# School of Engineering and Design

**Electronic and Computer Engineering**

# MSc Data Communication Systems (EE5506)

**Assignment 2.3**

**Evaluating UDP Performance in File Transfer**

**Vijay Maligireddy**

**0833536**

**Objectives**

1. To understand the use of sockets.
2. To do the socket programming using JAVA
3. To experience and analyse how UDP works when transferring data using different packet sizes.
4. To analyse and understand the main features involved when using Java programming and the Graphical User Interface (GUI).

**Table of Contents**

# 1.0 Introduction

# 2.0 Task 1- Graphical User Interface (GUI) Design

# 3.0 Task 2- Evaluating UDP Performance

# 4.0 Conclusions

# 5.0 References

**Introduction**

**UDP (User Datagram Protocol):**

UDP is connection less transport layer protocol which does the basic job of transport layer. This protocol receives the data from the application layer and sample data into the segments and adds the UDP header to it and drops into network layer. UDP do not generate any sequence numbers or does not establish any connection before transmission or asks any acknowledgement after transmission, that’s why it is connection less and unreliable protocol. Although it is unreliable many applications use UDP for

1. Faster transmission- no connection setup, no acknowledgement, no congestion control.
2. Small header- it uses only 8 bytes of header, less over load.
3. It won’t maintain any connection state in end systems so server can support many application processes
4. The UDP header looks like

|  |  |
| --- | --- |
| SOURCE PORT  2 bytes | DESTINATION PORT  2 bytes |
| LENGTH  2 bytes | CHECKSUM  2 bytes |
| APPLICATION DATA | |

In the UDP header only 4 fields are there which are source and destination port addresses, length of data field, check sum field for error correction. Even though it got the checksum field its not effective error correction.

UDP is used in SNMP(network management), RIP (routing protocol), DNS(name translation), internet telephony, streaming multimedia like application layer processes.

**UDP Socket:**

UDP is an uncomplicated, unreliable datagram protocol used to work with the SOCK\_DGRAM socket kind used in the internet protocol field. UDP sockets are connectionless and we can set a connection between two computers in order to send and receive data. Moreover, UDP sockets do not have the global uniqueness property, in other words, there are no connection-oriented and the UDP socket refers to just the local computer and only require a receive buffer.

UDP and TCP address formats are similar. In fact, UDP require a port identifier plus the normal internet address composition. UDP port cannot be connected to a TCP port because the UDP port domain is different or separate from the TCP port. The default send buffer size for UDP sockets is 65535 bytes and for the receiver part is 2147483647 bytes[2].

A datagram socket is the sending or receiving point for a packet or data delivery. All packets sent or received on a datagram socket are individually addressed and routed. When we send multiple packets from one computer to another may be these data can be routed in a different way, and may arrive in any order. UDP broadcasts sends are commonly enabled on a DatagramSocket. In other to obtain broadcast packets a DatagramSocket should be bound to the wildcard address or to a more specific address.

In addition to the UDP socket class, there is a UDP broadcast class and it can be set to send messages to an entire subnet rather than to an individual peer socket. Moreover, UDP sockets are used for creating real-time media streaming protocols and full duplex messaging services by using a pair of sockets together; one of them to send and the other one to receive data.

**Task 1- Graphical User Interface (GUI) Design**

To evaluate the performance of the udp we create a client and server classes in the java with GUI. In this mainly we use connection is UDP socket. Main features of this client is it will specify the server ip, port number, packet size, start time, end time, transfer time, file choosing and connection type. In the connection type we choose UDP only, we have choises like TCP and RMI but for this assisgnment we choose only UDP.

To write the code we use NETBEANS 6.5(beta)

The code is.

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package client;

/\*\*

\*

\* @author me

\*/

public class Main {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

new client().setVisible(true);

}

}

/\*

\* client.java

\*

\* Created on December 5, 2008, 6:06 PM

\*/

package client;

import java.io.\*;

import java.net.\*;

import java.io.File;

import java.io.InputStreamReader;

import java.util.Date;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

/\*\*

\*

\* @author vijay

\*/

public class client extends javax.swing.JFrame {

FileInputStream fis = null;

DatagramSocket soc;

Socket s;

InetAddress ip;

byte[] size;

String k;

String i1;

String p;

int i,po;

/\*\* Creates new form client \*/

public client() {

initComponents();

}

/\*\* 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() {

buttonGroup1 = new javax.swing.ButtonGroup();

jRadioButton1 = new javax.swing.JRadioButton();

jRadioButton2 = new javax.swing.JRadioButton();

jRadioButton3 = new javax.swing.JRadioButton();

jLabel1 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jTextField2 = new javax.swing.JTextField();

jLabel5 = new javax.swing.JLabel();

jTextField3 = new javax.swing.JTextField();

jLabel6 = new javax.swing.JLabel();

jTextField4 = new javax.swing.JTextField();

jLabel7 = new javax.swing.JLabel();

jLabel8 = new javax.swing.JLabel();

jTextField5 = new javax.swing.JTextField();

jTextField6 = new javax.swing.JTextField();

jButton2 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

jLabel9 = new javax.swing.JLabel();

jTextField7 = new javax.swing.JTextField();

jTextField8 = new javax.swing.JTextField();

jLabel10 = new javax.swing.JLabel();

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

setTitle("CLIENT");

setResizable(false);

jRadioButton1.setBackground(new java.awt.Color(0, 255, 255));

buttonGroup1.add(jRadioButton1);

jRadioButton1.setForeground(new java.awt.Color(102, 0, 0));

jRadioButton1.setText("UDP");

jRadioButton2.setBackground(new java.awt.Color(0, 255, 255));

buttonGroup1.add(jRadioButton2);

jRadioButton2.setForeground(new java.awt.Color(102, 0, 0));

jRadioButton2.setText("TCP");

jRadioButton3.setBackground(new java.awt.Color(0, 255, 255));

buttonGroup1.add(jRadioButton3);

jRadioButton3.setText("RMI");

jLabel1.setFont(new java.awt.Font("Arial", 1, 12));

jLabel1.setForeground(new java.awt.Color(0, 0, 204));

jLabel1.setText("CONNECTION");

jTextField1.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 1, 1, new java.awt.Color(102, 102, 0)));

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

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

jTextField1ActionPerformed(evt);

}

});

jButton1.setText("BROWSE ...");

jButton1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0, 204), 2, true));

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

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

jButton1ActionPerformed(evt);

}

});

jLabel2.setBackground(new java.awt.Color(204, 0, 0));

jLabel2.setFont(new java.awt.Font("Arial", 1, 12));

jLabel2.setForeground(new java.awt.Color(153, 0, 0));

jLabel2.setText("CHOOSE A FILE");

jLabel3.setFont(new java.awt.Font("Arial", 3, 11));

jLabel3.setForeground(new java.awt.Color(204, 0, 204));

jLabel3.setText("TIME");

jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 10));

jLabel4.setForeground(new java.awt.Color(153, 153, 0));

jLabel4.setText("Start time");

jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 10));

jLabel5.setForeground(new java.awt.Color(102, 102, 0));

jLabel5.setText("End time");

jTextField3.addActionListener(new java.awt.event.ActionListener() {

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

jTextField3ActionPerformed(evt);

}

});

jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 10));

jLabel6.setForeground(new java.awt.Color(102, 102, 0));

jLabel6.setText("Transfer time");

jLabel7.setFont(new java.awt.Font("Arial", 2, 12));

jLabel7.setForeground(new java.awt.Color(0, 102, 0));

jLabel7.setText("SERVER IP:");

jLabel8.setFont(new java.awt.Font("Arial", 2, 12));

jLabel8.setForeground(new java.awt.Color(0, 153, 0));

jLabel8.setText("PORT NO:");

jButton2.setBackground(new java.awt.Color(0, 0, 0));

jButton2.setFont(new java.awt.Font("Arial", 3, 12));

jButton2.setForeground(new java.awt.Color(0, 153, 153));

jButton2.setText("START TRANSFER");

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

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

jButton2ActionPerformed(evt);

}

});

jTextArea1.setBackground(new java.awt.Color(204, 204, 255));

jTextArea1.setColumns(70);

jTextArea1.setRows(60);

jTextArea1.setBorder(javax.swing.BorderFactory.createCompoundBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new java.awt.Color(0, 51, 0)), new javax.swing.border.LineBorder(new java.awt.Color(102, 0, 0), 2, true)));

jScrollPane1.setViewportView(jTextArea1);

jLabel9.setFont(new java.awt.Font("Arial", 3, 12));

jLabel9.setForeground(new java.awt.Color(0, 51, 51));

jLabel9.setText("UDP PACKETSIZE:");

jTextField7.addActionListener(new java.awt.event.ActionListener() {

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

jTextField7ActionPerformed(evt);

}

});

jLabel10.setForeground(new java.awt.Color(51, 51, 0));

jLabel10.setText("SPEED :");

org.jdesktop.layout.GroupLayout layout = new org.jdesktop.layout.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.addContainerGap()

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(jScrollPane1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 593, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(layout.createSequentialGroup()

.add(jLabel1)

.add(169, 169, 169)

.add(jLabel2))

.add(layout.createSequentialGroup()

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING, false)

.add(org.jdesktop.layout.GroupLayout.LEADING, jRadioButton1, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.add(org.jdesktop.layout.GroupLayout.LEADING, jRadioButton2, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.add(org.jdesktop.layout.GroupLayout.LEADING, jRadioButton3, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, 83, Short.MAX\_VALUE))

.add(layout.createSequentialGroup()

.add(jLabel7)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.UNRELATED)

.add(jTextField5, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 107, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE))

.add(org.jdesktop.layout.GroupLayout.TRAILING, jButton2))

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING, false)

.add(layout.createSequentialGroup()

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.add(18, 18, 18)

.add(jLabel8)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.UNRELATED)

.add(jTextField6, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 81, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE))

.add(layout.createSequentialGroup()

.add(28, 28, 28)

.add(jTextField1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 224, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)))

.add(28, 28, 28)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.add(5, 5, 5)

.add(jLabel10)

.add(18, 18, 18)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING, false)

.add(jTextField3)

.add(jTextField2, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, 143, Short.MAX\_VALUE)

.add(jTextField4)

.add(layout.createSequentialGroup()

.add(10, 10, 10)

.add(jLabel6))

.add(jTextField8)))

.add(layout.createSequentialGroup()

.add(79, 79, 79)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)

.add(jLabel3, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 50, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(jLabel4, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 87, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)))))

.add(layout.createSequentialGroup()

.add(120, 120, 120)

.add(jButton1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 85, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.add(jLabel5)

.add(83, 83, 83))))

.add(layout.createSequentialGroup()

.add(73, 73, 73)

.add(jLabel9)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.UNRELATED)

.add(jTextField7, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 135, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)))))

.addContainerGap(org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.addContainerGap()

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jLabel1)

.add(jLabel2)

.add(jLabel3, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 15, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE))

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.add(14, 14, 14)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jRadioButton1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 23, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(jTextField1, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, 22, Short.MAX\_VALUE))

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(jRadioButton2, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 26, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.UNRELATED)

.add(jRadioButton3, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 32, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(18, 18, 18)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jLabel7)

.add(jTextField5, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(jLabel8)

.add(jTextField6, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE))

.add(28, 28, 28)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jButton2)

.add(jLabel9)

.add(jTextField7, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE))

.add(19, 19, 19))

.add(org.jdesktop.layout.GroupLayout.TRAILING, layout.createSequentialGroup()

.add(jButton1)

.add(129, 129, 129))

.add(layout.createSequentialGroup()

.addPreferredGap(org.jdesktop.layout.LayoutStyle.UNRELATED)

.add(jLabel4)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(jTextField2, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(jLabel5)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(jTextField3, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(jLabel6)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(jTextField4, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jTextField8, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(jLabel10))))

.add(7, 7, 7)

.add(jScrollPane1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 106, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.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 jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jTextField3ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jTextField3ActionPerformed

private void jTextField7ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jTextField7ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jTextField7ActionPerformed

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

// TODO add your handling code here:

JFileChooser fc = new JFileChooser("c://");

int returnVal = fc.showOpenDialog(null);

String fileName=fc.getSelectedFile().getAbsolutePath();

jTextField1.setText(fileName);

jTextArea1.append("choosen file is:"+fileName);

}//GEN-LAST:event\_jButton1ActionPerformed

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

i1 = jTextField5.getText();

try {

ip = InetAddress.getByName(i1);

} catch (UnknownHostException ex) {

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

}

p = jTextField6.getText();

po = Integer.parseInt(p);

if (jRadioButton1.isSelected()==true)

{

try {

soc = new DatagramSocket();

jTextArea1.append("connceted by using udp conncetion"+"\n");

String input = jTextField1.getText();

k = jTextField7.getText();

i = Integer.parseInt(k);

size = new byte[i];

fis = new FileInputStream(input);

Date beg=new Date();

jTextField2.setText(""+beg.getTime());

jTextArea1.append("packet size is"+i+"\n");

jTextArea1.append("server port is"+po+"\n");

jTextArea1.append("starting time is"+beg.getTime()+"\n");

jTextArea1.append("sending file"+"\n");

jTextArea1.append("sending lenth to server is"+size.length+"\n");

while (fis.read(size) > 0) {

System.out.println(new String(size));

DatagramPacket sendPacket = new DatagramPacket(size, size.length, ip, po);

soc.send(sendPacket);

}

jTextArea1.append("file was sent"+"\n");

Date end=new Date();

jTextField3.setText(""+end.getTime());

jTextArea1.append("ending time is"+end.getTime()+"\n");

long transferTime=end.getTime()-beg.getTime();

jTextField4.setText(""+transferTime);

long fileLength=new File(input).length();

double speed=1000\*(double) fileLength/(double)transferTime;

jTextField8.setText(""+speed);

} catch (IOException ex) {

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

}

soc.close();

}//GEN-LAST:event\_jButton2ActionPerformed

if (jRadioButton2.isSelected()==true)

{

try {

s = new Socket(ip, po);

jTextArea1.append("connceted by using tcp conncetion");

String inp = jTextField1.getText();

fis = new FileInputStream(inp);

//code to send file to socket

} catch (IOException ex) {

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

}

}

}

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

private javax.swing.ButtonGroup buttonGroup1;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

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.JLabel jLabel9;

private javax.swing.JRadioButton jRadioButton1;

private javax.swing.JRadioButton jRadioButton2;

private javax.swing.JRadioButton jRadioButton3;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextArea jTextArea1;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

private javax.swing.JTextField jTextField4;

private javax.swing.JTextField jTextField5;

private javax.swing.JTextField jTextField6;

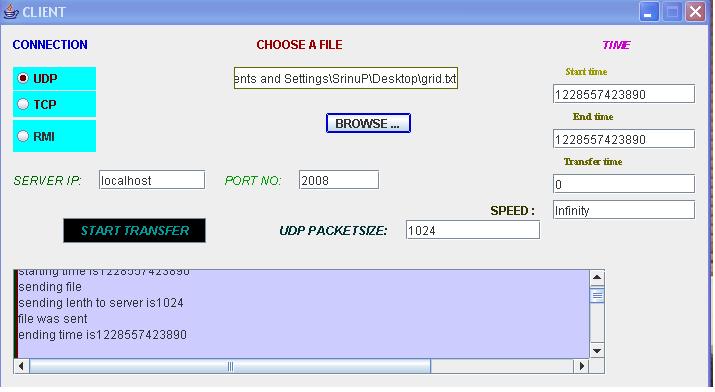
private javax.swing.JTextField jTextField7;

private javax.swing.JTextField jTextField8;

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

}

When we run this as a java application result as



**Design of server:**

In the design of server we use the options like.

* start listening
* stop listening
* socket ip
* port number
* connection type

the code for the server is

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package server;

/\*\*

\*

\* @author vijay

\*/

public class Main {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

new server().setVisible(true);

}

}

/\*

\* server.java

\*

\* Created on December 5, 2008, 6:46 PM

\*/

package server;

import java.io.IOException;

import java.net.\*;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author vijay

\*/

public class server extends javax.swing.JFrame {

String s;

DatagramSocket ser;

/\*\* Creates new form server \*/

public server() {

initComponents();

}

/\*\* 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() {

buttonGroup1 = new javax.swing.ButtonGroup();

jLabel1 = new javax.swing.JLabel();

jRadioButton1 = new javax.swing.JRadioButton();

jRadioButton2 = new javax.swing.JRadioButton();

jRadioButton3 = new javax.swing.JRadioButton();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

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

setTitle("SERVER");

jLabel1.setFont(new java.awt.Font("Verdana", 3, 12));

jLabel1.setForeground(new java.awt.Color(51, 0, 51));

jLabel1.setText("CONNECTION :");

jRadioButton1.setBackground(new java.awt.Color(102, 255, 102));

buttonGroup1.add(jRadioButton1);

jRadioButton1.setText("UDP");

jRadioButton2.setBackground(new java.awt.Color(51, 255, 0));

buttonGroup1.add(jRadioButton2);

jRadioButton2.setText("TCP");

jRadioButton3.setBackground(new java.awt.Color(0, 255, 0));

buttonGroup1.add(jRadioButton3);

jRadioButton3.setText("RMI");

jLabel2.setFont(new java.awt.Font("Times New Roman", 3, 12));

jLabel2.setForeground(new java.awt.Color(255, 0, 0));

jLabel2.setText("SERVER IP&PORT");

jLabel3.setFont(new java.awt.Font("Arial", 1, 12));

jLabel3.setForeground(new java.awt.Color(102, 102, 0));

jLabel3.setText("SOCKET IP :");

jLabel4.setFont(new java.awt.Font("Arial", 1, 12));

jLabel4.setForeground(new java.awt.Color(102, 102, 0));

jLabel4.setText("PORT :");

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

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

jTextField2ActionPerformed(evt);

}

});

jTextArea1.setColumns(60);

jTextArea1.setRows(10);

jTextArea1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(204, 204, 0), 2, true));

jScrollPane1.setViewportView(jTextArea1);

jTextArea1.getAccessibleContext().setAccessibleParent(jTextArea1);

jButton1.setBackground(new java.awt.Color(51, 255, 204));

jButton1.setFont(new java.awt.Font("Arial", 1, 12));

jButton1.setText("START LISTENING");

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

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

jButton1ActionPerformed(evt);

}

});

jButton2.setBackground(new java.awt.Color(0, 255, 153));

jButton2.setFont(new java.awt.Font("Arial", 1, 12));

jButton2.setText("STOP LISTENING");

org.jdesktop.layout.GroupLayout layout = new org.jdesktop.layout.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.add(15, 15, 15)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(jRadioButton1, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, 95, Short.MAX\_VALUE)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING, false)

.add(jRadioButton3, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.add(jRadioButton2, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, 95, Short.MAX\_VALUE)))

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.add(61, 61, 61)

.add(jLabel3)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED))

.add(org.jdesktop.layout.GroupLayout.TRAILING, layout.createSequentialGroup()

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 26, Short.MAX\_VALUE)

.add(jButton1)

.add(5, 5, 5))))

.add(layout.createSequentialGroup()

.addContainerGap()

.add(jLabel1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 99, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(133, 133, 133)))

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(jLabel2)

.add(layout.createSequentialGroup()

.add(jTextField1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 84, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(12, 12, 12)

.add(jLabel4)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.UNRELATED)

.add(jTextField2, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 71, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE))

.add(layout.createSequentialGroup()

.add(73, 73, 73)

.add(jButton2)))

.add(58, 58, 58))

.add(layout.createSequentialGroup()

.add(37, 37, 37)

.add(jScrollPane1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 419, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(91, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.addContainerGap()

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jLabel1)

.add(jLabel2))

.add(9, 9, 9)

.add(jRadioButton1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 25, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(0, 0, 0)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.add(jRadioButton2, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 27, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(jRadioButton3, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 26, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE))

.add(layout.createSequentialGroup()

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jLabel3)

.add(jTextField1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(jLabel4)

.add(jTextField2, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE))

.add(32, 32, 32)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jButton2)

.add(jButton1))))

.add(48, 48, 48)

.add(jScrollPane1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 129, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

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

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

// TODO add your handling code here:

jTextArea1.append("performing stop listening"+"\n");

ser.close();

}//GEN-LAST:event\_jTextField2ActionPerformed

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

// TODO add your handling code here:

System.out.println("listening");

s=jTextField2.getText();

int port= Integer.parseInt(s);

if (jRadioButton1.isSelected()==true)

{

try {

System.out.println("in to udp");

ser = new DatagramSocket(port);

System.out.println("socket created");

byte[] receiveData = new byte[1024];

System.out.println("socket created");

jTextArea1.append("listening"+"\n");

System.out.println("socket created");

while (true) {

DatagramPacket rec = new DatagramPacket (receiveData, receiveData.length);

try {

ser.receive(rec);

} catch (IOException ex) {

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

}

//process data

String data = new String(rec.getData());

System.out.println(data);

jTextArea1.append(data+"\n");

}

} catch (SocketException ex) {

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

}

}

if (jRadioButton1.isSelected()==true){

try {

ServerSocket ss = new ServerSocket(port);

//code to read file from socket

} catch (IOException ex) {

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

}

}

}//GEN-LAST:event\_jButton1ActionPerformed

/\*\*

\* @param args the command line arguments

\*/

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

private javax.swing.ButtonGroup buttonGroup1;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JRadioButton jRadioButton1;

private javax.swing.JRadioButton jRadioButton2;

private javax.swing.JRadioButton jRadioButton3;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextArea jTextArea1;

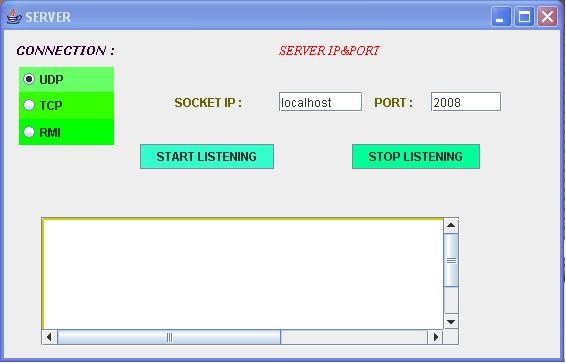
private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

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

}

When we run this code as a java application we see the result as



Now in the client side we choose a file grid.txt from the desktop. Then we choose thye server ip number as localhost. And I choosed the port number as 2008 then we click start transfer button. When transfer is started start time is displayed, when transfer is finished we see the end time and transfer time. When we need to transfer a file we click start listening button on the server side. When we finish the transfer we click stop listening button. Port number we will give the same as given on the client side.

**Task 2- Evaluating UDP Performance**

For this task we need to vary the packet size and we look at the delays. For three packet sizes we get three values of delay.

**Conclusion’s**

1. As we are using UDP there is no connection setup.
2. It is faster than the TCP.
3. Data transfer in UDP is not realiable when compare to TCP.
4. With the increase of data packet size the time delay increases.

**References**

1. **Java network programming**   
       Harold, Elliotte Rusty.
2. <http://eclipseworkspace.googlecode.com/svn/javalabs_udp_test/src/UDPGui/>
3. <http://www.netbeans.org/kb/trails/java-se.html>
4. <http://www.netbeans.org/kb/60/java/editor-tips.html>
5. <http://www.secguru.com/link/socket_programming_tutorial_ppt>