CS342 Networks Lab
Assignment - 4

Group 36:
Anuraag Mahajan - 190101110
Eshan Trehan - 190123070
Kshitij Singhal - 190123032
Hardik Suhag - 190101038

Application 6

Submission File Structure:

=> main.cc and main.h - source code
=> Task1 and Task2 folders contain the respective plots for every case segregated into .plt and .pdf formats.

Objective:

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.

Configuration:

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.

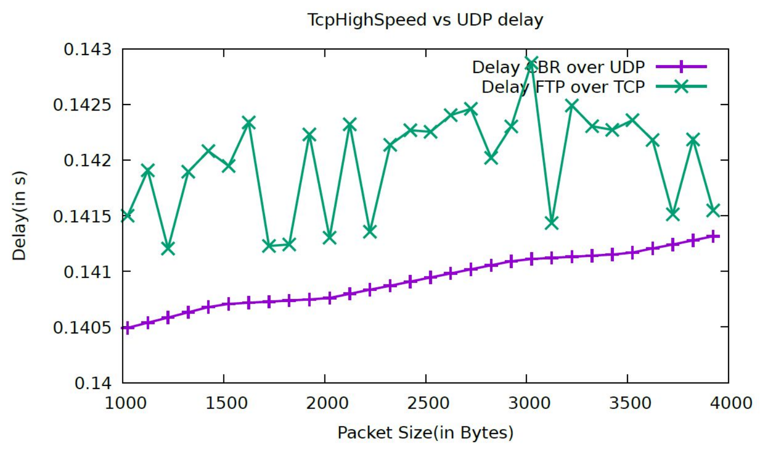
Common Parameters used to perform all experiments:

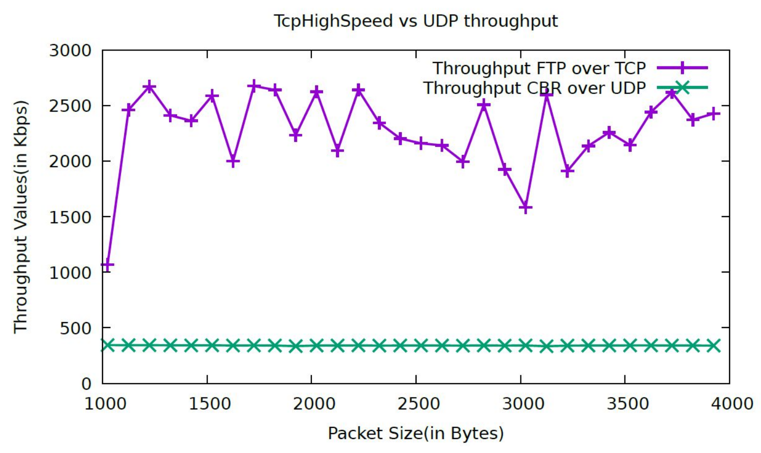
Input packet size - 1024 bytes
Loops - 30
Runtime - 1s

TASK 1:

Compare the delay (in ms) and throughput (in Kbps) of CBR and FTP traffic streams when only one of them is present in the network.

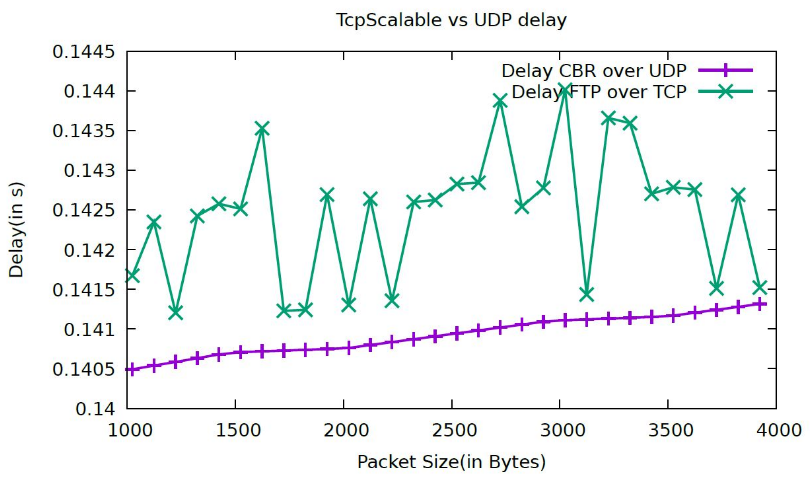
1) TCP HighSpeed

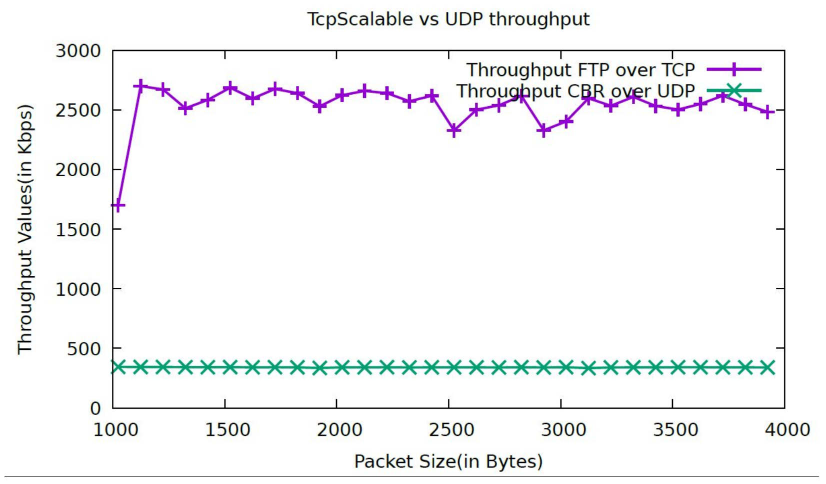




Observation:
FTP over TCP has higher and fluctuating delay than CBR over UDP and higher throughput.

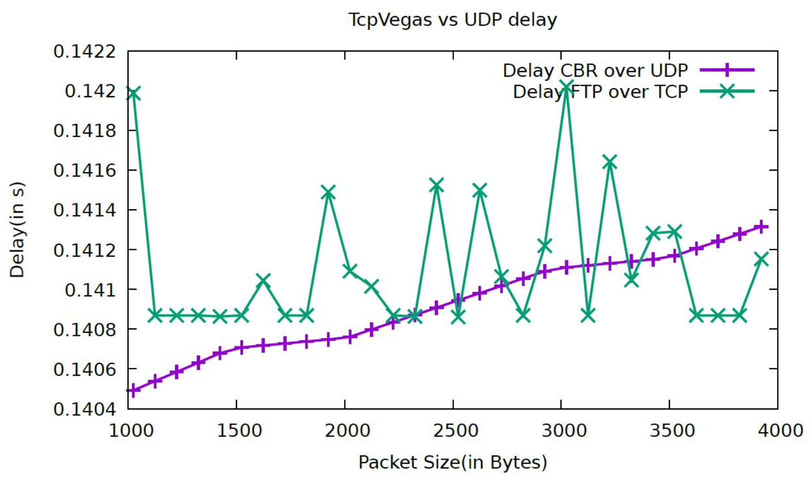
2) TCP Scalable

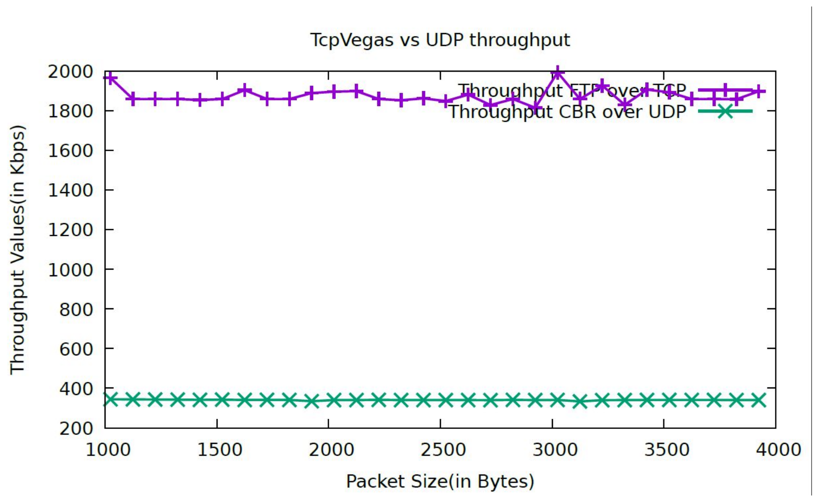




Observation:
We observe that the FTP over TCP has higher delay than CBR over UDP and higher throughput.

3) TCP Vegas





Observation:
In this case, we observe that the FTP over TCP has a delay comparable to CBR over UDP and even drops lower at times, unlike the previous two cases.
Throughput for FTP remains higher.

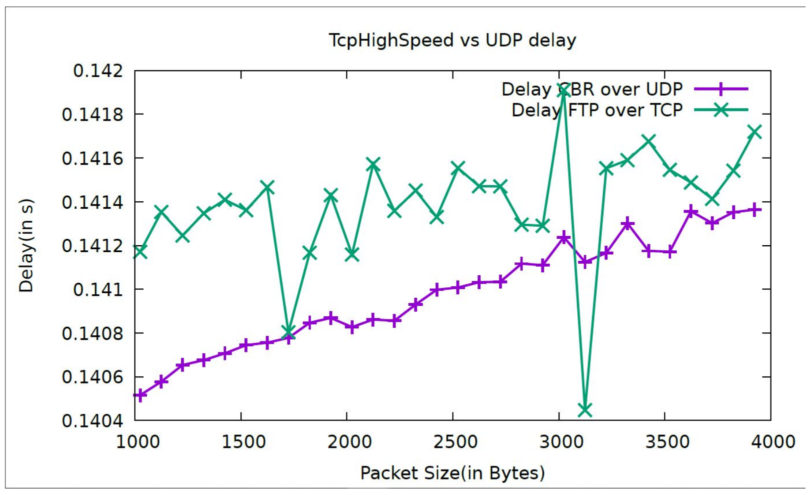
Overall, both Throughput and delay FTP over TCP are fluctuating as as compared to the stable outputs of CBR over UDP.

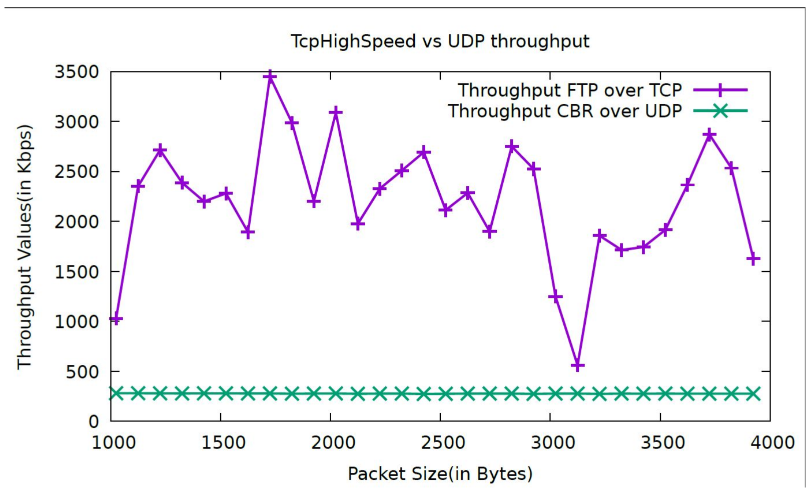
TASK 2

Compare the delay (in ms) and throughput (in Kbps) of CBR and FTP traffic streams by starting both the flows at the same time and also at different times.

1) TCP HighSpeed

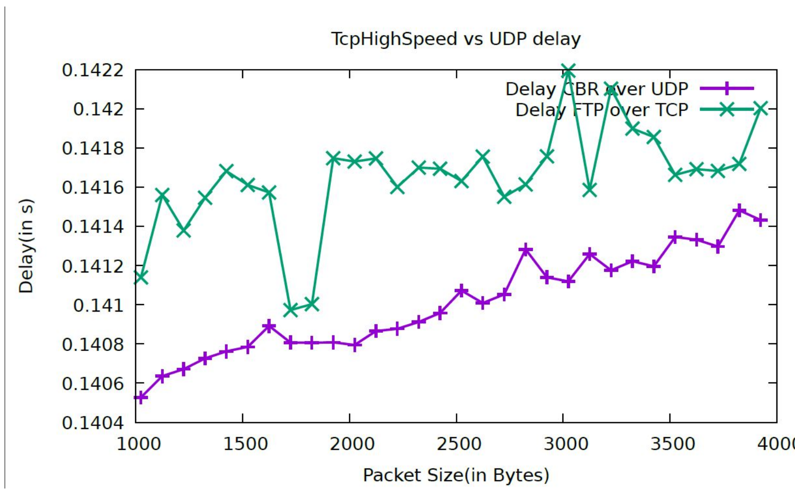
Same time

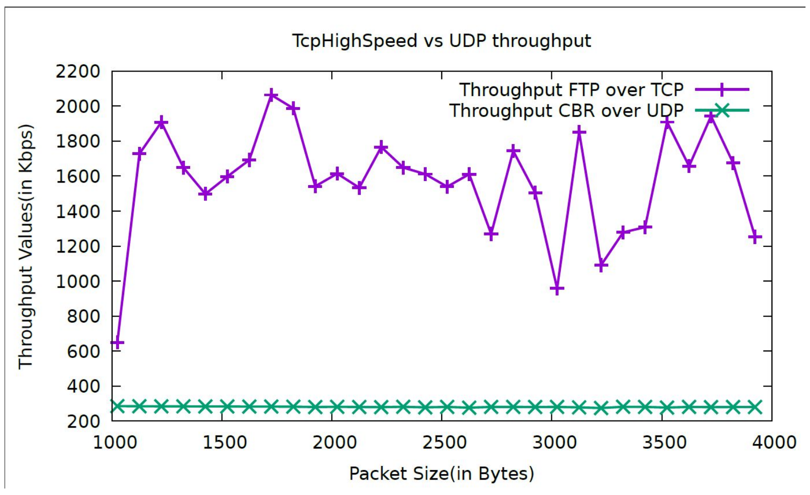




Observation:
We observe that the FTP over TCP has higher delay and throughput than CBR over UDP. Both are very erratic for FTP, as compared to TASK 1 observations.

Different time

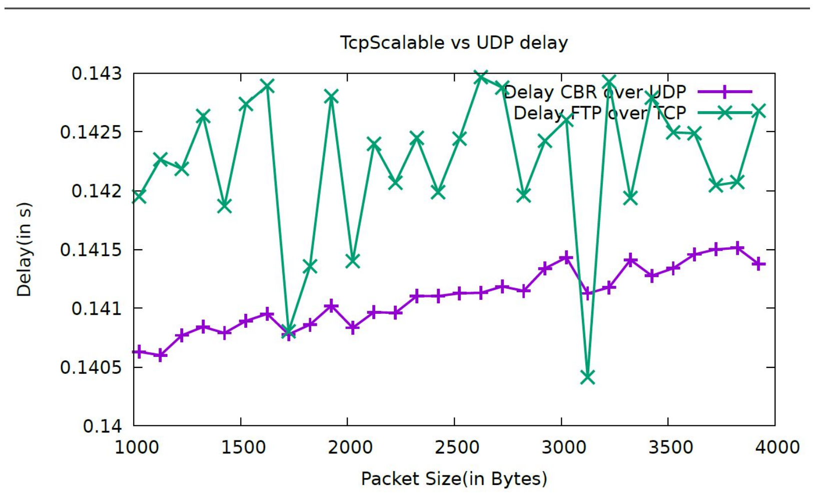


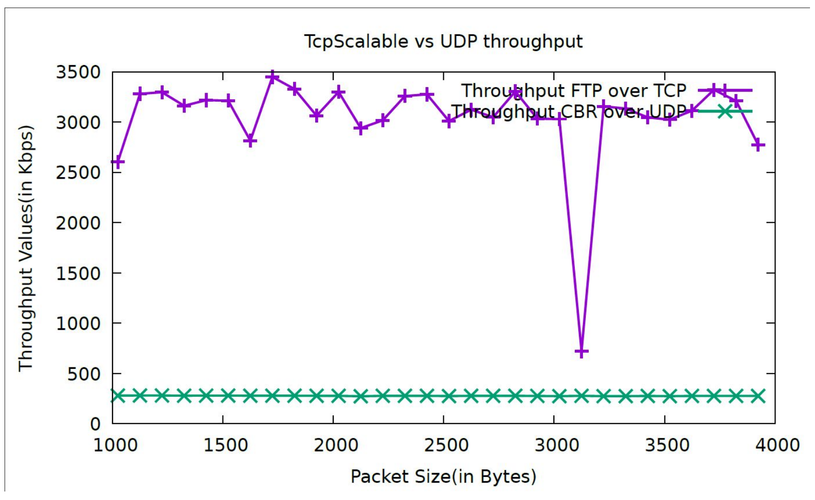


Observation:
Again, higher and fluctuating throughput and delay for FTP than UDP.
Though, the throughput for both UDP and TCP is a bit lower as compared to same time start..

2) TCP Scalable

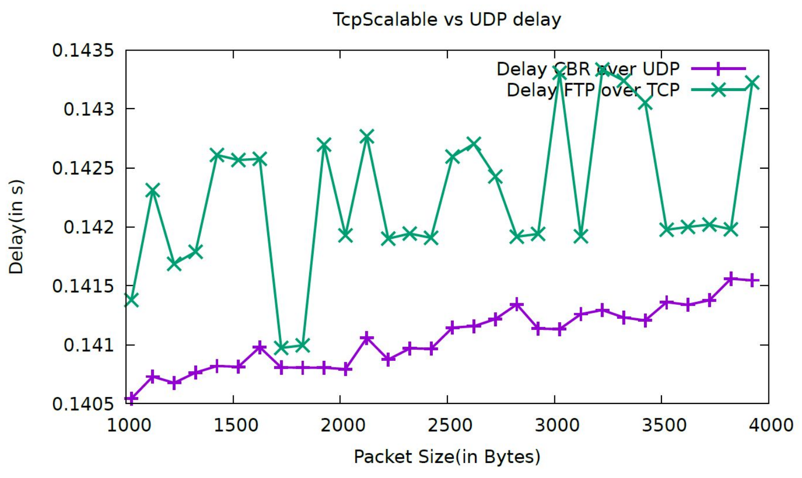
Same time

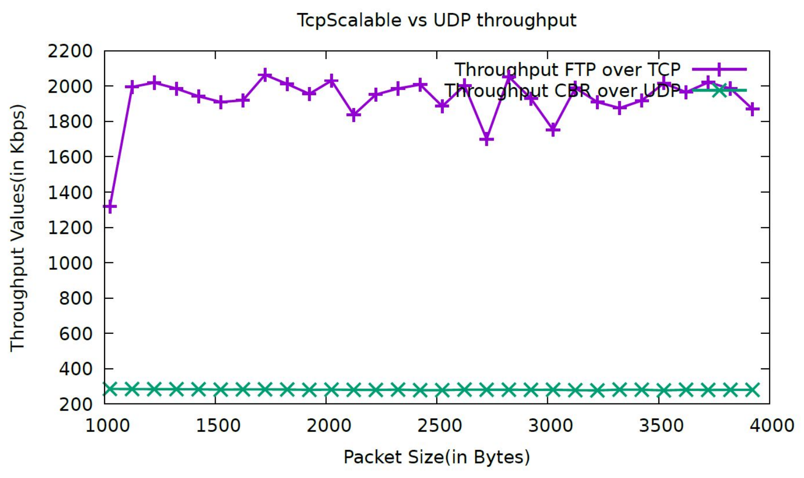




Observation:
FTP over TCP has higher but erratic delay than CBR
over UDP and higher throughput. The average throughput for TCP Scalable here seems to be higher than TCP HighSpeed.

Different time

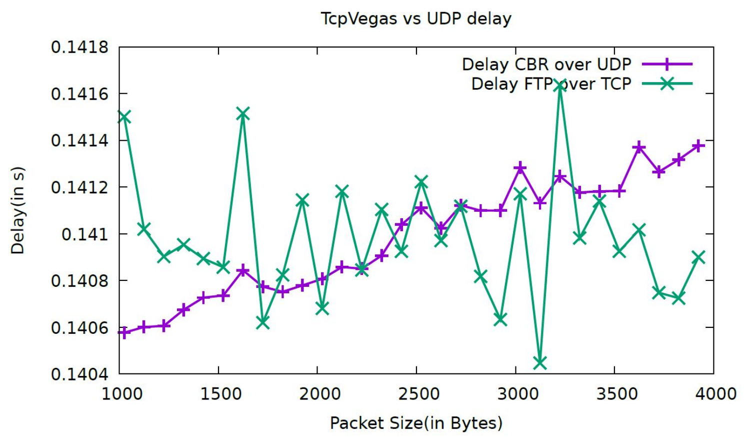


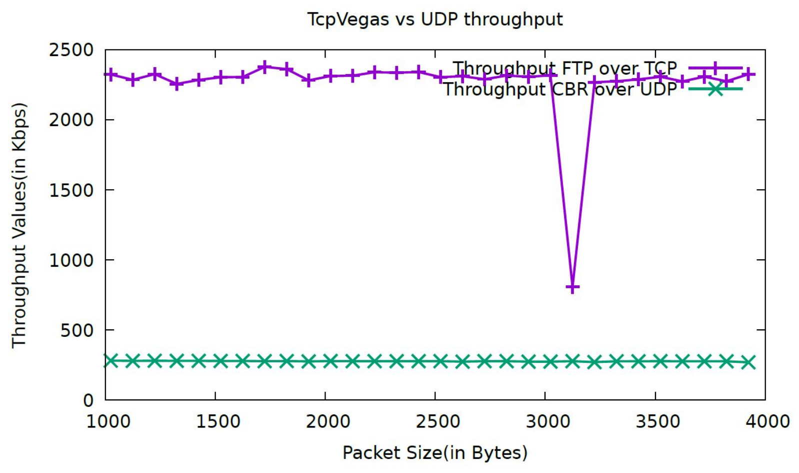


Observation:
FTP over TCP has higher and fluctuating delay than CBR over UDP and higher throughput.

3) TCP Vegas

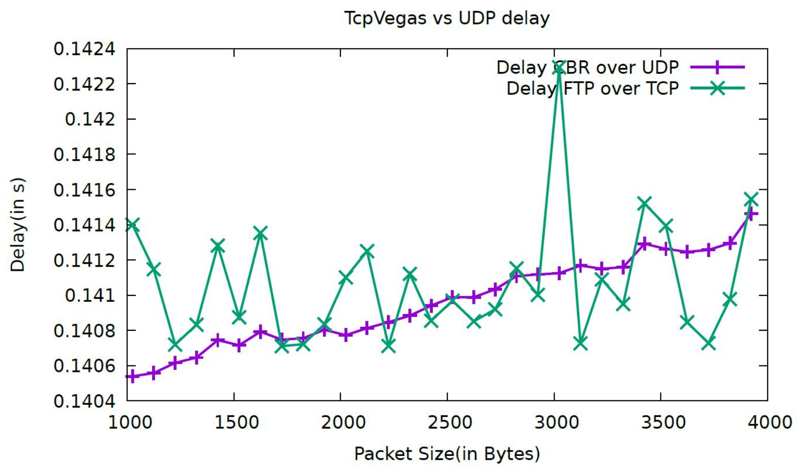
Same time

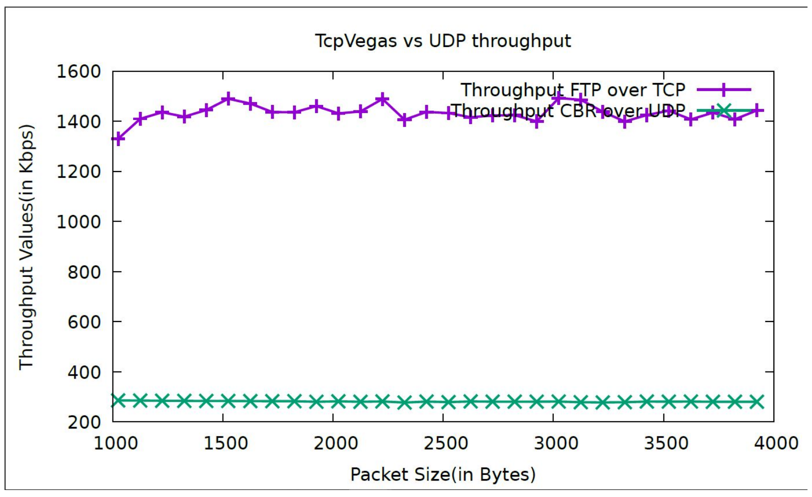




Observation:
As observed during TASK 1, the delay for TCP vegas is lower as compared to UDP and the TCP HighSpeed, Scalable.
Throughput for TCP follows the general trend here, of being higher than UDP.

Different time





Observation:
Delay for TCP is lower than UDP.
From all the plots we can also see that throughput for TCP Vegas is more stable when compared to TCP HighSpeed and TCP Scalable.

Reasoning the general trends and observations:

TCP-Vegas emphasizes packet delay, rather than packet loss, as a signal to determine the rate at which to send packets. It depends on the accurate calculation of the Base RTT value.
This explains why the packet **delay for Vegas is lower than High Speed and Scalable.**

Scalable TCP is a simple change to the traditional TCP congestion control algorithm which dramatically **improves TCP performance in high speed wide area networks**, this supports the observations that **throughput for Scalable is higher than its counterparts**.

Instructions for execution:

After successful installation of ns3, place main.cc and and main.h files inside the scratch folder in ns3 installation - *../ns-allinone-3.35/ns-3.35/scratch*

Run the following command for generating the plots -
./waf --run "scratch/main --tcpProtocol=<Protocol Name> --loopRuns=<count> --simultaneously=<0 or 1> --offset=<time in sec> --runTime=<time in sec> --packetSize=<size in bytes>

The input parameters can be given through this command to tailor every case mentioned in Task 1 and 2.

Example commands:
For TASK 1:
```
./waf --run "scratch/main --tcpProtocol=TcpHighSpeed --loopRuns=10 --simultaneously=0 --offset=0
--runTime=1 --packetSize=1024"
```

For TASK 2:
Same Time:
```
./waf --run "scratch/main --tcpProtocol=TcpHighSpeed --loopRuns=10 --simultaneously=1 --offset=0
--runTime=1 --packetSize=1024"
```

Different Time: (put non-zero offset)
```
./waf --run "scratch/main --tcpProtocol=TcpHighSpeed --loopRuns=10 --simultaneously=1 --offset=3
--runTime=1 --packetSize=1024"
```

The .plt files generated from the commands above can be converted to .pfd using -
gnuplot <filename>.plot