CS 342: Networks Lab

(July – November 2021)

Assignment – 4: Network Simulation using NS-3

Submission Deadline: 21st October 2021 (hard deadline)

In this assignment you need to simulate a computer network for a given application using the discrete event network simulator **ns-3**. You can download the software and documentation of **ns-3** from the website https://www.nsnam.org. The assignment will be solved in groups where each group, comprised of up-to 4 members, needs to work on an application assigned to it. The applications' network specifications, the required experiments, and related questions are given in pages 2-7 of this document. Follow the general instructions given below and any specific instructions mentioned in the application description.

General Instructions:

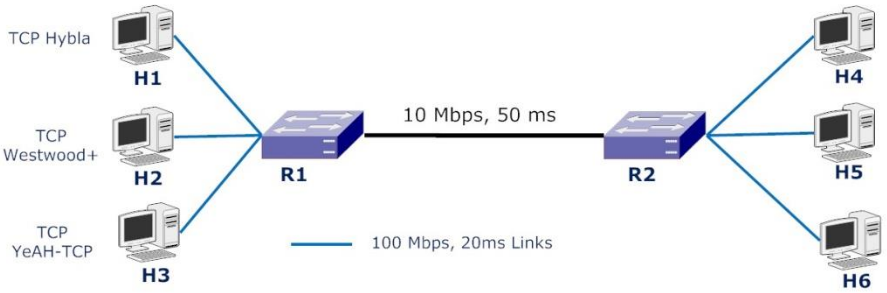
1. Each group needs to simulate one application assigned to them and make one single submission on Moodle. **Only one member from a group needs to make the submission**. The information about the allocation of applications to groups is contained in **Table 1** given below.
2. Install **ns-3** on your computer, write programs and simulate the network described in the given application assigned to you, perform the required experiments, and answer the given questions.
3. Use **ns-3**'s **Flow Monitor** module to collect and store the performance data of network protocols from the simulation. **Do not use PCAP + Wireshark for the trace collection**. No marks will be awarded if your application uses PCAP file for trace collection.
4. Submit your set of source code files, **along with a report clearly mentioning all observations, explanations, containing the graphs, and all the answers**, as a zipped file on Moodle. The **ZIP file's name should be the same as your group number**, for example, "Group_4.zip", or "Group_4.rar", or "Group_4.tar.gz**". Submission without a report will not be awarded any marks.**
5. **Grading Scheme:** Total marks 20. 10 marks for code compilation and execution. 10 marks for answers, observations, and plot analysis in report.
6. **Write your own source codes and do not copy from any source. Plagiarism and use of unfair means will be penalised by awarding NEGATIVE marks (equal to the maximum marks for the assignment).**

Table 1: Allocation of applications to groups

App ID	GROUP-IDs							
1	1	2	3	4	5	6	7	43
2	8	9	10	11	12	13	14	44
3	15	16	17	18	19	20	21	45
4	22	23	24	25	26	27	28	46
5	29	30	31	32	33	34	35	
6	36	37	38	39	40	41	42	

Application #1:

Analyse and compare TCP Hybla, TCP Westwood+, and TCP YeAH-TCP performance. Select a Dumbbell topology with two routers R1 and R2 connected by a (10 Mbps, 50 ms) wired link. Each of the routers is connected to 3 hosts, i.e. H1, H2, H3 (i.e. senders) are connected to R1, and H4, H5, H6 (i.e. receivers) are connected to R2. The hosts are attached with (100 Mbps, 20 ms) links. Both the routers use drop-tail queues with queue size set according to bandwidth-delay product. Senders (i.e. H1, H2 and H3) are attached with TCP Hybla, TCP Westwood+, and TCP YeAH-TCP agents, respectively. Choose a packet size of 1.3 KB and perform the following tasks. Make appropriate assumptions wherever necessary.



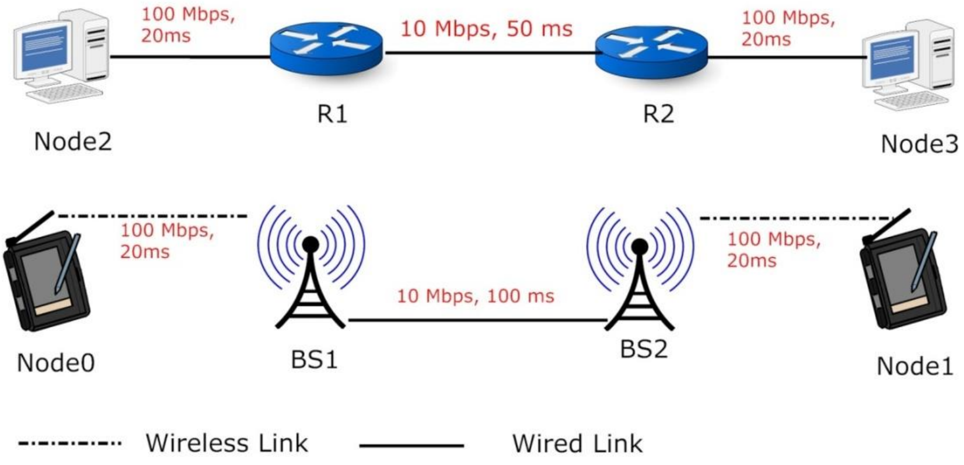
1. Start only one flow and analyse the throughput over sufficiently long duration. Mention how you select the duration. Plot the evolution of congestion window w.r.t. time. Perform this experiment with all the flows attached to all the three sending agents.

2. In the next experiment, start 2 other flows sharing the bottleneck while the first one is in progress and measure the throughput (in Kbps) of each flow. Plot the throughput and evolution of the TCP congestion window for each of the flow at a steady-state. Report the maximum throughput observed for each of the flows.

3. Measure the congestion loss and the goodput over the duration of the experiment for each of the flows.

Application #2:

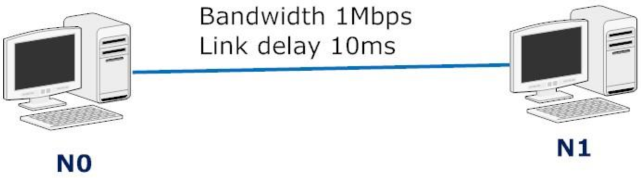
Compare the performance of TCP over wired and wireless networks. Consider a topology as described below. The network consists of two TCP sources Node0 and Node2, corresponding to two TCP destinations Node1 and Node3 respectively. Node2 and Node3 come in wired domain with two routers R1 and R2 (connected by a {10 Mbps, 50 ms} wired link) between them. Both the routers use drop-tail queues with queue size set according to bandwidth-delay product. Node0 comes in domain of Base Station 1 (BS1) and Node1 comes in domain of Base Station 2 (BS2). BS1 and BS2 are connected by a (10 Mbps, 100 ms) wired link. The hosts, i.e. Node0, Node1, Node2, Node3 are attached with (100 Mbps, 20ms) links to routers or base stations (as shown in the figure below). The sources (Node0 and Node2)) use three TCP agents (i.e. TCP Westwood, TCP Veno and TCP Vegas) to generate three different TCP flows. Study and plot the fairness index (Jain's fairness index) and throughput change when the TCP packet size is varied; all the other parameter values are kept constant. You should use the following TCP packet size values (in Bytes): 40, 44, 48, 52, 60, 552, 576, 628, 1420 and 1500 for your experiments. The throughput (in Kbps) and fairness index must be calculated at steady-state. Make appropriate assumptions wherever necessary.



Application #3:

Create a topology of two nodes N0 and N1 connected by a link of bandwidth 1 Mbps and link delay 10 ms. Use a drop-tail queue at the link. Set the queue size according to bandwidth-delay product. Create a TCP agent (type of the agent specified below) and FTP traffic at N0 destined for N1. Create 5 CBR traffic agents of rate 300 Kbps each at N0 destined for N1. Make appropriate assumptions wherever necessary. The timing of the flows are as follows:

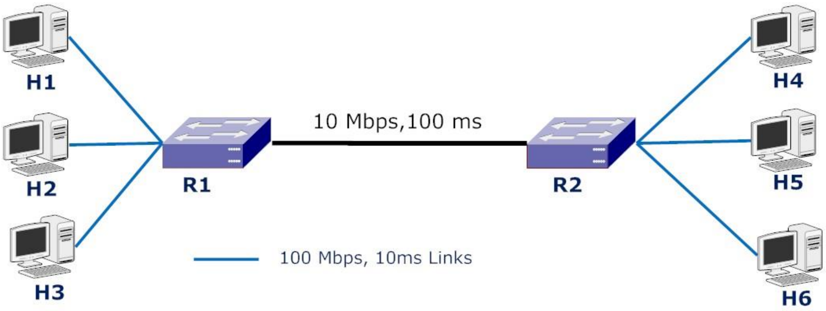
- FTP starts at 0 sec and continues till the end of simulation.
- CBR1 starts at 200 ms and continues till end.
- CBR2 starts at 400 ms and continues till end.
- CBR3 starts at 600 ms and stops at 1200 ms.
- CBR4 starts at 800 ms and stops at 1400 ms.
- CBR5 starts at 1000 ms and stops at 1600 ms.
- Simulation runs for 1800 ms.



1. Plot graph(s) of TCP congestion window w.r.t. time for following 5 TCP congestion control algorithm implementations, and describe the TCP congestion control algorithms' behaviour.
    - Case 1: use TCP New Reno
    - Case 2: use TCP Hybla
    - Case 3: use TCP Westwood
    - Case 4: use TCP Scalable
    - Case 5: use TCP Vegas
2. Draw a graph showing cumulative TCP packets dropped w.r.t. time comparing above 5 TCP congestion control algorithm implementations.
3. Draw a graph showing cumulative bytes transferred w.r.t. time comparing above 5 TCP congestion control algorithm implementations.

Application #4:

Compare the effect of buffer size on TCP and UDP flows. Select a Dumbbell topology with two routers R1 and R2 connected by a (10 Mbps, 100 ms) link. Each of the routers is connected to 3 hosts, i.e. H1, H2, H3 are connected to R1, and H4, H5, H6 are connected to R2. All the hosts are attached to the routers with (100 Mbps, 10 ms) links. Both the routers (i.e. R1 and R2) use drop-tail queues with equal queue size set according to bandwidth-delay product. Choose a packet size of 1.5 KB. Start 4 TCP New Reno flows, and after a while start 2 CBR over UDP flows each with 20 Mbps. These flows are randomly distributed across H1, H2 and H3. Increase the rate of one UDP flow up to 100 Mbps and observe its impact on the throughput of the TCP flows and the other UDP flow. Vary the buffer size in the range of 10 packets to 800 packets and repeat the above experiments to find out the impact of buffer size on the fair share of bandwidth and plot the necessary graphs. Make appropriate assumptions wherever necessary.



Application #5:

Using the network simulator ns-3, study the characteristics of IEEE 802.11. For the purpose of experiment, use the topology as follows. There are 3 nodes in the network located in a straight line at locations 250*i, with i=0, 1, 2. Node 0 and Node 2 both have TCP traffic to Node 1 (started randomly within 1 to 5 seconds of starting the simulation). Consider TCP Westwood+ or TCP Hybla for the TCP agents at Node 0 and Node 2, respectively. You have to run the simulations and measure the following from the trace output (the averages are taken over all the nodes). Do not use PCAP file for collecting the trace. Use Flow Monitor module in ns-3 for trace collection. *No marks will be given if you consider PCAP trace with Wireshark.*

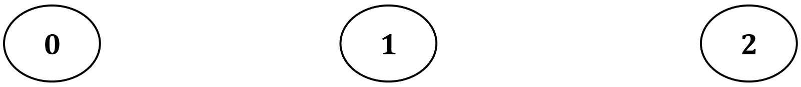
1. Average bandwidth spent in transmitting RTS, CTS, and ACK.

2. Average bandwidth spent in transmitting TCP segments and TCP acks.

3. Average bandwidth wasted due to collisions.

4. TCP throughput (number of acknowledged bytes per unit time) at each node.

You have to run the simulations for 50 seconds each with different RTS thresholds (i.e. 0, 256, 512 and 1000 bytes) and TCP segment size of 1000 bytes. You can use scripts for trace file analysis and to plot the results. Make appropriate assumptions wherever necessary.

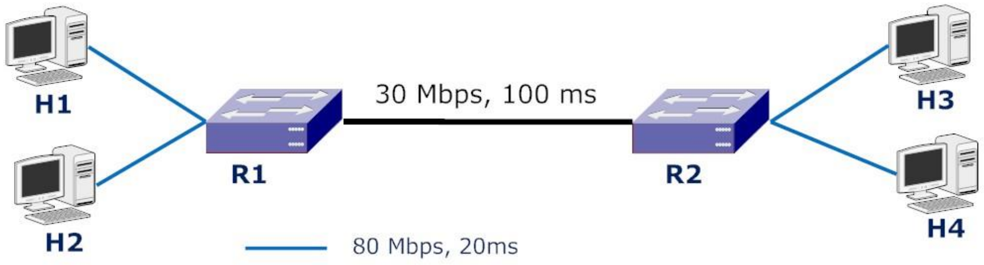


Application #6:

The objective is to compare the effect of CBR traffic over UDP agent and FTP traffic over TCP agent. Consider a TCP agent from TCP HighSpeed, TCP Vegas and TCP Scalable for the FTP traffic. Consider a Dumbbell topology with two routers R1 and R2 connected by a wired link (30 Mbps, 100 ms), and use drop-tail queues with queue size set according to bandwidth-delay product of the link. Each of the routers is connected to 2 hosts, i.e. H1, H2 are connected to R1, and H3, H4 are connected to R2. The hosts are attached to the routers with (80 Mbps, 20ms) links. The CBR traffic over UDP agent and FTP traffic over TCP agent are attached to H1 and H2 respectively. Choose appropriate packet size for your experiments and perform the following:

1. Compare the delay (in ms) and throughput (in Kbps) of CBR and FTP traffic streams when only one of them is present in the network. Plot the graphs for the delay (in ms) and throughput (in Kbps) observed with different packet sizes.

2. Start both the flows at the same time and also at different times. Also, compare the delay (in ms) and throughput (in Kbps) of CBR and FTP traffic streams. Plot the graphs for the delay (in ms) and throughput (in Kbps) observed with different packet sizes.



Make appropriate assumptions wherever necessary.