Université Paris 7
M2 IAP et M2 MIC
Protocole Internet

TP1

1 Prélimaires

1.1 TCP

On veut créer une application client/serveur. Cette application utilise des sockets TCP.

1. Le serveur attend les clients sur le port 1027 et envoie régulièrement à chaque client un message contenant le nombre de clients qui sont connectés. Le serveur doit donc être capable de gérer plusieurs clients simultanément.
2. Le client se connecte au serveur et affiche les nombres reçus. L'arrêt de la communication se fait à l'initiative du client.
   (a) Définir un protocole de communication entre le client et le serveur.
   (b) Donner le code du serveur et du client.

1.2 UDP

On réalise une communication client/serveur en UDP. Le client lit une chaine au clavier et l'envoie au serveur sur le port 9876, le serveur attend un message du client et l'affiche.

On propose les codes suivants pour le serveur et le client:

```
------------------------------
import java.io.*;
import java.net.*;

public class ServeurUDP {

    public static void main(String args[]) throws Exception {
        DatagramSocket serverSocket = new DatagramSocket(9876);
        byte[] receiveData = new byte[1024];
        while (true) {
```

```
            DatagramPacket receivePacket = new
               DatagramPacket(receiveData, receiveData.length);
            serverSocket.receive(receivePacket);
            String sentence = new String(receivePacket.getData());
            System.out.println(sentence);
        }
    }
}
----------------------------
import java.net.*;
import java.util.Scanner;

public class ClientUDP {
    public static void main(String args[]) throws Exception {
        Scanner inFromUser = new Scanner(System.in);
        DatagramSocket clientSocket = new DatagramSocket();
        InetAddress IPAddress = InetAddress.getByName("localhost");
        byte[] sendData = new byte[1024];
        String sentence = inFromUser.nextLine();
        sendData = sentence.getBytes();
        DatagramPacket sendPacket = new
           DatagramPacket(sendData, sendData.length, IPAddress, 9876);
        clientSocket.send(sendPacket);

}
    }
----------
```

Que se passe-t-il quand le serveur est utilisé par plusieurs clients? (en particulier quand un client envoie un message long et un autre un message court)

Modifier le code du serveur pour qu'il affiche aussi l'adresse et le port de la source du message. Modifier le code du serveur pour qu'il renvoie un message au client.

2 Petites annonces

On veut réaliser une application de petites annonces (plomberie, guitare, déménagement, objet a vendre...)

Un utilisateur s'adresse à un serveur (IP connu, port connu) et lui envoie des petites annonces.

Le serveur envoie au client toutes les petites annonces reçus. Le serveur, autant que possible, n'envoie que des annonces de clients connectés et n'envoie pas les annonces obsoletes. Le client choisit une/des petite(s) annonce(s) parmi celle proposées et entre en communication directe avec l'utilisateur proposant cette annonce.

1. Quelle architecture proposez vous pour cette application ?
2. Précisez le type de communication (TCP,UDP), les ports utilisées, le contenu des messages (et éventuellement les réponses attendues).
3. Une fois vos choix présentés et validés lors de la séance de TP, rédigez un document contenant votre specification.

3 Travail à rendre

Déposez sur Didel le 15 ou le 16 novembre avant le debut de votre TP:

1. Les codes java correspondant aux sections 1 et 2 en indiquant comment vos programmes doivent être utilisés.
2. L'architecture et la spécification du protocole de communication (un par groupe de 6 personnes en indiquant les noms des membres du groupe) pour les petites annonces.
3. Le code java correspondant à une première version des petites annonces (un par binôme en indiquant les noms des membres du binome) où on ne considère que les communications clients/serveur