TP Reseaux #1 : UDP & Multicast

Gilles Grimaud, Sébastien Jean, Laurent Noé, Marie-Hélène Verron

Le but de TP est de réaliser un client de discussion qui utilise le protocole UDP en *Multicasting*. Ce *service* de discussion en ligne ne nécessitera donc pas la présence d'un serveur centralisant toutes les communications.

Exercice 1 : Protocole UDP

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

Réaliser et tester (seul la même machine, puis entre deux machines) les deux programmes suivants :

- un programme qui écoute sur un port UDP donné, et affiche les messages reçus sous la forme de chaînes de caractères. Exemple d'utilisation : **`java ReceiveUDP 1500`**
- un programme qui envoie un message donné sous forme de chaîne de caractères, sur un port donné d'une machine donnée. Exemple d'utilisation : **`java SendUDP a13p13 1500 "you are welcome"`**

Vous aurez besoin des classes `java.net.DatagramSocket` et `java.net.DatagramPacket` ainsi que de la méthode `getBytes()` de `java.lang.String` et le constructeur `String(byte[])`

Exercice 2 : Multicast UDP

Modifier le plus légèrement possible les deux programmes de l'exercice précédent afin de réaliser les deux programmes suivants :

- un programme ouvrant une connexion UDP *Multicast* sur l'adresse `224.0.0.1` port `7654`. Tous les paquets reçus sont affichés à l(écran en les convertissant en chaînes de caractères ;
- un programme envoyant, à l'attention des machines enregistrées sur l'adresse multicast `224.0.0.1` port `7654`. un paquet UDP contenant un message quelconque.

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

Q1. Quelles sont les étapes pour émettre et recevoir un paquet UDP multicast sur le réseau local ?
Q2. Quelles sont les exceptions à traiter pour chaque étape du programme d'émission et de réception ?

Exercice 3 : Programme de Tchat

On souhaite maintenant réaliser un programme qui réunit les deux fonctions simultanément. On s'intéressera donc à la classe `java.lang.Thread` (ou l'interface `Runnable`, au choix)

Q1. Comment réaliser un client capable d'émettre et de recevoir des paquets UDP simultanément ?
Q2. Chaque message arrivant de machines différentes, on souhaite pouvoir associer un nom symbolique à chaque machine. Proposer une solution pour que chaque message soit associé au minimum à un nom de machine, et au mieux d'utilisateur afin qu'il soit plus facile de suivre les conversations.