Lab Report-02

Gandaki College of Engineering and Science

Distributed System

Lab Experiment: UDP Protocol

Submitted By:

Name: Dipendra Raut Kurmi

Roll.No: 18

Batch: 2021SE

Submitted To:

Er. Amrit Poudel

Lecturer at Gandaki College Of

Engineering and Science

Objective:

To implement data communication between a client and server using the UDP protocol in Java.

Theory:

UDP (User Datagram Protocol) is a connectionless transport layer protocol that sends data as independent packets called datagrams. It does not guarantee delivery, order, or duplicate protection, but is faster and has lower overhead than TCP.

Features of UDP:
- No connection setup (connectionless)
- No guarantee of delivery
- No ordering of packets
- Fast and low latency
- Suitable for real-time apps (VoIP, video streaming, online gaming)

Code:

// UDPServer.java

```
import java.net.*;

public class UDPServer {
public static void main(String[] args) {
int port = 9876;

try (DatagramSocket serverSocket = new DatagramSocket(port)) {
byte[] receiveData = new byte[1024];
byte[] sendData;

System.out.println("UDP Server started. Listening on port " + port);

while (true) {
DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
serverSocket.receive(receivePacket);

String sentence = new String(receivePacket.getData(), 0,
receivePacket.getLength());
System.out.println("RECEIVED: " + sentence);

InetAddress clientAddress = receivePacket.getAddress();
int clientPort = receivePacket.getPort();

String capitalizedSentence = sentence.toUpperCase();
```

```
sendData = capitalizedSentence.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
clientAddress, clientPort);
serverSocket.send(sendPacket);

System.out.println("Sent response to client.\n");
}

} catch (Exception e) {
System.err.println("Server exception: " + e.getMessage());
e.printStackTrace();
}
}
}
```

//UDPClient.java

```
import java.net.*;

import java.io.*;

public class UDPClient {
public static void main(String[] args) {
String serverAddress = "127.0.0.1";
int port = 9876;

try (DatagramSocket clientSocket = new DatagramSocket()) {
BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));

byte[] sendData;
byte[] receiveData = new byte[1024];

System.out.print("Enter message: ");
String sentence = userInput.readLine();

sendData = sentence.getBytes();
InetAddress serverIP = InetAddress.getByName(serverAddress);

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
serverIP, port);
clientSocket.send(sendPacket);

DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
```

```
clientSocket.receive(receivePacket);

String modifiedSentence = new String(receivePacket.getData(), 0,
receivePacket.getLength());
System.out.println("FROM SERVER: " + modifiedSentence);

} catch (Exception e) {
System.err.println("Client exception: " + e.getMessage());
e.printStackTrace();
}
}
}
```

Result:

When running the UDPServer.java and UDPClient.java programs:

1. The server listens on port 9876.
2. The client sends a message to the server using UDP datagrams.
3. The server receives the message, processes it (converts it to uppercase), and sends it back to the client.
4. The client receives and displays the server's response.

Server:

```
dipendra@dipendra-Vostro-15-3510:~/Documents/BE/7th Semester/DS_lab/Lab-02$ java UDPServer
UDP Server started. Listening on port 9876
RECEIVED: hello this is message from udp
Sent response to client.
```

Client:

```
dipendra@dipendra-Vostro-15-3510:~/Documents/BE/7th Semester/DS_lab/Lab-02$ java UDPClient
Enter message: hello this is message from udp
FROM SERVER: HELLO THIS IS MESSAGE FROM UDP
dipendra@dipendra-Vostro-15-3510:~/Documents/BE/7th Semester/DS lab/Lab-02$
```

Conclusion:

Hence, we successfully connected client and server through UDP protocol.In this lab, we implemented a basic UDP client-server program where the client sends a message to the server, and the server responds after processing it.