M1101
Initiation aux Réseaux
Travaux pratiques

Sami Evangelista

IUT de Villetaneuse

Département Réseaux et Télécommunications

2017–2018

http://www.lipn.univ-paris13.fr/~evangelista/cours/M1101

Table des matières

TP 1 — Réalisation et configuration de base d'un réseau

Le TP est à faire par groupe de 3. Chaque trinôme rendra à l'enseignant un compte-rendu dans lequel seront notées les réponses aux questions, les observations éventuelles ainsi que les commandes exécutées en recopiant, si nécessaire, les sorties de ces commandes (ce que la commande affiche dans le terminal).

L'objectif de ce TP est de créer un petit réseau puis réaliser quelques opérations de configuration et de test : attribution d'adresses IP, configuration des interfaces, test d'accessibilité entre machines, mesures de débit,... Le TP montre également comment utiliser SSH (Secure SHell) pour se connecter à distance sur une machine et copier des fichiers entre machines.

Quelques indications avant de commencer :

— Vous utiliserez l'image mageia3. Vous vous connecterez en utilisant le compte suivant :
  identifiant : `etudiant`
  mot de passe : `etudiant`
— Pensez à consulter le manuel (`man <nom-commande>`) ou Internet pour avoir une description plus complète des commandes utilisées.
— La plupart des commandes que nous utiliserons ne peuvent être exécutées que par l'utilisateur root (l'administrateur du système). De même, les fichiers de configuration que nous modifierons. La commande `su -` (pour *switch user*) permet d'ouvrir une session sous l'identité root. Le mot de passe de l'utilisateur root est `iutparis13`.
— Dans tout le TP on travaillera uniquement sur l'interface réseau connectée à la baie de brassage de la salle de TP. C'est normalement l'interface eth1. L'autre interface, eth0, est celle connectée au réseau de l'IUT et qui vous permet d'accéder à Internet.

Exercice 1 — Câblage des machines

Prendre un switch et y connecter les trois câbles Ethernet sortant de la baie de brassage et reliés aux trois machines du groupe.

Exercice 2 — Conception du réseau

Les trois machines du groupe reliées au switch via leurs interfaces eth1 formeront un réseau IP.

Q 2.1 *Calcul du masque* On souhaite que le masque de ce réseau soit le plus long possible (c'est-à-dire contenant le plus de bits à 1) en laissant pour le host-id juste assez de bits pour coder les identifiants des trois machines du réseau. Quel sera alors le masque du réseau ?

Correction

Il y a trois machines dans le réseau. On doit donc pouvoir coder trois adresses IP auxquels il faut rajouter l'adresse du réseau et l'adresse de diffusion, ce qui fait en tout 5 valeurs à coder. 2 bits ne sont pas suffisants mais avec 3 bits on peut coder $2^3 = 8$ valeurs différentes ce qui nous suffit. Le host-id sera donc codé sur 3 bits et le net-id sur $32 - 3 = 29$ bits.
Le masque en binaire obtenu en mettant les bits du net-id à 1 et ceux du host-id à 0 sera donc :

11111111 11111111 11111111 11111000

et en notation décimale pointé :

255.255.255.248

◆

Q 2.2 *Choix de l'adresse de réseau* Choisir une adresse pour le réseau de la forme $10.N.0.X$ où N est un numéro attribué au groupe par l'enseignant et X un octet que vous choisirez différent de 0. Justifier le choix de la valeur de X. Quelle est l'adresse de diffusion de votre réseau ?

Correction

Dans l'adresse de réseau, les 3 bits du host-id (les 3 bits de poids faible de l'octet X) doivent rester à 0. Par exemple, $X = 10000000_2 = 128$ est valide mais pas $X = 100000011_2 = 131$. On choisit $X = 128$. Notre réseau aura l'adresse 10.1.0.128.
L'adresse de diffusion est obtenue en gardant les bits du net-id et en mettant ceux du host-id à 1. Il suffit donc de mettre à 1 les trois derniers bits du dernier octet 128, ce qui donne $10000111_2 = 135$. L'adresse de diffusion est donc 10.1.0.135. ◆

Q 2.3 *Choix des adresses IP* Choisir trois adresses IP pour les machines du réseau. Justifier ces choix.

Correction

On peut attribuer à une machine n'importe quelle adresse IP se trouvant entre l'adresse de réseau et l'adresse de diffusion (non incluses). On choisit : 10.1.0.129, 10.1.0.130 et 10.1.0.131. ◆

Exercice 3 — Configuration des interfaces

Il y (au moins) deux façons de configurer une interface réseau : soit à l'aide de la commande `ifconfig`, soit en modifiant les fichiers de configuration des interfaces puis en utilisant les commandes `ifdown` et `ifup`. Nous utiliserons cette deuxième possibilité.

Cet exercice est à réaliser sur les trois machines du groupe.

Q 3.1 *Fichier de configuration* Supprimer le fichier de configuration actuel de l'interface `eth1` puis écrire un nouveau fichier afin que :
— l'adresse IP et le masque associés à cette interface soient ceux choisis à l'exercice précédent ;
— l'interface soit activée au démarrage de l'ordinateur.
Donner le contenu du fichier créé pour une des trois machines. Justifier le choix de la valeur du paramètre BOOTPROTO.

Correction

L'interface est activée statiquement étant donné que c'est nous qui fixons les paramètres de l'interface (adresse IP et masque ici) et que ces paramètres ne sont pas donnés par un serveur DHCP. Le paramètre BOOTPROTO vaut donc static.
Voici le fichier `/etc/sysconfig/network-scripts/ifcfg-eth1` pour la machine 10.1.0.129 :

```
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.1.0.129
NETMASK=255.255.255.248
```

Seule l'adresse IP change dans les fichiers de configuration des autres machines. ◆

Q 3.2 *Activation de l'interface* Désactiver puis réactiver l'interface à l'aide des commandes `ifdown` et `ifup`. Vérifier ensuite à l'aide de la commande `ifconfig` que l'interface a bien été configurée.

Correction

En lançant les commandes `ifdown eth1` puis `ifup eth1` on désactive eth1 puis on la réactive avec la nouvelle configuration.
En lançant ensuite `ifconfig` on vérifie que l'adresse et le masque ont bien été attribués :

```
$ ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:0A:F7:16:C1:68
          inet adr:10.1.0.129  Bcast:10.1.0.135  Masque:255.255.255.248
          adr inet6: fe80::20a:f7ff:fe16:c168/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:192 (192.0 b)  TX bytes:748 (748.0 b)
          Interruption:18
```

◆

Q 3.3 *Test de la connexion* Pour indiquer qu'une machine accepte de répondre aux demandes d'écho envoyées par la commande `ping`, il est nécessaire que les fichiers
— `/proc/sys/net/ipv4/icmp_echo_ignore_all` et
— `/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts`
contiennent tous les deux la valeur 0 (s'ils contiennent une valeur différente de 0 les demandes d'écho seront ignorées).
Utiliser la commande `echo` combinée avec une redirection pour que ces deux fichiers contiennent 0. À l'aide de la commande `ping`, vérifier ensuite que chaque machine peut être jointe par les autres machines du groupe.

Correction

Depuis la machine 10.1.0.129 on peut faire :

```
# echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all
# echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
$ ping 10.1.0.130
PING 10.1.0.130 (10.1.0.130) 56(84) bytes of data.
64 bytes from 10.1.0.130: icmp_seq=1 ttl=64 time=0.187 ms
```

```
--- 10.1.0.130 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.187/0.187/0.187/0.000 ms
```

◆

Q 3.4 *Options de la commande ping* Effectuer les tests suivants pour essayer les différentes options de ping :

(a) Envoi d'une demande d'écho aux deux autres machines simultanément (sans préciser les adresses des destinataires une à une).

(b) Envoi de 10 demandes d'écho.

(c) Envoi de 5 demandes d'écho à une fréquence de 1 paquet toutes les 2 secondes.

Correction

(a) Il faut pinguer l'adresse de diffusion en utilisant l'option `-b` (pour broadcast) de ping :

```
$ ping -b 10.1.0.135
WARNING: pinging broadcast address
PING 10.1.0.135 (10.1.0.135) 56(84) bytes of data.
64 bytes from 10.1.0.131: icmp_seq=1 ttl=64 time=0.203 ms
64 bytes from 10.1.0.130: icmp_seq=1 ttl=64 time=0.213 ms (DUP!)

--- 10.1.0.135 ping statistics ---
1 packets transmitted, 1 received, +1 duplicates, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.203/0.208/0.213/0.005 ms
```

On a bien reçu deux réponses : la première de 10.1.0.131 et la suivante de 10.1.0.130.

(b)

```
$ ping -c 10 10.1.0.130
```

```
$ ping -c 5 -i 2 10.1.0.130
```

◆

Exercice 4 — Mesure de performances

Nous allons mesurer les temps de transfert de données sur notre réseau. Cet exercice est à réaliser sur deux machines du groupe : la machine cliente (l'émetteur des données) et la machine serveur (le récepteur).

(c) **Q 4.1** *Débit de la liaison* À l'aide de la commande `ethtool` (décrite en annexes, page 7), déterminer le débit de l'interface eth1.

Correction

On exécute la commande suivante :

```
# ethtool eth1
```

Le paramètre `Speed` indique le débit de la ligne. Les switchs sont normalement à un débit de 100 Mbit/s. ◆

Q 4.2 *Temps de transfert théorique* Avec le débit observé quel est le temps théorique de transfert de 100 Mo de données entre deux machines du réseau ?

Correction

Avec un débit de 100 Mbit/s, on obtient le temps de transfert théorique suivant :

$$\text{temps} = \frac{\text{quantité de bits transmis}}{\text{débit (en bit/s)}} = \frac{8 \times 100\,000\,000}{100\,000\,000} = 8\ sec.$$

◆

Q 4.3 *Temps de transfert effectif* Utiliser la commande `iperf` (décrite en annexes, page 7) pour mesurer le temps de transfert de 100 Mo entre deux machines du groupe. Comparer au temps théorique calculé à la question précédente. Comment expliquer la différence entre ces deux temps ?

Correction

Côté serveur (10.1.0.130) :

```
$ iperf -s
```

Côté client :

```
$ iperf -n 100000000 -c 10.1.0.130
------------------------------------------------------------
Client connecting to 10.1.0.130, TCP port 5001
```

```
TCP window size: 22.9 KByte (default)
------------------------------------------------------------
[  3] local 10.1.0.129 port 35155 connected with 10.1.0.130 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 8.4 sec  95.4 MBytes  94.8 Mbits/sec
```

Le temps de transfert est de 8.4 secondes. On atteint un débit de 94.8 Mbits/sec. La différence s'explique principalement par les en-têtes qui sont rajoutés sur les paquets envoyés : on a vu que lors d'un transfert, les données sont découpées en paquets et que sur chaque paquet envoyé, un en-tête est rajouté contenant l'adresse IP de la machine destinataire, celle de la machine source, ... Par conséquent la quantité de bits réellement envoyée est supérieure aux 100 Mo transmis. ◆

Q 4.4 *Changement du mode de transmission* Utiliser `ethtool` pour passer la liaison au switch en half-duplex (à la fois sur le client et sur le serveur). Reprendre le test de la question précédente. Quel temps observe-t-on ? Lancer la commande `ifconfig`. Quelle information affichée par celle-ci permet d'expliquer la différence ?

Correction

Pour passer en half duplex on utilise maintenant la commande suivante :

```
# ethtool -s eth1 speed 100 duplex half
```

En relançant la copie en half duplex, le temps de transfert augmente de quelques secondes car le switch et la machine peuvent alors transmettre des données simultanément, ce qui crée des collisions. On peut le voir comme ceci :

```
$ ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:0A:F7:16:BD:63
          inet adr:10.1.0.130  Bcast:10.1.0.135  Masque:255.255.255.248
          adr inet6: fe80::20a:f7ff:fe16:bd63/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:87614 errors:18576 dropped:0 overruns:0 frame:125
          TX packets:161750 errors:0 dropped:0 overruns:0 carrier:0
          collisions:22926 lg file transmission:1000
          RX bytes:6334991 (6.0 MiB)  TX bytes:245375905 (234.0 MiB)
          Interruption:17
```

◆

Exercice 5 — Connexion et envoi de fichiers à distance

Dans cet exercice, nous allons utiliser les commandes `ssh` et `scp` qui permettent de se connecter à distance sur une autre machine et de copier des fichiers d'une machine à l'autre. Une machine du trinôme doit jouer le rôle du serveur. C'est sur celle-ci que les deux autres machines (les clients) se connecteront et copieront des fichiers.

Q 5.1 *Démarrage du service sshd* Pour pouvoir utiliser `ssh` et `scp` il faut que le service sshd qui traite les connexions des clients soit lancé sur la machine serveur. Démarrer ce service puis vérifier qu'il est bien lancé.

Correction

Sur la machine serveur (10.1.0.129) :

```
# systemctl start sshd.service
# systemctl status sshd.service
```

◆

Q 5.2 *Visualisation des processus* Lancer, toujours sur la machine serveur, la commande `ps aux`. Celle-ci affiche la liste des processus en cours d'exécution avec leurs pids, leurs consommations en mémoire, leurs heures de lancement, ... Retrouver la ligne correspondant au processus sshd. Quel est son pid ? Dans quel état est le processus ? Pourquoi ?

Correction

La ligne correspondant à sshd est la suivante.

```
root      4640  0.0  0.0  44124  2628 ?        Ss   13:27   0:00 /usr/sbin/sshd -D
```

Le pid se trouve dans la deuxième colonne : 4640. L'état du processus est `Ss`, ce qui signifie que le processus est en sommeil (sleeping). Il est en attente de requêtes provenant de clients. ◆

Q 5.3 *Copie à distance* Sur les machines clientes, créer un fichier vide ou contenant une suite de caractères quelconque. Copier ensuite, sous l'identité etudiant, ce fichier dans le répertoire `/tmp` de la machine serveur. Vérifier ensuite, sur la machine serveur, que ce fichier a bien été créé.

Correction

Sur les machines clientes :

```
$ echo caractères > fichier
$ scp fichier etudiant@10.1.0.130:/tmp
```

On peut vérifier, sur la machine serveur, que le fichier a bien été créé an affichant le contenu du répertoire `/tmp` :

```
$ ls /tmp
```

◆

Q 5.4 *Connexion à distance* Depuis les machines clientes, ouvrir une connexion ssh sur le serveur sous l'identité etudiant. Quel est le répertoire courant à l'ouverture de la session sur la machine serveur ? Effectuer ensuite ces deux tests dans la session ouverte par ssh :

(a) Utiliser la commande `wall` suivie d'une phrase quelconque (ex : `wall salut à toi`) puis vérifier dans le terminal de la machine serveur que cette phrase a bien été affichée.

(b) Créer un fichier quelconque dans le répertoire `/tmp` de la machine serveur. Vérifier ensuite dans le terminal de la machine serveur que le fichier a bien été créé.

Correction

Sur les machines clientes :

```
$ ssh etudiant@10.1.0.130
```

On doit alors saisir le mot de passe de etudiant. Une fois connecté on peut connaître le répertoire courant avec la commande pwd :

```
$ pwd
/home/etudiant
```

La session est ouverte sur le répertoire personnel de l'utilisateur qui s'est connecté à distance.
Ensuite toujours dans la session ssh :

```
$ wall salut à toi
$ touch /tmp/quelconque
```

Sur la machine serveur on devrait voir `salut à toi` s'afficher. On peut vérifier que le fichier `/tmp/quelconque` a bien été créé an affichant le contenu du répertoire `/tmp` :

```
$ ls /tmp
```

◆

Q 5.5 *Affichage des connexions ouvertes* Proposer une commande combinant `ps aux` avec un tube afin de n'afficher que les lignes contenant le mot `sshd`. Exécuter cette commande sur la machine serveur. Pour chaque connexion ouverte, on devrait voir un processus dont la commande est `sshd: etudiant@pts/XXX`. Ce processus est le terminal ouvert chez le client. Proposer puis exécuter une commande permettant d'interrompre une des deux connexions ouvertes. Vérifier sur la machine cliente correspondant que la connexion a bien été interrompue.

Correction

On redirige le résultat de la commande `ps aux` vers la commande `grep` qui va sélectionner les lignes contenant la chaîne `sshd` :

```
$ ps aux | grep sshd
root       4640  0.0  0.0  44124  2640 ?        Ss   13:27   0:00 /usr/sbin/sshd -D
root       5718  0.0  0.0  95908  3900 ?        Ss   13:34   0:00 sshd: etudiant [priv]
etudiant   5726  0.0  0.0  95908  2036 ?        S    13:34   0:00 sshd: etudiant@pts/2
root       5791  0.0  0.0  95908  3908 ?        Ss   13:35   0:00 sshd: etudiant [priv]
etudiant   5797  0.0  0.0  96060  2288 ?        S    13:35   0:00 sshd: etudiant@pts/3
```

On peut interrompre un processus avec la commande kill suivie du pid du processus à interrompre. On exécute donc sur la machine serveur :

```
# kill 5726
```

Cela ferme la session ouverte par le premier client. ◆

Q 5.6 *Fermeture de la session SSH* Fermer la session ssh ouverte sur l'autre machine cliente en utilisant la commande `exit` ou en tappant Ctrl+D.

Mesurer le débit/temps de transfert : iperf La commande `iperf` est utilisée pour mesurer les performances d'un réseau. Elle permet de mesurer le temps de transfert entre deux machines. Une des deux machines doit jouer le rôle de serveur. Elle recevra les données envoyées par la deuxième machine qui jouera le rôle de client.

Sur la machine serveur, il faut d'abord démarrer `iperf` en lui passant l'option `-s` pour passer en mode serveur. `iperf` attend alors des connexions de clients et affiche un message à chaque fois qu'un client se connecte.

Sur la machine cliente, il faut ensuite appeler `iperf` avec l'option `-c <ip-du-serveur>` pour passer en mode client et préciser le serveur à contacter et avec l'option `-n <nb-d-octets>` pour préciser le nombre d'octets à transférer. Les données transférées et n'ont aucune signification.

Obtenir/modifier les caractéristiques d'une liaison : ethtool On peut appeler la commande `ethtool` en lui passant en argument le nom d'une interface pour qu'elle affiche les caractéristiques de transmission de cette interface, comme ceci :

```
# ethtool <nom-de-l-interface>
```

Une liaison Ethernet est caractérisée par un code de la forme `DCT` où
— D est le débit de la liaison en Mbit/s ;
— C est la méthode de codage utilisée pour envoyer les signaux sur le câble (généralement `base` pour du codage en bande de base, ce qui correspond à des signaux carrés) ;
— T donne des informations sur le type de câble utilisé ainsi que sur le mode de transmission (full ou half duplex).
Par exemple :
— 10baseT-FD = débit de 10 Mbit/s, avec codage en bande de base sur de la paire torsadée en full duplex
— 10baseT-HD = même chose mais en half duplex
— 100base-FX = débit de 100 Mbit/s, avec codage en bande de base sur de la fibre optique

On peut également modifier ces caractèristiques en utilisant l'option `-s` (pour set) suivie des nouvelles caractéristiques :

```
# ethtool -s <nom-de-l-interface> <nouvelles-caracteristiques>
```

Les principales caractéristiques que l'on peut modifier sont le débit en Mbit/s (paramètre `speed <nouveau-debit-en-Mbit/s>`) et le mode (paramètre `duplex <full-ou-half>`).

TP 2 — NFS

Le TP est à faire en binôme. Chaque binôme rendra à l'enseignant un compte-rendu dans lequel seront notées les réponses aux questions, les observations éventuelles ainsi que les commandes exécutées en recopiant, si nécessaire, les sorties de ces commandes (ce que la commande affiche dans le terminal).

L'objectif de ce TP est de configurer un serveur NFS. Nous allons dans un premier temps configurer le serveur sur lequel des fichiers seront accessibles à des clients. Nous allons ensuite configurer un client pour qu'il puisse accéder à ces données.

Dans le TP nous n'utiliserons que l'interface eth0 des machines, celle qui connecte les machines aux réseaux de l'IUT et à Internet.

Exercice 1 — Nommage des machines

Q 1.1 *Associations nom↔adresse IP* Dans quel fichier trouve-t-on les correspondances entre adresses IP et noms de machine déjà connues ? Modifier ce fichier sur les deux machines afin que le serveur associe le nom `client-nfs` à l'adresse IP de la machine cliente et que le client associe le nom `serveur-nfs` à l'adresse IP de la machine serveur.

Dans la suite du TP, nous n'utiliserons plus les adresses IP des machines, mais uniquement leurs noms.

Correction

C'est dans le fichier `/etc/hosts` que l'on peut rajouter une correspondance entre une adresse IP et un nom de machine. Dans un premier temps il faut récupérer l'adresse IP de l'autre machine en utilisant la commande `ifconfig eth0` sur cette autre machine. En supposant que l'adresse du client est 192.168.43.2 et que celle du serveur est 192.168.43.1 il faut insérer la ligne suivante dans le fichier `/etc/hosts` du serveur :

```
192.168.43.2 client-nfs
```

et celle-ci dans le fichier `/etc/hosts` du client :

```
192.168.43.1 serveur-nfs
```

Q 1.2 *Vérification des associations* Effectuer des ping entre les deux machines pour vérifier que les associations nom↔adresse IP faites à la question précédente sont effectives.

Correction

Depuis le client :

```
$ ping serveur-nfs
```

Exercice 2 — Configuration du serveur

Q 2.1 *Le service NFS* Pour utiliser NFS, côté serveur, il existe un service `nfs-server` qui traite les requêtes NFS. À chaque action possible sur un fichier/répertoire stocké sur le serveur correspond une requête NFS spécifique (lecture, supression, ...).

Démarrer ce service en tant que root puis vérifier qu'il est actif.

Correction

Pour démarrer le service `nfs-server` puis vérifier qu'il est actif :

```
$ systemctl start nfs-server.service
$ systemctl status nfs-server.service
```

Q 2.2 *Création des répertoires exportés* Si un répertoire sur le serveur peut être accédé à distance par un client on dit qu'il est *exporté*. Créér, en tant qu'administrateur, les répertoires suivants :

— [] `/nfs`
— [] `/nfs/etudiant`

— [] `/nfs/etudiant/perso`
— [] `/nfs/etudiant/public`

Attribuer ensuite à l'utilisateur etudiant le répertoire `/nfs/etudiant` ainsi que le contenu de ce répertoire.

Correction

On exécute, en tant que root, les commandes suivantes :

```
$ mkdir /nfs
$ mkdir /nfs/etudiant
$ mkdir /nfs/etudiant/perso
$ mkdir /nfs/etudiant/public
```

Puis pour attribuer le répertoire `/nfs/etudiant` à étudiant :

```
$ chown -R etudiant /nfs/etudiant
```

L'option `-R` permet d'attribuer à etudiant les sous-répertoires et fichiers se trouvant dans `/nfs/etudiant` ainsi que tous les sous-sous-répertoires, et ainsi de suite. ◆

Q 2.3 *Export des répertoires* Le fichier `/etc/exports` placé sur le serveur contient la liste des répertoires exportés avec un répertoire par ligne. La syntaxe de chaque ligne est la suivante :

`répertoire machine(options)`

où

— `répertoire` est le chemin absolu (commençant par '/') sur le serveur du répertoire exporté.
— `machine` est le nom ou l'adresse IP de la machine à partir de laquelle le répertoire peut être accédé. On peut utiliser le caractère `*` pour désigner toutes les machines, ou un nom de la forme `*.domaine.fr` pour désigner toutes les machines du domaine `domaine.fr`. Enfin, on peut désigner les machines d'un réseau IP en donnant l'adresse du réseau et son masque en notation '/' (p.ex., 10.0.0.0/24).
— `options` sont les options de montage du répertoire, c'est-à-dire la manière dont le client accède au répertoire (en lecture/écriture, en lecture uniquement, ...).

Modifier ce fichier afin que les répertoires soient exportés avec les options suivantes :
— `/nfs/etudiant/public` accessible par toutes les machines de votre réseau en lecture seulement.
— `/nfs/etudiant/perso` accessible en lecture/écriture par la machine cliente uniquement.

Consulter le manuel du fichier `exports` (paragraphe *General Options*) pour connaître les options de montage à utiliser. Utiliser ensuite la commande `exportfs -a` pour que l'export soit effectif. Tester enfin si l'export est effectif avec la commande `showmount -e`.

Remarque. Après chaque modification de ce fichier, il faudra relancer la commande `exportfs -a` pour que les modifications soient prises en compte.

Correction

L'option pour exporter un répertoire en lecture seule est ro et celle pour exporter en lecture/écriture est rw. On doit donc insérer deux lignes dans `/etc/exports`, une pour chaque répertoire. Le réseau de la salle de TP a normalement l'adresse 192.168.43.0/24. Le contenu du fichier est donc :

```
/nfs/etudiant/public 192.168.43.0/24(ro)
/nfs/etudiant/perso  client-nfs(rw)
```

On peut ensuite exécuter les deux commandes suivantes :

```
$ exportfs -a
$ showmount -e
Export list for q20301:
/nfs/etudiant/public 192.168.43.0/24
/nfs/etudiant/perso  client-nfs
```

La commande `showmount` montre les deux répertoires exportés avec les machines qui peuvent y accéder (mais sans afficher les droits). ◆

Exercice 3 — Configuration du client

Sur la machine cliente, l'accès aux répertoires placés sur le serveur se fait en associant un répertoire local à un répertoire sur le serveur. On dit que le répertoire sur le serveur est *monté* sur le répertoire local appelé *point de montage*.

Q 3.1 *Création des points de montage* Créer deux répertoires `nfs-public` et `nfs-perso` dans le répertoire du travail de etudiant.

Correction

Pour créer les répertoires on exécute, en tant qu'utilisateur etudiant :

```
$ mkdir ~/nfs-public
$ mkdir ~/nfs-perso
```

◆

Q 3.2 *Montage des répertoires du serveur* La commande `mount` permet de monter un répertoire via NFS. Elle doit être utilisée comme ceci :

```
mount -t type serveur:repertoire point
```

où

— `type` est le type de système de fichiers monté (`nfs` dans notre cas).
— `serveur` est le serveur NFS depuis lequel on monte le répertoire.
— `repertoire` est le répertoire exporté (sur le serveur).
— `point` est le point de montage (sur le client).

Effectuer les deux montages suivants :

— `/nfs/etudiant/public` monté en lecture seule sur le répertoire `nfs-public` du répertoire de travail de etudiant
— `/nfs/etudiant/perso` monté en lecture/écriture sur le répertoire `nfs-perso` du répertoire de travail de etudiant

Correction

Les deux commandes de montage (à exécuter sur le client en tant que root) sont :

```
$ mount -t nfs serveur-nfs:/nfs/etudiant/public /home/etudiant/nfs-public
$ mount -t nfs serveur-nfs:/nfs/etudiant/perso  /home/etudiant/nfs-perso
```

On précise ainsi que le contenu du répertoire `/nfs/etudiant/public` sur la machine `serveur-nfs` est accessible depuis le répertoire `/home/etudiant/nfs-public` de la machine cliente. ◆

Q 3.3 *Vérification du montage* Vérifier à l'aide de la commande `df` que le montage fonctionne. Quelles sont les informations affichées par cette commande ?

Correction

Pour vérifier que les montages se sont bien passés on exécute :

```
$ df
Sys. de fichiers                 Taille Utilisé Dispo Uti% Monté sur
rootfs                              11G    2,8G  7,2G  28% /
devtmpfs                           1,9G       0  1,9G   0% /dev
tmpfs                              1,9G       0  1,9G   0% /dev/shm
tmpfs                              1,9G    396K  1,9G   1% /run
/dev/sda1                           11G    2,8G  7,2G  28% /
tmpfs                              1,9G       0  1,9G   0% /sys/fs/cgroup
none                               1,9G    202M  1,7G  11% /tmp
serveur-nfs:/nfs/etudiant/public    11G    2,8G  7,2G  28% /home/etudiant/nfs-public
serveur-nfs:/nfs/etudiant/perso     11G    2,8G  7,2G  28% /home/etudiant/nfs-perso
```

Les deux dernières lignes indiquent que les deux répertoires du serveur ont bien été montés sur les répertoires `/home/etudiant/nfs-public` et `/home/etudiant/nfs-perso` de la machine cliente. ◆

Exercice 4 — Test de l'export

Q 4.1 *Création de fichiers sur le serveur* Sur la machine serveur, créer, en tant que etudiant :

— le fichier `/nfs/etudiant/public/info.txt` contenant le texte `répertoire public de etudiant`
— et le fichier `/nfs/etudiant/perso/donnees.txt` contenant le texte `privé`.

Correction

Sur le serveur, on exécute en tant qu'utilisateur etudiant :

```
$ echo "répertoire public de etudiant" > /nfs/etudiant/public/info.txt
$ echo "privé" > /nfs/etudiant/perso/donnees.txt
```

◆

Q 4.2 *Vérification de l'accès par le client* Sur la machine cliente, essayer en tant qu'etudiant les opérations suivantes et commenter :

(a) afficher le contenu du répertoire `~/nfs-public`
(b) lire le fichier `~/nfs-public/info.txt`
(c) créer un fichier `~/nfs-public/test.txt`
(d) créer un fichier `~/nfs-perso/test.txt`
(e) supprimer le fichier `~/nfs-perso/donnees.txt`

Correction

Voici les cinq commandes à exécuter :

```
$ ls ~/nfs-public/
info.txt
$ cat ~/nfs-public/info.txt
répertoire public de etudiant
$ touch ~/nfs-public/test.txt
touch: impossible de faire un touch: Système de fichiers accessible en lecture seulement
$ touch ~/nfs-perso/test.txt
$ rm ~/nfs-perso/donnees.txt
```

Les deux premières opérations permettent de retrouver les fichiers qui avaient été créés sur le serveur. La troisième opération n'est pas réalisable car le répertoire `/nfs/etudiant/public` de la machine serveur à été exporté et monté en lecture seule. Par contre, les deux dernières opérations sont réalisables car le répertoire `/nfs/etudiant/perso` sur le serveur est exporté en lecture/écriture. ◆

Exercice 5 — Démontage et montage automatique

Q 5.1 *Démontage des répertoires* À l'aide de la commande `umount`, démonter en tant que root, sur la machine cliente, les répertoires montés. Afficher ensuite le contenu de ces répertoires, que constate-t-on ?

Correction

Pour utiliser `umount` (toujours en tant que root) il suffit de préciser le point de montage actuel à démonter :

```
$ umount /home/etudiant/nfs-public
$ umount /home/etudiant/nfs-perso
$ ls /home/etudiant/nfs-public
$ ls /home/etudiant/nfs-perso
```

En affichant le contenu de ces répertoires, on s'aperçoit qu'ils sont vides. Ces deux répertoires ne sont plus associés aux répertoires du serveur et on ne retrouve donc plus les fichiers qui sont sur le serveur (comme `info.txt` ou `test.txt`). On retrouve leurs contenus avant le montage. C'est pourquoi ils sont vides. ◆

Q 5.2 *Montage automatique au démarrage* Le fichier `/etc/fstab` contient la liste des points de montage possibles. Il est analysé au démarrage du système pour monter des répertoires. Modifier ce fichier à l'aide de la page de manuel de `fstab`, pour que le montage des deux répertoires soit automatique au démarrage. On peut ensuite utiliser la commande `mount` plus simplement en précisant uniquement le point de montage : `mount point`. Remonter le répertoire `/nfs/etudiant/public` en utilisant `mount` de cette façon. Redémarrer (en fin de séance) la machine pour vérifier que les deux répertoires sont montés.

Correction

Chaque ligne du fichier `/etc/fstab` doit contenir au moins quatre colonnes :

```
serveur:repertoire-monté  point-de-montage  type  options
```

On retrouve exactement les arguments de la commande `mount` lors du montage. L'option à indiquer pour que le montage soit automatique au démarrage est `auto`. Les deux lignes à rajouter en tant que root dans le fichier `/etc/fstab` du client seront donc :

```
serveur-nfs:/nfs/etudiant/public /home/etudiant/nfs-public nfs ro,auto
serveur-nfs:/nfs/etudiant/perso  /home/etudiant/nfs-perso  nfs rw,auto
```

On peut ensuite, en tant que root, monter un répertoire en indiquant uniquement le point de montage. La commande `mount` retrouvera dans ce fichier les informations pour effectuer le montage.

```
$ mount /home/etudiant/nfs-public
$ mount /home/etudiant/nfs-perso
```

◆

Exercice 6 — Accès au serveur en tant que root

On veut maintenant que le répertoire `/nfs/etudiant/perso` du serveur puisse être modifiable par l'utilisateur root de la machine cliente, autrement dit qu'un utilisateur connecté en tant que root sur le client soit également considéré comme root sur le serveur quand il monte le répertoire `/nfs/etudiant/perso`. On va voir dans un premier temps que ce n'est pas possible avec la configuration actuelle.

Q 6.1 *Accès par l'utilisateur root depuis le client* Sur la machine cliente, créer, en tant que root, un répertoire `/root/etudiant`. Monter ensuite en lecture/écriture le répertoire `/nfs/etudiant/perso` du serveur sur ce nouveau répertoire. Enfin, essayer de supprimer le fichier `test.txt` se trouvant dans le répertoire `perso` du serveur. Que constate-t-on ? Démonter ensuite le répertoire.

Correction

En tant que root sur la machine cliente :

```
$ mkdir /root/etudiant
$ mount -t nfs serveur-nfs:/nfs/etudiant/perso /root/etudiant
$ rm /root/etudiant/test.txt
```

Un message d'erreur indiquant que l'utilisateur n'a pas les droits nécessaires est affiché lors de la suppression.
Puis pour démonter le répertoire :

```
$ umount /root/etudiant
```

◆

Q 6.2 *L'option `root_squash`* Consulter la page de manuel de `exports` (section *User ID Mapping*) pour obtenir de l'aide sur l'option `root_squash`. Cette option est utilisée par défaut lors de l'export d'un répertoire. Que signifie-t-elle ? Expliquer pourquoi la suppression de la question précédente n'a pas marché.

Correction

L'option `root_squash` signifie que lorsqu'un client accède à un répertoire monté via NFS et qu'il est connecté en tant que root sur la machine cliente, toutes les requêtes envoyées vers le serveur NFS sont en fait exécutées avec l'identité de l'utilisateur anonymous. Cette option est activée par défaut pour des raisons de sécurité : même si un client peut se connecter en tant que root sur sa machine il ne sera pas root sur le serveur. ◆

Q 6.3 *Activation de l'accès à root* Sur le serveur, ajouter dans le fichier d'export l'option `no_root_squash` aux options de montage du répertoire `/nfs/etudiant/perso`. Refaire ensuite le test de la question Q 6.1. Expliquer en quoi l'ajout de cette option a résolu le problème observé précédemment.

Correction

La ligne du fichier `/etc/exports` du serveur doit maintenant contenir la ligne suivante :

```
/nfs/etudiant/perso  client-nfs(rw,no_root_squash)
```

Avec l'option `no_root_squash`, les requêtes envoyées vers le serveur NFS par un client connecté en tant que root sont bien exécutées sous l'identité root sur le serveur. La suppression du fichier est donc possible. ◆

TP 3 — Routage et DHCP

Le TP est à faire par groupe de 3. Chaque trinôme rendra à l'enseignant un compte-rendu dans lequel seront notées les réponses aux questions, les observations éventuelles ainsi que les commandes exécutées en recopiant, si nécessaire, les sorties de ces commandes (ce que la commande affiche dans le terminal).

L'objectif de ce TP est de réaliser le réseau de la Figure 1 composé de trois machines. R jouera le rôle de routeur entre M1 et M2. *N* est à remplacer par le numéro du groupe qui vous sera donné par l'enseignant. Dans un premier temps, les interfaces et les tables de routage seront configurées statiquement. Dans un second temps le routeur R jouera également le rôle de serveur DHCP pour configurer automatiquement les interfaces et les tables de routage de M1 et M2.

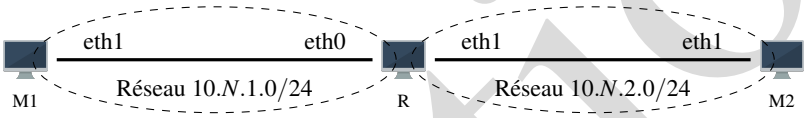

FIGURE 1 – Le réseau à réaliser

Exercice 1 — Configuration statique du réseau

Q 1.1 *Désactivation des interfaces* Désactiver les deux interfaces sur les trois ordinateurs puis vérifier que les tables de routage sont vides.

Correction

Sur M1, M2 et R :

```
$ ifdown eth0
$ ifdown eth1
```

La commande route affiche une table vide. ◆

Q 1.2 *Câblage* Réaliser le câblage de la Figure 1. Quel type de câble faut-il utiliser pour connecter deux ordinateurs ?

Correction

On utilise un câble croisé pour connecter deux ordinateurs. ◆

Q 1.3 *Configuration des interfaces* Choisir des adresses IP pour les quatres interfaces utilisées dans notre réseau. Configurer ensuite les interfaces en modifiant leurs fichiers de configuration pour leur attribuer les adresses IP choisies.

Correction

On choisit les adresses IP suivantes :
- eth1 de M1 : 10.1.1.1
- eth0 de R : 10.1.1.254
- eth1 de R : 10.1.2.254
- eth1 de M2 : 10.1.2.1

Il faut modifier les fichiers de configuration du répertoire `/etc/sysconfig/network-scripts`. Par exemple, pour M1, le fichier `/etc/sysconfig/network-scripts/ifcfg-eth1` doit contenir les lignes suivantes :

```
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.1.1.1
NETMASK=255.255.255.0
```

Sur R il faut modifier deux fichiers : celui de eth0 et celui de eth1. ◆

Q 1.4 *Activation des interfaces* Activer toutes les interfaces utilisées sur la Figure 1. Vérifier ensuite que les adresses IP et les masques ont été correctement attribués.

Correction

Sur M1 et M2 :

```
$ ifup eth1
```

```
$ ifconfig eth1
```

Sur R :

```
$ ifup eth0
$ ifup eth1
$ ifconfig eth0
$ ifconfig eth1
```

La commande ifconfig affiche les adresses IP et les masques. ◆

Q 1.5 *Test de la connexion* Vérifier que les deux connexions (M1↔R et M2↔R) sont actives.

Correction

Sur M1 :

```
$ ping -c 1 10.1.1.254
```

et sur M2 :

```
$ ping -c 1 10.1.2.254
```

◆

Q 1.6 *Tables de routage* Que contiennent les tables de routage des trois machines après l'activation des interfaces ? M1 peut-il pinguer M2 ? Pourquoi ?

Correction

Sur R :

```
$ route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
10.1.1.0        *               255.255.255.0   U     0      0        0 eth0
10.1.2.0        *               255.255.255.0   U     0      0        0 eth1
```

Les lignes ont été rajoutées à l'activation des deux interfaces. Chaque ligne correspond à une remise directe sur le réseau de l'adresse IP configurée.
Sur M1 :

```
$ route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
10.1.1.0        *               255.255.255.0   U     0      0        0 eth1
```

La table de routage n'indique qu'une seule destination : le réseau 10.1.1.0. Pour que M1 puisse envoyer des paquets à M2 il faut une ligne supplémentaire : soit une route par défaut indiquant comment joindre tout réseau autre que 10.1.1.0 ; soit une route contenant le réseau de M2 (10.1.2.0) comme destination. On peut le voir en essayant de pinguer M2 :

```
$ ping -c 1 10.1.2.1
connect: Network is unreachable
```

La machine ne sait pas comment joindre le réseau de destination (Network is unreachable = Le réseau n'est pas joignable) : il n'y a aucune ligne de la table de routage qui permette d'atteindre le réseau de la machine 10.1.2.1. ◆

Q 1.7 *Configuration de R comme routeur* La commande sysctl permet de visualiser ou de modifier certains paramètres réseau (et d'autres). Voici deux façons de l'utiliser :
— pour voir la valeur d'un paramètre : `sysctl <nom-du-parametre>`
— pour modifier la valeur d'un paramètre : `sysctl <nom-du-parametre>=<valeur>`
Le paramètre `net.ipv4.ip_forward` indique ce que la machine doit faire lorsqu'elle reçoit un paquet qui ne lui est pas destiné. S'il vaut 1 alors elle essaiera de le router. S'il vaut 0 le paquet sera ignoré. Changer ce paramètre afin que R accepte de router les paquets.

Correction

Sur R :

```
$ sysctl net.ipv4.ip_forward=1
```

◆

Q 1.8 *Ajout de routes* Proposer et ajouter dans la table de routage de M1 une route qui lui permette d'envoyer des paquets à M2. Réaliser l'opération symétrique sur M1.

Correction

On a deux solutions pour que M1 et M2 puissent s'échanger des paquets.
Solution 1 : rajouter des routes par défaut, ce qui donne sur M1 :

```
$ route add default gw 10.1.1.254 dev eth1
```

et sur M2 :

```
$ route add default gw 10.1.2.254 dev eth1
```

Solution 2 : rajouter sur M1 et M2 une route vers le réseau de l'autre machine, ce qui donne sur M1 :

```
$ route add -net 10.1.2.0 netmask 255.255.255.0 gw 10.1.1.254 dev eth1
```

et sur M2 :

```
$ route add -net 10.1.1.0 netmask 255.255.255.0 gw 10.1.2.254 dev eth1
```

La commande route affiche ensuite les tables modifiées avec la nouvelle destination. ◆

Q 1.9 *Test de la connexion* Vérifier que M1 est maintenant accessible depuis M2 et vice versa.

Correction

Depuis M1 :

```
$ ping -c 1 10.1.2.1
```

et depuis M2 :

```
$ ping -c 1 10.1.1.1
```

◆

Q 1.10 *Désactivation des interfaces* Désactiver eth1 sur M1 et M2. Vérifier que les interfaces eth1 de M1 et M2 n'ont plus adresses IP associées et que leurs tables de routages sont vides.

Correction

Sur M1 et M2 :

```
$ ifdown eth1
$ ifup eth1
```

On vérifie avec `ifconfig eth1` et `route -n`. ◆

Exercice 2 — Configuration du réseau via DHCP

On souhaite maintenant que les interfaces eth1 et les tables de routage de M1 et M2 soient configurées grâce au protocole DHCP. C'est R qui jouera le rôle du serveur DHCP et accordera des *baux* aux clients M1 et M2. Un bail (lease en anglais) est une adresse IP attribuée pour une certaine durée au client. Ce délai passé, le client ne peut plus utiliser l'adresse.

Q 2.1 *Création du fichier de configuration du serveur* Le fichier `/etc/dhcpd.conf` permet de configurer le service dhcpd pour indiquer les réponses qu'il fera aux clients DHCP. Le fichier ci-dessous est un exemple simple de fichier `/etc/dhcpd.conf`. Le caractère # permet de commencer un commentaire qui se termine à la fin de la ligne.

```
subnet 1.2.0.0 netmask 255.255.0.0 {  # l'adresse du réseau et son masque
   max-lease-time 60;                 # durée des baux en secondes
   option routers 1.2.255.254;        # l'adresse du routeur sur le réseau qui mène à l'extérieur
   range 1.2.0.1 1.2.255.253;         # la plage des adresses que le serveur
}                                     # peut attribuer aux clients
```

Une section **subnet** permet de configurer les réponses que fera le serveur DHCP aux machines d'un réseau particulier. Pour qu'une section soit correcte il faut qu'une interface du serveur DHCP soit associée à une adresse IP de ce réseau. Dans notre exemple, il faut donc que le serveur DHCP ait une interface avec une adresse IP de la forme 1.2.*X*.*Y*.
Créer, à partir de cet exemple, le fichier `/etc/dhcpd.conf` de R en supprimant au préalable le fichier existant. Le fichier devra contenir deux sections **subnet** : une par réseau auquel R est connecté. On considérera que les deux réseaux ont des caractéristiques différentes :

	Réseau 10.*N*.1.0	Réseau 10.*N*.2.0
Nombre max. de clients pouvant se connecter	5	100
Durée des baux	1 journée	1 heure

Correction

Le fichier `/etc/dhcpd.conf` de R :

```
subnet 10.1.1.0 netmask 255.255.255.0
{
   max-lease-time 86400;
   option routers 10.1.1.254;
   range 10.1.1.101 10.1.1.105;
}
subnet 10.1.2.0 netmask 255.255.255.0
{
   max-lease-time 3600;
   option routers 10.1.2.254;
   range 10.1.2.101 10.1.2.200;
}
```

Q 2.2 *Lancement du service* Lancer le service dhcpd sur R. En cas d'erreur ou de modification dans le fichier de configuration, le service devra être relancé.

Correction

Sur R :

```
$ systemctl start dhcpd.service
$ systemctl status dhcpd.service
```

Q 2.3 *Configuration des interfaces de M1 et M2* Changer le fichier de configuration de l'interface eth1 sur M1 et M2 afin que l'interface soit maintenant configurée via DHCP. Quels paramètres sont maintenant inutiles dans ces fichiers ?

Correction

Le fichier `/etc/sysconfig/network-scripts/ifcfg-eth1` des machines M1 et M2 doit maintenant contenir :

```
DEVICE=eth1
BOOTPROTO=dhcp
ONBOOT=yes
```

Il est maintenant inutile de préciser l'adresse IP ou le masque : l'interface obtiendra ces informations directement à partir du serveur DHCP.

Q 2.4 *Activation des interfaces* Activer eth1 sur M1 et M2. Vérifier que les adresses IP et les tables de routage de M1 et M2 ont bien été configurées par le serveur. Vérifier également que toutes les informations envoyées par le serveur ont bien été sauvegardées par le client dans le fichier `/var/lib/dhclient/dhclient--eth1.lease`.

Correction

Sur M1 et M2 :

```
$ ifdown eth1
$ ifup eth1
```

L'adresse IP de M1 affichée par ifconfig est normalement 10.1.1.101, le serveur DHCP attribuant les adresses IP séquentiellement à partir de la première adresse disponible.
Pour M1, la commande route affiche deux lignes :

```
$ route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
0.0.0.0         10.1.1.254      0.0.0.0         UG    0      0        0 eth1
10.1.1.0        0.0.0.0         255.255.255.0   U     0      0        0 eth1
```

La table contient une ligne avec la destination 10.1.1.0 en remise directe et une route par défaut avec 10.1.1.254 comme routeur. L'option routers du fichier de configuration permet donc de donner une route par défaut aux clients.

Q 2.5 *Test de la connexion* Vérifier que M1 est toujours accessible depuis M2 et vice versa.

Correction

Depuis M1 :

```
$ ping -c 1 10.1.2.101
```

et depuis M2 :

```
$ ping -c 1 10.1.1.101
```

◆

Q 2.6 *Analyse du journal* Le fichier `/var/log/messages`, appelé fichier de journal, est utilisé par le système d'exploitation pour enregistrer des événements importants, et particulièrement les événements en rapport avec les services. Après l'activation des interfaces il devrait contenir les messages DHCP échangés entre le serveur et les clients. Quels sont-ils ?

Correction

En utilisant la commande `grep dhcpd /var/log/messages` on peut retrouver les messages échangés :

(a) client → diffusion : DISCOVER
(b) serveur → client : OFFER
(c) client → diffusion : REQUEST
(d) serveur → client : ACK

◆

Exercice 3 — Renouvellement du bail

L'objectif de cet exercice est d'analyser le fonctionnement du renouvellement du bail.

Q 3.1 *Modification de la durée des baux* Modifier le fichier de configuration du serveur pour que la durée des baux soit de 1 minute seulement sur le réseau 10.*N*.2.0.

Correction

Il suffit de changer la valeur de **max-lease-time** par 60 dans le bloc **subnet** du deuxième réseau. On redémarre ensuite le service dhcpd sur R.

◆

Q 3.2 *Analyse du renouvellement* Désactiver puis réactiver eth1 sur M2. Consulter le fichier `/var/log/messages` du serveur après une minute.

(a) Par quels échanges de messages DHCP se fait le renouvellement du bail ?
(b) À quel intervalle ces messages sont-il échangés ?
(c) Sachant que le serveur peut refuser le renouvellement, pourquoi envoyer la demande avant l'expiration du bail ?
(d) Quel est l'inconvénient d'accorder des baux trop court ?

Correction

(a) Le client envoie un message REQUEST au serveur qui répond par un ACK.
(b) Environ toutes les 30 secondes (la moitié de la durée du bail) le client envoie son message REQUEST.
(c) Si le serveur répond négativement (si il ne veut pas renouveler le bail) le client peut alors demander à un autre serveur DHCP et ne se retrouve pas immédiatement sans adresse IP.
(d) Le réseau est alors encombré de messages REQUEST et ACK utilisés lors du renouvellement.

◆

Exercice 4 — Attribution d'adresses IP fixes

On imagine que M1 est un serveur NFS et on voudrait par commodité que son interface eth1 ait toujours la même adresse IP. Or, dans la configuration actuelle, rien ne garantit que ce sera le cas. Il existe un moyen de préciser dans le fichier `/etc/dhcpd.conf` qu'une interface particulière aura une adresse IP fixe. Consulter la page de manuel de *dhcpd.conf* (en cherchant les déclarations *host* ou *fixed-address*) ou rechercher sur Internet (en réactivant temporairement eth0 sur M1 ou M2) pour trouver comment faire. Choisir ensuite une adresse IP fixe pour M1 et mettre en œuvre cette solution puis tester.

Correction

Afin que le serveur reconnaisse la requête envoyée par M1 il faut qu'il y trouve l'*adresse MAC* de M1. Toute carte ethernet a une adresse MAC qui lui est propre et qui permet de l'identifier. Cette adresse est gravée sur la carte et apparaît dans tous les messages envoyés par la carte. Ainsi le serveur DHCP peut reconnaître une demande qui vient de M1. Une adresse MAC est composée de 6 couples de caractères hexadécimaux séparés par le caractère ' :', par exemple : `A1:B2:C3:D4:E5:F6`. On peut connaître l'adresse MAC d'une interface grâce à la commande ifconfig :

```
$ ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:0A:F7:16:C1:68
UP BROADCAST MULTICAST  MTU:1500  Metric:1
```

```
RX packets:1881 errors:0 dropped:0 overruns:0 frame:0
TX packets:1896 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1000
RX bytes:189231 (184.7 KiB)  TX bytes:152414 (148.8 KiB)
Interruption:18
```

HWaddr signifie *hardware address* ou adresse matérielle.

Si l'on connaît l'adresse MAC de l'interface à laquelle on veut attribuer une adresse IP fixe, il faut ensuite rajouter la déclaration suivante dans le fichier `/etc/dhcpd.conf` :

```
#  déclaration de la machine serveur-nfs (nom utilisé uniquement par l'administrateur)
host serveur_nfs {

   #  adresse MAC de l'interface
   hardware ethernet 00:0A:F7:16:C1:68;

   #  adresse IP fixe qui lui est associée
   fixed-address 10.1.1.50;
}
```

En désactivant puis en réactivant eth1 sur M1 on devrait ensuite voir avec ifconfig que l'adresse IP qui lui est associée est 10.1.1.50.◆

TP 4 — DNS

Le TP est à faire en binôme sur trois machines. Chaque binôme rendra à l'enseignant un compte-rendu dans lequel seront notées les réponses aux questions, les observations éventuelles ainsi que les commandes exécutées en recopiant, si nécessaire, les sorties de ces commandes (ce que la commande affiche dans le terminal).

L'objectif de ce TP est de mettre en place un serveur DNS simple et d'étudier les interactions entre un serveur DHCP et un serveur DNS. On réalisera le réseau de la Figure 2 dans lequel la machine appelée dns jouera le rôle de serveur DNS pour les deux autres machines. La machine appelée dhcp est le serveur DHCP du réseau. Enfin la machine util est la machine d'un utilisateur. Ce réseau a l'adresse 10.0.0.0/8 et sera associé au domaine `dut.edu`. Les noms complets des trois machines seront donc `dns.dut.edu`, `dhcp.dut.edu` et `util.dut.edu`. Tous les exercices sont à faire en tant que root. Vous trouverez en annexes (page 24) la description et la syntaxe des fichiers que l'on créera sur les serveurs DNS et DHCP.

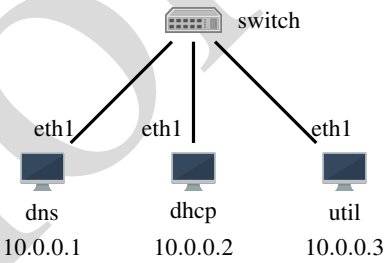

FIGURE 2 – Le réseau à réaliser

Fichiers de configuration Sur la machine dns deux fichiers seront créés :

— `/etc/named.conf` — C'est le fichier de configuration du service DNS named programmé par l'université de Berkeley et présent sur la plupart des systèmes Linux. Ce fichier contient la liste des zones sur lesquelles le serveur a des informations (p.ex., celles sur lesquelles il fait autorité).

— `/var/named/dut.edu` — C'est le fichier de la zone `dut.edu`. Il contient, entre autres, les adresses IP des trois machines de la zone.

Sur la machine dhcp, nous travaillerons sur le fichier `/etc/dhcpd.conf` de configuration du service DHCP. On partira d'un fichier qui permet d'attribuer aux machines dns et util une adresse IP fixe en fonction de leurs adresses MAC.

Après toute modification sur un de ces fichiers, le service correspondant devra être redémarré pour que les modifications soient prises en compte.

Exercice 1 — Configuration des interfaces via le serveur DHCP

Q 1.1 Modifier le fichier de configuration de l'interface eth1 de dhcp pour que celle-ci soit configurée statiquement avec l'adresse IP indiquée sur la Figure 2. Activer ensuite cette interface.

Correction

Le fichier `/etc/sysconfig/network-scripts/ifcfg-eth1` doit contenir :

```
DEVICE=eth1
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.0.0.2
NETMASK=255.0.0.0
```

Q 1.2 Utiliser la commande `wget` pour récupérer dans le terminal ouvert sur la machine dhcp le fichier à l'adresse suivante :

`http://www.lipn.univ-paris13.fr/~evangelista/cours/M1101/TP4/dhcpd.conf`

Placer ensuite ce fichier dans le répertoire `/etc`. Nous partirons de ce fichier pour configurer le serveur DHCP. Assurez vous de l'avoir bien compris avant de passer à la suite.

Correction

On exécute :

```
$ wget http://www.lipn.univ-paris13.fr/~evangelista/cours/M1101/TP4/dhcpd.conf
$ mv dhcpd.conf /etc
```

Q 1.3 Adapter le fichier `/etc/dhcpd.conf` en changeant les adresses MAC des machines dns et util par celles de leurs interfaces eth1. Lancer ensuite le service dhcpd.

Correction

On récupère l'adresse MAC de eth1 avec la commande `ifconfig eth1`. Il suffit ensuite de modifier les deux lignes `hardware ethernet 12:34:56:78:AB:CD;` du fichier. ◆

Q 1.4 Vérifier que le mode d'activation de eth1 sur dns et util est bien DHCP et modifier les fichiers de configuration si ce n'est pas le cas. Activer ensuite ces interfaces puis tester qu'elles sont correctement configurées. Les interfaces eth1 de dns et util doivent en particulier avoir les adresses IP fixes indiquées dans le fichier `/etc/dhcpd.conf` de dhcp.

Correction

On doit avoir la ligne `BOOTPROTO=dhcp` dans les fichiers `/etc/sysconfig/network-scripts/ifcfg-eth1` des deux machines, ce qui est normalement le cas. Après l'activation des interfaces, les machines devraient normalement avoir les adresses IP fixes indiquées dans le fichier `dhcpd.conf` du serveur. ◆

Exercice 2 — Configuration du serveur DNS

Nous allons maintenant configurer la machine dns pour qu'elle soit le serveur DNS faisant autorité sur la zone `dut.edu`. *Cet exercice doit être fait sur la machine dns uniquement.*

Q 2.1 Avec wget, récupérer dans le terminal le fichier à l'adresse suivante :
`http://www.lipn.univ-paris13.fr/~evangelista/cours/M1101/TP4/named.service`
Déplacer ensuite ce fichier dans le répertoire `/lib/systemd/system`. Ce fichier donne des indications au système d'exploitation sur la façon de lancer et d'arrêter le service named. Nous le remplaçons car le fichier initial n'est pas adapté pour ce TP.
Pour que la modification soit prise en compte il faut lancer la commande ci-dessous :

```
$ systemctl --system daemon-reload
```

Q 2.2 Créer le fichier `/etc/named.conf` (après l'avoir supprimé si il existe déjà). Ce fichier devra uniquement définir la zone `dut.edu` décrite dans le fichier `/var/named/dut.edu`. Dans un premier temps, il ne contiendra aucune option.

Correction

Le fichier `/etc/named.conf` doit contenir les lignes suivantes :

```
zone dut.edu. {
   type master;
   file "/var/named/dut.edu";
};
```

On indique ainsi que la machine dns fait autorité sur la zone `dut.edu` et que cette zone est décrite dans le fichier `/var/named/dut.edu`. ◆

Q 2.3 Créer le fichier de zone `/var/named/dut.edu`. Ce fichier devra définir :
- L'adresse mail du responsable de la zone : `root@dut.edu`
- Le serveur de noms de la zone : la machine dns.
- Les adresses IP des trois machines du réseau.
- L'alias de la machine dns : noms.
- La description de la zone : un texte de votre choix.

Correction

Le fichier `/var/named/dut.edu` doit contenir les lignes suivantes :

```
dut.edu.    SOA     dns.dut.edu. root.dut.edu. (1 3600 3600 3600 3600)
            NS      dns.dut.edu.
            TXT     "DUT R&T de Villetaneuse"
noms        CNAME   dns
dns         A       10.0.0.1
dhcp        A       10.0.0.2
util        A       10.0.0.3
```

◆

Q 2.4 Démarrer le service named. Le fichier `/var/log/messages` contiendra, en cas d'erreur dans un des fichiers DNS, un message expliquant le problème.

Exercice 3 — Test du serveur DNS

Q 3.1 Donner et tester (sur n'importe quelle machine) les commandes host permettant de récupérer les informations suivantes :

(a) l'adresse IP de `util.dut.edu`

Correction

```
host -t A util.dut.edu 10.0.0.1
```

◆

(b) le nom dont `noms.dut.edu` est un alias

Correction

```
host -t CNAME noms.dut.edu 10.0.0.1
```

◆

(c) le serveur DNS de la zone avec son adresse IP (Deux commandes host sont nécessaires.)

Correction

`host -t NS dut.edu 10.0.0.1` qui indique le nom du serveur DNS (`dns.dut.edu`) puis
`host -t A dns.dut.edu 10.0.0.1` pour avoir son adresse IP ◆

Exercice 4 — Envoi d'informations DNS par le serveur DHCP

Dans l'exercice précédent nous devions préciser à la commande host l'adresse IP du serveur DNS que nous interrogions. Pour éviter cela, nous allons changer la configuration du serveur DHCP pour qu'il fournisse directement aux clients l'adresse IP d'un serveur DNS qu'ils interrogeront par la suite lorsqu'ils voudront effectuer une résolution de nom.

Q 4.1 Modifier le fichier de configuration du serveur DHCP pour qu'il contienne à la fois la définition du serveur DNS à donner aux clients et leur nom de domaine.

Correction

On doit rajouter les lignes suivantes dans `dhcpd.conf` :

```
option domain-name-servers 10.0.0.1;
option domain-name "dut.edu";
```

◆

Q 4.2 Désactiver eth0 sur les trois machines afin qu'il n'y ait pas de conflit entre les informations données par le serveur DHCP de l'IUT et celles données par notre serveur DHCP.

Q 4.3 Désactiver puis réactiver eth1 sur dns et util puis vérifier que le fichier `/etc/resolv.conf` de dns et util contient bien les informations attendues. L'interface eth1 de dhcp étant configurée statiquement, il faut remplir manuellement ce fichier pour qu'il ait le même contenu que sur dns et util.

Correction

Le fichier `/etc/resolv.conf` de dns et util devrait contenir les lignes suivantes :

```
nameserver 10.0.0.1
search dut.edu
```

◆

Q 4.4 Retenter les tests de l'exercice 3 mais sans donner à la commande host de serveur DNS.

Q 4.5 La modification du fichier `/etc/resolv.conf` implique que dns est maintenant le serveur DNS utilisé par tous les programmes (ping, firefox, ...) pour effectuer une résolution. Vérifier que la commande `ping dns` sur la machine util fonctionne. Pourquoi n'a-t-on plus besoin de préciser le nom complet de la machine (`dns.dut.edu`) ?

Correction

La ligne search du fichier `/etc/resolv.conf` indique que tous les noms relatifs (non terminés par un '.') sont complétés par `dut.edu` lorsque le serveur DNS 10.0.0.1 est interrogé. Demander l'adresse IP de `dns` est donc équivalent à demander celle de `dns.dut.edu`. ◆

Exercice 5 — Interactions entre le serveur DHCP et le serveur DNS

Actuellement, les adresses IP des machines des utilisateurs comme util sont définies statiquement dans le fichier de configuration du serveur DHCP et dans le fichier de la zone `dut.edu`. Pour simplifier le travail de l'administrateur, on souhaite maintenant que ces adresses ne soient plus fixes mais attribuées dynamiquement par le serveur DHCP et que la machine dns soit toujours au courant de ces adresses. La difficulté est que, ces adresses étant inconnues a priori, il est impossible de les écrire dans le fichier de zone. Il est donc nécessaire de configurer le serveur DHCP pour qu'à chaque fois qu'il attribue une adresse IP à un utilisateur il en informe la machine dns pour que celle-ci mette à jour sa base de données d'adresses IP. Il est aussi nécessaire de configurer la machine dns pour qu'elle accepte ces requêtes.

Q 5.1 Sur la machine dns :
- Modifier le fichier `/etc/named.conf` pour que dns accepte les requêtes de mise à jour sur la zone `dut.edu` en provenance de la machine dhcp.
- Attribuer le répertoire `/var/named` à l'utilisateur named. Ceci est nécessaire car à la réception d'une requête de mise à jour le serveur DNS enregistre cette requête dans un fichier placé dans le répertoire `/var/named`. Il faut donc que l'utilisateur named (l'utilisateur système qui lance le service DNS) ait les droits d'écriture sur ce répertoire.

Quelle(s) modification(s) doit-on apporter au fichier de la zone `dut.edu` ?

Correction

À l'intérieur de la déclaration de la zone `dut.edu` de dns il faut rajouter la ligne suivante pour indiquer que dns accepte les requêtes de mise à jour sur cette zone venant de 10.0.0.2 (la machine dhcp) :

```
allow-update { 10.0.0.2; };
```

Pour attribuer le répertoire `/var/named` à l'utilisateur bind :

```
$ chown -R named /var/named
```

Dans le fichier `/var/named/dut.edu` il faut supprimer la ligne qui définit l'adresse IP de util, cette adresse étant maintenant attribuée par le serveur DHCP. ◆

Q 5.2 Sur la machine dhcp : modifier le fichier `/etc/dhcpd.conf` de telle sorte que :
- util n'ait plus une adresse IP fixe.
- Les mises à jour DNS soient activées pour la zone `dut.edu`.

Correction

Pour que util n'ait plus d'adresse IP fixe il suffit de supprimer la déclaration **fixed-address** dans le bloc **host** qui déclare la machine util et de rajouter la définition suivante dans ce bloc afin d'indiquer le nom que le serveur DHCP demandera d'ajouter au serveur DNS :

```
ddns-hostname "util";
```

Ensuite, pour activer les envois de requêtes de mise à jour pour la zone `dut.edu` on rajoute :

```
ddns-updates on;
ddns-update-style interim;
zone dut.edu. {
   primary 10.0.0.1;  # adresse IP de dns
}
```

◆

Q 5.3 Sur la machine util : désactiver puis réactiver eth1.

Q 5.4 Sur la machine dns : retrouver dans le fichier de journal `/var/log/messages` les messages affichés par le service named à l'activation de l'interface eth1 de util (le service named étant très bavard, on pourra directement rechercher le mot updating dans le fichier).

Correction

En exécutant la commande `grep updating /var/log/messages` sur dns on devrait normalement voir le message ci-dessous :

```
updating zone 'dut.edu/IN': adding an RR at 'util.dut.edu' A
```

Cela signifie que quand le serveur DHCP a accordé un bail avec une adresse IP à util il a demandé au serveur DNS d'ajouter un enregistrement de type A pour la machine `util.dut.edu`. (RR signifie resource record, ou enregistrement de ressource) ◆

Q 5.5 Sur la machine dhcp ou dns : vérifier la mise à jour à l'aide de la commande host ou avec un ping vers la machine util.

Exercice 6 — Attribution de noms aux machines

Actuellement une machine qui obtient un bail auprès du serveur DHCP ne connaît pas son propre nom. Nous allons maintenant changer la configuration du serveur DHCP pour qu'il donne cette information aux clients.

Q 6.1 Modifier les déclarations des hôtes dns et util dans le fichier `/etc/dhcpd.conf` afin que le serveur DHCP leur donne leurs noms.

Correction

La déclaration **host** des deux machine doit contenir l'option **host-name**. Par exemple, pour dns :

```
option host-name "dns";
```

Q 6.2 Modifier le fichier de configuration de eth1 de dns et util pour qu'il contienne le paramètre **NEEDHOSTNAME** fixé à yes. Sans cela dns et util refuseront le nom donné par le serveur DHCP.

Q 6.3 Désactiver puis réactiver les interfaces eth1 de dns et util. (À l'activation, le message d'erreur affiché au sujet de avahi n'a pas d'importance).

Q 6.4 Tester sur dns et util que :

— La commande hostname affiche bien le nom attendu.

— En ouvrant un nouveau terminal, le prompt (le message affiché en début de ligne) contient bien ce nom.

Correction

À la fin du TP les fichiers de configuration devraient avoir les contenus ci-dessous.

Pour le fichier `/etc/named.conf` sur la machine dns :

```
zone dut.edu. {
   type master;
   file "/var/named/dut.edu";
   allow-update { 10.0.0.2; };
};
```

Pour le fichier `/var/named/dut.edu` sur la machine dns :

```
dut.edu.    SOA     dns.dut.edu. root.dut.edu. (1 3600 3600 3600 3600)
            NS      dns.dut.edu.
            MX      10 dhcp.dut.edu.
            TXT     "DUT R&T de Villetaneuse"
mail        CNAME   dhcp
dns         A       10.0.0.1
dhcp        A       10.0.0.2
```

Pour le fichier `/var/named/dhcpd.conf` sur la machine dhcp :

```
default-lease-time 36000;

option domain-name-servers 10.0.0.1;
option domain-name "dut.edu";

ddns-updates on;
ddns-update-style interim;

zone dut.edu. {
   primary 10.0.0.1;
}

host dns {
   hardware ethernet 12:34:56:78:AB:CD;
   fixed-address 10.0.0.1;
   option host-name "dns";
}
host util {
   hardware ethernet 12:34:56:78:AB:CD;
   ddns-hostname "util";
   option host-name "util";
}

subnet 10.0.0.0 netmask 255.0.0.0 {
   range 10.0.0.100 10.0.0.200;
}
```

Fichiers de configuration du serveur DNS

Le fichier `/etc/named.conf` La forme typique de ce fichier est la suivante :

```
options {
  <options-globales-valables-pour-toutes-les-zones>
};
zone <nom-absolu-de-la-zone1> {
  <options-et-infos-sur-la-zone1>
};
zone <nom-absolu-de-la-zone2> {
  <options-et-infos-sur-la-zone2>
};
...
```

Chaque nom de zone est un nom DNS complet terminé par un point (p.ex., `dut.edu.`).

À l'intérieur d'une zone on doit avoir la ligne suivante qui définit le rôle du serveur pour la zone :

```
type <type-du-serveur-pour-la-zone>;
```

Il existe de nombreux types mais dans ce TP nous n'utiliserons que le type **master** qui signifie que le serveur fait autorité sur la zone. La zone doit alors contenir la déclaration du fichier de zone qui contient les informations sur celle-ci :

```
file "<chemin-absolu-du-fichier-de-zone>";
```

L'option **allow-update** permet de préciser qu'un serveur DNS accepte des requêtes de mise à jour (voir la Q 5.1). Sa syntaxe est la suivante :

```
allow-update { <adresses-de-machines> };
```

Les adresses indiquées sont les adresses IP des machines qui sont autorisées à envoyer des requêtes de mise à jour vers le serveur. Chaque adresse doit être suivie d'un point-virgule.

Les fichiers de zone Un fichier de zone contient une liste d'*enregistrements* de la forme suivante :

```
<nom>   <type>   <valeur>
```

Le nom identifie une machine ou un domaine. Il peut être absolu (terminé par un point), relatif (par rapport au nom de la zone) ou vide (auquel cas il est automatiquement remplacé par le dernier nom utilisé). Plusieurs enregistrements peuvent être associés à un nom. On les différencie par un type. Les types d'enregistrements que nous utiliserons dans ce TP sont :

— SOA (Start Of Authority) : Cet enregistrement doit être le premier du fichier si le serveur fait autorité sur la zone. Il donne des informations générales sur la zone (voir l'exemple plus bas).
— A (Address) : Définition d'une adresse IP.
— TXT : Description d'une machine ou un domaine.
— CNAME (Canonical NAME) : Définition d'un alias.
— NS (Name Server) : Définition d'un serveur DNS.

Voici un exemple de fichier de zone :

```
1 domaine.fr.     SOA     dns.domaine.fr. admin.domaine.fr. (1 3600 3600 3600 3600)
2                 NS      dns
3                 TXT     "Mon domaine"
4 dns             A       1.2.3.1
5 www             A       1.2.3.2
6 mail            A       1.2.3.3
7 messagerie      CNAME   mail
```

— Ligne 1 : Définition de la zone `domaine.fr.` sur laquelle le serveur fait autorité. L'enregistrement SOA définit le nom du serveur DNS faisant autorité sur la zone (`dns.domaine.fr.`), l'adresse mail de l'administrateur réseau de la zone (`admin.domaine.fr`, avec un '.' à la place du '@' car le '@' a une autre signification dans ce fichier). Les nombres entre parenthèses correspondent au numéro de version du fichier (1) et à différents délais en secondes (fixés à 1 heure ici) mais qui ne nous intéressent pas dans ce TP. On laissera donc ces délais à 3600.
— Ligne 2 : Le serveur DNS faisant autorité sur la zone doit être défini avec un enregistrement NS.
— Ligne 3 : Description du domaine `domaine.fr`.
— Ligne 4 à 6 : Définition d'adresses IP.
— Ligne 7 : Définition de `messagerie.domaine.fr` qui est un alias de `mail.domaine.fr`.

Options DHCP liées à DNS

Envoi d'informations DNS aux clients (Q 4.1) On peut préciser l'adresse IP du serveur DNS qui sera indiqué aux clients en utilisant l'option **domain-name-servers**.

```
option domain-name-servers <adresse-ip-du-serveur-dns>;
```

Il faut utiliser l'option **domain-name** pour préciser le nom du domaine sur lequel se trouvent les machines.

```
option domain-name "<nom-du-domaine>";
```

Envoi de requêtes de mise à jour (Q 5.2) La déclaration suivante indique que l'envoi de requêtes de mise à jour vers les serveurs DNS est activée (ddns = dynamic dns) :

```
ddns-updates on;
ddns-update-style interim;
```

À l'intérieur de la déclaration d'un hôte (un bloc **host**) il faut ensuite placer la déclaration suivante qui indique le nom que le serveur DHCP demandera au serveur DNS d'insérer dans sa base de données d'adresses IP pour cette machine.

```
ddns-hostname "<nom-de-la-machine>";
```

Enfin, il faut déclarer chaque zone pour laquelle on souhaite envoyer des requêtes de mise à jour avec l'adresse IP du serveur DNS vers lequel les requêtes seront envoyées :

```
zone <nom-absolu-de-la-zone> {
  primary <adresse-ip-du-serveur-DNS-destinataire>;
}
```

Envoi de noms aux clients (Q 6.1) Le serveur DHCP peut donner son nom à un client. À l'intérieur de la déclaration de l'hôte correspondant (un bloc **host**) il faut placer la déclaration suivante :

```
option host-name "<nom-de-la-machine>";
```

TP 5 — Routage statique avec Marionnet

Ce TP se fera sous Marionnet qui est un simulateur de réseaux fonctionnant sous Linux. C'est un logiciel libre développé par Jean-Vincent Loddo, enseignant à l'IUT de Villetaneuse. Il peut être téléchargé gratuitement à l'adresse `http://www.marionnet.org`.

L'objectif est de créer le réseau de la Figure 3 et de configurer les interfaces et tables de routage des machines pour qu'elles puissent communiquer entre elles et avec le monde réel via la passerelle G. La passerelle doit être vue comme un routeur qui connecte notre réseau virtuel au reste du réseau Internet.

Trois réseaux sont interconnectés :
— le réseau 10.0.1.0/24 composé des machines R1(eth0) et M1 ;
— le réseau 10.0.2.0/24 composé des machines R2(eth0) et M2 ;
— le réseau 10.0.3.0/24 composé des machines R1(eth1), R2(eth1) et de la passerelle G reliés par le switch S.
R1 et R2 joueront le rôle de routeurs entre ces trois réseaux.

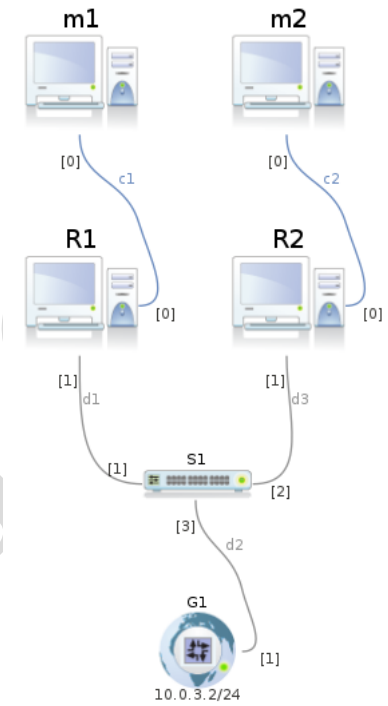

FIGURE 3 – Le réseau à réaliser

Exercice 1 — Création du réseau

Créer un nouveau projet et ajouter tous les équipements présents sur la Figure 3. À l'ajout des machines R1 et R2, il faut bien préciser qu'elles ont deux cartes Ethernet. Il faut aussi faire attention aux interfaces connectées lors de l'ajout de câbles. Par exemple, eth0 de R1 est connectée à M1 et eth1 est connectée au switch S, comme indiqué sur la figure.

Une fois tous les équipements ajoutés, cliquer sur *Tout démarrer* puis ouvrir une session root sur tous les terminaux. Le mot de passe root est root.

Exercice 2 — Configuration des interfaces

Q 2.1 Choisir des adresses IP pour les six interfaces réseau utilisées.

Correction

On choisit les adresses suivantes :

Machine	Interface	Adresse IP
M1	eth0	10.0.1.1
M2	eth0	10.0.2.2
R1	eth0	10.0.1.254
	eth1	10.0.3.10
R2	eth0	10.0.2.254
	eth1	10.0.3.20

L'adresse IP de G a été fixée à la création. Elle doit normalement être 10.0.3.2.

Q 2.2 Attribuer son adresse IP à chaque interface, sans modifier les fichiers de configuration mais directement avec la commande ifconfig.

Correction

Sur R1 :

```
$ ifconfig eth0 10.0.1.254 netmask 255.255.255.0
$ ifconfig eth1 10.0.3.10  netmask 255.255.255.0
```

Q 2.3 Tester que deux machines sur le même réseau peuvent s'envoyer des messages ping.

Exercice 3 — Configuration des tables de routage

Q 3.1 Donner la table de routage de M1.

Correction

Destination	Masque	Routeur	Interface
10.0.1.0	255.255.255.0	-	eth0
0.0.0.0	0.0.0.0	10.0.1.254	eth0

Q 3.2 Donner la table de routage de R1.

Correction

Destination	Masque	Routeur	Interface
10.0.1.0	255.255.255.0	-	eth0
10.0.3.0	255.255.255.0	-	eth1
10.0.2.0	255.255.255.0	10.0.3.20	eth1
0.0.0.0	0.0.0.0	10.0.3.2	eth0

Q 3.3 Modifier la valeur du paramètre système `net.ipv4.ip_forward` sur R1 et R2 pour qu'elles acceptent de router les paquets.

Correction

```
$ sysctl net.ipv4.ip_forward=1
```

Q 3.4 Compléter les tables de routage de M1, M2, R1 et R2.

Correction

Toutes les lignes en remise directe ont déjà été ajoutées automatiquement par ifconfig lors de l'activation des interfaces. Il reste donc à ajouter les lignes en remise indirecte.
Pour M1 :

```
$ route add default gw 10.0.1.254 dev eth0
```

Pour R1 :

```
$ route add -net 10.0.2.0 netmask 255.255.255.0 gw 10.0.3.20 dev eth1
$ route add default gw 10.0.3.2 dev eth1
```

Q 3.5 Tester que deux machines sur des réseaux différents peuvent s'envoyer des messages ping.

Q 3.6 Sur M1 : utiliser la commande traceroute pour déterminer le chemin vers M2.

Correction

```
$ traceroute 10.0.2.2
```

◆

Exercice 4 — Test de l'accès au monde extérieur

La passerelle G bloque les messages ping : une machine de notre réseau virtuel ne peut pas envoyer de ping à une machine du monde réel. Par contre toute les autres messages sont relayés par la passerelle. Nous allons faire deux tests d'accès à l'extérieur.

Q 4.1 *Accès à la machine physique*

Q 4.1.1 Sur votre machine physique : récupérer l'adresse IP.

Q 4.1.2 Sur M1 : ouvrir firefox puis rentrer l'adresse IP de la machine physique dans la barre d'adresse. Un message indiquant que le serveur http est bien démarré sur la machine physique devrait s'afficher dans le navigateur.

Q 4.2 *Téléchargement d'un fichier*

Q 4.2.1 Sur M1 : essayer de récupérer le fichier `www.iutv.univ-paris13.fr/index.php` avec la commande wget. Quel est le problème ?

Correction

```
$ wget www.iutv.univ-paris13.fr/index.php
```

On constate un problème de résolution de nom. Avant d'envoyer la requête pour demander le fichier `index.php`, la commande wget va interroger un serveur DNS pour obtenir l'adresse IP correspondant à `www.iutv.univ-paris13.fr`. Or le fichier `/etc/resolv.conf` contenant le nom du serveur DNS interrogé par wget est vide. La résolution est donc impossible. ◆

Q 4.2.2 Proposer une solution puis retester.

Correction

Il faut connaître l'adresse IP d'un serveur DNS. On peut par exemple utiliser 193.54.26.1 qui est l'adresse IP du serveur de l'université. On insère ensuite la ligne suivante dans le fichier `/etc/resolv.conf` :

```
nameserver 193.54.26.1
```

La commande wget devrait alors fonctionner. wget interroge d'abord 193.54.26.1 pour connaître l'adresse IP de `www.iutv.univ-paris13.fr` et envoie ensuite sa requête de demande de fichier à cette adresse. ◆

TP 6 — Filtrage

L'objectif est de créer avec marionnet le réseau de la Figure 4 et de réaliser les opérations suivantes :
— configuration des interfaces et tables de routage des machines (exercice 2)
— mise en place de règles de filtrage pour protéger le routeur R (exercice 3)
— mise en place de règles de filtrage pour protéger le réseau composé des machines nfs et anna de l'extérieur (exercice 4 et 5)

Vous trouverez en page 32 une description du logiciel iptables utilisé dans les exercices 3, 4 et 5.

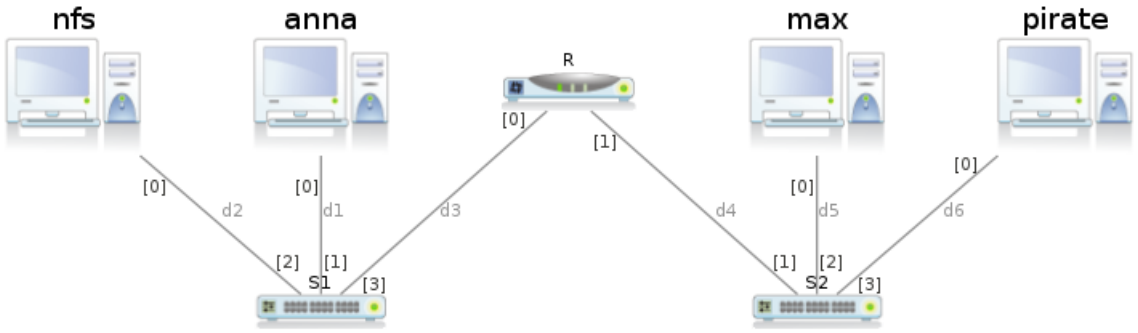

FIGURE 4 – Le réseau à réaliser

Le réseau composé des machines nfs, anna et R(eth0) aura l'adresse 10.1.0.0/24. Celui composé des machines R(eth1), pirate et max aura l'adresse 10.2.0.0/24.

Exercice 1 — Création du réseau

Créer un nouveau projet et ajouter tous les équipements présents sur la Figure 4. À l'exception du routeur, tous les équipements pourront être ajoutés avec les valeurs proposées par le logiciel. Pour R il est important de cocher l'option *Show unix terminal* afin de pouvoir le configurer par la suite et de changer l'adresse IP proposée pour eth0 en choisissant une adresse cohérente avec les adresses de réseau données en introduction. À l'ajout des câbles reliant R aux switchs il faut aussi faire attention aux interfaces utilisées : eth0 de R est connectée au switch S1 et eth1 est connectée au switch S2.

Une fois tous les équipements ajoutés, cliquer sur *Tout démarrer* puis ouvrir une session root sur tous les terminaux. Le mot de passe root est root.

Exercice 2 — Configuration des interfaces et tables de routage

Q 2.1 Choisir une adresse IP pour chacune des 6 interfaces ethernet utilisées.

Correction

On choisit les adressees IP ci-dessous :

Machine	Interface	Adresse IP
nfs	eth0	10.1.0.1
anna	eth0	10.1.0.2
R	eth0	10.1.0.254
	eth1	10.2.0.254
max	eth0	10.2.0.1
pirate	eth0	10.2.0.2

◆

Q 2.2 Attribuer son adresse IP à chaque interface, sans modifier les fichiers de configuration mais directement avec la commande ifconfig.

Q 2.3 Sur nfs, anna, max et pirate, ajouter une route leur permettant d'envoyer des paquets vers le réseau voisin.

Correction

Sur ces quatre machines il faut rajouter une route par défaut passant par l'adresse IP de l'interface de R étant sur le même réseau que cette machine.
Par exemple, sur anna :

```
$ route add default gw 10.1.0.254 dev eth0
```

et sur max :

```
$ route add default gw 10.2.0.254 dev eth0
```

Q 2.4 Tester que toutes les machines peuvent se pinguer, en particulier les machines de réseaux différents.

Q 2.5 Tester que R peut ouvrir une connexion SSH sur nfs. Fermer ensuite la connexion SSH. Pour démarrer le service SSH sur nfs il faut utiliser la commande ci-dessous (la commande systemctl vue dans les TP précédents ne marche pas sur les distributions de Linux utilisées par Marionnet) :

```
$ /etc/init.d/ssh start
```

Exercice 3 — Sécurisation du routeur R

Le routeur R ne doit pas pouvoir émettre ou recevoir de messages (autres que ceux qu'il route). L'unique exception à cette règle est pour les messages du protocole ICMP. ICMP (*Internet Control Message Protocol*) est le protocole utilisé par la commande ping. Il est aussi utilisé par les routeurs pour signaler des problèmes de réseau comme, par exemple, l'impossibilité de router un paquet. On veut donc que R puisse émettre ou recevoir des messages ICMP.

Q 3.1 D'après l'énoncé ci-dessus quelles sont les chaînes que l'on doit modifier sur R pour mettre en œuvre ces contraintes ? Quelle politique par défaut doit on choisir pour ces chaînes et quelles règles doit-on y ajouter ? Modifier ces chaînes en conséquence.

Correction

Ce sont les chaînes INPUT et OUTPUT qui permettent de filtrer les messages envoyés ou reçus par le routeur. Dans notre cas, on souhaite interdire toute émission ou réception sauf pour les paquets ICMP qui seront acceptés. On fixe donc la politique par défaut de ces deux chaînes à DROP et on rajoute deux règles avec l'action ACCEPT afin d'accepter l'émission ou la réception de paquets ICMP.

```
$ iptables -P INPUT  DROP
$ iptables -P OUTPUT DROP
$ iptables -A INPUT  -p icmp -j ACCEPT
$ iptables -A OUTPUT -p icmp -j ACCEPT
```

Q 3.2 Vérifier que R répond toujours aux pings et peut envoyer des pings aux autres machines.

Q 3.3 Vérifier que la connexion SSH de la question Q 2.5 n'est plus possible.

Exercice 4 — Sécurisation du réseau 10.1.0.0/24

On veut maintenant que les contraintes ci-dessous soient vérifiées :

— nfs est un serveur NFS qui doit être accessible uniquement depuis le réseau 10.1.0.0/24. nfs de doit pas pouvoir envoyer de paquets vers l'extérieur ni en recevoir depuis l'extérieur.

— La machine pirate a tenté à plusieurs reprises d'accéder au serveur nfs. On veut donc lui bloquer complétement l'accès au réseau 10.1.0.0/24 en lui interdisant d'envoyer des paquets sur ce réseau. De plus, on veut enregistrer dans le fichier de journal de R les tentatives d'envoi de paquets de pirate sur ce réseau.

Q 4.1 Quelle est maintenant la chaîne à modifier sur R pour mettre en œuvre ces contraintes ? Quelle politique par défaut doit on choisir pour cette chaîne et quelles règles dont-on y ajouter ? Modifier cette chaîne en conséquence.

Correction

Les contraintes énoncées sont des contraintes de routage et c'est la chaîne FORWARD qui indique ce que le routeur est autorisé à router. Ici, on est dans un cas où R laisse passer (route) tous les paquets sauf dans les cas particuliers énoncés. On fixe donc la politique par défaut de la chaîne FORWARD à ACCEPT et on rajoute des règles pour indiquer les paquets que R refuse de router :

— Lignes 2 et 3 : pour la première contrainte.

— Lignes 4 et 5 : pour la deuxième contrainte.

```
1 $ iptables -P FORWARD ACCEPT
2 $ iptables -A FORWARD -i eth0 -o eth1 -s 10.1.0.1 -j DROP
3 $ iptables -A FORWARD -i eth1 -o eth0 -d 10.1.0.1 -j DROP
4 $ iptables -A FORWARD -i eth1 -o eth0 -s 10.2.0.2 -j LOG
```

```
5 $ iptables -A FORWARD -i eth1 -o eth0 -s 10.2.0.2 -j DROP
```

La règle LOG à la ligne 4 doit absolument être ajoutée avant la règle DROP de la ligne 5. Dans le cas contraire, la règle LOG serait inutile car si les conditions de cette règle étaient remplies pour un paquet alors R appliquerait d'abord la règle DROP (car les conditions sur les paquets sont les mêmes pour les deux règles). Le paquet serait rejeté et la règle LOG ne serait jamais appliquée. ◆

Q 4.2 Tester en vérifiant que :
— pirate ne peut pinguer ni nfs ni anna.
— nfs ne peut pas pinguer max et inversement, max ne peut pas pinguer nfs.
— anna peut pinguer max et nfs.
Vérifier également que les tentatives de la machine pirate de pinguer nfs et anna ont été enregistrées dans le fichier de journal (`/var/log/messages`).

Exercice 5 — Interdiction des paquets SSH

On veut enfin empêcher les utilisateurs hors du réseau 10.1.0.0/24 de pouvoir se connecter par SSH sur les machines de ce réseau. Rechercher sur Internet comment mettre en place sur R les règles de filtrage appropriées. Mettre en place ces règles puis tester.

Correction

(Cette question n'est pas au programme de l'examen. Elle fait appel à des notions abordées dans des modules futurs.)

Les protocoles comme SSH utilisent un code pour communiquer appelé *numéro de port* et fonctionnent avec un *protocole de transport* qui est soit TCP soit UDP. Dans le cas de SSH on utilise TCP comme protocole de transport et le numéro de port utilisé par un serveur SSH est 22. On peut donc rajouter sur R la règle suivante qui lui permet de refuser de laisser entrer sur le réseau 10.1.0.0/24 un paquet destiné à un serveur SSH :

```
$ iptables -A FORWARD -p tcp --dport 22 -i eth1 -o eth0 -j DROP
```

L'option dport signifie *destination port*. ◆

Le pare-feu iptables Le pare-feu *iptables* est présent dans tous les systèmes Linux. Il permet d'établir des règles de filtrage sur des ordinateurs personnels afin, par exemple, d'empêcher l'utilisation de certains programmes ; ou sur des routeurs afin de sécuriser des réseaux.

Les règles de filtrage sont réparties par iptables dans trois *chaînes* :
— INPUT ⇔ règles de filtrage pour les paquets entrants (ceux destinés à la machine)
— OUTPUT ⇔ règles de filtrage pour les paquets sortants (ceux émis par la machine)
— FORWARD ⇔ règles de filtrage pour les paquets routés (ceux reçus par la machine mais qui ne lui sont pas destinés et doivent donc être routés)

Une règle de filtrage consiste en une action à appliquer sur un paquet dans des conditions particulières (p.ex., le paquet est destiné à la machine 10.1.2.3). Les deux actions que nous utiliserons dans ce TP sont :
— DROP ⇔ le paquet est rejeté
— ACCEPT ⇔ le paquet est accepté

Ainsi, pour un paquet entrant, sortant ou routé, iptables essaiera de lui appliquer une par une les règles se trouvant dans la chaîne correspondante. Si une des règles s'applique (c'est-à-dire que les conditions de la règle sont remplies pour le paquet) iptables lui appliquera l'action (DROP ou ACCEPT) et sortira de la chaîne (ignorera les règles suivantes de la chaîne). Si aucune des règles composant la chaîne ne s'applique alors iptables appliquera une *politique par défaut* qui sera DROP ou ACCEPT.

Il existe aussi une action particulière appelée LOG qui n'indique pas ce qu'iptables doit faire du paquet mais est utilisée dans une chaîne pour indiquer qu'on souhaite enregistrer dans le fichier de journal (`/var/log/messages`) le traitement d'un paquet répondant à des conditions particulières. Si une règle dont l'action est LOG s'applique, alors iptables ne sortira pas de la chaîne correspondante : il continuera à chercher une règle permettant de décider si le paquet doit être accepté ou refusé. Il faut donc faire attention à la position des règles LOG dans une chaîne : si une règle DROP portant sur les mêmes conditions se trouve avant la règle LOG dans la chaîne, alors cette dernière ne sera jamais exécutée (car la règle DROP provoque la sortie de la chaîne).

Voici les différentes façons d'utiliser iptables dont nous aurons besoin dans ce TP :
— Pour modifier la politique par défaut d'une chaîne :
```
iptables -P <chaîne> <action>
```
— Pour afficher les règles se trouvant dans une chaîne :
```
iptables -L <chaîne>
```
— Pour supprimer toutes les règles d'une chaîne :
```
iptables -F <chaîne>
```
— Pour ajouter une règle de filtrage à une chaîne :
```
iptables -A <chaîne> <conditions> -j <action>
```

Voici les conditions dont nous aurons besoin pour exprimer les règles de routage :
`-p icmp` ⇔ Le paquet contient un message ICMP.
`-i <int>` ⇔ L'interface d'entrée du paquet est int (condition incorrecte pour la chaîne output).
`-o <int>` ⇔ L'interface de sortie du paquet est int (condition incorrecte pour la chaîne input).
`-s <ip>` ⇔ L'adresse IP de la source du paquet est ip.
`-d <ip>` ⇔ L'adresse IP du destinataire du paquet est ip.

TP 7 — Administration des systèmes

Le TP doit être fait sur l'image mageia4 qui doit être restaurée en début de séance.

Exercice 1 — Tâches à exécution différée

La commande `at` permet de programmer des tâches à exécuter une seule fois à un instant donné.

Q 1.1 Les tâches sont lancées par le service `atd`. Démarrer ce service si nécessaire.

Q 1.2 Créer un fichier `/tmp/fic` vide. Programmer la suppression du contenu du répertoire `/tmp` dans 1 minute. Vérifier après 1 minute que `/tmp/fic` a bien été supprimé.

Correction

```
$ touch /tmp/fic
$ at now + 2 minutes
at> rm -rf /tmp/*
at> <EOT>
job 1 at Fri Oct 4 16:17:00 2013
```

◆

Q 1.3 Programmer le redémarrage de l'ordinateur (commande `reboot`) à 1 heure du matin.

Correction

```
$ at 01:00
at> reboot
at> <EOT>
job 2 at Sat Oct 5 01:00:00 2013
```

◆

Q 1.4 Afficher la liste des tâches programmées (commande `atq`) et annuler le redémarrage.

Correction

```
$ atq
2       Sat Oct 12 01:00:00 2013 a evangelista
$ at -d 2
```

Le premier numéro affiché sur chaque ligne par atq est un numéro de tâche (ici 2). C'est ce numéro qu'on doit donner à at pour annuler une tâche. ◆

Exercice 2 — Tâches périodiques

La commande `crontab` permet de créer des tâches qui seront exécutées périodiquement par le service `crond`. Les possibilités d'utilisation les plus courantes sont :

— `crontab -e` ⇒ pour éditer les tâches
— `crontab -l` ⇒ pour voir la liste des tâches
— `crontab -r` ⇒ pour supprimer toutes les tâches

L'éditeur par défaut lancé par `cron` pour saisir des règles est `vi`. On peut le modifier via la variable d'environnement `EDITOR` qui doit contenir le chemin complet de l'éditeur à utiliser. Par exemple :

```
$ export EDITOR=/usr/bin/emacs
$ crontab -e  #  on ouvre emacs plutôt que vi
```

Q 2.1 Il faut que les services `crond` et `postfix` soient démarrés pour pouvoir exécuter les tâches cron et envoyer les mails écrits par crond. Démarrer ces services si nécessaire.

Q 2.2 Modifier l'éditeur pour un éditeur de votre choix (emacs, gedit, ...) puis éditer les règles cron et créer une règle ajoutant la date courante à la fin du fichier `/tmp/la_date` toutes les minutes. Utiliser pour cela la commande `date` suivie d'une redirection.

Correction

```
$ export EDITOR=/usr/bin/emacs
$ crontab -e
```

Puis dans l'éditeur emacs on écrit la règle suivante :

```
* * * * * date >> /tmp/la_date
```

On sauvegarde et on ferme emacs puis le message suivant s'affiche dans le terminal pour indiquer que la règle est sauvegardée :

```
crontab: installing new crontab
```

◆

Q 2.3 Éditer les règles cron et affecter à la variable `MAILTO` votre adresse email pour recevoir à cette adresse le résultat des tâches exécutées par crond. Toujours en utilisant la commande `date`, créer une règle affichant "`salut, la date est <date>.`" toutes les 5 minutes pendant la séance de TP (et pendant la séance uniquement).

Correction

On lance crontab en mode édition et on saisit dans emacs :

```
MAILTO=monemail@chezmoi.fr
*/5 9-12 9 10 3 echo "salut, la date est `date`"
```

La règle marche pour un TP de 9h à midi le mercredi 9 octobre. ◆

Q 2.4 Supprimer le contenu de la table cron après avoir vérifié que les tâches programmées sont bien exécutées.

Correction

```
$ crontab -r
```

◆

Exercice 3 — Analyse des fichiers des utilisateurs et groupes

Les descriptions des fichiers et des commandes relatives aux groupes et utilisateurs sont données dans le cours 4 et dans le memo des commandes.

Q 3.1 Consulter le fichier `/etc/passwd` pour trouver les informations ci-dessous :
- UID de l'utilisateur root
- GID du groupe primaire de l'utilisateur root
- shell de l'utilisateur root
- répertoire personnel de l'utilisateur root

Correction

La ligne concernant l'utilisateur root dans `/etc/passwd` est la suivante :

```
root:x:0:0:root:/root:/bin/bash
```

L'UID est le 3ème champ : 0. Sous Linux, l'utilisateur root a toujours l'identifiant 0.
Le GID est le 4ème champ : root a aussi le GID 0.
Le répertoire personnel est le 6ème champ : `/root`.
Le shell est le 7ème champ : `/bin/bash`. ◆

Q 3.2 Quel est le nom du groupe primaire de l'utilisateur root ?

Correction

L'information est dans le fichier `/etc/group`. Il faut trouver la ligne dont le 3ème champ (le GID) vaut 0.

```
root:x:0:
```

Le groupe primaire de l'utilisateur root s'appelle aussi root. ◆

Q 3.3 Quels sont les utilisateurs ayant un mot de passe crypté ?

Correction

L'information est dans le fichier `/etc/shadow`. Ce sont tous les utilisateurs pour lesquels le 2ème champ commence par un $ suivi du mot de passe crypté. Il y en a deux : root et etudiant.

```
root:$2a$08$Qoyn3ovr/4ypl6l3ct41R.YYAJhXN/FQkbeqOeT6bUH5RiDehAWI.:15797:0:99999:7:::
etudiant:$2a$08$phfFA0ZxALi1DZ4Oyet4sefbPIqtTylIsgeV6cMGLUpzU/uzc7M/K:15797:0:99999:7:::
```

Les autres comptes sont soit désactivés soit en attente de saisie d'un mot de passe. ◆

Exercice 4 — Modification des groupes et utilisateurs

Q 4.1 Créer un utilisateur `bilou` d'UID 1234 ayant pour shell `/bin/bash`. Consulter le manuel pour trouver les options adéquates.

Correction

```
$ useradd -u 1234 -s /bin/bash bilou
```
◆

Q 4.2 Consulter les fichiers `/etc/passwd`, `/etc/shadow` et `/etc/group`. Que constate-t-on après la création de la question précédente ? Quel est le répertoire personnel du nouvel utilisateur ? Quels sont les droits associés à ce répertoire ?

Correction

Une ligne a été rajoutée dans `/etc/passwd` :

```
bilou:x:1234:1234::/home/bilou:/bin/bash
```

dans `/etc/shadow` :

```
bilou:!:15995:0:99999:7:::
```

et dans `/etc/group` :

```
bilou:x:1234:
```

On remarque que :
- Un groupe appelé bilou a été créé avec le GID 1234. C'est un groupe privé contenant uniquement l'utilisateur bilou.
- Le x dans le fichier `/etc/passwd` indique que le mot de passe de bilou n'apparaît pas dans `/etc/passwd` mais dans `/etc/shadow`.
- Pour l'instant le compte est bloqué (dû au ! dans le fichier `/etc/shadow`). Il faut saisir un mot de passe pour bilou pour le débloquer.

Le répertoire personnel de bilou est `/home/bilou`. Il a été créé par la commande useradd. On vérifie les droits avec :

```
$ ls -l /home
drwxr-xr-x  5 bilou    bilou    4096 oct.  17 12:14 bilou/
```

Le répertoire appartient à l'utilisateur bilou et au groupe bilou. L'utilisateur bilou a tous les droits dessus. Les autres utilisateurs ont uniquement le droit en lecture et en exécution (se déplacer dans le répertoire). ◆

Q 4.3 Affecter un mot de passe au nouvel utilisateur. Quel modification a été apportée au fichier `/etc/shadow` ?

Correction

Pour changer le mot de passe de bilou :

```
$ passwd bilou
```

Le mot de passe a été crypté et inséré dans le fichier `/etc/shadow` à la place du ! :

```
bilou:$2a$08$39SSXE1DIsRySx/gQBKgCO49cl7RF7DCn1KwmK5PNthZUS.lQBD/S:15995:0:99999:7:::
```
◆

Q 4.4 Vérifier que le compte a bien été créé avec le mot de passe associé en ouvrant un terminal puis en ouvrant dans celui-ci une session sous l'identité `bilou`. Fermer ensuite le terminal.

Correction

```
$ su - bilou
$ exit
```
◆

Q 4.5 Créer un groupe `TP-RT`.

Correction

```
$ groupadd TP-RT
```

◆

Q 4.6 Consulter le fichier `/etc/group`. Que constate-t-on ?

Correction

On a une ligne en plus :

```
TP-RT:x:1235:
```

Son GID a été calculé en ajoutant 1 au plus grand GID précédent. ◆

Q 4.7 Peut-on supprimer le groupe primaire de `bilou` avec la commande `groupdel` ? Pourquoi ?

Correction

```
$ groupdel bilou
groupdel : impossible de supprimer le groupe primaire de l'utilisateur bilou
```

Le système ne nous laisse pas supprimer un groupe si au moins un utilisateur a ce groupe en groupe primaire. Il ne faut pas qu'un utilisateur se retrouve sans groupe primaire. ◆

Q 4.8 Ajouter le groupe `TP-RT` en groupe secondaire de l'utilisateur `bilou` en éditant le fichier `/etc/group`. Vérifier avec la commande `id` que l'ajout s'est bien passé.

Correction

On rajoute bilou à la fin de la ligne du groupe TP-RT dans `/etc/group` :

```
TP-RT:x:1235:bilou
```

Le dernier champ est la liste des utilisateurs ayant ce groupe en groupe secondaire.
La commande id nous affiche bien le groupe principal de bilou et tous les groupes auxquels il appartient :

```
$ id
uid=1234(bilou) gid=1234(bilou) groupes=1234(bilou),1235(TP-RT)
```

◆

Exercice 5 — Le fichier `.bashrc`

Le fichier `.bashrc` à la racine du répertoire personnel d'un utilisateur est un script lancé à l'ouverture d'une session bash (c'est-à dire quand on ouvre un terminal). On l'utilise généralement pour modifier certaines variables d'environnement ou pour afficher des informations pour l'utilisateur.

Q 5.1 Modifier le fichier `.bashrc` de votre répertoire personnel afin que :

(a) Le message "`Bienvenue <nom-utilisateur>, tu es sur la machine <nom-machine>`" soit affiché à l'ouverture de la session. Utiliser pour cela les variables d'environnement `$USER` et `$HOSTNAME` qui contiennent le nom de l'utilisateur connecté et le nom de sa machine.

(b) La date soit affichée.

(c) L'umask soit tel qu'à la création d'un nouveau fichier (ou répertoire) :

i. aucun droit ne vous soit retiré

ii. le droit en écriture soit retiré à tous les utilisateurs du groupe

iii. tous les droits soient retirés pour les autres utilisateurs

Correction

On édite le fichier `/home/etudiant/.bashrc` avec emacs et on rajoute les trois lignes suivantes à la fin du fichier :

```
echo "Bienvenue␣$USER, tu es sur la machine $HOSTNAME"
date
umask 027
```

L'umask permet de définir les droits qui seront associés à tout nouveau fichier ou répertoire créé (mais pas à ceux déjà existant).
Avec un umask de 000 tout nouveau fichier a les droits 666 (lecture/écriture pour tout le monde). Avec un umask de 027 on ne retire aucun droit au propriétaire du fichier créé, on retire le droit en écriture aux utilisateurs du groupe principal de l'utilisateur et on retire tous les droits aux autres utilisateurs. Les droits d'un nouveau fichier seront donc 640 (ou `rw-r---`).
Avec un umask de 000 tout nouveau répertoire a les droits 777 (lecture/écriture/exécution pour tout le monde). Avec notre umask de 027 tout nouveau répertoire aura donc les droits 750 (ou `rwxr-x--`). ◆

Q 5.2 Tester en ouvrant un nouveau terminal, puis en créant un fichier et un répertoire et en vérifiant les droits qui leur sont associés.

Q 5.3 La variable PS1 contient une chaîne de caractères qui correspond au *prompt*. Le prompt est le message d'invite affiché en début de ligne par l'interpréteur de commandes. Modifier votre fichier `.bashrc` pour que ce prompt soit de la forme suivante :

```
<login> sur <machine> dans le répertoire <chemin-complet-du-repertoire-courant>
<heure> $
```

Ce qui pourra par exemple donner :

```
etudiant sur q20305 dans le répertoire ~/TP7
10:44 $
```

La variable PS1 peut contenir des méta-caractères de la forme '\X' qui seront remplacés à l'affichage du prompt par une valeur donnée (le login de l'utilisateur connecté, par exemple). Rechercher sur Internet les méta-caractères adéquats.

Correction

Il faut rajouter la ligne suivante dans le fichier `.bashrc`

```
PS1='\u sur \h dans le répertoire \W\n\A $'
```