

TP réseau - Java

Sockets UDP - Partie 1

Frédéric Armetta (frederic.armetta@univ-lyon1.fr)

1 Sockets UDP en java

Se documenter :

- Documentation en ligne - librairie standard JCL
(url : <http://docs.oracle.com/javase/7/docs/api/>)
- Complément de cours : JAVA_TCPUDP.pdf sur Spiral

Principales classes :

- Les classes utiles sont regroupées dans le package **java.net**
- classe `DatagramSocket` : permet d'émettre et recevoir des données (par l'intermédiaire de `DatagramPackets`)
- classe `DatagramPacket` : deux utilisations :
 - Émission : définir les données à envoyer, l'IP et le port du destinataire
 - Réception : récupérer les données, l'IP et le port de l'émetteur
- classe `InetAddress` : manipulation des adresses IP

Question 1 *Imaginer un scénario simple de communication Client/Serveur. Donner son diagramme de séquence (deux lignes de vie Client et Serveur, ainsi que les messages échangés) en précisant :*

- les objets d'écoute et d'émission (`DatagramSocket`) et leur paramétrage (port et suivant le cas `@IP`)
- les objets enveloppant les données (`DatagramPacket`) et leur paramétrage (données, `@IP :port`)

Question 2 *Préciser le code java associé aux principales parties du scénario (initialisation sockets, création datagrammes, émission, réception)*

2 Un scanner de ports UDP

Question 3 *Développer un scanner de ports UDP, sous la forme d'une fonction statique prenant en paramètre la plage de ports à scanner. Le scanner retourne ouvert/fermé pour chacun des ports. Pour tester individuellement un port, utiliser l'exception levée lors de l'appel de l'ouverture d'un `DatagramSocket` sur un port occupé.*

3 Un client serveur "UDP RX302"

Dans cette partie, vous développez un serveur UDP RX302 et un client du serveur RX302 donc le principe de fonctionnement est le suivant :

Le serveur attend une requête du client de la forme "hello serveur RX302". Lorsqu'il l'a reçue, il affiche à l'écran le message "Nouveau client : @IP :port" et renvoie au client une phrase standard "Serveur RX302 ready". Le client affichera sur son écran le message : "Serveur RX302 ready : @IP : port".

Question 4 *Développer le client/serveur. Valider le fonctionnement localement (127.0.0.1 ou localhost) puis sur le réseau.*

Question 5 *Enrichir le client/serveur de la fonction suivante :*

le client transmet en données un message au clavier (se documenter sur la classe Scanner), le serveur affiche le message et le renvoie.