      = 3630
  297585 = ARUBA
DNIC      = 5050
  699300 = VK2XB
  656200 = VK2TV
  626300 = VK7HDM
DNIC = 7100
  505522 = BBS
DNIC      = 7120
  282700 = TI2HAS
End
```

- The resulting ROSE route table is :

Address Port Description

2080,175302 uhf-1 RX:430,325/TX:439,725 9600 Bauds

ROSE routes :

DNIC	Address	Primary Route	1st Alt Route	2nd Alt Route
7100,.....	YN1BBS-9	Opened	TI2HAS-9 Closed	P43L-4 Opened
7120,.....	TI2HAS-9	Closed	YN1BBS-9 Opened	P43L-4 Opened
5050,.....	VK7HDM-5	Opened	VK2XB-2 Closed	VK2TV-2 Opened
3630,.....	P43L-4	Opened	YN1BBS-9 Opened	TI2HAS-9 Closed
3620,.....	P43L-4	Opened	YN1BBS-9 Opened	TI2HAS-9 Closed
3100,.....	K4GBB-9	Opened	KD4YAL-9 Closed	KP4DJT-9 Closed
3020,.....	K4GBB-9	Opened		
2080,.....	F5KBW-9	Opened	F3KT-11 Opened	F4BWT-11 Closed
2080,9.....	F5KBW-9	Opened		
2080,8.....	F8COJ-11	Closed	F5KBW-9 Opened	
2080,7.....	F5KBW-9	Opened		
2080,1.....	F6BVP-9	Opened	F6BVP-7 Closed	
2080,428...	F4BWT-11	Closed		
2080,847...	F6GGY-9	Closed	F4BWT-11 Closed	
2080,444...	F3KT-11	Opened	F6GGY-9 Closed	F1MVP-5 Closed
2080,8335..	F5KBW-9	Opened		
7120,282700	TI2HAS-9	Closed		
7100,505522	YN1BBS-9	Opened		
5050,626300	VK7HDM-5	Opened		
5050,656200	VK2TV-2	Opened		
5050,699300	VK2XB-2	Closed		
3630,297585	P43L-4	Opened		
3100,813626	KP4DJT-9	Closed		
3100,727489	KD4YAL-9	Closed		
3100,352726	K4GBB-9	Opened		
2080,886801	F1MVP-5	Closed		

2080,428501 F4BWT-11 Closed |
2080,847501 F6GGY-9 Closed |
2080,833501 F5KBW-9 Opened |
2080,444501 F3KT-11 Opened |
2080,854509 F8COJ-11 Closed |
2080,175520 F6BVP-9 Opened |
2080,175521 F6BVP-7 Closed |
2080,175522 F6BVP-5 Closed |

- With this route table, packets will be directed using the first opened route, starting with the route having an address matching the destination address. The table is screened from bottom to the top and from left to right (primary route, 1st alternate, 2nd alternate).
 - When all FPAC nodes run populated routing table, packet routing will be handled transparently from the calling station application program to destination, giving only its callsign if its ROSE address is already in the White Page data base. And all this work is performed by Linux kernel (actually by the ROSE module itself !).
 - You will have to edit the sample file in order to match your own setup. Edit *L2*, *L3* and *Trcall*, *DNIC*, *Address*, *InetAddr*, *Password*, *City*, *locator* items. Do not change *InetPort 10093* the FPAC standard port. Edit *callsign* and *address* in the **Command** paragraph. Edit adjacent nodes at will. Keep Port 0 for Internet access. Each *callsign on path* lines must have a route defined into */usr/local/etc/ax25ipd.conf route* lines.
 - Note that more than one adjacent node may provide the same route. In the case of link failure, this will offer automatically an alternate route to your packets.
 - [You will find here](#) a very complete FPAC sample configuration file from Ray VK2TV, including configuration for an scc interface card.
-

9. AX25 startup scripts

- Nowadays it is not uncommon to have several home computer interconnected on a Local Area Network. The LAN gets it's Internet connection via a router which provides the gateway. For FPAC it is not a good idea to use an address assigned by a DHCP server. We need to use a fixed IP address for our Linux host. For the purpose of this document our LAN address will be **192.168.0.0**. and the gateway address **192.168.0.1**.
- The following script should be placed in */etc/rc.d/rc.ax25* file and the file given the execute rights.
chmod a+x /etc/rc.d/rc.ax25
- We need to initialize AX25 devices and network ports. Before starting FPAC, next script is an example for installing one AXUDP port and two serial ports (connected to a dual ports TNC) following Unix98 pseudo TTY standard. Nodal ampr.org address (f6bvp.ampr.org is 44.151.75.15) should be replaced by your own ampr.org or network addresses.
- The following shell script can be [downloaded from here](#). There is another [rc ax25 script for](#)

[Legacy PTYs here](#). It is strongly recommended to read **kissattach**, **kissparms** and **mkiss** manuals to understand what the scripts do (command **man mkiss**).

- Notes : **f6bvp.ampr.org** or IP address **44.151.75.15** may be omitted in the following script on lines starting **kissattach**. This will prevent unwanted samba broadcast on hamradio links.

```
#!/bin/sh
# this script file is /etc/rc.d/rc.ax25
# after K4GBB ax25 scripts
echo 1 > /proc/sys/net/ipv4/ip_forward
#
echo "Installing ax25ipd Unix98 master pseudo tty"
/bin/date > /var/log/ax25ipd.log
/usr/bin/killall -KILL /usr/local/sbin/kissattach
/usr/bin/killall -KILL /usr/local/sbin/ax25ipd
/usr/bin/killall -KILL /usr/local/sbin/mkiss
/sbin/ifconfig /dev/ax0 down
/sbin/ifconfig /dev/ax1 down
/sbin/ifconfig /dev/ax2 down
/usr/local/sbin/ax25ipd > /tmp/unix98
echo $! > /var/run/ax25ipd.pid
export AXUDP=`tail -1 /tmp/unix98`
#
echo "Installing a KISS link on ethernet port"
/usr/local/sbin/kissattach $AXUDP axudp 44.151.75.15 > /tmp/ax25-config.tmp
awk '/device/ { print $7 }' /tmp/ax25-config.tmp > /tmp/ax25-config-tmp
read Device < /tmp/ax25-config-tmp
# Check for Device
if [ -d /proc/sys/net/ax25/$Device ]
then echo "Port axudp attached to $Device"
    cd /proc/sys/net/ax25/$Device/
# ifconfig $Device parameters
    ifconfig $Device netmask 255.255.255.255
    ifconfig $Device broadcast 0.0.0.0
else echo "*** Error setting $Device parms ***"
fi
/bin/sleep 4
echo "Installing two KISS connections on serial port ttyS0"
# aprs & uhf-1 are ports names in /etc/ax25/axports
echo "Installing two KISS connexions on serial port ttyS0"
# -c enables a one-byte checksum on each incoming and outgoing KISS frame on
# the serial port. This checksum is used by G8BPQ KISS roms to maintain the
# integrity of KISS frames. Do not use -c with standard TNC KISS EPROM.
/usr/local/sbin/mkiss -s 38400 -c -x 2 /dev/ttyS0 > /tmp/unix98
/usr/local/sbin/mkiss -s 38400 -x 2 /dev/ttyS0 > /tmp/unix98
export PTS0=`more /tmp/unix98 | grep -w /dev | cut -b -11`
export PTS1=`more /tmp/unix98 | grep -w /dev | cut -b 12-`
# Attach KPC9612 TNC in kiss mode - on primary PORT ttyS0 = 1200 bauds ---> ax1 =
aprs
/usr/local/sbin/kissattach $PTS0 aprs f6bvp.ampr.org > /tmp/ax25-config.tmp
awk '/device/ { print $7 }' /tmp/ax25-config.tmp > /tmp/ax25-config-tmp
read Device < /tmp/ax25-config-tmp
# Install Parameter : P=250, W=10, TX-Delay=50
/usr/local/sbin/kissparms -p aprs -r 250 -s 10 -l 20 -t 220
# Check for Device
if [ -d /proc/sys/net/ax25/$Device ]
then echo "Port aprs attached to $Device"
    cd /proc/sys/net/ax25/$Device/
# ifconfig $Device parameters
    ifconfig $Device netmask 255.255.255.0
    ifconfig $Device broadcast 0.0.0.0
else echo "*** Error setting $Device parms ***"
fi
# Port end
# Attach KPC9612 TNC in kiss mode - on secondary PORT ttyS0 = 9600 bauds ---> ax2 =
uhf-1
/usr/local/sbin/kissattach $PTS1 uhf-1 f6bvp.ampr.org > /tmp/ax25-config.tmp
```



```

awk '/device/ { print $7 }' /tmp/ax25-config.tmp > /tmp/ax25-config-tmp
read Device < /tmp/ax25-config-tmp
# Install Parameter : P=250, W=10, TX-Delay=50
/usr/local/sbin/kissparms -p uhf-1 -r 250 -s 10 -l 20 -t 220
# Check for Device
if [ -d /proc/sys/net/ax25/$Device ]
then echo "Port uhf-1 attached to $Device"
    cd /proc/sys/net/ax25/$Device/
# ifconfig $Device parameters
ifconfig $Device netmask 255.255.255.0
ifconfig $Device broadcast 0.0.0.0
else echo "*** Error setting $Device parms ***"
fi
# Port end
#

```

- AX25 system can be started automatically at boot time if script */etc/rc.d/rc.ax25* is called by */etc/rc.d/rc.local*. Here is a sample of */etc/rc.d/rc.local* that will initialize AX25 and FPAC node each time the computer is started.

```

#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.
#
echo 60 > /proc/sys/kernel/panic
echo 60 > /proc/sys/kernel/panic_on_oops
echo 1 > /proc/sys/kernel/sysrq
#
/etc/rc.d/rc.ax25
/usr/local/sbin/fpac.sh -start
#
touch /var/lock/subsys/local
#

```

- If you prefer not to start AX25 automatically, edit *rc.local* and put a # at the beginning of the line with

```
/etc/rc.d/rc/ax25
```

- I like to start a task without typing a lot of characters, so I made a shortcut to start AX25.

```
cd /usr/local/bin
```

```
ln -s /etc/rc.d/rc.ax25 ax25
```

- Now type **ax25** on a command line to start the FPAC application.
- You can also make a symbolic link named **fpac** that will actually run **fpacnode** (**ln -s /usr/local/sbin/fpacnode /usr/local/bin/fpac**)

10.1. Router and Firewall configurations

If your Linux computer is directly connected to your DSL box without a router, the following applies to your DSL router settings.

A router not only performs the basic routing functions but it usually includes a firewall to protect your computer and LAN from Internet intruders. By default the traffic passing through the router may be

filtered in both directions. IP packets going from LAN to Internet are routed only if you give the permission. You may already be setup to pass normal traffic. However AX25 applications are using ports that must be opened in both directions. Ports up to 1024 are reserved for Standard applications and must not be used. See `/etc/services` file. FPAC application needs **port 10093** to be opened in both directions for **UDP** and **TCP** protocols. Check your router or DSL modem documentation to find out how to enable these ports and redirect these ports to your computer local IP address. Same for other AX25 applications such as **port 6300** (TCP/UDP) for xfbdd (BBS), **port 9000** UDP for DxNet, **port 7300** for DxSpider.

Linux Mandriva is using a firewall application to protect computer access via network ports. It is called **shorewall**. You may want to disable it (not recommended) or if not, you need to allow traffic for some application protocols and specific ports. This is done via **Mandriva control Center (MCC)** menu **Security**. Select **Configure your own firewall**. Select **all** if you want to let all ports opened or select some applications like : SSH, FTP, Web server etc... Additionally go into **Advanced** and write the following port numbers you will use for FPAC, BBS, Cluster etc.. : 6300/tcp 6300/udp 7300/tcp 7300/udp 10093/tcp 10093/udp

10.2. Interfacing FPAC with application programs

Some application programs that will run on the same Linux machine can listen to a local port and may be easily interconnected with fpac node.

FBB and DxNet can communicate with FPAC without additive programs. **NetRomd** needs **ax25d** to be connected internally with FPAC.

10.2.1. Interfacing FPAC with a BBS application

If your Linux box is running a BBS it can be internally connected to ROSE FPAC and interfaced with NetRom. This will give your BBS a worldwide access to AX25 stations

Important notice : in modern distros Linux kernel is customized for Simultaneous Parallel Multiprocessing (SMP), allowing the kernel to use multiple core CPUs when present. However LinFBB BBS sometimes does not work well when running with more than one CPU. In that case the trick is to tell the kernel to only use one CPU with maxcpus=1 argument in GRUB /boot/grub/menu.lst file.

Note : maxcpus=1 should replace nosmp argument that I recommended previously. nosmp must not be used as it creates bad issues with interrupts leading to instable systems ! (technical reason is that ACPI is also disabled).

Edit `/boot/grub/menu.lst` file and add **maxcpus=1** like in the following example :

```
title 2.6.37
kernel (hd0,0)/boot/vmlinuz-2.6.37 BOOT_IMAGE=2.6.37 root=/dev/hda1 splash=silent vga=788
maxcpus=1 initrd (hd0,0)/boot/initrd-2.6.37.img
```

Recommended FBB version is always the last one as it is supposed to remove bugs present in earlier versions (...)

Follow this link : <http://f6bvp.free.fr/logiciels/BBS-f6fbb/> and download latest source package like : **xd705-src.tar.bz2**

Lets now have a look at different FBB configuration files under Linux.

This is a partial listing from FBB BBS showing the port description file :
/usr/local/etc/ax25/fbb/port.sys

```
#Com Interface Adress (Hex)   Baud
  1   9          *****      115200
  2   9          189C          0
#
#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   1   rose0  250   2      6     10     12/04  XULWY Rose0
  2   4   1   axudp  250   2      2     10     25/20  XUWY  AXUDP
  3   8   2   0      250   2      4     10     04/10  TUWY  Telnet
  4   4   1   uhf-1  250   2      2     10     05/06  XULWY UHF-1
#
```

MultCh represents pseudo TTY devices created by kissattach.

Here is again an example of the [/etc/rc.d/rc.ax25 script](#) I am using to initialize pseudo TTYs.

Devices names are declared in **/usr/local/etc/ax25/axports**.

Like in the above example Multch names can be different from 1, 2, 3,... More clearly identified names like **vhf-1**, **vhf-2**, **uhf**, **axudp**, **aprs**, are encouraged.

Remember, **Multch** device names must be identical to the names declared in axports file, or be a zero for Telnet device.

TNC 4 is linked to AX25 port named uhf-1 in /etc/ax25/axports file.

Telnet com port address 189C is decimal 6300. Connection command is : telnet <hostname> 6300 to enter the BBS.

Pseudo **uhf-1** will interface the BBS directly with AX25 port.

Pseudo **rose0** is the interface toward and from ROSE network handled by FPAC packet switch application.

There is a correspondance between above TNC numbers in port.sys and port letters in forward files : A=1, B=2,...D=4.

In the example above P A means that forward for F6BVP-1 will use TNC 1 and actually rose0 pseudo interface.

Line C C may refer to a forward via either symbolic or IP Internet address number. Example :

```
P C
C C K4GBB-0 3100352726
```

or

```
P A
C C F6CDD 88.171.57.115 23
V .F6BVP$WPASSWORD$W
```

Remember to declare BBS telnet **port 6300** in **/etc/services** as explained earlier. Add the following lines at the end of the file.

```
xfbbd 6300/tcp # FBB BBS daemon
xfbbd 6300/udp # FBB BBS daemon
```

Move the source package into **/usr/local/src/ax25**

cd /usr/local/src/ax25

- Enter the following commands (changing the exact file names) :

```
tar xf xd705c3-src.tar.bz2
cd fbbsrc.705c3
cd src
make
make install
```

- As an option you may want to use X11 version of FBB server and client. If so you first need to install three sets of libraries.

```
urpmi lesstif
urpmi lesstif-devel
urpmi libopenmotif
```

Then enter fbb X11 subdirectory fbbsrc.705c3/src/X11

```
cd X11
make
```

- Copy the following executable files into */usr/local/sbin*

```
cp xfbX      /usr/local/sbin
cp xfbX_cl  /usr/local/bin
cp xfb      /usr/local/sbin
```

- you can run X11 fbb server in background from a Konsole :

```
xfb -a &
```

- or start xfb daemon using rc.fbb script (*/etc/rc.d/rc.fbb start*)

- and run X11 fbb client from a Konsole :

```
xfbX &
```

To start BBS application, enter command **fb** that will execute */usr/local/sbin/fb* script. The first time it will proceed to initialization of the necessary configuration files asking a few questions and create */usr/local/etc/ax25/fb.conf* and */usr/local/etc/ax25/port.sys* if they did not exist yet. Next times it will start **xfb** the BBS daemon application. You may put a copy of fb in */etc/rc.d* with other startup scripts.

To help you configuring FBB there is a [BBS keywords documentation](#) provided by charlie K4GBB.

For local connections, edit **sysop CALLSIGN** and **PASSWORD** in file */etc/ax25/fb/passwr.d.sys*

```
# Passwords
123456789
PASSWORD OF ALL NON DEFINED CALLSIGNS.
CALLSIGN 63 1023 PASSWORD
44.151.75.15 *
#
```

You can start a console FBB client manually with the command :

```
xfbC -c -i CALLSIGN -w PASSWORD
```

However it is more convenient to write a script to start a console FBB client :

```
#!/bin/sh
#
/usr/local/sbin/xfbbC -c -i CALLSIGN -w PASSWORD
#
```

Replace *CALLSIGN* and *PASSWORD* by the corresponding names in */etc/ax25/fbb/passwrds*
Let's call the following script file *bbs* and put it in */usr/local/bin* and make it executable :

```
chmod +x bbs
```

Simple *bbs* command will now allow you the SYSOP to enter your local BBS by running the FBB client *xfbbC*.

- Finally, [you may find a complete documentation ON LINE](#) for the FBB BBS or download these files : [FBB7UT.DOC](#), [fbb7ut_g.zip](#)

10.2.2. Interfacing FPAC with DxCluster or DxSpider applications

DxCluster

Dxnet system/*dxnet.cfg* file will look like :

```
set/node +f8kgk-3 +f5mtz-3 +f5kbw-3 +f6cdd-3 +tu5ex-3 +kp4ig-5
set/call f6bvp-2
set/sys +F6BVP
set/ssid +2
set/port 0 1 3 4 rose0 inet:9000 netclu
set/tz +2
```

The Dx cluster can be accessed via either ax25 radio ports **0, 1, 3** and **4**, via FPAC node port **rose0**, telnet port **9000** and NetRom port **netclu**.

DxSpider

If you want to provide FPAC users access to DxSpider (callsign F6BVP-2) you can use *ax25d* daemon.

Install DxSpider following its documentation. Follow the examples for configuring telnet or radio ports user access. Don't forget to change the file name of **startup.issue** and rename it **startup** after editing it according to your needs, as it took me a long time since I found out why DxSPIDER did not work properly for it did not find **startup** file but did not complain about it !

For access through FPAC here is the content of necessary */usr/local/etc/ax25/ax25d.conf* file

```
# a ROSE port is declared between { }
{F6BVP-2 via rose0}
NOCALL * * * * * L
default * * * * * - sysop /spider/src/client client %u ax25
#
```

Replace **%u** with **%f** if you want to handle users or other nodes with SSID.
After editing this file you need to restart **/usr/local/sbin/ax25d** daemon.
If Spider main directory is not **/spider**, then change it accordingly in the above example configuration file.

Add the following lines at the end of **/etc/services** file

```
spdlogin 7300/tcp # Spider
spdlogin 7300/udp # spider
```

Note : when starting ax25d from a script it must be started *after* fpac in order to have **rose0** device already set up.

When a user issues a connect request to F6BVP-2 via ROSE FPAC network, it will be connected to the DX cluster.

You may add a command to give users access to a Dx Cluster from your FPAC node.
This command is to be added into fpac.conf at paragraph **Command**. For example :

Command

```
APRS    = /usr/local/sbin/monitor -rt -p aprs
MONitor = /usr/local/sbin/monitor -art -p %1
HRD     = /usr/local/bin/mheard -d m
BBs     = connect f6bvp-1 2080175520
CLUster = /usr/bin/telnet 192.168.0.64 9000
CLSpider = /usr/bin/telnet 192.168.0.64 7300
VERsions = /usr/local/bin/versions
```

End

10.2.3. Interfacing FPAC with NetRom

Creating NetRom devices

Run the following sample script during AX25 setup. It will create **NetRom devices** that will be used by the following applications and start **NetRomd** daemon.

Edit the file with your own ampr address (44.xxx.xx.xx) and adjacent NetRom nodes.

```
#!/bin/sh
# /etc/rc.d/rc.netrom
#
#echo "NETROM"
# /sbin/modprobe netrom
#
killall -KILL netromd
ifconfig nr1 down
ifconfig nr0 down
ifconfig nr2 down
# creating NET/ROM devices (please edit your own ampr.org network address)
/usr/local/sbin/nrattach -i 44.151.75.15 -m 236 netnod
/usr/local/sbin/nrattach -i 44.151.75.15 -m 236 netbbs
/usr/local/sbin/nrattach -i 44.151.75.15 -m 236 netclu
# adjacent NET/ROM nodes creation via port axudp = ax0 (ax25 over ip)
/usr/local/sbin/nrparms -nodes k4gbb-14 + CIT03 120 6 axudp k4gbb-14
/usr/local/sbin/nrparms -nodes kp4djt-14 + DJT14 120 6 axudp kp4djt-14
/usr/local/sbin/nrparms -nodes vk2tv-14 + KEMPC 120 6 axudp vk2tv-14
/usr/local/sbin/nrparms -nodes vk2vy-2 + VYNODE 120 6 axudp vk2vy-2
# start ax25d daemon to link NetRom and FPACnode application
```

```

/usr/local/sbin/ax25d
# Broadcast NET/ROM
/usr/local/sbin/netromd -lid -t 10
# Setup Netrom subnetmask
/sbin/ifconfig nr0 netmask 255.255.255.255
/sbin/ifconfig nr1 netmask 255.255.255.255
/sbin/ifconfig nr2 netmask 255.255.255.255
#

```

For deeper understanding see *nrattach* and *nrparms* manuals.

nrports file describes NetRom port names

Example of `/usr/local/etc/ax25/nrports` file

```

# /usr/local/etc/ax25/nrports
#
# The format of this file is:
# name callsign alias paclen description
#
netnod F6BVP-10 BVPN10 236 F6BVP FPAC node
netbbs F6BVP BVPBBS 236 F6BVP BBS
netclu F6BVP-2 BVPDX 236 F6BVP Dx Cluster
#

```

Note : for best FPAC routing do not set any SSID for BBS callsign in above file. This will allow FPAC to choose the right protocol depending on the SSID when a user make a call request. If call is **C F6BVP** or **C BVPBBS** then FPAC will use NetRom protocol. If call is **C F6BVP-1** then FPAC will use ROSE protocol.

Daemon `/usr/local/sbin/ax25d` should run to redirect NetRom protocol calls to FPACnode
This is the content of your `/usr/local/etc/ax25/ax25d.conf` file

```

#
<netrom>
parameters 1 10 * * * * *
NOCALL * * * * * L
default * * * * * * root /usr/sbin/fpacnode fpacnode -q %S
#

```

Add the letter **b** after UDP port number of the route description in `/usr/local/etc/ax25ipd.conf` to allow broadcast of NetRom routes.

10.2.4. Adding RMS link to FPAC

This page by Charley K4gbb describes [how to configure user RMS gate access via FPAC network.](#)

10.2.5. Allowing remote users access to FPAC node via Internet

*This is how to configure Internet acces to your FPAC node using either a TELNET or SSH connection.
Note : this will not give users access to your Linux machine but only give them access to FPAC node.*

If you did not already create a user named **fpacuser**, create it with the following command :

adduser -m fpacuser

Create user **fpacuser** password **fpac** :

passwd fpacuser

enter NEW UNIX password : **fpac**
ignore the remark and retype
NEW UNIX password : **fpac**

Edit file **/etc/group** for adding users to **fpacuser** group

Use your usual editor, the following syntax is for **vi**

vi /etc/group

Use arrow keys to get down to the last line comprising

fpacuser:x:150:

Note : the number may be different.

Type on key **i** to get into insert mode and add **fpacuser** and **root**

fpacuser:x:150:fpacuser,root

Type on key **Esc** then on key semicolon **:** then enter **wq** and press **Enter** key to leave the editor.

We now need to edit the users password file to change the shell program name invoked when a **fpacuser** user is connecting

vi /etc/passwd

Type on key **i** to get into insert mode use arrow keys to get down to the last line comprising user **fpacuser**

and replace existing **/bin/sh** or **/bin/bash** by **/usr/local/sbin/fpacnode**

Type on key **Esc** then on key semicolon **:** then enter **wq** and press **Enter** key to leave the editor.

It is now necessary to give users the rights for using **fpacnode** :

chmod 4775 /usr/local/sbin
chmod 4775 /usr/local/sbin/fpacnode

It is also necessary to change loggedin file permission :

chmod a+w /usr/local/var/ax25/fpac/loggedin

You can now check that your configuration is working by making a local connection via telnet or ssh to your FPAC Linux :

telnet localhost -l fpacuser
or
ssh -l fpacuser localhost

and enter password **fpac** to get access to **fpac** node.

10.3. Other Network Utility Programs

Net-tools includes some useful programs. Among them, **netstat** is interesting for it can display a list of Linux opened sockets, kernel route table and network statistics.

In present distro net-tools netstat handles **NetRom** and **AX.25** sockets, but does not provide any information about **ROSE** sockets used by FPAC.

I wrote two patches for the newest package that can be downloaded from Berlios.de net-tools repository to make netstat know about **ROSE sockets**.

Open a terminal and type the following command lines :

```
cd /usr/local/src/
```

```
cvs -d:pserver:anonymous@cvs.net-tools.berlios.de:/cvsroot/net-tools login
```

Type <return> when asked for a password.

```
cvs -z3 -d:pserver:anonymous@cvs.net-tools.berlios.de:/cvsroot/net-tools co net-tools
```

This will create net-tools sub directory with all net-tools modules including netstat.

Download the following patch with your browser and move it into **/usr/local/src**

<http://f6bvp.free.fr/logiciels/netstat/netstat-CVS-1.70.patch>

```
cd /usr/local/src and apply the patch :
```

```
patch -p0 < netstat-CVS-1.70.patch
```

Enter net-tools directory :

```
cd net-tools
```

Compile net-tools utility programs :

```
make (you need to answer y (yes) or n (no) to a number of questions to customize net-tools compilation
```

default response is Yes)

```
make install
```

You can now use netstat to display ROSE sockets information with the commands :

```
netstat --ax25
```

```
or netstat --netrom
```

```
or netstat --rose
```

```
[root@f6bvp-9 net-tools-1.60-23]# netstat --ax25
```

```
Active AX.25 sockets
```

```
Dest Source Device State Vr/Vs Send-Q Recv-Q
```

```
F6BVP-12 F6BVP-14 ax2 ESTABLISHED 006/003 0 0
```

```
F6BVP-11 F6BVP-9 ax2 ESTABLISHED 003/000 0 0
```

```
* F6BVP-1 ax2 LISTENING 000/000 0 0
```

```
* F6BVP-1 ax1 LISTENING 000/000 0 0
```

```
* F6BVP-1 ax0 LISTENING 000/000 0 0
```

```
* F6BVP-15* ??? LISTENING 000/000 0 0
```

```
* F6BVP-15 ??? LISTENING 000/000 0 0
```

```
* F6BVP-8 ??? LISTENING 000/000 0 0
```

```
* F6BVP-8* ??? LISTENING 000/000 0 0
```

```
* F6BVP-9* ??? LISTENING 000/000 0 0
```

```
[root@f6bvp-9 net-tools-1.60-23]# netstat --netrom
```

```
Active NET/ROM sockets
```

```
User Dest Source Device State Vr/Vs Send-Q Recv-Q
```

```
* * F6BVP-0 nr1 LISTENING 000/000 0 0
```

```
[root@f6bvp-9 net-tools-1.60-23]# netstat --rose
```

```
Active ROSE sockets
```

```
dest_addr dest_call src_addr src_call dev lci neigh state
```

```
2080175521 ROUTE-0 2080175520 F6BVP-9 rose0 0 0 LISTENING
```

```
2080175520 F6BVP-0 2080175520 F6BVP-15 rose0 0 1 LISTENING
```

```
* * 2080175520 F6BVP-1 rose0 0 0 LISTENING
```

```
2080428501 F4BWT-10 2080175520 F6BVP-0 rose0 0 0 LISTENING
```

```
* * 2080175520 ROUTE-0 rose0 0 0 LISTENING
```

```
* * 2080175520 F6BVP-15 rose0 0 0 LISTENING
```

```
2080175520 FPAD-0 2080175520 WP-0 rose0 0 0 LISTENING
```

```
* * 2080175520 WP-0 rose0 0 0 LISTENING
```

```
2080175520 WP-0 2080175520 FPAD-0 rose0 1 1 ESTABLISHED
```

```
* * 2080175520 ??????-? rose0 0 0 LISTENING
```

You may want to display kernel ROSE routing table :

```
[root@f6bvp-11 bernard]# netstat --route --rose
```

```
Table de routage ROSE du noyau
```

```
Destination neigh1 callsign device neigh2 callsign device neigh3 callsign device
```

2080175502 00001 RSLOOP-0 ???
2080175522 00016 F6BVP-5 ax0
2080175521 00015 F6BVP-7 ax0
2080175520 00014 F6BVP-9 ax0
2080854509 00012 F8COJ-11 ax0
2080444501 00013 F3KT-11 ax0
2080833501 00008 F5KBW-9 ax0
2080847501 00011 F6GGY-9 ax0
2080428501 00010 F4BWT-11 ax0
2080886801 00009 F1MVP-5 ax0
3100352726 00007 K4GBB-9 ax0
3100727489 00006 KD4YAL-9 ax0
3100813626 00005 KP4DJT-9 ax0
5050699300 00004 VK2XB-2 ax0
5050656200 00003 VK2TV-2 ax0
7120282700 00002 TI2HAS-9 ax0
2080833500 00008 F5KBW-9 ax0
2080444000 00013 F3KT-11 ax0 00011 F6GGY-9 ax0 00009 F1MVP-5 ax0
2080847000 00011 F6GGY-9 ax0 00010 F4BWT-11 ax0
2080428000 00010 F4BWT-11 ax0
3100772000 00006 KD4YAL-9 ax0 00007 K4GBB-9 ax0 00003 VK2TV-2 ax0
2080100000 00014 F6BVP-9 ax0 00015 F6BVP-7 ax0
2080700000 00008 F5KBW-9 ax0
2080800000 00012 F8COJ-11 ax0 00008 F5KBW-9 ax0
2080900000 00008 F5KBW-9 ax0
2080000000 00008 F5KBW-9 ax0 00013 F3KT-11 ax0 00010 F4BWT-11 ax0
3100000000 00006 KD4YAL-9 ax0 00007 K4GBB-9 ax0 00005 KP4DJT-9 ax0
5050000000 00003 VK2TV-2 ax0 00004 VK2XB-2 ax0 00006 KD4YAL-9 ax0
7120000000 00002 TI2HAS-9 ax0 00007 K4GBB-9 ax0 00006 KD4YAL-9 ax0
7100000000 00002 TI2HAS-9 ax0 00007 K4GBB-9 ax0 00006 KD4YAL-9 ax0

11. More HOWTOs

There are many other documents that relate to Linux networking in a general way. I strongly recommend you also read them. They will assist your efforts and provide a stronger insight into other possible configurations.

They are:

[The Linux Kernel HOWTO](#)

[The AX25-HOWTO](#),

[The HAM-HOWTO](#),

[The NET-3-HOWTO](#),

[The Ethernet-HOWTO](#),

and:

[The Firewall-HOWTO](#)

More general Linux information may be found in other [Linux HOWTO](#) documents.

12. Getting Help

Remember that FPAC is a network switch. You must be able to connected to at least one other FPAC node. The more adjacent nodes you are linked to the better your network will perform.

Stations already running an FPAC node is a good place to get help.

Reading messages from FPAC News list is a must.. To see the collection of prior postings to the list, visit the [fpac Archives](#).

Subscribe to the FPAC list by [going to this page](#).

An FPAC users group has been created recently here <http://groups.google.com/group/fpac-operators>

Global help about Linux and Ham radio can be requested via the Linux Ham list. First check the [Linux-hams archive](#), or the [actual Linux-ham list](#).

Linux-Hams is a mailing list for the discussion of the Linux operating system, and it's use in Amateur Radio. Anyone with an interest in both subjects is encouraged to subscribe. The discussion is mostly concentrated on the use and development of the AX.25 code in the Linux kernel. If you would like to subscribe to the Linux-hams mailing list and receive these messages in your own mailbox, send a message to majordomo@vger.kernel.org, placing the line

subscribe linux-hams

in the body of the message.

Other useful links are :

[FPAC International Home page](#)

[Florida Amateur Digital Communication Association](#)

Readers interested in ROSE protocol may start with those readings :

- [User Guide to ROSE X.25 Packet Networking](#), Donald A. Rotolo, N2IRZ.
- [RATS Open Systems Environment \(ROSE\)](#)

The most recent version of this **FPAC MINI HOWTO** may be found here :
<http://rose.fpac.free.fr/MINI-HOWTO/>

I would appreciate any error report or remarks about this documentation. Please contact Bernard F6BVP at the address mentioned in the title.

Happy networking !