

Course	COMP 7005
Program	BScACS, Network Security
Term	September 2024

- This is a **pair** assignment.
- ***You must work in pairs unless you have explicit permission from the instructor.***

## Objective

- This assignment introduces students to creating a reliable communication protocol using UDP. You will implement a client, a server, and a proxy server to simulate unreliable network conditions. This exercise builds skills in handling UDP sockets, implementing reliable transmission over an unreliable medium, and simulating packet loss and delay.

## Learning Outcomes

- Upon completing this assignment, students will be able to:
  - Understand the challenges of reliable data transmission using UDP.
  - Develop a protocol for reliable communication using acknowledgments and retransmissions.
  - Implement a proxy server to simulate packet loss and delays.
  - Test and analyze the performance of a custom reliable protocol.

## Assignment Details

### Client

- Reads input from the keyboard and sends messages to the server using UDP.
- Implements a simple protocol to handle acknowledgments and retransmissions.
- Retries sending messages if no acknowledgment is received within a timeout period.
- Uses command-line arguments to specify the server's IP address and port.

### Server

- Listens for incoming messages on a specified IP address and port using UDP.
- Receives messages from the client and displays them on the screen.
- Sends acknowledgments back to the client to confirm message receipt.

## Proxy Server

- Sits between the client and server, relaying UDP packets while simulating an unreliable network.
- Has configurable parameters for packet loss and delay:
  - Drop Chance: Probability (0% - 100%) of dropping packets from the client or server.
  - Delay Chance: Probability (0% - 100%) of delaying packets from the client or server.
  - Delay Time: Specifies how long to delay packets. This can be a fixed value or a random value within a specified range.
- Must handle independent configuration for client-to-server and server-to-client directions.
- Listens on a specified IP address and port for incoming packets from the client.

## Command-Line Arguments

### Client

- --target-ip: IP address of the server.
- --target-port: Port number of the server.
- --timeout: Timeout (in seconds) for waiting for acknowledgments.

```
client --target-ip 192.168.1.10 --target-port 5000 --timeout 2
```

### Server

- --listen-ip: IP address to bind the server.
- --listen-port: Port number to listen on.

```
server --listen-ip 0.0.0.0 --listen-port 5000
```

## Proxy Server

- --listen-ip: IP address to bind the proxy server.
- --listen-port: Port number to listen for client packets.
- --target-ip: IP address of the server to forward packets to.
- --target-port: Port number of the server.
- --client-drop: Drop chance (0% - 100%) for packets from the client.
- --server-drop: Drop chance (0% - 100%) for packets from the server.
- --client-delay: Delay chance (0% - 100%) for packets from the client.
- --server-delay: Delay chance (0% - 100%) for packets from the server.
- --client-delay-time: Delay time in milliseconds (fixed or range).
- --server-delay-time: Delay time in milliseconds (fixed or range).

```
proxy --listen-ip 0.0.0.0 --listen-port 4000 --target-ip
192.168.1.10 --target-port 5000 --client-drop 10
--server-drop 5 --client-delay 20 --server-delay 15
--client delay-time 100 --server-delay-time 200
```

## Constraints

- All client, proxy server, and server communication must use UDP.
- Ensure your protocol handles packet loss and delay effectively, implementing retries and timeouts.
- The proxy server must be configurable without restarting the process (bonus marks).
- You must test your solution with the following configurations:
  - Drop Chance: 0%, 50%, and 100%.
  - Delay Chance: 0%, 50%, and 100%.
  - Delay Time: Test with a delay longer than the client's retry timeout.
- Provide a test report with observations for each configuration.

## Graphs and Analysis

- Create graphs showing:
  - Number of packets sent, received, retransmitted, and lost for the client and server.
  - Latency was observed with different delay configurations in the proxy server.
- Bonus Marks
  - Implement the ability to dynamically change proxy server parameters (drop chance, delay chance, delay time) without restarting the proxy process.

## Submission

- Ensure your submission meets all the [guidelines](#), including formatting, file type, and [submission](#).
- Follow the [AI usage guidelines](#).
- Be aware of the [late submission policy](#) to avoid losing marks.
- **Note: Please strictly adhere to the submission requirements to ensure you don't lose any marks.**

## Evaluation

Topic	Value
Reliable Protocol Design	40%

Proxy Server Implementation	30%
Testing and Analysis	20%
Documentation	10%
Total	100%

## Hints

- Start by implementing a basic client and server without the proxy server.
- Introduce the proxy server after testing client-server communication.
- Use select or poll to handle timeouts for receiving acknowledgments.
- Test your protocol with various configurations of drop and delay parameters.
- Use elevated privileges (e.g., sudo) to bind to certain ports if necessary.