Experiment 8: Write a program on Datagram Socket for the Client/Server to display the messages on the client side typed at the server-side

Aim

To write a Java program using datagram sockets for client-server communication, where the messages typed on the server side are displayed on the client side.

Background Theory

Java provides the DatagramSocket and DatagramPacket classes in the java.net package for UDP communication.

- DatagramSocket: Used to send and receive datagram packets.
- DatagramPacket: Represents the actual data being sent or received.
- Unlike TCP sockets, UDP sockets do not establish a persistent connection, making them faster but less reliable.

Algorithm

Server Side:

1. Create a DatagramSocket.
2. Read user input from console.
3. Create DatagramPacket with client address and port.
4. Send the packet to the client.
5. Repeat steps 2–4.

Client Side:

1. Create a DatagramSocket and bind it to a port.
2. Create a DatagramPacket to receive data.
3. Receive packet from server.
4. Display the message.
5. Repeat steps 2–4.

Java UDP Server Code

```
import java.net.*;
import java.util.Scanner;

public class UDPServer {
   public static void main(String[] args) {
      try {
         DatagramSocket serverSocket = new DatagramSocket();
```

```
        InetAddress clientAddress = InetAddress.getByName("localhost");
        int clientPort = 12345;

        Scanner sc = new Scanner(System.in);
        System.out.println("UDP Server is running... Type messages to send");

        while (true) {
            System.out.print("Enter message: ");
            String message = sc.nextLine();
            byte[] buffer = message.getBytes();

            DatagramPacket packet = new DatagramPacket(
                    buffer, buffer.length, clientAddress, clientPort);

            serverSocket.send(packet);
            System.out.println("Message sent to client: " + message);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
  }
}
```

Java UDP Client Code

```
import java.net.*;

public class UDPClient {
  public static void main(String[] args) {
    try {
        DatagramSocket clientSocket = new DatagramSocket(12345);
        byte[] buffer = new byte[1024];

        System.out.println("UDP Client is ready to receive messages...");

        while (true) {
            DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
            clientSocket.receive(packet);

            String message = new String(packet.getData(), 0, packet.getLength());
            System.out.println("Message from server: " + message);
        }
    } catch (Exception e) {
        e.printStackTrace();
```

```
    }
  }
}
```

Expected Output

Server types: Hello Client!
Client displays: Message from server: Hello Client!

Result

The UDP client-server program in Java was successfully executed. The client was able to display the messages typed at the server side, demonstrating datagram socket communication.

Pre-Viva Questions

1. Differentiate between TCP and UDP.
2. What is a datagram socket?
3. Why is UDP considered unreliable?
4. Which real-world applications use UDP?

Post-Viva Questions

1. How does UDP handle packet loss?
2. Can you make UDP communication reliable? If yes, how?
3. How would you modify this program for two-way communication?
4. What changes are required to run this on different machines instead of localhost?