

Program 1

A simple network topology with four nodes connected by three duplex links and simulates two different traffic flows (CBR traffic) in the network.

```
set ns [new Simulator]
```

This creates a new NS-2 simulator object and assigns it to the variable `ns`.

```
$ns color 1 Blue  
$ns color 2 Red
```

This sets the colors used by the simulation visualization tool (nam) for the different types of packets in the simulation.

```
set nf [open out.nam w]  
$ns namtrace-all $nf
```

This opens a new file called `out.nam` for writing, and tells the simulator to log all events to this file for use by the nam visualization tool.

```
proc finish {} {  
    global ns nf  
    $ns flush-trace  
    close $nf  
    exec nam out.nam  
    exit 0  
}
```

This defines a procedure called `finish` that is called at the end of the simulation. The procedure flushes the trace data to the output file, closes the file, runs the nam visualization tool to display the simulation results, and exits the script.

```
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
```

This creates four nodes and assigns them to the variables `n0`, `n1`, `n2`, and `n3`.

```
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n3 $n2 1Mb 10ms DropTail
```

This creates three duplex links between the nodes, with 1 Mbps bandwidth, 10 ms delay, and a DropTail queue.

```
$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
```

This sets the orientation of the links in the visualization tool to make the diagram more readable.

```
$ns duplex-link-op $n2 $n3 queuePos 0.5
```

This sets the position of the queue on the link between nodes 2 and 3 to be in the middle of the link.

```
set udp0 [new Agent/UDP]
$udp0 set class_ 1
$ns attach-agent $n0 $udp0

set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
```

This creates an Agent/UDP object called `udp0` and sets its traffic class to 1. It also creates a CBR traffic application called `cbr0` with a packet size of 500 bytes and an inter-packet interval of 5 milliseconds. Finally, it attaches the `udp0` agent to node 0 and attaches the `cbr0` application to `udp0`.

```
set udp1 [new Agent/UDP]
$udp1 set class_ 2
$ns attach-agent $n1 $udp1

set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
```

In this section of the code, a new UDP agent `udp1` is created and attached to node `n1` using the "attach-agent" method of the `ns` object.

Then, a new CBR traffic source application `cbr1` is created with a packet size of 500 bytes and an inter-packet interval of 0.005 seconds. The newly created UDP agent `udp1` is attached to the CBR traffic source application `cbr1` using the "attach-agent" method.

```
set null0 [new Agent/Null]
$ns attach-agent $n3 $null0
```

Next, a new Null agent `null0` is created and attached to node `n3` using the "attach-agent" method of the `ns` object.

```
$ns connect $udp0 $null0
$ns connect $udp1 $null0
```

Then, the UDP agents `udp0` and `udp1` are connected to the `null0` agent using the "connect" method of the `ns` object.

```
$ns at 0.5 "$cbr0 start"
$ns at 1.5 "$cbr1 start"
```

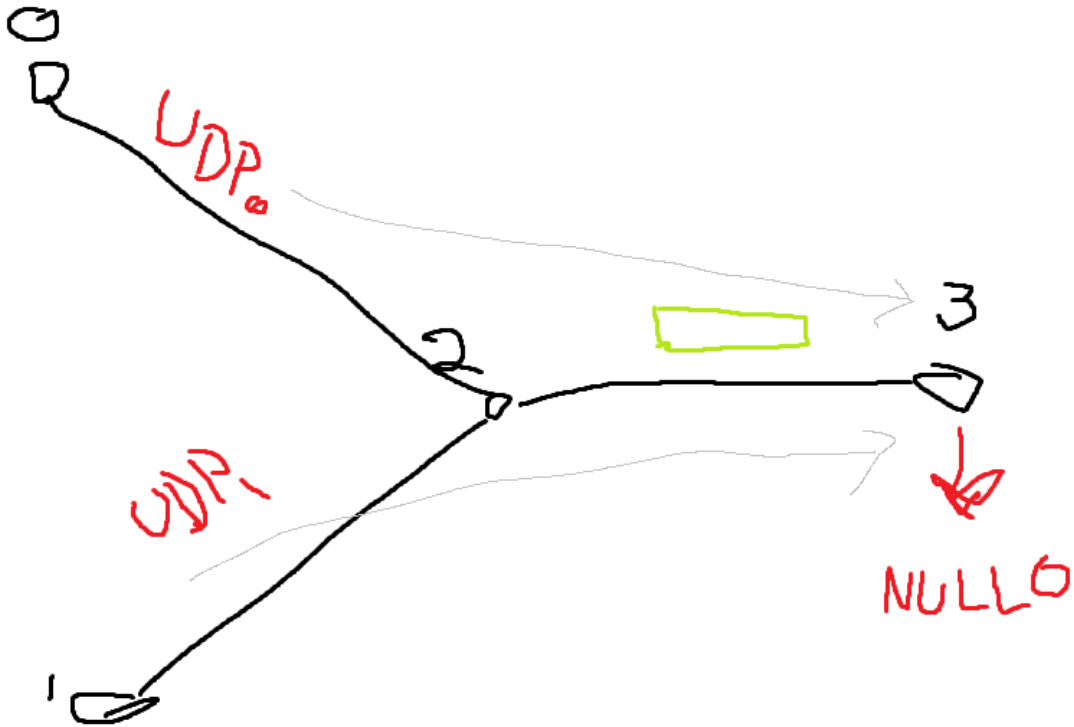
```
$ns at 4.5 "$cbr0 stop"
```

After that, three events are scheduled using the "at" method of the ns object:

- At time 0.5, the cbr0 application starts sending traffic using the "start" method.
- At time 1.5, the cbr1 application starts sending traffic using the "start" method.
- At time 4.5, the cbr0 application stops sending traffic using the "stop" method.

```
$ns at 5.0 "finish"  
$ns run
```

Finally, an event is scheduled to terminate the simulation at time 5.0 using the "at" method and the "finish" procedure is called, which flushes the trace information and closes the trace file, and then the nam visualization tool is launched to display the simulation trace. The "run" method is called to start the simulation.



Overall, this section of the code sets up a simple network topology with 4 nodes connected in a chain, and two CBR applications that send traffic from nodes n0 and n1 to node n3 via node n2. The traffic is routed using DropTail queues and the simulation is scheduled to run for 5 seconds.

Complete code

```

set ns [new Simulator]

$ns color 1 Blue
$ns color 2 Red

set nf [open out.nam w]
$ns namtrace-all $nf

proc finish {} {
global ns nf

```

```
$ns flush-trace
close $nf
exec nam out.nam
exit 0
}
```

```
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
```

```
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n3 $n2 1Mb 10ms DropTail
```

```
$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
```

```
$ns duplex-link-op $n2 $n3 queuePos 0.5
```

```
set udp0 [new Agent/UDP]
$udp0 set class_ 1
$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 udp1 [new Agent/UDP]
$udp1 set class_ 2
$ns attach-agent $n1 $udp1
```

```
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
```

```
set null0 [new Agent/Null]  
$ns attach-agent $n3 $null0
```

```
$ns connect $udp0 $null0  
$ns connect $udp1 $null0
```

```
$ns at 0.5 "$cbr0 start"  
$ns at 1.5 "$cbr1 start"  
$ns at 4.5 "$cbr0 stop"
```

```
$ns at 5.0 "finish"  
$ns run
```