Problem 1

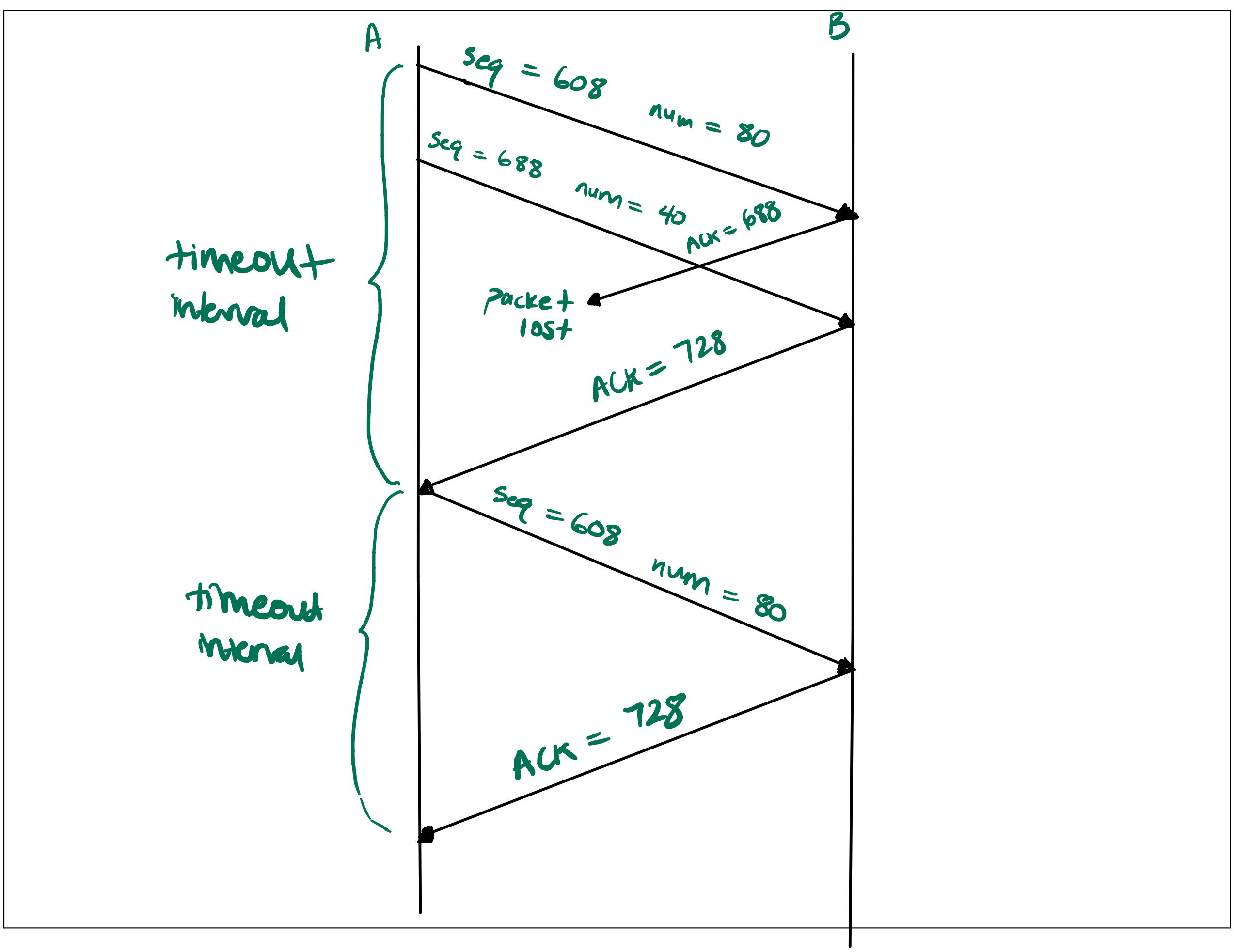
Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 607. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 608, the source port number is 35214, and the destination port number is 8080. Host B sends an acknowledgment whenever it receives a segment from Host A. Fill in the blanks for questions (a) – (c) directly; work out the diagram in the box for question (d).

(a) In the second segment sent from Host A to B, the sequence number is 688, source port number is 35214, and destination port number is 8080.

(b) If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, the ACK number is 688, the source port number is 8080, and the destination port number is 35214.

(c) If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, the ACK number is 608.

(d) Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after A's timeout intervals for both the first and the second packets. Draw a timing diagram in the box below, showing these segments and acknowledgments until A receives all the acknowledgments of re-transmitted packets. Assume no additional packet loss. For each segment in your diagram, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the ACK number.



Problem 2

Compare Go-Back-N (GBN), Selective Repeat (SR), and TCP (no delayed ACK). Assume that timeout values for all three protocols are sufficiently long, such that 8 consecutive data segments and their corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (Host A), respectively. Suppose Host A sends 8 data segments to Host B, and the 4th segment (sent from A) is lost. In the end, all 8 data segments have been correctly received by Host B. Supposing for TCP, the first packet starts from sequence number 1 and each packet contains 1 byte.

(a) How many segments has Host A sent in total and how many ACKs has Host B sent in total? What are their sequence numbers? Answer this question for all three protocols.

(b) If the timeout values for all three protocols are very long (e.g., longer than several RTTs), then which protocol successfully delivers all 8 data segments in the shortest time interval?

a)

GBN

8 is sent, 4th lost
so resend 4→8 ⇒
8 + 5 = 13 segments
1, 2, 3, _, 3, 3, 3, 3, 4, 5, 6, 7, 8 = 12 ACKS

SR

8 is sent, and resend 4th ⇒ 9 segments
1, 2, 3, _, 5, 6, 7, 8, 4 ⇒ 8 ACK's

TCP

8 is sent, resend 4th ⇒ 9 segments
4 4 4 4 4 9 ⇒ 6 ACKS

b) TCP — it uses fast re-transmit without waiting for any timeouts.

Problem 3

In the Fast Retransmit algorithm, we saw TCP waits until it has received three duplicate ACKs before performing a fast retransmit. Why do you think the TCP designers chose not to perform a fast retransmit after the first or second duplicate ACKs for a segment received?

TCP performs a fast retransmit only after 3 duplicates to minimize the possibility of unnecessary retransmissions. If packets arrive out of order, waiting for 3 duplicates reduces the chance of mistaking out of order delivery for actual packet loss, which would then trigger a fast retransmission and create redundant data transmission.

Additionally, it helps with maximizing throughput and minimizing congestion by using the state of the network to infer transmission rate.

Problem 4

Suppose that instead of a multiplicative decrease, TCP decreased the window size (upon timeout or 3 duplicate ACK) by a constant amount. Would the resulting AIAD algorithm converge to an equal share (fair) algorithm?

No, the decrease from the AIAD algorithm would not converge to a fair share algorithm. Since the decrease is constant, the decrease would be the same ~~for~~ all connections. If one connection has a connection with a larger window size, it would experience a smaller relative decrease in comparison with a window with a smaller window size, resulting in un-even bandwidth allocation, could result in inefficiencies and instability.

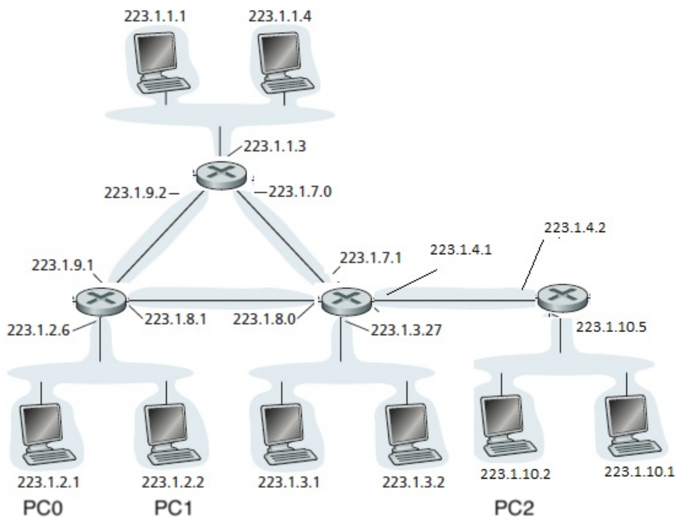
Problem 5

In this problem, we'll explore the impact of NATs on P2P applications. Suppose a peer with username Arnold discovers through querying that a peer with username Bernard has a file it wants to download. Also, suppose that Bernard and Arnold are both behind a NAT. Try to devise a technique that will allow Arnold to establish a TCP connection with Bernard without application-specific NAT configuration. If you have difficulty devising such a technique, discuss why.

Establishing a TCP connection between Bernard and Arnold would be difficult. A TCP connection would require the two hosts knowing each other's IP addresses and port numbers, but since they are behind separate NATs, their IP addresses and port numbers would be modified and sometimes never sent.

Problem 6

Consider the network shown below.



(a) How many subnets are in this network?

(b) What information in DHCP will be exchanged if PC0 moves to PC2's network?

a) 4 subnets

b) The address of PC0 that is newly assigned if PC0 moves to PC2's network