NS Tutorial: mobile and wireless network simulation

Ya Xu

USC/ISI

September 10th, 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

Get started

- Download the lastest ns snapshot from:
  - 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

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}

A simple wireless simulation(1)

- **Scenario**
  - containing 3 mobile nodes
  - moving within 670mX670m flat topology
  - using DSDV ad hoc routing protocol
  - Random Waypoint mobility model
  - TCP and CBR traffic
  - See:
- *ns-2/tcl/ex/wireless-demo-csci694.tcl*

A simple wireless simulation(2)

#Define Global Variables

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

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

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

A simple wireless simulation (3)

#Define standard ns/nam trace

set *tracefd* [open *694demo.tr* w]

*$ns_* ***trace-all*** *$tracefd*

set *namtrace* [open *694demo.nam* w]

*$ns_* ***namtrace-all-wireless*** *$namtrace* 670 670

Jump to first page

A simple wireless simulation (4)

#Create “God”

set god_ [create-god 3]

• God is used to store an array of the shortest number of hops required to reach from one node to an other.

•For example:

•$ns_ at 899.00 “$god_ setdist 2 3 1”

A simple wireless simulation (5)

#Define how a mobile node should be created

```
$ns_ node-config -adhocRouting DSDV\
                 -llType LL \
                 -macType Mac/802_11\
                 -ifqLen 50 \
                 -ifqType Queue/DropTail/PriQueue \
                 -antType Antenna/OmniAntenna \
                 -propType Propagation/TwoRayGround \
                 -phyType Phy/WirelessPhy \
                 -channelType Channel/WirelessChannel \
                 -topoInstance $topo
                 -agentTrace ON \
                 -routerTrace OFF \
                 -macTrace OFF
```

Jump to first page

A simple wireless simulation (6)

#Create a mobile node and attach it to the channel

set node [$ns_ ***node***]
$node ***random-motion*** 0 ;# disable random motion

– Use "for loop" to create 3 nodes:

```
for {set i < 0} {$i<3} {incr i} {
    set node_($i) [$ns_ node]
}
```

A simple wireless example(7)

#Define node movement model

source movement-scenario-files

#Define traffic model

source traffic-scenario-files

A simple wireless example(8)

#Define node initial position in nam

for {set i 0} {$i < 3 } { incr i} {
    $ns_ ***initial_node_position*** $node_($i) 20
}

#Tell ns/nam the simulation stop time

*$ns_* at 200.0 "$ns_ ***nam-end-wireless*** 200.00"
$ns_ at 200.00 "$ns_ ***halt***"

#Start your simulation

*$ns_* run

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/

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/

Sensor Node extension

- Node is energy-aware
- Define node by adding new options:

```
$ns_ node-config -energyModel EnergyModel
                 -initialEnergy  100.0
                 -txPower        0.6
                 -rxPower        0.2
```

Summary of the API changes

New API

- $ns_ node-config
- $ns node
- no global variable dependency
- namtrace-all-wireless
- Energy model support
- No global definition of chan and prop

OLD API

- dsr/dsdv/tora-create-mobile-node
- strong global variable dependency
- no nam support
- no energy model
- need global chan and prop

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

Visualize your simulation

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

Feature summary(1)

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

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)

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, tora, 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

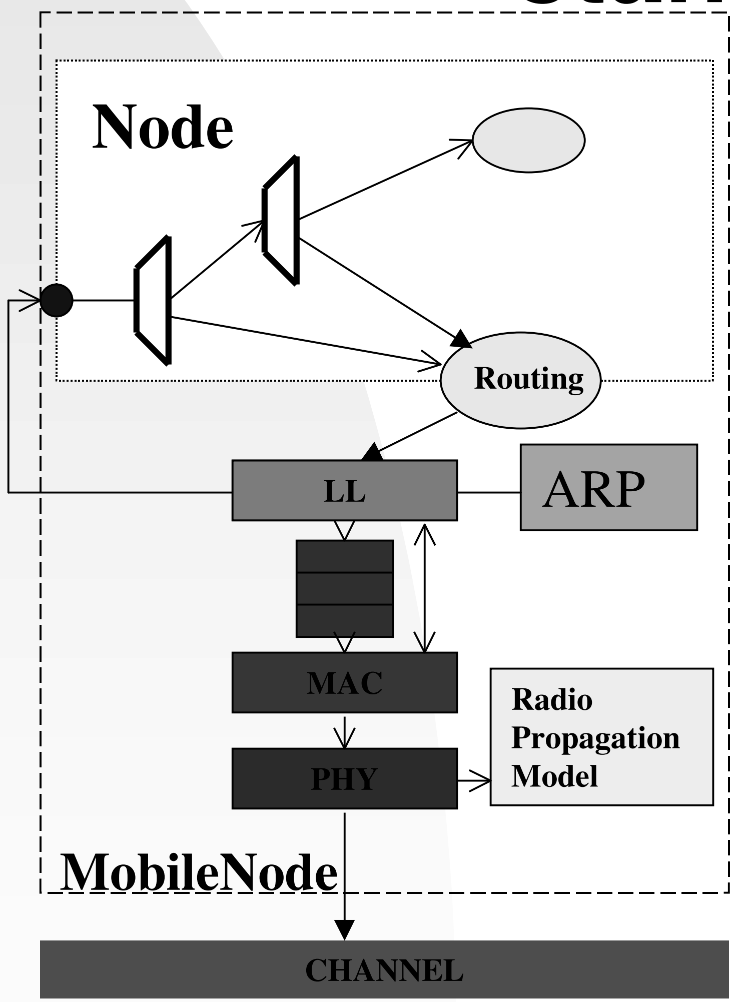
Abstract the real mobile world into your simulation

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

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

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

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

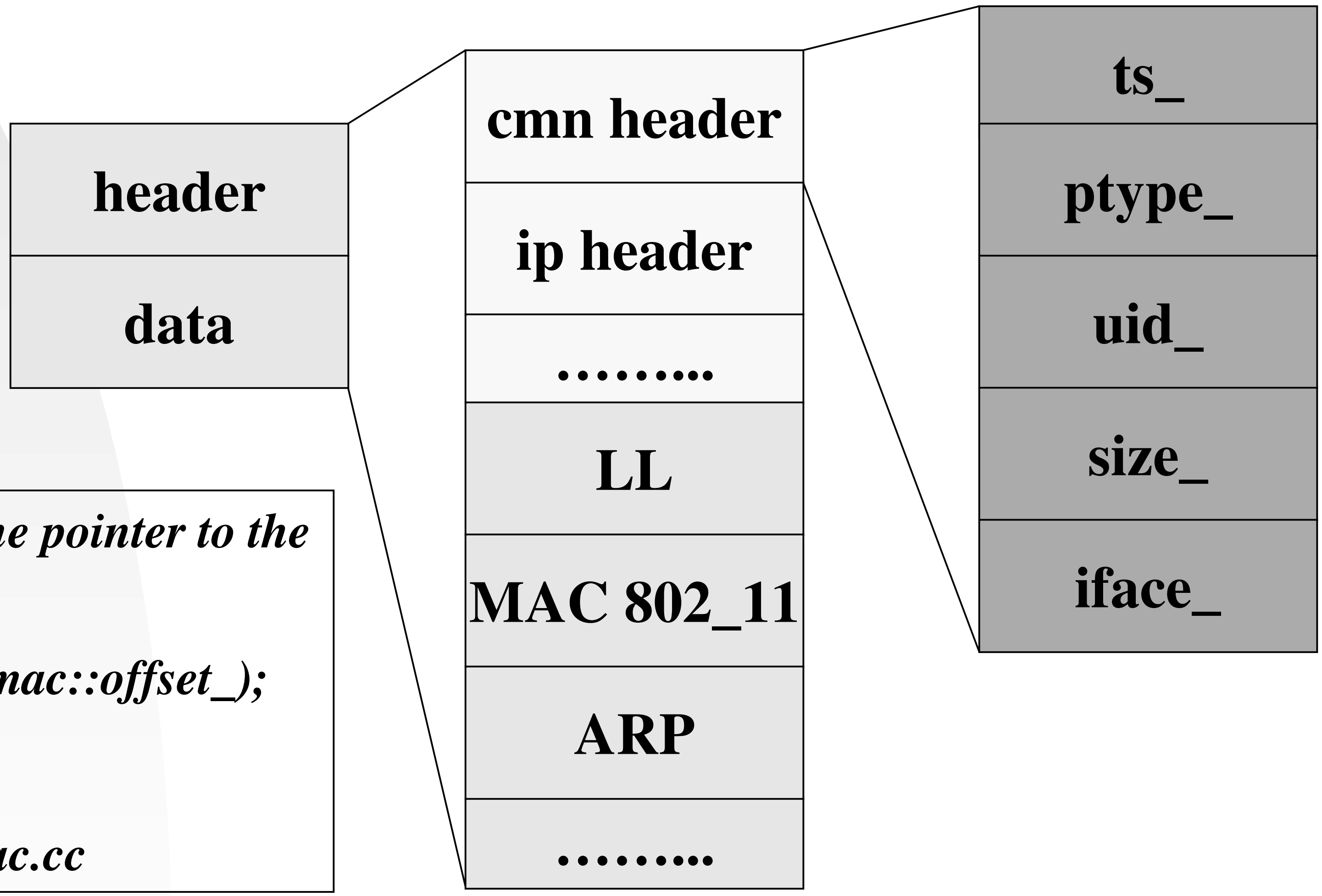
Jump to first page

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:

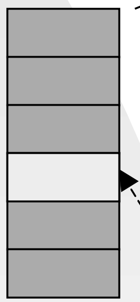
p->access(hdr_mac::offset_);

Source: ns-2/mac.cc

Jump to first page

Discrete Event Scheduler

head_ ->
head_ ->



p = deque(); // get current event
p->handler_->handle(p)

An example: node position handler
node->update_position();
node->random_destination();

insert

Insert new event back to the queue
s = Scheduler::instance();
s.schedule(&node->handle, &node->pos_intr, interval);

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

Appendix A: Movement file

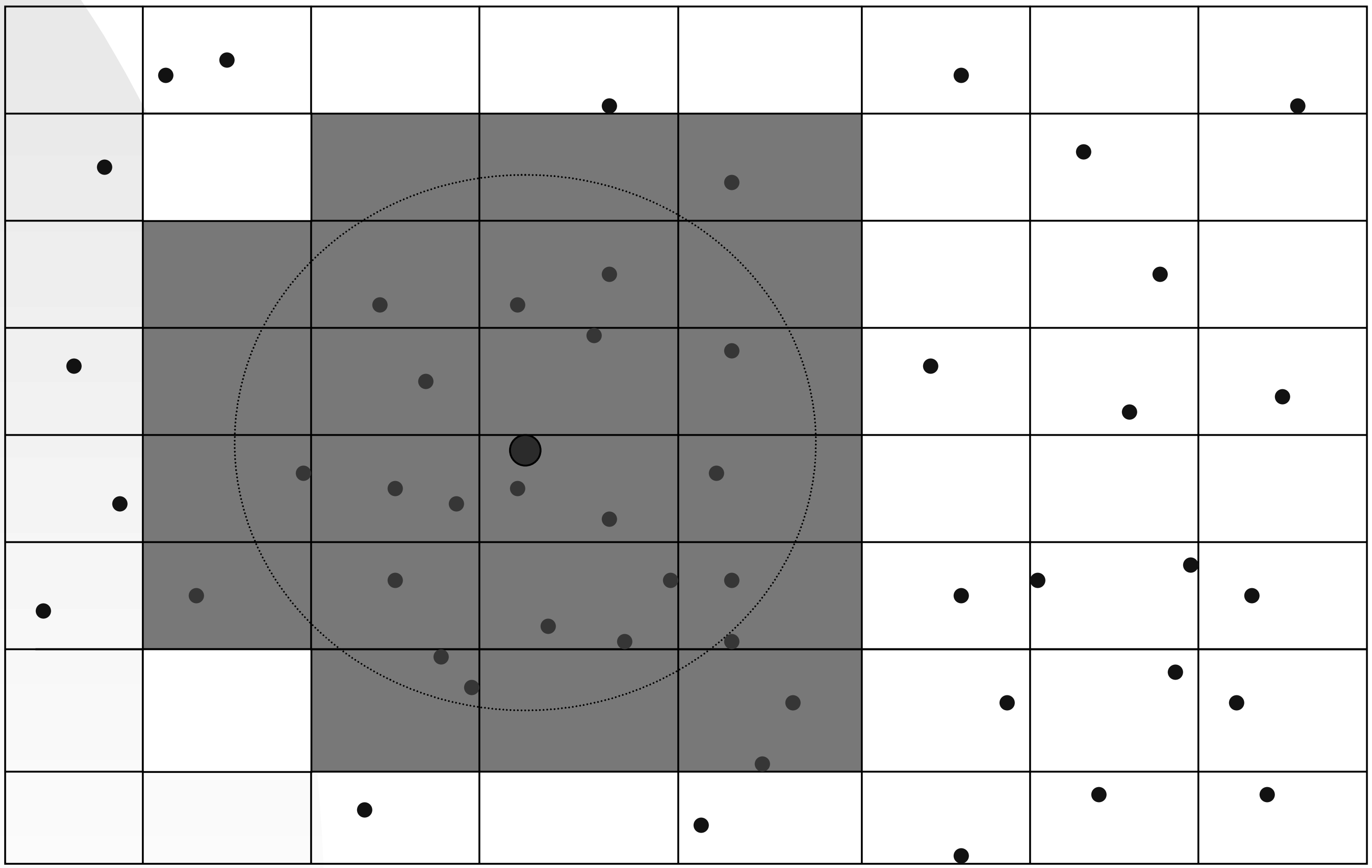
```
$node_(2) set Z_ 0.000000000000
$node_(2) set Y_ 199.373306816804
$node_(2) set X_ 591.256560093833
$node_(1) set Z_ 0.000000000000
$node_(1) set Y_ 345.357731779204
$node_(1) set X_ 257.046298323157
$node_(0) set Z_ 0.000000000000
$node_(0) set Y_ 239.438009831261
$node_(0) set X_ 83.364418416244
$ns_ at 50.000000000000 "$node_(2) setdest 369.463244915743
      170.519203111152 3.371785899154"
$ns_ at 51.000000000000 "$node_(1) setdest 221.826585497093
      80.855495003839 14.909259208114"
$ns_ at 33.000000000000 "$node_(0) setdest 89.663708107313
      283.494644426442 19.153832288917"
```

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