Name: Soham Desai
XIE ID: 202003021

Roll no:11
Batch: B

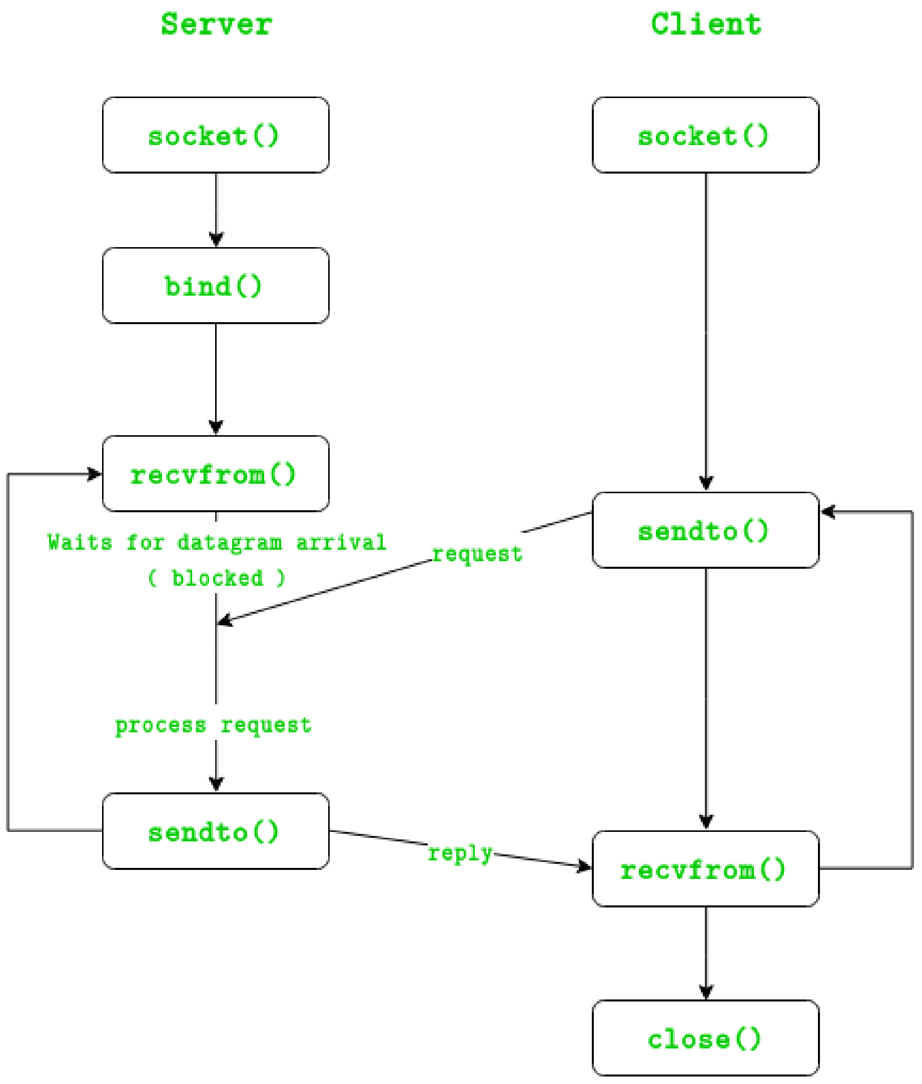
Experiment 7

Aim: Study and Implement Socket Programming using UDP

LO 4: Implement the socket programming for client server architecture.

Theory:

In UDP, the client does not form a connection with the server like in TCP and instead just sends a datagram. Similarly, the server need not accept a connection and just waits for datagrams to arrive. Datagrams upon arrival contain the address of the sender which the server uses to send data to the correct client.



The entire process can be broken down into the following steps :

Name: Soham Desai　　　　　　　　　　　　　　　　　　　　　　　　Roll no:11
XIE ID: 202003021　　　　　　　　　　　　　　　　　　　　　　　　Batch: B

UDP Server :

1. Create a UDP socket.
2. Bind the socket to the server address.
3. Wait until the datagram packet arrives from the client.
4. Process the datagram packet and send a reply to the client.
5. Go back to Step 3.

UDP Client :

1. Create a UDP socket.
2. Send a message to the server.
3. Wait until response from the server is received.
4. Process reply and go back to step 2, if necessary.
5. Close socket descriptor and exit.

Code :

1. Client :

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;
public class UDP_client {
 public static void main(String args[]) throws IOException {
Scanner sc = new Scanner(System.in);
 DatagramSocket ds = new DatagramSocket();
 InetAddress ip = InetAddress.getLocalHost();
 byte buf[] = null;
 while (true) {
     String inp = sc.nextLine();
     buf = inp.getBytes();
     DatagramPacket DpSend =
             new DatagramPacket(buf, buf.length, ip, 1234);
     ds.send(DpSend);
     if (inp.equals("over"))
               break;
                    }
                }
}
```

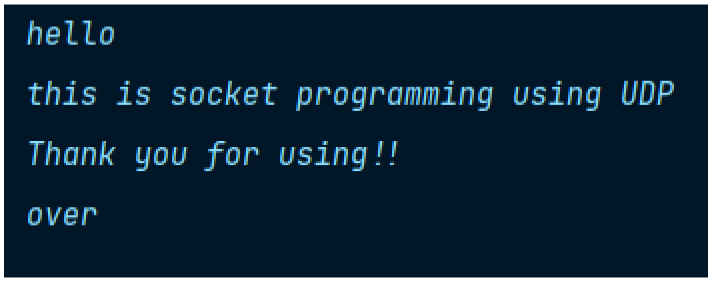
Name: Soham Desai   Roll no:11
XIE ID: 202003021   Batch: B

2. Server:

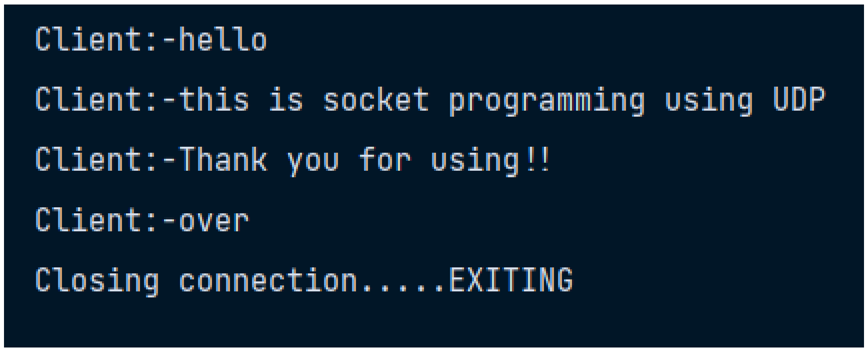
```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
public class UDP_server {
    public static void main(String[] args) throws IOException {
        DatagramSocket ds = new DatagramSocket(1234);
        byte[] receive = new byte[65535];
        DatagramPacket DpReceive = null;
        while (true) {
            DpReceive = new DatagramPacket(receive, receive.length);
            ds.receive(DpReceive);
            System.out.println("Client:-" + data(receive));
            if (data(receive).toString().equals("over")) {
                System.out.println("Closing connection.....EXITING");
                break;
            }
            receive = new byte[65535];
        }
    }
    public static StringBuilder data(byte[] a) {
        if (a == null)
            return null;
        StringBuilder ret = new StringBuilder();
        int i = 0;
        while (a[i] != 0) {
            ret.append((char) a[i]);
            i++;
        }
        return ret;
    }
}
```

Name: Soham Desai   Roll no:11
XIE ID: 202003021   Batch: B

Output:



```
hello
this is socket programming using UDP
Thank you for using!!
over
```



```
Client:-hello
Client:-this is socket programming using UDP
Client:-Thank you for using!!
Client:-over
Closing connection.....EXITING
```

Conclusion: From this experiment we have learned to do socket programming using UDP in Javaand also saw how it works.