Project 1: TCP Throughput Measurements

NAME: OM KALE
GTID: 903041445
COURSE: ECE-6110
Username: okale6

The aim of this project is to measure the throughput of a single TCP connection through a bottleneck link, as a function of queue limits, segment sizes, and window sizes using the ns-3 simulator.

First, the simulation model of the topology is created as in the question creating the source, sink and two leaf nodes. Then using command line instructions I displayed the output for different cases for a single flow.

Here for my inferences, I consider a single flow first and vary one parameter at a time from the command line. Then in the second part I consider all 10 flows. I represent this in the form of tables and graphs as below:

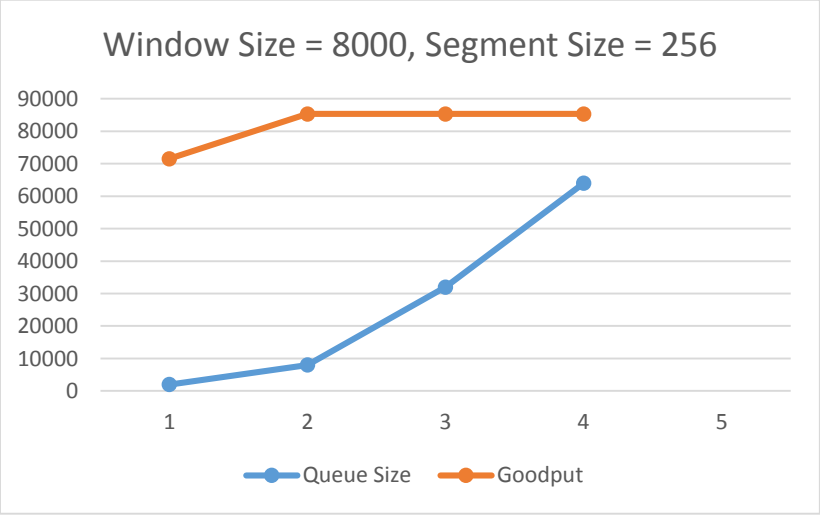
Observations:

VARYING QUEUE SIZE:

Considering a single flow, varying queue size and keeping segment size and window size constant gives the following results:

Number of flows = 0, Window Size = 8000, segSize = 256

Queue Size	Goodput
2000	71516
8000	85303.7
32000	85303.7
64000	85303.7



The goodput is in bytes per second and the queue size is in bytes.

Explanation and interpretation:

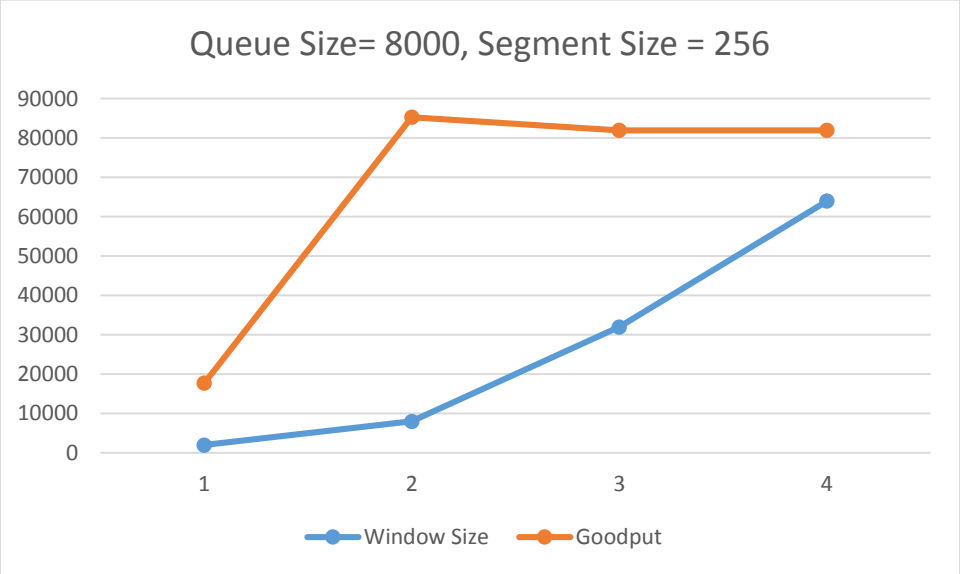
As we see from the above graph as queue size increases, the goodput increases initially but then it saturates and remains constant. The reason for this is the fact that first the queue size is small (in my case 2000), however advertised window (window size) is 8000, thus lots of bytes are transferred and hence goodput increases. Next, the queue size is increased to 8000 and window size is still constant at 8000, goodput remains same. After this even, if queue size is increased, the receiver still advertises 8000 bytes as its window size to the queue and hence the same goodput of 85303.7 bytes per second is seen. Thus, we can say that the window size is actually a limiting factor on the goodput in this case.

VARYING WINDOW SIZE:

Considering a single flow, varying window size and keeping segment size and queue size constant gives the following results:

Number of flows = 0, Queue Size = 8000, segSize = 256

Window Size	Goodput
2000	17730.8
8000	85303.7
32000	81953.4
64000	81953.4



The goodput is in bytes per second and the window size is in bytes

Explanation and interpretation:

As we see from the above graph as window size is small initially, less number of bytes are transferred and hence goodput is comparatively less. As window size is advertised to be 8000, there is an increase in the goodput as lots of bytes will be transferred to the receiver. After that, if window size increases from 8000 to 32000, the goodput drops. The reason for this is that, here the queue size i.e. the congestion window is remaining constant equal to 8000 and hence there will be dropping of packets initially as the advertised window is growing.

Thereafter, the queue size becomes the limiting factor as it is the minimum of the two and now, even if advertised window size increases the goodput is constant at 81953.4 bytes per second.

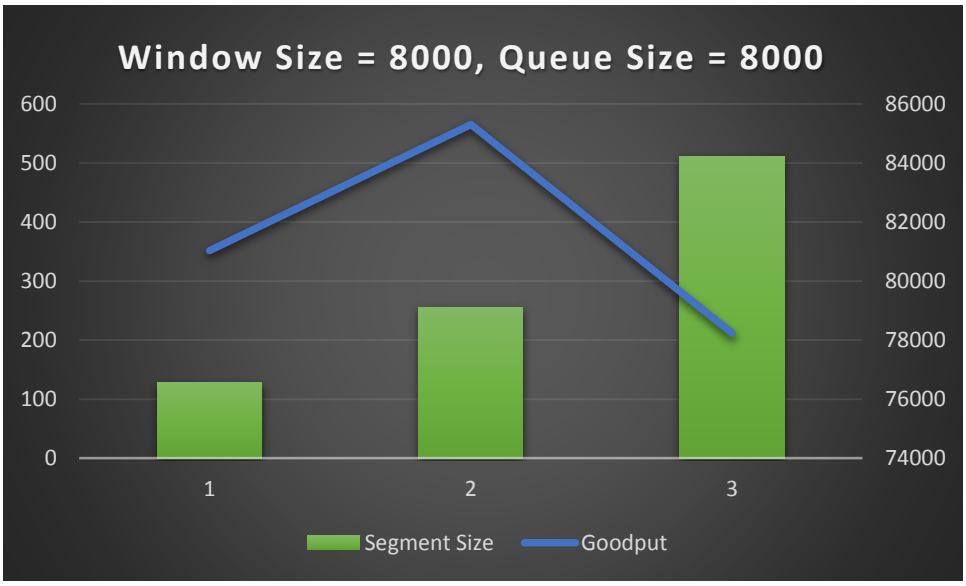
VARYING SEGMENT SIZE

Considering a single flow, varying segment size and keeping window size and queue size constant gives the following results:

Number of flows = 0, Queue Size = 8000, Window Size = 8000

Segment size is in bytes.

Segment Size	Goodput
128	81038.6
256	85303.7
512	78242.3

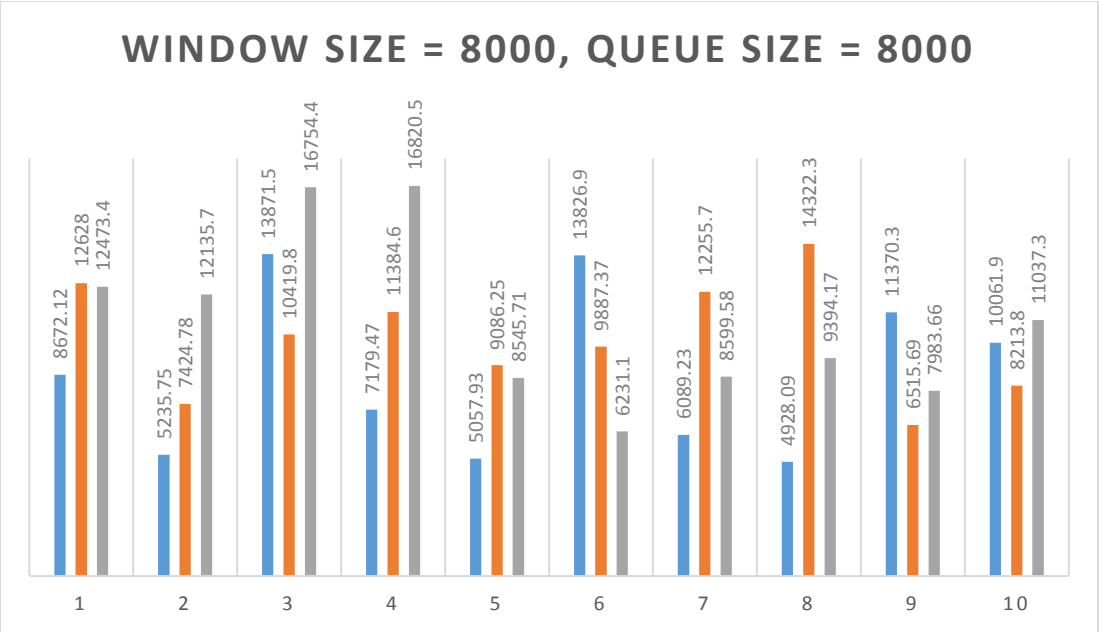


Explanation and interpretation:

Here for constant window size and queue size, when we increase the segment size, initially the goodput increases. The reason being, first the segment sizes of 128 and 256 bytes are small and hence transferred successfully to receiver without dropping packets, giving an increase in goodput. Next, the segment size is increased to 512. However, the queue (congestion window) size and window size are same and constant. This might cause dropping of packets in the bottleneck, thus reducing goodput.

For nflows = 10,

Varying segment sizes, I found results for different cases of queue size and window size. I have plotted one of the cases that gave me good results for goodput on an average.



Here, the blue, red and grey bars represents the varying segment sizes, keeping window and queue size constant equal to 8000 bytes.

Explanation and interpretation:

Here we see that, when the queue size is equal to the window size, the goodput is higher, on average for all flows. The reason for this may be the fact that the queue is adjusting exactly the right amount depending on segment size and hence there is no loss of packets from each of the flows. However, I still observed certain cases where goodput varies i.e. it increases and decreases indefinitely.

Also as number of flows increases the average goodput seems to be decreasing. This result is observed for most of the cases when all 10 flows are considered. This seems probable as more flows might lead to congestion in the link thus reducing goodput.