IT 304 COMPUTER NETWORKS

Sample Questions

1. Suppose we have a file of 2000KB. The propagation delay is 5 ms and packet size is 1KB, and an initial 2xRTT of "handshaking" before data is sent. Calculate the total time required to transfer the file in the following cases.

   a. The bandwidth is 1.5 Mbps and data packets can be sent continuously
   We have a 2xRTT initial handshake, followed by the time to transmit 2000 KB, and finally the propagation delay (half RTT) for the last bit to reach the other side. Therefore, we have:

   Waiting time= 2RTT. – (1)

   Transmit delay for 2000KB = 2000(1KB*8) / (1.5*10^6) –(2)*

   Propagation delay = 5ms – (3)'

   *Total time = **<u>(1) + (2) +(3)</u>***

   b. The bandwidth is 1.5 Mbps, however we must wait 1 RTT between sending consecutive packets.

   There is a waiting time of 1 RTT between adjacent packet trans- mission. Since the propagation time for a packet is RTT/2, each packet will reach receiver within the waiting time. In other words, the prev packet has reached half the time while waiting to send the next packet. However, we need to consider prop delay for the last packet (total time for the last byte to reach receiver)

   *Initial Handshaking = 2 * RTT -(1)*

   Total number of packets = 2000 (packet size 1KB)

   Network Delay = Transmit time + Prop delay(last packet)

   *= 2000 *[(1KB*8) / (1.5*10^6)] + 5ms –(2)*

   Time to wait between consecutive packets = 1RTT

   After transmitting a packet, we wait for one RTT. So the remaining 1999 packets are sent for every RTT, which is 1999RTT – (3)

   *Total Time = **<u>(1) + (2) + (3)</u>***

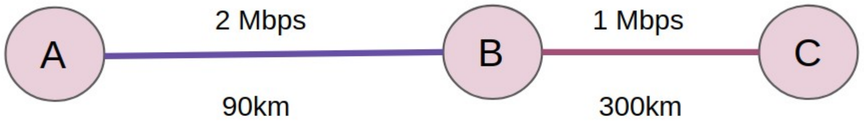
c. Assume the bandwidth is infinite (transmission delay is 0) and 20 packets can be sent for 1 RTT

packet size is 1 KB , so 2000 packets will be transmitted but only 20 per RTT

So, it takes total of 2000 / 20 = 100 RTTs.

Total time = 2RTT + 100 RTT + RTT/2 (prop delay)
= ***<u>102.5 RTT</u>***

2. Consider the diagram below. The links between nodes A-B and B-C are 90km and 300km long respectively. Both links allow packets to be propagated at speed $3 * 10^8$ m/s. The first link has bandwidth 2 Mbps and the second one has bandwidth 1 Mbps.



d1=90km = 90000m; d2 = 300km=300000m

$d_{propA\text{-}B}$ = d1/speed=0.3ms; $d_{propB\text{-}C}$ = d2/speed=1ms;

a. Calculate the time it takes to send a packet of size 1KB from Node A to C and back to Node A? Assume there is no processing delay.

$d_{transA\text{-}B} = 1KB/2Mbps = 8000/2Mbps=4ms$

$d_{transB\text{-}C} = 1KB/1MBps = 8ms$

Total delay$= (d_{transA\text{-}B} + d_{propA\text{-}B}) + (d_{transB\text{-}C} + d_{propB\text{-}C})$

b. Assume a packet of size 1KB is sent from Node A to C. Immediately following, a packet of size 3KB is sent from Node A to C as well. How long would it take for Node C to receive the second packet?

$d_{transA\text{-}B(P1)} = 1KB/2Mbps; = 8000/2Mbps=4ms$

$d_{transA\text{-}B\ (P2)}= 3KB/2Mbps; = 24000/2Mbps=12ms$

$d_{transB\text{-}C(P1)} = 1KB/1Mbps= 8ms;$

$d_{transB\text{-}C(P2)} = 3KB/1Mbps = 24ms;$

A can send P1 followed by P2. So

t=0, transmission of P1 (A-B link)

t= 4ms, transmission of P2 (P2 is ready to be put on link, after 4ms of P1 being put) *t= 4ms+0.3ms, last bit of A reaches B ------1KB transmit time + prop distance of 90km*

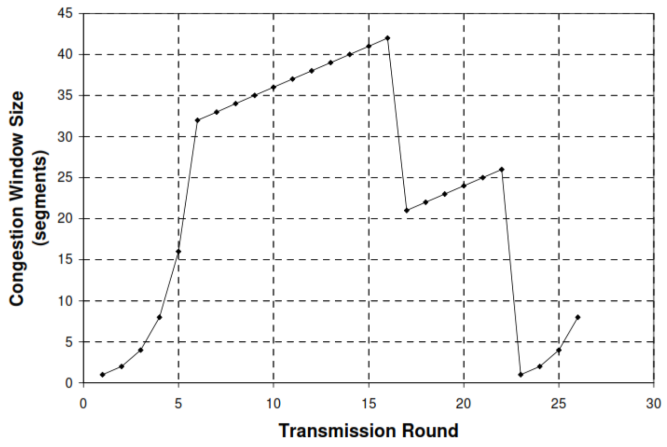
At t=4.3ms , P1 ready to be sent from B to C -----1KB packet with 1Mbps

After 4.3+ $d_{transB\text{-}C(P1)}$ + $d_{propB\text{-}C}$, P1 reaches C

At t=16.3ms (4+0.3+12), last bit of P2 reaches B ----------3KB transmit time + prop distance of 90km + 4ms

At t=16.3 + $d_{transB\text{-}C(P2)}$ + $d_{propB\text{-}C}$, P2 reaches C

3. Consider the following plot of TCP window size as a function of time:



Assuming TCP Reno is the protocol experiencing the behavior shown above, answer the following questions.

(a) Identify the intervals of time when TCP slow start is operating.
Solution: 1-6, 23-26

(b) Identify the intervals of time when TCP congestion avoidance is operating (AIMD).
Solution: 6-23

(c) After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
Solution: 3 dupack

(d) What is the initial value of ssthreshold at the first transmission round?
Solution: 32

(e) What is the value of ssthreshold at the 18th transmission round?
Solution: 21

(f) What is the value of ssthreshold at the 24th transmission round?
Solution: 13

(g) During what transmission round is the 70th segment sent?
Solution: 7

(h) 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 congesion-window size and of ssthreshold?
Solution: 4,4

4. At time t, a TCP connection has a congestion window of 4000 bytes. The maximum segment size used by the connection is 1000 bytes. What is the congestion window after it sends out 4 packets and receives acks for all of them? Suppose there is one ack per packet.
   a. If the connection is in slow-start?
      Solution: 8000 bytes. In slow start, the sender increases its window for each byte successfully received.

   b. If the connection is in congestion avoidance (linear mode)?
      Solution: 5000 bytes. The sender increases its window by one segment each window.

5. Using TCP, a sender has sent out a total of ten data segments, each of 1000 bytes. Assume that the sender's initial sequence number (ISN) is 5000. If a TCP segment were to include 1000 bytes with sequence numbers 20,001 through 21,000, you can refer to the segment simply as "segment 21,000". Answer the following:
   a. After the sender receives a packet with "ACK 8001", how many TCP segments, if any, does the sender know that the receiver definitely received?

      3 packets (6000, 7000, 8000)

   b. The sender subsequently receives two more packets with "ACK 8001". Given this occurrence, which of the TCP segments, if any, does the sender conclude might be lost and thus should be retransmitted?
      8001-9000

6. In socket programming, lets say you use *write* in your code and execute the following line:

   write (fd, buf +2000, 1000)

   Remember that the *write* system call takes a file descriptor, a pointer to a memory region, and the length (in bytes) of data from that region to write to the file descriptor. Answer the following questions:
   a. Which byte range are you trying to transfer?
      Bytes 2000-2999 (inclusive) in buf.

   b. If this write call returned 500, what does this signify?
      Only bytes 2000-2499 were written

   c. Why might this occur?
      The socket buffer only had space for 500 bytes, not the full 1000. The application should call write again with write (fd, buf+2500, 500);

   d. If you issue close() on the socket *before* the operating system finishes transmitting the data, will the remaining data be sent to B? (why or why not)?
      yes, a close() will flush any data remaining in the buffer before it actually tears down the connection

7. Suppose that the three MeasuredRTT values are 106ms, 120ms, 140ms. Compute the EstimatedRTT after each of these MeasuredRTT values is obtained, using a value of alpha = 0.125 and assuming that the value of EstimatedRTT was 100ms just before the first of these 3 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 5ms just before the first of these three samples was obtained. Last, Compute the TCP TimeoutInterval after each of these samples is obtained.

   EstimatedRTT = xSampleRTT + (1 - x) EstimatedR TT

   DevRTT = y SampleRTT - EstimatedRTT + (1 - y) DevRTT

   *TimeoutInterval = EstimatedRTT + 4 * DevRTT*

   *After obtaining first sampleRTT is EstimatedRTT = 0.125 * 106 + 0.875 * 100 = 100.75ms*

   *DevRTT = 0.25 * 106 - 100.75 + 0.75 * 5 = 5.06ms*

   *TimeoutInterval = 100.75 + 4 * 5.06 = 120.99ms*

   After obtaining second sampleRTT = 120ms:

   *EstimatedRTT = 0.125 * 120 + 0.875 * 100.75 = 103.15ms*
   *DevRTT = 0.25 * 90 - 105.54 + 0.75 *14.06 = 14.42ms*

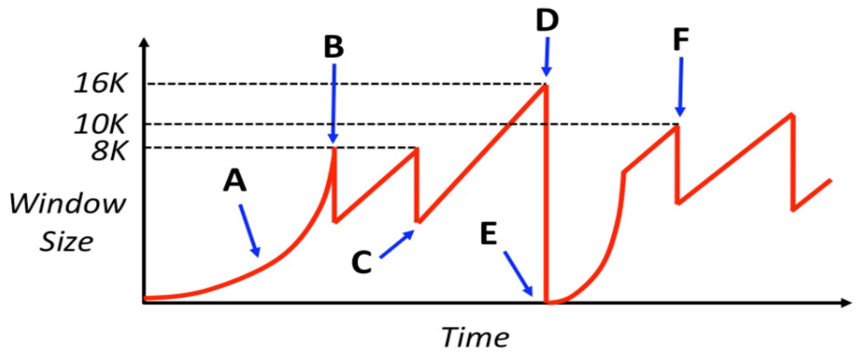
   *TimeoutInterval = 105.54 + 4 *14.42 = 163.22ms*
   After obtaining Third sampleRTT = 140ms:

   *EstimatedRTT = 0.125 * 140 + 0.875 * 103.15 = 107.76ms*

   *DevRTT = 0.25 * 140 - 107.76 + 0.75 * 8 = 14.06ms*

   *TimeoutInterval = 107.76 + 4 *14.06 = 164ms*

8. Consider the following graph of TCP throughput, where the y-axis describes the TCP window size of the sender.



The window size of the TCP sender decreases at several points in the graph, including those marked by B and D. Answer the following questions.

a. Name the event at B which occurs that causes the sender to decrease its window.
Triple Dup Ack

b. Name the event at D which occurs that causes the sender to decrease its window.
Time out

c. What event occurred at F?
Triple Dup Ack

d. What is the value of ssthreshold at event F?
5K

9. Suppose that a sender and a receiver are using ARQ to perform reliable data delivery.

a. How many sequence numbers are needed to implement Stop-and-Wait?
Two sequence numbers

b. Assume Go-Back-N protocol. Let the window size be six. The unacknowledged but sent segments are 1 to 5. Yet to be sent segments are 6, 7, 8, 9 and 10. Now, receiver sent an ACK for segment two. Which segment(s) can the sender send before it must wait for the next ACK from the receiver?
Assuming segments 1 through 5 have been sent, the window size of 6 still allows one more segment (segment 6) to be transmitted. When the ACK for segment 2 is received, that opens up the window by an additional 2 segment which means that segment 7,8 can also be sent

c. After a while segment 20 to 26 were sent. But during the transit, segment 22 was lost. If Go-Back- N is used, what segment(s) would the sender have to retransmit?
Assuming that the ACKs for 20 and 21 are received, the sender would have to retransmit 22, 23, 24, (25), and 26.

d. Suppose the same situation as above but sender and receiver use Selective-Repeat ARQ. What segment(s) would the sender need to retransmit?
Again, assuming that ACKs for 20 and 21 are received, only the lost segment will need to be retransmitted in Selective-Repeat ARQ. (25 and/or 22)

10. Consider sending a large file from a host to another over a TCP connection that has no loss. Suppose TCP uses AIMD for its congestion control w/o slow start. Assuming cwnd increases by 1MSS every time a batch of ACKs is received and assuming approximately constant roundtrip times.

a. How long does it take for cwnd increase from 6MSS to 12MSS (Assuming no loss events)?

It takes 1 RTT to increase CongWin to 6 MSS; 2 RTTs to increase to 7 MSS; 3 RTTs to increase to 8 MSS; 4 RTTs to increase to 9 MSS; 5 RTTs to increase to 10 MSS; 6 RTTs to increase to 11 MSS; and 7 RTTs to increase to 12MSS.

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

In the first RTT 5 MSS was sent; in the second RTT 6 MSS was sent; in the third RTT 7 MSS was sent; in the fourth RTT 8 MSS was sent; in the fifth RTT, 9 MSS was sent; and in the sixth RTT, 10 MSS was sent. Thus, up to time 6 RTT, 5+6+7+8+9+10 = 45 MSS were sent (and acknowledged). Thus, we can say that the average throughput up to time 6 RTT was (45 MSS)/(6 RTT) = 7.5 MSS/RTT.

11. State True or False (1 point each)

A user requests a Web page that consists of some text and three images. For this page,

a. the client will send one request message and receive four response messages. False
b. Two distinct Web pages (for example, www.mit.edu/research.html and www.mit.edu/students.html) can be sent over the same persistent connection. True
c. The Date: header in the HTTP response message indicates when the object in the response was last modified. False
d. HTTP response messages never have an empty message body. False