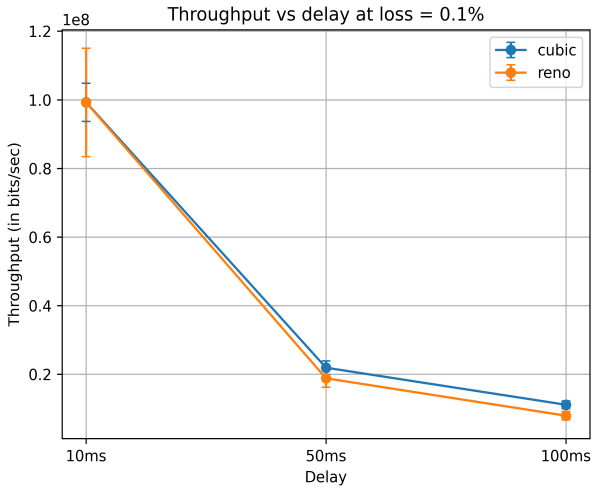
CS 252: Lab 6

Abhinav Gupta, Gurnoor Singh Khurana, Sambit Behera
190050003, 190050045, 190050104

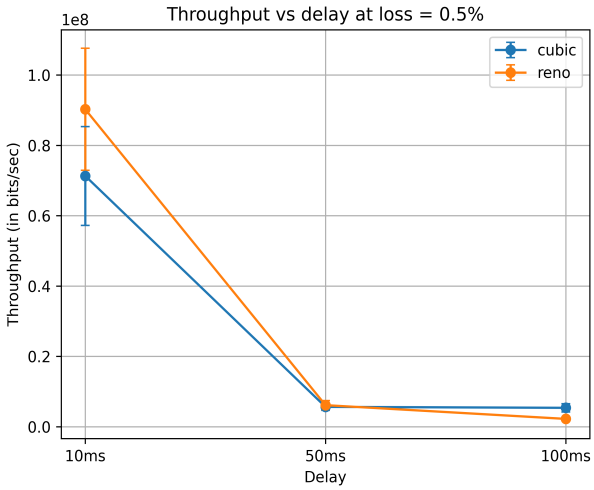
May 11, 2021

Results of Experiment

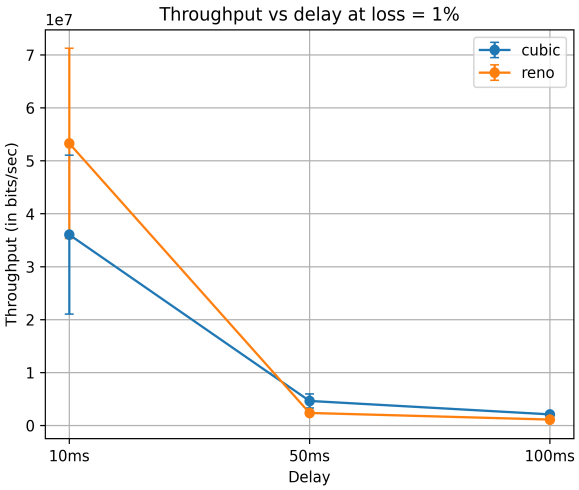
Plots



(a) Loss= 0.1%



(b) Loss= 0.5%

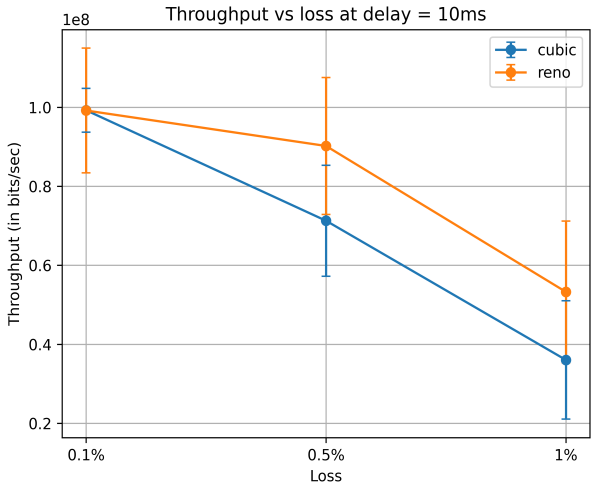


(c) Loss = 1%

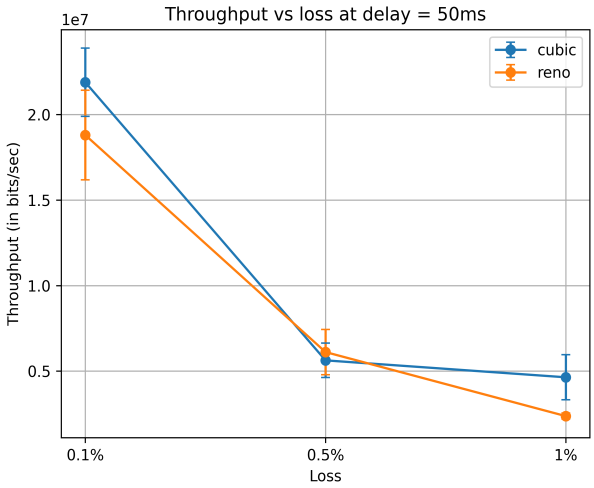
Figure 1: Plots Throughput vs delay with different losses

In this section, we vary the delay for certain values of packet loss. There is a clear decreasing throughput trend as the delay increases for both the TCP variants as more delay corresponds to higher Round Trip Time (RTT) and hence less throughput of data overall (because throughput equals window/RTT).

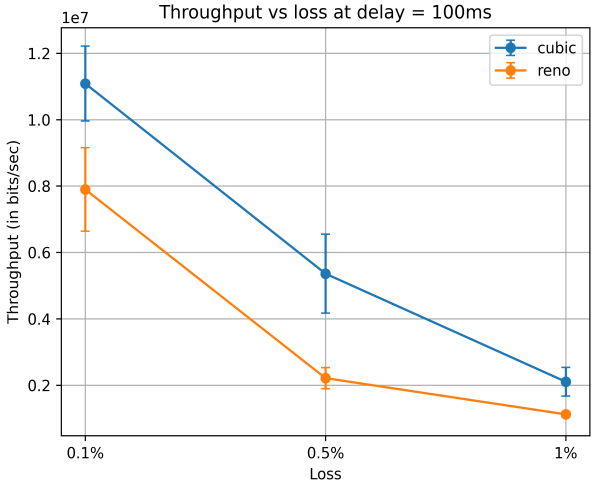
We can observe TCP Reno dominating TCP Cubic at lower values of delay, and both being similar at higher values, with cubic having slightly more throughput. This is because Reno increases its window size depending on RTT (in both Slow Start phase and CA phase). However, the window size of TCP Cubic increases only as a function of time elapsed from the last packet loss. Hence as delay increases, the CA phase of TCP Cubic becomes more aggressive than TCP Reno, leading to higher throughput.



(a) Delay= 10ms



(b) Delay = 50ms



(c) Delay = 100ms

Figure 2: Plots Throughput vs loss with different delays

In this section, we vary the loss for certain values of delay. There is a clear decreasing throughput trend for both the TCP variants in all three graphs. As the packet loss increases, there will be lesser number of packets delivered in a time period because of retransmitting old packets, reducing the overall throughput rate.
We observe TCP Reno to have lower throughput for higher delays and higher throughput for lower delays. This is due to the fact TCP Reno's window size increase is dependent on RTT, while for TCP Cubic, it is dependent on time elapsed since last packet loss. Hence at higher delays, TCP Cubic increases its window size more aggressively leading to higher throughput.

Cubic is supposed to be more aggressive than Reno in CA mode
From the above six plots, we can confirm that TCP Cubic is more aggressive than Reno in Congestion Avoidance. This is inferred by the fact that at higher delay values, the protocols are in CA mode and in these cases we have a higher throughput value for Cubic, because of faster increase in window

size(cubic).
On increasing delay, there is increase in the RTT value, and the width of window is increased after a higher period of time than before, resulting in lesser packets transmitted and hence lesser throughput values. (can also be seen by theoretical throughput value i.e. $throughput = window\_size/RTT$.
On increasing the percentage of loss, the protocol will now have to retransmit more number of packets that were already transmitted and hence decreasing the total amount of packets delivered. Therefore, there is a decrease in throughput value with increase in loss.

Annotated *Wireshark* window scaling graphs

In the following graphs, the height of blue points (the blue piles are separated by time equal to RTT) indicate the maximum unACKed frames sent by the receiver . Hence, also the current window size. At slow start, the window size (and hence height of blue piles) doubles every RTT, in CA window size increases linearly (in TCP Reno). At packet loss, the window size drops. Regions are marked as *slow start*, *packet loss* and *congestion avoidance* wherever we are in initial phase, we have packet loss or we enter the CA phase respectively.

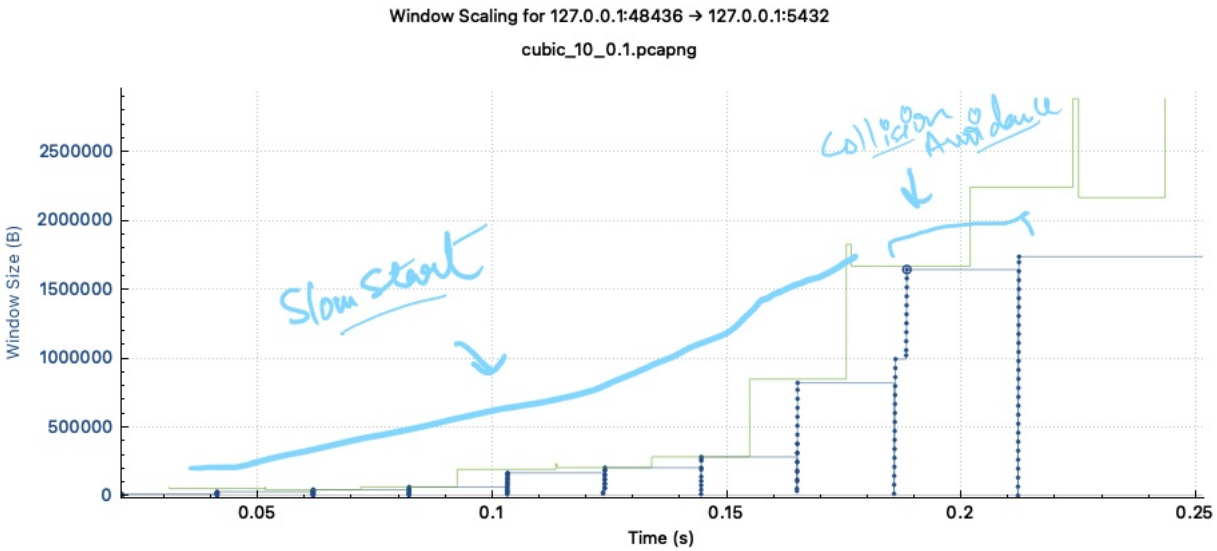


Figure 3: Window scaling graph for TCP cubic with 10ms delay and 0.1% loss

Note: In Figure 3, no significant loss was observed.

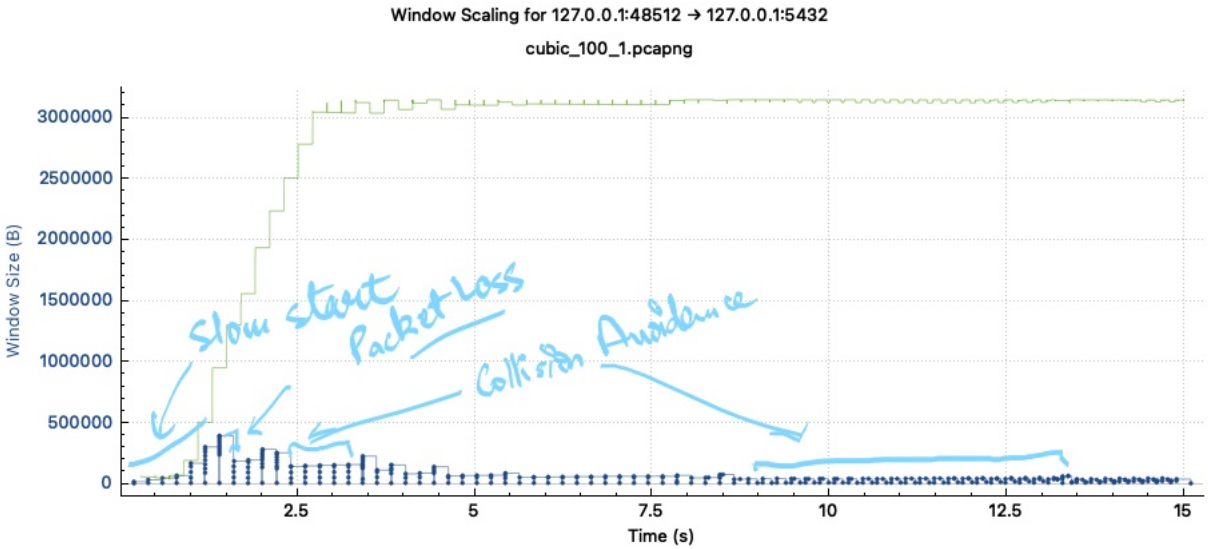


Figure 4: Window scaling graph for TCP cubic with 100ms delay and 1% loss

Note: In Figure 4, large portion was under the *collision avoidance* phase.

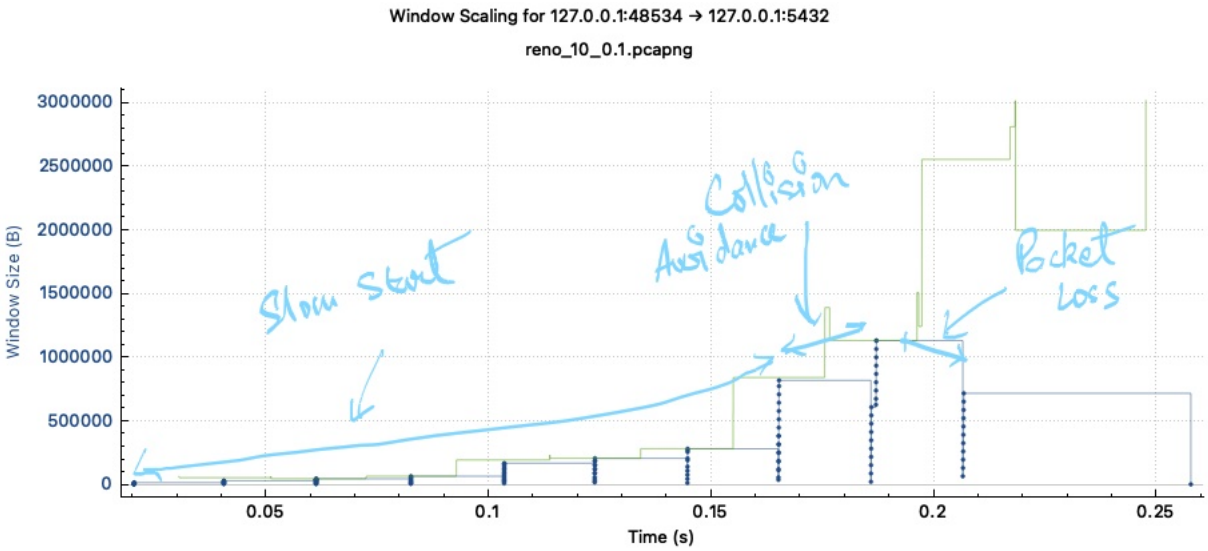


Figure 5: Window scaling graph for TCP reno with 10ms delay and 0.1% loss

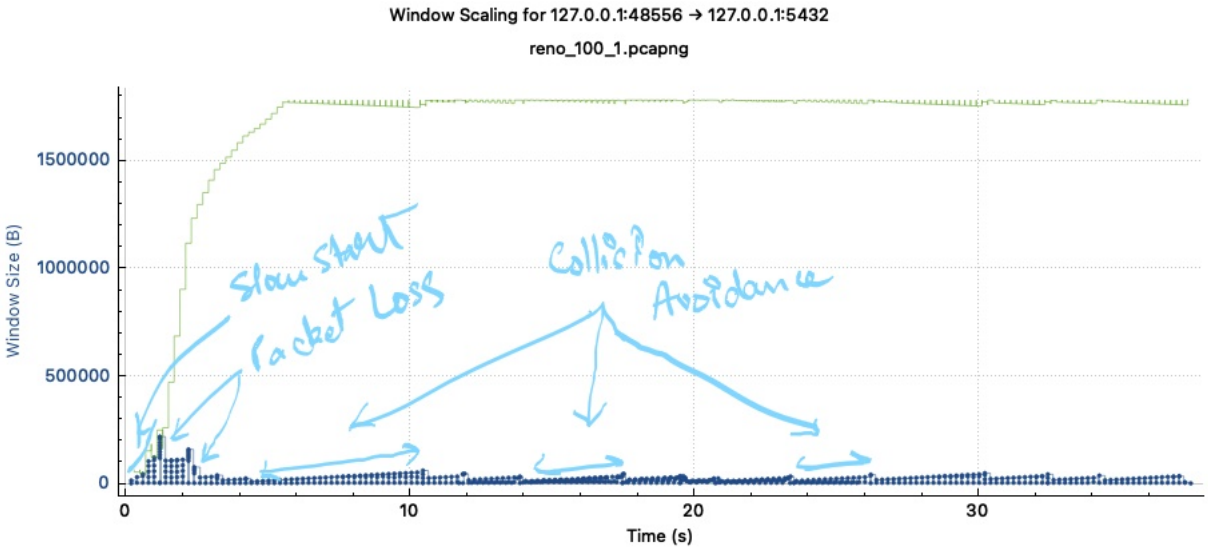


Figure 6: Window scaling graph for TCP reno with 100ms delay and 1% loss

Note: In Figure 6, again for most of the time, the connection was in *congestion avoidance* phase (all are not marked), and one can see the linear increase of window-size in those periods clearly.

Note: At some places there has been a typo, collision avoidance. It should instead be congestion avoidance.