**1)To type tcl: gedit prog1.tcl**

**2)to get animation: ns prog1.tcl**

**3)to type awk :gedit prog1.awk**

**4)to get pckt drop: awk -f prog1.awk prog1.tr**

**1.Implement three nodes point – to – point network with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped.**

# Create a simulator object (capital S in Simulator)

set ns [new Simulator]

# Open a NAM trace file in write mode

set nf [open lab1.nam w]

$ns namtrace-all $nf

# Open a trace file in write mode

set tf [open lab1.tr w]

$ns trace-all $tf

# Procedure to finish simulation

proc finish { } {

global ns nf tf

$ns flush-trace ;# Flush trace file contents

close $nf ;# Close NAM file

close $tf ;# Close trace file

exec nam lab1.nam & ;# Execute NAM animator

exit 0 ;# Exit simulation

}

# Create 4 nodes (n0, n1, n2, n3)

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

# Define duplex links with bandwidth (Mb), delay (ms), and queue type DropTail

$ns duplex-link $n0 $n2 200Mb 10ms DropTail

$ns duplex-link $n1 $n2 100Mb 5ms DropTail

$ns duplex-link $n2 $n3 1Mb 1000ms DropTail

# Set queue size (limit = 10 packets) on links

$ns queue-limit $n0 $n2 10

$ns queue-limit $n1 $n2 10

# Define UDP agent at node n0

set udp0 [new Agent/UDP]

$ns attach-agent $n0 $udp0

# Define CBR traffic at n0

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize\_ 500 ;# Packet size 500 bytes

$cbr0 set interval\_ 0.005 ;# Interval between packets

$cbr0 attach-agent $udp0 ;# Attach CBR to UDP agent

# Define UDP agent at node n1

set udp1 [new Agent/UDP]

$ns attach-agent $n1 $udp1

# Define CBR traffic at n1

set cbr1 [new Application/Traffic/CBR]

$cbr1 attach-agent $udp1

# Define UDP agent at node n2

set udp2 [new Agent/UDP]

$ns attach-agent $n2 $udp2

# Define CBR traffic at n2

set cbr2 [new Application/Traffic/CBR]

$cbr2 attach-agent $udp2

# Define Null agent (sink) at node n3

set null0 [new Agent/Null]

$ns attach-agent $n3 $null0

# Connect UDP agents to sink

$ns connect $udp0 $null0

$ns connect $udp1 $null0

# Start traffic at different times

$ns at 0.1 "$cbr0 start"

$ns at 0.2 "$cbr1 start"

# End simulation at 1 second

$ns at 1.0 "finish"

# Run the simulation

$ns run

# Create a simulator object (capital S in Simulator)

set ns [new Simulator]

# Open a NAM trace file in write mode

set nf [open lab1.nam w]

$ns namtrace-all $nf

# Open a trace file in write mode

set tf [open lab1.tr w]

$ns trace-all $tf

# Procedure to finish simulation

proc finish { } {

global ns nf tf

$ns flush-trace ;# Flush trace file contents

close $nf ;# Close NAM file

close $tf ;# Close trace file

exec nam lab1.nam & ;# Execute NAM animator

exit 0 ;# Exit simulation

}

# Create 4 nodes (n0, n1, n2, n3)

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

# Define duplex links with bandwidth (Mb), delay (ms), and queue type DropTail

$ns duplex-link $n0 $n2 200Mb 10ms DropTail

$ns duplex-link $n1 $n2 100Mb 5ms DropTail

$ns duplex-link $n2 $n3 1Mb 1000ms DropTail

# Set queue size (limit = 10 packets) on links

$ns queue-limit $n0 $n2 10

$ns queue-limit $n1 $n2 10

# Define UDP agent at node n0

set udp0 [new Agent/UDP]

$ns attach-agent $n0 $udp0

# Define CBR traffic at n0

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize\_ 500 ;# Packet size 500 bytes

$cbr0 set interval\_ 0.005 ;# Interval between packets

$cbr0 attach-agent $udp0 ;# Attach CBR to UDP agent

# Define UDP agent at node n1

set udp1 [new Agent/UDP]

$ns attach-agent $n1 $udp1

# Define CBR traffic at n1

set cbr1 [new Application/Traffic/CBR]

$cbr1 attach-agent $udp1

# Define UDP agent at node n2

set udp2 [new Agent/UDP]

$ns attach-agent $n2 $udp2

# Define CBR traffic at n2

set cbr2 [new Application/Traffic/CBR]

$cbr2 attach-agent $udp2

# Define Null agent (sink) at node n3

set null0 [new Agent/Null]

$ns attach-agent $n3 $null0

# Connect UDP agents to sink

$ns connect $udp0 $null0

$ns connect $udp1 $null0

# Start traffic at different times

$ns at 0.1 "$cbr0 start"

$ns at 0.2 "$cbr1 start"

# End simulation at 1 second

$ns at 1.0 "finish"

# Run the simulation

$ns run

AWK Script:

# BEGIN block: executed once before processing input

BEGIN {

c = 0; # Initialize counter for dropped packets

}

{

# Check if the first column indicates packet drop (d)

if ($1 == "d") {

c++; # Increment drop counter

printf("%s\t%s\n", $5, $11); # Print source node and packet size

}

}

# END block: executed after processing input

END {

printf("The number of packets dropped = %d\n", c); # Print total drops

}

**How to Run**

1. Save simulation script as **lab1.tcl**
2. ns lab1.tcl

This will generate:

* + lab1.nam → network animation file
  + lab1.tr → trace file

1. Save AWK script as **drop.awk**
2. awk -f drop.awk lab1.tr

Output example:

2 500

1 500

The number of packets dropped = 12