

fli4l – flexible internet router for linux

Version 4.0.0-trunk-x86-r60720

Frank Meyer
courriel: frank@fli4l.de

L'équipe fli4l
courriel: team@fli4l.de

23 août 2022

Table des matières

1. Documentation du paquetage base	5
1.1. Introduction	5
2. Installation et configuration	8
2.1. Décompacter les archives	8
2.2. Configuration	9
2.2.1. Éditer les fichiers de configurations	9
2.2.2. Configuration via un fichier de configuration spéciale	9
2.2.3. Variables	10
2.3. Procédures d'installation	10
2.3.1. Routeur sur une clé USB	11
2.3.2. Routeur sur CD ou par le réseau	11
2.3.3. Type A : Routeur sur disque dur – Une seule partition FAT	11
2.3.4. Type B : Routeur sur disque dur – Partition FAT et ext3	11
3. Configuration de la base	12
3.1. Exemple de fichier	13
3.2. Configuration générale	22
3.3. Configuration de la console	28
3.4. Fichier log pour la séquence de Boot et du chargement des modules	29
3.5. Réglage personnel dans opt/etc/inittab	30
3.6. Configuration du clavier	31
3.7. Pilotes des cartes réseaux Ethernet	31
3.8. Réseaux	33
3.9. Configuration du préfixe réseau	35
3.9.1. Préfixe réseau de type "stable"	36
3.9.2. Préfixe réseau de type "generated-ula"	36
3.10. Route supplémentaire (optionnel)	37
3.11. Le filtrage de paquets	37
3.11.1. Action pour le filtrage de paquets	39
3.11.2. Restriction dans les règles	40
3.11.3. Utilisation d'un modèle pour le filtrage de paquets	43
3.11.4. Configuration du filtrage de paquets	47
3.11.5. Exemples	53
3.11.6. Configuration par défaut	57
3.11.7. DMZ – Zone démilitarisée	61
3.11.8. Conntrack Helpers	61
3.12. Configuration du domaine	64
3.13. Configuration de Imond	65
3.14. Configuration du circuit général	67

4. Les paquetages	68
4.1. Outils dans le paquetage base	68
4.1.1. OPT_SYSLOGD - Enregistre tous les messages du système	68
4.1.2. OPT_KLOGD - Messages du Kernel lors du boot	70
4.1.3. OPT_LOGIP - Journalisation des adresses IP WAN	70
4.1.4. OPT_Y2K - Correctif pour avant l'année 2000	70
4.1.5. OPT_PNP - Installation des cartes ISAPnP	71
4.1.6. OPT_HOTPLUG_PCI - Activation du PCI hot-plugging	72
5. Création une archive fli4l/Média de Boot	73
5.1. Création de l'archive fli4l/Média de Boot sous Linux, dérivé Unix et Mac OS X	73
5.1.1. Lignes de commandes optionnelle	73
5.2. Création d'une archive fli4l/Média de Boot sous Windows	76
5.2.1. Ligne de commande en option	76
5.2.2. Boîte de dialogue - Définition du répertoire de configuration	77
5.2.3. Boîte de dialogue - Paramètres généraux	77
5.2.4. Boîte de dialogue - Paramètres pour la mise à jour à distance	78
5.2.5. Boîte de dialogue - Paramètres pour une pré-installation du HD	79
5.3. Paramètre pour le fichier mkfli4l.txt	80
6. Réglage des PCs dans le LAN	83
6.1. Adresse IP	83
6.2. Nom de l'ordinateur et de domaine	83
6.2.1. Windows 2000	83
6.2.2. NT 4.0	84
6.2.3. Windows 95/98	84
6.2.4. Windows XP	84
6.2.5. Windows 7	85
6.2.6. Windows 8	85
6.3. Gateway (ou Passerelle)	85
6.4. Serveur DNS	86
6.5. Divers points	86
7. Interface client/serveur imon	87
7.1. Server imon avec imond	87
7.1.1. Mode de fonctionnement du Moindre-Coût-Routage	87
7.1.2. Calcul des frais on-line (en ligne)	92
7.2. Client Windows imonc.exe	92
7.2.1. Introduction	92
7.2.2. Paramètre de démarrage	93
7.2.3. Concernant l'aperçu de imonc	94
7.2.4. Paramètres de configuration	95
7.2.5. Concernant les appels tél	100
7.2.6. Concernant les connexions	101
7.2.7. Concernant les FAX	101
7.2.8. Concernant les courriels	102
7.2.9. Admin	102

Table des matières

7.2.10. Concernant les erreurs syslog et firewall	103
7.2.11. Concernant les News	103
7.3. Client imonc pour Unix/Linux	103
A. Annexe du paquetage de Base	106
A.1. Câble Null-modem	106
A.2. Console par câble Série	106
A.3. Programmes	107
A.4. Autre outils-i4l	107
A.5. Dépannage	107
A.6. Références	108
A.7. Préfixe	108
A.8. Aucune responsabilité et de garantie	109
A.9. Merci	109
A.9.1. Fondateur du Projet	109
A.9.2. L'équipe de développeurs et de testeurs	109
A.9.3. L'équipe de développeurs et de testeurs (qui ne sont plus actifs)	110
A.9.4. Sponsor	111
A.10. Réaction	112
Table des figures	113
Liste des tableaux	114
Index	115

1. Documentation du paquetage base

1.1. Introduction

fli4l est un routeur basé sur Linux, il est capable de traiter les connexions ISDN (en france RNIS), DSL, UMTS et Ethernet, avec une petite configuration matériel : une clé USB pour booter, un processeur Intel Pentium MMX, 64 Mio de RAM, (au moins) une carte réseau Ethernet, cela est tout à fait suffisant pour créer un routeur. Les médias nécessaires pour le Boot peuvent être créés sous Linux ou sous MS Windows. Vous n'avez pas besoin de connaissances spécifiques sur Linux, mais cela est utile. Cependant, vous devez posséder quelques connaissances sur les réseaux TCP/IP, DNS et sur le routage. Pour développer vos propres extensions qui seront ajoutées à la configuration de base, vous aurez besoin d'un système Linux ainsi que des compétences Linux.

fli4l prend en charge différents médias de boot, parmi eux, les clés USB, les disques durs, les CDs et en particulier le boot par le réseau. Une clé USB est l'idéal à bien des égards : aujourd'hui, presque tous les PC peut démarrer à partir de celle-ci, elle est relativement pas cher, elle a une grande capacité et l'installation de fli4l est relativement facile sous MS Windows et Linux. En outre, elle est ouverte en écriture et peut contenir des données de configuration non volatiles (par exemple, les baux du serveur DHCP) contrairement à un CD.

- Caractéristiques générales
 - Création du média de Boot sous [Linux](#) (Page 73), [Mac OS X](#) (Page 73) et [MS Windows](#) (Page 76)
 - Configuration des fichiers via ASCII/UTF-8
 - Supporte l'IP Masquerading et le Port Forwarding
 - Least-Cost-Routing (LCR) (ou Routage à Moindre-Coût) : pour choisir automatiquement le fournisseur d'accès Internet, selon l'heure d'utilisation
 - Affiche/Calcul/Enregistre les temps de connexion et les coûts
 - Client imonc pour MS Windows/Linux converse avec imond et telmond
 - Télécharge les mis à jour des fichiers de configurations via le client imonc sous MS Windows ou via le SCP sous Linux
 - Les médias de Boot utilisent le système de fichier VFAT pour le stockage durable les données
 - Filtrage de paquets : les accès aux ports externes bloqués sont enregistrés
 - Affectation uniforme des interfaces WAN et des soi-disant Circuits
 - Utilisation possible des Circuits ISDN et DSL/UMTS en parallèle
- Fonctionnalité basic du routeur
 - Kernels Linux 3.18 ou 3.19
 - Filtrage de paquets et IP Masquerading
 - Serveur DNS local afin de réduire le nombre de requêtes DNS sur les serveurs DNS externes
 - Accessibilité à distance du serveur du démon imond pour surveiller et contrôler le Least Cost Routing (ou Moindre-Coût-Routage)

1. Documentation du paquetage base

- Accessibilité à distance du serveur du démon telmond pour les détails des appels téléphoniques entrants
- Supporte l'Ethernet
 - Pilote de carte réseau : actuellement supporte plus de 140 types de cartes
- Supporte la DSL
 - Le pilote Roaring Penguin PPPoE, supporte la connexion à la demande (peut être désactivé)
 - PPTP pour les fournisseurs DSL en Autriche et aux Pays-Bas
- Supporte l'ISDN
 - Supporte aux moins 60 types d'adaptateurs
 - Multiples possibilités de connexions ISDN : entrant/sortant/rappel, "roh"/point-to-point (ppp)
 - Regroupement de canaux : adaptation automatique de la bande passante ou activation manuelle du deuxième canal en utilisant le logiciel client sous MS Windows/Linux
- Paquetages optionnels
 - Serveur DNS
 - Serveur DHCP
 - Serveur SSH
 - Affichage online/offline par simple LED
 - Console série
 - Serveur Web minimaliste pour la surveillance des connexions RNIS et DSL ainsi que pour la reconfiguration et/ou la mise à jour du routeur
 - Droit d'accès pour configurer certains réseaux extérieur
 - Possibilité d'installer des carte PCMCIA (appelé de nos jours carte PC)
 - Enregistrement des messages du système
 - Configuration des cartes ISAPnP en l'utilisant l'outil isapnp
 - Outils supplémentaires pour le débogage (ou correction d'erreurs)
 - Configuration de l'interface série
 - Système de sauvetage avec l'administration à distance via le réseau ISDN
 - Logiciel pour afficher des informations de configurable sur un écran LCD, par exemple les taux de transmissions, la charge du CPU, etc.
 - Serveur/Routeur PPP par l'interface série
 - Modem ISDN par l'interface série
 - Serveur d'impression
 - Synchronisation de l'heure avec les serveurs de temps externe
 - Exécution des commandes définies par l'utilisateur, pour les appels téléphoniques entrants (par ex. pour composer un numéro sur Internet)
 - Supporte l'IP Aliasing (plusieurs adresses IP par interface réseau)
 - Supporte le VPN
 - Supporte l'IPv6
 - Supporte le WLAN : fli4l peut être à la fois point d'accès et client
 - Outil RRD pour la surveillance du routeur fli4l
 - Et beaucoup plus ...
- Matériels requis
 - Un processeur Intel Pentium avec le support MMX
 - 64 Mio de RAM, mieux 128 Mio
 - Une carte réseau Ethernet

1. Documentation du packaging base

- ISDN : un adaptateur supportant l'ISDN
- Une clé USB, un disque dur ATA ou d'une carte CF (qui sera accessible de la même manière qu'un disque dur ATA), il est également possible de booter à partir d'un CD
- Logiciels requis
 - Sous Linux, les programmes suivants sont demandés :
 - GCC et GNU make
 - syslinux
 - mtools (mcopy)
 - Sous MS Windows, aucun outil supplémentaire n'est demandé, fli4l apporte tout le nécessaire.

Vous avez en plus, le client imonc qui commande et affiche l'état du routeur fli4l. Ce programme est disponible pour Windows (windows/imonc.exe) et pour Linux (unix/gtk-imonc).
Et maintenant ...

Amusez-vous bien avec fli4l !

Frank Meyer et l'équipe fli4l
courriel: team@fli4l.de

2. Installation et configuration

2.1. Décompresser les archives

Sous Linux :

```
tar xvfz fli4l-4.0.0-trunk-x86-r60720.tar.gz
```

Si cela ne fonctionne pas, essayez la procédure suivante :

```
gzip -d < fli4l-4.0.0-trunk-x86-r60720.tar.gz | tar xvf -
```

Pour ceux qui veulent installer fli4l dans un répertoire existant, ils doivent utiliser le script `mkfli4l.sh -c` avant l'installation :

```
cd fli4l-4.0.0-trunk-x86-r60720
sh mkfli4l.sh -c
```

Il est toutefois recommandé d'utiliser un nouveau répertoire pour chaque nouvelle version – la configuration peut être prise en charge très simplement avec un outil servant à la comparaison des fichiers.

Sous Windows, l'archive de compression .tar peut être extraite, par exemple, avec WinZip. Il faut faire attention que les fichiers dans les sous-répertoires soient bien décompressés (voir les paramètres dans Winzip!). Il faut vérifier que l'option "Smart TAR CR conversion" est décochée dans *Options* ⇒ *Configuration*. Si cette option est cochée il peut y avoir quelques erreurs (plus ou moins important) à l'extraction des fichiers.

7-Zip (<http://www.7-zip.org/>) est un programme alternatif, il est aussi puissant que WinZip et en plus il est open source, je vous le recommande.

Les fichiers suivants sont installés dans le sous-répertoire `fli4l-4.0.0-trunk-x86-r60720/` :

- Documentation :
 - `doc/deutsch/*` Documentation Allemande
 - `doc/english/*` Documentation Anglaise
 - `doc/french/*` Documentation Française
- Configuration :
 - `config/*.txt` Fichiers de configurations, ils doivent être adaptés
- Scripts/Procédures :
 - `mkfli4l.sh` création du média de boot pour les fichiers configurés : Version-Linux/Unix
 - `mkfli4l.bat` création du média de boot pour les fichiers configurés : Version-Windows
- Kernel/Fichier de boot :
 - `img/kernel` Linux-Kernel
 - `img/boot*.msg` texte avec écran de démarrage
- Paquetage supplémentaire :
 - `opt/*.txt` Ces fichiers décrivent, la direction des programmes source et de configuration pour l'archive OPT.img.
 - `opt/...` Optionnel Module-Kernel, fichiers et programmes.

- Code source :
 - src/* Code source/outils pour Linux, voir src/README
- Programme :
 - unix/mkfli4l* Création du disque de Boot : Version-Unix/Linux
 - windows/* Création du disque de Boot : Version-Windows
 - unix/imonc* client-imond pour Unix/Linux
 - windows/imonc/* client-imond pour Windows

2.2. Configuration

2.2.1. Éditer les fichiers de configurations

Pour configurer fli4l, vous devez paramétrez dans config/*.txt les fichiers. Il est recommandé pour pouvoir comparer par la suite sa configuration ou pour pouvoir gérer plusieurs configurations, de créer une copie du répertoire config et d'effectuer la configuration dans cette copie. La comparaison des fichiers de configurations sera alors possible au moyen d'outil approprié (par ex. "diff" sous *nix) est relativement facile. Supposons que votre copie de config se trouve dans le répertoire "ma_config", vous devez d'abord aller dans le répertoire fli4l et utiliser la commande :

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

```
--- config/build/rc.cfg      2007-03-22 15:34:39.085103706 +0100
+++ ma_config/build/rc.cfg    2007-03-22 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='anonymer'
-PPPOE_PASS='surfer'
+PPPOE_USER='moi'
+PPPOE_PASS='mon mot de passe'
-OPT_SSHD='no'
+OPT_SSHD='yes'
```

On voit très bien ici, les différents paramètres qui sont configurés pour un simple routeur-DSL, même si à première vue le fichier de configuration effraie avec sa profusion de réglages.

2.2.2. Configuration via un fichier de configuration spéciale

La configuration se répartit sur différents fichiers avec le concept de module, le travail devient parfois un peu laborieux, on peut placer les fichiers de configuration dans un fichier unique *<liste config>/_fli4l.txt*. Il est plus facile de lire ou comparer son contenu que d'ouvrir la liste des fichiers *.txt un par un, mais l'on doit quand même configurer et garder les fichiers originaux pour la construction de fli4l. Pour rester sur l'exemple mentionné ci-dessus, on peut configurer un simple routeur-DSL dans ce fichier :

```
PASSWORD='3P8F3KbjYgzUc'  
NET_DRV_N='1'  
NET_DRV_1='pcnet32'  
START_IMOND='yes'  
OPT_PPPOE='yes'  
PPPOE_USER='moi'  
PPPOE_PASS='mon mot de passe'  
OPT_SSHD='yes'
```

Vous devez éviter de mélanger différente version de configuration.

2.2.3. Variables

Vous remarquerez que certaines variables sont commentées. Si c'est le cas, ils sont réduit à une information raisonnable. Cette attribution par défaut est documentée pour chaque variable. Si vous souhaitez insérer un autre commentaire pour cette variable, vous devez supprimer le commentaire et définir le votre, vous devez garder la caractère ('#') au début du commentaire.

2.3. Procédures d'installation

Dans les versions précédentes, la seule option pour fli4l était de démarrer sur une disquette. Maintenant, ce n'est plus possible pour les raisons mentionnée ci-dessus, en alternatif vous pouvez utiliser une clé USB.

Maintenant vous pouvez démarrer sur d'autre média comme par exemple (CD, HD, Réseau, Compact-Flash, DoC, ...), fli4l peut être installé sur divers médias (HD, Compact-Flash, DoC). En plus, fli4l peut être démarré de trois manières différentes :

Single Image Le Bootloader (ou chargeur automatique) charge le noyau Linux ensuite, fli4l est dans une seule image, ainsi fli4l peut être lancé sans avoir accès à aucun média de boot. Exemples d'utilisation pour les différents types de boot *integrated*, *attached*, *netboot* et *cd*.

Split Image Le Bootloader charge le noyau Linux, dans une première étape l'image rudimentaire de fli4l configure et monte sur le média. Dans une deuxième étape les fichiers restants sont chargés dans ce média à partir du média de boot. Exemple d'utilisation pour les différents types de boot *hd (Typ A)*, *ls120*, *attached*, *cd-emul*.

Installation Medium Le Bootloader charge le noyau Linux ensuite l'image rudimentaire de fli4l installe les fichiers systèmes sur le média existant, il n'a pas besoin de décompresser d'autre archive. Exemple pour une installation de disque dur avec le type B

Vous devez d'abord installer fli4l une fois dans la version minimale et ainsi acquérir de expériences. Ensuite vous pourrez utiliser fli4l comme un répondeur téléphonique ou comme un Proxy-HTTP. Vous avez ainsi l'avance d'avoir l'expérience d'avoir un routeur essentiel qui fonctionne.

Pour l'installation, nous distinguons au total quatre versions :

Clé-USB Le routeur sur une clé-USB

Lecteur-CD Le routeur sur un CD

réseau pour booter sur le réseau filaire

Installation-HD Typ A routeur sur un disque dur, CF, DoC – avec une seule partition FAT

Installation-HD Typ B routeur sur un disque dur, CF, DoC – avec une partition FAT et une partition ext3

2.3.1. Routeur sur une clé USB

Linux traite les clés USB comme des disques durs, donc les explications sont les mêmes que pour une installation sur un disque dur. Notez s'il vous plaît, que les pilotes en fonction du port USB doivent être chargés avec `OPT_USB` pour accéder à la clé USB puis avec `OPT_HDINSTALL` pour l'installation

2.3.2. Routeur sur CD ou par le réseau

Tous les fichiers nécessaires se trouvent sur le média de boot et décompactés dans un disque RAM dynamique. Dans une configuration minimale le routeur fli4l a besoin de seulement 64 Mio RAM. Pour une configuration maximale le routeur est limité que par la capacité du média de boot et la mémoire principale installée.

2.3.3. Type A : Routeur sur disque dur – Une seule partition FAT

C'est la même installation qu'avec la version CD, sauf que les fichiers sont stockés sur un disque dur. Quand nous employons le terme «Disque dur» cela signifie également d'autres dispositifs comme, un compact flash de 8 Mio et d'autres dispositifs que Linux peut traiter comme un disque dur. Depuis la version 2.1.4 fli4l peut utiliser le DiskOnChip mémoire flash de M-System et le disque SCSI.

La taille de l'archive `OPT.img` est limitée à la capacité du disque, tous les fichiers systèmes doivent être installés sur un disque RAM la taille de la RAM doit être appropriée. La consommation de RAM augmente par rapport au nombre de paquetage.

Pour une mise à jour des progiciels (c.-à-d. : mettre à jour `opt.img` et `rc.cfg` par le réseau) la partition FAT doit avoir assez de place pour le kernel, le fichier `RootFS` doit être plus ou moins égal à DEUX FOIS l'archive `OPT.img` ! Si vous voulez utiliser une option supplémentaire, encore une fois le besoin d'espace augmente par rapport à l'archive `OPT.img`.

2.3.4. Type B : Routeur sur disque dur – Partition FAT et ext3

Contrairement au type A on utilise pas de disque virtuel. Les fichiers de l'archive `OPT.img` sont copiés lors du démarrage du routeur sur la partition `ext3` et seront chargés depuis cette partition lorsque cela est nécessaire. Cette version a besoin de moins de mémoire RAM et le nombre de paquetage n'est seulement limité que par la taille du disque dur.

Pour plus d'informations sur l'installation des disques durs consultez la documentation du paquetage HD, qui sera téléchargé séparément - Pour commencer activer la variable `OPT_HDINSTALL`.

3. Configuration de la base

A partir de la version 2.0 la distribution fli4l est devenue modulaire, elle est partagée en plusieurs paquetages, ils peuvent être téléchargés séparément. Le paquetage `fli4l-4.0.0-trunk-x86-r60720.tar` contient uniquement le logiciel de base pour le routeur Ethernet. On téléchargera ensuite les paquetages dont on a besoin pour une connexion DSL, ISDN ils seront extraits dans le répertoire `fli4l-4.0.0-trunk-x86-r60720/` (!) Vous avez le choix du Kernel (ou noyau) pour le système d'exploitation fli4l, les Kernels ont été sous-traités dans des paquetages différents. Vous avez besoin au minimum du paquetage de base et d'un Kernel pour l'installation. Dans le tableau 3.1 vous trouverez un aperçu des paquetages supplémentaires.

Les fichiers utilisés pour configurer le routeur fli4l se trouvent dans le répertoire `config/` et seront décrits dans les pages suivantes de la documentation.

Ces fichiers peuvent être modifiés avec un *simple* éditeur de texte ou avec un éditeur spécialement adapté pour fli4l. Vous trouverez cette éditeur et d'autres logiciels sous Windows pour vous aidé à configurer fli4l à cette adresse

<http://www.fli4l.de/fr/telechargement/paquetages-annexes/addons/>.

Si des adaptations/extensions sont nécessaires pour des réglages spécifiques, autres que ceux décrits ci-dessus, vous aurez besoin d'installer un système linux afin d'éditer le rootf. Dans ce cas vous devriez lire le fichier README dans le répertoire `src/README`.

3. Configuration de la base

TABLE 3.1. – Aperçu des paquetages supplémentaires

Archive à télécharger	Paquetage
fli4l-4.0.0-trunk-x86-r60720	Base, nécessaire !
kernel_4_19	Kernel Linux, nécessaire !
fli4l-4.0.0-trunk-x86-r60720-doc	Documentation complète
advanced_networking	Configuration pour réseau étendue
cert	Gestion des certificats
chrony	Serveur/Client de temps
dhcp_client	Divers clients DHCP
dns_dhcp	Serveur DNS et serveur DHCP
dsl	Routeur DSL (PPPoE, PPTP)
dyndns	Supporte le service DYNDNS
easycron	Service de planification
hd	Installation sur disque dur
hwsupp	Supporte du matériel spécifique
httpd	Mini serveur Web pour le statut - information
imonc_windows	Imonc pour Windows
imonc_unix	Imonc pour GTK-Unix
ipv6	Internet Protocole Version 6
isdn	Routeur ISDN
openvpn	Supporte le VPN
pcmcia	Supporte les cartes PCMCIA
ppp	Liaison PPP sur interface série
proxy	Serveur proxy
qos	Quality of Service (ou service de qualité)
sshd	Serveur SSH
tools	Divers outils et programmes pour Linux
umts	Connexion UMTS via Internet
usb	Supporte les interfaces USB
wlan	Supporte les cartes WLAN

3.1. Exemple de fichier

L'exemple fichier de `base.txt` qui est dans le répertoire `config/` a le contenu suivant :

```
##-----
## 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 60717 2022-08-23 07:29:41Z florian $
##
```

3. Configuration de la base

```
## 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='5.4.210'  # 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. Configuration de la base

```
# 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. Configuration de la base

```
#-----
# 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. Configuration de la base

```
# 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. Configuration de la base

```
# 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. Configuration de la base

```
# 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. Configuration de la base

```
#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. Configuration de la base

```
#-----
# Domain configuration:
# settings for DNS, DHCP server and HOSTS -> see package DNS_DHCP
#-----
DOMAIN_NAME='lan.fli41'          # 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.fli41'
                                # 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. Configuration de la base

```
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
```

Ce fichier est enregistré sous le format DOS. Cela signifie, qu'à l'extrémité de chaque ligne, il y a un retour chariot (CR). J'ai décidé d'utiliser ce format car la plupart des éditeurs Unix ne rencontreront aucun problème avec. Le bloc-notes de Windows, ne peut pas manipuler ces fichiers sans CRs!

Si vous avez, des problèmes avec votre éditeur Unix/Linux favori, vous pouvez employer la commande suivante avant d'éditer le fichier au format Unix :

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

Lors de la création du support de boot, il n'y a aucune importance si le fichier contient ou pas des CRs en fin de lignes. Lorsque le fichier sera écrit sur le support de boot ou sur le disque dur, tout les CRs et tous les commentaires, seront complètement ignorés.

Maintenant nous pouvons commencer ...

3.2. Configuration générale

HOSTNAME Configuration par défaut : `HOSTNAME='fli4l'`

Tout d'abord, vous devez donner un nom au routeur fli4l.

PASSWORD Configuration par défaut : `PASSWORD='fli4l'`

3. Configuration de la base

Ici le mot de passe est nécessaire pour accéder au routeur fli4l – que ce soit par le clavier branché au routeur ou via le SSH depuis un autre ordinateur (pour cela, le paquetage `sshd` est nécessaire). Le mot de passe doit être composé d’au moins un à 126 caractères.

BOOT_TYPE Configuration par défaut : `BOOT_TYPE='hd'`

`BOOT_TYPE` dans cette variable on configure le média de boot. cette variable recherche le pilote supplémentaire (module kernel) et le script de démarrage dans `RootFS`. Voici une courte explication du processus de boot (ou démarrage) :

- Le BIOS de l’ordinateur charge le média et démarre le Bootloader.
- Le Bootloader décompacte (en règle générale `syslinux`) et commence à exécuter le kernel.
- Le kernel décompacte `RootFS` (= il contient les fichiers systèmes, les programmes, les scripts), monte `RootFS` et commence à traiter les scripts.
- Maintenant selon le `BOOT_TYPE` les modules du kernel sont chargés pour booter le média respectif, monte les partitions, décompacte l’archive `OPT (opt.img)` et charge les programmes supplémentaires.
- La configuration des différents services de fli4l commence.

Voici les options pour la variable `BOOT_TYPE` :

ls120 Démarre à partir d’un `LS120/240` ou un disque `ZIP`.

hd Démarre à partir d’un disque dur `IDE` ou `SATA`, il sont détectés directement. Pour un support `SCSI`, `USB` ou un contrôleur spéciale, le paquetage `HD` et/ou `USB` est requis pour l’installation. Pour plus informations voir la documentation (Page ??) du paquetage `hd`.

cd Démarre à partir d’un `CD-ROM`. Vous devez simplement créer une image `ISO` pour le `CD`, exemple `fli4l.iso`, qui sera ensuite graver sur le `CD` avec l’un de vos programmes préféré. Si vous avez besoins d’un pilote spécifique pour le `CD-ROM`, par exemple `SCSI`, `USB` ou un contrôleur spéciale, le paquetage `HD` et/ou `USB` est requis pour l’installation.

integrated Choisissez cette option si vous ne prévoyez pas d’utiliser un support de boot classique, mais une installation par le réseau. L’archive `OPT` est intégrée dans le `RootFS`, ainsi le kernel extrait tout à la fois et n’a pas besoin de monter un support de boot. La variable `BOOT_TYPE` est nécessaire pour une installation depuis le réseau.

Notez : la mise à jour par le réseau n’est naturellement pas possible.

attached Ce paramètre est similaire à **integrated** mais l’archive `OPT opt.img` n’est pas intégrée dans le `RootFS`. il sera copié dans le répertoire `/boot` et sera extrait pendant le processus de démarrage.

La mise en garde décrite pour **integrated** est identique ici.

netboot Ce paramètre est similaire à **integrated**. Toutefois, le script `mknetboot.sh` sera exécuté pour créer l’image, celle-ci sera exécuté sur le `LAN` (ou réseau local). S’il vous plaît lire la documentation sur Wiki <https://ssl.networks.org/wiki/display/f/fli4l+und+Netzboot> pour plus d’informations.

pxeboot Deux images seront générées, le kernel et le `rootfs.img`. Ces deux fichiers seront utilisés par le chargeur de boot `PXE`. Pendant l’exécution vous pouvez créer un répertoire `TFTP`, vous pouvez même créer un sous-répertoire `TFTP` avec (`-pxesubdir`). Vous pouvez vous référer à la documentation sur Wiki, à cette adresse : <https://ssl.networks.org/wiki/display/f/fli4l+und+Netzboot>.

3. Configuration de la base

Notez : fli4l doit être configuré comme un serveur de boot avec les paramètres (pxe/tftp) appropriés, vous trouverez de la documentation dans le paquetage `dns_dhcp`

LIBATA_DMA Avec cette variable vous pouvez désactiver le DMA pour les périphériques basés sur libata. Il est parfois nécessaire d'utiliser cette fonction, lorsque plusieurs périphérique différent sont raccordés à l'IDE, exemple un adaptateurs Compact Flash. le paramètre par défaut est : 'disabled'

MOUNT_BOOT Configuration par défaut : `MOUNT_BOOT='rw'`

Ici on règle, la manière de "monter" un média de boot. Il y a trois possibilités :

rw – Read/Write – Possibilité de lecture et d'écriture.

ro – Read-Only – Possibilité de lecture uniquement.

no – None – Le média sera démonté après le démarrage. Il pourra être enlevé si besoin.

Certaines configurations nécessitent le montage du média au démarrage avec le paramètre Read/Write, par exemple, si vous voulez si vous voulez installer un serveur DHCP ou installer un fichier log (ou journal) pour imond sur le média.

BOOTMENU_TIME Configuration par défaut : `BOOTMENU_TIME='20'`

Ici on règle le temps d'attente du Bootloader de syslinux avant de lancer automatiquement l'installation standard.

Dans le paquetage HD il y a la possibilité d'activer la fonction `OPT_RECOVER` en cas d'instabilité de la version une installation secondaire peut être générée, au cas où l'installation courante aurait un problème. Celle-ci peut être activée dans le menu boot vous pouvez choisir la version Recover.

Si vous mettez la valeur '0', le système attend que l'utilisateur active le chargement du Bootloader syslinux standard ou la version Recover que vous avez sélectionnée !

TIME_INFO Configuration par défaut : `TIME_INFO='MEZ-1MESZ,M3.5.0,M10.5.0/3'`

Les heures passent dans le monde d'Unix, elles passent aussi sous fli4l avec la norme UTC (Coordinated Universal Time) une heure unique dans le monde et qui sera convertit pour chaque localité. la variable `TIME_INFO` donne les informations nécessaires à fli4l sur les noms du fuseau horaire et règle automatiquement les heures d'été et d'hiver. Pour que cela fonctionne correctement il faut régler l'heure UTC (correspond à l'heure d'hiver de Londres). on peut utiliser pour la synchronisation le paquetage `chrony` serveur de temps (il est livré avec UTC).

On paramètre `TIME_INFO` avec les valeurs suivantes :

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

- *MEZ-1* : fuseau horaire de l'Europe centrale (*MEZ*), avec une heure d'avance *MEZ-1=UTC*.
- *MESZ* : réglage de l'heure d'été (heure d'été en Europe centrale). S'il n'y a aucune indication une heure sera ajoutée automatiquement à l'heure été.
- *M3.5.0,M10.5.0/3* : date des changements d'horaires d'été et d'hiver, le dernier dimanche d'octobre c'est le passage en heure d'hiver.

Normalement on n'a pas besoin de toucher ces valeurs à moins que l'on soit dans un autre fuseau horaire. Si vous voulez adapter ces valeurs, vous devez d'abord jeter un coup d'oeil sur les spécifications des variables d'environnements elles se trouvent à cette adresse URL (en Anglais) : http://pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap08.html

3. Configuration de la base

RTC_SYNC Paramètre par défaut : `RTC_SYNC='hwclock'`

De nombreux ordinateurs disposent d'une horloge matérielle qui est alimentée par une batterie, ainsi l'horloge continue de compter les heures pendant l'arrêt de l'ordinateur afin qu'elle soit à nouveau disponible en tant qu'horloge système au prochain démarrage. Il est important de faire la distinction entre *l'horloge système* et *l'horloge matériel* :

- *L'horloge matériel* est l'heure stockée et maintenue par l'horloge matérielle. L'heure est généralement lu à partir de l'horloge matérielle lorsque le système est démarré et adopté en tant qu'horloge système.
- *L'horloge système* est l'heure réelle que le système Linux utilise, c'est elle qui est affichée quand vous utilisez la commande `date -u`. elle est maintenu à jour par le kernel (ou noyau) Linux, elle est basé sur l'interruption matérielle régulière (Timer Interrupt), elle désigne toujours l'heure en utilisant le temps universel coordonné (UTC) et n'est pas influencé par le fuseau horaire terrestre.
- *L'horloge système localisée* est juste la conversion de l'heure système en appliquant l'heure du fuseau horaire terrestre, elle est configuré sur le routeur fli4l via la variable d'environnement `TZ` (voir la variable [TIME_INFO](#) (Page 24)), elle ne joue aucun rôle dans la suite de cette section.

Cette variable informe comment fli4l calibre l'horloge matériel avec l'horloge du système, c'est-à-dire à quelle fréquence l'horloge matériel doit être réglée sur l'horloge système. Un tel ajustement est nécessaire, car même la meilleure horloge matérielle n'est pas précise à 100 % et tend à dériver systématiquement, c'est-à-dire que l'horloge sera légèrement trop lente ou trop rapide à long terme.

Il y a essentiellement deux façons de synchroniser l'heure :

- Mode "Kernel" : le client NTP est utilisé pour déterminer l'heure réelle extérieur (généralement via Internet ou par une horloge (radio) externe) et pour maintenir à jour l'horloge système du routeur fli4l. Le kernel Linux est chargé de mettre à jour l'horloge matériel, de sorte qu'aucune synchronisation supplémentaire ne soit nécessaire. La mise à jour par le kernel Linux est légèrement moins précise par rapport à l'utilisation de `hwclock` (voir le mode "hwclock" ci-dessous), cependant, la qualité de mise à jour est beaucoup moins importante, en raison des erreurs inévitables qui seront compensées par le client NTP.

Ce mode doit également être utilisé s'il n'existe aucune horloge matérielle. Bien sûr, le kernel Linux ne mettra pas à jour l'horloge matérielle car il n'y en a pas. Cependant, le client NTP doit être utilisé pour s'assurer que le routeur fli4l dispose d'une horloge système raisonnable.

- Mode "Hwclock" : en utilisant le programme `hwclock` l'horloge se synchronise à intervalle régulière (toutes les 24 heures) et à l'arrêt du système (avec l'exécution du script `stop /etc/rc0.d/rc950.hwclock`). Non seulement l'horloge matérielle est définie, mais `hwclock` mesure également la différence entre l'horloge système et l'horloge matérielle. Lors du démarrage du système, l'horloge système n'est pas prise en compte directement à partir de l'horloge matérielle, mais l'écart entre les deux sera pris en compte afin de réduire au maximum la dérive de l'horloge système. Cette écart est noté dans le fichier `/etc/adjtime`. Si un support réinscriptible est installé, l'écart sera stocké sous `/var/lib/persistent/base/adjtime`, dans ce cas, `/etc/adjtime` sera un lien symbolique.

Ce mode est incompatible avec le client NTP pour la mise à jour de l'horloge système.

3. Configuration de la base

En effet, le client NTP permet la mise à jour automatique de l'horloge matérielle par le kernel Linux. Il n'est pas logique que `hwclock` et le kernel Linux essayent de mettre à jour en même temps l'horloge matériel.

Veuillez noter que si une horloge matérielle est disponible (avec une batterie), l'heure stockée sera *toujours* interprétée comme le temps universel coordonné (UTC). Le fuseau horaire défini dans la variable `TIME_INFO` n'affecte pas l'heure stockée dans l'horloge matérielle. Le stockage d'une heure non UTC localisée par l'horloge matérielle n'est *pas* supporté par `fli4l`.

L'établissement de l'horloge système à partir de l'horloge matérielle est effectuée une fois que le système est démarré. Le kernel Linux lit l'horloge matérielle et définit l'horloge système immédiatement au début du processus de démarrage. En mode "`hwclock`", l'horloge système sera définie plus tard lors de l'exécution du script de démarrage `/etc/rc.d/rc100.hwclock`, cette fois en tenant compte de l'écart systématique.

KERNEL_VERSION Ici on détermine la version du kernel (ou noyau) à utiliser, cette variable doit correspondre au kernel `img/kernel-<kernel version>.<extension compression>`, on peut voir la version du Kernel dans le répertoire `opt/lib/modules/<kernel version>`.

KERNEL_BOOT_OPTION Configuration par défaut : `KERNEL_BOOT_OPTION=""`

Ici vous pouvez ajouter les variables en ligne de commande pour le kernel, ils seront rajoutées dans le fichier `syslinux.cfg`. Par exemple certain système on besoin pour rebooter correctement d'ajouter `'reboot=bios'`, avec un système WRAP vous pouvez ajouter `'nokdb reboot=bios'`.

COMP_TYPE_ROOTFS Configuration par défaut : `COMP_TYPE_ROOTFS='xz'`

Le contenu de cette variable détermine la méthode de compression pour l'archive RootFS. Les valeurs possibles sont `'xz'`, `'lzma'` et `'bzip2'`.

COMP_TYPE_OPT Configuration par défaut : `COMP_TYPE_OPT='xz'`

Le contenu de cette variable détermine la méthode de compression pour l'archive OPT. Les valeurs possibles sont `'xz'`, `'lzma'` et `'bzip2'`.

POWERMANAGEMENT Configuration par défaut : `POWERMANAGEMENT='acpi'`

Le kernel supporte différents formats de gestion d'énergie, l'APM qui est un peu âgé et l'actuel ACPI. Vous pouvez placer ici le format que vous voulez utiliser. Les valeurs possibles sont : `'none'` (aucune gestion d'énergie), `'ACPI'` et les deux variantes de APM `'apm'` et `'apm_rm'`. Ce dernier commute en mode processeur spécial, avant que le routeur s'arrête.

FLI4L_UUID Configuration par défaut : `FLI4L_UUID=""`

Vous pouvez indiquer dans cette variable un UUID (ou IDentifiant Universellement Unique), dans lequel `fli4l` pourra enregistrer des données persistantes par exemple sur une clé USB. Cette UUID peut être créé avec n'importe quel Système Linux (et aussi avec `fli4l`) avec la commande `'cat /proc/sys/kernel/random/uuid'` Chaque exécution de la commande produit un nouvel UUID que vous devez entrer dans la variable. Sur un support persistant (par exemple, un disque dur (`OPT_HD`) ou une clé USB (voir paquetage `OPT_USB` et `OPT_HD`) vous devez créer un répertoire avec le même nom que l'UUID. Ce répertoire sera utilisé pour stocker les changements de configuration ainsi que les données d'exécution persistante (par ex. pour le `dhcp leases` (ou baux DHCP). Naturellement les paquetages correspondants pour ce nouveau mécanisme doivent être

3. Configuration de la base

supportés (voir leur documentation). Le paramètre pour sauvegarder le chemin sera généralement 'auto'.

Si vous avez installé fli4l avant d'utiliser l'application UUID et que des données sont déjà stockées dans le répertoire fli4l, vous devez naturellement déplacer ces données dans le nouveau répertoire /boot/persistant. Il est recommandé par conséquent de configurer l'UUID à l'installation de fli4l pour éviter de déplacer les données.

En outre vous ne devez pas paramétrer la variable comme ceci `MOUNT_BOOT='ro'`, tant que l'emplacement de stockage fait partie de la partition /boot.

Un endroit recommandé pour le répertoire persistant est situé dans la partition /data (niveau supérieur) ou sur une clé USB. de la clé USB doit être de type VFAT ou activer le fichier système pour OPT_HD avec les autorisations en écriture et lecture.

IP_CONNTRACK_MAX Configuration par défaut : `IP_CONNTRACK_MAX=""`

Avec cette variable, vous pouvez régler manuellement la quantité maximum de connexions simultanées. Normalement une valeur rationnelle est trouvée automatiquement par rapport à la mémoire vive installée. Le tableau 3.2 représente la configuration par défaut.

TABLE 3.2. – Réglage Automatique du nombre de connexions maximum

Mémoire RAM Mio	Connexions simultanées
16	1024
24	1280
32	2048
64	4096
128	8192

Si vous utilisez sur le routeur des programmes de partage de fichiers en arrière ou si le routeur a peu de RAM. Le nombre maximal de connexions simultanées sera rapidement atteint et les connexions supplémentaires ne pourront plus être développées.

Cela se traduit par un message erreur qui s'affichera :

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

Autre message

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

Maintenant au moyen de la variable `IP_CONNTRACK_MAX` vous pouvez régler précisément la valeur du nombre maximum de connexions simultanées. Cependant vous devez savoir. Pour chaque connexions simultanées cela coûte 350 Octets de mémoire RAM en moins, qui ne seront plus utilisés pour autre chose. Si vous indiquez 10000, on perd à peu près 3,34 Mo de mémoire RAM pour l'utilisation (du kernel, de Ramdisks et des programmes). Avec 32 Mio de RAM, il ne devrait pas y avoir de problème, pour la table `ip_conntrack` 2 ou 3 Mio seront réservés, voir le tableau. Avec 16 Mio de RAM c'est juste, mais avec 12 ou même 8 Mio on est sur d'avoir un message erreur.

Le réglage en cours d'utilisation peuvent être affichées sur la console en tapant

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

et peut être modifié à la volée en tapant

3. Configuration de la base

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

"XXX" indique la quantité de connexions simultanées à entrer. Vous pouvez afficher sur la console, le nombre de connexion de la variable `IP_CONNTRACK` en tapant

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

Pour voir les détails

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

LOCALE Configuration par défaut : `LOCALE='de'`

Certains composants sont devenus entre-temps multi langues. Par exemple, le menu de l'interface Web. Avec cette variable, vous pouvez choisir votre langue préférée. Différents composants ont leur propre paramètre de base, avec ce réglage le paramètre sera tronqué, si la langue indiquée n'est pas (encore) disponible pour ces composants, l'anglais sera la langue par défaut.

Si la variable est sur `KEYBOARD_LOCALE='auto'` on utilise le clavier commun à la langue qui est indiquée dans la variable `LOCALE`.

Les réglages suivants sont possibles : de, en, es, fr, hu, nl.

3.3. Configuration de la console

fli4l peut être exécuté sur différentes plates-formes matérielles. Sur bon nombre de ces plates-formes, il est possible de connecter un clavier et un moniteur pour interagir avec fli4l, cette combinaison d'entrées et de sorties est généralement appelée *console*.

fli4l peut également être utilisé sans clavier ni carte graphique. Si vous voulez voir les messages de démarrage du noyau (kernel) du routeur et si vous n'avez pas de connexion réseau, il est possible d'utiliser une console distante pour recevoir les entrées et sorties en passant par l'interface série. Pour cela, il est nécessaire de paramétrer les variables suivantes [SER_CONSOLE](#) (Page 29), [SER_CONSOLE_IF](#) (Page 29) et [SER_CONSOLE_RATE](#) (Page 29)

Enfin, vous pouvez utiliser en parallèle une console avec clavier et moniteur et aussi utiliser l'interface série.

En général, fli4l offre la possibilité de se connecter à *n'importe* quelle console et donc au *Shell* (interpréteur de commandes), vous pouvez vous connecter avec le nom d'utilisateur "fli4l" et le mot de passe configuré dans la variable [PASSWORD](#) (Page 22)

CONSOLE_BLANK_TIME Configuration par défaut : `CONSOLE_BLANK_TIME=""`

Lorsque vous n'utilisez pas la console du kernel Linux (de fli4l) pendant un certain temps, normalement l'économiseur d'écran s'active. Avec la variable `CONSOLE_BLANK_TIME` on peut désactiver complètement le mode économiseur d'écran, avec le paramétrage (`CONSOLE_BLANK_TIME='0'`).

BEEP Configuration par défaut : `BEEP='yes'`

Signale sonore au démarrage et à l'arrêt de fli4l.

Si vous placez 'yes' dans cette variable, un signal sonore retentira au démarrage et à l'arrêt du processus. S'il manque de la place sur le média de boot et aussi pour gagner quelques octets, ou si vous ne voulez pas que le signal sonore soit émit, vous pouvez indiquer 'no'.

3. Configuration de la base

SER_CONSOLE Configuration par défaut : `SER_CONSOLE='no'`

Cette variable active ou désactive la console sur le port série. La console série peut être configurée en trois modes différents :

SER_CONSOLE	Entrée/Sortie sur la console
no	Entrée et sortie (uniquement) par le clavier et le moniteur (tty0)
yes	Entrée et sortie (uniquement) par l'interface série (ttyS0)
primary	Entrée et sortie par la console série ainsi que par le clavier et le moniteur, sortie des messages du noyau sur tty0
secondary	Entrée et sortie par la console série ainsi que par le clavier et le moniteur, sortie des messages du noyau sur ttyS0

Si la valeur `SER_CONSOLE` est modifiée, cette modification ne prendra effet lors de la création d'un nouveau support de démarrage ou lors de la mise à jour à distance du fichier `syslinux.cfg`.

Important: *Lorsque vous coupez la console série, veillez à maintenir un accès alternatif au routeur avec (le SSH ou directement à partir du clavier et du moniteur) !*

Vous trouverez des informations complémentaires en cliquant sur [Console serie](#) (Page 106).

SER_CONSOLE_IF Configuration par défaut : `SER_CONSOLE_IF='0'`

Numéro de l'interface série pour la console série.

Vous indiquez dans cette variable le numéro d'interface sur laquelle la console série est connectée. 0 correspond à `ttyS0` sous Linux ou `COM1` sous Microsoft Windows.

SER_CONSOLE_RATE Configuration par défaut : `SER_CONSOLE_RATE='9600'`

Vitesse de transmission de l'interface série pour la console.

Ici vous indiquez la vitesse en Baud avec laquelle les données seront transmises sur l'interface série. Les valeurs sont : 4800, 9600, 19200, 38400, 57600, 115200.

3.4. Fichier log pour la séquence de Boot et du chargement des modules

`fl4l` écrit l'ensemble du processus de boot (ou démarrage) dans le fichier (`/var/tmp/boot.log`). Ce fichier, peut être vu à la fin du processus de boot sur la console ou sur l'interface-Web dans menu correspondant.

Il est parfois utile en cas de problème, de générer des traces détaillées de la séquence de boot, pour ensuite examiner le processus de boot plus en détail. On utilise pour cela la variable `DEBUG_STARTUP`. Dans certaines situations les développeurs ont besoin d'autres paramètres pour les aider à résoudre des erreurs, ces paramètres supplémentaires sont documentés dans cette section.

DEBUG_STARTUP Configuration par défaut : `DEBUG_STARTUP='no'`

Si la valeur est sur `'yes'`, chaque commande exécutée est écrite sur l'écran de contrôle pendant le boot. Comme un changement dans le fichier `syslinux.cfg` est nécessaire pour

3. Configuration de la base

l'activation de cette fonctionnalité, c'est aussi valable pour la variable `SER_CONSOLE`. Vous pouvez adapter le fichier `syslinux.cfg` manuellement en ajoutant `fli4ldebug=yes`. Toutefois `DEBUG_STARTUP` doit être placé malgré tout sur 'yes'.

DEBUG_MODULES Configuration par défaut : `DEBUG_MODULES='no'`

Certains modules du Kernel sont chargés automatiquement, sans pouvoir les détecter à l'avance. Si vous activez la variable `DEBUG_MODULES='yes'` vous pouvez voir entièrement la séquence de chargement de ces modules, qu'ils soient chargés par un script ou émis par le Kernel.

DEBUG_ENABLE_CORE Configuration par défaut : `DEBUG_ENABLE_CORE='no'`

Si vous activez cette variable, tout accident causé sur le routeur créera un soi-disant fichier-"core", C'est une image mémoire du processus qui est enregistrée juste avant le crash. Ce fichier se trouve sur le routeur dans `/var/log/dumps`. Ce fichier peut ensuite être utilisé pour trouver plus facilement le bug du programme. Pour plus de détails, reportez-vous dans la section "programme de débogage sur fli4l" (Page ??) dans la documentation du paquetage SRC.

DEBUG_MDEV Configuration par défaut : `DEBUG_MDEV='no'`

Si la variable `DEBUG_MDEV='yes'` est activée, toutes les actions qui sont en rapport avec les Démons-mdev, sur l'ajout ou la suppression de nœud de périphériques dans `/dev` ou encore au chargement d'un firmware, seront consignées dans le fichier `/dev/mdev.log`.

DEBUG_IPTABLES Configuration par défaut : `DEBUG_IPTABLES='no'`

Si la variable `DEBUG_IPTABLES='yes'`, est activée, tous les appels-iptables y compris les valeurs de retour seront consignés dans le fichier `/var/log/iptables.log`.

DEBUG_IP Configuration par défaut : `DEBUG_IP='no'`

Si vous activez la variable `DEBUG_IP='yes'` tous les requêtes vers le programme `/sbin/ip` seront consignés dans le fichier `/var/log/wrapper.log`.

3.5. Réglage personnel dans `opt/etc/inittab`

On peut lancés au démarrage du système des programmes supplémentaires, ou ajouter des commandes supplémentaires à partir de la console ou changer les commandes standard dans le fichier de configuration `inittab`. Voici une description :

`device:runlevel:action:command`

device est le périphérique, sur lequel le programme doit faire ses Entrées/Sorties. Pour les terminaux normaux `tty1` `tty4` ou pour les terminaux serie `ttyS0` `ttySn` avec $n < \text{le numéro du ports serie}$.

action décrit l'action à exécuter comme par exemple *askfirst* ou *respawn*. *askfirst* fonctionne comme *respawn* à la différence prêt qu'il demande à l'utilisateur d'appuyer sur une touche avant l'exécution d'un programme. *respawn* permet d'exécuter automatiquement un programme à la fin de l'initialisation.

command est le programme qui doit être exécuté. On doit spécifier le chemin d'accès complet.

Voici la documentation de Busybox <http://www.busybox.net> le site contient une description exacte du format `inittab`.

Cela pourrait ressembler à ce qui suit :

3. Configuration de la base

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

On pourrait par exemple rajouter ceux-ci

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

Pour obtenir un deuxième login sur le terminal numéro deux. Il suffit simplement de rechercher le fichier `opt/etc/inittab` puis de copier la <ligne de config> ci-dessus dans le fichier `/etc/inittab` avec un éditeur de texte.

3.6. Configuration du clavier

KEYBOARD_LOCALE Configuration par défaut : `KEYBOARD_LOCALE='auto'`

Lorsque l'on travail directement sur le routeur `fli4l` avec un clavier de son pays c'est une aide non négligable. Avec `KEYBOARD_LOCALE='auto'` le clavier est réglé par rapport à la variable `LOCALE` et correspond au pays. Si aucun paramètre " est indiqué à l'installation du routeur `fli4l`, le clavier standard présent dans le Kernel est alors utilisé. On peut aussi indiquer directement le nom du pilote dans `keyboard_locale`, par ex. on indique `'fr-latin1'`, au démarrage du Buildprocess (ou processus de construction), il examine le répertoire `opt/etc` s'il trouve le fichier `fr-latin1.map` il charge le fichier-.map pour le code clavier demandé.

OPT_MAKEKBL Configuration par défaut : `OPT_MAKEKBL='no'`

Si vous voulez créer un fichier pour un code clavier spécifique, procéder comme indiqué si dessous :

- `OPT_MAKEKBL` mettez ici `'yes'`.
- On appel le programme `'makekbl.sh'`. Vous devez utilisez de préférence une connexion `ssh`, car le changements de disposition du clavier et qui peut être gênant.
- Exécuter les instructions.
- Le nouveau fichier `<locale>.map` est dans le répertoire `/tmp`.
- La création du fichier avec le routeur est maintenant achevée.
- Copier maintenant le nouveau code clavier généré dans votre `fli4l`= dans le répertoire `opt/etc/<locale>.map`. Vous pouvez utiliser le nouveau code clavier créé `KEYBOARD_LOCALE='<locale>'` dans le prochaine processus de construction.
- N'oubliez pas de remettre la variable `OPT_MAKEKBL` sur `'no'`.

3.7. Pilotes des cartes réseaux Ethernet

NET_DRV_N Configuration par défaut : `NET_DRV_N='1'`

Indiquer ici le nombre de pilote de cartes réseau.

Si le routeur est utilisé pour l'ISDN (ou numéris), il y a habituellement une seule carte réseau, la valeur par défaut est donc `'1'`.

3. Configuration de la base

Avec l'utilisation d'un modem DSL, on installe souvent deux cartes réseau.

Il faut distinguer deux cas :

1. Les deux cartes réseaux sont du même type (identique). On doit indiquer un seul pilote pour charger les deux cartes donc `NET_DRV_N='1'`.
2. Les deux cartes réseaux sont de type différent, vous indiquez '2' et spécifier un pilote pour chaque carte.

NET_DRV_x Configuration par défaut : `NET_DRV_1='ne2k-pci'`

On indique ici le pilote pour la ou les cartes réseaux. Dans la variable `NET_DRV_1` le pilote par défaut est NE2000 = carte réseau compatible elle sera chargée à l'installation, vous pouvez modifier le pilote selon votre configuration. L'ensemble des cartes réseaux sont indiquées dans les tableaux suivant ?? et ??.

Au sujet de la carte 3COM EtherLinkIII (3c509) vous avez un outil sous DOS, 3c509cfg.exe pour modifier les paramètres de la carte (téléchargeable ici <ftp://ftp.ihg.uni-duisburg.de/Hardware/3com/3C5x9n/3C5X9CFG.EXE>)

Vous pouvez éventuellement configurer l'IRQ et le port I/O pour les connecteurs (BNC/TP).

NET_DRV_x_OPTION Configuration par défaut : `NET_DRV_x_OPTION=""`

En général la variable peut rester vide.

Les pilotes de certaines cartes ISA ont besoin d'informations supplémentaires pour que le système trouve la carte, par exemple, l'adresse I/O. C'est le cas de la carte compatible NE2000 ISA et de EtherExpress16. Par exemple :

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

Indiquer (la valeur numérique correspondante).

Si aucun paramètre est nécessaire, la variable peut rester vide.

Si plusieurs paramètres sont nécessaires, ceux-ci sont à séparer par un espace (ou un blanc), par exemple :

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

Si deux cartes réseaux identiques sont utilisées, par exemple avec la NE2000-ISA, les valeurs des adresses I/O des cartes seront donc différentes et doivent être séparées par une virgule.

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

Les deux valeurs I/O doivent être séparées par une virgule sans espace !

Cela ne fonctionne pas avec tous les pilotes de carte réseau. Sur quelques une vous devez doubler le chargement du pilote, donc `NET_DRV_N='2'`. Dans ce cas, vous devez attribuer l'option "-o" avec un nom différent, par exemple

```
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'
```

Notre conseil : essayez la première méthode, puis essayez la seconde méthode avec l'option "-o".

Quelques exemples pour la configuration des cartes réseaux :

3. Configuration de la base

```
— 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=''
Voir aussi les faq sur les cartes (en Allemand) :
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'
Les valeurs IRQ doivent être placées ici :
    NET_DRV_1_OPTION='io=0x320,0x340 irq=3,5'
Vous devriez d'abord essayer de booter sans indiquer des interruptions. Si le pilote
réseau n'est pas identifié, alors ajouter les interruptions.
— 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'
```

Vous pouvez voir la liste de tous les pilotes qui peuvent être installés dans la documentation du paquetage kernel.

Si vous avez besoin d'un périphérique factice, vous pouvez indiquer 'dummy' dans la variable `NET_DRV_x` et

dans la variable `IP_NET_x_DEV` (Page 34)='dummy<Numéro>' pour le nom du périphérique.

3.8. Réseaux

IP_NET_N Configuration par défaut : `IP_NET_N='1'`

Dans cette variable on indique le nombre de réseaux qui sera associé au Protocole IP, en général '1' réseau est déjà indiqué. S'il n'y a pas de réseaux ou s'ils sont configurés sur un autre chemin, alors la variable `IP_NET_N` sera placé sur '0'. Un message d'avertissement sera indiqué lors de la construction de archive, on peut annuler cette avertissement avec la variable `IGNOREIPNETWARNING='yes'`.

IP_NET_x Configuration par défaut : `IP_NET_1='192.168.6.1/24'`

3. Configuration de la base

Présentation du dispositif pour l'adressage IP et du masque de sous-réseau avec CIDR ¹ dans le routeur fli4l. Si l'adresse IP est attribuée dynamiquement par le client DHCP, la valeur 'dhcp' sera alors indiquée dans cette variable.

Dans le tableau ci-dessous, vous pouvez voir les relations entre CIDR, le masque de sous-réseau et le nombre d'adresse IP

CIDR	Masque réseau	Nombre d'IPs
/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

Remarque : Puisque l'on réserve respectivement une adresse IP pour le Broadcast et une pour le réseau, le calcul du nombre maximal des hôtes dans le réseau est le suivant : $\text{Nombre_H\^otes} = \text{Nombre_IPs} - 2$. Le plus petit masque de sous-réseau est /30, correspondant à 4 adresses IP - 2 reste 2 adresses IP pour les hôtes.

IP_NET_x_DEV Configuration par défaut : `IP_NET_1_DEV='eth0'`

Requis : le nom du périphérique de la carte réseau.

Dès la version 2.1.8, le nom du périphérique utilisé est nécessaire ! Les noms des périphériques commencent dans la plupart des cas par 'eth' et suivi par d'un chiffre. La première carte réseau reconnue par le système reçoit le nom 'eth0', la deuxième 'eth1' etc...

Exemple :

```
IP_NET_1_DEV='eth0'
```

fli4l maîtrise aussi l'IP Aliasing, c'est l'attribution de plusieurs adresses IPs sur une carte réseau. on définit d'autres réseaux sur une même interface avec simplement des IPs supplémentaires. Lors de la vérification des informations de configuration, "mkfli4l" indique qu'un alias est défini – vous pouvez ignorer cet avertissement.

Exemple :

```
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 Configuration par défaut : `IP_NET_1_MAC=""`

Optionnel : adresse MAC de la carte réseau.

1. Classless inter-domaine Routing

3. Configuration de la base

Avec cette variable on peut installer l'adresse (MAC) de la carte réseau. Par exemple, si vous voulez utiliser un fournisseur d'accès DHCP qui attend uniquement une adresse MAC déterminée. Si la variable `IP_NET_x_MAC` est vide ou pas installée l'adresse MAC de la carte réseau pré-réglée sera installée automatiquement. La plupart des utilisateurs n'auront pas besoin de cette variable.

Exemple :

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

IP_NET_x_NAME Configuration par défaut : `IP_NET_x_NAME=""`

Optionnelle : On peut donner un nom à la carte réseau.

Lors de la résolution de nom inverse, un nom apparaîtra à la place de l'adresse IP selon le nom par défaut sous la forme `'f14l-ethx.<domain>'`. Avec la variable `IP_NET_x_NAME` vous pouvez indiquer le nom que vous voulez. Ce nom sera vu dans la résolution de nom inverse. Avec une adresse IP publique, on peut accéder au nom public de celle-ci, voir ci-dessous.

Exemple :

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

IP_NET_x_TYPE

IP_NET_x_COMMENT Configuration par défaut : `IP_NET_x_COMMENT=""`

Optionnelle : Cette variable sert à donner une indication à un périphérique avec un nom 'parlant'. Celui-ci peut être utilisé pour l'identification du réseau dans des paquets comme par exemple `opt rrdtool`.

3.9. Configuration du préfixe réseau

OPT_NET_PREFIX Avec cette variable vous activez la prise en charge personnalisée du préfixe réseau.

Techniquement un préfixe réseau n'est rien de plus qu'une adresse réseau, il désigne généralement un réseau qui doit être subdivisé davantage. Ceci est particulièrement utile si le routeur `f14l` ne gère pas seul le réseau, mais laisse la gestion des sous-réseaux à un autre routeur. Par cette définition et aussi par cette désignation le réseau sera disponible et distribué en totalité, il est possible d'utiliser l'adresse réseau en plusieurs endroits sans avoir à réécrire le préfixe à chaque fois.

Vous trouverez ci-dessous des exemples concrets sur la définition du préfixe réseau et pour les différents types de préfixes réseau.

Paramètre par défaut : `OPT_NET_PREFIX='yes'`

NET_PREFIX_x Ce tableau définit les différents préfixes du réseau. Les composants sont expliqués individuellement ci-dessous.

NET_PREFIX_x_NAME Nom du préfixe réseau.

Vous indiquez dans cette variable le nom du préfixe. Ce nom peut ensuite être utilisé dans les informations de l'adresse réseau et pour l'utilisation le préfixe. Le nom est utilisé comme un nom de circuit, c'est-à-dire qu'il doit être écrit entre les accolades.

NET_PREFIX_x_TYPE Type de préfixe de réseau.

Vous indiquez dans cette variable le type de préfixe. Les types supportés sont indiqués dans le tableau 3.3.

Type	Signification
static	Le préfixe réseau est spécifié directement en tant qu'adresse fixe.
generated-ula	Le préfixe de réseau est généré par fli4l en tant que ULA ² selon RFC 4193. ³ Si fli4l a accès à un stockage persistant, le préfixe est généré qu'une seule fois, il reste donc intact même après un redémarrage du routeur.

TABLE 3.3. – Types de préfixes réseau

3.9.1. Préfixe réseau de type "stable"

Pour les préfixes réseau "static", les paramètres suivants sont disponibles :

NET_PREFIX_x_STATIC_IPV4 **NET_PREFIX_x_STATIC_IPV6** Adresse(n) du préfixe réseau.

Vous indiquez dans ces variables les paramètres de l'adresse IPv4 et/ou IPv6 du préfixe réseau que vous allez utiliser.

Exemple :

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

3.9.2. Préfixe réseau de type "generated-ula"

Pour les préfixes réseau "generated-ula", les paramètres suivants sont disponibles :

NET_PREFIX_x_ULA_DEV Interface Ethernet.

Vous indiquez dans cette variable l'interface Ethernet, ainsi l'adresse MAC sera utilisée pour générer l'ULA.

Exemple :

2. "Unique Local Address"

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

```

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

```

3.10. Route supplémentaire (optionnel)

IP_ROUTE_N Configuration par défaut : `IP_ROUTE_N='0'`

Vous indiquez ici le nombre de routes supplémentaires pour le réseau. Une route supplémentaire est nécessaire par exemple, lorsqu'on a dans le LAN un routeur supplémentaire ou une passerelle sur lequel est connecté un autre réseau, ce réseau doit être accessible par le routeur `fli4l`.

Normalement il n'est pas nécessaire d'indiquer de route supplémentaire pour le réseau.

IP_ROUTE_x Les routes supplémentaires `IP_ROUTE_1`, `IP_ROUTE_2`, ... ont la structure suivante :

```
network/netmaskbits gateway
```

Pour se connecter il faut l'adresse du réseau `network` et son masque de sous-réseau `/netmaskbits`, avec la notation [CIDR](#) (Page 34) et l'adresse de la `gateway` (ou passerelle). Le routeur `fli4l` et la passerelle doivent être naturellement dans la même classe d'adresse IP, par exemple, pour que le réseau 192.168.7.0 avec son masque de sous-réseau 255.255.255.0 accède à la passerelle 192.168.6.99, on écrit alors :

```

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

```

Si le routeur `fli4l` n'est pas installé comme routeur Internet mais seulement comme un pur routeur Ethernet (un pont), on peut indiquer dans `IP_ROUTE_x` une route par défaut. On enregistre alors 0.0.0.0/0 à la place de `network/netmaskbits`, voir l'exemple suivant :

```

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. Le filtrage de paquets

Le Kernel de Linux utilisé par `fli4l` met à disposition un filtrage de paquets. A l'aide de ce filtrage de paquets, on contrôle les flux qui communiquent avec le routeur et au de là de celui-ci. Par ailleurs, vous pouvez réaliser des dispositifs comme la redirection de port (redirige

3. Configuration de la base

les paquets reçus par le routeur et les transmettre à un ordinateur du réseau interne) et le masquage (en anglais masquerading. Les paquets provenant d'un ordinateur du réseau interne derrière le routeur sont modifiés, de telle sorte que ces paquets semblent provenir du routeur lui-même).

La structure du filtrage de paquets est indiquée sur le schéma 3.1. Les paquets qui entrent par l'interface réseau, parcourent la chaîne **PREROUTING** (en anglais "chain"). le routeur qui reçoit les paquets sont manipulés, ils sont ensuite renvoyés vers un autre ordinateur en utilisant l'adresse et le port de destination. Si les paquets sont adressés au routeur, ils parcourent la chaîne **INPUT**, sinon ils parcourent la chaîne **FORWARD**. Les deux chaînes examinent si les paquets sont autorisés à passer. S'il sont acceptés, les paquets sont envoyés sur le processus cible locale ou vers la chaîne **POSTROUTING** (ici le masquage de paquets a lieu) ensuite les paquets passent par l'interface réseau et peuvent atteindre leur destination. Les paquets générés localement sont filtrés dans la chaîne **OUTPUT** et enfin (en cas de succès) les paquets sont envoyés dans la chaîne **POSTROUTING** et sont également transmis en passant par l'interface réseau.



FIGURE 3.1. – Structure du Filtrage de paquets

La configuration des chaînes de filtrage de paquets, peut être paramétrée séparément. En plus il y a une liste pertinente pour chaque chaîne importante, c.-à-d. pour la chaîne **INPUT** vous avez (**PF_INPUT_%**), pour la chaîne **FORWARD** vous avez (**PF_FORWARD_%**), pour la chaîne **PREROUTING** vous avez (**PF_PREROUTING_%**) dans laquelle vous effectuez la redirection de ports et pour la chaîne **POSTROUTING** vous avez (**PF_POSTROUTING_%**) dans laquelle vous exécutez le masquage de paquets.

Le paramétrage de la liste se compose principalement d'une action (voir ci-dessous), qui peut être limitée par des conditions supplémentaires. Ces conditions portent sur les propriétés du paquet. Un paquet contient des informations sur son origine (la source quel ordinateur a envoyé le paquet), sa cible (à quel ordinateur et quelle application le paquet doit aller), etc. les conditions du paquet peuvent être basées sur les propriétés suivantes :

- La source (adresse source, port source, ou les deux)
- La destination (adresse de destination, le port de destination, ou les deux)
- Le protocole
- L'interface sur laquelle le paquet entre ou sort
- L'adresse MAC de l'ordinateur qui a envoyé le paquet
- L'état du paquet ou du lien dont le paquet fait partie

Si un paquet entre, les enregistrements ou les règles générées sont récupérées de haut en bas, la première action est exécutée et toutes les conditions. Si aucune des règles ne s'applique,

3. Configuration de la base

l'action par défaut est exécutée, vous pouvez spécifier des règles pour (presque) toutes les tables.

Un enregistrement a la forme suivant, vous devez faire attention que toutes les restrictions sont optionnelles :

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

Sur les points qui concerne le réseau vous devez spécifier une adresse IP ou un hôte. vous pouvez aussi utiliser les variables `IP_NET_%`, `IP_NET_%_IPADDR` ou un `@non_d'hôte` via les variables `HOST_%`. Si la variable `OPT_DNS` est activées, vous pouvez référencer un nom externe au réseau local via `@fqdn`, mais vous ne devez *pas* récupérer le nom dans la variable `HOST_%`. Cela est particulièrement utile, quand il s'agit d'hôtes externes, et qu'ils possèdent en plus des adresses IP dynamique (ou changeante).

3.11.1. Action pour le filtrage de paquets

Les actions peuvent être les suivants :

Action	Chaîne(n)	Importance
ACCEPT	Tous	Accepte le paquet
DROP	INPUT FORWARD OUTPUT	Le paquet est rejeté (l'expéditeur ne recevra pas de réponse et aucun message ne lui reviendra).
REJECT	INPUT FORWARD OUTPUT	Le paquet est rejeté mais (l'expéditeur recevra un message d'erreur).
LOG	Tous	Le paquet est enregistré et continu sur la règle suivante. Vous pouvez utiliser un préfixe pour différencier les entrées dans le fichier journal, en spécifiant <code>LOG :log-prefix</code> . Le préfix peut avoir une longueur de 28 caractères et peut contenir des lettres, des chiffres, le trait d'union (-) et le tiret bas (_).
MASQUERADE	POSTROUTING	Le paquet est masqué : l'adresse source du paquet sera remplacé par la propre adresse de l'interface, adresse que vous lui aviez attribuée, assurez-vous que la réponse est correctement transmise à l'ordinateur source.
SNAT	POSTROUTING	L'adresse et le port source du paquet seront remplacés, la variable sera paramétrée comme ceci <code>SNAT</code> spécifier l'adresse (considéré que tous les paquets appartiennent à cette connexion).
DNAT	PREROUTING	L'adresse et le port de destination seront remplacés, la variable sera paramétrée comme ceci <code>DNAT</code> spécifier l'adresse (considéré que tous les paquets appartiennent à cette connexion).

3. Configuration de la base

Action	Chaîne(n)	Importance
REDIRECT	PREROUTING OUTPUT	Le port de destination du paquet sera remplacer, la variable sera paramétrée comme ceci REDIRECT spécifier le port, le paquet généré sera mappé au niveau local (considéré que tous les paquets appartiennent à cette connexion).
NETMAP	PREROUTING POSTROUTING	Crée une image de l'adresse cible ou de la source du paquet, la variable sera paramétrée comme ceci NETMAP spécifier l'adresse du domaine, les ports restent inchangés (considéré que tous les paquets appartiennent à cette connexion, dans la chaîne PREROUTING l'adresse de destination sera modifiée et dans la chaîne POSTROUTING c'est l'adresse de source qui sera modifiée).

TABLE 3.4. – Action des règles du filtrage de paquets

On peut modifier le comportement de certaines de ces actions avec les options suivant BIDIRECTIONAL, LOG ou NOLOG. L'option BIDIRECTIONAL génère à nouveau la même règle, mais avec une adresse source et destination inversée (un changement de port source et destination et/ou un changement de l'interface réseau sortant si elle est spécifiée). Les options LOG/NOLOG active ou n'active pas le fichier journal pour cette règle.

3.11.2. Restriction dans les règles

les restrictions peuvent être réalisés, elle sont indiquées dans ce chapitre ci-dessous. Si vous ne voulez pas faire de restriction vous pouvez toujours spécifier **any**, mais vous devez toujours indiquer quelque chose. Les restrictions peuvent être spécifiées dans n'importe quel ordre, si le préfixe est préposé. Cela s'applique à toutes les restrictions, sauf pour spécifier une adresse source ou destination. Ceux-ci doivent toujours être directement devant l'action, les autres restrictions doivent être faites avant. Les restrictions peuvent également être annulés, en préposant tout simplement le symbole !

Restriction de la source et de la destination

Chaque paquet contient une source et une cible, respectivement sous la forme multiplet d'adresse IP et de port.⁴ Cette source ou cette cible peut être utilisé pour une restriction. La spécification de la source ou de la destination peut être faite comme ceci :

Expression	Importance
ip	Une seule adresse IP
network	Spécification du réseau sous la forme <ip>/<netmask>
port [-port]	Port ou une plage de ports

4. Le port est disponible seulement pour les paquets avec le protocole TCP et UDP.

3. Configuration de la base

Expression	Importance
IP_NET_x_IPADDR	Adresse IP de l'interface x du routeur
IP_NET_x	Sous-réseau x du routeur
IP_ROUTE_x	Spécifier la route x du sous-réseau (Les routes par défaut ne peuvent pas être utilisés, elles seraient toutes any , et sont exclus par précaution)
@name	Nom ou alias attribué, dans la variable HOST_%_* l'adresse IP est utilisé au nom associée
<ip ou réseau>:port[-port]	Hôte ou l'adresse du réseau de l'une des variantes ci-dessus, combiné à un port ou une plage de ports

TABLE 3.5. – Restrictions de la source et de destination dans les règles de filtrage de paquets

Cela pourrait par exemple, ressembler à ceci : '192.168.6.2 any DROP'

Si on regarde ces paramètres, le premier est la source, le second est considéré comme la cible. Dans cet exemple, nous rejetons les paquets qui ont été envoyés par l'ordinateur avec l'adresse IP 192.168.6.2 et peu importe sur quelle destination ils sont adressés.

Si un seul paramètre est indiqué, on peut décider en fonction de la valeur, si c'est la source ou si c'est la destination qui est concernée, la décision est relativement simple :

- Si le port est paramétré, ce sera la cible qui est concernée.
- Sinon, ce sera la source qui est concernée.

Si nous voulons écrire plus brièvement l'exemple ci-dessus : '192.168.6.2 DROP'. Aucun port n'est indiqué, donc l'IP de l'ordinateur est la source (c'est lui qui envoie les paquets).

Si nous voulons communiquer avec le démon **ssh**, nous pouvons indiquer 'any any:22 ACCEPT' (les paquets seront acceptés depuis n'importe quel ordinateur sur le Port 22 du **ssh** et n'importe quel ordinateur les acceptera), vous pouvez indiquer aussi '22 ACCEPT' seul un port est indiqué, nous pouvons dire que c'est la cible, les paquets seront dirigés sur le port 22.

Pour simplifier la quantité de règle à écrire, on peut utiliser l'action **BIDIRECTIONAL** elle indique que les communications se feront dans les deux sens. Les règles sont paramétrées simplement avec l'IP source, l'IP destination et le port ou avec les interfaces réseau, les échanges entre ces deux réseaux restent les mêmes.

Exemple :

127.0.0.1 ACCEPT	La communication locale (Source 127.0.0.1) est acceptée
any 192.168.12.1 DROP	Les paquets vers l'adresse IP 192.168.12.1 sont rejetés
any 192.168.12.1 DROP LOG	Les paquets vers l'adresse IP 192.168.12.1 sont rejetés et sont également enregistrés
any 192.168.12.1 DROP NOLOG	Les paquets vers l'adresse IP 192.168.12.1 sont rejetés, ne sont pas enregistrés
22 ACCEPT	Les paquets vers le port 22 (ssh) sont acceptés
IP_NET_1_NET ACCEPT	Les paquets du sous-réseau de la première interface sont acceptés
IP_NET_1_NET IP_NET_2_NET	La communication entre la première et la seconde
ACCEPT BIDIRECTIONAL	L'interface du sous-réseau est acceptée

Restriction des interfaces

Une règle peut restreindre une interface sur laquelle les paquets arrivent et sortent. Le format de restriction est le suivant : `if:in:out`

Dans la chaîne `INPUT` on ne peut pas restreindre l'interface pour les paquets sortant (les paquets ne sortiront plus). Dans la `POSTROUTING` on ne peut pas restreindre l'interface pour les paquets entrant, car l'information n'est plus disponible à ce moment là. Vous pouvez restreindre une interface uniquement dans la chaîne `FORWARD` pour les deux conditions (entrant et sortant).

Les valeurs suivantes sont possibles pour *in* ou *out* :

- `lo` (Interface de bouclage, communication locale sur le routeur)
- `IP_NET_x_DEV`
- `pppoe` (Interface PPPoE, seulement si le `dsl` ou le `pppoe_server` est activé)
- `any`

Restriction du protocole

Une règle peut restreindre le protocole donc le paquet appartient. Le format est le suivant : `prot:protocol` ou bien `prot:icmp:icmp-type`. *protocol* peut prendre les valeurs suivantes :

- `tcp`
- `udp`
- `gre` (Generic Routing Encapsulation)
- `icmp` (Vous pouvez spécifier un nom pour le type de filtrage ICMP (`echo-reply` ou `echo-request`, en gros `prot:icmp:echo-request`)
- Valeur numérique du protocole ID (exemple 41 pour IPv6)
- `any`

Si vous voulez utiliser un numéro de port avec des protocoles différents dans une règle, une telle restriction ne sera pas disponible, vous devez créer la règle en *deux fois*, une fois pour le `tcp` et une fois pour le `udp`.

Restriction des adresses MAC

Vous pouvez utiliser `mac:mac-address` pour faire une restriction de l'adresse MAC.

Restriction sur l'état d'un paquet

pour le filtrage de paquets `fi4l` utilise les informations de l'état des connexions. Ces informations peuvent ensuite être utilisées pour filtrer les paquets, pour une description plus détaillée sur l'état des connexions : ⁵

Etat	Importance
INVALID	Le paquet n'appartient à aucune connexion connue.
ESTABLISHED	Le paquet appartient à une connexion, le paquet a déjà circulé dans l'autre sens (réponse).
NEW	Le paquet fait partie d'une nouvelle connexion ou appartient à une connexion, mais le paquet n'a pas encore circulé dans l'autre sens.

5. voir http://www.sns.ias.edu/~jns/files/iptables_talk/x38.htm

Etat	Importance
RELATED	Le paquet fait partie d'une nouvelle connexion, mais il est déjà en relation avec une connexion existante (par ex. établissement d'une connexion avec le ftp pour le transfère de données).

TABLE 3.6. – Restriction des règles sur de filtrage de paquets

L'état du paquet est défini comme ceci : **state:état(s)**. Si vous souhaitez spécifier plusieurs conditions, vous devez les sépare par une virgule. Par exemple si vous voulez laisser passer seulement les paquets qui appartiennent directement ou indirectement à une connexion vous paramétrez **state:ESTABLISHED,RELATED**, il est utile d'écrire (ces états dans la chaîne **INPUT** ou **FORWARD**).

Restriction sur les fréquences des actions

Dans certaine circonstance, on aimerait limiter la fréquence des actions, par ex. faire seulement une demande d'écho ICMP par seconde. Cela peut être spécifié avec la commande limitation **limit**, le format sera le suivant : **limit:fréquence :Burst**. La fréquence est en *n/unité de temps* qui sera donnée en (second, minute, hour, day), de plus vous pouvez indiquer une suite d'événements successifs (Burst). Par exemple en spécifiant **limit:3/minute:5** un maximum de trois événements par minute sera permis et cinq événements successifs seront acceptés.

3.11.3. Utilisation d'un modèle pour le filtrage de paquets

Il est possible pour l'utilisateur de simplifier la configuration des données de filtrage de paquets, en l'utilisant un modèle (Template) c'est un condensé de règles prés enregistré qui est fréquemment utilisées. Il est ainsi possible de combiner un certain nombre de règles de filtrage de paquets, dans cette collection de règles un nom symbolique y est associé. Au lieu d'écrire directement dans la variable le protocole et le numéros de port il suffit d'écrire le nom symbolique, si vous voulez utiliser le protocole **ssh** dans une règle il suffit d'écrire **tmpl:ssh**. Comment faut-il procéder avec le modèle **ssh**, vous avez un exemple d'utilisation ci-dessous.

Si vous voulez atteindre votre routeur fli4l par Internet avec le **ssh**, vous devez écrire dans la variable **PF_INPUT_%** le nom du service correspondant (ici **ssh**), précédée par **tmpl:** et l'action qui consiste à appliquer ce service. Par exemple :

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

Voici comment utiliser *tmpl* : pour appliquer une règle dans un modèle. Vous donnez le nom du service après les ' : ', dans notre exemple **ssh**. Enfin, vous pouvez spécifier quelle action doit être connecté au service. Puisque vous voulez communiquer avec fli4l à partir d'Internet, nous autorisons la connexion avec **ACCEPT**. La limitation des adresses IP ou des réseaux ne sont pas spécifiées, Avec le service **ssh** tous les réseaux et toutes les interfaces sont accessibles. Vous pouvez utiliser en cas de besoin la configuration habituelle du filtrage de paquets pour limiter l'accès au service **ssh**.

Pour quels services des règles sont ils préparées (c.-à-d. le modèle existent), vous pouvez trouver dans le fichier **opt/etc/fwrules.tmpl/templates** le modèle de service prés configuré. Voir ci-dessous la liste (dans le tableau 3.7).

3. Configuration de la base

Modèle	Protocole	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
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

3. Configuration de la base

Modèle	Protocole	Port(s)
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
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

3. Configuration de la base

Modèle	Protocole	Port(s)
xmppserver	tcp	5269

TABLE 3.7. – Modèles inclus dans fli4l de base

La syntaxe pour cette forme de règles de filtrage de paquets est toujours

```
tmpl:<Nom du service> <Restriction> <Action souhaitée>
```

Les **<restrictions>** permis sont décrites dans la section 3.11.2. Les valeurs possibles pour les **<actions souhaitées>** sont énumérées et décrites dans la section 3.11.1

Voici quelques exemples pour illustrer la démarche. Nous allons d'abord utiliser 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'
```

La règle PF_PREROUTING_1 fournit à Xbox tout le nécessaire pour Xbox Live. Plus précisément, avec **tmpl:xbl** vous avez tous les ports et les protocoles nécessaires afin qu'il y ait une transmission entre Xbox Live et les Hôtes **xbox**. Au lieu de saisir l'adresse IP vous pour enregistrer un nom depuis les paramètres **HOST_%_NAME**. Si vous indiquez **dynamic** fli4l sait que les ports seront transmis par l'interface connectée à Internet.

La deuxième règle dirige les paquets correspondants au protocole **https** sur un serveur Web par l'intermédiaire de la DMZ.

Maintenant, nous allons utiliser 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'
```

La première règle permet à tous les réseaux défini dans **IP_NET_1** d'accéder au routeur. La deuxième règle permet à tous les paquets **oident** d'accéder. sur le port **ident** est ouvert. la troisième règle et dernière permet à Xbox d'accéder au serveur DNS sur fli4l. Vous avez également comment utiliser un alias d'hôte dans une règle.

Avec PF_FORWARD et PF_POSTROUTING il n'y a rien de plus que le **tmpl** spécifique.

Il est possible de créer vos propre fichier de modèle ou d'utiliser le fichier existant pour ajouter vos règles. Pour créer votre propre modèle vous devez simplement créer un fichier avec un nom de modèle que vous voulez et enregistrer les règles correspondants selon vos besoins. Si vous avez décidé de créer un fichier modèle, vous devez copier le fichier dans le sous-répertoire **etc/fwrules.tmpl** qui sera dans le répertoire **config**, comme indiqué dans la figure 3.2. Si le sous-répertoire **etc/fwrules.tmpl** dans le répertoire **config** n'existe pas, vous devez créer ce sous-répertoire. les développeurs de paquets ou les utilisateurs qui souhaitent créer leurs règles pour plusieurs configurations, peuvent stocker directement le fichier modèle dans le répertoire **opt/etc/fwrules.tmpl**. Dans ce répertoire, sont enregistré que les nouveaux fichiers modèles. Dans le répertoire **config** les fichiers modèles pour les règles sont prioritères au répertoire utilisateur. Enfin, vous pouvez copier le fichier modèle original fourni par fli4l

3. Configuration de la base



FIGURE 3.2. – Structure du répertoire fli4l

qui est déjà «configuré», puis coller le contenu dans votre propre fichier, vous pouvez ensuite enregistrer dans le répertoire `config`.

Par exemple, si vous voulez ajouter le modèle `vpn_freunde` vous devez d'abord créer le fichier `vpn_freunde`. Ensuite vous devez enregistrer les services suivants, `ssh`, `smtp`, `dns` et `samba`. Le fichier `vpn_freunde` doit ressembler à ceci :

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

Maintenant à chaque fois que vous utilisez le modèle `vpn_freunde` des règles que vous avez enregistré seront générées pour tous les protocoles et les ports spécifiés. Exemple avec une action `PF_FORWARD_x='tmpl:vpn_freunde ACCEPT'`, les règles du FORWARD seront les suivantes :

```
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 du filtrage de paquets

Le filtrage de paquets est configurable essentiellement par cinq chaînes voir le tableau :

3. Configuration de la base

- PF_INPUT_% se configure avec la chaîne INPUT,
- PF_FORWARD_% se configure avec la chaîne FORWARD,
- PF_OUTPUT_% se configure avec la chaîne OUTPUT,
- PF_PREROUTING_% se configure avec la chaîne PREROUTING et
- PF_POSTROUTING_% se configure avec la chaîne POSTROUTING.

Dans la variable on règle le niveau de journalisation, valable pour l'ensemble des variables, voici les valeurs que l'on peut paramétrer : `debug`, `info`, `notice`, `warning`, `err`, `crit`, `alert`, `emerg`.

Chaîne INPUT

Configuration de la chaîne INPUT, c'est ici que les paquets entrent dans le routeur, les hôtes peuvent interroger le routeur. S'il n'y a pas de règle définie, pour la chaîne INPUT l'action par défaut sera déterminée, que faut-il faire du paquet lorsqu'il est refusé, la variable de protocole détermine si le paquet doit être écrit dans le journal du système.

Il y a deux restrictions, au sujet des paramètres à utiliser :

- Seul les valeurs `ACCEPT`, `DROP` et `REJECT` sont spécifiés comme action.
- Lors d'une restriction d'interface vous ne pouvez que restreindre l'interface d'entrée.

PF_INPUT_POLICY Cette variable décrit l'action par défaut, qui est utilisée lorsque aucune autre règle ne s'applique. Les options sont :

- `ACCEPT` (pas recommandé)
- `REJECT`
- `DROP` (pas recommandé)

PF_INPUT_ACCEPT_DEF Si cette variable est sur 'yes', les règles par défaut sont générées, cela est nécessaire pour un bon fonctionnement du routeur. Vous devez indiquer 'yes' pour une configuration par défaut.

Si vous avez besoin de définir le comportement du routeur, vous devez indiquer 'no'. Vous devez alors paramétrer toutes les règles vous-mêmes. Un comportement par défaut pour une configuration équivalente devrait ressembler à ceci, (la description des chaînes définies par l'utilisateur est [ici](#) (Page 51)) :

```
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'

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'
```

La première règle limite le contrôle des erreurs avec la chaîne "usr-in-icmp". La deuxième règle accepte seulement les paquets qui appartiennent à une connexion existante (c. à d. que les paquets sont dans un état `ESTABLISHED` ou `RELATED`), la troisième règle permet

3. Configuration de la base

la communication locale avec (`if:lo:any ACCEPT`). La quatrième règle filtre les paquets qui prétendent avoir une communication locale, mais qui n'ont pas été acceptées par la règle précédente.

Si vous travaillez avec OpenVPN, vous devez ajouter des règles, l'utilisation de ces paquets comportent les chaînes suivante :

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

PF_INPUT_LOG Ici on définit, si les paquets refusés doivent être enregistrés par le Kernel. Pour recevoir les messages vous devez activer la variable `OPT_KLOGD` via le démon syslog, les messages reçus sont fonction de votre configuration.

PF_INPUT_LOG_LIMIT Vous définissez dans cette variable la fréquence les entrées générée dans le journal. La fréquence pour la limites de restriction est décrite de façon analogue comme ceci *n/Unité de temps* avec un Burst (ou en rafale). Par exemple `3/minute:5`. Si la valeur par défaut est vide, la valeur `1/second:5` sera utilisé. Si vous indiquez `none` aucune limite ne sera effectuée.

PF_INPUT_REJ_LIMIT PF_INPUT_UDP_REJ_LIMIT Ici on définit la fréquence de refus du paquet entrant, le paquet générée sera REJECT. La fréquence de la limite de restriction est décrite de façon analogue comme ceci *n/Unité de temps* avec un Burst (ou en rafale) par exemple `3/minute:5`. Lorsque la limite est dépassée, le paquet sera simplement ignoré (DROP). Si la valeur par défaut est vide, la valeur `1/second:5` sera utilisé. Si vous indiquez `none` aucune limite ne sera effectuée.

PF_INPUT_ICMP_ECHO_REQ_LIMIT Ici on définit la fréquence de comment répondre à une demande echo ICMP. La fréquence de la limite de restriction est décrite de façon analogue comme ceci *n/Unité de temps* avec un Burst (ou en rafale) par exemple `3/minute:5`. Lorsque la limite est dépassée, le paquet sera simplement ignoré (DROP). Si la valeur par défaut est vide, la valeur `1/second:5` sera utilisé. Si vous indiquez `none` aucune limite ne sera effectuée.

PF_INPUT_ICMP_ECHO_REQ_SIZE Vous définissez dans cette variable la taille d'une demande d'écho ICMP reçu en (en octets). Ce chiffre vient s'ajouter à la charge de "l'entête" du paquet, cela est à pendre en considération. La valeur par défaut est de 150 octets.

PF_INPUT_N PF_INPUT_x PF_INPUT_x_COMMENT Vous indiquez dans cette liste de règles, les paquets qui sont acceptés ou rejetés par le routeur.

Chaîne FORWARD

Configuration de la chaîne FORWARD, c'est ici que le routeur redirige les paquets. S'il n'y a pas de règle définie, pour la chaîne FORWARD l'action par défaut sera déterminée, que faut il faire du paquet lorsqu'il est refusé, la variable de protocole détermine si le paquet doit être écrit dans le journal du système.

Les paramètres utilisés pour les actions de restriction sont, ACCEPT, DROP et REJECT

PF_FORWARD_POLICY Cette variable décrit l'action par défaut qui sera utilisée, lorsque aucune autre règle ne s'applique. Les options sont :

3. Configuration de la base

- ACCEPT
- REJECT
- DROP

PF_FORWARD_ACCEPT_DEF Ici on détermine si le routeur accepte les paquets appartenant à des connexions existantes. Si cette variable est paramétrée sur 'yes', `firewall` génère automatiquement la règle qui accepte les paquets dans un état approprié :

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

poursuite de la règle, rejette des paquets avec l'état inconnu :

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

et enfin la règle ignore les paquets avec une adresse IP usurpée :

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

En outre, d'autres sous-systèmes génèrent les règles par défaut – Voici une configuration sans les règles par défaut, avec la redirection de port et l'OpenVPN la configuration devrait contenir au moins les règles suivantes :

```
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 Ici on définit, si les paquets refusés doivent être enregistrés par le Kernel. Pour recevoir les messages vous devez activer la variable `OPT_KLOGD` via le démon `syslog`, les messages reçus sont fonction de votre configuration.

PF_FORWARD_LOG_LIMIT Vous définissez dans cette variable la fréquence des entrées générées dans le journal. La fréquence pour la limite de restriction est décrite de façon analogue comme ceci *n/Unité de temps* avec un Burst (ou en rafale). Par exemple `3/minute:5`. Si la valeur par défaut est vide, la valeur `1/second:5` sera utilisé. Si vous indiquez `none` aucune limite ne sera effectuée.

PF_FORWARD_REJ_LIMIT PF_FORWARD_UDP_REJ_LIMIT Ici on définit la fréquence de refus du paquet entrant, le paquet générée sera REJECT. La fréquence de la limite de restriction est décrite de façon analogue comme ceci *n/Unité de temps* avec un Burst (ou en rafale) par exemple `3/minute:5`. Lorsque la limite est dépassée, le paquet sera simplement ignoré (DROP). Si la valeur par défaut est vide, la valeur `1/second:5` sera utilisé. Si vous indiquez `none` aucune limite ne sera effectuée.

PF_FORWARD_N PF_FORWARD_x PF_FORWARD_x_COMMENT Vous indiquez dans cette liste de règles, les paquets qui sont redirigés ou rejetés par le routeur.

Chaîne OUTPUT

Configuration de la chaîne OUTPUT, c'est ici que le routeur gère ces paquets sortant. S'il n'y a pas de règle définie, pour la chaîne OUTPUT l'action par défaut sera déterminée, que faut-il faire du paquet lorsqu'il est refusé, la variable de protocole détermine si le paquet doit être écrit dans le journal du système.

Les paramètres utilisés pour les actions de restriction sont :

3. Configuration de la base

- Vous devez utiliser seulement les actions ACCEPT, DROP et REJECT.
- Lors d'une restriction d'interface, la chaîne ne peut que restreindre l'interface de sortie.

PF_OUTPUT_POLICY Cette variable décrit l'action par défaut qui sera utilisée, lorsque aucune autre règle ne s'applique. Les options sont :

- ACCEPT
- REJECT
- DROP

PF_OUTPUT_ACCEPT_DEF Si cette variable est sur 'yes', les règles par défaut sont générées, cela est nécessaire pour un bon fonctionnement du routeur. vous devez indiquer 'yes' pour une configuration par défaut.

Si vous avez besoin de définir le comportement du routeur, vous devez indiquer 'no'. Vous devez alors paramétrer toutes les règles vous-mêmes. Un comportement par défaut pour une configuration équivalente devrait ressembler à ceci :

```
PF_OUTPUT_ACCEPT_DEF='no'

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

La première règle (et la seule) accepte seulement les paquets qui appartiennent à une connexion existante (c'est à dire les paquets qui ont soit l'état ESTABLISHED ou RELATED)

PF_OUTPUT_LOG Ici on définit, si les paquets refusés doivent être enregistrés par le Kernel. Pour recevoir les messages vous devez activer la variable OPT_KLOGD via le démon syslog, les messages reçus sont fonction de votre configuration.

PF_OUTPUT_LOG_LIMIT Vous définissez dans cette variable la fréquence les entrées générées dans le journal. La fréquence pour la limite de restriction est décrite de façon analogue comme ceci *n/Unité de temps* avec un Burst (ou en rafale). Par exemple 3/minute:5. Si la valeur par défaut est vide, la valeur 1/second:5 sera utilisé. Si vous indiquez none aucune limite ne sera effectuée.

PF_OUTPUT_REJ_LIMIT PF_OUTPUT_UDP_REJ_LIMIT Ici on définit la fréquence de refus du paquet entrant, le paquet générée sera REJECT. La fréquence de la limite de restriction est décrite de façon analogue comme ceci *n/Unité de temps* avec un Burst (ou en rafale) par exemple 3/minute:5. Lorsque la limite est dépassée, le paquet sera simplement ignoré (DROP). Si la valeur par défaut est vide, la valeur 1/second:5 sera utilisé. Si vous indiquez none aucune limite ne sera effectuée.

PF_OUTPUT_N PF_OUTPUT_x PF_OUTPUT_x_COMMENT Vous indiquez dans cette liste de règles, les paquets qui sont envoyés ou rejetés par le routeur.

Chaîne personnalisée

Parfois pour différentes raisons, on a besoin d'élaborer nos propres chaînes pour régler plus précisément le filtrage de paquets. Ces chaînes peuvent être définies et paramétrées en utilisant la variable PF_USR_CHAIN_%. Les noms de chaîne doivent commencer obligatoirement par *usr-* suivi de ce que vous voulez, ils peuvent être utilisés n'importe où dans les chaînes INPUT ou FORWARD pour spécifier une action. Par exemple, voici utilisation de la chaîne de filtrage ICMP :

```
PF_USR_CHAIN_N='1'
```

3. Configuration de la base

```
#
# 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 Dans cette variable vous indiquez le nombre de chaîne définie par l'utilisateur.

PF_USR_CHAIN_x_NAME Dans cette variable vous indiquez le nom de la chaîne. Le nom doit commencer par *usr-*.

PF_USR_CHAIN_x_RULE_N

PF_USR_CHAIN_x_RULE_x

PF_USR_CHAIN_x_RULE_x_COMMENT Vous indiquez dans cette liste de règles, les règles définies par l'utilisateur. Les règles doivent être insérées dans la chaîne **FORWARD**. Si aucune règle ne s'applique le processus sort de la chaîne **USR**, remonte à la chaîne d'origine et continue sur la règle suivante.

Chaîne NAT (Network Address Translation)

Les paquets peuvent être manipulés, avant et après les décisions de routage pour le moment. Par exemple, vous pouvez obtenir une nouvelle adresse de destination pour la transmission à un autre ordinateur (port forwarding) ou recevoir une adresse source différente pour masquer le réseau situé derrière le routeur. Le masquage est utilisé par exemple, pour faire un réseau privé avec une adresse IP publique ou de cacher une configuration DMZ du réseau local à partir d'un ordinateur de la DMZ.

La configuration se fait sur deux chaînes, la **PREROUTING** et la **POSTROUTING**. Au sujet de la chaînes **POSTROUTING** on configure, les paquets qui seront masqués par le routeur. Si aucune des règles de chaînes **POSTROUTING** ne s'applique, les paquets seront acheminés démasqué.

Pour le masquage, il existe deux versions : une pour l'interface réseau qui est assignée lors de la connexion une seule adresse IP (**MASQUERADE**) et une pour l'interface réseau qui utilise une adresse IP statique(**SNAT**). Si vous utilisez **SNAT** l'adresse IP doit être enregistrée dans le paquet source. Les données peuvent être comme ceci.

- Adresse IP (exemple : **SNAT :1.2.3.4**),
- Plage d'adresses IP (exemple : **SNAT :1.2.3.4-1.2.3.10**)
- ou comme une référence symbolique (exemple : **SNAT :IP_NET_1_IPADDR**)

Vous pouvez indiquer un port ou une plage de ports dans **SNAT** et aussi dans **MASQUERADE** pour mapper les ports (ou établir une correspondance entre les ports). Normalement ce n'est pas nécessaire car seul le Kernel peut sélectionner les ports pour établir une correspondance. Cependant il y a des applications qui nécessitent que le port source reste inchangé (et imposent

3. Configuration de la base

un NAT 1 :1 ou elles interdisent le PAT (Port Address Translation) ou NAPT (Network Address and Port Translation)). La plage de ports est simplement ajouté après l'adresse IP, par exemple : `SNAT :IP_NET_1_IPADDR :4000-8000`.

La chaîne `POSTROUTING` peut utiliser les actions suivantes `ACCEPT`, `SNAT`, `NETMAP` et `MASQUERADE`.

PF_POSTROUTING_N PF_POSTROUTING_x PF_POSTROUTING_x_COMMENT

Vous indiquez dans cette liste de règles, des paquets qui seront masqués par le routeur (ou transmis non masqué). Si vous ne voulez pas masquer les paquets qui arrive sur le routeur, vous pouvez placer la règle `Accept` à la place de la règle `Masquerade`.

Dans la chaîne `PREROUTING` on configure les paquets qui doivent être transmis à un autre ordinateur. Si aucune règles de la chaîne `PREROUTING` ne s'applique, les paquets seront ensuite traités sans être changés. L'action `DNAT` attend une adresse IP qui doit être enregistré dans le paquet en tant que cible. Les données peuvent être comme ceci.

- Adresse IP (exemple : `DNAT :1.2.3.4`),
- Plage d'adresses IP (exemple : `DNAT :1.2.3.4-1.2.3.10`)
- Ou comme un nom d'hôte (exemple : `DNAT :@client1`)

Vous pouvez indiquer un port ou une plage de ports, le port de destination sera mappée. Cela est nécessaire que si le port doit être changé. Le port est simplement ajouté après l'adresse IP, par exemple : `DNAT :@server :21`.

L'action `REDIRECT` se comporte comme l'action `DNAT`, sauf que l'adresse IP de destination est toujours une adresse IP (primaire) - c'est l'adresse de l'interface qui est configurée et sur laquelle le paquet arrive, le paquet sera ensuite livré localement. Cela est nécessaire par exemple pour un proxy transparent, voir `OPT_TRANSPROXY` (Page ??).

Si vous voulez faire de la redirection de port sur des interfaces qui utilise une adresse IP dynamique, pendant le démarrage on ne connaît pas l'adresse IP du PC vers lequel les paquets seront dirigés. on peut utiliser la chaîne `PREROUTING` avec le paramètre `dynamic` comme espace réservé pour assignée l'adresse IP plus tard. Par exemple :

```
'dynamic:80 DNAT:1.2.3.4'          # rediriger les paquets http vers
                                   # l'adresse IP 1.2.3.4
'prot:gre any dynamic DNAT:1.2.3.4' # rediriger les paquets gre (fait partie
                                   # du protocole PPTP) vers l'adresse
                                   # 1.2.3.4
```

La chaînes `PREROUTING` peut utiliser les actions suivantes `ACCEPT`, `DNAT`, `NETMAP` et `REDIRECT`.

Pour d'autres exemples sur la façon d faire de la redirection de port, voir la paragraphe suivant.

PF_PREROUTING_N PF_PREROUTING_x PF_PREROUTING_x_COMMENT

Vous indiquez dans cette liste de règles, les paquets transmis par le routeur vers une cible différente.

3.11.5. Exemples

Voici quelques exemples de configuration du filtrage de paquets.

Configuration par défaut de fli4l

Ci-dessous la configuration par défaut de la chaîne INPUT, cela nous permet d'atteindre la distribution fli4l :

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

Ainsi, nous obtenons, une

- autorisation pour l'accès au routeur des ordinateurs du réseau local (PF_INPUT_1='IP_NET_1 ACCEPT'),
- la communication local est autorisé sur le routeur (PF_INPUT_ACCEPT_DEF='yes'),
- les paquets appartenant à une connexions établies par le routeur seront acceptées (PF_INPUT_ACCEPT_DEF='yes'),
- tout le reste est rejeté (PF_INPUT_POLICY='REJECT'),
- rien ne sera écrit dans le journal du système (PF_INPUT_LOG='no').

Pour la chaîne FORWARD voici la configuration : seuls les paquets sur le réseau local et les paquets correspondant à une connexions établies par les ordinateurs du réseau local doivent être transmis. En outre, les paquets NetBIOS et CIFS seront rejetés.

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

Ce que l'on voit ici, est la dépendance de l'ordre des règles : en *premier* on rejette les paquets Netbios et *ensuite* les paquets du réseau local sont acceptés.

Maintenant, le réseau local communique avec le routeur, les paquets sont transmis, il ne manque que le masquage, il est nécessaire pour accéder au réseau privé sur Internet :

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

Trusted Nets

Si vous voulez mettre en place plusieurs sous-réseaux locaux qui puissent communiquer les uns avec les autres librement et sans être masqués, nous devons nous assurer que les paquets ne seront pas rejetés entre ces sous-réseaux et qu'ils ne soient pas masqués. Pour cela il suffit de rajouter une règle ou modifier l'existante.

Supposons que nous ayons un accès DSL via PPPoE, avec deux sous-réseaux IP_NET_1 (192.168.6.0/24) et IP_NET_2 (192.168.7.0/24). La configuration ressemblerait alors à ce qui suit :

```
PF_FORWARD_POLICY='REJECT'  
PF_FORWARD_ACCEPT_DEF='yes'  
PF_FORWARD_LOG='no'  
PF_FORWARD_N='4'
```

3. Configuration de la base

```
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'
```

Maintenant, les règles occupent les paquets qui sont acheminés entre les deux sous-réseaux sans examen plus approfondi. La troisième et quatrième règles font en sorte que les deux sous-réseaux sont disponible pour aller sur Internet. La première règle de la chaîne POSTROUTING assure que la communication entre les sous-réseaux se fait démasqué.

Alternativement, on peut dire que seuls les paquets qui dépassent par l'interface `pppoe` doivent être masqués :

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

De même, on pourrait limiter le filtrage des ports sur l'interface `pppoe` et les deux sous-réseaux peuvent être combinés en un seul, voici la configuration :

```
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'
```

Les paquets qui passent par l'interface `pppoe`, qui sont adressée au `udp` Ports 137-138 ou au `tcp` Ports 139 et 445 seront rejeté (règle 1), tous les autres paquets qui viennent du sous-réseau 192.168.6.0/23, sont transmis (la règle 2).

Route Network

Si vous voulez ajouter le réseau 10.0.0.0/24 dans le réseau existant (par ex. pour avoir un accès à distance sur ce réseau), de plus si vous voulez communiquer en étant démasqué et rejeter les paquets des `udp` Ports 137-138 et aussi `destcp` Ports 139 et 445, la configuration se présentera comme ceci :

```
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'
```

3. Configuration de la base

```
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'
```

- Règle 1 permet une communication claire entre les sous-réseaux IP_NET_1 et IP_NET_2.
- Règle 2 rejette les paquets pour les ports samba.
- Règle 3 et 4 permet la transmission de paquets provenant des sous-réseaux 192.168.6.0/24, 192.168.7.0/24 und 10.0.0.0/24, dans l'autre direction cela c'est déjà inclus dans le paramètre PF_FORWARD_ACCEPT_DEF='yes'
- Règle 1 la chaîne POSTROUTING garantit que les paquets ne sont pas masqués dans ou sur le sous-réseau 10.0.0.0/24

Une alternative à la configuration précédente :

```
PF_POSTROUTING_N='1'
```

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

Dans cette règle seul les paquets qui dépassent par l'interface **pppoe** doivent être masqués.

Liste noir, liste blanche

Listes noires (ou blacklists) (on refuse aux ordinateurs de cette liste "de faire quelque chose") et liste blanches (ou Whitelists) (on permet aux ordinateurs de cette liste "de faire quelque chose") la mise en place est en principe semblable. Les règles écrites au début de la liste sont très spécifiques et sont plus génériques vers la fin de la liste. Dans une liste noire les règles au début de la liste seront interdites de fait, quoi qu'il se soit et en fin de liste ils pourront faire quelque chose. Avec une liste blanche, c'est tout le contraire.

Exemple 1 : tous les ordinateurs du sous-réseau 192.168.6.0/24 peuvent accéder à Internet sauf l'ordinateur 12, ils ne pourront pas communiquer avec le protocole CIFS par les ports 137-138 (udp), 139 et 445 (tcp)

```
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='tmp1:samba DROP'
```

```
PF_FORWARD_3='192.168.6.0/23 ACCEPT'
```

```
PF_POSTROUTING_N='1'
```

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

Exemple 2 : l'ordinateur 12 peut accéder à Internet (mais on interdit toujours les Ports ...), tous les sous-réseaux locaux peuvent communiquer entre eux.

```
PF_FORWARD_POLICY='REJECT'
```

```
PF_FORWARD_ACCEPT_DEF='yes'
```

```
PF_FORWARD_LOG='no'
```

```
PF_FORWARD_N='3'
```


3. Configuration de la base

```
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. Configuration par défaut

Simple routeur masquant un réseau derrière lui

```
#
# Accès au routeur
#
PF_INPUT_POLICY='REJECT'
PF_INPUT_ACCEPT_DEF='yes'
PF_INPUT_LOG='no'
PF_INPUT_N='1'
PF_INPUT_1='IP_NET_1 ACCEPT'      # Tous les hôtes du réseau local
                                   # peuvent communiquer avec le routeur

#
# Accès à "Internet"
#
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'

PF_FORWARD_N='2'
PF_FORWARD_1='tmpl:samba DROP'   # Les paquets samba qui veulent
                                   # sortir du réseau sont rejetés
PF_FORWARD_2='IP_NET_1 ACCEPT'   # Tous les paquets du réseau local
                                   # peuvent sortir

#
# Masquage du réseau local
#
PF_POSTROUTING_N='1'
PF_POSTROUTING_1='IP_NET_1 MASQUERADE'  # Masque des paquets qui quittent
                                           # le sous-réseau
```

Simple routeur masquant deux réseaux derrière lui

```
#
# Accès au routeur
#
PF_INPUT_POLICY='REJECT'
PF_INPUT_ACCEPT_DEF='yes'
PF_INPUT_LOG='no'
PF_INPUT_N='2'
PF_INPUT_1='IP_NET_1 ACCEPT'      # Tous les hôtes du réseau local
                                   # peuvent communiquer avec le routeur
PF_INPUT_2='IP_NET_2 ACCEPT'      # Tous les hôtes du réseau local
```

3. Configuration de la base

```
# peuvent communiquer avec le routeur

#
# Accès à "Internet"
#
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'

#
# Libre communication entre les réseaux
#
PF_FORWARD_N='4'
PF_FORWARD_1='IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_FORWARD_2='tmpl:samba DROP' # Les paquets samba qui veulent
                                # sortir du réseau sont rejetés
PF_FORWARD_3='IP_NET_1 ACCEPT' # Tous les paquets du réseau local
                                # peuvent sortir
PF_FORWARD_4='IP_NET_2 ACCEPT' # Tous les paquets du réseau local
                                # peuvent sortir

#
# Masquage des réseaux locaux, la communication entre les réseaux
# ne sont pas masquées
#
PF_POSTROUTING_N='3'
PF_POSTROUTING_1='IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_POSTROUTING_2='IP_NET_1 MASQUERADE' # les paquets quittent le sous-réseau
                                         # masqués
PF_POSTROUTING_3='IP_NET_2 MASQUERADE' # les paquets quittent le sous-réseau
                                         # masqués
```

Masquage de deux réseaux derrière le routeur DSL avec un accès SSH/HTTP par Internet

```
#
# Accès au routeur
#
PF_INPUT_POLICY='REJECT'
PF_INPUT_ACCEPT_DEF='yes'
PF_INPUT_LOG='no'
PF_INPUT_N='4'
PF_INPUT_1='IP_NET_1 ACCEPT' # Tous les hôtes du réseau local
                              # peuvent communiquer avec le routeur
PF_INPUT_2='IP_NET_2 ACCEPT' # Tous les hôtes du réseau local
                              # peuvent communiquer avec le routeur
PF_INPUT_3='tmpl:ssh ACCEPT' # Permettre l'accès au service SSH
                              # depuis n'importe où
PF_INPUT_4='tmpl:http 1.2.3.4/24 ACCEPT' # Permettre aux ordinateurs
                                          # du sous-réseau d'avoir un accès
                                          # spécifique au service HTTP

#
```

3. Configuration de la base

```
# Accès à "Internet"
#
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'

#
# Pas de communication entre les réseaux, les deux réseaux peuvent
# avoir accès à Internet, paquets Samba sont rejetés
#
PF_FORWARD_N='2'
PF_FORWARD_1='tmpl:samba if:any:pppoe DROP' # Les paquets samba qui sortent
# du réseau sont rejetés
PF_FORWARD_2='if:any:pppoe ACCEPT' # Tous les autres paquets peuvent
# quitter le réseau local

#
# Masquage des réseaux locaux, la communication entre les réseaux
# ne sont pas masquées
#
PF_POSTROUTING_N='1'
PF_POSTROUTING_1='if:any:pppoe MASQUERADE' # Les paquets sont masqués
#en quitter le sous-réseau
```

Port Forwarding

La redirection de port peut être personnalisé avec la chaîne PREROUTING, vous devez paramétrer la règle de la façon suivante, dans (TARGET vous indiquez l'adresse IP de destination d'origine (optionnel) et le port de destination d'origine, dans NEW_TARGET vous indiquez la nouvelle adresse de destination et le nouveau port de destination (optionnel), dans PROTOCOL vous indiquez le protocole correspondant) :

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

Proxy transparent

Si vous souhaitez autoriser un accès spécifique à Internet, uniquement via un proxy local sans que le client s'en aperçoive, vous pouvez utiliser les chaînes PREROUTING et POSTROUTING. Fondamentalement, trois étapes sont nécessaires :

3. Configuration de la base

1. Les requêtes qui arrivent sur le port http seront détournées sur le proxy (PREROUTING).
2. Modification des paquets du proxy pour les rediriger, afin que le routeur pense que les paquets viennent de lui, si bien qu'il les renvoie à nouveau (POSTROUTING).
3. Les paquets passent à travers la chaîne (PREROUTING) s'il n'existe pas de règle dans la chaîne (FORWARD)

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

Exemple 1 : supposons que nous ayons un seul réseau IP_NET_1, sur lequel on a installé Squid sur un ordinateur avec le nom **proxy** et que vous voulez router le trafic **http** sur cette ordinateur. Squid écoute sur le port 3128. Par souci de simplicité, nous allons nous référer au nom d'hôte @proxy enregistré dans HOST_1_NAME='proxy' (voir [Domaine de configuration](#) (Page 64)).

L'ensemble devrait ressembler à ceci :

```
...
PF_PREROUTING_x='@proxy ACCEPT'
    # Les paquets du Proxy ne doivent pas être détournés

PF_PREROUTING_x='prot:tcp IP_NET_1 80 DNAT:@proxy:3128'
    # Les paquets HTTP de IP_NET_1 allant sur n'importe quelle destination
    # seront redirigés vers @proxy, Port 3128

PF_POSTROUTING_x='any @proxy:3128 SNAT:IP_NET_1_IPADDR'
    # Tous les paquets du Proxy sur le Port du 3128 seront réécrits
    # comme s'ils venaient de fli4l (IP_NET_1_IPADDR)

PF_FORWARD_x='prot:tcp @proxy 80 ACCEPT'
    # La règle de la chaîne FORWARD laisse passés les paquets HTTP du proxy (si nécessaire)
...
```

Il peut y avoir plusieurs conflits potentiels avec d'autres réseaux ou de redirection de port (se n'est rien d'autre qu'une règle DNAT), il va falloir formuler encore plus rigoureusement les règles.

Exemple 2 : notre Proxy qui s'appelle **proxy** se trouve dans le réseau IP_NET_1, il écoute sur le port 3128 et sera efficace uniquement pour les clients du réseau IP_NET_1. Le réseau IP_NET_1 est accessible via l'interface IP_NET_1_DEV. Les paquets provenant des autres réseaux ne seront pas pris en considération.

```
...
PF_PREROUTING_x='if:IP_NET_1_DEV:any !@proxy 80 DNAT:@proxy:3128'
    # Les demandes vers le port HTTP, ne viennent pas du proxy, mais via
    # l'interface interne (IP_NET_1_DEV) et rediriger vers le port proxy.
    # A ce stade, le contrôle if:IP_NET_1_DEV:any est important pour vérifier
    # si les paquets viennent bien de l'intérieur, autrement les paquets
    # seraient également diriger vers l'extérieur (faille de sécurité~!)

PF_POSTROUTING_x='prot:tcp IP_NET_1 @proxy:3128 SNAT:IP_NET_1_IPADDR'
    # Les paquets HTTP provenant de IP_NET_1 sont réécrits pour être envoyés
    # sur le port 3128 du proxy, comme s'ils venaient de fli4l (IP_NET_1_IPADDR)
```

3. Configuration de la base

```
PF_FORWARD_x='prot:tcp @proxy 80 ACCEPT'
# La règle de la chaîne FORWARD laisse passer les paquet http du Proxy (si nécessaire)
...
```

Exemple 3 : pour vous rendre la vie plus facile et pour rendre les règles un peu plus courtes, vous pouvez également utiliser le modèle (voir [Modèle pour le filtrage de paquets](#) (Page 43)). A ce stade `tmpl:http` est utilisé, il se traduit par `prot:tcp any any:80`. Par exemple à partir de `tmpl:http IP_NET_1 DNAT:@proxy:3128` ou alors `prot:tcp IP_NET_1 80 DNAT:@proxy:3128`.

Les deux réseaux `IP_NET_1` et `IP_NET_2` doivent être transparent sur le Proxy. Cela pourrait vous aider à simplifier l'écriture

```
...
PF_PREROUTING_x='tmpl:http @proxy ACCEPT'
# Les paquets http ne doivent pas être détournés

PF_PREROUTING_x='tmpl:http IP_NET_1 DNAT:@proxy:3128'
# Les paquets HTTP provenant de IP_NET_1 sont détournés

PF_PREROUTING_x='tmpl:http IP_NET_2 DNAT:@proxy:3128'
# Les paquets HTTP provenant de IP_NET_2 sont détournés

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'
...
```

Cela peut se poursuivre indéfiniment ...

3.11.7. DMZ – Zone démilitarisée

`fi4l` permet également la construction d'une simple DMZ. Tout d'abord, vous pouvez voir sur le site wiki <https://ssl.networks.org/wiki> un exemple de configuration.

3.11.8. Conntrack Helpers

Bien que l'utilisation du masquage d'IP a l'avantage que plusieurs ordinateurs du réseau local peuvent être acheminés via une adresse IP publique, mais il y a aussi des inconvénients que vous devez prendre en compte.

Le gros problème est, par exemple, qu'aucun ordinateur de l'extérieur ne peut se connecter à un ordinateur du réseau local. C'est souhaitable pour des raisons de sécurité, mais certains protocoles ne fonctionnent pas car ils requièrent une connexion depuis l'extérieur.

Un exemple classique le FTP. En plus du canal de communication, les commandes et les réponses sont échangés sur un autre canal (sous la forme d'un port-IP) pour envoyer les données. `fi4l` utilise pour cela Conntrack Helper qui permet de transmettre les ports supplémentaires, ils sont utilisés pour déverrouiller le ad hoc et aussi pour les ordinateurs internes quand c'est nécessaire. Conntrack Helper "écoute" le flux de données afin de détecter si un port supplémentaire est nécessaire.

3. Configuration de la base

Les applications typiques pour Conntrack Helper sont le protocole pour le Chat et pour les jeux sur Internet.

Vous activez Conntrack Helper avec des règles et un ensemble de variables spécifiques. Dans la liste de règles PF_PREROUTING_CT_% les affectations contiendra les Helpers pour les paquets venant de l'extérieur, dans la liste de règles PF_OUTPUT_CT_% les affectations contiendront les Helpers pour les paquets générés par le routeur. Quelques exemples pratiques viendront illustrer cela.

Exemple 1 : si vous voulez autoriser le mode FTP actif sur le réseau local, le routeur sera visible à partir de cette connexion par les routeurs extérieur, pour cela vous devez créer une règle dans la chaîne PF_PREROUTING_CT_% comme ceci :

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

pour toutes les connexions TCP depuis le réseau local (IP_NET_1) vers toute autre adresse sur le port 21 (c'est le Port du ftp) le module auxiliaire ftp est chargé. Ce module permet alors de se connecter, le serveur FTP peut établir une connexion vers client pour le transfert de données, un "trou" est temporairement ouvert dans le pare-feu.

Exemple 2 : avec le mode FTP passif, il vous permet d'activer un serveur FTP sur le réseau local (ainsi une connexion de données sera établie vers l'extérieur, là aussi un trou dans le pare-feu sera ouvert), ici le routeur sera également visible à partir de cette connexion par les routeurs extérieur. La règle est représentée de la manière suivante :

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

Cette règle se traduit de la manière suivante, toutes les connexions FTP seront envoyés à l'adresse dynamique du routeur, associé à Conntrack Helper du FTP. Ici *dynamic* a été utilisé car il est supposé que le routeur est responsable de la connexion Internet et a donc une adresse IP externe. Si le routeur effectue une connexion via DSL, la règle peut aussi s'écrire :

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

Cette règle se traduit de la manière suivante, toutes les connexions FTP sur l'interface DSL (pppoe), seront associées à Conntrack Helper du FTP.

Si le routeur ne se connecte pas, il est par exemple derrière un autre routeur (box FRITZ!, modem câble, etc), la règle suivante peut être utilisée :

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

On suppose que dans l'exemple la connexion s'effectue via une interface vers l'autre routeur, cette interface est associée au deuxième sous-réseau (IP_NET_2_DEV).

On notera bien sûr, *en plus* de la configuration il sera nécessaire de paramétrer la chaîne FORWARD, pour réellement faire parvenir les paquets FTP. Voici une règle typique :

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

3. Configuration de la base

On suppose que l'hôte sur lequel le programme du serveur FTP fonctionne, a le nom `ftpserver`.

Exemple 3 : enfin, le must, si vous souhaitez utiliser le mode FTP actif directement à partir de `fi4l` (avec l'aide du programme `ftp` qui est dans le paquetage `tools`). Le pare-feu doit être préparé pour cela, cette fois on utilisera la chaîne `OUTPUT` et la liste de règles `PF_OUTPUT_CT_%` ou l'on va configurer les règles :

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

Cette règle n'est toutefois pas nécessaire si la variable `FTP_PF_ENABLE_ACTIVE='yes'` est activée – S'il vous plaît reportez-vous à la documentation du `OPT_protocolFTP` dans le paquetage `tools`

Voici un aperçu de l'actuel Conntrack Helpers :

Helpers	Explication
<code>ftp</code>	File Transfer Protocol
<code>h323</code>	H.323 (Voice over IP)
<code>irc</code>	Internet Relay Chat
<code>pptp</code>	PPTP Masquerading (Ce module peut être plus qu'un client PPTP il fonctionne simultanément derrière un routeur <code>fi4l</code> .)
<code>sip</code>	Session Initiation Protocol
<code>sane</code>	SANE Network Protocol
<code>snmp</code>	Simple Network Management Protocol
<code>tftp</code>	Trivial File Transfer Protocol

TABLE 3.8. – Disponibilité de Conntrack Helpers dans le filtrage de paquets

Voici un aperçu des variables à configurer :

PF_PREROUTING_CT_ACCEPT_DEF Si cette variable est sur 'yes', les règles par défaut sont générés, elles sont nécessaires pour le bon fonctionnement du routeur. Vous devriez indiquer 'yes' pour l'utilisation par défaut.

PF_PREROUTING_CT_N PF_PREROUTING_CT_x PF_PREROUTING_CT_x_COMMENT
Vous indiquez dans cette liste de règles, les paquets entrants à partir du routeur seront connectés par Conntrack Helpers.

PF_OUTPUT_CT_ACCEPT_DEF Si cette variable est sur 'yes', les règles par défaut sont générés, elles sont nécessaires pour le bon fonctionnement du routeur. Vous devriez indiquer 'yes' pour l'utilisation par défaut.

PF_OUTPUT_CT_N PF_OUTPUT_CT_x PF_OUTPUT_CT_x_COMMENT
Vous indiquez dans cette liste de règles, les paquets qui sont générés par le routeur pour une connexion au routeur avec Conntrack Helpers.

3.12. Configuration du domaine

Dans un LAN les ordinateurs Windows ont des caractéristiques désagréables : si vous avez besoin d'utiliser un serveur de nom (ou DNS), vous devez configurer vos PC Windows pour ce service. Le problème c'est que les PCs Windows questionnent à intervalle régulier le serveur – même si personne n'utilise l'ordinateur ! Si vous configurez un serveur-DNS sur Internet pour votre PC Windows, cela pourrait revenir très cher ...

Le truc est le suivant : s'il n'y a pas de serveur DNS disponible sur le LAN (ou réseau local), on peut utiliser le routeur fli4l comme serveur DNS.

DNSMASQ est utilisé en tant que serveurs DNS.

Avant que nous commençons la configuration du DNS, vous devez d'abord réfléchir au nom de domaine et aux noms des PCs avant de les écrire dans votre réseau local. Le nom de domaine que vous emploierez ne sera pas visible sur Internet. Ainsi vous êtes libre pour employer presque n'importe quel nom de domaine.

Vous devez donner un nom à chaque ordinateur de Windows dans votre réseau. En outre le routeur-fli4l doit avoir connaissance de ces noms.

DOMAIN_NAME Configuration par défaut : `DOMAIN_NAME='lan.fli4l'`

Dans la version fli4l le nom de domaine "lan.fli4l" est paramétré par défaut. Vous êtes libre d'écrire votre propre nom de domaine. Vous devez éviter d'utiliser un nom qui pourrait exister sur Internet. Si vous employez un nom de domaine existant (par ex. france3.fr), vous ne pourrez pas accéder à ce domaine.

DNS_FORWARDERS Configuration par défaut : `DNS_FORWARDERS=""`

On indique dans cette variable l'adresse IP du serveur DNS de votre Fournisseur Accès Internet (ou FAI), si fli4l est utilisé comme routeur pour Internet. Le routeur fli4l expédie à cette adresse toutes les requêtes DNS auxquelles il ne peut pas répondre.

Vous pouvez entrer plusieurs adresses IP pour les serveurs DNS, vous devez séparer toutes les adresses IP par un espace (ou un blanc).

Si plusieurs serveurs DNS sont configurés, les requêtes DNS seront utilisés dans l'ordre de configuration des serveurs, ainsi le deuxième serveur spécifié sera utilisé seulement si le premier n'a pas répondu à la requête DNS, etc.

Il est également possible d'ajouter en option un numéro de port à l'adresse IP, pour cela, il faut séparer l'adresse IP et le port par deux points. Toutefois, il est nécessaire d'activer la variable `OPT_DNS='yes'` (Page ??) dans le (paquetage `dns_dhcp` (Page ??)), cependant, cette variable ne doit jamais être substituée à la variable `*_USEPEERDNS`.

Attention :

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

L'une de ces variables doit être paramétrées sur (`=yes`), cela est nécessaire pour que le serveur DNS externe soit enregistré, sinon après le démarrage du routeur aucune résolution de nom ne sera possible. Le serveur DNS externe ne fonctionnera pas.

Exception : si vous configurez le routeur fli4l dans un réseau local *sans* connexion Internet ou dans un réseau avec un serveur DNS supplémentaire (réseau d'entreprise). Dans ce cas vous devez paramétrer l'adresse IP 127.0.0.1 pour empêcher le forwarding (empêcher les connexions extérieure).

HOSTNAME_IP (optionnelle)

Avec cette variable optionnelle vous pouvez définir, le réseau 'IP_NET_x' qui sera rattaché au HOSTNAME (ou nom d'Hôte).

HOSTNAME_ALIAS_N (optionnelle)

Dans cette variable vous indiquez, le nombre d'alias (ou de surnom) supplémentaire pour le routeur.

HOSTNAME_ALIAS_x (optionnelle)

Dans cette variable vous indiquez, l'alias pour le routeur.

3.13. Configuration de Imond

OPT_IMOND Configuration par défaut : OPT_IMOND='no'

En mettant OPT_IMOND sur 'yes' vous décidez, d'activer le serveur imond. Imond est le centre de surveillance et le contrôle du moindres coût des connexions pour le routeur fli4l. Par conséquent, un chapitre supplémentaire lui est consacré [Description d'imon](#) (Page 87).

Important : le dispositif LC-Routing de fli4l peut uniquement être utilisé par imond. Les commutations basées sur le temps de connexion n'est pas possible, sans utiliser imond !

Avec le routage ISDN (RNIS en France) et DSL il est nécessaire d'employer la version 1.5 d'imon. Pour l'activer, vous paramétrez OPT_IMOND='yes'.

Si fli4l est uniquement utilisé comme routeur avec deux cartes réseaux, vous devez paramétrer la variable sur OPT_IMOND='no'.

IMOND_PORT Dans cette variable on indique le port TCP/IP d'écoute pour les connexions, La valeur par défaut '5000' il doit être modifié uniquement par nécessité.**IMOND_PASS** Configuration par défaut : IMOND_PASS=""

Dans cette variable on peut placer un mot de passe spécial utilisateur pour imond. Si un client se connecte sur le port 5000, imond demandera le mot de passe avant qu'il réponde à n'importe quelle commande. Excepté : Les commandes "quit", "help" et "pass" si IMOND_PASS est vide, aucun mot de passe est demandé.

Le client peut exécuter les instructions déterminées, comme les commandes Dial, Enable, Reboot, la commutation de la route par défaut peut déjà exécuté ou pour entrer les mots de passe administrateur nécessaire, la commuter la route par défaut, vous pouvez paramétrer les variables

- [IMOND_ENABLE](#) (Page 66),
- [IMOND_DIAL](#) (Page 66),
- [IMOND_ROUTE](#) (Page 66) et
- [IMOND_REBOOT](#) (Page 66)

voir plus bas.

IMOND_ADMIN_PASS Configuration par défaut : IMOND_ADMIN_PASS=""

En utilisant le mot de passe admin, le client reçoit toutes les droits et peut donc utiliser toutes les commandes du serveur imond, plus exactement les variables IMOND_ENABLE, IMOND_DIAL etc indépendamment. Si vous laissez IMOND_ADMIN_PASS vide il n'est pas possible d'utiliser ces commandes, le mot de passe utilisateur est nécessaire pour avoir tous des droits !

3. Configuration de la base

IMOND_LED Maintenant imond peut indiquer le statut Online/Offline par une LED. Elle est branché sur un port COM de la manière suivante :

Connecteur 25-broches :

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

Connecteur 9-broches :

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

S'il y a une connexion établie avec ISDN (RNIS) ou la DSL, la LED est allumée, autrement elle est éteinte. Si vous voulez inverser l'éclairage de la LED, vous devez inverser la polarité de celle-ci. Dans le cas où la luminosité de la LED serait trop faible, vous pouvez réduire la résistance à 470 ohms.

Il est possible de relier deux LED de couleurs différente. La deuxième LED peut être reliée en utilisant une deuxième résistance entre DTR et GND, et inverser la polarité par rapport à la première. Une des deux LED sera allumée selon le statut. Vous pourrez employer une LED-DUO (deux couleurs, trois connecteurs).

Actuellement, le connecteur RTS de l'interface série comporte un comme DTR. Vous pourriez relier une autre LED a ce connecteur (RTS), pour le statut d'affichage Online/Offline. Cela pourrait changer dans les versions futures de fli4l.

Dans la variable **IMOND_LED** un port COM doit être indiqué, c-à-d. 'com1', 'com2', 'com3' ou 'com4'. Si aucune LED n'est branchée, vous devriez laisser la variable vide.

IMOND_BEEP Si la variable est placée sur **IMOND_BEEP='yes'**, imond produit deux signaux sonores par le PC = haut-parleur, selon l'état de Offline à Online et vice versa. Dans le premier cas un ton grave, dans le second cas un ton aigu. Lors de la modification de l'état offline le son sera d'abord le plus élevé, il émettra ensuite un son plus faible.

IMOND_LOG Configuration par défaut : **IMOND_LOG='no'**

Si la variable est placée sur **IMOND_LOG='yes'**, les connexions sont enregistrées dans **/var/log/imond.log**. Ce fichier peut être recopié pour être analyse sur un ordinateur du réseau local (LAN), en utilisant par exemple le programme SCP. Si vous voulez utiliser SCP vous devrez installer le paquetage sshd et le configurer, de sorte que le programme SCP soit également disponible.

La description des formats du fichier journal est décrit dans le tableau 3.9.

Les frais sont donnés en Euro. Il est important pour cette fonction que la variable **ISDN_CIRC_x_TIMES** (Page ??) du circuit soient correctement réglée.

IMOND_LOGDIR Si vous activez l'enregistrement les connexions, vous pouvez avec la variable **IMOND_LOGDIR** enregistrer un répertoire alternatif au lieu d'utiliser **/var/log**, vous pouvez indiquez par exemple **'/boot'**. Le fichier journal **imond.log** sera ensuite créé sur le média de démarrage. De plus le média doit être en «lecture/écriture». Par défaut la variable est sur **'auto'** l'emplacement est déterminé automatiquement. Selon la configuration du **FLI4L_UUID** le chemin déterminé se trouvera alors sous **/boot/persistent/base** ou un autre chemin. Si le chemin **/boot** n'est pas en lecture/écriture, le répertoire **persistent** avec **FLI4L_UUID** ne sera pas actif et le fichier log sera enregistré dans le répertoire **/var/run**.

IMOND_ENABLE IMOND_DIAL IMOND_ROUTE IMOND_REBOOT Dans ces variables, vous pouvez activer plusieurs commandes qui seront utilisées par le clients **imonc** et envoyées au serveur **imond**, elles seront exécutées en mode utilisateur.

3. Configuration de la base

TABLE 3.9. – Format du fichier Log d’Imond

Les entrées	Signification
Circuit	Nom du Circuit dans lequel les notations d’événement sont produites
Heure connexion	Date et heure où la connexion a été établie
Heure déconnexion	Date et heure où la déconnexion a été terminée
Temps en ligne	Durée de connexion
Temps restant	Temps de connexion restante offre du FAI/mois (dépend du réglage des unités)
Coût (prix)	Prix du temps de connexion du FAI
Bande passante	Bande passante utilisée, les valeurs sont représentée séparément une pour les entrées l’autre pour les sorties, et seront additionnées dans la bande passante = $4Gio * <premier\ chiffre> + <deuxième\ chiffre>$
Device	Périphérique utilisé pour la communication
Relevé d’unités de compte	Unité consultées par le fournisseur d’accès pour le relevé de compte (données à configurer)
Prix de l’unité	Prix de l’unité par connexion (des données à configurer)

Avec ces commandes vous pouvez, par l’intermédiaire du serveur imond allumer/éteindre l’interface ISDN, composer/raccrocher, ajouter une nouvelle route par défaut et redémarrer le routeur.

Configuration par défaut :

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

Des dispositifs additionnels pour l’interface imond sont décrits [dans ce chapitre](#) (Page 87) pour le client et le serveur.

3.14. Configuration du circuit général

IP_DYN_ADDR Si une connexion avec une IP dynamique est utilisée, vous devez placée la variable IP_DYN_ADDR sur ‘yes’, ou sur ‘no’ si statique. La plupart des fournisseurs d’accès utilisent une IP dynamique.

Configuration par défaut : IP_DYN_ADDR=‘yes’

DIALMODE Par défaut fli4l utilise ‘auto’ pour le mode de numérotation, c-à-d. une connexion sera établie automatiquement dès qu’un ordinateur du réseau local essayera d’accéder à une adresse IP extérieure par ex. Internet. Il est également possible de spécifier le modes de connexion ‘manual’ ou ‘off’. Dans ce cas, la connexion peut uniquement être déclenchée en utilisant le client imonc.

Configuration par défaut : DIALMODE=‘auto’

4. Les paquetages

En plus de l'installation de la base (BASE) il existe d'autres paquetages. Il s'agit notamment de "OPTs"¹ supplémentaire, qui peuvent au besoin être installés dans la base. Certains de ces OPTs sont intégrés dans le paquetage base, les autres sont à télécharger part. Une vue d'ensemble les paquetages fournis par l'équipe fli4l peuvent être téléchargés sur cette page Web (<http://www.fli4l.de/fr/telechargement/version-stable/>). D'autres paquetages créés par des concepteurs privés peuvent être trouvés dans la banque de données OPT (http://extern.fli4l.de/fli4l_opt-db3/). Nous allons voir dans les paragraphes suivant une description des paquetages créés et utilisés par l'équipe fli4l.

4.1. Outils dans le paquetage base

Dans le paquetage base vous trouverez les OPTs suivants :

Nom	Description
OPT_SYSLOGD	Programme qui enregistre tous les messages (Page 68)
OPT_KLOGD	Programme qui enregistre les messages Kernel (Page 70)
OPT_LOGIP	Programme qui enregistre les protocoles IP WAN (Page 70)
OPT_Y2K	Correctif pour les ordinateurs avant l'année 2K (Page 70)
OPT_PNP	Outil pour l'installation des cartes ISAPnP (Page 71)
OPT_HOTPLUG_PCI	Activation du PCI hot-plugging (Page 72)

4.1.1. OPT_SYSLOGD - Enregistre tous les messages du système

Beaucoup de programmes utilisent l'interface de syslogd pour visualiser les messages du système. Pour rendre les messages visibles, installer le démon syslogd.

Si vous voulez voir les messages debug, placer OPT_SYSLOGD sur 'yes', si vous ne voulez pas de message sur 'no'.

Voir également ISDN_CIRC_x_DEBUG (Page ??) et PPPOE_DEBUG (Page ??).

Configuration par défaut : OPT_SYSLOGD='no'

SYSLOGD_RECEIVER Avec la variable SYSLOGD_RECEIVER on peut définir, si fli4l doit recevoir ou non les messages syslog par le réseau.

SYSLOGD_DEST_N SYSLOGD_DEST_x Avec la variable SYSLOGD_DEST_x on indique les emplacements, où vous voulez voir les messages système enregistrés par l'interface syslogd. Normalement c'est sur la console de fli4l que l'on voit les messages :

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

Si vous souhaitez utiliser un fichier pour enregistrer les messages :

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

1. abréviation pour "module OPTionnel"

4. Les paquetages

Si un hôte dans le réseau veut lire les messages, vous pouvez réorienter des messages vers cette ordinateur – en indiquant l'adresse IP.

Exemple :

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

Il faut préfixer par le caractère @ avant d'écrire l'adresse IP ou le nom hôte.

Si vous voulez envoyer les messages sur différent système, il est nécessaire d'augmenter le nombre dans la variable SYSLOGD_DEST_N (nombre de description) et de remplir les variables en conséquence, par ex. SYSLOG_DEST_1, SYSLOG_DEST_2 etc.

Les caractères '*. *' désignent l'ensemble des services et des priorités des messages, on peut limiter les priorités pour une "destination" déterminée. Dans ce cas, on remplace l'étoile après le point Par l'un des mots clés suivant :

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

L'ordre dans la liste reflète le "poids" des annonces. Les mots clés "error", "warn" et "panic" sont obsolètes et ne devaient plus être utilisés ils sont remplacés par err, warning et emerg.

Vous pouvez remplacer l'astérisque (*) devant le point par un soi-disant "sélecteur", cependant il serait trop long d'expliquer ici tous des paramètres. Le lecteur peut essayer trouver toutes les informations nécessaires sur un moteur de recherche. Vous pouvez voir la configuration dans le manuel de syslog.conf :

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

<http://okki666.free.fr/docmaster/articles/linux068.htm>

Normalement, l'astérisque, est tout à fait suffisant. Exemple :

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

Non seulement les ordinateurs Unix et Linux, mais aussi les ordinateurs Windows peuvent servir d'hôte pour les logs (ou fichiers journal). Sur <http://www.fli4l.de/fr/divers/liens/> vous trouverez des liens pour avoir des logiciels appropriés. L'application d'un serveur log est recommandée, pour l'enregistrement détaillé des protocoles, l'enregistrement des protocoles aide également au dépistage des erreurs. Le protocole syslog est aussi compatible avec imonc client Windows et peut ainsi recevoir les messages log.

Malheureusement, les informations de Boot fli4l ne peuvent pas être enregistrées avec le démon syslogd. Toutefois, on peut configurer fli4l pour que les informations de Boot puissent sortir sur une console de terminal série (voir [Configuration de la console](#) (Page 28)).

SYSLOGD_ROTATE Vous pouvez définir avec la variable SYSLOGD_ROTATE si fli4l doit faire une rotation des messages syslog une fois par jour. Ainsi les derniers messages seront enregistrés tous les x jours.

SYSLOGD_ROTATE_DIR La variable SYSLOGD_ROTATE_DIR est optionnelle, vous pouvez définir ici le répertoire pour l'enregistrement des fichiers syslog de rotation. Si cette variable est vide, le répertoire par défaut /var/log sera utilisé.

SYSLOGD_ROTATE_MAX La variable SYSLOGD_ROTATE_MAX est optionnelle, elle vous permet de spécifier un nombre d'enregistrements par rotation des fichiers syslog.

SYSLOGD_ROTATE_AT_SHUTDOWN La variable SYSLOGD_ROTATE_AT_SHUTDOWN est optionnelle, elle vous permet de désactiver la rotation du fichier syslog lors d'un arrêt du routeur. Attention vous ne pouvez pas désactiver la rotation, si vos fichiers syslog sont écrits directement vers une destination permanente.

4.1.2. OPT_KLOGD – Messages du Kernel lors du boot

Parfois des erreurs apparaissent lors du Boot de Linux Kernel, ils sont écrits directement sur la console (ou écran) et il est difficile de les visualiser. En utilisant OPT_KLOGD='yes' ces messages sont réorientés sur le syslogd, ils peuvent être soit expédiés sur un client log ou écrits dans un fichier voir ci-dessus. Ainsi nous ne sommes pas obligés de surveiller la console.

Il est recommandé de paramétrer : OPT_SYSLOGD='yes' et aussi de paramétrer OPT_KLOGD='yes'.

Configuration par défaut : OPT_KLOGD='no'

4.1.3. OPT_LOGIP – Journalisation des adresses IP WAN

Avec LOGIP il est possible, d'enregistrer les messages IP WAN dans un fichier journal pour cela il faut activer la variable OPT_LOGIP='yes'.

Configuration par défaut : OPT_LOGIP='no'

LOGIP_LOGDIR - Définit le répertoire des fichiers LOG

Avec la variable LOGIP_LOGDIR on définit le répertoire, dans lequel les fichiers Log sont créés ou 'auto' pour l'autodétection.

Configuration par défaut : LOGIP_LOGDIR='auto'

4.1.4. OPT_Y2K – Correctif pour avant l'année 2000

Dans la plupart des cas les routeurs fli4l sont assemblés avec du vieux matériel. Parfois les cartes mères ne sont pas compatibles pour passer l'année 2000. Lorsque vous réglerez la date du 27/05/2000 dans le BIOS, au prochain démarrage la date dans le BIOS sera peut-être indiquée 27/05/2094 ! Et dans Linux elle sera indiquée 27/05/1994 :-)

Si votre date n'est pas correcte il n'y a pas vraiment d'importance pour le routeur fli4l. Mais si le routeur est utilisé en tant que gestionnaire de coût (ou frais de connexion Internet), cela est important.

La raison : le 27/05/1994 est un vendredi et le 27/05/2000 est un samedi et les week-ends les prix des connexions Internet et/ou les fournisseurs sont meilleur marché. ...

La première alternative : vous indiquez dans la BIOS la date du 28/05/1994, au lieu du 27/05/2000, qui est un samedi. Mais le problème n'est pas complètement résolu. parce que fli4l utilise non seulement la semaine mais aussi l'heure actuelle pour le réglage du LC-routage, il prend également en compte les jours fériés.

Y2K_DAYS – Ajouter N jours à la date du système

Puisque la différence exacte de la date est de 2191 jours par rapport à la date réelle, on peut indiquer :

```
Y2K_DAYS='2191'
```

En ajoutant 2191 jours à la date du BIOS, la date dans Linux sera à jour. Cependant, la date du BIOS ne doit pas être modifier. Autrement la date sera remise à zéro à 2094 (ou à 1994) au prochain démarrage :-)

Il y a une autre alternative :

En accédant à un serveur de temps fli4l peut rechercher la date et l'heure exacte sur Internet. Pour cela vous avez le paquetage CHRONY (Page ??) qui fait cette recherche. Vous pouvez combiner les deux variables, ainsi la date sera corrigée en utilisant Y2K_DAYS et l'heure exacte sera recherché sur le serveur de temps.

Si vous n'avez aucun problème lié à Y2K, placer OPT_Y2K='no' et oublier ce fonction ...

4.1.5. OPT_PNP – Installation des cartes ISAPnP

Quelques cartes ISAPnP doivent être configurées en utilisant l'outil isapnp. cela concerne les cartes ISDN du ISDN_TYPE 7, 12, 19, 24, 27, 28, 30 et 106 – Mais uniquement si vous avez vraiment une carte ISAPnP.

Au démarrage, il est nécessaire de créer un fichier de configuration etc/isapnp.conf.

Voici une description courte pour le créer :

- Dans le fichier <config>/base.txt régler les variables comme ceci OPT_PNP='yes' et MOUNT_BOOT='rw'
- La carte ISAPnP ne sera probablement pas identifiée au boot (ou démarrage)
- Écrire sur la console du routeur fli4l :

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

Maintenant la configuration doit être sauvegardée sur un média de boot.

On continue la configuration sur le PC (Unix/Linux/Windows) :

- Copier le fichier isapnp.conf depuis le média de boot dans le répertoire <config>/etc/isapnp.conf de votre PC
- Ouvrez isapnp.conf avec un éditeur de texte

On peut garder les valeurs avancées ou remplacer ces valeurs par d'autres. Les lignes suivantes sont importantes dans l'exemple qui suit :

```
#      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) – Dans la «BASE» on indique les adresses Minimum et Maximum utilisées, on doit toujours prendre en considération «l'alignement de base».

Si vous avez plus d'une carte ISA dans votre système, vous devez toujours vérifier s'il n'y a pas d'intersection entre les adresses et faire attention à la quantité d'adresses nécessaire (number of addresses required).

- 2) – La liste des IRQ suivante est un mauvais choix pour le paramétrage de la carte ISA. 2, (9), 3, 4, 5 et 7 ils sont normalement utilisé par le système, les ports serie, le

port parallèle, ect.

On ne peut pas diviser une IRQ pour plusieurs cartes ISA, c'est pourquoi on ne doit jamais utiliser une IRQ déjà occupée.

- Copier les valeurs (IRQ/IO) et les écrire dans le fichier <config>/isdn.txt
- Il est nécessaire de régler la variable OPT_PNP dans <config>/base.txt sur 'yes' autrement les fichiers ne seront pas copiés sur le média de boot. Vous pouvez remodifier la variable MOUNT_BOOT selon votre choix.
- Créer un nouveau média de boot

Le fichier qui a été généré automatiquement est sauvegardé dans le format Unix et ne contient aucun CRs (ou Retours Chariots). Si on lance l'éditeur Notepad sous Windows on verra le fichier sur une seul ligne. L'éditeur Notepad sous DOS "édite" et peut traiter des fichiers Unix. Il faut le sauvegarder comme un fichier DOS avec les CRs.

Remède :

- Lancer la boîte de commande DOS
- Charger le répertoire <config>/etc
- Entrer : edit isapnp.conf
- Éditer le fichier et sauvegardez le

Ensuite, on peut travailler sur le fichier avec Notepad.

On peut aussi utiliser simplement l'éditeur de Wordpad sous Windows.

De plus les CRs générés sont filtrés, ils ne causeront pas de problème lors du Boot de fli4l.

Au début, vous devriez essayer sans activer OPT_PNP. Au cas où la carte ne serait pas identifiée, suivre la procédure décrite ci-dessus.

Quand vous installez une nouvelle version fli4l, vous pouvez récupérer le fichier isapnp.conf qui a été créé, il ne doit pas être créé à nouveau, mais peut être réutilisé.

Configuration par défaut : OPT_PNP='no'

4.1.6. OPT_HOTPLUG_PCI – Activation du PCI hot-plugging

Si vous activez cette variable OPT_HOTPLUG_PCI='yes' les modules seront copiés dans fli4l et chargés au démarrage ainsi le PCI hot-plugging (ou branchement PCI à chaud) sera activé, c'est à dire que vous pourrez ajouter ou retirer des cartes PCI pendant que le système est en marche. Pour que cela fonctionne un contrôleur PCI hot-plug doit être présent sur l'ordinateur.

Cette option ne doit *pas* être activé pour ajouter ou supprimer des périphériques virtuels dans un *environnement de virtualisation* comme KVM, puisque cela se fait par l'intermédiaire du mécanisme ACPI et que les pilotes ACPI sont activés en permanence dans le Kernel.

5. Création une archive fli4l/Média de Boot

Lorsque tous les fichiers de configuration seront paramétrés, l'archive fli4l/Média de Boot peut être construite, on peut soit utiliser une carte Compact Flash pour booter ou créer une image ISO, soit uniquement faire une mise à jour des fichiers.

5.1. Création de l'archive fli4l/Média de Boot sous Linux, dérivé Unix et Mac OS X

La construction se fait à l'aide du Scripts (`.sh`) qui se trouve dans la racine du répertoire de fli4l.

```
mkfli4l.sh
```

Build-Script (ou script de construction) reconnaît indépendamment les différentes [variantes de Boot](#) (Page 23).

La simple commande sous Linux est :

```
sh mkfli4l.sh
```

Les trois mécanismes suivant gèrent le démarrage de Build-Scripts :

- La configuration de la variable `BOOT_TYPE` dans le fichier `<config>/base.txt`
- La configuration du fichier `<config>/mkfli4l.txt`
- Les paramètres du Build-Scripts

On décide au moyen de la variable `BOOT_TYPE` (Page 23), le type de support de construction (Build-Scripts) pour fli4l :

- Démarrer fli4l avec un CD-ROM par une image ISO
- Faire une mise à jour des fichiers, pour une nouvelle version fli4l
- Créer les fichiers fli4l et faire une mise à jour à distance via SCP
- etc.

Vous trouverez la description des variables dans le fichier de configuration `<config>/mkfli4l.txt` et dans le chapitre [Paramètres mkfli4l.txt](#) (Page 80).

5.1.1. Lignes de commandes optionnelle

Les mécanismes de contrôle sont à ajouter aux paramètres d'option lorsque vous appelez le script de compilation par ligne de commande. Les options de contrôle sont semblables à ceux du fichier de commande `mkfli4l.txt`. Les spécifications des paramètres d'options remplacent les valeurs du fichier de contrôle. Pour des raisons de confort on a différencié, les paramètres optionnels et les variables du fichier de construction. les paramètre existe sous une forme courte et longue :

5. Création une archive fli4l/Média de Boot

Utiliser : `mkfli4l.sh [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 boot-media
<code>--no-squeeze</code>	don't compress shell scripts
<code>-h, --help</code>	display this usage

`config-dir` sets 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

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

*** Netboot options

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

*** Developer options

<code>-u, --update-ver</code>	set version to <fli4l_version>-rev<svn revision>
<code>-v, --verbose</code>	verbose - some debug-output
<code>-k, --kernel-pkg</code>	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
<code>--filesonly</code>	create only fli4l-files - do not create a boot-media
<code>--no-squeeze</code>	don't compress shell scripts
<code>--rebuild</code>	rebuild mkfli4l and related tools; needs make, gcc

Avec l'option `--hdinstallpath <dir>` il est possible de faire une pré-installation sur une

5. Création une archive fli4l/Média de Boot

carte compact-flash en utilisant un lecteur de carte USB ou sur une clé USB, les supports doivent être formater en (FAT16/FAT32). Cette fonction est surtout utilisée *à vos propres risques* pour la création de carte compact-flash ou de clé USB. Les fichiers nécessaires pour fli4l seront copiés sur la partition spécifiée. Le script ci-dessous appelle le répertoire fli4l.

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

Les fichiers fli4l seront copiés sur la carte CF ou sur la clé USB.

Pour effectuer les prochaines étapes, les conditions suivantes doivent être remplies :

- `chmod 777 /dev/brain`
- Droits-super-utilisateur
- Installer `syslinux`
- Installer `fdisk`

Ensuite le script contrôle, si le support de données est un lecteur USB et si la première partition est une partition FAT. Puis le Bootloader et les fichiers nécessaires sont copiés sur le volume spécifié. A la fin du script, vous recevrez un message indiquant le succès ou l'échec de l'installation.

Après la construction, vous devez exécuter.

```
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
```

Pour finir, la carte CF ou la clé USB sera amorçable. Ne pas oublier de démonter le périphérique (avec `umount`).

Avec les derniers paramètres d'optionnel, On peut créer un répertoire de configuration alternatif. Le répertoire de configuration normal s'appelle `config` et se trouve directement à la racine du répertoire de fli4l. Dans ce répertoire, sont enregistrés tous les fichiers de configuration des paquetages fli4l. Si on veut gérer plus d'une configuration, on peut créer un répertoire supplémentaire, par exemple `hd.conf`, ici une copie des fichiers de configuration est faite et si vous voulez vous pouvez modifier ces fichiers selon vos besoins. Quelques exemples :

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

5.2. Création d'une archive fli4l/Média de Boot sous Windows

Le programme utilisé est 'AutoIt3' voir le site (<http://www.autoitscript.com/site/autoit/>). il permet une construction 'graphique' de fli4l et aussi des dialogues dans lesquels les variables sont décrites dans ce paragraphe, voici la commande.

```
mkfli4l.bat
```

Build-Script reconnaît indépendamment les différentes [variantes de Boot](#) (Page 23).

Le démarrage de 'mkfli4l.bat' peut s'opérer directement dans l'Explorer de Windows, sans utiliser aucun paramètre optionnel.

Les différents mécanismes gèrent la construction du programme Build :

- Configuration de la variable `BOOT_TYPE` dans le fichier `<config>/base.txt`
- Configuration du fichier `<config>/mkfli4l.txt`
- Les Paramètres du Programme Build
- Le Réglage interactif avec le GUI

On décide au moyen de la variable `BOOT_TYPE` (Page 23), le type de média de construction (Build-Scripts) pour fli4l :

- Démarrer fli4l avec un CD-ROM par une image ISO
- Faire une mise à jour des fichiers, les copier sur le média
- Faire une mise à jour des fichiers, les envoyer sur le routeur via SCP
- Pré-installer un Disque Dur ou un CF (Compact Flash) en utilisant un lecteur de carte
- etc.

Vous trouverez la description des variables dans le fichier de configuration `<config>/mkfli4l.txt` dans ce chapitre [Paramètres mkfli4l.txt](#) (Page 80).

5.2.1. Ligne de commande en option

On a la possibilité de rajouter des paramètres optionnels dans le fichier de commande `mkfli4l.txt`, qui appelle le programme de construction (Build-Programms). Ces paramètres ont les mêmes orientations que le programme 'graphique'. Pour des raisons de confort on a différencié, les paramètres optionnels et les variables du fichier de construction. Les paramètres existent sous une forme courte et une forme longue les voici :

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

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

```
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]
```

Avec les derniers paramètres optionnel, Vous pouvez créer un répertoire de configuration alternatif. Le répertoire de configuration normal s'appelle **config** il se trouve directement à la racine du répertoire fli4l. Dans ce répertoire, sont enregistrés tous les fichiers de configuration des paquetages fli4l. Si on veut gérer plusieurs configurations, on peut créer un répertoire supplémentaire, par exemple **hd.conf**, on copie dans celui-ci les fichiers de configuration du répertoire **config**, vous pouvez ensuite modifier ces fichiers selon vos besoins. Ici quelques exemples pour démarrer le Build :

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

5.2.2. Boîte de dialogue - Définition du répertoire de configuration

Il y a dans la fenêtre principale de la boîte de dialogue, une liste de paramètres de configurations pour différents réglages, on peut ouvrir la fenêtre de son choix pour paramétrer le programme.

Attention dans 'Config-Dir' on peut modifier le répertoire des fichiers de constructions dans Paramètres 'mkfli4l.txt' (Page 80) qui est stocké sur votre disque.

Si mkfli4l.bat ne trouve pas le fichier 'base.txt' dans le répertoire fli4l-x.y.z\ une fenêtre s'ouvre immédiatement pour rechercher le fichier de configuration. Cela permet d'administrer facilement une liste de plusieurs configurations pour fli4l.

Exemple :

```
fli4l-x.y.z\config
fli4l-x.y.z\config.fd
fli4l-x.y.z\config.cd
fli4l-x.y.z\config.hd
fli4l-x.y.z\config.hd-construction
```

5.2.3. Boîte de dialogue – Paramètres généraux

On définit dans cette fenêtre, la sauvegarde des paramètres et la création du média :
— Build-Dir – Répertoire pour l'archive/l'image CD/...

5. Création une archive fli4l/Média de Boot



FIGURE 5.1. – Paramètre

- BOOT_TYPE — Régle l’affichage/utilisé BOOT_TYPE — il ne peut pas être modifié ici
- Verbose — Affiche les informations pendant la construction du programme fli4l
- Fileonly — Sauvegarde uniquement les fichiers, pas de création d’image
- Remoteupdate — Active la mise à jour par SCP

Avec le bouton **les paramètres du programme fli4l-build peuvent être sauvegardés à tout moment**, les paramètres seront enregistrés dans le fichier mkfli4l.txt, ils peuvent être modifié manuellement en ouvrant ce fichier.

5.2.4. Boîte de dialogue – Paramètres pour la mise à jour à distance

- On définit dans cette fenêtre, les réglages pour l’installation d’une mise à jour :
- Adresse IP ou Nom d’Hôte



FIGURE 5.2. – Paramètre pour la mise à jour

- Nom d'utilisateur sur l'hôte distant
- Remote-path (Par défaut : /boot)
- Remote-port (Port par défaut : 22)
- Utiliser SSH-Keyfile (Format ppk de Putty)

5.2.5. Boîte de dialogue – Paramètres pour une pré-installation du HD

On définit dans cette fenêtre, les paramètres pour la pré-installation d'un disque dur, une Carte CompactFlash, une clef USB formaté et partitionné.

Options possibles :

- Activer la pré-installation du Disque Dur
- Lettre du lecteur ou de la Carte-CF



FIGURE 5.3. – Paramètre pour pré-installation du DD

Information pour partitionner et formater CF (Compact Flash) : Pour cette installation utiliser le TYPE A, de plus (nous avons besoin du paquetage HD), une partition FAT primaire doit être active et formatée sur la CF. Si l'on veut utiliser une partition bootable il faut installer une partition Linux supplémentaire formatée avec le système ext3, on aura besoin du fichier `hd.cfg` sur la partition FAT (pour cela il faut absolument installer et configurer le paquetage HD).

5.3. Paramètre pour le fichier `mkfli4l.txt`

Il existe depuis la Version fli4l 2.1.9, le fichier de configuration `<config>/mkfli4l.txt`. Toutes les commandes du programme 'graphique' fli4l-Build sont enregistrées dans le fichier `mkfli4l.txt`. Le fichier est construit comme tous les fichiers fli4l. Toutes les variables de confi-

guration sont optionnelles, mais il ne faut pas, modifier les variables spécifiques.

BUILDDIR Valeur par défaut : 'build'

On indique ici le nom du répertoire pour enregistrer les fichiers de construction pour le boot de fli4l. Si la variable n'est pas définie, mkfli4l sous Windows utilisera par défaut le sous-répertoire `build` de la racine du répertoire fli4l :

`Chemin/fli4l-x.y.z/build`

En lançant mkfli4l le programme enregistre des fichiers de construction produits dans le répertoire `<config>/build`.

Vous devez utiliser les conventions des systèmes d'exploitation de Windows ou *Unix pour paramétrer le chemin d'accès BUILDDIR. Si vous avez paramétré un chemin relatif, ce chemin sera converti par le processus de construction de Windows ou *Unix.

VERBOSE Valeur par défaut : `VERBOSE='no'`

Valeurs possibles sont 'yes' ou 'no'. Affiche les *les Informations* du processus Build (ou processus de construction).

FILESONLY Valeur par défaut : `FILESONLY='no'`

Valeurs possibles 'yes' ou 'no'. Vous permet de créer un Boot média, peut être désactivé de sorte à créer uniquement les fichiers d'archives.

REMOTEUPDATE Valeur par défaut : `REMOTEUPDATE='no'`

Valeurs possibles 'yes' ou 'no'. Si on veut transmettre automatiquement des fichiers de boot sur le Routeur au moyen de SCP. Cela suppose que le paquetage SSHD (Page ??) est installé et en plus la variable `scp` soit activée dans se paquetage.

REMOTEHOSTNAME Valeur par défaut : `REMOTEHOSTNAME=""`

On indique ici le nom d'hôte du destinataire pour le transfert des données avec SCP. Si vous n'avez indiqué aucun nom, le nom de la variable `HOSTNAME` (Page 22) est utilisée pour le transfert des données.

REMOTEUSERNAME Valeur par défaut : `REMOTEUSERNAME='fli4l'`

Nom d'utilisateur pour la transmission des données SCP.

REMOTEPATHNAME Valeur par défaut : `REMOTEPATHNAME='/boot'`

Chemin d'accès du destinataire pour la transmission des données SCP.

REMOTEPORT Valeur par défaut : `REMOTEPORT='22'`

Port du destinataire pour la transmission des données SCP.

SSHKEYFILE Valeur par défaut : `SSHKEYFILE=""`

Ici on peut indiquer le fichier de clef-SSH pour la mise à jour avec SCP. Un mot de passe peut aussi être demandé pour la mise à jour.

REMODEREMOUNT Valeur par défaut : `REMODEREMOUNT='no'`

Les valeurs possibles sont 'yes' ou 'no'. Si vous indiquez 'yes', vous remontez le boot device `"/boot"` en lecture/écriture si le boot est en lecture seule, c'est pour monter et rendre possible la mise à jour distante.

TFTPBOOTPATH Le chemin d'accès pour installer l'image de boot par le réseau.

TFTPBOOTIMAGE Nom de l'image de boot sur le réseau.

PXESUBDIR Sous-répertoire pour les fichiers PXE qui est en rapport avec TFTPBOOT-PATH.

SQUEEZE_SCRIPTS Active ou désactive Squeeze (compression des scripts). Par ex. un Script qui contient en plus des lignes de commentaires, ces lignes seront supprimées à la compression par Squeeze. Normalement on devrait toujours indiquer 'yes' dans cette variable.

MKFLI4L_DEBUG_OPTION Options supplémentaires de débogage, peut être transmis au Programme-mkfli4l (Page ??).

6. Réglage des PCs dans le LAN

Réglage des ordinateurs dans le LAN (ou réseau local) :

1. Adresse IP (voir [Adresse IP](#))
2. Nom de l'ordinateur et Nom de Domaine (voir [Nom de l'ordinateur et de Domaine](#))
3. Gateway-Standard (Passerelle Standard) (voir [Gateway](#))
4. Adresse IP et serveur-DNS (voir [Serveur-DNS](#))

6.1. Adresse IP

Les adresses IP du réseau local doivent se trouver dans le même réseau que l'adresse IP du routeur fli4l (de l'interface Ethernet), par ex. 192.168.6.2 pour l'ordinateur local dans le cas où le routeur aurait l'adresse IP 192.168.6.1. Les adresses IP doivent être uniques dans le réseau, changer uniquement le dernier chiffre de l'adresse IP est un bon moyen pour ne pas se tromper. Vous devez vous assurer que l'adresse IP indiquée ici est la même adresse IP que vous avez configurée pour cet ordinateur dans le fichier config/base.txt.

6.2. Nom de l'ordinateur et de domaine

Le nom de l'ordinateur est par ex. "mon-pc", et le nom de Domaine "lan.fli4l".

Important: *Le domaine qui est réglé dans le PC doit être identique au domaine choisi dans l'ordinateur fli4l, si on veut utiliser le routeur fli4l comme serveur DNS, il peut y avoir d'énormes problèmes dans le réseau si les domaines sont différents.*

La raison : les ordinateurs Windows cherchent régulièrement les ordinateurs avec le même nom de groupe de travail WORKGROUP.mon-domain.fli4l. Si fli4l ne répond pas à la requête du domaine (ici : mon-domain.fli4l), alors fli4l essaiera de chercher le domaine en se connectant sur Internet ...

Le domaine doit être enregistré dans les réglages TCP/IP de l'ordinateur.

6.2.1. Windows 2000

Pour Windows 2000 se trouve sous :

Démarrer ⇒

Paramètre ⇒

Panneau de configuration ⇒

Connexion réseau ⇒

Connexion au réseau local ⇒

Bouton droit propriétés ⇒

Protocole Internet (TCP/IP) ⇒

Sélectionner ⇒

Avancé ...⇒

DNS ⇒

Suffix DNS pour cette connexion ⇒

Entrer "lan.fli4l" (ou indiquer votre domaine) (sans les " ") ⇒ et appuyez sur OK.

6.2.2. NT 4.0

Démarrer ⇒

Paramètre ⇒

Panneau de configuration ⇒

Réseau ⇒

Protocole ⇒

TCP/IP ⇒

Propriétés ⇒

DNS ⇒

- Nom d'hôte entrer (le Nom de l'ordinateur)
- Domaine entrer (le même Nom que dans le fichier config/base.txt)
- Ajouter adresse IP le même réseau que le routeur fli4l
- Ajouter suffix DNS (Domaine le même que la ligne 2)

6.2.3. Windows 95/98

Démarrer ⇒

Paramètre ⇒

Panneau de configuration ⇒

Réseau ⇒

Configuration ⇒

TCP/IP (sélectionner la carte réseau qui va au routeur) ⇒

propriétés ⇒

Configuration DNS :

Cliquer activé DNS, dans le champ "Domaine" : entrer "lan.fli4l" (ou indiquer votre domaine) (sans les " ").

6.2.4. Windows XP

Pour Windows XP se trouvent sous :

Démarrer ⇒

Paramètre ⇒

Panneau de configuration ⇒

Connexions réseau ⇒

Connexion au réseau local ⇒

Propriétés ⇒

Protocole Internet (TCP/IP) ⇒

Propriétés ⇒

Avancé...⇒

DNS ⇒

Suffixe DNS pour cette connexion ⇒

Indiquez "lan.fli4l" (ou indiquer votre domaine) (sans les " ") ⇒ Cliquez sur OK.

6.2.5. Windows 7

Pour Windows 7 se trouvent sous :

Bouton Windows (ex. Démarrer) ⇒

Contrôle ⇒

Panneau de configuration ⇒

Centre Réseau et partage ⇒

Connexion au réseau local ⇒

Propriétés ⇒

Protocole Internet version 4 (TCP/IPv4) ⇒

Propriétés ⇒

Avancé... ⇒

DNS ⇒

Suffixe DNS pour cette connexion ⇒

Indiquez "lan.fli4l" (ou indiquer votre domaine) (sans les " ") ⇒ Cliquez sur OK.

6.2.6. Windows 8

Pour Windows 8 se trouvent sous :

Appuyez simultanément sur la touche Windows et X ⇒

Contrôle ⇒

Connexions réseau ⇒

Sélectionnez votre réseau (Ethernet ou WLAN) ⇒

Clique droit ⇒

Propriétés ⇒

Protocole Internet version 4 (TCP/IPv4) ⇒

Propriétés ⇒

Avancé... ⇒

DNS ⇒

Suffixe DNS pour cette connexion ⇒

Indiquez "lan.fli4l" (ou indiquer votre domaine) (sans les " ") ⇒ Cliquez sur OK.

6.3. Gateway (ou Passerelle)

Il est absolument nécessaire d'indiquer une adresse IP dans le paramètre passerelle par défaut de votre PC, car s'il n'y a pas d'adresse IP d'indiquée, rien ne fonctionnera. Ainsi vous devrez indiquer l'adresse IP du routeur fli4l - (Interface Ethernet) par exemple 192.168.6.4, selon l'adresse IP qui est configurée dans le fichier config/base.txt du routeur fli4l.

Il est incorrect de configurer le routeur fli4l comme un proxy dans Windows ou dans de votre navigateur – sauf si vous définissez un proxy sur votre routeur fli4l. Normalement fli4l a pas de proxy, s'il vous plaît ne spécifiez *pas* fli4l comme un proxy !

6.4. Serveur DNS

Pour l'adresse IP du serveur DNS, vous ne devez pas indiquer d'adresse IP de votre fournisseur d'accès Internet mais l'adresse IP du routeur (interface Ethernet), car le routeur peut répondre aux requêtes DNS et faire suivre ceux-ci par Internet si nécessaire.

Quand fli4l est utilisé comme serveur DNS, beaucoup de requêtes DNS sont envoyées par les PCs client Windows, c'est le routeur fli4l qui leur répond directement, elles ne sont pas expédiées sur Internet.

6.5. Divers points

Les points 1 et 4 n'ont pas besoin d'être enregistrés avec un serveur DHCP puisque le routeur fli4l communique les données nécessaires automatiquement.

Dans Options Internet : et dans la fenêtre connexion vous ne devez "sélectionner aucun lien". Dans Paramètre réseau local (LAN) : ne RIEN indiquer (sauf si vous utilisez le paquetage OPT_Proxy). Par défaut les deux paramètres n'ont pas besoin d'être modifié pour une utilisation normale.

7. Interface client/serveur imon

7.1. Server imon avec imond

Imond est un programme serveur qui répond à certaines enquêtes sur la gestion du réseau et accepte aussi des commandes qui peuvent contrôler le routeur sur le réseau local.

Imond contrôle également les Moindres-Coûts-Routages. Il utilise le fichier de configuration /etc/imond.conf qui est produit automatiquement au moment du boot, à partir de la variable ISDN_CIRC_x_XXX du fichier config/isdn.txt, le fichier est généré par un script shell.

imond est un démon qui fonctionne en permanence en tâche de fond, il écoute le port 5000 TCP/IP sur le périphérique /dev/isdninfo.

Voici toutes les commandes qui peuvent être envoyées par le port 5000 TCP/IP :

Le port 5000 TCP/IP est accessible uniquement depuis un réseau LAN masqué. Avec la configuration standard du firewall l'accès est bloqué de l'extérieur.

Imond supporte deux modes d'administrations, le Mode Utilisateur et le Mode Admin. On peut installer un Mot de Passe pour ces deux modes au moyen des variables IMOND_PASS et IMOND_ADMIN_PASS. Si le Mot de Passe n'est pas transmis au serveur imond le client imonc a accès uniquement à deux commandes "pass" et "quit" toutes les autres commandes sont rejetées et une erreur s'affiche.

Si plus tard, vous voulez limiter l'accès au serveur imond à un seul PC, la configuration du Firewall doit être modifier.

Les commandes

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

peuvent être activées ou désactivées dans la variable IMOND_XXX voir (le chapitre "configuration").

Avec un ordinateur Unix/Linux (ou un ordinateur Windows par la fenêtre DOS) vous pouvez facilement entrer les commandes après la connexion telnet.

Connexion telnet :

```
telnet fli4l 5000          \# ou le Nom correspondant au routeur fli4l
```

Vous pouvez directement entrer les commandes mentionnées ci-dessus.

Par exemple la commande "help" active l'aide sur l'écran ou "quit" démonte (ou arrête) le serveur imond.

7.1.1. Mode de fonctionnement du Moindre-Coût-Routage

imond construit une Time-Table (ou Plage Horaire) à partir du fichier de configuration /etc/imond.conf (qui est créé au boot avec la variable de configuration ISDN_CIRC_x_TIMES. Ce "calendrier" est composé d'une semaine par intervalle d'une heure, une semaine = 168 heures

Commandes Admin

addlink ci-index	Ajouter un canal au circuit (Channel-Bundling)
adjust-time seconds	Incrémente la date sur le routeur en secondes
delete filename pw	Supprime le fichier sur le routeur
hup-timeout #ci-index [value]	Affiche ou compose le HUP-Timeout pour des circuits RNIS (ou numéris)
removelink ci-index	Enlever le canal supplémentaire
reset-telmond-log-file	Supprime le fichier journal de telmond
reset-imond-log-file	Supprime le fichier journal de imond
receive filename #octets pw	Transfère d'un fichier au routeur. Imond donne l'ordre avec ACK (0x06). Après, le fichier est transféré par blocs de 1024 Octets qui sont également confirmé avec ACK. En conclusion, imond répond OK.
send filename pw	Si le mot de passe est correct et que le fichier existe, imond répond OK avec un #octet. Puis, imond transfère le fichier par blocs de 1024 octets, chaque fois confirmés avec ACK (0x06). A la fin, imond répond OK.
support pw	Montre le statut/configuration du routeur
sync	Synchronise le Cache des lecteurs montés

Commandes Admin et Utilisateur

dial	Choix du FAI (Default-Route-Circuit)
dialmode [auto manual off]	Réglage des actions dans Dialmode
disable	Raccroche et place dialmode sur "off"
enable	Mets dialmode sur "auto"
halt	Descend proprement le Routeur
hangup [#channel-id]	Raccroche
poweroff	Descent le routeur et mise hors tension
reboot	Reboot le routeur fli4l!
route [ci-index]	Met le routeur par Defaut sur un Circuit X (0=automatique)

Commandes Utilisateur

channels	Nombre de Canaux ISDN disponibles
charge #channel-id	Edite les frais de connexion pour un Canal en ligne
chargetime #channel-id	Temps et frais de connexion pour un canal en ligne
circuit [ci-index]	Edite le numéro du Circuit
circuits	Edite le nombre de Default-Route-Circuits
cpu	Donne la charge du CPU en pourcentage
date	Edite la date et heure
device ci-index	Circuits du périphérique utilisé
driverid #channel-id	Edite Driver-ID pour le Canal X
help	Edite l'aide
inout #channel-id	Edite la direction (entrante/sortante)
imond-log-file	Edite le fichier du Protocole imond
ip #channel-id	Edite l'adresse IP
is-allowed command	Edite si la commande est valide
	commandes possibles : dial dialmode route reboot imond-log telmond-log mgetty-log
is-enabled	Edite si dialmode est sur off (0) ou auto (1)
links ci-index	Edite le nombre de canaux 0, 1 ou 2, 0 utilisé, ou alors : Aucun Channel-Bundling possible
log-dir imond telmond mgetty	Donne la direction des fichiers Log
mgetty-log-file	Edite le protocole du fichier mgetty
online-time #channel-id	Edite le temps en ligne, et de connexion en hh :mm :ss
pass [password]	Vérifie, le mot de passe qui a été saisi par 1 Mot de passe Utilisateur est fixé 2 Mot de passe Admin est fixé 4 imond se trouve dans le mode Admin
phone #channel-id	Edite le numéro de Tél et le nom du "correspondant"
pppoe	Donne le numéro du périphérique pppoe (0 ou 1)
quantity #channel-id	Donne l'ensemble des transmissions (en octet)
quit	Coupe la connexion avec imond
rate #channel-id	Edite les connexions (entrant/sortant en Octet/sec)
status #channel-id	Edite le statut pour le Canal X
telmond-log-file	Edite le protocole telmond
time #channel-id	Edite le temps total en ligne, au Format hh :mm :ss
timetable [ci-index]	Edite la time-table LC-Routing
uptime	Edite le temps d'utilisation du Routeur en secondes
usage #channel-id	Edite les réponses des connexions : Fax, Répondeur, Net, Modem, Raw
version	Edite la version du protocole et la version du Pro- gramme

7. Interface client/serveur imon

= 168 octets. La table se compose de circuits, dans lequel sont définis des Défaut-Routes (ou connexion par défaut au FAI).

Avec la commandement "timetable" on peut voir la table imond. Exemple de configuration :
Supposons que nous définissions 3 circuits de connexions pour chaque FAI c'est à dire :

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

Les deux premiers circuits sont réglés avec Défaut-Route c.à d. que l'itinéraire par défaut est écrit dans la variable ISDN_CIRC_x_ROUTE avec la valeur '0.0.0.0/0'.

Les variables ISDN_CIRC_x_TIMES se présentent de la manière suivante :

```
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'
```

Puis le fichier /etc/imond.conf est créé de cette façon :

#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 produit alors Time-Table (ou Plage Horaire) dans la mémoire. voici la table des données sorties avec la commande "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

7. Interface client/serveur imon

3	Addcom	yes	ipp0	0.0248	60
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

Pour le circuit 1 (Addcom) il y a trois éléments définis (1-3), pour le circuit 2 il y a quatre éléments (4-7), et pour le circuit 3 il y a trois éléments (8-10).

Les index des circuits activés sont inscrits toutes les heures dans la Time-Table respectivement. Ici les index (2-4) apparaissent, car les autres ne passent pas par LC-Défaut-Route.

Si vous avez des zéros dans Time-Table, c'est qu'il manque des données dans la variable ISDN_CIRC_X_TIMES. Si vous avez des zéro sur certaine plage horaire, cela veut dire qu'il n'y aura pas de Défaut-Route et aucun accès Internet possible sur ces plages horaires !

Au démarrage du programme, imond vérifie le jour de la semaine et l'heure, puis les index dans la Time-Table et enfin règle les Défauts-Routes correspondants. Le Défaut-Route (ou connexion par Défaut au FAI) est alors activé par rapport à l'indexation.

Lors d'un changement de statut, par exemple sur un canal, une connexion ou un rattachement de la ligne, si la commande mais plus d'une minute, le processus de démarrage est réactualisé, vérification de l'horaire et du jour, consultation de la table, sélection du Circuit-Défaut-route.

Si par exemple le lundi à 18 :00 la connexion change, Défaut-Route est supprimé, les connexions existantes sont arrêtées (désolé...), ensuite imond contrôle dans la Time-Table si un nouveau Circuit-Défaut-route existe, si oui imond mettra environ 60 secondes pour se reconnecter. Donc la connexion se fera au plus tard à 18 :00 :59.

Il n'y aura aucun changement pour les circuits qui n'utilisent pas un Défaut-Route. Le contenu ISDN_CIRC_x_TIMES sera uniquement employé pour le calcul des frais téléphoniques. Ceci peut être pertinent, si vous arrêtez temporairement le client imonc et que vous choisissiez manuellement un Circuit-Défaut-route.

Vous pouvez également regarder dans l'indexation de Time-Table (exemple précédent de 1 à 10) les circuits non activés "Non-LC-Default-Route-Circuits".

Commande pour vérifier un index dans le Time-Table :

```
timetable "index"
```

Exemple :

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

La sortie des données apparaîtront comme ceci :

	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

7. Interface client/serveur imon

We	5	5	5	5	5	5	5	5	5	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	5	5	5	5	5	5
Fr	5	5	5	5	5	5	5	5	5	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

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

Tout est clair jusque là ?

Avec la commande "Route" d'imond vous pouvez commuter "Marche/Arrêt" de LC-Routing, et vous pouvez indiquer l'index du Circuit-Défaut-Route (1..N), il se connectera sur le circuit. Si l'index est 0, le LC-Routing est activé et le circuit sera choisi automatiquement.

7.1.2. Calcul des frais on-line (en ligne)

Le mode de calcul des frais de connexions fonctionnera correctement uniquement si l'unité téléphonique est constante tout au long de la semaine, elle doit être inscrite dans la variable ISDN_CIRC_x_CHARGEINT) en seconde. Normalement c'est la règle pour les fournisseurs d'accès Internets. Toutefois, si vous choisissez Telekom (je ne parle pas de T-Online !) par exemple, pour un réseau d'entreprise, qui sera considéré comme des conversations téléphoniques normales. et changement passe de 90 secondes à 4 minutes après 18 :00 (Stand Juni 00). Par conséquent, la définition

Mais si vous utilisez votre société téléphonique par exemple pour un accès Internet avec TéléKom (Allemagne) l'unité Tél change (information juin 2000).

En France l'unité Tél est toujours constante 60 secondes, on n'a pas ce problème, c'est juste le tarif qui change en heure creuse 0,018 euro et en heure pleine 0,033 euro (8 :00 à 19 :00 heure pleine).

```
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'
```

est en fait pas tout à fait exact. Le tarif le soir est de 3 cents la minute (donc 12 cents les 4 minutes de télécommunication), mais la mesure est fautive. C'est pour cette raison qu'il se produit des différences d'affichage par rapport au prix réel.

Il est possible que ce problème soit peut être corrigé plus tard. En attendant on peut définir dans la variable ISDN_CIRC_x_CHARGEINT) 2 Circuits : un pour la journée avec ISDN_CIRC_1_CHARGEINT='90' et l'autre pour la soirée ISDN_CIRC_2_CHARGEINT='240' naturellement vous devez configurer ISDN_CIRC_x_TIMES, avec cette configuration vous utiliser le Circuit 1 pendant la journée et le Circuit 2 en soirée.

Comme nous l'avons dit plus haut : l'utilisation des connexions avec un fournisseur d'accès Internet, ne pose pas de problème parce que l'unité Tél est toujours constante et le coût par minute ne change pas (il a encore quelque chose ? je ne fais pas confiance à T-* pour tout :-).

7.2. Client Windows imonc.exe

7.2.1. Introduction

Le démon Imond sur le routeur fli4l gère deux modes d'utilisations différents : le mode Administrateur (Admin) et le mode Utilisateur. Dans le mode Admin toutes les commandes sont activées automatiquement. Dans le mode Utilisateur vous devez activer les variables `IMOND_ENABLE`

(Page 66), `IMOND_DIAL` (Page 66), `IMOND_ROUTE` (Page 66) et `IMOND_REBOOT` (Page 66), dans le fichier `/config/base.txt` pour avoir les commandes. Si les variables sont sur 'no' les commandes ne seront pas activées, même les commandes Exit et mode Admin ne seront pas activées dans le client imonc. Le choix de l'utilisation entre le mode Utilisateur et le mode Admin se fait par l'intermédiaire d'un Mot de Passe qui sera transféré au routeur. Vous pouvez activer le mode Admin ou Utilisateur, en cliquant sur l'icône située dans la barre de tâches et entrer le Mot de Passe n'oubliez pas de redémarrer le client imonc.

Lorsque imonc a démarré, une icône supplémentaire apparaît dans la barre de tâches, il indique le statut des canaux de la connexion Internet pour le (numéris).

Les couleurs de l'icône signifient :

Rouge : offline (déconnecté)

Jaune : en cours de connexion

Vert clair : online (en ligne il y a du trafic sur le canal)

Vert foncé : online (en ligne il n'y a pas de trafic sur le canal)

Suivant le Windows que vous utilisez le comportement d'imonc diverge, il peut être réduit à une icône dans la barre des tâches près de l'heure. pour ouvrir la fenêtre il suffit de faire un double clic avec le bouton gauche de la souris sur l'icône. Pour ouvrir le menu contextuel vous utilisez le bouton droit, delà vous pouvez choisir directement les commandes imonc.

Un (grand nombre de paramètres) peuvent être adaptés selon vos propres besoins, ils seront enregistrés et sauvegardés dans la base de registre de Windows à cet endroit `HKCU\Software\fli4l`.

Il y a toujours quelques erreurs dans la documentation d'imonc et du routeur fli4l, malgré des relectures. Si vous rencontrez des problèmes, allez dans la page "A propos" cliquer sur le bouton systeminfo puis sur le bouton support info, ensuite le mot de passe du routeur vous sera demandé (pas le mot de passe d'imond!). Imonc produira un fichier `fli4lsup.txt`, qui inclura toutes les informations importantes sur le routeur fli4l et sur imonc. Ce fichier peut être ajouté dans le Newsgroup pour demander de l'aide. Cela maximisera les chances d'avoir de l'aide plus rapidement.

Vous pouvez trouver des détails concernant le développement du client imonc pour Windows sur le site <http://www.imonc.de/>, vous trouverez des informations sur les nouveaux dispositifs les futures versions d'imonc, les résolutions de bug et aussi la dernière version à télécharger (si elle n'est pas déjà incluse dans la distribution fli4l).

7.2.2. Paramètre de démarrage

Le client imonc a besoin du Nom ou de Adresse IP du routeur fli4l pour pouvoir établir une connexion avec celui-ci "l'ordinateur fli4l". Si l'ordinateur du client imonc est enregistré correctement dans le DNS, il devrait fonctionner sans problème. Voici les paramètres que l'on peut transmettre :

- `/Server :IP` ou Nom d'Hôte du routeur (Forme abrégée : `/S :IP` ou Nom d'Hôte)
- `/Password :Mot de Passe` (Forme abrégée : `/P :Mot de Passe`)
- `/log` Active le protocole de communication entre imonc et imond, lorsque cette option est activée un fichier `imonc.log` est créé. Ce fichier enregistre toutes les communications, il peut être très volumineux. C'est pour cette raison que l'on active ce paramètre uniquement si il y a des problèmes de configurations.
- `/iport :N° port` Par défaut imond écoute sur le Port : 5000

- /tport :N° port Par défaut telmond écoute sur le Port : 5001
- /rc : "Commande" Les commandes écrites ici sont transmis au routeur sans aucun contrôle supplémentaire. Si plusieurs commandes sont exportées simultanément elles doivent être séparées par un point virgule. Pour être sûr du fonctionnement de imonc vous devez retaper le Mot de Passe (si configuré ?) car il n'y aura aucune redemande de Mot de Passe. les commandes possibles sont documentées dans le Chapitre 8.1. La commande dialtimesync n'ai plus utilisée elle est remplacée par «dial ; timesync», qui force le routeur à synchroniser l'heure avec le client.
- /d : "Répertoire-fli4l" Cette option permet d'écrire le répertoire du dossier fli4l avec des paramètres de démarrage, c'est intéressant pour ceux qui utilisent plusieurs versions de fli4l.
- /wait Si le Nom d'Hôte ne peut pas être résolu, imonc se bloque, il faut redémarrer imonc par un double clic sur l'icône de celui-ci.
- /nostartcheck Cela coupe le contrôle d'imonc, s'il est en fonction. c'est uniquement nécessaire si vous avez plusieurs routeurs fli4l différents à surveiller dans votre Réseau. Si des fonctions supplémentaires étaient connectées comme syslog ou e-mail ils resteront désactivées.

Utilisation (enregistrement de lien) :

```
X:\...imonc.exe [/Server:Nom d'Hôte] [/Password:Mot de passe] [/iport:Numéro port]
               [/log] [/tport:Numéro port] [/rc:"Commande"]
```

Exemple d'enregistrement avec une adresse-IP :

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

Ou avec le nom et le Mot de Passe :

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

Ou avec le nom, le Mot de Passe et une commande au routeur :

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

7.2.3. Concernant l'aperçu de imonc

Imonc client Windows interroge imond pour avoir les informations sur les connexions Internet existantes, il les affiche dans un tableau. Sur cette page il y a aussi le statut général du routeur, l'heure, la date, le bouton synchronisation, etc. Voici les descriptions de ces fenêtres :

Statut	Calling/Online/Offline (appel/en ligne/raccrocher)
Nom	Le numéro de Tél ou le Nom du circuit
Direction	On voit si c'est une connexion entrante ou sortante
IP	Adresse IP qui à été assignée
I/Octets	Octets Entrants
O/Octets	Octets Sortants
T/enligne	Temps en ligne
T/Total	Temps total en ligne
Prix/Unit	Prix de l'unité par connexion
Prix	Prix total de la connexion

Les données seront actualisées toutes les deux secondes. (Maintenant) cette intervalle peut être changé. Dans le menu on est en mesure de voir le canal sur lequel le routeur est en ligne en temps réel. Copiez l'Adresse IP réelle dans le presse-papier et installez le canal indiqué explicitement. Ceci peut être intéressant s'il y a plusieurs connexions différents par ex. une pour naviguer sur Internet et l'autre connectée à votre entreprise, de cette façon vous pouvez débrancher l'une ou l'autre connexion.

En plus si vous avez activé telmond sur votre routeur fli4l, imonc sera en mesure d'afficher les informations sur les appels téléphoniques entrants (le nom et le numéro de Tél du correspondant). Le dernier appel téléphonique reçu sera vu au-dessus des boutons de commande. Un protocole des appels téléphoniques entrants peut être vu en utilisant les pages d'appels.

Les six boutons mentionnés ci-dessous vous permettront de choisir les commandes suivantes :

Bouton	Description	Fonction
1	Connecter/Raccrocher	Connecter ou raccrocher la ligne
2	Ajout Canal/Supp Canal	Ajoute ou supprime un canal, cette caractéristique n'est disponible que dans le Mode Admin
3	Redémarrer	Redémarre fli4l!
4	Éteindre	Arrête fli4l proprement et met le routeur hors tension
5	Arrêter	Arrête fli4l proprement, pour éteindre le routeur en toute sécurité
6	Sortir	Sort du programme client imonc

Les cinq premières commandes en mode Utilisateur peuvent être activées ou désactivées dans le fichier de configuration /config/base.txt pour le routeur fli4l. En mode administrateur toutes les commandes sont toujours activées. Le choix de la commande Dialmode modifie le comportement du routeur :

Auto	Le routeur établira automatiquement une connexion Internet s'il y a une demande dans réseau local.
Manuel	L'utilisateur doit établir la connexion manuellement.
Couper	Il n'y a aucune connexion possible, ni manuellement ni automatiquement. La sélection du bouton de "connexion" est désactivée.

La volonté de fli4l par défaut c'est d'établir automatiquement une connexion Internet sur une demande de requête Internet par n'importe quel Hôte du réseau local. En principe on ne doit jamais modifier la commande pour se connecter ...

Il y a également la possibilité de changer manuellement le Circuit-Défaut-Route, c.à d. commuter marche/arrêt ou automatique, c'est pourquoi la liste de sélection de "Default route" (choix du FAI) est prévu dans la version de Windows d'imonc. En outre, on peut maintenant configurer directement dans imonc l'heure de déconnexion. Utiliser le Bouton "config" en dessous de Défaut-Route ici la configuration de tous les circuits pour le routeur sont indiqués. La valeur de la variable Hup-timeout peut être éditée directement dans le fichier isdn.txt du Circuit ISDN (ne fonctionne pas pour le moment avec la DSL).

Un aperçu de LCR-Routing se trouve sur la page Admin/Plage Horaire. Là, vous pouvez voir, le Circuit qui sera démarré automatiquement.

7.2.4. Paramètres de configuration

On peut accéder à la configuration par le bouton "config" dans la barre d'état. La fenêtre qui s'ouvre est divisée en deux, dans le tableau de gauche vous avez les répertoires et sous-

répertoires, dans celui de droite la configuration de imonc. Voici les répertoires en détail :

- Répertoire général :
 - Synchroniser tous les : on ajuste ici le nombre de rafraîchissement en seconde de la page d'accueil.
 - Synchroniser au démarrage : synchronise l'heure et la date du routeur avec le client au démarrage. on peut activer cette fonction manuellement avec le bouton "Synchroniser" sur la page d'accueil.
 - Réduire au démarrage : au démarrage le programme sera réduit en icône. Vous verrez seulement l'icône à côté de l'heure.
 - Lancer imonc au démarrage Windows : ici le client imonc démarre automatiquement après le démarrage de Windows. on peut entrer dans la fenêtre "paramètre" des commandes supplémentaires.
 - Voir l'actualité de fli4l.de : ici on peut recevoir les (News) du site fli4l.de chargées automatiquement par imonc, les titres sont alors montrés dans la fenêtre "Nouvelles" et qui pourront être lus.
 - Appel du fichier log : on indique ici le nom du fichier pour enregistrer la liste des appels locaux.
 - Attendre la réponse du routeur : temps d'attente d'une réponse du routeur en seconde, avant que la connexion soit perdue.
 - Langue : on choisit ici la langue pour imonc.
 - Confirmer les commandes du routeur : si la case est cochée, toutes les commandes envoyées au routeur demande une confirmation, ex. redémarrage, déconnexion etc ...
 - Arrêt même avec trafic : si aucune réponse n'aboutit, la connexion s'arrête même si il y a toujours du trafic sur cette connexion.
 - Reconnexion automatique au routeur : une reconnexion du routeur est faite automatiquement, si une coupure de la connexion a eu lieu, (p. ex. un redémarrage du routeur).
 - Reduire la fenêtre système : si activée, en cliquant sur le bouton "Sortir" imonc se réduit en icône vers la barre des taches à côté de l'heure, au lieu de s'arrêter.
- Sous-répertoire proxy : ici on enregistre le proxy pour les demandes http. Celui-ci est utilisé à présent pour l'actualisation des fenêtres, time-table et news.
 - Active le proxy pour le protocole http : ici on active Proxy
 - Adresse : ici l'adresse du serveur proxy
 - Port : ici le numéro de port du serveur proxy (default : 8080)
- Sous-répertoire icône : ici on peut personnaliser les couleurs des icônes. Dans l'avenir on pourra choisir les couleurs de fond de l'icône pour dialmode (mode de connexion) qui sera placé dans la barre de tache.
- Répertoire d'appel, le réglage de la position de la fenêtre avis d'appel sur l'écran, sera stockée et sauvegardée dans la base de Registre. Vous pouvez déplacer la fenêtre à l'endroit de votre choix. Après ce réglage, la fenêtre apparaîtra exactement cet endroit à chaque fois.
- Mise à jour : on peut choisir ici, comment imonc reçoit les informations des nouveaux appels tél, Il y a trois possibilités différentes. La premier consiste à interroger régulièrement de service telmond sur le routeur. Une autre possibilité consiste à interroger les annonces de Syslog, cette variante est la préférer – On doit Naturellement activer Syslog dans le client imonc. Imonc doit être connecté à une direction approprié, la troisième possibilité proposé est d'utiliser le paquetage Capi2Text pour la signalisa-

tion d'appel.

- Effacer premier zéro : parfois devant le numéro de téléphonique est placé un zéro supplémentaire. Celui-ci peut être supprimé avec cette option.
- Indicatif régional : la présélection personnelle du numéro de tél peut être écrite ici. Quand un appel arrive avec la même présélection. La présélection ne sera pas visible.
- Annuaire : ici on indique le fichier dans lequel l'annuaire téléphonique local sera sauvegardé pour les numéros de téléphones. Si le fichier n'existe pas, il est automatiquement installé.
- Fichier log : on indique ici le nom du fichier, utile pour enregistrer la liste des appels sur l'ordinateur local. Ce paramètre est visible uniquement si la variable TELMOND_LOG être sur 'yes', c'est également valable pour la liste des appels réelle.
- Recherche externe : un programme peut être indiqué dans cette fenêtre, que l'on appelle si un numéro de téléphone ne peut pas être résolu au moyen de l'annuaire téléphonique local. Des infos plus précises devraient être jointes aux programmes correspondants. Il y a jusqu'à présent un CD d'annuaire téléphonique de Marcel Wappler KlickTel ainsi qu'un lien vers une base de données.
- Sous-Répertoire des appels tél : ces options sont destinées, à détailler des instructions des appels téléphoniques et de les afficher, voir les illustrations ci-dessous.
 - Notification d'appel actif : détermine si des appels doivent être signalés.
 - Indication des notifications d'appels : lors d'un appels tél une fenêtre d' apparait, elle détaille les Infos suivantes : l'appel MSN, le numéro de tél du correspondant et la date/heure de l'appel. Pour cela il est nécessaire que la variable OPT_TELMOND soit placée sur 'yes' dans le fichier config/isdn.txt
 - Ne pas enregistrer les numéros non transmis : les appels ne doivent pas être écrit dans la fenêtre d'appel, si aucun numéro de Tél n'a été transféré.
 - Temps d'affichage : cette indication influe sur la durée de fermeture de la fenêtre avis d'appel, la fenêtre doit rester ouverte un certain temps. Si on indique "0" la fenêtre ne se fermera pas automatiquement.
 - Fontsize (ou police) : ici on choisit la taille des caractères pour la fenêtre. Celle-ci affecte la taille de la fenêtre, puisque la taille de la fenêtre sera calculée par rapport à la taille du message.
 - Couleur : ici on choisit la couleur des textes dans la fenêtre d'appel. J'emploie le rouge pour l'identification des messages.
- Sous-répertoire annuaire : la fenêtre contient l'annuaire téléphonique qui est utilisé pour la définition des numéros de téléphones des appels entrants et aussi si vous possédez un MSN. Cette fenêtre apparait même si la variable TELMOND_LOG est sur 'no' parce que cette fenêtre est utilisée aussi pour le dernier appel entrant vu dans la fenêtre principale. On peut choisir un fichier qui sera placé sur le routeur.

Construction d'un appel entrant :

```
# Format:
# Telefonnummer=anzuzeigender Name[, Wavefilename]
# 0241123456789=Testuser
00=unbekannt
508402=Fax
0241606*=Elsa AG Aachen
```

Les trois premières lignes sont des commentaires. La quatrième ligne est créée si aucun numéro n'est transmis, "unbekannt" (ou inconnu) sera affiché. La cinquième ligne indique

le numéro de tél "508402" et le Nom "Fax" , dans tous les cas le format sera toujours le même, Numéro de Tél=Nom. La sixième ligne détermine l'ensemble des numéros de Tél, pour toutes appel ex. 0241606 le Nom sera affiché. Souvenez-vous que le dernier numéro d'appel du correspondant est indiqué sur la première fenêtre principale. Optionnel, un fichier son peut être défini et sera joué lors d'un appel Tél.

Dés la Version 1.5.2, il est possible d'installer un annuaire Téléphonique sur le routeur sous la forme d'un fichier il sera enregistré et synchronisé dans (/etc/phonebook). Si un même numéro de téléphone avec un Nom différent sont enregistrés dans l'annuaire Du routeur et dans l'annuaire de imonc, il sera demandé à l'utilisateur qu'elle est l'entrée valide. les appels ne sont pas juste recopiés mais sont enregistrés sur les deux annuaires. La synchronisation du fichier d'annuaire est faite dans la mémoire RAM, cela veut dire, lorsque l'on reboot (redémarre) le routeur, le fichier sera perdu.

- Répertoire son, les fichiers son qui seront installés ici seront joués, si l'événement indiqué se produit.
 - Courriel : le fichier son sera joué, si un nouveau courriel se trouve sur votre Serveur POP3.
 - Erreur courriel : le fichier son sera joué, si une erreur se produit lors de la réception du Courriel.
 - Connexion perdu : le fichier son sera joué, si la connexion avec le routeur est perdue (ex. redémarrage du routeur). Si l'option "reconnexion automatique au routeur" n'est pas activée, un messagebox s'ouvrira pour demander une nouvelle connexion au routeur.
 - Connexion : le fichier son sera joué, si le routeur établit une connexion Internet.
 - Déconnexion : le fichier son sera joué, lorsque le routeur désactive la connexion Internet.
 - Avis appel : le fichier son sera joué, si l'annonce des appels est activée et si un nouvel appel est reçu.
 - Annonce de fax : le fichier son sera joué, après la réception de nouveaux fax.
- Répertoire courriel
 - Comptes : cette fenêtre sert à configurer les comptes POP3.
 - Activer le contrôle courriel : si vous avez un compte courriel il recherchera automatiquement les nouveaux courriels.
 - Vérifier x/Min : cette option définit un intervalle temps entre chaque contrôle Courriel sur le compte. Attention : en définissant un intervalle trop court, le routeur peut rester constamment en ligne ! Ceci se produit lorsque l'intervalle est plus court que "Delai attente" du circuit utilisé.
 - Temps d'attente x/Sec : temps d'attente d'une réponse du Serveur POP3 avant l'arrêt de celui-ci, si la valeur est à "0" cela signifie qu'aucun TimeOut (temps d'attente) n'est installé.
 - routeur déconnecté : cette option permet au routeur de se connecter automatiquement pour rechercher les nouveaux courriels sur le Serveur POP3. Après le téléchargement des courriels le routeur se déconnecte. pour pouvoir utiliser ce dispositif on doit mettre Dialmode sur 'auto'. Attention : cela occasionne des frais supplémentaires de connexion si aucun tarif unitaire est utilisé !
 - Circuit à utiliser : cette option définit le circuit qui sera utilisé pour la connexion aux courriels.
 - Rester en ligne après contrôle : la déconnexion doit être faire manuellement ou l'arrêt doit être réalisé automatiquement par l'option Délai attente.

- Charger en-têtes des courriels : télécharger les en-têtes des courriels ou uniquement le nombre de courriel disponible ? Cette option doit être activée pour supprimer les courriels directement sur le serveur POP3.
- M'avertir de nouveaux courriels : faut-il un message sonore et une icône dans la barre de tâche pour m'annoncer de nouveaux courriels.
- Exécuter le programme de messagerie : démarrer automatiquement le programme de messagerie pour lire les nouveaux courriels disponibles.
- Programme : indiquer ici le programme de messagerie.
- Paramètre : entrer les paramètres additionnels qui seront transférés au démarrage du programme de messagerie. Si Outlook est utilisé comme programme courriel (pas Outlook Express !) vous pouvez entrer comme paramètre "/recycle" empêche de lancer Outlook dans une nouvelle fenêtre s'il est déjà ouvert.
- Répertoire Admin
 - Mot de passe Root : ici on entre le mot de passe du routeur qui est dans le fichier (/config/base.txt dans la variable PASSWORD) pour pouvoir par exemple configurer Portforwarding sur votre ordinateur et l'envoyer sur le routeur.
 - Voir les fichiers sur le routeur : tous les fichiers log (ou journal) qui se trouvent sur le routeur sont à indiquer ici, ils peuvent être lus, avec un simple clic de la souris dans la page Admin/fichier, ainsi on peut afficher les fichiers log du routeur directement dans imonc.
 - Fichier de configuration : ici on peut choisir, si tous les fichiers seront ouverts avec le programme éditeur de texte ou uniquement les fichiers *.txt pour étudier et travailler dessus. On peut également ouvrir un ensemble de fichiers.
 - DynEisfairLog : si vous avez créé un compte sur DynEisfair, vous pouvez enregistrer ici les données d'accès et de voir avec le fichier Log les mises à jours des fonctions sur la page Admin/DynEisfairLog.
- Répertoire démarrage auto, sert à configurer une liste de programmes qui sera lancée automatiquement. Celle-ci est exportée après une connexion réussie si l'option "Activer la liste des programmes" est cochée.
 - Programme : tous les programmes installés ici seront lancés automatiquement, si le routeur est connecté et que La Liste des programmes est cochée.
 - Activer la liste des programmes : la liste doit-elle être activé pour exécution des programmes après une connexion réussie ?
- Répertoire trafic du réseau, est utilisé pour la configuration (personnalisée) de la fenêtre de Info trafic. Un utilisateur m'a averti qu'il y avait quelques erreurs sur la définition des données avec des versions anciennes de DirectX.
 - Voir les informations sur le trafic : voulez-vous afficher une utilisation graphique des canaux dans une fenêtre à part ? Dans le menu contextuel vous pouvez choisir l'attribut StayOnTop, cette option provoque l'affichage de la fenêtre sur toutes les autres fenêtres. Cette option sera enregistrée dans la base de registre et sera en service après un redémarrage du programme.
 - Voir les titres : doit-on monter la barre de titre dans la fenêtre Traffic-Info ? Cette fenêtre montrera les informations des circuits utilisés par le routeur.
 - Voir l'utilisation CPU : montrer l'utilisation du CPU dans la barre de titre ?
 - Voir le temps de communication : le temps en ligne du canal doit-il aussi être indiqué dans la barre de titre ?
 - Fenêtre semi-transparente : la fenêtre doit-elle être représentée en transparence ? Cette

- fonction n'est disponible que sous Windows 2000 et Windows XP.
- Couleur : les couleurs sont définies ici pour la fenêtre Traffic-Info. Maintenant le canal DSL et le premier canal ISND utiliseront les mêmes couleurs.
 - Limite : entrer les valeurs maximales des taux de transmission xDSL – pour T-Online : Débit Montant (upload) 128 Ko/s et Débit Descendant (download) 1024 Ko/s.
 - Répertoire Syslog, est utilisé pour la configuration de l'affichage des messages Syslog.
 - Activer le client Syslog : montrez les messages Syslog dans imonc ? Cette option doit être arrêtée, si vous utilisez un autre client Syslog externe, par exemple le client Kiwi's Syslog.
 - Indiquer les messages Syslog : monter les messages Syslog avec un niveau de prioritaire ? Vous pouvez indiquer ici les niveaux prioritaires des messages Syslog, par défaut le message debug est coché, vous pouvez cocher le niveau selon vos besoins.
 - Enregistrer les messages Syslog : les messages lus doivent-ils être sauvegardés ? Dans la fenêtre on peut choisir les messages que l'on veut sauvegarder. On peut insérer des caractères supplémentaires avec nom de fichier à sauvegarder, les voici :
 - %y** – On l'ajoute pour avoir l'année actuel
 - %m** – On l'ajoute pour avoir le mois actuel
 - %d** – On l'ajoute pour avoir le jour actuel
 - Voir le nom des ports : doit on afficher la description du port au lieu du numéro de port ?
 - Voir les messages pare-feu : ici, on indique les messages du firewall (ou pare-feu), il seront aussi indiqués en mode utilisateur.
 - Répertoire fax, sert à configurer les fax (ou télécopie) dans imonc. Pour que ce dossier soit visible vous devez installer sur le routeur le paquetage mgetty et/ou faxrcv, (vous pouvez les trouver sur le site de fli4l).
 - Fichier Log pour fax : ici on peut enregistrer les fax reçus sous forme de fichier dans un dossier de l'ordinateur.
 - Répertoire local des fax : configurer le répertoire pour stocker les fax reçus, avant de les consulter.
 - Actualisation : il y a deux possibilités, lorsque imonc reçoit un nouveau fax. Soit c'est imonc Syslog qui reçoit les fax (naturellement le client imonc-Syslog doit être activé), soit imonc regarde régulièrement le fichier log. La première variante est la meilleure. Si vous utilisez la deuxième variante, vous pouvez indiquer combien de fois la page d'aperçu de fax doit être actualisée. Il faut faire attention cette valeur n'est pas une indication en seconde, mais c'est une indication en multiple, en général c'est une intervalle d'actualisation.
 - Répertoire tableau, sert à ajuster les colonnes des (tableaux) dans imonc par rapport à vos besoins. D'une part, pour chaque tableau on peut régler les en-têtes les colonnes qui doivent être affichées, d'autre part pour chaque service de communication il y a un tableau différent, appel Tél, fax, on peut régler le moment où les Infos doivent être affichées.

7.2.5. Concernant les appels tél

L'annuaire Téléphonique sera uniquement vu, que si la variable TELMOND_LOG est placée sur 'yes', sinon aucun journal d'appels ne sera conservée. Dans cet annuaire sera enregistré tous les

appels téléphoniques qui seront entrées sur le routeur. Vous pouvez commuter entre les appels enregistrés sur le PC local et les appels enregistrés sur le routeur, vous pouvez effacer le fichier sur le routeur avec le bouton-Réinitialisé.

Dans l'annuaire téléphonique, vous pouvez cliquer avec le bouton droit de la souris sur le Numéro de Tél pour attribuer un Nom au numéro, comme cela le Nom apparaîtra à la place du Numéro de Tél.

7.2.6. Concernant les connexions

L'affichage des connexions internet par le routeur dans une page est utilisé de puis la Version 1.4, elle donnera une bonne vue d'ensemble du comportement du routeur connecté à Internet. Pour voir cette page la variable `IMOND_LOG` doit être placée sur 'yes' dans le fichier `/config/base.txt`.

De la même façon que l'annuaire-Tél, vous pouvez commuter les connexions enregistrées localement et celles enregistrées sur le routeur. Vous pouvez aussi effacer le fichier des données sur le routeur en cliquant sur le bouton-rafraîchir.

Affichage du tableau de connexions.

- Nom du FAI
- Date et heure de départ
- Date et heure de fin
- Temps en ligne
- Prix de l'unité
- Prix Total
- Réception du signal
- Émission du signal

7.2.7. Concernant les FAX

Pour que soit affichée la page FAX il faut installer le paquetage `OPT_MGETTY` par M. Michael Heimbach sur le routeur ou `OPT_MGETTY` par M. Felix Eckhofer. Sur le site Internet de fli4l, à la page d'accueil vous avez un raccourci pour les paquetages-OPT. Dans cette fenêtre tous les FAX reçus seront enregistrés, le menu contextuel offre plusieurs options de configuration qui seront uniquement disponibles en mode Administrateur :

- Concernant les Fax reçus, il faut correctement configurer le chemin d'accès pour fli4l dans répertoire Admin/Remoteupdate, pour que les FAX reçus sur le routeur soient enregistrés et compressés avec le programme `gzip`, qui se trouve dans le paquetage fli4l, le programme `gzip.exe` et le fichier `win32gnu.dll` peuvent aussi être copié dans le répertoire `imonc`. Si `gzip.exe` n'est pas trouvé dans l'un des deux emplacements, et si le routeur est connecté à Internet, il recherchera le programme sur internet (directement sur le site CGIs).
- Supprimer un FAX reçu. Cela signifie que le FAX sera supprimé sur votre PC local et sur le routeur (le fichier FAX réel, et aussi dans le fichier log).
- Supprimer tous les FAX présents sur routeur. Ici tous les FAX sur le routeur dans le fichier log seront effacés. Les FAX ne seront pas effacés du fichier log de votre PC local.

Comme dans la page des appels Tél, vous pouvez commuter entre les Fax enregistrer localement et les Fax enregistrés sur le routeur.

7.2.8. Concernant les courriels

Cette page apparaît, si dans le répertoire Config courriel il y a au moins un compte courriel avec serveur POP3 qui a été configuré et activé.

Description de la page courriel. Maintenant on a intégré dans cette section le contrôleur de courriel. Si l'option "le routeur n'est pas en ligne" dans config courriel n'est pas activée, le contrôleur de Mail vérifiera tous les comptes courriel, ensuite il utilisera l'intervalle Temps pour vérifier le Serveur (le routeur doit être connecté, il utilise le circuit présélectionné). Si le routeur n'est pas connecté, activer l'option "le routeur n'est pas en ligne" et indiquer le circuit à utiliser, il établira une connexion en utilisant le circuit choisi et téléchargera les courriels de tous les comptes courriels configurés ensuite il fermera la connexion. Pour utiliser cette option vous devrez placer Dialmode sur "auto".

Si des courriels sont disponible sur le serveur POP3, le programme courriel client sera démarré automatiquement ou une icône apparaîtra près de l'heure dans la barre de tâche, il indiquera le nombre de courriel sur le serveur. En double cliquant dessus l'ensemble du courriel client sera lancés. Si une erreur se produit sur un compte courriel, d'une part, une note sur l'erreur sera écrit dans le dossier Histoire du courriel, d'autre part, l'icone du courriel affichera dans le coin supérieur droit une couleur rouge.

Dans la fenêtre courriel, on peut effacer directement les courriels sur le Serveur sans les avoir préalablement téléchargés. Il faut avoir téléchargé les en-têtes des courriels, vous devez marquer les cellules à supprimer, puis en cliquant sur le bouton droit de la souris le menu contextuel s'ouvre, et cliquer sur Delete MailMessage.

7.2.9. Admin

Cette partie est uniquement disponible si imonc est démarré en mode Admin.

Premier point, cette page offre une vue d'ensemble des circuits utilisés, –les fournisseurs d'accès Internet – qui ont été choisis automatiquement par le routeur (par l'intermédiaire du LC Routing). En double cliquant sur un fournisseur d'accès dans l'aperçu fournisseur d'accès vous obtiendrez l'affichage des définitions des plages horaires pour ce fournisseur qui à été défini dans /config/base.txt.

Deuxième point, cette page donne l'occasion d'installer les mises à jour à distance sur le routeur. Vous pouvez choisir l'un ou les cinq programmes (Kernel, fichier système, fichier OPT, rc.cfg et syslinux.cfg) qui seront copiés sur le routeur. Pour pouvoir faire la mise à jour à distance, vous devrez indiquer le répertoire de fli4l dans imonc et les fichiers nécessaires à copier. En plus, vous devez écrire le sous-répertoire des fichiers de configuration (par défaut : /config/*.txt) pour devez construire tous les fichiers systèmes de fli4l. Il est conseillé de Rebooter (ou redémarrer) après avoir envoyé les fichiers système sur le routeur pour que les modifications soient prises en compte. Si un mot de passe est demandé par le routeur, il est inscrit dans la variable PASSWORD dans /config/base.txt.

Troisième point, cette page traite des contraintes du Port Forwarding, un port est connecté exactement et uniquement à un ordinateur client. Maintenant il est possible d'éditer et de configurer Port-Forwarding du routeur. après les modifications des ports ils seront activées, la connexion doit être active. Puisque les fichiers sont enregistrés dans la mémoire virtuelle (Ram-disk), tous les changements seront uniquement sauvegardés jusqu'au prochain redémarrage du routeur. Pour sauvegarder des changements de manière permanente vous devez changer des Port Forward dans le fichier /config/base.txt et installer le nouveau fichier-OPT sur le routeur.

Quatrième point, dans la fenêtre Admin, puis – fichier – vous pouvez utilisée et voir la configuration des fichiers Log du routeur, en cliquant simplement sur la souris. La liste de choix peut être configurée dans le dossier config-Admin d'imonc "voir les fichiers sur le routeur". Ensuite, vous pouvez simplement choisir les fichiers qui sont indiqués dans le menu déroulant.

Cinquième point, cette fenêtre montre DynEisfair log, elle apparaît uniquement si dans le répertoire de configuration Config-Admin les enregistrements les données pour un accès à un compte DynEisfair a été configuré (pour simuler une IP fixe, lorsque l'on a une IP dynamique). Si cela est fait, le fichier log des services sera indiqué dans cette fenêtre.

Dernier point, fenêtre hôtes, tous les ordinateurs enregistrés dans le fichier /etc/hosts sont indiqués ici, à l'avenir on essaiera de configurer chacun des ordinateurs enregistrés pour pouvoir les "pinger" (ou interroger) individuellement, ainsi on pourra rapidement vérifier l'ordinateur qui est connecté au réseau local.

7.2.10. Concernant les erreurs syslog et firewall

Les pages erreur, syslog et Firewall (pare-feu), s'affiche uniquement s'il y a des événements enregistrés dans ce fichier, en plus il faut être en mode Admin pour que les pages soit affichées.

Toutes les erreurs spécifiques à imonc/imond seront enregistrées dans la fenêtre erreur. Si vous avez des problèmes vous pouvez aller vérifier dans cette liste pour voir les causes des erreurs que vous avez rencontrées.

Dans la fenêtre Syslog les messages de syslog seront affichés, excepté des messages du pare-feu. Ceux-ci sont affichés dans une page indépendante (voir ci-dessous). Pour que la page syslog fonctionne vous devrez placer la variable OPT_SYSLOGD sur "yes" dans le fichier de configuration /config/base.txt En plus dans la variable SYSLOGD_DEST on doit placer l'adresse IP du client qui bien entendu utilise imonc (par exemple : SYSLOGD_DEST='@ 100.100.100.100 – adresse IP de votre client !). Il n'y aura pas que les messages syslog qui seront affichés, mais aussi la date, l'heure, l'IP et le niveau de priorité.

Des messages du Firewall (pare-feu) seront affichés dans une page indépendante. Pour que la page fonctionne, vous devez placer la variable OPT_KLOGD sur 'yes' dans le fichier de configuration /config/base.txt.

7.2.11. Concernant les News

Cette page News (ou d'actualité), doit être activée dans le répertoire config-Imonc. Les News mentionnés sur la page accueil du site fli4l, seront visibles directement dans Imonc à la page accueil. On peut directement aller sur le site <http://www.fli4l.de/german/news.xml> avec le bouton-plus. Vous avez une fenêtre à côté des titres des News, qui indique les 10 derniers paquetages-OPT enregistrés sur le site http://www.fli4l.de/german/imonc_opt_show.php, en double cliquant sur le paquetage choisi, vous allez directement sur le site. En plus, il est indiqué dans la barre de statut en bas de Imonc, les titres des News.

7.3. Client imonc pour Unix/Linux

Il y a deux versions pour le Linux : une version de base (imonc) en texte uniquement et une version avec une interface graphique (ximonc). On peut trouver dans le répertoire /src les fichiers sources pour ximonc. La documentation pour le ximonc sera disponible dans la version

1.5 finale. Les utilisateurs expérimentés de Linux ne devraient pas avoir de problème avec les fichiers sources.

Nous nous limiterons ici à la version de base imonc en texte : C'est un programme qui fonctionne uniquement par commande clavier. il n'a donc aucune interface graphique. Les fichiers sources peuvent être trouvés dans le répertoire unix.

Installation :

```
cd unix
make install
```

Imonc est installé dans /usr/local/bin

Démarrer le programme :

```
imonc "hostname"
```

Le nom ou adresse IP du routeur fli4l doit être indiqué à la place de "hostname", par exemple.

```
imonc fli4l
```

imonc montre les information suivantes :

- Data/Heure du routeur fli4l
- La connexion du FAI du moment
- Le Circuit par défaut (Default-Route-Circuits)
- Le canal ISDN (numéris)

Status : Appel Tél en-ligne/déconnecté

Name : Le numéro de Téléphone du Fournisseur d'accès

Time : Temps de connexion

Charge-Time : Connexion par unité de temps

Charge : Prix de la connexion

Les commandes sont :

N°	Commande	Signification
0	quit	Arrêt du programme
1	enable	Activer
2	disable	Déactiver
3	dial	Composer le N°
4	hangup	Raccrocher
5	reboot	Redémarrer
6	timetable	Table de plage horaire
7	dflt route	Nouveau Default-Route-Circuit
8	add channel	Ajouter le deuxième canal
9	rem channel	Supprimer le deuxième canal

Explication des commandes :

0 – quit Quitter le serveur imond, le programme est arrêté.

1 – enable Tous les circuits seront placés en numérotation "auto". C'est l'état par défaut de fli4l après l'avoir initialisé. Cela signifie : lorsqu'il y a une demande de connexion du réseau interne sur Internet, fli4l composera automatiquement le numéro de Tél du FAI.

- 2 – **disable** Tous les circuits du mode de composition seront placés sur "OFF". Après cette action fli4l est presque "mort" (ou en sommeil). fli4l sera réveillé au moyen de la commande "enable".
- 3 – **dial** Composer manuellement le numéro de Tél du FAI, cette fonction sert de test. Puisque cette commande est normalement sur automatique par l'intermédiaire du circuit par défaut. Elle est utilisée pour des essais, depuis que fli4l existe la connexion est habituellement automatique.
- 4 – **hangup** Raccrocher manuellement : de cette façon, on peut devancer le raccrochement automatique de fli4l.
- 5 – **reboot** fli4l sera redémarré. Commande pas vraiment utile...
- 6 – **timetable** Table des Horaires pour arrêter ou démarrer les circuits par défaut voir les détails page précédente.
- 7 – **default route circuit** Changer manuellement un circuit par défaut. Peut être logique par ex. pour arrêter un moment LC-Routing automatique de fli4l, parfois les fournisseurs ne permettent pas l'accéder à votre propre boîte mail, vous devez utiliser un autre fournisseur d'accès.
- 8 – **add channel** On l'utilise pour ajouter le deuxième canal ISDN (ou numéris en français). Vous devez placer la variable `ISDN_CIRC_x_BUNDLING` sur 'yes'.
- 9 – **remove channel** Coupe le deuxième canal ISDN. Voir également "add channel".

Avec les commandes imond, les mêmes remarques son valables qu'avec le client `imonc.exe` sous Windows.

Remarque complémentaire : Avec la version 1.4 de fli4l il est maintenant possible d'installer un client imonc "allégé" sur le Routeur de fli4l. Pour se faire il faut placer le paquetage option sur `OPT_IMONC='yes'` dans le paquetage `TOOLS`.

De cette façon, on peut maintenant configurer certains paramètres avec imonc par ex. pour faire du routage, etc. en utilisant la console fli4l. Attention : Ce Mini-imonc fonctionne uniquement sur le routeur fli4l ! Sous Linux/Unix, il faut toujours utiliser le Client imonc/unix "son grand frère".

A. Annexe du paquetage de Base

A.1. Câble Null-modem

Pour utiliser le programme facultatif PPP (Page ??) vous aurez besoin d'un câble null-modem.

Il faut avoir branchées au moins trois fils sur le connecteur voici le schéma :



Vous devez souder les fils sur les broches du connecteur en suivant le schéma.

A.2. Console par câble Série

fi4l peut être utilisé sans écran et sans clavier. L'inconvénient, vous ne pourrez pas voir les messages d'erreur du boot, et les messages ne peuvent pas être réorientés vers l'interface de syslog.

Une possibilité est de réorienter les messages de la console sur son PC ou un terminal classique en utilisant l'interface série. La configuration s'effectue avec les variables suivantes [SER_CONSOLE](#) (Page 29), [SER_CONSOLE_IF](#) (Page 29) et [SER_CONSOLE_RATE](#) (Page 29).

Les ordinateurs avec des cartes mères anciennes ne soutiennent pas des vitesses supérieures à 38400 Baud. Par conséquent, il faudra d'abord essayer avec 38400 Baud, avant d'utiliser des vitesses plus élevées. Puisque seules des sorties texte sont écrites sur la console, des vitesses plus élevées ne sont pas nécessaires.

Maintenant, tous les messages sont envoyés sur la console par le port série, ainsi que les messages de Boot (ou démarrage) !

Le câble [Null modem](#) (Page 106) est utilisé entre l'émulation et le terminal ou le PC. Nous déconseillons toutefois d'utiliser un câble null modem standard, parce que normalement toutes les connexions de la prise serie sont branchés. Si le terminal ou le PC ne reçoit rien (ou l'émulateur n'arrive pas à émettre) avec la connexion fli4l, cela peut venir de l'utilisation du câble null modem standard !

Par conséquent, un câblage spécial est nécessaire, pour pouvoir arrêter fli4l avec le terminal du PC. Pour cela il suffit de brancher uniquement les 3 broches dans le connecteur, tous les autres contacts du connecteur ne sont pas utilisés (pas de parasite). Voir le câblage du [câble Null-modem](#) (Page 106).

A.3. Programmes

Pour économiser de la place sur le média on utilise le packaging "BusyBox". Qui est un programme exécutable standard Unix unique, dans lequel est incorporé :

```
[, [[, arping, ash, base64, basename, bbconfig, blkid, bunzip2, bzip2,
cat, chgrp, chmod, chown, chroot, cmp, cp, cttyhack, 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, lsmod, lzcat, makedevs, md5sum, mdev, mkdir, mknod,
mkswap, modprobe, mount, mv, nameif, nice, nslookup, ping, ping6, poweroff,
ps, pscan, pwd, reboot, reset, rm, rmdir, 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
```

Ce sont principalement des "mini-programmes", ils ne couvrent pas toutes les fonctions, cependant ils suffisent à remplir les demandes modestes de fli4l.

BusyBox est sous licence GPL et les fichiers sources sont complètement accessibles.

<http://www.busybox.net/>

A.4. Autre outils-i4l

Il y a beaucoup d'autres outils, pour isdn4linux, et aussi pour enrichir fli4l. Le problème est malheureusement un manque de place ! On pourrait utiliser isdnlog comme outil largement plus approprié pour le calcul des connexions en ligne, mais isdnlog est simplement trop gros pour une installation sur un média !

Imond a besoin d'au moins 10% de place sur le média, pour l'utilisation de contrôles et du Routing-LC, même si se n'est pas tout à fait parfait.

A.5. Dépannage

On peut dépister les erreurs en les lisent sur l'écran de contrôle, après le boot de fli4l ils sont affichées uniquement sur la dernière page de l'écran. Pour pouvoir lire les pages précédentes

ou suivantes, vous devez utiliser les touches MAJUSCULE [PAGE PREC] et MAJUSCULE [PAGE SUIV].

A l'installation du routeur si vous avez un message d'erreur du genre "try-to-free-pages" qui apparaît, ce message indique que vous n'avez pas assez de mémoire RAM pour les programmes utilisés. Comme solution les options suivantes sont alors disponibles :

- augmenter la mémoire RAM
- utiliser moins de paquetage-Opt à l'installation
- effectuer une installation sur le disque dur avec [Type B](#) (Page 11)

Le fichier proc peut également aider à dépister des erreurs, par exemple :

```
cat /proc/interrupts
```

Avec le paramètre Interrupts on peut visualiser les pilotes matériels et ceux qui ne sont pas activés !

Voici d'autres paramètres intéressants avec la commande /proc : dma, ioports, kmsg, meminfo, modules, uptime, version et pci (si le routeur a un Bus-PCI).

Le plus souvent il s'agit d'un problème de connexion avec ipppd, en particulier lors de l'authentification, vous pouvez utiliser les variables dans config/base.txt

```
OPT_SYSLOGD='yes'
```

```
OPT_KLOGD='yes'
```

et dans config/isdn.txt

```
ISDN_CIRC_x_DEBUG='yes'
```

pour essayer de résoudre certains problèmes.

A.6. Références

- 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. Préfixe

Les unités préfixer, abordé dans ce présent document sont après la norme IEC 60027-2. Voir : <http://physics.nist.gov/cuu/Units/binary.html>. Pour les unités en français voir : <http://fr.wikipedia.org/wiki/Octet>

A.8. Aucune responsabilité et de garantie

Naturellement on ne garantit pas que tous les paquetages-fli4l fonctionnent ou que tous les dossiers ou sous dossiers de cette documentation soit correcte.

Toute responsabilité pour les dommages causés et éventuellement pour les frais engager seront déclinés !

A.9. Merci

Dans cette partie de cette documentation, je remercie toutes les personnes qui ont contribué ou beaucoup plus contribué au développement de fli4l. Voici ceux qui mon autorisé à mentionner leurs noms.

A.9.1. Fondateur du Projet

Meyer, Frank

Frank a commencé le projet fli4l le 04.05.2000 !

Voir : <http://www.fli4l.de/fr/fli4l/caracteristique/historique/>

A.9.2. L'équipe de développeurs et de testeurs

L'équipe fli4l de développeurs est formée (dans l'ordre alphabétique) :

Charrier, Bernard (*traduction française*)
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, site Web, gestion des versions*)
Schliesing, Manfred (*testeur*)
Schulz, Christoph (*FBR, IPv6, Kernel*)
Siebmanns, Harvey (*Documentation, Traduction anglaise*)
Spieß, Carsten (*Dsltool, Hwsupp, Rrdtool, Webgui*)
Vosselman, Arwin (*LZS-Compression, Documentation*)
Weiler, Manuela (*Copie de CD, trésorière*)
Weiler, Marcel (*Gestion de la qualité*)
Wolters, Florian (*Firmware, Kernel*)

A.9.3. L'équipe de développeurs et de testeurs (qui ne sont plus actifs)

Arndt, Kai-Christian (*USB*)
Bauer, Jürgen (*LCD-Package, fliwiz*)
Behrends, Arno (*Support*)
Blokland, Kees (*Traduction anglaise*)
Bork, Thomas (*lpdsrv*)
Bußmann, Lars (*testeur*)
Cerny, Carsten (*Site Web, fliwiz*)
Dawid, Oliver (*dhcp, uClibc*)
Ebner, Hannes (*QoS*)
Fischer, Joerg (*testeur*)
Frauenhoff, Peter (*testeur*)
Grabner, Hans-Joerg (*imonc*)
Grammel, Matthias (*Traduction anglaise*)
Gruetzmacher, Tobias (*Mini-httpd, imond, proxy*)
Hahn, Joerg (*IPSEC*)
Hanselmann, Michael (*Mac OS X/Darwin*)
Hoh, Jörg (*Newsletter, NIC-DB, manifestation*)
Hornung, Nicole (*Verein*)
Horsmann, Karsten (*Mini-httpd, WLAN*)
Janus, Frank (*LCD*)
Kaiser, Gerrit (*Logo*)
Karner, Christian (*PPTP-Package*)
Klein, Marcus (*Problèmes réactions*)
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 (*testeur*)
Strigler, Stefan (*GTK-Imonc, Opt-DB, NG*)
Wallmeier, Nico (*Windows-Imonc*)
Walter, Gerd (*UMTS*)
Walter, Oliver (*QoS*)
Wolter, Jean (*Paketfilter, uClibc*)
Zierer, Florian (*Liste de souhaits*)

A.9.4. Sponsor

Le nom et le logo de fli4l sont enregistrée comme marque déposée. Les utilisateurs suivants (et ceux qu'ils ne veulent pas être nommés) ont aidé financièrement au développement de fli4l :

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

Depuis un certain temps, fli4l a maintenant ses propres sponsors, ils soutiennent le développement de fli4l par (des dons de matériels). Il s'agit d'adaptateurs de Compact Flash et de cartes Ethernet.

Donateurs de matériel (dans l'ordre alphabétique) :

Baglatzis, Stephanos
Bauer, Jürgen
Dross, Heiko
Kappenhagen, Wenzel
Kipka, Joachim
Klopper, Tom
Peiser, Steffen
Reichelt, Detlef
Reinard, Louis
Stärkel, Christopher

Une liste des autres sponsors est sur la page d'accueil de fli4l :

<http://www.fli4l.de/fr/divers/sponsors/>

A.10. Réaction

Les critiques et les réactions sont toujours les bienvenues pour la collaboration de fli4l.

Pour les services d'aide, adressez-vous sur les Newsgroups de fli4l. Si vous avez des problèmes d'installation avec le routeur fli4l, voir avant de s'adresser au Newsgroups, les FAQ, Howtos et les archives Newsgroups. On trouve sur le site Web fli4l différentes informations et en plus d'autres sites internet au sujet de fli4l :

<http://www.fli4l.de/fr/aide/newsgroup/>

<http://www.fli4l.de/fr/aide/faq/>

<http://www.fli4l.de/fr/aide/howtos/>

C'est justement parce qu'on utilise en général du vieux matériel pour le routeur fli4l, que l'on peut avoir des problèmes avec ce genre de matériel. Ces informations peuvent aider d'autres utilisateurs fli4l à résoudre les problèmes de matériel, car il y a sans cesse des problèmes avec les cartes installées dans le PC par rapport aux adresses I/O, aux Interruptions, et autres.

Sur le site Web fli4l une banque de données pour les cartes réseau et wireless, sur laquelle on peut écrire les informations, par exemple, le pilote correspondant à une carte déterminée et la compatibilité avec fli4l. Voici l'adresse du site :

<http://www.fli4l.de/fr/aide/bd-cartes-reseaux/>

Amusez-vous bien avec fli4l !

Table des figures

3.1. Structure du Filtrage de paquets	38
3.2. Structure du répertoire fli4l	47
5.1. Paramètre	78
5.2. Paramètre pour la mise à jour	79
5.3. Paramètre pour pré-installation du DD	80

Liste des tableaux

3.1. Aperçu des paquetages supplémentaires	13
3.2. Réglage Automatique du nombre de connexions maximum	27
3.3. Types de préfixes réseau	36
3.4. Action des règles du filtrage de paquets	40
3.5. Restrictions de la source et de destination dans les règles de filtrage de paquets	41
3.6. Restriction des règles sur de filtrage de paquets	43
3.7. Modèles inclus dans fli4l de base	46
3.8. Disponibilité de Conntrack Helpers dans le filtrage de paquets	63
3.9. Format du fichier Log d'Imond	67

Index

base.txt, [13](#)
BEEP, [28](#)
BOOT_TYPE, [23](#)
BOOTMENU_TIME, [24](#)
BUILDDIR, [81](#)

COMP_TYPE_OPT, [26](#)
COMP_TYPE_ROOTFS, [26](#)
CONSOLE_BLANK_TIME, [28](#)

DEBUG_ENABLE_CORE, [30](#)
DEBUG_IP, [30](#)
DEBUG_IPTABLES, [30](#)
DEBUG_MDEV, [30](#)
DEBUG_MODULES, [30](#)
DEBUG_STARTUP, [29](#)
DIALMODE, [67](#)
DNS_FORWARDERS, [64](#)
DOMAIN_NAME, [64](#)

Exemple de fichier (base.txt), [13](#)

FILESONLY, [81](#)
FLI4L_UUID, [26](#)
ftp, [63](#)

h323, [63](#)
HOSTNAME, [22](#)
HOSTNAME_ALIAS_N, [65](#)
HOSTNAME_ALIAS_x, [65](#)
HOSTNAME_IP, [64](#)

IMOND_ADMIN_PASS, [65](#)
IMOND_BEEP, [66](#)
IMOND_DIAL, [66](#)
IMOND_ENABLE, [66](#)
IMOND_LED, [65](#)
IMOND_LOG, [66](#)
IMOND_LOGDIR, [66](#)
IMOND_PASS, [65](#)

IMOND_PORT, [65](#)
IMOND_REBOOT, [66](#)
IMOND_ROUTE, [66](#)
IP_CONNTRACK_MAX, [27](#)
IP_DYN_ADDR, [67](#)
IP_NET_N, [33](#)
IP_NET_x, [33](#)
IP_NET_x_COMMENT, [35](#)
IP_NET_x_DEV, [34](#)
IP_NET_x_MAC, [34](#)
IP_NET_x_NAME, [35](#)
IP_NET_x_TYPE, [35](#)
IP_ROUTE_N, [37](#)
IP_ROUTE_x, [37](#)
irc, [63](#)

KERNEL_BOOT_OPTION, [26](#)
KERNEL_VERSION, [26](#)
KEYBOARD_LOCALE, [31](#)

LIBATA_DMA, [24](#)
LOCALE, [28](#)
LOGIP_LOGDIR, [70](#)

Masquerading, [61](#)
MKFLI4L_DEBUG_OPTION, [82](#)
MOUNT_BOOT, [24](#)

NET_DRV_N, [31](#)
NET_DRV_x, [32](#)
NET_DRV_x_OPTION, [32](#)
NET_PREFIX_x, [35](#)
NET_PREFIX_x_NAME, [35](#)
NET_PREFIX_x_STATIC_IPV4, [36](#)
NET_PREFIX_x_STATIC_IPV6, [36](#)
NET_PREFIX_x_TYPE, [35](#)
NET_PREFIX_x_ULA_DEV, [36](#)

OPT_HOTPLUG_PCI, [72](#)

- OPT_IMOND, [65](#)
- OPT_KLOGD, [70](#)
- OPT_LOGIP, [70](#)
- OPT_MAKEKBL, [31](#)
- OPT_NET_PREFIX, [35](#)
- OPT_PNP, [71](#)
- OPT_SYSLOGD, [68](#)
- OPT_Y2K, [70](#)

- PASSWORD, [22](#)
- PF_FORWARD_ACCEPT_DEF, [50](#)
- PF_FORWARD_LOG, [50](#)
- PF_FORWARD_LOG_LIMIT, [50](#)
- PF_FORWARD_N, [50](#)
- PF_FORWARD_POLICY, [49](#)
- PF_FORWARD_REJ_LIMIT, [50](#)
- PF_FORWARD_UDP_REJ_LIMIT, [50](#)
- PF_FORWARD_x, [50](#)
- PF_FORWARD_x_COMMENT, [50](#)
- PF_INPUT_ACCEPT_DEF, [48](#)
- PF_INPUT_ICMP_ECHO_REQ_LIMIT, [49](#)
- PF_INPUT_ICMP_ECHO_REQ_SIZE, [49](#)
- PF_INPUT_LOG, [49](#)
- PF_INPUT_LOG_LIMIT, [49](#)
- PF_INPUT_N, [49](#)
- PF_INPUT_POLICY, [48](#)
- PF_INPUT_REJ_LIMIT, [49](#)
- PF_INPUT_UDP_REJ_LIMIT, [49](#)
- PF_INPUT_x, [49](#)
- PF_INPUT_x_COMMENT, [49](#)
- PF_LOG_LEVEL, [48](#)
- PF_NEW_CONFIG, [37](#)
- PF_OUTPUT_ACCEPT_DEF, [51](#)
- PF_OUTPUT_CT_ACCEPT_DEF, [63](#)
- PF_OUTPUT_CT_N, [63](#)
- PF_OUTPUT_CT_x, [63](#)
- PF_OUTPUT_CT_x_COMMENT, [63](#)
- PF_OUTPUT_LOG, [51](#)
- PF_OUTPUT_LOG_LIMIT, [51](#)
- PF_OUTPUT_N, [51](#)
- PF_OUTPUT_POLICY, [51](#)
- PF_OUTPUT_REJ_LIMIT, [51](#)
- PF_OUTPUT_UDP_REJ_LIMIT, [51](#)
- PF_OUTPUT_x, [51](#)
- PF_OUTPUT_x_COMMENT, [51](#)
- PF_POSTROUTING_N, [53](#)
- PF_POSTROUTING_x, [53](#)
- PF_POSTROUTING_x_COMMENT, [53](#)
- PF_PREROUTING_CT_ACCEPT_DEF, [63](#)
- PF_PREROUTING_CT_N, [63](#)
- PF_PREROUTING_CT_x, [63](#)
- PF_PREROUTING_CT_x_COMMENT, [63](#)
- PF_PREROUTING_N, [53](#)
- PF_PREROUTING_x, [53](#)
- PF_PREROUTING_x_COMMENT, [53](#)
- PF_USR_CHAIN_N, [52](#)
- PF_USR_CHAIN_x_NAME, [52](#)
- PF_USR_CHAIN_x_RULE_N, [52](#)
- PF_USR_CHAIN_x_RULE_x, [52](#)
- PF_USR_CHAIN_x_RULE_x_COMMENT, [52](#)
- POWERMANAGEMENT, [26](#)
- pptp, [63](#)
- PXESUBDIR, [81](#)

- REMOTEHOSTNAME, [81](#)
- REMOTEPATHNAME, [81](#)
- REMOTEPORT, [81](#)
- REMOTEREMOUNT, [81](#)
- REMOTEUPDATE, [81](#)
- REMOTEUSERNAME, [81](#)
- RTC_SYNC, [24](#)

- sane, [63](#)
- SER_CONSOLE, [28](#)
- SER_CONSOLE_IF, [29](#)
- SER_CONSOLE_RATE, [29](#)
- sip, [63](#)
- snmp, [63](#)
- SQUEEZE_SCRIPTS, [82](#)
- SSHKEYFILE, [81](#)
- SYSLOGD_DEST_N, [68](#)
- SYSLOGD_DEST_x, [68](#)
- SYSLOGD_RECEIVER, [68](#)
- SYSLOGD_ROTATE, [69](#)
- SYSLOGD_ROTATE_AT_SHUTDOWN, [70](#)
- SYSLOGD_ROTATE_DIR, [69](#)
- SYSLOGD_ROTATE_MAX, [69](#)

- tftp, [63](#)
- TFTPBOOTIMAGE, [81](#)

TFTPBOOTPATH, [81](#)

TIME_INFO, [24](#)

VERBOSE, [81](#)

Y2K_DAYS, [70](#)