ECE 6110 – CAD for Computer Networks

Project 1 – TCP Throughput measurement

Name – Vivian Dsilva gtID – 903041189

This project was aimed at understanding Transmission Control Protocol (TCP) and analyzing the goodput obtained when segments of different sizes were sent across a bottleneck. The queue size and window size were also varied to understand how the variation of these would affect the goodput on the network.

The topology consisted of a source and a sink and a bottleneck network in between. The project consisted of two parts. In the first part, there was only one flow across the bottleneck (i.e. there was one source and one sink) while in the second part, there were 10 flows across the bottleneck (i.e. there were 10 sources and 10 sinks)

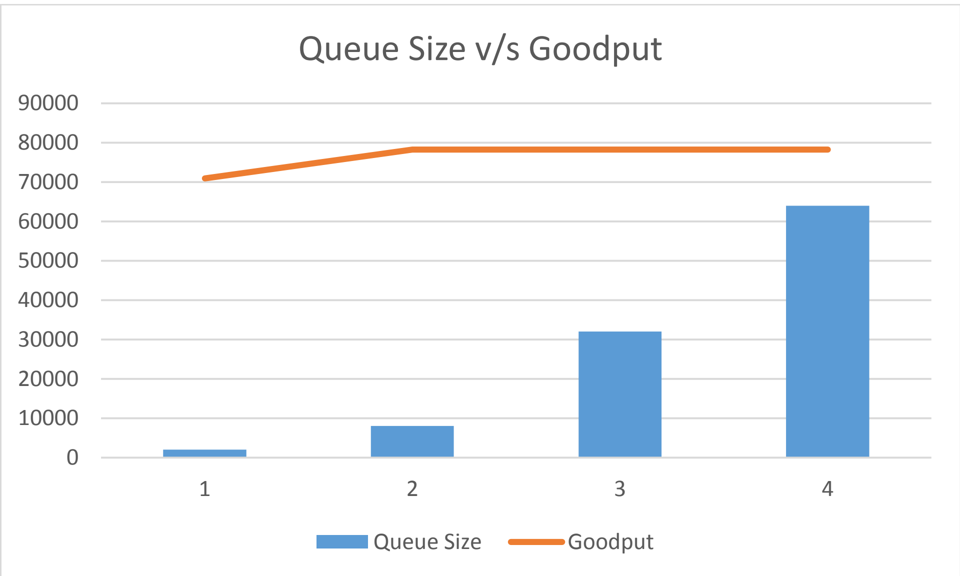
The following are the outputs of the TCP throughput measurement. I have considered the conditions for a single flow initially.

- For a single flow, I changed the queue size and kept the other parameters (i.e. segment size and window size) constant.

Keeping window size as 8000 bytes, segment size as 512 bytes and varying the Queue Size, we get the following output.

Queue Size (Bytes)	Goodput (Bytes/sec)
2000	70923.2
8000	78242.3
32000	78242.3
64000	78242.3

I plotted the above table into a graph. This is shown below



As we can see from the graph above, for a particular value of Segment Size and Window Size, as the queue size increases, the goodput increases initially and then becomes constant. This might be due to the fact that the goodput saturates after some time. The reason for this is explained below.
The goodput initially increases even though the queue size is 2000, since the advertised window size is 8000. Hence more bytes are transmitted increasing the goodput. Later, we increase the queue size but the

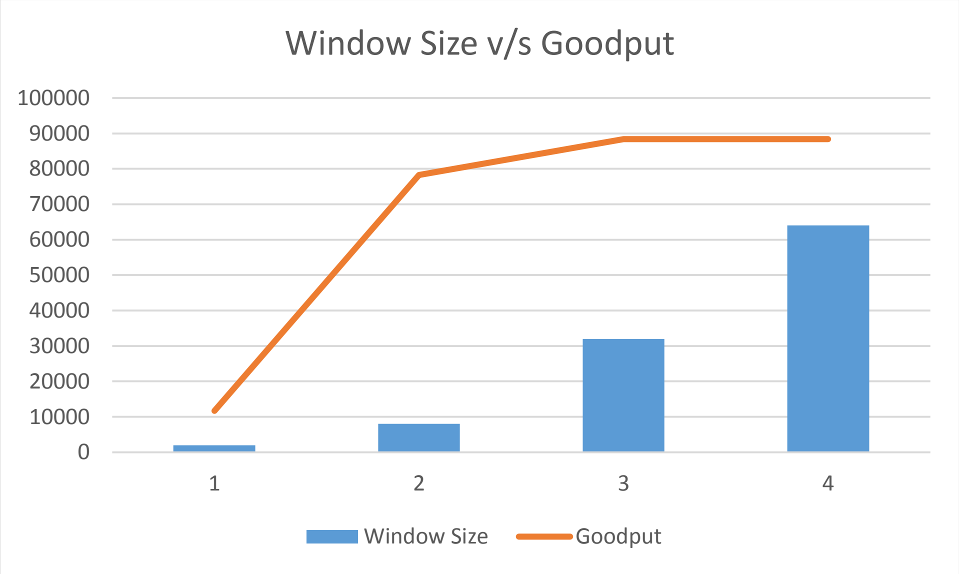
receiver still advertises a window of 8000. Hence, the receiver window becomes a limiting factor for the goodput and it does not increase beyond 78242.3 bytes/second.

- Next, for a single flow, I changed Window size from 2000 to 64000 and kept the Queue Size and Segment Size as constant.

Keeping Queue Size as 8000 bytes, segment size as 512 bytes and varying Window Size, we get the following output

Window Size (Bytes)	Goodput (Bytes/sec)
2000	11648.7
8000	78242.3
32000	88396.3
64000	88396.3

I plotted a graph for the above output. It is shown below



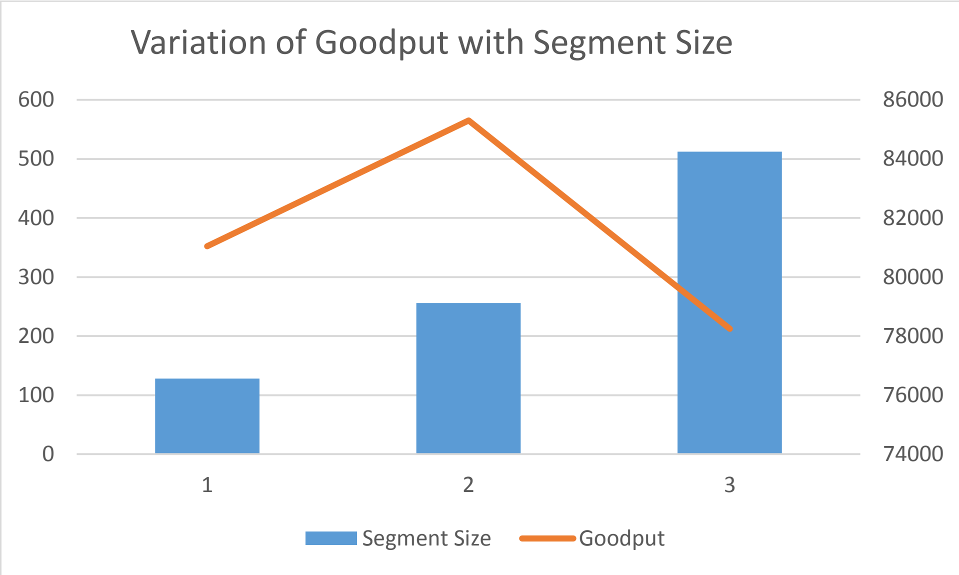
Now, from the above graph, we see that by varying Window size for constant Queue Size and Segment Size, the goodput initially increases and then it becomes constant. This is once again due to the fact that the throughput saturates after sometime. The reason for this can be explained as follows. Initially, when the advertised window size was small, the goodput was less since lesser number of bytes were transmitted. When the window size was gradually increased, the goodput increased as more number of bytes could be transmitted. But when the window size exceeds the queue size, we see that the goodput converges to a constant value, since here the queue size becomes a limiting factor (since it is lesser of the window size and queue size) which keeps the goodput at 88396.3 bytes/second. So even if the window size keeps on increasing, the constant queue size limits the number of bytes that can be sent.

- Now, finally for a constant window size and queue size, I changed the segment size from 128 to 512 bytes.

Keeping the window size and queue size constant at 8000 bytes and varying the segment size from 128 to 512 bytes, the following output was obtained.

Segment Size (Bytes)	Goodput (Bytes/second)
128	81038.6
256	85303.7
512	78242.3

I plotted the above table into a graph. This can be shown below.

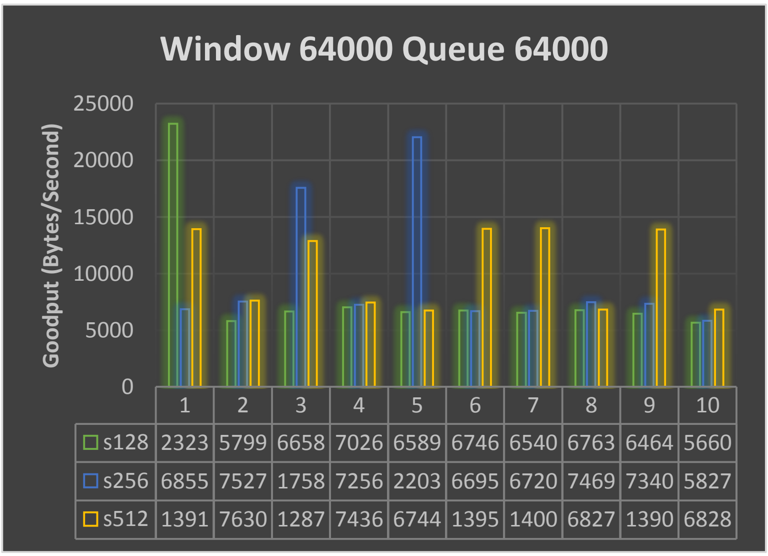


In the above graph, the scale on the left is for the segment size while the scale on the right is for the goodput. We can see that as the segment size keeps on increasing, the goodput initially increases and then suddenly drops. This might be due to the fact that for a case of the segment size and queue size each being 8000 bytes, the smaller segments (128 and 256 byte segments) have not been dropped. Thus, increasing the goodput initially. But when the segment size becomes 512 bytes, packets are dropped and hence there is a decrease in goodput. This could be one possible explanation for the decrease in throughput.

Now, I consider for the case of 10 Flows:

For the case of 10 flows, a number of outputs could be obtained since there would be so many combinations of segment size, queue size and Window Size. But for the sake of simplicity, I just considered one case of window Size and Queue size each being 64000 bytes and the segment size varying from 128 bytes to 512 bytes.

The graph can be shown as follows



I used the above graph in conjunction with a number of other graphs and observed that when the Queue Size was increased, there was some unfairness among flows leading to poor performance. This was due to the fact that certain flows were dominant in the network and certain flows were not. Hence, there was an overall decrease in goodput.

Also, it could be inferred from a number of cases that when there were 10 flows in the network the average goodput was lesser than the goodput obtained for the single flow case for a particular value of Segment Size, Queue Size and Window size. This is because the 10 flows would lead to more congestion in the network, thereby decreasing goodput.

Conclusion:

Thus in this project, the TCP goodput was measured. We saw that a number of parameters like the Segment Size, Queue Size and Window Size affects the goodput. Also, we observed that the goodput can vary when there are more than one flow in the network.