NS2 based Networking Experiments

Expt. 4. Study of network simulator(ns2) and simulation.

Expt. 4 (a) Simulate three nodes point-to-point networks with a duplex link between them. Set the queue size and vary the bandwidth and find the number of packets dropped.

```
#====================================
#       Initialization
#====================================
#Create a ns simulator
set ns [new Simulator]

#Open the NS trace file
settracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
setnamfile [open out.nam w]
$ns namtrace-all $namfile

#====================================
#       Nodes Definition
#====================================
#Create 3 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

#====================================
#       Links Definition
#====================================
#Createlinks between nodes
$ns duplex-link $n0 $n1 10.0Mb 1ms DropTail
$ns queue-limit $n0 $n1 10
$ns duplex-link $n1 $n2 10.0Mb 1ms DropTail
$ns queue-limit $n1 $n2 10

#Give node position (for NAM)
#$ns duplex-link-op $n0 $n1 orient right
#$ns duplex-link-op $n1 $n2 orient right

######################
#1.to create drop scenario at first node itself ----- >> change the packet size of application protocol and packet size of Transport
# layere.g packet size of cbr =10000 , packet size of tcp =100
#2. Drop at n1 = set queue size ratio to be 5:2 ,BWXDelay between no and n1 = 10Mb X 0.05ms , between n1 and n2 0.05Mb X 100ms
#3 .to count the number of packets dropped grep -c "^d" out.tr
```

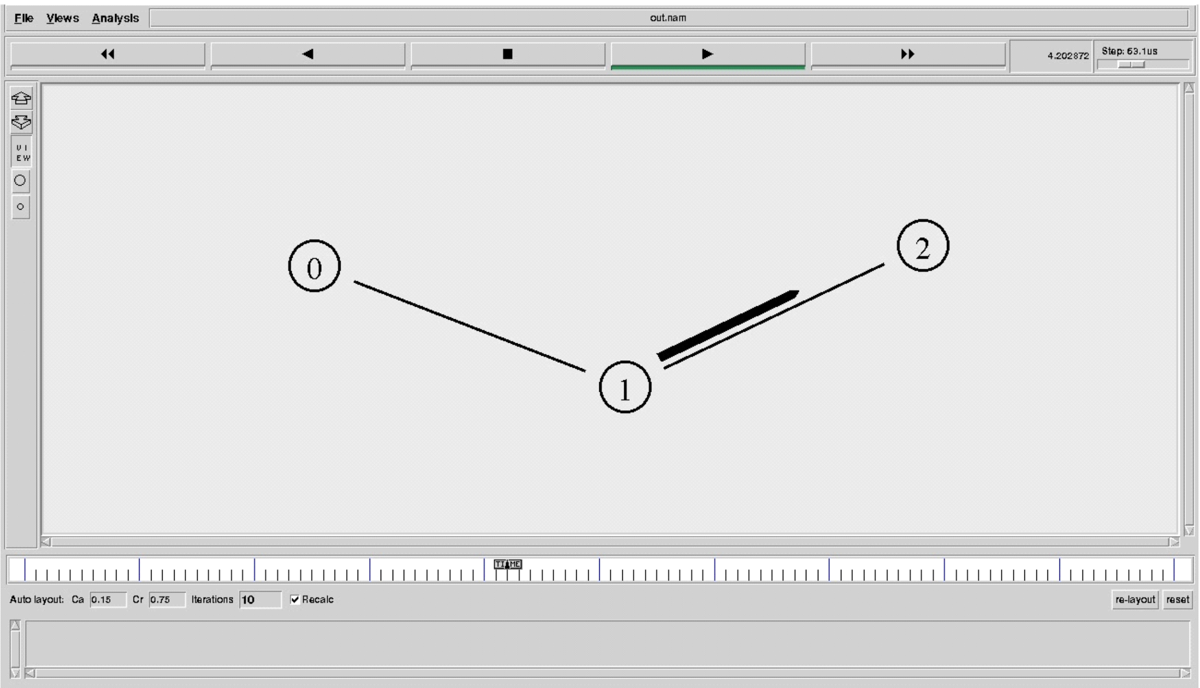
```
#===================================
#       Agents Definition
#===================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n2 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500

#===================================
#       Applications Definition
#===================================
#Setup a CBR Application over TCP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $tcp0
$cbr0 set packetSize_ 1000
$cbr0 set rate_ 0.1Mb
$cbr0 set random_ null
$ns at 1.0 "$cbr0 start"
$ns at 10.0 "$cbr0 stop"

#===================================
#       Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
global ns tracefilenamfile
   $ns flush-trace
close $tracefile
close $namfile
execnamout.nam&
exit 0
}

$ns at 10.0 "finish"

$ns run
```

Expected Results:

```
$cc 4a.tcl
$ ns 4a.tcl

$ grep -c “^d” out.tr
0
```

(After changing parameters)

```
$ cc 4a.tcl
$ ns 4a.tcl
$ grep -c “^d” out.tr
8
```



4.202872

Step: 63.1us

Auto layout: Ca 0.15 Cr 0.75 Iterations 10 ☑ Recalc

Expt. 4 (b). Simulate a transmission of ping message over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.

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

#Open the NS trace file
settracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
setnamfile [open out.nam w]
$ns namtrace-all $namfile

$ns color 1 Red
$ns color 2 Green

#===================================
#       Nodes Definition
#===================================
#Create 6 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]

#===================================
#       Links Definition
#===================================
#Createlinks between nodes
$ns duplex-link $n0 $n1 10.0Mb 0.05ms DropTail
$ns queue-limit $n0 $n1 5
$ns duplex-link $n1 $n2 0.05Mb 100ms DropTail
$ns queue-limit $n1 $n2 2
$ns duplex-link $n2 $n3 10.0Mb 1ms DropTail
$ns queue-limit $n2 $n3 10
$ns duplex-link $n3 $n4 10.0Mb 1ms DropTail
$ns queue-limit $n3 $n4 10
$ns duplex-link $n4 $n5 10.0Mb 1ms DropTail
$ns queue-limit $n4 $n5 10

## to create congestion and to depict the packet drop
# 1. BW X Delay [n0->n1 10MB X 0.05 ms ,Queue Size =5 ] + [ n1->n2 0.05Mb X 100 ms , Queue size =2 ]
# add 4 sends from p0 at 1.0 , similarly add 4 sends from p2 at 1.0 === drop at n1
# repeat the same scenario for p2 , p3 , p4 and p5 to create congestion scenario
```

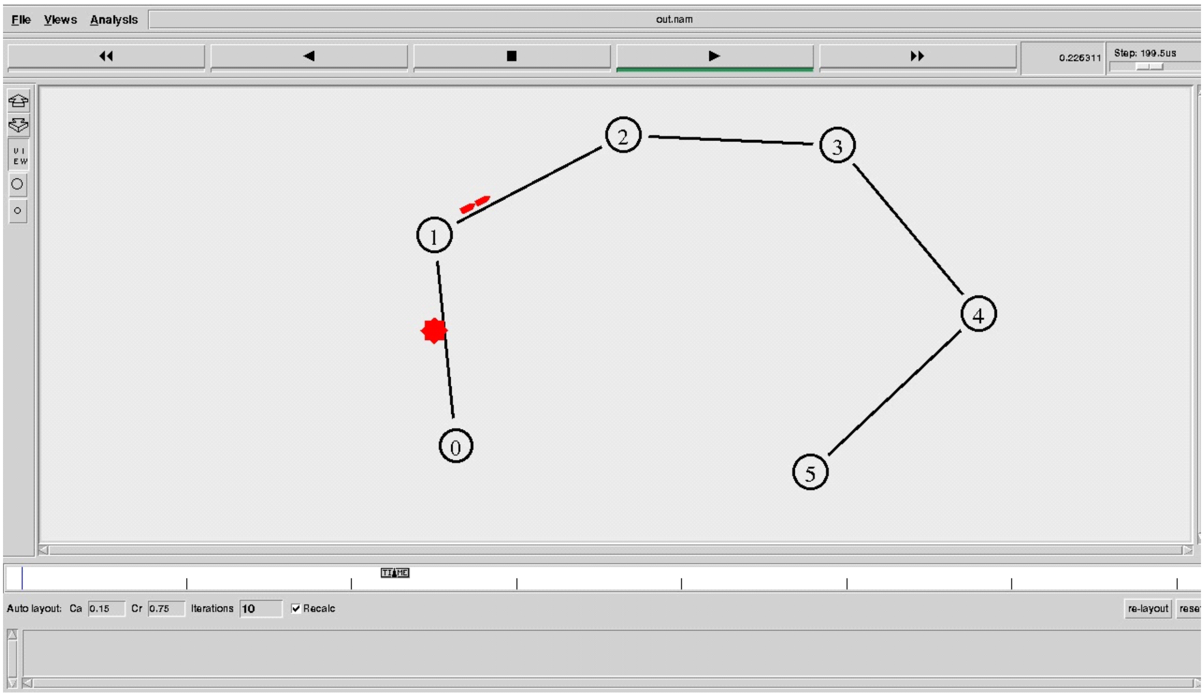
```
#Give node position (for NAM)
#$ns duplex-link-op $n0 $n1 orient right
#$ns duplex-link-op $n1 $n2 orient right
#$ns duplex-link-op $n2 $n3 orient right-down
#$ns duplex-link-op $n3 $n4 orient left
#$ns duplex-link-op $n4 $n5 orient left

Agent/Ping instprocrecv {from rtt} { $self instvar node_ puts "node [$node_ id] received
ping answer from $from with round-trip-time $rttms."
}

#===================================
#       Agents Definition
#===================================
set p0 [new Agent/Ping]
$ns attach-agent $n0 $p0

$p0 set fid_ 1

set p1 [new Agent/Ping]
$ns attach-agent $n5 $p1

$p1 set fid_ 2

#Connect the two agents
$ns connect $p0 $p1

#===================================
#       Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
global ns tracefilenamfile
   $ns flush-trace
close $tracefile
close $namfile
execnamout.nam&
exit 0
}

# to create drop at n1 following sends
$ns at 0.2 "$p0 send"
$ns at 0.2 "$p0 send"
$ns at 0.2 "$p0 send"
$ns at 0.2 "$p0 send"
$ns at 0.4 "$p1 send"
$ns at 0.4 "$p1 send"
$ns at 0.4 "$p1 send"
$ns at 0.4 "$p1 send"
```

```
$ns at 2.0 "finish"

$ns run
```

```
$ cc 4b.tcl
$ ns 4b.tcl
node 0 received ping answer from  #5 with round-trip-time 227.0 ms.
node 0 received ping answer from  #5 with round-trip-time 237.2 ms.
node 5 received ping answer from  #0 with round-trip-time 227.0 ms.
node 5 received ping answer from  #0 with round-trip-time 237.2 ms.
node 5 received ping answer from  #0 with round-trip-time 247.5 ms.
node 5 received ping answer from  #0 with round-trip-time 257.7 ms.
```

(Identifying drop of packets)

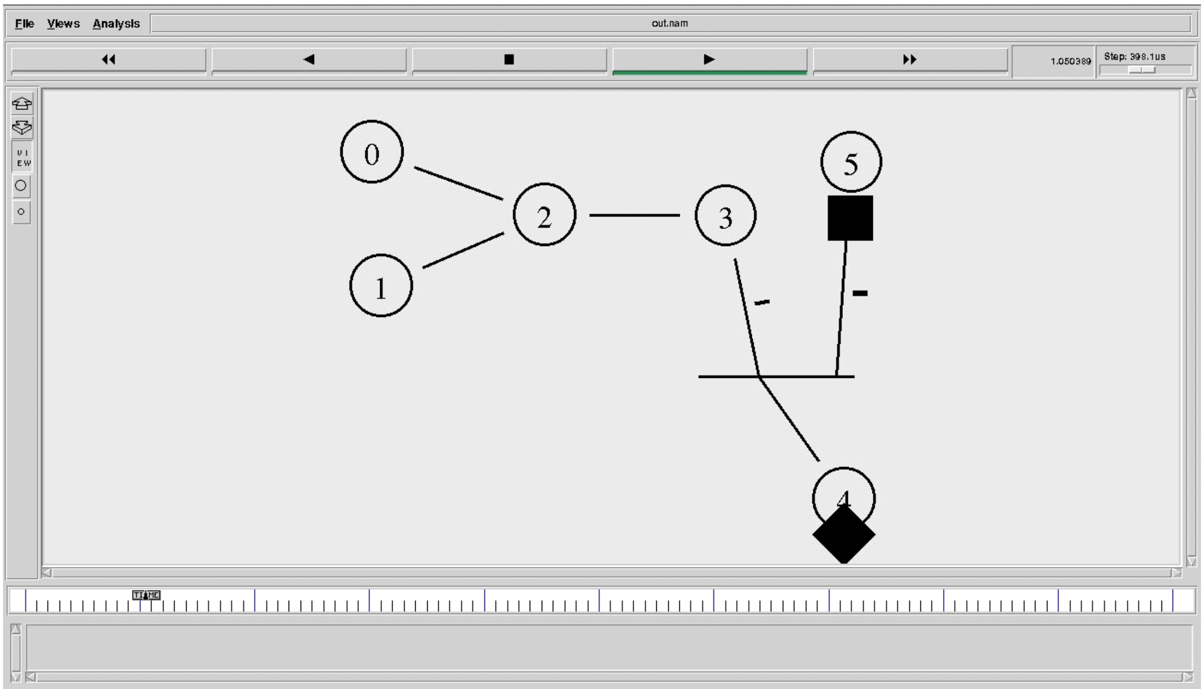
```
$ grep -c “^d” out.tr
2
```



Auto layout: Ca 0.15 Cr 0.75 Iterations 10 ☑ Recalc

Expt. 4 (c) . Simulate an Ethernet LAN with TCL script and simulate using ns2.

```
set ns [new Simulator]

#Open the NS trace file
settracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
setnamfile [open out.nam w]
$ns namtrace-all $namfile

## The code you need to add –Change 1
set winFile0 [open WinFile0 w]
set winFile1 [open WinFile1 w]

#========================================
#       Nodes Definition
#========================================
#Create 6 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]

#========================================
#       Links Definition
#========================================
#Createlinks between nodes
$ns duplex-link $n0 $n2 10.0Mb 1ms DropTail
$ns queue-limit $n0 $n2 10
$ns duplex-link $n1 $n2 10.0Mb 1ms DropTail
$ns queue-limit $n1 $n2 10
$ns simplex-link $n2 $n3 10.0Mb 1ms DropTail
$ns queue-limit $n2 $n3 10
$ns simplex-link $n3 $n2 10.0Mb 1ms DropTail
$ns queue-limit $n3 $n2 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns simplex-link-op $n2 $n3 orient right
$ns simplex-link-op $n3 $n2 orient left

## change 2 –setting up the lan
setlan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/DropTail MAC/802_3 Channel]
#========================================
#       Agents Definition
#===================================

#Setup a TCP/Newreno connection
set tcp0 [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp0
set sink2 [new Agent/TCPSink]
$ns attach-agent $n4 $sink2
$ns connect $tcp0 $sink2
$tcp0 set packetSize_ 1500
$tcp0 set window 5000   # change 3 –set the tcp window size

#Setup a TCP/Newreno connection
set tcp1 [new Agent/TCP/Newreno]
$ns attach-agent $n5 $tcp1
set sink3 [new Agent/TCPSink]
$ns attach-agent $n1 $sink3
$ns connect $tcp1 $sink3
$tcp1 set packetSize_ 1500
$tcp1 set window 500   # change 4 –set the tcp window size

#===================================
#       Applications Definition
#===================================
#Setup a FTP Application over TCP/Newreno connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 10.0 "$ftp0 stop"

#Setup a FTP Application over TCP/Newreno connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 1.0 "$ftp1 start"
$ns at 10.0 "$ftp1 stop"

# change 4 –setting up error model between $n2 $ n3 in random fashion
set var [new ErrorModel]
$varranvar [new RandomVariable/Uniform]
$var drop-target [new Agent/Null]
$ns lossmodel $var $n2 $n3

#===================================
#       Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
global ns tracefilenamfile
   $ns flush-trace
close $tracefile
close $namfile
execnamout.nam&
        execxgraph WinFile0 WinFile1 &   # change 5 executing x-graph
exit 0
}

# change 6 adding plot window function
procPlotWindow {tcpSource file} {
global ns
set time 0.1     # increment =0.1
set now [$ns now]  # it will set now -> current time
setcwnd [$tcpSource set cwnd_] # set the window of tcp to tcp1 & tcp2
puts $file "$now $cwnd"   # file contains 2 values time & Congestion #Window
$ns at [expr $now+$time] "PlotWindow $tcpSource $file"
}

# change 7 schedule it
$ns at 0.1 "PlotWindow $tcp0 $winFile0"
$ns at 0.1 "PlotWindow $tcp1 $winFile1"
$ns at 10.0 "finish"

$ns run
```



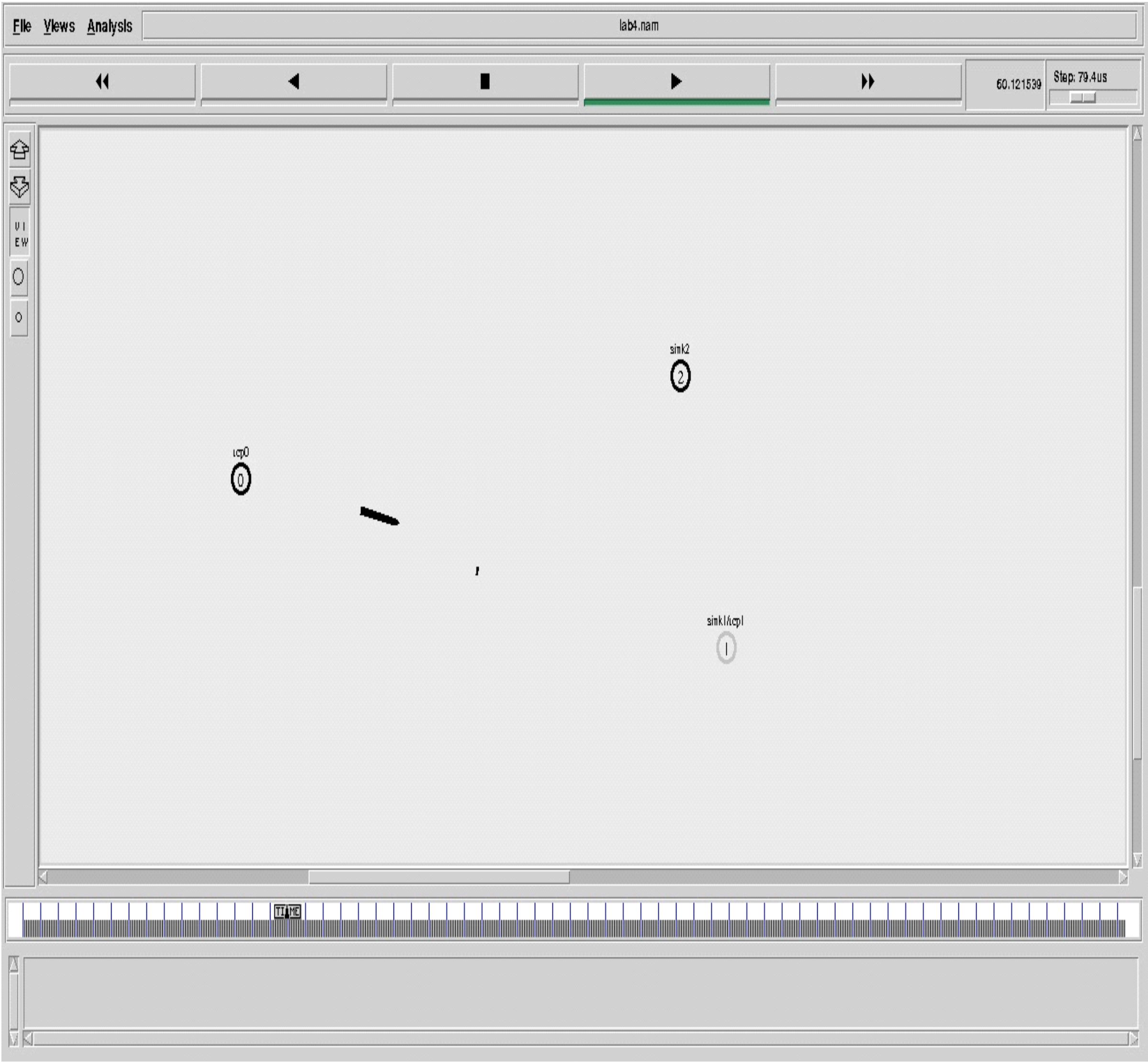
Expt. 4 (d). Simulate simple ESS with transmitting nodes in wireless LAN by simulation and determine the performance w.r.t transmission of packets

```
set ns [new Simulator]

#setup trace support by opening file lab4.tr and call the procedure trace-all
set tf [open lab4.tr w]
$ns trace-all $tf

#create a topology object that keeps track of movements of mobile nodes within the
topological boundary.
set topo [new Topography]
$topo load_flatgrid 1000 1000

set nf [open lab4.nam w]
$ns namtrace-all-wireless $nf 1000 1000

# creating a wireless node you MUST first select (configure) the node configuration
parameters to "become" a wireless node.
$ns node-config -adhocRouting DSDV \
-llType LL \
-macType Mac/802_11 \
-ifqType Queue/DropTail \
-ifqLen 50 \
-phyType Phy/WirelessPhy \
-channelType Channel/WirelessChannel \
-propType Propagation/TwoRayGround \
-antType Antenna/OmniAntenna \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON

# Create god object
create-god 3
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

$n0 label "tcp0"
$n1 label "sink1/tcp1"
$n2 label "sink2"

$n0 set X_ 50
$n0 set Y_ 50
$n0 set Z_ 0
$n1 set X_ 100
$n1 set Y_ 100
$n1 set Z_ 0
$n2 set X_ 600
$n2 set Y_ 600
```

```
$n2 set Z_ 0

$ns at 0.1 "$n0 setdest 50 50 15"
$ns at 0.1 "$n1 setdest 100 100 25"
$ns at 0.1 "$n2 setdest 600 600 25"

set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp0 $sink1
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2
$ns connect $tcp1 $sink2

$ns at 5 "$ftp0 start"
$ns at 5 "$ftp1 start"

$ns at 100 "$n1 setdest 550 550 15"
$ns at 190 "$n1 setdest 70 70 15"

proc finish { } {
global ns nf tf
$ns flush-trace
exec nam lab4.nam &
close $tf
exit 0
}
$ns at 250 "finish"
$ns run
```

awk file:

```
BEGIN{
count1=0
count2=0
pack1=0
pack2=0
time1=0
time2=0
}
{
if($1 == "r" && $3 == "_1_" && $4 == "AGT")
{
count1++
pack1=pack1+$8
time1=$2
}
if($1 == "r" && $3 == "_2_" && $4 =="AGT")
{
count2++
pack2=pack2+$8
time2=$2
}
}
END{
printf("The Throughput from n0 to n1: %f Mbps \n", ((count1*pack1*8)/(time1*1000000)));
printf("The Throughput from n1 to n2: %f Mbps \n", ((count2*pack2*8)/(time2*1000000)));
}
```



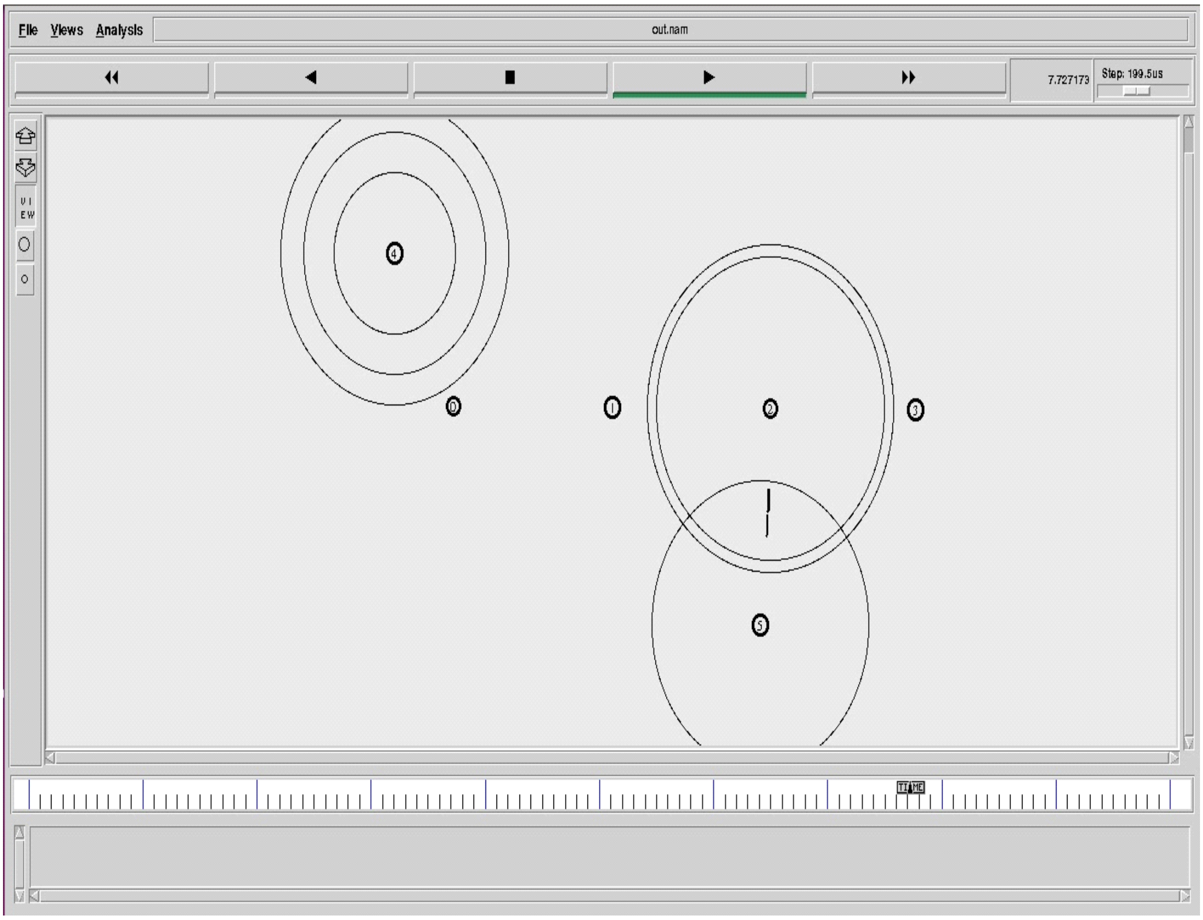
Expt. 4 (e) . Implement and study the performance of GSM on NS2 (Using MAC layer) or equivalent environment.

```
#===================================
# Simulation parameters setup
#===================================
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 6 ;# number of mobilenodes
set val(rp) AODV ;# routing protocol
set val(x) 1052 ;# X dimension of topography
set val(y) 600 ;# Y dimension of topography
set val(stop) 10.0 ;# time of simulation end

#===================================
# Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#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
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
# Mobile node parameter setup
#===================================
$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channel $chan \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace ON \
-movementTrace ON

#====================================
# Nodes Definition
#====================================
#Create 6 nodes
set n0 [$ns node]
$n0 set X_ 303
$n0 set Y_ 302
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 527
$n1 set Y_ 301
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 748
$n2 set Y_ 300
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 952
$n3 set Y_ 299
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 228
$n4 set Y_ 500
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 305
$n5 set Y_ 72
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20

#====================================
# Generate movement
#====================================
$ns at 2 " $n5 setdest 900 72 75 "

#====================================
# Agents Definition
#===================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n4 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n5 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500

#===================================
# Applications Definition
#===================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 10.0 "$ftp0 stop"

#===================================
# Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
global ns tracefile namfile
$ns flush-trace
close $tracefile
close $namfile
exec nam out.nam &
exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
$ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

Expected results:

Awk file:

```
BEGIN{
count1=0
pack1=0
time1=0
}
{
if($1=="r" && $3=="_5_" && $4=="AGT")
{
count1++
pack1=pack1+$8
time1=$2
}
}
END{
printf("The Throughput from n4 to n5: %f Mbps \n", ((count1*pack1*8)/(time1*1000000)));
}
```


File
Views
Analysis
out.nam
7.727173
Step: 199.5us
VIEW
4
0
1
2
3
5
TIME