

#### Part A:

The number of TCP flows was calculated by finding the number of SYN packets sent. The flow's source port, source IP address, destination port, and destination IP address were found from the sent SYN packet as well. All of the packets in each flow were stored in to the Flow data structure in a packets array, which allows for finding transactions easier. For finding transactions, I wrote a function that takes as input a packet, and finds the corresponding packet that is sent by the receiver to the sender. I return both of these as a tuple to represent the transaction. The sender throughput is defined as the number of bytes transferred over the time between the first and last packet from the sender. I have calculated the number of bytes transferred by summing the length of the tcp packet for all packets that have the same source port as the flow, are not after the FIN packet from the sender, and are after the three-way handshake protocol. The start time is calculated as the first sent packet after the threeway handshake and the end time is the timestamp of the the FIN packet. Throughput is calculated by dividing the number of bytes by the difference in time.

#### Part B:

My estimation of the congestion window size is based on the RTT calculated from the time difference of the sent SYN to the received SYN/ACK. To find the congestion window I check how many packets are being sent within the calculated RTT timeframe. The congestion window appears approximately to double for the first two flows. The third flow has an increase in size, but it seems to taper off from the second to third congestion windows. The triple duplicate acks are based on the number of sequence numbers that appear multiple times and the ack numbers that appear multiple times. I calculated the intersection of the two lists to get the number of triple acks. The timeout retransmissions are calculated by subtracted the number of triple ack retransmissions by the total number of calculated retransmissions (based on the sequence-duplicate\_count dictionary).