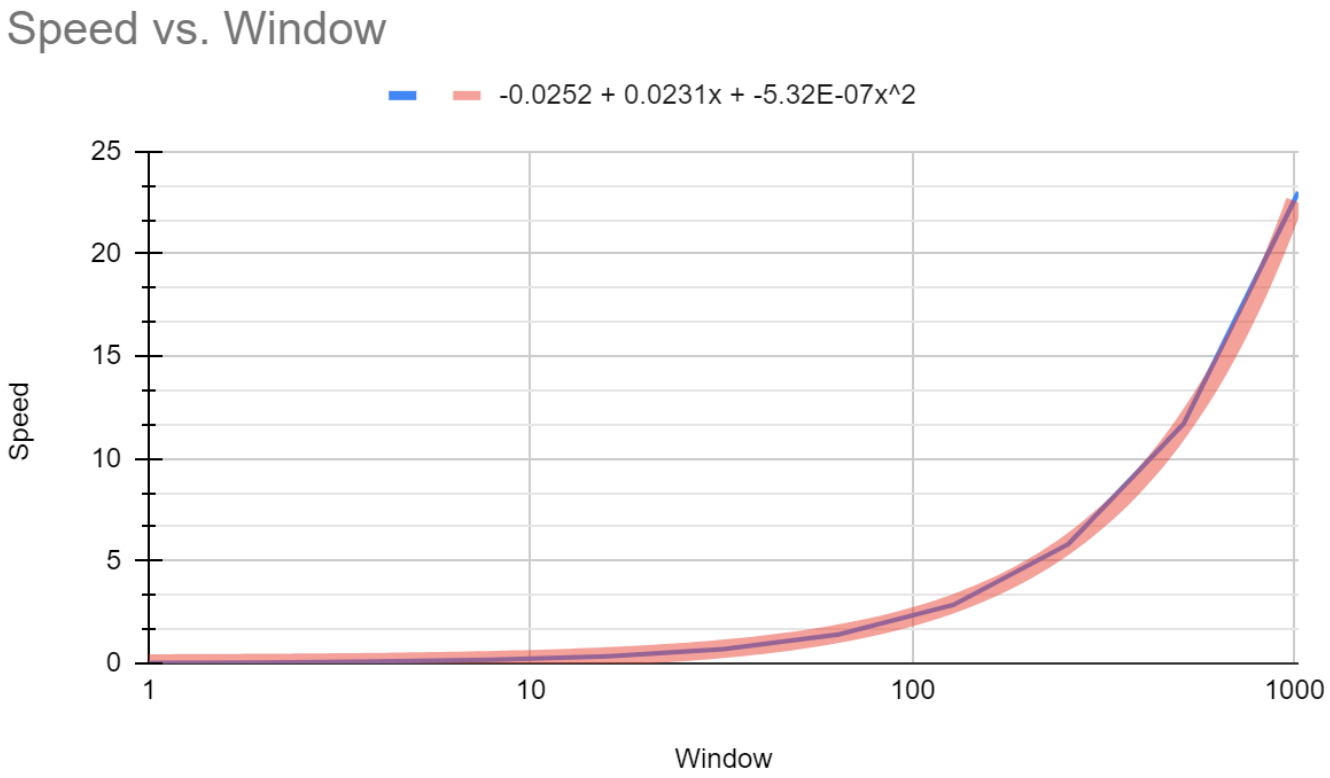
CSCE:612 Home 3-part 3 Report

Name - Abhishek Sinha
UID: 232003268

1.

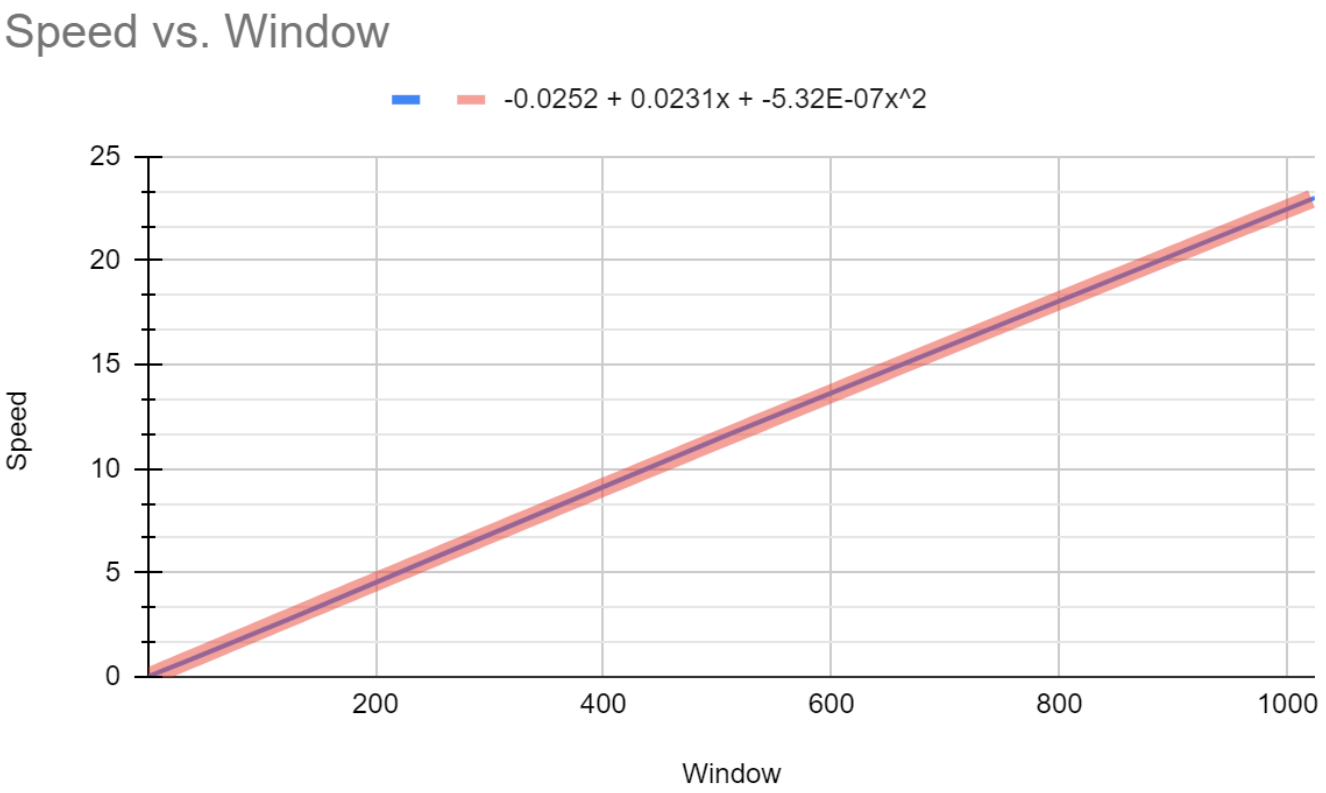
Window	RTT(ms)	Speed(Mbps)
1	0.5	0.022
2	0.5	0.043
4	0.5	0.088
8	0.5	0.176
16	0.5	0.348
32	0.5	0.701
64	0.5	1.412
128	0.5	2.855
256	0.5	5.8233



From the equation for curve fitting, since the leading term (x^2) has negligible value, thus the equation is approximately linear. This equation is also valid from the theory as for AIMD part of TCP,

$$r_{av} = \frac{\frac{3}{4} * W * MSS}{RTT}$$

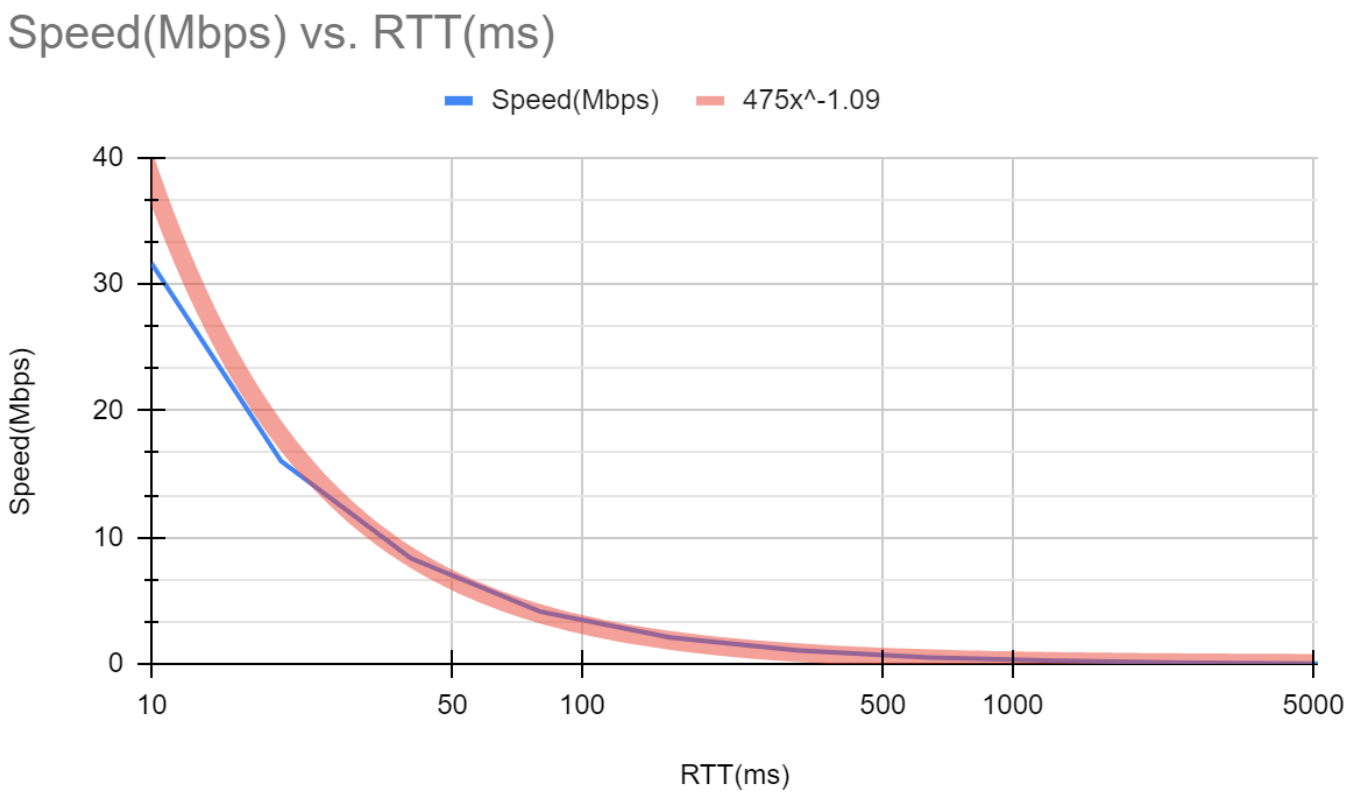
Thus rate is proportion to windowSize given all other parameters are kept constant or $r_{av} \alpha W$, which is evident from the graph. Note: The graph has a log scale X axis, thus looks more of an exponential or polynomial curve. Keeping both the scales linear gives the following graph:



The above graph clearly shows that the dependence is linear.

2.

Window	RTT(ms)	Speed(Mbps)
30	10	31.6298
30	20	16.008
30	40	8.346
30	80	4.126
30	160	2.094
30	320	1.04028
30	640	0.50539
30	1280	0.241172
30	2560	0.094064
30	5120	0.024993



The above plot again reconfirms the equation that

$$r_{av} = \frac{\frac{3}{4} * W * MSS}{RTT}$$

The only difference being r_{av} is inversely proportional to RTT. This is evident from the trendline of the graph.

Where the equation is $r_{av} = \frac{475}{RTT^{1.09}}$ or $r_{av} \alpha \frac{1}{RTT}$

3. My CPU has the following configurations:
   Name : AMD Ryzen 5 4600H
   # CPU cores : 6
   # Threads : 12
   Frequency : 3Ghz
   Turbo Boost to 4Ghz

In both the settings, I was not able to achieve 1 Gbps. My speed got limited to around 650Mbps with the link speed at 1Gbps and reaches to approx 700 Mbps with link speed set to 10 Gbps. The aforementioned speeds were achieved with a window size of 9000. However, trying the same code on ts2.cse.tamu.edu remote PC virtual machine, increased the speed to 750 Mbps, that too while communicating on the network (with s3.irl.cs.tamu.edu) and not with the dummy receiver. Thus, I believe my CPU's base frequency has quite a significant role to play. Below is the trace for the dummy with my computer.

(Note: Traces are present in the home zip file itself with names such as Gbps_1_win_9000.txt. The name is self explanatory)

4.With buffer size 223 DWORDs, RTT = 200 ms, window size W = 300 packets, link capacity S = 10 Mbps and and loss only in the reverse direction equal to p = 0.1, the code should not and did not expect any significant change in the sending rates. It is because when there is loss on the return path, it means only

timeout occurs, not duplicacy of ACKs on the path or packet being dropped altogether. This is also evident from the traces.

(Note: Traces are present in the home zip file itself with names such as no_reverse_loss.txt and reverse_loss.txt. The names are self explanatory).

5. The algorithm in the receiver increases the capacity and advertises the same according to the number of packets received. For example if the sequence received is 80 then it would advertise a window size of 81. In TCP this is known as slow start. However, the receiver window grows to a max of 80000 or 80k before saturating to that value.