<u>Assignment 2</u>
<u>Name: Akshit Verma   Roll No: 200091</u>

Question1:
TCP Lab

1. Source IP: 192.168.1.102:1161
2. Destination IP: 128.119.245.12:80
3. IP: 192.168.1.102:1161
4. 0
5. 0
6. 1
7. Sequence numbers : 1, 566, 2026, 3486, 4946, 6406

	Sent TIme	**ACK Received Time**	**RTT (seconds)**
Segment 1	0.026477	0.53937	0.02746
Segment 2	0.041737	0.077294	0.035557
Segment 3	0.054026	0.124085	0.070059
Segment 4	0.054690	0.169118	0.11443
Segment 5	0.077405	0.217299	0.13989
Segment 6	0.078157	0.267802	0.18964

<u>EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT</u>

Segment 1:
EstimatedRTT = RTT for Segment 1 = 0.02746 second

Segment 2:
EstimatedRTT = *0.875 * 0.02746 + 0.125 * 0.035557 = 0.0285*

Segment 3:
EstimatedRTT = *0.875 * 0.0285 + 0.125 * 0.070059 = 0.0337*

Segment 4:
EstimatedRTT = *0.875 * 0.0337+ 0.125 * 0.11443 = 0.0438*

Segment 5:
EstimatedRTT = *0.875 * 0.0438 + 0.125 * 0.13989 = 0.0558*

Segment 6:
EstimatedRTT = *0.875 * 0.0558 + 0.125 * 0.18964 = 0.0725 second*

8. First Packet: 565
   Rest 5: 1460
9. Min buffer advertised by gaia.cs.umass.edu for the entire trace is 5840 bytes, which shows in the first acknowledgement from the server. This receiver window grows steadily until a maximum receiver buffer size of 62780 bytes which is never throttled.
10. The trace file contains no retransmitted portions. We may examine the sequence numbers of the TCP segments in the trace file to confirm this. All sequence numbers from the source (192.168.1.102) to the destination (128.119.245.12) increase monotonically with respect to time in this trace's Time-Sequence-Graph (Stevens). If a segment is retransmitted, its sequence number should be less than the sequence numbers of its nearby segments.
11. The difference between the acknowledged sequence numbers of two consecutive ACKs indicates the data received by the server between these two ACKs.

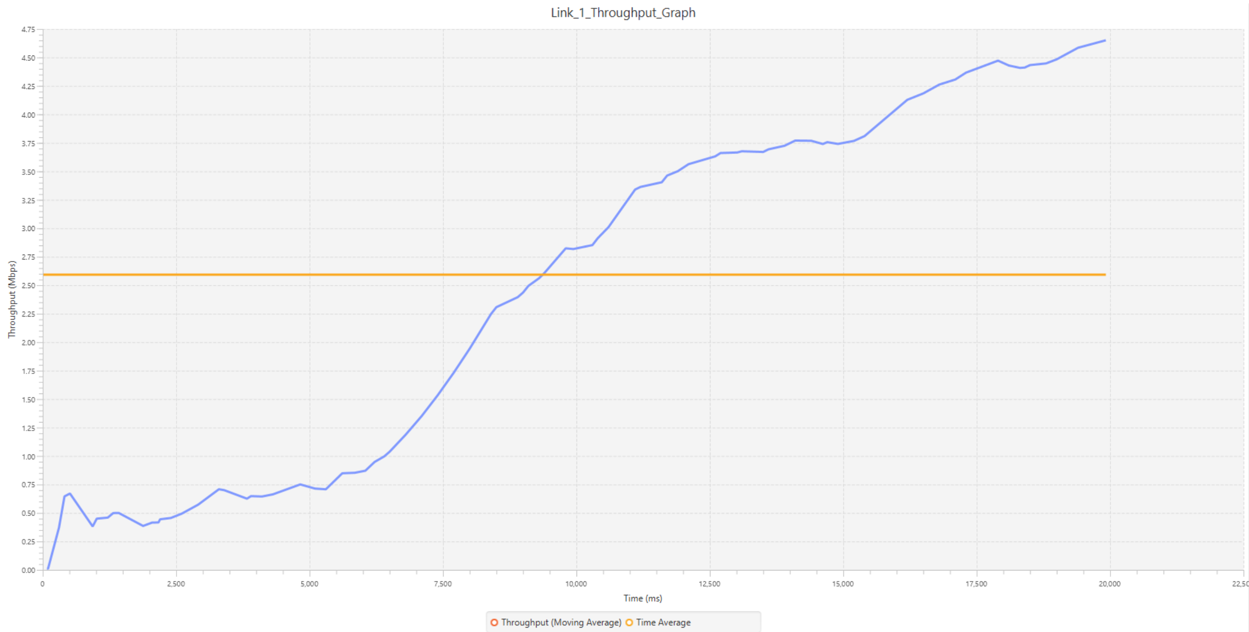
	ACKd sequence number	ACkd data
ACK 1	566	566
ACK 2	2026	1460
ACK 3	3486	1460
ACK 4	4946	1460
ACK 5	6046	1460
ACK 6	7866	1460
ACK 7	9013	1460
ACK 8	10473	1460
ACK 9	11933	1460
ACK 10	13393	1460
ACK 11	14853	1460
ACK 12	16313	1460

12. TCP throughput computation is heavily influenced by the average time period chosen. In this question, we use the average time period as the entire connection duration as a conventional throughput computation. The average throughput for this TCP connection is then calculated as the ratio of total data to total transmission time. The difference between the sequence number of the first TCP segment (i.e. 1 byte for No. 4 segment) and the acknowledged sequence number of the last ACK (164091 bytes for No. 202 segment) can be used to calculate the total amount of data sent. As a result, the total data is 164091 - 1 = 164090 bytes. The total transmission time is the difference between the time of the first TCP segment (i.e., 0.026477 second for No. 4 segment) and the time of the last ACK (i.e., 5.455830 second for No. 202 segment). The total transmission time is thus 5.455830 - 0.026477 = 5.4294 seconds. As a result, the TCP connection's throughput is calculated as 164090/5.4294 = 30.222 KBytes/sec.
13.

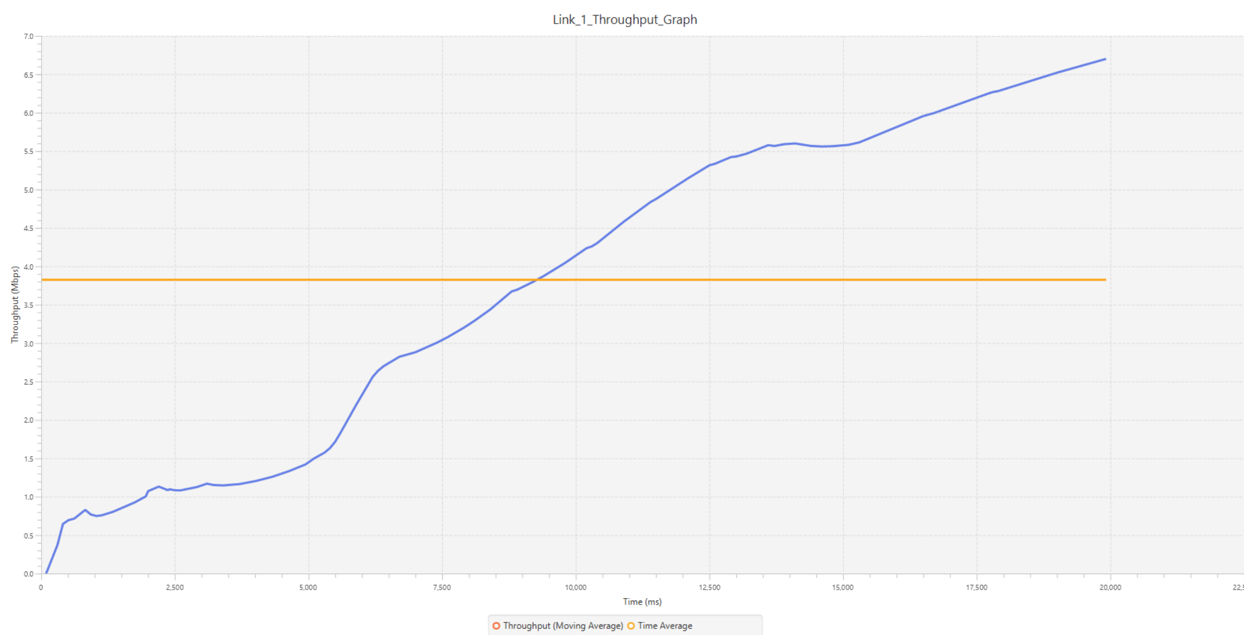
UDP Lab

1. The UDP header contains 4 fields: source port, destination port, length, and checksum.
2. Each of the UDP header fields is 2 bytes long.
3. The value in the length field is the sum of the 8 header bytes, plus the 42 encapsulated data bytes.
4. The maximum number of bytes that can be included in a UDP payload is 216 – 1 less the header bytes. This gives 65535 – 8 = 65527 bytes.
5. The largest possible source port number is $2^{16} - 1 = 65535$.
6. The IP protocol number for UDP is 0x11 hex, which is 17 in decimal value.
7. The source port of the UDP packet sent by the host is the same as the destination port of the reply packet, and conversely the destination port of the UDP packet sent by the host is the same as the source port of the reply packet.

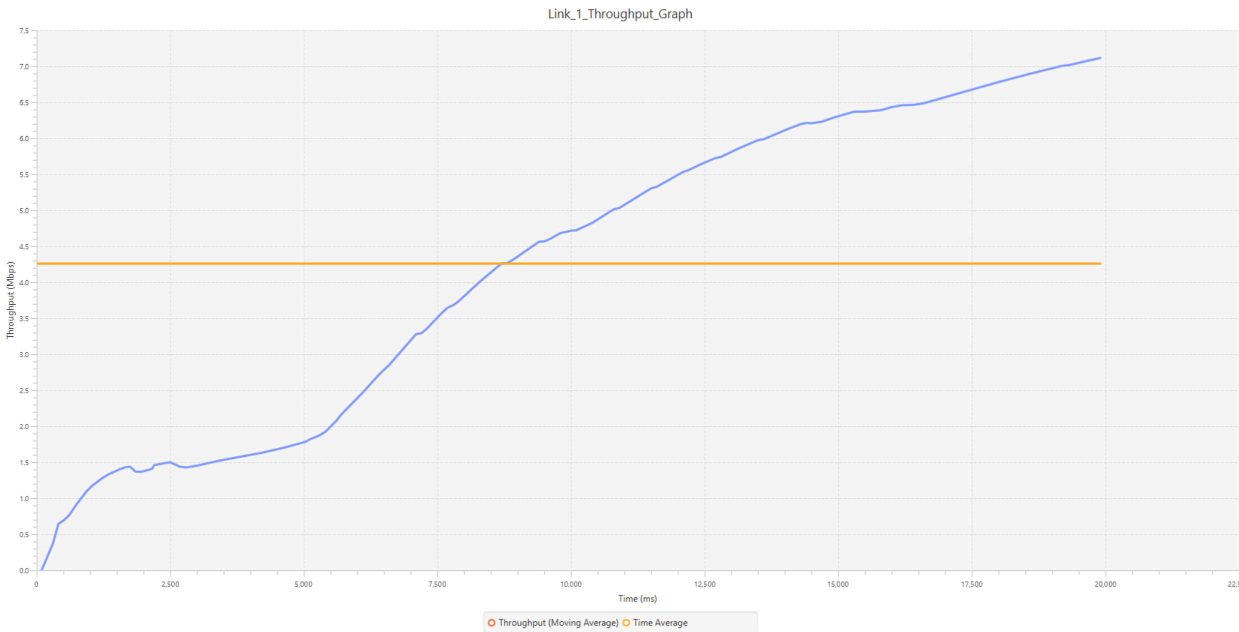
Question2:



OLD TAHOE



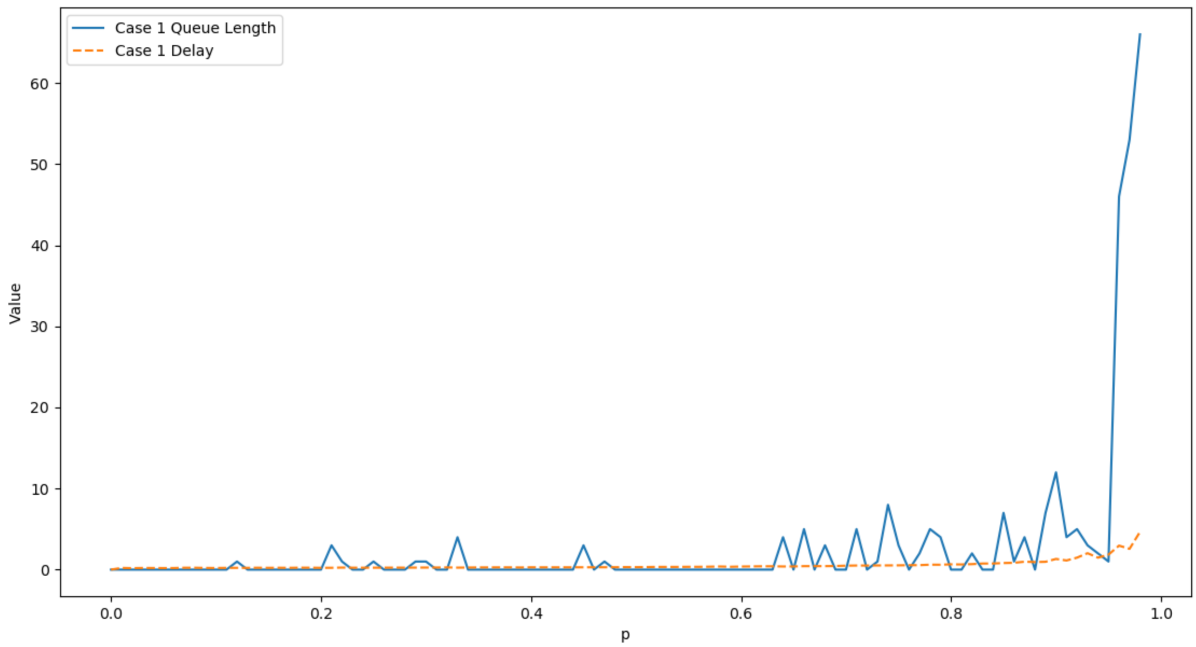
RENO

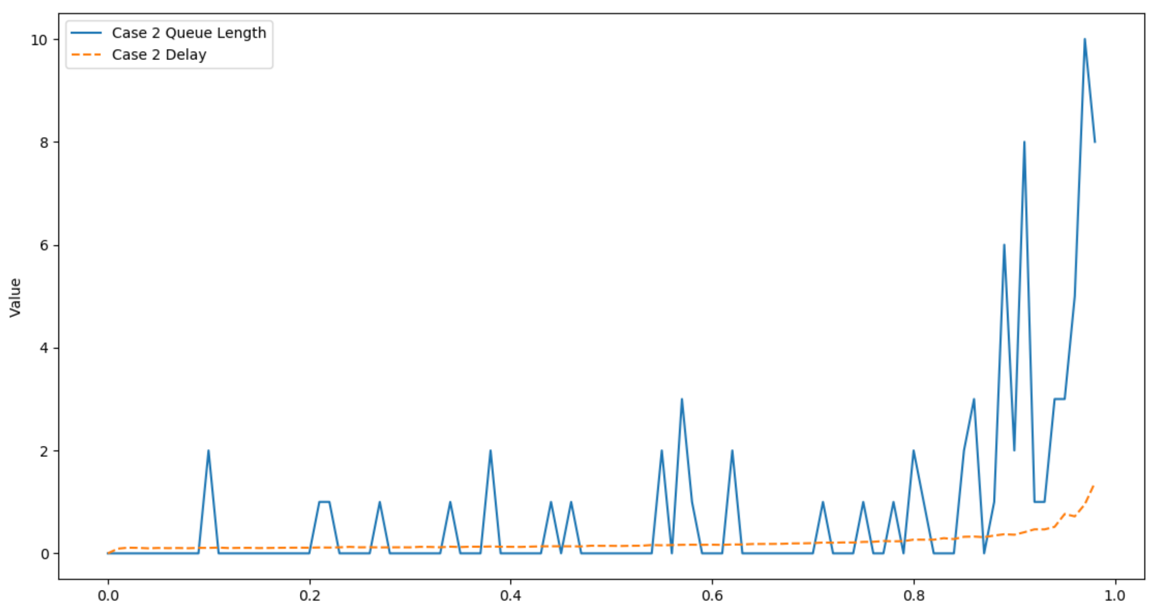


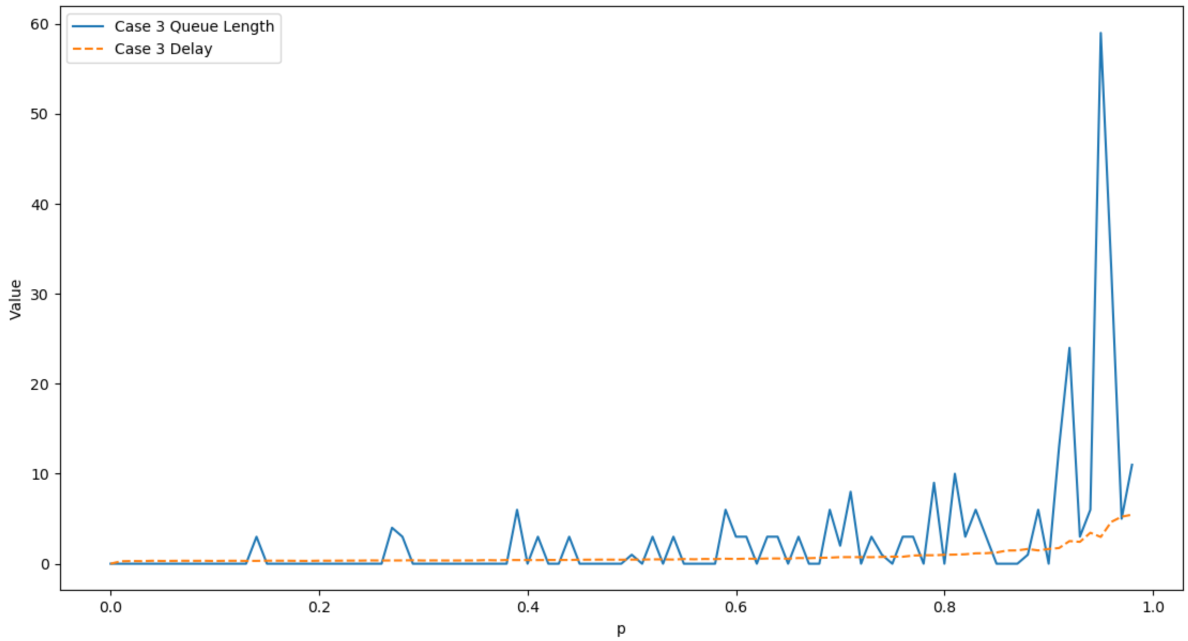
Cubic

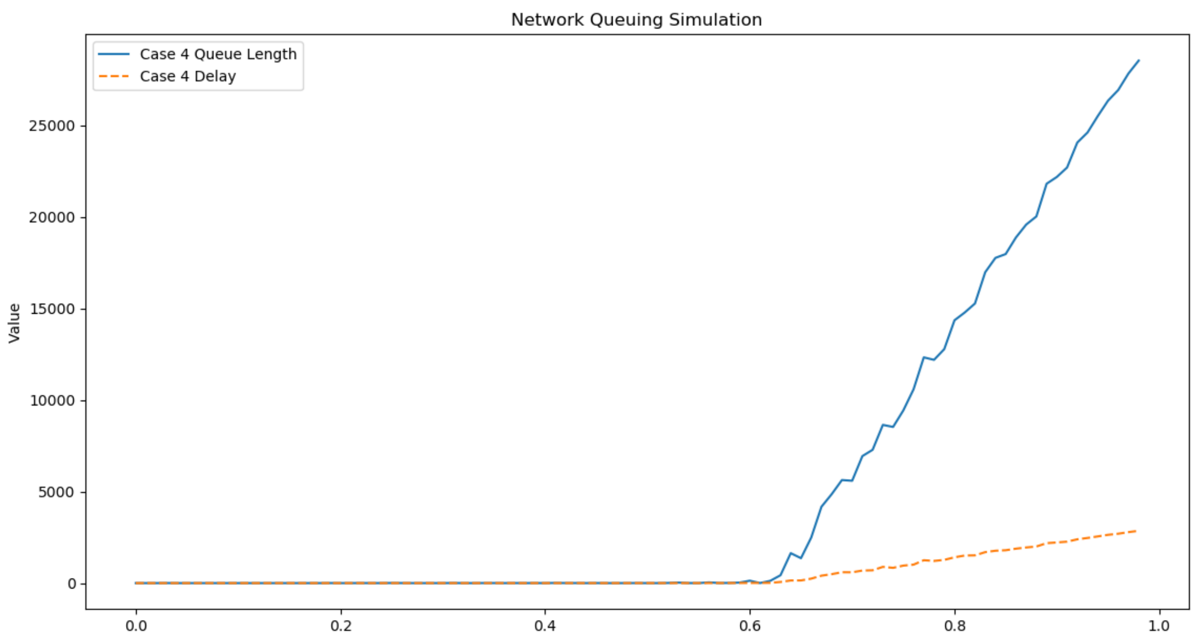
As seen from the figure, the maximum link utilization is done by TCP cubic congestion control algorithm. It provides the highest average throughput of all the 3 algorithms. The cubic hyperparameters can be tweaked to get a higher throughput.

Question3:
Part A)
The following are the graphs of queue size (in Bytes) vs p and average packet delay (in seconds) vs p.









Part B)

$E[X] = p \times (p_a \times a + p_b \times b + p_c \times c + p_d \times d)$
$E[X^2] = p \times (p_a \times a^2 + p_b \times b^2 + p_c \times c^2 + p_d \times d^2)$
$Var(X) = E[X^2] - E[X]^2$

1. **Case 1** $[p_a = p_b = p_c = p_d = 0.25]$
   $E[X] = p\,(0.25 \times (2+4+6+8))$
   $E[X] = 5p$

$E[X^2] = p\ (0.25 \times (4+16+36+64))$
$E[X^2] = 30p$

$Var(X) = 30p - 25p^2$

2. **Case 2** $[p_a = p_d = 0,\ p_b = p_c = 0.5]$
   $E[X] = p\ (0.5 \times (4+6))$
   $E[X] = 5p$

   $E[X^2] = p\ (0.5 \times (16+36))$
   $E[X^2] = 26p$

   $Var(X) = 26p - 25p^2$

3. **Case 3** $[p_a = p_d = 0.5,\ p_b = p_c = 0]$
   $E[X] = p\ (0.5 \times (2+8))$
   $E[X] = 5p$

   $E[X^2] = p\ (0.5 \times (4+64))$
   $E[X^2] = 34p$

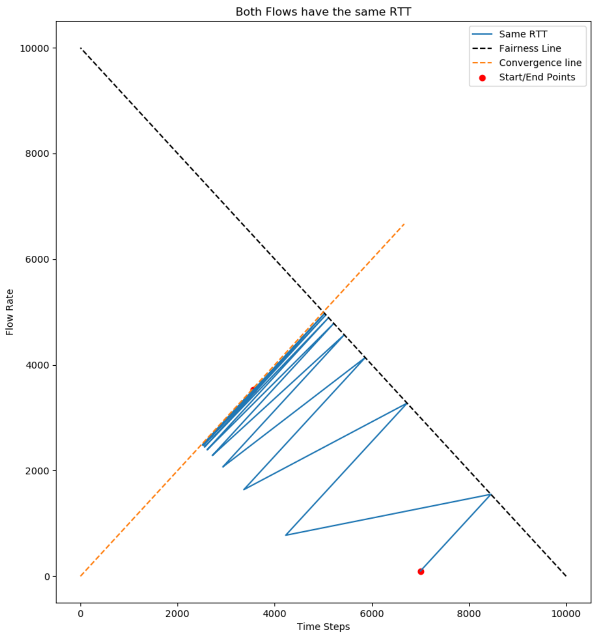
   $Var(X) = 34p - 25p^2$
4. **Case 4** $[p_a = p_b = p_c = 0, = p_d = 1]$
   $E[X] = p\ (1 \times (10))$
   $E[X] = 10p$

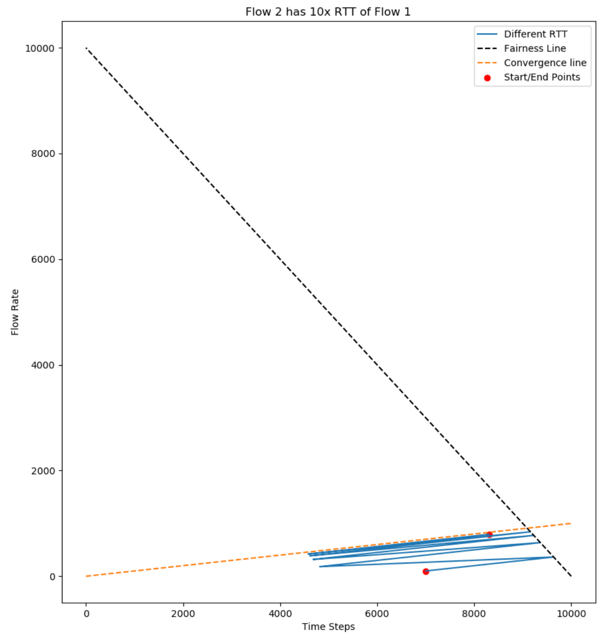
   $E[X^2] = p\ (1 \times (64))$
   $E[X^2] = 64p$

   $Var(X) = 64p - 100p^2$

As seen from the expectation values, the incoming communication traffic exceeds the network capacity for p = 1.

Question4:





These are the plots for the 2 cases. There is a difference in the line on which it converges. The slope of the line of convergence depends on the ratio of RTT of the clients (1 and 10 in this case). Approximately 16,000 timestamps are required for the convergence of both the cases.