International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 10, October- 2014

Simple File Transfer System using GUI and Socket Programming for Window Operating System

Charles Okanda Nyatega
Mbeya University of Science and Technology

Abstract This paper is focused on the file transferring through the WAN without using server where by two or multiple users connect and transfer files to each other using the system designed. The system also save log so that it will be easy to read the history of the file transfer any time the user logs into the computer. For the successful implementation of the project I use Graphic User Interface (GUI) programming in java and socket programming for Window Operating System. In this paper a general Java language is used so that the design codes can run on any system. In the last part of this paper are the results showing in details for each file transfer from computer with IP address 192.168.0.2 to 192.168.0.3, finally the problems faced and how to solve them so that the system is safe and can run in high efficiency.

Keywords: TCP/IP; file transfer; Java

1. INTRODUCTION

1.1 Requirement

This system is a file-transfer system that can help users connect in the network, it works in the Wide Area Network. The system could help users find anyone who use the same system in the WAN, meanwhile it can transfer file, The file-transfer module allows the user to transfer file to the other. This module could provide the user with the log of the file-transfer including time, the name of the file, as well as the directory where the file is saved if you are the receiver, or the directory where you choose the file if you are the sender..

Between users, it could save the log of the talk after the file transfer showing both IP addresses of the machine used, time and the name of the file transferred. Since file transferring includes either server-client model or client-client model, this system is a little different from the above mentioned software-designs, this time when implementing it, the user could not use any server to transfer the file, in other words the user transfer file directly without any transfer-station.

Through the use of these sharing systems things become much easier especially in economic level in such a way that people started to share information and data such as files containing electronic books, mp3 etc., this is simply by one or few people buying the product and sharing it with the rest hence economically better, Examples of existing systems include Bittorent, Utorrent etc, where by These services also allowed users to download files other than music, such as movies and games[4]

1.2 Architecture of the design

What's interesting is that more computers and devices are getting used to talking to one another over networks in an automated fashion[1]

As two machines in the WAN using remote IPs and this composed of participants that make a portion of their files directly available to other network participants, without the need for central coordination instances. Both users are suppliers and consumers of files, in contrast to the traditional client-server model where only servers supply, and clients consume.

IJERT

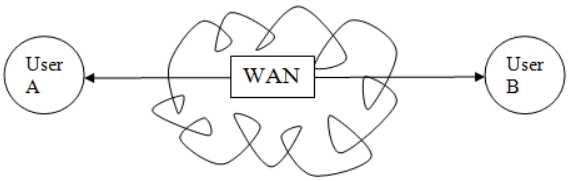

Figure 1 System design

2. DESIGN OF THE SOFTWARE

In the architecture of this system, it is about designing modules that are generally grouped into three main categories which are the graphical user interface (GUI), Socket and file read/write. By connecting three of them together, the formation of file transfer system will be possible. The figure below is the summary of this part.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

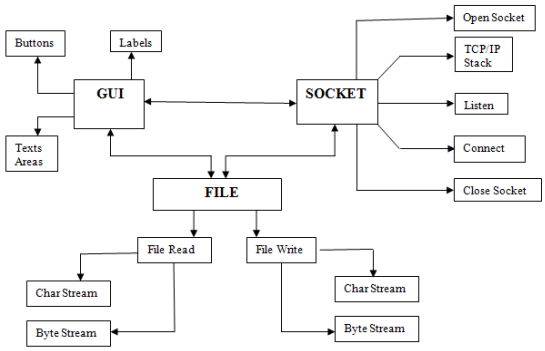

Figure 2 Modules

Figure 2 shows the relationship between graphical user interface, socket and file.

The following figures shows both searching for file (Sender) and downloading file (Receiver) flowcharts for this system

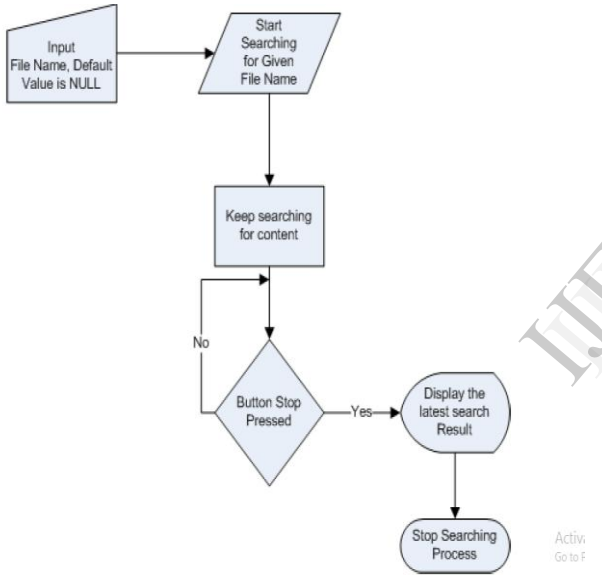

Figure 3 Searching for File flowchart[6]

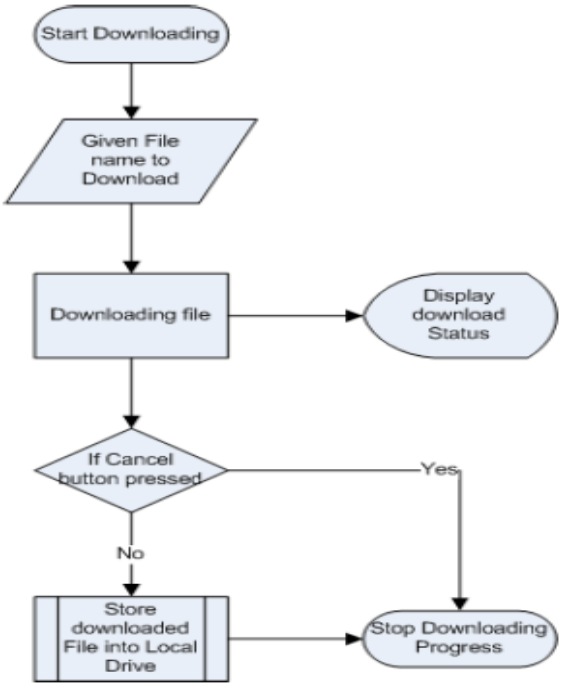

Figure 4 Downloading Flowchart[7]

3. TEST AND RESULTS

This chapter gives a quick implementation of the paper using java technology. Talk about java technology with most programming languages, you either compile or interpret a program so that you can run it on your computer. With the compiler, first you translate a program into an intermediate language called *java bytecodes*. The interpreter runs each java bytecode instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed.

A platform is the *hardware* or software environment in which a program runs. Most of the platforms can be described as the combination of the operating system and the hardware.

It has two components which are Java Virtual Machine (Java VM) and Java Application Programming Interface (Java API). in this project there is one main class called *FileSender.java* and other four subclasses *Login.java, Send.java, Receive.java and History.java.* whereby FileSender.java is the main class connecting all of the sub-classes, Login.java allows user to log in and saves data in *log.txt* with time in the format *yyyy:mm.dd hh:mm:ss* , *Send.java* assigns port number and IP address here to match the host's computers to enable the sending of the file. *Receive.java* works opposite to *Send.java* whereby as long as *Receive* button will be pushed it will start *Receiving File…*and it will *connec*t in the netbean window where by *Receive* must know the IP address of the server and both port numbers should match and finaly *History.java* helps the user to read the history of the previous talk, it serves the log file showing both sender and receiver IP addresses, file names, and the location of the files in the given machines as well as the time they were being sent or received.

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

In order to see test and results in this project firewall must be turned off. If the window firewalls are turned on, it will not be possible for the ping process since it will automatically block the unauthorized communication between the host machine and other machines on the network.

```
C:\Documents and Settings\Charles.HOME-A972C9918E>ping 192.168.0.2

Pinging 192.168.0.2 with 32 bytes of data:

Reply from 192.168.0.2: bytes=32 time<1ms TTL=64
Reply from 192.168.0.2: bytes=32 time<1ms TTL=64
Reply from 192.168.0.2: bytes=32 time<1ms TTL=64
Reply from 192.168.0.2: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\Charles.HOME-A972C9918E>_
```

Figure 5 Ping process

3.1 Sending process between two computers in the network

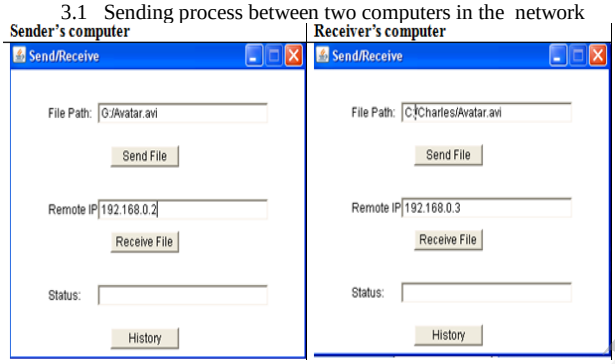

Figure 6 Sending/Receiving File

A receiver **must** know the sender's IP address, where by a sender doesn't need to know receiver's IP address. Using two computers with IP addresses *192.168.0.2* and *192.168.0.3* respectively. If *192.168.0.2* wants the file from*192.168.0.3*, since the sender computer doesn't need to know the IP of the receiver, the field should be empty, or with the *localhost* IP or with the loopback IP (127.0.0.1), to make a connection.

The previous figure shows both computers initial interfaces just before send button is pushed by the sender. When *Send File* is pushed at last by the sender, the status will change, and the whole process will be shown in the table below in both computers.

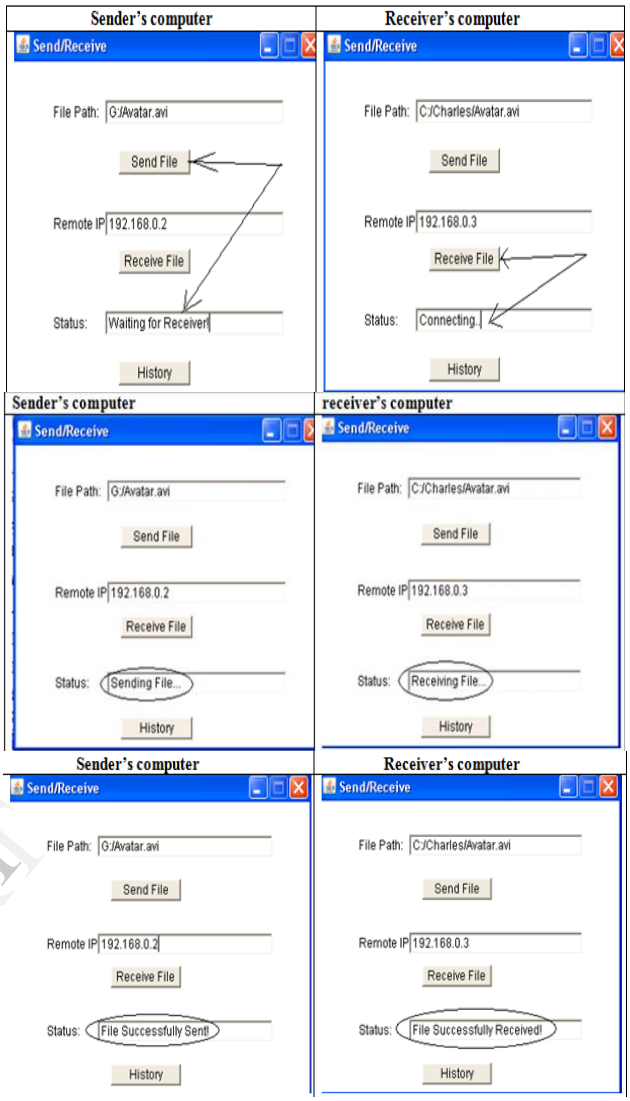

Figure 7 Status Change

The *Avatar.avi* file with the size *1.36G* was successfully sent and successfully received from *192.168.0.2* computer to *192.168.0.3* in few seconds!

3.2 History

We will be able to read the history just by hitting the history button since the program serves the log, this will make the job done to be remembered any time someone logs into the computers and as it will show time the file was transferred, IP of both host and user computers and the name of the files being transferred. The log of the talk serves as many history as the user wants and it can be deleted manually when the user doesn't want it any more. It is often stored in the program file.

The following is an example of the Avatar file exchange in the next figure;

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

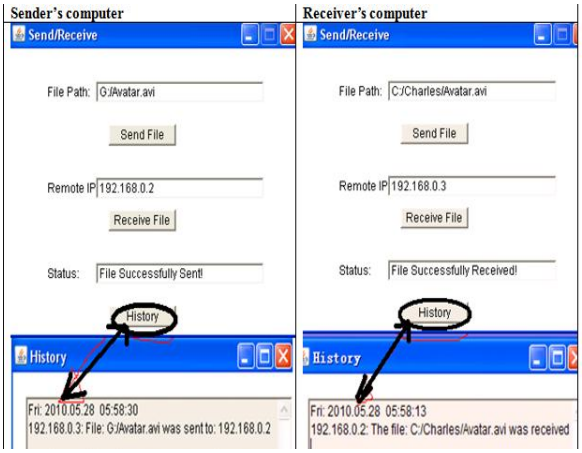

Figure 7 History

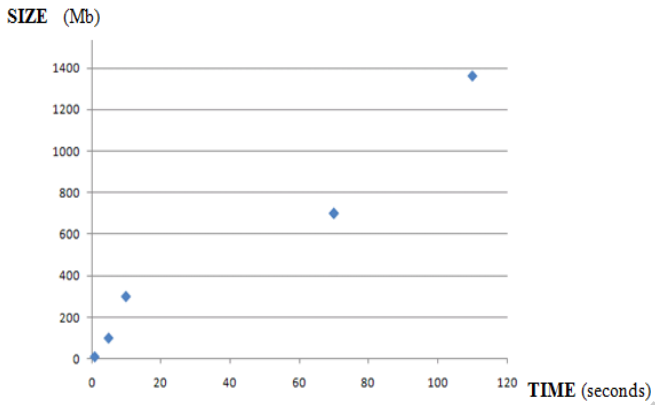

Figure 8 The graph of size against time

The graph shows the relationship between the file transfer with the time, as file gets bigger takes more time also it takes less time as the file becomes smaller.

4. CONCLUSION

As this system work as peer to peer, we know that Peer-to-peer file sharing is also efficient in terms of cost[2], the system administration overhead is smaller because the user is the provider and usually the provider is the administrator as well. Hence each network can be monitored by the users themselves. At the same time, large servers sometimes require more storage and this increases the cost since the storage has to be rented or bought exclusively for a server. However, usually peer-to-peer file sharing does not require a dedicated server[3]

This paper based heavily on socket programming and used Java language to design it due to the advantage of Java language over other languages.

Even though this paper was successfully done there some of the problem I encountered during the work, in order to make a possible transfer of the files the following should be taken care of.

4.1 Checking the port numbers

I learn that if port numbers are not careful observed the transfer could not be possible, so the first thing to do is checking the ports numbers put in the codes in both machines to make a possible transfer. If the ports numbers are not the same it will not be possible for transfer.

4.2 Turning off Window Firewall

If the window firewalls are turned on, it will not be possible for the ping process since it will automatically block the unauthorized communication between the host machine and other machines on the network. So by turning it off will make it possible for the ping and pong processes hence communication will take place though by doing this applications act as servers as well as clients, meaning that they can be more vulnerable to remote exploits[5] which is the main challenge i will try to solve in the future though now it is possible to ping without disabling it, depends on different operating systems.

5. REFERENCES

[1] Danny Briere, and Pat Hurley, "Wireless Home network For Dummies", 4th Edition, pp50.
[2] Babaoglu, Ozalp (2012). "Introduction to Peer-to-Peer Systems". *Complex Systems*. Universitá di Bologna. Retrieved 6 February 2013.
[3] Winkelman, Dr. Roy. "Software". Florida Center for Instructional Technology College of Education, University of South Florida,. Retrieved 6 February 2013.
[4] Tyson, Jeff." How the Old Napster Workded".
[5] Vu, Quang H.; et al. (2010). *Peer-to-Peer Computing: Principles and Applications*. Springer. p. 8
[6] Seyed Saeed Ghiassy, "Peer-to-Peer File Sharing",63CP3261 Final Year Project, pp23.
[7] Seyed Saeed Ghiassy, "Peer-to-Peer File Sharing",63CP3261 Final Year Project, pp2

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)