PCD - H1 - Technical Report

Popa Stefan-Eduard

1. Introduction

This technical report examines the performance of data transfer protocols, specifically **UDP (User Datagram Protocol), TCP (Transmission Control Protocol), and QUIC (Quick UDP Internet Connections)**. The primary objective is to evaluate the time required for transferring different amounts of data across various message sizes and transmission mechanisms, providing a comparative analysis of their efficiency and reliability.

1.1. Homework objectives

The task entails designing a program capable of measuring data transfer time for different message sizes (1 to 65535 bytes) and amounts of data (500MB, 1GB) using UDP, TCP and Quic protocols. Additionally, the program must support both streaming and stop-and-wait transmission mechanisms.

1.2. Implementation requirements

- **Supported Protocols:** UDP, TCP and Quic protocols must be supported as parameters for both client and server components.
- **Message Size:** The program should facilitate data transfer across a range of message sizes to assess performance implications.
- **Transmission Mechanisms:** Two transmission mechanisms, streaming and stop-and-wait, should be implemented for comprehensive analysis.

1.3. Output requirements

- **Server Output:** The server will print the protocol used, number of messages read, and bytes read after each session.
- **Client Output:** Upon completion, the client will display transmission time, number of sent messages, and total bytes sent.

2. Experiment

The experiments were conducted on a local machine, using a laptop equipped with an AMD Ryzen 7 5800HS processor and 16GB of RAM. Each protocol and transmission mechanism combination was executed five times on the local host, and the minimum, maximum, and average values were computed for each statistical metric.

For the Stop-and-Wait mechanism, a 1-second timeout was implemented on the client side. If an acknowledgment is not received within this timeframe, the client does not proceed to the next packet but instead continues retransmitting the same packet until the acknowledgment is successfully received.

2.1. TCP

I used for both of the mechanisms a buffer of 1024 bytes.

2.1.1. Streaming

```
Statistics for Client TCP Streaming (500MB):
--------------------------------------------------
Messages: Min = 512000, Avg = 512000.00, Max = 512000
Bytes: Min = 524288000, Avg = 524288000.00, Max = 524288000
Time (s): Min = 5.43, Avg = 6.56, Max = 8.01
--------------------------------------------------
```

```
Statistics for Client TCP Streaming (1GB):
--------------------------------------------------
Messages: Min = 1048576, Avg = 1048576.00, Max = 1048576
Bytes: Min = 1073741824, Avg = 1073741824.00, Max = 1073741824
Time (s): Min = 9.80, Avg = 12.72, Max = 16.97
--------------------------------------------------
```

Image 1: TCPS Client statistics

```
Statistics for Server TCP Streaming (500MB):
--------------------------------------------------
Messages: Min = 573682, Avg = 588320.60, Max = 601428
Bytes: Min = 524288000, Avg = 524288000.00, Max = 524288000
Time (s): Min = 5.43, Avg = 6.56, Max = 8.01
--------------------------------------------------
```

```
Statistics for Server TCP Streaming (1GB):
--------------------------------------------------
Messages: Min = 1159679, Avg = 1201959.00, Max = 1229235
Bytes: Min = 1073741824, Avg = 1073741824.00, Max = 1073741824
Time (s): Min = 9.80, Avg = 12.72, Max = 16.97
--------------------------------------------------
```

Image 2: TCPS Server statistics

2.1.2. Stop and wait

```
Statistics for Client TCP Stop and Wait (500MB):
--------------------------------------------------
Messages: Min = 512000, Avg = 512000.00, Max = 512000
Bytes: Min = 524288000, Avg = 524288000.00, Max = 524288000
Time (s): Min = 41.12, Avg = 45.19, Max = 54.11
--------------------------------------------------


Statistics for Client TCP Stop and Wait (1GB):
--------------------------------------------------
Messages: Min = 1048576, Avg = 1048576.00, Max = 1048576
Bytes: Min = 1073741824, Avg = 1073741824.00, Max = 1073741824
Time (s): Min = 73.98, Avg = 87.38, Max = 97.24
--------------------------------------------------
```

Image 3: TCPSW Client statistics

```
Statistics for Server TCP Stop and Wait (500MB):
--------------------------------------------------
Messages: Min = 512000, Avg = 512000.00, Max = 512000
Bytes: Min = 524288000, Avg = 524288000.00, Max = 524288000
Time (s): Min = 41.12, Avg = 45.19, Max = 54.11
--------------------------------------------------


Statistics for Server TCP Stop and Wait (1GB):
--------------------------------------------------
Messages: Min = 1048576, Avg = 1048576.00, Max = 1048576
Bytes: Min = 1073741824, Avg = 1073741824.00, Max = 1073741824
Time (s): Min = 73.98, Avg = 87.38, Max = 97.26
--------------------------------------------------
```

Image 4: TCPSW Server statistics

2.2. UDP

I used for both of the mechanisms a buffer of 1024 bytes on the local host.

2.2.1. Streaming

```
Statistics for Client UDP Streaming (500MB):
-------------------------------------------------
Messages: Min = 512000, Avg = 512000.00, Max = 512000
Bytes: Min = 524288000, Avg = 524288000.00, Max = 524288000
Time (s): Min = 79.08, Avg = 85.57, Max = 98.01
-------------------------------------------------
```

Image 5: UDPS Client statistics

```
Statistics for Server UDP Streaming (500MB):
-------------------------------------------------
Messages: Min = 512000, Avg = 512000.00, Max = 512000
Bytes: Min = 524288000, Avg = 524288000.00, Max = 524288000
Time (s): Min = 81.08, Avg = 87.37, Max = 99.42
-------------------------------------------------
```

Image 6: UDPS Server statistics

2.2.2. Stop and wait

```
Statistics for Client UDP Stop and Wait (500MB):
-------------------------------------------------
Messages: Min = 512000, Avg = 512000.00, Max = 512000
Bytes: Min = 524288000, Avg = 524288000.00, Max = 524288000
Time (s): Min = 103.92, Avg = 111.80, Max = 126.61
-------------------------------------------------
```

Image 7: UDPSW Client statistics

```
Statistics for Server UDP Stop and Wait (500MB):
-------------------------------------------------
Messages: Min = 512000, Avg = 512000.00, Max = 512000
Bytes: Min = 524288000, Avg = 524288000.00, Max = 524288000
Time (s): Min = 105.92, Avg = 113.59, Max = 127.98
-------------------------------------------------
```

Image 8: UDPSW Server statistics

2.3 Quic

2.3.1 Streaming

Image 9: Quic Streaming Client statistics

Image 10: Quic Streaming Server statistics

3. Discussion

3.1. Local host

The results from the experiments highlight significant differences between TCP and UDP in terms of performance, reliability, and efficiency across various transmission mechanisms.

TCP Streaming demonstrates the best performance among all tested protocols, delivering all the data to the server while maintaining low transmission times. The streaming mechanism allows for continuous data transfer without waiting for acknowledgments, which significantly reduces latency. In contrast, the Stop-and-Wait mechanism for TCP introduces a considerable delay because the client must wait for an acknowledgment before sending the next packet. This results in a much higher transmission time, as seen in the statistics where Stop-and-Wait takes significantly longer than Streaming for both 500MB and 1GB data transfers.

Unlike TCP, UDP does not provide guaranteed delivery, and that may lead to noticeable packet loss, even for smaller data sizes. Although UDP Streaming is generally faster than TCP, it suffers from inconsistent transmission due to packet loss, which could be critical in real-world applications requiring data integrity.

In my tests Quic was was too slow compared to TCP and UDP. Despite this, QUIC has advantages in reducing blocking and providing built-in encryption.

4. Conclusion

In the end, we can’t say that one protocol or mechanism is better than another. All have their advantages and disadvantages, and we need to analyse the use case to find which is best suited for it.