

## PR6 – Programmation réseaux

### TP n° 10 : Multi-diffusion

Dans ce TP, nous réécrivons un serveur de tchat, mais cette fois-ci à l'aide de serveurs de diffusion.

#### I) Diffusion intégrale

##### Exercice 1 : Le serveur

Écrire, en Java et en C, un programme qui reçoit sur le port 12321 des datagrammes UDP et qui en diffuse “intégralement” le contenu précédé de l'adresse de l'émetteur sur le port numéro 10101 de chaque machine du réseau.

On supposera que la taille du message contenu dans les datagrammes ne dépasse pas 1024 caractères. Le format des messages envoyé par le serveur sera l'adresse IP de l'émetteur suivie d'une tabulation et du contenu du message reçu par le serveur, par exemple :

172.28.46.176 Good morning!

##### Exercice 2 : Le client

Écrire, en C ou en Java au choix, un client pour ce serveur :

- le client lit les messages sur l'entrée standard et les envoie à partir d'un port quelconque sous forme de datagrammes UDP au serveur, et
- *en même temps*, il lit les messages qu'il reçoit du serveur sur le port 10101 et les affiche sur la sortie standard.

Que se passe-t-il quand plusieurs serveurs diffusent leurs messages sur le port 10101 ? Que se serait-il passé si les serveurs écoutaient sur le même port que les clients ?

##### Exercice 3 : UI – bonus

Pour améliorer l'interface utilisateur du client, vous pouvez afficher les messages que vous recevez du serveur sur un terminal autre que le terminal sur lequel vous lisez les messages tapés au clavier.

Pour ce faire, il faut que vous ouvriez un autre terminal, que vous récupériez le lien sur son fichier IO à l'aide de la commande `tty` et que vous écriviez dans ce fichier.

#### II) Multi-diffusion

##### Exercice 4 : Le serveur

Modifier le serveur de sorte qu'il envoie les messages, qu'il reçoit maintenant sur le port 12121, à une adresse de diffusion sur le port 10201.

##### Exercice 5 : Le client

Modifier le client afin qu'il s'abonne à l'adresse de diffusion et y reçoive les messages.