Program 2

Simulate a network topology with seven nodes and two duplex links using dynamic routing

```
#Create a simulator object
set ns [new Simulator]
```

This line creates a new simulator object and stores it in a variable called 'ns'.

```
#Tell the simulator to use dynamic routing
$ns rtproto DV
```

This line sets the routing protocol for the network to use dynamic routing with the Distance Vector (DV) algorithm.

```
#Open the nam trace file
set nf [open out.nam w]
```

This line creates a new file object and stores it in a variable called 'nf'. The file will be used to store the trace output for the network animation.

```
$ns namtrace-all $nf
```

This line tells the simulator to trace all network events and store them in the trace file '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 out.nam &
exit 0
}
```

1. `proc finish {} {...}`: This line defines a procedure called 'finish' that will be called at the end of the simulation.
2. `global ns nf`: This line declares that the variables 'ns' and 'nf' are global, so they can be accessed from within the 'finish' procedure.
3. `$ns flush-trace`: This line tells the simulator to write all remaining trace information to the trace file.
4. `close $nf`: This line closes the trace file.
5. `exec nam out.nam &`: This line executes the nam animation tool on the trace file and runs it in the background.
6. `exit 0`: This line exits the script with a success status.

```
#create seven nodes
for {set i 0} {$i < 7} {incr i} {
set n($i) [$ns node]
}
```

This line creates 7 nodes and stores them in an array called 'n'.

```
#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
}
```

This line creates links between the nodes in a ring topology with a bandwidth of 1Mbps, a delay of 10ms, and a DropTail queue.

```
#create a UDP agent and attach it to node n(0)
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
```

1. `set udp0 [new Agent/UDP]`: This line creates a new UDP agent object and stores it in a variable called 'udp0'.
2. `$ns attach-agent $n(0) $udp0`: This line attaches the UDP agent to node n(0).

```
# 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
```

1. `set cbr0 [new Application/Traffic/CBR]`: This line creates a new CBR traffic source object and stores it in a variable called 'cbr0'.
2. `$cbr0 set packetSize_ 500`: This line sets the packet size of the CBR traffic source to 500 bytes.
3. `$cbr0 set interval_ 0.005`: This line sets the interval between sending packets of the CBR traffic source to 5 milliseconds.
4. `$cbr0 attach-agent $udp0`: This line attaches the CBR traffic source to the UDP agent.

```
#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
```

1. `set null0 [new Agent/Null]`: This line creates a new Null agent object and stores it in a variable called 'null0'.
2. `$ns attach-agent $n(3) $null0`: This line attaches the Null agent to node n(3).

```
#connect the traffic source with the traffic sink
$ns connect $udp0 $null0

#Schedule event for the CBR agent and the network dynamics
$ns at 0.5 "$cbr0 start"
```

1. `$ns connect $udp0 $null0`: This line connects the UDP agent to the Null agent.
2. `$ns at 0.5 "$cbr0 start"`: This line schedules an event to start the CBR traffic source at 0.5 seconds.

```
$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"
#call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#run the simulation
$ns run
```

- `$ns rtmodel-at 1.0 down $n(1) $n(2)`: This line sets the link between nodes `n(1)` and `n(2)` to be down at time 1.0 seconds using dynamic routing. This means that the link will be temporarily unavailable for data transmission.
- `$ns rtmodel-at 2.0 up $n(1) $n(2)`: This line sets the link between nodes `n(1)` and `n(2)` to be up at time 2.0 seconds using dynamic routing. This means that the link will be available again for data transmission.
- `$ns at 4.5 "$cbr0 stop"`: This line schedules an event to stop the CBR traffic source attached to the UDP agent (`$udp0`) at time 4.5 seconds.
- `$ns at 5.0 "finish"`: This line schedules an event to call the `finish` procedure at time 5.0 seconds. The `finish` procedure closes the nam trace file and exits the simulation.
- `$ns run`: This line starts the simulation and runs it until all scheduled events have been executed. This command will block until the simulation is complete.

Demonstration

Complete Code

```
#Create a simulator object
set ns [new Simulator]
#Tell the simulator to use dynamic routing
$ns rtproto DV

#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
#Execute nam on the trace file
exec nam out.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 event 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"
#call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#run the simulation
$ns run
```

Code availabe on Github