

COMP 7005

Project 1

User Guide

Brandon Rada
A01345707
Dec 2nd, 2025

Purpose

Understand the limitations of UDP and the need for reliability mechanisms. Design a message-based protocol with identifiers and acknowledgments. Implement and evaluate retry logic, timeout handling, and error cases. Simulate packet loss and delay using a configurable proxy server. Measure and describe how the system performs under degraded conditions. Implement reliable communication over UDP by simulating network unreliability and developing a simple protocol that utilizes retransmissions and acknowledgments.

Three programs: a client, a server, and a proxy server that introduces packet loss and delay.

Installing

Obtaining

```
git clone https://github.com/BrandonRada/COMP7005-project
```

Building

```
cd into the cloned repo  
cd /COMP7005-project/source/reliable_udp
```

Running

CLIENT:

Without proxy:

```
python3 client.py --target-ip SERVER_IP --target-port 9001  
--timeout 1 --max-retries 5
```

With proxy:

```
python3 client.py --target-ip PROXY_IP --target-port 4000  
--timeout 1 --max-retries 5
```

SERVER:

```
python3 server.py --listen-ip SERVER_IP --listen-port 9001
```

PROXY:

```
python3 proxy.py \  
    --listen-ip PROXY_PC_IP --listen-port 4000 \  
    --target-ip SERVER_IP --target-port 9001 \  
    --client-drop 0 --server-drop 0 \  
    
```

```
--client-delay 0 --server-delay 0
```

EXTRA:

In the same folder the 3 programs are run, there is an extra program called clear_logs.py. Run it with **python3 clear_logs.py** to clear all 3 log files of their contents.

I made this as a tool to help quickly move onto the next test without manually clearing the contents of each file.

Environment Variables

The following environment variables alter the behaviour of main:

Variable	Purpose

Configuration

The following configuration values can be set in <file>:

Variable	Purpose

Command Line Arguments

The following configuration values can be set in <file>:

Variable	Purpose
Client	
target-ip	IP address to send packets to
target-port	Port to send packets to
timeout	timeout in seconds for ACKs
max-retries	maximum number of retries for sending a message
Server	
listen-ip	IP address to bind for incoming packets

listen-port	Port to listen on for client packets
Proxy	
listen-ip	IP address to bind for client packets
listen-port	Port to listen on for client packets
target-ip	Server IP address to forward packets to
target-port	Server port number
client-drop	percentage chance of dropping client -> server packets
server-drop	percentage chance of dropping server -> client packets
client-delay	percentage chance of delaying client -> server packets
server-delay	percentage chance of delaying server -> client packets
client-delay-time-min	minimum delay time in milliseconds for client -> server packets
client-delay-time-max	maximum delay time in milliseconds for client -> server packets
server-delay-time-min	minimum delay time in milliseconds for server -> client packets
server-delay-time-max	maximum delay time in milliseconds for server -> client packets

Examples

Lets try an example with a server and client with no proxy:

So the client will run:

```
python3 client.py --target-ip 127.0.0.1 --target-port 9001 --timeout 1 --max-retries 5
```

```
▶ ~/De/COMP7005/p/COMP7005-project/s/reliable_udp▶ p main +4 !5 ?2▶ python3 client.py --target-ip 127.0.0.1 --target-port 9001 --timeout 1 --max-retries 5
[2025-12-02 21:27:41] Client sending to 127.0.0.1:9001
```

And the server will run:

```
python3 server.py --listen-ip 127.0.0.1 --listen-port 9001
```

```
■ ➤ ~/De/COMP7005/p/COMP7005-project/s/reliable_udp ➤ ⚡ main +4 !5 ?2 ➤ python3 server.py --listen-ip 127.0.0.1 --listen-port 9001
```

Then in the client console type some messages:

```
■ ➤ ~/De/COMP7005/p/COMP7005-project/s/reliable_udp ➤ ⚡ main +4 !5 ?2 ➤ python3 client.py --target-ip 127.0.0.1 --target-port 9001 --timeout 1 --max-retries 5
[2025-12-02 21:27:41] Client sending to 127.0.0.1:9001
hello
[2025-12-02 21:27:50] SENT | seq=1 retries=0 payload=b'hello'
[2025-12-02 21:27:50] RECEIVED ACK for seq=1
[2025-12-02 21:27:50] Message seq=1 delivered
wow, so cool!
[2025-12-02 21:27:56] SENT | seq=2 retries=0 payload=b'wow, so cool!'
[2025-12-02 21:27:56] RECEIVED ACK for seq=2
[2025-12-02 21:27:56] Message seq=2 delivered
■
```

And see the server's prints!

```
■ ➤ ~/De/COMP7005/p/COMP7005-project/s/reliable_udp ➤ ⚡ main +4 !5 ?2 ➤ python3 server.py --listen-ip 127.0.0.1 --listen-port 9001
[2025-12-02 21:27:43] Server listening on 127.0.0.1:9001
[2025-12-02 21:27:50] RECEIVED from ('127.0.0.1', 52143) | seq=1 payload=b'hello'
hello
[2025-12-02 21:27:50] SENT ACK to ('127.0.0.1', 52143) | ack=1
[2025-12-02 21:27:56] RECEIVED from ('127.0.0.1', 52143) | seq=2 payload=b'wow, so cool!'
wow, so cool!
[2025-12-02 21:27:56] SENT ACK to ('127.0.0.1', 52143) | ack=2
■
```

and then on both programs hit ctrl-c:

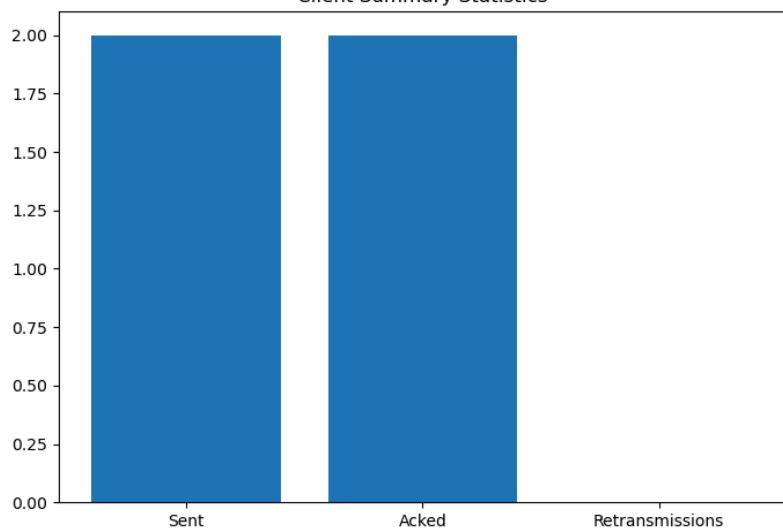
```
[plot] Saved client_graph.png
[plot] Saved server_graph.png
■ ➤ ~/De/COMP7005/p/COMP7005-project/s/reliable_udp ➤ ⚡ main +4 !5 ?2 ➤
```

Plots will have been made:

These can be found in the same folder as the files that you are running:

/COMP7005-project/source/reliable_udp/

Client Summary Statistics



Server Summary Statistics

