
A Novel Approach to Chat box using socket in java

Dr.S.Mohan kumar,

¹New Horizon College of Engineering, Research Centre,Bangalore,India.

1.Introduction

Communication is a mean for people to exchange message. It has started since the beginning of human creation. Distant communication began as early 18 century with the introduction of television, next telegraph and then telephone. Interestingly enough, telephone communication stands out as the fastest growing technology, from fixed line to mobile wireless to transfer the data. The materialization of computer network and telecommunication technology bears the same objective that is to allow people to communicate each other. Chatting is the best tools to bring people and their idea together. The technology was available for years but it was accepted was quit recent. This project is an example of a chat server. It is made up of application the client application the client application which runs on the users and server application which runs on any pc on the network.

It uses TCP socket communication .We have a server as well as a client. Both can be run in the same machine or different machines. If both are running in the machine, the address to be given at the client side is local host address. If both are running in different machines, then in the client side we need to specify the IP address of machine in which server application is running.

Chat box application in java using sockets, is a chat application build .this chat appliance which can handle one-to-one user. It is entirely based on java and consists of two division: client and server. It support for private message. Messages will be transmitted through socket. It uses TCP socket communication. This has a server as a client. Both can be run in the same or different machine, the address to be given at client side is local host address. If both are running in different machines, then in the client side we need to specify

the ip address of machine in which server application is running. Implementing a chat server application provider a good and easy design. It put into practice in java programming language.

TCP (Transmission Control Protocol) is a standard that defines how to establish and maintain a network conversation via which application programs can exchange data. TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other. Together, TCP and IP are the basic rules defining the Internet.

The Transmission Control Protocol provides a communication service at an intermediate level between an application program and the Internet Protocol. It provides host-to-host connectivity at the transport layer of the Internet model. An application does not need to know the particular mechanisms for sending data via a link to another host, such as the required IP fragmentation to accommodate the maximum transmission unit of the transmission medium. At the transport layer, TCP handles all handshaking and transmission details and presents an abstraction of the network connection to the application typically through a network socket interface.

At the lower levels of the protocol stack, due to network congestion, traffic load balancing, or unpredictable network behavior, IP packets may be lost, duplicated, or delivered out of order. TCP detects these problems, requests re-transmission of lost data, rearranges out-of-order data and even helps minimize network congestion to reduce the occurrence of the other problems. If the data still remains undelivered, the source is notified of this failure. Once the TCP receiver has reassembled the sequence of octets originally transmitted, it

passes them to the receiving application. Thus, TCP abstracts the application's communication from the underlying networking details.

TCP is used extensively by many internet applications, including the World Wide Web, email, File Transfer Protocol, Secure Shell, peer-to-peer file sharing, and streaming media.

A socket is one endpoint of a two-way communication link between two programs running on the network. Socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent.

PURPOSE OF STUDY

This chat application was switched a free to paid services to avoid the cost that is primary cost. It is a platform where the chats of the user can be secured. This application allows the sending text message.

- Just click a button, message can reach any location by matching their IP address.
- Send unlimited messages.
- Send the messages very easily.
- It has desktop service.
- Saves money and time.
- Now- a-days almost all of us have their own pc. So they can use this application and start the conversation.
- They are also there to improve our English language as we read from this project.

The users wants their meeting and other stuffs should be in the control so that no other can access it. In that case this application helps.

Just log in into it and just giving an IP address to the person and connect it.

- Communication enhancement :
 - The application not only comes with text feature but it also offers a popular “hold to talk” that makes the messenger a good communicator.
 - It provide end-to-end encryption.
 - The messages is encrypted between the user and server of the application. When end-to-end encryption is used ,the messages is scrambled from the time it leaves the sender's

device to the it arrives at the recipient's, and only the recipient device possesses the key necessary to un-scramble the message.

We are using PC in our day in and day out, this can be used. To develop an instant messaging solution to enable users to seamlessly communicate with each other.

These are the learning media to the user. If any issues occur then they can approach to the other person such as colleagues and solve the problem. The above given points are which make to study this project and create a program as to learn the language and improve the language skill. And even the ideas can be shared through the application. It provides the end to end encryption.

1.2 PROBLEM STATEMENT

This project is to create a chat box application using java programming language using sockets with server and user or client to enable the users to chat with each others.

To develop an instant messaging solution to enable users to effortlessly communication with each other.

1.3 MOTIVATION

Traditionally, when people need to communicate with others they will have a face to face conversation to be delivered the message, same goes to the education field. To learn and experience developing a chatting application in pc targeting a small set of users. Obtaining knowledge to develop any similar kind of application. This project is to create a chat application with a server and clients to enable the clients to chat with many other clients.

This project can play an main and significant role in organizational field where host and guest can connect together through the IP address and port number. Existing system requires internet connection; whereas in the proposed system only intranet connection i.e. only a port number connection required. This system is useful for those who cannot afford to have an internet connection. The idea was to develop a simple to use, free chat system with translation capability of sent and received message. This was to be achieved by developing a chat system that used a Peer-To-Peer communication method

in order to be server-less and use some free translation service.

1.4 METHODOLOGY

In this project, sockets, TCP and port number are the methods used to execute the program.

Sockets: A very basic one-way Client and Server setup where a Client connects sends messages to server and the server shows them using socket connection. There's a lot of low-level stuff that needs to happen for these things to work but the Java API networking package takes care of all of that, making network programming very easy for programmers.

Establish a Socket Connection

- First argument – IP address of Server. (127.0.0.1 is the IP address of localhost, where code will run on single stand-alone machine).
- Second argument – TCP Port. (Just a number representing which application to run on a server. For example, HTTP runs on port 80. Port number can be from 0 to 65535)

Communication - To communicate over a socket connection, streams are used to both input and output the data.

TCP port - TCP stands for Transmission Control Protocol, which allows for reliable communication between two applications. TCP is typically used over the Internet Protocol, which is referred to as TCP/IP.

IP Address - Devices connected to the Internet are called nodes. Nodes that are computers are called hosts. Each node or host is identified by at least one unique number called an Internet address or an IP address. Most current IP addresses are 4-byte-long IPv4 addresses. However, a small but growing number of IP addresses are 16-byte-long IPv6 addresses. (4 and 6 refer to the version of the Internet Protocol, not the number of the bytes in the address.) Both IPv4 and IPv6 addresses are ordered sequences of bytes, like an array. They aren't numbers, and they aren't ordered in any predictable or useful sense.

2. System Requirement

2.1 Hardware and software requirement

Processor	-Intel Core i5
Speed	-1.8GH
Ram	-256MB(min)

Hard disk -10GB

Software system configuration

Operating system -windows 10

Programming language -java

Compiler -java compiler (javac).

2.2 ABOUT THE LANGUAGE

Java is a programming language created by James Gosling from Sun Microsystems (Sun) in 1991. The target of Java is to write a program once and then run this program on multiple operating systems. The first publicly available version of Java (Java 1.0) was released in 1995. Sun Microsystems was acquired by the Oracle Corporation in 2010. Oracle has now the steersmanship for Java. In 2006 Sun started to make Java available under the GNU General Public License (GPL). Oracle continues this project called OpenJDK.

Over time new enhanced versions of Java have been released. The current version of Java is Java 1.8 which is also known as Java 8.

Java is defined by a specification and consists of a programming language, a compiler, core libraries and a runtime (Java virtual machine) The Java runtime allows software developers to write program code in other languages than the Java programming language which still runs on the Java virtual machine. The Java platform is usually associated with the Java virtual machine and the Java core libraries.

The Java language was designed with the following properties:

- Platform independent: Java programs use the Java virtual machine as abstraction and do not access the operating system directly. This makes Java programs highly portable. A Java program (which is standard-compliant and follows certain rules) can run unmodified on all supported platforms, e.g., Windows or Linux.
- Object-orientated programming language: Except the primitive data types, all elements in Java are objects.
- The two-part Introduction to Java programming tutorial is meant for software developers who are new

to Java technology. Work through both parts to get up and running with object-oriented programming (OOP) and real-world application development using the Java language and platform.

- This first part is a step-by-step introduction to OOP using the Java language. The tutorial begins with an overview of the Java platform and language, followed by instructions for setting up a development environment consisting of a Java Development Kit (JDK) and the Eclipse IDE.

2.2.1 Interpreted and compiled language

Java source code is transferred into the bytecode format which does not depend on the target platform. These bytecode instructions will be interpreted by the Java Virtual machine (JVM). The JVM contains a so called Hotspot-Compiler which translates performance critical bytecode instructions into native code instructions.

- Automatic memory management: Java manages the memory allocation and de-allocation for creating new objects. The program does not have direct access to the memory. The so-called garbage collector automatically deletes objects to which no active pointer exists.
- Secure – with java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- Portable – Being architecture-neutral and having no implementation dependent aspects of the specification makes the compiled code executable on many processors, with the presence of java runtime system.
- Portable – Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- High performance – with the use of Just-In-Time compilers, java enables high performance.
- Dynamic – Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving

environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

2.2.2 Popular java editors

- To write your java programs, you will need a text editor. There are even more sophisticated IDEs available in the market. But for now, following-
- Notepad – on windows machine, we can use any simple text editor like Notepad, TextPad.
- Netbeans – A Java IDE that is open-source and free which can be downloaded from online.
- Eclipse – A java IDE developed by the eclipse open-source community and can be downloaded from online.

When we consider a java program, it can be defined as a collection of objects that communicate via invoking each other's methods.

- Object – Object have states and behaviors'. An object is an instance of a class.
- .Class – A class can be defined as a template/blueprint that describes the behavior/state that the object of its type supports.
- Methods – A method is basically a behavior. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the action are executed.

It is very important to keep in mind the following points.

- Case sensitivity – java is case sensitive, which means identifier would have different meaning in java.
- Class names – for all class names the first letter should be in upper case. If several words are used to form a name of the class, each inner word's first letter should be in upper case.
- Method names – All methods names should start with a lower case letter. If several words are used to form the name of the method, then each inner

word's first letter should be in upper case.

- Program file name - Name of the program file should exactly match the class name. when the file name and the class name do not match, your program will not compile

2.2.3 Instance Variable

Instant variable are declared in a class, but outside a method, constructor or any block Access modifiers can be given for instance variable. Instance variables can be declared in class level before or after use. When a space is allocated for an object in a heap, a slot for each instance variable valye is created.

The instance variable are visible for all methods, constructors and block in the class. Normally, it is recommended to make these variables private. However, visibility for subclass can be given for these variable with the use of access modifiers.

2.2.4 Access control modifiers

Java provides a number of access modifiers to set access levels for classes, variables, constructors and methods. As they follows

- Visible to the package, the default.
- No modifiers are needed.
- Visible to the class only.
- Visible to world.
- Visible to the package and all sub classes.

2.2.5 Java Classes

Java is an object-oriented programming language.

Everything in Java is associated with classes and objects, along with its attributes and methods. For example: in real life, a car is an object. The car has attributes, such as weight and color, and methods, such as drive and brake.

A Class is like an object constructor, or a "blueprint" for creating objects.

2.2.6 Abstraction in java

Abstraction is selecting data from a larger pool to show only the relevant details to the object. It helps to reduce programming complexity and effort. In Java, abstraction is accomplished using Abstract classes and interfaces. It is one

of the most important concepts of OOPs.

2.2.7 Abstract Class

A class which is declared "abstract" is called as an abstract class. It can have abstract methods as well as concrete methods. A normal class cannot have abstract methods.

2.2.8 Abstract Method

A method without a body is known as an Abstract Method. It must be declared in an abstract class. The abstract method will never be final because the abstract class must implement all the abstract methods.

- Java Class is an entity that determines how an object will behave and what the object will contain
- A Java object is a self-contained component which consists of methods and properties to make certain type of data useful
- A class system allows the program to define a new class (derived class) in terms of an existing class (super class) by using a technique like inheritance, overriding and augmenting.

2.2.9 Java Applet

Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.

There are many advantages of applet. They are as follows:

- It works at client side so less response time.
- Secured
- It can be executed by browsers running under many platforms, including Linux, Windows, Mac Os etc.

3. SYSTEM DESIGN

3.1 ARCHITECTURE

3.1 ARCHITECTURE

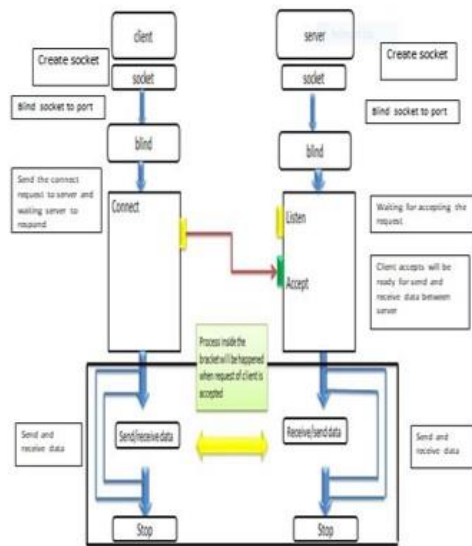


Fig 3.1 architecture of chat box application.

3.2 ALGORITHM

Create a GUI chat panel with below elements

- >> IP Address
- >> Port
- >> Guest/Host
- >> Connect/Disconnect
- >> Text chat area
- >> Connection Status bar

Start host chat canvas and provide below inputs

Host Inputs :

- >> IP Address
- >> Port
- >> Host

Start guest chat canvas and provide below inputs

Guest Inputs :

- >> IP Address(host IP address)
- >> Port(should be same as host port)
- >> Guest

On clicking Connect button on Host machine, Host machine opens a socket connection on the given port number and waits for client.

On clicking Connect button on Guest machine, Guest machine search for open(available) Host socket connection on the given port number.

If IP address and Port number matches

>> Connection between Host and Guest will be established

>> Status bar will display 'Connected' with Green color indication

else

>> Connection between Host and Guest will fail

>> Status bar will display 'Error! Could not connect!' information message

On successful connection,

>> Host/Guest will be able to communicate each other

>> Messages will be separated with OUTGOING(while sending), INCOMING(while receiving) indicators

On completion of communication Host/Guest can click on Disconnect button to terminate

>> Status bar will display 'Disconnected' with Red color indication

3.3 CODE

```

import java.lang.*;
import java.util.*;
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.net.*;

/*
created by Tousif Khan <tousifkhan510@gmail.com>
visit www.techzoo.org
*/

//JChatBox

public class JChatBox implements Runnable {

    // Connect status constants
    public final static int NULL = 0;

```

```

public final static int DISCONNECTED = 1;
public final static int DISCONNECTING = 2;
public final static int BEGIN_CONNECT = 3;
public final static int CONNECTED = 4;

// Other constants
public final static String statusMessages[] = {
    " Error! Could not connect!", " Disconnected",
    " Disconnecting...", " Connecting...", " Connected"
};

public final static JChatBox jchat = new JChatBox();

public final static String END_CHAT_SESSION =
    new Character((char)0).toString(); // Indicates the end
of a session

// Connection atate info
public static String hostIP = "localhost";
public static int port = 1234;
public static int connectionStatus = DISCONNECTED;
public static boolean isHost = true;
public static String statusString =
statusMessages[connectionStatus];
public static StringBuffer toAppend = new
StringBuffer("");
public static StringBuffer toSend = new StringBuffer("");

// Various GUI components and info
public static JFrame mainFrame = null;
public static JTextArea chatText = null;
public static JTextField chatLine = null;
public static JPanel statusBar = null;
public static JLabel statusField = null;
public static JTextField statusColor = null;
public static JTextField ipField = null;
public static JTextField portField = null;
public static JRadioButton hostOption = null;
public static JRadioButton guestOption = null;
public static JButton connectButton = null;

public static JButton disconnectButton = null;

// TCP Components
public static ServerSocket hostServer = null;
public static Socket socket = null;
public static BufferedReader in = null;
public static PrintWriter out = null;

////////////////////////////////////

private static JPanel initOptionsPane() {
    JPanel pane = null;
    ActionListener buttonListener = null;

    // Create an options pane
    JPanel optionsPane = new JPanel(new GridLayout(4,
1));

    // IP address input
    pane = new JPanel(new
FlowLayout(FlowLayout.RIGHT));
    pane.add(new JLabel("Host IP:"));
    ipField = new JTextField(10);
    ipField.setText(hostIP);
    ipField.setEnabled(false);
    ipField.addFocusListener(new FocusAdapter() {
        public void focusLost(FocusEvent e) {
            ipField.selectAll();
            // Should be editable only when disconnected
            if (connectionStatus != DISCONNECTED) {
                changeStatusNTS(NULL, true);
            }
            else {
                hostIP = ipField.getText();
            }
        }
    });
    pane.add(ipField);
    optionsPane.add(pane);

```

```
// Port input
pane = new JPanel(new
FlowLayout(FlowLayout.RIGHT));
pane.add(new JLabel("Port:"));
portField = new JTextField(10);
portField.setEditable(true);
portField.setText((new Integer(port)).toString());
portField.addFocusListener(new FocusAdapter() {
    public void focusLost(FocusEvent e) {
        // should be editable only when disconnected
        if (connectionStatus != DISCONNECTED) {
            changeStatusNTS(NULL, true);
        }
    }
});
```

4.RESULT AND DISCUSSION

4.1 SUMMARY OF RESULT OBTAINED



Fig4.1 Results

5.CONCLUSION

The main point of this project is to expand a secure chat application. Here, we had taken a socket and TCP and even the port number. The portability of the application have been completed by using the above given technique and file. The IP address are used to connect the two herald – it stand for internet protocol and it is responsible for delivery packets from host to the destination. The TCP is known to provide

reliable and error-free communication between end systems. It executes some sequential and segmentation of data. It also has acknowledgement feature and controls the flow of the data through flow of the data through flow control mechanism. It is a very effective protocol but has a lot of over-head due to such feature. Increased overhead leads to increased cost.

As a result, the product has been successfully developed in terms of expendability, portability, and maintainability and test in order to meet the basic needs.

6.REFERENCE

Text books

- Head First Java, 2nd edition by Kathy Sierra & Bert Bates
- A beginner's guide by Herbert Schildt.
- TCP/IP sockets in java – 2nd edition by Kenneth Calvert,

Michael Donahoo

Links:

http://people.cs.uchicago.edu/~sanket/chatserver/Chat_Server.pdf

<https://users.cs.duke.edu/~chase/cps196/slides/sockets.pdf>