

# Assignment 2: Hands-on with the Transport Layer

## Exercise 1

Beginning at time 0, when the first stack of segments (i.e., round 1) is issued, through time the eighth stack of segments (i.e., round 8) is issued, how does the send grow? That is, how does the number of bytes (and segments) sent in round  $i$  to the number sent in round  $i - 1$ ?

The number of bytes sent in round  $i$  is the twice the number of bytes sent in round  $i - 1$ . The first few rounds had approximately the following values: in round 1, 8 bytes were sent; in round 2, 16 bytes were sent; in round 3, 32 bytes were sent; etc. Note that these are approximate amount and the actual number of packets sent varied a little bit. However, based on this trend I can conclude that the number of bytes doubles in each round.

## Exercise 2

Based on your response to the previous problem, what congestion control state would say that the sender is in during the sending of these first 8 rounds?

The congestion control is in the slow start phase because of the exponential growth. In each RTT, the CWND (number of bytes that can be sent) doubles, indicating an exponential increase. This means that for each packet sent, the CWND increments by one.

## Exercise 3

How does the idle time change as the rounds increase? Briefly explain why.

The idle time does not change in each RTT. This means that the network is able to efficiently handle higher traffic load efficiently. Although there may be more packet, and therefore ACKs to send, efficient processing on the receiver end means that the sender does not have to wait longer.

## Exercise 4

Explain what the graph will look like if the current pattern holds.

The size of the stack of segments will continue to grow exponentially until it reaches the slow-start-threshold (`ssthresh`), where it switches to congestion avoidance mode. Then the segment stack size will then grow linearly. In congestion avoidance mode, the window will only grow by 1 each time an ACK is received.

However, if we run into a loss event, then `ssthresh` will update to be half of the congestion window, and the congestion window will be reset to the maximum segment size (`mss`). This means that the segment size will go back to growing exponentially but starts at the same size of the stack from round 1.

This pattern will repeat so the size of the segment will eventually be more consistent. Overall, it will continue the pattern of growth and flat idle until all the packets have been transmitted.