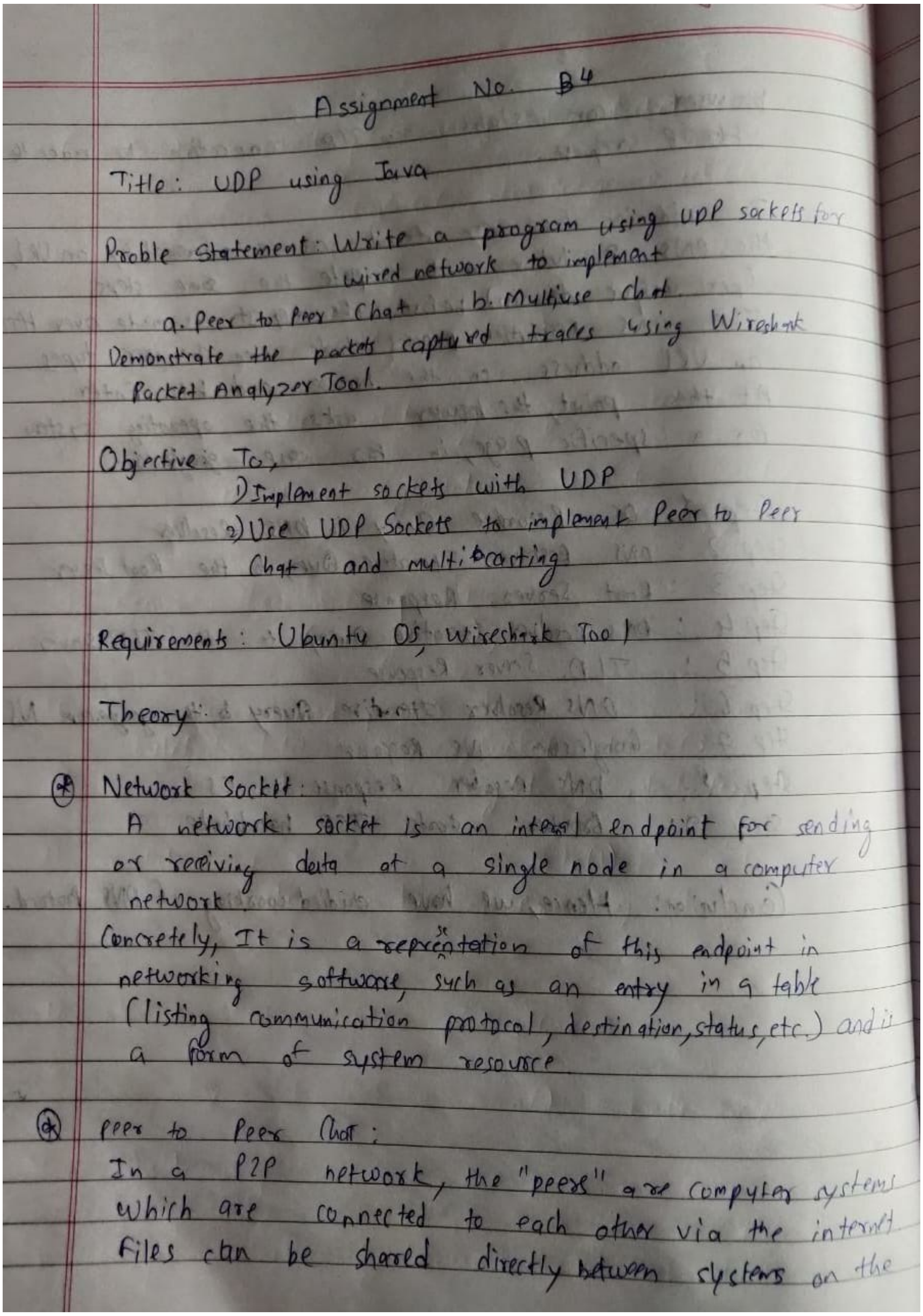
//31139- Durvesh- CNLB4 (UDP using Java)

Writeup



Assignment No. ~~B~~4

Title: UDP using Java

Proble Statement: Write a program using UDP sockets for wired network to implement
a. Peer to Peer Chat b. Multiuse Chat
Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool.

Objective: To,
1) Implement sockets with UDP
2) Use UDP Sockets to implement Peer to Peer Chat and multi~~b~~casting

Requirements: Ubuntu OS, Wireshark Tool

Theory:

⊛ Network Socket:
A network socket is an internal endpoint for sending or receiving data at a single node in a computer network.
Concretely, It is a representation of this endpoint in networking software, such as an entry in a table (listing communication protocol, destination, status, etc.) and is a form of system resource.

⊛ Peer to Peer Chat:
In a P2P network, the "peers" are computer systems which are connected to each other via the internet. Files can be shared directly between systems on the

network without the need of a central server. In other words, each computer on a P2P network becomes a file server as well as a client.

Steps for peer to peer UDP Socket Programming in Java.

1. Import following classes:
   java.io.BufferedReader
   java.io.InputStreamReader
   java.net.DatagramPacket
   java.net.DatagramSocket
   java.net.InetAddress.

2. Initialize the Port no. & Server_IP (localhost)

3. Create & initialize the BufferedReader & DatagramSocket objects.

```
DatagramSocket socket = new DatagramSocket()
BufferedReader br = new BufferedReader (
                    new InputStreamReader (System.in));
```

4. Create and initialize the inet Address object to store server's IP:

```
InetAddress server = InetAddress.getByName (SERVER_IP);
```

5. Read the message from console and store it in byte array

```
String s = br.readline() ;
byte[] sendMsg = s.getBytes();
```

6. Create a UDP packet using DatagramPacket class for sending the message:
DatagramPacket sendingPacket = new DatagramPacket(sendmsg, sndmsg.length, server, PORT)

7. Send the UDP Packet
socket.send(sending Packet)

8. Create a UDP Packet using DatagramPacket class for receiving message:
DatagramPacket receivingPacket = new DatagramPacket(replyMsg, replyMsg.length) ~~server, PORT~~

9. Receive the UDP packet:
socket.receive(receiving Packet);

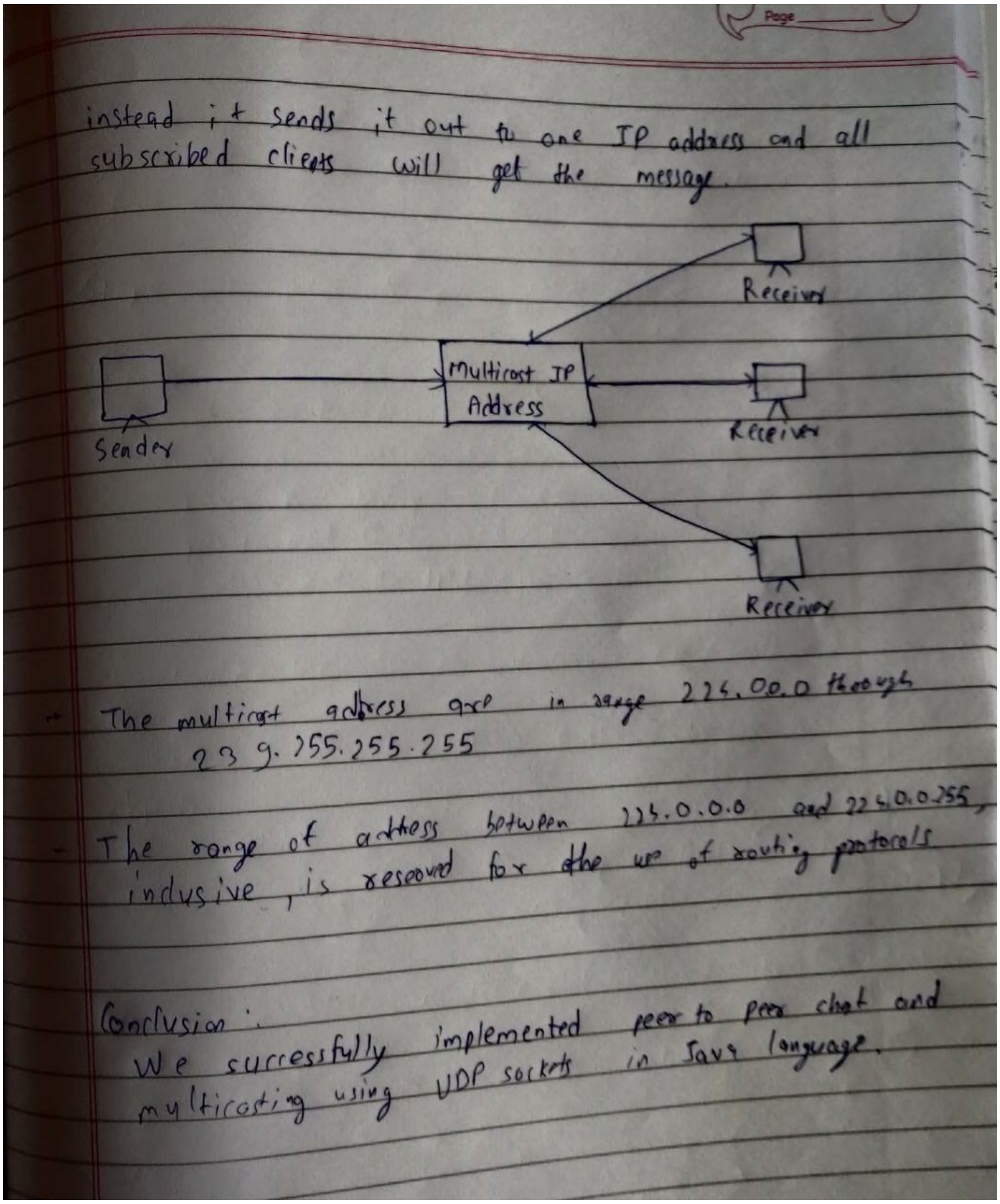
10. Store the received packets data into a byte array:
byte[] data = receivingPacket getData();
String S1 = new String(data, 0, data.length)

11. Include step 5 to 11 in while(1) for continous message exchange. Use "bye" as terminating string on both ends.

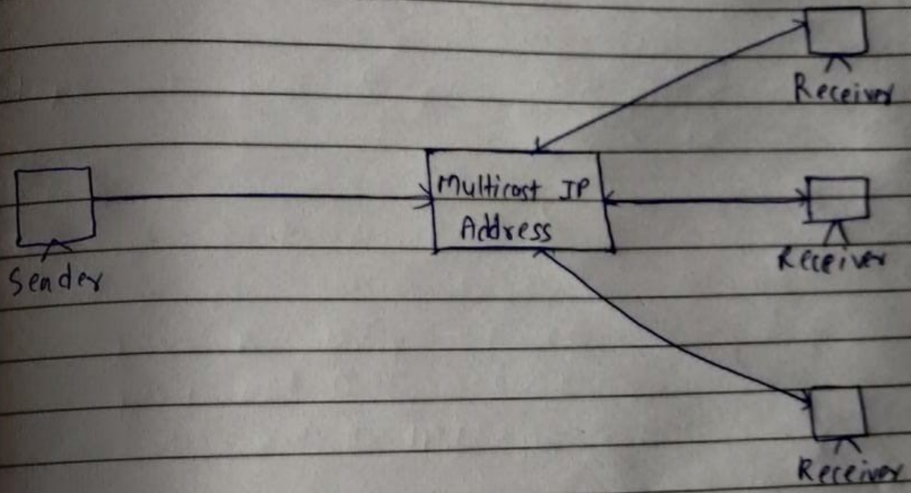
12. Close Client Socket: socket.close()

13. Use try-catch block in main() to handle Exceptions.

* Multicasting:
multicasting is ~~one~~ a type of Datagram Socket. Unlike regular Datagrams, multicasting doesn't handle each client individually



instead it sends it out to one IP address and all subscribed clients will get the message.



→ The multicast address are in range 224.0.0.0 through 239.255.255.255

- The range of address between 224.0.0.0 and 224.0.0.255, inclusive, is reserved for the use of routing protocols

Conclusion:
We successfully implemented peer to peer chat and multicasting using UDP sockets in Java language.

Code

------P2pclient.java

```
package udp;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
```

```java
import java.net.InetAddress;
import java.net.SocketException;
import java.nio.charset.StandardCharsets;

public class P2pclient {
    public static void main(String[] args) throws SocketException, IOException {

        BufferedReader clientRead =new BufferedReader(new
InputStreamReader(System.in));

        InetAddress IP = InetAddress.getByName("127.0.0.1");

        DatagramSocket clientSocket = new DatagramSocket();
        while(true)     //true
        {
            byte[] sendbuffer = new byte[1024];
            byte[] receivebuffer = new byte[1024];

            System.out.print("\n\nClient: ");
            String clientData = clientRead.readLine();
            sendbuffer = clientData.getBytes();
            DatagramPacket sendPacket =
                    new DatagramPacket(sendbuffer, sendbuffer.length, IP, 2604);
            clientSocket.send(sendPacket);
            if(clientData.equalsIgnoreCase("bye"))
            {
                System.out.println("\nConnection ended by client");
                break;
            }


            DatagramPacket receivePacket =
                    new DatagramPacket(receivebuffer, receivebuffer.length);
            clientSocket.receive(receivePacket);
            receivebuffer = receivePacket.getData();
            String serverData = new String(receivebuffer , StandardCharsets.UTF_8);

            System.out.print("\nServer: " + serverData);
            if(serverData.equals("bye"))
            {
                System.out.println("\n\nConnection ended by server...");
                break;
            }

        }
        clientSocket.close();
    }
}
```

-------P2pserver.java

```java
package udp;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.nio.charset.StandardCharsets;


public class P2pserver {


    public static void main(String[] args) throws SocketException, IOException {
        DatagramSocket serverSocket = new DatagramSocket(2604);

        while(true)
        {
            byte[] receivebuffer = new byte[1024];
            byte[] sendbuffer  = new byte[1024];
            DatagramPacket recvdpkt = new DatagramPacket(receivebuffer,
receivebuffer.length);
            serverSocket.receive(recvdpkt);
            receivebuffer = recvdpkt.getData();
            InetAddress IP = recvdpkt.getAddress();
            int portno = recvdpkt.getPort();
            String clientdata = new String(receivebuffer , StandardCharsets.UTF_8);
            if(clientdata.equals("bye"))
            {
                System.out.println("\n\nConnection ended by client...");
                break;
            }
            System.out.println("\nClient : "+ clientdata);
            System.out.print("\nServer : ");
            BufferedReader serverRead = new BufferedReader(new InputStreamReader
(System.in) );
            String serverdata = serverRead.readLine();

            sendbuffer = serverdata.getBytes();
            DatagramPacket sendPacket = new DatagramPacket(sendbuffer,
sendbuffer.length, IP,portno);
            serverSocket.send(sendPacket);

            if(serverdata.equalsIgnoreCase("bye"))
            {
                System.out.println("\nConnection ended by server...");
                break;
            }


        }
        serverSocket.close();
    }

}
```

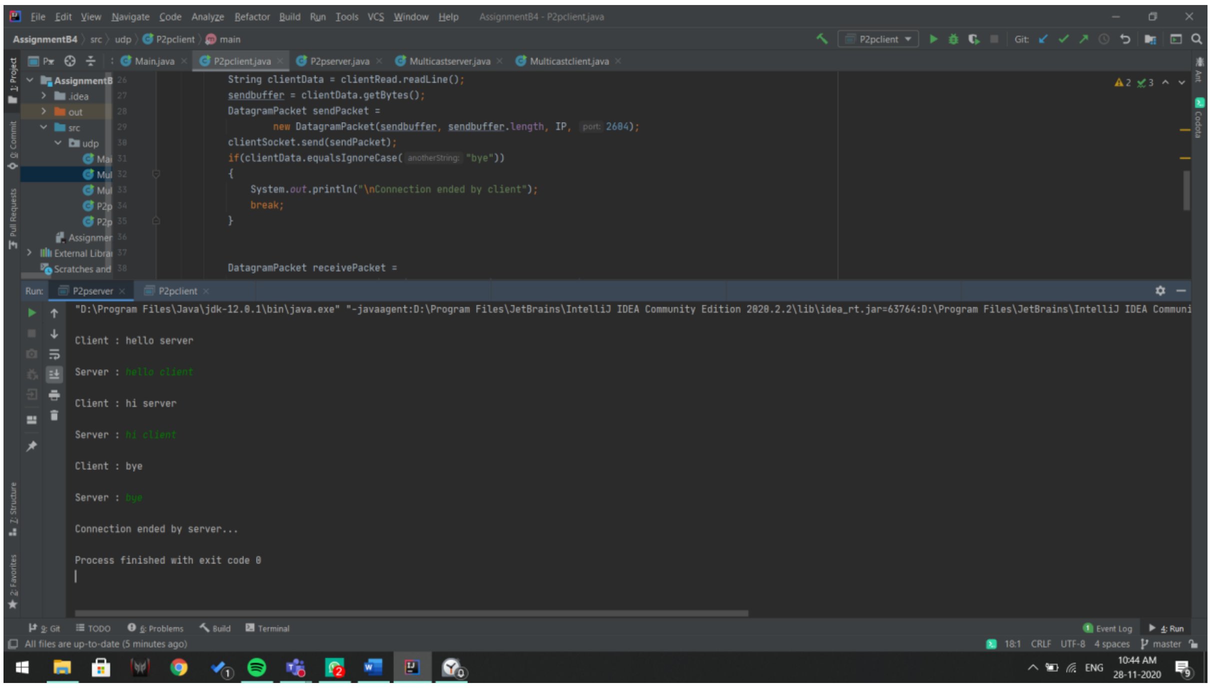
-------Multicastclient.java

```java
package udp;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.net.SocketException;
import java.nio.charset.StandardCharsets;
import java.net.*;
public class Multicastclient{
    public static void main(String args[]) {
        try{
            InetAddress group = InetAddress.getByName("225.4.5.6");
            MulticastSocket multiSocket = new MulticastSocket(2604) ;
            multiSocket.joinGroup(group);
            byte[] buffer = new byte[1024];
            DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
            while(true) {
                multiSocket.receive(packet);
                buffer = packet.getData();
                String data = new String(buffer , StandardCharsets.UTF_8);
                System.out.println("Server > "+ data);
                if(data.equals("exit")){
                    System.out.println("\nServer connection terminated...");
                    break;
                }
            }
            multiSocket.close();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

-------Multicastserver.java

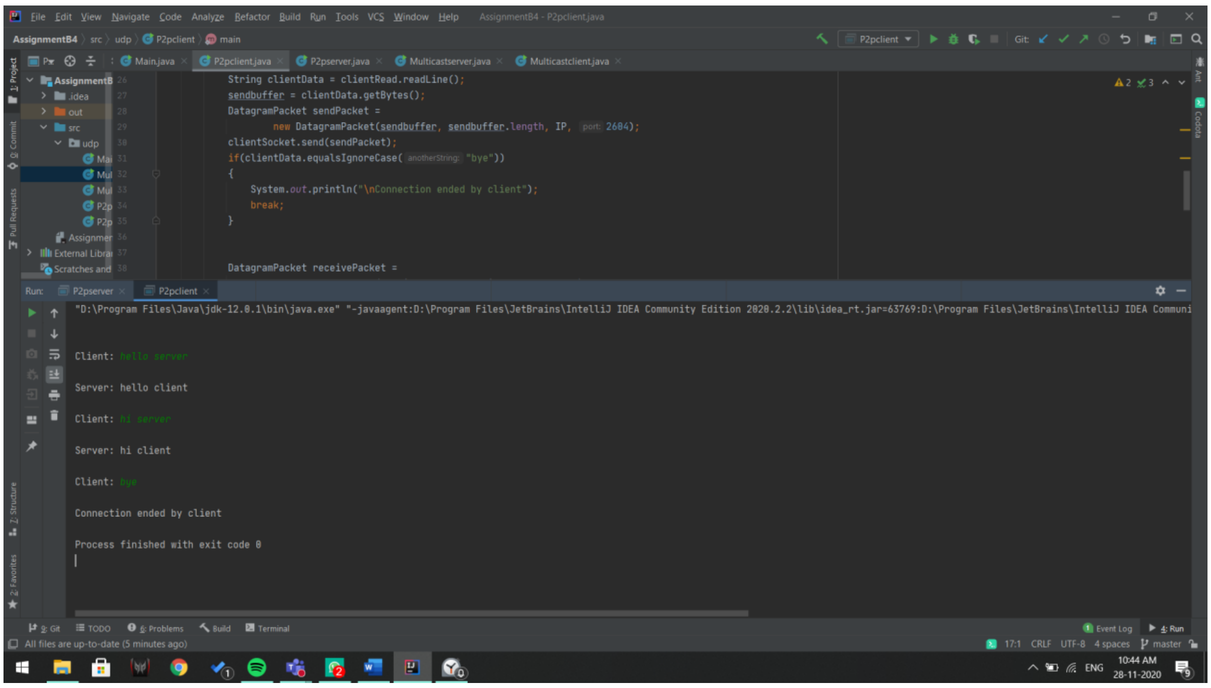
```
package udp;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.MulticastSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Scanner;
public class Multicastserver{
    public static void main(String args[]) {
        try {
            InetAddress group = InetAddress.getByName("225.4.5.6");
            MulticastSocket multiSocket= new MulticastSocket(2604) ;
            multiSocket.joinGroup(group);
            String data = "";
            BufferedReader ip =new BufferedReader(new
InputStreamReader(System.in));
            while(!data.equals("exit")) {
                System.out.println("Server > ");
                data = ip.readLine();
                DatagramPacket sendPacket = new
                        DatagramPacket(data.getBytes(),data.length(),group,2604);
                multiSocket.send(sendPacket);
            }
            multiSocket.close();
            System.out.println("\nSocket Closed...");
        }
        catch(Exception e){
            e.printStackTrace();
            System.out.println("ERROR");
        }
    }
}
```

Outputs



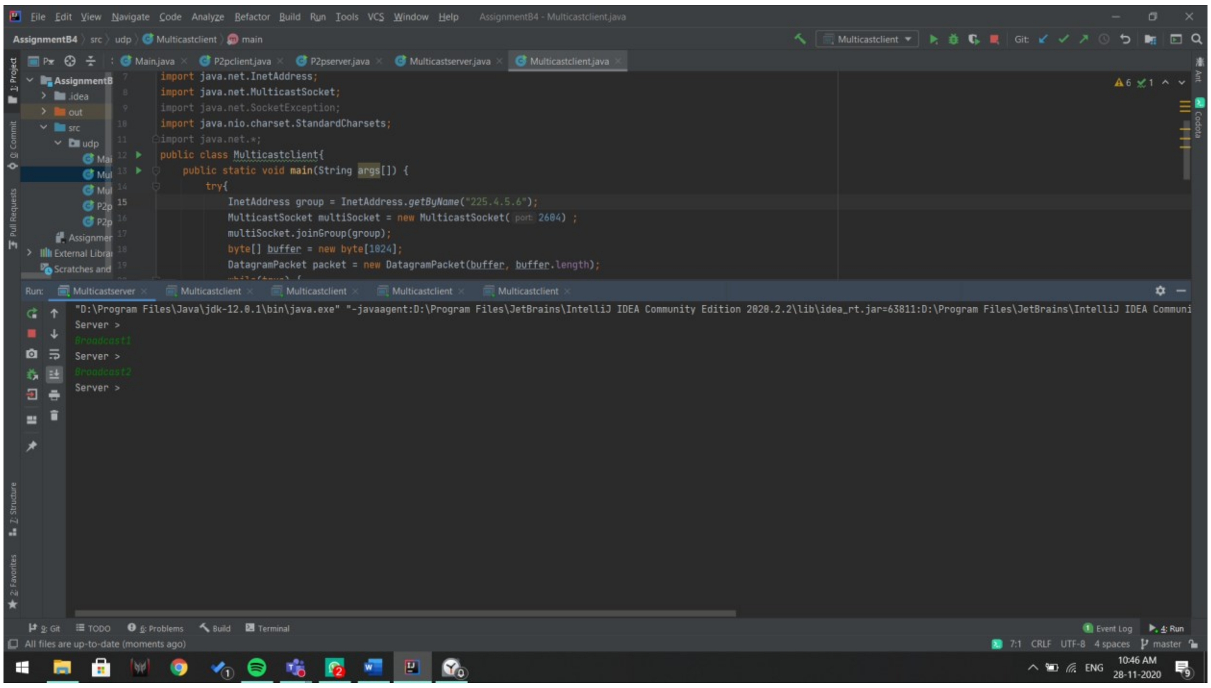
```
String clientData = clientRead.readLine();
sendbuffer = clientData.getBytes();
DatagramPacket sendPacket =
        new DatagramPacket(sendbuffer, sendbuffer.length, IP, port: 2604);
clientSocket.send(sendPacket);
if(clientData.equalsIgnoreCase( anotherString: "bye"))
{
    System.out.println("\nConnection ended by client");
    break;
}


DatagramPacket receivePacket =
```

```
"D:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:D:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=63764:D:\Program Files\JetBrains\IntelliJ IDEA Communi

Client : hello server

Server : hello client

Client : hi server

Server : hi client

Client : bye

Server : bye

Connection ended by server...

Process finished with exit code 0
```



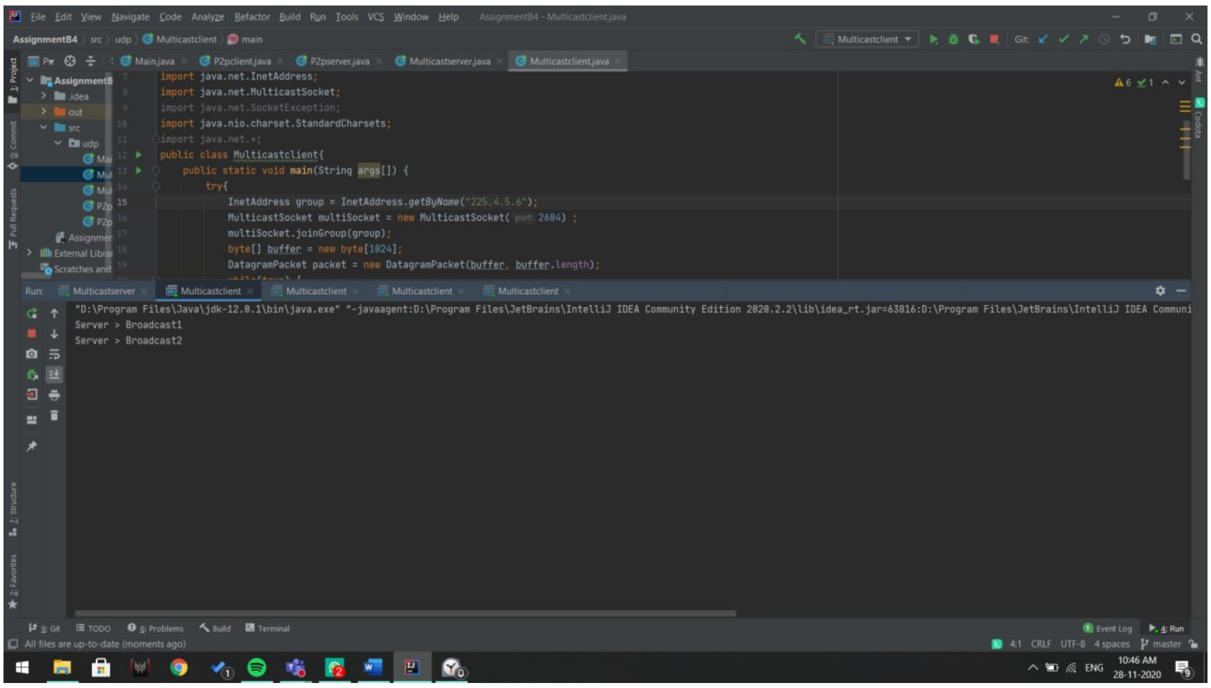
```
String clientData = clientRead.readLine();
sendbuffer = clientData.getBytes();
DatagramPacket sendPacket =
        new DatagramPacket(sendbuffer, sendbuffer.length, IP, port: 2604);
clientSocket.send(sendPacket);
if(clientData.equalsIgnoreCase( anotherString: "bye"))
{
    System.out.println("\nConnection ended by client");
    break;
}


DatagramPacket receivePacket =
```

```
"D:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:D:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=63769:D:\Program Files\JetBrains\IntelliJ IDEA Communi


Client: hello server

Server: hello client

Client: hi server

Server: hi client

Client: bye

Connection ended by client

Process finished with exit code 0
```



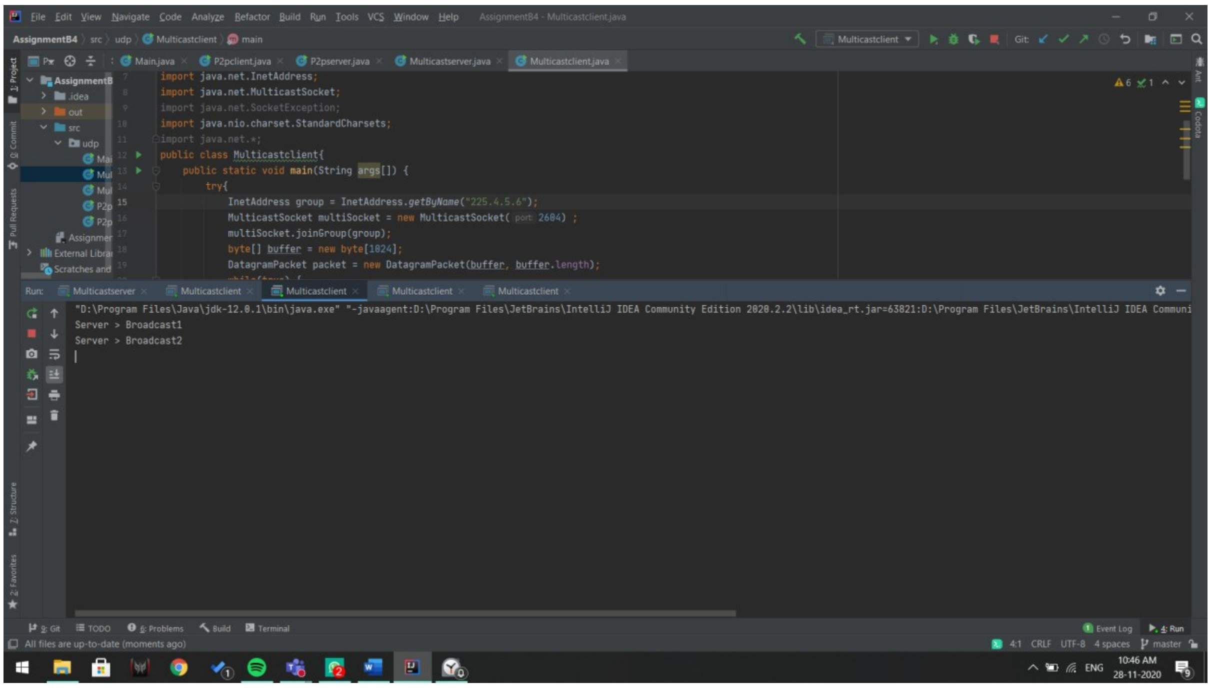
```java
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.net.SocketException;
import java.nio.charset.StandardCharsets;
import java.net.*;
public class Multicastclient{
    public static void main(String args[]) {
        try{
            InetAddress group = InetAddress.getByName("225.4.5.6");
            MulticastSocket multiSocket = new MulticastSocket( port: 2604) ;
            multiSocket.joinGroup(group);
            byte[] buffer = new byte[1024];
            DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
```

```
"D:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:D:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=63811:D:\Program Files\JetBrains\IntelliJ IDEA Communi
Server >
Broadcast1
Server >
Broadcast2
Server >
```



```java
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.net.SocketException;
import java.nio.charset.StandardCharsets;
import java.net.*;
public class Multicastclient{
    public static void main(String args[]) {
        try{
            InetAddress group = InetAddress.getByName("225.4.5.6");
            MulticastSocket multiSocket = new MulticastSocket( port: 2604) ;
            multiSocket.joinGroup(group);
            byte[] buffer = new byte[1024];
            DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
```

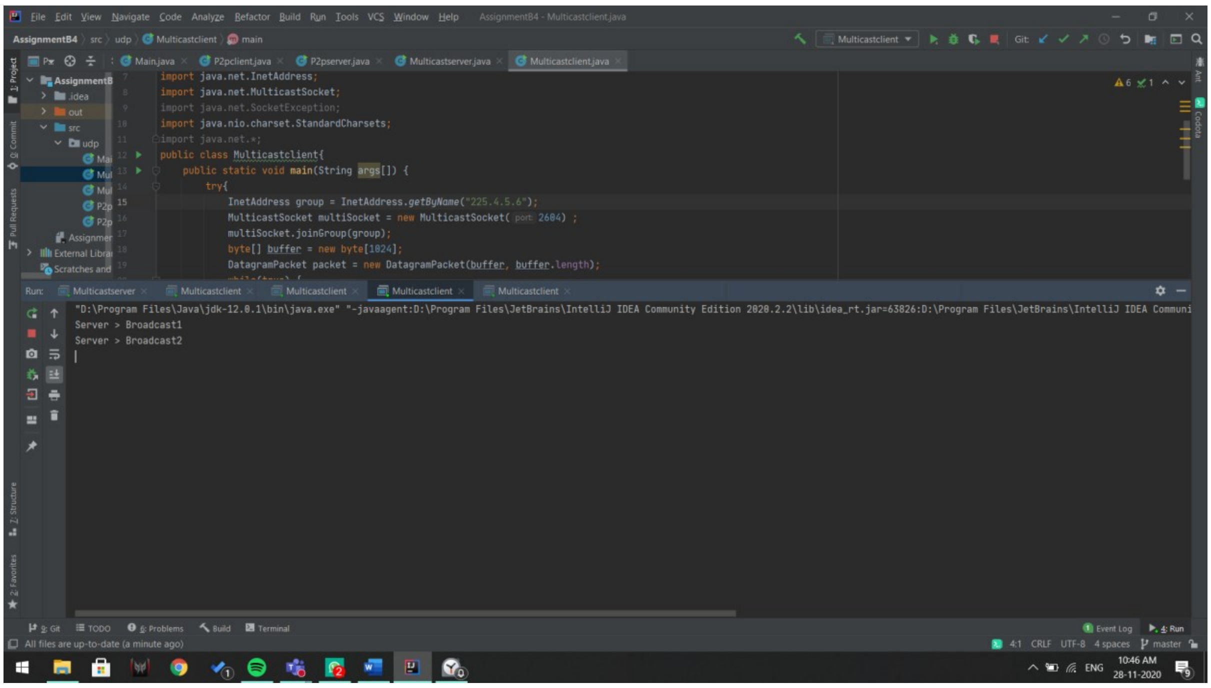
```
"D:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:D:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=63816:D:\Program Files\JetBrains\IntelliJ IDEA Communi
Server > Broadcast1
Server > Broadcast2
```


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
AssignmentB4 - Multicastclient.java
AssignmentB4 › src › udp › Multicastclient › main
Main.java
P2pclient.java
P2pserver.java
Multicastserver.java
Multicastclient.java
Run: Multicastserver Multicastclient Multicastclient Multicastclient Multicastclient
Git TODO Problems Build Terminal Event Log Run
10:46 AM 28-11-2020

```java
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.net.SocketException;
import java.nio.charset.StandardCharsets;
import java.net.*;
public class Multicastclient{
    public static void main(String args[]) {
        try{
            InetAddress group = InetAddress.getByName("225.4.5.6");
            MulticastSocket multiSocket = new MulticastSocket( port: 2604) ;
            multiSocket.joinGroup(group);
            byte[] buffer = new byte[1024];
            DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
```

```
"D:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:D:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=63821:D:\Program Files\JetBrains\IntelliJ IDEA Communi
Server > Broadcast1
Server > Broadcast2
```

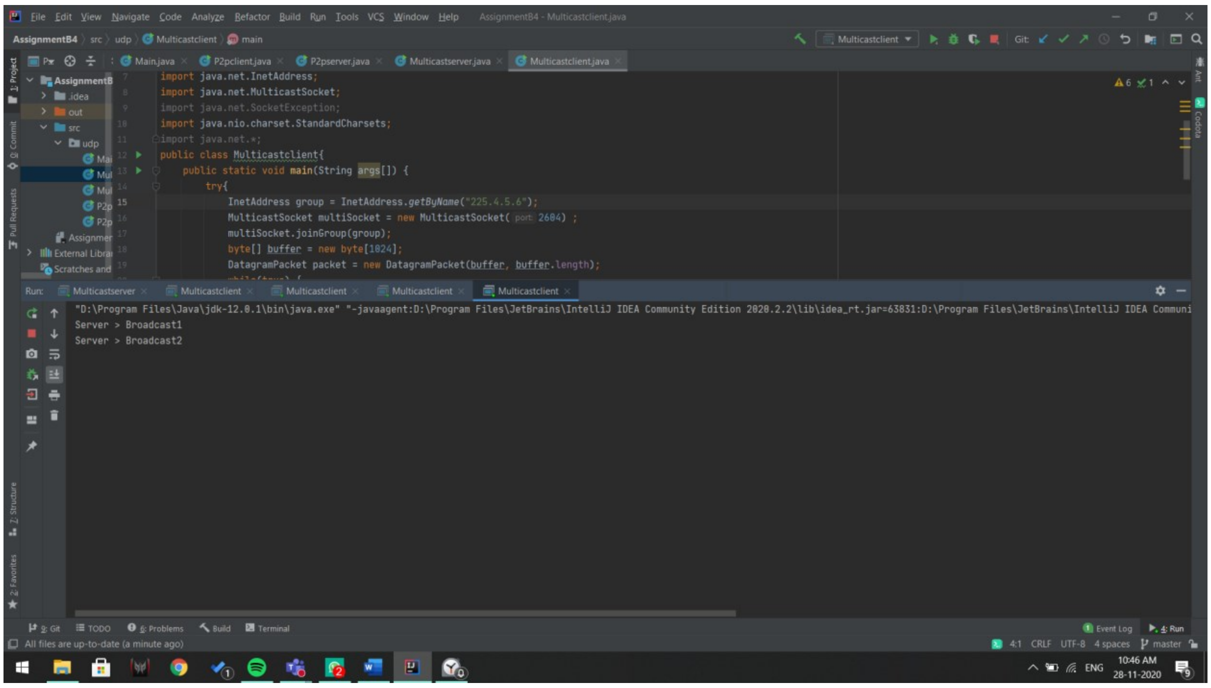
All files are up-to-date (moments ago)


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
AssignmentB4 - Multicastclient.java
AssignmentB4 › src › udp › Multicastclient › main
Main.java
P2pclient.java
P2pserver.java
Multicastserver.java
Multicastclient.java
Run: Multicastserver Multicastclient Multicastclient Multicastclient Multicastclient
Git TODO Problems Build Terminal Event Log Run
10:46 AM 28-11-2020

```java
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.net.SocketException;
import java.nio.charset.StandardCharsets;
import java.net.*;
public class Multicastclient{
    public static void main(String args[]) {
        try{
            InetAddress group = InetAddress.getByName("225.4.5.6");
            MulticastSocket multiSocket = new MulticastSocket( port: 2604) ;
            multiSocket.joinGroup(group);
            byte[] buffer = new byte[1024];
            DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
```

```
"D:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:D:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=63826:D:\Program Files\JetBrains\IntelliJ IDEA Communi
Server > Broadcast1
Server > Broadcast2
```

All files are up-to-date (a minute ago)



```java
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.net.SocketException;
import java.nio.charset.StandardCharsets;
import java.net.*;
public class Multicastclient{
    public static void main(String args[]) {
        try{
            InetAddress group = InetAddress.getByName("225.4.5.6");
            MulticastSocket multiSocket = new MulticastSocket( port: 2604) ;
            multiSocket.joinGroup(group);
            byte[] buffer = new byte[1024];
            DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
```

```
"D:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:D:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=63831:D:\Program Files\JetBrains\IntelliJ IDEA Communi
Server > Broadcast1
Server > Broadcast2
```