14 February 2024

CS 558: Computer Systems Lab

Report on Network Simulation Using NS-3 (Assignment 3)

Group 5

Akshay Bhosale	234101006	a.bhosale@iitg.ac.in
Ritik Kumar Koshta	234101044	r.koshta@iitg.ac.in
Rumit Gore	234101045	g.rumit@iitg.ac.in

Assignment 2: Network Protocol Analysis Using Wireshark

Title: Simulating a Computer Network with ns-3

Introduction

In this project, we're using a computer network simulator called ns-3. We want to see how two data types, Constant Bit Rate (CBR) and File Transfer Protocol (FTP), travel through the network. CBR data goes through a fast, direct route (like delivering a letter quickly), while FTP data follows a slower, more reliable path (like mailing a package).

To set up ns-3 on Ubuntu, we followed a tutorial video on YouTube. The video guided us through the installation process step by step. We found it helpful and easy to follow, which made the installation process smooth.

Network Setup

To create our network, we made a virtual version using ns-3. We chose a Dumbbell shape, which looks like a dumbbell with two routers (R1 and R2) in the middle. Each router connects to two hosts, making four hosts (H1, H2, H3, and H4).

The main link connecting the routers was like the backbone of our network. It had a speed of 30 Mbps and took 100 ms to send data between routers. This link is important because it connects everything together.

Each computer was connected to its router with its link. These links had different speeds and delays. For example, the links for H1 and H2 had a speed of 80 Mbps and took 20 ms to send data. The same goes for H3 and H4.

We set up the network this way to see how different types of data (CBR and FTP) perform in different situations.

Traffic Generation

- ➢ CBR Traffic over UDP: Attached to H1
- ➢ FTP Traffic over TCP: Attached to H2

Experimental Setup

- ➢ Packet sizes are chosen appropriately for the experiments.
- ➢ We compare the delay and throughput of CBR and FTP traffic streams under different scenarios.

How to Run

- ➢ Put the code file in the Scratch folder.
- ➢ Open Terminal.
- ➢ Go to the directory "ns-allinone-3.40/ns-3.40 ".
- ➢ Use “./ns3 run main.cc” with the required parameters to run the code.
- ➢ As per our code, it will generate “.plt" files for delay and throughput.
- ➢ We used gnuplot to plot the required graphs using the command "gnuplot x.plt".

Experiment 1: Testing Single Traffic

We tested sending just one type of data (CBR or FTP) through the network.
- We looked at how long data took to get through (delay) and how fast it traveled (throughput).
- We tested different sizes of data packets to see how it affected things.
- terminal commands used are provided
-Compared results with different packet sizes (x-axis) affecting delay (y-axis).

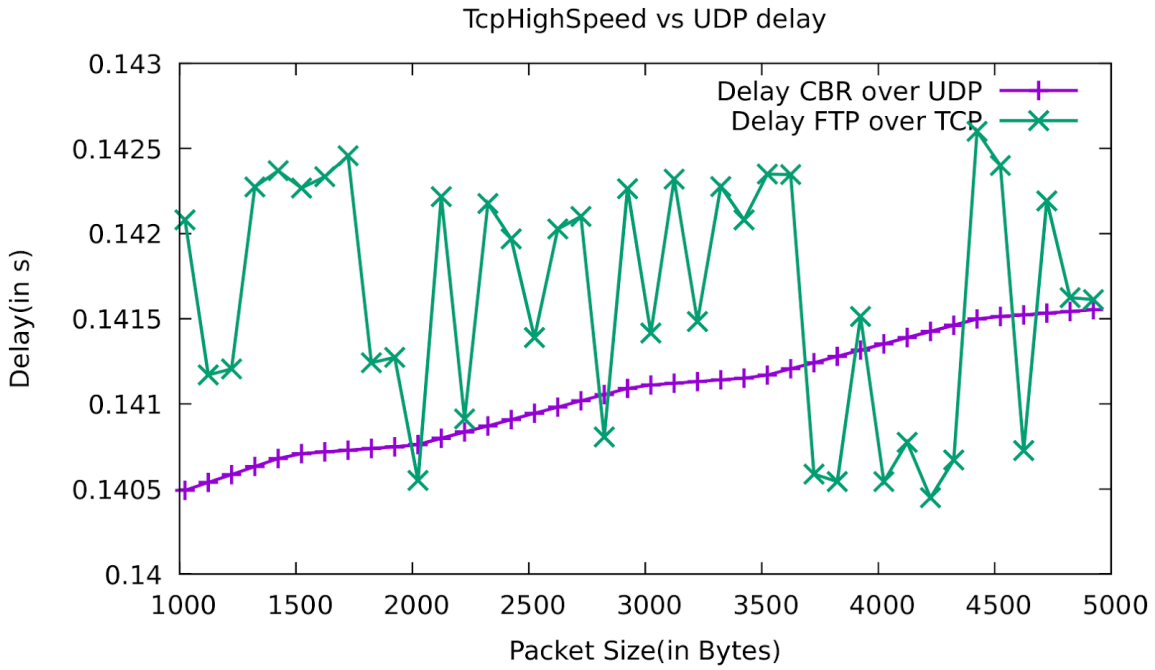
For TCP Highspeed
- terminal commands used

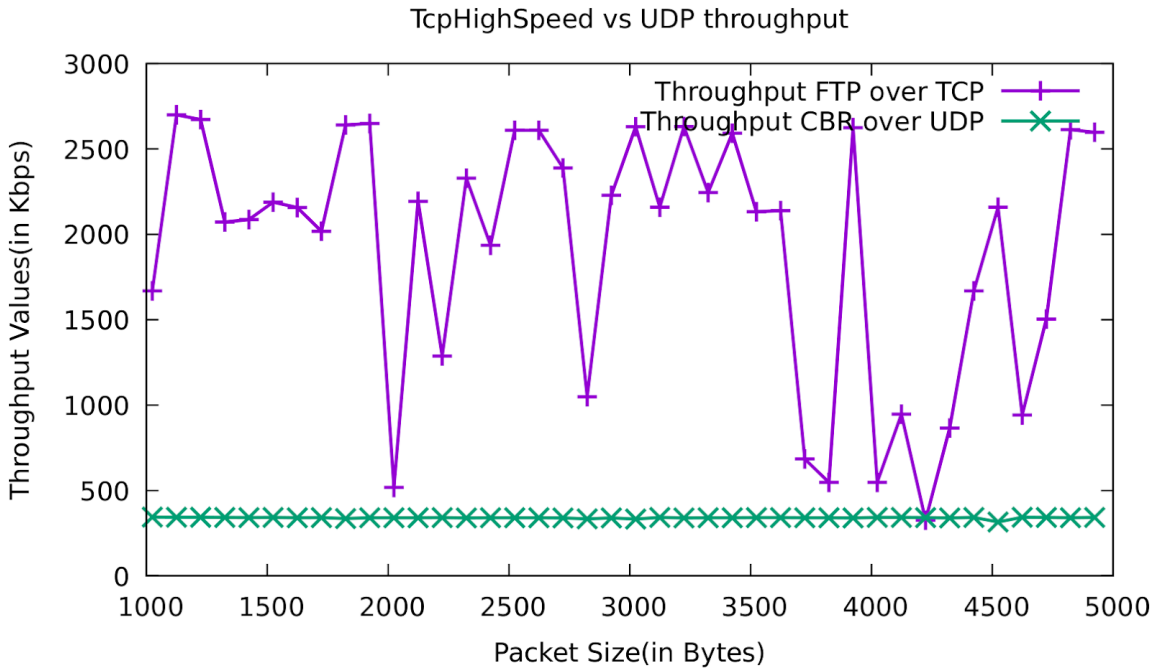
```
./ns3 run "scratch/main --prot=TcpHighSpeed --for_loop=40
--simultaneously=0 --offset=2 --run_time=1 --packetsize=1024"
```

- then for Plotting Graph

```
gnuplot TcpHighspeed_delay_Seperate.plt
gnuplot TcpHighspeed_throughput_Seperate.plt
```

- graph for delay and throughput observed with different packet sizes





For TCP Vegas

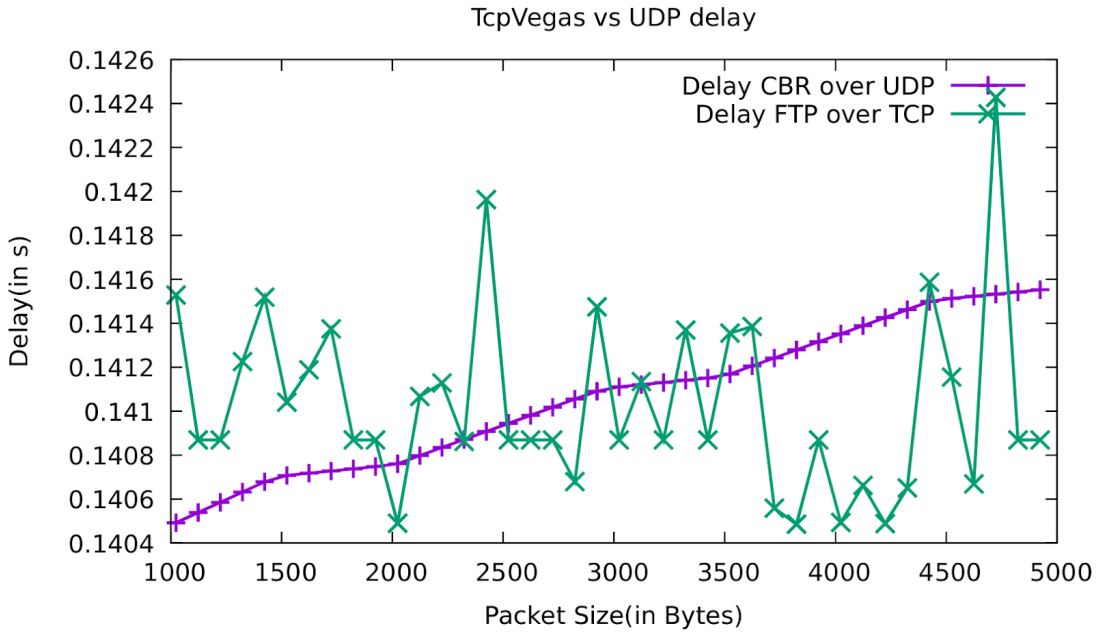
- terminal commands used

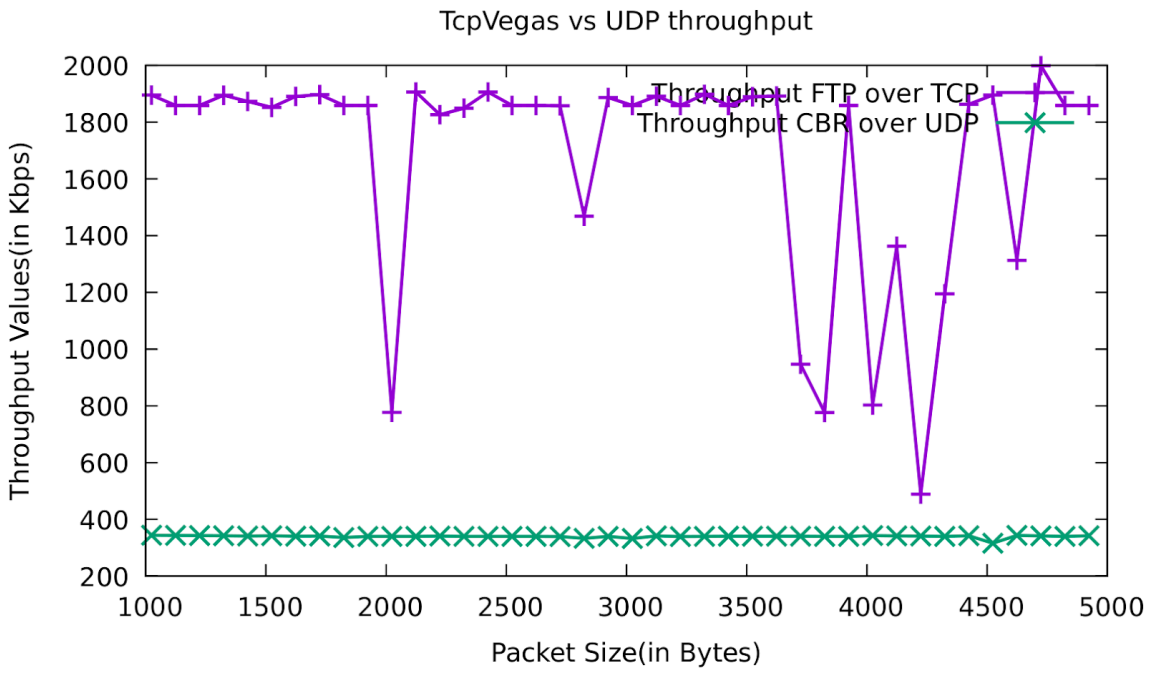
```
./ns3 run "scratch/main --prot=TcpVegas --for_loop=40 --simultaneously=0
--offset=2 --run_time=1 --packetsize=1024"
```

- then for Plotting Graph

```
gnuplot TcpVegas_delay_Seperate.plt
gnuplot TcpVegas_throughput_Seperate.plt
```

- graph for delay and throughput observed with different packet sizes





For TCP Scalable

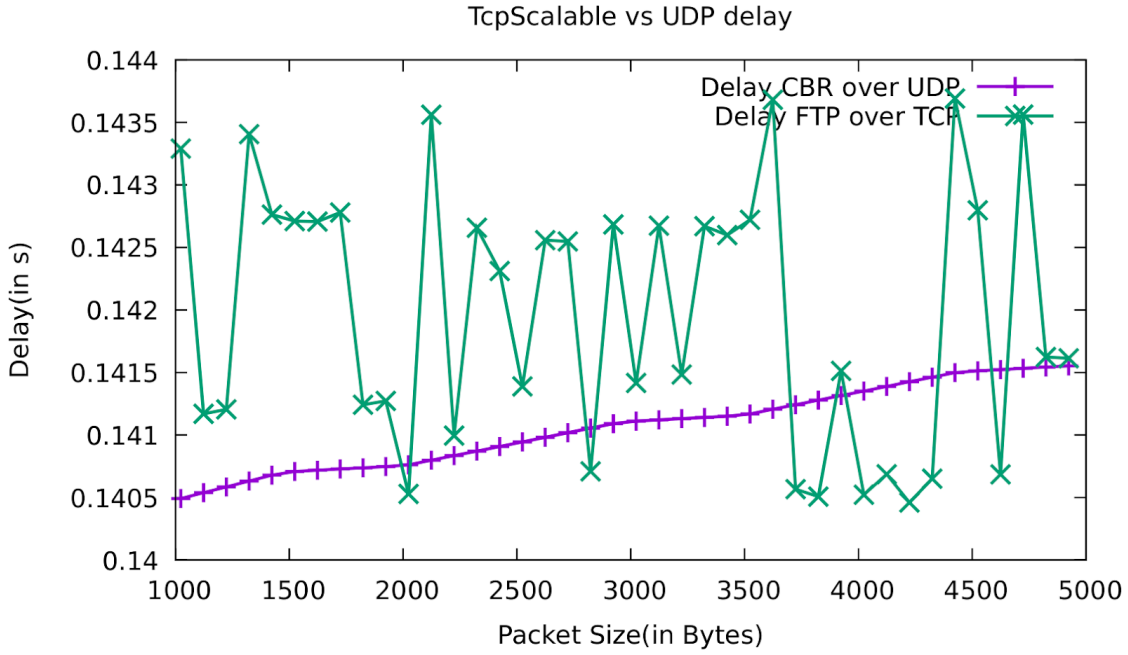
- terminal commands used

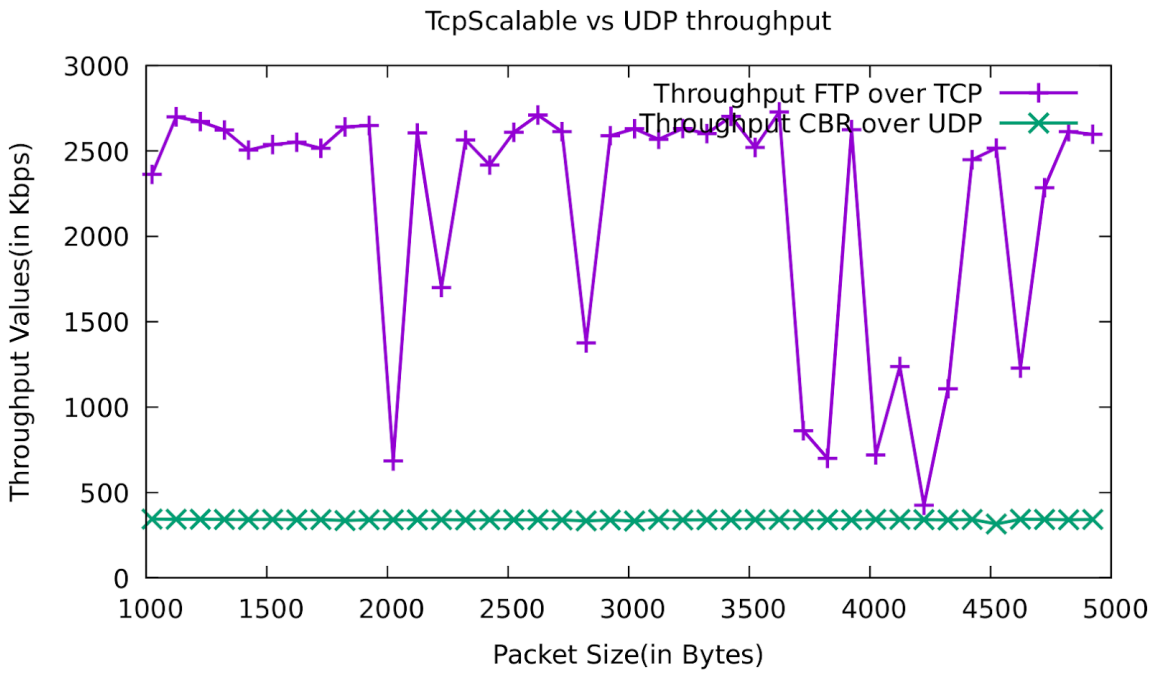
```
./ns3 run "scratch/main --prot=TcpScalable --for_loop=40
--simultaneously=0 --offset=2 --run_time=1 --packetsize=1024"
```

- then for Plotting Graph

```
gnuplot TcpScalable_delay_Seperate.plt
gnuplot TcpScalable_throughput_Seperate.plt
```

- graph for delay and throughput observed with different packet sizes





Experiment 2: Testing Multiple Traffic

Next, we sent both data types through the network simultaneously, sometimes one after the other.

- Again, we looked at how long it took for data to travel and how fast it went.
- We tested different sizes of data packets to compare results.

For Same Time

For TCP Highspeed

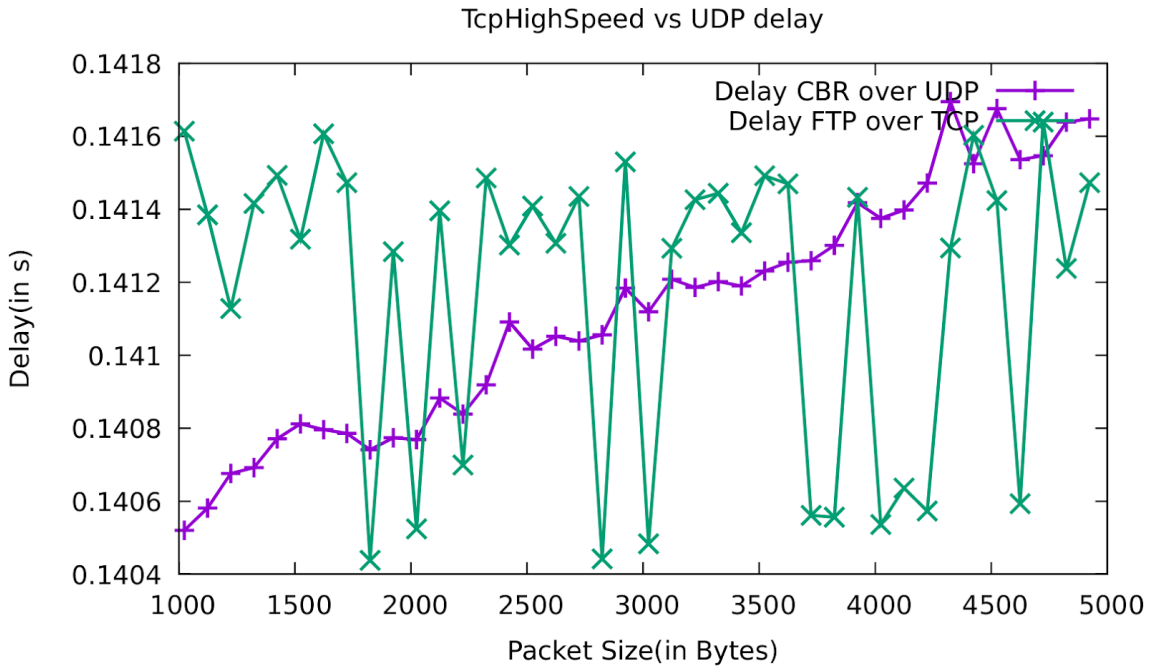
- terminal commands used

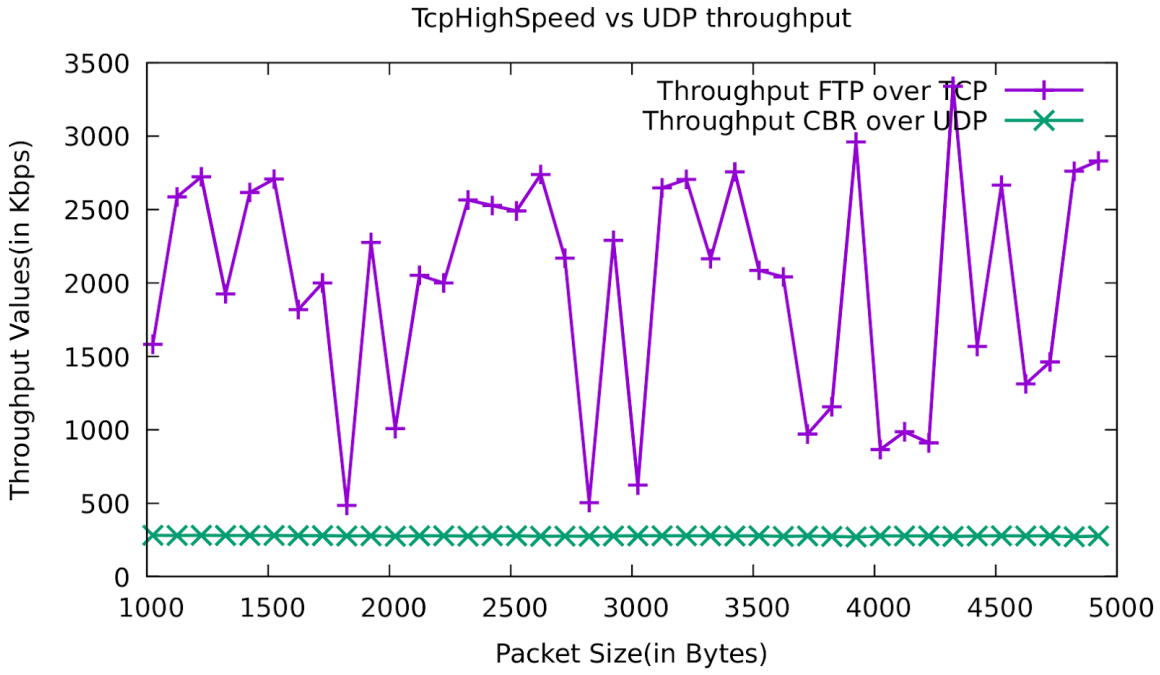
```
./ns3 run "scratch/main --prot=TcpHighSpeed --for_loop=40
--simultaneously=1 --offset=0 --run_time=1 --packetsize=1024"
```

- then for Plotting Graph

```
gnuplot TcpHighspeed_delay_Simultaneous_Same_Start.plt
gnuplot TcpHighspeed_throughput_Simultaneous_Same_Start.plt
```

- graph for delay and throughput observed with different packet sizes





For TCP Vegas

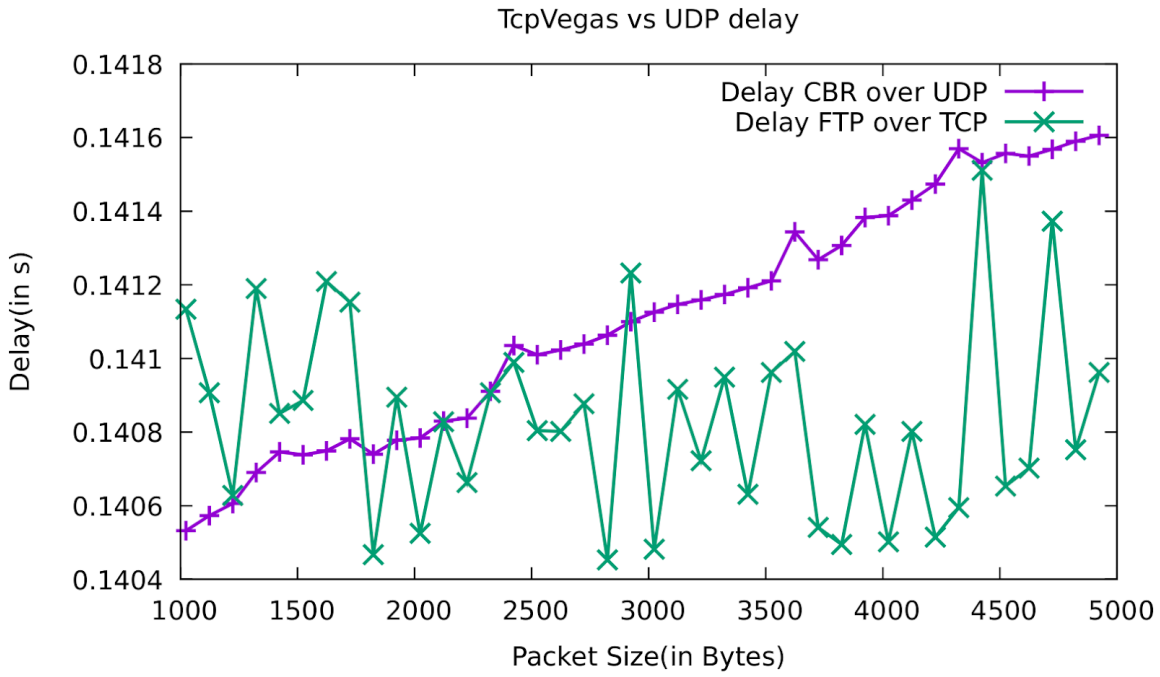
- terminal commands used

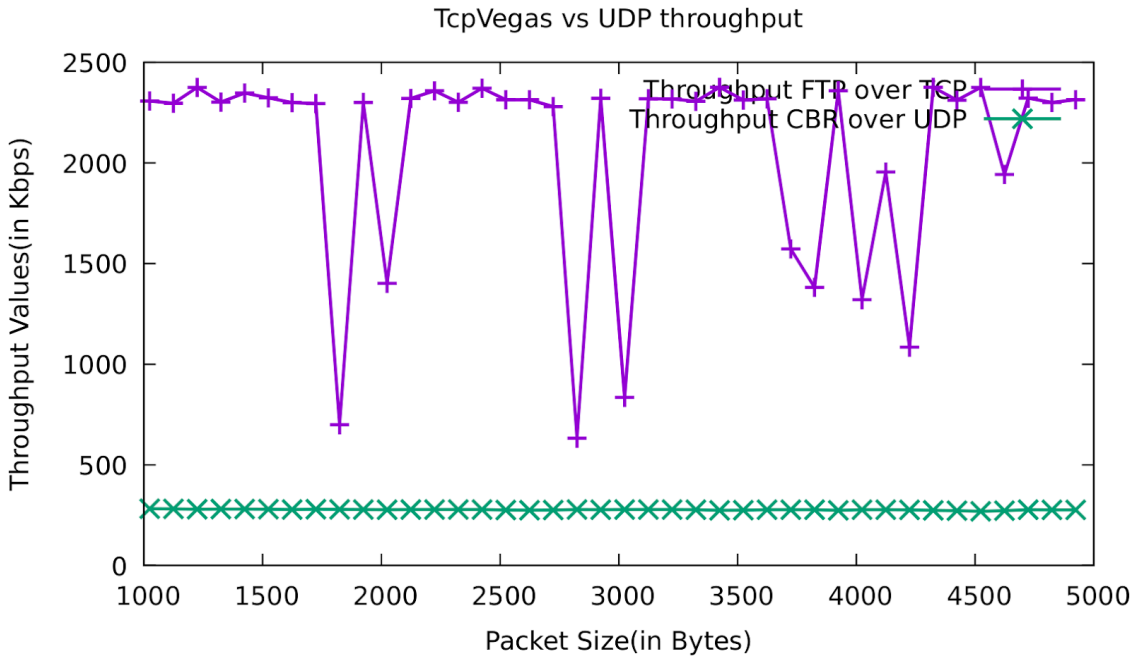
```
./ns3 run "scratch/main --prot=TcpVegas --for_loop=40 --simultaneously=1
--offset=0 --run_time=1 --packetsize=1024"
```

- then for Plotting Graph

```
gnuplot TcpVegas_delay_Simultaneous_Same_Start.plt
gnuplot TcpVegas_throughput_Simultaneous_Same_Start.plt
```

- graph for delay and throughput observed with different packet sizes





For TCP Scalable

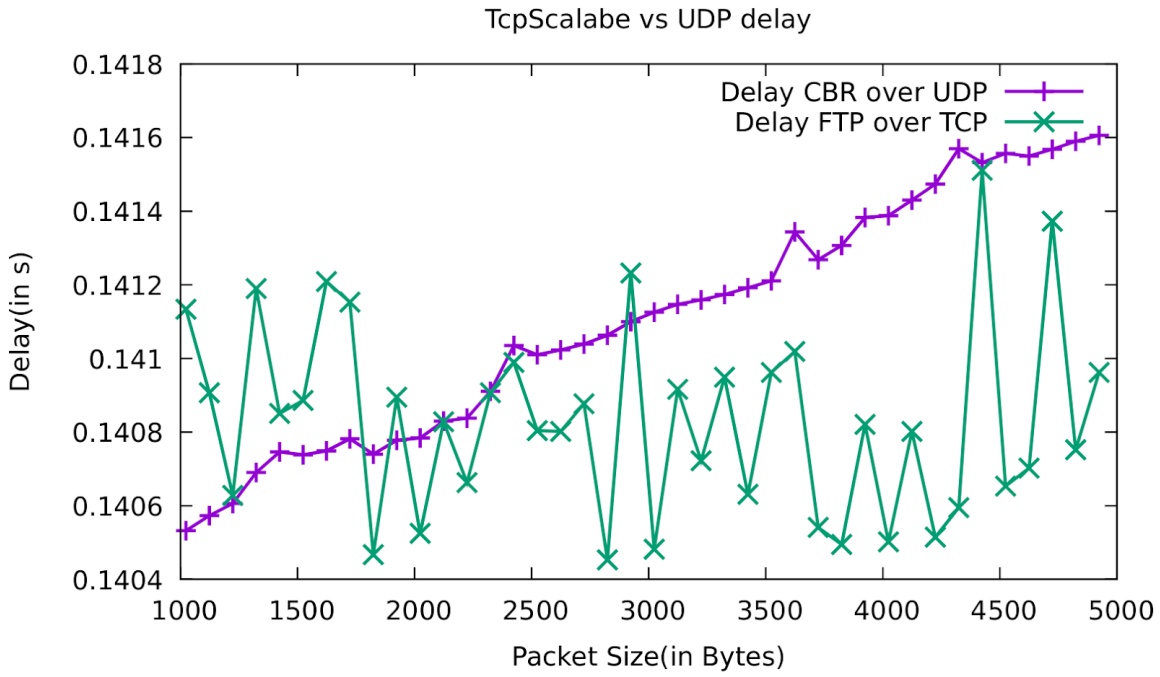
- terminal commands used

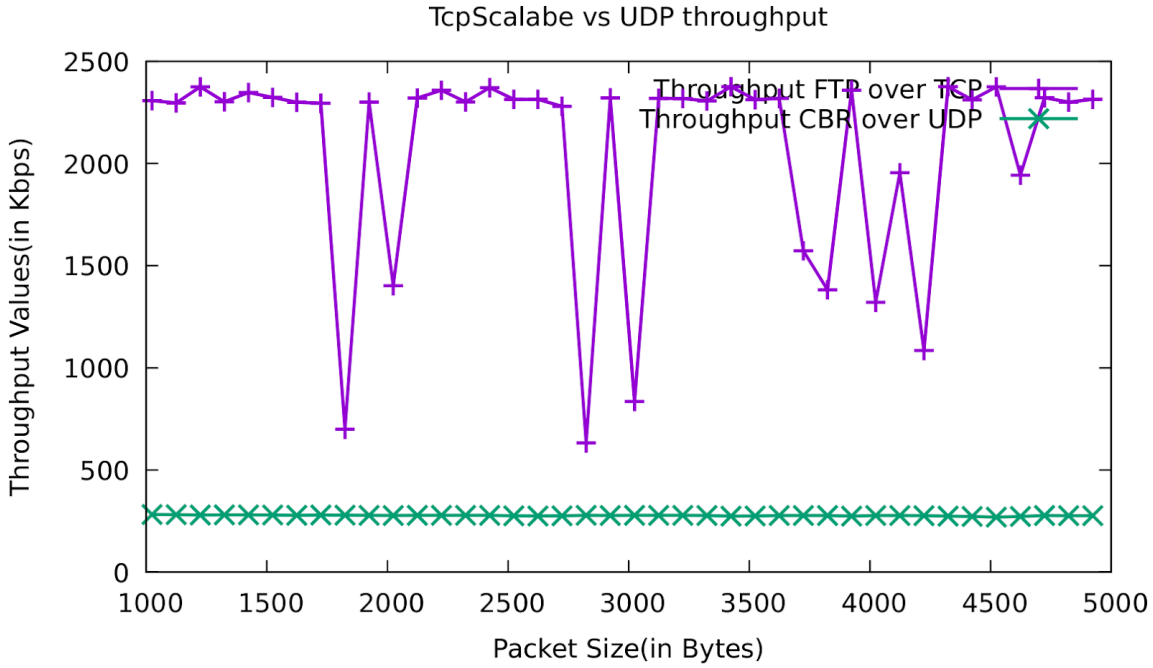
```
./ns3 run "scratch/main --prot=TcpScalable --for_loop=40
--simultaneously=1 --offset=0 --run_time=1 --packetsize=1024"
```

- then for Plotting Graph

```
gnuplot TcpScalable_delay_Simultaneous_Same_Start.plt
gnuplot TcpScalable_throughput_Simultaneous_Same_Start.plt
```

- graph for delay and throughput observed with different packet sizes





For Different Time

For TCP Highspeed

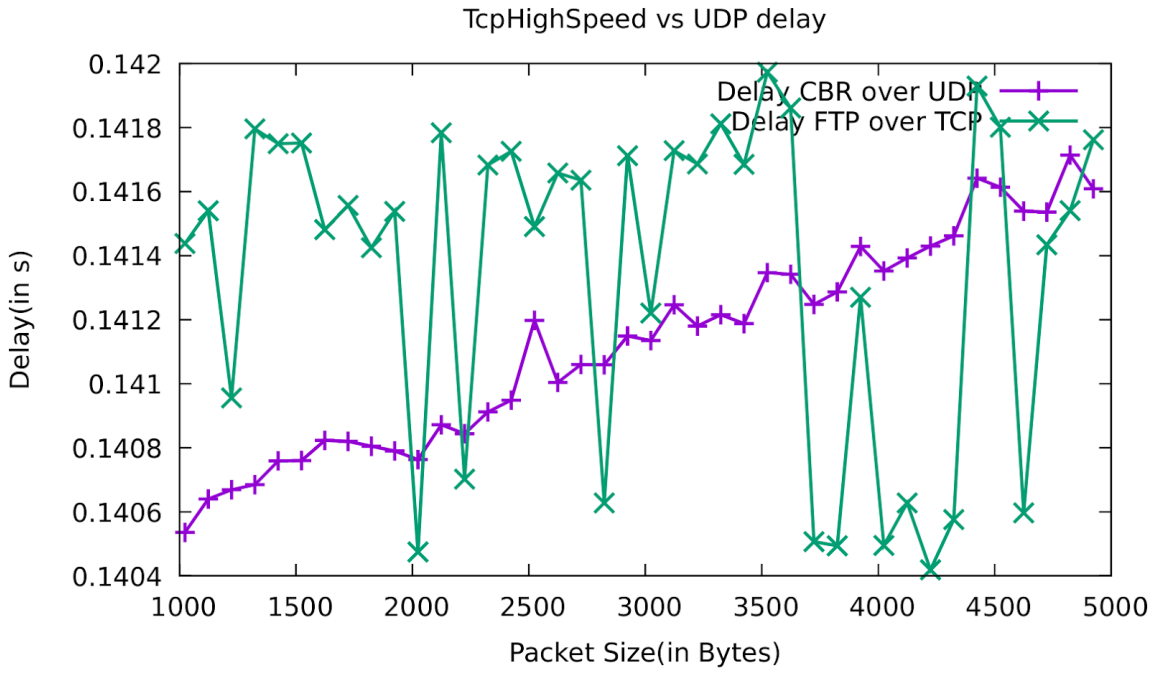
- terminal commands used

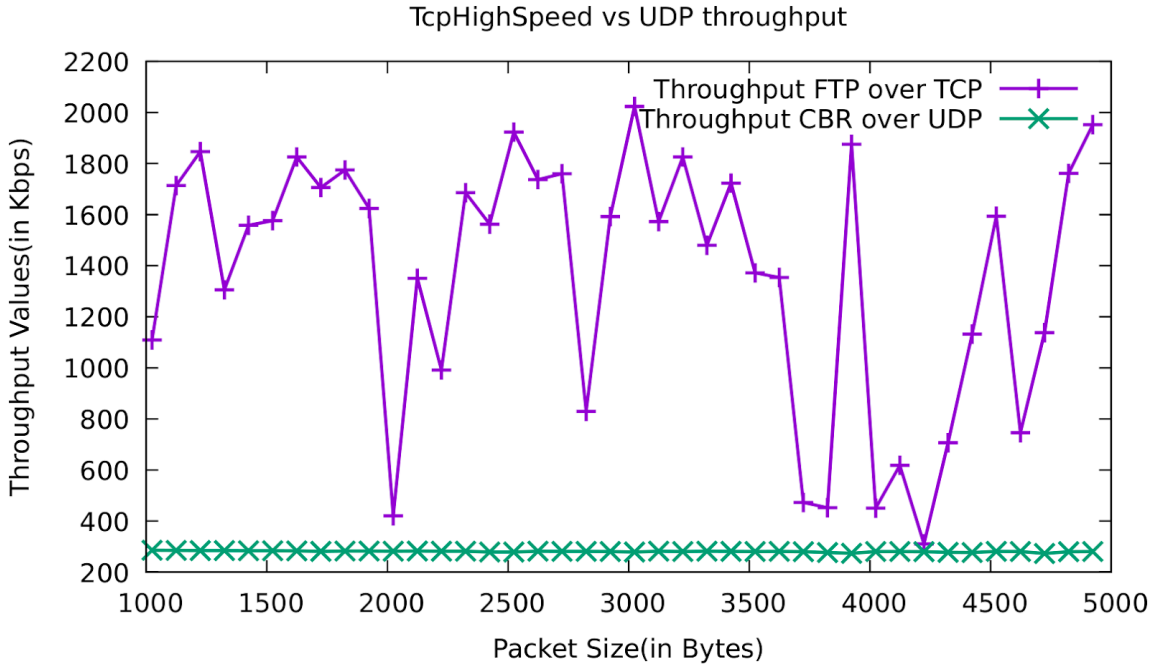
```
./ns3 run "scratch/main --prot=TcpHighSpeed --for_loop=40
--simultaneously=1 --offset=2 --run_time=1 --packetsize=1024"
```

- then for Plotting Graph

```
gnuplot TcpHighspeed_delay_Seperate.plt
gnuplot TcpHighspeed_throughput_Seperate.plt
```

- graph for delay and throughput observed with different packet sizes





For TCP Vegas

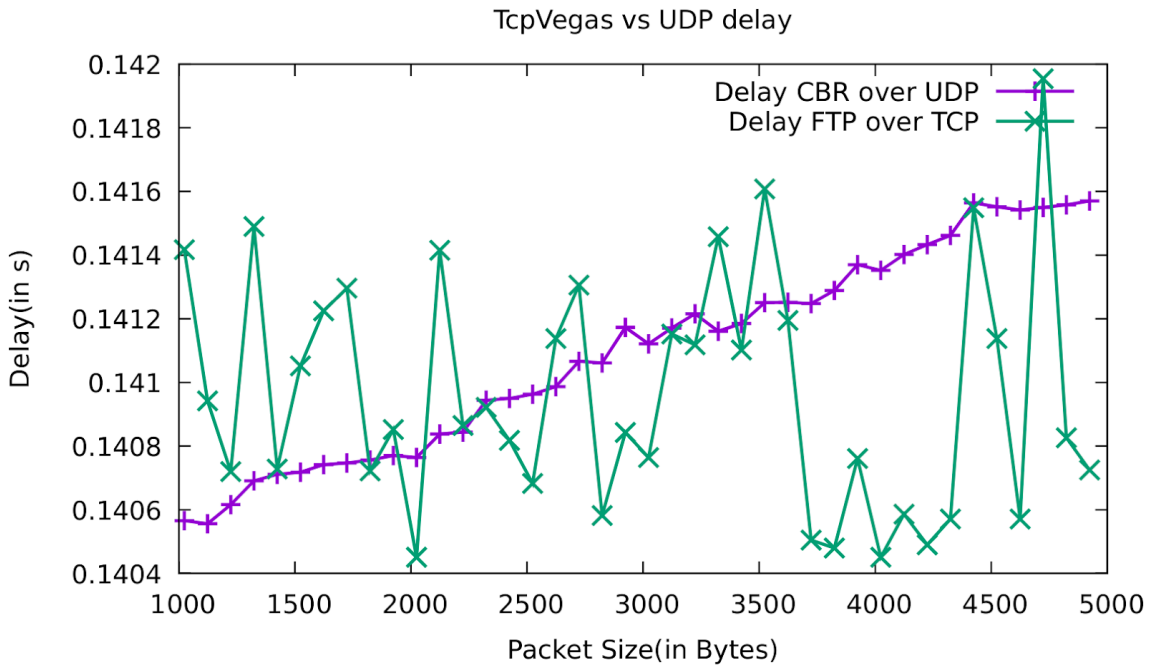
- terminal commands used

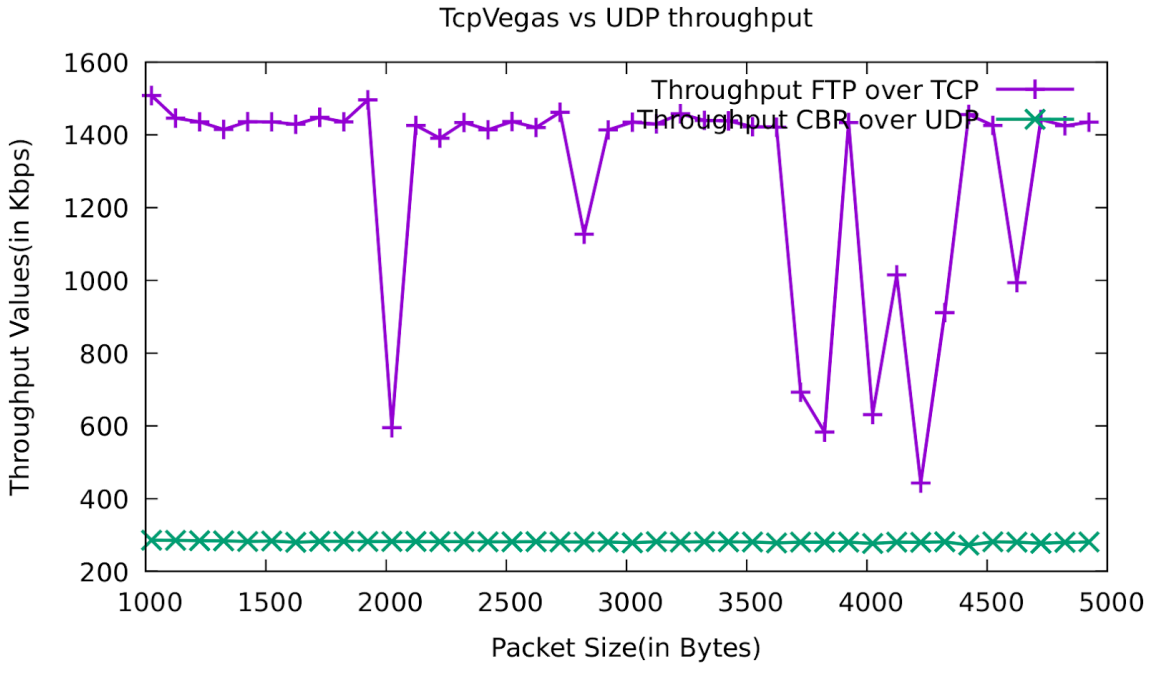
```
./ns3 run "scratch/main --prot=TcpVegas --for_loop=40 --simultaneously=1
--offset=2 --run_time=1 --packetsize=1024"
```

- then for Plotting Graph

```
gnuplot TcpVegas_delay_Seperate.plt
gnuplot TcpVegas_throughput_Seperate.plt
```

- graph for delay and throughput observed with different packet sizes





For TCP Scalable

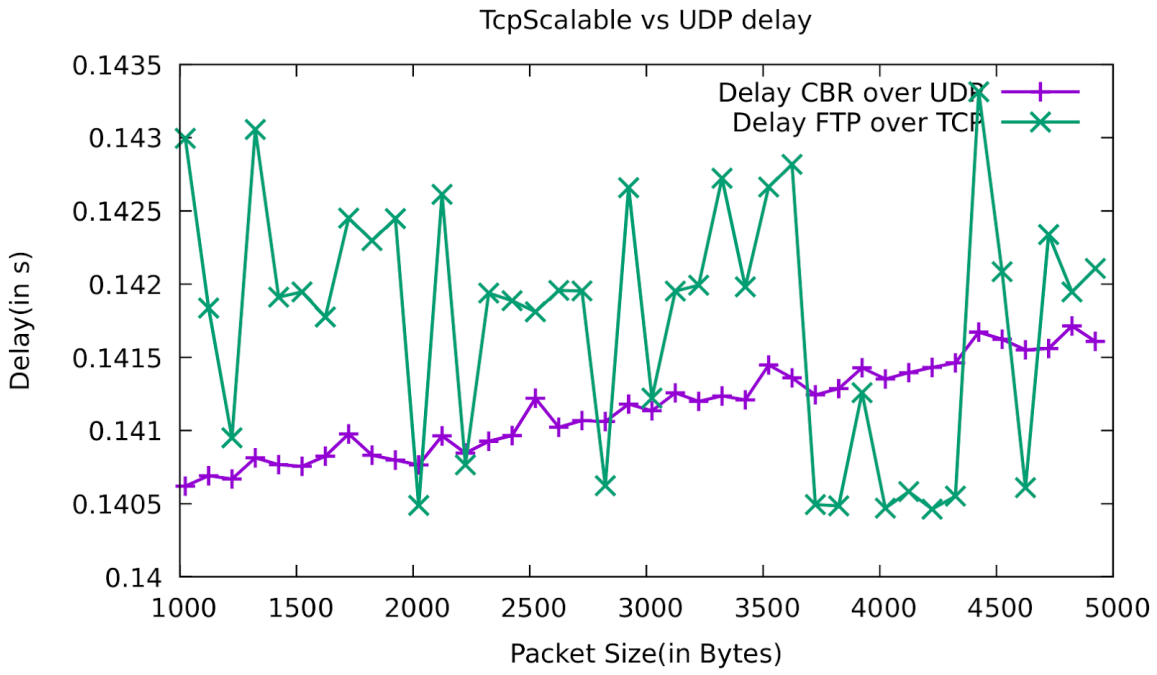
- terminal commands used

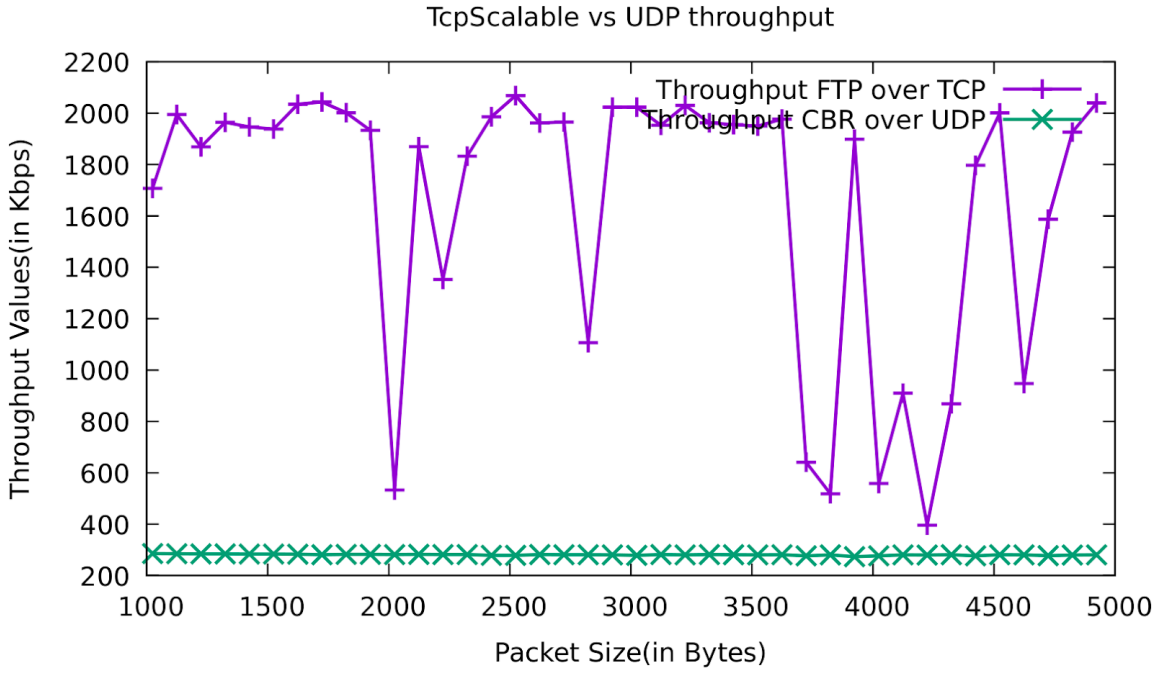
```
./ns3 run "scratch/main --prot=TcpScalable --for_loop=40
--simultaneously=1 --offset=2 --run_time=1 --packetsize=1024"
```

- then for Plotting Graph

```
gnuplot TcpScalable_delay_Seperate.plt
gnuplot TcpScalable_throughput_Seperate.plt
```

- graph for delay and throughput observed with different packet sizes





Conclusion

- CBR data gets through quickly but doesn't handle large amounts well.
- FTP data takes longer but handles large amounts better.
- Choosing the right data type depends on what's important: speed or reliability.
