CN Lab (17CSL57)

Exp 04: ESS in Wireless LAN

Dr. Ram P Rustagi
Dept of CSE, KSIT
KRP-KSGI
rprustagi@ksit.edu.in

Ex04 Resources

- References:
  - **https://www.github.com/rprustagi/VTU-CNLab**
  - Local area networks
    - https://www.isi.edu/nsnam/ns/doc/node169.html
  - NS2 and wireless nodes
    - https://www.isi.edu/nsnam/ns/tutorial/nsscript5.html
    - https://www.nsnam.org/docs/release/3.8/tutorial/tutorial_27.html#Building-a-Wireless-Network-Topology
    - github.com/rprustagi/VTU-CNLab/Exp04/
    - http://intronetworks.cs.luc.edu/current/html/ns2.html#wireless-simulation

Lab04 Program

- Program 04
  - Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets

Wireless Simulation

- No physical links
- Configuration work is to setup
  - Nodes, traffic, and wireless behaviour
  - Nodes are responsible for queuing (and not links)
- Wireless specific attributes
  - Antenna type, radio propagation
- Wired links: Doesn't play direct role
  - Need to define propagation delay and bandwidth
- Wireless links: distance plays significant part
  - Signal strength $\propto 1/d^2$
  - Bandwidth is built into wireless model
  - Attribute `datarate_` can be set for Mac/802_11
    - Default is 1Mb (NS2.35)

Wireless Simulation

- Routing protocol needs to be defined
  - e.g. DSDV for Adhoc routing
  - required when a nodes out from a range of current node to another node.
  - Protocol choice depends upon speed of mover node
- Max range of node is defined by power level
  - Attribute `txPower` of `node-config` parameter
  - Value 0.28183815 corresponds to 250m
- The simulation has number of attributes to configure
  - common convention - make attribute of 1 object `opt`
  - opt(channel), opt(mac), opt(x), opt(y), …

Creating Wireless Scenario

- Mobile node components in ns2
  - Link Layer (LL),
  - Interface Queue (IfQ),
  - MAC layer,
  - Wireless channel
  - Radio parameters
    - antenna,
    - radio-propagation model,
    - type of ad-hoc routing

Wireless/Radio Parameters

```
set opt(chan)   Channel/WirelessChannel
  # physical terrestrial wireless medium
set opt(prop)   Propagation/TwoRayGround
  # radio propagation model
```

TwoRayGround take ground reflection

power level ∝ $1/d^4$

Freespace model: power level ∝ 1/d2

```
set opt(netif)    Phy/WirelessPhy
  # wireless node interface to network
  # for satellite, it is Phy/Sat
set opt(mac)      Mac/802_11
  # other values: CSMA/CA, Aloha, Satellite
set opt(nn)       3  # number of wireless nodes
```

Wireless/Radio Parameters

```
set opt(ifq)    Queue/DropTail/PriQueue
  # queuing behaviour of each node
set opt(ifqlen) 50
  # queue length, like queue-limit for wired links
set opt(ant)    Antenna/OmniAntenna
  # Standard OmniAntenna,
  # other Parabolic disk, waveguide : cantennas
set opt(ll)     LL
  # defines behaviour of ARP on link layer
set opt(x)      300
set opt(y)      300
  # defines topology dimension in meters.
  # distance √(200^2+150^2)=250  defines range
```

Topology Creation

- Create a topology object that keeps track of movements of mobilenodes
  - `set topo     [new Topography]`
- Create the topography object with x and y co-ordinates
  - `$topo load_flatgrid 500 500`
- Create the object God (General Operations Director)
  - store global information about the state of the environment,
  - tracks of nodes, node-to-node reachability
    - parameter required: number of wireless nodes
  - `create-god $val(nn)`
- Trace file creation for wireless animation
  - `set namtr [open wireless.nam w]`
  - `$ns namtrace-all-wireless $namtr $opt(x) $opt(y)`

Config API to Create Wireless Nodes

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

Topology

- Provide initial (X,Y) co-ordinates for 3 nodes:
  - node_(0), node_(1) and node_(2)

```
set node_(0) [$ns node]
set node_(1) [$ns node]
set node_(2) [$ns node]

$node_(0) set X_ 10.0
$node_(0) set Y_ 20.0

$node_(1) set X_ 300.0
$node_(1) set Y_ 400.0

$node_(2) set X_ 100.0
$node_(2) set Y_ 450.0
```

Node movements

- Define some node movements,

```
# $node setdest <dst_x> <dst_y> <speed>
$ns at 10.0 "$node(0) setdest 20.0 20.0 1.0”
$ns at 20.0 "$node(1) setdest 50.0 40.0 5.0”
:
# define node movement as per requirement
```

- Define initial size (for nam to display these nodes)

```
$ns initial_node_pos $node(0) 20
$ns initial_node_pos $node(1) 20
```

- Invocation

```
ns wireless.tcl
nam wireless.nam
```

Wireless Trace File Format

- First field: r - received, s- sent, f- forward, D:dropped
- 2nd field: time of event occurrence
- 3rd field: node number
- 4th field: trace name e.g.
  - AGT: Application, RTR: Routing, MAC: Link kayer,
  - IFQ: interface priority queue
- 5th field : flags (generally dashes)
- 6th field: Global unique seq number of a packet
- 7th field: traffic type: CBR, TCP, message, Ack
- 8th field: pkt size in bytes
- 9th/11th field: Mac and Routing Layer separated by dash
  - e.g. src and destination

Wireless Trace File Format

- 9th field : [a b c d]
  - a: packet duration in mac layer header
  - b: mac address of destination
  - c: mac address of source
  - d: mac type of pkt body
- 10th field: dashes
- 11th field: [a b c d]
  - a: source node : port number
  - b: dstn node (-1 means broadcast): port number
  - c: IP hdr TTL
  - d: ip address of next hop (0 means broadcast or node 0)

–

awk Script

```
BEGIN {
  count1=0; count2=0
  pack1=0; pack2=0
  time1=0; time2=0
}
{
  if($1=="r" && $3=="_1_" && $4==“AGT"){
    count1++
    pack1 = pack1+$8
    time1 = $2
   }
  if($1=="r" && $3=="_2_" && $4==“AGT") {
    count2++
    pack2 = pack2+$8
    time2 = $2
 }
}
```

Awk processing script

```
END{
  printf("Thruput from n0 to n1: %f Mbps\n”,
    ((count1*pack1*8)/(time1*1000000)));
  printf("Thruput from n1 to n2: %f Mbps\n",
   ((count2*pack2*8)/(time2*1000000)));
 }
```

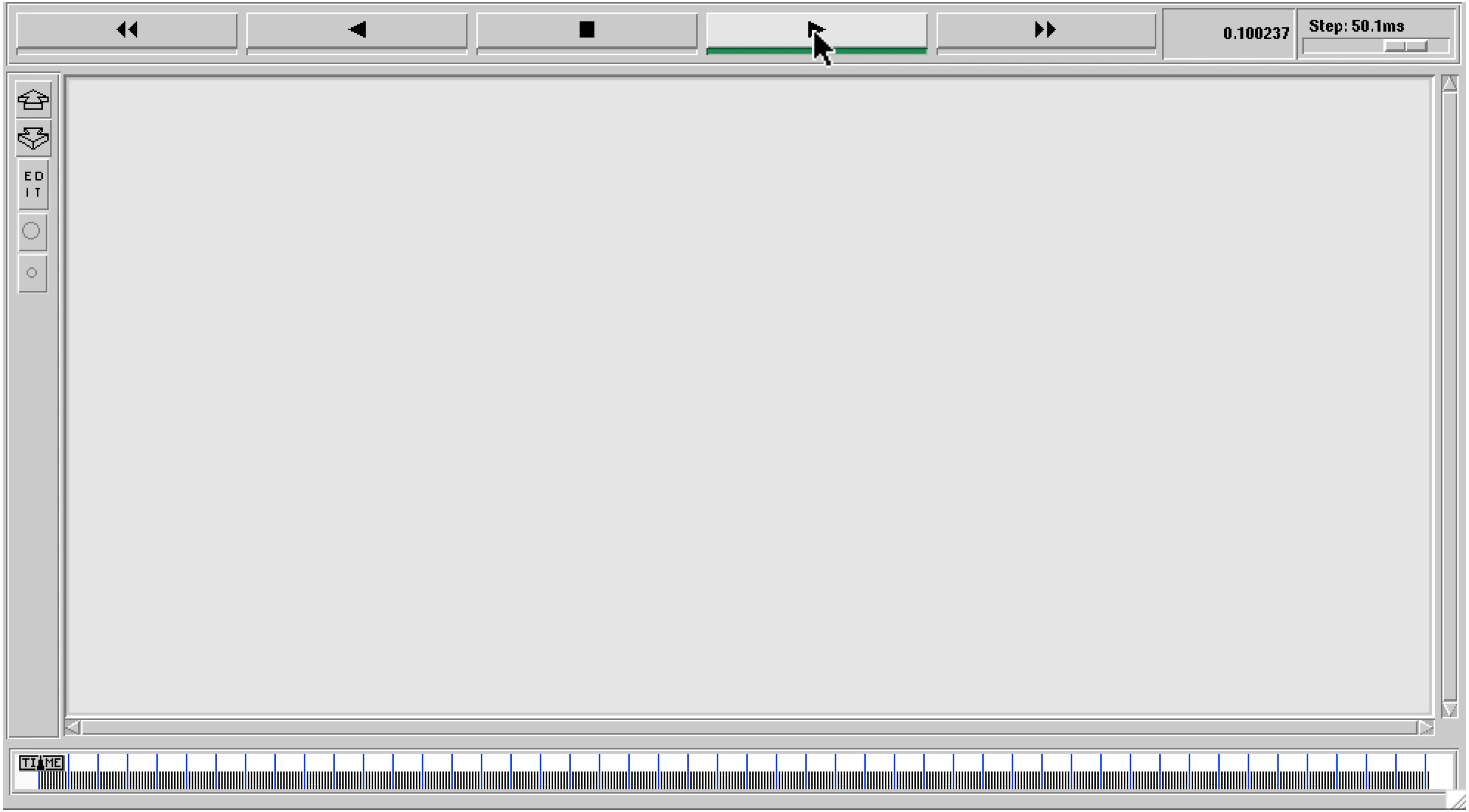
Simple Wireless - 2 Nodes

- Two wireless nodes connected in adhoc mode
  - Moving towards each other.
  - Communication occurs when in the range.
  - Animation: simple-wireless.mov

```
$node_(0) set X_ 5.0
$node_(0) set Y_ 2.0
$node_(1) set X_ 390.0
$node_(1) set Y_ 385.0
:
$ns at 5.0 "$node_(1) setdest 10.0 50.0 15.0”
$ns at 1.0 "$node_(0) setdest 40.0 20.0 2.0”
```

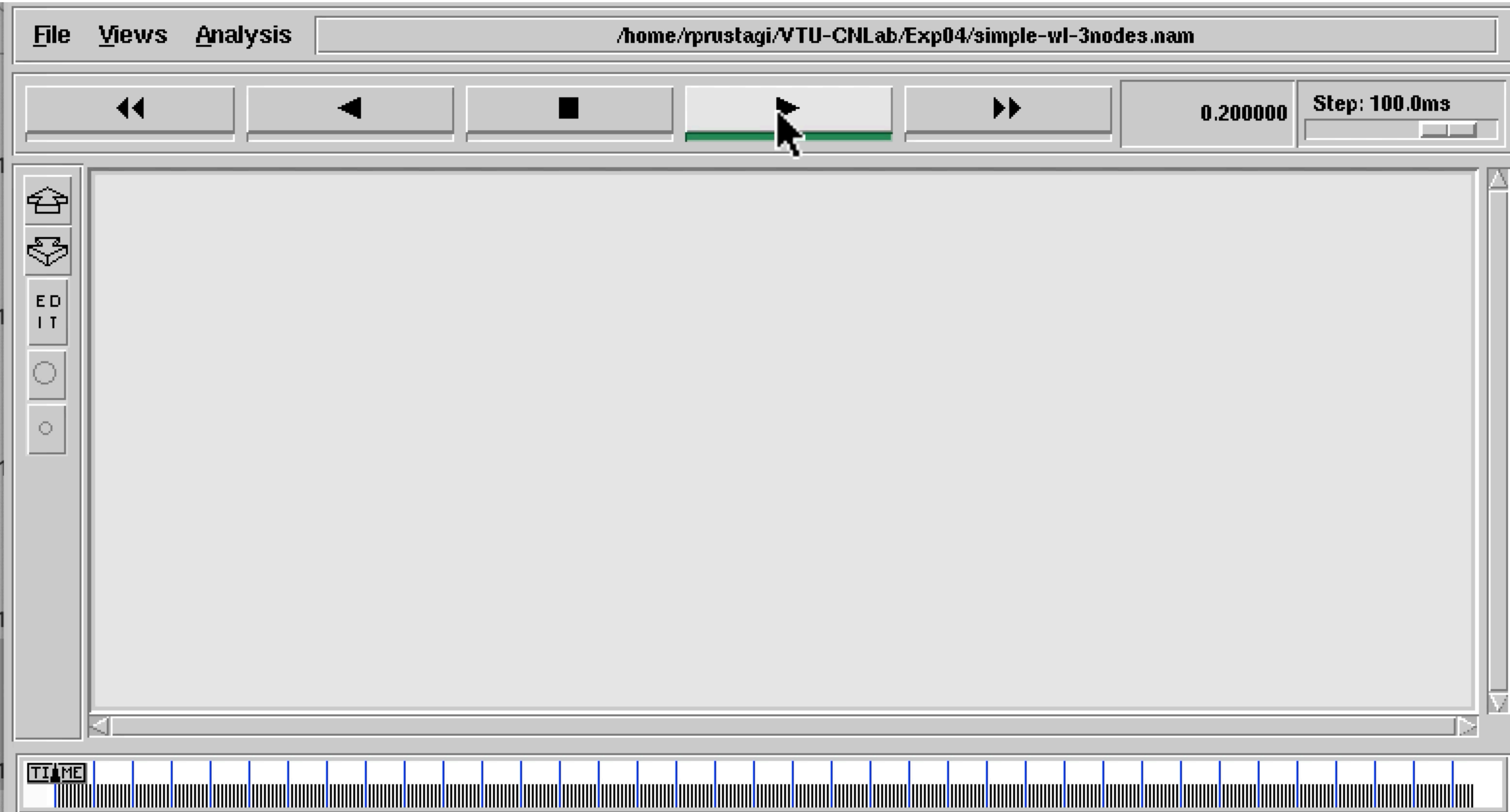
- Original source:
  - https://www.isi.edu/nsnam/ns/tutorial/nsscript5.html

Animation: Simple Wireless - 2 Nodes



Animation: Simple Wireless - 3 nodes

- Extension of 2 nodes adhoc wireless.
- Commuincation still between 2 nodes.
- 3rd node is just for demo inclusion.

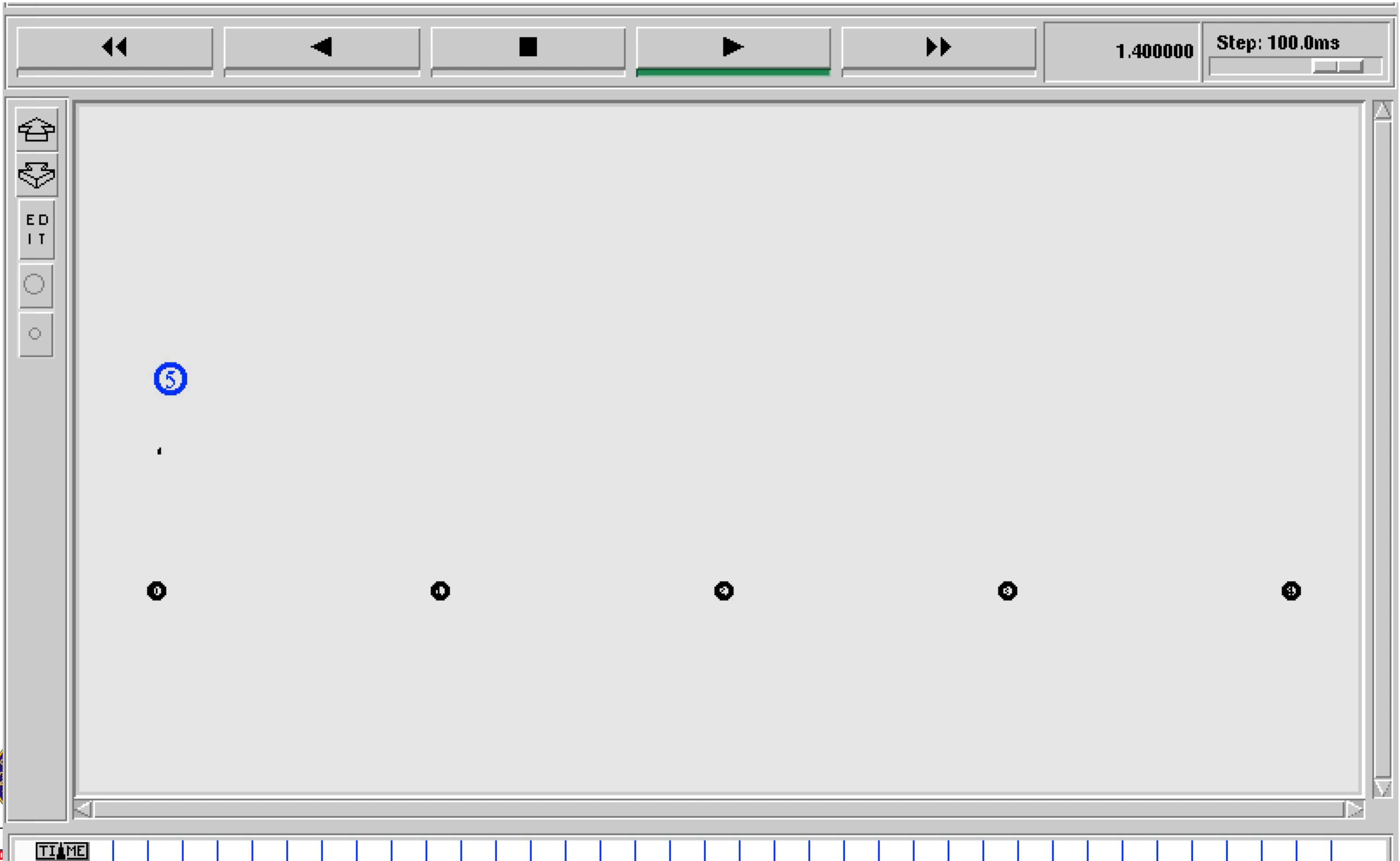


Adhoc Wireless - 6 nodes

- src: http://intronetworks.cs.luc.edu/current/html/ns2.html#wireless-simulation
- Demonstration of adhoc routing among 6 nodes.
- Topology:
  - 5 nodes are fixed
  - 6th node keeps moving over these 5 nodes.
  - Whenever distance from existing node becomes larger and reachable from another nearby node,
    - associations with nearby node occurs
    - Communication happens with nearby node,
    - But data transer continues with original connection.

Animation: Adhoc Wireless - 6 nodes

asource file: adhoc-ess.mov



WLAN-BSS

- AP (or BS) is connected to wired network and serving mobile nodes (wifi nodes)
  - This requires hierarchical routing.
    - To route packets between wireless and wired domains.
  - Routing for wired nodes are based on topology connectivity
    - The connectivity information (links) is used to build forwarding tables
  - Routing in wireless topology is based on adhoc routing
    - forwarding table is built by exchanging routing queries
- Thus, base station work as gateway between wired and wireless domains.
  - Need to define separate wired and wireless domains
  - Wired/wireless nodes are placed in respective domains
  - Domains/subdomains are defined by hierarchical structure

WLAN-BSS

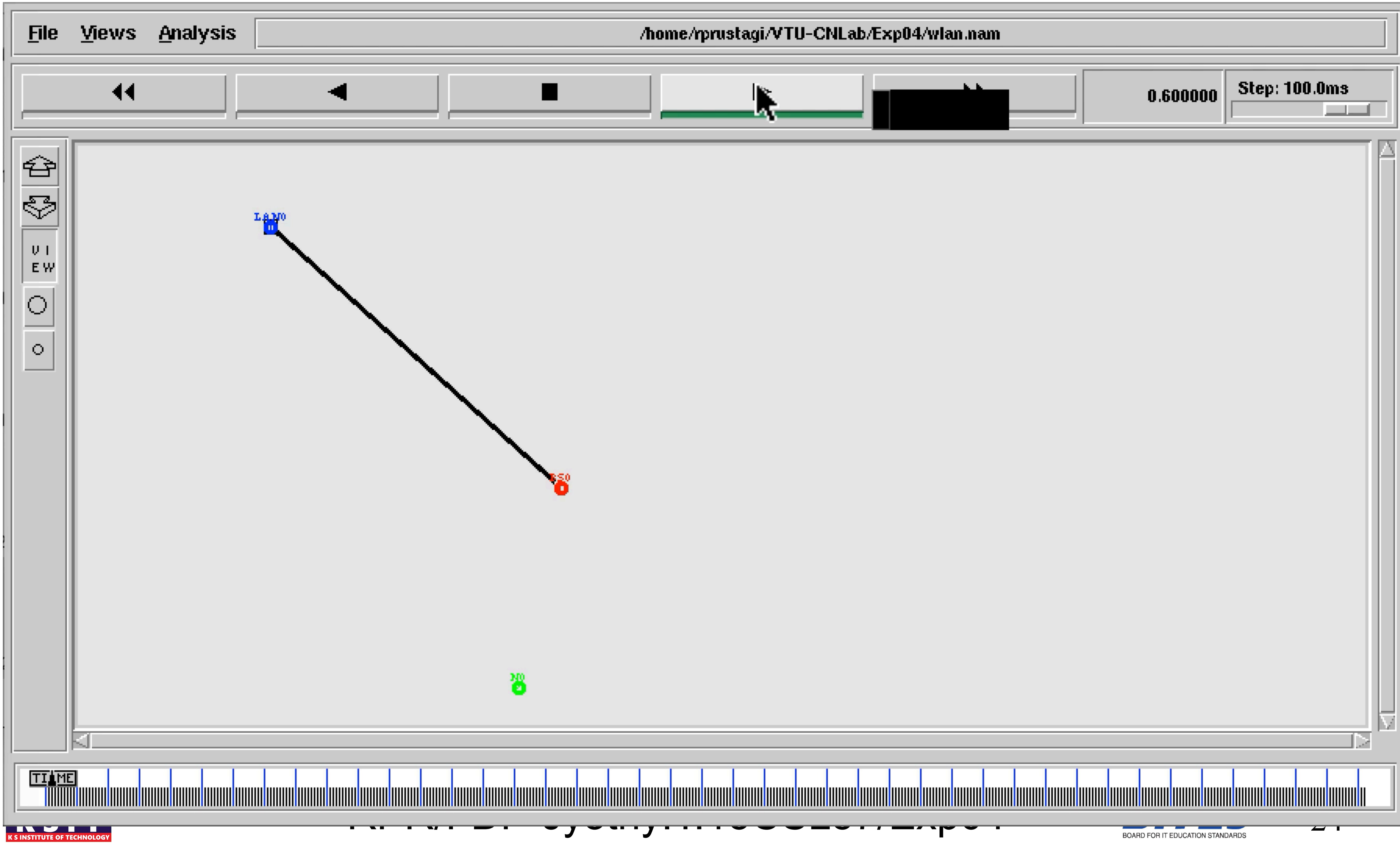
- Key configurations (github src: `wlan-bss.tcl`)

```
$ns node-config -addressType hierarchical
AddrParams set domain_num_ 2
lappend cluster_num 2 1
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 1 4
AddrParams set nodes_num_ $eilastlevel
```

  - `set W0 [$ns node [lindex 0.0.0]]`
  - `set BS0 [$ns node [lindex 1.0.0]]`
  - `set n0 [ $ns node [lindex 1.0.1]]`
  - `$n0 base-station [AddrParams addr2id [$BS0 node-addr]]`
  - original src: `tcl/ex/wired-and-wireless-sim.tcl` as part of ns2.35 distribution

Animation: WLAN-BSS

- wlan-bss.mov



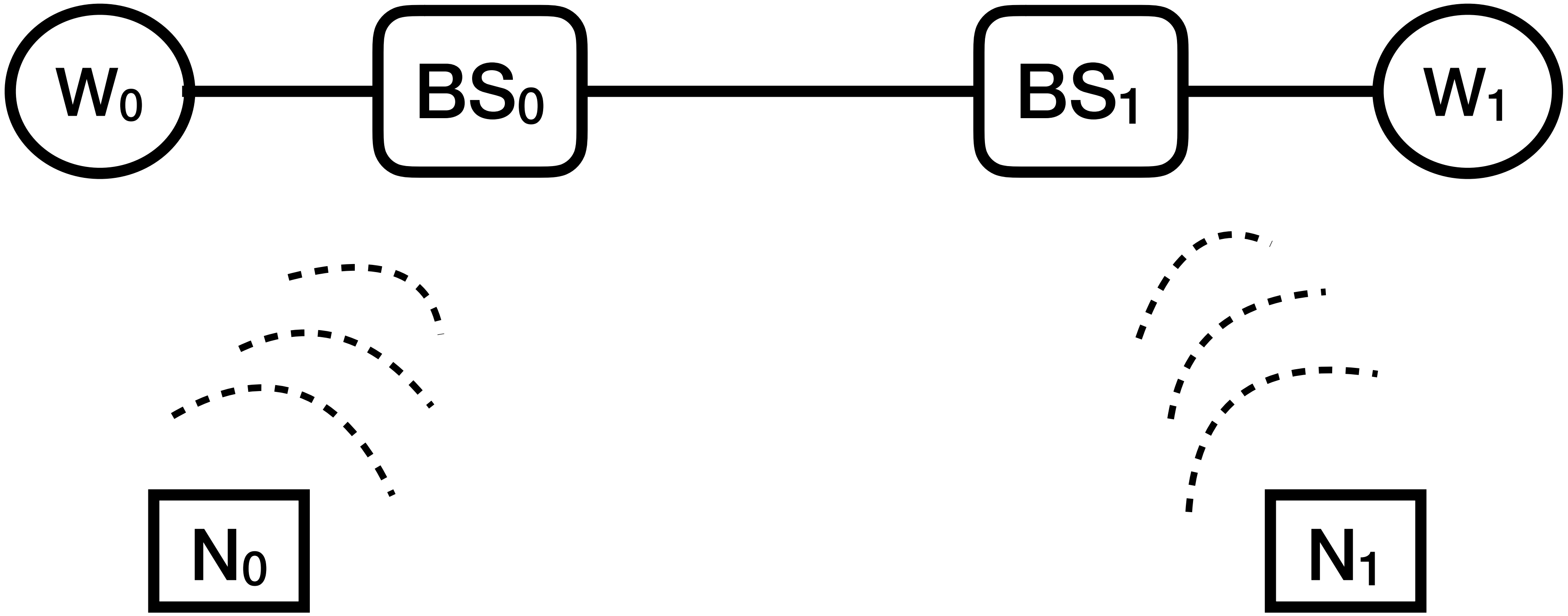
[redacted]

Hands on Exercises

- Exercise 1:
  - On the WLAN-BSS setup, add
    - 2nd wired node and 2nd wireless node.
    - Setup UDP/CBS traffic between these 2 nodes
    - This is in parallel to TCP/FTP between wired-0 and wireless-0.
- Exercise 2:
  - Add 2nd base status node e.g. BS1
  - Add another wired (W1) and wirless (N1) node to BS1
  - Connect BS1 to BS0 (existing base station) with wired link (10Mbps ethernet)
  - Establish applcation commuination as follows
    - TCP/FTP: N0-W1, and
    - UDP/CBR: N1-W0

ESS

- Example ESS Connection



Summary

- Wireless networking simulation
- Simple adhoc networks
- Adhoc routing with 6 nodes
- WLAN-BSS
- Exercises to explore