

# Explanation of the script 'simpleSim.tcl'

First of all, you need to create a simulator object. You can do this with the following command:

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

Now we open a file for writing that is going to be used by the network animator nam.

```
set file1 [open out.nam w]
```

```
$ns namtrace-all $file1
```

The first line opens the 'out.nam' file for writing and gives it the handle 'file1'. The second line tells the simulator object to write all simulation data related to nam into this file. The same principle holds for the out.tr trace file (the following 2 lines).

The next step is to add a 'finish' procedure that closes the trace file and starts nam.

```
proc finish {} {  
    global ns file1 file2  
  
    $ns flush-trace  
  
    close $file1  
  
    close $file2  
  
    exec nam out.nam &  
  
    exit 0  
}
```

You don't really have to understand everything in the code above, yet. Focus on the bigger picture, and it will all become clear to you once you see what the rest of the code does.

Now, we create two nodes, n0 and n1:

```
set n0 [$ns node]
```

```
set n1 [$ns node]
```

We connect them:

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

This line tells the simulator object to connect the two nodes with a duplex link that has bandwidth 1Megabit and delay 10ms. It also instructs the link to use the 'DropTail' queue; when we try to queue a new packet on an already-full link, the link simply drops that new packet, instead.

Next, we create an agent object that sends data from node n0, and another agent object that receives the data on node n1.

First, we create a UDP (transport-layer) agent and attach it to the node n0, then we attach a 'constant bit rate' (CBR) traffic generator to the UDP agent. The packetSize is set to 500 bytes, and a packet will be sent every 0.005 seconds. You can find the relevant parameters for each agent type in the [ns manual page](#).

```
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
```

Then, we create a Null agent which acts as a traffic sink, and attach it to node n1.

```
set null0 [new Agent/Null]

$ns attach-agent $n1 $null0
```

Next, we connect the two transport-layer 'agents' to each other.

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

Afterwards, we instruct the CBR agent when to send data and when to stop sending.

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

$ns at 1.5 "$cbr0 stop"
```

The next line tells the simulator object to execute the 'finish' procedure after 2.0 seconds of simulation time. You probably understand what this line does just by looking at it. ns provides you with a very simple way to schedule events with the 'at' command.

```
$ns at 2.0 "finish"
```

Finally, the last line starts the simulation.

```
$ns run
```

Once your script is ready, you can run it with:

```
ns simpleSim.tcl
```

When you click on the 'play' button in the nam window, you will see that after 0.5 seconds of simulation, node 0 starts sending data packets to node 1. You might want to slow nam down by using the 'Step' slider.