

### **Program 3: Simulate an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.**

```
#Create Simulator
set ns [new Simulator]

#Use colors to differentiate the traffics
$ns color 1 Blue
$ns color 2 Red

#Open trace and NAM trace file
set ntrace [open prog5.tr w]
$ns trace-all $ntrace
set namfile [open prog5.nam w]
$ns namtrace-all $namfile

#Use some flat file to create congestion graph windows
set winFile0 [open WinFile0 w]
set winFile1 [open WinFile1 w]

#Finish Procedure
proc Finish {} {
    #Dump all trace data and Close the files
    global ns ntrace namfile
    $ns flush-trace
    close $ntrace
    close $namfile

    #Execute the NAM animation file
    exec nam prog5.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 0.1
    set now [$ns now]
    set cwnd [$tcpSource set cwnd_]

    # To plot graph over x and y axis
    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) 1.0Mb 100ms DropTail
```

```

#Nodes n(3) , n(4) and n(5) are considered in a LAN
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
$ns duplex-link-op $n(2) $n(3) queuePos 0.5

#Set error model on link n(2) to n(3) (optional- to analyse the amt of drop removed pkts in tr file)
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 $winFile0"
$ns at 0.5 "$ftp1 start"
$ns at 0.5 "PlotWindow $tcp1 $winFile1"
$ns at 25.0 "$ftp0 stop"
$ns at 25.1 "$ftp1 stop"
$ns at 25.2 "Finish"

```

#Run the simulation  
\$ns run