CS 349 Networks Lab - Assignment 4

Network Simulator Assignment using NS-3

Arpit Gupta 170101012

Annanya Pratap Singh Chauhan 170101008

Shivang Dalal 170101086

I Question

The objective is to compare the effect of CBR traffic over UDP agent and FTP traffic over TCP agent. Consider a TCP agent from TCP HighSpeed, TCP Vegas and TCP Scalable for the FTP traffic. Consider a Dumbbell topology with two routers R1 and R2 connected by a wired link (30 Mbps, 100 ms), and use drop-tail queues with queue size set according to bandwidth-delay product of the link. Each of the routers is connected to 2 hosts, i.e. H1, H2 are connected to R1, and H3, H4 are connected to R2. The hosts are attached to the routers with (80 Mbps, 20ms) links. The CBR traffic over UDP agent and FTP traffic over TCP agent are attached to H1 and H2 respectively. Choose appropriate packet size for your experiments and perform the following:

1. Compare the delay (in sec) and throughput (in Kbps) of CBR and FTP traffic streams when only one of them is present in the network. Plot the graphs for the delay (in sec) and throughput (in Kbps) observed with different packet sizes.

2. Start both the flows at the same time and also at different times. Also, compare the delay (in sec) and throughput (in Kbps) of CBR and FTP traffic streams. Plot the graphs for the delay (in sec) and throughput (in Kbps) observed with different packet sizes.

II Introduction

As specified in the question we used a dumbell style setup , having two routers R1 and R2. Two Senders on R1 , namely S1 and S2 and two receivers on R2, D1 and D2. The assumptions we used are as follows , the data rate for CBR is 1Mbps , and we run each simulation for a total of 10 seconds with a given packet size. We have set these parameters to keep the throughputs of CBR and FTP comparable.

Each link in the setup has its own p2p link and p2p helper and its own ipv4 address space.

Packet size range has been taken from 512 to 4992 bytes in increments of 64 bytes.

The three tcp protocol used are :

1. **TCP Vegas**: It uses packet delay, rather than packet loss, as a signal to determine the rate at which to send packets. It detects congestion at an incipient stage based on increasing RTT values of the packets in the connection. Thus Vegas is aware of congestion in the network before packet losses occur.
2. **TCP Highspeed**: It is a modification of the TCPReno congestion control mechanism for use with TCP connections with large congestion windows. It is a loss-based algorithm, using additive-increase/multiplicative-decrease to control the TCP congestion window. It increases its aggressiveness (in particular, the rate of additive increase) as the time since the previous loss increases.
3. **TCP Scalable:** is a simple change to the traditional TCP congestion control algorithm which dramatically improves the performance in high speed wide area networks. Scalable TCP changes the algorithm to update TCP's congestion window to the following: cwnd:=cwnd+0.01 for each ACK received while not in loss recovery and cwnd:=0.875*cwnd on each loss event.

The routers have been configured to have drop tail queues based on the bandwidth delay products. It is specified to use No. of packets and not bytes, hence we have used the following formulae. #packets= Transmission rate * propagation delay(others aren't significant) / packet size.

III CBR and FTP delay (Individual- part 1)

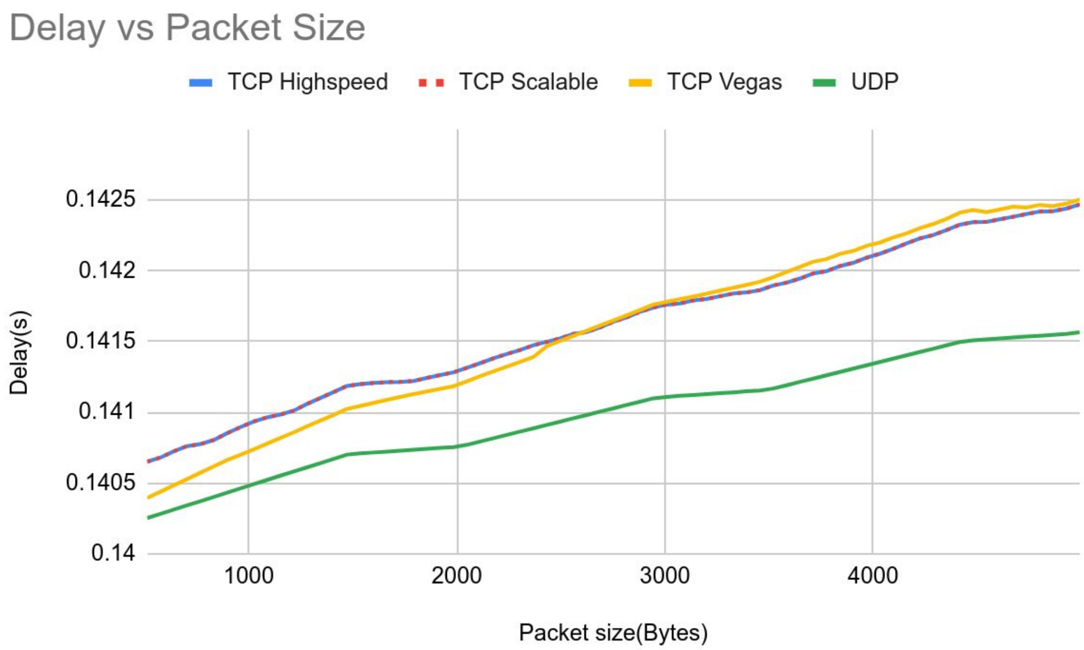
The below graph depicts the delay for the three TCP protocols and UDP based upon the packet size when the TCP and UDP stream are run independently (part 1). Note that TCP Highspeed and TCP Scalable have almost identical graphs.

The total delay is the sum of the following :

1. Transmission delay
2. Propagation delay
3. Queuing delay.

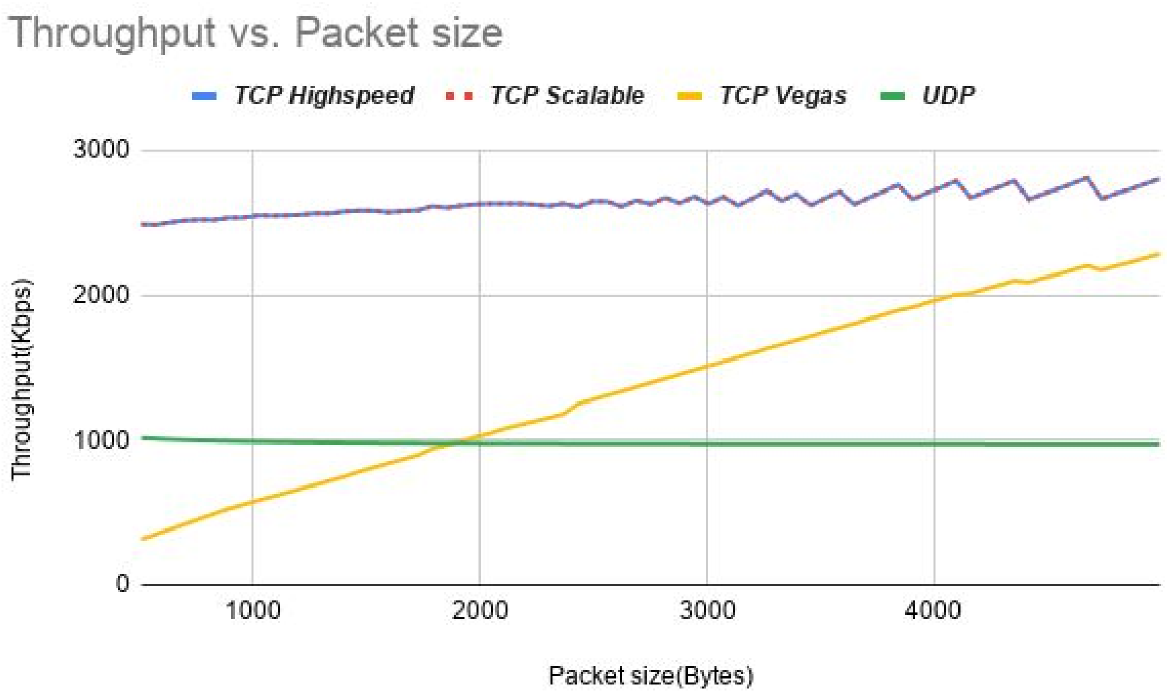
The propagation delay is fixed by the question S -> R1 (20ms) + R1->R2 (100ms) + R2->D

(20ms) = 140ms. Hence even with the smallest packet size the minimum delay is 140ms. The other two delays are variable and especially the transmission is directly proportional to packet size, hence we can see a steady increase in delay. However the transmission rate is upwards of 30Mbps and the packet size is less than 5 KB, hence the delay is negligible compared to the propagation delay.



IV CBR and FTP Throughput (Individual - part 1)

The below graph depicts the Throughput for the three TCP protocols and UDP based upon the packet size when the TCP and UDP stream are run independently (part 1). Note that TCP Highspeed and TCP Scalable have almost identical graphs. For TCP we have only considered the data sending streams throughput and not the ACK throughput. As we can see the UDP stream's throughput does not increase with packet size as we have provided a fixed data rate of 1Mbps. Hence it automatically decreases the number of packets needed to maintain the data rate. TCPs throughput depends on the protocol used, for Vegas it steadily increases and the other two although vary has an overall increase.



V Observations for part 2

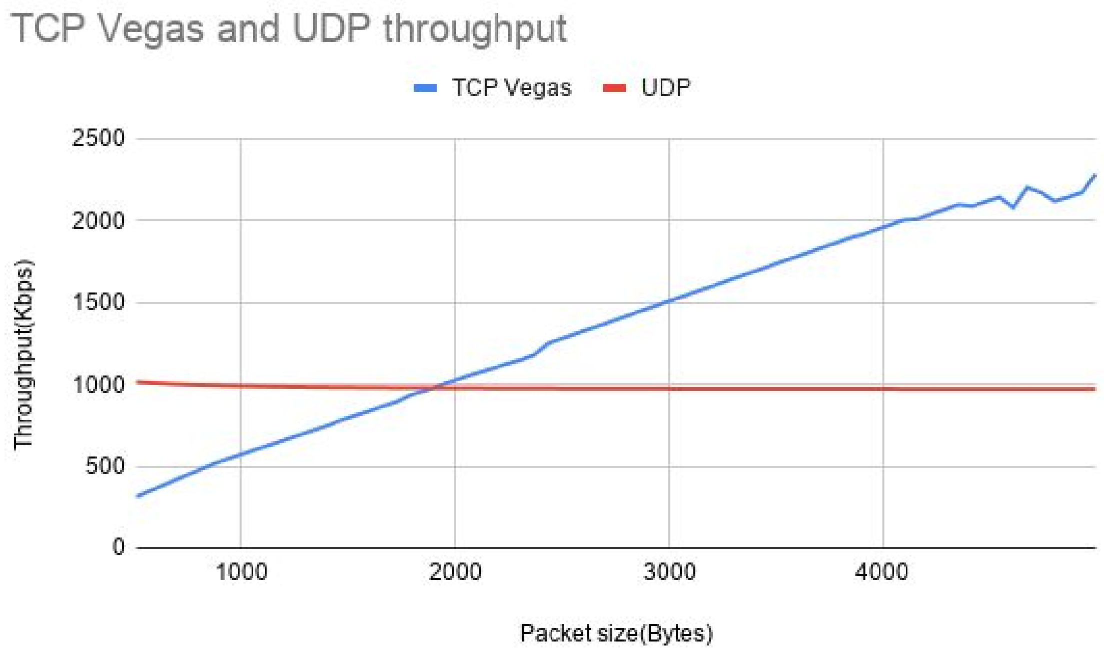
On running both the flows together there isn't any significant difference as compared to when they are run individually. We attribute this behavior to the comparatively large queue size based on the bandwidth delay product and the large transmission speed compared to the TCP and UDP throughput. The reasoning for the graphs are similar to the explanations for part 1.

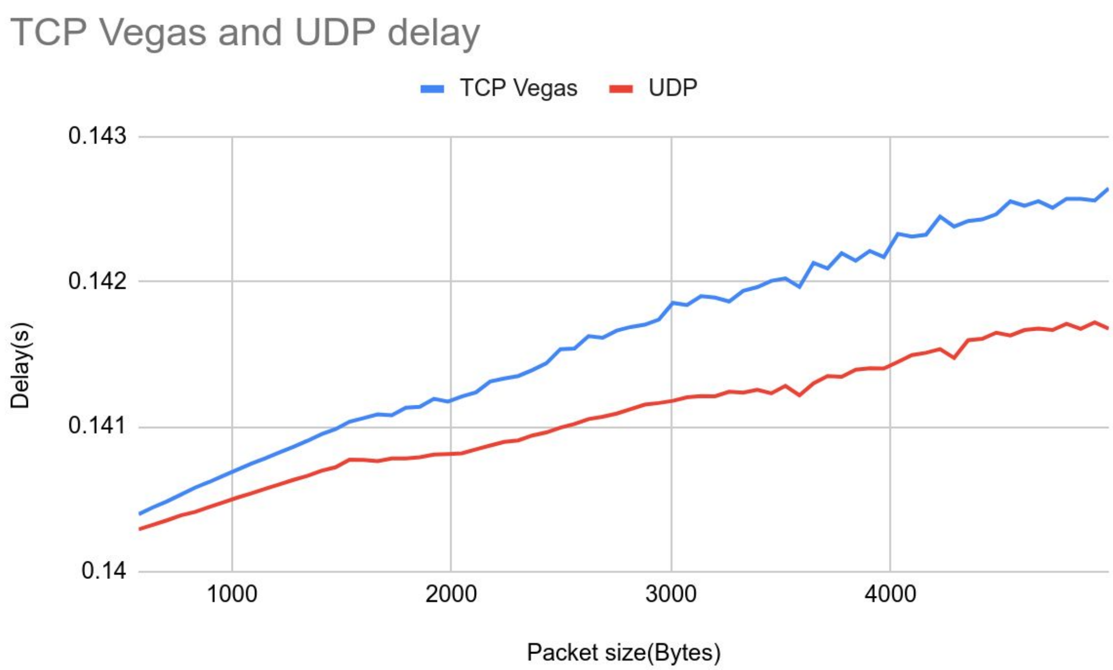
Only when the transmission speed and the throughputs are comparable does running the two flows simultaneously affect each other. Even then UDP retains its rate and the FTPs throughput plummets as it depends on ACKs which it does not receive as all the bandwidth is used by the UDP connection since UDP constantly sends data.

Below we have listed the delay and throughput comparison of UDP and TCP based on packet size corresponding to different TCP protocols.

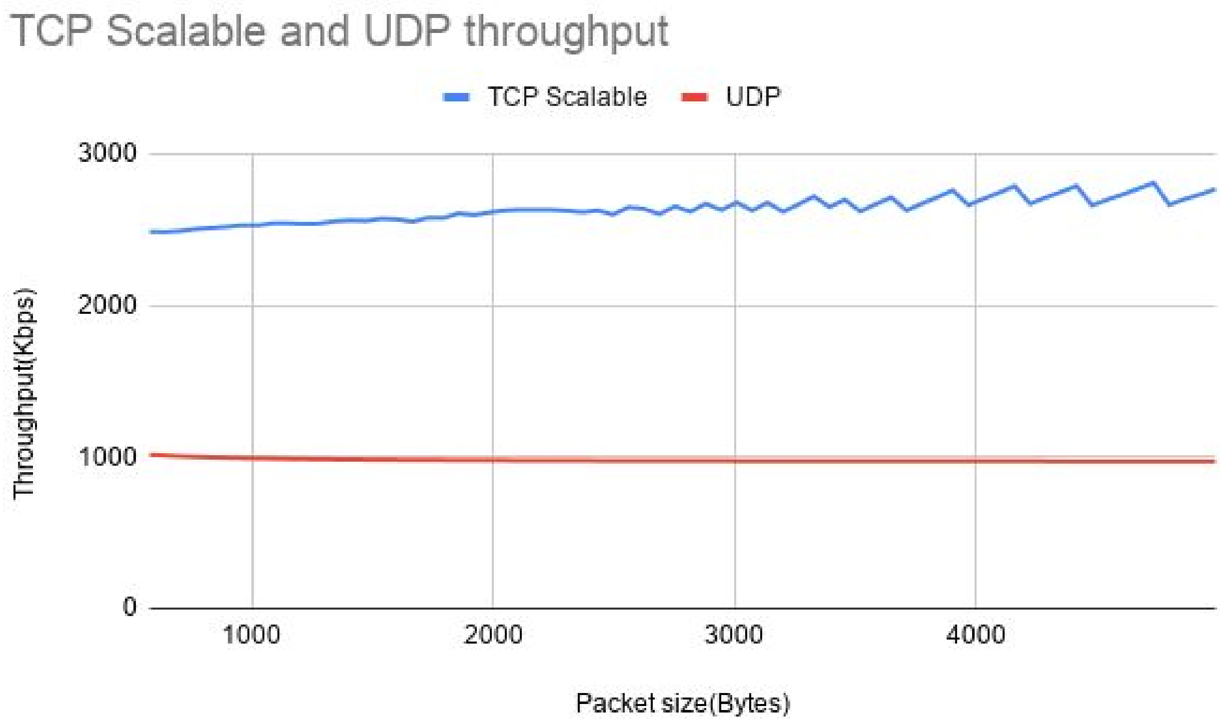
Also, we have not listed the graphs for starting the flows at different times, we did however run them. The graphs came out to be identical. Both the streams don't interact with each other. (No difference up to the 5th decimal of throughput).

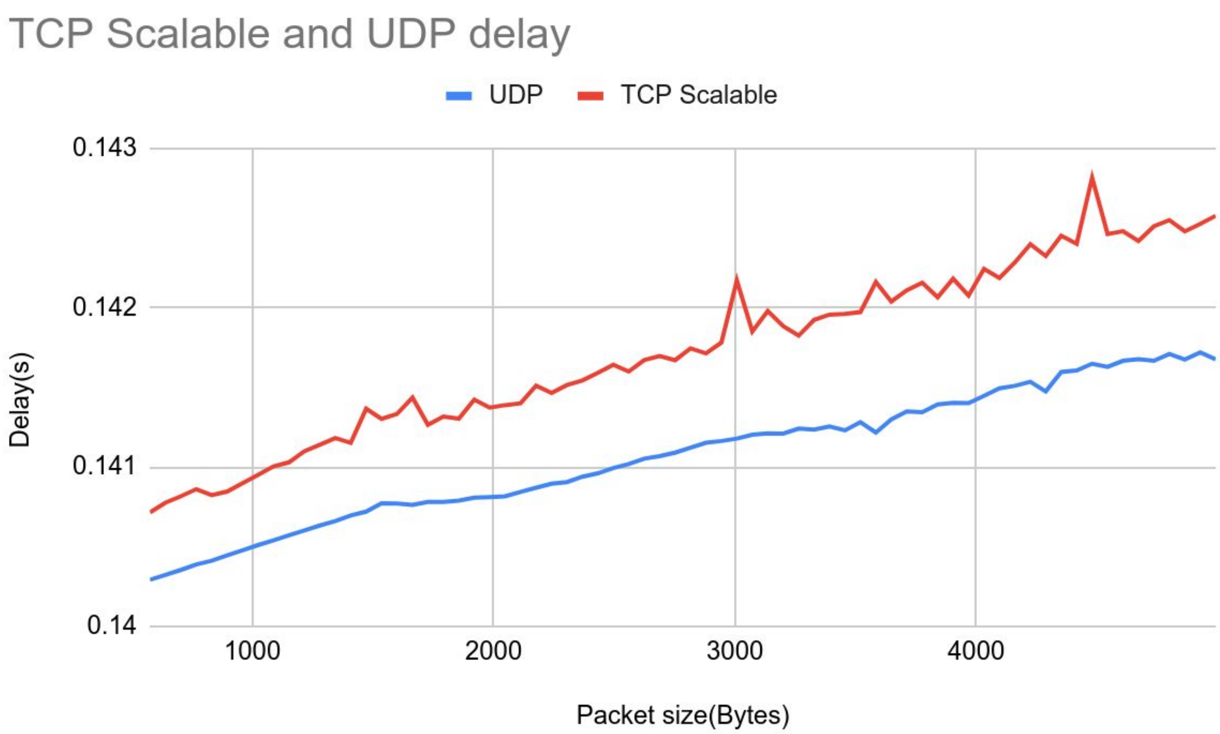
VI CBR vs TCP Vegas (together - part 2)





VII CBR vs TCP Scalable (together - part 2)





VIII CBR vs TCP Highspeed (together - part 2)