Linux Dial-On-Demand mini-HOWTO

Rodney van den Oever, `Rodney.van.den.Oever@tip.nl` Version 1.0, 28 Juillet 1996 mise à jour le 29 février 1997

(Adaptation française par Olivier Fourmaux, `Olivier.Fourmaux@lip6.fr`, 28 Novembre 1997). Mini-HOWTO décrivant comment utiliser une machine de son réseau local comme routeur PPP avec connexion à la demande. Le **dial-on-demand** correspond à l'utilisation au besoin du modem, soit littéralement : *composition à la demande* que nous traduisons en **connexion à la demande**.

Table des matières

1 Préambule

1.1 Blabla légal

1.2 Blabla de l'auteur (*Disclaimer*)

Ce document décrit une configuration qui fonctionne pour moi, utilisez la à vos propres risques. S'il manque quoi que ce soit d'important ou si quelque chose est complètement faux, s'il vous plait, communiquez le moi !

Une chose que l'on peut rajouter, c'est la manière de configurer un serveur de noms local.

1.3 Blabla du traducteur

Les commentaires sur la traduction sont à envoyer à Olivier Fourmaux, `Olivier.Fourmaux@lip6.fr`. De même si vous trouvez des erreurs de fond, demandez moi d'abord, j'ai peut-être traduit un peu vite :)

2 Introduction

Ce document décrit comment gérer une connexion *dial-on-demand* et comment utiliser un hôte Linux connecté à un réseau local en tant que *routeur PPP* gérant la connexion à la demande.

Le *kerneld-mini-HOWTO* décrit déjà comment kerneld (daemon du noyaux linux) exécute le script `request-route` pour l'ajout dynamique de route. Pour plus d'information, le *kerneld-mini-HOWTO* existe en francais. Ici nous ne décrivons qu'une configuration détaillé adaptée à l'ajout de route avec une connexion modem.

Si vous utilisez un petit réseau local chez vous, la connexion à la demande fournit une solution intéressante de gestion transparente d'un accès Internet pour plusieurs machines.

```
  ------------------------------
  __|__                 ___|___            Liens PPP vers le
 |     |  reseau local  |       |   ______ fournisseur d'acces
 | PC  |  192.168.1.0   |Routeur|--| modem|------------------>
 |     |                | Linux |  |______|       x.x.x.x
 |_____|                |_______|             (adresse IP
                          named                   dynamique)
                          pppd
```

Cette configuration nécessite :

- Noyau 2.0.0 ou plus récent

- `ppp-2.2.0f.tar.gz` ou plus récent
- `modules-2.0.0.tar.gz` ou plus récent
- `ipfwadm-2.3.tar.gz` (`http://www.xos.nl/linux/ipfwadm`)

Les étapes suivantes sont à réaliser :

1. Configuration du noyau
2. Installation des modules
3. Installation de PPP
4. Configuration du port série et du modem
5. Configuration de PPP
6. Installation de la connexion à la demande
7. Configuration du réseau
8. Configuration de l' IP-Masquerade
9. Configuration des stations

3 Configuration du Noyau

Avant tout, il faut compiler un noyau avec *kerneld*, *PPP* et *IP-masquerade*. Ci-dessous sont indiquées les options nécessaires pour cette configuration (en Majuscule), les autres options réseau telles que *firewall logging* ou *transparent proxying* peuvent être choisies mais ne sont pas obligatoires.

```
cd /usr/src/linux
make config
```

ou

```
make menuconfig
```

```
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [Y/n/?]
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?]
Set version information on all symbols for modules (CONFIG_MODVERSIONS) [N/y/?]
```

```
Kernel daemon support (e.g. autoload of modules) (CONFIG_KERNELD) [Y/n/?]
*
Networking support (CONFIG_NET) [Y/n/?]
*
* Networking options
*
Network firewalls (CONFIG_FIREWALL) [Y/n/?]
TCP/IP networking (CONFIG_INET) [Y/n/?]
IP: forwarding/gatewaying (CONFIG_IP_FORWARD) [Y/n/?]
IP: firewalling (CONFIG_IP_FIREWALL) [Y/n/?]
IP: masquerading (EXPERIMENTAL) (CONFIG_IP_MASQUERADE) [Y/n/?]
*
* Protocol-specific masquerading support will be built as modules.
*
IP: always defragment (CONFIG_IP_ALWAYS_DEFRAG) [Y/n/?]
IP: optimize as router not host (CONFIG_IP_ROUTER) [Y/n/?]
*
* Network device support
*
Network device support (CONFIG_NETDEVICES) [Y/n/?]
*
PPP (point-to-point) support (CONFIG_PPP) [M/n/y/?]
*
* CCP compressors for PPP are only built as modules.
*
Ethernet (10 or 100Mbit) (CONFIG_NET_ETHERNET) [Y/n/?]
NE2000/NE1000 support (CONFIG_NE2000) [Y/m/n/?]
```

Cette dernière option dépend bien sur de la carte réseau utilisée.

Compilez le noyau avec :

```
make zlilo
make modules
make modules_install
```

Si vous n'utilisez pas LILO, copiez `/usr/src/linux/arch/i386/boot/zImage` à la bonne place pour l'initialisation. Les modules sont installés dans `/lib/modules/x.x.x`, avec `x.x.x` correspondant à la version du noyau utilisé.

4 Installation des Modules

Les programmes suivants sont nécessaires :

/sbin/insmode

Charge les modules nécessaires.

/sbin/rmmod

Supprime les modules inutiles.

/sbin/depmod

Crée les dépendances.

/sbin/kerneld

Extension de `Kerneld` pour charger les modules automatiquement.

Si vous ne les avez pas encore, installez les avec :

```
cd /usr/src
tar zxf modules-2.0.0.tar.gz
make install
```

Pour charger *Kerneld* au démarrage, vous devez éditer `/etc/rc.d/rc.S` ou créer un script `rc.modules` qui est appelé par `rc.S`. Ajoutez ou modifiez les lignes suivantes :

```
/etc/rc.d/rc.modules:

if [ ! -f /lib/modules/`uname -r`/modules.dep ] ; then
        echo "Updating module dependencies for Linux `uname -r`:"
        /sbin/depmod -a
fi

[ -x /sbin/kerneld ] && /sbin/kerneld
```

La première partie vérifie les dépendances des modules et les crée si elles n'existent pas encore.

5 Installation de PPP

Les programmes suivants sont nécessaires pour l'installation d'un lien PPP :

/usr/sbin/pppd

Daemon PPP.

/usr/sbin/pppstats

Statistiques PPP.

/usr/sbin/chat

Exécution de scripts.

A installer avec :

```
cd /usr/src
tar zxf ppp-2.2.0f.tar.gz
cd ppp-2.2.0f
./configure
make kernel
```

Les noyaux récents incluent le code de PPP, `make kernel` ne remplacera donc rien du tout. Vérifier les fichiers `README` et `Makefile` pour le support du kit *Shadow*. Exécutez alors :

```
make
make install
```

6 Configuration du Port Série et du Modem

Cette configuration nécessite un modem. La configuration du port serie est réalisée dans le fichier `rc.serial` :

```
/etc/rc.d/rc.serial:

${SETSERIAL} /dev/cua0 uart 16450 port 0x3F8 irq 4
${SETSERIAL} /dev/cua1 uart 16550A port 0x2F8 irq 3 spd_vhi
```

Dans cette configuration, COM1 (`cua0`) est utilisé par la souris et COM2 (`cua1`) par un modem interne à 28,8 kb/s. Le paramètre `spd_vhi` remplace la valeur `38400 bps` par `115200 bps` dans la table des vitesses, de même `spd_hi` est utilisé pour la valeur `57600 bps`.

Il faut utiliser `spd_vhi` pour un modem a 28,8 kb/s et `spd_hi` pour un à 14,4 kb/s. Mais on peut démarrer par prudence avec la valeur `38400 bps`.

Vérifiez votre configuration modem à l'aide d'un programme de type émulation de terminal, tel que *minicom*. Le modem doit afficher (*echo*) les caractères frappés. Utilisez `Ctrl-A P` pour changer les paramètres de communication. Vérifiez si cela fonctionne encore à 57600 ou 115200 b/s. Tapez `ATZ`, le modem devrait répondre `OK`. Utilisez `Ctrl-A X` pour quitter minicom.

Minicom peut aussi être utilisé pour se connecter à son fournisseur de services manuellement. Notez préalablement ce que vous devez taper pour initier la connexion et quelles sont les réponses du serveur distant. Si votre fournisseur de services n'utilise pas PAP ou CHAP (protocoles d'échange de mots de passe), il faudra probablement entrer un nom d'utilisateur et un mot de passe, et peut être aussi une commande pour initialiser la liaison en mode PPP.

7 Configuration de PPP

Le daemon `pppd` recherche dans le repertoire `/etc/ppp` les scripts et le fichier `options` :

```
/etc/ppp/options:

/dev/modem 38400 crtscts modem noipdefault defaultroute idle-disconnect
120
```

Ce qui signifie :

/dev/modem
: Dans ma configuration, un lien vers `/dev/cua1`.

38400
: A remplacer par 115200 (paramètre `spd_vhi` dans `/etc/rc.d/rc.serial`)

crtscts
: Contrôle de flux matériel, RTS/CTS, *ne pas utiliser le controle de flux Xon-Xoff!*

modem
: Utilisation avec un modem.

noipdefault
: L'adresse IP est attribuée automatiquement.

defaultroute
: Affecte la route par défaut à l'interface PPP.

idle-disconnect
: Avec PPP-2.2.0f déconnecte automatiquement un lien inutilisé apres 120 secondes.

Nous créons maintenant un script, nommé `defaut`, pour la connexion vers notre fournisseur de service habituel. Pour d'autres connexions, créer d'autres scripts vers ces autres sites et les nommer avec leur adresse IP distante, par exemple.

```
/etc/ppp/defaut:

"" ats11=55dtXXXXXXXX CONNECT "" rname: UUUUU sword: PPPPP > ppp
```

Ce qui signifie :

ats11=55
: Racourcit la durée des tonalités pour accélérer la connexion.

XXXXXXXX
: Correspond au numéro de téléphone du fournisseur de service.

CONNECT

`chat` attend que le modem renvoi `CONNECT`.

rname:

`chat` attend `rname`, et entre `UUUUU`.

sword:

`chat` attend `sword`, et entre `PPPPP`.

Remplacez `UUUUU` et `PPPPP` par votre nom d'utilisateur et votre mot de passe. Utilisez `rname` et `sword` (à la place de `username` et `password`) car quelquefois les premières lettres sautent ou sont soit en majuscules soit en minuscules. Certains fournisseurs de services demandent l'entrée du mot `ppp` après le nom d'utilisateur et le mot de passe, d'où l'attente de l'invite du routeur (>).

Si vous avez des erreurs du type `serial line looped back` ou `serial line not 8-bit clean`, la liaison n'est pas encore sous PPP.

La commande de lancement est assez longue pour mériter un petit script :

```
/etc/ppp/ppp:

#!/bin/sh
/usr/sbin/pppd connect '/usr/sbin/chat -v -f /etc/ppp/default'
```

Vous devriez être capable de vous connecter à votre fournisseur de service. Vous pouvez rajouter le paramètre `debug` au fichier `/etc/ppp/options` et augmenter le *loglevel* de `syslogd` pour voir ce qui ce passe :

```
/etc/syslog.conf:

*.debug                                /var/log/messages
```

Si vous éditez `syslog.conf`, soyez sûr d'indiquer au daemon `syslogd` le changement. Suivant la localisation de son fichier PID, faîtes :

```
kill -1 `cat /var/run/syslog.pid`
```

ou

```
kill -1 `cat /etc/syslog.pid`
```

Après que la connexion soit établie, vérifiez avec `ifconfig` l'existence de l'interface `ppp0`. Vérifiez la route par défaut avec `netstat -r`, elle devrait indiquer le fournisseur de service. Essayer aussi de *pinguer* (`ping`) un serveur distant, tel que le serveur de noms du fourniseur de service. Si ça marche, vous pouvez ajouter ce serveur de noms dans le fichier `/etc/resolv.conf` :

```
/etc/resolv.conf:

search your.isp.domain
nameserver x.x.x.x
```

8 Installation de la Connexion à la Demande

Le daemon `kerneld` peut être utilisé pour charger à la demande des modules tels que PPP. Une autre possibilité signalée dans le **kerneld-HOWTO** est le lancement d'un script si une route est absente. `Kerneld` surveille le fichier `/sbin/request-route`. Le script qui suit est la copie de celui fourni dans `modules-2.0.0.tar.gz`:

```
/sbin/request-route:

#!/bin/sh
LOCK=/var/run/request-route.pid
export PATH=/usr/sbin:$PATH     # for ppp-2.2*
#
# Ce script sera appele par kerneld avec la route demandee $1
#
chatfile=/etc/ppp/$1
#
# Si aucun script chat utilise une valeur par defaut (lien symbolique
# vers le script prefere) :
#
[ ! -f $chatfile ] && chatfile=/etc/ppp/default
#
# V\'{e}rifie si le programme tourne toujours :
#
if [ ! -f $LOCK ] ; then
        pppd connect "chat -f $chatfile" &
        #
        # Temporisation supprimee par ip-up, a regler. Verifier les
        # delais de kerneld aussi.
        #
        sleep 60 &
        sleepid=$!
        echo $sleepid > $LOCK
        wait $sleepid
        rm -f $LOCK
        exit 0
fi
exit 1
```

Kerneld exécute ce script en utilisant la route absente comme argument :

```
/sbin/request-route x.x.x.x
```

Le script utilise un fichier verrou pour eviter plusieurs request-route simultanés.

Au début, j'utilisais un script qui était déclenché par une requête au serveur de noms, mais depuis je préfère fonctionner avec un serveur de noms local, le resultat sera dans la mémoire cache et un second accès ne déclenchera pas la connexion. C'est la raison pour laquelle j'ai changé le script pour utiliser un script chat si aucun n'existe.

Cela permet aussi de créer de multiples scripts chat pour les différentes liaisons et d'utiliser un script chat par défaut pour se connecter à son fournisseur de service.

Après que la liaison ait été initiée, le daemon pppd exécute le script /ppp/ip-up qui retire le fichier verrou. Ajoutez les commandes que vous voulez exécuter à chaque fois que la liaison est initiée, par exemple surveiller la queue de **sendmail** :

```
#!/bin/ash
LOCK=/var/run/request-route.pid
[ -f $LOCK ] && kill `cat $LOCK`
#
# Execute ces commandes apres la connexion (exemple) :
#
/usr/sbin/sendmail -q
```

L'option idle-disconnect déconnecte automatiquement une liaison inutilisée. Pour se déconnecter immédiatement, utilisez ppp-off :

```
/usr/sbin/ppp-off:.

#!/bin/sh
LOCK=/var/run/request-route.pid
DEVICE=ppp0

rm -f $LOCK
#
# Si le fichier ppp0.pid est present alors le  programme tourne. Stoppez
# le.
#
if [ -r /var/run/$DEVICE.pid ]; then
        kill -INT `cat /var/run/$DEVICE.pid`
#
# Si ca marche, verifier que le fichier pid est retire.
```

```
#
        if [ ! "$?" = "0" ]; then
                echo "removing stale $DEVICE pid file."
                rm -f /var/run/$DEVICE.pid
                exit 1
        fi
#
# Reussite. Terminer avec le bon statut.
#
        echo "$DEVICE link terminated"
        exit 0
fi
#
# La liaison n'est pas active
#
echo "$DEVICE link is not active"

exit 1
```

9 Configuration du Réseau

Si votre machine Linux a une autre interface réseau, tel qu'une carte ethernet ou si vous utilisez PLIP pour vous connecter vers un autre hôte, vous pouvez configurer la machine multi-interface comme un routeur. C'est ainsi que j'ai configuré mon réseau. J'entre toujours toutes mes adresses IP dans `/etc/hosts` pour n'avoir qu'un fichier à modifier.

```
/etc/hosts:

127.0.0.1       localhost
10.0.0.1        your.domain.com host
10.0.0.2        some.other.machine
```

```
/etc/networks:

loopback        127.0.0.0
localnet        10.0.0.0
```

```
/etc/rc.d/inet1:

# Loopback interface:
ifconfig lo localhost
route add -net loopback
```

```
# Local interface:
ifconfig eth0 host
route add -net localnet eth0
```

La dernière ligne ajoute une route au réseau local. Nous n'avons pas besoin d'entrer la passerelle par défaut ici, puisque nous n'en avons pas, tant que la liaison PPP n'existe pas.

10 Configuration de l'IP-Masquerade

Si vous devez connecter différentes machines, mais possédez seulement une adresse IP, vous pouvez utiliser les fonctionnalités d'IP-masquerade de Linux. Cela vous permet de configurer des adresses IP interne à un réseau qui seront converties en l'adresse IP assignée à l'intreface PPP.

Vous pouvez utiliser librement les adresses de réseau suivantes :

```
Class A: 10.0.0.0
Class B: 172.16.0.0 - 172.31.0.0
Class C: 192.168.0.0 - 192.168.254.0
```

(NdT : *à utiliser sans angoisse, adresses non routées dans l'Internet*)

Ajouter ensuite les lignes suivantes à `/etc/rc.d/rc.inet1` :

```
/etc/rc.d/rc.inet1:

FW=/sbin/ipfwadm

${FW} -F -f

${FW} -F -a m -P tcp -S localnet/8 -D default/0
${FW} -F -a m -P udp -S localnet/8 -D default/0

# Masquerading timeout-values (tcp tcpfin udp), defaults:
${FW} -M -s 3600 0 0

# Load module for special protocols:
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
/sbin/modprobe ip_masq_irc
```

Les deux lignes d'IP-masquerade vont adapter les trafics UDP et TCP à l'adresse IP utilisée par la passerelle par défaut (l'interface PPP). Les autres protocoles ne fonctionneront pas à partir des autres hôtes. Par exemple `ping` qui utilise le protocole ICMP, ne fonctionnera que du routeur. Ici on utilise le réseau **10.0.0.0** avec le masque de sous réseau **255.0.0.0**, ou en notation CIDR : **/8**.

Les lignes suivantes évitent la fin prématurée des connexions FTP, en augmentant la valeur des temporisation TCP (timeout).

Les modules sont nécessaires seulement si vous désirez utiliser ces protocoles à travers un pare-feu (firewall). C'est à dire, si vous voulez utiliser un FTP normal à travers un pare-feu, vous devez charger le module `ip_masq_ftp`.

11 Configuration des Stations

Configurez vos autres machines (stations) pour votre réseau local, puis pour utiliser la machine Linux (routeur) comme passerelle par defaut.