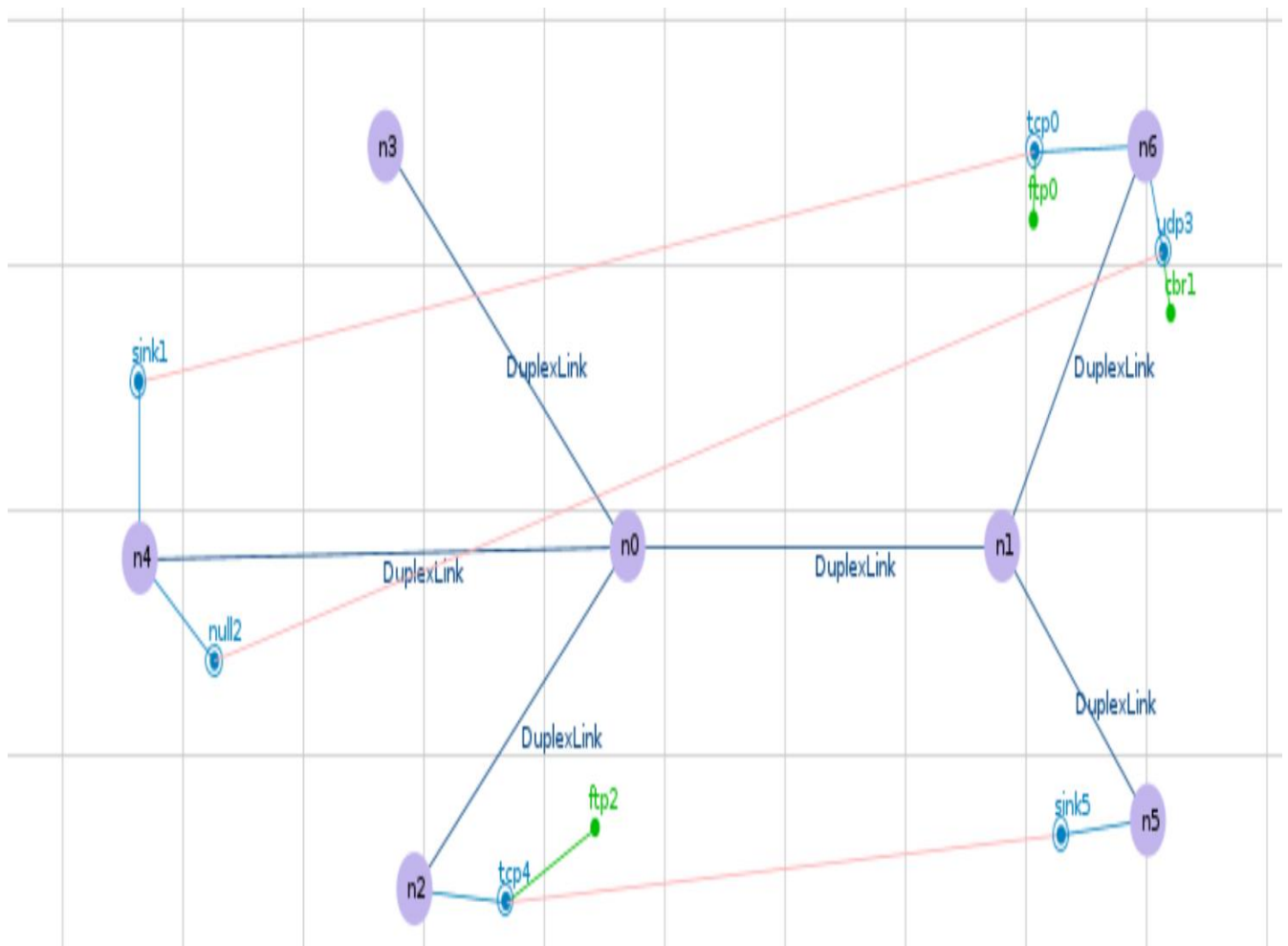


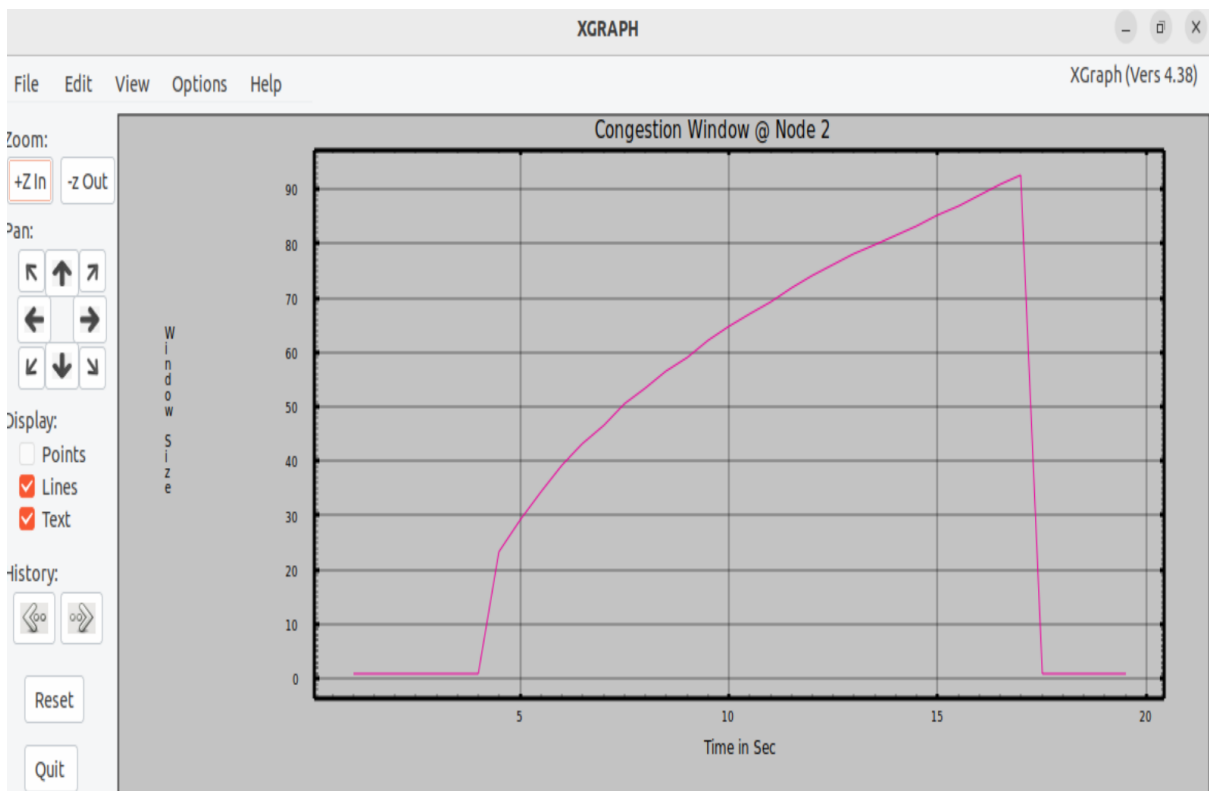
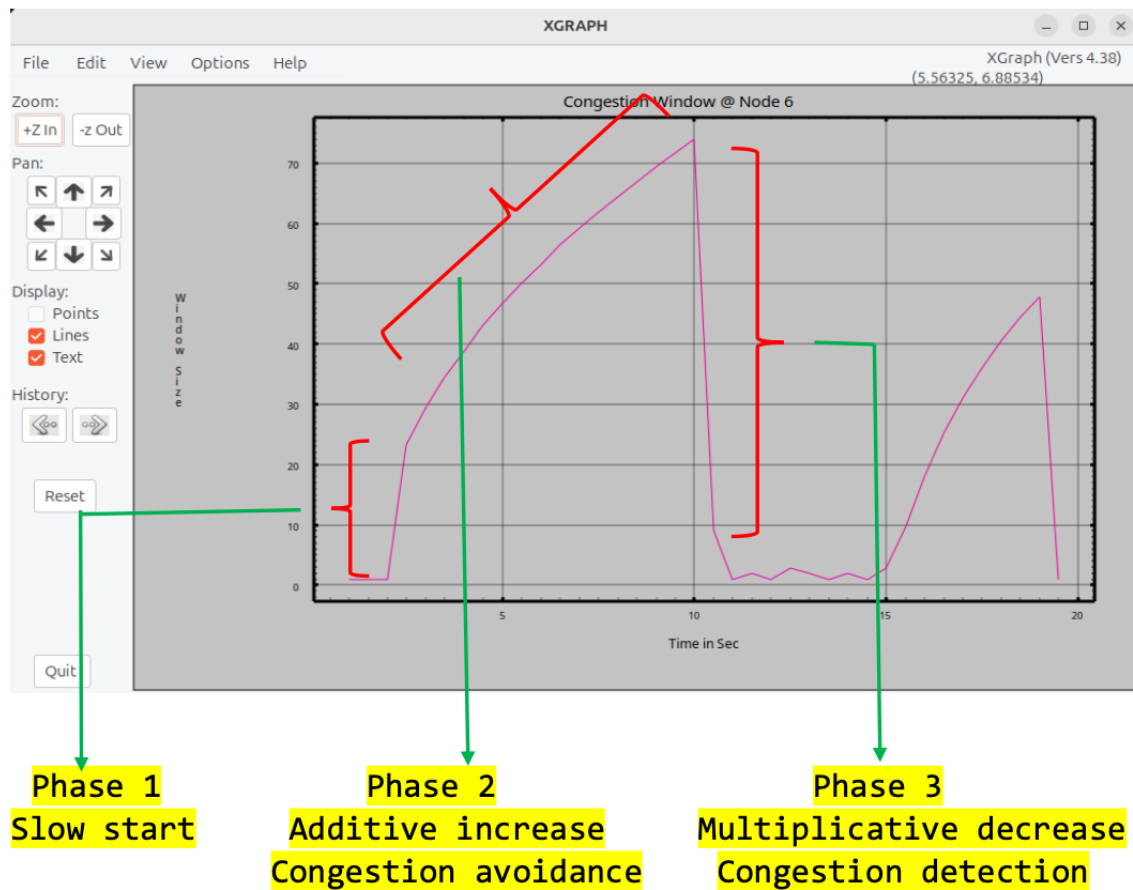
# UE23CS252B COMPUTER NETWORKS

LAB #04 29-0-2025 PESU ECC

ABHISHEK P PES2UG23AM002 CSE(AIML)-4A



## THREE PHASES OF TCP CONGESTION CONTROL



Phases of TCP Congestion Control:

## 1. Slow Start (SS) Phase:

- In this phase, the congestion window size increases exponentially.
- In the graph, this is the steep rising part at the beginning (from around 0 to 5 seconds).

## 2. Congestion Avoidance (CA) Phase:

- The window growth slows down to a linear increase.
- In the graph, this corresponds to the gradual increase from around 5 to 15 seconds.

## 3. Congestion Detection & Recovery (Fast Recovery or Timeout):

- A sudden drop in the congestion window indicates that congestion occurred.
- In the graph, the sharp drop around 15 seconds represents packet loss and congestion control kicking in.

# This script is created by NSG2 beta1

# <<http://wushoupong.googlepages.com/nsg>>

#=====

```

# Simulation parameters setup
#=====
set val(stop) 20.0 ;# time of simulation end
#=====
# Initialization
#=====
#Create a ns simulator
set ns [new Simulator]
#Open the NS trace file
set tracefile [open NEW.tr w]
$ns trace-all $tracefile
#Open the NAM trace file
set namfile [open NEW.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 $n3 $n0 100.0Mb 10ms
DropTail
$ns queue-limit $n3 $n0 50
$ns duplex-link $n4 $n0 100.0Mb 10ms DropTail
$ns queue-limit $n4 $n0 50
$ns duplex-link $n2 $n0 100.0Mb 10ms DropTail
$ns queue-limit $n2 $n0 50
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 100
$ns duplex-link $n1 $n6 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n6 50
$ns duplex-link $n1 $n5 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n5 50
#Give node position (for NAM)
$ns duplex-link-op $n3 $n0 orient right-down
$ns duplex-link-op $n4 $n0 orient right
$ns duplex-link-op $n2 $n0 orient right-up
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n6 orient right-up
$ns duplex-link-op $n1 $n5 orient right-down
#=====
# Agents Definition
#=====
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n6 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize
3000

—
#Setup a TCP connection
set tcp4 [new Agent/TCP]

```

```

$ns attach-agent $n2 $tcp4
set sink5 [new Agent/TCPSink]
$ns attach-agent $n5 $sink5
$ns connect $tcp4 $sink5
$tcp4 set packetSize
1000

—
#Setup a UDP connection
set udp3 [new Agent/UDP]
$ns attach-agent $n6 $udp3
set null2 [new Agent/Null]
$ns attach-agent $n4 $null2
$ns connect $udp3 $null2
$udp3 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 TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp4
#Setup a CBR Application over UDP connection
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp3
$cbr1 set packetSize
1000

—
$cbr1 set rate
99.9Mb

—
$cbr1 set random
null

—
proc record {} {
global tcp0 tcp4 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 [$tcp0 set cwnd
_]
set cw2 [$tcp4 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"
$ns at 10.0 "$cbr1 start"
$ns at 15.0 "$cbr1 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/abhishekp/Downloads/xgraph/xgraph/bin/xgraph cw6.out
& exec/home/abhishekp/Downloads/xgraph/xgraph/bin/xgraph
cw2.out & exit 0
}
$ns at $val(stop) "finish"

```



\$ns run