

## CS-349 Networks Lab

### Assignment - 3: Network Simulator Assignment

**Submission deadline: 11:55 PM on 27<sup>th</sup> March, 2016 (Hard Deadline)**

In this assignment you need to simulate the given application using the simulator assigned to your group number. Check the application and simulator assigned to each group in the table given below.

Application No.	Simulator Assigned	Group Numbers
1	NS - 3	1, 13, 25
2	Mininet	2, 14, 26
3	NS - 3	3, 15, 27
4	OMNET++	4, 16, 28
5	Mininet	5, 17, 29
6	NS - 3	6, 18, 30
7	Mininet	7, 19, 31
8	NS - 3	8, 20, 32
9	Mininet	9, 21, 33
10	NS - 3	10, 22, 34
11	OMNET++	11, 23, 35
12	NS - 3	12, 24, 36

**NOTE: Submit your code together with graphs (preferably in a report format) in a zipped folder (max size 1 MB) on Moodle by 11:55 PM on 27<sup>th</sup> March 2016 (hard deadline). Assignment will be evaluated through Viva-voce on 30<sup>th</sup> March 2016; during your lab session (evaluation schedule and TA assignment will be notified later). Late submission will be penalized by taking necessary actions.**

---

#### Application #1:

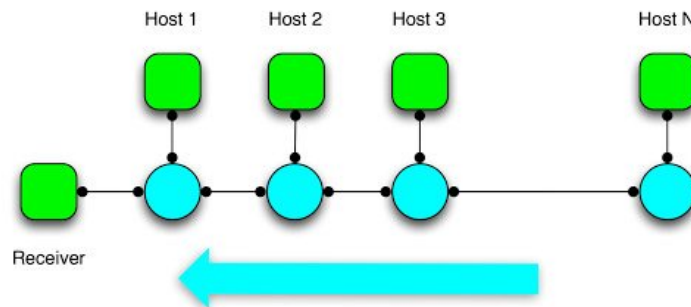
Compare the effect of buffer size on TCP and UDP flows. Select a Dumbbell topology with two routers R1 and R2 connected by a (10 Mbps, 100 ms) link. Each of the routers is connected to 3 hosts i.e., H1, H2 and H3 are connected to R1 and, H4, H5 and H6 are connected to R2. All the hosts are attached to the routers with (100 Mbps, 10ms) links. Both the routers use drop-tail queues with equal queue size set according to bandwidth-delay product. Choose a packet size of 1.5KB. Start 4 TCP New Reno flows and after a while start 2 CBR over UDP flows each with 20 Mbps. 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 #2:

The application has two parts and is as follows:

- A. Using Mininet, create a d-cell topology with  $n=2,3,4$  and do the following tests:
  - a. Reachability - check the connectivity of nodes (nodes should be mutually connected)
  - b. Compare the performance of dcell-2, dcell-3 and dcell-4 in terms of TCP throughput, latency using "iperf" and "tcpdump" (or any other tool of your choice).

- B. Create the topology given below for n hosts (where n is a command line input).
- Simulation parameters: All the green nodes are hosts and circular ones are switches. Hosts 1 to n generate data to be sent to the receiver.
  - Results required
    - Considering each of the link is 10Mbps with delay of 10ms, plot the behavior of TCP Congestion Control, namely, Effective Data rate v/s time and Congestion window v/s time.
    - Repeat (i), when bandwidth of each link is reduced to 1Mbps
    - Repeat (i), when delay is increased to 50ms
    - For topology created in PART-A, make one host as server and remaining as clients, and then plot graphs as in (i) for the server and any one client.



NOTE: MAKE sure that congestion builds up at the router by increasing the rate at which iperf generates the data.

### Application #3:

Create a topology of two nodes N0 and N1 connected by a link of bandwidth 1Mbps and link delay 10ms. Use a drop-tail queue at the link. 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 is 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
- Plot a graph of TCP congestion window w.r.t. time for following 5 TCP agent implementations and explain.
    - Case 1: use TCP Reno
    - Case 2: use TCP Tahoe
    - Case 3: use TCP Westwood
    - Case 4: use TCP New Reno
    - Case 5: use TCP Fack
  - Draw a graph showing cumulative TCP packets dropped w.r.t. time comparing above 5 TCP implementations.
  - Draw a graph showing cumulative bytes transferred w.r.t. time comparing above 5 TCP implementations.

#### Application #4:

Under wired/wireless scenario do single source multicast using tree topology. Consider dynamic nature of nodes to join and leave multicast group and maintain an optimal tree height at every time instant. Plot the average and maximum height of the tree and source to node delay for the variable number of nodes. The number of nodes varies from 10 to 100. Also plot the average number of node transitions needs to be done to maintain optimal tree height when a new node joins the multicast group. Assume number of nodes in this scenario varies from 10 to 50.

#### Application #5:

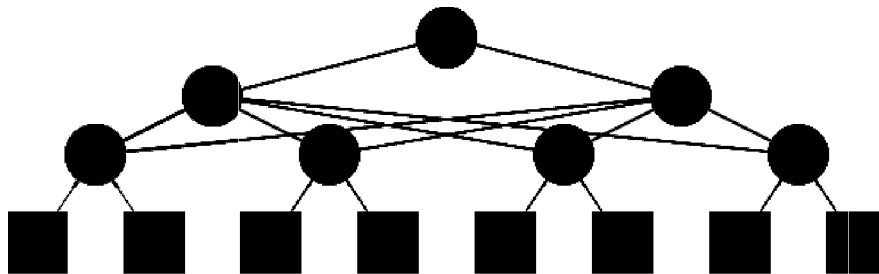
The application has two parts and is as follows:

- A. Using Mininet, create a Fat-tree like topology (given below) and do the following tests:
  - a. Reachability - check the connectivity of nodes (nodes should be mutually connected)
  - b. Compare the performance of TCP in terms of throughput and latency, under different type of traffic patterns i.e. one-to-many and many-to-many. You can use tools like "iperf" and "tcpdump" (or any other tool of your choice).
- B. TCP with increased congestion window(TCP-I) vs default TCP:

Perform simulation in Mininet, using a topology where a switch is connected to three nodes of which one is the server (S, serving the requests) and remaining two are the clients (A,B).

  - a. Simulation parameters:

Client A requests data of different sizes (1MB,10MB,100MB,1GB) from S while B establishes a longer TCP connection with S, which competes with A.
  - b. Results required:
    - i. Compare the performance of default TCP and TCP-I in terms of throughput and latency when initial congestion window for TCP-I is 5 segments, 8 segments and 10 segments (note: default TCP has 3 segments).
    - ii. Do a similar comparison as in (i), this time with segment size of 10 and bandwidth varying as (10Kbps, 100Kbps, 500Kbps, 1000Kbps, 5000Kbps)
    - iii. Repeat (ii) for the topology in PART-A



Topology for PART-A. Here circles are switches and squares are servers while the lines indicate presence of direct link.

#### Application #6:

Analyze and compare TCP Reno, TCP Westwood, and TCP Fack (i.e. Reno TCP with "forward acknowledgment") 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 to H3 (i.e. senders) are connected to R1 and H4 to H6 (i.e. receivers) are connected to R2. The hosts are attached with (100 Mbps, 20ms) 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 Reno, TCP Westwood, and TCP Fack agents respectively. Choose a packet size of 1.2KB and perform the following task. Make appropriate assumptions wherever necessary.

1. Start only one flow and analyze the throughput over sufficiently long duration. Mention how you select the duration. Plot of evolution of congestion window over time. Perform this experiment with flows attached to all the three sending agents.
2. Next, 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 congestion window of each flow at steady-state. What is the maximum throughput of each flow?
3. Measure the congestion loss and Goodput over the duration of the experiment for each flow.

#### Application #7:

The application has two parts and is as follows:

- A. Using Mininet, create a hypercube topology (of switches) in 2-D and 3-D. Each switch is connected with two hosts and do the following tests:
  - a. Reachability - check the connectivity of nodes (nodes should be mutually connected)
  - b. Compare the performance of TCP in terms of throughput and latency, under different type of traffic patterns i.e. one-to-many and many-to-many. You can use tools like "iperf" and "tcpdump" (or any other tool of your choice).
- B. Compare the performance of MPTCP (vs) TCP Reno (vs) TCP Cubic on the topology created in PART-A, in terms of throughput and latency. Assume one of the hosts to be the server and other to be client.

Results required:

- a. Effect of bandwidth: Performance comparison when bandwidth increases from 10 Kbps to 5Mbps with at least 10 data points.
- b. Effect of data size: Performance comparison when request size increases from 1MB to 1GB with at least 10 data points (on fixed bandwidth)
- c. Growth of congestion window: Show for a given bandwidth (of your choice) how congestion window varies with time (the TCP connection time should be at least 5 minutes)

#### Application #8:

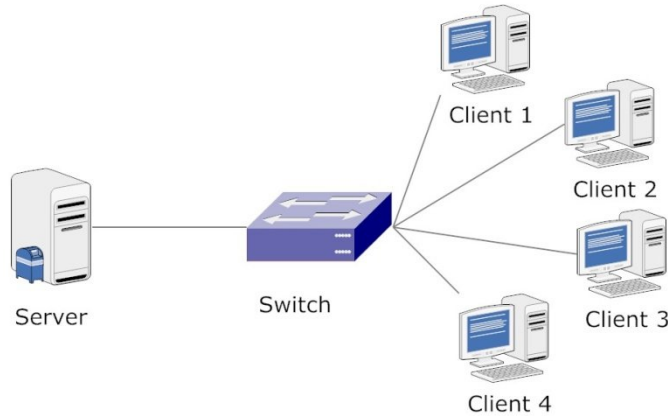
Compare the effect of CBR traffic over UDP agent and FTP traffic over TCP agent. Consider a Dumbbell topology with two routers R1 and R2 connected by a (30 Mbps, 100 ms) wired link and use drop-tail queues with queue size set according to bandwidth-delay product of the link. Each of the routers is connected to 2 hosts i.e., H1 and H2 are connected to R1 and H3 and H4 are connected to R2. The hosts are attached to the routers with (80 Mbps, 20ms) links. 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.
2. Start both the flows at the same time and on different time. Also, compare the delay (in ms) and throughput (in Kbps) of CBR and FTP traffic streams.

Make appropriate assumptions wherever necessary.

#### Application #9:

- A. Create the following topology using Mininet (see figure below) and check the reachability of nodes.
- B. Make the required changes in the (Mininet) controller so that the switch can act as a NAT. The internal interface should face the clients while external interface should be connected to the server
- C. NOTE: - The functionality of NAT has to be implemented using the Mininet controller and use of the Linux IP tables is prohibited. Make sure your controller is scalable.



#### Application #10:

Compare the performance of TCP over wired and wireless networks. Consider a topology as described below. Network consists of two TCP sources Node0 and Node2 and corresponding to two TCP destinations Node1 and Node3 respectively. Node2 and Node3 come in wired domain with two routers R1 and R2 in between them. 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 sources use TCP Reno as the sending TCP agent. 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 KB): 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 #11:

In a static wireless scenario, simulate hidden node and exposed terminal problem of CSMA/CA. Also simulate the solutions to these problems with the help of 802.11 RTS/CTS messages exchanged between the nodes. Calculate the throughput with and without RTS/CTS messages. Also calculate the overhead incurred due to control messages.

#### Application #12:

Using network simulator, ns, study the characteristics of IEEE 802.11. For the purpose of experiment, use the topology as follows. There are 3 (three) nodes in the network located in a straight line at locations  $250 \times I$ ,  $i=0, 1, 2$ . Node 0 and Node 2 both have TCP traffic to Node 1 (started randomly within 1 to 5 seconds of starting the simulation). You have to run the simulations and measure the following from the trace output (the averages are taken over all the nodes).

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 threshold (0, 256, 512 and 1000 bytes) and TCP segment size of 1000 bytes. Use scripts for trace file analysis and plot the results. Make appropriate assumptions wherever necessary.

