

Paquetage ISDN

Version 3.10.19

Frank Meyer
courriel: frank@fli4l.de

L'équipe fli4l
courriel: team@fli4l.de

2 février 2020

Table des matières

1. Documentation du paquetage ISDN	3
1.1. ISDN - Communication avec les cartes ISDN (ou Numéris) actives et passives .	3
1.1.1. Établir une connexion par ISDN	3
1.1.2. Carte ISDN	4
1.1.3. OPT_ISDN_COMP (Expérimental)	8
1.1.4. Circuits ISDN	8
1.1.5. OPT_TELMOND - Configuration telmond	17
1.1.6. OPT_RCAPID - Le démon CAPI distant	19
A. Annexe du paquetage ISDN	21
A.1. ISDN	21
A.1.1. Détails techniques sur la connexion et le routage ISDN	21
A.1.2. Messages d'erreur du sous-système ISDN (Documentation-i4l en Anglais)	22
Table des figures	24
Liste des tableaux	25
Index	26

1. Documentation du paquetage ISDN

1.1. 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'utilisé 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'`

1.1.1. 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` (Page ??) 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 ??) 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 établit 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'
```

1.1.2. 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)

1. Documentation du paquetage ISDN

Type	Carte	Paramètres nécessaires
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)
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 ¹	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

1. Indiquer le type 84 qui correspondait au type 39 ancienne version.

1. Documentation du paquetage ISDN

Type	Carte	Paramètres nécessaires
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	HFCS-USB Adapters	no parameter
304	AVM Fritz !Card PCI (v1 and v2) cards	no parameter
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 - Berkom 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 ??). 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 fi4l-FAQ ou sur Mailing-liste au sujet des numéros de type pour les pilotes de cartes ISDN, si vous ne saviez 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 ??). 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 fi4l. 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 ??).

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/>

1.1.3. OPT_ISDN_COMP (Expérimental)

Si vous 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 Debugage
- '1' Information normale de Debugage
- '2' Information élargie de Debugage
- '3' Information total de Debugage (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.

1.1.4. 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 isdn4linux 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 `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'
```

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 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, 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 activer 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, `Imonc` 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 ??) 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 21).

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 régleme 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 AAAAAAAAAAATTTTTT#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 ca 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 raccoche, 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 serais 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 utiliser l'agrégation des canaux, le deuxième numéros est a 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 ipppd, vous devez activez la variable ISDN_CIRC_x_DEBUG régler celle-ci sur 'yes'. Ces informations supplémentaire sur ipppd 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 ??)).

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 ??)).

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 :

```
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.*

Important: *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 cet exemple les jours ouvrables de 18-09 Heure sont 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'
```

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 allemands sont traités comme un dimanche.

1.1.5. OPT_TELMOND - Configuration telmond

On utilise la variable OPT_TELMOND pour activer le serveur-telmond. Il écoute via le port TCP 5001 les appels téléphoniques entrants 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 l'accès du port pour le réseau LAN, cela est possible en utilisant la variable de configuration de telmond, 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 modifiée 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'imonc 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/persistent/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 `TELMONS_MSN_N`.

Configuration par défaut : `TELMOND_MSN_N='0'`

TELMOND_MSN_x Vous devez indiquer pour chaque Filtre-MSN (ou le 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 visible, 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 facultatif 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 :

```
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 appel 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
```

1.1.6. 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

1. Documentation du paquetage ISDN

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 utilisées 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.

A. Annexe du paquetage ISDN

A.1. ISDN

A.1.1. 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 choisie 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 mise 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 `ipppd`, 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 `ipppd`).
- Puis l'interface sera de nouveau reconfigurée et le routage fonctionne à nouveau indépendamment de la configuration d'origine.

La configuration du circuit sera paramétrée 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.

A.1.2. 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.
- 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.

A. Annexe du paquetage ISDN

- 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.

Table des figures

Liste des tableaux

Index

ISDN_CIRC_N, [9](#)
ISDN_CIRC_x_AUTH, [15](#)
ISDN_CIRC_x_BANDWIDTH, [10](#)
ISDN_CIRC_x_BUNDLING, [10](#)
ISDN_CIRC_x_CALLBACK, [13](#)
ISDN_CIRC_x_CBDELAY, [14](#)
ISDN_CIRC_x_CBNUMBER, [14](#)
ISDN_CIRC_x_CHARGEINT, [15](#)
ISDN_CIRC_x_CLAMP_MSS, [11](#)
ISDN_CIRC_x_DEBUG, [14](#)
ISDN_CIRC_x_DIALIN, [13](#)
ISDN_CIRC_x_DIALOUT, [13](#)
ISDN_CIRC_x_EAZ, [14](#)
ISDN_CIRC_x_FRAMECOMP, [12](#)
ISDN_CIRC_x_HEADERCOMP, [11](#)
ISDN_CIRC_x_HUP_TIMEOUT, [15](#)
ISDN_CIRC_x_LOCAL, [11](#)
ISDN_CIRC_x_MRU, [11](#)
ISDN_CIRC_x_MTU, [11](#)
ISDN_CIRC_x_NAME, [9](#)
ISDN_CIRC_x_PASS, [12](#)
ISDN_CIRC_x_REMOTE, [11](#)
ISDN_CIRC_x_REMOTENAME, [12](#)
ISDN_CIRC_x_ROUTE_N, [13](#)
ISDN_CIRC_x_ROUTE_X, [13](#)
ISDN_CIRC_x_SLAVE_EAZ, [14](#)
ISDN_CIRC_x_TIMES, [15](#)
ISDN_CIRC_x_TYPE, [9](#)
ISDN_CIRC_x_USEPEERDNS, [9](#)
ISDN_CIRC_x_USER, [12](#)
ISDN_DEBUG_LEVEL, [7](#)
ISDN_FILTER, [8](#)
ISDN_IO, [4](#)
ISDN_IO0, [4](#)
ISDN_IO1, [4](#)
ISDN_IP, [4](#)
ISDN_LZS_COMP, [8](#)
ISDN_LZS_DEBUG, [8](#)
ISDN_LZS_TWEAK, [8](#)
ISDN_MEM, [4](#)
ISDN_PORT, [4](#)
ISDN_TYPE, [4](#)
ISDN_VERBOSE_LEVEL, [7](#)

OPT_ISDN, [3](#)
OPT_ISDN_COMP, [8](#)
OPT_RCAPID, [19](#)
OPT_TELMOND, [17](#)

RCAPID_PORT, [20](#)

TELMOND_CAPI_CTRL_N, [19](#)
TELMOND_CAPI_CTRL_x, [19](#)
TELMOND_CMD_N, [18](#)
TELMOND_CMD_x, [18](#)
TELMOND_LOG, [17](#)
TELMOND_LOGDIR, [17](#)
TELMOND_MSN_N, [17](#)
TELMOND_MSN_x, [18](#)
TELMOND_PORT, [17](#)