NS Tutorial: mobile and wireless network simulation

Ya Xu

USC/ISI

June 18, 1999

Jump to first page

Outlines

- **Use NS to simulate wireless network**
- Extend NS to support mobile and wireless application: Internal implementation
- Get support for your NS simulation
- Credit

Jump to first page

Get started

- Download ns-2.1b5 or lastest from the web
  - http://mash.cs.berkeley.edu/ns
- Install ns in your system
  - Binary release is provided for windows 9x/NT
  - NS-allinone package is strongly recommended
- Download nam if visualization is needed
  - http://mash.cs.berkeley.edu/nam
  - Included in ns-allinone package

Jump to first page

TCL Basics

- set a 123 ;#creates a variable whose name is a and whose value is 123
- set b $a ;#creates b whose value is a's value
- set c [expr $b+10] ;#evaluate the characters between the brackets as a TCL script and use the result as the value of c
- for {set i 0} {$i < 3} {incr i} {puts $i}

Jump to first page

A simple wireless simulation(1)

#Define Global Variables

set ns_ [new **Simulator**] ; create a ns simulator instance

set chan [new **Channel/WirelessChannel**]
; create a wireless channel

set prop [new **Propagation/TwoRayGround**]
; create a Radio propagation model

set topo [new **Topography**] ; create a topology and

$topo ***load_flatgrid*** 670 670 ; define it in 670x670 area and

*$prop **topology** $topo* ; propogation model keep a
; pointer to topology object

Jump to first page

A simple wireless simulation (2)

#Define standard ns/nam trace

set *nsf* [open *nstrace.tr* w] ; open file *nstrace.tr* for writing
*$ns_* ***trace-all*** *$nsf* ; ns trace information

set *namf* [open *namtrace.tr* w] ; open file *namtrace.tr* for
*$ns_* ***namtrace-all-wireless*** *$namf* 670 670
; writing nam trace info

A simple wireless simulation (3)

#Define mobileNode

set mnode [$*opt(rp)*-create-mobile-node *$id*]

; $opt(rp) defines what ad hoc routing protocols
; are, either “dsdv” or “dsr” is acceptable so far.
; $id defines the node id of the mobile node

 # Define node location and movement

$*mnode* set X_ 12.0 ;set node coordinate (x,y,z) to
$*mnode* set Y_ 27.0 ; (12.0, 27.0, 0.0)
$*mnode* set Z_ 0.0;
$*ns*_ at 1.0 $*mnode* setdest <x> <y> <speed>

; At time 1.0, node will start moving from its
;current location to (<x>,<y>, 0.0) at speed
;of <speed>

Jump to first page

A simple wireless example(4)

```
#Create 3 mobile nodes with dsdv routing
  for {set i 0} {$i < 3} {incr i} {
     dsdv-create-mobile-node $i
  }
#Include node movement scenario files
 source movement-scenario-files
#Include traffic scenario files
  source traffic-scenario-files
```

Jump to first page

A simple wireless example(5)

#Define simulation stop time

$*ns*_ at 10.0 “stop” ; stop simulation at time 10.0

#Start your simulation

*$ns*_ run

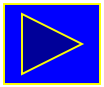
Source: See ns-2/tcl/ex/wireless-test.tcl for details

Jump to first page

Wireless Scenario Generator(1)

- Mobile Movement Generator

  setdest -n <num_of_nodes> -p pausetime -s <maxspeed> -t <simtime> -x <maxx> -y <maxy>

  See an example 

- Random movement

  $node start

  Source: See ns-2/indep-utils/cmu-scen-gen/setdest/

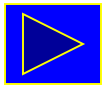
Jump to first page


Wireless Scenario Generator(2)

- Generating traffic pattern files
  - CBR traffic

  ns cbrgen.tcl [-type cbf|tcp] [-nn nodes] [-seed seed] [-mc connections] [-rate rate]
  - TCP traffic

  ns tcpgen.tcl [-nn nodes] [-seed seed]

  See an example 

  Source: See ns-2/indep-utils/cmu-scen-gen/

Network Components inside a mobilenode

- Link Layer
- ARP
- Interface Queue
- Mac Layer: IEEE 802.11
- Network Interface
- Radio Propagation Model
  - Friss-space attenuation($1/r^2$) at near distance
  - Two ray Ground ($1/r^4$) at far distance

Jump to first page

More about Trace

- Currently, 3 types of trace object
  - CMUTrace/Drop, CMUTrace/Recv, CMUTrace/Send
- Tracing packets that are dropped, received and sent by agents, router, mac layers or interface queues

  set sndT [cmu-trace Send “RTR” $node]

Visualize your simulation

- Use nam to visualize:
  - mobilenode position
  - mobilenode moving direction and speed
  - control the speed of playback
- See an example:

Jump to first page

Feature summary(1)

- Mac Layer: IEEE 802.11
- Address Resolution Protocol (ARP)
- Ad hoc routing protocols: DSDV, DSR
- Radio Propagation Model
  - Friss-space attenuation at near distances
  - Two ray ground at far distances
- Antenna: an omni-directional antenna having unity gain

Jump to first page

Feature summary (2)

- Scenario generator for traffic and node movement
- Base station node to bridge wired domain and wireless domain
- MobileIP
- Symmetric architecture with wired LAN (IEEE 802.3)

Jump to first page

Feature summary(3)

- Visualization of node movement and reachability
- Gridkeeper optimizer for some scenarios
- Energy consumption model for sensor networks
- Validation test-suites for dsdv, dsr, base station, mobileIP, gridkeeper

Jump to first page

Outlines

- Use NS to simulate wireless network
- **Extend NS to support mobile and wireless application: Internal Implementation**
- Get support for your NS simulation
- Credit

Jump to first page

Abstract the real mobile world into your simulation

- Node
- Packets
- Wireless channel and channel access
- Forwarding and routing
- Radio propagation model
- Trace/Visualization
- Event scheduler to make everything running

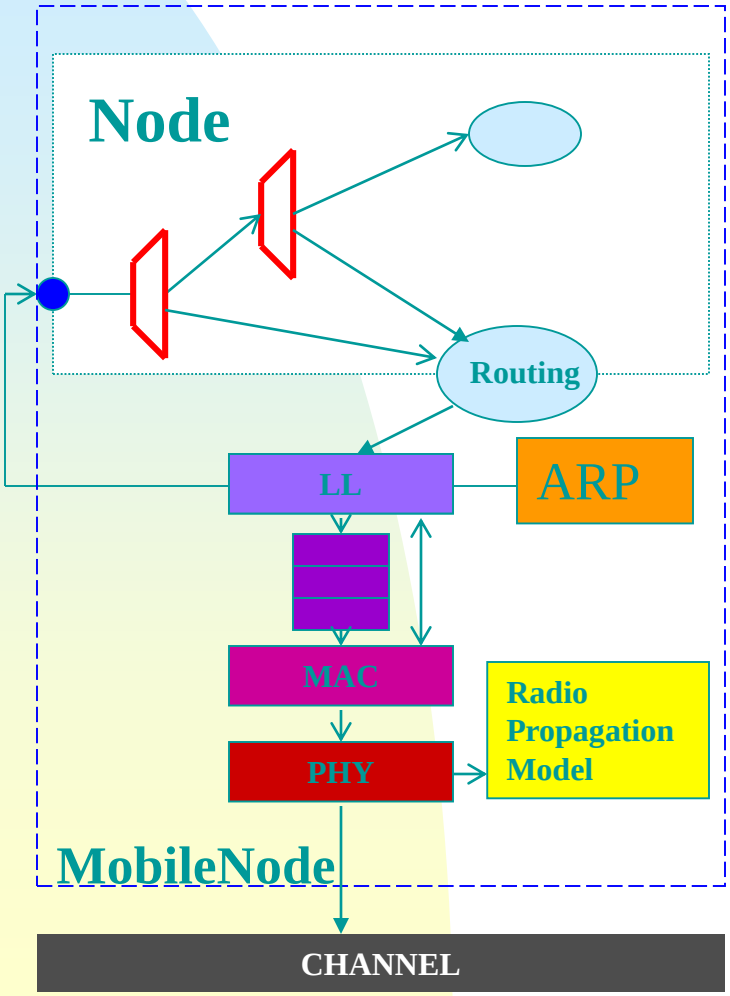
Jump to first page

A mobile node abstraction

- Location
  - coordinates (x,y,z)
- Movement
  - speed,direction, starting/ending location,time ...
- Forwarding
- Network stack for channel access
  - IEEE 802.11

Jump to first page

Implementing mobile node by Extending "standard" NS node

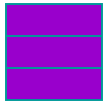


Classifier:Forwarding

Agent: Protocol Entity

Node Entry

LL:Link layer object



IFQ:Interface queue

MAC:Mac object

PHY:Net interface

Jump to first page

Wireless Channel

- Duplicate packets to all mobile nodes attached to the channel except the source itself.
- It is the receiver's responsibility to decide if it can receive the packet

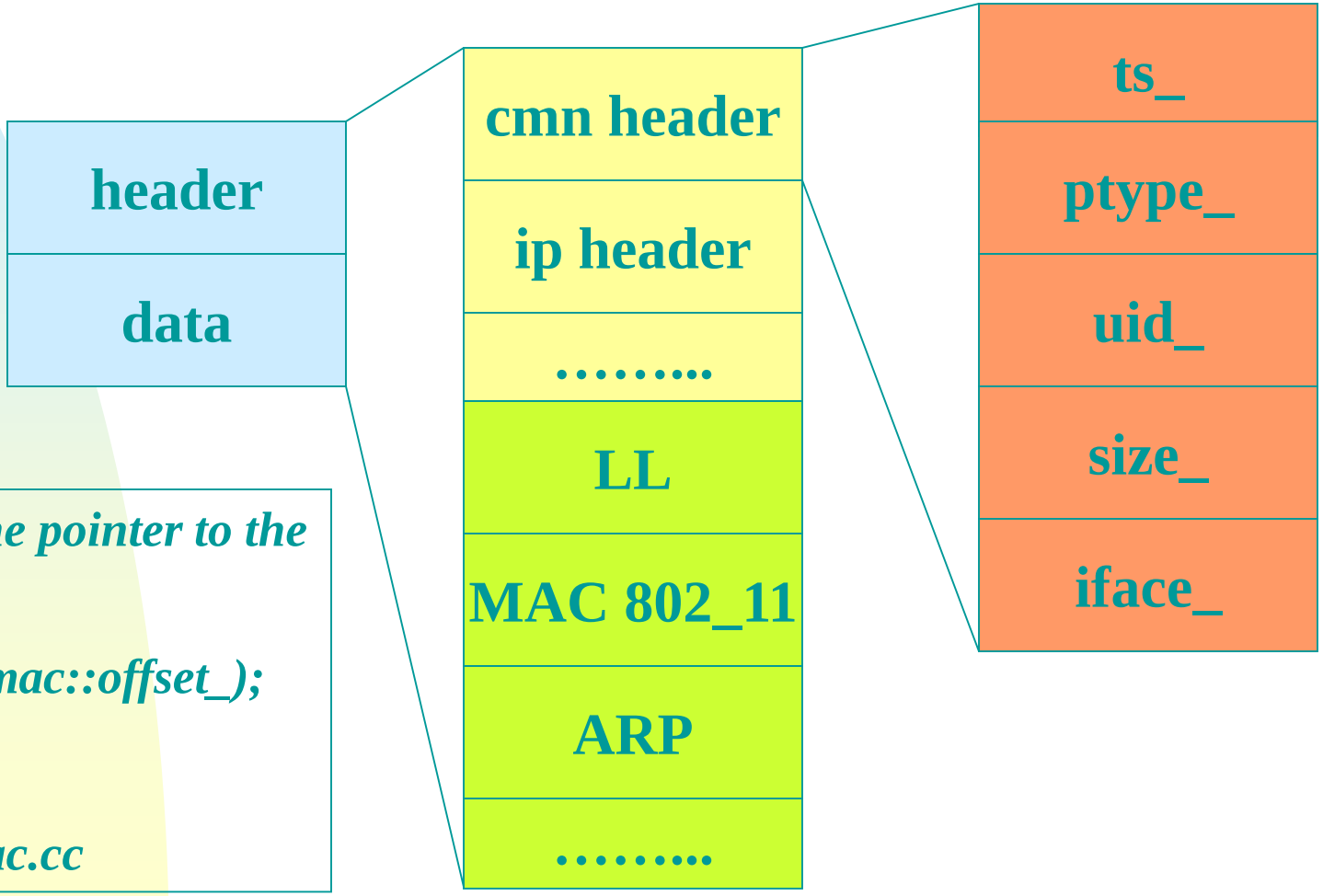
Jump to first page

NS split model in the MobileNode

- Control/”Data” separation
  - control operations in otcl: plumbing
  - data pass through C++ object:composible

Jump to first page

Extending NS Packet Format to support wireless simulation



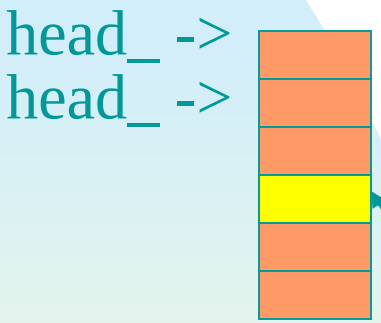
Example: Get the pointer to the Mac header:

p->access(hdr_mac::offset_);

Source: ns-2/mac.cc

Jump to first page

Discrete Event Scheduler



Jump to first page

Outlines

- Use NS to simulate wireless network
- Extend NS to support mobile and wireless application: Internal Implementation
- **Get support for your NS simulator**
- Credit

Jump to first page

Get Help

- Main ns-2 web pages
  - **http://titan.cs.uni-bonn.de/~greis/ns/ns.html**
  - http://mash.cs.berkeley.edu/ns
- Mailing lists
  - ns-users@mash.cs.berkeley.edu
  - ns-announce@mash.cs.berkeley.edu
- To subscribe
  - majordomo@mash.cs.berkeley.edu
- Ask your classmates because ns is popular

Outlines

- Use NS to simulate wireless network
- Extend NS to support mobile and wireless application: Internal Implementation
- Get support for your NS simulation
- **Credit**

Jump to first page

Who committed the code

- CMU
- UC Berkeley
- Sun Microsystem Inc.
- USC/ISI

Jump to first page

Appendix A: Movement file

$node_(2) set Z_ 0.000000000000
$node_(2) set Y_ 199.373306816804
$node_(2) set X_ 591.256560093833
$node_(1) set Z_ 0.000000000000
$node_(1) set Y_ 345.357731779204
$node_(1) set X_ 257.046298323157
$node_(0) set Z_ 0.000000000000
$node_(0) set Y_ 239.438009831261
$node_(0) set X_ 83.364418416244
$ns_ at 50.000000000000 "$node_(2) setdest 369.463244915743 170.519203111152 3.3
71785899154"
$ns_ at 51.000000000000 "$node_(1) setdest 221.826585497093 80.855495003839 14.9
09259208114"
$ns_ at 33.000000000000 "$node_(0) setdest 89.663708107313 283.494644426442 19.1
53832288917"

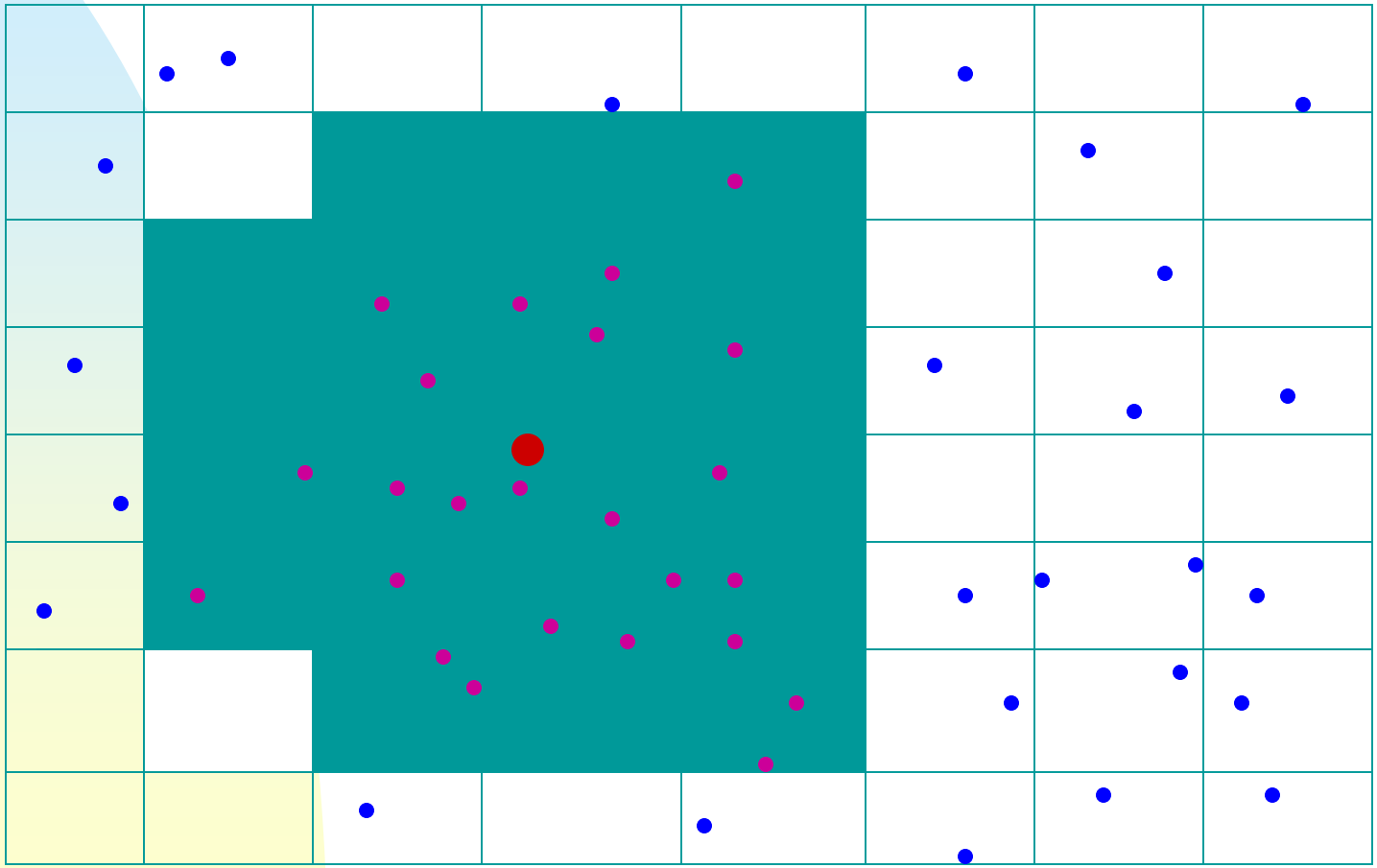
Jump to first page

Appendix B: Traffic Scenario

```
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(0) $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_ 4.0
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 127.93667922166023 "$cbr_(0) start"
…….
```

Jump to first page

Greedkeeper: an optimizer*



** optimization depends on your scenario*

Jump to first page