Nepal College of Information Technology

Balkumari, Lalitpur

(Affiliated to Pokhara University)

A Lab Report

On

UDP Echo Communication Using Datagram Sockets in C

Submitted as partial fulfillment of requirement of the curriculum of Bachelor's of Engineering in Software Engineering (6th Semester)

Submitted by:

Harsh Chaudhary Kalwar

(221715)

Submitted to:

Madan Bhandari

Date:

19th June, 2025

Objective:

The objective of this lab was to implement a connectionless communication model using the User Datagram Protocol (UDP). We built a simple Echo Server and Client where the client sends a message and the server responds with the same message, demonstrating the usage of sendto() and recvfrom() system calls.

Lab Tasks and Execution:

1. UDP Echo Server (UDPEchoServer.c):

Functionality:

· Binds to a user-defined UDP port.
· Receives datagrams from any client.
· Prints client IP and received message.
· Echoes back the same message to the client.

Key System Calls & Concepts:

· socket(): Creates a UDP socket.
· bind(): Binds socket to local IP and port.
· recvfrom(): Receives datagrams.
· sendto(): Sends response back to sender.

Important Snippet:

```
recvMsgSize = recvfrom(sock, echoBuffer, ECHOMAX, 0,
(struct sockaddr *) &echoClntAddr, &cliAddrLen);
sendto(sock, echoBuffer, recvMsgSize, 0,
(struct sockaddr *) &echoClntAddr, sizeof(echoClntAddr));
```

2. UDP Echo Client (UDPEchoClient.c):

Functionality:

· Takes server IP and optional port as arguments.
· Prompts the user to input a message.
· Sends the message to the server.
· Receives the echoed message from the server and displays it.

Key System Calls & Concepts:

· socket(): Creates UDP socket.
· sendto(): Sends message to server.
· recvfrom(): Waits for and receives echoed response.

Important Snippet:

```
fgets(echoString, ECHOMAX, stdin);
sendto(sock, echoString, strlen(echoString), 0,
(struct sockaddr *)&echoServAddr, sizeof(echoServAddr));
recvfrom(sock, echoBuffer, ECHOMAX, 0,
(struct sockaddr *)&fromAddr, &fromSize);
```

Output / Observations:

- The client successfully sent a message like "Hello UDP Server\n" to the server.
- The server displayed:
- Handling client 127.0.0.1
- Received size = 17
- Received message: Hello UDP Server
- The client printed:
- Received: Hello UDP Server

Note: The communication happened over the loopback address 127.0.0.1, ensuring local testing.

Conclusion:

This lab demonstrated how to implement basic UDP-based communication using datagram sockets. Unlike TCP, UDP is connectionless and does not guarantee delivery or ordering, but is simpler and faster. We successfully developed a basic Echo Server and Client, verifying the use of sendto() and recvfrom().