PROJECT ON

IP MOBILITY SUPPORT

Instructor: Dr. Zygmunt Haas

CS6390- Advanced Computer Network -Fall 2013

By,

Saranath GovindaRaju (2021197841)

Nithin Venkatraman J R (2021189146)

Table of Contents

[ABSTRACT: 3](#_Toc374518289)

[MODULE 1: 4](#_Toc374518290)

[DATA FLOW WITHIN A NETWORK: 4](#_Toc374518291)

[ARCHITECTURE: 4](#_Toc374518292)

[MODULE 2: 5](#_Toc374518293)

[HANDOVER AND FOREIGN AGENT REGISTRATION: 5](#_Toc374518294)

[HANDOVER: 5](#_Toc374518295)

[REGISTRATION: 5](#_Toc374518296)

[MODULE 3: 6](#_Toc374518297)

[FORWARDING/TUNNELLING: 6](#_Toc374518298)

[TUNNELING: 6](#_Toc374518299)

[CODE: 7](#_Toc374518300)

[Server.java: 7](#_Toc374518301)

[Client1.java: 11](#_Toc374518302)

[Client2.java: 19](#_Toc374518303)

[SCREENSHOTS OF OUTPUT: 28](#_Toc374518304)

[HOME AGENT SERVER: 28](#_Toc374518305)

[DATA FLOW WITHIN A NETWORK: 28](#_Toc374518306)

[Sending Host (Client 1): 28](#_Toc374518307)

[Mobile Node (Client 2): 29](#_Toc374518308)

[FOREIGN AGENT (SERVER 2): 29](#_Toc374518309)

[REGISTRATION AND FORWARDING: 30](#_Toc374518310)

[Sending Host (Client 1) : 30](#_Toc374518311)

[Mobile Node (Client 2): 30](#_Toc374518312)

[CONCLUSION AND FUTURE WORKS: 31](#_Toc374518313)

# ABSTRACT:

The main aim of the project is to provide solutions for the problem of routing packets to the mobile host. This is achieved by performing two mechanisms, which is explained later. Initially each host in the network will be given an IP address which is supposed to be the permanent home address of the hosts. The hosts within the network transfer messages via the home agent of the network.

The problem arises when the mobile hosts switch from one network to another network. If a particular host would like to send the messages to the mobile host, then it would forward the message to the home agent in the network. Since the mobile host has switched the network, the home agent cannot forward the message to the specified destination.

The solution for this scenario is to implement forwarding and registration mechanisms. When the mobile host disconnects the home network and attaches to a foreign network then it hears the advertisement from the foreign network and registers with the foreign agent. The foreign agent provides the care of address to the home agent of the mobile host’s home network and thus the Registration takes place

The home agent then forwards the message, sent from the sending host, to the Care Of Address which was provided by the foreign agent and the foreign agent then finally forwards the message to the mobile host and thus the message is transferred from the sending host in one network to the mobile host in the other network.

# MODULE 1:

## DATA FLOW WITHIN A NETWORK:

Initially the mobile host is said to have a permanent IP Address in its home network and this is used by the Sending host to forward the message to the mobile host. The Message is sent from the sending host to the Home Agent. The Home Agent contains the IP Address of all the hosts in the network and hence forwards the message to the Mobile host. Thus the data is sent successfully from the sending host to the mobile host within a network. (Refer fig 1.1)

## ARCHITECTURE:



Server

**Fig 1.1 BASIC CLIENT-SERVER CONNECTION**

Connection

Message

# MODULE 2:

## HANDOVER AND FOREIGN AGENT REGISTRATION:

The Home Agent sends an advertising message to the Mobile Host and thus the mobile host is provided with the IP Address of the Home Agent. The mobile host then disconnects from the home network and establishes a connection with a new network. Thus the **Handover** occurs as the Home agent (HA)-Mobile IP connection gets disconnected and Mobile IP establishes a link with the foreign agent in the new network (see fig 2.1). The Mobile host sends IP address of FA to HA in order to perform forwarding or tunnelling. Hence **Registration** is performed (see fig 2.2).

## HANDOVER:

HA

A

Host

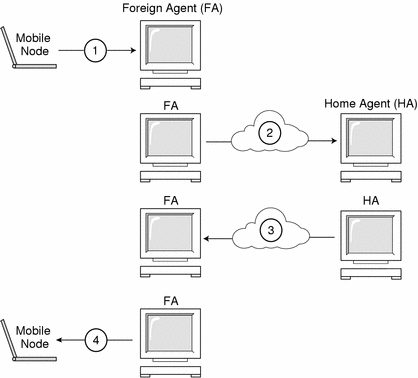
FA

MH

MH

**Fig 2.1 Handover**

## REGISTRATION:

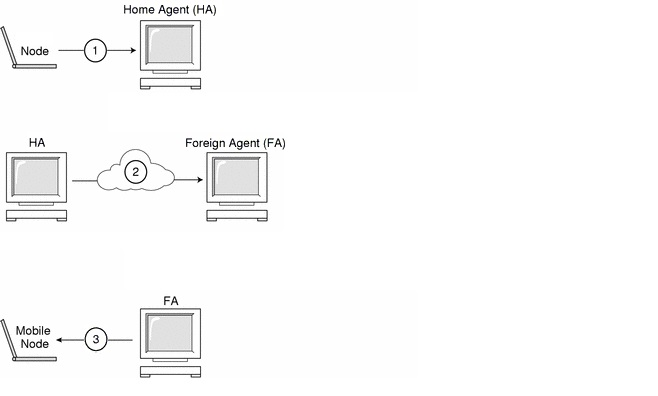
 **Fig 2.2 Registration**

# MODULE 3:

## FORWARDING/TUNNELLING:

The problem that arises when the mobile host has moved from the home network to the foreign network is that how will the home Agent forward the packet to the Foreign Agent. The **Tunneling technique** provides the solution for this scenario. The Home Agent that receives the packet from the sending host now wraps that data packet in the IP header and transmits to the foreign agent. Thus an IP tunnel is established between the Home Agent and Foreign Agent. The Data packet forwarded to the foreign agent is then sent to the Mobile host with the help of its new IP address in the foreign network. Thus the data packet from the sending host in a network is sent to the mobile host in another network with the help of forwarding technique.

## TUNNELING:



# CODE:

## Server.java:

import java.net.ServerSocket;

import java.net.Socket;

import java.io.IOException;

public class MultiThreadServer implements Runnable{

protected int serverPort = 8080;

protected ServerSocket serverSocket = null;

protected boolean isStopped = false;

protected Thread runningThread= null;

public MultiThreadServer(int port){

this.serverPort = port;

}

public void run(){

synchronized(this){

this.runningThread = Thread.currentThread();

}

openServerSocket();

while(! isStopped()){

Socket clientSocket = null;

try {

clientSocket = this.serverSocket.accept();

} catch (IOException e) {

if(isStopped()) {

System.out.println("Server Stopped.") ;

return;

}

throw new RuntimeException(

"Error accepting client connection", e);

}

new Thread(

new WorkerRunnable(

clientSocket, "Multithreaded Server",this.serverPort)

).start();

}

System.out.println("Server Stopped.") ;

}

private synchronized boolean isStopped() {

return this.isStopped;

}

public synchronized void stop(){

this.isStopped = true;

try {

this.serverSocket.close();

} catch (IOException e) {

throw new RuntimeException("Error closing server", e);

}

}

private void openServerSocket() {

try {

this.serverSocket = new ServerSocket(this.serverPort);

} catch (IOException e) {

throw new RuntimeException("Cannot open port 8080", e);

}

}

}

import java.io.BufferedReader;

import java.io.InputStream;

import java.io.OutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.InetAddress;

import java.net.Socket;

/\*\*

\*/

public class WorkerRunnable implements Runnable {

protected Socket clientSocket = null;

Socket s, s1;

PrintWriter out\_fa;

static Socket[] client\_s = new Socket[100];

static Socket[] client\_fa\_pair = new Socket[100];

protected String serverText = null;

InetAddress hip, fip, fagent;

String fip\_string, f\_host;

static int[] client\_id = new int[100];

int port, f\_port;

static int client\_counter = 0;

static int fa\_counter = 0;

static String[] client\_ip = new String[100];

static String[] client\_fa\_ip = new String[100];

private int fport;

public WorkerRunnable(Socket clientSocket, String serverText, int port) {

this.clientSocket = clientSocket;

this.serverText = serverText;

this.port = port;

}

public void run() {

try {

BufferedReader input = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);

String ans = new String();

//long time = System.currentTimeMillis();

// String ans = input.readLine();

while (true) {

//if(ans.)

if (!((ans = input.readLine()).isEmpty())) {

if (ans.equalsIgnoreCase("connection request")) {

// String ans = input.readLine();

// client\_id[client\_counter]=Integer.parseInt(ans);

client\_s[client\_counter] = clientSocket;

client\_ip[client\_counter] = clientSocket.getInetAddress().toString().substring(1);

System.out.println("Client " + client\_id[client\_counter] + " ip: " + client\_ip[client\_counter]);

client\_counter++;

out.println("Connected successfully......");

}

if (ans.equalsIgnoreCase("forward")) {

String ip, msg;

ip = input.readLine();

msg = input.readLine();

boolean flag = false;

System.out.println("Client " + clientSocket.getInetAddress().toString().substring(1) + "sending msg to " + ip);

int i, j;

for (i = 0; i < 100; i++) {

if (ip.equalsIgnoreCase(client\_ip[i])) {

for (j = 0; j < 100; j++) {

if (client\_ip[i].equalsIgnoreCase(client\_fa\_ip[j])) {

flag = true;

break;

}

}

if (flag) {

j++;

System.out.println("match found " + ip);

// System.out.println(client\_fa\_ip[j]);

fagent = InetAddress.getByName(client\_fa\_ip[j]);

//PrintWriter out1 =new PrintWriter(client\_s[13].getOutputStream(), true);

s1 = new Socket(fagent, 9006);

PrintWriter out1 = new PrintWriter(s1.getOutputStream(), true);

BufferedReader input1 = new BufferedReader(new InputStreamReader(s1.getInputStream()));

//System.out.println("123"+out1);

out1.println("forward");

out1.println(ip);

out1.println(msg);

} else {

System.out.println("match found " + ip);

PrintWriter out1 = new PrintWriter(client\_s[i].getOutputStream(), true);

out1.println(clientSocket.getInetAddress().toString().substring(1));

out1.println(msg);

}

}

}

}

if (ans.equalsIgnoreCase("FA\_reg")) {

String fa\_ip, c\_ip;

c\_ip = input.readLine();

fa\_ip = input.readLine();

f\_host = input.readLine();

f\_port = Integer.parseInt(input.readLine());

int i;

// System.out.println("1");

for (i = 0; i < 100; i++) {

//System.out.println("2");

if (c\_ip.equalsIgnoreCase(client\_ip[i])) {

//System.out.println("3");

//System.out.println("match found "+ip);

System.out.println("Client " + c\_ip + " moved to other network having Foreign Agent " + fa\_ip);

client\_fa\_ip[fa\_counter] = c\_ip;

//client\_fa\_pair[fa\_counter];

fa\_counter++;

client\_fa\_ip[fa\_counter] = f\_host;

client\_fa\_pair[0] = clientSocket;

client\_s[13] = clientSocket;

// out\_fa=out;

out.println("FA\_reg Successful");

}

}

}

if (ans.equalsIgnoreCase("handover")) {

String answer = input.readLine();

hip = InetAddress.getByName(answer);

fport = Integer.parseInt(input.readLine());

fip = InetAddress.getByName("NITHIN-TOSH");

fip\_string = InetAddress.getByName("NITHIN-TOSH").toString();

client\_s[client\_counter] = clientSocket;

client\_ip[client\_counter] = clientSocket.getInetAddress().toString().substring(1);

System.out.println("Hand over : Client " + client\_id[client\_counter] + " ip: " + client\_ip[client\_counter]);

client\_counter++;

s = new Socket(hip, fport);

PrintWriter out1 = new PrintWriter(s.getOutputStream(), true);

BufferedReader input1 = new BufferedReader(new InputStreamReader(s.getInputStream()));

//to HA....

out1.println("FA\_reg");

out1.println("c" + client\_ip[client\_counter]);

out1.println("f" + fip\_string);

answer = input1.readLine();

System.out.println(answer);

}

}

}

} catch (IOException e) {

//report exception somewhere.

e.printStackTrace();

}

}

}

## Client1.java:

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*

\* Client1.java

\*

\* Created on Oct 31, 2013, 9:12:25 PM

\*/

package A;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.Inet4Address;

import java.net.InetAddress;

import java.net.Socket;

import java.net.UnknownHostException;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author Saranath Raju

\*/

public class Client1 extends javax.swing.JFrame {

/\*\* Creates new form Client1 \*/

String homeip,ip, coa;

int home\_port,coa\_port;

InetAddress ip1,f\_hostname;

Socket s,s1;

PrintWriter out ;

BufferedReader input;

public Client1() throws UnknownHostException, IOException {

initComponents();

try {

ip=InetAddress.getLocalHost().getHostAddress().toString();

} catch (UnknownHostException ex) {

ip="192.168.56.1";

}

// homeip=jTextField5.getText();

ip1=InetAddress.getByName("SARANATHRAJU") ;

homeip = InetAddress.getByName("SARANATHRAJU").toString();

// jLabel1.setText(jLabel1.getText()+" "+homeip);

// connection();

}

/\*\* This method is called from within the constructor to

\* initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is

\* always regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jLabel5 = new javax.swing.JLabel();

jTextField4 = new javax.swing.JTextField();

jLabel6 = new javax.swing.JLabel();

jScrollPane2 = new javax.swing.JScrollPane();

jTextArea2 = new javax.swing.JTextArea();

jButton3 = new javax.swing.JButton();

jPanel2 = new javax.swing.JPanel();

jScrollPane3 = new javax.swing.JScrollPane();

jTextArea3 = new javax.swing.JTextArea();

jLabel7 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jLabel4 = new javax.swing.JLabel();

jTextField3 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

jLabel8 = new javax.swing.JLabel();

jTextArea1.setColumns(20);

jTextArea1.setRows(5);

jScrollPane1.setViewportView(jTextArea1);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Mobile\_Client1");

jLabel1.setText("Home Agent IP : ");

jLabel2.setText("Port #");

jTextField1.setText("1234");

jTextField1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField1ActionPerformed(evt);

}

});

jButton1.setText("Run");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jLabel5.setText("To IP Address :");

jTextField4.setText("127.0.0.1");

jLabel6.setText("Message :");

jTextArea2.setColumns(20);

jTextArea2.setRows(5);

jScrollPane2.setViewportView(jTextArea2);

jButton3.setText("Send");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(10, 10, 10)

.addComponent(jButton3))

.addComponent(jLabel1)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jButton1)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addComponent(jLabel6)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(64, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jButton1)

.addGap(23, 23, 23)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel5)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jButton3)

.addContainerGap(90, Short.MAX\_VALUE))

);

jTextArea3.setColumns(20);

jTextArea3.setEditable(false);

jTextArea3.setRows(5);

jScrollPane3.setViewportView(jTextArea3);

jLabel7.setText("Activity Log :");

jButton2.setText("HandOver");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jLabel4.setText("Port #");

jTextField3.setText("port #");

jTextField2.setText("Host Name");

jTextField2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField2ActionPerformed(evt);

}

});

jLabel3.setText("Foreign Agent IP :");

jLabel8.setText("Hostname :");

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 291, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(79, 79, 79)

.addComponent(jButton2))

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 64, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel3)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addComponent(jLabel7)

.addGap(18, 18, 18)))

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jLabel8)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(32, Short.MAX\_VALUE))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel8)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(5, 5, 5)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 44, Short.MAX\_VALUE)

.addComponent(jButton2)

.addGap(22, 22, 22)

.addComponent(jLabel7)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 194, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(20, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addContainerGap())

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jTextField1ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jTextField1ActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed

// TODO add your handling code here:

home\_port=Integer.parseInt(jTextField1.getText());

jLabel1.setText((jLabel1.getText()+" "+homeip));

try {

connection();

} catch (UnknownHostException ex) {

Logger.getLogger(Client1.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(Client1.class.getName()).log(Level.SEVERE, null, ex);

}

}//GEN-LAST:event\_jButton1ActionPerformed

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton3ActionPerformed

// TODO add your handling code here:

String ip= jTextField4.getText();

String msg = jTextArea2.getText();

out.println(ip);

out.println(msg);

}//GEN-LAST:event\_jButton3ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton2ActionPerformed

try {

// TODO add your handling code here:

s.close();

int f\_port;

String f\_ip;

//f\_ip= jTextField2.getText();

f\_port=Integer.parseInt(jTextField3.getText());

f\_hostname=InetAddress.getByName(jTextField2.getText());

f\_ip=(f\_hostname.getAddress()).toString();

s1=new Socket(f\_hostname, f\_port);

out =new PrintWriter(s1.getOutputStream(), true);

input =new BufferedReader(new InputStreamReader(s1.getInputStream()));

out.println(ip);

String ans=input.readLine();

} catch (Exception ex) {

Logger.getLogger(Client1.class.getName()).log(Level.SEVERE, null, ex);

}

}//GEN-LAST:event\_jButton2ActionPerformed

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jTextField2ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jTextField2ActionPerformed

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

try {

new Client1().setVisible(true);

} catch (Exception ex) {

Logger.getLogger(Client1.class.getName()).log(Level.SEVERE, null, ex);

}

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JTextArea jTextArea1;

private javax.swing.JTextArea jTextArea2;

private javax.swing.JTextArea jTextArea3;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

private javax.swing.JTextField jTextField4;

// End of variables declaration//GEN-END:variables

public void connection() throws UnknownHostException, IOException {

s = new Socket(ip1, home\_port);

out =new PrintWriter(s.getOutputStream(), true);

String temp = new String();

temp="1112"+ip;

out.println(ip);//sends to server

input =new BufferedReader(new InputStreamReader(s.getInputStream()));

String answer = input.readLine();//reads from server

if(!(answer=input.readLine()).isEmpty())

{

System.out.println("From server :" +answer);

answer=input.readLine();

System.out.println(answer);

}

}

## Client2.java:

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*

\* Client1.java

\*

\* Created on Oct 31, 2013, 9:12:25 PM

\*/

package A;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.Inet4Address;

import java.net.InetAddress;

import java.net.Socket;

import java.net.UnknownHostException;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author Saranath Raju

\*/

public class Client1 extends javax.swing.JFrame {

/\*\* Creates new form Client1 \*/

String homeip,ip, coa;

int home\_port,coa\_port;

InetAddress ip1,f\_hostname;

Socket s,s1;

PrintWriter out ;

BufferedReader input;

public Client1() throws UnknownHostException, IOException {

initComponents();

try {

ip=InetAddress.getLocalHost().getHostAddress().toString();

} catch (UnknownHostException ex) {

ip="192.168.56.1";

}

// homeip=jTextField5.getText();

ip1=InetAddress.getByName("SARANATHRAJU") ;

homeip = InetAddress.getByName("SARANATHRAJU").toString();

// jLabel1.setText(jLabel1.getText()+" "+homeip);

// connection();

}

/\*\* This method is called from within the constructor to

\* initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is

\* always regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jLabel5 = new javax.swing.JLabel();

jTextField4 = new javax.swing.JTextField();

jLabel6 = new javax.swing.JLabel();

jScrollPane2 = new javax.swing.JScrollPane();

jTextArea2 = new javax.swing.JTextArea();

jButton3 = new javax.swing.JButton();

jPanel2 = new javax.swing.JPanel();

jScrollPane3 = new javax.swing.JScrollPane();

jTextArea3 = new javax.swing.JTextArea();

jLabel7 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jLabel4 = new javax.swing.JLabel();

jTextField3 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

jLabel8 = new javax.swing.JLabel();

jTextArea1.setColumns(20);

jTextArea1.setRows(5);

jScrollPane1.setViewportView(jTextArea1);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Mobile\_Client1");

jLabel1.setText("Home Agent IP : ");

jLabel2.setText("Port #");

jTextField1.setText("1234");

jTextField1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField1ActionPerformed(evt);

}

});

jButton1.setText("Run");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jLabel5.setText("To IP Address :");

jTextField4.setText("127.0.0.1");

jLabel6.setText("Message :");

jTextArea2.setColumns(20);

jTextArea2.setRows(5);

jScrollPane2.setViewportView(jTextArea2);

jButton3.setText("Send");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(10, 10, 10)

.addComponent(jButton3))

.addComponent(jLabel1)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jButton1)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addComponent(jLabel6)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(64, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jButton1)

.addGap(23, 23, 23)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel5)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jButton3)

.addContainerGap(90, Short.MAX\_VALUE))

);

jTextArea3.setColumns(20);

jTextArea3.setEditable(false);

jTextArea3.setRows(5);

jScrollPane3.setViewportView(jTextArea3);

jLabel7.setText("Activity Log :");

jButton2.setText("HandOver");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jLabel4.setText("Port #");

jTextField3.setText("port #");

jTextField2.setText("Host Name");

jTextField2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField2ActionPerformed(evt);

}

});

jLabel3.setText("Foreign Agent IP :");

jLabel8.setText("Hostname :");

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 291, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(79, 79, 79)

.addComponent(jButton2))

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 64, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel3)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addComponent(jLabel7)

.addGap(18, 18, 18)))

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jLabel8)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(32, Short.MAX\_VALUE))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel8)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(5, 5, 5)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 44, Short.MAX\_VALUE)

.addComponent(jButton2)

.addGap(22, 22, 22)

.addComponent(jLabel7)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 194, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(20, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addContainerGap())

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jTextField1ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jTextField1ActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed

// TODO add your handling code here:

home\_port=Integer.parseInt(jTextField1.getText());

jLabel1.setText((jLabel1.getText()+" "+homeip));

try {

connection();

} catch (UnknownHostException ex) {

Logger.getLogger(Client1.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(Client1.class.getName()).log(Level.SEVERE, null, ex);

}

}//GEN-LAST:event\_jButton1ActionPerformed

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton3ActionPerformed

// TODO add your handling code here:

String ip= jTextField4.getText();

String msg = jTextArea2.getText();

out.println(ip);

out.println(msg);

}//GEN-LAST:event\_jButton3ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton2ActionPerformed

try {

// TODO add your handling code here:

s.close();

int f\_port;

String f\_ip;

//f\_ip= jTextField2.getText();

f\_port=Integer.parseInt(jTextField3.getText());

f\_hostname=InetAddress.getByName(jTextField2.getText());

f\_ip=(f\_hostname.getAddress()).toString();

s1=new Socket(f\_hostname, f\_port);

out =new PrintWriter(s1.getOutputStream(), true);

input =new BufferedReader(new InputStreamReader(s1.getInputStream()));

out.println(ip);

String ans=input.readLine();

} catch (Exception ex) {

Logger.getLogger(Client1.class.getName()).log(Level.SEVERE, null, ex);

}

}//GEN-LAST:event\_jButton2ActionPerformed

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jTextField2ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jTextField2ActionPerformed

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

try {

new Client1().setVisible(true);

} catch (Exception ex) {

Logger.getLogger(Client1.class.getName()).log(Level.SEVERE, null, ex);

}

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JTextArea jTextArea1;

private javax.swing.JTextArea jTextArea2;

private javax.swing.JTextArea jTextArea3;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

private javax.swing.JTextField jTextField4;

// End of variables declaration//GEN-END:variables

public void connection() throws UnknownHostException, IOException {

s = new Socket(ip1, home\_port);

out =new PrintWriter(s.getOutputStream(), true);

String temp = new String();

temp="1112"+ip;

out.println(ip);//sends to server

input =new BufferedReader(new InputStreamReader(s.getInputStream()));

String answer = input.readLine();//reads from server

if(!(answer=input.readLine()).isEmpty())

{

System.out.println("From server :" +answer);

answer=input.readLine();

System.out.println(answer);

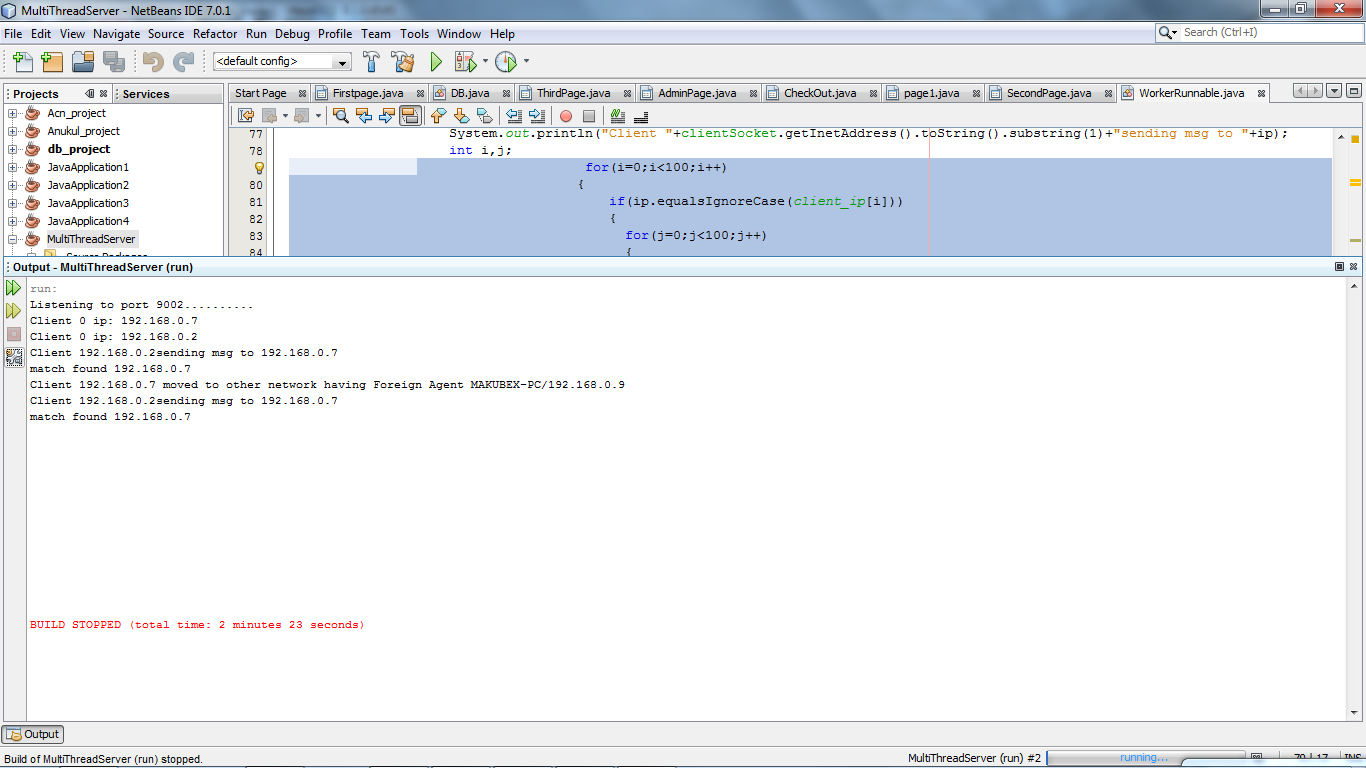
}

}

}

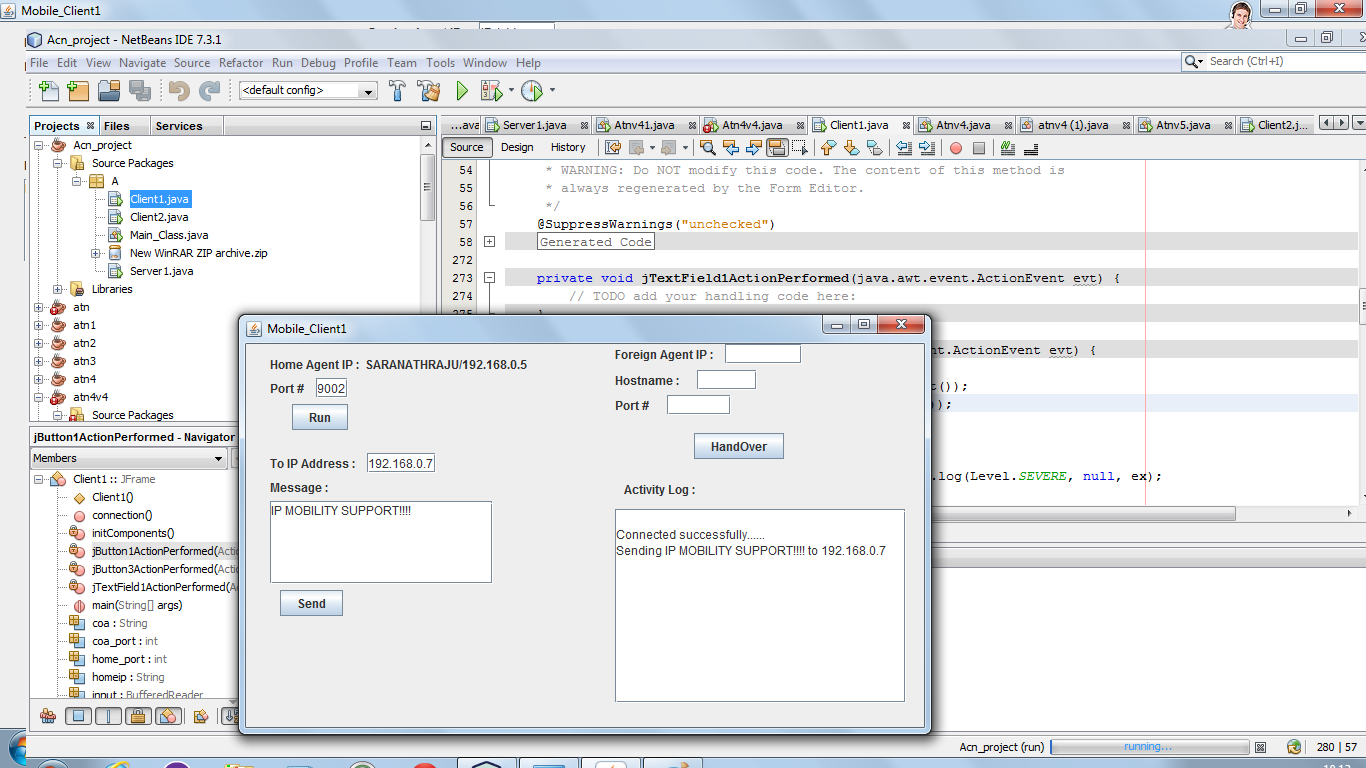
# SCREENSHOTS OF OUTPUT:

## HOME AGENT SERVER:

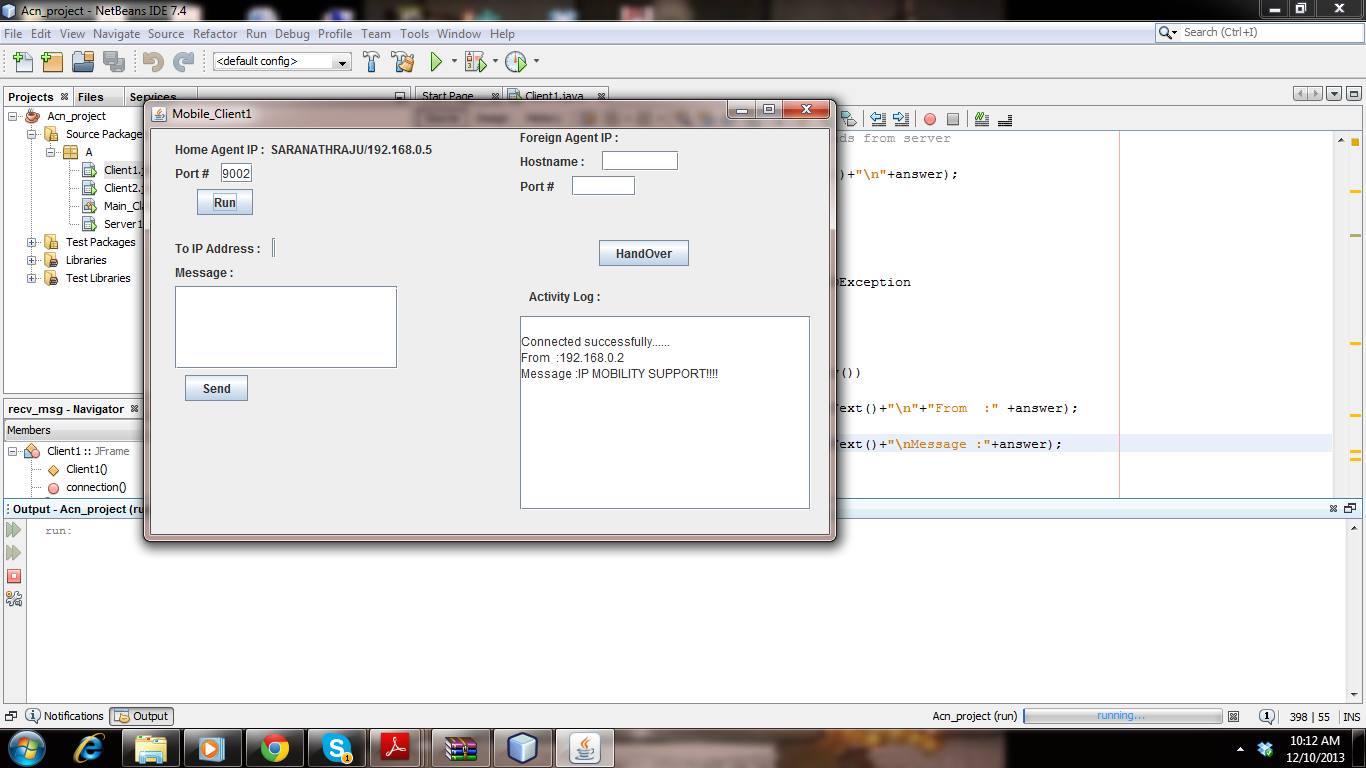
****

## DATA FLOW WITHIN A NETWORK:

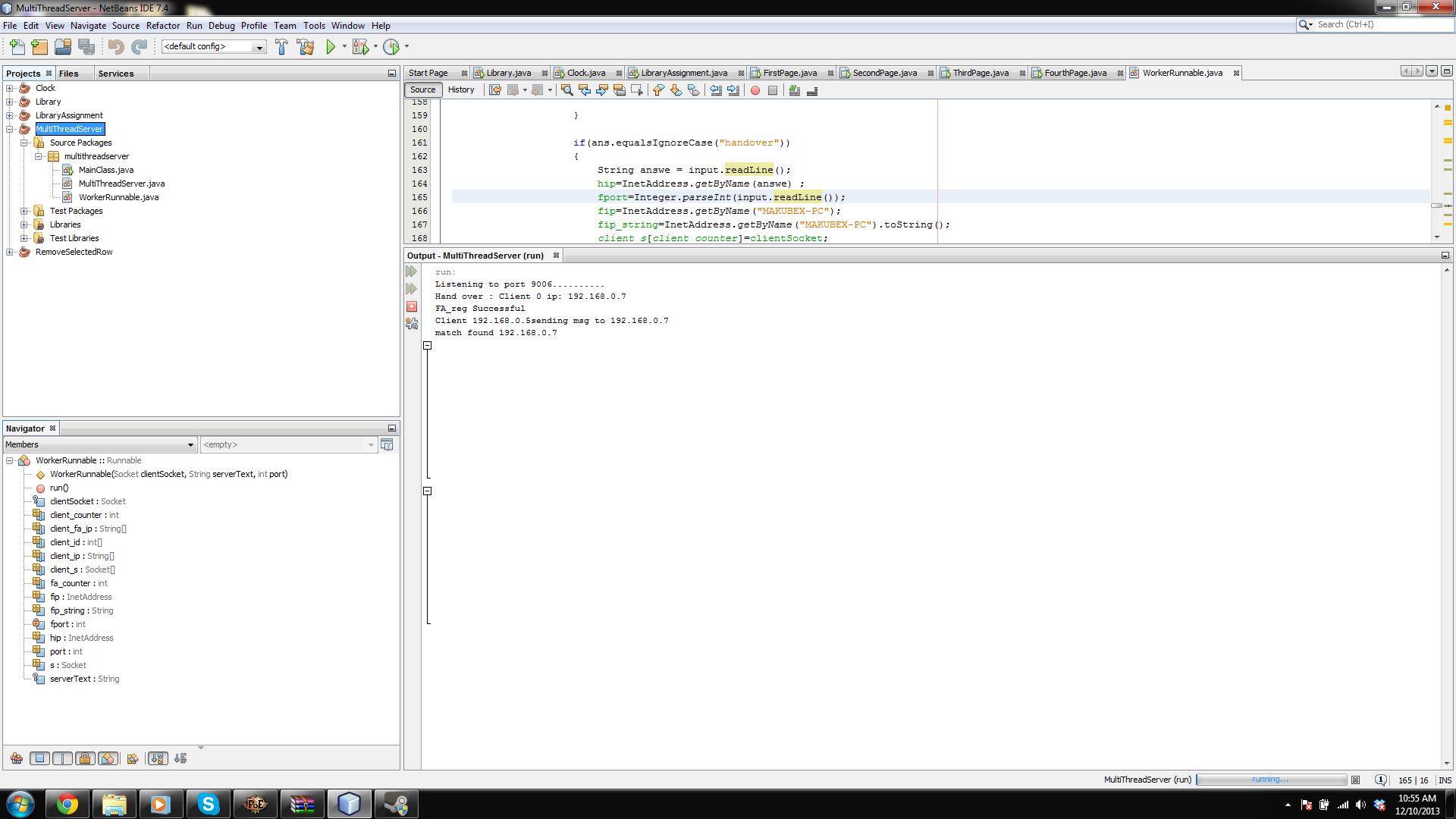
### Sending Host (Client 1):



### Mobile Node (Client 2):

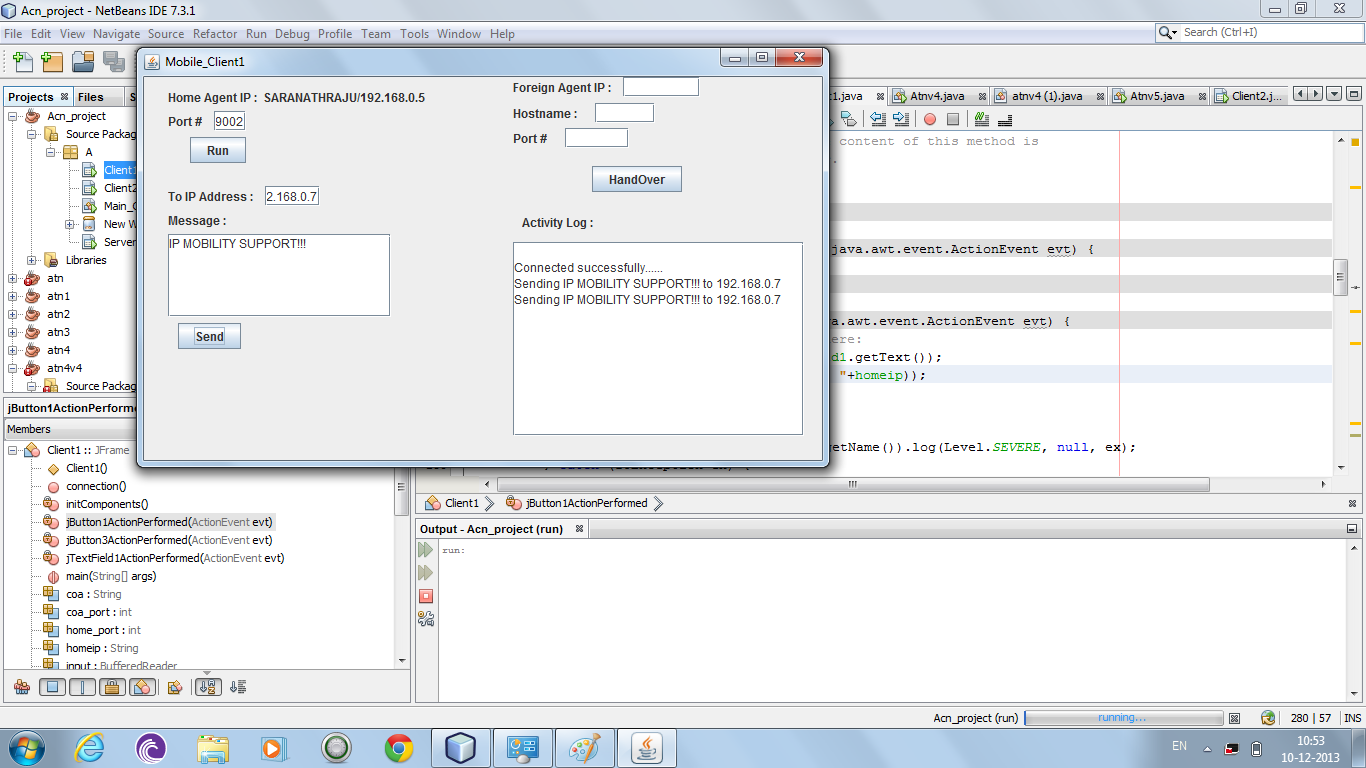


## FOREIGN AGENT (SERVER 2):

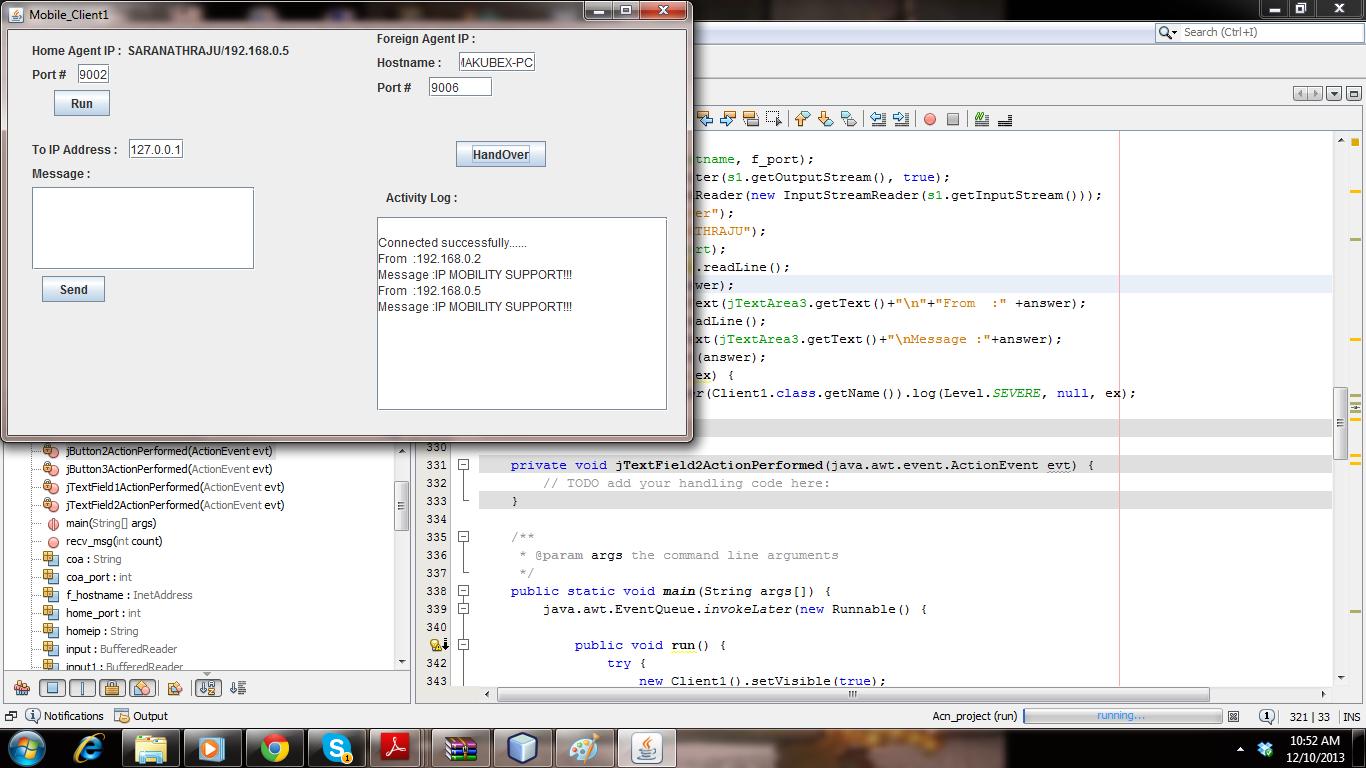


## REGISTRATION AND FORWARDING:

### Sending Host (Client 1) :



### Mobile Node (Client 2):



# CONCLUSION AND FUTURE WORKS:

Thus the data packet from the Sending host is sent through various internetworks and reaches the Mobile host in a new foreign network. The data traversing across various networks to the destination is achieved through Registration and Tunneling techniques. The Main reason for going for Mobile IP instead of static IP is to provide IP Routing and Identity for the node when it moves from one network to another network. However there are certain issues that remain in the IP Mobility. In this approach the route that is taken from the packet might not be an optimum route as the packets need to traverse all the way to the home agent of the mobile IP’s home network and then forwarded to the destination rather than directly forwarded from the sending host to destination. Hence our future work concentrates on the fact so as to make the route of the data packet from the source to destination as much efficient as possible. Hence it can be designed in such a way that it satisfies the trade-off between the successful delivery of packets and optimum route being chosen between the source and destination.