NETWORK SIMULATOR 2

EX.NO:1 Generating network topology using NS2

AIM:

To write a tcl script to create a simple topology in NS2.

EXPLANATION:

A simple topology of four nodes are created. A duplex link is established between the node n0 and n1, n0 and n2 and n0 and n3.

PROGRAM:

set ns [new Simulator]

**#create file for analysis mode**

set tr [open out.tr w]

$ns trace-all $tr

**#create file for Animation Mode**

set namtr [open out.nam w]

$ns namtrace-all $namtr

**#Create Node**

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

**#Create Link**

$ns duplex-link $n0 $n1 10Mb 5ms DropTail

$ns duplex-link $n2 $n0 10Mb 5ms DropTail

$ns duplex-link $n3 $n0 10mb 5ms DropTail

**#Create Orientation**

$ns duplex-link-op $n0 $n1 orient right

$ns duplex-link-op $n0 $n2 orient left-up

$ns duplex-link-op $n0 $n3 orient left-down

$ns at 10.0 "$ns halt"

$ns run

OUTPUT:

N1

N0

N3

N2

EX.2 CREATE UDP DATA TRAFFIC

AIM:

To create a UDP traffic between two nodes in a given scenario.

EXPLANATION:

A simple topology of four nodes is created. A duplex link is established between the node n0 and n2, n1 and n2 and n2 and n3. UDP CBR traffic is created between the nodes n1 and n3 and another CBR traffic between n2 and n3.

PROGRAM:

set ns [new Simulator]

set nf [open out.nam w]

$ns namtrace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam &

exit 0

}

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

$ns duplex-link $n0 $n2 1Mb 10ms DropTail

$ns duplex-link $n1 $n2 1Mb 10ms DropTail

$ns duplex-link $n3 $n2 1Mb 10ms DropTail

$ns duplex-link-op $n0 $n2 orient right-down

$ns duplex-link-op $n1 $n2 orient right-up

$ns duplex-link-op $n2 $n3 orient right

#Create a UDP agent and attach it to node n0

set udp0 [new Agent/UDP]

$ns attach-agent $n0 $udp0

# Create a CBR traffic source and attach it to udp0

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize\_ 500

$cbr0 set interval\_ 0.005

$cbr0 attach-agent $udp0

#Create a UDP agent and attach it to node n1

set udp1 [new Agent/UDP]

$ns attach-agent $n1 $udp1

# Create a CBR traffic source and attach it to udp1

set cbr1 [new Application/Traffic/CBR]

$cbr1 set packetSize\_ 500

$cbr1 set interval\_ 0.005

$cbr1 attach-agent $udp1

set null0 [new Agent/Null]

$ns attach-agent $n3 $null0

$ns connect $udp0 $null0

$ns connect $udp1 $null0

$ns at 0.5 "$cbr0 start"

$ns at 1.0 "$cbr1 start"

$ns at 4.0 "$cbr1 stop"

$ns at 4.5 "$cbr0 stop"

$udp0 set class\_ 1

$udp1 set class\_ 2

$ns color 1 Blue

$ns color 2 Red

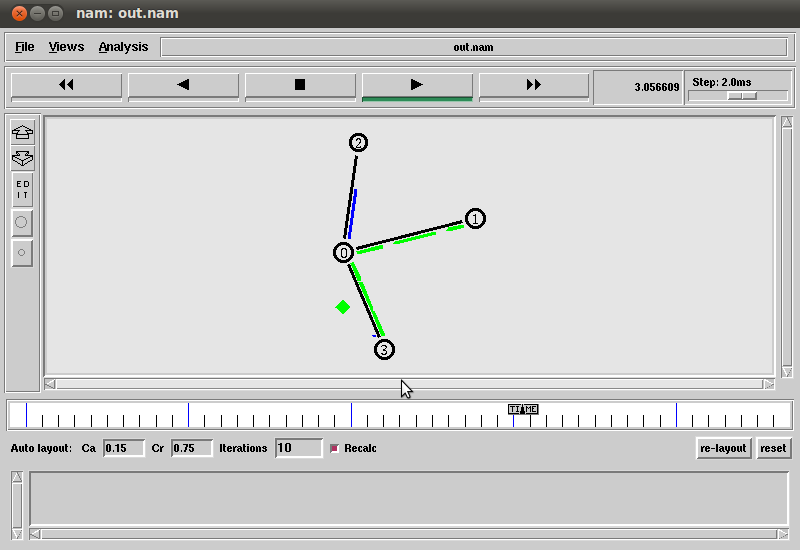
$ns duplex-link-op $n2 $n3 queuePos 0.5

$ns duplex-link $n3 $n2 1Mb 10ms SFQ

$ns at 5.0 "finish"

$ns run

OUTPUT:



EX.3 TCP DATA TRAFFIC

AIM:

To create tcp data traffic between two nodes in a given scenario.

EXPLANATION:

A simple topology of four nodes is created. A duplex link is established between the node n0 and n2, n1 and n2 and n2 and n3. TCP FTP traffic is created between the nodes n1 and n3 and another TCP traffic between n2 and n3.

PROGRAM:

set ns [new Simulator]

set nf [open out.nam w]

$ns namtrace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam &

exit 0

}

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

$ns duplex-link $n0 $n1 1Mb 10ms DropTail

$ns duplex-link $n0 $n2 1Mb 10ms DropTail

$ns duplex-link $n0 $n3 1Mb 10ms DropTail

#create a tcp connection and attach to node n1

set tcp1 [new Agent/TCP]

$ns attach-agent $n1 $tcp1

$tcp1 set window\_ 8

$tcp1 set fid\_ 1

#create a tcp connection and attach to node n2

set tcp2 [new Agent/TCP]

$ns attach-agent $n2 $tcp2

$tcp2 set window\_ 8

$tcp2 set fid\_ 2

#create the sink nodes 1 and 2

set sink1 [new Agent/TCPSink]

set sink2 [new Agent/TCPSink]

#attach sink 1 and 2 to node n3

$ns attach-agent $n3 $sink1

$ns attach-agent $n3 $sink2

$ns connect $tcp1 $sink1

$ns connect $tcp2 $sink2

#Create FTP applications and attach them to agents

set ftp1 [new Application/FTP]

$ftp1 attach-agent $tcp1

set ftp2 [new Application/FTP]

$ftp2 attach-agent $tcp2

$ns at 0.1 "$ftp1 start"

$ns at 0.5 "$ftp2 start"

$ns at 3.5 "$ftp1 stop"

$ns at 5.0 "$ftp2 stop"

$ftp1 set class\_ 1

$ftp2 set class\_ 2

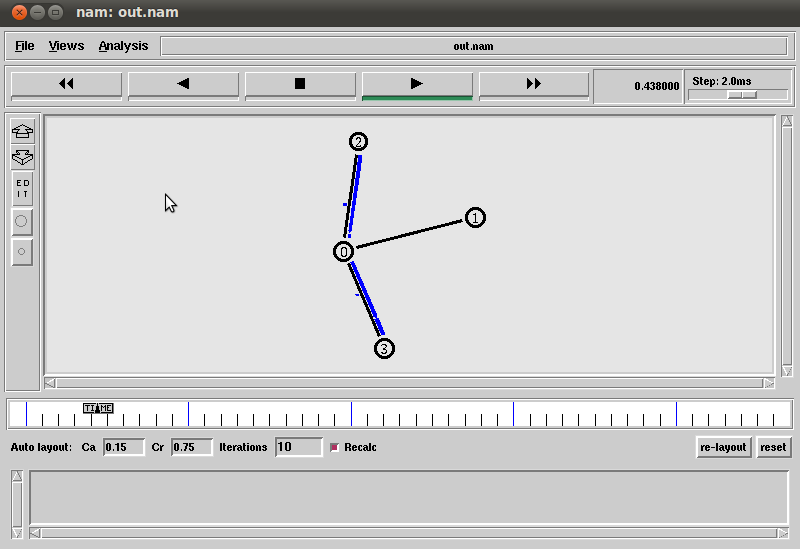
$ns color 1 Red

$ns color 2 Blue

$ns at 5.0 "finish"

$ns run

OUTPUT:



EX.3 WIRELESS SCENARIO

AIM:

To create a simple wireless scenario in NS2

EXPLANATION:

A simple topology of three nodes is created. UDP CBR traffic is created between the nodes n1 and n0 and another CBR traffic between n2 and n1.

PROGRAM:

# Define options

# ======================================================================

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(x) 1000 ;# X dimension of topology

set val(y) 1000 ;# Y dimension of topology

set val(cp) "" ;# node movement model file

set val(sc) "" ;# traffic model file

set val(ifqlen) 50 ;# max packet in ifq

set val(nn) 3 ;# number of mobilenodes

set val(seed) 0.0

set val(stop) 1000.0 ;# simulation time

set val(tr) hidden-out.tr ;# trace file name

set val(rp) DSDV ;# routing protocol

set AgentTrace ON

set RouterTrace ON

set MacTrace OFF

# Open trace file

$ns\_ use-newtrace ;# Use new trace format

set namfd [open hidden-out.nam w]

$ns\_ namtrace-all-wireless $namfd $val(x) $val(y)

set tracefd [open $val(tr) w]

$ns\_ trace-all $tracefd

# set up topography object

set topo [new Topography]

$topo load\_flatgrid $val(x) $val(y)

# create channel

set chan [new $val(chan)]

# Create God

set god\_ [create-god $val(nn)]

# Create the specified number of mobile nodes [$val(nn)] and "attach" them

# to the channel. Three nodes are created : node(0), node(1) and node(2)

$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 OFF \

-movementTrace OFF

for {set i 0} {$i < $val(nn) } {incr i} {

set node\_($i) [$ns\_ node]

$node\_($i) random-motion 0 ;# disable random motion

}

# Provide initial (X,Y, for now Z=0) co-ordinates for mobilenodes

#

$node\_(0) set X\_ 250.0

$node\_(0) set Y\_ 500.0

$node\_(0) set Z\_ 0.0

$node\_(1) set X\_ 500.0

$node\_(1) set Y\_ 500.0

$node\_(1) set Z\_ 0.0

$node\_(2) set X\_ 750.0

$node\_(2) set Y\_ 500.0

$node\_(2) set Z\_ 0.0

# Load the god object with shortest hop information

#

$god\_ set-dist 1 2 1

$god\_ set-dist 0 2 2

$god\_ set-dist 0 1 1

# Now produce some simple node movements

# Node\_(1) starts to move upward and then downward

set god\_ [God instance]

# Setup traffic flow between nodes 0 connecting to 1 at time 100.0

set udp\_(0) [new Agent/UDP]

$udp\_(0) set fid\_ 1

$ns\_ attach-agent $node\_(0) $udp\_(0)

set udp\_(1) [new Agent/UDP]

$udp\_(1) set fid\_ 2

$ns\_ attach-agent $node\_(2) $udp\_(1)

set null\_(0) [new Agent/Null]

$ns\_ attach-agent $node\_(1) $null\_(0)

set cbr\_(0) [new Application/Traffic/CBR]

$cbr\_(0) set packetSize\_ 500

$cbr\_(0) set set interval\_ 0.005

$cbr\_(0) set maxpkts\_ 10000

$cbr\_(0) attach-agent $udp\_(0)

set cbr\_(1) [new Application/Traffic/CBR]

$cbr\_(1) set packetSize\_ 500

$cbr\_(1) set set interval\_ 0.005

$cbr\_(1) set maxpkts\_ 10000

$cbr\_(1) attach-agent $udp\_(1)

$ns\_ connect $udp\_(0) $null\_(0)

$ns\_ at 100.0 "$cbr\_(0) start"

$ns\_ connect $udp\_(1) $null\_(0)

$ns\_ at 100.0 "$cbr\_(1) start"

#Define node initial position in nam, only fro nam

for {set i 0} {$i < $val(nn)} {incr i} {

# The function must be called after mobility model is defined.

$ns\_ initial\_node\_pos $node\_($i) 60

}

# Tell nodes when the simulation ends

for {set i 0} {$i < $val(nn) } {incr i} {

$ns\_ at $val(stop) "$node\_($i) reset";

}

$ns\_ at $val(stop) "stop"

$ns\_ at $val(stop) "puts \"NS EXITING...\" ; $ns\_ halt"

proc stop {} {

global ns\_ tracefd namfd

$ns\_ flush-trace

close $tracefd

close $namfd

exec nam hidden-out.nam &

exit 0

}

puts $tracefd "M 0.0 nn $val(nn) x $val(x) y $val(y) rp $val(rp)"

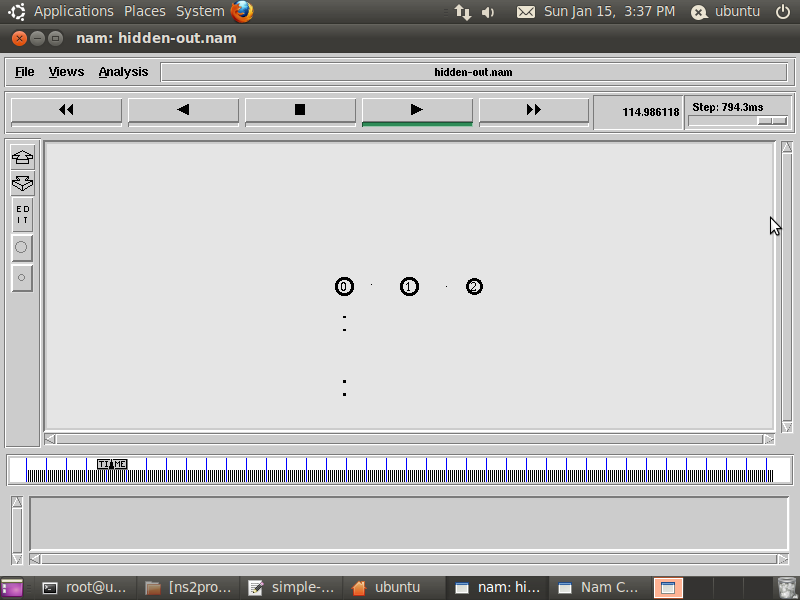
puts $tracefd "M 0.0 sc $val(sc) cp $val(cp) seed $val(seed)"

puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"

puts "Starting Simulation..."

$ns\_ run

OUTPUT:



EX. 4 TOPOLOGY AND MOVEMENT GENERATION

AIM:

To write a tcl script to generate movement of nodes of a given wireless topology.

EXPLANATION:

A simple topology of two nodes is created. UDP CBR traffic is created between the nodes n1 and n0. After initial topology move the two nodes towards each other, when the nodes is within the range data transfer is done, when they move away from each other the data packets are dropped.

PROGRAM:

**#Static (Fixed) Topology**

$node\_(0) set X\_ 50.0

$node\_(0) set Y\_ 50.0

$node\_(0) set Z\_ 0.0

$node\_(1) set X\_ 150.0

$node\_(1) set Y\_ 50.0

$node\_(1) set Z\_ 0.0

$node\_(2) set X\_ 300.0

$node\_(2) set Y\_ 50.0

$node\_(2) set Z\_ 0.0

**#Node Movement**

$ns\_ at 3.0 "$node\_(2) setdest 450 100 50"

$ns\_ at 3.0 "$node\_(1) setdest 250 100 50"

**#Grid Topology**

set val(rlen) 3

set val(nn) [expr $val(rlen) \* $val(rlen)]

set gridspace [expr $val(x) / $val(rlen)]

for {set i 0} {$i < $val(rlen) } {incr i} {

for {set j 0} {$j < $val(rlen) } {incr j} {

set a [expr $j + [expr $i \* $val(rlen)]]

$node\_($a) set X\_ [expr 0.0 + [ expr $i \* $gridspace]]

$node\_($a) set Y\_ [expr 0.0 + [ expr $j \* $gridspace]]

$node\_($i) set Z\_ 0.0

}

}

**#Random Topology**

#To apply random movement we need **setdest** command

#Syntax

setdest –n Nunmber of node –p pause time -M Max\_Speed -t total\_simulation\_time –x x-axisvalue

–y y-axis-value

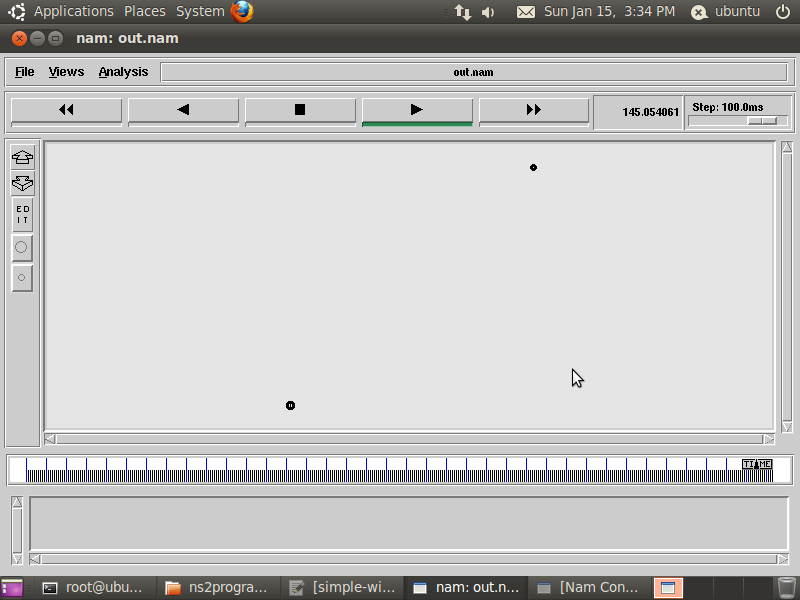
#Example

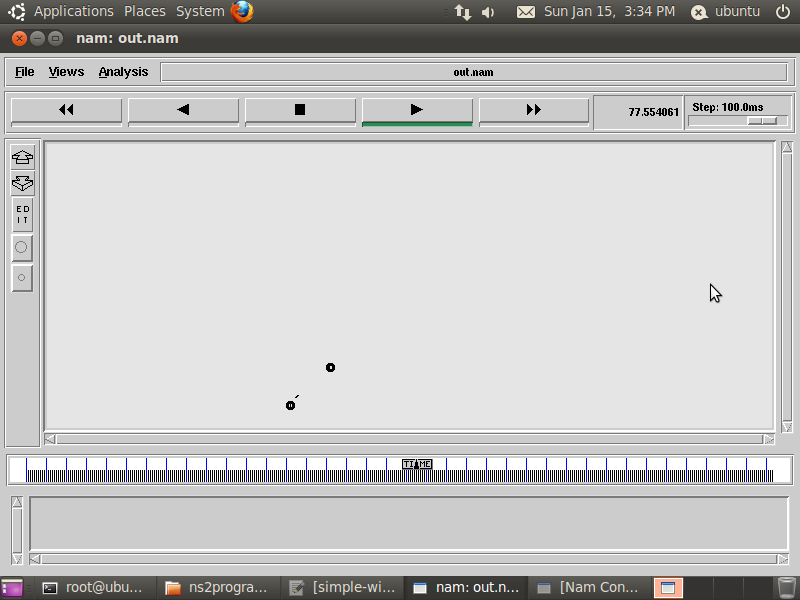
setdest -n 20 -p 10 -M 20 -t 100 -x 500 -y 500 >scen-20-20

setdest -n 30 -p 10 -M 20 -t 100 -x 500 -y 500 >scen-30-20

setdest -n 40 -p 10 -M 20 -t 100 -x 500 -y 500 >scen-40-20

OUTPUT:





EX.5 ENERGY MODEL

AIM:

To create a energy model for a network topology using NS2.

PROGRAM:

$ns\_ node-config -adhocRouting $val(routing) \

-llType LL \

-macType Mac/802\_11 \

-ifqType Queue/DropTail/PriQueue \

-ifqLen 100 \

-antType Antenna/OmniAntenna \

-propType Propagation/TwoRayGround \

-phyType Phy/WirelessPhy \

-channelType Channel/WirelessChannel \

**-energyModel EnergyModel \**

**-rxPower 0.3 \**

**-txPower 0.6 \**

**-initialEnergy 10 \**

EX. 6 WIRED CUM WIRELESS

AIM:

To create a wired cum wireless scenario using ns2 simulator.

EXPLANATION:

A wired cum wireless scenario consists of both wired and wireless topology interlinked and data can be sent from a wired node to a wireless node and vice- versa.

PROGRAM:

set val(x) 800

set val(y) 800

set val(ifqlen) 50

set val(tr) wireless.tr

set val(namtr) wireless.nam

set val(nn) 3

set val(adhocRouting) DSDV

set val(stop) 250

set num\_wired\_nodes 2

set num\_bs\_nodes 2

set ns\_ [new Simulator]

**# set up for hierarchical routing**

$ns\_ node-config -addressType hierarchical

AddrParams set domain\_num\_ **3**

lappend cluster\_num **2 1 1**

AddrParams set cluster\_num\_ $cluster\_num

lappend no\_of\_nodes **1 1 4 1**

AddrParams set nodes\_num\_ $no\_of\_nodes

set tracefd [open $val(tr) w]

$ns\_ trace-all $tracefd

set namtracefd [open $val(namtr) w]

$ns\_ namtrace-all-wireless $namtracefd $val(x) $val(y)

set topo [new Topography]

$topo load\_flatgrid $val(x) $val(y)

MIT - Chennai SAT Infosys - CBE 9944219934

7

# god needs to know the number of all wireless interfaces

create-god [expr $val(nn) + $num\_bs\_nodes]

set W(0) [$ns\_ node **0.0.0**]

$W(0) label [$W(0) node-addr]

set W(1) [$ns\_ node **0.1.0**]

$W(1) label [$W(1) node-addr]

$ns\_ node-config -adhocRouting $val(adhocRouting) \

-llType LL \

-macType Mac/802\_11 \

-ifqType Queue/DropTail/PriQueue \

-ifqLen 50 \

-antType Antenna/OmniAntenna \

-propType Propagation/TwoRayGround \

-phyType Phy/WirelessPhy \

-channelType Channel/WirelessChannel \

-topoInstance $topo \

-wiredRouting ON \

-agentTrace ON \

-routerTrace ON \

-macTrace OFF

**#create Base Station**

set BS(0) [$ns\_ node **1.0.0**]

set BS(1) [$ns\_ node **2.0.0**]

$BS(0) random-motion 0

$BS(1) random-motion 0

**#create links between wired and BS nodes**

$ns\_ duplex-link $W(0) $W(1) 5Mb 40ms DropTail

$ns\_ duplex-link $W(1) $BS(0) 5Mb 100ms DropTail

$ns\_ duplex-link $W(1) $BS(1) 5Mb 50ms DropTail

$ns\_ duplex-link-op $W(0) $W(1) orient right

$ns\_ duplex-link-op $W(1) $BS(0) orient right-down

$ns\_ duplex-link-op $W(1) $BS(1) orient right-up

#configure for mobilenodes

$ns\_ node-config -wiredRouting OFF

**#create Mobile Node**

set node\_(0) [$ns\_ node **1.0.1**]

**#connect Mobile Node with Base Station**

$node\_(0) base-station [AddrParams addr2id **1.0.0**]

**#Display Node Address for NAM**

$node\_(0) label [$node\_(0) node-addr]

set node\_(1) [$ns\_ node **1.0.2**]

$node\_(1) base-station [AddrParams addr2id 1.0.0]

$node\_(1) label [$node\_(1) node-addr]

set node\_(2) [$ns\_ node 1.0.3]

$node\_(2) base-station [AddrParams addr2id 1.0.0]

$node\_(2) label [$node\_(2) node-addr]

MIT - Chennai SAT Infosys - CBE 9944219934

8

$BS(0) set X\_ 350.0

$BS(0) set Y\_ 600.0

$BS(0) set Z\_ 0.0

$BS(0) label [$BS(0) node-addr]

$BS(1) set X\_ 350.0

$BS(1) set Y\_ 300.0

$BS(1) set Z\_ 0.0

$BS(1) label [$BS(1) node-addr]

$node\_(0) set X\_ 350

$node\_(0) set Y\_ 600

$node\_(0) set Z\_ 0

$node\_(1) set X\_ 580

$node\_(1) set Y\_ 600

$node\_(1) set Z\_ 0

$node\_(2) set X\_ 620

$node\_(2) set Y\_ 600

$node\_(2) set Z\_ 0

for {set i 0} {$i < $val(nn)} {incr i} {

$ns\_ initial\_node\_pos $node\_($i) 20

}

$ns\_ at 5.0 "$node\_(0) setdest 300 300 50"

$ns\_ at 5.0 "$node\_(1) setdest 300 400 50"

$ns\_ at 5.0 "$node\_(2) setdest 300 500 50"

**#create UDP Source**

set udp0 [new Agent/UDP]

$ns\_ attach-agent $W(0) $udp0

**#create UDP Destination**

set null0 [new Agent/Null]

$ns\_ attach-agent $node\_(2) $null0

**#connecting UDP Source & Destination**

$ns\_ connect $udp0 $null0

**#create application traffic**

set cbr0 [new Application/Traffic/CBR]

$cbr0 attach-agent $udp0

**#Application start time**

$ns\_ at 1.0 "$cbr0 start"

#Application Stop time

$ns\_ at 20.0 "$cbr0 stop"

for {set i } {$i < $val(nn) } {incr i} {

$ns\_ at $val(stop).0000010 "$node\_($i) reset";

}

$ns\_ at $val(stop).1 " $ns\_ halt"

$ns\_ run

EX. 8 ANALYSE THE PERFORMANCE OF A NETWORK USING XGRAPH

AIM:

To analyse the performance of the given network topology using xgraph.

THEORY:

xgraph is a plotting program which can be used to create graphic representations of simulation results. You can create output files in your Tcl scripts, which can be used as data sets for xgraph. Call xgraph to display the results with the command “**xgraph** <**data-file**>”. Below you can see a screenshot of an xgraph window

PROGRAM:

#

# Per-packet tracing from node n0 to node n1

#

# create file for monitoring all packets from n0 to n1

set trace\_file [open trace\_file.out w]

$ns at 0.0 "$ns trace-queue $n0 $n1 $trace\_file"

# SimulationTime

set SimTime 10.0

set old\_data 0

#Call the procedure “finish” at the end of simulation

$ns at $SimTime "finish"

proc finish {} {

global ns old\_data

$ns flush-trace

# Get data from trace file, put current-time and calculated-throughput in a new file

exec awk {

{

if (($1=="-" && $5=="tcp") || ($1=="-" && $5=="cbr")) {

old\_data=old\_data + $6

print $2, old\_data\*8.0/$2/1000000

}

}

} trace\_file.out > throughput.data

# Call xgraph plotting program to plot throughput vs time

exec xgraph throughput.data &

exit 0

}

OUTPUT:

