Réseau

TD n°2 — *TD Socket UDP*

Exercice 1. Client/Serveur en UDP

1.1 Écrire un programme client/serveur en utilisant cette fois les sockets en UDP. Le client envoie sa requête (peut importe le contenu), et le serveur renvoie la date actuelle au client, qui l'affiche avec l'adresse du serveur

1.2 Faire évoluer le programme. Le serveur envoie une réponse différente en fonction de la requête du client :

- date : la date actuelle est envoyée
- os : le système d'exploitation
- user : le nom d'utilisateur

La requête du client est donnée en argument du programme. Si la requête est différente des trois précédentes, un message d'erreur est envoyé par le serveur.

Exercice 2. Broadcast

2.1 Le but de cet exercice est d'envoyer la date toutes les secondes, à toutes les machines du réseau local, qui l'affichent.

- Faire une classe qui envoie en broadcast (255.255.255.255) la date du jour, toutes les secondes.
- Faire une classe qui récupère via l'adresse de broadcast le datagramme, et affiche la date contenue. Pensez à activer le broadcast sur la socket, via la commande `socket.setBroadcast(true)`

Exercice 3. Multicast

3.1 Le but de cet exercice est de saisir une chaîne de caractères au clavier, et de l'envoyer à un groupe de clients, grâce au multicast.

- Faire une classe qui lit une chaîne au clavier, et l'envoie dans un datagramme, sur une adresse de multicast. Utilisez un `DatagramSocket` pour l'envoyer
- Faire une classe qui lit la chaîne sur cette même adresse de multicast. Vous devez créer une `MulticastSocket`, et cette socket doit rejoindre le groupe avec `joinGroup(InetAddress multicas`
  La lecture se fait dans une boucle infinie, et fonctionne comme une socket udp classique. Vous affichez la chaîne reçue.

Réseau (TD n°2)

Les adresses multicast sont comprises entre 224.0.0.0 à 239.255.255.255 (les adresses 224.0.0.0 à 224.0.0.255 sont réservées)

Rappels :

Constructeur de paquet à envoyer :

```
new DatagramPacket
   (buffer,buffer.length,
         InetAddress.getByName("127.0.0.1"),port)
```

`new DatagramPacket(buffer, buffer.length)` : Constructeur de paquet en réception

`new DatagramSocket(port)` : Constructeur de Socket UDP

`main(String args[])` : `args[0]` premier argument, `args[1]` deuxième argument . . .

`dP.getData()` : récupère à partir d'un datagramPacket (dP) la donnée

`dP.getAddress()` : récupère l'adresse de l'expéditeur d'un datagramPacket (dP)

`Date date = new Date()` : retourne la date actuelle (la méthode `toString()` permet de la convertir en chaîne de caractères)

`System.getProperty("os.name")` : retourne le système d'exploitation (chaîne de caractères)

`System.getProperty("user.name")` : retourne le nom d'utilisateur (chaîne de caractères)

`"chaine".getBytes()` : convertit une chaîne de caractères en tableau d'octets

`new String(byte[] bytes)` : créer une chaîne de caractères à partir d'un tableau d'octets