**StarTopology.Tcl**

#Create a simulator object

set ns [new Simulator]

#Open the nam trace file

set nf [open out.nam w]

$ns namtrace-all $nf

#Define a 'finish' procedure

proc finish {} {

global ns nf

$ns flush-trace

#Close the trace file

close $nf

#Executenam on the trace file

exec nam out.nam &

exit0

}

#Create four 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]

#Change the shape of center node in a star topology

$n0 shape square

#Create links between the nodes

$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 $n0 $n4 1Mb 10ms DropTail

$ns duplex-link $n0 $n5 1Mb 10ms DropTail

#Create a TCP agent and attach it to node n0

set tcp0 [new Agent/TCP]

$tcp0 set class\_ 1

$ns attach-agent $n1 $tcp0

#Create a TCP Sink agent (a traffic sink) for TCP and attach it to node n3

set sink0 [new Agent/TCPSink]

$ns attach-agent $n3 $sink0

#Connect the traffic sources with the traffic sink

$ns connect $tcp0 $sink0

# Create a CBR traffic source and attach it to tcp0

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize\_ 500

$cbr0 set interval\_ 0.01

$cbr0 attach-agent $tcp0

#Schedule events for the CBR agents

$ns at 0.5 "$cbr0 start"

$ns at 4.5 "$cbr0 stop"

#Call the finish procedure after 5 seconds of simulation time

$ns at 5.0 "finish"

#Run the simulation

$ns run

**Routing.Tcl**

#Create a simulator object

set ns [new Simulator]

#Define different colors for data flows

$ns color 1 Blue

#Open the nam trace file

set nf [open out.nam w]

$ns namtrace-all $nf

#Create four nodes

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

set n4 [$ns node]

#Create links between the nodes

$ns duplex-link $n0 $n1 0.1Mb 10ms DropTail

$ns duplex-link $n1 $n4 0.1Mb 10ms DropTail

$ns duplex-link $n3 $n4 0.1Mb 10ms DropTail

$ns duplex-link $n0 $n2 0.1Mb 10ms DropTail

$ns duplex-link $n1 $n3 0.1Mb 10ms DropTail

$ns duplex-link $n2 $n3 0.1Mb 10ms DropTail

#Create a UDP agent and attach it to node n1

set udp0 [new Agent/UDP]

$udp0 set class\_ 1

$ns attach-agent $n0 $udp0

# Create a CBR traffic source and attach it to udp1

set cbr1 [new Application/Traffic/CBR]

$cbr1 set packetSize\_ 50

$cbr1 set interval\_ 0.01

$cbr1 attach-agent $udp0

#Create a Null agent (a traffic sink) for UDP and attach it to node n3

set null0 [new Agent/Null]

$ns attach-agent $n2 $null0

#Connect the traffic sources with the traffic sink

$ns connect $udp0 $null0

# Enable Distance Vector Routing Protocol

$ns rtproto DV

# Schedule Link Shutdowns & Powerups

$ns rtmodel-at 1.0 down $n0 $n2

$ns rtmodel-at 1.5 down $n1 $n3

$ns rtmodel-at 2.0 up $n0 $n2

#Schedule events for the CBR agents

$ns at 0.5 "$cbr1 start"

$ns at 4.5 "$cbr1 stop"

#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 out.nam &

exit 0

}

#Call the finish procedure after 5 seconds of simulation time

$ns at 5.0 "finish"

#Run the simulation

$ns run