Manish Yadav
3836-6483
m.yadav@ufl.edu

Homework 4

CNT5106C : Fall 2020 : Dr. Ye Xia
Due Tue, Nov 3, 2020

Contents

Problem 1

P31

Suppose that the five measured `SampleRTT` values (see Section 3.5.3) are 106 ms, 120 ms, 140 ms, 90 ms, and 115 ms. Compute the `EstimatedRTT` after each of these `SampleRTT` values is obtained, using a value of $\alpha = 0.125$ and assuming that the value of `EstimatedRTT` was 100 ms just before the first of these five samples were obtained. Compute also the `DevRTT` after each sample is obtained, assuming a value of $\beta = 0.25$ and assuming the value of `DevRTT` was 5 ms just before the first of these five samples was obtained. Last, compute the TCP `TimeoutInterval` after each of these samples is obtained

---

Answer

- For RTT = 106ms

$$\text{EstimatedRTT} = \alpha * \text{SampleRTT} + (1 - \alpha) * \text{EstimatedRTT}$$
$$= 0.125 * 106 + (1 - 0.125) * 100$$
$$= 0.125 * 106 + 0.875 * 100$$
$$= 100.75ms$$

$$\text{DevRTT} = \beta * |\text{SampleRTT} - \text{EstimatedRTT}| + (1 - \beta) * \text{DevRTT}$$
$$= 0.25 * |106 - 100.75| + (1 - 0.25) * 5$$
$$= 0.125 * 106 + 0.875 * 100$$
$$= 5.06ms$$

$$\text{TCPTimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$
$$= 100.75 + 4 * 5.0625$$
$$= 121ms$$

- For RTT = 120ms

$$\text{EstimatedRTT} = \alpha * \text{SampleRTT} + (1 - \alpha) * \text{EstimatedRTT}$$
$$= 0.125 * 120 + (1 - 0.125) * 100.75$$
$$= 103.15ms$$

$$\text{DevRTT} = \beta * |\text{SampleRTT} - \text{EstimatedRTT}| + (1 - \beta) * \text{DevRTT}$$
$$= 0.25 * |120 - 103.15625| + (1 - 0.25) * 5.0625$$
$$= 8ms$$

$$\text{TCPTimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$
$$= 103.15 + 4 * 8$$
$$= 135.15ms$$

- For RTT = 140ms

$$\text{EstimatedRTT} = \alpha * \text{SampleRTT} + (1 - \alpha) * \text{EstimatedRTT}$$
$$= 0.125 * 140 + (1 - 0.125) * 103.15$$
$$= 107.75ms$$

Problem 1 [P31] continued on next page...

$$\texttt{DevRTT} = \beta * |\texttt{SampleRTT} - \texttt{EstimatedRTT}| + (1 - \beta) * \texttt{DevRTT}$$
$$= 0.25 * |140 - 107.75| + (1 - 0.25) * 8$$
$$= 14.06ms$$

$$\texttt{TCPTimeoutInterval} = \texttt{EstimatedRTT} + 4 * \texttt{DevRTT}$$
$$= 107.75 + 4 * 14.06$$
$$= 164ms$$

- For RTT = 90ms

$$\texttt{EstimatedRTT} = \alpha * \texttt{SampleRTT} + (1 - \alpha) * \texttt{EstimatedRTT}$$
$$= 0.125 * 90 + (1 - 0.125) * 107.75$$
$$= 105.53ms$$

$$\texttt{DevRTT} = \beta * |\texttt{SampleRTT} - \texttt{EstimatedRTT}| + (1 - \beta) * \texttt{DevRTT}$$
$$= 0.25 * |90 - 105.53| + (1 - 0.25) * 14.06$$
$$= 14.42ms$$

$$\texttt{TCPTimeoutInterval} = \texttt{EstimatedRTT} + 4 * \texttt{DevRTT}$$
$$= 105.53 + 4 * 14.42$$
$$= 163.21ms$$

- For RTT = 115ms

$$\texttt{EstimatedRTT} = \alpha * \texttt{SampleRTT} + (1 - \alpha) * \texttt{EstimatedRTT}$$
$$= 0.125 * 115 + (1 - 0.125) * 105.53$$
$$= 106.715ms$$

$$\texttt{DevRTT} = \beta * |\texttt{SampleRTT} - \texttt{EstimatedRTT}| + (1 - \beta) * \texttt{DevRTT}$$
$$= 0.25 * |115 - 106.715| + (1 - 0.25) * 14.42$$
$$= 12.885ms$$

$$\texttt{TCPTimeoutInterval} = \texttt{EstimatedRTT} + 4 * \texttt{DevRTT}$$
$$= 106.715 + 4 * 12.885$$
$$= 158.255ms$$

Problem 2

Consider Figure 3.58 . Assuming TCP Reno is the protocol experiencing the behavior shown above, answer the following questions. In all cases, you should provide a short discussion justifying your answer.

P40 a

Identify the intervals of time when TCP slow start is operating.

> **Answer**
>
> [1,6] and [23,16]

P40 b

Identify the intervals of time when TCP congestion avoidance is operating.

> **Answer**
>
> [6, 16] and [7, 22]

P40 c

After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

> **Answer**
>
> It would be triple duplicate ACK. In case of timeout, congestion size would have been reduced.

P40 d

After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

> **Answer**
>
> It is detected due to timeout as the congestion window size is 1

P40 e

What is the initial value of `ssthresh` at the first transmission round?

> **Answer**
>
> 32 (At this size, slow start stops and congestion avoidance begins)

P40 f

What is the value of `ssthresh` at the 18th transmission round?

> **Answer**
>
> $\frac{42}{2} = 21$

P40 g

What is the value of `ssthresh` at the 24th transmission round?

> **Answer**
>
> $\frac{29}{2} = 14$

P40 h

During what transmission round is the 70th segment sent?

> **Answer**
>
> - Transmission round 1: pkt_1
>
> - Transmission round 2: pkt_{2-3}
>
> - Transmission round 3: pkt_{4-7}
>
> - Transmission round 4: pkt_{8-16}
>
> - Transmission round 5: pkt_{17-32}
>
> - Transmission round 6: pkt_{33-63}
>
> - Transmission round 7: pkt_{64-96}
>
> Therefore pkt_70 is sent in round 7

P40 i

Assuming a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, what will be the values of the congestion window size and of `ssthresh` ?

> **Answer**
>
> Congestion window size $= \frac{8}{2} + 3 = 7$
> `ssthresh` $= \frac{8}{2} = 4$

P40 j

Suppose TCP Tahoe is used (instead of TCP Reno), and assume that triple duplicate ACKs are received at the 16th round. What are the ssthresh and the congestion window size at the 19th round?

> **Answer**
>
> `ssthresh` $= 21$
> Congestion Window Size $= 1$

P40 k

Again suppose TCP Tahoe is used, and there is a timeout event at 22nd round. How many packets have been sent out from 17th round till 22nd round, inclusive?

> **Answer**
>
> - Round 17: 1 packet
> - Round 18: 2 packets
> - Round 19: 4 packets
> - Round 20: 8 packets
> - Round 21: 16 packets
> - Round 22: 21 packets

Problem 3

P41

Refer to Figure 3.55 , which illustrates the convergence of TCP's AIMD algorithm. Suppose that instead of a multiplicative decrease, TCP decreased the window size by a constant amount. Would the resulting AIAD algorithm converge to an equal share algorithm? Justify your answer using a diagram similar to Figure 3.55.

Answer
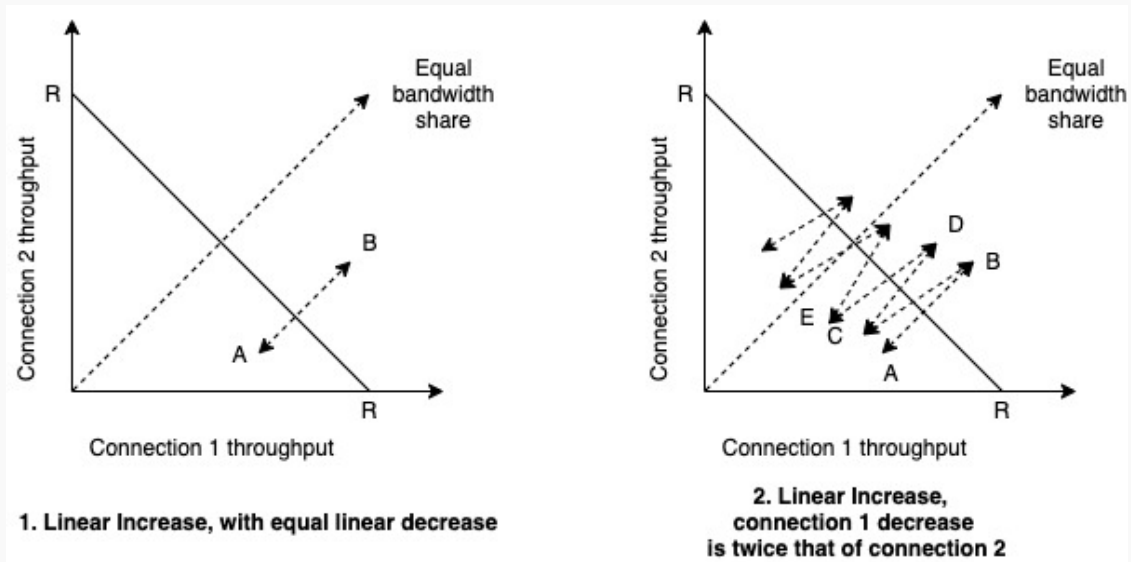
In fig 1, the ratio of linear decrease on loss of connection 1 and connection 2 is proportional to ratio of linear increase of connection 1 and connection. Hence there is no fluctuation of throughput and it stays on line AB.

In fig 2, linear decrease on loss between connection 1 and 2 is proportional to 2:1 i.e. whenever there is a loss, window size on connection 1 reduces by 2 times the window size of connection 2. Subsequently, the repeats and after few increase and decrease in losses, connection 1 throughput goes to 0 and full bandwidth is allocated for connection 2.

Problem 4

Consider sending a large file from a host to another over a TCP connection that has no loss.

P44 a

Suppose TCP uses AIMD for its congestion control without slow start. Assuming cwnd increases by 1 MSS every time a batch of ACKs is received and assuming approximately constant round-trip times, how long does it take for cwnd increase from 6 MSS to 12 MSS (assuming no loss events)?

> **Answer**
>
> After 1 RTT, Congestion Window is set to 6 MMS, each ACKs increases it by 1; therefore by 2 RTT it is 7 MSS, 3 RTTs to 8 MSS and so on so forth. After 7 RTTs, it is increased to 12 MSS.

P44 b

What is the average throughout (in terms of MSS and RTT) for this connection up through time=6 RTT?

> **Answer**
>
> - RTT 1: 5 MSS sent
>
> - RTT 2: 6 MSS sent
>
> - RTT 3: 7 MSS sent
>
> - RTT 4: 8 MSS sent
>
> - RTT 5: 9 MSS sent
>
> - RTT 6: 10 MSS sent
>
> Total MSS Sent $= 5 + 6 + 7 + 8 + 9 + 10 = 45MSS$. Therefore average throughput $= \frac{45}{6} = 7.5MSS/RTT$

Problem 5

Recall the macroscopic description of TCP throughput. In the period of time from when the connection's rate varies from $W/(2 \cdot RTT)$ to W/RTT, only one packet is lost (at the very end of the period).

P45 a

Show that the loss rate (fraction of packets lost) is equal to L=loss rate= $\frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$

> **Answer**
>
> Loss Ratio = Number of packets lost over number of packets sent
>
> The packets send in each round is given using the equation:
>
> $$\begin{aligned}
> &= \frac{W}{2} + (\frac{W}{2} + 1) + ... + W) \\
> &= \sum_{n=0}^{W/2} (\frac{W}{2} + n) \\
> &= (\frac{W}{2} + 1)\frac{W}{2} + \frac{\frac{W}{2}(W/2 + 1)}{2} \\
> &= \frac{W^2}{4} + \frac{W}{2} + \frac{W^2}{8} + \frac{W}{4} \\
> &= \frac{3}{8}W^2 + \frac{3}{4}W
> \end{aligned}$$
>
> Therefore loss rate $= \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$

P45 b

Use the result above to show that if a connection has loss rate L, then its average rate is approximately given by $\approx \frac{1.22 \cdot MSS}{RTT \cdot \sqrt{L}}$

> **Answer**
>
> For sufficiently large value of W, $\frac{3W^2}{8} >> \frac{3W}{4}$, therefore loss rate is dependent on $\frac{8W^2}{3}$ i.e. $W = \sqrt{\frac{8}{3L}}$
>
> Therefore average throughput $= \frac{3}{4}\sqrt{\frac{8}{3L}}\frac{MSS}{RTT} = \frac{1.22 MSS}{RTT\sqrt{L}}$

Problem 6

P47

Consider the scenario described in the previous problem. Suppose that the 10Mbps link can buffer a finite number of segments. Argue that in order for the link to always be busy sending data, we would like to choose a buffer size that is at least the product of the link speed C and the two-way propagation delay between the sender and the receiver.

Answer

Assuming W to be max window size and S to be buffer size. Assuming TCP to be operating in round by round fashion. One round = 1 RTT. Packet loss will occur when the window size reaches W, in this scenario, sender would reduce the congestion window size by $\frac{1}{2}$ and wait for the ACKs of remaining $\frac{W}{2}$ packets before it begin sending data again. By sending data every W/(2*C) period, we keep the link busy. This is when where the sender waits for the ACKs from W/2 packets. Therefore S/C $\geq$ W/2.

Upon reaching window size of W/2 and emptying the buffer, we need to be sure that link is busy sending data as well. There we need to ensure that $\frac{\frac{W}{2}}{2T_p} \geq C$ i.e. $\frac{W}{2} \geq C * 2T_p$ i.e. $S \geq C * 2T_p$ where T_p is propagation delay between sender and receiver.

Problem 7

Wireshark Lab: TCP

1

What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?

> **Answer**
>
> IP address: 192.168.1.102
> TCP port number: 1161

2

What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

> **Answer**
>
> IP address: 128.119.245.12
> Port: 80

3

What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

> **Answer**
>
> IP address: 10.76.65.181
> Port: 80

4

What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

> **Answer**
>
> TCP SYN sequence number is 0.
> SYN Flag in that segment is set to 1.

5

What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

> **Answer**
>
> SYNACK segment number is 0.
> Acknowledgement filed in the SYNACK segment is 1.
> The server adds 1 to the intial sequence number of SYN segment from the client computer.
> Both SYN flag and Acknowledgement in the segment are set to 1

6

What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.

> **Answer**
>
> TCP segment 6 contains the POST request. Sequence number is 1

7

Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the EstimatedRTT value (see Section 3.5.3, page 242in text) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 242 for all subsequent segments

> **Answer**
>
> | Segment Num | Time | Sequence Num | ACK Segment Num | ACK Time | RTT | Est RTT |
> |---|---|---|---|---|---|---|
> | 6 | 1.475113 | 1 | 9 | 1.553474 | 0.078361 | 0.078361 |
> | 7 | 1.47534 | 738 | 9 | 1.553474 | 0.078361 | 0.078361 |
> | 8 | 1.47535 | 2086 | 11 | 1.558114 | 0.082765 | 0.0789115 |
> | 10 | 1.55357 | 3434 | 12 | 1.558121 | 0.08277 | 0.07939381 |
> | 13 | 1.55824 | 4782 | 17 | 1.62947 | 0.075897 | 0.07895671 |
> | 14 | 1.55824 | 6130 | 20 | 1.637328 | 0.079084 | 0.07897262 |
> | 6 | 1.475113 | 1 | 9 | 1.553474 | 0.078361 | 0.078361 |

8

What is the length of each of the first six TCP segments?

> **Answer**
>
> Segment Num 6: Sequence Num: 1, Length: 737
> Segment Num 7: Sequence Num: 738, Length: 1348
> Segment Num 8: Sequence Num: 2086, Length: 1348
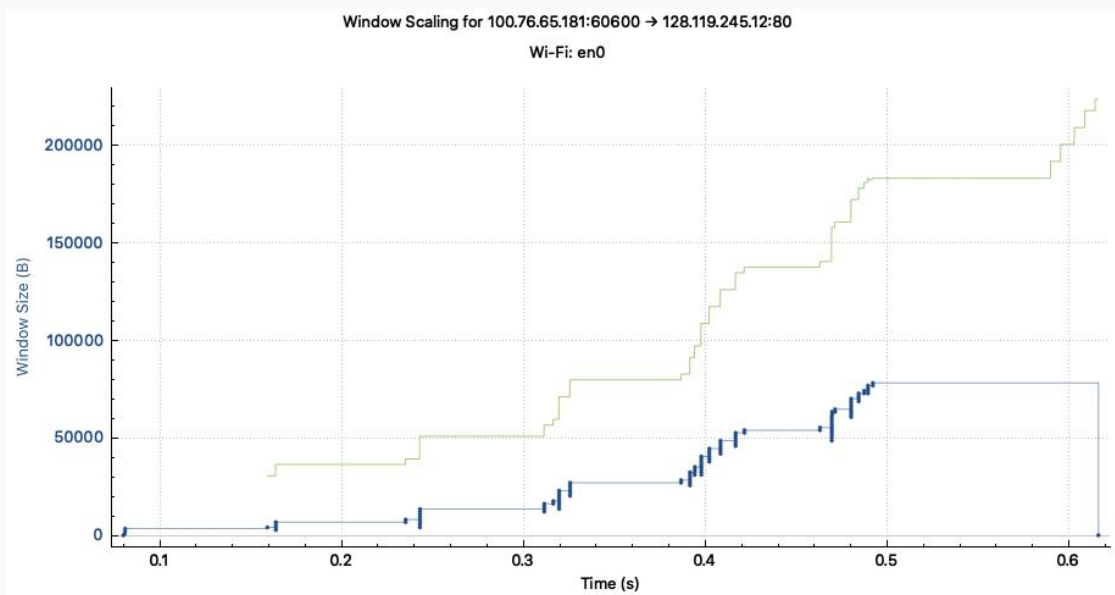> Segment Num 10: Sequence Num: 3434, Length: 1348

> Segment Num 13: Sequence Num: 4782, Length: 1348
> Segment Num 14: Sequence Num: 6130, Length: 1348

9

What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

> **Answer**
>
> Window Size = 28960.
>
>
>
> Window Scaling for 100.76.65.181:60600 → 128.119.245.12:80
> Wi-Fi: en0
>
> As seen from figure there was no throttling

10

Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

> **Answer**
>
> There isn't seem to be any retransmitted segment. Retransmitted packet are highlighted using black color and there's none.

11

How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 250 in the text).

> **Answer**
>
> The difference between the acknowledged sequence numbers of two consecutive ACKs indicates the data received by the server between these two ACKs. The receiver typically acknowledges

1400 - 2000 bytes. This almost triples is almost constant in the end as they are received in ordered fashion and hence ACKs every other segment.

12

What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

> **Answer**
>
> Throughput = 48k bytes/sec (As per file statistics)

13

Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP's slow-start phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

> **Answer**
>
> We could slow start happening at 0-0.15s and then the sender transmits in the batches of 6. Now in this case we do not observe any linear increase in batch size. This is different from the one in text as it does not AIMD strategy as window size does not increase even when there is no congestion.

14

Answer each of two questions above for the trace that you have gathered when you transferred a file from your computer to gaia.cs.umass.edu

> **Answer**
>
>
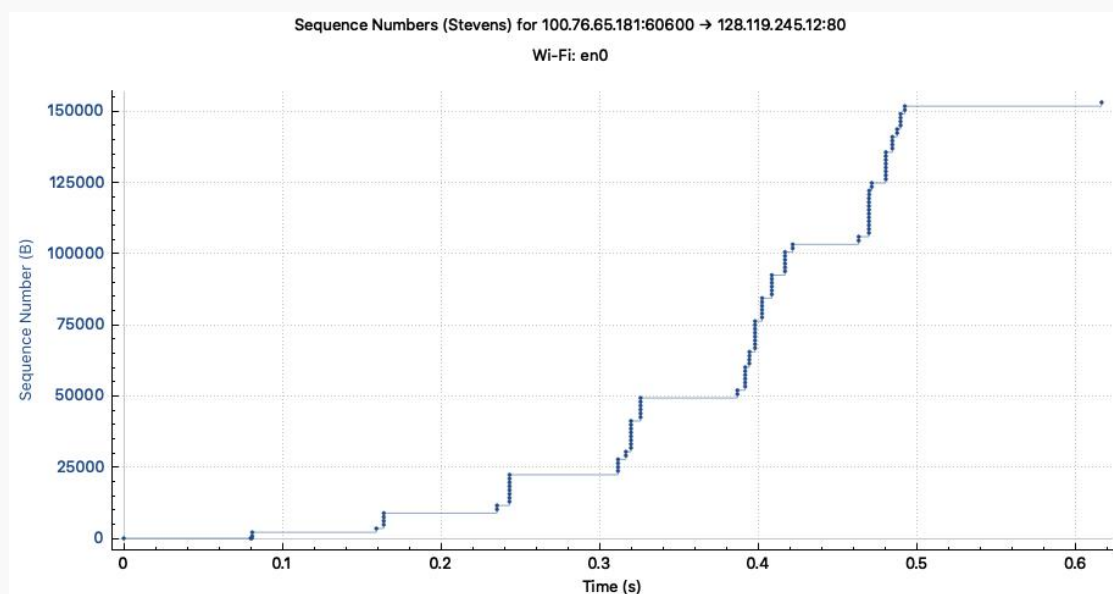
From the above stevens graph it is clear that congestion starts at 0ms but doesn't seem to end as file upload finishes even before congestion could happen or window slow start phase could. This behaviour is due higher bandwidth of my machine. Hence no comparison could be made with the one in text due to incomplete slow start behaviour.

References