

# ns-2 Tutorial (1)

Multimedia Networking Group,  
The Department of Computer Science, UVA  
Jianping Wang

## Contents:

---

- Objectives of this week
  - What is ns-2?
  - Working with ns-2
  - Tutorial exercise
  - ns-2 internals
  - Extending ns-2
- } Today

Partly adopted from Nicolas' slides.

## Objectives of this week

- **Get some exposure to one of the most useful tools in networking research and development.**
- **Understand and work with a popular network simulator.**
- **Get a better understanding of the networking dynamics.**
- **“Smooth the learning curve”.**

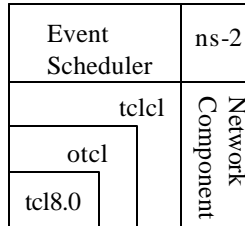
## What is ns-2?

- **ns-2 stands for Network Simulator version 2.**
- **ns-2:**
  - **Is a discrete event simulator for networking research**
  - **Work at packet level.**
  - **Provide substantial support to simulate bunch of protocols like TCP, UDP, FTP, HTTP and DSR.**
  - **Simulate wired and wireless network.**
  - **Is primarily Unix based.**
  - **Use TCL as its scripting language.**
- **ns-2 is a standard experiment environment in research community.**

## What is ns-2 (cont.) ?



You are here.



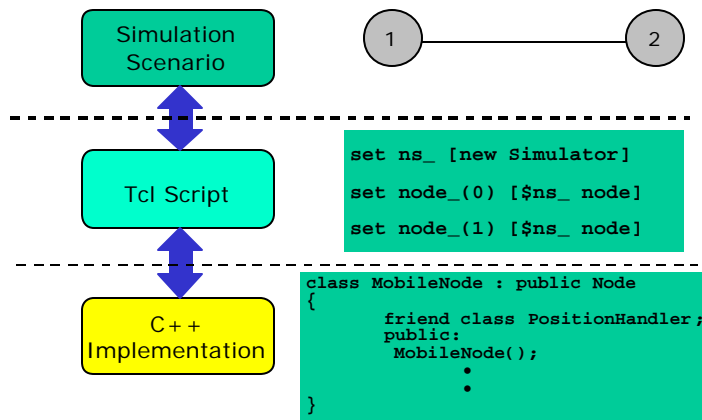
- otcl: Object-oriented support
- tclcl: C++ and otcl linkage
- Discrete event scheduler
- Data network (the Internet) components

Jianping Wang, 2004

cs757

5

## ns-2 implementation



Jianping Wang, 2004

cs757

6

## Why two language? (Tcl & C++)

- C++: Detailed protocol simulations require systems programming language
  - byte manipulation, packet processing, algorithm implementation
  - Run time speed is important
  - Turn around time (run simulation, find bug, fix bug, recompile, re-run) is slower
- Tcl: Simulation of slightly varying parameters or configurations
  - quickly exploring a number of scenarios
  - iteration time (change the model and re-run) is more important

Jianping Wang, 2004

cs757

7

## Other network simulators

- **OPNET (<http://www.opnet.com>)**
  - Leading Commercial Software
  - Support Windows and Unix
  - Graphical Interface
  - Not free
- **GloMoSim (<http://pcl.cs.ucla.edu/projects/gloMosim>)**
  - Simulation environment for wireless network
  - Scalable to support thousands of nodes
  - Using layered approach to build different simulation layers
  - Free for educational users
- **More Resources**
  - <http://www.icir.org/models/simulators.html>

Jianping Wang, 2004

cs757

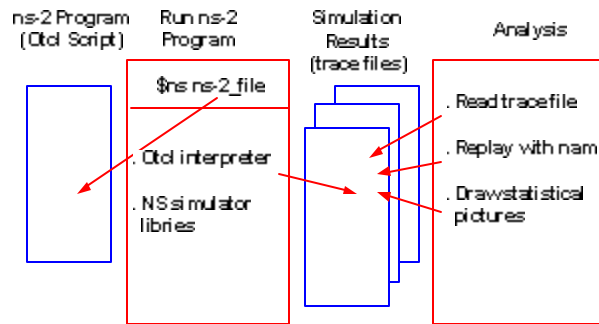
8

# Working with ns-2

## Getting started: install ns-2

- **Download software package from:**  
<http://www.isi.edu/nsnam/ns/>
  - **Easy installation way: all at once**
- **The latest version is 2.27 released at Jan 18, 2004. It contains:**
  - Tk release 8.4.5
  - Tk release 8.4.5
  - Otcl release 1.8
  - TclCL release 1.15
  - Ns release 2.27
  - Nam release 1.10
  - Xgraph version 12
  - CWeb version 3.4g
  - SGB version 1.0
  - Gt-itm gt-itm and sgb2ns 1.1
  - Zlib version 1.1.4
- **Works on Unix and cygwin for windows 9x/2000/xp.**

## Running ns-2 program



Jianping Wang, 2004

cs757

11

## Hello World - Interactive mode

```
[jwang@iodine jwang]$ ns
% set ns [new Simulator]
_o4
% $ns at 1 "puts \"Hello World!\""
1
% $ns at 1.5 "exit"
2
% $ns run
Hello World!
[jwang@iodine jwang]$
```

Jianping Wang, 2004

cs757

12

## Hello World - Batch mode

simple.tcl:

```
set ns [new Simulator]
$ns at 1 "puts \"Hello World!\""
$ns at 1.5 "exit"
$ns run
```

```
[jwang@iodine jwang]$ ns simple.tcl
```

```
Hello World!
```

```
[jwang@iodine jwang]$
```

## Basic tcl

```
proc test {} {
  set a 43 ; a = 43
  set b 27 ; b = 27
  set c [expr $a + $b] ; c = a + b
  set d [expr [expr $a - $b] * $c] ; d = (a - b) * c
  for {set k 0} {$k < 10} {incr k} { ; for (k=0; k<10; k++)
    puts "k = $k"
  }
}
```

## Basic otcl

```
Class mom
mom instproc greet {} {
    $self instvar age_
    puts "$age_ years old mom:
    How are you doing?"
}

set a [new mom]
$a set age_ 45
set b [new kid]
$b set age_ 15

$a greet
$b greet

Class kid -superclass mom
kid instproc greet {} {
    $self instvar age_
    puts "$age_ years old kid:
    What's up, dude?"
}

$a greet
$b greet
}
Jianping Wang, 2004 cs757 15
```

## Basic ns-2

- Create a new simulator object
- [Turn on tracing]
  - [Open your own trace files]
- Create network (physical layer)
- Create link and queue (data-link layer)
- Define routing protocol
- Create transport connection (transport layer)
- Create traffic (application layer)
- Insert errors



## Creating simulator instance

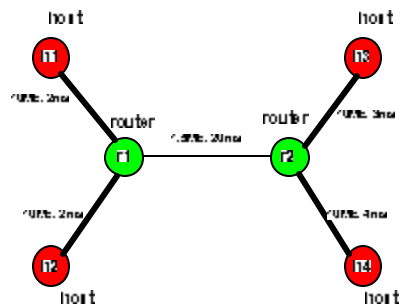
- Create simulator instance  
**set ns [new Simulator]**
  - Usually the first non-comment statement in ns-2 script
  - Initialize the packet format
  - Create a scheduler (default is a calendar scheduler)
  - Create a “null agent”

## Turning on a tracefile

- Open file for NS tracing  
**set f [open out.tr w]**  
**\$ns trace-all \$f**
- Open file for nam tracing  
**set nf [open out.nam w]**  
**\$ns namtrace-all \$nf**
- Open your own trace file  
**set my\_f [open my\_out.tr w]**  
**puts \$my\_f “[\$ns now] [expr \$x(1) + \$y(1)]”**

## Creating a network(1)

- Network topology



Jianping Wang, 2004

cs757

19

## Creating a network (2)

- Creating nodes

```
set node_(h1) [$ns node]
set node_(h2) [$ns node]
set node_(r1) [$ns node]
set node_(r2) [$ns node]
set node_(h3) [$ns node]
set node_(h4) [$ns node]
```

Jianping Wang, 2004

cs757

20

## Creating a network(3)

- Creating Link and Queue

```
$ns duplex-link $node_(h1) $node_(r1)
  10Mb 2ms DropTail
$ns duplex-link $node_(h2) $node_(r2)
  10Mb 3ms DropTail
$ns duplex-link $node_(r1) $node_(r2)
  1.5Mb 20ms DropTail
$ns queue-limit $node_(r1) $node_(r2) 50
.....
```

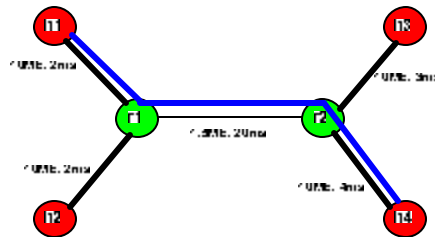
Jianping Wang, 2004

cs757

21

## Creating a TCP connection

```
set tcp0 [$ns create-connection TCP/Reno
  $node_(h1) TCPSink/DelAck $node_(h4) 0]
```



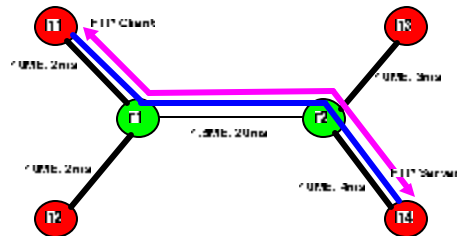
Jianping Wang, 2004

cs757

22

## Creating traffic

- Attaching FTP traffic on the top of TCP  
`set ftp0 [$tcp0 attach-app FTP]`



Jianping Wang, 2004

cs757

23

## Insert errors

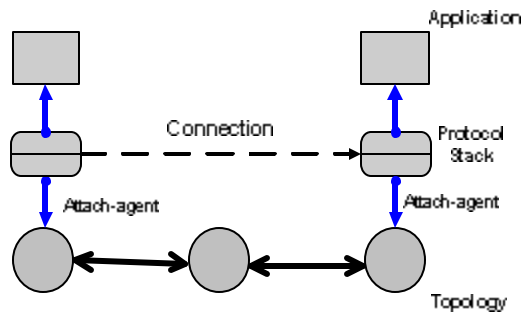
```
set loss_module [new ErrorModel]
$loss_module set rate_ 0.01
$loss_module unit pkt
$loss_module ranvar [new
    RandomVariable/Uniform]
$loss_module drop-target [new Agent/Null]
$ns lossmodel $loss_module $n0 $n1
```

Jianping Wang, 2004

cs757

24

## Summary



Jianping Wang, 2004

cs757

25

## Start/Stop ns

- Schedule an event to start traffic at time 1.0  
**\$ns at 1.0 "\$ftp0 start"**
- Schedule an event to stop ns at time 17.0  
**\$ns at 17.0 "\$ftp0 stop"**
- Start ns  
**\$ns run**  
- last statement
- Stop ns  
**exit 0**

Jianping Wang, 2004

cs757

26

## Visualization tool: nam

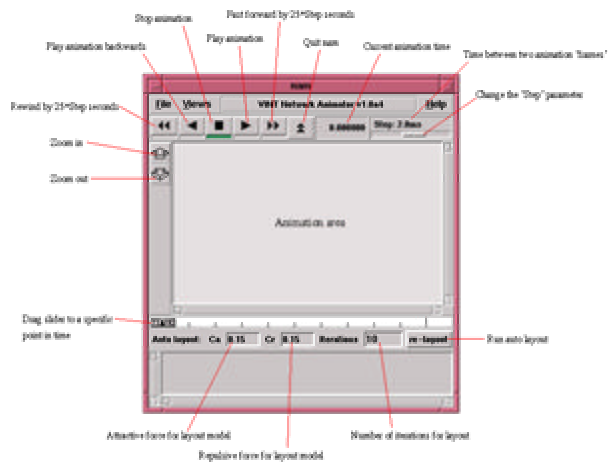
- Replay events from a nam trace file
- The nam trace file can be huge when simulation time is long or events happen intensively. Be careful!
- Run nam:
  - `$nam -a nam_trace_file.nam`
  - In ns-2 script:

```
Proc finish{} {  
    .....  
    exec nam -a nam_trace_file.nam &  
    exit  
}
```

Jianping Wang, 2004

cs757

27



Jianping Wang, 2004

cs757

28

## Draw plots using xgraph

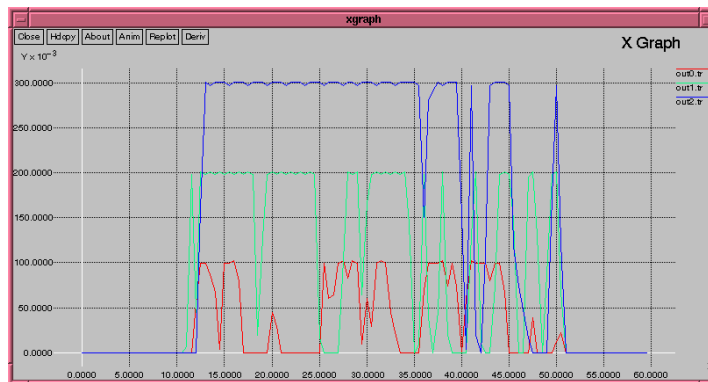
- Create your own output files
- Collect statistical data synchronized.
- Run xgraph:
  - `$xgraph out0.tr, out1.tr -geometry 800x400`
  - In ns-2 script:

```
Proc finish{} {  
.....  
exec xgraph out0.tr, out1.tr out2.tr -geometry  
800x400 &  
exit  
}
```

Jianping Wang, 2004

cs757

29



Jianping Wang, 2004

cs757

30

## Useful URLs

- <http://www.isi.edu/nsnam/ns> - the official ns homepage
- <http://www.isi.edu/nsnam/ns/ns-documentation.html> - NS manual
- [http://jan.netcomp.monash.edu.au/ProgrammingUnix/tcl/tcl\\_tut.html](http://jan.netcomp.monash.edu.au/ProgrammingUnix/tcl/tcl_tut.html) - Tcl tutorial
- <http://bmrc.berkeley.edu/research/cmt/cmtdoc/otcl/> - oTcl tutorial