CN Assignment – 4

Noopur R Kalawatia

1980 – 9834

Problems:

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 α = 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 β = 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.

Solution: Consider the SampleRTT values, 106 ms, 120 ms, 140 ms, 90 ms, 115 ms.

α = 0.125

β = 0.25

Estimated RTT = 100 ms

Formula for Estimated RTT is,

EstimatedRTT = (1 – α) \* EstimatedRTT + α \* SampleRTT, with α = 0.125

EstimatedRTT = 0.875 \* EstimatedRTT + 0.125 \* SampleRTT

Formula for DevRTT is,

DevRTT = (1 – β) \* DevRTT + β \* | SampleRTT – EstimatedRTT |

With β = 0.25

DevRTT = (0.75) \* DevRTT + 0.25 \* | SampleRTT – EstimatedRTT |

Formula for Timeout Interval is,

TimeoutInterval = EstimatedRTT + 4 \* DevRTT

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Estimated RTT | Dev RTT | Timeout interval |
| 106 | = 0.875 \* 100 + 0.125 \* 106  =100.75 | = (0.75) \* 5 + 0.25 \* | 106 – 100.75 |  = 5.0625 | = 100.75 + 4 \* 5.0625  = 121 |
| 120 | = 0.875 \* 100.75 + 0.125 \* 120  = 103.156 | = (0.75) \* 5.0625 + 0.25 \* |120 – 103.156|  = 8.0078 | = 103.156 + 4\* 8.0078  = 135.1872 |
| 140 | = 0.875 \* 103.156 + 0.125 \* 140  = 107.76 | = 0.75 \* 8.0078 + 0.25 \* |140 – 107.76|  = 14.065 | = 107.76 + 4 \* 14.065  = 164.0234 |
| 90 | = 0.875 \* 107.76 + 0.125 \* 90  = 105.54 | = 0.75 \* 14.065 + 0.25 \* | 90 – 105.54|  = 14.4337 | = 105.54 + 4 \* 14.4337  = 163.275 |
| 115 | = 0.875 \* 105.54 + 0.125 \* 115  = 106.7225 | = 0.75 \* 14.4337 + 0.25 | 115 – 106.7225|  = 12.8945 | = 106.7225 + 4 \* 12.8945  = 158.3005 |

P40. 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.

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

Solution: The time interval from [1,6] and [23,26]

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

Solution: The time interval from [6,16] and [17,22].

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

Solution: Is indicated by a triple duplicate ACK and the congestion window drops to 1.

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

Solution: Timeout, hence the congestion window drops to 1.

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

Solution: 32. This is window size where the slow start ends and the congestion avoidance begins.

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

Solution: The threshold will be set to half when the packet loss takes place. When loss is detected over the 16th round, the window size is 42 respectively. Thus, at the 18th round the value decreases to window size – 21.

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

Solution: The threshold will be set to half when the packet loss takes place. When loss is detected over the 22th round, the window size is 29 respectively. Thus, at the 24th round the value decreases to window size – 14.

1. During what transmission round is the 70th segment sent?

Solution: The 70th segement will be transmitted in the seventh round.

1. 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?

Solution:

The threshold will be set to half the current value of the congestion window when the loss occurred and congestion window will be set to the new threshold value + 3 MSS . Thus the new values of the threshold and window will be 4 and 7.

1. 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?

Solution:

ssthresh = 21

Congestion window size = 1

1. 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?

Solution:

Round 17 – 1 packet

Round 18 – 2 packet

Round 19 – 4 packets

Round 20 – 8 packets

Round 21 – 16 packets

Round 22 – 21 packets

The total number of packets transmitted are 52.

P41. Refer to Figure 3.56, 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 3.56.

AIMD – Additive Increase and Multiplicative Decrease in throughput,

From the algorithm, with every connection the congestion window increases linearly

And the decrease in the same is multiplicative in nature when the receiver receives 3 ACKs.

AIAD refers to Additive Increase and Additive Decrease.

Suppose there are two connections made at the time, then if we assume that both the connections begin at the throughput A.

If each of the connection decide to increase their window size by 3 MSS, then their throughput will increase and will be at point – D.

If there is a loss of packet then both the connections will decrease their window size by 3MSS and will fall back to throughput – A.

Full bandwidth

Utilization line

R

Equal bandwidth

share

Connection 2

throughput

D

A

Connection 1 throughput

R

The main criteria of the algorithm are to have equal sharing of bandwidth between the two connections, but if we keep a constant increase or decrease ratio it will result in one connection having a throughput subsequently going to 0 and the other using the entire bandwidth. Thus the algorithm will never converge.

P44. Consider sending a large file from a host to another over a TCP connection

that has no loss.

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)?

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

Solution:

1. Let r bytes/s be the transmission rate of the TCP connection.

Let the current round trip time be RTT seconds.

The throughput of the connection is w / RTT

Given that, the cwnd increases by 1 MSS every time a batch of ACKs is received.

Following is the time taken for the cwnd to increase from 6MSS to 12 MSS respectively,

It takes 1 RTT to increase cwnd to 7 MSS.

It takes 2 RTT to increase cwnd to 8 MSS.

It takes 3 RTT to increase cwnd to 9 MSS.

It takes 4 RTT to increase cwnd to 10 MSS.

It takes 5 RTT to increase cwnd to 11 MSS.

It takes 6 RTT to increase cwnd to 12 MSS.

1. For the data,

In first RTT, 7 MSS were sent.

In second RTT, 8 MSS were sent.

In third RTT, 9 MSS were sent.

In fourth RTT, 10 MSS were sent.

It fifth RTT, 11 MSS were sent.

It sixth RTT, 12 MSS were sent.

Thus the total MSS sent till sixth RTT = 6 + 7 + 8 + 9 + 10 + 11 = 51

The average throughput is 51 / 6 = 8.5 MSS/RTT.

P45. Recall the macroscopic description of TCP throughput. In the period of time

from when the connection’s rate varies from W/(2 · RTT) to W/RTT, only one

packet is lost (at the very end of the period).

a. Show that the loss rate (fraction of packets lost) is equal to

L = loss rate =

b. Use the result above to show that if a connection has loss rate L, then its average rate is approximately given by,

Solution:

1. The loss rate, is the ratio of the number of packets lost over the number of packets sent. In a cycle, 1 packet is lost. The number of packets sent in a cycle is L

Thus the loss rate is,

L =

1. For a large value of W, Thus, L ≅ , and W =

Therefore, the average throughput is,

47.

Solution: Let W denote max window size. Let S denote the buffer size. For simplicity, suppose TCP sender sends data packets in a rounds fashion, with each round corresponding to a RTT. If the window size reaches W, then a loss occurs.

When the loss occurs, the sender will cute the congestion window by half and waits for the outstanding W/2 packets in the buffer, before it begins transmitting again.

In order to make sure the link is always busying sending data, we need to let the link be busy sending data in the period *W*/(2\**C*) (this is the time interval where the sender is waiting for the ACKs for the W/2 outstanding packets). Thus, S/C must be no less than *W*/(2\**C*), that is, *S*>=*W*/2.

Let Tp denote the one-way propagation delay between the sender and the receiver.

When the window size reaches the minimum W/2 and the buffer is empty, we need to make sure the link is also busy sending data. Thus, we must have

Thus,

Wireshark Labs – 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? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details of the selected packet header window” (refer to Figure 2 in the “Getting Started with Wireshark” Lab if you’re uncertain about the Wireshark windows.

Solution:

IP Address: 192.168.0.12

Port Number: 52996

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

Solution: gaia.cs.umass.edu IP address 128.119. 245.12, port no 80

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

Solution:

IP Address: 192.168.0.12

Port Number: 52996

1. 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?

Solution:

The sequence number of the SYN packet is 0.

The information that identifies the packet as SYN, is the flags field.

For the SYN packet only the SYN field of the packet will be set, as shown below,

A screenshot of a cell phone

Description automatically generated

1. 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?

Solution:

The sequence number is 0.

The value of the Acknowledgement field is 1.

gaia.cs.umass.edu computed the value in order to acknowledge the previous SYN packet sent by the client computer.

The segment is identified as SYN ACK because the acknowledgment and syn flags in the packet are set to 1.

A screenshot of a cell phone

Description automatically generated

1. 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.

The sequence number of the POST message is 1.

1. 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 242 in 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. Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the “listing of captured packets” window that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics->TCP Stream Graph- >Round Trip Time Graph.

Solution:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Packet sequence number | Time sent | ACK received | RTT | Estimated RTT |
| 1 | 2.444367 | 2.447832 | 0.003465 | 0.003465 |
| 689 | 2.444451 | 2.504288 | 0.059837 | 0.010509 |
| 3585 | 2.504397 | 2.504669 | 0.000272 | 0.003064 |
| 5033 | 2.504727 | 2.560643 | 0.055916 | 0.072925 |
| 6481 | 2.504728 | 2.564525 | 0.059797 | 0.010504 |
| 7929 | 2.504728 | 2.564956 | 0.060231 | 0.078318 |

The estimated RTT can be calculated by,

Estimated RTT calculated using the below formula -

Estimated RTT = 0.875 \* Estimated RTT + 0.125 \* Sample RTT

Estimated RTT for the first segment is 0.003465.

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

Solution:

|  |  |
| --- | --- |
| Packet sequence number | Length of the packet |
| 1 | 688 |
| 689 | 1448 |
| 3585 | 1448 |
| 5033 | 1448 |
| 6481 | 1488 |
| 7929 | 1488 |

1. 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?

Solution: The minimum amount of available buffer space advertised at the receiver for the entire trace is verified by the first ACK received by gaia.cs.umass.edu. The size of the same is 28960 bytes. This receiver window grows steadily until it reaches the size, 220160 bytes. The sender is never throttled due to lack of receiver buffer space.

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

Solution: To answer the question consider the time sequence number graph,

A close up of a map

Description automatically generated

On analyzing the graph, we can conclude that the sequence numbers have increased over time. If there was a retransmitted packet then the sequence number of the packet would be smaller than the neighboring packets.

1. 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).

Solution:

Consider the acknowledgements of the following segments,

|  |  |  |
| --- | --- | --- |
| Acknowledgment Number | Sequence number | Length of the data |
| ACK1 | 689 | 1448 |
| ACK2 | 3585 | 1448 |
| ACK3 | 5033 | 1448 |
| ACK4 | 6481 | 1448 |
| ACK5 | 7929 | 1448 |
| ACK6 | 10825 | 1448 |
| ACK7 | 12273 | 1448 |
| ACK8 | 13721 | 1448 |
| ACK9 | 15169 | 1448 |

From the traces it can be evaluated that the receiver acknowledges almost each and every packet sent to it.

1. What is the throughput (bytes transferred per unit time) for the TCP connection?

Explain how you calculated this value.

A close up of a map

Description automatically generated

By generating a graph for throughput, we can see how the throughput has averages over the entire time period. If we consider the entire duration, we evaluate the smalled sequence number, 0 to the largest sequence number – 149426.

The total time of transmission is 0.42 seconds.

For the duration of the entire connection the average throughput goes up to 1.2 x 106 bits/s.

1. And 14 . 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.

Consider the graph from tcp-etheral-trace-1,

It is an ideal situation where the slow start phase remains a constant.

A close up of a map

Description automatically generated

Consider the graph for the trace I have captured on my system. The green box shows the end of the slow start phase for the trace. The MSS has increased exponentially and after that the congestion avoidance has come into picture. I have highlighted the same with a red box.

A close up of a map

Description automatically generated