package lte;

import java.net.DatagramPacket;

import java.net.InetAddress;

import java.net.MulticastSocket;

import java.util.HashMap;

import java.util.StringTokenizer;

import java.util.Vector;

public class Intermediatemultireceiver extends Thread

{

public String innode, insysno, inportno,traffic;

public HashMap allinsysno = new HashMap();

public HashMap allinport = new HashMap();

public HashMap allnodesys = new HashMap();

public HashMap allnodeport = new HashMap();

public HashMap allintraffic = new HashMap();

public Vector allinnodes = new Vector();

public Intermediatemultireceiver(String node,String sysno,String portno,String traffic)

{

this.innode = node;

this.insysno = sysno;

this.inportno = portno;

this.traffic = traffic;

start();

}

public void run()

{

while (true)

{

try

{

InetAddress in = InetAddress.getByName("225.89.67.45");

MulticastSocket ms = new MulticastSocket(4567);

ms.joinGroup(in);

byte[] b = new byte[1024];

DatagramPacket dp = new DatagramPacket(b, b.length);

ms.receive(dp);

String data1 = new String(dp.getData()).trim();

StringTokenizer str = new StringTokenizer(data1, "$");

String status = str.nextToken();

if(status.equalsIgnoreCase("INTERMEDIATE"))

{

String node = str.nextToken();

String sys = str.nextToken();

String port = str.nextToken();

String traffic = str.nextToken();

allinport.put(node,port);

allinsysno.put(node,sys);

allintraffic.put(node, traffic);

if(!allinnodes.contains(node))

{

allinnodes.add(node);

}

}

else if(status.equalsIgnoreCase("NODEDETAILS"))

{

String node = str.nextToken();

String sys = str.nextToken();

String port = str.nextToken();

allnodesys.put(node,sys);

allnodeport.put(node,port);

}

}

catch(Exception e)

{

e.printStackTrace();

}

}

}

}