ECEN 602 HW 5
Network Simulation Assignment

Team Number 8

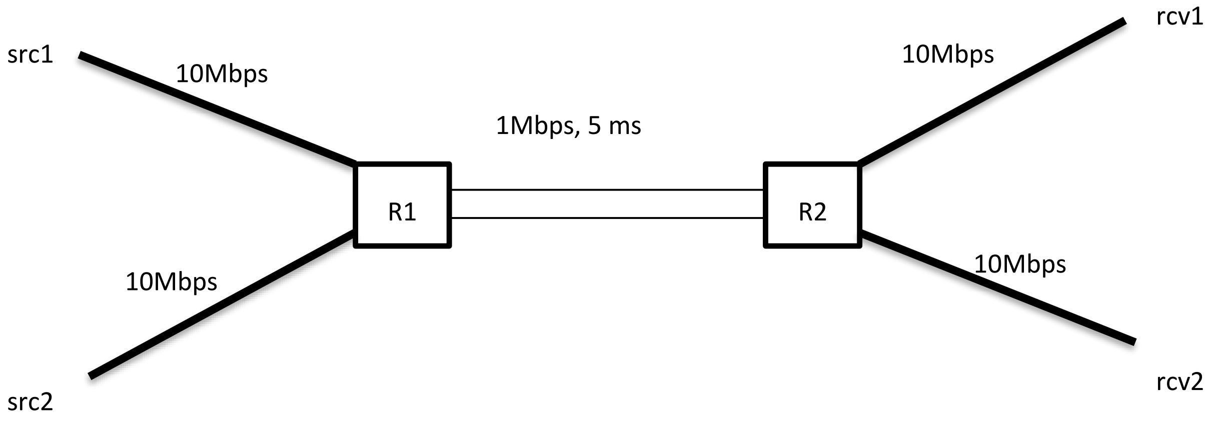
1. Garuda Suma Pranavi (UIN 926009146)
2. Dedeepya Venigalla (UIN 726006161)

We have both equally contributed in developing the NS-2 assignment.

1. Test Setup:

We have built the following network topology with the configuration details as follows:

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



We test the network for 2 TCP versions: TCP Sack and TCP Vegas

<u>Case 1</u>:

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

<u>Case 2</u>:

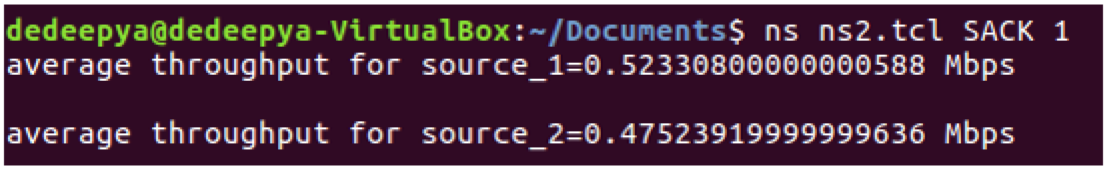
- src1-R1 and R2-rcv1 end-2-end delay = 5 ms
- src2-R1 and R2-rcv2 end-2-end delay = 20 ms

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

2. Test Procedure:

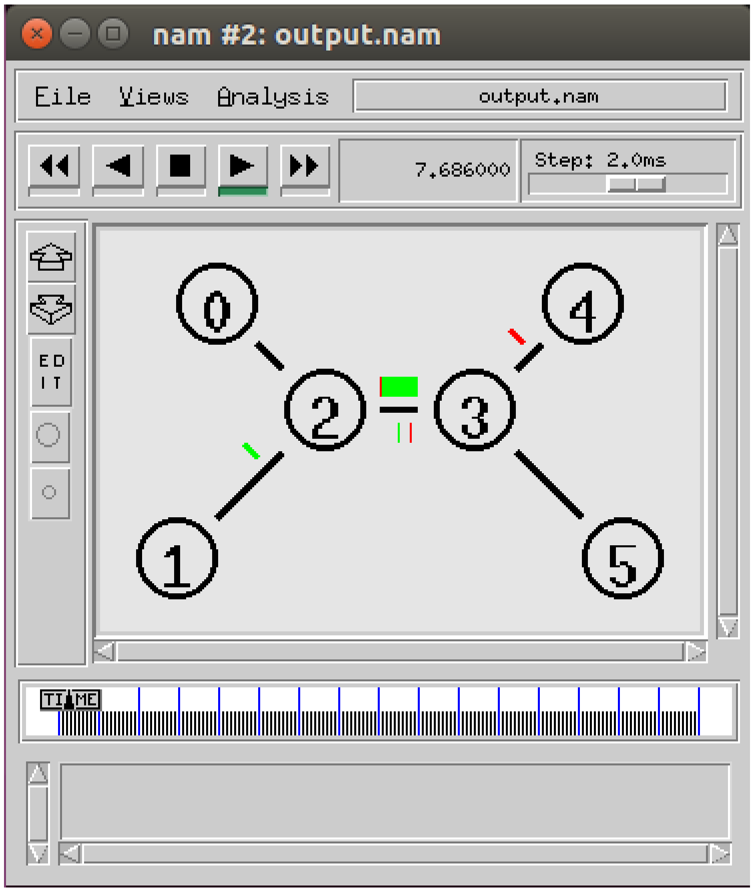
After writing the code, we debug the same through the tcl command script and run using: ns ns2.tcl <flavor> <case> to start our simulation.

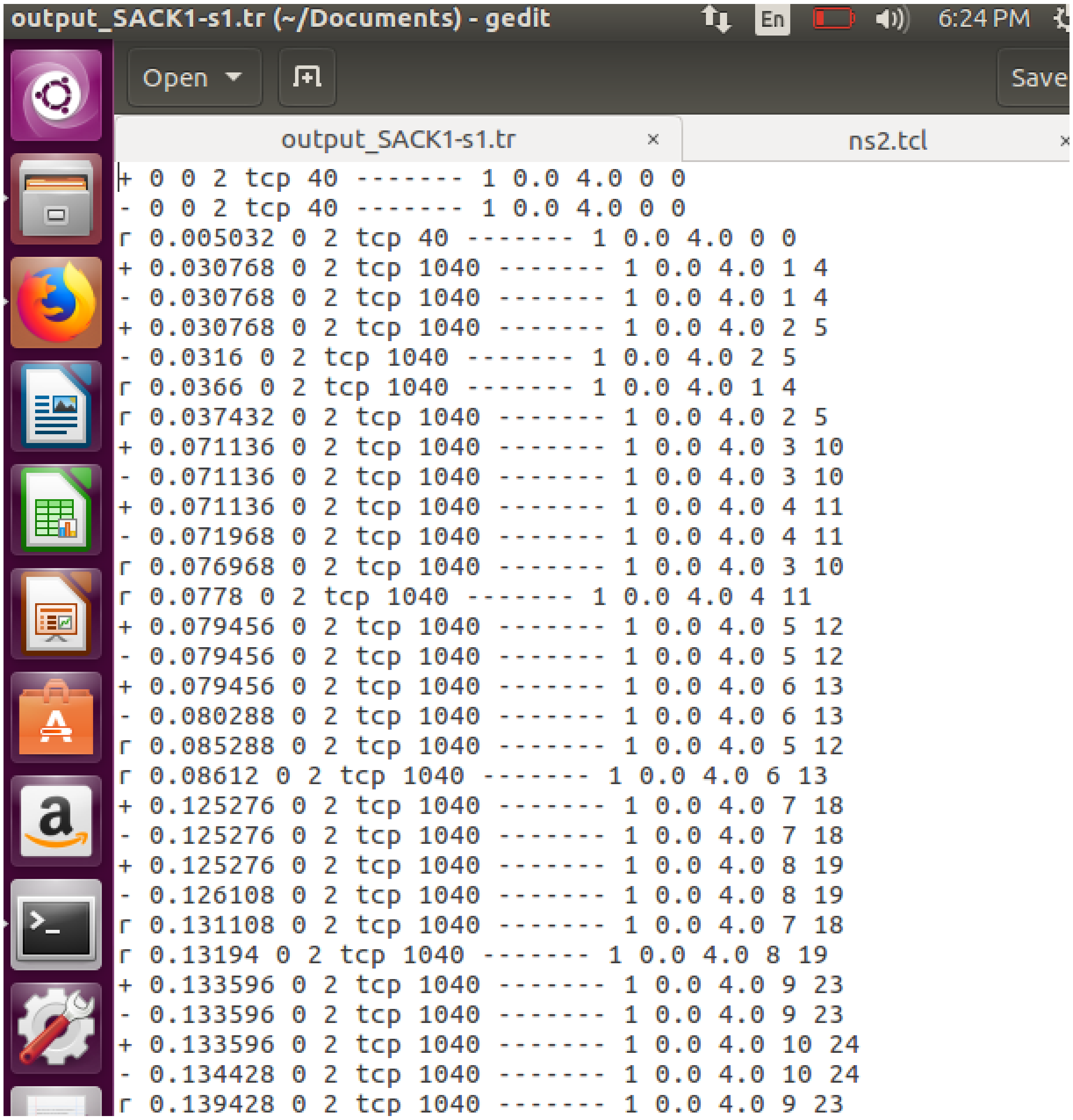


```
dedeepya@dedeepya-VirtualBox:~/Documents$ ns ns2.tcl SACK 1
average throughput for source_1=0.5233080000000588 Mbps

average throughput for source_2=0.4752391999999636 Mbps
```

Running the nam to show the packet flow in the simulation process:

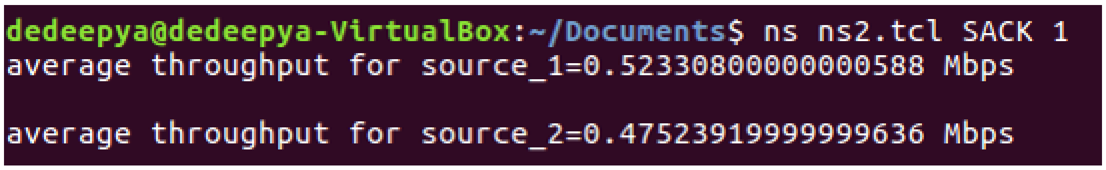




```
+ 0 0 2 tcp 40 ------- 1 0.0 4.0 0 0
- 0 0 2 tcp 40 ------- 1 0.0 4.0 0 0
r 0.005032 0 2 tcp 40 ------- 1 0.0 4.0 0 0
+ 0.030768 0 2 tcp 1040 ------- 1 0.0 4.0 1 4
- 0.030768 0 2 tcp 1040 ------- 1 0.0 4.0 1 4
+ 0.030768 0 2 tcp 1040 ------- 1 0.0 4.0 2 5
- 0.0316 0 2 tcp 1040 ------- 1 0.0 4.0 2 5
r 0.0366 0 2 tcp 1040 ------- 1 0.0 4.0 1 4
r 0.037432 0 2 tcp 1040 ------- 1 0.0 4.0 2 5
+ 0.071136 0 2 tcp 1040 ------- 1 0.0 4.0 3 10
- 0.071136 0 2 tcp 1040 ------- 1 0.0 4.0 3 10
+ 0.071136 0 2 tcp 1040 ------- 1 0.0 4.0 4 11
- 0.071968 0 2 tcp 1040 ------- 1 0.0 4.0 4 11
r 0.076968 0 2 tcp 1040 ------- 1 0.0 4.0 3 10
r 0.0778 0 2 tcp 1040 ------- 1 0.0 4.0 4 11
+ 0.079456 0 2 tcp 1040 ------- 1 0.0 4.0 5 12
- 0.079456 0 2 tcp 1040 ------- 1 0.0 4.0 5 12
+ 0.079456 0 2 tcp 1040 ------- 1 0.0 4.0 6 13
- 0.080288 0 2 tcp 1040 ------- 1 0.0 4.0 6 13
r 0.085288 0 2 tcp 1040 ------- 1 0.0 4.0 5 12
r 0.08612 0 2 tcp 1040 ------- 1 0.0 4.0 6 13
+ 0.125276 0 2 tcp 1040 ------- 1 0.0 4.0 7 18
- 0.125276 0 2 tcp 1040 ------- 1 0.0 4.0 7 18
+ 0.125276 0 2 tcp 1040 ------- 1 0.0 4.0 8 19
- 0.126108 0 2 tcp 1040 ------- 1 0.0 4.0 8 19
r 0.131108 0 2 tcp 1040 ------- 1 0.0 4.0 7 18
r 0.13194 0 2 tcp 1040 ------- 1 0.0 4.0 8 19
+ 0.133596 0 2 tcp 1040 ------- 1 0.0 4.0 9 23
- 0.133596 0 2 tcp 1040 ------- 1 0.0 4.0 9 23
+ 0.133596 0 2 tcp 1040 ------- 1 0.0 4.0 10 24
- 0.134428 0 2 tcp 1040 ------- 1 0.0 4.0 10 24
r 0.139428 0 2 tcp 1040 ------- 1 0.0 4.0 9 23
```

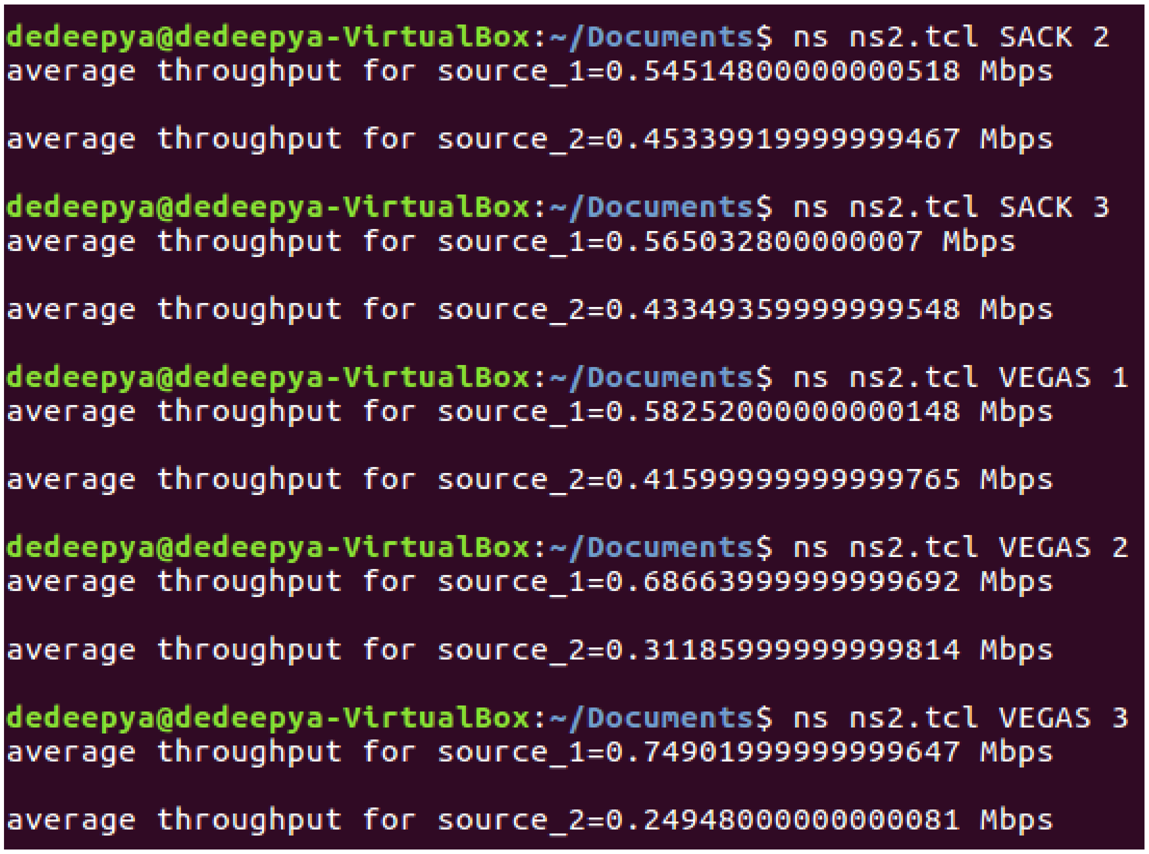
3. **Test Results:**

After we run all our test cases, we get the following results in our command line:



```
dedeepya@dedeepya-VirtualBox:~/Documents$ ns ns2.tcl SACK 1
average throughput for source_1=0.52330800000000588 Mbps

average throughput for source_2=0.47523919999999636 Mbps
```



```
dedeepya@dedeepya-VirtualBox:~/Documents$ ns ns2.tcl SACK 2
average throughput for source_1=0.54514800000000518 Mbps

average throughput for source_2=0.45339919999999467 Mbps

dedeepya@dedeepya-VirtualBox:~/Documents$ ns ns2.tcl SACK 3
average throughput for source_1=0.565032800000007 Mbps

average throughput for source_2=0.43349359999999548 Mbps

dedeepya@dedeepya-VirtualBox:~/Documents$ ns ns2.tcl VEGAS 1
average throughput for source_1=0.58252000000000148 Mbps

average throughput for source_2=0.41599999999999765 Mbps

dedeepya@dedeepya-VirtualBox:~/Documents$ ns ns2.tcl VEGAS 2
average throughput for source_1=0.68663999999999692 Mbps

average throughput for source_2=0.31185999999999814 Mbps

dedeepya@dedeepya-VirtualBox:~/Documents$ ns ns2.tcl VEGAS 3
average throughput for source_1=0.74901999999999647 Mbps

average throughput for source_2=0.24948000000000081 Mbps
```

4. **Throughput Ratios:**

After we observe the three RTT cases for each of our TCP flavors, TCP VEGAS and TCP SACK, we find the ratio of the average throughput of src1 to src2.

TCP Flavor	Throughput Src1 (Mbps)	Throughput Src2 (Mbps)	Ratio
SACK 1	0.5233	0.4752	1.1012
SACK 2	0.5451	0.4534	1.2022
SACK 3	0.5650	0.4335	1.3033

TCP Flavor	Throughput Src1 (Mbps)	Throughput Src2 (Mbps)	Ratio
VEGAS 1	0.5825	0.4160	1.3999
VEGAS 2	0.6866	0.3118	2.2020
VEGAS 3	0.7490	0.2495	3.0020

(i) We observe from our table that for TCP SACK, the decrease in throughput is not proportional to the increase in delay. The ned-to-end RTT's for the two sources are in the ratio, 1:2, 1:3 and 1:4 respectively, but the ratio of the throughput's are 1.1012, 1.2022 and 1.3033.

In the case of TCP VEGAS, this increase in delay is much more noticeable in the throughput ratios obtained. For the RTT ratio 1:2, 1:3 and 1:4, the ratios obtained are 1.3999, 2.2020 and 3.0020 respectively.

(ii). Hence, we can conclude that RTT has a significant influence on the throughput of TCP VEGAS. We observe higher throughputs for smaller delays i.e. 5ms, the performance of TCP VEGAS was considerably better (src1) than TCP SACK.

(iii). Reasons for better performance of TCP VEGAS in comparison to TCP SACK:

- TCP VEGAS introduced the concept of congestion avoidance due to its capability of better estimating congestion (by measuring the change in the change in throughput rather than packet loss) whereas TCP SACK works on the concept of fast recovery that is it implements a slow start mechanism and detects lost packets so that the pipeline is not emptied every time a packet is lost.
- It is comparatively difficult to implement SACK because Selective Acknowledgements are to be provided by the receiver and hence are difficult to incorporate in the current TCP.

5. References

1. An Introduction to NS-2 by Kiran Kotla (Recitation Session)
2. https://www.isi.edu/nsnam/ns//
3. "Performance Comparison between TCP SACK and TCP VEGAS using NS-2 Simulator" by Heena Dave, Vikas Gupta and Parul Dihulia.
4. Stackoverflow