**Homework 2: HTTP, TCP, and Wireshark**

**Part A Wireshark Programming Task**

1. **High level view of the ‘analysis\_pcap\_tcp’ code:** We have created two class objects – IP headers and TCP headers. For each packet, we have captured the required data such as source port, destination port, source IP, destination IP, flags, time stamp, and window size. Then we followed these steps:
2. We have calculated the total TCP connections/ flows.We have collected source port and destination ports for each flow.
3. For first two transactions we took the sequence number, ack number, and window size from both sent and received packets
4. We have calculated the through put, loss rate, RTT.
5. We calculated the congestion window size for first 10 acknowledgements using TCP Tahoe protocol
6. Finally, we calculated the total retransmissions due to time out and due to triple duplicate acks
7. analysis\_pcap\_tcp.py file consists of the code for the implementation of above flow.
8. **Brief note about the estimation each value:**
   1. **TCP flows count** is 3 – [Sender port, Receiver port]: [43498, 80], [43500, 80], [43502, 80], For each flow we have set three flags – SYN FLAG, SYN-ACK FLAG, and FIN FLAG. If all the three flags are set, then a TCP connection is setup and closed. We maintain a dictionary (Key is tuple of source port and destination port and value is a list of flags) to collect these values and count the length of the dictionary.
9. [**Sequence number, Ack number, and Receive Window size**]:

(43498, 80): [[705669103, 1921750144, 49152], [705669127, 1921750144, 49152]],

(80, 43498): [[1921750144, 705669127, 49152], [1921750144, 705670575, 49152]],

(43500, 80): [[3636173852, 2335809728, 49152], [3636173876, 2335809728, 49152]],

(80, 43500): [[2335809728, 3636173876, 49152], [2335809728, 3636175324, 49152]],

(43502, 80): [[2558634630, 3429921723, 49152], [2558634654, 3429921723, 49152]]}

(80, 43502): [[3429921723, 2558634654, 49152], [3429921723, 2558636102, 49152]]}

For each flow, the values are as follow:

For each TCP flow, pick the first two transaction packets – sent and acknowledgement packets and collect the sequence number, acknowledgement number, and window size from the packets. Sequence number in the sent packet consists of the byte number to be sent and acknowledgement number consists of byte number the sender is expecting in the response. Similarly sequence number in received packet consists of byte number present in the packet and acknowledgement number is the byte number the receiver is expecting to receive.

Window size in the sender’s packets corresponds to the buffer size at sender end and window size in acknowledgement packets corresponds to the buffer size at the receiver end.

1. **Through Put:**

<src port, dst port, through put>

< 43498 , 80, 5.251391112912558 bytes/sec >

< 43502 , 80, 1.4815063848257195 bytes/sec >

< 43500, 80, 1.285420726825806 bytes/sec >

For each flow, the through put is calculated by summing up the length of all sent packets and dividing by the time difference between first sent packet and last sent packet.

1. **Loss Rate:**

<src port, dst port, loss rate>

< 43498 , 80, 0.00042998423391138463 >

< 43500 , 80, 0.013299377475947893 >

< 43502 , 80, 0.0 >

For each flow, check if the TCP packet is sent packet, if so increment the count. From those sent packet get the packets with sequence number not repeated before. Divide the unique sequence packets count with total packets sent. Finally, subtract this value from 1.

1. **Average RTT:**

<src port, dst port, RTT>

< (43498, 80) , 80, 0.0733245085029639 sec >

< (43502, 80) , 80, 0.07208042657166197 sec>

< (43500, 80) , 80, 0.13018349359578762 sec>

Get the time difference between each sent packet and its respective acknowledgement packet. Now sum all these time differences and divide by the number of time differences calculated.

Empirical Through Put for TCP Flow (43498 , 80): 5.25 bytes/msec

Theoretical Through Put for TCP Flow (43498 , 80): 1.2 bytes/msec

Since there is some loss in packets, the through put has less.

Empirical Through Put for TCP Flow (43500, 80): 1.28 bytes/msec

Theoretical Through Put for TCP Flow (43500, 80): 1.21 bytes/msec

Since there is huge loss compared to other two flows, the through put is least of all the three flows.

Empirical Through Put for TCP Flow (43502, 80): 1.48 bytes/msec

Theoretical Through Put for TCP Flow (43502, 80): infinity bytes/msec

There is no loss of packets in this case, thus through put is highest in this case.