Sandeep Gupta

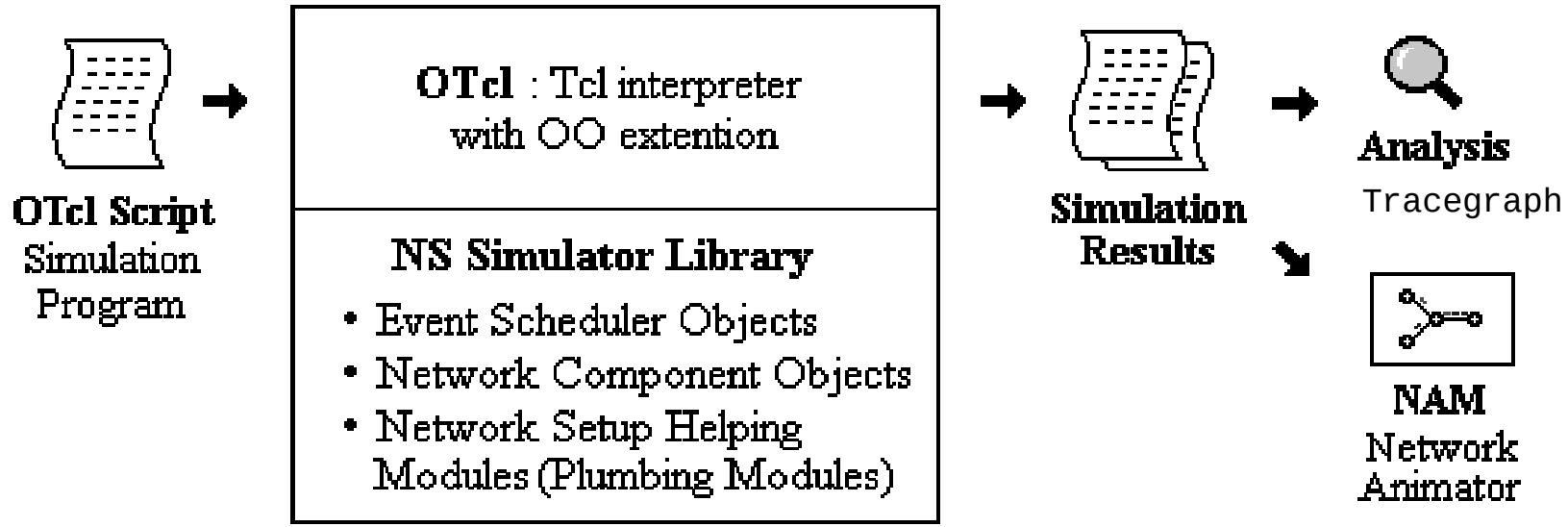
A brief guide to NS-2

Topics

- Introduction
- Writing scripts
- Writing wireless scripts
- Post simulation analysis
  - NAM
  - Tracegraph
- Useful links

Introduction

"NS (version 2) is an object-oriented, discrete event driven network simulator written in C++ and Otcl"



Topics

- Introduction
- Writing scripts
- Writing wireless scripts
- Post simulation analysis
  - NAM
  - Tracegraph
- Useful links

Writing Scripts

'example1.tcl'

Creating a simulator object

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

Open file for writing trace data

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

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

Add procedure finish

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

Writing Scripts

'example1.tcl'

When to call procedure finish

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

Start simulation

```
$ns run
```

Adding nodes

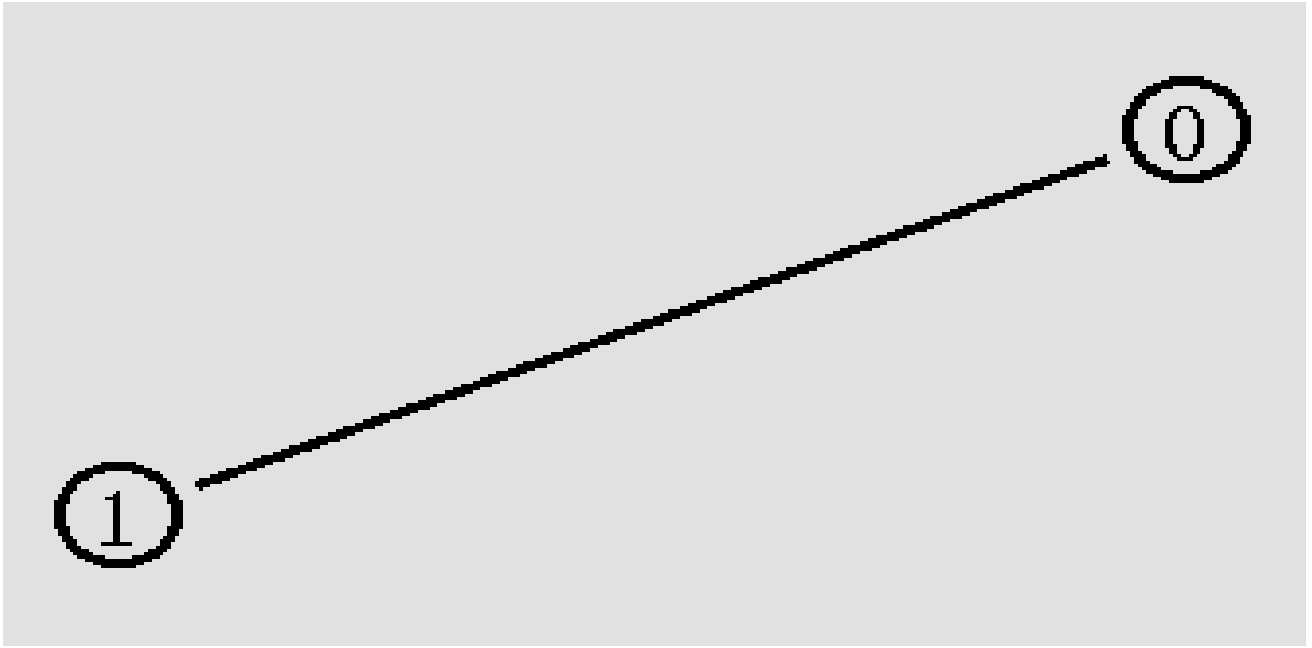
```
set n0 [$ns node]
set n1 [$ns node]
```

Connect two nodes

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

Writing Scripts

'example1.tcl'



Writing Scripts

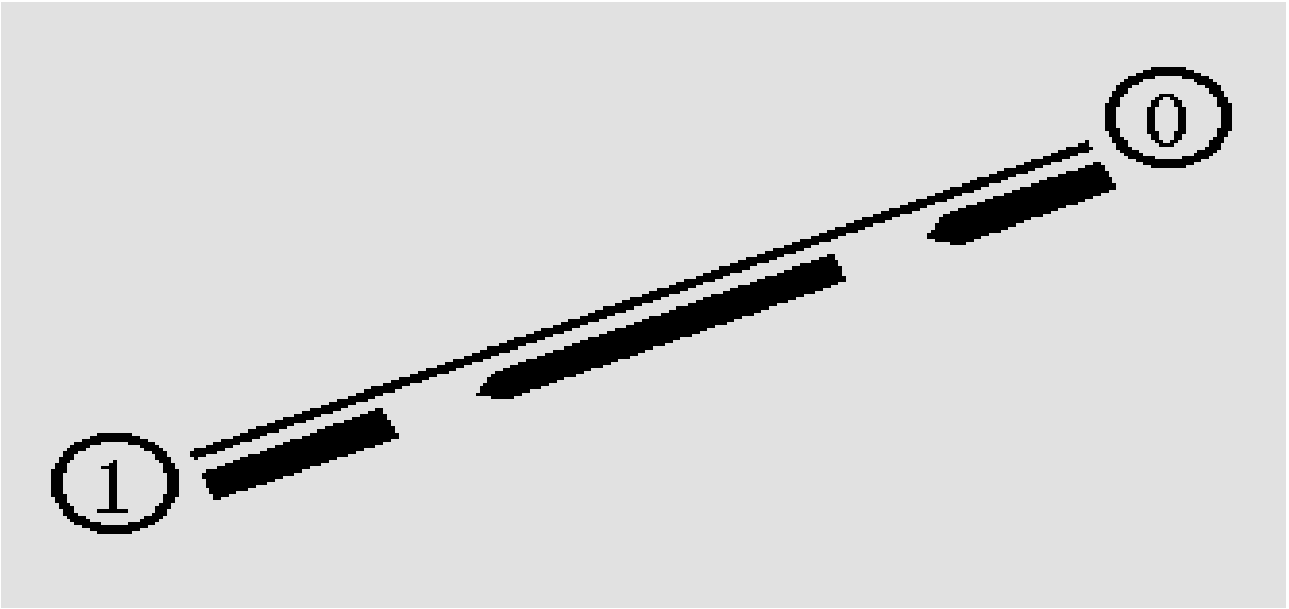
'example1.tcl'

Adding agents and traffic source

```
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
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$ns connect $udp0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```

Writing Scripts

'example1.tcl'



Topics

- Introduction
- Writing scripts
- Writing wireless scripts
- Post simulation analysis
  - NAM
  - Tracegraph
- Useful links

Writing Wireless Scripts

'example2.tcl'

Defining wireless options

```
set val(chan)           Channel/WirelessChannel          ;# channel type
set val(prop)           Propagation/TwoRayGround ;# radio-propagation
  model
set val(ant)            Antenna/OmniAntenna           ;# Antenna type
set val(ll)             LL                           ;# Link layer type
set val(ifq)            Queue/DropTail/PriQueue  ;# Interface queue type
set val(ifqlen)         50                           ;# max packet in ifq
set val(netif)          Phy/WirelessPhy      ;# network interface type
set val(mac)            Mac/802_11                   ;# MAC type
set val(rp)             DSDV                 ;# ad-hoc routing protocol
set val(nn)             2                     ;# number of mobilenodes
```

Writing Wireless Scripts

'example2.tcl'

Create & define topology

```
set topo   [new Topography]
$topo load_flatgrid 500 500        #500m x 500m
```

Create God – General Operations Descriptor

```
create-god $val(nn)
```

Writing Wireless Scripts

'example2.tcl'

Configure node

```
$ns_ node-config -adhocRouting $val(rp) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
                 -antType $val(ant) \
                 -propType $val(prop) \
                 -phyType $val(netif) \
                 -topoInstance $topo \
                 -channelType $val(chan) \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace OFF \
                 -movementTrace OFF
```

Writing Wireless Scripts

'example2.tcl'

Create nodes

```
for {set i 0} {$i < $val(nn) } {incr i} {
                set node_($i) [$ns_ node ]
                $node_($i) random-motion 0   ;# disable random motion
        }
```

Provide start positions

```
$node_(0) set X_ 5.0
$node_(0) set Y_ 2.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 390.0
$node_(1) set Y_ 385.0
$node_(1) set Z_ 0.0
```

Writing Wireless Scripts

'example2.tcl'

Create movement

```
$ns_ at 50.0 "$node_(1) setdest 25.0 20.0 15.0"
$ns_ at 10.0 "$node_(0) setdest 20.0 18.0 1.0"
$ns_ at 100.0 "$node_(1) setdest 490.0 480.0 15.0"
```

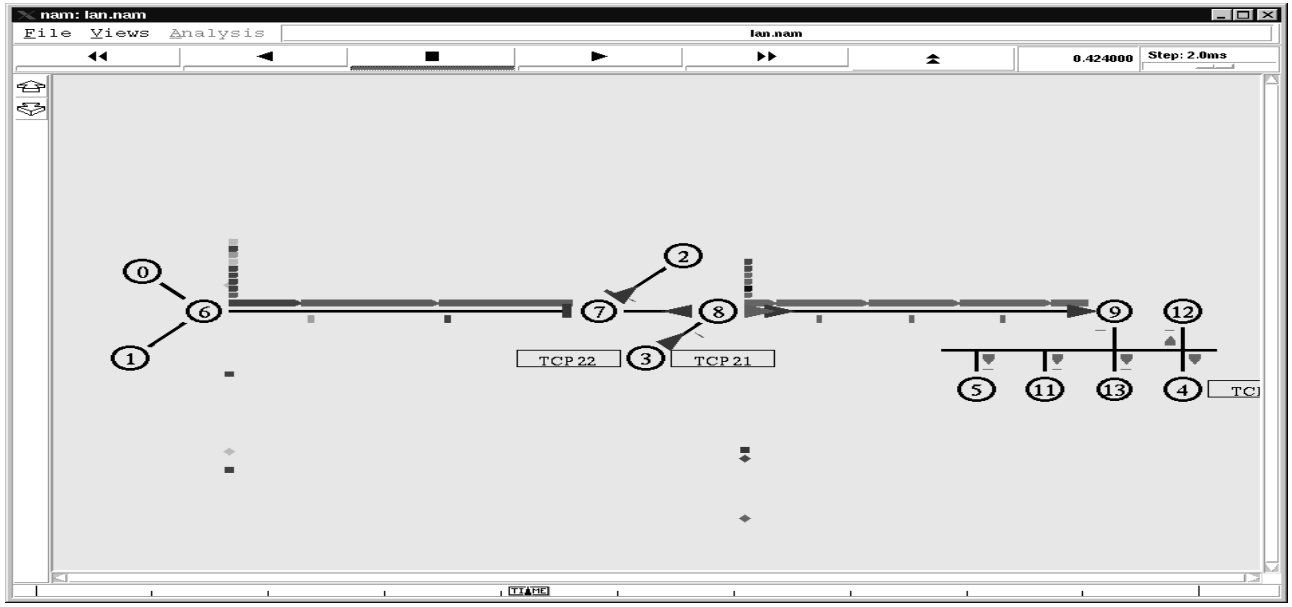
Topics

- Introduction
- Writing scripts
- Writing wireless scripts
- Post simulation analysis
  - NAM
  - Tracegraph
- Useful links

Post Simulation Analysis

Network Animator – NAM

“ Nam is a Tcl/TK based animation tool for viewing network simulation traces and real world packet traces.”



Post Simulation Analysis

Trace File format

event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id

r : receive (at to_node)
+ : enqueue (at queue)
- : dequeue (at queue)
d : drop (at queue)

src_addr : node.port (3.0)
dst_addr : node.port (0.0)

```
r 1.3556 3 2 ack 40 ------- 1 3.0 0.0 15 201
+ 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
- 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
r 1.35576 0 2 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
d 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
- 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
```

Post Simulation Analysis

TraceGraph

“ Trace graph is a Network Simulator ns-2 trace files analyser.”

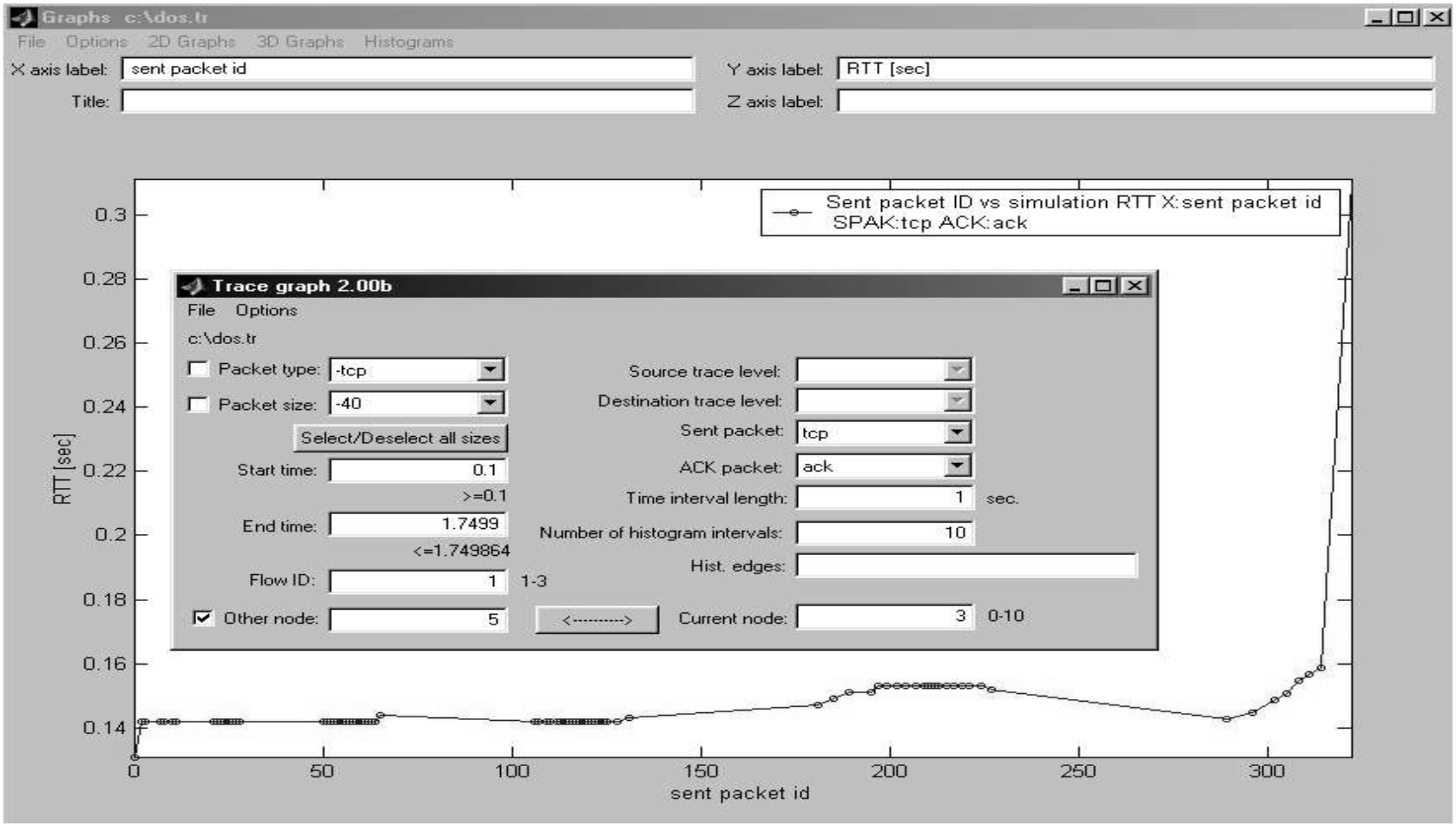
Supported ns-2 trace file formats:

- wired
- satellite
- wireless (old and new trace)
- new trace
- wired-cum-wireless

Requires Matlab 6.0 libraries

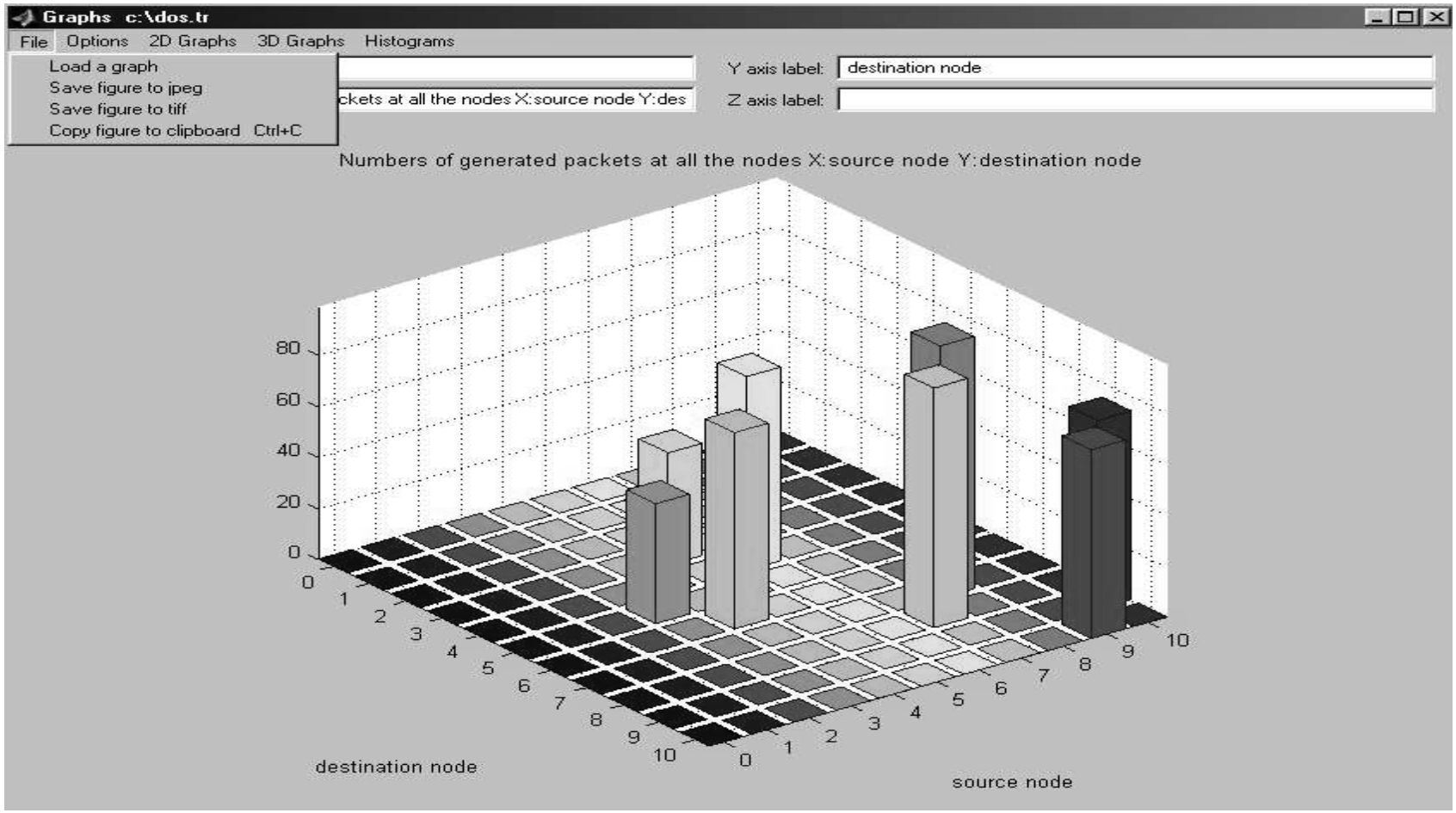
Post Simulation Analysis

TraceGraph



Post Simulation Analysis

TraceGraph



Topics

- Introduction
- Writing scripts
- Writing wireless scripts
- Post simulation analysis
  - NAM
  - Tracegraph
- Useful links

Useful Links

- The Network Simulator – ns 2 http://www.isi.edu/nsnam/ns/index.html
- Marc Greis's tutorial http://www.isi.edu/nsnam/ns/tutorial/index.html
- ns by example – http://nile.wpi.edu/NS/
- The ns documentation - http://www.isi.edu/nsnam/ns/ns-documentation.html
- ns users mailing list (ns-users@isi.edu)
- Tracegraph (www.geocities.com/tracegraph)
- My ns page – http://profile.iiita.ac.in/sandeep_wc02/