CS1680: TCP Congestion Control

Jason Crowley

Spring 2022

Abstract

In the original RFC, the TCP protocol was proposed only with simple flow control to ensure reliable, in-order delivery between hosts. Implementations typically use a sliding window of in-flight data segments to increase throughput and transmit all available data as quickly as possible. However, too many hosts transmitting data at the same time can cause the network to quickly become congested, leading to severe packet loss and degraded performance. Congestion control is a concept that enables TCP implementations to detect packet loss and adjust transmission rates accordingly. For my capstone, I implemented the TCP Tahoe congestion control algorithm, which uses *slow start*, *congestion avoidance*, and *fast retransmit* heuristics to improve performance in congested networks. Additionally, I compared the performance of sending large files using our TCP implementation with and without the congestion control algorithm both in drop-free and lossy networks. Though, from my testing, performance was actually worse using the congestion control algorithm, I suspect that this is because packets in our virtual network are dropped at random and not from network congestion.

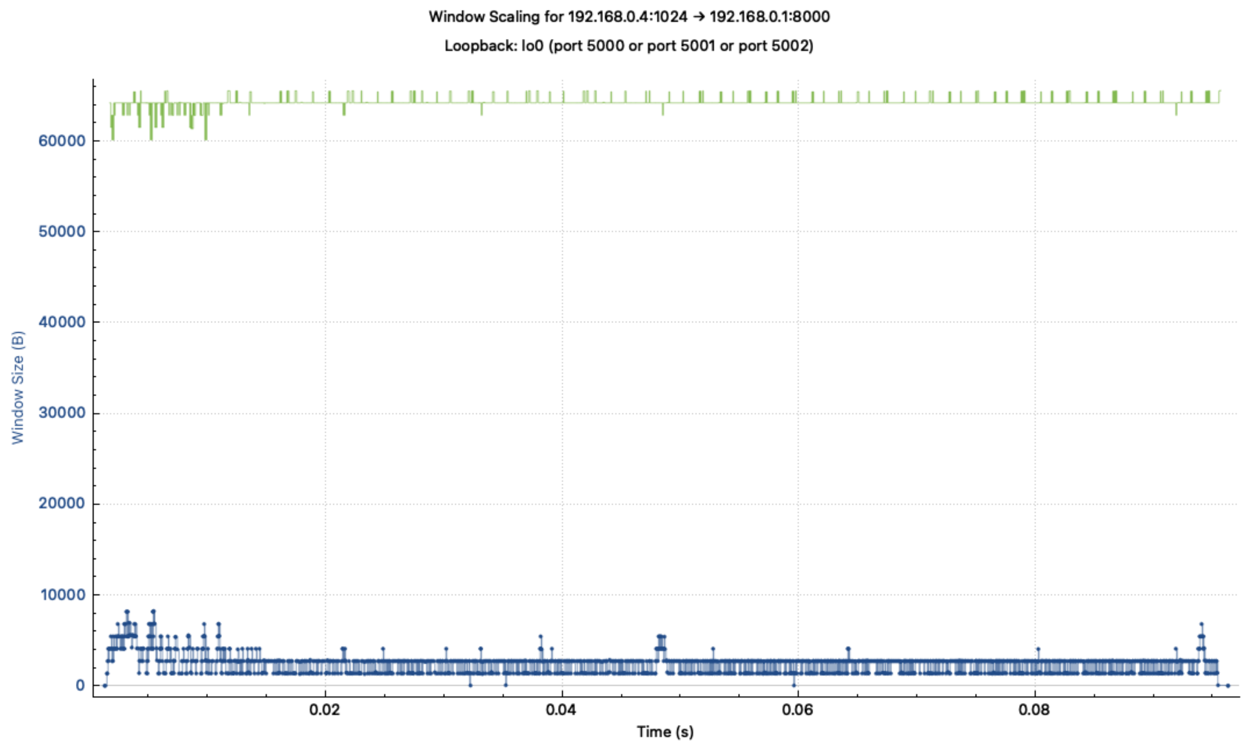

Figure 1: bytes in-flight over time in a drop free network with no added delay

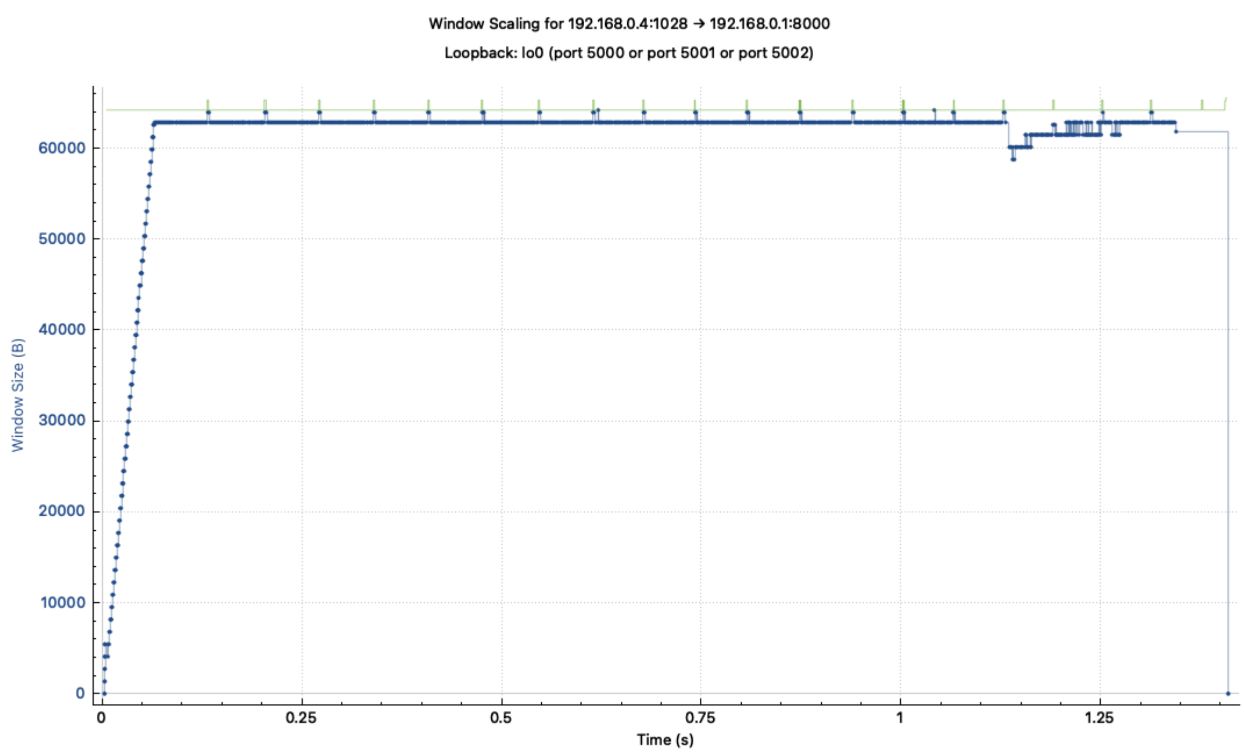

Figure 2: bytes in-flight over time in a drop free network with 1ms delayed responses

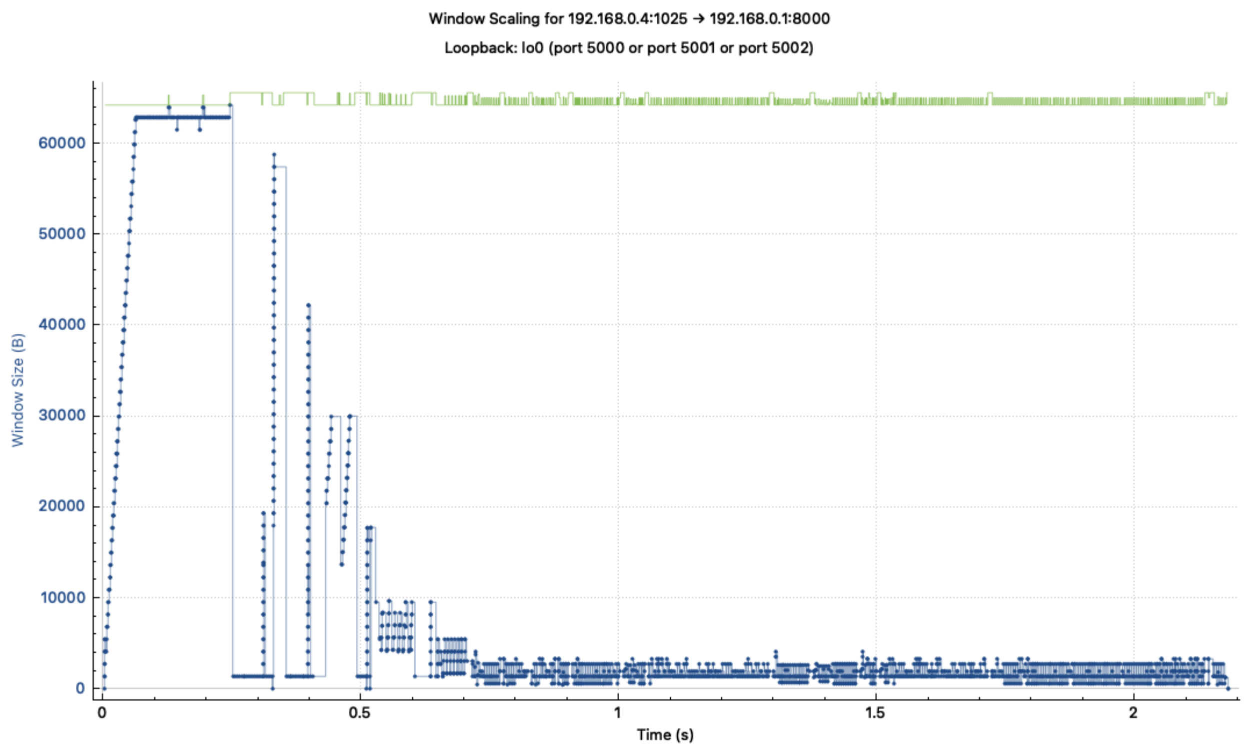

Figure 3: bytes in-flight over time in a lossy network (2% drop rate) with 1ms delayed responses