Program No. 3:Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.

Program Objective:

- Understand the Implementation of the Ethernet LAN using n nodes.

```
#Create Simulator
set ns [new Simulator]
#Use colors to differentiate the traffics
$ns color 1 Blue
$ns color 2 Red
set ntrace [open prog6.tr w]
$ns trace-all $ntrace
set namfile [open prog6.nam w]
$ns namtrace-all $namfile
#Use some flat file to create congestion graph windows
set File0 [open WinFile0 w]
set File1 [open WinFile1 w]

#Finish Procedure
proc Finish {} {
global ns ntrace namfile
$ns flush-trace
close $ntrace
close $namfile
exec nam prog6.nam &
#Plot the Congestion Window graph using xgraph
exec xgraph WinFile0 WinFile1 &
exit 0
}

#Plot Window Procedure
proc PlotWindow {tcpSource file} {
global ns
set time 10.0
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "PlotWindow $tcpSource $file"
}
#Create 6 nodes
for {set i 0} {$i<6} {incr i} {
set n($i) [$ns node]
}
```

```
#Create duplex links between the nodes
$ns duplex-link $n(0) $n(2) 2Mb 10ms DropTail
$ns duplex-link $n(1) $n(2) 2Mb 10ms DropTail
$ns duplex-link $n(2) $n(3) 0.6Mb 100ms DropTail
#Nodes n(3) , n(4) and n(5) are considered in a LAN

#Creates a Lan from a set of nodes, Bandwidth, delay characteristics along with link
#layer,
#interface queue, Mac Layer and channel type for the LAN are defined

set lan [$ns newLan "$n(3) $n(4) $n(5)" 0.5Mb 40ms LL Queue/DropTail MAC/802_3 Channel]

#Orientation to the nodes
$ns duplex-link-op $n(0) $n(2) orient right-down
$ns duplex-link-op $n(1) $n(2) orient right-up
$ns duplex-link-op $n(2) $n(3) orient right

#Setup queue between n(2) and n(3) and monitor the queue
$ns queue-limit $n(2) $n(3) 20

#Set error model on link n(2) to n(3)
set loss_module [new ErrorModel]
$loss_module ranvar [new RandomVariable/Uniform]
$loss_module drop-target [new Agent/Null]
$ns lossmodel $loss_module $n(2) $n(3)

#Set up the TCP connection between n(0) and n(4)
set tcp0 [new Agent/TCP/Newreno]
$tcp0 set fid_ 1
$tcp0 set window_ 8000
$tcp0 set packetSize_ 552
$ns attach-agent $n(0) $tcp0

set sink0 [new Agent/TCPSink/DelAck]
$ns attach-agent $n(4) $sink0
$ns connect $tcp0 $sink0

#Apply FTP Application over TCP
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ftp0 set type_ FTP

#Set up another TCP connection between n(5) and n(1)
set tcp1 [new Agent/TCP/Newreno]
```

```
$tcp1 set fid_ 2
$tcp1 set window_ 8000
$tcp1 set packetSize_ 552
$ns attach-agent $n(5) $tcp1

set sink1 [new Agent/TCPSink/DelAck]
$ns attach-agent $n(1) $sink1
$ns connect $tcp1 $sink1

#Apply FTP application over TCP
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ftp1 set type_ FTP
#Schedule Events
$ns at 0.1 "$ftp0 start"
$ns at 0.1 "PlotWindow $tcp0 $File0"
$ns at 0.5 "$ftp1 start"
$ns at 0.5 "PlotWindow $tcp1 $File1"
$ns at 25.0 "$ftp0 stop"
$ns at 25.1 "$ftp1 stop"
$ns at 25.2 "Finish"
#Run the simulation
$ns run
```

Output

Steps for execution

1. Open gedit and type program. Program name should have the extension “ .tcl
   student@cnpc022:~/ student$ gedit prog6.tcl
2. Save the program.
3. Run the simulation program
   student@cnpc022:~/ student$ ns prog6.tcl
4. Here “ns” indicates network simulator. We get the topology shown in the snapshot.
5. Now press the play button in the simulation window and the simulation will begin.
6. To see the trace file contents open the file as ,
   student@cnpc022:~/ student$ gedit prog6.tr