CS 342 : Computer Networks Lab

LAB 04

GROUP NUMBER : 49

<u>GROUP MEMBERS</u>:

TANMAY JAIN(180123050)
AB SATYAPRAKASH (180123062)

Application 6

Application Details.

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 the 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:

Question 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. Plot the graphs for the delay (in ms) and throughput (in Kbps) observed with different packet sizes.

We have attached the graphs for this question. For each graph we have the condition values and a brief observation.

Case a.

Conditions:

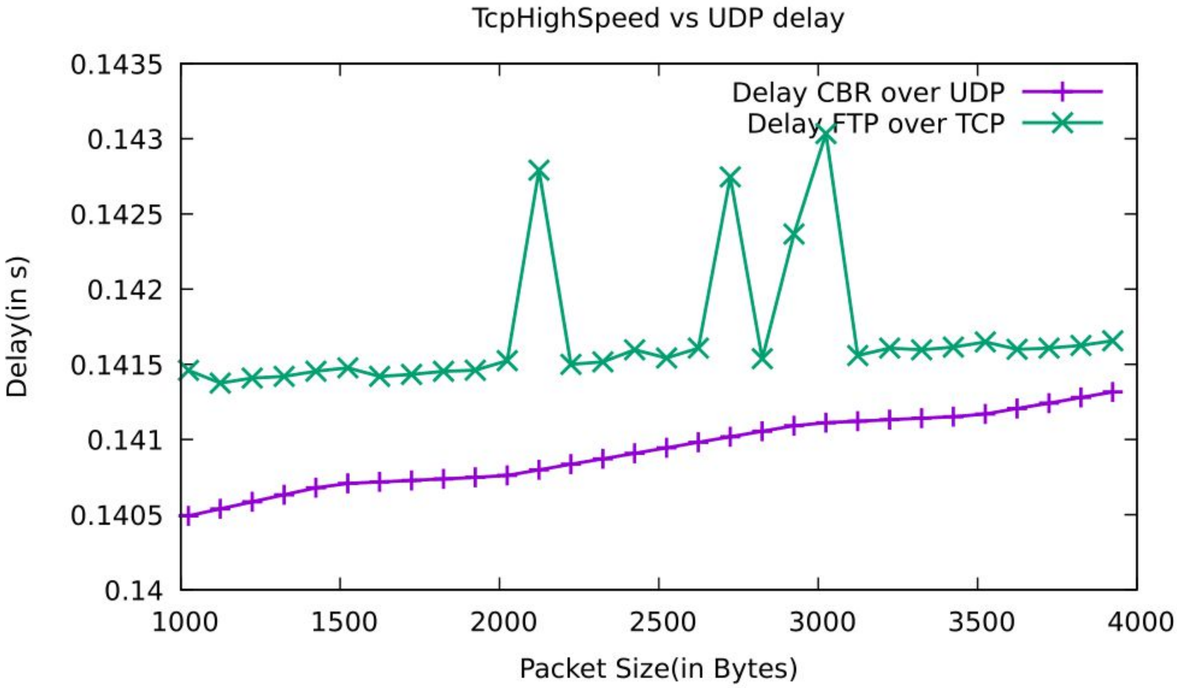
TcpHighSpeed, 5s runtime, 1024 bytes packet size.

Observation:

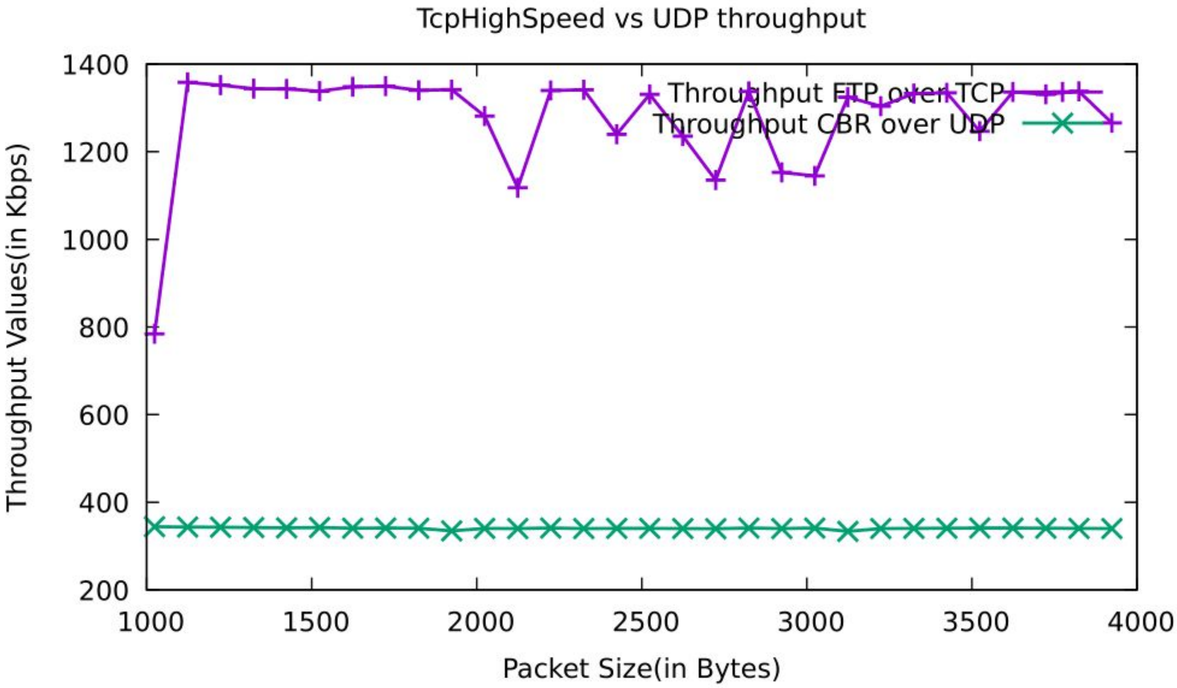
We observe that the FTP over TCP has **higher** delay than CBR over UDP and **higher** throughput.

Graphs: (for ms the graph just scales)

For delay:



For throughput:



Case b.

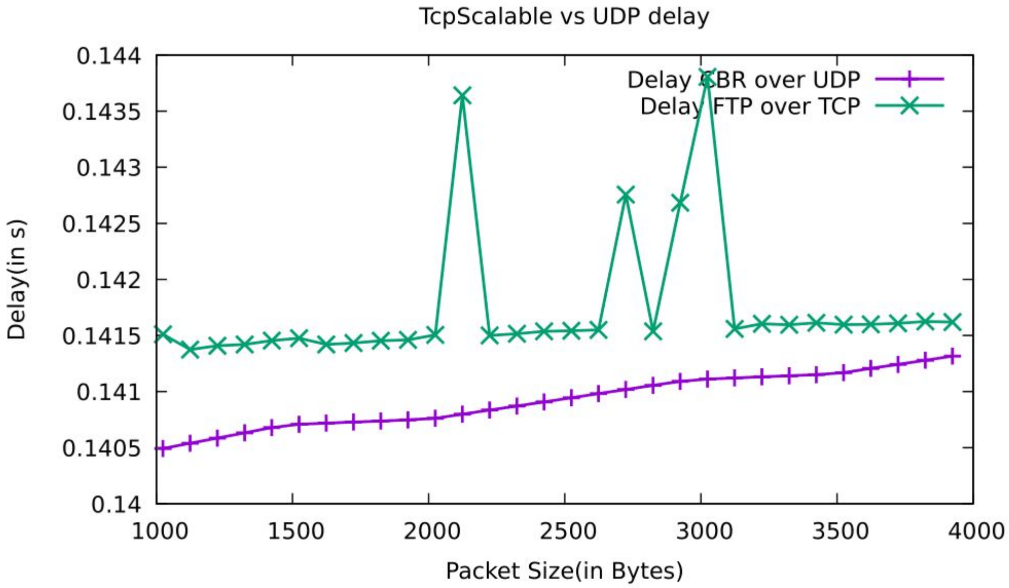
Conditions: **TcpScalable**, 5s runtime, 1024 bytes packet size.

Observation:

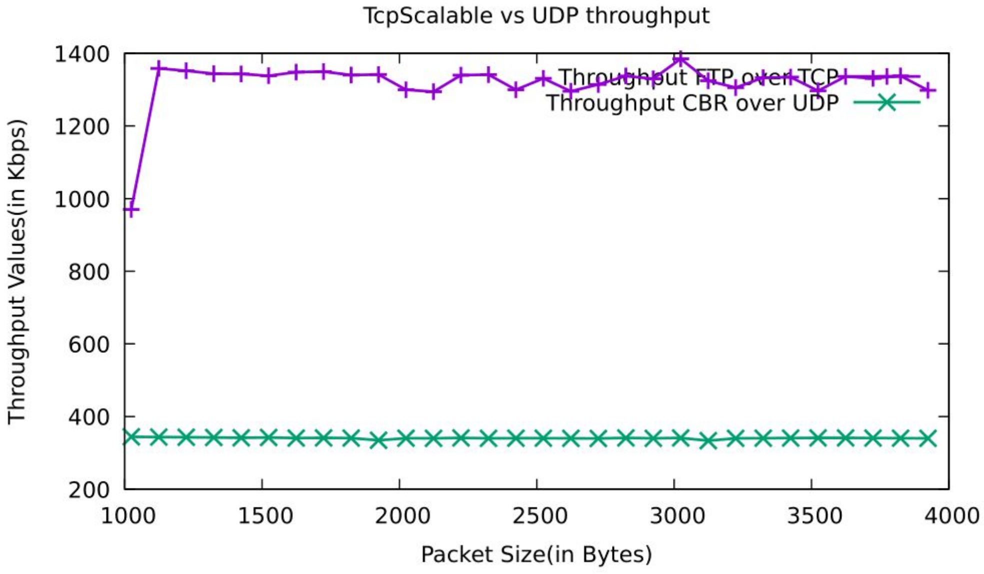
We observe that the FTP over TCP has **higher** delay than CBR over UDP and **higher** throughput.

Graphs: (for ms the graph just scales)

For delay:



For throughput:



Case c.

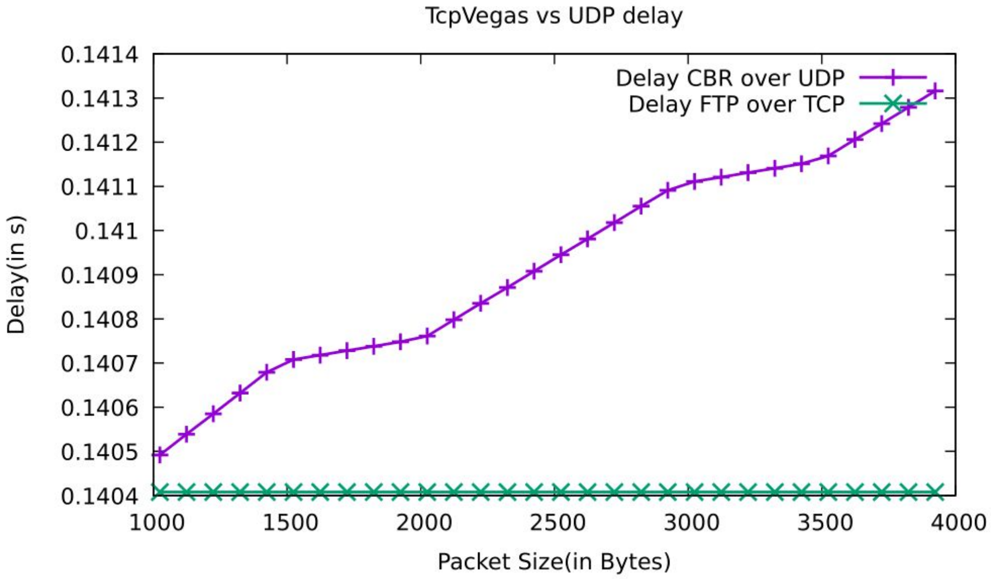
Conditions: **TcpVegas**, 5s runtime, 1024 bytes packet size.

Observation:

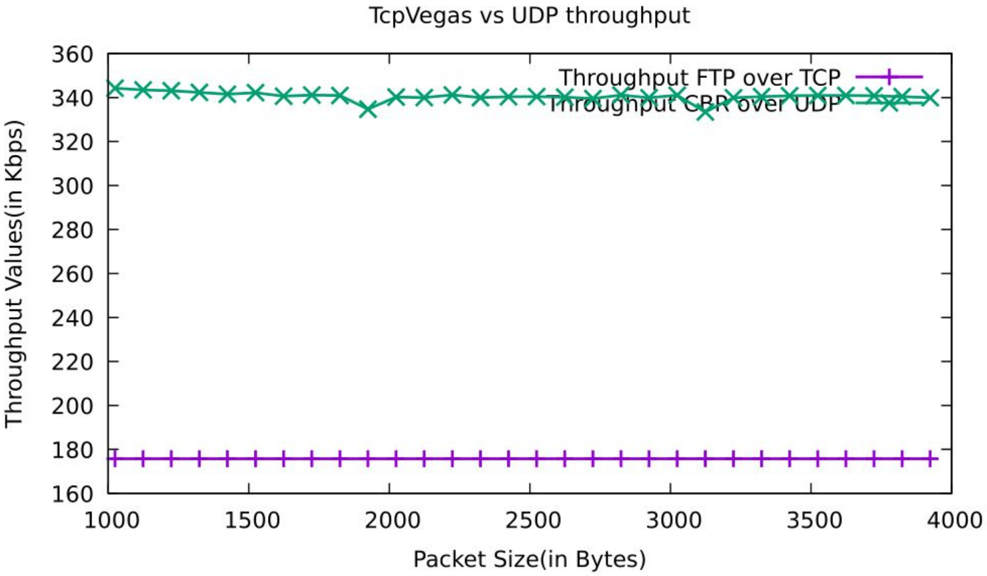
We observe that the FTP over TCP has much **lower** delay than CBR over UDP and much **lower** throughput.

Graphs: (for ms the graph just scales)

For delay:



For throughput:



Question 2.

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

We have attached the graphs for this question. For each graph we have the condition values and a brief observation.

Case a.

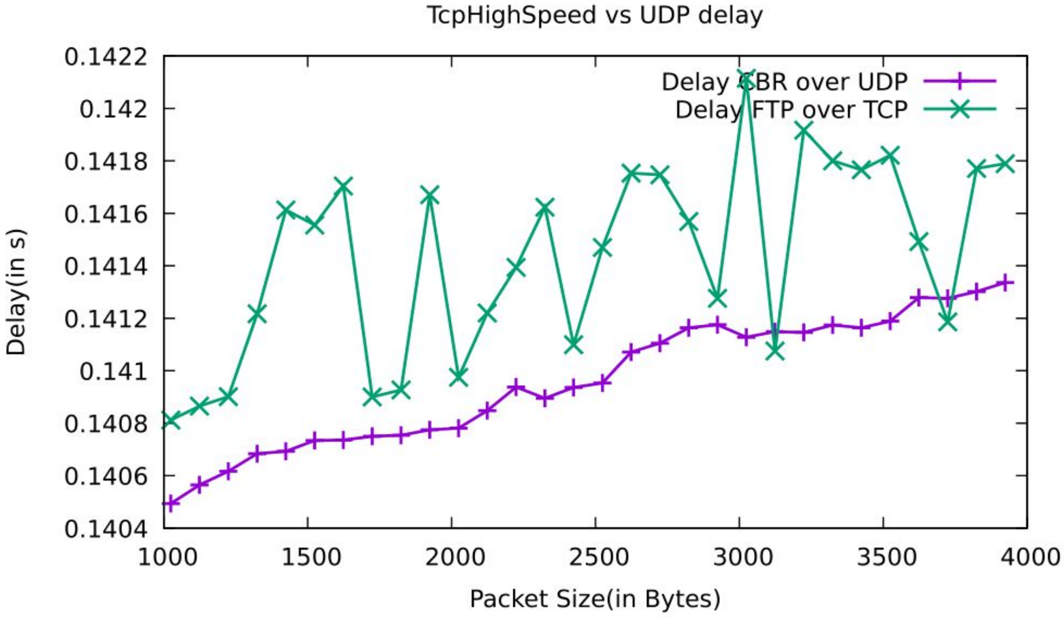
Conditions: **TcpHighSpeed**, 5s runtime, 1024 bytes packet size, **DIFFERENT START TIMES.**

Observation:

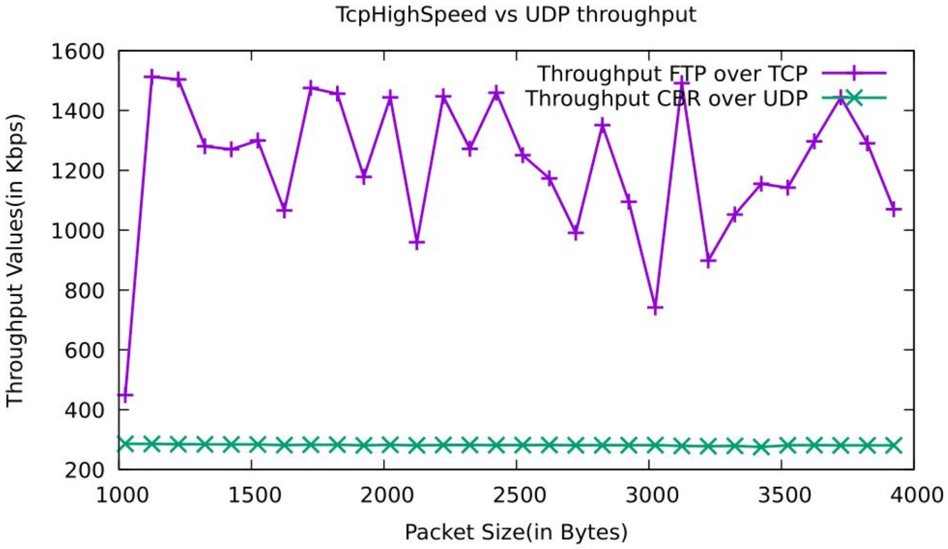
We observe that the FTP over TCP has **higher** but erratic delay than CBR over UDP and **higher** throughput.

Graphs: (for ms the graph just scales)

For delay:



For throughput:



Case b.

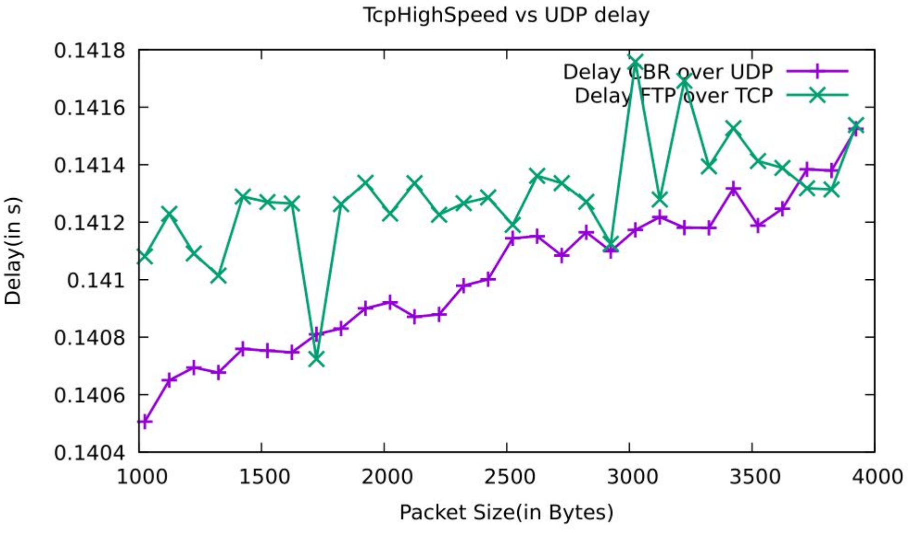
Conditions: **TcpHighSpeed**, 5s runtime, 1024 bytes packet size, **SAME START TIMES.**

Observation:

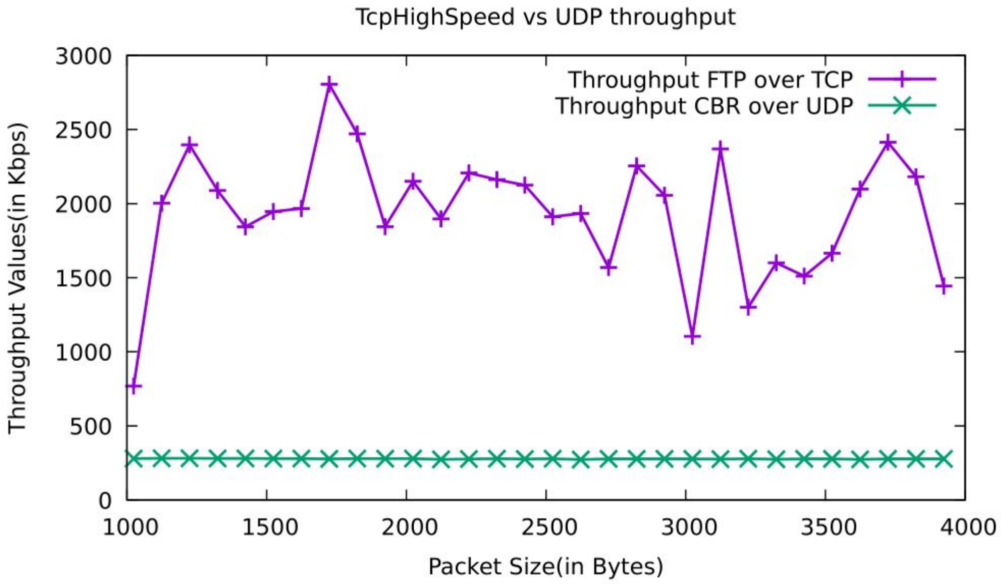
We observe that the FTP over TCP has **higher** but erratic delay than CBR over UDP and **higher** throughput.

Graphs: (for ms the graph just scales)

For delay:



For throughput:



Case c.

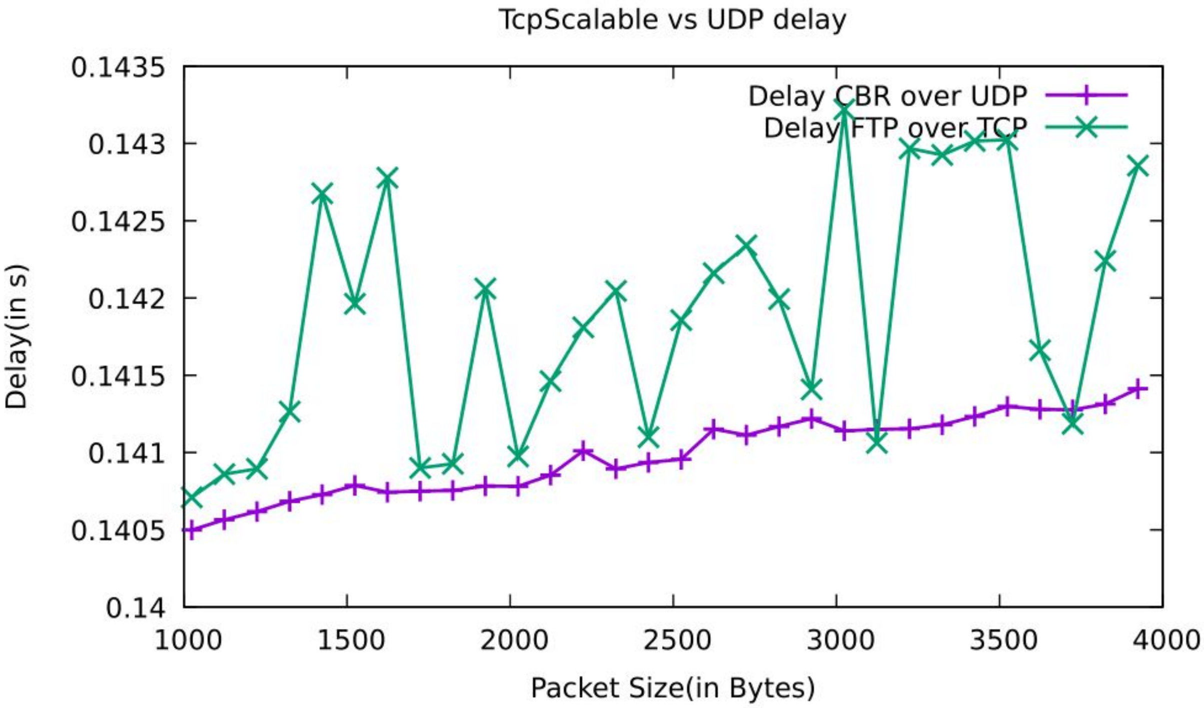
Conditions: **TcpScalable**, 5s runtime, 1024 bytes packet size, **DIFFERENT START TIMES.**

Observation:

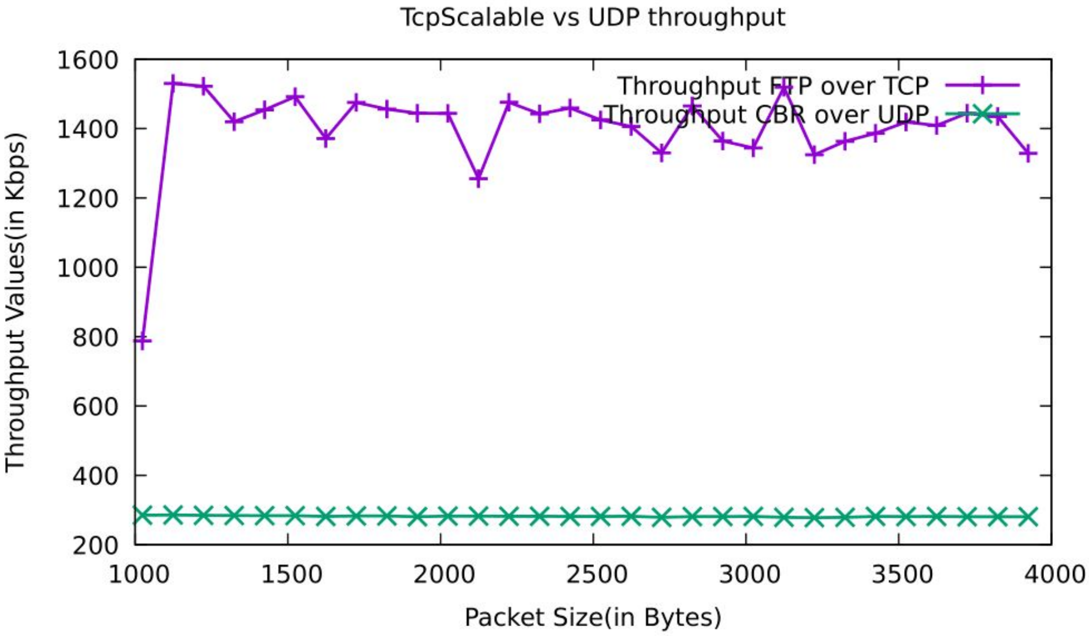
We observe that the FTP over TCP has **higher** but erratic delay than CBR over UDP and **higher** throughput.

Graphs: (for ms the graph just scales).

For delay:



For throughput:



Case d.

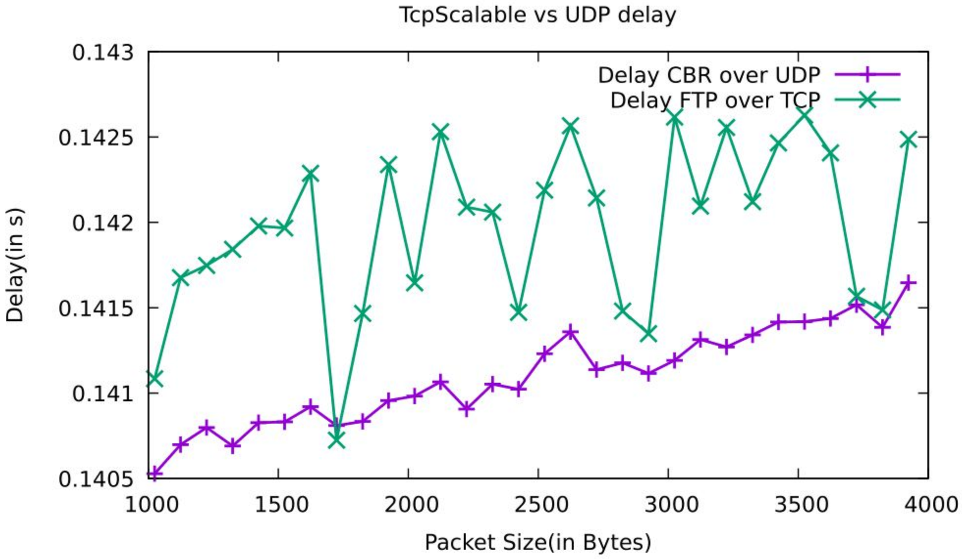
Conditions: **TcpScalable**, 5s runtime, 1024 bytes packet size, **SAME START TIMES.**

Observation:

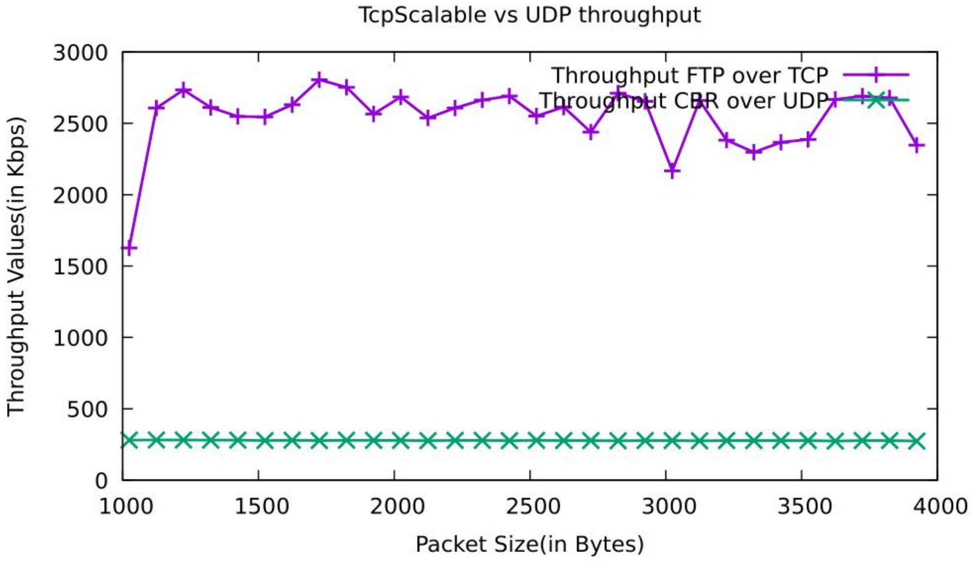
We observe that the FTP over TCP has **higher** but erratic delay than CBR over UDP and **higher** throughput.

Graphs: (for ms the graph just scales)

For delay:



For throughput:



Case e.

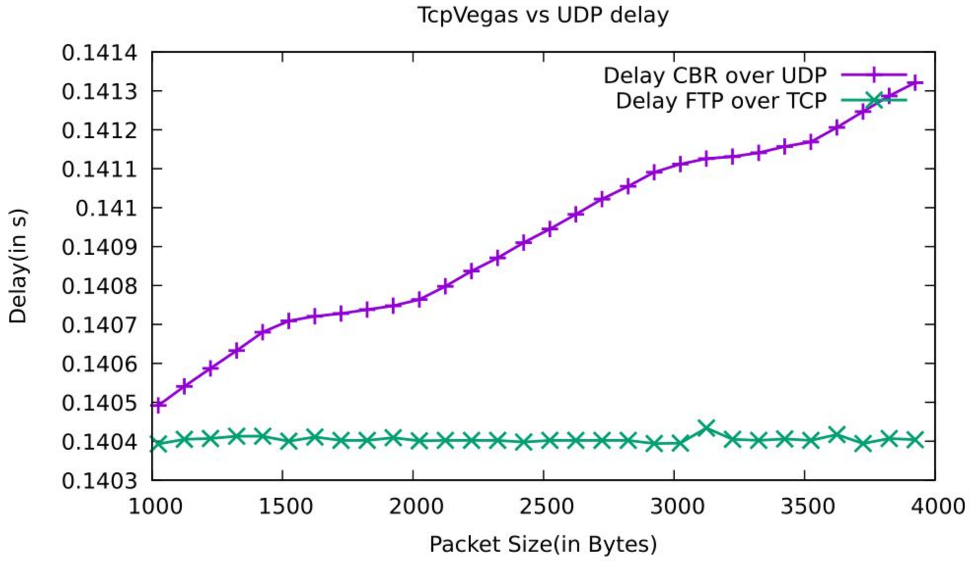
Conditions: **TcpVegas**, 5s runtime, 1024 bytes packet size, **DIFFERENT START TIMES.**

Observation:

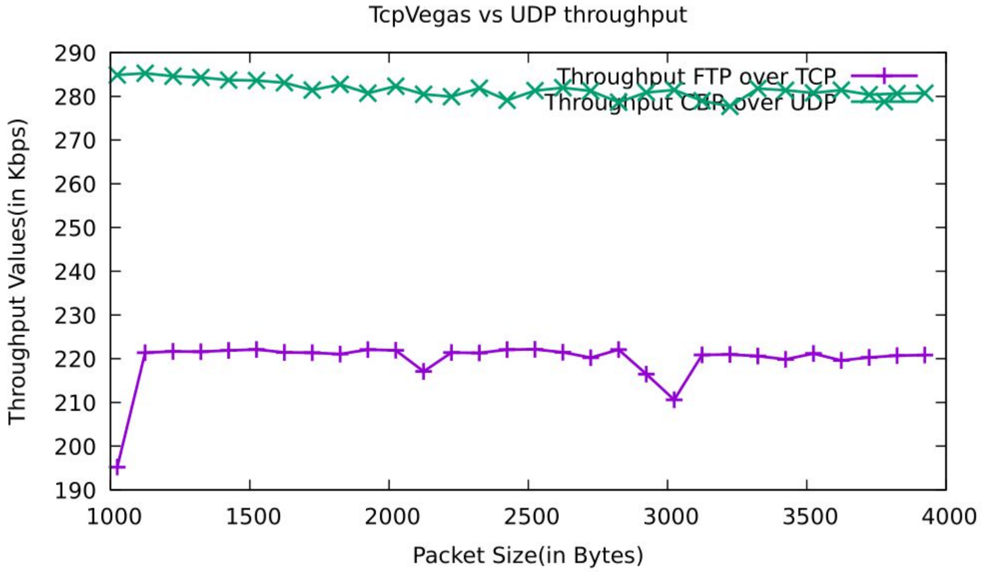
We observe that the FTP over TCP has **lower** delay than CBR over UDP and **lower** throughput.

Graphs: (for ms the graph just scales)

For delay:



For throughput:



Case f.

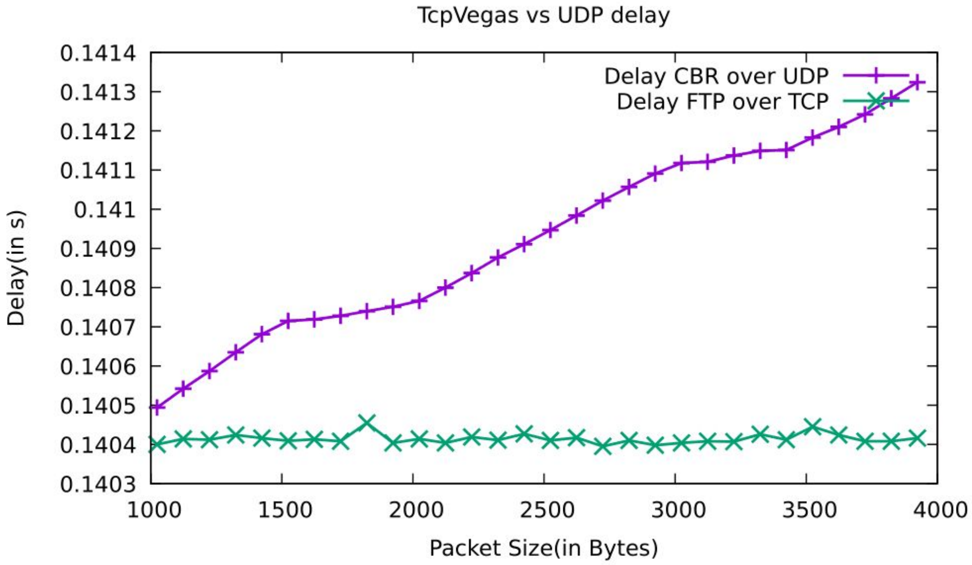
Conditions: **TcpVegas**, 5s runtime, 1024 bytes packet size, **SAME START TIMES.**

Observation:

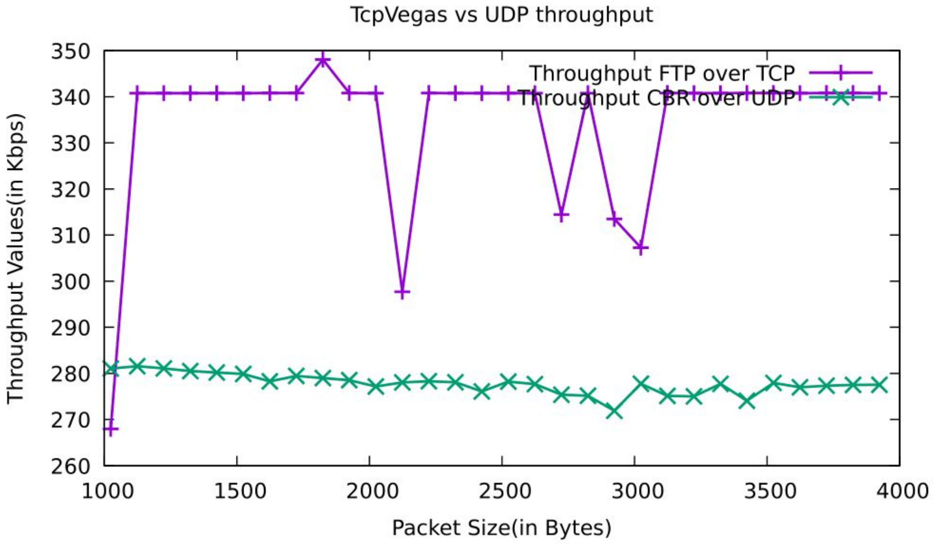
We observe that the FTP over TCP has **lower** delay than CBR over UDP and **higher** throughput.

Graphs: (for ms the graph just scales)

For delay:



For throughput:



Additional Info: ***The readme.txt file has all the instructions for running. Here is a brief summary.***

This is the directory structure:

```
Group_49
     - readme.txt
     - pltFiles/
    - UdpandTcpHighSpeed_delay_Separate.plt
    - UdpandTcpHighSpeed_delay_Simulataneous_Different_Start.plt
     ........ 16 more such files
  - pdfFiles/
    - UdpandTcpHighSpeed_delay_Separate.pdf
    - UdpandTcpHighSpeed_delay_Simulataneous_Different_Start.pdf
     ........ 16 more such files
  - main-app6.cc
  - main-app6.h
```

Prerequisites before running:

1. Make sure that **ns3** is downloaded and installed in the Ubuntu environment. The instructions are here:

Download : https://www.nsnam.org/releases/ns-3-31/
Install : https://www.nsnam.org/wiki/Installation
YouTube video : https://youtu.be/TbsNjQBU9dI

2. Make sure **gnuplot** is installed in the Ubuntu environment. If not simply open a terminal (**Ctrl+Alt+T**) and type:

```
$ sudo apt-get install gnuplot
```

How to run and test:

Method = (typing all commands one after the other).

a. First make sure that the files main-app.cc and main-app.h are in the follwing path (basically inside the scratch folder) of ns3 installation.

/home/<user-name>/ns-allinone-3.31/ns-3.31/scratch/main-app6.cc
/home/<user-name>/ns-allinone-3.31/ns-3.31/scratch/main-app6.h

b. Now run the following command in the terminal (inside the ns-3.31 folder).

```
 $ ./waf --run "scratch/main-app6 --tcpProtocol=<Protocol
Name> --loopRuns=<count> --simultaneously=<0 or 1>
--offset=<time in sec> --runTime=<time in sec>
--packetSize=<size in bytes>"
```

All these parameters are optional, since we have already initialised it inside the code. Also note that runtime is in a factor of 5 seconds.
As is obvious 18 .plt files will be generated (9*2 (1 for delay, 1 for throughput)).

d. For anyfile, the **nomenclature** clarifies the **usage**. For instance,
UdpandTcpHighSpeed_delay_Simulataneous_Different_Start -> Uses **TcpHighSpeed**, computes **delay**, starts **simultaneously** , and has a **non-zero offset.**
Here are the **9 commands for easy access:**
Question 1.

```
 $ ./waf --run "scratch/main-app6
--tcpProtocol=TcpHighSpeed --loopRuns=30 --simultaneously=0
--offset=0 --runTime=1 --packetSize=1024"
```

```
    $ ./waf --run "scratch/main-app6
--tcpProtocol=TcpScalable --loopRuns=30 --simultaneously=0
--offset=0 --runTime=1 --packetSize=1024"

    $ ./waf --run "scratch/main-app6 --tcpProtocol=TcpVegas
--loopRuns=30 --simultaneously=0 --offset=0 --runTime=1
--packetSize=1024"
```

Question 2.

```
$ ./waf --run "scratch/main-app6 --tcpProtocol=TcpHighSpeed
--loopRuns=30 --simultaneously=1 --offset=0 --runTime=1
--packetSize=1024"

    $ ./waf --run "scratch/main-app6
--tcpProtocol=TcpScalable --loopRuns=30 --simultaneously=1
--offset=0 --runTime=1 --packetSize=1024"

    $ ./waf --run "scratch/main-app6 --tcpProtocol=TcpVegas
--loopRuns=30 --simultaneously=1 --offset=0 --runTime=1
--packetSize=1024"

    $ ./waf --run "scratch/main-app6
--tcpProtocol=TcpHighSpeed --loopRuns=30 --simultaneously=1
--offset=3 --runTime=1 --packetSize=1024"

    $ ./waf --run "scratch/main-app6
--tcpProtocol=TcpScalable --loopRuns=30 --simultaneously=1
--offset=3 --runTime=1 --packetSize=1024"

    $ ./waf --run "scratch/main-app6 --tcpProtocol=TcpVegas
--loopRuns=30 --simultaneously=1 --offset=3 --runTime=1
--packetSize=1024"
```

e. Total of **18 .plt** files will be generated. For each .plt file the corresponding .pdf file is simply obtained by running :

```
    $ gnuplot x.plt
```

where x is a file name.