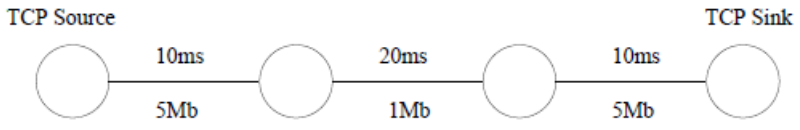
ECE 6110 – CAD for Communication Networks
Lab 1 – TCP Throughput Measurements

By Yash Shah

I. Introduction

A simple TCP connection was set up with a client (source node), two routers and a server (sink node). Figure 1 shows the network topology. The client sends infinite amount of data to the server over three links. The two outside links have 10 ms propagation delay and a bandwidth of 5 Mbps. The link connecting the two routers is a bottleneck with a 20 ms propagation delay and a 1 Mbps bandwidth.



Topology for Project 1

The overall topology can also be altered to create a Dumbbell topology with n-source and sink nodes through the use of the nFlows command line argument.

II. TCP Single-Flow Throughput

Using Network Simulator 3 this topology was constructed and simulated. During the simulation the throughput was measured as a function of the receiver window size, the bottleneck queue size and the segment size. Also, TCP-Tahoe protocols was used as the primary TCP protocol for this simulation. Based on a general understanding of TCP communication the expected behavior of the network is as follows:

- As receiver window size increases, goodput should increase.
- As source segment size increases, goodput should increase given a large enough queue size.
- As queue size increases, goodput should increase only if the segment size is close to the queue size.
- The ideal throughput would be approximately the bandwidth of the bottleneck, which is 1 Mbps * 1 byte/8 bits = 125000 bytes/sec.

Figure 2 shows how the goodput varied as a function of segment size and queue size, with each different segment size being represented by a different color. It is evident from the figure that network goodput using TCP-Tahoe protocol generally increased as the queue size increased. With smaller segment sizes, goodput was inversely proportional to queue size, but with larger segment sizes, as queue size increased so did goodput. This is most likely due to the fact that smaller segment sizes enable the sender to get closer to the queue and window size limits. But when the queue size is larger enough, smaller segment sizes result in more packets sent and thus create longer round-trip times than larger segment sizes would, and thus are more prone to timeouts.
There are also slight observable differences with the changes in window size. For window size of 2000 the goodput is the least for the different runs while for other window values the goodput mostly remained

the same when the queue size was fixed. This is most likely due to the fact that for a given simulation the goodput is optimal for a max threshold of window size. Any further increase in the window size does not affect the goodput beyond this threshold.

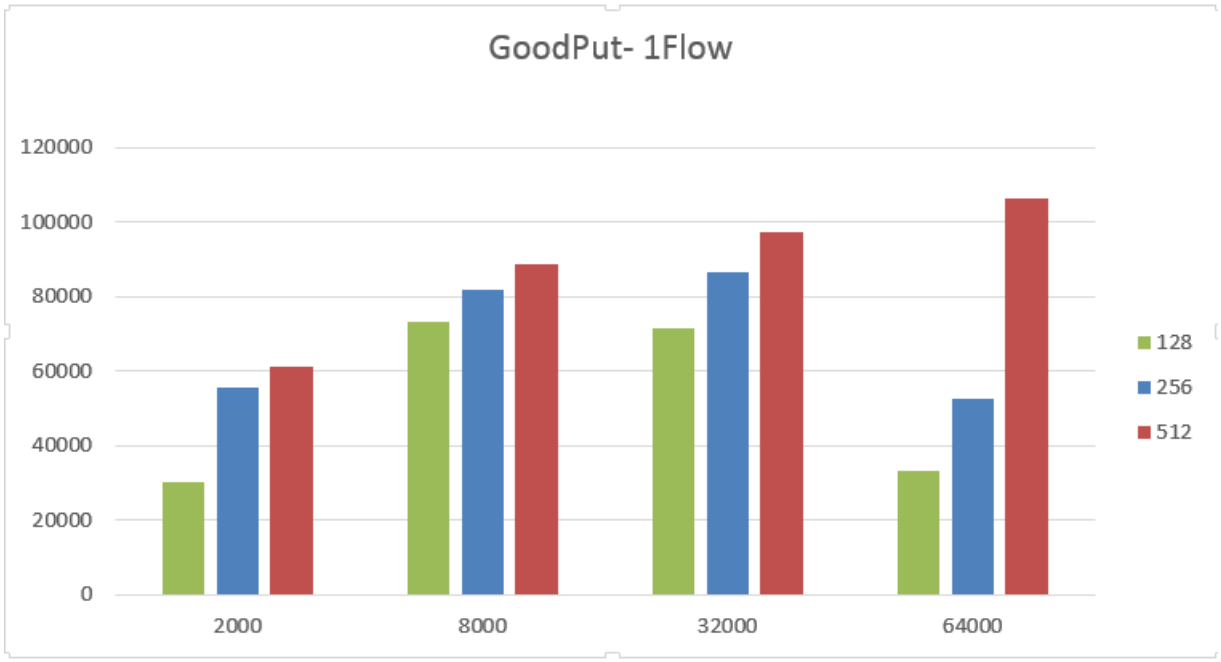


Figure 1: TCP-Tahoe Goodputs with Increasing Queue Sizes

Each colored bar represents a different packet segment size in bytes.

From the above chart we observe that TCP-Tahoe protocol performs best with segment size of 512 and with larger queue sizes. But with variable segment sizes and queue sizes the optimal solution would be to select median values of queue size to better serve varying segment sizes.
The best solution as observed is for segment size of 512 and queue size of 64000, where the goodput nearly reaches the link bandwidth.

III. TCP Multi-Flow Throughput

The second setup consisted of ten simultaneous flows all sharing the same three links. To measure the variation between the ten flows, ten bulk send applications each transmitted data to a different port on the receiver node. In other words, all ten flows are sharing the bandwidth of the links. Each bulk send application was started at a random time between 0 – 0.1 seconds, with times generated using a pseudo-random number generator with a uniform distribution.

One would expect the flows to share the bandwidth relatively evenly, but from the goodput observed some links are better served with high bandwidth while others are throttled resulting in lower servicing bandwidth.

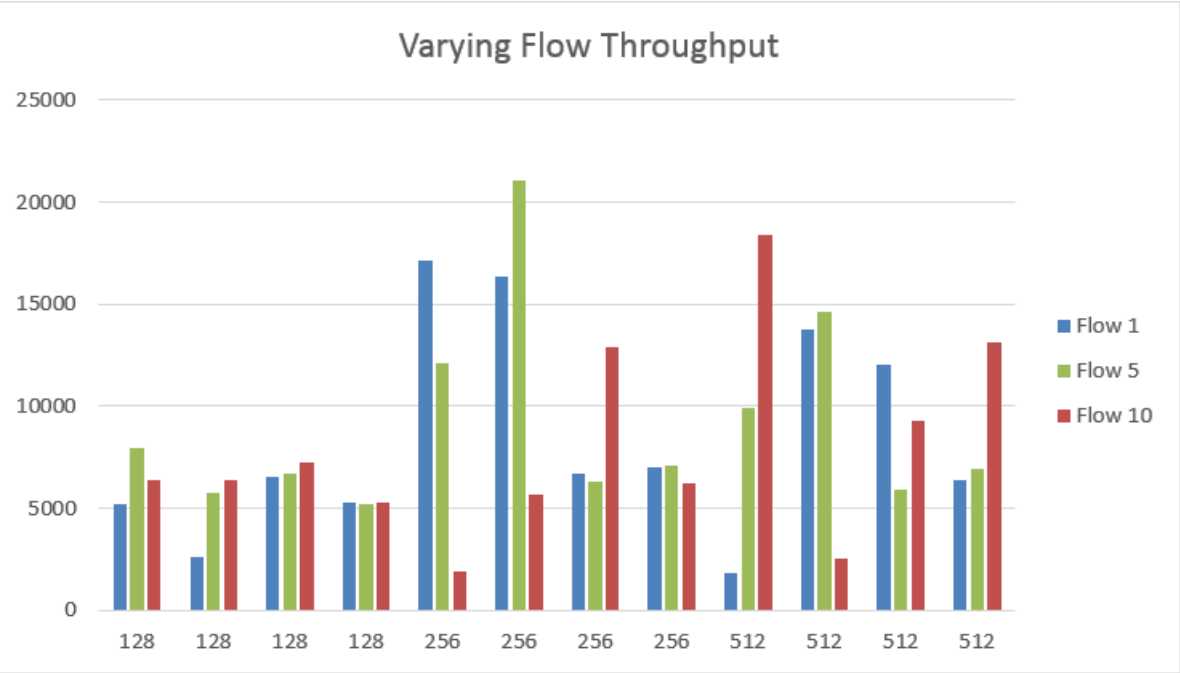


Figure 2: Various flow goodput for different segment and queue sizes

The above figure shows how the goodput of the 10 simultaneous flows vary compared for different segment sizes and queue sizes for Tcp-Tahoe. Only a small sample of simulation data is represented here for brevity and clarity. Each colored bar represents a different transmission flow.

As observed the ten different transmission flows did not share the bandwidth evenly at all. Instead their goodputs varied tremendously. This is likely due to the event in which if one or more flows have a timeout or a triple duplicate ACK and slow their transmission rates significantly due to a drop in the congestion window and slow start threshold, it leaves more bandwidth for the other flows. Therefore, which flow results in the largest goodput largely based on the dataflow of the various flows and also based primarily on the pseudo-random number generator which is used to simulate real world transmissions.

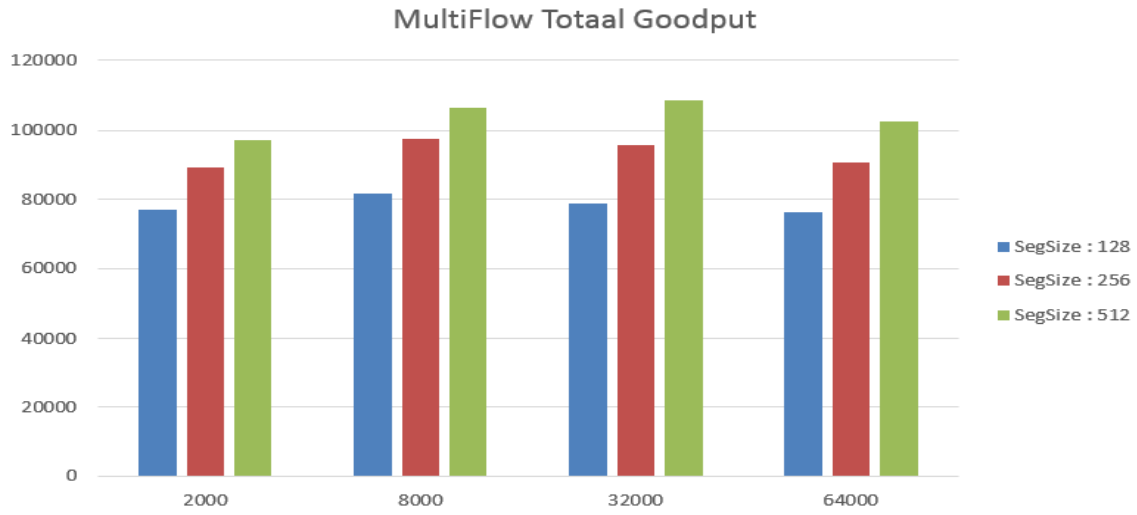


Figure 3: MultiFlow total goodput for varying SegSize and Queue Size

From (Fig:3) we observe that the link bandwidth is better utilized by multi-flow topology than a single flow topology (Fig:1). This is due to the presence of multiple nodes using the link simultaneously generating more traffic, leading to a better link bandwidth use.

IV. Conclusion

Overall, the performance of the simulations were a bit surprising in some areas and as expected in others. Segment size interacted with queue size to perform better with smaller segment sizes due to the ability to get closer to the queue size before overflowing or splitting up packets. In future simulations more information about packets losses and duplicate acknowledgements can be recorded for better insight.

When performing the ten simultaneous flows, a better approach would be to use different seeds for the pseudo-random number generator between runs to simulate a more realistic topology setup. Also, testing different distributions for the pseudo-random number generator may provide more insight, and a more realistic average result.