PROJECT REPORT ON

IMPLEMENTATION OF FTP AND RDP PROTOCOLS USING JAVA

Report submitted to the SASTRA Deemed to be University

as the requirement for the course

CSE302: COMPUTER NETWORKS

Submitted by

MANCHIKANTI TEJA

Reg. No.: 222004052

Programme: ELECTRONICS AND COMMUNICATION ENGINEERING

February 2021

DEPARTMENT OF ECE/SRC

KUMBAKONAM, TAMIL NADU, INDIA – 612001

SASTRA
ENGINEERING - MANAGEMENT - LAW - SCIENCES - HUMANITIES - EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 OF THE UGC ACT, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

DEPARTMENT OF ECE/SRC

KUMBAKONAM, TAMILNADU-612001

Bonafide Certificate

This is to certify that the report titled **"Implementation of FTP and RDP Protocols Using Java"** submitted as a requirement for the course, **CSE302: COMPUTER NETWORKS** for B.Tech is a bonafide record of the work done by **Shri. Manchikanti Teja (Reg.No: 222004052, ECE)** during the academic year 2020-21, in the Department of ECE.

Project Base Work held on

Examiner 1 **Examiner 2**

ACKNOWLEDGMENT

Foremost, we might want to express gratitude towards God Almighty for his unlimited favours.

We might want to express our earnest thanks to Dr S. Vaidhyasubramaniam, Vice-Chancellor for providing all facilities and his constant support during our study at SASTRA Deemed University.

We would always stay greatful and we might want to express gratitude towards R. Chandramouli, Registrar for giving an opportunity to do this project.

We express our heartfelt gratitude to Dr V. Ramaswamy, Dean, SRC for his encouragement during our course span.

We are thankful to Dr A. Alli Rani, Associative Dean, SRC, for her inspiration and supervision.

We feel to acknowledge our indebtedness and deep sense of gratitude to Armstrong C Chandrasekaran, Professor for his valuable inputs, guidance and encouragement.

We would like to show our greatful feeling to all the faculty members of our department who have directly and indirectly helped for the successful completion of the project.

LIST OF FIGURES

ABBREVIATIONS

RDP	-	Remote Desktop Protocol
FTP	-	File Transfer Protocol
TCP	-	Transmission Control Protocol
UDP	-	User Datagram Protocol
IP	-	Internet Protocol
GUI	-	Graphical User Interface

ABSTRACT

At present, Computer Networks plays a key role in all our daily life activities. One of the activity which uses system networks is FTP protocol. We are using RDP protocol also in our daily life activities. In this project java programming is used to implement FTP and RDP protocols. We can easily share the files using FTP protocol and we can regulate one system with the help of another system using RDP protocol. FTP is a usual internet protocol providing by TCP used for transferring the files from one system to another system. With FTP, file sharing will be done which has become a major use for all members of the society and it is more utilized because file sharing can be done very faster when compared to normal operation.

RDP allows a client to interconnect with the server. With RDP protocol, we will be able to run applications on the server from the remote client. RDP is a client/server software package permitting remote network admittance with graphical desktop. This software enables us to get the interpretation of the remote machine desktop and thus operate it with our mouse and keyboard. A security enriched approach is implemented using RDP protocol. A growing necessity on the technology tools and an extensive use of technology and communication of facts are detected in our era in a variety of applications and fields like management, economics and society, this project is helpful. The objective of this project is to implement RDP and FTP protocols using java language. In case of RDP protocol, the system should be able to link a remote system system as if it was a local system. This language should permit a complete access to a remote device and all its means. By this project, we are able to share the files and we can access another laptop remotely.

KEY WORDS: Security, Client, Server, Sharing, RDP, FTP.

Table of Contents

CHAPTER 1

INTRODUCTION

FTP PROTOCOL:

FTP means File Transfer Protocol. FTP uses TCP to acknowledge for confirmation after transferring the files. FTP is a usual internet protocol providing by TCP used for transferring the files from one system to another system. It provides the files to be shared quickly. The data that is transferred by FTP protocol is very efficient. FTP is used for transferring the files from the system that acts as client to the system that acts as server for other systems. The data that is transferred by the FTP protocol is also reliable. It is used to inspire the use of remote systems. We know that transferring of files from one system to another is simple but at times it can cause complications. When we consider an example that is two systems can have dissimilar file agreements. Two systems can have dissimilar ways to symbolise text and data. Two systems can have dissimilar directory structures. FTP protocol overwhelms these complications by forming two networks between hosts. One link is used for data transmission and also another link is used for regulate association.

FTP client is a program that is used to implement a file transfer protocol which permits us to transmit the files among the two systems. It permits the operator to link to a remote system and also for uploading files. It is also used for downloading the files. It has some group of instructions that one can use to link to a system and transmit the files among you and your linked system and exit the connected system. The FTP program is also accessible as a built-in module in the web browser. With the help of FTP protocol, transmission has become easy and also does not need to recall the FTP commands.

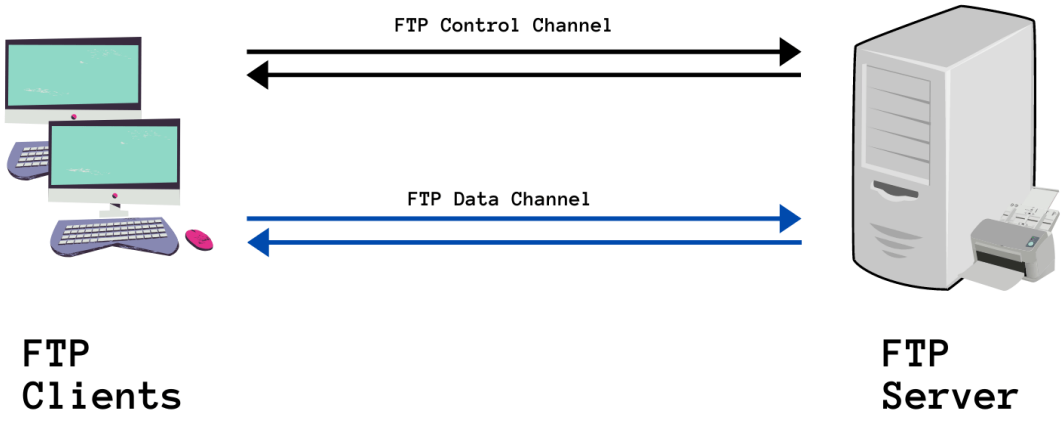


Fig. 1.1 Diagram of FTP protocol

FTP uses discrete regulate and data networks between the client and the server and based on client server architecture. FTP operators can confirm by them itself using some protocols that is commonly in the form of checking the password that is provided and the operator name that is provided by us. We can also connect secretly if the server is organized to allow it. For data transmission to be secured, it is protected by the password and the operator name that is provided and the content provided will be verified and give us a chance to link with the system with the help of another system. FTPs are still shipped with most Windows operating systems. FTP is repeatedly protected by its secure connections. SFTP technologically dissimilar sometimes we can use that also to implement.

The first FTP client requests was command-line requests which are developed before the operational schemes. Numerous FTP clients and because of the growing technology all together developed the mobile devices, servers based on the number of operators, hardware, desktop and FTP has been improved its applications and has been assimilated into throughput application, one example is as Web page editors.

RDP PROTOCOL:

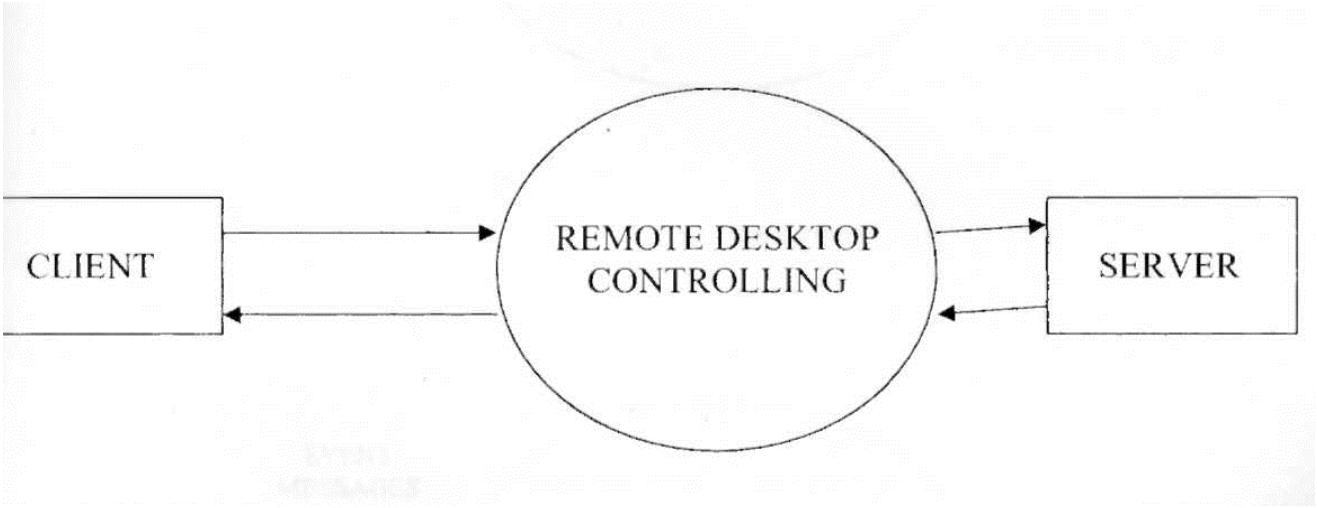
RDP means Remote Desktop Protocol. RDP protocol is the protocol which is developed by Microsoft. Remote desktop protocol is the facility to connect with the other device and use a faraway system from a separate system. By using this RDP protocol, we can regulate one system with the help of another system. Remote desktop operators can regulate their desktop, they can open the files and can also correct the files and use the all apps and all the documents present in the system as if they were actually present at their own system. RDP protocol works by existing operators that permits them to link to another system remotely.

RDP works with the help of multiple dissimilar types of network technologies. RDP is accessible for almost all the versions of windows operating systems. RDP provides more security enhanced approach in order to avoid the complications. RDP uses authentication means that operator name and password are provided and one can access the other laptop only with the help of the operator name and password that is provided. It also provides encryption to avoid the traffic. With the help of this protocol, we are able to run the applications on server system with the help of a client system.

Remote access protocol can also be illuminated as remote regulate of a system through another system linked over the internet or over the personal area network. This is extensively used by countless system constructors and huge businesses help line members for practical problems of their customer's complications. There are innumerable specialized RDP applications. Remote desktop protocol is a tool that is proficient to transmit a system information that is present away from the other system. In another way it can be expressed as this way, it permits the access on your system and it will be running on other system also. Consider an example, we can use RDP to link one system that is present in office and another system that is present in your home and access the all information and all the display information, all files and all system properties as if we are sitting in the office.

Remote Desktop Organiser is the client or server software file permitting remote network access to another desktop. This RDP software permits us to get the display of the remote system and we are able to regulate it with our own mouse and also keyboard. RDP protocol is helpful to achieve remote system connection. RDP software needs a TCP link among the server and the client. Every system will be having a different IP address that is not same with other IP address and can also have the title in DNS. DNS means Domain Name Server. The one who wants to connect should know the IP address and also it can connect when the name of the server is known. In this TCP connection used so the initial handshaking should be present and it involves the Client Initialization and also the Server Initialization communications. When the linking among a client and a server is first recognised, the server begins by asking for confirmation from the client that normally results in the operator that being urged for a password from the client side. Then the server and the client interchange messages to assign the measurements of the desktop and the format of pixel that should be used.

The server and the client are designed as unpretentious as possible and that is usually up to the server to achieve any necessary conversions. Consider an example, the server must give the information of the format of



pixel that should be used and also what the client wants. Every desktop is resembling a simulated X display with the help of origin window on that contains numerous X requests could be exhibited. It make available the key operator of a PC through remote permit to their system. The server handling consist of re-claiming the information of the pixel.

Fig. 1.2 Diagram of RDP protocol

The origin side is established on ordinary work space ideal of a keyboard and also mouse aiming device. The events of Input are transmitted to the server by means of the client when the operator correspondents a key or when the mouse pointer is relocated. RDP programs also ask for all the needs of all probable exact constraints that which the server could manage for example consider mode of the colour, events of pointer and so on.

The suggested system which is RDP protocol is made to be established by using Java language. But the suggested system will be able to execute on any operating systems such as Linux. System explorer is mostly focused on the improvement of the request which is helpful to discover the computers in the arrangement. The

essential portion of the request is a GUI which is helpful to browse other systems present in the arrangement. The GUI must be technologically advanced with the help of Java Swing. The system must be divided into two parts, one is client part and other is server part. With the help of that, every part can be executed differently. The server part must be acted as the server of the file for the both client and for the remote server. Coming to the client part, the client part must give a GUI for relaxed operator communication. All client requests are must be sent to the local server. From local server the requests are sent to the remote server. RDP is the client -server software which is helpful for accessing the remote desktop. This RDP software is fundamentally allocated into two modules. First one is Server Module, it means that executing on the remote machine stand-in as same as the server machine. Second one is Client Module, it means that executing on the client machine by which the user admittances and also manage the remote server machine.

Remote desktop protocol is helpful when we are facing some problems in our system. At that time, we can access our laptop with the help of another system. It means that controlling the system remotely. In order to overcome the problem of losing our files and losing our information and there is no risk of our information when we connect with another system remotely at that time we can use RDP protocol. A security based approach is implemented so that no problem of security in the case of RDP protocol.

A proper authentication is present in the case of RDP which contains username and password protected. A user can connect only when he knows that username and password and also IP address of the system should be known in order to connect the system. It means that authentication is compulsory. So by all these things we can tell that there is no chance of insecurity based approach and it is helpful to almost all the people who are having troubles with their systems. It can implemented on any software that is also one of the major advantage. There are different programs for accessing different parts of the system and for connection also there are so many programs which are present in the source code of the page.

CHAPTER 2

NETWORK CONCEPTS

We have to know some basic concepts and also functionalities regarding implementation of Remote Desktop Protocol. Some of the network concepts which are used in Remote Desktop Controller are as follows.

1. **TCP:**

TCP is abbreviated as Transmission Control Protocol. Transmission Control Protocol gives trustworthy data transmission between the two systems. When two systems want to link to another system consistently, they link together and transmit the required information back and forth with the help of that link. This is equivalent to how to do a phone call. A link between two members is recognized when we dial a mobile number and when he answers the call. We send some information back and forth with the help of that between two members who are speaking to each other with the help of phone lines. Same as the telephone company, Transmission Control Protocol promises that the information transferred from one side of the link really transferred to the another side and also with the same order which has been sent by the sender. Or else an error is informed.

2. **UDP:**

UDP is abbreviated as User Datagram Protocol. User Datagram Protocol is a protocol that transfers the some separate packets of data known as datagrams, it transfers from one system to another system which is not reliable data transfer. UDP is different from TCP.

3. **SOCKETS:**

A Socket is one end theme of a two way interconnection bond considering the two programs executing on the networks. A Socket is assured to a port number so that the Transmission Control Protocol layer can recognize the request that information is intended to be sent.

4. **IP ADDRESS:**

An Internet Protocol (IP) address is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication. IP address is assigned to your device by your internet service provider, like your cable, telephone, wireless it is a requirement to use the internet. This basic information regarding computer networks is enough to understand this project. Now we will see implementation of project.

CHAPTER 3

WORKING

WORKING OF FTP PROTOCOL:

1. Consider one as Client and another one as Server.
2. First we have to execute the server program and after executing that we have to execute the client program.
3. We have to know IP address of server in order to establish connection.
4. Enter IP address of the server in the client side and then we are able to send the files.
5. After files have been sent an acknowledgement will be sent.

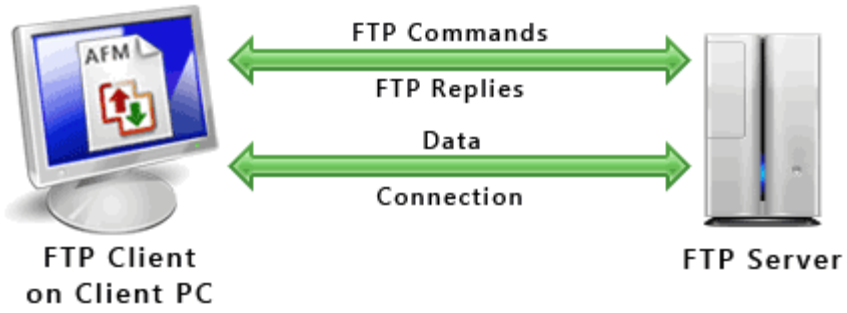


Fig. 3.1 Data flow diagram of FTP

WORKING OF RDP PROTOCOL:

1. To execute this project we need two desktops one will act like a server and another will act like a client. We have several programs in server and client so we will discuss each and every program how they are going to use in both server and client in coding session.
2. First we have to run server program. To run server program first of all we have to setup environment and directories properly. For that in one desktop, first create a folder in any drive and within that folder create one folder and then save all server programs.

3. In another desktop create a folder in any location and within that create one folder and within that save all client programs.
4. **Server Side:** Now in the server desktop open command prompt and run server programs. Then it will generate one small dialog box and it will ask to set password to server. Then you have to set password in server then it will display that it is waiting for connect to client.
5. **Client Side:** When we run server programs then server will wait for client request. Now in another desktop we have to run client programs then small dialog box will open. Now we have to enter server IP address in that dialog box after submitted again small dialog box open in that it will ask to enter server password. Enter server password then we can see server screen in client laptop.
6. In this way we can execute our programs and we will see how remote desktop controller will work.

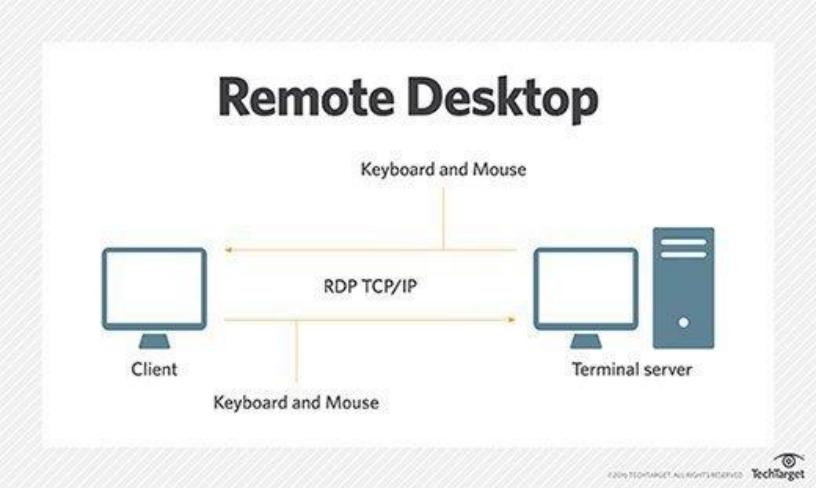


Fig. 3.2 Data flow diagram of RDP

CHAPTER 4

MERITS AND DEMERITS

MERITS OF FTP PROTOCOL:

1. FTP protocol is mostly used in our daily life whenever sharing of files to be done very fast.
2. Major advantage of FTP protocol is that we can send not only more than one file but also we can transmit numerous directories with in single time itself.
3. FTP protocol allows multiple tasking.
4. Even if the connection is lost in the middle of the transmission, this protocol has ability to transmit the files. In case if the connection is lost accidentally, there is no chance of bothering about again file should be transmitted from the first, we can pick up from where the connection has been lost.
5. FTP most important feature is automatic back up. It works well for medical purposes because patient's data are to be protected.
6. FTP protocol is reliable protocol and it is efficient.
7. FTP protocol enables us to share the files very faster.
8. Cost is very low to implement.
9. Security is better because of reliable TCP connection.
10. These are main advantages of FTP protocol.

MERITS OF RDP PROTOCOL:

1. RDP protocol is one of the protocol which is mostly used in our daily life whenever there are troubles facing by us with our devices.
2. It has an easy access. It permits the user to link to the user's required desktop and the user can access the data by using RDP protocol.
3. It has low cost. This RDP service reduce the amounts to be invested on the physical desktop systems. It permits the user to work from his own device.
4. Security is most important feature in the case of RDP protocol.
5. The server device can be very easily updated with fixes that should be done to secure the device. RDP backup the data that is required by the user when there is server crash down.
6. Easy management is possible by the use of RDP protocol.
7. It has more flexibility. The main purpose of RDP protocol is to permit the workers to perform their tasks from literally at everywhere and at any time.
8. RDP protocol is most helpful to the people who have troubles with their own systems.

9. RDP protocol is able to implement only when proper authentication is done, so the security is given top most priority.
10. These are the main advantages of RDP protocol.

DEMERITS OF FTP PROTOCOL:

1. There are some disadvantages in FTP also, some of them are listed below.
2. We are unable to execute simultaneous transmissions using this protocol. FTP does not support some operations such as scheduling transmission among multiple FTP servers.
3. FTP protocol has security but it lacks the complete security.
4. Some changes are very hard to change.
5. It does not contains simple options to transmit the files.
6. To notify our activity, it is impossible to create notifications.
7. These are the some of the disadvantages of FTP protocol.

DEMERITS OF RDP PROTOCOL:

1. There are some disadvantages in RDP protocol, some of them are listed below.
2. Downtime is one of the disadvantage in the case of RDP protocol.
3. Network dependency occurs in case of RDP.
4. Based on the number of users, system performance will be decreased.
5. These are some of the disadvantages of RDP protocol.

CHAPTER 5

SOURCE CODE

SOURCE CODE FOR FTP PROTOCOL:

In this project, we have server and client programs.

Server Code:

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

public class Fileserver
{
        public static void main(String[] args) throws Exception {
                ServerSocket s=new ServerSocket(6777);
                Socket sr=s.accept();
                FileInputStream fr=new FileInputStream("F:\\Filetext.txt")
                byte b[]=new byte[20002];
                fr.read(b,0,b.length);
                OutputStream os=sr.getOutputStream();
                os.write(b,0,b.length);
                }
}
```

Client Code:

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

public class Fileclient
{
        public static void main(String[] args) throws Exception {
                byte []b=new byte[20002];
                Socket sr=new Socket("localhost",6777);
                InputStream is=sr.getInputStream();
                FileOutputStream fr=new FileOutputStream("I:\\tejamanchika.txt");
                is.read(b,0,b.length);
                fr.write(b,0,b.length);
                System.out.println("File transferred");

                }
}
```

These are the programs of server and client to implement FTP protocol.

SOURCE CODE FOR RDP PROTOCOL:

In this project we have server and client programs. We will see each and every program of both server and client.

Server Code:

In server we have 5 programs. We store these programs in a package and if we run main program then all server programs will run and we will get our required output.

5 programs are

- Start.java
- Set_password.java
- InitConnection.java
- Send_screen.java
- Receive_events.java

We will see each and every programs in order.

Start.java

```
public class Start
{
        public static void main(String arg[])
        {
                SetPassword frame1=new SetPassword();
                frame1.setSize(300,80);
                frame1.setLocation(500,300);
                frame1.setVisible(true);
        }
}
```

The above program is main program to the server and by running this program we will start server

Set_password.java

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class SetPassword extends JFrame implements ActionListener
{
        static String port="4903";
        JButton submit;
        JPanel panel;
        JTextField text1,text2;
        String value1,value2;
        JLabel label,label1,label2;

        SetPassword()
        {
                label1=new JLabel();
                label1.setText("set password");
                text1=new JTextField(15);
                label=new JLabel();
                label.setText("");
                this.setLayout(new BorderLayout());
                submit=new JButton("submit");
                panel=new JPanel(new GridLayout(2,1));
                panel.add(label1);
                panel.add(text1);
                panel.add(label);
                panel.add(submit);
                add(panel,BorderLayout.CENTER);
                submit.addActionListener(this);
                setTitle("setting password for client");
        }
        public void actionPerformed(ActionEvent e)
        {
                value1=text1.getText();
                dispose();
                new InitConnection(Integer.parseInt(port),value1);

        }

        public String getValue1()
        {
                return value1;
        }

        public static void main(String arg[])
        {
                SetPassword frame1=new SetPassword();
                frame1.setSize(300,80);
                frame1.setLocation(500,300);
                frame1.setVisible(true);
        }
```

The above program is used to set password to the server desktop.

InitConnection.java

```
import java.awt.*;
import java.io.*;
import java.net.*;
import javax.swing.*;

public class InitConnection
{
	ServerSocket socket=null;
	DataInputStream password=null;
	DataOutputStream verify=null;
	String width="";
	String height="";

	InitConnection(int port,String value1)
	{
		Robot robot=null;
		Rectangle rectangle=null;
		try
		{
			System.out.println("waiting for client to get connected.....");
			socket=new ServerSocket(port);
			GraphicsEnvironment gEnv=GraphicsEnvironment.getLocalGraphicsEnvironment();
			GraphicsDevice gDev=gEnv.getDefaultScreenDevice();

			while(true)
			{
				Socket sc=socket.accept();
				password=new DataInputStream(sc.getInputStream());
				verify=new DataOutputStream(sc.getOutputStream());

				String psword=password.readUTF();
				if(psword.equals(value1))
				{
					verify.writeUTF("valid");
					verify.writeUTF(width);
					verify.writeUTF(height);
					new SendScreen(sc,robot,rectangle);
					new ReceiveEvents(sc,robot);
				}
				else
				{
					verify.writeUTF("invalid");
				}
			}


		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
	}

	private void drawGUI()
	{}


}
```

The above program is use to form connection between client and server. When client program run after entering IP address and password, if both password and IP address are correct then it form connection between client and server.

Send_screen.java

```
import java.awt.*;
import java.io.*;
import java.net.Socket;
import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;

public class SendScreen extends Thread
{
	Socket socket=null;
	Robot robot=null;
	Rectangle rectangle=null;
	boolean continueLoop=true;
	OutputStream oos=null;

	public SendScreen(Socket socket,Robot robot,Rectangle rect)
	{
	this.socket=socket;
	this.robot=robot;
	rectangle=rect;
	start();
	}

	public void run()
	{
		try
		{
			oos=socket.getOutputStream();
		 catch(IOException e)
		 {
			e.printStackTrace();
		 }

		 while(continueLoop)
		 {
			BufferedImage image=robot.createScreenCapture(rectangle);
			try
			{
				ImageIO.write(image,"jpeg",oos);
			}
			catch(IOException e)
			{
				e.printStackTrace();
			}
			try
			{
				Thread.sleep(10);
			}
			catch(InterruptedException e)
			{
				e.printStackTrace();
			}
		 }

	}
}
```

When client program want to access the server screen then, the above program is used to share server screen to client desktop.

Receive_events.java

```
import java.awt.Robot;
import java.io.*;
import java.net.Socket;
import java.util.*;

public class ReceiveEvents extends Thread
{
	Socket socket=null;
	Robot robot=null;
	boolean continueLoop=true;

	public ReceiveEvents(Socket socket,Robot robot)
	{
		this.socket=socket;
		this.robot=robot;
		start();

	}


	p
	{			while(continueLoop)
				{
					int command=scanner.nextInt();
					switch(command)
					{
						case-1:
						robot.mousePress(scanner.nextInt());
						break;
						case-2:
						robot.mouseRelease(scanner.nextInt());
						break;
						case-3:
						robot.keyPress(scanner.nextInt());
						break;
						case-4:
						robot.keyRelease(scanner.nextInt());
						break;
						case-5:
						robot.mouseMove(scanner.nextInt(),scanner.nextInt());
						break;
					}
				}
		}

		catch(IOException e)
		{
			e.printStackTrace();
		}
	}
```

The above program is used when we do any events like mouse clicking, opening a file, typing in client desktop then server will receive events from client can achieve by this program.

CLIENT CODE:

In client we have 6 programs. We store these programs in a package and if we run main program then all client programs will run and we will get our required output.

6 programs are

- Start.java
- Authentication.java
- CreateFrame.java
- ReceivingScreen.java
- SendEvents.java
- Commands.java

We will see each and every programs in order.

Start.java

```
import java.net.*;
import javax.swing.*;
public class Start
{
 static String port="4903";
 public static void main(String arg[])
 {

        String ip=JOptionPane.showInputDialog("please enter server ip address");
        new Start().initialize(ip,Integer.parseInt(port));
 }
 public void initialize(String ip,int port)
 {
        try
        {
                Socket sc=new Socket(ip,port);
                System.out.println("connecting to the server....");
                Authentication frame1=new Authentication(sc);
                frame1.setSize(300,80);
                frame1.setLocation(500,300);
                frame1.setVisible(true);

        }
        catch(Exception e)
        {
                e.printStackTrace();
    }


 }
}
```

The above program is the main program to start client program. Don't confuse with Start.java name there are 2 Start.java programs one is in server and another one is client.

Authentication.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
class Authentication extends JFrame implements ActionListener
{
        private Socket cSocket=null;
        DataOutputStream passchk=null;
        DataInputStream verification=null;
        String verify="";
        JButton submit;
        JPanel panel;
        JLabel label,label1;
        String width="",height="";
        JTextField text1;
        Authentication(Socket cSocket)
    {
                label1=new JLabel();
                label1.setText("password");
                text1=new JTextField(15);
                this.cSocket=cSocket;
                label=new JLabel();
                label.setText("");
                this.setLayout(new BorderLayout());
                submit=new JButton("submit");
                panel=new JPanel(new GridLayout(2,1));
                 passchk.writeUTF(value1);
                 verify=verification.readUTF();
        }
        catch(IOException e)
        {
                e.printStackTrace();
        }
        if(verify.equals("valid"))
        {
                try
                {
                        width=verification.readUTF();
                        height=verification.readUTF();
                }
                catch(IOException e)
                {
                        e.printStackTrace();
                }
                CreateFrame abc=new CreateFrame(cSocket,width,height);
                dispose();
        }
        else
        {
                System.out.println("please enter valid password");
                JOptionPane.showMessageDialog(this,"password is incorrect","Error",JOptionPane.ERROR_MESSAGE);
                dispose();
        }
    }
}
```

CreateFrame.java

```
import javax.swing.*;
import java.net.Socket;
import java.io.InputStream;
import java.beans.PropertyVetoException;
import java.awt.BorderLayout;
import java.util.zip.*;
import java.io.IOException;

class CreateFrame extends Thread
{
	String width="",height="";
	private JFrame frame=new JFrame();
	private JDesktopPane desktop=new JDesktopPane();
	private Socket cSocket=null;
	private JInternalFrame interFrame=new JInternalFrame("Server Screen",true,true,true);
	private JPanel cPanel=new JPanel();
	public CreateFrame(Socket cSocket,String width,String height)
	{
		this.width=width;
		this.height=height;
		this.cSocket=cSocket;
		start();
	}
	public void drawGUI()
	{
		frame.add(desktop,BorderLayout.CENTER);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setExtendedState(frame.getExtendedState()|JFrame.MAXIMIZED_BOTH);
		try
		{
			interFrame.setMaximum(true);
		}
		catch(PropertyVetoException ex)
		{
			ex.printStackTrace();

		}
		cPanel.setFocusable(true);
		interFrame.setVisible(true);

	}


	public void run()
	{
		InputStream in=null;
		drawGUI();
		try
		{
			in=cSocket.getInputStream();
		}
		catch(IOException e)
		{
			e.printStackTrace();
		}
		new ReceivingScreen(in,cPanel);
		new SendEvents(cSocket,cPanel,width,height);


	}


}
```

ReceivingScreen.java

```
import java.io.ObjectInputStream;
import java.io.InputStream;
import javax.swing.JPanel;
import java.awt.Image;
import javax.imageio.ImageIO;
import java.awt.Graphics;
import java.io.IOException;
import java.io.ByteArrayInputStream;


class ReceivingScreen extends Thread
{
	private ObjectInputStream cObjectInputStream=null;
	private JPanel cPanel=null;
	private boolean continueLoop=true;
	InputStream oin=null;
	Image image1=null;

	public ReceivingScreen(InputStream in,JPanel p)
	{
		oin=in;
		cPanel=p;
		start();
	}

	public void run()
	{
		try
		{
		  while(true)
		  {
			byte[] bytes=new byte[1024*1024];
			int count=0;
			do
			{
				count+=oin.read(bytes,count,bytes.length-count);
			}while(!(count>4&&bytes[count-2]==(byte)-1&&bytes[count-1]==(byte)-39));
			image1=ImageIO.read(new ByteArrayInputStream(bytes));
			image1=image1.getScaledInstance(cPanel.getWidth(),cPanel.getHeight(),Image.SCALE_FAST);

			Graphics graphics=cPanel.getGraphics();
			graphics.drawImage(image1,0,0,cPanel.getWidth(),cPanel.getHeight(),cPanel);
		  }
		}
		catch(IOException e)
		{
			e.printStackTrace();
	}
	}


}
```

This program is use to receive screen from server to client desktop.

SendEvents.java

```
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseMotionListener;
import java.io.PrintWriter;
import java.io.IOException;
import java.net.Socket;
import javax.swing.JPanel;


class SendEvents implements KeyListener,MouseMotionListener,MouseListener
{
		private Socket cSocket=null;
		private JPanel cPanel=null;
		private PrintWriter writer=null;
		String width="",height="";
		double w;
		double h;

		SendEvents(Socket s,JPanel p,String width,String height)
		{
			cSocket=s;
			cPanel=p;
			this.width=width;
			this.height=height;
			w=Double.valueOf(width.trim()).doubleValue();
			h=Double.valueOf(width.trim()).doubleValue();

	    cPanel.addKeyListener(this);
			cPanel.addMouseMotionListener(this);
			cPanel.addMouseListener(this);

			try
			{
			writer=new PrintWriter(cSocket.getOutputStream());
			}
			catch(IOException e)
			{
			e.printStackTrace();

		public void mouseDragged(MouseEvent e)
		{
		}

		public void mouseMoved(MouseEvent e)
		{
			double xScale=(double)w/cPanel.getWidth();
			double yScale=(double)h/cPanel.getHeight();
			writer.println(Commands.MOVE_MOUSE.getAbbrev());
			writer.println((int)(e.getX()*xScale));
			writer.println((int)(e.getX()*yScale));
			writer.flush();
		}


		public void mouseClicked(MouseEvent e)
		{}


		public void mousePressed(MouseEvent e)
		{
			writer.println(Commands.PRESS_MOUSE.getAbbrev());
			int button=e.getButton();
			int xButton=16;
			if(button==3)
			{
				xButton=4;
			}
			writer.println(xButton);
			writer.flush();
		}
```

```
		public void mouseReleased(MouseEvent e)
		{
			writer.println(Commands.RELEASE_MOUSE.getAbbrev());
			int button=e.getButton();
			int xButton=16;
			if(button==3)
			{
				xButton=4;
			}
			writer.println(xButton);
			writer.flush();
		}

		public void mouseEntered(MouseEvent e)
		{}

		public void mouseExited(MouseEvent e)
		{}

		public void keyTyped(KeyEvent e)
		{}

		public void keyPressed(KeyEvent e)
		{
			writer.println(Commands.PRESS_KEY.getAbbrev());
			writer.println(e.getKeyCode());
			writer.flush();
		}

		public void keyReleased(KeyEvent e)
		{
			writer.println(Commands.RELEASE_KEY.getAbbrev());
			writer.println(e.getKeyCode());
			writer.flush();
		}

	}
```

The above program is used to transfer events from client desktop to server desktop.

Commands.java

```
public enum Commands
{
	PRESS_MOUSE(-1),
	RELEASE_MOUSE(-2),
	PRESS_KEY(-3),
	RELEASE_KEY(-4),
	MOVE_MOUSE(-5);

	private int abbrev;
	Commands(int abbrev)
	{
		this.abbrev=abbrev;
	}
	public int getAbbrev()
	{
		return abbrev;
	}
}
```

Through these 11 programs we can achieve remote desktop controller of one desktop to another.

CHAPTER 6

SNAPSHOTS

FTP PROTOCOL:

1. First save the both client and server programs.
2. Run the server and after that run the client programs.

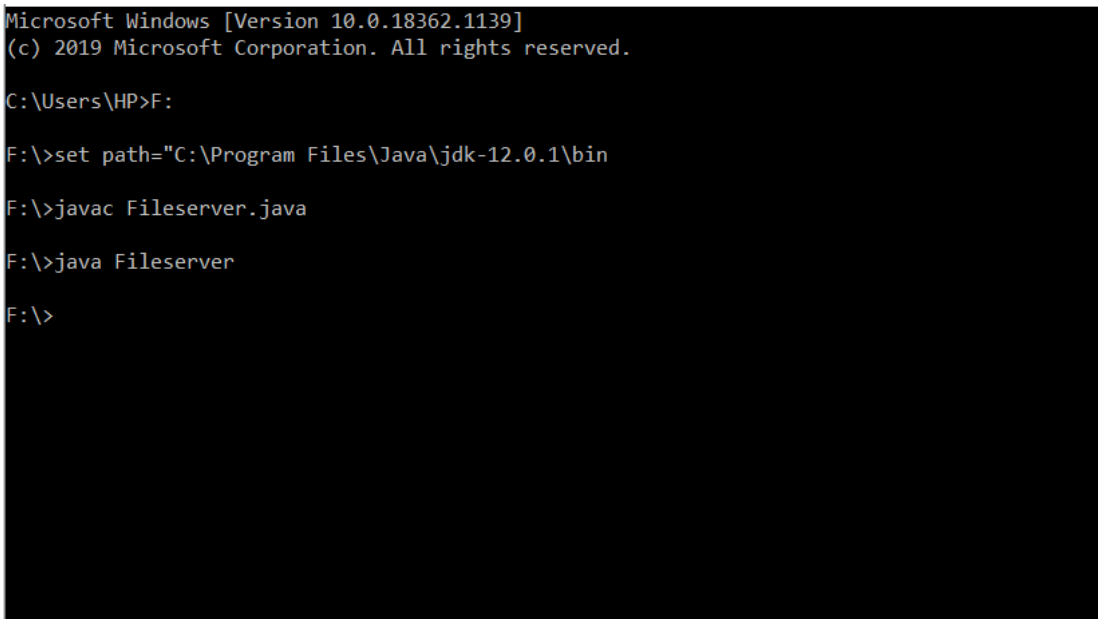


Fig. 6.1 Execution of server program

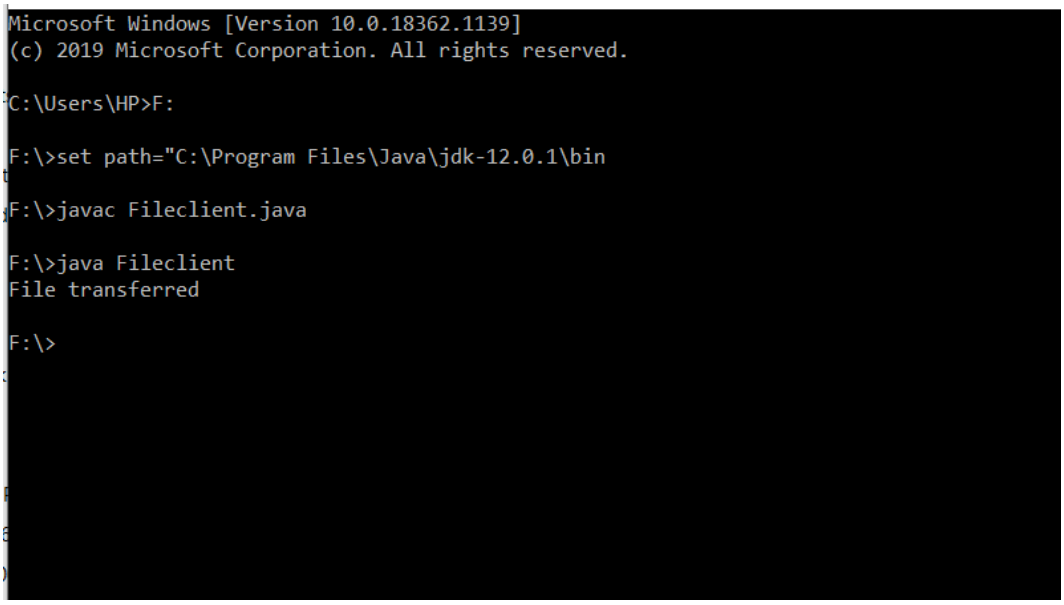


Fig. 6.2 Execution of client program

3. The output is as follows.

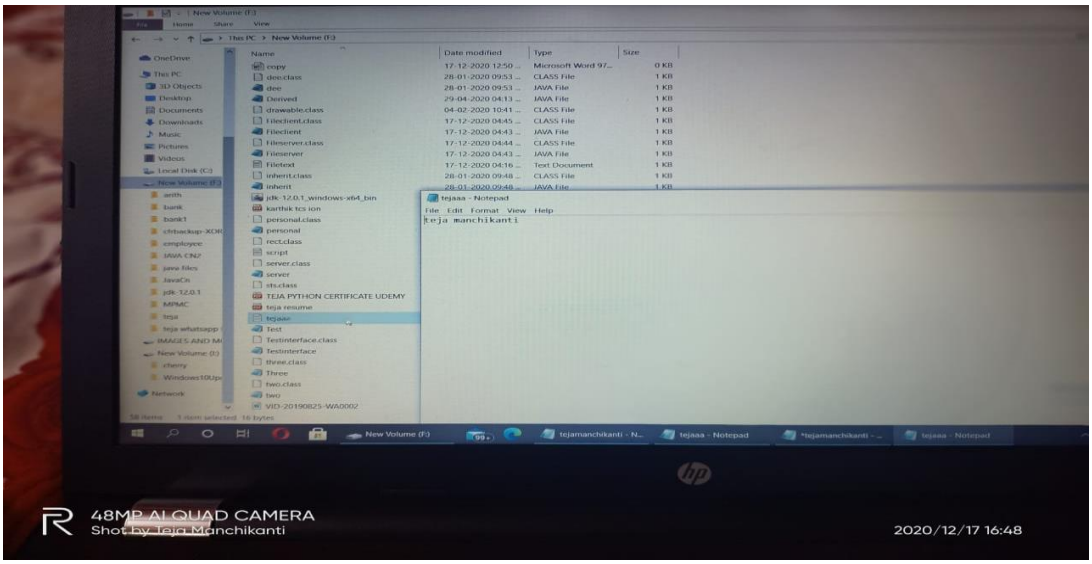


Fig. 6.3 Initial location of file

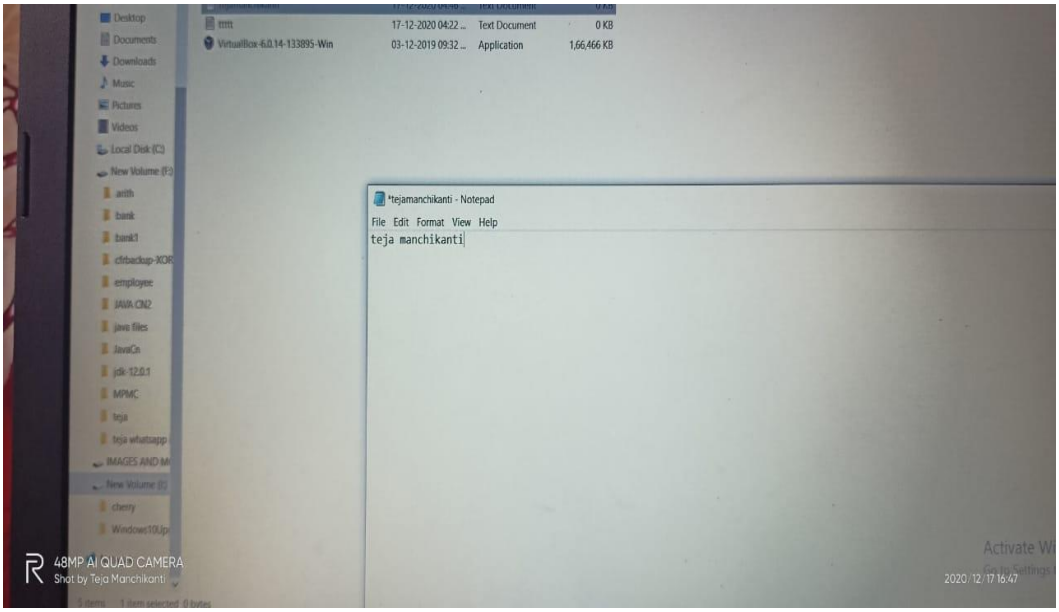


Fig. 6.4 Final location of File

RDP PROTOCOL:

1. Initially we have to take two laptops and run server program in one desktop and client program in another desktop.
2. First run server program after running small dialog window open and ask to set password for server

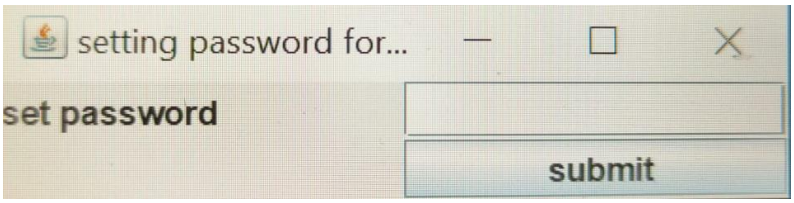


Fig. 6.5 setting password

3. After setting password then server is waiting to connect with client.

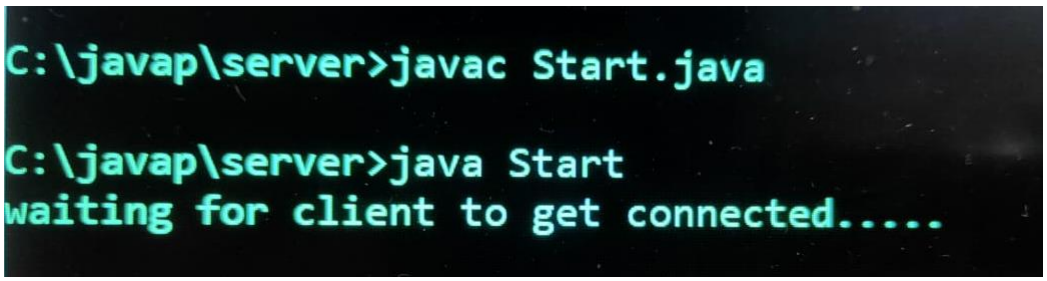


Fig. 6.6 Dialog box of waiting for client

4. Now we have to run client program then one small dialog box will open and it will ask to enter server IP address.

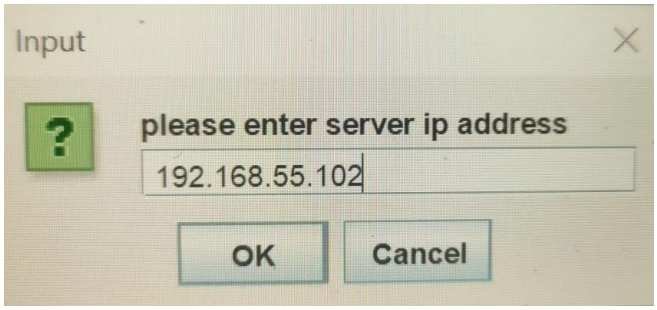


Fig. 6.7 Dialog box for entering Ip address

5. Now after entering server IP address again small dialog box will open and it will ask to enter password.

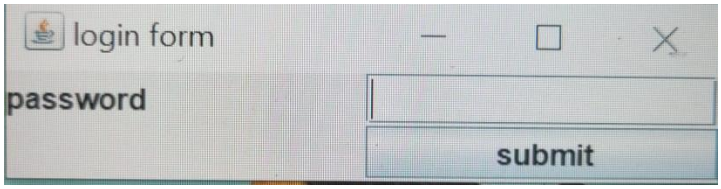


Fig. 6.8 Dialog box for entering password

6. After enter correct password then we can see server screen in client desktop as shown in figure.

Fig. 6.9.Server screen in client desktop

7. Two laptops have same screen as shown in figure, now client can access server desktop.

Fig. 6.10. Final display of two systems

CHAPTER 7

CONCLUSION AND FUTURE PLANS

CONCLUSION:

- By considering the networking location the Manager has to manage the total things. For that reason itself, proficient computing abilities are done in order to avoid losses.
- This project implementation of FTP and RDP protocols after checking, it has been establish that it has achieved the desired output.
- The system has been secured from all types of unknown activities by providing user name and also password when we have to log in to the other system. RDP software is almost manageable software.
- Implementation of both protocols has been tested with real information and it has been found to be with no error.
- It is also found that the system has been working effectively after implementing the protocols.
- Security is one of the most considered part of the project.
- All the required authentications are done in this implementation, because of that all kind of people will be able to use this software and required information.

FUTURE PLANS:

- Since the Technology has been updating so quickly and also by considering the location of networking the manager has to manage the total things.
- RDP has vast capability over the Internet-based architectures and also, has vast capability over the service-oriented architectures.
- To subsist as of that race each and every system have to do some changes to that in the future.
- New modifications will give the system a good look so that it can fascinate so many administrators.
- Because of this reason it is a requirement that the system has to be changed based on the users needs. Some of the future improvements is

    1. The present usefulness mainly meant for environments of intranet, for example virtual classrooms is also comes under huge range of application areas of intranet environments.
    2. Next generation of value package hypothetical on the internet based applications.
    3. File compression before transferring that file to the requested machine and also performing decompression on the reception.
    4. Additional differences target to take account of abilities such as controls and extra operator controls on the Remote systems.

CHAPTER 8

REFERENCES

1. https://en.wikipedia.org/wiki/Remote_Desktop_Protocol
2. https://www.parallels.com/blogs/ras/rdp-meaning/
3. https://en.wikipedia.org/wiki/File_Transfer_Protocol#:~:text=The%20File%20Transfer%20Protocol%20(FTP,the%20client%20and%20the%20server.
4. https://www.geeksforgeeks.org/file-transfer-protocol-ftp-in-application-layer/
5. https://www.javatpoint.com/computer-network-ftp
6. https://computingforgeeks.com/how-to-enable-remote-desktop-protocol-rdp-on-windows-server-2019/
7. https://searchenterprisedesktop.techtarget.com/definition/Remote-Desktop-Protocol-RDP
8. https://www.howtogeek.com/175087/how-to-enable-and-secure-remote-desktop-on-windows/