

Chapter 1

Introduction

Let's start at the very beginning,
a very nice place to start,
when you sing, you begin with A, B, C,
when you simulate, you begin with the topology,¹
...

This document (*ns Notes and Documentation*) provides reference documentation for ns. Although we begin with a simple simulation script, resources like Marc Greis's tutorial web pages (originally at his web site, now at <http://www.isi.edu/nsnam/ns/tutorial/>) or the slides from one of the ns tutorials are probably better places to begin for the ns novice.

We first begin by showing a simple simulation script. This script is also available in the sources in `~ns/tcl/ex/simple.tcl`.

This script defines a simple topology of four nodes, and two agents, a UDP agent with a CBR traffic generator, and a TCP agent. The simulation runs for *3s*. The output is two trace files, `out.tr` and `out.nam`. When the simulation completes at the end of *3s*, it will attempt to run a nam visualisation of the simulation on your screen.

```
# The preamble
set ns [new Simulator]                                ;# initialise the simulation

# Predefine tracing
set f [open out.tr w]
$ns trace-all $f
set nf [open out.nam w]
$ns namtrace-all $nf
```

¹with apologies to Rodgers and Hammerstein

```

# so, we lied. now, we define the topology
#
#      n0
#      |
#      5Mb \
#      2ms  \
#            \
#            n2 ----- n3
#            /          1.5Mb
#      5Mb /          10ms
#      2ms /
#      /
#      n1
#
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n2 5Mb 2ms DropTail
$ns duplex-link $n1 $n2 5Mb 2ms DropTail
$ns duplex-link $n2 $n3 1.5Mb 10ms DropTail

# Some agents.
set udp0 [new Agent/UDP]                                ; # A UDP agent
$ns attach-agent $n0 $udp0                                ; # on node $n0
set cbr0 [new Application/Traffic/CBR]                    ; # A CBR traffic generator agent
$cbr0 attach-agent $udp0                                   ; # attached to the UDP agent
$udp0 set class_ 0                                         ; # actually, the default, but...

set null0 [new Agent/Null]                                ; # Its sink
$ns attach-agent $n3 $null0                                ; # on node $n3

$ns connect $udp0 $null0
$ns at 1.0 "$cbr0 start"

puts [$cbr0 set packetSize_]
puts [$cbr0 set interval_]

# A FTP over TCP/Tahoe from $n1 to $n3, flowid 2
set tcp [new Agent/TCP]
$tcp set class_ 1
$ns attach-agent $n1 $tcp

set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink

set ftp [new Application/FTP]                                ; # TCP does not generate its own traffic
$ftp attach-agent $tcp
$ns at 1.2 "$ftp start"

$ns connect $tcp $sink
$ns at 1.35 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

```


The simulation runs for 3s.
The simulation comes to an end when the scheduler invokes the finish{} procedure below.
This procedure closes all trace files, and invokes nam visualization on one of the trace files.

```
$ns at 3.0 "finish"  
proc finish {} {  
    global ns f nf  
    $ns flush-trace  
    close $f  
    close $nf  
  
    puts "running nam..."  
    exec nam out.nam &  
    exit 0  
}
```

Finally, start the simulation.
\$ns run