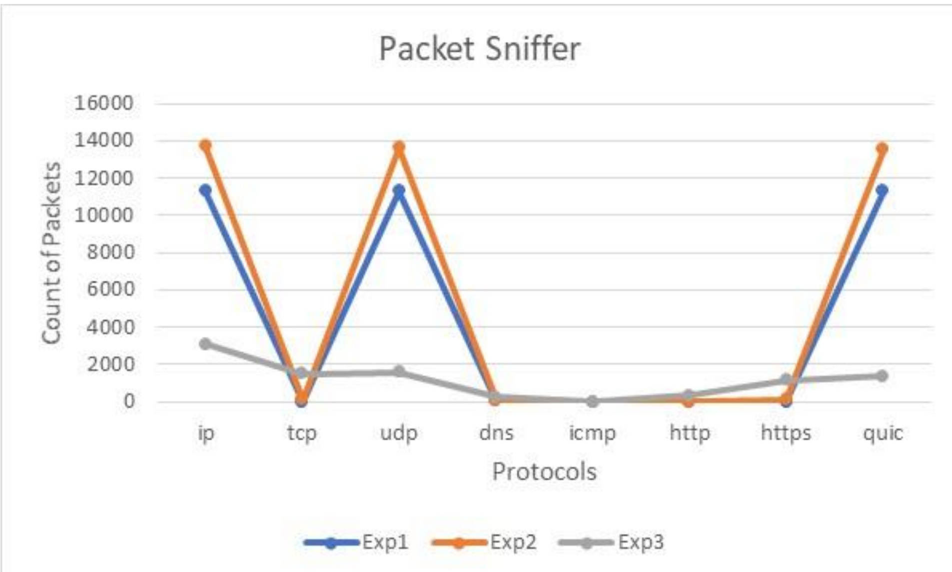
CSC/ECE 573: Internet Protocols
Mini Projects Report

Name: Shashank Shekhar **Unity ID: sshekha4**

Project 1: Packet Sniffer

Plot of packet counts:



Analysis:

The above plot represents three experiments conducted in a Packet Sniffer Application. Below are the experiments:

Experiment 1: First a youtube video is played at the highest resolution in chrome browser. Once video started playing, the packet sniffer is run. Packet Sniffer application exits after running for 30 seconds.

Experiment 2: First the packet sniffer application is run. Then Chrome browser is opened and a youtube video is played quickly. Packet Sniffer application exits after running for 30 seconds.

Experiment 3: First the packet sniffer application is run. Then random searches are made on google in Chrome browser and different other websites are opened. Packet Sniffer application exits after running for 30 seconds.

Note: Chrome browser is used for all the above experiments with QUIC Protocol enabled.

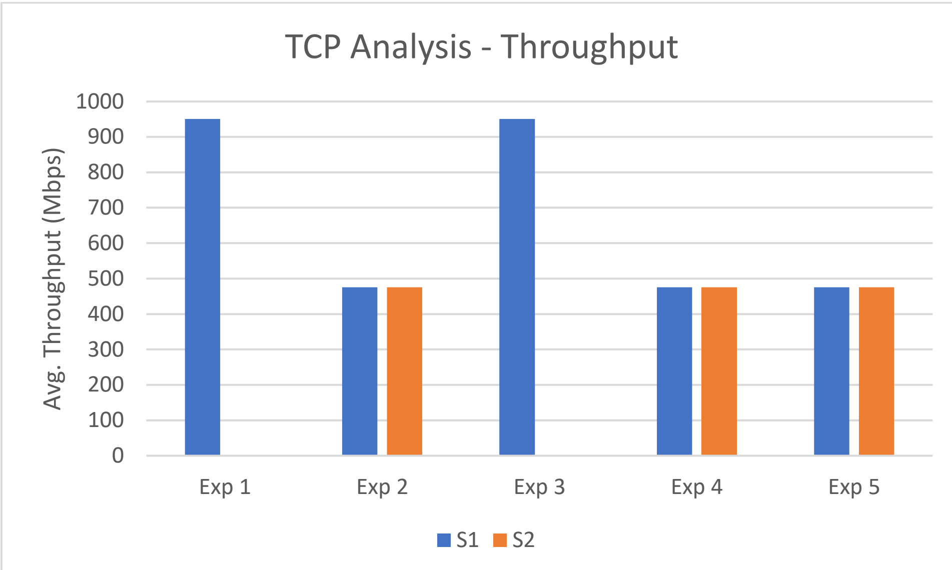
- ✓ **Number of Packets:** It is observed that the number of packets (in general) generated in experiment three is significantly less than the number of packets generated in the first two experiments. This is because in the first two experiments, videos are played which generates burst traffic (i.e. a large number of packets are generated in a short period of time) whereas in the third experiment, fewer packets are generated because websites are accessed which do not play any video and only return the small number of objects required to load their homepage. Hence, lesser packets are generated in the third experiment. Also, first experiment generates lesser number of packets compared to second experiment. This is due to the fact that in the first experiment, youtube video is played first and by the time the Packet Sniffer application is run, a lot of packets have already been put in the buffer for the video playback for the next few seconds. Hence, the number of packets generated in the second experiment (where the video is played after running the Packet Sniffer) is more that the number of packets generated in the first experiment.

✓ **Type of Packets:**

- **For experiments 1 and 2:** All the packets generated are of type QUIC which runs on UDP transport layer protocol. All these UDP packets in turn are IP Packets and hence the count of packets for IP, UDP and QUIC is high. Also, their count is equal indicating that all the IP packets were of type UDP and ran QUIC. Since, QUIC is enabled, there are no TCP and hence no HTTPS packets in these experiments. Also, there are no HTTP packets for these two experiments.
- **For experiment 3:** IP packets generated were divided between UDP and TCP packets. A few searches made on google webpage resulted in generation of QUIC packets and hence the UDP packets. Opening some other webpages resulted in generation of HTTPS packets and hence the TCP packets. Also, there are a very few HTTP requests over UDP in this experiment.
- **Common to all three experiments:** There are no ICMP packets because no explicit diagnostic utilities like ping or traceroute are run in these experiments. Also, there was no error reporting needed by ICMP since the end systems (hosting the video / webpages) were reachable. Hence, no ICMP Packets were generated. The count of DNS packets is almost zero because very few dns queries were made to get the IP address of the server hosting the video/webpages from the corresponding authoritative dns server.

Project 2: TCP analysis

Plot of average throughput:



Analysis:

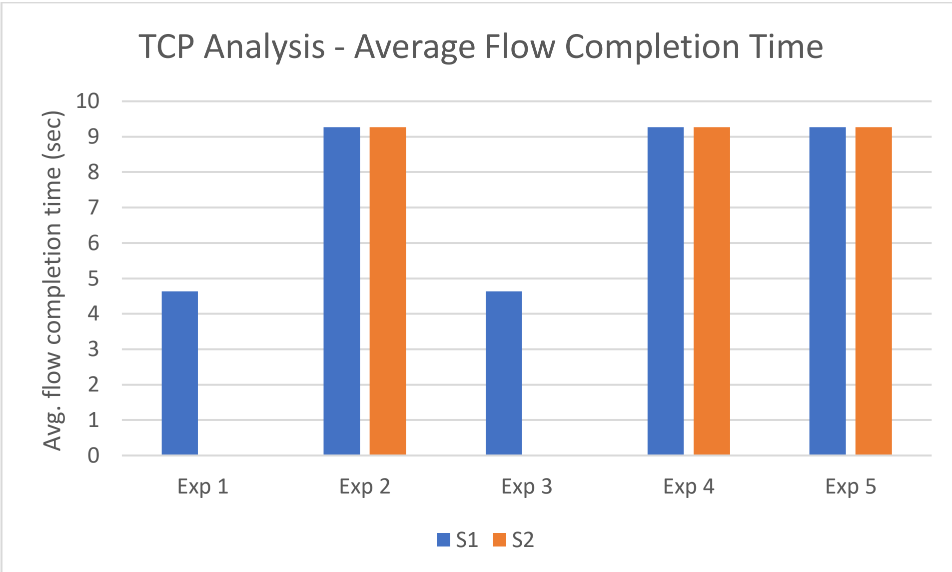
Note: For all the 5 experiments, links in the network have a transfer rate of 1 Gbps and 500 MB of data is transferred from the sender to the receiver. Delay in the links is set to zero. In all the experiments, senders start sending the data at 1 second. Simulation is run for 1 minute to allow for buffer time in case there is a delay in sending the data due to packet losses and hence retransmissions.

When there is only one sender in the network i.e. in experiments 1 and 3, throughput achieved is approximately 950 Mbps which is close to the ideal transfer rate of 1 Gbps. The difference between the expected and the achieved throughput is due to a small number of packet losses incurred during packet transfer.

When there are two senders in the network and one TCP variant is used for sending the data i.e. in experiments 2 and 4, then the throughput (bottleneck bandwidth) is almost equally divided between the two senders.

When there are two senders in the network and two TCP variants (TCP Cubic and DCTCP) are used for sending the data i.e. in experiment 5 where sender 1 uses TCP Cubic and sender 2 uses DCTCP, then it is observed that both these variants share the bottleneck bandwidth equally.

Plot of average flow completion time:



Analysis:

When there is only one sender in the network i.e. in experiments 1 and 3, average flow completion time is around 4.63 seconds. The difference between the expected average flow completion time of approximately 1 second and the achieved average flow completion time is due to a small number of packet losses incurred during packet transfer.

When there are two senders in the network and one TCP variant is used for sending the data i.e. in experiments 2 and 4, then the average flow completion time of 9.27 seconds is double compared to when there is only one sender. This is because the bandwidth in the bottleneck link is reduced to half of the total available bandwidth of 1 Gbps and hence the average flow completion gets doubled.

When there are two senders in the network and two TCP variants (TCP Cubic and DCTCP) are used for sending the data i.e. in experiment 5, then the average flow completion time doubles to approximately 9.27 seconds. This is because two senders are sending data simultaneously and the available bandwidth is shared.