CIS-427

HW2

With Dr. Zheng Song

Student: Demetrius Johnson

October 13, 2022

CIS-427 HW2 FALL 2022 – Demetrius Johnson

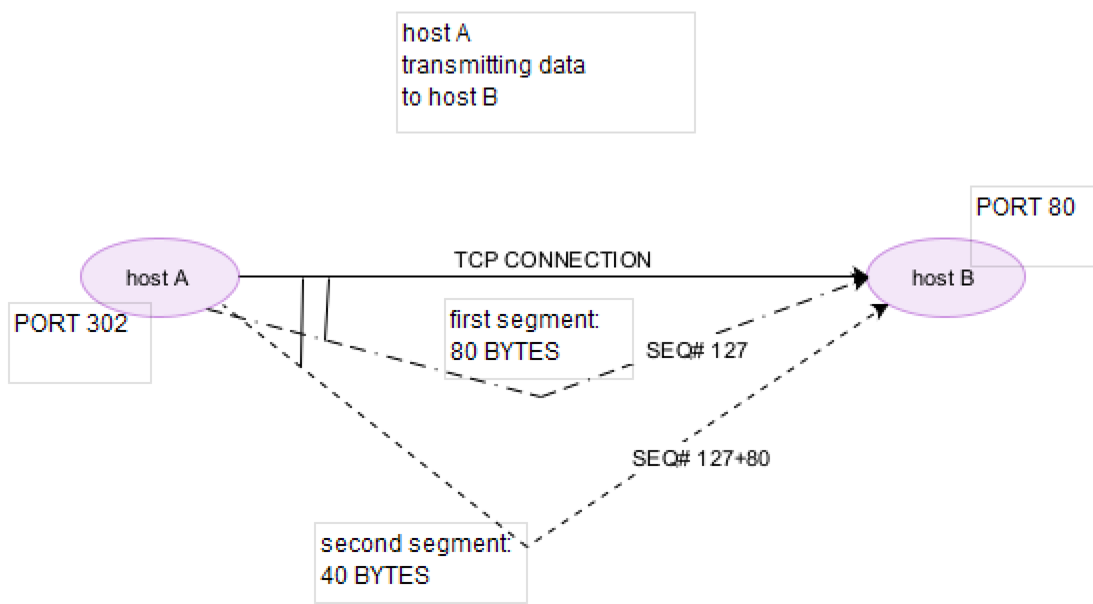
Homework assignment 2, CIS 427, Fall 2022

Submission Due: 11:59 PM, 10/13/2021.

Question 1

1. **28 points, 6 points each for a, b, and c, 10 for d.**
   Hosts A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

***The following diagram I drew using Visual Paradigm to represent the question:**



a - In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?

- Source and destination port over this TCP connection is the same for all segments sent from A to B (B to A transmission is simply the reverse source and destination IP and Port numbers).
  - Thus: for A→B transmission, source port = 302, destination port = 80.
- The second segment to be transmitted will have a sequence number of the previous starting byte stream from previously sent segment + total number of bytes sent in that stream 127 + 80 = 207.
- Summary: **second segment → source port=302, destination port=80, sequence number=207.**

b - If the first segment arrives before the second segment, in the acknowledgment of the first arriving segments, what is the ACK number, the source port number. and the destination port number?

- If the segments arrive in order to B, and the first segment is acknowledged (by host B), then:
  - Assuming previous ACK sent by B was 127 (meaning that B received all bytes through byte 126, acknowledging and is ready for next byte stream beginning at byte 127).
  - First segment sent by host A is SEQ#=127 and will send 80 bytes (meaning that A is sending a byte stream of 80 bytes, beginning at byte 127).
  - Now after A's first segment is received, 126+80 bytes = 206 bytes have been received by host B, meaning host B will acknowledge it is ready for the next byte stream, beginning at byte 206+1 = 207 = (127+80).
  - Thus when host B sends an ACK for the first segment sent from A that it received, then
    - **Source port = 80, destination port = 302, ACK=207.**

c - If the second segment arrives before the first segment, in the ACK of the first arriving segment, what is the ACK number?

- According to RFC 5681

TCP Receiver: ACK generation [RFC 5681]

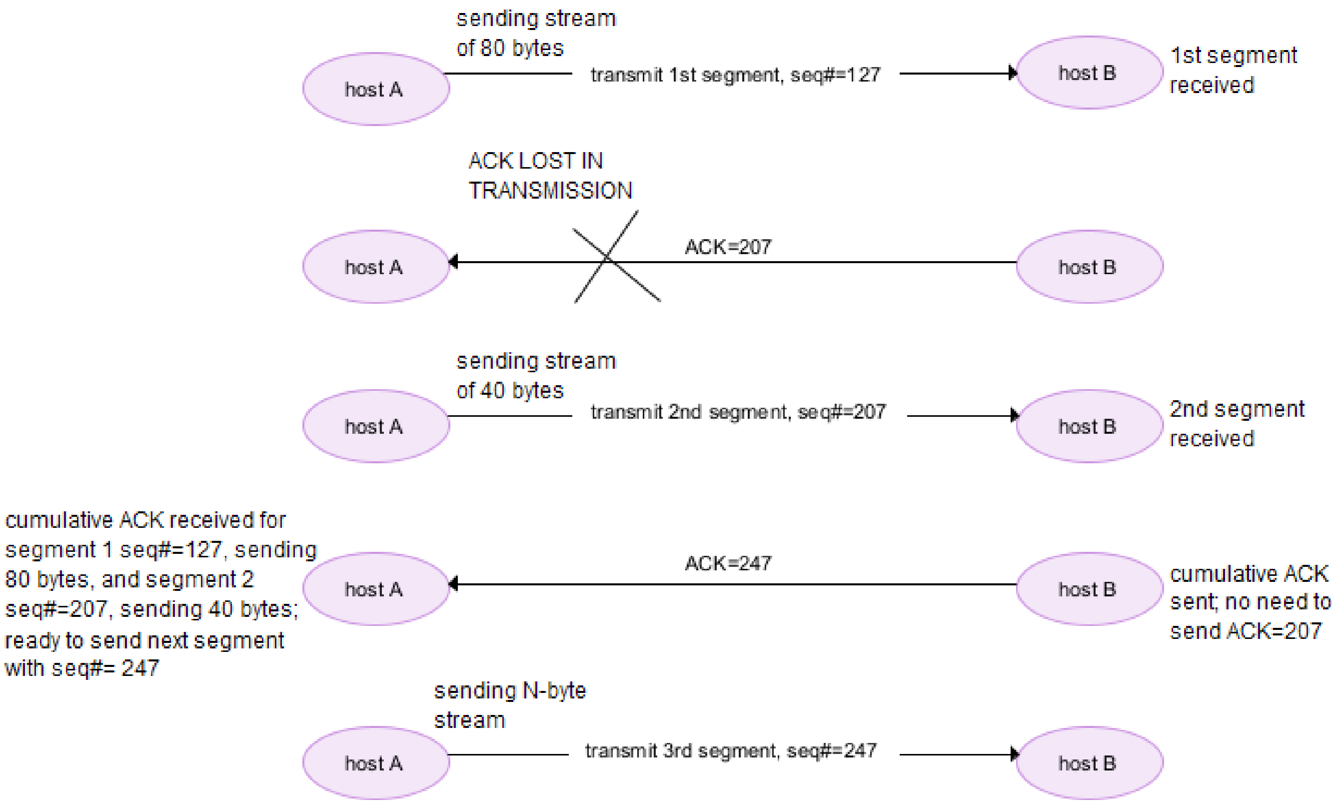
Event at receiver	*TCP receiver action*
arrival of in-order segment with expected seq #. All data up to expected seq # already ACKed	delayed ACK. Wait up to 500ms for next segment. If no next segment, send ACK
arrival of in-order segment with expected seq #. One other segment has ACK pending	immediately send single cumulative ACK, ACKing both in-order segments
arrival of out-of-order segment higher-than-expect seq. # . Gap detected	immediately send *duplicate ACK,* indicating seq. # of next expected byte
arrival of segment that partially or completely fills gap	immediate send ACK, provided that segment starts at lower end of gap

1. **acknowledgements are only sent for every two segments received**
2. **Cumulative ACK**

-
- Thus, if the second segment arrives before the first segment, then we have a scenario where an out-of-order arrival of a segment with a higher-than-expected seq.# gap detected, thus host B will immediately send a duplicate **ACK=127** since it is still expecting the next byte stream starting at byte 127 to be sent (it is still expecting the first segment).

d – Suppose the two segments sent by A arrive in order at B. The first acknowledgement is lost and the second acknowledgement arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgment sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number.

***The following I drew using Visual Paradigm:**



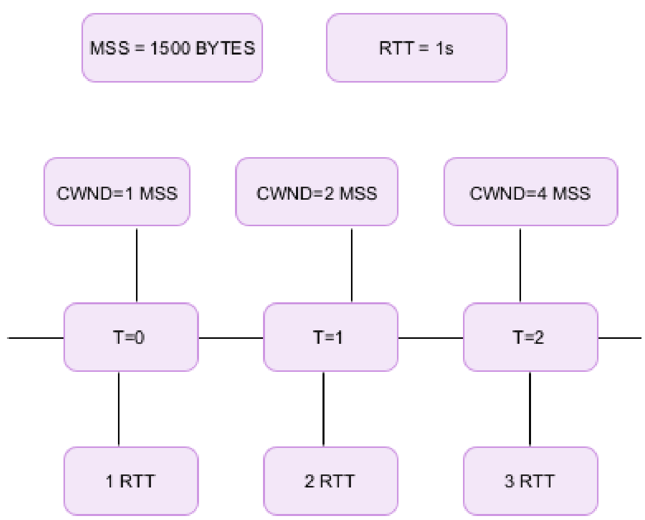
Question 2

2. **40 points, 10 points each**

   Consider a TCP flow over a 1 Gb/s link with a round-trip time of 1 second that transfers a 10 MB file. The receiver advertises a receiver window (rwnd) size of 1MB, and the sender has no limitation on its congestion window (i.e., no ssthresh or ssthresh is set to a very high value and will not affect this transmission). Assume the TCP packet size is 1500 Bytes (i.e., MSS).

a- If the initial send window starts from 1MSS, how many transmission rounds does it take until slow start opens the send window to 1 MB?

- Summary of situation:
  - 1Gbps link
  - RTT=1s
  - 10MB file = (10*10^6)bytes * 8bits/byte = (80*10^6)-bit file.
  - Receiver window, rwnd= 1MB = (1*10^6)bytes * 8bits/byte = (80*10^6) bits window.
  - No limit to size of sender control window (cwnd is very large, so sender has no limitation).
  - TCP segment size (MSS) = 1500 bytes = (8*1500) bits.
- Slow start increases the rate exponentially packet-wise (segment-wise) until first loss event.
- ***Note: cwnd is just a factor of MSS; so cwnd = 1 is the same as cwnd = 1 MSS***
- Initially, cwnd = 1 MSS = 1*1500bytes = 1500 bytes.
- We double cwnd *every RTT (1 RTT = 1s, so cwnd is doubled every 1 second)* by incrementing cwnd for every ACK received (thus every RTT cwnd= → 2*MSS, 4*MSS, 8*MSS; thus number of ACKs received will grow exponentially by cwnd*2^(RTT - 1); thus we increment cwnd for each ACK received so as to make transmission rate to grow exponentially as well by
  - cwnd = cwnd * 2^(RTT-1).
    - Note, we do RTT-1 above since at time t = 0, RTT = 1, and cwnd=1 MSS to start; so in the first RTT, the window size = cwnd = 1 MSS is transmitted. *Number of RTTs must always be >= 1.*



    - The diagram above I generated using Visual Paradigm.
- We know that TCP transmission rate is (approximately)= (cwnd/RTT) bytes/s, since the protocol is to wait RTT seconds for ACKS to be received (and thus cwnd incremented for all ACKs received) before transmitting more bytes (for congestion control).
  - Thus cwnd is (approx)= RTT*TCP_transRate
- Thus:

- If initial cwnd = 1 MSS, MSS=1500bytes, 1 RTT=1s; we need to solve for RTTs = number of transmission rounds, so that TCP_transRate = 1MB
- cwnd grows exponentially by cwnd*2^(RTT - 1)
- thus, [updated cwnd] = cwnd*2^(RTT - 1) = 1500bytes * 2^(RTT - 1).
- now solve for RTT needed to reach sender control window size (transmission rate) → 1MB
- cwnd*2^(RTT - 1) = 1500bytes * 2^(RTT - 1) = 1MB
    - 2^(RTT - 1) = (1*10^6) bytes / 1500 bytes =
    - log_2(2^(RTT - 1)) = log_2((1*10^6) bytes / 1500 bytes)
    - RTT - 1 = log_2((1*10^6) bytes / 1500 bytes)
    - now use change of base formula
    - RTT - 1 = log((1*10^6) bytes / 1500 bytes) / log(2)
    - RTT - 1 = 9.38
    - RTT = 10.38
    - **If RTT = 10, THEN cwnd = 1 * 2^(10-1) MSS =1 * 2^(9) * 1500 BYTES = 768000 BYTES = 0.768MB**
    - **If RTT = 11, THEN cwnd = 1* 2^(11-1) MSS = 1*2^(10) * 1500BYTES = 1536000 BYTES = 1.536MB.**
    - **Thus, we need 11 RTTs to reach transmission rate of *at least* 1MB.**

b- How many transmission rounds does it take to send the file?

- Summary of situation:
    - 1Gbps link
    - RTT=1s
    - 10MB file = (10*10^6)bytes * 8bits/byte = (80*10^6)-bit file.
    - Receiver window, rwnd= 1MB = (1*10^6)bytes * 8bits/byte = (80*10^6) bits window.
    - No limit to size of sender control window (cwnd is very large, so sender has no limitation).
    - TCP segment size (MSS) = 1500 bytes = (8*1500) bits.
    - [updated cwnd] = cwnd*2^(RTT - 1) = 1500bytes * 2^(RTT - 1)
- We need to do a summation of all rounds that will = at least 10MB.
    - → SUM_[from RTT=1 to RTT=x]{cwnd*2^(RTT - 1)} >= 10MB
    - → SUM_[from RTT=1 to RTT=x]{1500bytes*2^(RTT - 1)} >= 10MB
    - First, we can simplify by pulling out 1500 factor:
    - → 1500bytes * SUM_[from RTT=1 to RTT=x]{2^(RTT - 1)} >= 10MB
    - → SUM_[from RTT=1 to RTT=x]{2^(RTT - 1)} >= 10MB / 1500 bytes
    - Now I will use DESMOS Calculator to find out how many RTTs there needs to be, where n = RTT:

- $$1500 \cdot \sum_{n=1}^{12} 2^{(n-1)} = 6142500$$
- $$1500 \cdot \sum_{n=1}^{13} 2^{(n-1)} = 12286500$$
- **We need enough RTTs to send *at least 10MB; thus:***
- **It will take 13RTTs to send the 10MB file.**

c- If the time to send the file is given by the number of required transmission rounds multiplied by the RTT of the link, what is the effective throughput of the transfer?

- 1 RTT = 1s; we have 13 RTTs needed to send the file
  - Thus: it will take 13 RTT * 1sec/RTT = 13 seconds to send the file.
- If we send 10 MB in 13 seconds then:
  - 10*10^6BYTES * 8bits/byte / 13s = **6153846 bps = 6153Kbs = 6.15Mbps.**

d- What percentage of the link bandwidth is utilized?

- the link has 1 Gbps bandwidth
- thus just take throughput / bandwidth to find bandwidth utilization
- 6.15*10^6 bps / 1*10^9 bps = 0.00615 * 100% = **0.615% utilization.**

Hints: Throughput tells you how much data was transferred from a source at any given time and bandwidth tells you how much data could theoretically be transferred from a source at any given time.

Question 3

3. **32 points,4 points each**
   Assuming TCP Reno is the protocol experiencing the behavior shown in the Figure. Answer the following questions:

a- Identify time intervals where TCP slow-start is operating.

- Slow start is operating at intervals [6,16) and [17, 23)

b- identify time intervals where TCP congestion-avoidance is operating

- At intervals (16, 18) and (22,24)

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

- Triple ACK

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

- Timeout.

e- What is the ssthreshold value at the first transmission round?

- Cwnd=32

f- Ssthreshold at 18th transmission round?

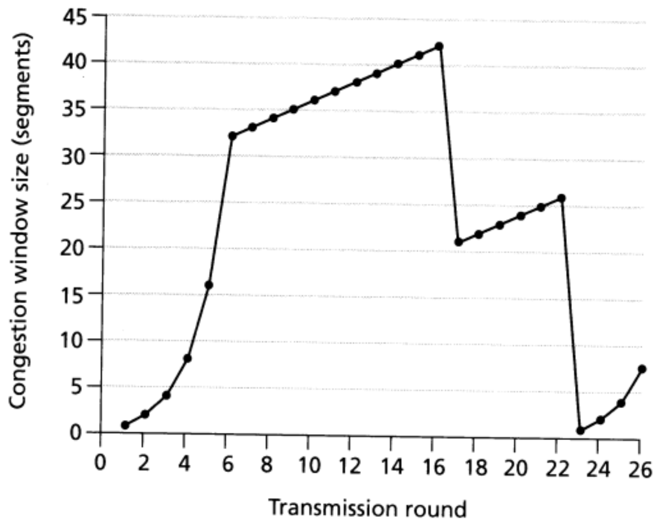
- Cwnd=26

g- Ssthreshold at 24th transmission round?

- Cwnd = 13

h- Cwind and ssthreshhold at 26th transmission round after receiving triple duplicate ACKs?

- Cwind and ss thresh = 4


Congestion window size (segments)
45
40
35
30
25
20
15
10
5
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26
Transmission round