

Network Simulation Assignment #1

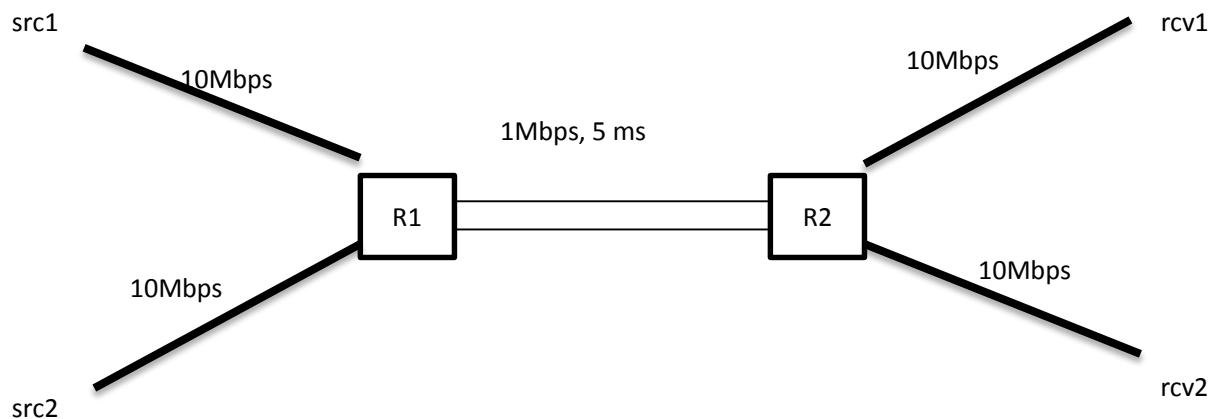
ECEN 602, Fall 2015

Due Nov. 10, 2015 NLT 5:00 pm

NS-2 Familiarization

Use the NS-2 simulator to build the following configuration:

- 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



Run 400s simulations for the following variable parameters:

TCP version = (TCP SACK | TCP VEGAS)

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

Note that in the cases above the end-to-end RTTs of the two sources are in the ratio 1:2, 1:3 and 1:4 respectively. This should give you 6 independent simulations, 2 TCP versions x 3 end-2-end delays.

(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).

(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).

Note: Run the simulations for 400 seconds, and ignore the first 100 seconds while measuring metrics.

Usage Syntax:

telsh ns2.tcl <TCP_flavor> <case_no>

example: telsh ns2.tcl VEGAS 2

Submission Guidelines

1. The NS-2 simulation report (**report.pdf**), code (**ns2.tcl**) and readme file (**readme.txt**) should be uploaded on google drive folder before the beginning of class on the date that it is due. The Google drive folder (teamxx) has been shared to each team. NS1 folder will be added shortly.
2. Report (**report.pdf**) explaining your test setup, procedure, results, and explanations to the preceding questions. Include tables showing throughput of each test.
3. Make sure your code (**ns2.tcl**) takes command line arguments as mentioned above. Your code should display throughput on terminal.

4. The **readme.txt** should contain a summary of your code: architecture, usage, and errata, etc.
5. When submitting, delete all the demo files and upload only these file: **report.pdf**, **ns2.tcl** and **readme.txt**
6. Make sure all binaries and object code have been cleaned from your project before you submit.
7. This assignment does not require you to use the machines in 213A, you may install the simulator on your own machines (Ubuntu) for development.
8. Should you have any doubts, please ask questions.
9. All submissions will go through Stanford Moss to detect plagiarism. Don't copy code from somewhere else.