

TP4 Sockets UDP -Multicast

Exercice 1: (Unicast UDP)

Il vous est demandé d'utiliser le protocole UDP en unicast pour envoyer un paquet vers un destinataire donné.

1. Ecrire la classe **Receiver** qui écoute sur un port donné (exemple du port: 7676) et affiche le message reçu
2. Ecrire la classe **Sender** qui envoie un message (bonjour tout le monde) sous forme de chaîne de caractères, vers une machine qui écoute sur un port donné (identique au port d'écoute du receiver)
3. Tester l'application en démarrant le **Receiver** en premier et en lançant plusieurs
4. Modifier la classe **Receiver** afin de préciser le port grâce aux arguments main (exemple du port: 7676) et affiche le message reçu
 - Si on exécute la classe directement sur une invité de commande, on tapera la commande :


```
java Receiver 7676
```
 - Si on travaille sous Eclipse, il faut donner les arguments avant d'exécuter le programme :
 - i. Sélectionner le menu Run-> "Run Configurations",
 - ii. Choisir l'onglet (x)=Arguments
 - iii. Préciser les arguments dans la zone "Program arguments", (7676 de notre exemple).
5. Modifier la classe **Sender** qui envoie un proverbe aléatoire sous forme de chaîne de caractères, vers une machine qui écoute sur un port saisi au clavier.
6. Tester l'application à nouveau en démarrant le **Receiver** en premier et en lançant plusieurs **Sender**

Exercice 2: (Proverbe du jour en mode Unicast UDP)

Il vous est demandé d'utiliser le protocole UDP en unicast pour envoyer un proverbe par un **Sender** vers un **Receiver**. Le choix du proverbe se fait de façon aléatoire depuis une liste existante.

1. Ecrire la classe **Receiver** qui écoute sur un port donné (exemple du port: 7676) et affiche le message reçu
2. Ecrire la classe **Sender** qui envoie un proverbe aléatoire sous forme de chaîne de caractères, vers une machine qui écoute sur un port donné (même port utilisé par le **Receiver**).
3. Tester l'application en démarrant le **Receiver** en premier et en lançant plusieurs **senders**.

Exercice 3 (Proverbe du jour en mode Multicast UDP)

Le but est de modifier le plus légèrement possible les deux programmes (Sender et Receiver) de l'exercice précédent afin de réaliser les deux programmes suivants :

- Une classe **MulticastReceiver** en mode UDP Multicast sur l'adresse 224.0.0.1 port 8686. Tous les paquets reçus sont affichés à l'écran (en chaînes de caractères)
- Une classe **MulticastSender** qui envoie, aux machines enregistrées sur l'adresse multicast 224.0.0.1 port 8686, un paquet UDP contenant un proverbe aléatoire.

Indication : La classe MulticastSocket permet d'envoyer ou de recevoir des paquets multicast.

Exercice 4 (Application de chat)

L'objectif est de créer une **application de chat** en utilisant une adresse multicast commune à tous les clients. Un client est lancé avec un pseudo (pour connaître l'émetteur du msg)

1. Ecrire la classe **ClientChat** qui permet d'émettre (saisie au clavier) et de recevoir (affichage sur console) des paquets UDP.

Exemple de message reçu **Ahmed : message**

2. Les fonctions de lecture et écriture des paquets doivent être exécutées de façon simultanée

Indications :

- La classe MulticastSocket permet d'envoyer ou de recevoir des paquets multicast.
- Implémentation des processus légers via La classe Thread ou bien l'interface Runnable.