Congestion Control Part 2

Brandt Elison
Joe Eklund

March 31, 2016

1 Introduction

The purpose of our project was to verify our implementation of TCP congestion control by testing using a series of experiments. We divided our experiments into two categories: basic and advanced. The basic experiments tested the performance of our implementation of TCP congestion control by varying the number of flows that concurrently transfer a 1 MB file over a single link between two hosts. The advanced experiments also manipulated the link congestion, but they altered the network topology and the way TCP responded to loss events.

2 Basic Experiments

For our basic experiments we created a two host network with the following parameters:

- 10 ms - propagation delay
- 100 packets - queue size
- 10 Mbps - transmission rate
- 500,000 bytes - intitial slow start threshold

Each of the experiments uses a different number of flows, meaning a different number of connections transfering 1 MB files simultaneously on the link.

2.1 One Flow

Figure 1: The graph of our one flow receiver's rate over time.

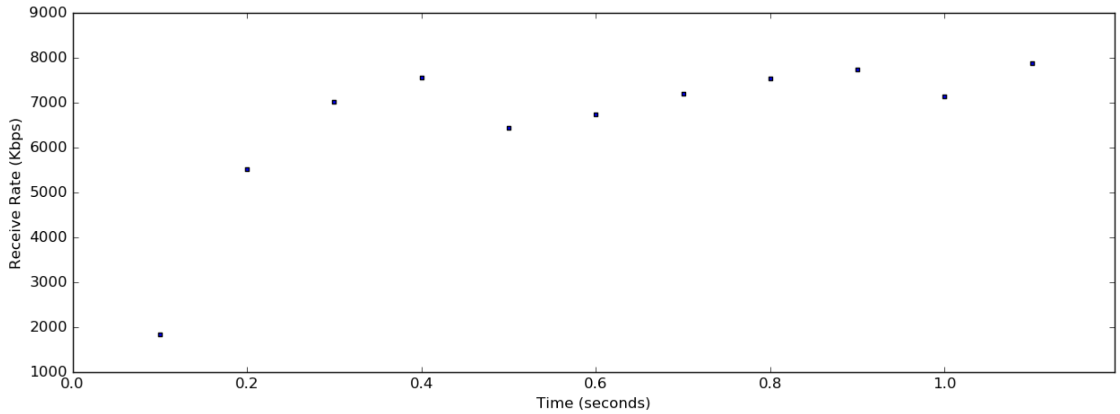


Figure 2: The graph of our one flow queue size over time.

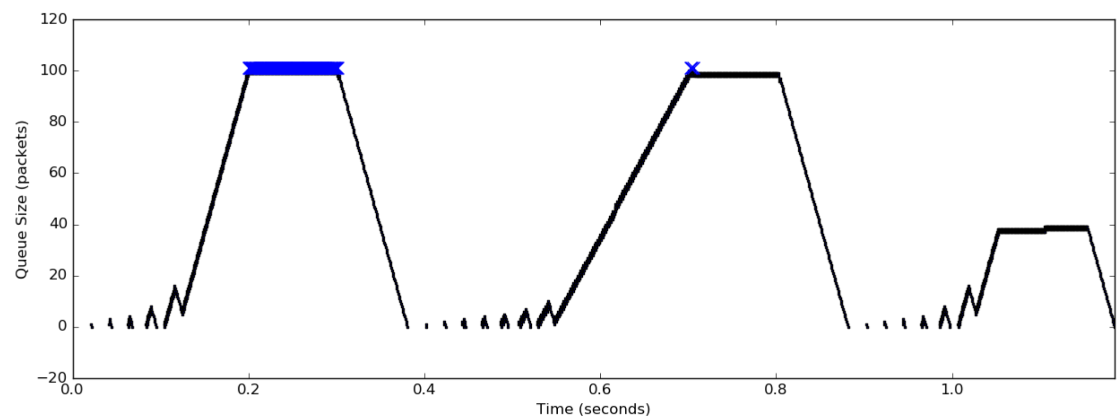


Figure 3: The graph of our one flow congestion window size over time.

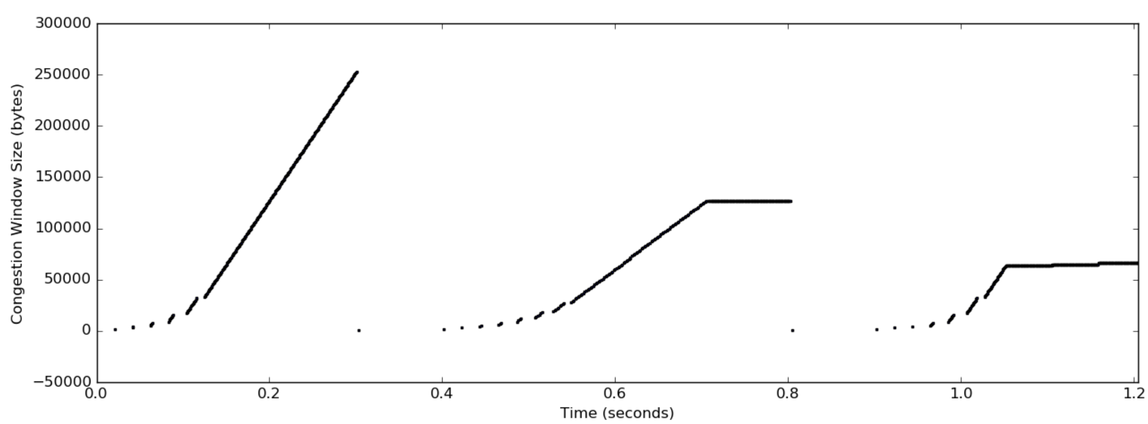


Figure 4: The graph of our one flow sequence plot.

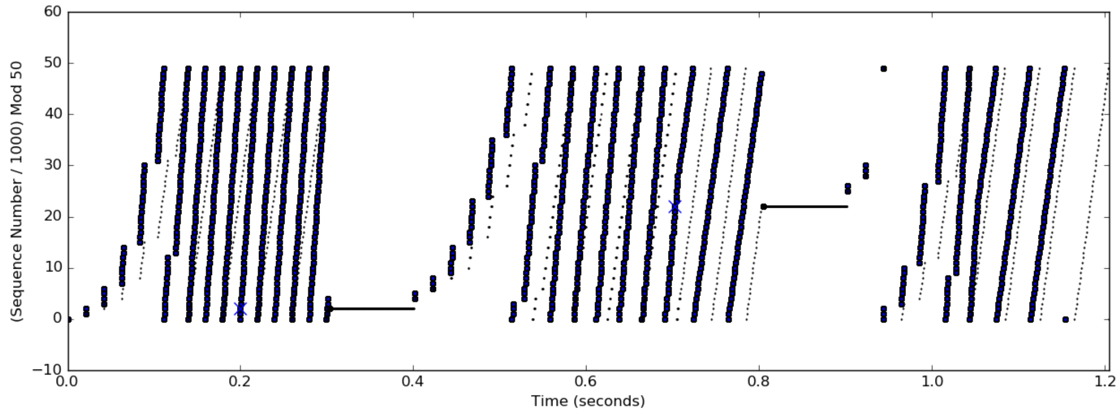


Figure 1 shows how the receive rate changes over time for a single flow. We can see that the receive rate goes up very quickly and levels off; this is expected behavior because the flow has access to the full bandwidth. In Figure 2 we see how the queue size is quickly filled during slow start because the threshold is much larger than the queue size. However after some loss, congestion control helps keep the queue size under its max of 100 packets. If the file that we transfered was larger than 1 MB, we would see the additive increase section at around 1.1 seconds continue without loss for a long time. Figure 3 supports Figure 2 by showing the typical saw tooth pattern that reflects TCP's attempt to find a congestion window size appropriate for the link.

2.2 Two Flow

Figure 5: The graph of our two flow receivers' rates over time.

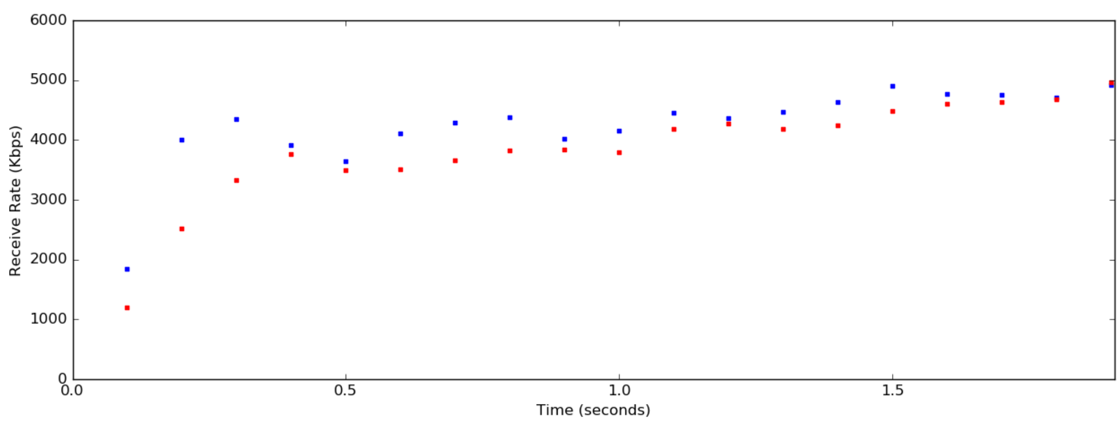


Figure 6: The graph of our two flow queue size over time.

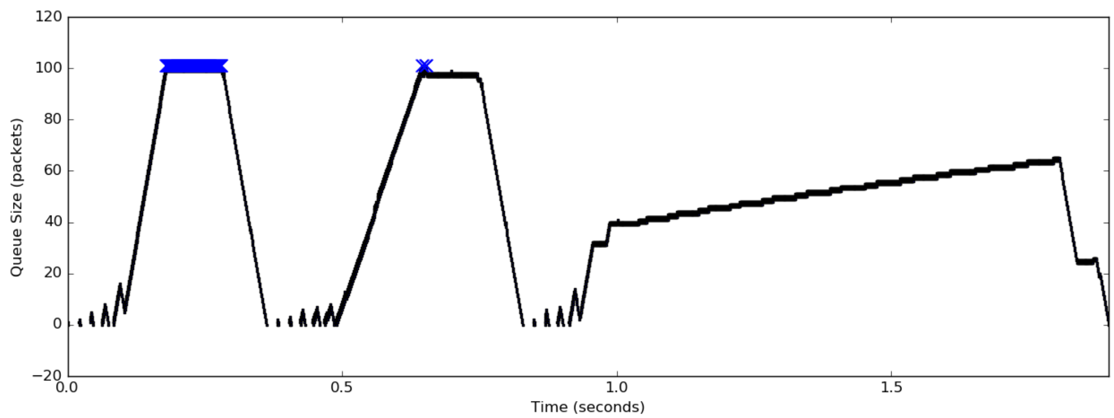


Figure 7: The graph of our two flow congestion window size over time for flow A.

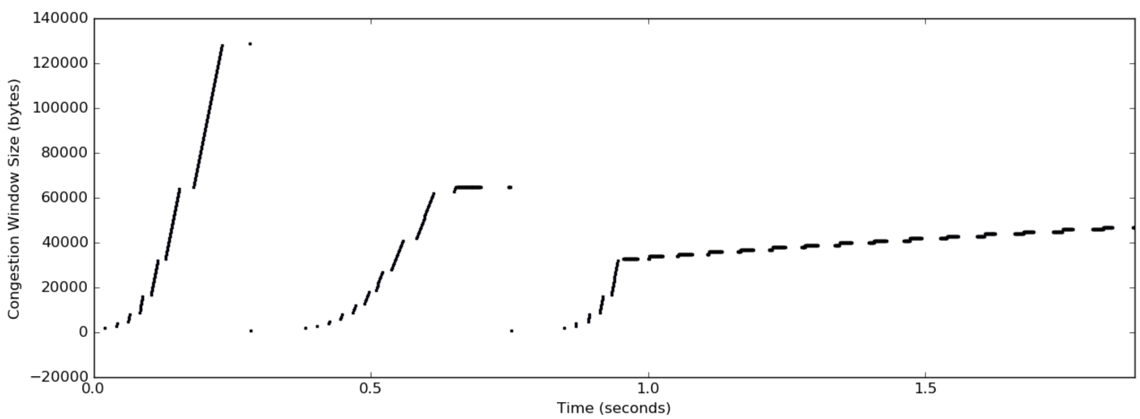


Figure 8: The graph of our two flow congestion window size over time for flow B.

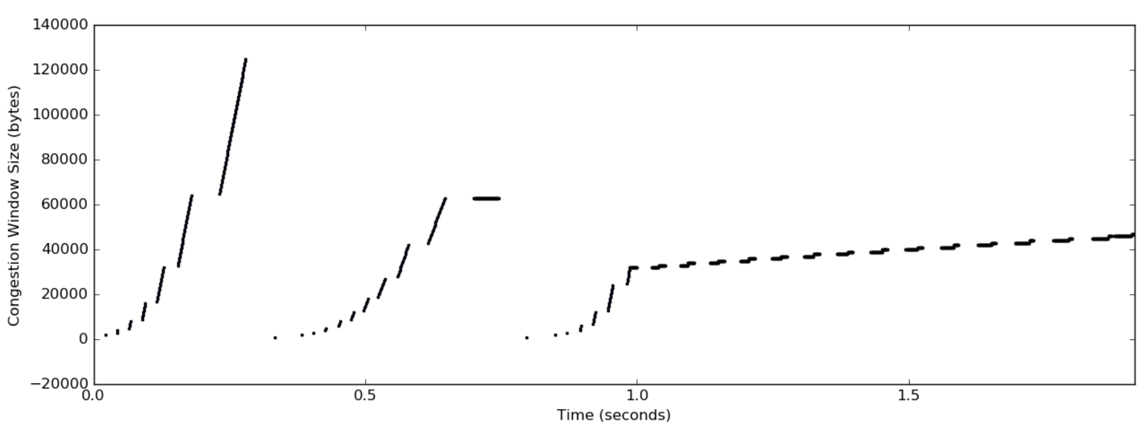


Figure 9: The graph of our two flow sequence plot for flow A.

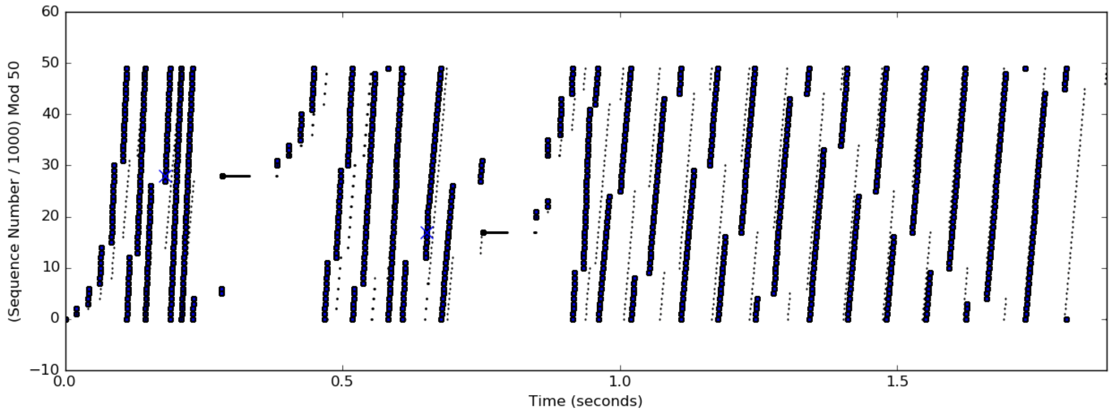
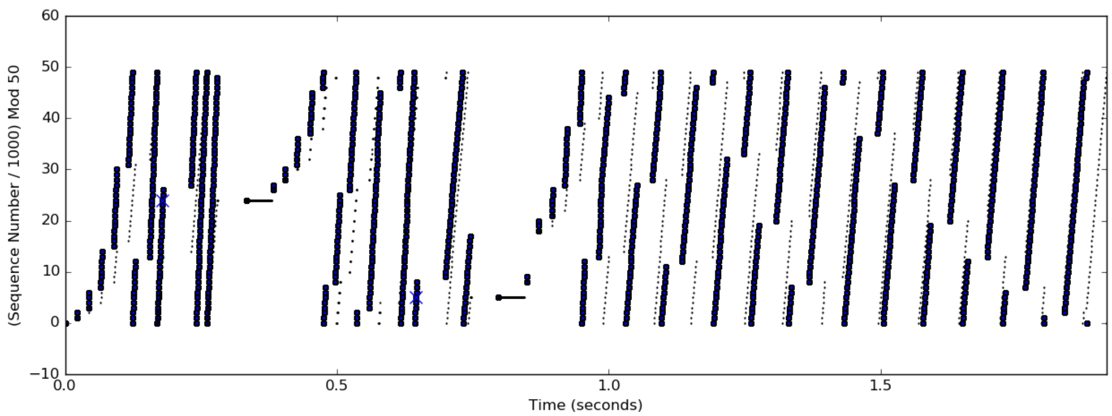


Figure 10: The graph of our two flow sequence plot for flow B.



Because the two flows in this experiment were identical, we would expect both data series in Figure 5 to look nearly identical. Figure 5 shows this behavior. We would also expect Figures 7 and 8 to look nearly identical because both flows experience loss at nearly the same time and adjust their windows to the same values. Figures 7 and 8 do look nearly identical as expected. Figure 6 also demonstrates expected behavior; this is clear particularly when you compare Figure 6 with Figure 2. On both graphs we see two peaks loss events occuring while TCP is still identifying a good window for the link. After the second peak both flows start additive increase before the queue is filled. However, in Figure 6 the slope of the increase in queue size during the additive increase period is steeper than the slope in Figure 2 because Figure 6 shows two flows, both doing additive increase.

2.3 Five Flow

Figure 11: The graph of our five flow receivers' rates over time.

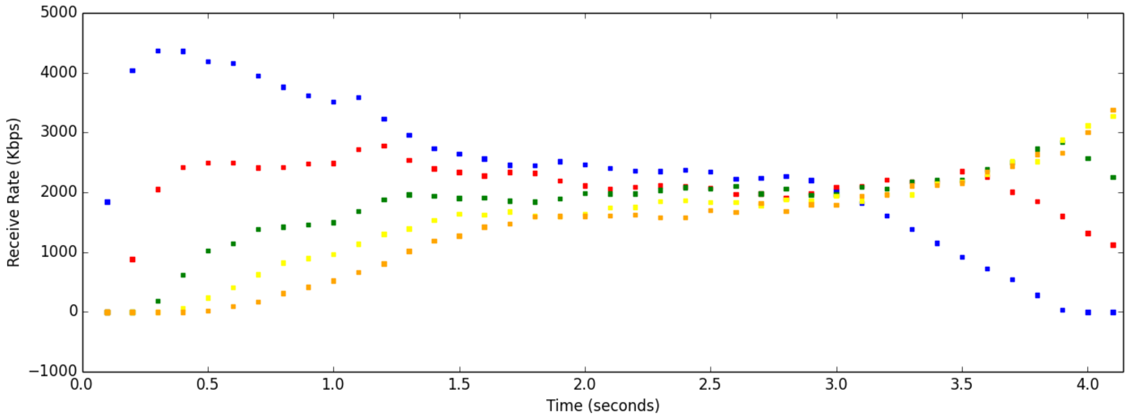
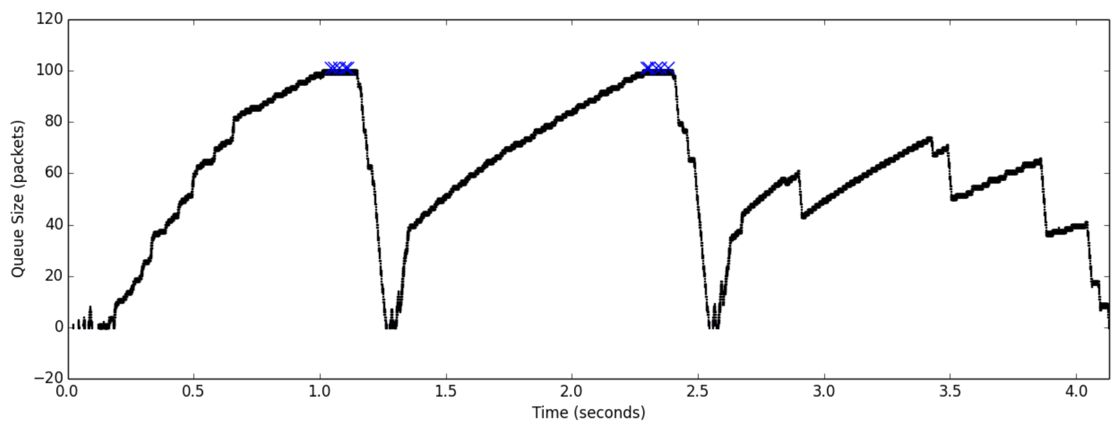


Figure 12: The graph of our five flow queue size over time.



For five flow we decided a smaller threshold size of 16,000 bytes. We chose to do this because with such a small file there was not enough time for the TCP connections to balance their share of the link before flows started finishing their file transfer. Five flow also differed from one and two flow because each flow began transfering 0.1 seconds after the preceeding flow (i.e. 0.0, 0.1, 0.2, 0.3, 0.4).

In a correct implementation of TCP, the first flow to begin transferring will immedietely take advantage of all the bandwidth. As more connections begin sharing the link the receive rate of the initial flows should drop while the receive rate of the newer flows should increase until they are roughly the same. This means that they are equally sharing the bandwidth. As early flows finish transferring their files the remaining flows start to take advantage of the newly available bandwidth. Figure 11 is evidence of this behavior.

3 Advanced Experiments

3.1 Additive Increase - Additive Decrease (AIAD)

Figure 13: The graph of our one flow receiver's rate over time with AIAD.

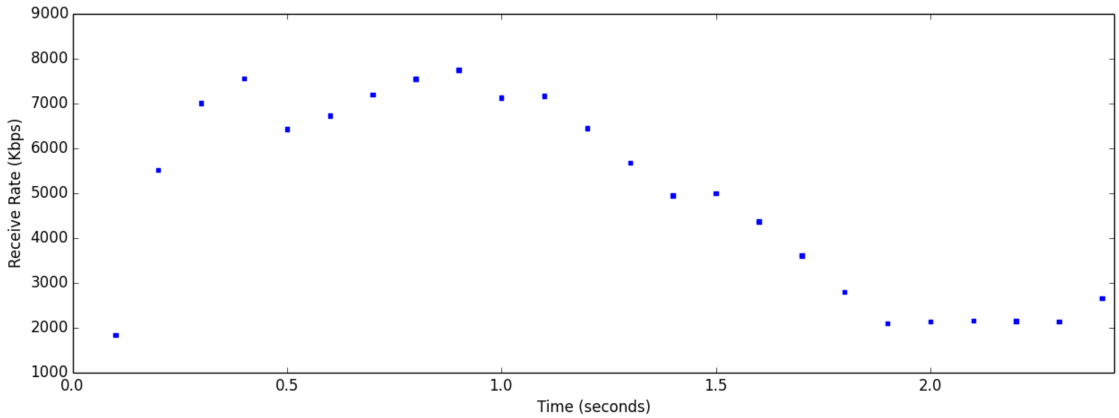
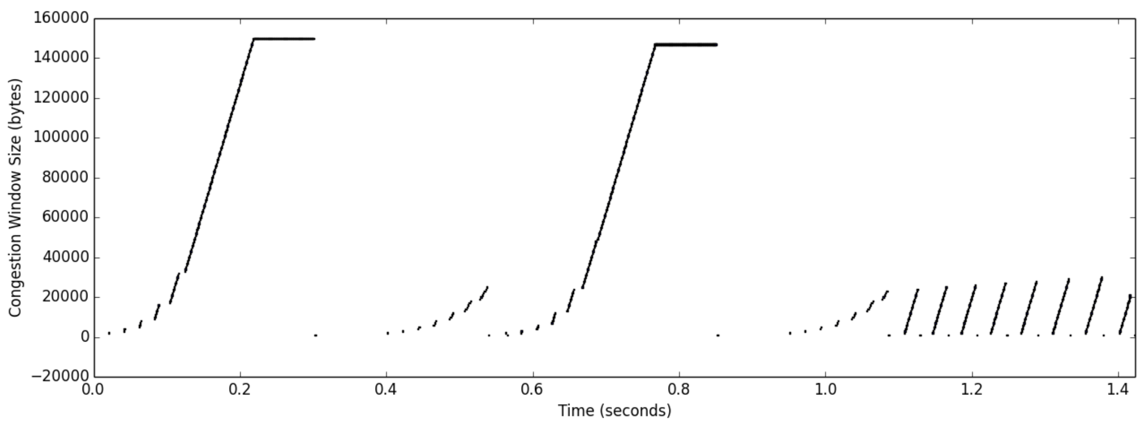


Figure 14: The graph of our one flow congestion window size over time with AIAD.



discuss AIAD here

3.2 Additve Increase - 5/6 Multiplicative Decrease (AIMD)

Figure 15: The graph of our one flow receiver's rate over time with a 5/6 multiplicative decrease.

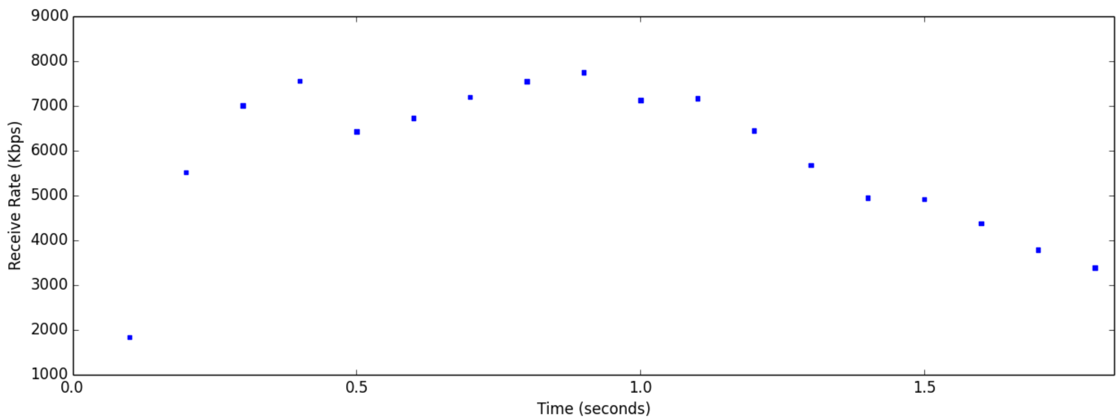
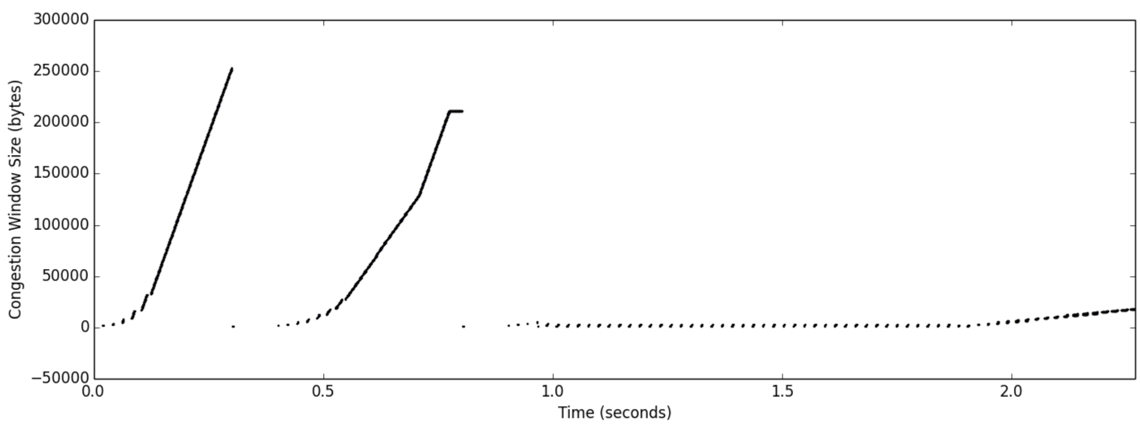


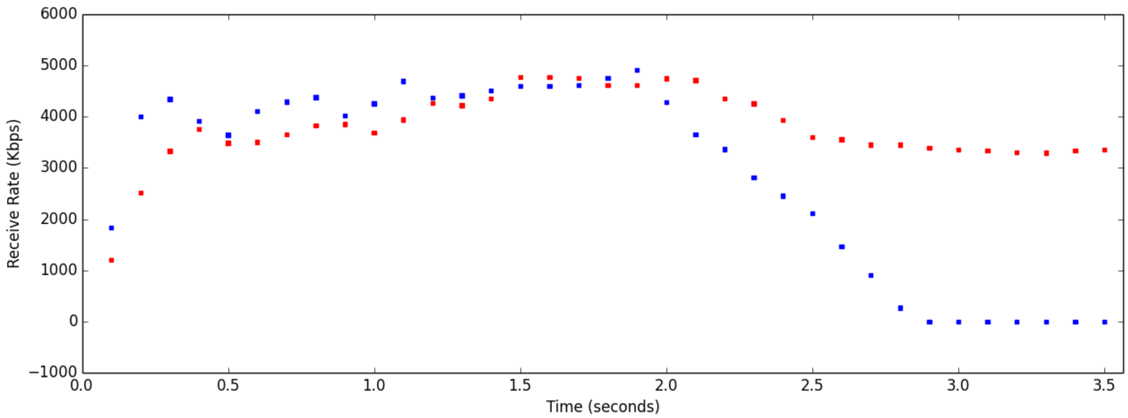
Figure 16: The graph of our one flow congestion window size over time with a 5/6 multiplicative decrease.



discuss AIMD here

3.3 Competing AIMD

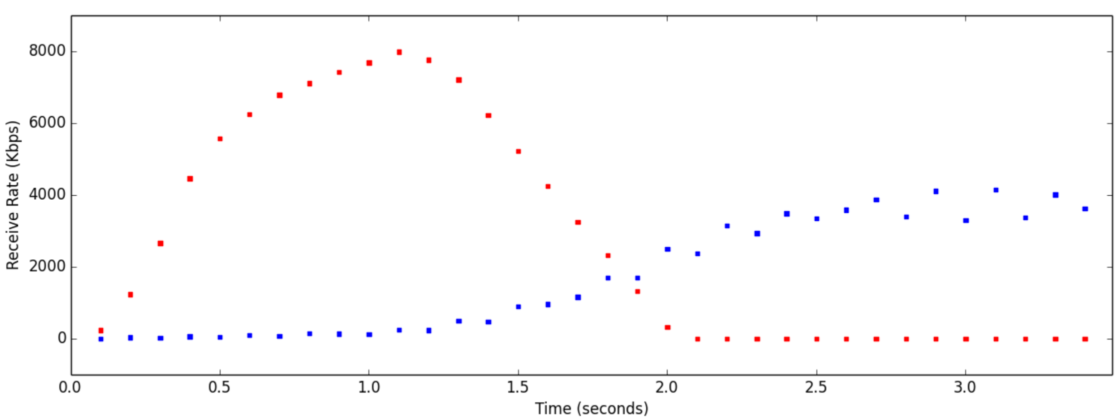
Figure 17: The graph of our two flow receivers' rates over time with competing AIMD.



discuss competing AIMD here

3.4 Competing Round Trip Time (RTT)

Figure 18: The graph of our two flow receivers' rates over time with competing RTT.



discuss competing RTT here