R203
Bases des services réseaux
Travaux pratiques

Sami Evangelista

IUT de Villetaneuse

Département Réseaux et Télécommunications

2023–2024

http://www.lipn.univ-paris13.fr/~evangelista/cours/R203

Table des matières

Introductions aux travaux pratiques

Marionnet Les travaux pratiques du module R203 sont à faire avec l'outil marionnet (`https://marionnet.org`) qui est déjà installé sur les machines de l'IUT. Pour utiliser marionnet sur votre ordinateur personnel, vous pouvez télécharger à cette adresse l'image virtualbox d'un système ubuntu sur lequel l'outil est déjà installé :

`https://www.marionnet.org/site/index.php/fr/documentation/demarrage-rapide`

Compte-rendus Vous devez écrire un compte-rendu pour chaque TP. Ces compte-rendus seront notés. Les compte-rendus doivent contenir :
— des copies d'écran et explications des manipulations réalisées pour les différentes instructions du sujet (les boîtes bleues) ;
— et des réponses aux questions du sujet (les boîtes vertes), avec éventuellement des copies d'écran.
Vous enverrez vos comptes-rendus à l'issue du module.

Consignes d'envoi des compte-rendus

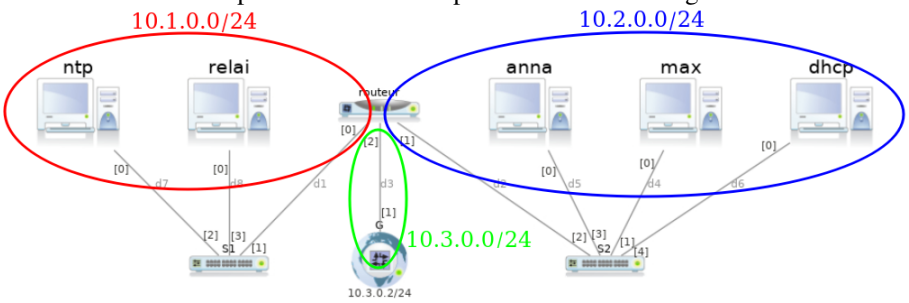
— La date limite est fixée au *Mercredi 3 avril*.
— Vous enverrez un seul fichier PDF contenant vos 4 compte-rendus. Seuls les *fichiers PDF* seront acceptés.
— Le fichier doit être envoyé à votre chargé de TP.
— La réception d'un fichier après la date limite entraînera une pénalité de 2 points par jour.

Remerciements Je remercie Camille Coti qui a écrit un TP DHCP dont le TP 1 de ce sujet reprend de nombreux éléments :

`https://www-lipn.univ-paris13.fr/~coti/cours/M2106_TP.pdf`

TP 1 — Le service DHCP

L'objectif de ce TP est de mettre en place un service dhcp sur le réseau de la figure ci-dessous :



Le réseau est scindé en trois sous-réseaux interconnectés par un routeur.

Le sous-réseau 10.1.0.0/24 est composé de serveurs. On y a placé un seul serveur (ntp, un serveur de temps) pour simplifier. Un relai dhcp y est aussi présent pour relayer les requêtes des hôtes de ce sous-réseau vers le serveur dhcp.

Le sous-réseau 10.2.0.0/24 est composé de machine d'utilisateurs (anna et max) et d'un serveur dhcp.

Enfin, le sous-réseau 10.3.0.0/24 contient la passerelle (gateway) G. Une passerelle est un équipement de marionnet qui permet de relier les hôtes de notre réseau à Internet.

Pilotage des services Les machines virtuelles marionnet utilisent un ancien système de démarrage des services qui n'inclut pas la commande `systemctl`. Pour pouvoir interagir avec un service, il faut utiliser la commande suivante :

```
$ /etc/init.d/service action
```

dans laquelle `service` doit être remplacé par le nom du service, et `action` par l'action à effectuer sur ce service (`stop`, `restart`, `status`, ...). Par exemple pour redémarrer le service `dhcpd` puis vérifier s'il est en exécution, on fera :

```
$ /etc/init.d/dhcpd restart
$ /etc/init.d/dhcpd status
```

Exercice 1 — Création du réseau

Le service de relai dhcp n'est pas présent par défaut sur les machines virtuelles marionnet. Il faut donc l'installer. Comme l'opération est un peu longue, nous allons partir du projet à l'URL suivante :

`https://www.lipn.univ-paris13.fr/~evangelista/cours/R203/dhcp.mar`

Il contient uniquement l'hôte relai sur lequel le service est déjà installé.

I 1.1 Téléchargez le projet à l'URL indiquée.

I 1.2 Lancez marionnet et ouvrez le projet téléchargé.

I 1.3 Ajoutez les équipements pour obtenir la réseau de la figure en introduction. Attention :

- À la création du routeur, cochez la case *Montrer le terminal Unix* et laissez l'IP proposée par défaut.
- À la création de la passerelle G, choisissez l'adresse de réseau 10.3.0.0/24. Marionnet attribuera l'IP 10.3.0.2 à la passerelle (voir figure).
- Enfin, respectez le câblage de la figure pour le routeur : son port 0 mène au switch des serveurs, son port 1 mène au switch des utilisateurs et son port 2 mène à la passerelle.

I 1.4 Démarrez tous les équipements et ouvrez une session root (mot de passe = root) sur chaque hôte.

Exercice 2 — Configuration statique des interfaces

Un certain nombre d'interfaces de notre réseau ne seront pas configurées automatiquement mais statiquement. Ce sont les interfaces du routeur, du relai et du serveur dhcp. En effet, pour que le service dhcp fonctionne correctement sur notre réseau, il faut que le relai et le serveur puissent communiquer, ce qui suppose qu'ils aient des IP ainsi que les routes le permettant. Nous choisirons les IP suivantes pour ces interfaces :

hôte	interface	IP
routeur	eth0	10.1.0.254
	eth1	10.2.0.254
	eth2	10.3.0.254

hôte	interface	IP
relai	eth0	10.1.0.100
dhcp	eth0	10.2.0.100

I 2.1 Attribuez au routeur, au relai et au serveur dhcp les IP indiquées dans le tableau. Utilisez les commandes suivantes en remplaçant `@ip` et `ethX` par l'IP et l'interface voulues :

```
$ ip addr add @ip/24 dev ethX
$ ip link set ethX up
```

N'oubliez pas le masque (`/24`) dans la première commande. En cas d'erreur, exécutez :

```
$ ip addr flush ethX
```

pour supprimer l'IP associée à `ethX`. Vous pouvez vérifier l'IP associée à `ethX` avec :

```
$ ip addr show ethX
```

I 2.2 Attribuez une route par défaut au routeur, au relai et au serveur dhcp. Utilisez la commande suivante :

```
$ ip route add default via @ip
```

en remplaçant `@ip` par l'IP du routeur. Vous pouvez vérifier le contenu de la table de routage avec :

```
$ ip route
```

I 2.3 Vérifiez que le relai et le serveur dhcp peuvent bien s'échanger des messages ping.

En ajoutant une route par défaut au routeur, les hôtes relai et dhcp ont maintenant accès à Internet. Cependant G bloque les messages ICMP (et donc les messages ping). Pour tester cet accès nous allons plutôt utiliser le client DNS `host`.

I 2.4 Sur la machine relai, utilisez la commande ci-dessous qui interroge le serveur DNS 193.54.26.1 (un des serveurs DNS de l'université) pour obtenir l'IP associée au serveur web de l'université.

```
$ host www.univ-paris13.fr 193.54.26.1
```

Exercice 3 — Configuration du réseau des utilisateurs

Nous allons maintenant configurer le serveur dhcp pour qu'il accorde des baux aux hôtes du réseau des utilisateurs. Même s'il n'y en a qu'une, il faut d'abord préciser sur quelle interface le serveur attendra des requêtes. Sur dhcp :

I 3.1 Éditez le fichier `/etc/default/isc-dhcp-server` avec l'éditeur nano :

```
$ nano /etc/default/isc-dhcp-server
```

I 3.2 Recherchez la ligne `INTERFACES=""` et insérez `eth0` entre les guillemets.
I 3.3 Sauvegardez le fichier (ctrl+o puis entrée) puis quittez nano (ctrl+x).

Le fichier de configuration du service dhcp est `/etc/dhcp/dhcpd.conf`. Sur dhcp :

I 3.4 Supprimez le fichier de configuration dhcp puis recréez le.
I 3.5 Avec nano, éditez le fichier pour qu'il ait le contenu suivant :

```
1 option domain-name-servers 193.54.26.1;  # serveur DNS de paris 13
2 subnet 10.2.0.0 netmask __________ {
3   range 10.2.0.1 __________;
4   option routers __________;
5   max-lease-time 3600;  # durée max. des baux = une heure
6 }
```

Complètez les blancs sachant qu'il y aura au maximum 10 hôtes connectés simultanément sur le réseau.
I 3.6 Redémarrez le service `dhcpd`.

Une fois le service dhcp démarré, anna et max pourront obtenir un bail. Cela se fait grâce à la commande `dhclient`. Cette commande est indirectement exécutée lorsque l'on active avec `ifup` une interface configurée automatiquement.

I 3.7 Sur les hôtes anna et max, demandez un bail pour l'interface eth0 :

```
$ dhclient eth0
```

Vous devriez observer le message `bound to ...` indiquant que le client a reçu une IP. Faites ensuite les tests suivants :

I 3.8 Anna peut envoyer un message ping à max et au relai.
I 3.9 Anna peut télécharger la page d'accueil de google avec `wget` (client web en mode terminal) :

```
$ wget http://www.google.fr
```

(Ceci devrait télécharger un fichier `index.html` que vous pouvez ensuite supprimer.)

Q 3.1 Sur anna, quelle commande peut-on exécuter pour vérifier que la ligne 4 du fichier de configuration du service a bien été prise en compte ? Testez la commande.
Q 3.2 Sur anna, quel fichier peut-on consulter pour vérifier que la ligne 1 du fichier de configuration du service a bien été prise en compte ? Consultez ce fichier.
Q 3.3 Expliquez pourquoi ces deux lignes (1 et 4) sont nécessaires pour que le test de l'instruction I 3.9 fonctionne.

Exercice 4 — Configuration du réseau des serveurs

Passons maintenant au réseau des serveurs. Il est d'abord nécessaire de configurer le relai pour qu'il écoute sur son interface eth0 et qu'il redirige les requêtes dhcp reçues sur cette interface vers le serveur dhcp. Sur relai :

I 4.1 Répétez les instructions I 3.1 à I 3.3 mais sur le fichier `/etc/default/isc-dhcp-relay`. Modifiez ausi la ligne `SERVERS=""` en insérant l'IP du serveur dhcp entre les guillemets.
I 4.2 Redémarrez le service `isc-dhcp-relay`.

Dans un premier temps on considèrera que le réseau des serveurs est déclaré, sur le serveur dhcp, comme celui des utilisateurs. Cela va nous permettre de vérifier que le relai fonctionne.

I 4.3 Dans le fichier de configuration du service dhcp, copiez-collez les lignes déclarant le réseau 10.2.0.0 et adaptez les pour le réseau 10.1.0.0 avec les mêmes contraintes (10 machines max. et même durée max. de bail).
I 4.4 Redémarrez le service `dhcpd`.
I 4.5 Demandez un bail sur le serveur ntp pour vérifier qu'il obtient bien une IP.

Q 4.1 De qui le serveur ntp a-t-il reçu les messages OFFER et ACK ?

Nous allons maintenant observer les messages relayés par le relai.

I 4.6 Sur le serveur ntp, annulez le bail :

```
$ dhclient -r eth0
```

I 4.7 Tout en capturant les trames sur le serveur dhcp (avec wireshark), redemandez un bail sur le serveur ntp.
I 4.8 Sur le serveur dhcp, arrêtez la capture puis ajoutez le filtre bootp pour ne voir que les messages dhcp.

Q 4.2 En analysant le contenu du message DISCOVER reçu par le serveur dhcp (affichez pour cela le détail du message *Bootstrap Protocol* dans le pannneau du milieu de wireshark), trouvez l'information qui lui permet de déterminer que la requête vient d'un hôte du réseau 10.1.0.0 et pas du réseau 10.2.0.0 (et qu'il doit donc lui attribuer une IP dans le réseau 10.1.0.0). Ne regardez pas dans les options pour trouver cette information.

Enfin, nous allons faire en sorte que le serveur ntp ait une adresse IP fixe (10.1.0.101). Pour identifier le serveur ntp, le serveur dhcp se basera sur son adresse MAC.

I 4.9 Sur le serveur ntp, trouvez l'adresse MAC de l'interface eth0.
I 4.10 Ajoutez les lignes suivantes à la fin du fichier de configuration dhcp :

```
1 host ntp {
2   hardware ethernet __________;  # à remplacer par l'adresse MAC de ntp
3   fixed-address 10.1.0.101;
4 }
```

I 4.11 Redémarrez le service `dhcpd`.
I 4.12 Renouvelez le bail sur le serveur ntp (`dhclient -r` puis `dhclient`) et vérifiez qu'il obtient bien l'IP 10.1.0.101.

Q 4.3 Cette méthode d'identification (avec l'adresse MAC) est-elle sûre ? Autrement dit est-ce qu'un autre hôte peut obtenir l'IP 10.1.0.101 auprès du serveur dhcp ?

Exercice 5 — Configuration du serveur NTP

Le serveur dhcp peut fournir de nombreuses informations aux clients. Il peut par exemple leur indiquer un serveur NTP (*Network Time Protocol*) qui joue le rôle d'une horloge. Cela permet de synchroniser les horloges d'un réseau.

I 5.1 Recherchez dans le manuel des options dhcp (commande `man dhcp-options`), l'option permettant d'indiquer un serveur ntp aux clients qui en font la demande.
I 5.2 Ajoutez cette ligne dans fichier de configuration du service dhcp, à l'intérieur de la déclaration du réseau 10.2.0.0.
I 5.3 Redémarrez le service `dhcpd`.

Pour tester que l'horloge des clients se synchronisent bien sur l'horloge du serveur ntp, nous allons changer l'heure sur le serveur ntp.

I 5.4 Sur ntp, changez l'heure avec la commande `date` en ajoutant dix minutes à l'heure actuelle :

```
$ date -s HH:MM
```

Nous allons ensuite configurer le service ntp. Le fichier de configuration est `/etc/ntp.conf`. Sur ntp :

I 5.5 Supprimez puis recréez le fichier `/etc/ntp.conf`.
I 5.6 Éditez avec nano le fichier `/etc/ntp.conf` pour avoir le contenu suivant :

```
server 127.127.1.0 prefer
fudge 127.127.1.0 stratum 2
driftfile /var/lib/ntp/ntp.drift
disable auth
logfile /var/log/ntp.log
```

(Nous ne rentrerons pas dans les détails du fichier car il faudrait détailler le fonctionnement du protocole ntp. En gros cela permet que ntp soit un serveur de temps de confiance pour les hôtes du réseau.)
I 5.7 Redémarrez le serveur `ntp`.

Ensuite sur anna :

I 5.8 Redémarrez le serveur `ntp`. (Il faut aussi qu'un serveur ntp soit en exécution sur anna afin qu'il se synchronise périodiquement avec celui de l'hôte ntp.)
I 5.9 Démarrez une capture de trame avec wireshark.
I 5.10 Renouvelez le bail.
I 5.11 Arrêtez la capture puis entrez ntp comme filtre.
I 5.12 Vérifiez que la capture fait bien apparaître les requêtes NTP d'anna ainsi que les réponses du serveur ntp.

À l'obtention du bail, vous verrez normalement que le serveur ntp d'anna a été redémarré car il doit maintenant se synchroniser avec le serveur ntp fourni par le serveur dhcp. Il ne reste plus qu'à vérifier que la synchronisation a eu lieu.

I 5.13 Sur anna, affichez la date :

```
$ date
```

Il se peut que la synchronisation prenne du temps : plusieurs dizaines de secondes, voire plusieurs minutes. On se basera alors sur la capture de paquets NTP (instruction I 5.12) pour vérifier que la synchronisation fonctionne.

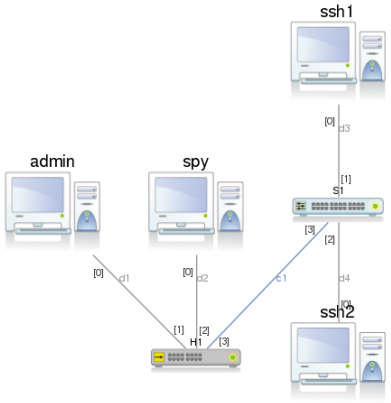
Exercice 6 — Configuration du client

On peut aussi configurer les clients pour paramétrer les requêtes qu'ils enverront au serveur dhcp. Le fichier de configuration du client dhcp est `/etc/dhcp/dhclient.conf`. En regardant le manuel de ce fichier ou en recherchant sur internet, trouvez (puis testez) comment faire en sorte que :

— le client demande une IP particulière au serveur (par exemple, si anna veut toujours utiliser 10.2.0.8) ;
— le client demande une longueur de bail particulière (par exemple une demi-heure) ;
— le client rejette une offre ne contenant pas une option particulière (par exemple, rejeter une offre qui ne contient pas l'adresse d'un routeur).

TP 2 — Le service SSH

L'objectif de ce TP est d'étudier les possibilités du protocole SSH (surtout côté client) et de se familiariser avec les mécanismes de chiffrement mis en œuvre dans SSH. On travaillera sur le réseau suivant :



Tous les hôtes font partie du réseau 10.1.0.0/24. On imagine que l'hôte admin est une machine d'administration et que ssh1 et ssh2 sont deux hôtes sur lesquels l'administrateur peut se connecter à distance par SSH. Enfin, spy est un hôte espion utilisé pour illustrer le fonctionnement des tunnels SSH (dernier exercice).

Exercice 1 — Création du réseau et configuration des interfaces

I 1.1 Ouvrez marionnet et créez un nouveau projet.
I 1.2 Ajoutez les équipements pour avoir le réseau de la figure en introduction. H est un hub et S un switch. (Attention au type de câble …)
I 1.3 Démarrez tous les équipements.
I 1.4 Ouvrez une session root (mot de passe=root) sur chaque hôte.

Toutes les interfaces seront configurées statiquement avec les IP suivantes :

hôte	IP
admin	10.1.0.1
spy	10.1.0.2
ssh1	10.1.0.100
ssh2	10.1.0.101

I 1.5 À l'aide de la commande `ip` (voir exercice 2 du TP 1), attribuez les IP aux interfaces.
I 1.6 Vérifiez que les hôtes peuvent bien s'échanger des messages ping.

Exercice 2 — Utilisation de la commande ssh

L'objectif de cet exercice est de se familiariser avec l'utilisation de la commande ssh. Pour que les hôtes ssh1 et ssh2 acceptent les connexions, il faut au préalable que le serveur `ssh` soit démarré.

I 2.1 Redémarrez le serveur `ssh` sur ssh1 et ssh2 :

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

Dans sa forme la plus simple, la commande `ssh` prend en argument l'IP de l'hôte sur lequel on veut se connecter.

I 2.2 Sur admin, ouvrez une session sur ssh1 :

```
$ ssh 10.1.0.100
```

Vous devriez alors voir un message vous indiquant que l'hôte 10.1.0.100 est inconnu, ce qui est normal puisqu'on s'y connecte pour la première fois. Une demande de confirmation vous est faite.

I 2.3 Entrez `yes` puis le mot de passe `root`.

Une fois le mot de passe validé, un message s'affiche dans le terminal d'admin (`The programs included ...`). C'est ce qu'on appelle une bannière, un message envoyé par le serveur à la connexion.

Q 2.1 Quel est maintenant le message d'invite dans la fenêtre d'admin ? Pourquoi ?
Q 2.2 Quel est le répertoire courant ?

I 2.4 Utilisez la commande suivante sur admin :

```
$ echo hello | wall
```

Elle permet d'afficher le message `hello` dans tous les terminaux.
I 2.5 Vérifiez que le message s'est bien affiché dans la fenêtre de ssh1.
I 2.6 Quittez la session ssh ouverte sur admin (ctrl+d ou `exit`).

On peut aussi se connecter avec un login particulier en précisant `login@` dans la commande `ssh`.

I 2.7 Sur admin, ouvrez une session `student` sur ssh1 :

```
$ ssh student@10.1.0.100
```

Notez que le message observé à la première connexion n'apparaît plus.

I 2.8 Entrez le mot de passe `student`.

Q 2.3 Quel est maintenant le message d'invite dans la fenêtre d'admin ? Pourquoi ?
Q 2.4 Quel est le répertoire courant ?

I 2.9 Quittez la session ssh ouverte sur admin (ctrl+d ou `exit`).

Quand on souhaite exécuter une unique commande sur un serveur ssh, on peut directement la passer en argument supplémentaire. Le client ssh se connecte au serveur, exécute la commande puis se déconnecte automatiquement.

I 2.10 Sur admin, exécutez la commande ci-dessous puis entrez le mot de passe :

```
$ ssh 10.1.0.100 touch /tmp/hello
```

Notez qu'après l'exécution de la commande, vous êtes toujours logé sur admin.

I 2.11 Sur ssh1, vérifiez que la commande `touch` a bien été exécutée.
I 2.12 Sur admin, exécutez la commande ci-dessous puis entrez le mot de passe :

```
$ ssh student@10.1.0.100 ls /root
```

Q 2.5 Cette dernière commande a-t-elle fonctionné ? Pourquoi ?

Exercice 3 — Utilisation de la commande scp

La commande scp (secure copy) permet de copier des fichiers à distance en utilisant le protocole SSH.
Nous allons créer quelques fichiers et répertoires sur admin pour la tester.

I 3.1 Sur admin, créez les fichiers et répertoires suivants (dans le répertoire courant) : `fic1`, `rep1`, `rep1/fic2`, `rep1/rep2` et `rep1/rep2/fic3`.

Nous allons ensuite copier ceux-ci à divers emplacements du système de fichiers de ssh1. Sur admin :

I 3.2 Exécutez les commandes suivantes :

```
$ scp fic1 10.1.0.100:
$ scp rep1/fic2 student@10.1.0.100:
$ scp rep1/rep2/fic3 10.1.0.100:/etc
$ scp -r rep1 10.1.0.100:
```

Q 3.1 À quoi sert l'option `-r` utilisée pour la dernière commande ?
Q 3.2 À quels emplacements ont été copiés les fichiers et répertoires sur ssh1, et quels sont leurs propriétaires ?

On peut aussi copier dans l'entre sens (du serveur vers le client).

I 3.3 Donnez et testez sur admin les commandes permettant de :
— Copier le fichier `/etc/dhcp/dhcpd.conf` sur ssh1 dans le répertoire courant.
— Copier (avec l'identité student) le répertoire personnel de student sur ssh2 dans le répertoire `/tmp`.

Exercice 4 — Observation du mécanisme de chiffrement

Dans cet exercice nous allons nous familiariser avec le fonctionnement des clés de chiffrement utilisées par le client et le serveur et les problèmes de sécurité qui peuvent survenir.

Pour rappel, SSH utilise à la fois un chiffrement asymétrique et un chiffrement symétrique. Le chiffrement asymétrique est utilisé en début de session, pour que le client et le serveur négocient une clé de session. La clé de session est une clé de chiffrement symétrique utilisée pour chiffrer toute la suite de la session (et en particulier le login et le mot de passe). Durant la première phase (chiffrement asymétrique) on a donc un échange de clés publiques entre le client et le serveur.

Les clés de chiffrement se trouvent dans le répertoire `/etc/ssh`. Pour chaque algorithme (ou type) de chiffrement `XXX` on a deux fichiers :
— `ssh_host_XXX_key` qui est la clé privée utilisée pour cet algorithme ;
— et `ssh_host_XXX_key.pub` qui est la clé publique.

I 4.1 Sur ssh1, listez le contenu du répertoire `/etc/ssh` en utilisant l'option `-l`.

Q 4.1 Quels sont les algorithmes de chiffrement supportés ?
Q 4.2 Quels sont les droits associés aux fichiers contenant les clés privées ? Et à ceux contenant les clés publiques ? Pourquoi ?

Quand un client se connecte à un serveur ssh pour la première fois, il stocke la clé publique du serveur dans le fichier `.ssh/known_hosts` de son répertoire personnel. Ainsi, le message indiquant que le serveur est inconnu ne s'affiche qu'à la première connexion (comme dans l'exercice 2). Aux connexions suivantes, le client compare la clé publique qu'il reçoit du serveur avec celle qu'il avait précédement stockée dans le fichier. Si les deux sont identiques, la connexion peut se faire. Par contre, si le client observe une différence entre les deux clés il refusera la connexion comme nous allons le voir.

I 4.2 Sur ssh1, générez une nouvelle paire de clés pour le service ssh :

```
$ ssh-keygen -t ecdsa -f /etc/ssh/ssh_host_ecdsa_key
```

Entrez `y` pour la première question (pour écraser le fichier existant), et laissez un passphrase vide (voir l'exercice suivant pour la notion de passphrase). Dans cette commande, on précise le type de chiffrement `ecdsa` qui est celui par défaut utilisé par ssh.

I 4.3 Sur admin, ouvrez une session sur ssh1.

Vous devriez voir un message `WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!`. En effet, le client constate que la clé publique qu'il vient de recevoir du serveur n'est pas celle qu'il avait mémorisée. Il vous prévient donc que :
— soit la clé a effectivement changé (*It is also possible ...*) ;
— soit un attaquant essaie de se faire passer pour le serveur (*Someone could be eavesdropping ...*) et vous a envoyé sa clé publique pour, par exemple, vous extorquer votre login et votre mot de passe.

Dans tous les cas, la commande se termine immédiatement et comme vous l'indique le message, il convient de contacter l'administrateur du serveur pour vérifier que la clé a bien changé.

Si c'est bien le cas, on peut supprimer du fichier `.ssh/known_hosts` la ligne contenant l'ancienne clé publique associée au serveur ssh. Sur admin :

I 4.4 Supprimez la clé publique de 10.1.0.100 :

```
$ ssh-keygen -R 10.1.0.100
```

I 4.5 Retentez une connexion ssh sur 10.1.0.100 pour vérifier que c'est maintenant possible.
I 4.6 Fermez la session ssh.

Exercice 5 — Authentification par clé

Un utilisateur peut aussi créer sa propre paire de clés et copier sa clé publique sur chaque hôte sur lequel il voudra se connecter par la suite. Plutôt que d'utiliser les mots de passe associés aux différents comptes qu'il possède sur ces hôtes il utilisera à la place le passphrase associé à sa paire de clés. On remplace donc plusieurs mots de passe (1 par hôte) par un seul passphrase (mot de passe de la clé). C'est le mécanisme d'authentification par clé.

I 5.1 Sur admin, générez une paire de clés :

```
$ ssh-keygen
```

Laissez la valeur par défaut pour toutes les questions qui vous sont posées (tapez directement sur entrée).

Ceci a créé une paire de clés dans le répertoire `/root/.ssh` avec un passphrase vide. On peut laisser un passphrase vide si l'hôte sur lequel on a généré les clés est sécurisé (physiquement et électroniquement) — auquel cas la clé privée ne peut pas être volée (le vol de clé privée permettrait au voleur de se connecter sans utiliser ni mot de passe ni passphrase).

Pour s'authentifier sur un serveur avec la paire créée, la clé publique générée doit être placée, sur le serveur, dans le fichier `/root/.ssh/authorized_keys`.

Sur admin :

I 5.2 Avec scp copiez la clé publique `/root/.ssh/id_rsa.pub` dans `/root/.ssh/authorized_keys` sur ssh1.
I 5.3 Faites de même sur ssh2.
I 5.4 Ouvrez une session ssh sur ssh1.

Le passphrase étant vide, vous devriez être directement connecté sur ssh1 sans avoir à saisir de mot de passe.

I 5.5 Fermez la session.
I 5.6 Suivez à nouveau les deux instructions précédentes mais cette fois-ci avec ssh2.

L'avantage du passphrase vide est aussi que l'on peut exécuter des scripts contenant des commandes ssh étant donné que ces commandes ne requièrent plus d'action de l'utilisateur (la saisie du mot de passe). Nous allons par exemple, écrire un script qui teste si le service dhcpd est lancé sur les deux serveurs ssh. Voici le code (incomplet) de ce script :

```
#!/bin/bash

for ip in 10.1.0.100 10.1.0.101  # boucle qui parcoure les IP des deux serveurs
do
   echo __________
   ssh __________
done
```

Le script devra afficher une ligne contenant l'IP du serveur puis l'état du service dhcpd. Voici un exemple d'exécution :

```
=== 10.1.0.100 ===
Status of ISC DHCP server: dhcpd is not running.
=== 10.1.0.101 ===
Status of ISC DHCP server: dhcpd is not running.
```

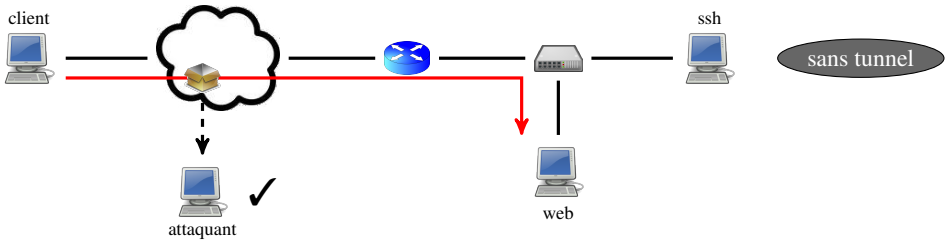
Sur admin :

I 5.7 Créez un fichier `test-dhcp.sh` et rendez le exécutable.
I 5.8 Éditez le avec nano pour avoir le contenu donné plus haut et complétez les blancs.
I 5.9 Testez votre script.

Exercice 6 — Création d'un tunnel ssh

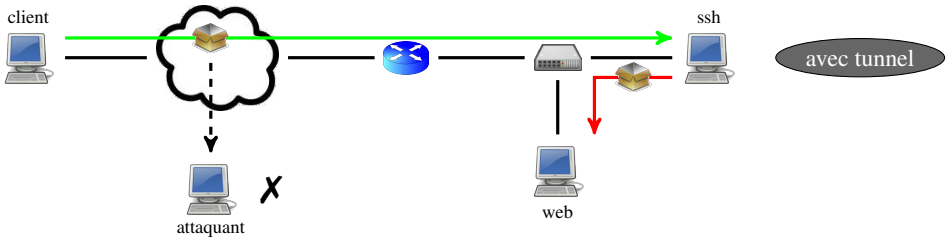
La technique du tunnel ssh permet de faire passer les paquets d'un protocole sans chiffrement (par ex, http) sur une connexion ssh pour bénéficier des possibilités de chiffrement offertes par ssh.

Prenons le réseau suivant :



Supposons que le client veuille accéder au serveur web par http. S'il se connecte directement au serveur, les paquets http qu'ils échangeront pourront éventuellement être interceptés et lus par un attaquant.

La technique du tunnel ssh consiste à ouvrir une connexion ssh entre le client et le serveur ssh (le tunnel) puis à faire passer les paquets http par cette connexion :



Ainsi, un paquet http envoyé par le client circulera d'abord *chiffré* du client vers le serveur ssh (paquet vert), puis *non chiffré* du serveur ssh vers le serveur web (paquet rouge). L'attaquant ne verra passer que des paquets chiffrés. Le serveur ssh joue donc le rôle d'intermédiaire entre le client et le serveur web. Bien entendu, un attaquant pourrait intercepter le trafic entre le serveur web et le serveur ssh. Il faut donc que la liaison entre ces deux serveurs soit sûre.

Nous allons mettre en place un tunnel entre admin et ssh1 pour qu'admin accède à des fichiers sur ssh2 (par http). Nous verrons dans les deux cas (avec et sans tunnel) les paquets capturés par spy.

I 6.1 Démarrez le service `apache2` (un serveur web) sur ssh2.
I 6.2 Démarrez une capture avec wireshark sur spy.
I 6.3 Sur admin, ouvrez le navigateur epiphany (sans oublier l'esperluette pour que le terminal vous rende la main) :

```
$ epiphany &
```

I 6.4 Dans epiphany, entrez l'URL `http://10.1.0.101/` puis validez.
I 6.5 Arrêtez la capture sur spy et trouvez un paquet contenant du code HTML envoyé par ssh2. (Le code devrait apparaître dans le panneau du bas, celui qui fait apparaître la trame sous forme brute.)

Laissez le navigateur ouvert pour la suite de l'exercice.

Nous allons maintenant mettre en place le tunnel.

I 6.6 Sur admin, lancez la commande ssh créant le tunnel :

```
$ ssh 10.1.0.100 -L 10000:10.1.0.101:80
```

L'argument `-L 10000:10.1.0.101:80` crée un tunnel connectant le port 10000 de l'hôte local (admin) au port 80 (port du service http) de l'hôte 10.1.0.101 (ssh2). Ainsi, tout processus qui communique avec le port 10000 de admin, communique en fait, grâce au tunnel, avec le port 80 de ssh2.

Nous allons maintenant observer le trafic capturé par spy une fois le tunnel mis en place.

I 6.7 Démarrez une capture avec wireshark sur spy.
I 6.8 Dans epiphany, rentrez maintenant l'URL `http://localhost:10000` puis validez. (Avec cette URL, epiphany se connecte au port 10000 de localhost (l'hôte admin lui-même).)
I 6.9 Arrêtez la capture sur spy.

Q 6.1 Que constatez-vous dans les paquets capturés ?

TP 3 — Configuration d'un serveur web

L'objectif de ce TP est de configurer un serveur web multi-sites et d'expérimenter certains modules d'apache.

Exercice 1 — Création du réseau

Aucun compte-rendu n'est demandé pour cet exercice.

I 1.1 Créez un réseau avec deux hôtes : client (un client web) et www (un serveur web) ; connectés par un câble croisé.
I 1.2 Démarrez chaque PC et ouvrez une session root.
I 1.3 Attribuez au client l'IP 10.0.0.1/24 et à www l'IP 10.0.0.100/24 et vérifiez qu'ils peuvent s'échanger des ping.

Dans la suite, toutes les instructions doivent être suivies sur www, sauf quand il est explicitement précisé le contraire.

Exercice 2 — Création du site www.p13.edu

Nous allons dans un premier temps créer le site `www.p13.edu`. Le rendu de sa page d'accueil est donné sur la figure de droite. Le lien hypertexte pointe sur l'URL `http://www.iutv.p13.edu`. Les fichiers du site seront stockés dans le répertoire `/var/www/p13`.

I 2.1 Créez le répertoire `/var/www/p13`.
I 2.2 Créez puis éditez un fichier HTML `/var/www/p13/index.html` avec le rendu de la figure ci-dessus.

Nous allons maintenant configurer le serveur apache pour déclarer le site `www.p13.edu`. Quand un serveur web héberge plusieurs sites web comme c'est le cas dans ce TP, le plus simple est de créer des *hôtes virtuels* dans apache. Voici l'hôte virtuel pour `www.p13.edu` :

```
<VirtualHost *:80>
   ServerName     www.p13.edu  # nom de domaine associé à l'hôte virtuel
   DocumentRoot  /var/www/p13  # répertoire contenant les fichiers du site.  par exemple,
                               # http://www.p13.edu/rep/page.html renvoie le fichier /var/www/p13/rep/page.html
   DirectoryIndex index.html   # page par défaut des répertoires.  par exemple,
                               # http://www.p13.edu/rep/  renvoie le fichier  /var/www/p13.edu/rep/index.html
</VirtualHost>
```

Les fichiers de configuration des hôtes virtuels sont dans `/etc/apache2/sites-available`.

I 2.3 Créez puis éditez le fichier `/etc/apache2/sites-available/p13.conf` pour qu'il ait le contenu ci-dessus.

Le répertoire `/etc/apache2/sites-enabled` contient quant à lui des liens symboliques vers les fichiers des hôtes virtuels du répertoire `sites-available` qui sont lus au démarrage d'apache.

I 2.4 Créez le lien symbolique `/etc/apache2/sites-enabled/p13.conf` pointant vers le fichier créé en I 2.3.
I 2.5 Redémarrez le serveur apache :

```
$ /etc/init.d/apache2 restart
```

Enfin le nom `www.p13.edu` n'est pas déclaré auprès d'un serveur DNS. Nous allons donc ajouter une ligne dans le fichier `/etc/hosts` du client permettant d'associer le nom `www.p13.edu` à 10.0.0.100.

Sur le client :

I 2.6 Éditez le fichier `/etc/hosts` pour ajouter la ligne suivante :

```
10.0.0.100 www.p13.edu
```

I 2.7 Ouvrez le navigateur epiphany et entrez l'URL `http://www.p13.edu` pour vérifier que la page HTML s'affiche.

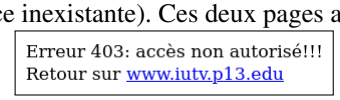
Exercice 3 — Création du site www.iutv.p13.edu

I 3.1 En suivant les étapes de l'exercice précédent, faites en sorte que le site `www.iutv.p13.edu` soit également servi par le serveur web. Ce site a les caractéristiques suivantes :
- fichier de conf. : `/etc/apache2/sites-available/iutv.conf`
- répertoire contenant ses fichiers : `/var/www/iutv`
- Le rendu de sa page d'accueil est donné sur la figure de droite. Le lien pointe sur l'URL `http://www.p13.edu`.

Dans la suite du TP nous ne travaillerons plus que sur ce site.

Exercice 4 — Fichiers d'erreur et journaux

Il est possible de définir la page qui sera retournée à l'utilisateur en cas d'erreur, en remplacement de la page par défaut retournée par apache. Nous allons créer deux pages : l'une pour l'erreur 403 (accès interdit), l'autre pour l'erreur 404 (ressource inexistante). Ces deux pages auront le rendu ci-dessous :



I 4.1 Créez le répertoire `/var/www/iutv/erreurs`.
I 4.2 Créez puis éditez dans ce répertoire un fichier `403.html` avec le rendu de la figure ci-dessus.
I 4.3 Faites de même avec le fichier `404.html`.

Il faut maintenant ajouter dans la configuration du site les directives permettant de retourner ces pages en cas d'erreur.

I 4.4 Dans `/etc/apache2/sites-available/iutv.conf`, ajoutez, dans la déclaration de l'hôte virtuel :

```
ErrorDocument 403 /erreurs/403.html
ErrorDocument 404 /erreurs/404.html
```

Les chemins fournis sont des chemins absolus ayant pour racine le répertoire racine du site (donc `DocumentRoot`).
I 4.5 Redémarrez le serveur apache.
I 4.6 Sur le client, faites les tests suivants dans epiphany :
- Pour l'erreur 404, il suffit d'entrer l'URL d'un nom de fichier inexistant sur le serveur, p.ex., `http://www.iutv.p13.edu/truc.html`.
- Pour l'erreur 403, vous pouvez retirer temporairement les droits en lecture sur le fichier `index.html` puis recharger l'URL `http://www.iutv.p13.edu` dans epiphany.

Nous allons aussi stocker les événements dans des journaux se trouvant dans le répertoire `/var/log/apache2/iutv`. Nous ferons en sorte que le serveur sauvegarde dans le fichier `access.log` chaque requête reçue ; et qu'il sauvegarde dans le fichier `error.log` chaque erreur (p.ex., 403, 404) relevée.

I 4.7 Créez le répertoire `/var/log/apache2/iutv`.
I 4.8 Dans `/etc/apache2/sites-available/iutv.conf`, ajoutez, dans la déclaration de l'hôte virtuel :

```
CustomLog /var/log/apache2/iutv/access.log common
ErrorLog  /var/log/apache2/iutv/error.log
```

Le premier fichier contiendra une ligne par requête reçue par le serveur web (`common` étant le format des messages stockés). Le second contiendra une ligne par erreur renvoyée par le serveur web.
I 4.9 Redémarrez le serveur apache.
I 4.10 Sur le client, faites les tests suivants dans epiphany :
- rechargez la page `http://www.iutv.p13.edu` ;
- puis rentrez l'URL d'un fichier inexistant (comme en I 4.6) pour générer une erreur 404.

I 4.11 Affichez les contenus des deux fichiers (`access.log` et `error.log`) pour vérifier que vous avez bien de nouvelles lignes suite aux deux requêtes envoyées par le client à la question précédente.

Exercice 5 — Activation des scripts CGI

On suppose que le répertoire `admin/cgi` du site contient des scripts écrits par l'administrateur qui souhaite pouvoir les exécuter via un navigateur.

I 5.1 Créez les répertoires `/var/www/iutv/admin` et `/var/www/iutv/admin/cgi`.
I 5.2 Créez dans ce dernier le fichier `test-services.sh` et rendez le exécutable.
I 5.3 Éditez ce fichier pour que ce soit un script CGI retournant du texte brut et affichant l'état des services apache2, cron, rsyslog et ssh. Utilisez de préférence une boucle plutôt que d'écrire 4 fois l'instruction `/etc/init.d/...`.
I 5.4 Testez votre script dans le terminal.

Il faut maintenant ajouter, pour le répertoire `cgi` des directives permettant l'exécution de scripts CGI dans ce répertoire.

I 5.5 Dans `/etc/apache2/sites-available/iutv.conf`, ajoutez, dans la déclaration de l'hôte virtuel :

```
<Directory /var/www/iutv/admin/cgi>  # définit des directives ne s'appliquant qu'à ce répertoire
   Options +ExecCGI                  # on autorise l'exécution de script CGI dans ce répertoire
   AddHandler cgi-script .sh         # tous les fichiers d'extension .sh sont des scripts CGI
</Directory>
```

Il faut également activer le module apache `cgi` afin de permettre l'exécution de scripts CGI. Comme pour les sites virtuels, cela se fait par la création de liens symboliques.

I 5.6 Créez le lien symbolique `/etc/apache2/mods-enabled/cgi.load` pointant vers le fichier `/etc/apache2/mods-available/cgi.load`.
I 5.7 Redémarrez le serveur apache.
I 5.8 Sur client, testez en entrant l'URL `http://www.iutv.p13.edu/admin/cgi/test-services.sh` dans epiphany.

Exercice 6 — Authentification

Apache inclut des modules d'authentification permettant de protéger l'accès à certains répertoires du site. Nous allons utiliser cette possibilité pour protéger l'accès au répertoire `admin`. Dans le mode d'authentification d'apache par défaut, les utilisateurs et mots de passe ne sont pas ceux du système, mais sont ceux stockés dans un fichier créé grâce à l'outil `htpasswd` inclus avec apache.

I 6.1 Créez un mot de passe pour `admin` :

```
$ htpasswd -c /etc/apache2/passwd admin
```

L'option `-c` permet de créer le fichier des mots de passe qui est, ici, `/etc/apache2/passwd`.
I 6.2 Faites de même pour `anna` (retirez le `-c` sinon le fichier est écrasé et le mot de passe d'`admin` perdu).

Nous pouvons maintenant activer l'authentification sur le répertoire `admin` pour que seul `admin` y ait accès.

I 6.3 Dans `/etc/apache2/sites-available/iutv.conf`, ajoutez, dans la déclaration de l'hôte virtuel :

```
<Directory /var/www/iutv/admin>
   AuthType Basic                    # méthode d'authentification (basic = par mot de passe apache)
   AuthUserFile /etc/apache2/passwd  # fichier contenant les mots de passe
   AuthName "Fichiers d'admin"       # message affiché dans le navigateur quand on accède au répertoire
   Require user admin                # il faut se connecter en admin pour accéder au répertoire
</Directory>
```

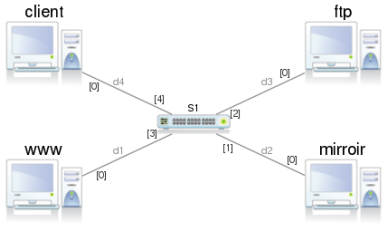
I 6.4 Créez le lien symbolique `/etc/apache2/mods-enabled/auth_basic.load` pointant vers le fichier `/etc/apache2/mods-available/auth_basic.load`. (Il se peut que le lien existe déjà auquel cas vous pourrez ignorer le message d'erreur.)
I 6.5 Redémarrez le serveur apache.
I 6.6 Sur le client, testez en entrant l'URL `http://www.iutv.p13.edu/admin` dans epiphany que :
— vous ne pouvez pas voir le contenu du répertoire avec le compte `anna` ;
— mais que ça fonctionne bien avec le compte `admin`.

TP 4 — Configuration d'un serveur FTP répliqué

L'objectif de ce TP est de mettre en place un serveur FTP (File Transfer Protocol) qui sera accessible par un proxy (ou serveur mandataire) puis répliqué sur un serveur miroir afin d'assurer une tolérance aux pannes (les données restent accessibles en cas de panne du serveur principal). On utilisera pour cela :

— le service proftpd pour le serveur FTP ;
— le service apache2 pour mettre en place le proxy ;
— et l'outil rsync (basé sur SSH) pour synchroniser les données entre le serveur FTP et le serveur miroir.

Le réseau sur lequel nous travaillerons est le suivant :



Exercice 1 — Création du réseau et configuration des interfaces

Le service proftpd n'est pas présent par défaut sur les machines virtuelles marionnet. Il faut donc l'installer. Comme l'opération est un peu longue, nous allons partir du projet à l'URL suivante :

`https://www.lipn.univ-paris13.fr/~evangelista/cours/R203/ftp.mar`

Il contient uniquement les hôtes ftp et miroir sur lequels proftpd est déjà installé.

I 1.1 Téléchargez le projet à l'URL indiquée.
I 1.2 Lancez marionnet et ouvrez le projet téléchargé.
I 1.3 Ajoutez les équipements pour obtenir le réseau de la figure en introduction.
I 1.4 Démarrez tous les équipements et ouvrez une session root (mot de passe = root) sur chaque hôte.

Toutes les interfaces seront configurées statiquement avec le masque /24 et les IP suivantes :

hôte	IP
client	10.1.0.1
www	10.1.0.100
ftp	10.1.0.101
miroir	10.1.0.102

I 1.5 À l'aide de la commande `ip` (voir exercice 2 du TP 1), attribuez les IP aux interfaces.
I 1.6 Vérifiez que les hôtes peuvent bien s'échanger des messages ping.

Exercice 2 — Test et configuration du serveur FTP

Dans un premier temps nous n'allons rien changer à la configuration du service proftpd sur l'hôte ftp. Par défaut, le serveur autorise les utilisateurs du système à se connecter. Nous allons nous familiariser avec les commandes ftp après avoir créé deux fichiers pour illustrer le fonctionnement de ces commandes.

I 2.1 Sur ftp, créez le fichier `/home/student/fic_serveur`.
I 2.2 Sur le client, créez le fichier `fic_client` dans le répertoire courant.
I 2.3 Sur client, connectez vous en ftp sur l'hôte ftp :

```
$ ftp 10.1.0.101
```

Entrez le nom `student` et le mot de passe `student`.

Une fois la session ftp ouverte, vous pouvez lancer des commandes ftp dans le terminal. Les commandes cd, ls, mkdir et pwd sont aussi des commandes ftp mais elles sont appliquées sur le serveur. Deux commandes permettent de télécharger ou d'envoyer des fichiers :

`get FIC` — télécharge le fichier `FIC` du serveur dans le répertoire courant du client
`put FIC` — envoie le fichier `FIC` du client dans le répertoire courant du serveur

I 2.4 Testez et commentez les commandes suivantes dans la session ftp du client :

```
ftp> pwd
ftp> get fic_serveur
ftp> mkdir rep
ftp> cd rep
ftp> put fic_client
ftp> exit
```

Les deux hôtes devraient avoir les fichiers `fic_client` et `fic_serveur`.

Q 2.1 Dans quel répertoire se trouvent les fichiers sur le client ?
Q 2.2 Dans quels répertoires se trouvent les fichiers sur ftp ?

Un serveur ftp peut donner accès aux utilisateurs ne disposant pas d'un compte avec le compte spécial *anonymous*. Par défaut, le serveur proftpd refuse cet accès. Nous allons l'activer. Sur l'hôte ftp :

I 2.5 Éditez le fichier de configuration du service proftpd (`/etc/proftpd/proftpd.conf`).
I 2.6 Trouvez et décommentez (supprimez le caractère dièse en début de ligne) dans le fichier la section permettant l'accès anonymous (`<Anonymous ~ftp> ... </Anonymous>`).
I 2.7 Redémarrez le service proftpd.

Quand un utilisateur se connecte en anonymous sur le serveur ftp, il a accès au répertoire `/srv/ftp` du serveur. Pour les tests ultérieurs nous allons créer quelques fichiers dans ce répertoire. Sur ftp :

I 2.8 Créez le répertoire `/srv/ftp/rt`.
I 2.9 Créez, dans ce nouveau répertoire, les fichiers `R201.txt`, `R202.txt` et `R203.txt`.
I 2.10 Éditez ces fichiers pour y ajouter du contenu (peu importe lequel).

On peut maintenant tester l'accès anonymous. Sur client :

I 2.11 Connectez vous en ftp sur l'hôte ftp. Entrez le nom `anonymous` et un mot de passe quelconque. (Aucun contrôle du mot de passe n'est effectué pour `anonymous`.)
I 2.12 Déplacez vous dans le répertoire `rt` du serveur.
I 2.13 Téléchargez le fichier `R203.txt`.
I 2.14 Essayez de déposer le fichier `fic_client` du client dans le répertoire courant du serveur.

Q 2.3 Est-ce que la dernière commande a fonctionné ? Pourquoi ?

I 2.15 Fermez la session ftp (`exit`).

Exercice 3 — Configuration du proxy

Nous allons maintenant configurer l'hôte www pour qu'il fonctionne comme un *proxy* FTP, c'est-à-dire un serveur intermédiaire entre un client et un serveur FTP. Pour télécharger des fichiers du serveur ftp, le client communiquera (en utilisant le protocole HTTP) avec www qui relaiera ensuite les requêtes vers le serveur ftp (en utilisant le protocole FTP). Le proxy a donc aussi une fonction de réécriture des requêtes HTTP en requêtes FTP ; et des réponses FTP en réponses HTTP. L'avantage de cette technique est qu'elle permet de faire passer les requêtes et réponses par un point unique (le proxy) permettant, p.ex., de mettre facilement en place des politiques de sécurité (comme filtrer certaines requêtes) ou des techniques de répartition de charges (comme dans l'exercice 5).

Apache inclut des fonctionnalités de proxy. Pour cela, il est d'abord nécessaire d'activer les modules `proxy` et `proxy_ftp` en créant des liens symboliques (comme dans le TP précédent). Sur www :

I 3.1 Créez le lien symbolique `/etc/apache2/mods-enabled/proxy.load` pointant vers le fichier `/etc/apache2/mods-available/proxy.load`.
I 3.2 Faites de même avec le fichier `proxy_ftp.load`.
I 3.3 Faites de même avec le fichier `proxy_ftp.conf`.

Passons maintenant à la configuration du proxy. Il faut pour cela créer un hôte virtuel. On suppose que les clients qui voudront accéder au serveur ftp via le proxy entrerons l'URL `http://ftp.iutv.edu`. Sur www :

I 3.4 Créez le fichier `/etc/apache2/sites-available/ftp`.
I 3.5 Éditez ce fichier pour qu'il ait le contenu suivant :

```
1 <VirtualHost *:80>
2    ServerName ftp.iutv.edu
3    ProxyPass / ftp://10.1.0.101/
4 </VirtualHost>
```

(N'oubliez pas le / final à la ligne 3.)

C'est bien sûr la troisième ligne du fichier qui passe le serveur www en mode proxy pour toutes les requêtes adressées au site `ftp.iutv.edu`. Cette ligne signifie que toutes les requêtes dont l'URI commence par / (donc toutes les requêtes) sont retransmises à l'hôte 10.1.0.101 (ftp) en utilisant FTP comme protocole.

Il reste maintenant à activer l'hôte virtuel en créant un lien symbolique, comme dans le TP précédent. Sur www :

I 3.6 Créez le lien symbolique `/etc/apache2/sites-enabled/ftp` pointant vers le fichier `/etc/apache2/sites-available/ftp`.
I 3.7 Redémarrez le serveur apache2.

On peut maintenant tester le proxy. Sur le client :

I 3.8 Rajoutez dans `/etc/hosts` une ligne associant l'IP 10.1.0.100 et le nom `ftp.iutv.edu`.
I 3.9 Lancez le navigateur web epiphany et entrez l'URL `http://ftp.iutv.edu`.
I 3.10 Vérifiez que vous pouvez bien aller dans le répertoire `rt` et télécharger le fichier `R203.txt`.

Notez que, par défaut, le proxy se connecte avec le compte anonymous au serveur ftp. Ce qui convient dans notre cas.

Exercice 4 — Prise en main de l'outil rsync

L'outil rsync (remote synchronisation) permet de synchroniser le contenu de deux répertoires distants (placés sur des hôtes distincts), c'est-à-dire de s'assurer que les deux répertoires ont le même contenu. Il repose sur le protocole ssh pour copier à distance. Comme nous allons l'utiliser sur l'hôte miroir pour synchroniser le contenu de son répertoire `/srv/ftp` avec celui de l'hôte ftp il faut que le serveur ssh soit démarré sur l'hôte ftp.

I 4.1 Sur ftp, démarrez le service ssh.
I 4.2 Sur miroir, lancez une synchronisation du répertoire `/srv/ftp` avec celui de l'hôte ftp (sans oublier les / à la fin des chemins) :

```
$ rsync -v -a 10.1.0.101:/srv/ftp/ /srv/ftp/
```

rsync ayant besoin d'ouvrir une session ssh sur ftp il faut alors rentrer le mot de passe de root.

Quelques explications sur la commande que l'on vient d'exécuter :
— L'option `-v` active le mode *verbeux* permettant d'afficher la liste des fichiers transférés et diverses informations (le débit, le nombre d'octets envoyés et reçus, ...).
— L'option `-a` active le mode *archive* qui est un raccourci pour plusieurs autres options (conservation des dates des fichiers, des droits des fichiers, parcours récursif des répertoires, ...).
— `10.1.0.101:/srv/ftp/` désigne le répertoire source (sur l'hôte ftp).
— `/srv/ftp/` désigne le répertoire destination (local, donc sur l'hôte miroir).
La synchronisation effectuée par rsync est incrémentale : rsync copie uniquement les fichiers nouveaux ou modifiés.

I 4.3 Relancez la commande de l'instruction I 4.2

Observez qu'aucun fichier n'est transféré. Nous allons refaire le test après avoir apporté des modifications au contenu du répertoire `/srv/ftp` sur l'hôte ftp.

I 4.4 Sur ftp, créez le fichier `/srv/ftp/rt/R204.txt`.
I 4.5 Sur ftp, modifiez le contenu du fichier `/srv/ftp/rt/R203.txt`.
I 4.6 Relancez la commande de l'instruction I 4.2

Observez que seuls le fichier créé et le fichier modifié ont été transférés.

Exercice 5 — Configuration et test du serveur miroir

Nous allons maintenant configurer l'hôte miroir pour qu'il ait le même contenu que le serveur ftp, ceci afin qu'il puisse remplacer le serveur ftp si celui-ci tombe en panne. Dans l'exercice précédent nous avons déjà copié le contenu du serveur ftp (le répertoire `/srv/ftp`) sur le miroir. Il faut également que les fichiers de configuration ftp soient identiques. Sur miroir :

I 5.1 Copiez, avec scp, le fichier `/etc/proftpd/proftpd.conf` du serveur ftp dans le répertoire `/etc/proftpd`.
I 5.2 Redémarrez le serveur proftpd.

Nous avons maintenant deux serveurs ftp rigoureusement identiques : l'un sur l'hôte ftp, l'autre sur l'hôte miroir.

L'étape suivante est de configurer le serveur apache sur www pour qu'il s'adresse au serveur miroir en cas de panne du serveur principal. Il faut tout d'abord activer un autre module : `proxy_balancer`. Ce module permet de mettre en place des stratégies de *répartition de charges* (ou load balancing) afin de répartir les requêtes sur un ensemble de serveurs pour, p.ex., éviter de surcharger un seul serveur. Sur www :

I 5.3 Créez le lien symbolique `/etc/apache2/mods-enabled/proxy_balancer.load` pointant vers le fichier `/etc/apache2/mods-available/proxy_balancer.load`.
I 5.4 Faites de même avec le fichier `proxy_balancer.conf`.
I 5.5 Éditez le fichier de configuration d'apache :
— commentez la ligne `ProxyPass ...` qu'on avait écrite précédemment ;
— et ajoutez les lignes suivantes dans la définition de l'hôte virtuel :

```
1 ProxyPass / balancer://serveurs/
2 <Proxy balancer://serveurs/>
3    BalancerMember ftp://10.1.0.101
4    BalancerMember ftp://10.1.0.102 status=+H
5 </Proxy>
```

I 5.6 Redémarrez le serveur apache2.

Quelques explications sur les lignes ajoutées au fichier de configuration d'apache :
— La ligne 1 indique que toute requête est redirigée vers un ensemble de serveurs. `serveurs` est un nom quelconque que nous donnons à cet ensemble de serveurs.
— Les lignes 2 à 5 définissent les serveurs vers lesquels une requête pourra être redirigée. On voit donc les deux IP de ftp et de miroir.
— Pour le serveur miroir (ligne 4), `status=+H` signifie que le serveur ne sera utilisé que si aucun autre serveur n'est disponible. Le serveur www retransmet donc en priorité les requêtes vers ftp et, seulement en cas de panne de ftp, vers miroir.

On peut maintenant tester :

I 5.7 Sur le client, lancez epiphany et entrez l'URL `http://ftp.iutv.edu/rt/R203.txt`.
I 5.8 Sur ftp, arrêtez le service proftpd.
I 5.9 Sur le client, rechargez la page.

Si le contenu du fichier `R203.txt` s'affiche correctement, c'est que le miroir a bien pris le relai suite à l'arrêt du serveur principal.

Exercice 6 — Mise à jour automatique du serveur miroir

Dans l'exercice 4 nous avons exécuté manuellement la commande rsync pour synchroniser le contenu du serveur miroir avec le serveur ftp. En pratique, la synchronisation se fait automatiquement (sans intervention de l'administrateur).

Mettez en œuvre et testez une solution permettant d'effectuer automatiquement cette synchronisation à intervalle régulier (p.ex., toutes les heures).

Indication. La commande rsync de synchronisation doit pouvoir s'exécuter automatiquement (sans qu'un utilisateur ait à saisir un mot de passe). Il faut donc mettre en place une méthode d'authentification par clé comme nous l'avions fait dans l'exercice 5 du TP 2.