

CIS 632 / EEC 687

Mobile Computing

MC Platform #5 - Ns-2

Chansu Yu

ns-2 Overview

- What is ns-2?
 - Abbreviation of Network Simulator
 - Discrete event simulator targeted at networking (wired and wireless) research
 - Basically, a TCL interpreter
 - Where to get?
 - Free and open source
 - ns website <http://www.isi.edu/nsnam/ns/>
 - Working platforms
 - Most UNIX or UNIX-like systems; e.g. Linux
 - Windows (using cygwin)
-

ns-2 uses two languages? (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
-

ns-2 Environment

Simulation
Scenario

1

2

Tcl Script

```
set ns_ [new Simulator]
set node_(0) [$ns_ node]
set node_(1) [$ns_ node]
```

C++
Implementation

```
class MobileNode : public Node
{
    friend class PositionHandler;
public:
    MobileNode();
    :
}
}
```

How to Use ns-2?

- ❑ Design simulation
 - Determine simulation scenario, parameters.
 - ❑ Build ns-2 script using tcl
 - If necessary implement algorithm using C++.
 - ❑ Run simulation
 - For convenience use shell batch file.
 - ❑ Analyze simulation results
 - Use shell command or programming languages.
-

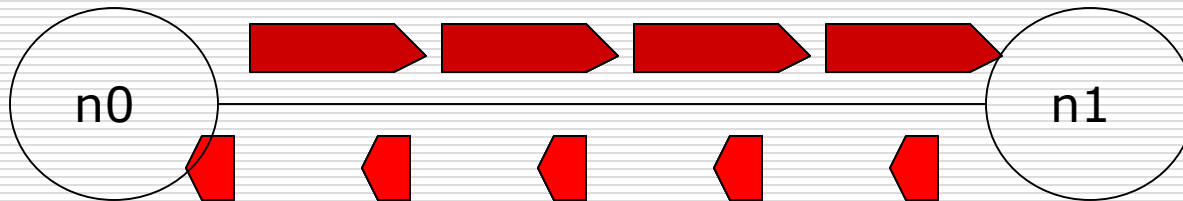
Simulation with ns-2

- ❑ Creating the event scheduler
 - ❑ Creating network: nodes, links & queue
 - ❑ Computing routes
 - ❑ Creating connection
 - ❑ Creating traffic

 - ❑ Inserting errors
 - ❑ Tracing

 - ❑ Wireless Support
-

Example Script



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

```
$ns duplex-link $n0 $n1 1.5Mb 10ms
    DropTail
set tcp [$ns create-connection TCP $n0 TCPSink
    $n1 0]
```

```
set ftp [new
    Application/FTP]
$ftp attach-agent $tcp
$ns at 0.2 "$ftp start"
$ns at 1.2 "exit"
$ns run
```

Creating Event Scheduler

- Create scheduler
 - set ns [new Simulator]
 - Schedule event
 - \$ns at <time> <event>
 - <event>: any legitimate ns/tcl commands
 - Start scheduler
 - \$ns run
-

Creating Network: Node, Link & Queue

□ Nodes

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

□ Links & Queuing

- \$ns duplex-link \$n0 \$n1 <bandwidth> <delay>
<queue_type>

where, <queue_type>: DropTail, RED, CBQ, FQ, SFQ, DRR

Creating Network: LAN

□ LAN

- `$ns make-lan <node_list> <bandwidth>
<delay> <ll_type> <ifq_type>
<mac_type> <channel_type>`

where, <ll_type>: LL

<ifq_type>: Queue/DropTail,

<mac_type>: MAC/802_3

<channel_type>: Channel

Computing routes

□ Unicast

- `$ns rproto <type>`
- `<type>`: Static, Session, DV, cost, multi-path

□ Multicast

- `$ns multicast` (right after [new Simulator])
 - `$ns mrtproto <type>`
 - `<type>`: CtrMcast, DM, ST, BST
-

Creating Connection: UDP

□ UDP

- `set udp [new Agent/UDP]`
 - `set null [new Agent/NULL]`
 - `$ns attach-agent $n0 $udp`
 - `$ns attach-agent $n1 $null`
 - `$ns connect $udp $null`
-

Creating Connection: TCP

□ TCP

- `set tcp [new Agent/TCP]`
 - `set tcpsink [new Agent/TCPSink]`
 - `$ns attach-agent $n0 $tcp`
 - `$ns attach-agent $n1 $tcpsink`
 - `$ns connect $tcp $tcpsink`
-

Creating Traffic: On Top of TCP

□ FTP

- set ftp [new Application/FTP]
- \$ftp attach-agent \$tcp
- \$ns at <time> “\$ftp start”

□ Telnet

- set telnet [new Application/Telnet]
 - \$telnet attach-agent \$tcp
-

Creating Traffic: On Top of UDP

□ CBR

- set src [new Application/Traffic/CBR]

□ Exponential or Pareto on-off

- set src [new Application/Traffic/Exponential]
 - set src [new Application/Traffic/Pareto]
-

Creating Traffic: Trace Driven

- Trace driven
 - set tfile [new Tracefile]
 - \$tfile filename <file>
 - set src [new Application/Traffic/Trace]
 - \$src attach-tracefile \$tfile

 - <file> :
 - Binary format
 - inter-packet time (msec) and packet size (byte)
-

Inserting Errors

□ Creating Error Module

- `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]`

□ Inserting Error Module

- `$ns lossmodel $loss_module $n0 $n1`
-

Tracing

□ Trace packets on all links

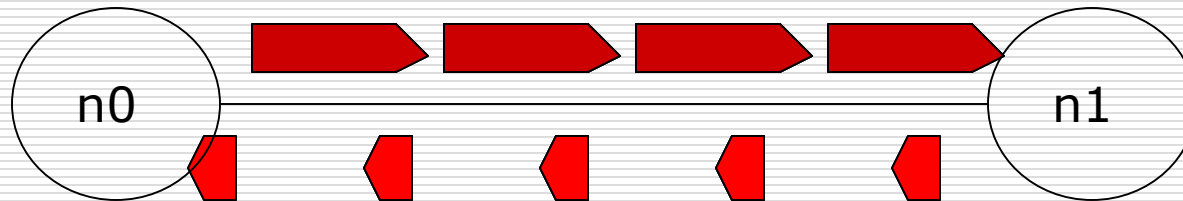
- `$ns trace-all [open test.out w]`

```
<event> <time> <from> <to> <pkt> <size>--<flowid> <src> <dst> <seqno> <aseqno>
+ 1 0 2 cbr 210 ----- 0 0.0 3.1 0 0
- 1 0 2 cbr 210 ----- 0 0.0 3.1 0 0
r 1.00234 0 2 cbr 210 ----- 0 0.0 3.1 0 0
```

□ Trace packets on all links in nam-1 format

- `$ns namtrace-all [open test.nam w]`
-

Example Script (again)

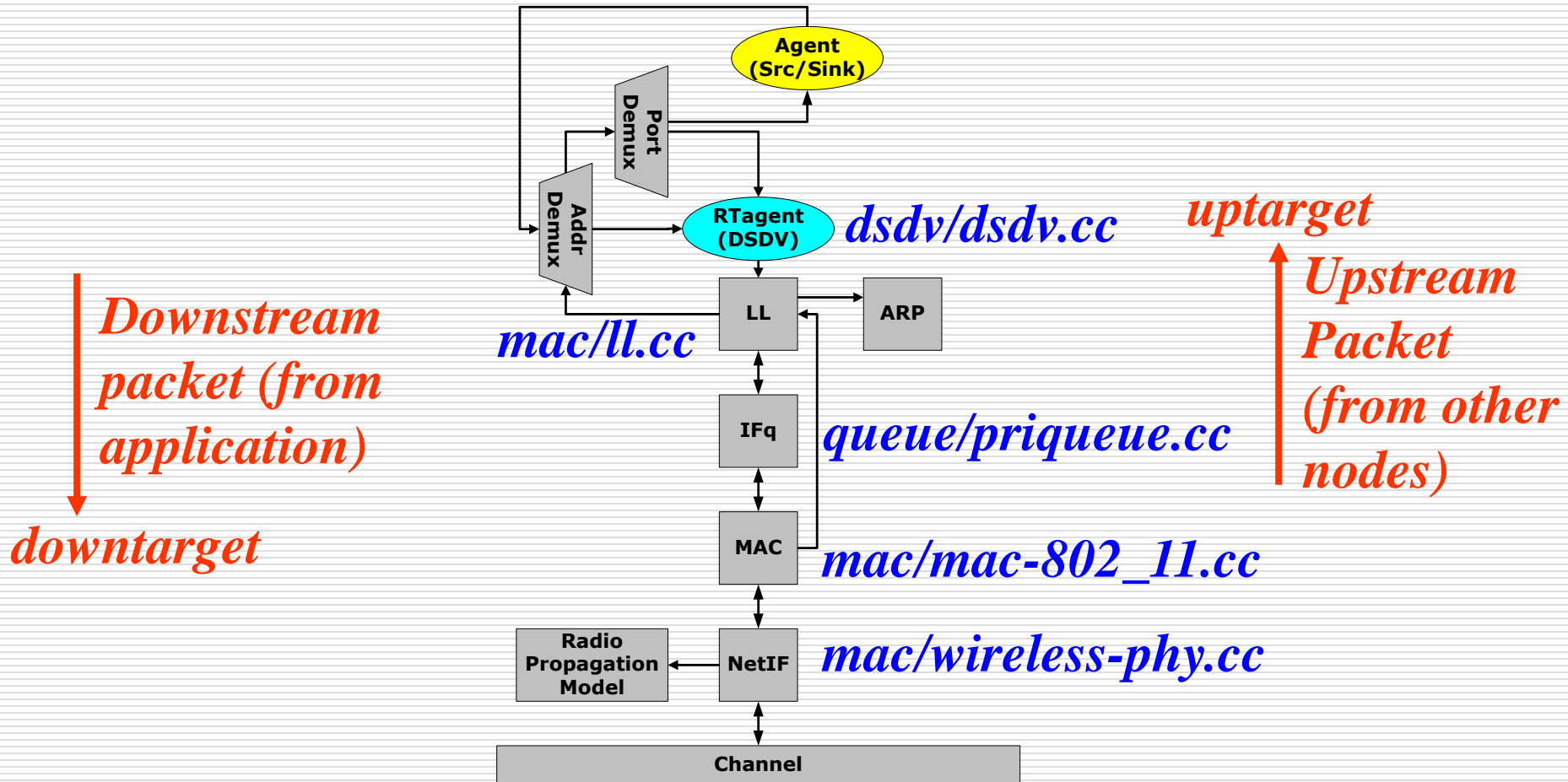


```
set ns [new Simulator]
set n0 [$ns node]
set n1 [$ns node]
```

```
$ns duplex-link $n0 $n1 1.5Mb 10ms
    DropTail
set tcp [$ns create-connection TCP $n0 TCPSink
    $n1 0]
```

```
set ftp [new
    Application/FTP]
$ftp attach-agent $tcp
$ns at 0.2 "$ftp start"
$ns at 1.2 "exit"
$ns run
```

Mobile Node Modules



Wireless Support: Movement

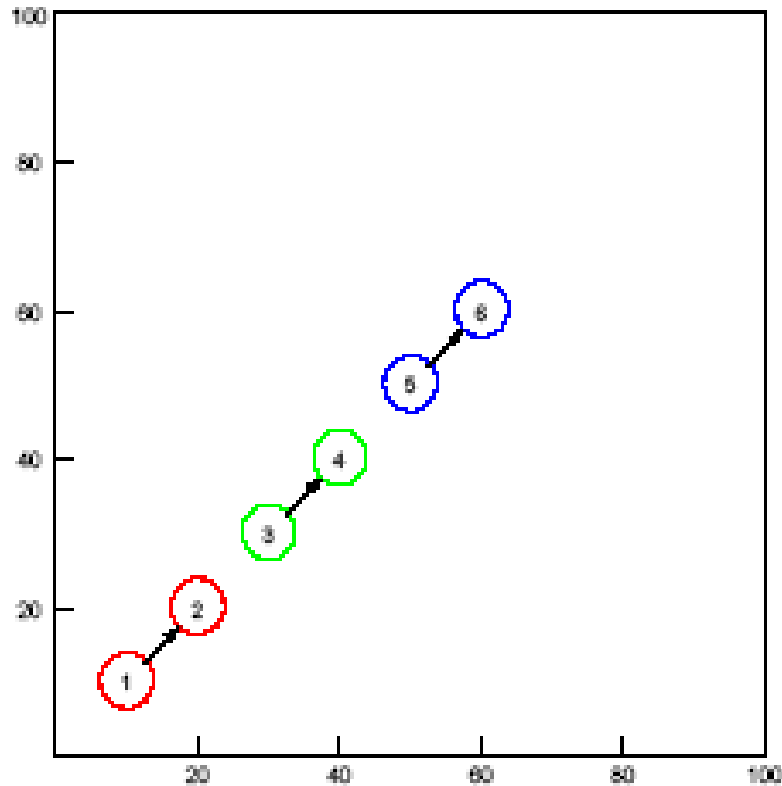
□ Specified

- \$ns at 1.0 “\$mnode setdest <x> <y> <speed>”

□ Random

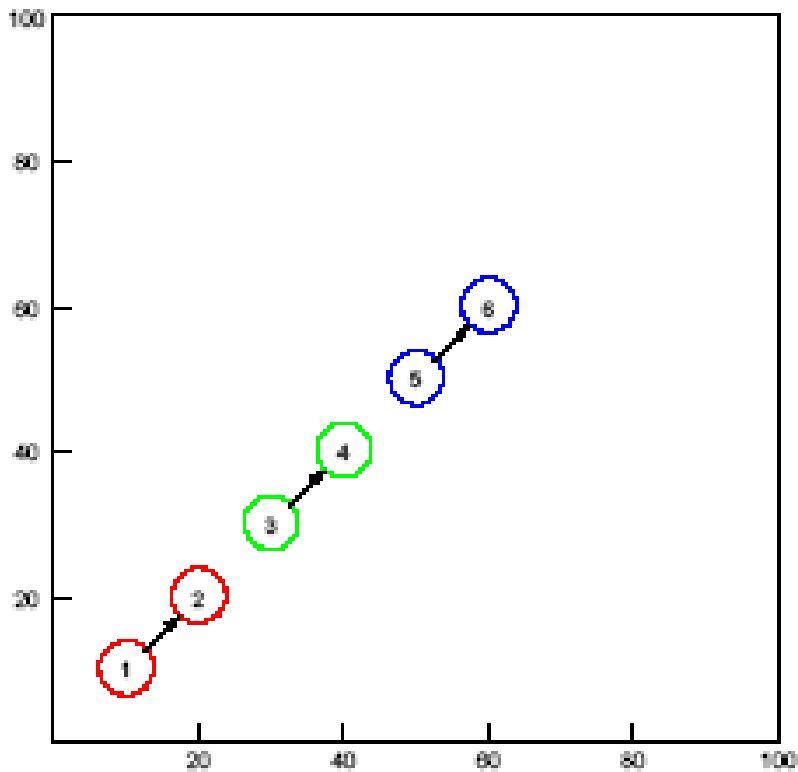
- \$ns at 1.0 “\$mnode start”
-

Mobile Network Simulation with ns-2



- 6 nodes
- Located (10,10), (20,20), ...
- CBR (Constant bit rate) traffic based on UDP 1→2, 3→4, 5→6

Mobile Network Simulation with ns-2



- Defining 6 nodes

```
set ns_ [new Simulator]  
set opt(nn) 6
```

```
set WT1 [$ns_ node $1]
```

...

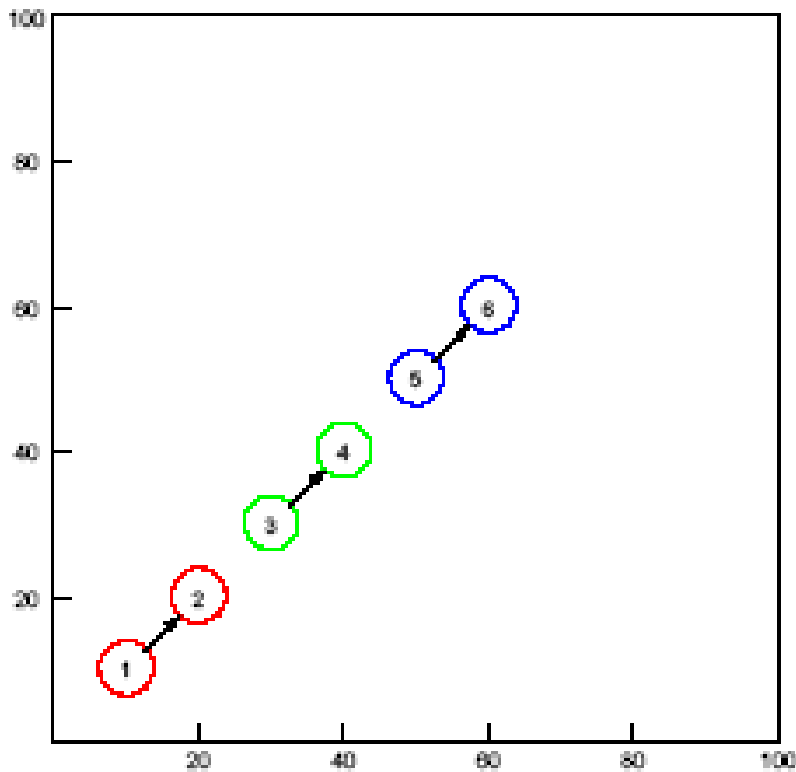
```
set WT6 [$ns_ node $6]
```

=>

```
for {set i 1} {$i<=$opt(nn)} {incr i} {  
    set WT($i) [$ns_ node $i]  
}
```

Mobile Network Simulation with ns-2

- Located (10,10), (20,20), ...



```
$WT1 set X_ 10
```

```
$WT1 set Y_ 10
```

```
$WT1 set Z_ 0.0
```

```
...
```

```
$WT6 set X_ 60
```

```
$WT6 set Y_ 60
```

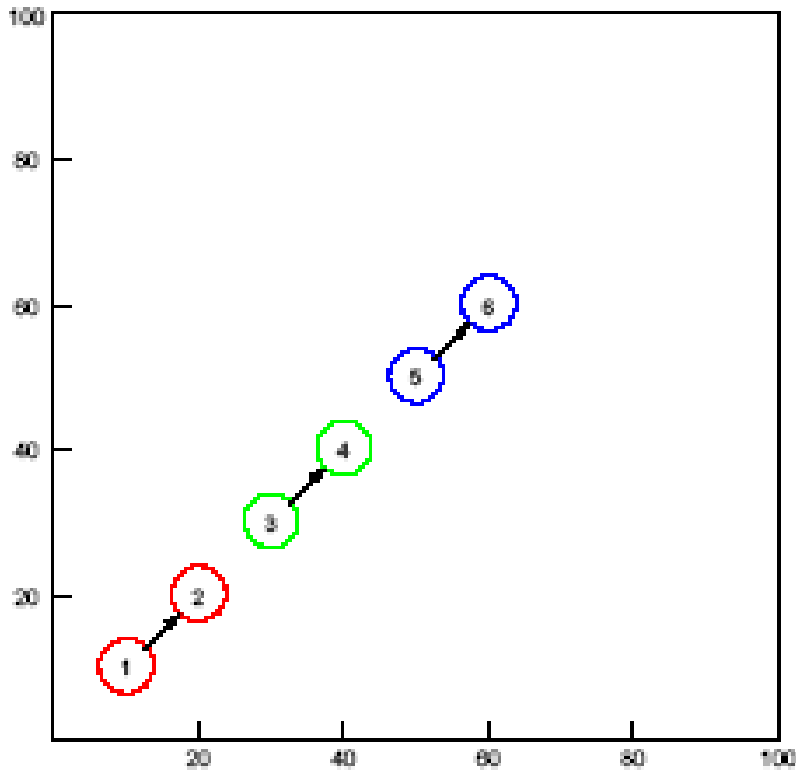
```
$WT6 set Z_ 0.0
```

```
=>
```

```
for {set i 1} {$i<=$opt(nn)} {incr i} {  
    $WT($i) set X_ [expr 10*$i]  
    $WT($i) set Y_ [expr 10*$i]  
    $WT($i) set Z_ 0.0
```

```
}
```


Mobile Network Simulation with ns-2



- CBR (Constant bit rate) traffic based on UDP 1→2, 3→4, 5→6

```
set udp1 [new Agent/UDP]
```

```
$ns_ attach-agent $WT1 $udp1
```

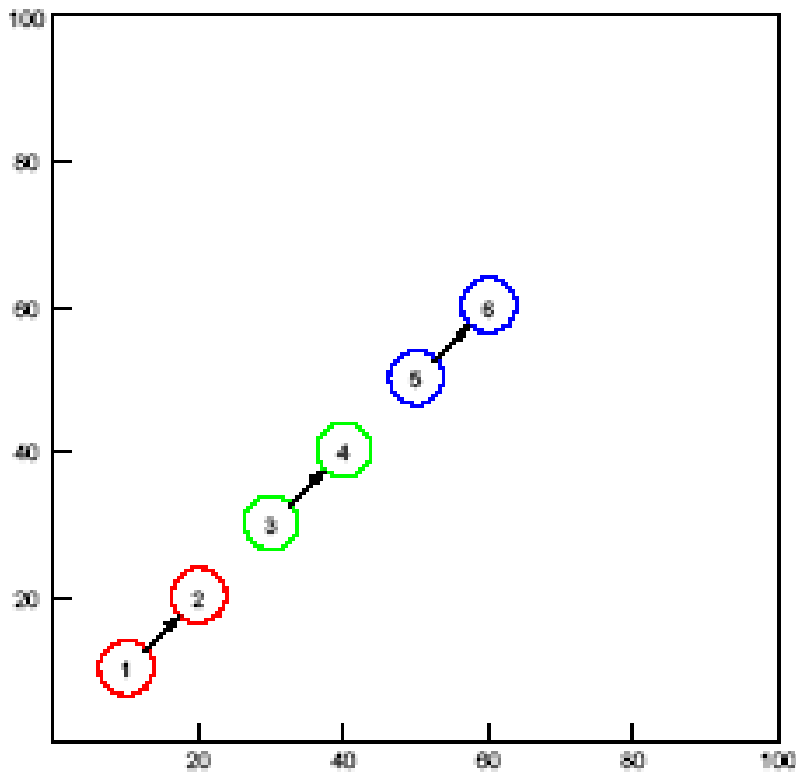
```
set sink1 [new Agent/Null]
```

```
$ns_ attach-agent $WT2 $sink1
```

```
$ns_ connect $udp1 $sink1
```

```
...
```

Mobile Network Simulation with ns-2



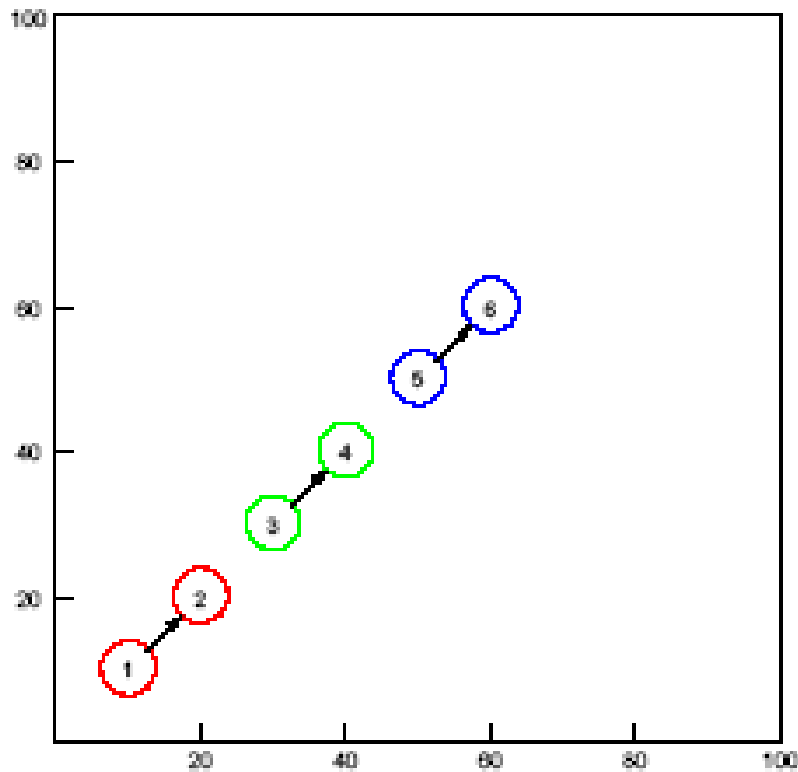
- CBR (Constant bit rate) traffic based on UDP 1→2, 3→4, 5→6

```
set cbr1 [new Application/Traffic/CBR]  
$cbr1 set packetSize_ 1000  
$cbr1 set interval_ 0.005  
$cbr1 attach-agent $udp1
```

```
$ns_ at 20.0 "$cbr1 start"  
$ns_ at 150.0 "$cbr1 stop"
```

...

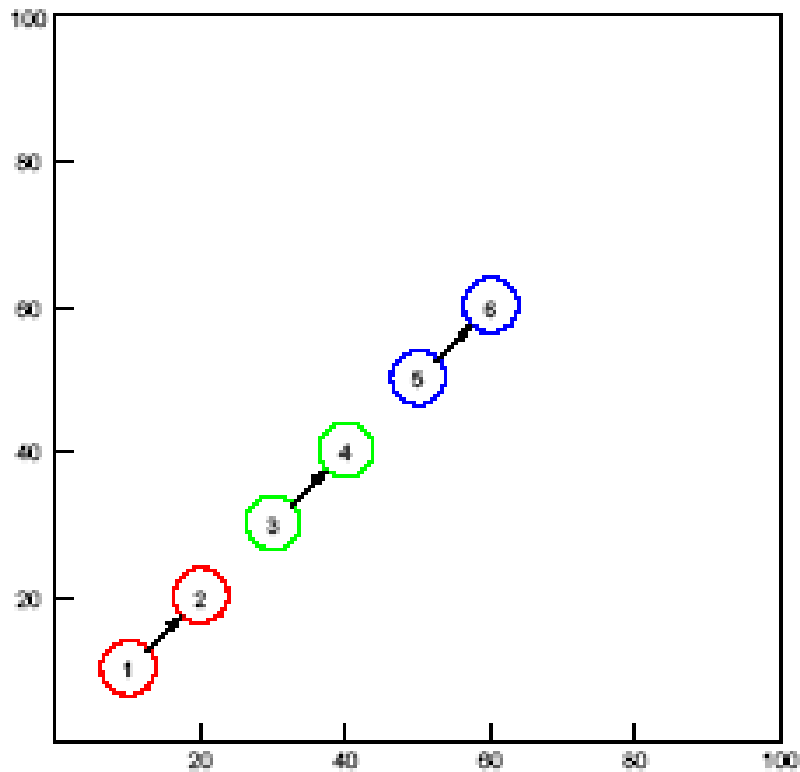
Mobile Network Simulation with ns-2



- CBR (Constant bit rate) traffic based on UDP 1→2, 3→4, 5→6

```
for {set i 1} {$i <= $opt(nn)} {incr i 2} {  
  set udp($i) [new Agent/UDP]  
  $ns_ attach-agent $WT($i) $udp($i)  
  
  set sink($i) [new Agent/Null]  
  $ns_ attach-agent $WT([expr $i+1]) $sink($i)  
  
  $ns_ connect $udp($i) $sink($i)
```

Mobile Network Simulation with ns-2



- CBR (Constant bit rate) traffic based on UDP 1→2, 3→4, 5→6

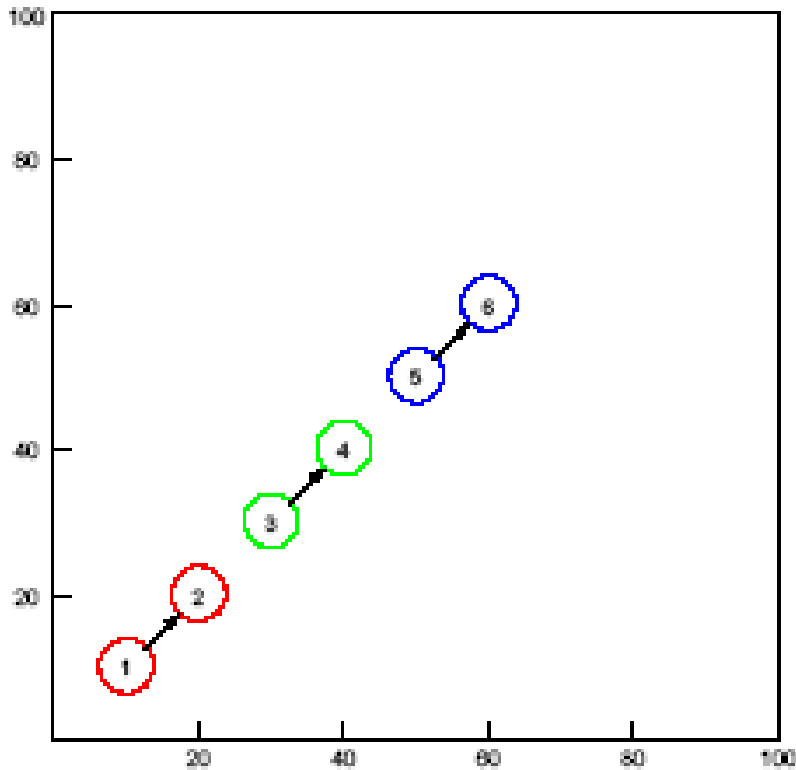
```
set cbr($i) [new Application/Traffic/CBR]  
$cbr($i) set packetSize_ 1000  
$cbr($i) set interval_ 0.005  
$cbr($i) attach-agent $udp($i)
```

```
$ns_ [expr 20.0*$i] "$cbr($i) start"  
$ns_ at $opt(stop) "$cbr($i) stop"
```

```
}
```

```
set opt(stop) 150
```


Mobile Network Simulation with ns-2



- Topology configuration

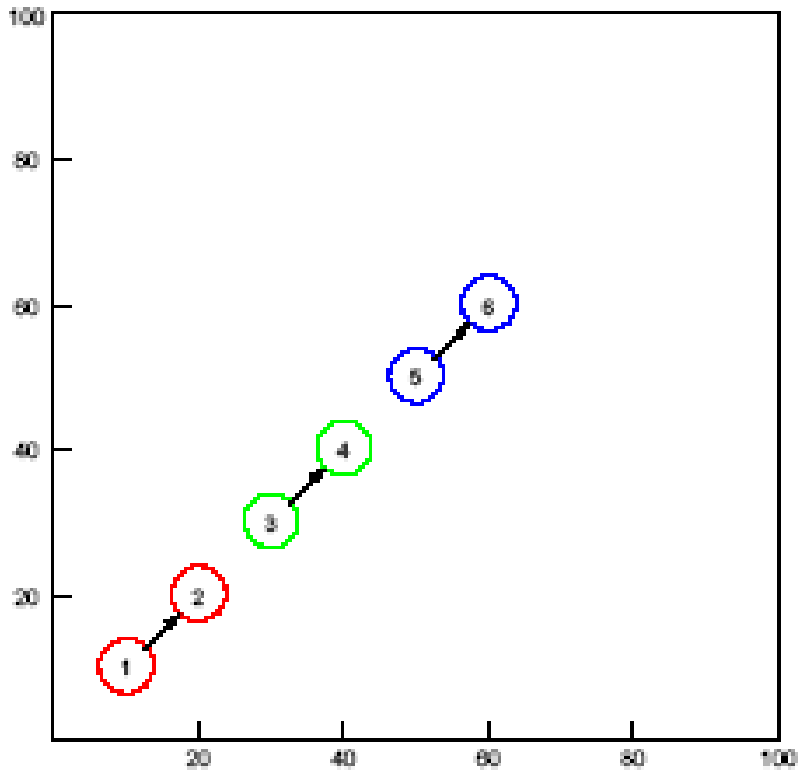
```
set topo [new Topography]  
$topo load_flatgrid $opt(x) $opt(y)
```

<i>set opt(x)</i>	<i>100</i>
<i>set opt(y)</i>	<i>100</i>

- GOD (General operations director)

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

Mobile Network Simulation with ns-2



- Node configuration (cont'd)

```
$ns_ node-config
```

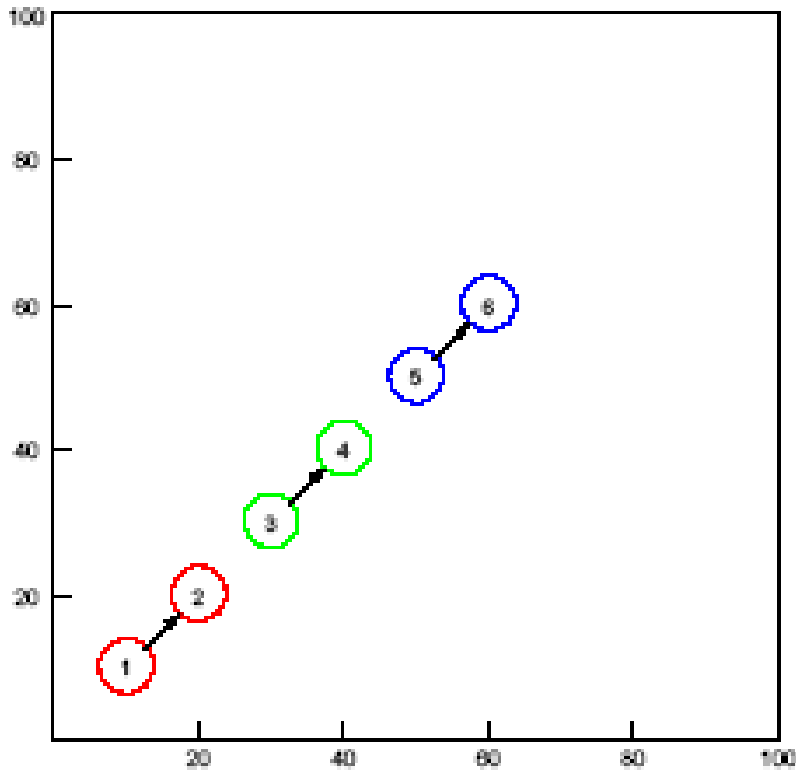
```
...
```

```
-agentTrace ON \  
-routerTrace OFF \  
-macTrace OFF \  
-movementTrace OFF
```

```
set tracefd [open $opt(tr) w]  
$ns_ trace-all $tracefd
```

```
set opt(tr) out.tr
```

Mobile Network Simulation with ns-2



- End procedure & Start ns2 simulation

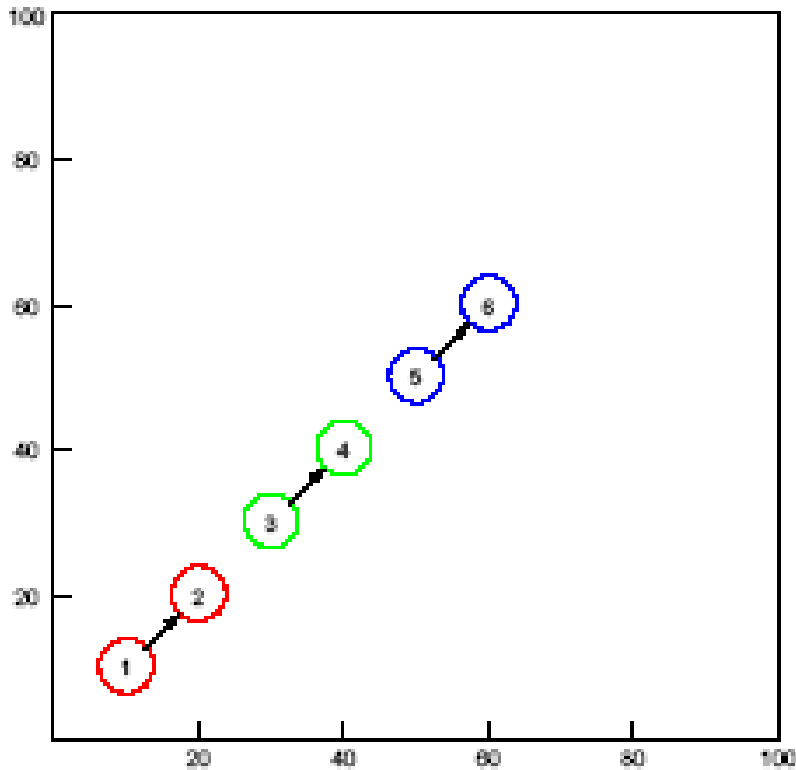
```
for {set i} {$i <= $opt(nn)} {incr i} {  
    $ns_ at $opt(stop).0000010 "$WT($i) reset";  
}
```

```
$ns_ at $opt(stop).2 "$ns_ halt"
```

```
puts "Starting Simulation..."
```

```
$ns_ run
```


Mobile Network Simulation with ns-2



- Adding finish procedure

```
$ns_ at $opt(stop).1 "finish"  
$ns_ at $opt(stop).2 "$ns_ halt"
```

```
proc finish {} {  
  for {set i 2} {$i <= 6} {incr i 2} {  
    exec rm -f out$i.xgr  
    exec awk -f fil$i.awk out.tr > out$i.xgr  
  }  
  exec xgraph out2.xgr out4.xgr out6.xgr &  
  puts "Finishing ns.."  
  exit 0  
}
```

Summarize the Results

- Ns2 manual, “Trace and Monitoring support,” Ch. 26

 - Linux commands to quickly get results

 - Gets two files (if specified)
 - out-test.tr - use awk script to summarize the results
 - nam-out-test.nam - visualization
-

Visualize the Simulation Runs (nam file)

□ nam-out-test.nam

- Nam: network animator

□ Try these

- nam nam-1.14/edu/C2-sliding-color.nam
 - nam nam-1.14/tcl/test/test-wireless-2.nam
-

Nodes are numbered
as 0, 1, 2 , ... inside ns2.
I.e., node WT1 is node 0.

Trace File Format

```
r 100.381997477 1_ AGT --- 82 cbr 1060 [13a 1 0 800] --  
----- [0:0 1:0 32 1] [32 0] 1 0
```

r:receive event,

1:node 1,

82:event(pkt) id,

1060:packet size,

13a(hex):expected duration of pkt transmission (not working),

1:sender mac id,

0:transmitter mac id,

800:pkt type IP (806 for ARP), **0:0**: sender address:port#

1:0: receiver address:port#, **32**: TTL

1: next hop address,

[32 0]: TCP sequence #, ack #

Summarize Trace File

- Using simple Linux commands

cat, grep, wc, |, >, >>, etc.

eg. Calculate **packet delivery ratio** from a trace file (out.tr)

```
cat out.tr | grep AGT | grep cbr | grep ^s | wc -l
```

```
cat out.tr | grep AGT | grep cbr | grep ^r | wc -l
```

```
cat out.tr | grep AGT | grep cbr | grep ^s | grep _0_ | wc -l
```


```
cat out.tr | grep AGT | grep cbr | grep ^r | grep _1_ | wc -l
```

- Simple programming

shell, awk, etc.

- Advanced programming

C/C++, Java, VB, etc.



Base 0

ns Tutorials

- ❑ NS website <http://www.isi.edu/nsnam/ns/>
 - ❑ NS Manual
http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf
 - ❑ Marc Greis's Tutorial
<http://www.isi.edu/nsnam/ns/tutorial>
 - ❑ <http://www.cs.virginia.edu/~cs757/slidespdf/cs757-ns2-tutorial-exercise1.pdf>
 - ❑ <http://nile.wpi.edu/NS/>
 - ❑ http://nesl.ee.ucla.edu/courses/ee206a/2002s/guest_presentations/GP02_Park_ns2.ppt
 - ❑ http://www.ece.ubc.ca/~elec565/ns2_tutorial.ppt
-

The Network Simulator - ns-2

Note: The project has migrated these web pages to a [wiki](#). This page can now be found [here](#).

Ns is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols (including satellite) networks.

Ns began as a variant of the [REAL network simulator](#) in 1989 and has evolved substantially over the past few years. In 1995 ns development was supported by [project](#) at LBL, Xerox PARC, UCB, and USC/ISI. Currently ns development is supported through DARPA with [SAMAN](#) and through NSF with [CONSERVATION](#) researchers including [ACIRI](#). Ns has always included substantial contributions from other researchers, including wireless code from the UCB Daedalus group and Sun Microsystems. For documentation on recent changes, see the version 2 [change log](#).

Read this first:

While we have considerable confidence in ns, ns is not a polished and finished product, but the result of an on-going effort of research and development. Bugs are still being discovered and corrected. Users of ns are responsible for verifying for themselves that their simulations are not invalidated by bugs. We encourage you to do this by significantly expanding and automating the [validation tests and demos](#).

Similarly, users are responsible for verifying for themselves that their simulations are not invalidated because the model implemented in the simulator does not match their expectations. The ongoing [Ns Manual](#) should help in this process.

Links to help getting started

- **Getting ns and avoiding problems**
 - [Download and Build ns](#)
 - [Installation Problems and Bug Fixes](#)
 - [Validation Tests and Demos](#)
 - [Copyright statement](#)
 - [Bug reporting](#)
- **Documentation:**
 - **core documentation:**
 - [ns frequently asked questions](#)
 - limitations: [limitations and assumptions of ns](#), [advice on running simulations](#)