

1. Implement a point-to-point network with four nodes and duplex links between them. Analyse the network performance by setting the queue size and varying the bandwidth.

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#####
#      Simulation parameters setup
#####
set val(stop)    100.0          ;# time of simulation end

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

#Open the NS trace file
set tracefile [open expt1.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open expt1.nam w]
$ns namtrace-all $namfile

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

#####
#      Links Definition
#####
#Createlinks between nodes
$ns duplex-link $n0 $n2 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n2 50
$ns duplex-link $n1 $n2 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n2 50
$ns duplex-link $n2 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 50
#Give node position (for NAM)
$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

#####
```

```
#           Agents Definition
#=====
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink2 [new Agent/TCPSink]
$ns attach-agent $n3 $sink2
$ns connect $tcp0 $sink2
$tcp0 set packetSize_ 1500

#Setup a TCP connection
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3
$ns connect $tcp1 $sink3
$tcp1 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"
$ns at 50.0 "$ftp0 stop"

#Setup a FTP Application over TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 51.0 "$ftp1 start"
$ns at 100.0 "$ftp1 stop"

#=====
#           Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam expt1.nam &
    exit 0
}
$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
```

AWK SCRIPT

```
BEGIN{
a=0
b=0
c=0
}
{
  if($1=="r"&& $3=="2"&& $4=="3"&& $5=="tcp" && $6=="1540")
  {
    a=a+1;
  }
  if($1=="r" && $3=="0" && $4=="2" && $5=="tcp" && $6=="1540")
  {
    b=b+1;
  }
  if($1=="r" && $3=="1" && $4=="2" && $5=="tcp" && $6=="1540")
  {
    c=c+1;
  }
}
END {
{
printf("\n total no of data packets received at node 3:%d",a);
printf("\n total no of data packets sent  at node 2 by node 0:%d",b);
printf("\n total no of data packets sent at node 2 by node 1:%d\n",c);
}
}
```

2. Implement a four-node point to point network with links n0-n2, n1-n2 and n2-n3. Apply TCP agent between n0-n3 and UDP between n1-n3. Apply relevant applications over TCP and UDP agents changing the parameter and determine the number of packets sent by TCP/UDP.

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#####
#      Simulation parameters setup
#####
set val(stop)    100.0                ;# time of simulation end

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

#Open the NS trace file
set tracefile [open expt2.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open expt2.nam w]
$ns namtrace-all $namfile

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

#####
#      Links Definition
#####
#Createlinks between nodes
$ns duplex-link $n0 $n2 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n2 50
$ns duplex-link $n1 $n2 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n2 50
$ns duplex-link $n2 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 50

#Give node position (for NAM)
$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

#####
#      Agents Definition
```

```
#=====
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink2 [new Agent/TCPSink]
$ns attach-agent $n3 $sink2
$ns connect $tcp0 $sink2
$tcp0 set packetSize_ 1500

#Setup a UDP connection
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set null3 [new Agent/Null]
$ns attach-agent $n3 $null3
$ns connect $udp1 $null3
$udp1 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"
$ns at 50.0 "$ftp0 stop"

#Setup a CBR Application over UDP connection
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$cbr1 set packetSize_ 1000
$cbr1 set rate_ 1.0Mb
$cbr1 set random_ null
$ns at 51.0 "$cbr1 start"
$ns at 100.0 "$cbr1 stop"

#=====
#           Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam expt2.nam &
    exit 0
}
$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
```

AWK SCRIPT

```
BEGIN{
a=0
b=0
}
{
  if ($1=="r" && $3=="0" && $4=="2" && $5=="tcp" && $6=="1540")
  {
    a=a+1;
  }

  if ($1=="r" && $3=="1" && $4=="2" && $5=="cbr" && $6=="1000")
  {
    b=b+1;
  }
}
END{
{
printf("\n total no of data packets sent at node 2 by node 0:%d",a);
printf("\n total no of data packets sent at node 2 by node 1:%d\n",b);
}
}
```

3. Implement Ethernet LAN using n (6-24) nodes. Compare the throughput by changing the error rate and data rate.

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>
#=====
#      Simulation parameters setup
#=====
set val(stop)    10.0      ;# time of simulation end

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

#Open the NS trace file
set tracefile [open expt3.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open expt3.nam w]
$ns namtrace-all $namfile

#=====
#      Nodes Definition
#=====
#Create 8 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 [$ns node]
set n7 [$ns node]
```

```
set lan [$ns newLan "$n0 $n1 $n2 $n3" 0.5Mb 40ms LL
Queue/DropTail MAC/Csma/Cd Channel]
set lan [$ns newLan "$n7 $n6 $n5 $n4" 0.5Mb 40ms LL
Queue/DropTail MAC/Csma/Cd Channel]
#=====
#           Links Definition
#=====
#Createlinks between nodes
$ns duplex-link $n4 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n4 $n3 50
#Give node position (for NAM)
$ns duplex-link-op $n4 $n3 orient left-up
#=====
#           Agents Definition
#=====
#Setup a UDP connection
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set null2 [new Agent/Null]
$ns attach-agent $n5 $null2
$ns connect $udp1 $null2
$udp1 set packetSize_ 1500
#=====
#           Applications Definition
#=====
#Setup a CBR Application over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp1
$cbr0 set packetSize_ 1000
$cbr0 set rate_ 10.0Mb
$cbr0 set random_ null
$ns at 1.0 "$cbr0 start"
$ns at 99.0 "$cbr0 stop"
set em_ [new ErrorModel]
```

```
$ns lossmodel $em_ $n3 $n4
$em_ set rate_ 0.1

#=====
#           Termination
#=====

#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam expt3.nam &
    exit 0
}

$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
```

AWK SCRIPT

```
BEGIN{
a=0;
}
{
if($1=="r" && $4=="5" && $5=="cbr" && $6=="1000")
{
    a++;
}
}
END{
{
printf("\n Total No of Data packets at node 5:%d\n",a);
}
}
```