3I 023 – TP 3

Administration réseaux simple
(Services, NAT, DHCP, Pare-feu)

Semaine du 20 Mars 2017

---

1 Introduction à l'administration réseaux

Le but de cette partie est de vous initier aux rudiments de l'administration réseau à travers des scénarios simples utilisant les commandes de base incluses sur de nombreuses distributions Linux. Nous utiliserons Netkit comme émulateur pour créer la topologie réseau demandée. Netkit contient par défaut toutes les commandes et services que nous utiliserons par la suite. Afin d'acquérir une meilleure autonomie, il est indispensable que vous ayez le réflexe de consulter les pages de manuelles des commandes utilisée : `man <commande>`.

Voici quelques aides qui vous seront utiles pour ce TP :

- Utilisez la commande `lstart -f` pour lancer les machines virtuelles du lab en parallèle
- Créer un dossier `shared/` à la racine du lab permet de copier des fichiers dans **toutes** les machines virtuelles du lab. Par exemple, placer le fichier `shared/etc/hosts` à la racine du lab permet de partager le fichier `hosts` entre toutes les machines virtuelles.
  Pour plus d'informations, vous pouvez vous rendre à l'URL `http://wiki.netkit.org/man/man1/lstart.1.html`.

320

1.1 Mise en place de la topologie réseau avec Netkit

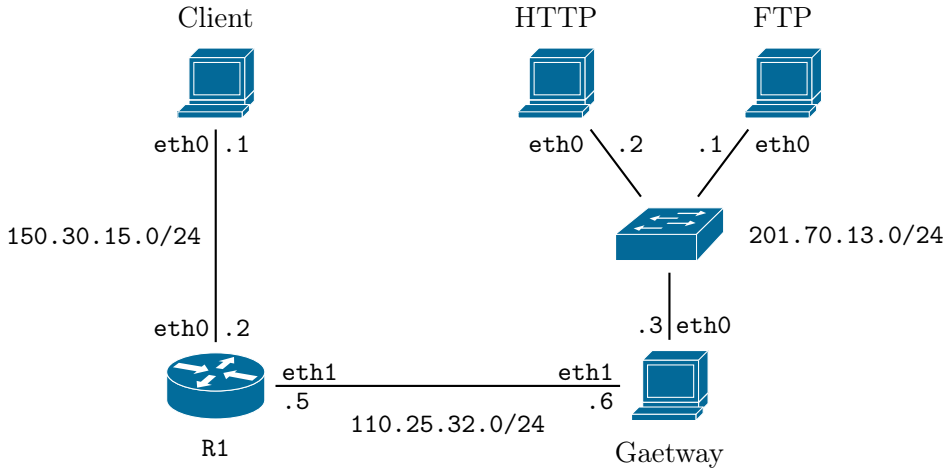

FIGURE 1 – Topologie réseau utilisée pour tester les services

La topologie réseau utilisée dans cette partie (Fig. 1) est proche de celle déjà étudiée lors du précédent TP (TP2 dernière partie) ; il sera donc important de vous inspirer de la configuration du routage statique déjà effectué. Comme nous allons aborder des notions d'administration et le déploiement de services, la topologie étudiée comportera une machine cliente, un routeur, une passerelle et deux serveurs qui hébergeront les service Web et de transfert de fichier. Le tableau ci dessous résume les principales composantes que vous devez mettre en place.

Machine	Nom	Domaine	Interface	Adresse	Sous-réseau
Client	**client**	mondomaine.fr	eth0	`150.30.15.1`	`150.30.15.0/24`
R1	**routeur**	mondomaine.fr	eth0	`150.30.15.2`	`150.30.15.0/24`
R1	**routeur**	monisp.fr	eth1	`110.25.32.5`	`110.25.32.0/24`
Gateway	**gateway**	monisp.fr	eth1	`110.25.32.6`	`110.25.32.0/24`
Gateway	**gateway**	private.fr	eth0	`201.70.13.3`	`201.70.13.0/24`
HTTP	**http**	private.fr	eth0	`201.70.13.2`	`201.70.13.0/24`
FTP	**ftp**	private.fr	eth0	`201.70.13.1`	`201.70.13.0/24`

Création du Lab

1. Créez un dossier `Lab_TP3_Admin`.
2. Dans le dossier `Lab_TP3_Admin`, créez un fichier `lab.conf`.
3. Dans le fichier `lab.conf`, vous allez déclarer les machines virtuelles à lancer, ainsi que les interfaces réseaux de ces machines, et les domaines de diffusion auxquelles elles sont raccordées. Pour cela, pour chaque interface, ajoutez une ligne :
   `nom_machine[numero_interface]="nom_domaine_diffusion"`
4. Dans le dossier `Lab_TP3_Admin`, créez un dossier `<nom_machine_virtuelle>` pour chaque machine virtuelle.
5. Dans le dossier `Lab_TP3_Admin`, créez un fichier `<nom_machine_virtuelle>.startup` pour chaque machine virtuelle.
6. Dans chaque fichier `*.startup`, écrivez les commandes permettant de configurer les interfaces réseaux des machines virtuelles (`ifconfig eth0` ...) et les commandes permettant la configuration du routage statique (`route add` ...).
7. Placez vous dans le dossier `Lab_TP3_Admin` et lancer votre Lab à l'aide de la commande `lstart`.

1.2 Noms de machines et des noms de domaine

Nous avons vu lors du précédent TP qu'il est possible de communiquer entre machines à l'aide de leurs adresses IP. Cependant, cette notation n'est pas pratique pour l'administrateur et les utilisateurs préférant se référer à des noms de machines ou des noms de domaines. Avant d'aborder, dans le prochain TP, les notions de DNS, nous vous proposons d'effectuer la configuration des noms de machines et des noms de domaine de façon statique à l'aide des fichiers de configurations ci dessous.

— Configuration du nom de machine : Le fichier `/etc/hostname` contient le nom de la machine (`hostname`). Il est utilisé lors du démarrage de la machine pour positionner le nom de la machine.

— Configuration du nom de domaine : Le fichier `/etc/hosts` est l'ancêtre du DNS. A la création de l'Internet, il y avait très peu d'ordinateurs connectés, et ce fichier contenait la liste de ces ordinateurs et de leurs adresses IP. Chaque fois que de nouveaux ordinateurs étaient connectés à Internet, il fallait mettre à jour ce fichier pour accéder à ces nouveaux ordinateurs. Vu le nombre croissant d'ordinateurs connectés à Internet, ce fonctionnement n'était plus adapté et à donné lieu à l'invention du DNS que nous aborderons dans le prochain TP. Afin de connaitre les adresses IP d'ordinateurs locaux ou distant, le fichier `/etc/hosts` est constitué d'une ligne pour l'entrée `loopback` et de plusieurs lignes pour les machines distantes.

```
127.0.0.1     localhost       localhost.localdomain
150.30.15.1   client          client.mondomaine.fr
201.70.13.1   ftp             ftp.private.fr
...           ...             ...
::1           ip6-localhost   ip6-loopback
...           ...             ...
```

— Configuration des nom de réseaux : le fichier `/etc/networks` contient le nom des réseaux et leur adresse réseau :

```
default         0.0.0.0
loopback        127.0.0.1
link-local      169.30.15.0
mondomaine.fr   150.30.15.0
private.fr      201.70.13.0
```

Configuration et test des noms de machines et des noms de domaine

1. A l'aide des fichiers de configuration présentés précédemment, effectuez la configuration des noms de machines et des noms de domaines sur toutes les machines du lab. Notez qu'il vous sera indispensable de relancer le service réseau sur chaque machines pour que la configuration soit effective, pour cela lancez la commande `/etc/init.d/networking restart`.
2. Utilisez la commande `route` afin de vérifier les modifications effectuées sur chaque machine.
3. Lancez la commande `ping` de la machine Client vers la machine FTP : `ping ftp.private.fr`.

1.3 Configuration des utilisateurs

La création de comptes utilisateurs différents sur une machine est essentielle pour permettre à chaque utilisateur de disposer de son espace personnel afin d'y effectuer ses propres opérations (stockage, programmes, connexion, etc.). De plus, on distingue les utilisateurs selon leurs droits : l'administrateur de la machine doit avoir le contrôle sur tout le système (`root`) alors que de simples utilisateurs doivent avoir un accès limité à leur répertoire personnel et à certaines commandes. Afin de faciliter l'administration, on utilise la notion de groupe pour regrouper un nombre d'utilisateurs selon leurs droits d'accès.

1.3.1 Principe de l'ajout des utilisateurs et des groupes

L'ajout d'un utilisateur consiste à :

1. Ajout d'un utilisateur avec la commande `useradd` :
   — Syntaxe : `useradd -m` <nom-utilisateur>`-g` <groupe>`-d` <repertoire-personnel>
   L'option `-m` permet de recopier les fichiers de configuration du shell. On peut remplacer le shell courant par un shell spécifique avec l'option `-s` (par exemple `-s /etc/ftponly`).
   — La suppression d'un utilisateur se fait en utilisant la commande
   `userdel -r` <nom-utilisateur>
   L'option `-r` permet de supprimer le répertoire personnel de l'utilisateur à supprimer
2. Associer un mot de passe à l'utilisateur avec la commande `passwd` <nom-utilisateur>, ajout d'une entrée dans le fichier `/etc/passwd`. Pour l'utiliser en ligne de commande, utilisez la commande suivante : `echo -e "password\npassword" | passwd myuser`.
3. Définir à quel groupes appartient l'utilisateur (ajout d'une entrée dans le fichier `/etc/group`)
   — Si le groupe n'existe pas, créer un groupe avec la commande `groupadd` <nom-groupe>.

   - On rajoute l'utilisateur à un groupe secondaire avec la commande : `usermod -G <groupe-secondaire1>, <groupe-secondaire2><nom-utilisateur>`.
4. Créer le répertoire personnel de l'utilisateur avec les commandes suivantes :
   - `mkdir /home/<nom-utilisateur>`
   - `cp /etc/skel/* /home/<nom-utilisateur>`
5. Créer le fichier de configuration personnelle du shell.

Certaines commandes sont aussi utiles pour changer le groupe, et les droits d'accès de certains répertoires après avoir créer des utilisateurs.

- Changement d'utilisateur sur un répertoire :
  `chown <nom-utilisateur>/home/<nom-utilisateur-bis>`
- Changement de droits sur les fichiers d'un répertoire : `chmod u+rwx /home/<nom-utilisateur>`
- Changement de groupe sur un répertoire : `chgrp <nouveau-groupe>/home/<nom-utilisateur>`

1.3.2 Ajout d'utilisateurs et de groupes sur les machines du Lab

1. Sur les machines Client, HTTP et FTP créez un utilisateur `myuser` lié à un groupe `mygroup`.
2. Vérifiez que l'utilisateur et le groupe sont crées en regardant les fichier `/etc/passwd` et `/etc/group`.
3. Connectez vous sur le compte de l'utilisateur en utilisant la commande : `su myuser`.

1.4 Déploiement de services élémentaires

1.4.1 Connexion à distance avec SSH

Pour administrer des machines à distance, il est nécessaire de se connecter de façon sécurisée. Pour cela, nous utiliserons le *Secure Shell* (SSH) qui est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite toutes les trames sont chiffrées. Il devient donc impossible d'utiliser un *sniffer* pour voir ce que fait l'utilisateur. Pour récupérer des données distantes de façon sécurisée, un programme similaire est utilisé, `scp`.

1.4.2 Utilisation de SSH

1. Vérifier que le processus du programme `ssh` est actif : `ps aux | grep ssh`
2. Si le processus n'est pas actif, lancez-le à l'aide de la commande : `/etc/init.d/ssh restart`
3. Connectez-vous à l'aide du mot de passe configuré précédemment sur le compte `myuser` de la machine HTTP : `ssh myuser@http.private.fr`
4. Déconnectez-vous en utilisant la commande `exit`

1.4.3 Serveur web : Apache

Dans cette section, nous allons voir comment lancer un serveur web basé sur Apache. *Apache HTTP Server*, souvent appelé Apache, est un logiciel de serveur HTTP produit par l'Apache Software Foundation. C'est l'un des serveurs HTTP les plus populaires du Web.

1. Connectez-vous sur la machine HTTP.
2. Vérifiez l'existence du programme Apache et regardez le fichier de configuration par défaut : `vi /etc/apache2/sites-enabled/000-default`

3. A partir de ce fichier, retrouver la page `index.html` utilisée par défaut.
4. Lancez le *deamon* apache avec la commande : `/etc/init.d/apache2 start`
5. Lancez une capture à l'aide de `tcpdump` pour analyser l'échange de trames lié au serveur web.
6. Depuis la machine Client, connectez-vous au serveur web avec l'outil `telnet` : `telnet http.private.fr 80`
7. Lorsque le prompt de telnet s'affiche, récupérez le contenu de la page par défaut en utilisant la syntaxe : `GET / HTTP/1.0` (appuyez sur entrée 2 fois pour que la requête soit envoyée).
8. Regardez les traces capturées par `tcpdump`, que constatez-vous ?

1.4.4 Transfert de fichiers : Serveur FTP

Le protocole FTP est l'un des protocoles les plus connus avec HTTP(s), il permet à plusieurs personnes de partager des fichiers. Dans cette partie, nous utiliserons le programme ProFTP qui est l'un des serveurs FTP les plus connus sous Unix. Il possède plusieurs fonctions avancées, comme les ratios ou les *virtuals hosts*, dont nous ne parlerons pas ici. Pour en savoir plus, vous trouverez des informations complémentaires sur `proftp.org`.

Configuration du service FTP :

1. **L'utilisateur `nobody`.** Par défaut le *daemon* `proftpd` se lance avec les privilèges de `root` et cela pose évidemment des problèmes de sécurité. C'est pourquoi il est conseillé d'utiliser un utilisateur sans droits particuliers. Le plus indiqué étant `nobody` du groupe `nobody`.
   — Vérifiez l'existence de l'utilisateur et du groupe `nobody` dans les fichier `/etc/group` et `/etc/passwd`.
   — Si le groupe et l'utilisateur `nobody` n'existent pas, créez les avec les commandes `groupadd` et `useradd` :
   ```
   groupadd nobody
   useradd nobody -d / -s /bin/false
   usermod nobody -g nobody
   ```
2. **Les utilisateurs.** Nous allons créer deux utilisateurs, le premier, `adminftp` qui sera le login qui vous permettra de vous loguer depuis les machines distantes pour rajouter des fichiers sur le serveur FTP. Le second `userftp` qui sera le login que devront entrer vos utilisateurs pour se loguer à votre FTP. Vos utilisateurs auront un accès en lecture seule, c'est à dire qu'ils pourront juste récupérer des fichiers. Si vous souhaitez que d'autres utilisateurs puissent déposer des fichiers sur votre serveur, soit vous leur donnez le mot de passe correspondant au login adminftp, soit vous leur créez un compte.
   — Création des utilisateurs : `useradd <utilisateur>-s /bin/false`
   — Affectation des mots de passe : `passwd <utilisateur>`
   — Il est très important que les utilisateurs de votre FTP aient des droits restreints et donc ne soient pas des utilisateurs réguliers de votre Linux d'où le `/bin/false` au lieu de `/bin/bash` habituel. Maintenant éditez `/etc/shells` et rajoutez la ligne `/bin/false` (si elle n'y est pas).
3. **Création du partage.** Il est conseillé de mettre tous les fichiers que vous souhaitez rendre disponibles par FTP dans un même répertoire, par exemple `/mnt/ftp` avec autant de sous répertoires que vous désirez comme `/mnt/ftp/images`, `/mnt/ftp/docs`. Tout ceci se fait simplement par les commandes suivantes :
   ```
   mkdir /mnt/ftp
   cd /mnt/ftp mkdir images docs
   chmod -R 777 /mnt/ftp
   ```

4. **Configuration.** La configuration de ProFTP est effectuée à l'aide de deux fichiers, le fichier `/etc/ftpusers` permet d'exclure l'accès à certains utilisateurs et le fichier `/etc/proftpd/proftpd.conf` permet de configurer en détail le serveur. Pour cette partie, nous allons nous concentrer sur la configuration du serveur à l'aide d'un exemple de fichier de configuration. Remplacez les lignes correspondantes du fichier ci dessous dans le fichier `/etc/proftpd/proftpd.conf` :

```
# le nom du serveur
ServerName "ProFTP server"

# le daemon reste en émmoire et ecoute les connections
ServerType standalone

# Autoriser l'usage de /etc/ftpusers
UseFtpUsers on

# Repertoire dans lequel arrivent les utilisateurs acceptes
DefaultChdir /mnt/ftp

# Repertoire racine, les connectes au ftp ne verront que lui et son contenu
DefaultRoot /mnt/ftp

# Utile surtout pour les "virtuals hosts" mais laissez ainsi
DefaultServer on

# Le daemon ecoute sur le port 21
Port 21

# Seul le proprietaire d'un fichier peut le modifier
Umask 022

# Proftp sera lance avec les privileges de nobody (c'est a dire aucuns)
User nobody
Group nobody

# Un utilisateur peut ecraser ses propres fichiers
AllowOverwrite on

# Seul l'utilisateur adminftp a le droit d'ecrire dans /mnt/ftp
 <Directory /mnt/ftp>
    <Limit WRITE>
        AllowUser adminftp
        DenyAll
    </Limit>
 </Directory>
```

5. **Lancement du *deamon*.** Comme notre serveur FTP est configuré en *standalone*, il faut le supprimer de la liste des *deamons* exécutables par `inetd`. Pour cela utilisez les commandes suivantes :
   — Commentez (rajouter un `#`) sur la ligne suivante dans `/etc/inetd.conf` (s'il n'y en pas, ne faites rien) :
   `ftp stream tcp nowait root /usr/sbin/tcpd proftpd`
   — Relancer `inetd` par : `killall -HUP inetd`
6. Maintenant vous pouvez lancer votre serveur FTP à l'aide de la commande :
   `/usr/sbin/proftpd`
7. Vérifiez que le *deamon* est bien lancé, pour cela utilisez la commande `ps aux | grep proftp`
8. Connexion et ajout de fichiers sur le serveur FTP :

— Utilisez la commande `ftp` avec l'utilisateur `userftp` : `ftp ftp.private.fr`
— Essayez de regarder le contenu du serveur et ajouter un fichier puis quittez la connexion.
— Utilisez la commande `ftp` avec l'utilisateur `adminftp` : `ftp ftp.private.fr`
— Essayez de regarder le contenu du serveur et ajouter un fichier puis quittez la connexion.

2 NAT, DHCP et pare-feu

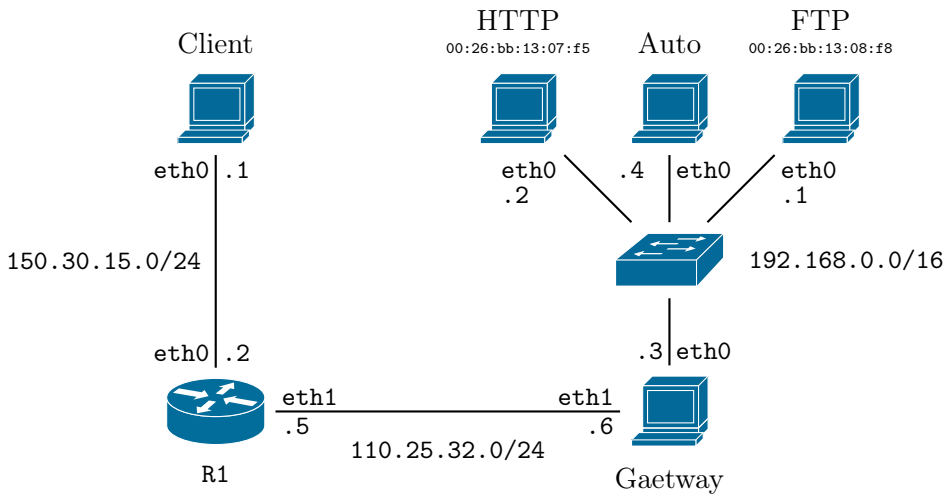

FIGURE 2 – Topologie réseau utilisée pour la passerelle NAT et DHCP

Le but de cette partie est la configuration d'un réseau privé connecté aux réseaux extérieurs grâce à une passerelle implémentant du NAT. La configuration des interfaces du réseau privé sera automatique et basée sur le protocole DHCP. Nous proposons aussi de configurer la passerelle comme pare-feu, ce qui représente une introduction aux techniques de filtrage de trafic.

Le NAT est défini dans la RFC 1631. Le NAT permet d'utiliser des adresses n'ayant pas de signification globale (par exemple des adresses privées définies dans la RFC 1918, non routables) pour se connecter a travers l'Internet en traduisant celles-ci en adresses globales routables.

2.1 Mise en place de la topologie réseaux avec Netkit

La topologie réseau utilisée dans cette partie (Fig. 2) est identique à celle utilisée dans la première partie à l'exception de l'adressage proposé pour le réseau privé. Pour cela, la passerelle aura son interface `eth1` connectée au réseau extérieur (`110.25.32.0/24`) et une interface `eth0` connecté au réseau privé. Les machines HTTP et FTP du réseau privé on respectivement des adresses MAC distinctes (`00:26:bb:13:07:f5` et `00:26:bb:13:08:f8`) qui servirons à la configuration de leurs interfaces par DHCP. Nous vous proposons de garder le Lab utilisé précédemment et de le modifier en fonction comme cela sera proposé dans la suite de cette partie.

2.2 Mise en place d'une passerelle NAT et DHCP avec un Pare-feu

2.2.1 Configuration de la passerelle sur le réseau

Notre réseau local utilise une classe d'adresse réservée aux réseaux privés de classe C : `192.168.0.x`. La passerelle possède deux cartes réseau :

— `eth0` est branchée sur le réseau local (adresse IP `192.168.0.3`).
— `eth1` est branchée sur le réseau extérieur (adresse IP `110.25.32.6`).

Afin d'éviter l'utilisation de la commande `ifconfig` pour chaque configuration d'interfaces, le système Debian, utilisé par netkit, propose de regrouper les configurations des interfaces dans un même fichier : `/etc/network/interfaces`.

1. Dans le fichier de configuration `/etc/network/interfaces` de la passerelle, remplacer les informations relatives aux interfaces en fonction du fichier exemple fournit ci dessous :
```
# /etc/network/interfaces configuration file for ifup(8), ifdown(8)
# The loopback interface
auto lo
iface lo inet loopback

# eth0 (lan) : la carte reseau eth0 est branchee sur le LAN
auto eth0
iface eth0 inet static
address <adresse_IP>
network <adresse_reseau>
netmask <netmask>
broadcast <broadcast>

# la carte eth1 est branchee sur le reseau exterieur
auto eth1
iface eth1 inet static
address <adresse_IP>
network <adresse_reseau>
netmask <netmask>
broadcast <broadcast>
```
2. Relancez le *deamon* qui gère la configuration réseau : `/etc/init.d/networking restart`

Il est important que la passerelle puisse *forwarder* les paquets vers les machines du LAN. Pour cela, il existe plusieurs possibilités :

— Décommentez la ligne `net.ipv4.ip_forward=1` du fichier `/etc/sysctl.conf`
— Activation de la fonction de *forwarding* IP au niveau du noyau :
```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

2.2.2 Configuration des machines du réseau privé

De même que pour la passerelle, nous utiliserons le fichier `/etc/network/interfaces` pour configurer les interfaces des machines du réseau privé.

1. Configuration des machines qui auront une adresse IP fournie par le DHCP en fonction de leurs adresse MAC.
2. Pour les machines HTTP et FTP, modifiez le fichier `/etc/network/interfaces` avec les informations suivantes :
```
auto eth0
iface eth0 inet dhcp
hwaddress ether <Adresse_Mac>
```
3. Pour la machine Auto, modifiez le fichier `/etc/network/interfaces` avec les informations suivantes :
```
auto eth0
iface eth0 inet dhcp
```
4. Sur chaque machine modifiée, relancer le *deamon* `networking` : `/etc/init.d/networking restart`

2.2.3 Configuration du serveur DHCP

Le serveur DHCP de la passerelle permettra d'attribuer automatiquement aux ma- chines du réseau une adresse IP faisant partie de `192.168.0.X`. La règle d'adressage est la suivante : certaines machines sont enregistrées (grâce à leur adresse MAC) et auront toujours la même adresse IP, les autres auront une adresse IP attribuée dynamiquement, entre `192.168.0.100` et `192.168.0.200`. Tout ceci est défini dans le fichier `/etc/dhcp3/dhcpd.conf` :

1. Configuration du serveur DHCP : recopiez les informations utiles de l'exemple ci dessous dans le fichier `/etc/dhcp3/dhcp.conf` :

```
# Fichier de configuration du serveur DHCP Debian GNULinux
# Consultez "man dhcpd.conf" pour avoir toutes les informations sur les options

# DEBUT des options GLOBALES

# Les options globales s'appliqueront par defaut a tous les sous-reseaux
# Nom de domaine du reseau local
option domain-name "private.fr";

# Adresses des serveurs DNS (ésparees par une virgule)
# option domain-name-servers 212.27.32.176, 212.27.32.177 ;

# Duree du bail en secondes
defaultleasetime 6000;
maxleasetime 6000;

# Le serveur DHCP est autoritaire pour les sous-reseaux declares ci-dessous
authoritative ;
# FIN des options GLOBALES

# DEBUT de la declaration des sous-reseaux et des machines

# Declaration du sous-reseau 192.168.0.0/255.255.0.0
subnet 192.168.0.0 netmask 255.255.0.0
{
    # Plage d'adresses a attribuer pour les machines non declarees
    range 192.168.0.100 192.168.0.200;

    # Adresse du routeur
    option routers 192.168.0.3;
}

# Declaration de la machine "ftp"
# La declaration d'une machine permet de lui attribuer une adresse IP fixe
host ftp
{
    # Adresse MAC de la machine
    hardware ethernet 00:26:bb:13:08:f8;

    # Adresse IP a attribuer
    fixed-address 192.168.0.1;
}

# Declaration de la machine "HTTP"
# La declaration d'une machine permet de lui attribuer une adresse IP fixe
host http
{
    # Adresse MAC de la machine
```

```
        hardware ethernet 00:26:bb:13:07:f5;

        # Adresse IP a attribuer
        fixed-address 192.168.0.2;
    }
```

2. Lancer le *deamon* `dhcp` sur la passerelle Gateway : `/etc/init.d/dhcp3-server`

2.3 Configuration d'`iptables` pour partager la connexion

La commande `iptables` est la commande par défaut qui permet de contrôler le trafic et donc d'établir des règles pour votre pare-feu. L'annexe, présente brièvement les options utilisée par `iptables`. Pour résumer, `iptables` remplit deux fonctions :

— Pare-feu, en contrôlant les paquets qui veulent passer par les ports de la passerelle ;
— Partage de connexion, en redirigeant les paquets qui transitent par la passerelle (NAT).

Pour regarder les règles de filtrages déjà existantes, utilisez la commande : `iptables -L -v`.

Remise à zéro des règles de filtrage. Utilisez les commandes ci dessous pour réinitialiser les règles de filtrage par défaut :

— Vider les règles iptables : `iptables -F`
— Supprimer les tables et règles iptables : `iptables -X`
— Vider les règles relatives au nat : `iptables -t nat -F`
— Supprimer les règles relatives au nat : `iptables -t nat -X`
— Bloquer les connexions entrantes par défaut : `iptables -P INPUT DROP`
— Accepter les connections *forwardées* par défaut : `iptables -P FORWARD ACCEPT`
— Accepter les connections sortantes par défaut : `iptables -P OUTPUT ACCEPT`

2.3.1 Configuration des règles du pare-feu

Avant de configurer les règles de partage de connexion, nous proposons de définir certaines règles pour le filtrage du trafic en entrée. Les commandes à utiliser sont données ci-dessous.

```
# Accepter les paquets entrants relatifs a des connexions deja etablies
iptables -A INPUT -m conntrack --ctstate ESTABLISHED -j ACCEPT

# Autoriser les connexions TCP entrantes sur les ports 20 et 21
iptables -A INPUT -p tcp -dport 20 -j ACCEPT
iptables -A INPUT -p tcp -dport 21 -j ACCEPT

# Autoriser les connexions TCP entrantes sur le port 22
# (pour que le serveur SSH soit joignable de l'exterieur)
iptables -A INPUT -p tcp -dport 22 -j ACCEPT

# Autoriser les connexions TCP entrantes sur le port 80
# (pour que le serveur HTTP soit joignable de l'exterieur)
iptables -A INPUT -p tcp -dport 80 -j ACCEPT

# Autoriser les flux TCP et UDP entrants sur les ports 5000 a 6000
iptables -A INPUT -p tcp -dport 5000 :6000 -j ACCEPT
iptables -A INPUT -p udp -dport 5000 :6000 -j ACCEPT

# Accepter le protocole ICMP (i.e. le "ping")
iptables -A INPUT -p icmp -j ACCEPT

# Bloquer le reste du trafic (changement de la politique sur le chaine INPUT)
iptables -P INPUT DROP
```

2.4 Règles pour le partage de connexion (i.e., le NAT)

Pour que notre système fasse office de "serveur NAT", nous devons utiliser la table `nat` de `iptables`. Si l'on veut permettre à un utilisateur du LAN d'aller sur un serveur extérieur, les paquets auront leur champ IP source modifié par le NAT. Ainsi, les paquets qui passeront par la chaine `FORWARD` seront modifiés pour faire comme s'ils provenaient du NAT. Il est donc nécessaire d'agir après cette chaîne. Pour cela `iptables` doit agir sur la chaîne `POSTROUTING` pour changer les en-têtes relatives à la source. La commande utilisée est la suivante :

```
# (Remplacez "eth1" par votre interface connectee au reseau exterieur)
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

2.4.1 Règles pour la translation de ports

Maintenant, nous voulons rediriger les paquets venant de l'extérieur vers les machines correspondantes dans notre LAN. Si une requête est faite sur le port 80, nous voulons la rediriger vers la machine HTTP. Cela s'appelle de la traduction d'adresse de destination tout simplement parce que nous changeons l'adresse IP de destination afin que les paquets arrivent sur la bonne machine. Heureusement, `iptable` permet, grâce à la cible `DNAT` (Destination NAT), de changer l'adresse IP de destination. Il est cependant nécessaire de le faire avant que la décision de routage soit prise. Les commandes ci-dessous indiquent les règles utilisées par `iptables` pour le pré-routage (`PREROUTING`) :

```
# Debut des regles de PORT FORWARDING
# Les requetes TCP recues sur le port 22 sont forwardees a la machine http
iptables -t nat -A PREROUTING -p tcp -dport 22 -i eth1 -j DNAT --to-destination
    192.168.0.2:22

# Les requetes TCP recues sur le port 80 sont forwardees a la machine 192.168.0.2
iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth1 -j DNAT --to-destination
    192.168.0.2:80

# Les requetes TCP recues sur le port 21 sont forwardees a la machine
iptables -t nat -A PREROUTING -p tcp --dport 21 -i eth1 -j DNAT --to-destination
    192.168.0.1:21
```

2.4.2 Test des connections

1. Vérifiez que les adresses des machines HTTP, FTP et Auto sont bien configurées par le DHCP.
2. Depuis la machine HTTP *pinger* le réseau extérieur vers la machine Client.
3. Est ce que le `ping` de la machine Client vers l'adresse privée de la machine HTTP marche ?
4. A l'aide des commandes `iptables` fournies précédemment, empêchez le `ping` (ICMP echo/-reply) d'être activé.
5. Vérifiez que le `ping` ne marche plus.