Calculate the transmission delay and propagation delay and show that the amount of delay matches the time at which the packet arrives at node n2.

Ans) We know that the formula for calculating the transmission delay is

Transmission Delay=Length of Packet/ Transmission Rate ie L/R

It is given that the length is 40 bytes converting into bits we have

L=40*8 bits

From the File Reno.tcl while creating the links we know that

R=5Mb converting into bits we get

R=5*10^6

Hence, we get

d_{trans}=(40*8)/(5*10^6)

d_{trans}=320/5*10^6

d_{trans}=0.000064sec.

Also, while creating the link, the Propagation delay=20ms=20*10^-3

d_{prop}=0.02sec

We can calculate the total delay by adding the d_{trans} and d_{prop}

Total Delay=0.000064+0.02 sec

=0.020064 sec

Moreover, it is given that the packet starts loading at time=0.1 sec

Therefore, the total time taken by the packet to reach the destination is

=0.020064+0.1

=0.120064

In the file trace.tr the packet reaches at the time 0.120064

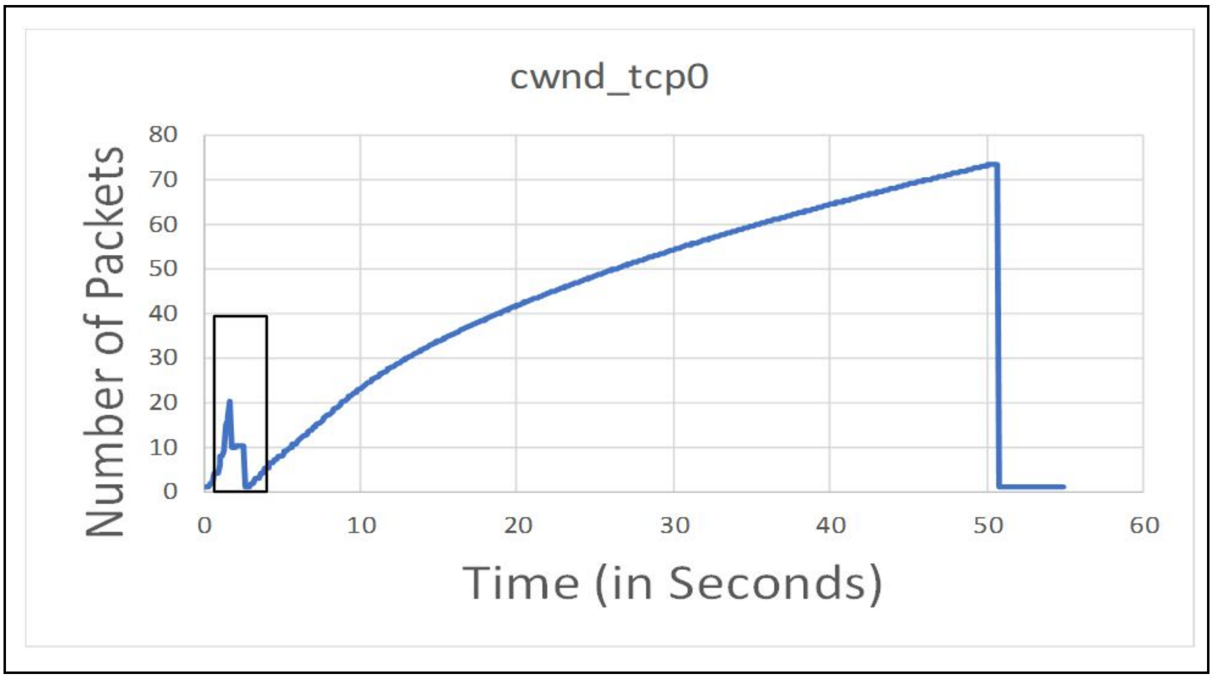
This is same as the total time which we calculated

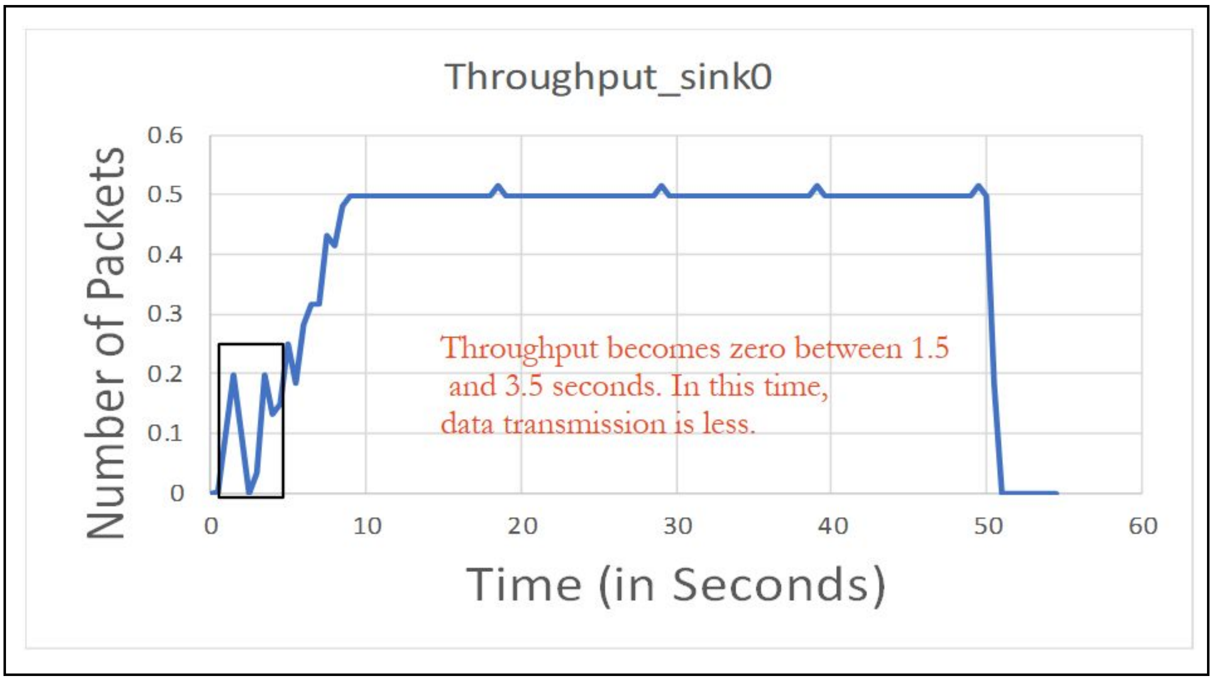
Now compare the moments that a drop occurs in the trace file with the fluctuations in the cwnd and throughput charts. Write down your observations and conclusions in your report file.

Ans) **TCP RENO**

The below two charts were plotted from the cwnd.tr and thrpt.tr

One shows the fluctuations in congestion window while other shows fluctuations in throughputs





In TCP RENO there is fast recovery ie whenever the TCP sender receives duplicate Acks before receiving a new Ack (a new Ack is nothing but an Ack with a higher value that the previous highest Ack value)then it does not go to the slow start process in a fast retransmit, but what it does is, it will reduce the size of the congestion window to half of the current congestion widow size and it also would reset the ssthresh to match this value.

According to the file trace.tr the drops occur at

```
d 1.371968 2 3 tcp 1040 ------- 0 0.0 4.0 23 38

d 1.388608 2 3 tcp 1040 ------- 0 0.0 4.0 25 40

d 1.405248 2 3 tcp 1040 ------- 0 0.0 4.0 27 42

d 1.421888 2 3 tcp 1040 ------- 0 0.0 4.0 29 44

d 1.672704 2 3 tcp 1040 ------- 0 0.0 4.0 39 66
```

Our Discussions are as follows

```
d 1.371968 2 3 tcp 1040 ------- 0 0.0 4.0 23 38
```

From the above line which is taken from the trace.tr file we can say that

At time 1.371968 a drop occurs. This drop is expected because of three duplicate Acks. As a result of this drop, we observe that cwnd changes from 20 to 10 because RENO reduces it to half.The sequence number is 23 and we know that the packet belonging to sequence number 23 ie 38 is dropped .So now the node 2 becomes the source and the node 0 becomes the destination for sending an ACK now the destination ie node 0 receives three duplicate ACKS which are shown below

```
r 1.70096 2 0 ack 40 ------- 0 4.0 0.0 22 53

r 1.7176 2 0 ack 40 ------- 0 4.0 0.0 22 54

r 1.73424 2 0 ack 40 ------- 0 4.0 0.0 22 57

r 1.75088 2 0 ack 40 ------- 0 4.0 0.0 22 60
```

From this data we can say that the time interval in which duplicate ACKs were sent is [1.70096,175088]

If we observe this time interval in the throughput graph we see that throughput drops to zero

```
d 1.388608 2 3 tcp 1040 ------- 0 0.0 4.0 25 40
```

At time 1.388608 a drop occurs. This drop is expected because of a timeout. As a result of this drop, we observe that cwnd falls to one (1) and throughput falls in the same range that is it falls to zero.In chart of congestion window we see that at time 2.9 the window drops to one which means that packet was dropped at 1.388608 and then never reached the receiver and source waited for TTL of 1.52(2.9-1.38=1.52) and very little data is being transmitted

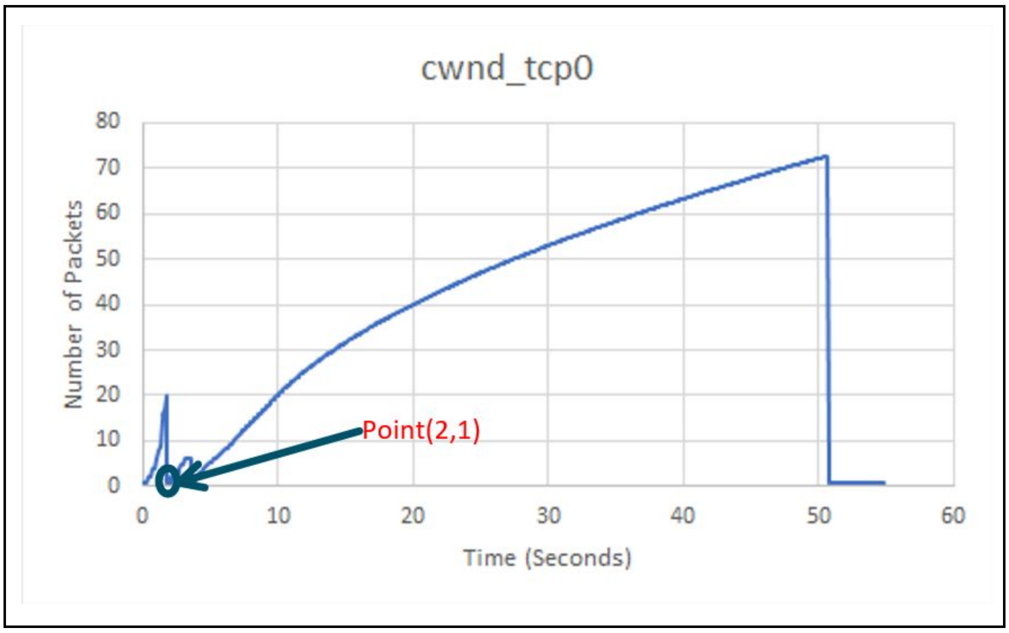
So whenever a packet is dropped in TCP RENO the same process is repeated.

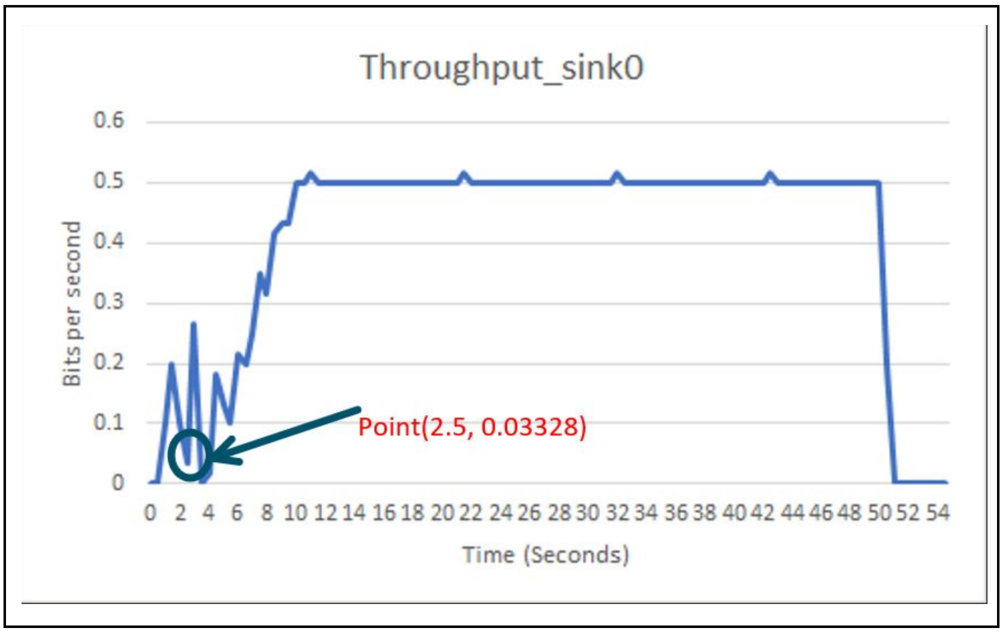
Now for this new scenario, compare the moments that a drop occurs in the trace file with the cwnd and throughput charts. Write down your observations and conclusions in your report file. Also compare the two congestion control algorithms referring to the results.

TCP TAHOE

The below two charts were plotted from the cwnd.tr and thrpt.tr

One shows the fluctuations in congestion window while other shows fluctuations in throughputs





InTCP TAHOE whenever three duplicates are received , it performs Fast Retransmit and it sets the slow start threshold to half of the current congestion window and reduces the congestion window to 1 MSS , resets to slow start state.

According to the file trace.tr the drops occur at

1.371968 2 3 tcp 1040 ------- 0 0.0 4.0 23 38

1.388608 2 3 tcp 1040 ------- 0 0.0 4.0 25 40

1.405248 2 3 tcp 1040 ------- 0 0.0 4.0 27 42

1.421888 2 3 tcp 1040 ------- 0 0.0 4.0 29 44

1.672704 2 3 tcp 1040 ------- 0 0.0 4.0 39 66

2.731328 2 3 tcp 1040 ------- 0 0.0 4.0 43 102

2.741312 2 3 tcp 1040 ------- 0 0.0 4.0 44 103

Our Discussions are as follows

1.371968 2 3 tcp 1040 ------- 0 0.0 4.0 23 38

At time 1.371968 a drop occurs and the packet number is 38. This drop is expected because of a three duplicate ACKs. As a result of this drop, we observe that cwnd changes or drops from 20 to 1 which means that the packet which dropped at time 1.371968 has never reached the receiver and the Duplicate ACKs are received as shown below

1.388608 2 3 tcp 1040 ------- 0 0.0 4.0 25 40

At time 1.388608 a drop occurs and the packet number is 40. This drop is expected because of Timeout. As a result of this drop, we observe that cwnd changes from 6 to 1 which means that the packet which dropped at time 1.371968 has never reached the receiver and the source has waited for TTL of 2.11(3.5-1.388=2.11) and never received ACK for that packet

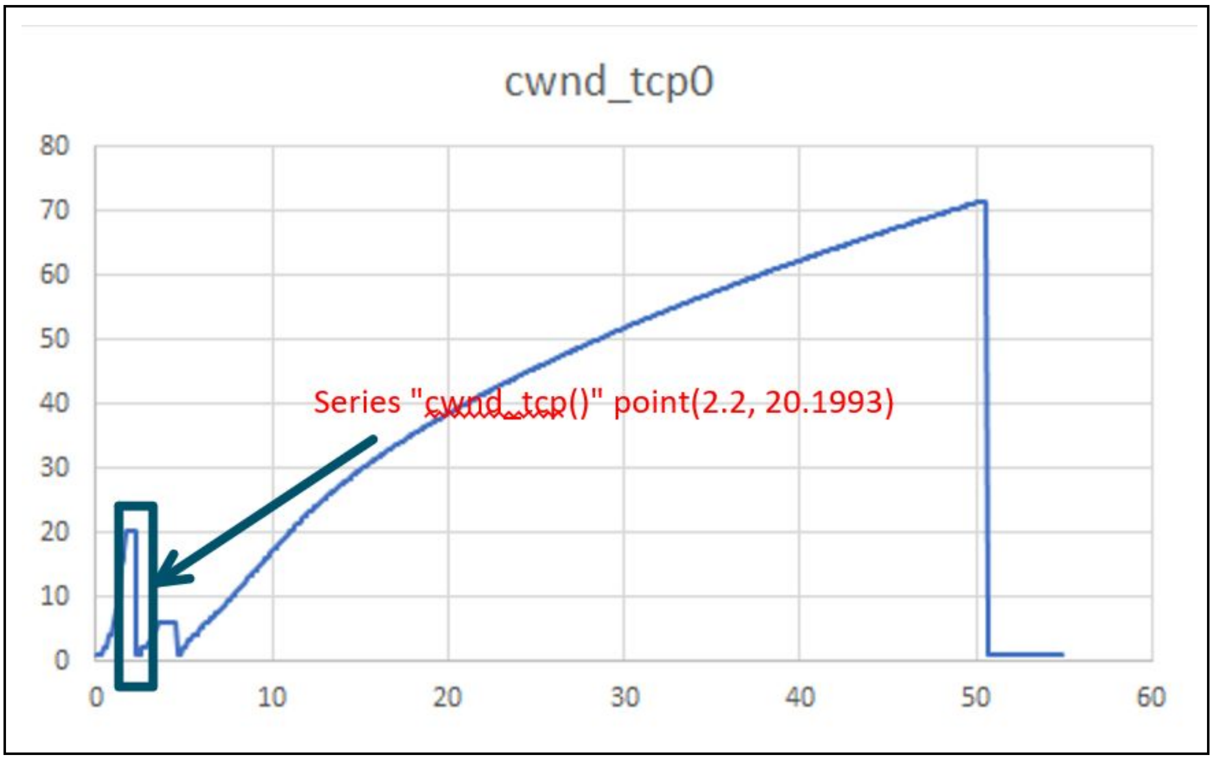
For the time interval 1.5 to 3 in the throughput chart the throughput drops to 2.5 and in the same chart in the time interval 3.0 to 4.0 throughput drops to 0

TCP SACK1

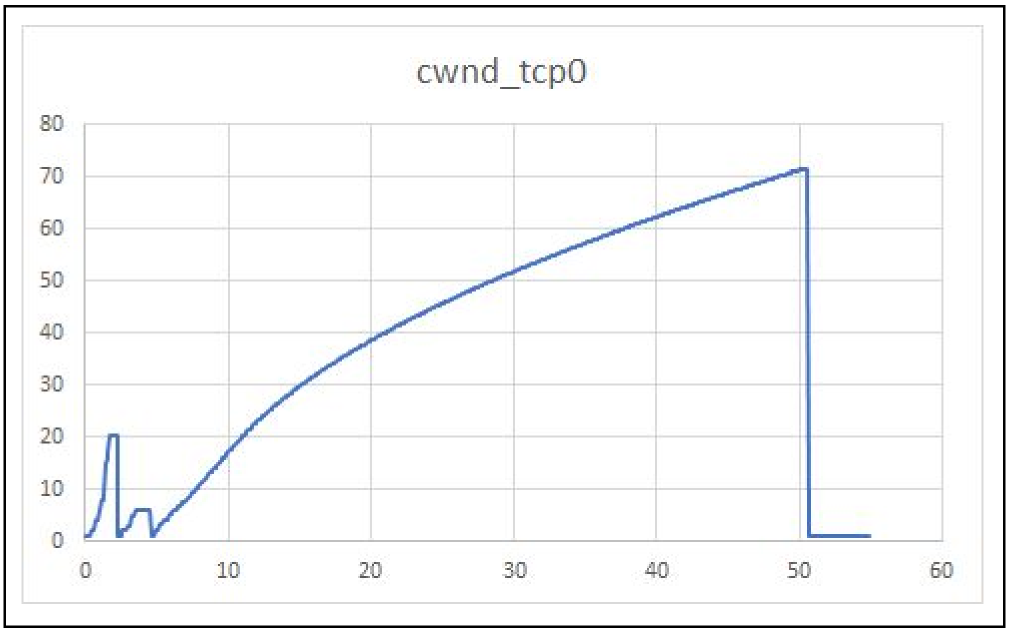
The below two charts were plotted from the cwnd.tr and thrpt.tr

One shows the fluctuations in congestion window while other shows fluctuations in throughputs

cwnd



Throughput



```
r 2.051616 2 0 ack 40 ------- 0 4.0 0.0 22 81

r 2.068256 2 0 ack 40 ------- 0 4.0 0.0 24 82

r 2.368992 2 0 ack 40 ------- 0 4.0 0.0 26 85
```

There are some limitations for TCP like it might experience poor performance when the multiple packets are lost from one window of data.As TCP uses cumulative acknowledgements it has limited amount of information from which a TCP sender would only be able to learn about a single lost packet per round trip time.Also an aggressive sender could retransmit packets early but such retransmitted segments might have already been received.So,inorder to overcome these limitations a Selective Acknowledgement(SACK) mechanism combined with the selective repeat retransmission policy can be used.Here the receiver sends SACK packets to sender informing about data received,so it becomes easy for the sender to retransmit only the segments which are missing.

According to the file trace.tr the drops occur at

```
d  1.371968     2        3        tcp      1040     -------    0        0        4        23       38
```

```
d  1.388608    2     3     tcp    1040   -------  0     0     4     25     40

d  1.405248    2     3     tcp    1040   -------  0     0     4     27     42

d  1.421888    2     3     tcp    1040   -------  0     0     4     29     44

d   1.672704   2     3     tcp    1040   -------  0     0     4     39     66

d  3.204768    2     3     tcp    1040   -------  0     0     4     43     100
```

Our Discussions are as follows

```
d   1.371968   2     3     tcp    1040   -------  0     0     4     23     38
```

At time 1.371968 a drop occurs and the Packet number is 38. This drop is expected because of a timeout. As a result of this drop, we observe that cwnd changes from 20 to 1 at 2.2 which means that the packet which was dropped at 1.371968 has never reached the receiver and also the source waited for TTL of 0.38(2.2-1.37=0.38) and it never received ACK for that packet

```
d  1.388608    2     3     tcp    1040   -------  0     0     4     25     40
```

At time 1.388608 a drop occurs and the Packet number is 40. This drop is expected because of a timeout. As a result of this drop, we observe that cwnd changes from 6 to 1 at 4.5 which means that the packet which was dropped at 1.388608 has never reached the receiver and also the source waited for TTL of 3.11(4.5-1.388=3.11) and it never received ACK for that packet

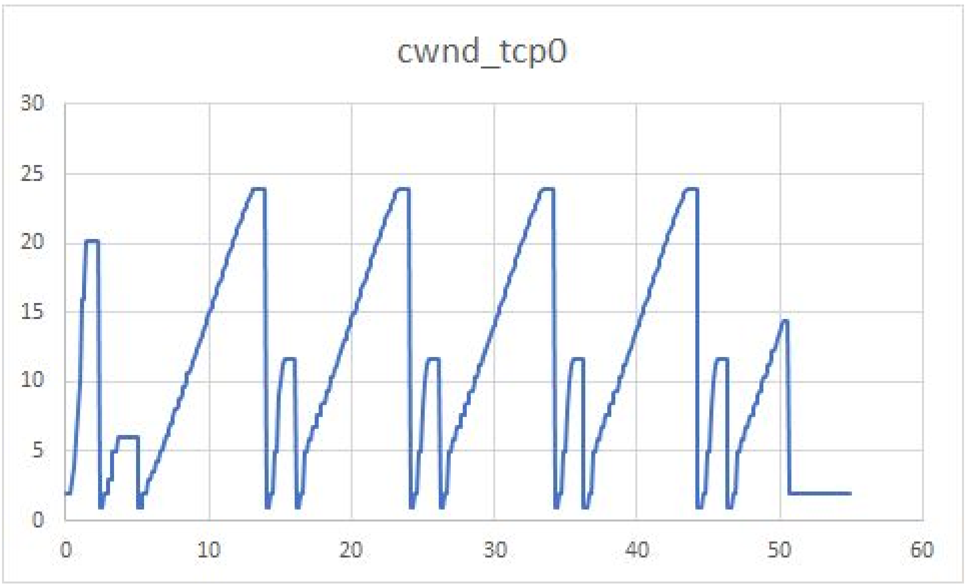
If we see the chart for throughput between the time interval 2.5 to 3.5 the throughput drops to 0.03328 and also for the time interval between 4.0 to 5.0 the throughput drops to zero (o).

TCP LINUX

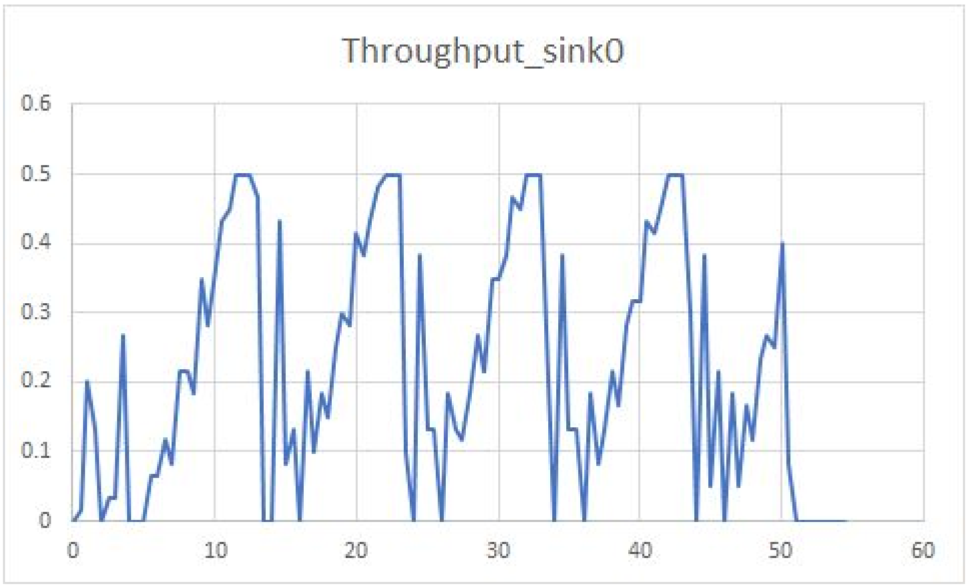
The below two charts were plotted from the cwnd.tr and thrpt.tr

One shows the fluctuations in congestion window while other shows fluctuations in throughputs

Cwnd



Throughput



The TCP Linux runs the TCP congestion control modules which are imported from the Linux Kernel.The congestion control modules of LInux are compiled into the NS-2 binary.The TCP Linux generates simulation results which are consistent in the congestion window trajectory level with the behaviour of the Linux hosts.The implementation loosely follows the Linux TCP packet processing routine and calls the congestion control source codes from the Linux Kernel to change congestion control related parameters for example congestion window, slow start threshold etc.Users can select different congestion control algorithms , different congestion control module parameters,different Linux TCP parameters for different instances of this agent.This agent supports SACK.A receiver that supports SACK is recommended to work with this agent

According to the file trace.tr the drops occur at

```
d 1.071232 2 3 tcp 1040 ------- 0 0.0 4.0 22 36

d 1.087872 2 3 tcp 1040 ------- 0 0.0 4.0 24 38

d 1.104512 2 3 tcp 1040 ------- 0 0.0 4.0 26 40

d 1.121152 2 3 tcp 1040 ------- 0 0.0 4.0 28 42

d 1.371968 2 3 tcp 1040 ------- 0 0.0 4.0 38 64

d 3.284688 2 3 tcp 1040 ------- 0 0.0 4.0 42 98

d 12.751024 2 3 tcp 1040 ------- 0 0.0 4.0 330 668

d 14.930288 2 3 tcp 1040 ------- 0 0.0 4.0 369 753

d 22.892128 2 3 tcp 1040 ------- 0 0.0 4.0 657 1324

d 25.041392 2 3 tcp 1040 ------- 0 0.0 4.0 696 1409
```

```
d 32.993232 2 3 tcp 1040 ------- 0 0.0 4.0 984 1980

d 35.142496 2 3 tcp 1040 ------- 0 0.0 4.0 1023 2065

d 43.094336 2 3 tcp 1040 ------- 0 0.0 4.0 1311 2636

d  45.2436 2 3 tcp 1040 ------- 0 0.0 4.0 1350 2721
```

Our Discussions are as follows

```
d 1.071232 2 3 tcp 1040 ------- 0 0.0 4.0 22 36
```

At time 1.071232 a drop occurs and the packet number is 36 when the packet was forwarding from node 2 to node 3 as the outgoing queue at node 2 was full.The TCP agent at the source node 0 does not know about the drop as it has occured along the path at node 2.In the chart of congestion window we see that the window size falls from 20 to 1 at time 2.3 sec which means that packet got dropped at time 1.071232 has never reached the receiver and receiver is replying to the same Ack numbers for packet.

```
d 1.087872 2 3 tcp 1040 ------- 0 0.0 4.0 24 38
```

At time 1.087822 a drop occurs and the packet number is 38. This drop is expected because of a timeout. As a result of this drop, we observe that cwnd changes falls to 1 at time 5.1 and the packet got dropped at time 1.071232 has never reached the receiver and the source has waited for TTL 4.013(5.1-1.087872=4.013)

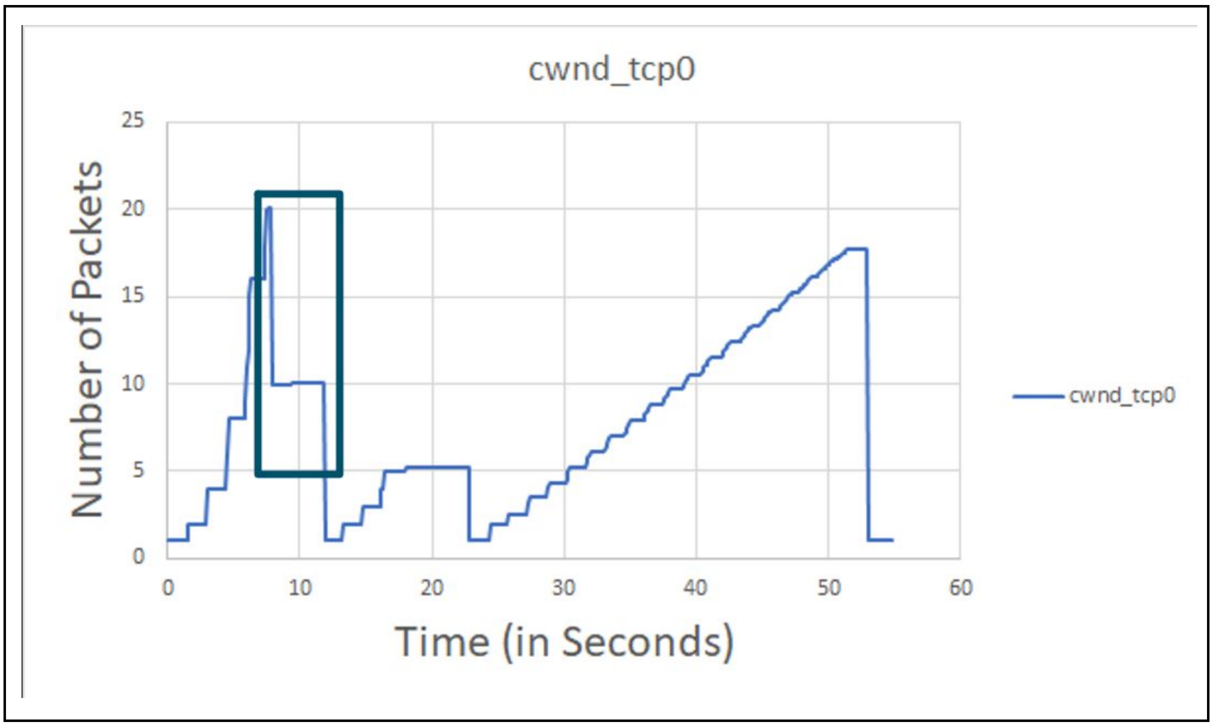
During the time interval 5 – 5.5 in the throughput chart ,the throughput at the receiver falls to zero. That means very little data has been sent from the source.

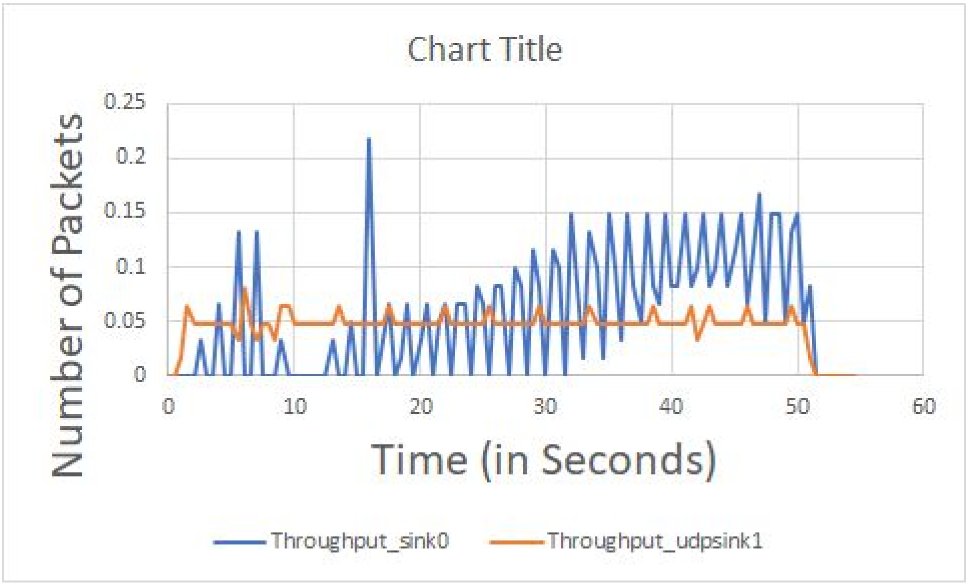
STEP2

UDP with 0.05 Mbps rate

The below two charts were plotted from the cwnd.tr and thrpt.tr

One shows the fluctuations in congestion window while other shows fluctuations in throughput





According to the file trace.tr the drops occur at

```
d   6.16992     2      3      tcp     1040    0      0      4      23     76

d   6.21152     2      3      tcp     1040    0      0      4      25     78

d   6.25312     2      3      tcp     1040    0      0      4      27     80
```

```
d   6.284        2     3     cbr    1000   0     1     5     38    82
```

```
d     6.29888    2     3     tcp    1040   0     0     4     30    84
```

Our Discussions are as follows

```
d   6.16992      2     3     tcp    1040   0     0     4     23    76
```

At time 6.16992 a drop occurs and the Packet number is 76.This drop is expected because of a 3 Duplicate Ack. As a result of this drop, we observe that cwnd changes from 20 to 10 at time 7.8 which means that the packet which was dropped at 6.16992 has never reached the receiver and the receiver repeatedly answers the same ACK numbers for next packets.

```
d    6.21152     2     3     tcp    1040   0     0     4     25    78
```

At time 6.21152 a drop occurs and the Packet number is 78. This drop is expected because of a timeout. As a result of this drop, we observe that cwnd changes from 10 to 1 at time 11.6 which means that the packet which was dropped at 6.21152 has never reached the receiver and also the source waited for TTL of 5.38848 (11.6-6.21152=5.38848) and it never received ACK for that packet.

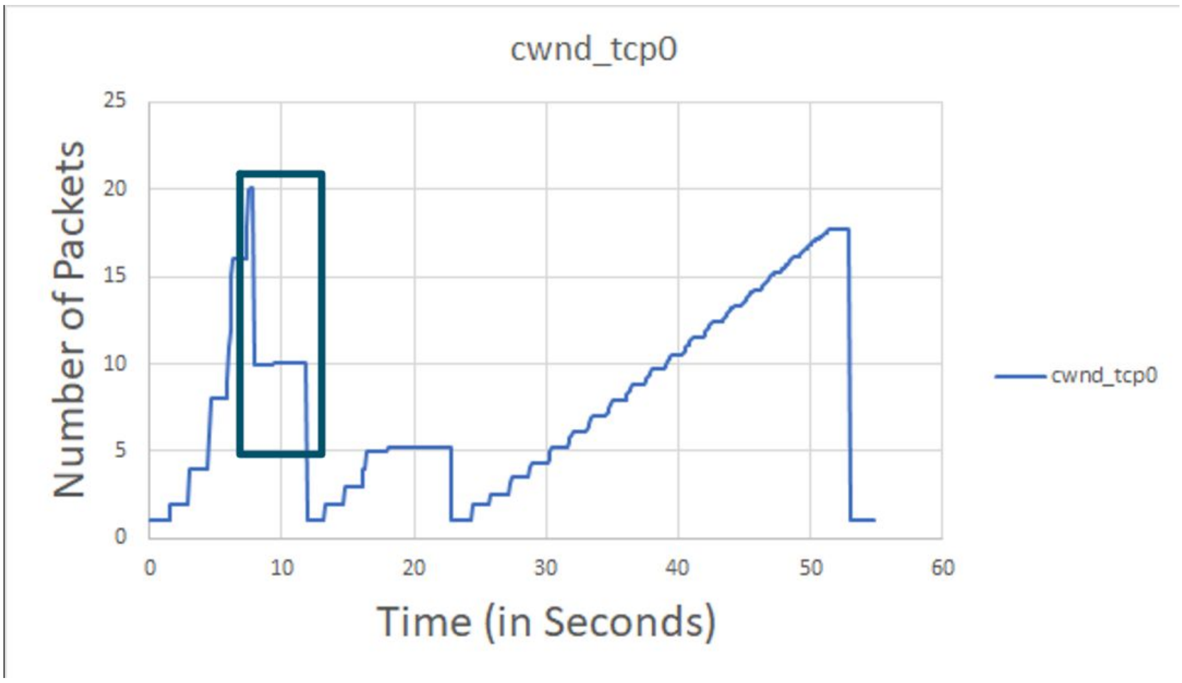
Considering the throughput graph, drop in TCP sinks were easily noticeable while the UDP sinks maintained a constant input. The TCP sink drops to zero two times in the time interval of 3 to 5.5.

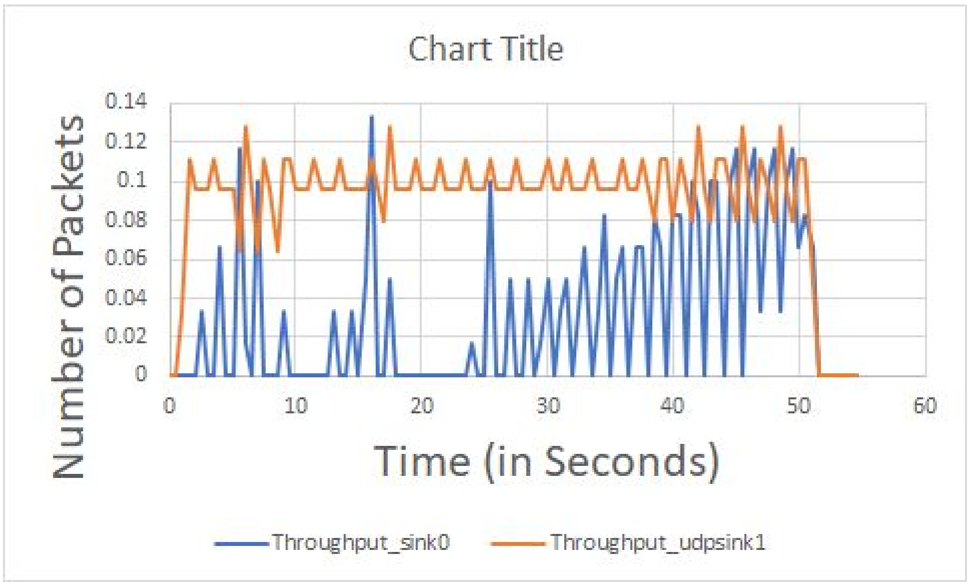
UDP sink doesn't have an easily noticeable curve down and hence has a constant transmission speed.

UDP with 0.1Mbps rate

The below two charts were plotted from the cwnd.tr and thrpt.tr

One shows the fluctuations in congestion window while other shows fluctuations in throughput





According to the file trace.tr the drops occur at

d	4.684	2	3	cbr	1000	0	1	5	56	78
d	6.12832	2	3	tcp	1040	0	0	4	21	111
d	6.20992	2	3	tcp	1040	0	0	4	23	114

```
d    6.25152   2    3    tcp    1040   0    0    4    25    116

d      6.284   2    3    cbr    1000   0    1    5    76    118

d    6.29312   2    3    tcp    1040   0    0    4    27    119

d   6.33888    2    3    tcp    1040   0    0    4    30    122

d      6.364   2    3    cbr    1000   0    1    5    77    123

d    7.58432   2    3    tcp    1040   0    0    4    38    157

d      7.724   2    3    cbr    1000   0    1    5    94    161

d   16.45264   2    3    tcp    1040   0    0    4    40    303

d    16.4568   2    3    tcp    1040   0    0    4    41    304

d  16.46096    2    3    tcp    1040   0    0    4    42    305
```

Our Discussions are as follows

```
d      4.684   2    3    cbr    1000   0    1    5    56    78
```

The drop occurs at 4.684 seconds. We see that this is not a TCP packet and basically the drop is due to traffic created in the network

```
d   6.12832    2    3    tcp    1040   0    0    4    21    111
```

At time 6.12832 a drop occurs and the Packet number is 111. This drop is expected because of a 3 Duplicate Ack. As a result of this drop, we observe that cwnd changes from 20 to 10 at time 7.8 which means that the packet which was dropped at 6.12832 has never reached the receiver and the receiver repeatedly answers the same ACK numbers for next packets.

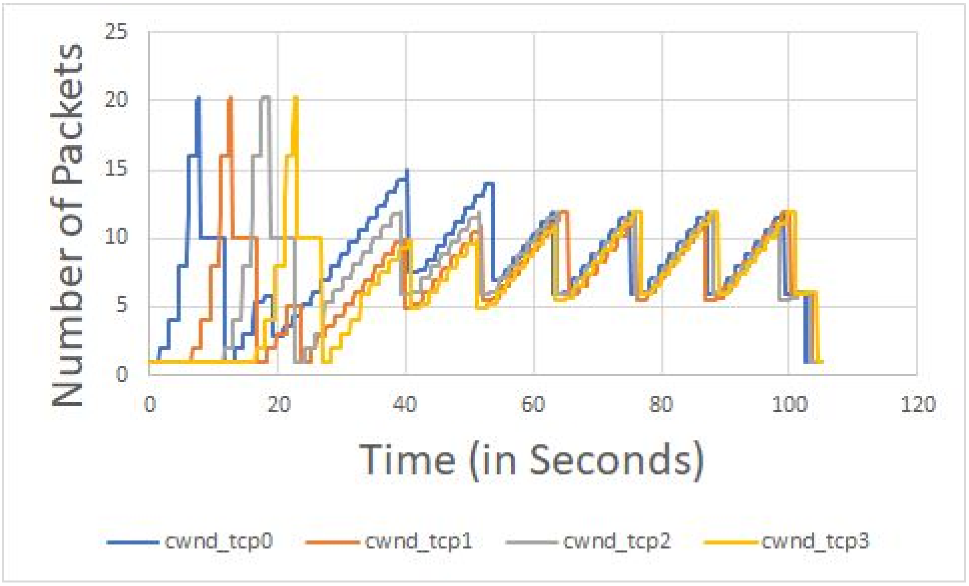
```
d    6.33888    2      3      tcp     1040    0      0      4      30      122
```

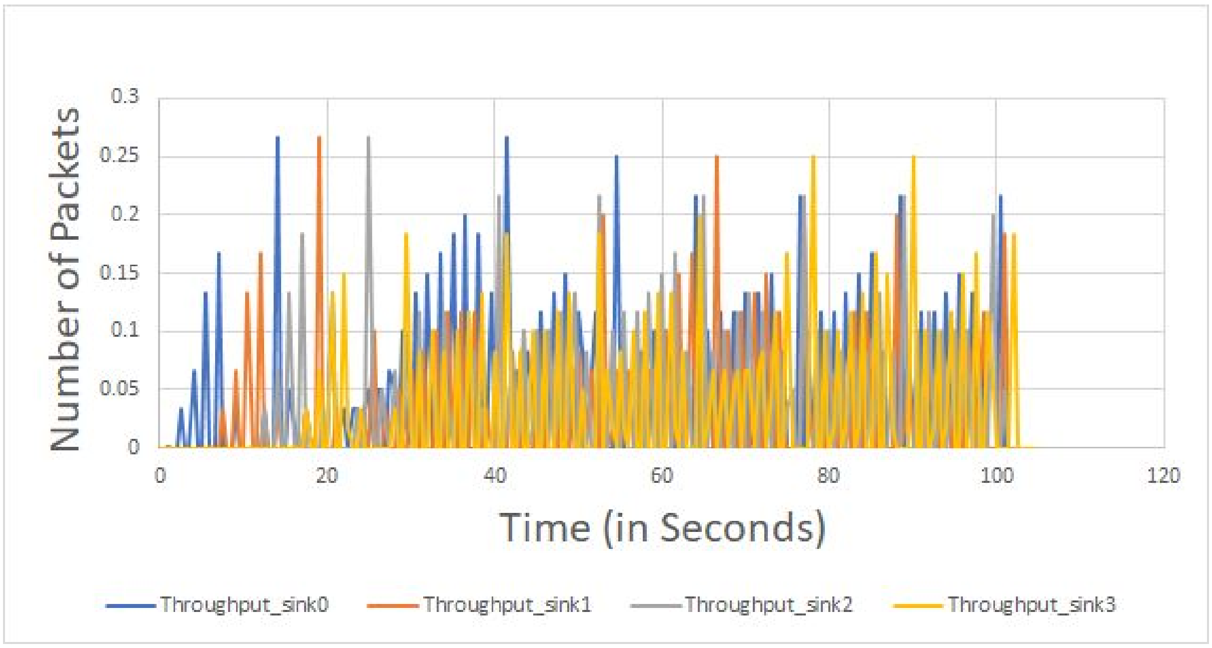
At time 6.33888 a drop occurs and the Packet number is 122. This drop is expected because of a timeout. As a result of this drop, we observe that cwnd changes from 10 to 1 at time 11.6 which means that the packet which was dropped at 6.33888 has never reached the receiver and also the source waited for TTL of 5.26 (11.6-6.34=5.26) and it never received ACK for that packet.

Considering the throughput graph, drop in TCP sinks were easily noticeable while the UDP sinks maintained a constant input. The TCP sink drops to zero in the time interval of 3.0 to 3.5.

UDP sink doesn’t have an easily noticeable curve down and hence has a constant transmission speed. One change from previous 0.05 Mbps throughput graph is that since UDP rate is high, UDP has a higher transmission rate comparatively.

Step 3:





The ‘d’ event occurs at:

d	6.07152	4	5	tcp	1040	-------	0	0	6	25	42
d	6.11312	4	5	tcp	1040	-------	0	0	6	27	44
d	6.15472	4	5	tcp	1040	-------	0	0	6	29	46
d	11.07152	4	5	tcp	1040	-------	0	1	7	25	131
d	11.11312	4	5	tcp	1040	-------	0	1	7	27	133
d	11.15472	4	5	tcp	1040	-------	0	1	7	29	135
d	16.07568	4	5	tcp	1040	-------	0	2	8	26	235
d	16.11728	4	5	tcp	1040	-------	0	2	8	28	237
d	16.15888	4	5	tcp	1040	-------	0	2	8	30	239
d	16.24208	4	5	tcp	1040	-------	0	0	6	51	242

d	19.73232	4	5	tcp	1040	-------	0	1	7	47	332
d	19.7656	4	5	tcp	1040	-------	0	1	7	48	333
d	21.21744	4	5	tcp	1040	-------	0	3	9	25	359
d	21.25904	4	5	tcp	1040	-------	0	3	9	27	361
d	21.30064	4	5	tcp	1040	-------	0	3	9	29	363
d	21.34224	4	5	tcp	1040	-------	0	1	7	49	365
d	22.66928	4	5	tcp	1040	-------	0	3	9	40	400
d	37.64944	4	5	tcp	1040	-------	0	2	8	106	876
d	38.06544	4	5	tcp	1040	-------	0	1	7	98	897
d	38.73104	4	5	tcp	1040	-------	0	0	6	164	930
d	39.06384	4	5	tcp	1040	-------	0	3	9	82	947
d	49.48464	4	5	tcp	1040	-------	0	3	9	128	1376
d	49.90064	4	5	tcp	1040	-------	0	2	8	171	1397
d	50.31664	4	5	tcp	1040	-------	0	1	7	155	1418
d	50.94064	4	5	tcp	1040	-------	0	0	6	248	1449
d	61.31984	4	5	tcp	1040	-------	0	0	6	308	1880
d	61.65264	4	5	tcp	1040	-------	0	3	9	185	1897
d	62.02704	4	5	tcp	1040	-------	0	2	8	236	1916
d	62.65104	4	5	tcp	1040	-------	0	1	7	222	1947
d	73.19664	4	5	tcp	1040	-------	0	1	7	275	2384
d	73.44624	4	5	tcp	1040	-------	0	0	6	373	2397
d	74.07024	4	5	tcp	1040	-------	0	3	9	252	2428
d	74.27824	4	5	tcp	1040	-------	0	2	8	301	2439
d	85.11504	4	5	tcp	1040	-------	0	1	7	330	2888
d	85.57264	4	5	tcp	1040	-------	0	0	6	438	2911

d	86.19664	4	5	tcp	1040	-------	0	3	9	317	2942
d	86.40464	4	5	tcp	1040	-------	0	2	8	366	2953
d	96.78384	4	5	tcp	1040	-------	0	2	8	419	3382
d	97.44944	4	5	tcp	1040	-------	0	1	7	397	3415
d	97.65744	4	5	tcp	1040	-------	0	0	6	503	3426
d	98.28144	4	5	tcp	1040	-------	0	3	9	382	3457

Observe first d event from source node 0

d	6.07152	4	5	tcp	1040	-------	0	0	6	25	42

The first drop occurs at 6.07512 seconds and the packet number is 42.The drop occurs when data is forwarded from node 4 to node 5 at time 6.07152 and got dropped at node 4 as the outgoing queue at node 4 was full and hence dropped.This drop is expected because of a 3 Duplicate Ack. As a result of this drop, we observe that cwnd changes from 20 to 10 at time 7.8 which means that the packet which was dropped at 6.07512 has never reached the receiver and the receiver repeatedly answers the same ACK numbers for next packets

Observe drop shown below

d	6.11312	4	5	tcp	1040	-------	0	0	6	27	44

The drop occurs at 6.11312 seconds and the packet number is 44.This drop is expected because of a timeout. As a result of this drop, we observe that cwnd changes from 10 to 1 at time 11.6 which means that the packet which was dropped at 6.11312 has never reached the receiver and also the source waited for TTL of 5.48688 (11.6-6.11312=5.48688) and it never received ACK for that packet.

Considering the throughput chart, we see that during the time interval 8 - 9, the throughput at the receiver falls to zero. That means very little data has been sent from the source.

For Source node 1

d	11.07152	4	5	tcp	1040	-------	0	1	7	25	131

The first drop occurs at 11.07152 seconds and the packet number is 131.The drop occurs when data is forwarded from node 4 to node 5 at time 11.07152 and got dropped at node 4 as the outgoing queue at node 4 was full and hence dropped.This drop is expected because of a 3 Duplicate Ack. As a result of this drop, we observe that cwnd changes from 20 to 10 at time 12.7 which means that the packet which was dropped at 11.07152 has never reached the receiver and the receiver repeatedly answers the same ACK numbers for next packets.

Observe other drop

d	11.15472	4	5	tcp	1040	-------	0	1	7	29	135

The drop occurs at 11.15472 seconds and the packet number is135 .This drop is expected because of a timeout. As a result of this drop, we observe that cwnd changes from 10 to 1 at time 16.6 which means that the packet which was dropped at 11.15472 has never reached the receiver and also the source waited for TTL of 5.45 (16.6-11.154=5.45) and it never received ACK for that packet.

Considering the throughput chart, we see that during the time interval 13-14, the throughput at the receiver falls to zero.which shows that very little data has been sent from the source.

3. At what time all flows reach a fair share of the available bandwidth? What is the average throughput for each flow during the time where all flows get a fair share of bandwidth? Calculate the averages in your Excel file. Provide a **discussion** of the fairness referring to the results you got from throughput and cwnd charts and also drop events in your trace.tr file. You may need to add more charts that enlarge the scale in some part of a chart or separate some flow from other flows to clarify your discussion. Refer to the book, section 3.7.1.

Referring to the book ,section 3.7.1 we see that if we consider K TCP connections and, each with an alternate start to finish way, however all going through a bottleneck interface with transmission rate R bps. (By bottleneck interface, we imply that for every association, the various connections along the association's way are not congested and have abundant transmission limit as contrasted and the transmission limit of the bottleneck interface.) Suppose every association is moving an enormous document and there is no UDP traffic going through the bottleneck connect. A congestion control mechanism is said to be Fair if the normal transmission pace(average transmission rate) of every association is around R/K; that is, every connection gets an equivalent portion of the connection data transfer capacity.

According to our observations,

The time all flows reach a fair share of the available bandwidth is 52.25

the average throughput for each flow during the time where all flows get a fair share of bandwidth are

Flow 1 :- 0.04081066667

Flow 2 :- 0.03225295238

Flow 3 :- 0.03431314286

Flow 4 :- 0.03106438095

According to the charts which we get for cwnd and throughput for all the four flows are similar and also the average throughputs are almost equal and nearly equal to R/K Which proves that TCP is fair in this scenario.