PR6 – Programmation réseaux
TP n° 9 : Le protocole UDP

I) Un récepteur UDP

Exercice 1 :

Écrivez en Java un programme qui reçoit sur un port des datagrammes UDP et qui en affiche le contenu précédé de l'adresse de la socket émettrice, c'est-à-dire l'adresse IP de la machine et le numéro du port. Par exemple :

```
127.0.0.1:56472  Bonjour!
127.0.0.1:64659  Salut!
...
```

Supposer que la taille du message contenu dans les datagrammes ne dépasse pas 1024 caractères.

Exercice 2 :

Écrivez le même programme en C.

Exercice 3 :

Modifiez vos programmes de façon à ce que, plutôt qu'afficher le contenu des datagrammes, ils les renvoient à l'expéditeur, avec le même format que précédemment.

II) Un client UDP

On se propose d'écrire un client pour le serveur précédent.

Exercice 4 :

Dans un premier temps, le client doit lire un message sur l'entrée standard, l'envoyer au serveur sous forme d'un paquet UDP et attendre ensuite l'écho du message de la part du serveur. Pour cela, a-t-on besoin de lier la socket UDP d'un tel client ?
A la réception du message du serveur, le client l'affiche et son travail est fini. Doit-il se déconnecter ?
Écrivez ce client en C et en Java.

Exercice 5 : bonus

Maintenant, écrivez, en C ou en Java au choix, un client un peu plus sophistiqué qui se comporte de la manière suivante :

- il lit les messages sur l'entrée standard et les envoie sous forme de datagrammes UDP au serveur, et
- *en même temps*, il lit les messages qu'il reçoit du serveur et les affiche sur la sortie standard.

III) Un serveur UDP de messagerie textuelle

Exercice 6 : bonus

Modifiez un de vos récepteurs UDP de façon à ce que :

- il garde une liste des adresses de toutes les sockets de la part desquelles il a reçu des datagrammes (mais chaque adresse seulement une fois), et
- à la réception de chaque message, il le réexpédie, précédé de l'adresse de la socket émettrice comme avant, à toutes les adresses dans cette liste.