Networks Assignment 5

Submitted By:
Mainak Sethi (11010134)
Vivek Bhargav (11010172)
Chaitanya Agarwal (11010115)

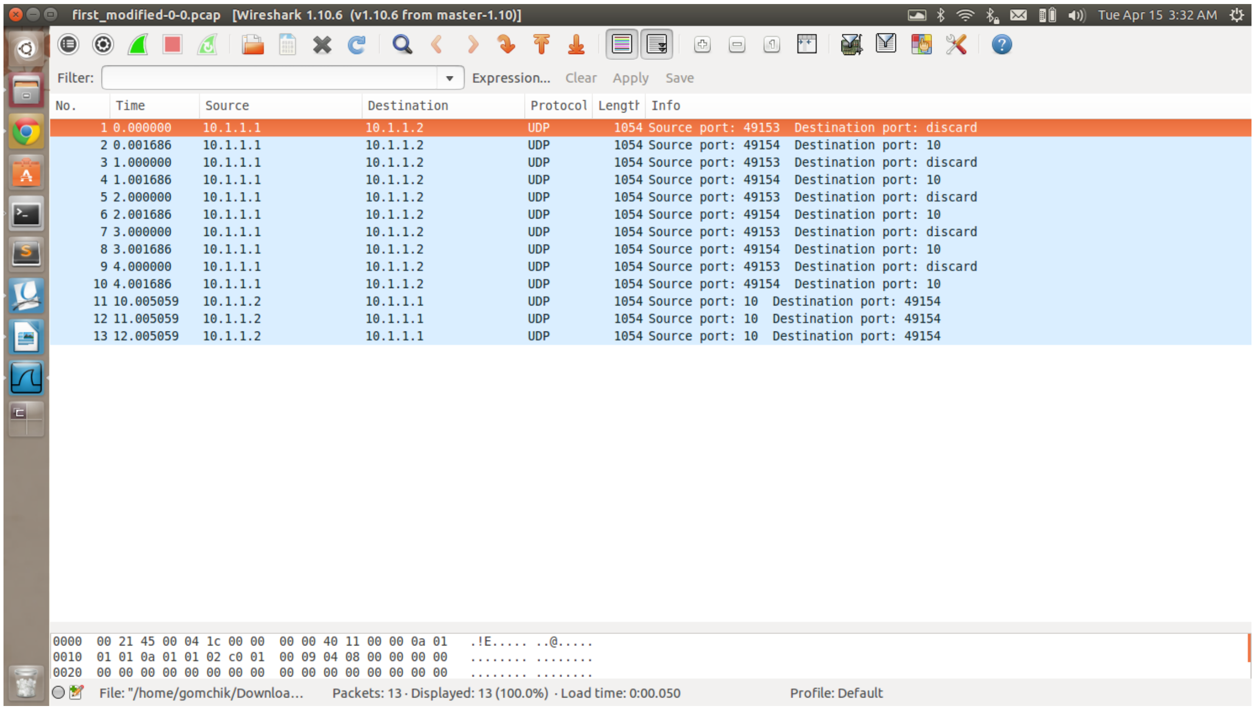
Question 1:

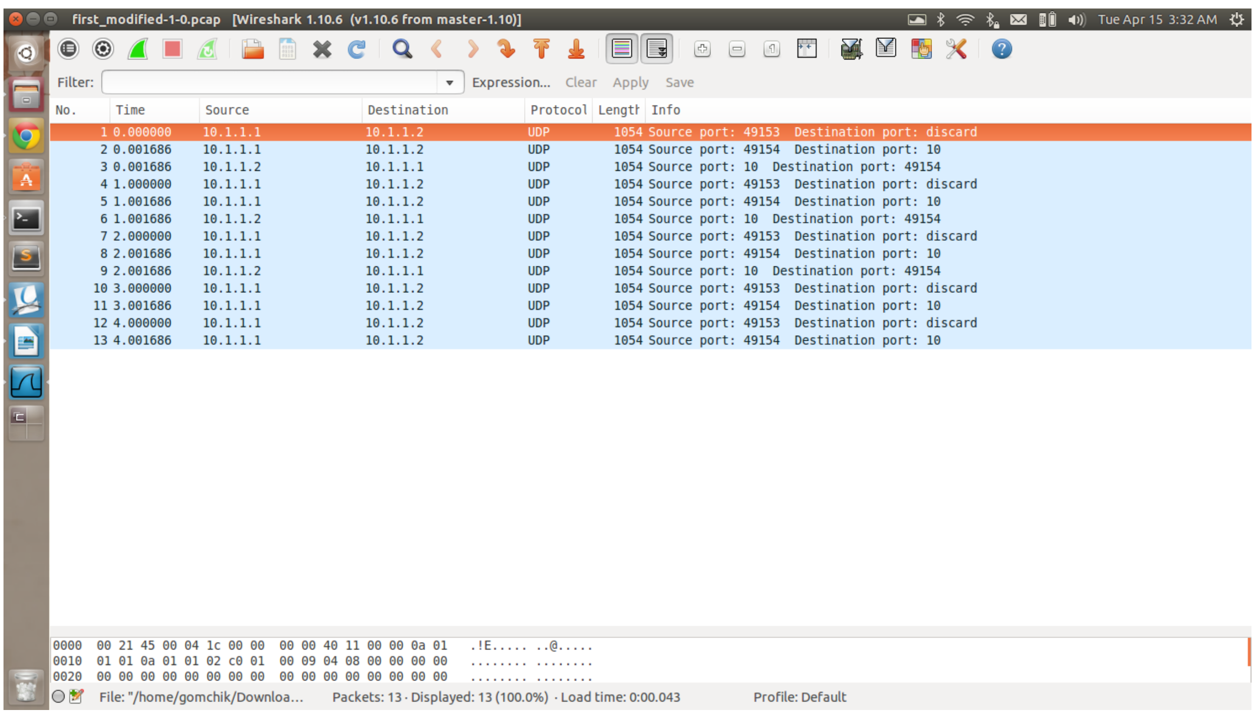
[Introductory] NS-3 (www.nsnam.org) is a discrete event, packet level network simulator for Internet systems. Download and install NS-3. Create a simple topology of two nodes – Node1 and Node2, separated by a point-to-point link. Setup a UdpClient on Node1 and UdpServer on Node2. Start the client application, and measure end to end throughput while varying the latency of the link. Now add another client application to Node1 and a server instance to Node2. What you need to configure to ensure that there is no conflict? Measure end-to-end throughput with the extra client and server application instances. Show screenshots of pcap traces which indicate that delivery is made to the appropriate server instance.

Solution 1:

(a)

Serial	Delay	Throughput
1	5	4602.7
2	10	4597.68
3	15	4592.67

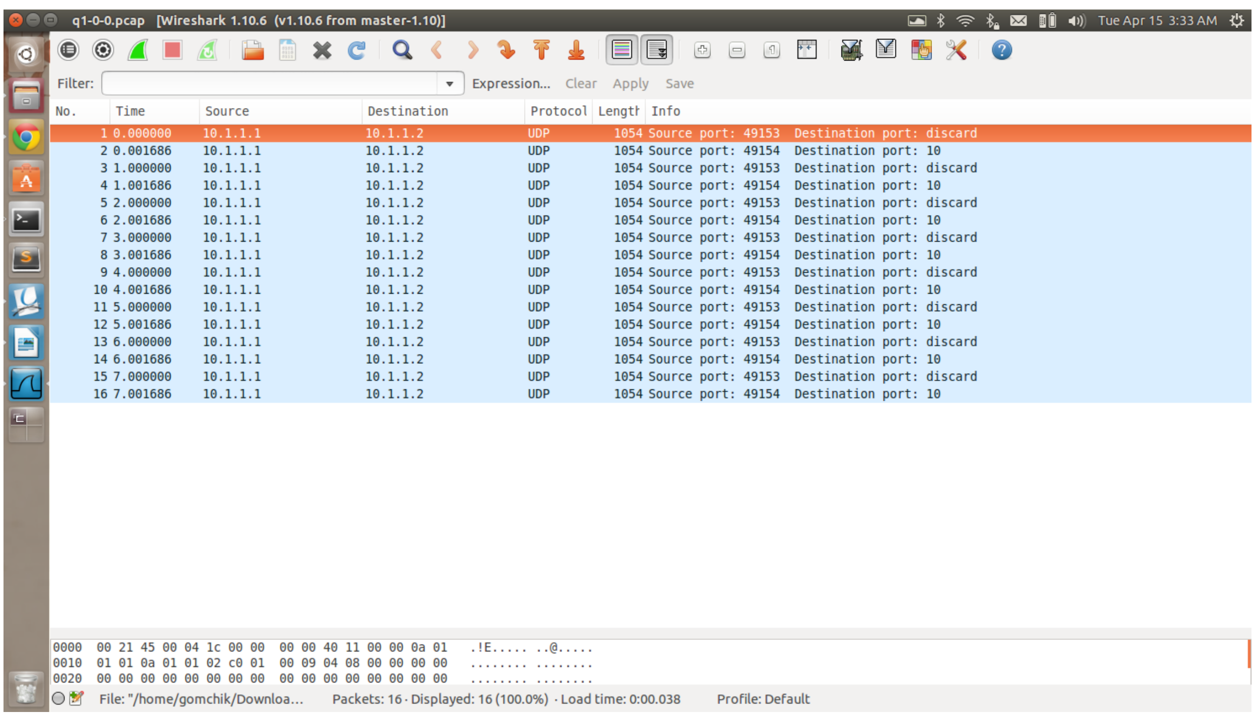

first_modified-0-0.pcap [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
Tue Apr 15 3:32 AM
Filter:
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
1 0.000000 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49153 Destination port: discard
2 0.001686 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49154 Destination port: 10
3 1.000000 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49153 Destination port: discard
4 1.001686 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49154 Destination port: 10
5 2.000000 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49153 Destination port: discard
6 2.001686 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49154 Destination port: 10
7 3.000000 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49153 Destination port: discard
8 3.001686 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49154 Destination port: 10
9 4.000000 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49153 Destination port: discard
10 4.001686 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49154 Destination port: 10
11 10.005059 10.1.1.2 10.1.1.1 UDP 1054 Source port: 10 Destination port: 49154
12 11.005059 10.1.1.2 10.1.1.1 UDP 1054 Source port: 10 Destination port: 49154
13 12.005059 10.1.1.2 10.1.1.1 UDP 1054 Source port: 10 Destination port: 49154
0000 00 21 45 00 04 1c 00 00 00 00 40 11 00 00 0a 01 .!E..... ..@.....
0010 01 01 0a 01 01 02 c0 01 00 09 04 08 00 00 00 00 ....
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
File: "/home/gomchik/Downloa... Packets: 13 · Displayed: 13 (100.0%) · Load time: 0:00.050 Profile: Default

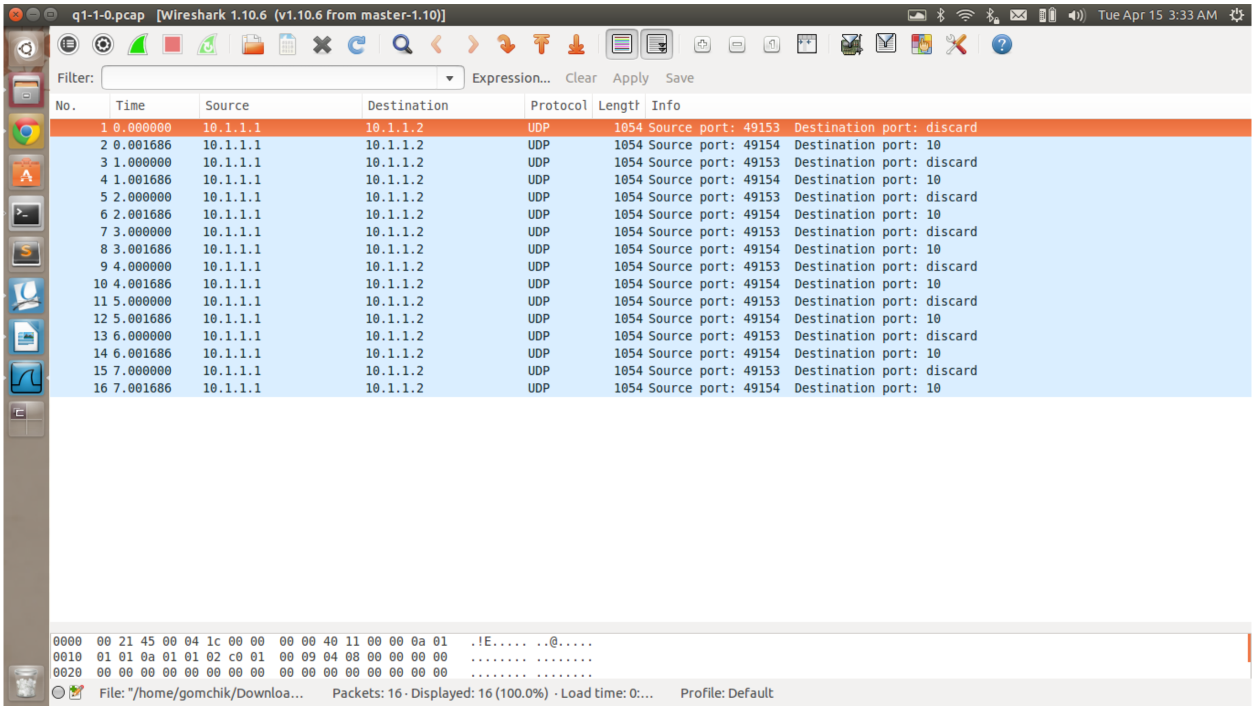

first_modified-1-0.pcap [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
Tue Apr 15 3:32 AM
Filter:
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
1 0.000000 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49153 Destination port: discard
2 0.001686 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49154 Destination port: 10
3 0.001686 10.1.1.2 10.1.1.1 UDP 1054 Source port: 10 Destination port: 49154
4 1.000000 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49153 Destination port: discard
5 1.001686 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49154 Destination port: 10
6 1.001686 10.1.1.2 10.1.1.1 UDP 1054 Source port: 10 Destination port: 49154
7 2.000000 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49153 Destination port: discard
8 2.001686 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49154 Destination port: 10
9 2.001686 10.1.1.2 10.1.1.1 UDP 1054 Source port: 10 Destination port: 49154
10 3.000000 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49153 Destination port: discard
11 3.001686 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49154 Destination port: 10
12 4.000000 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49153 Destination port: discard
13 4.001686 10.1.1.1 10.1.1.2 UDP 1054 Source port: 49154 Destination port: 10
0000 00 21 45 00 04 1c 00 00 00 00 40 11 00 00 0a 01 .!E..... ..@.....
0010 01 01 0a 01 01 02 c0 01 00 09 04 08 00 00 00 00 ....
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
File: "/home/gomchik/Downloa... Packets: 13 · Displayed: 13 (100.0%) · Load time: 0:00.043 Profile: Default

(b)

To make sure that there is no conflict in the instances of client and server application instances, we need to make sure that the port numbers are different.

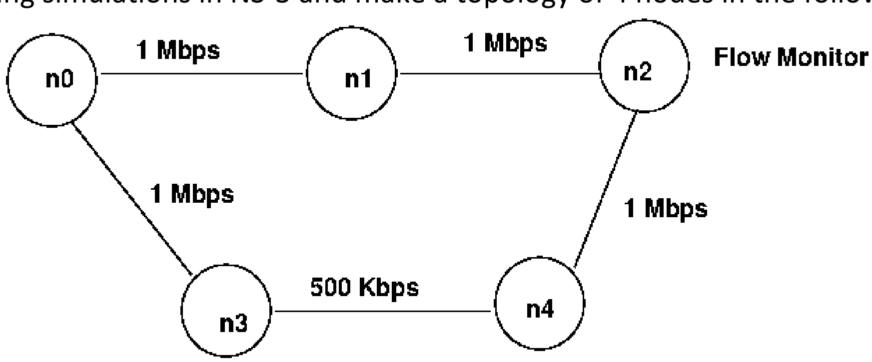
Serial	Delay	Throughput
1	5	5264.695
2	10	5259.65
3	15	5256.87





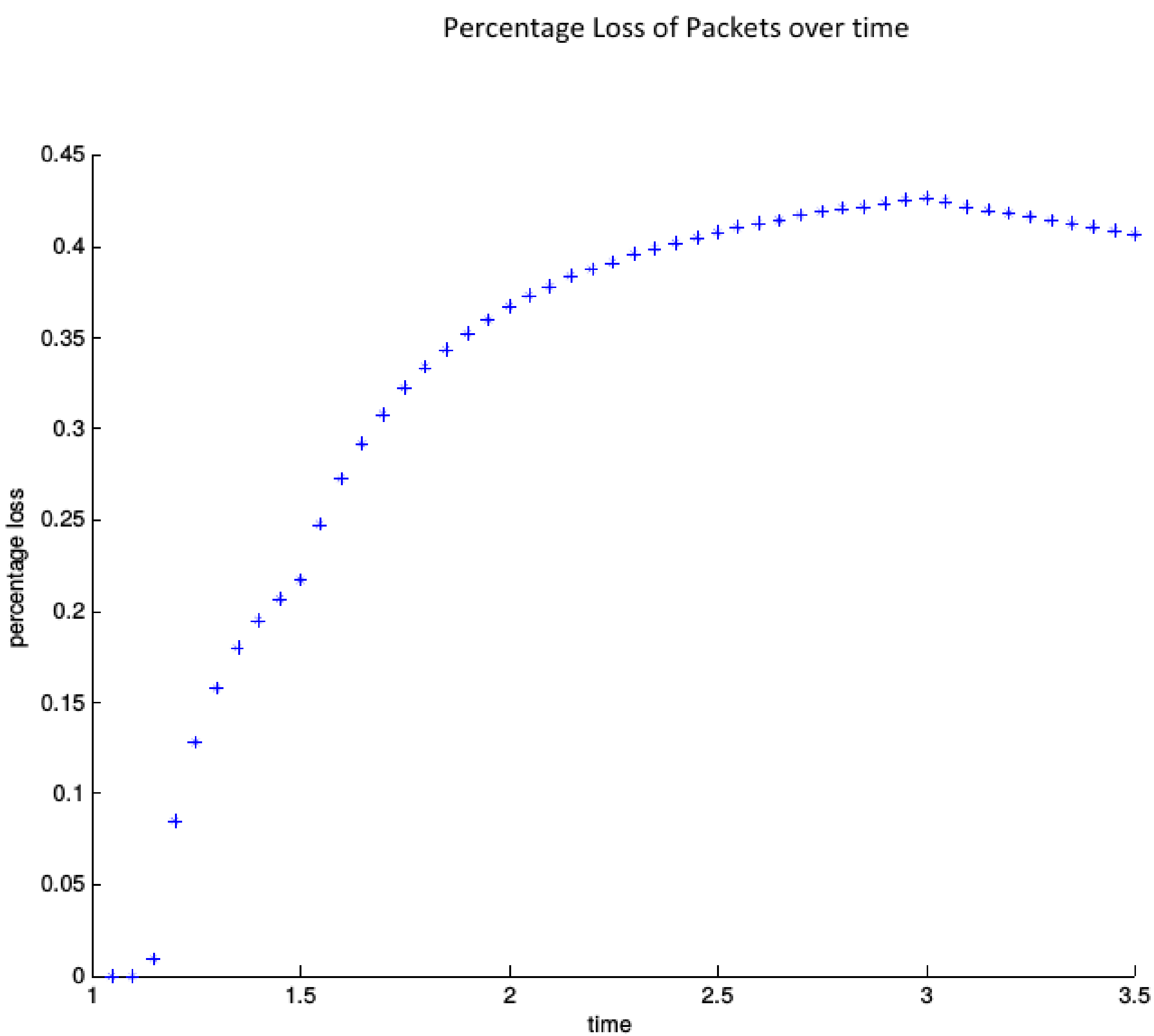
Question 2:

Make the following simulations in NS-3 and make a topology of 4 nodes in the following way.



n0 starts CBR traffic at time 1.0 of rate 900 Kbps destined for n2. n0 starts another CBR traffic at time 1.5 of rate 300 Kbps destined for node n1. At time 2.0, link from n0 to n1 goes down. Use a dynamic routing protocol so that path n0-n3-n2 is used now At time 2.7, link n0-n1 comes up again. At time 3.0, CBR traffic destined for node n1 stops. CBR destined for n2 stops at time 3.5. Use a Flow monitor to monitor losses at n2. Draw a graph of percentage loss as a function of time for the duration of simulation. Give an explanation for results you find.

Solution 2:



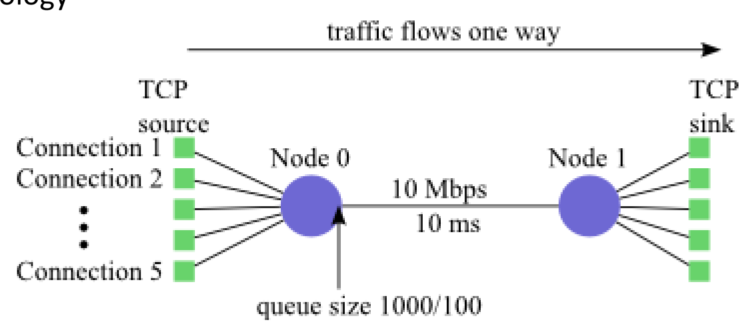
Note that the plot is the %age of total packets lost right from starting as a function of time. So, the slope of graphs give us the rate increase/decrease in packet loss.

Initially, owing to the global routing protocol, all the routing tables are set up. Assuming hop count as a measure of link costs, the routing tables would have been such that the CBR traffic from node0 goes to node2 via node1. With routing tables already set up, the packet loss was zero. Now at T=2.0sec, the node0node1 link went down. So, until the routing protocol modifies the table again, the packets will be lost. As a result the %age of packet loss spikes after T=2.0 sec, signifying a sudden increase in the no. of packets lost. With periodic and triggered updates, the

routing protocol (like OSPF) sets up the table and figures out an alternative path (node0node3node4node2) and the rate of increase of packet
losses (slope of the curve) start dropping. However, since the traffic on link n0n3 has increased a lot (due to the other already running n0n3 flow), packet losses continue to occur. At 2.7 sec, the n0n1 link becomes live again. The resetting up of table takes time and there is a momentory spike in loss% curve near T=2.7. However, after sometime (once tables are resetup), the n0n2 flow starts using the n0n1n2 path. This frees the congestion on n0n3 link and with routing tables set up, the packet losses start decreasing (negative slope means that lost/transmitted ratio is decreasing. But “transmitted” value increases. So, “lost” value must be decreasing at a faster rate). Eventually, at T=3.5 sec, with all the flows complete, the curve flattens out, signifying no more losses (in fact, no more transmissions as well).

Question 3:

Consider following topology



Connection 1 starts at time 0, Connection 2 at time 5, Connection 3 at time 10, Connection 4 at time 15, and Connection 5 at time 20. End Connection 1 at time 50.1, Connection 2 at time 45, Connection 3 at time 40, Connection 4 at time 35, and Connection 5 at time 30.

Using the generated trace files, produce graphs for the following:

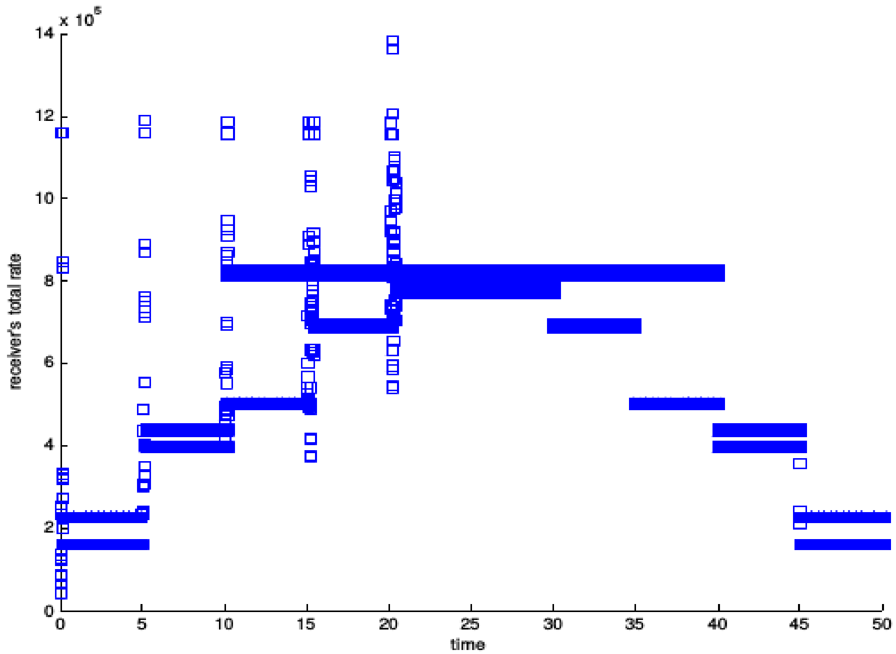
- The receivers' rates over time. Calculate the rate for each TCP connection separately. Plot the rate for all of the receivers on the same graph. To plot a smooth rate, calculate the rate over a moving window of 10 packets. You can do this by plotting a point every time a packet is received. if *packet*$_i$ is received at time t_i, plot a point at t_i equal to the sum of the lengths of *packet*$_{i-10}$ through *packet*$_i$, divided by the t_i - t_{i-10}. You will need to modify this slightly to handle the start of the connection, when there are fewer than 10 previous packets.

- The queue size over time and each packet drop event. You can calculate the queue size at any time by observing all packet enqueue, dequeue, and drop events. Plot each drop event at the maximum queue size when the drop occurs using an "X" symbol.
- The Congestion Window over time for each TCP connection.

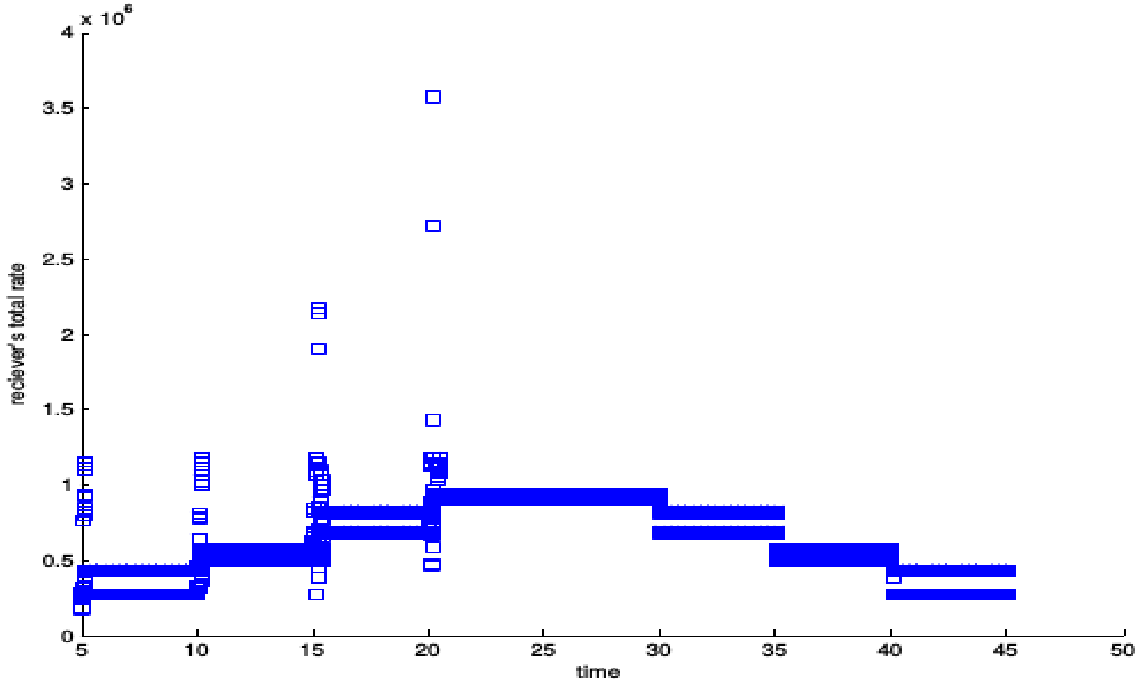
Solution 3:

Receiver rate over time Graphs

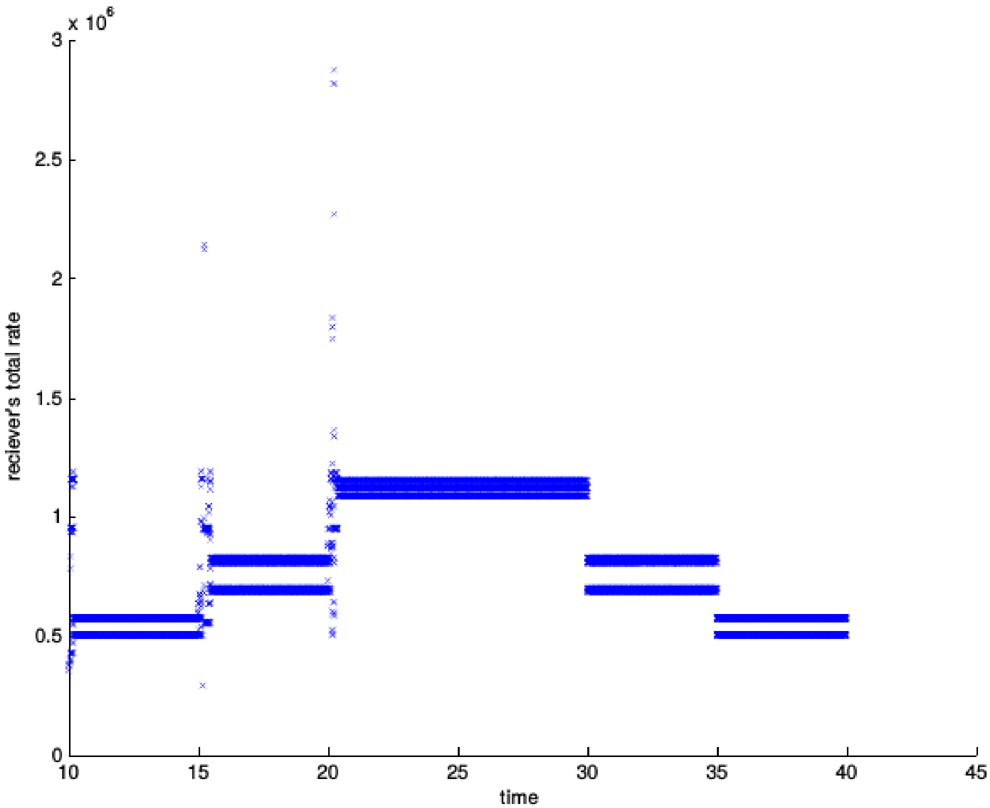
a) Port 1



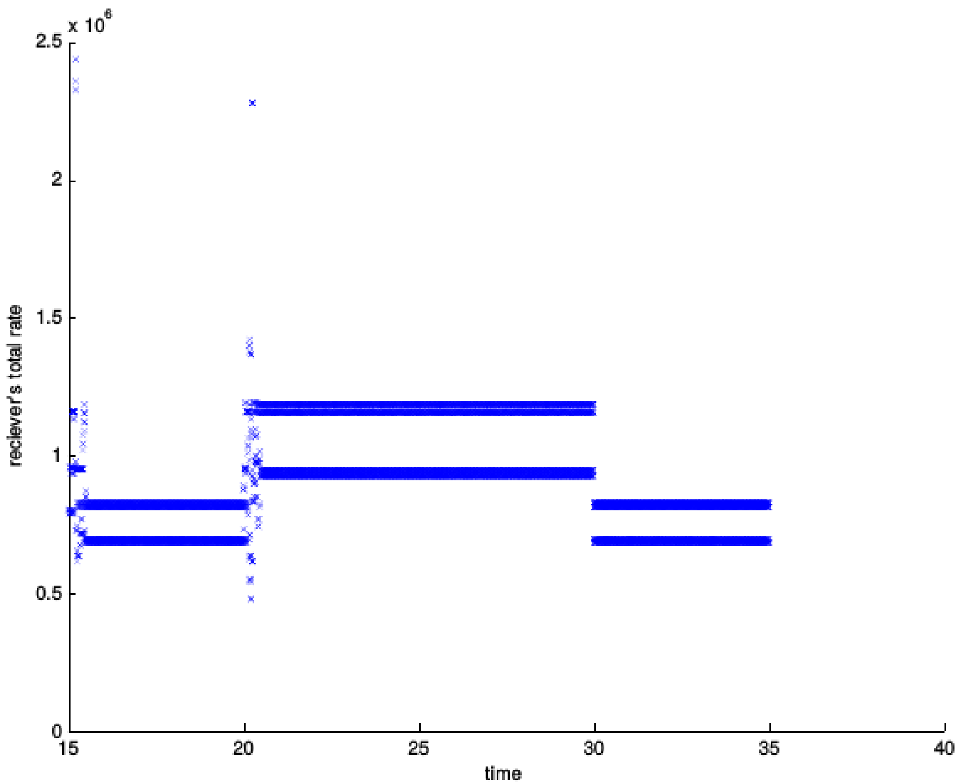
b) Port2


x 10^6
reciever's total rate
time
0
0.5
1
1.5
2
2.5
3
3.5
4
5
10
15
20
25
30
35
40
45
50

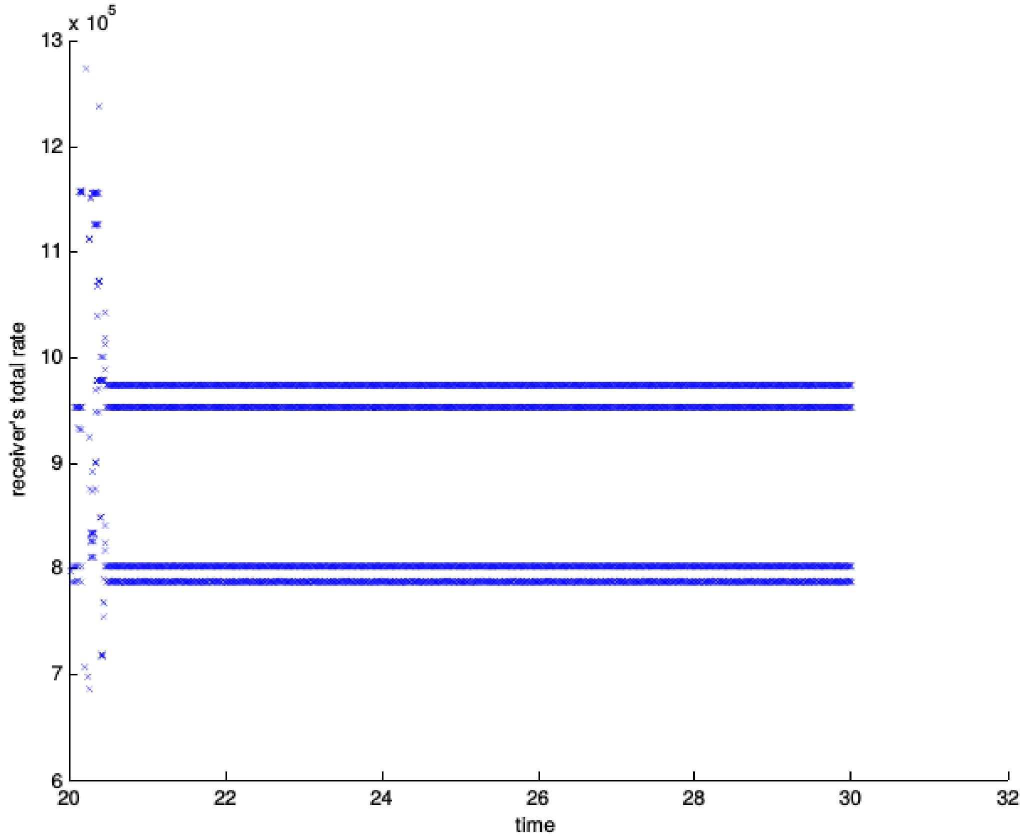
c) Port3


x 10^6
3
2.5
2
1.5
1
0.5
0
reciever's total rate
10
15
20
25
30
35
40
45
time

d) Port4

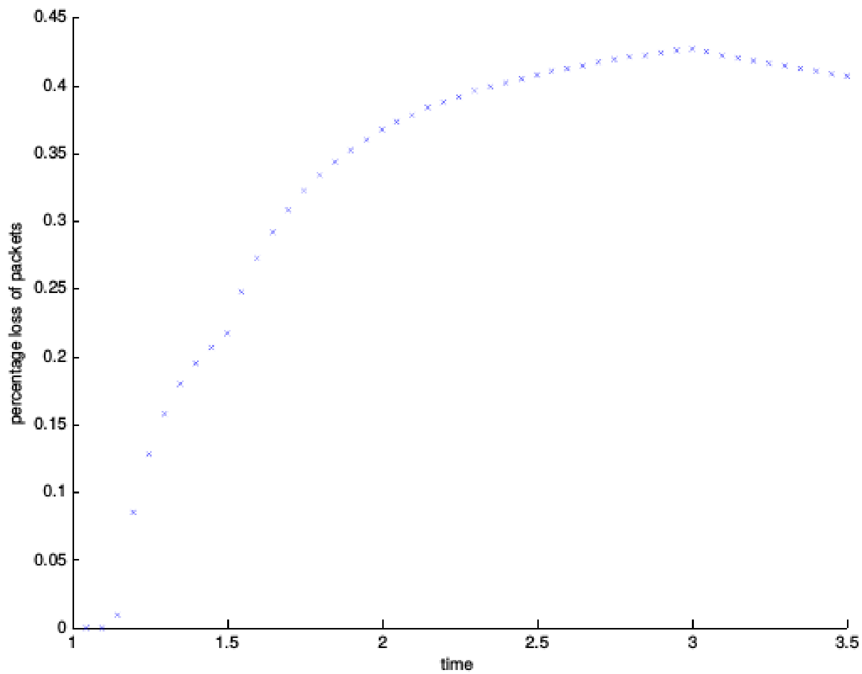


e) Port 5


x 10^5
13
12
11
10
9
8
7
6
receiver's total rate
20
22
24
26
28
30
32
time

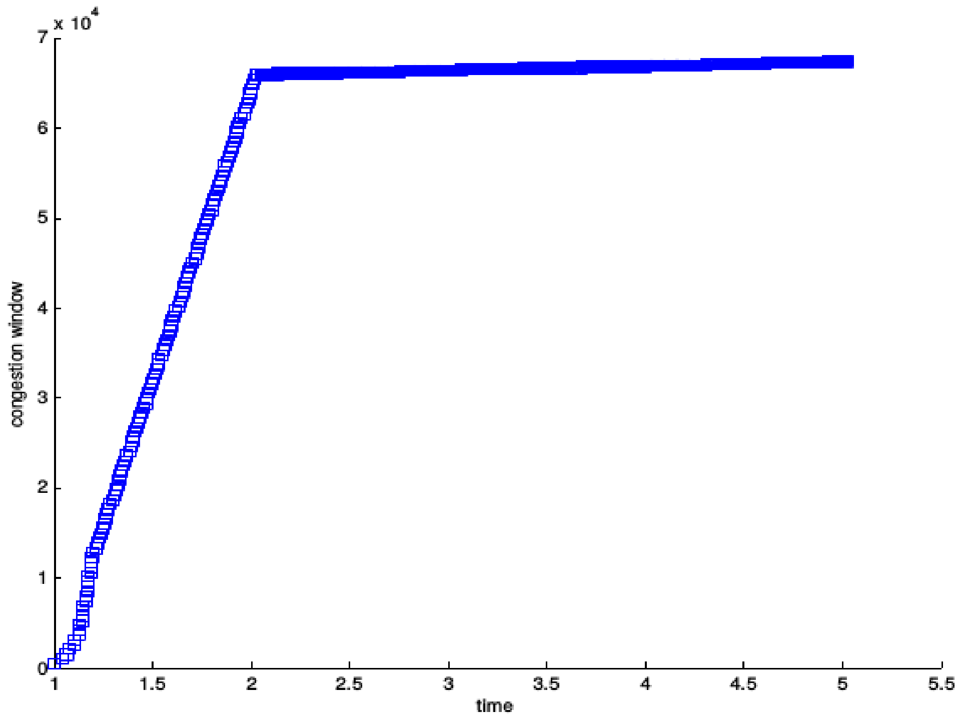
2.

Queue Size over Time Graph



3.

Congestion Window over time for each TCP connection



Use only TCP NewReno for this experiment. Turn in all of your graphs and then answers to the following questions :

Ques (a) : What do you observe about fairness among the various TCP connections?

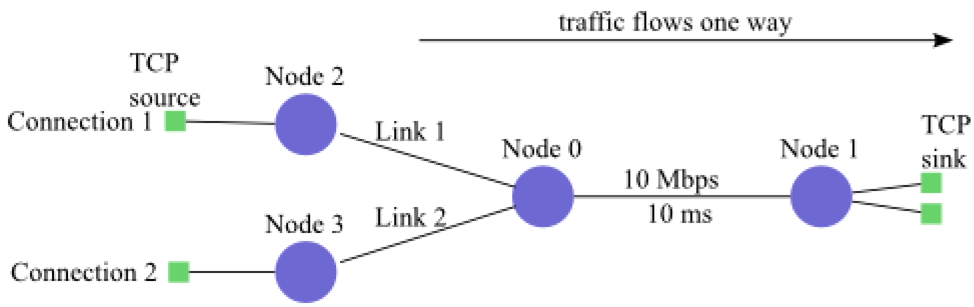
Ans (a) : Can't exactly measure it with the output but fit should be fair.

Ques (b) : What do you observe about the queue size and packet drops?

Ans (a) : The queue is 40% full most of the time.

Question 4:

Consider following topology :



The queue size limit on all nodes is 10. The bandwidth on Links 1 and 2 is 1.5 Mbps. The propagation delay on Links 1 and 2 is initially 10 ms, but you will be varying the propagation delay on Link 2 for your experiments.

Connection 1 starts at time 0, using Link 1. Connection 2 starts at time 0. Both connections run for 5 seconds.

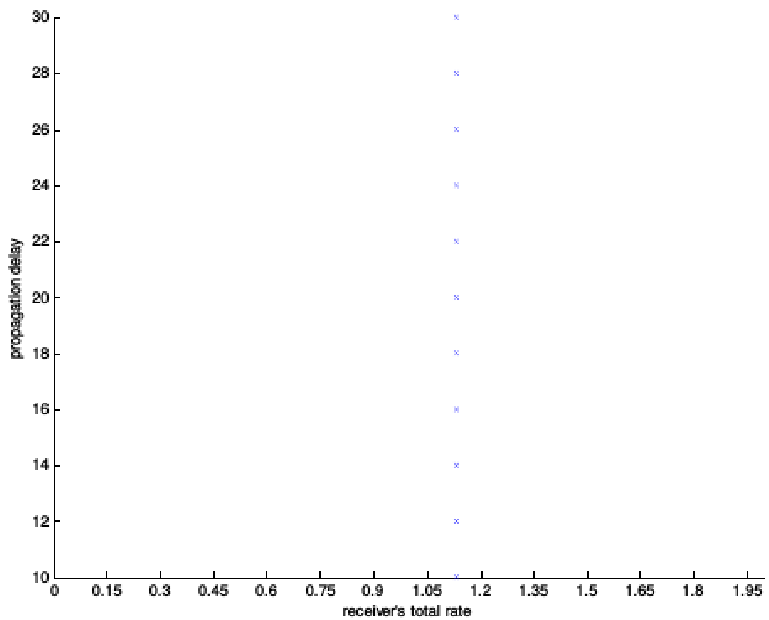
Plot a graph to show the throughput for each connection over time and observe that they get approximately the same amount. Now run a set of experiments, varying the propagation delay on Link 2 so that Connection 2 has an increasingly longer RTT. Produce the following graph:
The receiver's total rate (averaged over the length of the connection) versus the propagation delay on Link 2. Plot each connection separately.

Use only TCP NewReno for these experiments. What do you observe about the relationship between RTT and fairness in TCP?

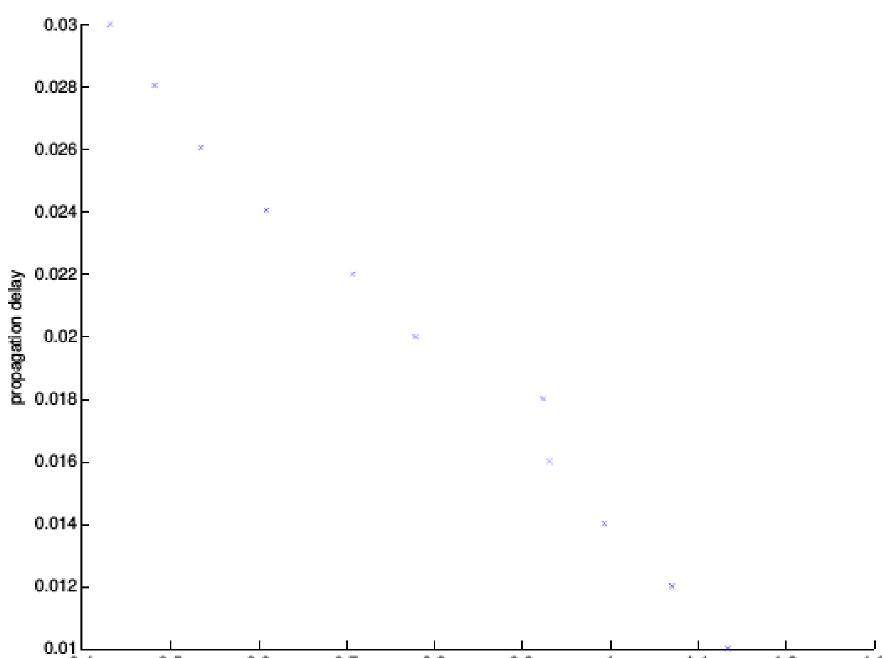
Solution 4:

Q4:

Receiver Total Rate vs Propagation Delay on Link 2



Receiver Total Rate vs Propagation Delay on Link 2



As we can see from the plots the final throughput of link1 is almost constant but the final throughput of link2 is getting reduced as the rtt is increasing. This has to be the case as the bandwidth of n0n1 link(10 Mbps) is larger than the sum of bandwidth on the n2n0 and n3n0(both being 1.5 Mbps thus the total being 10Mbps). So, no scheduling needs to be done on the n0 node.
And, as it takes more time for the packets of node 3 to reach node 0 as compared to those of node 2(as link1 has lesser rtt). So, the throughput of n3 is getting reduced.