```
# Create Simulator set
ns [new Simulator]
# Open Trace file and NAM file set
ntrace [open prog1.tr w]
$ns trace-all $ntrace set
namfile [open prog1.nam w]
$ns namtrace-all
$namfile # Finish
Procedure proc Finish {} {
   global ns ntrace namfile
   # Dump all the trace data and close the files
   $ns flush-trace
close $ntrace    close
$namfile
   # Execute the nam animation file
   exec nam prog1.nam &
   exec echo "The number of packets dropped
are:" &
   exec  grep  -c  "^d"  prog1.tr  &
exit 0
}
# Create 3 nodes set
n0 [$ns node] set
n1 [$ns node] set
n2 [$ns node]
# Create Links between nodes
# Modify the bandwidth to observe variation in
packet drop
$ns duplex-link $n0 $n1 0.2Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
# Set Queue Size
# Modify the queue length to observe variation in
packet drop
$ns queue-limit $n0 $n1 10
$ns queue-limit $n1 $n2 10
# Set up a Transport layer connection
set udp [new Agent/UDP]
$ns attach-agent $n0 $udp
set null [new Agent/Null]
$ns attach-agent $n2 $null
$ns connect $udp $null
# Set up an Application layer Traffic set
cbr0 [new Application/Traffic/CBR]
# $cbr0 set type_ CBR
# $cbr0 set packetSize_ 100
# $cbr0 set rate_ 1Mb
# $cbr0 set random_ false
$cbr0 attach-agent $udp
# Schedule Events
$ns at 0.0 "$cbr0 start"
$ns at 5.0 "Finish"
# Run the Simulation
$ns run
```

```
# Create Simulator set
ns [new Simulator]
# Open Trace file and NAM file set
ntrace [open prog2.tr w]
$ns trace-all $ntrace set
namfile [open prog2.nam w]
$ns namtrace-all
$namfile # Finish
Procedure proc Finish {} {
   global ns ntrace namfile
   $ns flush-trace    close
$ntrace    close $namfile
exec nam prog2.nam &
   exec echo "The number of TCP packets sent
are" &
   exec grep "^+" prog2.tr | cut -d " " -f 5 | grep
-c "tcp" &
   exec echo "The number of UDP packets sent
are" &
   exec grep "^+" prog2.tr | cut -d " " -f 5 | grep
 -c "cbr" &
    exit 0
}
# Create nodes set
n0 [$ns node] set
n1 [$ns node] set
n2 [$ns node] set
n3 [$ns node]
# Create Links between nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 2Mb 20ms DropTail #
Extra code for node orientation and labels
$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
$n0 label "TCP Source"
$n1 label "UDP Source"
$n3 label "Destination"
$n0 color blue
$n1 color orange
$n3 color red
#Set up TCP agents and
connections set tcp0 [new
Agent/TCP] $ns attach-agent $n0
$tcp0 set sink0 [new
Agent/TCPSink]
$ns attach-agent $n3 $sink0
$ns connect $tcp0 $sink0
# Set up UDP agents and
connections set udp0 [new
Agent/UDP] $ns attach-agent $n1
$udp0 set null0 [new Agent/Null]
$ns attach-agent $n3 $null0
$ns connect $udp0 $null0
# Set up Application layer Traffic set
ftp0 [new Application/FTP]
$ftp0 set type_ FTP
$ftp0 attach-agent $tcp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set type_ CBR
$cbr0 set packetSize_ 1000
$cbr0 set rate_ 0.01Mb
$cbr0 set random_ false
$cbr0 attach-agent $udp0
# Set colors for agents
$ns color 1 magenta
$ns color 2 green
# Set classes for TCP and UDP
$tcp0 set class_ 1
$udp0 set class_ 2
# Schedule events
$ns at 0.1 "$cbr0 start"
$ns at 1.5 "$ftp0 start"
$ns at 1.0 "$cbr0 stop"
$ns at 2.5 "$ftp0 stop"
$ns at 5.0 "Finish"
# Run the simulation
```

```
# Create Simulator set
ns [new Simulator]
# Open Trace file and NAM file
set trf [open p3.tr w]
$ns trace-all $trf set naf
[open p3.nam w]
$ns namtrace-all $naf #
Create Nodes set n0
[$ns node] $n0 color
"red" $n0 label "Source
1" set n1 [$ns node] $n1
color "blue" $n1 label
"Source 2" set n2 [$ns
node] $n2 color
"magenta" $n2 label
"Destination 1" set n3
[$ns node] $n3 color
"green"
$n3 label "Destination 2" #
Create LAN with nodes
set lan [$ns newLan "$n0 $n1 $n2 $n3" 5Mb 10ms
LL Queue/DropTail
Mac/802_3] # Set up TCP
connection set tcp [new
Agent/TCP] $ns attach-agent
$n0 $tcp set ftp [new
Application/FTP] $ftp attach-
agent $tcp set sink [new
Agent/TCPSink]
$ns attach-agent $n2 $sink
$ns connect $tcp $sink #
Set up UDP connection set
udp [new Agent/UDP] $ns
attach-agent $n1 $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp set
null [new Agent/Null] $ns
attach-agent $n3 $null
$ns connect $udp
$null # Finish
Procedure proc finish
{} {    global ns naf trf
$ns flush-trace
   exec nam p3.nam &
   close $trf
close $naf
   exec echo "The number of packet drops due to
collision are" &
   exec  grep  -c  "^d"  p3.tr  &
exit 0
}
# Schedule events
$ns at 0.1 "$cbr start"
$ns at 2.0 "$ftp start"
$ns at 1.9 "$cbr stop"
$ns at 4.3 "$ftp stop"
$ns at 6.0 "finish"
# Run the simulation
$ns run
```

```
# Create Simulator set
ns [new Simulator]
# Open Trace file and NAM file set
trf [open prog5.tr w]
$ns trace-all $trf
set naf [open prog5.nam w]
$ns namtrace-all $naf #
Create nodes set n0
[$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 [$ns node] #
Create LAN with nodes
set lan [$ns newLan "$n0 $n1 $n2 $n3 $n4 $n5
$n6 $n7" 5Mb 10ms LL Queue/DropTail
Channel]
# Set up TCP connection set
tcp [new Agent/TCP] $ns
attach-agent $n0 $tcp set ftp
[new Application/FTP] $ftp
attach-agent $tcp set sink
[new Agent/TCPSink]
$ns attach-agent $n7 $sink
$ns connect $tcp $sink #
Set up UDP connection set
udp [new Agent/UDP] $ns
attach-agent $n1 $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp set
null [new Agent/Null] $ns
attach-agent $n5 $null
$ns connect $udp $null #
Finish Procedure proc
finish {} {    global ns naf
trf    $ns flush-trace
exec nam prog5.nam &
   close $trf
close $naf
   set tcpsize [ exec grep "^r" prog5.tr | grep
"tcp" | tail -n 1 | cut -d " " -f 6]
   set numtcp [ exec grep "^r" prog5.tr | grep -c
"tcp"]
   set tcptime 2.3
   set udpsize [ exec grep "^r" prog5.tr | grep
"cbr" | tail -n 1 | cut -d " " -f 6]
   set numudp [ exec grep "^r" prog5.tr | grep c
"cbr"]
   set udptime 4.0
   puts "The throughput of FTP is"
   puts  "[  expr  ($numtcp*$tcpsize)/$tcptime]
bytes per second"
   puts "The throughput of CBR is"
   puts  "[ expr  ($numudp*$udpsize)/$udptime]
bytes per second"
   exit 0
}
# Schedule events
$ns at 0.1 "$cbr start"
$ns at 2.0 "$ftp start"
$ns at 1.9 "$cbr stop"
$ns at 4.3 "$ftp stop"
$ns at 6.0 "finish"
# Run the simulation
$ns run
```

```
 set ns [new Simulator] set
namfile [open p12.nam w] $ns
namtrace-all $namfile set
tracefile [open p12.tr w] $ns
trace-all $tracefile
proc finish {} {
   global ns namfile tracefile
   $ns flush-trace
close $namfile    close
$tracefile
   exec nam p12.nam &
   exit 0
}
set n0 [$ns node] set
n1 [$ns node] set n2
[$ns node] set n3
[$ns node] set n4
[$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
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n1 $n4 1Mb 10ms DropTail
$ns duplex-link $n2 $n4 1Mb 10ms DropTail
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n0 $n3 orient down
$ns duplex-link-op $n1 $n2 orient left-down
$ns duplex-link-op $n1 $n4 orient down $ns
duplex-link-op $n2 $n4 orient right-down
set udp0 [new Agent/UDP] $ns
attach-agent $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0 set
null0 [new Agent/Null] $ns
attach-agent $n4 $null0 $ns
connect $udp0 $null0 set
udp1 [new Agent/UDP] $ns
attach-agent $n2 $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1 set
null1 [new Agent/Null] $ns
attach-agent $n4 $null1
$ns connect $udp1 $null1
$ns rtproto LS
$ns rtmodel-at 20.0 down $n1 $n4
$ns rtmodel-at 23.0 up $n1 $n4
$ns rtmodel-at 25.0 down $n2 $n4
$ns rtmodel-at 40.0 up $n2 $n4
$udp0 set class_ 1
$udp1 set class_ 2
$ns color 1 Red
$ns color 2 Green
$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr1 start"
$ns at 45 "finish"
$ns run
```

```
# Create Simulator set
ns [new Simulator]
# Open trace and NAM trace
file set ntrace [open p6.tr w]
$ns trace-all $ntrace
set namfile [open p6.nam w]
$ns namtrace-all
$namfile # Finish
Procedure proc Finish {} {
   global ns ntrace namfile
   # Dump all trace data and close the file
   $ns flush-trace
close $ntrace    close
$namfile
   # Execute the nam animation file
   exec nam p6.nam &
   exit 0
}
# Create 3 nodes set
n0 [$ns node] set
n1 [$ns node] set
n2 [$ns node]
# Create duplex-links between the nodes
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
# Define the recv function for the class 'Agent/Ping'
# instproc adds class method called "RECEIVE" to calculate RTT
Agent/Ping instproc recv {from rtt} {
   # instvar adds instance variable, and brings them to the local
scope
   $self instvar node_
   # RTT is the length of time it takes for a signal to be sent plus
the length of time it takes for an acknowledgement of that signal
to be received.    puts "Node $from received ping answer from
Node [$node_ id] with Round Trip Time of $rtt ms"
}
# Create two ping agents and attach them to n(0) and
n(2) set p0 [new Agent/Ping] $ns attach-agent $n0 $p0
set p1 [new Agent/Ping] $ns attach-agent $n2 $p1
$ns connect $p0 $p1
# Schedule events for ping transmission
$ns at 0.2 "$p0 send"
$ns at 0.4 "$p1 send"
$ns at 1.2 "$p0 send"
$ns at 1.7 "$p1 send"
$ns at 1.8 "Finish"
# Run the Simulation
$ns run
```

```
# Create a simulator object
set ns [new Simulator]
# Tell the simulator to use dynamic routing
# Distance vector routing is an asynchronous algorithm in
which node x sends the copy of its distance vector to all its
neighbors. When node x receives the new distance vector
from one of its neighboring vector, v, it saves the distance
vector of v and uses the Bellman-Ford equation to update its
own distance vector.
$ns rtproto DV
# Open the nam trace file set
nf [open p7.nam w]
$ns namtrace-all $nf
# Define a 'finish' procedure
proc finish {} {
global ns nf    $ns
flush-trace    # Close
the trace file    close
$nf
   # Execute nam on the trace file
   exec nam p7.nam &
   exit 0
}
# Create seven nodes
for {set i 0} {$i < 7} {incr i} {
   set n($i) [$ns node]
}
# Create links between the nodes
for {set i 0} {$i < 7} {incr i} {
   $ns duplex-link $n($i) $n([expr ($i+1)%7]) 1Mb 10ms
DropTail
}
# Create a UDP agent and attach it to node n(0)
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $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 Null agent (a traffic sink) and attach it to node
n(3) set null0 [new Agent/Null] $ns attach-agent $n(3) $null0
# Connect the traffic source with the traffic sink
$ns connect $udp0 $null0
# Schedule events for the CBR agent and the network
dynamics
$ns at 0.5 "$cbr0 start"
$ns rtmodel-at 1.0 down $n(1) $n(2)
$ns rtmodel-at 2.0 up $n(1) $n(2)
$ns at 4.5 "$cbr0 stop"
$ns at 5.0 "finish"
# Run the simulation
$ns run
```

```
# Create a simulator object set
ns [new Simulator]
# Open the trace and NAM trace files
set nf [open p8.tr w] $ns
trace-all $nf set ntrace
[open p8.nam w]
$ns namtrace-all $ntrace
# Create 4 nodes
for {set i 0} {$i < 4} {incr i} {
set n($i) [$ns node]
}
# Create duplex links between the nodes
for {set i 0} {$i < 4} {incr i} {
   $ns duplex-link $n($i) $n([expr ($i+1)%4]) 1Mb 10ms DropTail
}
# Create UDP agent and attach it to node 0
set udp [new Agent/UDP]
set null [new Agent/Null] $ns
attach-agent $n(0) $udp
$ns attach-agent $n(1) $null
$ns connect $udp $null
# Create a CBR traffic source and attach it to udp set
cbr [new Application/Traffic/CBR]
$cbr set interval_ 0.005
$cbr set packetSize_ 500
$cbr attach-agent $udp
# Create the second UDP agent and attach it to nodes 1 and
2 set udp1 [new Agent/UDP] set null1 [new Agent/Null] $ns
attach-agent $n(1) $udp1
$ns attach-agent $n(2) $null1
$ns connect $udp1 $null1
# Create a CBR traffic source and attach it to udp1 set
cbr1 [new Application/Traffic/CBR]
$cbr1 set interval_ 0.005
$cbr1 set packetSize_ 500
$cbr1 attach-agent $udp1
# Create the third UDP agent and attach it to nodes 2 and
3 set udp2 [new Agent/UDP] set null2 [new Agent/Null]
$ns attach-agent $n(2) $udp2
$ns attach-agent $n(3) $null2
$ns connect $udp2 $null2
# Create a CBR traffic source and attach it to udp2 set
cbr2 [new Application/Traffic/CBR]
$cbr2 set interval_ 0.005
$cbr2 set packetSize_ 500
$cbr2 attach-agent $udp2
# Create the fourth UDP agent and attach it to nodes 3 and
0 set udp3 [new Agent/UDP] set null3 [new Agent/Null] $ns
attach-agent $n(3) $udp3
$ns attach-agent $n(0) $null3
$ns connect $udp3 $null3
# Create a CBR traffic source and attach it to udp3 set
cbr3 [new Application/Traffic/CBR]
$cbr3 set interval_ 0.005
$cbr3 set packetSize_ 500
$cbr3 attach-agent $udp3 #
Define the finish procedure
proc Finish {} {
global ns nf ntrace
$ns flush-trace
close $nf    close
$ntrace
   exec nam p8.nam &
   exit 0
}
# Schedule the events
$ns at 0.5 "$cbr start"
$ns at 4.5 "$cbr stop"
$ns at 0.5 "$cbr1 start"
$ns at 4.5 "$cbr1 stop"
$ns at 0.5 "$cbr2 start"
$ns at 4.5 "$cbr2 stop"
$ns at 0.5 "$cbr3 start"
$ns at 4.5 "$cbr3 stop"
$ns at 5.0 "Finish"
# Run the simulation
$ns run
```