
Computer Networks

Midterm Exam

April 24, 2014

1. Suppose your boss calls you up while visiting a client and says he left a 250,000 byte file on a server at the office, and he needs you to send it to him. The connection between the office and his current site is 2 million bits/second.

1A. [2 points] Assuming that all bandwidth is consumed by your data (e.g., there aren't any packet headers, there's no other traffic on the link, etc.) and that you can immediately start sending data at the maximum rate, how long will it take to transmit the file? (Easy)

*250,000 bytes * (8 bits / 1 byte) * 1/2,000,000 bits/s = 1 second*

1B. [5 points] Now assume that you are sending the file via TCP, with a maximum segment size of 1000 bytes. You have to follow TCP's transport algorithm, so you start transmitting the file at a low rate. How many network round trip times (RTTs) will it take to transmit the entire file? Again assume that all bandwidth is consumed by your data.

TCP needs to first establish a connection via a handshake (SYN/SYN-ACK) - 1 RTT

TCP then enters slow-start to transmit the file, so we run slow-start either until we've transmitted the file or until we experience a loss (at 2Mbps). In slow-start, TCP will initially transmit a window of 1 MSS (=1000 bytes) per RTT, then double the window size each round (RTT). So, we have:

1 MSS, 2 MSS, 4, 8, 16, 32, 64, 128

The sum of the above 8 transfers is 255 MSS = 255,000 bytes > 250,000 byte file, so we declare ourselves finished after the 8th RTT.

Thus, the total time to transmit the file would be 1 + 8 = 9 RTT.

Note: One might debate what "transfer" means, so we also accepted 7.5 RTTs for the number of runs in slow-start, as the final transfer of 128 might not strictly need to ACK. We did not take off any points if people also included 1 RTT for FINs.

1C. [4 points] Let's now consider how TCP reacts when it encounters packet loss (the "basic" variant we learned in class). Your boss now wants you to transmit a

much larger file (say, 1 GB). This time around, after TCP reaches an instantaneously transmission rate of 32,000 bytes per second, you notice that a single packet is lost. Assuming that you don't currently have any other packets in transmission, what is the instantaneous rate TCP will send at after noticing that loss? How long (in terms of RTTs) will it take to reach 32,000 bytes per second again?

This question was meant to seek how TCP reacts to detecting loss via a triple duplicate ACK. In that case, TCP reacts by "MD" -- multiplicative decrease -- cutting its window size by half and continuing by "AI" -- additive increase.

So the next transmission rate is 16,000 bytes per second, and it will take 16 RTTs to recover to its prior levels (each RTT increases by 1 MSS = 1000 bytes).

1D. [4 points] In the previous example of 1c, would anything be different if all packets were lost after you reached 32,000 bytes per second, instead of just a single packet? If so, what is the next instantaneous rate TCP will send at in this example, and how long (in terms of RTTs) would it now take to reach 32,000 bytes per second again?

Because all packets are lost, we only detect loss through a timeout. This causes the sender's window to be reduced to a single MSS (1000 bytes / RTT), at which time it enters "slow-start restart". Namely, it will perform slowstart until half of its prior cwnd (16,000 bytes/second), then do additive-increase from 16,000 to 32,000 bytes/second.

(In fact, the question as written didn't include quite enough information, as you actually do need to know the RTT in this example in order to calculate the window size for "16,000 bytes / seconds", i.e., if RTT is 500ms, window size is 8 MSS/RTT, while window size would be 16 MSS/RTT if RTT is 1s. Everybody just assumed implicitly that RTT=1s, and that the next rate after the timeout was 1000 bytes/sec. We initially had written these questions with an explicit RTT and asked you to solve the question in terms of time, rather than RTTs, but subsequently removed this information to reduce needed calculations.)

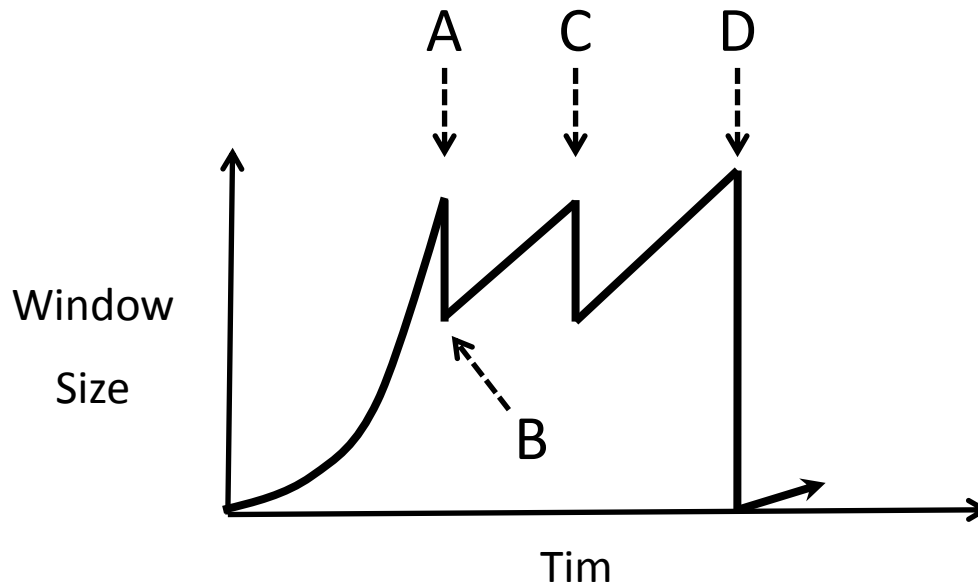
That said, assuming RTT=1s, then the answer is:

Slow-start from 1 MSS to 16 MSS = 4 RTTs, then additive increase 16-32 MSS = 16 RTTs

Total = 20 RTTs (we also accepted 21 RTTs given various ways to "count" these values)

QUESTION 2 : TCP and Congestion Control (19 POINTS)

Consider the following graph of TCP throughput (NOT DRAWN TO SCALE), where the y-axis describes the TCP window size of the sender. We will later ask you to describe what happens on the right side of the graph as the sender continues to transmit.



1. [3 points] The window size of the TCP sender decreases at the points marked by A, C, and D on the graph, which happens when a packet belonging to the stream is lost. Name the event which occurs that causes the sender to decrease its window.

A: Triple Duplicate ACK

C: Triple Duplicate ACK

D: Timeout

2. [4 points] Assume that the network has an MSS of 1000 bytes. If point A occurs 1 second after the sender begins its transfer, and the sender has written 15,000 bytes to the network by that time, what is the round-trip-time (RTT) of the network? Assume at time 0 the sender attempts to open the connection. Also assume that the sender can “write” a full window’s worth of data instantaneously, so the only latency you need to worry about is the actual propagation delay of the network.

First, calculate the number of RTTs needed. 1 RTT for client handshake (SYN → SYN-ACK). Then window size grows during slow start: 1 MSS, 2 MSS, 4 MSS, 8

MSS = 15 MSS = 15,000 bytes. Note that A occurs at the point of detecting triple duplicate ACK, and therefore the client must have seen at least some ACKs from the 8 MSS windows.

Hence a total of 5 RTTs over 1 second = 200 ms

3. [4 points] What is the sender's window size (in bytes) at point B?

After triple duplicate ACK, the sender halves its window size (part of "multiplicative decrease" of AIMD), thus leaving us at $\frac{1}{2}$ cwnd of A = 4 MSS = 4,000 bytes

4. [4 points] If point C occurs 2 seconds after point B, what is the sender's window size (in bytes) at point C?

Network RTT = 200 ms, thus 2 seconds = 10 RTT. The sender's window is currently growing linearly in the "additive increase" part of AIMD, and hence increases by 1MSS every RTT.

Therefore, our prior cwnd was 4 MSS + 10 MSS = 14 MSS = 14,000 bytes

5. [4 points] Assume that the window size at point D is 16,000 bytes, at which time it drops to 1,000 bytes. How much time will it take for the sender to return to a window size of 16,000 bytes (assuming that there is no further packet loss in the network)? You can express your answer in terms of the number of RTTs, as opposed to the number of seconds.

After a timeout, the window drops to 1 MSS. It then proceeds to grow via slow-start (i.e., doubling each RTT) until it reaches $\frac{1}{2}$ of the cwnd at the time it detected the timeout. Afterwards, it grows linearly (+1 MSS / RTT).

Thus, we experience slow-start until 8 MSS, which takes 3 RTT. We then increase linearly until 16 MSS, which takes an additional 8 RTT, giving us a total of 11 RTTs.