Department of Communications and Networking
Helsinki University of Technology

S-72.3235 Introduction to Network Simulator 2

Shekar Nethi

Practical Arrangements

Arrangements:

- Course Schedule

  13.10, 14.10, 15.10

- We have 15 Ubuntu Terminals here, Ns2 is installed on a common shared directory,

  - ssh **user_account@lunni.hut.fi**
  - source /**p/edu/s-72.3235/shekar/setpath.csh**
  - ns

- I will update the material on the noppa portal

- Email shekar@cc.hut.fi.

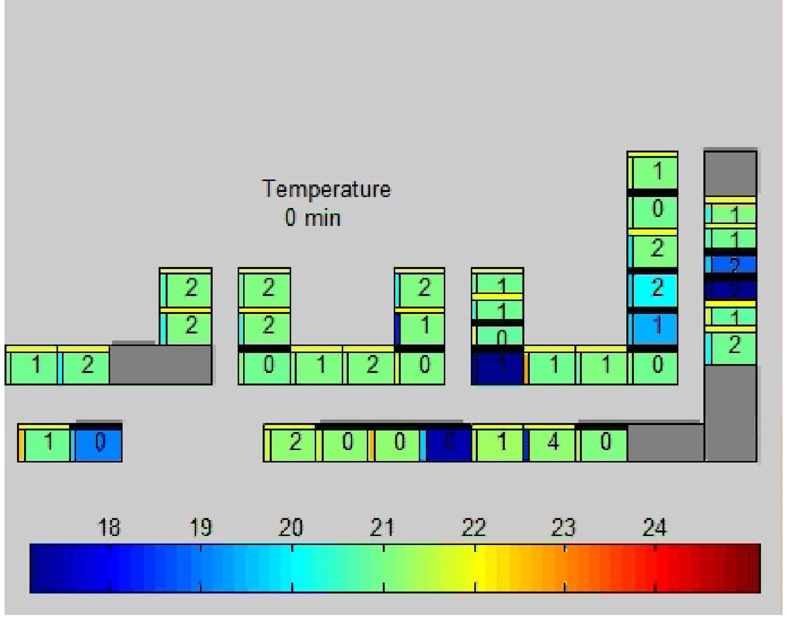
Plan:

1. **Ns2 Basics & Visualization tool # Try installation at least once!!**
2. Basic Scenario's building & RTS/CTS simulations
3. Our small project (New routing protocol or modify existing channel model)

Why Ns2

- Powerful simulator, open source, active research
- 2 out of every 3 papers in Wireless networks is done using Ns2
- Support for real time applications
- Industrial applications. Building Automation, Factory warehouse

  - http://www.control.tkk.fi/dev/MoCoNet/



My experience

Project: A multipath DSR protocol

Task: Implementation

Step 1: I ran to the library

1. Ns2 Manual
2. The complete reference C++
3. Introduction to TCL scripting
4. Perl, AWK, BASH
5. DSR source code (15-20 files and more than 1000 lines of code)

Fundamental concepts of programming

Ns2 Basics

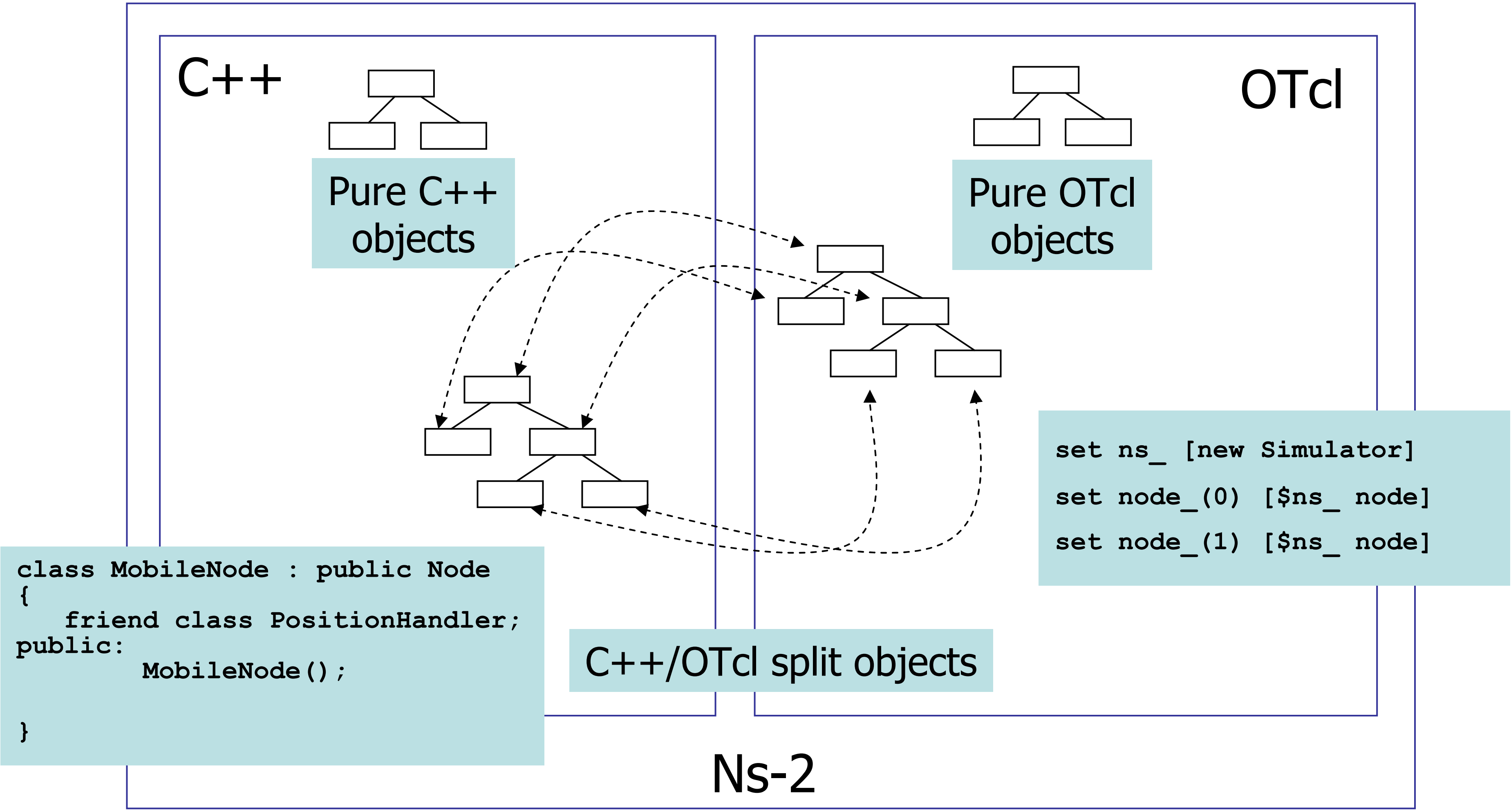
Shekar Nethi.

Network Simulator 2 (Ns2)

- A de-factor standard for testing wireless networks.
- Developed at the University of California by LBL, Xerox PARAC, UCB and USC/ISI
- VINT project supported be DARPA.
- Its an discrete event driven simulation
- Models Available
  - Transport layer TCP(reno, tahoe, vegas, sack), UDP,
  - MAC(802.11, 802.3, TDMA, 802.15.4)
  - Ad-hoc Routing (DSDV, DSR, AODV, TORA)
  - Sensor Network (diffusion, gaf)
  - Radio channel models (Freespace, TwoRayGround, Shadow fading)
  - Multicast protocols, Satellite protocols, and lots more

Examples of some contribution from research community WiMAX, Blueware, Ricean and Rayleigh fading

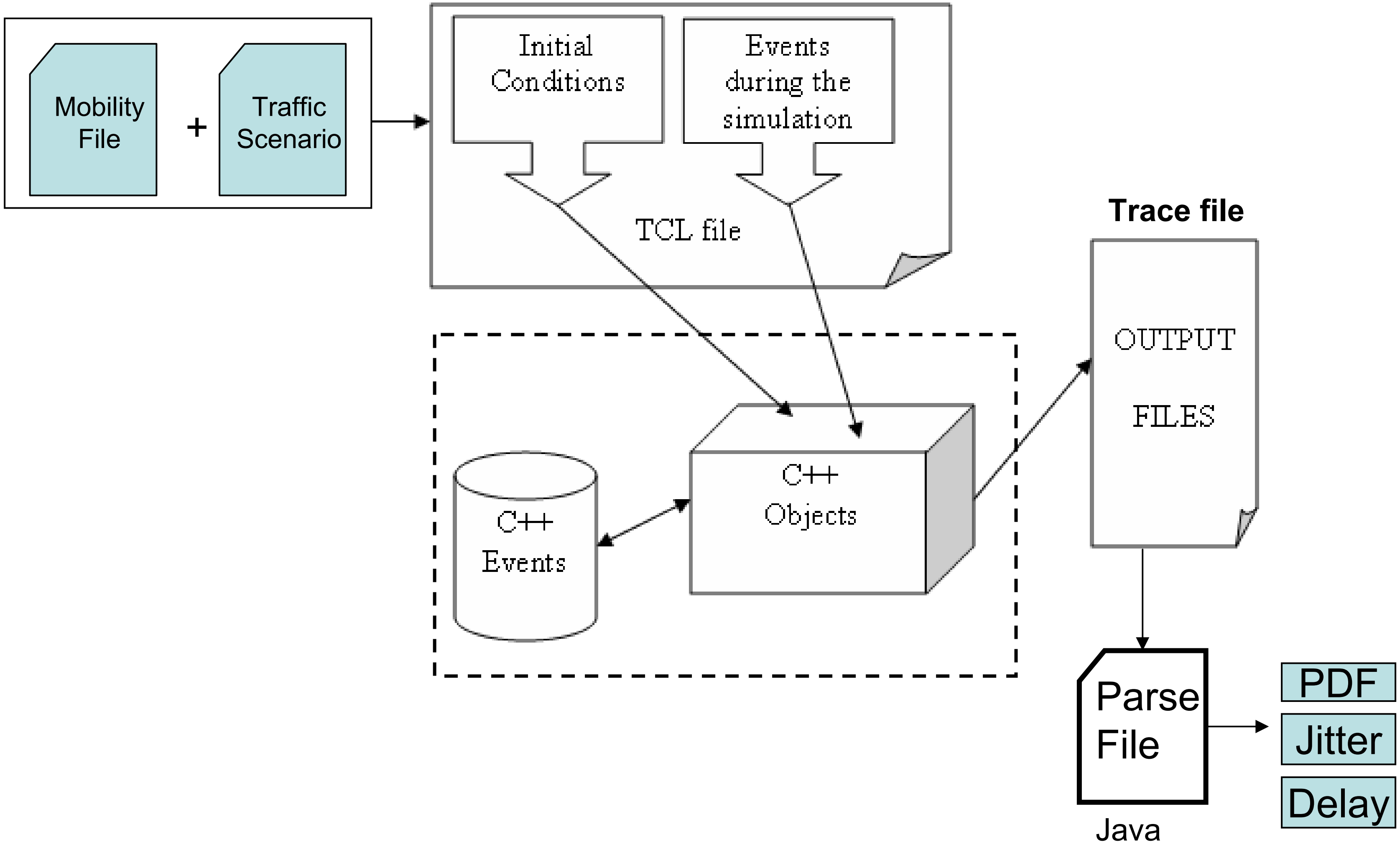
Two Languages: OTcl and C++



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

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

}
```

Remember this



Wireless Simulation (Example: Wireless.tcl)

```
###############################################################################
                                   wireless.tcl
###############################################################################

set val(chan)           Channel/WirelessChannel    ;# Channel Type
set val(prop)           Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)          Phy/WirelessPhy            ;# network interface type
set val(mac)            Mac/802_11                 ;# MAC type
set val(ifq)            Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)             LL                         ;# link layer type
set val(ant)            Antenna/OmniAntenna        ;# antenna model
set val(ifqlen)         50                         ;# max packet in ifq
set val(x)              800                        ;# x range in meters
set val(y)              600                        ;# y range in meters
set val(rp)             AODV                       ;# routing protocol
set val(nn)             4                          ;# number of mobile nodes
set val(stime)          150.0                      ;# simulation time 100 hours
set val(tr)             Simple.tr                  ;# trace file
set val(nm)             Simple.nam                 ;# nam file
```

Set Global Variables

```
set ns_ [new Simulator];            #initiate Network simulator
set tracefd [open  $val(tr) w]      #Define trace file
$ns_ trace-all $tracefd

set namtrace [open $val(nm) w]      #Define nam file
$ns_ namtrace-all-wireless
$namtrace $opt(x) $opt(y)

$ns_ use-newtrace                   #Use new trace format

set topo [new Topography]           #Set new topography
$topo load_flatgrid $val(x) $val(y)

create-god $val(nn)                  #Create Object God
```

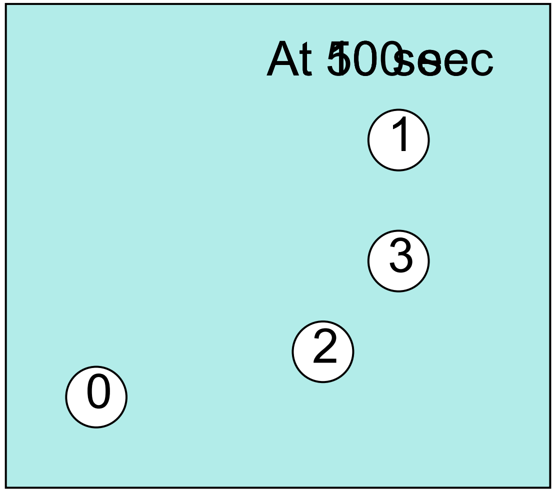
Node Configuration

```
# Configure wireless nodes
 $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) \
                                            -channelType $val(chan) \
                                            -topoInstance $topo \
                                            -agentTrace ON \
                                            -routerTrace ON \
                                            -macTrace OFF \
                                            -movementTrace OFF
```

```
for {set i 0} {$i < $val(nn) } {incr i} {
        set node_($i) [$ns_ node ]
        $node_($i) random-motion 0 ;# disable random motion
         #ns_ start                            # if random-motion 1
}
```

Configuring Movement

```
########### Configure Initial Position ##########


$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
$node_(2) set X_ 25.0
$node_(2) set Y_ 332.0
$node_(2) set Z_ 0.0
$node_(3) set X_ 290.0
$node_(3) set Y_ 385.0
$node_(3) set Z_ 0.0
```



```
############  Create Movement ####################
# Node_(1) starts to move towards node_(0)
$ns_ at 10.0 "$node_(1) setdest 25.0 20.0 15.0"
$ns_ at 50.0 "$node_(0) setdest 320.0 3.0 10.0"
# Node_(1) then starts to move away from node_(0)
$ns_ at 100.0 "$node_(1) setdest 490.0 480.0 15.0"
```

Generate UDP traffic

```
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(19) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.2500
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 1000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 0.785882446815208 "$cbr_(0) start"
```

Final formalities

```
for {set i 0} {$i < $val(nn) } {incr i} {
$ns_ at 150.0 "$node_($i) reset";
}

#Define stop time
$ns_ at 150.0001 "stop"
$ns_ at 150.0002 "puts \"NS EXITING...\" ;
$ns_ halt" proc stop {} { global ns_
tracefd close $tracefd }

# start the simulation
puts "Starting Simulation..."
$ns_ run
```

Topology and traffic generation

Setdest:

Change directory "ns-2.28/indep-utils/cmu-scen-gen/setdest"

```
Usage:
./setdest -v 2 -n $numnodes -s $speedtype -m $minspeed -M $maxspeed
 -t $simtime -P $pausetype -p $pause -x $maxx -y $maxy >> position.txt

#Add the following lines in the tcl script
set opt(mobility)  "position.txt”
source $opt(mobility)
```

$speedtype (1 for uniform; 2 for normal)

$pausetype (1 for constant; 2 for uniform)

Cbrgen.tcl:

Change directory "indep-utils/cmu-scen-gen/cbrgen.tcl"

```
Usage:
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections]
[-rate cbrrate] >> File.txt

#Add the following lines in the tcl script
set opt(traff) "traffic.txt”
source $opt(traff)
```

Topology and traffic generation

Setdest:
Change directory "ns-2.28/indep-utils/cmu-scen-gen/setdest"

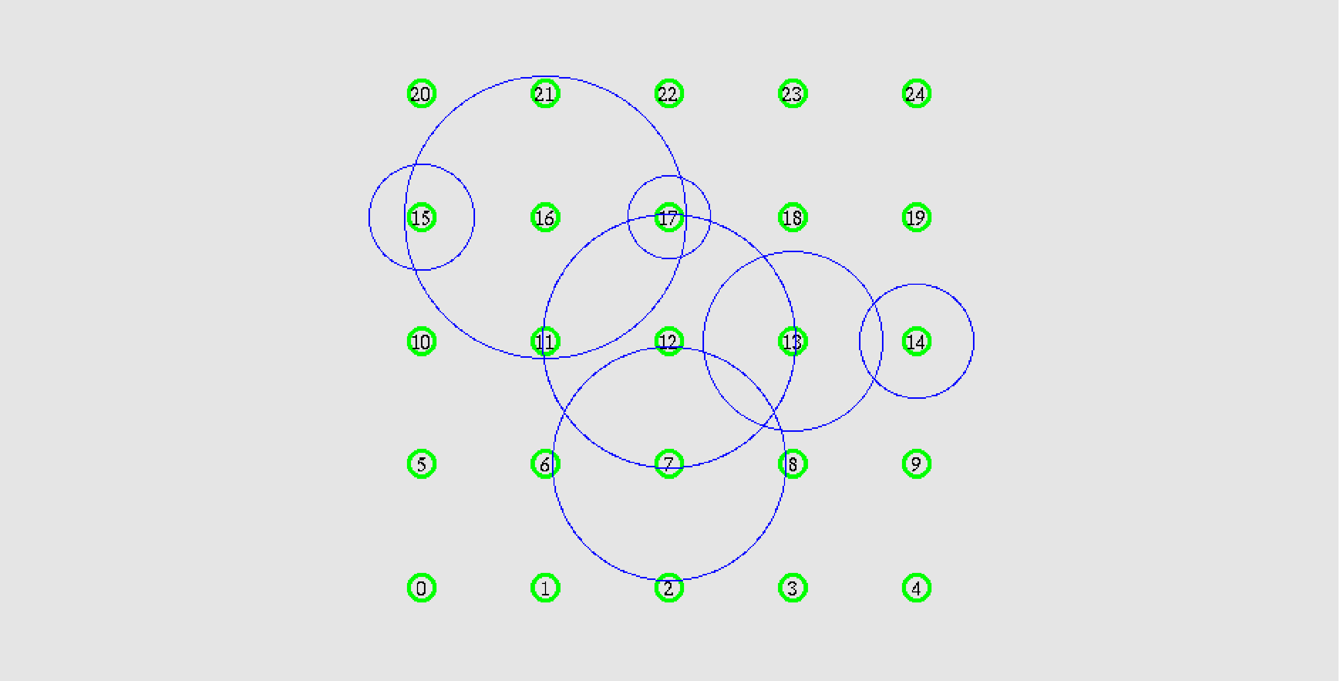
```
Usage:
./setdest -v 2 -n $numnodes -s $speedtype -m $minspeed -M $maxspeed
 -t $simtime -P $pausetype -p $pause -x $maxx -y $maxy >> position.txt

#Add the following lines in the tcl script
set opt(mobility)  "position.txt”
source $opt(mobility)
```

$speedtype (1 for uniform; 2 for normal)
$pausetype (1 for constant; 2 for uniform)

Cbrgen.tcl:
Change directory "indep-utils/cmu-scen-gen/cbrgen.tcl"

```
Usage:
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections]
[-rate cbrrate] >> File.txt

#Add the following lines in the tcl script
set opt(traff) "traffic.txt”
source $opt(traff)
```

Network animation (NAM)

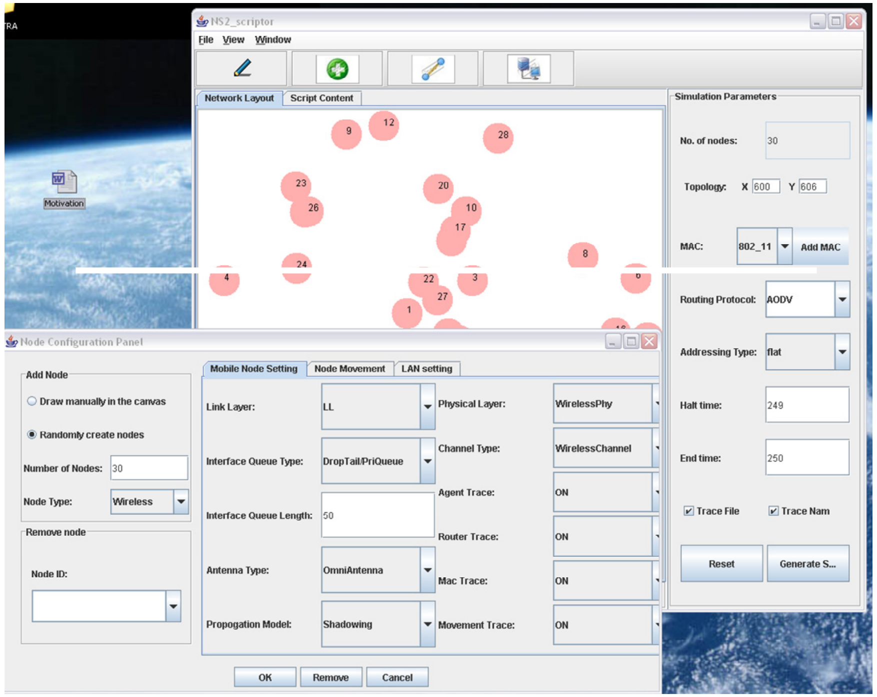


Usage:
nam xxxxxx.nam

Network Simulator (Ns2) Visualization Tool:

Bachelor’s Thesis: Huang Chen

- Simple java based GUI:



Simulation Parameters

No. of nodes: 30

Topology: X 600 Y 606

MAC: 802_11

Routing Protocol: AODV

Addressing Type: flat

Halt time: 249

End time: 250

☑ Trace File ☑ Trace Nam

Add Node

○ Draw manually in the canvas

● Randomly create nodes

Number of Nodes: 30

Node Type: Wireless

Remove node

Node ID:

Link Layer: LL

Interface Queue Type: DropTail/PriQueue

Interface Queue Length: 50

Antenna Type: OmniAntenna

Propogation Model: Shadowing

Physical Layer: WirelessPhy

Channel Type: WirelessChannel

Agent Trace: ON

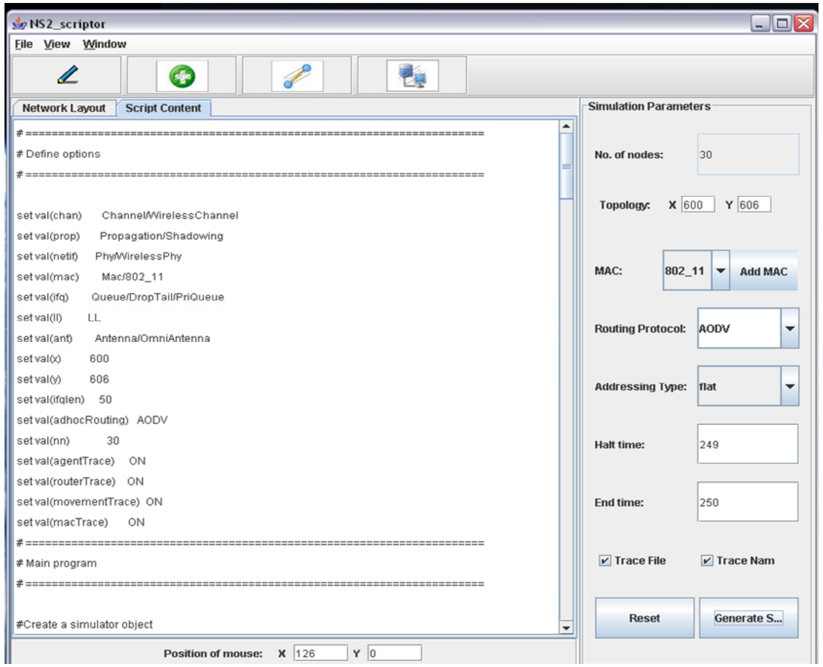
Router Trace: ON

Mac Trace: ON

Movement Trace: ON

Drag and drop

TCL Script



```
#========================================================================
# Define options
#========================================================================

set val(chan)      Channel/WirelessChannel
set val(prop)      Propagation/Shadowing
set val(netif)     Phy/WirelessPhy
set val(mac)       Mac/802_11
set val(ifq)       Queue/DropTail/PriQueue
set val(ll)        LL
set val(ant)       Antenna/OmniAntenna
set val(x)         600
set val(y)         606
set val(ifqlen)    50
set val(adhocRouting)  AODV
set val(nn)            30
set val(agentTrace)    ON
set val(routerTrace)   ON
set val(movementTrace) ON
set val(macTrace)      ON
#========================================================================
# Main program
#========================================================================

#Create a simulator object
```

Simulation Parameters

No. of nodes: 30

Topology: X 600 Y 606

MAC: 802_11

Routing Protocol: AODV

Addressing Type: flat

Halt time: 249

End time: 250

☑ Trace File ☑ Trace Nam

Position of mouse: X 126 Y 0

Installation

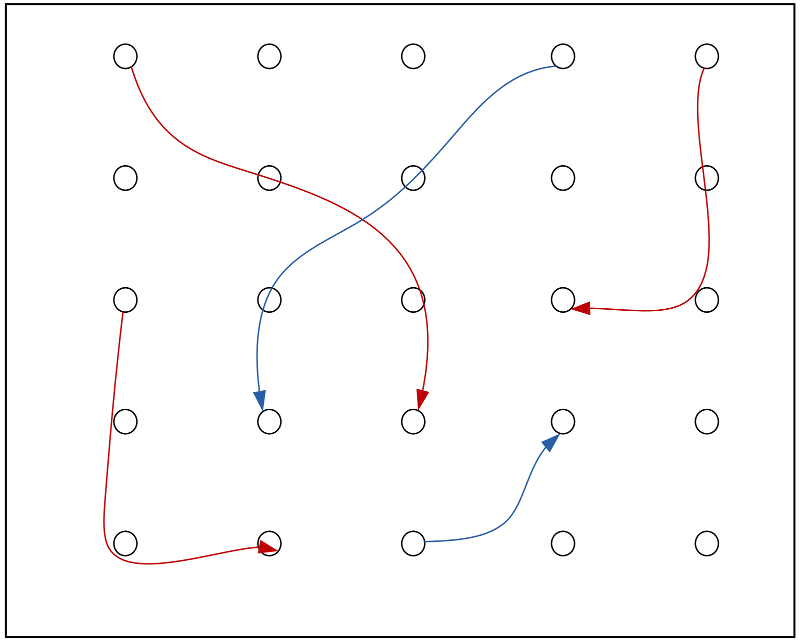
Shekar Nethi.

```
$ssh user_account@bell.ee.hut.fi

$mkdir Ns2

$scp -r useraccount@vipunen.hut.fi:/p/edu/s-72.3235/ns-allinone-2.31.tar.gz Ns2/

$Cd Ns2/

$gzip -d ns-allinone-2.31.tar.gz

$tar -xvf ns-allinone-2.31.tar.gz

$cd ns-allinone-2.31

$./install
```

Alternative servers

http://www.ee.hut.fi/unix/hardware.shtml.en

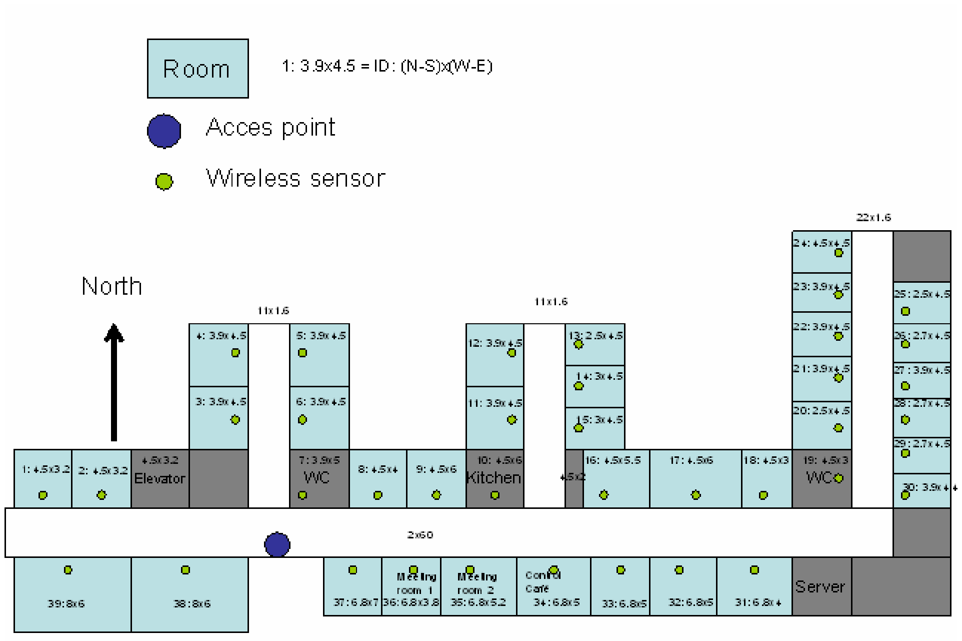
Grid Topology



5 X 5 Grid Network

Simulation time	100 sec.
No. of nodes	25 (grid),
macType	IEEE 802.11
ifqLen variable	(50 default)
antType	Antenna/OmniAntenna
propType	Propagation/TwoRayGround
channel	Channel/WirelessChannel
Traffic type	CBR
CBR generate rate	(4pkt/sec)/(8pkt/sec)/(16pkt/sec)
No. of CBR links	5
Packet size	512 bytes

Building automation model



Simulation time	250 sec.
No. of nodes	25,
macType	IEEE 802.15.4/802.11
ifqLen variable	(50 default)
antType	Antenna/OmniAntenna
propType	Propagation/Shadowing
channel	Channel/WirelessChannel
Traffic type	CBR
CBR generate rate	(1pkt/sec)
No. of CBR links	10
Packet size	54 bytes

Results

	Packet delivery ratio	Normalized Routing Overhead	Delay
Grid Topology			
CBR -4pkts/sec			
CBR -10pkts/sec			
CBR -16pkts/sec			
Building Automation			
Channel model: FreeSpace MacTYPE: 802.11			
Channel model: Shadowing MacTYPE: 802.11			
Channel model: FreeSpace MacTYPE: 802.15.4			
Channel model: Shadowing MacTYPE: 802.15.4			