Design and Implement Chat Program Using TCP/IP

Article · April 2019

1 author:

Mohammed A. Ahmed
College of Engineering, Al-Iraqia University
7 PUBLICATIONS   **5** CITATIONS

DESIGN AND IMPLEMENT CHAT PROGRAM USING TCP/IP

Mohammed A. Ahmed[1], Sara Ammar Rafea[2] , Lara Moufaq Falah[3] , Liqaa Samir Abd Ullah [4]

[1] Network Department/ College of Engineering/ Al-Iraqia University

mohtikr03@yahoo.com

[2,3,4] Students at Network Department/ College of Engineering/ Al-Iraqia University

saraammar75@gmail.com

Abstract - LAN Chat Messenger using TCP/IP offers reliability, security and zero cost communication among staff members of a company. As well, this study offers file transfer. It helps to solve the communication problems that related to time and cost. The proposed protocol facilitates information exchange among individuals by providing many communication options. It is a standalone application using JAVA as the programming language and tested at LANs of our institute (college's Labs networks).

Keywords - LAN; FTP; UDP; Chat; TCP; IP.

I. INTRODUCTION

The objective of this work is to visualize the history of communication via focusing on important technologies through previous decades. The elements of communication here are basically which from the pillars of communication history are telephone, mail and telegraph. The Internet has introduced the opportunity to form the cloud of "digital convergence" which covers communications, computing and broadcasting in a single coherent system within a global network. The variation between broadcast communication and point-to-point communication began to blur through the previous few decades. Point-to-point machine-mediated communications have been realized by introducing voice response and answering machines. Net news which is facilitated though personal web pages have opened the gate to cover large audience and thus, large audience can be covered. We can expect this evolution of communications to continue, and eventually to achieve that convergence in which there will be a continuum between point-to-point and broadcast communication. A common feature of all communication services has been that prices have usually had only a slight relation to costs. Average prices have, over long periods, decreased in line with the decline in costs. However, the structure of prices has traditionally included crosssubsidies. It is often difficult to tell when such subsidies exist. One could hope that at least the economics of postal systems would be well understood. These systems are large, relatively slowly changing, and unusually open to public scrutiny .Many telephone administrations have made great mistakes in their assessment of the future development of telecommunications. They have been altogether too pessimistic concerning the possibilities of expansion. Both as regards number of subscribers, telephones, and traffic, certain saturation values were assumed at an early stage. These values have constantly proved to be too low [1].

Nowadays, we can see the old bureaucracy in many organizations facilities although of the advancements of communication and network technology. Actually, these organizations did not adopt new pattern of communication between staff members besides not comprehending the concept of Instant Messaging Service. The basic pattern of communications realized by relying on the conventional methods of communication, as example, the staff members walk from office to office to conduct the required communication. In addition, the staff may use mobile phone calls, which is costly and stressful to communicate between each other. Some organization might deal with emails to send work-based information. But in this case, the intended receipt must be online in order to check and read message [2]. The advances in electronic messaging services and its synchronous nature can bring different advantages via improving business activities, save cost, energy and time. It also takes place in different forms such as speech, text, signals and the likes. It can also be said to be a conceptual way of exchanging information between entities [3]. This paper is structured as follows: Section 2 of this research discusses about the project interface. Section 3 describes how the client connected to server. Section 4 is the project's protocol. Section 5 gives the output of project's implantation. Finally is the conclusion of the research or the project.

II. THE PROJECT INTERFACE

A.   GUI Server

Figure1 represents server GUI (interface) which consists of text area, four button and text filed. When server starts working the start button, stop button, online users' button and clear button appear in disable mode. The server cannot start working without a port number, the start button still disable until the port number is entered, the port number should be between (0 to 1024). By clicking on start. the text area will display the date, port number and host name, the start button became disable and all other buttons become enable. The server wait for client to connect, text area display all connected and disconnected users. The stop button responsible for closing all sockets. The online users button responsible for displaying online users. The clear button used for clear text area.
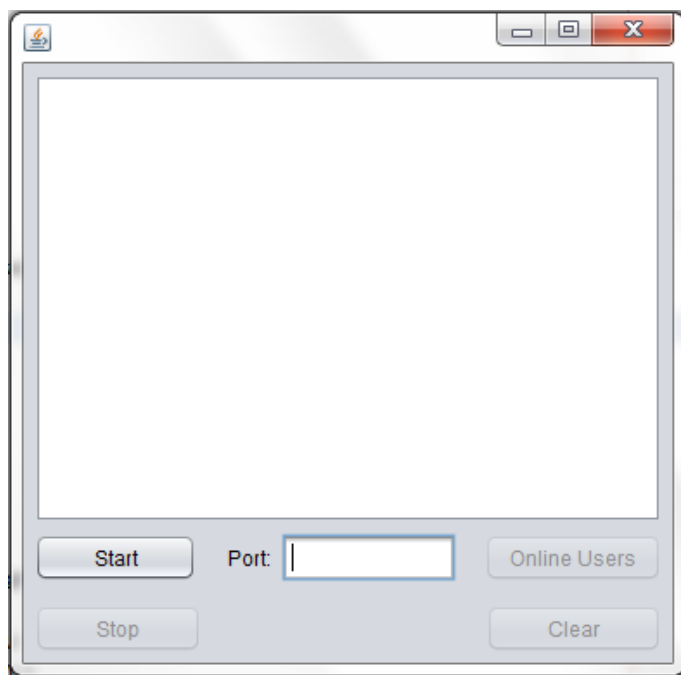
Fig.1: Server Interface

Fig.2: Client Interface

C.   GUI of Private Client

Fig.3: Private Client Interface

B.   GUI Client

Figure2 represents client interface which consists of four text field for user name (which represents current user name), address (which represents host name of IP address), port (which represents port number) and message to be send and three button for connect, disconnect and send and text area to display the messages and jlist (which displays online users). To allow client connect to server must enter username, port number and address of server, then click on connect button, if one of this field empty the client ask user to enter the empty field, if the username is already connected to the server, the server ask client to change username. The disconnect button is enable when client is connect and connect button became disable. When client is connect the port, address and username is unable to change (text field is unable) until client is disconnected also cannot close the frame without click disconnected button. The Send button responsible for sending messages. jlist displays all connected users, Which enables the current user to chat privately with any another user by click on his/her name, the first name on jlist is the username of client and don't do any action when press on it.
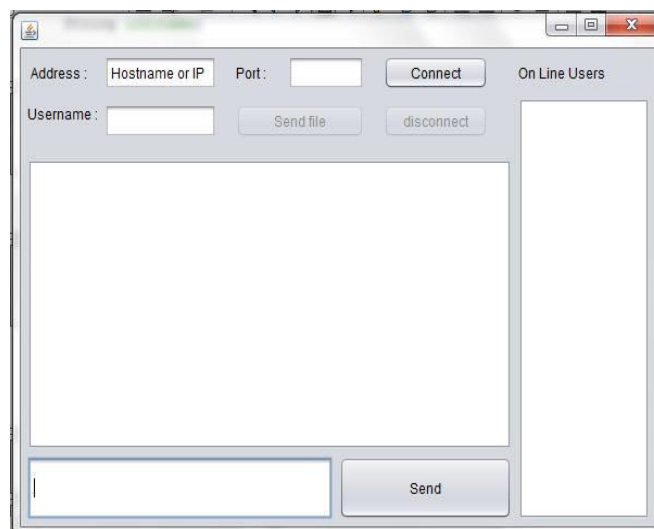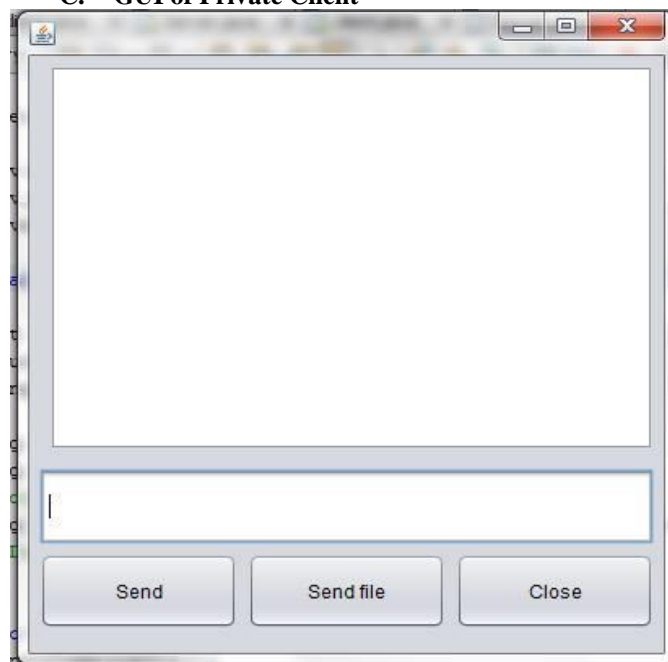
By clicking on any username listed in jlist in order to start a private conversation. The frame shown in Figure3 well display to each of them: The frame contains three buttons to send messages, files and to close connection. The text area display conversation and a text field to write messages

III. HOW CLIENT CONNECT TO SERVER

Server must be running first and waiting (listening) for a connection, server listening at specific port number and create thread for each client connection. Each client has the same port number that the server listening at it and hostname of server (or IP).
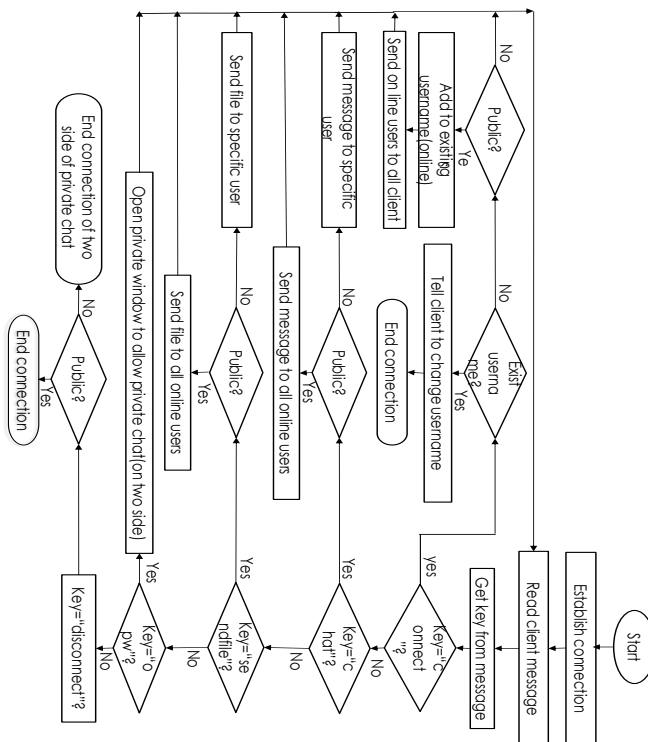

Fig.4: The mechanism of the proposed protocol

IV. The Project's Protocol

The protocol designed to control transfer of data between clients/server (the protocol is a format used to send the data back and forth). Flowchart shown in Figure4 explain the work mechanism of the proposed protocol.

Every message exchanges between client and server contain a key so this protocol depends on key to identify the operation that the server should perform (connect user, disconnect user, send message to all online users, send message to specific user, send file to all online user or send file to specific user). The format of each message is:

User name: message: key

User name: the server should know the source of message (the sender) to tell all online user the sender (in public chat) and to know the destination of message (in private chat)

message: the message to be send

key: to tell server what to do to this message(send to all online users, send to specific user, ……)

Now how the server/client get the key value from the message transfer between client/server?

The server and client used "split" class to split the message according to ":" and put the result in array (data [ ]), so the content of array is:

data=[username message key]

that's mean:

The first element in this array is the username and the second element is the message and the third element is the key(data[0]=username, data[1]=message, data[2]=key).

Each received key by server/client compare with the key in server/client then the server/client decide what to do.

There are many keys used in server and in client:

keys in server:

- "connect": when server received "connect" that's mean that the new client(public or private) is connect, The server perform some operation:Check if the new username is used by other user. if username is exist the server send message to client to change username and then end the connection, else the server add username and socket to hashtable(ht). Then check if the username is public or private. If public add it to arraylist (users), then send the arraylist to all connected users to update client online list.

- "disconnect": when server received "disconnect" that's mean that the client(public or private) is disconnected, so the server remove username from hashtable and from arraylist if the client is public, or remove the username of both side of private connection.

- "chat": this tell server to send message to all online users if username is public else send message to specific user.

- "File": if client public the server send file to all other connected client, else if client is private the server send the file to specific client.

- "PrivateChat": make server determine the client that the other client want to connect with it private.

Keys in client:

- "chat": make client display the message to user (used in public chat)

- "list": when client receive "list", that's mean the new client is connected or some client is disconnected so the client update the list of online users.(used in public chat)

- "done": clear the arraylist of online user to be ready for new list of online user when new user is connect or disconnect.(used in public chat)

- "PrivateChat": make client open private window to create new private connection.(used in public chat)

- "ChangeUsername": make client close connection with server to change username because this username is used by other client.(used in public chat)

- "File": to receive file. (used in public and private chat)

- "ppchat": make client display the message to user (used in private chat)

- "disconnect":make client disconnect when the other client is disconnect in private chat
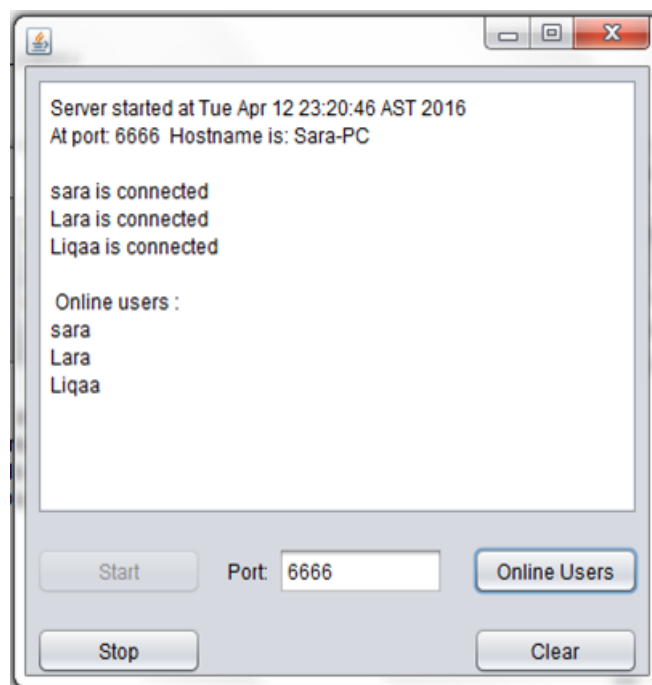
V. Results

Fig.5: a running Server interface

As show in Figure5 when click on start button the server displays the date, time, port number and hostname of server to allow client's connected to it. The start button became disabling and other button became enable, the server display any new client connected and disconnected and by click on online users button the server display all online users connected to server.
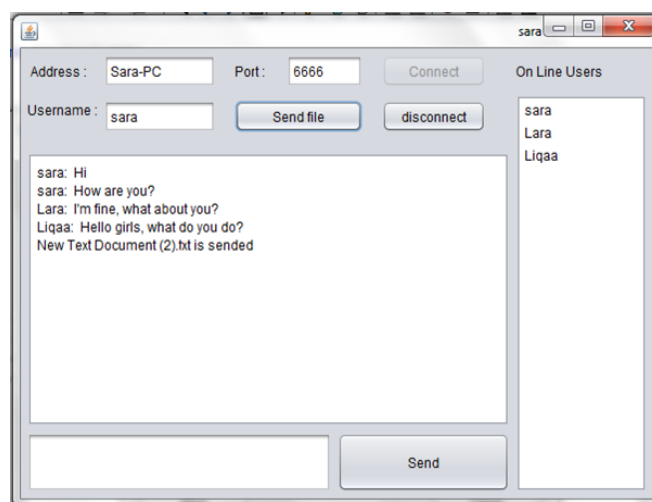

Fig.6: a running client interface

Figure6 shows a client user name "sara" connected to server with a host name "Sara-pc" and a port number 6666. All connected users display in online list.
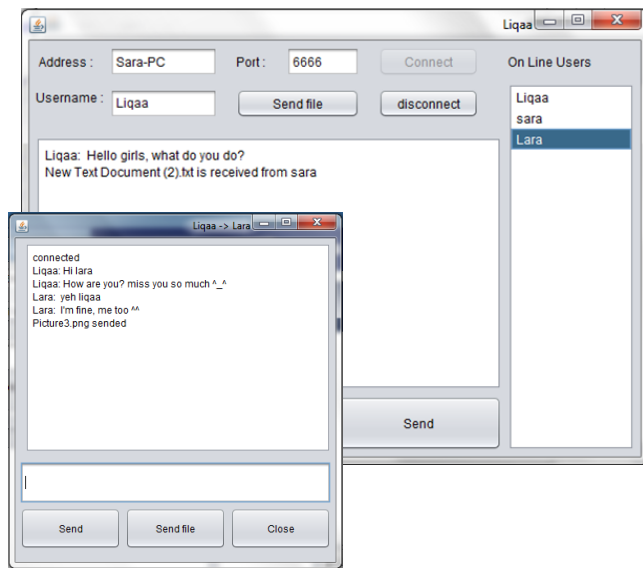
Fig.7: a running client private chat interface

When click on any online users the private window will be open to allow private chat between both sides as show in Figure7.

VI - CONCLUSION

The proposed protocol solved the communication problem among staff members and reduces time of communication with no cost. The LAN Chat Messenger supports many chat types

like (private chat with more than one online user) and (public chat which enable a group of online user to chat together) without internet connection. This LAN Messenger allows files exchange as well in the same local connection.

Acknowledgement

REFERENCES

[1] A. Odlyzko, "The history of communications and its implications for the Internet," *Soc. Sci. Res. Netw.*, vol. 6, pp. 1–28, 2000.

[2] Handel, M and Herbsleb, J.D, "What Is Chat Doing in the Workplace?", (2002)ACM Publishers, New York.

[3] R.B. Jennings, E.M. Nahum, D.P. Olshefski, "A study of Internet instant messaging and chat protocols", IEEE Network Magazine, Vol.20, PP16-21,2006.

[4] Tommy Mattsson ,"A Peer-To-Peer based chat system", thesis UPPSALA University- Department of Information Technology.,2012.

[5] Hamdan O. Alanazi, Rafidah Md Noor ,"The Guide of Implementing Chat Protocol: Study Case on Using the Socket Programming Concept and Object Oriented Programming", JOURNAL OF COMPUTING, VOL. 2, 2010.

[6] A. A. Sawant and B. B. Meshram, "Network programming in Java using Socket," vol. 3, no. 1, pp. 1299–1305, 2013.

[7] Christopher Joseph, "Social Interaction in Second Language Chat Rooms", (City University Hong Kong) Jenks,2014.

[8] T. Müldner, "Analysis of Java Client / Server and Web Programming Tools for Development of Educational Systems.", WebNet 98 - World Conference on the WWW and Internet & Intranet, Orlando, Florida, USA, November 7-12, 1998.

[9] Network Protocols Handbook 2nd Edition. 2004 - 2005 Javvin Technologies Inc.