TCP Taheo & Reno Throughput Simulations
Project 1

Jonathan Jones #903012648

February 8, 2016

1. Introduction

The results from my ns-3 simulations do not match the expected results. I'm not exactly sure why this occurs. To keep things simple, the final goodput results analyzed here are from the expect results and are not the results of my actual *p1.cc* program.

For verification of the topology, the round trip times from the source to sink nodes is shown in figure 1 and briefly touched on in the next section.

2. Simulated Topology & Results Dataset

The network topology that was simulated consisted of fours nodes and three data links. Two TCP variants, TCP Tahoe & TCP Reno, were simulated across the topology. A total of 192 test cases were simulated that varied the tcp variant in use, number of flows, advertised window size, queue size of the bottleneck link, and segment size.

The delay time for the two outermost data links was $10ms$, while the delay time for the bottleneck link was $20ms$. In figure 1, we can verify the constructed topology since the average round trip time appears to be $80ms$ $(2 \cdot [10+20+10])$.

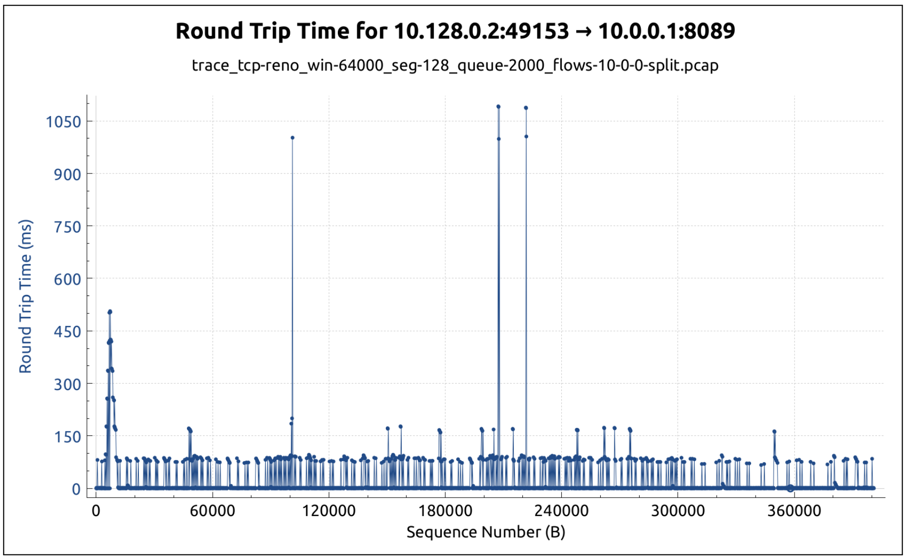

Figure 1: Round trip time across the topology.

3. Goodput for 10 Simultaneous TCP Flows

For when there are 10 simultaneous TCP flows, we can divide the results into three subsets according to the simulated segment size. The following section contains the results of those three subsets when the segment size was 128, 256, and 512 bytes respectively.

3.1 Isolating Segment Size

In figure 2, we can see that Tahoe does not provide the same fairness that we see from Reno. We can see that most of the highest goodput flows are using Tahoe; However, we can also see that the a majority of the lowest goodput flows are also using Tahoe. This trend seems to be amplified as the queue size is increased. For the minimum window size of 2000 bytes, Tahoe and Reno

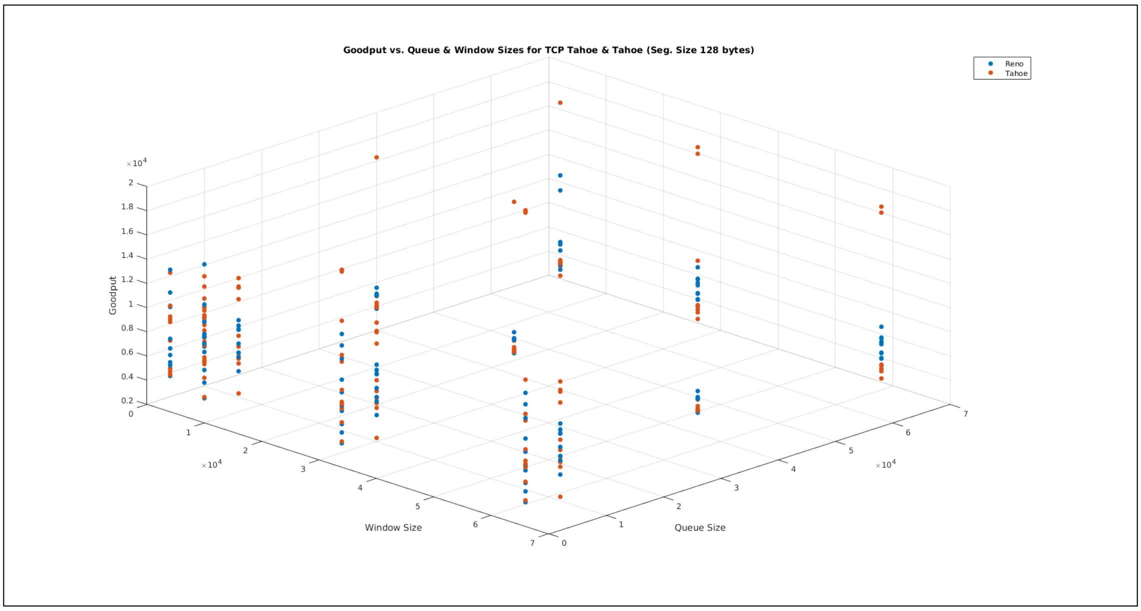

Figure 2: Goodput vs. windows size and queue size when segment size is 128 bytes.

From figure 3, an interesting observation can be made when the queue size is 32000 bytes at a 256 byte segment size. That is, for both TCP variants across all window sizes, a higher level of fairness is seen across all of the TCP flows. This is possibly a result of the queues becoming saturated before the window size limits the data flows.

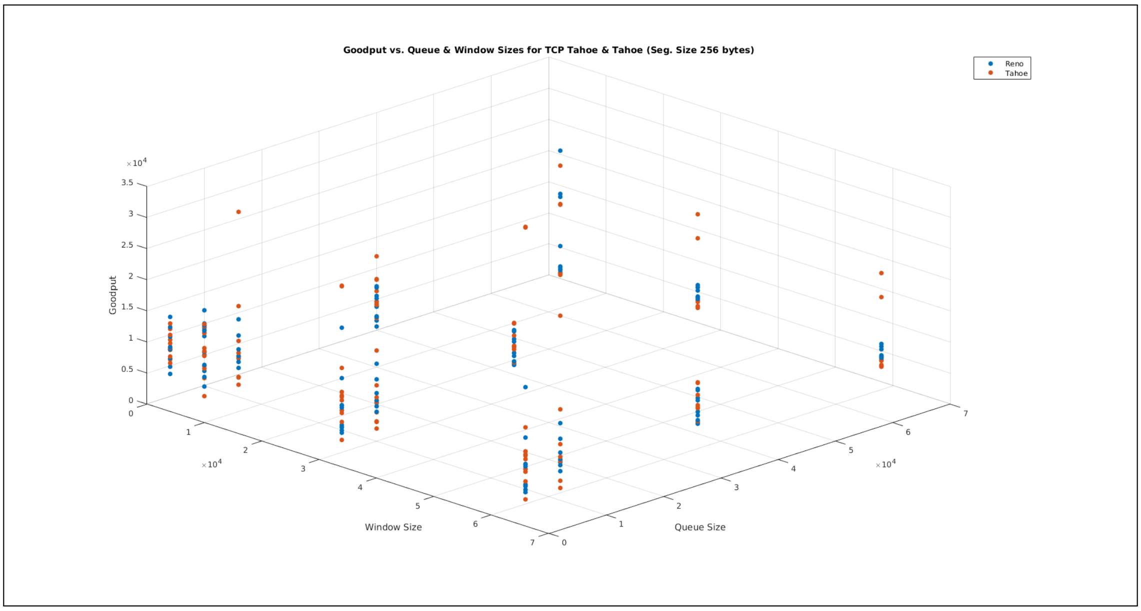

Figure 3: Goodput vs. windows size and queue size when segment size is 256 bytes.

In figure 4, when the segment size is 512 bytes, we generally see the same results from the previous subsets. However, an interesting observation can be made when the window size is 2000 bytes. This is emphasized in figure 5, and it shows that the TCP variants show no significant difference across flows. This is likely a result of the higher segment size, so the window size is quickly saturated. This means that the queue size becomes a non-critical factor, and that new segments are not sent until space becomes available in the receiving window.

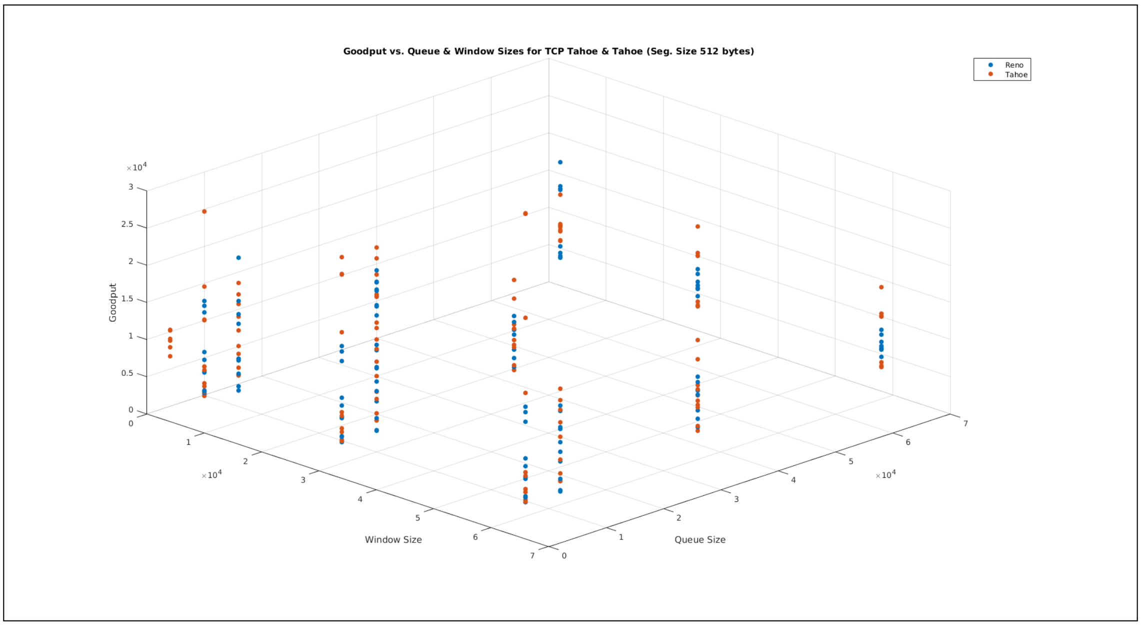

Figure 4: Goodput vs. windows size and queue size when segment size is 512 bytes.

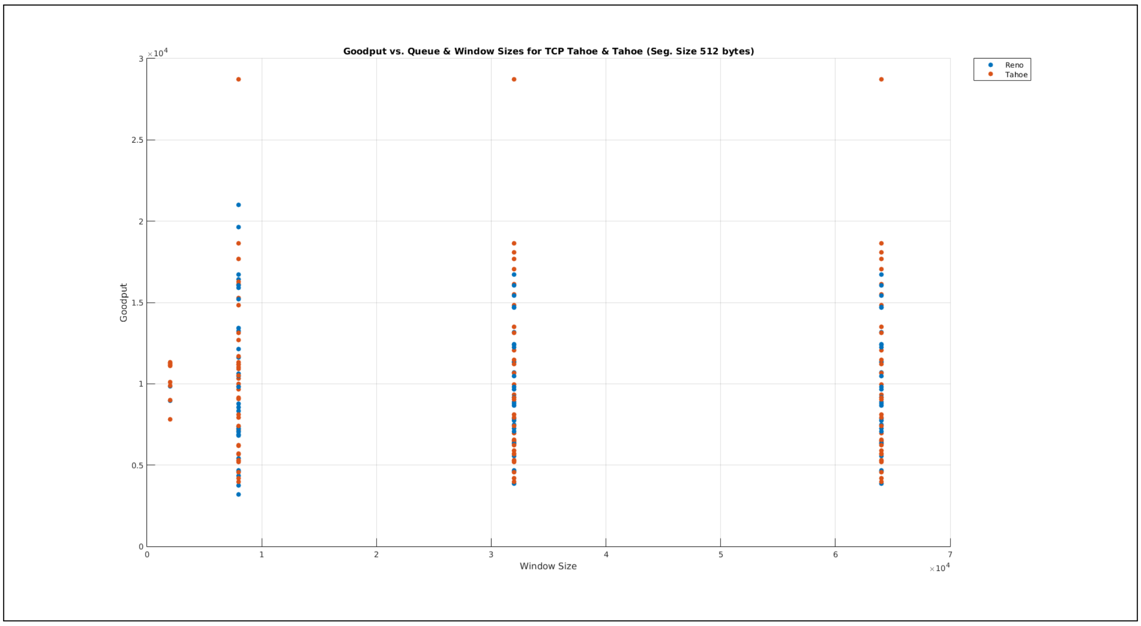

Figure 5: Goodput vs. queue size when segment size is 512 bytes.

3.2 Guaranteed Fairness

With a tightly controlled simulation environment, we can see what parameters result in optimal fairness. However, it becomes difficult to determine how the same set of parameters would react to different types of network traffic. As an example, TCP flows that enter and exit the network at random times - each consisting of a random amount of data sent over the network.

To answer the question - could we guarantee fairness for a network that performed exactly like our simulations? For the most part, yes. However, we could not guarantee the same level of fairness outside the scope of the simulated network traffic and topology.