**Practical 8-b**

**AIM:** Demonstration of NS2

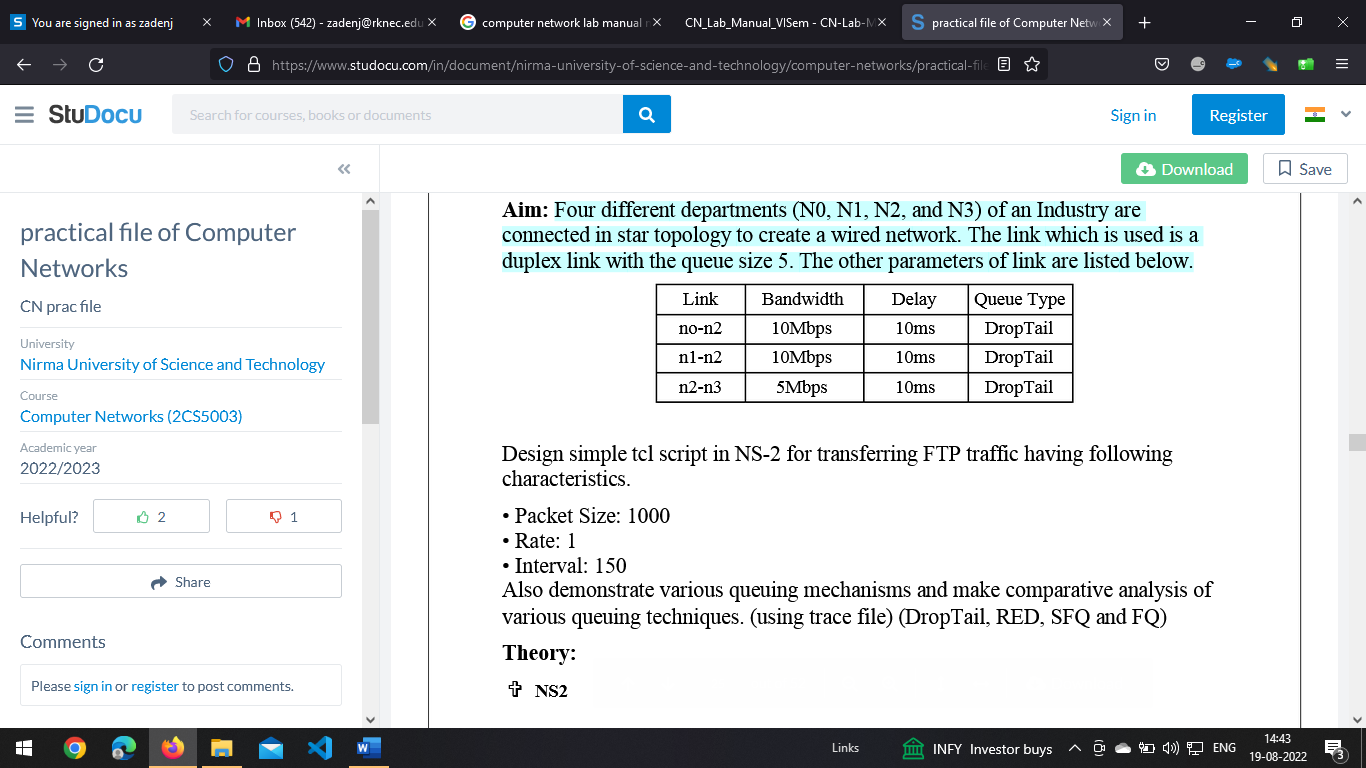
Consider Four different departments (N0, N1, N2, and N3) of an Industry they are connected in a star topology to create a wired network. The link which is used is a duplex link with queue size 5. The other parameters of the link are listed below. Design a simple TCL script in NS-2 for transferring FTP traffic having the following characteristics.

• Packet Size: 1000

• Rate: 1

• Interval: 150

Also, demonstrate various queuing mechanisms and make a comparative analysis of various queuing techniques. (Using trace file) (DropTail, RED, SFQ,a nd FQ)



**Demonstration of NS2**

To implement TCL script to simulate link state routing (Dijkstra algorithm) in ns2

**Step-1: Initializing the network**:

set ns [new Simulator]

$ns rtproto LS

**Step-2: Creating number of nodes:**

set node1 [$ns node]

set node2 [$ns node]

set node3 [$ns node]

set node4 [$ns node]

set node5 [$ns node]

set node6 [$ns node]

set node7 [$ns node]

**Step-3: Creating the trace file:**

set tf [open out.tr w]

$ns trace-all $tf

set nf [open out.nam w]

$ns namtrace-all $nf

**Step-4: Labeling the nodes:**

$node1 label "node 1"

$node1 label "node 2"

$node1 label "node 3"

$node1 label "node 4"

$node1 label "node 5"

$node1 label "node 6"

$node1 label "node 7"

$node1 label-color blue

$node2 label-color red

$node3 label-color red

$node4 label-color blue

$node5 label-color blue

$node6 label-color blue

$node7 label-color blue

**Step-5: Creating duplex links:**

 to create duplex links between the nodes forming a ring in the end. This can be achieved by using the duplex-link instance along with specifying three parameters: data rate (1.5Mb), delay (10ms) and kind of queue (**DropTail**).

$ns duplex-link $node1 $node2 1.5Mb 10ms DropTail

$ns duplex-link $node2 $node3 1.5Mb 10ms DropTail

$ns duplex-link $node3 $node4 1.5Mb 10ms DropTail

$ns duplex-link $node4 $node5 1.5Mb 10ms DropTail

$ns duplex-link $node5 $node6 1.5Mb 10ms DropTail

$ns duplex-link $node6 $node7 1.5Mb 10ms DropTail

$ns duplex-link $node7 $node1 1.5Mb 10ms DropTail

**Step-6: Orient the links between the nodes:**

to orient the links between the nodes appropriately to obtain proper alignment. The **duplex-link-op** instance is used for the same.

$ns duplex-link-op $node1 $node2 orient left-down

$ns duplex-link-op $node2 $node3 orient left-down

$ns duplex-link-op $node3 $node4 orient right-down

$ns duplex-link-op $node4 $node5 orient right

$ns duplex-link-op $node5 $node6 orient right-up

$ns duplex-link-op $node6 $node7 orient left-up

$ns duplex-link-op $node7 $node1 orient left-up

**Step-7: Attaching TCP agents:**

to attach TCP agents (using attach-agent) at two nodes let’s say node 1 and node 4. For this create the source and sink objects and connecting them using connect instance.

set tcp2 [new Agent/TCP]

$ns attach-agent $node1 $tcp2

set sink2 [new Agent/TCPSink]

$ns attach-agent $node4 $sink2

$ns connect $tcp2 $sink2

**Step-8: Creating FTP traffic:**

to create FTP traffic and attach to TCP source. The traffic then flows across node 1 and node 4. We can do this by creating an FTP agent and attaching it to tcp2.

set traffic\_ftp2 [new Application/FTP]

$traffic\_ftp2 attach-agent $tcp2

**Step-9: Adding a finish procedure:**

The next step is to add a finish procedure to flush all data into trace file and then and then run the nam file.

proc finish{} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam &

exit 0

**Step-10: Scheduling the FTP:**

The final step is to schedule the FTP traffic at the required time intervals. Here we can disable the link between any pair of nodes at a certain timestamp using **rtmodel-at** instance and then enable it after a certain time. This is majorly done for testing purposes. Here we have disabled the link between nodes 2 and 3. The program ends with the run command.

$ns at 0.5 "traffic\_ftp2 start"

$ns rtmodel-at 1.0 down $node2 $node3

$ns rtmodel-at 2.0 up $node2 $node3

$ns at 3.0 "traffic\_ftp2 start"

$ns at 4.0 "traffic\_ftp2 stop"

$ns at 5.0 "finish"

$ns run