

fli4l – flexible internet router for linux

Version 4.0.0-testing-sunxi-r60699

Frank Meyer
email: frank@fli4l.de

the fli4l-Team
email: team@fli4l.de

August 12, 2022

Contents

1. Documentation of the base package	5
1.1. Introduction	5
2. Setup and Configuration	8
2.1. Unpacking the archives	8
2.2. Configuration	9
2.2.1. Editing the configuration files	9
2.2.2. Configuration via a special configuration file	10
2.2.3. Variables	10
2.3. Setup flavours	10
2.3.1. Router on a USB-Stick	11
2.3.2. Router on a CD, or network boot	11
2.3.3. Type A: Router on hard disk—only one FAT partition	11
2.3.4. Type B: Router on hard disk—one FAT and one ext3 partition	11
3. Base configuration	13
3.1. Example file	14
3.2. General settings	23
3.3. Console settings	29
3.4. Hints To Identify Problems And Errors	30
3.5. Usage of a customized /etc/inittab	31
3.6. Localized keyboard layouts	32
3.7. Ethernet network adapter drivers	33
3.8. Networks	35
3.9. Network prefix configuration	37
3.9.1. Network prefixes of type “stable”	38
3.9.2. Network prefixes of type “generated-ula”	38
3.10. Additional routes (optional)	38
3.11. The Packet Filter	39
3.11.1. Packet Filter Actions	41
3.11.2. Restrictions For Rules	42
3.11.3. Using Templates With The Packet Filter	45
3.11.4. Configuration Of The Packet Filter	50
3.11.5. Example	56
3.11.6. Default Configurations	59
3.11.7. DMZ – Demilitarized Zone	63
3.11.8. Conntrack-Helpers	63
3.12. Domain configuration	66
3.13. imond configuration	67
3.14. General circuit configuration	69

4. Packages	70
4.1. Tools In The Package 'Base'	70
4.1.1. OPT_SYSLOGD – Logging system messages	70
4.1.2. OPT_KLOGD – Logging kernel messages	72
4.1.3. OPT_LOGIP – Logging WAN IP addresses	72
4.1.4. OPT_Y2K – Date correction for systems that are not Y2K-safe	72
4.1.5. OPT_PNP – Installation of ISAPnP tools	73
4.1.6. OPT_HOTPLUG_PCI – Aktivating PCI hotplugging	74
5. Creating the fli4l Archives/Boot media	75
5.1. Creating the fli4l Archives/Boot media under Linux or other Unix derivatives and Mac OS X	75
5.1.1. Command line options	75
5.2. Creating the fli4l Archives/Boot media under Windows	78
5.2.1. Command line options	78
5.2.2. Configuration dialog – Setting the configuration directory	79
5.2.3. Configuration dialog – General Preferences	80
5.2.4. Configuration dialog – Settings for Remote update	81
5.2.5. Configuration dialog – Settings for HD pre-install	82
5.3. Control file mkfli4l.txt	83
6. Connecting PCs in the LAN	85
6.1. IP address	85
6.2. Host and domain name	85
6.2.1. Windows 2000	85
6.2.2. NT 4.0	86
6.2.3. Win95/98	86
6.2.4. Windows XP	86
6.2.5. Windows 7	87
6.2.6. Windows 8	87
6.3. Gateway	87
6.4. DNS server	88
6.5. Miscellaneous	88
7. Client/Server interface imond	89
7.1. imon-Server imond	89
7.1.1. Least-Cost-Routing – how it works	89
7.1.2. Annotations to the calculation of the online changes	94
7.2. Windows-Client imonc.exe	94
7.2.1. Introduction	94
7.2.2. Start Parameters	95
7.2.3. Overview	96
7.2.4. Config-Dialog	97
7.2.5. Calls Page	102
7.2.6. Connections Page	103
7.2.7. Fax Page	103
7.2.8. E-Mail Page	104

Contents

7.2.9. Admin	104
7.2.10. Error, Syslog and Firewall Pages	105
7.2.11. News Page	105
7.3. Unix/Linux-Client imonc	105
A. Appendix for the base package	108
A.1. Null Modem Cable	108
A.2. Serial Console	108
A.3. Programs	109
A.4. Other i4l-Tools	109
A.5. Debugging	109
A.6. Literature	110
A.7. Prefixes	110
A.8. Warranty and Liability	110
A.9. Credits	111
A.9.1. Foundation Of The Project	111
A.9.2. Developer- and Testteam	111
A.9.3. Developer- and Testteam (inactive)	112
A.9.4. Sponsors	113
A.10.Feedback	114
List of Figures	115
List of Tables	116
Index	117

1. Documentation of the base package

1.1. Introduction

fli4l is a Linux-based router, capable of handling ISDN, DSL, UMTS, and ethernet connections, with little hardware requirements: an USB stick used for booting, an Intel Pentium MMX processor, 64 MiB RAM as well as (at least) one ethernet network adapter are completely sufficient. The necessary boot medium can be created under Linux, Mac OS X or MS Windows. You don't need any specific Linux knowledge, but it is definitely helpful. However, you should possess basic knowledge about networking, TCP/IP, DNS, and routing. For developing your own extensions exceeding the basic configuration, you will need a working Linux system as well as Linux skills.

fli4l supports various boot media, among them USB sticks, hard disks, CDs, and last but not least booting over the network. An USB stick is in many respects ideal: Today, almost every PC can boot from it, it is relatively cheap, it is big enough, and installing fli4l onto it is relatively easy under both MS Windows and Linux. In contrast to a CD it is writable and thus additionally able to hold non-volatile configuration data (as e.g. DHCP leases).

- General features
 - Creation of boot media under [Linux](#) (Page 75), [Mac OS X](#) (Page 75), and [MS Windows](#) (Page 78)
 - Configuration through flat ASCII/UTF-8 files
 - Support for IP masquerading and port forwarding
 - Least Cost Routing (LCR): automatic provider selection based on daytime
 - Displaying/Computing/Logging of connection times and costs
 - MS Windows/Linux client imonc talking to imond and telmond
 - Upload of updated configuration files via MS Windows client imonc or via SCP under Linux
 - Boot media use the VFAT file system as permanent storage
 - Packet filter: External access to blocked ports is logged
 - Uniform mapping of WAN interfaces to so-called circuits
 - Running ISDN and DSL/UMTS circuits in parallel is possible
- Router basics
 - Linux kernel 3.18 or 3.19
 - Packet filter and IP masquerading
 - Local DNS server in order to reduce the number of DNS queries to external DNS servers

1. Documentation of the base package

- Remotely accessible imond server daemon for monitoring and controlling Least Cost Routing
 - Remotely accessible telmond server daemon logging incoming phone calls
- Ethernet support
 - Up-to-date network device drivers: Support for more than 140 adapter types
- DSL support
 - Roaring Penguin PPPoE driver supporting Dial-on-Demand (can be switched off)
 - PPTP for DSL providers in Austria and the Netherlands
- ISDN support
 - Support for some 60 adapter types
 - Multiple possibilities for ISDN connectons: incoming/outgoing/callback, raw/point-to-point (ppp)
 - Channel bundling: automatic band width adaptation or manual activation of the second channel using MS Windows/Linux client software
- Optional software packages
 - DNS server
 - DHCP server
 - SSH server
 - Simple online/offline display using a LED
 - Serial console
 - Minimalistic Web server for ISDN and DSL monitoring as well as for reconfiguring and/or updating the router
 - Ability to let external hosts access LAN hosts in a controlled manner
 - Support for PCMCIA cards (called PC cards nowadays)
 - Logging of system messages
 - Configuration of ISAPnP cards by the use of isapnp tools
 - Additional tools for debugging
 - Configuration of the serial port
 - Rescue system for remote administration over ISDN
 - Software for displaying configurable information on an LCD, e.g. transmission rates, CPU load etc.
 - PPP server/router over the serial port
 - ISDN modem emulator over the serial port
 - Print server
 - Time synchronization with external time servers

1. Documentation of the base package

- Execution of user-defined commands on incoming phone calls (e.g. to perform Internet dial-up)
 - Support for IP aliasing (multiple IP addresses per network interface)
 - VPN support
 - IPv6 support
 - WLAN support: fli4l can be an access point as well as a client
 - RRD tool for monitoring the fli4l
 - and much more...
- Hardware requirements
 - Intel Pentium processor with MMX support
 - 64 MiB RAM, better 128 MiB
 - Ethernet network adapter
 - ISDN: supported ISDN adapter
 - an USB stick, an ATA hard disk or a CF card (which is accessed the same way as an ATA hard disk); alternatively, booting from a CD is also possible
 - Software requirements
- The following tools are required on Linux systems:
- GCC and GNU make
 - syslinux
 - mtools (mcopy)

No additional tools are required on MS Windows systems, all necessary tools are provided by fli4l.

Last but not least, the client utility imonc exists for controlling the router and for displaying the router's state. This tool is available for MS Windows (windows/imonc.exe) and also for Linux (unix/gtk-imonc).

And now ...

Have fun with fli4l!

Frank Meyer and the fli4l team
email: team@fli4l.de

2. Setup and Configuration

2.1. Unpacking the archives

Under Linux:

```
tar xvfz fli4l-4.0.0-testing-sunxi-r60699.tar.gz
```

If this does not work, try the following:

```
gzip -d < fli4l-4.0.0-testing-sunxi-r60699.tar.gz | tar xvf -
```

If you unpack the current version into a directory which already contains fli4l files from a previous installation, you should execute `mkfli4l.sh -c`:

```
cd fli4l-4.0.0-testing-sunxi-r60699
sh mkfli4l.sh -c
```

However, we recommend to use a fresh directory for a new version as you can easily take over the configuration with a file comparing tool.

If you use a MS Windows system, you can extract the compressed tar archive e.g. with WinZip. You have to check, however, that all files are extracted together *with* their corresponding directories (there is a WinZip setting to achieve this). Also, you have to disable the so-called “Smart TAR CR conversion” under *Settings* \Rightarrow *Configuration*, as some important files are corrupted during extraction otherwise.

Alternatively, we recommend using the OpenSource application 7-Zip (<http://www.7-zip.org/>) which is as powerful as WinZip.

The following files are installed in the subdirectory `fli4l-4.0.0-testing-sunxi-r60699/`:

- Documentation:
 - `doc/deutsch/*` German documentation
 - `doc/english/*` English documentation
 - `doc/french/*` French documentation
- Configuration:
 - `config/*.txt` Configuration files, these ones have to be edited to suit your needs
- Scripts/Procedures:
 - `mkfli4l.sh` Creates boot media or files: Linux/Unix version
 - `mkfli4l.bat` Creates boot media: Windows version
- Kernel/Boot files:

2. Setup and Configuration

- img/kernel Linux kernel
- img/boot*.msg bootscreen texts
- Additional packages:
 - opt/*.txt These ones describe which files will be included in the opt.img archive.
 - opt/... Optional kernel modules, files, and programs
- Source code:
 - src/* Source code/tools for Linux, see src/README
- Programs:
 - unix/mkfli4l* Creates boot medium: Linux/Unix version
 - windows/* Creates boot medium: Windows version
 - unix/imonc* imond client for Unix/Linux
 - windows/imonc/* imond client for Windows

2.2. Configuration

2.2.1. Editing the configuration files

To configure fli4l, you only have to adapt the files config/*.txt. We recommend to make a copy of the “config” directory and perform the adaption within this copy. So you will be able to compare your own configuration with the distributed one later and you are also able to manage multiple configurations. Comparing two configurations is relatively simple by using an adequate tool (i.e. “diff” under *nix). Let’s assume that your own config files reside in a directory named my_config under the fli4l directory. In this case you could execute:

```
~/src/fli4l> diff -u {config,my_config}/build/rc.cfg | grep '^[+-]'
```

```
--- config/build/rc.cfg      2014-02-18 15:34:39.085103706 +0100
+++ my_config/build/rc.cfg    2014-02-18 15:34:31.094317441 +0100
-PASSWORD='/P6h4i0IN5Bbc'
+PASSWORD='3P8F3KbjYgzUc'
-NET_DRV_1='ne2k-pci'
+NET_DRV_1='pcnet32'
-START_IMOND='no'
+START_IMOND='yes'
-OPT_PPPOE='no'
+OPT_PPPOE='yes'
-PPPOE_USER='anonymous'
-PPPOE_PASS='surfer'
+PPPOE_USER='me'
+PPPOE_PASS='my-passwd'
-OPT_SSHD='no'
+OPT_SSHD='yes'
```

You can also see by this example that a simple DSL router can be configured without much effort, even if you feel at first overwhelmed by the sheer amount of possible settings.

2.2.2. Configuration via a special configuration file

Due to the module concept of fli4l, the configuration is distributed across different files. As editing these separate files may become tedious, it is possible to store the configuration into a single file called `<config directory>/_fli4l.txt`. This file is read in addition to the other configuration files and its contents override any settings found in the other configuration files. Recall the example above: In order to configure a simple DSL router, we could simply write the following lines into this file:

```
PASSWORD='3P8F3KbjYgzUc'
NET_DRV_N='1'
NET_DRV_1='pcnet32'
START_IMOND='yes'
OPT_PPPOE='yes'
PPPOE_USER='me'
PPPOE_PASS='my-passwd'
OPT_SSHD='yes'
```

You should avoid to mix both flavours of configuration.

2.2.3. Variables

You will notice that the lines of some variables are prefixed with a `'#'` and thus are commented. If this is the case a reasonable default setting is already in effect. Those defaults are documented for each variable. If you wish to set another value delete the `'#'` at the beginning of the line and put your value between the apostrophs.

2.3. Setup flavours

Previous versions of fli4l only supported booting from a floppy disk which is not possible anymore due to causes already described. There are many alternative possibilities nowadays, amongst them using an USB stick.

Many other boot media (CD, HD, network, Compact-Flash, DoC, ...) exist and fli4l may also be installed permanently on some of them (obviously only the read-write ones). fli4l may be booted in three different ways:

Single Image The boot loader loads the linux kernel and then fli4l in a single image. After that, fli4l is able to continue the boot process without the need to access other boot media. Examples are the boot types *integrated*, *attached*, *netboot*, and *cd*.

Split Image The boot loader loads the linux kernel and then a rudimental fli4l image which mounts the boot medium in a first step, then loads the configuration and the remaining files from an archive residing on that mounted medium. Examples for this are the boot types *hd (Type A)*, *ls120*, *attached*, and *cd-emul*.

Installation on a Medium The boot loader loads the linux kernel and then a rudimental fli4l image which mounts an existing fli4l installation without the need to extract any further archives. An example for this is a type B hard disk installation.

2. Setup and Configuration

Before you try the more advanced installation procedures you should make yourself comfortable with fli4l by setting up a minimal version. If you want to use your fli4l as an answering machine or a HTTP-proxy later on, you already feel confident and have the experience of setting up a basic running system.

That said, four variants of installations are possible:

USB-Stick Router on an USB stick

CD-router Router on a CD

Netzwerk Network boot

HD-Installation Typ A Router on a hard disk, CF, DoC – only one FAT-Partition

HD-Installation Typ B Router on a hard disk, CF, DoC – one FAT- and one ext3-Partition

2.3.1. Router on a USB-Stick

USB-Sticks are addressed as harddisks by Linux hence the same explanations as for the hard-disk installation are valid here. Please note that the according drivers for the USB port have to be loaded via `OPT_USB` in order to access the stick with `OPT_HDINSTALL`.

2.3.2. Router on a CD, or network boot

All necessary files are on the boot medium and are extracted to a dynamically sized RAM disk while booting. Using a minimalistic configuration, it is possible to run the router with only 64 MiB of RAM. The maximum setup is only limited by the capacity of the boot medium and available RAM.

2.3.3. Type A: Router on hard disk—only one FAT partition

This corresponds to the CD version, with the only difference of the files residing on a hard disk instead, the term “hard disk” also enclosing Compact Flash from 8 MiB upwards and other devices which are accessed like hard disks under Linux. As of fli4l 2.1.4, you can also use DiskOnChip Flash memory from M-Sys or SCSI hard disks.

The limit for the archive `opt.img` is removed by disk capacity, but all these files have to be installed into a RAM disk of suitable size during the boot process. This increases the necessary amount of RAM if you use many software packages.

In order to update software packages (i.e. the archive `opt.img` and the configuration `rc.cfg` over the network), the FAT partition has to provide enough space for the kernel, the RootFS and TWICE the size of the `opt.img` archive! If you also want to enable the recovery option, the required space is increased one more time by the size of the `opt.img` archive.

2.3.4. Type B: Router on hard disk—one FAT and one ext3 partition

In contrast to type A, most of the files are not put into the RAM disk. Instead, they are copied from the `opt.img` archive to the ext3 partition on the hard disk at the very first start after the initial installation or an update. On successive reboots they are loaded from the ext3 partition. Using this type of installation, the amount of RAM needed for running the router

2. Setup and Configuration

is the smallest, such that running the router with very low memory is possible in the majority of cases.

You can find further information on the hard disk installation in the documentation of the HD package (a separate download) starting at the description of the configuration variable `OPT_HDINSTALL`.

3. Base configuration

Since fli4l 2.0 the distribution is designed to be modular and consists of multiple packages which have to be downloaded separately. The package `fli4l-4.0.0-testing-sunxi-r60699.tar.gz` contains only the base software for a pure ethernet router. For DSL, ISDN, and other software you will have to download further packages and extract them into the directory `fli4l-4.0.0-testing-sunxi-r60699/`. In order to allow free choice of the fli4l's linux system kernel, the kernel has been removed from the base package and was put into an own package. This implies that at least both the base and kernel packages are required. Table 3.1 gives an overview of additional software packages.

The files necessary for configuring the fli4l router are placed in the directory `config/`. They are described later in this chapter.

These files can be edited with a *simple* text editor, or alternatively with an editor specially designed for fli4l. Miscellaneous editors can be found under

<http://www.fli4l.de/en/download/additional-packages/addons/>.

If you need to adapt/extend the fli4l system in addition to the possible settings described below, you will need a working Linux system in order to adjust the RootFS. The file `src/README` will provide you with more information.

3. Base configuration

Table 3.1.: Overview of additional packages

Archive to download	Package
fli4l-4.0.0-testing-sunxi-r60699	BASE, required!
kernel_4_19	Linux kernel, required!
fli4l-4.0.0-testing-sunxi-r60699-doc	Complete documentation
advanced_networking	Extended network configuration
cert	Certificate management
chrony	Time server/client
dhcp_client	Miscellaneous DHCP clients
dns_dhcp	DNS und DHCP servers
dsl	DSL router (PPPoE, PPTP)
dyndns	Support for DYNDNS services
easycron	Time planning service
hd	Needed for hard disk installation
httpd	Minimalistic Web server
hwsupp	Hardware support
imonc_windows	Windows imonc client
imonc_unix	GTK/Unix imonc client
ipv6	Internet Protocol Version 6
isdn	ISDN router
openvpn	VPN support
pcmcia	Support for PCMCIA (PC cards)
ppp	PPP connection over serial port
proxy	Proxy server
qos	Quality of Service
sshd	SSH server
tools	Miscellaneous Linux tools
umts	Connection to the Internet via UMTS
usb	USB support
wlan	Support for WLAN cards

3.1. Example file

The content of the example file `base.txt` in directory `config/` is as follows:

```
##-----
## fli4l __FLI4LVER__ - configuration for package "base"
##
## P L E A S E   R E A D   T H E   D O C U M E N T A T I O N !
##
## B I T T E   U N B E D I N G T   D I E   D O K U M E N T A T I O N   L E S E N !
##
##-----
## Creation:      26.06.2001   fm
## Last Update:   $Id: base.txt 60699 2022-08-12 10:51:37Z lanspezi $
##
```

3. Base configuration

```
## Copyright (c) 2001-2016 - Frank Meyer, fli4l-Team <team@fli4l.de>
##
## 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.
##-----

#-----
# General settings:
#-----
HOSTNAME='fli4l'           # name of fli4l router
PASSWORD='fli4l'           # password for root login (console, sshd,
                           # imond)
BOOT_TYPE='hd'             # boot device: hd, cd, ls120, integrated,
                           # attached, netboot, pxeboot
LIBATA_DMA='disabled'      # Use DMA on ATA Drives ('enabled') or not
                           # ('disabled'). The default 'disabled' allows
                           # ancient IDE CF cards to be booted from.
                           # Use 'enabled' if you boot from a VirtualBox's
                           # virtual device.
MOUNT_BOOT='rw'            # mount boot device: ro, rw, no
BOOTMENU_TIME='5'         # waiting time of bootmenu in seconds
                           # before activating normal boot
TIME_INFO='MEZ-1MESZ,M3.5.0,M10.5.0/3'
                           # description of local time zone,
                           # don't touch without reading documentation
RTC_SYNC='hwclock'         # how to synchronize the hardware clock?
KERNEL_VERSION='5.4.209'  # kernel version
KERNEL_BOOT_OPTION=''     # append option to kernel command line
COMP_TYPE_OPT='xz'         # compression algorithm if compression is
                           # enabled for OPT archive;
                           # NOTE that some boot types may disallow
                           # some compression algorithms
IP_CONNTRACK_MAX=''       # override maximum limit of connection
                           # tracking entries
POWERMANAGEMENT='acpi'    # select pm interface: none, acpi, apm, apm_rm
                           # apm_rm switches to real mode before invoking
                           # apm power off

#-----
# Localisation
#-----
LOCALE='de'                # defines the default language for several
                           # components, such as httpd

#-----
# Console settings (serial console, blank time, beep):
#-----
CONSOLE_BLANK_TIME=''     # time in minutes (1-60) to blank
                           # console; '0' = never, '' = system default
BEEP='yes'                # enable beep after boot and shutdown
SER_CONSOLE='no'          # use serial interface instead of or as
```

3. Base configuration

```
# additional output device and main input
# device
SER_CONSOLE_IF='0'          # serial interface to use, 0 for ttyS0 (COM1)
SER_CONSOLE_RATE='9600'     # baudrate for serial console

#-----
# Debug Settings:
#-----
DEBUG_STARTUP='no'          # write an execution trace of the boot

#-----
# Keyboard layout
#-----
KEYBOARD_LOCALE='auto'      # auto: use most common keyboard layout for
                             # the language specified in 'LOCALE'
#OPT_MAKEKBL='no'           # set to 'yes' to make a new local keyboard
                             # layout map on the fli4l-router

#-----
# Ethernet card drivers:
#-----
#
# please see file base_nic.list in your config-dir or read the documentation
#
# If you need a dummy device, use 'dummy' as your NET_DRV
# and IP_NET_%_DEV='dummy<number>' as your device
#
#-----
#NET_DRV[]='ne2k-pci'        # 1st driver: name (e.g. NE2000 PCI clone)
#{
#  OPTION=''                 # 1st driver: additional option
#}
#NET_DRV[]='ne'              # 2nd driver: name (e.g. NE2000 ISA clone)
#{
#  OPTION='io=0x320'          # 2nd driver: additional option
#}

#-----
# Network prefixes
#-----
#OPT_NET_PREFIX='no'        # enable use of network prefixes: yes or no
#NET_PREFIX                 # network prefixes not bound to an interface
#{
#  []                        # network prefix assignment
#  {
#    NAME="site"              # name of network prefix
#    TYPE="static"            # type of network prefix
#    STATIC_IPV4="192.168.10.0/24" # static IPv4 prefix
#    STATIC_IPV6="fd6e:d748:fd6d::/48" # static IPv6 prefix
#  }
#}
#}
```


3. Base configuration

```
#-----
# ULA prefixes
#-----
#OPT_NET_PREFIX_ULA='no'          # enable generation of ULAs: yes or no
#NET_PREFIX
#{
# []
# {
#   NAME="LAN"                    # name of network prefix
#   TYPE="generated-ula"          # type of network prefix
#   ULA_DEV='eth0'                # Ethernet interface of which the MAC is taken
# }
#}

#-----
# Networks
#-----
OPT_IPV4='yes'                    # enable IPv4 networking
                                  # WARNING: Don't set this to 'no', this is
                                  # currently not supported!

#IP_NET[1]='192.168.6.1/24'        # IP address of your n'th ethernet card and
                                  # netmask in CIDR (no. of set bits)

#{
#   DEV='eth0'                    # required: device name like ethX
#}

#OPT_IPV6='no'                    # set to 'yes' to activate IPv6 support

#IPV6_NET[1]='{internet-v6}+::1:0:0:0:1/64'
                                  # The router address and net mask of
                                  # this subnet. If this subnet is associated
                                  # with a circuit (i.e. the address is
                                  # prefixed by {<circuit>}), use an address
                                  # WITHOUT the subnet prefix; when the
                                  # associated circuit comes up, its prefix
                                  # will be combined with the address
                                  # specified here to yield a complete
                                  # address.
                                  #
                                  # NOTE that the net mask must be equal to
                                  # 64 if you want to use stateless IPv6
                                  # autoconfiguration!
                                  #
                                  # In this example, a /48 subnet prefix is
                                  # assumed which is extended by the subnet
                                  # '1' and the host part '0:0:0:1'. So with
                                  # e.g. '2001:db8:13bc/48' as subnet prefix
                                  # provided by circuit 'internet-v6', the
                                  # complete address and mask becomes
                                  # '2001:db8:13bc:1::1/64'.
                                  #
                                  # If no circuit prefix is used, no circuit
```

3. Base configuration

```
# is associated, so the address
# specification is taken "as is" and is not
# completed by any prefix

#{
#  DEV='IP_NET_1_DEV'          # interface this subnet is bound to
#  ADVERTISE='yes'            # should the subnet prefix be advertised
#                             # automatically via RA in order to enable
#                             # stateless autoconfiguration?
#  ADVERTISE_DNS='no'         # should the DNS service be advertised
#                             # within this subnet via RA?
#}

#-----
# Additional routes, optional
#-----
#IP_ROUTE[]='192.168.7.0/24 192.168.6.99'
#                             # network/netmaskbits gateway
#IP_ROUTE[]='0.0.0.0/0 192.168.6.99'
#                             # example for default-route

#IPV6_ROUTE[]='2001:db8:13bc:2::/64 2001:db8:900:530::1'
#                             # example route

#-----
# Packet filter configuration
#-----
#-----
# INPUT chain
#-----
PF_INPUT_POLICY='REJECT'      # be nice and use reject as policy
PF_INPUT_ACCEPT_DEF='yes'     # use default rule set
PF_INPUT_LOG='no'             # don't log at all
PF_INPUT_LOG_LIMIT='3/minute:5' # log 3 events per minute; allow a burst of 5
#                             # events
PF_INPUT_REJ_LIMIT='1/second:5' # reject 1 connection per second; allow a burst
#                             # of 5 events; otherwise drop packet
PF_INPUT_UDP_REJ_LIMIT='1/second:5'
#                             # reject 1 udp packet per second; allow a burst
#                             # of 5 events; otherwise drop packet
#PF_INPUT[]='IP_NET_1 ACCEPT'  # allow all hosts in the local network to
#                             # access the router
#PF_INPUT[]='tmpl:samba DROP NOLOG'
#                             # drop (or reject) samba access

#{
#  COMMENT='no samba traffic allowed'
#                             # without logging, otherwise the log file will
#                             # be filled with useless entries
#}

PF6_INPUT_POLICY='REJECT'     # be nice and use reject as policy
PF6_INPUT_ACCEPT_DEF='yes'    # use default rule set
PF6_INPUT_LOG='no'            # don't log anything
PF6_INPUT_LOG_LIMIT='3/minute:5'
```

3. Base configuration

```
# log 3 events per minute; allow a burst of 5
# events
PF6_INPUT_REJ_LIMIT='1/second:5'
# reject 1 connection per second; allow a burst
# of 5 events; otherwise drop packet
PF6_INPUT_UDP_REJ_LIMIT='1/second:5'
# reject 1 udp packet per second; allow a burst
# of 5 events; otherwise drop packet

#PF6_INPUT[]='fe80::0/10] ACCEPT'
# allow all hosts in the local network to
# access the router
#PF6_INPUT[]='IPV6_NET_1 ACCEPT'
# allow all hosts in the first subnet to access
# the router
#PF6_INPUT[]='tmp1:samba DROP NOLOG'
# drop (or reject) samba access
#{
# COMMENT='no samba traffic allowed'
# without logging, otherwise the log file will
# be filled with useless entries
#}

#-----
# FORWARD chain
#-----
PF_FORWARD_POLICY='REJECT'      # be nice and use reject as policy
PF_FORWARD_ACCEPT_DEF='yes'     # use default rule set
PF_FORWARD_LOG='no'             # don't log at all
PF_FORWARD_LOG_LIMIT='3/minute:5'
# log 3 events per minute; allow a burst of 5
# events
PF_FORWARD_REJ_LIMIT='1/second:5'
# reject 1 connection per second; allow a burst
# of 5 events; otherwise drop packet
PF_FORWARD_UDP_REJ_LIMIT='1/second:5'
# reject 1 udp packet per second; allow a burst
# of 5 events; otherwise drop packet
#PF_FORWARD[]='tmp1:samba DROP' # drop samba traffic if it tries to leave the
# subnet
#PF_FORWARD[]='IP_NET_1 ACCEPT' # accept everything else

PF6_FORWARD_POLICY='REJECT'      # be nice and use reject as policy
PF6_FORWARD_ACCEPT_DEF='yes'     # use default rule set
PF6_FORWARD_LOG='no'             # don't log anything
PF6_FORWARD_LOG_LIMIT='3/minute:5'
# log 3 events per minute; allow a burst of 5
# events
PF6_FORWARD_REJ_LIMIT='1/second:5'
# reject 1 connection per second; allow a burst
# of 5 events; otherwise drop packet
PF6_FORWARD_UDP_REJ_LIMIT='1/second:5'
# reject 1 udp packet per second; allow a burst
```

3. Base configuration

```
# of 5 events; otherwise drop packet

#PF6_FORWARD[]='tmp1:samba DROP'
# drop samba traffic if it tries to leave the
# subnet
#PF6_FORWARD[]='IPV6_NET_1 ACCEPT'
# accept everything else

#-----
# OUTPUT chain
#-----
PF_OUTPUT_POLICY='ACCEPT'      # default policy for outgoing packets
PF_OUTPUT_ACCEPT_DEF='yes'     # use default rule set
PF_OUTPUT_LOG='no'             # don't log at all
PF_OUTPUT_LOG_LIMIT='3/minute:5'
# log 3 events per minute; allow a burst of 5
# events
PF_OUTPUT_REJ_LIMIT='1/second:5'
# reject 1 connection per second; allow a burst
# of 5 events; otherwise drop packet
PF_OUTPUT_UDP_REJ_LIMIT='1/second:5'
# reject 1 udp packet per second; allow a burst
# of 5 events; otherwise drop packet
#PF_OUTPUT[]='any 217.197.80.132 REJECT'
# don't allow the fli4l to reach fli4l.de

PF6_OUTPUT_POLICY='ACCEPT'     # default policy for outgoing packets
PF6_OUTPUT_ACCEPT_DEF='yes'    # use default rule set
PF6_OUTPUT_LOG='no'            # don't log anything
PF6_OUTPUT_LOG_LIMIT='3/minute:5'
# log 3 events per minute; allow a burst of 5
# events
PF6_OUTPUT_REJ_LIMIT='1/second:5'
# reject 1 connection per second; allow a burst
# of 5 events; otherwise drop packet
PF6_OUTPUT_UDP_REJ_LIMIT='1/second:5'
# reject 1 udp packet per second; allow a burst
# of 5 events; otherwise drop packet
#PF6_OUTPUT[]='any 2001:bf0:c000:a::2:132 REJECT'
# don't allow the fli4l to reach fli4l.de

#-----
# POSTROUTING chain
#-----
#PF_POSTROUTING[]='IP_NET_1 MASQUERADE'
# masquerade traffic leaving the subnet

#PF6_POSTROUTING[]='IPV6_NET_1 MASQUERADE'
# masquerade traffic leaving the subnet

#-----
# PREROUTING chain
#-----
```

3. Base configuration

```
#PF_PREROUTING[]='1.2.3.4 dynamic:22 DNAT:@client2'
# forward ssh connections coming from 1.2.3.4
# to client2

#PF6_PREROUTING[]='tmpl:ssh [2001:db8::1] DNAT:@client2'
# forward ssh connections coming from
# [2001:db8::1] to client2

#-----
# PREROUTING_CT chain
#-----
PF_PREROUTING_CT_ACCEPT_DEF='yes'
# use default rule set
#PF_PREROUTING_CT[]='tmpl:ftp IP_NET_1 HELPER:ftp'
# associate FTP conntrack helper for active FTP
# forwarded from within the LAN to some FTP
# server outside
#PF_PREROUTING_CT[]='tmpl:ftp any dynamic HELPER:ftp'
# associate FTP conntrack helper for passive
# FTP forwarded to the router's external IP
# (some PREROUTING rule needs to forward the
# packets to some FTP server within the LAN)

#PF6_PREROUTING_CT[]='tmpl:ftp IPV6_NET_1 HELPER:ftp'
# associate FTP conntrack helper for active FTP
# forwarded from within the LAN to some FTP
# server outside
#PF6_PREROUTING_CT[]='tmpl:ftp any IPV6_NET_1 HELPER:ftp'
# associate FTP conntrack helper for passive
# FTP forwarded to some FTP server within the
# LAN

#-----
# OUTPUT_CT chain
#-----
PF_OUTPUT_CT_ACCEPT_DEF='yes' # use default rule set
#PF_OUTPUT_CT[]='tmpl:ftp HELPER:ftp'
# associate FTP conntrack helper for outgoing
# active FTP on the router (this rule is added
# automatically by the tools package if
# OPT_FTP='yes' and FTP_PF_ENABLE_ACTIVE='yes')

#PF6_OUTPUT_CT[]='tmpl:ftp HELPER:ftp'
# associate FTP conntrack helper for outgoing
# active FTP on the router (this rule is added
# automatically by the tools package if
# OPT_FTP='yes' and FTP_PF_ENABLE_ACTIVE='yes')

#-----
# USER chain
#-----
#PF_USR_CHAIN[]='...' # some user-defined rule
#PF6_USR_CHAIN[]='...' # some user-defined rule
```

3. Base configuration

```
#-----
# Domain configuration:
# settings for DNS, DHCP server and HOSTS -> see package DNS_DHCP
#-----
DOMAIN_NAME='lan.fli4l'      # your domain name
DNS_FORWARDERS='194.8.57.8'  # DNS servers of your provider,
                              # e.g. ns.n-ix.net

# optional configuration for the host-entry of the router in /etc/hosts
#HOSTNAME_IP='IP_NET_1_IPADDR' # IP to bind to HOSTNAME
#HOSTNAME_IP6='IPV6_NET_1_IPADDR'
                              # optional, can be used to explicitly set
                              # the router's IPv6 address; if left empty,
                              # this setting is taken from the first
                              # configured /64 IPv6 subnet (see below)
#HOSTNAME_ALIAS[]='router.lan.fli4l'
                              # first ALIAS name
#HOSTNAME_ALIAS[]='gateway.my.lan'
                              # second ALIAS name

#-----
# optional package: syslogd
#-----
#OPT_SYSLOGD='no'             # start syslogd: yes or no
#SYSLOGD_RECEIVER='yes'      # receive messages from network
#SYSLOGD_DEST[]='*. * /dev/console'
                              # n'th prio & destination of syslog msgs
#SYSLOGD_DEST[]='*. * @192.168.6.2'
                              # example: loghost 192.168.6.2
#SYSLOGD_DEST[]='kern.info /var/log/dial.log'
                              # example: log infos to file

SYSLOGD_ROTATE='no'          # rotate syslog-files once every day
SYSLOGD_ROTATE_DIR='/data/syslog'
                              # move rotated files to ....
SYSLOGD_ROTATE_MAX='5'       # max number of rotated syslog-files

#-----
# Optional package: klogd
#-----
#OPT_KLOGD='no'              # start klogd: yes or no

#-----
# Optional package: logip
#-----
#OPT_LOGIP='no'              # logip: yes or no
LOGIP_LOGDIR='auto'          # log-directory, e.g. /boot or auto-detected

#-----
# Optional package: y2k correction
#-----
#OPT_Y2K='no'                # y2k correction: yes or no
```

3. Base configuration

```
Y2K_DAYS='0'                                # correct hardware y2k-bug: add x days

#-----
# Optional package: PNP
#-----
#OPT_PNP='no'                                # install isapnp tools: yes or no

#-----
# Optional: PCI hotplugging
#-----
#OPT_HOTPLUG_PCI='no'                        # if yes, various PCI hotplugging drivers are
#                                           # loaded at boot time; note that ACPI hot-
#                                           # plugging (as used by e.g. KVM) is built into
#                                           # the kernel and does not require this OPT to
#                                           # be enabled (but it doesn't hurt neither)

#-----
# Optional package: lua
# (Note: This package will eventually be integrated into the base package as
# it is planned to implement core fli4l services in Lua!)
#-----
#OPT_LUA='no'                                # enable Lua

#-----
# Optional package: luatests
#-----
#OPT_LUATESTS='no'                          # enable Lua test suite
#LUATESTS_RUNATBOOTTIME='yes'               # set to 'yes' if test suite should run when
#                                           # the fli4l boots
```

Please note that this file is stored with DOS line endings, i.e. each line contains an additional carriage return (CR) at the end. Since most Unix editors can handle such files it was decided to use this style, as Windows editors typically do have problems if no CR/LF line endings are used!

If your favourite Unix/Linux editor does not like editing some configuration file due to the DOS line endings, you can convert the DOS line endings to Unix ones with the following command before you start editing the file:

```
sh unix/dtou config/base.txt
```

For the creation of the boot media it is irrelevant whether the file contains DOS oder Unix line endings. They are always converted to Unix style when being written to the boot media.

But let's proceed to the contents ...

3.2. General settings

HOSTNAME Default Setting: `HOSTNAME='fli4l'`

At the very beginning you should choose a name for your fli4l router.

PASSWORD Default Setting: `PASSWORD='fli4l'`

3. Base configuration

This password is needed for logging on to the router—regardless whether you use a keyboard attached to the router or a remote SSH console (for the latter you will need the `sshd` package). The minimum password length is 1, the maximum 126 characters.

BOOT_TYPE Default setting: `BOOT_TYPE='hd'`

`BOOT_TYPE` determines the boot medium in the broadest sense and affects the drivers (kernel modules) and start scripts being included in the RootFS. A short description of the boot process for better understanding:

- The BIOS of the computer loads and starts the boot loader on the boot medium.
- The boot loader (typically `syslinux`) extracts, loads, and starts the kernel.
- The kernel extracts the RootFS (= the basic file system containing tools and scripts needed for booting), mounts the RootFS and begins to execute the start scripts.
- Depending on `BOOT_TYPE`, the start scripts loads the kernel modules for the boot medium, mounts the boot partition, and extracts the OPT archive (`opt.img`) containing additional programs.
- Subsequently, `fli4l` starts to configure the individual services.

The following values are valid for `BOOT_TYPE` at the moment:

ls120 Choose this to boot from LS120/240 and ZIP disks.

hd Choose this to boot from a hard disk. You will find more information in the Documentation (Page ??) of the HD package.

cd Choose this to boot from CD-ROM. With this setting, the ISO image `fli4l.iso` will be created which you have to burn onto CD with your favourite CD burning application. Please pay attention to choosing the right driver for your CD drive.

integrated Choose this if you do not plan to use a conventional boot medium but e.g. want to boot over a network. This setting integrates the OPT archive into the RootFS so the kernel can extract everything at once and does not need not mount a boot medium.

Note: You cannot perform a remote update of your `fli4l` router in this case.

attached This setting is similar to **integrated** but it does not integrate the contents of the OPT archive into the RootFS; rather, the OPT archive is put “as is” into the `/boot` directory. From there it will be extracted during the boot process. Apart from that, the caveats described for **integrated** apply to this boot type as well.

netboot This setting corresponds to **integrated**. However, the script `mknetboot.sh` is additionally run to create an image for booting over the local network. Please read the wiki <https://ssl.networks.org/wiki/display/f/fli4l+und+Netzboot> for further information.

pxeboot Two images are generated, kernel and `rootfs.img`. These are the two files to be loaded by the PXE bootloader. During execution the local tftp directory may be specified and in addition a subdirectory in the tftp directory (`-pxesubdir`). Refer to the Wiki here as well: <https://ssl.networks.org/wiki/display/f/fli4l+und+Netzboot>.

3. Base configuration

Note: How to configure fli4l as a boot-server (pxe/tftp) you can find in the documentation of `opt dns_dhcp`!

LIBATA_DMA Default Setting: `LIBATA_DMA='disabled'`

This options selects if DMA is used for libata based Devices. It is needed for example for incompletely wired IDE to CompactFlash Adapters. Select 'enabled' to use DMA.

MOUNT_BOOT Default Setting: `MOUNT_BOOT='rw'`

This variable specifies how to mount the boot medium. There are three possibilities:

- rw** – Read/Write – Writing and reading is possible
- ro** – Read-Only – Only reading is possible
- no** – None – Medium will be unmounted after booting and can then be removed if desired

Some configurations require mounting the boot medium read/write, e.g. if you want to run a DHCP server or if you want the imond log file to be stored on the boot medium.

BOOTMENU_TIME Default setting: `BOOTMENU_TIME='20'`

This variable controls how LONG the syslinux boot loader should wait until the default installation is booted automatically.

The `OPT_RECOVER` variable of the HD package allows you to activate a function which enables you to create a recovery installation from a working installation. This recovery installation can be activated in the boot menu by choosing the recovery version.

If this variable contains the value '0', the syslinux boot loader will wait indefinitely until the user chooses either the default or the recovery installation!

TIME_INFO Default Setting: `TIME_INFO='MEZ-1MESZ,M3.5.0,M10.5.0/3'`

Normally, Unix operating systems use the UTC (Coordinated Universal Time) for clocks running under their control, and so does fli4l. The UTC is consistent around the world and has to be converted to local time before use. By using `TIME_INFO`, you provide the necessary information for fli4l about your time zone, its difference to UTC, and about daylight saving time. Your local hardware clock must be set to UTC (corresponds to London Standard Time) in order to make these settings effective. Alternatively, you may use the chrony Package which allows fli4l to synchronize its clock with an external time server (time servers always provide the current time in UTC).

The meaning of the possible settings `TIME_INFO` are as follows:

`TIME_INFO='MEZ-1MESZ,M3.5.0,M10.5.0/3'`

- *MEZ-1*: “MEZ” is the German abbreviation for “Central European Time” (you could also use “CET” here). The “-1” means that *MEZ-1=UTC*, i.e. the MEZ is one hour ahead of UTC.
- *MESZ*: “MESZ” is the abbreviation for “Central European Summer Time” and means that fli4l has to handle daylight saving changes. Because no further information (“+x” or “-x”) is given, the clock is adjusted forward one hour when reaching the daylight saving time.

3. Base configuration

- *M3.5.0,M10.5.0/3*: This means that the change to the daylight saving time occurs on the last Sunday in March at two o'clock and that the change to standard time occurs on the last Sunday in October at three o'clock.

Normally you do not have to touch these settings, unless your fli4l router resides in another time zone. In this case you have to adjust these settings accordingly. In order to do this properly, it is helpful to take a look at the specification of the TZ environment variable which can be found at the following URL:

http://pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap08.html

RTC_SYNC Default Setting: `RTC_SYNC='hwclock'`

Many computers have a battery-backed hardware clock, which is also supplied with power while the system is powered off. This enables the system time to be running continuously even then in order to have a valid system time at the next system startup. At this point it is important to differentiate between the *system time* and the *hardware time*:

- *Hardware time* is the time stored in the hardware clock and kept up to date by the device. It is usually read at system startup and becomes system time then.
- *System time* is the time used by the linux system, i.e. if executing the command `date -u`. It is kept up to date by the Linux kernel, for example using hardware interrupts on a regular basis (timer interrupts) and always indicates a time in coordinated Universal Time (UTC). Hence it is not affected by any time zone settings.
- *localized system time* is only the conversion of system time to another time zone. On the fli4l router it is configured by the environment variable TZ (see [TIME_INFO](#) (Page 25)). This is not of any interest for the further explanations below.

With the help of this variable you may configure how fli4l handles the adjustment between hardware time and system time, meaning if and how often hardware time should be set to the system time. Such an adjustment is necessary because even the best hardware clock is not 100 percent accurate and tends to systematic drifting, in the long run it will be a bit too slow or too fast.

There are basically two ways of synchronization:

- “kernel” mode: An NTP client is used to get the real time from outside (usually via the Internet or an external (radio) clock) and keep the system time of the fli4l router up to date. The Linux kernel takes care of updating the hardware time, so no further synchronization is needed. The update by the Linux kernel is a little less accurate than updating by `hwclock` (see “`hwclock`” mode below), however, the quality of the update is less important because errors are compensated by the NTP client.

This mode must also be used if no hardware clock at all exists. The Linux kernel will not keep hardware time up-to-date in this case simply because there is none. In order to have a realistic system time at all, the use of a NTP client should be mandatory.

- Modus “`hwclock`”: At shutdown of the system (when running the `stop-script /etc/rc0.d/rc950.hwclock`) and in regular intervals (every 24 hours) a synchronization using the `hwclock` program will take place. Not only the hardware time is

3. Base configuration

set, but `hwclock` also measures to what extent the system time differs from the hardware time. When starting the system the system time is not taken directly from the hardware time, the time drift is also taken into account in order to minimize drifting of the system time. The drift is stored in the file `/etc/adjtime`. If a writable persistent medium is available, the drift is stored in `/var/lib/persistent/base/adjtime`, in this case `/etc/adjtime` is a symbolic link pointing there.

This mode is incompatible with updating the system time by the help of an NTP client. That's because an NTP client automatically enables updating the hardware clock by the Linux kernel. It is however, of little sense or problematic that both `hwclock` and the Linux kernel at the same time try to keep the hardware time up to date.

It should be noted that if a hardware clock is available, the time stored there is *always* interpreted as coordinated world time (UTC). The time zone defined by the variable `TIME_INFO` does not affect the time stored in the hardware clock. Saving a localized non-UTC time in the hardware clock is *not* supported by fli4l.

Loading the system time from the hardware time is done only once at system startup. The Linux kernel then reads the time stored in the hardware clock and sets system right at the beginning of the boot process. In “`hwclock`” mode the time will be set again later when running the boot script `/etc/rc.d/rc100.hwclock`, this time taking into account the system's time drift.

KERNEL_VERSION Chooses the version of the kernel to be used. According to the contents of this variable, the kernel and the kernel modules are selected from `img/kernel-<kernel version>.<compression extension>` and `opt/lib/modules/<kernel version>`, respectively.

KERNEL_BOOT_OPTION Default Setting: `KERNEL_BOOT_OPTION=""`

The contents of this variable is appended to the kernel's command line defined in `syslinux.cfg`. Some systems require `'reboot=bios'` for proper rebooting, i.e. WRAP systems.

COMP_TYPE_ROOTFS Default setting: `COMP_TYPE_ROOTFS='xz'`

This variable selects the compression method to be used for the RootFS archive. Possible values are `'xz'`, `'lzma'`, and `'bzip2'`.

COMP_TYPE_OPT Default setting: `COMP_TYPE_OPT='xz'`

This variable selects the compression method to be used for the OPT archive. Possible values are `'xz'`, `'lzma'`, and `'bzip2'`.

POWERMANAGEMENT Default Setting: `POWERMANAGEMENT='acpi'`

The kernel supports different flavours of power management: the somewhat aged APM and the newer ACPI. This variable lets you choose which flavour is to be used. Possible values are `'none'` (no power management), `'acpi'`, and the two APM variants `'apm'` and `'apm_rm'`. The latter uses a special processor mode before switching the router off.

FLI4L_UUID Default Setting: `FLI4L_UUID=""`

This variable contains an universally unique identifier (UUID) which is used to point to a place where persistent data can be stored, e.g. on a USB stick. The UUID can be generated on any Linux system (e.g. on the fli4l router) by executing `'cat /proc/sys/kernel/random/uuid'`.

3. Base configuration

Each execution of this command above produces a new UUID which you can use in `FLI4L_UUID` variable. If you create a directory on a persistent medium by the name of this UUID, this directory will be used to store configuration changes as well as persistent run-time data (e.g. DHCP leases). However, the corresponding packages has to support this persistence mechanism (see the documentation to check this). Typically, use 'auto' for the according storage location, instead of a hard-coded path.

If fli4l already stored data using this mechanism before configuring an UUID and creating the directory, this data can be found under `/boot/persistent`. In this case, you will have to manually move the data to the new location. We advice that you generate and configure the UUID at the very beginning, avoiding the migration later on.

Additionally, please note that `MOUNT_BOOT='rw'` is needed if the storage directory is located on the `/boot` partition.

We suggest using the `/data` partition (with the UUID-named directory being a top-level directory there) or an USB stick for the storage location of persistent configuration and run-time data. The file systems allowed are VFAT or, if you use `OPT_HD` all read-writable filesystems supported there.

IP_CONNTRACK_MAX Default Setting: `IP_CONNTRACK_MAX=""`

This variable enables you to change the maximum number of simultaneously existing connections. Normally, a sensible value for this setting is computed automatically, based on the amount of your router's physical RAM. Table 3.2 shows the defaults used.

Table 3.2.: Automtically generated maximum number of simultaneous connections

RAM in MiB	simultaneous connections
16	1024
24	1280
32	2048
64	4096
128	8192

If you use file sharing programs behind or on the router and your router has only little RAM, you will hit the maximum number of simultaneous connections fastly. This will prevent further connections to be established.

This causes error messages as

```
ip_conntrack: table full, dropping packet
```

or

```
ip_conntrack: Maximum limit of XXX entries exceeded
```

The variable `IP_CONNTRACK_MAX` changes the maximum number of simultaneously existing connections to a fixed value. Each possible connection consumes 350 bytes of RAM, which cannot be used for other things. If you e.g. choose the value '10000', you reserve about 3,34 MB RAM that are lost for any other usage (kernel, RAM disks, programs).

3. Base configuration

If your router has 32 MiB RAM, it should not be much of a problem to reserve 2 or 3 MiB for the `ip_conntrack` table. If only 16 MiB RAM or less are available you should be more conservative to prevent your router from running out of RAM.

The setting currently being used can be display on the console by executing

```
cat /proc/sys/net/ipv4/ip_conntrack_max
```

and can be set on-the-fly by executing

```
echo "XXX" > /proc/sys/net/ipv4/ip_conntrack_max
```

where XXX denotes the number of entries. The entries of the `IP_CONNTRACK` table can be displayed on the console by executing

```
cat /proc/net/ip_conntrack
```

and can be counted by executing

```
cat /proc/net/ip_conntrack | grep -c use
```

LOCALE Default setting: `LOCALE='de'`

Meanwhile, some fli4l components support multiple languages, for example the console menu and the Web GUI. This variable lets you choose your preferred language. In addition, some components support a private setting to override this global setting if necessary. English is used as a fallback if the language chosen is not supported for some component.

`KEYBOARD_LOCALE='auto'` tries to find a keyboard layout that is compatible with the `LOCALE` setting.

By now, the following values are possible: `de`, `en`, `fr`.

3.3. Console settings

fli4l can be operated on different hardware platforms. On many of these platforms it is possible to have a keyboard and a monitor connected to interact with fli4l; this input/output combination is generally *console*.

fli4l can also be used completely without keyboard and graphics card. In order to use the router as well without network access to it and to see all kernel boot messages, it is possible to use a console on a serial port catching inputs from the serial interface or sending output there. This requires the variables [SER_CONSOLE](#) (Page 30), [SER_CONSOLE_IF](#) (Page 30) und [SER_CONSOLE_RATE](#) (Page 30) to be set resp. adapted.

It is also possible to use a console on both keyboard and monitor as well as via the serial port.

In general, fli4l provides the option to login and thus a *Shell* on *any* console giving you the ability to login as the user “fli4l” with the password defined in [PASSWORD](#) (Page 23).

3. Base configuration

CONSOLE_BLANK_TIME Default Setting: `CONSOLE_BLANK_TIME=""`

Typically, the Linux kernel activates the console's screen saver after some time without console input activity. The variable `CONSOLE_BLANK_TIME` allows you to configure the timeout to be used or to disable the screen saver completely (`CONSOLE_BLANK_TIME='0'`).

BEEP Default Setting: `BEEP='yes'`

Causes a beep at the end of the boot or shutdown process.

If you enter 'yes' here, there will be a beep at the end of the boot or shutdown process. If you suffer from an extreme shortage of space on your boot media or if you don't like your router to beep, use 'no' instead.

SER_CONSOLE Default Setting: `SER_CONSOLE='no'`

This variable enables or disables a console on a serial port. The serial console can be operated in three modes:

SER_CONSOLE	console input/output
no	Input and output (only) via keyboard and monitor (tty0)
yes	Input and output (only) via serial interface (ttyS0)
primary	Input and output via serial console as well as via keyboard and monitor, output of kernel messages tty0
secondary	Input and output via serial console as well as via keyboard and monitor, output of kernel messages ttyS0

Changing the value of `SER_CONSOLE` affects the router only if you also update your boot media or if you perform a remote update of the `syslinux.cfg` file.

Important: *When turning off the serial console, be sure to keep an alternate access to the router (SSH or directly from the keyboard and monitor)!*

You will find further information in the appendix under [Serial console](#) (Page 108).

SER_CONSOLE_IF Default Setting: `SER_CONSOLE_IF='0'`

Number of the serial interface for the serial console.

Enter the number of the interface to which the serial console is connected. 0 corresponds to ttyS0 under Linux or COM1 under Microsoft Windows.

SER_CONSOLE_RATE Default Setting: `SER_CONSOLE_RATE='9600'`

Transmission rate of the serial port for console output.

This variable contains the Baud rate to use for transmitting data over the serial port. Reasonable values are: 4800, 9600, 19200, 38400, 57600, 115200.

3.4. Hints To Identify Problems And Errors

fl4l logs all output produced while booting into the file (`/var/tmp/boot.log`). After the boot process has finished you can review this file at the console or using the web interface.

Sometimes it is useful to generate a more detailed trace of the start sequence, e.g. to analyze the boot process in case of problems. The variable `DEBUG_STARTUP` exists for this very

3. Base configuration

reason. Other settings help developers to find bugs in certain situations; these settings are also documented in this section.

DEBUG_STARTUP Default Setting: `DEBUG_STARTUP='no'`

If set to 'yes', each command to be executed is written to the console while booting. As a change in `syslinux.cfg` is necessary for enabling this functionality, everything mentioned for `SER_CONSOLE` also applies to this case. If you want to adapt `syslinux.cfg` by hand, you need to insert `fl4ldebug=yes` to it. Nevertheless, `DEBUG_STARTUP` needs to be set to 'yes'.

DEBUG_MODULES Default Setting: `DEBUG_MODULES='no'`

Some modules are loaded automatically by the kernel without further notification. `DEBUG_MODULES='yes'` activates a mode showing the sequence of all modules being loaded, regardless whether they are loaded explicitly by a script or automatically by the kernel.

DEBUG_ENABLE_CORE Default Setting: `DEBUG_ENABLE_CORE='no'`

If this setting is activated, every program crash causes the creation of a so-called “core dump”, a memory image of the process just before the crash. These files are saved in the directory `/var/log/dumps` on the router and can be helpful in finding program errors. More details can be found in the section “Debugging programs on the fl4l” (Page ??) in the documentation of the SRC package.

DEBUG_MDEV Default Setting: `DEBUG_MDEV='no'`

With `DEBUG_MDEV='yes'` all actions related to the `mdev` daemon will be logged, in detail all additions or removals of device nodes in `/dev` or the loading of firmware. Output is directed to the file `/dev/mdev.log`.

DEBUG_IPTABLES Default Setting: `DEBUG_IPTABLES='no'`

With `DEBUG_IPTABLES='yes'` all `iptables` invocations are logged to `/var/log/iptables.log`, including the return values.

DEBUG_IP Default Setting: `DEBUG_IP='no'`

With `DEBUG_IP='yes'` all invocations of the program `/sbin/ip` are logged to the file `/var/log/wrapper.log`.

3.5. Usage of a customized `/etc/inittab`

It is possible to let the “init” process start additional programs on additional consoles or to change the default commands. An `inittab` entry is structured as follows:

```
device:runlevel:action:command
```

The *device* denotes the terminal used for program input/output. Possible devices are terminals `tty1-tty4` or serial terminals `ttyS0-ttySn` with $n < \text{the number of available serial ports}$.

The possible *actions* are typically *askfirst* or *respawn*. Using *askfirst* lets “init” wait for a keypress before running that command. The *respawn* action causes the command to be automatically restarted whenever it terminates.

3. Base configuration

command specifies the program to execute. You have to use a fully qualified path.

The documentation of the Busybox toolkit at <http://www.busybox.net> contains a detailed description of the inittab format.

The normal inittab file is as follows:

```
::sysinit:/etc/rc
::respawn:cttyhack /usr/local/bin/mini-login
::ctrlaltdel:/sbin/reboot
::shutdown:/etc/rc0
::restart:/sbin/init
```

You could e.g. extend it by the entry

```
tty2::askfirst:cttyhack /usr/local/bin/mini-login
```

in order to get a second login process on the second terminal. To achieve this, simply copy the file `opt/etc/inittab` to `<config directory>/etc/inittab` and edit the copy accordingly.

3.6. Localized keyboard layouts

KEYBOARD_LOCALE Default Setting: `KEYBOARD_LOCALE='auto'`

If you sometimes work directly at the router's console you will appreciate a localized keyboard layout. With `KEYBOARD_LOCALE='auto'`, `fli4l` tries to find a keyboard layout that is compatible with the `LOCALE` setting. With `KEYBOARD_LOCALE=""`, no keyboard layout will be installed on the `fli4l` router, causing the kernel's default layout to be used. Alternatively, you may set the variable to the name of a local keyboard layout map. If you e.g. use `KEYBOARD_LOCALE='de-latin1'`, the build process checks whether there is a file named `de-latin1.map` in the directory `opt/etc`. If this is the case, this file will be used when configuring the keyboard layout.

OPT_MAKEKBL Default Setting: `OPT_MAKEKBL='no'`

If you want to create a map file for your keyboard, you have to proceed as follows:

- Set `OPT_MAKEKBL` to 'yes'.
- Invoke `'makekbl.sh'` on the router. Preferably, you use a SSH connection as the keyboard layout changes and this can be quite annoying.
- Follow the instructions.
- You will find your new `<locale>.map` file in `/tmp`.

The tasks to be done directly on the router are now completed.

- Copy the keyboard layout map you have just created to your `fli4l` directory under `opt/etc/<locale>.map`. If you now set `KEYBOARD_LOCALE='<locale>'`, your freshly created keyboard layout will be used when building the `fli4l` images the next time.
- Don't forget to set `OPT_MAKEKBL` to 'no' again.

3.7. Ethernet network adapter drivers

NET_DRV_N Number of needed network adapter drivers.

If you use the router with ISDN, you typically have one network adapter only, hence the default value is 1. However, if you use DSL, you will have two network adapters in the majority of cases.

You have to separate two cases:

1. Both adapters are of the same type. Then you will have to specify only one driver communicating with both adapters, hence `NET_DRV_N='1'`.
2. The types of the adapters used differ. Then you have to set the variable to '2' and to configure the drivers separately for both adapters.

NET_DRV_x NET_DRV_x_OPTION This variable contains the name of the driver to be used for the network adapter. The default for `NET_DRV_1` is to load the driver for a NE2000 compatible network adapter. More available drivers for a large amount of families of network adapters are included in the tables ?? and ??.

The 3COM EtherLinkIII network adapter (3c509) has to be configured by the DOS tool `3c509cfg.exe`, available under:

<ftp://ftp.ihg.uni-duisburg.de/Hardware/3com/3C5x9n/3C5X9CFG.EXE>

It should be used for setting the IRQ and I/O port and, if necessary, the type of connection (BNC/TP). The entry `NET_DRV_x_OPTION=` can normally be left empty.

Some ISA adapters require the driver to have additional information in order to find the adapter, e.g. the I/O address. This is the case e.g. for NE2000 compatible ISA adapters and the EtherExpress16.

In such a case, you can set

```
NET_DRV_x_OPTION='io=0x340'
```

(or the corresponding numerical value).

If no options are required, you can leave this variable empty.

If you need to specify more than one option, you have to separate them by blanks, e.g.

```
NET_DRV_x_OPTION='irq=9 io=0x340'
```

If you use two identical network adapters, e.g. NE2000 ISA adapters, you have to separate the different I/O ports by commas:

```
NET_DRV_x_OPTION='io=0x240,0x300'
```

No space is allowed before or after the comma!

This does not work with all network adapter drivers. Some of them need to be loaded twice, i.e. you have to use `NET_DRV_N='2'`. In this case you will have to assign different names to the adapters by using the “-o” option, e.g.

3. Base configuration

```
NET_DRV_N='2'
NET_DRV_1='3c503'
NET_DRV_1_OPTION='-o 3c503-0 io=0x280'
NET_DRV_2='3c503'
NET_DRV_2_OPTION='-o 3c503-1 io=0x300'
```

We recommend to try the “comma” method first before falling back to loading the driver multiple times.

Some more examples:

- 1 x NE2000 ISA

```
NET_DRV_1='ne'
NET_DRV_1_OPTION='io=0x340'
```

- 1 x 3COM EtherLinkIII (3c509)

```
NET_DRV_1='3c509'
NET_DRV_1_OPTION=''
```

For this adapter, see also:

http://extern.fli4l.de/fli4l_faqengine/faq.php?display=faq&faqnr=132&catnr=7&prog=1

http://extern.fli4l.de/fli4l_faqengine/faq.php?display=faq&faqnr=133&catnr=7&prog=1

http://extern.fli4l.de/fli4l_faqengine/faq.php?display=faq&faqnr=135&catnr=7&prog=1

- 2 x NE2000 ISA

```
NET_DRV_1='ne'
NET_DRV_1_OPTION='io=0x320,0x340'
```

Here, you will often need to specify the IRQ values:

```
NET_DRV_1_OPTION='io=0x320,0x340 irq=3,5'
```

You should first try the configuration without specifying any IRQs and enter IRQs only if the network adapters are not found otherwise.

- 2 x NE2000 PCI

```
NET_DRV_1='ne2k-pci'
NET_DRV_1_OPTION=''
```

- 1 x NE2000 ISA, 1 x NE2000 PCI

```
NET_DRV_1='ne'
NET_DRV_1_OPTION='io=0x340'
NET_DRV_2='ne2k-pci'
NET_DRV_2_OPTION=''
```

- 1 x SMC WD8013, 1 x NE2000 ISA

```
NET_DRV_1='wd'
NET_DRV_1_OPTION='io=0x270'
NET_DRV_2='ne2k'
NET_DRV_2_OPTION='io=0x240'
```

3. Base configuration

You can find the complete list of available drivers in the documentation of the respective kernel package.

If you need a dummy device, use 'dummy' as your `NET_DRV_x` and `IP_NET_x_DEV` (Page 35)='dummy<number>' as your device.

3.8. Networks

IP_NET_N Default Setting: `IP_NET_N='1'`

Number of networks to bound to the IP protocol, normally one ('1'). If you set `IP_NET_N` to zero because you don't have any IP networks or because you configure them in a different way, `mkfli4l` will emit a warning when building the archives. You can disable this warning by using `IGNOREIPNETWARNING='yes'`.

IP_NET_x Default Setting: `IP_NET_1='192.168.6.1/24'`

The IP address and the net mask of the router's n-th device using the CIDR¹ notation. If you want the router to receive its IP address dynamically via a DHCP-client it is possible to set this variable to 'dhcp'.

The following table shows how the CIDR notation and the dot notation for net masks are connected.

CIDR	Net mask	Number of IP addresses
/8	255.0.0.0	16777216
/16	255.255.0.0	65536
/23	255.255.254.0	512
/24	255.255.255.0	256
/25	255.255.255.128	128
/26	255.255.255.192	64
/27	255.255.255.224	32
/28	255.255.255.240	16
/29	255.255.255.248	8
/30	255.255.255.252	4
/31	255.255.255.254	2
/32	255.255.255.255	1

Note: As one address is reserved for the network and one for broadcasting, the maximum number of hosts in the network is computed by: `Number_Hosts = Number_IPs - 2`. Consequently, the smallest possible net mask is /30 which corresponds to four IP addresses and hence to two possible hosts.

IP_NET_x_DEV Default Setting: `IP_NET_1_DEV='eth0'`

Required: device name of the network adapter.

Starting with version 2.1.8, the name of the device used has to be supplied! Names of network devices typically start with 'eth' followed by a number. The first network adapter recognized by the system receives the name 'eth0', the second one 'eth1' and

¹Classless Inter-Domain Routing

3. Base configuration

so on.

Example:

```
IP_NET_1_DEV='eth0'
```

The fli4l router is also able to do IP aliasing, i.e. to assign multiple IP addresses to a single network adapter. Additional IP addresses are simply specified by linking another network to the same device. When mkfli4l checks the configuration you are informed that you define such an alias—you can ignore the warning in this case.

Example:

```
IP_NET_1='192.168.6.1/24'  
IP_NET_1_DEV='eth0'  
IP_NET_2='192.168.7.1/24'  
IP_NET_2_DEV='eth0'
```

IP_NET_x_MAC Default Setting: `IP_NET_1_MAC=""`

Optional: MAC address of the network adapter.

With this variable you are able to change the hardware address (MAC) of your network adapter. This is useful if you want to use a DHCP provider expecting a certain MAC address. If you leave `IP_NET_x_MAC` empty or remove the variable definition completely, the original MAC address of your network adapter will be used. Most users will never need to use this variable.

Example:

```
IP_NET_1_MAC='01:81:42:C2:C3:10'
```

IP_NET_x_NAME Default Setting: `IP_NET_x_NAME=""`

Optional: Assigning a name to the IP address of a network adapter.

If you perform reverse DNS lookup of the network adapter's IP address, the result is typically a name like `'fli4l-ethx.<domain>'`. You can use the variable `IP_NET_x_NAME` in order to change that name which will be returned when performing reverse DNS lookup. If the IP address is globally accessible, you can use this setting to enforce that reverse DNS lookups always return a globally accessible name.

Example:

```
IP_NET_2='80.126.238.229/32'  
IP_NET_2_NAME='ajv.xs4all.nl'
```

IP_NET_x_TYPE**IP_NET_x_COMMENT** Default Setting: `IP_NET_x_COMMENT=""`

Optional: You can use this setting to assign a ‘meaningful’ name to a network device. This name can then be used in packages like rrdtool for identifying the network.

3.9. Network prefix configuration**OPT_NET_PREFIX** Enables support for custom network prefixes.

A network prefix is technically nothing else than the address of a network, but it usually stands for a network that shall be divided further. This is especially useful if a fli4l router should not manage the whole network, but leave subnets for other routers. By the definition (and thus naming) of the whole network it is possible to use the network address in several places without always having to write the prefix again.

Concrete examples of how to define a network prefix can be found below for the different types of network prefixes.

Default Setting: `OPT_NET_PREFIX='yes'`

NET_PREFIX_x This array defines the various network prefixes. The individual components are explained below.

NET_PREFIX_x_NAME Name of the network prefix.

This variable contains the name of the prefix. This name can then be used in address informations in order to use the prefix. The name has to be set like circuit names, i.e. it must be specified in curly brackets.

NET_PREFIX_x_TYPE Type of the network prefix.

This variable contains the type of the prefix. The supported types are explained in Tab. 3.3.

Typw	Explanation
static	The network prefix is specified directly as a fixed address.
generated-ula	The network prefix is generated by fli4l as an ULA ² according to RFC 4193. ³ If the fli4l has access to persistent storage, then the prefix is only generated once, so it also remains intact during reboots of the router.

Table 3.3.: Types of network prefixes

²“Unique Local Address”

³<https://tools.ietf.org/html/rfc4193>

3.9.1. Network prefixes of type “stable”

For network prefixes of type “stable” the following settings apply:

NET_PREFIX_x_STATIC_IPV4 NET_PREFIX_x_STATIC_IPV6 Adresse(s) of the network prefix.

This setting can be used to specify the IPv4 and/or IPv6 address of the network prefix.

Example:

```
NET {
  PREFIX {
    [] {
      NAME='site'
      TYPE='static'
      STATIC {
        IPV4='10.1.0.0/16'
        IPV6='fdce:1c35:301f::/48'
      }
    }
  }
}
```

3.9.2. Network prefixes of type “generated-ula”

For Network prefixes of type “generated-ula” the following settings apply:

NET_PREFIX_x_ULA_DEV Ethernet-Interface.

This setting specifies the Ethernet interface whose MAC address is used to generate the ULA.

Example:

```
NET {
  PREFIX {
    [] {
      NAME='site'
      TYPE='generated-ula'
      ULA {
        DEV='eth0'
      }
    }
  }
}
```

3.10. Additional routes (optional)

IP_ROUTE_N Number of additional network routes. Additional network routes are mandatory if e.g. other routers in the LAN exist which have to be used to access other networks.

3. Base configuration

Normally, you do not need to specify any other network routes.

Default setting: `IP_ROUTE_N='0'`

IP_ROUTE_x The additional routes `IP_ROUTE_1`, `IP_ROUTE_2`, ... are structured as follows:

```
network/netmaskbits gateway
```

In this case, `network` is the network address, `/netmaskbits` the net mask using the [CIDR](#) (Page 35) notation and `gateway` the address of the router needed for accessing the other network. Obviously, the gateway and the fli4l router are required to be in the same network! For example, if the network 192.168.7.0 with net mask 55.255.255.0 can be accessed through the gateway 192.168.6.99 you have to add the following entry:

```
IP_ROUTE_N='1'
IP_ROUTE_1='192.168.7.0/24 192.168.6.99'
```

If you use the fli4l router as a pure Ethernet router and not for routing Internet traffic, you can use some `IP_ROUTE_x` variable for specifying a default route. In order to achieve this, you have to specify '0.0.0.0/0' for 'network/netmaskbits', as can be seen in the following example.

```
IP_ROUTE_N='3'
IP_ROUTE_1='192.168.1.0/24 192.168.6.1'
IP_ROUTE_2='10.73.0.0/16 192.168.6.1'
IP_ROUTE_3='0.0.0.0/0 192.168.6.99'
```

3.11. The Packet Filter

The Linux kernel used by fli4l provides a packet filter which controls who is allowed to communicate with or through the Router. Furthermore, things like port forwarding (a packet addressed to the router is forwarded to another internal computer) and masquerading (packets sent from a computer behind the router are changed to look as if they came from the router itself) can be realized.

The structure of the packet filter is shown in [Figure 3.1](#).

Packets arrive over a network interface and pass through the `PREROUTING`-chain. Here the packets addressed to the router are passed to another computer by changing destination address and destination port. If the packet is addressed to the router it is sent to the `INPUT`-chain, if not, to the `FORWARD`-chain. Both chains will check if the packet is permitted. If the packet is accepted, it is delivered to the local destination process or passed via the `POSTROUTING`-chain (in which packet masquerading is done) to the network interface by which it can reach its target. Locally generated packets are filtered in the `OUTPUT`-chain and finally (if successfully) also pass through the `POSTROUTING`-chain to the correct network interface.

With the packet filter configuration, the individual chains of the packet filter can be modified directly. An individual array exists for each chain, one for the `INPUT`-chain (`PF_INPUT_%`), one for the `FORWARD`-chain (`PF_FORWARD_%`), one for the `OUTPUT`-chain (`PF_OUTPUT_%`), one for the

3. Base configuration



Figure 3.1.: Packet Filter Structure

3. Base configuration

PREROUTING-chain (managing port forwarding) (PF_PREROUTING_%), and one for the POSTROUTING-chain, managing packet masquerading (PF_POSTROUTING_%).

An entry in one of these arrays consists mainly of an action (see below) which can be restricted by additional conditions. These conditions relate to properties of the considered packet. A packet contains information about its origin (source PC that has sent the packet), its target (to which PC and which application should the packet be delivered) and much more. Conditions can refer to the following properties of a packet:

- source (source address, source port or both)
- destination (destination address, destination port or both)
- protocol
- interface on which the packet comes in or goes out
- MAC-address of the originating PC
- state of the packet or the connection the packet comes from

If a packet comes in, the entries resp. the resulting rules generated are processed from top to bottom and the first action to which all conditions apply is performed. If none of the rules matches, the default action is executed, which may be specified for (almost) any table.

An entry has the following format, bearing in mind that all restrictions are optional:

```
restriction{0,} [[source] [destination]] action [BIDIRECTIONAL|LOG|NOLOG]
```

At all points where networks, IP addresses or hosts need to be specified, you can also refer to IP_NET_%, IP_NET_%_IPADDR or via @hostname to a host from HOST_%. If OPT_DNS is enabled, then outside of actions via @fqdn also hosts which are *nicht* mentioned in HOST_% can be referenced by their names. This is particularly useful if dealing with external hosts which also possess many (and changing) IP addresses.

3.11.1. Packet Filter Actions

The following actions apply:

Action	chain(s)	Meaning
ACCEPT	all	Accept the packet.
DROP	INPUT FORWARD OUTPUT	Drop the packet (the sender recognizes that just because no answer and no error message comes back).
REJECT	INPUT FORWARD OUTPUT	Reject the packet (the sender gets a corresponding error message).
LOG	all	Log the packet and proceed to the next rule. To distinguish log entries a prefix may be used, specified by LOG:log-prefix. The maximum length of this prefix is 28 characters and it may contain letters, numbers, hyphens (-), and underscores (_).

3. Base configuration

Action	chain(s)	Meaning
MASQUERADE	POSTROUTING	Mask the packet: Replace the source address of the packet by the own one and make sure that replies for this connection are redirected to the correct computer.
SNAT	POSTROUTING	Replace source address and source port of the packet by the address specified as a parameter for SNAT (for all packets belonging to the connection in consideration).
DNAT	PREROUTING	Replace destination address and destination port of the packet by the address specified as a parameter for SNAT (for all packets belonging to the connection in consideration).
REDIRECT	PREROUTING OUTPUT	Replace destination port of the packet by the address specified as a parameter for SNAT (for all packets belonging to the connection in consideration).
NETMAP	PREROUTING POSTROUTING	Copy destination resp. source address of the packet to the range specified as a parameter for NETMAP; the ports stay unchanged (for all packets belonging to the connection in consideration; while changing the destination address in the PREROUTING-chain and the source address of the POSTROUTING-chain).

Table 3.4.: Packet Filter Actions

Some of these actions may be modified in behaviour by using the options `BIDIRECTIONAL`, `LOG` or `NOLOG`. `BIDIRECTIONAL` generates the same rule a second time with source and destination adresse exchanged (and source and destination port exchanged and/or in- and outbound network interface exchanged if specified). `LOG/NOLOG` activates resp. deactivates logging for this rule.

3.11.2. Restrictions For Rules

Restrictions may be defined by constraints explained in the following sections. You may use **any** at any place where you don't want restrictions but want/have to specify something. Constraints can be specified in any order if they have a preceding prefix. This applies to all restrictions, except for specifying a source or destination address which must always be placed directly in front of the action, other constraints must be specified before. Restrictions can also be negated, simply prefix them by a `!`.

Constraints For Source And Target

Each packet contains source and target informations in a tuple of an IP address and ports.⁴ This source resp. target can serve as a constraint and may be addressed like this:

Expression	Meaning
<code>ip</code>	a simple IP address
<code>network</code>	a network declaration in the form of <code><ip>/<netmask></code>
<code>port[-port]</code>	a port resp. a port range
<code>IP_NET_x_IPADDR</code>	the IP address of the <code>x</code> router's interface
<code>IP_NET_x</code>	the <code>x</code> router's subnet
<code>IP_ROUTE_x</code>	the subnet <code>x</code> specified in the route (default routes can't be used, they would match any and are excluded precautiously)
<code>@name</code>	one of the names or aliases set via <code>HOST_%_*</code> ; the associated IP address will be filled in here
<code><ip oder netzwerk>:port[-port]</code>	Host- resp. network address in one of the variants above, combined with a port resp. port range

Table 3.5.: Constraints For Source And Target In Paket Filter Rules

Example: `'192.168.6.2 any DROP'`

If two of these lines shine up the first will be considered as source and the second as target. Hence, in this example we drop the packets originating from the computer with the IP address 192.168.6.2, regardless of where they are targeted.

If only one line exists the decision if target or source is meant will be made depending on the value, which is quite easy:

- If it contains a port value, target is meant,
- in all other cases the source is.

If you would like to shorten the example above you could write `'192.168.6.2 DROP'`. No port is mentioned, hence the constraint is valid for the source (the machine the packet originated from).

If we were to allow communication with the `ssh`-daemon, we could write `'any any:22 ACCEPT'` (packets from any machine to `ssh`-port 22 of any machine will be accepted) or even shorter `'22 ACCEPT'`. Only a port is mentioned, hence we address the target and thus all packets targeted to port 22.

For simplification you may append `BIDIRECTIONAL` to the action to express that the rule is valid for both communication directions. Then rules will be generated with source and target addresses and if applicable ports and network interfaces exchanged while leaving the rest untouched.

⁴A port only exists for TCP- and UDP-packets.

3. Base configuration

Examples:

127.0.0.1 ACCEPT	local communication (source 127.0.0.1) is allowed
any 192.168.12.1 DROP	packets to address 192.168.12.1 will be dropped
any 192.168.12.1 DROP LOG	packets to address 192.168.12.1 will be dropped and logged additionally
any 192.168.12.1 DROP NOLOG	packets to address 192.168.12.1 will be dropped but not logged
22 ACCEPT	packets to port 22 (ssh) will be accepted
IP_NET_1_NET ACCEPT	packets from the subnet connected to the first interface will be accepted
IP_NET_1_NET IP_NET_2_NET ACCEPT BIDIRECTIONAL	communication between the subnets connected to the first and second interface are allowed

Interface Constraints

A rule can be restricted concerning the Interface on which a packet was received resp. will be transmitted. The format is as follows: **if:in:out**

In the INPUT-chain the interface for outbound packets is not restrictable (the packet does not leave anyway), in the POSTROUTING-chain the interface for received packets is not restrictable, because the informations about it do not exist anymore. Only in the FORWARD-chain constraints for both can be defined.

Possible values for *in* resp. *out*:

- **lo** (Loopback-interface, local communication on the router)
- **IP_NET_x_DEV**
- **pppoe** (the PPPoE-interface; only with package **dsl** or **pppoe_server** activated).
- **any**

Protocol Constraints

A rule can be restricted concerning the protocol a packet belongs to. The format is as follows: **prot:protocol** resp. **prot:icmp:icmp-type**. *protocol* can be set to one of the following values:

- **tcp**
- **udp**
- **gre** (Generic Routing Encapsulation)
- **icmp** (additionally you can specify a name for the ICMP-type to be filtered (**echo-reply** or **echo-request**), i.e. **prot:icmp:echo-request**)
- numeric value of the protocol-ID (i.e. 41 for IPv6)
- **any**

If such a constraint does not exist, but port numbers should be used in a rule, then the rule is generated *twice*, once for the **tcp** and once for the **udp** protocol.

MAC-Address Constraints

Via **mac:mac-address** constraints based on the MAC address may be specified.

Packet State Constraints

fli4l's packet filter gathers informations on the state of connections. This informations can be used to filter packets, i.e let only packets pass that belong to connections already existing. The state of a connection can take this values:⁵

State	Meaning
INVALID	The packet does not belong to a know connection.
ESTABLISHED	The packet belongs to a connection, where packets have already been transmitted in both directions.
NEW	The packet has established a new connection or belongs to a connection that did not have packets transmitted in both directions.
RELATED	The packet establishes a new connection, but has a relation to an already existing connection (i.e. <code>ftp</code> establishes a separate connection for data transfer).

Table 3.6.: Packet State Constraints in Packet Filter Rules

States are defined as follows: `state:state(s)`. If you want to specify more than one state they have to be separated by commas. I.e. to let packets pass that belong directly or indirectly to established connections write `state:ESTABLISHED,RELATED` (this makes sense in INPUT- or FORWARD-chain).

Constraints Based On The Frequency Of Actions

Under certain circumstances you may wish to restrict the frequency of actions, i.e. allow only one ICMP-Echo request per second. This may be reached with `limit`-constraints, which look like this: `limit:Frequency:Burst`. The frequency is specified as *n/time units* (second, minute, hour, day), however, events may also occur in rapid succession (Burst). `limit:3/minute:5` for example means that a maximum of three events per minute is allowed, but also five events in rapid succession will be accepted.

3.11.3. Using Templates With The Packet Filter

To simplify dealing with the packet filter you may summarize rules frequently occuring in templates. Thus, it is possible to provide a wide range of packet filtering rules and combine them in a collection with a symbolic name. Instead of directly using protocols and port numbers, you may then use entries such as `tmpl:ssh` if you want to use the `ssh` protocol in a rule. How to deal with templates is shown here using the example of `ssh`.

If you want to reach your fli4l from the Internet via `ssh`, write into an entry in the array variable `PF_input_%` the corresponding service name (here `ssh`) preceded by `tmpl` and the action to apply for this service. Example:

```
PF_INPUT_2='tmpl:ssh ACCEPT'
```

⁵see http://www.sns.ias.edu/~jns/files/iptables_talk/x38.htm for a detailed description

3. Base configuration

tmpl: means that the rule should be based on a template. Specify the name of the service after the ‘:’, adapted to our example hence **ssh**. At last you have to set an action to be bound to the service. Since we want to access the **fl4l** over the internet, we allow the connection with **ACCEPT**. Restrictions for IP-addresses or nets are not provided so the **ssh**-service will be accessible on all interfaces from all networks. If you want to invoke further restrictions for accessing the **ssh**-service you may use the packet filter notation already explained above.

For which services rules are predefined (e.g. templates exist) can be seen in the template file at `opt/etc/fwrules.tmpl/templates`. A list in a table follows (see table 3.7).

Template	Protocol	Port(s)
ad	tcp	389
ad	udp	389
ad	tcp	636
ad	tcp	3268
ad	tcp	3269
ad	udp	88
ad	tcp	88
ad	udp	53
ad	tcp	53
ad	udp	445
ad	tcp	445
ad	tcp	135
ad	tcp	5722
ad	udp	123
ad	udp	464
ad	tcp	464
ad	udp	138
ad	tcp	9389
ad	udp	67
ad	udp	2535
ad	udp	137
ad	udp	139
checkmk	tcp	6556
checkmk	tcp	161
checkmk	udp	161
checkmk	tcp	162
checkmk	udp	162
dhcp	udp	67-68
dns	tcp/udp	53
elster	tcp	159.154.8.2:21
elster	tcp	159.154.8.35:21
elster	tcp	193.109.238.26:8000
elster	tcp	193.109.238.27:8000
elster	tcp	193.109.238.58:80
elster	tcp	193.109.238.59:80
elster	tcp	62.157.211.58:8000
elster	tcp	62.157.211.59:8000
elster	tcp	62.157.211.60:8000
elster	tcp	80.146.179.2:80
elster	tcp	80.146.179.3:80
ftp	tcp	21
http	tcp	80
https	tcp	443
hylafax	tcp	4559

3. Base configuration

Template	Protocol	Port(s)
imap	tcp	143
imaps	tcp	993
imond	tcp	5000
ipmi	tcp	22
ipmi	tcp	2937
ipmi	tcp	443
ipmi	tcp	5120
ipmi	tcp	5123
ipmi	tcp	5900
ipmi	tcp	5901
ipmi	tcp	80
ipmi	tcp	8889
ipmi	udp	623
irc	tcp	6667
ldap	tcp/udp	389
mail	tcp	110
mail	tcp	143
mail	tcp	25
mail	tcp	465
mail	tcp	587
mail	tcp	993
mail	tcp	995
mysql	tcp	3306
nfs	tcp/udp	111
nfs	tcp/udp	2049
nntp	tcp	119
ntp	udp	123
oracle	tcp	1521
pcanywhere	tcp	5631-5632
ping	icmp:0	
ping	icmp:8	
pop3	tcp	110
pop3s	tcp	995
privoxy	tcp	8118
proxmox	tcp	8006
proxmox	tcp	5900
proxmox	tcp	3128
rdp	tcp	3389
rsync	tcp	873
samba	tcp	139
samba	tcp	445
samba	udp	137-138
sip	tcp/udp	5060-5061
smtp	tcp	25
snmp	tcp/udp	161
socks	tcp	1080
squid	tcp	3128
ssh	tcp	22
ssmtp	tcp	465
submission	tcp	587
svn	tcp	3690
syslog	udp	514
teamspeak	tcp	14534
teamspeak	tcp	51234
teamspeak	udp	8767

3. Base configuration

Template	Protocol	Port(s)
telmond	tcp	5001
telnet	tcp	23
teredo	udp	3544
tftp	udp	69
time	tcp/udp	37
traceroute	udp	33404-33464
vdr	tcp	6419
vnc	tcp	5900
whois	tcp	43
xbl	tcp/udp	3074
xbl	udp	88
xmppclient	tcp	5222
xmppserver	tcp	5269

Table 3.7.: Templates Included With fli4l

The Syntax for this kind of packet filter rules is

```
tmpl:<Name of the service> <Constraint> <Action>
```

<Constraint> allows everything mentioned at 3.11.2. Possible values for <Action> are listed and described in 3.11.1.

Some more examples should clarify the process. At first let's have a look at PF_PREROUTING:

```
PF_PREROUTING_N='2'
PF_PREROUTING_1='tmpl:xbl dynamic DNAT:@xbox'
PF_PREROUTING_2='tmpl:https dynamic DNAT:192.168.193.250'
```

The rule PF_PREROUTING_1 supplies the Xbox with everything necessary for Xbox Live. By the use of `tmpl:xbl` all ports and protocols used for Xbox Live will be forwarded to the `xbox`. Instead of using an IP address we use an entry from the `HOST_%_NAME`-array. `dynamic` tells the fli4l to forward all ports from the internet interface.

The second rule forwards the `https`-protocol to a webserver in a DMZ (Demilitarized Zone). No let's have a look at PF_INPUT:

```
PF_INPUT_N='3'
PF_INPUT_1='if:IP_NET_1_DEV:any ACCEPT'
PF_INPUT_2='if:pppoe:any prot:tcp 113 ACCEPT'
PF_INPUT_3='if:br0:any tmpl:dns @xbox IP_NET_1_IPADDR ACCEPT'
```

The first rule allows access to the router for everyone from the net defined in `IP_NET_1`. The second rule opens the `ident`-port needed for package `oident`. The third rule allows the xbox to access fli4l's DNS server. Notice the use of a host alias here.

PF_FORWARD and PF_POSTROUTING do not provide `tmpl`-specific content.

It is also possible to create templates yourself or for other packages to provide their own ones. To create a template you only need to create a text file with the rules in it and name it like the template. For a private template file use the directory `etc/fwrules.tmpl` (create it if necessary) under your `config` directory as shown in picture 3.2. Package developers or users

3. Base configuration



Figure 3.2.: Directory Structure fli4l

needing templates for more than one configuration may place their template files directly in `opt/etc/fwrules.tmpl`. The templates in the user's `config` directory override other settings, though. The templates included in fli4l will be interpreted as the last ones. This enables you to „override“ fli4l's templates when providing templates by the same name in your `config`-directory.

If, for example you like to create the template `vpn_friends`, create a file by the name `vpn_friends`. The template should contain the services `ssh`, `smtp`, `dns` and `samba`. Hence you write the following to `vpn_friends`:

```
prot:tcp 22
prot:tcp 25
53
prot:udp 137-138
prot:tcp 139
prot:tcp 445
```

Every time you use the template `vpn_friends` rules will be created for all contained protocols and ports. `PF_FORWARD_x='tmpl:vpn_friends ACCEPT'` will create theses FORWARD-rules:

```
prot:tcp 22 ACCEPT
prot:tcp 25 ACCEPT
53 ACCEPT
prot:udp 137-138 ACCEPT
prot:tcp 139 ACCEPT
prot:tcp 445 ACCEPT
```

3.11.4. Configuration Of The Packet Filter

The packet filter is mainly configured by four array-variables:

- `PF_INPUT_%` configures the INPUT-chain,
- `PF_FORWARD_%` configures the FORWARD-chain,
- `PF_OUTPUT_%` configures the OUTPUT-chain,
- `PF_PREROUTING_%` configures the PREROUTING-chain and
- `PF_POSTROUTING_%` configures the POSTROUTING-chain.

For all chains following applies the setting of the protocol level in `PF_LOG_LEVEL`, which may be set to one of these values: `debug`, `info`, `notice`, `warning`, `err`, `crit`, `alert`, `emerg`.

Then INPUT-Chain

The INPUT-chain defines who is allowed to access the router. If no rule of the INPUT-chain matches, the default action handles the packet and the protocol variable decides whether a rejection will be written to the system-protocol or not.

The following restrictions apply to the parameters:

- Only `ACCEPT`, `DROP` and `REJECT` can be specified as actions.
- If using interface constraints only the receiving interface can be restricted.

PF_INPUT_POLICY This variable describes the default action to be taken if no other rule applies. Possible values:

- `ACCEPT` (not recommended)
- `REJECT`
- `DROP` (not recommended)

PF_INPUT_ACCEPT_DEF If this variable is set to 'yes' default rules will be generated needed for the correct function of the router. Use 'yes' as a default here.

If you want to configure the router's behaviour completely yourself you may enter 'no' here but you will have to define all rules on your own then. An equivalent to the default behaviour would look like this (the explanation of user defined chains can be found [here](#) (Page 54)):

```
PF_INPUT_ACCEPT_DEF='no'
#
# limit ICMP echo requests - use a separate chain
#
PF_USR_CHAIN_N='1'
PF_USR_CHAIN_1_NAME='usr-in-icmp'
PF_USR_CHAIN_1_RULE_N='2'
PF_USR_CHAIN_1_RULE_1='prot:icmp:echo-request length:0-150 limit:1/second:5 ACCEPT'
PF_USR_CHAIN_1_RULE_2='state:RELATED ACCEPT'
```

3. Base configuration

```
PF_INPUT_N='4'  
PF_INPUT_1='prot:icmp usr-in-icmp'  
PF_INPUT_2='state:ESTABLISHED,RELATED ACCEPT'  
PF_INPUT_3='if:lo:any ACCEPT'  
PF_INPUT_4='state:NEW 127.0.0.1 DROP BIDIRECTIONAL'
```

The first rule branches to the rate limited “usr-in-icmp”-chain. The second only accepts packets belonging to established connections (packets that have either the state `ESTABLISHED` or `RELATED`), and the third one allows local communication (`if:lo:any ACCEPT`). The fourth filters packets that pretend to be local communication but are not accepted by the rules defined before.

If you work with OpenVPN, the rules have to be enhanced to enable packets used by the chains there.

```
PF_INPUT_N='5'  
...  
PF_INPUT_5='ovpn-chain'
```

PF_INPUT_LOG Defines if rejected packets should be logged by the kernel. Log output can be directed to the syslog deamon by activating `OPT_KLOGD`.

PF_INPUT_LOG_LIMIT Defines how often log entries will be generated. The frequency is described as *n/time units* with bursts in analog to the limit constraints, e.g. `3/minute:5`. If this entry is empty a default of `1/second:5` is used, if set to `none`, the limit constraints are disabled.

PF_INPUT_REJ_LIMIT PF_INPUT_UDP_REJ_LIMIT Specifies how often a REJECT-packet is generated when rejecting incoming packets. The frequency is described as *n/time units* with bursts in analog to the limit constraints, e.g. `3/minute:5`. If this entry is empty a default of `1/second:5` is used, if set to `none`, the limit constraints are disabled.

PF_INPUT_ICMP_ECHO_REQ_LIMIT Defines how often fli4l should react to a ICMP-Echo-request. The frequency is described as *n/time units* with bursts in analog to the limit constraints, e.g. `3/minute:5`. If the limit is reached packets will be ignored (DROP). If this entry is empty a default of `1/second:5` is used, if set to `none`, the limit constraints are disabled.

PF_INPUT_ICMP_ECHO_REQ_SIZE Defines the allowed size of an ICMP-Echo-request (in bytes). The packet header has to be included in this setting besides the pure data. The default is 150 bytes.

PF_INPUT_N PF_INPUT_x PF_INPUT_x_COMMENT A list of rules that describe which packets the router should accept resp. reject.

The FORWARD-Chain

By using the FORWARD-chain will be configured which packets are forwarded by the router. If no rule of the FORWARD-chain matches, the default action handles the packet and the protocol variable decides whether a rejection will be written to the system-protocol or not.

With the used parameters the restriction applies that only the actions ACCEPT, DROP and REJECT are allowed.

PF_FORWARD_POLICY This variable describes the default action to be taken if no other rule applies. Possible values:

- ACCEPT
- REJECT
- DROP

PF_FORWARD_ACCEPT_DEF Determines if the router accepts packets belonging to established connections. If this variable is set to 'yes', `fl4l` generates a rule for accepting packets of the according state automatically:

```
'state:ESTABLISHED,RELATED ACCEPT',
```

as well as a rule to drop packets of unknown state:

```
'state:INVALID DROP'.
```

and at last a rule to drop packets with faked IP addresses:

```
'state:NEW 127.0.0.1 DROP BIDIRECTIONAL'.
```

In addition the other subsystems will generate some default rules – a configuration without default rules with port forwarding and OpenVPN would contain at least the following rules:

```
PF_FORWARD_ACCEPT_DEF='no'
PF_FORWARD_N='5'
PF_FORWARD_1='state:ESTABLISHED,RELATED ACCEPT'
PF_FORWARD_2='state:INVALID DROP'
PF_FORWARD_3='state:NEW 127.0.0.1 DROP BIDIRECTIONAL'
PF_FORWARD_4='pfwaccess-chain'
PF_FORWARD_5='ovpn-chain'
```

PF_FORWARD_LOG Defines if rejected packets should be logged by the kernel. Log output can be directed to the syslog daemon by activating `OPT_KLOGD`.

PF_FORWARD_LOG_LIMIT Defines how often log entries will be generated. The frequency is described as *n/time units* with bursts in analog to the limit constraints, e.g. `3/minute:5`. If this entry is empty a default of `1/second:5` is used, if set to `none`, the limit constraints are disabled.

PF_FORWARD_REJ_LIMIT PF_FORWARD_UDP_REJ_LIMIT Specifies how often a REJECT-packet is generated when rejecting incoming packets. The frequency is described as *n/time units* with bursts in analog to the limit constraints, e.g. `3/minute:5`. If this entry is empty a default of `1/second:5` is used, if set to `none`, the limit constraints are disabled.

PF_FORWARD_N PF_FORWARD_x PF_FORWARD_x_COMMENT A list of rules that describe which packets the router should forward resp. reject.

The OUTPUT-Chain

The OUTPUT-chain configures what the router is allowed to access. If no rule of the OUTPUT-chain matches, the default action handles the packet and the protocol variable decides whether a rejection will be written to the system-protocol or not.

With the used parameters the following restrictions apply:

- Only ACCEPT, DROP and REJECT can be specified as actions.
- For interface constraints only the output interface can be restricted.

PF_OUTPUT_POLICY This variable describes the default action to be taken if no other rule applies. Possible values:

- ACCEPT
- REJECT
- DROP

PF_OUTPUT_ACCEPT_DEF If this variable is set to 'yes' default rules necessary for correct function of the router will be generated. Use 'yes' as a default here.

If you want to configure the router's behaviour completely yourself you may enter 'no' here but you will have to define all rules on your own then. An equivalent to the default behaviour would look like this:

```
PF_OUTPUT_ACCEPT_DEF='no'

PF_OUTPUT_N='1'
PF_OUTPUT_1='state:ESTABLISHED,RELATED ACCEPT'
```

This single rule accepts only packets belonging to established connections (e.g. packets of the state ESTABLISHED or RELATED).

PF_OUTPUT_LOG Defines if rejected packets should be logged by the kernel. Log output can be directed to the syslog daemon by activating OPT_KLOGD.

PF_OUTPUT_LOG_LIMIT Defines how often log entries will be generated. The frequency is described as *n/time units* with bursts in analog to the limit constraints, e.g. 3/minute:5. If this entry is empty a default of 1/second:5 is used, if set to none, the limit constraints are disabled.

PF_OUTPUT_REJ_LIMIT PF_OUTPUT_UDP_REJ_LIMIT Specifies how often a REJECT-packet is generated when rejecting incoming packets. The frequency is described as *n/time units* with bursts in analog to the limit constraints, e.g. 3/minute:5. If the limit is exceeded packets will be ignored (DROP). If this entry is empty a default of 1/second:5 is used, if set to none, the limit constraints are disabled.

PF_OUTPUT_N PF_OUTPUT_x PF_OUTPUT_x_COMMENT A list of rules that describe which packets the router should transmit resp. drop.

User Defined Lists

In several cases you may want to establish own chains to filter packets in detail there. These chains can be defined and filled with rules via `PF_USR_CHAIN_%`. The names of the chains have to start with *usr-* and after their definition can be used everywhere in the `INPUT-` or `FORWARD-`chain as actions. The ICMP-filter chain used before will serve as an example here:

```
PF_USR_CHAIN_N='1'
#
# create usr-in-icmp
#
PF_USR_CHAIN_1_NAME='usr-in-icmp'
#
# add rule to usr-in-icmp
#
PF_USR_CHAIN_1_RULE_N='2'
PF_USR_CHAIN_1_RULE_1='prot:icmp:echo-request length:0-150 limit:1/second:5 ACCEPT'
PF_USR_CHAIN_1_RULE_2='state:RELATED ACCEPT'
#
# use chain in PF_INPUT
#
PF_INPUT_2='prot:icmp usr-in-icmp'
```

PF_USR_CHAIN_N Defines the number of user defined chains.

PF_USR_CHAIN_x_NAME Defines the name of an user defined chain. The name has to be prefixed by *usr-*.

PF_USR_CHAIN_x_RULE_N

PF_USR_CHAIN_x_RULE_x

PF_USR_CHAIN_x_RULE_x_COMMENT These variables define the rules to be inserted in the user defined chain. All rules may be used that are also valid for the `FORWARD-`chain. If no rule of the user defined chains matches, the router will return to the parent chain and check the next rule after the branching to the user defined rules.

The NAT-Chains (Network Address Translation)

Packets still can be changed after the routing decision. For example they may get a new target address to be forwarded to another computer (port forwarding) or a new source address may be inserted to mask the network behind the router. Masquerading is used i.e. to provide internet access for a private net over one public IP or a in DMZ-setup to hide the structure of the local net from computers in the DMZ.

Configuration is done with two chains, `PREROUTING-` and `POSTROUTING-`chain. By the `POSTROUTING-`chain the packets are defined that have to be masked by the router. If no rule of the `POSTROUTING-`chain matches, the packets will be forwarded unmasked.

Two variants exist for masquerading: one for network interfaces that do get an IP address allocated on dialin (`MASQUERADE`) and one for network interfaces with static IP address (`SNAT`). `SNAT` in addition expects the source IP address to be inserted into the packet. It may be specified as an:

3. Base configuration

- IP address (Example: SNAT:1.2.3.4),
- IP range (Example: SNAT:1.2.3.4-1.2.3.10)
- or as symbolic reference (Example: SNAT:IP_NET_1_IPADDR)

For both SNAT and MASQUERADE a port or port range may be set to which the source port may be redirected. Usually this notation is necessary because the kernel can choose the ports on its own. But there exist applications that desire the source port unchanged (and thus require 1:1-NAT) or which forbid PAT (Port Address Translation) or NAPT (Network Address and Port Translation). The port range is simply added to the end, like this: SNAT:IP_NET_1_IPADDR:4000-8000.

With the POSTROUTING-chain only ACCEPT, SNAT, NETMAP and MASQUERADE may be used as actions.

PF_POSTROUTING_N PF_POSTROUTING_x PF_POSTROUTING_x_COMMENT

A list of rules that describe which packets the router should mask resp. forward un-masked. If packets should be excluded from masking an ACCEPT-rule for these packets may be put in front of the MASQUERADE rule.

The PREROUTING-chain configures which packets should be transferred to another computer. If no rule of the PREROUTING-chain matches the packets will be processed further without changes. The action DNAT expects the IP address to be inserted as the target address. It may be specified as an:

- IP address (Example: DNAT:1.2.3.4),
- IP range (Example: DNAT:1.2.3.4-1.2.3.10)
- or as a hostname (Example: DNAT:@client1)

At last a port or port range may be set to which the target port may be redirected. This is only necessary if the target port should be changed. The port (range) is simply added to the end, like this: DNAT:@server:21.

REDIRECT behaves like DNAT, except for that the Destination IP address is always set to the (primary) IP address of the interface on which the packet came in so the packet is delivered locally. This is needed i.e. for transparent proxies, see OPT_TRANSPROXY (Page ??).

If you want a port forwarded to an interface with a dynamic address you do not know to which IP the packet should be sent (at the time of configuration). Thus you can use **dynamic** in the PREROUTING-chain as a wildcard for the IP address assigned later on, like this:

```
'dynamic:80 DNAT:1.2.3.4'           # forward http-packets to
                                   # IP address 1.2.3.4
'prot:gre any dynamic DNAT:1.2.3.4' # forward gre-packets (part of the PPTP-
                                   # protocol) to IP address 1.2.3.4
```

Only ACCEPT, DNAT, NETMAP and REDIRECT may be used as actions with the PREROUTING-chain.

For further examples on port forwarding see the next paragraph.

PF_PREROUTING_N PF_PREROUTING_x PF_PREROUTING_x_COMMENT

A list of rules that describe which packets should be forwarded to another target by the router.

3.11.5. Example

Below see some examples of the packet filter configuration.

The fli4l Default Configuration

fli4l's default configuration for the INPUT-chain looks like this:

```
PF_INPUT_POLICY='REJECT'  
PF_INPUT_ACCEPT_DEF='yes'  
PF_INPUT_LOG='no'  
PF_INPUT_N='1'  
PF_INPUT_1='IP_NET_1 ACCEPT'
```

By this we accomplish that

- computers in the local net are allowed to access the router (PF_INPUT_1='IP_NET_1 ACCEPT'),
- local communication on the router itself is allowed (PF_INPUT_ACCEPT_DEF='yes'),
- packets belonging to connections established by the router are accepted (PF_INPUT_ACCEPT_DEF='yes'),
- everything else is rejected (PF_INPUT_POLICY='REJECT'),
- but nothing is logged to the syslog (PF_INPUT_LOG='no').

The FORWARD-chain looks alike: Only packets of our local net and packets belonging to connections that were established by machines in our local net should be forwarded. In addition NetBIOS- and CIFS-packets will be dropped.

```
PF_FORWARD_POLICY='REJECT'  
PF_FORWARD_ACCEPT_DEF='yes'  
PF_FORWARD_LOG='no'  
PF_FORWARD_N='2'  
PF_FORWARD_1='tmp1:samba DROP'  
PF_FORWARD_2='IP_NET_1 ACCEPT'
```

Note the dependance on the order of rules: *At first* the NetBIOS-packets are dropped and *afterwards* the packets of the local net are accepted.

The local net may communicate with the router, its packets get forwarded, only the masking which is necessary for the internet access of a local network is still missing:

```
PF_POSTROUTING_N='1'  
PF_POSTROUTING_1='IP_NET_1 MASQUERADE'
```


3. Base configuration

Trusted Nets

If we do want to have several local subnets which should communicate with each other free and unmasked we have to ensure that packets between those nets don't get dropped or masked. In order to achieve this we add a rule or edit the existing one.

Let's assume we have a DSL connection over PPPoE and the two subnets are IP_NET_1 (192.168.6.0/24) and IP_NET_2 (192.168.7.0/24). In this case the configuration would be as follows:

```
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'
PF_FORWARD_N='4'
PF_FORWARD_1='IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_FORWARD_2='tmpl:samba DROP'
PF_FORWARD_3='IP_NET_1 ACCEPT'
PF_FORWARD_4='IP_NET_2 ACCEPT'

PF_POSTROUTING_N='3'
PF_POSTROUTING_1='IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_POSTROUTING_2='IP_NET_1 MASQUERADE'
PF_POSTROUTING_3='IP_NET_2 MASQUERADE'
```

The first rule ensures forwarding of packets between both subnets without further processing. The third and fourth rule ensure that both subnets also have Internet access. The first rule of the POSTROUTING-chain provides unmasked communication between both subnets.

In other words we could say that only packets transferred over the **pppoe**-interface have to be masked:

```
PF_POSTROUTING_N='1'
PF_POSTROUTING_1='if:any:pppoe MASQUERADE'
```

We could as well have restricted the port filtering to the **pppoe**-interface and combined both subnets to one, as seen here:

```
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'
PF_FORWARD_N='2'
PF_FORWARD_1='if:any:pppoe tmpl:samba DROP'
PF_FORWARD_2='192.168.6.0/23 ACCEPT'

PF_POSTROUTING_N='1'
PF_POSTROUTING_1='if:any:pppoe MASQUERADE'
```

Packets going out over the **pppoe**-interface and those addressed to **udp**-ports 137-138 or to **tcp**-ports 139 and 445 will be dropped (rule 1), all other packets from subnet 192.168.6.0/23 will be forwarded (rule 2).

Route Network

Let's add a net 10.0.0.0/24 (i.e. a dial-in network) which we want to communicate with unmasked, but packets to udp-ports 137-138 and to tcp-Ports 139 and 445 should be dropped:

```
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'
PF_FORWARD_N='4'
PF_FORWARD_1='IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_FORWARD_2='tmpl:samba DROP'
PF_FORWARD_3='192.168.6.0/23 ACCEPT'
PF_FORWARD_4='10.0.0.0/24 ACCEPT'

PF_POSTROUTING_N='2'
PF_POSTROUTING_1='10.0.0.0/24 ACCEPT BIDIRECTIONAL'
PF_POSTROUTING_2='192.168.6.0/23 MASQUERADE'
```

- rule 1 allows unrestricted communication between the subnets IP_NET_1 and IP_NET_2.
- rule 2 drops packets to the samba ports.
- rule 3 and 4 allow forwarding of packets originating from the subnets 192.168.6.0/24, 192.168.7.0/24 and 10.0.0.0/24; the reverse direction is included by writing PF_FORWARD_ACCEPT_DEF='yes'.
- rule 1 of the POSTROUTING-chain ensures that packets to resp. from the subnet 10.0.0.0/24-Subnetz are not masked.

An alternative:

```
PF_POSTROUTING_N='1'
PF_POSTROUTING_1='if:any:pppoe MASQUERADE'
```

This rule enables masking only for packets going out over the **pppoe**-interface.

Blacklists, Whitelists

Blacklists (a machine in this list is forbidden to do something) and Whitelists (a machine in this list is allowed to do something) are defined in a very similar way. Rules are written that are very special at the beginning and to the end are becoming more universal. With a blacklist rules are defined that at the beginning forbid something and at the end allow something to all not previously mentioned. With a Whitelist it is exactly the other way round.

Example 1: All machines in subnet 192.168.6.0/24 except number 12 are allowed to access the Internet as long as they don't use CIFS Ports 137-138 (**udp**), 139 and 445 (**tcp**) to communicate:

```
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'
PF_FORWARD_N='3'
PF_FORWARD_1='192.168.6.12 DROP'
PF_FORWARD_2='tmpl:samba DROP'
```

3. Base configuration

```
PF_FORWARD_3='192.168.6.0/23 ACCEPT'

PF_POSTROUTING_N='1'
PF_POSTROUTING_2='192.168.6.0/24 MASQUERADE'
```

Example 2: Only machine 12 has Internet access (with exception of the ports mentioned above...), all others are only allowed to communicate with another local subnet:

```
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'
PF_FORWARD_N='3'
PF_FORWARD_1='192.168.6.0/24 192.168.7.0/24 ACCEPT BIDIRECTIONAL'
PF_FORWARD_2='tmpl:samba DROP'
PF_FORWARD_3='192.168.6.12 ACCEPT'

PF_POSTROUTING_N='1'
PF_POSTROUTING_1='if:any:pppoe MASQUERADE'
```

3.11.6. Default Configurations

Simple Router Masking A Net Behind Itself

```
#
# Access to the router
#
PF_INPUT_POLICY='REJECT'
PF_INPUT_ACCEPT_DEF='yes'
PF_INPUT_LOG='no'
PF_INPUT_N='1'
PF_INPUT_1='IP_NET_1 ACCEPT'    # all hosts of the local net are allowed
                                # to access the router

#
# Internet access
#
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'

PF_FORWARD_N='2'
PF_FORWARD_1='tmpl:samba DROP' # Samba-packets, that want to leave the
                                # net are dropped
PF_FORWARD_2='IP_NET_1 ACCEPT' # all other packets are allowed
                                # to leave the local net

#
# Maskieren des lokalen Netzes
#
PF_POSTROUTING_N='1'
PF_POSTROUTING_1='IP_NET_1 MASQUERADE' # mask packets leaving the
                                         # subnet
```

Simple Router Masking Two Nets Behind Itself

```
#
# Access to the router
#
PF_INPUT_POLICY='REJECT'
PF_INPUT_ACCEPT_DEF='yes'
PF_INPUT_LOG='no'
PF_INPUT_N='2'
PF_INPUT_1='IP_NET_1 ACCEPT'      # all hosts of the local net are allowed
                                   # to access the router
PF_INPUT_2='IP_NET_2 ACCEPT'      # all hosts of the local net are allowed
                                   # to access the router

#
# Internet access
#
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'

#
# Free communication between the nets
#
PF_FORWARD_N='4'
PF_FORWARD_1='IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_FORWARD_2='tmp1:samba DROP'    # Samba-packets, that want to leave the
                                   # net are dropped
PF_FORWARD_3='IP_NET_1 ACCEPT'    # all other packets are allowed
                                   # to leave the local net
PF_FORWARD_4='IP_NET_2 ACCEPT'    # all other packets are allowed
                                   # to leave the local net

#
# Masking of local nets, unmasked communication between those nets
#
PF_POSTROUTING_N='3'
PF_POSTROUTING_1='IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_POSTROUTING_2='IP_NET_1 MASQUERADE' # mask packets leaving the
                                         # subnet
PF_POSTROUTING_3='IP_NET_2 MASQUERADE' # mask packets leaving the
                                         # subnet
```

Masking DSL-Router With Two Nets Behind It And SSH/HTTP-Access From the Internet

```
#
# Access to the router
#
PF_INPUT_POLICY='REJECT'
PF_INPUT_ACCEPT_DEF='yes'
PF_INPUT_LOG='no'
PF_INPUT_N='4'
```

3. Base configuration

```
PF_INPUT_1='IP_NET_1 ACCEPT'    # all hosts of the local net are allowed
                                # to access the router
PF_INPUT_2='IP_NET_2 ACCEPT'    # all hosts of the local net are allowed
                                # to access the router
PF_INPUT_3='tmpl:ssh ACCEPT'     # allow access to the SSH service
                                # from everywhere
PF_INPUT_4='tmpl:http 1.2.3.4/24 ACCEPT' # allow machines from
                                # a defined subnet access to the
                                # HTTP service

#
# Internet access
#
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'

#
# No communication between the nets, both nets have
# Internet access, Samba-packets are dropped
#
PF_FORWARD_N='2'
PF_FORWARD_1='tmpl:samba if:any:pppoe DROP' # Samba-packets, that want to leave the
                                # net are dropped
PF_FORWARD_2='if:any:pppoe ACCEPT' # all other packets are allowed
                                # to leave the local net

#
# Masking of local nets, unmasked communication between those nets
#
PF_POSTROUTING_N='1'
PF_POSTROUTING_1='if:any:pppoe MASQUERADE' # mask packets leaving the
                                # subnet
```

Port Forwarding

Port forwarding can be accomplished with the PREROUTING-rules like this (TARGET refers to the original target address (optional) and the original target port, NEW_TARGET refers to the new target address and new target port (optional), PROTOCOL refers to the protocol in use):

```
TARGET='<port>'
NEW_TARGET='<ip>'
PROTOCOL='<proto>'
PF_PREROUTING_x='prot:<proto> dynamic:<port> DNAT:<ip>'

TARGET='<port1>-<port2>'
NEW_TARGET='<ip>'
PROTOCOL='<proto>'
PF_PREROUTING_x='prot:<proto> dynamic:<port1>-<port2> DNAT:<ip>'

TARGET='<ip>:<port-a>'
NEW_TARGET='<ip>:<port-b>'
```

3. Base configuration

```
PROTOCOL=<proto>
PF_PREROUTING_x='prot:<proto> any <ip>:<port-a> DNAT:<ip>:<port-b>'
```

Transparent Proxy

If access to the Internet should only be allowed over a local proxy you may force this behaviour by the help of the `PREROUTING`- and `POSTROUTING`-chains without the client noticing it. In principle you need to do this in three steps:

1. Redirect all HTTP-port-request to the Proxy except for its own ones (`PREROUTING`).
2. Change the redirected packets in a way that fools the proxy to think they all come from the router so it will return its answers there (`POSTROUTING`).
3. Allow the packets to pass the `FORWARD`-chain, as far as an entry like

```
PF_FORWARD_x='IP_NET_1 ACCEPT'
```

does not exist (`FORWARD`).

Example 1: Let's assume we only have one net `IP_NET_1`, a squid proxy is running there on a host by the name of `proxy` and the whole `http`-traffic should be processed by it. Squid listens on port 3128. For simplicity we refer via `@proxy` to the host entered in `HOST_1_NAME='proxy'` (see [Domain Configuration](#) (Page 66)).

Here are the resulting rules:

```
...
PF_PREROUTING_x='@proxy ACCEPT'
    # packets from the proxy should not be redirected

PF_PREROUTING_x='prot:tcp IP_NET_1 80 DNAT:@proxy:3128'
    # HTTP-packets from IP_NET_1 will be redirected to @proxy, Port 3128
    # independet of the target

PF_POSTROUTING_x='any @proxy:3128 SNAT:IP_NET_1_IPADDR'
    # change all packets to port 3128 in a way as if they came from
    # fli4l (IP_NET_1_IPADDR)

PF_FORWARD_x='prot:tcp @proxy 80 ACCEPT'
    # let HTTP-packets from the proxy pass the FORWARD-chain (if necessary)
...
```

If more nets or conflicting port forwardings (which are also `DNAT`-rules) exist, the rules may have to be more differentiated.

Example 2: Our proxy by the name of `proxy` resides in `IP_NET_1`, listens to port 3128 and should only serve clients from `IP_NET_1`. `IP_NET_1` is reachable over `IP_NET_1_DEV`. Packets from other nets should not be considered.

```
...
PF_PREROUTING_x='if:IP_NET_1_DEV:any !@proxy 80 DNAT:@proxy:3128'
    # Redirect queries to the HTTP-port that do not emerge from the proxy but
```

3. Base configuration

```
# come in on an internal interface (IP_NET_1_DEV) to the proxy's port.
# At this point it is important to check with if:IP_NET_1_DEV:any that the
# packets are coming from inside because otherwise packets from outside
# would also be redirected (security breakage)

PF_POSTROUTING_x='prot:tcp IP_NET_1 @proxy:3128 SNAT:IP_NET_1_IPADDR'
# Change HTTP-packets originating from IP_NET_1 and destined to proxy-port 3128
# in a way as if they came from fli4l (IP_NET_1_IPADDR)

PF_FORWARD_x='prot:tcp @proxy 80 ACCEPT'
# let HTTP-packets from the proxy pass the FORWARD-chain (if necessary)
...
```

Example 3: To ease our live and shorten the rules we may use templates (see [Using Templates With The Packet Filter](#) (Page 45)). At this point `tmpl:http`, translated in `prot:tcp any:80` is of advantage. `tmpl:http IP_NET_1 DNAT:@proxy:3128` then changes to `prot:tcp IP_NET_1 80 DNAT:@proxy:3128`.

Both `IP_NET_1` and `IP_NET_2` should be redirected transparently over the proxy. Simplified you could write:

```
...
PF_PREROUTING_x='tmpl:http @proxy ACCEPT'
# HTTP-packets from the proxy should not be redirected

PF_PREROUTING_x='tmpl:http IP_NET_1 DNAT:@proxy:3128'
# HTTP-packets from IP_NET_1 should be redirected

PF_PREROUTING_x='tmpl:http IP_NET_2 DNAT:@proxy:3128'
# HTTP-packets from IP_NET_2 should be redirected

PF_POSTROUTING_x='IP_NET_1 @proxy:3128 SNAT:IP_NET_1_IPADDR'
PF_POSTROUTING_x='IP_NET_2 @proxy:3128 SNAT:IP_NET_2_IPADDR'

PF_FORWARD_x='tmpl:http @proxy ACCEPT'
...
```

You may continue here forever...

3.11.7. DMZ – Demilitarized Zone

`fli4l` may also serve to build a DMZ. As this is only another additional ruleset for the router please refer to the wiki at <https://ssl.networks.org/wiki> for the time being.

3.11.8. Conntrack-Helpers

Using IP-Masquerading has the advantage that a bunch of machines in the LAN can be routed over only one official IP-address. However, there are also disadvantages that you have to take into account.

A big problem for example is that no machine from outside can contact the machines in the LAN. This may be desired for security reasons but certain protocols will not work anymore because they require a connection from outside.

3. Base configuration

A classic example is FTP. Beside a communication channel to exchange commands and answers another channel is needed (an IP-port) to transfer the actual data. `fi4l` uses certain `conntrack`-helpers for this in order to open such ports instantaneously and redirect them to the machine in question when needed. The `conntrack`-helper “listens” to the data stream to recognize when such an additional port is needed.

Typical applications for `conntrack`-helpers are i.e. chat-protocols and Internet games.

`Conntrack`-helper are activated over rules in two special arrays. The array `PF_PREROUTING_CT_%` contains helper-assignments to packets coming from outside, the array `PF_OUTPUT_CT_%` contains helper-assignments to packets generated on the router. Some practical examples help to illustrate this.

Example 1: If active FTP from the LAN should be allowed this is, from the router’s view, a connection from outside the router, thus an entry in `PF_PREROUTING_CT_%` has to be created:

```
PF_PREROUTING_CT_N='1'
PF_PREROUTING_CT_1='tmpl:ftp IP_NET_1 HELPER:ftp'
```

The `ftp`-helper module will be loaded for all TCP connections from the local network (`IP_NET_1`) to any other addresses’ port 21 (which is the `ftp`-Port). This module will allow the FTP server to establish a data transfer connection back to the client during this connection by opening a “hole” in the firewall temporarily.

Example 2: If you want to enable passive ftp for a FTP server on the LAN (the data connection is established from the outside to the inside, so that a hole in the firewall must be opened here as well), this is also seen as a connection from outside by the router. Here we see the rule as for this:

```
PF_PREROUTING_CT_N='1'
PF_PREROUTING_CT_1='tmpl:ftp any dynamic HELPER:ftp'
```

By this rule it is expressed that all FTP connections to the dynamic address of the router are associated to the FTP `conntrack` helper. Here `dynamic` was used because it is assumed that the router is responsible for dialing in to the Internet and thus has an external IP address. If the router performs dial-in via DSL, the rule can also be written as:

```
PF_PREROUTING_CT_N='1'
PF_PREROUTING_CT_1='tmpl:ftp if:pppoe:any HELPER:ftp'
```

By this rule it is expressed that all FTP connections coming from the DSL interface (`pppoe`) are associated to the `conntrack` helper.

If the router is not dialing, but e.g. is behind another router (Fritz! box, cable modem, a.s.o.) the following rules can be used:

```
PF_PREROUTING_CT_N='1'
PF_PREROUTING_CT_1='tmpl:ftp if:IP_NET_2_DEV:any HELPER:ftp'
```

It is assumed in the Example, that the connection to the other router is performed over the interface associated with the second subnet (`IP_NET_2_DEV`).

Remember that of course an *additional* configuration of the `FORWARD`-chain is needed to really forward the FTP-packets. A typical rule would be

```
PF_PREROUTING_1='tmpl:ftp any dynamic DNAT:@ftpserver'
```


3. Base configuration

assuming that the host running the FTP-server has the name **ftpserver**.

Example 3: If you like to use active FTP directly from fli4l (perhaps with the help of the **ftp** program from the **Tools**-package) the firewall has to be prepared, this time in the **OUTPUT**-chain by using the array **PF_output_CT_%**:

```
PF_OUTPUT_CT_N='1'  
PF_OUTPUT_CT_1='tmpl:ftp HELPER:ftp'
```

This rule is not necessary if **FTP_PF_ENABLE_ACTIVE='yes'** is used – see the documentation for the **ftp-OPT** in the **tools**-package.

Following is an overview over the existing conntrack-helpers:

Helper	Explanation
ftp	File Transfer Protocol
h323	H.323 (Voice over IP)
irc	Internet Relay Chat
pptp	PPTP Masquerading (By the use of this module it is possible to run more than one PPTP-Client behind the fli4l router at the same time.)
sip	Session Initiation Protocol
sane	SANE Network Procotol
snmp	Simple Network Management Protocol
tftp	Trivial File Transfer Protocol

Table 3.8.: Available Conntrack Helpers In The Packet Filter

Here is an overview over the variables to configure:

PF_PREROUTING_CT_ACCEPT_DEF If this variable is set to ‘yes’, default rules are generated that are necessary for proper functioning of the router. By default, you should use ‘yes’ here.

PF_PREROUTING_CT_N PF_PREROUTING_CT_x PF_PREROUTING_CT_x_COMMENT
List of rules that describe which incoming packets are associated with conntrack helpers by the router.

PF_OUTPUT_CT_ACCEPT_DEF If this variable is set to ‘yes’, default rules are generated that are necessary for proper functioning of the router. By default, you should use ‘yes’ here.

PF_OUTPUT_CT_N PF_OUTPUT_CT_x PF_OUTPUT_CT_x_COMMENT
List of rules that describe which packets generated on the router are associated with conntrack helpers by the router.

3.12. Domain configuration

Windows PCs exhibit a somewhat annoying behaviour: If a DNS server is needed and configured at the Windows system, the server is queried regularly (every five minutes) – even if you don't work at the PC!

If you configured an Internet DNS server at your Windows PC, your next bill might become quite expensive :-)

If you don't already run a DNS server in your LAN, this problem can be solved by enabling the DNS server of your fli4l router. The DNS server software used is DNSMASQ.

Before you start configuring your DNS, however, you should give careful consideration to the domain name and the names of the PCs in your network. The domain name you use will not be visible in the Internet. Therefore, you are free to choose any domain name you like.

Additionally, each of your PCs in the LAN has to have a name assigned. These names have to be known by the fli4l router.

DOMAIN_NAME Default Setting: `DOMAIN_NAME='lan.fli4l'`

You can freely choose any domain name as this local domain is not visible in the Internet. However, you should avoid choosing a name that may exist in the Internet (e.g. somewhat.com) because you won't be able to access that Internet domain.

DNS_FORWARDERS Default Setting: `DNS_FORWARDERS=""`

This variable contains the address of your Internet provider's DNS server if you want your fli4l router to route Internet traffic. The fli4l router will forward all DNS queries which it is not able to answer on its own to the address in this variable.

You can specify more than one DNS forwarder by separating the addresses by blanks.

If more DNS server are specified, they will be queried in the order given by the configuration file, meaning the second will only be used in case that the first server does not return a valid answer and so on.

It is also possible to specify a port number for each DNS forwarder address which is then to be separated from the address by a colon. However, in this case it is required to set `OPT_DNS='yes'` (Page ??) (Package `dns_dhcp` (Page ??)), and you are not allowed to use any of the various `*_USEPEERDNS` options.

Beware: Even if

- `PPPOE_USEPEERDNS` (Page ??),
- `ISDN_CIRC_x_USEPEERDNS` (Page ??) or
- `DHCPCLIENT_x_USEPEERDNS` (Page ??)

are set to 'yes', you need to fill this variable with a valid DNS server address as otherwise no DNS resolution will be possible directly after the router has booted.

Exception: If you use fli4l as a local router *without* a connection to the Internet or other (company) networks with DNS servers, you should set this variable to '127.0.0.1' in order to disable DNS forwarding completely.

HOSTNAME_IP (optional)

This variable can optionally specify to which network 'IP_NET_x' the hostname set by `HOSTNAME` is bound.

HOSTNAME_ALIAS_N (optional)

Number of additional alias host names for the router.

HOSTNAME_ALIAS_x (optional)

Additional alias host name for the router.

3.13. imond configuration

OPT_IMOND Default setting: `OPT_IMOND='no'`

`OPT_IMOND` controls whether to start the imond server or not. The imond server is responsible for monitoring/controlling the fli4l router and for the so-called least cost routing. You can find a detailed description of the [Client/Server interface imond](#) (Page 89) in a separate appendix.

Important: The least cost routing functionality of fli4l can only be used when imond is running. Time-based switching of connections is impossible without imond!

Starting with version 1.5, imond is mandatory for ISDN and DSL routing. In this case you have to set `OPT_IMOND='yes'`. If you use fli4l as a router between LANs only, you should set `OPT_IMOND='no'`.

IMOND_PORT The TCP/IP port where imond should wait for connections. You shouldn't change the default value '5000' unless in very exceptional cases.

IMOND_PASS Default setting: `IMOND_PASS=""`

This variable can be used to set a user password for imond. If a client connects to imond at port 5000, imond expects the client to provide this password before processing any requests, with the exception of the commands "quit", "help", and "pass". If you leave `IMOND_PASS` empty, no password is necessary.

The variables

- [IMOND_ENABLE](#) (Page 68),
- [IMOND_DIAL](#) (Page 68),
- [IMOND_ROUTE](#) (Page 68), and
- [IMOND_REBOOT](#) (Page 68)

control whether providing the user password is sufficient to execute the control commands like Dial, Hangup, Reboot, or Changing the Default Route, or whether you need a special admin password for these requests (see below).

IMOND_ADMIN_PASS Default setting: `IMOND_ADMIN_PASS=""`

Using the Admin Passwords the client receives all the rights and can thus use all control functions of the server imond – regardless of the content of the variables `IMOND_ENABLE`, `IMOND_DIAL` etc. If you leave `IMOND_ADMIN_PASS` empty, the user password is sufficient to gain all rights!

IMOND_LED The imond server is able to display the router's online/offline state via a LED. This LED is connected to a serial port as follows:

25 pin connector:

3. Base configuration

20 DTR ----- 1k0hm ----- >| ----- 7 GND

9 pin connector:

4 DTR ----- 1k0hm ----- >| ----- 5 GND

The LED is on if an ISDN or DSL connection is established, otherwise it is off. If you want this the other way round you have to reverse the polarity of the LED. You can reduce the dropper resistor down to 470 ohm if the LED is lit too dimly.

It is also possible to use two different coloured LEDs. In this case you have to connect the second LED together with a dropper resistor between DTR and GND too, but with reversed polarity. Then either the first or the second LED will be lit depending on the router's state. Another possibility is to use a DUO LED (two-coloured, three pins).

Currently, the serial port's RTS pin behaves exactly as the DTR pin. You could even attach a third LED for displaying the online/offline state. However, this behaviour may change in the future.

The variable `IMOND_LED` has to be set to the name of the serial port to where the LED is attached; possible values are 'com1', 'com2', 'com3', and 'com4'. Leave the variable empty if you don't use an LED.

IMOND_BEEP If setting `IMOND_BEEP='yes'`, imond will emit a two-tone sound over the PC speaker whenever the router's state changes from offline to online and the other way round. In the first case, the higher tone follows the lower one. In the second case, the higher tone is emitted before the lower one.

IMOND_LOG Default setting: `IMOND_LOG='no'`

You can set `IMOND_LOG='yes'` in order to log connections in the file `/var/log/imond.log`. This file can be copied i.e. by scp to another host e.g. for statistical purposes. However, using scp requires you to install and configure the sshd package appropriately.

The structure of the log file entries is described in Table 3.9.

The costs are denoted in Euro. These values are only meaningful if you correctly define the corresponding circuit variables `ISDN_CIRC_x_TIMES` (Page ??).

IMOND_LOGDIR If the imond log is activated, this variable can be used to choose an alternative log directory instead of the default `/var/log`, e.g. `/boot`. This is useful in order to make the log persistent on the boot medium. However, this requires the boot medium to be mounted read/write.

The default value is 'auto' which lets the fli4l router to determine the storage location automatically. Depending on further configuration, the storage path is `/boot/persistent/base` or some other path determined by the `FLI4L_UUID` variable. If neither `FLI4L_UUID` is set nor `/boot` is mounted read/write, the log file can be found under `/var/run`.

IMOND_ENABLE IMOND_DIAL IMOND_ROUTE IMOND_REBOOT These variables make certain imond commands available in user mode (enabling/disabling the ISDN interface, dialing/hanging up, changing the default route, rebooting the router).

Default settings:

3. Base configuration

Table 3.9.: Structure of Imond log files

Entry	Meaning
Circuit	the name of the circuit for which the entry has been created
Start time	the date and time of dialing this circuit
Stop time	the date and time of hanging up this circuit
Online time	the time this circuit was online
Billed time	the time for which the provider will charge you (depends on the timing)
Costs	the costs the provider will charge to your account
Bandwidth	the bandwidth used, separated into “in” and “out” (“in” coming first), presented as two unsigned integer numbers for which the following applies: Bandwidth = $4GiB * <first\ number> + <second\ number>$
Device	the device used for communication
Invoice pulse	the invoice pulse used by the provider for charging (taken from the circuit configuration)
Call charge	the fee charged per invoice pulse (taken from the circuit configuration)

```
IMOND_ENABLE='yes'
IMOND_DIAL='yes'
IMOND_ROUTE='yes'
IMOND_REBOOT='yes'
```

All other features of imond’s Client-Server interface are described in a [separate chapter](#) (Page 89).

3.14. General circuit configuration

IP_DYN_ADDR If you use connections with dynamic IP address assignment, you need to set IP_DYN_ADDR to ‘yes’, otherwise to ‘no’. Most Internet providers use dynamic IP address assignment.

Default setting: IP_DYN_ADDR=‘yes’

DIALMODE fli4l’s default dial mode is ‘auto’, i.e. fli4l dials automatically if an IP packet has to be routed to an IP address outside the LAN. However, you may also set the dial mode to ‘manual’ or ‘off’. In these cases, dialing to establish a connection is only possible using the imonc client or the Web-Interface.

Default setting: DIALMODE=‘auto’

4. Packages

Besides the BASE installation there are also packages. Each package contains one or more “OPTs”¹ which can be installed in addition to the base installation. Some of the OPTs are part of the BASE package, other have to be downloaded separately. The download site (<http://www.fli4l.de/en/download/stable-version/>) gives an overview over the packages provided by the fli4l team, the OPT database (http://extern.fli4l.de/fli4l_opt-db3/) contains packages offered by other authors. In the following, we describe the packages supplied by the fli4l team.

4.1. Tools In The Package ‘Base’

The following OPTs are contained in the BASE package:

Name	Description
OPT_SYSLOGD	Tool for logging system messages (Page 70)
OPT_KLOGD	Tool for logging kernel messages (Page 72)
OPT_LOGIP	Tool for logging WAN IP addresses (Page 72)
OPT_Y2K	Date correction utility for systems that are not Y2K-safe (Page 72)
OPT_PNP	Installation of ISAPnP tools (Page 73)
OPT_HOTPLUG_PCI	Aktivating PCI hotplugging (Page 74)

4.1.1. OPT_SYSLOGD – Logging system messages

Many programs use the Syslog interface to log messages. If you want to see these messages on your fli4l console you have to start the syslogd daemon.

Setting OPT_SYSLOGD to ‘yes’ enables debugging messages, ‘no’ disables them.

See also ISDN_CIRC_x_DEBUG (Page ??) and PPPOE_DEBUG (Page ??).

Default Setting: OPT_SYSLOGD=‘no’

SYSLOGD_RECEIVER SYSLOGD_RECEIVER controls whether fli4l can receive Syslog messages from other hosts in the network.

SYSLOGD_DEST_N **SYSLOGD_DEST_x** SYSLOGD_DEST_x describes where the system messages being received by syslogd should be displayed. Normally, this is fli4l’s console, hence:

```
SYSLOGD_DEST_1='*.* /dev/console'
```

If you want to log the messages into a file, you can e.g. use:

```
SYSLOGD_DEST_1='*.* /var/log/messages'
```

¹abbreviation for “OPTional module”

4. Packages

If you have a so-called “log host” in your network you can redirect the Syslog messages to that host if you supply its IP address.

Beispiel:

```
SYSLOGD_DEST_1='*.* @192.168.4.1'
```

The “@” sign has to be prepended to the IP address.

If you want the Syslog messages to be delivered to multiple destinations it is necessary to increase the variable `SYSLOGD_DEST_N` (number of destinations used) accordingly and to fill the variables `SYSLOG_DEST_1`, `SYSLOG_DEST_2` etc. with appropriate content.

The syntax “*.*” directs syslogd to log all messages. However, you are also able to constrain the messages to be logged for certain destinations by the use of so-called “priorities”. In this case you need to replace the asterisk (*) after the dot (.) by one of the following keywords:

- debug
- info
- notice
- warning (deprecated: warn)
- err (deprecated: error)
- crit
- alert
- emerg (deprecated: panic)

The items in the list are descending sorted according to severity. The keywords “error”, “warn”, and “panic” are deprecated—you should not use them anymore.

You can replace the asterisk (*) in front of the dot by a so-called “facility”. However, a detailed explanation is outside this scope. You can find an overview over the available facilities at the man page of `syslog.conf`:

<http://linux.die.net/man/5/syslog.conf>

In most cases an asterisk is completely sufficient. Example:

```
SYSLOGD_DEST_1='*.warning @192.168.4.1'
```

Windows hosts can serve as log hosts as well as Unix/Linux hosts. You can find links to adequate software at <http://www.fli4l.de/en/other/links/>. Using a log host is strongly recommended if you want a detailed logging protocol. The protocol is also useful for debugging purposes. The Windows client `imonc` also “understands” the Syslog protocol and is able to display the messages in a window.

Unfortunately, messages generated during the boot process cannot be directed to syslogd. However, you can configure `fli4l` to use a serial port as a terminal. You can find more information on this topic in the section [Console settings](#) (Page 29).

SYSLOGD_ROTATE You can use `SYSLOGD_ROTATE` in order to control whether Syslog message files are rotated once a day, thereby archiving the messages of the last x days.

SYSLOGD_ROTATE_DIR The optional variable `SYSLOGD_ROTATE_DIR` lets you specify the directory where the archived Syslog files should be stored. Leave it empty to use the default directory `/var/log`.

SYSLOGD_ROTATE_MAX The optional variable `SYSLOGD_ROTATE_MAX` lets you specify the number of archived/rotated Syslog files.

SYSLOGD_ROTATE_AT_SHUTDOWN With the optional variable `SYSLOGD_ROTATE_AT_SHUTDOWN` you can disable the rotate of syslog files at shutdown. Please only do this, if your syslogfiles are written directly to a destination on a permanent disk.

4.1.2. OPT_KLOGD – Logging kernel messages

Many errors, e.g. a dial-in that failed, are written directly to the console by the Linux kernel. If you set `OPT_KLOGD='yes'`, these messages are redirected to the Syslog daemon which can log them to a file or send them to a log host (see above). This keeps your fli4l console (almost) clear.

Recommendation: If you use `OPT_SYSLOGD='yes'` you should also set `OPT_KLOGD` to 'yes'.

Default setting: `OPT_KLOGD='no'`

4.1.3. OPT_LOGIP – Logging WAN IP addresses

LOGIP logs your WAN IP address to a log file. You activate this logging by setting `OPT_LOGIP` to 'yes'.

Default setting: `OPT_LOGIP='no'`

LOGIP_LOGDIR – Configure directory of log file

The variable `LOGIP_LOGDIR` contains the directory where the log file should be created or 'auto' for autodetect.

Default setting: `LOGIP_LOGDIR='auto'`

4.1.4. OPT_Y2K – Date correction for systems that are not Y2K-safe

fli4l routers are often assembled from old hardware parts. Older mainboards may have a BIOS that is not Y2K-safe. This can lead to the situation that setting the system date to the 27th May 2000 causes the BIOS date to become the 27th May 2094 after a reboot. By the way, Linux will then show the 27th May 1994 as system date.

Normally the system date reflected by fli4l is not important and should not matter at all. If you use the LCR (Least Cost Routing) functionality of your fli4l router this may very well play a role.

The reason: The 27th May 1994 was a Friday, the 27th May 2000 in contrast was a Saturday. And for the weekend there are lower-priced rates or providers, respectively ...

A first solution to that problem is as follows: The BIOS date is changed from the 27th May 2000 to the 28th May 1994 which was a Saturday, too. However, the problem is not solved completely yet: Not only does fli4l use the day of week and the current time for least-cost routing but it also respects bank holidays.

Y2K_DAYS – add N days to the system date

Because the BIOS date differs from the actual one by exactly 2191 days, the setting

```
Y2K_DAYS='2191'
```

causes the fli4l router to add 2191 days to the BIOS date before using it as the Linux system date. The BIOS date is left untouched because otherwise the year would be wrong (2094 or 1994, resp.) again after the next boot.

There is an additional alternative:

Using a time server, fli4l is able to fetch the current date and time from the Internet. The package CHRONY (Page ??) is designed for this purpose. Both settings can be combined. This is useful as it allows to correct the date via Y2K_DAYS before setting the exact time using the information from the time server.

If you do not have any problems with Y2K, set OPT_Y2K='no' and forget it ...

4.1.5. OPT_PNP – Installation of ISAPnP tools

Some ISAPnP adapters have to be configured by the “isapnp” tool. This especially affects ISDN adapters with a ISDN_TYPE of 7, 12, 19, 24, 27, 28, 30, and 106 – but only if the adapter is really an ISAPnP adapter.

For proper configuration you have to create the file “etc/isapnp.conf”.

Brief instructions to create this file follow:

- In <config>/base.txt, set OPT_PNP='yes' and MOUNT_BOOT='rw'
- boot your fli4l – the ISAPnP adapter will most likely not be detected
- Logon to the fli4l's console and type:

```
pnpdump -c >/boot/isapnp.conf
umount /boot
```

This saves the ISAPnP configuration to your boot medium.

Continue on your PC (Unix/Linux/Windows):

- Copy the file isapnp.conf from your boot medium to <config>/etc/isapnp.conf
 - Edit isapnp.conf and save your changes
- The default values can be left unchanged or be replaced by the values proposed. The relevant lines are shown in the following example:

```
#      Start dependent functions: priority acceptable
#      Logical device decodes 16 bit IO address lines
#      Minimum IO base address 0x0160
#      Maximum IO base address 0x0360
#      IO base alignment 8 bytes
#      Number of IO addresses required: 8
1)    (IO 0 (SIZE 8) (BASE 0x0160))
#      IRQ 3, 4, 5, 7, 10, 11, 12 or 15.
#      High true, edge sensitive interrupt (by default)
2)    (INT 0 (IRQ 10 (MODE +E
```

1) – Here, you can choose the I/O „BASE“ address. This address must lie between the minimum and maximum address and conform to the „base alignment“.

If your system uses more than one ISA adapter, you will have to ensure that there are no overlaps between address ranges. The address range starts at „BASE“ and ends at „BASE + Number of IO addresses required“.

2) – Here you can pick an IRQ from the list shown. The IRQs 2(9), 3, 4, 5, and 7 should be avoided as these IRQs may clash with your serial and parallel interfaces or the interrupt cascading.

ISA adapters are not able to share IRQs, thus the IRQ you choose here may not be used elsewhere.

- Put the chosen configuration (IRQ/IO) into <config>/isdn.txt
- In order to let the necessary files be copied to the boot medium, you must set OPT_PNP to 'yes' in <config>/base.txt. The variable MOUNT_BOOT can be chosen freely, however.
- Create new boot medium

The automatically generated file contains Unix line endings (LF without CR). Thus, if you use Notepad under Windows, all content is shown in a single line. In contrast, the DOS editor “edit” is able to cope with the Unix line endings. When saved, however, they are changed to DOS line endings (CR+LF).

Workaround:

- start DOS box
- change to the directory <config>/etc
- type: edit isapnp.conf
- edit file and save your changes

After that you can also use Notepad to edit the file.

Under Windows you may also use the Wordpad editor.

The CRs generated by the “edit” tool are filtered when fli4l boots and thus do not disturb.

Please try first to get along without using OPT_PNP. If the adapter is not recognized you may follow the procedure described above.

If you update to a more recent fli4l version, you may reuse the previously created isapnp.conf.

Default setting: OPT_PNP='no'

4.1.6. OPT_HOTPLUG_PCI – Aktivating PCI hotplugging

Specifying OPT_HOTPLUG_PCI='yes' copies some modules to fli4l activating PCI hotplugging and loads them during the boot process. This enables adding and removing of PCI adapters during runtime. A suitable PCI hotplug controller has to be used for this functionality.

This option does *not* have to be activated for adding and removing of virtual devices in *virtualization environments* like KVM, those use ACPI mechanisms and ACPI drivers are activated in the kernel anyway.

5. Creating the fli4l Archives/Boot media

If all configuration is completed, the fli4l archives/boot media may be created as either bootable Compact-Flash, a bootable ISO image, or only the files needed for a remote update.

5.1. Creating the fli4l Archives/Boot media under Linux or other Unix derivatives and Mac OS X

This is done by using scripts (`.sh`), which can be found in the fli4l root directory.

`mkfli4l.sh`

The Build Script recognizes the different [boot types](#) (Page 24).
The simplest call on Linux looks like this:

```
sh mkfli4l.sh
```

The actions of the build scripts are controlled by three mechanisms:

- Configuration variable `BOOT_TYPE` from `<config>/base.txt`
- Configuration file `<config>/mkfli4l.txt`
- Build-Script Parameters

The variable `BOOT_TYPE` (Page 24) decides which action of the Build Scripts is executed:

- Create a bootable fli4l CD-ISO-Image
- Generating the fli4l files needed for a remote update
- Generating the fli4l files and directly do a remote update via SCP
- a.s.o.

The description of the variables in the configuration file `<config>/mkfli4l.txt` can be found in Chapter [Control file mkfli4l.txt](#) (Page 83).

5.1.1. Command line options

The last control mechanism is appending option parameters to the call of the Build Script on the command line. The control options correspond to those in the file `mkfli4l.txt`. Option parameters override the values from the control file. Out of convenience, the names of the option parameters differ from the names of the variables from the control file. There is a long and, to some extent, a short form:

5. Creating the fli4l Archives/Boot media

Usage: mkfli4l.sh [options] [config-dir]

-c, --clean cleanup the build-directory
-b, --build <dir> set build-directory to <dir> for the fli4l-files
-h, --help display this usage
--batch don't ask for user input

config-dir set other config-directory - default is "config"

--hdinstallpath <dir> install a pre-install environment directly to
 usb/compact flash device mounted or mountable to
 directory <dir> in order to start the real installation
 process directly from that device
 device either has to be mounted and to be writable
 for the user or it has to be mountable by the user
 Do not use this for regular updates!

*** Remote-Update options

--remoteupdate remote-update via scp, implies "--filesonly"
--remoteremount make /boot writable before copying files and
 read only afterwards
--remoteuser <name> user name for remote-update - default is "fli4l"
--remotehost <host> hostname or IP of remote machine - default
 is HOSTNAME set in [config-dir]/base.txt
--remotepath <path> pathname on remote machine - default is "/boot"
--remoteport <portnr> portnumber of the sshd on remote machine

*** Netboot options (only on Unix/Linux)

--tftpbootpath <path> pathname to tftpboot directory
--tftpbootimage <name> name of the generated bootimage file
--pxesubdir <path> subdirectory for pxe files relative to tftpbootpath

*** Developer options

-u, --update-ver set version to <fli4l_version>-rev<svn revision>
-v, --verbose verbose - some debug-output
-k, --kernel-pkg create a package containing all available kernel
 modules and terminate afterwards.
 set COMPLETE_KERNEL='yes' in config-directory/_kernel.txt
 and run mkfli4l.sh again without -k to finish
--filesonly create only fli4l-files - do not create a boot-media
--no-squeeze don't compress shell scripts
--rebuild rebuild mkfli4l and related tools; needs make, gcc

An HD pre-installation of a suitably formatted (FAT16/FAT32) CompactFlash (in a USB cardreader) or an USB Stick can be done by using the option `--hdinstallpath <dir>`. You

5. Creating the fli4l Archives/Boot media

are using this script *at your own risk*. The necessary fli4l files will be copied onto the specified partition. At first, run in the fli4l directory:

```
sh mkfli4l.sh --hdinstallpath <dir>
```

This will generate the fli4l files and copy them to the CF-Card or USB Stick.

To run the next steps, you have to make sure:

- `chmod 777 /dev/brain`
- superuser rights
- installed `syslinux`
- installed `fdisk`

The script will ensure that this storage device is a FAT-partitioned USB-Drive. After that the boot loader and the files needed will be copied to the disk. You will get notified about success or failure.

After the build you have to execute the following:

```
syslinux --mbr /dev/brain

# make partition bootable using fdisk
#   p - print partitions
#   a - toggle bootable flag, specify number of fli4l partition
#       usually '1'
#   w - write changes and quit
fdisk /dev/brain

# install boot loader
syslinux -i /dev/brain
```

Now the CF resp. USB-drive should be bootable. Don't forget to unmount the device (via `umount`).

An alternative configuration directory can be specified by appending its name to the end of the command line. The normal configuration directory is called `config` and can be found under the fli4l root directory. This is where all fli4l packages place their configuration files. If you want to maintain more than one configuration, create another directory, i.e. `hd.conf`, place a copy of the configuration files there and change them according to the requirements. Here are some examples:

```
sh mkfli4l.sh --filesonly hd.conf
sh mkfli4l.sh --no-squeeze config.test
```

5.2. Creating the fli4l Archives/Boot media under Windows

Utilize the tool ‘AutoIt3’ (<http://www.autoitscript.com/site/autoit/>). This enables a ‘graphical’ edition, as well as dialogues which allow to change the variables described in the following sections.

`mkfli4l.bat`

The Build program automatically recognizes the different [boot types](#) (Page 24).

The ‘mkfli4l.bat’ can be invoked directly from Windows Explorer, if you need no optional parameters.

The actions of the Build program are controlled by different mechanisms:

- Configuration variable `BOOT_TYPE` from the `<config>/base.txt`
- Configuration file `<config>/mkfli4l.txt`
- Parameter of the build program
- Interactive settings in the GUI

The variable `BOOT_TYPE` (Page 24) decides which action the Build program executes:

- Create a bootable fli4l CD-ISO-Image
- Making the fli4l files available, for remote update
- Generating the fli4l files and direct remote update via SCP
- Hard drive pre-install of a suitably formatted CF in the Cardreader
- a.s.o.

The description of the variables in the configuration file `<config>/mkfli4l.txt` can be found in Chapter [Control file mkfli4l.txt](#) (Page 83).

5.2.1. Command line options

The last control mechanism is appending of option parameters to the call of the Build program on the command line. The control options correspond to those in the control file `mkfli4l.txt`. Option parameters override the values from the control file. Out of convenience, the names of the option parameters differ from the names of the variables from the control file. There is a long and, to some extent, a short form:

Usage: `mkfli4l.bat [options] [config-dir]`

<code>-c, --clean</code>	cleanup the build-directory
<code>-b, --build <dir></code>	sets build-directory to <dir> for the fli4l-files
<code>-v, --verbose</code>	verbose - some debug-output
<code>--filesonly</code>	creates only fli4l-files - does not create a disk
<code>--no-squeeze</code>	don't compress shell scripts
<code>-h, --help</code>	display this usage

5. Creating the fli4l Archives/Boot media

```
config-dir          sets other config-directory - default is "config"

*** Remote-Update options
--remoteupdate      remote-update via scp, implies "--filesonly"
--remoteuser <name> user name for remote-update - default is "fli4l"
--remotehost <host> hostname or IP of remote machine - default
                    is HOSTNAME set in [config-dir]/base.txt
--remotepath <path> pathname on remote machine - default is "/boot"
--remoteport <portnr> portnumber of the sshd on remote machine

*** GUI-Options
--nogui             disable the config-GUI
--lang              change language
                    [deutsch|english|espanol|french|magyar|nederlands]
```

An alternative configuration directory can be passed by appending its name to the end of the command line. The normal configuration directory is called `config` and can be found under the fli4l root directory. This is where all fli4l packages place their configuration files. If you want to maintain more than one configuration, create another directory, e.g. `hd.conf`, place a copy of the configuration files there and change it according to the requirements. Here are some examples:

```
mkfli4l.bat hd.conf
mkfli4l.bat -v
mkfli4l.bat --no-gui config.hd
```

5.2.2. Configuration dialog – Setting the configuration directory

In the main window the configuration directory setting is indicated and a window can be opened for the selection of the configuration directory.

It should be noted that any change in the 'Config-Dir' causes all options to be set to the values contained in the control file 'mkfli4l.txt' (Page ??) placed in that directory, or to the values given as command-line parameters, respectively.

If `mkfli4l.bat` does not find a directory `fli4l-x.y.z\config` or if there is no file in that directory named 'base.txt', a window is immediately opened for the selection of the configuration directory. This makes it possible to easily manage several fli4l configuration directories in a simple manner.

Example:

```
fli4l-x.y.z\config
fli4l-x.y.z\config.fd
```

5. Creating the fli4l Archives/Boot media

```
fli4l-x.y.z\config.cd  
fli4l-x.y.z\config.hd  
fli4l-x.y.z\config.hd-create
```

5.2.3. Configuration dialog – General Preferences



Figure 5.1.: Preferences

In this dialogue the settings are specified for the archive/boot-media creation:

- Build-Dir – Directory for the Archives/CD-Images/...
- BOOT_TYPE – Display of the utilized/settings BOOT_TYPE – unchangeable
- Verbose – Activation of additional output during the creation
- Filesonly – Only the archives are created – no bootmedia/no image
- Remoteupdate – Activation of the remote update via SCP

Using the button **Current settings in mkfli4l.txt** buffer the current settings can be stored in mkfli4l.txt.

5.2.4. Configuration dialog – Settings for Remote update



Figure 5.2.: Settings for Remote update

In this dialogue the settings for Remote update are specified:

- IP address or Hostname
- User name on the Remote host
- Remote path (default: /boot)
- Remote port (default: 22)
- SSH keyfile to use (format ppk from Putty)

5.2.5. Configuration dialog – Settings for HD pre-install



Figure 5.3.: Settings for HD pre-install

In this dialogue the options are set for HD pre-install on an accordingly partitioned and formatted Compact Flash card in a USB reader.

Possible Options:

- Activate HD pre-install
- Drive letter to be used to access the CF card

Regarding the partitioning and formatting of the CF: A Type-A HD installation (see package HD) must be based on a primary, active, and formatted FAT partition on the CF card. If you would like to use a data partition additionally, a Linux partition which is formatted with the ext3 file system, as well as the file `hd.cfg` are also needed on the FAT Partiton (in this case please make sure to read the documentation of the HD package).

5.3. Control file mkfli4l.txt

Since fli4l-Version 2.1.9 the control file `<config>/mkfli4l.txt` exists. This file can e.g. be used to specify directories which differ from the standard settings. The control file has a similar structure as the normal fli4l configuration files. All configuration variables here are optional, i.e. they need not exist or they can be commented out.

BUILDDIR Default: 'build'

Specifies the directory where fli4l files will be created. If the variable is undefined, the Windows mkfli4l sets it to 'build' relative to the fli4l root directory, resulting in the directory. build in the fli4l root directory:

Path/fli4l-x.y.z/build

Under *nix mkfli4l is using `<config>/build` and is thus filing the generated files together with the configuration.

The path for BUILDDIR must use the conventions of the Operating Systems Windows oder *nix. If relative paths configured there are converted by the build to the syntax of windows or *nix.

VERBOSE Default: VERBOSE='no'

Possible values are 'yes' or 'no'. Controls the *Verbosity* of the Build Processes.

FILESONLY Default: FILESONLY='no'

Possible values are 'yes' or 'no'. This will actually turn off the creation of the boot-media and only the files will be created –

REMOTEUPDATE Default: REMOTEUPDATE='no'

Possible values are 'yes' or 'no'. Enables automatic transferring of files by means of SCP to the router. This requires the package SSHD (Page ??) with activated `scp`. See also the following variables.

REMOTEHOSTNAME Default: REMOTEHOSTNAME=""

The target host name for the SCP data transfer. If no name is set, the variable [HOSTNAME](#) (Page 23) is used.

REMOTEUSERNAME Default: REMOTEUSERNAME='fli4l'

User name for the SCP data transfer.

REMOTEPATHNAME Default: REMOTEPATHNAME='/boot'

Destination path for the SCP data transfer.

REMOTEPORT Default: REMOTEPORT='22'

Destination port for the SCP data transfer.

SSHKEYFILE Default: SSHKEYFILE=""

Here you can specify a SSH key file for the SCP Remote update. Thus, an update can be made without specifying a password.

5. Creating the fli4l Archives/Boot media

REMOTEREMOUNT Default: REMOTEREMOUNT='no'

Possible values are 'yes' or 'no'. If 'yes' is set, a boot device "/boot" mounted read-only will be remounted read-write to allow remote updates of the boot files.

TFTPBOOTPATH Path where the remote Netboot image is saved to.

TFTPBOOTIMAGE Name of the Netboot image.

PXESUBDIR Subdirectory for the PXE files relative to TFTPBOOTPATH.

SQUEEZE_SCRIPTS Enable or disable the Squeezing (Compression) scripts. Compressing a script with Squeeze removes all comments and line indentations. Under normal conditions the default value of 'yes' can be used.

MKFLI4L_DEBUG_OPTION Additional debugging options can be handed over to themkfli4l-Programm (Page ??).

6. Connecting PCs in the LAN

For every host in the LAN you will have to set up:

1. IP address (see [IP address](#))
2. Name of the host plus desired domain name (see [Host and domain name](#))
3. Default gateway (see [Gateway](#))
4. IP address of the DNS server (see [DNS server](#))

6.1. IP address

The IP address of the host has to belong to the same network as the IP address of the fli4l router (on the Ethernet interface), for example 192.168.6.2 in case the router has the IP address 192.168.6.1. IP addresses have to be unique throughout the network, so it's a good idea to change (only) the last number. You will also have to make sure you specify the same IP address as specified in the file `config/base.txt`.

6.2. Host and domain name

The name of the host is for example “my-pc”, the domain “lan.fli4l”.

Important: *The domain set up on the host has to be identical to the domain set up on the fli4l if you want to use fli4l as a DNS server. Otherwise it could cause massive problems in the network.*

The reason: Windows hosts regularly search for hosts within their workgroup, trying to resolve the name WORKGROUP.my-domain.fli4l. If the domain (here: my-domain.fli4l) doesn't match the one set up on the router, fli4l will try to answer the query by forwarding it to the Internet ...

The domain has to be entered in the TCP/IP settings of the host.

6.2.1. Windows 2000

On Windows 2000 the settings can be found under:

Start ⇒

Settings ⇒

Control Center ⇒

Network and Dial-up Connections ⇒

LAN Connection ⇒

Properties ⇒

Internet protocol (TCP/IP) ⇒

Properties ⇒
Extended... ⇒
DNS ⇒
Add DNS-Suffix ⇒

Type “lan.fli4l” (or the domain set up – without “”) ⇒ Click OK.

6.2.2. NT 4.0

Start ⇒
Settings ⇒
Control Center ⇒
Network ⇒
Protocols ⇒
TCP/IP ⇒
Properties ⇒
DNS ⇒

- Enter hostname (of the client)
- Enter domain (same as in config/base.txt)
- Add IP address of fli4l router
- Add DNS suffix (add domain – see two lines above)

6.2.3. Win95/98

Start ⇒
Settings ⇒
Control Center ⇒
Network ⇒
Configuration ⇒
TCP/IP (the one that is bound to the network
interface to the router) ⇒
Properties ⇒
DNS Configuration:
Activate DNS and under “Domain:” enter “lan.fli4l” (or the domain set up – without “”)

6.2.4. Windows XP

On Windows XP the settings can be found at:

Start ⇒
Settings ⇒
System Settings ⇒
Network Connections ⇒
LAN-Connection ⇒
Properties ⇒

Internetprotocol (TCP/IP) ⇒
Properties ⇒
Advanced... ⇒
DNS ⇒
DNS-Suffix for this connection ⇒

Specify “lan.fli4l” (resp. the domain you use) (without “”!) ⇒Press OK.

6.2.5. Windows 7

On Windows 7 the settings can be found at:

Windows Button (ex. Start) ⇒
System settings ⇒
Network and Internet ⇒
Network- and Sharecenter ⇒
LAN-Connection⇒
Properties ⇒
Internetprotocol Version 4 (TCP/IPv4) ⇒
Properties ⇒
Advanced ... ⇒
DNS ⇒
DNS-Suffix for this connection ⇒

Specify “lan.fli4l” (resp. the domain you use) (without “”!) ⇒Press OK.

6.2.6. Windows 8

On Windows 8 the settings can be found at:

Press Windows- and X-key simultaneously ⇒
System settings ⇒
Network and Internet ⇒
Network- and Sharecenter ⇒
Choose your net (Ethernet or WLAN) ⇒
Properties ⇒
Internetprotocol Version 4 (TCP/IPv4) ⇒
Properties ⇒
Advanced ... ⇒
DNS ⇒
DNS-Suffix for this connection ⇒

“lan.fli4l” (bzw. die eingestellte domain) eingeben (ohne “”!) ⇒OK drücken.

6.3. Gateway

It is absolutely necessary to specify a default gateway, because without the correct IP address provided here nothing will work. So you will have to specify the IP address of the fli4l router

here (the Ethernet interface's one) – for example 192.168.6.4, depending on the IP address that has been specified in the file `config/base.txt` for the router.

It is wrong to enter `fli4l` as a proxy in the Windows or browser configuration unless you use a proxy on the router. Normally `fli4l` is not a proxy, thus please do *not* specify `fli4l` as a proxy!

6.4. DNS server

As for the IP address, you should not specify the IP address of the provider's DNS server, but the address of the router (Ethernet interface), as it will answer queries itself or forward them to the Internet if needed.

When `fli4l` is used as a DNS server, many queries from Windows hosts are not forwarded to the Internet but answered by `fli4l` itself.

6.5. Miscellaneous

The items 1 to 4 do not have to be specified when a DHCP server is configured as `fli4l` transmits the needed data automatically.

Internet options: Under connections you will have to select “do not dial”. Under settings for local network (LAN): Do NOT enter anything (unless you use `OPT_Proxy`). Both are default settings and should already exist.

7. Client/Server interface imond

7.1. imon-Server imond

imond is a network-capable server program that responds to certain queries or accepts commands that can control the router.

imond also controls the Least-Cost-Routing. It uses the configuration file `/etc/imond.conf`, that is created automatically from the variables `ISDN_CIRC_x_XXX` from the file `config/isdn.txt` and other at boot time by a shell script.

imond runs permanently as daemon and listens on TCP/IP port 5000 and the device `/dev/isdninfo`.

All possible commands that can be sent to TCP/IP port 5000:

The TCP/IP port 5000 is only reachable from the masqueraded LAN. Access from remote is blocked by the firewall configuration by default.

Imond supports two user levels: the user and the admin mode. For both levels you can set a password using `varIMOND_PASS` and/or `IMOND_ADMIN_PASS`. Then, clients are forced by imond to submit a password. As long as no password has been submitted, only the commands “pass” and “quit” are accepted. Others are rejected.

If you want to further restrict access, e.g. only allow access from a single computer, the firewall configuration has to be changed.

At present this is not possible using the standard configuration files `config/base.txt`. You will have to change the file `/etc/rc.d/rc322.masq`.

The commands

```
enable/disable/dialmode    dial/hangup    route    reboot/halt
```

Can be globally enabled/disabled using the configuration variables `IMOND_XXX` (see “Configuration”).

From a Unix/Linux computer (or a Windows computer in a DOS box) you can easily try it out: Type

```
telnet fli4l 5000           \# or the appropriate name of the fli4l-Routers
```

and you will be able to directly enter the listed commands and look at the output.

For example after entering “help” the help is shown, after “quit” the connection to imond is terminated.

7.1.1. Least-Cost-Routing – how it works

imond constructs a table (time table) from the configuration file `/etc/imond.conf` (which is created on bootup from the config variables `ISDN_CIRC_x_TIMES` and others). It contains a complete calendar week in a raster of 1 hour (168 hours = 168 Bytes). But the table only contains the circuits that have a default route defined.

Admin commands

addlink ci-index	Add channel to the circuit (channel bundling)
adjust-time seconds	Increments the date on the router by the number of seconds specified
delete filename pw	Deletes the file on the router
hup-timeout #ci-index [value]	Show or set the HUP timeout for ISDN circuits
removelink ci-index	Remove additional channel
reset-telmond-log-file	Deletes the telmond log file
reset-imond-log-file	Deletes the imond log file
receive filename #bytes pw	Transfer a file to the router. Imond acknowledges the command using an ACK (0x06). After that, the file is transfered in blocks of 1024 bytes that are also acknowledged with an ACK. Finally, imond replies with an OK.
send filename pw	If the password is correct and the file exists, imond replies with OK #bytes. Then, imond transfers the file in blocks of 1024 bytes that have to be acknowledged with an ACK (0x06). Finally, imond replies with an OK.
support pw	Shows the status/configuration of the router
sync	Syncs the cache of mounted drives

Admin or User commands

dial	Dials the provider (Default-Route-Circuit)
dialmode [auto manual off]	Shows or sets the dialmode
disable	Hangs up and sets the dialmode to “off”
enable	Sets the dialmode to “auto”
halt	Cleanly shuts down the router
hangup [#channel-id]	Hangs up
poweroff	Shuts down the router and powers it off
reboot	Reboots the router
route [ci-index]	Set the default route to circuit X (0=automatically)

User commands

channels	Shows the number of available ISDN channels
charge #channel-id	Shows the online fee for a specific channel
chargetime #channel-id	Time charged in consideration of the charge interval
circuit [ci-index]	Shows a circuit name
circuits	Shows number of default-route-circuits
cpu	Shows the CPU load in percent
date	Shows date/time
device ci-index	Shows the device of the circuit
driverid #channel-id	Shows driver-id of the channel X
help	Shows help
inout #channel-id	Shows direction (incoming/outgoing)
imond-log-file	Shows imond log file
ip #channel-id	Shows IP
is-allowed command	Shows, whether a command is configured/allowed Possible commands: dial dialmode route reboot imond-log telmond-log mgetty-log
is-enabled	Shows, whether dialmode is off (0) or auto (1)
links ci-index	Show number of channels 0, 1 or 2, 0 means: No channel bundling possible
log-dir imond telmond mgetty	Shows the log directory
mgetty-log-file	Shows mgetty logfile
online-time #channel-id	Shows online time of the current connection in hh:mm:ss
pass [password]	Show, whether it is necessary to enter a password or enter a password 1 Userpassword is set 2 Adminpassword is set 4 imond is in admin mode
phone #channel-id	Show telephone number/name of the peer
pppoe	Show the number of pppoe devices (i.e. 0 or 1)
quantity #channel-id	Show the data transferred in bytes
quit	Terminates the connection to imond
rate #channel-id	Show transfer rates (incoming/outgoing in B/sec)
status #channel-id	Show status of channel X
telmond-log-file	Shows telmond log files
time #channel-id	Show the sum of online times, format hh:mm:ss
timetable [ci-index]	Shows the time table for the LC-Routing
uptime	Shows the uptime of the router in seconds
usage #channel-id	Show the type of connection, that is available responses: Fax, Voice, Net, Modem, Raw
version	Show the protocol and program version

7. Client/Server interface imond

Using the imond command “timetable” you can have a look at it.

Here an example:

Supposing 3 circuits are defined:

```
CIRCUIT_1_NAME='Addcom'
CIRCUIT_2_NAME='AOL'
CIRCUIT_3_NAME='Firma'
```

Only the first two circuits have a default circuit defined, i.e. the corresponding variables ISDN_CIRC_x_ROUTE have the value '0.0.0.0'.

If the variables ISDN_CIRC_x_TIMES look like this:

```
ISDN_CIRC_1_TIMES='Mo-Fr:09-18:0.0388:N Mo-Fr:18-09:0.0248:Y
Sa-Su:00-24:0.0248:Y'

ISDN_CIRC_2_TIMES='Mo-Fr:09-18:0.019:Y Mo-Fr:18-09:0.049:N
Sa-Su:09-18:0.019:N Sa-Su:18-09:0.049:N'

ISDN_CIRC_3_TIMES='Mo-Fr:09-18:0.08:N Mo-Fr:18-09:0.03:N
Sa-Su:00-24:0.03:N'
```

it results in the following /etc/imond.conf being created:

#day	hour	device	defroute	phone	name	charge	ch-int
Mo-Fr	09-18	ipp0	no	010280192306	Addcom	0.0388	60
Mo-Fr	18-09	ipp0	yes	010280192306	Addcom	0.0248	60
Sa-Su	00-24	ipp0	yes	010280192306	Addcom	0.0248	60
Mo-Fr	09-18	ipp1	yes	019160	AOL 0.019	180	
Mo-Fr	18-09	ipp1	no	019160	AOL 0.049	180	
Sa-Su	09-18	ipp1	no	019160	AOL 0.019	180	
Sa-Su	18-09	ipp1	no	019160	AOL 0.049	180	
Mo-Fr	09-18	isd2	no	0221xxxxxxx	Firma	0.08	90
Mo-Fr	18-09	isd2	no	0221xxxxxxx	Firma	0.03	90
Sa-Su	00-24	isd2	no	0221xxxxxxx	Firma	0.03	90

imond creates the following time table in memory – here the output of the imond command “timetable”:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Su	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Mo	2	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2
Tu	2	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2
We	2	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2
Th	2	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2
Fr	2	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2
Sa	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

No.	Name	DefRoute	Device	Ch/Min	ChInt
1	Addcom	no	ipp0	0.0388	60
2	Addcom	yes	ipp0	0.0248	60
3	Addcom	yes	ipp0	0.0248	60

7. Client/Server interface imond

4	AOL	yes	ipp1	0.0190	180
5	AOL	no	ipp1	0.0490	180
6	AOL	no	ipp1	0.0190	180
7	AOL	no	ipp1	0.0490	180
8	Firma	no	isd2	0.0800	90
9	Firma	no	isd2	0.0300	90
10	Firma	no	isd2	0.0300	90

For circuit 1 (Addcom) there are three time ranges (1-3) defined. For circuit 2 (AOL) there are four time ranges (4-7) and for the last one there are three time ranges (8-10).

In the time table, the indices are printed that are valid in the corresponding hour. Only the indices 2-4 show up here, as the others are not default routes.

If there are zeros in the table, there are gaps in the values of the ISDN_CIRC_X_TIMES variables. At this point there is no default route, no internet access is possible!

On program start, imond checks for the weekday and the hour. Then, the index from the time table is picked out and the corresponding circuit. The default route is then set to this circuit.

If the status of a channel changes (e.g. offline – online) or at least after one minute, this procedure is repeated: check time, lookup in table, pick default route circuit.

If the used circuit changes, e.g. on Mondays, 18:00, the default route is deleted, existing connections are terminated (sorry...) and after that the default route is set to the new circuit. Imond may notice this up to 60 seconds too late – so at least at 18:00:59 the route is changed.

If a circuit does not have a default route, nothing will change. The value of ISDN_CIRC_x_TIMES is only used to calculate the fee. This can be important if the LC routing is disabled temporarily, e.g. using the client imonc, and a circuit is dialed manually.

But you can have a look at the tables for other time-range-indices (in our example 1 to 10), also at the ones of the “Non-LC-Default-Route-Circuits”.

Command:

```
timetable index
```

Example:

```
telnet fli41 5000
timetable 5
quit
```

The output will look like:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Su	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mo	5	5	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	5	5	5	5	5	5
Tu	5	5	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	5	5	5	5	5	5
We	5	5	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	5	5	5	5	5	5
Th	5	5	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	5	5	5	5	5	5
Fr	5	5	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	5	5	5	5	5	5
Sa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

No.	Name			DefRoute	Device	Ch/Min	ChInt																	
5	AOL			no	ipp1	0.0490	180																	

Got everything?

Using the command “route”, the LC routing can be enabled or disabled. If a positive circuit index is specified (1...N) the default route is changed to the circuit specified. If the index is 0, LC routing will be activated again and the active circuit is chosen automatically.

7.1.2. Annotations to the calculation of the online changes

The whole model how the online charges are calculated will only work correctly, if the charge interval for a single circuit (variable ISDN_CIRC_x_CHARGEINT) remains constant throughout the whole week.

Normally, this is correct for most of the internet providers. But if you dial in, e.g. to your companies network, using the (German) Telecom (not the internet provider T-Online), the change interval changes at 18:00 from 90 seconds to 4 minutes (information from June 2000). Because of that, the definition

```
ISDN_CIRC_3_CHARGEINT='90'  
ISDN_CIRC_3_TIMES='Mo-Fr:09-18:0.08:N Mo-Fr:18-09:0.03:N Sa-Su:00-24:0.03:N'
```

is not absolutely correct. After 18:00 the price is converted to 3 cents (4 minutes cost 12 cents), but the charge interval is wrong. Because of that, the displayed charge could differ from the actual one.

Here is a tip, how different charge intervals can be handled correctly, anyhow (also important for ISDN_CIRC_x_CHARGEINT): Just define 2 circuits – one for each charge interval. Of course you will have to adjust ISDN_CIRC_x_TIMES so that the valid circuit is always dialed, depending on the charge interval.

Once again: If you connect to an ISP you most likely will not have this problem, because the charge interval is constant all the time and only the prices per minute change (or doesn't it? I guess the German provider T-* could even introduce such a price model :-).

7.2. Windows-Client imonc.exe

7.2.1. Introduction

imond on the router and on the client imonc as a team have two use modes: the user and the admin mode. In admin mode, all controls are enabled. In user mode the variables [IMOND_ENABLE](#) (Page 68), [IMOND_DIAL](#) (Page 68), [IMOND_ROUTE](#) (Page 68) and [IMOND_REBOOT](#) (Page 68) control if the respective functions are available. If all of these variables are set to 'no', this means that all buttons in the overview page are disabled except for the exit and the admin mode button. The decision whether the user or admin mode is used, is based on the transmitted password. By clicking the button admin mode, located in the status bar, you may switch to the admin mode at any time by entering the admin password. To switch back imonc must be restarted.

Once imonc is started an additional tray icon is displayed, which shows the connection status of existing channels.

The colors mean:

Red : Offline

Yellow : A connection is in the process of being established

Light Green : Online and traffic on the channel

Dark Green : Online and (nearly) no traffic on the channel

imonc shows a behavior a little different from the Windows standard when clicking on the minimize button in the title bar. This minimizes imonc to the system tray only a tray icon near the clock remains. Double clicking on the tray icon with the left mouse button brings imonc's window back to the foreground. With the right mouse button you may use the context menu, of the tray icon to execute the main imonc commands die angezeigten without displaying its main window.

Imonc stores many properties (including all columns widths of the string grids) in the registry, so its appearance may be widely adapted to your needs. Imonc uses the registry key HKCU\Software\fli4l to store this informations

If even after careful reading of the documentation problems on imonc or the router itself still remain you can post to the newsgroup. It makes sense to note the support information you get when choosing SystemInfo and then Info on the About page of the imonc client. The router password will be queried then (not the imond password) and after that imonc will create a file fli4lsup.txt that contains all the important information regarding the router, including imonc. This file can be posted on the newsgroup when explicitly demanded and will give much better chance for quick help.

Further details concerning the development of the Windows imonc client can be found on the homepage of Windows Imonc <http://www.imonc.de/>. Here you can see what new features and bug fixes will be included in the next version. In addition, you will find the latest imonc version, if it is not included in the FLI4L distribution already.

7.2.2. Start Parameters

Imonc requires the name or IP address of the router fli4l. As the default the program attempts to establish a connection with the computer "fli4l". If this entry is correctly entered in the DNS, this should work out of the box. Otherwise you can pass following parameters in the link:

- /Server:The router's IP or hostname (short form: /S:IP or hostname)
- /Password:password (short form: /P:password)
- /log Option for logging communication between imonc und imond. When entered a file imonc.log will be written each time when imonc exits. It contains the complete communication and thus can grow quite large. Please use this parameter only in case of problems.
- /iport:Portnumber Portnumber imond listens to. Default: 5000
- /tport:Portnummer Portnumber telmond listens to. Default: 5001
- /rc:"Command" The command provided here will be transmitted to the router without further checking and imonc will exit afterwards. If more than one command should be transmitted at once, they must be devided by semicolons. You will have to provide an imond password in addition for this to work because no password query will be queried. Possible command are documented with imond, see chapter 8.1. In addition to

7. Client/Server interface imond

the commands there another one exists: timesync. If used imonc will synchronize the clock of the client with the router's clock. The command dialtimesync is not supported anymore, it is substituted by "dial; timesync".

- /d:"fli4l-Directory" Pass the fli4l-directory as a start parameter. May be of interest when using more than one fli4l version.
- /wait If the hostname can't be resolved imonc will not exit anymore – start another try by doubleclicking the tray icon.
- /nostartcheck Disables checking of imonc already running. Only makes sense to monitor several different fli4l routers in a net. When using more instances the builtin syslog- and E-Mail-capabilities will be disabled.

Usage (to be entered in the link):

```
X:\...imonc.exe [/Server:Host] [/Password:Password] [/iport:Portnumber]
                [/log] [/tport:Portnumber] [/rc:"Command"]
```

Example with IP address:

```
C:\wintools\imonc /Server:192.168.6.4
```

or with name and password:

```
C:\wintools\imonc /S:fli4l /P:secret
```

or with name and password and router command:

```
C:\wintools\imonc /S:fli4l /P:secret /rc:"dialmode manual"
```

7.2.3. Overview

The Windows client queries some imond information on the existing connections and displays it in its window. In addition to general status information such as uptime of the router or the time (both locally and on the router), for each existing connection the following informations are shown:

Status	Connection establishment/Online/Offline
Name	Telephone number of the caller or circuit name
Direction	Indicates if a connection is incoming or outgoing
IP	IP, that was assigned to the router
IBytes	Bytes received
OBytes	Bytes sent
Online time	Actual online time
Time	Sum of all online times
KTime	Sum of all online times in consideration of charge intervals
Cost	Computed costs

The data is updated every 2 seconds by default. In the context menu of this overview it is possible to copy the assigned IP to the clipboard as well as hanging up the channel (for

each available channel which is online at the moment). This is of interest in case that several different connections exist, e.g. one to surf the Internet and another to a private net and only one of them should be terminated.

If the telmond process is active on the router, imonc can show information about incoming phone calls (ie calling and called MSN) in addition. The last incoming phone call is displayed above the buttons. a log of phone calls received can be obtained by viewing the calls page.

By the six buttons in imonc the following commands can be selected:

Button	Caption	Function
1	Connect/Disconnect	Dial/Hang up
2	Add link/Rem link	Bundle channels: yes/no – only available in admin mode
3	Reboot	Reboot fli4l!
4	PowerOff	Shut down fli4l and power off afterwards
5	Halt	Shut down fli4l to power off safe afterwards
6	Exit	Exit client

The first five commands can be switched on and off individually in the router's configuration file config/base.txt for the user mode. In admin mode all are enabled. The dialmode controls the dialing behavior of the router:

Auto	The router will establish connections automatically when getting queries from the local net for the circuit concerned.
Manual	The user himself has to establish connections.
Off	No connections can be established. The dial button is deactivated.

Note that fli4l by default will dial out independently. So you never actually will have to press the connect button...

It is also possible to manually switch the default route circuit, setting the automatic LCR on and off. In the Windows version of imonc the selection list "default route" is provided for this. In addition you can configure the hangup TimeOut time directly in imonc. use the Config button next to the default route for this. All configured circuits of the router are displayed here. The value in the column Hup-timeout can be edited directly in the string grid for ISDN circuits (not yet working for DSL).

An overview over LCR can be found on the page admin/Timetable. There you'll see what circuit imond selects automatically at which time.

7.2.4. Config-Dialog

The configuration is reached using the Config button in the status bar. The window is divided into the following areas:

- The Area General:
 - Actualization Interval: Set here how often the overview should get actualized.
 - Synchronize Time on Startup: When starting the client's system time and date will get synchronized with the router's system time. You can execute this manually by using the button Synchronize on the Overview-page.
 - Start Minimized: Program will start minimized to the system tray.

7. Client/Server interface imond

- Start with Windows: Specify here if the client should start automatically with system start. Provide necessary start-parameters in the field Parameter.
- Fetch News from fli4l.de: Should news from fli4l's homepage be fetched and displayed by imonc? Then headlines are shown in the status bar. A new page News is displayed to show the complete messages.
- Logfile for Connections: The file name here is used to save connection lists locally on the imonc's system.
- TimeOut for router to answer: How long should we wait for an answer from the router before assuming that the connection has failed.
- Language: Pick the language for imoncs to use.
- Confirm Router Commands: If activated all commands influencing the router generally have to be confirmed, i.e. Reboot, Hangup ...
- Hang up even when traffic: No information should be displayed when the connection is closed and there is still traffic on the line.
- Automatic Connection to the router: Should we try to reconnect to the router in case of lost connections (i.e. when rebooting the router).
- Minimize Window To System Tray: Should imonc minimize to system tray instead of terminating itself when clicking the Exit button.
- Proxy Settings: Define a proxy for imonc's http-queries here. It will be used for fetching news.
 - Activate Proxy-support for Http-queries: Should we use a proxy
 - * Address: Address of the proxy server
 - * Port: Port number of the proxy server (default: 8080)
- TrayIcon: Set the colors of the tray icon next to the clock to your own needs. In addition you can specify that the actual dialmode will set the background color of the tray icon.
- Calls: The position of the call notification window will be saved in the registry in order to allow to set a fixed position for the window. Simply drag the window to the desired position.
 - Update: Set here in what way imonc will be informed about new calls. There are three possibilities. The first is querying the telmond service on the router in regular time intervals. A second is evaluating the syslog messages. This variant is preferred to the first - of course, the imon's syslog client has to be enabled. If imonc is used with a routed eisfair system the third possibility is to use the Capi2Text package for call signaling.
 - Delete Leading Zeros (Phone Boxes): Phone boxes often use an additional Zero to prefix the caller's number. This option will suppress the digit.
 - Own Area Code: Save your own area code here. If a call with the same area code is received it will get suppressed when displaying.
 - Telephone Book: Here, the file can be specified, in which the local Phone book is saved for resolving of the phone number. If the file does not exist, it is created by the program.

7. Client/Server interface imond

- Logfile: The file name you can specify here is used to save the call list locally on the computer. This menu item is only visible if the config variable TELMOND_LOG is set to ‘yes’ (this also applies to the call list).
- Use External Search: In this area, a program may be specified that will be called when a phone number can not be resolved using the local phone book. Info should be provided by the corresponding program. Until now there exists a connection to the telephone CD KlickTel and from Marcel Wappler a connection to the Palm database.
- Call Notification: Here can be determined whether an indication of incoming phone calls should be displayed, and how this is presented visually
 - Activate Call Notification: Indicate Calls or not.
 - Show Call Notification: Should a notification window be displayed on incoming calls? Infos: MSN called, Calling ID of the caller and date/time. Set variable OPT_TELMOND to ‘yes’ in config/isdn.txt for this to work.
 - * Suppress If no number is transmitted: Should the call notification be displayed if no calling number was transferred?
 - * Display Time: This setting specifies how long the call notification window should remain open. Setting this to “0” will avoid that the window is closed automatically.
 - * Fontsize: Sets the font size. This is of influence for the window size because it is computed based on the font size.
 - * Color: Set the font color here. I use red for better recognition.
- Phonebook: The page Phonebook contains the numbers for resolving caller IDs (MSN). The page will be shown even if the variable TELMOND_LOG is set to ‘no’ caller number resolving is also used for showing the last caller on the summary page. Alternatively a local file can be picked as phone book here.

The structure of the entry is as follows:

```
# Format:
# PhoneNumber=displayed Name[, Wavefilename]
# 0241123456789=Testuser
00=unknown
508402=Fax
0241606*=Elsa AG Aachen
```

The first three lines are comments. The fourth line accomplishes that “unknown” will be shown if no caller ID is submitted. In the fifth line the name “Fax” is assigned to MSN number 508402. In all other cases the format that will be shown is always PhoneNumber=Name. The sixth line demonstrates the possibility to define a group number. This will resolve all numbers matching the condition 0241606* to one name. Note that the first entry found in the phone book that matches will be picked. Optionally a wave-file can be set that will get played when a call with this number comes in.

As of version 1.5.2: on the page Names it is also possible to synchronize the local phone book with the router's one (stored in `/etc/phonebook`) and vice versa. The files are not simply replaced but missing entries will be added. If a phone number exists in both phone books with different name you will be prompted for the entry to be taken. Note that the synchronization of the phone book on the router is only changed in the ramdisk, so, after a reboot all changes will be lost.

- Sound: Wave-files specified here will be played when the specific event has occurred.
 - E-Mail: If E-Mail-Checking finds new E-Mails on the POP3-Server specified, the selected wave-file will be played.
 - E-Mail-Error: If an error occurs when loading E-Mails auftritt, this wave-file will be played.
 - Connection lost: When the connection to the router is gone (i.e. the router is rebooted), this wave-file will be played. If the option "Automatic Reconnect to router" is not activated a MessageBox will pop up asking you to reconnect.
 - Connection Notification: When establishing a connection this wave-file will be played.
 - Connection closed: When a connection is closed this wave-file will be played.
 - Call Notification: If Call Notification is activated this wave-file will be played on incoming calls.
 - Fax Notification: If a new FAX is received this wave-file will be played.
- E-Mails
 - Accounts: Configure your POP3-Accounts here.
 - Activate E-Mail-Check: Should E-Mail-check look for new E-Mails automatically?
 - * Check every x Min: How often should the E-Mail-check look for E-Mails automatically. Attention: a short interval can keep the router online forever! This will be the case if the interval is chosen shorter than the Hangup-Timeout of the circuit in use.
 - * TimeOut x Sec: How long should we wait for the POP3-Server until it answers? The value "0" means no timeout is in effect.
 - * Also if Router is offline: The router will perform a dialin to look for E-Mails. After checking all POP3-accounts the connection will be shut again. To use this feature the Dialmode has to set to 'auto'. Attention: If not using a flatrate additional costs will arise!
 - * Circuit to use: Which circuit should be used for checking E-Mails?
 - * Stay online afterwards: Should the connection stay until Hangup-timeout or hung up directly after E-Mail-Check.
 - * Load E-Mail-Header: Should the E-Mail-Headers be loaded instead of only querying the number of E-Mails? Loading E-Mail-Headers is a precondition for deleting E-Mails directly on the server.
 - * Notify only of new E-Mails: Should only be noted for new E-Mails acoustically and with the tray icon?

7. Client/Server interface imond

- * Start E-Mail-Client: Should the E-Mail-Client be started automatically if new E-Mails were found?
- * E-Mail-Client: Specify the E-Mail-Client to start.
- * Param: Provide additional parameters for starting the E-Mail-Client. If using Outlook as E-Mail-Client (not Outlook Express), you should set /recycle as a parameter. This will use an already existing instance of Outlook when loading new E-Mails.
- Admin
 - root-Password: Set the router password (PASSWORD in config/base.txt) here, i.e. to edit port forwarding locally and copy it back to the router.
 - Files on the router that should be displayed: All router files mentioned here can be displayed on the page admin/files easily via a mouse click. This way you can review logfiles of the routers very easy directly in imonc.
 - Edit Config files: Choose here if config files should all be opened in an editor (if the TXT-files are linked to an external editor this may lead to a huge number of open editor instances). Alternatively only the directory can be opened to give you a chance to pick the files to rework yourself.
 - DynEisfaiLog: If an account at DynEisfair exists you may set the login data here to review a logfile for the actualization of the service on the page Admin/DynEisfairLog.
- LaunchList serves for configuring the launch list (did you guess?). It will be executed after a successful connect if the option “Activate Launchlist” is activated.
 - Programs: All programs mentioned here will be started automatically when the router established a connection and the launch list is activated.
 - Activate LaunchList: Should it be executed on a successful connect?
- Traffic serves for adapting the look of the TrafficInfo window to your needs. A user reported problems with older versions of DirectX.
 - Separate Traffic-Info-Window: Should a graphical channel visualization be displayed in a separate window? In the context menu of the window you can define whether the window get the StayOnTop attribute. This causes the window to be placed on top of all other windows. This value is also saved in the registry and thus is available even after a program restart.
 - Show title bar: should the title bar of the traffic info window be displayed? It shows with which Circuit the router is online at the moment.
 - * CPU usage in title bar: Should the CPU utilization be displayed in the title bar?
 - * Online time in title bar: Should the online time of the channel also be displayed in the title bar?
 - Semi-transparent window: Should the window be transparent? This feature is available only on Windows 2000 and above.

- Colors: Define the main colors for the Traffic Information window. It should be taken into account that the DSL channel and the first ISDN channel will be assigned the same color value.
- Limits: Set the maximum transfer values for DSL here - upload and download.
- The syslog area is used to configure the display of syslog messages.
 - Activate Syslog-Client: Should imonc display syslog messages? This option be switched off if an external syslog client (for example Kiwi's Syslog Client) is used.
 - Show All Messages From: Messages should be shown from which priority on? It makes sense to start with debug priority to see which messages are interesting for you. After that you may set the priority to your preference.
 - Save Syslog Messages To A File: Should syslog messages be saved to a file in addition? Choose the messages to be logged to the file in the groupbox. The following placeholders are present:
 - %y** – will be replaced by the current year
 - %m** – will be replaced by the current month
 - %d** – will be replaced by the current day
 - Show Port Names: Should we display port names instead numbers?
 - Firewall-Messages In User Mode: Specify here whether Firewall Messages should also be shown in user mode or not.
- The Fax Area serves to configure Fax display in imonc. This area only appears if mgetty resp. faxrcv are installed on the router (OPT- packages on fli4l's homepage).
 - Fax Logfile: The filename set here is used to save Fax lists locally.
 - Local Directory: To display Faxes they have to be stored locally. Set the directory destination for this option here.
 - Actualization: There are two different ways for imonc to recognize a new Fax that has been received. Either imonc monitors the syslog messages (the syslog-client in imonc must be activated then) or it checks the logfiles in intervals. Prefer the first option if possible. If using the second option you may specify the time interval to actualize the page Fax overview. Note that this setting is not given in seconds but will be multiplied with the setting in Common/Actualization interval.
- The area grids serves for adapting the tables in imonc to your own needs. Set for each grid which columns should be shown and for grids in the area calls, connections and Faxes since what time the infos should be displayed.

7.2.5. Calls Page

The calls page is only displayed if the configuration variable `TELMOND_LOG` is set to 'yes' because no call log exists otherwise. All incoming calls that were logged while the router was working are displayed on this page. You may choose between viewing calls saved locally or on the router. When clicking on the reset button while reviewing the calls saved on the router the logfile there will be erased.

In the call overview you may right click on the number or MSN to copy it to the phone book and assign a name to it there which will shown instead from this point on.

7.2.6. Connections Page

As of version 1.4 this page displays the connections established by the router. This helps to monitor the router's behavior especially when automatic dialin is configured. `IMOND_LOG` in `config/base.txt` has to be set to 'yes' for this page to appear.

You may choose between viewing connections saved locally or on the router. When clicking on the reset button while reviewing the router's connection log it will be erased.

The following infos will be shown

- Provider
- Start date and -time
- End date und -time
- Online time
- Charged time
- Costs
- Inbytes
- Outbytes

7.2.7. Fax Page

Either `OPT_MGETTY` or `OPT_MGETTY` has to be installed on the router. You will find both on the fli4l homepage in the opt database. All incoming faxes will be listed on this page then. The context menu of the overview provides several options only available in admin mode:

- A Fax may be displayed. In Admin/Remoteupdate the fli4l directory path has to be set correctly because Faxes on the router are gzip-packed and thus need this program to exist in the path. You may also copy `gzip.exe` and `win32gnu.dll` to the `imonc` directory. If `gzip.exe` is not found at this places `imonc` tries to open the webserver of the router on the right page.
- A may be deleted. If chosen the Fax will be deleted locally and on the router (the fax file and the corresponding entry in the logfile).
- All Faxes o the router may be deleted. Files and logfile on the router are both deleted, but not from the local logfile.

You may switch between Fax overview local and on the router.

7.2.8. E-Mail Page

This page is shown only if at least one POP3-E-Mail-account is configured and activated in the config dialog.

The page E-Mail should be self-explaining. Here the E-Mail- Check is monitored. If the option “Check even if the router is offline” is not activated the E-Mail-Check will check all E-Mail-accounts for E-Mails in the specified time interval when the router is online. If the option is activated the E-Mail-Check will go online if necessary with the circuit in use at this moment and after this close the connection again. Dialmode has to be set to “auto” for this to work.

If E-Mails are found on the POP3 server vorhanden either the configured E-Mail-Client will be started or a new symbol is shown in the tray icon containing the number of E-Mails on the server. A double click will start the E-Mail-Client then. If an error occurred with one of the E-Mail-accounts a message is shown in the E-Mail-History and the E-Mail-TrayIcon shows a red colored upper right edge.

In the E-Mail-overview you may delete mails directly on the server by using the context menu (right click) without having to download them before. The cell of the E-Mail to be deleted should be selected before. Choose Delete MailMessage to perform the action.

7.2.9. Admin

This area is only visible if imonc is in admin mode.

The first item shows an overview on the circuits – resp. Internet providers – which the router can choose automatically via LCR. A double click on a provider will show the times defined for it in config/base.txt.

The second item enables you to do a remote update for the router. You may choose which from the five packages (Kernel, Root filesystem, Opt-file, rc.cfg and syslinux.cfg) should be copied to the router. To copy the update you have to specify the fli4l directory to inform imonc from where it should obtain the files needed. Also the subdirectory for the config files (default config) is needed for creating the Opt-file, rc.cfg and syslinux.cfg. A reboot should be performed after the update to enable the new configuration. If a password is queried while updating the one from config/base.txt at PASSWORD is meant here.

To avoid port forwarding only binding to exactly one client PC you may now edit the configuration directly on the router. For the change to come to effect you have to reconnect. Because the file is only edited in the ram disk all changes are lost with the next router reboot. To save your changes permanently you have to adapt the base.txt in config and update the Opt-File on the router.

The fourth item on the admin page – Files – is used for easy review of configuration and log-files simply via a mouse click. The list is configured in Config/Admin and then “files on router to view”. After that you may pick which file to show in the ComboBox on this page.

The fifth item is the page DynEisfairLog. It only appears if you provided the access data for your DynEisfair account in the Config-file. The logs of the service will be displayed then.

The last item is the Hosts page. All computers in /etc/hosts are shown here. All these will be pinged and the result is shown as well. In this way you can check if a PC is on.

7.2.10. Error, Syslog and Firewall Pages

Those pages are only visible if entries are present in the respective logs and imonc is in admin mode.

An the errors page all imonc/imond-specific errors are noted. If problems occur reviewing this page may help.

On the Syslog page all incoming Syslog messages are shown except for those of the firewall. They have an own page Firewall. In order for this to work the variable `OPT_SYSLOGD` in `config/base.txt` has to be set to 'yes'. The variable `SYSLOGD_DEST` must contain the clients IP (i.e. `SYSLOGD_DEST=@100.100.100.100` – of course with the real IP of the clients). Syslog message and according date, time, IP of the Senders and priority will be shown.

Firewall messages are displayed on an own page Firewall to be better readable. `OPT_KLOGD` must be set to 'yes' in `config/base.txt` in addition.

7.2.11. News Page

If the option is activated in imonc's config news from the fli4l homepage are shown here directly in imonc. Via http protocol the URL `http://www.fli4l.de/german/news.xml` will be loaded.

The five newest opt-packages are shown here as well. For this the URL `http://www.fli4l.de/german/imonc_opt` will be queried. In the status bar the headers of the news will be shown alternatingly.

7.3. Unix/Linux-Client imonc

There are 2 different versions for Linux: a text-based imonc) and a graphical user interface version(ximonc). The source of ximonc can be found under the directory `src`. The documentation for ximonc will only be available in the 1.5-Final-Version. An experienced Linux-User should have no problems with the source.

Let's limit to the text-based version. This is a curses based program, thus it has no graphical interface. The source lies under the directory `unix`.

Installation:

```
cd unix
make install
```

imonc will be installed to `/usr/local/bin`.

Command line parameters:

```
imonc hostname
```

hostname can be the name or the IP address of the fli4l router, e.g.

```
imonc fli4l
```

imonc shows the following:

- Date/Time of the fli4l router
- Momentarily configured route
- Default-Route-Circuits

- ISDN channels

Status : Calling/Online/Offline

Name : Phone number of the peer or the circuit-name

Time : Online time

Charge-Time : Online time considering the charge interval

Charge : The actual charge

Possible commands:

Nr	Command	Meaning
0	quit	Quit program
1	enable	Activate
2	disable	Deactivate
3	dial	Dial
4	hangup	Hang up
5	reboot	Reboot
6	timetable	Output timetable
7	dflt route	Set Default-Route-Circuit
8	add channel	Add 2. channel
9	rem channel	Remove 2. channel

Detailed information on every command:

- 0 – quit** The connection to the imond server is terminated and the program quits.
- 1 – enable** All circuits are set to dialmode “auto”. This is the default state after boot. It results in fli4l dialing automatically on demand as soon as it receives a request by a host from the LAN.
- 2 – disable** All circuits are set to dialmode “off”. This means fli4l is virtually “dead” until it is revived by the enable command.
- 3 – dial** Manual dial using the Default-Route-Circuit. You won’t need this normally as fli4l normally dials automatically.
- 4 – hangup** Manual hangup. You can make fli4l hangup before it does it automatically.
- 5 – reboot** fli4l is rebooted. Pretty unnecessary command ...
- 6 – timetable** The timetable for the Default-Route-Circuits is printed out. Example: see above.
- 7 – default route circuit** Manually changing the default route circuit can make sense, if you want to disable the automatic LC routing of fli4l for a while, as some providers will only let you access your email if you are dialed in to their servers.
- 8 – add channel** The second ISDN channel is manually added. Prerequisite: `ISDN_CIRC_x-BUNDLING` is set to ‘yes’.
- 9 – remove channel** Removes the second ISDN channel. See also “add channel”.

7. Client/Server interface imond

Apart from that, the same annotations as for the windows client `imonc.exe` apply.

A little remark: From fli4l-1.4 on, it is possible, to install a “minimalistic” imon client on the fli4l router itself using `OPT_IMONC='yes'` in package `TOOLS`.

You will be able to change some settings, e.g. routing etc. on the fli4l console locally. But Beware: This mini-imonc will only work on the fli4l router itself! On a Linux or Unix client you should always use the “big brother” `unix/imonc`.

A. Appendix for the base package

A.1. Null Modem Cable

For using the optional package PPP (Page ??) a null modem cable is needed.

It needs at least three wires. This is the pin layout:



The plugs have to be soldered with the bridges shown above.

A.2. Serial Console

fli4l can be used without monitor and keyboard. A drawback of this setup is that eventual error messages will not get noticed because not all messages can be piped to the syslog-port.

A possible solution is redirecting of all console messages to a PC or a classic terminal using the serial port of the router. Configuration is done by the variables [SER_CONSOLE](#) (Page 30), [SER_CONSOLE_IF](#) (Page 30) and [SER_CONSOLE_RATE](#) (Page 30).

Machines with older mainboards/cards don not support higher serial speeds than 38400 Bd. This is why you should try with a maximum of 38400 Bd at first before testing higher port speeds. Since only text messages are displayed on the console higher speeds are not evne necessary.

All messages that usually would go to the console are now redirected to the serial port – also messages of the boot process!

As a cable to the terminal or PC with terminal emulation a [Null Modem Cable](#) (Page 108) is used. Using a standard null modem cable is discouraged because these have bridges on the handshake wires. If Terminal or PC are powered off (or no terminal emulation is loaded) the use of a standard null modem cable can thus lead to a hangup.

This is why a special wiring is needed here for using fli4l also when the terminal is deactivated. You need a 3-wire cable, with some bridges on the plug. See [Nullmodemkabel](#) (Page 108).

A.3. Programs

To save space on the boot media the program “BusyBox” is used. It is a single executable containing the standard Unix programs

```
[, [[, arping, ash, base64, basename, bbconfig, blkid, bunzip2, bzip2,
cat, chgrp, chmod, chown, chroot, cmp, cp, ctttyhack, cut, date, dd, df,
dirname, dmesg, dnsdomainname, echo, egrep, expr, false, fdflush, fdisk, find,
findfs, grep, gunzip, gzip, halt, hdparm, head, hostname, inetd, init, insmod,
ip, ipaddr, iplink, iproute, iprule, iptunnel, kill, killall, klogd, less, ln,
loadkmap, logger, ls, lsmmod, lzcat, makedevs, md5sum, mdev, mkdir, mknod,
mkswap, modprobe, mount, mv, nameif, nice, nslookup, ping, ping6, poweroff,
ps, pscan, pwd, reboot, reset, rm, rmmmod, sed, seq, sh, sleep, sort, swapoff,
swapon, sync, sysctl, syslogd, tail, tar, test, top, tr, true, tty, umount,
uname, unlzma, unxz, unzip, uptime, usleep, vi, watch, xargs, xzcat, zcat
```

. These are mostly “minimalistic” implementations which do not cover the full functional range but fully reflect the modest requirements of fli4l.

BusyBox is GPL'ed and its source can be obtained at

<http://www.busybox.net/>

A.4. Other i4l-Tools

There are other tools for isdn4linux that could be used for fli4l. It could be that isdnlog is more adequate as a tool to compute online-fees but it's size is 10 times higher than imond's which additionally does monitoring, controlling and Least-Cost-Routing.

A.5. Debugging

Console-Outputs are most helpful for hunting bugs. But these go by so fast on the screen, don't they? Hint: SHIFT-PAGE-UP scrolls back, SHIFT-PAGE-DOWN scroll forwards.

If the error message “try-to-free-pages” occurs during router use there is not enough RAM left for the programs. Try the following options to recover:

- add more RAM
- use less Opt-Packages
- try a harddisk-installation according to [Typ B](#) (Page 11)

proc-files can help debugging, for example executing

A. Appendix for the base package

```
cat /proc/interrupts
```

shows the interrupts used by the drivers – not those used by the hardware!

More interesting files under /proc are devices, dma, ioports, kmsg, meminfo, modules, uptime, version and pci (if the router has a PCI-Bus).

Often a connection problem with ipppd is caused by failing authentication. The variables

```
OPT_SYSLOGD='yes'
```

```
OPT_KLOGD='yes'
```

in config/base.txt and

```
ISDN_CIRC_x_DEBUG='yes'
```

in config/isdn.txt can help here.

A.6. Literature

- Computer Networks, Andy Tanenbaum
- TCP/IP Netzanbindung von PCs, Craig Hunt
- TCP/IP, Kevin Washburn, Jim Evans, Verlag: Addison-Wesley, ISBN: 3-8273-1145-4
- TCP/IP Netzanbindung von PCs, ISBN 3-930673-28-2
- TCP/IP Netzwerk Administration, ISBN 3-89721-110-6
- Linux-Anwenderhandbuch, ISBN 3-929764-06-7
- TCP/IP im Detail:
<http://www.nickles.de/c/s/ip-adressen-112-1.htm>
- Generell das online Linuxanwenderhandbuch von Lunetix unter:
<http://www.linux-ag.com/LHB/>
- Einführung in die Linux-Firewall: <http://www.little-idiot.de/firewall/>

A.7. Prefixes

For units, prefixes addressed in this document are according to IEC 60027-2.

See: <http://physics.nist.gov/cuu/Units/binary.html>.

A.8. Warranty and Liability

There is no warranty and liability whatsoever for the whole fli4l distribution or parts of it. Also there is no guarantee for function or correct documentation wherever you may find it.

There is no Liability at all for eventual damages or costs that may arise! In other words: Don't complain if it eats your hamster.

A.9. Credits

In this part of the documentation all people are honored who contribute or have contributed to the development of fli4l.

A.9.1. Foundation Of The Project

Meyer, Frank

Frank started the Projekt fli4l on May, 4th 2000!

See: <http://www.fli4l.de/home/eigenschaften/historie/>

A.9.2. Developer- and Testteam

The fli4l-Team consists of (in alphabetical order):

Charrier, Bernard (*french translation*)
Eckhofer, Felix (*documentation, howtos*)
Franke, Roland (*OW, FBR*)
Hilbrecht, Claas (*VPN, kernel*)
Klein, Sebastian (*kernel, wlan*)
Knipping, Michael (*accounting*)
Krister, Stefan (*opt-Cop, lcd4linux*)
Miksch, Gernot (*LCD*)
Schiefer, Peter (*fli4l-CD, opt-cop, website, releasemanagement*)
Schliesing, Manfred (*testing*)
Schulz, Christoph (*FBR, IPv6, kernel*)
Siebmanns, Harvey (*documentation, english translation*)
Spieß, Carsten (*dsltool, hwsupp, rrdtool, webgui*)
Vosselman, Arwin (*LZS-compression, documentation*)
Weiler, Manuela (*CD-shipping, treasurer*)
Weiler, Marcel (*quality management*)
Wolters, Florian (*firmware, kernel*)

A.9.3. Developer- and Testteam (inactive)

Arndt, Kai-Christian (*USB*)
Bauer, Jürgen (*LCD-Package, fliwiz*)
Behrends, Arno (*Support*)
Blokland, Kees (*english translation*)
Bork, Thomas (*lpdsrv*)
Bußmann, Lars (*testing*)
Cerny, Carsten (*Website, fliwiz*)
Dawid, Oliver (*dhcp, uClibc*)
Ebner, Hannes (*QoS*)
Fischer, Joerg (*testing*)
Frauenhoff, Peter (*testing*)
Grabner, Hans-Joerg (*imonc*)
Grammel, Matthias (*english translation*)
Gruetzmacher, Tobias (*Mini-httpd, imond, proxy*)
Hahn, Joerg (*IPSEC*)
Hanselmann, Michael (*Mac OS X/Darwin*)
Hoh, Jörg (*Newsletter, NIC-DB, events*)
Hornung, Nicole (*Verein*)
Horsmann, Karsten (*Mini-httpd, WLAN*)
Janus, Frank (*LCD*)
Kaiser, Gerrit (*Logo*)
Karner, Christian (*PPTP-Package*)
Klein, Marcus (*Problemfeedback*)
Lammert, Gerrit (*HTML-documentation*)
Lanz, Ulf (*LCD*)
Lichtenfeld, Nils (*QoS*)
Neis, Georg (*fli4l-CD, documentation*)
Peiser, Steffen (*FAQ*)
Peus, Christoph (*uClibc*)
Pohlmann, Thorsten (*Mini-httpd*)
Raschel, Tom (*IPX*)
Reinard, Louis (*CompactFlash*)
Resch, Robert (*PCMCIA, WLAN*)
Schäfer, Harald (*HDD-Support*)
Schmitts, Jupp (*testing*)
Strigler, Stefan (*GTK-Imonc, Opt-DB, NG*)
Wallmeier, Nico (*windows-imonc*)
Walter, Gerd (*UMTS*)
Walter, Oliver (*QoS*)
Wolter, Jean (*Paketfilter, uClibc*)
Zierer, Florian (*wishlist*)

A.9.4. Sponsors

Meanwhile fli4l is a registered als Word-/trademark. The following fli4l-Users (there are more that did not want to be mentioned) have helped to raise the money needed for this:

Bebensee, Norbert
Becker, Heiko
Behrends, Arno
Böhm, Stefan
Brederlow, Ralf
Groot, Vincent de
Hahn, Olaf
Hogrefe, Paul
Holpert, Christian
Hornung, Nicole
Kuhn, Robert
Lehnen, Jens
Ludwig, Klaus-Ruediger
Mac Nelly, Christa
Mahnke, Hans-Jürgen
Menck, Owen
Mende, Stefan
Mücke, Michael
Roessler, Ingo
Schiele, Michael
Schneider, Juergen
Schönleber, Suitbert
Sennewald, Matthias
Sternberg, Christoph
Vollmar, Thomas
Walter, Oliver
Wiebel, Christian
Woelk, Fabian

For some time fli4l has its own sponsors, whose (Hardware-)donations have helped to support the fli4l development. In detail this are adapters, CompactFlash and Ethernet cards.

Hardware-donours (in alphabetical order):

Baglatzis, Stephanos
Bauer, Jürgen
Dross, Heiko
Kappenhagen, Wenzel
Kipka, Joachim
Klopfer, Tom
Peiser, Steffen
Reichelt, Detlef
Reinard, Louis
Stärkel, Christopher

More sponsors can be found on the fli4l-homepage:

<http://www.fli4l.de/sonstiges/sponsoren/>

A.10. Feedback

Critics, feedback and cooperation are always welcome.

The primary point of contact are the fli4l-Newsgroups. Those having problems in the setup of a fli4l-Routers, should at first read the FAQ, Howtos and the NG-Archives, before posting in the newsgroups. Informations on the different groups and netiquette can be found on the fli4l-Webseite:

<http://www.fli4l.de/hilfe/newsgruppen/>

<http://www.fli4l.de/hilfe/faq/>

<http://www.fli4l.de/hilfe/howtos/>

Because mostly older hardware is used for fli4l problems may be inevitable. Informations can help other fli4l-users having hardware problems with PC-Cards (I/O-Addresses, Interrupts and so on).

On fli4l's website is a link to a network card database, where appropriate drivers for certain cards are listed and can be entered.

<http://www.fli4l.de/hilfe/nic-db/>

Have fun with fli4l!

List of Figures

3.1. Packet Filter Structure	40
3.2. Directory Structure fli4l	49
5.1. Preferences	80
5.2. Settings for Remote update	81
5.3. Settings for HD pre-install	82

List of Tables

3.1. Overview of additional packages	14
3.2. Automtically generated maximum number of simultaneous connections	28
3.3. Types of network prefixes	37
3.4. Packet Filter Actions	42
3.5. Constraints For Source And Target In Paket Filter Rules	43
3.6. Packet State Constraints in Packet Filter Rules	45
3.7. Templates Included With fli4l	48
3.8. Available Conntrack Helpers In The Packet Filter	65
3.9. Structure of Imond log files	69

Index

base.txt, [14](#)
BEEP, [30](#)
BOOT_TYPE, [24](#)
BOOTMENU_TIME, [25](#)
BUILDDIR, [83](#)

COMP_TYPE_OPT, [27](#)
COMP_TYPE_ROOTFS, [27](#)
CONSOLE_BLANK_TIME, [29](#)

DEBUG_ENABLE_CORE, [31](#)
DEBUG_IP, [31](#)
DEBUG_IPTABLES, [31](#)
DEBUG_MDEV, [31](#)
DEBUG_MODULES, [31](#)
DEBUG_STARTUP, [31](#)
DIALMODE, [69](#)
DNS_FORWARDERS, [66](#)
DOMAIN_NAME, [66](#)

Example file(base.txt), [14](#)

FILESONLY, [83](#)
FLI4L_UUID, [27](#)
ftp, [65](#)

h323, [65](#)
HOSTNAME, [23](#)
HOSTNAME_ALIAS_N, [66](#)
HOSTNAME_ALIAS_x, [67](#)
HOSTNAME_IP, [66](#)

IMOND_ADMIN_PASS, [67](#)
IMOND_BEEP, [68](#)
IMOND_DIAL, [68](#)
IMOND_ENABLE, [68](#)
IMOND_LED, [67](#)
IMOND_LOG, [68](#)
IMOND_LOGDIR, [68](#)
IMOND_PASS, [67](#)

IMOND_PORT, [67](#)
IMOND_REBOOT, [68](#)
IMOND_ROUTE, [68](#)
IP_CONNTRACK_MAX, [28](#)
IP_DYN_ADDR, [69](#)
IP_NET_N, [35](#)
IP_NET_x, [35](#)
IP_NET_x_COMMENT, [37](#)
IP_NET_x_DEV, [35](#)
IP_NET_x_MAC, [36](#)
IP_NET_x_NAME, [36](#)
IP_NET_x_TYPE, [36](#)
IP_ROUTE_N, [38](#)
IP_ROUTE_x, [39](#)
irc, [65](#)

KERNEL_BOOT_OPTION, [27](#)
KERNEL_VERSION, [27](#)
KEYBOARD_LOCALE, [32](#)

LIBATA_DMA, [25](#)
LOCALE, [29](#)
LOGIP_LOGDIR, [72](#)

Masquerading, [63](#)
MKFLI4L_DEBUG_OPTION, [84](#)
MOUNT_BOOT, [25](#)

NET_DRV_N, [33](#)
NET_DRV_x, [33](#)
NET_DRV_x_OPTION, [33](#)
NET_PREFIX_x, [37](#)
NET_PREFIX_x_NAME, [37](#)
NET_PREFIX_x_STATIC_IPV4, [38](#)
NET_PREFIX_x_STATIC_IPV6, [38](#)
NET_PREFIX_x_TYPE, [37](#)
NET_PREFIX_x_ULA_DEV, [38](#)

OPT_HOTPLUG_PCI, [74](#)

- OPT_IMOND, [67](#)
- OPT_KLOGD, [72](#)
- OPT_LOGIP, [72](#)
- OPT_MAKEKBL, [32](#)
- OPT_NET_PREFIX, [37](#)
- OPT_PNP, [73](#)
- OPT_SYSLOGD, [70](#)
- OPT_Y2K, [72](#)

- PASSWORD, [23](#)
- PF_FORWARD_ACCEPT_DEF, [52](#)
- PF_FORWARD_LOG, [52](#)
- PF_FORWARD_LOG_LIMIT, [52](#)
- PF_FORWARD_N, [52](#)
- PF_FORWARD_POLICY, [52](#)
- PF_FORWARD_REJ_LIMIT, [52](#)
- PF_FORWARD_UDP_REJ_LIMIT, [52](#)
- PF_FORWARD_x, [52](#)
- PF_FORWARD_x_COMMENT, [52](#)
- PF_INPUT_ACCEPT_DEF, [50](#)
- PF_INPUT_ICMP_ECHO_REQ_LIMIT, [51](#)
- PF_INPUT_ICMP_ECHO_REQ_SIZE, [51](#)
- PF_INPUT_LOG, [51](#)
- PF_INPUT_LOG_LIMIT, [51](#)
- PF_INPUT_N, [51](#)
- PF_INPUT_POLICY, [50](#)
- PF_INPUT_REJ_LIMIT, [51](#)
- PF_INPUT_UDP_REJ_LIMIT, [51](#)
- PF_INPUT_x, [51](#)
- PF_INPUT_x_COMMENT, [51](#)
- PF_LOG_LEVEL, [50](#)
- PF_NEW_CONFIG, [39](#)
- PF_OUTPUT_ACCEPT_DEF, [53](#)
- PF_OUTPUT_CT_ACCEPT_DEF, [65](#)
- PF_OUTPUT_CT_N, [65](#)
- PF_OUTPUT_CT_x, [65](#)
- PF_OUTPUT_CT_x_COMMENT, [65](#)
- PF_OUTPUT_LOG, [53](#)
- PF_OUTPUT_LOG_LIMIT, [53](#)
- PF_OUTPUT_N, [53](#)
- PF_OUTPUT_POLICY, [53](#)
- PF_OUTPUT_REJ_LIMIT, [53](#)
- PF_OUTPUT_UDP_REJ_LIMIT, [53](#)
- PF_OUTPUT_x, [53](#)
- PF_OUTPUT_x_COMMENT, [53](#)
- PF_POSTROUTING_N, [55](#)
- PF_POSTROUTING_x, [55](#)
- PF_POSTROUTING_x_COMMENT, [55](#)
- PF_PREROUTING_CT_ACCEPT_DEF, [65](#)
- PF_PREROUTING_CT_N, [65](#)
- PF_PREROUTING_CT_x, [65](#)
- PF_PREROUTING_CT_x_COMMENT, [65](#)
- PF_PREROUTING_N, [55](#)
- PF_PREROUTING_x, [55](#)
- PF_PREROUTING_x_COMMENT, [55](#)
- PF_USR_CHAIN_N, [54](#)
- PF_USR_CHAIN_x_NAME, [54](#)
- PF_USR_CHAIN_x_RULE_N, [54](#)
- PF_USR_CHAIN_x_RULE_x, [54](#)
- PF_USR_CHAIN_x_RULE_x_COMMENT, [54](#)
- POWERMANAGEMENT, [27](#)
- pptp, [65](#)
- PXESUBDIR, [84](#)

- REMOTEHOSTNAME, [83](#)
- REMOTEPATHNAME, [83](#)
- REMOTEPORT, [83](#)
- REMOTEREMOUNT, [83](#)
- REMOTEUPDATE, [83](#)
- REMOTEUSERNAME, [83](#)
- RTC_SYNC, [26](#)

- sane, [65](#)
- SER_CONSOLE, [30](#)
- SER_CONSOLE_IF, [30](#)
- SER_CONSOLE_RATE, [30](#)
- sip, [65](#)
- snmp, [65](#)
- SQUEEZE_SCRIPTS, [84](#)
- SSHKEYFILE, [83](#)
- SYSLOGD_DEST_N, [70](#)
- SYSLOGD_DEST_x, [70](#)
- SYSLOGD_RECEIVER, [70](#)
- SYSLOGD_ROTATE, [71](#)
- SYSLOGD_ROTATE_AT_SHUTDOWN, [72](#)
- SYSLOGD_ROTATE_DIR, [71](#)
- SYSLOGD_ROTATE_MAX, [72](#)

- tftp, [65](#)
- TFTPBOOTIMAGE, [84](#)

TFTPBOOTPATH, [84](#)

TIME_INFO, [25](#)

VERBOSE, [83](#)

Y2K_DAYS, [72](#)