#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