Report | Lab 8

CS378 - Computer Networks Lab

Omkar Shirpure (22B0910)

Contents

1 Introduction

The instructions to run the scripts is given in the provided **`README.md`**.

All the plots are generated using the `plot.py` script from the data obtained from running `script.py`.

The given report compares performance of two TCP congestion control variants, Reno and Cubic, by analyzing throughput under different network conditions using a loopback interface. The experiments evaluate TCP throughput across combinations of network delay (10 ms, 50 ms, and 100 ms) and packet loss rates (0.1%, 0.5%, and 1%), providing insights into how each variant responds to varying levels of latency and packet loss.

2 Plots

a (Loss=0.1%) Throughput (y-axis) vs. Delay (x-axis) for Reno and Cubic

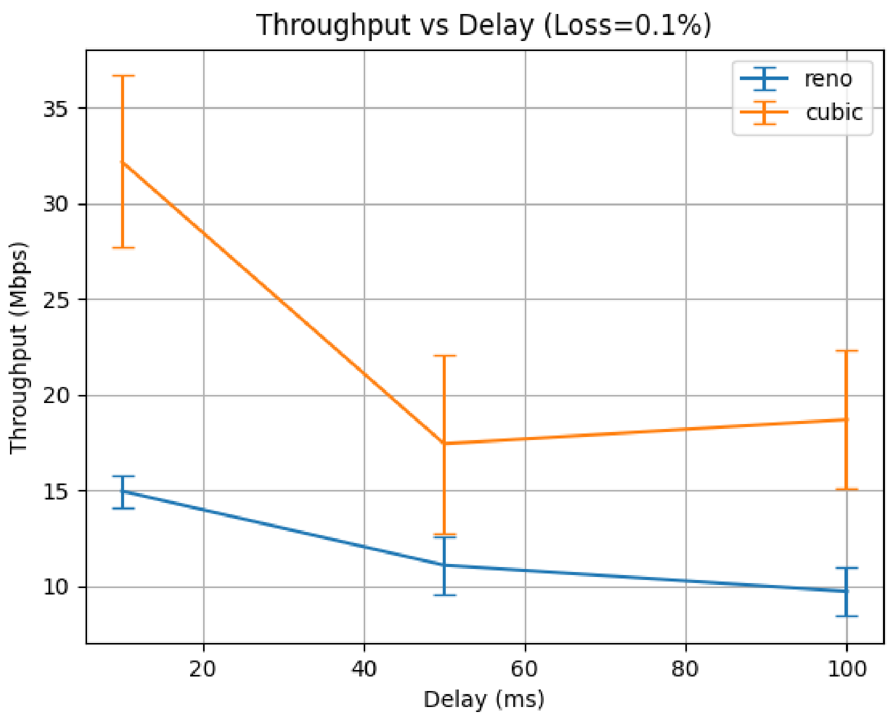

Figure 1: Throughput vs. Delay (Loss=0.1%)

b (Loss=0.5%) Throughput (y-axis) vs. Delay (x-axis) for Reno and Cubic

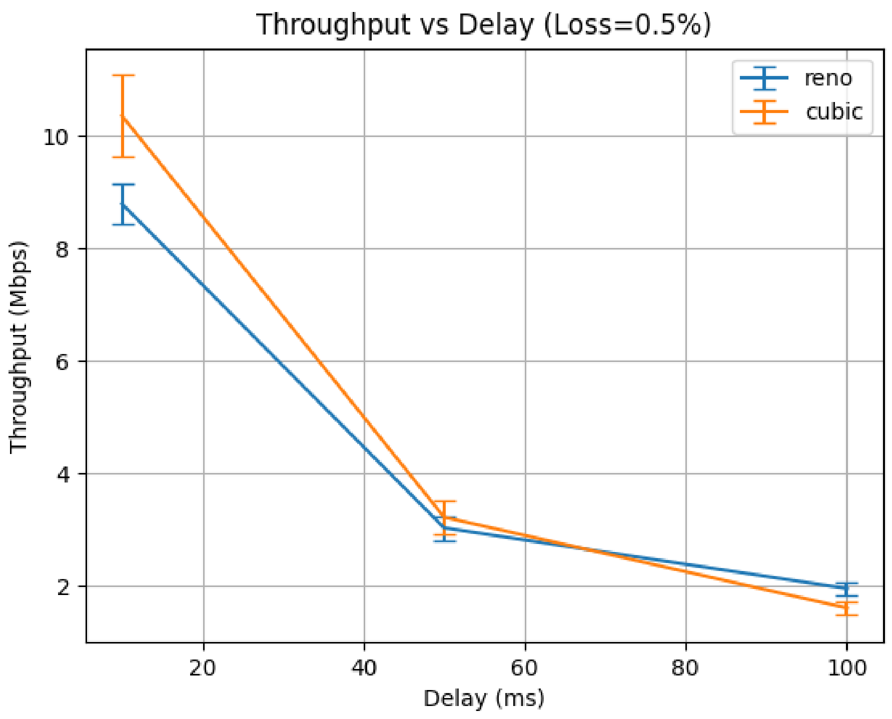

Figure 2: Throughput vs. Delay (Loss=0.5%)

c (Loss=1%) Throughput (y-axis) vs. Delay (x-axis) for Reno and Cubic

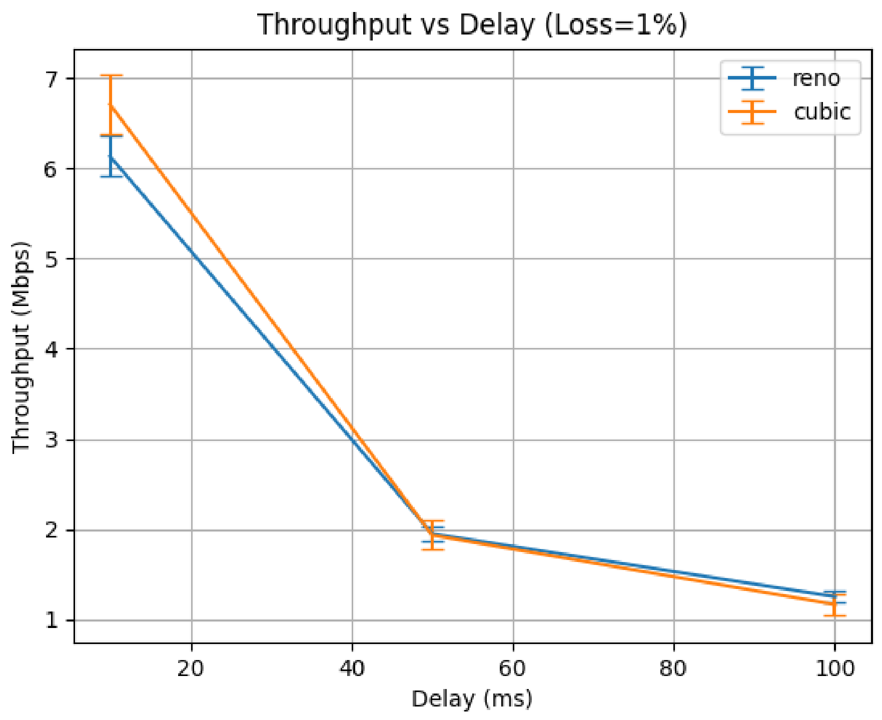

Figure 3: Throughput vs. Delay (Loss=1%)

d (Delay=10ms) Throughput vs. Loss for Reno and Cubic

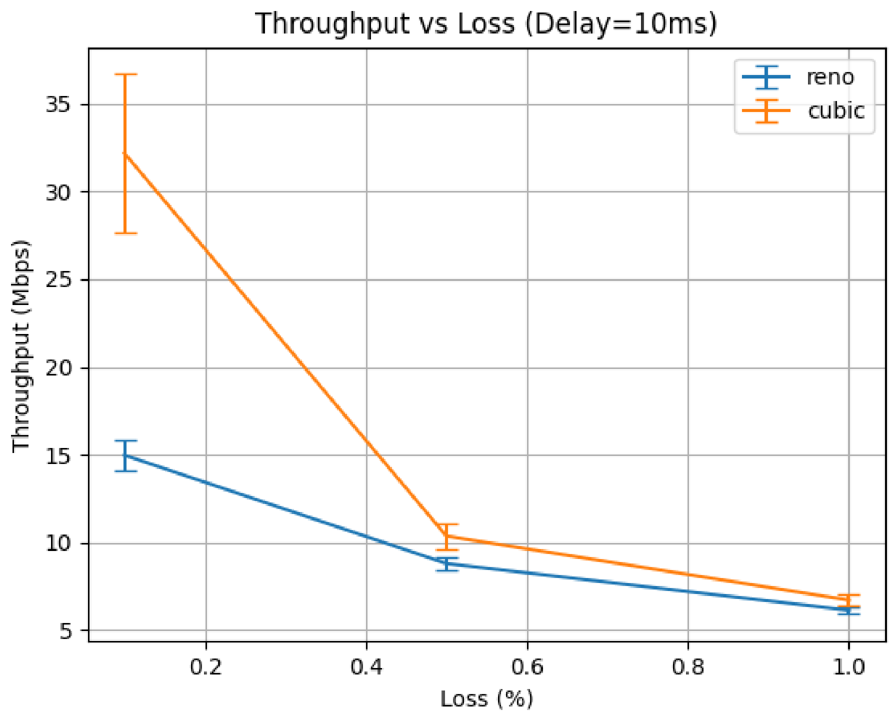

Figure 4: Throughput vs. Loss (Delay=10ms)

e (Delay=50ms) Throughput vs. Loss for Reno and Cubic

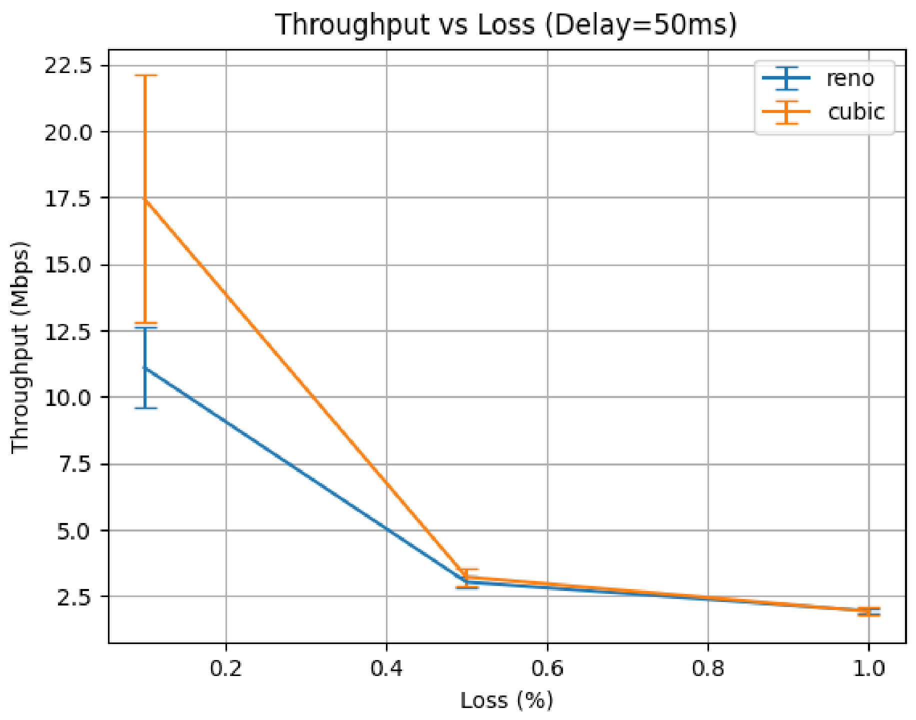

Figure 5: Throughput vs. Loss (Delay=50ms)

f (Delay=100ms) Throughput vs. Loss for Reno and Cubic

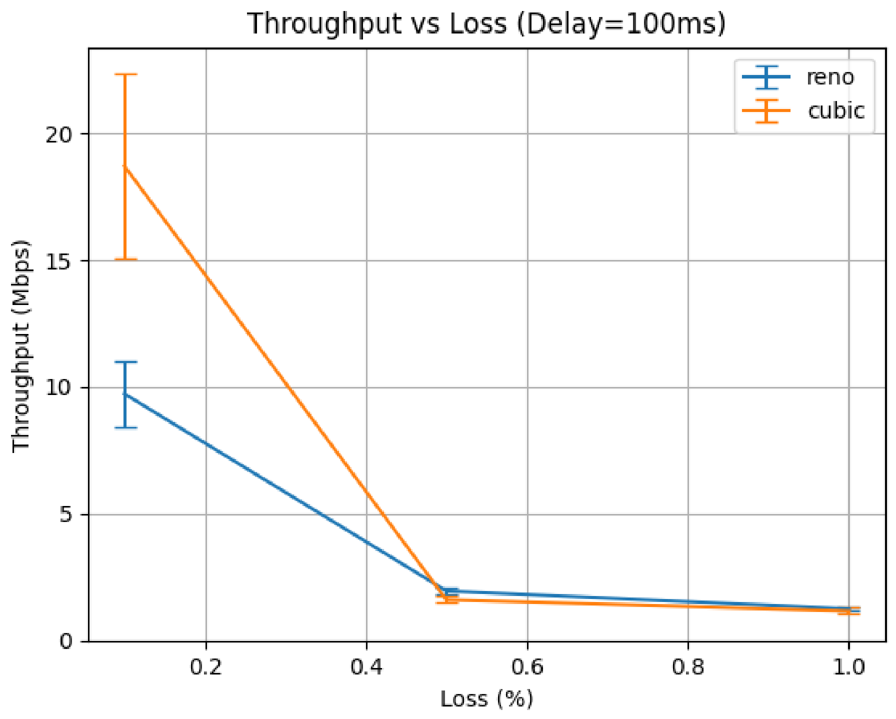

Figure 6: Throughput vs. Loss (Delay=100ms)

3 Throughput vs. Delay

General Observation: For both TCP Reno and TCP Cubic, throughput generally decreases as the delay increases. This is expected, as higher round-trip times (RTTs) reduce the rate at which acknowledgments are received, slowing down the rate at which the congestion window can grow.

- **TCP Reno**: TCP Reno's throughput decreases more sharply with increased delay, especially in cases of low packet loss (e.g., 0.1%). This aligns with Reno's linear congestion window growth, which struggles to maintain throughput as delay increases.
- **TCP Cubic**: TCP Cubic shows a more gradual decline in throughput with increasing delay and generally outperforms Reno at lower delays, particularly when packet loss is minimal. This reflects Cubic's more aggressive nature due to its cubic window growth, which makes it less affected by delay, allowing it to maintain higher throughput at lower delays.

4 Throughput vs. Packet Loss

General Observation: Both TCP Reno and TCP Cubic show a significant drop in throughput as packet loss increases, which is expected, as TCP's congestion control algorithms react to packet loss as a sign of network congestion.

- **TCP Reno**: Its linear congestion window growth strategy seems and makes it more conservative in response to packet loss. As packet loss increases, Reno reduces its throughput significantly, showing a more larger drop compared to Cubic. This conservative approach minimizes the risk of additional packet loss but results in lower throughput, especially at higher loss rates.
- **TCP Cubic**: While TCP Cubic, with its more aggressive congestion window growth during congestion avoidance, keeps higher throughput at lower packet loss rates (e.g., 0.1%) compared to Reno. However, as packet loss reaches higher levels (e.g., 1%), Cubic also sees a substantial decrease in throughput, demonstrating that even its fast recovery approach cannot fully counteract the significant packet loss on throughput.

5 Summary

Expected vs. Observed Behavior: TCP Cubic is generally expected to achieve higher throughput in high-bandwidth, high-delay networks, and in some cases, this is observed in the plots. However, the results also show that Reno sometimes performs similarly to Cubic under specific conditions, which suggests that Cubic's advantages may not be as pronounced in the given network setup, particularly with certain delay-loss combinations.

Also, the results indicate that throughput is highly sensitive to both delay and packet loss. Both variants show reduced throughput as delay and loss increase, but Cubic's design allows it to perform slightly better in some higher-delay or higher-loss conditions, aligning with its more aggressive nature.

6 Cubic vs Reno Comparison

The plots confirm TCP Cubic's aggressive nature, especially at lower delays and loss rates, where it achieves higher throughput than TCP Reno. However, under higher delay and loss conditions, TCP Cubic's advantage diminishes, and it performs similarly to Reno. This suggests that Cubic may be limited by its sensitivity to higher packet loss in these conditions, preventing it from fully leveraging its aggressive window growth.

7 WS Graphs of different states of both TCPs

a TCP Cubic

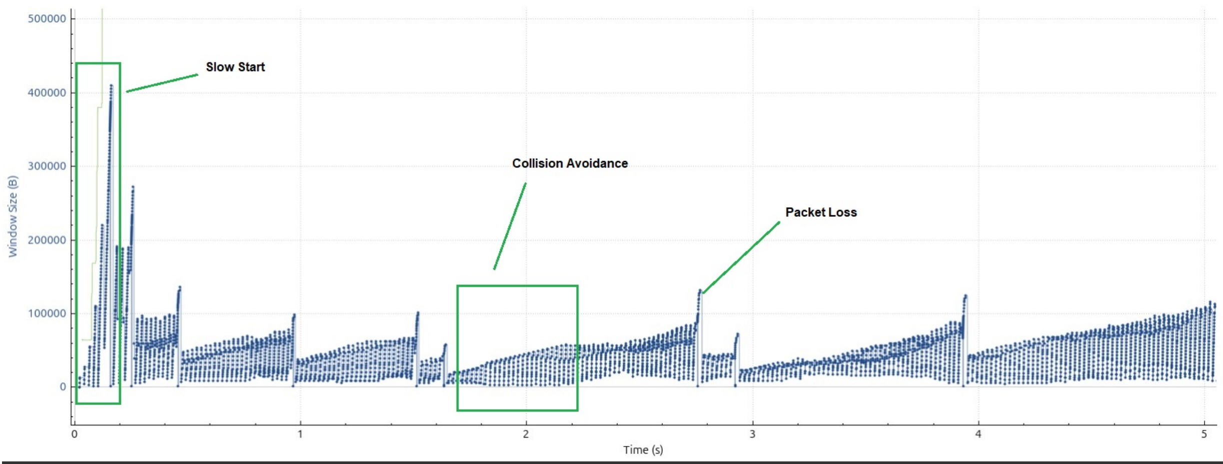

Figure 7: `bytes_in_flight` with Delay = 10ms, Loss=0.1%

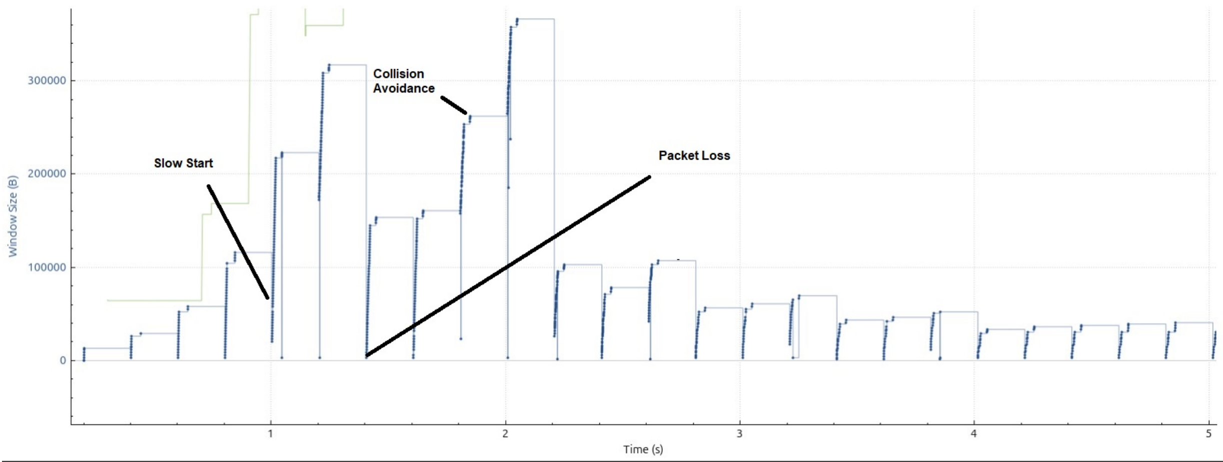

Figure 8: `bytes_in_flight` with Delay = 100ms, Loss=1%

b TCP Reno

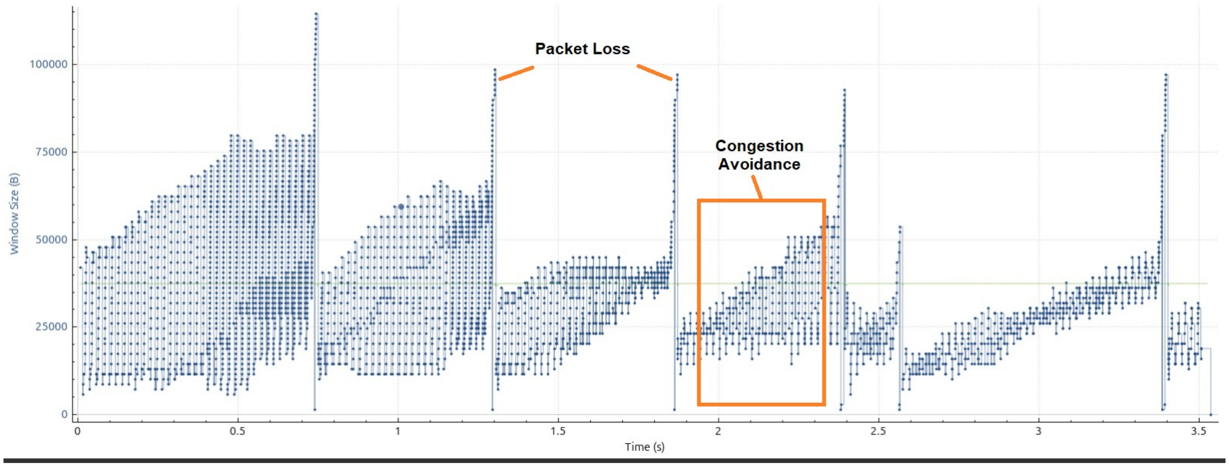

Figure 9: `bytes_in_flight` with Delay = 10ms, Loss=0.1%

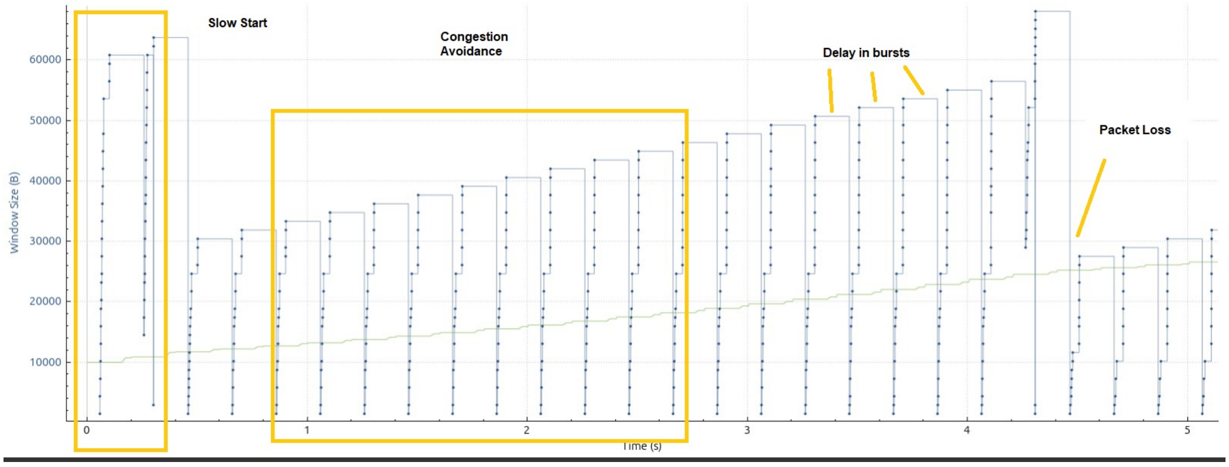

Figure 10: `bytes_in_flight` with Delay = 100ms, Loss=1%