Protocoles réseaux
TP n° 1 : services réseaux

I) connexion à l'UFR

L'UFR offre plusieurs serveurs, par exemple `lulu` (sous Linux) et `nivose` (sous SunOS). Les stations de travail que vous avez devant vous (si vous êtes en salle 2031) sont sous Linux. Lisez la page

`http://www.informatique.univ-paris-diderot.fr/wiki/doku.php/wiki/linux`

pour savoir comment utiliser les serveurs Linux. Les TPs sont prévus pour être fait sous Linux. Si vous avez un Mac il peut y avoir des différences, mais les TPs restent faisables. Si vous êtes sous Windows vous aurez nettement plus de problèmes...

Remarque importante, si vous avez votre propre ordinateur portable, vous êtes *à l'extérieur* de la salle de TP, d'un point de vue réseau ! Car vous ne pouvez vous connecter que par Wi-Fi et arrivez sur un réseau différent de celui de la salle machine. Or pour certains TPs (et en particulier, celui-ci) il faut être *à l'intérieur* et donc, il vous faut maîtriser la connection distante aux serveurs de l'UFR, qui est décrite ici

`http://www.informatique.univ-paris-diderot.fr/wiki/doku.php/wiki/howto_connect`

Au menu, passage par la passerelle `lucette/lucy` et génération de clef SSH...

Exercice 1 :

Connectez-vous à une machine de l'UFR (soit sur un ordinateur d'une salle de tp de l'ufr, soit sur `lulu` via ssh). Vous devez être récompensé par un «*QUOI DE NEUF ?*»

Exercice 2 : Premières expériences avec les services réseau

1. À l'aide de la commande « `host` », déterminez l'adresse IP de la machine

   `www.informatique.univ-paris-diderot.fr`

   Quel est le nom d'hôte associé à l'adresse obtenue ?
2. Déterminez l'adresse IP de `www.free.fr`.
3. Déterminez le port associé au service *daytime*.
4. Téléchargez, puis lisez la RFC 867.
5. À l'aide de la commande « `telnet` », déterminez l'heure qu'il est sur `monjetas`. Même question avec la commande « `nc` ».
6. À l'aide de la commande « `nc` », testez les services ouverts sur `nivose` et sur `lulu` et qui vous attendent sur les *well-known ports* (quésaco ?)
7. À l'aide de la commande « `nc` », testez le service *time* sur `monjetas`. Que constatez-vous ? Allez lire la RFC 868 pour comprendre.

II) Le protocole HTTP

Le protocole HTTP est le protocole utilisé par le *web*. HTTP/1.0 était un protocole requête-réponse très simple : le client se connectait à la machine, envoyait une requête terminée par une ligne blanche, puis attendait la réponse. HTTP/1.1 est un protocole un peu plus compliqué, qui permet l'envoi de plusieurs requêtes sur la même connection ; il faut donc pouvoir déterminer la fin de la réponse du serveur.
Une fois la connexion établie (en utilisant `telnet` par exemple), la seule requête dont vous aurez besoin est la requête `GET`. La première ligne est composée du mot-clé `GET` suivi d'un espace, du chemin de la page à laquelle on veut acceder, d'un espace, et de la version du protocole :

```
GET /wiki/doku.php/enseignement/les-masters/ HTTP/1.1
```

Lorsque l'on utilise HTTP/1.1, cette requête doit être suivie de lignes contenant des entêtes de la forme *nom : valeur*. Une entête dont vous aurez besoin est l'entête `Host`, qui spécifie le nom du serveur et qui est obligatoire en HTTP/1.1. La liste des entêtes est terminée par une ligne blanche.

Exercice 3 : Le web à la main

1. À l'aide de la commande « `telnet` », faites une requête `HTTP/1.0` pour la page

`http://www.informatique.univ-paris-diderot.fr/`

Cette page contient juste une redirection vers une autre page.

2. Comment faire pour faire une autre requête `HTTP/1.0` ?

3. Refaites les deux questions précedentes avec des requêtes `HTTP/1.1`, la première persistante et la deuxième non persistante.

4. Dans un navigateur web, téléchargez la page

`http://localhost:1234/`

Que se passe-t-il ? Maintenant, dans un terminal libre, lancez la commande `nc -l localhost 1234`. Puis, re-téléchargez la page

`http://localhost:1234/`

Que se passe-t-il ? Dans le terminal, tapez ce qu'il faut pour afficher « `Hello, world!` » en gras dans le navigateur.

III) Le protocole SMTP

Le protocole SMTP (*Simple Mail Transfer Protocol*, RFC 821, port 25) sert à envoyer du courrier électronique à des utilisateurs locaux ou distants. Il s'agit d'un protocole de structure *requête-réponse*, dans lequel le client envoie une commande au serveur puis attend la réponse de ce dernier.

> **Remarque :** les techniques utilisées dans cette partie permettent, en théorie, d'envoyer des mails en se faisant passer pour quelqu'un d'autre. Ce serait non seulement discourtois, mais aussi illégal, et je vous déconseille fortement de sous-estimer les capacités d'investigation de nos administrateurs système.

Les commandes SMTP les plus utiles sont les suivantes :

« `HELO` » *machine* : à envoyer au début d'une connection SMTP, juste après avoir reçu l'invite. La chaîne *machine* est le nom d'hôte à partir duquel vous vous connectez.

« `MAIL FROM:` » *utilisateur* : commence une transaction SMTP visant à envoyer un message. La chaîne *utilisateur* est l'adresse de l'auteur du message (de la forme *user@domain*).

« `RCPT TO:` » *utilisateur* : déclare un destinataire de la transaction courante ; peut être repétée plusieurs fois.

« `DATA` » : déclare le début du message, qui commence sur la ligne suivante et se termine par un point `.` tout seul sur une ligne. Cette commande a pour effet de terminer la transaction courante.

« `QUIT` » : termine une connection SMTP.

Le message passé à la commande « `DATA` » doit avoir le format défini par le RFC 822. Il doit commencer par une série d'entêtes (« `From:` », « `To:` » , « `Subject:` » , etc...) suivis d'une ligne vide suivie du corps du message lui-même.

Exercice 4 : Le courrier à la main

1. En vous connectant directement au port SMTP d'une machine à l'aide de la commande « `telnet` » , envoyez un mail à l'un de vos collègues.
2. Répétez l'expérience précédente en donnant des addresses différentes dans l'enveloppe (commande « `RCPT TO:` » de SMTP) et dans le message (entête « `To:` » de RFC 822). Que se passe-t-il ?