

**Computer Networks Laboratory**

**COT-411/IT 312**

**LAB Manual**

## Experiments Using Network Simulator2 (ns2)

### **Experiment 1: Fundamentals of Network Simulation**

#### **1. Ns2 Installation and Beginning. Refer to:**

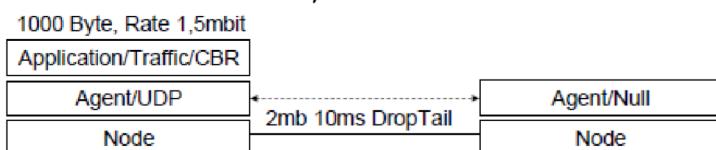
- (i) <http://www.isi.edu/nsnam/ns>
- (ii) <http://nile.wpi.edu/NS/>
- (iii) [http://www.winlab.rutgers.edu/~zhibinwu/html/network\\_simulator\\_2.html](http://www.winlab.rutgers.edu/~zhibinwu/html/network_simulator_2.html)
- (iv) Book: High Performance TCP/IP Networking – Concepts, Issues, and Solutions by M. Hassan and Raj Jain, Prentice Hall India, 2009, Chapter 4: TCP/IP Network Simulation

First learn various components of ns2 and C++ class hierarchy. Learn the concept of event driven programming, configure, install, path setup, install directory etc. as required for installation of ns2. In ns2, learn about setting of nodes, links, Source, Sink, Agents, GOD (General Operations Director), energy model, nam, tracefile and its contents, gnuplot, wired and wireless simulation, and basics of OTCL scripting. Chapter 3 may also be referred from the above referred book.

### **Experiment 2: Basics of ns2**

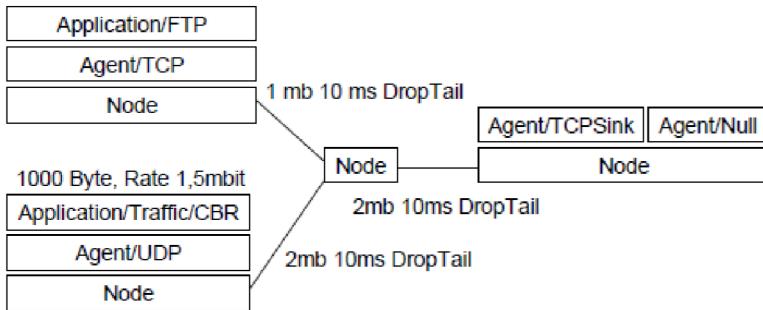
#### **1. Create the following scenario with two nodes n0 and n1 and link in between.**

- Sender agent: Agent/UDP
- Receiver agent: Agent/Null
- Connect agents
- Data source: Application/Traffic/CBR
- Run from 0.5 to 4.5 sec, finish at 5.0 sec



#### **2. Create the following scenario and connect the appropriate agents**

- Start the FTP application at t = 0.5s
- Start the CBR data source at t = 1s
- Terminate both at t = 4.5 s
- Visualize the bottleneck queue



### **3. Write a TCL script to simulate a file transfer with using ns2:**

Consider a client and a server. The server is running a FTP application (over TCP). The client sends a request to download a file of size 10 MB from the server. Write a script to simulate this scenario. Let node #0 be the server and node #1 be the client. TCP packet size is 1500 B. Assume typical values for other parameters.

### **4. Write a TCL script to simulate the network described below using ns2:**

Consider a small network with five nodes n0, n1, n2, n3, n4, forming a star topology. The node n4 is at the center. Node n0 is a TCP source, which transmits packets to node n3 (a TCP sink) through the node n4. Node n1 is another traffic source, and sends UDP packets to node n2 through n4. The duration of the simulation time is 10 seconds.

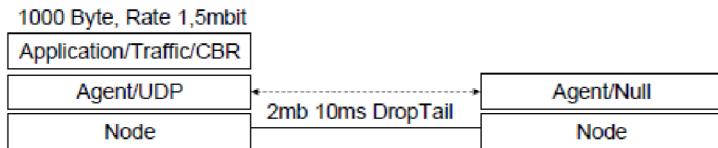
## **Experiment 3: Simulating a Local Area Network**

### **1. Setting up a local area network with ns2:**

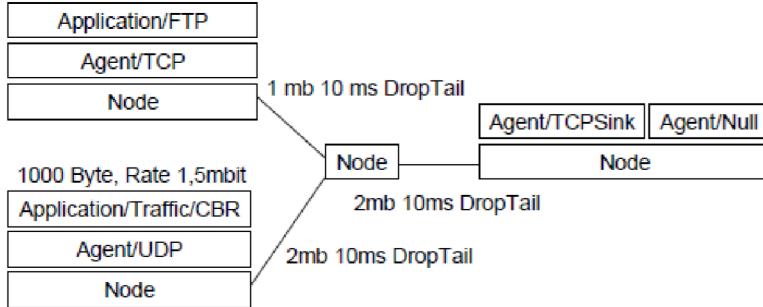
Consider the LAN with seven nodes to be an isolated one i.e. not connected to the Internet. Node #0 in the LAN act as a UDP traffic source, and node #6 is the destination node. Assume CBR traffic to be flowing between the nodes. The simulation lasts for 25 seconds. In Ethernet a packet is broadcasted in the shared medium, and only the destination node accepts the packet. Other nodes simply drop it. How many hops a packet should take to travel from node #0 to node #6? Verify this from the "Hop Count" plot.

### **2. Create the following scenario with two nodes and link in between.**

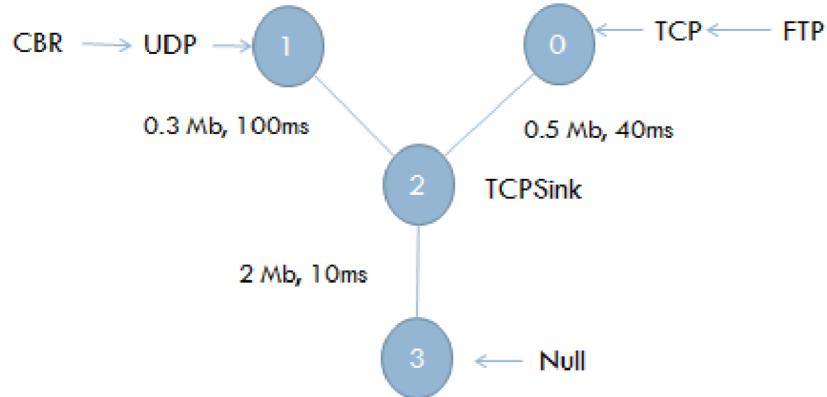
- Sender agent: Agent/UDP
- Receiver agent: Agent/Null
- Connect agents
- Data source: Application/Traffic/CBR
- Run from 0.5 to 4.5 sec, finish at 5.0 sec



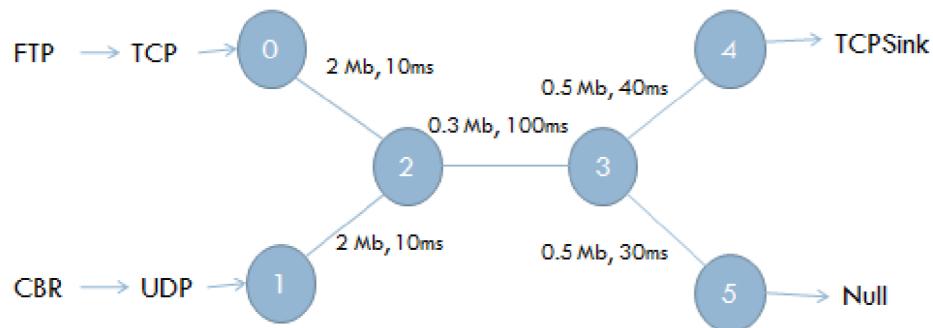
3. Create the following scenario and connect the appropriate agents
- Start the FTP application at  $t = 0.5$ s
  - Start the CBR data source at  $t = 1$ s
  - Terminate both at  $t = 4.5$  s
  - Visualize the bottle neck queue



4. In the following simulation scenario set the following parameters:
- Duplex link between n1 and n2
  - Simplex link between node n0 and n2
  - Queue Size of link n2-n3 to 40
  - CBR traffic packet size: 1000B, inter-arrival time: 8ms, start at time 1.0; TCP window size 8000, packet size 512B
  - Generate nam trace and show animation



5. Generate the following simulation scenario

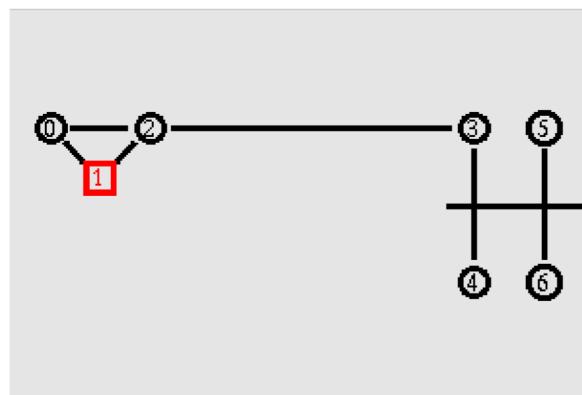


**Figure3**

6. Design Ring Topology of 10 nodes (n0 to n9) with nodes connected in ascending order and n9 connected to n0. Create and connect the nodes using for loop.

#### Experiment 4: Simulating link errors using ns2

Consider the following topology:



Here node #2 act as a router. Any traffic to or from the LAN passes through it. Consider node #1 running a FTP server, and node #5 is downloading a file of size 4 MB. However, the link between node #2 and #3 is fault. It drops packets with a fixed probability of 0.2. Implement a link error model to reflect this.

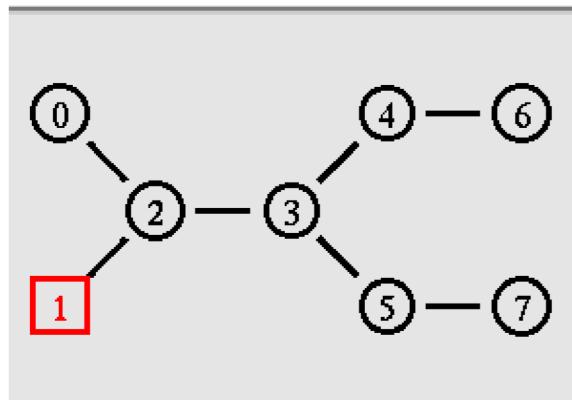
It may be noted here that the file download time will be more than the we had in exercise #2 of experiment #1. Try different values of the simulation time to ensure that the file has been entirely transferred. Is the plot of bytes received a linear curve or non-linear? Why?

Presence of link errors cause one or more packets to be retransmitted. Verify this from the "Packet Retransmissions" plot.

## Experiment 5: Measuring Network Performance

### 1. Simulating Bottleneck in the network using ns2:

Consider a dumbbell topology with eight nodes as shown as in the following figure. Consider nodes #2 and 3 to be two routers connecting two different networks. When the bandwidth of the link 2-3 is much lower than the sum of bandwidths of the other links in the network, it act as a bottleneck.



Assume node #0 running a FTP application (over TCP) and sending data to node #6. Node #1 is sending CBR data to node #7. Assume all the links except 2-3 has a bandwidth of 1 Mb, propagation delay of 10ms and queue type as DropTail. (All are duplex links).

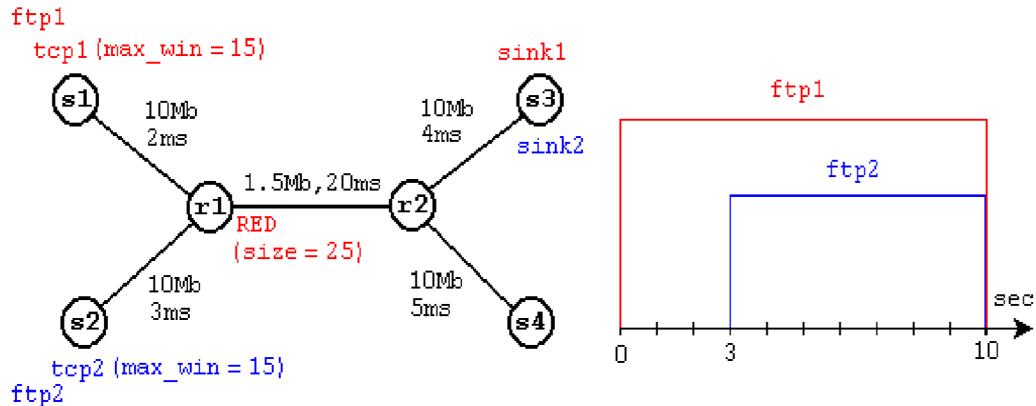
#### Tasks:

- The link 2-3 has a propagation delay of 10 ms. Vary its bandwidth from 0.5 Mb to 2.5 Mb in steps of 0.25Mb.
- Compute the throughput for node #3 in each case
- Plot the throughput vs. bandwidth data in the "Custom Plot" section below

Based on the above plots, suggest what should be the recommended bandwidth of the link 2-3.

Now, plot the end-to-end delay between nodes 0 and 6 for the above chosen values of link 2-3 bandwidth. Revisit your previous answer (i.e. optimum bandwidth of link 2-3) based on these graphs.

2. Set up the network topology and run the simulation scenario as shown below



3. Create a 10 node wireless topology. The nodes use AODV (an ad-hoc routing protocol) to route packets among themselves. They also use

- 802.11 mac
  - two-ray ground radio propagation model
  - priority queue with queue length of 50
  - omni-directional antenna
  - topology of 670 X 670.
  - the simulation stops at 200s
4. An Ethernet with  $n$  nodes, where  $n-1$  nodes send Ethernet frames to one designated node acting as receiver. Experiment with two frame sizes (64 bytes, and 1518 bytes). generating performance plots for: (a) two different types of delay vs. offered load, (b) Demonstrate the effect of buffer size on drop probability, where the offered load is kept constant at a high value, but lower than the network capacity (say, 8-10 M bits/sec) and use medium sized frame (around 512 bytes). The offered load should be varied from 1 Mb/s to about 100 Mb/s.

## Experiment 6: Using Wireshark Tool

<https://www.wireshark.org/>

[http://cs.gmu.edu/~astavrou/courses/ISA\\_674\\_F12/Wireshark-Tutorial.pdf](http://cs.gmu.edu/~astavrou/courses/ISA_674_F12/Wireshark-Tutorial.pdf)

[http://www-scf.usc.edu/~csci571/Special/Tutorials/wireshark\\_html/wireshark.html](http://www-scf.usc.edu/~csci571/Special/Tutorials/wireshark_html/wireshark.html)

- Each line in the top pane of the Wireshark window corresponds to a single packet seen on the network. The default display shows the time of the packet (relative to the initiation of the capture), the source and destination IP addresses, the protocol used and some information about the packet. Learn Wireshark color coding from help documents.

2. To locate specific packets related to individual requests or responses from a within larger capture containing more traffic, we can perform even more specific filtering using a variety of expressions relating to various header fields and their contents. Study how to applying a display filter to the captured packets.
3. Use your Web browser to access a file from a Web server. Using Wireshark capture the packets arriving at your computer. Also download a Wireshark-readable packet trace from the Web server from which you downloaded the file. Using this server trace
  - Find the packets that were generated by your own access of the Web server.
  - Analyze the client- and server-side traces to explore aspects of TCP.
  - Evaluate the performance of the TCP connection between your computer and the Web server.
  - Trace TCP's window behavior, and infer packet loss, retransmission, flow control and congestion control behavior, and estimated roundtrip time.
4. Using Wireshark explores the use of the ICMP protocol in the ping and traceroute commands.
5. Using Wireshark explore several aspects of the HTTP protocol: the basic GET/reply interaction, HTTP message formats, retrieving large HTML files, retrieving HTML files with embedded URLs, persistent and non-persistent connections, and HTTP authentication and security.
6. Using Wireshark perform a packet capture and analysis of an application that uses UDP (for example, DNS or a multimedia application such as Skype). Also, investigate the header fields in the UDP segment as well as the checksum calculation.
7. Using Wireshark examines the operation of the IP protocol, and the IP datagram format.
8. Using Wireshark lab perform capture and study the 802.11 frames exchanged between a wireless laptop and an access point.