Multi-User Chat Application using Sockets in Java – Windows 32-bit Operating System

\*Name\*

School

Address

Phone

Email

***Abstract*— Teleconferencing, is being used to connect ideas or people together around the world. This paper describes the implementation of a multiuser chat application using sockets in Java programming. The chat application is based on client-server model.**

***Keywords—sockets, chat application, multiuser, text, audio***

# Introduction

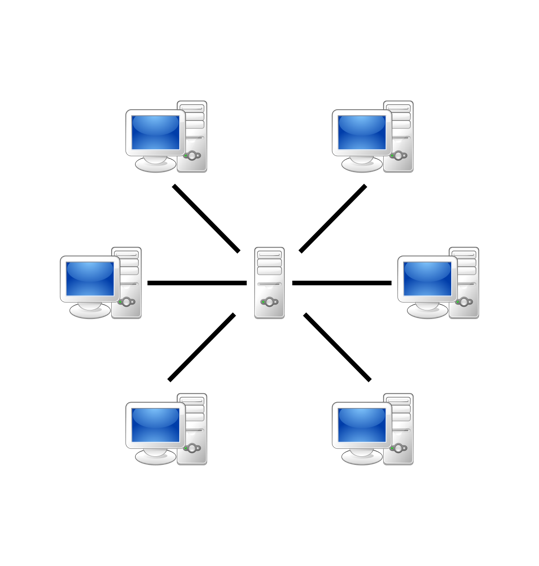


Figure 1: Client-Server Map

Networking is a major branch of programming that is important to connecting users through devices. As such, many programming languages have multiple ways to form connections, that is users and servers or between peers. The multiuser chat application is an application through which the user can communicate with others connected in the same network area. This chat application was developed to only work in Windows O/S 32 bit and is programmed in Java. Transmission Control Protocol, also known as TCP, is a standard that defines how to establish and maintain a network conversation between two or more computers exchanging data. TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other [1].

Figure 1 shown above demonstrates show clients are all connected to one main server computer that is having a public IP address on the server side. A client is a computer that requests services from a server computer. The clients however do not know each other’s IP addresses but just the server must have a public IP address for the server app to run. A computer that provides services to clients and controls access to hardware software and other resources is called a server. In this paper, it will mainly focus on how the application works and the tools which are used and why.

# Literature review

This research study is supported by concepts and principles used in socket programming in a distributed computing environment.

## Client Server Communication

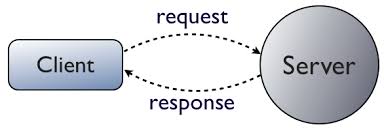


Figure 2: Client Server Model

Figure 1 shown above describes the use of client server model based on its request – reply protocol. In this model, the client directly sends request to the server asking for an appropriate service, and later server does the work and returns immediately the data or error code as a response back to the client. The server offers various services to the clients based on client’s request.

## Sockets and Socket Programming

In distributed computing environment, client server application (consist of client program and server program) can be designed using socket programming. Figure 2 shown below shows a Socket which is a two-way communication link between client and server programs that are running in a network environment [2]. It has an effective communication mechanism between two computers.

In Java programming, socket classes are used to represent connections between the client and the server. Java provides two classes for this purpose: Socket and the ServerSocket. These sockets connect two programs and implement two sides of the program (client and server).

The client server program allows the user to run the client software to create a query [3]. The client connects and sends the query to the identified server through the use of socket (client side). The server once received the request from the client analyzes the query and sends the result directly to the client.

## Port

In computer networking, a port represents a communication channel or endpoint. A port can be either physical or virtual. In Figure 3 shown below shows a computer to computer connection identifies the IP addresses and ports.

## Communication Protocol

There are two communication protocols that are commonly used in socket programming, the datagram communication and stream communication. The datagram communication protocol is popularly known as UDP (user datagram protocol) is considered a connectionless protocol. This means that every time the client sends datagram, it also requires them to send the local socket descriptor as well as the receiving socket’s address [4]. However, the stream communication protocol, popularly known as TCP (transfer control protocol) is a connection-oriented protocol [5]. This means that in order to establish communication over this type of protocol, a connection must be established first between two identified sockets. One socket listens for a connection request (main responsibility of the server) and the other socket asks for a connection (main responsibility of the client) in order to establish connections. Once they have already established the connection, it can be used to transmit data in both directions.

# Methodology

The chat application was designed using socket programming in Java Eclipse supported by Java Media Framework. The client-server-based application has many functions. The two programs (client and server) were simulated, demonstrated and analyzed.

## Implementation of Multi-user Chat Application

Through the development of a multi-user chat system, user can easily communicate with each other in an organization. Through the chat application, users can avail the facility of either private or group chat. This chat application also has ability to send over file while communicating through messages. Another objective achieved is the storage and archive of sent messages [6]. The design utilizes the concepts of object-oriented programming, and the architecture of C/S [7]. The Socket and the multi-thread technology are the core technology of the design. It implements the MVC design patterns. This design is realized through Eclipse platform [8].

## Process of Multi-user Chat Application

The server continuously accepts the request from the client server for a connection [9]. After every request is accepted, a new thread is assigned for communication. The name of the client is stored into a vector which keeps a track of devices with an established connection. The thread is stored with the correspondent request. This vector can determine the recipient through the helper class. The messages are delivered to the clients respectively as the streams are handled by the vector. The Start () method is invoked for the communication.  As the string is received to the handler, it is broken into two parts; message and the information of recipient. The handler searches the name of the recipient stored as the vector through the list of clients having an established connection with the server. If the name of the recipient is stored in the list, the message is sent to the recipient with the name of the sender in its header.

## Implementation in Java

The feasibility and advantage of implementing the design in Java is that the sockets include built-in input streams and output streams. When the Java program is written to communicate over a network, it indicates that the coding is done at the application layer. Another advantage of using Java for this application is that it is specifically designed in a way that it can be used in Internet’s innovative environment. It can also be used to develop end to end applications. These applications can be run on a single computer as well as can be distributed among multiple clients and servers over a network.

## Implementation of Multi-user Chat Application in JAVA

## The multiuser java chat application is developed in Java while using a client server architecture, based on a layered approach [10]. This is developed and implemented in Java while using API’s of Java network and Java swing. The socket of server is developed through java multithread API’s, encryption API’s and java network API’s. The client is connected to server and then the server is responsible for sending messages to other clients which are connected to it. For secure transmission, the messages should be encrypted through one of the algorithms of public key infrastructure. Through jdbc API’s the messages are archived and retrieved when needed. All the information is stored and then retrieved back from data base when required, using the SQL statements.

## Application Concepts

The application concept of this project is to develop such a system which can facilitate communication among users working at same place or even far away. This application strives to provide the user private chat, group chat, share files while sending other messages. In addition to this, it also stores the communicated messages. It also allows to download an application which is installed on local system. The user can easily communicate in a safer way.

*a. JMF*

The Java Media Framework (JMF) is a Java library that enables audio, video and other time-based media to be added to Java applications and applets [11]. This optional package, which can capture, play, stream and transcode multiple media formats, extends the Java Platform, Standard Edition (Java SE) and allows development of cross-platform multimedia applications.

*b. JMF 2.1.1e*

JMF 2.1.1e is used to run JMF for Windows Operating System. JMF for Windows is an optimized implementation of JMK that contains class files and native libraries for running JMF players.

## Main User Interface

The main user interface in the multi-client chat application can be determined and designed by using the java development kit 8 as it provides an extended functionality of inheritance of interfaces while implementation, but the design of a completely original interface is yet not supported as it becomes a very arduous task to complete and the front-end becomes a very technically taxing job to do.

## Socket Programming Style

Socket programming is an approach in network implementation in which two nodes are connected so that they can communicate with each other. They, however do not carry an equal relationship. The relationship is known as a client/server relationship in which server becomes the listener while the client reaches out to the server for information. Such a communication infrastructure is used extensively nowadays in the field of software development. The term “socket” has been coined as one end-point of a two-way communication link between two programs running on the network. Generally, a socket is identified by a port number and an IP address. In a multi-client chat application, built-in protocols are generally used to establish connections but there are frameworks available which allow you to characterize sockets manually i.e. by defining their port numbers and IP addresses.

# Advantages and Disadvantages of using a multiuser chat application

*200 words*

*Advantages*

* Commercial value: A well-developed multi-client chat app that provides data encryption is very much in demand as new data leakages are being discovered in the already built infrastructures every other day.
* Users can personalize their profiles and make the app as per their own liking. They can even change the color of the app interface if they deem it necessary.
* The market for multi-chat applications is enormous and any slight popularity in the usage of an app can generate tremendous revenues.

A single server app, that does not cost a lot can provide a large number of services to clients.

*Disadvantages*

* Cyber-attacks: An unguarded server is a very highly prone to cyber-attacks such as distributed denial of service attack or a distributed denial of service attack. The biggest issue with such attacks is the fact that these attacks are very easy to generate but they are very expensive to stop. DOS mitigation software is in the market but it provides a very cost intensive solution to the problem especially if you are a new startup.
* Your whole app is based on a single server and if that one server malfunctions, your whole startup will come crashing down on your head. Cyber-security professionals hired for the protection of such a server can be very expensive.

# conclusion

In this research paper, we have discussed the concept, design and development of a multi-chat application using java. The java media framework provides for options to integrate data options such as photo, video etc. The java media framework is used in the backend of this multi-chat infrastructure. The overall interface is developed using the java development toolkit with the implementation style being that of socket programming so that this application can provide support to already existing protocols in the networking world. The data is accessed and sent using a TCP/IP model in which the role of client and server tends to switch multiple times during the route of communication. The functionality of two-way communication is provided using the already existing networking protocols in which data is the main object of interest.

# References

[1] M. J. Donahoo and K. L. Calvert, “The Pocket Guide to TCP/IP Sockets: C Version,” p. 25.

[2] M. Xue and C. Zhu, “The Socket Programming and Software Design for Communication Based on Client/Server,” in 2009 Pacific-Asia Conference on Circuits, Communications and Systems, 2009, pp. 775–777.

[3] D. R. Yee and D. B. Gaunce, “Object-relational query builder which determines existence of structures from information loaded from the server and cached locally on the client computing system,” US6119126A, 12-Sep-2000.

[4] E. Farchi, Y. Krasny, and Y. Nir, “Automatic simulation of network problems in UDP-based Java programs,” in 18th International Parallel and Distributed Processing Symposium, 2004. Proceedings., 2004, pp. 267-.

[5] K. L. Calvert and M. J. Donahoo, TCP/IP Sockets in Java: Practical Guide for Programmers. Morgan Kaufmann, 2011.

[6] “Design & Implement of Chat System Based on Java Socket--《Microprocessors》2010年05期.” [Online]. Available: http://en.cnki.com.cn/Article\_en/CJFDTotal-WCLJ201005019.htm. [Accessed: 17-Dec-2019].

[7] K. Lieberherr, I. Holland, and A. Riel, “Object-oriented Programming: An Objective Sense of Style,” in *Conference Proceedings on Object-oriented Programming Systems, Languages and* Applications, New York, NY, USA, 1988, pp. 323–334.

[8] “Design and Realization of Instant Messaging Software Based on Socket Technology--《Computer Development & Applications》2011年05期.” [Online]. Available: http://en.cnki.com.cn/Article\_en/CJFDTotal-DNKF201105023.htm. [Accessed: 17-Dec-2019].

[9] “International Journal of Scientific Research in Computer Science, Engineering and Information Technology,” p. 5, 2018.

[10] A. Sinha, “Client-server computing: time shared computing,” Communications of the ACM, 01-Jul-1992. [Online]. Available: https://link.galegroup.com/apps/doc/A12340664/AONE?sid=lms. [Accessed: 18-Dec-2019].

[11] S. Sullivan, L. Winzeler, D. Brown, and J. Deagen, Programming with the Java Media Framework. New York, NY, USA: John Wiley & Sons, Inc., 1998.