Wireless world in NS

Padma Haldar
USC/ISI

Outline

- Introduction
  - Wireless basics
  - Wireless internals
- Ad hoc routing
- Mobile IP
- Satellite networking
- Directed diffusion

Contributions to mobility in ns

- Original mobility model in ns contributed by CMU's Monarch group
- Other major contributions from UCB, Sun microsystems, univ of cincinnati, ISI etc
- Other contributed models (not integrated) in wireless ns includes Blueware, BlueHoc, Mobiwan, GPRS, CIMS etc

Wireless model

- Mobilenode at core of mobility model
- Mobilenodes can move in a given topology, receive/transmit signals from/to wireless channels
- Wireless network stack consists of LL, ARP, MAC, IFQ etc
- Allows simulations of multi-hop ad hoc networks, wireless LANs, sensor networks etc

Wireless Example for ad hoc routing

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

An Example – Step 1

```
# Define Global Variables
# create simulator
set ns [new Simulator]

# create a flat topology in a 670m x 670m area
set topo [new Topography]
$topo load_flatgrid 670 670
```

An Example – Step 2

```
# Define standard ns/nam trace
# ns trace
set tracefd  [open demo.tr w]
$ns trace-all $tracefd
# nam trace
set namtrace [open demo.nam w]
$ns namtrace-all-wireless $namtrace 670 670
```

GOD
(General Operations Director)

- Stores smallest number of hops from one node to another
- Optimal case to compare routing protocol performance
- Automatically generated by scenario file
- set god [create-god <no of mnodes>]
- $god set-dist <from> <to> <#hops>

Example –Step 3

- *Create God*

  set god [create-god 3]

  $ns at 900.00 “$god setdist 2 3 1”

An Example – Step 4

```
# Define how a mobile node is configured
$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
```

An Example – Step 5

```
# Next create a mobile node, attach it to the
channel
set node(0) [$ns node]
# disable random motion
$node(0) random-motion 0

# Use “for” loop to create 3 nodes:
for {set i < 0} {$i < 3} {incr i} {
    set node($i) [$ns node]
    $node($i) random-motion 0
}
```

Mobilenode Movement

- Node position defined in a 3-D model
- However z axis not used

  ```
  $node set X_ <x1>
  $node set Y_ <y1>
  $node set Z_ <z1>
  $node at $time setdest <x2> <y2>
    <speed>
  ```
- Node movement may be logged

Scenario Generator: Movement

- Mobile Movement Generator

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

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

- Random movement
  - `$node random-motion 1`
  - `$node start`

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"
```

Scenario Generator: Traffic

- Generating traffic pattern files
  - CBR traffic

    ```
    ns cbrgen.tcl [-type cbr|tcp] [-nn nodes]
      [-seed seed] [-mc connections] [-rate
      rate]
    ```

  - TCP traffic

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

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

A 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"
…….
```

An Example – Step 6

```
# Define node movement model
source <movement-scenario-files>

# Define traffic model
source <traffic-scenario-files>
```

An Example – Step 7

```
# 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 200.0”
$ns at 200.0 “$ns halt”

# Start your simulation
$ns run
```

Energy 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
```

19

nam Visualization

- Use nam to visualize:
  - Mobile node position
  - Mobile node moving direction and speed
  - Energy consumption at nodes (color keyed)

20

nam Visualization

- Replace

```
$ns namtrace-all $fd
```

with

```
$ns namtrace-all-wireless $fd
```

At the end of simulation, do

```
$ns nam-end-wireless [$ns now]
```

21

Outline

- Introduction
  - Wireless basics
  - Wireless internals
- Ad hoc routing
- Mobile IP
- Satellite networking
- Directed diffusion

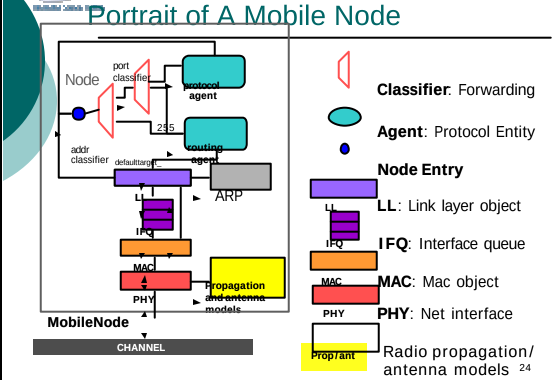
22

Wireless Internals

- Mobilenode
  - Basic node that has address and port de-muxes, routing agent etc
  - Stack of network components consisting of LL, MAC, NetIF radio-model etc
- Wireless channel

23

Portrait of A Mobile Node



Classifier: Forwarding

Agent: Protocol Entity

Node Entry

LL: Link layer object

IFQ: Interface queue

MAC: Mac object

PHY: Net interface

Radio propagation/ antenna models

24

Mobile Node : Components

- Classifiers
  - defaulttarget_ points to routing agent object
  - 255 is the port id assigned for rtagent_
- Routing agent
  - May be ad hoc routing protocol like AODV, DSDV or directed diffusion

Mobile Node: Components

- Link Layer
  - Same as LAN, but with a separate ARP module
  - Sends queries to ARP
- ARP
  - Resolves IP address to hardware (MAC) address
  - Broadcasts ARP query
- Interface queue
  - Gives priority to routing protocol packets
  - Has packet filtering capacity

Mobile Node: Components

- MAC
  - 802.11
    - IEEE RTS/CTS/DATA/ACK for unicast
    - Sends DATA directly for broadcast
  - SMAC (work in progress)
- Network interface (PHY)
  - Used by mobilenode to access channel
  - Stamps outgoing pkts with meta-data
  - Interface with radio/antenna models

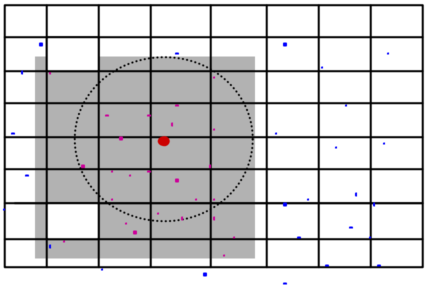
Mobile Node: Components

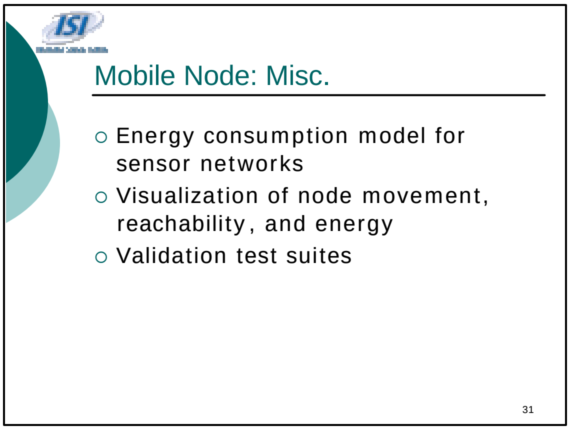
- Radio Propagation Model
  - Friss-space model – attenuation at near distance
  - Two-ray ground reflection model for far distance
  - Shadowing model - probabilistic
- Antenna
  - Omni-directional, unity-gain

Wireless Channel

- Duplicate packets to all mobile nodes attached to the channel except the sender
- It is the receiver's responsibility to decide if it will accept the packet
  - Collision is handled at individual receiver
  - O(N^2) messages → grid keeper, reference-copying etc

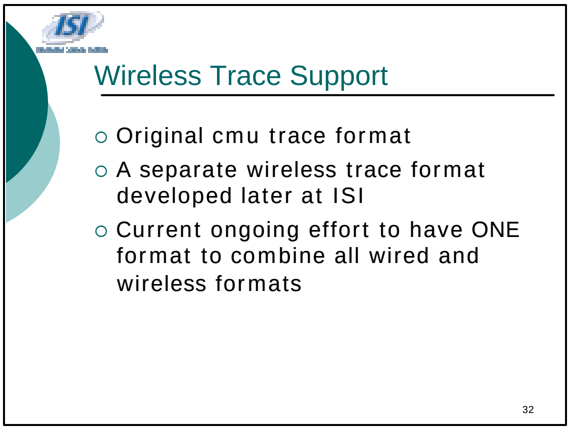
Grid-keeper: An Optimization





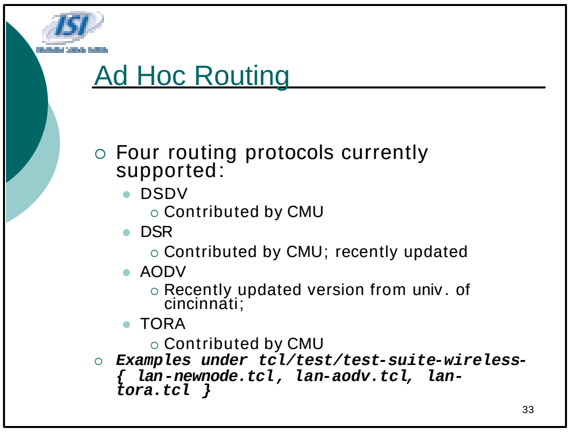
Mobile Node: Misc.

- Energy consumption model for sensor networks
- Visualization of node movement, reachability, and energy
- Validation test suites



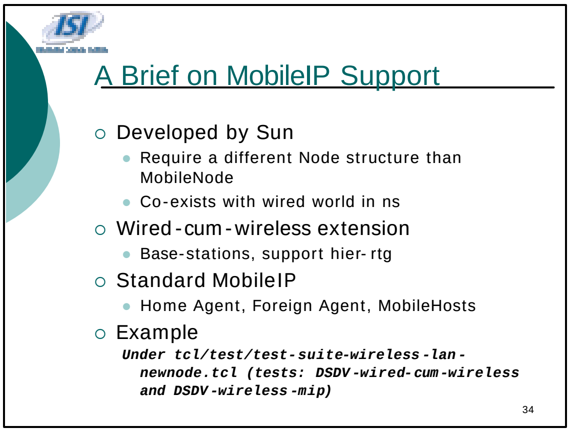
Wireless Trace Support

- Original cmu trace format
- A separate wireless trace format developed later at ISI
- Current ongoing effort to have ONE format to combine all wired and wireless formats



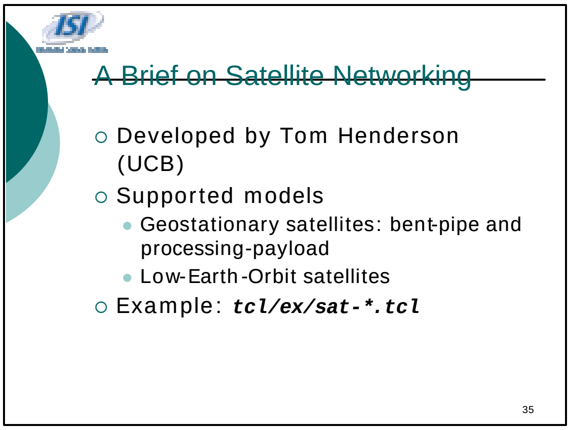
Ad Hoc Routing

- Four routing protocols currently supported:
  - DSDV
    - Contributed by CMU
  - DSR
    - Contributed by CMU; recently updated
  - AODV
    - Recently updated version from univ. of cincinnati;
  - TORA
    - Contributed by CMU
- ***Examples under tcl/test/test-suite-wireless-{ lan-newnode.tcl, lan-aodv.tcl, lan-tora.tcl }***



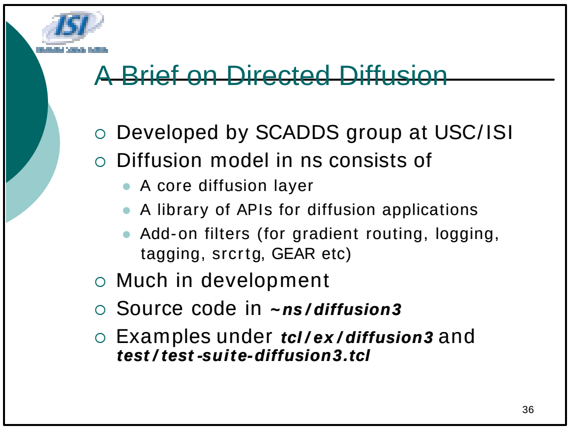
A Brief on MobileIP Support

- Developed by Sun
  - Require a different Node structure than MobileNode
  - Co-exists with wired world in ns
- Wired-cum-wireless extension
  - Base-stations, support hier-rtg
- Standard MobileIP
  - Home Agent, Foreign Agent, MobileHosts
- Example
  Under tcl/test/test-suite-wireless-lan-newnode.tcl (tests: DSDV-wired-cum-wireless and DSDV-wireless-mip)



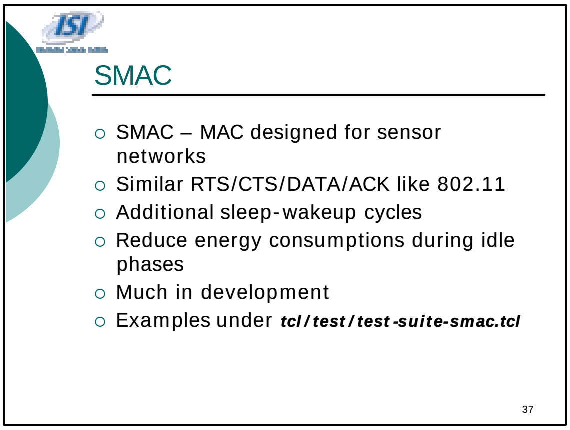
A Brief on Satellite Networking

- Developed by Tom Henderson (UCB)
- Supported models
  - Geostationary satellites: bent-pipe and processing-payload
  - Low-Earth-Orbit satellites
- Example: ***tcl/ex/sat-*.tcl***



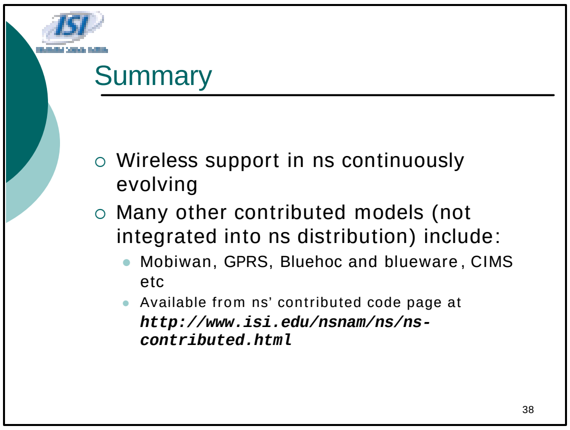
A Brief on Directed Diffusion

- Developed by SCADDS group at USC/ISI
- Diffusion model in ns consists of
  - A core diffusion layer
  - A library of APIs for diffusion applications
  - Add-on filters (for gradient routing, logging, tagging, srcrtg, GEAR etc)
- Much in development
- Source code in ***~ns/diffusion3***
- Examples under ***tcl/ex/diffusion3*** and ***test/test-suite-diffusion3.tcl***



SMAC

- SMAC – MAC designed for sensor networks
- Similar RTS/CTS/DATA/ACK like 802.11
- Additional sleep-wakeup cycles
- Reduce energy consumptions during idle phases
- Much in development
- Examples under ***tcl/test/test-suite-smac.tcl***



Summary

- Wireless support in ns continuously evolving
- Many other contributed models (not integrated into ns distribution) include:
  - Mobiwan, GPRS, Bluehoc and blueware, CIMS etc
  - Available from ns' contributed code page at ***`http://www.isi.edu/nsnam/ns/ns-contributed.html`***