part_c

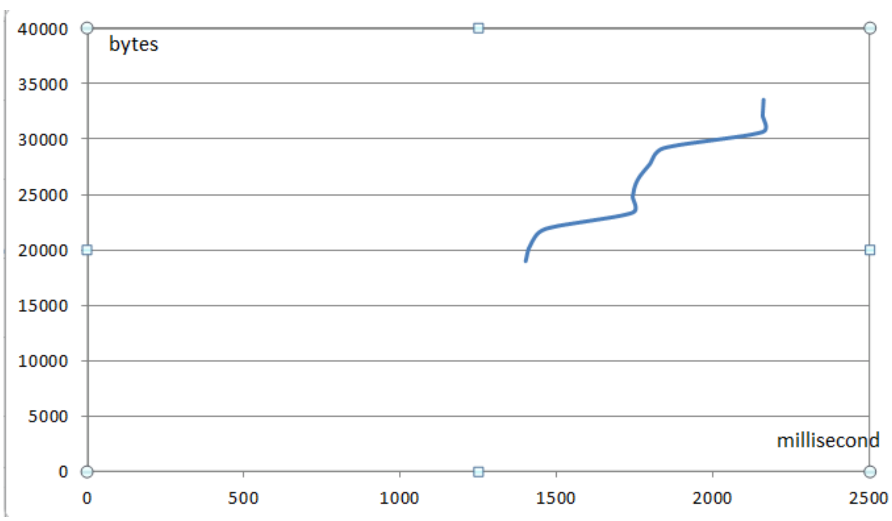
(1)

The initial congestion window is 4380 byes, which is 3 times the mss(1460byes)

And after the transimission starts, each time the server receive the ack, the congestion windows increases by 1 mss.

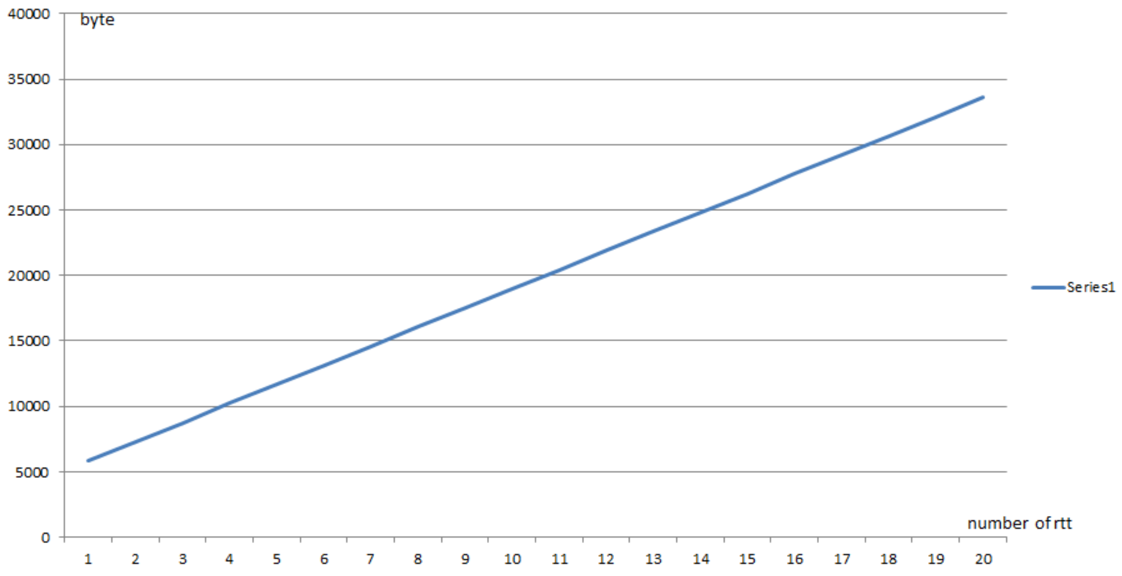
Here I dont consider the case when it reaches the threashold since i dont know how to calculate the threashold.

Below is the graph for congestion window. The y axis is the size of the congestion window. **The x axis is the time when the recever receives an ack from sender**.



For the following graph, The y axis is the size of the congestion window. **The x axis is the number of rtt (the number of ack) but not the actual time**.

Since congestion window increases by 1 mss for one ack, the graph is a linear line.



(2)

Here i show the rtt and rto in server and client sides,respectively.

For the rtt calculated by me at the http server, I compare the results to the rtt graph given by wireshark and they are exactly the same.

However for the rtt with respect to the http client, the rtt graph in wireshark is wierd(it only has 3 rtt) so i am not able to do the comparison

The first 4 retransmission timeout values estimated at the **HTTP server** is as below.

rtt number: 1 from packet 5 to packet 7

rtt:_213.743896484375 millsecond

srtt: 213.743896484375 millsecond

rttvar: 106.8719482421875 millsecond

rto: 1213.743896484375 millsecond

rtt number: 2 from packet 6 to packet 8

rtt:_213.652099609375 millsecond

srtt: 213.732421875 millsecond

rttvar: 80.17691040039062 millsecond

rto: 1213.732421875 millsecond

rtt number: 3 from packet 9 to packet 14

rtt:_209.388916015625 millsecond

srtt: 213.18948364257812 millsecond

rttvar: 61.21855926513672 millsecond

rto: 1213.1894836425781 millsecond

rtt number: 4 from packet 10 to packet 15

rtt:_208.801025390625 millsecond

srtt: 212.64092636108398 millsecond

rttvar: 47.01103401184082 millsecond

rto: 1212.640926361084 millsecond

first 4 retransmission timeout values estimated at the **HTTP client** is as below.

rtt number: 1 from packet 4 to packet 5

rtt:_0.421142578125 millsecond

srtt: 0.421142578125 millisecond

rttvar: 0.2105712890625 millisecond

rto: 1000.421142578125 millisecond

rtt number: 2 from packet 8 to packet 9

rtt:_0.701904296875millsecond

srtt: 0.45623779296875 millisecond

rttvar: 0.228118896484375 millisecond

rto: 1000.4562377929688 millisecond

rtt number: 3 from packet 15 to packet 16

rtt:_1.135986328125millsecond

srtt: 0.5412063598632812 millisecond

rttvar: 0.34102630615234375 millisecond

rto: 1000.5412063598633 millisecond

rtt number: 4 from packet 18 to packet 19

rtt:_0.907958984375millsecond

srtt: 0.5870504379272461 millisecond

rttvar: 0.3474578857421875 millisecond

rto: 1000.5870504379272 millisecond

From the above experiment, we can tell that the rtt calculated in http client is much smaller than the one calculated in server.

Actually I am confused with the small rtt in http client. I think it should be almost the same as the one calculated in server.