Michael Lee

CSE 310 Assignment 2

Summary of analysis_pcap_tcp code

The main() function goes through all important implementations of helper functions to complete part A and B of this assignment.

1. filter_packets(): This function takes the raw input of the packets read using the Scapy library. It will sort the packets by their unique TCP flow, each flow going into a list() of connections. The number of TCP flows is found by len(connections)
2. analyze_tcp_flows(): This function solves the part A of the assignment.
    a. For each TCP flow the unique four element tuple address is found by taking the first packet of each TCP flow and dissecting it using the Scapy library.
    b. The first two transactions' Seq. number ACK number and Window size is found by dissecting the first two transactions (after the three-way handshake) in each TCP flow using the Scapy library.
    c. The sender throughput is calculated summing the total number of data in bytes and dividing by the total time taken for the TCP flow. The total number of data is calculated by finding the size of each packet and subtracting the IP header size and TCP header size. The time is calculated using the .time functionality from Scapy library.
3. print_congest_window_sizes(): This function solves part 1 of part B. It calculates the congestion window segment size by looking at the first three transactions since the handshake. Until receiving an ACK from the receiever, the congestion window segment is increased. The congestion window size is grown until it hits the ssthresh or retransmits. From assignment2.pcap, it can be seen that the congestion window size usually starts large and decreases to a lower segment size indicating that it has hit retransmission.
4. print_retransmissions(): This function solves part 2 of part B. It traverses each packet of each TCP flow and checks if the Sequence number has previously been repeated. If so, it checks for any triple duplicate ACKs to determine the type of retransmission (triple ack

or timeout). I had a problem of detecting which sequence number repeat was a timeout retransmission and which is a out-of-order sequence and wasn’t able to differentiate the two in my final code. (**NOTE):** This function also takes a considerable amount of time to compute as my data structure is a list within a list and doesn’t use any hashtable. As such, the traversing of every packet and checking for repetitions make take O(n^2) and on my computer, took around 4-6 minutes to compute.