Work Integrated Learning Programmes

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Computer Networks – Group Assignment

Exercise #4: Network Simulations using NS2

Submitted By:

Deepanshu Parnami (2019HS70003)

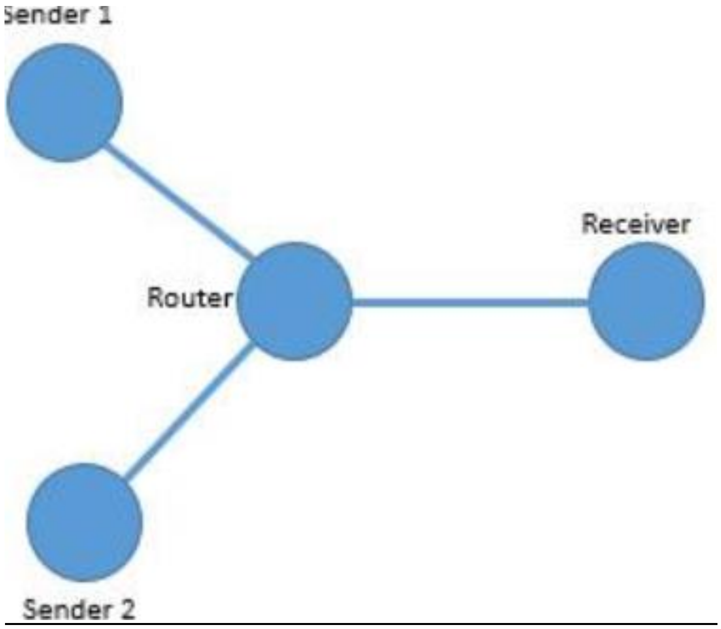
Umesh Kakkar (2019HS70007)

Atul Bhatt (2109HS70012)

Amit Krishna (2019HS70035)

Objective: Simulate different network types using NS22) Simple performance analysis

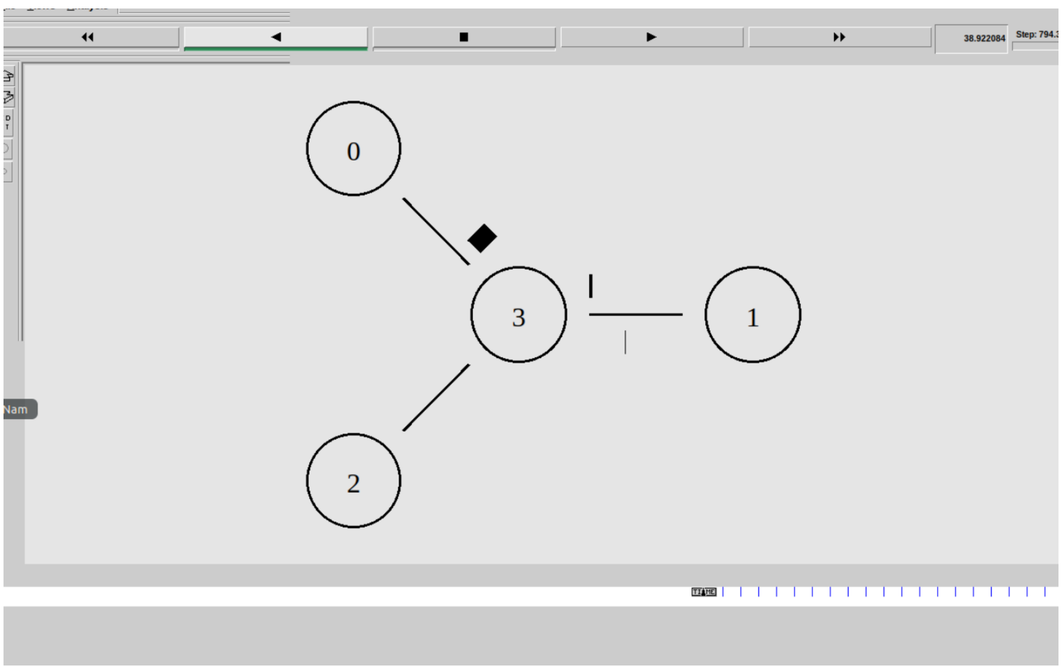
Simulate the following network in NS2. Sender 1 and Sender 2 send data through the Router to the Receiver, which acts as the sink.



TCL Script – TCP

```
# time of simulation end
set val(stop)   60.0

#Define tracefile o/p
set f0 [open __out0.tr w]
set f1 [open __out1.tr w]

#Create a ns simulator
set ns [new Simulator]

set namfile [open out.nam w]
$ns namtrace-all $namfile

#Create 4 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#links between nodes
$ns duplex-link $n3 $n0 10.0Mb 10ms DropTail
$ns queue-limit $n3 $n0 50
```

```
$ns duplex-link $n2 $n3 10.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 50
$ns duplex-link $n3 $n1 10.0Mb 10ms DropTail
$ns queue-limit $n3 $n1 50

#node positioning
$ns duplex-link-op $n3 $n0 orient left-up
$ns duplex-link-op $n2 $n3 orient right-up
$ns duplex-link-op $n3 $n1 orient right


#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink0 [new Agent/TCPSink]
$ns attach-agent $n1 $sink0
$tcp0 set packetSize_ 512
#$tcp0 set window_ 1000
#$tcp0 tracevar cwnd_
$ns connect $tcp0 $sink0

set tcp1 [new Agent/TCP]
$ns attach-agent $n2 $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$tcp1 set packetSize_ 512
#$tcp1 set window_ 1000
#$tcp1 tracevar cwnd_
$ns connect $tcp1 $sink1

#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 2.0 "$ftp0 start"
$ns at 60.0 "$ftp0 stop"

set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 1.5 "$ftp1 start"
$ns at 60.0 "$ftp1 stop"

#To get the X,Y Coords for Plotting
proc record {} {
    global sink0 sink1 f0 f1

        set ns [Simulator instance]
    set time 0.5

    set bw0 [$sink0 set bytes_]
    set bw1 [$sink1 set bytes_]

    set now [$ns now]

    puts $f0 "$now [expr $bw0/$time*8/1000000]"
    puts $f1 "$now [expr $bw1/$time*8/1000000]"

    $sink0 set bytes_ 0
    $sink1 set bytes_ 0

    $ns at [expr $now+$time] "record"
}

#Runs at End
proc finish {} {
  global ns namfile f0 f1
  $ns flush-trace
  close $namfile
  close $f0
  close $f1
  exec nam out.nam &
  #exec xgraph __out0.tr __out1.tr -geometry 800x400 &
        exit 0
}

#Setting Congestion and Advertised window
set cwnd0 [$tcp0 set cwnd_ 1000]
set cwnd1 [$tcp1 set cwnd_ 1000]
set awnd0 [$tcp0 set awnd_ 1000]
set awnd1 [$tcp1 set awnd_ 1000]


$ns at 0.0 "record"
$ns at 60.0 "finish"


puts "Congestion Window Size = $cwnd0"
puts "Advertised Window Size = $awnd0"

$ns run
```



TCL Script – XGraph

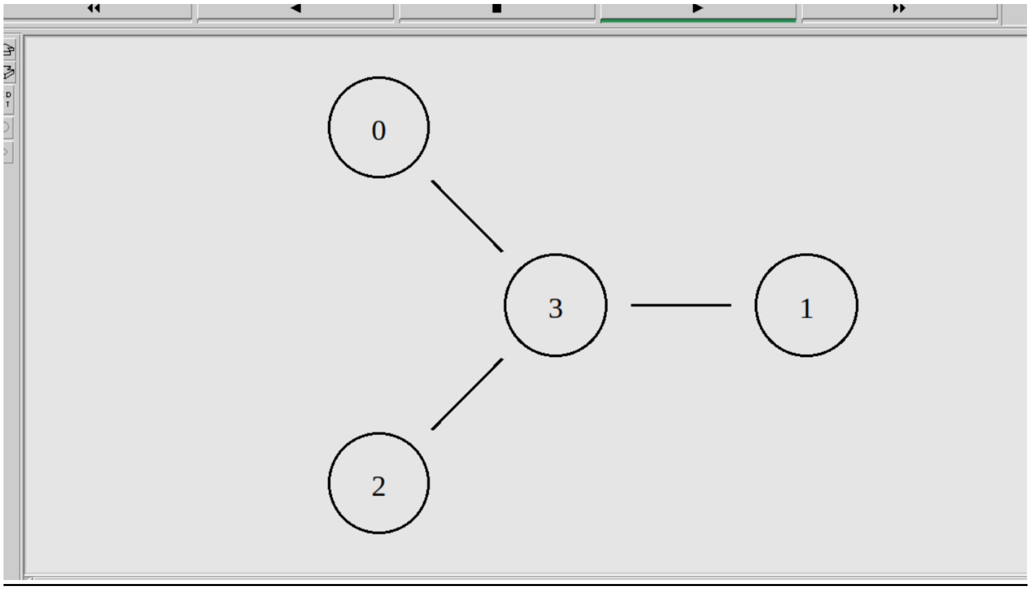
```
# time of simulation end
set val(stop)   60.0

#Define tracefile o/p
set f0 [open __out0.tr w]
set f1 [open __out1.tr w]

#Create a ns simulator
set ns [new Simulator]

set namfile [open out.nam w]
$ns namtrace-all $namfile

#Create 4 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#links between nodes
$ns duplex-link $n3 $n0 10.0Mb 10ms DropTail
$ns queue-limit $n3 $n0 50
$ns duplex-link $n2 $n3 10.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 50
$ns duplex-link $n3 $n1 10.0Mb 10ms DropTail
```

```
$ns queue-limit $n3 $n1 50

#node positioning
$ns duplex-link-op $n3 $n0 orient left-up
$ns duplex-link-op $n2 $n3 orient right-up
$ns duplex-link-op $n3 $n1 orient right


#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink0 [new Agent/TCPSink]
$ns attach-agent $n1 $sink0
$tcp0 set packetSize_ 512
#$tcp0 set window_ 1000
#$tcp0 tracevar cwnd_
$ns connect $tcp0 $sink0

set tcp1 [new Agent/TCP]
$ns attach-agent $n2 $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$tcp1 set packetSize_ 512
#$tcp1 set window_ 1000
#$tcp1 tracevar cwnd_
$ns connect $tcp1 $sink1

#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 2.0 "$ftp0 start"
$ns at 60.0 "$ftp0 stop"

set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 1.5 "$ftp1 start"
$ns at 60.0 "$ftp1 stop"

#To get the X,Y Coords for Plotting
proc record {} {
    global sink0 sink1 f0 f1

        set ns [Simulator instance]

    set time 0.5

    set bw0 [$sink0 set bytes_]
    set bw1 [$sink1 set bytes_]

    set now [$ns now]

    puts $f0 "$now [expr $bw0/$time*8/1000000]"
    puts $f1 "$now [expr $bw1/$time*8/1000000]"

    $sink0 set bytes_ 0
    $sink1 set bytes_ 0

    $ns at [expr $now+$time] "record"
}

#Runs at End
proc finish {} {
  global ns namfile f0 f1
  $ns flush-trace
  close $namfile
  close $f0
  close $f1
  exec xgraph __out0.tr __out1.tr -geometry 800x400 &
        exit 0
}

#Setting Congestion and Advertised window
set cwnd0 [$tcp0 set cwnd_ 1000]
set cwnd1 [$tcp1 set cwnd_ 1000]
set awnd0 [$tcp0 set awnd_ 1000]
set awnd1 [$tcp1 set awnd_ 1000]


$ns at 0.0 "record"
$ns at 60.0 "finish"


puts "Congestion Window Size = $cwnd0"
puts "Advertised Window Size = $awnd0"

$ns run
```

TCL Script – UDP

```
#Create a simulator object
set ns [new Simulator]

$ns color 1 Red
$ns color 2 Blue

#Open the output files
set f [open udp.tr w]
$ns trace-all $f
set nf [open out.nam w]
$ns namtrace-all $nf

#Create 4 nodes
set sender1 [$ns node]
set sender2 [$ns node]
set router [$ns node]
set receiver [$ns node]

#Connect the nodes
$ns duplex-link $sender1 $router 10Mb 10ms DropTail
$ns duplex-link $sender2 $router 10Mb 10ms DropTail
$ns duplex-link $router $receiver 10Mb 10ms DropTail

$ns duplex-link-op $sender1 $router orient right-down
$ns duplex-link-op $sender2 $router orient right-up
$ns duplex-link-op $router $receiver orient right

$ns duplex-link-op $router $receiver queuePos 0.5

set udp0 [new Agent/UDP]
$udp0 set class_ 1
$ns attach-agent $sender1 $udp0

set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 512
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0

set udp1 [new Agent/UDP]
$udp1 set class_ 2
$ns attach-agent $sender2 $udp1

set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 512
```

```
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1

set sink0 [new Agent/LossMonitor]
$ns attach-agent $receiver $sink0

$ns connect $udp0 $sink0
$ns connect $udp1 $sink0

proc finish {} {
        global f nf ns
        #Close the output files
        $ns flush-trace
        close $nf
        close $f
        puts "running nam..."
        exec nam out.nam &
      exit 0
}


$ns at 10.0 "$cbr0 start"
$ns at 10.0 "$cbr1 start"
$ns at 50.0 "$cbr0 stop"
$ns at 50.0 "$cbr1 stop"
$ns at 60.0 "finish"

$ns run
```



Observations:

- When bandwidth is 10 Mbps and packet size is 512, there are no dropped packets and the no. of frames in queue at the router is very less
- When packet size is increased to 1024, the no. of packet waiting in queue at the router increases and the packets are split into two (1000 and 24)
- In both the above scenarios, the queuing method used doesn't seem affect observations
- When bandwidth is decreased to 1 Mbps, packet dropping occurs
- When DropTail is used as the queuing method between router and receiver, it is observed that only packets from sender 2 is dropped. The no. of packets being queued also increases
- When SFQ is the queuing method, there is more of an even distribution of packet lost between both sender 1 and 2