

fli4l – flexible internet router for linux

Version 3.10.4

L'équipe fli4l
courriel: team@fli4l.de

25 octobre 2015

Table des matières

1. Documentation du packaging base	9
1.1. Introduction	9
2. Installation et configuration	12
2.1. Décompacter les archives	12
2.2. Configuration	13
2.2.1. Éditer les fichiers de configurations	13
2.2.2. Configuration via un fichier de configuration spéciale	13
2.2.3. Variables	14
2.3. Procédures d'installation	14
2.3.1. Routeur sur une clé USB	15
2.3.2. Routeur sur CD ou par le réseau	15
2.3.3. Type A : Routeur sur disque dur – Une seule partition FAT	15
2.3.4. Type B : Routeur sur disque dur – Partition FAT et ext3	15
3. Configuration de la base	16
3.1. Exemple de fichier	17
3.2. Configuration générale	23
3.3. Configuration de la console	27
3.4. Fichier log pour la séquence de Boot et du chargement des modules	28
3.5. Réglage personnel dans opt/etc/inittab	29
3.6. Configuration du clavier	30
3.7. Pilotes des cartes réseaux Ethernet	30
3.8. Réseaux	38
3.9. Route supplémentaire (optionnel)	40
3.10. Le filtrage de paquets	40
3.10.1. Action pour le filtrage de paquets	42
3.10.2. Restriction dans les règles	43
3.10.3. Utilisation d'un modèle pour le filtrage de paquets	46
3.10.4. Configuration du filtrage de paquets	50
3.10.5. Exemples	56
3.10.6. Configuration par défaut	59
3.10.7. DMZ – Zone démilitarisée	63
3.10.8. Conntrack Helpers	63
3.11. Configuration du domaine	66
3.12. Configuration de Imond	67
3.13. Configuration du circuit général	69

4. Les paquetages	70
4.1. Outils dans le paquetage de base	70
4.1.1. OPT_SYSLOGD - Enregistre tous les messages du système	70
4.1.2. OPT_KLOGD - Messages du Kernel lors du boot	72
4.1.3. OPT_LOGIP - Journalisation des adresses IP-WAN	72
4.1.4. OPT_Y2K - Correctif pour avant l'année 2000	72
4.1.5. OPT_PNP - Installation des cartes ISAPnP	73
4.2. Advanced Networking	74
4.2.1. Relais broadcast - Transmission par IP broadcast	75
4.2.2. Bonding - Regroupées plusieurs cartes réseaux pour avoir un seul lien	75
4.2.3. VLAN - Supporte le 802.1Q	79
4.2.4. Périphérique MTU - Réglage du MTU	80
4.2.5. BRIDGE - Pont Ethernet pour fli4l	81
4.2.6. Remarque	83
4.2.7. EBTables - EBTables pour fli4l	84
4.2.8. ETHTOOL - Paramètres pour carte réseau Ethernet	84
4.2.9. Exemples	85
4.3. CHRONY - Protocole serveur/client pour la diffusion de l'heure sur le réseau	87
4.3.1. Configuration de l'OPT_CHRONY	88
4.3.2. Aide	89
4.3.3. Littératures	89
4.4. DHCP_CLIENT - Configuration du protocole dynamic pour les hôtes	89
4.4.1. OPT_DHCP_CLIENT	89
4.5. DNS_DHCP - Serveur DNS et DHCP - Relay DHCP et serveur DNS esclave	90
4.5.1. Nom d'hôte	90
4.5.2. Serveur DNS	92
4.5.3. Serveur DHCP	97
4.5.4. Relais DHCP	100
4.5.5. Serveur TFTP	101
4.5.6. YADIFA - Serveur DNS esclave	101
4.6. DSL - DSL pour PPPoE, Fritz!DSL et PPTP	102
4.6.1. Variables de configuration générales	103
4.6.2. OPT_PPPOE - DSL avec PPPoE	106
4.6.3. OPT_PPPOE_CIRC - Plusieurs accès DSL avec PPPoE (Expérimental)	108
4.6.4. OPT_FRITZDSL - DSL avec carte Fritz!DSL	108
4.6.5. OPT_PPTP - DSL avec PPTP pour l'Autriche/les Pays-Bas (Expérimental)	109
4.6.6. OPT_POESTATUS - Moniteur pour l'état du PPPoE sur la console fli4l	110
4.7. DYNDNS - Mise à jour dynamiques des services de noms de domaine	110
4.8. EASYCRON - Exécuter une commande planifiée	116
4.8.1. Configuration	116
4.8.2. Exemples	117
4.8.3. Conditions	117
4.8.4. Installation	117
4.9. HD - Supporte les disques dur, CompactFlash, clé USB, ...	118
4.9.1. OPT_HDINSTALL - Installation sur disque dur/CompactFlash	118
4.9.2. OPT_MOUNT - Montage automatique du système de fichiers	120

4.9.3.	OPT_EXTMOUNT - Montage manuel du fichier système	121
4.9.4.	OPT_HDSLEEP - Règle pour l'arrêt automatique du disque dur	121
4.9.5.	OPT_RECOVER - Option de secours	122
4.9.6.	OPT_HDDRV - Pilote pour contrôleur de disque dur	122
4.10.	HTTPD - Statut du routeur avec le serveur web	123
4.10.1.	OPT_HTTPD - Mini serveur web comme moniteur de statut	123
4.10.2.	Gestion des utilisateurs	124
4.10.3.	OPT_OAC - Contrôle d'accès en ligne (OAC)	125
4.11.	HWSUPP - Supporte du matériel spécifique	127
4.11.1.	Description	127
4.11.2.	Configuration du paquetage HWSUPP	127
4.11.3.	Paramètre pour expert	131
4.11.4.	Prise en charge des cartes VPN	132
4.12.	IPv6 - Internet protocole version 6	132
4.12.1.	Introduction	132
4.12.2.	Format de l'adresse	133
4.12.3.	Configuration	134
4.12.4.	WebGUI	144
4.13.	ISDN - Communication avec les cartes ISDN (ou Numéris) actives et passives	144
4.13.1.	Établir une connexion par ISDN	144
4.13.2.	Carte ISDN	145
4.13.3.	OPT_ISDN_COMP (Expérimental)	149
4.13.4.	Circuits ISDN	150
4.13.5.	OPT_TELMOND - Configuration telmond	158
4.13.6.	OPT_RCAPID - Le démon CAPI distant	161
4.14.	OpenVPN - Supporte le VPN	162
4.14.1.	OpenVPN - Introduction et exemple	162
4.14.2.	OpenVPN - Configuration	163
4.14.3.	OpenVPN - Configuration du bridge	166
4.14.4.	OpenVPN - Configuration du tunnel	167
4.14.5.	Paramètres experts	169
4.14.6.	OpenVPN - WebGUI	178
4.14.7.	OpenVPN - Aide pour différentes versions OpenVPN	181
4.14.8.	OpenVPN - Exemples	182
4.14.9.	Liens sur le thème OpenVPN	185
4.15.	PCMCIA - Supporte les cartes PC	186
4.15.1.	Pilote PCMCIA	186
4.16.	PPP - Connecter un ordinateur via le port série	186
4.17.	PROXY - Différent Serveur proxy	188
4.17.1.	OPT_PRIVOX - Filtrage de la publicité avec un proxy HTTP	188
4.17.2.	OPT_TOR - Système de communication anonyme pour Internet	191
4.17.3.	OPT_SS5 - Proxy Socks 4/5	192
4.17.4.	OPT_TRANSPROXY (Expérimental) - Proxy HTTP transparent	193
4.17.5.	OPT_SIPPROXY (Expérimental) - Proxy pour Session Initiation Protocol	194
4.17.6.	OPT_IGMPPROXY - Proxy pour Internet Group Management Protocol	194
4.17.7.	OPT_STUNNEL - Tunnel avec une connexion SSL/TLS	201

4.18. QoS - Qualité de service	207
4.18.1. Configuration	207
4.18.2. Applications et exemples	214
4.19. SSHD - Secure-Shell, Secure-Copy	221
4.19.1. Installation du service Secure-shell	221
4.19.2. Installation du dbclient	225
4.19.3. Installation du client plink	226
4.19.4. Installation d'un serveur sftp	226
4.19.5. Littérature	226
4.20. TOOLS - Outils supplémentaires pour le débogage	226
4.20.1. Outils pour le réseau	227
4.20.2. Outils pour la détection du matériel	229
4.20.3. Outils pour fichier	230
4.20.4. Outils pour développeur	231
4.21. UMTS - Connexion UMTS via Internet	231
4.21.1. Configuration	232
4.21.2. Exemple de configuration pour rrdtool	234
4.22. USB - Support pour périphérique USB	234
4.22.1. Problèmes avec les périphériques USB	235
4.22.2. Précautions d'utilisation	235
4.22.3. Monter les périphériques USB	236
4.23. WLAN - Supporte le WLAN (ou réseau sans fil)	236
4.23.1. Configuration du WLAN	236
4.23.2. Exemple	241
4.23.3. Point d'accès virtuel (VAP) (Expérimental)	243
4.23.4. Réglage de l'heure pour l'arrêt du WLAN avec easycron	243
4.23.5. Remarque et dons	243
4.24. SRC - Le Buildroot fli4l	243
4.24.1. Vue d'ensemble des répertoires sources	244
4.24.2. Compiler un programme pour fli4l	244
4.24.3. Tester d'un programme compilé	247
4.24.4. Déboguer un programme compilé	248
4.24.5. Information sur le FBR	251
4.24.6. Modification de la configuration du FBR	252
4.24.7. Mise à jour des FBRs	253
4.24.8. Intégrer vos propres programmes dans le FBR	254
5. Création une archive fli4l/Média de Boot	255
5.1. Création de l'archive fli4l/Média de Boot sous Linux, dérivé Unix et Mac OS X	255
5.1.1. Lignes de commandes optionnelle	255
5.2. Création d'une archive fli4l/Média de Boot sous Windows	258
5.2.1. Ligne de commande en option	258
5.2.2. Boîte de dialogue - Définition du répertoire de configuration	259
5.2.3. Boîte de dialogue - Paramètres généraux	259
5.2.4. Boîte de dialogue - Paramètres pour la mise à jour à distance	260
5.2.5. Boîte de dialogue - Paramètres pour une pré-installation du HD	261
5.3. Paramètre pour le fichier mkfli4l.txt	262

6. Réglage des PCs dans le LAN	265
6.1. Adresse IP	265
6.2. Nom de l'ordinateur et de domaine	265
6.2.1. Windows 2000	265
6.2.2. NT 4.0	266
6.2.3. Windows 95/98	266
6.2.4. Windows XP	266
6.2.5. Windows 7	267
6.2.6. Windows 8	267
6.3. Gateway (ou Passerelle)	267
6.4. Serveur DNS	268
6.5. Divers points	268
7. Interface client/serveur imon	269
7.1. Server imon avec imond	269
7.1.1. Mode de fonctionnement du Moindre-Coût-Routage	269
7.1.2. Calcul des frais on-line (en ligne)	274
7.2. Client Windows imonc.exe	274
7.2.1. Introduction	274
7.2.2. Paramètre de démarrage	275
7.2.3. Concernant l'aperçu de imonc	276
7.2.4. Paramètres de configuration	277
7.2.5. Concernant les appels tél	282
7.2.6. Concernant les connexions	283
7.2.7. Concernant les FAX	283
7.2.8. Concernant les courriels	283
7.2.9. Admin	284
7.2.10. Concernant les erreurs syslog et firewall	285
7.2.11. Concernant les News	285
7.3. Client imonc pour Unix/Linux	285
8. Documentation pour développeur	288
8.1. Règles générales	288
8.2. Compiler les programmes	288
8.3. Concept modulaire	289
8.3.1. mkffi4l	289
8.3.2. Structure	289
8.3.3. Configuration du paquetage	289
8.3.4. Liste des fichiers à copier	291
8.3.5. Analyse des variables de configuration	295
8.3.6. Définitions pour contrôler les variables de configuration	297
8.3.7. Contrôle détaillé de la configuration	302
8.3.8. Supporte différent choix de version du Kernel	317
8.3.9. Documentation	317
8.3.10. Formats de fichier	319
8.3.11. Développer la documentation	319
8.3.12. Programme-Client	319

8.3.13. Code source	320
8.3.14. Les autres fichiers	320
8.4. Conditions générales de création de script pour <code>fli4l</code>	320
8.4.1. Structure	320
8.4.2. Gestion des variables de configuration	321
8.4.3. Recherche d'erreur	322
8.4.4. Remarques	323
8.5. Utiliser le filtrage de paquets	324
8.5.1. Ajouter vos propres chaînes et règles	324
8.5.2. Classer les règles dans une infrastructure	324
8.5.3. Extension pour le test des paquets de filtrage	325
8.6. Création d'un CGI pour le paquetage <code>httpd</code>	326
8.6.1. Informations générales sur le serveur Web	326
8.6.2. Nom du script	326
8.6.3. Configuration du menu	326
8.6.4. Construction d'un script CGI	328
8.6.5. Divers	333
8.6.6. Dépannage	333
8.7. Démarrer, arrêter, se connecter et se déconnecter avec <code>fli4l</code>	333
8.7.1. Concept de Boot	333
8.7.2. Scripts de démarrage et d'arrêt	334
8.7.3. Fonctions auxiliaires	336
8.7.4. Périphériques <code>ttyI</code>	338
8.7.5. Scripts de connexion et de déconnexion par modem	338
8.8. Paquetage Template	340
8.9. Construction du Boot sur un support de données	340
8.10. Fichiers de configurations	341
8.10.1. Configuration du fournisseur	341
8.10.2. Configuration DNS	341
8.10.3. Fichier hôte	342
8.10.4. Configuration de <code>imond</code>	342
8.10.5. Le fichier <code>/etc/.profile</code>	343
A. Supplément du paquetage de Base	344
A.1. Câble Null-modem	344
A.2. Console par câble Série	344
A.3. Programmes	345
A.4. Autre outils- <code>i4l</code>	345
A.5. Dépannage	345
A.6. Références	346
A.7. Préfixe	346
A.8. Aucune responsabilité et de garantie	347
A.9. Merci	347
A.9.1. Fondateur du Projet	347
A.9.2. L'équipe de développeurs et de testeurs	347
A.9.3. L'équipe de développeurs et de testeurs (qui ne sont plus actifs)	349
A.9.4. Sponsor	350

A.10.Réaction	351
B. Supplément des paquetages optionnels	352
B.1. CHRONY - d'autre information sur applications Timewarps	352
B.2. DSL - PPPD et filtre actif	352
B.3. DYNDNS	353
B.3.1. Ajouter un nouveau fournisseur DynDNS	353
B.3.2. Remercement	355
B.3.3. Licence	356
B.4. EASYCRON - Complément de Crontab pour la phase de boot	356
B.5. HD - Rapport d'erreur sur les disques durs/CompactFlashes	357
B.6. HTTPD	358
B.6.1. Paramètre supplémentaire	358
B.6.2. Observation générale	358
B.7. HWSUPP - Paramètres dépendant du périphérique	359
B.7.1. Périphérique disponible pour la LED	359
B.7.2. Bouton disponible du périphérique	360
B.7.3. Note sur le matériel spécifique	361
B.8. HWSUPP - Exemple de configuration	361
B.8.1. PC générique	361
B.8.2. PC engines APU	361
B.8.3. PC engines APU avec GPIO	362
B.9. HWSUPP - Séquence de clignotement de la LED	362
B.10.HWSUPP - Conseil pour les développeurs de paquetage	363
B.10.1.Extension pour LED	363
B.10.2.Extension pour le bouton	364
B.10.3.Action du bouton	364
B.11.IPV6 - Connexion Internet en utilisant un tunnel IPv6 SixXS	365
B.11.1.Créer un compte	365
B.11.2.Configurer le tunnel	365
B.11.3.Configurer le sous-réseaux	367
B.12.ISDN	371
B.12.1.Détails techniques sur la connexion et le routage ISDN	371
B.12.2.Messages d'erreur du sous-système ISDN (Documentation-i4l en Anglais)	372
B.13.UMTS	374
B.13.1. Matériel pris en charge	374
B.13.2. Si l'interface modem n'est pas activée	375
B.14.SRC - Développement de son propre paquetage	376
B.15.Différences entre la version 3.10.4 et 3.6.2	376
B.16.Différences entre la version 3.10.4 et 3.10.3	381
Table des figures	382
Liste des tableaux	383
Index	384

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 255), [Mac OS X](#) (Page 255) et [MS Windows](#) (Page 258)
 - 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 3.14
 - 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-3.10.4.tar.gz
```

Si cela ne fonctionne pas, essayez la procédure suivante :

```
gzip -d < fli4l-3.10.4.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-3.10.4
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-3.10.4/` :

- 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/etc/* Fichiers de configuration par défaut pour de nombreux programmes (normalement ils ne doivent pas être traités).

- opt/files/* 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/full_rc.cfg | grep '^[+-]'
```

```
--- config/build/full_rc.cfg      2007-03-22 15:34:39.085103706 +0100
+++ ma_config/build/full_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 décompacté 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 installé.

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 doivent être téléchargés séparément. Le paquetage `fli4l-3.10.4.tar.gz` contient uniquement le logiciel base pour un Routeur Ethernet. On téléchargera ensuite les paquetages dont on a besoin pour DSL, ISDN ils seront extraits dans le répertoire `fli4l-3.10.4/` (!) Vous avez aussi le choix du Kernel 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. 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 sur les pages suivantes.

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

<http://www.fli4l.de/fr/telechargement/paquetage-annexe/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-3.10.4	Base, nécessaire!
kernel_3_14	Kernel 3.14, recommandé
kernel_3_14_virt	Kernel 3.14-virt, alternatif, pour une utilisation dans un environnement virtuel
kernel_3_14_nonfree	Kernel 3.14-nonfree, supporte les pilotes non-GPL
kernel_3_14_virt_nonfree	Kernel 3.14-virt-nonfree, alternatif, supporte les pilotes non-GPL
fli4l-3.10.4-doc	Documentation complète
advanced_networking	Configuration pour réseau étendue
chrony	Serveur-Time/Client
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
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
lcd	Pilote-LCD Software + Affichage du statut
lpdsrv	Serveur d'imprimante (sans Spooling)
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 pour le programme-Linux
umts	Connexion UMTS via Internet
usb	Supporte l'interface 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 !
```

3. Configuration de la base

```
##
##-----
## Creation:      26.06.2001  fm
## Last Update:  $Id: base.txt 42028 2015-10-19 05:32:49Z florian $
##
## Copyright (c) 2001-2014 - 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
KERNEL_VERSION='3.14.54'  # 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):
#-----
```

3. Configuration de la base

```
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
                                # 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_N='1'                  # number of ethernet drivers to load, usually 1
NET_DRV_1='ne2k-pci'           # 1st driver: name (e.g. NE2000 PCI clone)
NET_DRV_1_OPTION=''            # 1st driver: additional option
NET_DRV_2='ne'                  # 2nd driver: name (e.g. NE2000 ISA clone)
NET_DRV_2_OPTION='io=0x320'     # 2nd driver: additional option

#-----
# Ether networks used with IP protocol:
#-----
IP_NET_N='1'                   # number of IP ethernet networks, usually 1
IP_NET_1='192.168.6.1/24'       # IP address of your n'th ethernet card and
                                # netmask in CIDR (no. of set bits)
IP_NET_1_DEV='eth0'             # required: device name like ethX

#-----
# Additional routes, optional
#-----
IP_ROUTE_N='0'                 # number of additional routes
IP_ROUTE_1='192.168.7.0/24 192.168.6.99'
                                # network/netmaskbits gateway
```

3. Configuration de la base

```
IP_ROUTE_2='0.0.0.0/0 192.168.6.99'
# example for default-route

#-----
# Packet filter configuration
#-----

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_N='1'                # number of INPUT rules
PF_INPUT_1='IP_NET_1 ACCEPT'  # allow all hosts in the local network to
                                # access the router
PF_INPUT_2='tmp1:samba DROP NOLOG'
                                # drop (or reject) samba access
PF_INPUT_2_COMMENT='no samba traffic allowed'
                                # without logging, otherwise the log file will
                                # be filled with useless entries

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_N='2'              # number of FORWARD rules
PF_FORWARD_1='tmp1:samba DROP' # drop samba traffic if it tries to leave the
                                # subnet
PF_FORWARD_2='IP_NET_1 ACCEPT' # accept everything else

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'
```

3. Configuration de la base

```
# reject 1 udp packet per second; allow a burst
# of 5 events; otherwise drop packet
PF_OUTPUT_N='0'          # number of OUTPUT rules

PF_POSTROUTING_N='1'      # number of POSTROUTING rules
PF_POSTROUTING_1='IP_NET_1 MASQUERADE'
                          # masquerade traffic leaving the subnet

PF_PREROUTING_N='0'       # number of PREROUTING rules
PF_PREROUTING_1='1.2.3.4 dynamic:22 DNAT:@client2'
                          # forward ssh connections coming from 1.2.3.4
                          # to client2

PF_PREROUTING_CT_ACCEPT_DEF='yes'
                          # use default rule set
PF_PREROUTING_CT_N='1'    # number of conntrack PREROUTING rules
PF_PREROUTING_CT_1='tmpl:ftp IP_NET_1 HELPER:ftp'
                          # associate FTP conntrack helper for active FTP
                          # forwarded from within the LAN
PF_PREROUTING_CT_2='tmpl:ftp any dynamic HELPER:ftp'
                          # associate FTP conntrack helper for active FTP
                          # forwarded to the router's external IP

PF_OUTPUT_CT_ACCEPT_DEF='yes' # use default rule set
PF_OUTPUT_CT_N='0'          # number of conntrack OUTPUT rules
PF_OUTPUT_CT_1='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')

PF_USR_CHAIN_N='0'         # number of user-defined rules

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

# optional configuration for the host-entry of the router in /etc/hosts
#HOSTNAME_IP='IP_NET_1_IPADDR' # IP to bind to HOSTNAME
#HOSTNAME_ALIAS_N='0'          # how many ALIAS names for the router
#HOSTNAME_ALIAS_1='router.lan.fli4l'
                          # first ALIAS name
#HOSTNAME_ALIAS_2='gateway.my.lan'
                          # second ALIAS name

#-----
# imond configuration:
#-----
START_IMOND='no'          # start imond: yes or no
```

3. Configuration de la base

```
IMOND_PORT='5000'          # port (tcp), don't open it to the outside
IMOND_PASS=''              # imond-password, may be empty
IMOND_ADMIN_PASS=''        # imond-admin-password, may be empty
IMOND_LED=''               # tty for led: com1 - com4 or empty
IMOND_BEEP='no'            # beep if connection is going up/down
IMOND_LOG='no'             # log /var/log/imond.log: yes or no
IMOND_LOGDIR='auto'        # log-directory, e.g. /var/log or auto for
                           # saving in auto-detected savedir
IMOND_ENABLE='yes'         # accept "enable/disable" command
IMOND_DIAL='yes'           # accept "dial/hangup" command
IMOND_ROUTE='yes'          # accept "route" command
IMOND_REBOOT='yes'         # accept "reboot" command

#-----
# Generic circuit configuration:
#-----
IP_DYN_ADDR='yes'          # use dyn. IP addresses (most providers do)
DIALMODE='auto'           # standard dialmode: auto, manual, or off

#-----
# optional package: syslogd
#-----
#OPT_SYSLOGD='no'          # start syslogd: yes or no
#SYSLOGD_RECEIVER='yes'    # receive messages from network
SYSLOGD_DEST_N='1'         # number of destinations
SYSLOGD_DEST_1='*.* /dev/console'
                           # n'th prio & destination of syslog msgs
SYSLOGD_DEST_2='*.* @192.168.6.2'
                           # example: loghost 192.168.6.2
SYSLOGD_DEST_3='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
Y2K_DAYS='0'               # correct hardware y2k-bug: add x days
```

3. Configuration de la base

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

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'`

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 118) du paquetage `hd`.

3. Configuration de la base

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 intrégrée 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 intrégrée 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>.

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

3. Configuration de la base

courante aurait un problème. Celle-ci peut être activée dans le menu boot avec le choix de la version Recover.

Si vous mettez la valeur '0', le système attend que l'utilisateur active Le chargement du Bootloader de syslinux standard ou la Version Recover 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 `fi4l` avec la norme UTC (Coordinated Universal Time), une heure unique dans le monde entier et qui sera convertit pour chaque localité. `TIME_INFO` donne les informations nécessaires à `fi4l` sur les noms des fuseaux horaires 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 indications suivantes :

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

- *MEZ-1* : fuseau horaire de 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 en été.
- *M3.5.0,M10.5.0/3* : date des changements d'horaires d'été et d'hiver, le dernier dimanche d'octobre 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

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/files/lib/modules/<kernel version>`. Il est possible d'utiliser des machines virtuelles, pour cela vous devez prendre le Kernel avec l'extension `-virt`. Cependant, cette version nécessite au moins un processeur Pentium avec prise en charge du PAE.

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'indiquer `'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 ces variables 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 ces variables détermine la méthode de compression pour l'archive OPT. Les valeurs possibles sont `'xz'`, `'lzma'` et `'bzip2'`.

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

3. Configuration de la base

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 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
```

3. Configuration de la base

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, 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

```
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

CONSOLE_BLANK_TIME Configuration par défaut : `CONSOLE_BLANK_TIME=""`

Lorsqu'on n'utilise 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 on place '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 on ne veut pas que le signal sonore soit émit, on peut indiquer ici 'no'.

3. Configuration de la base

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

fli4l peut être utilisé sans clavier et sans carte vidéo. Il est possible voir les messages de boot du kernel sans accéder au routeur par le Telnet ou le SSH, pour cela vous devez rediriger les entrées et les sorties vers un port série. Il faut paramétrer les variables suivante `SER_CONSOLE`, `SER_CONSOLE_IF` et `SER_CONSOLE_RATE`.

La console série peut être configurée en trois modes différents :

SER_CONSOLE	Entrée/Sortie sur la console
no	Entrée/Sortie sur tty0
yes	Entrée/Sortie sur la console série
primary	Entrée/Sortie via la console série, vous pouvez aussi voir les messages du kernel sur la sortie tty0
secondary	Entrée/Sortie sur tty0, vous pouvez aussi voir les messages du kernel sur la console série

Si la valeur `SER_CONSOLE` est modifiée, cette modification ne prendra effet lors de l'installation d'un nouveau média de boot ou d'une mise à jour de `syslinux.cfg`.

Faites attention en coupant le port série vous maintenez un accès au routeur (par le ssh ou directement avec le clavier).

Vous trouverais des informations complémentaires en cliquant sur [Console serie](#) (Page 344).

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

Numéro de l'interface série du port de sortie.

Ici le numéro de l'interface sur laquelle le port série est branché. '0' correspond au ttyS0 ou au COM1 sous DOS.

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

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

Ici on inscrit 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

fli4l 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 on besoin d'autre 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 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'.

3. Configuration de la base

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 248) 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 d'Init. 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* < 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 :

```
::sysinit:/etc/rc
::respawn:cttyhack /usr/local/bin/mini-login
```

3. Configuration de la base

```
::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 `opt/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'.

Avec l'utilisation de DSL, on installe souvent deux cartes réseau.

il faut distingue deux cas :

3. Configuration de la base

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 NET_DRV_x_OPTION 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` par défaut le pilote NE2000=carte réseau compatible est chargée, à modifier selon votre configuration. La famille des cartes réseaux sont enregistrées les tableaux 3.3 et 3.4.

Pour la carte 3COM EtherLinkIII (3c509) vous devez l'outil-DOS, 3c509cfg.exe (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 l'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, avec par exemple 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 sur tous les pilotes de carte réseau. Sur quelques une vous devez double le chargement du pilote, alors `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 :

– 1 x NE2000 ISA

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

3. Configuration de la base

- 1 x 3COM EtherLinkIII (3c509)

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

Voir aussi les faq sur les cartes (en Allemand) :

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

[//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=132&catnr=7&prog=1)

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

[//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=133&catnr=7&prog=1)

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

[//extern.fli4l.de/fli4l_faqengine/faq.php?display=faq&faqnr=135&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 possibles à installer dans le [Tableau des pilotes LAN](#) et dans le [Tableau des pilotes WLAN](#).

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 39)='dummy<Nombre>' pour le nom du périphérique.

Kernel 3.14				Bus	NET_DRV_x	Nom de la carte
v	n	vn				
x	x	x	x	isa	3c509	3Com Etherlink III (3c509, 3c509B, 3c529, 3c579) ethernet
x	x	x	x	isa	3c515	3Com 3c515 Corkscrew
x	x	x	x	pcmcia	3c574_cs	3Com 3c574 series PCMCIA ethernet
x	x	x	x	pcmcia	3c589_cs	3Com 3c589 series PCMCIA ethernet
x	x	x	x	pci	3c59x	3Com 3c59x/3c9xx ethernet
x	x	x	x	pci	8139cp	RealTek RTL-8139C+ series 10/100 PCI Ethernet
x	x	x	x	pci	8139too	RealTek RTL-8139 Fast Ethernet

3. Configuration de la base

Kernel 3.14				Bus	NET_DRV_x	Nom de la carte
v	n	vn				
x	x	x	x	pci	acenic	AceNIC/3C985/GA620 Gigabit Ethernet
x	x	x	x	pci	alx	Qualcomm Atheros(R) AR816x/AR817x PCI-E Ethernet Network
x	x	x	x	pci	amd8111e	AMD8111 based 10/100 Ethernet Controller
x	x	x	x	usb	asix	ASIX AX8817X based USB 2.0 Ethernet Devices
x	x	x	x	pci	atl1	Atheros L1 Gigabit Ethernet
x	x	x	x	pci	atl1c	Qualcom Atheros 100/1000M Ethernet Network
x	x	x	x	pci	atl1e	Atheros 1000M Ethernet Network
x	x	x	x	pci	atl2	Atheros Fast Ethernet Network
x	x	x	x	isa	atp	RealTek RTL8002/8012 parallel port Ethernet
x	x	x	x	usb	ax88179_178a	ASIX AX88179/178A based USB 3.0/2.0 Gigabit Ethernet Devices
x	x	x	x	pcmcia	axnet_cs	Asix AX88190 PCMCIA ethernet
x	x	x	x	pci	b44	Broadcom 44xx/47xx 10/100 PCI ethernet
x	x	x	x	pci	be2net	Emulex OneConnect NIC Driver 10.0.600.0u
x	x	x	x	pci	bna	Brocade 10G PCIe Ethernet
x	x	x	x	pci	bnx2	Broadcom NetXtreme II BCM5706/5708/5709/5716
x	x	x	x	pci	bnx2x	Broadcom NetXtreme II BCM57710/ 57711/ 57711E/ 57712/ 57712_MF/ 57800/ 57800_MF/ 57810/ 57810_MF/ 57840/ 57840_MF
x	x	x	x	pci	cassini	Sun Cassini(+) ethernet
x	x	x	x	usb	catc	CATC EL1210A NetMate USB Ethernet
x	x	x	x	usb	cdc_eem	USB CDC EEM
x	x	x	x	usb	cdc_ether	USB CDC Ethernet devices
x	x	x	x	usb	cdc_mbim	USB CDC MBIM host
x	x	x	x	usb	cdc_ncm	USB CDC NCM host
x	x	x	x	usb	cdc_subset	Simple 'CDC Subset' USB networking links
x	x	x	x	usb	cx82310_eth	Conexant CX82310-based ADSL router USB ethernet
x	x	x	x	pci	cxgb	Chelsio 10Gb Ethernet
x	x	x	x	pci	cxgb3	Chelsio T3 Network
x	x	x	x	pci	cxgb4	Chelsio T4/T5 Network
x	x	x	x	pci	cxgb4vf	Chelsio T4/T5 Virtual Function (VF) Network
x	x	x	x	pci	de2104x	Intel/Digital 21040/1 series PCI Ethernet
x	x	x	x	isa	de4x5	Digital DE425, DE434, DE435, DE450, DE500
x	x	x	x	pci	defxx	DEC FDDIcontroller TC/EISA/PCI (DEFTA/DEFEA/DEFP) driver v1.10 2006/12/14

3. Configuration de la base

Kernel 3.14				Bus	NET_DRV_x	Nom de la carte
v	n	vn				
x	x	x	x	pci	dl2k	D-Link DL2000-based Gigabit Ethernet Adapter
x	x	x	x	usb	dm9601	Davicom DM96xx USB 10/100 ethernet devices
x	x	x	x	pci	dmfe	Davicom DM910X fast ethernet
x	x	x	x	virtual	dummy	Dummy Network Interface
x	x	x	x	pci	e100	Intel(R) PRO/100 Network
x	x	x	x	pci	e1000	Intel(R) PRO/1000 Network
x	x	x	x	pci	e1000e	Intel(R) PRO/1000 Network
x	x	x	x	pci	enic	Cisco VIC Ethernet NIC
x	x	x	x	pci	epic100	SMC 83c170 EPIC series Ethernet
x	x	x	x	pci	fealnx	Myson MTD-8xx 100/10M Ethernet PCI Adapter
x	x	x	x	pcmcia	fmvj18x_cs	fmvj18x and compatible PCMCIA ethernet
x	x	x	x	pci	forcedeth	Reverse Engineered nForce ethernet
x	x	x	x	usb	gl620a	GL620-USB-A Host-to-Host Link cables
x	x	x	x	pci	hamachi	Packet Engines 'Hamachi' GNIC-II Gigabit Ethernet
x	x	x	x	pci	hp100	HP CASCADE Architecture Driver for 100VG-AnyLan Network Adapters
x	x	x	x	usb	hso	USB High Speed Option
x	x	x	x	usb	huawei_cdc_ncm	USB CDC NCM host driver with encapsulated protocol support
x	x	x	x	pci	i40e	Intel(R) Ethernet Connection XL710 Network
x	x	x	x	pci	i40evf	Intel(R) XL710 X710 Virtual Function Network
x	x	x	x	pci	igb	Intel(R) Gigabit Ethernet Network
x	x	x	x	pci	igbvf	Intel(R) Gigabit Virtual Function Network
x	x	x	x	usb	int51x1	Intellon usb powerline adapter
x	x	x	x	pci	ipg	IC Plus IP1000 Gigabit Ethernet Adapter Linux
x	x	x	x	usb	ipheth	Apple iPhone USB Ethernet
x	x	x	x	pci	ixgb	Intel(R) PRO/10GbE Network
x	x	x	x	pci	ixgbe	Intel(R) 10 Gigabit PCI Express Network
x	x	x	x	pci	ixgbev	Intel(R) 82599 Virtual Function
x	x	x	x	pci	jme	JMicron JMC2x0 PCI Express Ethernet
x	x	x	x	usb	kalmia	Samsung Kalmia USB network
x	x	x	x	usb	kaweth	KL5USB101 USB Ethernet
x	x	x	x	pci	ksz884x	KSZ8841/2 PCI network
x	x	x	x	isa	lance	AMD LANCE and PCnet (AT1500, NE2100)
x	x	x	x	usb	lg-vl600	LG-VL600 modem's ethernet link
x	x	x	x	usb	mcs7830	USB to network adapter (MCS7830)
x	x	x	x	pci	mlx4_core	Mellanox ConnectX HCA low-level
x	x	x	x	pci	myri10ge	Myricom 10G driver (10GbE)

3. Configuration de la base

Kernel 3.14				Bus	NET_DRV_x	Nom de la carte
v	n	vn				
x	x	x	x	pci	natsemi	National Semiconductor DP8381x series PCI Ethernet
x	x	x	x	isa	ne	NE1000/NE2000 ISA/PnP Ethernet
x	x	x	x	pci	ne2k-pci	PCI NE2000 clone
x	x	x	x	usb	net1080	NetChip 1080 based USB Host-to-Host Links
x	x	x	x	pci	netxen_nic	QLogic/NetXen (1/10) GbE Intelligent Ethernet
x	x	x	x	isa	ni65	AMD Lance Am7990
x	x	x	x	pci	niu	Sun Neptun Ethernet
x	x	x	x	pcmcia	nmclan_cs	New Media PCMCIA ethernet
x	x	x	x	pci	ns83820	National Semiconductor DP83820 10/100/1000
x	x	x	x	pci	pch_gbe	EG20T PCH Gigabit ethernet
x	x	x	x	pci	pcnet32	PCnet32 and PCnetPCI based ethercards
x	x	x	x	pcmcia	pcnet_cs	NE2000 compatible PCMCIA ethernet
x	x	x	x	usb	pegasus	Pegasus/Pegasus II USB Ethernet
x	x	x	x	usb	plusb	Prolific PL-2301/2302/25A1 USB Host to Host Link
x	x	x	x	pci	qla3xxx	QLogic ISP3XXX Network Driver v2.03.00-k5
x	x	x	x	pci	qlcnlc	QLogic 1/10 GbE Converged/Intelligent Ethernet
x	x	x	x	pci	qlge	QLogic 10 Gigabit PCI-E Ethernet
x	x	x	x	usb	qmi_wwan	Qualcomm MSM Interface (QMI) WWAN
x	x	x	x	pci	r6040	RDC R6040 NAPI PCI FastEthernet
x	x	x	x	usb	r8152	Realtek RTL8152/RTL8153 Based USB Ethernet Adapters
x	x	x	x	pci	r8169	RealTek RTL-8169 Gigabit Ethernet
x	x	x	x	usb	rndis_host	USB Host side RNDIS
x	x	x	x	usb	rtl8150	rtl8150 based usb-ethernet
x	x	x	x	pci	s2io	Neterion 10GbE Server NIC
x	x	x	x	isa	sb1000	General Instruments SB1000
x	x	x	x	pci	sc92031	Silan SC92031 PCI Fast Ethernet Adapter
x	x	x	x	pci	sfc	Solarflare Communications network
x	x	x	x	pci	sis190	SiS sis190/191 Gigabit Ethernet
x	x	x	x	pci	sis900	SiS 900 PCI Fast Ethernet
x	x	x	x	pci	skfp	SysKonnect FDDI PCI adapter
x	x	x	x	pci	skge	SysKonnect Gigabit Ethernet
x	x	x	x	pci	sky2	Marvell Yukon 2 Gigabit Ethernet
x	x	x	x	isa	smc-ultra	SMC Ultra/EtherEZ ISA/PnP Ethernet
x	x	x	x	isa	smc9194	SMC's 9000 series of Ethernet cards
x	x	x	x	pcmcia	smc91c92_cs	SMC 91c92 series PCMCIA ethernet

3. Configuration de la base

Kernel 3.14				Bus	NET_DRV_x	Nom de la carte
v	n	vn				
x	x	x	x	usb	smc75xx	SMSC75XX USB 2.0 Gigabit Ethernet Devices
x	x	x	x	pci	smc9420	SMSC LAN9420
x	x	x	x	usb	smc95xx	SMSC95XX USB 2.0 Ethernet Devices
x	x	x	x	usb	sr9700	SR9700 one chip USB 1.1 USB to Ethernet device from http://www.corechip-sz.com/
x	x	x	x	usb	sr9800	SR9800 USB 2.0 USB2NET Dev : http://www.corechip-sz.com
x	x	x	x	pci	starfire	Adaptec Starfire Ethernet
x	x	x	x	pci	stmmac	STMMAC 10/100/1000 Ethernet device
x	x	x	x	pci	sundance	Sundance Alta Ethernet
x	x	x	x	pci	sungem	Sun GEM Gbit ethernet
x	x	x	x	pci	sunhme	Sun HappyMealEthernet(HME) 10/100baseT ethernet
x	x	x	x	pci	tehuti	Tehuti Networks(R) Network
x	x	x	x	pci	tg3	Broadcom Tigon3 ethernet
x	x	x	x	pci	tlan	TI ThunderLAN based ethernet PCI adapters
x	x	x	x	pci	tulip	Digital 21*4* Tulip ethernet
x	x	x	x	pci	typhoon	3Com Typhoon Family (3C990, 3CR990, and variants)
x	x	x	x	pci	uli526x	ULi M5261/M5263 fast ethernet
x	x	x	x	pci	via-rhine	VIA Rhine PCI Fast Ethernet
x	x	x	x	pci	via-velocity	VIA Networking Velocity Family Gigabit Ethernet Adapter
		x	x	virtio	virtio_net	Virtio network
		x	x	pci	vmxnet3	VMware vmxnet3 virtual NIC
x	x	x	x	pci	vxge	Neterion's X3100 Series 10GbE PCIe I/OVirtualized Server Adapter
x	x	x	x	isa	wd	Western Digital wd8003/wd8013; SMC Elite, Elite16 ethernet
x	x	x	x	pci	winbond-840	Winbond W89c840 Ethernet
		x	x	xen	xen-netfront	Xen virtual network device frontend
x	x	x	x	pcmcia	xirc2ps_cs	Xircom PCMCIA ethernet
x	x	x	x	pci	xircom_cb	Xircom Cardbus ethernet
x	x	x	x	pci	yellowfin	Packet Engines Yellowfin G-NIC Gigabit Ethernet
x	x	x	x	usb	zaurus	Sharp Zaurus PDA, and compatible products

TABLE 3.3.: Tableau des pilotes LAN ; légende : v=virt, n=nonfree, vn=virt-nonfree

3. Configuration de la base

3.14				Kernel		
	v	n	vn		v	n
x	x	x	x	pci	adm8211	IEEE 802.11b wireless cards based on ADMtek ADM8211
x	x	x	x	isa,pci	airo	Cisco/Aironet 802.11 wireless ethernet cards
x	x	x	x	pcmcia	airo_cs	Cisco/Aironet 802.11 wireless ethernet cards
x	x	x	x	usb	ar5523	Atheros AR5523 based USB dongles
x	x	x	x	usb	at76c50x-usb	Atmel at76x USB Wireless LAN
x	x	x	x	pci	ath10k_pci	Driver support for Atheros QCA988X PCIe devices
x	x	x	x	pci	ath5k	5xxx series of Atheros 802.11 wireless LAN cards
x	x	x	x	usb	ath6kl_usb	Driver support for Atheros AR600x USB devices
x	x	x	x	pci	ath9k	Atheros 802.11n wireless LAN cards
x	x	x	x	usb	ath9k_htc	Atheros driver 802.11n HTC based wireless devices
x	x	x	x	pcmcia	atmel_cs	Atmel at76c50x 802.11 wireless ethernet cards
x	x	x	x	pci	atmel_pci	Atmel at76c50x 802.11 wireless ethernet cards
x	x	x	x	pci,pcmcia	b43	Broadcom B43 wireless
x	x	x	x	pci	b43legacy	Broadcom B43legacy wireless
x	x	x	x	usb	brcmfmac	Broadcom 802.11 wireless LAN fullmac
x	x	x	x	pci	brcmsmac	Broadcom 802.11n wireless LAN
x	x	x	x	usb	carl9170	Atheros AR9170 802.11n USB wireless
x	x	x	x	pcmcia	hostap_cs	Intersil Prism2-based 802.11 wireless LAN cards (PC Card)
x	x	x	x	pci	hostap_pci	Intersil Prism2.5-based 802.11 wireless LAN PCI cards
x	x	x	x	pci	hostap_plx	Intersil Prism2-based 802.11 wireless LAN cards (PLX)
x	x	x	x	pci	ipw2100	Intel(R) PRO/Wireless 2100 Network
x	x	x	x	pci	ipw2200	Intel(R) PRO/Wireless 2200/2915 Network
x	x	x	x	pci	iwl3945	Intel(R) PRO/Wireless 3945ABG/BG Network Connection driver for Linux
x	x	x	x	pci	iwl4965	Intel(R) Wireless WiFi 4965 driver for Linux
x	x	x	x	pci	iwlwifi	Intel(R) Wireless WiFi driver for Linux
x	x	x	x	pcmcia	libertas_cs	Marvell 83xx compact flash WLAN cards
x	x	x	x	usb	libertas_tf_usb	8388 USB WLAN Thinfirm
x	x	x	x	virtual	mac80211_hwsim	Software simulator of 802.11 radio(s) for mac80211
x	x	x	x	pci	mwifiex_pcie	Marvell WiFi-Ex PCI-Express Driver version 1.0
x	x	x	x	usb	mwifiex_usb	Marvell WiFi-Ex USB Driver version1.0
x	x	x	x	pci	mwl8k	Marvell TOPDOG(R) 802.11 Wireless Network
x	x	x	x	pcmcia	orinoco_cs	PCMCIA Lucent Orinoco, Prism II based and similar wireless cards
x	x	x	x	pci	orinoco_nortel	wireless LAN cards using the Nortel PCI bridge
x	x	x	x	pci	orinoco_plx	wireless LAN cards using the PLX9052 PCI bridge
x	x	x	x	pci	orinoco_tmd	wireless LAN cards using the TMD7160 PCI bridge
x	x	x	x	usb	orinoco_usb	Orinoco wireless LAN cards using EZUSB bridge
x	x	x	x	pci	p54pci	Prism54 PCI wireless
x	x	x	x	usb	p54usb	Prism54 USB wireless
x	x	x	x	pcmcia	ray_cs	Raylink/WebGear wireless LAN
x	x	x	x	usb	rndis_wlan	RNDIS based USB Wireless adapters
x	x	x	x	pci	rt2400pci	Ralink RT2400 PCI & PCMCIA Wireless LAN
x	x	x	x	pci	rt2500pci	Ralink RT2500 PCI & PCMCIA Wireless LAN
x	x	x	x	usb	rt2500usb	Ralink RT2500 USB Wireless LAN
x	x	x	x	pci	rt2800pci	Ralink RT2800 PCI & PCMCIA Wireless LAN
x	x	x	x	usb	rt2800usb	Ralink RT2800 USB Wireless LAN
x	x	x	x	pci	rt61pci	Ralink RT61 PCI & PCMCIA Wireless LAN
x	x	x	x	usb	rt73usb	Ralink RT73 USB Wireless LAN
x	x	x	x	pci	rtl8180	RTL8180 / RTL8185 PCI wireless
x	x	x	x	usb	rtl8187	RTL8187/RTL8187B USB wireless
x	x	x	x	pci	rtl8188ee	Realtek 8188E 802.11n PCI wireless
x	x	x	x	pci	rtl8192ce	Realtek 8192C/8188C 802.11n PCI wireless
x	x	x	x	usb	rtl8192cu	Realtek 8192C/8188C 802.11n USB wireless
x	x	x	x	pci	rtl8192de	Realtek 8192DE 802.11n Dual Mac PCI wireless
x	x	x	x	pci	rtl8192se	Realtek 8192S/8191S 802.11n PCI wireless
x	x	x	x	pci	rtl8723ae	Realtek 8723E 802.11n PCI wireless

3. Configuration de la base

3.14				Kernel		
	v	n	vn		v	n
x	x	x	x	usb	sierra_net	USB-to-WWAN Driver for Sierra Wireless modems
x	x	x	x	pcmcia	spectrum_cs	Symbol Spectrum24 Trilogy cards with firmware downloader
x	x	x	x	usb	usb8xxx	8388 USB WLAN
x	x	x	x	pci	wil6210	60g WiFi WIL6210 card
x	x	x	x	pcmcia	wl3501_cs	Planet wl3501 wireless
x	x	x	x	usb	zd1201	ZyDAS ZD1201 based USB Wireless adapters
x	x	x	x	usb	zd1211rw	USB driver for devices with the ZD1211 chip

TABLE 3.4.: Tableau des pilotes WLAN ; légende : v=virt, n=nonfree, vn=virt-non

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'`

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 indiquer 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

1. Classless inter-domaine Routing

3. Configuration de la base

suivant : `Nombre_Hôtes = 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 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.

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 '`fli4l-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 paquetages comme par exemple `opt-rrdtool`.

3.9. 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 38) 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'
```

3.10. Le filtrage de paquets

Le Kernel de Linux utilisé par fli4l met à disposition un filtrage de paquets. À l'aide de ce filtre 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 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

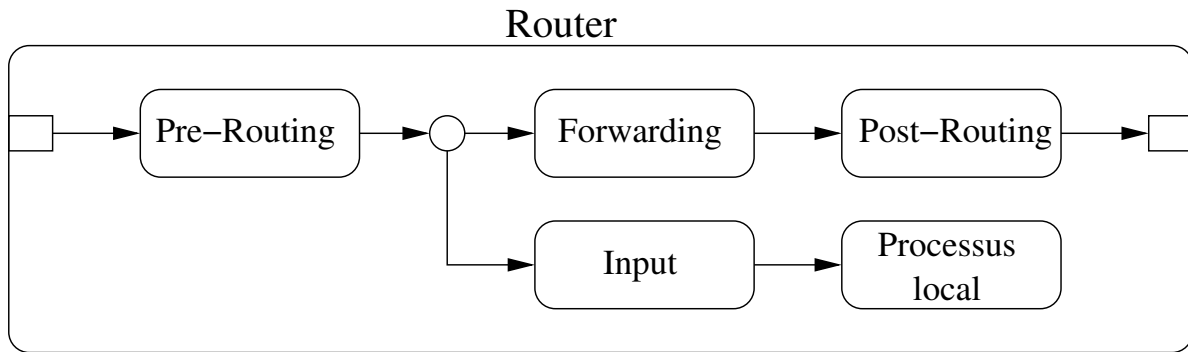


FIGURE 3.1.: Structure du Filtrage de paquets

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.

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, 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 concernent 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 activée, 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).

3.10.1. 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 LOG :log-prefix. 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 remplacée 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 SNAT 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 DNAT spécifier l'adresse (considéré que tous les paquets appartiennent à cette connexion).
REDIRECT	PREROUTING OUTPUT	Le port de destination du paquet sera remplacé, 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).

Action	Chaîne(n)	Importance
--------	-----------	------------

TABLE 3.5.: 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.

3.10.2. 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.² 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 <code><ip>/<netmask></code>
port[-port]	Port ou une plage de ports
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 <code>HOST_%_*</code> 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 3.6.: Restrictions de la source et de destination dans les règles de filtrage de paquets

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

3. Configuration de la base

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 :

<code>127.0.0.1 ACCEPT</code>	La communication locale (Source 127.0.0.1) est acceptée
<code>any 192.168.12.1 DROP</code>	Les paquets vers l'adresse-IP 192.168.12.1 sont rejetés
<code>any 192.168.12.1 DROP LOG</code>	Les paquets vers l'adresse-IP 192.168.12.1 sont rejetés et sont également enregistrés
<code>any 192.168.12.1 DROP NOLOG</code>	Les paquets vers l'adresse-IP 192.168.12.1 sont rejetés, ne sont pas enregistrés
<code>22 ACCEPT</code>	Les paquets vers le port 22 (<code>ssh</code>) sont acceptés
<code>IP_NET_1_NET ACCEPT</code>	Les paquets du sous-réseau de la première interface sont acceptés
<code>IP_NET_1_NET IP_NET_2_NET</code>	La communication entre la première et la seconde
<code>ACCEPT BIDIRECTIONAL</code>	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 chaîne `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 : ³

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.
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 <code>ftp</code> pour le transfère de données).

TABLE 3.7.: 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`).

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

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.

3.10.3. 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 3.8).

Modèle	Protocole	Port(s)
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

3. Configuration de la base

Modèle	Protocole	Port(s)
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
ipmi	udp	623
irc	tcp	6667
ldap	tcp/udp	389
mail	tcp	110
mail	tcp	143
mail	tcp	25
mail	tcp	465
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

3. Configuration de la base

Modèle	Protocole	Port(s)
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
xmppserver	tcp	5269

TABLE 3.8.: 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>** permisent sont décrites dans dans la section 3.10.2. Les valeurs possibles pour les **<actions souhaitées>** sont énumérées et décrites dans la section 3.10.1

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.

3. Configuration de la base

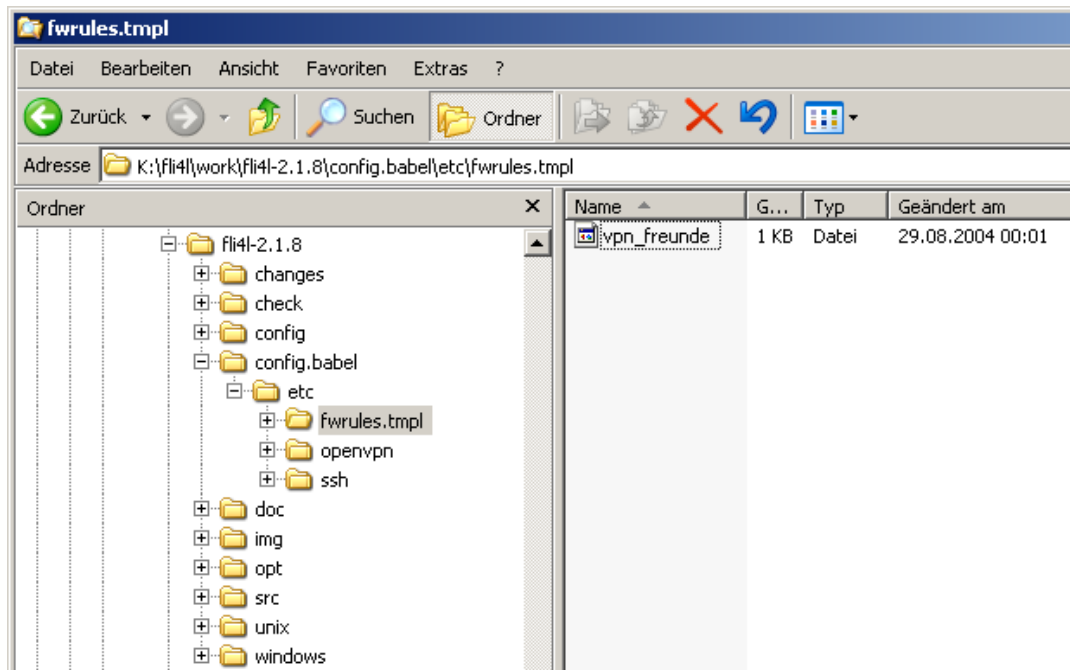


FIGURE 3.2.: Structure du répertoire fli4l

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 paquetages 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 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 en-

3. Configuration de la base

registrez 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
```

3.10.4. Configuration du filtrage de paquets

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

- `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 entre 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 54)) :

```
PF_INPUT_ACCEPT_DEF='no'
#
# limit ICMP echo requests - use a separate chain
#
PF_USR_CHAIN_N='1'
```

3. Configuration de la base

```
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 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 activez 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 :

- 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', flitl 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 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_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 :

- 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 nécessaires 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 activez 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 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_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 envoyer 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'
#
# 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 chaîne `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.

3. Configuration de la base

- 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 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 ajoutée 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 arrivent 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 règle 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 enregistrée 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.

REDIRECT se comporte comme DNAT, sauf que l'adresse IP cible est toujours mise sur 127.0.0.1 et donc le paquet sera livré localement. C'est par ex. nécessaire pour un proxy transparent, voir [OPT_TRANSPROXY](#) (Page 193).

Si vous voulez faire de la redirection de port sur des interfaces qui utilisent 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 assigner 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 chaîne PREROUTING peut utiliser les actions suivantes ACCEPT, DNAT, NETMAP et REDIRECT.

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

3.10.5. 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='tmp1: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 :

3. Configuration de la base

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

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

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 `desttcp` Ports 139 et 445, la configuration se présentera comme ceci :

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

3. Configuration de la base

```
PF_FORWARD_N='4'
PF_FORWARD_1='IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_FORWARD_2='tmpl:samba DROP'
PF_FORWARD_3='192.168.6.0/23 ACCEPT'
PF_FORWARD_4='10.0.0.0/24 ACCEPT'

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

- 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='tmpl: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.

3. Configuration de la base

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

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

3.10.6. 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'
```

3. Configuration de la base

```
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
                                   # 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
```

3. Configuration de la base

```
# du sous-réseau d'avoir un accès
# spécifique au service HTTP

#
# 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 :

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 66)).

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
```

3. Configuration de la base

```
# 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)

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 46)). 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 ...

3.10.7. DMZ – Zone démilitarisée

`fli4l` 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.

3.10.8. 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.

3. Configuration de la base

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.

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 :

3. Configuration de la base

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

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 **fli4l** (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
ftp	File Transfer Protocol
h323	H.323 (Voice over IP)
irc	Internet Relay Chat
pptp	PPTP Masquerading (Ce module peut être plus qu'un client PPTP il fonctionne simultanément derrière un routeur fli4l.)
sip	Session Initiation Protocol
sane	SANE Network Protocol
snmp	Simple Network Management Protocol
tftp	Trivial File Transfer Protocol

TABLE 3.9.: 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.

3.11. 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 92) dans le (paquetage `dns_dhcp` (Page 90)), cependant, cette variable ne doit jamais être substituée à la variable `*_USEPEERDNS`.

Attention :

- `PPPOE_USEPEERDNS` (Page 103),
- `ISDN_CIRC_x_USEPEERDNS` (Page 151) ou
- `DHCPCLIENT_x_USEPEERDNS` (Page 90)

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.

3.12. Configuration de Imond

START_IMOND Configuration par défaut : START_IMOND='no'

En mettant START_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'imond](#) (Page 269).

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'imond. Pour l'activer, vous paramétrez START_IMOND='yes'.

Si fli4l est uniquement utilisé comme routeur avec deux cartes réseaux, vous devez paramétrer la variable sur START_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 68),
- [IMOND_DIAL](#) (Page 68),
- [IMOND_ROUTE](#) (Page 68) et
- [IMOND_REBOOT](#) (Page 68)

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 3.10.

Les frais sont donnés en Euro. Il est important pour cette fonction que la variable **ISDN_CIRC_x_TIMES** (Page 157) 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 3.10.: 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 269) pour le client et le serveur.

3.13. 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 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 de base

Dans le paquetage de base vous trouverez les OPTs suivants :

Nom	Description
OPT_SYSLOGD	Programme qui enregistre tous les messages (Page 70)
OPT_KLOGD	Programme qui enregistre les messages Kernel (Page 72)
OPT_LOGIP	Programme qui enregistre les protocoles WAN-IP (Page 72)
OPT_Y2K	Correctif pour les ordinateurs avant l'année 2K (Page 72)
OPT_PNP	Outil pour l'installation des cartes ISAPnP (Page 73)

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 156) et [PPPOE_DEBUG](#) (Page 103).

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 27)).

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 `fl4l` peut rechercher la date et l'heure exacte sur Internet. Pour cela vous avez le paquetage `CHRONY` (Page 87) 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-`fl4l` :

```
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 seule 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'

4.2. Advanced Networking

Avec le paquetage advanced_networking il est possible d'élargir les fonctionnalités du routeur fli4l en utilisant une liaison VLAN (ou réseau local virtuel), la fonction bridge (ou pont), et la fonction bonding. Il supporte aussi EBTables que l'on peut activer, ainsi il sera possible de mettre un filtre transparent dans les paquets voir (<http://ebtables.sourceforge.net/>).

En règle générale le paquetage advanced_networking peut s'installer avec tous les autres paquetages :

Ce logiciel a été pensé uniquement pour les utilisateurs qui ont une bonne connaissance des réseaux. En particulier il est nécessaire d'avoir des connaissances solides sur le routage.

En activant EBTables vous pouvez rencontrer des problèmes très inhabituelles, si vous ne connaissez pas à 100% les différents impacts des couches 2 et 3. Il faut activer EBTables pour travailler avec certaines règles de filtrages de paquets, qui est totalement différent par rapport à ce que l'on a vu.

4.2.1. Relais broadcast - Transmission par IP broadcast

On peut utiliser d'un relais broadcast pour une transmission par IP broadcast via une autre interface. Ce dispositif est nécessaire pour certaines applications, il utilise le broadcast pour la transmission sur le réseau (par exemple avec l'utilitaire QNAP Finder), en généralement le broadcast ne peut pas transmettre sur le réseau à travers un routeur. Si vous utilisez un relais broadcast ce problème peut être contourné.

Un relais broadcast diffuse toujours les paquets broadcast à toutes les interfaces connectées. Cela signifie qu'il n'est pas nécessaire de configurer un autre relais broadcast pour les échanges entre les interfaces. En outre, il n'est pas souhaitable d'inclure plusieurs relais broadcast pour une même interface.

OPT_BCRELAY Transmission par broadcast

Par défaut : `OPT_BCRELAY='no'`

Si vous indiquez `'yes'` dans cette variable le relais broadcast sera activé. Si vous indiquez `'no'` vous désactivez complètement le paquetage du relais broadcast.

BCRELAY_N Par défaut : `BCRELAY_N='0'`

Dans cette variable vous indiquez le nombre de relais broadcast à configurer.

BCRELAY_x_IF_N Par défaut : `BCRELAY_x_IF_N='1'`

Dans cette variable vous indiquez le nombre d'interfaces qui sont affectés au relais broadcast.

BCRELAY_x_IF_x Par défaut : `BCRELAY_x_IF_x=""`

Dans cette variable vous indiquez le nom de l'interface qui sera affectés au relais broadcast

Pour plus de précisions voici un exemple avec un ordinateur sur le réseau interne (connecté sur `eth0`) dans lequel un utilitaire (par exemple QNAP Finder) est installé, le NAS est dans un autre réseau (connecté sur `eth1`).

```
OPT_BCRELAY='yes'
BCRELAY_N='1'
BCRELAY_1_IF_N='2'
BCRELAY_1_IF_1='eth0'
BCRELAY_1_IF_2='eth1'
```

4.2.2. Bonding - Regroupées plusieurs cartes réseaux pour avoir un seul lien

On entend par bonding le regroupement d'au moins deux cartes réseau, qui peuvent être également de différents types (c'est-à-dire 3Com et Intel) et de vitesse (10 Mbit/s ou 100 Mbit/s) pour avoir une seule connexion. vous pouvez raccorder soit directement à un ordinateur linux, ou soit à un switch. ainsi par ex. du routeur fli4l au switch vous serez connecté à 200 Mbit/s en Full-Duplex sans grande difficulté. Toute personne qui s'intéresse au bonding doivent lire la documentation (`bonding.txt`) dans le dossier kernel. Les noms des variables bonding utilisées sont dans la mesure du possible similaires. Sous le kernel 2.6.x de Linux le fichier `bonding.txt` se trouve dans le répertoire `Documentation/networking` du kernel source.

OPT_BONDING_DEV Par défaut : `OPT_BONDING_DEV='no'`

Avec `'yes'` vous activez le paquetage bonding. Si vous indiquez `'no'` vous désactivez complètement le paquetage bonding.

BONDING_DEV_N Par défaut : `BONDING_DEV_N='0'`

Nombre de périphérique à configurer.

BONDING_DEV_x_DEVNAME Par défaut : `BONDING_DEV_x_DEVNAME=""`

Vous devez indiquer ici un nom pour l'unité de liaison. Le nom doit commencer obligatoirement par 'bond' et doit être suivi par un nombre '0'. Les numéros des équipements bondings ne doivent pas commencer par un '0' et se suivre. Voici par ex. quelques noms 'bond0', 'bond8' ou 'bond99'.

BONDING_DEV_x_MODE Par défaut : `BONDING_DEV_x_MODE=""`

Il existe plusieurs méthodes de bonding. Le mode par défaut est Round-Robin (l'équilibrage de charge) 'balance-rr'. Les méthodes possibles sont les suivantes :

balance-rr Round-Robin (équilibrage de charge) : les données sont transmises séquentiellement de la première à la dernière interface. Ce mode permet à la fois l'équilibrage de charge et la tolérance de pannes.

active-backup Active Backup (Sauvegarde active) : seul une interface active est réellement utilisé. En cas de panne, l'interface active suivante prend la relève. L'adresse MAC du bond est visible uniquement sur un port (adaptateur réseau), pour ne pas embrouiller le switch. Ce mode permet la tolérance de pannes.

balance-xor XOR-Methode (mode XOR) : une interface est affectée à l'envoi vers une même adresse MAC. Ainsi les transferts sont parallélisés et le choix de l'interface suit les règles [(Adresse-MAC-source XOR Adresse-MAC-destination) modulo nombre d'interfaces]. Ce mode offre à la fois d'équilibrage de charge (ou Load-Balancing) et la tolérance de pannes.

broadcast Mode broadcast (diffusion) : les données sont envoyées à toutes les interfaces actives.

802.3ad IEEE 802.3ad agrégation dynamique des liens : permet de créer des groupes qui partagent le même paramétrage. L'intérêt est de disposer d'un mode qui ne nécessite pas forcément de configuration manuelle, les liens se découvrent mutuellement et sont agrégés automatiquement.

Conditions :

- Les pilotes des interfaces doivent supporter le dispositif ethtool, pour le contrôle de la vitesse et du mode duplex dans chaque périphérique.
- Implique que le switch, gère le mode IEEE 802.3ad pour l'agrégation dynamique des liens.

balance-tlb Équilibrage de charge auto-adaptatif en émission : seule la bande passante en sortie est adaptée à la charge de chaque interface active (Load-Balancing), (elle est calculé en fonction de la vitesse). le flux entrant est affecté à l'interface courante. Si celle-ci devient inactive, une autre prend alors l'adresse MAC de l'interface inactive et devient l'interface courante.

Conditions :

- Les pilotes des interfaces doivent supporter le dispositif ethtool, pour le contrôle de vitesse et du mode duplex, cela pour chaque interface

balance-alb Équilibrage de charge auto-adaptatif en émission et en réception : ce mode inclut en plus du mode balance-tlb un Load-Balancing (ou charge équilibrée) sur

le flux entrant et seulement pour le trafic IPV4. L'équilibrage est réalisé au niveau des réponses ARP. Le pilote du bonding intercepte les réponses des clients pour y réécrire l'adresse physique (ou adresse MAC) de l'une des interfaces du lien tout en tenant compte des spécificités du protocole ARP. La répartition entre les différentes interfaces, ce fait de façon séquentiel (Round-Robin)

Réception du trafic créé par le serveur et équilibre les charges. Quand le client envoie une requête ARP, le pilote bonding récupère les informations l'IP du client dans l'ARP. Lorsque la réponse ARP du bonding revient au client, celui-ci récupère l'adresse physique (ou adresse MAC) de l'une des interfaces du pilote bonding. Le problème résulte dans la négociation des requêtes ARP pour l'équilibrage de charge, à chaque fois qu'une requête ARP est diffusée il utilise l'une des adresses physiques du lien donc l'une des deux interfaces. Par conséquent, les clients récupèrent l'adresse physique du lien pour l'équilibrage de charge et reçoivent le trafic descendant de l'interface active. Cela est pris en charge par l'envoi d'une mise à jour (réponse ARP) à l'ensemble des clients, assignant individuellement l'adresse physique de telle sorte à redistribuer le trafic. L'équilibrage de charge est réparti de façon séquentielle (Round-Robin) parmi le groupe d'interfaces pour un plus grand débit dans le bond.

Lorsqu'une connexion est rétablie ou une nouvelle interface est rajoutée sur le bond, le trafic entrant est redistribué entre toutes les interfaces actives du bond, en initiant les réponses ARP de tous les clients en récupérant leur adresse MAC. La valeur updelay (détaillée ci-dessous) doit être réglé sur une valeur supérieure ou égale à la transmission retardée du switch (forwarding delay), de sorte que les réponses ARP les clients ne soient pas bloquées par le switch.

Conditions :

- Les pilotes des interfaces doivent supporter le dispositif ethtool, pour le contrôle de la vitesse et du mode duplex dans chaque périphérique (ou interface).
- Supporte les pilotes de base, avec l'utilisation de l'adresse physique (ou adresse MAC) de l'interface lorsque celle-ci est active. Cette adresse physique est nécessaire afin qu'il y ait toujours qu'une seule interface utilisée dans l'équipement du bond (avec `curr_active_slave`) (ou interface active) tout en ayant une adresse physique unique pour chaque interface dans le bond. Si le `curr_active_slave` (ou interface active) échoue son adresse physique est échangé avec la nouvelle `curr_active_slave` qui a été choisie.

BONDING_DEV_x_DEV_N Par défaut : `BONDING_DEV_x_DEV_N='0'`

On indique ici le nombre d'interface pour le bonding. Si par exemple vous avez pour le bonding 'eth0' et 'eth1' vous indiquez '2' dans la variable (pour les deux interfaces eth).

BONDING_DEV_x_DEV_x Par défaut : `BONDING_DEV_x_DEV_x=""`

On indique ici le nom de l'interface qui sera actif dans le bonding pour l'agrégation du lien, vous indiquez par exemple 'eth0' dans la variable. Veuillez noter, l'interface que vous utilisez pour le bonding doit être exclusif au bonding, cet interface ne doit pas être utilisée pour d'autres raccordements tels que le modem DSL, le Bridge, le VLAN ou dans le fichier de configuration base.txt

BONDING_DEV_x_MAC Par défaut : `BONDING_DEV_x_MAC=""`

Cette variable est optionnelle et peut être ignoré.

4. Les paquetages

Vous pouvez paramétrer ici l'adresse MAC de l'interface bonding que vous utilisez par défaut, qui sera utilisée pour l'agrégation des liens. Vous n'êtes pas obligé de paramétrer l'adresse MAC de l'interface bonding par défaut.

BONDING_DEV_x_MII MON Par défaut : `BONDING_DEV_x_MII MON='100'`

Cette variable est aussi optionnelle et peut également être ignoré.

Indique la fréquence de surveillance (en millisecondes) sur l'état de fonctionnement des interfaces du bonding, avec la valeur '0' la surveillance de MII MON est désactivée.

BONDING_DEV_x_USE_CARRIER Par défaut : `BONDING_DEV_x_USE_CARRIER='yes'`

Cette variable est aussi optionnelle et peut également être ignoré.

Si la surveillance du statuts est activé avec MII MON et si la (variable est sur 'yes') on demande alors à l'algorithme de surveillance d'utiliser la primitive `netif_carrier_ok()` à la place de la surveillance des registres MII ou `ETHTOOL ioctl()`. Toutes les cartes ne supportent pas la primitive `netif_carrier_ok()` auquel cas le lien est toujours considéré actif, vous pouvez (désactiver ce système par 'no')

BONDING_DEV_x_UPDELAY Par défaut : `BONDING_DEV_x_UPDELAY='0'`

Cette variable est aussi optionnelle et peut également être ignoré.

La valeur de ce paramètre doit être un multiple de la valeur `BONDING_DEV_x_MII MON`, permet de spécifier la valeur en millisecondes permettant d'activer une interface lors de la détection d'une reconnexion sur le lien (par ex. l'interface eth).

BONDING_DEV_x_DOWNDELAY Par défaut : `BONDING_DEV_x_DOWNDELAY='0'`

Cette variable est aussi optionnelle et peut également être ignoré.

La valeur de ce paramètre doit être un multiple de la valeur `BONDING_DEV_x_MII MON`, permet de spécifier la valeur en millisecondes permettant de désactiver une interface après la détection d'un problème (par ex. l'interface eth). De cette façon, la connexion du dispositif bonding est toujours active, jusqu'à ce que l'état de la liaison redevient "active".

BONDING_DEV_x_LACP_RATE Par défaut : `BONDING_DEV_x_LACP_RATE='slow'`

Cette variable est aussi optionnelle et peut également être ignoré.

Si vous avez paramétré dans la variable `BONDING_DEV_x_MODE=""` le mode '802.3ad', cette variable permet de paramétrer la fréquence d'envoi des paquets sur (un switch ou sur un ordinateur Linux), dans le cas l'agrégation dynamique 802.3ad. le paramètre 'slow' transmission des paquets toutes les 30 secondes, le paramètre 'fast' transmission des paquets toutes les 1 secondes.

BONDING_DEV_x_PRIMARY Par défaut : `BONDING_DEV_x_PRIMARY=""`

Cette variable est aussi optionnelle et peut également être ignoré.

Avec cette variable vous pouvez spécifier un périphérique de sortie primaire si le mode est réglé sur 'active-backup'. Ceci est particulièrement utile lorsque les différentes interfaces ont des vitesses différentes. les interfaces (eth0, eth2, etc) peuvent être utilisées comme interface primaire. Si une valeur est entrée dans cette variable et que l'interface est en ligne, elle sera utilisée comme premier moyen de sortie. Lorsqu'il y a un problème sur cette interface, le trafic est dirigé vers une autre interface de secours. Cependant en cas de réactivation de l'interface prioritaire, le trafic sera de nouveau redirigé vers cette interface. On favorise l'interface prioritaire la plus rapide par ex. 1000 Mbit/s les autre à 100 Mbit/s. Si l'interface à 1000 Mbit/s tombe en panne et si celle-ci est réactivé cela est beaucoup plus avantageux que de rester sur l'interface de secours à 100 Mbit/s.

BONDING_DEV_x_ARP_INTERVAL Par défaut : `BONDING_DEV_x_ARP_INTERVAL='0'`

Cette variable est aussi optionnelle et peut également être ignoré.

Permet de spécifier la fréquence en millisecondes, pour vérifier (avec leur réponse ARP) l'intervalle avec l'adresses IP spécifiées dans `BONDING_DEV_x_ARP_IP_TARGET_x`. Si aucune réponse n'est reçus après cette demande, l'interface est considérée comme perdue. Si la surveillance par ARP est utilisée dans le mode-Load-Balancing (mode 0 ou 2), le switch doit être configuré dans le mode qui permet de distribuer des paquets à travers tous les liens - par ex. Round-Robin. Si le switch est configuré pour distribuer les paquets dans le mode XOR, toutes les réponses des cibles ARP seront reçues sur le même lien et pourrait entraîner l'échec des autres membres du réseau. La valeur par défaut est '0' qui désactive la surveillance par ARP.

BONDING_DEV_x_ARP_IP_TARGET_N Par défaut : `BONDING_DEV_x_ARP_IP_TARGET_N=""`

Cette variable est aussi optionnelle et peut également être ignoré.

On indique dans cette variable le nombre d'adresse IP, pour la surveillance ARP. On peut indiquer jusqu'à 16 adresses IP maximum.

BONDING_DEV_x_ARP_IP_TARGET_x Par défaut : `BONDING_DEV_x_ARP_IP_TARGET_x=""`

Cette variable est aussi optionnelle et peut également être ignoré.

On indique ici les adresses IP à surveiller, si la variable `BONDING_DEV_x_ARP_INTERVAL > 0` est supérieur à 0. Les adresses seront spécifiées dans le format `ddd.ddd.ddd.ddd`, ces adresses seront contrôlées par les requêtes ARP pour établir la qualité de la connexion. Au moins une adresse IP doit être listée pour que la surveillance ARP fonctionne.

4.2.3. VLAN - Supporte le 802.1Q

Le VLAN supporte le standard IEEE 802.1Q, à condition que les connexions sont en association avec un switch approprié. La gestion basée sur un VLAN par ports n'est *pas* approprié. Une introduction générale au sujet du VLAN se trouve à l'adresse suivante <http://www.inetdoc.net/articles/inter-vlan-routing/inter-vlan-routing.vlan.html> ils est justement approprié pour l'entrée en matière du VLAN. Vous trouverez d'autres information et documentation sur le Net qui concerne ce sujet par ex. <http://fr.wikipedia.org/wiki/VLAN>

Faites attention S.V.P., toutes les cartes réseaux ne supportent pas le WLAN. Certaines cartes réseaux ne peuvent pas du tout traiter VLANs, d'autres ont besoin d'un MTU adapté et d'autres cartes fonctionnent parfaitement sans problèmes. L'auteur du paquetage `advanced_networking` utilise des cartes réseaux Intel avec le pilote 'e100' sans aucun problème, le réglage du MTU n'est pas nécessaire. Avec le pilote 3COM '3c59x' il est nécessaire d'adapter le MTU, le MTU doit être réglé sur 1496, autrement la carte ne fonctionnera pas correctement. Le pilote 'starfire' ne fonctionne pas correctement si l'interface VLAN est sur un bridge, dans ce cas, aucun paquet ne peut plus être reçu. pour ceux qui veulent travailler avec VLAN ils doivent veiller à ce que les pilotes de cartes réseaux pour le VLAN soient correctement pris en charge par Linux.

OPT_VLAN_DEV Par défaut : `OPT_VLAN_DEV='no'`

Avec 'yes' vous activez le programme VLAN, avec 'no' vous le désactive.

VLAN_DEV_N Par défaut : `VLAN_DEV_N=""`

Nombre d'interface VLAN à configurer.

VLAN_DEV_x_DEV Par défaut : `VLAN_DEV_x_DEV=""`

Le nom de l'interface, le switch compatible avec le VLAN. Cela peut être par ex. `'eth0'`, `'br1'` ou `'eth2'`.

VLAN_DEV_x_VID Par défaut : `VLAN_DEV_x_VID=""`

Ici on paramètre l'identification de l'interface VLAN, le nom de l'interface VLAN est identifié avec le préfix `'ethX'` (le `'0'` n'est plus le référent de l'interface). Par ex. `'42'` est le nom de l'interface VLAN, il sera indiqué sur le routeur fli4l `'eth0.42'`.

Les interfaces VLAN sur le routeur fli4l sont toujours appelées `'<device>.<vid>'`. Donc, si j'ai une interface eth pour mon réseau VLAN, un switch compatible, et si je veut configurer 3 interfaces VLANs 10, 11 et 23 virtuelle avec l'interface eth sur mon routeur fli4l, je paramètre la variable `VLAN_DEV_x_DEV='ethX'` et l'ID du VLAN dans la variable `VLAN_DEV_x_VID=""`. Mais, comme toujours, un exemple vaut mieux que mille mots, voici l'exemple :

```
OPT_VLAN_DEV='yes'
VLAN_DEV_N='3'
VLAN_DEV_1_DEV='eth0'
VLAN_DEV_1_VID='10' # Nom de l'interface : eth0.10
VLAN_DEV_2_DEV='eth0'
VLAN_DEV_2_VID='11' # Nom de l'interface : eth0.11
VLAN_DEV_3_DEV='eth0'
VLAN_DEV_3_VID='23' # Nom de l'interface : eth0.23
```

S.V.P., pensez toujours à examiner le MTU de l'interface. L'en-tête de la trame Ethernet est supérieur de 4 octets pour le VLAN. Pour certaines interface et si c'est nécessairement vous devrez changer le MTU et indiquer la valeur 1496

4.2.4. Périphérique MTU - Réglage du MTU

Dans de rare circonstance, il peut être nécessaire de régler le MTU d'une interface. Par exemple 100% des cartes réseau ne sont pas compatibles avec le VLAN et certaines on besoin d'un réglage MTU. N'oubliez pas que seul un petit nombre de cartes réseaux sont en mesure de traiter les trames Ethernet avec plus de 1500 octets !

DEV_MTU_N Par défaut : `DEV_MTU_N=""`

Cette variable est optionnelle et peut être ignoré.

Indiquez ici le nombre d'interface pour laquelle vous devez modifier la valeur MTU.

DEV_MTU_x Par défaut : `DEV_MTU_x=""`

Cette variable est aussi optionnelle et peut également être ignoré.

Indiquer ici le nom de l'interface suivi du réglage MTU. Ces deux éléments sont séparés par un espace. Par ex. pour `'eth0'` le MTU sera `'1496'`, vous pouvez voir ici l'exemple :

```
DEV_MTU_N='1'
DEV_MTU_1='eth0 1496'
```


4.2.5. BRIDGE - Pont Ethernet pour fli4l

Il s'agit ici d'un bridge Ethernet (ou pont Ethernet) authentique, il travaille selon le protocole Spanning Tree. Le fonctionnement de l'ordinateur avec le bridge semble travailler comme un switch Layer 3 avec la configuration des ports.

Si vous voulez plus d'information sur le bridging, vous pouvez aller voir le site :

La page d'accueil du projet Linux sur le bridge : <http://bridge.sourceforge.net/>.

Description détaillée du bridging standard norme 802.1d à lire obligatoirement : <http://standards.ieee.org/getieee802/download/802.1D-2004.pdf>. Les informations à partir de la page 153 sont surtout intéressante. Veuillez noter que le code source standard du bonding pour Linux est de 1998. c'est-à-dire qu'il y a seulement 16 bits pour les valeurs du PathCost.

Ici, on peut voir les différentes valeurs du délai d'attente pour le Spanning pour le calcul du protocole : <http://www.dista.de/netstpcalc.htm>

Sur la page STP, différent moyen pour travailler agréable, exemple : <http://web.archive.org/web/20060114052801/http://www.zyxel.com/support/supportnote/ves1012/app/stp.htm>

OPT_BRIDGE_DEV Par défaut : OPT_BRIDGE_DEV='no'

Avec 'yes' le paquetage bridge est activé, avec 'no' il est désactivé.

BRIDGE_DEV_BOOTDELAY Par défaut : BRIDGE_DEV_BOOTDELAY='yes'

Cette variable est optionnelle et peut être ignoré.

Le bridge a au moins $2 \times \text{BRIDGE_DEV_x_FORWARD_DELAY}$ de délai attente en seconde, c'est le temps nécessaire pour l'activer, voir si ce laps de temps est nécessaires pour le démarrage des périphériques sur fli4l, par ex. envoyer des messages sur syslog ou se connecter par DSL. Si l'entrée est laissé à 'yes' il y a automatiquement $2 \times \text{BRIDGE_DEV_x_FORWARD_DELAY}$ de délai attente. Si le bridge n'est pas directement nécessaire au démarrage, vous pouvez indiquer le paramètre 'no' pour accélérer le processus de démarrage du routeur fli4l.

BRIDGE_DEV_N Par défaut : BRIDGE_DEV_N='1'

Vous indiquez ici le nombre de bridge, ils sont indépendants les uns des autres. Chaque bridge est à considérer différents, il sont complètement isolé. C'est valable en particulier pour le réglage de la variable BRIDGE_DEV_x_STP. Le bridge devient un périphérique virtuel avec son nom 'br<numéro>'.

BRIDGE_DEV_x_NAME Par défaut : BRIDGE_DEV_x_NAME=""

Le nom du bridge est symbolique. Ce nom peut être utilisé par d'autres paquetages qui fonctionne avec le bridge indépendamment des noms des l'interfaces.

BRIDGE_DEV_x_DEVNAME Par défaut : BRIDGE_DEV_x_DEVNAME=""

Chaque bridge nécessite un nom sous la forme de 'br<numéro>'. Le <numéro> est un nombre compris entre '0' et '99'. Les entrées possibles sont 'br0', 'br9' ou 'br42'. Les noms peuvent être choisis indifféremment, les premiers bridge peuvent s'appeler 'br3' et le deuxième 'br0'.

BRIDGE_DEV_x_DEV_N Par défaut : BRIDGE_DEV_x_DEV_N='0'

Indiquer ici le nombre de périphérique qui doivent être attaché aux bridge, combien d'interface réseau peut comporter le bridge? Il peut en avoir '0', si on utilise le bridge seulement comme un espace réservé pour une adresse IP qui sera alors repris par un tunnel VPN attaché aux bridge.

BRIDGE_DEV_x_DEV_x_DEV On indique ici les interfaces réseaux attachées au bridge.

Peut être enregistré une interface eth (par ex. 'eth0'), un bonding (par ex. 'bond0') ou

4. Les paquetages

également un périphérique Vlan (par ex. 'vlan11'). Un périphérique intégré ici ne peut plus être utilisé ailleurs et ne peut recevoir aucune adresse IP.

```
BRIDGE_DEV_1_DEV_N='3'  
BRIDGE_DEV_1_DEV_1_DEV='eth0.11' #VLAN 11 sur eth0  
BRIDGE_DEV_1_DEV_2_DEV='eth2'  
BRIDGE_DEV_1_DEV_3_DEV='bond0'
```

BRIDGE_DEV_x_AGING Par défaut : `BRIDGE_DEV_x_AGING='300'`

Cette variable est optionnelle et peut être ignoré.

On indique ici le temps pour supprimer les anciennes entrée MAC dans la table du bridge. Si pendant temps spécifié ici en secondes, l'ordinateur n'a pas envoyé ou reçue de données par la carte réseau les adresses MAC la table MAC du Bridge sera supprimé.

BRIDGE_DEV_x_GARBAGE_COLLECTION_INTERVAL Par défaut : `BRIDGE_DEV_x_GARBAGE_COLLECTION_INTERVAL`

Cette variable est aussi optionnelle et peut également être ignoré.

On indique ici le temps en seconde avant de faire un «nettoyage». Dans ce cas, les entrées dynamiques du bridge sont vérifiées pour connaître les entrées obsolètes ou invalides. Cela signifie que les anciens liens ne sont plus valides et seront supprimés.

BRIDGE_DEV_x_STP Par défaut : `BRIDGE_DEV_x_STP='no'`

Cette variable est aussi optionnelle et peut également être ignoré.

Le Spanning Tree Protocole permet d'entretenir de multiples liens avec d'autres switches. De cette manière la redondance garantit l'état de marche du réseau en cas de panne. Sans la mise en place du STP, la redondance entre les liens ne serai pas possible, le réseau ne pourrais pas fonctionner. STP essaye toujours d'utiliser la connexion plus rapide entre deux switches, ainsi l'installation de deux circuits différentes est judicieuse. On pourrait par ex. installer pour une connexion 1000 Mbit/s en tant que lien principal et utiliser une deuxième connexion à 100 Mbit/s en sécurité.

Vous pouvez consulter cette page, il y a un bon article avec quelques informations de fond : http://fr.wikipedia.org/wiki/Spanning_tree_protocol.

BRIDGE_DEV_x_PRIORITY Par défaut : `BRIDGE_DEV_x_PRIORITY="`

Cette variable est aussi optionnelle et peut également être ignoré.

seulement valable si la variable `BRIDGE_DEV_x_STP='yes'` est paramétré sur yes!

Quelle priorité pour le bridge? Le bridge avec la plus petite priorité dans l'architecture bridge sera élu par rapport au bridge principal. Chaque bridge devrait avoir une priorité différente. Veuillez considérer que le bridge avec la plus faible priorité devrait disposer de la plus grande bande passante. Puisque ces toutes les 2 secondes (à modifier dans `BRIDGE_DEV_x_HELLO`) que les paquets sont envoyé, également le flux de données restant est acheminé sur celui-ci.

Les valeurs valides sont de '0' à '61440' par multiples de 4096.

BRIDGE_DEV_x_FORWARD_DELAY Par défaut : `BRIDGE_DEV_x_FORWARD_DELAY='15'`

Cette variable est aussi optionnelle et peut également être ignoré.

Seulement valable si la variable `BRIDGE_DEV_x_STP='yes'` est paramétré sur yes!

Si un lien du bridge est désactivé et qui doit être réactivé, ou bien un lien nouvellement ajoutés sur le bridge, on indique dans cette variable un laps de temps (en secondes) $\times 2$ avant que les données soit transmis de nouveau sur le lien. Ce paramètre est décisif pour

4. Les paquetages

reconnaître la durée nécessaire d'une connexion morte sur le bridge. Ce laps de temps est calculé en secondes avec la formule suivante :

BRIDGE_DEV_x_MAX_MESSAGE_AGE + (2 × **BRIDGE_DEV_x_FORWARD_DELAY**)

Il en découle avec les valeurs par défaut : $20 + (2 \times 15) = 50$ Seconde. la durée nécessaire pour reconnaître d'une connexion morte il peut être réduit, si la variable **BRIDGE_DEV_x_HELLO** est à 1 seconde et si la variable **BRIDGE_DEV_x_FORWARD_DELAY** est à 4 secondes, la variable **BRIDGE_DEV_x_MAX_MESSAGE_AGE** doit être réglée à 4 secondes. Le calcul de la valeur sera la suivante : $4 + (2 \times 4) = 12$ Secondes. Plus rapide cela ne fonctionne pas.

BRIDGE_DEV_x_HELLO Par défaut : **BRIDGE_DEV_x_HELLO**='2'

Cette variable est aussi optionnelle et peut également être ignoré.

Seulement valable si la variable **BRIDGE_DEV_x_STP**='yes' est paramétré sur yes !

Avec la variable **BRIDGE_DEV_x_HELLO** on donne l'intervalle temps en secondes dans le quelle les Hello à vrai dire les messages sont envoyés sur le bridge principal. Ces messages sont nécessaires à la configuration automatique de STP.

BRIDGE_DEV_x_MAX_MESSAGE_AGE Par défaut : **BRIDGE_DEV_x_MAX_MESSAGE_AGE**='20'

Cette variable est aussi optionnelle et peut également être ignoré.

Seulement valable si la variable **BRIDGE_DEV_x_STP**='yes' est paramétré sur yes !

On indique ici la durée de validité maximum du dernier Hello ou message. Si pendant ce temps (en secondes) aucun Hello ou message est récupéré par le bridge principale, un nouveau bridge principal sera choisi. Par conséquent, cette valeur ne peut être **jamais** plus petit que $2 \times \text{BRIDGE_DEV_x_HELLO}$.

BRIDGE_DEV_x_DEV_x_PORT_PRIORITY Par défaut : **BRIDGE_DEV_x_DEV_x_PORT_PRIORITY**='128'

Cette variable est aussi optionnelle et peut également être ignoré.

Seulement valable si la variable **BRIDGE_DEV_x_STP**='yes' est paramétré sur yes !

Cette variable est important seulement lorsque plusieurs connexions a le même paramètre dans **BRIDGE_DEV_x_DEV_x_PATHCOST** et la même destination. Si tel est le cas, la connexion avec la priorité la plus faible sera retenue.

les valeurs valides sont de '0' à '240' avec un multiple de '16'.

BRIDGE_DEV_x_DEV_x_PATHCOST Par défaut : **BRIDGE_DEV_x_DEV_x_PATHCOST**='100'

Cette variable est aussi optionnelle et peut également être ignoré.

Seulement valable si la variable **BRIDGE_DEV_x_STP**='yes' est paramétré sur yes !

Détermine indirectement la bande passante pour cette connexion. En fonction de la valeur faible ou élevée, plus la bande passante est élevé et plus la connexion est prioritaire.

La base de calcul proposée est 1000000 / kbit/s voici les valeurs listées dans le tableau 4.1. S'il vous plaît, faites attention qu'au calcul la bande passante réelle utilisable qui doit être employée dans la formule. Il en résultera, surtout des valeurs pour le WLAN nettement plus faibles par rapport à ce que l'on pourrait s'y attendre.

Remarque : le standard IEEE actuel de 2004 utilise pour le calcul de bande passante 32 Bit entiers qui ne sont pas encore supporté par Linux.

4.2.6. Remarque

Un bridge transmet tout type de données Ethernet - Par exemple il peut être normal avec un modem DSL via WLAN (ou sans fil) réagir comme une interface sans fil. Ce qui permet, par

Bande passante	Valeur de <code>BRIDGE_DEV_x_DEV_x_PATHCOST</code>
64 kbit/s	15625
128 kbit/s	7812
256 kbit/s	3906
10 Mbit/s	100
11 Mbit/s	190
54 Mbit/s	33
100 Mbit/s	10
1 Gbit/s	1

TABLE 4.1.: Les valeurs de la variable `BRIDGE_DEV_x_DEV_x_PATHCOST` sont en fonction de la bande passante

exemple, d'utiliser uniquement le bridge comme un point d'accès WLAN, il faut être attentif aux risques en matière de sécurité, un contrôle est recommandé. Il ne s'agit pas d'un paquet indésirable qui arrive sur le bridge vienne infecter le réseau (cela veut dire que le filtrage de paquets du bridge n'est pas actif dans `fli4l`!). Il est possible d'activer `EBTables` qui est supporté par `fli4l` pour ce filtrage.

4.2.7. `EBTables` - `EBTables` pour `fli4l`

A partir de la version 2.1.9 `fli4l` support un `EBTables` rudimentaire. En mettant la variable `OPT_EBTables='yes'` sur `yes` vous activez `EBTables`. Tous les modules `EBTables` sont chargés dans le Kernel (ou noyau) et le programme `ebtables` est mis à disposition sur le routeur `fli4l`. Cela signifie que nous devons écrire le script `EBTables` complètement soi-même.

Pour plus d'information sur le support `EBTables` vous pouvez lisez la documentation sur le site Web `EBTables` : <http://ebtables.sourceforge.net>.

Il y a la possibilité d'indiquer les commandes `EBTables` avant et après `netfilters` (comme ceci `PF_INPUT_x`, `PF_FORWARD_x` etc) sur le routeur `fli4l`. Mettez selon vos besoins, dans le répertoire `config/ebtables` les fichiers `ebtables.pre` et `ebtables.post`. Le fichier `ebtables.pre` est exécuté avant la configuration de `netfilters`, puis le fichier `ebtables.post` sera exécuté. S'il vous plaît rappelez-vous qu'une erreur dans le scripts `ebtables` peuvent interrompre le processus de démarrage du routeur `fli4l`!

Avant d'utiliser le support `EBTables` vous devez lire complètement la documentation `EBTables`. Avec utilisation `EBTables` vous pouvez changer le comportement du routeur `fli4l`! Par exemple le filtrage du module `Mac` : `PF_FORWARD` ne fonctionnera pas comme d'habitude.

D'une façon très intéressant, sur l'adresse Web qui suit un petit aperçu du fonctionnement du support `EBTables` : http://ebtables.sourceforge.net/br_fw_ia/br_fw_ia.html.

4.2.8. `ETHTOOL` - Paramètres pour carte réseau Ethernet

Si vous activez cette variable `OPT_ETHTOOL='yes'` le programme `ethtool` sera copié dans `fli4l`, d'autres paquetages installés pourront aussi l'utiliser. Grâce à ce programme, différents paramètres de cartes réseaux Ethernet et de pilotes pourront être affichés et modifiés.

ETHTOOL_DEV_N Vous pouvez indiquer ici le nombre de paramètres, qui peuvent être définis au moment du démarrage.

Par défaut : `ETHTOOL_DEV_N='0'`

ETHTOOL_DEV_x Vous spécifiez dans cette variable `ETHTOOL_DEV_x` le périphérique réseau pour lequel les réglages devrait s'appliquer.

Exemple : `ETHTOOL_DEV_1='eth0'`

ETHTOOL_DEV_x_OPTION_N Vous indiquez ici `ETHTOOL_DEV_x_OPTION_N` le nombre de paramètres pour le périphérique.

ETHTOOL_DEV_x_OPTION_x_NAME

ETHTOOL_DEV_x_OPTION_x_VALUE Dans cette variable `ETHTOOL_DEV_x_OPTION_x_NAME` vous donnez un nom et dans celle-ci `ETHTOOL_DEV_x_OPTION_x_VALUE` vous indiquez la valeur à modifier.

Voici une liste d'options et de valeurs possibles, qui pourront être activées :

- `speed 10|100|1000|2500|10000` vous pouvez ajouter HD ou FD (par défaut FD = Full-Duplex)
- `autoneg on|off`
- `advertise %x`
- `wol p|u|m|b|a|g|s|d`

Exemple :

```
OPT_ETHTOOL='yes'
ETHTOOL_DEV_N='2'
ETHTOOL_DEV_1='eth0'
ETHTOOL_DEV_1_OPTION_N='1'
ETHTOOL_DEV_1_OPTION_1_NAME='wol'
ETHTOOL_DEV_1_OPTION_1_VALUE='g'
ETHTOOL_DEV_2='eth1'
ETHTOOL_DEV_2_OPTION_N='2'
ETHTOOL_DEV_2_OPTION_1_NAME='wol'
ETHTOOL_DEV_2_OPTION_1_VALUE='g'
ETHTOOL_DEV_2_OPTION_2_NAME='speed'
ETHTOOL_DEV_2_OPTION_2_VALUE='100hd'
```

Pour plus d'informations, vous pouvez consulter la documentation sur `ethtool` : <http://linux.die.net/man/8/ethtool>

4.2.9. Exemples

Un exemple simple est certainement plus utile pour comprendre. Nous allons dans notre exemple connecter 2 immeubles, avec un liaison à 2 x 100 Mbit/s. A ce sujet il y aura d'un bâtiment à l'autre 4 réseaux séparés.

Pour réaliser cela, une combinaison bonding (avec 2 connexions à 100 Mbit/s pour une transmission de performance), un VLAN (une agrégation de la ligne pour pouvoir transporter plusieurs réseaux distincts) et un bridging (pour pouvoir installer les réseaux dans le bâtiment avec l'entité Bond/VLAN (tester avec succès avec les cartes 2x Intel e100 et la carte ANA6944 1x Adaptec 4-Port). Dans cet exemple les deux interfaces e100 ce nomme '`eth0`' et '`eth1`', sont utilisés pour la connexion des bâtiments. Nous n'avons pas d'autres types de carte connus à par

4. Les paquetages

Intel e100 qui s'associent sans problème avec le VLAN. en principe les cartes gigabit doivent aussi fonctionner. Les 4 ports de la carte multi-port est utilisé pour les réseaux respectifs ont les noms d'interfaces de 'eth2' à 'eth5'.

D'abord on construit les deux lignes à 100 Mbit/s pour le Bonding :

```
OPT_BONDING_DEV='yes'
BONDING_DEV_N='1'
BONDING_DEV_1_DEVNAME='bond0'
BONDING_DEV_1_MODE='balance-rr'
BONDING_DEV_1_DEV_N='2'
BONDING_DEV_1_DEV_1='eth0'
BONDING_DEV_1_DEV_2='eth1'
```

Nous avons produit le périphérique 'bond0'. Nous allons construire maintenant les VLANs sur la liaison bonding, nous utiliserons dans cette exemple les ID-VLANs (ou identifiant-VLAN) 11, 22, 33 et 44 :

```
OPT_VLAN_DEV='yes'
VLAN_DEV_N='4'
VLAN_DEV_1_DEV='bond0'
VLAN_DEV_1_VID='11'
VLAN_DEV_2_DEV='bond0'
VLAN_DEV_2_VID='22'
VLAN_DEV_3_DEV='bond0'
VLAN_DEV_3_VID='33'
VLAN_DEV_4_DEV='bond0'
VLAN_DEV_4_VID='44'
```

Et maintenant sur ces liaisons VLAN, nous allons construire le bridge pour mettre en relation les différent segments du réseau. Un routage n'est pas nécessaire.

```
OPT_BRIDGE_DEV='yes'
BRIDGE_DEV_N='4'
BRIDGE_DEV_1_NAME='_VLAN11_'
BRIDGE_DEV_1_DEVNAME='br11'
BRIDGE_DEV_1_DEV_N='2'
BRIDGE_DEV_1_DEV_1='bond0.11'
BRIDGE_DEV_1_DEV_2='eth2'
BRIDGE_DEV_2_NAME='_VLAN22_'
BRIDGE_DEV_2_DEVNAME='br22'
BRIDGE_DEV_2_DEV_N='2'
BRIDGE_DEV_2_DEV_1='bond0.22'
BRIDGE_DEV_2_DEV_2='eth3'
BRIDGE_DEV_3_NAME='_VLAN33_'
BRIDGE_DEV_3_DEVNAME='br33'
BRIDGE_DEV_3_DEV_N='2'
BRIDGE_DEV_3_DEV_1='bond0.33'
BRIDGE_DEV_3_DEV_2='eth4'
BRIDGE_DEV_4_NAME='_VLAN44_'
BRIDGE_DEV_4_DEVNAME='br44'
BRIDGE_DEV_4_DEV_N='2'
BRIDGE_DEV_4_DEV_1='bond0.44'
BRIDGE_DEV_4_DEV_2='eth5'
```

Maintenant, tous les 4 réseaux sont interconnectés et complètement transparent pour partager la connexion à 200 Mbit/s. Même avec une perte de connexion on pourra encore être connecté à 100 Mbit/s. Si nécessaire, vous pouvez activer même l'assistance EBTables, par exemple pour le filtrage de paquets spécifiques.

Cette configuration est construite sur deux routeurs fli4l. Je pense que ces exemples ont permis de montrer les possibilités impressionnantes du paquetage `advanced_networking`.

4.3. CHRONY - Protocole serveur/client pour la diffusion de l'heure sur le réseau

`OPT_CHRONY` a été développé pour fli4l avec le protocole (NTP) [Network Time Protocol](#) (Page 89). Ne pas confondre avec le protocole *normal* Time, lequel est disponible dans l'ancien `OPT_TIME`. Les deux protocoles ne sont pas compatibles, ainsi le nouveau protocole est nécessaire si vous avez un programme client qui possède le pro. NTP. Si vous ne voulez pas renoncer au protocole *normal* Time, il sera possible d'activer ce protocole en plus du NTP. `OPT_CHRONY` fonctionne en mode serveur et en mode client. En mode client, fli4l aura la même heure référencée sur le (serveur Time) d'Internet. En utilisant le réglage de base, vous pouvez paramétrer dans `OPT_CHRONY` trois serveur-Time différents sur le site [pool.ntp.org](#) (Page 89). Il est également possible, de sélectionner dans le fichier de configuration d'autres serveurs-Time. Vous pouvez choisir, par exemple le serveur Europe `sinvoll`. Il est possible avec le serveur `pool.ntp.org` d'indiquer, si le routeur ou le fournisseur d'accès est en Allemagne. Pour plus d'informations voir le site [pool.ntp.org](#) (Page 89).

En mode serveur `OPT_CHRONY` est utilisé comme horloge de référence pour le réseau local (LAN). Le pro. NTP fonctionne sur le port 123.

Chrony se distingue par le fait qu'il n'est pas connecté en permanence sur Internet. Dès que la connexion est arrêtée (off-line), chrony compare l'horloge par rapport au serveur Time reçu sur Internet. Ainsi chrony ne déclenche aucune nouvelle connexion. En outre, le temps de reconnexion automatique de chrony est réglé par la variable `HUP_TIMEOUT`, c'est-à-dire durant la période qui est indiquée dans celle-ci, aucune donnée ne sera échangée sur Internet.

Pour que l'heure de référence fonctionne sans problème, respectez les points suivants :

- Chrony attend que l'heure du BIOS soit dans le fuseau horaire UTC. Sinon, cela doit être modifié dans le fichier de configuration!
UTC = Heure Française, plus 1 heure en (hiver), plus 2 heures en (été)
- Depuis la version 2.1.12, Chrony corrige l'heure avec la première connexion sur Internet, même si la différence de temps est très grande (batterie de la carte-mère défectueuse).
- Si le BIOS n'indique pas correctement les chiffres de l'année 1999 (Bug de l'année 2000) ou l'implémentation défectueuse de l'heure du BIOS, la variable `OPT_Y2K='yes'` (Page 72) doit être activée !

Il est possible d'atteindre le serveur Time sur Internet avec le routage par défaut (0.0.0.0/0), étant donné que le routage par défaut de Chrony on-line change d'état. Si le routeur est configuré comme un routeur LAN, donc sans Circuit DSL et RNIS défini, Chrony sera alors en permanence en état on-line.

Mentions légales : *l'auteur ne donne aucune garantie sur la capacité de fonctionnement d'OPT_CHRONY, il ne sera pas responsable des dégâts, par exemple de la perte de données qui peut apparaître avec l'utilisation d'OPT_CHRONY.*

4.3.1. Configuration de l'OPT_CHRONY

La configuration se fait, comme tous les autres Opts de fli4l, en paramétrant le fichier. `Disque/fli4l-3.10.4/<config>/chrony.txt` selon vos besoin. La plus par les variables d'OPT_CHRONY sont optionnelles. Optionnelle veut dire que les variables, peuvent ne pas apparaître dans le fichier de configuration. C'est pour cela que le fichier de configuration `chrony` est presque vide, les variables optionnelles sont logiquement préconfigurées. Pourtant si vous voulez, une autre configuration, les variables doivent être insérées à la main. Nous pouvons voir maintenant, la description des variables séparément :

OPT_CHRONY Par défaut : `OPT_CHRONY='no'`

Avec le paramètre 'no' OPT_CHRONY sera complètement désactivé. Il n'y aura aucun changement sur le média de boot de fli4l ou dans l'archive `opt.img`. Sur le principe OPT_CHRONY ne remplace aucun élément dans l'installation de fli4l, à une exception près. Il change le fichier de filtrage, et fait en sorte, que les demandes extérieur de `chrony` ne soit pas considéré comme du trafic (fli4l accède au site sécurisé avant de raccroché). Le nouveau fichier de filtrage définit le trafic de `chrony` pour qu'il ne soit pas pris en compte par fli4l, ainsi le routeur raccrochera sûrement.

Pour activer OPT_CHRONY, mettez la variable OPT_CHRONY Sur 'yes'.

CHRONY_TIMESERVICE Par défaut : `CHRONY_TIMESERVICE='no'`

Avec la variable CHRONY_TIMESERVICE, un autre Protocole peut être activé pour la transmission du temps de référence. Il est nécessaire de le modifier seulement si les ordinateurs locaux ne peuvent pas fonctionner avec NTP. Le protocole RFC 868 supplémentaire est compatible, il fonctionne sur le port 37. Si c'est possible toujours utiliser, le protocole NTP.

Merci bien à Christoph Schulz qui a contribué au programme `srv868`.

CHRONY_TIMESERVER_N Par défaut : `CHRONY_TIMESERVER_N='3'`

Avec la variable CHRONY_TIMESERVER_N vous indiquez le nombre de serveur de temps de référence. Vous devez placer les noms de serveurs dans la variable CHRONY_TIMESERVER_x, les noms doivent correspondre à la quantité indiqué ici. Vous devez changer l'index x par rapport à la quantité croissante et totale des serveurs demandés.

Dans le réglage de base `chrony` utilise trois serveurs de temps de référence sur Internet qui sont associés au site pool.ntp.org (Page 89).

CHRONY_TIMESERVER_x Par défaut : `CHRONY_TIMESERVER_x='pool.ntp.org'`

Avec la variable CHRONY_TIMESERVER_x vous indiquez votre propre liste de serveur de temps de référence sur Internet. Les serveurs de temps de référence peuvent être spécifiés par leurs adresses IP et aussi par leurs noms de DNS.

CHRONY_LOG Par défaut : `CHRONY_LOG='/var/run'`

Avec la variable CHRONY_LOG, vous pouvez lister toutes les informations dans le répertoire défini concernant `chrony`, sur l'heure du BIOS et la correction de l'heure de référence. Ne devrait normalement pas être changé.

CHRONY_BIOS_TIME Par défaut : `CHRONY_BIOS_TIME='utc'`

Pour que `chrony` exploite correctement l'heure du BIOS (RTC = real time clock), vous devez indiquer dans la variable CHRONY_BIOS_TIME, si l'heure est au niveau local 'local' ou si l'heure est universel 'utc' (UTC - Universal Coordinated Time).

4.3.2. Aide

support technique uniquement dans le cadre de fli4l [fli4l Newsgroups](#) (Page 89).

4.3.3. Littératures

Page d'accueil de chrony : <http://chrony.tuxfamily.org/>

NTP : The Network Time Protocol : <http://www.ntp.org/>

pool.ntp.org : public ntp time server for everyone : <http://www.pool.ntp.org/fr/>

RFC 1305 - Network Time Protocol (Version 3) Specification, Implementation :
<http://www.faqs.org/rfcs/rfc1305.html>

Les newsgroups de fli4l et sont règlement : <http://www.fli4l.de/fr/aide/newsgroup/>

4.4. DHCP_CLIENT - Configuration du protocole dynamic pour les hôtes

A l'aide de ce paquetage, le routeur peut attribuer des adresses IP dynamiquement sur l'interface. Les paramètres de ce paquetage sont décrits ci-dessous.

4.4.1. OPT_DHCP_CLIENT

Le client DHCP peut être utilisé pour recevoir une adresse IP sur une ou plusieurs interface(s) du routeur - cela émane le plus souvent du fournisseur d'accès. Actuellement cette possibilité de liaison provient principalement de la Suisse, des Pays-Bas et de la France, avec l'utilisation d'un modem câblé. On a aussi parfois besoin de cette configuration, quand le routeur est intégré derrière un autre routeur qui distribue les adresses IP par DHCP (par ex. derrière une FritzBox).

Au démarrage du routeur, les interfaces spécifiées obtiennent une adresse IP. Ensuite, cette interface est affectée et si besoin une route par défaut est définie pour cette interface.

OPT_DHCP_CLIENT Il faut paramétrer 'yes' dans cette variable, si vous voulez utiliser le client DHCP.

Configuration par défaut : `OPT_DHCP_CLIENT='no'`

DHCP_CLIENT_TYPE Le paquetage client DHCP a actuellement deux protocoles différents le dhclient et le dhcpcd. Vous pouvez choisir et indiquer ici le protocole que vous souhaitez.

Configuration par défaut : `DHCP_CLIENT_TYPE='dhcpcd'`

DHCP_CLIENT_N Vous indiquez ici, le nombre d'interfaces à configurer.

DHCP_CLIENT_x_IF Vous indiquez ici, l'interface à configurer qui est référencée par `IP_NET_x_DEV`, par ex. `DHCP_CLIENT_1_IF='IP_NET_1_DEV'`. Le client-DHCP récupère le périphérique associé à la variable correspondante. Celle-ci doit être enregistrée dans le fichier base.txt, toujours dans le fichier base.txt à la place de l'adresse IP et du masque de sous réseau du périphérique vous devez paramétrer '*dhcp*'

DHCP_CLIENT_x_ROUTE Si vous voulez appliquée une route pour l'interface, vous pouvez l'indiquer ici. La variable peut être paramétrée avec les valeurs suivantes :

none Aucune route n'est appliquée sur l'interface.

default Une route par défaut est appliquée sur l'interface.

imond Imond gère la route par défaut pour cette interface.

Configuration par défaut : `DHCP_CLIENT_x_ROUTE='default'`

DHCP_CLIENT_x_USEPEERDNS Si cette variable est paramétrée sur 'yes' et si la route par défaut est configurée sur ce périphérique, alors les demandes DNS du routeur seront transférées au serveur DNS du fournisseur d'accès Internet, pour que cela fonctionne vous devez paramétrer les serveurs DNS dans le fichier base.txt.

Configuration par défaut : `DHCP_CLIENT_x_USEPEERDNS='no'`

DHCP_CLIENT_x_HOSTNAME Certains fournisseurs d'accès Internet demandent un nom d'hôte pour la connexion Internet. Ce nom doit être fournie par le FAI et doit être indiqué ici. Ce nom ne doit pas être identique au nom d'hôte du routeur.

Configuration par défaut : `DHCP_CLIENT_x_HOSTNAME=""`

DHCP_CLIENT_x_STARTDELAY Cette variable est optionnelle, elle sert à retarder le départ du client-DHCP.

Dans certaines installations (par exemple lorsque fli4l est configuré en tant que client DHCP derrière un modem câblé, une FritzBox, ...) il est nécessaire d'attendre que le serveur DHCP soit redémarré pour le paramétrage du client, par exemple lors d'une coupure d'électricité.

Configuration par défaut : `DHCP_CLIENT_x_STARTDELAY='0'`

DHCP_CLIENT_x_WAIT Normalement, le client DHCP démarre en arrière-plan. Cela signifie que le processus de Boot n'est pas retardé par la création de l'adresse IPv4. si un paquetage installé sur le routeur à besoin rapidement d'une adresse configurée, il est nécessaire, que l'adresse IP soit créée avant que le processus de Boot ne démarre, (par exemple pour l'OPT_IGMP). Dans ce cas, vous pouvez activer la variable `DHCP_CLIENT_x_WAIT='yes'`, pour forcer la surveillance de l'adresse IP.

Configuration par défaut : `DHCP_CLIENT_x_WAIT='no'`

DHCP_CLIENT_DEBUG Vous enregistrez avec cette variable des informations supplémentaires, lorsqu'une adresse IP est attribuée sur le client DHCP.

Configuration par défaut : ou à délaissier `DHCP_CLIENT_DEBUG='no'`

4.5. DNS_DHCP - Serveur DNS et DHCP - Relay DHCP et serveur DNS esclave

4.5.1. Nom d'hôte

Hôte

OPT_HOSTS Avec la variable optionnelle `opt_HOSTS`, vous pouvez désactiver la configuration des noms d'hôtes !

HOST_N HOST_x_{attribute} Tous les ordinateurs du réseau local doivent être enregistrés avec une adresse IP, un nom, un alias (ou pseudo) et éventuellement une adresse MAC pour la configuration du dhcpd. Pour ce faire, on indique d'abord le nombre d'ordinateur dans la variable `HOST_N`.

Remarque : depuis la version 3.4.0, l'enregistrement des informations du routeur sont générées dans le fichier `<config>/base.txt`. Des alias supplémentaires peuvent être ajoutés dans celui-ci, voir la variable `HOSTNAME_ALIAS_N` (Page 67).

Ensuite on définit les caractéristiques de chaque hôte dans ces variables. certains paramètres sont obligatoires comme par ex. l'adresse IP, le nom et d'autres sont optionnels, c.-à-d. qu'ils ne sont pas obligatoires.

NAME – Nom d'hôte par n-fois

IP4 – Adresse IP (ipv4) de l'hôte par n-fois

IP6 – Adresse IP (ipv6) de l'hôte par n-fois (optionnelle) Si vous indiquez 'auto', l'adresse sera automatiquement calculée – soit en utilisant l'adresse `::ffff:<IPv4>`, soit (si vous avez activé `OPT_IPV6`) en utilisant l'adresse IPv6 "ordinaire", composée du préfixe IPv6 (avec le masque de sous-réseau /64) et de l'adresse MAC correspondant à l'hôte. Pour que cela fonctionne, vous devrez configurer `HOST_x_MAC` (voir ci-dessous) et configurer le paquetage ipv6.

DOMAIN – Domaine DNS de l'hôte par n-fois (optionnelle)

ALIAS_N – Nombre d'alias (ou pseudo)

ALIAS_m – m-fois le nom d'alias de l'hôte par n-fois

MAC – Adresse MAC de l'hôte par n-fois

MAC2 – Adresse MAC pour une autre interface de l'hôte par n-fois

DHCPTYP – Attribution de l'adresse IP par DHCP en fonction de l'adresse MAC ou du Nom (optionnelle)

Dans l'exemple du fichier `dns_dhcp.txt`, 4 ordinateurs sont configurés - Pour les PCs "client1", "client2", "client3" et "client4".

```
HOST_1_NAME='client1'           # 1st host: ip and name
HOST_1_IP4='192.168.6.1'
```

Les noms d'alias doivent être compatibles avec le nom complet du domaine spécifié dans `fli4l`.

L'adresse MAC est optionnelle, elle est pertinente que si `fli4l` utilise un serveur DHCP. Pour cela vous devez voir la description des options de la variable `"OPT_DHCP"` indiqué plus bas dans ce document. Si vous n'utilisez pas de serveur DHCP, vous pouvez juste indiquer l'adresse IP, le nom de l'ordinateur et peut-être un alias. L'adresse MAC à un adressage de 48 bits et se compose de 6 hexadécimales séparées par deux points.

Exemple :

```
HOST_2_MAC='de:ad:af:fe:07:19'
```

Remarque : si vous ajoutez à `fli4l` le paquetage IPv6, il n'y a pas besoin d'indiquer les adresses IPv6, si l'adresse MAC est présente dans la configuration, le paquetage IPv6 créera automatiquement les adresses IPv6 (EUI-64 modifié) en utilisant l'adresse MAC. Bien sûr, vous n'êtes pas obligé d'utiliser l'adressage automatique, vous pouvez indiquer les adresses IPv6 manuellement sur l'hôte, si vous le souhaitez.

Extra hôte

HOST_EXTRA_N HOST_EXTRA_x_NAME HOST_EXTRA_x_IP4 HOST_EXTRA_x_IP6

Avec ces variables, vous pouvez ajouter d'autres hôtes qui n'appartiennent pas au domaine local, pas exemple des hôtes qui se trouvent sur un autre domaine à travers une connexion VPN

4.5.2. Serveur DNS

OPT_DNS Pour activer le serveur DNS vous devez paramétrer la variable `OPT_DNS` sur 'yes'.

Si aucun ordinateur Windows n'est utilisé dans le LAN (ou réseau local), ou si un serveur DNS est déjà disponible dans le LAN, vous pouvez placer la variable `OPT_DNS` sur 'no' et ignorer le reste de ce paragraphe

Dans le doute, toujours utiliser la (configuration par défaut) : `OPT_DNS='yes'`

Option générale pour serveur DNS

DNS_BIND_INTERFACES Si vous indiquez 'yes' le Dnsmasq n'écouterait *pas* toutes les adresses IP mais uniquement les adresses IP qui sont configurées dans la variable `DNS_LISTEN`. Si vous indiquez 'no' le Dnsmasq écouterait toutes les adresses IP sur toutes les interfaces et en principe le Dnsmasq ne rejetterait pas les requêtes des adresses IP. Cela peut provoquer des problèmes, lorsque vous souhaitez utiliser une adresse IP pour d'autre serveur DNS. Sur fli4l un serveur DNS différent a un sens, que si vous voulez utiliser un serveur DNS esclave directement sur le routeur fli4l. Donc, si vous voulez un Dnsmasq exclusif pour certaine adresse ip vous devez sélectionner le paramètre 'yes' dans cette variable, ainsi le Dnsmasq sera utilisé uniquement pour les requêtes des adresses IP configurées via la variable `DNS_LISTEN`.

DNS_LISTEN_N DNS_LISTEN_x Si vous avez choisi d'activer la variable `OPT_DNS='yes'`, vous pouvez indiquer dans la variable `DNS_LISTEN_N` le nombre de variable à configurer et indiquez dans la variable `DNS_LISTENIP_1` l'adresse IP locale, sur laquelle les requêtes DNS ne pourront pas accéder au programme `Dnsmasq`. Si vous paramétrez la variable `DNS_LISTEN_N` sur '0' toutes les requêtes DNS des adresses IP locaux pourront accéder au programme `Dnsmasq`.

À ce stade, seul les adresses IP des interfaces existantes (ethernet, wlan ...) peuvent être utilisées, sinon vous aurez un message d'avertissement au démarrage du routeur. Il est désormais possible d'utiliser alternativement les alias, par exemple `IP_NET_1_IPADDR`.

Pour toutes les adresses indiquées, les règles ACCEPT de la chaîne INPUT seront créées pour le pare-feu si `PF_INPUT_ACCEPT_DEF='yes'` et/ou `PF6_INPUT_ACCEPT_DEF='yes'`. Si la variable `DNS_LISTEN='0'` est à zéro, les règles qui permettent l'accès au DNS seront également générées pour *toutes* les interfaces configurées.

En cas de doute, vous pouvez accepter les paramètres par défaut.

DNS_VERBOSE Enregistrement des requêtes DNS : 'yes' ou 'no'

Si vous voulez avoir des détails sur les requêtes DNS, mettez la variable `DNS_VERBOSE` sur 'yes'. Dans ce cas, les messages des requêtes DNS seront consignés sur le serveur de nom - A savoir sur l'interface syslog. Si vous voulez rendre visible et lire ce fichier journal, vous devez activer la variable `OPT_SYSLOGD='yes'` (Page 70), voir ci-dessous.

DNS_MX_SERVER Avec cette variable, vous pouvez enregistrer un nom d'hôte pour le MX (Mail-Exchanger) et pour définir dans `DOMAIN_NAME` un domaine. Si un MTA (Mail="Transport"=Agent, par exemple sendmail) est installé sur le serveur interne, une demande de DNS sera faite par Mail-Exchanger, l'objectif est d'envoyer un mail au domaine recherché. Le serveur DNS fournit ici l'hôte au MTA, pour l'envoi du mail au `DOMAIN_NAME` du domaine compétent.

Il ne s'agit pas de configuration automatique un clients de messagerie, comme par exemple Outlook ! Veuillez ne pas enregistrer votre adresse mail ici, après ne vous étonnez pas si Outlook ne fonctionne pas.

DNS_FORBIDDEN_N DNS_FORBIDDEN_x Avec ces variables, vous pouvez paramétrer les domaines pour lesquels les requêtes DNS n'auront "pas accès" au serveur DNS, ils n'auront aucune réponse.

Exemple :

```
DNS_FORBIDDEN_N='1'
DNS_FORBIDDEN_1='foo.bar'
```

Dans cet exemple une demande d'accès au domaine `www.foo.bar` répondra par une erreur. On peut aussi interdire un Top-Level-Domains (ou domaines de premier niveau) les plus connus sont `.com`, `.fr`, `.de` :

```
DNS_FORBIDDEN_1='de'
```

Dans cet exemple, la résolution de nom pour 'de' du Top-level-domain sur le Web sera supprimé pour tous les ordinateurs du réseau local.

DNS_REDIRECT_N DNS_REDIRECT_x DNS_REDIRECT_x_IP Avec ces variables, vous pouvez spécifier les domaines sur lesquels les requêtes DNS seront redirigées vers une autre adresse IP du serveur DNS.

Exemple :

```
DNS_REDIRECT_N='1'
DNS_REDIRECT_1='yourdom.dyndns.org'
DNS_REDIRECT_1_IP='192.168.6.200'
```

Dans cet exemple, une demande d'accès au domaine `yourdom.dyndns.org` sera dérivé vers l'adresse IP `192.168.6.200`. Ainsi vous pouvez dériver les domaines externe de votre choix, sur une adresse IP local de votre choix.

DNS_BOGUS_PRIV Si vous placez cette variable sur 'yes' vous ne transmettez pas les recherches inversées pour les adresses IP RFC1918 (classe d'adresse IP privée, non routables sur Internet) ces adresses ne seront pas transmis au serveur DNS, mais le Dnsmasq répondra.

DNS_FILTERWIN2K Si vous placez cette variable sur 'yes' les requêtes DNS du type SOA, SRV et ANY seront bloquées. Les services qui utilisent ce type de requête ne fonctionneront plus sans une configuration supplémentaire.

Par exemple :

- XMPP (Jabber)
- SIP
- LDAP

- Kerberos
- Teamspeak3 (à partir de la version du client 3.0.8)
- Minecraft (à partir de toutes les versions 1.3.1)
- Recherche pour la gestion du contrôleur de domaine (Win2k)

Pour plus d'informations :

- Voir les explications des types de requête DNS à cette adresse :

http://en.wikipedia.org/wiki/List_of_DNS_record_types

- Voir le manuel du Dnsmasq à cette adresse :

<http://www.thekelleys.org.uk/dnsmasq/docs/dnsmasq-man.html>

- Voir les requêtes SRV dans le détail à cette adresse :

http://en.wikipedia.org/wiki/SRV_resource_record

Si vous avez indiqué le paramètre 'no' en plus des problèmes de transmission des requêtes DNS, cela peut aussi provoquer des connexions indésirables ou empêcher la fermeture d'une connexion déjà existante. Surtout si vous utilisez l'ISDN (ou Numéris) ou l'UMTS, des coûts de connexion supplémentaires peuvent survenir. Vous devez choisir ce qui est le plus important pour vous.

DNS_FORWARD_LOCAL Si vous placez cette variable sur 'yes' le routeur fli4l peut être configuré dans un domaine avec la variable DOMAIN_NAME='example.local', et via la variable DNS_ZONE_DELEGATION_x_DOMAIN='example.local' les requêtes seront résolues à partir d'un autre serveur de nom.

DNS_LOCAL_HOST_CACHE_TTL Vous indiquez dans cette variable le TTL (Time To Live, en seconde) pour les noms enregistrés dans le fichier /etc/hosts et pour les adresses IP affectés par le DHCP. La valeur par défaut pour fli4l est de 60 secondes. La valeur par défaut du TTL pour les noms enregistrés dans le Dnsmasq locale doit être de 0, en fait la mise en cache des entrées DNS sera désactivée. L'idée, est que l'exécution des baux du DHCP, etc. pourront être transmis rapidement. Exemple d'un Proxy IMAP local, il peut demander plusieurs fois par seconde un nom enregistré dans le DNS, cela occasionne des lourdes charges sur le réseau. Le compromis est donc un TTL relativement court, avec 60 secondes. Il peut même fonctionner sans ce court TTL de 60 secondes, avec une simple mise hors tension à tous moment de l'hôte, de sorte que le logiciel qui interroge ne traitera pas de toute façon la réponse de l'hôte.

DNS_SUPPORT_IPV6 (optionnelle)

Si vous placez cette variable sur 'yes' vous activez le serveur DNS supportant l'adressage IPv6.

Configuration d'une zone DNS

Le Dnsmasq peut également gérer un domaine DNS de manière autonome, c'est à dire, il a "autorité" pour ce domaine. Par conséquent, il faut faire deux choses : la première consiste à spécifier le nom du service DNS externe (!) sur fli4l pour l'envoi des requêtes, la deuxième est savoir sur quelle interface réseau que tous cela se passera. La spécification du référencement externe est nécessaire, car le domaine qui gère fli4l, est toujours un sous-domaine d'un autre domaine.² La spécification de l'interface "externe" est important parce que Dnsmasq se comporte différemment par rapport à une autre interface "interne" : le Dnsmasq ne répond

2. Nous allons supposer que personne n'utilise fli4l comme serveur DNS à la racine...

jamais aux requêtes de extérieur, en dehors de sa propre configuration de nom de domaine. En interne le Dnsmasq fonctionne naturellement comme un relay DNS pour que la résolution de noms qui n'est pas sur le routeur, fonctionne sur Internet.

D'autre part, vous devez configurer le réseau pour que la résolution de nom soit accessible vers l'extérieur. La configuration du réseaux ne doit être spécifiée avec une adresses IP publiques, parce que les adresses des hôtes privées, ne peuvent pas atteindre une IP publique qui est externe.

Ci-dessous, un exemple de configuration est décrite. Cet exemple suppose que le paquet IPv6 ainsi que le préfixe IPv6 est routé vers le réseau publique. Cette adresse peut par exemple, être fournis par le fournisseur de tunnel 6in4 comme SixXS ou Hurricane Electric.

DNS_AUTHORITATIVE Si vous activez la variable `DNS_AUTHORITATIVE='yes'`, vous activez le module Dnsmasq qui fait autorité. Toutefois, cela ne suffit pas, car vous devez fournir plus d'information (voir ci-dessous).

Paramètre par défaut : `DNS_AUTHORITATIVE='no'`

Exemple : `DNS_AUTHORITATIVE='yes'`

DNS_AUTHORITATIVE_NS Dans cette variable vous configurez le nom du DNS externe pour fli4l, vous indiquez ici le Nom de Domaine du DNS. Cela peut être un nom de DNS qui appartient à un service de DNS Dynamique.

Exemple : `DNS_AUTHORITATIVE_NS='fli4l.noip.me'`

DNS_AUTHORITATIVE_IPADDR Dans cette variable vous configurez l'adresse ou l'interface, sur la quelle les demandes de DNS du Dnsmasq doit répondre par le domaine qui fait autorité. Les noms symboliques comme `IP_NET_2_IPADDR` sont autorisés. Le Dnsmasq peut répondre seulement à *une* adresse/interface qui fait autorité.

Actuellement vous pouvez affecter seulement les adresses fixe. Les adresses qui sont produites par un accès à distance (par ex. en utilisant une connexion PPP), ne peut pas être utilisée. Ce problème sera résolu dans une version ultérieure de fli4l.

Important: *Il faut faire attention à se que l'adresse/interface ne dépendre jamais du réseau local, autrement aucun nom ne sera résolu dans le LAN!*

Exemple : `DNS_AUTHORITATIVE_IPADDR='IP_NET_2_IPADDR'`

DNS_ZONE_NETWORK_N DNS_ZONE_NETWORK_x Dans cette variable vous configurez l'adresse réseau, pour que le Dnsmasq qui fait autorité puisse résoudre les noms. Il fonctionne à la fois en recherche normal (le nom de l'adresse IP) ainsi qu'en recherche inversée (l'adresse IP du nom).

Un exemple complet :

```
DNS_AUTHORITATIVE='yes'
DNS_AUTHORITATIVE_NS='fli4l.noip.me'
DNS_AUTHORITATIVE_IPADDR='IP_NET_2_IPADDR' # Uplink dépend de eth1
DNS_ZONE_NETWORK_N='1'
DNS_ZONE_NETWORK_1='2001:db8:11:22::/64'    # IPv6-LAN local
```

Il est supposé que "2001:db8:11::/48" est le réseau public et sera routé vers le préfix IPv6 dans fli4l, et que 22 sous-réseau dans le LAN ont été sélectionnés.

Délégation de zone DNS

DNS_ZONE_DELEGATION_N DNS_ZONE_DELEGATION_x Il y a des situations particulières, où le référencement d'un ou plusieurs serveurs DNS est utilisé, par exemple lorsque l'on utilise fli4l en Intranet sans connexion Internet ou un mélange des deux (un Intranet avec son propre serveur DNS et en plus une connexion Internet).

Si nous imaginons le scénario suivant :

- Circuit 1 : Avec une connexion Internet
- Circuit 2 : Avec une connexion à un réseau d'entreprises 192.168.1.0 nom de domaine (firma.de)

Nous allons configurer ISDN_CIRC_1_ROUTE sur '0.0.0.0' et ISDN_CIRC_2_ROUTE sur '192.168.1.0'. Pour accéder aux ordinateurs avec l'adresse IP 192.168.1.x fli4l utilisera le circuit 2, autrement le circuit 1 sera utilisé. Si le réseau d'entreprise n'est pas public, il est possible de mettre en service un serveur DNS interne dans le réseau. Supposons, que l'adresse de ce serveur DNS est 192.168.1.12 et le nom de domaine est "firma.de".

Vous devez alors paramétrer les variables suivantes :

```
DNS_ZONE_DELEGATION_N='1'
DNS_ZONE_DELEGATION_1_UPSTREAM_SERVER_N='1'
DNS_ZONE_DELEGATION_1_UPSTREAM_SERVER_1_IP='192.168.1.12'
DNS_ZONE_DELEGATION_1_DOMAIN_N='1'
DNS_ZONE_DELEGATION_1_DOMAIN_1='firma.de'
```

Après cette configuration, les requêtes DNS seront envoyées au domaine firma.de ils utiliseront le serveur DNS interne de l'entreprise. Tous les autres requêtes DNS iront comme d'usage vers un serveur DNS sur Internet.

Autre cas :

- Circuit 1 : Internet
- Circuit 2 : Réseau d'entreprise 192.168.1.0 *avec* une connexion Internet

Ici vous avez deux possibilités d'accéder à Internet. Si vous souhaitez séparer le travail et la vie privée, vous pouvez alors paramétrer :

```
ISDN_CIRC_1_ROUTE='0.0.0.0'
ISDN_CIRC_2_ROUTE='0.0.0.0'
```

Vous définissez donc les deux circuits avec une route par défaut et vous commutez alors les circuits en utilisant le client-imond - en fonction de la demande. Dans ce cas vous devez paramétrer les variables DNS_ZONE_DELEGATION_N et DNS_ZONE_DELEGATION_x_DOMAIN_x comme décrit ci-dessous.

Si vous voulez utiliser la résolution de DNS inversé sur votre réseau, par ex. faire une recherche inversée pour certains serveurs de messageries, vous pouvez indiquer dans la variable optionnelle DNS_ZONE_DELEGATION_x_NETWORK_x, le ou les réseaux (mis en oeuvre), cela active la recherche inversée. Voici un exemple :

```
DNS_ZONE_DELEGATION_N='2'
DNS_ZONE_DELEGATION_1_UPSTREAM_SERVER_N='1'
DNS_ZONE_DELEGATION_1_UPSTREAM_SERVER_1_IP='192.168.1.12'
DNS_ZONE_DELEGATION_1_DOMAIN_N='1'
DNS_ZONE_DELEGATION_1_DOMAIN_1='firma.de'
DNS_ZONE_DELEGATION_1_NETWORK_N='1'
```


4. Les paquetages

```
DNS_ZONE_DELEGATION_1_NETWORK_1='192.168.1.0/24'  
DNS_ZONE_DELEGATION_2_UPSTREAM_SERVER_N='1'  
DNS_ZONE_DELEGATION_2_UPSTREAM_SERVER_1_IP='192.168.2.12'  
DNS_ZONE_DELEGATION_2_DOMAIN_N='1'  
DNS_ZONE_DELEGATION_2_DOMAIN_1='bspfirma.de'  
DNS_ZONE_DELEGATION_2_NETWORK_N='2'  
DNS_ZONE_DELEGATION_2_NETWORK_1='192.168.2.0/24'  
DNS_ZONE_DELEGATION_2_NETWORK_2='192.168.3.0/24'
```

Avec l'option de configuration `DNS_ZONE_DELEGATION_x_UPSTREAM_SERVER_x_QUERYSOURCEIP` vous pouvez définir l'adresse IP sortante qui interrogera le serveur DNS amont. C'est utile, par exemple quand vous allez sur le serveur amont via un VPN et si vous ne voulez pas que l'adresse locale du VPN fli4l apparaisse comme l'adresse IP source dans le serveur amont. Autre cas d'application, l'adresse IP du serveur DNS amont ne sera pas routable (cela se produit éventuellement à travers une interface VPN). Dans autre cas, il est logique que le Dnsmasq utilise l'adresse IP sortante qui est paramétré sur le routeur fli4l et que l'adresse IP du serveur DNS amont soit défini pour être accessible.

```
DNS_ZONE_DELEGATION_N='1'  
DNS_ZONE_DELEGATION_1_UPSTREAM_SERVER_N='1'  
DNS_ZONE_DELEGATION_1_UPSTREAM_SERVER_1_IP='192.168.1.12'  
DNS_ZONE_DELEGATION_1_UPSTREAM_SERVER_1_QUERYSOURCEIP='192.168.0.254'  
DNS_ZONE_DELEGATION_1_DOMAIN_N='1'  
DNS_ZONE_DELEGATION_1_DOMAIN_1='firma.de'  
DNS_ZONE_DELEGATION_1_NETWORK_N='1'  
DNS_ZONE_DELEGATION_1_NETWORK_1='192.168.1.0/24'
```

DNS_REBINDOK_N DNS_REBINDOK_x_DOMAIN Habituellement le serveur de noms *Dnsmasq* refuse de répondre à d'autre serveur de nom, s'il contient des adresses IP de réseau privé. Il empêche ainsi une certaine forme d'attaques réseau. Mais si vous avez un nom de domaine avec une adresse IP dans votre réseau et si un serveur de nom distinct responsable du réseau privé fournit les réponses exactes, il sera rejeté par le serveur *Dnsmasq*. On peut faire une liste de ces domaines dans la variable `DNS_REBINDOK_x`, les réponses appropriées des demandes au sujet de ce domaine, seront ensuite acceptées. Un autre exemple d'un serveur de nom qui fournirait les réponses aux adresses IP privées, ces serveurs sont soi-disant des "serveurs Blacklist en temps réel". Voici un exemple basé sur ces serveurs :

```
DNS_REBINDOK_N='8'  
DNS_REBINDOK_1_DOMAIN='rfc-ignorant.org'  
DNS_REBINDOK_2_DOMAIN='spamhaus.org'  
DNS_REBINDOK_3_DOMAIN='ix.dnsbl.manitu.net'  
DNS_REBINDOK_4_DOMAIN='multi.surbl.org'  
DNS_REBINDOK_5_DOMAIN='list.dnswl.org'  
DNS_REBINDOK_6_DOMAIN='bb.barracudacentral.org'  
DNS_REBINDOK_7_DOMAIN='dnsbl.sorbs.net'  
DNS_REBINDOK_8_DOMAIN='nospam.login-solutions.de'
```

4.5.3. Serveur DHCP

OPT_DHCP Avec la variable `OPT_DHCP`, vous pouvez si vous le voulez activer un serveur DHCP.

DHCP_TYPE (optionnelle)

Avec cette variable, vous déterminez si vous voulez utiliser la fonction DHCP interne avec Dnsmasq, ou si vous voulez recourir à la fonction ISC-DHCPD externe. Dans ce cas avec ISC-DHCPD le support DDNS (ou DNS dynamique) sera supprimé.

DHCP_VERBOSE Avec cette variable, vous activez les messages sur les transactions DHCP dans le log (ou fichier journal).

DHCP_LS_TIME_DYN Avec cette variable, vous indiquez le Lease-Time (ou délai du bail) standard pour des adresses IP fournies dynamiquement.

DHCP_MAX_LS_TIME_DYN Avec cette variable, vous indiquez le Lease-Time (ou délai du bail) maximum pour des adresses IP fournies dynamiquement.

DHCP_LS_TIME_FIX Avec cette variable, vous indiquez le Lease-Time standard pour des adresses IP assignées statiquement.

DHCP_MAX_LS_TIME_FIX Avec cette variable, vous indiquez le Lease-Time maximum pour des adresses IP assignées statiquement.

DHCP_LEASES_DIR Avec cette variable, vous indiquez le répertoire pour le fichier du bail DHCP. Il est possible de spécifier un chemin absolu ou d'indiquer le paramètre *auto*. Si vous avez défini *auto* le fichier du bail sera stocké dans le sous-répertoire persistant du DHCP (voir la documentation de la Base)

DHCP_LEASES_VOLATILE Le répertoire *Leases* se trouve dans le disque RAM (car avec une installation par CD le routeur n'a pas autres supports), au boot le routeur enverra un message d'avertissement, à cause de l'absence du répertoire *Leases* pour l'installation des fichiers. Cet avertissement sera annulé, si vous indiquez dans la variable DHCP_LEASES_VOLATILE la valeur *yes*.

DHCP_WINSERVER_1 Avec cette variable, vous indiquez l'adresse du premier serveur WINS. Le serveur WINS doit être installé et activé, l'adresse du serveur WINS est dans le paquetage SAMBA.

DHCP_WINSERVER_2 Avec cette variable, vous indiquez l'adresse du deuxième serveur WINS. Le serveur WINS doit être installé et activé, l'adresse du serveur WINS est dans le paquetage SAMBA.

Plage DHCP locale

DHCP_RANGE_N Avec cette variable, vous indiquez le nombre de DHCP-Ranges (ou de plage d'adresses IP).

DHCP_RANGE_x_NET Avec cette variable, vous indiquez le référencement du réseau défini dans la variable IP_NET_x.

DHCP_RANGE_x_START Avec cette variable, vous indiquez la première adresse IP.

DHCP_RANGE_x_END Avec cette variable, vous indiquez la dernière adresse IP. Les deux variables DHCP_RANGE_x_START et DHCP_RANGE_x_END peuvent aussi être laissées vides, alors, aucun DHCP-Range ne sera installé. Vous devez utiliser les autres variables, pour référencer le DHCP-IP de l'hôte avec l'attribution d'une adresse MAC, pour pouvoir transmettre les valeurs de la variable.

DHCP_RANGE_x_DNS_SERVER1 Avec cette variable, vous définissez l'adresse IP du serveur DNS pour les hôtes du DHCP. Cette variable est optionnelle. Si rien n'est enregistré,

la variable sera simplement omise et la variable utilisera l'adresse IP, associé au réseau. Il est également possible de placer 'none' dans cette variable, alors aucun serveur DNS ne sera utilisé.

DHCP_RANGE_x_DNS_SERVER2 Avec cette variable, vous définissez la seconde adresse IP du serveur DNS. Les options sont les même que dans la variable précédente.

DHCP_RANGE_x_DNS_DOMAIN Avec cette variable, vous définissez un domaine DNS spécifique pour les hôtes du DHCP de cette plage. Cette variable est optionnelle. Si rien n'est enregistré, la variable sera simplement omise et le domaine DNS par défaut DOMAIN_NAME sera utilisé.

DHCP_RANGE_x_NTP_SERVER Avec cette variable, vous définissez un serveur NTP spécifique pour les hôtes du DHCP de cette plage. Cette variable est optionnelle. Si rien n'est enregistré, la variable sera omise et l'adresse IP référencée dans la variable DHCP_RANGE_x_NET sera utilisée, pour les paquets du serveur de temps qui est activé sur le routeur. Il est également possible de placer 'none' dans cette variable, alors aucun serveur NTP ne sera utilisé.

DHCP_RANGE_x_GATEWAY Avec cette variable, vous définissez la Gateway (ou passerelle) pour les hôtes du DHCP de cette plage. Cette variable est optionnelle. Si rien n'est enregistré, la variable sera simplement omise et l'adresse IP référencée dans la variable DHCP_RANGE_x_NET sera utilisée. Il est également possible de placer 'none' dans cette variable, alors aucune Gateway ne sera utilisé.

DHCP_RANGE_x_MTU Avec cette variable, vous définissez la plage MTU du client. Cette variable est optionnelle.

DHCP_RANGE_x_OPTION_N Avec ces variables vous pouvez définir des options spécifiques pour ce domaine. Les options peuvent être trouvées dans le Manuel Dnsmasq (<http://thekelleys.org.uk/dnsmasq/docs/dnsmasq.conf.example>). Ces options n'ont pas été contrôlées, ils peuvent causer des erreurs ou des problèmes avec le serveur DNS/DHCP. Cette variable est optionnelle.

Extra plage DHCP

DHCP_EXTRA_RANGE_N Avec cette variable, vous indiquez le nombre de serveur DHCP qui ne sont pas dans le réseau local. Pour cela vous devez installer un relais DHCP qui sera sur le réseau de la gateway (ou passerelle).

DHCP_EXTRA_RANGE_x_START Avec cette variable, vous indiquez la première adresse IP.

DHCP_EXTRA_RANGE_x_END Avec cette variable, vous indiquez la dernière adresse IP.

DHCP_EXTRA_RANGE_x_NETMASK Avec cette variable, vous indiquez le masque de sous réseau.

DHCP_EXTRA_RANGE_x_DNS_SERVER Avec cette variable, vous indiquez l'adresse du serveur DNS pour ce domaine.

DHCP_EXTRA_RANGE_x_NTP_SERVER Avec cette variable, vous indiquez l'adresse du serveur NTP pour ce domaine.

DHCP_EXTRA_RANGE_x_GATEWAY Avec cette variable, vous indiquez l'adresse de la Gateway par défaut pour ce domaine.

DHCP_EXTRA_RANGE_x_MTU Avec cette variable, vous indiquez la plage MTU du client. Cette variable est optionnelle.

DHCP_EXTRA_RANGE_x_DEVICE Avec cette variable, vous indiquez l'interface réseau pour accéder à ce domaine.

Clients DHCP non autorisés

DHCP_DENY_MAC_N Avec cette variable, vous indiquez le nombre d'adresses MAC des hôtes, dont l'accès aux adresses du serveur DHCP sera refusé.

DHCP_DENY_MAC_x Avec cette variable, vous indiquez les adresses MAC des hôtes, dont l'accès aux adresses du serveur DHCP sera refusé.

Supporte le boot par le réseau

Dnsmasq supporte les clients, qui lancent le Bootp/PXE via le réseau pour booter (ou démarrer) [fili4l](#). Les informations nécessaires sont fournies par le Dnsmasq pour configurer l'hôte sur le sous-réseau. Les variables nécessaires sont **DHCP_RANGE_%-** et **HOST_%-**. Ce paragraphe décrit l'installation et le fichier de boot avec (***_PXE_FILENAME**), le serveur met à disposition les variables (***_PXE_SERVERNAME** et ***_PXE_SERVERIP**), éventuellement (***_PXE_OPTIONS**) si nécessaire pour les options. De plus, on peut activer un serveur TFTP interne, si bien que le boot sera complètement supporté par Dnsmasq.

HOST_x_PXE_FILENAME DHCP_RANGE_x_PXE_FILENAME Avec cette variable, vous indiquez l'image boot à lancer. Avec PXE vous indiquez ici le pxe-Bootloader à charger, par exemple pxegrub, pxelinux ou un autre Bootloader qui convient.

HOST_x_PXE_SERVERNAME HOST_x_PXE_SERVERIP DHCP_RANGE_x_PXE_SERVERNAME Avec ces variables, vous indiquez Le nom et l'adresse IP du serveur TFTP, ces variables doivent rester vides, si le routeur est utilisé en tant que serveur TFTP.

DHCP_RANGE_x_PXE_OPTIONS HOST_x_PXE_OPTIONS Certains Bootloader ont besoin d'options pour booter. On demande par exemple, avec pxegrub, l'option 150 pour le nom du fichier menu. Cette option peut être indiquée dans cette variable et sera reprise alors par le fichier config. Dans l'exemple pxegrub, on pourrait paramétrer comme ceci :

```
HOST_x_PXE_OPTIONS='150, "(nd)/grub-menu.lst"'
```

S'il est nécessaire d'indiquer plusieurs options, ils seront simplement séparés par un espace.

4.5.4. Relais DHCP

Le relais DHCP est utilisé, lorsqu'un autre serveur DHCP assume la gestion de la plage d'adresses IP et qui ne peut pas directement être atteint par les clients.

OPT_DHCPRELAY Vous devez paramétrer cette variable sur 'yes' pour que le routeur puisse faire fonctionner le relais DHCP. Il ne faut pas activer un serveur DHCP en même temps. Configuration par défaut : **OPT_DHCPRELAY='no'**

DHCPRELAY_SERVER À ce stade, le serveur DHCP est correctement enregistré, pour que les demandes puissent passer.

DHCPRELAY_IF_N DHCPRELAY_IF_x Avec la variable `DHCPRELAY_IF_N`, on indique le nombre de cartes réseaux sur lesquelles le serveur Relay doit écouter. Dans la variable `DHCPRELAY_IF_x` on indique la carte réseau correspondantes.

L'interface sur laquelle le serveur DHCP répond aux demandes, doit être mentionnée dans la liste. En outre, il faut s'assurer que les routes de l'ordinateur, sur lequel le serveur DHCP est installé fonctionne correctement. La réponse du serveur DHCP doit provenir via l'adresse IP de l'interface sur laquelle le client DHCP dépend. Prenons le scénario suivant :

- Relais sur deux interfaces
- Interface client : `eth0`, `192.168.6.1`
- Interface serveur DHCP : `eth1`, `192.168.7.1`
- Serveur DHCP : `192.168.7.2`

Il doit y avoir pour le serveur DHCP, une route qui accède à l'objectif, ici il faut répondre à l'adresse `192.168.6.1`, est-ce que le routeur sur lequel le Relais fonctionne par défaut sur la passerelle peut accéder au serveur DHCP, si oui tout est ok.

Si ce n'est pas le cas, nous allons avoir besoin d'une extra route supplémentaire. Si le serveur DHCP veut accéder au client par le routeur `fli4l`, vous devez enregistrer dans `config/base.txt` : `IP_ROUTE_x='192.168.6.0/24 192.168.7.1'`

Pendant le fonctionnement, il y a parfois des messages d'avertissements au sujet de certains paquets ignorés, ne tenez pas compte de ces avertissements, cela a aucune incidence sur le fonctionnement normal.

Exemple :

```
OPT_DHCPRELAY='yes'
DHCPRELAY_SERVER='192.168.7.2'
DHCPRELAY_IF_N='2'
DHCPRELAY_IF_1='eth0'
DHCPRELAY_IF_2='eth1'
```

4.5.5. Serveur TFTP

Un serveur TFTP peut être utilisé dans `fli4l`, pour la transmission de fichiers. Cela peut servir, par exemple, à un client pour récupérer des fichiers sur son portable par Internet.

OPT_TFTP Cette variable active le serveur TFTP interne du `Dnsmasq`. Le paramètre par défaut est `'no'`.

TFTP_PATH Vous indiquez ici le répertoire du serveur TFTP dans lequel sera placé les fichiers, pour que les clients puissent les récupérer. Vous pouvez déposer les fichiers dans le chemin correspondant, à l'aide d'un programme adapté (par ex. `scp`).

4.5.6. YADIFA - Serveur DNS esclave

OPT_YADIFA Avec cette variable vous activez YADIFA, un serveur DNS esclave. Le paramètre par défaut est `'no'`.

OPT_YADIFA_USE_DNSMASQ_ZONE_DELEGATION Si cette variable est activé, `yadifa` produira un script de démarrage qui générera automatiquement les entrées de toutes les zones esclaves correspondant au délégation de zone pour `Dnsmasq`. Ainsi, les zones

d'esclaves sont directement interrogés par le Dnsmasq, en principe il ne sera pas nécessaire de configurer plusieurs fois YADIFA_LISTEN_x. Les réponses des requêtes de Dnsmasq sont transmises à yadifa qui écoute uniquement sur le port localhost :35353.

YADIFA_LISTEN_N Si vous avez activé OPT_YADIFA='yes', avec l'aide de la variable YADIFA_LISTEN_N vous indiquez le nombre d'adresses, si vous indiquez YADIFA_LISTEN_1 vous devez spécifier une adresse IP du réseau local, sur laquelle YADIFA devra accepter les requêtes DNS. Un numéro de port est facultatif, si vous indiquez 192.168.1.1 :5353 le serveur DNS esclave YADIFA écoutera les requêtes DNS sur le port 5353. Assurez-vous que le Dnsmasq n'écoute pas sur toutes les interfaces (voir DNS_BIND_INTERFACES). A ce stade, seuls les adresses IP des interfaces existantes (Ethernet, Wlan, ...) peuvent être utilisés, sinon il y aura un message d'avertissement au démarrage du routeur. Il est également possible d'utiliser les alias, par ex. IP_NET_1_IPADDR

YADIFA_ALLOW_QUERY_N

YADIFA_ALLOW_QUERY_x Dans ces variables vous indiquez le nombre et les adresses IP ou les réseaux pour que YADIFA puisse avoir une autorisation d'accès. YADIFA utilise les informations du filtrage de paquets de fli4l qui doit être configuré en conséquence, il faut aussi paramétrer les fichiers de configuration de YADIFA. En ajoutant le préfixe '!' à l'adresse, l'accès à l'adresse IP ou au réseau sera rejeté par YADIFA.

Le filtrage de paquets de fli4l est configuré pour YADIFA de sorte que tous les réseaux autorisés soient ajoutés dans chaque zone pour l'ensemble de la liste ipset (yadifa-allow-query). Une différenciation entre les zones pour le filtrage de paquets n'est pas possible. De plus toutes les adresses IP et de réseaux de la configuration globale dont l'accès est refusé seront ajoutées à cette liste. Il n'est donc pas possible d'étendre l'accès de chaque zone ultérieurement.

YADIFA_SLAVE_ZONE_N Dans cette variable vous indiquez le nombre de zone DNS pour YADIFA esclave.

YADIFA_SLAVE_ZONE_x Dans cette variable vous indiquez le nom de la zone DNS esclave.

OPT_YADIFA_SLAVE_ZONE_USE_DNSMASQ_ZONE_DELEGATION Dans cette variable vous activez (=yes) ou désactivez (=no) la délégation de zone pour la zone esclave du Dnsmasq.

YADIFA_SLAVE_ZONE_x_MASTER Dans cette variable vous indiquez l'adresse IP avec le numéro de port facultatif du serveur DNS maître.

YADIFA_SLAVE_ZONE_x_ALLOW_QUERY_N

YADIFA_SLAVE_ZONE_x_ALLOW_QUERY_x Dans ces variables vous indiquez le nombre et les adresses IP ou les réseaux, pour que YADIFA puisse avoir une autorisation d'accès. En outre l'accès peut être limité à des zones DNS spécifiques. YADIFA utilise ces informations pour créer un fichier de configuration YADIFA.

En ajoutant le préfixe '!' à l'adresse, l'accès à l'adresse IP ou au réseau sera rejeté par YADIFA.

4.6. DSL - DSL pour PPPoE, Fritz !DSL et PPTP

fli4l supporte la connexion DSL avec trois variantes différentes :

- PPPoE (Modem-DSL externe, avec un raccordement Ethernet, il supporte le protocole pppoe)
- PPTP (Modem externe, avec un raccordement Ethernet, il supporte le protocole pptp)
- Fritz!DSL (DSL avec adaptateur-DSL de AVM)

On peut choisir une seule connexion DSL, l'exploitation de plusieurs connexions différentes n'est malheureusement pas encore possible.

La configuration de ces différentes variantes se ressemble, c'est pour cela que nous allons décrire d'abord les variables générales, ensuite nous évoquerons séparément les variables spécifiques avec leurs options. Imond administre l'accès-DSL comme un Circuit ³, lorsque l'on active l'une des variante-DSL vous pouvez activer imond (voir [START_IMOND](#) (Page 67)).

4.6.1. Variables de configuration générales

Les paquetages utilisent les mêmes variables de configuration, ils se distinguent uniquement par leurs noms de paquetages placés en tête. Par ex. dans tous les paquets on demande le nom d'utilisateur, la variable s'appelle selon le paquetage PPPOE_USER, PPTP_USER ou FRITZDSL_USER. Par la suite, les variables seront décrites sans leur préfixe, le préfixe manquant sera remplacé par une étoile donc dans l'exemple concret la partie manquante est PPPOE (cela est valable pour tous les autres préfixes).

***_NAME** Dans cette variable nous devons donner un nom pour le circuit - un maximum de 15 caractères. il sera affiché dans le Client-imonc. Le nom ne doit pas contenir d'espaces.

Exemple : `PPPOE_NAME='DSL'`

***_USEPEERDNS** On détermine ici si les fournisseurs d'accès Internet utilisent un serveur de nom (ou DNS), nous devons enregistrer ce serveur de nom dans notre réseau local pour la durée de connexion Internet.

Logiquement cette option est uniquement utilisée pour la connexion au fournisseur d'accès Internet. En même temps, presque tous les Fournisseurs supportent ce type de transfert. Vous devez enregistrer les adresses-IP du serveur de nom (ou DNS) fournies par votre FAI dans le fichier base.txt dans la variable `DNS_FORWARDERS` et vous devez supprimer le serveur de nom qui est configuré sur votre PC du réseau local. Ensuite vous devez mettre à la place, l'adresse IP de votre routeur. Avec ce réglage la résolution des noms ne se perd pas dans le cache du serveur de nom.

Cette option offre l'avantage de toujours pouvoir travailler avec un serveur de nom le plus proche, dans la mesure où le fournisseur d'accès à une adresse IP correcte - ainsi, la résolution de nom sera plus rapide.

En cas d'une défaillance d'un serveur de DNS du fournisseur d'accès, en règle général, on pourra corriger plus rapidement la transmission des adresses des serveurs DNS du fournisseur d'accès.

Malgré tout, il est nécessaire d'indiquer un serveur de nom valide dans la variable `DNS_FORWARDERS` du fichier base.txt pour se connecter, autrement lors de la première connexion Internet la demande ne pourra pas être résolue correctement. En outre, la configuration originale du serveur de nom local est restaurée à la fin de la connexion.

Configuration standard : `*_USEPEERDNS='yes'`

3. Malheureusement en ce moment un seul Circuit-DSL est possible – Si vous voulez utiliser plusieurs Circuits, il faut construire plusieurs média

4. Les paquetages

***_DEBUG** Pour avoir des informations de débogages supplémentaires par exemple sur pppd, vous devez activer la variable *_DEBUG régler celle-ci sur 'yes'. Ces informations supplémentaires sur pppd seront enregistrées avec interface syslogd

IMPORTANT : pour que le démon syslogd fonctionne, il faut activer la variable OPT_SYSLOGD régler celle-ci sur 'yes'

***_USER, *_PASS** Dans ces variables on indique les données du Fournisseur d'accès. *_USER l'identification de l'utilisateur, *_PASS le mot de passe.

ATTENTION : pour un accès au FAI T-Online (pour l'Allemagne) il est à noter :

Le nom d'utilisateur AAAAAAAAAAAATTTTTT#MMMM est composé, du numéro co-utilisateur, puis du numéro T-online de 12 chiffres et de l'identification. Le dernier chiffre du numéro T-Online doit se terminer par '#' si le numéro de T-Online ne comporte pas les 12 chiffres.

Avec ça si cela ne fonctionne pas ! (évidemment cela peut provenir du centrale téléphonique), le caractère '#' doit être placé entre le numéro T-Online et l'identification.

Évidemment si le (numéro T-Online comporte les 12 chiffres) il n'y a pas besoin de mettre le caractère '#'.

Le nom d'utilisateur de T-Online doit se terminer par '@t-online.de'

Exemple :

```
PPPOE_USER='111111111111222222#0001@t-online.de'
```

Des infos sur la configuration des autres fournisseurs d'accès se trouvent dans la FAQ :

– http://extern.fli4l.de/fli4l_faengine/faq.php?list=category&catnr=3&prog=1

***_HUP_TIMEOUT** On peut paramétrer dans cette variable le Timeout (ou le temps d'arrêt) en seconde, s'il y a aucune transmission sur le circuit DSL (ou Internet), la connexion se coupera. Vous disposez des paramètres, pour régler le délai '0' ou aucun délai 'never' - Si vous sélectionnez 'never' le routeur a en plus une nouvelle contrainte, c'est de raccrocher immédiatement. Les modifications dans mode Dial ne sont pas possibles – Le réglage doit être sur 'auto' et doit y rester. Le paramètre 'never' est uniquement utilisé pour PPPOE et FRITZDSL.

***_CHARGEINT** On utilise cette variable pour placer une unité de temps : on paramètre ici le coût par unité téléphonique en seconde. Pour calculer le prix total des communications. En Allemagne les FAI les plus gros, facture l'unité Tél. exactement à la minute, le paramètre correct dans la variable est donc '60'. Certain FAI facture l'unité exactement en seconde dans ce cas la variable *_CHARGEINT sera de '1'.

Malheureusement, les unités de temps DSL ne son pas exploitée pleinement, comme avec ISND. c'est après le temps paramétré ici *_HUP_TIMEOUT que la transmission se coupe.

C'est pour cette raison que la variable *_CHARGEINT sert uniquement à calculer le prix des communications

***_TIMES** Dans cette variable on paramètre temps d'activation et d'arrêt de la connexion, ainsi que le prix de l'unité Tél. Il est possible d'activer des circuits 'Default-Route' différents et aussi l'utilisation de (Least-Cost-Routing). Le contrôle du routage s'effectue avec le démon (ou programme) imond.

Structure de la variable :

```
PPPOE_TIMES='times-1-info [times-2-info] ...'
```


4. Les paquetages

Il y a dans chaque times-?-info 4 sous-paramètres - ces sous-paramètres sont séparés par deux points (':').

1. Sous-paramètre : W1-W2

On indique ici les périodes des jours ouvrables, par ex. Mo-Fr ou Sa-Su, il est possible de décrire les jours en Anglais ou en Allemand. Si l'on paramètre un seul jour, il sera écrit W1-W1 par ex. Su-Su.

2. Sous-paramètre : hh-hh

On indique ici la période horaire, par ex. 09-18 ou 18-09. De 18-09 est synonyme à 18-24 plus 00-09. de 00-24 correspond à toute une journée.

3. Sous-paramètre : Charge

On indique ici le prix par minute de connexion ou par unité téléphonique en euro, par ex. 0.032 correspond à 3.2 Centimes par minute. Les unités téléphonique, sont calculées en tenant compte du temps de conversion pour un coût réel, et seront alors affichées dans le client-imonc.

4. Sous-paramètre : LC-Default-Route

Le contenu de ce sous-paramètre peut être Y ou N. cela signifie :

Y : Autorise la plage horaire et LC-Routing (ou calcul des frais) avec Default-Route (ou routage par défaut).

N : Autorise la plage horaire et le calcul des frais Tél automatiquement avec LC-Routing, il n'est pas utilisé pour autre chose.

Exemple (lire cet exemple comme une seule ligne) :

```
PPPOE_TIMES='Mo-Fr:09-18:0.049:N
             Mo-Fr:18-09:0.044:Y
             Sa-Su:00-24:0.039:Y'
```

Important: les paramètres de la variable *_TIMES doit couvrir toute la semaine. si ce n'est pas le cas, aucune connexion valide ne peut se produire.

Si vous avez placé le paramètre ("Y") pour LC-Default-Route-Circuits et que vous n'avez pas réglé la semaine complète, Il aura des interruptions dans la période de la semaine avec Default-Route. Alors il sera impossible de surfer sur Internet pendant ces périodes !

Encore un exemple simple :

```
PPPOE_TIMES='Mo-Su:00-24:0.0:Y'
```

Cet exemple est pour ceux qui utilisent un Flatrate (ou forfait d'accès internet illimité).

Encore une dernière remarque pour le LC-Routing : *Les jours fériés sont traités comme un dimanche.*

***_FILTER** fli4l se coupe automatiquement, si aucun transfert de données ne se fait sur l'interface pppoe pendant le temps paramétré dans timeout. Malheureusement, l'interface, évalue également les transferts de données, qui viennent de l'extérieur, par exemple, des tentatives de connexion avec des Clients-P2P comme eDonkey. A présent on est contacté en permanence par d'autres clients externes, il peut arriver que le routeur fli4l ne se coupe jamais.

Avoir la possibilité de filtrer le trafic et de couper la connexion, même si quelqu'un tente de se connecter.

Si on paramètre la variable `*_FILTER` sur `yes`, seul le trafic produit par sa propre machine est générée, le trafic qui vient de l'extérieur est ignoré complètement. Lorsque le trafic entrant est généré, le routeur se trouvant entre Internet et les ordinateurs réagit, et rejette celui-ci par ex. une demande de connexion, de plus les quelques paquets sortant seront ignorés. On peut voir le fonctionnement exact ici :

- <http://www.fli4l.de/hilfe/howtos/basteleien/hangup-problem-loesen/> et
- <http://web.archive.org/web/20061107225118/http://www.linux-bayreuth.de/dcforum/DCForumID2/46.html>.

Une description plus précise se trouve dans l'annexe sur l'expression et l'intégration des codes de filtrages - C'est intéressant, uniquement si vous voulez effectuer des modifications voir à la fin de la doc.

***_MTU *_MRU** Ces variables sont optionnelles, on peut paramétrer le **MTU** (maximum transmission unit) et le **MRU** (maximum receive unit). Optionnelle signifie que les variables ne sont peut être pas dans le fichier de configuration, elles sont à insérer par l'utilisateur si besoin !

normalement le MTU et le MRU sont réglés à 1492. Ce réglage doit être modifier uniquement dans des cas exceptionnels ! Ces variables ne sont pas utilisées pour OPT_PPTP.

***_NF_MSS** Avec certains fournisseurs d'accès les effets suivants se produisent :

- Le navigateur Web reçoit un lien, mais après plus rien ne se passe.
- Fonctionne avec des petits Mail, mais pas avec des plus gros Mail.
- Avec la fonction ssh, coupure du scp après avoir établi une connexion.

Afin d'éviter ces problèmes, configurez fli4l avec le MTU par défaut. Dans certains cas, ce n'est pas suffisant, c'est pourquoi fli4l permet explicitement de composer avec MSS (message segment size) et d'indiquer une valeur fournie par le fournisseur d'accès. Si le FAI ne fournit rien, vous pouvez essayer 1412 c'est une bonne valeur de départ ; la valeur est de 40 octets de moins par rapport à la valeur MTU ($mss = mtu - 40$). Cette variable est optionnelle, ce qui signifie, que la variable n'est peut être pas dans le fichier de configuration, elle est à insérer par l'utilisateur en cas de besoin ! Ces variables ne sont pas utilisées pour OPT_PPTP.

4.6.2. OPT_PPPOE - DSL avec PPPoE

En règle générale, pour la communication via ADSL, les paquets PPPoE sont nécessaires, parce que les fournisseurs d'accès ne fournissent pas de bon routeur, mais simplement un modem DSL. Entre le routeur-fli4l et le modem fournie, on utilise le protocole PPP, en particulier sur le réseau ethernet.

Une ou deux cartes ethernet peuvent être installées dans votre routeur-fli4l, pour obtenir dans le cas échéant :

- Une seule carte avec IP pour le LAN et le protocole PPP pour le modem-DSL
- Deux cartes : une avec IP pour le LAN, et l'autre avec PPP pour le modem-DSL

La meilleure option est la solution avec les deux cartes Ethernet. Parce que les deux protocoles - IP et PPPoE - sont séparés l'un de l'autre.

Mais la méthode avec une carte Ethernet fonctionne aussi. Dans ce cas, le Modem-DSL-T est simplement raccordé au Hub ou switch du réseau. Vous pouvez avoir éventuellement une légère perte de transmission lors du débit maximum.

Si vous avez des problèmes de communication entre le modem et la carte réseau, vous pouvez essayer de ralentir la vitesse de la carte réseau, en passant éventuellement en mode Half-Duplex.

4. Les paquetages

Toutes les cartes réseaux PCI peut être configurées pour fonctionner dans les différents modes, mais seulement quelques cartes ISA. Soit vous utilisez le programme `ethtool` qui est dans le paquetage `advanced_networking`, soit vous créez un support de démarrage sous DOS avec les outils de configuration de la carte. Démarrer `fli4l` avec ce support et exécuter l'outil de configuration de la carte pour choisir et enregistrer le mode de fonctionnement plus lent pour la carte. Le programme de configuration et le pilote sont généralement fournis sur une disquette à l'achat de la carte, ils peuvent aussi être téléchargeables depuis le site Web du fabricant. Éventuellement, vous pouvez rechercher et trouver des informations sur les cartes dans wiki :

– <https://ssl.networks.org/wiki/display/f/Netzwerkkarten>

Si vous utilisez deux cartes, la première sera pour le LAN (ou réseau local) et la seconde pour la connexion au modem DSL.

Seul la première carte, doit être paramétrée avec une adresse-IP.

En d'autres termes :

```
IP_NET_N='1'                # Seule *Une* carte avec l'adresse IP~!
IP_NET_1xxx='...'          # Les paramètres habituels
```

Dans la variable `PPPOE_ETH` on indique `'eth1'` et pour la deuxième carte Ethernet on définit `*aucun*` paramètre dans la variable `IP_NET_2-xxx`.

OPT_PPPOE Activé pour la prise en charge PPPoE. Configuration Standard :
`OPT_PPPOE='no'`.

PPPOE_ETH Nom des Interfaces-Ethernet

`'eth0'` 1. Carte-Ethernet

`'eth1'` 2. Carte-Ethernet

... ..

Configuration Standard : `PPPOE_ETH='eth1'`

PPPOE_TYPE *PPPOE* sert à transférer les paquets-PPP directement sur la ligne ethernet. C. à d. dans la première étape les paquets-ppp de données sont transmis par le Démon-ppp, puis dans la deuxième étape ils sont, transformés en paquets-pppoe et envoyés sur Ethernet pour arriver sur le modem-DSL. Dans la deuxième étape les paquets sont empaquetés par le Démon-pppoe ou par le Kernel. Au moyen de la variable `PPPOE_TYPE` on définit la manière d'empaquetage des paquets-pppoe.

Valeur	Description
async	Les paquets sont créés avec Démon-pppoe ; La communication entre <i>pppd</i> et <i>pppoed</i> est asynchrone.
sync	Les paquets sont créés avec Démon-pppoe ; La communication entre <i>pppd</i> et <i>pppoed</i> est synchrone. Cela conduit à une plus grande efficacité pour la communication, et une moindre charge du processeur.
in_kernel	Les paquets-ppp sont directement transformés par le Kernel-Linux, en paquet-pppoe. Il est alors inutile de communiquer avec le deuxième démon, donc une économie sur la transformation des paquets, et une moindre charge du processeur.

TABLE 4.2.: Type de création de paquet-pppoe

4. Les paquetages

Un utilisateur a fait une comparaison des différentes variantes avec un ordinateur Fujitsu Siemens PCD-H P75, voici le tableau 4.3 Présentation des résultats⁴.

fli4l	NIC	Bande passante (en aval)	Charge du CPU
2.0.8	rtl8029 + rtl8139	310 kB/s	100%
2.0.8	2x 3Com Etherlink III	305 kB/s	100%
2.0.8	SMC + 3Com Etherlink III	300 kB/s	100%
2.1.7	SMC + 3Com Etherlink III	375 kB/s	40%

TABLE 4.3.: Bande passante et charge CPU pour pppoe

PPPOE_HUP_TIMEOUT Si on utilise `in_kernel` pour le Type-PPPoE et `dialmode auto`, on peut indiquer 'never' dans `Timeout` (ou temps d'arrêt). Le routeur ne raccrochera plus et se reconnectera automatiquement après une déconnexion au FAI, Toute modification ultérieure dans `dialmodes` ne sera plus possible.

4.6.3. OPT_PPPOE_CIRC - Plusieurs accès DSL avec PPPoE (Expérimental)

Si l'on veut gérer plusieurs accès DSL, on peut le faire avec la variable `OPT_PPPOE_CIRC`. Si on place cette variable sur `yes`, on peut définir plusieurs Circuits-PPPoE. On détermine le nombre dans la variable `PPPOE_CIRC_N`, les options sont identiques aux variables `OPT_PPPOE` précédente, Il faut simplement rajouter `CIRC_x`, par exemple `PPPOE_CIRC_x_NAME` au lieu de `PPPOE_NAME`.

4.6.4. OPT_FRITZDSL - DSL avec carte Fritz !DSL

Ici la connexion Internet est activé pour une carte Fritz !DSL. On utilise Fritz !Card DSL de AVM pour se connecter à Internet. Les pilotes de ces cartes ne sont pas sous licence GPL, c'est pour cela qu'il ne sont pas livrés avec le paquetage DSL. Il est nécessaire de télécharger ces pilotes à cette adresse <http://www.fli4l.de/fr/telechargement/version-stable/pilote-avm/> et de les décompresser dans le répertoire `fli4l`.

Ces pilotes sont trop volumineux pour une disquette, il est absolument nécessaire d'installer `fli4l` sur un disque dur, si l'on veut utiliser ces pilotes.

la réalisation et la prise en charge des circuits pour les cartes Fritz !Card DSL a été rendue possible avec le soutien amical de Stefan Uterhardt (courriel: zer0@onlinehome.de).

OPT_FRITZDSL Activez cette variable pour la prise en charge des Fritz !Card DSL. Configuration Standard : `OPT_FRITZDSL='no'`.

FRITZDSL_TYPE Il existe plusieurs cartes Fritz !, avec lesquelles une connexion-DSL peut s'effectuer. On paramètre le type de carte dans la variable `FRITZDSL_TYPE`, les différents types disponible se trouvent dans le tableau 4.4.

Configuration Standard :

```
FRITZDSL_TYPE='fcdsl'
```

FRITZDSL_PROVIDER On règle avec cette variable le type de serveur. Les options possibles sont :

4. Les chiffres ont été prélevés d'un message des newsgroups `spline.fli4l` et n'a été objet d'aucun examen. L'article du Message portait ID <caf9fk\$ala\$1@bla.spline.inf.fu-berlin.de>.

4. Les paquetages

Type de carte	Application
fcdsl	Fritz !Card DSL
fcdsl2	Fritz !Card DSLv2
fcdslsl	Fritz !Card DSL SL
fcdslusb	Fritz !Card DSL USB
fcdslslusb	Fritz !Card DSL SL USB
fcdslusb2	Fritz !Card DSL USBv2

TABLE 4.4.: Cartes-Fritz

U-R2, ECI, Siemens, Netcologne, oldArcor, Switzerland, Belgium, Austria1, Austria2, Austria3, Austria4

En Allemagne il s'agit presque toujours de UR-2. Siemens et ECI sont uniquement utilisées pour d'ancien connexion.

Pour la Suisse et la Belgique Les options sont très explicites et pour l'Autriche, il faut essayer.

Si quelqu'un a une meilleure option possible pour l'Autriche, vous pouvez la communiquer merci.

Configuration Standard :

```
FRITZDSL_PROVIDER='U-R2'
```

4.6.5. OPT_PPTP - DSL avec PPTP pour l'Autriche/les Pays-Bas (Expérimental)

En Autriche (et dans d'autres pays européens), au lieu utiliser PPPoE ils utilisent Protocole-PPTP. Là aussi, une carte Ethernet est connectée au Modem-PPTP.

C'est à partir de la version 2.0 que l'accès au circuit avec le protocole PPTP a été réalisé - Avec le soutien amical de Rudolf Hämmelerle (courriel: rudolf.haemmerle@aon.at).

Deux cartes Ethernet sont utilisées pour la connexion PPTP. Cela devrait être la première carte pour le raccordement au réseau local (ou LAN), et la seconde pour la connexion au modem-DSL.

Seul la première carte, est paramétrée avec une adresse-IP.

En d'autres termes :

```
IP_NET_N='1'                # Seule *Une* carte avec l'adresse IP~!
IP_NET_1xxx='...'          # Les paramètres habituels
```

Dans la variable PPTP_ETH on indique 'eth1' et pour la deuxième carte Ethernet on définit *aucun* paramètre dans la variable IP_NET_2-xxx.

OPT_PPTP Activé pour la prise en charge avec une connexion PPTP. Configuration Standard : OPT_PPTP='no'.

PPTP_ETH Nom des Interfaces-Ethernet

'eth0' 1. Carte-Ethernet

'eth1' 2. Carte-Ethernet

... ..

Configuration Standard : PPTP_ETH='eth1'

PPTP_MODEM_TYPE Il existe différents types de modems PPTP, pour effectuer une connexion-pptp. Le type de modem est paramétré dans la variable PPTP_MODEM_TYPE, dans le tableau 4.5 sont énuméré les différents types pour les internautes.

Type-Modem	Utilisation
bbaa	Autriche
bcaa	Autriche
xdsl	Autriche, Inode xDSL@home
mxstream	les Pays-Bas

TABLE 4.5.: Type de Modem-PPTP

Configuration Standard :

PPTP_MODEM_TYPE='bcaa'

Inode xDSL@home

Le déploiement, et la mise en place pour la prise en charge de Inode xDSL@home est décrits dans l'assistance Inode⁵.

Pour l'instant, il y a peut être encore des problèmes avec le renouvellement du bail avec l'interface dhcp (L'adresse IP de l'interface dhcp qui est attribué automatiquement doit être renouvelée régulièrement) et la coupure de l'accès du circuit avec imonc, cela ne fonctionne pas toujours correctement. Ici toute aide par patch ou autre dispositif est le bien venu, les utilisateurs peuvent aussi donner leurs avis.

Avec xsdl, il y a deux autres options pour pptp :

PPTP_CLIENT_REORDER_TO Le client-pptp, qui utilise xsdl, peut régler les paquets entre la mémoire tampon et son PC. Normalement, le paquet attend 0,3 s avant d'être envoyé au PC. Grâce à cette variable, on peut faire varier le temps de 0,00 (pas de mémoire tampon) à 10.00. Le temps doit toujours être indiqué entre deux postes.

PPTP_CLIENT_LOGLEVEL Dans cette variable on indique, le nombre d'enregistrement possible que produit le client-pptp pour Debug. 0 (peu), 1 (défaut) et 2 (beaucoup).

4.6.6. OPT_POESTATUS - Moniteur pour l'état du PPPoE sur la console fli4l

Thorsten Pohlmann a développé un moniteur pour l'état du PPPoE avec les connexions DSL.

Si vous paramétrez la variable OPT_POESTATUS='yes' vous pouvez consulté le l'état de la DSL sur le 3ème écran fli4l à tout moment. Vous passez cette écran avec la combinaison de touches ALT-F3, vous pouvez revenir en arrière. 1. Ecran fli4l avec ALT-F1.

4.7. DYNDNS - Mise à jour dynamiques des services de noms de domaine

Ce paquetage est conçu pour actualiser automatiquement la connexion du nom d'hôte dynamique. Voici les services pris en charge par fli4l :

⁵. Voir http://www6.inode.at/support/internetzugang/xdsl_home/konfiguration_ethernet_linux.html

4. Les paquetages

Fournisseur	FreeDNS (afraid.org)
DYNDNS_x_PROVIDER	AFRAID
Page d'accueil	http://freedns.afraid.org

Important: Le mot de passe est la dernière partie à indiquer (après le point d'interrogation) dans l'URL, que l'on peut obtenir sur la page d'accueil du site Afraid.org (votre login dans \Rightarrow "Dynamic DNS" \Rightarrow de l'URL, est caché derrière le lien "Direct URL"). Toutes les autres données sont ignorées.

Fournisseur	Companity
DYNDNS_x_PROVIDER	COMPANITY
Page d'accueil	http://www.staticip.de/

Fournisseur	DHS International
DYNDNS_x_PROVIDER	DHS
Page d'accueil	http://www.dhs.org/

Fournisseur	DNS2Go
DYNDNS_x_PROVIDER	DNS2GO
Page d'accueil	http://dns2go.com/

Fournisseur	DNS-O-Matic
DYNDNS_x_PROVIDER	DNSOMATIC
Page d'accueil	http://www.dnsomatic.com

Fournisseur	DtDNS
DYNDNS_x_PROVIDER	DTDNS
Page d'accueil	http://www.dtdns.com/

Fournisseur	DynAccess
DYNDNS_x_PROVIDER	DYNACCESS
Page d'accueil	http://dynaccess.de/

Important: DynAccess vous offre dans le cadre de la coopération avec DynAccess-fl4l des sous-domaines. *.dyn-fl4l.de, *.dyn-fl4l.info et *.dyn-eisfair.de à des tarifs spéciaux. Si vous voulez plus d'informations à ce sujet, voir le site Internet <http://www.dyn-fl4l.de/> ou <http://www.dyn-eisfair.de/>.

Fournisseur	DynDNS.org
DYNDNS_x_PROVIDER	DYNDNS
Page d'accueil	http://dyn.com/

4. Les paquetages

Fournisseur	DynDNS.org (custom)
DYNDNS_x_PROVIDER	DYNDNSC
Page d'accueil	http://dyn.com/standard-dns/

Fournisseur	DynDNS DK
DYNDNS_x_PROVIDER	DYNDNSDK
Page d'accueil	http://dyndns.dk/

Fournisseur	dyndns :free
DYNDNS_x_PROVIDER	DYDNSFREE
Page d'accueil	http://dyndnsfree.de/

Fournisseur	eisfair.net
DYNDNS_x_PROVIDER	DYNEISFAIR
Page d'accueil	http://www.intersales.de/it-infrastruktur/dyneisfair.html

Important: *En utilisant ce service, vous soutenez le travail des développeurs de eisfair et fli4l.*

Fournisseur	DyNS
DYNDNS_x_PROVIDER	DYNSEX
Page d'accueil	http://www.dyns.cx/

Fournisseur	GnuDIP Dynamic DNS
DYNDNS_x_PROVIDER	GNUDIP
Page d'accueil	http://gnudip2.sourceforge.net/

Fournisseur	Provider Hurricane Electric
DYNDNS_x_PROVIDER	HE
Page d'accueil	https://dns.he.net/

Fournisseur	IN-Berlin e.V.
DYNDNS_x_PROVIDER	INBERLIN
Page d'accueil	http://www.in-berlin.de

Fournisseur	KONTENT
DYNDNS_x_PROVIDER	KONTENT
Page d'accueil	http://www.kontent.de/

Fournisseur	Nerdcamp.net
DYNDNS_x_PROVIDER	NERDCAMP
Page d'accueil	http://nerdcamp.net/dynamic/dns.cgi

4. Les paquetages

Fournisseur	No-IP.com
DYNDNS_x_PROVIDER	NOIP
Page d'accueil	http://www.no-ip.com/

Fournisseur	noxaDynDNS
DYNDNS_x_PROVIDER	NOXA
Page d'accueil	http://www.noxa.de/

Fournisseur	OVH.DE
DYNDNS_x_PROVIDER	OVHDE
Page d'accueil	http://www.ovh.de/

Fournisseur	PHPDYN
DYNDNS_x_PROVIDER	PHPDYN
Page d'accueil	http://www.webnmail.de/phpdyn/

Important: Vous devez héberger ce type pour votre self

Fournisseur	Regfish.com
DYNDNS_x_PROVIDER	REGFISH
Page d'accueil	http://www.regfish.de/

Fournisseur	SelfHost.de
DYNDNS_x_PROVIDER	SELFHOST
Page d'accueil	http://selfhost.de/cgi-bin/selfhost

Fournisseur	Securepoint Dynamic DNS Service
DYNDNS_x_PROVIDER	SPDNS
Page d'accueil	http://www.spdns.de/

Fournisseur	Strato
DYNDNS_x_PROVIDER	STRATO
Page d'accueil	http://www.strato.de/

Fournisseur	T-Link.de
DYNDNS_x_PROVIDER	TLINK
Page d'accueil	http://www.t-link.de/

Fournisseur	twodns.de
DYNDNS_x_PROVIDER	TWODNS
Page d'accueil	http://www.twodns.de/

4. Les paquetages

Fournisseur	ZoneEdit.com
DYNDNS_x_PROVIDER	ZONEEDIT
Page d'accueil	http://zoneedit.com/

Nous essayons de garder ces informations à jour. Néanmoins, nous déclinons toute responsabilité quand à l'exactitude de ces données. Si vous découvrez une erreur ou un changement vous pouvez envoyer un courriel à l'équipe fli4l (courriel: team@fli4l.de).

Cette liste est complète, Les autres fournisseurs d'accès ne seront pas supportés sans modification du programme. Dans annexe vous avez une explication pour ajouter sont propre fournisseur, utilisable pour les développeurs de paquetage.

Le nom d'hôte dynamique est automatiquement mis à jour, à chaque connexion Internet. Le paquetage comprend une barrière, qui empêche de mettre à jour plusieurs fois la même IP, car se n'est pas bien vu par certains fournisseurs-DynDNS et peut mener dans les cas extrêmes, au blocage du compte.

Remarque : cela peut prendre quelques minutes avant que la modification du nom d'hôte dynamique prenne effet.

Avant de commencer la configuration de ce paquetage, il faut ouvrir un compte à l'un des fournisseurs mentionnés ci-dessus. Si cela est déjà fait, vous pouvez commencer immédiatement. Si vous n'avez pas encore de compte, vous pouvez vous diriger vers le tableau ci-dessus et trouver un nom d'hôte, pour cela, il suffit de répondre aux exigences du fournisseur et à votre goût personnel.

Pour la configuration, on a besoin des données suivantes :

- Le nom du fournisseur
- Le nom d'utilisateur
- Un mot de passe
- Le nom d'hôte DynDNS

Les informations nécessaires peuvent varier selon le fournisseur, nous tenterons d'offrir autant que possible une configuration cohérente. Par exemple le nom d'hôte est parfois semblable au nom d'utilisateur, dans ce cas, nous essaierons d'utiliser toujours le champ hôte et ignorent simplement le nom d'utilisateur. Voyons à présent la suite :

OPT_DYNDNS Si cette variable est paramétrée sur 'yes', alors OPT_DYNDNS est activé.

DYNDNS_SAVE_OUTPUT Si cette variable est paramétrée sur 'yes', les demandes DynDNS peut être stockées dans un fichier sur le serveur web⁶.

DYNDNS_N Si vous avez ouvert un compte auprès de plusieurs fournisseurs DynDNS, vous pouvez avoir pour chaque connexion plusieurs noms, on adapte ce paramètre en conséquence.

DYNDNS_x_PROVIDER on indique dans cette variable le nom du fournisseur à utiliser (voir tableau ci-dessus et les instructions dans le fichier de configuration).

DYNDNS_x_USER On indique dans cette variable le nom d'utilisateur chez le fournisseur DynDNS. Souvent il s'agit, d'une adresse e-mail, on peut choisir un nom ou également un nom d'hôte DynDNS.

6. OPT_HTTPD il faut installer le paquetage HTTPD (Page 123) pour les visualiser, voir <http://www.fli4l.de/fr/telechargement/version-stable/>

DYNDNS_x_PASSWORD Vous indiquer dans cette variable le mot de passe du compte DynDNS. Prenez garde, que personne ne surveille lorsque vous éditez le fichier de configuration !

DYNDNS_x_HOSTNAME Vous indiquez ici le nom d'hôte DynDNS *complet* du compte indiqué. Par exemple, cela pourrait ressembler à ce qui suit :

- cool.nerdcamp.net
- user.dyndns.org
- fli4luser.fli4l.net

DYNDNS_x_UPDATEHOST Vous indiquez ici l'hôte qui sera mise à jour, cette variable est utilisée pour les fournisseurs qui utilisent un PHPDYN. Ce ne sont pas des fournisseurs classiques, un script est utilisé pour mettre à jour une base MySQL du serveur PowerDNS. Ce système est sous licence GPL.

DYNDNS_x_CIRCUIT vous pouvez spécifié ici, avec quel circuit le nom d'hôte sera mis à jour. Les différents circuits doivent être séparés par un espace. Il est souhaitable, d'utiliser un nom hôte seulement pour une connexion DSL. Voici quelques exemples :

```
DYNDNS_1_CIRCUIT='1 2 3'           # ISDN seul: Circuits 1 à 3
ou
DYNDNS_1_CIRCUIT='pppoe'           # DSL seul: pppoe-Circuit
ou
DYNDNS_1_CIRCUIT='dhcp'           # Mise à jour avec un fournisseur DHCP
                                   # (opt_dhcp est nécessaire)
ou
DYNDNS_1_CIRCUIT='pppoe 1'         # DSL et ISDN
ou
DYNDNS_1_CIRCUIT='static'          # fli4l derrière un routeur par ex. LTE
```

DYNDNS_x_RENEW Certains fournisseurs prévoient que tous les n jours une mise à jour sera exécutée, même si l'adresse IP n'a pas changé. On peut indiquer ici cet intervalle. Si l'on n'indique aucune valeur dans cette variable, le 29e jour la mise à jour sera exécutée. Il faut noter à cet égard, que, seulement une mise à jour est initié lors de la connexion - C'est-à-dire lors d'une connexion DSL ou ISDN ou un renouvellement du bail, sur l'interface de configuration via le DHCP, comme si vous le feriez avec un modem-câble. Lors d'une longue période sans connexion, vous devez utiliser une autre solution pour la mise à jour de la connexion.

DYNDNS_x_EXT_IP Avec cette variable vous configurez la méthode avec laquelle l'adresse IP externe sera détectée. Avec le paramètre '**no**' vous n'avez aucune possibilité d'interroger un service extérieur pour mettre à jour l'adresse IP externe, mais, il est possible d'utiliser directement l'interface WAN pour récupérer l'adresse IP externe. Cela fonctionne en général seulement quand une connexion WAN est directement installé sur fli4l, par ex. avec la DSL via le protocole PPPoE. Si vous indiquezle paramètre '**dyndns**', le service checkip.dyndns.org sera utilisé pour la mise à jour de l'adresse IP externe. Si vous utilisez le paramètre '**stun**', la liste des serveurs STUN seront interrogées un par un jusqu'à obtenir une réponse positive. Pour recevoir une adresse IP externe l'utilisation d'un service est nécessaire, si votre routeur ne peut pas recevoir d'adresse IP externe par un autre moyen. Vous devez faire attention que le routeur n'est pas en train de changer d'adresse IP externe, lorsque vous utilisez un service, si c'est le cas votre nom d'inscription dyndns ne sera pas mise à jour rapidement.

DYNDNS_x_LOGIN Certains fournisseurs exigent que l'utilisateur se connecte régulièrement à l'aide de leur compte utilisateur sur leur site internet, afin que le service ne soit pas désactivé. Si cette variable est paramétrée sur 'yes', fli4l le fera pour vous. Toutefois, cela fonctionne que si le paquetage dyndns a été préparé pour le fournisseur respectif. Actuellement, une telle activité régulière est seulement possible et nécessaire que pour le fournisseur "DYNDNS". S'il vous plaît, garder à l'esprit que cette fonction exige également de l'activation de la variable `OPT_EASYCRON='yes'` du paquetage easycron.

DYNDNS_LOGINTIME Utilisez-vous un fournisseur avec qui fli4l vous connecte régulièrement pour éviter la désactivation du service (voir ci-dessus), alors vous pouvez paramétrer cette variable, pour que cette application à lieu. Ce qui est nécessaire est de définir une valeur de temps au format Cron ; Pour plus de détails s'il vous plaît lisez la documentation du paquetage easycron. Le réglage par défaut est `0 8 * * *`, ce qui correspond à une notification journalière à huit heures du matin.

DYNDNS_ALLOW_SSL Si cette variable est paramétrée sur 'yes', la mise à jour est effectuée lorsque cela est possible en utilisant le SSL (connexion sécurisée).

DYNDNS_LOOKUP_NAMES La mise à jour de l'adresse-IP devrait être faite uniquement si l'adresse-IP change. Tous les routeur-fli4l n'ont pas de mémoire permanente, pour sauvegardée les informations de l'adresse-IP enregistrée, juste après le démarrage ces informations ne sont pas disponible. Pour éviter tout de même des mises à jour inutiles, fli4l peut dans cette situation (et seulement dans cette situation), demander au service de nom l'adresse-IP enregistrée. L'adresse-IP est alors sauvegardée dans le cache et sera vérifiée avant chaque mise à jour de celle-ci.

À noter, après un reboot (ou redémarrage), un nouvel intervalle de mise à jour commence, puis fli4l recherche dans le service de nom l'adresse-IP.

DYNDNS_DEBUG_PROVIDER Si cette variable est paramétrée sur 'yes', une trace du processus de mise à jour est enregistrée, vous pourrez par la suite examiner le problème qui c'est peut être produit.

Configuration par défaut : `DYNDNS_DEBUG_PROVIDER='no'`

4.8. EASYCRON - Exécuter une commande planifiée

Ce paquetage a été élaboré par Stefan Manske courriel: fli4l@stephan.manske-net.de et a été adapté pour la version 2.1 par l'équipe pour fli4l.

4.8.1. Configuration

Dans `OPT_EASYCRON` on peut exporter avec le fichier config des commandes et les exécuter à un moment donné.

Pour cela il faut enregistrer les paramètres suivants :

OPT_EASYCRON Avec la variable sur `OPT_EASYCRON='yes'` le paquetage est activé

Installation par défaut : `OPT_EASYCRON='no'`

EASYCRON_MAIL Depuis toujours, des problèmes se produisaient par l'envoi de Mails indésirables par crond, on peut généralement empêcher cela avec le paramètre suivant.

Installation par défaut : `EASYCRON_MAIL='no'`

EASYCRON_N Avec cette variable on indique le nombre de commande à exécuter par cron.

EASYCRON_x_CUSTOM Ceux qui sont familiers avec les paramètres de crontab, peut définir les paramètres supplémentaires comme MAILTO, PATH, ... Si vous définissez plusieurs paramètres, vous devez les séparer par \\. Vous devez être très familier avec cron pour utiliser ces options

Installation par défaut : `EASYCRON_CUSTOM=""`

EASYCRON_x_COMMAND Dans cette variable `EASYCRON_x_COMMAND` vous indiquez la commande à exécuter, par ex.

```
EASYCRON_1_COMMAND='echo 1 '>' /dev/console'
```

EASYCRON_x_TIME Avec cette variable `EASYCRON_x_TIME` vous indiquez le temps d'exécution de la syntaxe-cron.

4.8.2. Exemples

- L'ordinateur nous souhaite une "Bonne nouvelle année" chaque année.

```
EASYCRON_1_COMMAND = 'echo Bonne nouvelle année~! > /dev/console'
```

```
EASYCRON_1_TIME     = '0 0 31 12 *'
```

- La commande xxx est exécutée du lundi au vendredi de 7-20 heures à chaque heure pleine.

```
EASYCRON_1_COMMAND = 'xxx'
```

```
EASYCRON_1_TIME     = '0 7-20 0 * 1-5'
```

- Le routeur arrête la connexion DSL toutes les nuits au alentour 03 :40, il y a un temps d'attente 5 secondes avant la déconnexion. Il est possibles d'indiquer les noms de périphériques : pppoe, ipp[1-9], ppp[1-9].

```
EASYCRON_1_COMMAND = 'fli4lctrl hangup pppoe; sleep 5; fli4lctrl dial pppoe'
```

```
EASYCRON_1_TIME     = '40 3 * * *'
```

Pour plus d'informations sur les syntaxes de cron, consultez les sites

- <http://www.pro-linux.de/artikel/2/146/der-batchdaemon-cron.html>
- http://de.linwiki.org/wiki/Linuxfibel_-_System-Administration_-_Zeit_und_Steuerung#Die_Datei_crontab
- <http://web.archive.org/web/20021229004331/http://www.linux-magazin.de/Artikel/ausgabe/1998/08/Cron/cron.html>
- http://web.archive.org/web/20070810063838/http://www.newbie-net.de/anleitung_cron.html

4.8.3. Conditions

- S'utilise avec la version fli4l > 2.1.0
- Pour les anciennes versions de fli4l voir dans OPTBase-de-donnée `opt_easycron-version`

4.8.4. Installation

Décompresser simplement la paquetage `OPT_EASYCRON` dans le répertoire actuelle de fli4l.

4.9. HD - Supporte les disques dur, CompactFlash, clé USB, ...

4.9.1. OPT_HDINSTALL - Installation sur disque dur/CompactFlash

fli4l prend en charge divers supports d'installations (CD, HD, réseau, carte CompactFlash, ...) dans la version 4.0 les disquettes ne seront plus supportés, du au manque d'espace car la taille des fichiers fli4l augmente.

Toutes les étapes nécessaires à l'installation d'un disque dur sont expliquées ci-dessous.

La méthode habituelle pour une installation est d'utiliser un support de boot, vous pouvez aussi utiliser le boot par le réseau. La variable OPT_HDINSTALL prépare le disque dur. Pour l'installation si vous utilisez un support de boot et un autre support et si le paramètre BOOT_TYPE='hd' est le même pour les deux, les fichiers d'installation seront transférés directement. Si une copie directe n'est pas possible, vous pouvez transférer les fichiers plus tard en utilisant le SCP ou Imonc.

Une introduction sur les différentes variantes d'installation A ou B pour les disques durs se trouve [au début de la documentation fli4l](#) (Page 14). Veuillez SVP lire absolument la documentation avant de commencer !

Simple installation du HD en six étape

1. Créer un support de boot fli4l avec le paquetage base et en activant la variables OPT_HDINSTALL. De plus, ce support de boot doit permettre une mise à jour à distance. Il faut donc activer la variable OPT_SSHD ou activer la variable START_IMOND avec 'yes'. Pour accéder au disque dur, si les pilotes l'installation par défaut ne suffit pas, vous pouvez installer des pilotes supplémentaires en activant la variable OPT_HDDRV.
2. Démarrer le routeur avec le support de boot.
3. Lorsque le routeur est connecté, exécuter la commande "hdinstall.sh".
4. Lorsque l'installation du disque est terminée, vous pourrez copier les fichiers syslinux.cfg, Kernel, rootfs.img, opt.img et rc.cfg au moyens d'imonc ou du SCP sur le /boot du routeur. Il est recommandé de travailler avec deux répertoires fli4l, l'un pour la configuration et le second pour l'installation du hd. Pour la version HD vous allez définir la variable BOOT_TYPE='hd' et pour le support de boot il sera en fonction de son type.

Bien entendu, les fichiers systèmes pour l'installation de la version HD, doivent être transmis au routeur !

5. Enlever le support de boot, descendre ou redémarrer le routeur à l'aide (des commandes halt/reboot/poweroff). Le routeur redémarre maintenant sur le disque dur.
6. En cas de problème, lisez les sections suivantes.

Explication en détail de l'installation du HD (avec exemples)

Tout d'abord, vous devez activer le fichier config/hd.txt dans le support de boot du routeur, la variable OPT_HDINSTALL elle sert pour le script d'installation du HD et la variable OPT_HDDRV (si les pilotes supplémentaires sont nécessaires) ces variables doivent être configurées correctement. Veuillez également lire soigneusement le paragraphe OPT_HDDRV !

4. Les paquetages

La variable `BOOT_TYPE` dans `base.txt` sera sélectionné selon le support de configuration, enfin, vous pourrez effectuée la configuration. La Variable `MOUNT_BOOT` dans `base.txt` doit être paramétrée sur `'rw'`, afin de permettre plus tard de charger si c'est nécessaire une nouvelles archives (*.img) par le réseau.

Ensuite, vous démarrez le routeur à partir de la disquette. Sur la console de fli4l vous tapez `"hdinstall.sh"` le programme d'installation démarre. Après avoir répondu à quelques questions, le disque dur sera en cours de préparation pour le partitionner. A la fin de l'installation il s'affichera sur l'écran une invitation à copier les fichiers systèmes à distance, ces fichiers sont nécessaires au routeur pour booter sur le disque dur.

N'oublier en aucun cas de transférer les fichiers systèmes sur le disque dur, autrement le routeur ne démarrera pas. Après le transfert des fichiers, vous devez redémarrer le routeur, utiliser absolument les commandes `reboot/halt/poweroff`, pour redémarrer le routeur, dans le cas contraire les modifications ne seront pas prises en compte les fichiers systèmes peuvent être perdus.

Le script d'installation du routeur peut être lancé directement sur la console d'un autre PC via le SSH. Dans tous les cas, vous devez au préalable entrer le mot de passe du routeur. Vous pouvez utiliser par exemple le freeware Putty comme client SSH pour les ordinateurs Windows.

Configuration et installation du support de boot

<code>BOOT_TYPE</code>	Selon l'installation du support de boot
<code>MOUNT_BOOT='rw'</code>	Nécessaire pour pouvoir copier plus tard, une nouvelle archive (*.img) par le réseau sur le disque du routeur
<code>OPT_HDINSTALL='yes'</code>	Nécessaire pour l'installation du script et pour les outils, formatage, partitionnement du disque
<code>(OPT_HDDRV='yes')</code>	nécessaire uniquement si disque dur a besoin de pilotes spéciaux.
<code>OPT_SSHD='yes'</code>	Nécessaire, après installation du disque, pour transférer les fichiers systèmes à distance sur le routeur. Pour cela, il faut soit <code>SSHD</code> soit <code>Imond</code> (<code>IMOND='yes'</code>), ou un autre programme, qui permet le transfert des fichiers.

TABLE 4.6.: Exemple de configuration pour l'installation du support

Ici la configuration du réseau doit être paramétré correctement pour que les fichiers soient transférés sur le disque dur par le réseau. Il est recommandé de ne pas activer le `DNS_DHCP`, cela crée régulièrement des problèmes (le serveur DHCP doit installer un fichier pour les baux sur le routeur). Pour une mise à jour à distance sur le routeur vous pouvez utiliser le SCP (il se trouve dans le paquetage `SSHD`) en activant la variable `OPT_SSHD='yes'`. Alternativement, vous pouvez transférer les fichiers via le logiciel `Imond`, mais une configuration DSL ou RNIS valide est nécessaire. Ne pas installer les paquetages qui ne sont pas absolument nécessaire, donc pas de `DNS_DHCP`, `SAMBA_LPD`, `LCD`, `Portforwarding` etc.

Si à l'installation vous avez ce message d'erreur :

```
*** ERROR: can't create new partition table, see docu ***
```

Abandonner, plusieurs sources d'erreur sont possible :

- Le disque dur était en cours d'utilisation et a été interrompue lors de l'installation. Redémarrer simplement et essayer à nouveau.
- Un pilote supplémentaire est peut-être nécessaires voir `OPT_HDDRV`
- Il y a des problèmes de matériel, s'il vous plaît lisez l'annexe de ce document.

Dans la dernière étape, vous pouvez maintenant produire la version définitive de `fli4l`, vous pouvez rajouter les fichiers de configuration et les paquetages supplémentaire souhaités.

Exemple d'installation finalisé pour le type A ou type B :

Un exemple de chaque configuration est listé dans le tableau 4.7.

<code>BOOT_TYPE='hd'</code>	Nécessaire, si <code>fli4l</code> démarre sur le disque dur
<code>MOUNT_BOOT='rw ro no'</code>	Nécessaire, pour copier les nouvelles archives sur le disque par le réseau on a besoin du paramétrage 'rw'.
<code>OPT_HDINSTALL='no'</code>	Nécessaire, après une installation réussie ce programme ne sera plus utilisé.
<code>OPT_MOUNT</code>	Nécessaire, si une partition de données a été créé sur le disque dur.
<code>(OPT_HDDRV='yes')</code>	Nécessaire, uniquement si le disque dur doit être utilisée avec un pilotes supplémentaires.

TABLE 4.7.: Exemple d'installation pour le Type A ou B

La création d'une partition swap n'est pas utile sauf si le routeur dispose de moins de 32 Mio de RAM et si l'installation ne s'exécute pas sur un périphérique flash !

4.9.2. `OPT_MOUNT` - Montage automatique du système de fichiers

La variable `OPT_MOUNT` sert à monter une partition de donnée par ex. `/data`, en cas de besoin, la partition sera testée automatiquement pour vérifier les erreurs. Si éventuellement vous avez un lecteur de CD-ROM d'installé, il sera monté en `/cdrom` lorsque vous allez insérer un disque. Si vous avez activé la partition swap, vous n'avez pas besoin d'utiliser la variable `OPT_MOUNT` !

A l'installation la variable `OPT_MOUNT` lit le fichier de configuration `hd.cfg` sur la partition de démarrage et monte les partitions prés enregistrées. Si `OPT_MOUNT` est transféré via une mise à jour à distance sur un routeur déjà installé, le fichier `hd.cfg` doit être modifié manuellement.

Avec un boot à partir du CD-ROM, la variable `OPT_MOUNT` ne doit pas être utilisée. Le CD peut être monté en mettant la valeur `MOUNT_BOOT='ro'` dans la variable.

Voici le fichier `hd.cfg` qui est sur la partition DOS, le routeur fonctionne avec le type B et une partition Swap, le contenu de celui-ci sera (par exemple) :

```
hd_boot='sda1'
hd_opt='sda2'
hd_swap='sda3'
hd_data='sda4'
```



```
hd_boot_uuid='4A32-0C15'  
hd_opt_uuid='c1e2bfa4-3841-4d25-ae0d-f8e40a84534d'  
hd_swap_uuid='5f75874c-a82a-6294-c695-d301c3902844'  
hd_data_uuid='278a5d12-651b-41ad-a8e7-97ccbc00e38f'
```

Les partitions qui n'existe pas dans ce fichier seront simplement ignorées, exemple d'installation sur un disque SCSI avec le type A et sans autres partition, le fichier `hd.cfg` contiendra :

```
hd_boot='sda1'  
hd_boot_uuid='4863-65EF'
```

4.9.3. OPT_EXTMOUNT - Montage manuel du fichier système

Avec la variable `OPT_EXTMOUNT` vous pouvez monter le fichier système sur n'importe quelle partition et avec n'importe quel point de montage. Il est possible de monter manuellement le fichier système, pour disposer, par exemple d'un serveur `rsync`.

EXTMOUNT_N Vous indiquez dans cette variable le nombre d'extra partitions à monter.

EXTMOUNT_x_VOLUMEID Dans cette variable vous indiquez le nom ou l'UUID du volume à installer. Avec la commande `'blkid'` vous pouvez avoir des informations sur les noms ou les UUID des volumes installés.

EXTMOUNT_x_FILESYSTEM Dans cette variable vous indiquez le nom du fichier système de la partition. Actuellement `fi4l` supporte les fichiers systèmes suivant : `isofs`, `fat`, `vfat`, `ext2`, `ext3` et `ext4`.// (La valeur par défaut est `EXTMOUNT_x_FILESYSTEM='auto'`, avec cette valeur `fi4l` tente de déterminer le fichier système automatiquement.)

EXTMOUNT_x_MOUNTPOINT Vous indiquez ici Le chemin d'accès (point de montage) dans lequel le dispositif sera monté pour les fichiers systèmes. Le chemin d'accès ne doit pas exister sur le support, il est automatiquement généré.

EXTMOUNT_x_OPTIONS Vous devez indiquer les options supplémentaires pour le montage, ils seront transmis lors du montage du disque.

```
EXTMOUNT_1_VOLUMEID='sda2'          # device  
EXTMOUNT_1_FILESYSTEM='ext3'         # filesystem  
EXTMOUNT_1_MOUNTPOINT='/mnt/data'    # mountpoint for device  
EXTMOUNT_1_OPTIONS=''                # extra mount options passed via mount -o
```

4.9.4. OPT_HDSLEEP - Règle pour l'arrêt automatique du disque dur

Un disque dur peut être arrêté automatiquement après une certaine période d'inactivité. De cette manière la disque utilise moins d'énergie et ne fait pratiquement plus aucun bruit. Si un accès au disque dur à lieu, il redémarre à nouveau automatiquement.

Certain disque dur ne tolèrent pas les redémarrages fréquentes. C'est pour cela que nous ne devons pas régler un temps trop court. Les vieux disques durs IDE n'offrent même pas cette fonction. Avec les supports Flash Media, ce paramètre n'est pas utile ni nécessaire.

HDSLEEP_TIMEOUT On paramètre avec cette variable le temps d'inactivité, avant que le disque dur se met au repos. le disque dur s'éteindra automatiquement après le temps d'inactivité et redémarrera au prochain accès sur celui-ci. Le temps d'inactivité se paramètre en minute, de 1 minute à 20 minutes, au-delà le réglage passe en intervalle de 30 minutes jusqu'à 5 heures. Donc si on paramètre 21 ou 25 minutes il sera arrondi à 30 minutes. Si le paramètre est trop élevé certains disques durs ne tiennent pas compte de cette valeur et se mettent au repos avant le temps indiqué. Faites plusieurs tests avec des valeurs différents, car cela dépend beaucoup du matériel respectif!

```
HDTUNE_TIMEOUT='2'                # wait 2 minutes until power down
```

4.9.5. OPT_RECOVER - Option de secours

Cette variable est utilisé pour créer une option de secours (ou une restauration système), en cas de problème. Si l'option est activée, vous pouvez utiliser la commande "mkrecover.sh" pour le transfère des données sur le routeur. La commande de secours peut être activé à partir de la console. Si le paquetage "HTTPD" est installé vous pourrez alors activer la restauration système dans le menu Recover.

L'installation de secours sera disponible au prochain redémarrage de fli4l, dans le menu de Boot sélectionnez Recover sur la console.

```
OPT_RECOVER='yes'
```

4.9.6. OPT_HDDRV - Pilote pour contrôleur de disque dur

Si vous activez la variable OPT_HDDRV='yes' vous pouvez installer les pilotes supplémentaires si nécessaires, pour les disque IDE et SATA. Normalement vous n'avez pas besoin d'installer de pilote supplémentaire, parce qu'ils sont déjà chargés depuis le paquetage base.

HDDRV_N Vous indiquez ici le nombre de pilote à charger.

HDDRV_x Vous indiquez dans la variable HDDRV_1 le pilote correspondant aux contrôleurs utilisés. Une liste de contrôleurs supportés par fli4l est incluent dans le fichier de configuration.

HDDRV_x_OPTION Vous indiquez dans la variable HDDRV_x_OPTION les options qui peut être nécessitent aux contrôleurs utilisés. Par exemple une adresse I/O. On peut laisser cette variable vide dans la plupart des cas.

Vous pouvez voir dans [l'annexe](#) (Page 357) un aperçu des erreurs qui se produisent le plus souvent sur les disques durs et les CompactFlash.

Voici quelques exemples sur la façon de charger les pilotes HD dans le fichier de configuration.

Exemple 1 : Accès au disque dur SCSI avec l'Adaptec 2940

```
OPT_HDDRV='yes'                # install Drivers for Harddisk: yes or no
HDDRV_N='1'                    # number of HD drivers
HDDRV_1='aic7xxx'              # various aic7xxx based Adaptec SCSI
HDDRV_1_OPTION=''             # no need for options yet
```

Exemple 2 : Activez l'accès-IDE pour ALIX de PC-Engines

```
OPT_HDDRV='yes'           # install Drivers for Harddisk: yes or no
HDDRV_N='1'               # number of HD drivers
HDDRV_1='pata_amd'        # AMD PCI IDE/ATA driver (e.g. ALIX)
HDDRV_1_OPTION=''         # no need for options yet
```

4.10. HTTPD - Statut du routeur avec le serveur web

4.10.1. OPT_HTTPD - Mini serveur web comme moniteur de statut

Pour ceux qui n'ont pas la possibilité d'utiliser IMONC, pour certaines raisons, par ex. parce qu'il utilise un Mac, ils peuvent utiliser le serveur web pour obtenir ou modifier le statut du routeur fli4l. En activant la variable `OPT_HTTPD='yes'`, vous pouvez utiliser le serveur web et l'écran du statut fli4l.

Pour obtenir la page d'accueil du statut, il faut indiquer dans votre navigateur l'une des adresses suivantes :

```
http://fli4l/
http://fli4l.domain.lan/
http://192.168.6.1/
```

Si votre routeur fli4l a un nom différent, celui-ci doit être utilisé à la place de "fli4l". Cela vaut aussi pour le nom de domaine et aussi pour l'adresse IP. Si vous avez configuré le serveur web sur un autre port (avec la variable `HTTPD_PORT`), vous devez indiquer ceux-ci :

```
http://fli4l:81/
```

Depuis la version 2.1.12 vous pouvez accéder à une page d'accueil pour le login et le mot de passe. Si vous voulez aller directement sur la page d'authentification avec le mot de passe et le login, cette page se trouve dans le sous-répertoire `admin` et vous devez spécifier alors :

```
http://fli4l.domain.lan/admin/
```

Vous pouvez configurer le serveur web avec les variables suivantes :

HTTPD_GUI_LANG Vous pouvez régler avec cette variable la langue, dans laquelle l'interface web doit être affichée. Si vous enregistrez 'auto', le réglage linguistique de la variable `LOCALE` (dans `base.txt`) sera utilisé.

HTTPD_GUI_SKIN Cette variable influe sur l'apparence de l'interface web. On met à disposition avec le paquetage `httpd_skins` deux présentations alternatives de l'interface web, en utilisant les valeurs *fixed* et *notfixed*. La valeur par défaut de cette variable est *default*.

HTTPD_LISTENIP Normalement, le serveur web se connecte sur une adresse Wildcard (ou adresse pouvant prendre n'importe quelle valeur), de sorte qu'il puisse réagir sur n'importe quelle interface du routeur. On peut aussi connecter une seule adresse IP, cela peut être paramétré dans cette variable. Pour l'enregistrement d'une seule adresse IP il faut qu'elle soit indiquée dans : `IP_NET_x_IPADDR`. Normalement, le réglage de la variable doit rester vide, pour que la valeur par défaut puisse fonctionner (sur n'importe quel adresse IP réactif).

Ce paramètre sert uniquement pour le `httpd`, il est associé seulement à une adresse IP, il ne peut pas être utilisé, pour accéder aux différents sous-réseaux de l'interface Web

4. Les paquetages

du routeur, car les accès seront bloqués. Si vous voulez utiliser d'autres adresses IP dans cette variable, vous devez utiliser le filtrage de paquets, pour que l'ensemble des adresses soient reconnus sur le routeur. Il est possible d'indiquer plusieurs adresses IP, en les séparant par un espace.

HTTPD_PORT Si le serveur web doit fonctionner sur un autre port que 80, cette variable doit être adaptée. Normalement cela n'est pas très recommandé, car nous devons alors interroger le serveur web avec `http ://fi4l :81/`

HTTPD_PORTFW Si l'on met cette variable sur 'yes', on peut effectuer des changements sur la transmission de port par l'intermédiaire de l'interface de web. Les règles peuvent être supprimées ou ajoutées, ces modifications entrent en vigueur immédiatement. Mais ces modifications ne sont valables que pour la durée de fonctionnement du routeur. Si le routeur est redémarré, les modifications sont annulées.

Paramètre par défaut 'yes'.

HTTPD_ARPING Le serveur web montre l'état des hôtes en-ligne, ceux-ci sont énumérés dans la variable `HOST_x`. Le serveur utilise le "*cache-Arp*", pour enregistrer provisoirement dans la mémoire les adresses des hôtes locaux. Si un ordinateur n'a pas communiqué depuis longtemps avec le routeur, son adresse IP disparaît du "*cache-Arp*" et l'hôte semble être déconnecté. Vous devez tenir à jour le "*cache-Arp*" (cela empêchera les hôtes qui ne sont pas réellement déconnectés de ne plus être vu), pour cela vous devez activer la variable `HTTPD_ARPING` et mettre la valeur 'yes'.

HTTPD_ARPING_IGNORE_N Dans cette variable vous enregistrez le nombre de arping qui seront ignorés.

HTTPD_ARPING_IGNORE_x Dans cette variable vous indiquez l'adresse IP ou nom de l'hôte qui ne sera pas inclus dans le test arping. Cela peut être utile, pour les hôtes qui utilisent (le wifi du réseau local pour les ordinateurs portables ou les téléphones). Car les paquets de requêtes régulières passant par le réseau consomment plus rapidement la batterie.

4.10.2. Gestion des utilisateurs

Le serveur web offre à l'utilisateur une administration précise :

HTTPD_USER_N Avec cette variable on ajuste, le nombre d'utilisateurs. Si cette variable est placée sur 0, l'administration des utilisateurs est désactivée complètement et tout le monde a la possibilité d'interroger le serveur web.

HTTPD_USER_x_USERNAME HTTPD_USER_x_PASSWORD HTTPD_USER_x_RIGHTS

Avec ces variables on paramètre les différents utilisateurs avec, le nom d'utilisateur, le mot de passe et les droits. On règle Dans la variable `HTTPD_RIGHTS_x` les fonctions pour que chaque utilisateur puisse interroger le serveur web. Dans le cas le plus simple, on paramètre seulement 'all' ce qui signifie que l'utilisateur correspondant peut interroger toutes les fonctions. La variable peut avoir les constructions suivantes :

```
'fonction1:droit1,droit2,... fonction2:...'
```

Au lieu d'indiquer les différents droits pour chaque fonction, la valeur "all" peut être paramétrée, ainsi l'utilisateur aura tous les droits pour chaque fonction. Ci-dessous les fonctions et droit :

Fonction "status" Tout ce qui peut être vue dans le menu statut

view L'utilisateur peut lancer tous les points du menu.

dial L'utilisateur peut décrocher et raccrocher la ligne Tél.

boot L'utilisateur peut arrêter & redémarrer le routeur.

link L'utilisateur peut ajouter ou mettre hors circuit un canal (ISDN)

circuit L'utilisateur peut changer le circuit.

dialmode L'utilisateur peut modifier le mode-dial (Auto, Manual, Off).

contrack L'utilisateur peut voir les connexions actuelles en fonctionnent sur le Routeur.

dyndns L'utilisateur peut voir les informations du paquetage [DYNDNS](#) (Page 110).

Fonction "logs" Tout se que l'on peut faire avec le fichier log (ou fichier journal) (connexion, appels, Syslog)

view L'utilisateur peut voir le fichier journal des événements.

reset L'utilisateur peut supprimer le fichiers journal des événements.

Fonction "support" Tout ce qui est utile, pour chercher des informations par exemple de l'aide dans le Newsgroup.

view L'utilisateur peut lancer les links pour la Doc, pour aller sur à la page Web fli4l, etc.

systeminfo L'utilisateur peut voir les informations sur la configuration et l'état actuel du routeur (par ex. le pare-feu).

Voici quelques exemples :

HTTPD_USER_1_RIGHTS='all' Avec ce paramètre l'utilisateur peut tous faire !

HTTPD_USER_2_RIGHTS='status :view logs :view support :all' Avec ces paramètres l'utilisateur peut tout voir, mais rien modifier.

HTTPD_USER_3_RIGHTS='status :view,dial,link' Avec ces paramètres l'utilisateur peut suivre l'état de la connexion Internet, choisir l'agrégation des canaux (ISDN) ou de l'éteindre.

HTTPD_USER_4_RIGHTS='status :all' Avec ce paramètre l'utilisateur peut tout faire avec les connexions Internet, aussi de redémarrer (et naturellement arrêter le routeur). Cependant il ne peut pas voir les fichiers journal ou de les effacer, il ne peut pas non plus voir le Time-table...

4.10.3. OPT_OAC - Contrôle d'accès en ligne (OAC)

OPT_OAC (Variable optionnelle)

Avec cette variable vous activez le module pour le contrôle d'accès en ligne (OAC), l'accès Internet peut être configuré par rapport aux clients, sélectionnable dans le paquetage [dns_dhcp](#) (Page 90), ainsi les ordinateurs auront une mobilité réduite.

Cet outil existe également en ligne de commande, il permet de contrôler d'autres paquets, par ex. Easycron :

/usr/local/bin/oac.sh

Les options sont affichées avec la commande ci-dessous.

OAC_WANDEVICE (Variable optionnelle)

Avec cette variable vous indiquez le périphérique réseau qui sera utilisé pour restreindre ou verrouiller les accès. Par ex. 'pppoe'

OAC_INPUT (Variable optionnelle)

Avec cette variable vous assurez la protection contre l'environnement via le Proxy.

OAC_INPUT='default' bloque les ports des configurations de : Privoxy, Squid, Tor, SS5, Transproxy.

OAC_INPUT='tcp :8080 tcp :3128' bloque le Port TCP 8080 et 3128. Ici on bloque une liste de Ports avec le protocole qu'il l'accompagne (UDP, TCP), les ports seront séparés par un espace. Si le protocole udp ou tcp n'est pas indiqué, le port ne sera pas conforme.

Vous pouvez omettre cette variable ou indiquer 'no' pour désactiver cette fonction.

OAC_ALL_INVISIBLE (Variable optionnelle)

Avec cette variable on passe sur la vue d'ensemble, pour savoir s'il existe au moins un profil de groupe visible. S'il n'y a pas de profil de groupe visible, alors cette variable n'a aucune action.

OAC_LIMITS (Variable optionnelle)

On indique dans cette variable une limite temps, que vous pouvez choisir dans la liste ci-dessous. Les limites temps sont donnés en minutes. De cette façon, vous pouvez bloquer ou le débloquent temporairement un accès sur le réseau.

Par défaut : '30 60 90 120 180 360 540'

OAC_MODE (Variable optionnelle)

Dans cette variable les valeurs possibles sont : 'DROP' ou 'REJECT' (par défaut)

OAC_GROUP_N (Variable optionnelle)

Dans cette variable vous indiquez le nombre de groupes clients. Pour plus de clarté, vous pouvez également utiliser l'interface web pour créer l'ensemble des groupes, qui seront autoriser ou bloquer.

OAC_GROUP_x_NAME (Variable optionnelle)

Avec cette variable vous indiquez le nom du groupe. Ce nom sera affiché dans l'interface web et sera également utilisable avec le script 'oac.sh' en ligne de commande.

OAC_GROUP_x_BOOTBLOCK (Variable optionnelle)

Si vous indiquez dans cette variable 'yes', tous les clients du groupe seront bloqués lors du boot. En règle générale ces ordinateurs seront verrouillés et pas seulement dans certains cas exceptionnels.

OAC_GROUP_x_INVISIBLE (Variable optionnelle)

Dans cette variable vous pouvez marquer le groupe comme invisible. Si ces ordinateurs sont bloqués à l'avance, ces groupes ne seront pas visibles dans l'interface web.

OAC_GROUP_x_CLIENT_N (Variable optionnelle)

Dans cette variable vous indiquez le nombre de clients dans le groupe.

OAC_GROUP_x_CLIENT_x (Variable optionnelle)

Dans cette variable vous indiquez le nom du client - voir le paquetage [dns_dhcp](#). (Page 90)

OAC_BLOCK_UNKNOWN_IF_x (Variable optionnelle)

Dans cette variable vous indiquez la liste des interfaces définies dans le fichier base.txt, sur lesquels, seuls les hôtes définis dans le fichier (dns_dhcp.txt) pourront accéder à Internet. Les hôtes non définis seront généralement bloqués.

4.11. HWSUPP - Supporte du matériel spécifique

4.11.1. Description

Ce paquetage fournit un support pour l'utilisation de composants et de matériels spécifiques.

Matériels et composants prisent en charges :

- Capteurs de température
- LEDs
- Capteurs de tension
- Vitesse des ventilateurs
- Bouton poussoir
- Watchdog (ou chien de garde)
- Carte VPN

Il prend en charge aussi les systèmes/cartes mère/cartes VPN :

- Matériel pour PC standard
 - LEDs et bouton d'ordinateur
- Matériel ACPI
- Système embarqué
 - AEWIN SCB6971
 - Fujitsu Siemens Futro S200
 - PC Engines ALIX
 - PC Engines APU
 - PC Engines WRAP
 - Soekris net4801
 - Soekris net5501
- Carte mère
 - Commell LE-575
 - GigaByte GA-M521-S3
 - LEX CV860A
 - SuperMicro PDSME
 - SuperMicro X7SLA
 - Tyan S5112
 - WinNet PC640
 - WinNet PC680
- Carte VPN (PCI, miniPCI et miniPCIE)
 - vpn1401 vpn1411

4.11.2. Configuration du paquetage HWSUPP

La configuration se fait comme les autres paquetages fli4l, en paramétrant le fichier `Pfad/fli4l-3.10.4/<config>/hwsupp.txt` selon votre propre configuration.

OPT_HWSUPP La valeur 'no' dans cette variable désactive complètement le paquetage OPT_HWSUPP. Il n'y aura aucun changement sur le support de boot de l'archive fli4l rootfs.img n'y dans l'archive opt.img. Pour finir OPT_HWSUPP n'écasse aucune partie de l'installation fli4l.

Pour activer la variable OPT_HWSUPP du paquetage OPT_HWSUPP vous devez placer la valeur sur 'yes'.

HWSUPP_TYPE Dans cette variable vous indiquez le matériel à configurer. Les valeurs suivantes sont disponibles :

- sim
- generic-pc
- generic-acpi
- aewin-scb6971
- commell-le575
- fsc-futro-s200
- gigabyte-ga-m52l-s3
- lex-cv860a
- pcengines-alix
- pcengines-apu
- pcengines-wrap
- soekris-net4801
- soekris-net5501
- supermicro-pdsme
- supermicro-x7sla
- tyan-s5112
- winnet-pc640
- winnet-pc680

HWSUPP_WATCHDOG La valeur 'yes' dans cette variable active le démon Watchdog, si le matériel sélectionné est équipé du Watchdog. le Watchdog redémarre automatiquement le système qui a été momentanément arrêté pour une raison quelconque.

HWSUPP_CPUFREQ Si vous avez indiqué 'yes' dans cette variable, vous pouvez gérer la fréquence du processeur en fonction de la demande en ressources du système et des applications.

HWSUPP_CPUFREQ_GOVERNOR Dans cette variable vous sélectionnez le gouverneur pour la fréquence du processeur. Le gouverneur sélectionné contrôle le comportement et le réglage de la fréquence. Vous pouvez sélectionner l'un d'entre eux :

- performance
Le CPU fonctionne toujours avec la fréquence la plus élevée.
- ondemand
La fréquence du CPU sera ajustée en fonction de l'utilisation du CPU. La fréquence peut changer très rapidement.
- conservative
La fréquence du CPU sera ajustée en fonction de l'utilisation du CPU. La fréquence est modifiée étape par étape.
- powersave
Le CPU fonctionne toujours avec la fréquence la plus basse.
- userspace
La fréquence du CPU peut être réglée manuellement ou par un script utilisateur via la variable sysfs dans /devices/system/cpu/cpu<n>/cpufreq/scaling_setspeed.

HWSUPP_LED_N Vous indiquez dans cette variable le nombre de LEDs, les appareils sont différents et peuvent avoir un nombre de LED différente.

HWSUPP_LED_x Vous indiquez ici la valeur qui sera affichée par la LED, les valeurs suivantes sont disponibles :

4. Les paquetages

- ready - Le routeur fli4l est prêt
- online - Le routeur fli4l est connecté à l'Internet
- trigger - La LED est commandée par un déclencheur du kernel
- user - L'affichage est commandé par un script utilisateur

La liste des valeurs peut être prolongée en utilisant d'autres paquetages. Si vous chargez le paquetage WLAN l'affichage pour le sans fil sera possible.

- wlan - Le WLAN est activé

Pour créer une telle extension vous avez dans l'annexe B.10 des conseils pour les développeurs de paquetages.

HWSUPP_LED_x_DEVICE Dans cette variable vous indiquez un périphérique pour la LED du matériel utilisé. Soit vous paramétrez un périphérique pour la LED qui se trouve dans le répertoire `/sys/class/leds/` du routeur, soit vous paramétrez un nombre GPIO ⁷

Une liste de noms des périphériques pour la LED peut être trouvées dans la documentation [annexe](#), selon le matériel spécifique dans `HSUPP_TYPE`.

Si vous indiquez un nombre GPIO il doit être indiqué dans le format `gpio::x`. Si vous avez indiqué un GPIO, le périphérique correspondant à la LED sera créé automatiquement. Si vous placez le caractère `/` devant GPIO, son fonctionnement sera inversé.

Exemple :

```
HWSUPP_LED_1_DEVICE='apu::1'      # LED 1 sur PC engines APU
HWSUPP_LED_2_DEVICE='gpio::237'    # GPIO 237
HWSUPP_LED_3_DEVICE='/gpio::245'   # GPIO 245 inversé
```

HWSUPP_LED_x_PARAM Dans cette variable vous indiquez les paramètres pour l'affichage de la LED.

Selon la valeur de la variable `HWSUPP_LED_x`, la variable `HWSUPP_LED_x_PARAM` aura une signification différente.

Si vous avez paramétré la variable `HWSUPP_LED_x='trigger'`, avec la valeur `trigger` (ou déclencheur), le contrôle d'activation de la LED doit être paramétré dans la variable `HWSUPP_LED_x_PARAM`.

Dans cette variable vous définissez le trigger (ou déclencheur) qui commande la LED. Les déclencheurs disponibles peuvent être affichés avec la commande `cat /sys/class/leds/*/trigger`.

Parmi les déclencheurs créés, par exemple `netfilter` ou les pilotes de matériels comme `ath9k`, d'autres modules de déclencheurs peuvent être chargés via la variable `HWSUPP_DRIVER_x`.

Exemple :

```
HWSUPP_LED_1='trigger'
HWSUPP_LED_1_TRIGGER='heartbeat'
HWSUPP_LED_2='trigger'
HWSUPP_LED_2_TRIGGER='netfilter-ssh'
```

6. Si vous avez paramétré la variable `HWSUPP_LED_x='ready'` avec cette valeur `ready`, la progression du boot sera indiquée par une séquence de clignotement de la LED (voir [appendix B.9](#)).

7. GPIO (General Purpose Input/Output, c'est-à-dire entrée/sortie pour un usage général) le nombre correspond à la position physique de la broche sur le circuit intégré, le comportement de la broche peut être programmé au moment de son exécution, notamment s'il s'agit d'une entrée ou d'une sortie.

4. Les paquetages

Si dans la variable 'HWSUPP_LED_x' vous avez indiqué la valeur 'user', vous devez indiquer dans la variable HWSUPP_LED_PARAM le nom du script pour la LED ainsi que le chemin de celui-ci.

Exemple :

```
HWSUPP_LED_1='user'
HWSUPP_LED_1_PARAM='/usr/local/bin/myledscript'
```

Si vous avez indiqué dans la variable la valeur HWSUPP_LED_x='wlan', vous devez indiquer dans la variable HWSUPP_LED_x_PARAM le nom du périphérique.

Vous pouvez indiquer dans cette variable un ou plusieurs périphériques WLAN (ou sans fil). Si vous avez indiqué plusieurs périphériques WLAN ils seront séparés par un espace. Si vous avez indiqué plusieurs WLAN et si vous avez activé la LED, elle aura la signification suivante :

- Éteinte - Tous les périphériques sans fil sont inactifs
- Clignotante - Une partie des périphériques sans fil est actif
- Allumée - Tous les périphériques sans fil sont actifs

Exemple :

```
HWSUPP_LED_1='wlan'
HWSUPP_LED_1_WLAN='wlan0 wlan1'
```

HWSUPP_BOOT_LED Dans cette variable vous pouvez indiquer la séquence de clignotement de la LED par rapport à la progression du boot.

Si vous avez indiqué dans la variable la valeur HWSUPP_LED_x='ready', la variable HWSUPP_BOOT_LED sera ignorée.

HWSUPP_BUTTON_N Dans cette variable vous indiquez le nombre de boutons. Le nombre de boutons dépend du matériel utilisé.

HWSUPP_BUTTON_x Dans cette variable vous définissez l'action qui doit être exécutée lorsque vous pressez le bouton. Les actions suivantes sont possibles :

- reset - Réinitialise le routeur fli4l
- online - Active ou désactive la connexion Internet
- user - Le script utilisateur sera exécutée

La liste des valeurs peut être prolongée en utilisant d'autres paquetages. Si vous chargez le paquetage WLAN une action sur le sans fil sera possible.

- wlan - Active ou désactive le sans fil (ou WLAN)

HWSUPP_BUTTON_x_DEVICE Dans cette variable vous indiquez le périphérique pour le bouton.

Si vous indiquez un nombre GPIO, il doit être dans le format `gpio::x`. Si vous placez le caractère / devant GPIO, son fonctionnement sera inversé.

Une liste GPIO prédéfinie peut être trouvée dans la documentation annexe [B.7.2](#), selon le matériel spécifique dans HWSUPP_TYPE.

Exemples :

```
HWSUPP_BUTTON_1_DEVICE='gpio::252'
HWSUPP_BUTTON_2_DEVICE='/gpio::237'
```

HWSUPP_BUTTON_x_PARAM Dans cette variable vous indiquez les paramètres pour l'action du bouton.

Selon la valeur de la variable `HWSUPP_BUTTON_x`, la variable `HWSUPP_BUTTON_x_PARAM` aura une signification différente.

Si vous avez indiquez dans la variable la valeur `HWSUPP_BUTTON_x='user'`,

Vous devez placez dans `HWSUPP_BUTTON_x_PARAM` le nom du fichier script à exécuter lorsque le bouton sera enfoncé.

Exemple :

```
HWSUPP_BUTTON_1='user'
HWSUPP_BUTTON_2_WLAN='/usr/local/bin/myscript'
```

Si vous avez indiquer dans la variable la valeur `HWSUPP_BUTTON_x_ACTION='wlan'`, vous devez indiquer dans la variable `HWSUPP_BUTTON_x_PARAM` le nom du périphérique sans fil.

Si dans cette variable vous indiquez un ou plusieurs périphériques WLAN, ils seront surveillés par le système. Si vous indiquez plusieurs périphériques sans fil, vous devez les séparés par un espace. Grâce à la pression sur la bouton le périphérique sera activé ou désactivé.

Exemple :

```
HWSUPP_BUTTON_2='wlan'
HWSUPP_BUTTON_2_WLAN='wlan0 wlan1'
```

4.11.3. Paramètre pour expert

Les paramètres suivants doivent être utilisés uniquement lorsque vous savez exactement – quel matériel que vous avez et quel pilote supplémentaire est nécessaires. – l'adresse et le type de périphérique I²C⁸.

En activant les paramètres experts, vous aurez un message d'avertissement lors de la construction de `fli4l` avec la commande `mkfli4l`.

HWSUPP_DRIVER_N Dans cette variable vous indiquez le nombre de pilotes supplémentaires. Les pilotes dans la variable `HWSUPP_DRIVER_x` seront chargés dans l'ordre enregistré.

HWSUPP_DRIVER_x Dans cette variable vous indiquez le nom du pilote (sans l'extension du fichier `.ko`).

Exemple :

```
HWSUPP_DRIVER_N='2'
HWSUPP_DRIVER_1='i2c-piix4'      # pilote du bus I2C
HWSUPP_DRIVER_2='gpio-pcf857x'  # I2C extension GPIO
```

HWSUPP_I2C_N Dans cette variable vous indiquez le nombre de périphérique I²C à charger.

I²C ne supporte pas du tout le mécanisme PnP. Par conséquent, vous devez avoir un numéro de bus pour chaque périphérique I²C, l'adresse du périphérique et le type de périphérique pour la configuration.

8. Un bus I²C ou un SMBus est un bus série sur PC, il est utilisé par exemple pour la lecture de la température avec un capteur. Dans de nombreux cas, le bus I²C ou le SMBus est disponible par l'intermédiaire un connecteur à broche il peut être utilisé pour une extension de matériel personnel.

HWSUPP_I2C_x_BUS Dans cette variable vous indiquez le numéro de bus I²C donc le périphérique sera associé.

Le numéro de bus doit être saisi dans le format suivant `i2c-x`.

HWSUPP_I2C_x_ADDRESS Dans cette variable vous indiquez l'adresse du périphérique I²C.

L'adresse doit être indiquée comme un nombre hexadécimal et dans la plage de 0x03 à 0x77.

HWSUPP_I2C_x_DEVICE Dans cette variable vous indiquez le type de périphérique I²C qui sera supporté par le pilote précédemment chargé.

Exemple :

```
HWSUPP_I2C_N='1'
HWSUPP_I2C_1_BUS='i2c-1'
HWSUPP_I2C_1_ADDRESS='0x38'
HWSUPP_I2C_1_DEVICE='pcf8574a' # supporté par le pilote gpio-pcf857x
```

4.11.4. Prise en charge des cartes VPN

OPT_VPN_CARD La valeur 'no' dans cette variable désactive complètement le paquetage OPT_VPN_CARD. Il n'y aura aucun changement sur le support de boot de l'archive `fli4l rootfs.img` n'y dans l'archive `opt.img`. Pour finir OPT_VPN_CARD n'écrase aucune partie de l'installation `fli4l`.

Pour activer la variable OPT_VPN_CARD du paquetage OPT_VPN_CARD vous devez placer la valeur sur 'yes'.

VPN_CARD_TYPE Dans cette variable vous pouvez indiquer l'accélérateur VPN. La carte VPN décharge le CPU des tâches informatiques intensives de cryptage et de la fonction de hachage. Les valeurs suivantes sont disponibles :

- `hifn7751` - Pour carte Soekris `vpn1401` et `vpn1411`
- `hifnhipp`

4.12. IPv6 - Internet protocole version 6

4.12.1. Introduction

Ce paquetage permet au routeur `fli4l` avec bien des égards de rendre compatible l'IPv6. Les informations qui sont incluses dans le paquetage IPv6 pour le routeur sont les adresses IPv6, la gestion des (sous-)réseaux IPv6, la route IPv6 prédéfinie et les règles de pare-feu. Vous pouvez aussi configurer le service IPv6 par le DHCPv6. Enfin, il est possible de construire un tunnel automatiquement avec des fournisseurs IPv6. Maintenant cela fonctionne correctement, mais, seulement avec des tunnels 6in4, le fournisseur "SixXS" prend en charge cette technologie. Les autres technologies comme (AYIYA, 6to4, Teredo) ne sont pas encore prises en charge.

IPv6 est le successeur du protocole Internet IPv4. Il a été principalement conçu pour augmenter la quantité relativement faible des adresses Internet formelles : IPv4 supporte environ 2³² d'adresses,⁹ avec IPv6 on a déjà 2¹²⁸ d'adresses. Avec la communication IPv6, on peut

9. c'est seulement approximatif, car certaines adresses ont un objectif bien spécifique, comme le Broadcasting et le Multicasting,

attribuer une adresse unique pour chaque hôte, et nous ne sommes plus sur des techniques telles que le NAT, le PAT, le Masquerading, etc.

Outre cet aspect, les sujets comme l'autoconfiguration et la sécurité ont aussi joué un rôle lors du développement du protocole IPv6. Ces questions seront traitées dans les sections suivantes.

Le plus gros problème avec IPv6 est sa distribution : Actuellement, l'IPv6 – par rapport à IPv4 – est très peu utilisés. La raison est que le protocole IPv6 et IPv4 ne sont pas techniquement compatibles l'un avec l'autre et par conséquent tous les composants matériels et logiciels, qui sont impliqués dans la transmission de paquets sur Internet pour l'IPv6 doit être installé. Certains services comme le DNS (Domain Name System) pour IPv6 doivent être ouverts en conséquence.

Un cercle vicieux s'ouvre alors : la faible propagation des IPv6 chez les fournisseurs d'accès Internet amène l'indifférence de la part des fabricants à équiper les routeurs d'un dispositif pour le fonctionnement IPv6, cela signifie que les fournisseurs d'accès ont peur de la transition vers IPv6, parce qu'ils craignent qu'un tel effort ne vos pas la peine. Ce n'est que lentement que le vent tourne en faveur de l'IPv6, car des réserves d'adresses IPv4 s'épuisent.¹⁰

4.12.2. Format de l'adresse

Une adresse IPv6 se compose de huit valeurs de deux octets, elles sont classées en hexadécimal :

Exemple 1 : 2001:db8:900:551:0:0:0:2

Exemple 2 : 0:0:0:0:0:0:0:1 (IPv6-Loopback-Adresse)

Pour réduire l'encombrement des adresses, on peut fusionner une suite de zéros successifs, en les supprimant et en ajoutant seulement une paire de deux points. Les adresses ci-dessus peuvent également être écrites comme ceci :

Exemple 1 (compacté) : 2001:db8:900:551::2

Exemple 2 (compacté) : ::1

Une telle réduction est uniquement autorisée d'une fois, pour éviter toute ambiguïté. L'adresse 2001:0:0:1:2:0:0:3 peut être réduite comme ceci 2001::1:2:0:0:3 ou 2001:0:0:1:2::3, mais pas comme ceci 2001::1:2::3, parce que, il serait maintenant difficile de savoir comment les quatre zéros doivent être répartis sur les zones de réductions.

Une autre ambiguïté existe, si une adresse IPv6 doit être combinée avec un port (TCP ou UDP) : dans ce cas, il ne faut pas joindre le port directement avec les deux-points à l'adresse, parce que ces deux-points seront intégrés à l'intérieur de l'adresse et donc dans certains cas, il serait difficile de savoir si la spécification du port est peut-être ou pas un composant de l'adresse. Il faut donc, dans ce cas mettre l'adresse IPv6 entre deux crochets. Cette syntaxe est demandée dans les URL (par exemple lorsque l'utilisation doit indiquer une adresse IPv6 au format numérique dans le navigateur Web).

Exemple 3 : [2001:db8:900:551::2]:1234

Voici l'adresse sans mettre les crochets 2001:db8:900:551::2:1234, correspond à l'adresse intégrale 2001:db8:900:551:0:0:2:1234 vous voyez quelle ne possède aucune indication de port.

10. Maintenant les derniers blocs d'adresses IPv4 ont été attribués par l'IANA.

4.12.3. Configuration

Paramètres généraux

Les paramètres généraux contiennent d'abord, l'activation du support IPv6, d'autre part l'attribution optionnelle d'une adresse IPv6 sur le routeur.

OPT_IPV6 Avec cette variable, vous pouvez activer le support IPv6.

Configuration par défaut : `OPT_IPV6='no'`

HOSTNAME_IP6 (optionnelle) Cette variable règle explicitement l'adresse IPv6 du routeur.

Si la variable n'est pas définie, l'adresse IPv6 est placée sur la configuration de la première adresse du sous-réseau IPv6 (`IPV6_NET_x`, voir ci-dessous).

Exemple : `HOSTNAME_IP6='IPV6_NET_1_IPADDR'`

Configuration du sous-réseau

Dans ce paragraphe, nous allons décrire la configuration d'un ou plusieurs sous-réseaux IPv6. Un sous-réseau IPv6 est une adresse IPv6 étendue qui est spécifiée par un préfixe et qui est liée à une interface réseau spécifique. Les autres paramètres concernent l'édition du préfixe et le service DNS dans le sous-réseau, ainsi que le nom du routeur optionnel à l'intérieur du sous-réseau.

IPV6_NET_N Dans cette variable, vous indiquez le nombre de sous-réseaux IPv6 à utiliser.

Vous devez définir Au moins un sous-réseau IPv6, pour utiliser l'IPv6 dans le réseau local.

Configuration par défaut : `IPV6_NET_N='0'`

IPV6_NET_x Dans cette variable, vous indiquez l'adresse IPv6, contenu dans le sous-réseau IPv6 du routeur, ainsi que la taille du masque de sous-réseau en utilisant la notation CIDR. Si le sous-réseau est un routage public, il provient en générale d'Internet ou d'un prestataire de tunnel.

Important: *Si vous activez la configuration automatique sans-état dans le même sous-réseau (voir la section `IPV6_NET_x_ADVERTISE` ci-dessous), la longueur du préfixe du sous-réseau doit faire 64 bits !*

Important: *Si le sous-réseau est connecté à un tunnel (voir `IPV6_NET_x_TUNNEL` ci-dessous), vous devez indiquer seulement une partie de l'adresse du routeur, mais pas le préfixe du sous-réseau associé au tunnel (qui se trouve dans `IPV6_TUNNEL_x_PREFIX`), avec ce préfixe, l'adresse pourra être combiné ! Dans la version précédente du paquetage IPv6, la variable `IPV6_TUNNEL_x_PREFIX` n'existait pas, le préfixe et le sous-réseau de l'adresse du routeur étaient ensemble dans la variable `IPV6_NET_x`. Toutefois, cela ne s'applique pas si le préfixe du sous-réseau est assigné dynamiquement par le fournisseur à la construction du tunnel. De plus, la longueur du préfixe du sous-réseau (dans ce cas : /48) est cachée, si bien que le routage prédéfini ne peut pas être correctement réglé et que la route vers les destinations spécifiques conduit alors à des effets étranges.*

Exemples :

```
IPV6_NET_1='2001:db8:1743:42::1/64'      # sans Tunnel~: adresse complète
IPV6_NET_1_TUNNEL=''
```

```
IPV6_NET_2='0:0:0:42::1/64'           # avec Tunnel~: adresse partielle
IPV6_NET_2_TUNNEL='1'
IPV6_TUNNEL_1_PREFIX='2001:db8:1743::/48' # voir section "configuration du Tunnel"
```

IPV6_NET_x_DEV Avec cette variable, vous indiquez le nom de l'interface du sous-réseau IPv6 sur laquelle l'adresse IPv6 sera associée. Cette interface réseau n'entre *pas* en collision avec l'interface réseau qui a été attribuée dans la configuration de base (**base.txt**), les deux adresses IPv4 et IPv6 pourront être affectées sur cette interface réseau.

Exemple : `IPV6_NET_1_DEV='eth0'`

IPV6_NET_x_TUNNEL Dans cette variable, vous indiquez un sous-réseau IPv6 spécifique, à l'index du tunnel. Le préfixe du tunnel spécifié sera combiné avec l'adresse du routeur pour obtenir l'adresse IPv6 complète pour le routeur. Si la variable est vide ou non définie, aucun sous-réseau ne fera partie du tunnel, dans la variable `IPV6_NET_x` vous devez indiquer une 'adresse IPv6 complète pour le routeur, y compris le masque de réseau (voir plus haut).

Un tunnel peut être attribué à plusieurs sous-réseaux, le préfixe du sous-réseau du tunnel est généralement assez grand pour qu'il puisse être divisé en plusieurs sous-réseaux (/56 ou plus). Bien sûr, ce n'est pas possible dans l'autre sens, attribuer un sous-réseau à plusieurs préfixes du sous-réseau du tunnel, car l'adresse du sous-réseau serait ambiguë.

Exemple : `IPV6_NET_1_TUNNEL='1'`

IPV6_NET_x_ADVERTISE Avec cette variable, vous déterminez si le préfixe du sous-réseau sera distribué par "l'intermédiaire du routeur" dans le LAN. Cela est utilisé pour une "stateless autoconfiguration" (ou configuration automatique sans état) et ne doit pas être confondu avec le DHCPv6. Les valeurs possibles sont "yes" ou "no".

Il est recommandé d'activer ce paramètre, à moins que toutes les adresses dans le réseau soient affectées statiquement ou qu'un autre routeur est déjà compétent pour notifier le préfixe du sous-réseau.

Important: *La distribution automatique des sous-réseaux fonctionne seulement si le sous-réseau est un réseau /64, c.-à-d., si la longueur du préfixe du sous-réseau est de 64 bits ! La raison est que les hôtes du réseau calculent l'adresse IPv6 à partir du préfixe et de leur adresse MAC, si l'hôte ne partage pas les 64 bits cela ne fonctionne pas. Si la configuration automatique échoue, il faut vérifier le préfixe du sous-réseau, il a peut-être été spécifié de manière incorrecte (par exemple /48).*

Configuration par défaut : `IPV6_NET_1_ADVERTISE='yes'`

IPV6_NET_x_ADVERTISE_DNS Avec cette variable vous déterminez si le service DNS local sur le sous-réseau IPv6 sera distribué par "l'intermédiaire du routeur". Cela ne fonctionne que si la fonction IPv6 du service DNS est activé par le biais de la variable `DNS_SUPPORT_IPV6='yes'`. Les valeurs possibles sont "yes" ou "no".

Configuration par défaut : `IPV6_NET_1_ADVERTISE_DNS='no'`

IPV6_NET_x_DHCP Avec cette variable, vous activez le service DHCPv6 pour le sous-réseau IPv6. Les valeurs possibles sont "yes" ou "no". Le DHCPv6 est utilisé ici uniquement pour permettre aux hôtes du sous-réseau d'obtenir des informations sur le nom de domaine et l'adresse du serveur DNS à utiliser. Actuellement l'attribution d'adresse IPv6 via le DHCPv6 n'est pas possible avec fl4l.

L'adresse du serveur DNS ne sera pas publiée par le DHCPv6, si le support IPv6 du service DNS via la variable `DNS_SUPPORT_IPV6` dans le paquetage `dns_dhcp` n'est pas activé.

Important: *La variable `IPV6_NET_x_ADVERTISE_DNS` et `IPV6_NET_x_DHCP` ne sont pas mutuellement exclusif, mais les deux peuvent être activés. Dans ce cas, l'adresse du serveur DNS peut être attribuée de deux manières différentes sur l'hôte du réseau local.*

Un sous-réseau IPv6 au maximum peut être attaché à une interface réseau, pour configurer le DHCPv6 !

Configuration par défaut : `IPV6_NET_1_DHCP='no'`

IPV6_NET_x_NAME (optionnelle) Dans cette variable, vous pouvez paramétrer un nom d'hôte spécifique pour chaque interface du sous-réseau IPv6 du routeur.

Exemple : `IPV6_NET_1_NAME='fli41-subnet1'`

Configuration d'un Tunnel

Dans ce paragraphe nous allons présenter la configuration d'un tunnel IPv6-6in4. Un tel tunnel est utile lorsque votre propre fournisseur d'accès Internet ne supporte pas l'IPv6 par défaut. Ainsi, nous pouvons faire un tunnel-broker avec un hôte bien précis sur Internet, avec le soi-disant PoP (Point of Presence), il faut construire une connexion bidirectionnelle via IPv4, les paquets IPv6 seront ensuite empaquetés et acheminés (d'où 6 "in" 4 parce que les paquets IPv6 sont encapsulés dans les paquets IPv4).¹¹ Pour que le tunnel fonctionne, il faut configurer les routeurs avec le paquetage IPv6 des deux côtés de la connexion Internet. Le premier paragraphe décrit la configuration, le deuxième paragraphe décrit la connexion.

IPV6_TUNNEL_N Avec cette variable vous indiquez le nombre de tunnels 6in4 à mettre en place.

Exemple : `IPV6_TUNNEL_N='1'`

IPV6_TUNNEL_x_TYPE Avec cette variable, vous déterminez le type de tunnel. Actuellement, les valeurs possibles sont : "raw" pour un tunnel qui envoie des paquets "brut", "static" pour un tunnel statique, "sixxs" pour un tunnel Heartbeat dynamique du fournisseur SixXS et "he" pour un tunnel du fournisseur Hurricane Electric. Au sujet du tunnel Heartbeat voir le paragraphe plus bas.

Exemple : `IPV6_TUNNEL_1_TYPE='sixxs'`

IPV6_TUNNEL_x_DEFAULT Avec cette variable, vous déterminez si les paquets IPv6 qui ne sont pas adressés au niveau local ou aux réseaux locaux, doivent être routés sur un autre tunnel. Il ne peut y avoir qu'un seul tunnel (parce que seulement une route par défaut peut exister). Les valeurs possibles sont "yes" ou "no".

Important: *le tunnel doit exactement être une passerelle par défaut pour les données IPv6 sortantes, car la communication avec des hôtes IPv6 ne serait pas possible autrement sur Internet ! L'utilisation exclusive du tunnel pas par défaut, n'est utile que si le trafic IPv6 sortant est envoyé via une route par défaut configurée séparément et qui n'est pas en rapport avec un tunnel. Voir l'introduction du paragraphe "configuration de route" et aussi la description de la variable `IPV6_ROUTE_x` ci-dessous.*

Configuration par défaut : `IPV6_TUNNEL_1_DEFAULT='no'`

11. Il s'agit de l'IPv4 protocole 41, "encapsulation IPv6".

IPV6_TUNNEL_x_PREFIX Avec cette variable, vous indiquez le préfixe IPv6 du sous-réseau du tunnel dans la notation CIDR, c.-à-d. que vous indiquez la longueur du préfixe, mais aussi l'adresse IPv6. Cette information est précisée dans la convention du fournisseur de tunnel. En ce qui concerne certains fournisseurs de tunnel, si le préfixe est réaffecté à chaque construction du tunnel, alors cette information sera inutile. (Actuellement, de tels fournisseurs ne sont pas supportés).

Important: *Cette variable peut restée vide, si le tunnel n'a pas de préfixe de sous-réseau attribué. Toutefois, ce tunnel ne peut pas être affecté à un sous-réseau IPv6 par la variable (IPV6_NET_x), parce que les adresses IPv6 dans le sous-réseau ne peuvent pas être calculées. Il est logique d'une telle configuration ne soit que provisoire, en attendant l'activation du tunnel et avant que le fournisseur de tunnel attribue un préfixe du sous-réseau (cette procédure est par ex. courant chez le fournisseur de tunnel SixXS).*

Exemples :

```
IPV6_TUNNEL_1_PREFIX='2001:db8:1743::/48'      # /48-sous-réseau
IPV6_TUNNEL_2_PREFIX='2001:db8:1743:5e00::/56'   # /56-sous-réseau
```

IPV6_TUNNEL_x_LOCALV4 Dans cette variable, vous indiquez l'adresse IPv4 locale du tunnel ou le paramètre 'dynamic' si l'adresse IPv4 est allouée dynamiquement par le circuit WAN actif. S'il s'agit d'un tunnel Heartbeat (voir IPV6_TUNNEL_x_TYPE ci-dessus).

Exemple :

```
IPV6_TUNNEL_1_LOCALV4='172.16.0.2'
IPV6_TUNNEL_2_LOCALV4='dynamic'
```

IPV6_TUNNEL_x_REMOTEV4 Dans cette variable, vous indiquez l'adresse IPv4 distant du tunnel. Cette information est habituellement déterminée par le fournisseur du tunnel. Exemple (Correspond au PoP deham01 d'Easynet) :

```
IPV6_TUNNEL_1_REMOTEV4='212.224.0.188'
```

Important: *Si la variable PF_INPUT_ACCEPT_DEF est sur "no", c.-à-d que le pare-feu IPv4 est configuré manuellement, une règle est nécessaire pour accepter tous les paquets IPv6-in-IPv4 (Protocole-IP 41) de l'extrémité du tunnel. Surnommé point d'arrêt du tunnel, la règle correspondante est indiqué ci-dessous :*

```
PF_INPUT_x='prot:41 212.224.0.188 ACCEPT'
```

IPV6_TUNNEL_x_LOCALV6 Dans cette variable, vous indiquez l'adresse IPv6 local du tunnel avec le masque de sous-réseau, en utilisant la notation CIDR. Cette information est donnée par le fournisseur d'accès du tunnel. Lors d'une nouvelle configuration du tunnel, les fournisseurs de tunnel l'attribuent à chaque extrémité du tunnel. Cette information est inutile, (actuellement les fournisseurs ne supportent pas encore cette fonction).

Exemple : IPV6_TUNNEL_1_LOCALV6='2001:db8:1743::2/112'

IPV6_TUNNEL_x_REMOTEV6 Dans cette variable, vous indiquez l'adresse IPv6 distante du tunnel. Cette information est donnée par le fournisseur d'accès du tunnel. Le masque de sous-réseau n'est pas nécessaire, car il est récupéré dans La variable IPV6_TUNNEL_x_LOCALV6. Lors d'une nouvelle configuration du tunnel, les fournisseurs de tunnel l'attribuent à

4. Les paquetages

chaque extrémité du tunnel. Cette information est inutile, (actuellement les fournisseurs ne supportent pas encore cette fonction).

Exemple : `IPV6_TUNNEL_1_REMOTEV6='2001:db8:1743::1'`

IPV6_TUNNEL_x_DEV (optionnelle) Dans cette variable, vous indiquez le nom de l'interface réseau du tunnel à produire. Si vous avez plusieurs tunnels, ils doivent être nommés différemment, de sorte que tout fonctionne. Si la variable n'est pas définie, un nom pour le tunnel sera généré automatiquement ("v6tun" + index Tunnel).

Exemple : `IPV6_TUNNEL_1_DEV='6in4'`

IPV6_TUNNEL_x_MTU (optionnelle) Dans cette variable, vous indiquez la taille du MTU (Maximum Transfert Unit) en octets, c.-à-d. le plus grand paquet qui peut être envoyé sur le tunnel. en règle général cette information est précisée par le fournisseur de tunnel. Le réglage par défaut si non spécifié est de "1280", il doit être compatible avec tous les tunnels.

Configuration par défaut : `IPV6_TUNNEL_1_MTU='1280'`

Certains fournisseurs de tunnel exigent un signe de vie qui soit en permanence envoyée sur le routeur du fournisseur de tunnel, pour s'assurer que l'hôte sollicite le tunnel, bien que celui-ci n'est pas utilisé. En plus le soi-disant protocole Heartbeat ("battement de coeur") est utilisé. Les fournisseurs exigent généralement une ouverture de session réussie avec identifiant et mot de passe pour empêcher les abus. Si vous utilisez un tunnel Heartbeat (comme il est proposé avec SixXS), alors les informations appropriées doivent être renregistré, elles sont décrites plus bas.

IPV6_TUNNEL_x_USERID Dans cette variable, vous indiquez le nom d'utilisateur, nécessaires pour la connexion au tunnel.

Exemple : `IPV6_TUNNEL_1_USERID='ABCDE-SIXXS'`

IPV6_TUNNEL_x_PASSWORD Dans cette variable, vous indiquez le mot de passe pour le nom d'utilisateur spécifié ci-dessus. Il ne doit pas contenir d'espaces.

Exemple : `IPV6_TUNNEL_1_PASSWORD='password'`

IPV6_TUNNEL_x_TUNNELID Dans cette variable, vous indiquez, l'indénification du tunnel. Le nom du tunnel SixXS commence toujours avec un grand "T".

Exemple : `IPV6_TUNNEL_1_TUNNELID='T1234'`

IPV6_TUNNEL_x_TIMEOUT (optionnelle) Dans cette variable, vous indiquez le temps d'attente en seconde, avant la construction du tunnel. La valeur par défaut dépend du fournisseur d'accès du tunnel.

Exemple : `IPV6_TUNNEL_1_TIMEOUT='30'`

Configuration des routes

Les routes sont des chemins pour rediriger les paquets IPv6. Cela signifie que le routeur doit savoir où envoyer les paquets entrants, il s'appuie sur une table de routage pour trouver exactement les informations. Pour les paquets IPv6, il est important de savoir où sont envoyés les paquets qui ne font pas partie du réseau local. Pour cela, une route par défaut doit être configurée pour envoyer tous les paquets à l'autre extrémité du tunnel IPv6. Vous pouvez également ajouter d'autres routes qui relient les sous-réseaux IPv6 les uns aux autres.

IPV6_ROUTE_N Dans cette variable vous indiquez le nombre de routes IPv6. En général, aucune route supplémentaire IPv6 n'est nécessaire.

Configuration par défaut : `IPV6_ROUTE_N='0'`

IPV6_ROUTE_x Dans cette variable, vous indiquez la route sous la forme 'Réseau de destination Passerelle', le réseau de destination est écrit en utilisant la notation CIDR. Vous devez indiquer `::/0` pour la route par défaut du réseau de destination. Cependant, il n'est pas nécessaire de configurer la route par défaut qui passe par le tunnel (voir l'introduction de ce paragraphe).

Exemple : `IPV6_ROUTE_1='2001:db8:1743:44::/64_2001:db8:1743:44::1'`

IPv6-Firewall

Comme pour les réseaux IPv4, les réseaux IPv6 ont besoin d'un pare-feu, ainsi le monde extérieur ne pourra pas joindre les ordinateurs du réseau local. Cela est d'autant plus important, car chaque ordinateur est remplacé dans le cas normal, d'une adresse IPv6 unique, cette adresse qui peut être affectée à l'ordinateur de façon permanente, car elle est basée sur l'adresse MAC de la carte d'interface réseau.¹² Par conséquent, le pare-feu interdira toute demande provenant de l'extérieur, dans ce paragraphe vous allez voir comment ouvrir les entrées correspondantes petit à petit – selon vos besoins –.

La configuration du pare-feu IPv6, correspond grosso modo à la configuration du pare-feu IPv4. Les différences particulières seront examinées séparément.

PF6_LOG_LEVEL La configuration du système de journalisation dans la variable `PF6_LOG_LEVEL` est utilisée pour toutes les chaînes ci-dessous sans distinction, leur contenu peut être réglé sur l'une des valeurs suivantes : debug, info, notice, warning, err, crit, alert, emerg.

PF6_INPUT_POLICY Cette variable définit la politique par défaut pour les paquets entrants sur le routeur avec la (chaîne INPUT). Les valeurs possibles sont "REJECT" (par défaut, rejette tous les paquets), "DROP" (rejette en secret tous les paquets), "ACCEPT" (accepte tous les paquets). Pour une description plus détaillée, voir la documentation de la variable `PF_INPUT_POLICY`

Configuration par défaut : `PF6_INPUT_POLICY='REJECT'`

PF6_INPUT_ACCEPT_DEF Dans cette variable vous pouvez activer les règles prédéfinies pour la chaîne INPUT du pare-feu IPv6. Les valeurs possibles sont "yes" ou "no".

La règle par défaut pour l'ouverture entrante du trafic pings-ICMPv6 (un ping par seconde en tant que limite), ainsi que pour les paquets NPD (Neighbour Discovery Protocol) sur le pare-feu, qui sont nécessaires pour l'auto-configuration sans état des réseaux IPv6. La communication localhost et la réponse des paquets entre la communication d'origine locale, sont également autorisés. Enfin, le pare-feu IPv4 est réglé de telle sorte que pour chaque tunnel IPv6 encapsulé dans le paquet IPv4, la communication avec l'extrémité du tunnel sera acceptée.

Configuration par défaut : `PF6_INPUT_ACCEPT_DEF='yes'`

PF6_INPUT_LOG Cette variable active le fichier journal il enregistre tous les paquets entrants et rejetés. Les valeurs possibles sont "yes" ou "no". Pour une description plus détaillée, voir la documentation de la variable `PF_INPUT_LOG`.

12. Une exception existe, si "Privacy extension" est activé pour les hôtes du LAN, alors une partie de l'adresse IPv6 sera générée de façon aléatoire. Ces adresses par définition, ne sont pas connues du monde extérieur et donc la configuration du firewall sera partiellement ou pas du tout pertinente.

4. Les paquetages

Configuration par défaut : `PF6_INPUT_LOG='no'`

PF6_INPUT_LOG_LIMIT On configure avec cette variable une limite pour le fichier journal de la chaîne INPUT du pare-feu IPv6, pour garder le fichier journal en lecture. Pour une description plus détaillée, voir la documentation de la variable `PF_INPUT_LOG_LIMIT`.

Configuration par défaut : `PF6_INPUT_LOG_LIMIT='3/minute:5'`

PF6_INPUT_REJ_LIMIT On configure avec cette variable une limite pour le rejet des paquets TCP entrants. Les paquets TCP dépassant cette limite, seront rejetés avec la méthode douce (DROP). Pour une description plus détaillée, voir la documentation de la variable `PF_INPUT_REJ_LIMIT`.

Configuration par défaut : `PF6_INPUT_REJ_LIMIT='1/second:5'`

PF6_INPUT_UDP_REJ_LIMIT On configure avec cette variable une limite pour le rejet des paquets UDP entrants. Les paquets UDP dépassant cette limite, seront rejetés avec la méthode douce (DROP). Pour une description plus détaillée, voir la documentation de la variable `PF_INPUT_UDP_REJ_LIMIT`.

Configuration par défaut : `PF6_INPUT_UDP_REJ_LIMIT='1/second:5'`

PF6_INPUT_ICMP_ECHO_REQ_LIMIT Avec cette variable, vous définissez la façon de répondre à une demande de requête écho ICMPv6 commune. La fréquence et la limite de restriction est décrite analogiquement comme ceci 'n/unité de tempsrafales' par exemple, '3/minute :5'. Une fois que la limite est dépassée, le paquet est tout simplement ignoré (DROP). S'il la variable est vide, la valeur par défaut utilisé sera la suivante '1/seconde :5' si la variable contient 'none', alors, aucune limite ne sera effectuée.

Configuration par défaut : `PF6_INPUT_ICMP_ECHO_REQ_LIMIT='1/second:5'`

PF6_INPUT_ICMP_ECHO_REQ_SIZE Avec cette variable, vous définissez la taille (en octets) que peut recevoir la demande de requête écho ICMPv6. Ce chiffre vient "ajouter" des données à l'en-tête du paquet à prendre en considération. La valeur par défaut est de 150 octets.

Configuration par défaut : `PF6_INPUT_ICMP_ECHO_REQ_SIZE='150'`

PF6_INPUT_N Dans cette variable vous indiquez le nombre de règles du pare-feu IPv6 pour les paquets entrants (chaîne INPUT). Par défaut, deux règles sont activées : la première permet l'accès au routeur par tous des hôtes locaux via l'adresse du niveau de lien et la seconde permet la communication des hôtes du premier sous-réseau IPv6 défini avec le routeur.

Si plusieurs sous-réseaux IPv6 locaux sont définis, la seconde règle doit être reproduite au temps de fois que nécessaire. Voir le fichier de configuration.

Exemple : `PF6_INPUT_N='2'`

PF6_INPUT_x Dans cette variable, vous indiquez la règle pour la chaîne INPUT du pare-feu IPv6. Pour une description plus détaillée, voir la documentation de la variable `PF_INPUT_x`.

Les différences par rapport au pare-feu IPv4 :

- Au lieu de `IP_NET_x` vous devez mettre `IPV6_NET_x`.
- Au lieu de `IP_ROUTE_x` vous devez mettre `IPV6_ROUTE_x`.
- Les adresses IPv6 doivent être placées entre deux crochets. (y compris le masque de sous réseau, s'il est disponible).
- Tous les adresses IPv6 que vous indiquez (y compris les variables `IPV6_NET_x`, etc.) doivent être placées entre deux crochets, si l'adresse est suivi d'un port ou d'une plage de ports.

4. Les paquetages

Exemple :

```
PF6_INPUT_1='[fe80::0/10] ACCEPT'  
PF6_INPUT_2='IPV6_NET_1 ACCEPT'  
PF6_INPUT_3='tmpl:samba DROP NOLOG'
```

PF6_INPUT_x_COMMENT Dans cette variable, vous pouvez indiquer une description ou un commentaire associé à la règle INPUT.

Exemple : `PF6_INPUT_3_COMMENT='no_samba_traffic_allowed'`

PF6_FORWARD_POLICY Avec cette variable vous définissez la stratégie par défaut pour les paquets transmis par le routeur avec la (chaîne FORWARD). Les valeurs possibles sont "REJECT" (par défaut, rejette tous les paquets), "DROP" (rejette en secret tous les paquets), "ACCEPT" (accepte tous les paquets). Pour une description plus détaillée, voir la documentation de la variable `PF_FORWARD_POLICY`.

Configuration par défaut : `PF6_FORWARD_POLICY='REJECT'`

PF6_FORWARD_ACCEPT_DEF Cette variable active les règles prédéfinies pour la chaîne FORWARD du pare-feu IPv6. Les valeurs possibles sont "yes" ou "no".

Ouverture des règles par défaut sur le pare-feu pour les ping ICMPv6 sortants (un ping par seconde comme limite). Les paquets de réponses au ping seront également autorisés.

Configuration par défaut : `PF6_FORWARD_ACCEPT_DEF='yes'`

PF6_FORWARD_LOG Cette variable active le fichier journal il enregistre tous les paquets entrants et rejetés. Les valeurs possibles sont "yes" ou "no". Pour une description plus détaillée, voir la documentation de la variable `PF_FORWARD_LOG`.

Configuration par défaut : `PF6_FORWARD_LOG='no'`

PF6_FORWARD_LOG_LIMIT On configure avec cette variable une limite pour le fichier journal de la chaîne FORWARD du pare-feu IPv6, pour garder le fichier journal en lecture.

Pour une description plus détaillée, voir la documentation de la variable `PF_FORWARD_LOG_LIMIT`.

Configuration par défaut : `PF6_FORWARD_LOG_LIMIT='3/minute:5'`

PF6_FORWARD_REJ_LIMIT On configure avec cette variable une limite pour le rejet des paquets TCP entrants. Les paquets TCP dépassant cette limite, seront rejetés en secret avec (DROP). Pour une description plus détaillée, voir la documentation de la variable `PF_FORWARD_REJ_LIMIT`.

Configuration par défaut : `PF6_FORWARD_REJ_LIMIT='1/second:5'`

PF6_FORWARD_UDP_REJ_LIMIT On configure avec cette variable une limite pour le rejet des paquets UDP entrants. Les paquets UDP dépassant cette limite, seront rejetés avec la méthode douce (DROP). Pour une description plus détaillée, voir la documentation de la variable `PF_FORWARD_UDP_REJ_LIMIT`.

Configuration par défaut : `PF6_FORWARD_UDP_REJ_LIMIT='1/second:5'`

PF6_FORWARD_N Dans cette variable vous indiquez le nombre de règles du pare-feu IPv6 pour les paquets entrants (chaîne FORWARD). Par défaut, deux règles sont activées : la première empêche la transmission de tous les paquets samba dans d'autres réseaux qui ne proviennent pas du réseau local et la seconde permet la communication à partir des hôtes du premier sous-réseau IPv6 défini dans le routeur.

Si plusieurs sous-réseaux IPv6 locaux sont définis, la seconde règle doit être reproduite au temps de fois que nécessaire. Voir le fichier de configuration.

Exemple : `PF6_FORWARD_N='2'`

PF6_FORWARD_x Dans cette variable, vous indiquez la règle pour la chaîne FORWARD du pare-feu IPv6. Pour une description plus détaillée, voir la documentation de la variable PF_FORWARD_x.

Les différences par rapport au pare-feu IPv4 :

- Au lieu de IP_NET_x vous devez mettre IPV6_NET_x.
- Au lieu de IP_ROUTE_x vous devez mettre IPV6_ROUTE_x.
- Les adresses IPv6 doivent être placées entre deux crochets. (y compris le masque de sous réseau, s'il est disponible).
- Tous les adresses IPv6 que vous indiquez (y compris les variables IPV6_NET_x, etc.) doivent être placées entre deux crochets si l'adresse est suivi d'un port ou d'une plage de ports.

Exemple :

```
PF6_FORWARD_1='tmpl:samba DROP'
PF6_FORWARD_2='IPV6_NET_1 ACCEPT'
```

PF6_FORWARD_x_COMMENT Dans cette variable, vous pouvez indiquer une description ou un commentaire associé à la règle FORWARD.

Exemple : PF6_FORWARD_1_COMMENT='no_samba_traffic_allowed'

PF6_OUTPUT_POLICY Cette variable définit la stratégie par défaut pour les paquets sortants du routeur(chaine OUTPUT). Les valeurs possibles sont "REJECT" (par défaut, pour tous les paquets), "DROP" (rejette secrètement tous les paquets) et "ACCEPT" (accepte tous les paquets). Pour plus de détails, reportez-vous à la documentation de la variable PF_OUTPUT_POLICY.

Configuration par défaut : PF6_OUTPUT_POLICY='REJECT'

PF6_OUTPUT_ACCEPT_DEF Cette variable active les règles pré-réglées pour la chaîne OUTPUT du pare-feu IPv6. Les valeurs possibles sont "yes" ou "no". À l'heure actuelle, il n'existe pas de règle prédéfinie.

Configuration par défaut : PF6_OUTPUT_ACCEPT_DEF='yes'

PF6_OUTPUT_LOG Cette variable permet l'enregistrement tous les paquets sortants rejetés. Les valeurs possibles sont "yes" ou "no". Pour plus de détails, reportez-vous à la documentation de la variable PF_OUTPUT_LOG.

Configuration par défaut : PF6_OUTPUT_LOG='no'

PF6_OUTPUT_LOG_LIMIT On configure avec cette variable une limite pour le journal de la chaîne OUTPUT du pare-feu IPv6, pour garder le fichier journal en lecture. Pour une description plus détaillée de la documentation voir la variable PF_OUTPUT_LOG_LIMIT.

Configuration par défaut : PF6_OUTPUT_LOG_LIMIT='3/minute:5'

PF6_OUTPUT_REJ_LIMIT On configure avec cette variable une limite pour le rejet des paquets TCP sortants. Les paquets TCP dépassant cette limite, seront rejetés en secret avec (DROP). Pour une description plus détaillée, voir la documentation de la variable PF_OUTPUT_REJ_LIMIT.

Configuration par défaut : PF6_OUTPUT_REJ_LIMIT='1/second:5'

PF6_OUTPUT_UDP_REJ_LIMIT On configure avec cette variable une limite pour le rejet des paquets UDP sortants. Les paquets UDP dépassant cette limite, seront rejetés en secret avec (DROP). Pour une description plus détaillée, voir la documentation de la variable PF_OUTPUT_UDP_REJ_LIMIT.

Configuration par défaut : `PF6_OUTPUT_UDP_REJ_LIMIT='1/second:5'`

PF6_OUTPUT_N Dans cette variable vous indiquez le nombre de règles du pare-feu IPv6 pour les paquets entrants (chaîne OUTPUT). Par défaut, deux règles sont activées : la première permet l'accès au routeur par tous des hôtes locaux via l'adresse du niveau de lien et la seconde permet la communication des hôtes du premier sous-réseau IPv6 défini avec le routeur.

Si plusieurs sous-réseaux IPv6 locaux sont définis, la seconde règle doit être reproduite au temps de fois que nécessaire. Voir le fichier de configuration.

Exemple : `PF6_OUTPUT_N='1'`

PF6_OUTPUT_x Dans cette variable, vous indiquez la règle pour la chaîne OUTPUT du pare-feu IPv6. Pour une description plus détaillée, voir la documentation de la variable `PF_OUTPUT_x`.

Les différences par rapport au pare-feu IPv4 :

- Au lieu de `IP_NET_x` vous devez mettre `IPV6_NET_x`.
- Au lieu de `IP_ROUTE_x` vous devez mettre `IPV6_ROUTE_x`.
- Les adresses IPv6 doivent être placées entre deux crochets. (y compris le masque de sous réseau, s'il est disponible).
- Tous les adresses IPv6 que vous indiquez (y compris les variables `IPV6_NET_x`, etc.) doivent être placées entre deux crochets si l'adresse est suivi d'un port ou d'une plage de ports.

Exemple :

`PF6_OUTPUT_1='tmpl:ftp IPV6_NET_1 ACCEPT HELPER:ftp'`

PF6_OUTPUT_x_COMMENT Dans cette variable, vous pouvez indiquer une description ou un commentaire associé à la règle OUTPUT.

Exemple : `PF6_OUTPUT_3_COMMENT='no_samba_traffic_allowed'`

PF6_USR_CHAIN_N Dans cette variable, vous indiquez le nombre de chaînes, qui seront définies par l'utilisateur dans la table du pare-feu IPv6. Pour une description plus détaillée, voir la documentation de la variable `PF_USR_CHAIN_N`.

Configuration par défaut : `PF6_USR_CHAIN_N='0'`

PF6_USR_CHAIN_x_NAME Dans cette variable, vous indiquez le nom personnalisé de la table du pare-feu IPv6. Pour une description plus détaillée, voir la documentation de la variable `PF_USR_CHAIN_x_NAME`

Exemple : `PF6_USR_CHAIN_1_NAME='usr-myvpn'`

PF6_USR_CHAIN_x_RULE_N Dans cette variable, vous indiquez le nombre de règles personnalisées pour pare-feu IPv6 associé à la table du pare-feu IPv6. Pour une description plus détaillée, voir la documentation de la variable `PF_USR_CHAIN_x_RULE_N`.

Exemple : `PF6_USR_CHAIN_1_RULE_N='0'`

PF6_USR_CHAIN_x_RULE_x dans cette variable, vous indiquez la règle définie par l'utilisateur de la table du pare-feu IPv6. Pour une description plus détaillée, voir la documentation de la variable `PF_USR_CHAIN_x_RULE_x`

Les différences par rapport au pare-feu IPv4 :

- Au lieu de `IP_NET_x` vous devez mettre `IPV6_NET_x`.
- Au lieu de `IP_ROUTE_x` vous devez mettre `IPV6_ROUTE_x`.

4. Les paquetages

- Les adresses IPv6 doivent être placées entre deux crochets. (y compris le masque de sous réseau, s'il est disponible).
- Tous les adresses IPv6 que vous indiquez (y compris les variables `IPV6_NET_x`, etc.) doivent être placées entre deux crochets si l'adresse est suivi d'un port ou d'une plage de ports.

PF6_USR_CHAIN_x_RULE_x_COMMENT Dans cette variable, vous pouvez indiquer une description ou un commentaire associé à la règle.

Exemple : `PF6_USR_CHAIN_1_RULE_1_COMMENT='some_user-defined_rule'`

PF6_POSTROUTING_N Dans cette variable vous indiquez le nombre de règles du pare-feu IPv6 pour le masquage des paquets (chaîne POSTROUTING). Pour plus de détails, reportez-vous à la documentation de la variable `PF_POSTROUTING_N`.

Exemple : `PF6_POSTROUTING_N='2'`

PF6_POSTROUTING_x PF6_POSTROUTING_x_COMMENT

Vous indiquez dans ces variables la liste de règles qui décrivent les paquets IPv6 qui seront masqués par le routeur (ou transmis non masqué). Pour plus de détails, reportez-vous à la documentation de la variable `PF_POSTROUTING_x`

PF6_PREROUTING_N Dans cette variable, vous indiquez le nombre de règles du pare-feu IPv6 pour transmettre les paquets vers une autre destination (chaîne PREROUTING). Pour plus de détails, reportez-vous à la documentation de la variable `PF_PREROUTING_N`.

Exemple : `PF6_PREROUTING_N='2'`

PF6_PREROUTING_x PF6_PREROUTING_x_COMMENT

Vous indiquez dans ces variables la liste de règles qui décrivent la transmission des paquets IPv6 du routeur vers une autre destination. Pour plus de détails, reportez-vous à la documentation de la variable `PF_PREROUTING_x`.

4.12.4. WebGUI

Ce paquetage installe un menu supplémentaire dans le mini-HTTPD pour le "filtrage de paquets (IPv6)", sous lequel vous pourrez voir les enregistrements du filtrage de paquets de votre configuration IPv6.

4.13. ISDN - Communication avec les cartes ISDN (ou Numéris) actives et passives

fli4l a été pensé principalement pour l'application ISDN (ou Numéris) et/ou pour un routeur DSL. Avec le paramétrage de la variable `OPT_ISDN='yes'` le programme ISDN devient actif. A condition que la carte-ISDN soit supporté par fli4l.

Si vous n'utilisez pas d'ISDN, le paramètre de la variable sera alors `OPT_ISDN='no'` aucune carte ISDN ne sera installée. Vous pouvez alors ignorer la suite de ce chapitre.

Configuration par défaut : `OPT_ISDN='no'`

4.13.1. Établir une connexion par ISDN

La sélection du comportement de fli4l est déterminé par trois variables différentes, `DIALMODE`, `ISDN_CIRC_X_ROUTE_X`, `ISDN_CIRC_X_TIMES`. Lorsqu'un paquet arrive sur le circuit [DIALMODE](#)

4. Les paquetages

(Page 69) actif (dans le fichier <config>/base.txt), une connexion doit se construire automatiquement ou pas. La variable DIALMODE peut accepter les valeurs suivantes :

auto Si un paquet arrive sur le circuit-ISDN (il traverse l'interface-ISDN puis il est changé en ipp*) il ouvre automatiquement une connexion. Quand le paquet arrive sur le circuit-ISDN les deux variables sont utilisées ISDN_CIRC_X_ROUTE_X et ISDN_CIRC_X_TIMES.

manual En mode manuel, la connexion doit être déclenchés avec le client imonc/imond. La documentation se trouve dans la section imonc/imond.

off Aucune connexion ISDN n'est établie.

Les paquets qui arrivent sur le réseau, sont défini dans la variable ISDN_CIRC_X_ROUTE_X qui est prés configuré. Avec le réglage '0.0.0.0/0', on appelle ce réglage 'default route' (ou route par défaut) avec pondération. Cela signifie aussi que tous les paquets qui quittent le réseau local passent par ce réseau, s'il est actif. Quand le réseau est actif, la variable ISDN_CIRC_X_TIMES est analysée, puis fli4l li la variable *least cost routing* "moindre coût de routage" (voir la section [Least-Cost-Routing - Mode de fonctionnement](#) (Page 269) dans la Documentation du paquetage base). Si l'on ne veut pas que les paquets passent par tous les réseaux, mais uniquement sur certains réseaux (par ex. le réseau d'entreprise), on pourra indiquer ici un ou plusieurs réseaux différent. fli4l gère l'Interface-ISDN qui est en permanence actif et le réseau informatique qui lui est assigné. Maintenant si un paquet est envoyé par un PC du réseau local, une connexion automatiquement se crée.

Comme déjà mentionné, la variable ISDN_CIRC_X_TIMES sert à activer le coût des connexions pour le circuit-ISDN, lorsque le réseau 'default route' (ou routage par défaut) est activé le tableau des frais de connexion peut être déclenchés. 'Si' dans la variable on a spécifié la date, les deux premiers paramètres de time-info (par ex. Mo-Fr :09-18), et 'si' le quatrième paramètre de lc-default-route est sur (y/n). fli4l (par le démon imond) s'occupe alors des paquets qui quittent le réseau local, passent toujours par le réseau actif puis par l'Interface-ISDN et établi une connexion avec le fournisseur d'accès Internet à la date/heure indiqué.

En résumé selon l'utilisation par défaut on peut dire suivant les cas :

- Si on veut uniquement Internet, on met auto dans DIALMODE, on définit 1-n circuit, comme le premier route '0.0.0.0/0' et comme temps (avec lc-default-route = Y) pour toute la semaine.

```
ISDN_CIRC_%_ROUTE_N='1'
ISDN_CIRC_%_ROUTE_1='0.0.0.0/0'
ISDN_CIRC_%_TIMES='Mo-Su:00-24:0.0148:Y'
```

- Si on veut utiliser une connexion spécial pour un réseau d'entreprise, on définit le réseau (ou plusieurs réseaux) avec une Route différente de '0.0.0.0/0', ainsi vous avez en permanence un accès actif pour un réseau d'entreprise spécifique.

```
ISDN_CIRC_%_ROUTE_N='1'
ISDN_CIRC_%_ROUTE_1='network/netmaskbits'
ISDN_CIRC_%_TIMES='Mo-Su:00-24:0.0148:Y'
```

4.13.2. Carte ISDN

ISDN_TYPE ISDN_IO ISDN_IO0 ISDN_IO1 ISDN_MEM ISDN_IRQ ISDN_IP ISDN_PORT

On mentionne ici les données techniques pour la carte ISDN (ou RNIS).

Les valeurs mentionnées dans l'exemple fonctionnent avec la carte TELES 16.3, elle est réglée sur l'Adresse-IO 0xd80 (avec le Switches-Dip). Si vous changez le réglage de la carte, l'adresse doit être modifiée.

Erreur fréquemment faite (exemple) :

```
ISDN_IO='240' -- au lieu de~: ISDN_IO='0x240'
```

Si vous utilisez l'IRQ 12 vous devez couper la souris PS/2 qui est éventuellement disponible dans le BIOS. Mieux vaut choisir un autre IRQ ! «Les bons» IRQ sont généralement 5, 10 et 11.

ISDN_TYPE par principe le type correspond au numéro de pilote HiSax. Excepté : les cartes non-HiSax comme par exemple AVM-B1, la numérotation de ces types de cartes a été élargie. La liste de tous des types HiSax sont indiquées dans `linux-2.x.y/Documentation/isdn/README.HiSax`.

Type	Carte	Paramètres nécessaires
Numéro du type pour pilote factice :		
0	no driver (dummy)	none
Numéro du type pour pilote HiSax :		
1	Teles 16.0	irq, mem, io
2	Teles 8.0	irq, mem
3	Teles 16.3 (non PnP)	irq, io
4	Creatix/Teles PnP	irq, io0 (ISAC), io1 (HSCX)
5	AVM A1 (Fritz)	irq, io
5	AVM (Fritz !Card Classic)	irq, io
6	ELSA PCC/PCF cards	io or nothing for autodetect (the io base is required only if you have more than one ELSA card in your PC)
7	ELSA Quickstep 1000	irq, io (from isapnp setup)
8	Teles 16.3 PCMCIA	irq, io
9	ITK ix1-micro Rev.2	irq, io (from isapnp setup ?)
10	ELSA PCMCIA	irq, io (set with card manager)
11	Eicon.Diehl Diva ISA PnP	irq, io
11	Eicon.Diehl Diva PCI	no parameter
12	ASUS COM ISDNLink	irq, io (from isapnp setup)
13	HFC-2BS0 based cards	irq, io
14	Teles 16.3c PnP	irq, io
15	Sedlbauer Speed Card	irq, io
15	Sedlbauer PC/104	irq, io
15	Sedlbauer Speed PCI	no parameter
16	USR Sportster internal	irq, io
17	MIC card	irq, io
18	ELSA Quickstep 1000PCI	no parameter
19	Compaq ISDN S0 ISA card	irq, io0, io1, io (from isapnp setup io=IO2)
20	NETjet PCI card	no parameter
21	Teles PCI	no parameter
22	Sedlbauer Speed Star (PCMCIA)	irq, io (set with card manager)
23	reserved (AMD 7930)	n.a.
24	Dr. Neuhaus Niccy PnP	irq, io0, io1 (from isapnp setup)
24	Dr. Neuhaus Niccy PCI	no parameter
25	Teles S0Box	irq, io (of the used lpt port)
26	AVM A1 PCMCIA (Fritz !)	irq, io (set with card manager)
27	AVM PnP (Fritz !PnP)	irq, io (from isapnp setup)

4. Les paquetages

Type	Carte	Paramètres nécessaires
27	AVM PCI (Fritz !PCI)	no parameter
28	Sedlbauer Speed Fax+	irq, io (from isapnp setup)
29	Siemens I-Surf 1.0	irq, io, memory (from isapnp setup)
30	ACER P10	irq, io (from isapnp setup)
31	HST Saphir	irq, io
32	Telekom A4T	none
33	Scitel Quadro	subcontroller (4*S0, subctrl 1...4)
34	Gazel ISDN cards (ISA)	irq,io
34	Gazel ISDN cards (PCI)	none
35	HFC 2BDS0 PCI	none
36	W6692 based PCI cards	none
37	2BDS0 S+, SP	irq,io
38	NETspider U PCI	none
39	2BDS0 SP/PCMCIA ¹³	irq,io (set with card manager)
40	not used (hotplug)	n.a.
41	Formula-n enter :now PCI	none
81	ST5481 USB ISDN adapters	none
82	HFC USB based ISDN adapters	none
83	HFC-4S/8S based ISDN cards	none
84	AVM Fritz !Card PCI/PCIV2/PnP	none
Numéro du type pour pilote Capi :		
100	Dispositif générique CAPI sans la fonction ISDN, pour par ex. AVM Fritz !DSL SL	no parameter
101	AVM-B1 PCI	no parameter
102	AVM-B1 ISA	irq, io
103	AVM-B1/M1/M2 PCMCIA	no parameter
104	AVM Fritz !DSL	no parameter
105	AVM Fritz !PCI	no parameter
106	AVM Fritz !PNP	irq, io (from isapnp setup)
107	AVM Fritz !Classic	irq, io
108	AVM Fritz !DSLv2	no parameter
109	AVM Fritz !USBv2	no parameter
110	AVM Fritz !DSL USB	no parameter
111	AVM Fritz !USB	no parameter
112	AVM Fritz !X USB	no parameter
113	AVM FRITZ !DSL USBv2	no parameter
114	AVM FRITZ !PCMCIA	no parameter
160	AVM Fritz !Box Remote-Capi	ip,port
161	Melware Remote CAPI (rcapi)	ip,port
Numéro du type pour d'autres pilotes :		
201	ICN 2B	io, mem
Numéro du type pour pilote mISDN (expérimental) :		
301	HFC-4S/8S/E1 multiport cards	no parameter
302	HFC-PCI based cards	no parameter
303	HFC-USB Adapters	no parameter
304	AVM Fritz !Card PCI (v1 and v2) cards	no parameter

13. Indiquer le type 84 qui correspondait au type 39 ancienne version.

4. Les paquetages

Type	Carte	Paramètres nécessaires
305	cards based on Infineon (former Siemens) chips : - Dialogic Diva 2.0 - Dialogic Diva 2.0U - Dialogic Diva 2.01 - Dialogic Diva 2.02 - Sedlbauer Speedwin - HST Saphir3 - Develo (former ELSA) Microlink PCI (Quickstep 1000) - Develo (former ELSA) Quickstep 3000 - Berkomp Scitel BRIX Quadro - Dr.Neuhaus (Sagem) Niccy	no parameter
306	NetJet TJ 300 and TJ320 cards	no parameter
307	Sedlbauer Speedfax+ cards	no parameter
308	Winbond 6692 based cards	no parameter

Ma carte est une Teles 16.3 NON-PNP ISA, elle est donc du Type=3.

Par exemple pour une carte ICN-2B les paramètres IO et MEM doivent être, `ISDN_IO='0x320'`, `ISDN_MEM='0xd0000'`.

Les nouvelles cartes Teles-PCI doivent être indiqués avec le type=20 (au lieu de 21). Au sujet des paramètres ils sont indiqués dans "cat /proc/pci" avec "tiger" ou semblable. Si vous ne trouvez pas cet valeurs, je ne peux rien faire pour vous, désolé.

Pour la configuration des Types-ISDN 104 à 114, il est nécessaire au préalable de télécharger les pilotes ici <http://www.fli4l.de/fr/telechargement/version-stable/pilote-avm/> et de décompresser les fichiers dans le répertoire fli4l. malheureusement ces pilotes ne sont pas sous licence GPL, c'est pour cela qu'il ne sont pas fournis avec le paquetage ISDN.

Pour l'utilisation des Types-ISDN 81, 82, 109 à 113 et 303 il est nécessaire d'installer et d'activer le support USB. Voir la section [USB - Gestion des périphériques USB](#) (Page 234).

Pour l'utilisation des Types-ISDN 10, 22, 26, 39, 103 et 114 il est nécessaire d'installer et d'activer les cartes-PC PCMCIA. Voir la section [PCMCIA - Gestion des Cartes-PC](#) (Page ??).

Quelques Conseils sont disponibles sur fli4l-FAQ ou sur Mailing-liste au sujet des numéros de type pour les pilotes de cartes ISDN, si vous ne savait pas exactement quel Type de carte est dans votre PC.

Certains types de cartes sont identifiés avec la fonction «from isapnp setup» ils doivent être initialisés avec l'outil PnP-Tool isapnp - S'il s'agit réellement d'une carte PnP. Voir la documentation dans le chapitre [OPT_PNP - Outil d'installation pour isapnp](#) (Page 73).

Le Type-ISDN 0 est nécessaire, si l'on veut installer le paquetage ISDN **sans** carte ISDN dans votre PC, pour pouvoir par exemple utiliser imond sur le routeur et le client imonc sur le réseau local.

ISDN_DEBUG_LEVEL Pour activer Debug-Level avec les cartes HiSaX. Pour vous aidez, Debug-Level (ou niveau de débogage) se compose de valeurs suivantes (Documentation

original) :

Number	Debug-Information
1	Link-level <-> hardware-level communication
2	Top state machine
4	D-Channel Q.931 (call control messages)
8	D-Channel Q.921
16	B-Channel X.75
32	D-Channel l2
64	B-Channel l2
128	D-Channel link state debugging
256	B-Channel link state debugging
512	TEI debug
1024	LOCK debug in callc.c
2048	More debug in callc.c (not for normal use)

Valeur par défaut (ISDN_DEBUG_LEVEL='32') devrait être la plus adaptée.

ISDN_VERBOSE_LEVEL Avec cette variable on peut régler "divers" fonction dans le sous système ISDN du Kernel fli4l. Dans Verbose-Level chaque numéro correspond à un niveau du plus bas au plus haut. Voici les Verbose-Level :

- '0' Enregistrement d'aucune information supplémentaire
- '1' Enregistrement à la moindre connexion ISDN
- '2' et '3' Les appels Tél. son enregistrés dans un journal
- '4' et plus enregistrement régulier du taux de transfert de données.

Pour visualiser les messages du Kernel-Logging-Interface, il faut activer la variable `OPT_SYSLOGD` (Page 70).

Important: *si les appels Tél. sont activer avec telmond, le réglage pour l'enregistrement des appel Tél. ne doit pas être inférieur à 2 autrement aucun appel ne sera enregistré.*

Configuration par défaut : ISDN_VERBOSE_LEVEL='2'

ISDN_FILTER Active le mécanisme de filtrage du Kernel, afin d'assurer le bon fonctionnement du Hangup-Timeout. Voir pour de plus amples informations <http://www.fli4l.de/hilfe/howtos/basteleien/hangup-problem-loesen/>

4.13.3. OPT_ISDN_COMP (Expérimental)

Si vous avez activez la variable `OPT_ISDN_COMP='yes'` la compression de LZS et de BSD sera possible. Les paquets seront compressés lorsque la connexion sera établie. Merci à Arwin Vosselman (courriel: [arwin\(at\)xs4all\(dot\)nl](mailto:arwin(at)xs4all(dot)nl)). Cette variable supplémentaire à un statut expérimental.

Configuration par défaut : `OPT_ISDN_COMP='no'`

Détail des paramètres pour DEBUG, nécessaires avec la compression LZS :

ISDN_LZS_DEBUG (Expérimental) Réglage-Debug-Level :

- '0' aucune Information de Debogage
- '1' Information normale de Debogage
- '2' Information élargie de Debogage
- '3' Information total de Debogage (incl. dumping des paquets de données)

Configuration par défaut : `ISDN_LZS_DEBUG='1'`

en cas de problèmes de compression, pour avoir plus détail sur des messages de débogage, mettais la variable sur '2'.

ISDN_LZS_COMP (EXPERIMENTAL) Puissance de compression (pas de la décompression !). Restez sur la valeur '8'. Les valeurs possibles sont de 0 à 9

Plus le nombre est grand meilleure est la compression, cependant '9' est excessif, le CPU est trop sollicité.

Configuration par défaut : `ISDN_LZS_COMP='8'`

ISDN_LZS_TWEAK (EXPERIMENTAL) Dans cette variable vous pouvez laisser la valeur sur '7'.

Configuration par défaut : `ISDN_LZS_TWEAK='7'`

En plus de la configuration des 3 dernières variables, la variable `ISDN_CIRC_x_FRAMECOMP` doit être configurée, voir le chapitre suivant.

4.13.4. Circuits ISDN

Dans la configuration de `fli4l` on peut définir plusieurs connexions via ISDN. Au maximum 2 connexions égal sont possibles sur une mêmes carte ISDN.

La définition de ces connexions, se nomme Circuit dans la configuration de `fli4l`. Un Circuit est utilisé par connexion.

Dans notre fichier d'exemple `config.txt` on a définis deux Circuits :

- Circuit 1 : Dialout sur Internet-By-Call-Provider Microsoft Network, Sync-PPP
- Circuit 2 : Dialin/Dialout sur le routeur ISDN (par exemple on pourrait également indiquer, `fli4l`)

Avec Raw-IP (ce fonctionnement interne, utilise uniquement les numéros de téléphone pour établir une connexion), par exemple pour accéder au réseau d'entreprise de n'importe où.

Concrètement chez moi, c'est un Linux-Box avec `isd4linux` comme "adversaire".

Si le routeur `fli4l` sert uniquement de Gateway (ou passerelle) Internet, un seul circuit est nécessaire. Exception : si vous utilisez le routeur `fli4l` avec `Least-Cost-Router-Features` (ou calcul des coûts des connexions téléphoniques). Tous les circuits autorisés sont à définir dans des domaines différent, voir ci-dessous.

ISDN_CIRC_N Dans cette variable on indique le nombre de circuit ISDN à utiliser. Si vous utilisez uniquement `fli4l` comme écran d'affichage pour des appels Tél. avec ISDN vous pouvez paramétrer la variable :

```
ISDN_CIRC_N='0'
```

Si le routeur `fli4l` sert uniquement de Gateway (ou passerelle) Internet, un seul circuit est nécessaire. exception : pour le LC-Routing (ou calcul des coûts des connexions téléphoniques), voir ci-dessous.

ISDN_CIRC_x_NAME Dans cette variable on indique le nom du circuit maximum 15 caractères. Le nom sera visible sur le client-`imonc.exe` au lieu du numéro de Téléphone. Les caractères autorisés sont de 'A' à 'Z' (minuscule et majuscule), et les chiffres de '0' à '9' et aussi le trait d'union '-', par exemple.

```
ISDN_CIRC_x_NAME='msn'
```

4. Les paquetages

Le nom de circuit peut être utilisé pour configurer le filtrage de paquet ou OpenVPN. par ex. lorsque l'on configure le filtrage de paquet pour le Circuit ISDN, il faut indiquer d'abord le préfixe 'circuit_' puis le Nom de Circuit. Si le circuit s'appelle 'willi', on peut écrire dans le filtrage de paquet :

```
PF_INPUT_3='if:circuit_willi:any prot:udp 192.168.200.226 192.168.200.254:53 ACCEPT'
```

ISDN_CIRC_x_USEPEERDNS Il est établi que les fournisseurs d'accès Internet utilise un serveur de Nom (ou DNS) et nous devons enregistrer ce serveur de Nom dans notre réseau local pour la durée de connexion. Rationnellement cette option est seulement utiliser pour la connexion au fournisseur d'accès d'Internet. En même temps, presque tous les Fournisseurs supporte ce type de transfert.

Vous devez enregistrer les adresses-IP du serveur de Nom donné par votre FAI dans le fichier base.txt à la variable *DNS_FORWARDERS* et vous devez supprimer le serveur de Nom qui est configuré sur votre PC du réseau local. Ensuite vous devez mettre à la place l'adresse IP de votre routeur. Avec se réglage la résolution des Noms ne se perd pas dans le cache du serveur de nom.

Cette option offre l'avantage de toujours pouvoir travailler avec un serveur de nom le plus proche, dans la mesure où le fournisseur d'accès à une adresse IP correcte - Ainsi, la résolution de nom sera plus rapide.

En cas d'une défaillance d'un serveur de DNS du fournisseur d'accès, on pourra en règle général corriger plus rapidement les adresses des serveurs DNS du fournisseur d'accès.

Malgré tout, il est nécessaire d'indiquer un serveur de nom valide dans la variable *DNS_FORWARDERS* du fichier base.txt pour se connecter, autrement lors de la première connexion Internet la demande ne pourra pas être résolue correctement. En outre, la configuration originale du serveur de nom local est restaurée à la fin de la connexion.

Configuration par défaut : *ISDN_CIRC_x_USEPEERDNS*='yes'

ISDN_CIRC_x_TYPE Dans cette variable *ISDN_CIRC_x_TYPE* on indique le type de connexion-IP. Les valeurs possibles sont les suivantes :

'raw' RAW-IP
'ppp' Sync-PPP

Dans la plus par des cas on utilise PPP, Mais Raw-IP est un peu plus efficace, étant donné que PPP-Overhead est supprimé. Cependant une authentification de Raw-IP n'est pas possible, toutefois, on peut indiquer dans la variable *ISDN_CIRC_x_DIALIN* un accès limité à un numéro-ISDN (mot-clé "Clip"). Si on paramètre la variable *ISDN_CIRC_x_TYPE* avec 'raw' on peut créer un script analogique PPP up/down et un script raw up/down dans le dossier /etc/ppp.

ISDN_CIRC_x_BUNDLING Dans cette variable avec le protocole-MPPP (Multilink Protocol) RFC 1717, on permet l'agrégation des canaux-ISDN. Dans la pratique plupart du temps elle ne sont pas pertinentes, pour que ces restrictions soient applicables il faut :

- que se la soit possible uniquement avec la liaison PPP et non avec un Circuits-Raw
- que l'agrégation des canaux avec la nouvelle RFC 1990 (MLPPP) est pas possible

Le 2ème Canal peut est activé manuellement avec le client imonc ou activé automatiquement par rapport à la bande passante, pour cela voir la variable *ISDN_CIRC_x_BANDWIDTH*.

Configuration par défaut : *ISDN_CIRC_x_BUNDLING*='no'

Attention : lors de l'utilisation des canaux, associer à la compression cela peut occasionner des problèmes, voir aussi la description de la variable *ISDN_CIRC_x_FRAMECOMP*.

ISDN_CIRC_x_BANDWIDTH Si l'agrégation des canaux-ISDN est activée dans `ISDN_CIRC_x_BUNDLING='yes'`, vous pouvez paramétrer cette variable pour automatiser le 2ème canal-ISDN. Il y a 2 paramètres numériques à respecter :

1. La valeur du seuil en Octet/Seconde (S)
2. Interval temps en Seconde (Z)

Si la valeur du seuil (S) est dépassé pendant un interval temps (Z) en seconde, Imond active le processus de commutation du 2ème canal automatiquement. Si la valeur du seuil (S) est inférieur pendant un interval temps (Z) le 2ème canal se désactive automatiquement. Pour ne pas activer la bande passante automatiquement, il ne faut rien indiquer dans la variable `ISDN_CIRC_1_BANDWIDTH=""`, si vous voulez activer le 2ème canal il faut le faire manuellement avec le client imonc.

Exemple :

– `ISDN_CIRC_1_BANDWIDTH='6144 30'`

Si la valeur de transfert dépasse 6 Kilo-octets/seconde pendant 30 secondes le 2ème canal s'active.

– `ISDN_CIRC_1_BANDWIDTH='0 0'`

Le deuxième canal-ISDN sera immédiatement activé après 10 secondes au plus tard, sur une connexion Internet et restera active jusqu'à la coupure de la connexion.

– `ISDN_CIRC_1_BANDWIDTH=""`

Le deuxième canal-ISDN peut être uniquement activé manuellement, à condition que la variable `ISDN_CIRC_1_BUNDLING='yes'` soit configurée.

– `ISDN_CIRC_1_BANDWIDTH='10000 30'`

Normalement le deuxième canal sera activé lorsque la valeur de transfert atteint les 10 Ko/s pendant 30 secondes. Mais le deuxième canal ne s'activera jamais, car le maximum de transfert par canal est de 8 Ko/s.

Si la variable est sur `ISDN_CIRC_x_BUNDLING='no'`, la valeur de la variable `ISDN_CIRC_x_BANDWIDTH` est sans intérêts.

Configuration par défaut : `ISDN_CIRC_x_BANDWIDTH=""`

ISDN_CIRC_x_LOCAL On enregistre dans cette variable l'adresse IP du fournisseur d'accès Internet pour la partie ISDN. la variable n'est pas dans le fichier de configuration elle est à rajouter.

Si l'assignation de adresse IP est dynamique, le paramètre doit être **vide**. C'est au moment de la connexion, que l'adresse IP est négocié. Dans la plus par des cas les fournisseurs d'accès Internet donne une adresse IP dynamique. Cependant, si l'on doit attribuer une adresse IP, c'est ici que l'on doit l'inscrire. Cette variable est optionnelle et doit être configuré qu'en cas de besoin.

ISDN_CIRC_x_REMOTE On enregistre dans cette variable le l'adresse IP distant (renvoyer vers) et le masque de sous réseau pour la partie ISDN. le masque doit être écrit sous la forme CIDR (Classless Inter-Domain Routing). Vous trouverez plus d'information sur [CIDR](#) (Page 38) dans la documentation du paquetage Base à `IP_NET_x`.

Si l'assignation de adresse IP est dynamique, le paramètre doit être **vide**. C'est au moment de la connexion, que l'adresse IP est négocié. Dans la plus par des cas les fournisseurs d'accès Internet donne une adresse IP dynamique. Cependant, si l'on doit attribuer une adresse IP, c'est ici que l'on doit l'inscrire. Cette variable est optionnelle et doit être configuré qu'en cas de besoin.

Le numéro utilisé pour le masque de sous réseau est indiqué dans la configuration de l'interface. Il sera utilisé pour configurer les réseaux vers des hôtes, pour se connecter. En règle générale nous n'avons pas besoin de ce circuit, il est plus favorable, de créer uniquement un circuit directement à l'ordinateur de connexion. On place le masque de sous réseau sur /32, ici 32 est le nombre de bits du masque de sous réseau. Pour plus de détails, voir [chapitre : Détails techniques sur la Connexion](#) (Page 371).

ISDN_CIRC_x_MTU ISDN_CIRC_x_MRU Ces variables sont optionnelles, ont paramètres avec celles-ci le **MTU** (maximum transmission unit) et le **MRU** (maximum receive unit). Optionnelle signifie que les variables ne sont pas dans le fichier de configuration, Elles sont à insérer par l'utilisateur si besoin!

Normalement le réglage est : MTU 1500 et MRU 1524. Ce réglage doit être modifier uniquement dans des cas exceptionnels!

ISDN_CIRC_x_CLAMP_MSS On devrait mettre cette variable sur 'yes' si on utilise la synchronisation PPP (ISDN_CIRC_x_TYPE='ppp') et si l'un des symptômes suivants se produit :

- Si le navigateur du PC se connecte à un serveur web et qu'il n'y pas la page qui s'affichée et aucun message d'erreur, plus simplement il ne se passe rien.
- Lorsque vous envoyez de petits courriels cela fonctionne, mais si vous avez des problèmes pour envoyer des courriels plus important.
- Avec la fonction ssh, réinitialisation du scp après avoir établi une connexion.

Avec certain FAI, ces problèmes peuvent se produire par exemple Compuserve et aussi Mediaways (FAI Allemand).

Configuration par défaut : ISDN_CIRC_x_CLAMP_MSS='no'

ISDN_CIRC_x_HEADERCOMP Si vous paramétrez cette variable sur ISDN_CIRC_x_HEADERCOMP='yes' vous compressez les en-têtes proposé par (Van Jacobson TCP/IP Header Compression). Certain FAI ne supporte pas cette fonction. Si vous avez des problèmes avec la compression des en-têtes vous devez paramétrer la variable sur ISDN_CIRC_x_HEADERCOMP='no'.

Configuration par défaut : ISDN_CIRC_x_HEADERCOMP='yes'

ISDN_CIRC_x_FRAMECOMP (EXPERIMENTAL) Cette variable est uniquement pris en compte, que si la variable OPT_ISDN_COMP='yes' est activée. Cette variable réglemente les Frame-Compression (compression des en-têtes).

Les paramètres suivants sont possibles :

'no'	Aucune Compression de Frame (ou Trame)
'default'	LZS according RFC1974(std) and BSDCOMP 12
'all'	Negotiate lzs and bsdcomp
'lzs'	Negotiate lzs only
'lzsstd'	LZS according RFC1974 Standard Mode ("Sequential Mode")
'lzsext'	LZS according RFC1974 Extended Mode
'bsdcomp'	Negotiate bsdcomp only
'lzsstd-mh'	LZS Multihistory according RFC1974 Standard Mode ("Sequential Mode")

Quelle son les paramètres que l'on doit utilisé pour chaque fournisseur, on doit essayées. Avec le plus connu T-Online (Allemand) on paramètre uniquement 'lzsext'. Dans la plupart des cas, on peut se débrouiller avec le paramètre par défaut 'default' pour tous les autres fournisseurs d'accès.

Attention : lors de l'utilisation de l'agrégation des canaux en relation avec 'lzsext' il peut en résulter des problèmes. Ces problèmes sont largement connues, à la connexion au

serveur spécifique, et plus particulièrement à des FAI spécifique. cependant les problèmes ne provient pas uniquement des fournisseurs d'accès, mais peuvent provenir des différents noeuds de connexion.

le paramètre 'lzsstd-mh' a été pensé pour la communication de routeur à routeur. cette procédure n'est pas utilisé par les FAI mais lors de l'utilisation de deux routeurs fli4l amélioration considérable sur le transfert simultané de plusieurs fichiers. La compression des en-têtes est nécessaire et sera donc automatiquement activé.

ISDN_CIRC_x_REMOTENAME Cette variable est importante, uniquement pour la configuration de fli4l comme routeur de connexion distant. Vous pouvez enregistrer ici un nom hôte distant, normalement nous n'en avons pas besoin.

Configuration par défaut : `ISDN_CIRC_x_REMOTENAME=""`

ISDN_CIRC_x_PASS Dans ces variables on indique les données du fournisseur d'accès. Il s'agit Dans l'exemple ci-dessous, des données du fournisseur de Microsoft Network.

`ISDN_CIRC_x_USER` l'identification de l'utilisateur, `ISDN_CIRC_x_PASS` le mot de passe.

ATTENTION : pour un accès au FAI T-Online (pour l'Allemagne) il est à noter :

Le nom d'utilisateur `AAAAAAAAAAAAATTTTTT#MMMM` est composé, du numéro co-utilisateur, puis du numéro T-online de 12 chiffres et de l'identification. Le dernier chiffre du numéro T-Online doit se terminer par '#' si le numéro de T-Online ne comporte pas les 12 chiffres.

Avec ça si cela ne fonctionne pas ! (évidemment cela peut provenir du centrale téléphonique), le caractère '#' doit être placé entre le numéro T-Online et l'identification.

Evidemment si le (numéro T-Online comporte les 12 chiffres) il n'y a pas besoin de mettre le caractère '#'.

Exemple : `ISDN_CIRC_1_USER='123456#123'`

Avec le type de Circuit-Raw-IP ces variables n'ont aucune signification.

ISDN_CIRC_x_ROUTE_N Dans cette variable on configure le nombre de réseau à utiliser pour le circuit ISDN. Si vous avez qu'un routage par Défaut vous devez placez '1'.

ISDN_CIRC_x_ROUTE_X Dans cette variable vous indiquez le réseau ou les réseaux de routages pour le circuit ISDN. ici est enregistré pour la première variable tous les réseaux 0.0.0.0/0 (default route) ou (routage par défaut). le format est toujours 'network/netmaskbits' par exemple pour un réseau on écrira : '192.168.199.1/32'. Une société ou une université qui à plusieurs réseaux et veulent se connecter au routeur par exemple pour un accès Internet on configurera la variable :

```
ISDN_CIRC_%_ROUTE_N='2'
ISDN_CIRC_%_ROUTE_1='192.168.8.0/24'
ISDN_CIRC_%_ROUTE_2='192.168.9.0/24'
```

Lorsque vous avez plusieurs réseaux, vous devez paramétrer chaque réseau dans une variable `ISDN_CIRC_x_ROUTE_y=""` une ligne par réseau.

Si vous voulez utiliser LC-Routing-Features de fli4l, on peut configurer *plusieurs* réseaux comme Default-Route (ou routage par défaut). On pourra paramétrer un des circuits à utiliser pour le LC- dans la variable `ISDN_CIRC_x_TIMES` voir ci-dessous.

ISDN_CIRC_x_DIALOUT Dans cette variable `ISDN_CIRC_x_DIALOUT` on indique ici le numéro de téléphone par exemple celui du FAI. Il est possible d'indiquer plusieurs Numéros de Tél. (au cas où si l'un est occupé) - On sépare les numéros par un espace. Selon un rapport des Newsgroup il est possible d'indiquer un maximum de cinq numéros de Tél.

ISDN_CIRC_x_DIALIN Dans cette variable `ISDN_CIRC_x_DIALIN` on indique le numéro de Tél. personnel, si le circuit-ISDN est utilisé pour les appels Tél. - avec son indicatif, mais **sans** le premier 0. Par rapport au raccordement téléphonique derrière l'installation cela ne peut en être autrement, éventuellement si le premier voire le deuxième sont des numéros principaux.

Si le circuit le permet vous pouvez indiquer plusieurs numéros de correspondant ces numéros seront séparés par un espace. Le mieux est de rajouter un circuit par correspondant. Autrement cela pourrait à la connexion appeler les deux correspondants et créer des collisions bzgl sur le même circuit (mais c'est tout à fait possible avec l'ouverture des 2 canaux ISDN). C'est comme les adresses IP.

Si vous n'arrivez pas à avoir votre correspondant, vous pouvez essayer de mettre le '0' avant le numéro. Mais prudence : c'est permis uniquement, si le numéro d'appel ne peut pas être transmis !

Si on voulait réaliser une connexion indépendamment de MSN (ou Multiple numéros d'abonnés) et du correspondant, vous pouvez placer comme paramètre le caractère '*'.

Dans les deux derniers cas, une procédure d'authentification est indispensable (voir `ISDN_CIRC_x_AUTH`)

ISDN_CIRC_x_CALLBACK Réglage et processus Callback (ou rappel Téléphonique), valeurs possibles :

- 'in' fli4l appel et rappel Tél.
- 'out' fli4l appel, puis raccroche, attent et rappel de nouveau
- 'off' pas de Callback (rappel Tél.)
- 'cbcp' CallBack Control Protocol
- 'cbcp0' CallBack Control Protocol 0
- 'cbcp3' CallBack Control Protocol 3
- 'cbcp6' CallBack Control Protocol 6

Les protocoles de Contrôles CallBack (aussi appelé 'Microsoft CallBack'), le plus souvent utilisé est le protocole `cbcp6`.

Paramètre par défaut : 'off'

ISDN_CIRC_x_CBNUMBER Ici, on peut placer un numéro de rappel pour utilisation les protocoles `cbcp`, `cbcp3` et `cbcp6` (si on utilise `cbcp3` le numéro est obligatoire).

ISDN_CIRC_x_CBDELAY Dans cette variable on paramètre le délais en seconde du Callback (ou rappel Tél.). Selon le paramètre du rappel Tél. qui doit résulter, cette variable a une autre signification :

- `ISDN_CIRC_x_CALLBACK='in'` :
fli4l appel et rappel le numéro si occupé, on indique ici `ISDN_CIRC_x_CBDELAY` le temps en seconde entre les deux appels Tél. La bonne valeur est `ISDN_CIRC_x_CBDELAY='3'` pour rappeler le correspondant. Avec un valeur plus petite cela peu aussi fonctionner, la connexion téléphonique sera accélérer.
- `ISDN_CIRC_x_CALLBACK='out'` :
Dans ce cas fli4l appel et raccroche si occupé, on indique ici `ISDN_CIRC_x_CBDELAY` le temps attente en seconde pour appeler de nouveau. Là encore, `ISDN_CIRC_x_CBDELAY='3'` est une bonne valeur. Ce qui ma étonné : a l'appel Téléphonique, il suffit de 3 secondes pour "faire sonner", et avant que la serveur téléphonique réponde a la connexion Tél. en ville. Cette valeur ne doit jamais être plus basse. En cas de doute : Testez !

Si vous paramétrez cette variable sur `ISDN_CIRC_x_CALLBACK='off'`, la variable `ISDN_CIRC_x_CBDELAY` sera ignoré. De même, la variable `Callback Control Protocol`, n'aura pas d'importance.

ISDN_CIRC_x_EAZ Dans cette variable on paramètre l'indicatif régional, dans notre exemple MSN (l'appel du EAZ) est le 81330. Avec votre configuration MSN (ou Multiple numéros d'abonnés), vous avez peut être *pas* besoin d'indicatif à paramétrer.

Seul une ligne direct est configurer le plus souvent derrière une installation téléphonique sans indicatif régional. Cependant indiquer un '0' comme valeur peut aider, si vous avez des problèmes avec votre installation téléphonique. Avoir des remarques sur cette variable serait appréciable.

ISDN_CIRC_x_SLAVE_EAZ Si le routeur fli4l est équipé d'un Bus-S0 interne pour un deuxième numéro de téléphone et si l'on utilise l'agrégation des canaux, le deuxième numéros est à indiquer ici. Numéro du Tél. de diffusion sur le canal esclave.

Donc normalement cette variable peut rester vide.

ISDN_CIRC_x_DEBUG Pour avoir des informations débogage supplémentaire par exemple sur `ippdpd`, vous devez activer la variable `ISDN_CIRC_x_DEBUG` régler celle-ci sur 'yes'. Ces informations supplémentaire sur `ippdpd` seront enregistrées avec interface `syslogd`

IMPORTANT : pour que le démon `syslogd` fonctionne, il faut activer la variable `OPT_SYSLOGD` régler celle-ci sur 'yes' (Voir [OPT_SYSLOGD - Enregistrement des messages erreur système](#) (Page 70)).

Il faut aussi activer `klog` pour déboguer certains messages par exemple sur `ISDN`, il faut activer la variable `OPT_KLOGD` régler celle-ci sur 'yes' (Voir [OPT_KLOGD - Kernel-Message-Logger](#) (Page 72)).

Avec le Circuit-Raw-IP la variable `ISDN_CIRC_x_DEBUG` n'a pas d'importance.

ISDN_CIRC_x_AUTH Avec cette variable vous pouvez avoir une authentification PAP ou CHAP lors de la connexion et la communication avec votre correspondant, pour cela paramétrer `ISDN_CIRC_x_AUTH` sur 'pap' ou 'chap' - Et ensuite *nur*. Dans la plus par des cas vous pouvez laisser cette variable vide!

La raison : les fournisseurs accès internet refuse souvent l'authentification, là rejette même! L'exceptions confirment la règle, j'ai récemment lu dans le i4l-Mailingliste ...

Bien sur, il faut entrer le nom d'utilisateur et mot de passe, dans les variables `ISDN_CIRC_x_USER` et `ISDN_CIRC_x_PASS` pour l'utilisation.

Avec le circuits-Raw-IP, cette variable n'a aucune importance.

ISDN_CIRC_x_HUP_TIMEOUT Avec cette variable `ISDN_CIRC_x_HUP_TIMEOUT` on paramètre le temps après lequel ordinateur fli4l doit se déconnecter du fournisseur d'accès, s'il n'y a aucune transmission sur le réseau. Dans notre exemple la déconnexion Idle-Time se fait après 40 secondes, pour économiser de l'argent. S'il y a de nouveau une transmission sur le réseau la connexion se rétablira en une fraction de seconde. Il faut aussi que le FAI calcul à la seconde près!

Il faudrait, au moins dans la phase de test de choisir/la déconnexion automatique du routeur fli4l (soit sur la console ou soit avec le client `imonc` pour Windows), et de vérifier, si vous avez pas une configuration défectueuse du raccordement `ISDN`.

Si vous avez paramétré la valeur '0' aucun temps Idle-Time n'est pris en compte, c.-à-d. que fli4l ne se déconnectera plus de lui-même. S'il vous plaît appliquer cette valeur avec prudence.

ISDN_CIRC_x_CHARGEINT On utilise cette variable pour placez un espace temps : on paramètre ici le coût par unité téléphonique en seconde. Pour avoir le prix total des communications.

En Allemagne les plus grands FAI facture l'unité Tél. exactement à la minute, le paramètre correct dans la variable est donc '60'. Compuserve facture l'unité toute les 3 minutes (juin 2000), on paramètre alors la variable ISDN_CIRC_x_CHARGEINT='180'. Certain FAI facture l'unité exactement à la seconde (par ex. Planet-Interkom) dans ce cas la variable ISDN_CIRC_x_CHARGEINT sera de '1'.

La variable est à ISDN_CIRC_x_CHARGEINT >= 60 Secondes :

Cette variable ISDN_CIRC_x_HUP_TIMEOUT se paramètre en seconde pour la coupure de la connexion si aucun trafic. Raccroche 2 secondes avant la fin de la pulsation téléphonique. Le calcul du temps d'accès sera donc presque entièrement exploités. Une fonctionnalité vraiment fantastique de isdn4linux !

Si la facturation de l'unité est calculé à la seconde, bien sûr cette variable n'a pas de sens - Donc cette règle ne s'applique qu'à partir de 60 secondes par unité de Tél.

ISDN_CIRC_x_TIMES Dans cette variable on paramètre temps activation et l'arrêt de la connexion, aussi le prix de l'unité Tél. Il est possible d'activer des circuits 'Default-Route' différents et aussi l'utilisation de (Least-Cost-Routing). contrôle la route affectation avec le démon (ou programme) imond.

Structure de la variable :

```
ISDN_CIRC_x_TIMES='times-1-info [times-2-info] ...'
```

Il y a dans chaque times-?-info 4 sous-paramètres - Cet sous-paramètres sont séparés par deux points (':').

1. Sous-paramètre : W1-W2

On indique ici les périodes des jours ouvrables, par ex. Mo-Fr ou Sa-Su, il est possible décrire les jours en Anglais ou en Allemand. Si l'on paramètre un seul jour, il sera écrit W1-W1 par ex. Su-Su.

2. Sous-paramètre : hh-hh

On indique ici la période horaire, par ex. 09-18 ou 18-09. De 18-09 est synonyme à 18-24 plus 00-09. de 00-24 correspond à toute une journée.

3. Sous-paramètre : Charge

On indique ici le prix par minute de connexion ou par unité téléphonique en euro, par ex. 0.032 correspond à 3.2 Centimes par minute. Les unités téléphonique, sont calculées en tenant compte du temps de conversion pour un coût réel, et seront alors affiché dans le client-imonc.

4. Sous-paramètre : LC-Default-Route

Le contenu de ce sous-paramètre peut être Y ou N. cela signifie :

- Y : autorise la plage horaire et LC-Routing (ou calcul des frais) avec Default-Route (ou routage par défaut). Important : Dans ce cas, il faut aussi que la variable soit réglée ISDN_CIRC_x_ROUTE='0.0.0.0/0' comme ceci !
- N : autorise la plage horaire et le calcul des frais Tél automatiquement avec LC-Routing, il n'est pas utilisé pour autre chose.

Exemple :

4. Les paquetages

```
ISDN_CIRC_1_TIMES='Mo-Fr:09-18:0.049:N Mo-Fr:18-09:0.044:Y Sa-Su:00-24:0.044:Y'  
ISDN_CIRC_2_TIMES='Mo-Fr:09-18:0.019:Y Mo-Fr:18-09:0.044:N Sa-Su:00-24:0.044:N'
```

Interprétation de l'exemple ci-dessus : Circuit 1 (FAI Planet-Interkom) est utilisé le soir des jours ouvrables et toute la journée en fin de semaine, mais durant la journée les jours ouvrables de la Circuit 2 (Provider Compuserve) est utilisé.

Important: *les paramètres de la variable ISDN_CIRC_x_TIMES doit couvrir toute la semaine. si ce n'est pas le cas, aucune connexion valide ne peut se produire.*

Si vous avez placé le paramètre ("Y") pour LC-Default-Route-Circuits et que vous n'avez pas réglé la semaine complète, il y aura des interruptions dans la période de la semaine avec Default-Route. Alors il sera impossible de surfer sur Internet pendant cet périodes !

Exemple :

```
ISDN_CIRC_1_TIMES='Sa-Su:00-24:0.044:Y Mo-Fr:09-18:0.049:N Mo-Fr:18-09:0.044:N'  
ISDN_CIRC_2_TIMES='Sa-Su:00-24:0.044:N Mo-Fr:09-18:0.019:Y Mo-Fr:18-09:0.044:N'
```

Dans cette exemple les jours ouvrables de 18-09 Heure son paramétrés sur "N". Il n'y a pas de défaut route pour Internet : le surf est interdit !

Encore un exemple simple :

```
ISDN_CIRC_1_TIMES='Mo-Su:00-24:0.0:Y'
```

Cette exemple est pour ceux qui utilise un Flatrate (ou forfait d'accès internet illimité).

Encore une derrière remarque pour le LC-Routing :

Les jours fériés allemands sont traités comme un dimanche.

4.13.5. OPT_TELMOND - Configuration telmond

On utilise la variable OPT_TELMOND pour activer le serveur-telmond. Il écoute via le port TCP 5001 les appels téléphonique entrant et enregistre les informations, numéro de Tél. et le nom du correspondant. Ces informations pourront être visualisées avec le client imonc sous Windows et Unix/Linux (voir le chapitre "Client-/Interface-Serveur imonc").

Condition impérative : avoir installé une carte ISDN (carte numéris), et avoir configuré les variables du paquetage OPT_ISDN.

On peut vérifier le fonctionnement en cours de telmond sous Linux/Unix/Windows avec la commande :

```
telnet fli4l 5001
```

Vous devez voir le dernier appel Tél. puis vous verrez la liaison-telnet se fermer.

Le port 5001 est uniquement accessible depuis le LAN (réseau local). Par défaut la configuration du Firewall bloque l'accès de ce port de l'extérieur. Si vous voulez modifier accès du port pour le réseau LAN, cela est possible utilisé la variable de configuration de telmonc, voir ci-dessous.

Configuration par défaut : START_TELMOND='yes'

TELMOND_PORT Telmond écoute via le Port-TCP/IP. La valeur par défaut est '5001' et devrait être modifié seulement dans certains cas exceptionnels.

TELMOND_LOG Si on paramètre la variable sur `TELMOND_LOG='yes'` l'ensemble des appels téléphoniques seront sauvegardés dans le fichier `/var/log/telmond.log`. Le contenu du fichier d'imond peut être disponible avec le Client-imonc sous Unix/Linux et Windows.

Vous pouvez configurer des chemins différents pour le fichier log, voir ci-dessous.

Configuration par défaut : `TELMOND_LOG='no'`

TELMOND_LOGDIR Si vous avez activé les sauvegarde-log, vous pouvez paramétrer cette variable `TELMOND_LOGDIR` et configurer un autre répertoire d'origine `/var/log`, Par ex. `'/boot'`. Alors, les fichiers LOG de `telmond.log` seront sauvegardés sur un support de boot, il faut qu'il soit "monté" et paramétré en Read/Write. Si vous indiquez `'auto'`, le fichier log sera enregistré en fonction de la configuration dans `/boot/persistant/isdn` ou un autre chemin d'accès spécifique avec la variable `FLI4L_UUID`. Si le chemin `/boot` n'est pas monté en lecture/écriture, le fichier log sera créé dans le répertoire `/var/run`.

TELMOND_MSN_N Ici on peut filtrer les appels téléphoniques uniquement pour certains PC clients, ils seront visibles dans imonc, pour chaque appel spécial, le MSN (ou les numéros de téléphones internes) sera suivi du protocole PC client.

Si c'est nécessaire, Par ex. une collocation du Bat, vous pouvez paramétrer le nombre de filtre MSN dans la variable `TELMOND_MSN_N`.

Configuration par défaut : `TELMOND_MSN_N='0'`

TELMOND_MSN_x Vous devez indiquer pour chaque Filtre-MSN (ou les numéros de téléphones internes) une adresse IP, les appels seront ainsi enregistrés et visibles.

Si la variable `TELMOND_MSN_N` est configurée avec le nombre de filtre MSN.

Structure de la variable :

```
TELMOND_MSN_x='MSN IP-ADDR-1 IP-ADDR-2 ...'
```

Exemple simple :

```
TELMOND_MSN_1='123456789 192.168.6.2'
```

Si vous voulez un appel MSN (ou Tél. interne) sur plusieurs ordinateurs visibles, par ex. envoyer un Fax sur plusieurs PCs, écrire les adresses-IP et les séparer par un espace, exemple :

```
TELMOND_MSN_1='123456789 192.168.6.2 192.168.6.3'
```

TELMOND_CMD_N Dès qu'un appel téléphonique MSN entre, certaines commandes facultatives peuvent être exécutées sur le routeur fli4l. On configure ici le nombre de commande dans la variable `TELMOND_CMD_N`.

TELMOND_CMD_x Avec la variable `TELMOND_CMD_1` bis `TELMOND_CMD_n` on peut configurer les commandes, elles seront exécutées si un appel téléphonique entre.

Si la variable `TELMOND_CMD_N` est configurée avec le nombre de commande.

Structure de la variable :

```
MSN CALLER-NUMBER COMMAND ...
```

Le numéro MSN doit être paramétré sans séparé l'indicatif. A la place de `CALLER-NUMBER` on indique le numéro de Tél. complet - C'est-à-dire le numéro de téléphone avec l'indicatif. Si on écrit à la place de `CALLER-NUMBER` le caractère astérisque (*), telmond n'exploite aucun numéro de téléphone du correspondant.

Voir l'exemple :

4. Les paquetages

```
TELMOND_CMD_1='1234567 0987654321 sleep 5; imonc dial'  
TELMOND_CMD_2='1234568 * switch-on-coffee-machine'
```

Dans le premier cas, la commande "sleep 5; imonc dial" est exécuté si le correspondant avec numéro de Tél 0987654321 appelle le numéro MSN 1234567. En fait il y a 2 commandes. Tout d'abord, on attend 5 secondes, de sorte à libérer canal ISDN, sur lequel l'appel téléphonique entrera. Ensuite, le Client imonc de fli4l démarre avec l'argument "dial". Imonc transmet la commande 1 :1 sur le serveur imond lequel produit une connexion réseau par défaut, par ex. sur Internet. Quelles sont les autres commandes que le programme Client imonc peut envoyer vers le serveur imond, elles sont décrites dans le chapitre "Interface Client-/Serveur imond". Pour que cette option fonctionne, il faut installer OPT_IMONC dans le paquetage "tools".

Dans le deuxième cas la commande "switch-on-coffee-machine" est exécuté, si un appel MSN 1234568 entre, quel que soit, d'où l'appel provient. Naturellement la commande "switch-on-coffee-machine" ne fonctionne pas encore avec fli4l!

lors d'une commande vous pouvez utiliser les jokers suivants :

%d	date	Date
%t	time	Heure
%p	phone	Numéro de Tél du correspondant
%m	msn	MSN spécifique
%%	percent	Pourcentage

Ces données peuvent ensuite être utilisées par un programme, par exemple envoyer par E-Mail.

TELMOND_CAPI_CTRL_N Si vous utilisez un adaptateur ISDN sous CAPI ou un CAPI distant du (type 160 ou 161), il sera peut être nécessaire de configurer le contrôleur CAPI pour que telmond écoute des appels de façon plus explicite. Par exemple, la Fritz!Box offre un accès avec un maximum de cinq contrôleurs différents qui ne peuvent même pas être différenciés (voir les informations sur http://www.wehavemorefun.de/fritzbox/CAPI-over-TCP#Virtuelle_Controller). Pour limiter le nombre de contrôleurs à utiliser vous pouvez définir la quantité, dans le tableau les variables suivantes TELMOND_CAPI_CTRL_% vous pouvez régler les contrôleurs qui doivent être utilisés.

Si vous n'utilisez pas la variable telmond pour écouter sur *tous* les contrôleurs CAPI disponible.

TELMOND_CAPI_CTRL_x Si la variable TELMOND_CAPI_CTRL_N est égal à zéro, l'indice pour les contrôleurs CAPI doit être spécifié pour que telmond surveiller les appels entrants.

Exemple pour un CAPI distant et avec une Fritz!Box pour une "réel" connexion ISDN :

```
TELMOND_CAPI_CTRL_N='2'  
TELMOND_CAPI_CTRL_1='1' # listen to incoming ISDN calls  
TELMOND_CAPI_CTRL_2='3' # listen to calls on the internal S0-Bus
```

Exemple pour un CAPI distant avec une Fritz!Box pour une connexion analogique et SIP-Forwarding :

```
TELMOND_CAPI_CTRL_N='2'  
TELMOND_CAPI_CTRL_1='4' # listen to incoming analog calls  
TELMOND_CAPI_CTRL_2='5' # listen to incoming SIP-calls
```


4.13.6. OPT_RCAPID - Le démon CAPI distant

Avec cette OPT vous pouvez configurer le programme rcapid sur le routeur fli4l qui offre un accès à Interface par le ISDN sous CAPI via des routeurs sur le réseau. Les outils appropriés peuvent accéder sur la carte ISDN du routeur via le réseau comme s'il était installé localement. Ceci est similaire au paquetage "mtgcapri". La différence est que les systèmes Windows peuvent utiliser "mtgcapri" comme un client alors que l'interface réseau de rcapid supporte seulement les systèmes Linux au moment de l'écriture. Ainsi, les deux paquetages sont complémentaires idéaux dans les environnements mixtes Windows et Linux.

Configuration du routeur

OPT_RCAPID Cette variable permet d'activer un ISDN sous CAPI sur le routeur pour les clients distants. Les valeurs possibles sont "yes" et "no". Si la valeur est sur "yes", si le démon inetd sur Internet est configuré, si les demandes de requêtes rcapid sur le port 6000 fonctionne alors le démon rcapid démarre (peut être modifié en utilisant la variable RCAPID_PORT).

Exemple : OPT_RCAPID='yes'

RCAPID_PORT Cette variable contient le port TCP qui est utilisé par le démon rcapid.

Configuration par défaut : RCAPID_PORT='6000'

Configuration du client Linux

Pour utiliser l'interface CAPI distant sur un PC Linux vous devez utiliser le module de bibliothèque libcapi20. Une telle bibliothèque CAPI se trouve dans les dernières distributions Linux (par ex. Debian Wheezy). Sinon vous devez télécharger les sources à partir du lien http://ftp.de.debian.org/debian/pool/main/i/isdnutils/isdnutils_3.25+dfsg1.orig.tar.bz2. Après le dépaquetage et le chargement dans le répertoire "capi20" de la bibliothèque CAPI, il peut être compilé après les trois étapes "configure" "make" et "sudo make install" comme d'habitude. Lorsque la bibliothèque est installée le fichier de configuration /etc/capi20.conf doit être paramétré pour spécifier le client sur lequel rcapid tourne. Par exemple si le routeur est accessible par le nom "fli4l" le fichier conf se présentera comme ceci :

```
REMOTE fli4l 6000
```

C'est tout ! Pour le client Linux, le programme "capiinfo" est installé (fait partie du paquetage capi4k-utils de nombreuses distributions), vous pouvez tester immédiatement l'interface CAPI distant :

```
kristov@peacock ~ $ capiinfo
Number of Controllers : 1
Controller 1:
Manufacturer: AVM Berlin
CAPI Version: 1073741824.1229996355
Manufacturer Version: 2.2-00 (808333856.1377840928)
Serial Number: 0004711
BChannels: 2
[...]
```

Dans "Number of Controllers" la quantité de cartes RNIS est affichée qui peuvent être utilisés par le client. Si vous lisez "0" la connexion au programme rcapid fonctionne mais la carte RNIS n'est pas reconnue sur le routeur. Si la connexion au programme rcapid ne fonctionne pas du tout (peut-être la variable OPT_RCAPID est sur "no"), un message d'erreur "capi not installed - Connection refused (111)" sera affiché. Dans ce cas, recontrôler votre configuration.

4.14. OpenVPN - Supporte le VPN

Depuis la version 2.1.5, le paquetage OpenVPN fait partie intégrante de fli4l.

Important: Avec l'utilisation du paquetage VPN vous pouvez installer un tunnel VPN sur Internet, il est nécessaire d'avoir soit, un forfait d'accès Internet illimité soit un forfait avec un volume heure important ! Le routeur-fli4l reste allumé en permanence, la connexion ne peut pas être coupée, étant donné que les données sont transférées en permanence sur le tunnel (même si c'est quelques octets pendant quelques secondes), les frais de communications seront élevés si vous utilisez un forfait avec un volume d'heures limités, il en va de même si vous Choisissez une connexion ISDN.

Il y a dans la base de données OPT sur le site <http://www.fli4l.de/fr/telechargement/paquetage-annexe/>, en plus du tunnel OpenVPN le paquetage OPT_PoPToP pour la création de tunnel.

Le fait d'opter pour une solution VPN, dépend en premier lieu de la sécurité et des fonctionnalités l'installation. Voici Des rapports sur la sécurité et des solutions pour des réseaux privés virtuels, l'équipe fli4l n'est pas concerné sur ces rapport, voici des pages Web sur ces rapports :

Linux-Magazin numéro janvier 2004

<http://diswww.mit.edu/bloom-picayune/crypto/14238>

<http://sites.inka.de/bigred/archive/cipe-1/2003-09/msg00263.html>

L'équipe fli4l a une position claire sur la fonctionnalité d'OpenVPN. Sur ce point, OpenVPN est le gagnant, le meilleur par rapport à Poptop. OpenVPN supporte avec le tunnel le module Bridge et une compression des données, contrairement à Poptop, en plus il est stable sur le routeur-fli4l. il existe également une version OpenVPN pour Windows, qui peut être utilisé à partir de Windows 2000. Les seuls inconvénients d'OpenVPN par rapport à Poptop est la taille de l'archive-opt et la version 2.0.x de fli4l qui n'est pas supportée par OpenVPN.

4.14.1. OpenVPN - Introduction et exemple

Pour entrer plus facilement dans la configuration, vous pouvez voir dans l'exemple ci-dessous. Deux réseaux qui utilise chacun un routeur-fli4l connecté à Internet. Avec l'installation d'OpenVPN (tunnel codé) sur les routeurs-fli4l, les ordinateurs des deux réseaux pourront communiquer entre eux par Internet dans des villes différentes. Et aussi les variables de configuration dans la figure 4.1.

local net, remote net La figure représente deux réseaux reliés entre eux par un tunnel. Les deux réseaux reliés doivent avoir un TCP/IP différent et ne doivent pas non plus s'entrecroiser avec leurs masques de sous-réseau. Le réglage respectif `IP_NET_x` (Page 38) dans le fichier de configuration base.txt ne doivent pas être le même que le tunnel VPN. Si les deux réseaux utilisent la même adresse IP 192.168.6.0/24 ils ne peuvent pas être liés par un tunnel VPN.

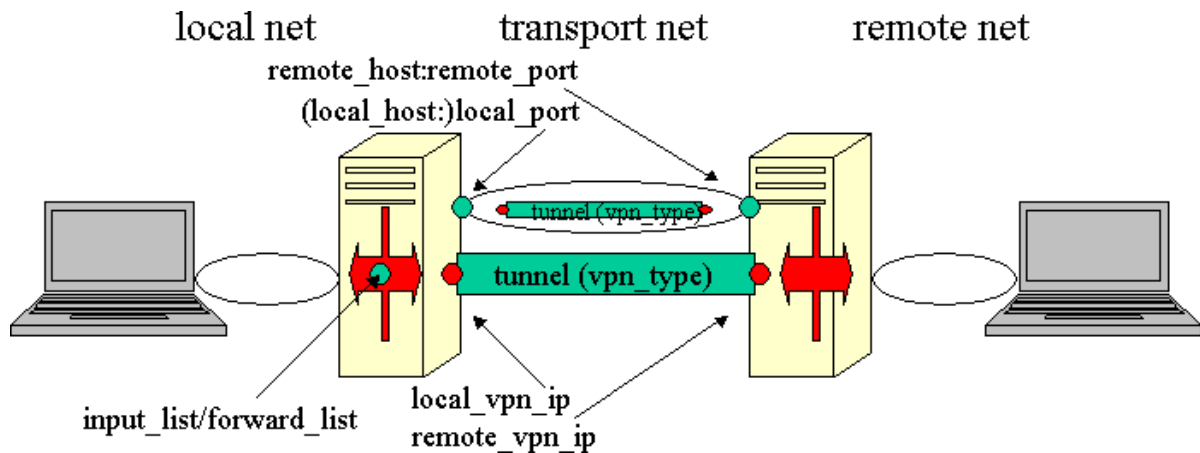


FIGURE 4.1.: Exemple de configuration-VPN — Tunnel entre deux routeurs

transport net Le réseau de transport se compose de deux éléments :

- La connexion entre les deux démons OpenVPN, sont décritent dans *remote_host : remote_port* et *local_host : local_port*. Cela correspond aux paramètres de configuration OpenVPN : `OPENVPN_x_REMOTE_HOST`, `OPENVPN_x_REMOTE_PORT`, `OPENVPN_x_LOCAL_HOST` et `OPENVPN_x_LOCAL_PORT`.
- Le tunnel, sur lequel la liaison entre les démons OpenVPN est établie, sont décritent dans *local_vpn_ip/remote_vpn_ip*. Cela correspond alors à : `OPENVPN_x_LOCAL_VPN_IP` et `OPENVPN_x_REMOTE_VPN_IP`. Les deux adresses-IP sont uniquement utilisé pour le VPN et se trouvent dans les deux routeurs du réseau connus.

Les variables [*input_list* et *forward_list*] servent à filtrer les paquets qui circulent dans le tunnel. Le seule filtre autorisé que l'on peut utiliser et qui permet à tester le tunnel par des messages standards est ICMP (exemple le ping). Dans tout les autres cas, on doit d'abord autoriser les paquets explicitement, voici le cas le plus simple :

```
OPENVPN_x_PF_INPUT_POLICY='ACCEPT'
OPENVPN_x_PF_FORWARD_POLICY='ACCEPT'
```

N'oubliez pas que si vous «ouvrez» complètement les liaisons VPN, cela pourrait être dangereux pour la sécurité. Utilisez plutôt le `tmpl` : fichier de syntaxe pour le filtrage de paquets, afin d'ouvrir uniquement les services que vous avez besoin.

Il n'est pas nécessaire de paramétrer d'avantage de variables pour un simple tunnel VPN. Toutes les autres possibilités de réglage traitent des fonctionnalités avancées, ils sont disponibles pour des applications spéciales. Avec un minimum de réglage le tunnel VPN peut fonctionner, si vous paramétrez les variables avancées celles-ci doivent d'être compatibles pour un bon fonctionnement.

4.14.2. OpenVPN - Configuration

Puisqu'OpenVPN est assez complexe, nous commencerons par les variables obligatoires, nécessaires pour une liaison VPN. Ce n'est que lorsque le routeur-fli4l sera connecté avec ces

paramètres, que vous pourrez vous lancer à configurer les autres variables pour une utilisation étendues d'OpenVPN.

OPT_OPENVPN Par défaut : `OPT_OPENVPN='no'`

Avec 'yes' vous activez le paquetage OpenVPN. Avec 'no' vous désactivez complètement le paquetage OpenVPN.

OPENVPN_N Par défaut : `OPENVPN_N='0'`

Dans cette variable vous indiquez le nombre de configuration OpenVPN à activer.

OPENVPN_x_REMOTE_HOST Par défaut : `OPENVPN_x_REMOTE_HOST=""`

Vous indiquez ici l'adresse-IP ou l'adresse-DNS du poste OpenVPN distant. Dans le cas d'un [Roadwarrior](#) (Page 184) (on peut dire "commerciaux nomade informatisés") vous devez laisser cette variable vide. Si le paramètre est omis, OpenVPN attendra une connexion, il ne tentera pas de se connexion.

OPENVPN_x_REMOTE_HOST_N Par défaut : `OPENVPN_x_REMOTE_HOST_N='0'`

Lorsque l'on utilise un service DNS dynamique, ce service n'est malheureusement pas fiable à 100%. C'est pourquoi qu'il est plus simple d'installer deux ou plusieurs services DNS-Dyn différent et d'enregistrer en même temps une seule adresse-IP actuelle pour tous ces services. Ainsi OpenVPN vérifiera tout les noms DNS-Dyn, dans cette variable on enregistre le nombre de nom de DNS *supplémentaire*. dans la variable `OPENVPN_x_REMOTE_HOST` on enregistrera la liste des adresses, OpenVPN essaiera de contacter cette liste dans un ordre aléatoire. La variable `OPENVPN_x_REMOTE_HOST` doit donc continuer d'exister !

OPENVPN_x_REMOTE_HOST_x Par défaut : `OPENVPN_x_REMOTE_HOST_x=""`

Il s'agit de la même description que la variable [OPENVPN_x_REMOTE_HOST](#) (Page 164) on place ici l'adresse DNS-Dyn ou l'adresse-IP statique.

OPENVPN_x_REMOTE_PORT Par défaut : `OPENVPN_x_REMOTE_PORT=""`

Lorsque OpenVPN est connecté, on a besoin sur le routeur-fli4l un Port qui n'est pas encore utilisé. Il est recommandé d'utiliser les ports au delà de 10000, car ces ports ne sont généralement pas utilisés. Lorsque vous voulez déployer une liaison vers un poste distant, si ce poste a une adresse-IP dynamique et si il n'a pas d'adresse DynDNS, vous pouvez laisser cette variable vide exactement comme la variable `OPENVPN_x_REMOTE_HOST`.

OPENVPN_x_LOCAL_HOST Par défaut : `OPENVPN_x_LOCAL_HOST=""`

On indique ici l'adresse-IP qui doit être connecté à l'OpenVPN. Cette entrée doit rester vide ou complètement omise lors de connexion Internet. Si une adresse-IP est indiqué ici, OpenVPN écoute les demandes de connexion entrante uniquement sur cet adresse-IP. Si vous voulez assurer une connexion WLAN, vous devez enregistrer ici l'adresse-IP de la carte WLAN qui est sur le routeur-fli4l.

OPENVPN_x_LOCAL_PORT Par défaut : `OPENVPN_x_LOCAL_PORT=""`

On indique ici le numéro du Port, sur lequel le démon OpenVPN écoute. Pour chaque configuration d'OpenVPN vous aurez besoin de réserver le port, c.-à-d. que ce port ne peut être utilisé que par la liaison OpenVPN et ne peut être utilisé par aucun autre programme sur le routeur-fli4l. Les paramètres des variables `OPENVPN_x_REMOTE_PORT` et `OPENVPN_x_LOCAL_PORT` doivent être installé pour une liaison OpenVPN ! Si vous paramétrez la variable `OPENVPN_x_REMOTE_PORT='10111'` d'un côté du tunnel, on *doit* impérativement placer de l'autre côté du tunnel dans la variable `OPENVPN_x_LOCAL_PORT='10111'` le même port.

Encore une fois : il est très important d'ajuster ces réglages sur les deux côtés respectifs de l'OpenVPN, autrement la liaison entre les partenaires OpenVPN ne sera pas possible. Afin qu'OpenVPN puisse écouter les connexions entrantes, OpenVPN ouvre indépendamment les ports qui sont indiqués dans la variable `OPENVPN_x_LOCAL_PORT` pour le filtrage de paquets. Si vous ne le souhaitez pas, vous pouvez ajuster les ports dans la variable `OPENVPN_DEFAULT_OPEN_OVPNPORT` (Page 171). Il n'est pas nécessaire d'activer la variable `OPENVPN_DEFAULT_OPEN_OVPNPORT='yes'` puisque c'est le paramètre par défaut !

Il n'est pas possible pour OpenVPN d'écouter des ports en dessous de 1025. Si vous voulez configurer un port en dessous comme par ex. si vous voulez configurer OpenVPN comme serveur-tcp sur le port 443 (port https), vous devez transmettre par le port 443 les paquets filtrés vers un port supérieur à 1024. Par ex votre OpenVPN écoute sur le port 5555 il transmettra les paquets vers le port 443, cela doit être enregistré dans la variable `PF_PREROUTING` comme ceci.

```
PF_PREROUTING_5='tmpl:https dynamic REDIRECT:5555'
```

OPENVPN_x_SECRET Par défaut : `OPENVPN_x_SECRET=""`

OpenVPN a besoin pour crypter une connexion OpenVPN d'un soi-disant fichier clé. Ce fichier clé peut être produit directement avec OpenVPN sous Windows ou Linux. Pour les débutants, vous avez soit le logiciel OpenVPN sous Windows ou soit le WebGUI, interface graphique pour OpenVPN. Si vous ne voulez pas utiliser OpenVPN sous Windows, mais créer seulement le ou les fichiers clés pour OpenVPN, il suffit, d'installer ces quelques fichiers *OpenVPN User-Space Components*, *OpenSSL DLLs*, *OpenSSL Utilities*, *Add OpenVPN to PATH* et *Add Shortcuts to OpenVPN*. Au démarrage d'OpenVPN vous avez dans le menu la commande *Generate a static OpenVPN key*, qui est nécessaire pour produire le fichier clé. Après avoir activé cette commande dans le menu, un message apparaît «Randomly generated 2048 bit key written to C:/Programme/OpenVPN/config/key.txt». Le fichier `key.txt` sera créé il est essentiel pour l'utilisation de ce fichier, de copier celui-ci dans le répertoire `<config>/etc/openvpn` et de renommer le fichier `key.txt` de façon à ce que le nom du fichier soit significatif pour être placé dans cette variable.

Vous pouvez aussi produire le fichier clé automatiquement au démarrage du routeur-fli4l, pour cela, vous devez mettre la variable `OPENVPN_CREATE_SECRET` sur 'yes'. Si vous configurez pour la première fois OpenVPN, vous devez enregistrer toutes les données du fichier config et placer la variable `OPENVPN_DEFAULT_CREATE_SECRET` (Page 170) sur 'yes', vous pouvez produire plusieurs fichiers clés pour plusieurs liaisons OpenVPN ou produire qu'un seul fichier clé pour une liaison OpenVPN, pour cela vous devez placer la variable `OPENVPN_x_CREATE_SECRET` sur 'yes'. Après le démarrage du routeur-fli4l, le ou les fichiers clé(s) seront alors produits automatiquement et placés ces fichiers dans le dossier `/etc/openvpn` avec leur nom respectif, voir ci-dessous. Le ou les fichiers clé(s) peuvent alors être transféré avec le SCP ou copié sur le support de boot. Vous devez remplacer la variable de création de fichier clé sur 'no'. Ne laissez pas la variable de création de clé sur 'yes' car à chaque redémarrage du routeur-fli4l de nouveaux fichiers clés seront produits par le démon OpenVPN. Il sera alors impossible de déployer le tunnel VPN.

Si vous voulez utiliser l'interface Web pour produire le ou les fichiers clé(s), vous devez mettre la variable `OPENVPN_x_CREATE_SECRET` sur 'webgui'. Vous devez vous connecter sur l'interface Web, dans la fenêtre générale sélectionner gestion clé. Pour plus de précision vous pouvez aller voir le paragraphe 4.14.6.

Astuce : avec la commande

4. Les paquetages

```
openvpn --genkey --secret <dateiname>
```

vous pouvez produire par ligne de command un fichier clé sur la console du routeur-fli4l. Les fichiers clés doivent être copiés dans le dossier `<config>/etc/openvpn` comme indiqué dans l'illustration ci-dessous. Le nom du ou des fichiers clés doivent ensuite être placé dans la variable `OPENVPN_x_SECRET`. Alors, les fichiers clés seront compactés et intégrés dans le fichier `opt-Archives`.

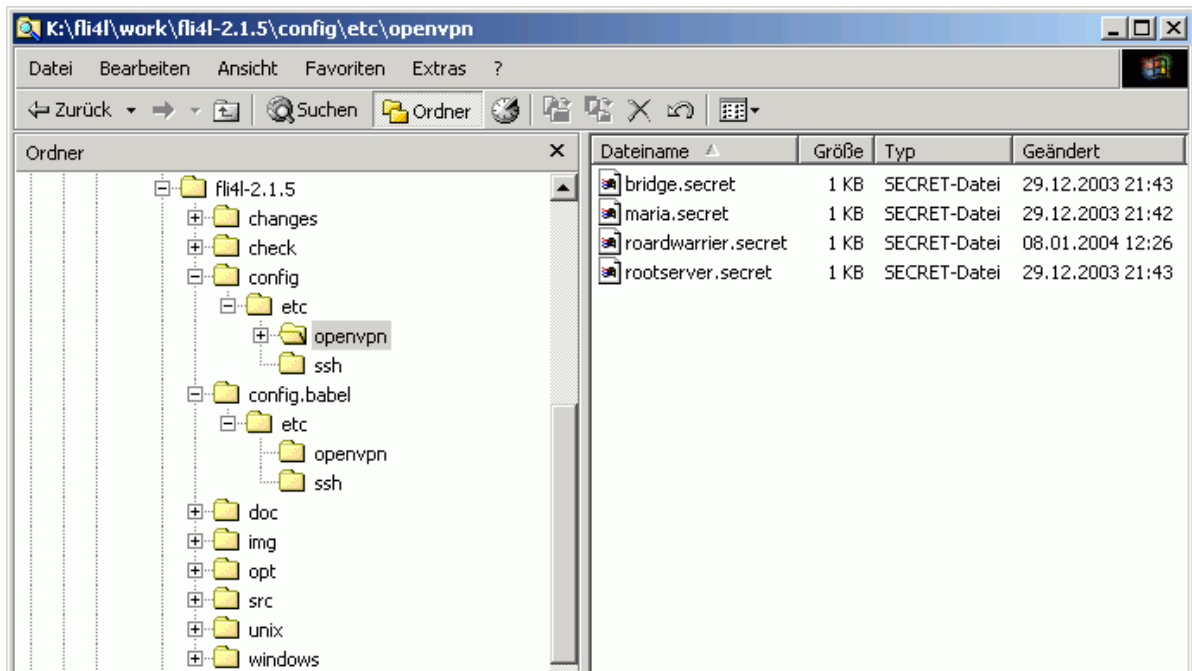


FIGURE 4.2.: fli4l répertoire OpenVPN avec les fichiers *.secret

OPENVPN_x_TYPE Par défaut : `OPENVPN_x_TYPE=""`

On peut utiliser une liaison OpenVPN soit par tunnel, soit par Bridge. OpenVPN par tunnel utilise exclusivement le trafic IP routé. OpenVPN par bridge, le transfert se fait non seulement en trafic IP et aussi en frames Ethernet, par ex. avec le protocole IPX ou NetBEUI. Si vous utilisez OpenVPN avec le transport frames Ethernet, dans tous les cas le paquetage `advanced_networking` est nécessaire. Veuillez considérer que l'utilisation du Bridging avec une connexion DSL peut être lente !

4.14.3. OpenVPN - Configuration du bridge

Si vous utilisez OpenVPN par Bridge, vous pouvez paramétrer les entrées suivantes, N'oubliez pas, que lors de l'utilisation d'un Bridge sur Internet, avec le trafic Broadcast on a besoin d'une bande passante relativement élevée.

Considérer que les réglages suivants ne sont valables que si la variable `OPENVPN_x_TYPE` (Page 166) a été paramétrée sur 'bridge' pour une liaison OpenVPN ! En outre, la configuration du bridge dans le paquetage `advanced_networking` est nécessaire, auquel cas la liaison VPN se bloque.

OPENVPN_x_BRIDGE Par défaut : OPENVPN_x_BRIDGE=""

On indique ici, le nom du Bridge, avec lequel la liaison OpenVPN doit se faire. Donc si dans la variable `BRIDGE_DEV_x_NAME='cuj-br'` du paquetage `advanced_networking` le nom est `'cuj-br'`, vous devez indiquer également le même nom ici pour avoir une liaison OpenVPN par Bridge valide.

OPENVPN_x_BRIDGE_COST Par défaut : OPENVPN_x_BRIDGE_COST=""

Si vous utilisez STP (voir http://de.wikipedia.org/wiki/Spanning_Tree ou la documentation dans le paquetage `advanced_networking`) vous pouvez indiquer ici le coût de la connexion.

OPENVPN_x_BRIDGE_PRIORITY Par défaut : OPENVPN_x_BRIDGE_PRIORITY=""

Si vous utilisez STP (voir http://de.wikipedia.org/wiki/Spanning_Tree ou la documentation dans le paquetage `advanced_networking`) vous pouvez indiquer ici la priorité de la connexion.

4.14.4. OpenVPN - Configuration du tunnel

OPENVPN_x_REMOTE_VPN_IP Par défaut : OPENVPN_x_REMOTE_VPN_IP=""

Considérer que les réglages suivants ne sont valables que si la variable `OPENVPN_x_TYPE` (Page 166) a été paramétrée sur `'tunnel'` pour une liaison OpenVPN!

Adresse-IP VPN du poste éloigné pour une liaison OpenVPN. Les adresses-IP VPN sont nécessaires et peuvent être choisies presque librement. Ci-dessous les restrictions pour le choix d'une adresse-IP VPN sont les suivantes :

- L'adresse-IP ne peut pas être utilisée dans le réseau local. Elle ne peut pas non plus se trouver dans le sous-réseau du routeur-fli4l.
- L'adresse-IP ne peut pas être utilisé pour la restauration système par le réseau.
- L'adresse-IP ne peut pas faire partie d'un réseau `IP_ROUTE_x`.
- L'adresse-IP ne peut pas faire partie d'un réseau `ISDN_CIRC_ROUTE_x`.
- L'adresse-IP ne peut pas faire partie d'un réseau `CIPE_ROUTE_x`.
- L'adresse-IP ne peut pas faire partie d'un réseau `OPENVPN_ROUTE_x`.
- Il est entendu que l'adresse IP ne doit pas appartenir à un réseau fli4l ou à un réseau du routeur-fli4l.

Comme vous le voyez, l'adresse-IP ne peut être utilisée nulle part ailleurs. Avant que vous commenciez la configuration d'OpenVPN, vous devez chercher une adresse-IP qui n'est pas utilisée par de réseau, avec laquelle vous pouvez installer une liaison VPN. L'adresse-IP du réseau devrait aussi absolument faire partie d'un réseau privé (voir <http://ftp.univie.ac.at/netinfo/rfc/rfc1597.txt>).

OPENVPN_x_LOCAL_VPN_IP Par défaut : OPENVPN_x_LOCAL_VPN_IP=""

Ce réglage est valable, que si la variable `OPENVPN_x_TYPE` (Page 166) est paramétrée sur `'tunnel'` pour pouvoir régler une liaison OpenVPN.

On indique ici l'adresse-IP de l'OpenVPN local périphérique tunX. Le choix de l'adresse IP est soumis à la même restriction que la variable `OPENVPN_x_REMOTE_VPN_IP` (Page 167). Il est d'ailleurs possible d'utiliser pour toutes les liaisons OpenVPN locale, la même adresse IP qui est dans la variable `OPENVPN_x_LOCAL_VPN_IP`. Ainsi, il est plus facilement pour un hôte utilisé la même adresse IP dans le VPN. Cela simplifie énormément les règles de filtrage de paquets.

OPENVPN_x_IPV6 Par défaut : `OPENVPN_x_IPV6='no'`

Avec cette variable vous pouvez activer le support IPv6 natif pour OpenVPN. Cette programmation est assez nouvelle et peut être qualifiée d'expérimentale. Pour cela vous devez activer le paquetage `OPT_IPV6` et le configurer. Si vous indiquez `OPENVPN_x_IPV6='no'` et/ou `OPT_IPV6='no'` ces variables sont en relations, elle sera ignorée.

Attention! Actuellement, il n'existe pas de contrôle si les informations se chevauchent avec d'autres parties de la configuration! Ceci s'applique aux variables `OPENVPN_x_LOCAL_VPN_IPV6`, `OPENVPN_x_REMOTE_VPN_IPV6` et `OPENVPN_x_ROUTE_x`.

OPENVPN_x_REMOTE_VPN_IPV6 Par défaut : `OPENVPN_x_REMOTE_VPN_IPV6=""`

La variable IPv6 est égale à celle-ci `OPENVPN_x_REMOTE_VPN_IP` (Page 167).

```
OPENVPN_X_REMOTE_IPV6='FD00::1'
```

OPENVPN_x_LOCAL_VPN_IPV6 Par défaut : `OPENVPN_x_LOCAL_VPN_IPV6=""`

La variable IPv6 est la même que la variable `OPENVPN_x_LOCAL_VPN_IP` (Page 167). Si vous n'indiquez pas de sous-réseau, /64 sera automatiquement utilisé comme sous-réseau.

```
OPENVPN_X_LOCAL_IPV6='FD00::2/112'
```

OPENVPN_x_ROUTE_N Par défaut : `OPENVPN_x_ROUTE_N=""`

Ce réglage n'est valable, que si la variable `OPENVPN_x_TYPE` (Page 166) est paramétrée sur `'tunnel'` pour pouvoir régler une liaison OpenVPN.

Les routes (ou les destinations) données seront lues automatiquement, aussitôt qu'OpenVPN est démarré. On indique ici le nombre de route, on peut mettre jusqu'à 50 réseaux pour une liaison OpenVPN. Il faudra tout de même pour chaque réseau indiquer une adresse-IP dans une variable différente `OPENVPN_x_ROUTE_x` pour le routeur.

Veuillez noter que vous devez paramétrer des règles de filtrage pour les paquets dans les variables `OPENVPN_PF_FORWARD_x`, `OPENVPN_PF_INPUT_x` ou `varOPENVPN_PF6_FORWARD_x`, `OPENVPN_PF6_INPUT_x`. OpenVPN permet uniquement l'envoi ICMP sur une liaison VPN tout autre flux de données est interdit. Vous trouverez plus de détails dans `OPENVPN_x_PF_INPUT_N` (Page 176) et `OPENVPN_x_PF_FORWARD_N` (Page 177) ou sous `OPENVPN_x_PF6_INPUT_N` (Page 177) et `OPENVPN_x_PF6_FORWARD_N` (Page 177).

OPENVPN_x_ROUTE_x Par défaut : `OPENVPN_x_ROUTE_x=""`

Vous devez indiquer ici les adresses-IP des réseaux, qui seront accessibles par le biais du poste éloigné de la liaison VPN. Par ex. les postes éloignés du tunnel OpenVPN veulent atteindre les réseaux 192.168.33.0/24 et 172.18.0.0/16, vous devez respectivement enregistrer ces deux réseaux dans `OPENVPN_x_ROUTE_x`. Vous pouvez également enregistrer un hôte à router avec la valeur (/32).

Si une route par défaut doit être paramétrée via un tunnel OpenVPN, entrez s.v.p. 0.0.0.0/0 ou ::/0 et un flag (ou option) optionnel pour la route. Encore une fois, vous devez activer `OPT_IPV6` pour les routes IPv6, l'adresse IPv6 locale et distante du tunnel doivent être établies et la variable `OpenVPN_x_IPV6` être sur `yes`. OpenVPN reconnaît les différentes possibilités de route par défaut mise en place, pour lesquels, vous pourrez choisir un flag. Chaque méthode qui définit une route par défaut, a ces avantages et ces inconvénients. Pour le moment, OpenVPN prend en charge les flags suivantes :

local Vous devez utiliser le flag *local*, si avec openVPN se trouve un serveur à l'intérieur d'un sous-réseau qui soit directement accessible du routeur-fli4l. Ce cas est fréquent, par exemple pour l'installation d'une route par défaut sur OpenVPN avec un serveur WLAN.

def1 Avec ce flag on peut enregistrer dans OpenVPN une route d'un hôte supplémentaire, par exemple 0.0.0.0/1 et 128.0.0.0/1 deux nouvelles routes. Ces deux routes fonctionnent comme une seule route par défaut et sur ces routes OpenVPN pourra transférer par le tunnel le trafic complètement (crypté) avec (en plus une route pour un hôte accessible).

Ces flags sont facultatives, pour l'installation de chacune des méthodes avec une route par défaut sur OpenVPN. Le choix de la méthode se fera sur la version actuelle OpenVPN, l'option *local* est utilisé pour une installation standard.

```
OPENVPN_1_ROUTE_N='3'  
OPENVPN_1_ROUTE_1='192.168.33.0/24'  
OPENVPN_1_ROUTE_2='172.18.0.0/16'  
OPENVPN_1_ROUTE_3='2001:db8:/32'
```

OpenVPN - Délégation de DNS et DNS-inversé

OPENVPN_x_DOMAIN Par défaut : OPENVPN_x_DOMAIN=""

Vous indiquez dans cette variable un domaine distant. Cette variable peut contenir plusieurs noms de domaine, vous devez les séparer par un espace. Si cette variable est uniquement paramétrée (sans indiquer de serveur DNS supplémentaire), alors, on suppose que l'ordinateur opposé dans le tunnel écoute l'adresse-IP du serveur DNS, (voir [OPENVPN_x_REMOTE_VPN_IP](#) (Page 167)). Bien sûr, pour cela, il faut que sur le routeur distant les requêtes-DNS entrantes soit acceptés, (par exemple via la variable OPENVPN_x_INPUT_y='tpl:dns ACCEPT').

OPENVPN_x_ROUTE_x_DOMAIN Par défaut : OPENVPN_x_ROUTE_x_DOMAIN=""

Dans cette variable différents sous-réseaux peuvent également être associés à différents domaines. Avec la variable OPENVPN_x_ROUTE_y vous pouvez configurer d'autre serveur de domaine. Si la variable OPENVPN_x_ROUTE_y_DNSIP est paramétrée et si le serveur existe il sera utilisé, dans le cas contraire, c'est le serveur de la variable OPENVPN_x_DNSIP qui est utilisé. L'action est la même que la variable OPENVPN_x_DOMAIN de la documentation, cette méthode convient également.

OPENVPN_x_DNSIP Par défaut : OPENVPN_x_DNSIP=""

Si le point de terminaison du tunnel n'a pas de serveur DNS, l'adresse-IP du serveur DNS compétent peut être spécifiée ici. Si vous n'indiquez rien dans cette variable, c'est la variable [OPENVPN_x_REMOTE_VPN_IP](#) (Page 167) qui sera utilisé.

OPENVPN_x_ROUTE_x_DNSIP Par défaut : OPENVPN_x_ROUTE_x_DNSIP=""

Dans cette variable, vous pouvez router différents sous-réseaux, pour desservir différents serveurs DNS - Vous pouvez définir dans la variable [OPENVPN_x_ROUTE_x](#) (Page 168) votre propre serveur compétent.

4.14.5. Paramètres experts

Dans ce chapitre tout les paramètres des variables sont facultatifs et ne doivent être modifiés que si la liaison OpenVPN fonctionne correctement avec les paramètres essentiels, vous

pouvez alors utiliser les paramètres optimisés, (par exemple pour avoir d'autres algorithmes de cryptages).

Tous les paramètres des variables `OPENVPN_DEFAULT_` décrits ci-dessous sont optionnels et agissent sur toutes les configurations OpenVPN, c.-à-d. que ces options n'ont pas besoin d'être rajoutées dans le fichier configuration `openvpn.txt`. Si l'entrée correspondante n'est pas dans le fichier `openvpn.txt`, lors du script de démarrage les valeurs par défaut sont quand même utilisées par OpenVPN. Si vous ne prévoyez pas de modifier les valeurs standard, ne rien écrire de plus dans le fichier de configuration `openvpn.txt` !

Paramètres généraux

OPENVPN_DEFAULT_CIPHER Par défaut : `OPENVPN_DEFAULT_CIPHER='BF-CBC'`

Méthodes de codage disponibles. L'algorithme de chiffrement 'BF-CBC' est utilisé par toutes les versions OpenVPN (aussi par les versions spécifiques de `fl4l`) c'est le réglage standard.

OPENVPN_DEFAULT_COMPRESS Par défaut : `OPENVPN_DEFAULT_COMPRESS='yes'`

OpenVPN utilise la compression de données LZO adaptative, pour augmenter le débit d'une connexion. Adaptatif cela signifie qu'OpenVPN reconnaît indépendamment, si c'est un paquet déjà compressé qui est envoyé sur la liaison OpenVPN par ex. un fichier ZIP. Dans ce cas, la compression de données est mise hors circuit, et sera à nouveau réactivée pour les données qui ont besoin d'une compression avant d'être transférées. Il y a aucune raison pour désactiver la compression de données, car le débit sera augmenté quasi gratuitement. Le seul désavantage de la compression de données est une faible augmentation du temps de latence elle est de quelques millisecondes. Les Online-Games qui jouent par l'intermédiaire du VPN, le temps de réaction ("meilleur" ping) est crucial, dans ce cas il est judicieux de mettre hors circuit la compression de données.

OPENVPN_DEFAULT_CREATE_SECRET Par défaut : `OPENVPN_DEFAULT_CREATE_SECRET='no'`

Avec cette variable, OpenVPN produit automatiquement une clé au démarrage du routeur-`fl4l`. Toutefois, la connexion entre OpenVPN n'est pas commencée. Pour plus de détails, veuillez lire le point suivant [OPENVPN_x_SECRET](#) (Page 165).

OPENVPN_DEFAULT_DIGEST Par défaut : `OPENVPN_DEFAULT_DIGEST='SHA1'`

On paramètre ici, l'algorithme de hachage disponible. OpenVPN utilise la méthode d'algorithme de hachage 'SHA1' comme réglage standard.

OPENVPN_DEFAULT_FLOAT Par défaut : `OPENVPN_DEFAULT_FLOAT='yes'`

Si le poste distant sur une liaison OpenVPN utilise une adresse DynDNS, il est possible qu'à tout moment l'adresse-IP du poste distant change sur OpenVPN. Pour qu'OpenVPN accepte l'adresse-IP modifiée, on doit paramétrer la variable `OPENVPN_DEFAULT_FLOAT` sur 'yes'. Avec le paramètre 'no' la modification d'adresse-IP n'est pas permise. Il est généralement judicieux de placer 'no' avec une liaison WLAN ou avec un poste distant qui a une adresse-IP statique sur une liaison OpenVPN (par ex. pour soumettre différents serveurs root soumissionnaires). Vous pouvez modifier ce paramètre et tous les autres paramètres `OPENVPN_DEFAULT_` pour définir une liaison OpenVPN.

OPENVPN_DEFAULT_KEYSIZE Par défaut : `OPENVPN_DEFAULT_KEYSIZE=""`

La longueur de code (=KEYSIZE) dépend de la méthode de codage utilisée. Modifiez ce réglage, si vous devez travailler en collaboration avec un poste distant d'OpenVPN, qui ne respectent pas les valeurs par défaut utilisées ou que vous ne pouvez pas agir sur

ces paramètres. Alors, vous pouvez déterminer vous-mêmes la longueur clé, cette valeur devrait toujours rester vide. OpenVPN applique une longueur de code optimale pour chaque méthode de codage.

OPENVPN_DEFAULT_OPEN_OVPNPORT Par défaut :

```
OPENVPN_DEFAULT_OPEN_OVPNPORT='yes'
```

Afin qu'un poste distant puisse prendre contact avec vous par OpenVPN, vous devez régler le filtrage de paquets conformément à votre routeur-fil4l. Vous devez généralement paramétrer les protocoles TCP ou UDP dans la variable `OPENVPN_x_PROTOCOL`, ainsi OpenVPN écoutera les adresses que vous avez réglées dans `PF_INPUT_x` (Page 40) du fichier `base.txt`. Avec le paramètre `'yes'` les règles de filtrage de paquets seront produites automatiquement. Pour certaines liaisons VPN, vous pouvez placer la variable sur `'no'` et de définir soi-même les règles du filtrage de paquets correspondants.

OPENVPN_DEFAULT_ALLOW_ICMPING Par défaut :

```
OPENVPN_DEFAULT_ALLOW_ICMPING='yes'
```

Si l'on paramètre `'yes'` dans cette variable, cela permet aux paquets de traverser le filtrage, pour tester la configuration d'une liaison, de sorte que le ping puisse passer le filtrage de paquets. Si vous n'avez pas de raison importante, le Ping ICMP devrait toujours être autorisé. Ce réglage n'a *rien* à voir avec l'option Ping d'OpenVPN !

OPENVPN_DEFAULT_PF_INPUT_LOG Par défaut : `OPENVPN_DEFAULT_PF_INPUT_LOG='BASE'`

Avec `'yes'` ou `'no'` on enregistre ou pas dans un fichier log le détaille du filtrage de paquets INPUT (ou entrant), si le paquet de données est refusé sur une liaison VPN et si vous avez placé `'BASE'` le paramètre de la variable `'PF_INPUT_LOG='` du fichier `base.txt` est pris en charge.

OPENVPN_DEFAULT_PF_INPUT_POLICY Par défaut :

```
OPENVPN_DEFAULT_PF_INPUT_POLICY='REJECT'
```

Ce paramètre correspond à la variable `'PF_INPUT_POLICY='` (Page 50) du fichier `base.txt`. Si vous paramétrez `'BASE'` le paramètre de la variable `'PF_INPUT_POLICY='` du fichier `base.txt` est pris en charge.

OPENVPN_DEFAULT_PF_FORWARD_LOG Par défaut :

```
OPENVPN_DEFAULT_PF_FORWARD_LOG='BASE'
```

Avec `'yes'` ou `'no'` on enregistre dans un fichier log le détaille du filtrage de paquets FORWARD pour une liaison VPN, si vous paramétrez `'BASE'` le paramètre de la variable `'PF_FORWARD_LOG='` du fichier `base.txt` est pris en charge.

OPENVPN_DEFAULT_PF_FORWARD_POLICY Par défaut :

```
OPENVPN_DEFAULT_PF_FORWARD_POLICY='REJECT'
```

Ce paramètre correspond à la variable `'PF_FORWARD_POLICY='` (Page 52) du fichier `base.txt`. Si vous paramétrez `'BASE'` le paramètre de la variable `'PF_FORWARD_POLICY='` du fichier `base.txt` est pris en charge.

OPENVPN_DEFAULT_PING Par défaut : `OPENVPN_DEFAULT_PING='60'`

Une fois le tunnel OpenVPN installé et reconnu, Pour savoir si le poste éloigné est toujours joignable, pour tenir la liaison ouvert, on envoie par intervalle régulier indiqué en secondes un ping cryptographié c'est plus sûr. Avec le paramètre `'off'` aucun ping n'est envoyé sur la liaison OpenVPN, les données seront uniquement transférées sur le tunnel VPN.

OPENVPN_DEFAULT_RENEG_SEC Par défaut : `OPENVPN_DEFAULT_RENEG_SEC='3600'`

Dans cette variable vous pouvez paramétrer le RENE-SEC pour OpenVPN, lorsque vous avez une connexion DSL (ou ISDN) un peu lente, vous n'aurez pas de délai d'attente avant l'arrêt de la connexion.

OPENVPN_DEFAULT_PING_RESTART Par défaut : `OPENVPN_DEFAULT_PING_RESTART='180'`

On indique ici l'intervalle en secondes. S'il n'y a pas de ping ou si aucune donnée n'est transmise avec succès sur la liaison OpenVPN, la liaison OpenVPN correspondante redémarrera. La valeur de la variable `OPENVPN_DEFAULT_PING_RESTART` doit être plus grande que la valeur de la variable `OPENVPN_DEFAULT_PING`. Le paramètre 'off' empêche le redémarrage automatique de la liaison OpenVPN.

OPENVPN_DEFAULT_RESOLV_RETRY Par défaut : `OPENVPN_DEFAULT_RESOLV_RETRY='infinite'`

Dans la variable `OPENVPN_x_REMOTE_HOST` ou `OPENVPN_x_LOCAL_HOST` si le nom du DNS est mis à la place d'adresse-IP, au démarrage de la liaison OpenVPN le nom sera transformé en une adresse-IP. Si la résolution a échoué, OpenVPN essaiera dans la période indiquée en seconde de résoudre à nouveau l'adresse DNS. Finalement s'il ne réussit pas dans le temps alloué, aucune liaison OpenVPN ne sera réalisée. Avec le paramètre 'infinite' (= à l'infini) dans la variable, OpenVPN essaiera infiniment de résoudre le DNS. Ce réglage ne devrait pas être modifié ou uniquement dans des cas particulier !

OPENVPN_DEFAULT_RESTART Par défaut : `OPENVPN_DEFAULT_RESTART='ip-up'`

Après une déconnexion de la liaison, il est logique que le tunnel OpenVPN correspondant redémarre immédiatement, afin que l'interruption du tunnel soit si possible la plus courte possible. Pour toutes les liaisons OpenVPN, qui disposent d'une connexion ADSL ou ISDN, il convient de paramétrer ici 'ip-up'. En revanche pour une liaison OpenVPN via une connexion WLAN (ou sans fil), vous devez paramétrer ici 'never'. Dans ce cas, la connexion ne sera pas redémarrée après une déconnexion, car une connexion WLAN est une connexion indépendante. Si le tunnel OpenVPN est sur une connexion ISDN et si la variable est sur `ISDN_CIRC_x_TYPE='raw'`, vous devez alors enregistrer ici 'raw-up'.

OPENVPN_DEFAULT_PROTOCOL Par défaut : `OPENVPN_DEFAULT_PROTOCOL='udp'`

Avec cette Variable, on règle le protocole par défaut qui doit être utilisée. Le protocole UDP est normalement un très bon choix, toutefois il est possible de travailler avec le protocole TCP. Mais avec celui-ci vous avez une surcharge considérable. Les réglages possibles sont 'udp' ou 'udp6', 'tcp-server' ou 'tcp-server6', 'tcp-client' ou 'tcp-client6'. Les paramètres 'tcp-server' ou 'tcp-client' sont, en règle générale utilisés lorsqu'un tunnel VPN est configuré pour d'autres filtrages de paquets ou pour d'autres tunnels spécifiques, Si vous n'envisagez pas d'utiliser un tunnel spécifique, vous devez *toujours* utiliser la valeur standard 'udp'. Si vous ajoutez '6' le tunnel IPv6 pourra passer par un (WAN) et pourra être accessible via Internet IPv6.

OPENVPN_DEFAULT_START Par défaut : `OPENVPN_DEFAULT_START='always'`

Une liaison OpenVPN peut être soit toujours en fonctionnement (= 'always') ou soit avec un démarrage manuelle (= 'on-demand'). Si vous avez besoin d'arrêter ou démarrer une liaison OpenVPN vous pouvez le faire par l'intermédiaire du WebGUI (voir [4.14.6](#)). Vous pouvez aussi contrôler le démarrage sur la console du routeur-fli4l. Pour cela vous devez écrire les commandes suivantes directement sur la console fli4l et les exporter :

```
cd /etc/openvpn
openvpn --config name.conf --daemon openvpn-name
```

De cette façon, le tunnel OpenVPN sera démarré et fonctionnera à présent en arrière-plan. Naturellement à la place du fichier `name.conf` vous devez mettre le nom de votre fichier de configuration qui est dans le répertoire `/etc/openvpn`.

OPENVPN_DEFAULT_VERBOSE Par défaut : `OPENVPN_DEFAULT_VERBOSE='2'`

Avec cette variable on indique comment OpenVPN doit communiquer. Si la liaison VPN fonctionne parfaitement, il est possible de mettre cette valeur sur `'0'` pour empêcher les messages de débogages. Pour les premiers essais, il est logique de mettre la valeur sur `'3'`. Plus la valeur augmente plus vous avez de messages de débogages et aident parfois à trouver les erreurs. La valeur maximale est de `'11'`.

OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE Par défaut : `OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE='100'`

Cette variable indique le nombre lignes à stocker dans le fichier Log. Ce fichier Log peut alors être consulté dans le [WebGUI](#) (Page 178).

OPENVPN_DEFAULT_MUTE_REPLAY_WARNINGS Par défaut : `OPENVPN_DEFAULT_MUTE_REPLAY_WARNINGS='no'`

Avec cette variable on règle, lors de la réception d'un double paquets une alerte est envoyé dans le fichier Log, car cela, référence peut être à un problème de sécurité dans le réseau. En particulier avec une connexion fragile par WLAN, souvent, il arrive que les paquets soient envoyés deux fois. Il faut afficher judicieusement les avertissements, afin que ceux-ci ne remplissent pas le fichier Log. Le réglage de cette variable n'a *pas* d'influence sur la sécurité d'une liaison OpenVPN.

OPENVPN_DEFAULT_MSSFIX Par défaut : `OPENVPN_DEFAULT_MSSFIX=""`

Dans la variable MSSFIX on paramètre la taille du paquet TCP pour une liaison VPN. Cette option sera désactivée si la variable est sur `OPENVPN_DEFAULT_MSSFIX='0'`. Si les options FRAGMENT et MSSFIX sont laissés vides, la taille de la fragmentation sera utilisée automatiquement. Ce réglage fonctionne que si la variable est paramétrée sur `OPENVPN_x_PROTOCOL='udp'`.

OPENVPN_DEFAULT_FRAGMENT Par défaut : `OPENVPN_DEFAULT_FRAGMENT='1300'`

Active la fragmentation interne de la taille des paquets en x octets sur OpenVPN. Ce réglage fonctionne que si la variable est sur `OPENVPN_x_PROTOCOL='udp'`. Avec le paramètre `OPENVPN_DEFAULT_FRAGMENT='0'`, la fragmentation est totalement désactivé.

OPENVPN_DEFAULT_TUN_MTU Par défaut : `OPENVPN_DEFAULT_TUN_MTU='1500'`

Réglage du MTU en x octets pour l'adaptateur OpenVPN virtuel. Si vous savez ce que vous faite cette option peut être modifiée. Il est plus logique de travailler principalement et seulement avec les options FRAGMENT ou MSSFIX.

OPENVPN_DEFAULT_TUN_MTU_EXTRA Par défaut : `OPENVPN_DEFAULT_TUN_MTU_EXTRA=""`

Si vous avez paramétrer la variable sur `OPENVPN_x_PROTOCOL='bridge'`, 32 octets de mémoires supplémentaires sont réservés sur le routeur pour l'administration de l'amortissement du débit. Avec le paramètre `OPENVPN_x_PROTOCOL='tunnel'` aucune mémoire supplémentaire n'est réservée. Ce réglage ne se répercute que sur le besoin de mémoire dans le routeur et n'a pas d'influence sur le volume de données envoyées sur le tunnel.

OPENVPN_DEFAULT_LINK_MTU Par défaut : `OPENVPN_DEFAULT_LINK_MTU=""`

Réglage du MTU en x octets pour une liaison OpenVPN. Si vous savez ce que vous faites, cette option peut être modifiée. Il est plus logique de travailler principalement et seulement avec les options FRAGMENT ou MSSFIX.

OPENVPN_DEFAULT_SHAPER Par défaut : OPENVPN_DEFAULT_SHAPER=""

Vous pouvez ici limiter le débit *sortant* du tunnel en octet par seconde, les valeurs possibles sont de 100 octets par seconde à 100000000 octets. Avec une valeur de 1000 octets par seconde, vous devriez réduire le MTU de la liaison VPN et le délai du ping augmentera fortement. Si vous voulez limiter le débit dans les deux sens du tunnel, vous devez ajuster le réglage sur les deux postes de chaque côté du tunnel VPN.

Dans la version OpenVPN actuel, le Shaping ne fonctionne pas correctement, c'est à dire que la vitesse de transfert dans un tunnel configuré au moyen du Shaping oscille, il peut-être extrêmement instable ou le débit peut tomber totalement. Le problème peut produire des comportements complètement différents selon le matériel employé. Actuellement, la fonction Shaping doit être utilisé avec prudence, dans le doute, lors de chaque changement, la liaison doit être testée intensivement.

OPENVPN_EXPERT Par défaut : OPENVPN_EXPERT='no'

Le mode-Expert vous permet d'utiliser des fichiers de configurations d'Openvpn natif. Ils doivent être placés dans le sous répertoire de configuration etc/openvpn. Tous les fichiers se trouvant dans ce répertoire seront transmis au routeur.

Le mode Expert ignore le reste des variables de configuration. Il faut donc régler la variable sur OPENVPN_N='0'.

Avec le mode Expert, des règles du Firewall ne sont pas établies. Vous devez les entrer manuellement dans le fichier base.txt

Connexion des paramètres spécifiques

Les options OpenVPN suivantes, s'appliquent uniquement pour les liaisons OpenVPN respectives. Ici aussi Il y a peu de définitions impératives. La plupart des options peuvent simplement être omises. On peut considérer, que toutes valeurs par défaut indiquées dans les variables OPENVPN_DEFAULT_x sont équivalentes aux variables suivantes. Donc si vous modifiez la valeur de la variable OPENVPN_DEFAULT_ correspondante, cette valeur par défaut vaut pour tous les liaisons OpenVPN, en général il ne faut pas écraser la valeur par défaut.

OPENVPN_x_NAME Par défaut : OPENVPN_x_NAME=""

On indique ici le nom de la liaison OpenVPN, la longueur du nom ne doit pas dépasser 16 caractères. Ce nom peut contenir des lettres, des chiffres et le signe '-'. Ce nom de fichier de configuration sera enregistré dans le répertoire /etc/openvpn (avec l'extension .conf). En outre, le nom apparaîtra dans le syslog. Par exemple si vous enregistrez le nom 'peter', l'entrée dans le syslog sera indiqué, 'openvpn-peter'. De cette façon, vous pouvez mieux distinguer les différentes liaisons OpenVPN.

OPENVPN_x_ACTIV Par défaut : OPENVPN_x_ACTIV='yes'

Dans cette variable si vous paramétrez 'no' vous désactivez la liaison OpenVPN, mais vous ne supprimez pas la configuration, Les données de configuration sont alors incluses dans le fichier rc.cfg, mais vous ne pouvez pas produire de liaison OpenVPN.

OPENVPN_x_CHECK_CONFIG Par défaut : OPENVPN_x_CHECK_CONFIG='yes'

Dans certaine circonstance, les contrôles étendus d'OpenVPN sont trop stricts. Par exemple si vous faites un Backup d'une connexion ISDN et si les entrées de routage utilisé sont les mêmes que la liaison vpn via Internet, le contrôle étendu de ces connexions produit un message d'erreur important. Dans ce cas, le Backup de la connexion ISDN sera désactivé. Pour remédier à ce problème, vous devez mettre la variable sur `OPENVPN_x_CHECK_CONFIG='no'` pour passer le contrôle de cette liaison.

OPENVPN_x_CIPHER Par défaut voir : `OPENVPN_DEFAULT_CIPHER`

Voir [OPENVPN_DEFAULT_CIPHER](#) (Page 170). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_COMPRESS Par défaut voir : `OPENVPN_DEFAULT_COMPRESS`

Voir [OPENVPN_DEFAULT_COMPRESS](#) (Page 170). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_CREATE_SECRET Par défaut voir : `OPENVPN_DEFAULT_CREATE_SECRET='no'`

Voir [OPENVPN_x_SECRET](#) (Page 165). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_DIGEST Par défaut voir : `OPENVPN_DEFAULT_DIGEST`

Voir [OPENVPN_DEFAULT_DIGEST](#) (Page 170). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_FLOAT Par défaut voir : `OPENVPN_DEFAULT_FLOAT`

Voir [OPENVPN_DEFAULT_FLOAT](#) (Page 170). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_KEYSIZE Par défaut voir : `OPENVPN_DEFAULT_KEYSIZE`

Voir [OPENVPN_DEFAULT_KEYSIZE](#) (Page 170). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_ISDN_CIRC_NAME Par défaut : `OPENVPN_x_ISDN_CIRC_NAME=""`

On indique ici, le circuit ISDN sur lequel la liaison OpenVPN doit être installée. Le nom du circuit ISDN correspondant est enregistré dans la variable `jumpISDN_CIRC_NAMEISDN_CIRC_x_NAME=""` du fichier `isdn.txt`. Le circuit ISDN doit être du type `'raw'`.

OPENVPN_x_PING Par défaut voir : `OPENVPN_DEFAULT_PING`

Voir [OPENVPN_DEFAULT_PING](#) (Page 171). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_PROTOCOL Par défaut : `OPENVPN_x_PROTOCOL='udp'`

On indique ici le protocole qui doit être utilisé pour un tunnel OpenVPN. Les réglages possibles sont `'udp'`, `'tcp-server'` ou `'tcp-client'`. Les paramètres `'tcp-server'` ou `'tcp-client'` sont, en règle généralement utilisé lorsqu'un tunnel VPN doit être développé pour d'autres filtrages de paquets ou pour d'autres tunnels. Si vous n'envisagez *pas* d'utiliser des tunnels spécifiques, vous devez toujours utiliser la valeur standard `'udp'`.

OPENVPN_x_RESOLV_RETRY Par défaut voir : `OPENVPN_DEFAULT_RESOLV_RETRY`

Voir [OPENVPN_DEFAULT_RESOLV_RETRY](#) (Page 172). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_PING_RESTART Par défaut voir : `OPENVPN_DEFAULT_PING_RESTART`

Voir [OPENVPN_DEFAULT_PING_RESTART](#) (Page 172). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_START Par défaut voir : `OPENVPN_DEFAULT_START`

Voir [OPENVPN_DEFAULT_START](#) (Page 172). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_VERBOSE Par défaut voir : `OPENVPN_DEFAULT_VERBOSE`

Voir [OPENVPN_DEFAULT_VERBOSE](#) (Page 173). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_MANAGEMENT_LOG_CACHE Par défaut voir : `OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE`

Voir [OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE](#) (Page 173). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_MUTE_REPLAY_WARNINGS Par défaut voir : `OPENVPN_DEFAULT_MUTE_REPLAY_WARNINGS`

Voir [OPENVPN_DEFAULT_MUTE_REPLAY_WARNINGS](#) (Page 173). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_RESTART Par défaut voir : `OPENVPN_DEFAULT_RESTART`

Voir [OPENVPN_DEFAULT_RESTART](#) (Page 172). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_ALLOW_ICMPING Par défaut voir : `OPENVPN_DEFAULT_ALLOW_ICMPING`

Voir [OPENVPN_DEFAULT_ALLOW_ICMPING](#) (Page 171). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_OPEN_OVPNPORT Par défaut voir : `OPENVPN_DEFAULT_OPEN_OVPNPORT`

Voir [OPENVPN_DEFAULT_OPEN_OVPNPORT](#) (Page 171). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_PF_INPUT_LOG Par défaut voir : `OPENVPN_DEFAULT_PF_INPUT_LOG`

Voir [OPENVPN_DEFAULT_PF_INPUT_LOG](#) (Page 171). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_PF_INPUT_POLICY Par défaut voir : `OPENVPN_DEFAULT_PF_INPUT_POLICY`

Voir [OPENVPN_DEFAULT_PF_INPUT_POLICY](#) (Page 171). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_PF_INPUT_N Par défaut : `OPENVPN_x_PF_INPUT_N='0'`

Dans cette variable `OPENVPN_x_PF_INPUT_x=` vous indiquez le nombre de variables pour le filtrage de paquets.

OPENVPN_x_PF_INPUT_x Par défaut : `OPENVPN_x_PF_INPUT_x=`

Ici les informations sur le filtrage de paquets sont les même que le paquetage base. On utilise précisément les même syntaxes que dans le fichier base.txt. Il est possible d'utiliser `tmpl` : et `Host_alias`. En plus, on a aussi la possibilité d'utiliser quelques noms symboliques spéciaux. Les noms symboliques suivants seront supportés :

VPNDDEV Correspond au périphérique actuel de la liaison OpenVPN respectif.

LOCAL-VPN-IP Définit l'adresse-IP de la variable `OPENVPN_x_LOCAL_VPN_IP`.

REMOTE-VPN-IP Définit l'adresse-IP de la variable `OPENVPN_x_REMOTE_VPN_IP`.

REMOTE-NET Définit l'adresse IP de la variable `OPENVPN_x_REMOTE_VPN_IP` et en plus tous les réseaux qui ont été indiqués dans la variable `OPENVPN_x_ROUTE_x`.

OPENVPN_x_P_F_FORWARD_LOG Par défaut voir : `OPENVPN_DEFAULT_P_F_FORWARD_LOG`

Voir [OPENVPN_DEFAULT_P_F_FORWARD_LOG](#) (Page 171). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_P_F_FORWARD_POLICY Par défaut voir : `OPENVPN_DEFAULT_P_F_FORWARD_POLICY`

Voir [OPENVPN_DEFAULT_P_F_FORWARD_POLICY](#) (Page 171). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_P_F_FORWARD_N Par défaut : `OPENVPN_x_P_F_FORWARD_N='0'`

Dans cette variable `OPENVPN_x_P_F_FORWARD_x=` vous indiquez le nombre de variables pour le filtrage de paquets.

OPENVPN_x_P_F_FORWARD_x Par défaut : `OPENVPN_x_P_F_FORWARD_x=""`

Voir [OPENVPN_x_P_F_INPUT_x](#) (Page 176).

OPENVPN_x_P_F_PREROUTING_N Par défaut : `OPENVPN_x_P_F_PREROUTING_N='0'`

Dans cette variable `OPENVPN_x_P_F_PREROUTING_x=` vous indiquez le nombre de variables pour le filtrage de paquets.

OPENVPN_x_P_F_PREROUTING_x Par défaut : `OPENVPN_x_P_F_PREROUTING_x=""`

Voir [OPENVPN_x_P_F_INPUT_x](#) (Page 176).

OPENVPN_x_P_F_POSTROUTING_N Par défaut : `OPENVPN_x_P_F_POSTROUTING_N='0'`

Dans cette variable `OPENVPN_x_P_F_POSTROUTING_x=` vous indiquez le nombre de variables pour le filtrage de paquets.

OPENVPN_x_P_F_POSTROUTING_x Par défaut : `OPENVPN_x_P_F_POSTROUTING_x=""`

Un changement de paramétrage est sortie pour cette variable avec la version 3.5.0 de `fi4l` (ou la version 3.5.0-rev18133 du tarball). Auparavant on pouvait paramétrer cette variable sous cette forme

```
OPENVPN_1_P_F_POSTROUTING_1='MASQUERADE'
```

Désormais nous devons spécifier une adresse source et une adresse destination. Cela est devenu nécessaire, car les règles `POSTROUTING` ne pouvaient pas être utilisés pleinement. Dans la plupart des cas, il suffit simplement de compléter la variable avec ces règles [IP_NET_x](#) (Page 38) et `REMOTE-NET`.

Voir [OPENVPN_x_P_F_INPUT_x](#) (Page 176).

OPENVPN_x_P_F6_INPUT_N Par défaut : `OPENVPN_x_P_F6_INPUT_N='0'`

Le numéro indiquez dans `OPENVPN_x_P_F6_INPUT_x=` donne le nombre d'enregistrement de variable.

OPENVPN_x_P_F6_INPUT_x Par défaut : `OPENVPN_x_P_F6_INPUT_x=""`

Comme dans le paquetage IPv6 voici les instructions pour le filtre de paquets. Les syntaxes utilisées sont exactement les mêmes que dans `ipv6.txt`. Il est possible d'indiquer le `tmpl` : et les alias des Hôtes. En outre, il est possible d'utiliser des noms symboliques spéciaux. Voir [OPENVPN_x_P_F_INPUT_x](#) (Page 176)

OPENVPN_x_P_F6_FORWARD_N Par défaut : `OPENVPN_x_P_F6_FORWARD_N='0'`

Le numéro indiquez dans `OPENVPN_x_P_F6_FORWARD_x=` donne le nombre d'enregistrement de variable.

OPENVPN_x_P_F6_FORWARD_x Par défaut : `OPENVPN_x_P_F6_FORWARD_x=""`

Voir [OPENVPN_x_P_F6_INPUT_x](#) (Page 177).

OPENVPN_x_MSSFIX Par défaut voir : `OPENVPN_DEFAULT_MSSFIX`

Voir [OPENVPN_DEFAULT_MSSFIX](#) (Page 173). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_FRAGMENT Par défaut voir : `OPENVPN_DEFAULT_FRAGMENT`

Voir [OPENVPN_DEFAULT_FRAGMENT](#) (Page 173). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_TUN_MTU Par défaut voir : `OPENVPN_DEFAULT_TUN_MTU`

Voir [OPENVPN_DEFAULT_TUN_MTU](#) (Page 173). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_TUN_MTU_EXTRA Par défaut voir : `OPENVPN_DEFAULT_TUN_MTU_EXTRA`

Voir [OPENVPN_DEFAULT_TUN_MTU_EXTRA](#) (Page 173). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_LINK_MTU Par défaut voir : `OPENVPN_DEFAULT_LINK_MTU`

Voir [OPENVPN_DEFAULT_LINK_MTU](#) (Page 173). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

OPENVPN_x_SHAPER Par défaut voir : `OPENVPN_DEFAULT_SHAPER=""`

Voir [OPENVPN_DEFAULT_SHAPER](#) (Page 174). Contrairement à la variable par défaut, ce réglage n'agit que sur cette liaison OpenVPN.

4.14.6. OpenVPN - WebGUI

Depuis la version 2.1.10, il est possible, de configurer, de démarrer, d'arrêter, d'exporter et d'utiliser d'autres fonctions fondamentales sur le Web-GUI pour une liaison OpenVPN. Le paquetage `mini_httpd` est nécessaire à l'installation. En outre, la variable `OPENVPN_WEBGUI` doit être placée sur 'yes' dans `openvpn.txt`. Le menu OpenVPN sera alors ajouté sur la page Web de `fl4l`. Si vous choisissiez ce menu, un aperçu de la configuration des liaisons OpenVPN apparaît, avec le statut et les actions pour chacune des liaisons respectives disponibles (voir l'illustration 4.3).

OpenVPN - WebGUI - Aperçut des connexions

Statut : Le statut d'une liaison est symbolisé par un signal pour piéton. Lorsque le piéton est rouge cela signifie que le processus-OpenVPN ne fonctionne pas, le piéton jaune que le processus ne fonctionne pas (encore) avec le poste éloigné, mais que la liaison peut être activée à tout moment et le piéton vert que la liaison est «établie». Des informations plus précises sont indiquées en dessous des icônes piétons. Cela peut être instructif en particulier, pour le statut du piéton «jaune».

Nom : Dans cette colonne, les noms des liaisons-OpenVPN sont indiqués comme dans la configuration. En cliquant sur le nom, cela vous conduit dans une vue d'ensemble, dans laquelle est indiquée des informations plus précises de la liaison.

Action : Ici, les actions sont symbolisées par des icônes. Que signifie chaque icône ? Voici ci-dessous un tableau récapitulatif :

OpenVPN-Verbindungen		
Status	Name	Aktion
 Verbunden	ktbs	  
 Verbindung getrennt	kthan	
 Verbindung angehalten	wlan-ellen	  
 Verbindung getrennt	wlan-qast	
 Verbindung wird aufgebaut ...	wlan-helmut	  

FIGURE 4.3.: Aperçut des connexions






Symbole	Explication
	Démarrer le processus-OpenVPN et tente de se connecter.
	Arrête le processus-OpenVPN.
	Réinitialise la connexion.
	stoppe la connexion, elle est en attente. Plus aucune donnée ne circule sur la liaison.
	Relance la connexion. Les données peuvent à nouveau circuler sur la liaison.

TABLE 4.9.: Commande du Webgui-OpenVPN



FIGURE 4.4.: Vue détaillée d'une connexion (gestion de clé)

OpenVPN - WebGUI - Vue détaillée d'une connexion

Statistique : on peut voir sur cet onglet, les statistiques intéressantes de la liaison. Les statistiques ne peuvent être visibles que si la liaison est démarrée et non arrêtée.

Log : On peut voir sur cet onglet, les 20 dernières lignes de la connexion. Si vous voulez voir plus de lignes, vous pouvez indiquer le nombre et cliquer sur "afficher". Si vous indiquez "all" vous pouvez voir la totalité du fichier log. Cet onglet est visible que si la liaison est démarrée.

Debug-Log : on peut voir sur cet onglet le processus de démarrage. On voit le démarrage de la liaison-OpenVPN et ces sorties. C'est utile lorsque que l'on veut démarrer la liaison avec l'icône démarrée et que la connexion ne veut pas s'activer, en plus dans le fichier log normal il n'y a rien d'indiqué sur le démarrage.

Filtrage de paquets : on peut voir sur cet onglet le filtrage de paquets valide pour une liaison. Le filtrage de paquets est configuré que si la liaison est démarrée et que le tunnel est installé.

Bridge : on peut voir sur cet onglet la configuration du Bridge sur le routeur. Cet onglet est visible que si la liaison avec Bridge est installée.

Configuration : on peut voir sur cet onglet, la configuration de la liaison générée par le boot.

Gestion de clé : sur cet onglet, on peut produire une clé pour une liaison et peut également

être téléchargés, (voir l'illustration 4.4). Aucune clé n'existe (au premier démarrage) du VPN, vous devez la produire automatiquement. Il peut être transféré directement avec le Symbole-Download ou être copier/coller dans un fichier texte. Cliquez sur l'icône-disquette pour enregistrer la clé qui a été nouvellement produite sur le routeur, ce processus peut-être annulé, par un clic sur l'icône-restauration.

Support informations : sont indiquées dans cet onglet, toutes les informations qui pourraient être pertinentes, si vous avez un problème. Vous pouvez transmettre ces renseignements, par exemple pour un article dans des Newsgroups en faisant un copier/coller.

4.14.7. OpenVPN - Aide pour différentes versions OpenVPN

Avec les versions différentes d'OpenVPN, vous devez veiller à l'utilisation des paramètres, car les valeurs standards diffèrent pour chaque liaison VPN. Cela concerne en particulier les réglages, MTU, FRAGMENT et MSSFIX. Si les valeurs correspondent ne sont pas «adaptées» aux config OpenVPN ou si la connexion fonctionne avec la commande ping, mais se bloque par exemple lors de l'utilisation ssh, alors ces valeurs ne sont peut être pas ajustées correctement. Voici les messages d'erreurs typiques, pour de tels cas :

```
FRAG_IN error flags=0xfa2a187b: FRAG_TEST not implemented
FRAG_IN error flags=0xfa287f34: spurious FRAG_WHOLE flags
```

Les paramètres cruciaux, pour la réalisation d'une liaison sont les suivants :

OPENVPN_x_TUN_MTU La valeur MTU du périphérique TUN est pour la version OpenVPN 1.x à 1300. La valeur à partir de la version OpenVPN 2.0 est de 1500, valeur standard.

OPENVPN_x_LINK_MTU Taille en octet de la liaison des deux démons OpenVPN. Cette valeur par défaut est fonction de l'utilisation de la version OpenVPN et du système d'exploitation.

OPENVPN_x_FRAGMENT Les paquets (peu importe que ce soit UDP ou TCP) dont la taille est au-dessus du seuil de fragmentation, ces paquets de données seront fragmentés, et ne seront pas supérieures à celles indiquées dans la variable **OPENVPN_x_FRAGMENT** en octet.

OPENVPN_x_MSSFIX Afin d'échanger les paquets de données TCP sur une liaison VPN, sans que ces paquets soient si possible fragmentés, vous pouvez indiquer ici la dimension maximale souhaitée des paquets de données TCP. Les systèmes d'exploitation actuels analysent mieux les normes de fragmentation, ainsi la fragmentation des paquets de données n'est plus nécessaire.

Les différentes versions OpenVPN utilisent les valeurs suivantes en tant que valeurs par défaut. Vous devez faire attention à ces valeurs, si vous voulez connecter ces versions OpenVPN qui ne fonctionnent pas sur le routeur-fli4l. Les valeurs par défaut spécifiées sur le routeur-fli4l sont dans le deuxième tableau.

En raison de ces différents paramètres, vous devez déterminer les valeurs par défaut à installer dans votre réseau et écrire alors explicitement ceux-ci dans le fichier config/openvpn.txt. Les valeurs sont dans la plupart des cas, des valeurs satisfaisantes pour des premiers tests.

```
OPENVPN_DEFAULT_TUN_MTU='1500'
OPENVPN_DEFAULT_MSSFIX='1300'
OPENVPN_DEFAULT_FRAGMENT='1300'
```

4. Les paquetages

Option/version OpenVPN	1.xx	2.00
OPENVPN_x_TUN_MTU	1300	1500
OPENVPN_x_TUN_MTU_EXTRA	inconnu	32
OPENVPN_x_FRAGMENT	inconnu	non configuré
OPENVPN_x_MSSFIX	non configuré	1450

TABLE 4.10.: Paramètre MTU des différentes versions OpenVPN.

Option/version fli4l	jusqu'à 2.1.8	à partir 2.1.9
OPENVPN_x_TUN_MTU	1300	1500
OPENVPN_x_TUN_MTU_EXTRA	64	32
OPENVPN_x_FRAGMENT	non configuré	1300
OPENVPN_x_MSSFIX	non configuré	1300

TABLE 4.11.: Paramètre MTU des différentes versions pour le routeur-fli4l.

Malheureusement, il n'est pas possible pour les versions fli4l antérieures à 2.1.9 de paramétrer directement le «tun-mtu». Toutefois, on peut influencer indirectement sur ce paramètre avec la variable OPENVPN_x_LINK_MTU. La valeur tun-mtu est d'environ 45 octets inférieurs à la valeur spécifiée dans OPENVPN_x_LINK_MTU. Pour déterminer la valeur précise, vous devez faire des essais.

4.14.8. OpenVPN - Exemples

Quelques exemples illustrent la configuration du paquetage OpenVPN.

Exemple - Joindre deux réseaux par le routeur-fli4l

Dans le premier exemple, nous allons effectuer une liaison OpenVPN sur deux routeur-fli4l. Il s'agit d'accéder au réseau derrière le routeur-fli4l du poste distant. Dans cet exemple, Peter et Maria veulent relier leurs réseaux par leur routeur-fli4l. Peter utilise l'adresse 192.168.145.0/24 pour le réseau privés et l'adresses peter.eisfair.net pour DynDNS. Maria, utilise de façon semblable l'adresse 10.23.17.0/24 pour le réseau et l'adresse maria.eisfair.net pour DynDNS. Les deux se font confiance mutuellement et pourrons avoir accès à l'ensemble de leurs réseaux.

Option OpenVPN	Peter	Maria
OPENVPN_1_NAME=	'maria'	'peter'
OPENVPN_1_REMOTE_HOST=	'maria.eisfair.net'	'peter.eisfair.net'
OPENVPN_1_REMOTE_PORT=	'10000'	'10001'
OPENVPN_1_LOCAL_PORT=	'10001'	'10000'
OPENVPN_1_SECRET=	'pema.secret'	'pema.secret'
OPENVPN_1_TYPE=	'tunnel'	'tunnel'
OPENVPN_1_REMOTE_VPN_IP=	'192.168.200.202'	'192.168.200.193'
OPENVPN_1_LOCAL_VPN_IP=	'192.168.200.193'	'192.168.200.202'
OPENVPN_1_ROUTE_N=	'1'	'1'
OPENVPN_1_ROUTE_1=	'10.23.17.0/24'	'192.168.145.0/24'
OPENVPN_1_PF_INPUT_N=	'1'	'1'
OPENVPN_1_PF_INPUT_1=	'ACCEPT'	'ACCEPT'
OPENVPN_1_PF_FORWARD_N=	'1'	'1'
OPENVPN_1_PF_FORWARD_1=	'ACCEPT'	'ACCEPT'

TABLE 4.12.: Configuration d'OpenVPN avec 2 routeurs-fli4l

Exemple - deux réseaux reliés par un Bridge (ou pont)

Dans l'exemple suivant, un Bridge est développé par l'intermédiaire une connexion sans fil. Avec un Bridge, le filtrage de paquets ne peut pas être configuré judicieusement, puisque seuls les frames Ethernet sont transmises, absolument pas les paquets IP. Merci de ne pas oublier, qu'un réseau commun doit être utiliser dans une configuration de bridge. En plus, il ne faut pas que l'adresse IP soit attribuée deux fois.

Option OpenVPN	Peter	Maria
OPENVPN_2_NAME	'bridge'	'bridge'
OPENVPN_2_REMOTE_HOST	'10.1.0.1'	'10.2.0.1'
OPENVPN_2_REMOTE_PORT	'10005'	'10006'
OPENVPN_2_LOCAL_HOST	'10.2.0.1'	'10.1.0.1'
OPENVPN_2_LOCAL_PORT	'10006'	'10005'
OPENVPN_2_FLOAT	'no'	'no'
OPENVPN_2_RESTART	'never'	'never'
OPENVPN_2_SECRET	'bridge.secret'	'bridge.secret'
OPENVPN_2_TYPE	'bridge'	'bridge'
OPENVPN_2_BRIDGE	'pema-br'	'pema-br'

TABLE 4.13.: Configuration d'OpenVPN avec 2 routeurs-fli4l leurs réseaux ont une connexion sans fil et utilise un Bridge

Naturellement En plus des paramètres d'OpenVPN, vous devez configurer le Bridge dans `advanced_networking` et aussi de configurer `base.txt`, de telle sorte que le Bridge soit utilisé en tant que périphérique réseau et non pas `eth0` pour le réseau interne. Ci-dessous nous avons adapté la configuration de `advanced_networking` et de `base`.

Option advanced_networking	Peter	Maria
OPT_BRIDGE_DEV	'yes'	'yes'
BRIDGE_DEV_BOOTDELAY	'no'	'no'
BRIDGE_DEV_N	'1'	'1'
BRIDGE_DEV_1_NAME	'pema-br'	'pema-br'
BRIDGE_DEV_1_DEVNAME	'br0'	'br0'
BRIDGE_DEV_1_DEV_N	'1'	'1'
BRIDGE_DEV_1_DEV_1_DEV	'eth0'	'eth0'

TABLE 4.14.: Configuration d'OpenVPN avec 2 routeurs-fli4l leurs réseaux ont une connexion sans fil et utilise un Bridge, configuration dans `advanced_networking`.

Option base	Peter	Maria
IP_NET_N	'1'	'1'
IP_NET_1	'192.168.193.254/24'	'192.168.193.1/24'
IP_NET_1_DEV	'br0'	'br0'

TABLE 4.15.: Configuration d'OpenVPN avec 2 routeurs-fli4l leurs réseaux ont une connexion sans fil et utilise un Bridge, configuration dans (`base.txt`).

Exemple - Configuration pour un accès Road-warrior (ou Guerre commerciale informatisée)

Pour cet exemple (Road warrior), un portable sous Windows XP est permis et un accès GPRS pour accéder au RÉSEAU LOCAL derrière le routeur-fli4l. Sur le portable est installé un OpenVPN pour Windows XP, le fichier *.ovpn correspondant doit être adapté. Malheureusement, les pilotes tun/tap sous Windows ne sont pas aussi souples que son homologue Unix. C'est pourquoi avec le Point-to-Point pour VPN les adresses-IP doivent se trouver dans le réseau 255.255.255.252 (ou /30). Si Road Warrior doit seulement accéder aux services du LAN derrière le routeur-fli4l, il ne sera pas nécessaire d'indiquer un itinéraire sur la page fli4l, comme cela il ne réagira pas. Avec Road warrior si nécessaire on peut utiliser une IP-adresse virtuels dans (OPENVPN_3_REMOTE_VPN_IP). Si Road warrior dispose d'une adresse-IP statique, on pourra également enregistrer un Hôte-route, par ex. si Road warrior a une adresse-IP 192.168.33.33 statique, vous pouvez insérer dans le fichier de configuration openvpn.txt de fli4l :

```
OPENVPN_3_ROUTE_N='1'
OPENVPN_3_ROUTE_1='192.168.33.33/32'
```

Au sujet de la configuration de filtrage de paquets ci-dessous, il vous permet une communication complète dans les deux sens. Road warrior ne peut pas interroger directement le routeur-fli4l. Mais si c'est nécessaire, il est possible d'utilise le serveur DNS du routeur-fli4l.

```
OPENVPN_3_PF_FORWARD_N='1'
OPENVPN_3_PF_FORWARD_1='ACCEPT'
```

Si Road warrior est autorisé pour accéder au serveur DNS interne du routeur-fli4l, il faut ajouter dans la configuration de fli4l les paramètres suivant :

```
OPENVPN_3_PF_INPUT_N='1'
OPENVPN_3_PF_INPUT_1='if:VPNDEV:any tmpl:dns ACCEPT'
```

Option OpenVPN routeur-fli4l	roadwarrior
OPENVPN_3_NAME='roadwarrior'	remote peter.eisfair.net
OPENVPN_3_LOCAL_PORT='10011'	rport 10011
OPENVPN_3_SECRET='roadwarrior.secret'	secret roadwarrior.secret
OPENVPN_3_TYPE='tunnel'	dev tun
OPENVPN_3_REMOTE_VPN_IP='192.168.200.238'	
OPENVPN_3_LOCAL_VPN_IP='192.168.200.237'	ifconfig 192.168.200.238 192.168.200.237
OPENVPN_3_ROUTE_N='0'	
OPENVPN_3_PF_FORWARD_N='1'	
OPENVPN_3_PF_FORWARD_1='ACCEPT'	
	route 192.168.145.0 255.255.255.0
	comp-lzo
	persist-tun
	persist-key
	ping-timer-rem
	ping-restart 60
	proto udp
	tun-mtu 1500
	fragment 1300
	mssfix

TABLE 4.16.: Configuration d'OpenVPN pour un ordinateur Windows avec GPRS

Exemple - Liaison WLAN sécurisé

Dans cet exemple, on passe par un WLAN (ou sans fil) pour accéder à la liaison OpenVPN. On part du principe que le WLAN est installé sur le routeur-fli4l, une carte Ethernet pour le réseau local et une carte WLAN, dont le point d'accès est activé. L'objectif doit être, un Client WLAN sans liaison-VPN a seulement accès au VPN par le port du routeur-fli4l. Ce n'est qu'après la connexion réussite avec OpenVPN, que l'échange sans restriction avec le réseau local du poste distant peut être possible. Pour cela des modifications du serveur DNSMASQ DHCP doivent être réalisées. En outre, le paquetage `advanced_networking` est nécessaire à l'installation. Il faut aussi paramétrer `IP_NET_1` pour le LAN (réseau local) et `IP_NET_2` pour le WLAN (réseau sans fil) dans le fichier `base.txt`.

```
IP_NET_N='2'
IP_NET_1='192.168.3.254/24'
IP_NET_1_DEV='br0'
IP_NET_2='192.168.4.254/24'
IP_NET_2_DEV='eth2'
```

Le DHCP-Range doit être réglé selon vos besoins. Pour la variable `IP_NET_2` vous devez absolument ajouter les paramètres suivants :

```
DNSDHCP_RANGE_2_DNS_SERVER1='none'
DNSDHCP_RANGE_2_NTP_SERVER='none'
DNSDHCP_RANGE_2_GATEWAY='none'
```

Paramètre `advanced_networking.txt` :

```
OPT_BRIDGE_DEV='yes'
BRIDGE_DEV_BOOTDELAY='yes'
BRIDGE_DEV_N='1'
BRIDGE_DEV_1_NAME='br'
BRIDGE_DEV_1_DEVNAME='br0'
BRIDGE_DEV_1_DEV_N='1'
BRIDGE_DEV_1_DEV_1_DEV='eth0'
```

4.14.9. Liens sur le thème OpenVPN

Pour terminer, encore quelques liens qui traitent de la configuration OpenVPN :

```
http://openvpn.net
http://de.wikipedia.org/wiki/OpenVPN
http://openvpn.se/
http://arnowelzel.de/wiki/en/fli4l/openvpn
http://wiki.freifunk.net/OpenVPN
http://w3.linux-magazine.com/issue/24/Charly.pdf
http://w3.linux-magazine.com/issue/25/WirelessLAN\_Intro.pdf
http://w3.linux-magazine.com/issue/25/OpenVPN.pdf
```

4. Les paquetages

Option OpenVPN routeur	Client-WLAN
OPENVPN_4_NAME='wlan1'	
OPENVPN_4_LOCAL_HOST='192.168.4.254'	remote 192.168.4.254
OPENVPN_4_LOCAL_PORT='20001'	rport 20001
OPENVPN_4_SECRET='wlan1.secret'	secret wlan1.secret
OPENVPN_4_TYPE='bridge'	dev tap
OPENVPN_4_BRIDGE='br'	
OPENVPN_4_RESTART='never'	
OPENVPN_4_MUTE_REPLAY_WARNINGS='yes'	
	comp-lzo
	persist-tun
	persist-key
	ping-timer-rem
	ping-restart 60
	proto udp
	tun-mtu 1500
	fragment 1300
	mssfix

TABLE 4.17.: OpenVPN sécurisé dans un WLAN

4.15. PCMCIA - Supporte les cartes PC

4.15.1. Pilote PCMCIA

fi4l peut également travailler avec les cartes PCMCIA. Si vous activez la variable `OPT_PCMCIA='yes'` vous devez installer les pilotes correspondants. Quels pilotes de cartes concrets doit on utiliser, on les paramètre dans la variable `NET_DRV_x` (Page 31) du fichier `config/base.txt`.

PCMCIA_PCIC - PCMCIA Socket-Driver

Voici les pilotes pour les contrôleurs de bus PCMCIA, vous pouvez choisir entre : 'i82365' ou 'tcic' pour les PCMCIA Bridges, ainsi que 'yenta_socket' et 'i82092' pour les cardbus Bridges.

Installation par défaut : `PCMCIA_PCIC='i82365'`

PCMCIA_PCIC_OPTS - Options pour Socket-Driver PCMCIA

Installation par défaut : `PCMCIA_PCIC_OPTS=""`

Réglage possible : `poll_interval=n` n par 10 Millisecondes - La valeur logique 1000 durée logique pour un changement de carte pcmcia

`irq_list=x,y,z,...` Pour indiquer une liste interruptions (IRQ) à utiliser

PCMCIA_MISC_N PCMCIA_MISC_x Dans la première variable on indique le nombre de module-PCMCIA, dans la seconde les modules-PCMCIA :

`serial_cs` Pour modem et cartes Combo.

`parport_cs` Pour interface parallèle (imprimante).

4.16. PPP - Connecter un ordinateur via le port série

En mettant la variable `OPT_PPP='yes'` sur yes, il sera possible d'utiliser un PC via le port série pour se connecter au réseau local. Ceci peut être utile pour intégrer dans le réseau par ex.

4. Les paquetages

un ordinateur portable, qui n'a pas de carte réseau. Voici, une documentation sur l'interface série pour PC client

PPP_DEV On indiquer ici, le port série de fli4l. Les valeurs suivantes sont permises :

'com1' Port-COM1 (en minuscules!)
'com2' Port-COM2 (en minuscules!)

PPP_SPEED On indiquer ici la vitesse de transmission (bit/sec). 38400 est supporté par les anciennes interfaces. Il peut y avoir éventuellement des problèmes si l'on paramètre les taux trop élevés de 57600, voire 115200 bit/s.

Exemple : `PPP_SPEED='38400'`

PPP_IPADDR PPP_PEER On paramètre dans la variable `PPP_IPADDR` l'adresse IP du routeur fli4l pour la connexion au port-COM, par ex. '192.168.4.1' et dans la variable `PPP_PEER` l'adresse IP du PC client, par exemple '192.168.4.2'.

PPP_NETWORK PPP_NETMASK On paramètre dans la variable `PPP_NETWORK` le réseau utilisé, par ex. '198.168.4.0' et dans la variable `PPP_NETMASK` le masque de sous-réseau utilisé, par ex. '255.255.0.0'. Ces deux variables sont complémentaires au paquetage 'samba_lpd'.

Important: *Il faut faire attention aux points suivants :*

1. Les adresses IP ne doivent **pas** provenir de la plage d'adresses du réseau Ethernet LAN, mais il faut avoir pour utiliser la configuration Point-to-Point une plage d'adresses réseau séparé!
2. Le PC client peut également accueillir une connexion Internet, si le mini réseau PPP est masquer comme le LAN (ou réseau local). Vous devez en plus de la liste des réseaux masquer, l'élargir au moyen de la variable suivante `PF_POSTROUTING_%` (Page 55) (voir le paragraphe suivant).
3. En plus, vous devez ajouter le PC client dans la table d'hôte du serveur DNS sur le routeur fli4l.

La raison est la suivante :

Si vous souhaitez que le PC client utilise telnet ou ftp pour se connecter au démon du routeur fli4l celui-ci doit faire une recherche de DNS inversée, pour établir une connexion. Si le PC client ne figure pas dans la table d'hôte, fli4l établira une connexion Internet, pour rechercher le nom du client sur Internet. C'est pour cela qu'il faut absolument écrire le PC client dans la table d'hôte du routeur fli4l.

Exemple de configuration avec une liaison PPP sur le port série :

```
PPP_DEV='com1'
PPP_SPEED='38400'
PPP_IPADDR='192.168.4.1'
PPP_PEER='192.168.4.2'
PPP_NETWORK='192.168.4.0'
PPP_NETMASK='255.255.255.0'
```

Et en plus dans le fichier config/base.txt :

```
PF_POSTROUTING_N='2'
PF_POSTROUTING_1='192.168.6.0/24 MASQUERADE'
PF_POSTROUTING_2='192.168.4.0/24 MASQUERADE'
```

Ci-dessus le premier réseau concerne le LAN-Ethernet, le second concerne le réseau PPP. Enfin il faut toujours configurer le fichier DNS, par exemple :

```
HOST_5='192.168.4.2 serial-pc'
```

Ne pas oublier d'incrémenter la variable `HOST_N` (Page 90) !

Si le PC client est un ordinateur Windows, vous devez configurer la carte d'accès distant pour une connexion PPP, pour pouvoir accéder au routeur fli4l.

Lorsque vous utilisez un ordinateur sous Linux, le mieux est de créer un script shell qui sera installé sur le PC client (par exemple `/usr/local/bin/ppp-on`) :

```
#!/bin/sh
dev='/dev/ttyS0'           # COM1, für COM2: ttyS1
speed='38400'              # Speed
options='defaultroute crtscts' # Options
myip='192.168.4.2'         # IP-Adresse Notebook
fli4lip='192.168.4.1'      # IP-Adresse fli4l-Router
pppd $dev $speed $options $myip:$fli4lip &
```

Si vous avez des problèmes avec le démon `pppd`

L'ordinateurs fli4l doit également être enregistré dans le serveur de DNS sur le PC client, si on souhaite se connecter à Internet. Il doit être enregistré dans le fichier `/etc/resolv.conf` du PC client, entrer les deux lignes suivantes : le Nom de domaine et adresse IP Ethernet du routeur fli4l.

Exemple :

```
search domain.de
nameserver 192.168.1.4
```

Les valeurs correspondantes aux "Domain.de" et "192.168.1.4" sont à remplacer par vos paramètres. Important : L'adresse IP doit être celle de la carte Ethernet du routeur fli4l ! La liaison série se fait avec un [cable-Nullmodem](#) (Page 344). Des informations à ce sujet sont dans l'annexe de la documentation Base.

Walter Oliver a rédigé un Howto (ou un guide), pour configurer le PC client Windows avec le protocole PPP qui peut être lu ici :

<http://www.fli4l.de/hilfe/howtos/basteleien/opt-ppp-howto/>

4.17. PROXY - Différent Serveur proxy

4.17.1. OPT_PRIVOX - Filtrage de la publicité avec un proxy HTTP

Privoxy "Privacy Enhancing Proxy" (= "filtrage avancé, pour la protection de la vie privée") voir le site Web officiel de Privoxy (<http://www.privoxy.org/>). Privoxy filtre le contenu des pages web sur votre navigateur, en remplaçant par des images vides les bannières publicitaires et les Popups, Il gère les cookies dans une mémoire cache (petit paquet de données avec lesquels un site web peut reconnaître certain surfer) et empêche l'affichage de ce que l'on appelle bugs-Web (ce sont de grandes images 1x1 pixels, qui sont utilisées, pour espionner le comportement des utilisateurs sur le Net).

Pendant que Privoxy fonctionne, vous pouvez tout simplement configurer et activer les paramètres par l'intermédiaire de l'interface Web. L'interface Web se trouve à l'adresse <http://config.privoxy.org/> ou en abrégée <http://p.p/>.

Une configuration ainsi activée ne survit pas à un redémarrage du routeur fli4l... (tobig)

Privoxy Internet Junkbusters à eu une évolution conséquente à partir de la version 2.1.0, voir le site Web (<http://www.junkbuster.com/>). L'innovation la plus importante, est que toutes les règles de filtrage sont centralisées dans un fichier `default.action`. Celui-ci se trouve dans le répertoire `fli4l/etc/privoxy`. Le grand avantage de cette méthode c'est que les nouvelles versions de ce fichier peuvent être télécharger séparément à cette adresse

<http://sourceforge.net/projects/ijbswa/files/>.

Ainsi, chaque utilisateur `fli4l` peut tenir ce fichier à jour, sans mettre à jour le routeur-`fli4l`. (Actuellement, la version 1.8 de ce fichier est dans ce paquetage)

PRIVOXY_MENU Avec cette variable, vous pouvez ajouter la section Privoxy au menu-httpd.

PRIVOXY_N Vous indiquez dans cette variable le nombre de Privoxy qui doit être enregistré pour chaque interface.

PRIVOXY_x_LISTEN Vous indiquez dans cette variable, l'adresse-IP ou le nom symbolique, y compris le numéro de Port de l'interface, sur lequel le Privoxy doit écouter les connexions des clients. C'est une bonne idée d'indiquer ici, seulement les adresses des interfaces que l'on fait confiance, car tous les ordinateurs auront un accès complet à travers le Privoxy (avec bien sur le navigateur configuré et activé). En règle générale il est judicieux d'indiquer, la valeur par défaut qui est `IP_NET_1_IPADDR:8118`

Avec l'adresse indiquée ici, le Privoxy écoute et offre ses services. Le port par défaut est 8118. Vous devez utiliser cette information pour configurer le proxy dans votre navigateur. Pour plus de détail sur la configuration d'Internet Explorer et de Netscape Navigator, voir le site Web :

<http://www.privoxy.org/>

URL précise

Vous devez enregistrer dans chaque navigateur, en tant que proxy l'ordinateur-`fli4l`, vous allez donc prendre le nom de la variable `HOSTNAME='fli4l'` ou l'adresse-IP (par ex. 192.168.6.1) de la variable `HOST_x_IP='192.168.6.1'` qui est dans le fichier config de `fli4l`. Avec le Port par défaut, on a ici tous les paramètres nécessaires, pour configurer votre navigateur Web, pour l'utilisation du Privoxy.

PRIVOXY_x_ALLOW_N Vous indiquez dans cette variable le nombre d'adresse réseau à installer.

PRIVOXY_x_ALLOW_x Vous indiquez dans cette variable l'adresse réseau ou l'adresse-IP pour le quelle le filtrage de paquets doit être ouvert. Normalement il est logique d'indiquer ici le paramètre `IP_NET_1`.

PRIVOXY_x_ACTIONDIR Avec cette variable vous indiquez l'emplacement, ou vous pouvez paramétrer l'ensembles des règles Privoxy (pour les fichiers `default.action` et `user.action`) sur le routeur. Le chemin d'accès spécifié est évaluée par rapport au répertoire racine. Cette variable peut être utilisé pour deux choses différentes :

Le déplacement dans la mémoire permanente de l'ensemble des règles Si vous spécifiez un répertoire dans un emplacement autre que le disque-RAM, au démarrage de `fli4l` l'ensemble des règles défini par défaut seront copiés et utilisé à partir de cet emplacement. Les modifications apportées à ces ensembles de règles survivront à un redémarrage du routeur. On doit aussi tenir compte du fait qu'après une mise à jour du paquetage Privoxy, ces règles seront toujours utilisées donc l'ensemble des règles du paquetage de mise à jour sera simplement ignoré.

l'utilisation de vos propres ensembles de règles L'utilisateur `fl4l` permet d'écrire des règles spécifiques à la place des règles standard. Vous devez dans cette variable indiquer votre propre sous-répertoire qui sera dans le répertoire *config* (par exemple *etc/mon_privoxy*, par contre vous ne devez pas indiquer *etc/privoxy*) ensuite vous placez dans ce sous-répertoire vos propres règles.

Le paramétrage de cette variable est optionnel.

PRIVOXY_x_HTTP_PROXY Si vous voulez utiliser en plus du Privoxy un autre Proxy HTTP, c'est-à-dire utiliser également des pages Web en cache, vous pouvez paramétrer cette variable. Le Privoxy utilise alors ce proxy. Avec cette variable vous avez l'avantage utilisé plusieurs Proxys. Le paramètre peut ressembler à cela :

```
PRIVOXY_1_HTTP_PROXY='mon.provider.de:8000'
```

Ce paramètre est optionnel.

PRIVOXY_x SOCKS_PROXY Si vous voulez utiliser en plus du Privoxy un autre Proxy SOCKS. Pour augmenter la surveillance privée de la transmission de données du Privoxy, par exemple, envoyé les données par le réseau Tor, vous pouvez paramétrer cette variable. Pour plus de détails sur Tor, reportez-vous à la [Documentation Tor](#) (Page 191). Le paramètre pour utiliser Tor peut ressembler à cela :

```
PRIVOXY_x SOCKS_PROXY='127.0.0.1:9050'
```

Ce paramètre est optionnel.

PRIVOXY_x_TOGGLE Avec cette variable vous pouvez arrêter le proxy par l'interface Web. Si le Privoxy est mis hors circuit, il réagira simplement comme Proxy-Forwarding et ne modifiera plus le contenu des pages Web transférées. Vous devez considérer, que ce réglage vaut pour TOUS les utilisateurs du Proxy, c.-à-d. que si un utilisateur arrête Privoxy, le Privoxy sera coupé pour tous les autres utilisateurs Web qui transfert par le Proxy.

PRIVOXY_x_CONFIG Avec cette variable, les utilisateurs ont la possibilité de configurer le proxy par l'interface web. Pour plus de détails, je vous demande de consulter la documentation Privoxy qui est ici.

PRIVOXY_x_LOGDIR Dans cette variable vous pouvez indiquer le répertoire du fichier log (ou journal) pour le privoxy. Cela peut être utile, par ex. lorsque l'utilisateur veut enregistrer les accès des sites Web. Si rien n'est spécifié (par défaut), les principaux messages seront enregistrés sur la console, de plus la variable `PRIVOXY_LOGLEVEL` sera ignorée.

Vous pouvez aussi indiquer 'auto', le chemin du fichier log sera alors déplacé dans le répertoire système, pour avoir des données persistantes. S'il vous plaît, assurez-vous que la variable `FLI4L_UUID` soit dans ce cas configuré correctement. Comme on peut si attendre une grandes quantités de données sera enregistrées et le fichier log dans le /boot ou dans le Disque-RAM sera rempli rapidement.

PRIVOXY_x_LOGLEVEL On indique dans cette variable les valeurs, pour que Privoxy puisse enregistrer les événements dans le fichier log. Il est possible d'ajouter plusieurs valeurs à la suite, vous devez les séparer par un espace. Les valeurs suivantes peuvent être ajoutées.

Valeur	ce qui sera enregistré ?
1	Chaque Requête (GET/POST/CONNECT).
2	Le statut de chaque connexion
4	Le statut-I/O
8	Header-Parsing
16	Toutes les données
32	Debug force-feature
64	Debug regular expression filters
128	Debug redirects
256	Debug GIF animation
512	Common Log Format (Analyse fichier-log)
1024	Debug kill pop-ups
2048	CGI (server web) user interface
4096	Startup banner and warnings
8192	Non-fatal errors

Pour produire un fichier log (ou journal) avec Common Log Format, vous devez indiquer SEULEMENT la valeur 512, si vous indiquez d'autres valeurs le fichier log sera "pollué" par d'autres enregistrements et vous aurez des problèmes pour l'analyser.

Privoxy offre de très nombreuses options de configurations. Cependant pour des raisons compréhensibles nous ne pouvons pas développer toutes ces options dans le fichier de configuration de fli4l. Beaucoup de ces options peuvent être paramétrées sur l'interface Web de Privoxy. Vous trouverez des infos plus précises pour la configuration de ces fichiers sur la page d'accueil de Privoxy. Les fichiers de configuration de Privoxy se trouvent dans le répertoire <fli4l-Version>/opt/etc/privoxy/. Ce sont des fichiers originaux du Paquetage-Privoxy, toutefois, pour gagner de la place, tous les commentaires ont été supprimés.

4.17.2. OPT_TOR - Système de communication anonyme pour Internet

Tor est l'outil d'un grand nombre d'organismes et de simples citoyens, qui veulent améliorer leur protection et leur sécurité sur Internet. L'utilisation de Tor vous aide à être anonymes lors de la navigation et de la publication sur le Web, messagerie instantanée, IRC, SSH et autres applications basées sur TCP. En outre, Tor fournit une plate-forme sur laquelle les développeurs de logiciels peuvent créer de nouvelles applications pour plus d'anonymat, sur la sécurité et la protection de la vie privée.

<https://www.torproject.org/index.html.fr>

TOR_LISTEN_N

TOR_LISTEN_x Dans la première variable, vous indiquez le nombre d'adresse réseau, dans la deuxième variable vous indiquez l'adresse-IP ou le nom symbolique, y compris le numéro de Port de l'interface, sur lequel Tor doit écouter les connexions des Clients. C'est une bonne idée, d'indiquer ici seulement les adresses des interfaces que l'on fait confiance, car tous les ordinateurs auront un accès complet à travers Tor (avec bien sur le navigateur configuré et activé). En règle générale il est judicieux d'indiquer, la valeur par défaut qui est IP_NET_1_IPADDR:9050

Avec l'adresse indiquée ici, Tor écoute et offre ses services. Le port par défaut est 9050. Vous devez utiliser cette information pour configurer le proxy dans votre navigateur.

Vous devez indiquer dans chaque navigateur en tant que proxy l'ordinateur-fli4l, vous allez donc prendre le nom de la variable `HOSTNAME='fli4l'` ou l'adresse-IP (par ex. 192.168.6.1) de la variable `HOST_x_IP='192.168.6.1'` qui est dans le fichier config de fli4l. Avec le Port par défaut, on a ici tous les paramètres nécessaires, pour configurer votre navigateur Web, pour l'utilisation de Tor.

TOR_ALLOW_N Vous indiquez dans cette variable le nombre d'adresse réseau à installer.

TOR_ALLOW_x Vous indiquez dans cette variable l'adresse réseau ou l'adresse-IP pour le quelle le filtrage de paquets doit être ouvert. Normalement il est logique d'indiquer ici le paramètre `IP_NET_1`.

TOR_CONTROL_PORT Vous indiquez dans cette variable, le port TCP que Tor doit utiliser, pour le contrôle d'accès via le protocole Tor. Cette variable est optionnelle, si rien n'ai indiqué cette fonction sera désactivée.

TOR_CONTROL_PASSWORD Vous spécifier dans cette variable, un mot de passe pour le contrôle d'accès.

TOR_DATA_DIR Cette variable est optionnelle. Si rien n'est indiqué, le dossier par défaut `/etc/tor` est utilisé.

TOR_HTTP_PROXY Si vous voulez utiliser en plus de Tor un autre Proxy http, Tor pourra alors utiliser ce proxy. Avec cette variable vous avez l'avantage utilisé plusieurs Proxys. Le paramètre peut ressembler à cela :

```
TOR_HTTP_PROXY='mein.provider.de:8000'
```

Ce paramètre est optionnel.

TOR_HTTP_PROXY_AUTH Une authentification peut-être nécessaire, vous devez la spécifier dans cette variable. Ainsi le mandataire sera enregistré sous la forme Nom d'utilisateur :Mot de passe.

TOR_HTTPS_PROXY Vous pouvez enregistrer dans cette variable, un Proxy-HTTPS. Voir [TOR_HTTP_PROXY](#).

TOR_HTTPS_PROXY_AUTH Voir pour ce sujet [TOR_HTTP_PROXY_AUTH](#).

TOR_LOGLEVEL On indique dans cette variable les valeurs pour que Tor puisse enregistrer les événements dans le fichier log. Les valeurs suivantes sont possibles : debug, info, notice, warn ou err. Les valeurs debug et info ne devraient pas si possible être utilisées, pour des raisons de sécurité.

TOR_LOGFILE Si vous voulez utiliser un autre système que syslog pour enregistrer les événements de Tor, vous devez l'indiquer dans cette variable.

Vous pouvez aussi indiquer 'auto', le chemin du fichier log sera alors déplacé dans le répertoire système, pour avoir des données persistantes. S'il vous plaît, assurez-vous que la variable `FLI4L_UUID` soit dans ce cas configuré correctement. Comme on peut si attendre une grandes quantités de données sera enregistrées et le fichier log dans le /boot ou dans le Disque-RAM sera rempli rapidement.

4.17.3. OPT_SS5 - Proxy Socks 4/5

Il est nécessaire d'installer le Proxy-Socks pour certains programmes, nous mettons à disposition ici le protocole SS5. Voir le site Web.

<http://ss5.sourceforge.net/>

SS5_LISTEN_N

SS5_LISTEN_x Dans la première variable vous indiquez le nombre d'adresse réseau, dans la deuxième variable vous indiquez l'adresse-IP ou le nom symbolique, y compris le numéro de Port de l'interface, sur lequel SS5 doit écouter les connexions des Clients. C'est une bonne idée, d'indiquer ici seulement les adresses des interfaces que l'on fait confiance, car tous les ordinateurs auront un accès complet à travers SS5 (avec bien sûr le navigateur configuré et activé). En règle générale il est judicieux d'indiquer, la valeur par défaut qui est `IP_NET_1_IPADDR:8050`

Avec l'adresse indiquée ici, SS5 écoute et offre ses services. Le port par défaut est 8050. Vous devez utiliser cette information pour configurer le proxy dans votre navigateurs.

Vous devez indiquer dans chaque navigateur en tant que proxy l'ordinateur-fli4l, vous allez donc prendre le nom de la variable `HOSTNAME='fli4l'` ou l'adresse-IP (par ex. 192.168.6.1) de la variable `HOST_x_IP='192.168.6.1'` qui est dans le fichier config de fli4l. Avec le Port par défaut, on a ici tous les paramètres nécessaires, pour configurer votre navigateur Web, pour l'utilisation de SS5.

SS5_ALLOW_N Vous indiquez dans cette variable le nombre d'adresse réseau à installer.

SS5_ALLOW_x Vous indiquez dans cette variable l'adresse réseau ou l'adresse-IP pour le quelle le filtrage de paquets doit être ouvert. Normalement il est logique d'indiquer ici le paramètre `IP_NET_1`.

4.17.4. OPT_TRANSPROXY (Expérimental) - Proxy HTTP transparent

Transproxy est un proxy "transparent", c'est une application qui permet, d'intercepter toutes les requêtes-HTTP qui passent par le routeur et de les transmettre à un Proxy-HTTP normal, par ex. au Privoxy. Pour parvenir à cette fonctionnalité, le filtre de paquets des requêtes HTTP qui doit aller sur Internet, passent par le transproxy celui-ci les traite et les transmet à un Proxy-HTTP. Iptables supporte cette fonction en utilisant le paramètre "REDIRECT" :

```
PF_PREROUTING_1='tpr: http IP_NET_1 REDIRECT:8081'
```

Cette règle transmet tous les paquets-HTTP du premier réseau défini (normalement c'est le LAN interne) au transproxy par le port 8081.

TRANSPROXY_LISTEN_N

TRANSPROXY_LISTEN_x Vous indiquez ici, les adresses IP ou les noms symboliques, ainsi que le numéro de port des interfaces, sur lesquels Transproxy doit écouter les connexions des clients. Si Toutes les interfaces spécifiées ici, utilise déjà un filtrage de paquets, Transproxy sera gêné par les paquets qui provient de ce filtrage. Avec la configuration par défaut `any:8081` Transproxy écoutera toutes les interfaces.

TRANSPROXY_TARGET_IP

TRANSPROXY_TARGET_PORT Grâce à cette option vous définissez le service pour lequel les requêtes HTTP doivent être redirigé. Cela peut être n'importe quel proxy standard HTTP (Squid, Privoxy, Apache, etc) et sur n'importe quel ordinateur (ou sur fli4l lui-même). Faire attention, que le Proxy ne se trouve pas dans le domaine du filtrage de paquet, dans lequel les requêtes http seront redirigées. Autrement un bouclage d'adresse apparaîtra.

TRANSPROXY_ALLOW_N

TRANSPROXY_ALLOW_x Liste des réseaux et/ou des adresses IP pour le filtrage de paquets ouvert. Cela devrait couvrir mêmes les réseaux, qui sont redirigés par le filtrage de paquets. Si aucun domaine n'est indiqué ici, vous devez indiquer les informations manuellement dans la configuration du filtrage de paquets.

4.17.5. OPT_SIPPROXY (Expérimental) - Proxy pour Session Initiation Protocol

Si vous souhaitez utiliser plusieurs applications SIP (Ekiga, x-lite ou du matériel téléphonique SIP) qui fonctionner derrière le routeur, il peut arriver que vous devez transmettre spécifiquement les ports réseau. Autrement, les connexions ne fonctionnent pas comme ils le devraient.

Pour éviter cela, on peut utiliser un proxy SIP spécial. Plusieurs de ces proxys sont en cours d'évaluation pour (fii4l V4.0.0). Si quelqu'un a des suggestions, il ne doit pas hésiter à contacter l'équipe!

4.17.6. OPT_IGMPPROXY - Proxy pour Internet Group Management Protocol

Depuis quelques années Telekom AG Allemand utilise le VDSL 25/50 (bande passante : 25/50 Mbit/s) pour envoyer des paquets de divertissement. Ainsi, il est possible de recevoir la télévision par Internet (IPTV).

La distribution de la télévision sur IP est effectuée en utilisant le multicast (ou la multidiffusion), à savoir, un émetteur source unique vers un groupe (fermé). Pour l'organisation de la diffusion de groupe le protocole réseau IGMP (Internet Group Management Protocol) est nécessaire. L'IGMP (http://fr.wikipedia.org/wiki/Internet_Group_Management_Protocol) offre la possibilité de gérer dynamiquement des groupes multicast. L'administration ne se trouve pas dans la station d'émission, mais dans le routeur sur laquelle les destinataires du groupe multicast sont connectés directement. L'IGMP fournit des fonctions par lesquelles une station notifie au routeur qu'il peut recevoir des paquets IP multicast pour un groupe multicast particulier.

Les routeurs Speedport (actuellement W700V/W701V/W722) fournissent un support IGMP. Au lieu d'utiliser des routeurs Speedport, vous pouvez utiliser fii4l avec le support IPTV, pour cela vous devez installer le proxy IGMP sur le routeur fii4l.

La documentation du paquetage OPT_IGMP décrit la configuration de fii4l avec une connexion VDSL et IPTV, vous devez avoir un décodeur (ou STB) X300T/X301T ou MR-303 derrière le routeur fii4l pour fonctionner. Dans cette documentation, on utilise une carte supplémentaire pour l'installation de l'IPTV via le réseau.

Condition préalable

Telekom VDSL Allemand est présenté comme un VLAN. Dans la phase de lancement (démarrage du réseau) un seul lien VLAN (ID7) a été utilisé, sur lequel tout le trafic circulait. Après des modifications (sur le réseau) deux liens VLAN (ID7, ID8) sont utilisés, le lien ID7 reste pour le trafic Internet et le nouveau lien ID8 est utilisé exclusivement pour le trafic du multicast IPTV. Actuellement les modifications du VDSL pour le réseau (deux liens VLAN ID7/ID8) sont en grande partie terminées.

Hardware (avec un Set-Top-Box (ou boîtier décodeur) et un modem-VDSL) :

4. Les paquetages

- Hardware pour fli4l : avec un VDSL 25/50 vous pouvez utiliser un processeur 486. Si vous avez des problèmes avec l'image/son le matériel utilisé est trop faible.
- Carte réseau haut de gamme (par exemple : 3Com, Intel PRO100). Le chipset Realtek est à mon avis un composant bas de gamme

Software :

- Paquetage : `advanced_networking`
- Paquetage : `dhcp_client` (pour le réseau et l'utilisation du lien ID8)

La configuration des fichiers de (`base.txt`, `dsl.txt`, `advanced_networking.txt`, `dhcp_client.txt`, `dns_dhcp.txt`) sont décrites dans ce manuel.

Configuration du hardware

Recommandation avec le routeur Speedport et une connexion au boîtier décodeur IPTV directement sur le routeur sans autres éléments dans le réseau, bien sûr, cela s'applique également à fli4l. Néanmoins, si un nœud est interposés (comme un concentrateur, commutateur, pont, passerelle, routeur) entre le boîtier IPTV et le routeur, il doit être compatible avec le multicast, pour éviter les interférences.

Le commutateurs (ou switch) est en général pas utilisé dans le réseau domestique, le réseau virtuel (VLAN) sert à soulager le trafic avec le lien (ID7) et le lien (ID8) pour le trafic multicast de l'IPTV.

Vous pouvez aussi, pour la configuration du hardware de fli4l utiliser des cartes réseaux séparés (une carte d'interface réseau = LAN ou carte Ethernet) et une carte pour la connexion du boîtier décodeur pour le trafic multicast, cela soulagera le reste de votre réseau domestique et écartera les problèmes par rapport à la configuration ci-dessus. Pour ceux qui préfèrent la méthode avec une 'simple' carte réseau, ils doivent savoir se qu'ils font (car elle n'est pas décrite ici).

Voici deux schémas un avec le routeur par défaut et un avec 3 cartes réseaux installées dans le routeur fli4l :

- Configuration par défaut :
 - La carte réseau `eth0` est configurée dans `base.txt` pour le LAN interne de la maison/bureau.
 - La carte réseau `eth1` est configurée dans `dsl.txt` pour l'interface DSL.

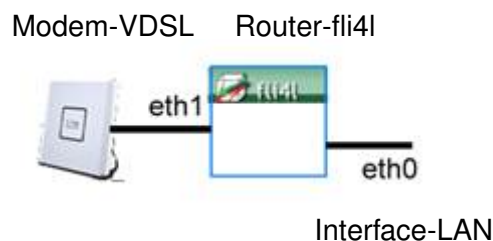


FIGURE 4.5.: fli4l avec la configuration par défaut

- Configuration avancée avec une carte réseau pour l'IPTV :
 - Après l'installation d'une carte réseau supplémentaire dans le routeur fli4l, vous devez configurer la carte supplémentaire `eth2` dans `base.txt`

Modem-VDSL Router-fli4l Interface pour le décodeur IPTV



Interface-LAN

FIGURE 4.6.: fli4l avec la configuration IPTV

Configuration du VLAN

Tout d'abord : l'OPT_IGMP ne dépend pas du VLAN. Le VLAN est plutôt utilisé par Deutsche Telekom pour le VDSL et doit être supporté par le routeur. Si le VLAN est nécessaire pour d'autres fournisseurs Internet (Arcor, Alice, etc ..), la configuration est actuelle au-delà de mes connaissances.

Pour que Internet fonctionne avec le VDSL 25/50 de T-Home, la carte réseau qui est connectée au modem VDSL doit être configuré comme une interface VLAN.

Une note pour ceux qui ont seulement le 'DSL normal', c'est à dire ADSL, ADSL2, ADSL2+ : le VLAN est nécessaire uniquement avec le VDSL, mais pas avec le 'DSL normal'. La configuration du VLAN ne doit pas être installé avec l'utilisation du 'DSL normal'.

Si vous utilisez deux lien VLAN (voir ci-dessus) le trafic se répartit comme ceci :

- VLAN ID7 : Trafic-Internet
- VLAN ID8 : Trafic-IPTV Multicast

Donc, le trafic Internet fonctionne indépendamment du trafic IPTV. La principale différence, il est nécessaire d'utiliser le VLAN ID7 pour les accès entrant PPPoE. Le VLAN ID8 est fournie par un serveur DHCP sans accès entrant. Dans cette architecture, il n'y a pas de redémarrage forcé après 24 heures.

Pour le VLAN la configuration suivante est requise (la carte réseau est indiquée à la section configuration du matériel) :

advanced_networking.txt

```
VLAN_DEV_N='2'
VLAN_DEV_1_DEV='eth1'      # interface of VDSL-Modem; example: eth1
                             # in our example 'eth1' connects to the VDSL modem
VLAN_DEV_1_VID='7'         # ID7 to support VLAN for internet
VLAN_DEV_2_DEV='eth1'      # interface of VDSL modem; example: eth1
VLAN_DEV_2_VID='8'         # ID8 to support VLAN for IPTV
```

La carte réseau virtual eth1.7 doit être paramétré dans le fichier de configuration DSL :

dsl.txt

4. Les paquetages

```
PPPOE_ETH='eth1.7'          # eth<number of the card connecting the vdsl modem>.7'
                             # i.e. 'eth1.7'
```

Pour la carte réseau virtual eth1.8 vous aurez besoin d'un client_dhcp, car le VLAN ID8 est configuré par un serveur DHCP sans accès entrant.

dhcp_client

```
OPT_DHCP_CLIENT='yes'
DHCP_CLIENT_TYPE='dhcpcd'
DHCP_CLIENT_INTERFACES='IP_NET_3_DEV' # listen on interface eth1.8
DHCP_CLIENT_USEPEERDNS='no'
DHCP_CLIENT_HOSTNAME=''
```

Depuis la v3.3 de fli4l, on ne peut plus définir l'interface avec cette valeur **eth1.8**, mais vous devez utiliser **IP_NET_x_DEV** pour définir l'interface depuis le fichier **base.txt**; Ici **IP_NET_3_DEV**.

Facultatif :

Si la carte réseau que vous utilisez a des problèmes avec la taille du MTU, vous pouvez là régler dans la variable **DEV_MTU**. Pendant les testes, la carte Intel Pro/100 (e100) et la 3-Com ont montrées aucun problème, d'autres utilisateurs ont signalés des problème avec la carte 3Com '3c59x', ils ont modifiés la valeur MTU à 1496.

```
DEV_MTU_1=''                # Adjust MTU size of NIC on VDSL-Modem
                             # Example: DEV_MTU_1='eth1 1496'
```

Les fichiers de configuration **base.txt** et **dns_dhcp.txt** doivent être modifiées, comme décrit dans le chapitre suivant.

Configuration de la carte réseau supplémentaire pour l'IPTV

Vous devez configurer **base.txt** et **dns_dhcp.txt** pour paramétrer la deuxième carte réseau et le VLAN.

Paramétrage de la deuxième carte réseau pour l'IPTV :

```
NET_DRV_N='2'
NET_DRV_1='via-rhine'       # 1. NIC interface for LAN
NET_DRV_2='3c59x'          # 2. NIC - here 3Com for IPTV SetTopBox
```

Maintenant nous devons spécifier la plage d'adressage pour la deuxième carte réseau. Nous allons utiliser pour le réseau local 192.168.2.0/24 et 192.168.3.0/24 pour la deuxième carte réseau. Nous avons besoin également de paramétrer les cartes virtuelle pour eth1.7 et eth1.8 :

4. Les paquetages

```
IP_NET_N='4'
IP_NET_1='192.168.2.1/24'          # home/office LAN
IP_NET_1_DEV='eth0'
IP_NET_2='192.168.3.1/24'          # iptv LAN
IP_NET_2_DEV='eth2'
IP_NET_3='dhcp'                    # dhcp client - IP via dhclient
IP_NET_3_DEV='eth1.8'
IP_NET_3_MAC='00:40:63:da:cf:32'   # new MAC (not the MAC of eth1)
IP_NET_4='dhcp'                    # eth1.7 connecting to the modem
IP_NET_4_DEV='eth1.7'
IP_NET_4_MAC='00:40:63:da:cf:33'   # new MAC (not the MAC of eth1)
```

Il est important de changer les adresses MAC pour eth1.7 et eth1.8, elles ne doivent pas coïncider avec eth1, sinon - selon le réseau VDSL - il peut éventuellement se produire des perturbations après la déconnexion forcée.

Pour que la nouvelle carte réseau puisse accéder à Internet, bien sûr, tout comme pour la première carte réseau. Ces paramètres supplémentaires sont nécessaires :

```
PF_INPUT_1='IP_NET_1 ACCEPT'
PF_INPUT_2='IP_NET_2 ACCEPT'
PF_INPUT_3='any 224.0.0.0/4 ACCEPT'
[...]
PF_FORWARD_3='any 224.0.0.0/4 ACCEPT'
PF_FORWARD_5='IP_NET_1 ACCEPT'
PF_FORWARD_6='IP_NET_2 ACCEPT'
[...]
PF_POSTROUTING_1='IP_NET_1 MASQUERADE'
PF_POSTROUTING_2='IP_NET_2 MASQUERADE'
```

Pour que cela fonctionne vous devez paramétrez l'adressage DHCP dynamique sur la carte réseau IPTV, pour accéder à la Set-Top Box (ou décodeur) et lui donner un nom. Les paramètres suivants dans le dns_dhcp.txt sont nécessaires :

```
HOST_10_NAME='igmp'
HOST_10_IP4='192.168.3.1'
HOST_11_NAME='iptv'
HOST_11_IP4='192.168.3.4'
HOST_11_MAC='00:D0:E0:93:49:34'    # MAC Adr T-Home X300T
[...]
DHCP_RANGE_2_NET='IP_NET_2'
DNSDHCP_RANGE_2_START='192.168.3.10'
DNSDHCP_RANGE_2_END='192.168.3.20'
DNSDHCP_RANGE_2_DNS_SERVER1=''
DNSDHCP_RANGE_2_DNS_SERVER2=''
DNSDHCP_RANGE_2_NTP_SERVER=''
DNSDHCP_RANGE_2_GATEWAY=''
```

Après avoir configuré la nouvelle carte réseau, il est judicieux de tester l'accès Internet avec un PC connecté au routeur. Si vous arrivez à vous connecter, la seconde carte réseau sera configuré correctement.

Fonctions de l'IGMP

Lors du démarrage du routeur fli4l les paramètres du fichier de configuration proxy.txt sont écrit dans le fichier /etc/igmpproxy.conf, ils sont ensuite lu lors du démarrage du proxy IGMP.

Contrairement aux versions antérieures le paquetage opt_igmp est lancé au démarrage, il est exécute aussi longtemps que la connexion Internet physique est disponible. Le proxy IGMP n'est pas affectée par la déconnexion forcé après 24 heures ou par la connexion/déconnexion manuel de l'accès Internet.

Configuration de l'IGMP

OPT_IGMPPROXY Si vous indiquez 'yes', vous activez le proxy IGMP. avec 'no' vous désactivez l'ensemble du paquetage.

IGMPPROXY_DEBUG Si vous indiquez 'yes' les messages du proxy IGMP sont envoyés à syslog.

IGMPPROXY_DEBUG2 Si vous indiquez 'yes' vous pouvez augmenter le niveau des messages du proxy IGMP.

IGMPPROXY_QUICKLEAVE_ON Avec Quickleave vous pouvez abaisser la charge de la liaison montante. Si vous avez indiqué 'yes', le multicast sera arrêté plus rapidement après un changement de canal et la charge de la liaison descendante sera réduite par le proxy IGMP, il se comporte comme un récepteur.

Si vous utilisez deux décodeurs et qu'il sont sur le même canal, il peut arriver (avec l'activation de Quickleave) que le programme soit interrompu sur l'un des décodeurs, vous devez alors changer le programme. Si vous utilisez un seule décodeur Quickleave peut être activé en toute sécurité.

```
IGMPPROXY_QUICKLEAVE_ON='yes'      # activate Quickleave mode
                                     # yes or no; Default: yes
```

IGMPPROXY_UPLOAD_DEV Pour le fonctionnement de l'IPTV avec le proxy IGMP vous avez besoin d'une interface avec une liaison montante et descendante. L'interface de la liaison montante est l'interface de la carte réseau qui sera connecté au modem VDSL. Normalement cela doit toujours être la même.

Le transfert de l'IPTV doit se faire sur le lien ID8 avec eth1.8, au lieu de ppp0. Cela doit être paramétré dans le fichier de configuration.

```
IGMPPROXY_UPLOAD_DEV='eth1.8'      # Upstream interface; Default: ppp0
                                     # eth1.8 for T-Home/VDSL with id7/id8
```

IGMPPROXY_DOWNLOAD_DEV L'interface de la liaison descendante (carte réseau pour le décodeur de l'IPTV) dépend de la configuration matériel. Dans fli4l la deuxième carte réseau eth2 est l'interface pour le décodeur.

```
IGMPPROXY_DOWNLOAD_DEV='eth2'      # Downstream Interface
```

IGMPPROXY_ALT_N Dans cette variable vous indiquez le nombre de plages d'adresse IP pour le flux multicast.

IGMPPROXY_ALT_NET_x Dans la variable IGMPPROXY_ALT_NET vous indiquez les adresses IP pour le trafic multicast provenant de extérieur pour le réseau local, ainsi que l'adresse IP local du décodeur.

4. Les paquetages

```
IGMPPROXY_ALT_N='3'                # Number of Multicast sources
IGMPPROXY_ALT_NET_1='239.35.0.0/16' # IPTV streams - always needed
IGMPPROXY_ALT_NET_2='217.0.119.0/24' # needed for T-Home
IGMPPROXY_ALT_NET_3='193.158.34.0/23' # needed for T-Home
                                     # before May 2013 '193.158.35.0/24'
# IGMPPROXY_ALT_NET_4='192.168.3.0/24' # Address range IPTV SetTop-Box/not
                                     # needed anymore
```

IGMPPROXY_WLIST_N Dans cette variable vous indiquez le nombre de listes blanches pour le rapport IGMP.

IGMPPROXY_WHLIST_NET_x :

Si vous utilisez IGMPv3 toutes les adresses sont regroupées dans un rapport, (<http://grinch.itg-em.de/entertain/artikel/zielnetzarchitektur-und-igmpproxy/>). Ces adresses seront ensuite complètement ignorées. Cela conduira à un arrêt complet de tout le trafic multicast par l'émetteur IGMP, en supposant qu'ils ne sont plus nécessaires. Pour éviter cela, la configuration de listes blanches est utilisée. Seuls les groupes multicast de cette liste seront épargnés par le WAN.

```
IGMPPROXY_WLIST_N='1'                # Number of Multicast sources
IGMPPROXY_WHLIST_NET_1='239.35.0.0/16' # IPTV streams - always needed
                                     # see above
```

Modification des autres fichiers de configuration

Avec la révision 32955 il n'est pas nécessaire de paramétrer les règles du pare-feu pour le proxy IGMP et pour le fux multicast si les règles standard (PF_INPUT_ACCEPT_DEF='yes' et PF_FORWARD_ACCEPT_DEF='yes') sont activés dans le fichier base.txt, le script de démarrage ajoutera automatiquement ces règles si la variable OPT_IGMPPROXY='yes' est activée.

Il y a deux règles qui sont ajouté dans la chaîne INPUT pour permettre aux messages entrants d'atteindre le proxy IGMP :

```
Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target  prot opt in  out  source      destination
[...]
0      0 ACCEPT  all  --  *   *    0.0.0.0/0   224.0.0.1    \
/* automatically added for IGMP Proxy */

0      0 ACCEPT  all  --  *   *    0.0.0.0/0   224.0.0.22   \
/* automatically added for IGMP Proxy */
[...]
```

Vous avez aussi une règle dans la chaîne FORWARD qui permet de transmettre le flux multicast entrant vers le récepteur de média :

```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
pkts bytes target  prot opt in  out  source      destination
[...]
0      0 ACCEPT  all  --  *   *    0.0.0.0/0   239.35.0.0/16 \
/* automatically added for IPTV streams */
[...]
```


Si l'une des règles par défaut n'est pas activée, vous devez au moins paramétrer et insérer manuellement les règles suivantes.

```
PF_INPUT_x='any 224.0.0.1/32 ACCEPT'  
PF_INPUT_x='any 224.0.0.22/32 ACCEPT'  
[...]  
PF_FORWARD_x='any 239.35.0.0/16 ACCEPT'
```

Remarque : contrairement aux versions précédentes de la documentation, les règles réellement nécessaires ont été écrites pour un réseau limité. Si l'IPTV ne fonctionne pas, n'hésitez pas à fournir des informations supplémentaires concernant le réseau que vous utilisez.

Important ! Depuis la fin du mois de mai 2013, Telekom a introduit de nouvelles routes (sans classe prédéfini) pour améliorer le service (<http://www.onlinekosten.de/forum/showthread.php?t=116415&page=38>). Cela est une bonne chose, car plus de 256 émetteurs ou adresses sont utilisables. Maintenant le serveur DHCP transfère les routes, elles ne sont plus incluses dans le sous-réseau comme précédemment. Tant que Telekom ne change pas son sous-réseau du serveur-IPTV (193.158.34.0/23) la route statique peut être définie vers l'interface vlan8, vous devez faire attention au changement de celle-ci, sinon le multicast ne fonctionnera plus.

Solution : Dans le fichier base.txt, vous devez spécifier une route supplémentaire.

```
IP_ROUTE_N='1'  
IP_ROUTE_1='193.158.34.0/23 eth1.8'
```

4.17.7. OPT_STUNNEL - Tunnel avec une connexion SSL/TLS

Le programme "stunnel" permet d'encapsuler les connexions non cryptées dans un tunnel SSL/TLS crypté. Ce protocole permet d'échanger du texte clair en toute sécurisée. Grâce aux possibilités du protocole SSL/TLS, vous pouvez paramétrer les différents modes Client/Serveur.

Configuration

OPT_STUNNEL Cette variable vous permet d'utiliser un tunnel SSL/TLS.

Configuration par défaut : OPT_STUNNEL='no'

Exemple : OPT_STUNNEL='yes'

STUNNEL_DEBUG Avec cette variable vous enregistrez paramètres de fonctionnement du "stunnel" les valeurs disponibles sont "yes" (tout est enregistré), "no" (les avertissements et les erreurs sont enregistrés), vous pouvez aussi indiquer une valeur comprise entre zéro et sept, cela spécifie la gravité maximale d'enregistrement des messages, si vous indiquez zéro vous enregistrez tous les messages d'urgences et sept tous les messages de débogage. Le paramètre "yes" correspond à sept pour la gravité maximale et "no" correspond à quatre pour la gravité maximale.

Configuration par défaut : STUNNEL_DEBUG='no'

Exemple 1 : STUNNEL_DEBUG='yes'

Exemple 2 : STUNNEL_DEBUG='5'

STUNNEL_N Avec cette variable vous configurez le nombre de tunnel. Chaque instance de tunnel "écoute" sur un port du réseau "A" et utilise une connexion entrante sur un autre port du réseau "B" (qui peut également être situé sur une autre machine) avant tout trafic de "A" vers "B". Les données qui partent de "A" via le SSL/TLS sont cryptées, ils passent par "stunnel", les données sont décodées par "B" ou vice versa et ensuite il sont transmis, on paramètre cette fonction avec la variable [STUNNEL_x_CLIENT](#) (Page 202).

Configuration par défaut : `STUNNEL_N='0'`

Exemple : `STUNNEL_N='2'`

STUNNEL_x_NAME Dans cette variable vous indiquez le nom du tunnel. Ce nom doit être unique pour chaque tunnel configuré.

Exemple : `STUNNEL_1_NAME='imond'`

STUNNEL_x_CLIENT Dans cette variable vous pouvez régler le tunnel qui communiquera en SSL/TLS crypté. Il y a deux options :

- *Mode client* : il reçoit les données non cryptées qui proviennent du tunnel distant et enverra les données cryptées à l'autre extrémité du tunnel. Pour cela vous devez indiquer `STUNNEL_x_CLIENT='yes'`.
- *Mode serveur* : il reçoit les données cryptées via le SSL/TLS, il procède au décodage et renvoie le résultat à l'autre extrémité du tunnel. Pour cela vous devez indiquer `STUNNEL_x_CLIENT='no'`.

Le tunnel en mode client est particulièrement adapté pour des connexions qui vont vers l'extérieur, par ex. pour un accès Internet (non protégé), les données seront cryptées avant de quitter le réseau local. Le site incontournable distant doit bien sûr avoir également un serveur avec le service SSL/TLS pour recevoir les données chiffrées. Exemple d'un client e-Mail sur le LAN, le protocole POP3 "parle" en utilisant des données non cryptées, vous pouvez utiliser un compte POP3 avec le service SSL sur Internet.¹⁴

Le tunnel en mode serveur est à l'inverse adapté pour des connexions qui "proviennent de l'extérieur", par ex. des données qui viennent Internet (non protégé), les données seront décodées lorsqu'ils l'arriveront. Si le serveur distant n'a pas de service SSL/TLS les données doivent être décodées avant l'envoi. Exemple, pour accéder sur l'interface web de fli4l via le HTTP (HTTPS) avec le service SSL/TLS crypté à travers le tunnel, fli4l est configuré pour recevoir via le SSL/TLS les données cryptées sur le port 443, ils seront décodées puis les transmet au serveur Web du `mini_httpd`, qui écoute sur le port 80.

La configuration pour ces applications sont représentées un peu plus bas.

Exemple : `STUNNEL_1_CLIENT='yes'`

STUNNEL_x_ACCEPT Dans cette variable vous pouvez régler le port (et l'adresse) du tunnel pour "écouter" les connexions entrantes. Il y a deux possibilités :

- Si le tunnel écouter *toutes* les adresses (sur toutes les interfaces). Vous devez indiquer "any" dans cette variable.
- Si le tunnel écoute que sur des adresses spécifiques. Vous devez indiquer la référence appropriée avec l'IP du sous-réseau configurée, par exemple `IP_NET_1_IPADDR` (pour IPv4) ou `IPV6_NET_2_IPADDR` (pour IPv6).

En outre, vous *devez* indiquer derrière l'adresse le numéro de port, pour cela vous devez placer les deux-points (":") entre l'adresse et le port.

14. Voir http://fr.wikipedia.org/wiki/Post_Office_Protocol

Exemple 1 : `STUNNEL_1_ACCEPT='any:443'`

Exemple 2 : `STUNNEL_1_ACCEPT='IP_NET_1_IPADDR:443'`

Exemple 3 : `STUNNEL_1_ACCEPT='IPV6_NET_2_IPADDR:443'`

Il convient de rappeler lors de l'utilisation d'une valeur dans la variable `IP_NET_x_IPADDR` ou dans `IPV6_NET_x_IPADDR` elle est de couche 3 du protocole (IPv4 ou IPv6). Le choix *doit* correspondre au affectation des variables `STUNNEL_x_ACCEPT_IPV4` et `STUNNEL_x_ACCEPT_IPV6`. Donc vous ne pouvez pas désactiver un tunnel IPv6 `STUNNEL_1_ACCEPT_IPV6='no'` et écouter sur une adresse IPv6 avec `STUNNEL_1_ACCEPT='IPV6_NET_2_IPADDR:443'` ; Ceci s'applique également pour la configuration inverse (`STUNNEL_1_ACCEPT_IPV4='no'` en utilisant `IP_NET_x_IPADDR`). En outre, cela dépend aussi du sens du paramètre "any", si vous utilisez la couche 3 du protocole (IPv4 ou IPv6), il écoutera seulement sur des adresses activées, via `STUNNEL_x_ACCEPT_IPV4` et `STUNNEL_x_ACCEPT_IPV6` du protocoles de couche 3.

STUNNEL_x_ACCEPT_IPV4 Dans cette variable vous pouvez indiquer le protocole IPv4 qui peut être utilisé pour la connexion *entrante* du tunnel. Généralement, cette variable devrait contenir la valeur "yes". Si vous indiquez "no" le protocole IPv6 sera utilisé pour la connexion entrante du tunnel. Cependant, cela nécessite une configuration IPv6 valide (reportez-vous à la documentation du paquetage IPv6).

Configuration par défaut : `STUNNEL_x_ACCEPT_IPV4='yes'`

Exemple : `STUNNEL_1_ACCEPT_IPV4='no'`

STUNNEL_x_ACCEPT_IPV6 Cette variable est analogue à la variable `STUNNEL_x_ACCEPT_IPV4`, si le protocole IPv6 est utilisé pour la connexion entrante vous devez indiquer la même valeur pour activer le protocole IPv6 dans le paquetage avec `OPT_IPV6='yes'`. Si vous indiquez "no" le protocole IPv4 sera utilisé pour la connexion entrante du tunnel.

Configuration par défaut : `STUNNEL_x_ACCEPT_IPV6=<valeur de OPT_IPV6>`

Exemple : `STUNNEL_1_ACCEPT_IPV6='no'`

STUNNEL_x_CONNECT Dans cette variable vous indiquez le nom ou l'adresse SSL/TLS du tunnel distant. Cela donne en principe trois possibilités, vous devez ajouter deux points ":" à la suite de ces trois possibilités et indiquer le numéro de port :

- L'adresse IPv4 ou IPv6 numérique.

Exemple 1 : `STUNNEL_1_CONNECT='192.0.2.2:443'`

- Le nom du DNS d'un hôte interne.

Exemple 2 : `STUNNEL_1_CONNECT='@webserver:443'`

- Le nom du DNS d'un hôte externe.

Exemple 3 : `STUNNEL_1_CONNECT='@www.example.com:443'`

Si un hôte interne est indiqué, à la fois pour une adresse IPv4 et IPv6, l'adresse IPv4 sera privilégiée. Si un hôte externe est indiqué, à la fois pour une adresse IPv4 et IPv6, cela dépend du retour de l'adresse du protocole de couche 3, le premier DNS résolu sera utilisé.

STUNNEL_x_OUTGOING_IP Dans cette variable optionnelle vous pouvez indiquer l'adresse locale pour la connexion sortante du tunnel. Cette variable est utilisée que si le tunnel distant a de multiples interfaces (ou routes) actif, par exemple si la cible utilise deux connexions Internet. Normalement, cette variable ne doit pas être configurée.

Exemple : `STUNNEL_1_OUTGOING_IP='IP_NET_1_IPADDR'`

STUNNEL_x_DELAY_DNS Si cette variable optionnelle est activée "yes", le nom de DNS externe utilisé dans STUNNEL_x_CONNECT sera converti en une adresse IP si la connexion *sortante* du tunnel est construite. Ainsi client local sera connecté à la première adresse qui travers le tunnel. Cette variable est utile que si le tunnel distant, est un ordinateur qui ne peut être atteint que par le nom du DNS dynamique et si les changements d'adresse sont fréquent derrière ce nom ou si une connexion active empêche le démarrage du "stunnel".

Configuration par défaut : STUNNEL_x_DELAY_DNS='no'

Exemple : STUNNEL_1_DELAY_DNS='yes'

STUNNEL_x_CERT_FILE Dans cette variable vous indiquez le nom du fichier du certificat pour le tunnel. Tunnel en mode serveur, la variable (STUNNEL_x_CLIENT='no') est désactivée, c'est le certificat du serveur qui est validé par le client contre un "Certificate Authority" (CA) si nécessaire. Tunnel en mode client, la variable (STUNNEL_x_CLIENT='yes') est activée, c'est le (généralement en option) certificat du client qui est validé par le serveur contre un CA si nécessaire.

Le certificat doit être dans le format dit PEM et doit être stocké sous le répertoire <répertoire-config>/etc/stunnel/. Seul le nom du fichier doit être stocké dans cette variable, pas le chemin.

Pour un tunnel en mode serveur le certificat est obligatoire !

Exemple : STUNNEL_1_CERT_FILE='myserver.crt'

STUNNEL_x_CERT_CA_FILE Dans cette variable vous indiquez le nom du fichier du certificat CA à utiliser pour valider le certificat de la station distante. Les clients valident généralement le certificat du serveur, cependant il est également possible de contourner la validation. Quelques détails pour la validation, s'il vous plaît lire la description de la variable [STUNNEL_x_CERT_VERIFY](#) (Page 204) ci-dessous.

Le certificat doit être dans le format dit PEM et doit être stocké sous le répertoire <répertoire-config>/etc/stunnel/. Seul le nom du fichier doit être stocké dans cette variable, pas le chemin.

Exemple : STUNNEL_1_CERT_CA_FILE='myca.crt'

STUNNEL_x_CERT_VERIFY Dans cette variable vous commandez la validation du certificat de la station distante. Il y a cinq options :

- *none* : le certificat de la station distante n'est pas du tout validé. Dans ce cas, la variable STUNNEL_x_CERT_CA_FILE peut rester vide.
- *optional* : le certificat de la station distante est disponible, il sera vérifié en utilisant le certificat CA qui est configuré dans la variable STUNNEL_x_CERT_CA_FILE. Si la station distante n'a *pas* de certificat disponible, cela ne sera pas indiqué comme une erreur et la connexion sera toujours acceptée. Ce paramètre est utile que pour un tunnel en mode serveur, car le tunnel en mode client doit *toujours* obtenir un certificat à partir du serveur.
- *onlyca* : le certificat de la station distante est vérifié en utilisant le certificat CA, qui est configuré dans la variable STUNNEL_x_CERT_CA_FILE. Si la station distante ne diffuse aucun certificat ou s'il ne correspond pas au CA configuré, la connexion sera chargée. Ce paramètre est utile si vous utilisez votre propre CA et que vous connaissez tous les stations distantes potentiels.
- *onlycert* : le certificat de la station distante est comparé avec le certificat qui est configuré dans la variable STUNNEL_x_CERT_CA_FILE. Il ne sera *pas* vérifié en utilisant

le certificat CA, mais il veillera que le certificat de la station distante *envoyé* est exactement le même entre le certificat du (serveur ou du client). Le fichier qui est référencé dans la variable STUNNEL_x_CERT_CA_FILE ne contient pas de CA, mais un certificat hôte. Ce paramètre garantit que seul un serveur déterminé connu (tunnel en mode serveur) peut être construite et connecté à un terminal connu (tunnel en mode client). Ce paramètre est utile pour les connexions peer-to-peer entre hôtes les deux sont sous contrôle et vous n'utilisez pas votre propre CA.

- *both* : le certificat de la station distante est comparé avec le certificat qui est configuré dans la variable STUNNEL_x_CERT_CA_FILE et sera également vérifié en utilisant le certificat CA qui est également dans cette variable. Les *deux* fichiers qui sont référencés dans la variable STUNNEL_x_CERT_CA_FILE, contient un CA et un certificat hôte. C'est une combinaison des options *onlycert* et *onlyca*. En comparaison avec le réglage *onlycert* la connexion sera refusée, si le certificat CA a expiré (même si le certificat de la station distante est validé).

Configuration par défaut : STUNNEL_x_CERT_VERIFY='none'

Exemple : STUNNEL_1_CERT_VERIFY='onlyca'

1ère exemple d'application : connexion au WebGUI de fli4l via le SSL/TLS

Avec cet exemple, le WebGUI de fli4l est prolongé par une connexion SSL/TLS.

```
OPT_STUNNEL='yes'
STUNNEL_N='1'

STUNNEL_1_NAME='http'
STUNNEL_1_CLIENT='no'
STUNNEL_1_ACCEPT='any:443'
STUNNEL_1_ACCEPT_IPV4='yes'
STUNNEL_1_ACCEPT_IPV6='yes'
STUNNEL_1_CONNECT='127.0.0.1:80'
STUNNEL_1_CERT_FILE='server.pem'
STUNNEL_1_CERT_CA_FILE='ca.pem'
STUNNEL_1_CERT_VERIFY='none'
```

2ère exemple d'application : contrôle de deux routeurs fli4l distant via le SSL/TLS sécurisé, par le programme imonc

Avec cet exemple, vous contournez les faiblesses connues du Protocole imonc/imond (envoi des mots de passe en clair) pour une connexion WAN. (Naturellement, la connexion au LAN avec le tunnel peut continuer à être exploitée!)

Configuration local de fli4l dans le LAN (tunnel en mode client) :

```
OPT_STUNNEL='yes'
STUNNEL_N='2'

STUNNEL_1_NAME='remote-imond1'
STUNNEL_1_CLIENT='yes'
STUNNEL_1_ACCEPT='any:50000'
STUNNEL_1_ACCEPT_IPV4='yes'
STUNNEL_1_ACCEPT_IPV6='yes'
```

4. Les paquetages

```
STUNNEL_1_CONNECT='@remote1:50000'  
STUNNEL_1_CERT_FILE='client.pem'  
STUNNEL_1_CERT_CA_FILE='ca+server1.pem'  
STUNNEL_1_CERT_VERIFY='both'
```

```
STUNNEL_2_NAME='remote-imond2'  
STUNNEL_2_CLIENT='yes'  
STUNNEL_2_ACCEPT='any:50001'  
STUNNEL_2_ACCEPT_IPV4='yes'  
STUNNEL_2_ACCEPT_IPV6='yes'  
STUNNEL_2_CONNECT='@remote2:50000'  
STUNNEL_2_CERT_FILE='client.pem'  
STUNNEL_2_CERT_CA_FILE='ca+server2.pem'  
STUNNEL_2_CERT_VERIFY='both'
```

Configuration du premier fli4l distant (tunnel en mode serveur) :

```
OPT_STUNNEL='yes'  
STUNNEL_N='1'
```

```
STUNNEL_1_NAME='remote-imond'  
STUNNEL_1_CLIENT='no'  
STUNNEL_1_ACCEPT='any:50000'  
STUNNEL_1_ACCEPT_IPV4='yes'  
STUNNEL_1_ACCEPT_IPV6='yes'  
STUNNEL_1_CONNECT='127.0.0.1:5000'  
STUNNEL_1_CERT_FILE='server1.pem'  
STUNNEL_1_CERT_CA_FILE='ca+client.pem'  
STUNNEL_1_CERT_VERIFY='both'
```

Configuration du second fli4l distant (tunnel en mode serveur) :

```
OPT_STUNNEL='yes'  
STUNNEL_N='1'
```

```
STUNNEL_1_NAME='remote-imond'  
STUNNEL_1_CLIENT='no'  
STUNNEL_1_ACCEPT='any:50000'  
STUNNEL_1_ACCEPT_IPV4='yes'  
STUNNEL_1_ACCEPT_IPV6='yes'  
STUNNEL_1_CONNECT='127.0.0.1:5000'  
STUNNEL_1_CERT_FILE='server2.pem'  
STUNNEL_1_CERT_CA_FILE='ca+client.pem'  
STUNNEL_1_CERT_VERIFY='both'
```

La connexion à "imond" distant est construite à travers une connexion fli4l local sur le port 5000 (première fli4l distant) et sur le port 5001 (seconde fli4l distant). fli4l se connecte via le tunnel SSL/TLS au fli4l distant respectif, le démon "imond" transmettra à son tour les données via un tiers (hôte interne) vers une connexion distante. Les réglages garantit la validation de chaque fli4l et acceptera uniquement l'autre fli4l comme partenaire de connexion.

4.18. QoS - Qualité de service

Grâce à Qos, vous pouvez régler et partager la bande passante disponible, sur les différents ports, adresses IP et d'autres support.

Le modem gère le Packet-Queue (ou la file d'attente) (Queue = file, en serie (être dans la file)) pour le stockage des paquets, quand la bande passante du modem est inférieur à la quantité de paquet disponible. Pour les modems DSL, la bande passante est plus grande. Ils ont l'avantage d'être relativement homogène et exploite au maximum de bande passante. Le routeur envoie au modem les paquets dans un court laps de temps (très court), le modem doit stockés ces paquets dans une file d'attente pour les traités. Ainsi, cette file d'attente a une organisation très simple, tous les paquets sont rangés et envoyés dans l'ordre d'arrivés, c'est juste un modem :-D

C'est ici que QoS entre en jeux. QoS gère également une file d'attente pour les paquets, dans le routeur lui-même, et là, on a la possibilité d'être le roi, on décide si les paquets doivent être envoyés en premier ou en second. Si QoS est configuré correctement, les paquets seront envoyés judicieusement et rapidement au modem sans qu'il n'attérissent dans la file d'attente du modem, s'est comme si on avait déplacé la file d'attente du modem sur le routeur.

Encore une chose au sujet des unités de vitesse commune : QoS prend en charge les Mibit/s (megabit/s) et les Kibit/s (kilobit/s) et surtout 1Mibit = 1024Kibit.

4.18.1. Configuration

OPT_QOS Vous pouvez placer 'yes' dans la variable OPT_QOS pour activer QoS ou 'no' pour ne pas l'utiliser

QOS_INTERNET_DEV_N Vous indiquez ici Le nombre d'interface du routeur, qui envoie les données sur Internet.

QOS_INTERNET_DEV_x Vous indiquez ici, la liste des interfaces sur lesquels les données seront transmis vers Internet. Exemples :

QOS_INTERNET_DEV_N='3'	Nombre de périphérique
QOS_INTERNET_DEV_1='ethX'	Pour cable et autres liaison Ethernet
QOS_INTERNET_DEV_2='ppp0'	Pour DSL sur protocole PPPoE
QOS_INTERNET_DEV_3='ipppX'	Pour ISDN

Le premier circuit du périphérique ISDN devrait s'appeler ippp0, et le deuxième circuit ippp1 etc ... Mais, si pour le premier circuit l'agrégation des canaux a été activée (2 canaux par circuit), alors du premier circuit avec deux canaux s'appellera ippp1 et le deuxième Circuit s'appellera ippp2. Il convient pour bénéficier du QoS avec ISDN, de désactiver l'agrégation des canaux des circuits.

QOS_INTERNET_BAND_DOWN On indique ici la bande passante maximum montante pour un accès Internet. Voir ci-dessus pour [les unités de vitesse](#) (Page 207).

Remarque : pour l'actualisation des tâches, comme pour l'affectation des paquets-ACK, il ne faut pas indiquer une quantité de bande passante (ou B-P) supérieure à la B-P vraiment disponible, les paquets de la file d'attente du routeur seront triés comme il se doit, mais cela ne sera pas tout à fait exacte, et finalement les paquets seront stockés dans la file d'attente du modem et sera ralenti. Il est possible que le fournisseur d'accès n'indique pas réellement la bande passante correspondante, cela peut être un peu plus ou un peu moins. Il faut donc faire des tests. Pour trouver la bonne quantité.

QOS_INTERNET_BAND_UP On indique ici la bande passante maximum descendante pour l'accès Internet. Voir OPT_QOS, pour les unités de vitesse.

Remarque : voir les informations ci-dessus QOS_INTERNET_BAND_DOWN.

QOS_INTERNET_DEFAULT_DOWN On indique ici les classes par défaut des paquets qui proviennent d'Internet. Tous les paquets qui n'ont pas été paramétrés dans les classes filtrées, atterrissent ensuite dans la classe supplémentaire indiqué ici.

Si aucune classe n'est paramètre et si la variable est sur

```
QOS_CLASS_x_DIRECTION='down'
```

alors, on indiquera :

```
QOS_INTERNET_DEFAULT_DOWN='0'
```

Exemple :

On met en place 2 classes et un seul filtre pour les paquets. On paramètre par ex. une adresse-IP spécifique à la 1ère des deux classes. Tous les autres paquets passeront par la 2ème classe, c'est une classe par défaut qui ne sera pas filtrée. On indique par conséquent dans la variable :

```
QOS_INTERNET_DEFAULT_DOWN='2'
```

Il faut faire attention à se que la variable QOS_INTERNET_DEFAULT_DOWN a une classe de paramétrée et que dans la variable QOS_CLASS_x_DIRECTION le paramètre 'down' soit indiqué.

QOS_INTERNET_DEFAULT_UP On indique ici les classes par défaut, pour des paquets qui vont sur Internet. Tous les paquets qui n'ont pas été paramétrés dans les classes filtrées, atterrissent ensuite dans la classe supplémentaire indiqué ici.

Si aucune classe n'est paramètre et si la variable est sur

```
QOS_CLASS_x_DIRECTION='up'
```

alors, on indiquera :

```
QOS_INTERNET_DEFAULT_UP='0'
```

L'ensemble des fonctions sont similaire à la variable QOS_INTERNET_DEFAULT_DOWN.

Il faut faire attention à se que la variable QOS_INTERNET_DEFAULT_UP a une classe de paramétrée et que dans la variable QOS_CLASS_x_DIRECTION le paramètre 'up' soit indiqué.

QOS_CLASS_N On indique ici le nombre souhaité de classe.

QOS_CLASS_x_PARENT Dans cette variable, on peut superposer les classes. On indique toujours un numéro pour la classe parent et on attribue la bande passante sur la classe parent, elle peut aussi être répartie en sous-classe. Le niveau maximal de d'imbrication s'élève à 8 niveaux, l'interface 0 représente déjà 1 niveau, il reste donc un maximum de 7 niveaux configurable

Si dans la classe il n'y a pas de sous-classe on indique :

```
QOS_CLASS_x_PARENT='0'
```

suivant la direction dont elle fait partie (avec QOS_CLASS_x_PORT) ou avec QOS_CLASS_x-PORT_TYPE et aussi la bande passante attribuée avec QOS_INTERNET_BAND_DOWN

Important : Si vous n'avez pas indiqué '0', il est important de s'assurer, que la classe se trouve bien entendu dans le niveau le plus haut (par rapport à la numérotation des niveaux).

QOS_CLASS_x_MINBANDWIDTH Vous indiquez ici bande passante minimum, que l'on veut attribuer aux classes. On peut parler aussi de rapport. Voir les indication sur les unités de vitesse dans OPT_QOS.

Exemple de classe, donc vous voulez limiter la bande passante à 128Kibit/s :

```
QOS_CLASS_1_PARENT='0'  
QOS_CLASS_1_MAXBANDWIDTH='128Kibit/s'  
QOS_CLASS_1_MINBANDWIDTH='128Kibit/s'
```

Maintenant on dispose de 3 classes supplémentaires, ont paramètre les variables QOS_CLASS_x_MINBANDWIDTH et QOS_CLASS_x_MAXBANDWIDTH des sous-classes de notre première classe :

```
QOS_CLASS_2_PARENT='1'  
QOS_CLASS_2_MINBANDWIDTH='60Kibit/s'  
QOS_CLASS_2_MAXBANDWIDTH='128Kibit/s'  
  
QOS_CLASS_3_PARENT='1'  
QOS_CLASS_3_MINBANDWIDTH='40Kibit/s'  
QOS_CLASS_3_MAXBANDWIDTH='128Kibit/s'  
  
QOS_CLASS_4_PARENT='1'  
QOS_CLASS_4_MINBANDWIDTH='28Kibit/s'  
QOS_CLASS_4_MAXBANDWIDTH='128Kibit/s'
```

Toutes les sous-classes peuvent avoir la même (ou pas) priorité (voir QOS_CLASS_x_PRIO). Maintenant un débit respectif est produit dans chacune des 3 classes avec la variable QOS_CLASS_x_MINBANDWIDTH. Ainsi on attribut dans chaque classe la bande passante correspondante dans QOS_CLASS_x_MINBANDWIDTH. Exemple si vous modifier la classe 2 et que vous lui attribuez 20Kibit/s, il restera donc 40Kibit/s à attribuer aux "autres" classes. Cette excédent peut être réparti avec la relation 40/28 dans les classes 3 et 4. Avec la variable QOS_CLASS_x_MAXBANDWIDTH chaque classe sera limité à 128Kibit/s, étant donné que toutes les sous-classes sont limitées à 128Kibit/s, ils ne pourront pas dépasser le débit autorisé de 128Kibit/s.

QOS_CLASS_x_MAXBANDWIDTH Vous indiquez ici bande passante maximum, que l'on veut attribuer aux classes. Il n'y a pas sens, d'indiquer une valeur plus basse que la variable QOS_CLASS_x_MINBANDWIDTH. Si l'on indique aucune valeur, la variable prend automatiquement la valeur qui est indiqué dans QOS_CLASS_x_MINBANDWIDTH. bien entendu pour cette classe il ne faut pas indiquer un surplus de bande passante.

Voir OPT_QOS pour les unités de vitesse.

QOS_CLASS_x_DIRECTION On indique dans cette variable la direction de la bande passante, donc la classe fait partie. Si elle fait parti du débit montant, on indique :

```
QOS_CLASS_x_DIRECTION='up'
```

Si elle fait parti du débit descendant, on indique :

```
QOS_CLASS_x_DIRECTION='down'
```

Attention : Up-stream (débit montant) c'est la transmission de votre ordinateur au serveur-Internet et Down-stream (débit descendant) c'est la transmission du serveur-Internet à votre ordinateur.

QOS_CLASS_x_PRIO Dans cette variable on règle le niveau de priorité de la classe. Plus le chiffre est bas plus le niveau de priorité est élevé. Les chiffres autorisés sont de 0 à 7. Si la variable est laissée vide le niveau de priorité sera à 0 donc le plus haut.

Lorsqu'il y a un excédent de bande passante, le système détermine la classe et ça priorité pour augmenter le débit. Pour expliquer cela, nous allons modifier légèrement l'exemple QOS_CLASS_x_MINIMUMBANDWIDTH, la première classe rien n'a été changée. On attribue à la classes 4 une priorité 2 :

```
QOS_CLASS_2_PARENT='1'
QOS_CLASS_2_MINBANDWIDTH='60Kibit/s'
QOS_CLASS_2_MAXBANDWIDTH='128Kibit/s'
QOS_CLASS_2_PRIO='1'

QOS_CLASS_3_PARENT='1'
QOS_CLASS_3_MINBANDWIDTH='40Kibit/s'
QOS_CLASS_3_MAXBANDWIDTH='128Kibit/s'
QOS_CLASS_3_PRIO='1'

QOS_CLASS_4_PARENT='1'
QOS_CLASS_4_MINBANDWIDTH='28Kibit/s'
QOS_CLASS_4_MAXBANDWIDTH='128Kibit/s'
QOS_CLASS_4_PRIO='2'
```

Dans cette exemple si la classe 2 ne consomme que 20Kibit/s, il y a donc un excédent de 40Kibit/s. les Classes 3 et 4 peuvent recevoir encore plus de bande passante. Cependant, la classe 3 a une priorité plus élevée que la classe 4, donc la classe 3 peut récupérer l'excédent de la bande passante les 40Kibit/s.

Cependant si la classe 3 a besoin seulement de 20Kibit/s de l'excédent des 40Kibit/s, alors la classe 4 recevra les 20Kibit/s restant.

Prenons un autre exemple, si la classe 4 ne consomme pas la bande passante et si les classes 2 et 3 on besoin de plus de bande passante. Alors, chaque classes utilise la bande passante spécifiée dans la variable QOS_CLASS_x_MINBANDWIDTH le reste sera divisé entre les deux avec le rapport 60/40, puisque les deux classes ont la même priorité.

La variable QOS_CLASS_x_PRIO influe seulement sur l'excédent de bande passante, pour éventuellement là répartir.

QOS_CLASS_x_LABEL Avec cette variable optionnelle, vous pouvez placer un intitulé pour une classe. Cet intitulé sera affiché pour le graphique de QOS dans OPT_RRDTOOL s'il est activé.

QOS_FILTER_N Vous indiquez ici le nombre de filtre souhaité.

Au sujet du filtrage, en général on peut dire ceci : les paramètres des différentes variables sont reliées et/ou les différentes paramètres d'une même variable sont aussi reliées. on peut dire par exemple : que dans un même filtre nous pouvons filtrer une adresse IP et un port, ainsi seul des paquets filtrés seront envoyé vers la cible de la (n) classe, et seront appliqués sur l'un et l'autre paramètre simultanément.

Un autre exemple : Dans le même filtre il y a deux ports (21 et 80) et une adresse IP. Bien sûr, un paquet de données ne peut pas être envoyé sur les deux ports en même temps. Le filtre se comporte alors comme ceci : les paquets filtrés utilisent le port 21 et en même temps l'adresse IP, ou le port 80 et en même temps l'adresse IP.

Important : Cela dépend de la séquence de filtrage !

Un exemple : on veut transporter les paquets, par l'intermédiaire du port 456 qui est ouvert, pour **tous** les ordinateurs de la classe A. De plus on voudrait faire passer tous les paquets de l'ordinateur de l'adresse IP 192.168.6.5 - par le port 456 ouvert - dans la classe B. Je règle maintenant le filtre de la première adresse IP, alors tous les paquets arrivent - sur le port 456 ouvert - dans la classe B, je règle un autre filtre pour le port 456 qui n'est pas altéré. Le filtre pour le port 456 doit donc être avant le filtre de l'adresse IP 192.168.6.5

QOS_FILTER_x_CLASS On indique dans cette variable, les classes qui doivent s'appliquer aux paquets filtrés. Par exemple, on veut filtrer les paquets de la variable spécifié ici `QOS_CLASS_25_MINBANDWIDTH` avec le numéro de classe, vous devez indiquer :

```
QOS_FILTER_x_CLASS='25'
```

Avec la variable `QOS_CLASS_x_DIRECTION` on peut indiquer ne classe appartenant maintenant au débit montant ou au débit descendant. Si maintenant on met en place un filtre pour filtrer les paquets par exemple dans la classe du débit montant, seul des paquets du débit montant seront ainsi filtrés par ce filtre et seront installé pour la classe donnée. Avec cette variable `QOS_CLASS_x_DIRECTION` on détermine la "direction" du débit à filtrer. Depuis la version 2.1, il est désormais possible de l'indiquer plus d'une classe. Par exemple si vous souhaitez, envoyer le trafic sur le port 456 à la fois dans les classes des débit montant et descendant, vous pouvez indiquer :

```
QOS_FILTER_x_CLASS='4 25'
```

Avec les numéros de classe 4 pour le débit montant et de classe 25 pour le débit descendant. Il n'y a aucun sens à indiquer ici un débit montant et descendant, cela est purement objectif à ne pas recopier.

QOS_FILTER_x_IP_INTERN On indique dans cette variable les adresses-IP et les adresses de domaine du réseau interne, qui doivent être filtrés. Elles sont séparées par un espace et peuvent être combinés librement.

Cela pourrait être par exemple :

```
QOS_FILTER_x_IP_INTERN='192.168.6.0/24 192.168.5.7 192.168.5.12'
```

Ici, toutes les adresses sous la forme 192.168.6.X sont filtrés en plus des adresses IP 192.168.5.7 et 192.168.5.12.

Cette variable peut aussi rester vide.

Si vous utilisez cette variable `QOS_FILTER_x_IP_EXTERN` avec une adresses-IP ou une plage d'adresses-IP, il y aura pas de trafic filtré entre `QOS_FILTER_x_IP_INTERN` et `QOS_FILTER_x_IP_EXTERN`.

Important: Si vous ajoutez la variable `QOS_FILTER_x_OPTION` avec les options de filtrage *ACK*, *TOSMD*, *TOSMT*, *TOSMR* ou *TOSMC*, la variable `QOS_CLASS_x_DIRECTION` avec le paramètre 'down' sera alors ignorée.

QOS_FILTER_x_IP_EXTERN On indique dans cette variable les adresses-IP et les adresses de domaine du réseau externe qui doivent être filtrés (elle se rapporte à la variable `QOS_INTERNET_DEV`). Les adresses sont séparées par un espace et peuvent être combinés librement. L'ensemble fonctionne de la même manière que `QOS_FILTER_x_IP_INTERN`.

Cette variable peut aussi rester vide.

Important: Si vous ajoutez la variable `QOS_FILTER_x_OPTION` avec les options de filtrages `ACK`, `TOSMD`, `TOSMT`, `TOSMR` ou `TOSMC`, la variable `QOS_CLASS_x_DIRECTION` avec le paramètre 'down' sera alors ignorée.

QOS_FILTER_x_PORT On peut paramétrer dans cette variable un port ou une plage de ports, ils seront séparés par un espace et peuvent être combinés librement. Si la variable est vide, le trafic se fait sur tous les ports.

Au sujet de la plage de port : si l'on souhaite filtrer l'ensemble des ports de 5000 à 5099, on indiquera :

```
QOS_FILTER_x_PORT='5000-5099'
```

Un autre exemple : On voudrait filtrer le trafic des ports 20 à 21, 137 à 139 et le port 80, dans la même classe. Alors, on indiquera :

```
QOS_FILTER_x_PORT='20-21 137-139 80'
```

Cette variable peut aussi être laissée vide.

Important :

- Avec le filtrage de port, la variable `QOS_FILTER_x_PORT_TYPE` doit aussi être activée.
- Si la variable `QOS_FILTER_x_OPTION` est activée avec les options de filtrages `ACK`, `TOSMD`, `TOSMT`, `TOSMR` ou `TOSMC`, la plage de ports sera ignorée.

QOS_FILTER_x_PORT_TYPE Cette variable est seulement importante, que si la variable `QOS_FILTER_x_PORT` est utilisée, dans ce cas on peut la paramétrer (autrement elle sera simplement ignorée).

Puisque les ports se différencient entre les ports du service-client et les ports du service-serveur, vous devez indiquer ici si les ports du serveur ou du client sont visés. Les ordinateurs du réseau seront considérés comme points de référence.

Les réglages suivants sont possibles :

```
QOS_FILTER_x_PORT_TYPE='client'
QOS_FILTER_x_PORT_TYPE='server'
```

Depuis la version 2.1 la combinaison des deux arguments dans une seule variable est possible, pour le trafic sur les ports dans n'importe quel réseau et pour le trafic sur les ports externes pour Internet, dans la même classe, par exemple :

```
QOS_FILTER_x_PORT_TYPE='client server'
```

Cela correspond à l'élaboration de deux filtres semblables, dans lequel on a placé le client et le serveur sur la même variable `QOS_FILTER_x_PORT_TYPE`.

QOS_FILTER_x_OPTION Avec cette variable on indique d'autres options pour le filtre actif. Il ne faut pas spécifier plus d'un paramètre, car une combinaison de plusieurs paramètres dans le même filtre n'a pas de sens. En revanche, il est parfaitement possible et même parfois utile, de filtrer les paquets-ACK, et un 2ème paquet. Par exemple filtrer les paquets-TOSMD, dans la même destination de la classe (voir `QOS_FILTER_x_CLASS`).

ACK Paquet de confirmation.

Si on applique ce paquet dans la variable option, lorsqu'un paquet de données arrive il sera renvoyé un paquet de confirmation comme un "accusé de réception". Si vous

téléchargez de grands fichiers beaucoup de paquets sont transmis, pour chaque paquet transmis vous devez envoyer une confirmation ou un accusé de réception pour indiquer que le paquet de données est bien arrivé. Si la confirmation se fait attendre, l'expéditeur attendra celui-ci avant d'envoyer un nouveau paquet, le prochain paquet ne sera pas pour vous si la confirmation n'est pas reçue.

Toutes les connexions asymétriques sont particulièrement importantes, actuellement (le débit montant/descendant est inégal) dans la plus par des offres DSL privées. Généralement, le débit montant est beaucoup plus faible, poussé à ses limites les paquets sont empilés avant d'être envoyés et indiscutablement quelque part dans l'immense tas, il y a les petits paquets de confirmation. Dans des circonstances normales, cela fonctionne séquentiellement. Jusqu'à ce que le paquet de confirmation à son tour est envoyé, il serait bien que notre expéditeur de données puisse faire une petite pause, mais ce n'est pas bon pour le débit descendant.

Nous devons veiller à ce que les paquets de confirmations soient bien à la suite, pour que les paquets de confirmation soient envoyés "normalement" au fur et à mesure, afin que l'expéditeur reçoive bien la confirmation. Cette option est logique d'être combinée à une classe, pour cet exemple d'application.

ICMP Paquet-Ping (Protocole ICMP)

On utilise un paquet Ping pour mesurer le temps en seconde que met ce paquet pour aller du point A au point B. Si vous voulez donner au paquet Ping une priorité plus élevée, vous pouvez indiquer cette valeur dans la variable option. Cette valeur n'a rien à voir pour jouer sur Internet. Ce n'est pas parce que vous avez activé le paquet Ping que vous serez prioritaire et que vous aurez un super Ping pour jouer sur le Net...

IGMP Paquet-IGMP (Protocole IGMP)

Si vous utilisez la TV par IP, le protocole IGMP sera utile pour le filtrage et la hiérarchisation.

TCP Petit paquet TCP

Vous pouvez utiliser ce filtrage pour les requêtes HTTP(s) sortantes, ce filtre sera prioritaire. Une combinaison avec un port de destination est possible et judicieux. Taille des paquets TCP : max. 800 octets.

TCP Paquet-TCP (Protocole TCP)

Avec ce paquet on filtre uniquement les paquets qui utilisent le protocole TCP.

UDP Paquet-UDP (Protocole UDP)

Avec ce paquet on filtre uniquement les paquets qui utilisent le protocole UDP.

TOS* Type of Service

TOS "Type of Service" est une application qui place 4 Bit-TOS dans l'entête IP pour chaque paquet envoyé. Ceux-ci ont pour effet de modifier la manière dont les paquets sont traités. Par exemple, on peut placer TOS-Minimum-Delay pour Améliore la réactivité SSH et TOS-Maximum-Throughput pour expédition de fichiers. Généralement ce sont des programmes Linux/Unix qui utilisent plus fréquemment ces Bit-TOS, que les programme Windows. En outre, on peut placer ces Bit-TOS pour certains paquets à destination des IT Firewall. Cela dépend bien sûr que les Routeurs acceptent ou pas les Bit-TOS. En réalité pour fl4l Minimum-Delay et Maximum-Throughput présentent un vrai intérêt.

TOSMD - TOS Minimum-Delay Ce service est utilisé pour améliorer la réactivité des connexions en réduisant le délai de transmission des paquets, Bit-TOS est recommandé pour le SSH, Telnet, FTP (contrôle), TFTP.

TOSMT - TOS Maximum-Troughput Ce service est utilisé pour améliorer le débit des gros de fichier, au prix d'une possible détérioration de l'interactivité de la session. Les temps de latence ne sont pas importants, Bit-TOS est recommandé pour le FTP-data et WWW.

TOSMR - TOS Maximum-Reliability Ce service est utilisé pour avoir la certitude que les données arrivent sans perte, améliore la fiabilité, on évite un revoie de paquet inutile, Bit-TOS est recommandé pour SNMP et DNS

TOSMC - TOS Minimum-Cost Ce service est utilisé pour minimiser le délai, une meilleure rentabilité, Bit-TOS est recommandé pour NNTP, SMTP et ICMP.

DSCP* Differentiated Services Code Point

Le DSCP est appelé marquage conformément défini dans la RFC 2474. Ce processus de marquage a largement remplacé le TOS depuis 1998.

Le filtrage des classes DSCP peut être configuré comme ceci :

```
QOS_FILTER_x_OPTION='DSCPef'  
QOS_FILTER_x_OPTION='DSCPcs3'
```

S'il vous plaît, faite attention que le DSCP soit écrit en lettre capital et la classe en minuscule.

Les classes suivantes peuvent être utilisées :

af11-af13, af21-af23, af31-af33, af41-af43, cs1-cs7, ef et be (par défaut)

4.18.2. Applications et exemples

Comment doit on configurer OPT_QoS précisément ? voici quelques exemples concret :

- Exemple 1 : exemple simple pour le partager de la bande passante sur trois ordinateurs.
- Exemple 2 : exemple de configuration pour la distribution de la bande passante sur 2 ordinateurs et en plus une seconde répartition de la bande passante sur les ports des ordinateurs respectif, en plus il restera de la B-P.
- Exemple 3 : exemple de fonctionnement général de QoS avec un peut plus de détail.
- Exemple 4 : exemple de configuration des paquets-ACK avant d'être transmis, sur le débit descendant de sorte que le débit montant ne chute pas.

Exemple 1

L'objectif de cet exemple simple, est de distribuer la bande passante sur 3 ordinateurs.

Pour ce faire, nous allons créer 4 classes (voir QOS_CLASS_N et se qui suit) pour différentes vitesses (voir QOS_CLASS_x_MINBANDWIDTH / QOS_CLASS_x_MINBANDWIDTH) cela dépend aussi de la classe 0 (voir QOS_CLASS_x_PARENT) Donc directement de l'interface pour "up" ou "down" (voir QOS_CLASS_x_DIRECTION).

La classe 4 est éventuellement pour un hôte en plus, avec moins de band passante. nous indiquons dans QOS_INTERNET_DEFAULT_DOWN='4' pour tous le transport non filtré et une quatrième Classe pour les "invités". Étant donné que nous avons rarement d'hôte, la bande passante sera

4. Les paquetages

la même pour les 3 autres classes, chaque ordinateur recevra 1/3 de l'ensemble de la bande passante, c'est-à-dire approximativement 256Kibit/s.

Nous avons tout d'abord configuré la structure fondamentale. Maintenant, nous devons encore choisir pour le réglage quelle transport pour quelle classe.

Pour ce faire, nous allons utiliser des filtres, pour associer le trafic à chacune des classes. Nous allons créer ainsi des filtres pour les 3 ordinateurs (voir QOS_FILTER_N et se qui suit) et la classification des filtres pour chaque classe (voir QOS_FILTER_x_CLASS). Maintenant, nous pouvons avec QOS_FILTER_x_IP_INTERN, QOS_FILTER_x_IP_EXTERN, QOS_FILTER_x_PORT, QOS_FILTER_x_PORT et QOS_FILTER_x_OPTION déterminer le réglage de chacune des classes, pour laquelle appartient le filtre.

Vous pouvez voir le principe dans la figure 4.7 l'Interface 0 les 3 classe 1, 2 et 3, les 3 filtres F1, F2 et F3.

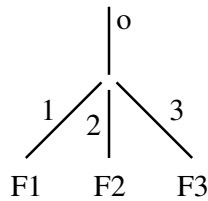


FIGURE 4.7.: exemple 1

La configuration ressemble alors à :

Trois ordinateurs filtrés par IP reçoivent chacune 1/3 du débit, si il n'y a pas d'hôte en plus

```
OPT_QOS='yes'
QOS_INTERNET_DEV_N='1'
QOS_INTERNET_DEV_1='ppp0'
QOS_INTERNET_BAND_DOWN='768Kibit/s'
QOS_INTERNET_BAND_UP='128Kibit/s'
QOS_INTERNET_DEFAULT_DOWN='4'
QOS_INTERNET_DEFAULT_UP='0'

QOS_CLASS_N='4'

QOS_CLASS_1_PARENT='0'
QOS_CLASS_1_MINBANDWIDTH='232Kibit/s'
QOS_CLASS_1_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_1_DIRECTION='down'
QOS_CLASS_1_PRIO=''

QOS_CLASS_2_PARENT='0'
QOS_CLASS_2_MINBANDWIDTH='232Kibit/s'
QOS_CLASS_2_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_2_DIRECTION='down'
QOS_CLASS_2_PRIO=''

QOS_CLASS_3_PARENT='0'
QOS_CLASS_3_MINBANDWIDTH='232Kibit/s'
QOS_CLASS_3_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_3_DIRECTION='down'
```

4. Les paquetages

```
QOS_CLASS_3_PRIO=''

QOS_CLASS_4_PARENT='0'
QOS_CLASS_4_MINBANDWIDTH='72Kibit/s'
QOS_CLASS_4_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_4_DIRECTION='down'
QOS_CLASS_4_PRIO=''

QOS_FILTER_N='3'

QOS_FILTER_1_CLASS='1'
QOS_FILTER_1_IP_INTERN='192.168.0.2'
QOS_FILTER_1_IP_EXTERN=''
QOS_FILTER_1_PORT=''
QOS_FILTER_1_PORT_TYPE=''
QOS_FILTER_1_OPTION=''

QOS_FILTER_2_CLASS='2'
QOS_FILTER_2_IP_INTERN='192.168.0.3'
QOS_FILTER_2_IP_EXTERN=''
QOS_FILTER_2_PORT=''
QOS_FILTER_2_PORT_TYPE=''
QOS_FILTER_2_OPTION=''

QOS_FILTER_3_CLASS='3'
QOS_FILTER_3_IP_INTERN='192.168.0.4'
QOS_FILTER_3_IP_EXTERN=''
QOS_FILTER_3_PORT=''
QOS_FILTER_3_PORT_TYPE=''
QOS_FILTER_3_OPTION=''
```

L'option QOS_INTERNET_DEFAULT_UP a été mis sur 0 parce le débit montant ne devrait pas être limité.

Exemple 2

L'objectif de cet exemple est de distribuer la bande passante sur 2 ordinateurs, une seconde répartition de la bande passante se fera sur les ports des deux ordinateurs respectif, il restera de la Bande passante pour le transfert.

Pour cela, nous avons de 2 classes avec leur vitesse respective, ils dépendent directement de l'interface pour "up" et/ou "down" (voir le premier exemple). Ensuite nous allons créer pour le première ordinateur de la première classe deux autre sous classes. Les deux sous classes sont créées comme la première classe directement sur l'interface, avec toutefois une particularité : la variable QOS_CLASS_x_PARENT n'est pas sur 0, mais sur le nombre de classe à laquelle les sous-classes sont attachées. Par exemple pour QOS_CLASS_1, nous devons maintenant ajouté 1 dans la classes QOS_CLASS_1, suivante. Il en sera de même pour le second ordinateur. On attache à nouveau deux sous classes à la classe du deuxième ordinateur, ils doivent être sur 2. Ceci peut être fait, non seulement pour deux ordinateurs, mais pour autant d'ordinateur que vous voulez. De même on peut créer autant de sous-classes par rapport à la classe supérieur.

nous avons tout d'abord configurer la structure fondamentale. Ensuite nous devons assigner les filtres de chaque classe pour le trafic. (Voir le premier exemple)

4. Les paquetages

Nous allons créer 2 filtres pour le premier ordinateur et 2 filtres pour le deuxième ordinateur. Il y aura respectivement un filtre pour le port, et d'un filtre pour le transfert des données. Il faut absolument tenir compte des séquences. le premier filtre uniquement pour le port, puis le reste. Il est impératif de respecter l'ordre. Autrement le filtre sera déjà attribué pour le reste les classes.

Vous pouvez voir le principe dans la figure 4.8 l'Interface 0, les 6 classes 1, 2, 3, 4, 5, et 6, les 4 filtres F1, F2, F3 et F4.

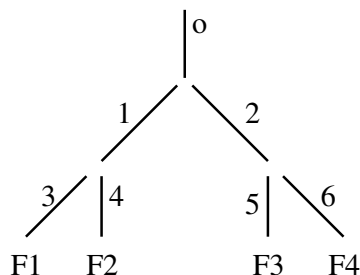


FIGURE 4.8.: exemple 2

La configuration ressemble alors à :

Deux classes pour 2 ordinateurs qui obtiennent 1/2 de BP, deux classes pour les ports qui obtiennent 2/3 de la B-P, il reste donc 1/3 de B-D pour chaque classe parent :

```
OPT_QOS='yes'
QOS_INTERNET_DEV_N='1'
QOS_INTERNET_DEV_1='ppp0'
QOS_INTERNET_BAND_DOWN='768Kibit/s'
QOS_INTERNET_BAND_UP='128Kibit/s'
QOS_INTERNET_DEFAULT_DOWN='7'
QOS_INTERNET_DEFAULT_UP='0'

QOS_CLASS_N='6'

QOS_CLASS_1_PARENT='0'
QOS_CLASS_1_MINBANDWIDTH='384Kibit/s'
QOS_CLASS_1_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_1_DIRECTION='down'
QOS_CLASS_1_PRIO=''

QOS_CLASS_2_PARENT='0'
QOS_CLASS_2_MINBANDWIDTH='384Kibit/s'
QOS_CLASS_2_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_2_DIRECTION='down'
QOS_CLASS_2_PRIO=''

QOS_CLASS_3_PARENT='1'
QOS_CLASS_3_MINBANDWIDTH='256Kibit/s'
QOS_CLASS_3_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_3_DIRECTION='down'
QOS_CLASS_3_PRIO=''
```

4. Les paquetages

```
QOS_CLASS_4_PARENT='1'  
QOS_CLASS_4_MINBANDWIDTH='128Kibit/s'  
QOS_CLASS_4_MAXBANDWIDTH='768Kibit/s'  
QOS_CLASS_4_DIRECTION='down'  
QOS_CLASS_4_PRIO=''
```

```
QOS_CLASS_5_PARENT='2'  
QOS_CLASS_5_MINBANDWIDTH='256Kibit/s'  
QOS_CLASS_5_MAXBANDWIDTH='768Kibit/s'  
QOS_CLASS_5_DIRECTION='down'  
QOS_CLASS_5_PRIO=''
```

```
QOS_CLASS_6_PARENT='2'  
QOS_CLASS_6_MINBANDWIDTH='128Kibit/s'  
QOS_CLASS_6_MAXBANDWIDTH='768Kibit/s'  
QOS_CLASS_6_DIRECTION='down'  
QOS_CLASS_6_PRIO=''
```

```
QOS_FILTER_N='4'
```

```
QOS_FILTER_1_CLASS='3'  
QOS_FILTER_1_IP_INTERN='192.168.0.2'  
QOS_FILTER_1_IP_EXTERN=''  
QOS_FILTER_1_PORT='80'  
QOS_FILTER_1_PORT_TYPE='client'  
QOS_FILTER_1_OPTION=''
```

```
QOS_FILTER_2_CLASS='4'  
QOS_FILTER_2_IP_INTERN='192.168.0.2'  
QOS_FILTER_2_IP_EXTERN=''  
QOS_FILTER_2_PORT=''  
QOS_FILTER_2_PORT_TYPE=''  
QOS_FILTER_2_OPTION=''
```

```
QOS_FILTER_3_CLASS='5'  
QOS_FILTER_3_IP_INTERN='192.168.0.3'  
QOS_FILTER_3_IP_EXTERN=''  
QOS_FILTER_3_PORT='80'  
QOS_FILTER_3_PORT_TYPE='client'  
QOS_FILTER_3_OPTION=''
```

```
QOS_FILTER_4_CLASS='6'  
QOS_FILTER_4_IP_INTERN='192.168.0.3'  
QOS_FILTER_4_IP_EXTERN=''  
QOS_FILTER_4_PORT=''  
QOS_FILTER_4_PORT_TYPE=''  
QOS_FILTER_4_OPTION=''
```

Avec cet exemple l'option `QOS_INTERNET_DEFAULT_DOWN` a été choisie de telle sorte que le transfert qui n'est pas assigné par une classe et un filtre, soit mis dans une classe non existante (non paramétrée). La raison, dans cet exemple, c'est qu'il reste 1/3 de bande passante non affectée. c'est pour cela que l'on conduit cette B-P sur une classe non existante, en plus elle

est transmise très lentement. S'il reste de la B-P dans la configuration vous devez absolument, l'indiquer dans une classe existante.

L'option `QOS_INTERNET_DEFAULT_UP` a été réglé sur 0 pour que Upstream (ou débit montant) ne soit pas limité.

Example 3

Exemple du mode de fonctionnement de QoS en général ou presque.

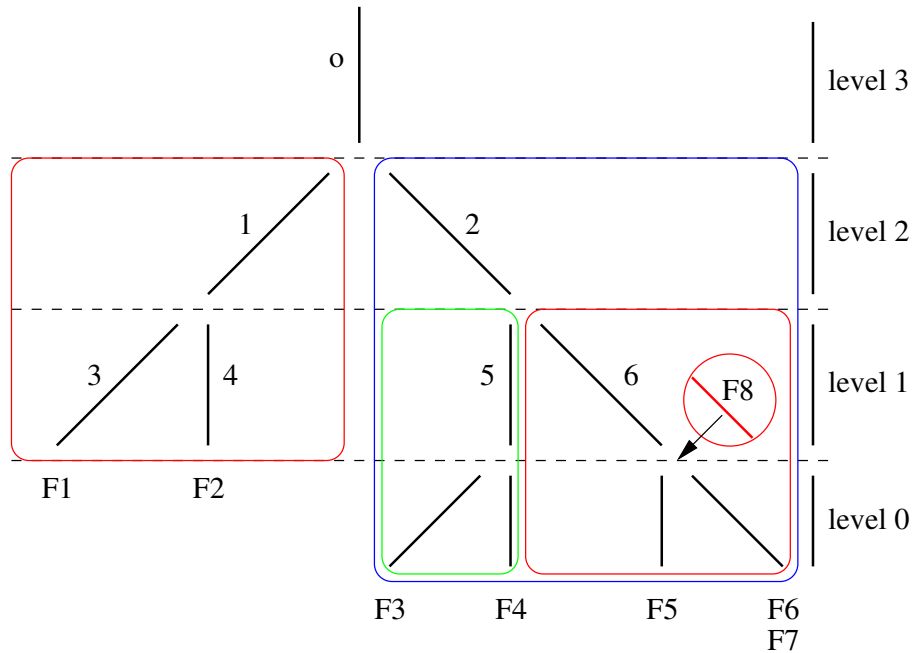


FIGURE 4.9.: exemple 3

Dans la figure 4.9 nous allons revoir la répartition du deuxième exemple, mais avec une extension supplémentaire. On ajoutées deux sous classe à la sous classe du deuxième niveau. Il est possible d'imbriquer des classes encore plus profondément dans cette représentation, la limite actuelle se situe ici à 8 branches, on peut produire au maximum de 7 branches après l'interface 0, ensuite c'est terminé. Cependant, dans la "largeur" aucune limite n'est fixée. Donc vous pouvez rajouter à l'une des sous-classes autant de classes que vous voulez.

On peut voir sur cette figure, qu'il est aussi possible d'intégrer plus d'un filtre à une classe comme avec la classe 10. Il est à noter au sujet du filtrage, qu'il n'est pas possible de placer un filtre au milieu de "l'arborescence" comme indiqué en F8.

Voyons maintenant, encore une fois le sens des classes et des sous-classes. Les classes sont réglementées et sont utilisées pour régler la vitesse de connexion. La répartition de la vitesse est effectué avec la variable `QOS_CLASS_x_MINBANDWIDTH`. Cependant, cela peut avoir un inconvénient si par exemple, toutes les sous classes dépendent d'une classe. Si on donnait, par exemple, à un ordinateur la moitié de la bande passante et l'autre moitié répartie en 2/3 pour le http et 1/3 pour le reste, c'est-à-dire 2/6 et 1/6 de l'ensemble. Cela se présente maintenant : à pleine charge chacune des branches reçoit une moitié. Quant à la deuxième, oui il ne reste pour le http que 2/6. Cependant, le 2e ordinateur ne reçoit pas les 2/6, mais cela est réparti selon la

méthode décrite. Pour éviter ceci, on crée des sous-classes. Le trafic des classes est distribué d'abord aux sous-classes, si celle-ci ne demande pas le transfert complet de la bande passante, le restant sera réparti sur d'autres classes. dans la figure, les zones qui sont ensembles sont entourées, en Rouge = 1, bleu = 2, vert = 5 et orange = 6

Exemple 4

Exemple de configuration, pour la priorité des paquets-ACK et pour que le Downstream (ou débit descendant) ne chute pas, il nous faut simultanément un Upstream (ou débit montant) fort :

```
OPT_QOS='yes'
QOS_INTERNET_DEV_N='1'
QOS_INTERNET_DEV_1='ppp0'
QOS_INTERNET_BAND_DOWN='768Kibit/s'
QOS_INTERNET_BAND_UP='128Kibit/s'
QOS_INTERNET_DEFAULT_DOWN='0'
QOS_INTERNET_DEFAULT_UP='2'
```

Nous configurons ici l'interface pour l'accès Internet (DSL) avec le protocole ppp0 et nous indiquons le débit de la bande passante Up/Down (ou montant/descendant). Qui est donné par (quelques fournisseur d'accès). Il est nécessaire de réduire au minimum la quantité du débit montant de la bande passante en Kibibit, pour cela vous devez faire des tests.

Étant donné que nous ne voulons pas de classe avec Downstream (ou débit descendant) nous indiquons

```
QOS_INTERNET_DEFAULT_DOWN='0'
```

Nous indiquons 2 pour le nombre de classe standard avec Upstream (ou débit montant). L'interface réseau affecté eth0 a un débit de 10Mibit/s

```
QOS_CLASS_N='2'

QOS_CLASS_1_PARENT='0'
QOS_CLASS_1_MINBANDWIDTH='127Kibit/s'
QOS_CLASS_1_MAXBANDWIDTH='128Kibit/s'
QOS_CLASS_1_DIRECTION='up'
QOS_CLASS_1_PRIO=''
```

Il s'agit ci-dessous de construire la classe dans laquelle nous allons insérer les paquets-ACK pour les (accusés de réception). Les paquets-ACK sont assez petites, c'est pour cette raison qu'il on besoin de peu de bande passante. Néanmoins, nous voulons en aucune façon partager ses 127Kibit. Nous laisserons donc 1Kibit/s pour le reste.

```
QOS_CLASS_2_PARENT='0'
QOS_CLASS_2_MINBANDWIDTH='1Kibit/s'
QOS_CLASS_2_MAXBANDWIDTH='128Kibit/s'
QOS_CLASS_2_DIRECTION='up'
QOS_CLASS_2_PRIO=''
```

4. Les paquetages

Dans la classe ci-dessus, on insère le reste (tout sauf les paquets-ACK). La bande passante que nous allons indiquer dans cette classe est 1Kibit/s restants, donc (128-127=1). La somme de 1Kibit/s que nous avons enregistré, n'est pas limité.

```
QOS_CLASS_2_MAXBANDWIDTH='128Kibit/s'
```

Notre première classe utilisera peut-être, à peine toute la bande passante attribuée et si il en reste un peu, alors, le reste de la B-P, passera sur la deuxième classe. Si l'on veut diviser encore davantage le Upstream (ou débit montant) (ce qui est généralement le cas), toutes les autres sous-classes "dépendent" de cette classe. Il faut, bien sûr paramétrer la variable QOS-INTERNET_DEFAULT_UP en conséquence.

```
QOS_FILTER_N='1'
```

```
QOS_FILTER_1_CLASS='1'  
QOS_FILTER_1_IP_INTERN=''  
QOS_FILTER_1_IP_EXTERN=''  
QOS_FILTER_1_PORT=''  
QOS_FILTER_1_PORT_TYPE=''  
QOS_FILTER_1_OPTION='ACK'
```

Ce filtre, filtre tous les paquets qui s'appliquent au option de filtrage, donc les paquets-ACK. Grâce à l'enregistrement de la variable QOS_FILTER_1_CLASS='1' nous somme sur de filtrer les paquets de la 1ère classe.

Pour tester, on doit rechercher au mieux une ou plusieurs bonnes sources en "envoyant/recevant des données", en sachant que l'on doit utiliser tout le débit montant/descendant et de faire "chauffer les câbles". Il faut jeter un coup d'oeil sur l'état du trafic avec Imonc (s'il est installé). Le meilleur moyen est de le faire sans activer QoS.

Le Downstream (ou débit descendant) ne devrait pas du tout chuter ou beaucoup moins fort que sans cette configuration. Comme je l'ai dit, on peut encore améliorer la situation du débit montant de la bande passante par incrémentation en Kibibit pour réduire au minimum et de vérifier les effets. Chez moi, par exemple, le meilleur débit atteint est de 121Kibit/s (en plus pas de chute avec le débit descendant). Il faut bien sûr pour chaque classe prendre en considération les valeurs de MAXBANDWIDTH et MINBANDWIDTH.

4.19. SSHD - Secure-Shell, Secure-Copy

Secure Shell offre la possibilité d'ajouter une connexion codée sur votre routeur fli4l. De plus, avec la commande Secure Copy, vous pouvez transférer des fichiers cryptés sur le routeur fli4l. Si à la connexion [vous utilisez une clé publique](#) (Page 224), vous pourrez alors exécuter des commandes sur le routeur fli4l et transférer des fichiers script qui pourront être exécutés. A partir de la version le 2.1.7 il a été rajouté le serveur SSH2.

4.19.1. Installation du service Secure-shell

OPT_SSHD Installation par défaut : OPT_SSHD='no'

Si vous voulez accéder au routeur au moyen du ssh, il faut paramétrer la variable OPT_SSHD sur 'yes'. Cela installe un serveur-ssh dropbear sur le routeur fli4l. Cela permet également de copier les fichiers sur le routeur.

SSHD_ALLOWPASSWORDLOGIN Installation par défaut :

`SSHD_ALLOWPASSWORDLOGIN='yes'`

Si vous paramétrez la variable `SSHD_ALLOWPASSWORDLOGIN` sur `'no'`, la connexion ssh, avec un mot de passe au routeur fli4l, ne sera plus possible. Alors, la connexion au routeur se fera seulement au moyen d'une clé privée et d'une clé publique (key private/public). Cela suppose qu'une [clé publique](#) (Page 224) soit installée sur le routeur.

SSHD_CREATEHOSTKEYS Installation par défaut : `SSHD_CREATEHOSTKEYS='no'`

Le serveur-ssh a besoin d'une hostKey (ou clé d'hôte) qui doit être exceptionnelle et unique pour que le serveur-ssh s'identifie clairement au client-ssh. Certe le paquetage `opt-sshd` est fourni avec une hostKey, qui permet de se connecter pour la première fois au routeur fli4l avec le client-ssh, mais cette hostKey qui a été livrée doit être remplacée le plus vite possible, la hostKey sera remplacée et connu que par vous sont même. Générer votre propre hostKey est important parce que c'est la seule manière possible de vous protéger contre les soi-disant Man-In-The-Middle-Attack. Votre client ssh peut remarquer, si un prétendu pirate est sur votre routeur fli4l, car le pirate ne connaît pas votre hostKey. Votre client ssh vous avertira par un message, si votre hostKey a été changé par le pirate.

La création de votre hostKey (ou clé d'hôte) est entièrement automatique, une fois que le paramètre la variable `SSHD_CREATEHOSTKEYS` est sur `'yes'`. Ce processus est très gourmand et peut prolonger, le temps du boot de plusieurs minutes. Si vous avez démarré le routeur fli4l avec l'activation de la variable `SSHD_CREATEHOSTKEYS`, une (ou plusieurs) hostKey(s) sera produite dans le répertoire `/tmp/ssh`. Les fichiers produits à cet endroit, doivent être copiés dans le répertoire, `etc/ssh` de votre sous-répertoire `config` (sur l'ordinateur donc vous avez créé le média de boot fli4l). Dans mon cas, voici l'arborescence du répertoire fli4l et le répertoire `config.babel` :

Faite attention, au sous-répertoire `config.babel`, vous devez avoir le répertoire, `etc` et le répertoire `ssh`. C'est précisément à cet endroit que la ou les hostKey(s) produite est copiée. A partir de la version 2.1.5 de fli4l, les fichiers de votre sous-répertoire `config`, sont prioritaire par rapport au sous-répertoire `opt`. Ainsi à la prochaine mise à jour de la version du routeur fli4l, les fichiers qui se trouvent dans le répertoire `config/etc/ssh` seront intégrés et non les fichiers qui se trouvent dans le répertoire `opt/etc/ssh`. Ainsi, il est possible pour chaque routeur fli4l d'utiliser ces propre hostKey. Lors de la construction des fichiers du routeur fli4l, un message apparaîtra à la fin, «appending config specific files to opt.img ...». Alors, tous les fichiers venant du répertoire `config` seront listés et non les fichiers du répertoire `opt`.

```
#
# appending config specific files to opt.img ...
#
etc/ssh/dropbear_dss_host_key
etc/ssh/dropbear_rsa_host_key
```

Si vous avez produit une nouvelle clé d'hôte, remettez le paramètre de la variable `SSHD_CREATEHOSTKEYS` sur `'no'`, de sorte que le script ne génère plus de nouvelle clé d'hôte à chaque démarrage du routeur fli4l.

Après une mise à jour de votre routeur fli4l, si vous vous connectez au routeur, un message d'avertissement sera indiqué (différent selon le programme) par votre client ssh,

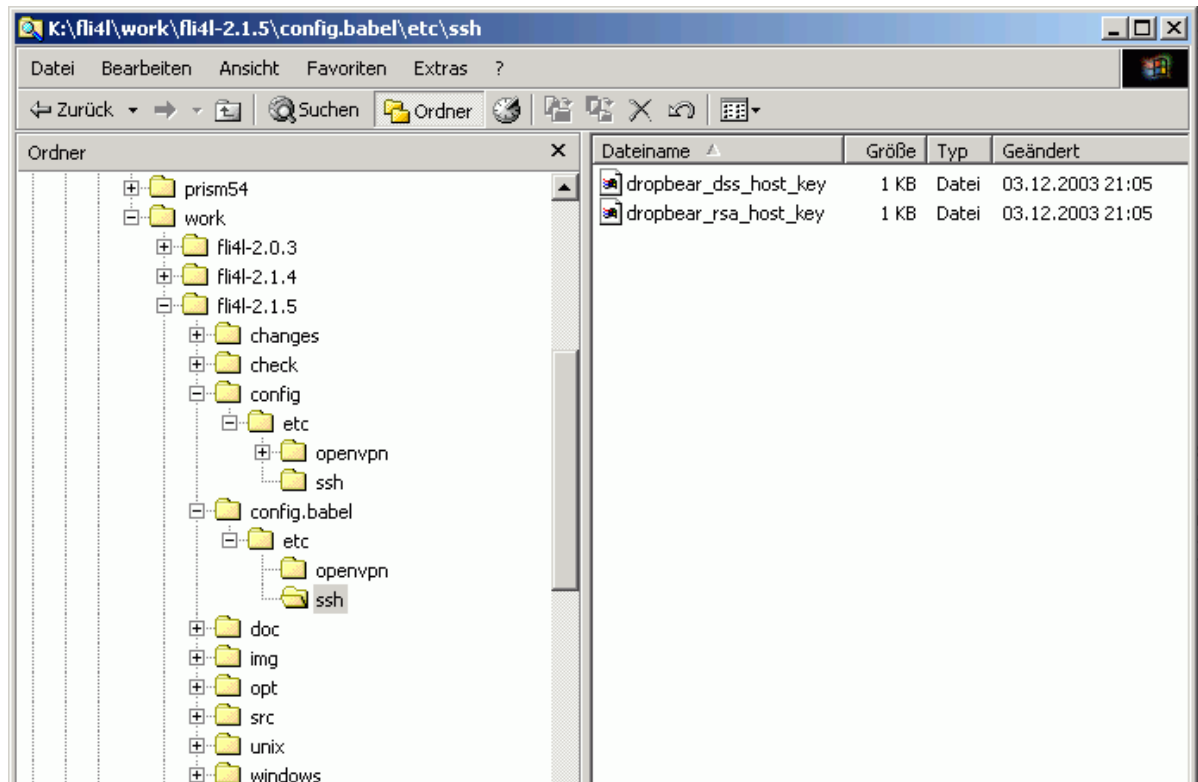


FIGURE 4.10.: Structure des répertoires fli4l

qui attire l'attention sur un changement de votre hostKey. C'est normal, puisque vous venez justement de changer la hostKey, par celle fournie par fli4l. Suivez les instructions de votre client ssh, et modifié de façon permanente votre hostKey. Si vous recevez encore une fois ce message d'avertissement à une date ultérieure, vous devriez vérifier dans tous les cas, le pourquoi de cet avertissement, qui a été émis et non pas accepter aveuglément le changement de la hostKey.

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!    @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)~!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
ca:a4:ab:e7:af:d8:68:05:d3:1f:e6:15:08:d6:ed:36.
Please contact your system administrator.
Add correct host key in /home/babel/.ssh/known_hosts to get rid of this message.
Offending key in /home/babel/.ssh/known_hosts:7
Password authentication is disabled to avoid man-in-the-middle attacks.

```

SSHD_PORT Installation par défaut : SSHD_PORT='22'

Avec la variable SSHD_PORT vous pouvez indiquer un port différent du port par défaut sur lequel le serveur ssh doit fonctionner.

Si vous voulez que l'on puisse accéder au ssh de l'extérieur, il faut paramétrer la variable [INPUT_ACCEPT_PORT_x](#) (Page 40)

Saisie des commandes pour utiliser le protocole SSH sur un ordinateur Unix ou Linux avec fli4l :

- ssh - Secure Shell
- scp - Secure Copy

Les programmes pour Windows sont aussi disponibles :

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>
<http://winscp.net/eng/docs/lang:fr>
<http://www.tectia.com/en/en.iw3>

SSHD_PUBLIC_KEY_N Installation par défaut : SSHD_PUBLIC_KEY_N='0'

Vous indiquez dans la variable SSHD_PUBLIC_KEY_N le nombre de clés publiques qui doit être copié sur le routeur fli4l.

SSH permet l'authentification, à l'aide d'une procédures de cryptage asymétrique. Le contrôle d'authentification est utilisé avec une key Public/Privat à la place du nom d'utilisateur et du mot de passe. On s'épargne ainsi d'entrée d'un mot de passe. On produit la paire de clés à l'aide de keygen-ssh (ou puttygen, si on emploi putty sous Windows avec le client ssh). Optionnel une passphrase (ou phrase confidentielle) peut être indiqué pour une signature de la clé (on a besoin de ce mot de passe, si vous voulez utiliser la clé) cela augmente encore plus la sécurité. Si vous utilisez une passphrase vous devez réfléchir à l'utilisation d'un agent de clé (voir ssh-agent ou pageant).

Important: *il y a deux clés, une clé publique et une clé privée, la clé privée, doit est traitée avec soin comme un mot de passe, puisqu'il remplit la même fonction. La clé privée*

4. Les paquetages

est installée dans le client ssh. La clé publique est installée sur le routeur fli4l. Nous avons mis à disposition les variables suivant `SSHD_PUBLIC_KEY_x` ou `SSHD_PUBLIC_KEYFILE_x` pour gérer la clé publique.

Pour de plus amples informations, voir les pages du manuel ssh et ces composants, pour la documentation de putty voir (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>).

SSHD_PUBLIC_KEY_x Dans cette variable vous pouvez indiquer la partie publique de la clé, pour l'utilisateur qui veut obtenir un accès ssh sur le routeur fli4l. Le plus simple pour récupérer la clé est d'utiliser le Cut-and-Paste (ou le couper-coller) à partir de la fenêtre du terminal. Cela pourrait par ex. ressembler à ceci :

```
SSHD_PUBLIC_KEY_1='1024 ... username@hostname'
```

Important: la clé ne contient pas de saut de ligne. Vous pouvez insérer les clés, produit par puttygen en externe, avec Cut-and-Paste (ou couper-coller). Cependant, les sauts de ligne doivent être supprimés.

SSHD_PUBLIC_KEYFILE_N Installation par défaut : `SSHD_PUBLIC_KEYFILE_N='0'`

Dans cette variable vous pouvez indiquer le nombre de fichiers Key. Au lieu de copier le contenu de la clé publique dans le fichier sshd.txt, vous pouvez copier la clé publique directement dans l'archive-opt. Cela fonctionne comme la variable `SSH_CREATEHOSTKEYS` décrit plus haut. Copiez votre clé publique dans un fichier et placez-le dans le répertoire `<config>/etc/ssh`.

SSHD_PUBLIC_KEYFILE_x Dans cette variable vous pouvez indiquer le nom du fichier de la clé publique que vous avez enregistré dans le répertoire `<config>/etc/ssh`.

```
SSHD_PUBLIC_KEYFILE_1='root@fli4l'
```

SSH_CLIENT_PRIVATE_KEYFILE_N Installation par défaut :
`SSH_CLIENT_PRIVATE_KEYFILE_N='0'`

Dans cette variable vous pouvez indiquer le nombre de fichiers Key. Si la clé privée est compatible avec le client ssh ou plink pour une connexion à un serveur SSH désiré, vous pouvez copier cette dernière dans le répertoire `<config>/etc/ssh`. Cela fonctionne comme la variable `SSH_CREATEHOSTKEYS` décrit plus haut. Si vous avez copié votre clé privée dans le répertoire `<config>/etc/ssh`. La clé privée au format OpenSSH sera convertie automatiquement à chaque processus de départ de fli4l au format dropbear.

SSH_CLIENT_PRIVATE_KEYFILE_x Dans cette variable vous pouvez indiquer le nom du fichier de la clé publique que vous avez enregistré dans le répertoire `<config>/etc/ssh`.

```
SSHD_PRIVATE_KEYFILE_1='babel@rootserver'
```

4.19.2. Installation du dbclient

OPT_SSH_CLIENT Installation par défaut : `OPT_SSH_CLIENT='no'`

Si vous voulez utiliser un authentique client ssh2, vous pouvez activer dbclient dropbear en paramétrant la variable sur `OPT_SSH_CLIENT='yes'`. Ce client a l'avantage de partager de nombreux programmes codés, avec le serveur ssh dropbear. Vous pouvez ainsi épargner de la place dans l'archive-OPT. Le dbclient est compatible avec de nombreux client ssh/scp, de plus, les paramètres de commandes sont semblables. Il y a également un lien

4. Les paquetages

symbolique créé vers /usr/bin/ssh, alors cela fonctionne habituellement avec ssh <host> ou scp <source> <target>.

Si vous voulez sauvegarder la clé d'hôte du dbclient de façon permanente, vous devez copier le fichier known_hosts du répertoire /.ssh sur le routeur fli4l, dans le répertoire config/etc/ssh. Cela se passe comme si l'on produisait une clé d'hôte. Dans l'exemple suivant le dépaquetage fli4l se trouve dans le répertoire (dans le quel le support de boot fli4l est créée) /home/babel/fli4l-3.10.4 . Tous les fichiers de configuration se trouvent dans le répertoire config.babel.

```
cd /home/babel/fli4l-3.10.4
mkdir -p config.babel/etc/ssh
scp fli4l:/.ssh/* config.babel/etc/ssh
```

4.19.3. Installation du client plink

OPT_PLINK_CLIENT Installation par défaut : OPT_PLINK_CLIENT='no'

Installer sur le routeur fli4l le client ssh1/ssh2/telnet. Le programme plink est la version Unix, du programme PuTTY connu sous Windows. Un appel de plink sur le routeur fli4l affiche la page d'aide, pour l'utilisation de plink.

Si vous voulez sauvegarder la hostKey (ou clé d'hôte) dans plink, de façon permanente, vous devez copier le fichier sshhostkeys à partir du répertoire /.putty sur le routeur fli4l, il doit être copier dans le répertoire config/etc/plink. Cela se passe comme si l'on produisait une hostKey. Dans l'exemple suivant le dépaquetage de fli4l se trouve dans le répertoire (dans le quel le support de boot fli4l est produite /home/babel/fli4l-3.10.4 . Tous les fichiers de configuration se trouvent dans le répertoire config.babel.

```
cd /home/babel/fli4l-3.10.4
mkdir -p config.babel/etc/plink
scp fli4l:/.putty/* config.babel/etc/plink
```

4.19.4. Installation d'un serveur sftp

OPT_SFTPSERVER Installation par défaut : OPT_SFTPSERVER='no'

Installe sur le routeur fli4l un serveur-sftp.

4.19.5. Littérature

Site Web de Dropbear SSH2 : <http://matt.ucc.asn.au/dropbear/dropbear.html>

Première version de la documentation par Claas Hilbrecht <babel@fli4l.de>, Avril 2004

4.20. TOOLS - Outils supplémentaires pour le débogage

Le paquetage Tools fournit un certain nombre de programmes Unix, pour l'administration et aussi pour le débogage. D'autres programmes sont intégrer comme wget, par ex. pour intercepter la première page (publicité) de certains fournisseurs d'accès. Si vous indiquez 'yes' le programme choisi sera installé dans le routeur fli4l. Le paramètre par défaut est 'no'. Voici une brève présentation des programmes, sur la façon de les utiliser, s.v.p. utiliser la commande man pour avoir plus informations sur les commandes des programmes de la distribution Unix/Linux, ou voir le site : <http://www.linuxmanpages.com>

4.20.1. Outils pour le réseau

OPT_DIG Couteau suisse pour DNS

La commande dig permet d'effectuer différentes requêtes DNS.

OPT_FTP Pour l'utilisation d'un client-FTP

Avec le programme ftp, vous pouvez mettre en place une connexion FTP en utilisant un serveur FTP et transmettre des fichiers entre le serveur FTP et le routeur.

FTP_PF_ENABLE_ACTIVE Si vous activez la variable `FTP_PF_ENABLE_ACTIVE='yes'` une règle sera ajoutée dans le filtrage de paquets pour le routeur elle permettra d'initialiser un FTP actif. Si la variable `FTP_PF_ENABLE_ACTIVE='no'` est désactivée, vous pouvez créer cette règle manuellement dans `PF_OUTPUT_%` elle sera ajoutée à la liste. Vous pouvez trouver un exemple dans la [section](#) (Page 63).

Un FTP passif est toujours possible, il n'est pas nécessaire d'utiliser cette variable ni de créer une règle de filtrage de paquets.

OPT_IFTOP Pour la surveillance du réseau

Avec la commande iftop, vous avez une liste de toutes les connexions réseaux et il affichera le débit direct de fli4l.

La commande iftop démarre, après avoir installé la commande iftop sur le routeur fli4l.

OPT_IMONC Pour la gestion du programme imond par ligne de commande

Ce programme est utilisé pour le contrôle de fli4l par ligne de commande sur le routeur, afin de gérer imond.

OPT_IPERF Pour mesurer la performance du réseau

Avec la commande iperf, vous pouvez effectuer des mesures sur la performance du réseau. En outre, la commande doit être lancée sur les deux systèmes serveur/client pour le test. Paramètre de la commande sur le serveur.

```
fli4l-server 3.10.4~# iperf -s
```

```
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
```

Ensuite pour démarrer, le serveur attend une connexion du client. Paramètre de la commande sur le client avec l'adresse-IP du serveur.

```
fli4l-client 3.10.4~# iperf -c 1.2.3.4
```

```
-----
Client connecting to 1.2.3.4, TCP port 5001
TCP window size: 16.0 KByte (default)
-----
```

```
[ 3] local 1.2.3.5 port 50311 connected with 1.2.3.4 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec   985 MBytes  826 Mbits/sec
```

Les mesures de performances démarre immédiatement et affiche les premiers résultats. iperf utilise un certain nombre d'options. Pour plus de détails sur ces options, visiter s'il vous plaît la page d'accueil du lien <http://iperf.sourceforge.net/>.

OPT_NETCAT Pour le transfert de données, basé sur un serveur TCP

OPT_NGREP Grep peut être utilisé directement sur le périphérique réseau.

OPT_NTTCP Pour tester le réseau

Avec la commande NTTCP, on peut tester la vitesse du réseau. Pour ce faire, on démarre d'un côté le serveur et de l'autre côté le client correspondant.

On lance le serveur en tapant `nttcp -i -v`. Puis, le serveur attend une demande de test du client. Maintenant pour tester la vitesse, on entre par exemple sur le client `nttcp -t <Adresse-IP du Serveur>`

Démarrer le serveur `nttcp` comme ceci :

```
fli4l-server 3.10.4~# nttcp -i -v
```

```
nttcp-1: nttcp, version 1.47
```

```
nttcp-1: running in inetd mode on port 5037 - ignoring options beside -v and -p
```

Test avec le client `nttcp` comme ceci :

```
fli4l-client 3.10.4~# nttcp -t 192.168.77.77
```

```
1~~8388608~~~~4.77~~~~0.06~~~~~14.0713~~~1118.4811~~~~2048~~~~429.42~~~34133.3
```

```
1~~8388608~~~~4.81~~~~0.28~~~~~13.9417~~~239.6745~~~~6971~~~1448.21~~~24896.4
```

Vous pouvez voir ci-dessous tous les paramètres `nttcp` :

Usage: `nttcp [local options] host [remote options]`

local/remote options are:

- t transmit data (default for local side)
- r receive data
- l# length of bufs written to network (default 4k)
- m use IP/multicasting for transmit (enforces -t -u)
- n# number of source bufs written to network (default 2048)
- u use UDP instead of TCP
- g#us gap in micro seconds between UDP packets (default 0s)
- d set SO_DEBUG in sockopt
- D don't buffer TCP writes (sets TCP_NODELAY socket option)
- w# set the send buffer space to #kilobytes, which is dependent on the system - default is 16k
- T print title line (default no)
- f give own format of what and how to print
- c compares each received buffer with expected value
- s force stream pattern for UDP transmission
- S give another initialisation for pattern generator
- p# specify another service port
- i behave as if started via inetd
- R# calculate the getpid()/s rate from # getpid() calls
- v more verbose output
- V print version number and exit
- ? print this help
- N remote number (internal use only)

default format is: %9b%8.2rt%8.2ct%12.4rbr%12.4cbr%8c%10.2rcr%10.1ccr

OPT_RTMON Pour le débogage

Si vous installez cette outil, il surveillera les changements du tableau de routage. Utilisation initial : le débogage

OPT_SOCAT Le programme "socat" est une version plus ou moins améliorée du [programme "netcat"](#) (Page 227) avec plus de fonctionnalités. En utilisant "socat" vous pouvez non seulement établir ou accepter différents types de connexions réseau, mais aussi d'envoyer des données ou lire des données avec les Sockets UNIX, les périphériques, FIFO, et ainsi de suite. Au sujet des sources et des destinations particulières, *différent* types de connexions peuvent être utilisées : l'exemple suivant serait un serveur réseau qui écoute sur un port TCP et qui écrit les données reçues dans une mémoire FIFO local ou de lire les données dans la mémoire FIFO, puis de les transmettre via le réseau à un client. Vous pouvez aller sur le site <http://www.dest-unreach.org/socat/doc/socat.html> pour avoir plus d'exemples sur les applications et sur la documentation.

OPT_TCPDUMP Pour le débogage réseau

Avec la commande tcpdump on peut observer des détails sur le trafic du réseau, et d'analyser en détail les paquets. Pour en savoir plus, faite une recherche par ex. sur Google ou avec le paramètre »tcpdump man».

tcpdump <paramètre>

OPT_WGET Client http/ftp

Avec la commande wget on peut télécharger des données sur un serveur Web avec un fichier batch de lancement, il travail en arrière-plan. Il est pratique (c'est pour cela qu'il est dans le paquetage-fli4l), on peut télécharger d'une manière simple la page web du fournisseur d'accès Internet et là placer sur son propre serveur web avec un lien. Par exemple sur le site de Freenet, Steffen Peiser a décrit les commande dans ce mini-HOWTO.

Voir : <http://www.fli4l.de/fr/aide/guide-pratique/debutant/wget-und-freenet/>

4.20.2. Outils pour la détection du matériel

En général, on ne sait jamais exactement le matériel qui est installé dans son propre routeur. Le matériel installé peut nous aider à configurer exactement le pilote de la carte réseau ou du chipset-USB. Pour nous fournir la liste des périphériques et si possible des pilotes correspondants, nous avons le choix de visualiser ces informations, soit sur la console, juste après le démarrage (recommandé pour une première installation) ou plus facilement, par l'intermédiaire de l'interface-Web de votre ordinateur. Vous pouvez voir ci-dessous un exemple des informations fournies, avec la commande :

```
fli4l 3.10.4 # cat /bootmsg.txt

#
# PCI Devices and drivers
#
Host bridge: Advanced Micro Devices [AMD] CS5536 [Geode companion] Host Bridge (rev 33)
Driver: 'unknown'
Entertainment encryption device: Advanced Micro Devices [AMD] Geode LX AES Security Block
Driver: 'geode_rng'
Ethernet controller: VIA Technologies, Inc. VT6105M [Rhine-III] (rev 96)
Driver: 'via_rhine'
Ethernet controller: VIA Technologies, Inc. VT6105M [Rhine-III] (rev 96)
Driver: 'via_rhine'
Ethernet controller: VIA Technologies, Inc. VT6105M [Rhine-III] (rev 96)
```

```
Driver: 'via_rhine'  
Ethernet controller: Atheros Communications, Inc. AR5413 802.11abg NIC (rev 01)  
Driver: 'unknown'  
ISA bridge: Advanced Micro Devices [AMD] CS5536 [Geode companion] ISA (rev 03)  
Driver: 'unknown'  
IDE interface: Advanced Micro Devices [AMD] CS5536 [Geode companion] IDE (rev 01)  
Driver: 'amd74xx'  
USB Controller: Advanced Micro Devices [AMD] CS5536 [Geode companion] OHC (rev 02)  
Driver: 'ohci_hcd'  
USB Controller: Advanced Micro Devices [AMD] CS5536 [Geode companion] EHC (rev 02)  
Driver: 'ehci_hcd'
```

Vous pouvez voir que 3 cartes réseaux identiques sont installées, gérées par le pilote 'via_rhine' et une carte wifi Atheros, gérée par le pilote madwifi (le nom n'est pas encore résolu).

OPT_HW_DETECT Cette variable s'occupe de vérifier les fichiers installés dans le routeur par rapport aux matériels identifiés. On peut alors voir le résultat sur la console après le boot, si vous mettez la variable `HW_DETECT_AT_BOOTTIME` sur 'yes' vous pouvez voir les informations sur l'Interface Web, bien entendu vous devez placer la variable [OPT_HTTPD](#) (Page 123) sur 'yes'. Sur l'interface Web, vous pourrez naturellement voir le contenu du fichier '/bootmsg.txt', si vous avez un accès réseau qui fonctionne.

HW_DETECT_AT_BOOTTIME Cette variable lance la détection du matériel lors du boot. La détection fonctionne en tâche de fond (cela prend un peu de temps), le résultat sera visible sur la console, puis sera écrit dans le fichier '/bootmsg.txt'.

OPT_LSPCI Pour lister tous les périphériques PCI

OPT_I2CTOOLS Outils pour accéder au bus I²C.

OPT_IWLEEPROM Outil pour accéder à l'EEPROM des cartes WLAN (ou cartes wifi) Intel et Atheros.

Nécessaire par exemple pour reprogrammer le domaine réglementaire de la carte ath9k (see <http://blog.asiantuntijakaveri.fi/2014/08/one-of-my-atheros-ar9280-minipcie-cards.html>).

OPT_ATH_INFO Outil pour accéder à l'EEPROM Intel et des cartes WLAN Atheros.

Cet outil peut extraire les informations détaillées du matériel utilisé et des cartes wifi Atheros, par exemple ath5k. Ceux-la comprennent, le chipset utilisé ou les données d'étalonnage.

4.20.3. Outils pour fichier

OPT_E3 Éditeur de texte pour fli4l

Il s'agit d'un éditeur de texte de très petite taille, écrit en assembleur. Vous avez à disposition différents modes d'éditeurs, comme d'autre éditeur plus ("grand"). Pour choisir l'un des mode, il suffit d'utiliser la bonne commande de E3 pour démarrer. On obtient un rapide aperçu des raccourcis clavier avec le paramètre man, si vous lancez E3 sans le paramètre man, vous pouvez appuyer sur Alt+H (sauf dans le mode-VI, en mode-CMD à la place de man il faut saisir " :h"). Notez également que le caractère (^) est représenté par la touche "Ctrl".

Commande	Mode
e3 / e3ws	WordStar, JOE
e3vi	VI, VIM
e3em	Emacs
e3pi	Pico
e3ne	NEdit

OPT_MTOOLS Avec mtools nous mettons à disposition une série de commandes (pour la copie, le formatage, etc.) similaire aux commandes DOS, ces commandes serviront à la gestion des données sur des supports DOS.

Vous trouverez dans le lien ci-dessous la documentation de mtools et les syntaxes des paramètres de commandes de chaque programme :

<http://www.gnu.org/software/mtools/manual/mtools.html>

OPT_SHRED Pour effacer un fichier

Si vous installez *shred* sur le routeur, cette commande effacera définitivement les blocs de données.

OPT_YTREE Gestionnaire de fichier

Si vous installez Ytree sur le routeur, vous aurez un gestionnaire de fichier sur votre routeur fli4l.

4.20.4. Outils pour développeur

OPT_OPENSSL Avec l'outil OpenSSL vous pouvez mesurer la vitesse des algorithmes d'encodage et le chiffrement du crypto.

openssl speed -evp des -elapsed openssl speed -evp des3 -elapsed openssl speed -evp aes128 -elapsed

OPT_STRACE Pour le débogage

Avec la commande strace, vous pouvez surveiller les appel systèmes, pour voir le déroulement des programmes.

strace <programme>

OPT_REAVER Attaque du code PIN WPS par Brute force sur le Wifi

Cette outil teste tous les codes PIN WPS possibles pour déterminer la vulnérabilité du mot de passe WPA sur votre routeur. Si vous voulez plus de détail pour l'utilisation par ligne de commande de reaver, lire la documentation sur le site <http://code.google.com/p/reaver-wps/>

OPT_VALGRIND Pour le débogage de programme

Si vous installez Valgrind sur le routeur, vous pouvez débusquer les failles dans un programme et mettre en évidence des fuites mémoires.

4.21. UMTS - Connexion UMTS via Internet

Utiliser fli4l pour la connexion Internet via l'UMTS (Universal Mobile Telecommunications System). D'autres paquetages optionnels sont nécessaires pour cette installation.

Accès de certains opérateurs de distributeur de réseau Allemand

Fournisseur	APN	Nom d'utilisateur	Mot de passe
T-Mobile	Internet.t-mobile	libre	mot de passe
Vodafone	Web.vodafone.de	libre	mot de passe
E-Plus	Internet.eplus.de	eplus	gprs
O2 (Vertragskunden)	Internet	libre	mot de passe
O2 (Prepaid-Kunden)	Pinternet.interkom.de	libre	mot de passe
Alice	Internet.partner1	libre	mot de passe

4.21.1. Configuration

OPT_UMTS Configuration standard : OPT_UMTS='no'

Indiquer 'yes' pour activer le paquetage.

UMTS_DEBUG Configuration standard : UMTS_DEBUG='no'

Pour afficher des informations de débogage supplémentaires avec pppd, vous devez définir la variable UMTS_DEBUG sur 'oui'. Dans ce cas, les informations écrites sur pppd seront utilisables dans l'interface syslog.

IMPORTANT : Pour que les informations soit délivrées le démon syslogd doit être activé avec la variable OPT_SYSLOGD (voir ci-dessus) elle doit être également paramétrée sur 'yes'.

UMTS_PIN Configuration standard : UMTS_PIN='disabled'

Pin pour carte-SIM

Vous devez indiquer un numéro à 4 chiffres ou le mot 'disabled'

UMTS_DIALOUT Configuration standard : UMTS_DIALOUT='*99***1#'

Propriétés de numérotation pour établir la connexion

UMTS_GPRS_UMTS Configuration standard : UMTS_GPRS_UMTS='both'

Dans cette variable vous indiquez le paramètre pour l'utilisation de la transmission, valeurs autorisées (both, GPRS, UMTS)

UMTS_APN Configuration standard : UMTS_APN='web.vodafone.de'

UMTS_USER Configuration standard : UMTS_USER='anonymer'

UMTS_PASSWD Configuration standard : UMTS_PASSWD='surfer'

Vous indiquez dans ces variables les données nécessaires pour la connexion.

vous devez indiquer, l'ID de l'utilisateur et le mot de passe fournie par le fournisseur. la variable UMTS_USER contient l'ID de l'utilisateur, la variable UMTS_PASSWD contient le mot de passe.

Pour certains opérateurs allemands, les fournisseurs d'accès réseaux sont les APNs (noeuds de connexion)

- <http://www.teltarif.de/mobilfunk/internet/einrichtung.html>

UMTS_NAME Configuration standard : UMTS_NAME='UMTS'

Vous indiquez dans ces variables, un nom pour le circuit - max. 15 caractères. Celui-ci sera affiché dans le client-imon d'imonc. les Espaces (blancs) ne sont pas autorisés.

UMTS_HUP_TIMEOUT Configuration standard : UMTS_TIMEOUT='600'

Vous indiquez dans cette variables, le temps en secondes, qui déterminera la fin de la connexion, si aucune communication ne circule sur la connexion UMTS. Si vous indiquez '0' vous avez aucun délai de fin de connexion.

UMTS_TIMES Configuration standard : `UMTS_TIMES='Mo-Su :00-24 :0.0 :Y'`

Déterminer les heures et jours d'activation pour ce circuit et combien cela coûte. Il est donc possible d'utiliser à des moments différents, différents circuits avec la route par défaut c'est du (Least Cost Routing). Dans ce cas, le démon imond contrôle la cession de route du circuit, si vous l'avez activé.

UMTS_CHARGEINT Configuration standard : `UMTS_CHARGEINT='60'`

Vous indiquez dans cette variables, l'intervall de charge : temps d'une unite de connexion en secondes. Elle est ensuite utilisée pour le calcul des coûts.

UMTS_USEPEERDNS Configuration standard : `UMTS_USEPEERDNS='yes'`

Vous indiquez dans cette variables, le DNS qu'utilise le fournisseur.

UMTS_FILTER Configuration standard : `UMTS_FILTER='yes'`

fli4l raccroche automatiquement après le délai spécifié dans timeout, si aucune information ne circule sur l'interface ppp0. Malheureusement, l'interface évalue également le transfert de données, qui proviennent de l'extérieur, par exemple les tentatives de connexion par un des clients P2P tels que eDonkey. Comme on est en fait en permanence contacté par les autres, il peut arriver que fli4l ne termine jamais le contact d'UMTS.

C'est là qu'intervient la variable `UMTS_FILTER`. Vous devez placer la variable sur 'yes', ce filtre estime seulement la circulation générée par votre propre machine, le trafic extérieur sera ignoré complètement. Puisque le trafic entrant "de l'extérieur" traverse habituellement le routeur et que les ordinateurs derrière celui-ci réagissent, en refusant par ex. des demandes de connexion, les paquets sortants sont aussi ignorés.

UMTS_ADAPTER (optionnelle)

Ici vous indiquez s'il s'agit d'une carte PCMCIA, un adaptateur USB ou un câble USB pour Téléphone.

Si la variable n'est pas paramétrée, seuls les fichiers nécessaires pour l'adaptateur USB sont copiés.

Valeurs admissibles : (PCMCIA, usbstick, usbphone)

Toutes les variables suivantes sont facultatives et sont nécessaire que si la détection automatique a échouée.

UMTS_IDVENDOR (optionnelle) `UMTS_IDVENDOR='xxxx'`

L'ID (ou Identification) du fabricant, après avoir allumer l'adaptateur

UMTS_IDDEVICE (optionnelle) `UMTS_IDDEVICE='xxxx'`

L'ID du produit, après avoir allumer l'adaptateur

Les informations des deux paramètres suivants sont nécessaire seulement, si l'identification change après l'initialisation.

UMTS_IDVENDOR2 (optionnelle) `UMTS_IDVENDOR2='xxxx'`

L'ID du fabricant, après l'initialisation de l'adaptateur

UMTS_IDDEVICE2 (optionnelle) `UMTS_IDDEVICE2='xxxx'`

L'ID du produit, après l'initialisation de l'adaptateur

UMTS_DRV (optionnelle) `UMTS_DRV='xxxx'`

Pilotes pour guider l'adaptateur lorsque 'usbserial' n'est pas spécifié

UMTS_SWITCH (optionnelle) `UMTS_SWITCH='-v 0x0af0 -p 0x6971 -M 555...000 -s 10'`

Paramètres pour passer en mode USB pour l'initialisation du modem. (Voir le site [usb-modeswitch](http://www.draisberghof.de/usb_modeswitch/)).

Tous les modems sur ce site Web devaient être reconnus automatiquement, à quelques exceptions près.

– http://www.draisberghof.de/usb_modeswitch/

UMTS_DEV (Optionnelle)

En cas de problèmes, l'interface de données pppd peut être indiquée ici. Pour les adaptateurs, ce sont la plupart du temps des suivants :

```
ttyUSB0 pour usbstick
ttyS2   pour pcmcia
ttyACM0 pour usbphone
```

UMTS_CTRL (optionnelle)

Certaines cartes ont des interfaces multiples à travers duquel le modem est contrôlé. Les informations sur le statut seront disponibles et lues seulement en 'offline' (ou déconnexion). Lors de la fusion des options de l'interface UMTS Quad par ex. : `ttyUSB2`

4.21.2. Exemple de configuration pour rrdtool

Selon le matériel, il est possible d'afficher sur [OPT_HTTPD](#) (Page 123) la force du signal et des erreurs de bits. Vous pouvez enregistrer l'historique de la force du signal ou du taux d'erreur binaire en utilisant `OPT_RRDTOOL`. Sur certains matériels, il n'est pas évident d'utiliser ces informations sur le statut, car ils peuvent être lus que pendant l'état offline. Comme source pour le RRD sur l'état du 'umts'.

Exemple de configuration pour rrdtool :

```
RRDTOOL_x_SOURCE='umts'
RRDTOOL_x_LABEL='UMTS Statut'
RRDTOOL_x_OPTIONS_N='2'
RRDTOOL_x_OPTIONS_1='signal'
RRDTOOL_x_OPTIONS_1_LABEL='puissance du signal'
RRDTOOL_x_OPTIONS_2='erreur'
RRDTOOL_x_OPTIONS_2_LABEL='erreurs de bit'
```

4.22. USB - Support pour périphérique USB

OPT_USB La prise en charge des périphériques USB n'est pas activée automatiquement. Vous devez placer ici 'yes', pour que les périphériques-USB puissent être utilisés. Si vous avez choisi dans le fichier `base.txt` un Périphérique-USB, vous devez indiquer impérativement 'yes'. Autrement, le périphérique ne sera pas utilisable. Si vous activez cette variable, les supports USB tels que les clés, les disques durs externes, les claviers seront activés.

Installation par défaut : `OPT_USB='no'`

USB_EXTRA_DRIVER_N Vous indiquez ici le nombre de périphérique spécifique à charger.

Installation par défaut : `USB_EXTRA_DRIVER_N='0'`

USB_EXTRA_DRIVER_x Vous indiquez ici les périphériques spécifique qui seront chargés.

Valeurs possibles en ce moment :

- printer - Prise en charge des imprimantes-USB
- belkin_sa - USB Belkin Serial converter
- cyberjack - REINER SCT cyberJack pinpad/e-com USB Chipcard Reader
- digi_acceleport - Digi AccelePort USB-2/USB-4 Serial Converter
- empeg - USB Empeg Mark I/II
- ftdi_sio - USB FTDI Serial Converter
- io_edgeport - Edgeport USB Serial
- io_ti - Edgeport USB Serial
- ipaq - USB PocketPC PDA
- ir-usb - USB IR Dongle
- keyspan - Keyspan USB to Serial Converter
- keyspan_pda - USB Keyspan PDA Converter
- kl5kusb105 - KLSI KL5KUSB105 chipset USB->Serial Converter
- kobil_sct - KOBIL USB Smart Card Terminal (experimental)
- mct_u232 - Magic Control Technology USB-RS232 converter
- omninet - USB ZyXEL omni.net LCD PLUS
- pl2303 - Prolific PL2303 USB to serial adaptor
- visor - USB HandSpring Visor / Palm OS
- whiteheat - USB ConnectTech WhiteHEAT

Installation par défaut : `USB_EXTRA_DRIVER_x=""`

USB_EXTRA_DRIVER_x_PARAM Paramètre supplémentaire pour les pilotes. Normalement, vous ne devez rien enregistrer ici.

Installation par défaut : `USB_EXTRA_DRIVER_x_PARAM=""`

USB_MODEM_WAITSECONDS Installation par défaut : `USB_MODEM_WAITSECONDS='21'`

Malheureusement, les modems USB Eagle et Speedtouch ont besoin d'une "demi-éternité" avant que ceux-ci soit prêts. Normalement pour une installation standard il suffit de 21 secondes d'initialisation. Parfois on a de la chance et l'on peut diviser cette valeur par deux avec les modems USB Eagle ou Speedtouch, il sera disponible après 10 secondes, vous pouvez indiquer alors 10 secondes. Mais si vous n'avez pas de chance, vous devez augmenter cette valeur. Malheureusement, nous ne pouvons pas vous aider, il vous faut essayer pour trouver la bonne valeur.

4.22.1. Problèmes avec les périphériques USB

Si vous avez des problèmes avec certains appareils USB. Cela peut provenir de différentes raisons, comme par exemple le pilote de périphérique ou le contrôleur de gestion USB.

Dans cette version le modem-DSL-USB Eagle, fonctionne seulement s'il est connecté à la ligne téléphonique. Si ce n'est pas le cas, le périphérique eth ne sera pas généré. Cela a pour conséquence l'impossibilité utilisable le modem. S'il vous plaît connectez le modem à la ligne téléphonique, avant de booter le routeur fli4l.

4.22.2. Précautions d'utilisation

Il faut faire attention à ce que le périphérique USB soit activé, dû côté contrôleur-hardware. Par exemple avec la carte WRAP elle est livrée sans module USB et peut être complétée par

un module USB supplémentaire. C'est pour cette raison que le périphérique USB est désactivé par défaut dans BIOS.

4.22.3. Monter les périphériques USB

Les appareils USB insérés sont reconnus certes, automatiquement, toutefois vous devez les déclarer manuellement. Par exemple à l'insertion d'une clé-USB, elle est reconnue comme un périphérique-SCSI. C'est pour cette raison que l'accès se fait via le périphérique `sd#` comme une super disquette ou via `sd#<numéro de la partition>` pour la clé-USB avec une table de partitions. Les clés-USB partitionnées sont traitées comme des disques durs, c'est-à-dire lors du branchement sur l'un des deux ports USB ils seront indiqués `sda1` et `sdb1`. En revanche pour les clés-USB ils seront indiqués `sda` ou `sdb`, donc sans l'indication du numéro de partition.

Cela permet de monter la clé-USB avec la commande suivante :

```
mount /dev/sda1 /mnt
```

Après /mnt la deuxième clé sera montée de la même façon :

```
mount /dev/sdb1 /mnt
```

Au sujet de la seconde clé-USB. Les clés sont spécifiées dans l'ordre d'insertion, donc la première clé-USB = `sda`, la deuxième clé-USB = `sdb` etc. on ne peut pas définir les ports-USB par rapport au 'nom', puisque cela dépend de l'ordre d'insertion des clés. Le démontage d'une clé-USB se fait par :

```
umount /mnt
```

Vous devez absolument éviter d'introduire simultanément de plusieurs clés-USB, dans le but de tous les monter. Je vous propose une solution, vous devez créer des sous répertoires dans le répertoire /mnt, dans lesquelles, les clés pourront alors être montées. Par exemple Cela peut être réalisé, comme indiqué ci-dessous :

```
mkdir /mnt/usba mkdir /mnt/usbb
```

Lorsque l'on monte des clés-USB ces répertoires seront alors définis comme indiqué ci-dessous :

```
mount /dev/sda1 /mnt/usba mount /dev/sdb1 /mnt/usbb
```

Ainsi, le contenu des clés-USB se trouve dans les sous répertoires /mnt/usba ou /mnt/usbb. Pour le démontage des clés-USB on utilise la commande :

```
umount /mnt/usba umount /mnt/usbb
```

S'il existe plusieurs partitions sur la clé-USB, il faudra structurer en conséquence les sous répertoires dans le répertoire /mnt.

4.23. WLAN - Supporte le WLAN (ou réseau sans fil)

Faites attention au carte-mère que vous utilisez et vérifiez les spécifications de la version du PCI il doit être au moins à 2.2. Sur des cartes-mères plus anciennes la version du PCI est seulement à 2.1, diverses erreurs viennent de là. L'ordinateur ne démarre pas du tout (il ne peut même pas être allumé), ou la carte WLAN n'est pas trouvée lors du scan PCI.

Le nom des cartes WLAN paramétré dans la variable `IP_NET_X_DEV` du fichier `base.txt` s'appelle maintenant `wlanX`. Si une carte WLAN est dans le système, elle aura le nom 'wlan0'.

4.23.1. Configuration du WLAN

OPT_WLAN Installation par défaut : `OPT_WLAN='no'`

Variable pour activer le Pack Wireless LAN indiquer ici 'yes'.

WLAN_WEBGUI Installation par défaut : `WLAN_WEBGUI='yes'`

Variable pour activer l'interface Web du Pack Wireless LAN.

WLAN_REGDOMAIN Avec cette variable, nous pouvons paramétrer et ajuster le pays spécifique. Les valeurs des codes des pays utilisent la norme ISO 3166-1 alpha-2, tels que 'FR', 'DE'. La sélection des canaux et des niveaux de puissance sont différents selon les exigences des pays.

WLAN_N On indique ici le nombre de configurations pour le WLAN, elles sont indépendantes les unes aux autres. Si vous indiquez '1' le comportement de la carte sera identique à l'ancienne version fli4l.

WLAN_x_MAC On indique ici l'adresse Mac de la carte WLAN, elle est écrite de cette manière :

`XX :XX :XX :XX :XX :XX`

Chaque X représente un Hexadécimale de l'adresse Mac de la carte, elle doit être valide pour la configuration. Si aucune adresse Mac n'est paramétrée, la configuration de l'adresse Mac de la première carte dans la variable `WLAN_1_*` sera appliquée automatiquement. Un message d'avertissement apparaîtra à la construction des archives, pour attirer votre attention. Ce message d'avertissement contient l'adresse Mac de la carte Wlan. Si vous voulez que l'interface web fonctionne correctement, vous devez enregistrer dans le fichier de configuration l'adresse Mac.

WLAN_x_MAC_OVERRIDE Avec cette variable vous pouvez changer l'adresse MAC de la carte WLAN, ainsi vous pouvez vous connecter en tant que client WLAN à travers le filtre-MAC, sans devoir régler ce filtre. Cela est utile lorsque vous voulez vous connecter au WAN, par ex. en paramétrant l'adresse Mac d'une clé USB-WLAN.

WLAN_x_ESSID Le SSID est le nom du réseau sans fil. Le "nom du réseau" peut avoir une longueur maximum de 32 caractères à la suite. Il est configuré dans l'AP (ou Point Accès) du WLAN ainsi tous les clients qui utilisent l'AP doivent s'identifier avec ce nom. Le SSID à aussi pour but, d'identifier une connexion Ad-Hoc les membres doivent avoir le même identifiant.

WLAN_x_MODE Modes à utiliser pour une carte WLAN.

Installation par défaut : `WLAN_x_MODE='ad-hoc'`

Valeurs possibles :

ad-hoc Pour un réseau Wlan sans Point d'accès
managed (ou infrastructure) Gestion du réseau sans fil avec plusieurs Point d'accès
master La carte WLAN fonctionne comme un Point d'accès

`WLAN_x_MODE='master'` fonctionne uniquement avec un pilote WLAN approprié.

WLAN_x_NOESSID Cette variable vous permet de cacher le nom ESSID à l'écran. Seulement possible qu'avec le pilote `hostap_*` et le Firmware `>= 1.6.3` et aussi avec le mode `WLAN_MODE='master'`.

Cette fonctionnalité est optionnelle et doit être ajoutée manuellement dans le fichier `config/wlan.txt`.

WLAN_x_CHANNEL On indique ici le canal de transmission du réseau.

Installation par défaut : `WLAN_x_CHANNEL='1'`

Les valeurs possibles sont : 1 à 13 et 36,40,44,48,52,56,60,64,100,104,108,112,116,120,124,128,132,136,140

4. Les paquetages

S'il vous plaît, lire la documentation de votre carte WLAN pour connaître les canaux autorisés dans votre pays. Si vous paramétrez un canal non permis, vous êtes le seul responsable. En Allemagne et en France les canaux de 1 à 13 sont permis sur la bande de fréquence 2,4 GHz : b et g. Les canaux spécifiés de 36 à 140 (voir ci-dessus) sont autorisées sur la bande de fréquence 5 GHz.

De plus la valeur '0' est permise, si cette variable est paramétrée sur `WLAN_x_MODE='managed'`. Ainsi aucun canal explicite n'est réglé, mais le wlan cherche l'AP sur tous les canaux disponibles. On peut ajouter à la valeur du canal une lettre a, b ou g (par exemple 5g) on sélectionne ainsi le mode de fonctionnement de la bande de fréquences souhaité.

Maintenant vous pouvez ajouter la lettre 'n' ou 'N' pour les cartes Wlan qui correspondes à la nouvelle norme 802.11n. Si vous indiquez la lettre en minuscule : la bande de fréquences utilisée sera de 20 MHz, si vous indiquez la lettre en majuscule : la bande de fréquences utilisée sera de 40 MHz.

Le paramétrage des majuscules a/b/g sont utilisés avec certains pilotes, (fonctionne actuellement seulement avec le pilote `ath_pci`), pour activer le turbo-WLAN affecté à la carte. Cette option est expérimentale et peut être ignorée.

WLAN_x_RATE On indique ici la vitesse de transmission du réseau.

Installation par défaut : `WLAN_x_RATE='auto'`

Les valeurs possibles sont : 1, 2, 5.5, 11, auto - Ils sont indiqué en mégabit/s, selon la carte utilisée les taux peuvent être de : 6, 9, 12, 18, 24, 36, 48 et 54. Avec certaines cartes le taux de 54 Mbit/s ne peut pas être indiqué. Vous devez alors inscrire 'auto'.

WLAN_x_RTS Permet d'activer RTS/CTS Handshake (ou échange de donnée). Cette option est utile pour de grand Wlan avec de nombreux clients, si les clients entre eux ne s'entendent pas pour émettre sur l'AP (ou Point d'accès), alors vous pouvez activer cette variable. Si cette option est activée le client à chaque processus d'émission envoie un RTS, demande d'autorisation d'émettre et obtient en retour un CTS, autorisation d'émettre par l'AP. De cette façon, chaque client écoute et ne transmet pas si un client est en train de transmettre. Ainsi, les collisions sont réduites, parce qu'on est garantie qu'il y a toujours un seul client qui transmet. Cette option est valable uniquement si la situation qui est décrite plus haut a un sens, car elle ajoute des données supplémentaires dans l'en-tête du paquet et donc réduit l'ensemble de la bande passante. Mais par la réduction des collisions, cette bande passante peut augmenter de nouveau.

Cette fonctionnalité est optionnelle et doit être ajoutée manuellement dans le fichier `config/wlan.txt`.

WLAN_x_ENC_N (obsolète) On indique ici le nombre de clé pour le cryptage du réseau sans fil.

Valeurs possibles : de 0 à 4

WLAN_x_ENC_x (obsolète) On place ici les clés pour le cryptage du réseau sans fil.

Valeurs possibles :

XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XX	128 Bit Hex-Key (X=0-F)
XXXX-XXXX-XX	64 Bit Hex-Key (X=0-F)
s :<5 Caractères>	64 Bit
s :<6-13 Caractères>	128 Bit
P :<1-64 Caractères>	128 Bit

Procédure d'attribution de Key-Hex avec l'option s : le texte pour la phrase mot de passe n'est `_pas_` compatible avec les pilotes Windows. Veuillez utiliser une Key-Hex pour Windows, la Key-Hex est utilisée généralement `_sans_` les traits d'union '-'. Procédure d'attribution de Key-Hex avec l'option P : le `<Texte>` pour la phrase mot de passe est compatible avec la plupart des pilotes WLAN de Windows (si c'est) `_uniquement_` dans le mode 128 bits. Linux permet de mélanger des longueurs de Key différentes. Mais dans la majorité des cas `_pas_` avec les pilotes WLAN de Windows !

WLAN_x_ENC_ACTIVE (obsolète) On indique dans cette variable le nombre de Key-Hex à activer pour le réseau sans fil.

Valeurs possibles : 1-4

Cette variable est à activer, si la variable `WLAN_x_ENC_N > 0` est supérieur à zéro. Cette variable est optionnelle.

WLAN_x_ENC_MODE (obsolète) On indique ici le mode de cryptage actif.

Valeurs possibles :

on/off	Avec ou sans cryptage
open	Accepte aussi les paquets non cryptés
restricted	Accepte seulement les paquets cryptés

Valeur logique : 'restricted'

Cette fonctionnalité est optionnelle et doit être ajoutée manuellement dans le fichier `config/wlan.txt`. Si la variable n'est pas dans le fichier elle est par défaut sur 'off' si aucune Key-WEP n'est défini, si le paramètre 'restricted' est activé alors il faut au moins définir une Key-Hex.

WLAN_x_WPA_KEY_MGMT Si vous voulez utiliser WPA au lieu du cryptage WEP, vous pouvez paramétrer ici le mode WPA. En ce moment, il n'y a que le mode WPA-PSK qui est supporté, la clé WPA est reconnu entre le client et le Point d'accès. Cette clé doit être choisie avec soin et ne doit pas être trop courte, sinon elle sera vulnérable contre des attaques par recherche de mot "dictionnaire".

Carte et pilote, supportant le mode-*managed* avec Wpa-Supplicant (voir http://hostap.epitest.fi/wpa_supplicant/ et le mode-*master* avec le démon Hostapd (voir <http://hostap.epitest.fi/hostapd/>).

Déjà quelques cartes ont été testées avec succès par ex. avec le chipset Atheros et le pilote hostap, ces cartes supportent (le mode managed et aussi master). Normalement il y a aussi les cartes atmel et quelques autres cartes. Il faut absolument que les développeurs d'Opt adaptent en conséquence leurs paquetages-Opt.

WLAN_x_WPA_PSK Vous indiquez ici la clé qui doit être utilisée entre la communication du client et le Point d'accès. Cette clé est enregistrée sous la forme d'une phrase mot de passe, cette (phrase) doit avoir une longueur minimum de 16 caractères et un maximum de 63 caractères. Les caractères suivants sont supportés :

a-z A-Z 0-9 ! # \$ % & () * + , - . / : ; < = > ? @ [\] ^ _ ` { | } ~

WLAN_x_WPA_TYPE Vous pouvez indiquer ici le type de cryptage, 1 pour WPA1, 2 pour WPA2 (IEEE 802.11i) et 3 pour les deux modes - le client peut alors décider s'il veut utiliser WPA1 ou WPA2. Si le matériel WLAN supporte le cryptage standard, il est préférable d'indiquer le procédé WPA2.

WLAN_x_WPA_ENCRYPTION Le protocole de cryptage TKIP et la version améliorée du CCMP (protocole Mac-AES CTR/CBC, parfois appelé AES) sont ici à votre disposition.

4. Les paquetages

Malheureusement le protocole CCMP n'est pas supporté par le matériel WLAN plus ancien. Il est également possible de spécifier les deux protocoles en même temps.

WLAN_x_WPA_DEBUG Si vous avez des problèmes avec la connexion WPA, vous pouvez définir cette variable sur 'yes', ainsi le démon compétent peut enregistrer toutes les informations dans un journal. Vous pourrez ensuite l'utiliser pour diagnostiquer les problèmes.

WLAN_x_AP Enregistré ici le nœud du Point d'accès.

Vous pouvez indiquer ici l'adresse Mac du Point d'accès. Si vous avez choisi le mode "Master" pour le WLAN, cette variable doit rester vide. Cette option est logique uniquement si `fi4l` ne trouve pas le PA par lui-même ou le Point d'accès qui est attaché. Uniquement destiné à être utilisé avec le mode "managed" pour le WLAN.

Cette fonctionnalité est optionnelle et doit être ajoutée manuellement dans le fichier `config/wlan.txt`

WLAN_x_ACL_POLICY Politique pour l'ACL (ou Liste des Contrôles d'Accès).

Installation par défaut : `WLAN_x_ACL_POLICY='allow'`

Valeurs pour lesquelles les adresses MAC sont soumises :

- `deny` Les adresses Mac de la liste ne peuvent pas se connecter
- `allow` Les adresses Mac de la liste peuvent se connecter
- `open` Toutes les adresses Mac reçoivent indépendamment un filtrage d'accès

Malheureusement `WLAN_ACL` est actuellement pris en charge uniquement par le pilote `hostap_*`. Comme alternative, vous avez la possibilité de paramétrer les options de contrôles d'accès dans le firewall, car des progrès significatifs ont été réalisés dans la version 3.0.x

WLAN_x_ACL_MAC_N AP-ACLs - Restriction des stations WLAN autorisées.

Installation par défaut : `WLAN_x_ACL_MAC_N='0'`

En indiquant un chiffre supérieur à '0' on active la liste de contrôle d'accès (filtrage les adresses Mac) et indique le nombre d'entrées ACL. La liste de contrôle d'accès est une liste d'adresses MAC, qui autorise ou interdit l'accès au point d'accès. Le nombre définit les adresses Mac qui peut être activées.

WLAN_x_ACL_MAC_x On indique ici les adresses Mac sous la forme : `00:00:E8:83:72:92`

WLAN_x_DIVERSITY Cette variable vous permet, d'activer la diversité des antennes manuellement.

Installation par défaut : `WLAN_x_DIVERSITY='no'` (choix automatique)

WLAN_x_DIVERSITY_RX Ici vous sélectionnez l'antenne de réception.

Installation par défaut : `WLAN_x_DIVERSITY_RX='1'`

- 0 = Sélection automatique
- 1 = Antenne 1
- 2 = Antenne 2

WLAN_x_DIVERSITY_TX Ici vous sélectionnez l'antenne de transmission.

Installation par défaut : `WLAN_x_DIVERSITY_TX='1'`

WLAN_x_WPS Avec cette variable vous pouvez activer le support WPS (WiFi Protected Setup). Dans le `WLAN_WEBGUI` il sera alors affiché un bouton de commande et un code PIN. Vous pouvez aussi contrôler le WPS par ligne de commande.

Installation par défaut : `WLAN_x_WPS='no'`

WLAN_x_PSKFILE Si vous activez PSKFILE, vous allez en plus de la configuration de WLAN_x_WPA_PSK pouvoir utiliser une clé pré-partagée pour se connecter à d'autres clients. Actuellement la configuration se fait par la fonction WLAN_x_WPS, qui utilise un fichier pour fournir les clés individuelle aux clients.

Si le fichier est désactivé et si la connexion des clients-WPS sont reliés à ce fichier, ils ne seront plus en mesure de se connecter au Point d'Accès.

Les clients-WPS qui se connectent après la désactivation de ce fichier ne seront pas affectés.

Installation par défaut : WLAN_x_PSKFILE='yes'

WLAN_x_BRIDGE Dans cette variable vous pouvez indiquer le bridge qui sera rattaché au Wlan, alternativement avec le paquetage ADVANCED_NETWORKING.

Exemple : WLAN_x_BRIDGE='br0'

Attention : soit vous indiquez la valeur ici soit dans Advanced-Network ! Mais pas dans les deux fichier de configuration !

4.23.2. Exemple

Connexion à un point d'accès via WPA

```
OPT_WLAN='yes'
WLAN_N='1'
WLAN_1_MAC='00:0F:A3:xx:xx:xx'
WLAN_1_ESSID='foo'
WLAN_1_MODE='managed'           # liaison au Point d'accès
WLAN_1_CHANNEL='1'
WLAN_1_RATE='auto'
#
# Configuration WPA
#
WLAN_1_ENC_N='0'                # Key WEP
WLAN_1_WPA_KEY_MGMT='WPA-PSK'   # WPA pre shared key
WLAN_1_WPA_TYPE='1'             # WPA 1
WLAN_1_WPA_ENCRYPTION='TKIP'
WLAN_1_WPA_PSK='phrase Mot de Passe solide entre (16-63 caractères)'
#
# Dans un contexte WPA insignifiant
#
WLAN_1_ENC_N='0'
WLAN_1_ENC_ACTIVE='1'
WLAN_1_ACL_POLICY='allow'
WLAN_1_ACL_MAC_N='0'
```

Connexion à un point d'accès avec le cryptage WPA2

```
OPT_WLAN='yes'
WLAN_N='1'
WLAN_1_MAC='00:0F:A3:xx:xx:xx'
WLAN_1_ESSID='foo'
WLAN_1_MODE='master'           # Point d'accès
WLAN_1_CHANNEL='1g'            # Channel 1, Mode sur 'g'
```

4. Les paquetages

```
# Carte-Atheros

WLAN_1_RATE='auto'
#
# Configuration WPA
#
WLAN_1_ENC_N='0'           # Key WEP
WLAN_1_WPA_KEY_MGMT='WPA-PSK' # WPA pre shared key
WLAN_1_WPA_TYPE='2'        # WPA 2
WLAN_1_WPA_ENCRYPTION='CCMP'
WLAN_1_WPA_PSK='Pass-phrase solide (16-63 caractères)'
#
# Contrôle d'accès des adresses MAC basé sur le AP
#
WLAN_1_ACL_POLICY='allow'
WLAN_1_ACL_MAC_N='0'
#
# Dans un contexte WPA insignifiant
#
WLAN_1_ENC_ACTIVE='1'
```

Connexion à un point d'accès avec le cryptage WEP

```
OPT_WLAN='yes'
WLAN_N='1'
WLAN_1_MAC='00:0F:A3:xx:xx:xx'
WLAN_1_ESSID='foo'
WLAN_1_MODE='master'      # Point d'accès
WLAN_1_CHANNEL='1'
WLAN_1_RATE='auto'
#
# Configuration WEP
#
WLAN_1_WPA_KEY_MGMT=''    # Key WPA
WLAN_1_ENC_N='4'          # 4 WEP-Keys
WLAN_1_ENC_1='...'
WLAN_1_ENC_2='...'
WLAN_1_ENC_3='...'
WLAN_1_ENC_4='...'
WLAN_1_ENC_ACTIVE='1'    # première clé actif
#
# Contrôle d'accès des adresses MAC basé sur le Point d'accès
#
WLAN_1_ACL_POLICY='allow'
WLAN_1_ACL_MAC_N='0'
#
# Configuration WEP insignifiant
#
WLAN_1_WPA_TYPE='2'
WLAN_1_WPA_ENCRYPTION='CCMP'
WLAN_1_WPA_PSK='...'
```

4.23.3. Point d'accès virtuel (VAP) (Expérimental)

Certaines cartes WLAN avec les (pilotes : ath_pci, ath5k, ath9k, ath9k_htc) peuvent distribuer jusqu'à 4 cartes WLAN virtuelles (VAP).

La configuration du WLAN pour un point d'accès virtuel (VAP) doit avoir les conditions suivantes : avoir le même canal et la même adresse MAC. Au moyen de l'adresse MAC utilisée plusieurs fois, la carte sera fractionnée et identifiée. Si vous avez plusieurs cartes, cette opération peut être faite plusieurs fois.

Le périphérique de base s'appellera wlan0 (pour la carte WLAN). Pour le VAP wlan0v2 etc... Si vous utilisez un bridge pour faire un lien, indiquer S.V.P. WLAN_x_BRIDGE='br0' etc...

Actuellement la configuration maximum pour le VAP est de 8x Master en fonction de la carte et du pilote.

4.23.4. Réglage de l'heure pour l'arrêt du WLAN avec easycron

Au moyen du paquetage *easycron* (Page 116), vous pouvez arrêter et redémarrer votre carte WLAN à une heure précise.

```
EASYCRON_N='2'
EASYCRON_1_CUSTOM = ''      # Arrêt tous les soirs à 24 heures.
EASYCRON_1_COMMAND = '/usr/sbin/wlanconfig.sh wlan0 down'
EASYCRON_1_TIME    = '* 24 * * *'

EASYCRON_2_CUSTOM = ''      # Reprise le matin à 8 heures.
EASYCRON_2_COMMAND = '/usr/sbin/wlanconfig.sh wlan0'
EASYCRON_2_TIME    = '* 8 * * *'
```

4.23.5. Remarque et dons

Les cartes WLAN avec le chipset RT25xx peuvent être utilisées avec fli4l dans les modes ad-hoc et managed, grâce à la donation généreuse de 2 cartes WLAN Ralink 2500. Le pilote a pu être ajouté dans le fichier base.txt sous le nom rt2500. Ces cartes ont été données par :

Computer Contor, Pilgrimstein 24a, 35037 Marburg, Allemagne marklabelbuildroot

4.24. SRC - Le Buildroot fli4l

Ce chapitre est intéressant, uniquement pour les développeurs qui veulent compiler des programmes binaires ou le Kernel Linux pour fli4l. Si vous utilisez fli4l seulement comme routeur et si vous ne voulez pas donner au routeur fli4l des opt-packages (ou paquetage optionnel) ou créer vos propre programmes binaires, vous pouvez ignorer complètement ce chapitre.

En général, un système Linux est nécessaire pour la création de progiciels pour fli4l. La création sur d'autres systèmes d'exploitation (Microsoft Windows, OS X, FreeBSD etc.) ne sera *pas* support.

Les exigences du système Linux pour le développement de fli4l sont les suivantes :

- GNU gcc et g++ dans la version 2.95 ou supérieur
- GNU gcc-multilib (en fonction de votre système)
- GNU binutils (y compris Binder ainsi que d'autres programmes sont nécessaires)
- GNU make dans la Version 3.81 ou supérieur

- GNU bash
- libncurses5-dev pour **fbr-make *-menuconfig** (en fonction de votre système)
- Les programmes sed, awk, which, flex, bison et patch
- Les programmes makeinfo (paquet texinfo) et msgfmt (paquet gettext)
- Les programmes tar, cpio, gzip, bzip2 et unzip
- Les programmes wget, rsync, svn et git
- Les programmes perl et python

Dans la documentation ci-dessous les caractères en **gras** indiquent une commande, le caractère ↵ représente la touche Entrée de votre clavier et ferme la commande.

4.24.1. Vue d'ensemble des répertoires sources

Dans le répertoire **src**, vous trouverez les sous-répertoires suivants :

Répertoire	Contenu
fbr	Ce répertoire contient le système de construction personnalisable qui est basé sur le buildroot uClibc (avec actuellement la version 0.9.33.2). Le FBR signifie "fli4l Buildroot". Il est ainsi possible de recompiler tous les programmes fli4l (le Kernel, les bibliothèques et les applications).
fli4l	Ce répertoire contient les sources spécifiques à fli4l, triées par paquets. Toutes les sources incluses dans les sous-répertoires ont été écrites spécifiquement pour l'utilisation de fli4l ou du moins fortement personnalisées.
cross	Ce répertoire contient les scripts nécessaires pour la compilation croisée, ils permettent de créer et de compiler le mkfli4l avec les différentes plates-formes.

4.24.2. Compiler un programme pour fli4l

Dans le répertoire "fbr" vous avez le script **fbr-make** qui contrôle la compilation de tous les programmes du paquet base du routeur fli4l. Ce script se charge de télécharger et de compiler tous les fichiers binaires requis pour fli4l. En règle générale, les fichiers script définient sont placés dans le répertoire `~/fbr` s'il n'existe pas encore, il sera créé. (Ce répertoire peut être modifié à l'aide de la variable d'environnement **FBR_BASEDIR**, voir ci-dessous.)

Précision : la quantité d'espace nécessaire pour le processus de compilation est (actuellement environ de 900 Mio pour les archives téléchargées et à peu près de 30 Gio avec les résultats intermédiaires en réalisant une compilation). Assurez-vous que dans le répertoire `~/fbr` vous avez suffisamment d'espace ! (Sinon, vous pouvez également utiliser l'option **FBR_TIDY**, voir ci-dessous).

La structure du répertoire `~/fbr` est la suivante :

4. Les paquetages

Répertoire	Contenu
fbr-<code><branch></code>-<code><arch></code>	Dans ce répertoire, le buildroot uClibc sera décompacté. <code><branch></code> est la branche de développement (par exemple <code>trunk</code>), à partir du quelle le FBR provient. A l'origine le FBR est un paquet <code>src</code> décompacté, qui était utilisé pour personnalisé le <code>fbr</code> . <code><arch></code> est l'architecture de processeur (par exemple <code>x86</code> ou <code>x86_64</code>). En plus de ce répertoire.
dl	Les archives téléchargées sont stockées ici.
own	Les paquets FBR peuvent être stockées ici, ils seront également compilés.

Ci-dessous le répertoire Buildroot `~/fbr/fbr-<branch>-<arch>/buildroot` les répertoires suivants sont intéressants :

Répertoire	Contenu
output/sandbox	Dans ce répertoire il y aura un sous-répertoire pour chaque paquet FBR, celui-ci contiendra les fichiers du paquet FBR après le processus de compilation. Dans le répertoire <code>output/sandbox/<code><paquet></code>/target</code> seront placé les fichiers qui sont prévus pour le routeur fli4l. Dans le répertoire <code>output/sandbox/<code><paquet></code>/staging</code> seront placé les fichiers qui sont nécessaires pour convertir <i>d'autre</i> paquet FBR qui ont besoin du paquet FBR principale.
output/target	Dans ce répertoire, <i>tous</i> les programmes stockés seront compilés pour le routeur fli4l. Cette liste reflète la structure des répertoires sur le routeur fli4l. Avec l'aide de la commande <code>chroot</code> vous pouvez changer ce répertoire et tester les programmes compilés ¹⁵

Paramètres généraux

Pour l'utilisation de la commande `fbr-make` vous pouvez affecter plusieurs variables d'environnement :

Variable	Description
FBR	Indique explicitement le chemin d'accès au FBR. L'utilisation du chemin par défaut est <code>~/fbr/fbr-<code><branch></code>-<code><arch></code></code> (voir ci-dessus).
FBR_BASEDIR	Indique explicitement le chemin d'accès au FBR. Par défaut, le chemin <code>~/fbr</code> (voir ci-dessus) est utilisé. Cette variable est ignorée si la variable d'environnement <code>FBR</code> est configurée.
FBR_DLDIR	Indique le répertoire qui contient les archives sources. L'utilisation du chemin par défaut est <code>\${FBR}/../dl</code> (par exemple <code>~/fbr/dl</code>).

15. cette fonction est soumis à certaines conditions, se référer au paragraphe "[tester un programme compilé](#)" (Page 247).

4. Les paquetages

Variable	Description
FBR_OWNDIR	Indique le répertoire qui contient les paquets spécifiques. L'utilisation du chemin par défaut est <code>\${FBR}/../own</code> (par exemple <code>~/fbr/own</code>).
FBR_TIDY	Si cette variable contient la valeur "y", les résultats intermédiaires qui apparaissent pendant la compilation des paquets FBR seront effacés directement après l'installation du répertoire <code>output/target</code> . Cela permet d'économiser beaucoup d'espace, en fait cette valeur est toujours recommandé, si vous ne vous sentez pas l'envie de vérifier les répertoires <code>output/build/...</code> après la construction des paquetages. Si cette variable contient la valeur "k", c'est seulement les résultats intermédiaires dans les différents répertoires du Kernel de Linux qui seront effacés, cela permet d'économiser relativement beaucoup d'espace sans perdre les fonctionnalités. Tous les autres déplacements (ou si la variable est manquante) sont effectués et tous les résultats intermédiaires seront conservés.
FBR_ARCH	Cette variable spécifie l'architecture du processeur pour lequel le FBR (ou les paquets FBR individuels) doit être construit. Si elle est absente le <code>x86</code> sera utilisé pour la construction. Voir ci-dessous les architectures supportées.

Le FBR prend actuellement en charge les architectures suivantes :

Architecture	Description
x86	Intel Architecture x86 (32-Bit), aussi connu sous le nom IA-32.
x86_64	AMD Architecture x86-64 (64-Bit), appelé aussi Intel 64 ou EM64T.

Compiler tous les paquets FBR

Lorsque vous exécutez la commande `fbr-make` avec l'argument `world`, pour que toutes les archives sources soit téléchargées et compilées, l'ordinateur devra être utilisé pendant plusieurs heures et ceci en fonction de l'ordinateur et le type de connexion Internet.¹⁶

Compiler avec Toolchain

Lorsque vous exécutez la commande `fbr-make` avec l'argument `toolchain`, tous les paquets FBR sont téléchargés et convertis, ils sont nécessaires pour construire les fichiers binaires réels pour fli4l (c-à-d construire Binder, la bibliothèque uClibc etc.). Normalement, cette commande n'est pas nécessaire car tous les paquets FBR dépendante du toolchain, les programmes toolchain téléchargés et sont de toute façon compilés.

16. Le téléchargement des archives source sera bien entendu effectuée qu'une seule fois, tant que vous ne mettez pas à jour le FBR, si vous le mettez à jour vous aurez besoin de nouvelles versions de paquets avec d'autres archives source.

Compiler un unique paquet FBR

Si vous voulez compiler seulement un certain paquet FBR (pour auto-développé un programme OPT), vous pouvez indiquer le nom du paquet FBR ou le nom de plusieurs paquets FBR avec la commande **fbr-make**, (vous pouvez indiquer **fbr-make openvpn** pour télécharger et compiler le programme openVPN). Toutes les fichiers nécessaires qui dépende du programme seront également téléchargés et compilés.

Recompiler un unique paquet FBR

Si vous voulez recompiler un certain paquet FBR (pour une raison quelconque), vous devez d'abord supprimer les informations du processus de compilation dans le FBR avant de recommencer. Vous pouvez utiliser la commande **fbr-make <paquet>-clean** (par ex. **fbr-make openvpn-clean**). Les informations de tous les paquets FBR qui dépendent du paquet FBR spécifié seront également réinitialisées, de sorte qu'ils pourront également être recompilés la prochaine fois avec **fbr-make world**.

Recompiler tous les paquets FBR

Si vous souhaitez recompiler complètement le FBR (par exemple parce que vous voulez l'utiliser un programme de référence pour développer votre nouveau système haut de gamme ;-). Vous pouvez tous supprimer en utilisant la commande **fbr-make clean** après avoir été invité à confirmé la commande, tous les artefacts qui ont été générés au cours de la dernière construction FBR seront supprimés.¹⁷ Cela est également utile pour libérer de l'espace disque.

4.24.3. Tester d'un programme compilé

Si un programme a été compilé avec **fbr-make**, il peut également être testé sur l'ordinateur de développement. Un tel test ne fonctionne que lorsque l'architecture du processeur de l'ordinateur de développement et l'architecture du processeur pour fli4l pour laquelle les programmes ont être compilés, correspondent. (Par exemple, il n'est pas possible d'exécuter les programmes fli4l x86_64 sur un système d'exploitation x86). Si la condition est remplie, vous pouvez faire un teste,

```
chroot ~/.fbr/fbr-<branch>-<arch>/buildroot/output/target /bin/sh
```

allez dans le répertoire cible fli4l et essayez directement avec le/les programme(s) compilé(s). S'il vous plaît faites attention, vous avez besoin pour utiliser **chroot** des droits administrateur et en fonction de vos préférences et de la configuration du système vous devez utiliser le service **sudo** ou **su** c'est une exigence! Vous devez aussi, avoir compilé dans le paquet FBR **busybox** (via **fbr-make busybox**) de sorte de pouvoir disposer d'un environnement shell dans le répertoire **chroot**. Voici un petit exemple :

```
$ sudo chroot ~/.fbr/fbr-trunk-x86/buildroot/output/target /bin/sh
Passwort:(Ihr Passwort)
```

```
BusyBox v1.22.1 (fli4l) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

17. Tout les répertoires `~/.fbr/fbr-<branch>-<arch>/buildroot/output` seront supprimés.

```
# ls
THIS_IS_NOT_YOUR_ROOT_FILESYSTEM  mnt
bin                                opt
dev                                proc
etc                                root
home                               run
img                                sbin
include                            share
lib                                sys
lib32                              tmp
libexec                            usr
man                                var
media                              windows
# bc --version
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free Software Foundation, Inc.
# echo "42 - 23" | bc
19
#
```

4.24.4. Déboguer un programme compilé

Si vous avez un problème sur un programme `fli4l`, en d'autres termes : s'il se bloque, vous avez la possibilité d'analyser l'état du programme plus tard, juste avant l'accident (aussi nommé "débogage post-mortem"). Pour cela il est d'abord nécessaire de configurer le paquet `base` et d'activer `DEBUG_ENABLE_CORE='yes'`. Lors du crash un vidage de la mémoire est généré dans `/var/log/dumps/core.<PID>`, le "PID" est le numéro de processus accidenté, on peut analyser de la manière suivante l'état du programme sur un ordinateur Linux avec un FBR complètement compilé. Dans l'exemple suivant, le programme à analyser est `/usr/sbin/collectd`, avec le SIGBUS accepté. Le vidage de la mémoire est placé dans le fichier `/tmp/core.collectd`.

```
fli4l@eisler:~$ .fbr/fbr-trunk-86/buildroot/output/host/usr/bin/i586-linux-gdb
GNU gdb (GDB) 7.5.1
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=x86_64-unknown-linux-gnu --target=i586-buildr
oot-linux-uclibc".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
(gdb) set sysroot /project/fli4l/.fbr/fbr-trunk-86/buildroot/output/target
(gdb) set debug-file-directory /project/fli4l/.fbr/fbr-trunk-86/buildroot/output/de
t/debug
(gdb) file /project/fli4l/.fbr/fbr-trunk-86/buildroot/output/target/usr/sbin/collec
llectd
Reading symbols from /project/fli4l/.fbr/fbr-trunk-86/buildroot/output/target/usr/
sbin/collectd...Reading symbols from /project/fli4l/.fbr/fbr-trunk-86/buildroot/outp
ut/debug/.build-id/8b/28ab573be4a2302e1117964edede2e54ebdbbf.debug...done.
done.
```


4. Les paquetages

```
(gdb) core /tmp/core.collected
[New LWP 2250]
[New LWP 2252]
[New LWP 2259]
[New LWP 2257]
[New LWP 2255]
[New LWP 2232]
[New LWP 2235]
[New LWP 2238]
[New LWP 2242]
[New LWP 2244]
[New LWP 2245]
[New LWP 2231]
[New LWP 2243]
[New LWP 2251]
[New LWP 2248]
[New LWP 2239]
[New LWP 2229]
[New LWP 2249]
[New LWP 2230]
[New LWP 2247]
[New LWP 2233]
[New LWP 2256]
[New LWP 2236]
[New LWP 2246]
[New LWP 2240]
[New LWP 2241]
[New LWP 2237]
[New LWP 2234]
[New LWP 2253]
[New LWP 2254]
[New LWP 2258]
[New LWP 2260]
Failed to read a valid object file image from memory.
Core was generated by `collected -f'.
Program terminated with signal 7, Bus error.
#0  0xb7705f5d in memcpy ()
    from /project/fli4l/.fbr/fbr-trunk-86/buildroot/output/target/lib/libc.so.0
(gdb) backtrace
#0  0xb7705f5d in memcpy ()
    from /project/fli4l/.fbr/fbr-trunk-86/buildroot/output/target/lib/libc.so.0
#1  0xb768a251 in rrd_write (rrd_file=rrd_file@entry=0x808e930, buf=0x808e268,
    count=count@entry=112) at rrd_open.c:716
#2  0xb76834f3 in rrd_create_fn (
    file_name=file_name@entry=0x808d2f8 "/data/rrdtool/db/vm-fli4l-1/cpu-0/cpu-interrupt.rrd.async", rrd=rrd@entry=0xacff2f4c) at rrd_create.c:727
#3  0xb7683d7b in rrd_create_r (
    filename=filename@entry=0x808d2f8 "/data/rrdtool/db/vm-fli4l-1/cpu-0/cpu-interrupt.rrd.async", pdp_step=pdp_step@entry=10, last_up=last_up@entry=1386052459,
    argc=argc@entry=16, argv=argv@entry=0x808cf18) at rrd_create.c:580
#4  0xb76b77fd in srrd_create (
    filename=0xacff33f0 "/data/rrdtool/db/vm-fli4l-1/cpu-0/cpu-interrupt.rrd.async",
```

4. Les paquetages

```

    pdp_step=10, last_up=1386052459, argc=16, argv=0x808cf18) at utils_rrdcreate
.c:377
#5  0xb76b78cb in srrd_create_thread (targs=targs@entry=0x808bab8)
    at utils_rrdcreate.c:559
#6  0xb76b7a8f in srrd_create_thread (targs=0x808bab8) at utils_rrdcreate.c:491
#7  0xb7763430 in ?? ()
    from /project/fli4l/.fbr/fbr-trunk-86/buildroot/output/target/lib/libpthread.so.
0
#8  0xb775e672 in clone ()
    from /project/fli4l/.fbr/fbr-trunk-86/buildroot/output/target/lib/libpthread.so.
0
(gdb) frame 1↵
#1  0xb768a251 in rrd_write (rrd_file=rrd_file@entry=0x808e930, buf=0x808e268,
    count=count@entry=112) at rrd_open.c:716
716     memcpy(rrd_simple_file->file_start + rrd_file->pos, buf, count);
(gdb) print (char*) buf↵
$1 = 0x808e268 "RRD"
(gdb) print rrd_simple_file->file_start↵
value has been optimized out
(gdb) list↵
711     if((rrd_file->pos + count) > old_size)
712     {
713         rrd_set_error("attempting to write beyond end of file");
714         return -1;
715     }
716     memcpy(rrd_simple_file->file_start + rrd_file->pos, buf, count);
717     rrd_file->pos += count;
718     return count;        /* mimmic write() semantics */
719 #else
720     ssize_t _sz = write(rrd_simple_file->fd, buf, count);
(gdb) list 700↵
695     * rrd_file->pos of rrd_simple_file->fd.
696     * Returns the number of bytes written or <0 on error. */
697
698     ssize_t rrd_write(
699         rrd_file_t *rrd_file,
700         const void *buf,
701         size_t count)
702     {
703         rrd_simple_file_t *rrd_simple_file = (rrd_simple_file_t *)rrd_file->
pvt;
704     #ifdef HAVE_MMAP
(gdb) print *(rrd_simple_file_t *)rrd_file->pvt↵
$2 = {fd = 9, file_start = 0xa67d0000 <Address 0xa67d0000 out of bounds>,
    mm_prot = 3, mm_flags = 1}

```

Vous pouvez voir ci-dessus, il faut un peu "fouiller" que dans le répertoire d'objet `rrd_simple_file_t` le pointeur est invalide ("Address ... out of bounds") dans cette suite de débogage, il est clair que l'échec de `posix_fallocate` est la cause de l'interruption du programme.

Ce qui est important ici, c'est que *tous* les chemins indiquer sont pleinement qualifié (`/project/...`) et qu'il n'y a aucun "raccourcis" utilisé (par exemple `~/...`). Si

cela n'est pas respecté, il peut arriver que les informations de débogage `gdb` ne trouve pas l'application et/ou n'utilise pas les bibliothèques. Les informations de débogage ne sont pas directement incluses dans le programme testé, mais stocké dans le répertoire `~/fbr/fbr-<branch>-<arch>/buildroot/output/debug/` sur fichier séparé.

4.24.5. Information sur le FBR

Accéder à l'aide

Que peut faire `fbr-make` pour vous, vous pouvez peut-être utiliser la commande `fbr-make help`.

Affichage des informations sur le programme

Vous pouvez voir tous les paquets FBR disponibles et leurs versions, en utilisant la commande `fbr-make show-versions` :

```
$ fbr-make show-versions↵
Configured packages

acpid 2.0.20
actctrl 3.25+dfsg1
add-days undefined
[...]
```

Affichage des associations de bibliothèques

Avec la commande `fbr-make links-against <soname>` et en indiquant le nom de la bibliothèque à la place de `soname`, vous pourrez voir tous les fichiers dans le répertoire `~/fbr/fbr-<branch>-<arch>/buildroot/output/target` qui sont associés à cette bibliothèque. Par exemple pour identifier tous les programmes et bibliothèques qui utilisent `libm` (bibliothèque de fonction mathématique), vous indiquez la commande `fbr-make links-against libm.so.0` car le nom de la bibliothèque `libm.so.0` est `libm`-bibliothèque. Un autre exemple :

```
$ fbr-make links-against librrd_th.so.4↵
Executing plugin links-against
Files linking against librrd_th.so.4
collectd usr/lib/collectd/rrdcached.so
collectd usr/lib/collectd/rrdtool.so
rrdtool usr/bin/rrdcached
```

Dans la première colonne le nom du paquet est indiqué et dans la seconde le chemin (relatif) du fichier qui est associé à la bibliothèque.

Pour trouver le nom d'une bibliothèque, vous pouvez utiliser `readelf`, comme ceci :

```
$ readelf -d ~/fbr/fbr-trunk-x86/buildroot/output/target/lib/libm-0.9.33.2.so |↵
> grep SONAME↵
0x0000000e (SONAME)                Library soname: [libm.so.0]
```

Affichage des changements de version

(Uniquement) intéressant pour les développeurs de l'équipe fli4l avec un accès en écriture au répertoire fli4l-SVN-Repository, utilisez la commande `fbr-make version-changes`. Vous pourrez voir la liste de tous les paquets dont la version FBR a été modifiée localement, la version différente de la copie de travail et la version du référentiel. Ainsi, le développeur peut voir un aperçu des paquets FBR mis à jour, avant d'écrire les changements dans le repo. Voici une entrée possible :

```
$ fbr-make version-changes↵
Executing plugin version-changes
Package version changes
KAMAILIO: 4.0.5 --> 4.1.1
```

Ici vous pouvez voir immédiatement que le paquet FBR `kamailio` avait la version 4.0.5 et a été mise à jour avec la version 4.1.1.

4.24.6. Modification de la configuration du FBR

Reconfiguration des FBRs

D'une part, à l'aide de la commande `fbr-make buildroot-menuconfig`, il est possible de choisir les paquets FBR à compiler. Ceci est utile si vous voulez compiler d'autres paquets FBR pour fli4l et qui ne sont pas activés par défaut, mais qui sont intégrés dans le buildroot uClibc, ou si vous souhaitez activer vos propres paquets FBR. D'autre part, des propriétés globales du FBRs peuvent être modifiées, telles que la version du compilateur GCC utilisée. Pour couronnée de succès la configuration du menu, la nouvelle configuration doit être sauvegardée dans le répertoire `src/fbr/buildroot/.config`.

S'il vous plaît noter, que des modifications dans la configuration du toolchain ne sont officiellement *pas* prisent en charge, car les fichiers binaires auront une forte probabilité d'incompatibilités avec la distribution officielle de fli4l. Donc, si vous avez besoin de fichiers binaires pour votre propre OPT et que vous souhaitez publier cette OPT, vous ne devez pas modifier un paramètre de chaîne de compilation !

Reconfiguration de la bibliothèque uClibc

Si vous utilisez de commande `fbr-make uclibc-menuconfig` vous pouvez modifier la fonctionnalité de la bibliothèque uClibc. Pour couronnée de succès la configuration du menu, la nouvelle configuration doit être enregistrée dans le répertoire `fbr-make uclibc-menuconfig`.

Il faut prendre également en compte comme dans le dernier paragraphe : une modifications sera hautement probable qu'elle ne soit pas compatible avec la distribution officielle de fli4l et ne sera donc pas pris en charge !

Reconfiguration de Busybox

A l'aide de la commande `fbr-make busybox-menuconfig`, vous pouvez régler le fonctionnement de Busybox. Pour couronnée de succès la configura-

tion du menu, la nouvelle configuration doit être enregistrée dans le répertoire `src/fbr/buildroot/package/busybox/busybox-<Version>.config`.

Ici aussi, un changement n'est probablement pas compatible avec la distribution officielle de fli4l et n'est donc pas pris en charge ! Tout au plus le fait de compléter le Busybox autour de nouvelles applets est sans problèmes, tant que vous utilisez le Busybox ainsi modifié seulement sur votre routeur fli4l, (il ne permet pas l'utilisation d'un tel Busybox modifié, pour vos propres OPTs).

Reconfiguration du paquet Kernel-Linux

A l'aide de la commande `fbr-make linux-menuconfig` ou `fbr-make linux-<version>-menuconfig` la configuration de tous les paquets Kernel ou d'un paquet Kernel spécifique est permis. Pour couronner de succès la configuration du menu, la nouvelle configuration doit être enregistrée dans le répertoire `src/fbr/buildroot/linux/linux-<version>/dot-config-<arch>`. Ceci est valable seulement pour un Kernel standard. Pour une variante du paquet Kernel, au lieu que le fichier soit indiqué `diff`, il sera indiqué `src/fbr/buildroot/linux/linux-<version>/linux-<version>_<variante>/dot-config-<arch>.diff`.

Il faut prendre également en compte comme dans le dernier paragraphe : un changement n'est probablement pas compatible avec la distribution officielle de fli4l et n'est donc pas pris en charge ! Tout au plus le fait de compléter le Kernel Linux autour de nouveaux modules c'est sans problèmes (il ne permet pas l'utilisation d'un tel Kernel modifié, pour vos propres OPTs).

4.24.7. Mise à jour des FBRs

Pour chacune des commandes décrites ci-dessus, un examen des FBRs est effectué pour une évolution de la mise à jour. En cas de divergence entre les sources, après avoir utilisé le `fbr-make` pour (décompacter le paquet `src` ou pour la copie de travail SVN) et le FBR dans `~/fbr/fbr-<branch>-<arch>/buildroot`, les différentes sources trouvées seront mise à jour. Les nouveaux paquets FBR ainsi que les anciens seront intégrés, les paquets FBR existants seront effacés. Les configurations sont comparées : les paquets FBR dont la configuration a été récemment modifiée ainsi que tous les paquets FBR qui en découle seront construits. Cela garantit que le FBR sur votre ordinateur est toujours conforme pour le développement de fli4l (excepté vos propres paquets FBR qui sont dans `~/fbr/own/`). **Cela signifie aussi que des modifications de la partie officielle et de la configuration du Buildroot seront perdues lors de la prochaine de la mise à jour !** C'est pourquoi, une reconfiguration des FBRs (voir ci-dessus) n'est pas recommandé, du moins pas si à la place des paquets `src` vous travaillez avec une copie de travail `src`. (Lors de la mise à jour d'une copie de travail SVN, vos changements de configuration locaux et les modifications apportées dans le SVN-Repository sont fusionnées, si bien que le problème de la perte de la configuration ne se produira pas ici). En revanche, vous pouvez reconfigurer vos propres paquets FBR facilement, sans provoquer une mise à jour avec la perte de données.

4.24.8. Intégrer vos propres programmes dans le FBR

La compilation des paquets FBR individuels est contrôlé par un petit fichier make. Si vous voulez développer vos propres paquets FBR, et si vous utilisez le fichier make une description de la configuration est dans `~/.fbr/own/<paquet>/`. Pour comprendre comment ceux-ci sont construits et comment écrire son propre fichier Make, une documentation décrit en détail le uClibc-Buildroots dans <http://buildroot.uclibc.org/downloads/manual/manual.html#adding-packages>.

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 mfkli4l.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 mfkli4l.txt](#) (Page 262).

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

5. Création une archive fli4l/Média de Boot

Avec l'option `--hdinstallpath <dir>` il est possible de faire une pré-installation sur une 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 262).

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 262) 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

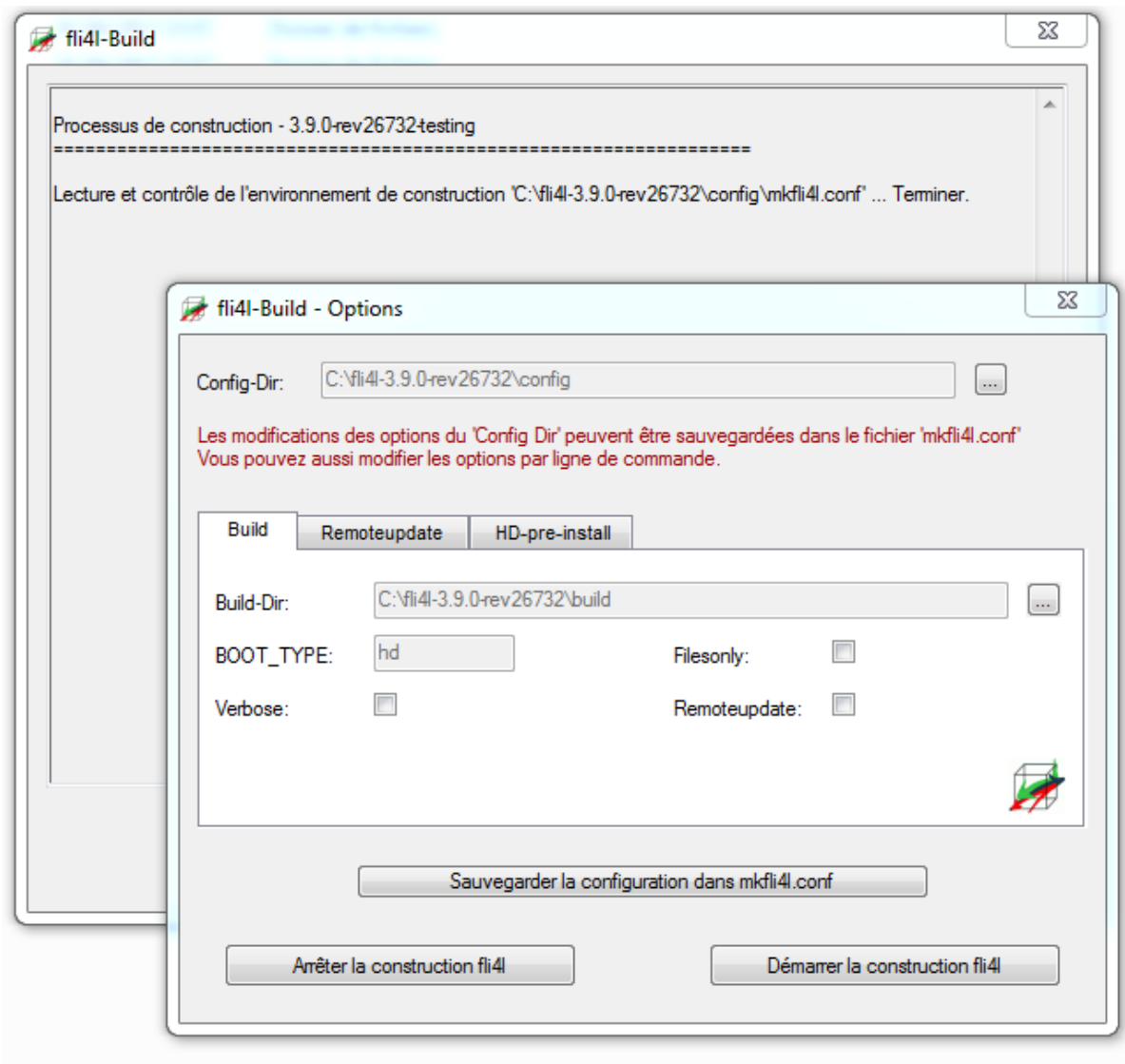


FIGURE 5.1.: Paramètre

- 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

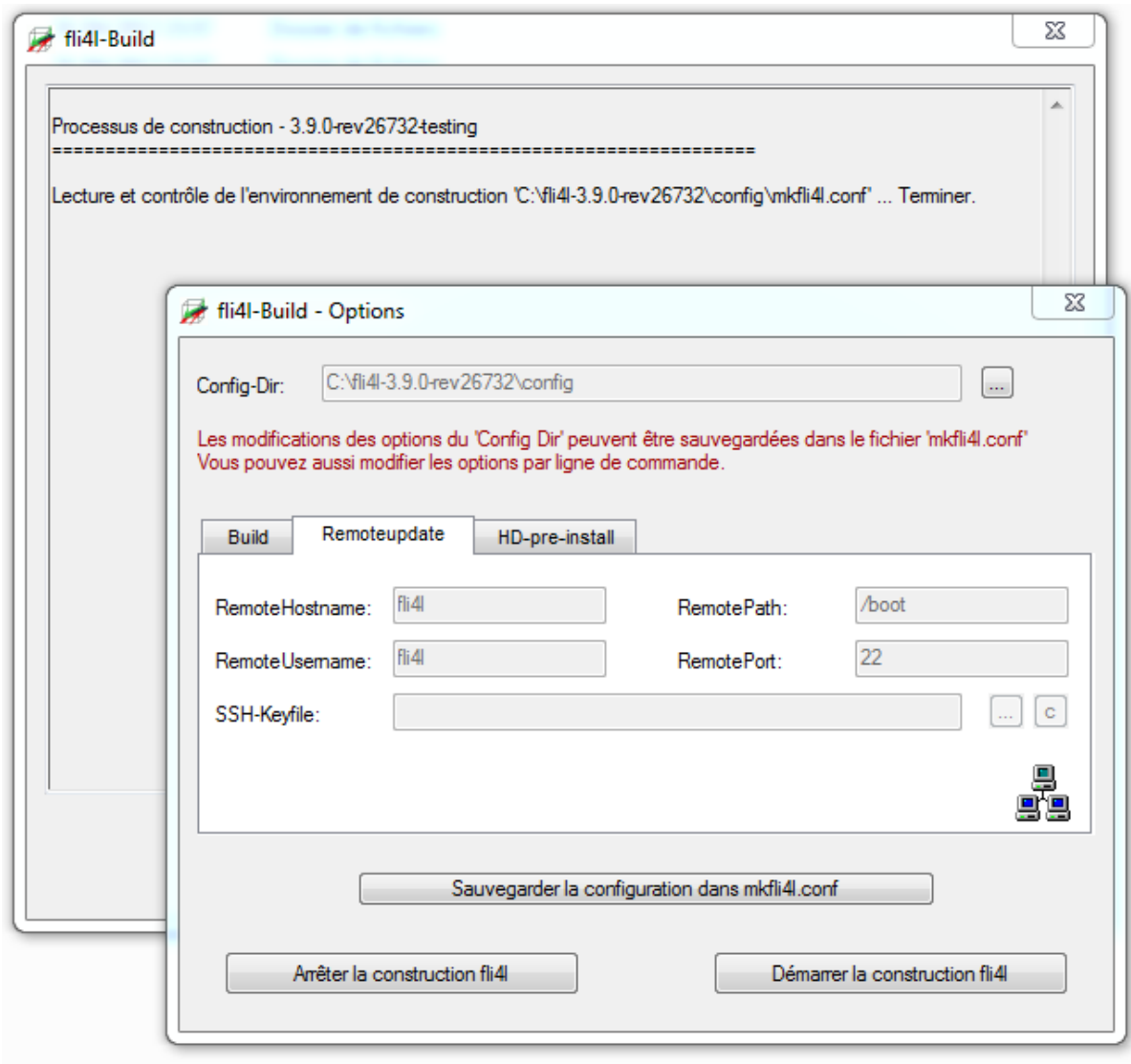


FIGURE 5.2.: Paramètre pour la mise à jour

- 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

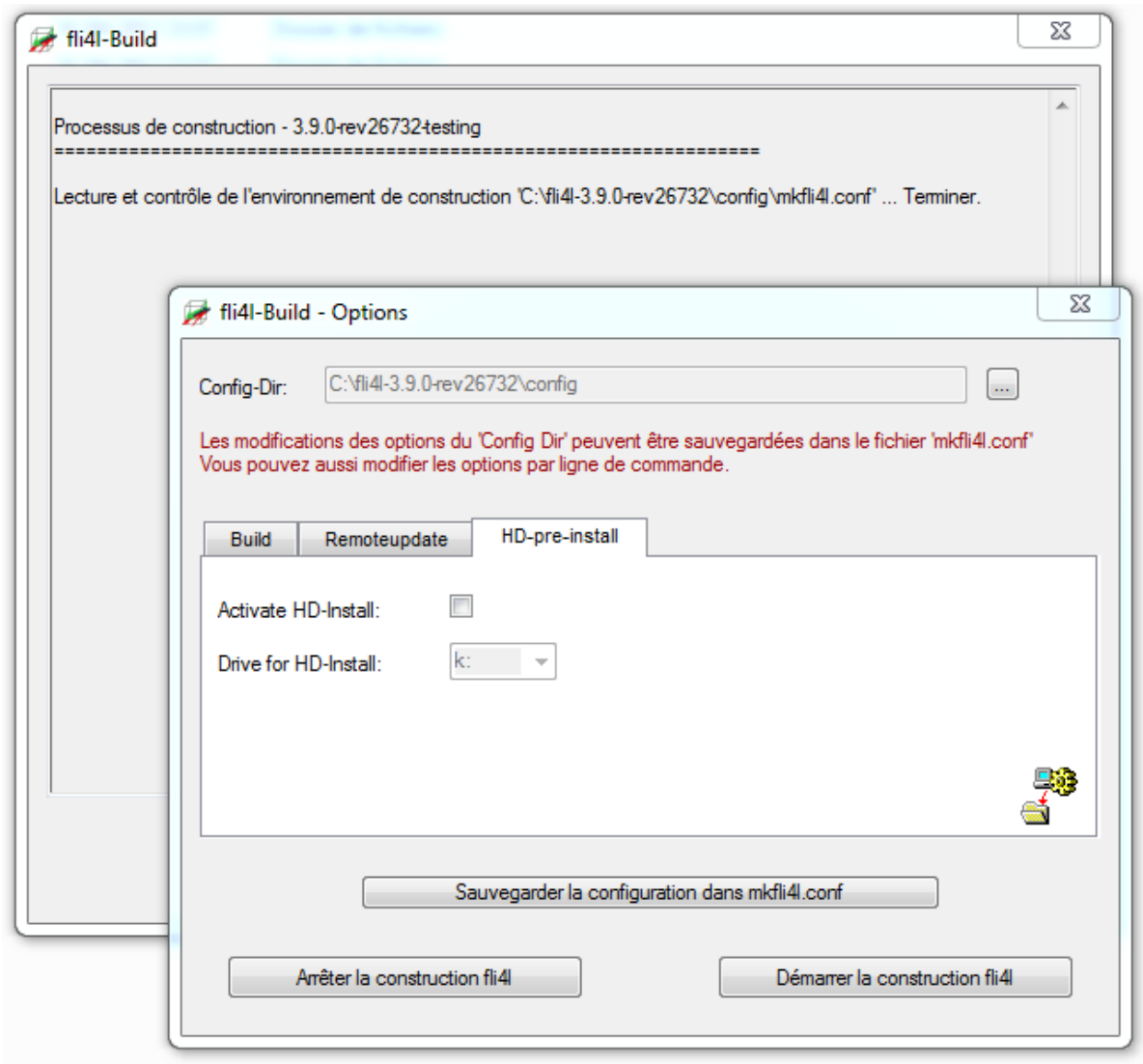


FIGURE 5.3.: Paramètre pour pré-installation du DD

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 221) 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 23) 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 289).

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 configuré 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 modifiés 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 TCP/IP-port 5000 :

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 Defaut-Route-Circuits
cpu	Donne la charge du CPU en pourcentage
date	Edite 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 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 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 faussée. 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 utiliserez 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 y 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 68), `IMOND_DIAL` (Page 68), `IMOND_ROUTE` (Page 68) et `IMOND_REBOOT` (Page 68), 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 taches 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 taches, 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 taches 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 affichent 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 tâches 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 tâche.
- 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 préférée – On doit Naturellement activer Syslog dans le client imonc. Imonc doit être connecté à une direction appropriée, la troisième possibilité proposée est d'utiliser le paquetage Capi2Text pour la signalisation d'appel.
- Effacer premier zéro : parfois devant le numéro de téléphonique est placé un zéro sup-

plémentaire. 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 est 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 "Délai 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 Port-forwarding 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 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 utilisiez 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 ou 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'icône 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 a 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 Re-booter (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 "ping" (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 paquets-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	dft 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".

8. Documentation pour développeur

8.1. Règles générales

Certaines règles doivent être respectées, lorsque vous ajoutez un nouveau paquetage dans la base de données OPT-fli4l, qui se trouve sur la page d'accueil du site fli4l. Le paquetage qui ne respectent pas à ces règles, sera supprimé de la base de données, sans avertissement préalable.

1. L'utilisateur ne doit faire, AUCUNE copie supplémentaire! fli4l fournit un système sophistiqué, Les données des paquetages-fli4l sont décompressés dans les répertoires de l'installation, tous les fichiers qui font fonctionner le routeur sont dans le répertoire `opt/`.
2. Les paquetages correctement emballer sont compressés : de telle sorte, que les paquetages soient facilement décompresser dans le répertoire-fli4l.
3. Les paquetages doivent être TOTALEMENT configurable dans le fichier de configuration. L'utilisateur ne doit pas faire de modifications sur d'autres fichiers de configurations. Vous ne devez pas mettre l'utilisateur en difficulté, sur des décisions difficiles à prendre, par exemple (à la fin du fichier de configuration avec une remarque en gros caractère : ONLY MODIFY IF YOU KNOW WHAT YOU DO).
4. Encore une remarque sur le fichier de configuration : les nom des variables doivent être claires et l'on doit savoir à quelle OPT elles appartiennent, par exemple `OPT_HTTPD` les nom des variables sont `OPT_HTTPD`, `HTTPD_USER_N`, etc.
5. S'il vous plaît, si vous avez compilé vous-même de petits (Programmes) binaire! Et si vous traduisez vous-même le FBR, pensez de désactiver les fonctionnalités inutiles.
6. Contrôler votre Copyright! Si vous utilisé un modèle de fichier, merci de respecter les droit d'auteur. Le copyright doit est remplacé ici par votre propre nom si vous créez vos fichiers. vérifié en particulier, les fichiers dans `config-`, `Check-` et les fichiers textes dans `opt-`. Si vous copiez La documentation mot à mot, le copyright de l'auteur d'origine doit être naturellement gardé!
7. Merci de diffuser seulement des types d'archivages, utilisant des formats libres. Il s'agit notamment de :
 - ZIP (`.zip`)
 - GZIP (`.tgz` ou `.tar.gz`)S'il vous plaît n'utilisez pas les autres formats tels que RAR, ACE, Blackhole, LHA, etc. Vous ne devez pas aussi utiliser les fichiers d'installation Windows (`.msi`) ou les archives auto-extractable et les fichiers d'installation (`.exe`).

8.2. Compiler les programmes

Pour pouvoir compiler des programmes vous allez avoir besoin du paquetage « src » qui est disponible à part. Il y a également une documentation pour compiler vos propre programme pour fli4l.

8.3. Concept modulaire

Depuis la version 2.0, fli4l est réparti en modules (ou paquets), par exemple

- fli4l-3.10.4 <— paquetage base
- dns-dhcp
- dsl
- isdn
- sshd
- et bien d'autres ...

Avec le paquetage base, fli4l est un simple routeur Ethernet. Pour ISDN et/ou DSL, vous devez décompacter le paquetage ISDN et/ou DSL dans le répertoire fli4l. Il en va de même pour les autres paquets.

8.3.1. mkfli4l

À partir des paquets et en fonction des paramètres spécifiques des fichiers de configurations, les fichiers suivants seront créés, le fichier avec les configurations, `rc.cfg` et deux archives `rootfs.img` et `opt.img`, ces fichiers sont produits à l'aide du programme `mkfli4l`, celui-ci lie les différents paquets et contrôle les fichiers de configurations pour repérer d'éventuelles erreurs.

`mkfli4l` accepte les options indiqués dans le tableau 8.1. Si aucune valeur n'est indiquée, les valeurs par défaut entre les parenthèses seront prises en compte. Vous pouvez voir la liste complète des options dans (le tableau 8.1) avec la commande :

```
mkfli4l -h
```

8.3.2. Structure

Un paquetage peut contenir plusieurs opts, mais si en général le paquetage contient seulement un opt, il est opportun de nommer le paquetage du même nom que l'OPT. Vous pourrez ensuite remplacer <PAQUETAGE> par le nom des paquets respectif. Un paquetage comprend les éléments suivants :

- Gestionnaire de fichiers
- Documentation
- Documentation du développeur
- Programme client
- Code source
- Autres fichiers

Les différentes parties sont détaillées ci-dessous.

8.3.3. Configuration du paquetage

Dans le fichier `config/<PAQUETAGE>.txt`, les modifications de la configuration du paquetage sont réalisées par l'utilisateur. Toutes les variables du fichier de configuration doivent commencer uniformément par le nom de ou des OPTs, par exemple :

```
#-----
# Optional package: TELNETD
```

TABLE 8.1.: Paramètres pour mkfli41

Option	Signification
-c, --config	Avec cette option, on définit le répertoire dans lequel mkfli41 recherche les fichiers de config des paquetages ; (par défaut : config)
-x, --check	On place ici la liste des fichiers des paquets, dans le quel mkfli41 doit contrôler et rechercher les erreurs (<PAQUETAGE>.txt, <PAQUETAGE>.exp et <PAQUETAGE>.ext ; (par défaut : check)
-l, --log	On place ici le fichier journal, dans le quel mkfli41 enregistre les messages d'erreurs et les avertissements ; (par défaut : img/mkfli41.log)
-p, --package	On place ici le paquetage qui doit être examiné, on peut indiquer dans cette option plusieurs paquetages à la fois pour qu'ils soient examinés. Cependant si vous utilisez l'option -p, l'option <check_dir>/base.exp est en principe d'abord vérifiée, afin de gérer les paramètres général du paquetage de base.
-i, --info	Avec cette option, on affiche des informations sur le déroulement de la construction (lecture des fichiers, vérification des fichiers, les problèmes rencontrés seront affichés lors du processus de contrôle).
-v, --verbose	Même signification que la variante -i
-h, --help	Avec cette option, on affiche l'aide
-d, --debug	Avec cette option, vous pouvez déboguer le processus de construction. Cette option sert surtout pour les développeurs de paquetage, qui souhaitent en savoir un peu plus, sur le déroulement du contrôle des paquetages.
Debugoption	Signification
check	show check process
zip-list	show generation of zip list
zip-list-skipped	show skipped files
zip-list-regexp	show regular expressions for zip list
opt-files	check all files in opt/<PAQUETAGE>.txt
ext-trace	show trace of extended checks

```
#-----
OPT_TELNETD='no'          # install telnetd: yes or no
TELNETD_PORT='23'        # telnet port, see also FIREWALL_DENY_PORT_x
```

Le fichier de configuration de l'OPT doit être configuré en conséquence avec une en-tête (voir ci-dessus). Cela augmente la clarté, en particulier si le paquetage contient plusieurs OPTs. Les variables associées à l'OPT - ne doivent pas être écrites en retrait - également pour plus de clarté. Les commentaires et lignes vides sont autorisés, les commentaires doivent commencer de manière uniforme jusqu'à la 33ème colonne. Si une variable ou un commentaire occupe plus de 32 caractères, la ligne sera décalée à la 33ème colonne. Les commentaires plus longs doivent être séparés dès la 33ème colonne pour commencer une nouvelle ligne. Cette mesure vise à améliorer la lisibilité du fichier de configuration.

Toutes les valeurs derrière le signe égale doivent être écrites entre des guillemets ¹, autrement, lors du boot vous pouvez avoir des problèmes.

Les variables incluses (voir plus bas) dans le fichier `rc.cfg` sont activées, toutes les autres sont ignorées. La seule exception est la variable qui porte le nom `<PAQUETAGE>_DO_DEBUG`. Celle-ci sert pour déboguer des paquetages et est enregistrée en globalité.

8.3.4. Liste des fichiers à copier

Le fichier `opt/<PAQUETAGE>.txt` doit contenir les instructions suivantes.

- Quels fichiers correspond, à quel OPT,
- Quand le fichier `opt.img` ou `rootfs.img` doit être transféré et généré,
- Quel ID-utilisateur (uid), ID-groupe (gid) et les droits qu'ils doivent obtenir,
- Quel conversion doit être faite à l'archive avant l'enregistrement.

Quand `mkfli41` est exécuté il se base sur les archives nécessaires à l'installation.

Les lignes vides et commençant par « # » sont ignorées. Dans l'une des deux premières lignes, vous devez indiquer la version du format du paquetage, qui doit être :

```
<première colonne>  <deuxième colonne>  <troisième colonne>
opt_format_version      1                -
```

Les autres lignes ont la syntaxe suivante :

```
<première colonne> <deuxième colonne> <troisième colonne> <colonne suivante>
Variable          valeur          Fichier          Option
```

1. Dans la première colonne se trouve les noms des variables, elle dépend du fichier appliqué de la troisième colonne. Si le nom de la variable apparaît plusieurs fois dans la première colonne, cela veut dire que plusieurs fichiers en dépendent. Chaque variable qui est placée dans le fichier `opt/<PAQUETAGE>.txt`, est marqué par `mkfli41`.

Si plusieurs variables doivent être testés avec la même valeur, une liste de variables peut être utilisée (elles seront séparés par une virgule). Dans ce cas, il sera suffisant d'enregistrer au moins *une* valeur requise dans la deuxième colonne pour toutes les variables. Il est important de ne *pas* mettre d'espace entre les variables !

1. Il est possible d'utiliser les guillemets simples ou les guillemets doubles, on peut donc écrire `F00='bar'` ou `F00="bar"`. Cependant, l'utilisation des guillemets doubles doivent être exceptionnelle, vous devez vérifier absolument comment le shell-Unix traite des guillemets simples et doubles

Pour les variables OPT (se sont les variables qui commence par OPT_ et qui accepte les valeurs YESNO) le préfixe ogOPT_fg peut être omis. En outre, il n'y a pas d'importance si les variables sont indiquées en majuscule, en minuscule (ou mixte).

2. Dans la deuxième colonne sont indiquées les valeurs. Si la variable de la première colonne correspond à la valeur indiqué dans la deuxième colonne et si elle est activé, (voir plus bas), de plus si vous avez dans la première colonne %-Variable qui se répète pour chaque indice différent, alors ces variables vérifient la valeur de la deuxième colonne, si elle correspond, alors tous les fichiers de la troisième colonne seront copiés. Il est à noter qu'en raison des valeurs des même variables un seul fichier sera copié.

Il est possible d'indiquer le caractère og!fg devant la valeur. Dans ce cas, le test est annulé, ce qui signifie que le fichier sera copié seulement si la variable ne contient *pas* de valeur.

3. Dans la troisième colonne est indiqué le nom du fichier. Le chemin est relatif au répertoire **opt**. Le fichier doit exister et être lisible, sinon, il y aura une erreur et le programme **mkfli41** qui génère automatiquement la construction du média de boot s'arrêtera.

Si le nom du fichier commence par **rootfs:-**préfixe, le fichier sera copié dans la liste et sera inclus dans le RootFS avec les autre fichiers. Si il y a un préfixe au fichier il sera supprimé avant la copie.

Si le fichier se trouve dans le sous répertoire config, il sera ajouté dans la liste des fichiers du répertoire config, mais ce fichier ne pourra pas avoir de préfixe **rootfs:-**, comme les fichiers qui proviennent du sous répertoire **opt**.

Si le fichier à copier est un module-kernel, on peut remplacer le nom de la version du kernel par `#{KERNEL_VERSION}`. **mkfli41** prendra alors la version configurée et l'intégrera. Cela permet d'avoir un packaging de module-kernel pour chaque versions différente, en plus la bonne version du kernel sera toujours copiée sur le routeur. Au sujet des modules-kernel le chemin peut être complètement omis, car **mkfli41** trouve le chemin des modules en utilisant les fichiers **modules.dep** et **modules.alias**, reportez-vous à la section « [Résolution automatique pour la traçabilité des modules-kernel](#) » (Page 294)

4. Vous pouvez voir dans la colonne du tableau 8.2 le détail des options sur les propriétés, les groupes, les droits d'accès des fichiers et les conversions.

Quelques exemples :

- Copie le fichier, si OPT_TELNETD='yes', placez uid/gid pour le root et placez les droits sur 755 (rwxr-xr-x).

```
telnetd    yes    files/usr/sbin/in.telnetd mode=755
```
- Copie le fichier, placez uid/gid pour le root, les droits sur 555 (r-xr-xr-x) et converti le fichier dans le format Unix en même temps il supprime tous les caractères superflus.

```
base      yes    etc/rc0.d/rc500.killall mode=555 flags=sh
```
- Copie le fichier, si PCMCIA_PCIC='i82365', placez uid/gid pour le root et placez les droits sur 644 (rw--r--r)

```
pcmcia_pcic i82365 files/lib/modules/#{KERNEL_VERSION}/pcmcia/i82365.ko
```
- Copie le fichier, si l'une des variables NET_DRV_% correspond à la deuxième colonne, placez uid/gid pour le root et placez les droits sur 644 (rw--r--r)

```
net_drv_%   3c503   3c503.ko
```
- Copie le fichier, si la variable POWERMANAGEMENT ne contient *pas* la valeur « none » :

```
powermanagement !none etc/rc.d/rc100.pm mode=555 flags=sh
```

TABLE 8.2.: Options pour les fichiers

Option	Signification	Valeur standard
type=	type d'entrée : <i>local</i> Objet fichiers-système <i>file</i> Fichier <i>dir</i> Répertoire <i>node</i> Matériel <i>symlink</i> Lien (symbolique) Si elle est présente, cette option doit venir en premier. Le type « local » représente un type d'objet existant dans le fichier système et correspond donc (le cas échéant) à « file », « dir », « node » ou « symlink ». Les autres types, à l'exception de « file » peuvent être utilisés pour enregistrer des archives, il ne doivent pas être présent dans le fichier système local. Par ex., vous pouvez les utiliser pour créer des fichiers de périphérique dans l'archive-RootFS.	local
uid=	Propriété du fichier, soit numériquement, soit en tant que mot de passe	root
gid=	Groupe du fichier, soit numériquement, soit en tant que nom de groupe	root
mode=	Pour les droits d'accès	Les Fichiers et matériels : rw-r--r-- (644) Les Répertoires : rw-r--r-x (755) Les liens : rw-rw-rw-x (777)
Les flags= (type=file)	Conversion avant l'enregistrement dans l'archive : <i>utxt</i> Conversion au format-Unix <i>dtxt</i> Conversion au format-DOS <i>sh</i> Script Shell : Conversion dans le format-UNIX, suppression des signes inutiles	
name=	Nom alternatif sous lequel l'entrée est enregistrée dans l'archive	
devtype= (type=node)	Décrit le type de matériel (« c » pour s'orienter vers la marque et « b » pour s'orienter vers le matériel de bloc. Doit être à la deuxième place.	
major= (type=node)	Décrit le nombre soi-disant « Majeur » -Numéro de fichier du périphérique. Doit être à la troisième place.	
minor= (type=node)	Décrit le nombre soi-disant « Mineur » -Numéro de fichier du périphérique. Doit être à la quatrième place.	
linktarget= (type=symlink)	Décrit la cible du lien symbolique. Doit être à la deuxième place.	

- Copie le fichier, si l'une des variables OPT OPT_MYOPTA ou OPT_MYOPTB contient la valeur « yes » :

```
myopta,myoptb yes files/usr/local/bin/myopt-common.sh mode=555 flags=sh
```

Cet exemple est juste un raccourci pour :

```
myopta yes files/usr/local/bin/myopt-common.sh mode=555 flags=sh
myoptb yes files/usr/local/bin/myopt-common.sh mode=555 flags=sh
```

Et celui-ci est un raccourci pour :

```
opt_myopta yes files/usr/local/bin/myopt-common.sh mode=555 flags=sh
opt_myoptb yes files/usr/local/bin/myopt-common.sh mode=555 flags=sh
```

- Copie du fichier opt/files/usr/bin/beep.sh dans l'archive rootfs, mais sera renommer bin/beep avant :

```
base yes rootfs:files/usr/bin/beep.sh mode=555 flags=sh name=bin/beep
```

Les fichiers sont uniquement copiés, si les conditions indiquées plus haut sont remplies, si la variable est placée sur `OPT_PAQUETAGE='yes'`. Quelle variable-OPT est associée ? Pour la vérification voir le fichier `check/<PAQUETAGE>.txt`

Si une variable est référencée dans le paquetage et que celle-ci n'est pas définie, il peut arriver que le paquetage correspondant n'est pas installé. Cela mènerait à un message d'erreur du programme `mkfli4l`, car `mkfli4l` attend que toutes les variables référencées dans le fichier `opt/<PAQUETAGE>.txt` soient définies.

Pour gérer correctement cette situation, on introduit la fonction-« weak » dans le format suivant :

```
weak      variable      -
```

Si la variable est définie, même si elle n'est pas disponible la valeur de celle-ci sera indiquée « undefiniert ». Toutefois, il convient de noter ici que le préfixe « `OPT_` » ne doit *pas* être omis (s'il existe), sinon la variable sera définie *sans* ce préfixe.

Un exemple du fichier `opt/rrdtool.txt` :

```
weak opt_openvpn -
[...]
openvpn yes files/usr/lib/collectd/openvpn.so
```

Sans la définition **weak**, la commande `mkfli4l` affichera un message d'erreur si vous utilisez le paquetage « `rrdtool` » et si le paquetage « `openvpn` » n'est pas aussi présent. La définition **weak** est également utilisé dans le cas où le paquetage « `openvpn` » n'existe pas, il n'y aura aucun message d'erreur.

Fichiers de configurations spécifiques

Dans certaine situations, on aimerait remplacer des fichiers de configurations originaux par des fichiers spécifiques, qui serait compacté dans l'archive `opt.img`, par exemple, ajouter une Key-hôte ou ajouter son propre Scripte-Firewall, ... `varmkfli4l` supporte ce scénario, il vérifie si un fichier est disponible dans le répertoire config, dans ce cas, ce fichier sera ajouté dans la liste des fichiers de l'archive `opt.img` ou `rootfs.img`.

Une autre façon d'ajouter des fichiers de configurations spécifiques dans l'archive, est décrit dans le chapitre [Contrôle de la configuration avancée](#) (Page 311).

Résolution automatique pour la traçabilité des modules-kernel

Dans certaine circonstance un module-kernel a parfois besoin d'autres modules-kernel. Ces modules doivent être chargés avant et seront également inclus dans l'archive. `mkfli4l` détermine la traçabilité des modules avec les fichiers `modules.dep` et `modules.alias` (les deux fichiers sont générés à la compilation du kernel) ainsi ils ajouteront automatiquement tous les modules nécessaires dans l'archive. Par exemple l'enregistrement peut être le suivant

```
net_drv_% ne2k-pci ne2k-pci.ko
```

Le pilote `ne2k_pci` dépend des fichiers suivant `8390.ko`, et `crc32.ko` ils seront ajoutés en premiers dans l'archive.

L'enregistrement du `modules.dep` et du `modules.alias` dans le RootFS est nécessaire, ensuite `modprobe` utilise ces fichiers pour charger les pilotes.

8.3.5. Analyse des variables de configuration

Avec le fichier `check/<PAQUETAGE>.txt` les variables peuvent être contrôlés pour leurs validités. Ce contrôle était intégré dans les versions précédentes du programme `mkfli4l`, mais les paquets `fli4l` sont modulaire et nous avons du installé un second-contrôle dans ce fichier. Dans ce fichier, une ligne est accessible pour chaque variable du fichier de configuration. Ces lignes se composent de quatre à cinq colonnes, qui ont les fonctions suivantes :

1. **Variable** : Cette colonne indique le nom des variables à vérifier, qui sont dans le fichier `config`. Cette colonne indique le nom des variables à vérifier, si c'est une *liste de variables*, elle peut apparaître plusieurs fois avec différents indices, donc à la place de l'indice le signe pourcentage (%) sera inséré dans la variable. Le signe sera toujours indiqué comme ceci « `_%` » au milieu de la variable et comme ceci « `_%` » à la fin de la variable. La variable peut contenir plusieurs signes pourcentages, de sorte à réaliser des matrices multidimensionnels. Normalement vous ne serez pas amené à voir ces variables un peut étranges avec deux signes pourcentage, tels que « `foo_%%` ».

Dans certaine situation, nous pouvons avoir besoin de variable optionnelle supplémentaire dans le fichier `config`. Ces variables sont présentes dans le fichier de contrôle, elles sont marquées et précédées du signe « `+` ». Dans le tableau, vous pouvez voir aussi des variables précédées du signe « `++` ». Avec le signe « `+` », ces variables sont présentes ou absente du fichier `config`. Avec le signe « `++` », ces variables sont totalement absente du fichier `config`. Vous pouvez les rajouter manuellement, si vous en avez besoin dans le fichier `config`.

2. **OPT_VARIABLE** : Cette colonne indique, les variables OPT. Ces variables sont seulement vérifiées pour leurs validités, à savoir si la variable est sur « `yes` ». S'il n'y a pas de variable OPT un « `-` » sera indiqué. Dans ce cas, la variable doit être définie dans le fichier de configuration, sauf si une valeur par défaut est défini (voir ci-dessous). Le nom de la variable OPT peut être arbitraire mais doit commencer par le préfixe « `OPT_` ».

Si une variable ne dépend d'aucune variables OPT, elle sera considérée comme *active*. Si elle dépend d'une variable OPT, elle sera explicitement active, si

- la variable OPT est active et
- la variable OPT contient la valeur "yes".

Dans tous les autres cas, la variable est inactive.

Remarque : les variables OPT inactifs seront remplacées sur « `no` » par `mkfli4l` si elle sont réglées sur « `yes` » dans le fichier de configuration, un message d'avertissement sera alors généré (c'est à dire « `OPT_Y='yes' ignoré, parce que OPT_X='no'` »). Pour les chaînes de dépendance transitive (OPT_Z dépend de OPT_Y qui à son tour dépend de OPT_X) cela fonctionnera de manière fiable, si les noms de tous les variables OPT commencent par « `OPT_` ».

3. **VARIABLE_N** : Si dans la première colonne, vous avez des noms de variables avec le signe %, elles spécifient la fréquence d'apparition de la variable (c'est la variable qui s'appelle *N-variable*). Si la variable est multidimensionnel, le dernier index est en l'occurrence spécifié. Si la variable dépend d'un OPT, la N-variable doit dépendre ou pas du même OPT. Si la variable ne dépend pas d'un OPT, la N-variable ne doit également pas en dépendre. Si aucune N-variable existe, le signe « `-` » sera spécifier.

Pour la compatibilité avec les futures versions de `fli4l` la variable spécifiée ici *doit* être identique à la variable qui est dans `OPT_VARIABLE` et le dernier signe « `%` » sera remplacé

par un « N » tout ce qui suit sera retiré. Pour la liste `HOST_%_IP4` vous devez assigner N-Variable `HOST_N` et pour la liste `PF_USR_CHAIN_%_RULE_%` vous devez aussi assigner N-Variable `PF_USR_CHAIN_%_RULE_N`. *Toutes les autres définitions de N-variable ne seront pas compatibles avec les versions futures de fli4l!*

4. **VALUE** : cette colonne donne les valeurs possibles que peut prendre la variable. Vous pouvez par ex. avoir les informations suivant :

Nom	Signification
NONE	ne déclenche aucun contrôle
YESNO	La variable doit être sur « yes » ou sur « no »
NOTEMPTY	La variable ne peut pas être vide
NOBLANK	La variable ne devrait pas contenir d'espaces
NUMERIC	La variable doit être numérique
IPADDR	La variable doit être une adresse IP
DIALMODE	La variable doit être sur « on », « off » ou « auto »

Si vous indiquez le préfixe « `WARN_` », la valeur sera irrégulière et sera indiqué par un message, `mkfli4l` ne s'arrêtera pas, mais affichera seulement un avertissement.

Le contrôle est défini par des expressions régulières dans le fichier `check/base.exp`. Ce fichier à été récemment étendu, il contient maintenant des contrôles complémentaires suivants : `HEX`, `NUMHEX`, `IP_ROUTE`, `DISK` et `PARTITION`.

Si les développeurs-opt ont besoin de rajouter une entrée, le nombre de termes peut, à tout moment être étendu.

En outre, les expressions régulières peuvent être ajoutées directement dans le fichier du répertoire `check`, on peut également se référer à des expressions existantes. Par exemple au lieu d'utiliser `YESNO` on pourrait également écrire

```
RE:yes|no
```

cela est utile pour un test qui est effectué qu'une seule fois, il est relativement simple. Pour de plus amples informations, voir le chapitre suivant.

5. **Paramètre par défaut** : dans cette colonne, une valeur facultative par défaut pour les variables peut être définie, dans le cas où la variable n'est pas spécifié dans le fichier de configuration.

Remarque : à présent si cela ne fonctionne pas pour les variables de la liste. La variable ne doit pas être facultative, donc il ne doit pas avoir le signe « + » devant le nom de la variable.

Exemple :

```
OPT_TELNETD      -          -          YESNO      "no"
```

La variable `OPT_TELNETD` est maintenant manquante dans le fichier de configuration, mais dans le fichier `rc.cfg` elle sera affichée avec la valeur « no ».

La variable avec le signe pourcentage peut être mieux expliquée avec un exemple. Voici une partie du fichier `check/base.txt` :

```
NET_DRV_N        -          -          NUMERIC
NET_DRV_%        -          NET_DRV_N    NONE
NET_DRV_%_OPTION -          NET_DRV_N    NONE
```

En d'autres termes, en fonction de la valeur indiquée dans `NET_DRV_N` les variables `NET_DRV_N`, `NET_DRV_1_OPTION`, `NET_DRV_2_OPTION`, `NET_DRV_3_OPTION`, etc. seront vérifiées.

8.3.6. Définitions pour contrôler les variables de configuration

Introduction sur les expressions régulières

Dans la version 2.0, il y a que 7 expressions, susceptibles d'examiner les variables : NONE, NOTEMPTY, NUMERIC, IPADDR, YESNO, NOBLANK, DIALMODE. Ces expressions ont été fixées dans `mkfli41` pour la vérification, ils ne sont pas extensibles et se limitent à l'essentielle aux « types de données », avec juste ce qu'il faut pour pouvoir faire le contrôle.

Dans la version 2.1 un nouveau contrôle a été créé. L'objectif de cette nouvelle création est de faire un contrôle plus flexible des variables, qui sera en mesure d'examiner des expressions plus complexes. C'est la raison pour laquelle les expressions régulières seront utilisées dans un ou plusieurs fichiers séparés. Il sera possible, d'une part, de vérifier les variables avant le contrôle par `mkfli41` et d'autre part, les développeurs pourront définir leurs propres expressions, pour la configuration et le contrôle de leur paquetage.

Vous pouvez trouver une description des expressions régulières, dans « man 7 regex », ou par exemple ici : <http://unixhelp.ed.ac.uk/CGI/man-cgi?regex+7>.

Spécification des expressions régulières

On peut spécifier des expressions de deux manières différentes :

1. Extension `exp` spécifique au paquetage, dans le fichier `check/<PAQUETAGE>.exp`

Ce fichier se trouve dans le répertoire `check` et porte le même nom que son paquetage, par ex. `base.exp`. Les expressions contiennent les définitions, qui sont référencées dans le fichier `check/<PAQUETAGE>.txt`. Ainsi, `check/base.exp` peut contrôler les définitions. Bien connu le fichier `check/isdn.exp` qui contrôle la définition de la variable `ISDN_CIRC_?_ROUTE` (à l'origine le contrôle de cette variable était absent cela a été modifié).

Chaque définition sera écrite entre deux apostrophes, la syntaxe est la suivante :

```
<Name> = '<expression régulière>' : '<le message d'erreur>'
```

Autre exemple de la `check/base.exp` :

```
NOTEMPTY = '.*[~ ]+.*'          : 'should not be empty'
YESNO     = 'yes|no'             : 'only yes or no are allowed'
NUMERIC   = '0|[1-9][0-9]*'      : 'should be numeric (decimal)'
OCTET     = '1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5]'
           : 'should be a value between 0 and 255'
IPADDR    = '((RE:OCTET)\.){3}(RE:OCTET)' : 'invalid ipv4 address'
EIPADDR   = '()|(RE:IPADDR)'
           : 'should be empty or contain a valid ipv4 address'
NOBLANK   = '[^ ]+'              : 'should not contain spaces'
DIALMODE  = 'auto|manual|off'     : 'only auto, manual or off are allowed'
NETWORKS  = '(RE:NETWORK)([[:space:]]+(RE:NETWORK))*'
           : 'no valid network specification, should be one or more
             network address(es) followed by a netmask,
             for instance 192.168.6.0/24'
```

Dans les expressions régulières, vous pouvez toujours rajouter une référence, dans une définition existante. Cela est plus facile que de construire une expression régulière. Il suffit simplement d'insérer la référence de cette manière `'(RE :référence)'`. (Voir la définition de l'expression `NETWORKS` ci-dessus pour un exemple approprié.)

Les messages d'erreur ont tendance à être trop longs. Il est donc possible, de les afficher sur plusieurs lignes. Vous devez toujours utiliser au début de la ligne un espace ou une tabulation. Lors de la lecture du fichier `check/<PAQUETAGE>.exp` des espaces supplémentaires sont réduits à un seul espace et une tabulation sera remplacé par des espaces.

La configuration du fichier `check/<PAQUETAGE>.exp` pourrait alors ressembler à ce qui suit :

```
NUM_HEX      = '0x[[:xdigit:]]+'
              : 'should be a hexadecimal number
                (a number starting with "0x")'
```

2. Les expressions régulières sont directement installées dans le fichier de contrôle `check/<PAQUETAGE>.txt`

Certaines expressions sont utilisé qu'une seule fois, ce n'est pas la peine de définir ces expressions régulières dans le fichier `check/<PAQUETAGE>.exp`. Nous pouvons alors, enregistrer simplement ces expressions dans le fichier-Check, par ex. :

# Variable	OPT_VARIABLE	VARIABLE_N	VALUE
MOUNT_BOOT	-	-	RE:ro rw no

La variable MOUNT_BOOT ne peut qu'accepter les valeurs « ro », « rw » ou « no » toutes les autres sont rejetées

Si vous voulez référencer une expression régulière existants, vous devez simplement ajouter la référence de cette manière « (RE :...) », par ex. :

# Variable	OPT_VARIABLE	VARIABLE_N	VALUE
LOGIP_LOGDIR	OPT_LOGIP	-	RE:(RE:ABS_PATH) auto

Extension d'une expression régulière déjà existant

Si vous ajoutez un paquetage optionnel supplémentaire, vous devez ajouter une expression régulière pour que la valeur de la variable soit examinée, l'expression régulière doit être agrandie, cela se passe simplement par l'ajout une nouvelle définition des valeurs dans l'expression régulière (comme décrit plus haut). Le complément de l'expression régulière existante est copié dans le fichier `check/<PAQUETAGE>.exp`. L'expression existante sera modifiée, par le caractère « + » qui sera ajouté au premier plan. L'expression existante sera complétée par une nouvelle valeur, ainsi la nouvelle valeur est ajoutée comme l'alternative à la valeur existante. Si une autre expression utilise l'expression qui a été complétée, le complément de cette expression sera aussi valable. Vous pouvez indiquer un message d'erreur, il sera simplement ajouté après l'expression.

Voici un exemple pour les pilotes-Ethernet :

- Le paquetage-base a un ensemble de pilotes Ethernet prédéfini, on sélectionne la variable `NET_DRV_x` avec l'expression régulière `NET_DRV` pour contrôler cette variable, voici ce qui est écrit dans le fichier de contrôle :

```
NET_DRV      = '3c503|3c505|3c507|...'
              : 'invalid ethernet driver, please choose one'
              : 'of the drivers in config/base.txt'
```

- Nous mettons à disposition le paquetage-PCMCIA pour avoir des pilotes de périphérique supplémentaires, vous devez alors compléter la variable `NET_DRV`, pour avoir quelque chose comme ceci :

```
PCMCIA_NET_DRV = 'pcnet_cs|xirc2ps_cs|3c574_cs|...' : ''
+NET_DRV       = '(RE:PCMCIA_NET_DRV)' : ''
```

Maintenant, vous pouvez également sélectionner les pilotes PCMCIA supplémentaires.

Expression régulière élargie en relation avec les variables YESNO

Après avoir ajouté le pilote PCMCIA dans NET_DRV comme décrit plus haut. Si le paquetage « pcmcia » est désactivé et si vous voulez quand même choisir un pilote PCMCIA dans `config/base.txt`, sans avoir de message d'erreur à la création du support de boot. Vous pouvez modifier l'expression régulière avec la variable YESNO dans la configuration. Pour cela vous devez ajouter des parenthèses immédiatement après le nom de la variable sur l'expression pour déterminer si l'expression est étendu. Si la variable est active avec la valeur « yes », l'expression sera étendu, autrement elle ne l'est pas.

```
PCMCIA_NET_DRV      = 'pcnet_cs|xirc2ps_cs|3c574_cs|...' : ''
+NET_DRV(OPT_PCMCIA) = '(RE:PCMCIA_NET_DRV)' : ''
```

Maintenant si on veut utiliser par ex. le pilote `xirc2ps_cs` dans `config/base.txt` et si la variable est paramétré sur `OPT_PCMCIA='no'`, il y aura un message d'erreur à la création des archives.

Remarque : si cela ne fonctionne *pas*, la variable n'est peut être pas définie explicitement dans le fichier de configuration mais elle obtient une valeur avec un paramètre par défaut dans `check/<PAQUETAGE>.txt`. Dans ce cas, vous devez définir explicitement la variable et enlever le paramètre par défaut si nécessaire.

Définir une expression régulière en fonction d'autres variables

Vous pouvez alternativement utiliser toutes les valeurs de la variable, à condition, que la syntaxe apparaît comme indiquer ci-dessous :

```
+NET_DRV(KERNEL_VERSION=~'^3\.14\..*$') = ...
```

Si l'expression `KERNEL_VERSION` correspond au donné, c'est-à-dire ici à utilisation du kernel version 3.14, alors la liste complète des pilotes réseaux qui tournent avec celui-ci seront autorisés.

Remarque : si cela ne fonctionne *pas*, la variable n'est peut être pas définie explicitement dans le fichier de configuration mais elle obtient une valeur avec un paramètre par défaut dans `check/<PAQUETAGE>.txt`. Dans ce cas, vous devez définir explicitement la variable et enlever le paramètre par défaut si nécessaire.

Message d'erreur

Si pendant le contrôle, `mkfli41` trouve une erreur, un message de cette erreur apparaîtra exemple :

```
Error: wrong value of variable HOSTNAME: '' (may not be empty)
Error: wrong value of variable MOUNT_OPT: 'rx' (user supplied regular expression)
```

La première erreur, a été définie dans le fichier `check/<PAQUETAGE>.exp`, une indication sur l'erreur sera affichée. La deuxième erreur a été spécifié directement dans le fichier `check/<PAQUETAGE>.txt`, il n'y aura aucune indication supplémentaire sur la raison de l'erreur.

Définition des expressions régulières

Les expressions régulières sont définies comme indiqué ci-dessous :

Expression régulière : vous pouvez paramétrer une ou plusieurs options, vous devez les séparer par le signe « | », par ex. « ro|rw|no ». Si l'une des options s'applique, alors l'expression sera appliquée (ici les expressions valides sont « ro », « rw » et « no »).

Une option est une concaténation de tronçon, qui sont simplement reliés entre eux.

Un tronçon est un « Atome », suivi pas un signe quantificateur « * », « + », « ? » ou « {min, max} ». La signification est la suivante :

- « a* » - Capture le caractère a, zéro ou plusieurs fois
- « a+ » - Capture le caractère a, une ou plusieurs fois
- « a? » - Capture le caractère a, zéro ou une fois
- « a{2,5} » - Capture le caractère a, entre 2 et 5 fois
- « a{5} » - Capture le caractère a, 5 fois
- « a{2,} » - Capture le caractère a, au moins 2 fois
- « a{,5} » - un maximum de cinq « a »s

Un « atome » est

- Une expression régulière entre parenthèses, par ex. « (a|b)+ » s'applique à toute chaîne qui contient au moins un « a » ou « b », dans n'importe quel ordre
- Une paire de parenthèses vide, représente une expression « vide »
- Une expression entre les crochets « [] » (voir ci-dessous)
- Le point « . » remplace n'importe quel caractère, par ex. « .+ » s'applique à toute chaîne qui contient au moins un caractère.
- « ^ » représente le début d'une chaîne de caractères, par ex. « ^a.* » s'applique à une chaîne de caractères qui commence par un « a » et suivi par un caractère quelconque, « a » ou « adkadhashdkash ».
- « \$ » fin d'une chaîne de caractères, fin de ligne
- « \ » suivis par l'un des caractères spéciaux ^ . [\$ () | * + ? { \ correspond au caractère réel sans sa signification spéciale.
- Un caractère écrit, est exactement un caractère normale, par ex. « a » est le caractère alphabétique « a ».

Signification d'une expression entre des crochets, lire la suite.

- « x-y » - Trouvera tout les lettres qui se situe entre « x » et « y », par ex. « [0-9] » correspond à tous les caractères de 0-9, « [a-zA-Z] » correspond à toutes les lettres, qu'elles soient majuscule ou minuscule
- « ^x-y » - Trouvera n'importe quel lettres qui est en dehors des crochets par exemple « [^0-9] » s'applique à toutes les lettres mais *pas* au chiffres
- « [:character-class:] » - Trouvera une classe de caractères *character-class*. Voici les classes de caractères standard : **alnum**, **alpha**, **blank**, **digit**, **lower**, **print**, **punct**, **space**, **upper** et **xdigit**. Donc « [:alpha:] » est utilisé pour toutes les lettres majuscules et minuscules et est identique à « [:lower:][:upper:] ».

Exemples d'expressions régulières

Jetons un coup d'oeil sur quelques exemples :

NUMERIC : Valeur numérique qui est constitué d'au moins de un, mais aussi d'un nombre quelconque de chiffres. Avec le signe « + » la valeur peut être répéter au moins une ou plusieurs

fois à partir d'un nombre, voici un exemple pour obtenir un nombre composé :

```
NUMERIC = '[0-9]+'
```

ou alors

```
NUMERIC = '[:digit:]+'
```

NOBLANK : valeur qui ne contient pas d'espace, les caractères peuvent être quelconque (à l'exception de l'espace), voici un exemple avec divers caractères :

```
NOBLANK = '[^ *]'
```

De plus cette valeur, ne doit pas être vide :

```
NOBLANK = '[^ ]+'
```

IPADDR : Une adresse IP se compose de 4 octet ils sont séparés par un « . » point. Un octet peut être un nombre compris entre 0 et 255. Si nous voulons définir le première octet, il faut.

Avoir un chiffre entre 0 et 9 :	[0-9]
un nombre compris entre 10 et 99 :	[1-9][0-9]
un nombre compris entre 100 et 199 :	1[0-9][0-9]
un nombre compris entre 200 et 249 :	2[0-4][0-9]
Avoir un nombre entre 250 et 255 :	25[0-5]

Suite de la solution, nous allons séparer tout simplement par le signe '|' chaque partie de l'adresse-IPv4 avec l'expression : « [0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5] » ainsi nous avons tous les octets. Cela nous permet désormais d'avoir notre adresse-IPv4 de 4 octets séparé par un point (pour écrire un point vous devez placer un *Backslashes* (ou barre oblique inversée) sinon, il remplacera tout caractère). Vous pouvez voir ci-dessous la syntaxe tiré du fichier exp :

```
OCTET  = '[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5] '
IPADDR = '((RE:OCTET)\.){3}(RE:OCTET)'
```

Aide à la conception des expressions régulières

Si vous voulez créer et tester les expressions régulières, vous pouvez utiliser l'outil regexp pour l'expression rationnelle, qui est situé dans le répertoire **unix** ou **windows** du paquetage « base ». Avec la syntaxe suivante :

```
utilisation de~: regexp [-c <check dir>] <regexp> <string>
```

Courte explication des paramètres :

- **<check dir>** est le répertoire contenant les fichiers de contrôle et de vérification avec les fichiers exp, ainsi « regexp » pourra recourir au expression déjà définie.
- **<regexp>** est une expression régulière (dans le doute, toujours utiliser les guillemets '...' ou "... " ils sont nécessaires si les apostrophes veulent apparaître dans l'expression)
- **<string>** est la chaîne à examiner

Voici quelques exemples :

```
./i586-linux-regex -c ../check '[0-9]' 0
adding user defined regular expression='[0-9]' ('^([0-9])$')
checking '0' against regexp '[0-9]' ('^([0-9])$')
'[0-9]' matches '0'

./i586-linux-regex -c ../check '[0-9]' a
adding user defined regular expression='[0-9]' ('^([0-9])$')
checking 'a' against regexp '[0-9]' ('^([0-9])$')
regex error 1 (No match) for value 'a' and regexp '[0-9]' ('^([0-9])$')

./i586-linux-regex -c ../check IPADDR 192.168.0.1
using predefined regular expression from base.exp
adding IPADDR='((RE:OCTET)\.){3}(RE:OCTET)'
('^(((1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.){3}(1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5]))$')
'IPADDR' matches '192.168.0.1'

./i586-linux-regex -c ../check IPADDR 192.168.0.256
using predefined regular expression from base.exp
adding IPADDR='((RE:OCTET)\.){3}(RE:OCTET)'
('^(((1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.){3}(1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5]))$')
regex error 1 (No match) for value '192.168.0.256' and regexp
'((RE:OCTET)\.){3}(RE:OCTET)'
(unknown:-1) wrong value of variable cmd_var: '192.168.0.256' (invalid ipv4 address)
```

8.3.7. Contrôle détaillé de la configuration

Il est parfois nécessaire d'effectuer des contrôles plus ou moins complexes. Exemple de choses complexe. Avoir une relation entre les paquetages ou avoir une condition à remplir lorsque des variables prennent certaines valeurs. Par ex. lors du choix d'un adaptateur-PCMCIA-ISDN dans le paquetage « pcmcia ».

Pour effectuer ces contrôles, on peut écrire dans le fichier `check/<PAQUETAGE>.ext` (également appelé ext-script) différent petit test. Ce langage se compose des éléments suivants :

1. Mot-clé :

- Contrôle de flux :
 - `if (expr) then statement else statement fi`
 - `foreach var in set_var do statement done`
 - `foreach var in set_var_1 ... set_var_n do statement done`
 - `foreach var in var_n do statement done`
- Relation :
 - `provides package version x.y.z`
 - `depends on package version x1.y1 x2.y2.z2 x3.y3 ...`
- Action :
 - `warning "warning"`
 - `error "error"`
 - `fatal_error "fatal error"`
 - `set var = value`
 - `crypt (variable)`
 - `stat (filename, res)`
 - `fgrep (filename, regex)`

- `split (string, set_variable, character)`
- 2. Type de donnée : chaîne de caractère, nombre entier positif, numéros de version
- 3. Opération logique : `<`, `==`, `>`, `!=`, `!`, `&&`, `||`, `~=`, `copy_pending`, `samenet`, `subnet`

Type de données

C'est-à-dire que les types de données de la variable sont associées à une expression régulière et sont affectés en permanence à un type de données :

- Variable dont le type commence par « NUM » numérique et qui contient un nombre entier positif
- Variable, représentant une N-variable pour tout type de liste, elle est également numériquement
- Toutes les autres variables sont traitées comme des chaînes

Cela signifie, entre autres chose, qu'une variable de type ENUMERIC ne peut *pas* être utilisé comme un indice, lors d'un accès à une liste de variables même si vous avez d'abord vérifié si elle n'était pas vide. Le code suivant ne fonctionnera pas comme prévu :

```
# TEST should be a variable of type ENUMERIC
if (test != "")
then
    # Error: You can't use a non-numeric ID in a numeric
    #         context. Check type of operand.
    set i=my_array[test]
    # Error: You can't use a non-numeric ID in a numeric
    #         context. Check type of operand.
    set j=test+2
fi
```

Voici une solution à ce problème [split](#) (Page 310) :

```
if (test != "")
then
    # all elements of test_% are numeric
    split(test, test_%, ' ', numeric)
    # OK
    set i=my_array[test_%[1]]
    # OK
    set j=test_%[1]+2
fi
```

Substitution de chaîne et de variable

À divers moments les chaînes sont nécessaires, par ex. lorsque vous devez émettre [Attention](#) (Page 306) un avertissement. Certains cas, sont décrits dans cette documentation, une telle chaîne est recherchée pour une variable précise, si elle est trouvée, elle est *remplacé* par son contenu ou par d'autres attributs. Ce remplacement est appelé *substitution de variable*.

Cela va être illustré par un exemple. Supposons cette configuration :

```
# config/base.txt
HOSTNAME='fli41'
# config/dns_dhcp.txt
```

```
HOST_N='1' # Nombre d'hôtes
HOST_1_NAME='client'
HOST_1_IP4='192.168.1.1'
```

Ensuite, les chaînes de caractères sont réécrits comme ceci, si la substitution de variable est actif dans ce contexte :

```
"Mon routeur s'appel $HOSTNAME"
# --> "Mon routeur s'appel fli4l"
"HOSTNAME fait partie du paquetage %{HOSTNAME}"
# --> "HOSTNAME fait partie du paquetage base"
"@HOST_N est $HOST_N"
# --> "Nombre d'hôte est 1"
```

Comme vous pouvez le voir, il y a essentiellement trois options pour le remplacement :

- `$<Name>` ou bien `${<Name>}` : remplace le nom de la variable par le contenu de la variable. C'est la forme la plus courante de substitution. Le nom doit être enfermé dans `{...}` si la chaîne est directement suivie d'un caractère qui peut être une partie valide d'un nom de variable soit une lettre, un chiffre ou un tire souligné. Dans tous les autres cas, l'utilisation des accolades est possible, mais pas obligatoire.
- `%<Name>` ou bien `%{<Name>}` : remplace le nom de la variable par le nom du paquetage, dans lequel la variable est définie. Cela ne fonctionne *pas* dans le script via la variable affectées `set` (Page 306) ou la variable de contrôle de boucle `foreach` (Page 312), étant donné que ces variables ne sont pas dans un paquetage, leur syntaxe est différent.
- `@<Name>` ou bien `@{<Name>}` : remplace le nom de la variable par le commentaire qui est après la variable dans la configuration. Encore une fois, cela n'a pas de sens pour les variables définies par le script.

Remarque : Les éléments de liste de variables ne peuvent *pas* être intégré dans des chaînes de caractères de cette façon, parce qu'il n'y a aucune possibilité de fournir un index.

En général, seule une *constante* peut être utilisé pour la substitution de variable, les chaînes qui proviennent d'une variable restent inchangées. Un exemple permettra de clarifier cela - voici la configuration suivante :

```
HOSTNAME='fli4l'
TEST='${HOSTNAME}'
```

Ensuite, le code :

```
warning "${TEST}"
```

Produit la sortie suivante :

```
Warning: ${HOSTNAME}
```

Et *pas* la sortie :

```
Warning: fli4l
```

Dans les sections suivantes, on notera explicitement les conditions des chaînes qui font l'objet d'une substitution de variable.

Définition d'un service avec un numéro de version associé : provides

Cela permet par exemple un OPT peut déclarer et mettre à disposition un serveur d'impression ou un service Web. Un seul paquetage, fourni un certain service. Cela empêche par exemple d'installer deux serveurs web en parallèle, cela ne fonctionnerais pas pour des raisons évidentes, puisque les deux serveurs utiliseraient le port 80. En outre, la version actuelle du service est prévu pour est mise à jour régulièrement. Le numéro de version se compose de deux ou trois nombres séparés par des points, comme « 4.0 » ou « 2.1.23 ».

Les services proviennent généralement de l'OPT, pas du paquetage. Par exemple, dans le paquetage « tools » il y a une série de programmes, qui ont chacun leur propre instruction `provides` elle sera activée via `OPT_...='yes'`.

La syntaxe est :

```
provides <Name> version <Version>
```

Exemple avec le paquetage « easycron » :

```
provides cron version 3.10.0
```

Le numéro de version doit être incrémenté par le développeur de l'OPT dans le troisième volet, si les fonctionnalités ont été seulement améliorations et que l'interface est toujours compatible avec l'OPT. Le numéro de version doit être augmentée avec le premier ou le deuxième chiffre, si l'interface a été modifiée et est incompatible (par exemple, en raison de variables renommées, des chemins changés, manquant ou le programme de service a été renommés, etc.).

Définition une dépendance à un service avec une Version spécifique : depends

Si un autre service est nécessaire pour assurer la fonction de votre propre service (par exemple un serveur web), on peut définir une dépendance par une version spécifique pour le service. La version peut être indiquée par deux (par ex. « 2.1 ») ou trois chiffres (par ex. « 2.1.11 »), la version à deux chiffres accepte toutes les versions à partir de ce nombre et la version à trois chiffres accepte seulement la version spécifié. En outre, vous pouvez spécifier une liste de numéros de version si plusieurs versions du service est compatible avec le paquetage.

La syntaxe est la suivante :

```
depends on <Name> version <Version>+
```

Exemple : le paquetage « serveur » contient :

```
provides server version 1.0.1
```

Avec le paquetage « client ». vous indiquez l'instruction `depends` dans cette exemple²

```
depends on server version 1.0      # OK, '1.0' s'adapte '1.0.1'
depends on server version 1.0.1    # OK, '1.0.1' s'adapte '1.0.1'
depends on server version 1.0.2    # Erreur, '1.0.2' s'adapte pas '1.0.1'
depends on server version 1.1      # Erreur, '1.1' s'adapte pas '1.0.1'
depends on server version 1.0 1.1  # OK, '1.0' s'adapte '1.0.1'
depends on server version 1.0.2 1.1 # Erreur, ni '1.0.2' ni '1.1' ne s'adapte '1.0.1'
```

2. bien sûr, un seul à la fois !

Message pour l'utilisateur : warning, error, fatal_error

Avec l'aide de ces trois fonctions, on peut avertir et signaler l'utilisateur, d'une erreur ou interrompre immédiatement le traitement de contrôle. La syntaxe est la suivante :

- `warning "text"`
- `error "text"`
- `fatal_error "text"`

Toutes les chaînes utilisées avec ces fonctions sont soumis à une [substitution de variable](#) (Page 303).

Affectation

Si vous avez besoin d'utiliser une variable temporaire pour une raison quelconque, vous pouvez la créer avec « `set var [= value]` ». *La variable ne peut pas être une variable de configuration!*³ Si vous omettez « `= value` » la variable sera alors simplement placée sur « `yes` », ensuite vous pouvez tester facilement la variable avec l'instruction `if`. Vous pouvez spécifier les données suivant après le signe égal, variable normale, variable indexée, nombre, chaîne de caractère, versions.

Il convient de noter de par l'assignation le *type* sera défini dans la variable temporaire. Si un numéro est attribué `mkfli4l` se « souviendra » que la variable contient un numéro et permettra plus tard de faire un calcul avec celui-ci. Si vous essayez de faire un calcul avec une variable de type différent, il échouera. Exemple :

```
set i=1    # OK, i est une variable numérique
set j=i+1  # OK, j est une variable numérique et contient la valeur 2
set i="1"  # OK, i est maintenant une variable de chaîne
set j=i+1  # Erreur "Vous ne pouvez pas utiliser un ID non numérique dans un
           #          contexte numérique. Vérifiez le type de l'opérande."
           # --> Pas de calculs avec des chaînes~!
```

Vous pouvez également créer des listes temporaires (voir ci-dessous). Exemple :

```
set prim_%[1]=2
set prim_%[2]=3
set prim_%[3]=5
warning "${prim_n}"
```

Le nombre de liste d'éléments est géré par `mkfli4l` et par la variable `prim_n`. Le code ci-dessus conduit donc à la sortie suivante :

```
Warning: 3
```

Si le côté droit de la cession est une constante de chaîne, elle est soumise à une [substitution de variable](#) (Page 303). L'exemple suivant montre le code :

```
set s="a"
set v1="$s" # v1="a"
set s="b"
```

3. C'est une restriction souhaitée : le contrôle du skript ne sera *pas* en mesure de modifier la configuration de l'utilisateur.

```

set v2="$s" # v2="b"
if (v1 == v2)
then
    warning "égal"
else
    warning "pas égal"
fi

```

La sortie produite n'est « pas égale », parce que les variables `v1` et `v2` sont remplacées par le contenu de la variable `s` déjà en cours de cession.

Remarque : Un ensemble de variables dans un script est visible lors du traitement des autres scripts - actuellement il n'existe pas de principe de localisation pour ces variables introduites. L'ordre dans lequel les scripts sont traités dans divers paquetages, n'est pas définie, vous ne devez pas compter sur une variable ayant des valeurs définies dans un autre paquetage.

Liste (ou Tableau)

Si vous voulez accéder aux éléments d'une %-variable (ou de la liste), vous devez utiliser le nom original de la variable comme mentionné dans le fichier `check/<PAQUETAGE>.txt` et d'ajouter un index pour chaque signe « % » en utilisant « *[Index]* ».

Exemple : Si vous voulez accéder aux éléments de la variable `PF_USR_CHAIN_%_RULE_%` vous avez besoin de deux index car la variable a deux signes « % ». Tous les éléments peuvent être enregistrés par exemple en utilisant la (boucle `foreach` et [voir ci-dessous](#) (Page 312)) :

```

foreach i in pf_usr_chain_n
do
    # un seul index nécessaire, seul un '%' dans la variable
    set j_n=pf_usr_chain_%_rule_n[i]
    # Attention: a
    # foreach j in pf_usr_chain_%_rule_n[i]
    # n'est pas possible, d'où l'utilisation de j_n~!
    foreach j in j_n
    do
        # deux index nécessaires, deux '%' dans la variable
        set rule=pf_usr_%_rule_%[i][j]
        warning "Rule $i/$j: ${rule}"
    done
done

```

Exemple de configuration

```

PF_USR_CHAIN_N='2'
PF_USR_CHAIN_1_NAME='usr-chain_a'
PF_USR_CHAIN_1_RULE_N='2'
PF_USR_CHAIN_1_RULE_1='ACCEPT'
PF_USR_CHAIN_1_RULE_2='REJECT'
PF_USR_CHAIN_2_NAME='usr-chain_b'
PF_USR_CHAIN_2_RULE_N='1'
PF_USR_CHAIN_2_RULE_1='DROP'

```

la sortie suivante est imprimé :

```
Warning: Rule 1/1: ACCEPT
Warning: Rule 1/2: REJECT
Warning: Rule 2/1: DROP
```

Alternativement, vous pouvez parcourir directement toutes les valeurs du tableau, mais les indices exacts ne sont pas toujours connus (car se n'est pas nécessaire) :

```
foreach rule in pf_usr_chain_%_rule_%
do
    warning "Rule %{rule}='${rule}'"
done
```

Cela produit la sortie suivante avec l'exemple de configuration à partir des informations ci-dessus :

```
Warning: Rule PF_USR_CHAIN_1_RULE_1='ACCEPT'
Warning: Rule PF_USR_CHAIN_1_RULE_2='REJECT'
Warning: Rule PF_USR_CHAIN_2_RULE_1='DROP'
```

Le deuxième exemple montre bien le sens de la syntaxe `%<Name>` : la chaîne `%rule` est substitué par le *nom* de la variable en question (par exemple `PF_USR_CHAIN_1_RULE_1`), tandis que `$rule` est substitué par son *contenu* (exemple `ACCEPT`).

Crypter un mot de passe : `crypt`

Dans le fichier `rc.cfg` certaines variables contiennent des mots de passe qui ne doivent pas apparaître en clair. Ces variables peuvent être cryptées en utilisant `crypt` et seront transférées dans un format qui est également utilisé par le routeur. On utilise pour cela :

```
crypt (<variable>)
```

La fonction `crypt` est *seulement* un endroit où la variable de configuration peut être modifiée.

Contrôle des propriétés d'un fichier : `stat`

`stat` vous permet de rechercher les propriétés d'un fichier. Il indique pour l'instant que la taille du fichier. Si vous souhaitez tester des fichiers de configuration dans le répertoire courant, vous pouvez utiliser la variable interne `config_dir`. La syntaxe est :

```
stat (<nom de fichier>, <clé>)
```

La commande ressemble à ceci (les paramètres utilisés ne sont que des exemples) :

```
foreach i in openvpn_%_secret
do
    stat("${config_dir}/etc/openvpn/${i}.secret", keyfile)
    if (keyfile_res != "OK")
    then
        error "OpenVPN: missing secretfile <config>/etc/openvpn/${i}.secret"
    fi
done
```

L'exemple suivant vérifie si un fichier existe dans le répertoire de configuration actuelle. Si la variable `OPENVPN_1_SECRET='test'` est définie dans le fichier de configuration, lors du premier contrôle d'exécution la boucle vérifiera l'existence du fichier `etc/openvpn/test.secret` dans le répertoire de configuration actuelle.

Après l'exécution deux variables sont définies :

- `<Clé>_res` : Résultat après l'exécution système `stat` (« OK », si l'exécution système est réussie, sinon il y aura message d'erreur de l'exécution système)
- `<Clé>_size` : taille du fichier

Cela pourrait alors ressembler à ceci :

```
stat ("unix/Makefile", test)
if ("${test_res}" == "OK")
then
    warning "test_size = $test_size"
else
    error "Error '${test_res}' while trying to get size of 'unix/Makefile'"
fi
```

Un nom de fichier passé comme une constante de la chaîne est soumis à une [substitution de variable](#) (Page 303).

Rechercher dans les fichiers : `fgrep`

Si vous voulez rechercher un fichier avec « `grep` », ⁴ vous avez aussi la possibilité d'utiliser la commande `fgrep`, la syntaxe est :

```
fgrep (<nomdefichier>, <regex>)
```

Si le fichier `<nom de fichier>` n'existe pas alors `mkfli41` s'arrête et affiche une erreur fatale ! Si vous n'êtes pas sûr que le fichier est toujours présent, vous devez avant utiliser la commande `stat` pour savoir si le `<nom de fichier>` existe. Après avoir exécuté `fgrep` le résultat de recherche sera présent dans un tableau et `FGREP_MATCH_%` sera disponible, avec l'indice *x* comme d'habitude vous allez avoir `FGREP_MATCH_N`. `FGREP_MATCH_1` fait référence à toute les lignes l'expression régulière correspondantes, tandis que `FGREP_MATCH_2` pour `FGREP_MATCH_N` contient la *n-1* partie entre parenthèses.

Un premier exemple simple vous montrera comment l'utiliser. Le fichier `opt/etc/shells` contient la ligne :

```
/bin/sh
```

Le code est le suivant

```
fgrep("opt/etc/shells", "~/(.)(.*)/")
foreach v in FGREP_MATCH_%
do
    warning "%v='$v'"
done
```

Produit la sortie suivante :

4. « `grep` » est une commande du système d'exploitation Unix, pour les flux de texte de filtrage.

```
Warning: FGREP_MATCH_1='/bin/'
Warning: FGREP_MATCH_2='b'
Warning: FGREP_MATCH_3='in'
```

Le RegEx correspond (seulement) à « /bin/ », (seul) cette partie de ligne est contenue dans la variable `FGREP_MATCH_1`. La première partie entre les parenthèses de l'expression correspond au premier caractère après le premier signe « / », c'est pourquoi « b » est contenu dans `FGREP_MATCH_2`. La deuxième partie restante contient entre les parenthèses « b » jusqu'au dernière signe « / », donc « in » est dans la variable `FGREP_MATCH_3`.

Le deuxième exemple suivant vous montrer une utilisation pratique de `fgrep` avec le fichier `check/base.ext`. Il sera testé si toutes les références `tmpl:` indiquées, sont vraiment présent dans `PF_FORWARD_x` :

```
foreach n in pf_forward_n
do
  set rule=pf_forward_%[n]
  if (rule =~ "tmpl:([[:space:]]+)" )
  then
    foreach m in match_%
    do
      stat("$config_dir/etc/fwrules.tmpl/$m", tmplfile)
      if(tmplfile_res == "OK")
      then
        add_to_opt "etc/fwrules.tmpl/$m"
      else
        stat("opt/etc/fwrules.tmpl/$m", tmplfile)
        if(tmplfile_res == "OK")
        then
          add_to_opt "etc/fwrules.tmpl/$m"
        else
          fgrep("opt/etc/fwrules.tmpl/templates", "^$m[[:space:]]+")
          if (fgrep_match_n == 0)
          then
            error "Can't find tmpl:$m for PF_FORWARD_${n}='$rule'!"
          fi
        fi
      fi
    done
  fi
done
```

La valeur du nom de fichier ainsi que l'expression régulière passée comme une constante de chaîne sont soumis à une [substitution de variable](#) (Page 303).

Désassemblage des paramètres : `split`

Souvent plusieurs paramètres sont appliqués dans une variable, ensuite dans le script de démarrage ces paramètres sont désassemblés séparément. Si vous voulez effectuer des tests lors du désassemblage, c'est la commande `split` qu'il vous faut. La syntaxe est :

```
split (<Chaîne>, <Liste>, <Séparateur>)
```

Une chaîne peut être spécifiée par une variable ou directement en tant que constante. `mkfli41` décompose la chaîne là où apparaît un séparateur et génère un élément pour chaque partie de la liste. Vous pouvez parcourir ces éléments plus tard et effectuer des tests. Si rien n'est trouvé entre deux séparateurs un élément de la liste est générée comme une valeur pour une chaîne vide. L'exception « » est : tous les espaces sont supprimés et aucune variable vide n'est créée.

Lors de la décomposition des éléments, si un contexte numérique apparaît dans la variable (par exemple sous forme d'indice) cela doit être précisé dans la commande `split`. Vous devez ajouter un attribut supplémentaire 'numéric'. Cette exécution se présente comme ceci :

```
split (<Chaîne>, <Liste>, <Séparateur>, numeric)
```

Voici un exemple :

```
set bar="1.2.3.4"
split (bar, tmp_%, '.', numeric)
foreach i in tmp_%
do
    warning "%i = $i"
done
```

Produit la sortie suivante :

```
Warning: TMP_1 = 1
Warning: TMP_2 = 2
Warning: TMP_3 = 3
Warning: TMP_4 = 4
```

Remarque : si vous utilisez une variable « numeric » `mkfli41` ne vérifiera *pas* les parties de chaîne générées si elle ne sont pas vraiment numérique ! Si vous utilisez une telle construction dans un contexte numérique `mkfli41` déclenchera une erreur fatale si une telle variable n'est pas numérique. Exemple :

```
set bar="a.b.c.d"
split (bar, tmp_%, '.', numeric)
# Fehler: invalid number 'a'
set i=tmp_%[1]+1
```

Une valeur utilisée dans le première paramètre de la constant de chaîne est soumis à une [substitution de variable](#) (Page 303).

Ajouter des fichiers dans l'archive : `add_to_opt`

Avec la fonction `add_to_opt` des fichiers supplémentaires peuvent être ajoutés dans une archive OPT ou dans le RootFS. Il est possible de sélectionner *tous* les fichiers du sous-répertoire `opt/` ou à partir du répertoire de configuration. Il n'y a pas de restriction sur l'ajout de fichiers dans un paquetage. Si un fichier doit être à la fois dans `opt/` et dans le répertoire de configuration, `add_to_opt` choisira de copier les fichiers dans le répertoire de configuration. La fonction `add_to_opt` est complexe et est en règle général logique, la fonction décide quel fichier sera copié en premier dans l'archive.

La syntaxe est la suivante :

```
add_to_opt <nom de fichier> [<Flags>]
```

Le `Flags` est optionnel. Les valeurs par défaut de la table 8.2 sont utilisés si aucun `Flags` n'est indiqué.

Ci-après un exemple à partir du paquetage « `sshd` » :

```
if (opt_sshd)
then
  foreach pkf in sshd_public_keyfile_%
  do
    stat("$config_dir/etc/ssh/$pkf", publickeyfile)
    if(publickeyfile_res == "OK")
    then
      add_to_opt "etc/ssh/$pkf" "mode=400 flags=utxt"
    else
      error "sshd: missing public keyfile %pkf=$pkf"
    fi
  done
fi
```

Utiliser d'abord `stat` (Page 308) pour vérifier si le fichier existe bien dans le répertoire config. Si le fichier existe, il sera ajouté à l'archive, sinon `mkfli41` renvoie un message d'erreur.

Remarque : `mkfli41` vérifie (Page 294) aussi avec la fonction `add_to_opt` si le fichier à copier, se trouve bien dans le répertoire config.

Les noms de fichiers et des `Flags` qui sont utilisés en tant que constantes de chaîne sont soumis à une [substitution de variable](#) (Page 303).

Contrôle de flux

```
if (expr)
then
    statement
else
    statement
fi
```

Un cas classique de restriction, comme nous connaissons. Si la condition est vraie, alors, l'expression `then` est exécutée, si la condition est fausse, l'expression `else` sera exécutée.

Si vous voulez effectuer des tests de %-variable, il faudra tester chaque variable. Pour éviter cela, il y a la boucle `foreach` en deux variantes.

1. Itération sur une liste de variables :

```
foreach <contrôle variable> in <Liste-Variable>
do
    <instruction>
done

foreach <contrôle variable> in <liste-Variable-1> <liste-Variable-2> ...
do
    <instruction>
done
```


Cette boucle parcourt tous le tableau de variables spécifié, en commençant par le premier et le dernier élément, le nombre d'éléments de cette liste est extraites du N-variable associée à ce tableau. La contrôle variable prend les valeurs des variables du tableau respectifs. Il est à noter vous pouvez ajouter un processus optionnel pour les variables du tableau si une valeur n'est pas présents dans la configuration, un élément vide sera généré. Vous pourriez en tenir compte dans le script, par exemple comme ceci :

```
foreach i in template_var_opt_%
do
    if (i != "")
    then
        warning "%i is present (%i='$i')"
    else
        warning "%i is undefined (empty)"
    fi
done
```

Comme vous pouvez le voir dans l'exemple, le *nom* des variables du tableau respectives peut être déterminée avec l'instruction `%<contrôle variable>`.

L'instruction dans la boucle ci-dessus peut être l'un des éléments de contrôle ou de fonctions (`if`, `foreach`, `provides`, `depends`, ...).

Si vous souhaitez accéder exactement à un élément du tableau, vous pouvez y remédier en utilisant la syntaxe `<Liste>[<Index>]`. L'index peut être une variable normale, une constante numérique ou encore un tableau indexé.

2. Itération sur N-variables :

```
foreach <contrôle variable> in <N-Variable>
do
    <instruction>
done
```

Cette boucle s'exécute de 1 à la valeur qui est indiqué dans la N-variable. Vous pouvez utiliser contrôle variable dans un tableau de variables indexé. Donc, si vous voulez parcourir non seulement un tableau de variable, mais plusieurs tableaux de variables en même temps, tous contrôlés par la *même* N-variable, vous prenez la variante de boucle et vous utilisez le contrôle variable pour l'indexation de plusieurs tableaux de variables. Exemple :

```
foreach i in host_n
do
    set name=host_%_name[i]
    set ip4=host_%_ip4[i]
    warning "$i: name=$name ip4=$ip4"
done
```

Le résultat du contenu du tableau de la liste `HOST_%_NAME` et de la liste `HOST_%_IP4` pour cet exemple :

```
Warning: 1: name=berry ip4=192.168.11.226
Warning: 2: name=fence ip4=192.168.11.254
Warning: 3: name=sandbox ip4=192.168.12.254
```

Expressions

Une expression est liée à une valeur et un opérateur à une autre valeur. Cette valeur peut être une variable normale, un élément d'un tableau, ou une constante (nombre, chaîne de caractère ou numéro de version). Toutes les constantes de chaîne dans les expressions sont soumises à une [substitution de variable](#) (Page 303).

Les opérateurs vous permettent de faire à peu près tout ce que vous attendez d'un langage de programmation. Un test pour l'égalité de deux variables pourrait donc ressembler à ceci :

```
var1 == var2
"$var1" == "$var"
```

À noter, que la comparaison selon le type de variables se fait dans le fichier `check/<PAQUETAGE>.txt` où ils ont été créés. Si l'une des deux variables est [numérique](#) (Page 303), la comparaison se fait sur une base numérique, c'est-à-dire que les chaînes de caractère sont converties en nombres, puis comparées. La comparaison est fondée par une chaîne, si on compare `"05" == "5"` cela donne un résultat « faux », une comparaison `"18" < "9"` donne « vrai » selon l'ordre lexicographique des chaînes de caractère : le chiffre « 1 » précède le chiffre « 9 » dans le jeu de caractères ASCII.

Pour introduire la comparaison des numéros de version de construction vous devez utiliser `numeric(version)`, cela génère la valeur numérique du numéro de version à des fins de comparaison. Ici on applique :

```
numeric(version) := major * 10000 + minor * 1000 + sub
```

« major » représente la première partie, « minor » la deuxième partie et « sub » la troisième partie du numéro de version. Si « sub » est manquant le nombre dans l'addition ci-dessus sera omis (en d'autres termes « sub » sera égale à zéro).

Vous trouverez dans le tableau 8.3 une liste complète de toutes les expressions. « val » indique une valeur de n'importe quel type, « number » une valeur numérique et « string » une chaîne de caractère.

Opérateur « match »

Avec l'opérateur-match `==~` vous pouvez vérifier, si une expression régulière correspond à la valeur d'une variable. En outre, l'opérateur peut également l'utiliser pour extraire une partie de l'expression de la variable. Après avoir appliqué avec succès une expression régulière à une variable, une table `MATCH_%` contiendra toutes les parties trouvées de la variable. Cela pourrait par exemple ressembler à ceci :

```
set foo="foobar12"
if ( foo ==~ "(foo)(bar)([0-9]*)" )
then
    foreach i in match_%
    do
        warning "match %i: $i"
    done
fi
```

Après une exécution de `mkfli41`, cela conduira vers la sortie suivante :

TABLE 8.3.: Expressions logiques

expression	vraie si
id	id == « yes »
val == val	valeurs de type identique sont égaux
val != val	valeurs de type identique sont inégaux
val == number	numérique la valeur de val == nombre
val != number	numérique la valeur de val != nombre
val < number	numérique la valeur de val < number
val > number	numérique la valeur de val > number
val == version	numeric(val) == numeric(version)
val < version	numeric(val) < numeric(version)
val > version	numeric(val) > numeric(version)
val =~ string	val correspond à une chaîne expression régulière
(expr)	l'expression entre les parenthèses est vraie
expr && expr	les deux expressions sont vraies
expr expr	au moins une des deux expressions est vraie
copy_pending(id)	voir la description ci-dessous
samenet (string1, string2)	chaîne1 décrit le même réseau que la chaîne2
subnet (string1, string2)	chaîne1 décrit un sous-réseau de chaîne2

```
Warning: match MATCH_1: foo
Warning: match MATCH_2: bar
Warning: match MATCH_3: 12
```

Lors de l'utilisation de =~ on peut faire référence à toutes les expressions régulières existantes. Si l'on veut par exemple vérifier, si le pilote de la carte Ethernet-PCMCIA a été sélectionné, sans que la variable OPT_PCMCIA soit sur le paramétrée sur « yes », vous pouvez écrire :

```
if (!opt_pcmcia)
then
  foreach i in net_drv_%
  do
    if (i =~ "^(RE:PCMCIA_NET_DRV)$")
    then
      error "If you want to use ..."
    fi
  done
fi
```

Comme démontré dans l'exemple, il est important *d'ancrer* l'expression régulière avec ^ et \$, si vous avez l'intention d'appliquer une expression dans la variable *entière*. Sinon, l'expression renvoie une valeur « vrai » si une *partie* de la variable est couverte par l'expression régulière, ce qui n'est certainement pas souhaitable dans ce cas.

Vérifier le fichier copié, en fonction de la valeur d'une variable : copy_pending

Avec le processus de vérification `copy_pending` vous pouvez vérifier si le fichier a été copié ou non en fonction de la valeur d'une variable. On peut l'utiliser, par exemple pour tester un

pilote spécifié par l'utilisateur, pour savoir s'il existe bien et s'il a bien été copié. `copy_pending` accepte les noms à tester, sous forme d'une variable ou d'une chaîne.⁵ `copy_pending` vérifie si

- la variable est active (si l'opt est chargé, la variable-OPT doit être placée sur « yes »),
- la variable a été référencée dans le fichier `opt/<PAQUETAGE>.txt`, et si
- le fichier a été copié en fonction de la valeur indiquée.

la fonction `copy_pending` renverra « vrai », s'il ne détecte emphpas le fichier à copier lors de la dernière étape, le processus de copie sera donc (encore en « attente »).

Vous pouvez trouver un petit exemple de l'utilisation de toutes ces fonctions dans le fichier `check/base.ext` :

```
foreach i in net_drv_%
do
    if (copy_pending("%i"))
    then
        error "No network driver found for %i='$i', check config/base.txt"
    fi
done
```

Tous les éléments de la liste `NET_DRV_%` seront détectés pour lesquelles aucune copie n'a été faite car il n'existe pas de configuration correspondante dans le fichier `opt/base.txt`.

Comparer des adresses réseaux avec : `samenet` et `subnet`

Pour vérifier la communication entre les réseaux, nous avons besoin d'un test, pour savoir si les deux réseaux sont identiques ou si l'un des deux est un sous-réseau donc différent. Pour cela, vous avez deux fonctions `samenet` et `subnet` qui vous permet de vérifier le réseau.

```
samenet (netz1, netz2)
```

Le retour est « vrai », si les deux réseaux sont identiques, et

```
subnet (netz1, netz2)
```

Le retour est « vrai », si « netz1 » est un sous-réseau de « netz2 ».

Extension du Kernel par ligne de commande

Obligatoire pour certain OPT, elle est utilisée pour rajouter d'autres paramètres dans le Kernel lors du Boot, on peut contrôler la variable `KERNEL_BOOT_OPTION`, pour savoir si les valeurs ont bien été incluses et au cas échéant un message d'avertissement ou d'erreur sera envoyé. Maintenant avec la variable interne `KERNEL_BOOT_OPTION_EXT` vous pouvez ajouter une option nécessaire mais manquante directement dans le script-ext. Un exemple tiré du fichier `check/base.ext` :

```
if (powermanagement =~ "apm.*|none")
then
    if ( ! kernel_boot_option =~ "acpi=off")
```

5. Comme décrit dans la chaîne des objet de substitution de variable, c'est à dire via la [boucle foreach](#) (Page 312) et le [%<Name> de substitution](#) (Page 303) tous les éléments de la liste (ou un tableau) peuvent être examinées.

```

then
    set kernel_boot_option_ext="${kernel_boot_option_ext} acpi=off"
fi
fi

```

Le paramètre « `acpi=off` » sera transmis au Kernel si aucune gestion d'alimentation ou aucun type « d'APM » n'est nécessaire.

8.3.8. Supporte différent choix de version du Kernel

Les différents choix de version du Kernel se distinguent souvent de quelques détails :

- La modification des pilotes disponible, certains ont disparu, d'autres ont été ajoutés.
- Une partie des modules ont un nom différent
- Certains modules ont un aspect différent
- Les modules se trouvent sur d'autres emplacements

Ces différences sont en grande partie traitées automatiquement par `mkfli41`. Pour définir ces modules disponibles, vous pouvez d'une part, les tester et examiner en fonction de la version les ([expressions régulières conditionnelles](#) (Page 299)) d'autre part, cela vous permet avec `mkfli41` d'utiliser le fichier `opt/<PAQUETAGE>.txt` selon la version utilisée. Ils seront ensuite renommés avec un tiret bas `opt/<PAQUETAGE>_<Kernel-Version>.txt`, ainsi, les composants selon la version du Kernel seront séparés les uns des autres. Le fichier du paquetage « base » sera dans le répertoire `opt` :

- `base.txt`
- `base_3_14.txt`
- `base_3_17.txt`

Le premier fichier (`base.txt`) est *toujours* traité. Les deux autres fichiers sont traités si la version du Kernel « `3.14(*)` » ou « `3.17(*)` » est utilisée. Comme on peut le voir, certaines parties de la version peuvent être omises dans le nom du fichier, si vous avez un groupe de Kernels les numéros de version seront « écrasés ». En supposant que l'on utilise la `VERSION_KERNEL='3.14.26'` les fichiers suivants (si existant) seront lus et traités pour plusieurs installations :

- `<PAQUETAGE>.txt`
- `<PAQUETAGE>_3.txt`
- `<PAQUETAGE>_3_14.txt`
- `<PAQUETAGE>_3_14_26.txt`

8.3.9. Documentation

Les fichiers de la documentation sont placés dans

- `doc/<LANGUE>/opt/<PAQUETAGE>.txt`
- `doc/<LANGUE>/opt/<PAQUETAGE>.html`

Les fichiers HTML peuvent également être divisés, c'est-à-dire être inclus dans chaque OPT différent. Il faut tout de même qu'un `<PAQUETAGE>.html`, soit le fichier référent pour tous les autres. Les modifications d'un paquetage doivent être documentées dans l'un des fichiers suivants :

- `changes/<PAQUETAGE>.txt`

L'ensemble de la documentation, ne doit pas contenir des tabulateurs et doit contenir un maximum de 79 caractères après un retour à la ligne. Vous devez vous assurer que la documentation pourra être lue correctement avec un éditeur sans retour de ligne automatique.

La documentation peut être produite dans le format \LaTeX et ensuite être transformé en format HTML et PDF. À titre d'exemple, cette documentation peut servir à fli4l. Dans la documentation qui se trouve dans le paquetage « template » traite du minimum requis pour les Macros- \LaTeX . Vous pouvez voir une brève description générale dans les paragraphes suivantes.

La documentation de fli4l est actuellement disponible dans les langues suivantes : allemande, anglaise ($\langle \text{LANGUE} \rangle = \text{« english »}$) et française ($\langle \text{LANGUE} \rangle = \text{« french »}$). C'est le responsable du paquetage qui prend la décision pour documenter son paquetage dans n'importe quelle langue. Pour plus de clarté, il est recommandé de créer une documentation en allemand et/ou en anglais (idéalement dans les deux langues).

Conditions pour créer une Documentation- \LaTeX

Pour créer des documents à partir des sources- \LaTeX , vous avez certaines exigences à respecter relatives à l'environnement :

- Sous l'environnement Linux/OS X : Pour faciliter la production de la documentation, vous avez le programme Makefile, avec celui-ci toutes les commandes sont automatisées (Cygwin peut également fonctionner, mais n'est pas testée par l'équipe fli4l)
- Installer LaTeX2HTML pour les versions HTML
- Bien sûr pour le \LaTeX (« TeX Live » pour Linux/OS X et « MiKTeX » pour Microsoft Windows sont recommandés) avec le programme « pdftex » et les paquets \TeX suivant :
 - L'actuel Script-KOMA (au moins la version 2)
 - Tous les paquets nécessaires pour pdftex
 - Le paquetage décompressé de la documentation fli4l fournit le makefile et les styles- \TeX

Nom de fichier

Les fichiers de la documentation sont désignés selon le schéma suivant :

$\langle \text{PAQUETAGE} \rangle_main.tex$: ce fichier contient la partie principale de la documentation. $\langle \text{PAQUETAGE} \rangle$ indique le nom du paquetage, il doit être écrit (en minuscules).

$\langle \text{PAQUETAGE} \rangle_appendix.tex$: si vous souhaitez ajouter des commentaires supplémentaires au sujet du paquetage, ils seront placés ici dans l'annexe.

Ces fichiers sont stockés dans le répertoire `fli4l/⟨PAQUETAGE⟩/doc/⟨LANGUE⟩/tex/⟨PAQUETAGE⟩`. Vous pouvez voir ci-dessous un exemple du paquetage « sshd » :

```
$ ls fli4l/doc/deutsch/tex/sshd/
Makefile sshd_appendix.tex  sshd_main.tex  sshd.tex
```

Makefile est chargé de générer la documentation, le fichier `sshd.tex` fournit la structure pour la documentation actuelle et son annexe, qui est situé dans les deux autres fichiers. Vous pouvez consulter des exemples dans la documentation du paquetage « template ».

Principes de base du L^AT_EX

L^AT_EX fonctionne un peu comme HTML « orienté balise », seulement les balises sont appelées ici « commandes », le format est le suivant : `\commande` ou `\begin{environnement} ... \end{environnement}`.

Dans la mesure du possible, vous devez utiliser ces commandes qui accentuent *l'importance* du texte et moins sa *présentation*. Il est donc avantageux de les utiliser par exemple.

`\warning{ne_fait_..._pas}`

Au lieu d'utiliser

`\emph{ne_fait_..._pas}`

cette commande.

Chaque commande ou environnement peut absorber plusieurs paramètres supplémentaires, on peut écrire `\commande{paramètre1}{paramètre2}{paramètreN}`.

Certaines commandes ont des paramètres optionnels (au lieu des accolades) pour fermer la commande vous utilisez les crochets : `\kommando[optionalerParameter]{parameter1} ...`. Habituellement, un seul paramètre optionnel est utilisé, dans des cas plus rares, il peut y en avoir plus.

Dans le document, certains paragraphes sont séparés par des lignes blanches. L^AT_EX gardera ces sauts de ligne pour séparer les paragraphes dans le texte.

Les caractères suivants ont une signification spéciale dans L^AT_EX ils doivent être précédés du caractère `\` dans le texte pour être écrit normalement : `# $ & _ % { }`. Le caractère « ~ » et « ^ », doit être écrit comme ceci : `\verb?~? \verb?^?`.

Les principales commandes L^AT_EX sont expliquées dans la documentation du paquetage « template ».

8.3.10. Formats de fichier

Dans le paquetage tous les fichiers texte (la documentation et les scripts d'installation qui sont sur le routeur) doivent être placés au format DOS, c'est à dire avec un CR/LF au lieu d'un simplement fin de ligne LF. Cela garantit que les utilisateurs Windows pourront également lire la documentation avec « Notepad » (ou bloc-notes), et pourront modifier les scripts sous Windows, ensuite ils seront toujours susceptibles de fonctionner sur le routeur. Les Scripts sont convertis à la construction des archives dans le format utilisé par le routeur (voir la description des flags dans le tableau 8.2).

8.3.11. Développer la documentation

Si vous devez définir un programme pour une nouvelle interface, à partir d'un paquetage et qui sera utilisé par d'autres programmes, la documentation de cette interface, sera séparée du reste de la documentation et se trouvera dans `doc/dev/<PAQUETAGE>.txt`.

8.3.12. Programme-Client

Si vous ajoutez un programme-client pour un paquetage supplémentaire, il sera placé dans le répertoire `windows/` pour les clients-Windows et dans le répertoire `unix/` pour les clients-Unix et Linux.

8.3.13. Code source

Les Programmes personnalisés et les codes sources peuvent être récupéré dans le répertoire `src/<PACKAGE>/`. Le programme peut être construit comme le programme `fli4l`, merci de jeter un œil à la documentation du [paquetage « src »](#) (Page ??).

8.3.14. Les autres fichiers

Tous les fichiers stockés sur le routeur sont dans les répertoires `opt/etc/` et dans `opt/files/`. Voici une description :

- Les scripts dans `opt/etc/boot.d` et `opt/etc/rc.d` sont exécutés pour l'amorçage du système
- Les Scripts dans `opt/etc/rc0.d` sont exécutés lors de l'arrêt du système
- Les Scripts dans `opt/etc/ppp` sont exécutés pour appeler et raccrocher une connexion téléphonique par modem
- Les programmes exécutables et les autres fichiers dans `opt/files/`, sont utilisés en fonction de leurs positions dans le fichier système (par exemple le fichier `opt/files/bin/busybox` est placé tout en haut dans le répertoire `/bin`)

Les scripts dans `opt/etc/boot.d/`, `opt/etc/rc.d/` et `opt/etc/rc0.d/` sont nommés de façon suivante :

```
rc<numéro>.<nom>
```

Le numéro détermine l'ordre d'exécution de l'installation, le nom donne une indication, pour quel programme/paquetage le script est traité.

8.4. Conditions générales de création de script pour fli4l

Nous n'avons *pas* écrit d'introduction générale pour le Scripts-Shell, mais vous pouvez lire cette introduction sur Internet, ici nous traitons que des situations particulières pour `fli4l`. Des informations complémentaires sont disponibles dans les différentes pages du manuel Unix-/Linux-. Les liens suivants peuvent servir de point de départ pour ce sujet :

- Introduction au Scripts-Shell :
 - <http://cip.physik.uni-freiburg.de/main/howtos/sh.php>
- Pages d'aide en ligne :
 - <http://linux.die.net/>
 - <http://heapsort.de/man2web>
 - <http://man.he.net/>
 - http://www.linuxcommand.org/superman_pages.php

8.4.1. Structure

Dans le monde Unix, il est essentiel de démarrer le script avec le nom de l'interpréteur de commande, à la première ligne vous devez indiquer :

```
#!/bin/sh
```


Pour que l'on puisse identifier plus facilement le script, à savoir, à quoi il sert, qui l'a écrit, il faut maintenant que celui-ci soit suivi d'une en-tête à peu près comme ceci :

```
#-----  
# /etc/rc.d/rc500.dummy - start my cool dummy server  
#  
# Creation:      19.07.2001  Toller Hecht <toller-hecht@example.net>  
# Last Update:   11.11.2001  Süße Maus <suesse-maus@example.net>  
#-----
```

Vous pouvez maintenant poursuivre le script ...

8.4.2. Gestion des variables de configuration

Généralités

La composition de la configuration de fli4l est dans le fichier `config/<PAQUETAGE>.txt`. Cette documentation contient les [Variables actives](#) (Page 295) et la création du support de boot avec le fichier `rc.cfg`. Lors du boot du routeur, ce fichier est lu avant tous les scripts-rc (les scripts sont dans `/etc/rc.d/`). Ce script peut accéder avec le `$<nom de variable>` à toutes les variables de configuration du routeur.

Avez-vous besoin des valeurs des variables de configuration, même après le boot ? Vous pouvez à partir du fichier `/etc/rc.cfg`, avec lequel vous avez écrit la configuration pour le support de boot. Par exemple, vous pouvez lire la valeur de la variable `OPT_DNS`, avec un script, vous devez le faire de la manière suivante :

```
eval $(grep "^OPT_DNS=" /etc/rc.cfg)
```

Cela fonctionne également avec plusieurs variables (c'est-à-dire en utilisant une seule fois le programme `grep`) :

```
eval $(grep "^\(HOSTNAME\|DOMAIN_NAME\|OPT_DNS\|DNS_LISTEN_N\)=" /etc/rc.cfg)
```

Stockage persistant des données

Les paquetages ont parfois besoin de stocker des données sur un support persistant, ils pourront survivre au redémarrage du routeur. Il existe une fonction `map2persistent`, elle peut être utilisée depuis un script qui sera enregistré dans `/etc/rc.d/`. Ce script doit contenir la variable avec le chemin et le sous-répertoire. L'idée est que la variable est configurée avec un chemin réel – Alors, ce chemin sera utilisé, car l'utilisateur l'a souhaité ou la variable sera configurée sur « auto » – Alors, un sous-répertoire correspondant sera créé en-dessous du répertoire sur un support persistant selon le deuxième paramètre. La fonction retourne le résultat de la variable, avec le nom qui a été indiqué dans le premier paramètre.

Un exemple permettra de clarifier cela. Soit la variable `VBOX_SPOOLPATH`, qui est paramétrée avec un chemin ou avec la valeur « auto ». Pour l'activation

```
begin_script VBOX "Configuring vbox ..."  
[...]  
map2persistent VBOX_SPOOLPATH /spool  
[...]  
end_script
```

Cela signifie que la variable `VBOX_SPOOLPATH`, ne sera pas modifié (si elle contient un chemin) ou le chemin sera remplacé par `/var/lib/persistent/vbox/spool` (Si elle contient la valeur « auto »). La valeur se réfère ⁶ `/var/lib/persistent` est le répertoire pour enregistrer les données sur un support de stockage non volatile, `<SCRIPT>` représente un minuscule script d'exécution (ce nom est dérivé du premier argument exécute `begin_script` (Page 322)). Si aucun support convenable n'existe (cela peut être possible), le répertoire `/var/lib/persistent` sera enregistré dans le disque RAM.

Il convient de noter, que le chemin utilisé par `map2persistent` n'est *pas* généré automatiquement – Cela doit être fait soi-même (peut-être avec la commande `mkdir -p <chemin>`).

Dans le fichier `/var/run/persistent.conf` vous pouvez vérifier si le support de stockage persistant pour les données est possible. Exemple :

```
. /var/run/persistent.conf
case $SAVETYPE in
persistent)
    echo "Stockage persistant possible!"
    ;;
transient)
    echo "Stockage persistant PAS possible!"
    ;;
esac
```

8.4.3. Recherche d'erreur

Au démarrage d'un script, il est souvent utile d'activer le mode débogage, pour déterminer si « un ver est dans le script » et pour savoir s'il est inséré au début ou à la fin du texte :

```
begin_script <OPT-Name> "start message"
<script code>
end_script
```

En fonctionnement normal, un texte apparaît au démarrage du script et à la fin de se même texte le préfixe « finished » sera spécifié.

Si vous voulez déboguer un script, vous devez faire deux choses :

1. Il faut mettre `DEBUG_STARTUP` (Page 28) sur « yes ».
2. Vous devez activer le débogage de l'OPT choisi. On le fait en général par la variable suivante dans les fichiers de configurations : ⁷

```
<OPT-Name>_DO_DEBUG='yes'
```

Maintenant vous pourrez voir sur la console, la représentation exact de l'exécution du programme.

D'autres variables pour le débogage

DEBUG_ENABLE_CORE Si cette variable est placée sur « yes », elle permet de créer un core-Dumps (ou image mémoire). Si un programme se bloque en raison d'une erreur, un fichier image enregistre l'état actuel du système, il pourra ensuite être utilisée pour une analyser du problème. L'image core-Dumps sera enregistrée dans le dossier `/var/log/dumps`.

6. à l'aide d'un soi-disant « lien » monté

7. parfois, plusieurs scripts de démarrage sont utilisés pour chaque variable de débogage, ces variables ont des noms différents pour le débogage. Voici un rapide coup d'oeil sur ces scripts.

- DEBUG_IP** Si cette variable est placée sur « yes », tous les appels du programme par le protocole ip seront enregistrés.
- DEBUG_IPUP** Si cette variable est placée sur « yes », lors de l'exécution des scripts ip-up/ip-down les instructions exécutées seront stockées dans le système de journalisation.
- LOG_BOOT_SEQ** Si cette variable est placée sur « yes », elle enregistre dans `bootlogd` tous le processus de Boot visible sur la console. Cette variable est placée par défaut sur « yes ».
- DEBUG_KEEP_BOOTLOGD** Normalement `bootlogd` se termine à la fin du processus de boot. Si on active cette variable, cela permet l'enregistrement au-delà de l'arrêt du processus de boot visible sur la console.
- DEBUG_MDEV** Si on active cette variable cela génère le protocole Démons-mdev et produit un fichier sur tous les périphériques dans le dossier `/dev`

8.4.4. Remarques

- Il est *toujours* préférable d'utiliser les accolades « `{...}` » à la place des parenthèses « `(...)` ». Il convient après l'ouverture de l'accolade de placer, un espace ou un saut de ligne avant la prochaine commande et de placer avant la fermeture de l'accolade un point-virgule ou un nouveau saut de ligne. Par exemple :


```
{ echo "cpu"; echo "quit"; } | ...
```

 Équivaut à :


```
{
    echo "cpu"
    echo "quit"
} | ...
```
- Un script peut être arrêté prématurément avec la commande « `exit` ». Mais pour le Script de démarrage (`opt/etc/boot.d/...`, `opt/etc/rc.d/...`), le Script d'arrêt (`opt/etc/rc0.d/...`) et les Scripts `ip-up/ip-down` avec (`opt/etc/ppp/*`) cela est carrément mortelle, il faut dire aussi que ces Scripts ne seront plus exécutés. En cas de doute, ne toucher à rien.
- KISS - Keep it small and simple. Vous voulez utiliser Perl en tant que langage script ? Les possibilités d'écriture pour `fl4l` ne te suffisent pas ? Penser à votre installation ! Votre OPT en a vraiment besoin ? `fl4l` est toujours « uniquement » un routeur, un routeur ne doit pas offrir de services, comme un serveur.
- Le message d'erreur : « `not found` » signifie le plus souvent que le script est encore au format-DOS. Autre problème : si le script n'est pas exécuté. Dans les deux cas, vous devez vérifier le fichier `opt/<PACKAGE>.txt`, pour voir si les options sont correctes (par rapport au « mode », « gid », « uid » et Flags). Si le script est produit seulement au démarrage du routeur, vous devez exécuter la commande « `chmod +x <nom du script>` ».
- Pour les fichiers temporaires, vous devez utiliser le chemin `/tmp`. Mais il est essentiel de veiller à ne pas utiliser trop d'espace, parce que le dossier est dans un Ramdisk-RootFS ! Si vous avez besoin de plus d'espace, il faut créer un Ramdisk et le monter. L'ensemble des détails à ce sujet se trouvent dans le paragraphe "RAM-Disks" de la documentation Dev.
- Afin que les fichiers temporaires obtiennent un nom unique, vous devez ajouter un ID de processus actuelle, dans la variable du Shell le caractère « `$` » sera ajouté au nom du fichier. `/tmp/<OPT-Name>.$$` et le nom du fichier sera correct, mais `/tmp/<OPT-Name>`

est plutôt moins, bien sûr <OPT-Name> ne doit pas laissé comme ceci, mais vous devez le modifier en fonction.

8.5. Utiliser le filtrage de paquets

8.5.1. Ajouter vos propres chaînes et règles

Un ensemble de routines est fourni pour manipuler le filtrage de paquets, elles permettent d'ajouter ou de supprimer des chaînes (en anglais "Chains") et des règles. Une chaîne sert à nommée une liste de règles ordonnées. Il y a déjà un ensemble de chaînes prédéfinies sur le routeur fli4l (PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING). Vous pouvez créer d'autre chaînes pour certaines fonctions et selon vos besoins.

add_chain/add_nat_chain <chain> : ajoute une chaîne de « filtrage » ou de table « nat »

flush_chain/flush_nat_chain <chain> : supprime toutes les règles d'une chaîne de « filtrage » ou de table « nat »

del_chain/del_nat_chain <chain> : supprime une chaîne à partir du « filtrage » ou de la table « nat ». Les chaînes doivent être vides avant qu'elles puissent être supprimées, et ces chaînes ne doivent pas avoir de référence. Une telle référence est par exemple l'action JUMP dont la cible est précisément cette chaîne.

add_rule/ins_rule/del_rule : ajoute une règle à la fin d'une chaîne (add_rule) ou partout dans la chaîne ins_rule) ou supprime une règle d'une chaîne (del_rule). l'exécution sera la suivante :

```
add_rule <table> <chain> <rule> <comment>
ins_rule <table> <chain> <rule> <position> <comment>
del_rule <table> <chain> <rule> <comment>
```

les paramètres ont les significations suivantes :

table La table dans laquelle la chaîne est insérée

chain La chaîne dans laquelle la règle est insérée

rule La règle qui doit être insérée, vous devez utiliser le même format que dans le fichier-config

position La position dans laquelle la règle doit être insérée (seulement avec ins_rule)

comment Le commentaire, il sera affiché avec la règle, il sera vu dans le filtrage de paquets.

8.5.2. Classer les règles dans une infrastructure

fli4l configure les règles du filtrage de paquets avec certaine norme. Si vous souhaitez insérer vos propres règles, vous pouvez ajouter ces règles, après les règles par défaut. Pour cela, vous devez savoir quelle action l'utilisateur a effectué pour rejeter les paquets. Ces informations peut être obtenue pour les chaînes FORWARD et INPUT en utilisant deux fonctions, get_defaults et get_count. Après l'exécution de

```
get_defaults <chain>
```

On obtient les résultats suivants :

drop : cette variable contient la chaîne dans laquelle le paquet sera dévié et rejeté.

reject : cette variable contient la chaîne dans laquelle le paquet sera dévié et refusé.

Après l'exécution de

```
get_count <chain>
```

la variable **res** contient le nombre de règle de la chaîne **<chain>**. Cette position est importante car vous ne pouvez *pas* utiliser simplement **add_rule** pour ajouter une règle à la fin de les chaînes de « filtrage » prédéfini **INPUT**, **FORWARD** et **OUTPUT**. Car ces chaînes sont déjà terminées par des règles par défaut, qui traite tous les paquets restants, en fonction de la disponibilité de la variable **PF_<Chaîne>_POLICY**. Donc si vous insérez cette dernière règle *après* les autres il n'y aura aucun effet. La fonction **get_count** permet maintenant, de détecter l'emplacement directement *avant* la dernière règle et de transmettre cette position à la fonction **ins_rule** avec le paramètre **<position>** ainsi, la règle souhaitée sera ajoutée à la fin de la chaîne appropriée, cependant devant la dernière règle saisie.

Voici un exemple à partir du script `opt/etc/rc.d/rc390.dns_dhcp` du paquetage « `dns_dhcp` » pour une petite explication :

```
case $OPT_DHCPRELAY in
  yes)
    begin_script DHCRELAY "starting dhcprelay ..."

    idx=1
    interfaces=""
    while [ $idx -le $DHCPRELAY_IF_N ]
    do
      eval iface='$DHCPRELAY_IF_'$idx

      get_count INPUT
      ins_rule filter INPUT "prot:udp if:$iface:any 68 67 ACCEPT" \
        $res "dhcprelay access"

      interfaces=$interfaces' -i '$iface
      idx=`expr $idx + 1`
    done
    dhcrelay $interfaces $DHCPRELAY_SERVER

    end_script
  ;;
esac
```

Ici vous pouvez voir dans le milieu de la boucle l'exécution de **get_count** suivie par l'exécution de la fonction **ins_rule**, entre autres la variable **res** est passé comme paramètre **position**.

8.5.3. Extension pour le test des paquets de filtrage

`fi4l` utilise la syntaxe **match:params** dans les règles de filtrage des paquets, cela permet d'avoir des conditions supplémentaires pour les paquets (voir **mac:**, **limit:**, **length:**, **prot:**, ...). Si vous souhaitez ajouter des tests supplémentaires, vous devez faire comme ci-dessous :

1. Créer d'un fichier `opt/etc/rc.d/fwrules-<name>.ext`. Le contenu de ce fichier ressemble à ceci :

```
# extension is available
foo_p=yes

# the actual extension, adding matches to match_opt
do_foo()
{
    param=$1
    get_negation $param
    match_opt="$match_opt -m foo $neg_opt --fooval $param"
}
```

2. Test de l'extension :

```
$ cd opt/etc/rc.d
$ sh test-rules.sh 'foo:bar ACCEPT'
add_rule filter FORWARD 'foo:bar ACCEPT'
iptables -t filter -A FORWARD -m foo --fooval bar -s 0.0.0.0/0 \
-d 0.0.0.0/0 -m comment --comment foo:bar ACCEPT -j ACCEPT
```

3. L'extension est incluse et tous les fichiers restants (les composants `iptables`) sont archivés avec le mécanisme connu.
4. Permet l'extension dans la configuration, il faut ajouter `FW_GENERIC_MATCH` dans le fichier-exp, par exemple :

```
+FW_GENERIC_MATCH(OPT_FOO) = 'foo:bar' : ''
```

8.6. Création d'un CGI pour le paquetage *httpd*

8.6.1. Informations générales sur le serveur Web

Le serveur web, qui est utilisé dans `fli4l` est un `mini_httpd` de ACME Labs. Les fichiers sources peuvent être téléchargés à partir du site http://www.acme.com/software/mini_httpd/. Cependant, quelques modifications ont été apportées pour `fli4l`. Les ajustements sont dans le paquetage `src` et dans le répertoire `src/fbr/buildroot/package/mini_httpd`.

8.6.2. Nom du script

Le nom du script doit être significatif, de sorte qu'il soit plus facile à distinguer par rapport aux autres scripts, il ne doit pas avoir de collision de noms avec les différents OPTs.

Les sauts de ligne dans les scripts exécutés sous DOS, seront convertis en sauts de ligne UNIX, vous devez aussi modifier le fichier `opt/<PACKAGE>.txt`, voir Table 8.2 (Page 293).

8.6.3. Configuration du menu

Pour faire une nouvelle entrée dans le menu, vous devez paramétrer le fichier `/etc/httpd/menu`. Ce mécanisme permet de faire des modifications du menu pendant le fonctionnement de l'OPT. Pour cela on utilise le script `/etc/httpd/menu`, il vérifie le format du fichier pour que celui-ci soit toujours consistant. Pour ajouter un nouvel élément dans le menu, vous devez utiliser la commande de la manière suivante :

```
httpd-menu.sh add [-p <priority>] <link> <name> [section] [realm]
```

Si vous indiquez un nom dans **<name>** il sera inclus dans la section, que vous avez indiqué dans **[section]**. Si vous omettez la section, le nom sera par défaut inclus dans la section du "paquetage-OPT". Vous indiquez dans **<link>** la cible du nouveau lien. Dans **<priority>** vous spécifiez la priorité de l'élément du menu dans sa section. S'il n'est pas spécifié, 500 sera utilisé pour la priorité par défaut. La priorité doit être un nombre à trois chiffres. Le lien qui se situe le plus haut dans la section a une priorité la plus faible, si vous voulez un lien le plus bas, vous devez par exemple choisir une priorité de 900. De même, la priorité des données sont triées avec la cible du lien. Dans **[realm]** vous indiquez le domaine pour que l'utilisateur connecté puisse avoir les autorisations nécessaires pour voir l'élément du menu qui sera affiché. S'il **[realm]** n'est pas spécifié, l'élément du menu sera toujours affiché, voir aussi la section « [Droits des utilisateurs](#) » (Page 331).

Exemple :

```
httpd-menu.sh add "Nouveau-fichier.cgi" "Cliquez ici" "Outils" "outils"
```

Cet exemple produit dans la section "Outils" un lien avec le titre "Cliquez ici", la destination du lien sera "Nouveau-fichier.cgi", la section sera créée si elle n'est pas définie.

Vous pouvez également supprimer dans le script un enregistrement de lien du menu :

```
httpd-menu.sh rem <link>
```

Avec cette commande le lien **<link>** qui a été enregistré sera supprimé.

Important: *Si plusieurs enregistrements se réfèrent au même fichier dans le menu, toutes ces enregistrements seront supprimés dans le menu.*

Puisque des sections peuvent aussi avoir des priorités, celles-ci peuvent être créées manuellement. Si une section est créée automatiquement lorsque vous ajoutez une entrée, il recevra automatiquement la priorité 500. Voici la syntaxe pour la création d'une section :

```
httpd-menu.sh addsec <priority> <name>
```

Ici aussi **<priority>** doit contenir un nombre à trois chiffres.

Pour apprendre à configurer judicieusement les priorités, il est intéressant de regarder le fichier `/etc/httpd/menu` pendant l'exécution de `fi4l` les priorités sont dans la deuxième colonne.

Pour être complet, voici un bref commentaire sur le format du fichier menu. Si la commande `httpd-menu.sh` ne suffit pas, vous pouvez sauter ce paragraphe. Le fichier `/etc/httpd/menu` a la structure suivante : il est divisé en quatre colonnes. La première colonne il y a une lettre de code, pour différencier les intitulés et les enregistrements. Dans la deuxième colonne il y a les priorités triées. La troisième colonne contient en intitulé un trait d'union, cela veut dire qu'il n'y a aucune signification pour le titre et de la configuration de la cible du lien. Dans le reste de la ligne se trouve le texte qui apparaîtra plus tard dans le menu.

Avec la lettre d'identification « t », une nouvelle section dans le menu sera installée. Pour configurer une entrée normale dans le menu vous utilisez la lettre d'identification « e ». Un exemple :

```
t 300 - Mon grand OPT
e 200 monopt1.cgi lien pour grand
e 500 monopt1.cgi?plus=oui lien plus pour grand
```

Lors de l'édition de ce fichier vous devez vous assurer que le script `httpd-menu.sh` soit toujours trié avant de sauvegarder le fichier. Que les différentes sections soit triées et que les sections enregistrées sont triées par section l'algorithme de tri exacte peut être repris par `httpd-menu.sh` – Cependant, il est possible d'étendre ce script avec de nouvelles fonctionnalités, pour que tous les modifications du menu se produisent à un emplacement central.

8.6.4. Construction d'un script CGI

L'en-tête

Tous les scripts du serveur Web sont de simples scripts shell (les interpréteurs tels que Perl, PHP, etc, sont beaucoup trop volumineux pour `fl4l`). Le script doit obligatoirement commencer par l'en-tête avec (la référence de l'interpréteur, le nom, une indication sur le scénario, l'auteur, la licence).

Script auxiliaire `cgi-helper`

Tout de suite après l'en-tête vous devez placer le script auxiliaire `cgi-helper` avec la commande suivante :

```
. /srv/www/include/cgi-helper
```

L'espace entre le point et le slash (ou la barre oblique) est important !

Ce script auxiliaire permet diverses fonctionnalités pour simplifier la création des CGIs, il est essentiel à `fl4l`. De plus, avec l'intégration de `cgi-helper` il effectue également des tâches standard, telles que l'analyse syntaxique des variables, qui ont été associées à des formulaires ou des transitions URL ou de l'affichage de la langue pour le texte ou des fichiers CSS.

Le tableau 8.4 donne un aperçu des fonctionnalités des scripts `cgi-helper`.

TABLE 8.4.: Les fonctions pour le script `cgi-helper`

Nom	Fonction
<code>check_rights</code>	Vérification des droits des utilisateurs
<code>http_header</code>	Édition d'une en-tête HTTP standard ou d'une en-tête spécifique, par ex., la manière de télécharger des fichiers
<code>show_html_header</code>	Édition complètes d'une en-tête de page (y compris les en-têtes HTTP et les en-têtes du Menu)
<code>show_html_footer</code>	Édition de la fin d'une page HTML
<code>show_tab_header</code>	Édition d'une fenêtre de contenu pour un tableau
<code>show_tab_footer</code>	Édition de la fin de contenu pour un tableau
<code>show_error</code>	Édition d'une fenêtre pour les messages d'erreur (couleur : rouge)
<code>show_warn</code>	Édition d'une fenêtre pour les messages d'alerte (couleur : jaune)
<code>show_info</code>	Édition d'une fenêtre pour les messages d'informations/réussites (couleur : vert)

Contenu d'un script CGI

Afin d'assurer un aspect homogène et surtout pour la compatibilité avec les futures versions-`fl4l`, il est fortement recommandé d'utiliser les fonctions du script auxiliaire `cgi-helper`, même si nous pouvons théoriquement générer toutes les sorties dans un même CGI.

Un simple script-CGI peut ressembler à ceci :


```
#!/bin/sh
# -----
# Header (c) Autor Datum
# -----
# get main helper functions
. /srv/www/include/cgi-helper

show_html_header "Mon premier CGI"
echo '    <h2>Bienvenue</h2>'
echo '    <h3>Ceci est un exemple de script-CGI</h3>'
show_html_footer
```

Fonction `show_html_header`

La valeur qui est indiqué dans la fonction `show_html_header`, est utilisé pour le titre. Il sera généré automatiquement dans le menu, il inclue aussi automatiquement le script CSS et le fichier langue. Il faut pour cela que les fichiers scripts ont le même nom et qu'ils soient dans les répertoires `/srv/www/css` et `/srv/www/lang` (bien sûr, avec une extension différente). Exemple :

```
/srv/www/admin/OpenVPN.cgi
/srv/www/css/OpenVPN.css
/srv/www/lang/OpenVPN.de
```

L'utilisation du fichier-langue ainsi que le fichier-CSS sont facultatif. Intégré toujours `css/nom.css` et `lang/nom.<lang>`, `<lang>` correspond à la langue choisie.

Vous pouvez placer à côté du titre de la fonction `show_html_header` d'autres paramètres. Une exécution de la fonction avec tous les paramètres pourrait ressembler à ceci :

```
show_html_header "Titre" "refresh=$time;url=$url;cssfile=$cssfile;showmenu=no"
```

Tous les paramètres supplémentaires, comme le montre l'exemple ci-dessus, doivent avoir des guillemets et être séparés par un point-virgule. Autrement le contrôle de syntaxe ne sera *pas* réussi ! Il est nécessaire de respecter la syntaxe des paramètres.

Voici un bref aperçu des fonctions de ces paramètres :

- `refresh=time` : Temps en secondes dans lequel la page sera rechargée par le navigateur.
- `url=url` : L'URL est rechargé après un rafraîchissement.
- `cssfile=cssfile` : Nom du fichier CSS, si celui-ci est différent du CGI.
- `showmenu=no` : L'affichage du menu et l'en-tête peut être supprimée.

D'autres conseils pour le contenu du CGI :

- Prenez votre temps :-)
- Écrivez proprement le HTML (SelfHTML ⁸ est un bon point de départ)
- Renoncer au bric-à-brac très moderne, (le JavaScript est OK, si cela ne crée pas de problème avec les utilisateurs, le tout fonctionnent également sans JavaScript)

Fonction `show_html_footer`

La fonction `show_html_footer` ferme le bloc dans le script CGI, qui a été lancé par la fonction `show_html_header`.

8. voir <http://de.selfhtml.org/>

Fonction show_tab_header

Dans le CGI, pour que le contenu de vos résultats soit bien rangé et affiché sur la page Web, avec l'aide du `cgi-helper` vous pouvez utiliser la fonction `show_tab_header`. Il peut générer des titres dans des onglets cliquables dans un 'Tableau', ainsi votre page peut être divisé en plusieurs zones logiquement séparées.

Les paramètres dans la fonction `show_tab_header` sont toujours écrits deux par deux. Le premier paramètre correspond au titre de l'onglet et le deuxième correspond au lien. Si vous indiquez 'no' comme deuxième paramètre, le titre sera seulement placé dans l'onglet (en couleur) et le lien ne sera pas cliquable.

Dans l'exemple suivant, nous allons créer une 'fenêtre' dans laquelle nous allons mettre le titre 'Une grande fenêtre'. Dans la fenêtre sera écrit 'foo bar' :

```
show_tab_header "Une grande fenêtre" "no"
echo "foo"
echo "bar"
show_tab_footer
```

Dans l'exemple suivant, deux onglets cliquables sont générés, la variable `action` dans le script va transmettre des valeurs différentes.

```
show_tab_header "1. onglet" "$myname?action=machdies" \
                "2. onglet" "$myname?action=machjenes"
echo "foo"
echo "bar"
show_tab_footer
```

Maintenant le contenu du script peut exploiter la variable `FORM_action` (voir ci-dessous pour l'exploitation de la variable) en fonction des différents paramètres. l'onglet apparaîtra et l'on pourra le sélectionner en cliquant dessus, si vous ne voulez pas que cet onglet soit sélectionnable, vous devez placer dans la fonction 'no', pour ce passé du lien comme indiqué plus haut. Cela sera plus facile, si vous utilisez l'exemple suivant :

```
_opt_machdies="1. onglet"
_opt_machjenes="2. onglet"
show_tab_header "$_opt_machdies" "$myname?action=opt_machdies" \
                "$_opt_machjenes" "$myname?action=opt_machjenes"
case $FORM_action in
    opt_machdies) echo "foo" ;;
    opt_machjenes) echo "bar" ;;
esac
show_tab_footer
```

Si vous utilisez une variable pour sélectionner un titre, ce nom correspond à l'action de la variable et commencera par un tiret (`_`), ainsi l'onglet correspond à ce nom pourra être sélectionné.

Fonction show_tab_footer

La fonction `show_tab_footer` ferme le bloc dans le script CGI, qui a été lancé par la fonction `show_tab_header`.

Supporte le multi-language

Le script `cgi-helper` contient également une fonction pour faire des scripts CGI en utilisant d'autres langues. Pour ce faire, vous devez 'juste' utiliser les variables commençant par un tiret bas (`_`) pour traduire le texte et de définir ces variables dans la langue de votre choix.

Exemple :

`lang/opt.de` contient :

```
_opt_machdies="Eine Ausgabe"
```

`lang/opt.en` contient :

```
_opt_machdies="An Output"
```

`admin/opt.cgi` contient :

```
...  
echo $_opt_machdies  
...
```

Utilisation de formulaires

Afin de traiter des formulaires, il faut savoir certaines choses. L'utilisateur doit choisir la méthode de formulaire `GET` ou `POST`, les paramètres peuvent être trouvées après le montage du script `cgi-helper` (qui à son tour appelle le de programme auxiliaire `proccgi`) avec la nouvelle variable `FORM_<Paramètre>`. Si vous avez entré un nom, une "adresse" dans le champ de formulaire et si vous indiquez dans le script CGI `$FORM_adresse`, cette adresse sera accessible.

Pour plus d'informations sur le programme `proccgi` vous pouvez les trouver ici <http://www.fpx.de/fp/Software/ProcCGI.html>.

Droits des utilisateurs : Fonction `check_rights`

Afin de vérifier si l'utilisateur a les droits suffisants pour utiliser un script CGI, la fonction `check_rights` est appelée au début du script CGI comme ceci :

```
check_rights <Section> <Action>
```

Le script CGI ne sera exécuté que si l'utilisateur est enregistré.

- A tous les droits (`HTTPD_RIGHTS_x='all'`), ou
- tous les droits dans un domaine spécifique (`HTTPD_RIGHTS_x='<Domaine>:all'`), ou
- a le droit d'effectuer une action spécifique dans un domaine spécifique (`HTTPD_RIGHTS_x='<Domaine>:<Action>'`).

Fonction `show_error`

Cette fonction renvoie un message d'erreur dans une fenêtre rouge. Il a besoin de deux paramètres : un titre et un message. Exemple :

```
show_error "Error: No key" "No key was specified!"
```

Fonction `show_warn`

Cette fonction renvoie un message d'avertissement dans une fenêtre jaune. Il a besoin de deux paramètres : un titre et un message. Exemple :

```
show_info "Warnung" "Actuellement, il n'y pas de connexion!"
```

Fonction `show_info`

Cette fonction renvoie un message de réussite ou d'information dans une fenêtre verte. Il a besoin de deux paramètres : un titre et un message. Exemple :

```
show_info "Info" "Action a été exécutée avec succès!"
```

Script auxiliaire `cgi-helper-ip4`

Immédiatement après le script `cgi-helper`, vous devez ensuite intégrer le script auxiliaire `cgi-helper-ip4` avec la commande suivante :

```
. /srv/www/include/cgi-helper-ip4
```

L'espace entre le point et le slash (ou la barre oblique) est important !

Ce script fournit des fonctions pour vous aidez à effectuer des tests d'adresse IPv4.

Fonction `ip4_isvalidaddr`

Cette fonction vérifie si une adresse IPv4 qui a été enregistrée est valide. Exemple :

```
if ip4_isvalidaddr ${FORM_inputip}
then
    ...
fi
```

Fonction `ip4_normalize`

Cette fonction supprime les zéros inutile dans l'adresse IPv4 enregistrée. Exemple :

```
ip4_normalize ${FORM_inputip}
IP=$res
if [ -n "$IP" ]
then
    ...
fi
```

Fonction `ip4_isindhcprange`

Cette fonction vérifie si l'adresse IPv4 enregistrée est dans la plage d'adresse de début et de fin. Exemple :

```
if ip4_isindhcprange $FORM_inputip $ip_start $ip_end
then
    ...
fi
```

8.6.5. Divers

Des choses et d'autres (et oui, c'est aussi important!) :

- Le `mini_httpd` ne protège pas les sous-répertoires par mot de passe. Vous devez avoir un répertoire `.htaccess` pour chaque utilisateur ou créer un lien vers un autre fichier `.htaccess`.
- KISS - Keep it simple, stupid!
- Ces informations peuvent être changées à tout moment et sans préavis!

8.6.6. Dépannage

Pour faciliter le débogage d'un script CGI, vous pouvez activer le mode débogage avant l'intégration du script `cgi-helper`. Pour cela, la variable `set_debug` doit être paramétrée sur "yes". Cela conduit à la création du fichier `debug.log`, que vous pourrez télécharger sur l'URL `http://<fli4l-Host>/admin/debug.log`. Cela inclut tous les appels vers ce script CGI. La variable `set_debug` n'est pas globale, et doit être renouvelée dans tous les scripts CGI en question. Exemple :

```
set_debug="yes"
. /srv/www/include/cgi-helper
```

En outre, `cURL`⁹ est idéal pour le dépannage, en particulier lorsque les en-têtes HTTP ne sont pas assemblés correctement ou que le navigateur affiche une page blanche. Et aussi, le comportement de mise cache des navigateurs modernes est un véritable problème.

Exemple : Avec l'exécution suivant, en-tête HTTP ("`dump`", `-D`) l'affichage normale du CGI `admin/mon.cgi` sera publié. Vous pouvez utiliser le nom d'utilisateur ("`user`", `-u`) "`admin`".

```
curl -D - http://fli4l/admin/mon.cgi -u admin
```

8.7. Démarrer, arrêter, se connecter et se déconnecter avec fli4l

8.7.1. Concept de Boot

`fli4l 2.0` peut être installé sur différent média, disque-dur ou carte-Compact-Flash(TM), il est aussi possible de l'installer sur un support Zip ou sur un CD-ROM amovible. En outre, l'installation d'une version sur un disque-dur n'est pas fondamentalement différente que sur une disquette.¹⁰

Ces exigences ont été réalisées grâce à l'archive `opt.img`, jusqu'ici il était installé sur un disque-RAM maintenant il peut être placé sur d'autre média. Il peut s'agir d'une partition sur un disque dur ou une carte-CF. Pour ce second volume, le répertoire `/opt` sera monté et les programmes auront des liens symboliques et seront intégrés dans le rootfs. La structure apparaissant dans le système de fichier RootFS correspond au répertoire `opt` de la distribution `fli4l` à une exception prête – le préfixe des fichiers seront omis. Le fichier `opt/etc/rc` se trouve directement dans `/etc/rc` et pour le fichier `opt/files/bin/busybox` il est dans `/bin/busybox`. Ces fichiers sont seulement des liens dans le média monté en lecture seule, on peut les ignorer, tant que les fichiers ne sont pas modifiés. Si vous voulez les modifier, il faut d'abord rendre ces fichiers accessibles en écriture avec la commande `mk_writable` (voir ci-dessous).

9. voir <http://de.wikipedia.org/wiki/CURL>

10. À l'origine `fli4l` pouvait s'exécuter à partir d'une disquette. Depuis `fli4l` est devenue trop volumineux, la disquette ne peut plus être utilisée.

8.7.2. Scripts de démarrage et d'arrêt

Les Scripts qui sont exécutés lors du boot système, sont dans les répertoires `opt/etc/boot.d` et `opt/etc/rc.d` ils sont également exécutés dans l'ordre.

Important: *Quand ces scripts sont exécutés aucun processus particulier n'est produit, ils ne peuvent pas être arrêtés avec la commande « exit ». Cette commande conduirait à une rupture du processus de Boot !*

Scripts de démarrage dans `opt/etc/boot.d/`

Les scripts de ce répertoire sont exécutés les premiers. Ils ont pour mission, de monter le périphérique de boot, le fichier de configuration `rc.cfg` se trouve sur le support de boot et décompresse l'archive `opt.img`. Selon le [type de Boot](#) (Page 23) il est plus ou moins complexe et font les choses suivantes :

- Charge les pilotes du matériels (optionnel)
- Monte le volume de boot (optionnel)
- Le fichier de configuration `rc.cfg` est lu depuis le volume de boot et sera écrit dans `/etc/rc.cfg`
- Monte le volume Opt (optionnel)
- Extrait les archives Opt (optionnel)

Lors de la construction des scripts, vous avez la chance d'en apprendre davantage sur la configuration de fli4l, le fichier de configuration est également intégré dans l'archive rootfs, dans ce fichier `/etc/rc.cfg` vous trouverez les variables de configuration qui seront analysées puis les scripts de démarrage seront exécutés depuis le répertoire `opt/etc/boot.d/`. Après le montage du volume de boot, le fichier `/etc/rc.cfg` est remplacé par le fichier de configuration dans le volume de boot, de sorte que les scripts de démarrage dans `opt/etc/rc.d/` soit disponible pour l'actuel configuration du volume de boot (voir ci-dessous).¹¹

Scripts de démarrage dans `opt/etc/rc.d/`

C'est les commandes qui sont exécutées à chaque démarrage du routeur, elles peuvent être stockés dans le répertoire `opt/etc/rc.d/`. Les conventions suivantes s'appliquent :

1. Il faut classer les noms de script comme ceci :

```
rc<nombre à trois chiffres>.<nom de l'OPT>
```

Les scripts sont démarrés dans l'ordre croissant des numéros. Si plusieurs scripts ont le même numéro attribué, c'est le caractère alphabétique après le point qui détermine l'ordre. L'installation des paquets s'effectue les uns après les autres, ils sont définis par un numéro.

Voici une estimation approximative, des numéros pouvant être utilisé pour une installation :

2. Vous devez *placer* dans ces scripts, toutes les fonctions nécessaires pour changer le RootFS. (par ex. pour la création d'un répertoire `/var/log/lpd`).

11. Normalement, ces deux fichiers sont identiques. Un changement n'est possible que si le fichier de configuration sur le volume de démarrage a été modifié manuellement, par exemple pour modifier une configuration qui sera utilisé plus tard, sans avoir à reconstruire l'archive fli4l.

Numéro	Fonction
000-099	Système de base (hardware, fuseau horaire, système de fichiers)
100-199	Module Kernel (drivers)
200-299	Connexions externe (PPPoE, ISDN4Linux, PPtP)
300-399	Réseau (routage, interface, filtrage de paquet)
400-499	Serveur (DHCP, HTTPD, Proxy, etc.)
500-900	Tout le reste
900-997	Tout ce qui peut générer un dial-up
998-999	Réservé (ne pas utiliser !)

- Vous ne *devez* pas effectuer d'écriture dans les fichiers scripts qui font partie de l'archive-opt, car ces fichiers sont en lecture seule sur le média. Pour modifier de tel fichier, il faut, au préalable rendre accessible ce fichier en écriture via `mk_writable` (voir ci-dessous). En exécutant cette fonction, le fichier si nécessaire, sera copié et sera accessible en écriture sur le RootFS. Si le fichier est déjà en écriture, rien ne se passera lors de l'exécution de la fonction `mk_writable`.

Important: *mk_writable doit être utilisé sur les fichiers qui sont dans le rootfs et non pas par le biais du dossier /opt. Si vous voulez modifier /usr/local/bin/foo vous exécutez mk_writable /usr/local/bin/foo.*

- Ces scripts ont besoin de vérifiés, avant d'exécuter les commandes réelles, si l'OPT correspondant est actif. Cela se fait habituellement par une simple distinction de cas :

```
if [ "$OPT_<OPT-Name>" = "yes" ]
then
    ...
    # ici OPT start!
    ...
fi
```

- Pour pouvoir déboguer plus facile, vous devez insérer dans le script les fonctions `begin_script` et `end_script` :

```
if [ "$OPT_<OPT-Name>" = "yes" ]
then
    begin_script F00 "configuring foo ..."
    ...
    end_script
fi
```

Pour juste déboguer de script au démarrage, vous devez simplement activer la variable `F00_D0_DEBUG='yes'`.

- Toutes les variables de configuration sont directement disponible par les scripts. Des explications pour accéder à d'autres scripts à partir des variables de configuration peuvent être trouvés dans la section « [Gestion des variables de configuration](#) » (Page 321)
- Le dossier /opt ne doit pas être utilisé comme stocker des données OPTs. Si de l'espace supplémentaire est nécessaire, l'utilisateur doit de définir un chemin approprié en utilisant la variable de configuration. Selon le type de données à stocker (données persistante ou transitoire) vous devez utiliser différentes affectations par défaut. Le chemin /var/run/

est logique pour les données transitoires, tandis que pour les données persistant, il est conseillé d'utiliser la fonction `map2persistent` (Page 321) combiné avec une variable de configuration appropriée.

Scripts d'arrêt dans `opt/etc/rc0.d/`

Chaque ordinateur a besoin d'un temps pour s'arrêter ou redémarrer. Il se pourrait bien que vous pouvez avoir à effectuer des opérations avant que l'ordinateur s'éteigne ou redémarre. Les commandes officielles sont « halt » et « reboot ». Ces commandes sont également placées dans IMONC ou dans le Web-GUI si vous l'avez installé, un clique sur le bouton suffira pour arrêter ou redémarrer le routeur.

Tous les scripts d'arrêt se trouvent dans le répertoire `opt/etc/rc0.d/`. Le nom du fichier est analogue à celui du script de démarrage. Ils sont également exécutés dans ordre *croissant* des numéros.

8.7.3. Fonctions auxiliaires

Dans `/etc/boot.d/base-helper` différentes fonctions sont mises à disposition, elles peuvent être utilisées pour les scripts de démarrage. Cela se applique pour certaines choses comme pour un support de débogage, pour le chargement des modules du Kernel ou pour la sortie des messages.

Les différentes fonctions sont les suivantes et brièvement décrites.

Contrôle des scripts

begin_script <Symbol> <Message> : envoie un message et active le débogueur de script en utilisant `set -x`, si <Symbol>_DO_DEBUG est sur « yes ».

end_script : envoie un message final et fournit l'état de débogage si vous avez activé le `begin_script`. Pour chaque activation du `begin_script` un `end_script` sera associé et activé (et vice versa).

Chargement des modules du Kernel

do_modprobe [-q] <Module> <Paramètre>*: Charge le module du Kernel, y compris ses paramètres, en résolvant en même temp les dépendances du module. Le paramètre « -q » empêche qu'un message erreur soit mis. La fonction retourne en cas de succès une valeur nulle, dans le cas d'une erreur une valeur non nulle. Cela permet écrire un code pour gérer les échecs lors du chargement les modules du Kernel :

```
if do_modprobe -q acpi-cpufreq
then
    # pas de contrôle de fréquence du CPU via ACPI
    log_error "le contrôle de fréquence du CPU via ACPI n'est pas disponible."
    # [...]
else
    log_info "le contrôle de fréquence du CPU via ACPI est activé."
    # [...]
fi
```


do_modrobe_if_exists [-q] <Chemin> <Module> <Paramètre>* :

vérifie d'abord si le module /lib/modules/<version du kernel>/<Chemin>/<Module> existe et ensuite exécute do_modprobe.

Important: *Le module doit exister précisément par ce nom, aucun alias ne peuvent être utilisé. Lorsque vous utilisez un alias **do_modprobe** sera exécuté immédiatement.*

Messages et gestion des erreurs

log_info <Message> : envoie un message sur la console et dans /bootmsg.txt. Si aucun message n'est enregistré dans ce paramètre le fichier log_info sera lu depuis l'entrée par défaut. La fonction renvoie toujours en retour une valeur nulle.

log_warn <Message> : envoie un message d'avertissement sur la console et dans /bootmsg.txt, la chaîne WARN: sera utilisée comme préfixe. Si aucun message n'est enregistré dans ce paramètre le fichier log_warn sera lu depuis l'entrée par défaut. La fonction renvoie toujours en retour une valeur null.

log_error <Message> : envoie un message d'erreur sur la console et dans /bootmsg.txt, la chaîne ERR: sera utilisée comme préfixe. Si aucun message n'est enregistré dans ce paramètre le fichier log_error sera lu depuis l'entrée par défaut. La fonction renvoie toujours en retour une valeur null.

set_error <Message> : envoie un message d'erreur qui sera définit dans une variable d'erreur interne, ensuite elle pourra être examiné via is_error.

is_error : réinitialise la variable d'erreur interne et renvoie true, si elle a été précédemment défini par set_error.

Fonctions réseau

translate_ip_net <Valeur> <Nom de Variable> [<Variable de Résultat>] :

remplace les références symboliques par des paramètres. Actuellement les traductions suivantes ont lieu :

.*.*., **none**, **default**, **pppoe** ne sont pas remplacé

any sera remplacé par 0.0.0.0/0

dynamic sera remplacée par une adresse IP du routeur, à travers lequel il existe une connexion Internet.

IP_NET_x sera remplacé par le réseau se trouvant dans la configuration.

IP_NET_x_IPADDR sera remplacé par l'adresse IP se trouvant dans la configuration.

IP_ROUTE_x sera remplacé par le réseau routé se trouvant dans la configuration

@<Hostname> sera remplacé par l'hôte ou par l'adresse IP se trouvant dans la configuration.

Le résultat de la traduction est mémorisée dans la variable dont le nom est transmis dans le troisième paramètre, si ce paramètre est manquant, le résultat sera stocké dans la variable **res**. Le nom de la variable qui est transmis dans le second paramètre est utilisé uniquement pour les messages d'erreur si la traduction échoue, cela permet d'appeler la source de la valeur à traduire. Si une erreur se produit, un message est alors exécuté.

Unable to translate value '<Valeur>' contained in <Nom de Variable>.

La valeur nulle est retournée si la traduction a réussie et une valeur non nulle sera retournée si une erreur s'est produite.

Divers

mk_writable <Fichier> : pour vous assurer que le fichier transmis sera accessible en écriture.

Seulement si le fichier est en lecture seule dans le fichier-système et juste monté par un lien symbolique, une copie locale sera alors créée pour avoir le fichier en écrit.

unique <Liste> : supprime les doublons dans la liste transmise. La liste est retournée avec la variable `list`.

8.7.4. Périphériques ttyI

Au sujet les périphériques `ttyI`, vous pouvez utiliser (`/dev/ttyI0 .../dev/ttyI15`), pour la création d'un « émulateur de modem » avec plusieurs cartes ISDN (ou RNIS), il existe un compteur pour les conflits entre les différents OPTs et les utilisations de ces périphériques, c'est un dispositif à éviter. Ces émulateur seront créés lors du démarrage du routeur, dans le fichier `/var/run/next_ttyI`, avec l'utilisation un compteur. Dans l'exemple de script suivant, la valeur est interrogé et peut être augmentée de un, il sera exporté de nouveau dans le prochain OPT.

```

ttydev_error=
ttydev=$(cat /var/run/next_ttyI)
if [ $ttydev -le 16 ]
then
    ttydev=$((ttydev + 1))          # ttyI device available? yes
    echo $ttydev >/var/run/next_ttyI # ttyI device + 1
    # save it
else
    # ttyI device available? no
    log_error "No ttyI device for <Nom de votre OPT> available!"
    ttydev_error=true              # set error for later use
fi

if [ -z "$ttydev_error" ]
then
    # start OPT only if next tty device
    # was available to minimize error
    ...
    # messages and minimize the
    # risk of uncomplete boot
fi

```

8.7.5. Scripts de connexion et de déconnexion par modem

Généralités

Après avoir établi ou coupé une connexion par modem, les scripts `/etc/ppp/` sont traitées. Voici les actions qui sont nécessaires pour activer ou désactiver une connexion, qui seront stocker dans l'OPT. Le schéma des noms de fichier est le suivant :

```

ip-up<numéro à trois chiffres>.<nom d'OPT>
ip-down<numéro à trois chiffres>.<nom d'OPT>

```

Le script `ip-up` est exécuté après la *connexion* et le script `ip-down` est exécuté après la *déconnexion*

Important: Dans le script `ip-down` aucune intervention ne doit être réalisé, autrement cela conduirait à une nouvelle connexion Internet, avec uniquement un accès à l'état online permanent, pour les utilisateurs qui non pas de forfait illimité, cela peut couter très cher.

Important: Car pour ce script, il n'y a aucun processus qui est généré, en plus, ce script ne peut pas être arrêté avec la commande « `exit` » !

Remarque : le scripts `ip-up` peut être examiné lors qu'il est exécuté, pour cela le fichier `rc400` sera vérifier avec la variable `ip_up_events`. Si c'est réglé sur "yes", une connexion par modem existe et le script `ip-up` sera exécuté. Si c'est réglé sur "no", une connexion par modem n'existe pas et le script `ip-up` ne sera pas exécuté. Il y a une exception pour cette règle : Si un routeur Ethernet pur n'est pas configuré pour des connexions commutées, mais configuré pour une route par défaut (0.0.0.0/0), le script `ip-up` sera exécuté qu'une fois, exactement à la fin du processus de boot. (De même le script `ip-down` sera exécuté qu'une fois avant l'arrêt du routeur).

Les variables

Grâce au concept d'appel spécial les scripts `ip-up` et `ip-down` sont exécutés et les variables suivantes sont utilisées :

<code>real_interface</code>	L'interface actuelle, par ex. <code>ppp0</code> , <code>ippp0</code> , ...
<code>interface</code>	Interface-IMOND, avec <code>pppoe</code> , <code>ippp0</code> , ...
<code>tty</code>	Terminal de connexion, peut-être vide !
<code>speed</code>	Vitesse de connexion, par ex. avec ISDN 64000
<code>local</code>	Adresse-IP spécifique
<code>remote</code>	Adresse-IP d'ordinateur auquel vous êtes connecté
<code>is_default_route</code>	Indique si l'actuel <code>ip-up/ip-down</code> est utilisé pour l'interface de la route par défaut peut-être « yes » ou « no »)

Route par défaut

Depuis la version 2.1.0, les scripts `ip-up` et `ip-down` sont exécutés non seulement pour l'interface sur laquelle la route par défaut est configurée, mais aussi pour toutes les connexions qui ont besoin les scripts `ip-up` et `ip-down`. Pour émuler des comportements anciens, vous devez inclure les éléments à déclencher dans les scripts `ip-up` et `ip-down` la requête suivante doit être insérée :

```
# is a default-route-interface going up?
if [ "$is_default_route" = "yes" ]
then
    # Les actions à déclencher
fi
```

Naturellement, les nouveaux comportements doivent être utilisés, pour des actions spécifiques.

8.8. Paquetage Template

Pour illustrer quelques-uns des objectifs décrits ci-dessus, vous avez un paquetage avec des modèles pour la distribution fli4l. Dans ce paquetage vous avez une série de petits exemples, tels que :

- Voir un fichier config dans (`config/template.txt`)
- Un fichier de contrôle qui est écrit dans (`check/template.txt`)
- L’extension des fonctions de contrôle dans (`check/template.ext`)
- Variables de configuration peuvent être stockés pour une utilisation ultérieure dans (`opt/etc/rc.d/rc999.template`)
- stockage des variables de configuration pour être lu à nouveau dans (`opt/files/usr/bin/template_show_config`)

8.9. Construction du Boot sur un support de données

Depuis de la version 1.5 fli4l utilise le programme `syslinux` pour booter. Il a l’avantage d’avoir un système de fichier DOS compatible sur le support de données.

Le support de données pour le boot contient les fichiers suivants :

<code>ldlinux.sys</code>	Le chargeur (« Boot loader ») <code>syslinux</code>
<code>syslinux.cfg</code>	Fichier de configuration pour <code>syslinux</code>
<code>kernel</code>	Linux-Kernel
<code>rootfs.img</code>	RootFS : contient les programmes nécessaires pour le Boot
<code>opt.img</code>	Fichier Optionnel : drivers et Opt-Paquetage
<code>rc.cfg</code>	Fichier de configuration des variables utilisées depuis le répertoire config
<code>boot.msg</code>	Texte pour le menu de démarrage <code>syslinux</code>
<code>boot_s.msg</code>	Texte pour le menu de démarrage <code>syslinux</code>
<code>boot_z.msg</code>	Texte pour le menu de démarrage <code>syslinux</code>
<code>hd.cfg</code>	Fichier de configuration pour l’attribution des partitions

Au démarrage du script `mkfli4l.sh` (ou `mkfli4l.bat` pour DOS) les fichiers `opt.img`, `syslinux.cfg`, `rc.cfg`, ainsi que `rootfs.img` sont d’abord exécutés. Le programme `mkfli4l` exécute les fichiers nécessaires des répertoires `unix` ou `windows` pour l’installation. Dans les deux archives, le Kernel et les autres paquetages à installer sont inclus. Le fichier `rc.cfg` se trouve à la fois dans l’archive-opt et sur le boot disque (ou disque de démarrage).¹²

Ensuite, l’ensemble des fichiers du Kernel, `rootfs.img`, `opt.img` et `rc.cfg`, sont copiés avec les fichiers `syslinux` sur le support de données.

Lors du boot de fli4l le script `/etc/rc` du `rc.cfg` est analysé et l’archive `opt.img` compressée est intégrée à la racine du système de fichiers du disque virtuel (selon le type d’installation, les fichiers seront décompressés directement à la racine du système de fichiers du disque virtuel ou vers le lien symbolique inclus). Pour terminer, les scripts dans le répertoire `/etc/rc.d` sont exécutés dans l’ordre alphanumérique, ensuite les pilotes sont chargés et les services démarrent.

12. Le contenu de ce fichier dans l’archive-opt est nécessaire au début de la phase de boot, car à ce moment le volume de boot n’est pas monté.

8.10. Fichiers de configurations

Voici les dossiers présélectionnés par le routeur-fli4l on-the-fly, qui sont générés au moment du boot.

1. Configuration du fournisseur
 - `etc/ppp/pap-secrets`
 - `etc/ppp/chap-secrets`
2. Configuration du DNS
 - `etc/resolv.conf`
 - `etc/dnsmasq.conf`
 - `etc/dnsmasq_dhcp.conf`
 - `tc/resolv.dnsmasq`
3. Fichier hôte
 - `etc/hosts`
4. Configuration-imond
 - `etc/imond.conf`

8.10.1. Configuration du fournisseur

Pour sélectionner le fournisseur d'accès le réglage du User-ID et du Password se fait dans le fichier `etc/ppp/pap-secrets` avec User-ID et Password.

Exemple pour le fournisseur Planet-Interkom :

```
# Secrets pour l'authentification en utilisant PAP
# client      server      secret      IP adresse
"anonymer"    *          "surfer"    *
```

Dans cet exemple, USER-ID est « anonymer ». l'accès au serveur-distant sera permis à tous (avec « * »). « surfer » est le mot de passe pour le fournisseur Planet-Interkom.

8.10.2. Configuration DNS

Vous pouvez utiliser le routeur-fli4l comme un serveur-DNS. Pourquoi cette fonction est utile et même obligatoire dans un LAN avec des ordinateurs Windows ? Tout est expliqué dans la documentation du paquetage « base ».

Le fichier-résolveur `etc/resolv.conf` contient les noms de domaine et les utilisateurs du serveur de nom. Vous pouvez voir ci-dessous le contenu (du « domain.de », il a seulement un endroit réservé pour indiquer la valeur de la variable de configuration `DOMAIN_NAME`) :

```
search domain.de
nameserver 127.0.0.1
```

Le serveur de noms dnsmasq est configuré en utilisant le fichier `etc/dnsmasq.conf`. Il boot à partir du script `rc001.base` et du `rc370.dnsmasq` qui sont générés automatiquement cela pourrait ressembler à ceci :

```

user=dns
group=dns
resolv-file=/etc/resolv.dnsmasq
no-poll
no-negcache
bogus-priv
log-queries
domain-suffix=lan.fli4l
local=/lan.fli4l/
domain-needed
expand-hosts
filterwin2k
conf-file=/etc/dnsmasq_dhcp.conf

```

8.10.3. Fichier hôte

Ce fichier contient l'ensemble des noms d'hôtes avec leurs adresses IP. Donc, ce classement est applicable seulement pour un fli4l local, pour les autres ordinateurs dans le LAN, il ne sera pas visible. Ce fichier n'est pas vraiment nécessaire, si un serveur DNS local est déjà démarré.

8.10.4. Configuration de imond

Les variables de configurations dans le fichier `etc/imond.conf`, doivent entre autre être activées `CIRC_x_NAME`, `CIRC_x_ROUTE`, `CIRC_x_CHARGEINT` et `CIRC_x_TIMES`. Il peut être constituée d'un maximum de 32 lignes (à l'exclusion des lignes de commentaires). Chaque ligne est composée de 8 colonnes :

1. Plage journalière
2. Plage horraire
3. Dispositif (`ippX` ou `isdX`)
4. Circuit pour Default-Route : « yes »/« no »
5. Numéro de Téléphone
6. Nom du Circuit
7. Prix de l'unité Téléphonique par Minute en EU
8. Compteur (interval d'unité-Tél) en seconde

Voici un exemple :

#day	hour	device	defroute	phone	name	charge	ch-int
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	Compuserve	0.019	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

D'autres explications peuvent être trouvées pour Least-Cost-Routing (ou routage au moindre coût) dans la documentation du paquetage « base ».

8.10.5. Le fichier `/etc/.profile`

Le fichier `/etc/.profile` contient les paramètres par défaut du Shell. Pour modifier le fichier `/etc/.profile` par défaut, il est nécessaire d'ajouter les paramètres en dessous des paramètres existant. Ces paramètres peuvent être utilisés pour paramétrer un raccourci d'une commande Prompt et ensuite (vous pourrez exécuter « ce raccourci »).

Important: *Ce fichier ne doit pas contenir le paramètre `exit` !*

Exemple :

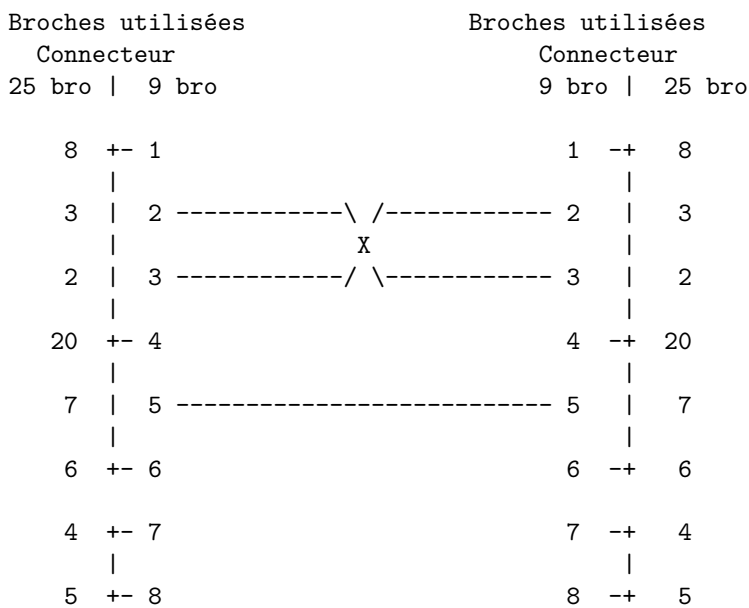
```
alias ll='ls -al'
```

A. Supplément du paquetage de Base

A.1. Câble Null-modem

Pour utiliser le programme facultatif [PPP](#) (Page 186) vous aurez besoin d'un câble null-modem.

Il faut avoir branchées au moins 3 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 28), [SER_CONSOLE_IF](#) (Page 28) et [SER_CONSOLE_RATE](#) (Page 28).

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 344) est utilisé entre l'émulation et le terminal ou le PC. Je déconseille 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 344).

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, lsmmod, lzcat, makedevs, md5sum, mdev, mkdir, mknod,
mkswap, modprobe, mount, mv, nameif, nice, nslookup, ping, ping6, poweroff,
ps, pscan, pwd, reboot, reset, rm, rmmmod, sed, seq, sh, sleep, sort, swapoff,
swapon, sync, sysctl, syslogd, tail, tar, test, top, tr, true, tty, umount,
uname, unlzma, unxz, unzip, uptime, usleep, vi, watch, xargs, xzcat, zcat
```

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 packaging-Opt à l'installation
- effectuer une installation sur le disque dur avec [Type B](#) (Page 15)

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 courriel: [frank\(at\)fli4l\(dot\)de](mailto:frank(at)fli4l(dot)de)

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) :

Eckhofer, Felix (*Documentation, Howtos*)

courriel: [felix\(at\)fli4l\(dot\)de](mailto:felix(at)fli4l(dot)de)

Franke, Roland (*OW, FBR*)

courriel: [fli4l\(at\)franke-prem\(dot\)de](mailto:fli4l(at)franke-prem(dot)de)

Hilbrecht, Claas (*VPN, Kernel*)

courriel: [claas\(at\)jucs-kramkiste\(dot\)de](mailto:claas(at)jucs-kramkiste(dot)de)

Klein, Sebastian (*Kernel, Wlan*)

courriel: -

Knipping, Michael (*Accounting*)

courriel: [fli4l\(at\)knibo\(dot\)de](mailto:fli4l(at)knibo(dot)de)

Krister, Stefan (*Opt-Cop, lcd4linux*)

courriel: [stefan\(dot\)krister\(at\)chreativ\(dot\)chaos\(dot\)de](mailto:stefan(dot)krister(at)chreativ(dot)chaos(dot)de)

Miksch, Gernot (*LCD*)

courriel: [gernot_miksch\(at\)gmx\(dot\)de](mailto:gernot_miksch(at)gmx(dot)de)

Schiefer, Peter (*fli4l-CD, Opt-Cop, site Web, gestion des versions*)

courriel: [peter\(at\)fli4l\(dot\)de](mailto:peter(at)fli4l(dot)de)

Schulz, Christoph (*FBR, IPv6, Kernel*)

courriel: [fli4l\(at\)kristov\(dot\)de](mailto:fli4l(at)kristov(dot)de)

Siebmans, Harvey (*Documentation*)

courriel: -

Spieß, Carsten (*Dsltool, Hwsupp, Rrdtool, Webgui*)

courriel: -

Vosselman, Arwin (*LZS-Compression, Documentation*)

courriel: [arwin\(at\)xs4all\(dot\)nl](mailto:arwin(at)xs4all(dot)nl)

A. Supplément du paquetage de Base

Wallmeier, Nico (*Windows-Imonc*)
courriel: [nico\(at\)fli4l\(dot\)de](mailto:nico(at)fli4l(dot)de)

Walter, Gerd (*UMTS*)
courriel: [fli4l\(at\)hgwb\(dot\)de](mailto:fli4l(at)hgwb(dot)de)

Walter, Oliver (*QoS*)
courriel: [owb\(at\)gmxdotde](mailto:owb(at)gmxdotde)

Weiler, Manuela (*Copie de CD, trésorière*)
courriel: -

Weiler, Marcel (*Gestion de la qualité*)
courriel: -

Eckhofer, Felix (*Dokumentation, Howtos*)
courriel: [felix\(at\)fli4l\(dot\)de](mailto:felix(at)fli4l(dot)de)

Franke, Roland (*OW, FBR*)
courriel: -

Hilbrecht, Claas (*VPN, Kernel*)
courriel: [claas\(at\)jucs-kramkiste\(dot\)de](mailto:claas(at)jucs-kramkiste(dot)de)

Klein, Sebastian (*Kernel, Wlan*)
courriel: -

Knipping, Michael (*Accounting*)
courriel: [fli4l\(at\)knibo\(dot\)de](mailto:fli4l(at)knibo(dot)de)

Krister, Stefan (*Opt-Cop, lcd4linux*)
courriel: [stefan\(dot\)krister\(at\)chreativ\(dot\)chaos\(dot\)de](mailto:stefan(dot)krister(at)chreativ(dot)chaos(dot)de)

Miksch, Gernot (*LCD*)
courriel: [gernot_miksch\(at\)gmxdotde](mailto:gernot_miksch(at)gmxdotde)

Schiefer, Peter (*fli4l-CD, Opt-Cop, Webseite, Releasemanagement*)
courriel: [peter\(at\)fli4l\(dot\)de](mailto:peter(at)fli4l(dot)de)

Schulz, Christoph (*FBR, IPv6, Kernel*)
courriel: [fli4l\(at\)kristov\(dot\)de](mailto:fli4l(at)kristov(dot)de)

Siebmans, Harvey (*Dokumentation*)
courriel: -

Spieß, Carsten (*Dsltool, Hwsupp, Rrdtool, Webgui*)
courriel: -

Vosselman, Arwin (*LZS-Kompression, Dokumentation*)
courriel: [arwin\(at\)xs4all\(dot\)nl](mailto:arwin(at)xs4all(dot)nl)

Wallmeier, Nico (*Windows-Imonc*)
courriel: [nico\(at\)fli4l\(dot\)de](mailto:nico(at)fli4l(dot)de)

Walter, Gerd (*UMTS*)
courriel: [fli4l\(at\)hgwb\(dot\)de](mailto:fli4l(at)hgwb(dot)de)

Walter, Oliver (*QoS*)
courriel: [owb\(at\)gmxdotde](mailto:owb(at)gmxdotde)

Weiler, Manuela (*CD-Versand, Kassenwart*)
courriel: -

Weiler, Marcel (*Qualitätsmanagement*)
courriel: -

L'équipe fli4l de testeurs et de traducteurs est formée (dans l'ordre alphabétique) :

Bußmann, Lars
Charrier, Bernard
Fischer, Joerg
Frauenhoff, Peter
Schliesing, Manfred
Schmitts, Jupp
Wolters, Forian

A.9.3. L'équipe de développeurs et de testeurs (qui ne sont plus actifs)

Arndt, Kai-Christian (*USB*)
Behrends, Arno (*Support*)
Bork, Thomas (*lpdsrv*)
Bauer, Jürgen (*LCD-Package*, *fliwiz*)
Blokland, Kees (*Traduction anglaise*)
Cerny, Carsten (*Site Web*, *fliwiz*)
Dawid, Oliver (*dhcp*, *uClibc*)
Ebner, Hannes (*QoS*)
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*)
Strigler, Stefan (*GTK-Imonc*, *Opt-DB*, *NG*)
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
Lehmen, 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 !

B. Supplément des paquetages optionnels

B.1. CHRONY - d'autre information sur applications Timewarps

Lorsque l'heure est différente et que chrony constate que l'heure est très loin de la réalité, chrony corrige l'heure et exécute un script en plusieurs étapes, pour informer les autres applications du changement de l'heure. Par exemple informer Imond du changement de l'heure, comme indiqué ci-dessous :

1. Inclure un script dans l'archive

Deux fichiers sont ajoutés l'archive Chrony :

```
start_imond yes etc/chrony.d/timewarp.sh mode=555 flags=sh
start_imond yes etc/chrony.d/timewarp100.imond mode=555 flags=sh
```

timewarp.sh exécute tous les scripts du même répertoire, dont les noms correspondent à timewarp<3 chiffres>.<nom>

2. Mettre à disposition le script

Chrony utilise le script ci-dessus dans l'archive :

```
# inform imond about time warp
imond-stat "adjust-time $timewarp 1"
```

Avec cela imond est informé du changement de l'heure et peut ajuster l'heure interne.

B.2. DSL - PPPD et filtre actif

Nous mettons à disposition dans fli4l l'expression suivant :

```
'outbound and not icmp[0] != 8 and not tcp[13] & 4 != 0'
```

Pour obtenir, en partant du principe, d'envoyer uniquement les paquets du réseau local à Internet, et de garder la connexion ouverte, avec quelques exceptions :

- *TCP-RST* : Réponses aux demandes de connexion qui on été refusées venant de l'extérieur, et ne pas mettre de Timeout (ou temps d'arrêt) derrière
- *ICMP* : Pour l'envoi de messages ICMP ne pas mettre également de Timeout derrière, à moins que vous envoyez une écho-requête.

Cette expression est généralement réalisé en PPPD, dans le filtrage de paquet par le Kernel. Cela ressemble à l'exemple ci-dessous :

```
#
# Expression: outbound and not icmp[0] != 8 and not tcp[13] & 4 != 0
#
(000) ldb      [0]
(001) jeq      #0x0          jt 17   jf 2
```



```
(002) ldh      [2]
(003) jeq      #0x21          jt 4    jf 18
(004) ldb      [13]
(005) jeq      #0x1          jt 6    jf 11
(006) ldh      [10]
(007) jset     #0x1fff        jt 18   jf 8
(008) ldx      4*([4]&0xf)
(009) ldb      [x + 4]
(010) jeq      #0x8          jt 18   jf 17
(011) jeq      #0x6          jt 12   jf 18
(012) ldh      [10]
(013) jset     #0x1fff        jt 18   jf 14
(014) ldx      4*([4]&0xf)
(015) ldb      [x + 17]
(016) jset     #0x4          jt 17   jf 18
(017) ret      #0
(018) ret      #4
```

B.3. DYNDNS

B.3.1. Ajouter un nouveau fournisseur DynDNS

L'ajout de nouveaux fournisseurs est en fait très facilement, étant donné que les scripts de mise à jour des fournisseurs d'accès sont séparées. Pour installer un nouveau fournisseur vous devez modifier le fichier suivant :

Fichier `opt/etc/dyndns/provider.NAME`

Ce fichier dans lequel est défini les paramètres, est utilisé pour la mise à jour de fournisseur d'accès spécial. Le plus souvent, le fichier ce compose seulement d'une liste de variables, il s'agit d'un script-shell tout a fait normal, cependant, des opérations plus complexes peuvent être exécutées, mais cela est rarement nécessaire. Dans ce fichier, les variables suivants peuvent être utilisés :

\$ip Adresse-IP de l'interface qui doit recevoir un nom d'hôte dynamique.

\$host Nom d'hôte complet, que l'utilisateur a donné dans sa configuration.

\$subdom Composants du nom d'hôte avec le point suivant (**name.provider.dom**)

\$domain Les deux dernières composantes du nom d'hôte (**name.provider.dom**)

\$provider Nom symbolique du fournisseur d'accès, que l'utilisateur a spécifié dans son fichier de configuration.

\$user Nom d'utilisateur pour ce service.

\$pass Mot de passe pour ce service.

Ces variables peuvent être écrites entre deux accolades, pour une séparation plus claire par rapport au texte, c'est à dire **\$ip** cela devient **\${ip}**. Lors de l'application de guillemets faite attention, de ne *pas* utiliser les guillemets simples avec les extensions de variables ci-dessus, mais utilisez les guillemets doubles. En règle générale, on peut donc dire : toujours utiliser des guillemets simples, mais dès que l'on utilise des extensions de variables, utiliser les guillemets doubles.

B. Supplément des paquetages optionnels

Les variables suivantes doivent être définies dans ce fichier, de manière à ce que le paquet sache comment mettre à jour le fournisseur d'accès correspondant :

provider_update_type Cela détermine la nature de la demande, qui est adressée au serveur du fournisseur d'accès. Pour le moment on prend en charge :

http On détermine l'appelle automatiquement une page Web, du fournisseur d'accès et ainsi on récupère la mise à jour de DynDNS actualisé.

netcat On détermine un simple texte, qui est envoyé au serveur du fournisseur d'accès pour déclencher la mise à jour.

gnudip Une simple authentification pour une mise à jour des procédures, ce lequel est exécuté plus de deux demandes HTTP.

provider_host Nom de l'hôte du fournisseur d'accès, qui est contacté pour la mise à jour.

provider_port Port de l'hôte du fournisseur d'accès, qui est concerné. Le port par défaut pour HTTP est 80.

Selon le type de mise à jour d'autres variables doivent être indiquées :

HTTP provider_url Ici on met URL relative (sans le nom d'hôte, mais avec un / au début du Script du fournisseur d'accès. Pour les exemples voir s'il vous plaît les fichiers des autres fournisseurs d'accès enregistrés.

provider_auth (optionnel) Les fournisseurs d'accès ont besoin pour l'ouverture de la session une Authentication basic, le format est "USER:PASSWORD".

Netcat provider_data On indique ici le texte qui sera envoyé au serveur du fournisseur d'accès. On peut indiquer par ex. `provider.DYNEISFAIR`.

GNUDip provider_script On indique ici le chemin d'accès au GNUDip script du serveur, ce qui ressemble généralement à quelque chose comme ça `'/cgi-bin/gdipupdt.cgi'`.

Fichier `opt/dyndns.txt`

Dans ce fichier une ou plusieurs lignes doivent être insérées pour le nouveau fournisseur d'accès. Le plus souvent une ligne suffit comme indiqué ci-dessous :

```
dyndns_%_provider    NAME    etc/dyndns/provider.NAME
```

Si l'Authentication Basic est utilisé pour le fournisseur d'accès HTTP, on a encore besoin de l'outil base64 :

```
dyndns_%_provider    NAME    files/usr/local/bin/base64
```

Si d'autres outils sont demandés, s'il vous plaît envoyer moi plus tôt un Mail, pour que je puisse les examiner et voir si ils conviennent au paquetage OPT_DYNDNS.

Fichier `check/dyndns.exp`

Vous devez indiquer dans ce fichier, à la ligne `DYNPROVIDER =` le nom du fournisseur d'accès, vous devez ajouter un trait vertical derrière des autres paramètre, de manière séparée le nouveau nom.

Fichier doc/<langue>/tex/dyndns/dyndns_main.tex

Enregistrer une nouvelle section dans la documentation. Là encore, les fournisseurs d'accès sont triés par ordre alphabétique selon le nom abrégé, c'est celui-ci que les utilisateurs paramètrent dans le fichier de Config.

Un macro-tableau est présent au début de la documentation, qui est suffisamment explicite.

B.3.2. Remerciment

Je tiens à remercier tout le monde qui a participé au lancement de ce paquet et une longue vie à ce paquet :

Thomas Müller (courriel: opt_dyndns@cs2h.cx) il a fait un excellent travail, sans lui il n'aurait pas été possible de proposer ce paquet dans la forme actuelle.

Je voudrais remercier Marcel Döring (courriel: m@rcel.to) qui a longtemps maintenu ce paquet.

Lors de l'élaboration du paquet de très nombreuses personnes m'ont aidé et ont trouvé des idées. Je tiens à remercier tous ces courageux volontaires.

En outre, je remercie Frank Meyer et le reste de l'équipe-fli4l de leur inlassable travail qui ont bricolé l'un des meilleurs routeur-disquette du monde (excusez c'est pas très sérieux ;-).

En outre, je tiens à remercier les personnes suivantes, pour leurs conseils, les rapports d'erreur des nouveaux fournisseurs, etc, ont participé au paquet :

- Paul Bischof pour le fournisseur AFRAID.
- Jens Fischer schrieb das Paket `opt_dtdns`, welches mich erst auf die Idee brachte, ein Paket für DynDNS.org zu schreiben.
- Till Jäger schrieb das Paket `opt_cjb`, welches in `opt_dyndns` übernommen habe.
- Tobias Gruetzmacher hat auf <http://portfolio16.de/index.de> Informationen zu weiteren DynDNS-Anbietern zusammengetragen, die hier verwendet werden.
- Die Anbieter dynamischer DNS, die auf ihren Webseiten zum Teil sehr gute, zum Teil weniger gute Beschreibungen des zu verwendenden Protokolls veröffentlicht haben.
- Die Programmierer diverser Update-Programme für DynDNS Anbieter, aus deren Code schamlos geklaut wurde. ;-)
- Heiko Ambos von dynaccess.de hat mich bei der Entwicklung der Unterstützung für diesen Anbieter unterstützt.
- Dennis Neuhäuser, der die Idee hatte, die Antworten der Dienste per Webserver verfügbar zu machen statt sie auf der Konsole auszugeben und auch gleich eine erste Implementation dafür geschickt hat.
- Lars Winkler der freundlicherweise die Änderungen, um das Paket unter 2.0pre2 zum Laufen zu bringen zur Verfügung gestellt hat.
- Markus Kraft und Tobias Gruetzmacher haben die Grundlage für die Anpassung an fli4l 2.0 gelegt.
- Diverse andere Leute haben mir ihre jeweilige Portierung auf fli4l 2.0 geschickt. Ich muss zu meiner Schande gestehen, dass ich mir die wenigsten davon angesehen habe.
- Georg Bärwald für die Daten zu Selfhost.de
- Mark C. Storck für die Daten zu Storck.org
- Arne Biermann für den Hinweis auf den Anbieter hn.org
- Detlef Paschke für die Daten zu dyn.ee und dyndns.dk
- Martin Kisser für seine Idee zum Vermeiden von Updates, wenn die IP sich nicht geändert

hat.

- Björn Hoffmann für die Daten von DnsArt.com
- Christian Busch für die Daten von no-ip.com.
- Ralf Gill für das Update der Daten von selfhost.de.
- Michael (HeinB) für eine weitere Möglichkeit sich mit fli4l selbst in den Fuss zu schiessen. ;-)
- Marcus Mönnig, dito.

B.3.3. Licence

Copyright ©2001-2002 Thomas Müller (courriel: opt_dyndns@s2h.cx)
Copyright ©2002-2003 Tobias Gruetzmacher (courriel: fli4l@portfolio16.de)
Copyright ©2004-201x L'équipe fli4l (courriel: team@fli4l.de)

Ce programme est un logiciel libre. Il est distribué selon les termes de la GNU License General Public comme prévu par la Free Software Foundation. Pour de plus amples informations sur la licence, reportez-vous s'il vous plaît à <http://www.gnu.org/licenses/gpl.txt>.

Ce programme est distribué dans l'espoir qu'il sera utile, mais SANS AUCUNE GARANTIE -. Sans même la garantie implicite de COMMERCIALISATION ou D'ADAPTATION À UN USAGE PARTICULIER Les détails peuvent être trouvés dans la GNU licence General Public.

Vous devriez avoir reçu une copie de la licence GNU General Public avec ce programme. Sinon, écrivez à :

Free Software Foundation Inc.
59 Temple Place
Suite 330
Boston MA 02111-1307 USA.

Le texte de la licence GNU General Public est également publié sur Internet <http://www.gnu.org/licenses/gpl.txt>. Une traduction non officielle en Allemand peut être trouvée ici <http://www.gnu.de/documents/gpl.de.html> et une traduction non officielle en Français ici <http://org.rodage.com//gpl-3.0.fr.html> Ces traductions sont cependant, les meilleures compréhensions de l'aide GPL, pour les droits juridiques vous devez utiliser la version anglaise.

B.4. EASYCRON - Complément de Crontab pour la phase de boot

Remarque : *Les informations qui suivent s'adressent uniquement aux développeurs pour le paquetage opt-fli4l du routeur.*

À partir de la version 2.1.7, le script rc de EasyCron met à disposition la fonction `add_crontab_entry()`. Lorsque vous utilisez cette fonction vous pourrez enregistrer d'autres scripts rc, de la position rc101 jusqu'à la position rc949. Ces entrées supplémentaires seront activées avec le démarrage du démon cron et la fin phase de boot.

`add_crontab_entry time command [custom]`

Avec `time` vous attribuez le temps d'exécution de cron, consultez le manuel crontab (5) (<http://linux.die.net/man/5/crontab>). Avec `command` vous utilisez la commande à exécuter. Avec le troisième paramètre `custom` qui est facultatif. Avec ce paramètre on peut installer des

extentions de commandes, selon votre convenance. Si vous placez plusieurs réglages, vous devez les séparer par `\\`. SVP ne changer **pas** la variable d'environnement `PATH`, car les entrées `crontab` ne fonctionnerons plus correctement.

```
#
# example I: normal use, 2 parameters
#
    crontime="0 5 1 * *"
    croncmd="rotate_i_log.sh"

    add_crontab_entry "$crontime" "$croncmd"

#
# end of example I
#

#
# example II: extended use, 3 parameters and
#               multiple environment values
#
    croncustom="source=/var/log/current \\ dest=/mnt/data/log"
    croncmd='cp $source $dest-`date +%Y%m%d`; > $source'
    crontime="59 23 * * *"

    add_crontab_entry "$crontime" "$croncmd" "$croncustom"

#
# end of example II
#
```

L'exactitude des paramètres doit être enregistrés dans le script qui est appelé.

B.5. HD - Rapport d'erreur sur les disques durs/CompactFlashes

Problème :

- Le routeur ne reconnaît pas le disque dur

Causes possibles :

- Avec la variable `OPT_HDDRV` on peut avoir besoin de pilotes supplémentaires pour le contrôleur HD.
- Le disque est mal configuré dans le BIOS.
- Le contrôleur est désactivé ou défectueux.
- Lors de l'installation il est spécifié un mauvais disque
- Le contrôleur n'est pas pris en charge par `fli4l`. Certains contrôleurs nécessitent des pilotes spécifiques qui ne sont pas inclus dans `fli4l`.

Problème :

- L'installation est interrompt
- Après une mise à jour du fichier `opt-Archives` le Routeur ne boot plus

- Il y a des messages d'erreurs au partitionnement ou au formatage du disque dur

Causes possibles :

- Le câble du disque dur IDE est peut être inadapté ou trop long.
- Sur les disques durs plus anciens, le réglage de la vitesse du transfert/PIO-mode dans le BIOS ou sur le contrôleur est peut-être trop rapide pour le disque.
- Le chipset est inadapté.

Remarques :

- Problèmes avec le DMA il peut éventuellement être résolu en indiquant la valeur `LIBATA_NODMA='no'~` (La valeur par défaut est 'yes') cela active le DMA avec les périphériques ATA.

Problème :

- Après l'installation, `fi4l` ne démarre pas sur le disque dur

Causes possibles :

- Si le démarrage à partir d'un module-CF a échoué vérifier si la CF a bien été reconnu en tant que LBA ou LARGE dans le Bios. Le réglage correct pour les petits modules de 512 Mo est NORMAL ou CHS.
- Si vous utilisez un contrôleur Adaptec 2940 avec un vieux BIOS et si l'affectation étendue pour les disques durs de plus de 1 Go est actif. Mettez à jour le BIOS de la carte SCSI ou affecter les micros-interrupteur.

En modifiant l'affectation les micros-interrupteur toutes les données du disque seront perdues !

Problème :

- Message d'erreur de Windows lors de la préparation d'une carte CF : «Le lecteur (X :) ne comporte pas de partition FAT. [Annuler]»

Causes possibles :

- La carte a été retirée du lecteur trop tôt / sans être démontage (ou éjecté). Windows n'avait pas terminé l'écriture et le système de fichiers est endommagé. Préparer à nouveau la carte CompactFlash avec `fi4l` via HD-install.

B.6. HTTPD

B.6.1. Paramètre supplémentaire

Normalement ces paramètres ne se trouvent pas dans le fichier de configuration, ils doivent être ajoutés si nécessaires.

HTTPD_USER Avec cette option, il est possible de faire fonctionner le serveur-Web avec les droits d'un autre utilisateur, en tant "root". Ceci est particulièrement utile, lorsque le serveur-Web est utilisé pour mettre à disposition d'autres pages que l'interface d'Admin. Attention : Il est possible que certains Scripts ne fonctionnent plus, car ils ont besoin d'accéder à certains fichiers de configuration. Le scripts par défaut de ce paquetage, fonctionne pour chaque utilisateur.

B.6.2. Observation générale

Si vous avez installé TELMOND, dans la fonction statut sur la page call de l'interface-Web, se trouve les numéros de téléphones des correspondants. Un classement par nom peut être

fait dans le fichier `opt/etc/phonebook`. Ce fichier a le même format que le fichier des numéros de téléphone d'IMONC. Ces annuaires téléphoniques peuvent être échangés entre IMONC et le routeur. Le format de chaque ligne du fichier est le suivant "Telefonnummer=Name [,fichier-WAV]" (sans les guillemets). Cependant le fichier WAV est utilisé uniquement par IMONC il est ignoré par le serveur-Web.

Le design des pages de l'interface-Web a été complètement remanié depuis la version 2.1.12 avec le fichier CSS. Les vieux navigateurs web pourraient avoir des problèmes avec cette version. Cette version a l'avantage, de pouvoir changer à volonté l'aspect de l'interface web, il suffit simplement d'adapter le fichier CSS (essentiellement le fichier `/opt/srv/www/css/main.css`).

Le paquetage serveur-Web a été produit par Thorsten Pohlmann (courriel: pohlmann@tetronik.com) et est maintenu à présent par Tobias Gruetzmacher (courriel: fli4l@portfolio16.de). La nouvelle conception (voir version 2.1.12) a été réalisée par Hummel Helmut (courriel: hh@fli4l.de).

B.7. HWSUPP - Paramètres dépendant du périphérique

B.7.1. Périphérique disponible pour la LED

Selon le matériel voir la variable `HWSUPP_TYPE` différents périphériques pour la LED seront disponibles. Pour le matériel ne figurant pas ici, les LEDs du clavier du PC peuvent être utilisées avec le [pc-générique](#).

Des périphériques supplémentaires pour la LED peuvent être disponibles par exemple pour le périphérique WLAN. Les noms valides des périphériques pour la LED peuvent d'être trouvés avec la commande `ls /sys/class/leds/`. En utilisant par exemple le ssh ou la console du routeur.

sim

Pour gérer une simulation de la LED, qui sera enregistré dans `syslog` :

```
- simu::1
- ...
- simu::8
```

PC générique

Les LEDs du clavier du PC :

```
- keyboard::scroll
- keyboard::caps
- keyboard::num
```

Générique acpi

Les LEDs du clavier du PC, comme dans [PC générique](#)

PC engines alix

- alix::1
- alix::2
- alix::3

PC engines apu

- apu::1
- apu::2
- apu::3

PC engines wrap

- wrap::1
- wrap::2
- wrap::3

Soekris net4801

- net48xx::error

Soekris net5501

- net5501::error

B.7.2. Bouton disponible du périphérique

Selon le matériel de la variable `HWSUPP_TYPE`, les périphériques GPIO suivants sont prédéfinis pour les boutons.

PC engines alix

- gpio::24

PC engines apu

- gpio::252

PC engines wrap

- gpio::40

soekris net5501

- gpio::25

Le bouton est nommé 'Reset' pour le matériel Soekris.
Attention : le bouton doit être activé dans le BIOS.

B.7.3. Note sur le matériel spécifique

PC engines alix

Un pilote défectueux du capteur de température LM90, entraîne une perte de contrôle de la température.

Pour contourner ce problème le pilote LM90 sera automatiquement déchargé et rechargé en utilisant une tâche de cron. Il est nécessaire que le paquetage easycron soit installé avec la variable (OPT_EASYCRON='yes').

B.8. HWSUPP - Exemple de configuration

B.8.1. PC générique

```
OPT_HWSUPP='yes'
HWSUPP_TYPE='generic-pc'

HWSUPP_WATCHDOG='no'
HWSUPP_CPUFREQ='no'

HWSUPP_LED_N='3'
HWSUPP_LED_1='ready'
HWSUPP_LED_1_DEVICE='keyboard::num'
HWSUPP_LED_2='online'
HWSUPP_LED_2_DEVICE='keyboard::caps'
HWSUPP_LED_3='wlan'
HWSUPP_LED_3_DEVICE='keyboard::scroll'
HWSUPP_LED_3_WLAN='wlan0'

HWSUPP_BUTTON_N='0'
```

B.8.2. PC engines APU

```
OPT_HWSUPP='yes'
HWSUPP_TYPE='pcengines-apu'

HWSUPP_WATCHDOG='yes'
HWSUPP_CPUFREQ='yes'
HWSUPP_CPUFREQ_GOVERNOR='ondemand'

HWSUPP_LED_N='3'
HWSUPP_LED_1='ready'
HWSUPP_LED_1_DEVICE='apu::1'
HWSUPP_LED_2='wlan'
HWSUPP_LED_2_DEVICE='apu::2'
HWSUPP_LED_2_WLAN='wlan0'
HWSUPP_LED_3='online'
HWSUPP_LED_3_DEVICE='apu::3'
```

```
HWSUPP_BUTTON_N='1'
HWSUPP_BUTTON_1='wlan'
HWSUPP_BUTTON_1_DEVICE='gpio::252'
HWSUPP_BUTTON_1_PARAM='wlan0'
```

B.8.3. PC engines APU avec GPIO

```
OPT_HWSUPP='yes'
HWSUPP_TYPE='pcengines-apu'

HWSUPP_WATCHDOG='yes'
HWSUPP_CPUFREQ='yes'
HWSUPP_CPUFREQ_GOVERNOR='ondemand'
```

```
HWSUPP_LED_N='5'
HWSUPP_LED_1='ready'
HWSUPP_LED_1_DEVICE='apu::1'
HWSUPP_LED_2='wlan'
HWSUPP_LED_2_DEVICE='apu::2'
HWSUPP_LED_2_WLAN='wlan0'
HWSUPP_LED_3='online'
HWSUPP_LED_3_DEVICE='apu::3'
HWSUPP_LED_4='trigger'
HWSUPP_LED_4_PARAM='phy0rx'
HWSUPP_LED_4_DEVICE='gpio::237'
HWSUPP_LED_5='trigger'
HWSUPP_LED_5_PARAM='phy0tx'
HWSUPP_LED_5_DEVICE='gpio::245'
```

```
HWSUPP_BUTTON_N='2'
HWSUPP_BUTTON_1='wlan'
HWSUPP_BUTTON_1_DEVICE='gpio::252'
HWSUPP_BUTTON_1_PARAM='wlan0'
HWSUPP_BUTTON_2='online'
HWSUPP_BUTTON_2_DEVICE='gpio::236'
```

B.9. HWSUPP - Séquence de clignotement de la LED

Les séquences de clignotants suivantes seront affichés pendant le processus de boot :

1.	⊗				⊗				...
2.	⊗	⊗			⊗	⊗			...
3.	⊗	⊗	⊗		⊗	⊗	⊗		...
4.	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	...

Lors de l'exécution de rc002.* à rc250.* La première séquence s'affiche,
(1 * flash - pause),
de rc250.* à rc500.* la seconde séquence (2 * flash - pause),
de rc500.* à rc750.* la troisième séquence et
de rc750.* jusqu'à la fin du processus de boot la quatrième séquence (clignotement continu).

B.10. HWSUPP - Conseil pour les développeurs de paquetage

Dans ce chapitre nous vous proposons une description pour ajouter une LED ou un bouton pour les développeurs qui veulent créer un paquetage¹.

B.10.1. Extension pour LED

Type de LED

Dans le fichier check/myopt.exp vous configurez une liste de plusieurs types de LEDs que vous pouvez indiquer dans la variable HWSUPP_LED_x.

Exemple :

```
+HWSUPP_LED_TYPE(OPT_MYOPT) = 'myopt'
                             : ', myopt'
```

Contrôle de paramètre

Dans le fichier check/myopt.ext vous configurez les paramètres qui seront vérifiés, vous pouvez les indiquer dans la variable HWSUPP_LED_x_PARAM.

Exemple :

```
if (opt_hwsupp)
then
    depends on hwsupp version 4.0

    foreach i in hwsupp_led_n
    do
        set action=hwsupp_led_%[i]
        set param=hwsupp_led_%_param[i]
        if (action == "myopt")
        then
            if (!(param =~ "(RE:MYOPT_LED_PARAM)"))
            then
                error "When HWSUPP_LED_\${i}='myopt', ...
                        must be entered in HWSUPP_LED_\${i}_PARAM"
            fi
        fi
    done
fi
```

1. Vous recherchez un endroit approprié dans ##HWSUPP## pour paramétrer le paquetage WLAN.

affichage de la LED

Quand vous définissez une LED dans le script pour un paquetage (par ex. `/usr/bin/<opt>_setled`) la requête `/usr/bin/hwsupp_setled <LED> <status>/` sera exécutée.

Le nombre de LED peut être lu dans le fichier `/var/run/hwsupp.conf`.

L'état de la LED peut être off, on ou blink.

Exemple :

```
if [ -f /var/run/hwsupp.conf ]
then
    . /var/run/hwsupp.conf
    [ 0$hwsupp_led_n -eq 0 ] || for i in `seq 1 $hwsupp_led_n`
    do
        eval action=\$hwsupp_led_${i}
        eval param=\$hwsupp_led_${i}_param
        if [ "$action" = "<opt>" ]
        then
            if [ <myexpression> ]
            then
                /usr/bin/hwsupp_setled $i on
            else
                /usr/bin/hwsupp_setled $i off
            fi
        fi
    done
fi
```

L'état actuel d'une LED peut être demandé avec la requête `/usr/bin/hwsupp_getled <LED>/`.
Le résultat sera off, on ou blink.

B.10.2. Extension pour le bouton

B.10.3. Action du bouton

Dans le fichier `check/myopt.exp` vous configurez une liste de plusieurs types de boutons que vous pouvez indiquer dans la variable `HWSUPP_BUTTON_x`.

Exemple :

```
+HWSUPP_BUTTON_TYPE(OPT_MYOPT) = 'myopt'
                                : ', myopt'
```

Contrôle de paramètre

Dans le fichier `check/myopt.ext` vous configurez les paramètres qui seront vérifiés, vous pouvez les indiquer dans la variable `HWSUPP_BUTTON_x_PARAM`.

Exemple :

```
if (opt_hwsupp)
then
    depends on hwsupp version 4.0
```

```
foreach i in hwsupp_button_n
do
    set action=hwsupp_buttonn_%[i]
    set param=hwsupp_button_%_param[i]
    if (action == "myopt")
    then
        add_to_opt "files/usr/bin/myopt_keyprog" "mode=555 flags=sh"
        if (!(param =~ "(RE:MYOPT_BUTTON_PARAM)"))
        then
            error "When HWSUPP_BUTTON_\${i}='myopt', ...
                    must be entered in HWSUPP_BUTTON_\${i}_PARAM"
        fi
    fi
done
fi
```

Fonction du bouton

Quand un bouton est pressé, le fichier de script `/usr/bin/myopt_keyprog` sera exécuté.
La valeur de la variable `HWSUPP_BUTTON_x_PARAM` est transmis.
Exemple :

```
##TODO## exemple
```

B.11. IPV6 - Connexion Internet en utilisant un tunnel IPv6 SixXS

Cette section décrit comment le paquetage IPv6 peut être utilisé pour vous aidez à créer un tunnel avec le fournisseur SixXS (<https://www.sixxs.net/>) pour connecter votre propre réseau à Internet IPv6.

B.11.1. Créer un compte

Tout d'abord, un compte SixXS doit être enregistré dans "Signup for new users". Une fois que vous avez surmonter cet obstacle, vous avez un nom d'utilisateur de la forme `SIxXS-YYYYYY` et un mot de passe associé. Ces données sont nécessaires plus tard pour la configuration du tunnel.

B.11.2. Configurer le tunnel

Préparations

Mais d'abord, vous devez vous connecter au tunnel. Cela sera fait après l'enregistrement via le menu "Request tunnel". Ici c'est important, car vous devez créer le type de tunnel dans cette deuxième partie de l'enregistrement, vous devez choisir "Dynamic IPv4 Endpoint using Heartbeat protocol", parce que cette configuration est directement pris en charge par `fli4l`. Troisième partie, le paramètre "Static IPv4 Endpoint" est également possible, si vous avez attribué une adresse IPv4 fixe, qui ne change jamais. Pour le moment le paramètre du tunnel

"Dynamic NAT-traversing IPv4 Endpoint using AYIYA" n'est pas supportée par le paquetage IPv6.

Une fois que vous avez indiqué les autres champs, sur l'emplacement du routeur, avec le paramètre "Next step" de la deuxième page, vous devez choisir un ou plusieurs PoP (Points of Presence), c'est important pour la construction du tunnel. Il faut prendre la personne qui est le plus proche, pour tester le tunnel avec les paquets IPv6 aussi efficacement que possible.

Si toutes les informations ont été enregistrées, vous pouvez activer "Place request for new tunnel", vous recevrez un E-Mail avec les données importantes nécessaires au tunnel. Il comportera :

1. le numéro d'identification du tunnel (T...)
2. Le nom des PoPs associés
3. L'adresse IPv4 du PoPs attribuée par ("SixXS IPv4")
4. l'adresse IPv6 locale du tunnel, y compris le masque de sous-réseau (typique à SixXS /64), c'est à dire l'adresse du routeur ("Votre" IPv6)
5. l'adresse IPv6 distant du tunnel, y compris le masque de sous-réseau (qui est identique au sous-réseau de l'adresse IPv6 locale), c.-à-d. l'adresse du PoP ("SixXS IPv6")

Configuration

Maintenant, le tunnel peut être configuré! En premier, vous devez placer la variable `IPV6_TUNNEL_N` sur "1" parce que vous devez construire exactement un tunnel :

```
IPV6_TUNNEL_N='1'
```

Les détails SixXS seront enregistrées dans la configuration IPv6 suivante :

1. Le numéro d'identification du tunnel est entré dans la variable `IPV6_TUNNEL_1_TUNNELID`.
2. ne pas enregistrer (le nom du PoP est sans intérêt).
3. L'adresse IPv4 du POP est à enregistrer dans la variable `IPV6_TUNNEL_1_REMOTEV4`.
4. L'adresse IPv6 locale du tunnel *avec* le masque sous-réseau doit être indiqué dans la variable `IPV6_TUNNEL_1_LOCALV6`.
5. L'adresse IPv6 distant du tunnel *sans* le masque sous-réseau doit être indiqué dans la variable `IPV6_TUNNEL_1_REMOTEV6`.

En outre, vous devez indiquer le nom d'utilisateur et le mot de passe dans la configuration du tunnel dans les variables `IPV6_TUNNEL_1_USERID` et `IPV6_TUNNEL_1_PASSWORD`. Enfin, il faut noter dans la variable `IPV6_TUNNEL_1_TYPE` que la configuration du tunnel est un tunnel SixXS :

```
IPV6_TUNNEL_1_TYPE='sixxs'
```

Si vous avez créé un PoP SixXS "deham01", avec l'adresse IPv4 212.224.0.188, les points d'arrêt de tunnel (distant) 2001:db8:900:551::1/64, le paramètre (local) 2001:db8:900:551::2/64, l'ID du tunnel "T1234", le nom d'utilisateur "USER1-SIXXS" et le mot de passe "sixxs" (n'utiliser *pas* ce mot de passe s'il vous plaît) alors configuration ressemblera à ceci :

```
IPV6_TUNNEL_N='1'  
IPV6_TUNNEL_1_LOCALV4='dynamic' # ou l'adresse fixe IPv4 locale  
IPV6_TUNNEL_1_REMOTEV4='212.224.0.188'  
IPV6_TUNNEL_1_LOCALV6='2001:db8:900:551::2/64'  
IPV6_TUNNEL_1_REMOTEV6='2001:db8:900:551::1'  
IPV6_TUNNEL_1_TYPE='sixxs'  
IPV6_TUNNEL_1_USERID='USER1-SIXXS'  
IPV6_TUNNEL_1_PASSWORD='sixxs'  
IPV6_TUNNEL_1_TUNNELID='T1234'
```

Test

Si vous avez réussi, vous pouvez commencer la mettre à jour et la configuration du routeur fli4l. Ensuite vous pouvez vous connecter au routeur (directement ou par ex. via SSH), vous devriez peut être pouvoir pingger l'extrémité du tunnel distant. Si cela a fonctionné, vous devriez voir les données circuler comme l'exemple ci-dessous :

```
garm 3.6.0-revXXXXX # ping -c 4 2001:db8:900:551:0:0:0:1  
PING 2001:db8:900:551::1 (2001:db8:900:551::1): 56 data bytes  
64 bytes from 2001:db8:900:551::1: seq=0 ttl=64 time=67.646 ms  
64 bytes from 2001:db8:900:551::1: seq=1 ttl=64 time=72.001 ms  
64 bytes from 2001:db8:900:551::1: seq=2 ttl=64 time=70.082 ms  
64 bytes from 2001:db8:900:551::1: seq=3 ttl=64 time=67.996 ms  
  
--- 2001:db8:900:551::1 ping statistics ---  
4 packets transmitted, 4 packets received, 0% packet loss  
round-trip min/avg/max = 67.646/69.431/72.001 ms
```

La dernière ligne est importante "0% packet loss", cela signifie que les paquets pingger ont tous reçu une réponse. Si vous n'avez aucune réponse de l'autre bout du tunnel, le résultat est différent :

```
garm 3.6.0-revXXXXX # ping -c 4 2001:db8:900:551:0:0:0:1  
PING 2001:db8:900:551::1 (2001:db8:900:551::1): 56 data bytes  
  
--- 2001:db8:900:551::1 ping statistics ---  
4 packets transmitted, 0 packets received, 100% packet loss
```

Ici aucun paquet pingger n'a reçu de réponse ("100% packet loss"). Cela signifie que soit la configuration n'est pas correcte, ou que la connexion au tunnel SixXS n'a pas encore été totalement établie. Dans le second cas, vous devriez attendre un certain temps parce que la configuration sur les POP peut ainsi prendre quelques heures. Si vous avez vérifié la configuration deux ou trois fois et que vous n'avez trouvé aucune erreur et un certain temps, c'est écoulé, vous devez contacter par e-mail SixXS et décrire le problème avec le plus de détail possible.

B.11.3. Configurer le sous-réseaux

Préparations

Si le tunnel fonctionne, vous avez réussi de paramétrer la première partie. Mais ce n'est pas encore terminé. Car le routeur n'a pas encore la possibilité de renvoyer les paquets IPv6 venant

d'Internet, vers l'hôte du réseau local. Il faut configurer seulement un sous-réseau IPv6 pour intégrer les hôtes du réseau local.

Voici une petite différence, mais significatif sur la façon de configurer un réseau IPv4 : À cause du manque d'adresse un seul hôte est directement connecté à Internet. Les autres hôtes dans le réseau local, transmettent seulement dans le réseau interne, c.-à-d. que les adresses de domaine 192.168.*.*, 172.16.*.*, bis 172.31.*.*, et 10.*.*.*, sont non routable (ils ne peuvent pas être connectés à Internet).² Avec les adresses IPv6 il y en a en abondance, il n'y a donc pas besoin d'utiliser des adresses réseau internes. Toutefois, en raison de la nature mondiale des sous-réseaux locaux on doit s'assurer que les adresses des hôtes locaux n'entrent pas en conflit avec les autres adresses sur Internet. Par conséquent, un sous-réseau doit être attribué par le fournisseur IPv6 afin d'éviter de telles collisions.

Sur SixXS, allez dans le menu "Request subnet". Ici vous devez principalement indiquer le tunnel à utiliser, qui est facile parce que jusqu'à présent un seul tunnel a été configuré. Ensuite, envoyer le formulaire avec "Place request for new subnet", vous recevrez un e-mail après un certain temps avec les informations suivantes :

1. L'adresse du sous-réseau IPv6, y compris le masque de sous-réseau ("Subnet IPv6")
2. L'adresse IPv6 du routeur dans le tunnel, où le sous-réseau transmettra vers SixXS ("routé vers")
3. L'adresse IPv4 du routeur ("Votre IPv4")³

Maintenant ces données sont suffisantes pour configurer fli4l avec votre propre sous-réseau IPv6. Cependant, il faut savoir une chose : le sous-réseau attribué est généralement très grand. SixXS attribue habituellement /48e de sous-réseaux, c.-à-d. dans une adresse IPv6 128 bits une partie est utilisé pour le préfixe du réseau avec /48 bits et le reste pour l'adressage des hôtes, donc $128 - 48 = 80$ Bits. Un tel grand sous-réseau a deux inconvénients majeurs. Le premier inconvénient est sa taille : on peut configurer $2^{80} \approx 1209$ trilliard d'adresses pour les hôtes. Il ne paraît pas conseillé de gérer cela sans utiliser une structure supplémentaire pour les adresses des hôtes. Le second inconvénient est plus grave : utiliser *l'autoconfiguration IPv6* dans un si grand sous-réseau, n'est pas possible. L'autoconfiguration est un processus, auquel un hôte IPv6 reçoit un protocole déterminé, avec un préfixe et l'adresse Mac de la carte réseau, ce bricolage crée l'adresse IPv6 de sous-réseau. L'adresse Mac se compose de six octets et avec l'aide du standard EUI-64, on peut étendre l'adresse à huit octets. Cela fait 64 bits, et puis c'est fini, il y a tout simplement pas assez d'informations pour arriver au 80 bits pour l'adressage de l'hôte.

Voici un long discours, en quelques mots : l'adressage du sous-réseau doit être plus petite, afin de pouvoir faire une configuration automatique. Il faudrait arriver à /64 bits pour adresser le sous réseau. C'est assez simple : Le masque de sous-réseau est facilement modifiable à /64. Par ex. le sous-réseau distribué par SixXS est 2001:db8:123::/48, ensuite le sous-réseau pour lequel vous souhaitez configurer fli4l est 2001:db8:123::/64. Cela signifie dans le détail qu'avec un masque de sous-réseau de /48, nous avons $2^{(64-48)} = 2^{16} = 65536$ sous-sous-réseaux à répartir, la premier adresse de sous-réseaux n'a que des zéros qui devrait être utilisé par fli4l. On doit se souvenir, que l'abréviation de l'adresse 2001:db8:123:: est en faite 2001:db8:123:0:0:0:0:0. Les trois premiers chiffres est la partie attribuée clairement par le fournisseur IPv6 pour alloués le sous-réseau, les quatre chiffres suivant est le sous-sous-réseau

2. voir RFC 1918 (<http://tools.ietf.org/html/rfc1918>) pour plus de détails

3. Si l'adresse IPv4 du routeur est dynamique, remplacer celle-ci par "heartbeat"

qui est représentée par des "zéros",⁴ ces quatre derniers chiffres sont réservés pour la partie hôte. Le résultat du (sous-) sous-réseau est encore énorme, il peut accueillir jusqu'à $2^{64} \approx 18,4$ billions d'hôtes. Merci à l'autoconfiguration IPv6, pour obtenir les adresses réelles qui ne seront jamais en contact. C'est une bonne chose ...

Configuration

Retour à la configuration ! Premièrement, la variable `IPV6_NET_N` est réglée sur "1", parce qu'un sous-réseau local IPv6 doit être mis en place. Vous indiquez l'adresse IPv6 de sous-réseau avec le masque de sous-réseau /64 dans la variable `IPV6_NET_1`. Mais ce n'est pas tout à fait juste : Car l'adresse IPv6 *de sous-réseau à l'intérieur du routeur*, n'a pas de préfixe de sous-réseau qui est affecté au tunnel. Celui-ci est configuré ailleurs, notamment dans la partie configuration du tunnel. Maintenant, vous devez placer le préfixe de sous-réseau dans la variable `IPV6_TUNNEL_1_PREFIX`

Si vous avez reçu de SIXXS /48 de sous-réseau IPv6 `2001:db8:123::/48`. Nous allons ajouter le nombre '456' dans la "1", adresse de sous-réseau à l'intérieur du routeur pour avoir un sous-sous-réseau de /64. nous allons ensuite créer la configuration suivante :

```
IPV6_NET_N='1'
IPV6_NET_1='0:0:0:456::1/64'          # Adresse-IPv6 du Routeur (aucun préfixe
                                      # de sous-réseau + Masque de sous-réseau)
IPV6_TUNNEL_1_PREFIX='2001:db8:123::/48' # /48-préfixe de sous-réseau
```

Il convient de noter, que les trois premiers zéros dans la variable `IPV6_NET_1` sont pour ainsi dire attribués au tunnel avec un préfixe de sous-réseau de /48. Conjointement avec le préfixe de sous-réseau de /48 qui est attribué par le fournisseur de tunnel, il sera notifié en sous-réseau de /64 par l'adresse `2001:db8:123:456::/64` et l'adresse IPv6 du routeur `2001:db8:123:456::1`.

Le nom de l'interface réseau sur laquelle ce sous-réseau est attaché n'est pas encore enregistré. Il faut affecter pour chaque sous-réseau une interface réseau. S'il n'y a pas plusieurs cartes réseau configurées dans le routeur, le nom de l'interface réseau de la manière typique sera "eth0" ou "wlan0" pour un périphérique wifi. Regardez en cas de doute le paramètre dans la variable `IP_NET_1_DEV` ("IP" sans "6") et recopiez simplement le contenu :

```
IPV6_NET_1_DEV='eth0' # Interface réseau pour le sous-réseau IPv6
```

Enfin, nous devons paramétrer toutes les variables pour l'autoconfiguration-IPv6 :

```
IPV6_NET_1_ADVERTISE='yes'          # /64-préfixe de sous-réseau et Route par défaut par RA
IPV6_NET_1_ADVERTISE_DNS='yes'      # Serveur-DNS par RA (exige le
                                      # DNS_SUPPORT_IPV6='yes'!)
IPV6_NET_1_DHCP='yes'               # Nom-Domaine et Serveur-DNS par DHCPv6
                                      # (celui-ci exige DNS_SUPPORT_IPV6='yes')
```

Les deux dernières variables ne sont pas nécessaires pour le fonctionnement du sous-réseau IPv6, mais très utiles. Elles sont utilisées pour diffuser des informations supplémentaires sur le sous-réseau IPv6, à savoir l'adresse IPv6 du serveur DNS et le nom de domaine utilisé. Le serveur DNS peut être même publié de deux façons différentes. Parce que les systèmes différents, mettent ici les préférences diverses à jour, il est avantageux à la fois d'activer le (RDNSS via les annonces de routeur et le DHCPv6).

4. Bien sûr, vous pouvez opter pour un sous-sous-réseau différent !

Test

Voici un exemple de l'ensemble de la configuration IPv6 (on suppose que la variable est sur `DNS_SUPPORT_IPV6='yes' !`) :

```
IPV6_NET_N='1'
IPV6_NET_1='0:0:0:456::1/64'    # Adresse-IPv6 du Routeur (aucun préfixe de
                                # sous-réseau + Masque de sous réseau)
IPV6_NET_1_DEV='eth0'          # Interface réseau pour le sous-réseau IPv6
IPV6_NET_1_ADVERTISE='yes'     # Préfixe du sous-réseau et la route par défaut par RA
IPV6_NET_1_ADVERTISE_DNS='yes' # Serveur-DNS par RA
IPV6_NET_1_DHCP='yes'          # Nom de domaine et Serveur-DNS par DHCPv6

IPV6_TUNNEL_N='1'
IPV6_TUNNEL_1_PREFIX='2001:db8:123::/48' # /48-Masque de sous-réseau
IPV6_TUNNEL_1_LOCALV4='dynamic'           # ou une adresse local IPv4 fixe
IPV6_TUNNEL_1_REMOTEV4='212.224.0.188'
IPV6_TUNNEL_1_LOCALV6='2001:db8:900:551::2/64'
IPV6_TUNNEL_1_REMOTEV6='2001:db8:900:551::1'
IPV6_TUNNEL_1_TYPE='sixxs'
IPV6_TUNNEL_1_USERID='USER1-SIXXS'
IPV6_TUNNEL_1_PASSWORD='sixxs'
IPV6_TUNNEL_1_TUNNELID='T1234'
```

Ce routeur fli4l est normalement configuré pour un hôte sous Windows 7, avec une configuration automatiquement des adresses IPv6 et une route par défaut, un serveur DNS et un domaine, rendant ainsi l'ordinateur apte pour l'IPv6. Maintenant vous pouvez par exemple tester avec un simple ping de l'ordinateur Windows vers Internet IPv6. Dans l'exemple suivant, nous essayons d'atteindre le serveur Web fli4l.de à partir d'un hôte Windows (nous utilisons directement l'adresse IPv6, afin de ne pas procéder à la fonctionnalité de correction DNS) :

```
C:\>ping 2001:bf0:c000:a::2:132
```

Ping exécuté pour 2001:bf0:c000:a::2:132 avec 32 octets de données~:

```
Réponse de 2001:bf0:c000:a::2:132: Temps=104ms
Réponse de 2001:bf0:c000:a::2:132: Temps=102ms
Réponse de 2001:bf0:c000:a::2:132: Temps=106ms
Réponse de 2001:bf0:c000:a::2:132: Temps=106ms
```

Statistique du Ping pour 2001:bf0:c000:a::2:132:

```
Paquets~: Envoyés = 4, Reçus = 4, Perdus = 0 (Perte 0%),
Durée approximative de boucle en milliseconde~:
    Minimum = 102ms, Maximum = 106ms, Meyenne = 104ms
```

Enfin, nous allons utiliser l'outil "tracert" (dans Windows : "tracert") pour déterminer si le paquet est routé correctement. Un exemple du réseau local de l'auteur est indiqué ci-dessous. Cela montre bien qu'un seul paquet vient du routeur fli4l (première ligne), puis descend de l'autre côté du tunnel (deuxième ligne) et enfin dans le monde de l'Internet IPv6 (à partir de la troisième ligne) :

```
C:\>tracert 2001:bf0:c000:a::2:132
```

Identification de la route vers virtualhost.in-berlin.de [2001:bf0:c000:a::2:132]
avec un maximum de 30 sauts~:

1	<1 ms	<1 ms	<1 ms	garm.example.org [2001:db8:13da:1::1]
2	70 ms	79 ms	71 ms	gw-1362.ham-01.de.sixxs.net [2001:db8:900:551::1]
3	67 ms	71 ms	76 ms	2001:db8:800:1003::209:55
4	68 ms	*	70 ms	2001:db8:1:0:87:86:71:240
5	69 ms	*	71 ms	2001:db8:1:0:87:86:77:67
6	72 ms	*	71 ms	2001:db8:1:0:86:87:77:81
7	71 ms	*	71 ms	2001:db8:1:0:87:86:77:83
8	90 ms	*	81 ms	2001:db8:1:0:87:86:77:62
9	84 ms	*	88 ms	2001:db8:1:0:87:86:77:71
10	99 ms	83 ms	83 ms	2001:db8:1:0:87:86:77:249
11	94 ms	87 ms	87 ms	20gigabitethernet4-3.core1.fra1.he.net [2001:7f8::1b1b:0:1]
12	96 ms	99 ms	99 ms	10gigabitethernet1-4.core1.ams1.he.net [2001:470:0:47::1]
13	105 ms	108 ms	107 ms	2001:7f8:8:5:0:73e6:0:1
14	106 ms	107 ms	104 ms	virtualhost.in-berlin.de [2001:bf0:c000:a::2:132]

La route est identifiée.

B.12. ISDN

B.12.1. Détails techniques sur la connexion et le routage ISDN

Ce chapitre concerne uniquement les personnes qui veulent comprendre ce qui se passe en interne du routeur ou qu'ils désirent faire une configuration spécifique ou encore de rechercher une solution à certain problème. Vous n'êtes *pas* obligé de lire ce chapitre, si cela ne vous intéresse pas.

Après avoir établi la connexion à votre FAI, qui a créé cette connexion avec le démon `ipppd` et avec l'interface qui a créé une nouvelle adresse IP. Le routage est produit automatiquement par le Kernel-Linux, pour accéder au Remote-IP (IP par défaut) et le masque de sous réseau. Le routage spécifique sera annulé si le masque de sous réseau est absent de la configuration. `ipppd` transmet à l'adresse Remote-IP à partir du masque sous réseau (Il utilise les différentes classe d'adresse A,B et C de masque de sous réseau). nous avons toujours eue des problèmes sur la disparition d'adresse et l'ajout automatique de nouvelle avec le routage :

- Les réseaux d'entreprise n'étaient plus accessibles, parce que le routage avait disparu ou avait été masqué par un nouveau routage installé.
- Une Interface était choisi apparemment sans raison, au lieu que le paquet aille sur la Routage par défaut le Kernel génère une nouvelle interface et le paquet se dirige vers celle-ci.
- ...

On essaie maintenant d'empêcher ces routages indésirables.

Pour cela nous avons modifié quelques paramètres :

- Remote-IP a été placé sur 0.0.0.0, si rien d'autre n'est spécifié. Ainsi disparaissent les problèmes de routages, lors de la configuration de l'interface mis en place par le Kernel.
- En plus le routage du circuit est sauvegardé dans un fichier cache

- Si un masque de sous réseau est paramétré pour le circuit, celui-ci passe par `ippdpd`, pour que l'adresse IP utilise l'interface configurée (afin de créer le routage).
- Après la connexion, le fichier cache du circuit est lu et les nouveaux paramètres enregistrés (Le Kernel efface le fichier et réenregistre avec les nouveaux paramètres de l'interface et de `ippdpd`).
- Puis l'interface sera de nouveau reconfigurée et le routage fonctionne à nouveau indépendamment de la configuration d'origine.

La configuration du circuit sera paramétré comme ci-dessous :

- Default route (ou routage par défaut)

`ISDN_CIRC_%_ROUTE='0.0.0.0'`

le circuit du lcr circuit est réglé il est "actif", un routage par défaut est installé sur son circuit (ou l'interface correspondant). A la connexion au FAI le routage de l'hôte apparaît, après la déconnexion l'état d'origine est reconstitué.

- Routage spécifique

`ISDN_CIRC_%_ROUTE='network/netmaskbits'`

On paramètre manuellement le routage du circuit (ou l'interface correspondant). A la connexion le Kernel efface et réenregistre le routage de l'hôte pour se connecter. Après la déconnexion l'état d'origine est reconstitué.

- remote ip (ou IP distant)

`ISDN_CIRC_%_REMOTE='ip address/netmaskbits'`

`ISDN_CIRC_%_ROUTE='network/netmaskbits'`

Pour cette configuration de l'interface le routage est utilisé pour un autre réseau distant (vous devez indiquer l'adresse IP et le masque de sous réseau). il se connecte à l'adresse IP spécifier (C'est-à-dire qu'il n'y a pas d'autres IP mise en place lors de la connexion) le routage reste valide.

Si toutefois vous appelez une autre IP, Le routage varie (nouvelle IP et masque de sous réseau)

Pour de nouveaux routage tous est dit plus haut.

Normalement cela doit résoudre provisoirement *tous* les problèmes qui se produisaient avec le routage spécial. Dans l'avenir la forme peut encore changer, mais rien ne changera dans le principe.

B.12.2. Messages d'erreur du sous-système ISDN (Documentation-i4l en Anglais)

Vous trouverez ci-dessous un extrait de la documentation `Isdn4Linux` (man 7 `isdn_cause`).

Cause messages are 2-byte information elements, describing the state transitions of an ISDN line. Each cause message describes its origination (location) in one byte, while the cause code is described in the other byte. Internally, when EDSS1 is used, the first byte contains the location while the second byte contains the cause code. When using 1TR6, the first byte contains the cause code while the location is coded in the second byte. In the Linux ISDN subsystem, the cause messages visible to the user are unified to avoid confusion. All user visible cause messages are displayed as hexadecimal strings. These strings always have the location coded in the first byte, regardless if using 1TR6 or EDSS1. When using EDSS1, these strings are preceded by the character 'E'.

LOCATION The following location codes are defined when using EDSS1 :

- 00 Message generated by user.

B. Supplément des paquetages optionnels

- 01 Message generated by private network serving the local user.
- 02 Message generated by public network serving the local user.
- 03 Message generated by transit network.
- 04 Message generated by public network serving the remote user.
- 05 Message generated by private network serving the remote user.
- 07 Message generated by international network.
- 0A Message generated by network beyond inter-working point.

CAUSE The following cause codes are defined when using EDSS1 :

- 01 Unallocated (unassigned) number.
- 02 No route to specified transit network.
- 03 No route to destination.
- 06 Channel unacceptable.
- 07 Call awarded and being delivered in an established channel.
- 10 Normal call clearing.
- 11 User busy.
- 12 No user responding.
- 13 No answer from user (user alerted).
- 15 Call rejected.
- 16 Number changed.
- 1A Non-selected user clearing.
- 1B Destination out of order.
- 1C Invalid number format.
- 1D Facility rejected.
- 1E Response to status enquiry.
- 1F Normal, unspecified.
- 22 No circuit or channel available.
- 26 Network out of order.
- 29 Temporary failure.
- 2A Switching equipment congestion.
- 2B Access information discarded.
- 2C Requested circuit or channel not available.
- 2F Resources unavailable, unspecified.
- 31 Quality of service unavailable.
- 32 Requested facility not subscribed.
- 39 Bearer capability not authorised.
- 3A Bearer capability not presently available.
- 3F Service or option not available, unspecified.
- 41 Bearer capability not implemented.
- 42 Channel type not implemented.
- 45 Requested facility not implemented.
- 46 Only restricted digital information bearer.
- 4F Service or option not implemented, unspecified.
- 51 Invalid call reference value.
- 52 Identified channel does not exist.
- 53 A suspended call exists, but this call identity does not.
- 54 Call identity in use.
- 55 No call suspended.
- 56 Call having the requested call identity.
- 58 Incompatible destination.
- 5B Invalid transit network selection.
- 5F Invalid message, unspecified.

- 60 Mandatory information element is missing.
- 61 Message type non-existent or not implemented.
- 62 Message not compatible with call state or message or message type non existent or not implemented.
- 63 Information element non-existent or not implemented.
- 64 Invalid information element content.
- 65 Message not compatible.
- 66 Recovery on timer expiry.
- 6F Protocol error, unspecified.
- 7F Inter working, unspecified.

B.13. UMTS

B.13.1. Matériel pris en charge

Les paquetages supplémentaires sont nécessaires pour le bon fonctionnement de UMTS. Pour les adaptateurs-USB le paquetage USB doit être activé avec `OPT_USB = 'yes'`. Ce paquetage prend en charge le matériel UMTS suivant~:

Matériel	tester	paquetage supplémentaire
Adaptateur Novatel~:		
Merlin U530	oui	PCMCIA, TOOLS (serial)
Merlin U630	non	PCMCIA, TOOLS (serial)
MC950D	oui	USB
OPTION d'adaptateur~:		
3G Datacard	non	PCMCIA, USB
GT 3G Quad	oui	PCMCIA, USB
GT Fusion	non	PCMCIA, USB
GT MAX HSUPA GX0301	oui	PCMCIA, USB
Pour ces quatre adaptateurs Cardbus vous devez indiquer PCMCIA_PCIC='yenta_socket'		
Icon 225 (GI0225)	oui	USB
Adaptateur Huawei~:		
E220, E230, E270	oui	USB
E510	oui	USB
E800	oui	USB
K3520	oui	USB
Adaptateur ZTE~:		
MF110	oui	USB
MF190	oui	USB

B.13.2. Si l'interface modem n'est pas activée

Il peut arriver, que certaines OPTIONS du Sticks UMTS ne soient pas activées pour l'interface du modem, ces options sont nécessaires pour le protocole pppd.

Exemple d'utilisation pour l'adaptateur GIO225

Contrôler avec la commande suivante~:

```
grep "" /sys/bus/usb/devices/*/tty*/hsotype
```

La sortie sur la console devrait ressembler à ceci~:

```
/sys/bus/usb/devices/2-1:1.0/tty/ttyHS0/hsotype:Control
/sys/bus/usb/devices/2-1:1.0/tty/ttyHS1/hsotype:Application
/sys/bus/usb/devices/2-1:1.1/tty/ttyHS2/hsotype:Diagnostic
```

Dans cette distribution l'entrée modem "hsotype:Modem" existent pas.

Maintenant, vous pouvez vérifier la configuration de l'interface avec la commande suivante~:

```
chat -e -t 1 '' "AT_OIFC?" OK >/dev/ttyHS0 </dev/ttyHS0
```

La sortie sur la console devrait ressembler à ceci~:

```
AT_OIFC?
_OIFC: 3,1,1,0
```

OK

ou à ceci~:

```
AT_OIFC?
_OIFC: 2,1,1,0
```

OK

Vous pouvez activer l'interface du modem avec la commande suivante~:

```
chat -e -t 1 '' "AT_OIFC=3,1,1,0" OK >/dev/ttyHS0 </dev/ttyHS0
```

Ensuite, retirer l'adaptateur et rebranchez le à nouveau, ensuite refaite un contrôle~:

```
grep "" /sys/bus/usb/devices/*/tty*/hsotype
```

Maintenant l'entrée modem existent.

```
/sys/bus/usb/devices/2-1:1.0/tty/ttyHS0/hsotype:Control
/sys/bus/usb/devices/2-1:1.0/tty/ttyHS1/hsotype:Application
/sys/bus/usb/devices/2-1:1.1/tty/ttyHS2/hsotype:Diagnostic
/sys/bus/usb/devices/2-1:1.2/tty/ttyHS3/hsotype:Modem
```

B.14. SRC - Développement de son propre packaging

Si vous voulez développer votre propre packaging, on pourrait appeler (packaging) un programme, qui n'est pas intégré dans le FBR, qui sera compilé pour fli4l, vous devez écrire au moins deux fichiers : un fichier make et un fichier pour la description de la configuration.

(TODO)

Comment ces fichiers sont construits et comment peut on écrire un dérivé de son propre Makefiles, vous trouverez une description dans la documentation de uClibc-Buildroots dans <http://buildroot.uclibc.org/downloads/manual/manual.html#adding-packages>

B.15. Différences entre la version 3.10.4 et 3.6.2

Paquetage ADVANCED_NETWORKING

Nouvelles variables

BCRELAY_N (Page 75)
BCRELAY_x_IF_N (Page 75)
BCRELAY_x_IF_x (Page 75)
ETHTOOL_DEV_N (Page 84)
ETHTOOL_DEV_x (Page 85)
ETHTOOL_DEV_x_OPTION_N (Page 85)
ETHTOOL_DEV_x_OPTION_x_NAME (Page 85)
ETHTOOL_DEV_x_OPTION_x_VALUE (Page 85)
OPT_BCRELAY (Page 75)
OPT_ETHTOOL (Page 84)
OPT_IPSET (Page ??)

Variables supprimées

Paquetage BASE

Nouvelles variables

ARCH (Page ??)
COMP_TYPE_ROOTFS (Page 25)
DEBUG_IPTABLES (Page 29)
FLI4L_BUILDDATE (Page ??)
FLI4L_BUILDDIR (Page ??)
FLI4L_BUILDTIME (Page ??)
FLI4L_VERSION (Page ??)
LIBATA_DMA (Page 24)
PF_DNS_EXCEPTIONS (Page ??)
PF_INPUT_ICMP_ECHO_REQ_SIZE (Page 51)
PF_OUTPUT_ACCEPT_DEF (Page 53)
PF_OUTPUT_CT_ACCEPT_DEF (Page 65)
PF_OUTPUT_CT_N (Page 65)
PF_OUTPUT_CT_x (Page 65)
PF_OUTPUT_CT_x_COMMENT (Page 65)
PF_OUTPUT_LOG (Page 53)
PF_OUTPUT_LOG_LIMIT (Page 53)
PF_OUTPUT_N (Page 53)
PF_OUTPUT_POLICY (Page 53)
PF_OUTPUT_REJ_LIMIT (Page 53)
PF_OUTPUT_UDP_REJ_LIMIT (Page 53)
PF_OUTPUT_x (Page 53)

Variables supprimées

COMPRESS_KERNEL
COMPRESS_OPT
COMPRESS_ROOTFS
DENY_ICMP
DMZ_LOG
DMZ_NAT
DMZ_ORANGE_RED_N
DMZ_ORANGE_RED_x
DMZ_ORANGE_ROUTER_N
DMZ_ORANGE_ROUTER_x
DMZ_RED_DEV
FORWARD_DENY_PORT_N
FORWARD_DENY_PORT_x
FORWARD_HOST_N
FORWARD_HOST_WHITE
FORWARD_HOST_x
IMOND_USE_ORIG
INPUT_ACCEPT_PORT_N
INPUT_ACCEPT_PORT_x
INPUT_POLICY
IP_NET_x_TYPE
MASQ_NETWORK

B. Supplément des paquetages optionnels

[PF_OUTPUT_x_COMMENT](#) (Page 53)
[PF_PREROUTING_CT_ACCEPT_DEF](#) (Page 65)
[PF_PREROUTING_CT_N](#) (Page 65)
[PF_PREROUTING_CT_x](#) (Page 65)
[PF_PREROUTING_CT_x_COMMENT](#) (Page 65)
[SYSLOGD_ROTATE_AT_SHUTDOWN](#) (Page 72)

[OPT_DMZ](#)
[OPT_EVSS](#)
[OPT_MOUNTFLOPPY](#)
[PACKETFILTER_LOG](#)
[PACKETFILTER_LOG_LEVEL](#)
[PF_NEW_CONFIG](#)
[PRESERVE](#)
[ROUTE_NETWORK](#)
[TRUSTED_NETS](#)

Paquetage DHCP_CLIENT

Nouvelles variables

[DHCP_CLIENT_x_WAIT](#) (Page 90)

Variables supprimées

Paquetage DNS_DHCP

Nouvelles variables

[DNS_AUTHORITATIVE](#) (Page 95)
[DNS_AUTHORITATIVE_IPADDR](#) (Page 95)
[DNS_AUTHORITATIVE_NS](#) (Page 95)
[DNS_BIND_INTERFACES](#) (Page 92)
[DNS_FORWARD_LOCAL](#) (Page ??)
[DNS_LISTEN_N](#) (Page 92)
[DNS_LISTEN_x](#) (Page 92)
[DNS_LOCAL_HOST_CACHE_TTL](#) (Page 94)
[DNS_ZONE_DELEGATION_N](#) (Page 96)
[DNS_ZONE_DELEGATION_x_DOMAIN_N](#) (Page ??)
[DNS_ZONE_DELEGATION_x_DOMAIN_x](#) (Page ??)
[DNS_ZONE_DELEGATION_x_NETWORK_N](#) (Page ??)
[DNS_ZONE_DELEGATION_x_NETWORK_x](#) (Page ??)
[DNS_ZONE_DELEGATION_x_UPSTREAM_SERVER_N](#) (Page ??)
[DNS_ZONE_DELEGATION_x_UPSTREAM_SERVER_x_IP](#) (Page ??)
[DNS_ZONE_DELEGATION_x_UPSTREAM_SERVER_x_QUERYSOURCEIP](#) (Page ??)
[DNS_ZONE_NETWORK_N](#) (Page 95)
[DNS_ZONE_NETWORK_x](#) (Page 95)
[HOST_x_IP6_NET](#) (Page ??)
[OPT_YADIFA](#) (Page ??)
[YADIFA_ALLOW_QUERY_N](#) (Page 102)
[YADIFA_ALLOW_QUERY_x](#) (Page ??)
[YADIFA_LISTEN_N](#) (Page 102)
[YADIFA_LISTEN_x](#) (Page ??)
[YADIFA_SLAVE_ZONE_N](#) (Page 102)
[YADIFA_SLAVE_ZONE_x](#) (Page 102)
[YADIFA_SLAVE_ZONE_x_ALLOW_QUERY_N](#) (Page 102)
[YADIFA_SLAVE_ZONE_x_ALLOW_QUERY_x](#) (Page 102)
[YADIFA_SLAVE_ZONE_x_MASTER](#) (Page 102)
[YADIFA_SLAVE_ZONE_x_USE_DNSMASQ_ZONE_DELEGATION](#) (Page ??)
[YADIFA_USE_DNSMASQ_ZONE_DELEGATION](#) (Page ??)

Variables supprimées

[DNS_LISTENIP_N](#)
[DNS_LISTENIP_x](#)
[DNS_SPECIAL_N](#)
[DNS_SPECIAL_x_DNSIP](#)
[DNS_SPECIAL_x_DOMAIN](#)
[DNS_SPECIAL_x_NETWORK](#)

Paquetage DSL

Nouvelles variables

[FRITZDSL_FILTER_EXPR](#) (Page ??)
[PPPOE_FILTER_EXPR](#) (Page ??)
[PPTP_FILTER_EXPR](#) (Page ??)

Variables supprimées

OPT_PFC
OPT_PPPOE_CIRC
PPPOE_CIRC_N
PPPOE_CIRC_x_CHARGEINT
PPPOE_CIRC_x_DEBUG
PPPOE_CIRC_x_ETH
PPPOE_CIRC_x_FILTER
PPPOE_CIRC_x_HUP_TIMEOUT
PPPOE_CIRC_x_MRU
PPPOE_CIRC_x_MTU
PPPOE_CIRC_x_NAME
PPPOE_CIRC_x_PASS
PPPOE_CIRC_x_TIMES
PPPOE_CIRC_x_TYPE
PPPOE_CIRC_x_USEPEERDNS
PPPOE_CIRC_x_USER

Paquetage DYNDNS

Nouvelles variables

[DYNDNS_LOGINTIME](#) (Page 116)
[DYNDNS_x_LOGIN](#) (Page 116)
[OPT_STUN](#) (Page ??)
[STUN_SERVER_N](#) (Page ??)
[STUN_SERVER_x](#) (Page ??)

Variables supprimées

Paquetage HD

Nouvelles variables

Variables supprimées

HDIT_DATA
HDIT_POWEROFF
HDIT_SIZES
OPT_HDINSTALL_TEST

Paquetage HTTPD

Nouvelles variables

Variables supprimées

[HTTPD_ARPING_IGNORE_N](#) (Page 124)
[HTTPD_ARPING_IGNORE_x](#) (Page 124)

Paquetage IPV6

Nouvelles variables

Variables supprimées

[IPV6_NET_x_ADVERTISE_PREF_LIFETIME](#)
(Page ??)
[IPV6_NET_x_ADVERTISE_VALID_LIFETIME](#)
(Page ??)
[PF6_DNS_EXCEPTIONS](#) (Page ??)
[PF6_INPUT_ICMP_ECHO_REQ_LIMIT](#) (Page ??)
[PF6_INPUT_ICMP_ECHO_REQ_SIZE](#) (Page 140)
[PF6_LOG_LEVEL](#) (Page 139)
[PF6_OUTPUT_ACCEPT_DEF](#) (Page 142)
[PF6_OUTPUT_CT_ACCEPT_DEF](#) (Page ??)
[PF6_OUTPUT_CT_N](#) (Page ??)
[PF6_OUTPUT_CT_x](#) (Page ??)
[PF6_OUTPUT_CT_x_COMMENT](#) (Page ??)
[PF6_OUTPUT_LOG](#) (Page 142)
[PF6_OUTPUT_LOG_LIMIT](#) (Page 142)
[PF6_OUTPUT_N](#) (Page 143)
[PF6_OUTPUT_POLICY](#) (Page 142)
[PF6_OUTPUT_REJ_LIMIT](#) (Page 142)
[PF6_OUTPUT_UDP_REJ_LIMIT](#) (Page 142)
[PF6_OUTPUT_x](#) (Page 143)
[PF6_OUTPUT_x_COMMENT](#) (Page 143)
[PF6_POSTROUTING_N](#) (Page 144)
[PF6_POSTROUTING_x](#) (Page 144)
[PF6_POSTROUTING_x_COMMENT](#) (Page 144)
[PF6_PREROUTING_CT_ACCEPT_DEF](#) (Page ??)
[PF6_PREROUTING_CT_N](#) (Page ??)
[PF6_PREROUTING_CT_x](#) (Page ??)
[PF6_PREROUTING_CT_x_COMMENT](#) (Page ??)
[PF6_PREROUTING_N](#) (Page 144)
[PF6_PREROUTING_x](#) (Page 144)
[PF6_PREROUTING_x_COMMENT](#) (Page 144)

Paquetage ISDN

Nouvelles variables

Variables supprimées

[ISDN_FILTER_EXPR](#) (Page ??)
[OPT_RCAPID](#) (Page 161)
[RCAPID_PORT](#) (Page 161)
[TELMOND_CAPI_CTRL_N](#) (Page 160)
[TELMOND_CAPI_CTRL_x](#) (Page 160)

Paquetage OPENVPN

Nouvelles variables

Variables supprimées

[OPENVPN_x_IPV6](#) (Page ??)
[OPENVPN_x_LOCAL_VPN_IPV6](#) (Page 168)
[OPENVPN_x_PF6_FORWARD_N](#) (Page 177)
[OPENVPN_x_PF6_FORWARD_x](#) (Page 177)
[OPENVPN_x_PF6_INPUT_N](#) (Page 177)
[OPENVPN_x_PF6_INPUT_x](#) (Page 177)
[OPENVPN_x_REMOTE_VPN_IPV6](#) (Page 168)
[OPENVPN_x_RENEG_SEC](#) (Page ??)

[OPENVPN_DEFAULT_PF_DMZ_TYPE](#)
[OPENVPN_VERSION](#)
[OPENVPN_x_PF_DMZ_TYPE](#)

Paquetage PCMCIA

Nouvelles variables

Variables supprimées

PCMCIA_CARDMGR_OPTS
PCMCIA_CORE_OPTS
PCMCIA_PCIC_EXTERN

Paquetage PROXY

Nouvelles variables

Variables supprimées

[IGMPPROXY_ALT_N](#) (Page 199)
[IGMPPROXY_ALT_NET_x](#) (Page 199)
[IGMPPROXY_DEBUG](#) (Page 199)
[IGMPPROXY_DEBUG2](#) (Page 199)
[IGMPPROXY_DOWNLOAD_DEV](#) (Page 199)
[IGMPPROXY_QUICKLEAVE_ON](#) (Page 199)
[IGMPPROXY_UPLOAD_DEV](#) (Page 199)
[IGMPPROXY_WLIST_N](#) (Page 200)
[IGMPPROXY_WLIST_NET_x](#) (Page 200)
[OPT_IGMPPROXY](#) (Page 194)
[OPT_KAMAILIO](#) (Page ??)
[OPT_RTTPROXY](#) (Page ??)
[OPT_SIPROXD](#) (Page ??)
[OPT_STUNNEL](#) (Page 201)
[STUNNEL_DEBUG](#) (Page 201)
[STUNNEL_N](#) (Page 202)
[STUNNEL_x_ACCEPT](#) (Page 202)
[STUNNEL_x_ACCEPT_IPV4](#) (Page 203)
[STUNNEL_x_ACCEPT_IPV6](#) (Page 203)
[STUNNEL_x_CERT_CA_FILE](#) (Page 204)
[STUNNEL_x_CERT_FILE](#) (Page 204)
[STUNNEL_x_CERT_VERIFY](#) (Page 204)
[STUNNEL_x_CLIENT](#) (Page 202)
[STUNNEL_x_CONNECT](#) (Page 203)
[STUNNEL_x_DELAY_DNS](#) (Page 204)
[STUNNEL_x_NAME](#) (Page 202)
[STUNNEL_x_OUTGOING_IP](#) (Page 203)

Paquetage QOS

Nouvelles variables

Variables supprimées

[QOS_CLASS_x_LABEL](#) (Page 210)

Paquetage SSHD

Nouvelles variables

Variables supprimées

OPT_SCP

Paquetage TOOLS

Nouvelles variables

[FTP_PF_ENABLE_ACTIVE](#) (Page 227)
[OPT_ATH_INFO](#) (Page 230)
[OPT_DIG](#) (Page 227)
[OPT_I2CTOOLS](#) (Page 230)
[OPT_IWLEEPROM](#) (Page 230)
[OPT_NGREP](#) (Page 228)
[OPT_OPENSSL](#) (Page 231)
[OPT_REAVER](#) (Page 231)
[OPT_SOCAT](#) (Page 229)

Variables supprimées

[OPT_ARP](#)
[OPT_BCRELAY](#)
[OPT_ETHOTOOL](#)
[OPT_NETSTAT](#)
[OPT_SERIAL](#)
[OPT_TRACEROUTE](#)
[OPT_TRACEROUTE6](#)
[WGET_SSL](#)

Paquetage USB

Nouvelles variables

Variables supprimées

[USB_LOWLEVEL](#)

B.16. Différences entre la version 3.10.4 et 3.10.3

Table des figures

3.1. Structure du Filtrage de paquets	41
3.2. Structure du répertoire fli4l	49
4.1. Exemple de configuration-VPN — Tunnel entre deux routeurs	163
4.2. fli4l répertoire OpenVPN avec les fichiers *.secret	166
4.3. Aperçut des connexions	179
4.4. Vue détaillée d'une connexion (gestion de clé)	180
4.5. fli4l avec la configuration par défaut	195
4.6. fli4l avec la configuration IPTV	196
4.7. exemple 1	215
4.8. exemple 2	217
4.9. exemple 3	219
4.10. Structure des répertoires fli4l	223
5.1. Paramètre	260
5.2. Paramètre pour la mise à jour	261
5.3. Paramètre pour pré-installation du DD	262

Liste des tableaux

3.1.	Aperçu des paquetages supplémentaires	17
3.2.	Réglage Automatique du nombre de connexions maximum	26
3.3.	Tableau des pilotes LAN ; légende : v=virt, n=nonfree, vn=virt-nonfree	36
3.4.	Tableau des pilotes WLAN ; légende : v=virt, n=nonfree, vn=virt-nonfree	38
3.5.	Action des règles du filtrage de paquets	43
3.6.	Restrictions de la source et de destination dans les règles de filtrage de paquets	43
3.7.	Restriction des règles sur de filtrage de paquets	45
3.8.	Modèles inclus dans fli4l de base	48
3.9.	Disponibilité de Conntrack Helpers dans le filtrage de paquets	65
3.10.	Format du fichier Log d'Imond	69
4.1.	Les valeurs de la variable BRIDGE_DEV_x_DEV_x_PATHCOST sont en fonction de la bande passante	84
4.2.	Type de création de paquet-pppoe	107
4.3.	Bande passante et charge CPU pour pppoe	108
4.4.	Cartes-Fritz	109
4.5.	Type de Modem-PPTP	110
4.6.	Exemple de configuration pour l'installation du support	119
4.7.	Exemple d'installation pour le Type A ou B	120
4.9.	Commande du Webgui-OpenVPN	179
4.10.	Paramètre MTU des différentes versions OpenVPN.	182
4.11.	Paramètre MTU des différentes versions pour le routeur-fli4l.	182
4.12.	Configuration d'OpenVPN avec 2 routeurs-fli4l	182
4.13.	Configuration d'OpenVPN avec 2 routeurs-fli4l leurs réseaux ont une connexion sans fil et utilise un Bridge	183
4.14.	Configuration d'OpenVPN avec 2 routeurs-fli4l leurs réseaux ont une connexion sans fil et utilise un Bridge, configuration dans advanced_networking.	183
4.15.	Configuration d'OpenVPN avec 2 routeurs-fli4l leurs réseaux ont une connexion sans fil et utilise un Bridge, configuration dans (base.txt).	183
4.16.	Configuration d'OpenVPN pour un ordinateur Windows avec GPRS	184
4.17.	OpenVPN sécurisé dans un WLAN	186
8.1.	Paramètres pour mkfli4l	290
8.2.	Options pour les fichiers	293
8.3.	Expressions logiques	315
8.4.	Les fonctions pour le script cgi-helper	328

Index

base.txt, 17
BCRELAY_N, 75
BCRELAY_x_IF_N, 75
BCRELAY_x_IF_x, 75
BEEP, 27
BONDING_DEV_N, 75
BONDING_DEV_x_ARP_INTERVAL, 78
BONDING_DEV_x_ARP_IP_-TARGET_N, 79
BONDING_DEV_x_ARP_IP_-TARGET_x, 79
BONDING_DEV_x_DEV_N, 77
BONDING_DEV_x_DEV_x, 77
BONDING_DEV_x_DEVNAME, 76
BONDING_DEV_x_DOWNDELAY, 78
BONDING_DEV_x_LACP_RATE, 78
BONDING_DEV_x_MAC, 77
BONDING_DEV_x_MIIMON, 78
BONDING_DEV_x_MODE, 76
BONDING_DEV_x_PRIMARY, 78
BONDING_DEV_x_UPDELAY, 78
BONDING_DEV_x_USE_CARRIER, 78
BOOT_TYPE, 23
BOOTMENU_TIME, 24
BRIDGE_DEV_BOOTDELAY, 81
BRIDGE_DEV_N, 81
BRIDGE_DEV_x_AGING, 82
BRIDGE_DEV_x_DEV_N, 81
BRIDGE_DEV_x_DEV_x_DEV, 81
BRIDGE_DEV_x_DEV_x_-PATHCOST, 83
BRIDGE_DEV_x_DEV_x_PORT_-PRIORITY, 83
BRIDGE_DEV_x_DEVNAME, 81
BRIDGE_DEV_x_FORWARD_DELAY, 82
BRIDGE_DEV_x_GARBAGE_-COLLECTION_INTERVAL, 82
BRIDGE_DEV_x_HELLO, 83
BRIDGE_DEV_x_MAX_MESSAGE_-AGE, 83
BRIDGE_DEV_x_NAME, 81
BRIDGE_DEV_x_PRIORITY, 82
BRIDGE_DEV_x_STP, 82
BUILDDIR, 263
CHRONY_BIOS_TIME, 88
CHRONY_LOG, 88
CHRONY_TIMESERVER_N, 88
CHRONY_TIMESERVER_x, 88
CHRONY_TIMESERVICE, 88
COMP_TYPE_OPT, 25
COMP_TYPE_ROOTFS, 25
COMPRESS_KERNEL, 376
COMPRESS_OPT, 376
COMPRESS_ROOTFS, 376
CONSOLE_BLANK_TIME, 27
DEBUG_ENABLE_CORE, 29, 322
DEBUG_IP, 29, 322
DEBUG_IPTABLES, 29
DEBUG_IPUP, 323
DEBUG_KEEP_BOOTLOGD, 323
DEBUG_MDEV, 29, 323
DEBUG_MODULES, 28
DEBUG_STARTUP, 28
DENY_ICMP, 376
DEV_MTU_N, 80
DEV_MTU_x, 80
DHCP_CLIENT_DEBUG, 90
DHCP_CLIENT_N, 89
DHCP_CLIENT_TYPE, 89
DHCP_CLIENT_x_HOSTNAME, 90
DHCP_CLIENT_x_IF, 89

- DHCP_CLIENT_x_ROUTE, 89
- DHCP_CLIENT_x_STARTDELAY, 90
- DHCP_CLIENT_x_USEPEERDNS, 90
- DHCP_CLIENT_x_WAIT, 90
- DHCP_DENY_MAC_N, 100
- DHCP_DENY_MAC_x, 100
- DHCP_EXTRA_RANGE_N, 99
- DHCP_EXTRA_RANGE_x_DEVICE, 100
- DHCP_EXTRA_RANGE_x_DNS_SERVER, 99
- DHCP_EXTRA_RANGE_x_END, 99
- DHCP_EXTRA_RANGE_x_GATEWAY, 99
- DHCP_EXTRA_RANGE_x_MTU, 99
- DHCP_EXTRA_RANGE_x_NETMASK, 99
- DHCP_EXTRA_RANGE_x_NTP_SERVER, 99
- DHCP_EXTRA_RANGE_x_START, 99
- DHCP_LEASES_DIR, 98
- DHCP_LEASES_VOLATILE, 98
- DHCP_LS_TIME_DYN, 98
- DHCP_LS_TIME_FIX, 98
- DHCP_MAX_LS_TIME_DYN, 98
- DHCP_MAX_LS_TIME_FIX, 98
- DHCP_RANGE_N, 98
- DHCP_RANGE_x_DNS_DOMAIN, 99
- DHCP_RANGE_x_DNS_SERVER1, 98
- DHCP_RANGE_x_DNS_SERVER2, 99
- DHCP_RANGE_x_END, 98
- DHCP_RANGE_x_GATEWAY, 99
- DHCP_RANGE_x_MTU, 99
- DHCP_RANGE_x_NET, 98
- DHCP_RANGE_x_NTP_SERVER, 99
- DHCP_RANGE_x_OPTION_N, 99
- DHCP_RANGE_x_OPTION_x, 99
- DHCP_RANGE_x_PXE_FILENAME, 100
- DHCP_RANGE_x_PXE_OPTIONS, 100
- DHCP_RANGE_x_PXE_SERVERIP, 100
- DHCP_RANGE_x_PXE_SERVERNAME, 100
- DHCP_RANGE_x_START, 98
- DHCP_TYPE, 97
- DHCP_VERBOSE, 98
- DHCP_WINSERVER_1, 98
- DHCP_WINSERVER_2, 98
- DHCPRELAY_IF_N, 100
- DHCPRELAY_IF_x, 100
- DHCPRELAY_SERVER, 100
- DIALMODE, 69
- DMZ_LOG, 376
- DMZ_NAT, 376
- DMZ_ORANGE_RED_N, 376
- DMZ_ORANGE_RED_x, 376
- DMZ_ORANGE_ROUTER_N, 376
- DMZ_ORANGE_ROUTER_x, 376
- DMZ_RED_DEV, 376
- DNS_AUTHORITATIVE, 95
- DNS_AUTHORITATIVE_IPADDR, 95
- DNS_AUTHORITATIVE_NS, 95
- DNS_BIND_INTERFACES, 92
- DNS_BOGUS_PRIV, 93
- DNS_FILTERWIN2K, 93
- DNS_FORBIDDEN_N, 93
- DNS_FORBIDDEN_x, 93
- DNS_FORWARD_LOCAL, 94
- DNS_FORWARDERS, 66
- DNS_LISTEN_N, 92
- DNS_LISTEN_x, 92
- DNS_LISTENIP_N, 377
- DNS_LISTENIP_x, 377
- DNS_LOCAL_HOST_CACHE_TTL, 94
- DNS_MX_SERVER, 92
- DNS_REBINDOK_N, 97
- DNS_REBINDOK_x_DOMAIN, 97
- DNS_REDIRECT_N, 93
- DNS_REDIRECT_x, 93
- DNS_REDIRECT_x_IP, 93
- DNS_SPECIAL_N, 377
- DNS_SPECIAL_x_DNSIP, 377
- DNS_SPECIAL_x_DOMAIN, 377
- DNS_SPECIAL_x_NETWORK, 377
- DNS_SUPPORT_IPV6, 94
- DNS_VERBOSE, 92
- DNS_ZONE_DELEGATION_N, 96
- DNS_ZONE_DELEGATION_x, 96
- DNS_ZONE_DELEGATION_x_DOMAIN, 96
- DNS_ZONE_DELEGATION_x_NETWORK, 96

- DNS_ZONE_DELEGATION_x -
UPSTREAM_SERVER_x, 96
- DNS_ZONE_DELEGATION_x -
UPSTREAM_SERVER_x_IP, 96
- DNS_ZONE_DELEGATION_x -
UPSTREAM_SERVER_x -
QUERYSOURCEIP, 96
- DNS_ZONE_NETWORK_N, 95
- DNS_ZONE_NETWORK_x, 95
- DOMAIN_NAME, 66
- DYNDNS_ALLOW_SSL, 116
- DYNDNS_DEBUG_PROVIDER, 116
- DYNDNS_LOGINTIME, 116
- DYNDNS_LOOKUP_NAMES, 116
- DYNDNS_N, 114
- DYNDNS_SAVE_OUTPUT, 114
- DYNDNS_x_CIRCUIT, 115
- DYNDNS_x_EXT_IP, 115
- DYNDNS_x_HOSTNAME, 115
- DYNDNS_x_LOGIN, 115
- DYNDNS_x_PASSWORD, 114
- DYNDNS_x_PROVIDER, 114
- DYNDNS_x_RENEW, 115
- DYNDNS_x_UPDATEHOST, 115
- DYNDNS_x_USER, 114

- EASYCRON_MAIL, 116
- EASYCRON_N, 116
- EASYCRON_x_COMMAND, 117
- EASYCRON_x_CUSTOM, 117
- EASYCRON_x_TIME, 117
- ETHTOOL_DEV_N, 84
- ETHTOOL_DEV_x, 85
- ETHTOOL_DEV_x_OPTION_N, 85
- ETHTOOL_DEV_x_OPTION_x -
NAME, 85
- ETHTOOL_DEV_x_OPTION_x -
VALUE, 85
- Exemple de fichier (base.txt), 17
- EXTMOUNT_N, 121
- EXTMOUNT_x_FILESYSTEM, 121
- EXTMOUNT_x_MOUNTPOINT, 121
- EXTMOUNT_x_OPTIONS, 121
- EXTMOUNT_x_VOLUMEID, 121

- FILESONLY, 263

- FLI4L_UUID, 26
- FORWARD_DENY_PORT_N, 376
- FORWARD_DENY_PORT_x, 376
- FORWARD_HOST_N, 376
- FORWARD_HOST_WHITE, 376
- FORWARD_HOST_x, 376
- FRITZDSL_CHARGEINT, 104
- FRITZDSL_DEBUG, 103
- FRITZDSL_FILTER, 105
- FRITZDSL_HUP_TIMEOUT, 104
- FRITZDSL_MRU, 106
- FRITZDSL_MTU, 106
- FRITZDSL_NAME, 103
- FRITZDSL_NF_MSS, 106
- FRITZDSL_PASS, 104
- FRITZDSL_PROVIDER, 108
- FRITZDSL_TIMES, 104
- FRITZDSL_TYPE, 108
- FRITZDSL_USEPEERDNS, 103
- FRITZDSL_USER, 104
- ftp, 65
- FTP_PF_ENABLE_ACTIVE, 227

- h323, 65
- HDDRV_N, 122
- HDDRV_x, 122
- HDDRV_x_OPTION, 122
- HDIT_DATA, 378
- HDIT_POWEROFF, 378
- HDIT_SIZES, 378
- HDSLEEP_TIMEOUT, 121
- HOST_EXTRA_N, 92
- HOST_EXTRA_x_IP4, 92
- HOST_EXTRA_x_IP6, 92
- HOST_EXTRA_x_NAME, 92
- HOST_N, 90
- HOST_x_ALIAS_N, 90
- HOST_x_ALIAS_x, 90
- HOST_x_DHCPTYP, 90
- HOST_x_DOMAIN, 90
- HOST_x_IP4, 90
- HOST_x_IP6, 90
- HOST_x_MAC, 90
- HOST_x_MAC2, 90
- HOST_x_NAME, 90
- HOST_x_PXE_FILENAME, 100
- HOST_x_PXE_OPTIONS, 100

HOST_x_PXE_SERVERIP, 100
 HOST_x_PXE_SERVERNAME, 100
 HOSTNAME, 23
 HOSTNAME_ALIAS_N, 67
 HOSTNAME_ALIAS_x, 67
 HOSTNAME_IP, 66
 HOSTNAME_IP6, 134
 HTTPD_ARPING, 124
 HTTPD_ARPING_IGNORE_N, 124
 HTTPD_ARPING_IGNORE_x, 124
 HTTPD_GUI_LANG, 123
 HTTPD_GUI_SKIN, 123
 HTTPD_LISTENIP, 123
 HTTPD_PORT, 124
 HTTPD_PORTFW, 124
 HTTPD_USER, 358
 HTTPD_USER_N, 124
 HTTPD_USER_x_PASSWORD, 124
 HTTPD_USER_x_RIGHTS, 124
 HTTPD_USER_x_USERNAME, 124
 HW_DETECT_AT_BOOTTIME, 230
 HWSUPP_BOOT_LED, 130
 HWSUPP_BUTTON_N, 130
 HWSUPP_BUTTON_x, 130
 HWSUPP_BUTTON_x_DEVICE, 130
 HWSUPP_BUTTON_x_PARAM, 131
 HWSUPP_CPUFREQ, 128
 HWSUPP_CPUFREQ_GOVERNOR, 128
 HWSUPP_DRIVER_N, 131
 HWSUPP_DRIVER_x, 131
 HWSUPP_I2C_N, 131
 HWSUPP_I2C_x_ADDRESS, 132
 HWSUPP_I2C_x_BUS, 131
 HWSUPP_I2C_x_DEVICE, 132
 HWSUPP_LED_N, 128
 HWSUPP_LED_x, 128
 HWSUPP_LED_x_DEVICE, 129
 HWSUPP_LED_x_PARAM, 129
 HWSUPP_TYPE, 127
 HWSUPP_WATCHDOG, 128
 IGMPProxy_ALT_N, 199
 IGMPProxy_ALT_NET_x, 199
 IGMPProxy_DEBUG, 199
 IGMPProxy_DEBUG2, 199
 IGMPProxy_DOWNLOAD_DEV, 199
 IGMPProxy_QUICKLEAVE_ON, 199
 IGMPProxy_UPLOAD_DEV, 199
 IGMPProxy_WHLIST_NET_x, 200
 IGMPProxy_WLIST_N, 200
 IMOND_ADMIN_PASS, 67
 IMOND_BEEP, 68
 IMOND_DIAL, 68
 IMOND_ENABLE, 68
 IMOND_LED, 67
 IMOND_LOG, 68
 IMOND_LOGDIR, 68
 IMOND_PASS, 67
 IMOND_PORT, 67
 IMOND_REBOOT, 68
 IMOND_ROUTE, 68
 IMOND_USE_ORIG, 376
 INPUT_ACCEPT_PORT_N, 376
 INPUT_ACCEPT_PORT_x, 376
 INPUT_POLICY, 376
 IP_CONNTRACK_MAX, 26
 IP_DYN_ADDR, 69
 IP_NET_N, 38
 IP_NET_x, 38
 IP_NET_x_COMMENT, 39
 IP_NET_x_DEV, 39
 IP_NET_x_MAC, 39
 IP_NET_x_NAME, 39
 IP_NET_x_TYPE, 39, 376
 IP_ROUTE_N, 40
 IP_ROUTE_x, 40
 IPV6_NET_N, 134
 IPV6_NET_x, 134
 IPV6_NET_x_ADVERTISE, 135
 IPV6_NET_x_ADVERTISE_DNS, 135
 IPV6_NET_x_DEV, 135
 IPV6_NET_x_DHCP, 135
 IPV6_NET_x_NAME, 136
 IPV6_NET_x_TUNNEL, 135
 IPV6_ROUTE_N, 138
 IPV6_ROUTE_x, 139
 IPV6_TUNNEL_N, 136
 IPV6_TUNNEL_x_DEFAULT, 136
 IPV6_TUNNEL_x_DEV, 138
 IPV6_TUNNEL_x_LOCALV4, 137
 IPV6_TUNNEL_x_LOCALV6, 137
 IPV6_TUNNEL_x_MTU, 138
 IPV6_TUNNEL_x_PASSWORD, 138

- IPV6_TUNNEL_x_PREFIX, 136
- IPV6_TUNNEL_x_REMOTEV4, 137
- IPV6_TUNNEL_x_REMOTEV6, 137
- IPV6_TUNNEL_x_TIMEOUT, 138
- IPV6_TUNNEL_x_TUNNELID, 138
- IPV6_TUNNEL_x_TYPE, 136
- IPV6_TUNNEL_x_USERID, 138
- irc, 65
- ISDN_CIRC_N, 150
- ISDN_CIRC_x_AUTH, 156
- ISDN_CIRC_x_BANDWIDTH, 151
- ISDN_CIRC_x_BUNDLING, 151
- ISDN_CIRC_x_CALLBACK, 155
- ISDN_CIRC_x_CBDELAY, 155
- ISDN_CIRC_x_CBNUMBER, 155
- ISDN_CIRC_x_CHARGEINT, 156
- ISDN_CIRC_x_CLAMP_MSS, 153
- ISDN_CIRC_x_DEBUG, 156
- ISDN_CIRC_x_DIALIN, 154
- ISDN_CIRC_x_DIALOUT, 154
- ISDN_CIRC_x_EAZ, 156
- ISDN_CIRC_x_FRAMECOMP, 153
- ISDN_CIRC_x_HEADERCOMP, 153
- ISDN_CIRC_x_HUP_TIMEOUT, 156
- ISDN_CIRC_x_LOCAL, 152
- ISDN_CIRC_x_MRU, 153
- ISDN_CIRC_x_MTU, 153
- ISDN_CIRC_x_NAME, 150
- ISDN_CIRC_x_PASS, 154
- ISDN_CIRC_x_REMOTE, 152
- ISDN_CIRC_x_REMOTENAME, 154
- ISDN_CIRC_x_ROUTE_N, 154
- ISDN_CIRC_x_ROUTE_X, 154
- ISDN_CIRC_x_SLAVE_EAZ, 156
- ISDN_CIRC_x_TIMES, 157
- ISDN_CIRC_x_TYPE, 151
- ISDN_CIRC_x_USEPEERDNS, 151
- ISDN_CIRC_x_USER, 154
- ISDN_DEBUG_LEVEL, 148
- ISDN_FILTER, 149
- ISDN_IO, 145
- ISDN_IO0, 145
- ISDN_IO1, 145
- ISDN_IP, 145
- ISDN_LZS_COMP, 150
- ISDN_LZS_DEBUG, 149
- ISDN_LZS_TWEAK, 150
- ISDN_MEM, 145
- ISDN_PORT, 145
- ISDN_TYPE, 145
- ISDN_VERBOSE_LEVEL, 149
- KERNEL_BOOT_OPTION, 25
- KERNEL_VERSION, 25
- KEYBOARD_LOCALE, 30
- LIBATA_DMA, 24
- LOCALE, 27
- LOG_BOOT_SEQ, 323
- LOGIP_LOGDIR, 72
- MASQ_NETWORK, 376
- Masquerading, 63
- MKFLI4L_DEBUG_OPTION, 264
- MOUNT_BOOT, 24
- NET_DRV_N, 30
- NET_DRV_x, 31
- NET_DRV_x_OPTION, 31
- OAC_ALL_INVISIBLE, 126
- OAC_BLOCK_UNKNOWN_IF_x, 126
- OAC_GROUP_N, 126
- OAC_GROUP_x_BOOTBLOCK, 126
- OAC_GROUP_x_CLIENT_N, 126
- OAC_GROUP_x_CLIENT_x, 126
- OAC_GROUP_x_INVISIBLE, 126
- OAC_GROUP_x_NAME, 126
- OAC_INPUT, 126
- OAC_LIMITS, 126
- OAC_MODE, 126
- OAC_WANDEVICE, 125
- OPENVPN_DEFAULT_ALLOW_-
ICMPPING, 171
- OPENVPN_DEFAULT_CIPHER, 170
- OPENVPN_DEFAULT_COMPRESS,
170
- OPENVPN_DEFAULT_CREATE_-
SECRET, 170
- OPENVPN_DEFAULT_DIGEST, 170
- OPENVPN_DEFAULT_FLOAT, 170
- OPENVPN_DEFAULT_FRAGMENT,
173
- OPENVPN_DEFAULT_KEYSIZE, 170

- OPENVPN_DEFAULT_LINK_MTU, [173](#)
- OPENVPN_DEFAULT_-
MANAGEMENT_LOG_-
CACHE, [173](#)
- OPENVPN_DEFAULT_MSSFIX, [173](#)
- OPENVPN_DEFAULT_MUTE_-
REPLAY_WARNINGS, [173](#)
- OPENVPN_DEFAULT_OPEN_-
OVPNPORT, [171](#)
- OPENVPN_DEFAULT_PF_DMZ_-
TYPE, [379](#)
- OPENVPN_DEFAULT_PF_-
FORWARD_LOG, [171](#)
- OPENVPN_DEFAULT_PF_-
FORWARD_POLICY, [171](#)
- OPENVPN_DEFAULT_PF_INPUT_-
LOG, [171](#)
- OPENVPN_DEFAULT_PF_INPUT_-
POLICY, [171](#)
- OPENVPN_DEFAULT_PING, [171](#)
- OPENVPN_DEFAULT_PING_-
RESTART, [172](#)
- OPENVPN_DEFAULT_PROTOCOL, [172](#)
- OPENVPN_DEFAULT_RENEG_SEC, [171](#)
- OPENVPN_DEFAULT_RESOLV_-
RETRY, [172](#)
- OPENVPN_DEFAULT_RESTART, [172](#)
- OPENVPN_DEFAULT_SHAPER, [174](#)
- OPENVPN_DEFAULT_START, [172](#)
- OPENVPN_DEFAULT_TUN_MTU, [173](#)
- OPENVPN_DEFAULT_TUN_MTU_-
EXTRA, [173](#)
- OPENVPN_DEFAULT_VERBOSE, [173](#)
- OPENVPN_EXPERT, [174](#)
- OPENVPN_N, [164](#)
- OPENVPN_VERSION, [379](#)
- OPENVPN_WEBGUI, [178](#)
- OPENVPN_x_ACTIV, [174](#)
- OPENVPN_x_ALLOW_ICMPPING, [176](#)
- OPENVPN_x_BRIDGE, [166](#)
- OPENVPN_x_BRIDGE_COST, [167](#)
- OPENVPN_x_BRIDGE_PRIORITY, [167](#)
- OPENVPN_x_CHECK_CONFIG, [174](#)
- OPENVPN_x_CIPHER, [175](#)
- OPENVPN_x_COMPRESS, [175](#)
- OPENVPN_x_CREATE_SECRET, [175](#)
- OPENVPN_x_DIGEST, [175](#)
- OPENVPN_x_DNSIP, [169](#)
- OPENVPN_x_DOMAIN, [169](#)
- OPENVPN_x_FLOAT, [175](#)
- OPENVPN_x_FRAGMENT, [178](#)
- OPENVPN_x_IPV6, [167](#)
- OPENVPN_x_ISDN_CIRC_NAME, [175](#)
- OPENVPN_x_KEYSIZE, [175](#)
- OPENVPN_x_LINK_MTU, [178](#)
- OPENVPN_x_LOCAL_HOST, [164](#)
- OPENVPN_x_LOCAL_PORT, [164](#)
- OPENVPN_x_LOCAL_VPN_IP, [167](#)
- OPENVPN_x_LOCAL_VPN_IPV6, [168](#)
- OPENVPN_x_MANAGEMENT_LOG_-
CACHE, [176](#)
- OPENVPN_x_MSSFIX, [177](#)
- OPENVPN_x_MUTE_REPLAY_-
WARNINGS, [176](#)
- OPENVPN_x_NAME, [174](#)
- OPENVPN_x_OPEN_OVPNPORT, [176](#)
- OPENVPN_x_PF6_FORWARD_N, [177](#)
- OPENVPN_x_PF6_FORWARD_x, [177](#)
- OPENVPN_x_PF6_INPUT_N, [177](#)
- OPENVPN_x_PF6_INPUT_x, [177](#)
- OPENVPN_x_PF_DMZ_TYPE, [379](#)
- OPENVPN_x_PF_FORWARD_LOG, [177](#)
- OPENVPN_x_PF_FORWARD_N, [177](#)
- OPENVPN_x_PF_FORWARD_-
POLICY, [177](#)
- OPENVPN_x_PF_FORWARD_x, [177](#)
- OPENVPN_x_PF_INPUT_LOG, [176](#)
- OPENVPN_x_PF_INPUT_N, [176](#)
- OPENVPN_x_PF_INPUT_POLICY, [176](#)
- OPENVPN_x_PF_INPUT_x, [176](#)
- OPENVPN_x_PF_POSTROUTING_N, [177](#)
- OPENVPN_x_PF_POSTROUTING_x, [177](#)
- OPENVPN_x_PF_PREROUTING_N, [177](#)

OPENVPN_x_PF_PREROUTING_x,
 177
 OPENVPN_x_PING, 175
 OPENVPN_x_PING_RESTART, 175
 OPENVPN_x_PROTOCOL, 175
 OPENVPN_x_REMOTE_HOST, 164
 OPENVPN_x_REMOTE_HOST_N, 164
 OPENVPN_x_REMOTE_HOST_x, 164
 OPENVPN_x_REMOTE_PORT, 164
 OPENVPN_x_REMOTE_VPN_IP, 167
 OPENVPN_x_REMOTE_VPN_IPV6,
 168
 OPENVPN_x_RESOLV_RETRY, 175
 OPENVPN_x_RESTART, 176
 OPENVPN_x_ROUTE_N, 168
 OPENVPN_x_ROUTE_x, 168
 OPENVPN_x_ROUTE_x_DNSIP, 169
 OPENVPN_x_ROUTE_x_DOMAIN,
 169
 OPENVPN_x_SECRET, 165
 OPENVPN_x_SHAPER, 178
 OPENVPN_x_START, 175
 OPENVPN_x_TUN_MTU, 178
 OPENVPN_x_TUN_MTU_EXTRA, 178
 OPENVPN_x_TYPE, 166
 OPENVPN_x_VERBOSE, 176
 OPT_ARP, 381
 OPT_ATH_INFO, 230
 OPT_BCRELAY, 75, 381
 OPT_BONDING_DEV, 75
 OPT_BRIDGE_DEV, 81
 OPT_CHRONY, 88
 OPT_DHCP, 97
 OPT_DHCP_CLIENT, 89
 OPT_DHCPRELAY, 100
 OPT_DIG, 227
 OPT_DMZ, 377
 OPT_DNS, 92
 OPT_DYNDNS, 114
 OPT_E3, 230
 OPT_EASYCRON, 116
 OPT_EBTABLES, 84
 OPT_ETHTOOL, 84, 381
 OPT_EVSS, 377
 OPT_EXTMOUNT, 121
 OPT_FRITZDSL, 108
 OPT_FTP, 227
 OPT_HDDRV, 122
 OPT_HDINSTALL, 118
 OPT_HDINSTALL_TEST, 378
 OPT_HDSLEEP, 121
 OPT_HOSTS, 90
 OPT_HTTPD, 123
 OPT_HW_DETECT, 230
 OPT_HWSUPP, 127
 OPT_I2CTOOLS, 230
 OPT_IFTOP, 227
 OPT_IGMPPROXY, 194, 199
 OPT_IMONC, 227
 OPT_IPERF, 227
 OPT_IPV6, 134
 OPT_ISDN, 144
 OPT_ISDN_COMP, 149
 OPT_IWLEEPROM, 230
 OPT_KLOGD, 72
 OPT_LOGIP, 72
 OPT_LSPCI, 230
 OPT_MAKEKBL, 30
 OPT_MOUNT, 120
 OPT_MOUNTFLOPPY, 377
 OPT_MTOOLS, 231
 OPT_NETCAT, 227
 OPT_NETSTAT, 381
 OPT_NGREP, 227
 OPT_NTTCP, 228
 OPT_OAC, 125
 OPT_OPENSSL, 231
 OPT_OPENVPN, 164
 OPT_PCMCIA, 186
 OPT_PFC, 378
 OPT_PLINK_CLIENT, 226
 OPT_PNP, 73
 OPT_POESTATUS, 110
 OPT_PPP, 186
 OPT_PPPOE, 107
 OPT_PPPOE_CIRC, 108, 378
 OPT_PPTP, 109
 OPT_PRIVOX, 188
 OPT_QOS, 207
 OPT_RCAPID, 161
 OPT_REAVER, 231
 OPT_RECOVER, 122
 OPT_RTMON, 228
 OPT_SCP, 380

- OPT_SERIAL, [381](#)
- OPT_SFTPSERVER, [226](#)
- OPT_SHRED, [231](#)
- OPT_SIPPROXY, [194](#)
- OPT_SOCAT, [228](#)
- OPT_SS5, [192](#)
- OPT_SSH_CLIENT, [225](#)
- OPT_SSHD, [221](#)
- OPT_STRACE, [231](#)
- OPT_STUNNEL, [201](#)
- OPT_SYSLOGD, [70](#)
- OPT_TCPDUMP, [229](#)
- OPT_TELMOND, [158](#)
- OPT_TFTP, [101](#)
- OPT_TOR, [191](#)
- OPT_TRACEROUTE, [381](#)
- OPT_TRACEROUTE6, [381](#)
- OPT_TRANSPROXY, [193](#)
- OPT_UMTS, [232](#)
- OPT_USB, [234](#)
- OPT_VALGRIND, [231](#)
- OPT_VLAN_DEV, [79](#)
- OPT_VPN_CARD, [132](#)
- OPT_WGET, [229](#)
- OPT_WLAN, [236](#)
- OPT_Y2K, [72](#)
- OPT_YADIFA, [101](#)
- OPT_YADIFA_SLAVE_ZONE_-
USE_DNSMASQ_ZONE_-
DELEGATION, [102](#)
- OPT_YADIFA_USE_DNSMASQ_-
ZONE_DELEGATION, [101](#)
- OPT_YTREE, [231](#)

- PACKETFILTER_LOG, [377](#)
- PACKETFILTER_LOG_LEVEL, [377](#)
- PASSWORD, [23](#)
- PCMCIA_CARDMGR_OPTS, [380](#)
- PCMCIA_CORE_OPTS, [380](#)
- PCMCIA_MISC_N, [186](#)
- PCMCIA_MISC_x, [186](#)
- PCMCIA_PCIC, [186](#)
- PCMCIA_PCIC_EXTERN, [380](#)
- PCMCIA_PCIC_OPTS, [186](#)
- PF6_FORWARD_ACCEPT_DEF, [141](#)
- PF6_FORWARD_LOG, [141](#)
- PF6_FORWARD_LOG_LIMIT, [141](#)

- PF6_FORWARD_N, [141](#)
- PF6_FORWARD_POLICY, [141](#)
- PF6_FORWARD_REJ_LIMIT, [141](#)
- PF6_FORWARD_UDP_REJ_LIMIT,
[141](#)
- PF6_FORWARD_x, [141](#)
- PF6_FORWARD_x_COMMENT, [142](#)
- PF6_INPUT_ACCEPT_DEF, [139](#)
- PF6_INPUT_ICMP_ECHO_REQ_-
LIMIT, [140](#)
- PF6_INPUT_ICMP_ECHO_REQ_-
SIZE, [140](#)
- PF6_INPUT_LOG, [139](#)
- PF6_INPUT_LOG_LIMIT, [140](#)
- PF6_INPUT_N, [140](#)
- PF6_INPUT_POLICY, [139](#)
- PF6_INPUT_REJ_LIMIT, [140](#)
- PF6_INPUT_UDP_REJ_LIMIT, [140](#)
- PF6_INPUT_x, [140](#)
- PF6_INPUT_x_COMMENT, [141](#)
- PF6_LOG_LEVEL, [139](#)
- PF6_OUTPUT_ACCEPT_DEF, [142](#)
- PF6_OUTPUT_LOG, [142](#)
- PF6_OUTPUT_LOG_LIMIT, [142](#)
- PF6_OUTPUT_N, [143](#)
- PF6_OUTPUT_POLICY, [142](#)
- PF6_OUTPUT_REJ_LIMIT, [142](#)
- PF6_OUTPUT_UDP_REJ_LIMIT, [142](#)
- PF6_OUTPUT_x, [143](#)
- PF6_OUTPUT_x_COMMENT, [143](#)
- PF6_POSTROUTING_N, [144](#)
- PF6_POSTROUTING_x, [144](#)
- PF6_POSTROUTING_x_COMMENT,
[144](#)
- PF6_PREROUTING_N, [144](#)
- PF6_PREROUTING_x, [144](#)
- PF6_PREROUTING_x_COMMENT,
[144](#)
- PF6_USR_CHAIN_N, [143](#)
- PF6_USR_CHAIN_x_NAME, [143](#)
- PF6_USR_CHAIN_x_RULE_N, [143](#)
- PF6_USR_CHAIN_x_RULE_x, [143](#)
- PF6_USR_CHAIN_x_RULE_x_-
COMMENT, [144](#)
- PF_FORWARD_ACCEPT_DEF, [52](#)
- PF_FORWARD_LOG, [52](#)
- PF_FORWARD_LOG_LIMIT, [52](#)

- PF_FORWARD_N, [52](#)
- PF_FORWARD_POLICY, [52](#)
- PF_FORWARD_REJ_LIMIT, [52](#)
- PF_FORWARD_UDP_REJ_LIMIT, [52](#)
- PF_FORWARD_x, [52](#)
- PF_FORWARD_x_COMMENT, [52](#)
- PF_INPUT_ACCEPT_DEF, [50](#)
- PF_INPUT_ICMP_ECHO_REQ_-
LIMIT, [51](#)
- PF_INPUT_ICMP_ECHO_REQ_SIZE,
[51](#)
- PF_INPUT_LOG, [51](#)
- PF_INPUT_LOG_LIMIT, [51](#)
- PF_INPUT_N, [51](#)
- PF_INPUT_POLICY, [50](#)
- PF_INPUT_REJ_LIMIT, [51](#)
- PF_INPUT_UDP_REJ_LIMIT, [51](#)
- PF_INPUT_x, [51](#)
- PF_INPUT_x_COMMENT, [51](#)
- PF_LOG_LEVEL, [50](#)
- PF_NEW_CONFIG, [40](#), [377](#)
- PF_OUTPUT_ACCEPT_DEF, [53](#)
- PF_OUTPUT_CT_ACCEPT_DEF, [65](#)
- PF_OUTPUT_CT_N, [65](#)
- PF_OUTPUT_CT_x, [65](#)
- PF_OUTPUT_CT_x_COMMENT, [65](#)
- PF_OUTPUT_LOG, [53](#)
- PF_OUTPUT_LOG_LIMIT, [53](#)
- PF_OUTPUT_N, [53](#)
- PF_OUTPUT_POLICY, [53](#)
- PF_OUTPUT_REJ_LIMIT, [53](#)
- PF_OUTPUT_UDP_REJ_LIMIT, [53](#)
- PF_OUTPUT_x, [53](#)
- PF_OUTPUT_x_COMMENT, [53](#)
- PF_POSTROUTING_N, [55](#)
- PF_POSTROUTING_x, [55](#)
- PF_POSTROUTING_x_COMMENT, [55](#)
- PF_PREROUTING_CT_ACCEPT_-
DEF, [65](#)
- PF_PREROUTING_CT_N, [65](#)
- PF_PREROUTING_CT_x, [65](#)
- PF_PREROUTING_CT_x_-
COMMENT, [65](#)
- PF_PREROUTING_N, [55](#)
- PF_PREROUTING_x, [55](#)
- PF_PREROUTING_x_COMMENT, [55](#)
- PF_USR_CHAIN_N, [54](#)
- PF_USR_CHAIN_x_NAME, [54](#)
- PF_USR_CHAIN_x_RULE_N, [54](#)
- PF_USR_CHAIN_x_RULE_x, [54](#)
- PF_USR_CHAIN_x_RULE_x_-
COMMENT, [54](#)
- POWERMANAGEMENT, [25](#)
- PPP_DEV, [187](#)
- PPP_IPADDR, [187](#)
- PPP_NETMASK, [187](#)
- PPP_NETWORK, [187](#)
- PPP_PEER, [187](#)
- PPP_SPEED, [187](#)
- PPPOE_CHARGEINT, [104](#)
- PPPOE_CIRC_N, [108](#), [378](#)
- PPPOE_CIRC_x_CHARGEINT, [108](#),
[378](#)
- PPPOE_CIRC_x_DEBUG, [108](#), [378](#)
- PPPOE_CIRC_x_ETH, [108](#), [378](#)
- PPPOE_CIRC_x_FILTER, [108](#), [378](#)
- PPPOE_CIRC_x_HUP_TIMEOUT,
[108](#), [378](#)
- PPPOE_CIRC_x_MRU, [108](#), [378](#)
- PPPOE_CIRC_x_MTU, [108](#), [378](#)
- PPPOE_CIRC_x_NAME, [108](#), [378](#)
- PPPOE_CIRC_x_PASS, [108](#), [378](#)
- PPPOE_CIRC_x_TIMES, [108](#), [378](#)
- PPPOE_CIRC_x_TYPE, [108](#), [378](#)
- PPPOE_CIRC_x_USEPEERDNS, [108](#),
[378](#)
- PPPOE_CIRC_x_USER, [108](#), [378](#)
- PPPOE_DEBUG, [103](#)
- PPPOE_ETH, [107](#)
- PPPOE_FILTER, [105](#)
- PPPOE_HUP_TIMEOUT, [104](#), [108](#)
- PPPOE_MRU, [106](#)
- PPPOE_MTU, [106](#)
- PPPOE_NAME, [103](#)
- PPPOE_NF_MSS, [106](#)
- PPPOE_PASS, [104](#)
- PPPOE_TIMES, [104](#)
- PPPOE_TYPE, [107](#)
- PPPOE_USEPEERDNS, [103](#)
- PPPOE_USER, [104](#)
- pptp, [65](#)
- PPTP_CHARGEINT, [104](#)
- PPTP_CLIENT_LOGLEVEL, [110](#)
- PPTP_CLIENT_REORDER_TO, [110](#)

- PPTP_DEBUG, [103](#)
- PPTP_ETH, [109](#)
- PPTP_FILTER, [105](#)
- PPTP_HUP_TIMEOUT, [104](#)
- PPTP_MODEM_TYPE, [109](#)
- PPTP_NAME, [103](#)
- PPTP_PASS, [104](#)
- PPTP_TIMES, [104](#)
- PPTP_USEPEERDNS, [103](#)
- PPTP_USER, [104](#)
- PRESERVE, [377](#)
- PRIVOXY_MENU, [189](#)
- PRIVOXY_N, [189](#)
- PRIVOXY_x_ACTIONDIR, [189](#)
- PRIVOXY_x_ALLOW_N, [189](#)
- PRIVOXY_x_ALLOW_x, [189](#)
- PRIVOXY_x_CONFIG, [190](#)
- PRIVOXY_x_HTTP_PROXY, [190](#)
- PRIVOXY_x_LISTEN, [189](#)
- PRIVOXY_x_LOGDIR, [190](#)
- PRIVOXY_x_LOGLEVEL, [190](#)
- PRIVOXY_x SOCKS_PROXY, [190](#)
- PRIVOXY_x_TOGGLE, [190](#)
- PXESUBDIR, [263](#)

- QOS_CLASS_N, [208](#)
- QOS_CLASS_x_DIRECTION, [209](#)
- QOS_CLASS_x_LABEL, [210](#)
- QOS_CLASS_x_MAXBANDWIDTH, [209](#)
- QOS_CLASS_x_MINBANDWIDTH, [208](#)
- QOS_CLASS_x_PARENT, [208](#)
- QOS_CLASS_x_PRIO, [209](#)
- QOS_FILTER_N, [210](#)
- QOS_FILTER_x_CLASS, [211](#)
- QOS_FILTER_x_IP_EXTERN, [211](#)
- QOS_FILTER_x_IP_INTERN, [211](#)
- QOS_FILTER_x_OPTION, [212](#)
- QOS_FILTER_x_PORT, [212](#)
- QOS_FILTER_x_PORT_TYPE, [212](#)
- QOS_INTERNET_BAND_DOWN, [207](#)
- QOS_INTERNET_BAND_UP, [207](#)
- QOS_INTERNET_DEFAULT_DOWN, [208](#)
- QOS_INTERNET_DEFAULT_UP, [208](#)
- QOS_INTERNET_DEV_N, [207](#)
- QOS_INTERNET_DEV_x, [207](#)

- RCAPID_PORT, [161](#)
- REMOTEHOSTNAME, [263](#)
- REMOTEPATHNAME, [263](#)
- REMOTEPORT, [263](#)
- REMOTEREMOUNT, [263](#)
- REMOTEUPDATE, [263](#)
- REMOTEUSERNAME, [263](#)
- ROUTE_NETWORK, [377](#)

- sane, [65](#)
- SER_CONSOLE, [27](#)
- SER_CONSOLE_IF, [28](#)
- SER_CONSOLE_RATE, [28](#)
- sip, [65](#)
- snmp, [65](#)
- SQUEEZE_SCRIPTS, [264](#)
- SS5_ALLOW_N, [193](#)
- SS5_ALLOW_x, [193](#)
- SS5_LISTEN_N, [192](#)
- SS5_LISTEN_x, [193](#)
- SSH_CLIENT_PRIVATE_KEYFILE_N, [225](#)
- SSH_CLIENT_PRIVATE_KEYFILE_x, [225](#)
- SSHD_ALLOWPASSWORDLOGIN, [221](#)
- SSHD_CREATEHOSTKEYS, [222](#)
- SSHD_PORT, [224](#)
- SSHD_PUBLIC_KEY_N, [224](#)
- SSHD_PUBLIC_KEY_x, [225](#)
- SSHD_PUBLIC_KEYFILE_N, [225](#)
- SSHD_PUBLIC_KEYFILE_x, [225](#)
- SSHKEYFILE, [263](#)
- START_IMOND, [67](#)
- STUNNEL_DEBUG, [201](#)
- STUNNEL_N, [201](#)
- STUNNEL_x_ACCEPT, [202](#)
- STUNNEL_x_ACCEPT_IPV4, [203](#)
- STUNNEL_x_ACCEPT_IPV6, [203](#)
- STUNNEL_x_CERT_CA_FILE, [204](#)
- STUNNEL_x_CERT_FILE, [204](#)
- STUNNEL_x_CERT_VERIFY, [204](#)
- STUNNEL_x_CLIENT, [202](#)
- STUNNEL_x_CONNECT, [203](#)
- STUNNEL_x_DELAY_DNS, [203](#)
- STUNNEL_x_NAME, [202](#)
- STUNNEL_x_OUTGOING_IP, [203](#)
- SYSLOGD_DEST_N, [70](#)

- SYSLOGD_DEST_x, [70](#)
- SYSLOGD_RECEIVER, [70](#)
- SYSLOGD_ROTATE, [71](#)
- SYSLOGD_ROTATE_AT_-
SHUTDOWN, [72](#)
- SYSLOGD_ROTATE_DIR, [71](#)
- SYSLOGD_ROTATE_MAX, [71](#)
- TELMOND_CAPI_CTRL_N, [160](#)
- TELMOND_CAPI_CTRL_x, [160](#)
- TELMOND_CMD_N, [159](#)
- TELMOND_CMD_x, [159](#)
- TELMOND_LOG, [158](#)
- TELMOND_LOGDIR, [159](#)
- TELMOND_MSN_N, [159](#)
- TELMOND_MSN_x, [159](#)
- TELMOND_PORT, [158](#)
- tftp, [65](#)
- TFTP_PATH, [101](#)
- TFTPBOOTIMAGE, [263](#)
- TFTPBOOTPATH, [263](#)
- TIME_INFO, [25](#)
- TOR_ALLOW_N, [192](#)
- TOR_ALLOW_x, [192](#)
- TOR_CONTROL_PASSWORD, [192](#)
- TOR_CONTROL_PORT, [192](#)
- TOR_DATA_DIR, [192](#)
- TOR_HTTP_PROXY, [192](#)
- TOR_HTTP_PROXY_AUTH, [192](#)
- TOR_HTTPS_PROXY, [192](#)
- TOR_HTTPS_PROXY_AUTH, [192](#)
- TOR_LISTEN_N, [191](#)
- TOR_LISTEN_x, [191](#)
- TOR_LOGFILE, [192](#)
- TOR_LOGLEVEL, [192](#)
- TRANSPROXY_ALLOW_N, [193](#)
- TRANSPROXY_ALLOW_x, [194](#)
- TRANSPROXY_LISTEN_N, [193](#)
- TRANSPROXY_LISTEN_x, [193](#)
- TRANSPROXY_TARGET_IP, [193](#)
- TRANSPROXY_TARGET_PORT, [193](#)
- TRUSTED_NETS, [377](#)
- UMTS_ADAPTER, [233](#)
- UMTS_APN, [232](#)
- UMTS_CHARGEINT, [233](#)
- UMTS_CTRL, [234](#)
- UMTS_DEBUG, [232](#)
- UMTS_DEV, [234](#)
- UMTS_DIALOUT, [232](#)
- UMTS_DRV, [233](#)
- UMTS_FILTER, [233](#)
- UMTS_GPRS_UMTS, [232](#)
- UMTS_HUP_TIMEOUT, [232](#)
- UMTS_IDDEVICE, [233](#)
- UMTS_IDDEVICE2, [233](#)
- UMTS_IDVENDOR, [233](#)
- UMTS_IDVENDOR2, [233](#)
- UMTS_NAME, [232](#)
- UMTS_PASSWD, [232](#)
- UMTS_PIN, [232](#)
- UMTS_SWITCH, [233](#)
- UMTS_TIMES, [232](#)
- UMTS_USEPEERDNS, [233](#)
- UMTS_USER, [232](#)
- USB_EXTRA_DRIVER_N, [234](#)
- USB_EXTRA_DRIVER_x, [234](#)
- USB_EXTRA_DRIVER_x_PARAM,
[235](#)
- USB_LOWLEVEL, [381](#)
- USB_MODEM_WAITSECONDS, [235](#)
- VERBOSE, [263](#)
- VLAN_DEV_N, [79](#)
- VLAN_DEV_x_DEV, [79](#)
- VLAN_DEV_x_VID, [80](#)
- VPN_CARD_TYPE, [132](#)
- WGET_SSL, [381](#)
- WLAN_N, [237](#)
- WLAN_REGDOMAIN, [237](#)
- WLAN_WEBGUI, [236](#)
- WLAN_x_ACL_MAC_N, [240](#)
- WLAN_x_ACL_MAC_x, [240](#)
- WLAN_x_ACL_POLICY, [240](#)
- WLAN_x_AP, [240](#)
- WLAN_x_BRIDGE, [241](#)
- WLAN_x_CHANNEL, [237](#)
- WLAN_x_DIVERSITY, [240](#)
- WLAN_x_DIVERSITY_RX, [240](#)
- WLAN_x_DIVERSITY_TX, [240](#)
- WLAN_x_ENC_ACTIVE, [239](#)
- WLAN_x_ENC_MODE, [239](#)
- WLAN_x_ENC_N, [238](#)

WLAN_x_ENC_x, [238](#)
WLAN_x_ESSID, [237](#)
WLAN_x_MAC, [237](#)
WLAN_x_MAC_OVERRIDE, [237](#)
WLAN_x_MODE, [237](#)
WLAN_x_NOESSID, [237](#)
WLAN_x_PSKFILE, [240](#)
WLAN_x_RATE, [238](#)
WLAN_x_RTS, [238](#)
WLAN_x_WPA_DEBUG, [240](#)
WLAN_x_WPA_ENCRYPTION, [239](#)
WLAN_x_WPA_KEY_MGMT, [239](#)
WLAN_x_WPA_PSK, [239](#)
WLAN_x_WPA_TYPE, [239](#)
WLAN_x_WPS, [240](#)

Y2K_DAYS, [72](#)
YADIFA_ALLOW_QUERY_N, [102](#)
YADIFA_ALLOW_QUERY_x, [102](#)
YADIFA_LISTEN_N, [102](#)
YADIFA_SLAVE_ZONE_N, [102](#)
YADIFA_SLAVE_ZONE_x, [102](#)
YADIFA_SLAVE_ZONE_x_ALLOW_-
QUERY_N, [102](#)
YADIFA_SLAVE_ZONE_x_ALLOW_-
QUERY_x, [102](#)
YADIFA_SLAVE_ZONE_x_MASTER,
[102](#)