`

# Assignment 1

**Ashrujit Ghoshal (14CS10060)**

**Sayan Ghosh (14CS10061)**

Part 1.

a)

Steps:

The mininet network is set up using the command **sudo mn –link tc,bw=1,delay=1ms,loss=0**

Observations:

TCP Throughput = 957 Kbps

TCP Bandwidth = 1 Mbps

b)

Steps:

The network is again set up using **sudo mn –link tc,bw=1,delay=1ms,loss=0** command

Opening xterm h2, we set up server in UDP mode by iperf -u -s

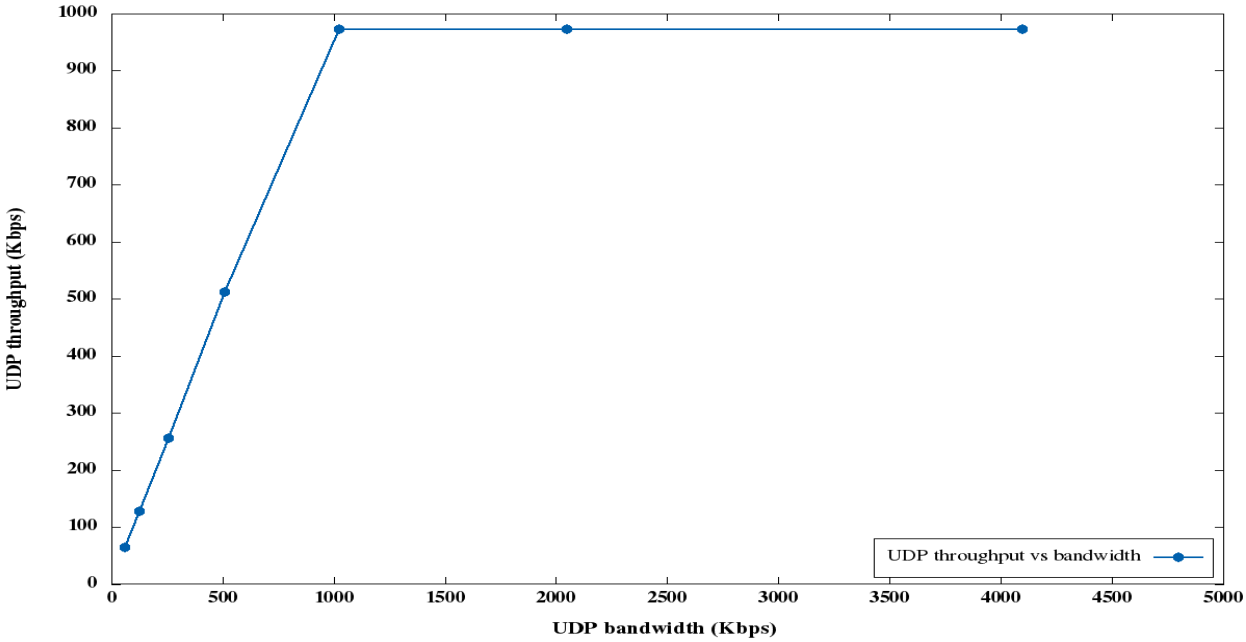
Openning xterm h1, we set up client and send packets to check throughput by using the command

Iperf -u -c 10.0.0.2 -b <bandwidth>

Observations :

Plot1:Plot of UDP bandwidth v UDP throughput

`




The graph showing the nature of variation of throughout w.r.t. bandwidth is shown below,

Justification :

The bandwidth of the connection is 1 Mbps. So, on sending less amount of data the thoughput is equal to the bandwidth of the UDP connection. But in case the UDP bandwidth is 1 Mbps or more, full amount of data is not being transferred due to limit of the link. Hence the value of the thoughput becomes less than the UDP bandwidth in these cases.

Part 2

a)

Steps:

To create a network with custom configuration we use the command **sudo mn –custom ~/mininet/custom/mytopo2.py –topo mytopo –link tc**

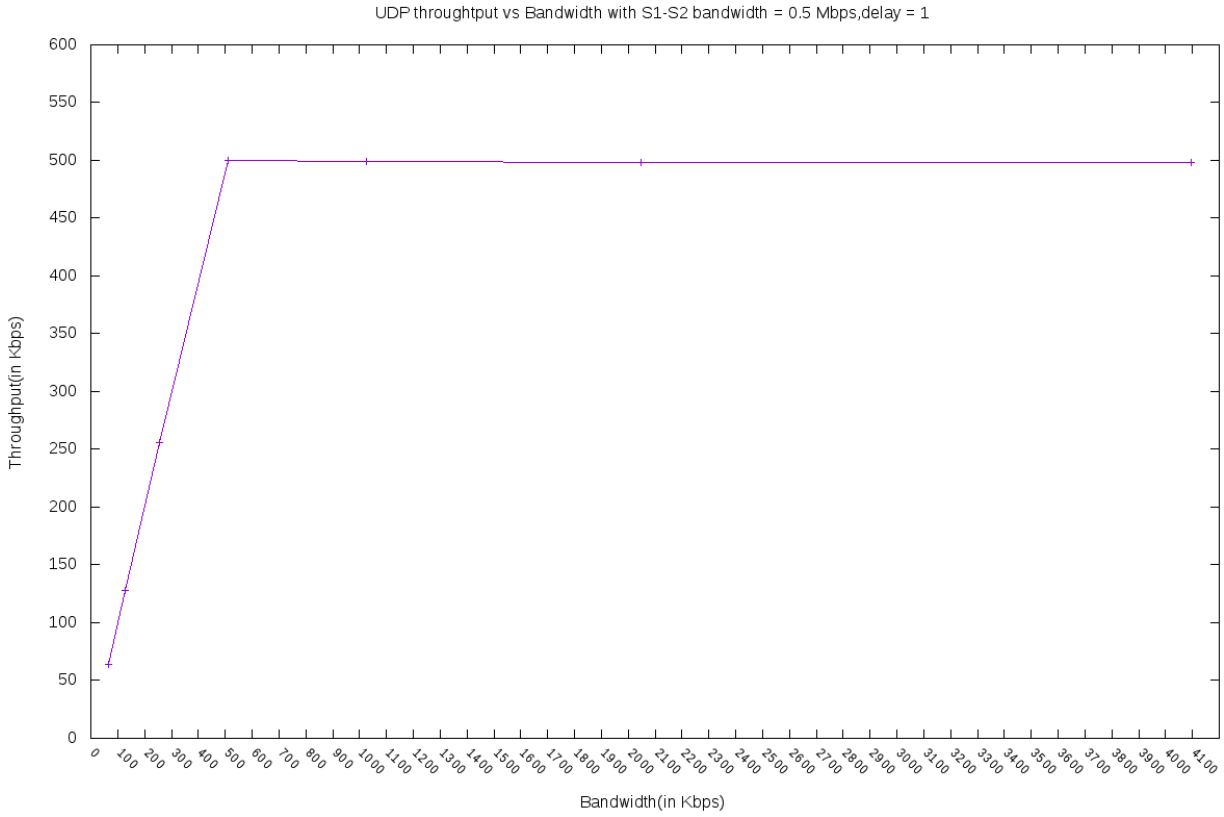
The file mytopo2.py is given in the folder. The parameters are changed every time and we need to destroy and recreate the network for the changes to be effected.

We take the readings similar to part 1.

Observations:

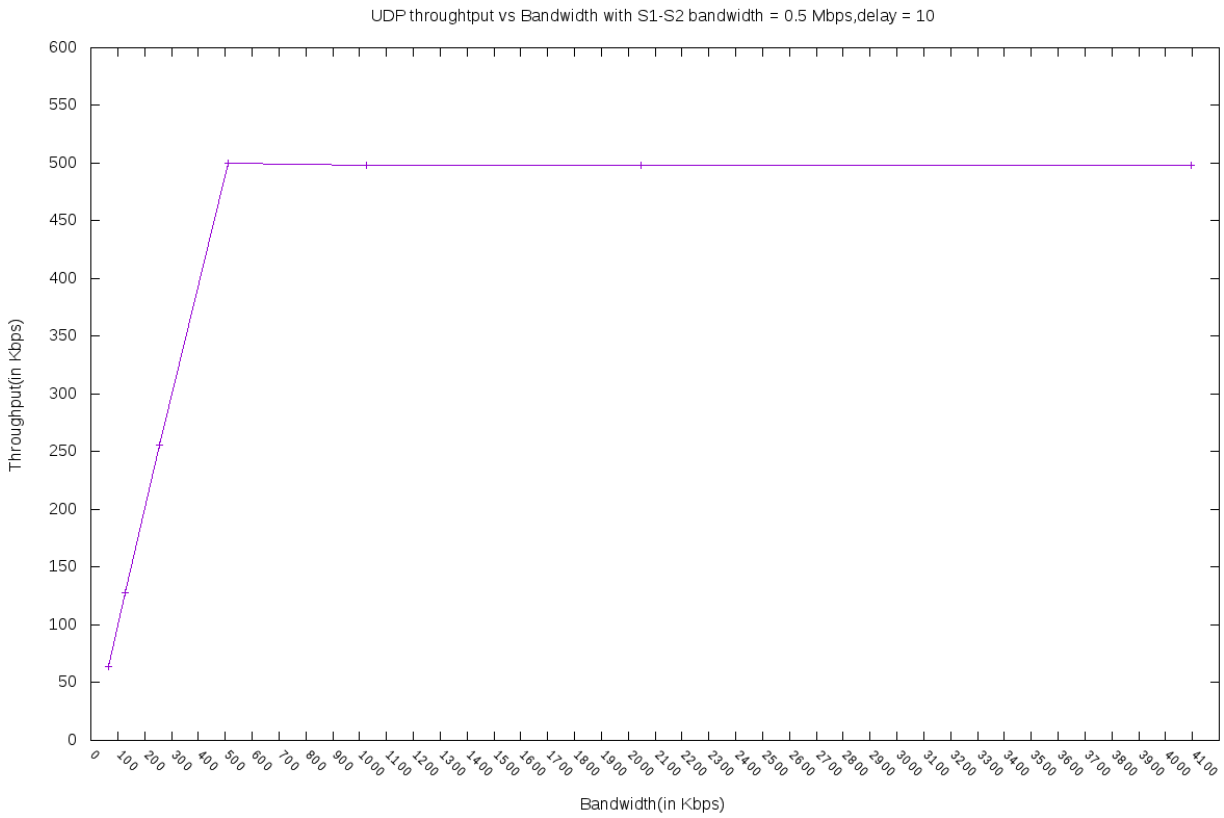
Plot2: UDP throughput vs Bandwidth with S1-S2 bandwidth=0.5Mbps,delay=1 ms

`



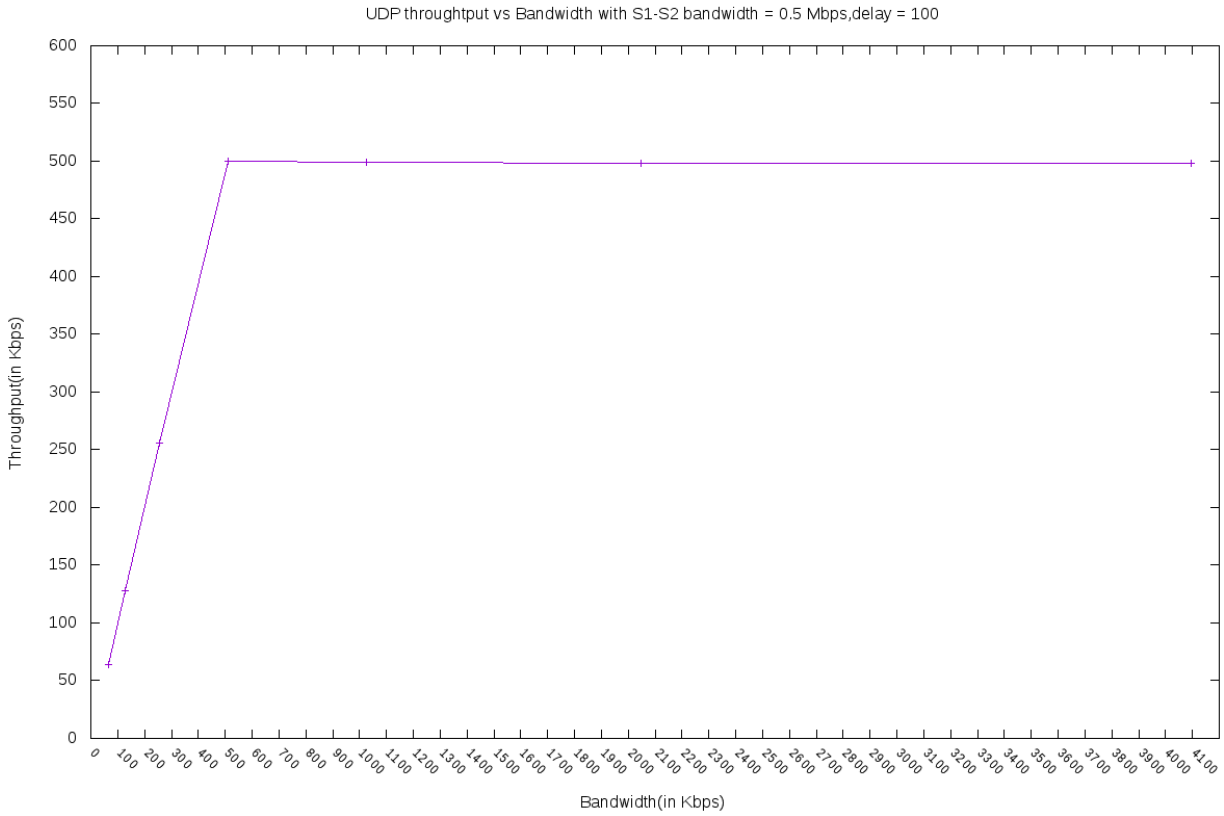

Plot3: UDP throughput vs Bandwidth with S1-S2 bandwidth=0.5Mbps,delay=10 ms

`



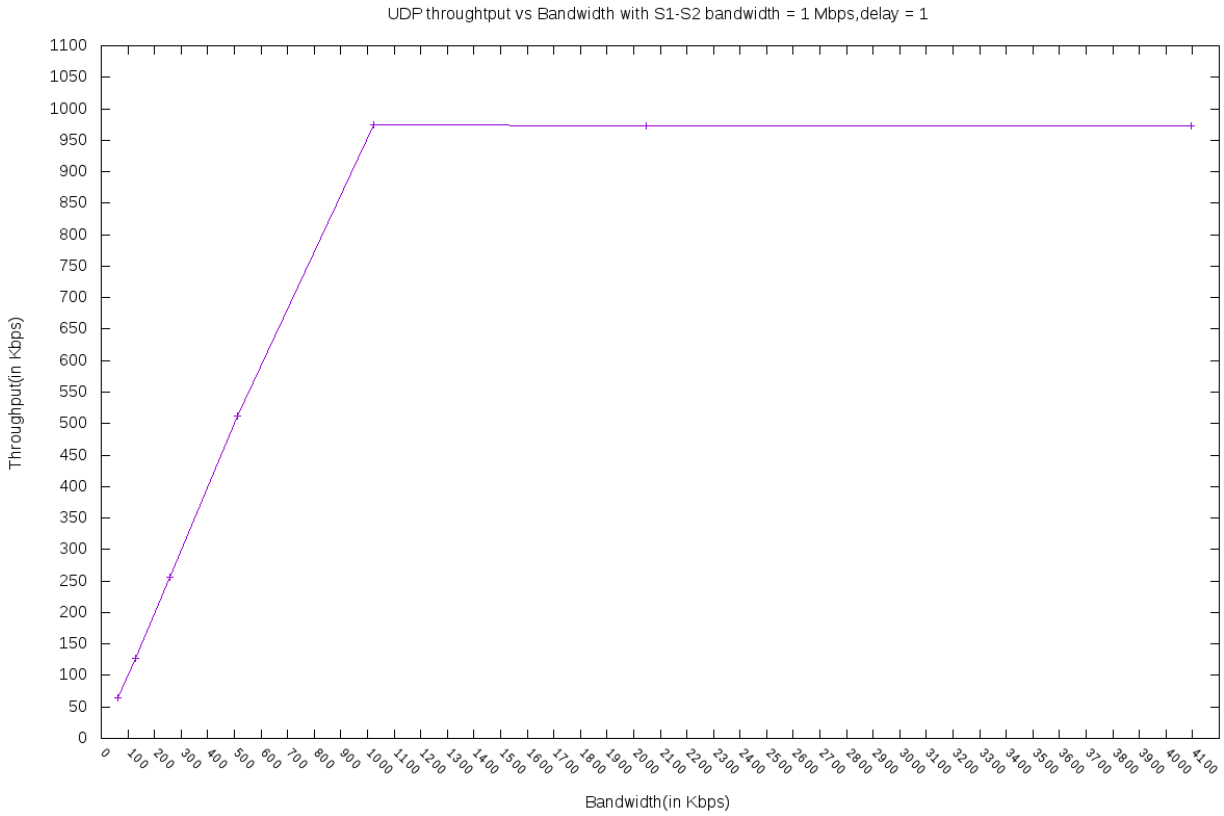

Plot4: UDP throughput vs Bandwidth with S1-S2 bandwidth=0.5Mbps,delay=100 ms

`



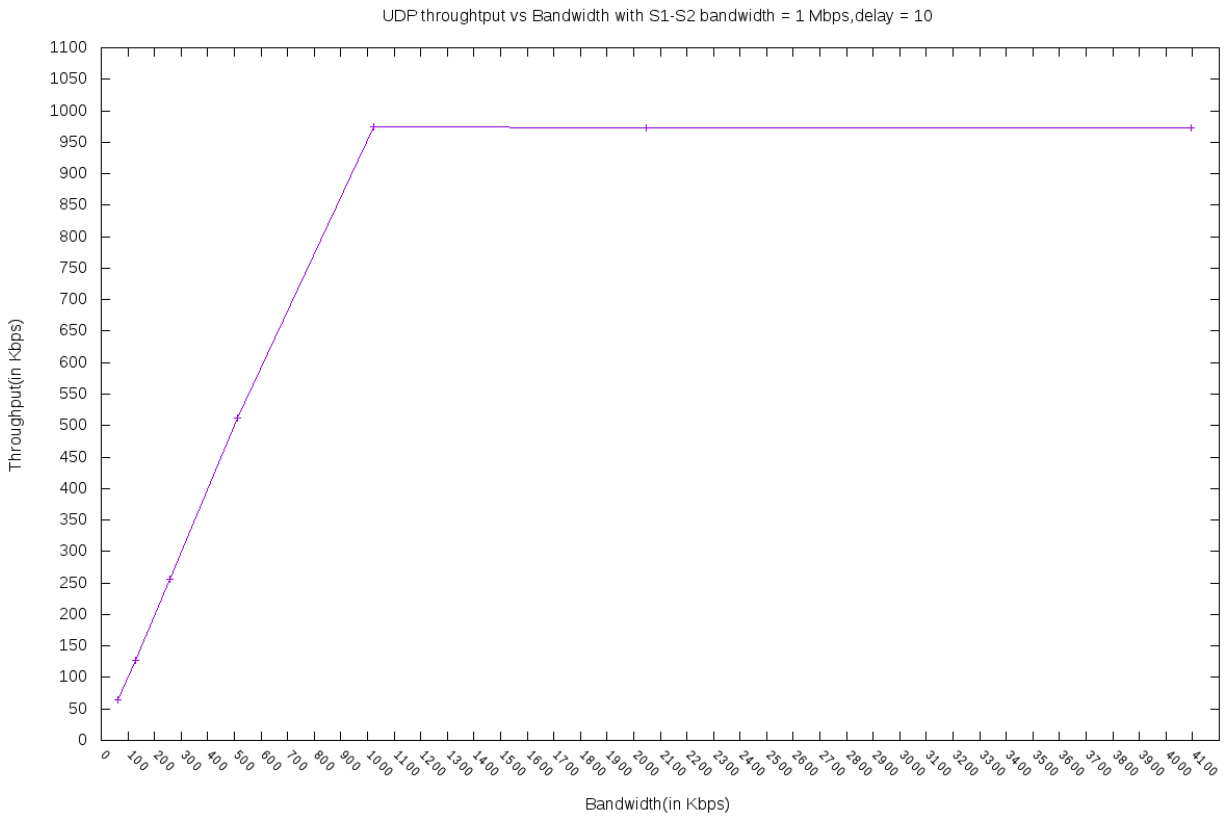

Plot5:UDP throughput vs Bandwidth with S1-S2 bandwidth=1Mbps,delay=1 ms

`



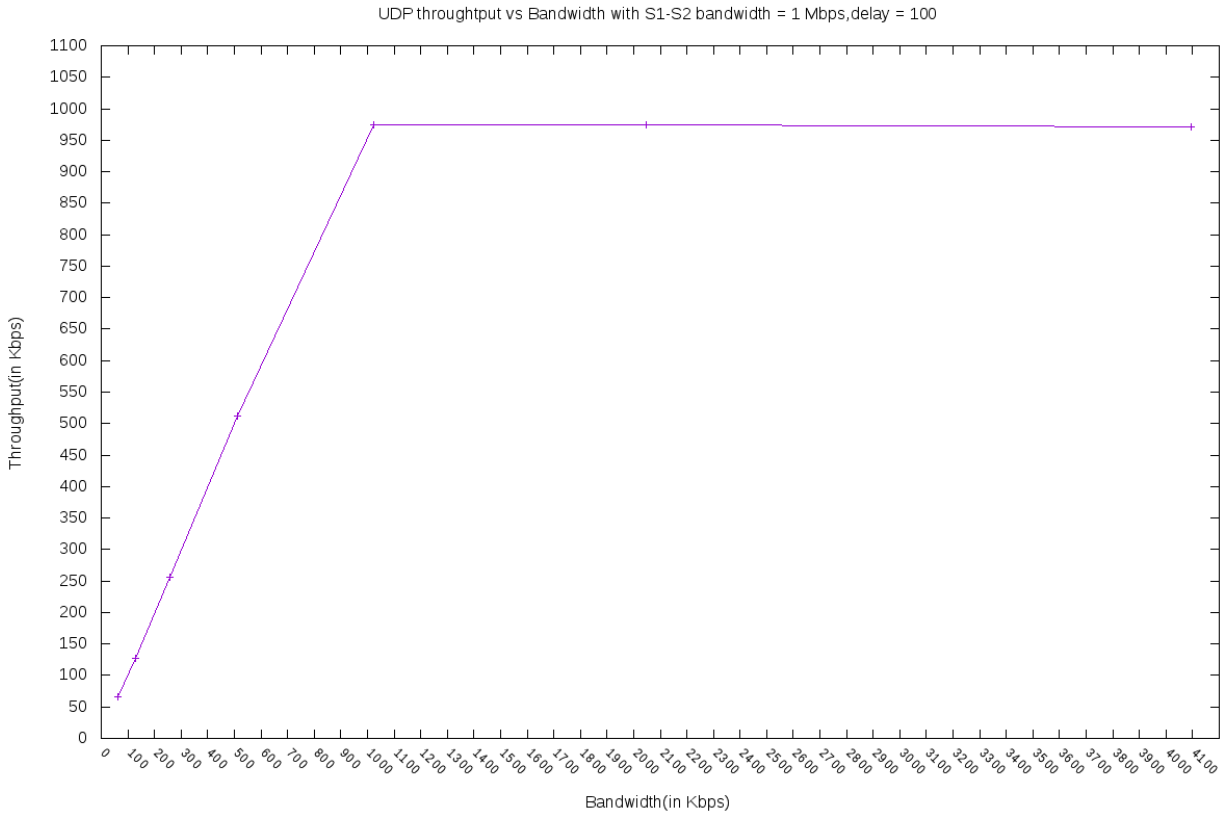

Plot6:UDP throughput vs Bandwidth with S1-S2 bandwidth=1Mbps,delay=10 ms

`



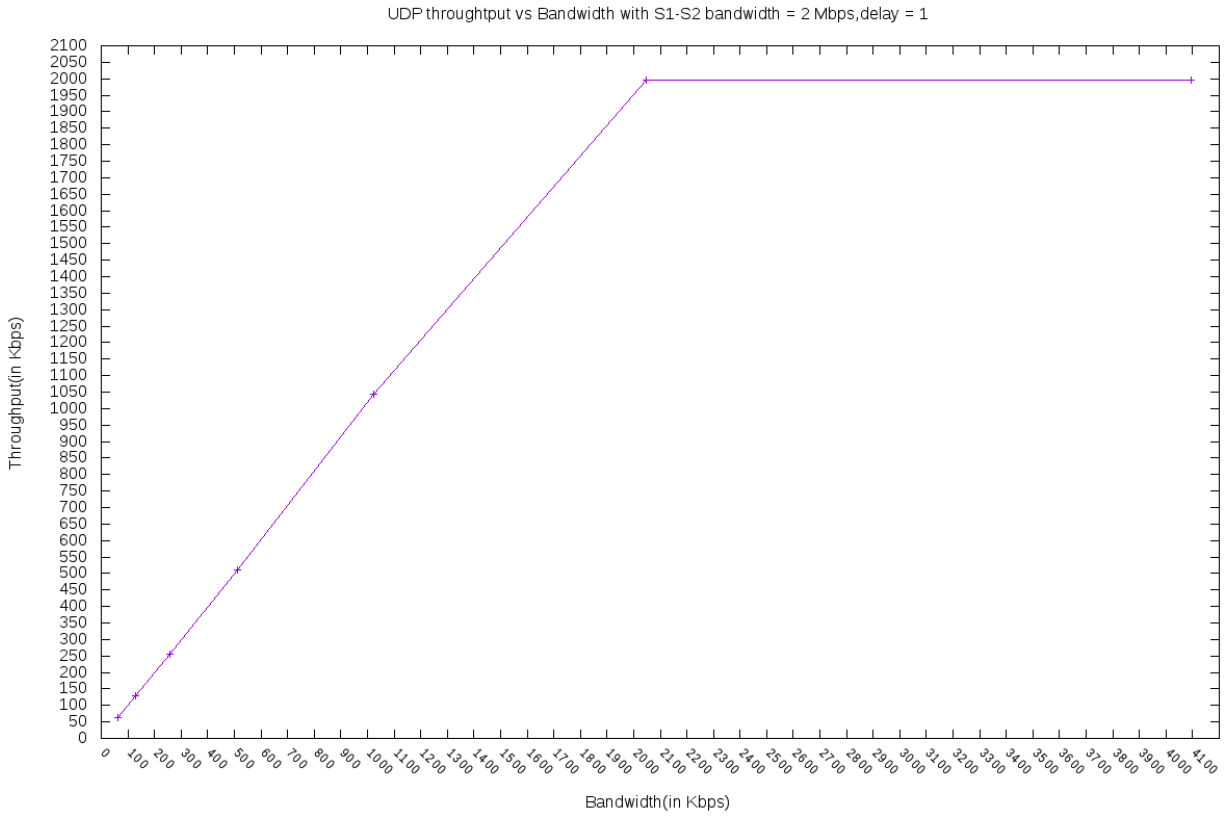

Plot7:UDP throughput vs Bandwidth with S1-S2 bandwidth=1Mbps,delay=100 ms

`



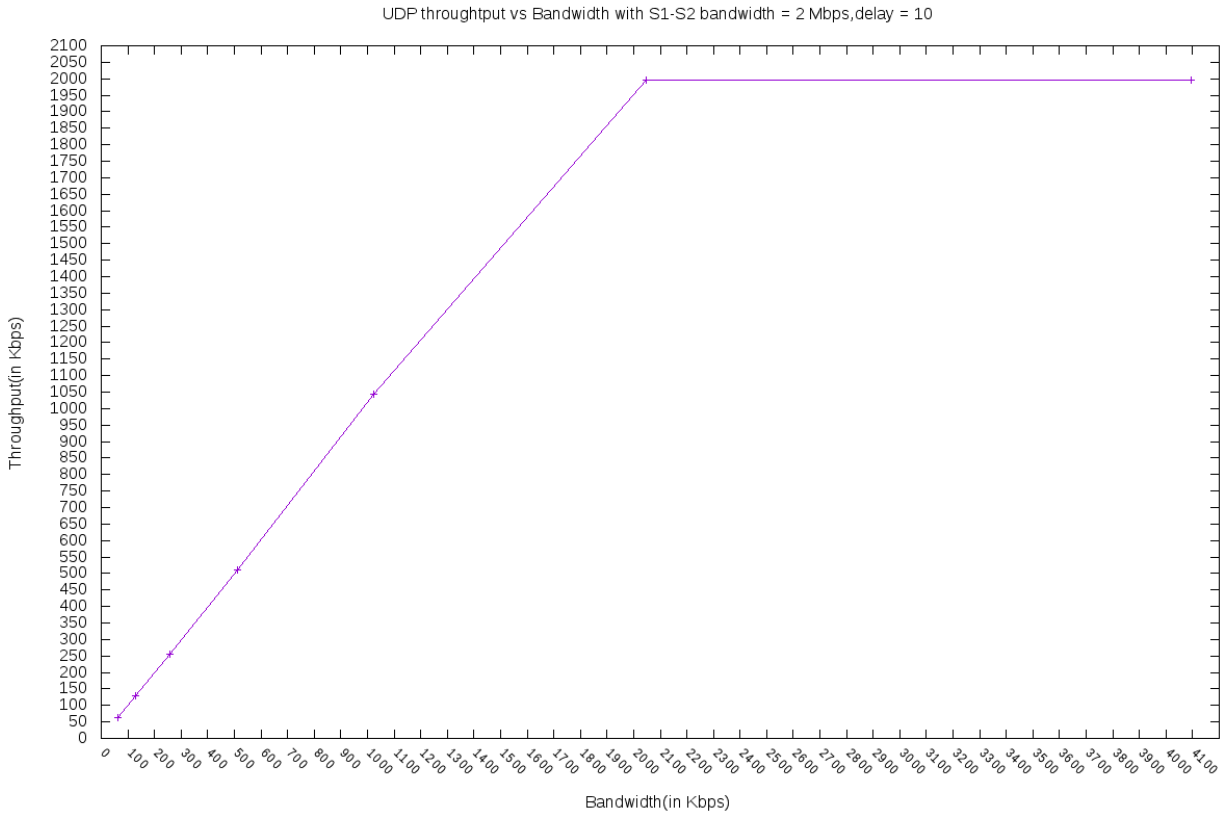

Plot8:UDP throughput vs Bandwidth with S1-S2 bandwidth=2Mbps,delay=1 ms

`



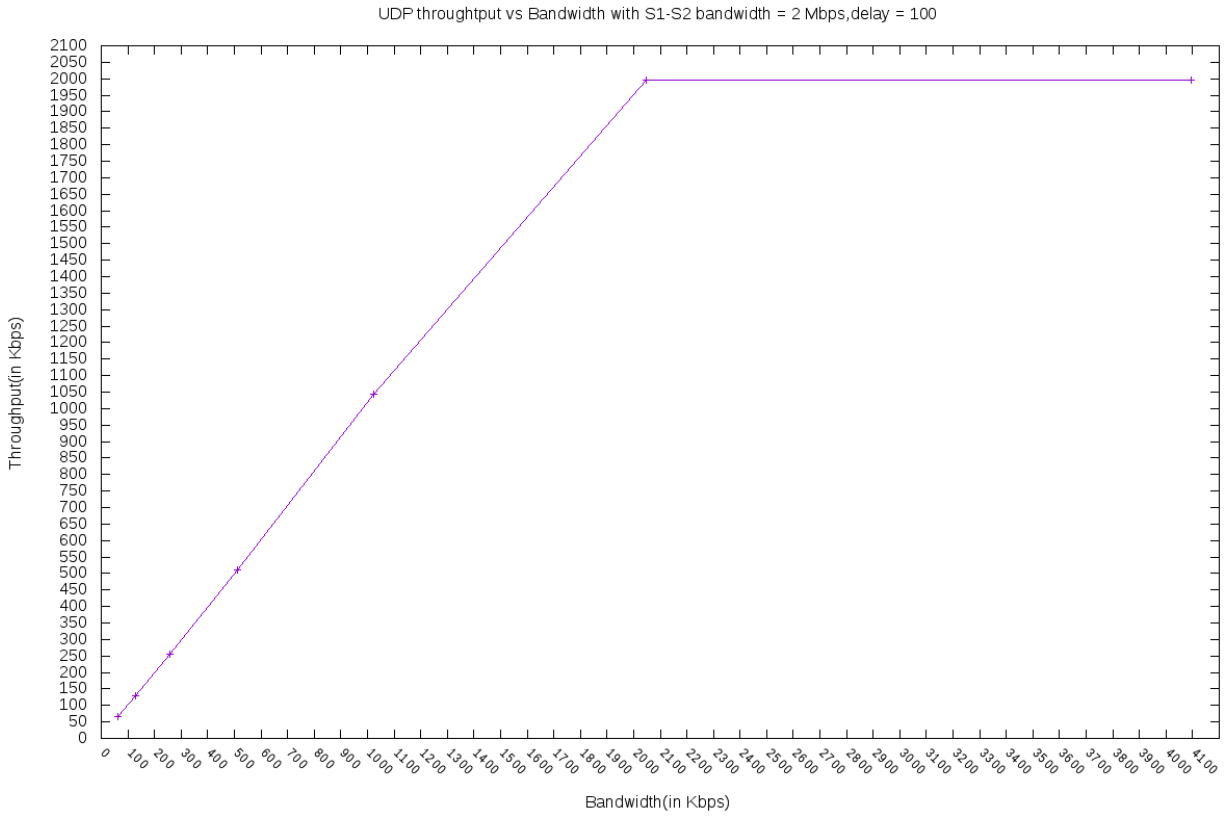

Plot9:UDP throughput vs Bandwidth with S1-S2 bandwidth=2Mbps,delay=10 ms

`




Plot10: UDP throughput vs Bandwidth with S1-S2 bandwidth=2Mbps,delay=100 ms

`

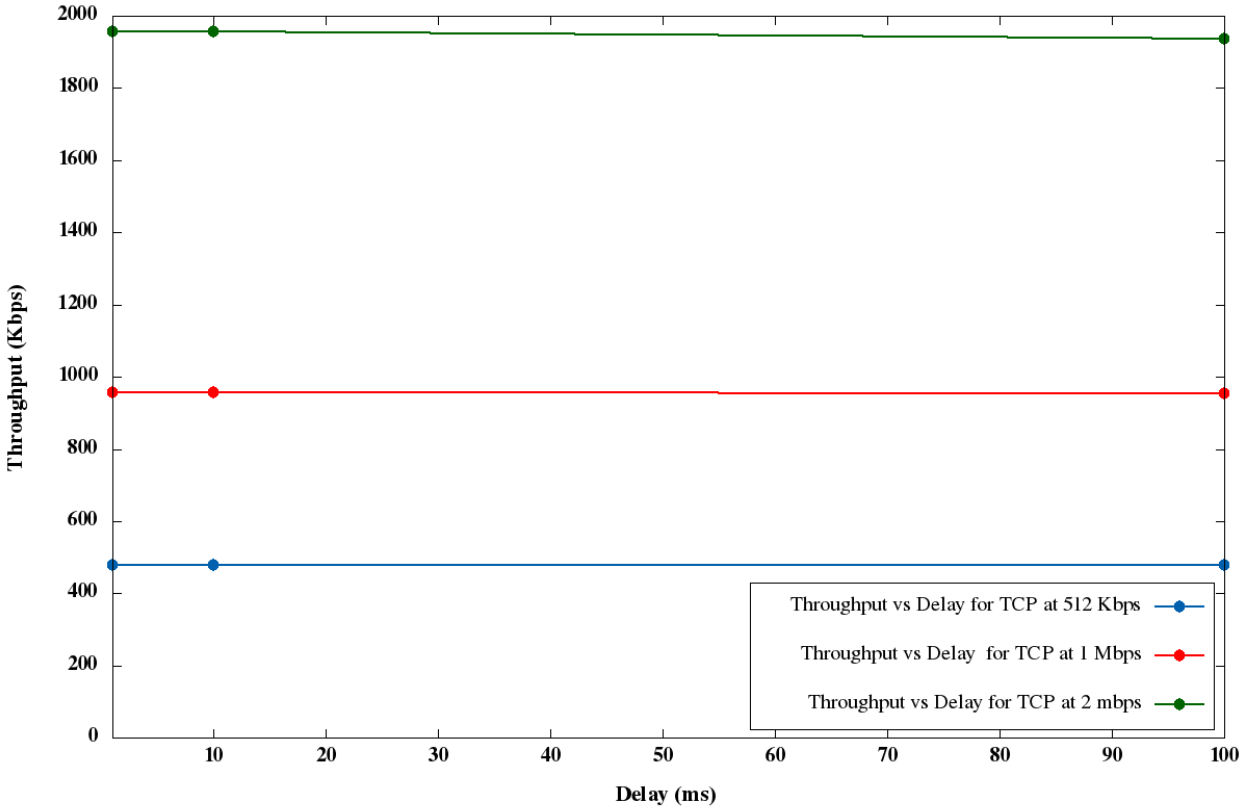



Some additional plots are :

For TCP:

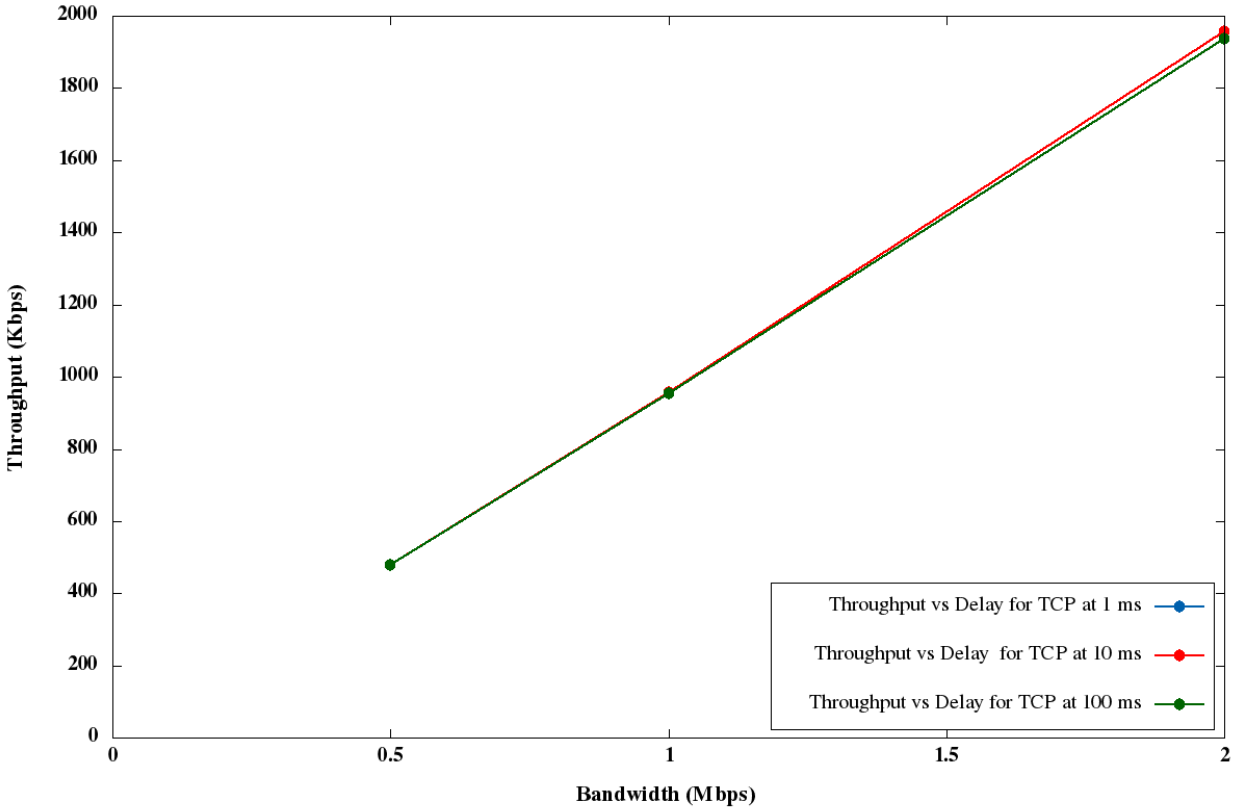
Plot 11 : Throughput vs delay for TCP at 512 Kbps, 1 Mbps, 2 Mbps

`




Plot 12 : Throughput vs bandwidth for TCP at delays 1ms,10ms and 100ms

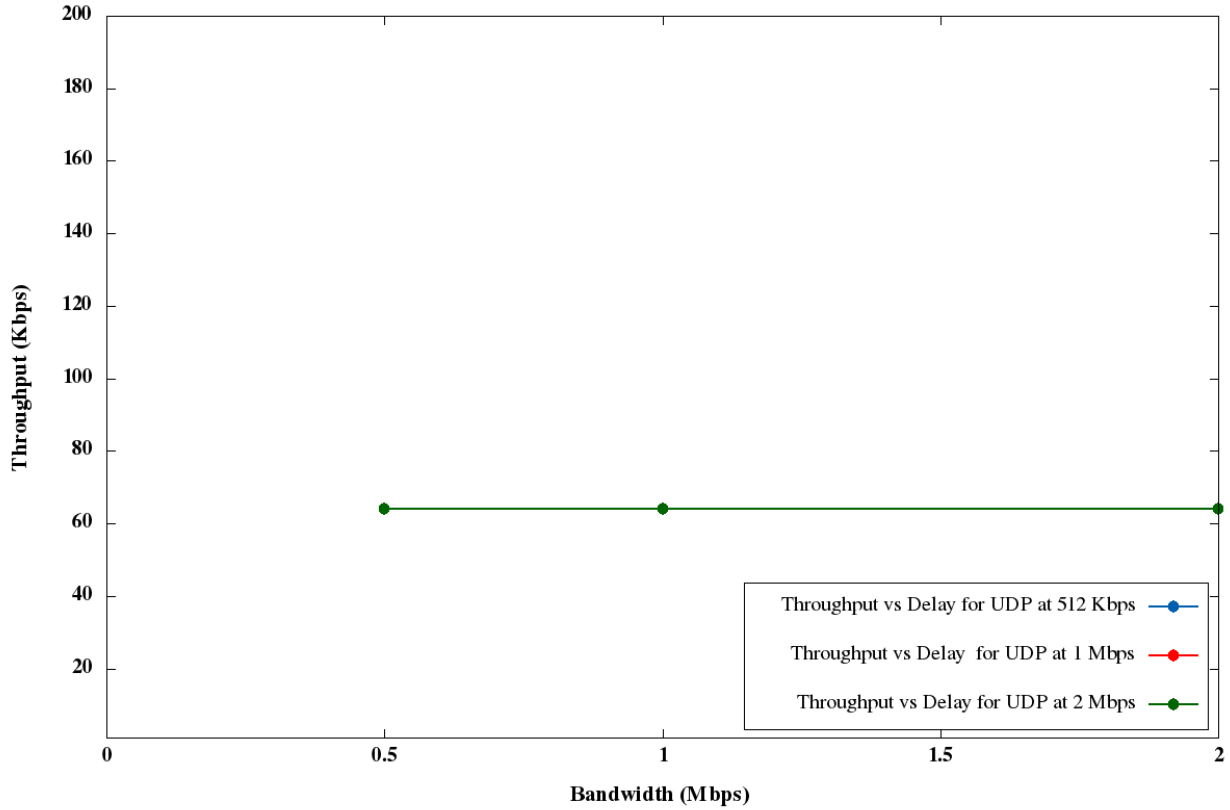
`




For UDP bandwidth = 64 Kbps

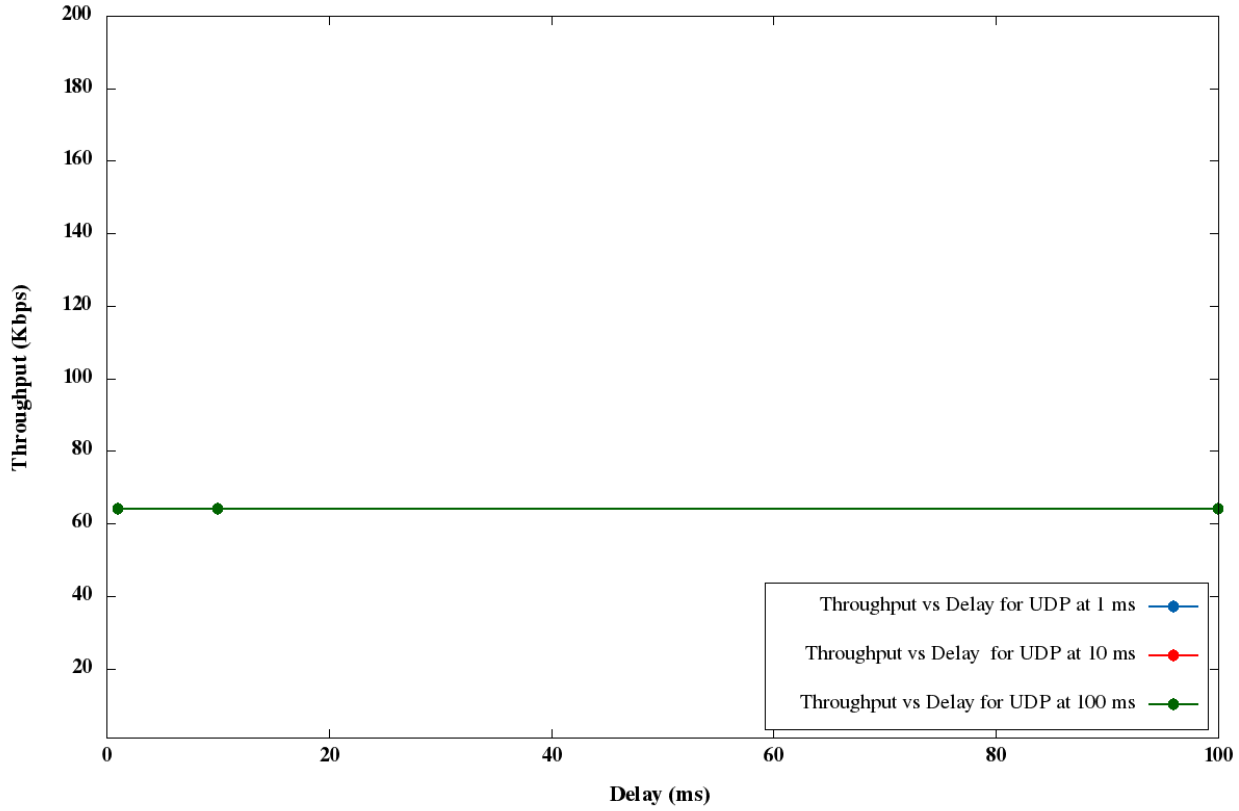
Plot 13 : Plot of UDP throughput vs bandwidth for delays 1ms , 10ms and 100ms

`




Plot 14 : Plot of UDP throughput vs delay for bandwidth 512 Kbps, 1 Mbps and 2 Mbps.

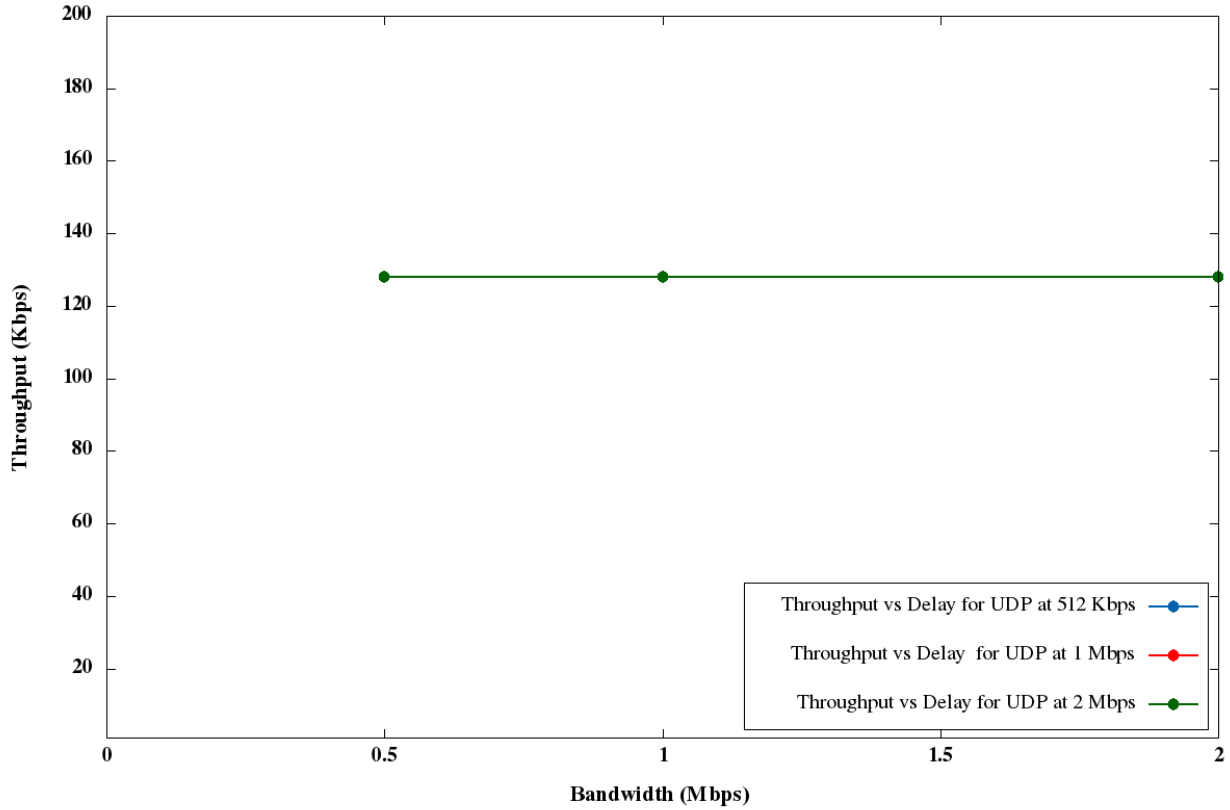
`




For UDP bandwidth = 128 Kbps

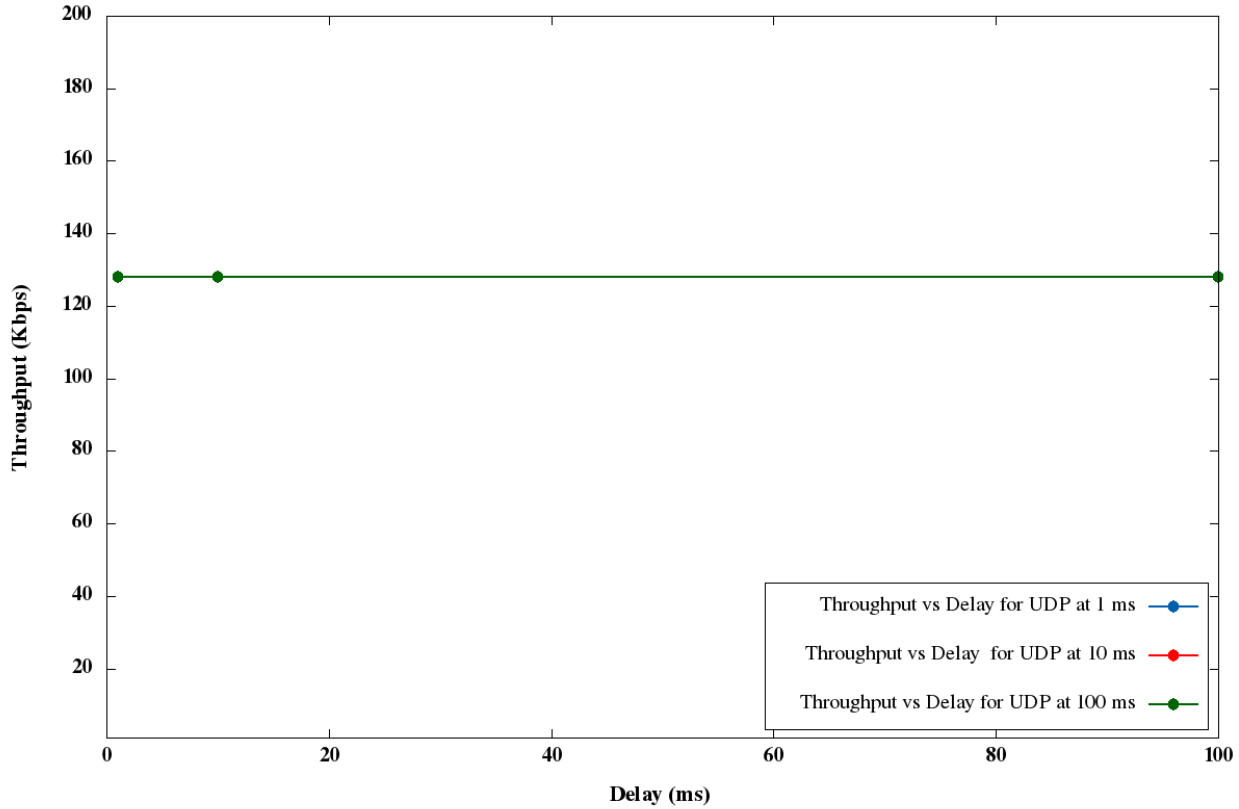
Plot 15 : Plot of UDP throughput vs bandwidth for delays 1ms, 10ms and 100ms

`




Plot 16 : Plot of UDP throughput vs delay for bandwidth 512 Kbps , 1 Mbps and 2 Mbps

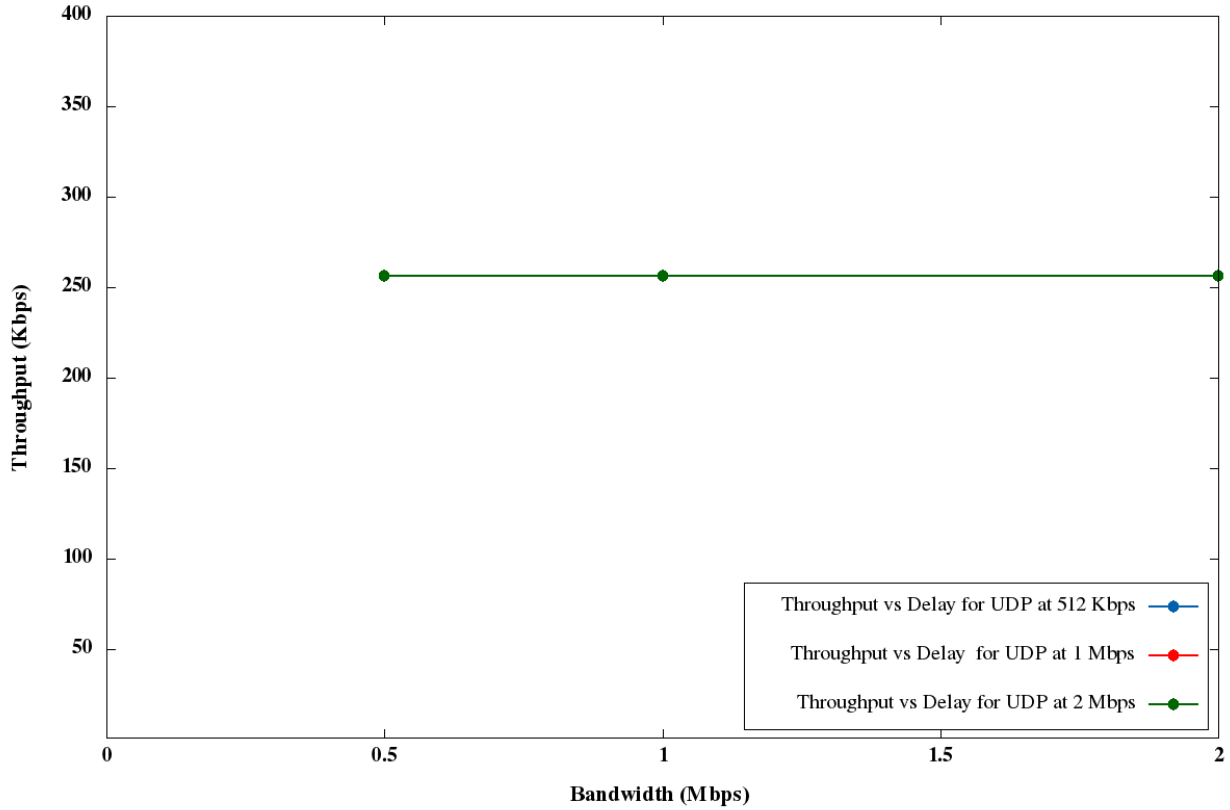
`




For UDP bandwidth = 256 Kbps

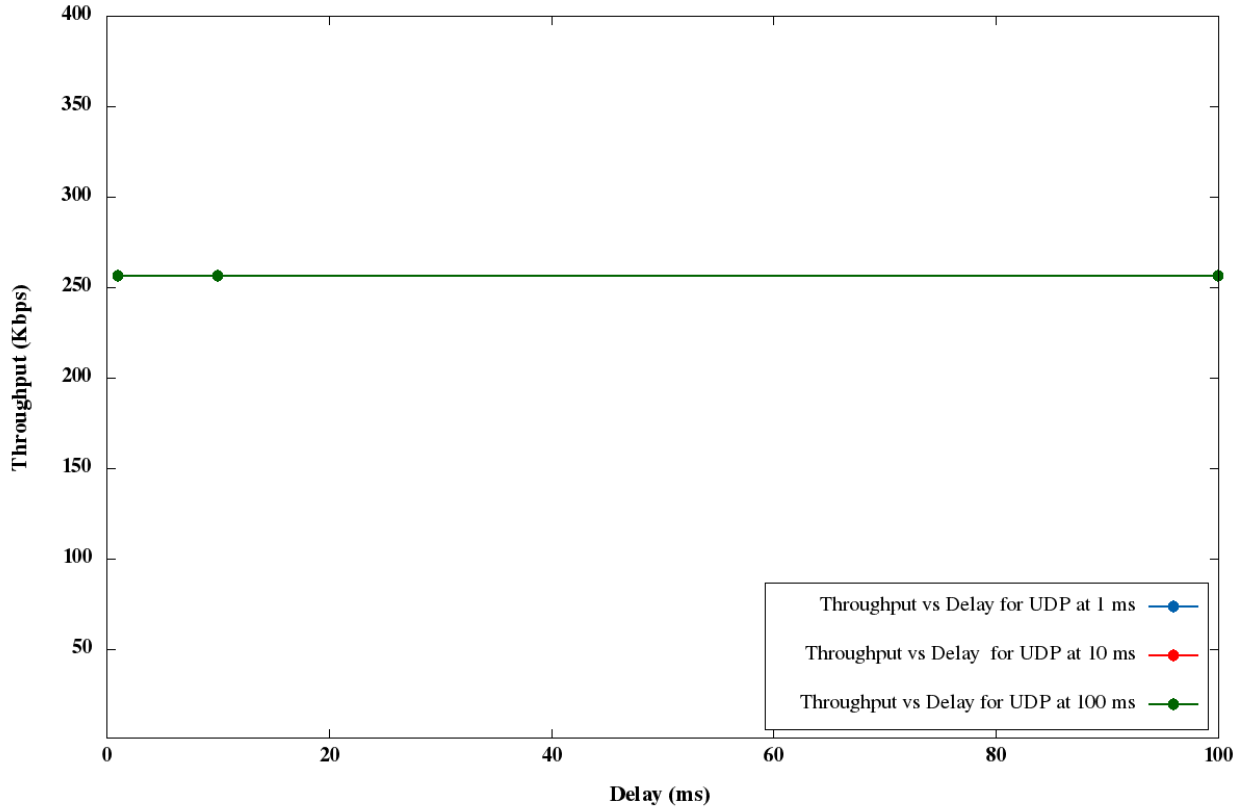
Plot 17 : Plot of UDP throughput vs bandwidth for delays 1ms,10ms and 100ms

`




Plot 18 : Plot of UDP throughput vs delay for bandwidth 512 Kbps,1 Mbps and 2Mbps

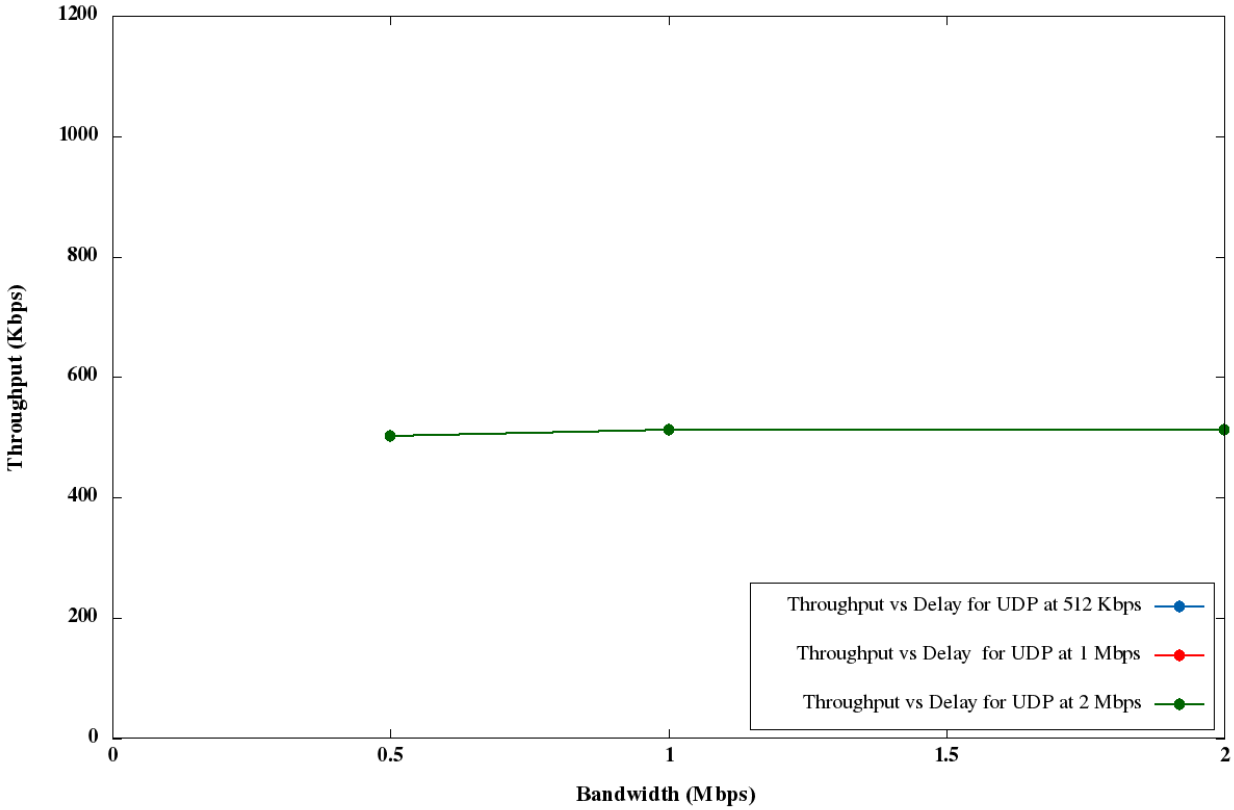
`




For UDP bandwidth = 512 Kbps

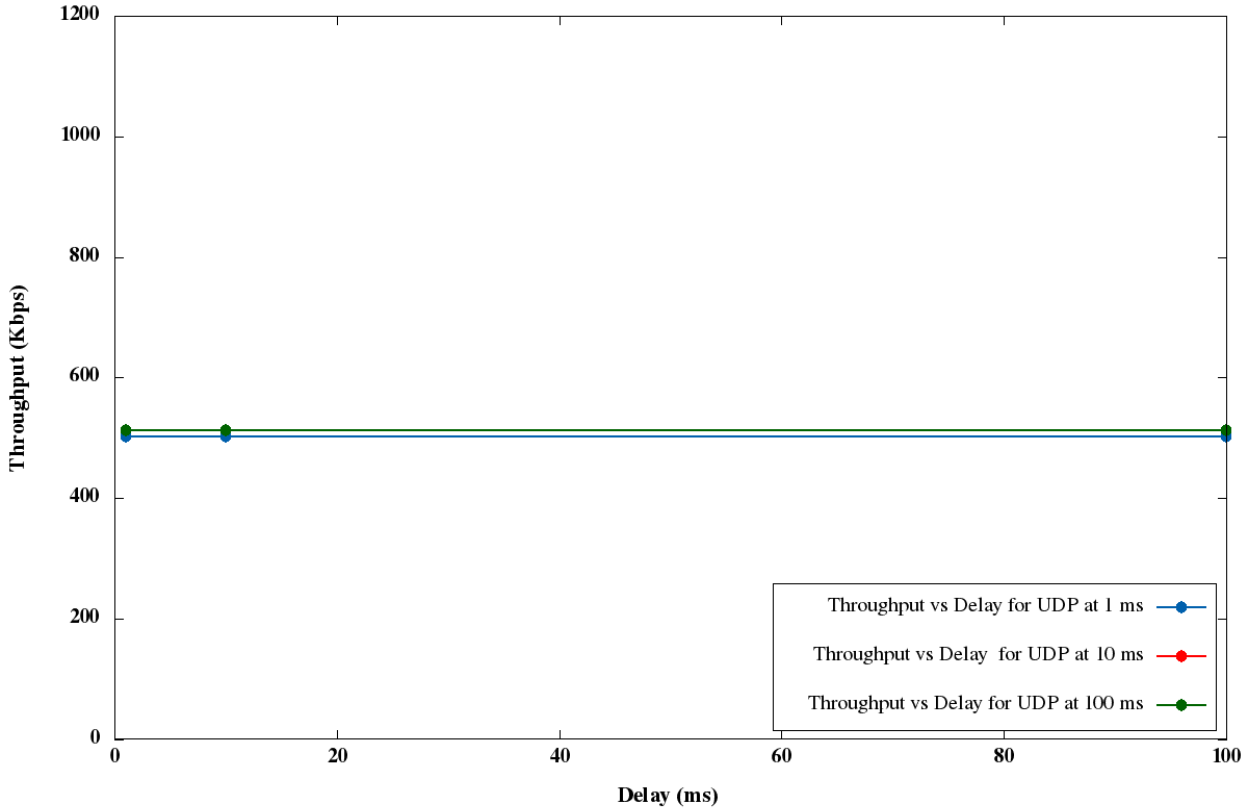
Plot 19 :

`


1200
1000
800
600
400
200
0
Throughput (Kbps)
0
0.5
1
1.5
2
Bandwidth (Mbps)
Throughput vs Delay for UDP at 512 Kbps
Throughput vs Delay for UDP at 1 Mbps
Throughput vs Delay for UDP at 2 Mbps

`

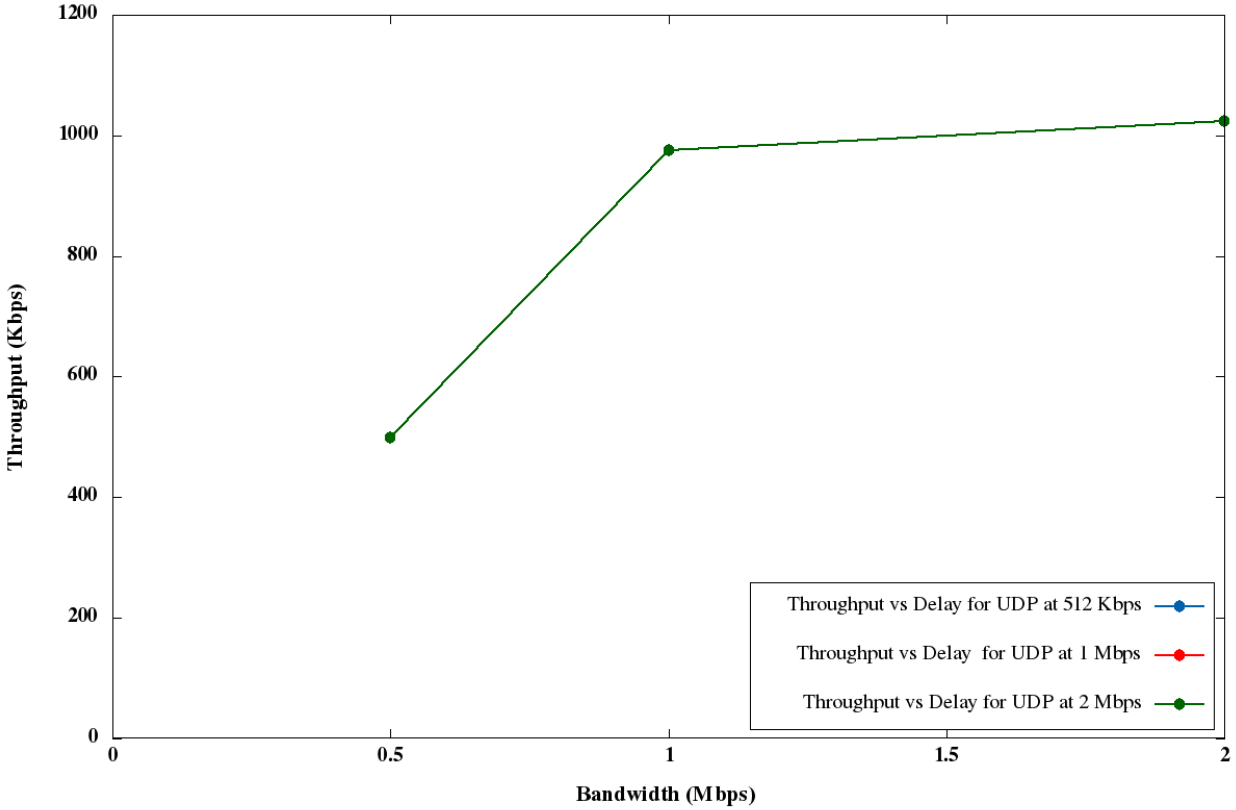
Plot 20 :


1200
1000
800
600
400
200
0
Throughput (Kbps)
0
20
40
60
80
100
Delay (ms)
Throughput vs Delay for UDP at 1 ms
Throughput vs Delay for UDP at 10 ms
Throughput vs Delay for UDP at 100 ms

`

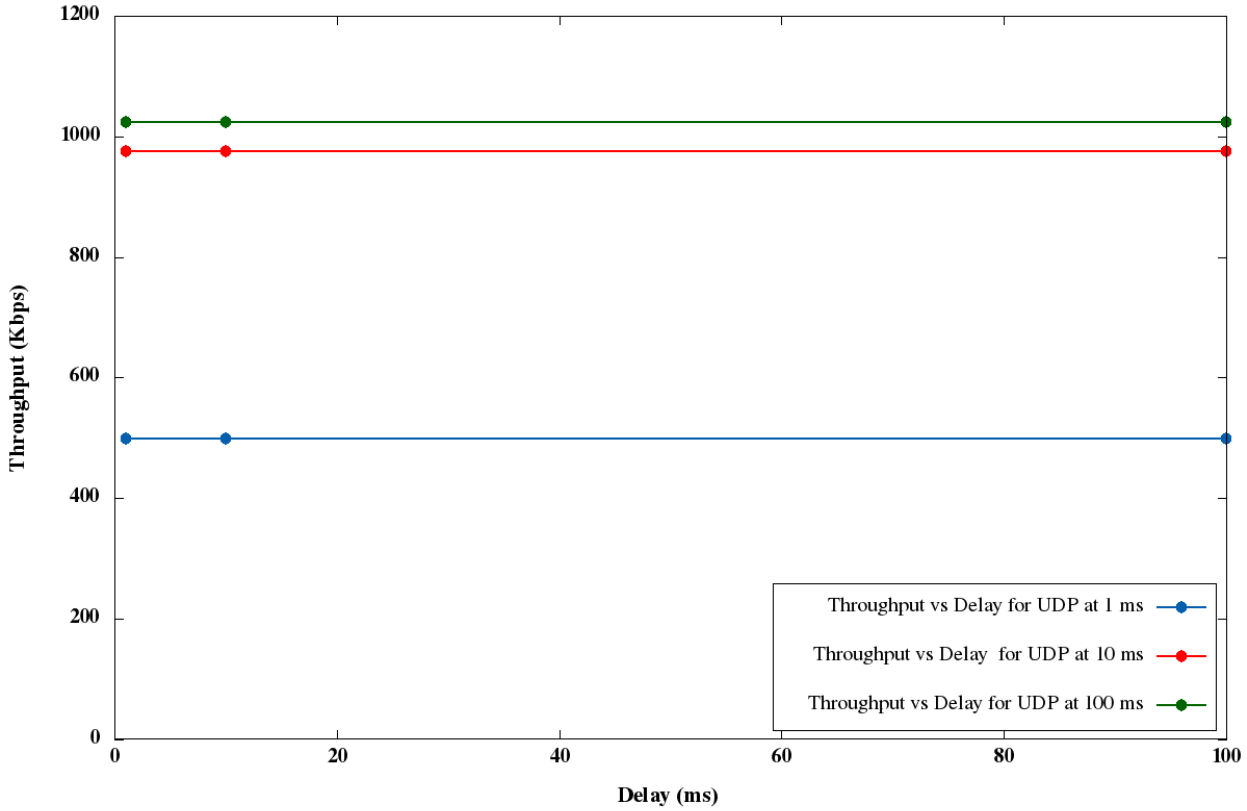
For UDP bandwidth = 1024 Kbps

Plot 21:



`

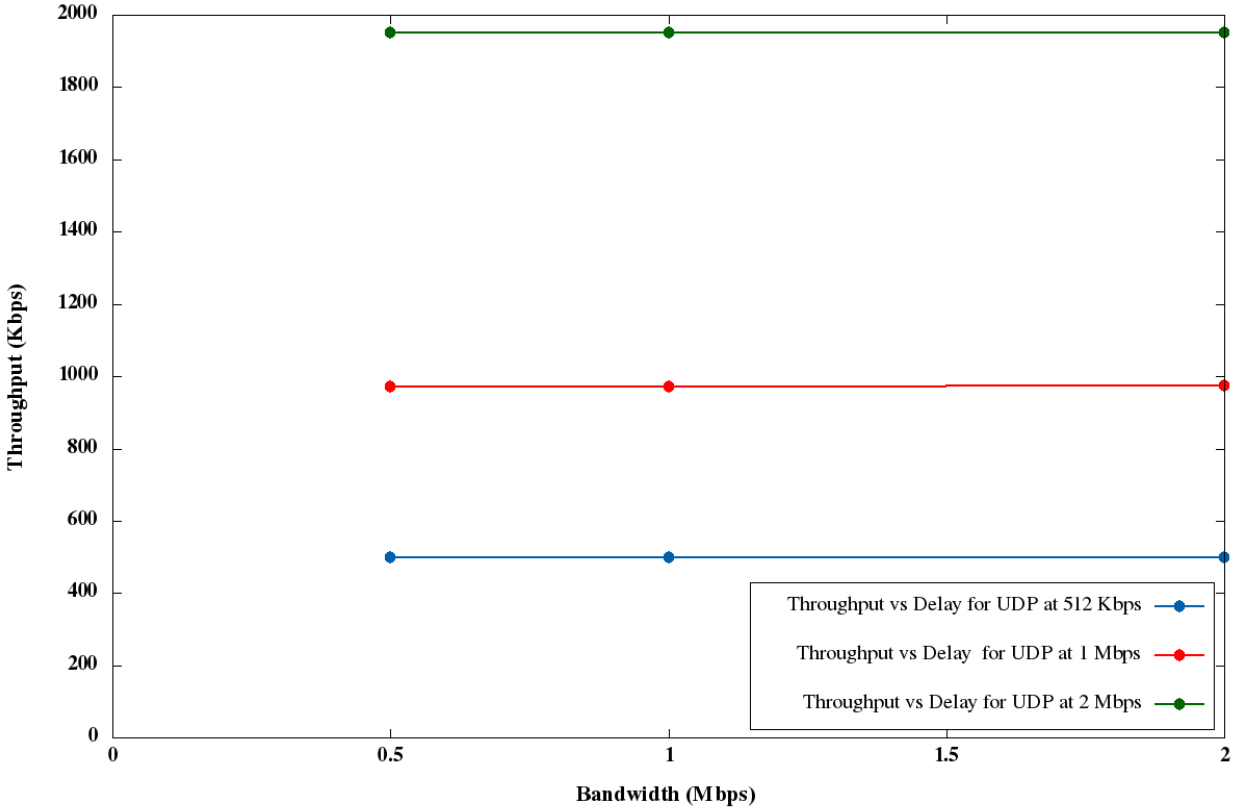
Plot 22:


1200
1000
800
600
400
200
0
Throughput (Kbps)
0
20
40
60
80
100
Delay (ms)
Throughput vs Delay for UDP at 1 ms
Throughput vs Delay for UDP at 10 ms
Throughput vs Delay for UDP at 100 ms

`

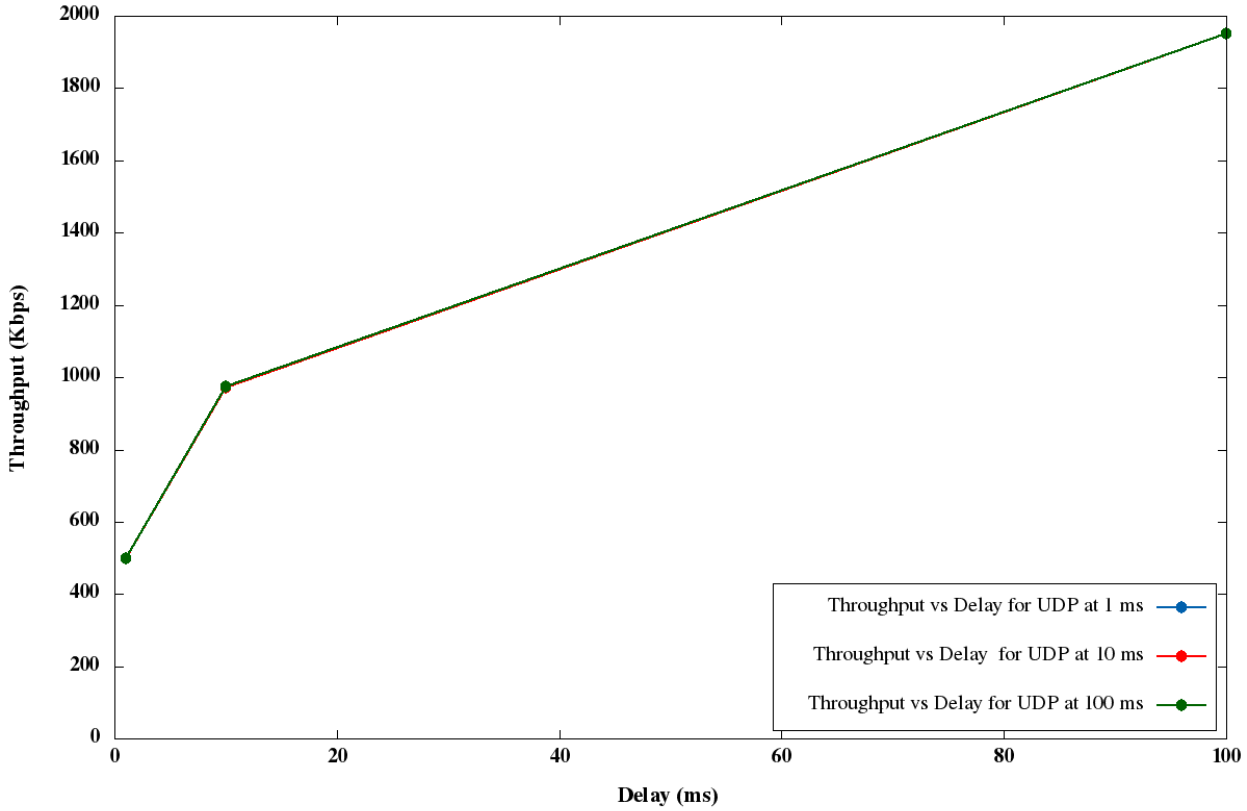
For UDP bandwidth = 2048 Kbps

Plot 23:



`

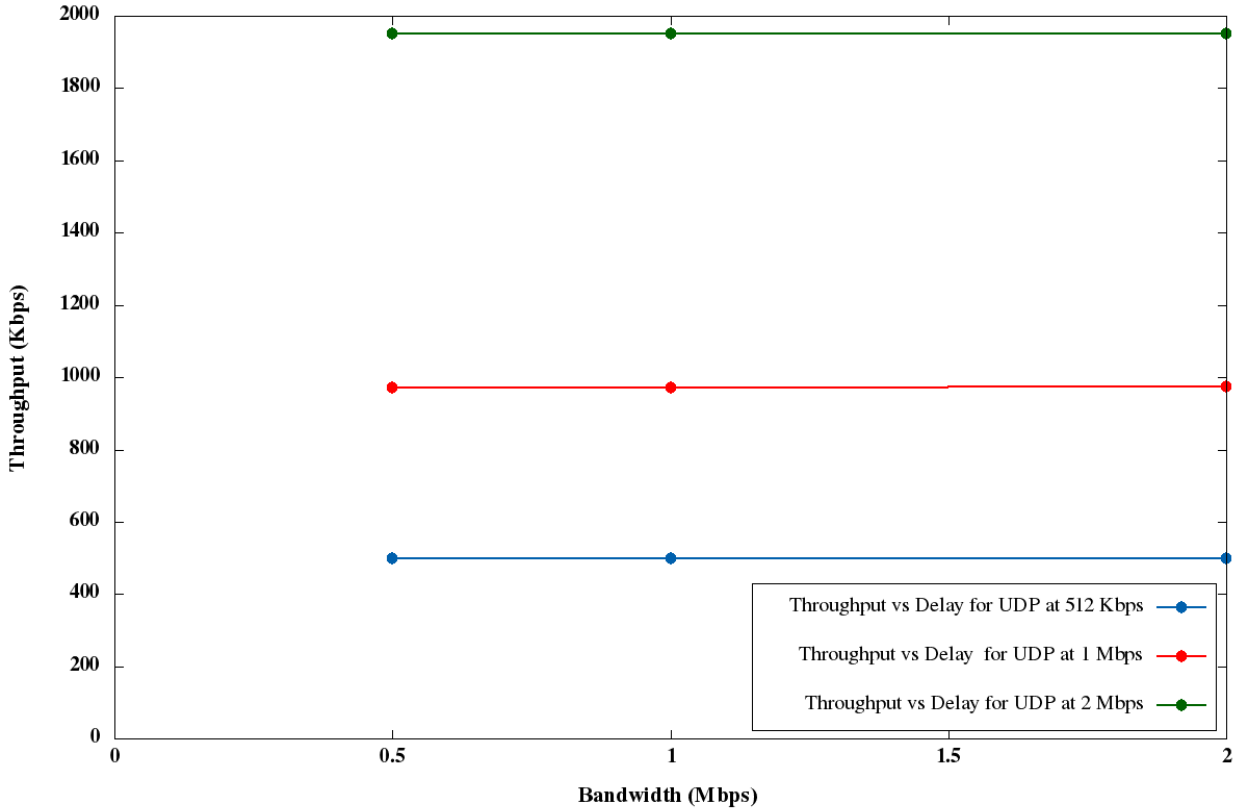
Plot 24:




For UDP bandwidth = 4096 Kbps

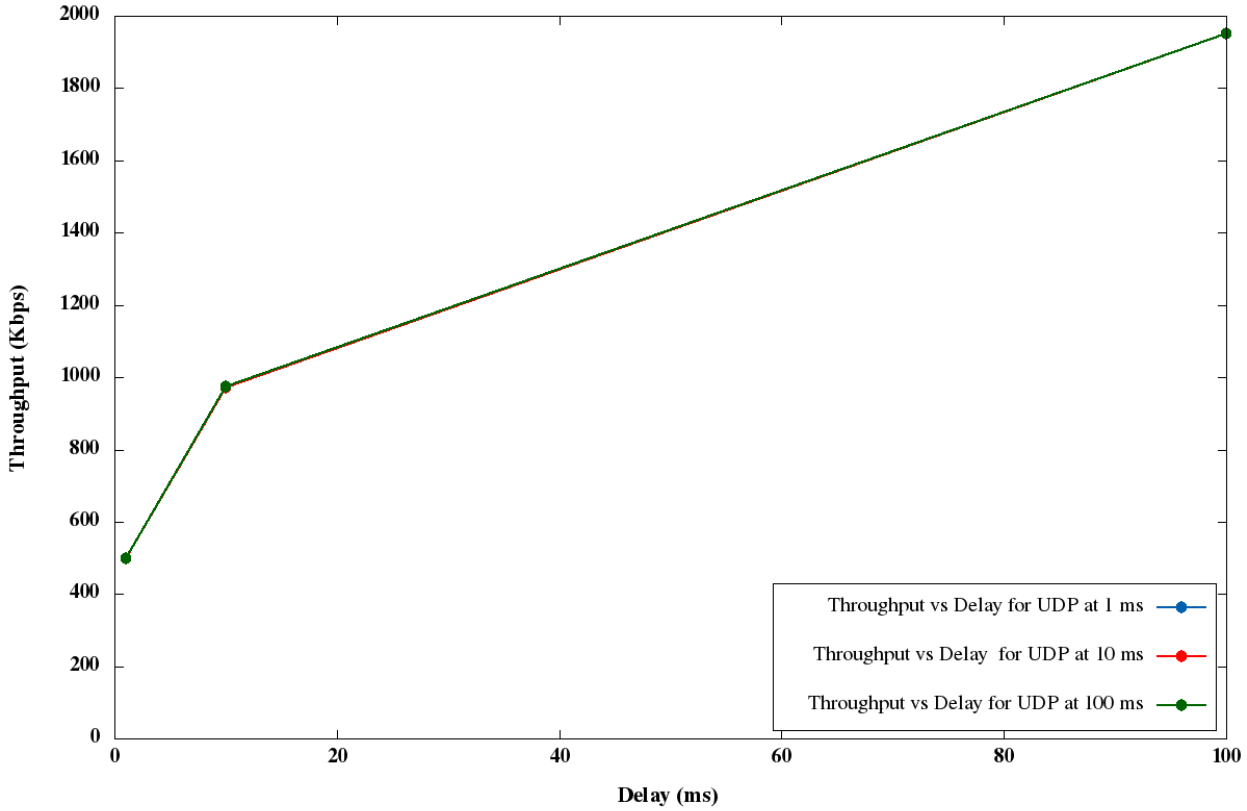
`

Plot 25:


2000
1800
1600
1400
1200
1000
800
600
400
200
0
Throughput (Kbps)
0
0.5
1
1.5
2
Bandwidth (Mbps)
Throughput vs Delay for UDP at 512 Kbps
Throughput vs Delay for UDP at 1 Mbps
Throughput vs Delay for UDP at 2 Mbps

`

Plot 26:




Justification:

1. Effect of S1-S2 link bandwidth on TCP and UDP throughput

   From the graph, we see the effect of changing the S1-S2 link bandwidth on the TCP and UDP throughput.

   On increasing the S1-S2 link bandwidth the we allow more amount of data to pass through and thus, the throughput increases. Thus, TCP and UDP throughput is directly proportional to the link bandwidth of S1-S2.

2. Effect of S1-S2 link delay on TCP and UDP throughput
   From the graph, we see that the throughput remains constant with very slight decrease even after changing the delay. The decrease is very insignificant.
   The amount of data passed does not get changed much after changing the delay, hence the throughput remains nearly constant and decreases slightly in some cases.
   Due to increase in delay, some packets may not be acknowledged and hence throughput might decrease in case delay is increased largely.

`

Part 3:

Steps:

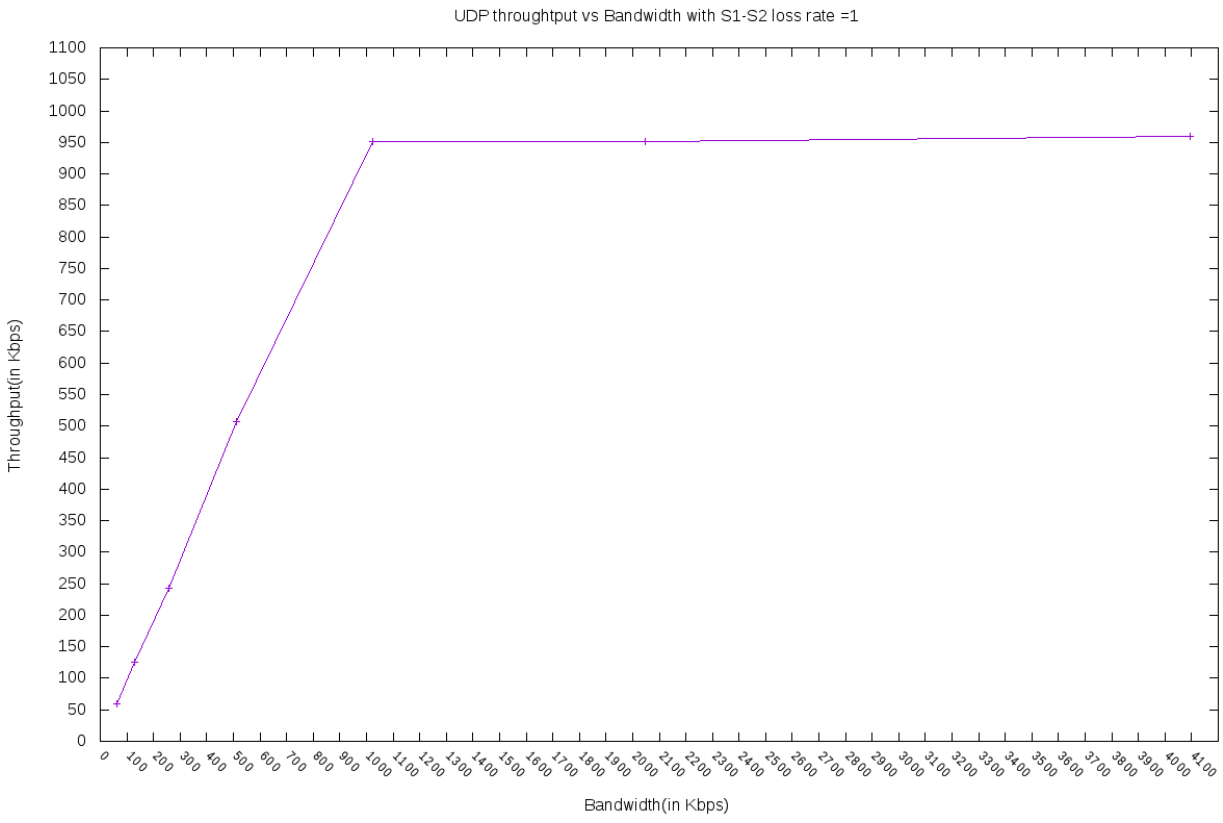
To create a network with custom configuration we use the command **sudo mn –custom ~/mininet/custom/mytopo2.py –topo mytopo –link tc**

The file mytopo2.py is given in the folder. The parameters are changed every time and we need to destroy and recreate the network for the changes to be effected.

We take the readings similar to part 1.

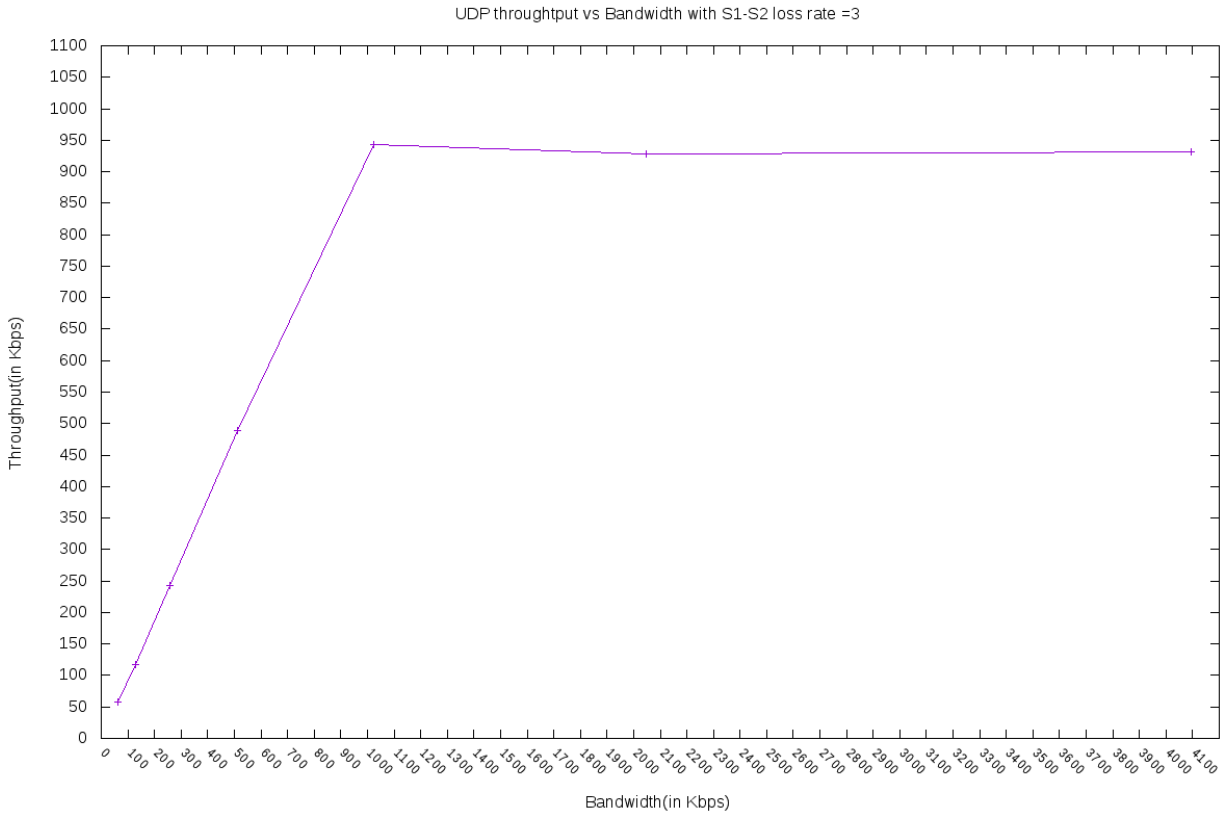
Observation:

Plot 27 :

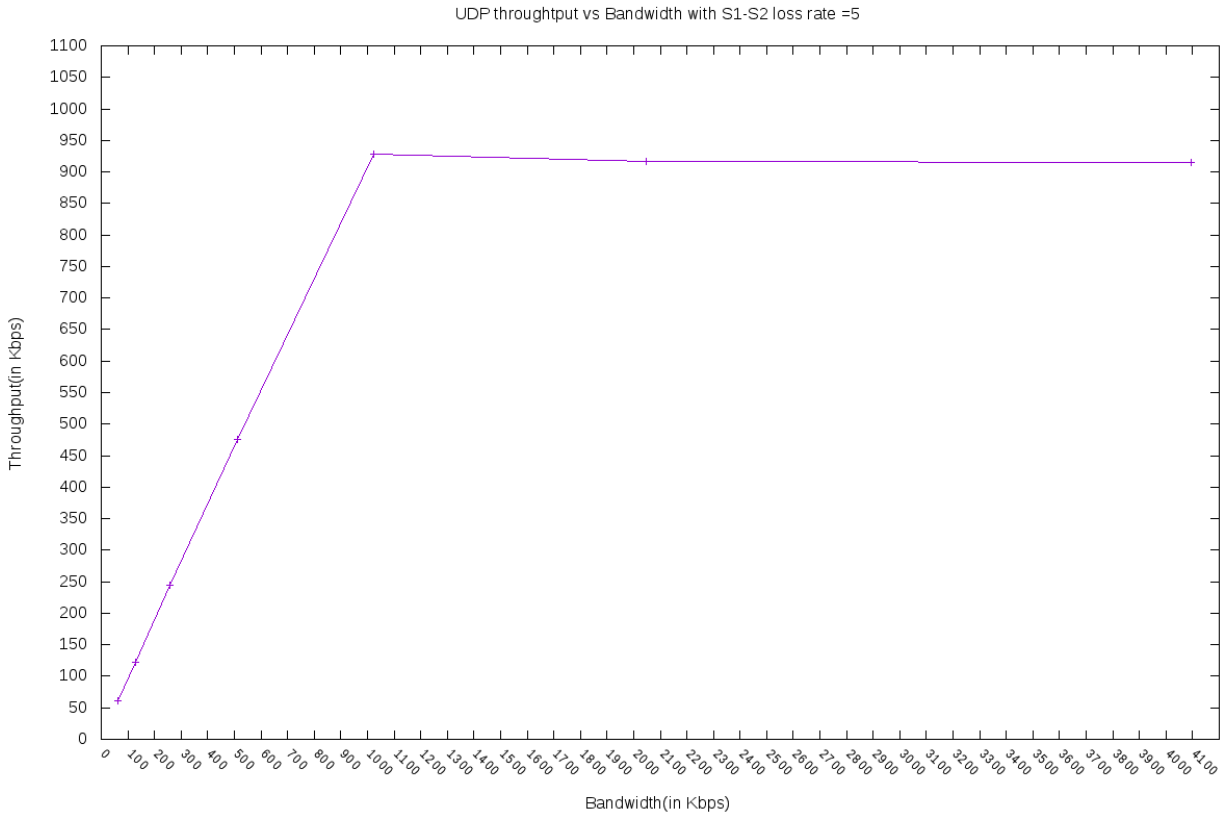



Plot 28:

`

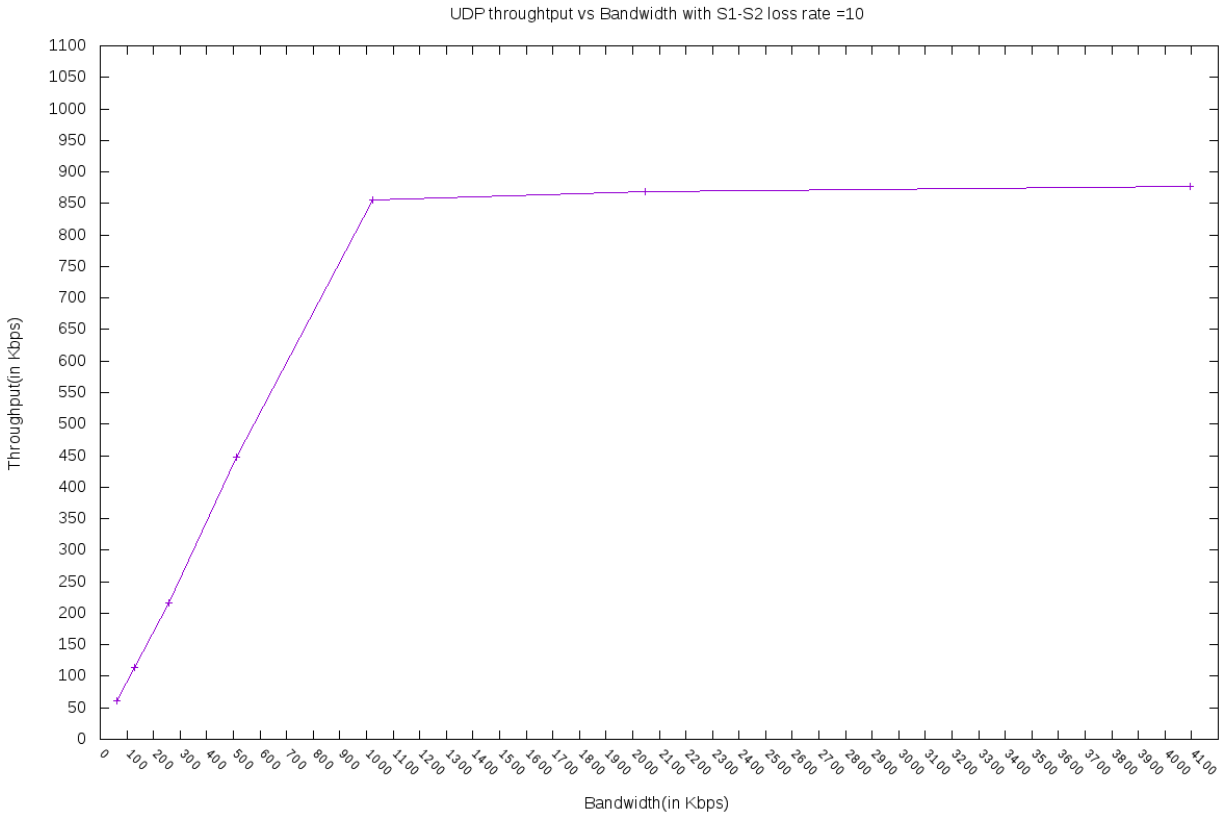

UDP throughtput vs Bandwidth with S1-S2 loss rate =3
Throughput(in Kbps)
Bandwidth(in Kbps)

Plot 29:


UDP throughtput vs Bandwidth with S1-S2 loss rate =5
Throughput(in Kbps)
Bandwidth(in Kbps)

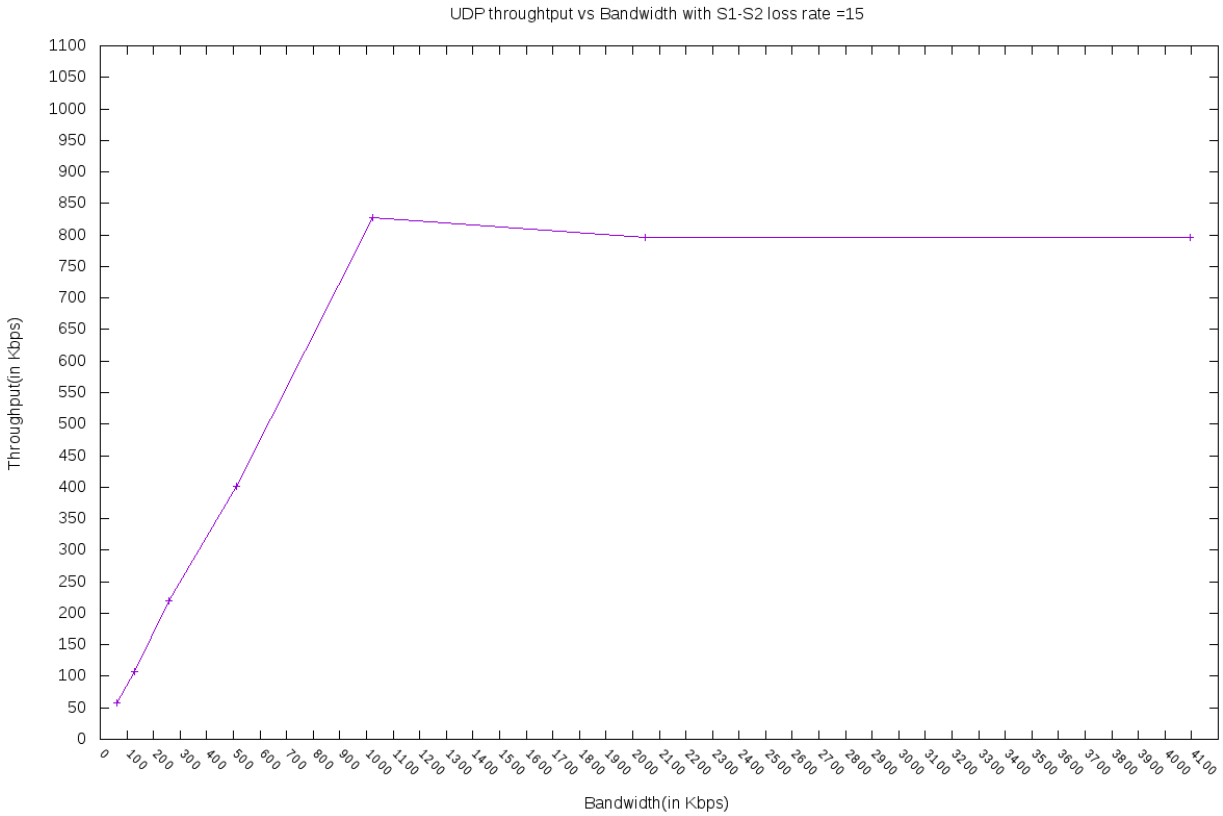
`

Plot 30:


UDP throughtput vs Bandwidth with S1-S2 loss rate =10
Throughput(in Kbps)
Bandwidth(in Kbps)
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
1050
1100
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100

`

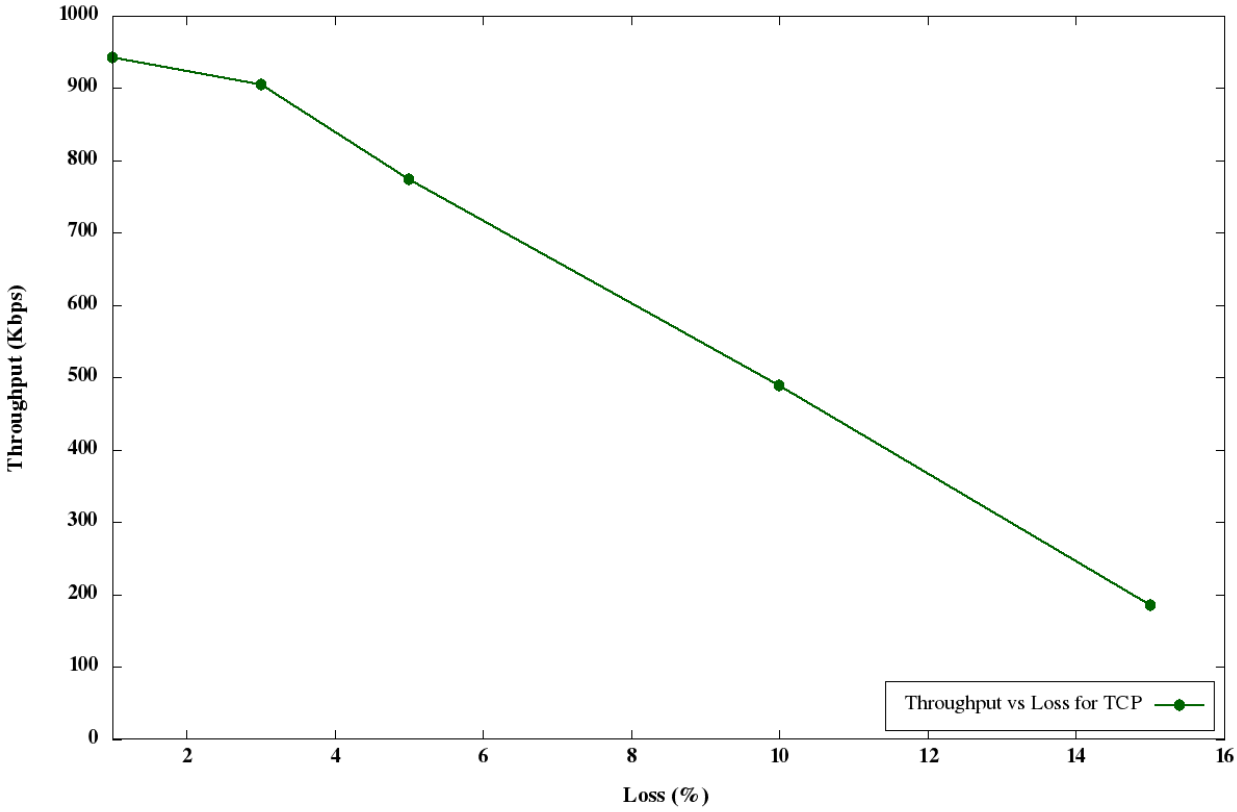
Plot 31 :


UDP throughtput vs Bandwidth with S1-S2 loss rate =15
Throughput(in Kbps)
Bandwidth(in Kbps)
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
1050
1100
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
3600
3700
3800
3900
4000
4100

`

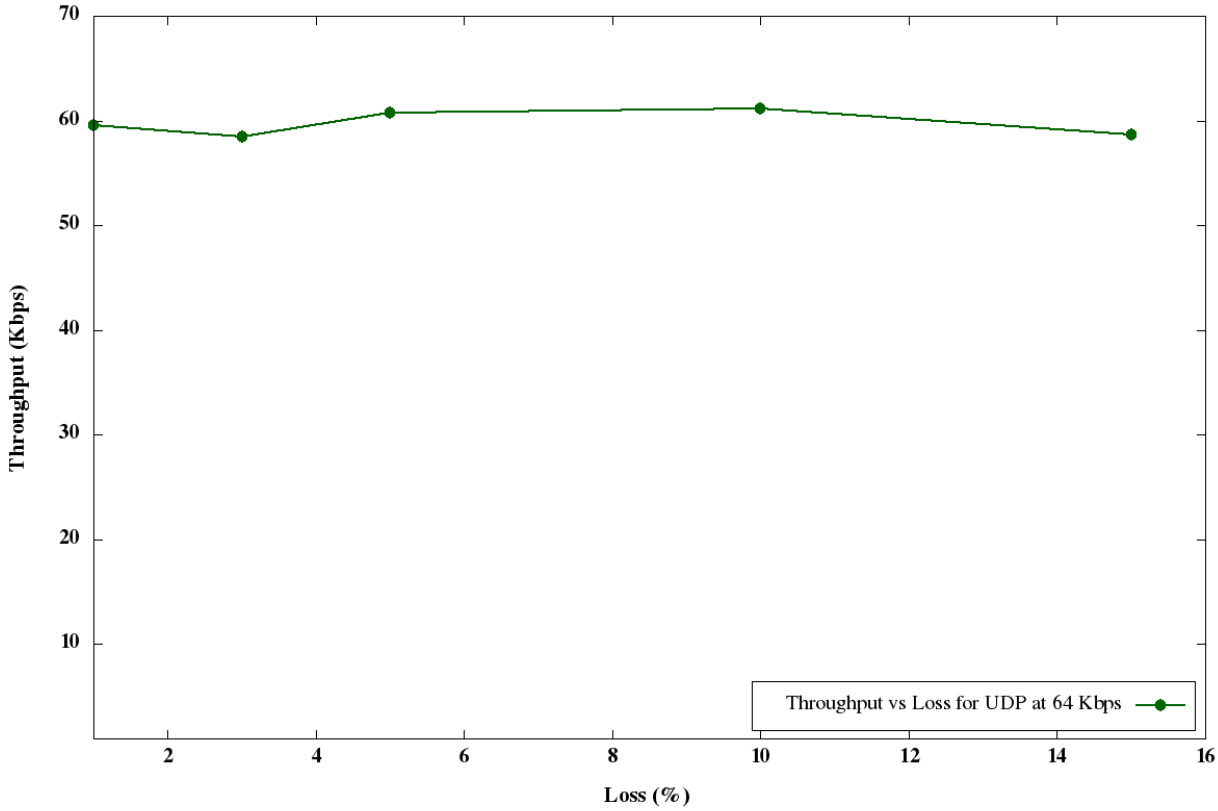
Other graphs :

Plot 32:



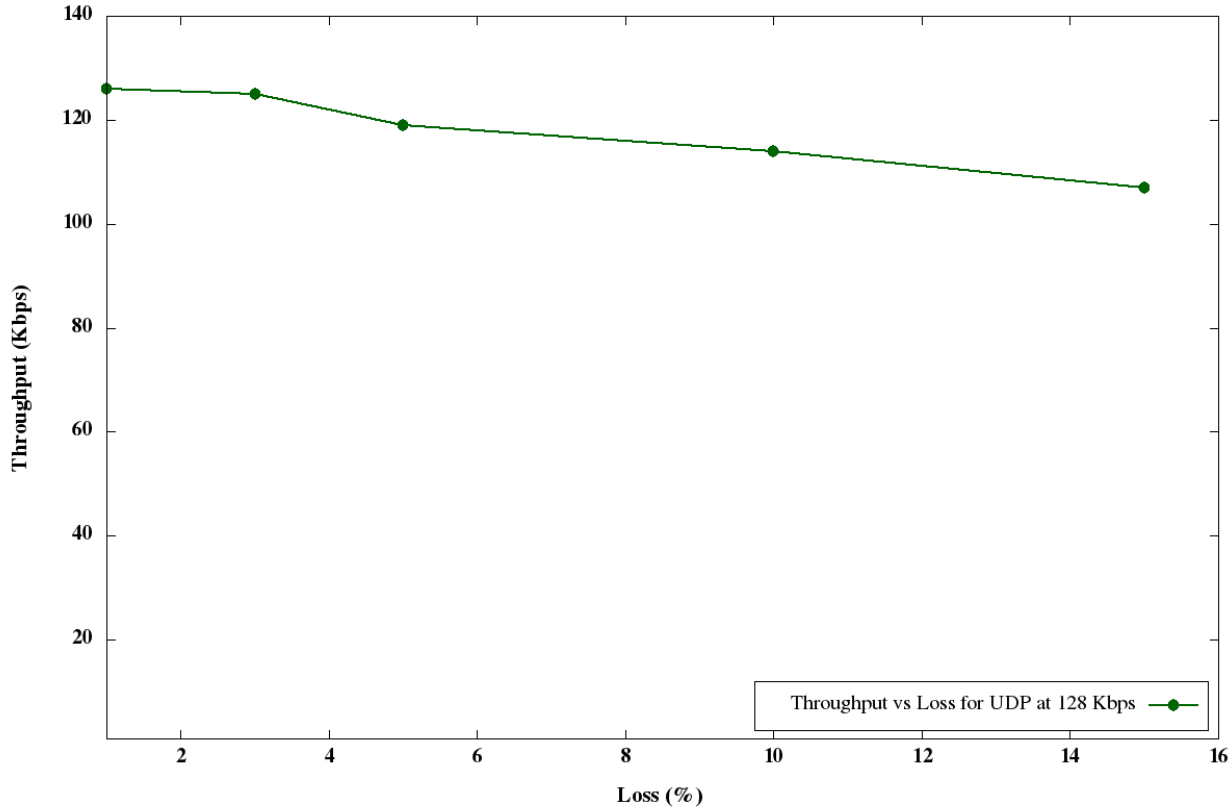
`

Plot 33:


70
60
50
40
30
20
10
Throughput (Kbps)
2
4
6
8
10
12
14
16
Loss (%)
Throughput vs Loss for UDP at 64 Kbps

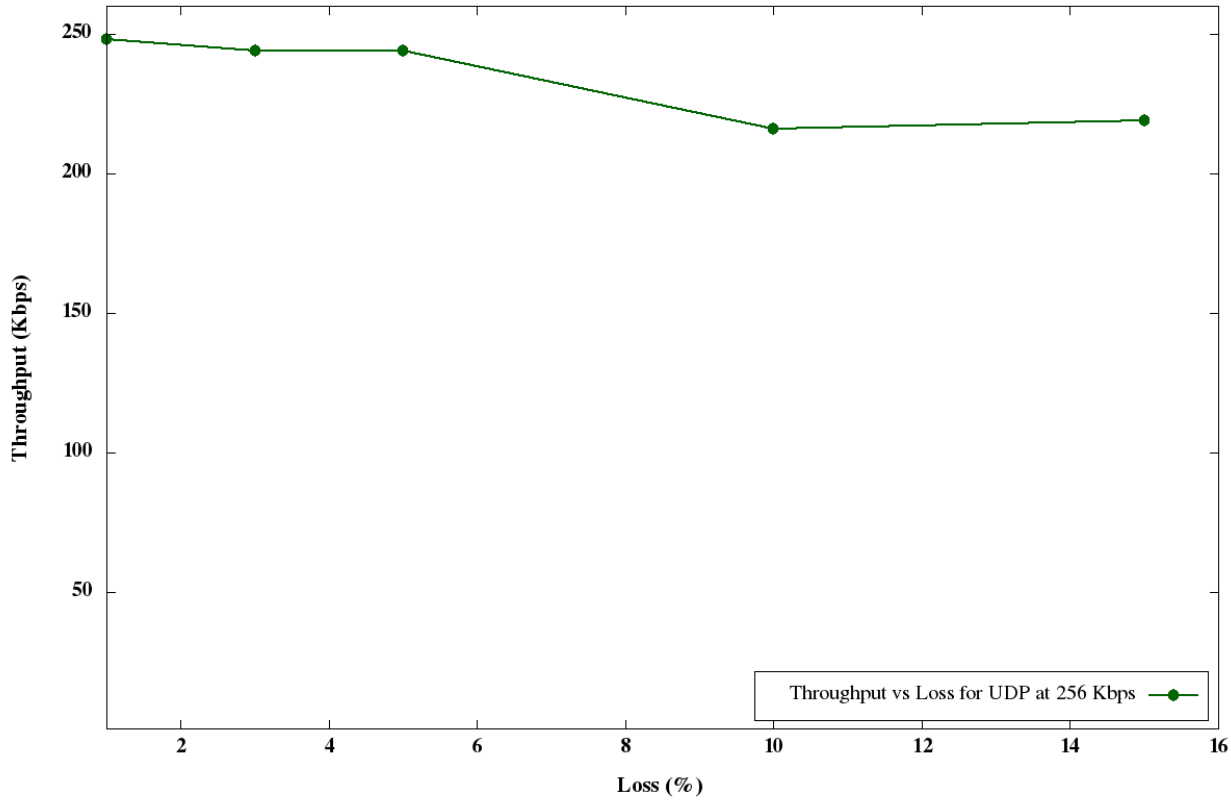
`

Plot 34:


140
120
100
80
60
40
20
Throughput (Kbps)
2
4
6
8
10
12
14
16
Loss (%)
Throughput vs Loss for UDP at 128 Kbps

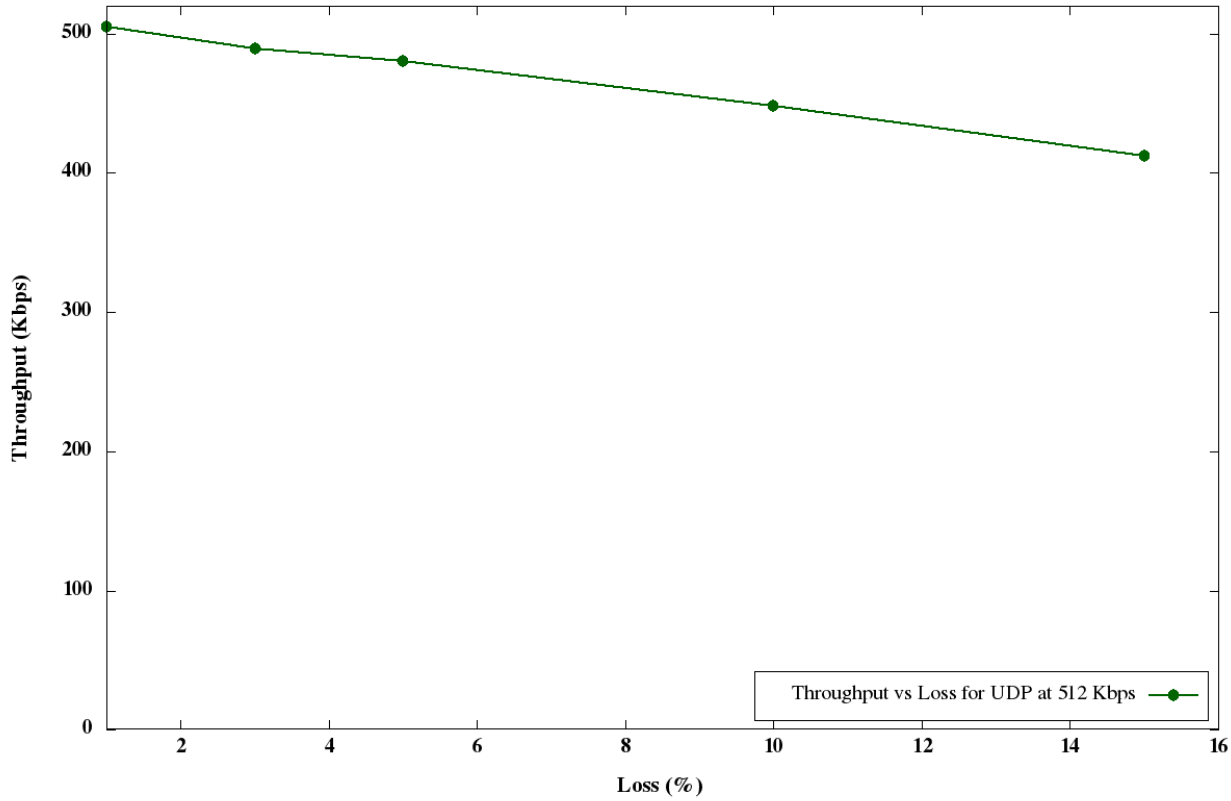
`

Plot 35:


250
200
150
100
50
Throughput (Kbps)
2
4
6
8
10
12
14
16
Loss (%)
Throughput vs Loss for UDP at 256 Kbps

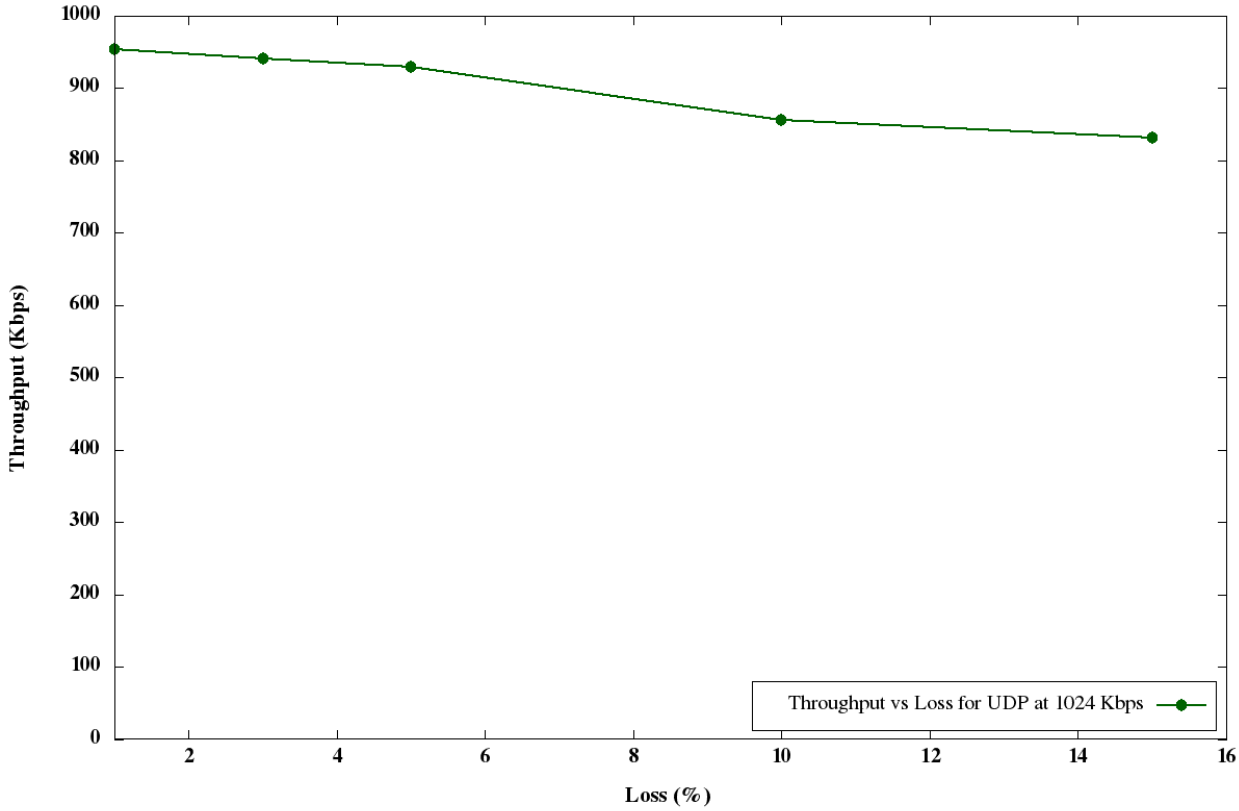
`

Plot 36:


Throughput (Kbps)
500
400
300
200
100
0
2
4
6
8
10
12
14
16
Loss (%)
Throughput vs Loss for UDP at 512 Kbps

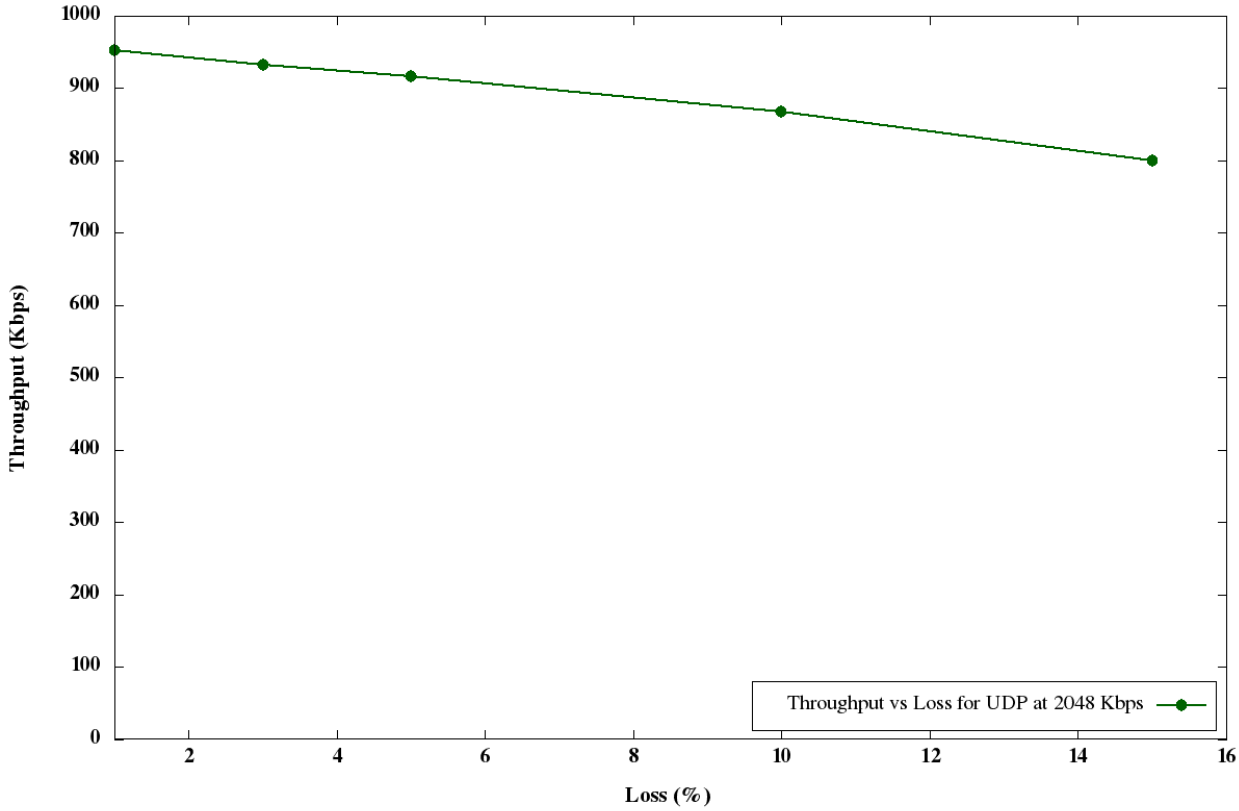
`

Plot 37:


1000
900
800
700
600
500
400
300
200
100
0
Throughput (Kbps)
2
4
6
8
10
12
14
16
Loss (%)
Throughput vs Loss for UDP at 1024 Kbps

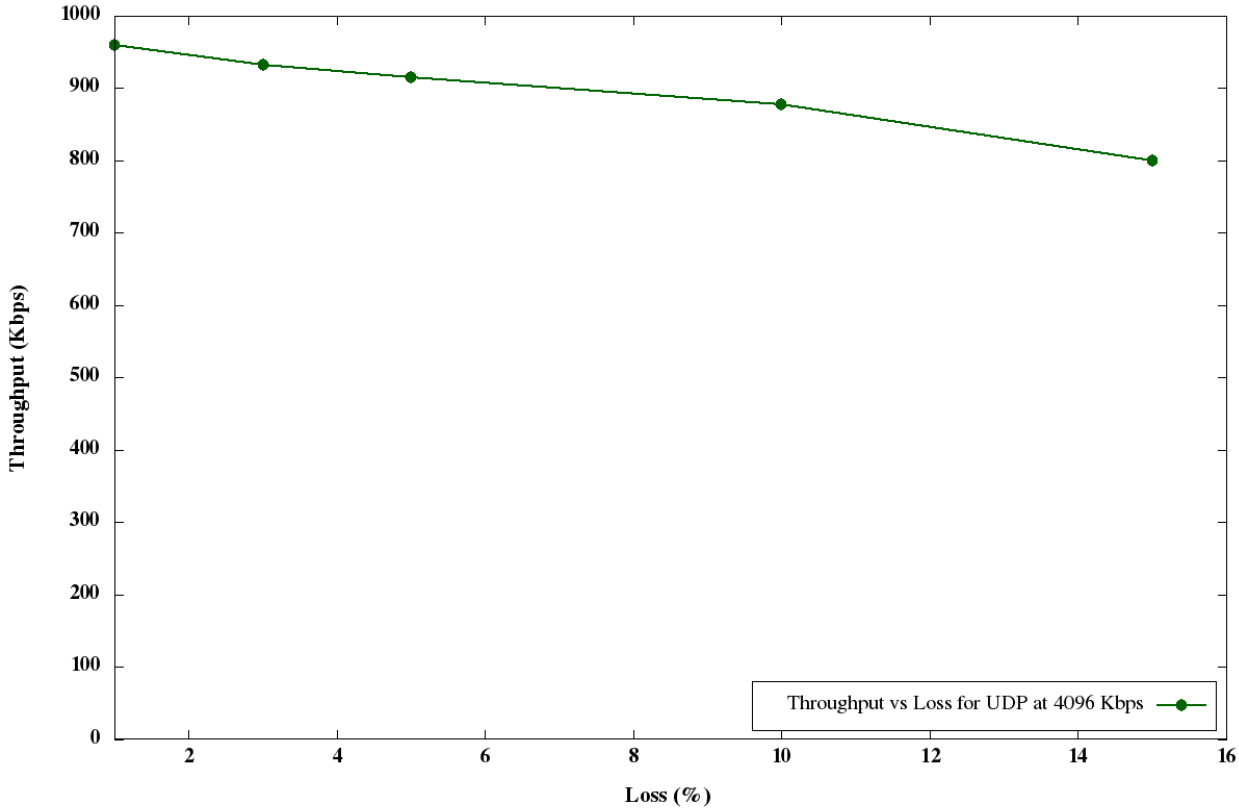
`

Plot 38:


1000
900
800
700
600
500
400
300
200
100
0
Throughput (Kbps)
2
4
6
8
10
12
14
16
Loss (%)
Throughput vs Loss for UDP at 2048 Kbps

`

Plot 39:




Justification:

On increasing the loss, the throughput decreases eventually in both TCP and UDP.
This is because, on increasing the loss percentage more number of packets get lost and are eventually not transferred to the target. This lesser number of packets reaching the target decreases throughput. Thus delay is inversely proportional to throughput.