CS378 Lab-8

Saksham Rathi
Department of Computer Science,
Indian Institute of Technology Bombay
22B1003

Result Plots

Here are the plots:

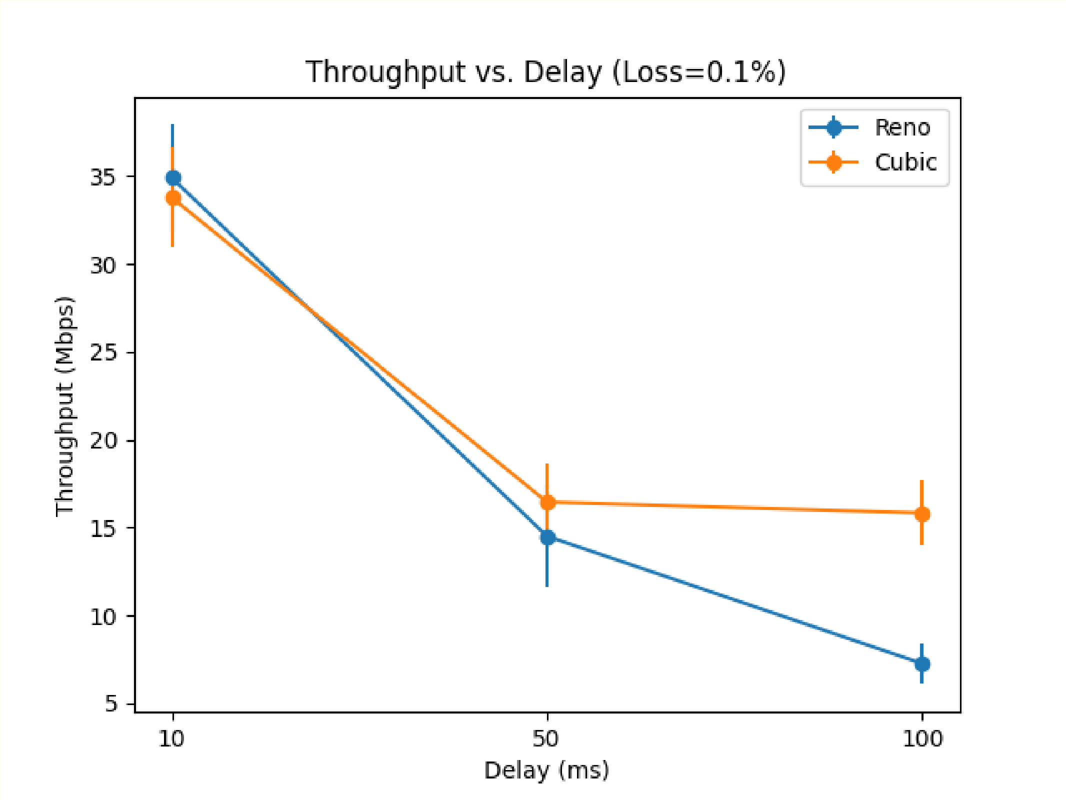


Figure 1: Throughput vs Delay for Loss 0.1%

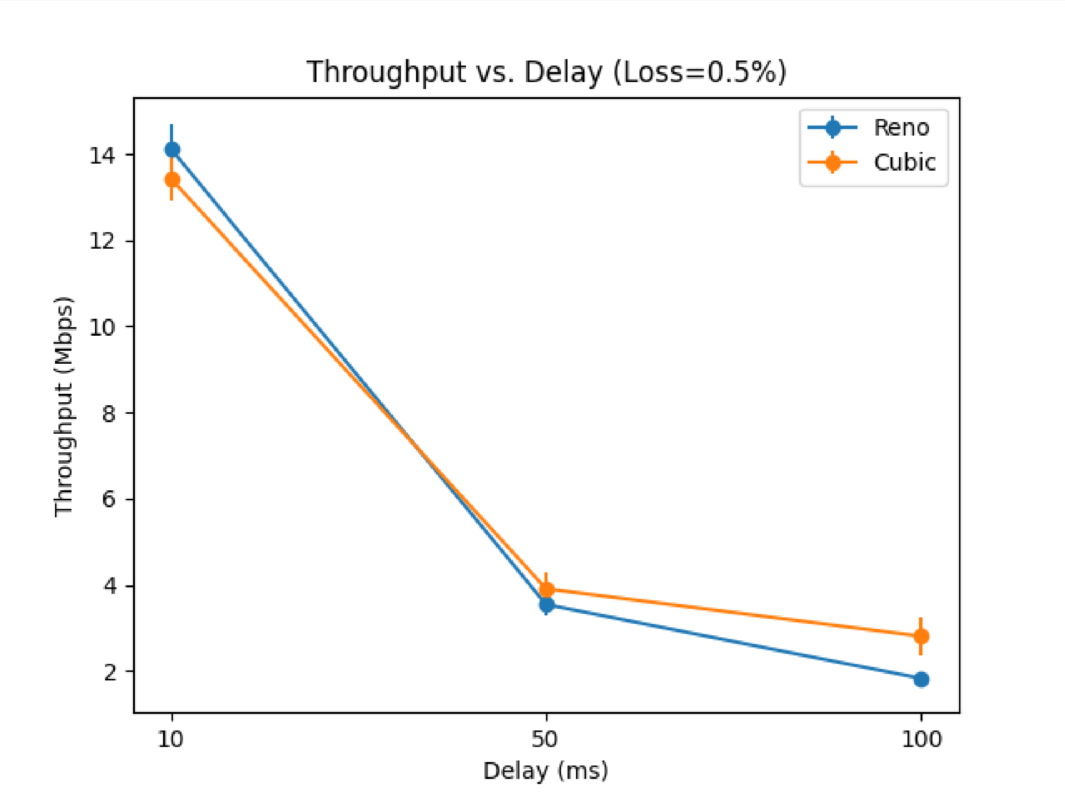


Figure 2: Throughput vs Delay for Loss 0.5%

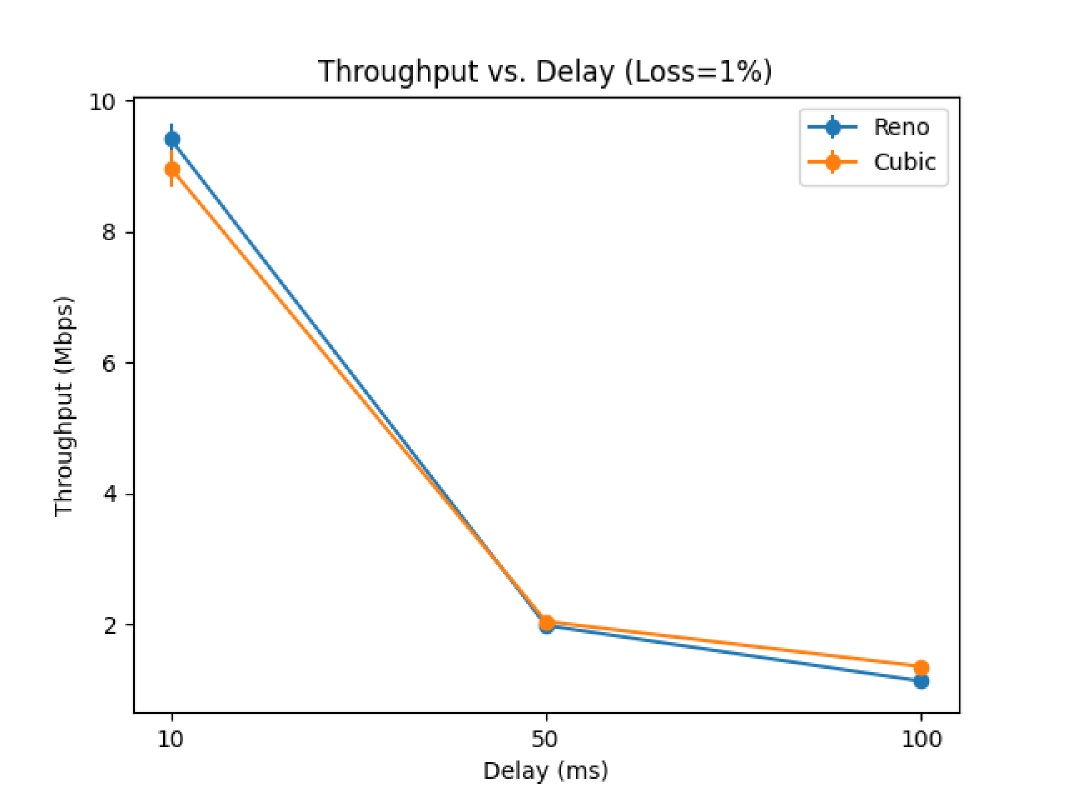


Figure 3: Throughput vs Delay for Loss 1.0%

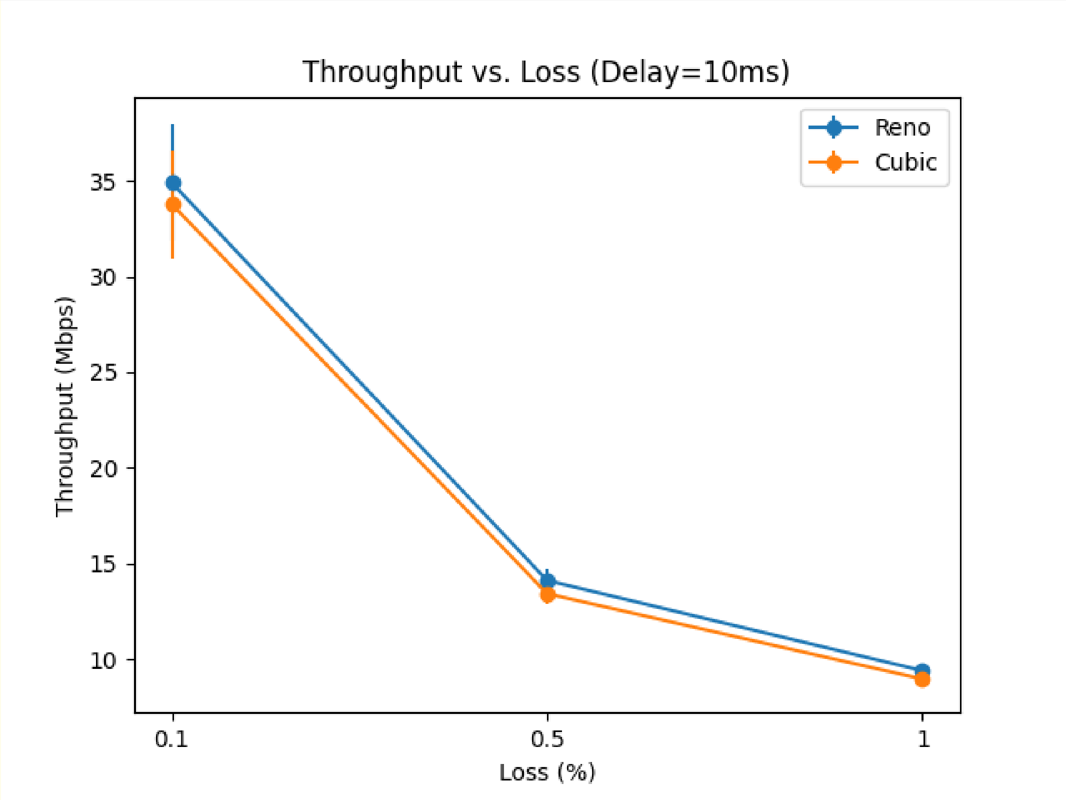


Figure 4: Throughput vs Loss for Delay 10ms

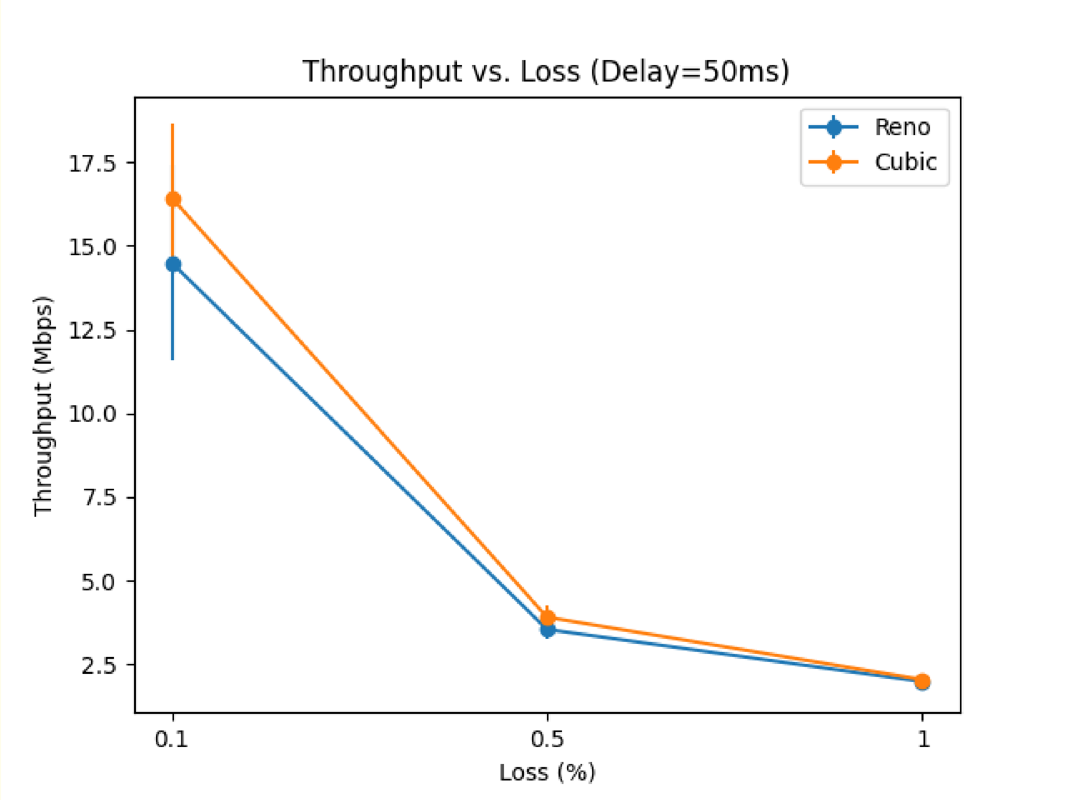


Figure 5: Throughput vs Loss for Delay 50ms

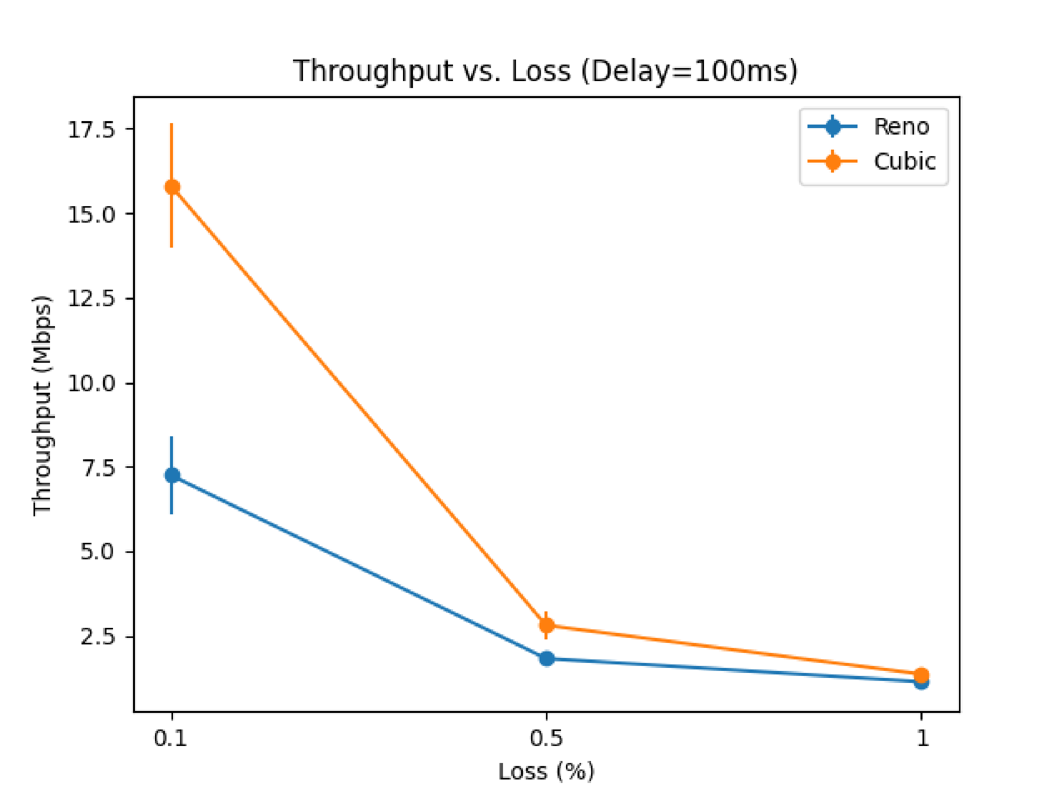


Figure 6: Throughput vs Loss for Delay 100ms

Observations

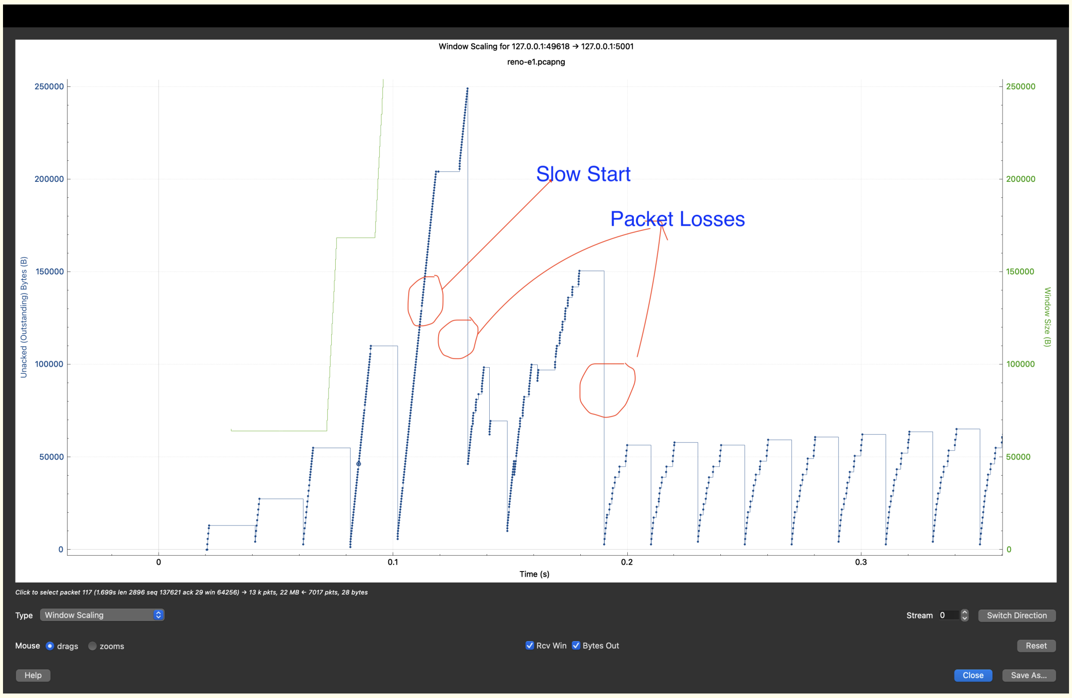
- In all of the plots (and for all of the datapoints), the throughput from cubic is higher than that from reno. This is because cubic is more aggressive in increasing the congestion window size. Therefore it is able to send more packets in the same time (and a higher throughput is observed).
- The throughput decreases as the delay increases. This is because the packets take more time to reach the destination and the sender has to wait for the ACKs. This increases the RTT and hence the throughput decreases. This again confirms the theoretical result that throughput is inversely proportional to the RTT.
- The loss percentage field represents the rate at which packets are dropped. As the loss percentage increases, the throughput decreases. This is because the sender has to retransmit the packets that are lost. This increases the RTT and hence the throughput decreases. When the packet loss rate is 1%, the throughput is really low because the sender has to retransmit almost all the packets.

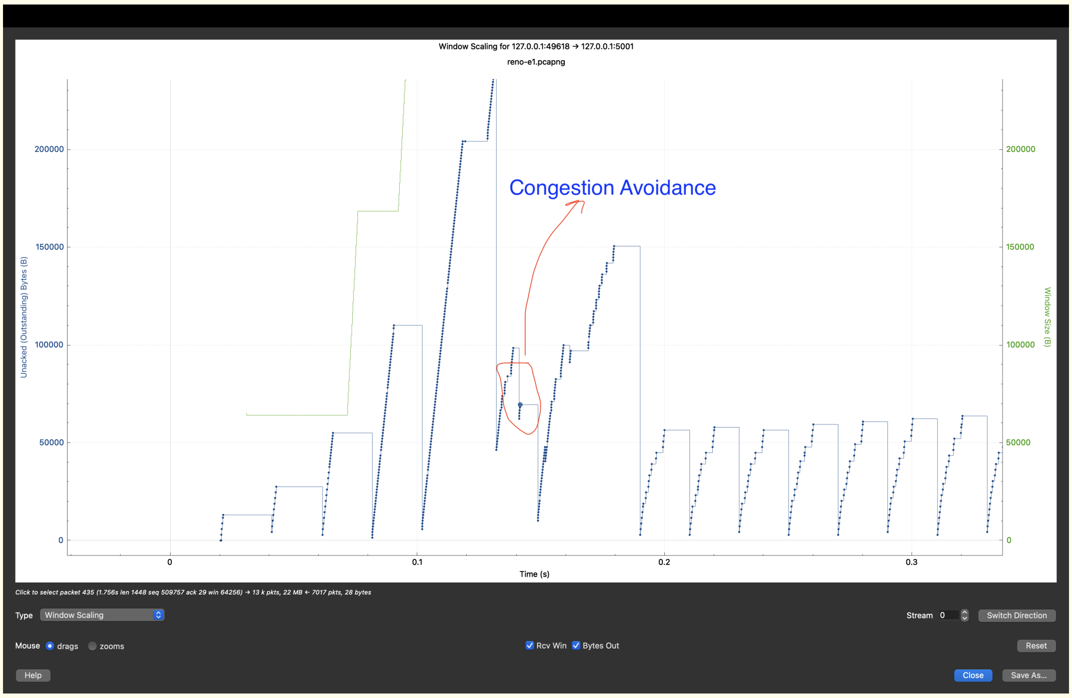
Wireshark Plots

Here are the window scaling graphs from Wireshark:

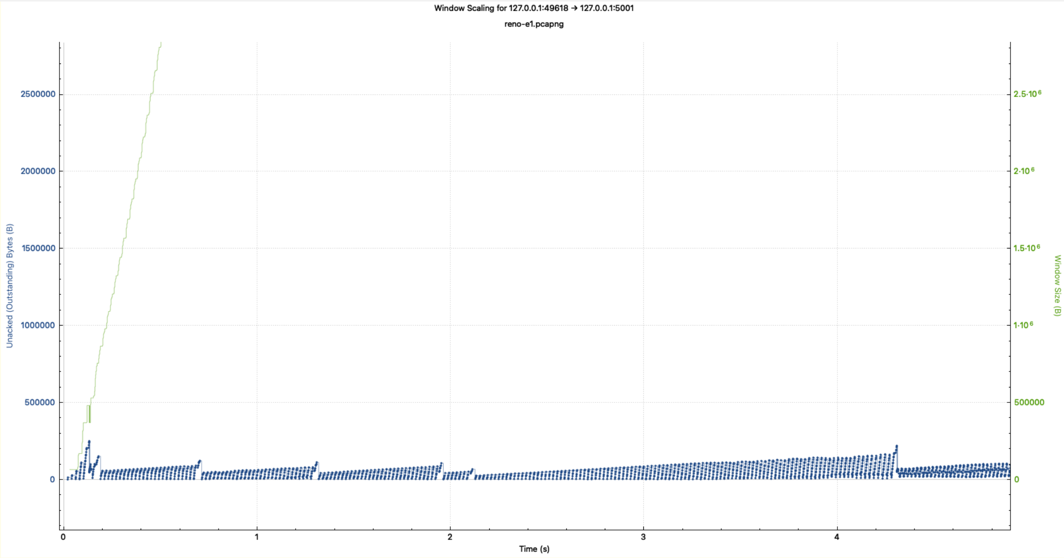
Reno

Delay=10ms, Loss=0.1%

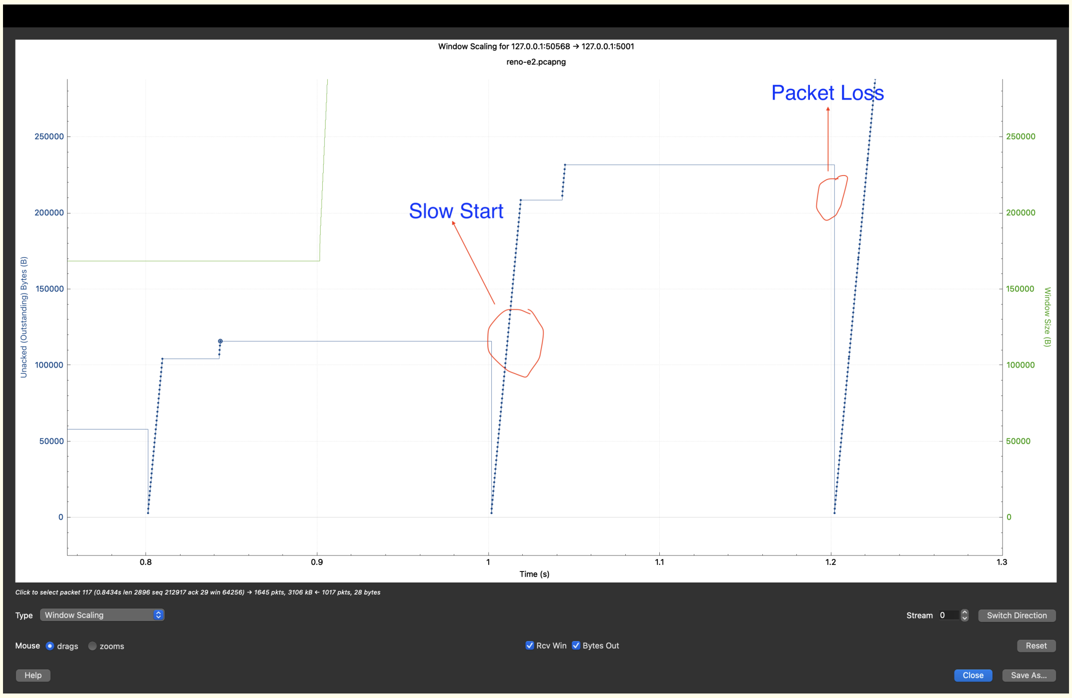


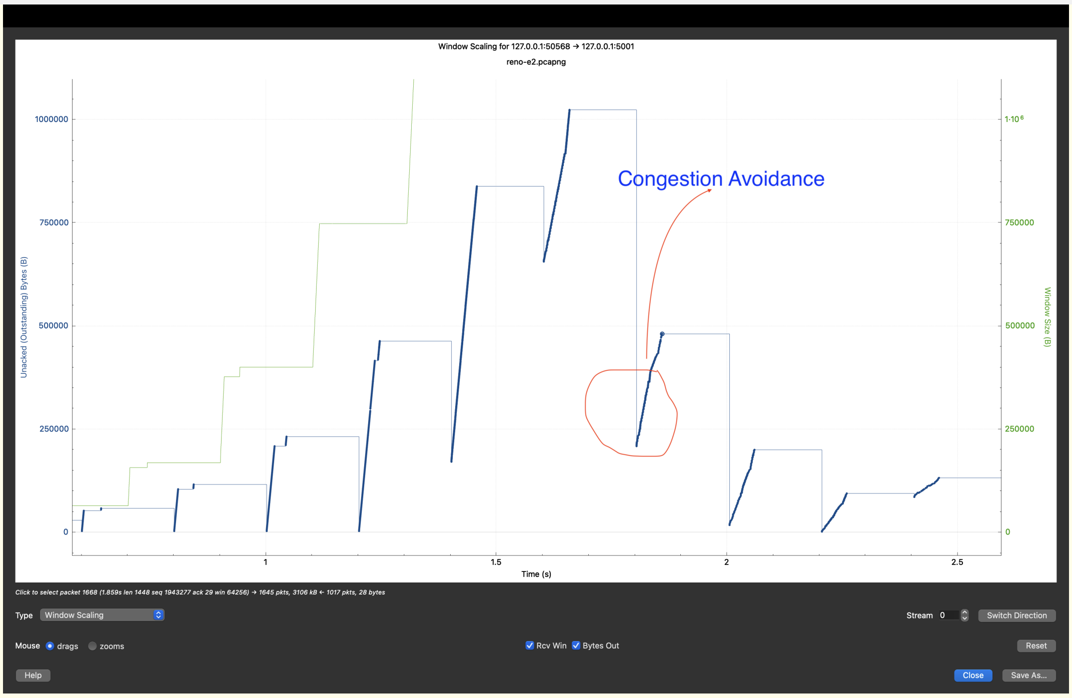


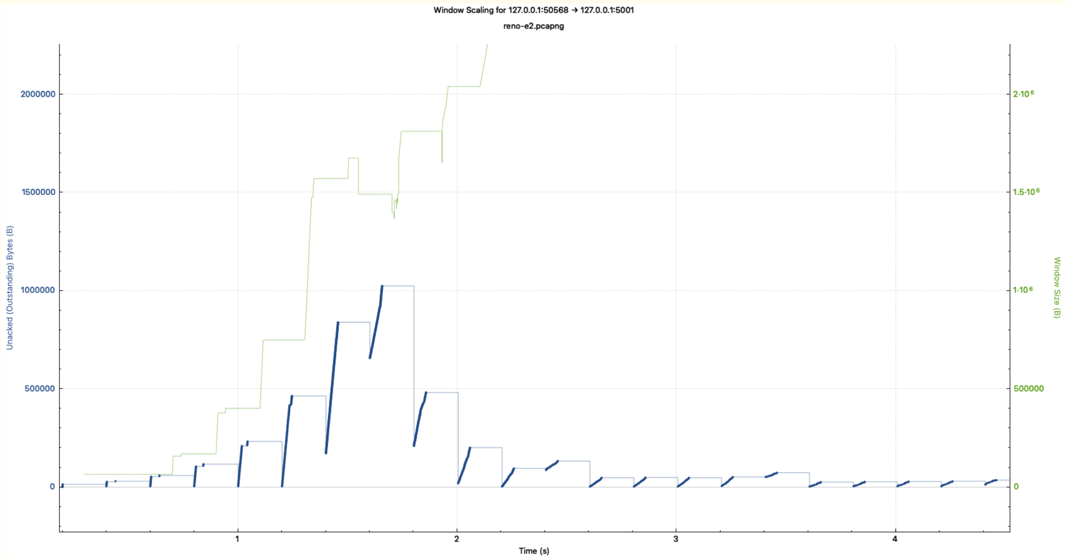
As we can see from the plots, slow start is from 1MSS, and exponentially increases. This is shown in the first image. Places where we see steep decrease in the window size correspond to packet losses. Places where we start from half the max congestion window size are congestion avoidance phases. Here is the complete graph:



Delay=100ms, Loss=1%

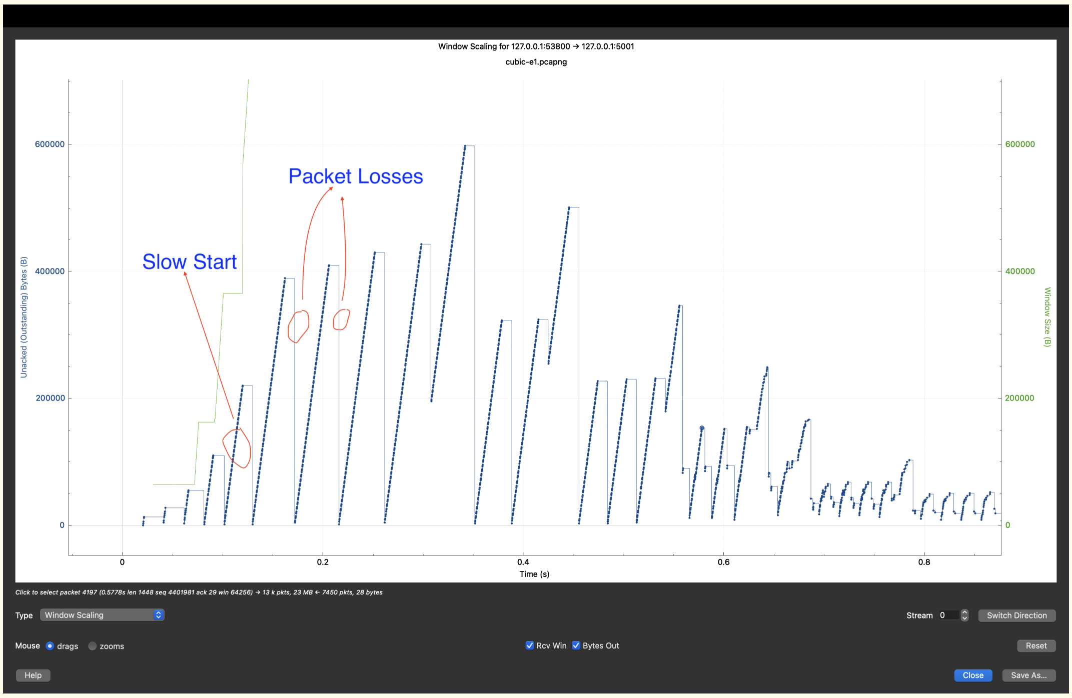


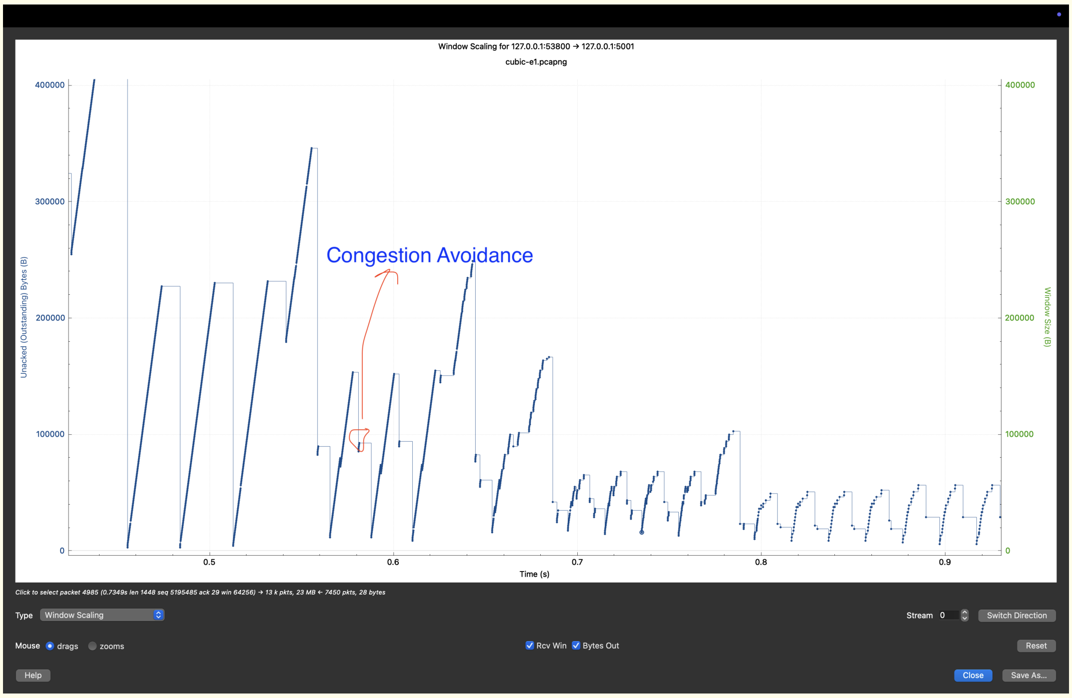

Window Scaling for 127.0.0.1:50568 → 127.0.0.1:5001
reno-e2.pcapng
Congestion Avoidance
Unacked (Outstanding) Bytes (B)
Window Size (B)
Time (s)
Type Window Scaling
Stream 0
Switch Direction
Mouse drags zooms
Rcv Win
Bytes Out
Reset
Help
Close
Save As...

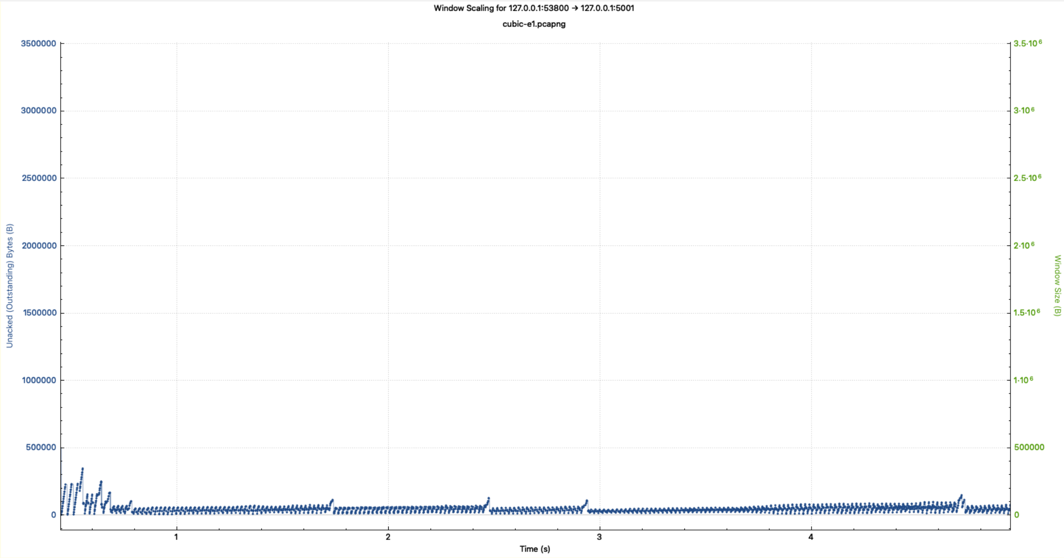

Window Scaling for 127.0.0.1:50568 → 127.0.0.1:5001
reno-e2.pcapng
Unacked (Outstanding) Bytes (B)
Window Size (B)
Time (s)

Cubic

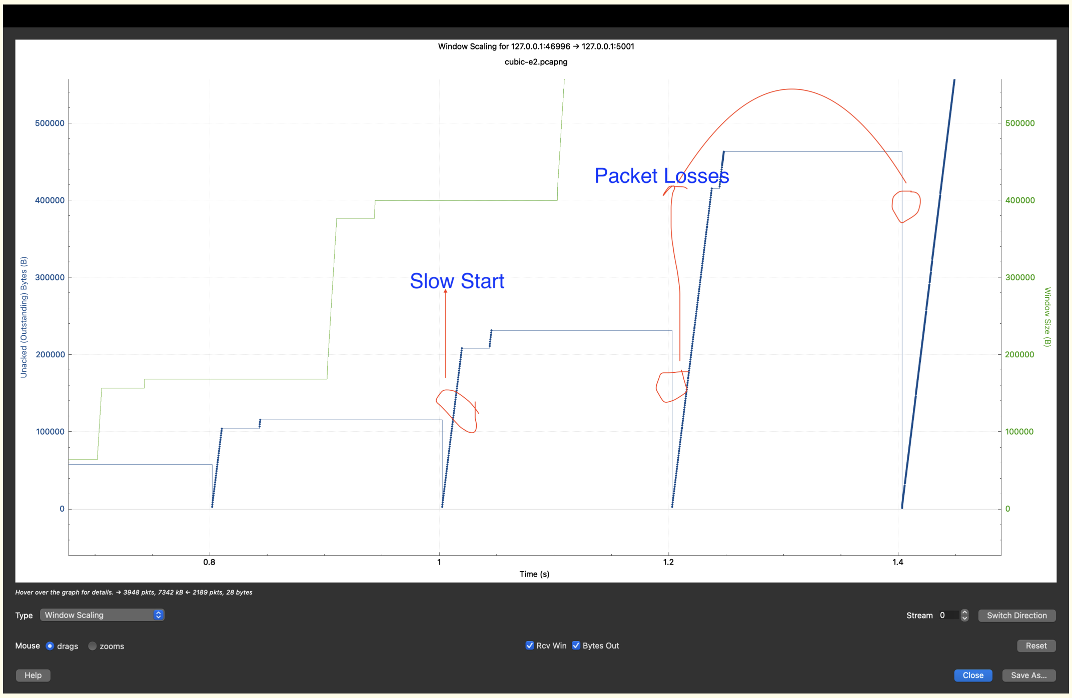
Delay=10ms, Loss=0.1%

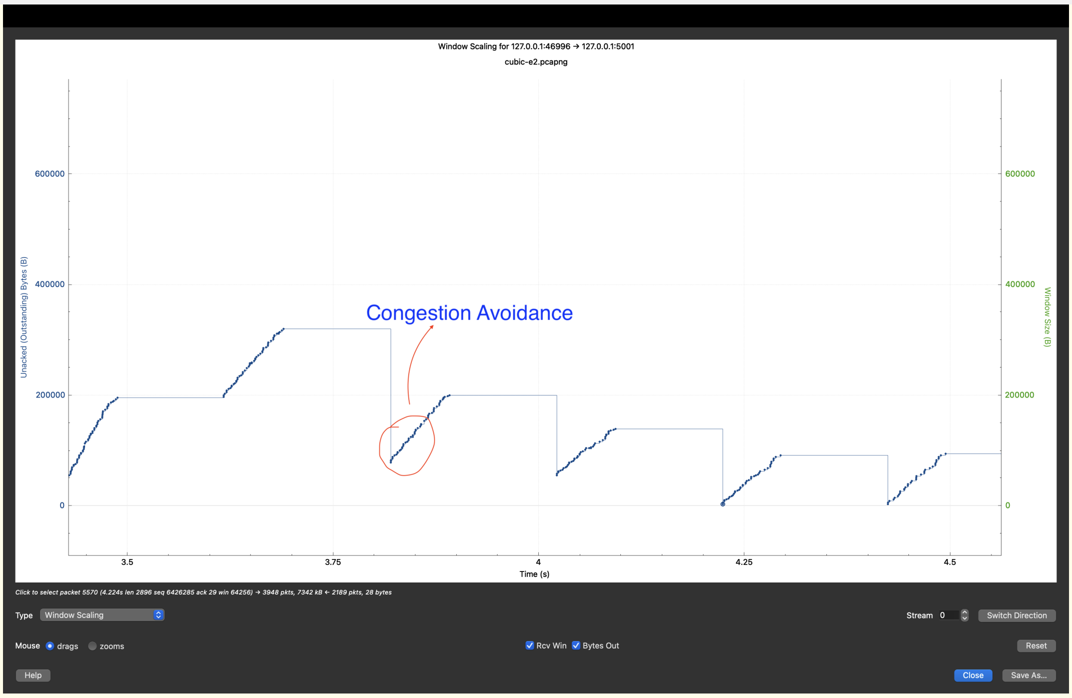


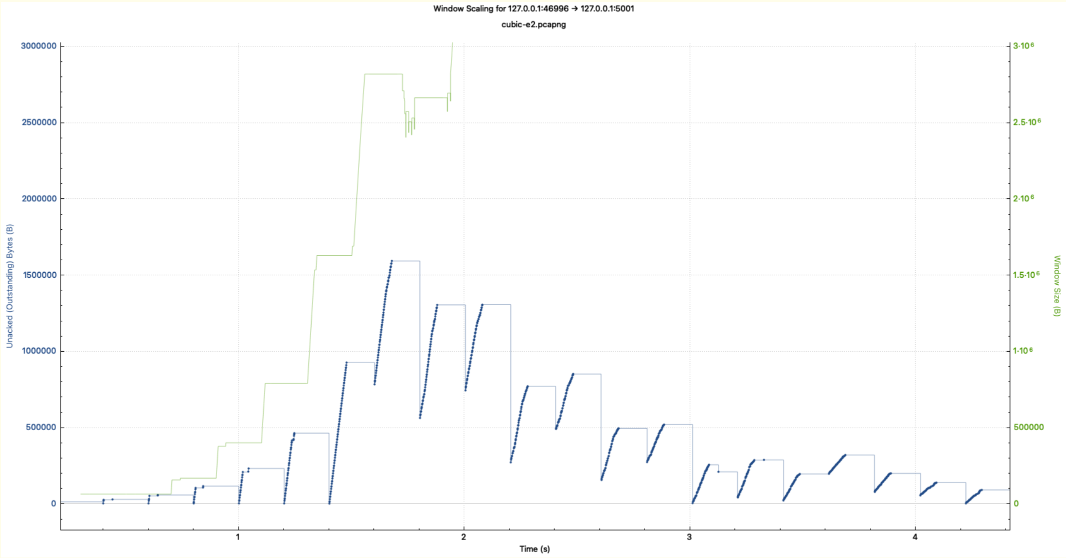




Delay=100ms, Loss=1%







- The plots in green are for window sizes, whereas the plots in blue are the unacked outstanding bytes (bytes in flight).
- The annotations are shown in red and blue.
- The window sizes are in general higher for cubic than for reno (specially after the initial section). This is because cubic is more aggressive in increasing the congestion window size.