Partage de connexion sous Debian Linux

Y. Morère

Juillet 2003

Résumé

Cette article est tiré d'une expérience personnelle. Je tiens à remercier Anthony pour son aide pour la configuration d'IPtables.

Un grand merci aussi au créateur du site http://christian.caleca.free.fr/, qui a été d'une grande utilité à la compréhension de divers concepts.

Table des matières

1 Introduction **2**

2 Installation des cartes réseaux. **3**

3 Configurations du réseau **4**

4 Installation du serveur DHCP(Dynamic Host Control Protocol) **6**
4.1 Installation des clients DHCP 7
4.1.1 Client Win98 7
4.1.2 Client indows NT4/2000/XP 9
4.1.3 Client Linux 9

5 Installation du Firewall **10**
5.1 La table de filtrage 11
5.2 La table de translation d'adresses 11
5.3 La table Mangle 12
5.4 Pratique 12
5.5 Remarques 16

6 Installation du DNS (Domaine Name Server) **16**
6.1 Le serveur Domain Name System. 16
6.2 Mise en place 17

7 Installation d'un modem standard **25**
7.1 Modem PCMCIA 26
7.2 Modem externe sur port série 26
7.3 Vérification du modem 26
7.4 Configuration de la connexion vers le fournisseur d'accès 26

8 Installation d'un partage de connextion à l'aide d'un routeur/firewall **28**
8.1 Installation et câblage 28
8.2 Configuration des paramètres de connexion 29

1 Introduction

Je considère tout d'abord que vous avez une distribtion linux installée sur votre machine. Pour ma part, tout l'article est basé sur une Debian Woody, installée sur un pentium 133, 40Mo de Ram et 1.2Go de disque dur.

J'ai pas d'interface graphique et tout se fera avec VI et le mode console.

Je considère aussi que vous savez installer (physiquement) une carte réseau dans votre machine.

Le but ultime de cet article est de présenter l'installation d'une passerelle internet sous linux, qui comprend un serveur DHCP, une DNS pour votre sous-réseau local, et un firewall.

Remarque : Les adresses IP sont fantaisistes afin d'éviter tout problèmes avec certains petits malins.

2 Installation des cartes réseaux.

Comme la manip s'est effectuée sur du matériel de récupération, et que la machine ne servira que de passerelle, l'installation et la configuration des carte réseau fut un peu plus longue.

Il s'agit en effet de carte réseau ISA, non PNP, compatible NE2000. De plus elles sont quasiment identiques.

À l'aide de l'utilitaire `modconf`, installez les modules correspondants à vos carte réseau.

Il se peut aussi que votre carte réseua soit déjà gérée dans le noyau ; pour vérifier cela, rien de tel qu'un petit `dmesg | grep eth0` devrai faire l'affaire.

Pour des information plus complète je vous renvoie sur le site de léalinux http://lea-linux.org/reseau/gateway.php3

Si comme moi, les cartes réseaux ne sont pas détectées automatiquement par les modules, il faut passer en paramètres l'adresse d'E/S et l'irq pour chaque carte réseau, soit par l'intermédiaire de `modconf`, soit directement par les fichier `/etc/modules` et `/etc/modules.conf`.

Si vous n'avez pas la documentation des cartes (parfois la configuration se trouve écrite sur la carte), il faudra essayer les différentes conbinaisons irq adresse E/S, ou encore tester les cartes sur une machines windows qui vous donnera les précieuses informations.

Il reste encore le bon vieux `pnpdump` livré avec `isapnp`, mais il n'a pas été d'un grand secours pour moi (carte non PNP).

voici donc les fichiers `/etc/modules` et `/etc/modules.conf` :

```
yann@pport:/etc$ more modules
# /etc/modules: kernel modules to load at boot time.
#
# This file should contain the names of kernel modules that are
# to be loaded at boot time, one per line.  Comments begin with
# a "#", and everything on the line after them are ignored.
ne

[snip]
### update-modules: start processing /etc/modutils/ne
options ne io=0x320,0x300 irq=10,5
[snip]
```

Un petit update-modules ou un redémarrage pour voir si tout cela fonctionne. Au final vous devriez avoir cela :

```
yann@pport:/etc$ dmesg | grep eth
NE*000 ethercard probe at 0x320: 00 00 b4 3f ff 92
eth0: NE2000 found at 0x320, using IRQ 10.
NE*000 ethercard probe at 0x300: 00 40 05 54 d2 bd
eth1: NE2000 found at 0x300, using IRQ 5.
yann@pport:/etc$
```

Nous deux cartes réseaux sont maintenant reconnues. Il nous reste à les configurer.

Afin de connaître la configuration dez votre carte réseau, vous pouvez utiliser les utilitaires du paquer `mii-diag`, un petit outil pour manipuler les carte réseau.

Voici un exemple de configuration d'une carte 3com configurée en 100Mbits full duplex. Il a fallu forcé cette configuration à l'aide `modconf`, ou vous trouveres les explications des options. Cela se traduit par l'option `option=8`.

```
tuxpowered# mii-diag eth0
Basic registers of MII PHY #24:  3000 782d 0040 6176 05e1 45e1 0003 0000.
 The autonegotiated capability is 01e0.
The autonegotiated media type is 100baseTx-FD.
 Basic mode control register 0x3000: Auto-negotiation enabled.
 You have link beat, and everything is working OK.
 Your link partner advertised 45e1: Flow-control 100baseTx-FD 100baseTx 10baseT-FD 10bas
   End of basic transceiver information.
tuxpowered#
```

Il est possible de tester différentes configurations à l'aide de `mii-tool`, par exemple pour forcer en 100Mbits FullDuplex :

```
mii-tool --force=100baseTx-FD eth0
```

Si tout se passe bien, il suffit alors de relancer `modconf` en appliquant les bonnes options afin d'avoir le bon mode activé au démarrage.

3 Configurations du réseau

Dans mon exemple, ma machine possède un adresse IP fixe dans le réseau de l'université, cette machine servira de passerelle pour un sous réseau local à mon bureau (1 machine pour le moment). Le réseau de la Fac arrivera sur l'interface ETH0 et le réseau local sera connecté à l'interface ETH1.

Pour le réseau local, nous choisirons un réseau de classe C en 192.168.1.0. L'adresse de la passerelle dans ce réseau sera 192.168.1.254.

tout ceci nous donne le fichier `/etc/network/interfaces` suivant :

```
yann@pport:/etc$ more /etc/network/interfaces
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

# The loopback interface
auto lo
iface lo inet loopback

# The first network card - this entry was created during the Debian installation
# (network, broadcast and gateway are optional)
#reseau fac
auto eth0
iface eth0 inet static
        address 160.60.2.101
        netmask 255.255.255.0
        network 160.60.2.0
        broadcast 160.60.2.255
        gateway 160.60.2.254

#reseau local
auto eth1
iface eth1 inet static
```

```
        address 192.168.1.254
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255

yann@pport:/etc$
```

Le fichier `/etc/hosts` :

```
yann@pport:/etc$ more /etc/hosts
127.0.0.1       localhost
160.60.2.153  pport.lasc.sciences.univ-metz.fr        pport

# The following lines are desirable for IPv6 capable hosts
# (added automatically by netbase upgrade)

::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
yann@pport:/etc$
```

et le fichier `/etc/resolv.conf` qui contient l'adresse du DNS de la fac.

```
yann@pport:/etc$ more /etc/resolv.conf
search lasc.sciences.univ-metz.fr
nameserver
yann@pport:/etc$
```

Il ne reste plus qu'a faire un `/etc/init.d/networking restart` pour redémarrer le réseau avec ces nouvelles configurations. Pour vérifier que les interfaces sont bien actives :

```
yann@pport:/etc$ /sbin/ifconfig
eth0      Lien encap:Ethernet  HWaddr 00:00:B4:3F:FF:92
          inet adr:160.60.2.153  Bcast:160.60.2.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7712 errors:0 dropped:0 overruns:0 frame:0
          TX packets:584 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          RX bytes:893247 (872.3 KiB)  TX bytes:79661 (77.7 KiB)
          Interruption:10 Adresse de base:0x320

eth1      Lien encap:Ethernet  HWaddr 00:40:05:54:D2:BD
          inet adr:192.168.1.254  Bcast:192.168.1.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

```
          Interruption:5 Adresse de base:0x300

lo        Lien encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:22 errors:0 dropped:0 overruns:0 frame:0
          TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:1925 (1.8 KiB)  TX bytes:1925 (1.8 KiB)

yann@pport:/etc$
```

4 Installation du serveur DHCP(Dynamic Host Control Protocol)

d'après http://christian.caleca.free.fr/.

Ce protocole permet aux administrateurs de réseaux TCP/IP de configurer les postes clients de façon automatique. Il a été utilisé par les fournisseurs d'accès à l'Internet par le câble, mais a été abandonné au profit d'une connexion point à point type PPP, comme pour l'ADSL.

DHCP reste cependant un protocole de configuration de clients extrêmement pratique sur un réseau local Ethernet.

Bien que dans la plupart des cas, DHCP soit un luxe sur un réseau domestique, il peut tout de même y avoir plusieurs raisons pour vous pousser à l'utiliser :

- ▷ Vous avez des portables que vous connectez sur divers réseaux, typiquement chez vous et sur votre lieu de travail (si votre administrateur vous laisse faire, c'est qu'il est bien confiant :-)),
- ▷ vous organisez chez vous des "Lan parties" avec les machines de vos collègues,
- ▷ votre réseau local contient plusieurs dizaines de machines (vous avez une famille nombreuse, peut-être),
- ▷ vous aimez bien vous compliquer la vie à bricoler avec votre Linux,
- ▷ vous aimez le luxe, tout simplement.

Il faut tout d'abord installer un serveur DHCP sur votre machine. Je vous conseille le paquet `dhcp3` de la distribution debian. Il n'est pas conseillé d'installer le client dhcp sur la machine serveur. Pour cela vous pouvez passer par `dselect` ou encore un bon vieux `apt-get install dhcp3-server`.

dans la cas de la configuration sous mandrake, je vous renvoie à la page http://christian.caleca.free.fr/dhcp/serveur_dhcp.htm.

Il faut ensuite configurer le serveur DHCP. Cela passe par le fichier de configuration `/etc/dhcp3/dhcp.conf`

```
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#

# option definitions common to all supported networks...
# Durée de vie du bail
default-lease-time 86400;
max-lease-time 86400;
```

```
# Les options que l'on va refiler aux clients
option domain-name "ecole-belan.org";
option domain-name-servers 192.168.1.254;

option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;

# La définition du seul "sous-réseau" dont nous disposons
# Avec la plage d'IP à distribuer.
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.1 192.168.1.253;
}
```

Voici pour un réseau simple. Le ligne sont assez facile à comprendre. Les deux première concerne la durée de vie du bail de connexion de la machine cliente.

Ensuite viennes les options pour les clients. On retrouve entre autre le nom de domaine du sous-réseau (pour notre futur DNS) ainsi que le serveur de ce domaine qui est notre machine.

Ensuite viennent les options propres au sous-réseau.

Avant de lancer le serveur DHCP, il faut vérifier que ce dernier écoute bien sur la bonne carte réseau ses clients. Pour cela il faut se rendre dans le fichier `/etc/default/dhcp3-server` et posistionner la variable `INTERFACES` sur la bonne carte réseau. Dans notre cas, il s'agit d'ETH1.

```
pport:/etc#  more /etc/default/dhcp3-server
# Defaults for dhcp initscript
# sourced by /etc/init.d/dhcp
# installed at /etc/default/dhcp3-server by the maintainer scripts

#
# This is a POSIX shell fragment
#

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth1"
pport:/etc#
```

Il ne reste plus qu'a relancer le serveur DHCP par `/etc/init.d/dhcp3-server restart`.

4.1 Installation des clients DHCP

4.1.1 Client Win98

On va prendre l'exemple d'une machine sous windows 98. Il faut tout d'abord dans la configuration TCP/IP, enlever tout ce qu'il y a concernant l'IP, le masque de sous réseau, DNS, passerelle et juste dire que vous voulez une configuration dynamique (DHCP). Relancez vos services réseaux, la méthode la plus simple et la plus bestiale étant le "reboot", et voilà. Une fois le système remonté, vous devez avoir hérité d'une configuration automatique.

la configuration pour notre sous réseau sera la suivante :

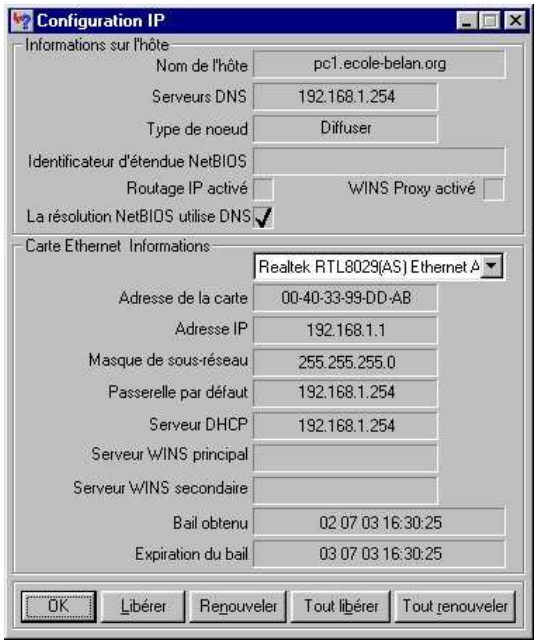

FIG. 1 – Fenêtre de `winipcfg` sous Win98

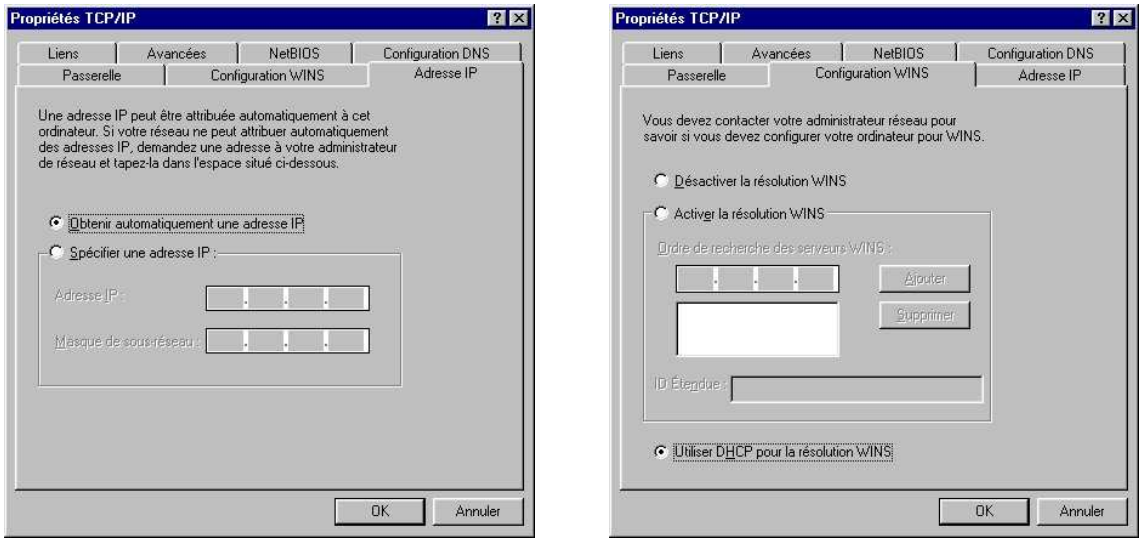

FIG. 2 – La configuration IP et WinS

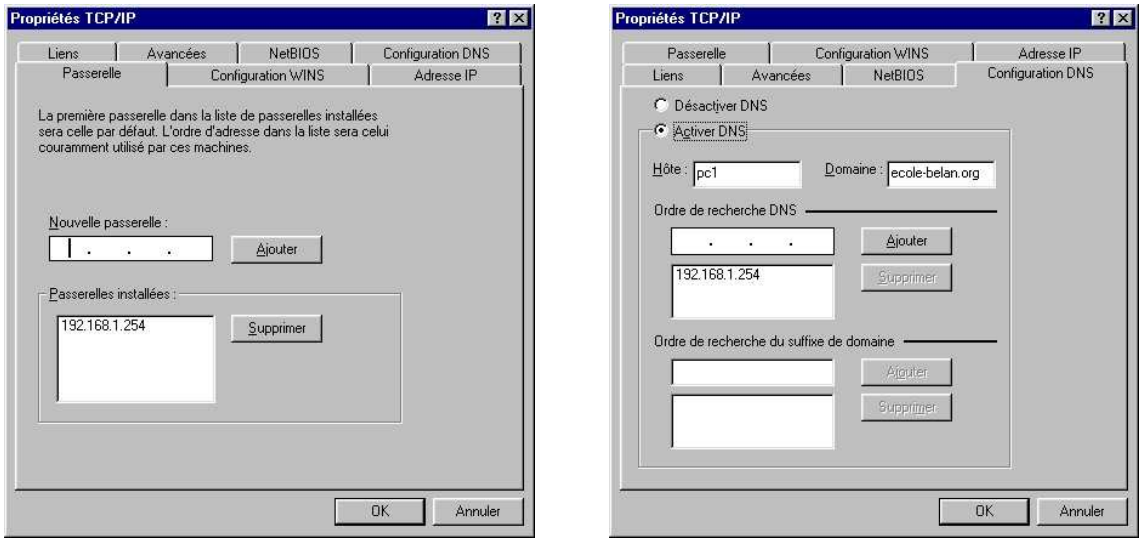

FIG. 3 – La configuration Passerelle et DNS

4.1.2 Client indows NT4/2000/XP

La configuration se fait dans le panneau de configuration, icône ”réseau”, onglet ”protocoles”, puis ”propriétés” de TCP/IP. Là, vous avez indiqué que la carte doit recevoir une adresse IP dynamiquement.

Pour une vérification, tapez dans une console, la commande ”ipconfig”

Votre adresse doit être affichée. Si vous voulez tous les détails, utilisez la commande ”ipconfig /all” :

La commande `ipconfig` permet également :
▷ De résilier le bail : `ipconfig /release`
▷ De renouveler le bail : `ipconfig /renew`
C'est cette commande qui est à utiliser pour essayer de récupérer une adresse IP lorsque vous avez des problèmes. Notes.

Les rubriques ”Bail obtenu” et ”Expiration du bail” contiennent des valeurs calculées par votre machine. Le serveur DHCP ne donne que la durée.

La commande en mode graphique ”winipcfg” n'existe pas nativement sous Windows NT mais vous pouvez la récupérer dans le kit de ressources techniques (téléchargeable sur le site MS en cherchant bien :-).

N'essayez pas d'utiliser celle de Windows 95/98, les dll winsock32 utilisées ici ne sont pas compatibles.

4.1.3 Client Linux

Tiré de http://christian.caleca.free.fr/dhcp/serveur_dhcp.htm

Avec cet OS c'est beaucoup plus compliqué, parce qu'il y a beaucoup plus de configurations possibles.

La configuration utilisée dans l'exposé est la suivante :
▷ Un portable Compaq équipé d'une carte réseau D-LINK PCMCIA
  ▷ MANDRAKE 8.2
  ▷ Eth0 et configurée avec DHClient.

Notez que DHClient n'est pas le seul client possible. Vous pouvez parfaitement le remplacer par PUMP, DHCPXD ou par DHCPCD. Tous ces clients sont disponibles dans la distribution Mandrake, qui installe d'ailleurs DHCPCD par défaut, et non pas celui que j'utilise.

- DHCPCD semble avoir la préférence du distributeur. Je n'ai jamais rencontré de problèmes avec, mais je ne l'utilise normalement pas pour la raison suivante : Son paramétrage ne se fait que par la ligne de commande, ce qui oblige à aller modifier des scripts pas toujours faciles à trouver si l'on veut par exemple utiliser son propre DNS à la place de celui proposé dans le bail.
- PUMP Fonctionne également sans problèmes, il dispose d'un fichier de configuration /etc/pump.conf dans le quel on peut par exemple spécifier très simplement que l'on ne veut pas modifier le paramétrage du DNS avec l'information récupérée par DHCP. (Le ou les DNS sont inscrits dans le fichier /etc/resolv.conf).
- Je n'ai pas vraiment étudié DHCPXD qui fonctionne lui aussi sans difficultés. Il dispose d'un répertoire /etc/dhcpxd dans lequel vous trouverez quelques fichiers qui vous donneront toutes les indications sur le bail en cours.
- DHCLIENT a ma préférence. Il est écrit par ISC (les auteurs de BIND le fameux DNS et DHCPD lque nous utilisons ici, c'est dire qu'ils savent de quoi ils parlent :). Ce client cumule à mon sens tous les avantages :
  - Un fichier de configuration /etc/dhclient.conf, sans doute encore plus performant que celui de PUMP. Notez que ce fichier n'existe pas dans la distribution Mandrake, il vous faudra éventuellement le créer si vous ne voulez pas vous contenter du fonctionnement par défaut.
  - Des scripts optionnels exécutés automatiquement avant l'obtention du bail et après l'obtention du bail, avec à disposition des variables contenant toutes les informations recueillies par le client auprès du serveur. Très pratique par exemple pour vous envoyer par mail l'adresse courante de votre machine si celle-ci change ; dans le cas par exemple où vous avez besoin de vous y connecter à distance par telnet ou ssh.
  - Il tient un historique des baux obtenus dans le fichier /var/lib/dhcp/dhclient.leases

Son seul inconvénient est sa richesse. Il n'est pas le plus facile à mettre en oeuvre.

Dans tous les cas vous devriez avoir un fichier `/etc/network/interfaces` de ce style :

```
yann@yoda:~$ more /etc/network/interfaces
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

# The loopback interface
auto lo
iface lo inet loopback

# The first network card - this entry was created during the Debian installation
# (network, broadcast and gateway are optional)
auto eth0
iface eth0 inet dhcp
yann@yoda:~$
```

5 Installation du Firewall

Pour en savoir beaucoup plus sur Netfilter, le NAT, le filtrage ... je vous renvoie à l'adresse suivante http://christian.caleca.free.fr/netfilter/.

Dans cette partie je vais être assez bref et juste donner les commandes et les choses à ne pas oublier afin de faire fonctionner votre passerelle et de protéger un peu votre réseau local par

quelques regles. Pour cette partie je remercie Anthony qui m'a fournit un script permettant de mettre en œuvre facilement ces notions.

Dans la suite nous utiliserons IPTables. IPtables est en quelques sortes l'interface utilisateur de Netfilter. Dans sa partie "visible", ça ressemble à IPchains, mais ici, ce n'est qu'une interface de commande de Netfilter. La syntaxe est plus complète et plus rigoureuse.

Nous allons mettre en œuvre 3 tables.

5.1 La table de filtrage

C'est la table qui va permettre de filtrer tous les paquets qui entrent et sortent de notre machine. Il n'y a ici aucune modification de ces paquets, ils seront comparés à des critères définis dans la table Filter. Dans notre cas, il peut se passer deux choses différentes :

▷ Un paquet qui entre est destiné à un processus de l'hôte (serveur HTTP, FTP...).

▷ Un paquet qui entre est destiné à un autre réseau, c'est alors une fonction de routage.

5.2 La table de translation d'adresses

La traduction d'adresse (NAT comme Network Address Translation) est à prendre ici au sens le plus large, puisque cette table permet non seulement de faire de la translation stricte d'adresses, mais également de la translation de ports et un mélange des deux, dont le masquage d'adresse est une forme particulière. Mais qu'est-ce que c'est exactement ?

Dans un datagramme, en plus des données, on y trouve également quelques informations concernant le protocole utilisé et des identificateurs de l'émetteur et du destinataire. Ce sont ces identificateurs qui nous intéressent :

▷ L'adresse IP du destinataire.

▷ Le port du service utilisé sur le destinataire.

Ces informations constituent une "socket", elles sont indispensables pour arriver à joindre le bon service sur le bon serveur, par exemple le service HTTP du serveur www.wanadoo.fr.

▷ L'adresse IP de l'émetteur.

▷ Le port de réponse.

Ces informations constituent une autre "socket", elles sont indispensables pour que l'émetteur d'un paquet puisse espérer recevoir une réponse.

Avec les fonctions NAT de Netfilter, Lorsqu'un paquet transite par notre passerelle, nous allons pouvoir "bricoler" ces sockets absolument comme on veut. Par exemple, nous pourrons changer l'adresse de l'émetteur ou le port de l'émetteur ou les deux. Nous pouvons aussi changer l'adresse du destinataires, ou le port du destinataire, ou les deux.

Mais à quoi ça sert ?

Ca sert à une quasi infinité de choses. Parmi les plus intéressantes, citons :

▷ Le masquage d'adresse : C'est une fonction fondamentale lorsque l'on souhaite connecter un réseau privé à l'Internet lorsque l'on ne dispose que d'une seule IP valide sur le Net, même si celle-ci est dynamique, ce qui est le cas qui nous intéresse le plus. Les clients sont sur le réseau privé et les serveurs sont sur le Net. C'est une forme particulière de SNAT (Source NAT)

C'est ce que sont capables de faire tous les routeurs SOHO (Small Office, Home Office) qui permettent de relier un petit réseau local à l'Internet, lorsque l'on ne dispose que d'un accès RTC, NUMERIS, Câble, ADSL... Un simple (très) vieux PC (un 486 suffit) équipé d'un Linux 2.4.x permet de le faire aussi bien sinon mieux.

▷ Le NAT de destination : Ici, c'est pour résoudre les problèmes qui apparaissent dans l'autre sens. Les clients sont sur le Net et les serveurs sont sur le réseau privé.

Imaginons que nous n'ayons qu'une seule IP valide sur le Net et que nous voulions tout de même offrir des services tels que HTTP, FTP, SMTP, POP et peut-être d'autres encore. Comment faire ? La réponse triviale consiste à dire : "J'ai droit à une seule IP, donc je place tous ces serveurs sur la même machine, celle qui a la seule IP à laquelle j'ai droit."

5.3 La table Mangle

Nous n'avons pas parlé de la table Mangle. Cette table permet d'effectuer un marquage des paquets.

Nous pouvions, avec les premières versions de Netfilter, marquer les paquets en PREROUTING ou en OUTPUT pour les sorties d'un processus local (en rouge sur l'illustration). Notez que depuis la version 2.4.18 du noyau, le système a été étendu à tous les "hooks" (en bleu sur l'illustration).

L'intérêt de ce marquage, qui n'est visible que dans la pile de la machine, est de pouvoir être relu par d'autres fonctions comme IPROUTE ou la gestion de la qualité de service (QoS).

Ainsi, nous pouvons disposer de toute la puissance de Netfilter pour la sélection de divers types de paquets, et utiliser ensuite ce marquage pour le routage ou les priorités de passage.

Donner ici des exemples précis nous mènerait trop loin parce qu'il faudrait étudier en détail IProute2 et les fonctions de QoS des noyaux 2.4 et la vie est courte.

Voici tout de même un cas de figure qui serait gérable par ce système :

- Nous disposons de deux liens sur le Net :
  - L'un, très rapide et très fiable, mais très cher et facturé au volume de données,
  - L'autre, classique, comme une connexion ADSL, avec les limites que nous leurs connaissons.
- Nous souhaitons exploiter au mieux ces deux connexions, par exemple de la façon suivante :
  - Nous devons mettre à jour le contenu d'un serveur distant. Il faut le faire de façon rapide et sûre. Il n'y a pas forcément beaucoup de données à transmettre, mais il est impératif que ce soit fait le plus rapidement et le plus sûrement possible.
  - Nous devons assurer un accès au Net pour les utilisateurs du réseau local, mais avec une qualité de service plus faible.

Avec le marquage de paquets associé à IProute, nous pourrons arriver à faire passer les mises à jour du serveur sur le lien rapide mais cher et tout le reste sur le lien ADSL.

- Nous disposons d'une connexion ADSL et il arrive très souvent que certains utilisateurs fassent du téléchargement FTP sur des serveurs rapides. Chaque fois qu'un téléchargement est lancé, toute la bande passante Download est utilisée et les autres utilisateurs ne peuvent plus surfer dans de bonnes conditions.
- En exploitant le marquage de paquets associé aux fonctions de QoS, nous pourrons restreindre la bande passante exploitée par le download FTP afin de laisser un peu d'espace pour les autres activités.
- Ceci peut aussi être appliqué aux transferts "peer-to-peer", qui ont l'inconvénient de monopoliser le peu de bande passante upload dont on dispose sur des connexions asymétriques comme ADSL ou câble.

5.4 Pratique

Le script est le suivant :

```
#!/bin/sh
```

```
IPT="/sbin/iptables"
MODPROBE="/sbin/modprobe"

IFACE_EXT="eth0"
#"ppp0"
IFACE_INT="eth1"

PRIVATE_ADDR="192.168.1.254"
PRIVATE_NET="192.168.1.0/255.255.255.0"

test -f $IPT || exit 0
test -f $MODPROBE || exit 0

case "$1" in
start)
     echo -n "Loading firwall's rules: "

        ############################
        # FLUSH TABLES
        ############################
        $IPT -t filter -F
        $IPT -t nat -F
        $IPT -t mangle -F
        ############################
        # MASQUERADING
        ############################
        $IPT -t nat -A POSTROUTING -s $PRIVATE_NET -j MASQUERADE

        ############################
        # FORWARDING RULES
        ############################
        $IPT -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT # ESTABLISHED
        #$IPT -A FORWARD -p UDP -i $IFACE_INT --dport 53 -j ACCEPT  # DOMAIN
        #$IPT -A FORWARD -p TCP -i $IFACE_INT --dport 21 -j ACCEPT  # FTP
        #$IPT -A FORWARD -p TCP -i $IFACE_INT --dport 22 -j ACCEPT  # SSH
        #$IPT -A FORWARD -p TCP -i $IFACE_INT --dport 25 -j ACCEPT  # SMTP
        #$IPT -A FORWARD -p TCP -i $IFACE_INT --dport 80 -j ACCEPT  # HTTP
        #$IPT -A FORWARD -p TCP -i $IFACE_INT --dport 110 -j ACCEPT # POP3
        #$IPT -A FORWARD -p TCP -i $IFACE_INT --dport 443 -j ACCEPT # HTTPS

        $IPT -A FORWARD -p TCP -i $IFACE_EXT --dport 3389 -j ACCEPT # REMOTE DESKTOP
        $IPT -A FORWARD -i $IFACE_INT -s $PRIVATE_NET -j ACCEPT

        $IPT -A FORWARD -j LOG --log-prefix "Forwarding table : "
        $IPT -A FORWARD -j DROP

        ############################
        # INPUT LOOPBACK
        ############################
        $IPT -A INPUT -i lo -j ACCEPT
```

```
    ############################
    # INPUT INTRANET
    ############################
    $IPT -A INPUT -p UDP -i $IFACE_INT --dport 53 -j ACCEPT  # DNS
    $IPT -A INPUT -p UDP -i $IFACE_INT --dport 123 -j ACCEPT # NTP
    $IPT -A INPUT -p UDP -i $IFACE_INT --dport 137 -j ACCEPT # NETBIOS-NS
    $IPT -A INPUT -p UDP -i $IFACE_INT --dport 138 -j ACCEPT # NETBIOS-DGM
    $IPT -A INPUT -p TCP -i $IFACE_INT --dport 139 -j ACCEPT # NETBIOS-SSN

    ############################
    # INPUT INTERNET
    ############################

    ############################
    # INPUT GENERAL
    ############################
    $IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT # ESTABLISHED
    $IPT -A INPUT -p ICMP -j ACCEPT                              # ICMP
    $IPT -A INPUT -p TCP --dport 21 -j ACCEPT                    # FTP
    $IPT -A INPUT -p TCP --dport 22 -j ACCEPT                    # SSH
    $IPT -A INPUT -p TCP --dport 80 -j ACCEPT                    # HTTP
    $IPT -A INPUT -p TCP --dport 113 -j ACCEPT                   # AUTH
    $IPT -A INPUT -p TCP --dport 443 -j ACCEPT                   # HTTPS
    $IPT -A INPUT -p TCP --dport 3000 -j ACCEPT                  # NTOP

    #$IPT -A INPUT -i $IFACE_EXT -j LOG --log-prefix "Input ppp0 : " # LOG..
    #$IPT -A INPUT -i $IFACE_INT -j LOG --log-prefix "Input eth0 : " # LOG..
    $IPT -A INPUT -j DROP                          # DENY ALL

    echo "Done."
    ;;
stop)
    echo -n "Flushing firwall's rules: "

    ############################
    # FLUSH TABLES
    ############################
    $IPT -t filter -F
    $IPT -t nat -F
    $IPT -t mangle -F

    echo "Done."
    ;;
restart)
    /etc/init.d/firewall stop
    /etc/init.d/firewall start
    ;;
status)
    # List tables
```

```
        echo
        echo "---------- FILTER TABLE -----------"
        echo
        $IPT -t filter -L -v
        echo
        echo "----------  NAT TABLE -----------"
        echo
        $IPT -t nat -L -v
        echo
        ;;
    *)
        echo "Usage: /etc/init.d/firewall {start|stop|status}"
        exit 1
        ;;
esac

exit 0
```

Il s'agit d'un script qui permet de modifier facilement les règles mises dans les différentes tables. Il est facilement compréhensible et très configurable.

Je rajouterai une petite procédure afin que le firewall soit activé automatiquement sous ma Debian.

Avant toute chose, il ne faut pas oublier de d'autoriser l'`ip_forward` entre les deux cartes réseaux. Pour cela il suffit d'éditer le fichier `/etc/sysctl.conf` :

```
pport:/etc# more /etc/sysctl.conf
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See sysctl.conf (5) for information.
#
#kernel.domainname = example.com
#net/ipv4/icmp_echo_ignore_broadcasts=1
#config pour faire du routage
net/ipv4/ip_forward=1
pport:/etc#
```

Ensuite voici la marche à suivre, pour avoir un firewall qui fonctionne au démarrage :

1. charger les règles en mémoire à l'aide de notre script `firewall` : `firewall start`
2. sauver ces tables dans le fichier `active` à l'aide le commande `/etc/init.d/iptables save active`
3. effacer les tables à l'aide de `firewall stop`
4. sauver ces tables dans le fichier `inactive` à l'aide le commande `/etc/init.d/iptables save inactive`
5. finalement faire un `dpkg-reconfigure iptables` et choisir de le lancer au démarrage.

Si cela ne fonctionne pas, il y a un autre problème :

Vérifier qu'au runlevel où on lance la machine il y ait bien un lien SXXiptable vers /etc/init.d/iptables. Sinon le faire. (man update-rc.d) Pour cela dans chaque répertoire `/etc/rcX.d/` créer un lien symbolique vers `/etc/init.d/iptables` :

```
ln -s /etc/init.d/iptables S20iptables
```

dans `/etc/rc3.d` pour le démarrage et

```
ln -s /etc/init.d/iptables K20iptables
```

dans `/etc/rc0.d` pour l'arrêt.
oubien utiliser `update-rc.d`

```
update-rc.d iptables start 20 2 3 4 5 . stop 20 0 1 6 .
```

Il est aussi possible que vous ne trouviez pas `iptables` dans `/etc/init.d/`, il faut alors copier votre fichier `firewall` dans le répertoire `/etc/init.d/`, puis à l'aide de la commande `update-rc.d`, on va créer automatiquement les liens dans les répertoires `/etc/rcX.d/` :

```
update-rc.d firewall start 20 2 3 4 5 . stop 20 0 1 6 .
```

Afin de voir les ports que vous avez ouverts, la commande `netstat` est très interessante :
▷ `netstat -an` pour voir les ports,
Pour voir l'états des tables, il suffit de lancer `iptables -L -v`.

5.5 Remarques

Si vous avez configurer votre firewall pour que vos machines du réseau local puissent faire du ftp, il est nécessaire d'ajouter des modules au noyau. En effet lors d'une connexion ftp, les commandes sont envoyées sur un port, mais les réponses arrivent sur un autre port (qui n'est pas autorisé par votre firewall). Il faut donc ajouter les modules `ip_nat_ftp` et `ip_conntrack_ftp`. Ce qui donne :

```
yann@pport:/etc$ more modules
# /etc/modules: kernel modules to load at boot time.
#
# This file should contain the names of kernel modules that are
# to be loaded at boot time, one per line. Comments begin with
# a "#", and everything on the line after them are ignored.
ne
ip_nat_ftp
ip_conntrack_ftp
```

6 Installation du DNS (Domaine Name Server)

6.1 Le serveur Domain Name System.

Votre fournisseur d'accès met à votre disposition un outil pour traduire les noms FQDN en adresses IP. Chez Wanadoo, les DNS fournis aux clients sont généralement 193.252.19.3 et 193.252.19.4. Ce n'est pas une règle absolue, principalement pour les câblés.

Ce Serveur DNS travaille de manière "récursive" . Autrement dit, vous n'avez pas de questions à vous poser (sauf lorsqu'il ne fonctionne plus). Vous (ou votre logiciel) demandez de traduire un nom et vous attendez sagement la réponse, il s'occupe de tout. Le mécanisme est complètement

transparent. Il est mis en route, par exemple, lorsque vous tapez quelque chose comme http://www.google.fr dans la barre d'adresse de votre navigateur.

Comme l'adresse de votre DNS vous est donnée avec le bail que vous accorde le DHCP (ou PPP), vous n'avez généralement même pas besoin de connaître son adresse IP.

Où les choses se compliquent un peu, c'est si vous avez par exemple réalisé la passerelle Linux entre le modem et votre (ou vos) ordinateur(s) personnel(s). Dans ce cas, la machine Linux est bien documentée sur l'adresse du DNS, mais pas votre ou vos postes qui sont, eux, configurés en "dur". Ils ont donc besoin de connaître une adresse de DNS pour pouvoir fonctionner.

Deux solutions sont alors possibles :

- ▷ Vous allez sur votre machine Linux, un petit coup de "nslookup" ou de "host -v" vous indiquera l'adresse du DNS, il ne vous restera plus qu'à la fournir à votre ou à vos postes de travail du réseau privé.
- ▷ Vous prenez votre courage à deux mains, vous lisez attentivement tout ce qui est dit dans ce chapitre et vous construisez sur votre passerelle Linux votre propre serveur DNS :-)

6.2 Mise en place

Il est bien évident que si votre sous-réseau ne comporte que quelques machines, un serveur DNS est un peu comme l'utilisation d'un char pour stopper une mouche. Mais bon, c'est surtout pour la beauté du geste :-).

Il est donc dans ce cas plus simple de passer par le fichier `/etc/hosts`

```
127.0.0.1 localhost
160.60.2.153 pport.lasc.sciences.univ-metz.fr pport

#mettre les hosts sous la forme suivante si l'on utilise pas
#de DNS sur la machine passerelle. Il faut ensuite faire de même
#sur les machines clientes (/etc/hosts et c:\windows\hosts)
 192.168.1.2 pc1.ecole-belan.org pc1
 192.168.1.3 pc2.ecole-belan.org pc2
 192.168.1.4 pc3.ecole-belan.org pc3
 192.168.1.5 pc4.ecole-belan.org pc4
 192.168.1.6 pc5.ecole-belan.org pc5
 192.168.1.7 pc6.ecole-belan.org pc6
 192.168.1.8 pc7.ecole-belan.org pc7
 192.168.1.9 pc8.ecole-belan.org pc8

# The following lines are desirable for IPv6 capable hosts
# (added automatically by netbase upgrade)

::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

Allons y pour le DNS. Il faut tout d'abord installer un serveur de Noms. Je vous propose `bind9` en standard sur la Debian. Toujours le même `apt-get install bind9` pour l'installation.

Il faut ensuite configurer notre DNS par les fichiers qui se trouvent dans le répertoire `/etc/bind`.

Notre réseau local sera composé de quelques machine dont le nom de domaine sera `ecole-belan.org`. La passerelle vers internet et le serveur DNS sera notre machine actuelle. Il convient alors de modifier le fichier `/etc/resolv.conf` afin de ne plus pointer vers un DNS externe, mais sur la machine locale, c'est à dire `127.0.0.1`. On fixe par la même occasion le domaine `ecole-belan.org`. Ceci nous donne : Fichier `/etc/resolv.conf`

```
search ecole-belan.org
nameserver 127.0.0.1
```

Il convient ensuite de configurer le fichier `/etc/bind/named.conf`, qui va permettre de définir le domaine de notre nouveau réseau. Il est assez facile de comprendre les différentes parties de ce fichier.

Ce fichier est le premier que BIND va lire. Nous y trouvons déjà des informations intéressantes...

```
options {
directory "/var/cache/bind";

// If there is a firewall between you and nameservers you want
// to talk to, you might need to uncomment the query-source
// directive below. Previous versions of BIND always asked
// questions using port 53, but BIND 8.1 and later use an unprivileged
// port by default.

// query-source address * port 53;

// If your ISP provided one or more IP addresses for stable
// nameservers, you probably want to use them as forwarders.
// Uncomment the following block, and insert the addresses replacing
// the all-0's placeholder.
/*
allow-transfer {
192.168.1.0/24;
};
allow-query {
192.168.1.0/24;
};
*/
forwarders {
197.230.169.1; //dns de la fac
};

// forwarders {
//  0.0.0.0;
// };

auth-nxdomain no;  # conform to RFC1035

};
```

Au paragraphe "options", on trouve :

▷ L'emplacement des autres fichiers de configuration. Ici, ce sera le répertoire `/var/cache/bind`.
▷ Notify no indique que ce serveur travaillera pour son compte. Cette option est utile lorsque plusieurs DNS doivent se synchroniser entre eux. Ce ne sera pas notre cas.
▷ La variable `forwarder` qui permet de transferer les demande DNS vers le DNS de la fac
▷ Les variables `allow-transfer` et `allow-query` qui permettent de sécuriser votre domaine en autorisant qu'un plage de machine à se connecter au DNS

```
zone "." {
type hint;
file "/etc/bind/db.root";
};
```

La zone "." est fondamentale pour ce que l'on veut réaliser. C'est elle qui permettra à BIND d'interroger les root-servers. Le fichier root.cache dont on a déjà parlé contient les adresses de ces root servers. Il doit être périodiquement mis à jour, nous verrons çà plus loin.

```
zone "127.in-addr.arpa" {
type master;
file "/etc/bind/db.127";
};
```

Cette zone est une zone de recherche inverse pour les adresses de type 127.0.0. Ces adresses sont utilisées exclusivement en interne par la pile TCP/IP et correspondent à l'hôte "localhost"

Maintenant nous allons ajouter un zone d'autorité. C'est un domaine ou un sous-domaine que le DNS sait traiter avec sa propre base de données. Nous pouvons en créer une pour résoudre les noms des hôtes de notre réseau privé. Ceci ne présente, encore une fois, pas un intérêt capital mais tant qu'on y est, pourquoi nous en priver ? (surtout si vous montez par exemple un petit serveur apache pour réaliser votre intranet :-) Préparation du travail.

Vous avez en tête, bien entendu, l'adresse de chacun de vos postes privés ainsi que leur nom "NetBIOS", si ces postes sont sous Windows. Par ailleurs, vous connaissez également l'adresse et le nom de votre passerelle Linux.

Fixons les idées par un exemple :
▷ Quatre hôtes Windows :

```
pc1  192.168.1.1
pc2  192.168.1.2
pc3  192.168.1.3
pc4  192.168.1.4
```

▷ La passerelle Linux qui supporte le DNS. Ce poste dispose de deux interfaces réseau, nous nous intéressons ici à celle qui est connectée au réseau privé. (L'autre recevant une adresse IP dynamique par le DNS de Wanadoo Câble).

```
pc  192.168.1.254
```

Nous allons nous choisir un joli nom de domaine : `ecole-belan.org`

```
zone "ecole-belan.org" {
notify no;
type master;
file "/etc/bind/db.ecole-belan.org";
};

zone "1.168.192.in-addr.arpa" {
```

```
notify no;
type master;
file "/etc/bind/db.192.168.1";
};
```

Ces deux zones ont été ajoutées, ce qui veut dire que l'on va trouver deux nouveaux fichiers dans le répertoire `/etc/bind/` :
▷ db.ecole-belan.org pour la résolution des noms
▷ db.192.168.1 pour la résolution inverse
Voici donc le fichier final :

Fichier `/etc/bind/named.conf`

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//

options {
directory "/var/cache/bind";

// If there is a firewall between you and nameservers you want
// to talk to, you might need to uncomment the query-source
// directive below. Previous versions of BIND always asked
// questions using port 53, but BIND 8.1 and later use an unprivileged
// port by default.

// query-source address * port 53;

// If your ISP provided one or more IP addresses for stable
// nameservers, you probably want to use them as forwarders.
// Uncomment the following block, and insert the addresses replacing
// the all-0's placeholder.
/*
allow-transfer {
192.168.1.0/24;
};
allow-query {
192.168.1.0/24;
};
*/
forwarders {
197.230.169.1;
};

// forwarders {
//  0.0.0.0;
// };
```

```
auth-nxdomain no;  # conform to RFC1035

};

// prime the server with knowledge of the root servers
zone "." {
type hint;
file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
type master;
file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
type master;
file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
type master;
file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
type master;
file "/etc/bind/db.255";
};

// add entries for other zones below here

zone "ecole-belan.org" {
notify no;
type master;
file "/etc/bind/db.ecole-belan.org";
};

zone "1.168.192.in-addr.arpa" {
notify no;
type master;
file "/etc/bind/db.192.168.1";
};
```

Fichier `/etc/bind/db.0`

```
;
; BIND reverse data file for broadcast zone
```

```
;
$TTL 604800
@ IN SOA localhost. root.localhost. (
      1 ; Serial
 604800 ; Refresh
  86400 ; Retry
2419200 ; Expire
 604800 ) ; Negative Cache TTL
;
@ IN NS localhost.
```

Fichier `/etc/bind/db.127`

```
;
; BIND reverse data file for local loopback interface
;
$TTL 604800
@ IN SOA localhost. root.localhost. (
      1 ; Serial
 604800 ; Refresh
  86400 ; Retry
2419200 ; Expire
 604800 ) ; Negative Cache TTL
;
@ IN NS localhost.
1.0.0 IN PTR localhost.
```

Le signifie qu'il est fait référence au serveur lui-même.
Notez le Start Of Authority (SOA) et le CNAME www de server1
Fichier `/etc/bind/db.192.168.1`

```
$TTL 86400
@ IN SOA pc.ecole-belan.org. root.ecole-belan.org. (
2000042702; serial
0; refresh
0; retry
0; expire
0; default TTL
)
@ IN NS pc.ecole-belan.org.
254 IN PTR pc.ecole-belan.org.

1 IN PTR pc1.ecole-belan.org.
2 IN PTR pc2.ecole-belan.org.
3 IN PTR pc3.ecole-belan.org.
4 IN PTR pc4.ecole-belan.org.
5 IN PTR pc5.ecole-belan.org.
6 IN PTR pc6.ecole-belan.org.
7 IN PTR pc7.ecole-belan.org.
8 IN PTR pc8.ecole-belan.org.
```

Pas grand chose à dire, de plus, si ce n'est que seul le dernier octet de l'adresse est signalé, c'est normal, nous sommes dans une classe C, les trois autres octets sont ceux du réseau.

Fichier `/etc/bind/db.ecole-belan.org`

```
$TTL 86400
@ IN SOA  pc.ecole-belan.org. root.ecole-belan.org.(
2000042701; serial
3600; refresh
900; retry
1209600; expire
43200; default TTL
)
@ IN NS pc.ecole-belan.org
pc IN A 192.168.1.254
pc IN HINFO "Pentium 133Mhz" "Debian Woody 2.4.18fb"

pc1 IN A 192.168.1.1
pc2 IN A 192.168.1.2
pc3 IN A 192.168.1.3
pc4 IN A 192.168.1.4
pc5 IN A 192.168.1.5
pc6 IN A 192.168.1.6
pc7 IN A 192.168.1.7
pc8 IN A 192.168.1.8
```

Fichier `/etc/bind/db.local`

```
$TTL 86400
@ IN SOA  pc.ecole-belan.org. root.ecole-belan.org.(
2000042701; serial
3600; refresh
900; retry
1209600; expire
43200; default TTL
)
@ IN NS pc.ecole-belan.org
pc IN A 192.168.1.254
pc IN HINFO "Pentium 133Mhz" "Debian Woody 2.4.18fb"

pc1 IN A 192.168.1.1
pc2 IN A 192.168.1.2
pc3 IN A 192.168.1.3
pc4 IN A 192.168.1.4
pc5 IN A 192.168.1.5
pc6 IN A 192.168.1.6
pc7 IN A 192.168.1.7
pc8 IN A 192.168.1.8
```

Fichier `/etc/bind/db.root`

```
;       This file holds the information on root name servers needed to
```

```
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC registration services
;       under anonymous FTP as
;           file                /domain/named.root
;           on server           FTP.RS.INTERNIC.NET
;       -OR- under Gopher at    RS.INTERNIC.NET
;           under menu          InterNIC Registration Services (NSI)
;              submenu          InterNIC Registration Archives
;           file                named.root
;
;       last update:    Aug 22, 1997
;       related version of root zone:   1997082200
;
;
; formerly NS.INTERNIC.NET
;
.                       3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000      A     198.41.0.4
;
; formerly NS1.ISI.EDU
;
.                       3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.     3600000      A     128.9.0.107
;
; formerly C.PSI.NET
;
.                       3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.     3600000      A     192.33.4.12
;
; formerly TERP.UMD.EDU
;
.                       3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.     3600000      A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.                       3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.     3600000      A     192.203.230.10
;
; formerly NS.ISC.ORG
;
.                       3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.     3600000      A     192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
.                       3600000      NS    G.ROOT-SERVERS.NET.
```

```
G.ROOT-SERVERS.NET.      3600000      A     192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.                        3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.      3600000      A     128.63.2.53
;
; formerly NIC.NORDU.NET
;
.                        3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.      3600000      A     192.36.148.17
;
; temporarily housed at NSI (InterNIC)
;
.                        3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.      3600000      A     198.41.0.10
;
; housed in LINX, operated by RIPE NCC
;
.                        3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.      3600000      A     193.0.14.129
;
; temporarily housed at ISI (IANA)
;
.                        3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.      3600000      A     198.32.64.12
;
; housed in Japan, operated by WIDE
;
.                        3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.      3600000      A     202.12.27.33
; End of File
```

Il faur alors relancer le serveur de noms par :`/etc/init.d/bind9 restart`.

Maintenant il est possible de tester le bon fonctionnement du réseau. A partir d'un client il suffit de faire un `ping pc` pour essayer de communiquer avec le serveur. Si le `pc` répond c'est que tout fonctionne bien.

7 Installation d'un modem standard

Cette section concerne l'installation d'un modem standard en vue de faire passer notre passerelle et DNS en une machine de partage de connexion internet dans le cadre d'une utilisation personnelle ou un petit réseau de machine.

Comme toujours, le matérielle étant de récupération, il séagit d'un modem 28800 Kbits branché sur le port série de la machine.

Nous allon utiliser le paquet `pppconfig` pour la configuration de notre modem et de la connection.

La suite est tirée de http://www.via.ecp.fr/~alexis/formation-linux/internet.html

Cette section explique comment se connecter à Internet avec un modem classique branché sur une ligne téléphonique classique. La procédure ci-dessous doit marcher sans problèmes avec un modem externe branché sur port série, ou avec un modem PCMCIA ; par contre, pour les modems PCI ou les modems intégrés, la procédure est différente et dépend de chaque modem...

7.1 Modem PCMCIA

Vérifiez que le package pcmcia-cs est bien installé (si vous avez bien suivi mes consignes pour la procédure d'installation, il doit l'être). Avec la commande suivante, il installe le package s'il n'est pas installé, et, dans le cas contraire, t'informe qu'il est déjà installé.

```
# apt-get install pcmcia-cs
```

7.2 Modem externe sur port série

Regardez sur quel port série le modem est branché :
▷ s'il est connecté sur le port série COM1, le device correspondant sera /dev/ttyS0 ;
▷ s'il est connecté sur le port série COM2, le device correspondant sera /dev/ttyS1.
Créez un lien symbolique `/dev/modem` pointant vers le bon périphérique ; par exemple, s'il est branché sur le port COM1, tapez :

```
# cd /dev
# ln -s ttyS0 modem
```

7.3 Vérification du modem

Si c'est un modem PCMCIA, insérez le carte dans votre portable ; si c'est un modem externe, allumez-le. Vous allez maintenant vérifier que le système a bien reconnu le modem :

```
# setserial /dev/modem
/dev/modem, UART: 16550A, Port: 0x03e8, IRQ: 0
```

▷ Si la ligne qui s'affiche contient UART : 16550A, alors cela signifie que le modem est bien reconnu.
▷ Si, par contre, la ligne qui s'affiche contient UART : unknown, alors cela signifie que le modem n'est pas reconnu (et là je ne sais pas trop ce qu'on peut faire...).

7.4 Configuration de la connexion vers le fournisseur d'accès

Le plus simple pour configurer la connexion vers votre fournisseur d'accès est d'utiliser l'assistant qui est installé par défaut :

```
# pppconfig
```

Sélectionnez Create - Create a connection et répondez aux questions successives :

1. Provider Name : rentrez un nom pour cette connexion (par exemple le nom de votre fournisseur d'accès Internet) ;
2. Configure Nameservers (DNS) : sélectionnez Use dynamic DNS pour obtenir automatiquement les adresses des serveurs DNS de votre fournisseur d'accès à chaque connexion ;

3. Authentication Method : sélectionnez PAP Peer Authentication Protocol [FIXME : je ne sais pas dans quel cas il faut sélectionner Chat] ;
4. User Name : tapez le login qui vous a été attribué par votre fournisseur d'accès (tapez-le entre guillements si le login contient des caractères de ponctuation) ;
5. Password : tapez le mot de passe qui vous a été donné par votre fournisseur d'accès (tapez-le entre guillements si le mot de passe contient des caractères de ponctuation) ;
6. Speed : laissez la valeur 115200 qui est présente par défaut ;
7. Pulse or Tone : si votre ligne téléphonique fonctionne à fréquences vocales (ce qui est le cas presque partout en France), sélectionnez Tone ; si votre ligne fonctionne avec les impulsions, sélectionnez Pulse ;
8. Phone Number : rentrez le numéro de téléphone de votre fournisseur d'accès ;
9. Choose Modem Config Method : répondez No ;
10. Manually Select Modem Port : tapez `/dev/modem`, qui est le lien symbolique qui pointe vers le bon périphérique ;
11. Properties of nom_de_la_connexion : si vous pensez avoir bien répondu à toutes les questions, sélectionnez Finished - Write files and return to main menu et OK ;
12. Main Menu : sélectionnez Quit - Exit this utility.

Pour créer une deuxième connexion, changer une connexion existante ou supprimer une connexion, relancez cet assistant et laissez-vous guider par les boites de dialogues (qui ne sont malheureusement pas encore traduites). Se connecter

Pour se connecter au fournisseur d'accès, c'est très simple :

```
# pon nom_de_la_connexion
```

où nom_de_la_connexion est le nom que vous aviez entré à la première question de l'assistant.

Vous devez normalement entendre le modem se connecter. Pour suivre l'établissement de la connexion, tapez :

```
# plog -f
```

Dès que vous voyez une ligne du genre :

```
Dec 27 19:42:54 alpy pppd[1825]: Script /etc/ppp/ip-up started (pid 1843)
```

cela signifie que la connexion est établie. Vous pouvez alors arrêter l'affichage des messages (encore appelés logs) par la combinaison de touches `Ctrl-c`.

Pour se déconnecter :

```
# poff nom_de_la_connexion
```

Pour permettre à un simple utilisateur de se connecter et se déconnecter, il faut le rajouter aux groupes dialout et dip ; et pour lui permettre d'utiliser la commande plog, il faut le rajouter au groupe adm :

```
# adduser toto dialout
# adduser toto dip
# adduser toto adm
```

où toto est le nom de l'utilisateur à qui vous voulez rajouter les droits. Il pourra alors lancer lui-même les commandes pon, poff et plog.

8 Installation d'un partage de connextion à l'aide d'un routeur/firewall

Vous n'avez pas forcément une machine sous la main pour la transformer en routeur/firewall et vous ne désirez pas avoir une alimentation de 300W qui tourne 24h/24h. La suite est faite pour vous.

La situation est la suivante : vous avez une connexion ADSL opérationnelle, c'est à dire que votre modem se synchronise bien. En bon linuxien, vous avez achetez un modem ethernet pour être sur de la compatibilité avec votre OS préféré. Je prendrai l'exemple du speedtouch ethernet avec lequel je n'ai eu aucun soucis pour l'instant.

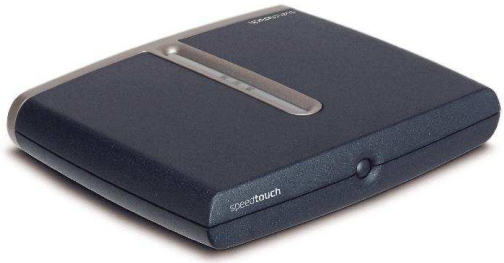

Fig. 4 – Modem ethernet speedtouch

Vous avez acheter au meilleur prix un routeur câble/ADSL, par exemple le routeur firewall ovislink IP-3047, et si vous avez plus de 4 machines, un petit switch 8 ports de la même marque evo-fsh8r et au meilleur prix lui aussi. L'ensemble ne devrait pas dépasser les 50 €. Vous pouvez aussi choisir le tout intégré, modem routeur firewall et switch si vous n'avez pas eu de modem avec votre abonnement.

Fig. 5 – Routeur/Firewall et Switch Ovislink

8.1 Installation et câblage

Il faut donc commencer par brancher convenablement le modem et le routeur/switch. Pour cela vous reporter sur le schéma suivant :

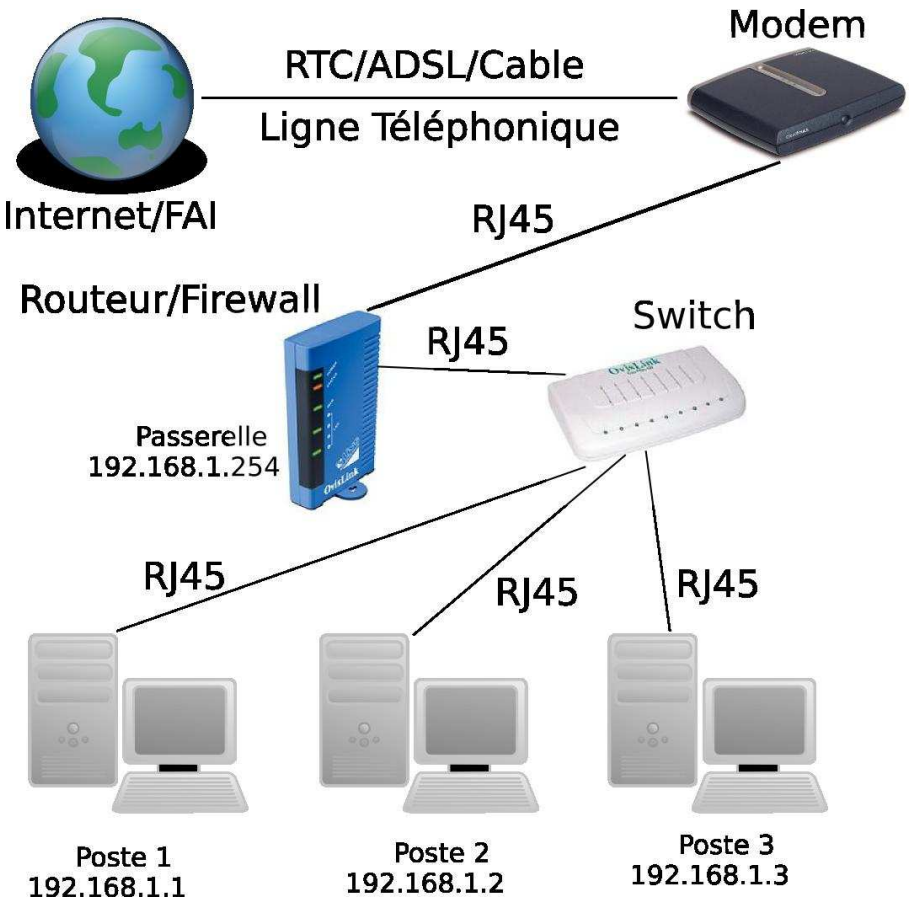

FIG. 6 – Montage du réseau

Vous obtenez donc un petit réseau chez vous qui se trouve derrière le routeur/firewall.

Le Routeur Internet Adsl/Câble vous permet de partager votre accès internet à tous les utilisateurs de votre réseau. Il intègre aussi des fonctions d'administration telles que le serveur DHCP, le Routage, la Restriction d'accès. Il est configurable par le protocole TELNET ou bien par votre navigateur internet. Vous avez à votre disposition un firewall intégré, la possibilité de créer des serveurs virtuels, rendant vos serveurs accessible aux internautes (HTTP, FTP).

Caractéristiques techniques / Spécifications :

- ▷ Routeur Internet pour modem Adsl/Câble 1 port WAN + switch 4 ports 10/100 Mbps Auto-MDI/MDIX
- ▷ Protocoles : IP, NAT, ARP, ICMP, DHCP client/server, PPPoE , PPP, PAP, CHAP
- ▷ Serveur DHCP pouvant gérer jusque 128 postes clients
- ▷ Client DHCP sur le portWAN récupérant automatiquement les paramètres de votre FAI
- ▷ Proxy DNS
- ▷ Firewall intégré permettant de sécuriser votre réseau
- ▷ Configuration : TELNET, Serveur HTTP intégré, Programme GUI pour environnement Windows

8.2 Configuration des paramètres de connexion

Votre machine est connectée au switch, ou directement au routeur, il va falloir configurer votre connexion à l'aide des données fournies par votre FAI. En effet c'est le routeur qui va se connecter directement à internet par l'intermédaire du modem dès sa mise sous tension.

Par défaut l'adresse du routeur est `192.168.1.254`. Il faut donc placer votre machine dans ce

sous-réseau, si vous avez une adresse IP fixe, ou la configurer en DHCP si vous désirez utiliser le serveur DHCP du routeur.

Pour cela vous devez éditer le fichier `/etc/network/interfaces`. Si vous utilisez une adresse IP fixe modifiez le ainsi :

```
auto lo
iface lo inet loopback

iface eth0 inet static
address 192.168.1.1
netmask 255.255.255.0
broadcast 192.168.1.255
network 192.168.1.0
gateway 192.168.1.254
```

Si vous utilisez une adresse DHCP modifiez le ainsi :

```
auto lo
iface lo inet loopback

# This entry was created during the Debian installation
auto eth0
iface eth0 inet dhcp
```

Ensuite redémarrer le réseau par la commande :

```
/etc/init.d/networking restart
```

Un petit `ifconfig` pour vérifier que tout s'est bien passé :

```
yann@tuxpowered:~/$/sbin/ifconfig
eth0      Lien encap:Ethernet  HWaddr 00:50:DA:E3:32:F9
          inet adr:192.168.1.1  Bcast:255.255.255.255  Masque:255.255.255.0
          adr inet6: fe80::250:daff:fee3:32f9/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:41213 errors:0 dropped:0 overruns:0 frame:0
          TX packets:46188 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:52017793 (49.6 MiB)  TX bytes:3701560 (3.5 MiB)
          Interruption:5 Adresse de base:0xe400

lo        Lien encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:224665 errors:0 dropped:0 overruns:0 frame:0
          TX packets:224665 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:39096708 (37.2 MiB)  TX bytes:39096708 (37.2 MiB)
```

vous pouvez tester la connexion à l'aide de la commande `ping` :

```
yann@tuxpowered:~/$ ping -c 5 192.168.1.254
PING 192.168.1.254 (192.168.1.254) 56(84) bytes of data.
64 bytes from 192.168.1.254: icmp_seq=1 ttl=100 time=0.201 ms
64 bytes from 192.168.1.254: icmp_seq=2 ttl=100 time=0.171 ms
64 bytes from 192.168.1.254: icmp_seq=3 ttl=100 time=0.177 ms
64 bytes from 192.168.1.254: icmp_seq=4 ttl=100 time=0.170 ms
64 bytes from 192.168.1.254: icmp_seq=5 ttl=100 time=0.203 ms

--- 192.168.1.254 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.170/0.184/0.203/0.019 ms
yann@tuxpowered:~/$
```

Ensuite vous démarrez votre navigateur préféré et connectez-vous à l'adresse suivante `http://192.168.1.2` par défaut il n'y a ni compte ni de mot de passe et vous obtenez la page suivante :

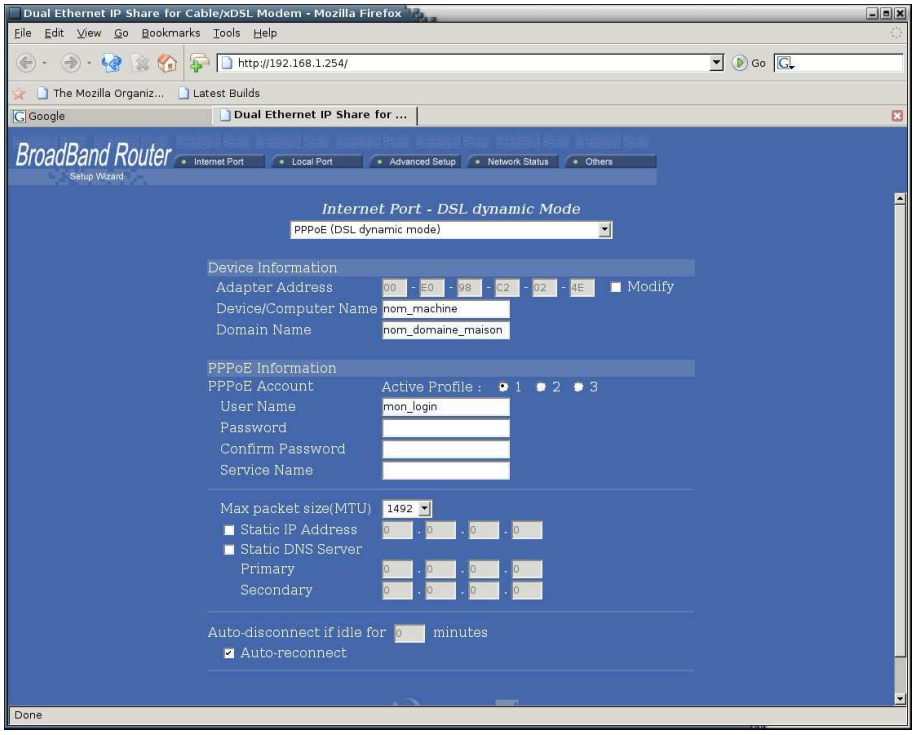

FIG. 7 – Configuration routeur port internet

Dans l'onglet `internet Port`, il suffit de choisir le type de configuration de connexion (elle dépend de votre fournisseur d'accès, si vous avez une adresse IP fixe ou non), d'entrer les données fournies par votre fournisseur d'accès (login et mot de passe), cochez la case `auto-reconnect`, puis sauvez la configuration par le bouton `save`.

Il est possible de démarrer manuellement la connexion par l'intermédiaire de l'onglet `Network Status -> Connection Status` puis du bouton `Renew/Connect`

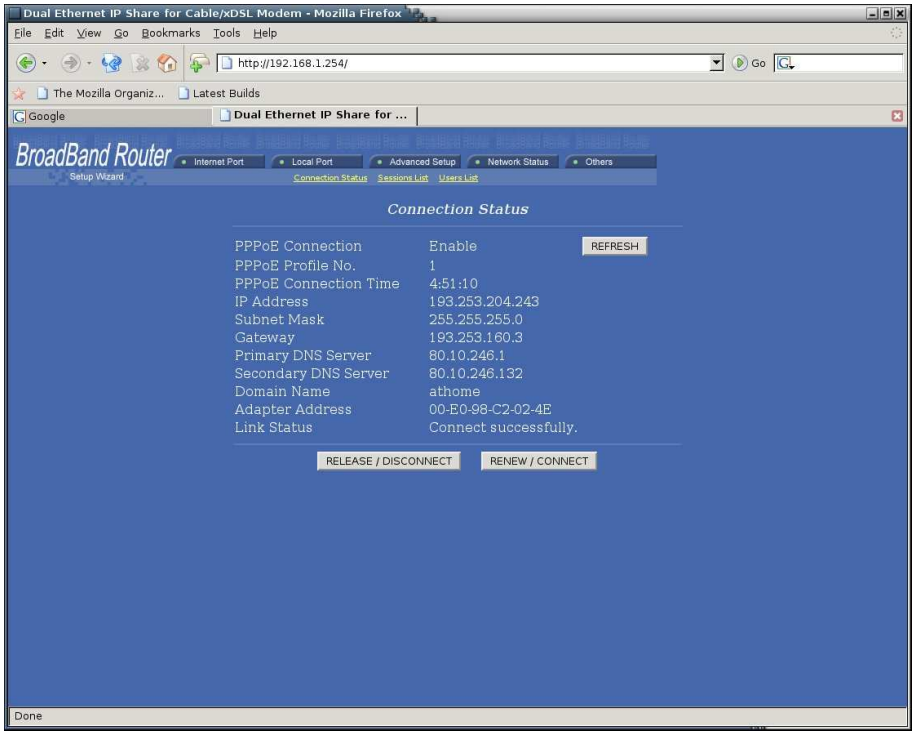

FIG. 8 – Connexion manuel à internet

Dans l'onglet `Local Port`, vous pouvez changer l'adresse IP de votre routeur si par exemple vous avez plusieurs sous réseaux chez vous. C'est dans cet onglet que vous pouvez activer ou non le serveur DHCP du routeur.

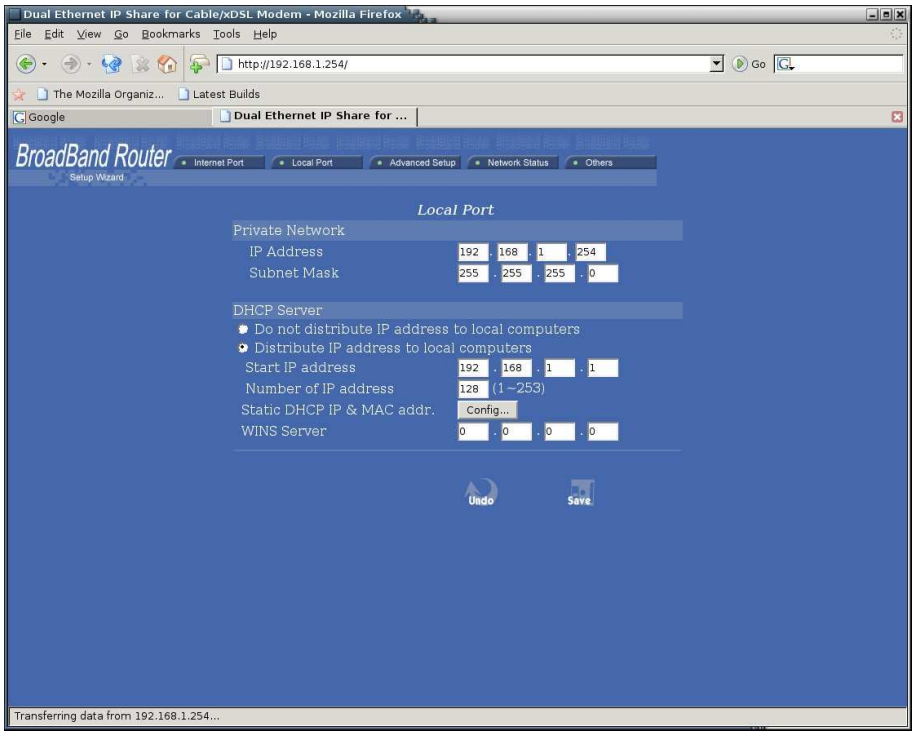

FIG. 9 – Configuration routeur port local

Il est aussi possible de fixer les adresses IP délivrées par le routeur en fonction des adresses MAC de vos composants réseaux. En effet cela peut être très pratiques si vous voulez donner

des noms de machine dans un fichier `/etc/hosts` ou encore accéder à une imprimante réseau sans devoir reconfigurer vos postes clients à chaque démarrage de l'imprimante.

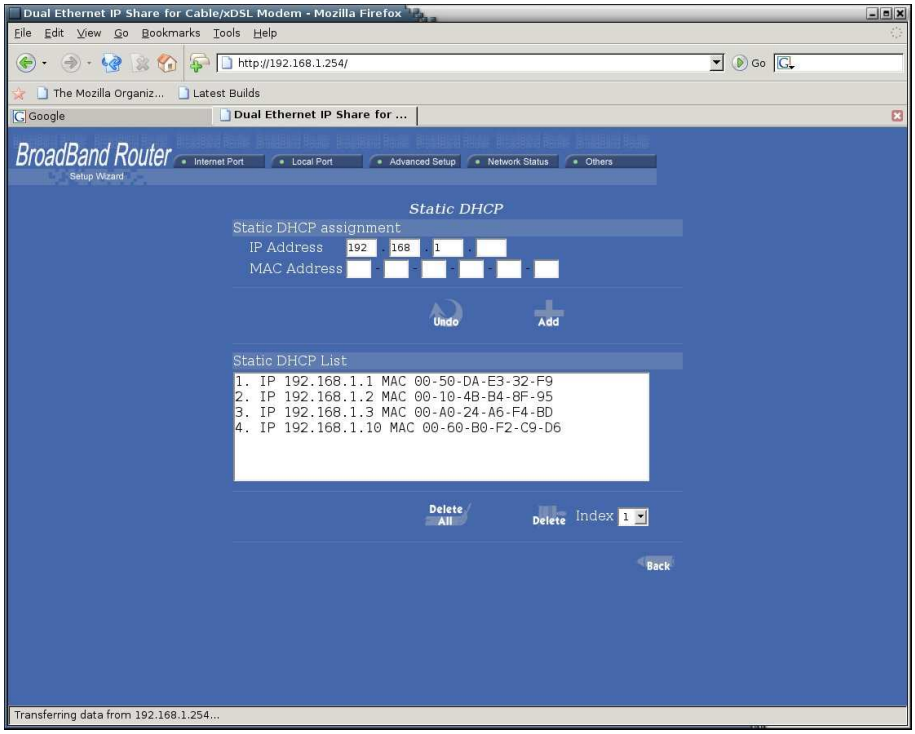

FIG. 10 – Configuration IP statique du DHCP

Le routeur comporte d'autres configurations interessantes que je ne détaillerai pas ici.

Maintenant il ne vous reste plus qu'a configurer les machines clientes en DHCP sur votre réseau privé.

Ensuite il ne reste qu'a tester votre partage de connexion par

```
yann@tuxpowered:~/$ ping -c 5 www.google.fr
PING www.google.akadns.net (66.102.9.104) 56(84) bytes of data.
64 bytes from 66.102.9.104: icmp_seq=1 ttl=243 time=63.2 ms
64 bytes from 66.102.9.104: icmp_seq=2 ttl=243 time=61.1 ms
64 bytes from 66.102.9.104: icmp_seq=3 ttl=243 time=61.6 ms
64 bytes from 66.102.9.104: icmp_seq=4 ttl=243 time=62.7 ms
64 bytes from 66.102.9.104: icmp_seq=5 ttl=243 time=63.5 ms

--- www.google.akadns.net ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4022ms
rtt min/avg/max/mdev = 61.169/62.483/63.569/0.937 ms
yann@tuxpowered:~/$
```

C'en est fini de cet article, toutes remarques et corrections sont les bienvenues à l'adresse morere@sciences.univ-metz.fr