NS 2

TCP/IP model

Application Layer

Applications agents are Telnet, FTP, Rlogin, CBR

Transport Layer

The transport uses two protocols, UDP and TCP. UDP which stands for User Datagram Protocol does no guarantee packet delivery and applications which use this must provide their own means of verifying delivery. TCP does guarantee delivery of packets to the applications which use it.

Network Layer

The network layer is concerned with packet routing and used low level protocols such as ICMP, IP, and IGMP. Here Network protocols are omitted for simple examples

Link Layer

The link layer is concerned with the actual transmittal of packets as well as IP to Ethernet address translati Data transfer in two modes i.e. simplex mode and in duplex mode.

All the links are created at data link layer. Network layer next is to data link layer

Fundamental Skills (II)

- NS2 is written in C++ and OTcl
  - OTcl = Tcl + OO
  - C++ implements the code that executed frequently
  - OTcl configures the system

```
set ns [new Simulator]
set n1 [new Node]
set n2 [new Node]
$ns duplex-link $n1 $n2 5Mb 2ms
```

DropTail

n1 n2

Where can you **write Scripts**?;

After login into terminal, follow step given below

1. vi filename.tcl
   It will open page, there you can write tcl scripts.

2. save file
   Press esc-> colon (shift + semicolon) ->wq (save and quit)
It save the file

3. To run tcl script
   ns filename.tcl

Basically **NS 2 programming** contains the following steps.

1.Create the event scheduler

2.Turn on tracing

3.Creating network

a) Computing setup routing - rtproto

b) Creating transport connection-Agents
c) Creating traffic-Applications

4. Monitoring

a) Visualization using nam

How to use Trace?

We use simulator to see results. How is it achieved? Using trace

Two types of trace

1. generic trace
   for use with xgraph, and other things

2. nam trace
   for use with visualization

```
# open trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf
```

Since we have started tracing, we should end also. For this we use finish procedure.

```
#Define a 'finish' procedure
proc finish {}
{
        global ns tracefile nf
        $ns flush-trace
        close $nf
        close $tracefile          # close tracefile
        exec nam out.nam &   #Execute nam on the trace file
        exit 0
}
```

Finish procedure is forced to be called at the end with the line

Example1:(CBR over UDP)

```
set ns [new Simulator]
```
Creatingns simulator object

```
set tracefile [open out.tr w]
$ns trace-all $tracefile
```
Open trace file

```
set nf [open out.nam w]
$ns namtrace-all $nf
```
Open the nam trace file

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

```
set n0 [$ns node]
set n1 [$ns node]
$ns simplex-link $n0 $n1 1Mb 10ms
DropTail
```
Create your topology
- set n0 nodes....
- $ns duplex-links...

```
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr[new Application/Traffic/CBR]
$cbr attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$ns connect $udp0 $null0
```
Create your agents
-transport layer and application layers stuff

Scheduling Events

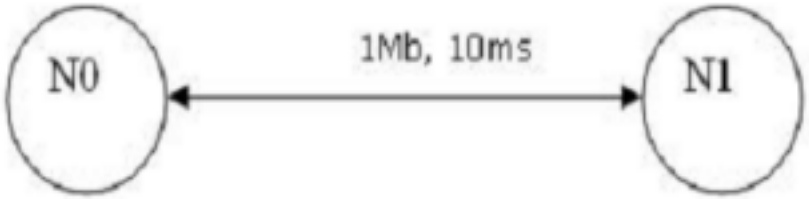
Template

Every ns2 script starts with creating simulator object set ns [new Simulator]

How to create node

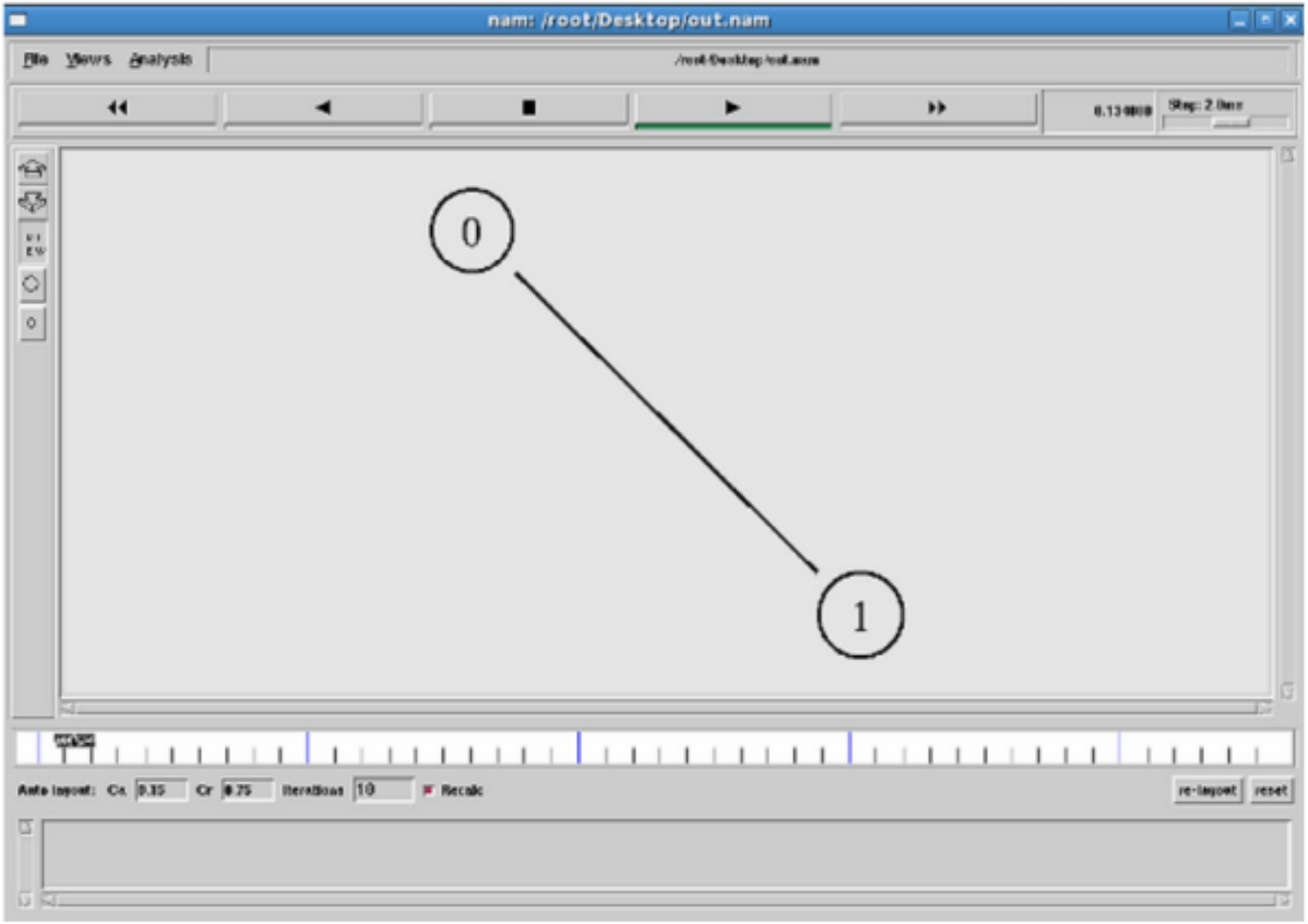
set n0 [$ns node]
set n1 [$ns node]

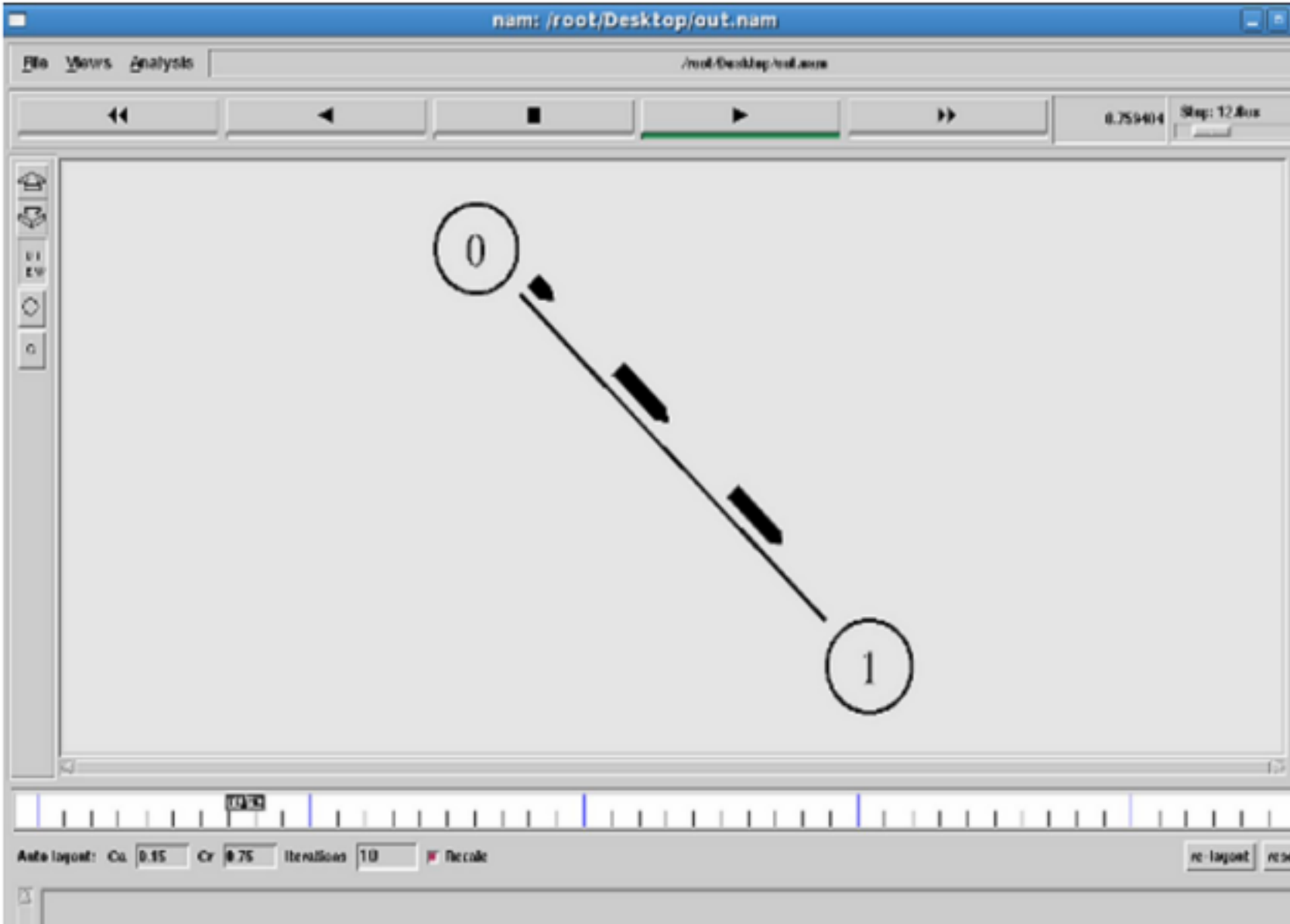
Creating link



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

This line tells the simulator object to connect the nodes n0 and n1 with a duplex link with the bandwidth.1Megabit, a delay of 10ms and a DropTail queue.


nam: /root/Desktop/out.nam
File
Views
Analysis
0
1
Recalc
re-layout
reset

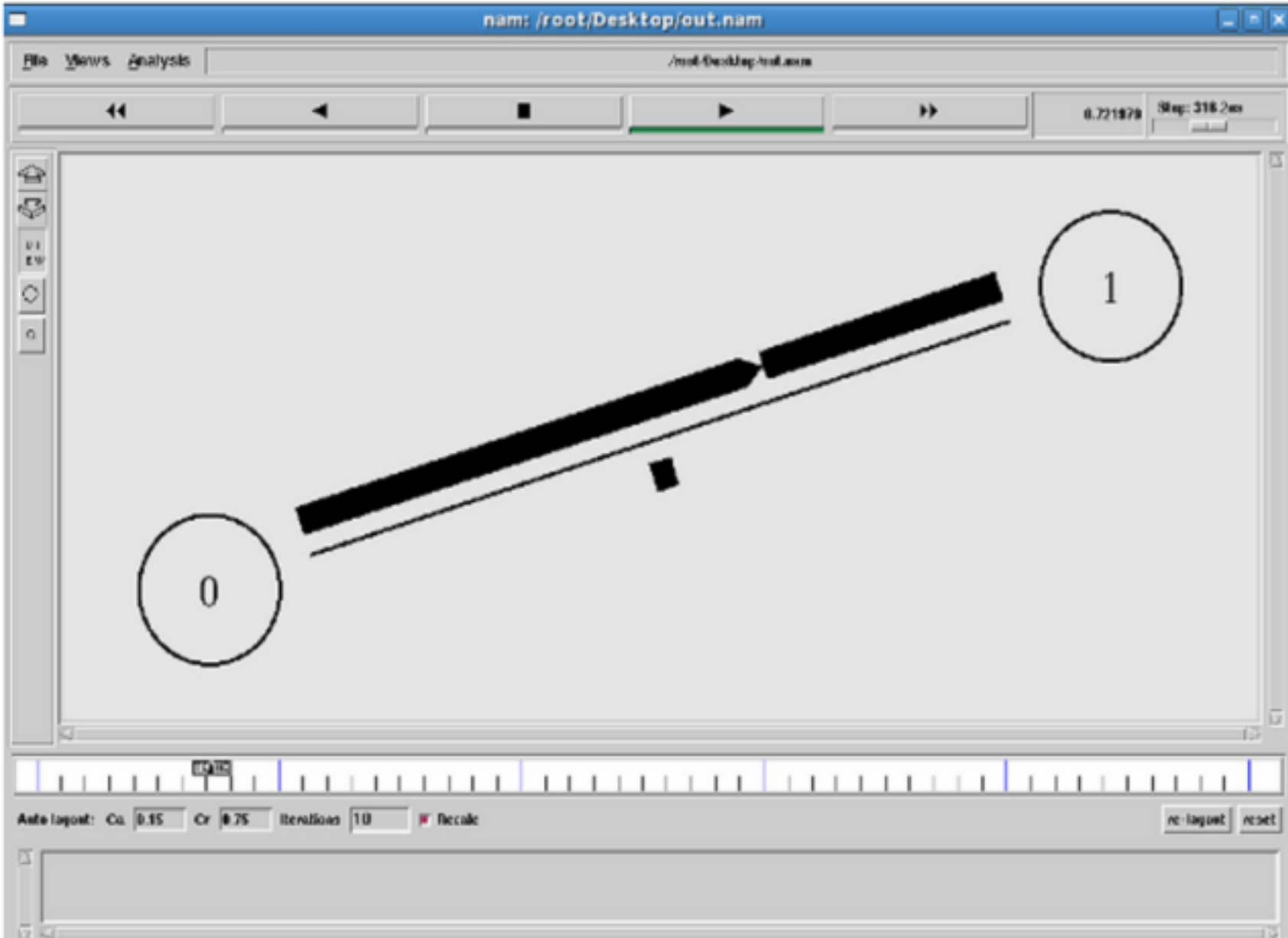

nam: /root/Desktop/out.nam
File
Views
Analysis
0
1

Example 2:(CBR over TCP)

```
set ns [new Simulator]
```
Creatingns simulator object

```
set tracefile [open out.tr w]
$ns trace-all $tracefile
```
Open trace file

```
set nf [open out.nam w]
$ns namtrace-all $nf
```
Open the nam trace file


nam: /root/Desktop/out.nam
File
Views
Analysis
1
0

Example3:(FTP over TCP)

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

Creatingns simulator objec

```
set tracefile [open out.tr w]
$ns trace-all $tracefile
```

Open trace file

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

Open the nam trace file

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

'finish' procedure

```
set n0 [$ns node]
set n1 [$ns node]
$ns simplex-link $n0 $n1 1Mb 10ms
DropTail
```

Create your topology
- set n0 nodes....
- $ns duplex-links...

```
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
set tcpsink0 [new Agent/TCPSink]
$ns attach-agent $n1 $tcpsink0
$ns connect $tcp0 $tcpsink0
```

Create your agents
-transport layer and application layers stuff

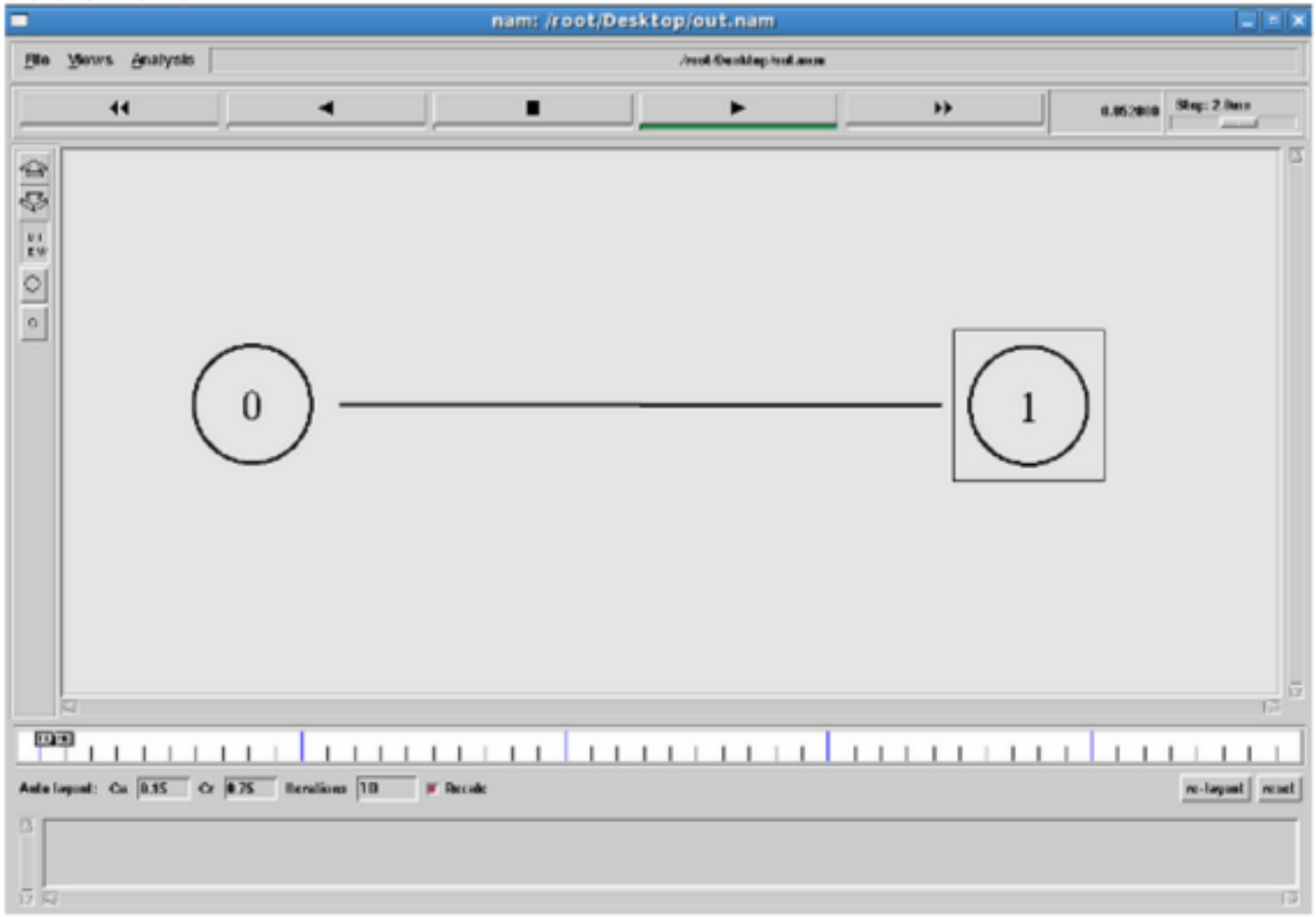
```
$ns at 1.0 "$ftp0 start"
$ns at 3.0 "finish"
```

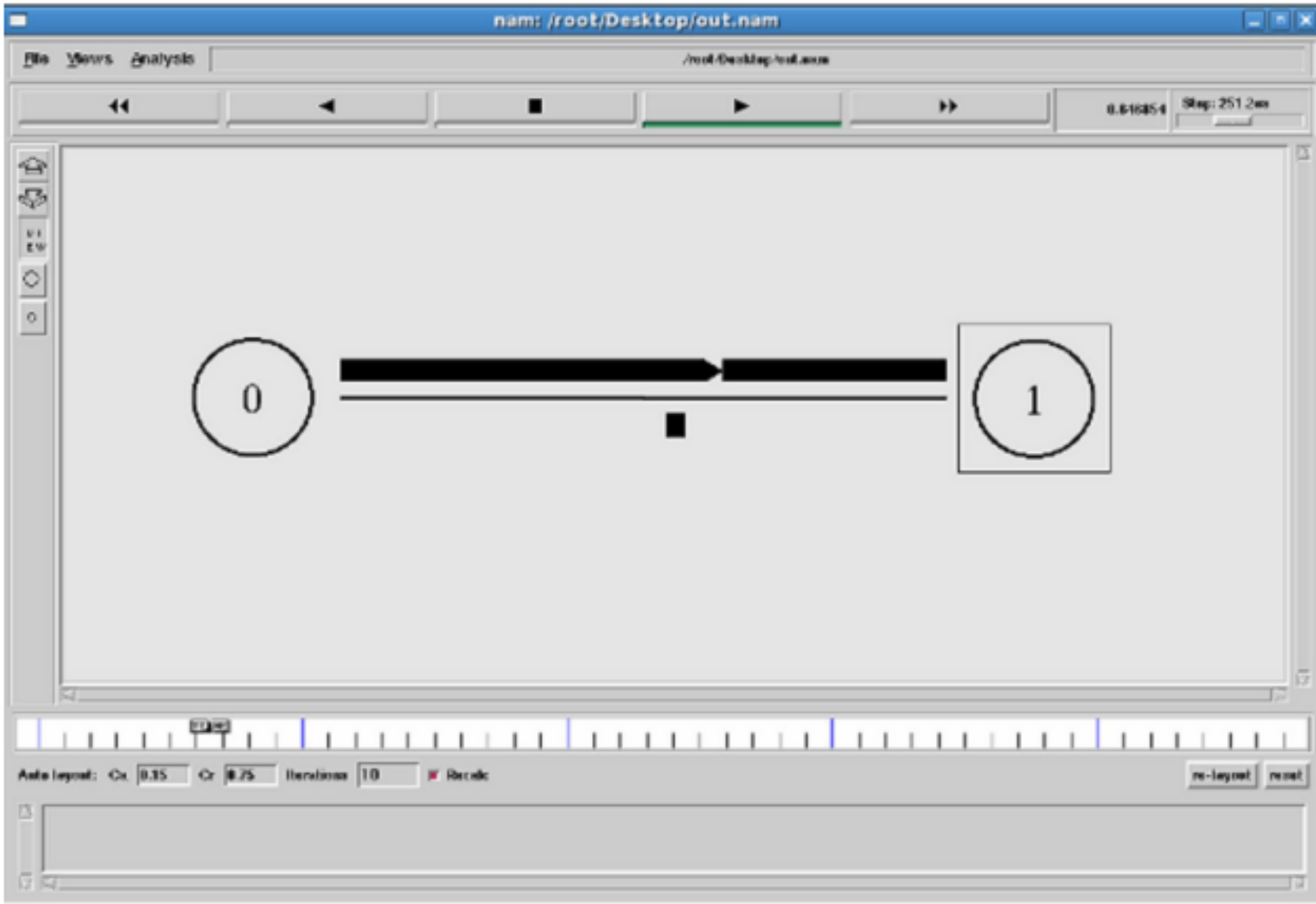
Scheduling Events
- $ns at 1.0 start
and at 3.0 finish

```
$ns run
```

starts the simulation

Before 1.0ms




nam: /root/Desktop/out.nam
File
Views
Analysis
0
1
Auto layout:
re-layout
reset

Wireless Nodes

- set n0 [$ns node]
- set n1 [$ns node]
- set n2 [$ns node]
- set n3 [$ns node]
- set n4 [$ns node]
- set n5 [$ns node]

Communication Range

- $ns initial_node_pos $n0 20
- $ns initial_node_pos $n1 20
- $ns initial_node_pos $n2 20
- $ns initial_node_pos $n3 20
- $ns initial_node_pos $n4 20
- $ns initial_node_pos $n5 50

Initial coordinates of the nodes

- $n0 set X_ 10.0
- $n0 set Y_ 20.0

- $n0 set Z_ 0.0

- $n1 set X_ 210.0
- $n1 set Y_ 230.0
- $n1 set Z_ 0.0

- $n2 set X_ 100.0
- $n2 set Y_ 200.0
- $n2 set Z_ 0.0

Initial coordinates of the nodes

- $n3 set X_ 150.0
  $n3 set Y_ 230.0
  $n3 set Z_ 0.0

  $n4 set X_ 430.0
  $n4 set Y_ 320.0
  $n4 set Z_ 0.0

  $n5 set X_ 270.0
  $n5 set Y_ 120.0
  $n5 set Z_ 0.0
  moty o te noes

#At what Time? Which node? Where to? at What Speed?

- $ns at 1.0 "$n1 setdest 490.0 340.0 25.0" •

$ns at 1.0 "$n4 setdest 300.0 130.0 5.0" •

$ns at 1.0 "$n5 setdest 190.0 440.0 15.0" •

$ns at 20.0 "$n5 setdest 100.0 200.0 30.0"

#creation of agents

- set tcp [new Agent/TCP]

- set sink [new Agent/TCPSink]
- $ns attach-agent $n0 $tcp •

$ns attach-agent $n5 $sink •

$ns connect $tcp $sink

- set ftp [new Application/FTP]
- $ftp attach-agent $tcp
- $ns at 1.0 "$ftp start"

#creation of agents

- set udp [new Agent/UDP]

- set null [new Agent/Null]
- $ns attach-agent $n2 $udp
- $ns attach-agent $n3 $null
- $ns connect $udp $null
- set cbr [new Application/Traffic/CBR]
- $cbr attach-agent $udp
- $ns at 1.0 "$cbr start"

Finish

- $ns at 30.0 "finish"

- proc finish {} {
- global ns tracefile namfile
- $ns flush-trace
- close $tracefile
- close $namfile
- exit 0
- }