Lab Assignment on Unit V: (Use JAVA/PYTHON)

Aim: Write a program using UDP sockets for wired network to implement
a. Peer to Peer Chat
b. Multiuser Chat
Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

Requirements: Wireshark Packet Analyzer Tool.

Theory:

Chat application in Java

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.

The **ChatSocketServer.java** is the server application.It simply creates a serverSocket on port 3339.Once a new connection comes , it accepts that connection and Socket object will be created for that connection.Now two threads will be created .One thread is for reading from the socket and the other is writing to socket.If the connection is terminated from client side , the server also exits.

ChatSocketServer.java

```
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.SocketException;
public class ChatSocketServer {
private ServerSocket severSocket = null;
private Socket socket = null;
private InputStream inStream = null;
private OutputStream outStream = null;

public ChatSocketServer() {

}

public void createSocket() {
try {
ServerSocket serverSocket = new ServerSocket(3339);
```

```
while (true) {
socket = serverSocket.accept();
inStream = socket.getInputStream();
outStream = socket.getOutputStream();
System.out.println("Connected");
createReadThread();
createWriteThread();


}
} catch (IOException io) {
io.printStackTrace();
}
}

public void createReadThread() {
Thread readThread = new Thread() {
public void run() {
while (socket.isConnected()) {
try {
byte[] readBuffer = new byte[200];
int num = inStream.read(readBuffer);
if (num > 0) {
byte[] arrayBytes = new byte[num];
System.arraycopy(readBuffer, 0, arrayBytes, 0, num);
String recvedMessage = new String(arrayBytes, "UTF-8");
System.out.println("Received message :" + recvedMessage);
} else {
notify();
}
;
//System.arraycopy();

} catch (SocketException se) {
System.exit(0);


} catch (IOException i) {
i.printStackTrace();
}


}
}
};
readThread.setPriority(Thread.MAX_PRIORITY);
```

```
readThread.start();
}

public void createWriteThread() {
Thread writeThread = new Thread() {
public void run() {

while (socket.isConnected()) {
try {
BufferedReader inputReader = new BufferedReader(new InputStreamReader(System.in));
sleep(100);
String typedMessage = inputReader.readLine();
if (typedMessage != null && typedMessage.length() > 0) {
synchronized (socket) {
outStream.write(typedMessage.getBytes("UTF-8"));
sleep(100);
}
}/* else {
notify();
}*/
;
//System.arraycopy();

} catch (IOException i) {
i.printStackTrace();
} catch (InterruptedException ie) {
ie.printStackTrace();
}

}
}
};
writeThread.setPriority(Thread.MAX_PRIORITY);
writeThread.start();

}

public static void main(String[] args) {
ChatSocketServer chatServer = new ChatSocketServer();
chatServer.createSocket();

}
}
```

The ChatSocketClient.java simply creates socket connection with the specified address on port 3339.Once a connection is established, two threads are creating.One for reading from the socket and other for writing to socket.Once the server disconnects the connection, the client exists itself.

ChatSocketClient.java

```
import java.io.*;
import java.net.Socket;
import java.net.SocketException;
import java.net.UnknownHostException;
public class ChatSocketClient {
private Socket socket = null;
private InputStream inStream = null;
private OutputStream outStream = null;

public ChatSocketClient() {

}

public void createSocket() {
try {
socket = new Socket("localHost", 3339);
System.out.println("Connected");
inStream = socket.getInputStream();
outStream = socket.getOutputStream();
createReadThread();
createWriteThread();
} catch (UnknownHostException u) {
u.printStackTrace();
} catch (IOException io) {
io.printStackTrace();
}
}

public void createReadThread() {
Thread readThread = new Thread() {
public void run() {
while (socket.isConnected()) {

try {
byte[] readBuffer = new byte[200];
int num = inStream.read(readBuffer);

if (num > 0) {
byte[] arrayBytes = new byte[num];
```

```
System.arraycopy(readBuffer, 0, arrayBytes, 0, num);
String recvedMessage = new String(arrayBytes, "UTF-8");
System.out.println("Received message :" + recvedMessage);
}/* else {
// notify();
}*/
;
//System.arraycopy();
}catch (SocketException se){
System.exit(0);

} catch (IOException i) {
i.printStackTrace();
}

}
}
};
readThread.setPriority(Thread.MAX_PRIORITY);
readThread.start();
}

public void createWriteThread() {
Thread writeThread = new Thread() {
public void run() {
while (socket.isConnected()) {

try {
BufferedReader inputReader = new BufferedReader(new InputStreamReader(System.in));
sleep(100);
String typedMessage = inputReader.readLine();
if (typedMessage != null && typedMessage.length() > 0) {
synchronized (socket) {
outStream.write(typedMessage.getBytes("UTF-8"));
sleep(100);
}
}
;
//System.arraycopy();

} catch (IOException i) {
i.printStackTrace();
} catch (InterruptedException ie) {
```

```
ie.printStackTrace();
}

}
}
};
writeThread.setPriority(Thread.MAX_PRIORITY);
writeThread.start();
}

public static void main(String[] args) throws Exception {
ChatSocketClient myChatClient = new ChatSocketClient();
myChatClient.createSocket();
/*myChatClient.createReadThread();
myChatClient.createWriteThread();*/
}
}
```

Output

Run the ChatSocketServer.java and then the ChatSocketClient.java.Once both hot connected, connected message will be displayed on the console.Now type messages on each console and press enter button.Messages will be transmitted throgh socket.

Output of ChatSocketServer.java

Connected

Received message :haai

I am server

who are u ?

Received message :i am client …How r u ?

Output of ChatSocketClient,java

Connected

haai

Received message :I am server

Received message :who are u ?

i am client …How r u ?