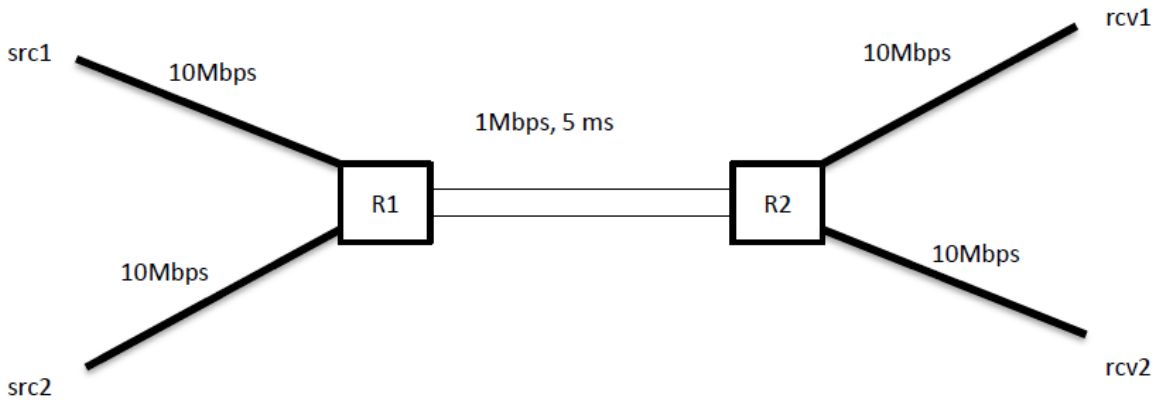
NS-2 simulation report
Team11
Srividhya Balaji : UIN : 827007169
Sanjana Srinivasan: UIN: 927008860

Individual Contribution :

1. Srividhya did coding, testing and documentation
2. Sanjana did coding, testing and documentation

In this assignment, NS-2 simulator has been used to build a simultaion of the following topology and configuration :

- Two routers R1, R2 connected with a 1Mbps link and having 5ms latency
- Two source/senders src1, src2 connected to R1 with 10Mbps links
- Two receivers rcv1, rcv2 connected to R2 with 10Mbps links
- Application sender is FTP over TCP



(1) Test Setup
The variable parameter for the three cases are :

Case 1:

- src1-R1 and R2-rcv1 end-2-end delay = 5 ms
- src2-R1 and R2-rcv2 end-2-end delay = 12.5 ms

Case 2:

- src1-R1 and R2-rcv1 end-2-end delay = 5 ms
- src2-R1 and R2-rcv2 end-2-end delay = 20 ms

Case 3:

- src1-R1 and R2-rcv1 end-2-end delay = 5 ms
- src2-R1 and R2-rcv2 end-2-end delay = 27.5 ms

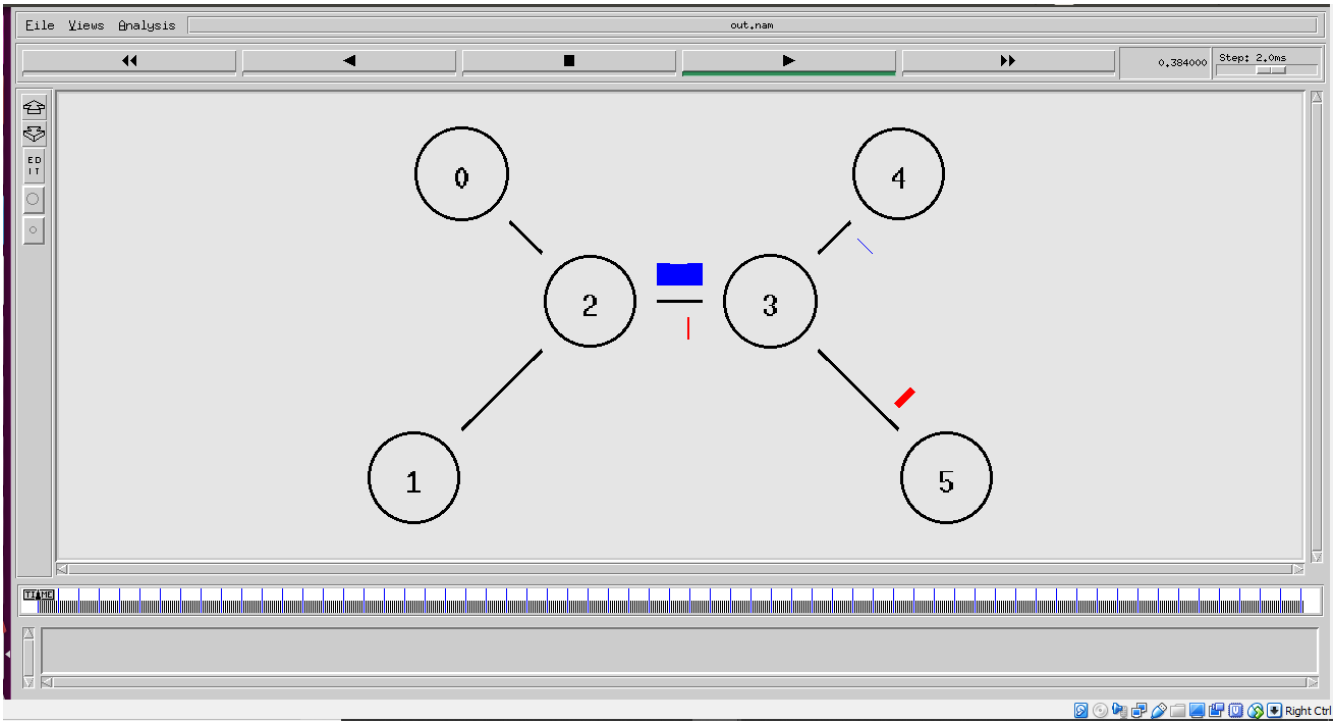
The code can be found in the readme file.

(2) Test procedure

After the code is finished, it is debugged and simulation is run using the command :

ns ns.tcl <flavor> <case>

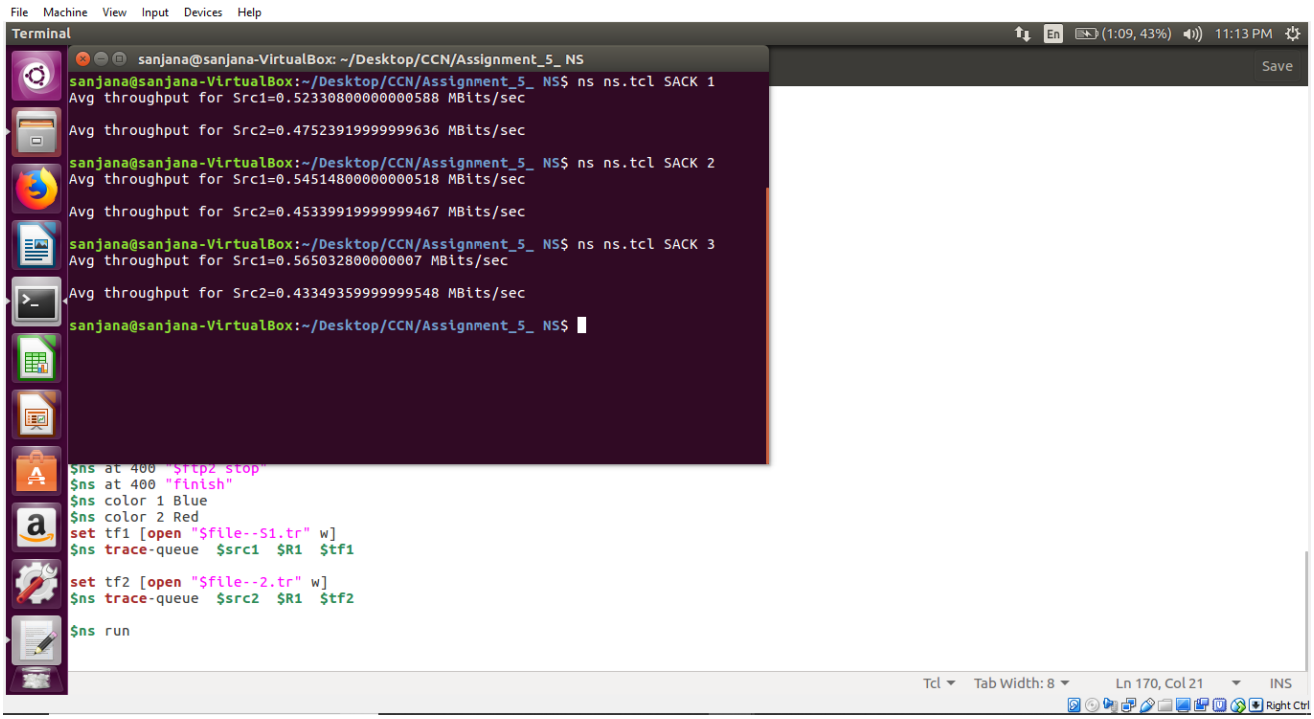
The simulation of the topology is seen through the nam.out file



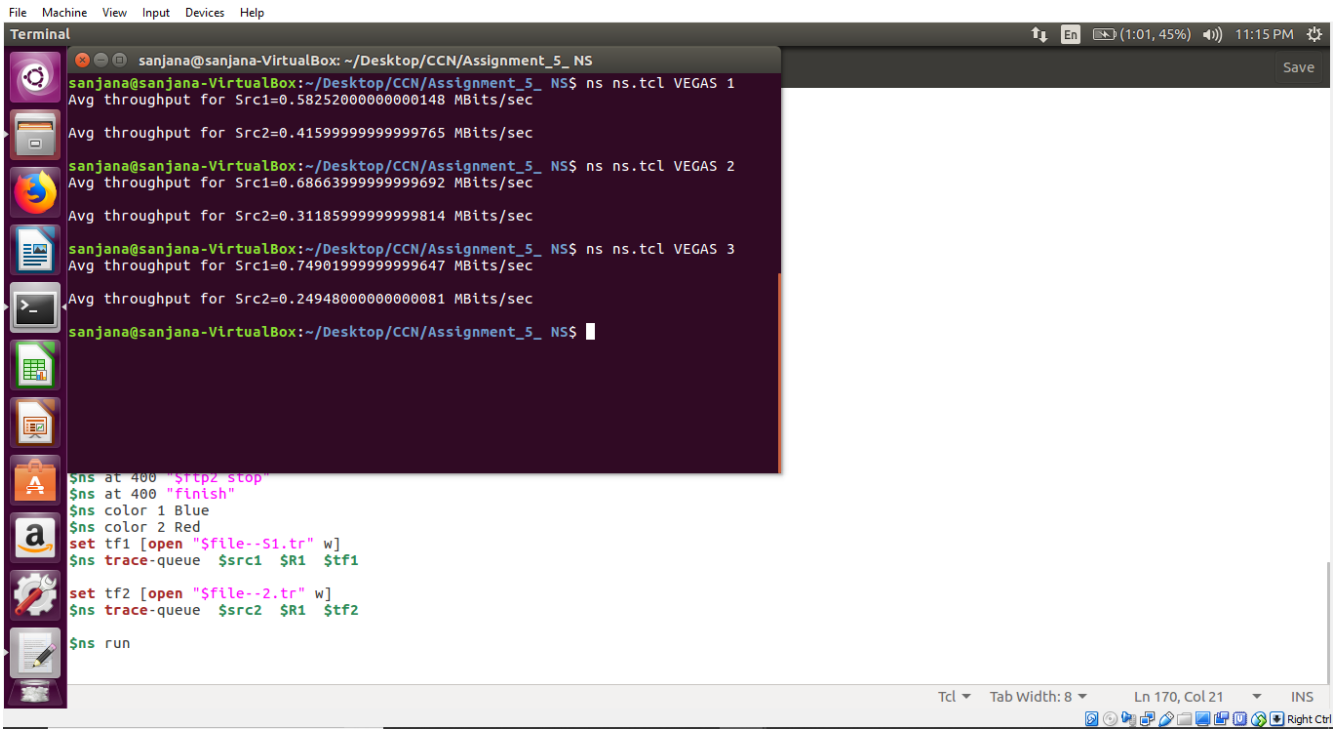
The detailed output can be viewed in out_flavor-case.tr

(3) Test case results :

- For flavor – SACK cases 1,2,3 :



- For flavor – VEGAS cases 1,2,3 :



4) Throughput ratio and explanation

(i) After running all six cases, the simulation results and the comparisons are listed below :

TCP_flavor Case	Throughput src1 Mbps	Throughput src2 Mbps	Ratio of throughputs
SACK 1	0.523308	0.475239	1.10114
SACK 2	0.545148	0.453399	1.20235
SACK 3	0.565032	0.433493	1.30343
VEGAS 1	0.582520	0.415999	1.32817
VEGAS 2	0.686639	0.311859	2.20176
VEGAS 3	0.749019	0.249480	3.00232

From the above table, it can be observed that when the RTT of the links are varied in the ration 1:2, 1:3, 1:4 in the three cases under consideration, only TCP VEGAS undergoes obvious changes while TCP SACK remains almost a constant throughout.

This significant influence of RTT in the throughput for TCP VEGAS implies that TCP VEGAS can have a bigger throughput when delay is small.
It can be seen from the simulation results, that for case 1 TCP VEGAS throughput (1.3034) is slightly higher than TCP SACK (1.1011). And in all cases, TCP Vegas outperforms SACK.

TCP Vegas outperform SACK with a better utilization of bandwidth and lesser congestion. There are a few reasons for the better performance of VEGAS

1. TCP VEGAS is more stable than SACK. The reason being that, SACK uses packet loss to denote congestion, so the sender continuously increases the sending rate until there is congestion and then they cut back. This cycle continues causing the system to keep oscillating. On the other hand TCP VEGAS flattens out the sending rate at the optimal bandwidth utilization point thus inducing stability in the system.
2. TCP VEGAS has a good estimation of incipient congestion and efficient estimation of congestion by measuring change in throughput rather than packet loss.