i) *For each of the TCP flavors (VEGAS and SACK) simulate the 3 cases (of RTT) and find the ratio of the average throughput of src1 to src2. Make two separate tables (one for each TCP flavor)*

For TCP Vegas

Case	RTT Ratio	Throughput at Sink1(bps)	Throughput at Sink2(bps)	Throughput Ratio
1	1:2	583328	416664	1.4:1
2	1:3	687488	312504	2.2:1
3	1:4	750000	250000	3:1

For TCP SACK

Case	RTT ratio	Throughput at Sink1(bps)	Throughput at Sink2(bps)	Throughput Ratio
1	1:2	523768	476176	1.1:1
2	1:3	545456	454544	1.2:1
3	1:4	565232	434768	1.3:1

ii) *Comment on the relationship between TCP throughput and RTT in light of these results for each TCP flavor. Later compare and comment about the 2 flavors of TCP. Which performs better (in terms of throughput) and why? (you can compare for just case 1, when RTT of 2 sources are in 1:2 ratio).*

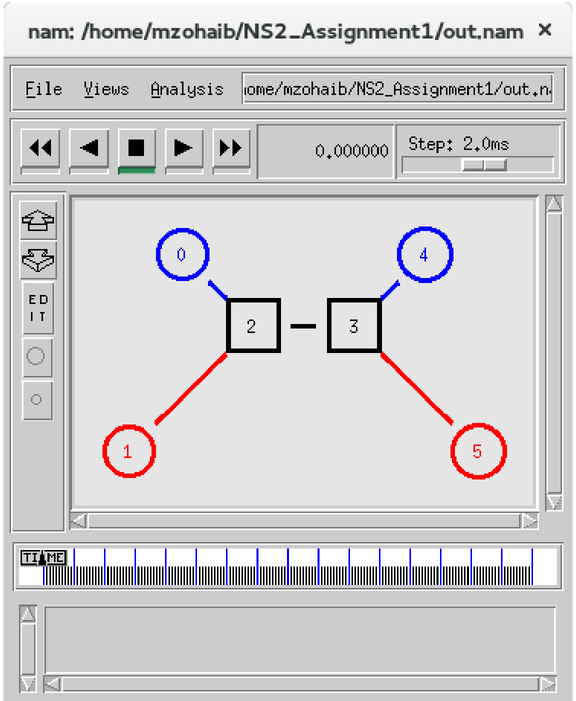
For TCP Vegas,

In case of src1 and rcv1, as end-to-end delay of the other pair is increased, throughput at rcv1 increases. But throughput at rcv2 decreases with increase of delay. TCP Vegas is **congestion avoidance algorithm**. The sources decrease their sending rate based on estimated and actual rate calculations, before actual packet drops and congestion take place at the router. Since RTT for second pair is larger in all cases compared to first pair, it takes long time for packets of src2 to reach buffer as compared to src1. This means packets of src1 occupy buffer at router more compared to src2. Src2 infers congestion more than src2, so it decreases its sending rate. As a result, throughput at rcv2 decreases. The sum of throughput at rcv2 and rcv1 should be around the capacity of bottleneck 1Mbps link (R1-R2), so consequently src1 increases its rate to fill the pipe and throughput at rcv1 increases. The results are more pronounced when delays are large for second pair as in case 3 compared to case 2. Therefore, as ratio of end-to-end delay of src1-rcv1 and src2-rcv2 changes from 1:2 to 1:2 to 1:4, the ratio of throughput or sending rate changes from 1.4:1 to 2.2:1 to 3:1.

For TCP SACK,

In case of src1 and rcv1, as end-to-end delay of the other pair is increased, throughput at rcv1 increases but not in much significant amount compared to Vegas. But throughput at rcv2 decreases with increase of delay. The decrease is also not that significant as compared to case of Vegas. TCP SACK does not use congestion avoidance mechanism, so no source decreases its sending rate or congestion window before a packet drop. Packets of src2 take longer time to reach the router and hence occupy less buffer space compared to packets from src1. Hence sending rate at src1 is more. As delay at src2-rcv2 pair increases, sending rate at src1 increases and hence throughput at rcv1 increases and at rcv2 decreases. The sum of throughputs is the same as capacity of R1-R2 link as in the previous case.

Setup:



There are two logical flows, one from src1 to rcv1 and another from src2 to rcv2.

In the first case, delay from src1 to R1 is 5ms, delay from R1 to R2 is 5ms and delay from R2 to rcv1 is 5ms. So total delay is 15ms. Similarly delay from src2 to R1 is 12.5ms, delay from R1 to R2 is 5ms and delay from R2 to rcv2 is 12.5ms. So total delay is 30ms. The ratio of end-to-end RTT for src1-rcv1 pair and src2-rcv2 pair is 1:2.

In the second case, delay from src1 to R1 is 5ms, delay from R1 to R2 is 5ms and delay from R2 to rcv1 is 5ms. So total delay is 15ms. Similarly delay from src2 to R1 is 20ms, delay from R1 to R2 is 5ms and delay from R2 to rcv2 is 20ms. So total delay is 45ms. The ratio of end-to-end RTT for src1-rcv1 pair and src2-rcv2 pair is 1:3.

In the first case, delay from src1 to R1 is 5ms, delay from R1 to R2 is 5ms and delay from R2 to rcv1 is 5ms. So total delay is 15ms. Similarly delay from src2 to R1 is 27.5ms, delay from R1 to R2 is 5ms and delay from R2 to rcv2 is 27.5ms. So total delay is 60ms. The ratio of end-to-end RTT for src1-rcv1 pair and src2-rcv2 pair is 1:3.

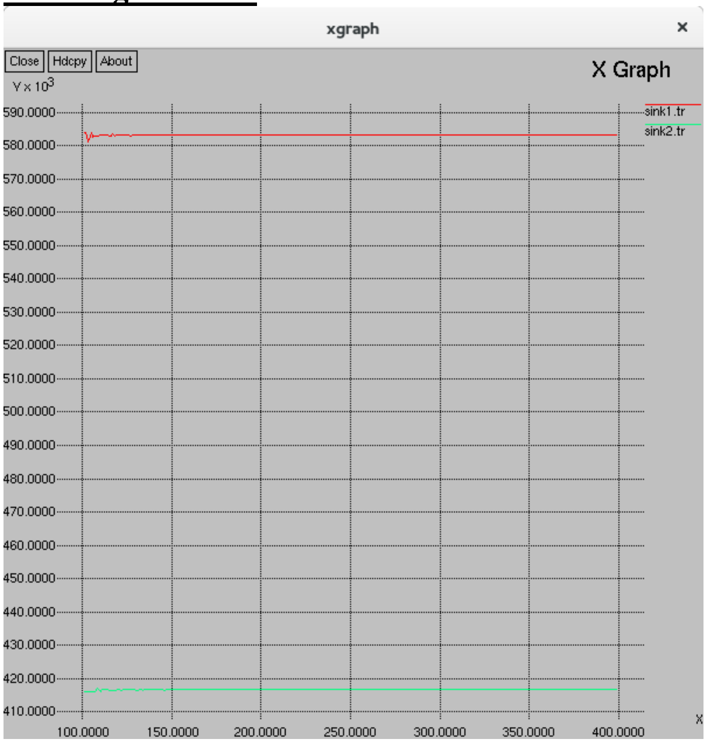
Procedure:

First 100 seconds of simulation are ignored so that transients may die down and we reach steady state. Results are recorded starting from t=100 to 400 sec. Both sources start sending packets to their respective sinks at t=100 sec and stop at 400 sec. Every 10 seconds starting from t=100 sec, both the sinks calculate the number of bytes received and subtract number of bytes received until t=100 from this value to calculate number of bytes received from t=100 to that time. Then these bytes are divided by time interval from 100 to this time to calculate throughput until this time.

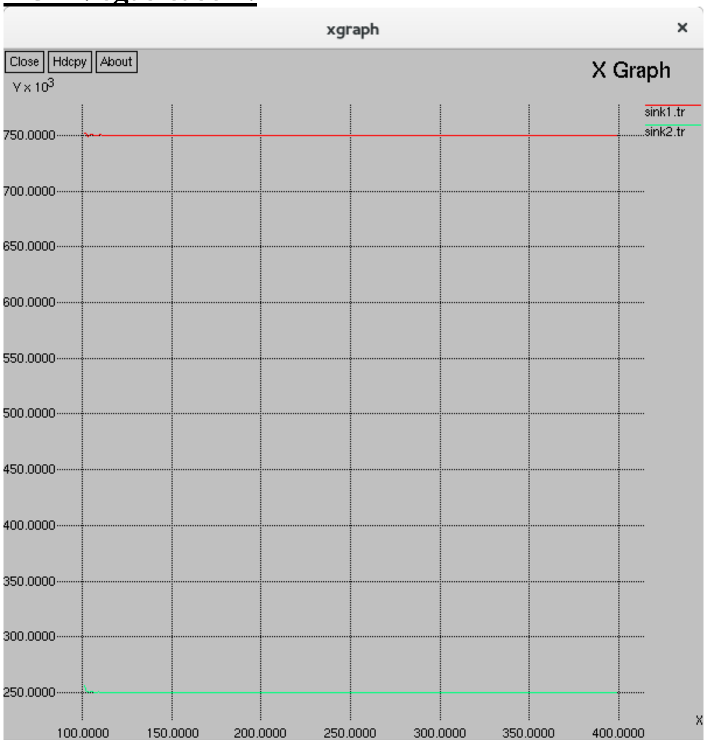
To calculate the ratio of throughput at sink1 to throughput at sink2, the value obtained at 400 seconds is used i.e. the total bytes received from 100 to 400 sec divided by 300 sec. This throughput is then converted to bits per second by multiplying it with 8 as the cumulative data received at a sink is in bytes.

Throughput vs time graphs:

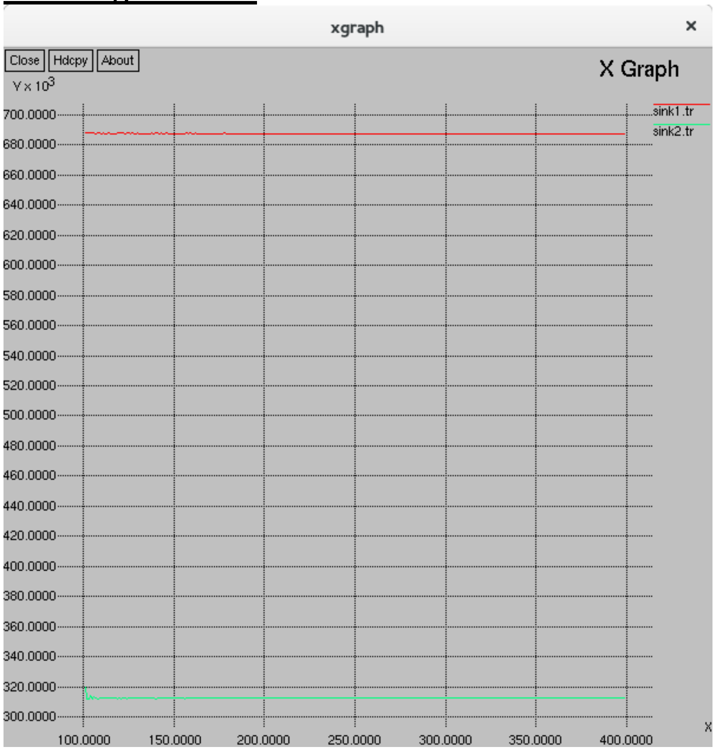
TCP Vegas case 1:



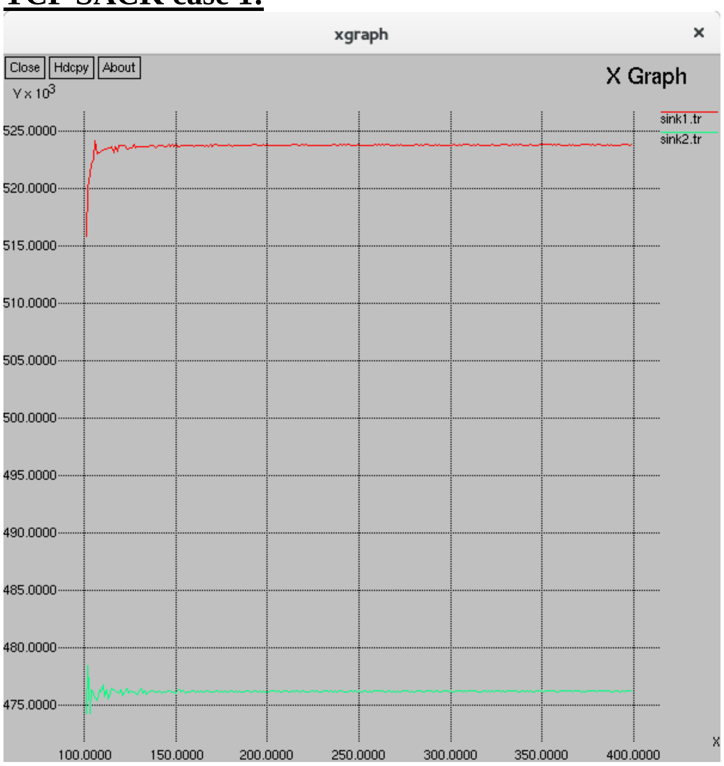
TCP Vegas case 2:



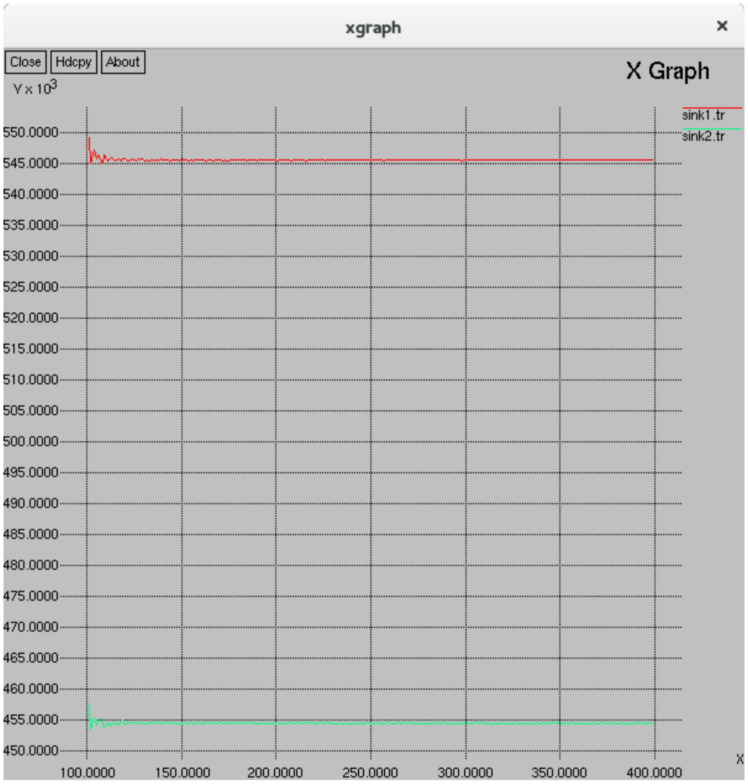
TCP Vegas case 3:



TCP SACK case 1:



TCP SACK case 2:



TCP SACK case 3: