Maîtriser son réseau local

- Objet : du tuto Fix Me!
- Niveau requis : Fix Me!
  débutant, avisé
- Commentaires : *Contexte d'utilisation du sujet du tuto.* Fix Me!
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.

Mise en place d'un routeur Debian

Configuration physique

```
BOX_adsl <---> (192.168.0.1:eth0) _ROUTEUR-DEBIAN_(eth1:192.168.1.1)
 |                                     |
sous-réseau A                       sous-réseau B
 |                                     |
 |_CLIENT-A1 (eth0:192.168.0.10)       |_CLIENT-B1 (eth0:192.168.1.3)
 |_CLIENT-AX (eth0:192.168.0.xx)       |_CLIENT-BX (eth0:192.168.1.x)
```

On pourra faire communiquer les clients du sous-réseau A avec ceux du sous-réseau B.
Donner un accès à internet aux clients au sous-réseau B.
Maîtriser les connexions internet des clients du réseau B. Par exemple en vue du contrôle parentale pour ce qui concerne le temps de connexion des clients.
Mettre en place le proxy squid, afin d'exclure les sites dangereux pour les enfants.
Mais aussi se servir de squid pour faire du cache, afin d'améliorer le débit des vidéos de kirbi...

Mais encore apprendre à utiliser iptables et la mise en place d'un pare-feu pour le réseau B (tests sur la connexion entre le réseau A et B) sans prendre trop de risque puisque on restera derrière le pare-feu du routeur de la box-adsl.

Installation de la passerelle debian

Cela fait d'un ordinateur un routeur.

```
vim /etc/network/interfaces
```

```
auto lo

# The loopback network interface
iface lo inet loopback
#carte vers la box
auto eth0
iface eth0 inet static
address 192.168.0.1
network 192.168.0.0
netmask 255.255.255.0
```

```
gateway 192.168.0.254
post-up iptables-restore < /etc/iptables.save # (1) ajout voir plus bas

#carte vers le LAN
auto eth1
iface eth1 inet static
address 192.168.1.1
network 192.168.1.0
netmask 255.255.255.0
```

- Puis modification du fichier /etc/sysctl.conf pour dé-commenter :

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

- Prise en compte des modifications :

```
sysctl -p
```

Mise en place du protocole NAT sur eth0

C'est-à-dire l'interface tournée vers la box-machin, c'est-à-dire vers web.

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

- Sauvegarde de cette modification :

```
iptables-save > /etc/iptables.save
```

Édition de /etc/network/interfaces

(1) Ajout de la ligne suivante :

```
post-up iptables-restore < /etc/iptables.save
```

Côté client

```
vim /etc/network/interfaces
```

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.3
network 192.168.1.0
```

```
netmask 255.255.255.0
gateway 192.168.1.1
```

Prise en compte des modifications :

```
/etc/init.d/networking start
```

Là NetworkManager ne signale plus qu'il n'y a pas de connexion, mais depuis le navigateur impossible d'afficher quoique ce soit, et de même les ping sur le réseau 192.168.0.0 échouent.

- Enfin côté client édition de /etc/resolv.conf

pour ajouter “nameserveur 127.0.0.1” ainsi que les DNS des forwarders.

On ré-active :

```
/etc/init.d/networking start
```

Et là ça marche !

Il faut maintenant installer un pare-feu sur le routeur debian.

Maîtriser le sous-réseau avec iptables

Maîtriser le sous-réseau avec squid

Installation et configuration DHCP

Pour en savoir un peu plus sur le fonctionnement du protocole DHCP : http://www.frameip.com/dhcp/

> Remarque Bind9 étant installé sur l'IP eth0 du système qui fait office de routeur et sur lequel il va être installé isc-dhcp-server sur eth1, il n'est pas sans intérêt d'ajouter dans le fichier /etc/resolv.conf la ligne **`nameserver 127.0.0.1`**, afin que le DNS fasse office de DNS pour ce système lui-même.
>
> Dans /etc/network/interfaces, il faut ajouter pour eth0 cette ligne :

En dessous de la ligne avec **`network`** :

```
dns-nameservers ip-du-routeur
```

Installation

```
apt-get install isc-dhcp-server
```

Il n'est pas content !

```
Traitement des actions différées (« triggers ») pour « man-db »...
Paramétrage de isc-dhcp-server (4.2.2.dfsg.1-5+deb70u6) ...
Generating /etc/default/isc-dhcp-server...
[FAIL] Starting ISC DHCP server: dhcpd[....] check syslog for diagnostics.
... failed!
 failed!
invoke-rc.d: initscript isc-dhcp-server, action "start" failed.
```

> Vérification des logs:

```
cat /var/log/syslog
```

```
Oct  3 10:48:59 debian-serveur dhcpd: Wrote 0 leases to leases file.
Oct  3 10:48:59 debian-serveur dhcpd:
Oct  3 10:48:59 debian-serveur dhcpd: No subnet declaration for eth0
(192.168.0.1).
Oct  3 10:48:59 debian-serveur dhcpd: ** Ignoring requests on eth0.  If this
is not what
Oct  3 10:48:59 debian-serveur dhcpd:    you want, please write a subnet
declaration
Oct  3 10:48:59 debian-serveur dhcpd:    in your dhcpd.conf file for the
network segment
Oct  3 10:48:59 debian-serveur dhcpd:    to which interface eth0 is
attached. **
Oct  3 10:48:59 debian-serveur dhcpd:
Oct  3 10:48:59 debian-serveur dhcpd:
Oct  3 10:48:59 debian-serveur dhcpd: No subnet declaration for eth1
(192.168.1.1).
Oct  3 10:48:59 debian-serveur dhcpd: ** Ignoring requests on eth1.  If this
is not what
Oct  3 10:48:59 debian-serveur dhcpd:    you want, please write a subnet
declaration
Oct  3 10:48:59 debian-serveur dhcpd:    in your dhcpd.conf file for the
network segment
Oct  3 10:48:59 debian-serveur dhcpd:    to which interface eth1 is
attached. **
Oct  3 10:48:59 debian-serveur dhcpd:
Oct  3 10:48:59 debian-serveur dhcpd:
Oct  3 10:48:59 debian-serveur dhcpd: Not configured to listen on any
interfaces!
Oct  3 11:08:59 debian-serveur -- MARK --
Oct  3 11:09:01 debian-serveur /USR/SBIN/CRON[4234]: (root) CMD (  [ -x
/usr/lib/php5/maxlifetime ] && [ -x /usr/lib/php5/sessionclean ] && [ -d
/var/lib/php5 ] && /usr/lib/php5/sessionclean /var/lib/php5
$(/usr/lib/php5/maxlifetime))
Oct  3 11:17:01 debian-serveur /USR/SBIN/CRON[4255]: (root) CMD (   cd / &&
run-parts --report /etc/cron.hourly)
```

> Cela est normal quand le serveur peut être connecté à plusieurs sous-réseaux. Pour démarrer le

> serveur DHCP il faut définir un unique sous-réseau qu'il devra écouter.
> Il faut donc éditer le fichier /etc/dhcp/dhcpd.conf pour lui indiquer

Édition de /etc/dhcp/dhcpd.conf

(Après s'être fait une sauvegarde !)

- Une petite vérification :

```
less /etc/default/dhcp3-server
```

```
INTERFACE= "eth1"
```

```
vim /etc/dhcp/dhcpd.conf
```

On ajoute tout à la fin :

```
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.2 192.168.1.50;
option domain-name-servers 192.168.1.1;
option domain-name "mondomaine.hyp";
option netbios-name-servers 192.168.1.1;
option routers 192.168.1.1;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
default-lease-time 86400;
max-lease-time 676800;
}
```

- Puis on redémarre :

```
/etc/init.d/isc-dhcp-server start
```

```
[ ok ] Starting ISC DHCP server: dhcpd.
```

Côté client

- Avant l'installation du DHCP sur la passerelle debian :

```
less /etc/network/interfaces
```

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.3
network 192.168.1.0
netmask 255.255.255.0
```

```
gateway 192.168.1.1
```

- Pour laisser le serveur DHCP attribuer une IP à ce client :

On édite côté client /etc/network/interfaces :

```
vim /etc/network/interfaces
```

```
auto lo
iface lo inet loopback
```

- On redémarre le système

Après le redémarrage...

```
ifconfig | grep 192
```

```
inet adr:192.168.1.2 Bcast:192.168.1.255 Masque:255.255.255.0
```

Le premier ordinateur allumé du sous-réseau en 192.168.1.* se voit attribué la première adresse de la plage d'adresse (“range”) entre 162.168.1.2 et 192.168.1.50.
Donc tout fonctionne !

Détail de la configuration de /etc/dhcpd/hcpd.conf

```
ls /etc/dhcp/
```

```
dhclient.conf       dhclient-exit-hooks.d
dhclient-enter-hooks.d  dhcpd.conf
```

Le principale de la configuration se fait par le fichier /etc/dhcp/dhcpd.conf

Voir http://www.delafond.org/traducmanfr/man/man5/dhcpd.conf.5.html

Il est composé de plusieurs sections, constituée de directives. Une directive a pour syntaxe soit :
des lignes : `clé + valeur + ;`
des directives : `directive;`
des directives avec accolades.

Ces directives organisée en sections :

1. des paramètres globaux qui s'appliquent à tout le fichier,
2. shared-network,
3. subnet,
4. host,
5. group.

Chaque section peut contenir des paramètres et des options. Une section **`group`** peut contenir des

sections **host**.
Au début du fichier, on peut placer des paramètres globaux, comme par exemple la durée des baux, les adresses des DNS... Chaque ligne du fichier de configuration doit se terminer par un **;**, sauf lorsqu'il y a une accolade.
Les commentaires sont possibles en ajoutant un # en début de ligne.

Les options

Voir http://www.delafond.org/traducmanfr/man/man5/dhcp-options.5.html Elles sont déterminées par le mot clé **option**.
Les options plus utiles sont les suivantes :

- **option subnet-mask** : indique le masque de sous-réseau pour la configuration IP.
- **option routers** : indique les routeurs à utiliser.
- **option domain-name-servers** : indique le ou les DNS à utiliser. (On peut aussi bien donner le nom que l'adresse IP ; on peut en donner plusieurs, par exemple, un pour chaque sous-réseau du réseau.
- **option host-name** : indique au client quel nom d'hôte il doit prendre.
- **option domain-name** : fournit au client le nom du domaine arpa dans lequel il se trouve.
- **option broadcast-address** : indique l'adresse de broadcast en vigueur sur le réseau.
- **option dhcp-lease-time** : indique au client la durée de validité du bail.

Les paramètres globaux

Il doivent avoir une signification applicable à toutes les autres déclarations du fichier. Par exemple, on peut redéfinir la durée des baux (max-lease-time et default-lease-time), empêcher le serveur de répondre à des requêtes venant d'hôtes non déclarés (deny unknown-clients;), indiquer le nom de domaine que les machines doivent utiliser, les serveurs DNS...

Par exemple :

```
    - max-lease-time 240;
    - default-lease-time 240;
    - deny unknown-clients;
    - option domain-name "bar.com";
    - option domain-name-servers foo1.bar.com, foo2.bar.com;
subnet 192.168.1.0 netmask 255.255.255.0
    {
    range 192.168.1.2 192.168.1.100;
    range 192.168.1.110 192.168.1.254;
    option broadcast-address 192.168.1.255;
    }
```

Explications : Les cinq premières lignes définissent les paramètres globaux. Les 2 premiers concernent les baux (leases). La ligne suivante dit au serveur de ne pas répondre aux requêtes DHCP venant d'hôtes qu'il ne connaît pas (i.e. non définis dans dhcpd.conf). On définit enfin les paramètres du domaine du réseau (nom de domaine et serveurs DNS).

On définit ensuite le sous-réseau sur lequel le serveur DHCP est censé intervenir : c'est la ligne

“subnet…”. Dans ce sous-réseau, on dit au serveur de ne fournir des adresses IP que dans les plages d'adresses définies par les lignes “range…”. la dernière ligne de la section définit l'adresse de broadcast à utiliser sur le sous-réseau.

- **shared-network**

Cela informe le serveur DHCP que les sous-réseaux (déclarés par différents **subnet**) partagent en réalité le même réseau physique. On utilise ce paramètre pour regrouper plusieurs zones **subnet** seulement si celles-ci concernent le même réseau physique.

Les paramètres rentrés en début de zone seront utilisés pour le boot des clients provenant des sous-réseaux déclarés, à moins de spécifier pour certains hôtes de ne pas booter (zone host). Son utilisation se rapproche de celle de **host** ; il faut l'utiliser si le réseau est divisé en différents sous-réseaux administrés par le serveur DHCP.

- syntaxe :

```
shared-network name {
 [paramètres du réseau partagé];
 subnet IP.0 netmask 255.255.255.224 {
 [paramètres au premier sous-réseau];
 range IP IP;
 }
subnet IP.30 netmask 255.255.255.0 {
 [paramètres du second sous-réseau];
 range IP IP;
 }
}
```

- Par exemple :

```
shared-network FOO-BAR
    {
    filename "boot";

    subnet 192.168.2.0 netmask 255.255.255.224
        {
        range 192.168.2.10 192.168.2.30;
        }

    subnet 192.168.2.32 netmask 255.255.255.224
        {
        range 192.168.2.40 192.168.2.50;
        }

    }
```

- **subnet**

Il permet de définir les sous-réseaux sur lesquels le serveur DHCP doit intervenir. C'est la partie la plus importante du fichier de configuration du serveur DHCP.

-Le paramètre global indispensable est : **range [dynamic-bootp] première-ip [dernière]**
qui définit la zone d'adresses IP (une tranche d'adresse IP) que le DHCP peut distribuer.

-Plusieurs **range** peuvent se suivre. Dans ce cas, on indique une seule adresse IP (pas celle de fin), et le paramètre **dynamic-bootp** doit être ajouté pour indiquer que le DHCP doit répondre aux requêtes *BOOTP* en donnant une adresse.

Sa syntaxe :

```
subnet IP_sous-reseau netmask netmask
    {
    [ paramètres globaux... ]
    [ déclarations... ]
    }
```

On peut bien commencer la zone par des paramètres globaux qui ne seront appliqués que pour les ordinateurs de ce sous-réseau.
Par exemple :
Le nom de domaine à appliquer sur ce sous-réseau (option **domain-name**).
Ensuite, on peut ajouter des déclarations d'hôtes.

- **host**

Il permet permet de déclarer des machines que le DHCP doit connaître pour leur appliquer une configuration particulière. Paramètre non obligatoire, sauf si on déclare un **deny unknown-clients;** en début de fichier pour empêcher le serveur DHCP de répondre à des requêtes provenant d'hôtes non déclarés.

Sa syntaxe :

```
host nom
    {
    paramètres...
    }
```

Un hôte peut être reconnu de deux façons : -en utilisant son nom (le nom qui suit le mot clé host : **host nom**) -en utilisant son adresse mac (ethernet ou wlan).
Dans le second cas, il faut ajouter une ligne dans la déclaration host : **hardware ethernet|token-ring adresse-hardware;**

Il est fortement recommandé d'authentifier les ordinateurs à partir de leur adresse mac plutôt que par leur nom, surtout qu'il sont supposés ne pas posséder de véritable nom de domaine et que l'on peut redéfinir ce nom.

ATTENTION
Si on déside de déclarer dans une directive **host** l'attribution d'une adresse fixe à un hôte,
comme ceci **fixed-address 192.168.2.41;**

dans ce cas cette adresse IP attribuée (fixe) ne doit pas faire partie des zones

d'adresses IP déclarées avec **range** (zone subnet qui indique de 192.168.2.10 à 192.168.2.30)

Exemple : pour fixer l'IP d'un client depuis le serveur DHCP

```
host nom-du-client {
  hardware ethernet j5:de:3a:e5:f6:i2;
  fixed-address 192.168.1.20;
}
```

- **group**

Cette zone est simplement utilisée pour rassembler plusieurs déclarations sur lesquelles on donnera les mêmes paramètres (hôtes, réseaux partagés, sous-réseau, ou d'autres groupe).

Par exemple on crée un groupe composé de plusieurs hôtes pour lesquels il y aura les mêmes paramètres.
Cela évite de ce répéter : on donne les paramètres qu'on veut pour tous les hôtes, réseaux partagés, etc. et on n'a plus qu'à spécifier les particularités qu'on veut pour chaque élément (ici hôte) composant le groupe.

```
group {
[paramètres du groupe]
 host nom1 {
  paramètres spécifiques à nom1
  }
 host nom2 {
  paramètres spécifiques à nom2
  }
 host nom3 {
  paramètres spécifiques à nom3
  }
}
```

- **client BOOTP** :

À savoir : Bootstrap Protocol (BOOTP) est un protocole réseau d'amorçage, qui permet à une machine cliente sans disque dur de découvrir sa propre adresse IP, l'adresse d'un hôte serveur, et le nom d'un fichier à charger en mémoire pour exécution. Le serveur isc-dhcp-server prend en charge le BOOTP.

Exemple de configuration :

```
host nom-du-client {
  filename "/tftpboot.img";
  server-name "nom-serveur";
  next-server nom-serveur;
  hardware ethernet adresse_MAC_du_client;
  fixed-address une_IP;
```

```
}
```

> où `nom-serveur` à la fonction de serveur DHCP, serveur TFTP et passerelle réseau

Il faut avoir installer un serveur TFTP : Voir Install tftp server in Debian

> où `/tftpboot.img` est le nom du fichier extrait par TFTP : voir mettre en place des images TFTP

Pour une installation par le réseau avec tftp-hpa voir : http://www.cyberciti.biz/faq/install-configure-tftp-server-ubuntu-debian-howto/

Pour explication détaillée du fonctionnement du BOOTP ainsi que sur l'amorçage depuis le réseau avec TFTP voir ici : Préparer les fichiers pour amorcer depuis le réseau avec TFTP

La maintenance /var/lib/dhcp/dhcpd.leases

Le fichier /var/lib/dhcp/dhcpd.leases permet d'accéder à une base de donnée persistante des baux attribués.

```
ls /var/lib/dhcp/
```

```
dhclient.br0.leases
dhclient.leases
dhclient-d3b7604e-6f32-4904-8fcd-b98398026559-eth0.lease
dhcpd.leases
dhcpd.leases~
dhclient-d3b7604e-6f32-4904-8fcd-b98398026559-eth1.lease
dhclient.eth0.leases
```

Le fichier `dhcpd.leases` contient les baux persistants.

Le fichier `dhcpd.leases~` est un fichier de sauvegarde des anciens baux.

- Quand il y a des problèmes avec les baux, on fait :

```
mv dhcpd.leases~ dhcpd.leases
```

Puis on redémarre : `/etc/init.d/isc-dhcp-server restart`

Pour plus de détail : man dhcpd.leases

Quand le serveur dhcp est mis en place, pour effectuer des tests depuis le client :

- On pourra supprimer le bail du client (donné par une première configuration du serveur DHCP) :

```
dhclient -r eth0
```

Avec **`ip a s`** ou **`ifconfig`** on voit que le client n'a plus d'IP attribuée.

- Puis pour le client dépourvu d'IP fasse une requête d'IP au serveur DHCP :

(ne pas oublier quand on modifie la configuration du serveur DHCP de le recharger avec la commande **`/etc/init.d/isc-dhcp-server restart`**) :

```
dhclient eth0
```

À propos de dhclient ;
le man dhclient ;
la configuration de dhclient

Agent relais DHCP

On veut maintenant mettre en place un sous-réseau supplémentaire sans ajouter une carte ethernet à notre routeur-passerelle-debian !

Mais les trames ARP et BOOTP ne traversent pas les routeurs.
Il faudrait donc plusieurs serveurs DHCP, un pour chaque segment (chaque sous-réseau).
Si on veut mettre en place plusieurs segments mais qu'on ne dispose que d'un seul serveur, il faut mettre en place sur chaque segment un relais DHCP.

Ce relais va transformer les requêtes multicast en unicast et il sera installé sur le client X. Ce client X du sous-réseau B va devoir répondre aux requêtes DHCP à la place du serveur DHCP installer sur la passerelle. Cet ordinateur X recevra des requêtes broadcast mais ne sait pas y répondre qu'on n'a pas installer et configuré un agent de relais. Cet agent lui permettra de savoir vers quel serveur DHCP envoyer la requête en unicast.

Pour un exemple concret voir : http://www.premont.fr/tutos/dhcp_relais.pdf

Avec Linux on a un agent dhcprelay nommé isc-dhcp-relay.

```
apt-get install isc-dhcp-relay
```

Configuration du relais DHCP

Elle se fait dans le fichier /etc/default/isc-dhcp-relay

- Configuration basique :

```
#adresse du vrai serveur dhcp
SERVER="192.168.1.1"

#interface utilisée par le relais DHCP
```

```
INTERFACE="eth0"
```

Installation

Utilisation