Wireless Linux TCP/IP server seminar

Gert Leunen
ON1BLU

Wireless Linux TCP/IP server seminar
Companion guideline

Gert E. Leunen
on1blu@qsl.net
on1blu@on0baf.baf.be.ampr.org

v1.0, 23 may 2000

This text is distributed as a companion guide to the Wireless Linux TCP/IP server seminar, which was first held during october 1999 in Sint−Truiden, Belgium.

The seminar was organized as an attempt for enabling amateur radio operators to start installing, configuring and maintaining wireless TCP/IP servers based on Red Hat Linux 5.2+ systems.

This text should not be considered a Linux tutorial, a networking course or any topic−covering document. It merely is a step−by−step guide for setting up a Linux TCP/IP server − starting from a Linux installation and ending with a fully−operational server that provides all currently well−known Internet services on an AX.25 network. A server, which is able to exchange information with the current legacy network of BBS systems, DX−clusters and Packet switching nodes.

As a round−up, information is provided as well on how to present benefits for the users of Internet−software.

Table of Contents

Table of Contents

Table of Contents

Introduction

As of 1995, some of the radio−amateurs in Sint−Truiden started looking for new challenges and as they noticed the status−quo of the well−known Packet Radio system − while Internet showed a spectacular breakthrough −, the challenge was easy to find: they wanted to start a wireless Internet. The initiative was initially supported by Gert Leunen (ON1BLU), Joachim Elen (ON1DDS) and Walter Machiels (ON4AWM).

Three motivations have driven us through our efforts:

- the typical desire for radio amateurs to muck around with technological stuff (that itchy feeling),
- providing a platform for fellow radio amateurs to experiment on real−life technology and
- making the use of the Packet Radio network as simple as possible, so that other radio amateurs − who wish to use the Packet Radio network solely as an information medium − can concentrate on their favorite (technological) aspects of the hobby.

Since that date, many developments have been initiated, many alternatives for realizing our goals have passed the scene.

For the servers, we originally planned for an OS/2 Warp server system. As this required development of specific drivers, a temporary solution was preferred, namely a regular OS/2 Warp system running WNOS software, soon to be replaced by JNOS software. Our thoughts started shifting towards an NT−server configuration (still requiring the development of drivers), eventually leading us to, in our opinion, the ultimate solution: LINUX.

While servers provide services to the people, the people need to know how to make use of these services. When we started the TCP/IP Task−Force in Sint−Truiden, the main criticism towards the Packet Radio network was its lack of user friendliness and the complex operations that were required to realize certain tasks (e.g. 7Plus). In essence, Packet Radio started to live its own life, where people invested huge amounts of personal time in developing products that could only serve (part of) the amateur radio community.

Our ultimate plan was to let users enjoy the fabulously user−friendly software that arose within the Internet community. If we would be able to use the same protocols on our Packet Radio network as those that are used on the Internet, a bi−directional exchange of software could emerge. The advantage is also bi−directional: the radio amateur can have a quick start for browsing our network (tons of software available), and he can start developing new applications and technologies that enable him to realize an important role of radio amateurism: public service.

Supporting users has therefor been equally important as setting up servers. While trying users to train operating JNOS−software at first, we soon started developing a new type of TNC. This TNC would bridge the historical gap between industry and amateur radio: it acts as an ordinary telephony modem towards the computer controlling it (allowing commercially available dialers to connect to the network), while talking AX.25 (aka Packet Radio) towards its radio port. The TNC, or better, the MicroController Board MCB−152 is currently being distributed with standard Packet Modems. Developments are now in progress for easy use of high speeds.

The seminar, for which this bundle will be a guideline, is the result of more than 1 year of experimenting with Linux and will present you − hopefully − all the help you'll need in configuring a Linux PC to serve as a full−featured Internet server. **It is, however, NOT a Linux course** (although we will cover some Linux operations since this is simply necessary to do the job).

Although this introduction already should have made clear that our efforts are not limited to simply configuring Internet servers, we prefer referencing another document: the IP proposal (from the same authors), published in 1998 to serve as a guideline for distributing IP addresses, assigning domain zones, setting up SMTP/NNTP<−>BBS gateways and creating standard NNTP newsgroups in Belgium. That document is the theoretical base and practical consensus for what will follow in this seminar. You can find it at our server web sites: http://www.baf.be.ampr.org and http://www.qsl.net/on1blu. Locate the TCP/IP TaskForce page and go to the Status track.

We hope you will find the opportunity to launch a server and become part of the information−sharing community soon. But for now, we hope you will enjoy attending this seminar.

Best 73s.

Gert Leunen (ON1BLU)

Special thanks to ON1AFN and ON4AWM for their contributions, support and/or feedback to this document.

Introducing Linux

This chapter could be the hardest part of this document. It’s not easy to find a good point to start from and keeping a simple line in a system so rich as Linux (or any UNIX−derivative for that matter).

Here’s my attempt to identify a few important parts.

The kernel

The most important part is the kernel (and the name Linux itself actually refers to this part only). For as soon as your computer hardware has found a way to access its hardware peripherals, it will start searching for a kernel to boot. As you might conclude from the name, the kernel is the center where all activities on your computer will be coordinated (the ’cerebellum’ and the subconscious in human terminology). The kernel processes all signals received from the peripherals, coordinates processes running in parallel, provides communication channels between those processes, protects those processes from each other, manages the memory, and so on. It’s the kernel that distinguishes the UNIX−derivative. While Linux is positioned low to mid−range, several computer companies developed kernels for high−end systems (like AIX from IBM, Solaris from Sun, HPUX from HP, etc.) − compare this to the low−end position of Windows NT. So, the Finnish student Linus Torvalds, who developed Linux, only develops the kernel (helped by many people on the Internet).

UNIX−kernels typically

- are multi−tasking, meaning that more than 1 program can run at the same time[1]
- are multi−user, meaning that more than 1 user can operate a single UNIX−workstation at the same time, within his own environment
- are equipped with TCP/IP networking capabilities
- support page−on−demand virtual memory

[1] *Recently, they are often multithreaded as well, allowing a single program to execute several internal tasks in parallel*

The shell

Next, we have a shell, which is necessary to let a user interact with the computer through devices like keyboards, mice, displays (or micro−phones and speaker sets in a not so distant future). Many are available, such as CSH (the C−shell), KSH (the Korn shell), BASH (the Bourne Again shell), and so on. The capabilities of these UNIX−shells clearly distinguish them from the shell we know from MS−DOS (command.com), for example. Their scripting features go far beyond the things we can do in DOS's batch−files and even NT's command scripts. They also allow for extensive job control, provide a very con−sistent PIPE communication channel − to let processes communicate their results to successive processes − and often add nifty features like command histories, auto path−completion and the like. The well−known shells are available on practically all UNIX−flavors (resulting the ability to develop highly portable scripts).

One could consider the X environment (window−like) as an interme−diate shell. An X configuration typically consists of an X server, which drives the graphical hardware, a window manager that provides the look&feel (and you can choose from many: TWM, FVWM, CDE, KDE, GNOME, etc.) and the applications that act as X−clients. Notice that the X−client application could be running on another machine while you're just looking at the X server's output. Linux is frequently combined with the Xfree86 X−environment (which is freely distributed including sources − just like the Linux kernel).

The system commands

UNIX systems always come with an incredible bunch of system com−mands and utilities (network administration, backup utilities, user ad−ministration, text formatting, just name it). For Linux, the eventual collection depends on the distribution you've chosen (Red Hat, Slack−ware, S.u.S.E., Debian, Caldera, etc.). Most are developed as part of the GNU project (another freely distributed project, including source files; actually the Linux kernel adopts the GNU licensing policy as you will learn in a minute).

The daemons

The daemons are the services that are provided by a UNIX worksta−tion. This topic is actually taking us beyond the scope of a Linux in−

troduction, since they rarely are committed to a UNIX−flavor (or even UNIX itself). Linux distributions are often bundled with a wealth of daemons, including for popular network services such as E−mail, USENET, WWW, FTP, IRC, etc.

Documentation and help

In Linux, several sources are available to get help:

- The man−pages provide extensive information on the syntax, usage and often configuration of practically all available commands. Use the man pages when you need to know how to apply a specific com−mand, there is no search index for finding commands. Issue ’`man <command−name>`’ from a shell.
- The info−pages are similar in use and goal to the man−pages. Issue ’`info <command−name>`’ from a shell.
- HOWTO−documents, which can be found in `/usr/doc/HOWTO` cover topics and explain − step by step − how those topics can be made operational. These documents should be your first step in tackling new areas, as they serve as detailed guides. You could take a look − for example − at the AX25−HOWTO, which guides you through the entire preparation and configuration process for enabling AX.25 in Linux.
- Documentation for specific software is stored in `/usr/doc/<package name>`. This is the place where you can find the manuals.
- FAQs (Frequently asked questions) can be found at `/usr/doc/FAQ` in several formats (HTML, Postscript or plain ASCII). Check these pages before calling the world for help!

The GNU general public license

Linux is covered by what is known as the GNU General Public License, or GPL. The GPL was developed for the GNU project by the Free Software Foundation and specifies several provisions for the distribution and modification of free software. Free, in this sense, refers to distribution, not cost. The GPL has always been subject to misinterpretation. We hope that this summary will help you understand the extent and goals of the GPL and its effect on Linux.

Originally, Linus Torvalds released Linux under a license more restrictive than the

GPL, which allowed the software to be freely distributed and modified, but prevented any money from changing hands for its distribution and use. On the other hand, the GPL allows people to sell and profit from free software, but does not allow them to restrict another's right to distribute the software in any way.

First, it should be explained that free software that is covered by the GPL is not in the public domain. Public domain software by definition is not copyrighted and is literally owned by the public. Software covered by the GPL, on the other hand, is copyrighted by the author. The software is protected by standard international copyright laws, and the author is legally defined. The GPL provides for software which may be freely distributed but is not in the public domain.

GPL−licensed software is also not shareware. Generally, shareware is owned and copyrighted by an author who requires users to send in money for its use. Software covered by the GPL may be distributed and used free of charge.

The GPL also lets people take, modify, and distribute their own versions of the software. However, any derived works of GPL software must also be covered by the GPL. In other words, a company may not take Linux, modify it, and sell it under a restrictive license. If the software is derived from Linux, that software must be covered under the GPL also.

The GPL allows free software to be distributed and used free of charge. It also lets a person or organization distribute GPL software for a fee, and even make a profit from its sale and distribution. However, a distributor of GPL software cannot take those rights away from a purchaser. If you purchase GPL software from a third−party source, you may distribute the software for free, and sell it yourself as well.

This may sound like a contradiction. Why sell software when the GPL allows you to get it for free? Let's say that a company decided to bundle a large amount of free software on a CD−ROM and distribute it. That company would need to charge for the overhead of producing and distributing the CD−ROM, and may even decide to profit from the sales of the software. This is allowed by the GPL. Organizations that sell free software must follow certain restrictions set forth in the GPL. They cannot restrict the rights of users who purchase the software. If you buy a CD−ROM that contains GPL software, you can copy and distribute the CD−ROM free of charge, or resell it yourself. Distributors must make obvious to users that the software is covered by the GPL. Distributors must also provide, free of charge, the complete source code to the software distributed. This permits anyone who purchases GPL software to make modifications to that software.

Allowing a company to distribute and sell free software is a good thing. Not everyone has access to the Internet and the ability to download software for free. Many organizations sell Linux on diskette, tape, or CD−ROM via mail order, and profit from the sales. Linux developers may never see any of this profit; that is the understanding reached between the developer and the distributor when software is licensed by the GPL. In other words, Linus Torvalds knew that companies may wish to sell Linux, and that he might not see a penny of the profits. In the free software world, the important issue is not money. The goal of free software is always to develop and distribute fantastic software and allow anyone to obtain and use it.

Interpretation quoted from the online ’Linux Installation and Getting Started’ guide by Matt Welsh, Phil Hughes, David Bandel, Boris Beletsky, Sean Dreilinger, Robert Kiesling, Evan Liebovitch and Henry Pierce; Copyright ©1992−1996 Matt Welsh, Copyright ©1998 Specialized Systems Consultants, Inc (SSC).

Installing Linux

If you attended the seminar, you should have received a CD−ROM containing Red Hat Linux 5.2. We presume the usage of this Linux version for the remainder of this text.

If you’re system is able to boot from CD−ROM, you may simply launch the system with the CD inserted in your CD−ROM drive. Don’t forget to make sure − from your BIOS settings − that your CD−ROM drive is in the boot sequence. If you already installed a bootable OS on the system before, you must give the CD−ROM drive higher boot priority than your hard drive.

If you can’t boot from CD−ROM, you will need to make a boot diskette. Just make sure you can boot your system in DOS, WINDOWS ’9x or Windows NT and that the OS can read the CD−ROM. On the CD, DOS−utilities can be found to create the boot diskettes. You can, of course, always call in someone for help (to provide you with such a prepared diskette, ...). If you installed a bootable OS before, the same applies here with regard to the BIOS boot sequence: the floppy drive should be tried before the hard drive.

Now that we’re talking about the BIOS settings, check whether the clock is set sufficiently accurate in **UTC**.

Now that you’ve something to boot, insert the appropriate media in their respective drives and turn the power on. This will bring you to the setup software.

Using the Keyboard to Navigate

You can navigate around the installation dialogs using a simple set of keystrokes. You will need to move the cursor around by using various keys such as [<−], [−>], [\uparrow], and [\downarrow]. You can also use [Tab], and [Alt]−[Tab] to cycle forward or backward through each widget on the screen. In most cases, there is a summary of available function keys presented at the bottom of each screen.

To ‘‘press’’ a button, position the cursor over the button (using [Tab], for instance) and press [Space] (or [Enter]). To select an item from a list of items, move the cursor to the item you wish to select and press [Enter]. To select an item with a check box, move the cursor to the check box and press [Space] to select an item. To deselect, press [Space] a second time.

Pressing [F12] accepts the current values and proceeds to the next dialog; it is usually equivalent to pressing the OK button.

Please Note: Unless a dialog box is waiting for your input, do not press any keys during the installation process − it may result in unpredictable behavior.

Partitioning and mount points

Now let’s have a talk on partitioning your hard drive. Linux has a complete different way with regard to hard drive management, as compared to what you’re used to with DOS/Windows environments. The first and probably most visible difference is that Linux (or any other UNIX) has no concept of ’drive letters’ whatsoever (the next major difference is that directories are not separated by a backslash − ’\’ −, but by a forward slash − ’/’).

Instead of working with drive letters, Linux has a concept of ’mount points’. What’s it all about? To Linux, there is only one file−hierarchy. It starts from the root directory (’/’) and from there you have subdirectories (kind of what you know from DOS). From the root itself, there are typically only a limited set of subdirectories (to keep things ’clean’). These subdirectories often have special meanings:

- `/boot` is where the OS loader and the kernel reside, accompanied with other files that are necessary for starting the first stages of Linux (typically needs only 1MB, allocation should be maximum 16MB to be on the safe side; the partition containing `/boot` should fit ENTIRELY within the first 1024 cylinders on one of your primary hard drives).
- `/usr` is where almost all software will reside (at least all software, and possibly sources, that are part of the base Red Hat Linux distribution). Its size will be relatively fixed when everything is installed, but it should − of course − be large enough to hold it all. Consider at least 300MB.
- `/tmp` is used − three guesses − for storing temporary data. Space required depends entirely on the the application software you will be using.
- `/var` holds the working data for the software, log files and − very important for our TCP/IP−servers − spool files. This means all mail

messages, news bulletins, etc. will end up somewhere under this path (e.g. `/var/spool/mail` for E−mail messages). For our particular use, a separate partition should be reserved for `/var`, and it will probably be one of the largest partitions.

- `/home` is where the home directories for the users will reside. Notice that some servers imply a special user account, and therefor they will own a home directory. Important servers, for whom this applies, are FTP, HTTP and SMB. This means, for example, that the web pages for your WWW−server will all be located in `/home/httpd/html`. You might want to enable your users for publishing their own web pages too; in that case these web pages will reside under the home−hierarchy too (depending on the setup in the home directory of the HTTP−server or in each user's own home directory). So this path also needs consideration with regard to disk space allocation.
- `/opt` is often used to store software packages. Typical example is StarOffice, which will be installed here by default.
- There are a number of other directories, like `/etc` (configuration files), `/dev` (file representation of devices like serial ports, etc), `/mnt` (foreign files systems will be mounted here), `/bin` (system commands), `/sbin` (system administration commands) and just a few more that have special meanings to the system. Wait until you're an expert before making changes there (except for `/etc`)! They are not taken into consideration for separate partitions.

What does all of this have to do with partitions? Well, since Linux does not know drive letters, it needs another way to provide access to different partitions. This is done through what we call 'mount points'. Each partition is assigned a 'mount point', that is the location from where the contents of that partition will be visible. For example, say that you want to allocate 1GB to store E−mails, logs, etc. You create a partition of size 1GB and you have to assign a mount point. Recall that `/var` is the location where such data will be stored, therefor we will assign '`/var`' as the mount point for that partition. From the opposite point of view, this means that the contents of this 1GB partition will be made visible starting from the location '`/var`'. **At least one partition should be assigned the root mount point '/'.**

Finally, you will probably want to assign a partition as swap space. Unlike Windows environments, Linux generally does not use files for swapping memory. A separate partition should be reserved for this. One swap partition can be maximally 128MB in size, optimally from

32MB to 64MB. A partition is marked as swap space by giving it the ’Linux swap’ system id (82) instead of the usual ’Linux native’ (83).

Starting the installation

So, now that we’re so far, let’s get back to work. **FIRST OF ALL, MAKE SURE THAT VIRUS PROTECTION OR WARNING OPTIONS IN THE BIOS ARE DISABLED** (otherwise the installation process will not be able to install the LILO bootloader in the master boot record, resulting in a non−bootable system after installation).

You launched your system with the proper media loaded, at the ’`boot:`’−prompt you pressed the ’enter’−button, at the ’Welcome to Red Hat Linux’−dialog you pressed the ’enter’−button as well, you selected the language during installation with your cursor keys and pressed once more the ’enter’−button, you selected your keyboard layout − probably be−latin1 or us − again using the cursor keys and the ’enter’−button.

Next, you choose ’Local CDROM’ as installation method and you hit the ’enter’−button when the installation software asks you to insert the CD−ROM. You confirm that you want to perform a new **install** and this leads us to the ’installation class’ dialog. For now, we will select ’**Custom**’.

The reason for this last decision is that the server class does not fit our needs. Furthermore, to have maximal functionality available as soon as possible, we prefer to do a full installation for now, which is only possible from the custom class. Disadvantage is that we end up answering lots of questions...

Next, indicate the presence of SCSI−adapters (you will probably want to say ’no’).

Now, we need to start partitioning. Use either DiskDruid or fdisk to do the partitioning. In theory, DiskDruid should be more user friendly, but I’ve always found fdisk to be far quicker (in DiskDruid I always end up struggling with the browse buttons). Add partitions as follows:

1. Create a partition to be mounted as ’`/boot`’. Make it min. 4MB, max. 16MB in size.
2. Optionally, but probably, you may want to create a partition to be mounted as ’`/var`’. This one should be at least 512MB.

3. Optionally, but likely, you may want to create a partition to be mounted as ’`/home`’. It should be at least 128MB (depends on how many users you will allow, whether you want to allow them to publish web pages, whether your server will publish web pages and what you want to publish from your FTP−site).
4. Optionally, but less likely, you may want to create a partition to be mounted as ’`/usr`’. This partition should be at least 1GB. If you consider this option, you might also want to consider a mount point for ’`/opt`’.
5. Create a partition that will be mounted as root ’`/`’. Its size depends on your further decisions. It should actually take the space that’s left over (minus the space you still need to reserve for swap space), but if you preferred to apply all options above, it still should be minimally 256MB.
6. Create a swap partition of size 64MB.

Upon completion, you will be asked to format the partitions with optional media checking.

As for the components, scroll down to the end of the list and enable the option ’**Everything**’ (if you want a full description of all available packages, consult The Official Red Hat Linux Installation Guide, appendix C: Package list). After pressing the ’enter’−button once more (notification that an installation log can be found), formatting of non−swap partitions begins and then installation of packages is started. Time for a coffee...

Finalizing the installation

Hopefully, the installation software has detected the right mouse type (if not, select it from the list; enable the ’Emulate 3 Buttons’ option if you have a 2−button mouse).

Next, the installation software hopefully detected the right graphics card. If not, try finding it in the presented list. If not found, cry for help. For the monitor, try finding your model (or an equivalent) from the list, otherwise one of the generics will probably do (at least until we can muck around with the X configuration software later on). When the installation software asks whether it may probe or not, you generally should prefer ’Don’t probe’. Next, indicate the amount of video RAM installed on your graphics card and select the clock chip (generally, there’s no separate clock chip). Finally, you can enable the resolution you’d like. Since this is only a preliminary configuration, try avoiding high resolutions (more than 640x480) and high color depths (more than 8 bit), so just select 8−bit 640x480...

This brings us to the network part. Select ’yes’ when the installation software asks if a LAN should be configured (even if you don’t have a network card): this will simplify network configuration later on (AX.25 is considered LAN here, but the installation software doesn’t know about AX.25). From the presented list, select the network card that’s installed in your computer; if you don’t have one, select ’PLIP (parallel port)’. If you selected a network card, auto−probing should work in the next dialog.

Servers are generally assigned static IP addresses. We typically assign IP addresses from the reserved test addresses to the regular LAN interfaces as these interfaces probably won’t be connected directly to the Internet. For good maintenance, we separate the networks based on their data−link protocol: we’ll use 192.168.6.x for Ethernet−networks, 192.168.28.x for SLIP−links, 192.168.34.x for PLIP−links, etc[2]. Within each range, it’s generally good practice to reserve the first available address (x=1) for servers. So, if you have a network card installed, enter IP address 192.168.6.1. If you configured PLIP, enter IP address 192.168.34.1. Netmask should be 255.255.255.0. There should be **NO default gateway**[3]. Enter 127.0.0.1 (loopback address) as primary name server (since this system will figure as primary domain server). Domain name is the name of the zone you received (in Sint−Truiden, it is baf.be.ampr.org) and host name is the name of the server (on0baf in Sint−Truiden); the domain name is appended automatically. The other name servers are optional and left open, like the primary one.

Now it’s time to configure the clock: enable the ’Hardware clock set to GMT’ option and select your zone from the list (probably ’Europe/Brussels’).

[2] *You might look a little surprised − as Walter (ON4AWM) did when evaluating this document − and ask yourself what the hell you needed to ask a 44.x.x.x address for. The 44−address may only be used on AX.25 interfaces; they will be configured later on since current commercial Linux distributions don’t have AX.25 enabled as a default. This means we still have work to do (as you will notice) before we can even start configuring this. The addresses above are chosen to avoid any kind of interference with regard to routing later on.*

[3] *Should you see errors upon booting, check the file /etc/sysconfig/network: either a valid default gateway and/or interface should be provided, or they should be empty.*

Then we have the list of daemons that should be started when rebooting (similar to ’services’ in Windows NT). The presented defaults are quite good. You might disable pcmcia (only useful on notebooks), disable sound (I can’t imagine one puts a sound card in a server), disable lpd (the printer daemon that handles print jobs: read next paragraph), enable snmpd (network management and follow−up, but you could leave this for later), and you are encouraged to enable xntpd (the service that allows for synchronising the clocks of all computers on a network). You could also start experimenting with squid, which is a proxy server (a server that stores webpages, for example, that have been visited: when some user requests this page again later, it can be served locally).

Servers don’t have printers attached (at least our servers: they typically reside in some basement or under a roof). If they do (even when the printer is remote, e.g. at your home), just follow the instructions (presented defaults are fine most of the time, also notice you should have left the lpd daemon enabled for autoboot in the previous dialog). You can still do this later on, if you want.

Now you can enter a root password. The installation automatically creates a user ’root’[4]; this user can do ANYTHING. So it’s not advisable to share its password... Upon remote administration, it is often sent in clear text, though.

Depending on the risk you’re willing to take, let the installation software create a bootdisk or not.

And finally (ugh), we may configure LILO, the Linux loader. Since we presume we’re configuring a Linux−only system, select the ’Master Boot Record’ as target for bootloader−installation. Normally no kernel options are required (neither is the linear mode).

And there it is... the congratulations screen (indeed!). Don’t forget to remove the CD before restarting (or give the hard drive back highest priority in the boot sequence upon rebooting).

Does the machine boot (does it show a `’login:’`−prompt)? OK, then real congratulations are in order...

[4] *The home directory for ’root’ is not located under the ’/home’ hierarchy, but straight under the root itself (’/root’). Don’t confuse the root of the file system with the user account ’root’!*

First steps

From the login prompt, enter the login name of the user you’re representing. At this time, the system only knows about one user account, so enter `root`. At the password prompt, enter the password you configured earlier. Normally, a shell prompt should appear now (for our server in Sint−Truiden, this would be ’`[root@on0baf /root] #`’: root is the user name that’s currently logged in, on0baf is the name of the server, /root is the current working directory and the ’#’−sign indicates the system is waiting for user input).

You can log in more than once from the console, as Linux configured 6 virtual consoles for you automatically. You can switch to another virtual console by holding down the [Alt]−key while pressing function keys [F1]−[F6].

You could also try starting Xwindows by entering the command `startx`. If this should fail and you do not return to the shell prompt automatically, issue [Ctrl]+[Alt]+[Backspace]. Adjusting configuration using the `XF86Setup` command might help, otherwise you might need to call for help.

If you happen to succeed in launching Xwindows, try starting NetScape (go to a blank desktop−area, press the left mouse button, move the mouse pointer over the ’Programs’ item in the pop−up menu, follow with your mouse pointer into the new pop−up menu and move it over the ’Networking’ item, follow the path once more and use left mouse button to click on the ’Netscape Navigator’ item). This will bring you to an index page for an important part of online documentation (if Netscape didn’t start with this index page, you can browse to it by entering the URL ’`file:///usr/doc/HTML/index.html`’).

Especially check out the bottom of the page, where you can find the official Red Hat Linux Installation Guide (which you actually would have needed for the installation process, but it’s always interesting to read − like chapter 11 for example), the Linux Installation and Getting Started Guide (particularly chapter 3 is interesting: it contains a Linux tutorial), and of course the Network administration and system administration guides (be careful on reading these, as they’re not quite up−to−date).

You can exit xwindows by moving again to a blank desktop area, pressing the left mouse button, moving the mouse pointer over the item ’Exit fvwm’, then over the item ’Yes, Really Quit’ and pressing the left mouse button there again.

A few more commands needed for now are `exit` or `logout` to log off and leave the system unattended, `halt` for shutting down and `reboot`.

Kickstart files (optional)

If you find your system to be working, you MIGHT consider making a kickstart file. A kickstart file is a text file that directs the installation software to automate all steps mentioned above. Once you have an operational system, determine the appropriate settings from that running system (take a look at appendix H in the Red Hat Linux Installation Guide). Here’s an example:

```
lang en
network −−bootproto static −−ip 192.168.6.1 −−netmask 255.255.255.0
cdrom
device ethernet 3c509 −−opts "io=0x210, irq=10"
keyboard us
zerombr yes
clearpart −−all
part /boot −−size 16
part swap −−size 64
part / −−size 1024
part /var −−size 100 −−grow
install
mouse −−kickstart logimman
timezone −−utc Europe/Brussels
xconfig −−server "Mach64"
rootpw YourRootPassword
%packages
@ Base
#@ Printer Support
#@ X Window System
@ Mail/WWW/News Tools
#@ DOS/Windows Connectivity
@ File Managers
#@ Graphics Manipulation
#@ X Games
#@ Console Games
#@ X multimedia support
@ Console Multimedia
@ Networked Workstation
@ Dialup Workstation
@ News Server
@ NFS Server
@ SMB (Samba) Connectivity
#@ IPX/Netware(tm) Connectivity
@ Anonymous FTP Server
@ Web Server
@ DNS Name Server
@ Postgres (SQL) Server
@ Network Management Workstation
@ TeX Document Formatting
@ Emacs
#@ Emacs with X windows
@ C Development
@ Development Libraries
@ C++ Development
#@ X Development
@ Extra Documentation
kernel−source
joe
unzip
ircii
```

Put this file − named as ks.cfg − on a disk (the Linux install boot−disk if you boot from floppy when installing). Now boot the computer from floppy or CD and at the `boot:` prompt enter `linux ks=floppy`.

Fixing problem when starting PLIP

If you configured a PLIP−interface, you might have noticed some errors while booting. Follow these instructions to solve the problem:

1. From the shell prompt, issue
   ```
   cd /etc/sysconfig/network−scripts
   ```
2. issue `joe ifup−plip`
3. At row 16, replace the line
   ```
   route add −net ${NETWORK} netmask {$NETMASK} ${DEVICE}
   ```
   with (or place a ’#’ comment sign before and add a new line):
   ```
   route add −host ${REMIP} ${DEVICE}
   ```
4. Hold [Ctrl]−key while pressing [k] and [x] sequentially.

You should also check whether a line containing the keyword ’`REMIP`’ exists in file `/etc/sysconfig/network−scripts/ifcfg−plip*` (replaces ’*’ with the actual interface sequence number). If not present, add it to the end of the file with as value the IP address of the computer you’ll connect to.

Your following reboot should work without errors[5].

Fixing problems with regard to network routes

For network routing, you must make sure that the entries ’GATEWAY’ and ’GATEWAYDEV’ are left empty in file `/etc/sysconfig/network`. These entries should only be made when an actual Internet connection is available.

Next, we’ll need to fix a problem on setting up static IP routes (which are read from file `/etc/sysconfig/static−routes`. By default, it’s configured to read route tables from Linuxconf’s configuration files. We have found this method not to be usable, so we’ll disable it. Open the file `/etc/sysconfig/network−scripts/ifup−routes` for editing and, at the end of the file, make sure the lines are

[5] *It might be a good idea to make sure the keywords GATEWAY and GATEWAYDEV have empty values (e.g. line with only ’GATEWAY=’) in the file /etc/sysconfig/network.*

commented as shown below:

```
#if [ −x /bin/linuxconf ] ; then
#  linuxconf −−hint routing "$1" | while read args; do
#      /sbin/route $args
#  done
#else
   if [ ! −f /etc/sysconfig/static−routes ]; then
       exit 0
   fi

   #note the trailing space in the grep gets rid of aliases
   grep "^$1 " /etc/sysconfig/static−routes | while read device args; do
       /sbin/route add −$args $device
   done
#fi
```

Fixing problem when mounting floppy disks

If your system has a floppy drive, your will typically want to read msdos−formatted floppy disks. By default, a floppy mount for Linux's second external file system (denoted by `ext2`) is pre−defined. In gen−eral, there's no particular reason to use that file system on floppies, due to their limited size. Therefor, it is advisable to alter the file `/etc/fstab` (using joe, for example) and change the file system for the line(s) starting with `/dev/fd`? (typically one line with `/dev/fd0`) from `ext2` to `msdos` or `vfat`. From now on, you can simply access floppy disks with the command `mount /mnt/floppy`.

Customizing the kernel

Now it’s time to let Linux meet up with our wonderful world of Packet Radio. Although the idea itself sounds horrifying, we actually are going to compile our own customized kernel. For our server, it well get even worse: we’re going to change the source code of the kernel! The kernel is the most basic part of an operating system: it’s the kernel that coordinates access to each hardware device, that coordinates tasks running in parallel, etc. And we’re going to muck around in it: spooky, isn’t it?

We have no choice, however, AX.25 protocol handling is part of the kernel but the commercial distributions don’t have it activated.

Adjusting the kernel for AX.25 server

First, we will need to adjust a little code. Issue
`cd /usr/src/linux/include/net`.

You will find there a header file named `ip_forward.h`. Open it for editing (using `joe ip_forward.h`). Under the last ’`#define`’, but before the ’`#endif`’, add the following line:

```
#define CONFIG_IP_NO_ICMP_REDIRECT
```

What’s it all about? This option is necessary due to our specific network topology, which can be seen as a star (see Illustration 1) where not all client stations can communicate directly with other client stations on the same network (=same frequency).

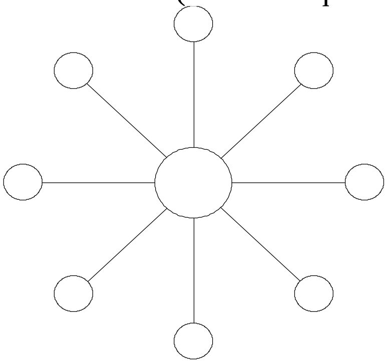


Illustration 1

The end result is that clients need to be configured to route ALL their traffic through the central server (even if they want to communicate to another station on the same frequency).

The problem is that the server will detect that frames will be sent through the same interface (=frequency) as through which it originally arrived. The standard network configuration in Linux will result in a network−maintenance message (ICMP redirect) being sent to the originating client, stating it should adapt its routing table to send all frames to that particular client DIRECTLY to that client.

If the client follows the advice, this could result in the 2 clients no longer being able to communicate with each other. Another problem is that the redirect−message is sent upon EACH individual frame sent through the server. That’s why we turn it off.

Actually, the above definition belongs in the ’`.config`’ file (the kernel configuration). However, this file will be overwritten each time a new kernel is configured. Right, so it eventually belongs in the configuration part of the make−process. I’m in the software engineering business, so you could have seen the right adjustment if I had a few more years to complete this text ☺.

Another solution would be to improve our network topology. One alternative we’re working on, for example, is to use a separate up− and downlink (or input vs. output) frequencies, whereas the downlink frequency is actually the amplified signal of the uplink frequency as received by the server. This way, each client would be able to communicate "directly" to all other clients (on the same frequency−pair). Another advantage of this approach is a reasonable reduction of collisions, caused by clients.

Configuring kernel compilation

So, let’s enter the area by issuing `cd /usr/src/linux`. Kernel configuration can be done in 3 ways: `make config` (you probably don’t want to use this), `make menuconfig` (looking nice, similar to the installation software: keyboard operations are explained at the top of the screen) or even `make xconfig` (in Xwindows).

Topics I don’t mention in the text below may be left untouched. So, let’s start browsing. In general, there appears to be a restriction on the maximal size of the kernel for being bootable. The size is not affected by options configured as ’modules’.

- General setup. You may probably **disable** ’Kernel math emulation’, you might **enable (modular)** ’Kernel support for JAVA binaries’ and you could select the appropriate ’Processor type’.

- Floppy, IDE, and other block devices. You probably want to **disable** ’Include IDE/ATAPI TAPE support’, ’Include IDE/ATAPI FLOPPY support’ and ’Support removable IDE interfaces (PCMCIA)’.
- Networking options. You might want to consider **disabling** ’The IPX protocol’ and ’Appletalk DDP’. You must **enable** (it’s your choice if you want to do it **modular**) ’Amateur Radio AX.25 level 2’ (after all, that’s what we’re here for). The latter operation will add 2 new options ’Amateur Radio NET/ROM’ and ’Amateur Radio X.25 PLP (Rose)’, you probably want these 2 left **disabled**.
- SCSI support. If your system doesn’t have a SCSI adapter, just turn it all off (item ’SCSI support).
- Network device support. **Enable** ’Radio network interfaces’, then **enable** (it’s your choice if you want to do it **modular**, but if you selected modular AX.25 support, you may only set it either disabled or modular) the TNC−type you’ll connect to the server (e.g. ’Serial port KISS driver for AX.25’). Make sure that your network card is **enabled** (if you have one), and **disable** all other options that appear AFTER the radio network interfaces, **except** maybe the ’Traffic Shaper’ at the bottom.
- ISDN subsystem. You’ll probably want all this disabled (turn it all off by disabling ’ISDN support’.
- CD−ROM drivers (not for SCSI or IDE/ATAPI CDROM drives). Except when you have such a proprietary CD−ROM device (in which case you should only leave your particular CD−ROM drive enabled), disable the whole lot by disabling ’Support non−SCSI/IDE/ATAPI CDROM drives’.
- Filesystems.
  Keep the following **enabled**: ’Quota support’, ’Second extended fs support’, ’Native language support (Needed for FAT and ISO9660)’, ’ISO9660 cdrom filesystem support’, ’DOS FAT fs support’, ’MSDOS fs support’ and ’/proc filesystem support’.
  Keep the following **modular**: ’VFAT (Windows−95) fs support’, ’Codepage 437’, ’Codepage 850’, ’NLS ISO 8859−1’, ’NFS file−system support’ and ’SMB filesystem support (to mount WFW shares etc..)’ (enable the ’SMB Win95 bug work−around’).

Disable all others (you’re on your own if you want to experiment with the others).

- Character devices. Only keep enabled what’s actually installed in your system. Make sure that ’Parallel printer support’ is either modular or disabled (or you can no longer use PLIP−networking). You might want to enable ’Advanced Power Management BIOS support’. Use the built−in help function to decide what behavior you want enabled/disabled. Also notice the existence of Watchdog timers.
- Sound. In the rare event your server is equipped with an audio card, make sure that audio card and its appropriate subsystems are the only ones enabled or modular. If you don’t have one, disabled the whole lot by disabling ’Sound card support’.

We’re through. You might want to save the current configuration: select ’Save Configuration to an Alternate File’ and enter the path name (we usually store it in ’/root/KernelON0BAF’).

Now press enter when your cursor is on the ’<exit>’ item at the bottom of the dialog and answer ’yes’ when asked for saving the new kernel configuration.

Compiling the kernel

Now that you’ve returned to a shell prompt, issue

```
make dep ; make clean
```

As soon as you return to the shell prompt after issuing the line above, enter the following command (this will build a compressed kernel image):

```
make zImage
```

As you might expect, this will take some time: depending on the system it will take less than half an hour (Pentium II with plenty of RAM) to more than 4 hours (386 with little RAM). So go fetch another cup of coffee OR you might already start enjoying the benefits of Linux and muck around in another virtual console (trying the stuff from the Linux tutorial, for example). Ever tried to do something on a Windows platform while compiling software? I did, but I will never try again...

Upon completion, do the following:

```
cp arch/i386/boot/zImage /boot/vmlinuz
```

Now, perform the following steps (once you have done a kernel com−pilation for the first time, you can always skip to step 6):

1. `cd /etc`
2. `joe lilo.conf`
3. Before the line containing the keyword ’image’, insert the follow−ing new lines[6]:

   ```
   image=/boot/vmlinuz
          label=linux
          root=/dev/hd??
          read−only
   ```

   the two question marks ’??’ should be replaced by the contents of the original block you’re inserting before (giving, for example, `root=/dev/hda1`)
4. Change the label of the original block by ’orig’ (giving `label=orig`).
5. Save and exit (hold [Ctrl]−key while pressing the sequence [k] [x]).
6. Issue the command `lilo` from the shell prompt. Lilo will now know how to boot the new kernel, but things haven’t finished yet...

Make sure you’re in ’/usr/src/linux’ and issue (will also take some time; modules are comparable to the DLLs in Windows environments):

```
make modules
```

And upon completion:

> *starting from your second kernel compilation, you might want to clean the modules directory first by issuing*
> *`rm −r /lib/modules/2.0.36`*

```
make modules_install
```

NOW we’re through: you may reboot (issue `reboot`). Do you realize yet what you have been doing here? Congratulations (if it works...

[6] *You could start by copying the original block by moving the cursor to the leftmost column of the line containing the ’image’−keyword, issuing the [Ctrl]+([k][b]) sequence, moving the cursor to the leftmost column AFTER the line containing ’read−only’ and issuing the [Ctrl]+([k][k]) sequence and moving the cursor back to the leftmost column of the line containing the ’image’−keyword to do a final [Ctrl]+([k][c]) sequence. Ever worked with WordStar in the CP/M era? Anyway, make sure there are no empty lines in the file.*

Don’t mind the errors while shutting down after a kernel compilation, though)!

Configuring AX.25

A very interesting source of information is, of course, the AX25−HOWTO (in /usr/doc/HOWTO). A procedure is described on where to obtain some required packages, how to compile, install and configure them. It should be noted, however, that since Red Hat Linux 5.2 (kernel 2.0.36) no longer requires a kernel patch to de downloaded/compiled!

For this seminar, we have created an RPM that will install the AX.25 utilities directly, so you only need to configure it all. There’s one kernel−related file you should still create, however. It’s called `conf.modules` and should be added in the `/etc` directory. Here’s a sample (you should at least include the line holding ax25 and one of the TNC types; bold text shows the minimal required for KISS−links):

```
alias net−pf−3         ax25
alias net−pf−6         netrom
alias net−pf−11        rose
alias tty−ldisc−1      slip
alias tty−ldisc−3      ppp
alias tty−ldisc−5      mkiss
alias bc0              baycom
alias nr0              netrom
alias pi0a             pi2
alias pt0a             pt
alias scc0             optoscc  (or one of the other scc drivers)
alias sm0              soundmodem
alias tunl0            newtunnel
alias char−major−4     serial
alias char−major−5     serial
alias char−major−6     lp
```

It’s also advisable to add a file called `rc.serial` in the `/etc/rc.d` directory to set the appropriate line speed (standard 38k4 is used on most systems):

```
#!/bin/sh
setserial /dev/ttyS0 divisor 6 spd_cust
```

The second line sets the line speed to 19k2 for COM1 (115200/6=19200)

You can set the IRQ for a COM−port like this (COM3 at IRQ15):

```
setserial /dev/ttyS2 irq 15
```

Append the following line to download firmware into an MCB−152 (connected to COM1 and provided the firmware is in a file kiss.hex in /etc/rc.d):

```
cat /etc/rc.d/kiss.hex > /dev/ttyS0
```

After having created this file, issue

```
chmod a+x /etc/rc.d/rc.serial
```

Now insert the applications CD you received at the seminar, and make it visible to Linux by issuing

```
mount /mnt/cdrom
```

Next, install the RPM[7] using

```
rpm −i /mnt/cdrom/EXTRA/RPMS/ax25−utils−2.1.42a−4.i386.rpm
```

This RPM will place a few configuration files in the directory `/etc/ax25`. First to start with is the file `axports`, which maps ports (= TNC devices) to call−signs and some parameters. So, you must add an entry for each AX.25 device you connect to your com−puter. If you have a multi−port device (like an SCC−card), you must add as many entries as there are ports on the device. Here's a sam−ple (in the file that's installed from the RPM, replace %CALL% with the appropriate call−sign[8]):

```
# /etc/ax25/axports
#
# The format of this file is:
#
# name callsign speed paclen window description
#
KISS   ON0BAF−10  19200    256     4       KISS−link to node
```

First, the port name is listed (this will be used later as an identification to link ax−type interfaces to AX.25 ports). Next, you must configure the call sign for each AX.25 port (the call and optional SSID must re−main unique within the axports file). Finally, you set the line speed be−tween your PC and the TNC, the packet length used on that port, the maximal number of frames that may be transmitted in one burst and a description (all in **ONE line**!).

As soon as the axports file is ready, you can activate the AX.25 ports. We will only illustrate a KISS−link here: the AX25−HOWTO docu−ment provides extensive information on activating a particular TNC−flavor. For KISS−devices, activate a configured port using

```
kissattach /dev/ttyS0 KISS
```

if your TNC is connected to port COM1.

[7] *Unfortunately, a little error slipped into the RPM. Execute−permissions are not restored for a few scripts, so please issue the following command immediately after installing the RPM:*

```
chmod a+x /etc/rc.d/init.d/ax25
chmod a+x /etc/sysconfig/network−scripts/if*−ax
chmod a+x /etc/ax25/static−axroute
```

[8] *You can do this automatically using the stream editor: try*

```
sed ’s/%CALL%/on0baf/’ axports
```

for example.

Once your AX.25 port is activated, you can use commands like ’call’ (for connecting another AX.25 station) or ’listen’ (for monitoring the traffic). All these AX.25 commands take the AX.25 port name from the axports file as their first parameter (e.g. `call KISS on0baf` or `listen KISS −a`).

Next, you could edit the `ax25d.conf` file. This file configures the AX.25 daemon that will listen to and accept connect requests from re−mote stations. Here’s the sample (once more, replace %CALL% with the appropriate call−sign and replace the 127.0.0.1 on the last line with the IP address of your server):

```
# /etc/ax25/ax25d.conf
#
# ax25d Configuration File.
#
# AX.25 Ports begin with a ’[’.
#
[ON0BAF−11 VIA KISS]
NOCALL   * * * * * *  L
default  * * * * * *  − root  /usr/sbin/ttylinkd ttylinkd

[ON0BAF−12 VIA KISS]
NOCALL   * * * * * *  L
default  * * * * * *  − root  /usr/bin/irc irc −d %U 44.144.181.1
```

The entry with SSID 12 shows you how you can enable AX.25 users to enjoy the IRC network. You can have the AX.25 daemon started upon system boot or by manual start, which is achieved by issuing (use ’stop’ for halting the service manually)

```
/etc/rc.d/init.d/ax25 start
```

If you want to make sure that the AX.25 daemon is started automati−cally when the computer boots, run `linuxconf`, open the ’`Con−trol`’−group (move with cursor, open/close with enter−button), open ’`Control panel`’ and enter ’`Control service activity`’. Locate (using cursor up/down) the line with ’`ax25`’ and use the left/right cursor keys to enable/disable automatic boot.

Finally, you can edit `ttylinkd.conf`, which defines who should handle incoming AX.25 ttylink connects. It’s not necessary that this person is connected to the server system itself, the following would be a valid example for our server on0baf:

```
sysop=on1blu@on1blu.baf.be.ampr.org
```

The example above would contact the host on1blu.baf.be.ampr.org and notify the user on1blu on that host that a ttylink session is requested.

There are more configuration files for setting up NetRom (`nrports`), Rose (`rsports`), AXIP tunneling, RIP, axspawn (which provides tel−net−operation to AX.25−users), a PMS, etc. However, they are not within the scope of this seminar. Consult the AX25−HOWTO if you want to enable these services. Sample configuration files can be found in `/usr/doc/ax25−utils−2.1.42a/etc`. You might consider creating at least the nrports and rsports files for avoiding warning messages to appear upon starting AX.25. From the /etc/ax25 directory, issue:

```
touch nrports
touch rsports
```

So far for the bare AX.25 functionality.

Enabling AX.25 for TCP/IP

Now, we'll plug the AX.25 devices into the TCP/IP protocol stack. For enabling this, it's advisable to launch the kissattach command with an additional IP−address parameter by listing the '−i' option, followed by the server's IP address[9]:

```
kissattach −i 44.144.181.1 /dev/ttyS0 KISS
```

The command actually creates ax−devices and makes them visible to the protocol stack, so you can activate those devices using ifconfig (in some cases the kissattach command even activates the ax−interfaces itself).

Plugging in the AX.25 doesn't end here. We still need to define the network mask, set up some minimal routing (incl. AX.25 routing), add ARP entries, and so on.

To facilitate all this, I created some scripts that make AX.25−interfaces behave just as any other Linux−supported network interfaces (like Ethernet, dial−up links, etc.). With these scripts in place (as is achieved by the RPM), you simply need to edit:

- `/etc/sysconfig/network−scripts/ifup−ax` if you need to change the commands for putting your TNC into KISS−mode.

[9] *Your previous kissattach may still be running, prohibiting this one to be launched. Here's another Linux−trick. The* `ps` *command shows a list of active processes, the* `grep` *command searches for a text in an input string or in a file. Put these two together and you can search your existing kissattach process:* `ps ax | grep` *kissattach. This will result in 2 lines being printed: one with the kissattach command as you issued it, another with 'grep kissattach' itself, ignore the latter. Note the (process) number in the first column of the former line and issue '*`kill <process number>`*'. The characters 'a' and 'x' are actually options for the ps command: the 'a' makes sure you can see process belonging to all users, the 'x' lists processes that are not owned by some console (for input/output).*

- `/etc/sysconfig/network−scripts/ifdown−ax` if you want to change the commands for getting your TNC out of KISS−mode.
- `/etc/sysconfig/network−scripts/ifcfg−ax0` for setting interface parameters (read further for details).
- `/etc/sysconfig/static−routes` to configure some routes, using the format:
  `<device> net | host <IP address> netmask <mask> gw <gateway>`
  A typical example for a workstation in the baf−region would be:
  `ax0 net 44.0.0.0 netmask 255.0.0.0 gw 44.144.181.1`
- `/etc/ax25/static−axroute` to configure some AX.25 level−routes (digipeaters), using the format (first line in file must be `#!/bin/sh`):
  `axparms −route add <port> <destination> <via1>... [−ipmode v]`
  as in:
  `axparms −route add KISS on0kul−10 on0baf−1 −ipmode v`
- `/etc/ax25/static−arp` to add ARP−entries, formatted as:
  `<destination IP−address> <destination call>`
  like:
  `44.144.181.4 on1blu`

TCP/IP networking basics

Time for some networking theory now. In TCP/IP networks, hosts are assigned IP addresses. Compare this to telephone numbers that are assigned to telephone devices, fax devices, etc. An IP address is a 32−bit number. For representation purposes, they are grouped in 4 numbers of 8 bits each (meaning that each of the 4 numbers can have a value between 0 and 255). In their decimal representation, these 4 numbers are separated by dots and one such example is: 44.144.181.4

These numbers are not chosen randomly: actually, the entire IP address has 2 parts: a network number and a host number. Compare again with telephone numbers: in Belgium, a telephone number like 011/67.15.65 represents a geographical zone (011), a telephone switch (67) within that zone − these 2 numbers form a network identification − and a device number (15.65) − a host identification (please note that IP addresses normally don’t follow geographical classifications, but rather organizational classifications like companies, governments, etc.).

The part that stands for network identification isn’t always the same for all IP addresses. The sizes of the 2 parts are governed by the net−

work mask. A network mask is also a 32−bit number: a 1−value for a specific bit means that the corresponding bit in the IP address is part of a network identification, a 0−value that the corresponding bit in the IP address is part of a host identification. Only the ’left−part’ of an IP address can represent a network and only the ’right−part’ of an IP address can represent a host, or − in other words − if you read the network mask from left (most significant bit) to right (least significant bit), the bit−values are all 1, or are all 0 or start with 1 and change once from 1 to 0 (at the network/host boundary, that is). The network/host boundary does not need to correspond to an 8−bit boundary!

Here’s an example (network identification in bold):

netmask:	**255.255.255**.0
	11111111.11111111.11111111.00000000
IP address:	**44.144.181**.4
	00101100.10010000.10110101.00000100
	44 represents the ampr.org network
	44.144 represents the be.ampr.org network
	44.144.181 represents the baf.be.ampr.org network
	4 represents host number 4 within the network above

The 32−bit numbers that form an IP address can’t be used continuously, they are grouped in a few classes from which they can be taken.

Class name	*IP addresses*	*Default network mask*
Class A	0.0.0.0 − 127.255.255.255	255.0.0.0
Class B	128.0.0.0 − 191.255.255.255	255.255.0.0
Class C	192.0.0.0 − 223.255.255.255	255.255.255.0
Class D	224.0.0.0 − 239.255.255.255	*Multicast*
Class E	240.0.0.0 − 247.255.255.255	

Table 1IP address classes

As you might gather, class A addresses are reserved for worldwide multinational companies, class B addresses could typically be used for country governments and class C addresses are typically reserved for small and medium sized companies. Notice that our ampr.org domain (first octet is 44) is class A!

If you extend the network portion as opposed to the implicit network mask from the class of the IP address (Table 1), then you’re creating subnets. For the example above, both 44.144 and 44.144.181 are actu−

ally subnets of the 44−network.

Now a few extra restrictions:

- You can never use the first and last address within a network. For the example above, addresses 44.144.181.0 and 44.144.181.255 are not available. The first figures for the network address, the last is the broadcast address within the specified network.
- The network 0 can not be used: 0.0.0.0 is the local host (on the local network) − which is typically used as a source address after booting a host to obtain an IP address from a DHCP server, for example. 0.x.y.z is host x.y.z on the current network (may only be used as source address).
- The network 127 is the loopback interface (127.0.0.1 typically is the loopback address).
- Address 255.255.255.255 is a generic (not forwarded) broadcast ad−dress.
- The IP addresses 10.0.0.0−10.255.255.255, 172.16.0.0−172.31.255.255 and 192.168.0.0−192.168.255.255 are reserved for testing or private network usage. These IP addresses should never appear as sources or destinations on the Internet (no routes are de−fined for these addresses, so they won’t get very far anyway).

Now, we can start editing the `ifcfg−ax0` file:

- DEVICE is the interface name as it will be known to the TCP/IP protocol stack. For AX.25 interfaces, their names start with ’ax’, followed by a sequence number (e.g. ax0).
- IPADDR is the IP address of the interface (e.g. 44.144.181.1).
- NETMASK is the (sub)netmask that applies to that interface (e.g. 255.255.255.0, except for when you add additional subnets for your interfaces).
- NETWORK is the (sub)network address (e.g. 44.144.181.0).
- BROADCAST is the (sub)network broadcast address (e.g. 44.144.181.255)
- ROUTER is the default gateway on that interface. This item should not be used on servers/routers (they are, however, typically set on workstations to pass all traffic through the local server/router, which simplifies routing), except when it’s node is actually an IP router

(and configured to act so): c.f. XNET.

- ONBOOT can have values ’yes’ or ’no’ to indicate whether this interface should be activated automatically after booting
- BOOTPROTO states how an IP address is assigned to this interface: ’none’ (static IP address, provided above), ’dhcp’ or ’bootp’
- DEVPORT is the hardware port where the AX.25 device is connected (e.g. /dev/ttyS0 for COM port 1)
- AXPORT is the name of the AX.25 port as mentioned in the axports file (e.g. KISS)
- MTU is the maximum transmission unit that the link protocol can carry in 1 frame (or, in other words: the maximum number of bytes that fits in 1 AX.25 packet, which is 256)
- PERSISTENCE is a collision−avoidance parameter: read the AX.25 documentation
- SLOTTIME is a time−slice parameter for collision−avoidance: read the AX.25 documentation
- TXDELAY is a delay time between bringing the radio on air and actually start sending data in milliseconds (use values that are a factor of 10)
- USERCTL controls whether ordinary users may start/stop this interface (’yes’ or ’no’).
- DEFAXROUTE enables you to configure a default AX.25 route. This is especially handsome for TCP/IP server that only have a KISS−wire to a node. For our setup at Sint−Truiden, where our TCP/IP server is connected solely by a KISS−link to a PC FlexNet system, we configure on0baf−1 as default AX.25 route. This avoids the need to adjust the /etc/ax25/static−axroute file for each user.

This file is read by the `/etc/sysconfig/network−scripts/ifup−ax` and `/etc/sysconfig/network−scripts/ifdown−ax` scripts. If you need to put your TNC in KISS mode first, you can do this from either the `/etc/rc.d/rc.serial` script (using the `echo` command) − which will only try putting your TNC in KISS mode upon booting your PC −, OR in these 2 `ifup` and `ifdown` script files (the latter allowing you to get back out of KISS−mode automatically afterwards), allowing you to switch into and from KISS mode upon activating/de−activating the network interface (using the `/sbin/ifup ax0` and `/sbin/ifdown ax0` commands, the `netcfg` utility, the `usernet` utility, etc.). Here are a few examples (comment − put ’#’ characters in front of − the lines you don’t want) for the `/etc/sysconfig/network−scripts/ifup−ax` script:

```
...
if [ "foo$2" = "fooboot" −o "foo$2" = "foore−init" ]
then
  echo −e "\027\001OHON\027\r" > ${DEVPORT}
               # Pull AEA−TNC out of HOST−mode
  echo −e "\300\300\377\300\r" > ${DEVPORT}
               # Pull a TNC out of its current KISS mode
  echo −e "\r\140A\rkiss on\r" > ${DEVPORT}
               # Put an AEA dual−port TNC in single−port KISS mode
               # (if channel−switch is ’‘’−character (c.f. 140)
# echo −e "\015\015\033@k" > ${DEVPORT}
               # Put a TNC−2 into KISS−mode
fi
...
```

Likewise for `/etc/sysconfig/network−scripts/ifdown−ax`:

```
...
if [ ! −d /proc/$PID ]; then
  /etc/sysconfig/network−scripts/ifdown−post $1
  echo −e "\300\377\300\r" > ${DEVPORT}
               # Pull AEA−TNC from KISS−mode
  echo −e "\r\1400\rhost on\r" > ${DEVPORT}
               # Put AEA−TNC into HOST−mode
  exit 0
fi
...
```

Repeated further down the file:

```
...
sleep 2
if [ ! −d /proc/$PID ]; then
  /etc/sysconfig/network−scripts/ifdown−post $1
  echo −e "\300\377\300\r" > ${DEVPORT}
  echo −e "\r\1400\rhost on\r" > ${DEVPORT}
  exit 0
fi

kill −KILL $PID > /dev/null 2>&1
if [ −d /proc/$PID ]; then
  logger −p daemon.info −t ifdown−ax0 "ifdown−ax0 unable to kill
kissattach on $DEVICE" &
else
  /etc/sysconfig/network−scripts/ifdown−post $1
  echo −e "\300\377\300\r" > ${DEVPORT}
  echo −e "\r\1400\rhost on\r" > ${DEVPORT}
fi
...
```

Next to the usual ifcfg−files, you may also find an ifcfg−file ending with ’:0’ from your AX.25 installation. Interface configuration files with colons (’:’) in their names are actually aliases for the mentioned interface. If you don’t plan on using it (which you probably don’t), delete it.

Routing

With TCP/IP and Packet Radio, we have two levels of routing:

- AX.25 level routing, which defines routes to connect AX.25 TCP/IP servers across intermediate nodes: if two such servers can not communicate directly without passing a few nodes, one needs to define AX.25 routes.
- IP level routing, which actually defines routes to connect networks together (that's why they speak about 'Internet': it's just a connection of networks, tied together by IP routes).

Illustration 2 shows 2 computers (with TCP/IP running) and a node. If the station on the left wants to connect the station on the right, it has to pass the node. This level of connection is achieved through AX.25-level routing (saying for example, that on0nol−12 is reachable VIA on0baf−2).

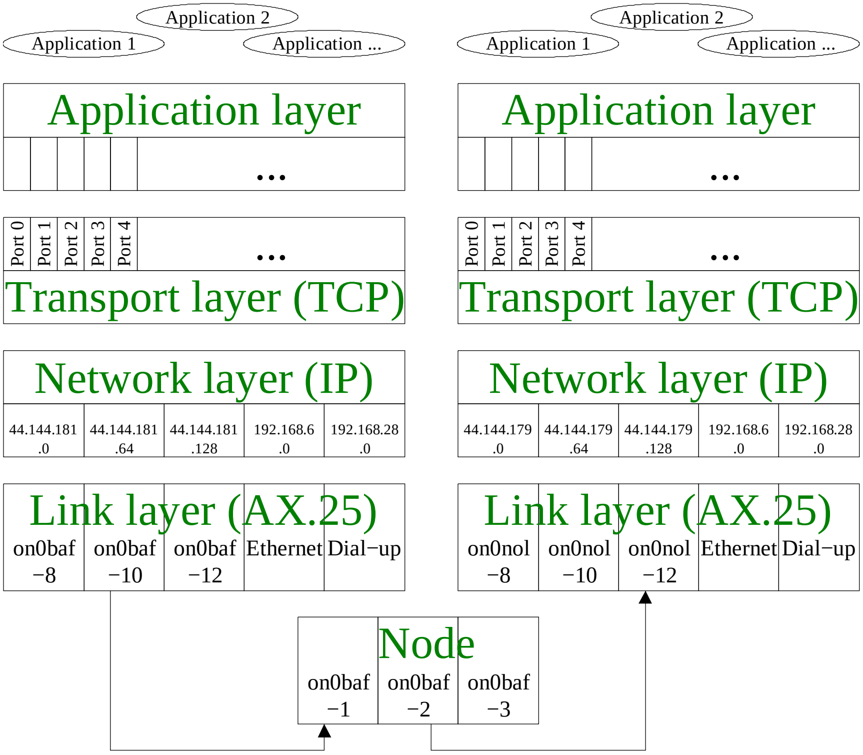


Illustration 2Communicating TCP/IP protocol stacks

This information can be entered in the file `/etc/ax25/static-axroute`. This file is a script (and could be launched from the shell). It's format is:

```
#!/bin/sh
/usr/sbin/axparms −route add <AXPORT−name> <destination
```

```
call> <via 1> <via 2> ... [−ipmode v]
```

For example:

```
/usr/sbin/axparms −route add KISS on0kul−10 on0baf−1 on0lvn−6
/usr/sbin/axparms −route add KISS db0pra−6 on0baf−1 −ipmode v
```

The optional ipmode option allows you to configure a virtual circuit (AX.25 connect) to be setup for this route.

At IP−level, the station at the left basically only knows where to put messages for 44.144.181.xxx stations (on the local interface to be more specific). A route must be defined to the network 44.144.179.0, however (which is still on a local interface: the AX.25 route will make it pass through the node). The users of on0baf, however, also know implicitly how to contact 44.144.181.xxx stations, but not how to contact 44.144.179.xxx stations. That’s why we tell the users to IP−route all 44.xxx.xxx.xxx traffic through 44.144.181.1. If one such user wants to contact 44.144.179.1, his local host will route it to 44.144.181.1 (meaning he will send a packet to AX.25−destination on0baf, with IP−destination 44.144.179.1), 44.144.181.1 will then further forward the IP−frame to 44.144.179.1 (from AX.25−source on0baf to AX.25−destination on0nol via AX.25−digipeater on0baf−2). Static IP−routes can be configured in the configuration file `/etc/sysconfig/static−routes`, which has the format:

```
<interface> <’host’ | ’net’> <destination IP address> [’netmask’ <net−
mask>] <interface | ’gw’ router IP address>
```

For example:

```
ax0 host 44.144.179.1 ax0
ax0 net 44.144.179.0 netmask 255.255.255.0 gw 44.144.179.1
```

The example will add a host−route through the local interface and a network−route via the newly added host. For the routers themselves, it is advisable to define a route to network 44.0.0.0 through a HAM−gateway nearby:

```
ax0 host 44.144.0.1 ax0
ax0 net 44.0.0.0 netmask 255.0.0.0 gw 44.144.0.1
```

Furthermore, it is advisable that explicit routes are added for neighboring regions, etc. In a near future, we hope to eliminate the need of entering IP routes through the use of dynamic routing protocols (like RIP, but rather something like RSPF).

Mapping IP addresses to hardware addresses

You might have guessed that there appears to be a need to do some mapping between network−level IP addresses and link−level callsigns. You’re right − it’s called address resolution − and, even better, YOU will need to configure it. Please locate the file `/etc/ax25/static−arp`. Just enter the pairs of IP addresses and corresponding call−signs for all your users (since Windows might not be answering ARP−requests: if your user uses an MCB−152, for ex− ample, Windows is operating in SLIP−mode and therefor never re− sponds to ARP requests) and all TCP/IP systems you use as gateways for your routes if those gateways are only reachable through nodes (ARP−requests won't travel through nodes).

The blocking of ARP−requests in nodes is actually a though prob− lem (the MCB−152 problem can be fixed by adjusting the SLIP−firm− ware), forcing us to manual maintenance. Basic principle is: you need to add an ARP−entry, each time you assign someone a new IP address.

We're looking forward to future AX.25 developments in Linux...

Routing: an example

Let's say that I (ON1BLU), a user of on0baf, want to contact ON1AFN, a user of on0nol. Here's some data:

ON1BLU
IP: 44.144.181.4
host route 44.144.181.1 via ax0
net route 44.0.0.0 via 44.144.181.1

ON0BAF−1..4 (Flexnet node)

ON0BAF−10
IP: 44.144.181.1
host route 44.144.179.1 via ax0
net route 44.144.179.0 via 44.144.179.1
implicit net route to 44.144.181.0 to ax0 due to interface configuration
ARP entry 44.144.179.1 on0nol−10
AX.25 route on0nol−10 via on0baf−1

ON0NOL−1..4 (Flexnet node)

ON0NOL−10
IP: 44.144.179.1
host route 44.144.181.1 via ax0
net route 44.144.181.0 via 44.144.181.1
implicit net route to 44.144.179.0 to ax0 due to interface configuration
ARP entry 44.144.181.1 on0baf−10

AX.25 route on0baf−10 via on0nol−1

ON1AFN
IP: 44.144.179.100
host route 44.144.179.1 ax0
net route 44.144.179.0 via 44.144.179.1

Now let’s say that I try opening a peer−to−peer chat to on1afn using: `ttylink 44.144.179.100`.

1. ttylink is an application as shown in Illustration 2. Data is sent down the protocol stack and when it arrives at the network layer an IP frame is constructed with source IP 44.144.181.4 and destination IP 44.144.179.100.
2. Then, while still in the network layer, a route is searched to reach that destination. This search starts from an exact match (a host route for 44.144.179.1) and for as long as the search fails, that search will go ’less accurate’: a subnet−match, a network−match and finally − if present − a default route. If no match is found, nothing will be sent.
   In practice, a host route for 44.144.179.100 is not found at on1blu, no subnet route to 44.144.179.0 is found at on1blu, but a network route to 44.0.0.0 is found at on1blu: via router 44.144.181.1.
3. Still in the network layer, a route is searched now for 44.144.181.1, an exact match is found: through interface ax0.
4. An interface is found now, so data can be pushed down. Before actually doing so, the destination link−address (call−sign in case of AX.25) that goes with 44.144.181.1 is searched (ARP request on the radio or from local configuration file, like static−arp): on0baf−10: in the link layer an AX.25 packet will be constructed with source on1blu and destination on0baf−10.
5. Packet is sent and arrives, through the node, at the TCP/IP server: destination is on0baf−10, so data (=IP−frame) is pulled up into network layer.
6. Destination IP is 44.144.179.100 so it’s not for the server. Host route to 44.144.179.100 is searched, but not found; subnet route to 44.144.179.0 is searched for and found: through router 44.144.179.1.
7. Host route to 44.144.179.1 is found through interface ax0.

8. Interface found: the ORIGINAL IP−frame (source 44.144.181.4, destination 44.144.179.100) is pushed back down. To do this, an ARP−entry is looked for 44.144.179.1, which is mapped to on0nol−10.
9. Back in the link layer, an AX.25 route is found for destination on0nol−10: via on0baf−1. A NEW AX.25 packet is created with source on0baf−10 and destination on0nol−10 VIA on0baf−1 (con−taining the original IP frame as data).
10. Packet is sent, and arrives through node on0baf−1 (and thanks to Flexnet routing automatically through on0nol) at on0nol−10. Des−tination call is on0nol−10, so data (=IP−frame) is pulled up into network layer.
11. Destination IP is 44.144.179.100, so it's not for the server. A host route to 44.144.179.100 is searched, but not found. A subnet route exists implicitly to 44.144.179.0 through ax0: this one is used.
12. We have a destination interface, so upon pushing the original IP−frame (source IP 44.144.181.4, destination IP 44.144.179.100) down once more, ARP mapping for locally assigned IP address 44.144.179.100 is searched: on1afn.
13. At the link layer, another new AX.25 packet is created (source on0nol−10, destination on1afn), containing the IP−frame once more as data.
14. Packet is sent and arrives at station on1afn: destination is on1afn, so packet−data (=IP−frame) is pulled up again into the network layer.
15. Destination IP is 44.144.179.100, which is the current station, so data is pulled up through transport layer and application layer into the ttylink−listener.

Now the reverse process can start to acknowledge the connection and then real data−transfer can start. And you thought it was simple...

ax25rtd

Ax25rtd is the AX.25 route daemon. It's not being handled in this seminar (since I don't have any experience with it). I'm certain, how−ever, that this daemon will be talked about a lot pretty soon, as it ap−pears to add the following automatically (if I understand correctly), from the monitored traffic:

- IP host route
- ARP entry
- AX.25 route

Another breakthrough might be the 2.2.x Linux−kernel (especially starting from Red Hat Linux 6.1).

Configuring services

Now it’s time to start configuring the Linux services.

BIND (Domain Name System)

So we’ve had the address resolution protocol that maps IP addresses to link−addresses (such as call−signs), now it’s time to look into the mapping of domain names to IP addresses. From our applications (like web browsers), we tend to use domain names (like www.baf.be.ampr.org). Before such an application actually can contact the remote system, it will need to obtain the IP address of that remote system.

Since the new IP addressing scheme, each region is authoritative for distributing IP addresses and the domain names that go with it. That’s why our servers no longer will be ’cache−only’ servers but actual primary zone−servers.

First, we’ll need to check the contents of `/etc/resolv.conf` (first search argument is an example for on0baf):

```
search baf.be.ampr.org be.ampr.org ampr.org
nameserver 127.0.0.1
```

The last line will make sure that your server takes name resolving responsibility. Another file involved is `/etc/host.conf`:

```
order hosts,bind
multi on
```

The first line states that first the `/etc/hosts` file should be checked for name resolving, then the name server should be queried.

In the file `/etc/hosts`, you’re advised to add some (IP−address, domain name)−pairs for frequently used stations (for example, routers you use):

```
127.0.0.1    localhost    localhost.localdomain
44.144.179.1 on0nol       on0nol.nol.be.ampr.org
```

And you may also want to name a few networks in `/etc/networks`:

```
amprnet   44.0.0.0
bafnet    44.144.181.0
ethnet    192.168.6.0
slipnet   192.168.28.0
plipnet   192.168.34.0
```

And now, the real work, first we need to edit `/etc/named.conf`, here’s an example:

```
options {
        directory "/var/named";
        /*
         * If there is a firewall between you and nameservers you want
         * to talk to, you might need to uncomment the query−source
         * directive below.  Previous versions of BIND always asked
         * questions using port 53, but BIND 8.1 uses an unprivileged
         * port by default.
         */
        // query−source address * port 53;
};

// Cache priming zone for root domain
zone "." {
        type hint;
        file "named.ca";
};

// Local reverse pointer zone
zone "0.0.127.in−addr.arpa" {
        type master;
        file "named.local";
};

// Zone we’re authoritive for
zone "baf.be.ampr.org" {
        type master;
        file "named.baf";
};

// Reverse pointer zone we’re authoritive for
zone "181.144.44.in−addr.arpa" {
        type master;
        file "named.rbaf";
};
```

The root (".") zone is required to define a reference toward a hierarchically higher level name server that can answer requests outside your zone. **If you’re running local tests, while you’re not connected to a network, put comment signs (’//’) before the entire root zone block (otherwise you will spend a lot of time waiting on DNS lookups performed by various servers − like sendmail while booting − or some servers will simply not function − like IRC)**. If on0kul is your nearest server that provides an up−to−date ampr.org domain file, your `/var/named/named.ca` file should contain these 2 lines − always a combination of a NameServer−entry and the DNS−to−IP mapping for that name server (put all other lines in comment!):

```
.                        3600000 IN      NS      on0kul−10.ampr.org.
on0kul−10.ampr.org.      3600000 IN      A       44.144.0.1
```

The zone "0.0.127.in−addr.arpa" is simply required. You can leave the file `/var/named/named.local` unchanged.

Next, you need to define the zone you’re responsible for. Here’s the authoritive domain file `/var/named/named.baf` − see named.conf definitions − for baf.be.ampr.org (especially notice the use of periods at the end of domain names):

```
;
; Zone file for baf.be.ampr.org
;
@ IN SOA on0baf.baf.be.ampr.org. on1blu.on0baf.baf.be.ampr.org. (
                              1999010900     ; serial
                              28800          ; refresh
                              7200           ; retry
                              3600000        ; expire
                              86400)         ; minimum
           IN NS    on0baf.baf.be.ampr.org.
           IN MX    10        on0baf.baf.be.ampr.org.
           IN MX    20        on0kul−10.ampr.org.
           IN HINFO "Pentium" "Linux 2.0.36"
           IN TXT   "UBA−RST domain located in Sint−Truiden"
;
; Server
;
localhost  IN A     127.0.0.1
on0baf     IN A     44.144.181.1
           IN MX    10        on0baf
           IN MX    20        on0kul−10.ampr.org.
ns         IN CNAME on0baf
ftp        IN CNAME on0baf
www        IN CNAME on0baf
mail       IN CNAME on0baf
news       IN CNAME on0baf
irc        IN CNAME on0baf
```

Important to notice is that white−space is the keyword to distinguish the fields of a resource record (RR): it doesn’t matter which one you use (space, tab, ...), but it does matter how many you use (for example there may be only one tab between ’IN’ and ’A’. In the example above, no adjacent white−spaces are used (except for the time parameters of the SOA record): only single space or single tabs are used.

The ’IN’ code signifies INternet addressing applies.

Here’s a short description of the RR records used in the example (the @ at the beginning represents the domain zone origin and is actually a reference to the zone name specified in named.conf; in the following lines no name is provided in the first column, resulting in the latest one − @ or baf.be.ampr.org − being ’dragged along’):

- SOA (Start Of Authority): should be the first RR in a zone definition. It’s data is respectively the name of the server that’s authoritative for the zone being defined, the E−mail address of the domain zone administrator (with the ’@’ being replaced by a ’.’) and as parameters (between parentheses) a serial number (**please change it everytime you apply modifications to this file**) and some time−out values that control when the cache of remote systems that queried your server earlier should be invalidated.
- NS (Name Server): indicates which domain server should be contacted to resolve addresses in the specified domain. It’s only data is the FQDN (Fully Qualified Domain Name) of the name server.

- A (Address): is a map−entry between a domain name (first column) and an IP address (after the RR−type − ’A’).
- MX (Mail eXchanger): identifies where mail to the host in the first column should go to (and some preference value to help selecting one from a list): this will be especially important for your user entries: read further. For the block being discussed, the MX record will make sure that E−mails to on1blu@baf.be.ampr.org will be routed to on0baf.baf.be.ampr.org.
- CNAME (Canonical NAME): can be used to define ’aliases’. The name stated in the first column will be transformed into the canonical representation (data + zone expansion if data doesn’t end with a dot). An address (A) record will then be searched on this canonical name for resolving.
- HINFO (Hardware INFO): indication of CPU and OS used at the host (both between double quotes).
- TXT (TeXT): Short description of the host (between double quotes).

Still in the same file, you should add, for each user (adding users is handled later), an address, at least one mail exchanger, a text and possibly one or more canonical names; for example:

```
on4awm      IN A     44.144.181.3
               MX    10        on0baf
               MX    20        on0kul−10.ampr.org.
               TXT   "Walter − Sint−Truiden"
on1bzo         CNAME on4awm
```

Concerning the mail exchanger, the two mail exchangers above make sure that if one attempts to write an E−mail to on4awm@on4awm.baf.be.ampr.org, the mail will be sent to on4awm@on0baf.baf.be.ampr.org and should on0baf.baf.be.ampr.org not be reachable for some reason, the mail will be sent to on0kul−10.ampr.org. Notice the use of canonical names to intercept changes in the call sign. Also notice that the ’IN’ keyword is not repeated on each line: a extra white−space character is still necessary, though (for instance, 2 tab characters are inserted before the ’MX’ keywords).

Generally, these domain files require FQDNs (ending with a dot), but abbreviations can be used: if the name isn’t terminated by a dot, the origin (.baf.be.ampr.org. in the examples above, since the zone is "baf.be.ampr.org" from the named.conf file) is appended to the abbreviations (so on1bzo IN CNAME on4awm is actually on1bzo.baf.be.ampr.org. IN CNAME on4awm.baf.be.ampr.org.)

And finally you also must provide a means for ’reverse lookup’. This is realized in a separate file (/var/named/named.rbaf for our example), and looks like this:

```
@ IN SOA on0baf.baf.be.ampr.org. on1blu.on0baf.baf.be.ampr.org. (
                                            1999010900     ; serial
                                            28800          ; refresh
                                            7200           ; retry
                                            604800         ; expire
                                            86400)         ; minimum
           IN NS    on0baf.baf.be.ampr.org.
;
; Server
;
1          IN PTR   on0baf.baf.be.ampr.org.
```

Consequently, the 1 is escalated to 1.181.144.44.in−addr.arpa. A new type of resource record is used: PTR (PoinTeR), which maps a reverse noted IP address onto the corresponding FQDN. This file must contain one pointer record for each user too:

```
3          IN PTR   on4awm.baf.be.ampr.org.
```

Don’t forget to issue /usr/sbin/ndc restart after every change to the domain or configuration files.

Telnet (remote login)

No specific configuration required here: most of the information is taken from the registered users. Conditions are that a valid shell is specified and that this shell is listed in the /etc/shells file. Re−member that the user ’root’ can never log in directly from a remote telnet session (use ’su’ command once logged in using your regular user account).

We advise not to give telnet−access by default. Instead, add a line containing the text ’/bin/ftponly’ to the file /etc/shells. Next, create a new file /bin/ftponly with the following contents:

```
#!/bin/sh
echo "To avoid users to accidently arrive in a Linux−prompt,"
echo "we disabled telnet access by default."
echo "If you’re familiar with Linux and you want to experiment,"
echo "contact the operators to obtain telnet−access"
echo "(sysop@on0baf.baf.be.ampr.org)."
echo "Best 73’s!"
```

and add execute permission to this file, using the command

```
chmod a+x /bin/ftponly
```

FTP (File Transfer Protocol)

The FTP−server itself requires minimal configuration (/etc/ftpaccess). You should configure the SysOp E−mail ad−

dress, like:

```
email sysop@on0baf.baf.be.ampr.org
```

You should also add the following line (after the class line, perhaps) to enable your regular users uploading their own webpages:

```
guestgroup popusers
```

To allow these users to replace their own web−files, remove ’guest’ at least from the delete, overwrite and rename revocation lists, leaving something like:

```
delete          no       anonymous
overwrite       no       anonymous
rename          no       anonymous
```

Finally (especially for systems that provide anonymous access), you should add write access for an incoming−directory.

Add these two lines at the bottom of the file:

```
upload  /home/ftp   *          no
upload  /home/ftp   /incoming  yes  ftp  ftp  0644   dirs
```

The first line will make sure that, by default, no write access is granted (anonymous users can not see beyond /home/ftp: they only can access subdirectories of /home/ftp). The second lines provides write access for the incoming subdirectory (under /home/ftp): files that are created there by anonymous users will be owned by user FTP and by group FTP, providing read/write access to the owner (’FTP’) and read access to the owner−group and everyone else.

After leaving the file /etc/ftpaccess, you still need to create a new directory /home/ftp/incoming and make ftp the owner using chown ftp:users /home/ftp/incoming. Finally, allow your SysOps to manipulate the contents: chmod g+w /home/ftp/incoming

Main consideration should be permissions. This is a complex topic. FTP sessions have the /home/ftp directory as ’root mount point’. This means, they have no way to access /bin, /etc and the like. The missing of /etc (and therefor /etc/passwd), for example, results in file−ownership being reported as user IDs and group IDs instead of names, when an FTP−user asks for the directory contents. That’s why we create the SynchUsers script in /home/ftp (don’t forget to issue chmod a+x SynchUsers after saving the file) that places a stripped passwd file somewhere within the /home/ftp directory to allow names to be printed (this script should be run each time a new user is added to the system):

```
#!/bin/sh
sed −e ’s/:[^:]*:/:*:/’ /etc/passwd > /home/ftp/etc/passwd
cp /etc/group /home/ftp/etc/group
```

Likewise, you will also find directories `bin`, `lib` and `etc` under `/home/ftp` to provide some commands to FTP−sessions.

The structure you can find under `/home/ftp` should also be repeated in your user’s home directories (except for `pub` and `incoming`, of course) for those users that have been created in the `popusers` group. Remember we made `popusers` a guestgroup earlier on: this restricts access for those users in a way that they can not go ’higher’ than their own home directory.

For anonymous access, typically a directory `/home/ftp/pub` is provided where files − available for download to everyone − are stored. Consider usage of (symbolic) links and, if you want to make the contents of a (shareware?) CD available, you could even consider mounting your CD−ROM to `/home/ftp/pub`.

sendmail (Simple Mail Transfer Protocol)

While configuration of sendmail is fairly limited to interfacing to the legacy Packet Radio network (linking to BBS−systems), we need to make one note. Sendmail requires the availability of MX−records for destination hosts in E−mail addresses. For example, if we − in Sint−Truiden − want to mail to on4aak@on4aak.osk.be.ampr.org, sendmail will require an MX record for on4aak.osk.be.ampr.org. In practice, this requires DNS zone delegation to be in place. Anyhow, while there are a few sendmail options related to this topic, it might be wise to choose for a solution based on the `FallbackMXhost`−option. If sendmail would be unable to find an MX−record, the mail would be forwarded to an intermediate mail−server. Find a nearby server that has up−to−date DNS records and also provides mail services. For us, we uncom−mented and configured the following line in the `/etc/sendmail.cf` file:

```
# fallback MX host
O FallbackMXhost=on0kul−10.ampr.org
```

Also consider setting your server name, by editing the line that starts with keyword `Cw` in the same file:

```
Cwon0baf.baf.be.ampr.org
```

Optionally, however, you could configure some aliases: in the `/etc/aliases` file, you could add entries at the bottom of the file, like:

```
root:       on1blu
sysop:      on1dct, on4awm, on1blu, on1dia, on1dds, on7qj
administrator:          on1blu
```

Don’t forget issuing `newaliases` after making changes to this file!

In the file `/etc/sendmail.cw` you may also list alias−names for your mail−server:

```
# sendmail.cw − include all aliases for your machine here.
mail.baf.be.ampr.org
on0baf.ampr.org
localhost
```

If your neighbor IP region doesn’t provide mail services, the users of that region might try using your mail server. They will, however, receive error messages while doing so, stating that relaying is not supported. We have also detected this behavior in an XNET environment, where an XNET router separates the server from the rest of the network. In such cases, edit the file `/etc/mail/name_allow` and add the domain regions you want to allow users to relay from, for example: `oud.be.ampr.org`

Imapd (Post Office Protocol)

No configuration required at all (it’s not even possible)! This system runs fully automatic and takes all its data from the user account information.

Apache (web−server)

Two file need some minor editing:

- `/etc/httpd/conf/httpd.conf`: you should set the `ServerAdmin` attribute (e.g. sysop@on0baf.baf.be.ampr.org) and, especially, the `ServerName` attribute (e.g. www.baf.be.ampr.org).
- `/etc/httpd/conf/srm.conf`: you will find an entry `User−Dir`. You have 2 options: either you create sub−directories for all your users (in which case you specify the absolute[10] path−name to this HTML root−directory, from where you will create all those user−directories: for example: `/home/httpd`) or you create a specific directory (like `public_html`) in each user’s home−directory

[10] *Meaning the path will start with ’/’*

(in which case you only specify the name of that directory − no leading '/'!). For as long as HTTP−posting is not yet possible and YOU are responsible that users can upload their web−pages easily using FTP, it's advisable to take the second option (which is the default). This has implications on security, however: at least make sure that the world has only 'execute' permissions to your user's home directories (and the specific HTML directory − e.g. `public_html`).

Your server's main web−pages reside in `/home/httpd/html`. If you provide a file `index.html`, surfers can simply enter your site URL (e.g. http://www.baf.be.ampr.org). Allow me to add a suggestion here: at Safraanberg, all SysOps are in group `users`. To allow all SysOps making modifications to these public webpages, I issued the following commands:

```
chgrp −R users /home/httpd/html
chmod −R g+w /home/httpd/html
```

You could also apply these permissions to `/home/httpd/cgi−bin` if you want to allow them to create server−side scripts (notice that server−side scripts are generally considered perfect candidates for severe security breaches!).

Surfers can access your user's personal web pages by appending their home−directory to the URL (e.g. http://www.baf.be.ampr.org/~on1blu if a file `index.html` is present there).

INND (InterNet News)

This is another big one with lots of configuring. Let's have a quick browse through (sample configuration files are already provided with short descriptions; man−pages are available for most of these configuration files!):

Edit the `/etc/news/inn.conf` file to contain the appropriate references and descriptions. For example:

```
domain:         baf.be.ampr.org
organization:   BAFARA/UBA−RST news server installed in Sint−Truiden
server:         on0baf.baf.be.ampr.org
```

User access is controlled by the file `/etc/news/nnrp.access`. At our radio frequencies, we never disallow access to anyone, so comment all lines present and add this one at the end (for fields host pattern, allowed operations, user−name, password and access pattern):

```
*:Read Post:::*
```

Consider editing the file `/etc/news/expire.ctl`: expiration of old messages is configured here at news−group level.

When you start linking up with other news servers, you will need to edit the `/etc/news/hosts.nntp` file. This file should contain the names of all servers you expect news−feeds to come from and, optionally, the individual password you expect from those remote systems before they can start feeding news (hosts that are not listed in this file are handed over to the nnrpd daemon that serves users). This file could also play a role when making a gateway. If you use JNOS as a gateway, you would have to add the JNOS hostname to this file. The file could have an entry like this:

```
on0ob.oud.be.ampr.org:
```

For feeding news towards other systems, 4 files need editing:

- The file `/etc/news/innfeed.conf` contains general options for the remote sites. At the very least add peer entries for the servers you will exchange news with. It’s advisable to comment existing peers (that have references to real Internet servers) using the ’#’ character. Here’s an example of a minimal peer definition:
  ```
  peer on0ob {
              ip−name:       on0ob.oud.be.ampr.org
  }
  ```
- The file `/etc/news/newsfeeds` controls what will be fed and how. Very sophisticated stuff can be realized here, we’ll just limit to listing site−names, the news groups that need be forwarded to that remote site and doing it internally using files. Leave the "sites" ’ME’, ’Crosspost’, ’Overview!’ and ’Innfeed!’ in place. Make sure no Internet sites are active. Now you can start adding feed configurations (representing the site name, listing news groups that should be fed, the file−mechanism ’Tf’ should be used and the file should contain message IDs and relative paths to the file that actually contains the message ’Wnm’):
  ```
  on0ob\
              :ampr.*,comp.*,local.*,!local.rst\
              :Tf,Wnm:
  ```
- The file `/etc/news/nntpsend.ctl` lists the the remote servers you will actually connect to. You can also provide options like maximal batch size (news−feed will be truncated if the size is exceeded) and you can provide option to pass to innxmit. Here’s another example (first column should match the site name as mentioned in news−feeds):
  ```
  on0ob:on0ob.oud.be.ampr.org::
  ```

- When necessary you can enter the passwords, required for being able to start your news−feed toward each individual remote site, in the file `passwd.nntp`.

Please notice you must always configure BOTH directions!

All you still need to do is adding news groups. This is done using the ’newgroup’−command of the ctlinnd utility: `ctlinnd newgroup <group name> <allowed operations> <creator>` (consult the description of the fourth field in the active file − using man active − to find the codes for <allowed operations>). Here’s an example (that will allow local postings):

```
ctlinnd newgroup local.debug y on1blu
```

The ctlinnd−utility is also used for restarting the news server (required when a configuration file has changed); a reason for the restart must be provided for the logs:

```
ctlinnd reload all Added a new feed−site
```

Consider visiting our project website (http://www.qsl.net/on1blu), where you can download a script to generate a pretty standardized newsgroup−tree automatically.

The NNTP−protocol (and also the INN daemon) allow for news groups to be created automatically across servers: no experiments on that have been performed yet.

Undernet IRC (Internet Relay Chat)

There’s no IRC daemon delivered with a standard Red Hat distribution. From the Internet, however, the sources for one of the IRC networks (Undernet) is available under the GPL. We’ve downloaded it for you and we’ve put it all into a single RPM that can be installed easily.

Make sure your CD−ROM with our extensions is mounted and issue:

```
rpm −i /mnt/cdrom/EXTRA/RPMS/ircu−2.10.06−1.i386.rpm
```

In `/etc/ircd`, edit the files `ircd.motd`, `remote.motd` (max. 3 lines!) and finally `ircd.conf`, which should at least contain the following lines (example for on0baf, in the appendixes you can find a printout of `example.conf` that’s normally distributed with the package and that contains a lot of explanation):

```
M:on0baf.baf.be.ampr.org:*:UBA−RST TCP/IP server, Sint−Truiden:7000:1
A:UBA−RST − Bafara:Undernet IRC server:IRC Admin (on1blu@on0baf.baf.be.ampr.org)
Y:90:90:300:1:1700000
Y:80:90:300:1:1700000
Y:70:90:300:1:1700000
Y:50:90:300:10:1700000
Y:10:90:0:100:160000
```

```
Y:2:90:0:5:80000
Y:1:90:0:400:160000
I:*@*:1:Unresolved::1
I:Resolved::*@*::1
I:44.144.181.*::*.baf.be.ampr.org::10
d:*::directcon(*)
O:*@*.baf.be.ampr.org:<encrypted password from /etc/passwd>:on1blu
P::::4400
P::::7000
P:*:::6667
```

Check that all these files are owned by user and group `nobody`. The first line defines the name, the description, the main port (7000) and a unique ID (1−64) within a connected network of IRC servers. The second line adds additional descriptions and contact information. The ’Y’−lines configure parameters based on connection classes. The ’I’−lines classify the incoming connections. The ’d’−line defines what connections are refused (’D’) and what connections are not automatically initiated (’d’). The ’O’−line defines an operator (last column contains the nickname he uses when doing IRC). And, finally, the P−lines map connections to ports, based on the origin of the connection. Once more: consult the example.conf appendix and you could also take a look at http://www.undernet.org.

Finally, you have the option of having the daemon waiting at all times or to have it launched (and stopped) from the inetd ’super’−daemon. The former option is realized by starting linuxconf, going to control, then control−panel and finally ’Control service activity’, locating ircd and enabling it. The latter option, which we prefer as it appears to be far more stable, requires editing `/etc/inetd.conf`, like this:

```
...
talk     dgram    udp   wait   root      /usr/sbin/tcpd    in.talkd
ntalk    dgram    udp   wait   root      /usr/sbin/tcpd    in.ntalkd
#dtalk   stream   tcp   wait   nobody    /usr/sbin/tcpd    in.dtalkd
irc      stream   tcp   wait   nobody    /usr/sbin/ircd    ircd −i
#
# Pop and imap mail services et al
...
```

After doing this editing, you should activate it by issuing (or you could reboot the PC):

```
killall −HUP inetd
```

NFS (Network File System)

NFS will probably only be used by UNIX−gurus, to have server file locations mounted on their local file−system.

Just one file needs configuration on the server: `/etc/exports`. Please consult the man−pages (`man exports`) for full details, here’s an example, however:

```
/home/on1blu      on1blu(no_root_squash)
/etc              on1blu(ro,no_root_squash),on4awm(ro,no_root_squash)
```

If users want to use the exposed mount point, they could edit `/etc/fstab`:

```
on0baf:/home/on1blu  /mnt/ON0BAF/home  nfs noauto,user 0 0
on0baf:/etc          /mnt/ON0BAF/etc   nfs noauto,ro   0 0
```

After doing this, they can mount such an exported mount point by simply issuing mount `/mnt/ON0BAF/home`

SMB (Session Management Block)

SMB, or Samba, is better known as the software that’s responsible for maintaining the network neighborhood as known by Windows systems. Basically, one file needs configuration: `/etc/smb.conf`. Please consult the man pages (`man smb.conf` provides a full description on all options that are available for use in `smb.conf`), the SMB−HOWTO, etc. Especially if you want to use a Linux−SMB system together with Windows NT SP3+, you will need to take additional actions for user authentication. Surely read the documents in `/usr/doc/samba−1.9.18p10/docs.` The comments in the `/etc/smb.conf` configuration file are also of great help.

As an example, here is what we did:

```
...
   workgroup = UBARST
...
   server string = ON0BAF UBA−RST TCP/IP server
...
   hosts allow = 192.168.6. 44.144.181. 127.
...
   encrypt passwords = yes
   smb passwd file = /etc/smbpasswd
...
   username map = /etc/smbusers
...
   interfaces = 192.168.6.0/24 44.144.181.0/24
...
   domain master = yes
...
   wins support = yes
...
```

Other options (not mentioned above) were left at their default setting. I refer once more to the documentation files /usr/doc/samba−1.9.18p10/docs/ENCRYPTION.txt, /usr/doc/samba−1.9.18p10/docs/WinNT.txt and /usr/doc/samba−1.9.18p10/docs/Win95.txt. These documents either allow you to disable password encryption in NT4SP3+, or to add SMB password encryption to Linux: notice the usage of the mksmbpasswd.sh script and the smbpasswd command in the ENCRYPTION document. Major disadvantage of the mksmbpasswd.sh script is that a second run clears all encrypted password entries from an existing /etc/smbpasswd file!

The username mapping allows for well−known user names from the Windows−environment to be mapped to well−known user names in a UNIX environment. An smbusers file is included with the distribution.

We set the domain master option to yes, to make sure the Linux server acts as a primary domain controller on our subnet (since we don’t have a Windows NT Server running as primary domain controller). We also activate the WINS server to allow for hostname lookups (since no NT server is providing us this service on our wireless network).

That’s it for the options in the [global] section. Next, the file contains various share−sections. You probably want to comment out the entire [printers] section. In the [public] section, you should replace ’@staff’ by ’@users, @popusers’ on line ’write list’. Such a write list contains the users that may write to the published directory, but Linux user groups can be included as well: the group name should be prefixed with the ’@’ sign.

Here are a few more examples on additional sections that could be added:

```
[WebSite]
   comment = System web pages
   path = /home/httpd
   public = no
   read only = yes
   write list = @users

[Config]
   comment = Linux configuration
   path = /etc
   public = no
   read only = yes
   write list = root

[DNSDatabase]
   comment = BIND DNS database
   path = /var/named
   public = no
   read only = yes
   write list = root
```

Connect a Windows−based computer to your server (be it via network or via AX.25) and go check your network neighborhood quickly! ☺

SMB client support is not limited to Windows workstations: for Linux, check the commands `smbclient` and `smbmount` (man pages, etc.)!

The X Window system

Never say Windows to X−Windows! They're simply incomparable. X−Windows was, from the very beginning, designed for distributed computing. Take this literally: the individual windows on your screen (like the menu controls towards their parent windows) communicate with each other (internally) using... TCP/IP!

Next to the X Window system itself (with the X server), you need a window manager, that allows you to organize multiple windows on a 'desktop'. While early window managers did no more than doing this organization and providing some toolbars, more recent window managers (KDE and GNOME) provide full−fledged desktop environments (allowing you, for example, to place data−files on your desktop). Don't be mistaken by the free availability of these latter 2 environments: they are entirely based on CORBA, an industrial standard for exchanging objects (another better known object exchange mechanism is COM/DCOM, which is the base of OLE, ActiveX, etc). Just to let you know what you may expect to be happening... :−)

To pick up our story−line again: if you want to run X applications from the server at your workstation, first make sure that the X application (which will figure as an X client) has access to your local X server (since the X server is the part that controls the graphical hardware, the X server is running at your local workstation). From a local command shell, issue: `xhost on0baf` (to grant X applications running at on0baf.baf.be.ampr.org access to your local X server). Next, in your telnet at the server (e.g. on0baf), make sure the `DISPLAY` environment variable is set (you can check this using the '`set`'−command). It should look something like this:

```
DISPLAY=on1blu.baf.be.ampr.org:0.0
```

(if it’s not, make it so, by just entering this line at the command shell as it’s stated here). This settings makes sure that all X−applications you will be starting from that particular telnet session will contact the X−server at on1blu.baf.be.ampr.org. Now you can start X−applications from the command shell: when you do so, while you will see (a) window(s) appearing on your screen, you should realize that the application itself is running at the server! I strongly discourage using this feature at 1200 baud links!

XNTPD (Network time protocol)

XNTP should normally need no configuration. A word should be placed, however, on the system clock. In Linux, you have 2 different clocks: a hardware clock (on your motherboard) and a system clock (a piece of software in the kernel). Since Linux assumes that hardware clocks are not accurate at all, a system clock was created in the opinion that it would be far more accurate.

That’s possible, but you’ll need to do some damned hard calculations (and you will need more than 1 week to complete the procedure), for being able to provide a time adjustment in `/etc/adjtime` (which is actually a system clock drift correction).

Especially when a system is running, the accuracy of the hardware clock is fairly sufficient. That’s why we preferred copying the hardware clock into the system clock on an hourly basis. In the directory `/etc/cron.hourly`, create the file `SynchClock` (don’t forget the `chmod a+x SynchClock`!):

```
#!/bin/sh
/sbin/hwclock −−hctosys −−utc
```

Adding users

1. Run `adduser −u <user−ID> −g <group name> −s <shell> −c <comment> <login name>`.
   For the user ID, it’s generally good practice to add 500 to the as−signed IP address (fourth octet value), or assign an incremental value starting from 800 if no IP address was assigned.
   Group name should either be `users` (typically reserved for Sy−

sOps) or `popusers` (all other registered users).
The shell should be a valid shell (`/bin/bash` is the most popular one) or `/bin/ftponly` (if you want to avoid a former JNOS user suddenly ending up in your Linux shell).
Use the comment option to specify the full name of the user, for example. If spaces are present, enclose the comment itself between double quotes.
The login name typically is the (lowercase) call sign of the user.
Adduser is actually a script that, amongst other things, adds an entry to /etc/passwd and copies the contents of the directory `/etc/skel` to a newly created home directory `/home/<login name>`.

2. Run `passwd <login name>` to change the password of the user.

3. Run `/home/ftp/SynchUsers`.

4. Run a newly created `/home/MakeUserHome <login name>` script (don’t forget the `chmod a+x MakeUserHome` after creating the file!):

   ```
   #!/bin/sh
   chmod 0711 $1
   mkdir $1/public_html
   chown $1:popusers $1/public_html
   chmod 0711 $1/public_html
   cp −r ~ftp/bin $1
   cp −r ~ftp/lib $1
   cp −r ~ftp/etc $1
   ```

5. Add the IP address − call−sign pair to `/etc/ax25/static−arp` and run

   ```
   /sbin/arp −H ax25 −i ax0 −f /etc/ax25/static−arp
   ```

   to update the system’s ARP table (not necessary if your server is solely connected to an IP router, like XNET).

6. Add an AX.25 route, if necessary (typically for users living far away who have no TCP/IP servers or routers in their vicinity): add an AX.25 route command to script `/etc/ax25/static−axroute` (and execute it manually too), like:

   ```
   /usr/sbin/axparms −route add KISS on0kul−10 on0baf−1 on0lvn−6
   ```

   You could append an additional `−ipmode v` option to have a AX.25−level virtual circuit created when using this route.

7. If you assigned an IP address and a domain name to this user, add the necessary resource records for the new user to your zone and

reverse zone files (`/var/named/named.baf` and `/var/named/named.rbaf` in our examples: read the BIND−section) and run `/usr/sbin/ndc restart`.

Step 4 is only required for users that will be member of the `popusers` group. Due to the guestgroup−characteristic of `popusers` (with regard to FTP), they can only see from their home−directory downward. That’s why a similar system as for anonymous users must be set up. If it concerns a user from group `users`, you should still adjust permissions to allow access to his `public_html` directory (which you still need to create yourself).

Remote administration (linuxconf)

Although remote access is an integral part of Linux (through telnet, rlogin and rsh), we especially want to mention linuxconf. Linuxconf is a recent addition to the Linux distributions that allows for ’user−friendly’ configuration. Linuxconf actually has 3 faces:

- A command shell application, offering a tree−structured access path towards all Linux’s configuration data. Using the cursor keys, you can simply browse through the setup of your Linux system.
- An X−application (originally designed to run in a GNOME window manager), which provides a tree control as well. Using the mouse you have easy access to configuration settings.
- And, last but not least, a web−interface. Using the Internet browser you’re familiar with (NetScape, Internet Explorer, etc.), you can easily browse to the linuxconf web−server, which is running at your server − listening at port 98 (e.g. http://www.baf.be.ampr.org:98). After logging in, you have access to a hierarchical set of hyperlinks that bring you to all those configuration topics.

For being able to use this latter interface, you’ll need some specific configuration (do this while you still have physical access to your server):

1. Start linuxconf
2. Open ’Config’, ’Networking’, ’Misc’ and then ’Linuxconf net−work access’. Enable the ’Enable network access’ check box. Next, you will have to enter a network or a host from which you want web−access to your linuxconf. This is a first level of security checking when trying to access linuxconf: this (at least one re−

quired) setting determines from what (set of) computer(s) you want yourself allowed access to linuxconf (for example, if you want to restrict access to your PC at home, enter that PC’s IP address; if you want to restrict access to your local user access ports, you could enter something like 44.144.181.0 and 255.255.255.0 as netmask).

3. The second level of security comes from your user account details. Each time you want to access linuxconf, you will need to provide a login name and a password (as it was configured for your generic Linux user account). In order for your linuxconf−access to be granted, open ’Config’, ’Users accounts’, ’Normal’, ’User accounts’ and locate your own user account. Open it and move to the ’privileges’ tab. In the ’General system control’ tab from there, you should at least grant yourself ’May use linuxconf’, ’May activate config changes’ and probably ’SuperUser equivalence’ (although I wonder what the latter actually does: I haven’t noticed any particular rights being granted to me). In the ’User accounts’ tab you will probably want to grant yourself ’POP accounts manager’.

Interfacing to the current Packet Radio network

In our opinion, legacy has always been a critical requirement for the acceptance of this "new" TCP/IP technology. While we already showed some examples to allow Packet Radio users to participate in the IRC network and others, we will present here the most important exchangeability areas: BBS systems, DX clusters and nodes.

Interfacing to BBSs: messages and bulletins

The ability to exchange E−mails and news articles with BBS systems is probably the ultimate condition for TCP/IP acceptance, but it's also the most demanding one. Currently, we have 2 options:

- Using a KA9Q NOS−compatible system. Until recently, we have been using the JNOS package to do the job. We compiled a highly specialized version, which only included a POP3, SMTP and NNTP server and BBS forwarding, of course. This JNOS was started on the Linux box that runs the native Linux services. On this single computer, a pseudo serial port was created, allowing the JNOS's TCP/IP stack to communicate with Linux's native TCP/IP stack using SLIP. Next, using the net2kiss software, we tapped Linux's AX.25 interface to allow JNOS to perform BBS forwarding. While we succeeded in making this sytem operational, we prefer not to show the details here, as we experienced this setup lacks robustness and requires long E−mail addresses.
- We find the combination of the mailgw package and a separate BBS forwarding package to be a more elegant solution. The BBS forwarding package can either be a full−featured BBS system like FBB, or a forwarding−only package like lnxforward. We will now present the setup of mailgw and lnxforward.

The mailgw package

The mailgw package from Heikki Hannikainen (oh7lzb) is, amongst other things, a Mail Transport Agent (MTA) that can be called by a Mail Router like sendmail. When the mailgw software receives an SMTP−compliant E−mail message (due to routing within sendmail), it transforms that message into an FBB−compliant format, which can be forwarded to a BBS. The process is also reversible. But mailgw also accepts newsfeeds from INND, for example, allowing for the exchange of BBS bulletins.

While configuring this package, you will notice that some files − which had already been configured earlier − need additional changes.

First, obtain the mailgw RPM package (which can be downloaded from the project website): mailgw−0.3.1−4.i386.rpm. Once down−loaded, install it using: `rpm −i  mailgw−0.3.1−4.i386.rpm`.

Now, move to the newly created directory `/etc/mailgw` and edit the following files (the map files use the same rewrite mechanism as the NOS packages):

- `mailgw.conf`: edit the `HostName`, `Callsign`, `HRoute` and `Qth` fields. You could reduce the `LogLevel` once your system ap−pears to function without problems. Other fields should normally not be touched. This will configure the mailgw software to communicate with the sendmail and INND packages.
- `email−dst.map`: this file configures the translation of the E−mail destination address into a destination BBS HADDRESS. Since mailgw will receive BBS HADDRESSes, no specific translation is required and no changes should be required to this file.
- `email−src.map`: this file configures the translation of the E−mail source address (the address of the user sending the E−mail) into a source BBS HADDRESS (the address that will be used when the destination uses the reply command at his BBS). You should, at least, replace the HADDRESS with your system's. The first line in the sample configuration file will transform addresses like on1b−lu@on0baf.baf.be.ampr.org into on1blu@on0baf.lm.bel.eu. The lat−ter can be used to reply to within BBS systems.
- `pbbs−dst.map`: defines the translation from a destination address in a BBS−origined mail to an E−mail address. At least edit the host−names to match your system.
- `pbbs−src.map`: as you might have expected by now, this file de−fines the translation of a source address in a BBS−origined mail to an E−mail address, which can be replied to using E−mail software. No changes should be required here.
- `bulletin.map`: here you can define what BBS bulletin areas should go in what newsgroups.

- `newsgroup.map`: the reverse file for `bulletin.map`, defining what BBS areas news posts should be forwarded to.

This completes the gateway configuration itself. Before configuring sendmail and INND to communicate with mailgw, one advise for your users might be in order. Since many BBS areas can be gathered into one newsgroup, but one newsgroup can only be mapped onto just one BBS area, a public reply to a bulletin could end up in an other BBS area than the bulletin was received from. The bulletin author might therefor never read the reply. You could propose your users to send a copy (CC) of their reply to the author personally, when they notice the news article arrived through BBS forwarding.

Let's handle sendmail now. We preferred the usage of a fake top−level domain '.bbs' to indicate BBS−destined messages. What it ultimately means for the users is that they should add '.bbs' to the destination address in their E−mail software. For example, if you want to send an E−mail to ON6DP, who's a BBS user, enter E−mail address on6dp@on0rat.lg.bel.eu.bbs, or even on6dp@on0rat.bbs. The authors proposed to add a fake top−level domain for each continent code (.eu in the sample above). This would require the users to enter the full hierarchical address, however.

You'll need to edit `/etc/sendmail.cf` once more.

- First, we'll add the fake top−level domain. Locate the line holding '`CP.`' and insert a new line with 'CPbbs' immediately below.
- Now, we must add the handling of mails to addresses that hold such a fake top−level domain. Move to Ruleset 0 (S0), Parse1 (SParse1) to be more precisely, and locate the area stating '`resolve fake top level domains by forwarding to other hosts`'. Insert a new line immediately below that comment line, as follows:
  ```
  R$* < @ $* . bbs . >    $#pbbs $@ $2 $: $1 < @ $2 >
  ```
  This will strip the '.bbs' from the destination address and pass the mail to the pbbs MTA.
- Finally, we need to configure the pbbs MTA. Move further to the '`Local and Program Mailer specification`' area, and after the definitions for MTAs local and prog, add another one, like this:
  ```
  Mpbbs,    P=/usr/sbin/smtp_rcv, F=IFDMn, S=10/30, R=40,
            T=X−Unix,
            A=smtp_rcv $f $h $u
  ```

This completes sendmail manipulations. If you decide to rely on another SMTP server to do the gateway stuff, you should rewrite to that SMTP server instead of handing the mail over to the pbbs MTA. You shouldn't strip the .bbs part either. Finally, you should coordinate with the gateway's administrator that address translation to your server is supported too from the `/etc/mailgw/email−src.map` and `/etc/mailgw/pbbs−dst.map` files. Alternatively, you could try to simply configure that gateway as your fallback MX−host (read the sendmail part about fallback MX hosts) and forget configuring fake top−level domains yourself. This hasn't been tested by us yet, however. Restart the mail server to activate the changes, by issuing:

```
killall −HUP sendmail
```

And now we only need to add a newsfeed to exchange news articles and BBS bulletins bi−directionally. Edit `/etc/news/newsfeeds` once more, to add the following:

```
fbb\
          :*\
          :Tc,Ap,G2,Wf\
          :/usr/sbin/nntp_rcv
```

You also need to restart INND to activate this new newsfeed:

```
ctlinnd reload all Added mailgw newsfeed
```

We're through.

The lnxforward package

The lnxforward package from Steve Fraser (VK5ASF) is FBB−compliant forwarding−only software.

Again, download the package (lnxforward−1.25−4.i386.rpm) from the project website and install it using

```
rpm −i lnxforward−1.25−4.i386.rpm
```

First, edit file `/etc/fbbforward/fbbforward.conf`. Normally, you should only change the `Callsign` field (don't forget the SSID, if you plan one!).

Next, within `/etc/fbbforward`, you should at least create control files for all BBSs you will be forwarding to. For example, issue:

```
touch on0rat.ctl
```

These control files allow for filtering of mails, based on suspicious terms within that mail. Read the (sparse) documentation that comes with the package for the right syntax.

Now, `cd` into `/usr/local/ffbforward`. Edit the file `lnxforward.equ` to configure the BBSs you will be forwarding to. `BBS_LIST` is a white−space separated list of BBS systems (no SSIDs), `COPY_LIST` is a colon (':') separated list that defines to which BBSs messages will be forwarded to (`fbb_copy` parameter syntax is used; last entry should contain keyword '`delete`'). The `CALL_LIST` is a colon (':') separated list that defines all outgoing connections (`fbbforward` parameter syntax). `COMP_LIST` is the compression algorithm (`lzhuf` or `gzip`) used when forwarding to the listed BBS, it's also a colon (':') separated list. Finally, set `MYCALL` to the same call you entered in `/etc/fbbforward/fbbforward.conf` and probably comment all `SAT_`−entries. Leave the rest untouched.

Once you completed editing `/usr/local/fbbforward/lnxforward.equ`, run the `lnxforward.configure` script to create the appropriate directory structure. If you add BBSs afterwards, edit `/usr/local/fbbforward/lnxforward.equ` and create the required directories manually (typically `/var/spool/fbbforward/in/<BBS>`, `/var/spool/fbbforward/in/<BBS>/hold` and `/var/spool/fbbforward/out/<BBS>`).

Upon completing the steps above, outgoing forwarding is scheduled at hourly basis automatically. To support incoming forwarding sessions, you need to make one final addition to `/etc/ax25/ax25d.conf`[11]. Here's an example:

```
[ON0BAF−8 via KISS]
NOCALL  * * * * * *  L
default * * * * * *  − root /usr/local/fbbforward/fbbforward fbbforward STDIO %u
```

You should also double check whether the files `/etc/ax25/axports` and `/etc/sysconfig/network−scripts/ifcfg−ax0` specify a paclen respectively a MTU of 256. If not, you'll notice several of the messages that arrive will be corrupted, etc. The lnxforward package interrogates the AX.25 interface for its MTU/paclen. The configured value is used for outgoing traffic only, however. And since AX.25 doesn't negotiate upon link establishment, the MTU/paclen issued by the remote station can not be determined, resulting in a potential mismatch or, eventually, bytes being missed.

Your gateway should be functional now.

[11] *Especially check this file if you installed the lnxforward using one of my RPMs: the sample included in one of those RPMs released contained a minor error: the separate 'fbbforward' was forgotten. This results in error messages appearing on your screen, each time a connect to the indicated call is initiated, starting with 'STDIO', followed by the allowed command line parameters for the fbbforward program.*

DX−cluster

This chapter handles the installation of version 5.0 of the CLX DX−cluster software. This chapter is the first that has been reworked entirely since the creation of this manual: previous editions handled version 4.06. We switched to version 5.0 as soon as it appeared, since this version uses the libraries that are distributed along with Red Hat Linux 5.2 (which is the glibc2 library).

First we'll need to remove the installed database server (postgreSQL), since we really need a more recent version. Here are the commands to perform:

```
rpm --erase postgresql−devel
rpm --erase postgresql−data
rpm --erase postgresql
rpm --erase postgresql−clients
```

Now, obtain a more recent version of postgreSQL (6.5.3): it's preferable you download it through the link on my website since this assures you obtain the most recent version that runs on RHL 5.2. You could also try obtaining a more recent companion CD.

We need to install at least (from the directory where you have a copy of the RPM−files):

```
rpm −i postgresql−6.5.3−3.i386.rpm
rpm −i postgresql−server−6.5.3−3.i386.rpm
```

After installing these packages, you need to define a password for the newly added postgres user account, using the command `passwd postgres`.

Now log in as `postgres` on another virtual console and start editing the profile: `joe .bash_profile` (don't forget the dot before the file−name!). Add the following lines:

```
PGLIB=/usr/lib/pgsql
PGDATA=/var/lib/pgsql
export PGLIB PGDATA
```

Log out, log back in as postgres and issue:

```
initdb
```

Now, we'll install the CLX package itself. You can obtain it from the CLX home−page (http://www.lurpac.lancs.ac.uk/~clx) or through my website. From a root−login, issue:

```
adduser −u101 −gusers −d/usr/local/clx −c"CLX account" clx_us
passwd clx_us
cd /
tar xvzf <path where you stored the packages>/clx_500.tgz
tar xvzf <path where you stored the packages>/upd_501.tgz
cd /usr/lib
```

```
ln −s libstdc++.so.2.8.0 libstdc++−libc6.1−1.so.2
/etc/rc.d/init.d/postgresql start
```

If the database starts fine, you can enable its automatic launch: start `linuxconf` once more, go to ’`Control`’, ’`Control panel`’, ’`Control service activity`’ and set `postgresql` to enable.

Logged in as root, edit the file `/etc/ld.so.conf` and add the following line:

```
/usr/local/clx/lib
```

Next, issue the command `ldconfig −v`

Now, we’ll need our postgres−login one last time to issue:

```
createuser
```

The user name should be clx_us, then just press enter to accept the UNIX−password and choose ’y’ on both presented options. You may quit the postgres account now, since you’ll no longer need it.

Now log in as clx_us and issue:

```
clx_db
clx_idx
touch config/passwd
```

From the root login, edit the following files:

1. `/etc/ax25/ax25d.conf`: add the following entry (use the appropriate call−sign!):
   ```
   [ON0BAF−7 via KISS]
   NOCALL  * * * * * *  L
   default * * * * * *  − clx_us /usr/local/clx/bin/net_usr net_usr −x %s
   ```
   and issue:
   ```
   killall −HUP ax25d
   ```
2. `/etc/services`: add the following entry:
   ```
   clx    41112/tcp # CLX DX−cluster access
   ```
3. `/etc/inetd.conf`: add the following entry:
   ```
   clx stream tcp nowait root /usr/local/clx/bin/clxd clx
   ```
   and issue:
   ```
   killall −HUP inetd
   ```

Strictly speaking, CLX is ready for work now. You will still need to obtain a call−sign key, however. From your clx_us login, issue clx −u > proof.txt. Mail the generated proof.txt file, accompanied by the call−sign of your DX−server, to the authors of CLX, asking for a key. This key must be entered in `/usr/local/clx/config/clx_par`.

When all that’s in place, issue the following command (from a login as root) to have CLX started upon booting the PC (manually starting and stopping the CLX software from the `clx_us` login prompt is done using, respectively, `clx -u` and `clx -s`):

```
ln -s /usr/local/clx/tools/startup /etc/rc.d/rc3.d/S99clx
```

Typically, the file `/usr/local/clx/config/cluster_par` still needs editing to link up with other DX−cluster servers. Don’t forget to consult the manual (in `/usr/local/clx/doc/sysop`)! As an indication, here’s what we use to make an active link to ON0EUL and a passive backup link to ON0NOL (I want to mention that you might encounter problems when trying to use the kernel AX.25 for outgoing connections or when you start shuffling the order of the parameters, playing with whitespaces, character case, comments, etc):

```
# ON0EUL-15
SECTION:  on0eul-15
conn_call:on0baf-7
conn_int: x
conn_path:on0eul.sh
conn_act: active
conn_type:non-clx
conn_prot:pc
conn_ping:yes
mail_fwd: no
conn_lock:no
# ON0NOL-9
SECTION:  on0nol-9
conn_call:on0baf-7
conn_int: x
conn_path:on0nol.sh
conn_act: passive
conn_type:clx
conn_prot:pc
conn_ping:yes
mail_fwd: no
conn_lock:no
```

Here’s a sample shell script `/usr/local/clx/exec/connect/on0eul.sh`:

```
#!/bin/sh
read $something
stty -echo raw
exec /usr/sbin/ax25_call KISS on0baf-7 on0eul-15 on0baf
```

In the sysop manual mentioned earlier, you can find more elegant, richer featured and parameterized alternatives for connection scripts.

Linking with node−software

While Linux supports operation with the NETROM and ROSE protocols, we don’t have any experience on using them (and we don’t plan on obtaining experience either). Of course we dream about a pure network of IP routers in some foreseeable future (which could be a very near future thanks to node software like XNET).

It’s quite possible this chapter will grow in future revisions of this document, but that’s not certain at all.

FlexNet

At the time of this writing, we use FlexNet to participate in the amateur radio packet network. The Linux server is connected to one of the node’s ports using a wired KISS link. Major advantage of this approach is that FlexNet takes care of selecting the right port to access the various users when it is the FlexNet node that handles all user access ports. Servers are typically ’linked’ through AX.25 digipeating routes, always passing the node itself. If the node is running fine within the network, the local node is even the only digipeater required to be specified in those AX.25 routes. There are no additional network issues for this kind of configuration.

XNET

XNET is a relatively new product that encompasses the NETROM and FLEXNET protocols, but also includes a true IP router. At the time of this writing, the author doesn’t have many experience with this software, but we consider extending our knowledge on this matter.

If you activate the IP router, then your XNET box should be assigned its own IP address. You typically might want to assign the last available one, for the region of baf.be.ampr.org that would be 44.144.181.254. Your users should then no longer configure your server as the interface gateway, but the XNET box instead.

The server should be configured with the same routing principle as the users: a host route should be defined to the XNET box and all other 44.xxx.xxx.xxx addresses should be routed through the XNET box. Notice that you no longer require to maintain ARP mappings, AX25 routes and any other IP routes than the ones just mentioned on your server: they must all be maintained on the XNET box now. If you don’t plan any AX.25 access to your server (not even forwarding as part of a mail gateway!), you could consider wiring the server to the XNET box using SLIP.

Anyhow, in this kind of configuration, your Linux server could become separated from the users subnet: please read the configuration part of the sendmail daemon again with regard to relaying.

What can I present to my users?

(x)NOS

The first approach ever taken to bring TCP/IP to our radio frequencies was the one achieved by KA9Q. The approach was to create a 'standardized' mechanism for plugging in new TNC−drivers and then to develop an entire TCP/IP protocol stack on top of it. Nos was created when TCP/IP protocol stack implementations still were very rare. Afterwards, many variants have been created, JNOS and TNOS being the most popular (and the only ones left that are somehow supported).

We're convinced that this software will only be used by people who are restricted to MS−DOS: installation and configuration is very cumbersome (even worse than Linux :−)). We do not intend to include a NOS−tutorial here, so we advice to obtain and read the accompanying manuals, as they are generally sufficient to get the show running.

MCB−152

Another approach to enable TCP/IP software to be used on our frequencies was to work on the firmware. We noticed that the lack of (user−friendly) packet software was partly due to the fact that our TNCs are specialized hardware, amateur radio being the only target market. This gap was introduced when it was decided that TNC−firmware should not look like telephony modem firmware.

First, we created new hardware because we feared current TNC architectures would never allow high speeds. That's what the MCB−152 hardware is all about. Next, firmware was developed to let the MCB−152 behave as a well−known KISS TNC (so existing Packet Radio Software can still be used). As soon as this was operational, the so−called SLIP firmware was developed. When downloaded into the MCB−152, it will initially accept the (for telephony modems) well−known 'AT' Hayes−compatible commands. Once the user has dialed−in the MCB−152 becomes a protocol translator (at the link layer): SLIP−frames that are received from the computer have their SLIP encoding removed, leaving the bare IP−frame, which is then encapsulated into an AX.25 packet before being transmitted. The reverse works as well: packets captured from the frequency have their AX.25 header removed, the obtained IP−frame is then SLIP−encoded and sent to the computer.

Since this approach actually simulates an ordinary telephony modem, no drivers need be installed and so there are no OS support problems. This reduces specific installation tasks to an absolute minimum: only the firmware needs to be downloaded. We preferred firmware down−loads to EPROMs, since updates could simply be distributed using E−mail, web−servers, etc...

The firmware download can be as simple as creating a batch file like (provided the `slip.hex` firmware−file is in the same directory as the batch file):

```
@echo off
mode com2 baud=19200 data=8 stop=1 parity=N
copy slip.hex com2:
```

A shortcut to this batch−file could be added in a startup−directory, so the firmware will be downloaded when the computer is started (if your MCB−152 is powered on when the computer starts).

For the remaining configuration, one only needs to set up a dial−up session, just as he would for an ordinary Internet dial−up. Use a stan−dard 19200bps modem as modem device and enter the link address of your TCP/IP server as telephone number (like on0baf.10: don’t use a dash (’−’) for the SSID!). You can also add VIAs: distinguish them using the ’@’−sign.

Detailed instructions can be found on the accompanying CD, and probably from the project web sites:
http://www.caseconsole.com/mcb152
http://gallery.uunet.be/Joachim.Elen
http://www.qsl.net/on1blu
http://home.worldonline.be/~vda10786

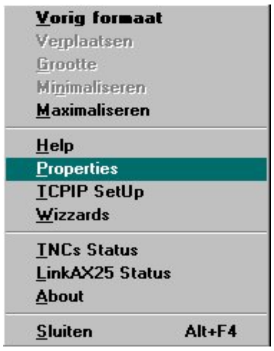
SV2AGW

So NOS was a software development, MCB−152 was a firmware de−velopment (although it also concerns new hardware), SV2AGW took yet another approach: driver development. A network driver for KISS−TNCs was developed on Windows ’95/’98 (c.f. the driver for your Ethernet card: Windows will actually ’think’ your TNC is on a LAN).

Pasquale (ON1DLB) provided us some hints for installing and config−uring[12] the software.

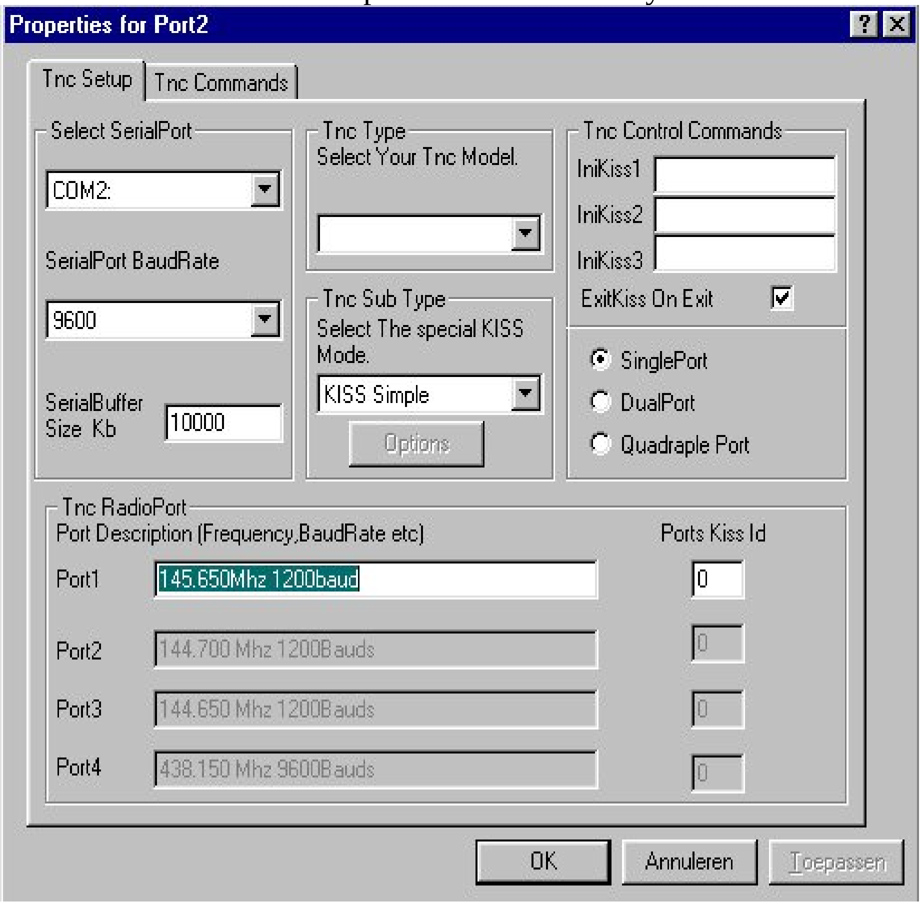
[12] It’s not actually the LAN−alternative but a modem plug−in for dial−up.

Unpack the software zip−file and create a shortcut for AGWPE.EXE. Launch the application and when it appears on the task−bar, right click on it, so it shows the following pop−up menu:



Select ’Properties’ and click on ’New Port’ in the dialog that appears.

Now select the COM port and modem you wish to use:



Properties for Port2

Tnc Setup | Tnc Commands

Select SerialPort: COM2:

SerialPort BaudRate: 9600

SerialBuffer Size Kb: 10000

Tnc Type
Select Your Tnc Model.

Tnc Sub Type
Select The special KISS Mode.: KISS Simple

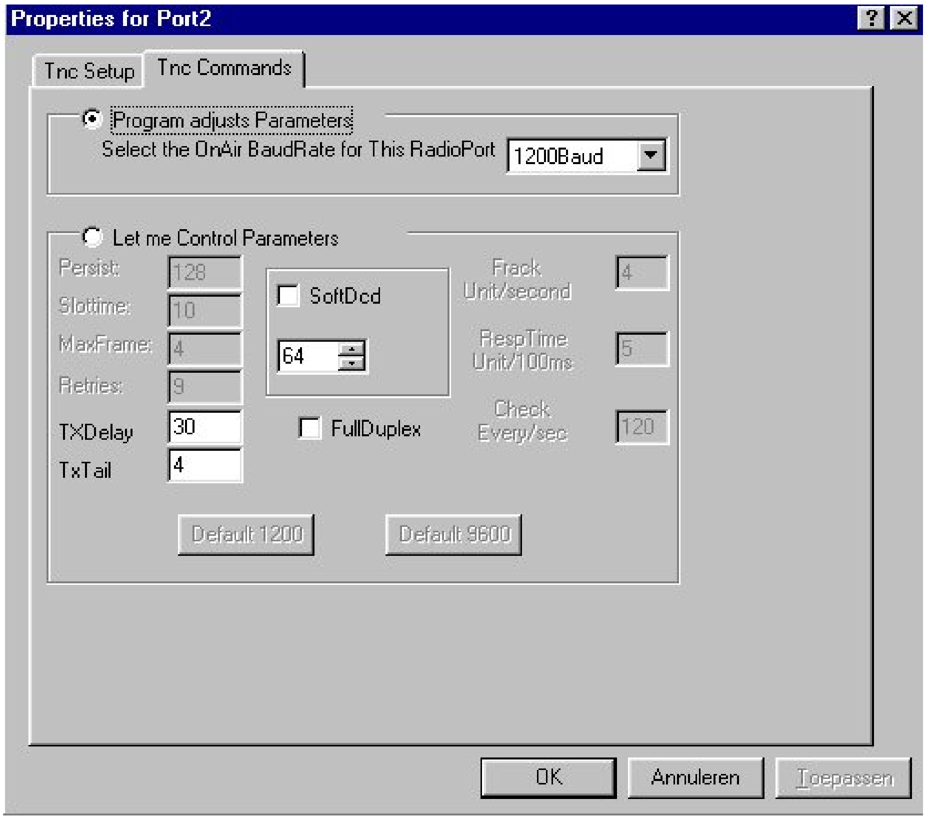
Tnc Control Commands
IniKiss1:
IniKiss2:
IniKiss3:
ExitKiss On Exit ☑

● SinglePort
○ DualPort
○ Quadraple Port

Tnc RadioPort

Port Description (Frequency,BaudRate etc)		Ports Kiss Id
Port1	145.650Mhz 1200baud	0
Port2	144.700 Mhz 1200Bauds	0
Port3	144.650 Mhz 1200Bauds	0
Port4	438.150 Mhz 9600Bauds	0

Next, you can decide for automatic AX.25 parameter configuration or to configure them yourself:



● Program adjusts Parameters

Select the OnAir BaudRate for This RadioPort: 1200Baud

○ Let me Control Parameters

Persist: 128

Slottime: 10

MaxFrame: 4

Retries: 9

TXDelay: 30

TxTail: 4

□ SoftDcd: 64

□ FullDuplex

Frack Unit/second: 4

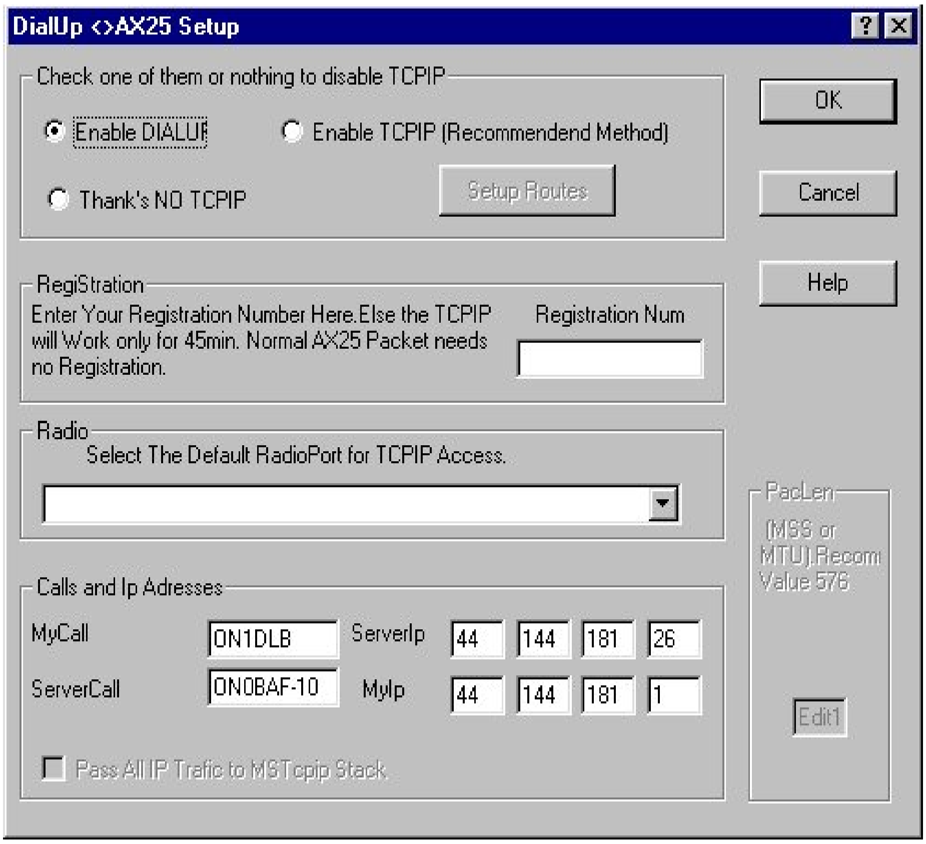
RespTime Unit/100ms: 5

Check Every/sec: 120

Upon completion, close AGWPE from the pop−up menu (you’ll need to restart AGWPE every time you changed a configuration).

Now, restart the AGWPE. The software should now be able to control the TNC. You could check this with the AGWMONITOR application, which monitors all traffic on the frequency.

Now, we’ll configure AGWPE for TCP/IP dial−up (SLIP must be available and you’ll also need WinSock 2.0, which is contained in WS2SETUP.EXE): select TCPIP SetUp from the pop−up menu.



DialUp <>AX25 Setup

Check one of them or nothing to disable TCPIP

● Enable DIALUI ○ Enable TCPIP (Recommendend Method)

○ Thank's NO TCPIP

RegiStration

Enter Your Registration Number Here.Else the TCPIP will Work only for 45min. Normal AX25 Packet needs no Registration.

Registration Num

Radio

Select The Default RadioPort for TCPIP Access.

Calls and Ip Adresses

MyCall	ON1DLB	ServerIp	44	144	181	26
ServerCall	ON0BAF-10	MyIp	44	144	181	1

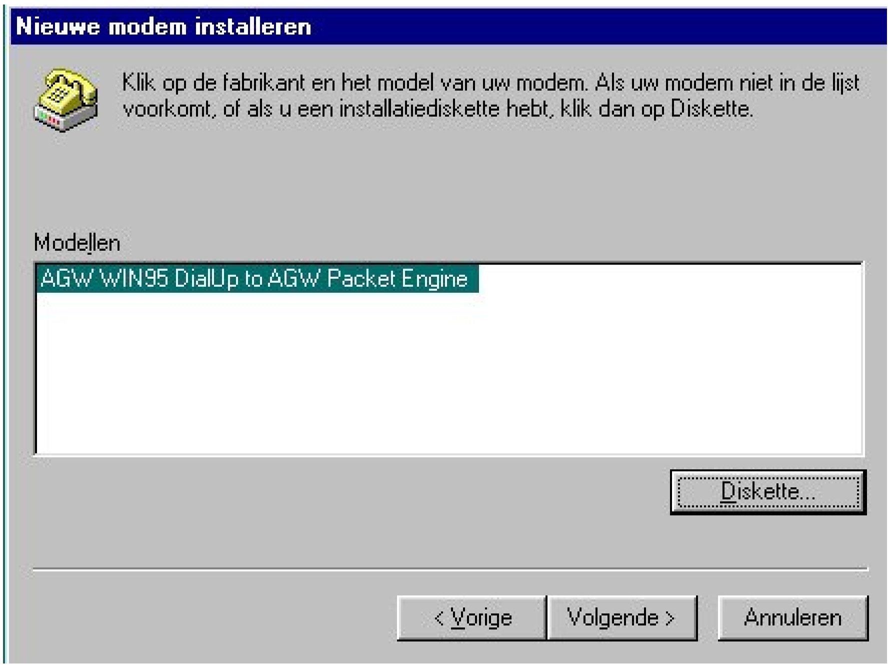
□ Pass All IP Trafic to MSTcpip Stack

PacLen

(MSS or MTU).Recomi Value 576

Don’t mind the ’ServerIp’ and the ’MyIp’ labels, since only the labels themselves are misplaced.

Now follow the standard procedure for adding an Internet dial−up, but when choosing the modem, browse yourself to the location where the files AGWDIAL.INF and AGWDIALUP.SYS reside. This should en−able you to add the following modem:



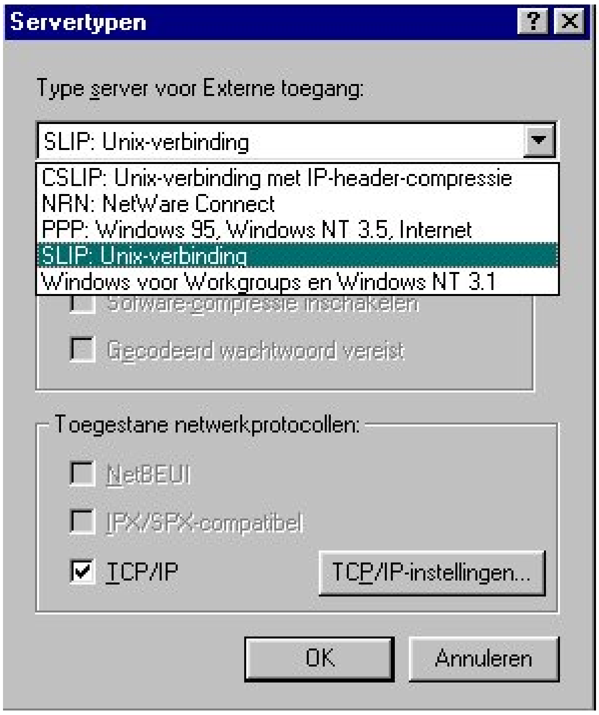
Nieuwe modem installeren

Klik op de fabrikant en het model van uw modem. Als uw modem niet in de lijst voorkomt, of als u een installatiediskette hebt, klik dan op Diskette.

Modellen

AGW WIN95 DialUp to AGW Packet Engine

This time, telephone number is the IP address of your TCP/IP server. As for the MCB−152, make sure you use the SLIP−protocol:



Servertypen

Type server voor Externe toegang:

SLIP: Unix-verbinding

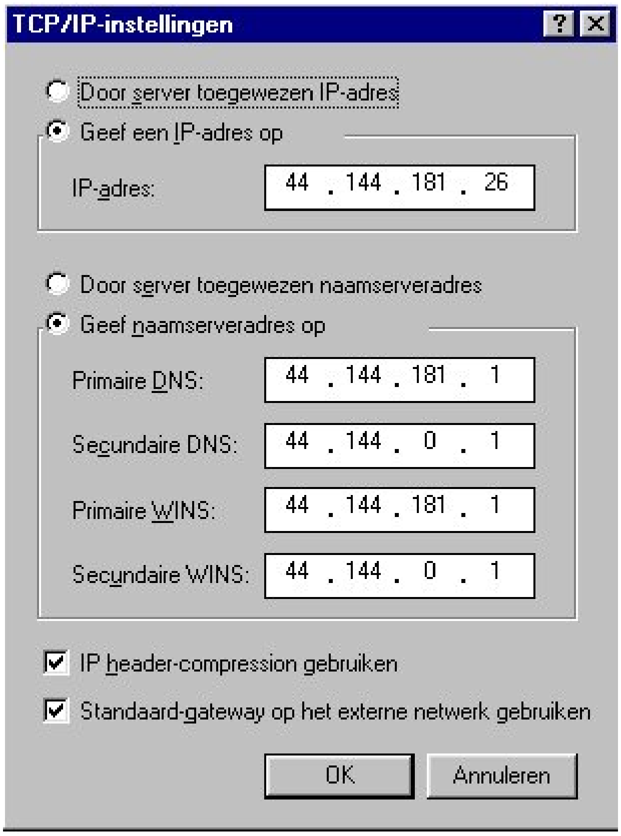
CSLIP: Unix-verbinding met IP-header-compressie
NRN: NetWare Connect
PPP: Windows 95, Windows NT 3.5, Internet
SLIP: Unix-verbinding
Windows voor Workgroups en Windows NT 3.1

□ Software-compressie inschakelen
□ Gecodeerd wachtwoord vereist

Toegestane netwerkprotocollen:

□ NetBEUI
□ IPX/SPX-compatibel
☑ TCP/IP

Still analogue to the MCB−152, configure the following TCP/IP settings (although it's best to disable IP header compression and default gateway for now):



TCP/IP-instellingen

○ Door server toegewezen IP-adres

● Geef een IP-adres op

IP-adres: 44 . 144 . 181 . 26

○ Door server toegewezen naamserveradres

● Geef naamserveradres op

Primaire DNS:	44 . 144 . 181 . 1
Secundaire DNS:	44 . 144 . 0 . 1
Primaire WINS:	44 . 144 . 181 . 1
Secundaire WINS:	44 . 144 . 0 . 1

☑ IP header-compression gebruiken

☑ Standaard-gateway op het externe netwerk gebruiken

To avoid an annoying dialog to appear before the actual dial−up, configure a script file, which contains the following lines:

```
proc main

endproc
```

Linux

Linux is not just another approach, it's simply another environment. It can be combined with the MCB−152 though (both figuring as a network device when KISS firmware is running, as well as an Internet dial−up when SLIP firmware is running).

The advantage of Linux is that AX.25 support was built into the kernel from nearly the beginning. So Linux really knows AX.25.

From the seminar, many of you might wonder why a user would ever decide to use Linux. During the seminar, we mainly worked in a text−mode command shell (remembering us to the MS−DOS era), some of you were able to start X Windows and probably were not quite impressed. Linux doesn’t stop there, however! Especially since Red Hat Linux 6.0 desktop environments like KDE and GNOME are fully integrated and when you read the architecture details of these environments (especially GNOME), you’ll notice they’re actually quite competitive to Windows 2000!

KISS−TNCs (including MCB−152 with KISS firmware)

If your users wants to use KISS (allowing him to do classical Packet Radio in parallel to TCP/IP), present him the chapters 4.Customizing the kernel and 5.Configuring AX.25. The change in the source code with regard to ICMP redirect would not be necessary, though.

For the file `ifcfg−ax0` (page 33), they should make sure the ’ROUTER’ parameter is set to their local router. When using ’transparant FlexNet access’ (meaning the user can reach the TCP/IP server by its call−sign, without needing to specify a via for the node), the DEFAXROUTE isn’t necessary.

MCB−152 with SLIP−firmware

If the user uses an MCB−152 with SLIP−firmware, it’s not even necessary to do a kernel−rebuild. We’ll need some minor manual adjustments, however, since the tools for setting up the SLIP−interface are not quite smart, unfortunately.

Use the tools you’re familiar with anyway: set the appropriate device settings, the user’s IP address (as local) and the server’s (or the router’s) IP address (as remote). This should generate 3 files in `/etc/sysconfig/network−scripts`: `chat−sl0`, `dip−sl0` and `ifcfg−sl0`. The only file we care about is `dip−sl0`, make sure it looks like this − NOT including the comment lines shown below that begin with # − (and never use the tool again to configure this interface, since all changes in `dip−sl0` will get lost):

```
...
main:
...
```

```
  get $mtu 215
  send AT\r
  reset
  sleep 1
  send AT&L41CON1BLU\r
# This is the init−string, consult the MCB−152 manuals!
  wait OK 10
  if $errlvl != 0 then goto error
  send ATDTon0baf.10\r
# This is the AX.25 address of your local TCP/IP server
  wait CONNECT 30
  if $errlvl != 0 then goto error
  mode SLIP
...
```

Users that have Red Hat Linux 6.1 installed, could add this ’finishing touch’: in the file /etc/rc.d/rc.sysinit, locate the lines:

```
...
# Initialize the serial ports.
if [−f /etc/rc.d/rc.serial ]; then
     /etc/rc.d/rc.serial
fi
...
```

and replace the rc.serial invocation (between the ’if’ and ’fi’ lines) with:

```
     action "Initializing serial devices" /etc/rc.d/rc.serial
```

Remaining common configuration

Whatever option they choose, they must configure their domain server, by editing resolv.conf to address their local domain zones and domain server; an example for users in Sint−Truiden would be:

```
search baf.be.ampr.org be.ampr.org ampr.org
nameserver 44.144.181.1
nameserver 44.144.0.1
```

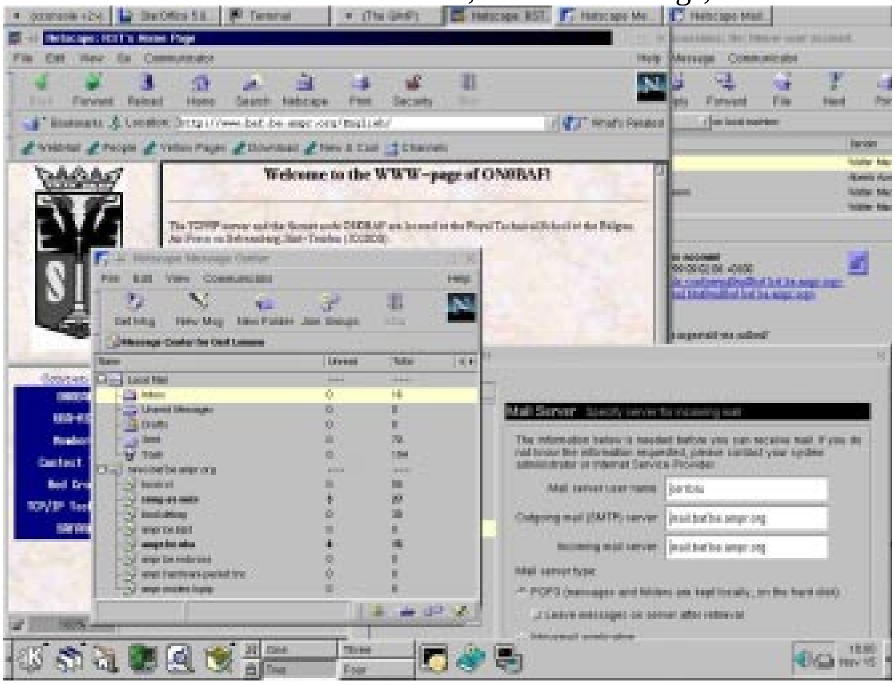
If the user prefers using an MCB−152 and he wants the network to boot when the computer starts up, he should create the /etc/rc.d/rc.serial file (page 27) and include the command for downloading the firmware.

The services that are available to your users

Having reached this point, we’ve spent an enormous amount of effort to introduce TCP/IP on Packet Radio. You probably expect a huge gain in features. For the server, consider that it is providing ALL popular services currently known at the Internet. For the users (except the NOS−alternatives), we’ve made sure that they can use the Internet−

software they’re accustomed to. Let’s have a look at some:

- Netscape is a rich−featured Internet application, which is available on practically all current operating systems (Windows, OS/2, Linux, etc.). With rich−featured we mean it provides a browser, a mail client and a news client, as well as FTP−features for downloading files, etc. Look at the URL in the browser, the mail−settings, etc.



Welcome to the WWW−page of ON0BAF!

Mail Server

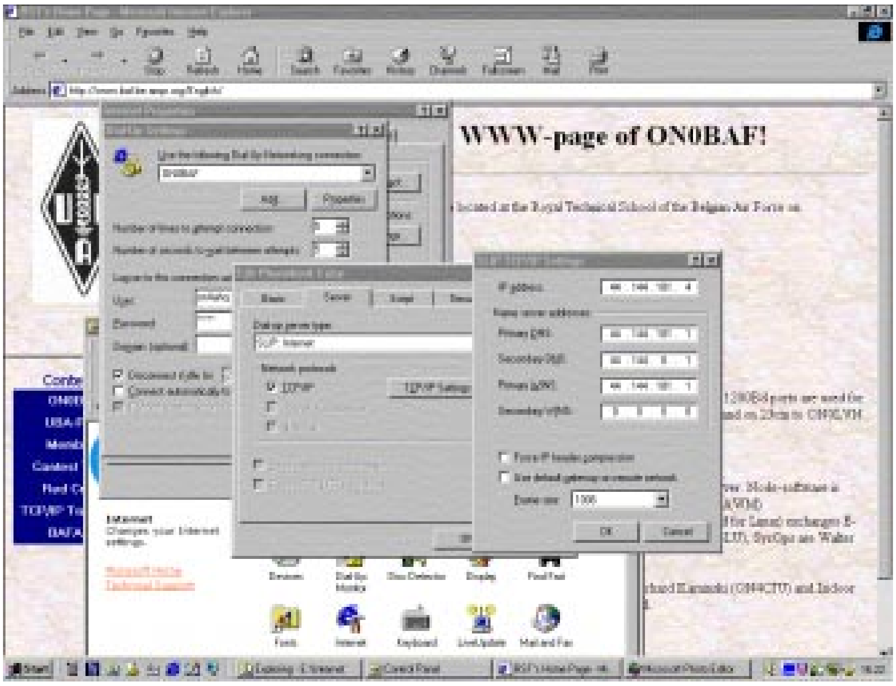
Mail server user name

Outgoing mail (SMTP) server

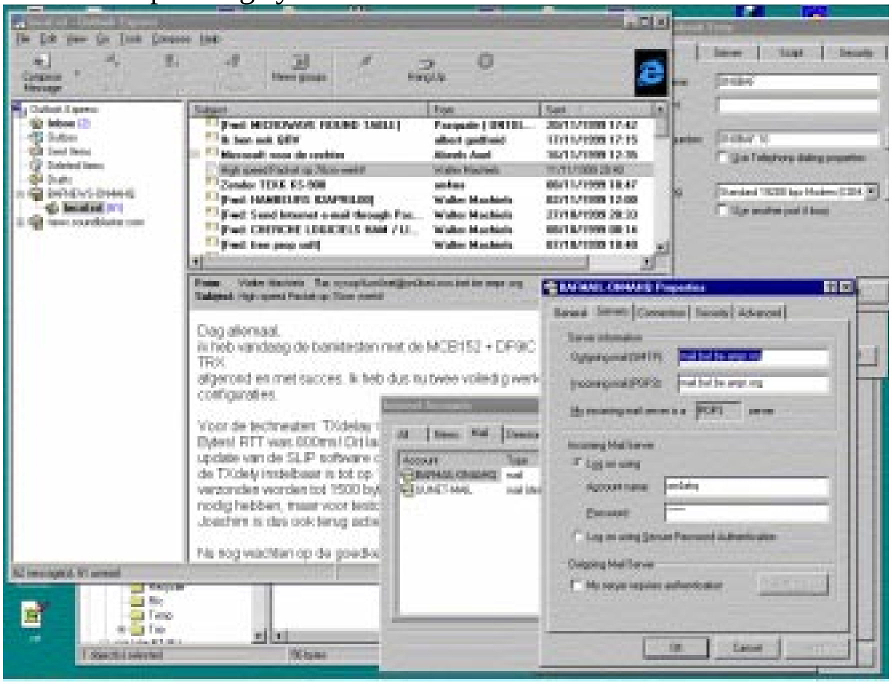
Incoming mail server

Mail server type

- Internet Explorer is the Internet browser, which is distributed as part of the Windows operating systems.



- Outlook and Outlook Express are mail/news clients. The first is part of the Office software package, the latter is distributed as part of the Windows operating systems



Dag allemaal.
Ik heb vandaag de baristeden met de MCB152 + DF9IC TRX
afgerond en met succes. Ik heb dus nu twee volledig werk
configuraties.

- Microsoft chat & netmeeting are conference applications, typically using IRC services.
- Windows network neighborhood provides file and print services across a network.
- ...

Concluding thoughts

We sincerely hope this text will help many people in launching new TCP/IP servers. We'll have work on connecting all these systems, but that's all part of the learning process.

For additional help, we plan on publishing all our configuration files through our web sites. On our wireless server, they will provide real−time views, from the Internet you will only be able to see configuration files that are updated on some regular basis (hopefully).

We also plan on moving to newer versions of Linux. We hope we can support this by providing both a 'differences' document based on the document you're currently reading and a fully updated document for people that enter the movement and want to start in a more recent version right away. We do not plan on supporting every possible Linux version and/or distribution. We want to install rather stable Linux configurations and it will therefor be impossible to try each upgrade and support it.

Some of you might wonder why we choose Linux. Many of you think Windows environments are more user−friendly and wouldn't take that much work. I insist commenting on that:

- It's currently impossible. The SLIP−trick can't be used on servers: SLIP is only usable for point−to−point connections and can not be used to serve many stations on a multiple access network. So, we need the KISS firmware, which is not known or supported by Windows environments. Sorry, correction: Windows '95/'98 now support this through the SV2AGW software. However, not all necessary services can be provided from Windows '95/'98 workstations and SV2AGW is not yet available for Windows NT.
- Obtaining all the necessary server packages (and Windows NT itself) would be very costly.
- Don't think it will be that much simpler: this document concerns server configuration. And server configuration is hard, whether it's on Linux, Windows NT, or whatever. You'll need a thorough knowledge on what the service is all about before you can even start thinking of setting up the appropriate server. And Linux itself grows dramatically fast, the need for all these command shell operations won't stay forever.

Appendix: company statements for commercial packages

Even though these products are available for free on Linux platforms, it appears not to be evident they may be included on a CD for experimental use. So, while they slipped onto our first seminar CD (unfortunately), we will no longer include these products on any subsequent seminar CDs to avoid any future ambiguity. You can still download these software packages for free from the respective company websites, however.

Sybase (Adaptive Server Enterprise 11.9.2)

Company website: http://www.sybase.com

Adaptive Server Enterprise for Linux Now Available

Adaptive Server Enterprise version 11.9.2 Development Kit is now available on the Linux operating system. By porting ASE to Linux, Sybase provides the Linux development community with the first highly scalable, high−performance database engine available for the platform. The package includes the standard features of Adaptive Server Enterprise and all related connectivity components.

Adaptive Server Enterprise 11.9.2 is offered FREE for development from the Sybase Adaptive Server Enterprise 11.9.2 download web page. You can purchase support even if you download the free development copy. You can get the software on CD with printed copy of Installation guide for $99 from Sybase sales. To deploy Adaptive Server Enterprise for Linux version 11.9.2 on a production box, contact Sybase sales at 1−800−8SYBASE.

Adaptive Server Enterprise for Linux version 11.0.3.3 continues to be offered as a free, unsupported release for development as well as deployment. Adaptive Server Enterprise for Linux version 11.0.3.3 is available from the Sybase Adaptive Server Enterprise for Linux version 11.0.3.3 download web page.

Three years ago, Sybase made its Open Client interface libraries (ct−lib) available for free on the Linux platform, a move that was well received in the Linux community. Recent customer interest on the Linux platform and its widespread acceptance in many parts of the world have prompted Sybase to take the next step in its commitment to this platform.

InterSystems (Caché PC)

Company website: http://cache.intersys.com.

This text is taken from their FAQ.

Caché PC product

What is Caché PC?

Caché PC is a single−user version of the Caché Post Relational Database. It includes all of the development and deployment capabilities of larger Caché implementations except the ability to network with other Caché systems and to connect terminals, instruments, or similar external input devices. Although it is limited to a single user, Caché PC can run twelve concurrent processes.

On what platforms is Caché PC available?

Caché PC is available for Windows 95, Windows 98, Windows NT, Windows 2000, and Red Hat Linux.

Does Caché PC include materials to help me learn Caché?

Yes, Caché includes a complete online documentation set as well as our Building Applications with Caché tutorial.

Types of licenses

What is the difference between a Registered license and an Unregistered license?

Caché PC is available in two forms: Registered and Unregistered. There is no functionality difference but there are packaging and support differences, as shown in the following table.

	Registered	**Unregistered**
Distribution	Web download or CD distribution	Web download or promotional CD
Documentation	Electronic and printed	Electronic only

	Registered	Unregistered
Support	Three free calls (any day, any time)	One free call (Mon − Fri, 6AM − 6PM EST)
	Additional call coupons can be purchased	Must register license to purchase call coupons
	Access to WRC Direct Web site, including interactive services and patches	Access to public information Web site
License Fee	Low	Free

Can commercial applications be deployed with Unregistered licenses?

Yes. There are no limitations on the types of applications that can be built or deployed with Unregistered Caché PC licenses.

Can an Unregistered license be converted to a Registered license?

Yes, simply by paying the license fee. Please contact our Order Processing Department at orderproc@intersys.com or +1(617) 577−3600.

How can I distinguish a Registered license from an Unregistered license?

During start up, Caché PC displays a customer name for a Registered license and an alternate message for an Unregistered license.

How do I get new versions of Caché PC?

Users of both Registered and Unregistered licenses can download new versions of Caché PC at no charge from InterSystems Web site. For Registered licenses, a CD and printed documentation for the new version can be purchased for one half of the new license fee.

Will applications developed with Caché PC work on other Caché license types and platforms?

Yes, because Caché is functionally compatible from the smallest licenses (Caché PC) to the largest (Caché Enterprise) applications can be developed on one platform and license type and run on another without change.

Why is InterSystems offering free Caché PC licenses?

Our experience is that, once an application developer gets a chance to use Caché, he or she will quickly discover how productive it is and will go on to build excellent applications. We want to remove price as a barrier for those developers to get started.

Sun Microsystems (StarOffice 5.1a)

Company website: http://www.sun.com.

Licensing and Distribution Policies for StarOffice(TM)(9/22/99)

Internal−Only Use[13]

You may freely install StarOffice on multiple computers within your own INTERNAL organization (or school.) The standard end user license, called the Binary Code License (BCL)**, is included with both the download and CD versions of StarOffice 5.1. We recommend that all key users register their copy at www.sun.com/staroffice in order to receive notification of updates and secure access to the web−based support information.

For Education Use

Educational Institution License Agreement

Please download, print out, read and sign the Educational Institution License Agreement.

The Educational Institution License Agreement must be returned by Mail or Fax.

Mailing address:
Sun Microsystems, Inc.
901 San Antonio Road
MS MPK29−206
Palo Alto, CA 94303
Fax: (650) 786−3912

For questions about the Educational Institution License Agreement, click here to send email: star−education−agreement@sun.com

[13] *So we distribute this package for our INTERNAL BELGIAN AMPR.ORG (be.ampr.org) organization. Please keep this in mind!*

We recommend that all key users register their copy at www.sun.com/staroffice in order to receive notification of updates and secure access to the web−based support information.

Bundling the StarOffice software/Cutting CDs

Any account that wants to cut their own CDs requires a legal agreement with Sun Microsystems.

Distribution to Third Parties

To install on computers located in other organizations or publish on a CD (i.e., to re−distribute StarOffice to third parties) requires a legal agreement with Sun Microsystems. However, resellers may wish to consider purchasing a service agreement from Sun to provide warranty, indemnification, and support services. Sun Microsystems plans to have a sample re−distribution agreement available by early November.

User Support Services

Customers may wish to plan to purchase some level of support services from Sun (or another supplier) to supplement the free web−based support. No telephone support is included with the free software. Services such as email and telephone response assistance must be purchased separately.

Sun Microsystems will be adding additional information on these topics to the web site at www.sun.com/staroffice over the coming weeks, so we suggest you check periodically for updates.

** Note: References to site licensing that may be found in the Setup.pdf file of some versions of StarOffice 5.1 are superceded by the Binary Code License and can be ignored.

Customer Inquiries

For support questions not found in the Knowledge Database click here:
starofficesupport@sun.com

For all other StarOffice questions (about orders, broken CDs, double billing, distribution of StarOffice, etc.) click here:
csr−customer−service@sun.com

* Customer pays media, shipping, and handling charges.

Appendix: example.conf for Undernet IRC daemon

```
# ircd.conf  configuration file for ircd version ircu2.9.mu and ircu2.10
#
# Written by Niels <niels@undernet.org>, based on the original example.conf,
# server code and some real−life (ahem) experience.
#
# Thanks and credits to: Run, Trillian, Cym, Morrissey, Chaos, Flynn,
#                        Xorath, WildThang, Mmmm, SeKs, Ghostwolf and
#                        all other Undernet IRC Admins and Operators,
#                        and programmers working on the Undernet ircd.

# This is an example of the configuration file used by the Undernet ircd.
#
# This document is based on a (fictious) server in Europe with a
# connection to the Undernet IRC network. It is primarily a leaf server,
# but if all the other hubs in Europe aren’t in service, it can connect
# to one in the US by itself.
#
# All configuration options start with a letter identifying the option,
# and a colon separated list of options. An asterisk indicates an
# unused field.
#
# Please note that when ircd puts the configuration lines into practice,
# it parses them exactly the other way round than they are listed here.
# This means that you should start your I: lines with the "fall through",
# most vanilla one and end with the most detailed.
#
# There is a difference between the ‘‘hostname’’ and the ‘‘server name’’
# of the machine that the server is run on. For example, the host can
# have ‘‘veer.cs.vu.nl’’ as FQDN, and ‘‘Amsterdam.NL.EU.undernet.org’’ as
# server name.
# A ‘‘server mask’’ is something like ’*.EU.UnderNet.org’’, which is
# matched by ’Amsterdam.NL.EU.undernet.org’ but not by
# ’Manhattan.KS.US.undernet.org’.
#
# First some information about the server.
# M:<server name>:*:<description>:<server port>:<server numeric>
#
# The <server port> is the port that other servers can connect to.
# Client ports need to be specified with a P: line, see below.
#
# Note that <server numeric> has to be unique on the network your server
# is running on, must be between 1 and 64, and is not updated on a rehash.

M:London.UK.Eu.UnderNet.org:*:[127.0.0.1] University of London, England:4400:1

#
# This sets information that can be retrieved with the /ADMIN command.
# It should contain at least an admin Email contact address.
# A:<line 1>:<line 2>:<line 3>

A:The University of London:Undernet IRC server:IRC Admins <irc@london.ac.uk>

#
# All connections to the server are associated with a certain ‘‘connection
# class’’, be they incoming or outgoing (initiated by the server), be they
# clients, servers or Martians. (Note that ircd doesn’t have direct support
# for Martians (yet?); they will have to register as normal users. ;−)
# Take the following Y: lines only as a guide.
# Y:<class>:<ping freq>:<connect freq>:<maximum links>:<sendq size>

# Server classes: 90 = all your uplinks for who you do not wish to hub;
# else in classes 80 and/or 70.
# 50 = leaf servers (only used if your server is a hub)

Y:90:90:300:1:1700000
Y:80:90:300:1:1700000
Y:70:90:300:1:1700000
Y:50:90:300:10:1700000

# Client classes. 10 = locals; 2 = for all .net and .com that are not
# in Europe; 1 = for everybody.

Y:10:90:0:100:160000
Y:2:90:0:5:80000
Y:1:90:0:400:160000
```

```
#
# To allow clients to connect, they need authorization. This can be
# done based on hostmask, address mask, and/or with a password.
# With intelligent use of classes and the maxconnections field in the
# Y: lines, you can let in a specific domain, but get rid of all other
# domains in the same toplevel, thus setting up some sort of ’reverse
# K: line’.
# I:<IP mask or crap to force resolving>:<opt passwd>:<hostmask>::<class>

# Technical description (for examples, see below):
# For every connecting client, the IP−number is know.  A reverse lookup
# on this IP−number is done to get the (/all) hostname(s).
# Each hostname that belongs to this IP−number is matched to <hostmask>,
# and the I: line is used when any matches; the client will then show
# with this particular hostname.  If none of the hostnames matches, then
# the IP−number is matched against the <IP mask ...> field, if this matches
# then the I: line is used nevertheless and the client will show with the
# first (main) hostname if any; if the IP−number did not resolve then the
# client will show with the dot notation of the IP−number.
# There is a special case for the UNIX domain sockets and localhost connections
# though; in this case the <IP mask ...> field is compared with the
# name of the server (thus not with any IP−number representation). The name
# of the server is the one returned in the numeric 002 reply, for example:
# 002 Your host is 2.undernet.org[jolan.ppro], running version ...
# Then the "jolan.ppro" is the name used for matching.
# Therefore, unix domain sockets, and connections to localhost would
# match this I: line:
# I:jolan.ppro::foobar::1
# Finally, I: lines with empty <IP mask ..> or <hostmask> fields are skipped.

# This is the ’fallback’ entry. All .uk, .nl, and all unresolved are
# in these two lines.
# By using two different lines, multiple connections from a single IP
# are only allowed from hostnames which have both valid forward and
# reverse DNS mappings.

I:*@*:1:Unresolved::1
I:Resolved::*@*::1

# If you don’t want unresolved dudes to be able to connect to your
# server, use just:
# I:NotMatchingCrap::*@*::1

# Here, take care of all American ISPs.

I:Resolved::*@*.com::2
I:Resolved::*@*.net::2

# Now list all the .com / .net domains that you wish to have access...
# actually it’s less work to do it this way than to do it the other
# way around − K: lining every single ISP in the US.
# I wish people in Holland just got a .nl domain, and not try to be
# cool and use .com...

I:Resolved::*@*.wirehub.net::1
I:Resolved::*@*.planete.net::1
I:Resolved::*@*.ivg.com::1
I:Resolved::*@*.ib.com::1
I:Resolved::*@*.ibm.net::1
I:Resolved::*@*.hydro.com::1
I:Resolved::*@*.NL.net::1

# You can request a more complete listing, including the "list of standard
# K−lines" from the Routing Committee; it will also be sent to you if
# you apply for a server and get accepted.

# Ourselves − this makes sure that we can get in, no matter how full
# the server is (hopefully).

I:*@193.37.*::*@*.london.ac.uk::10

# You can put a digit (0..9) in the password field, which will make ircd
# only accept a client when the total number of connections to the network
# from the same IP number doesn’t exceed this number.
# The following example would accept at most one connection per IP number
# from "*.swipnet.se" and at most two connections from dial up accounts
# that have "dial??.*" as host mask:
# I:Resolved:1:*@*.swipnet.se::1
# I:Resolved:2:*@dial??.*::1
#
# It is possible to show a different Message of the Day to a connecting
# client depending on its origin.
# T:<hostmask>:<path to motd file>
# DPATH/net_com.motd contains a special MOTD where users are encouraged
# to register their domains and get their own I: lines if they’re in
# Europe, or move to US.UnderNet.org if they’re in the USA.

T:*.net:net_com.motd
T:*.com:net_com.motd

# A different MOTD for ourselves, where we point out that the helpdesk
# better not be bothered with questions regarding irc...

T:*.london.ac.uk:london.motd

#
# One of the many nice features of Undernet is ‘‘Uworld’’, a program
# connected to the net as a server. This allows it to broadcast any mode
# change, thus allowing opers to, for example, ’unlock’ a channel that
# has been taken over.
# There is only one slight problem: the TimeStamp protocol prevents this.
# So there is a configuration option to allow them anyway from a certain
# server.
# Note: (1) These lines are agreed on by every server admin on Undernet;
# (2) These lines must be the same on every single server, or results
# will be disasterous; (3) This is a useful feature, not something that
# is a liability and abused regularly (well... :−)
# If you’re on Undernet, you MUST have these lines. I cannnot stress
# this enough.
# As of ircu2.10.05 is it possible to Jupe nicks. Juped nicks need to be
# added to U: lines.  As per CFV−0095, the following nicks must be juped,
# it is not allowed to jupe others as well.

U:Uworld.EU.undernet.org:EuWorld,E,protocol,StatServ,NoteServ,Undernet:*
U:Uworld2.undernet.org:UWorld2,W,ChanSvr,ChanSaver,ChanServ,COM1,COM2,COM3,COM4:*
U:Uworld.undernet.org:Uworld,X,NickSvr,NickSaver,NickServ,LPT1,LPT2,AUX:*

#
# While running your server, you will most probably encounter individuals
# or groups of persons that you do not wish to have access to your server.
#
# For this purpose, the ircd understands "kill lines".
# K:<host/IP mask>:"<opt reason>":<username mask>
# It is possible to use a file as comment for the ban.
# K:<host/IP mask>:!<path to file>:<usermask>
#
# The default reason is: "You are banned from this server"
# Note that K: lines are local to the server; if you ban a person or a
# whole domain from your server, they can get on IRC via any other server
# that doesn’t have them K: lined (yet).

# With a simple comment, using quotes:
K:*.au:"Please use a nearer server":*
K:*.edu:"Please use a nearer server":*

# With a file, prepending a ’!’ before the filename.
# The file can contain for example, a reason, a link to the
# server rules and a contact address.
K:unixbox.flooder.co.uk:!kline/youflooded.txt:*luser

#
# IP−based kill lines are designated with a lowercase ’k’.  These lines
# use the same format as normal K: lines, except they apply to all hosts,
# even if an IP address has a properly resolving host name.
k:192.168.*:!klines/martians:*

#
# A more flexible way of restricting access to your server is the use
# of "restriction lines". These tell the server to start up an (external)
# program, upon whose output is decided whether the client is allowed
# access. The program should print "Y" or "N <reason>" on its stdout.
# Note that the use of R: lines is discouraged and deprecated, needs a
# compile−time define, eats CPU cycles and may well be taken out in
# future releases of ircd.
# R:<host/IP mask>:<program name>:<username mask>

#
# You probably want your server connected to other servers, so your users
# have other users to chat with.
# IRC servers connect to other servers forming a network with a star or
# tree topology. Loops are not allowed.
# In this network, two servers can be distinguished: "hub" and "leaf"
# servers. Leaf servers connect to hubs; hubs connect to each other.
# Of course, many servers can’t be directly classified in one of these
# categories. Both a fixed and a rule−based decision making system for
# server links is provided for ircd to decide what links to allow, what
# to let humans do themselves, and what links to (forcefully) disallow.
#
# The Connection and Allowing connection lines (also known as C/N lines)
# define what servers the server connect to, and which servers are
# allowed to connect. Note that they come in pairs; they do not work if
# one if present and the other is absent.
# C:<remote hostname or IP>:<password>:<remote server name>:<port>:<class>
# N:<remote hostname or IP>:<password>:<remote server name>:<hostmask>:<class>
#
# If you wish to use ident, prepend "username@" to the hostname or IP
# address (the first field).
# If the "port" field is omitted, the server will not attempt to
# establish a link with that server ("not autoconnecting").
# The (optional) "host mask" field tells the server to represent itself
# with "hostmask" dot−seperateed fields stripped from its servername
# and replace it with "*.".
# For example, if hostmask == 2 and the local server name is
# "irc.sub.domain.com" it would be sent as "*.domain.com". This allows
# for easier routing and linking of new servers.
# This feature is not used on Undernet.

# Our primary uplink.
C:1.2.3.4:passwd:Amsterdam.NL.Eu.UnderNet.org:4400:90
N:1.2.3.4:passwd:Amsterdam.NL.Eu.UnderNet.org::90

#
# If your server starts on a bit larger network, you’ll probably get
# assigned one or two uplinks to which your server can connect.
# If your uplink(s) also connect to other servers than yours (which is
# probable), you need to define your uplink as being allowed to "hub".
# H:<allowed hostmask>::<server name>
H:*.*::Amsterdam.NL.Eu.UnderNet.org

#
# Of course, the opposite is also possible: forcing a server to be
# a leaf. L: lines follow Murphy’s Law: if you use them, there’s a big
# chance that routing will be screwed up afterwards.
# L:<opt disallowed hostmask>::<server mask>:<opt max depth>

#
# For an advanced, real−time rule−based routing decision making system
# you can use Disallow lines. For more information, see doc/readme.crules.
# D:<server mask that ircd will refuse to connect to>::<rule>
# d:<server mask that ircd will not autoconnect to>::<rule>
D:*.US.UnderNet.org::connected(*.US.UnderNet.org)
d:*.EU.UnderNet.org::connected(Amsterdam.NL.EU.*)

# The following line is recommended for leaf servers:
d:*::directcon(*)

#
# Inevitably, you have reached the part about "IRC Operators". Oper status
# grants some special privileges to a user, like the power to make the
# server break or (try to) establish a connection with another server,
# and to "kill" users off IRC.
# I can write many pages about this; I will restrict myself to saying that
# if you want to appoint somebody as IRC Operator on your server, that
# person should be aware of his/her responsibilities, and that you, being
# the admin, will be held accountable for their actions.
#
# There are two sorts of IRC Operators: "local" and "global". Local opers
# can squit, connect and kill − but only locally: their +o user mode
# is not not passed along to other servers. On Undernet, this prevents
# them from using Uworld as well.
# Depending on some defines in include/config.h, local operators are also
# not allowed to /DIE and /RESTART the server.
# Local operators are designated with a lowercase ’o’
# O:<host/IP mask>:<encrypted password>:<Nick>::<connection class>
# o:<host/IP mask>:<encrypted password>:<Nick>::<connection class>

O:*@*.cs.vu.nl:VRKLKuGKn0jLs:Niels::10

# Note that the <connection class> is optional, but leaving it away
# puts the O: lines in class 0, which usually only accepts one connection
# at a time.  If you want users to Oper up more then once per O: line,
# then use a connection class that allows more then one connection,
# for example (using class 10 as in the example above):
# Y:10:90:0:100:160000
#
# When your server gets fuller, you will notice delays when trying to
# connect to your server’s primary listening port. Via the Port lines
# it is possible to specify additional ports (both AF_UNIX and AF_INET)
# for ircd to listen to.
# De facto ports are: 6667 − standard; 6660−6669 − additional client
# ports;
#
# These are just hints, they are in no way official IANA or IETF policies.
#
# On a side note, the /UPING command uses port 7007/udp. If your server
# is located behind a firewall, you may want to make another hole in it
# for this port.
#
# P:<hostmask, or path>:::<client port number>
P::::6667
P::::6668
P:*.nl:::6666
P:/tmp/.ircd:::6667

#
# Well, you have now reached the end of this sample configuration file
# If you have any questions, feel free to mail <doco−com@undernet.org>
# or <wastelanders@undernet.org>.
# If you are interested in linking your server to the Undernet IRC network
# visit http://www.routing−com.undernet.org/, and if there are any problems
# then contact <routing−com@undernet.org> asking for information.
# Upgrades of the Undernet ircd can be found on http://coder−com.undernet.org/.
#
# For the rest:  Good Luck!
#
#                 −− Niels.
```

Appendix: GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111−1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software−−to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program).

Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

> *a) Accompany it with the complete corresponding machine−readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,*
>
> *b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine−readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,*
>
> *c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)*

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty−free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED

OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program’s name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111−1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.

This is free software, and you are welcome to redistribute it under certain conditions; type ‘show c’ for details.

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse−clicks or menu items−−whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program

‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Appendix: GNU LIBRARY GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111−1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software−−to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public Li−

cense, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non−free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non−free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.
(For example, a function in a library to compute square roots has a purpose that is entirely well−defined independent of the application. Therefore, Subsection 2d requires that any application−supplied function or table used by this function must be optional: if the application does not sup−

ply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine−readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

> *a) Accompany the work with the complete corresponding machine−readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine−readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)*
>
> *b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.*
>
> *c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.*
>
> *d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.*

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side−by−side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

> *a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.*
>
> *b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.*

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty−free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library’s name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place − Suite 330, Boston, MA 02111−1307 USA.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library ‘Frob’ (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990

Ty Coon, President of Vice

Index