***Année 2009/2010 RT3GROUPE2***

****

***Travaux Pratiques de réseaux :***

**Simulation du routage statique basée sur le modèle Client/serveur**

**ELLOUMI MOHAMED KARIM// JRIDI BAYREM**

1. ***Objectif Du TP :***

On se propose de créer une application en langage java dont le but est de modéliser un transfert de paquet entre plusieurs nœuds d’un réseau local.

***Graphe du réseau***

***II. Code Du Programme :***

Initialement nous avons mis chaque code d’un nœud dans un projet différent, mais cette approche ne nous a pas permis de faire fonctionner le programme (les paquets reçues ne sont pas afficher ni reconnue à leurs arrivées par les nœuds récepteurs), c’est pour cette raison que nous avons réuni tous les nœuds dans un seul projet.

***NB :*** Ce code est celui de l’application serveur A, les autres serveurs ont pratiquement le même code avec quelque modifications.

***Le code :***

package mains;

import java.net.\*;

import java.io.\*;

import java.util.Scanner;

import java.util.ArrayList;

import javax.swing.JOptionPane;

public class a {

public static void main(String[] argv) throws Exception {

ArrayList<Routage> tableroutage = new ArrayList();

String ps = new String();

int cd = 10, i, k;

String pq;

paquet p = new paquet();

Routage r1 = new Routage();

r1.SetDest("B");

r1.SetWH("B");

r1.SetCont(1);

tableroutage.add(r1);

Routage r2 = new Routage();

r2.SetDest("D");

r2.SetWH("B");

r2.SetCont(2);

tableroutage.add(r2);

Routage r3 = new Routage();

r3.SetDest("E");

r3.SetWH("B");

r3.SetCont(3);

tableroutage.add(r3);

Routage r4 = new Routage();

r4.SetDest("C");

r4.SetWH("B");

r4.SetCont(4);

tableroutage.add(r4);

Routage r5 = new Routage();

r5.SetDest("C");

r5.SetWH("C");

r5.SetCont(1);

tableroutage.add(r5);

Routage r6 = new Routage();

r6.SetDest("C");

r6.SetWH("C");

r6.SetCont(2);

tableroutage.add(r6);

Routage r7 = new Routage();

r7.SetDest("D");

r7.SetWH("C");

r7.SetCont(3);

tableroutage.add(r7);

Routage r8 = new Routage();

r8.SetDest("B");

r8.SetWH("C");

r8.SetCont(4);

tableroutage.add(r8);

Scanner sc = new Scanner(System.in);

Scanner dt = new Scanner(System.in);

Scanner des = new Scanner(System.in);

p.SetSrc("A");

String[] b = {"Envoyer un mesg", "mettre le srveur en cout", "Annuler"};

JOptionPane jop = new JOptionPane(), jop2 = new JOptionPane();

String choi = (String) jop.showInputDialog(null,

"Veuillez choisir votre action !",

"Serveur A !",

JOptionPane.QUESTION\_MESSAGE,

null,

b,

b[2]);

jop2.showMessageDialog(null, "Votre choix est " + choi, "Serveur A !", JOptionPane.INFORMATION\_MESSAGE);

if (choi.equals(b[0])) {

JOptionPane jop5 = new JOptionPane(), jop6 = new JOptionPane();

String msg = jop5.showInputDialog(null, "Veuillez entrer votre msg !", "Serveur A !", JOptionPane.QUESTION\_MESSAGE);

jop6.showMessageDialog(null, "Votre msg est " + msg, "Serveur A !", JOptionPane.INFORMATION\_MESSAGE);

p.SetData(msg);

String[] d = {"B", "C", "D", "E"};

JOptionPane jop3 = new JOptionPane(), jop4 = new JOptionPane();

String dest = (String) jop3.showInputDialog(null,

"Veuillez choisir la destination de votre msg !",

"Serveur A !",

JOptionPane.QUESTION\_MESSAGE,

null,

d,

d[2]);

jop4.showMessageDialog(null, "Votre destination est " + dest, "Serveur A !", JOptionPane.INFORMATION\_MESSAGE);

p.SetDest(dest);

pq = p.GetSrc() + p.GetDest() + p.GetData();

System.out.println("le paquet devient: " + pq);

//choisir le prochain saut:

for (i = 0; i < tableroutage.size(); i++) {

k = tableroutage.get(i).GetCont();

if (p.GetDest().equals(tableroutage.get(i).GetDest()) && k < cd) {

cd = k;

ps = tableroutage.get(i).GetWH();

}

}

System.out.println("prochain saut suivant la table de routage est" + ps);

int port = 0;

if (ps.equals("B")) {

port = 14406;

}

if (ps.equals("C")) {

port = 14407;

}

if (ps.equals("D")) {

port = 14408;

}

if (ps.equals("E")) {

port = 14409;

}

//clse la socket avec son client a lui:

//socketBienvenue.close();

//crée une socket avec le port approprié

System.out.println("le port suivant est " + port);

Socket socketenvoie = new Socket("127.0.0.1", port);

//créer le flux de sortie attaché au socket

OutputStream Sortie = socketenvoie.getOutputStream();

ObjectOutputStream sort = new ObjectOutputStream(Sortie);

sort.flush();

sort.writeObject(p);

}

// Créer un socket d'écoute sur le port 14405

ServerSocket socketBienvenue = new ServerSocket(14405);

System.out.println("> SERVER :Port n°" + socketBienvenue.getLocalPort() + "du serveur A est à l'écoute...");

while (true) {

//Attendre la connexion d'un client

Socket socketConnexion = socketBienvenue.accept();

System.out.println("> SERVER :Connction etablie");

//Creer le flux d'entree attachأ© au socket

ObjectInputStream Entree = new ObjectInputStream(socketConnexion.getInputStream());

System.out.println("> SERVER :Lecture de la donnee recue en cours");

//Lire la chaine de caracteres depuis le socket

try {

p = (paquet) Entree.readObject();

System.out.println("> SERVER :la chaine recue est:" + p.GetSrc() + p.GetDest() + p.GetData());

System.out.println("> SERVER :la destination est:" + p.GetDest());

} catch (ClassNotFoundException cnfex) {

System.out.println("> SERVER : Le type de l'objet recu est inconnu!");

}

if (p.GetDest().compareTo("A") != 0) {

//choisir le prochain saut:

for (i = 0; i < tableroutage.size(); i++) {

k = tableroutage.get(i).GetCont();

if (p.GetDest().equals(tableroutage.get(i).GetDest()) && k < cd) {

cd = k;

ps = tableroutage.get(i).GetWH();

}

}

System.out.println("prochain saut suivant la table de routage est" + ps);

int port = 0;

if (ps.equals("B")) {

port = 14406;

}

if (ps.equals("C")) {

port = 14407;

}

if (ps.equals("D")) {

port = 14408;

}

if (ps.equals("E")) {

port = 14409;

}

//clse la socket avec son client a lui:

//socketBienvenue.close();

//cree une new socket avec le port approprié

System.out.println("le port suivant est " + port);

Socket socketenvoie = new Socket("127.0.0.1", port);

//creer le flux de sortie attaché au socket

OutputStream Sortie = socketenvoie.getOutputStream();

ObjectOutputStream sort = new ObjectOutputStream(Sortie);

sort.writeObject(p);

} else {

System.out.println("la chaine recue est:" + p.GetSrc() + p.GetDest() +p.GetData());

System.out.println("la donnee est:" + p.GetData());

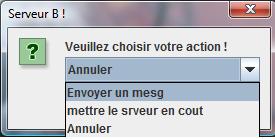
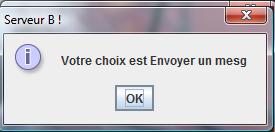
}

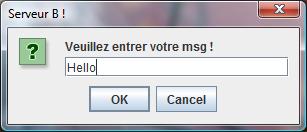
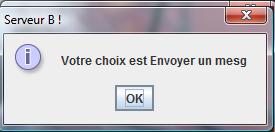
} //Reapartir et attendre une nouvelle connexion

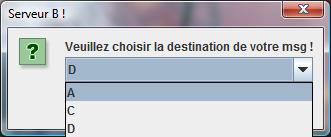
}

}

***Voici quelques images de l’interface graphique de l’application :***





***III.CONCLUSION :***

Ce TP fut une excellente occasion de programmer en java et de comprendre les subtilités de ce langage, et d’approfondir nos connaissances dans ce domaine.