

Computer Networks Laboratory
UE23CS252B
4th Semester, Academic Year 2025

Date:27/03/2025

| | | |
|----------------------|--------------------|------------|
| Name: Pranav Hemanth | SRN: PES1UG23CS433 | Section: G |
|----------------------|--------------------|------------|

Lab#____5_____ Task Number: ____1____

TITLE: Congestion Window Experiment

Code:

Tahoe:

```
# <http://wushoupong.googlepages.com/nsg>
#=====
#   Simulation parameters setup
#=====
set val(stop) 20;# time of simulation end
#=====
#   Initialization
#=====
#Create a ns simulator
set ns [new Simulator]
#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile
#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
```

```

set file6 [open cw6.out w]
puts $file6 "Title = Congestion Window @ Node 6"
puts $file6 "title_x = Time in Sec"
puts $file6 "title_y = Window Size"
set file2 [open cw2.out w]
puts $file2 "Title = Congestion Window @ Node 2"
puts $file2 "title_x = Time in Sec"
puts $file2 "title_y = Window Size"
#=====
#    Nodes Definition
#=====
#Create 7 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
    Links Definition
#=====
#Createlinks between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 100
$ns duplex-link $n0 $n2 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n2 50
$ns duplex-link $n0 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n3 50
$ns duplex-link $n0 $n4 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n4 50
$ns duplex-link $n1 $n5 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n5 50
$ns duplex-link $n1 $n6 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n6 50
#=====
#    Agents Definition
#=====
#Setup a TCP connection (n6 to n4)
set tcp0 [new Agent/TCP]
$ns attach-agent $n6 $tcp0
set sink0 [new Agent/TCPSink]
$ns attach-agent $n4 $sink0
$ns connect $tcp0 $sink0
$tcp0 set packetSize_ 3000
#Setup a TCP connection (n2 to n5)
set tcp1 [new Agent/TCP]

```

```

$ns attach-agent $n2 $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $n5 $sink1
$ns connect $tcp1 $sink1
$tcp1 set packetSize_ 1500
#Setup a UDP connection (n6 to n4)
set udp0 [new Agent/UDP]
$ns attach-agent $n6 $udp0
set sink1 [new Agent/LossMonitor]
$ns attach-agent $n4 $sink1
$ns connect $udp0 $sink1
$udp0 set packetSize_ 1500
#=====
#   Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
#Setup a FTP Application over 2nd TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
#Setup a CBR Application over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set rate_ 99.9Mb
$cbr0 set random_ null
proc record {} {
    global tcp0 tcp1 file6 file2

    #Get an instance of the simulator
    set ns [Simulator instance]

    #Set the time after which the procedure should be called again
    set time 0.5
    #How many bytes have been received by the traffic sinks?
    set cw6 [$tcp0 set cwnd_]
    set cw2 [$tcp1 set cwnd_]

    #Get the current time
    set now [$ns now]

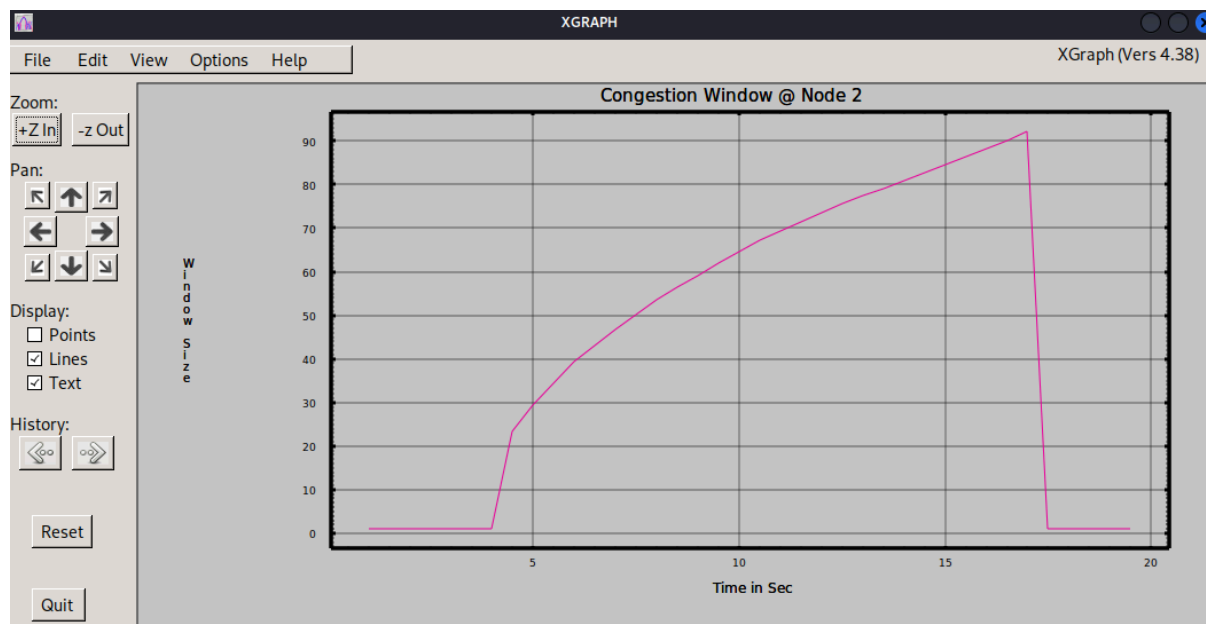
    puts $file6 "$now $cw6"
    puts $file2 "$now $cw2"
    #Re-schedule the procedure
    $ns at [expr $now+$time] "record"
}

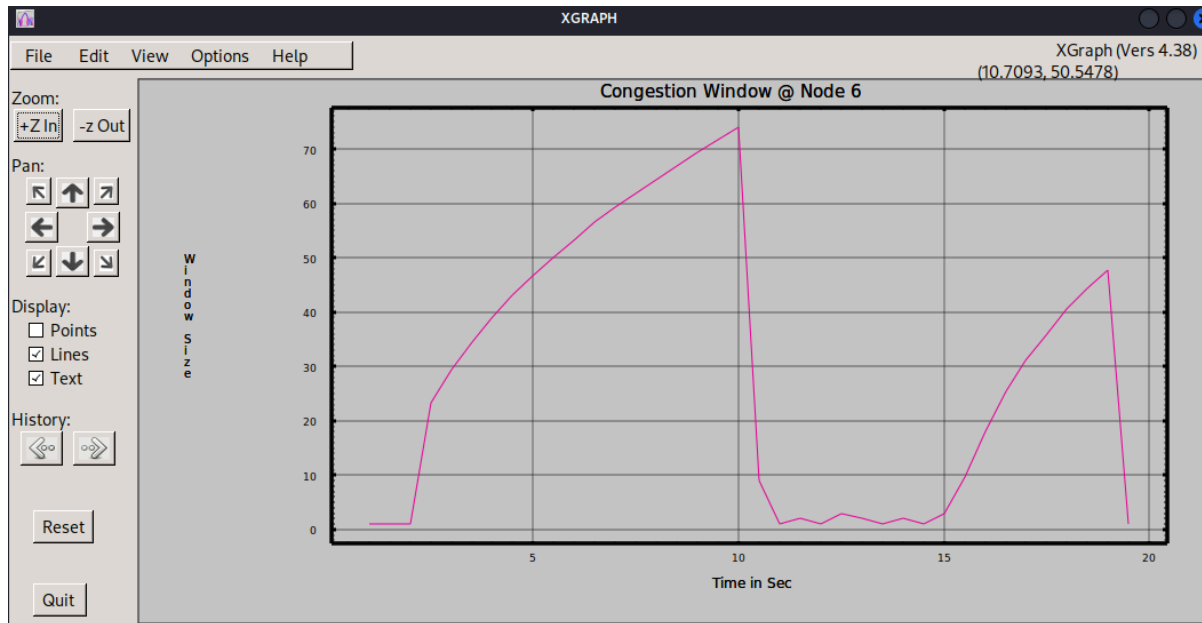
```

```

$ns at 1.0 "record"
$ns at 2.0 "$ftp0 start"
$ns at 4.0 "$ftp1 start"
$ns at 10.0 "$cbr0 start"
$ns at 15.0 "$cbr0 stop"
$ns at 17.0 "$ftp1 stop"
$ns at 19.0 "$ftp0 stop"
#=====
#   Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns namfile tracefile file6 file2
    $ns flush-trace
    close $tracefile
    close $namfile
    close $file6
    close $file2
    exec nam out.nam &
    exec /home/pranav/Downloads/xgraph/bin/xgraph cw6.out &
    exec /home/pranav/Downloads/xgraph/bin/xgraph cw2.out &
    exit 0
}
$ns at $val(stop) "finish"
$ns run

```





Reno:

```
# <http://wushoupong.googlepages.com/nsg>
#=====
#   Simulation parameters setup
#=====
set val(stop) 20;# time of simulation end
#=====
#   Initialization
#=====
#Create a ns simulator
set ns [new Simulator]
#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile
#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
set file6 [open cw6.out w]
puts $file6 "Title = Congestion Window @ Node 6"
puts $file6 "title_x = Time in Sec"
puts $file6 "title_y = Window Size"
set file2 [open cw2.out w]
puts $file2 "Title = Congestion Window @ Node 2"
puts $file2 "title_x = Time in Sec"
```

```

puts $file2 "title_y = Window Size"
#=====
#    Nodes Definition
#=====
#Create 7 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
    Links Definition
#=====
#Createlinks between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 100
$ns duplex-link $n0 $n2 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n2 50
$ns duplex-link $n0 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n3 50
$ns duplex-link $n0 $n4 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n4 50
$ns duplex-link $n1 $n5 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n5 50
$ns duplex-link $n1 $n6 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n6 50
#=====
#    Agents Definition
#=====
#Setup a TCP connection (n6 to n4)
set tcp0 [new Agent/TCP/Reno]
$ns attach-agent $n6 $tcp0
set sink0 [new Agent/TCPSink]
$ns attach-agent $n4 $sink0
$ns connect $tcp0 $sink0
$tcp0 set packetSize_ 4000
#Setup a TCP connection (n2 to n5)
set tcp1 [new Agent/TCP/Reno]
$ns attach-agent $n2 $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $n5 $sink1
$ns connect $tcp1 $sink1
$tcp1 set packetSize_ 2000
#Setup a UDP connection (n6 to n4)
set udp0 [new Agent/UDP]

```

```

$ns attach-agent $n6 $udp0
set sink1 [new Agent/LossMonitor]
$ns attach-agent $n4 $sink1
$ns connect $udp0 $sink1
$udp0 set packetSize_ 1500
#=====
#   Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
#Setup a FTP Application over 2nd TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
#Setup a CBR Application over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set rate_ 99.9Mb
$cbr0 set random_ null
proc record {} {
    global tcp0 tcp1 file6 file2

    #Get an instance of the simulator
    set ns [Simulator instance]

    #Set the time after which the procedure should be called again
    set time 0.5
    #How many bytes have been received by the traffic sinks?
    set cw6 [$tcp0 set cwnd_]
    set cw2 [$tcp1 set cwnd_]

    #Get the current time
    set now [$ns now]

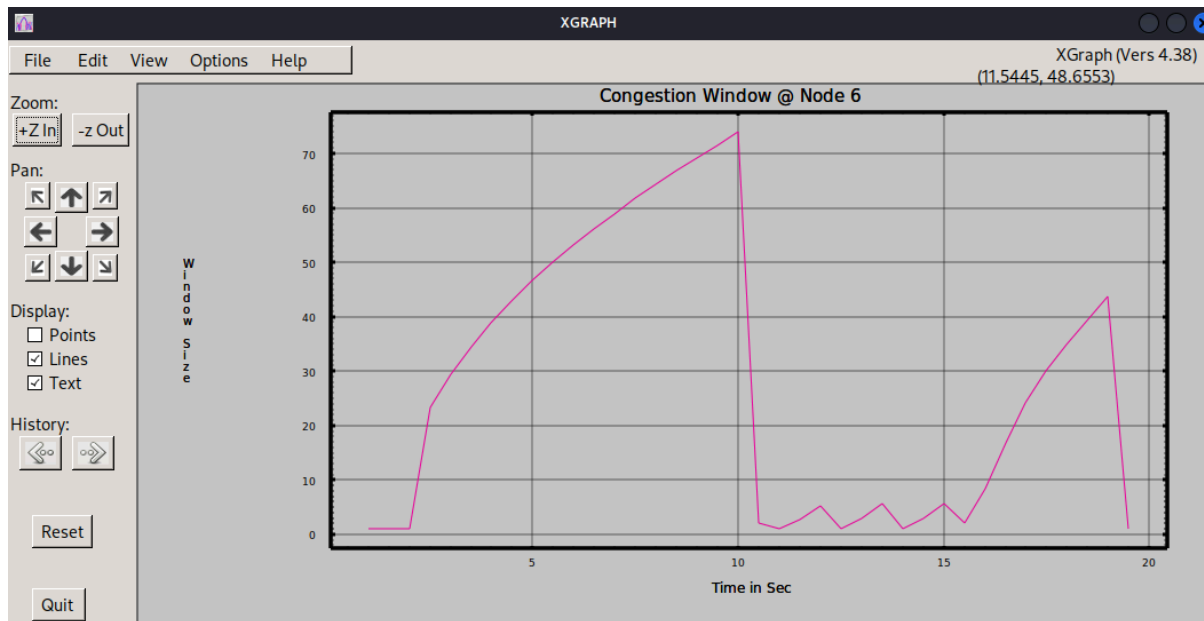
    puts $file6 "$now $cw6"
    puts $file2 "$now $cw2"
    #Re-schedule the procedure
    $ns at [expr $now+$time] "record"
}
$ns at 1.0 "record"
$ns at 2.0 "$ftp0 start"
$ns at 4.0 "$ftp1 start"
$ns at 10.0 "$cbr0 start"
$ns at 15.0 "$cbr0 stop"
$ns at 17.0 "$ftp1 stop"
$ns at 19.0 "$ftp0 stop"

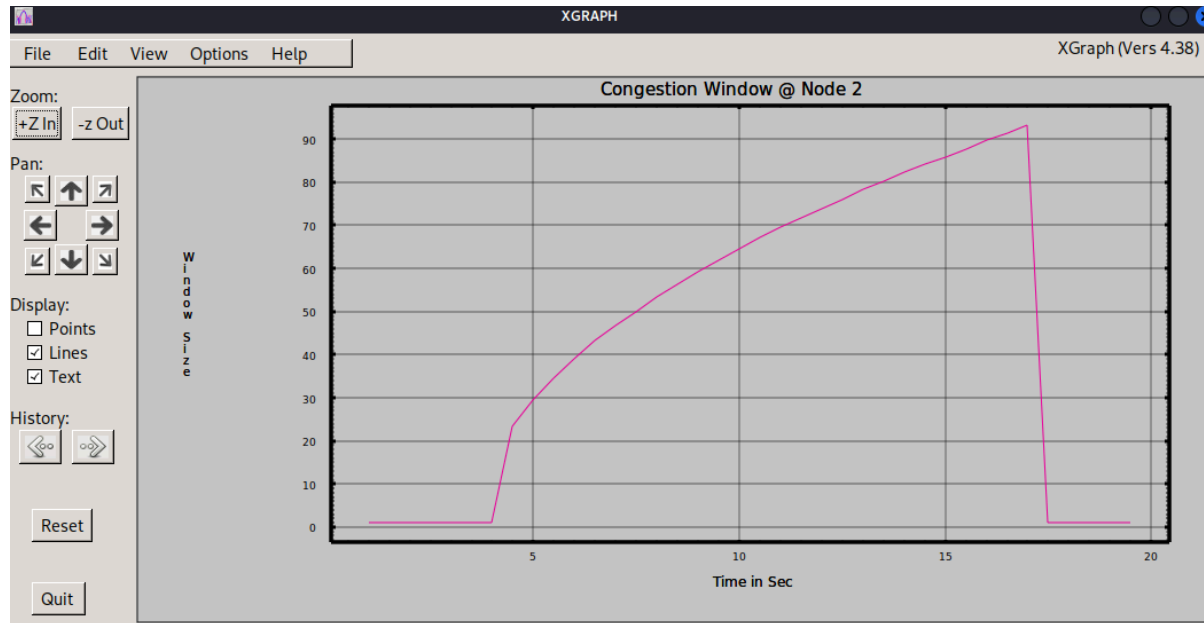
```

```

#=====
#   Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns namfile tracefile file6 file2
    $ns flush-trace
    close $tracefile
    close $namfile
    close $file6
    close $file2
    exec nam out.nam &
    exec /home/pranav/Downloads/xgraph/bin/xgraph cw6.out &
    exec /home/pranav/Downloads/xgraph/bin/xgraph cw2.out &
    exit 0
}
$ns at $val(stop) "finish"
$ns run

```





Vegas:

```
# <http://wushoupong.googlepages.com/nsg>
#=====
#   Simulation parameters setup
#=====
set val(stop) 20;# time of simulation end
#=====
#   Initialization
#=====
#Create a ns simulator
set ns [new Simulator]
#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile
#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
set file6 [open cw6.out w]
puts $file6 "Title = Congestion Window @ Node 6"
puts $file6 "title_x = Time in Sec"
puts $file6 "title_y = Window Size"
set file2 [open cw2.out w]
puts $file2 "Title = Congestion Window @ Node 2"
puts $file2 "title_x = Time in Sec"
puts $file2 "title_y = Window Size"
```

```

#=====
#   Nodes Definition
#=====
#Create 7 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]

    Links Definition
#=====
#Createlinks between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 100
$ns duplex-link $n0 $n2 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n2 50
$ns duplex-link $n0 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n3 50
$ns duplex-link $n0 $n4 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n4 50
$ns duplex-link $n1 $n5 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n5 50
$ns duplex-link $n1 $n6 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n6 50
#=====
#   Agents Definition
#=====
#Setup a TCP connection (n6 to n4)
set tcp0 [new Agent/TCP/Vegas]
$ns attach-agent $n6 $tcp0
set sink0 [new Agent/TCPSink]
$ns attach-agent $n4 $sink0
$ns connect $tcp0 $sink0
$tcp0 set packetSize_ 2000
#Setup a TCP connection (n2 to n5)
set tcp1 [new Agent/TCP/Vegas]
$ns attach-agent $n2 $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $n5 $sink1
$ns connect $tcp1 $sink1
$tcp1 set packetSize_ 1000
#Setup a UDP connection (n6 to n4)
set udp0 [new Agent/UDP]
$ns attach-agent $n6 $udp0

```

```

set sink1 [new Agent/LossMonitor]
$ns attach-agent $n4 $sink1
$ns connect $udp0 $sink1
$udp0 set packetSize_ 1000
#=====
#   Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
#Setup a FTP Application over 2nd TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
#Setup a CBR Application over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set rate_ 99.9Mb
$cbr0 set random_ null
proc record {} {
    global tcp0 tcp1 file6 file2

    #Get an instance of the simulator
    set ns [Simulator instance]

    #Set the time after which the procedure should be called again
    set time 0.5
    #How many bytes have been received by the traffic sinks?
    set cw6 [$tcp0 set cwnd_]
    set cw2 [$tcp1 set cwnd_]

    #Get the current time
    set now [$ns now]

    puts $file6 "$now $cw6"
    puts $file2 "$now $cw2"
    #Re-schedule the procedure
    $ns at [expr $now+$time] "record"
}
$ns at 1.0 "record"
$ns at 2.0 "$ftp0 start"
$ns at 4.0 "$ftp1 start"
$ns at 10.0 "$cbr0 start"
$ns at 15.0 "$cbr0 stop"
$ns at 17.0 "$ftp1 stop"
$ns at 19.0 "$ftp0 stop"
#=====

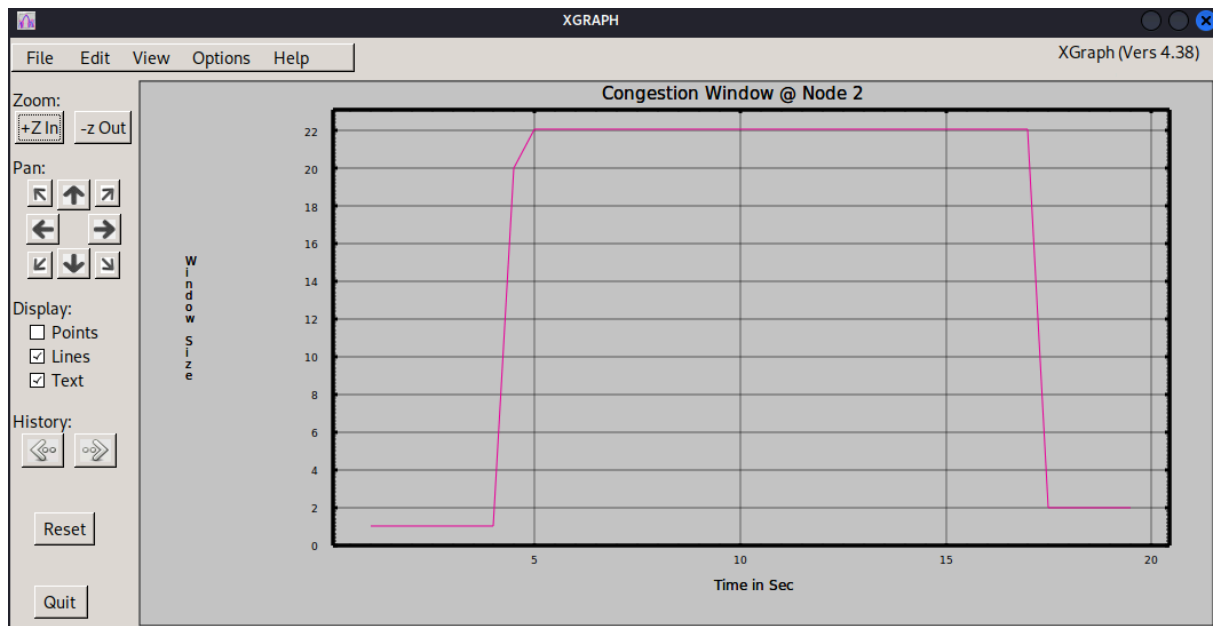
```

```

# Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns namfile tracefile file6 file2
    $ns flush-trace
    close $tracefile
    close $namfile
    close $file6
    close $file2
    exec nam out.nam &
    exec /home/pranav/Downloads/xgraph/bin/xgraph cw6.out &
    exec /home/pranav/Downloads/xgraph/bin/xgraph cw2.out &
    exit 0
}
$ns at $val(stop) "finish"
$ns run

```





Disclaimer:

- The programs and output submitted are duly written, verified and executed by me.
- I have not copied from any of my peers nor from external resources such as the internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Name: Pranav Hemanth

SRN: PES1UG23CS433

Section: G

Date: 27/03/25