

NS-2 Tutorial

Demokritos University of Thrace

Depts of:

Data Analysis

&

Computer Networks

ns-2 overview

- **Collection of various protocols at multiple layers:**
 - TCP(reno, tahoe, vegas, sack)
 - MAC(802.11, 802.3, TDMA)
 - Ad-hoc Routing (DSDV, DSR, AODV, TORA)
 - Sensor Network (diffusion, gaf)
 - Multicast protocols, Satellite protocols, and many others
- **Codes are contributed from multiple research communities:**
 - Good: Large set of simulation modules
 - Bad: Level of support and documentation varies
- **The source code and documentation is currently maintained by VINT project at ISI**

ns-2 Environment

Simulation Scenario

1

2

Tcl Script

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

C++
Implementation

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

Wireless Simulation in ns-2

- Contributed from CMU's Monarch project (Wireless extension to ns-2)
- Various modules were added to ns-2 to simulate node mobility and wireless networking
 - Mobile Node
 - Ad-hoc Routing(DSR, DSDV, TORA, AODV)
 - MAC802.11
 - Radio Propagation Model
 - Channel

Mobile Node Modules

- Agent
 - Responsible for packet generations and receptions
 - Can think of it as an Application layer
 - CBR(Constant Bit Rate), TCP, Sink, FTP, etc.
- RTagent(DSDV, TORA, AODV) or DSR
 - Ad-hoc network routing protocols
 - Configure multi hop routes for packets
- LL (Link Layer)
 - Runs data link protocols
 - Fragmentation and reassembly of packet
 - Runs Address Resolution Protocol(ARP) to resolve IP address to MAC address conversions

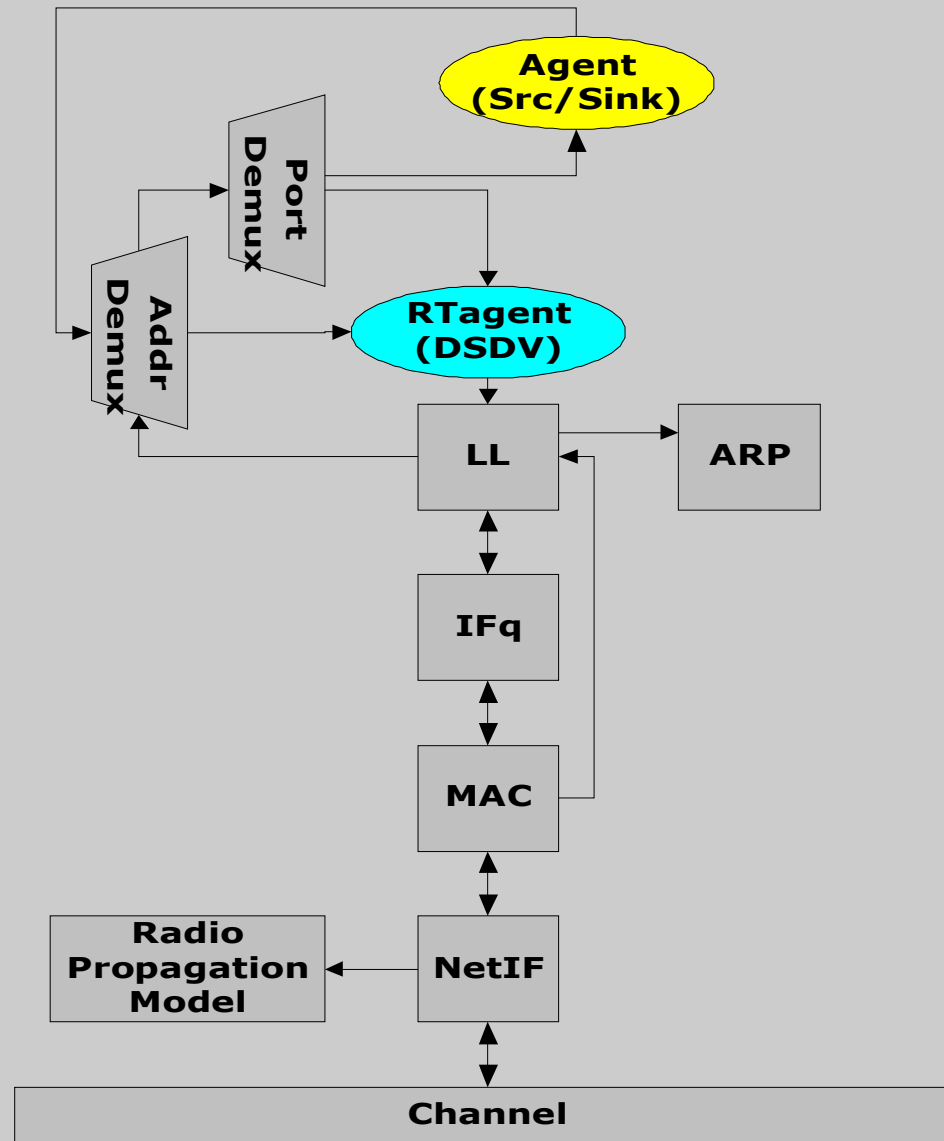
Mobile Node Modules (Continued)

- IFq (Interface Queue)
 - PriQueue is implemented to give priority to routing protocol packets
 - Supports filter to remove packets destined to specific address
- Mac Layer
 - IEEE 802.11 protocol is implemented
 - Uses RTS/CTS/DATA/ACK pattern for all unicast pkts and DATA for broadcast pkts

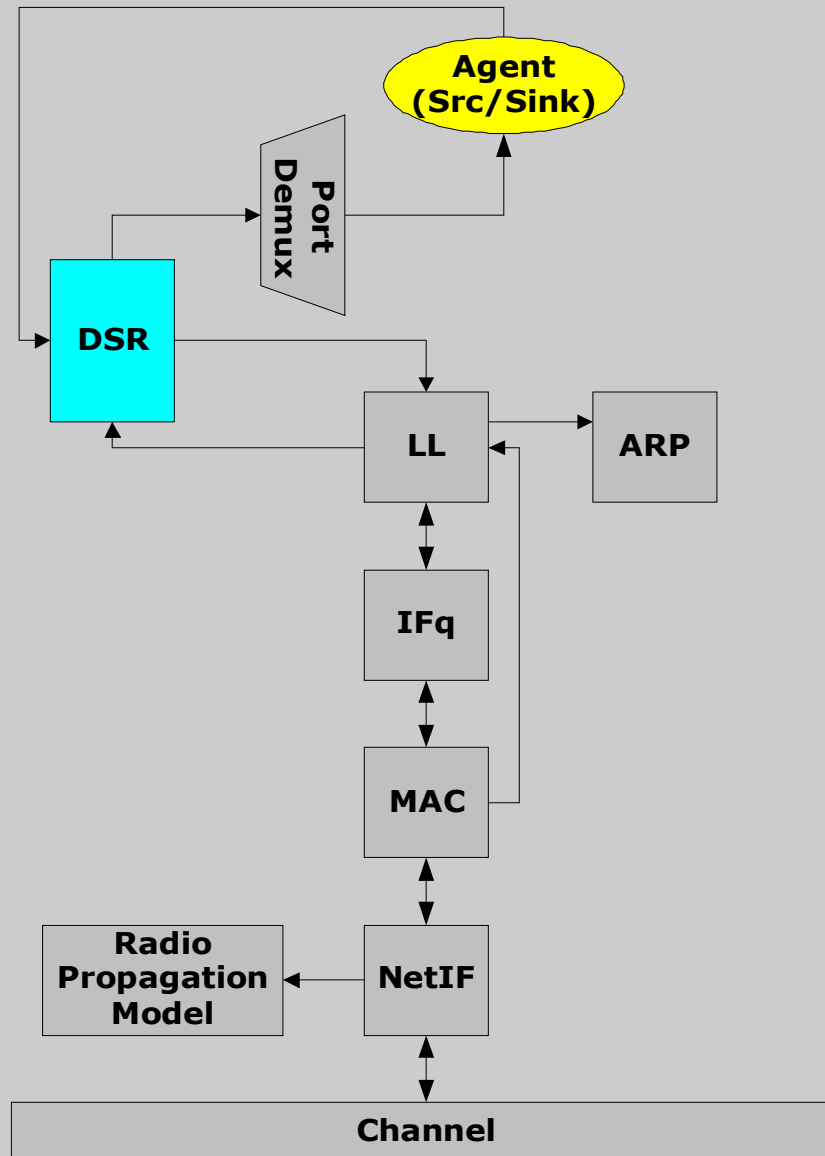
Mobile Node Modules (Continued)

- NetIF (Network Interfaces)
 - Hardware interface used by mobilenode to access the channel
 - Simulates signal integrity, collision, tx error
 - Mark each transmitted packet with transmission power, wavelength etc.
- Radio Propagation Model
 - Uses Friss-space attenuation($1/r^2$) at near distance and Two ray ground ($1/r^4$) at far distance
 - Decides whether the packet can be received by the mobilenode with given distance, transmit power and wavelength
 - Implements Omni Directional Antenna module which has unity gain for all direction

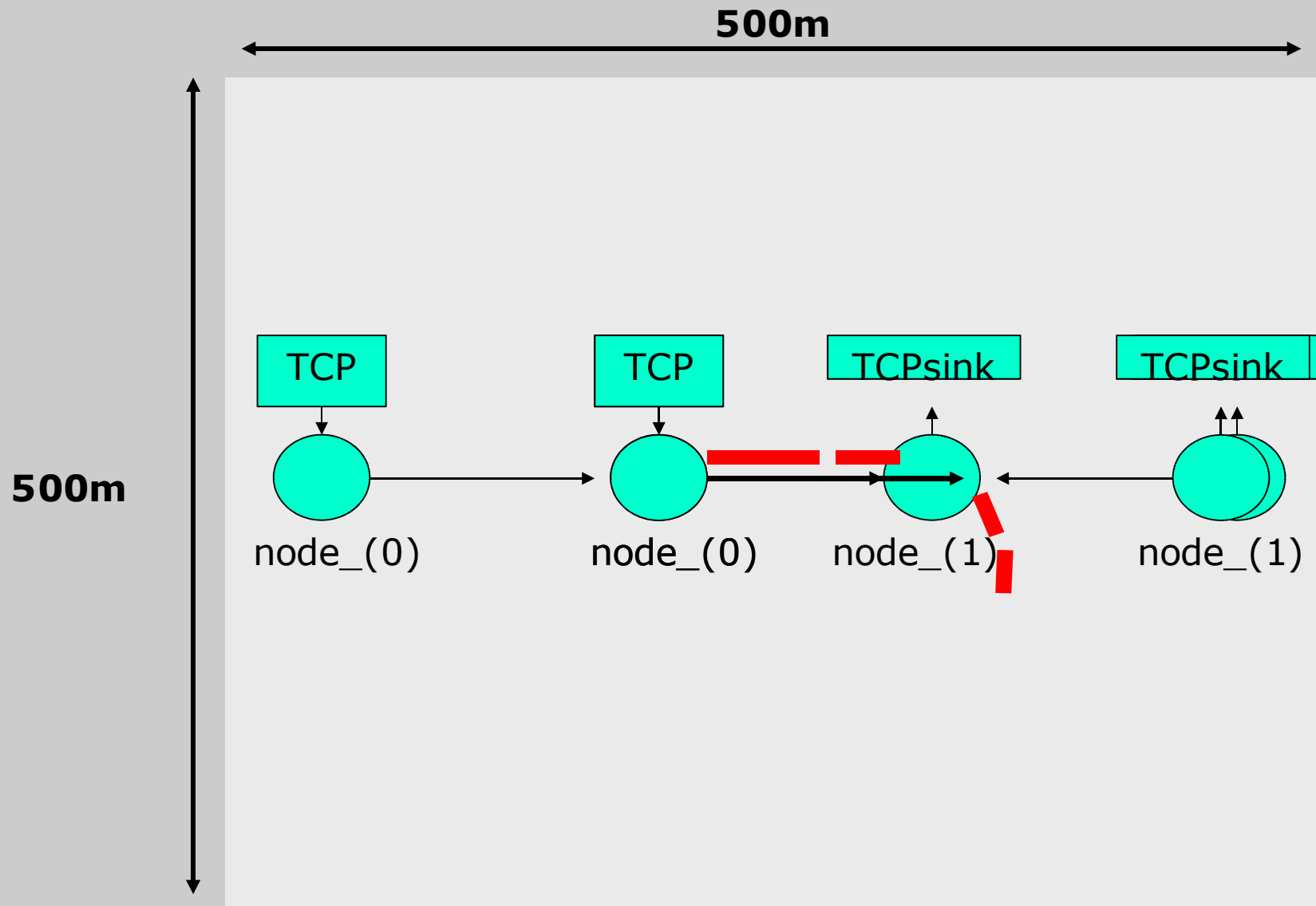
Wireless Simulation in ns-2 (Mobile Node Diagram - DSDV)



Wireless Simulation in ns-2 (Mobile Node Diagram - DSR)



Running a simulation-Scenario



Setting Up Variables

```
#=====
# Define options
#=====
set val(chan) Channel/WirelessChannel      ;# channel type
set val(prop) Propagation/TwoRayGround     ;# radio-propagation model
set val(ant) Antenna/OmniAntenna          ;# Antenna type
set val(ll) LL                             ;# Link layer type
set val(ifq) Queue/DropTail/PriQueue       ;# Interface queue type
set val(ifqlen) 50                         ;# max packet in ifq
set val(netif) Phy/WirelessPhy            ;# network interface type
set val(mac) Mac/802_11                   ;# MAC type
set val(rp) DSDV                           ;# ad-hoc routing protocol
set val(nn) 2                             ;# number of mobilenodes
```

Setting Up Variables

Create simulator object:

```
set ns_ [new Simulator]
```

Setup Trace File:

```
set tracefd [open simple.tr w]  
$ns_ trace-all $tracefd
```

Create Topography:

```
set topo [new Topography]  
$topo load_flatgrid 500 500
```

Create Object God:

```
create-god $val (nn)
```

Configuring Mobilenode

```
# Configure 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) \
    -topoInstance $topo \
    -channelType $val(chan) \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF
for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node ]
    $node_($i) random-motion 0 ;# disable random motion
}
```

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
```

Create Movement:

```
# Node_(1) starts to move towards node_(0)  
$ns_ at 50.0 "$node_(1) setdest 25.0 20.0 15.0"  
$ns_ at 10.0 "$node_(0) setdest 20.0 18.0 1.0"  
# Node_(1) then starts to move away from node_(0)  
$ns_ at 100.0 "$node_(1) setdest 490.0 480.0 15.0"
```

Setup traffic flow

```
set tcp [new Agent/TCP]
$tcp set class_2 set sink [new Agent/TCPSink]
$ns_ attach-agent $node_(0)
$tcp $ns_ attach-agent $node_(1)
$sink $ns_ connect $tcp
$sink set ftp [new Application/FTP]
$ftp attach-agent $tcp $ns_ at 10.0 "$ftp start"
```



Set Stop Time and Start Simulation

Set Simulation Stop Time:

```
for {set i 0} {$i < $val(nn)} {incr i} {  
  $ns_ at 150.0 "$node_($i) reset";  
}  
$ns_ at 150.0001 "stop"  
$ns_ at 150.0002 "puts \"NS EXITING...\" ;  
$ns_ halt" proc stop {} { global ns_ tracefd close $tracefd }
```

Finally, Start The Simulation:

```
puts "Starting Simulation..."  
$ns_ run
```

- cd into the directory and type “ns simple-wireless.tcl” to run the simulation
- The simulation will generate a trace file named “simple.tr”

Trace File

```
r 100.381997477 _1_ AGT --- 82 tcp 1060 [13a 1 0 800] ----- [0:0  
1:0 32 1] [32 0] 1 0
```

r:receive event,

1:node 1,

82:event(pkt) id,

1060:packet size,

13a(hex):expected duration of pkt transmission (not working),

1:sender mac id,

800:pkt type IP (806 for ARP),

1:0: receiver address:port#,

1: next hop address,

100.381997477:time stamps,

AGT:trace generated by agent,

tcp: tcp packet,

0:transmitter mac id,

0:0: sender address:port#

32: TTL

[32 0]: TCP sequence #, ack #

Reading Trace Files

- **New Wireless Trace file:**

**s -t 163.001503520 -Hs 0 -Hd -2 -Ni 0 -Nx 300.00 -Ny 500.00 -Nz 0.00
-Ne -1.000000 -NI AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id
2.0 -It cbr -Il 200 -If 1 -Ii 77 -Iv 32 -Pn cbr -Pi 32 -Pf 0 -Po 0**

Field 0: event type

s: send

r: receive

d: drop

f: forward

Field 1: General tag

-t: time

Field 3: Next hop info

-Hs: id for this node

-Hd: id for next hop towards the destination

Reading Trace Files

Field 4: Node property type tag

- Ni: node id**
- Nx -Ny -Nz: node's x/y/z coordinate**
- Ne: node energy level**
- Nl: trace level, such as AGT, RTR, MAC**
- Nw: reason for the event**

Field 5: Packet information at IP level

- Is: source address. Source port number**
- Id: dest address.dest port number**
- It: packet type**
- Il: packet size**
- If: flow id**
- Ii: unique id**
- Iv: ttl value**

Reading Trace Files

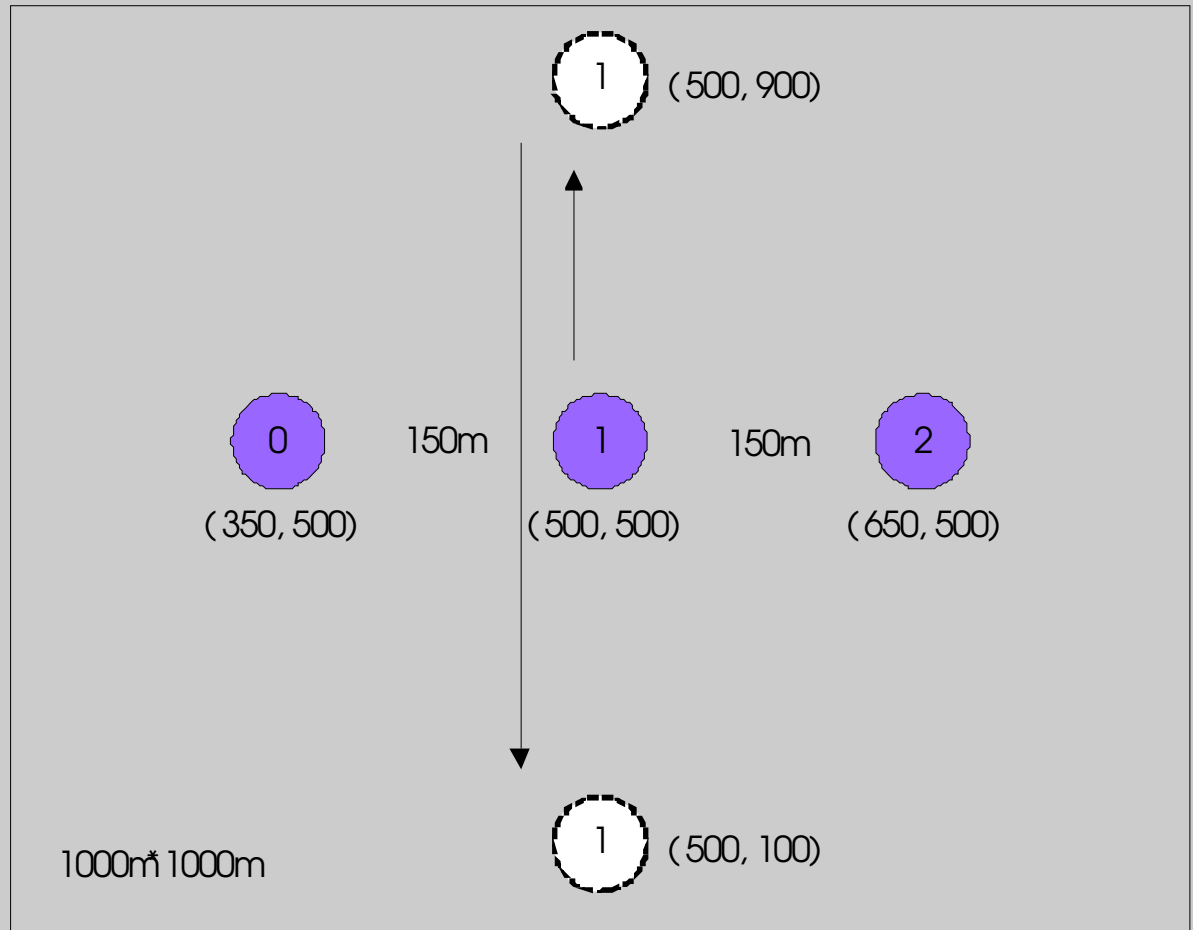
Field 6: packet info at MAC level

- Ma: duration**
- Md: dest's ethernet address**
- Ms: src's ethernet address**
- Mt: ethernet type**

Field 7: Packet info at “Application level” which consists of the type of application like arp, tcp, the type of adhoc routing protocol like DSDV, DSR, AODV etc. The field consists of a leading –P and the list of tags for different applications.

Simulation-Scenario II

- **1000m*1000m network with 3 node**
- **CBR/UDP traffic between node 0 to node 2**
- **At 200s, node 1 moves from (500, 500) to (500, 900), and then at 500s from (900, 500) to (500, 100).**
- **Simulation time 1000s.**



Setting Up Variables

Define options

```
# -----  
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(x)         1000                        ;# X dimension of topology  
set val(y)         1000                        ;# Y dimension of topology  
set val(cp)        ""                          ;# node movement model file  
set val(sc)        ""                          ;# traffic model file  
set val(ifqlen)    50                          ;# max packet in ifq  
set val(nn)        3                          ;# number of mobilenodes  
set val(seed)      0.0  
set val(stop)      1000.0                      ;# simulation time  
set val(tr)        exp.tr                      ;# trace file name  
set val(rp)        DSDV                       ;# routing protocol  
set AgentTrace    ON  
set RouterTrace   ON  
set MacTrace      OFF
```

Setting Up Variables

Initialize Global Variables

```
set ns_ [new Simulator]
$ns_ color 1 Blue
$ns_ color 2 Red
```

Open trace file

```
$ns_ use-newtrace ;# Use new trace format
set namfd [open nam-exp.tr w]
$ns_ namtrace-all-wireless $namfd $val(x) $val(y)
set tracefd [open $val(tr) w]
$ns_ trace-all $tracefd
```

set up topography object

```
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
```

create channel

```
set chan [new $val(chan)]
```

Create God

```
set god_ [create-god $val(nn)]
```

Configuring Mobilenodes

**# Create the specified number of mobile nodes [\$val(nn)] and "attach" them
to the channel. Three nodes are created : node(0), node(1) and node(2)**

```
$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) \  
    -channel $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
```


Configuring Movement

Provide initial (X,Y, for now Z=0) co-ordinates for mobilenodes

#

\$node_(0) set X_ 350.0

\$node_(0) set Y_ 500.0

\$node_(0) set Z_ 0.0

\$node_(1) set X_ 500.0

\$node_(1) set Y_ 500.0

\$node_(1) set Z_ 0.0

\$node_(2) set X_ 650.0

\$node_(2) set Y_ 500.0

\$node_(2) set Z_ 0.0

Load the god object with shortest hop information

#

\$god_ set-dist 1 2 1

\$god_ set-dist 0 2 2

\$god_ set-dist 0 1 1

Setup traffic flow

Now produce some simple node movements

Node_(1) starts to move upward and then downward

```
set god_ [God instance]
```

```
$ns_ at 200.0 "$node_(1) setdest 500.0 900.0 2.0"
```

```
$ns_ at 500.0 "$node_(1) setdest 500.0 100.0 2.0"
```

Setup traffic flow between nodes 0 connecting to 2 at time 100.0

```
set udp_(0) [new Agent/UDP]
```

```
$udp_(0) set fid_ 1
```

```
$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_ 200
```

```
$cbr_(0) set interval_ 2.0
```

```
$cbr_(0) set random_ 1
```

```
$cbr_(0) set maxpkts_ 10000
```

```
$cbr_(0) attach-agent $udp_(0)
```

```
$ns_ connect $udp_(0) $null_(0)
```

```
$ns_ at 100.0 "$cbr_(0) start"
```

Sewing Up

```
#Define node initial position in nam, only fro nam
for {set i 0} {$i < $val(nn)} {incr i} {
# The function must be called after mobility model is defined.
    $ns_ initial_node_pos $node_($i) 60
}
# Tell nodes when the simulation ends
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at $val(stop) "$node_($i) reset";
}
$ns_ at $val(stop) "stop"
$ns_ at $val(stop) "puts \"NS EXITING...\" ; $ns_ halt"
proc stop {} {
    global ns_ tracefd namfd
    $ns_ flush-trace
    close $tracefd
    close $namfd
}
puts $tracefd "M 0.0 nn $val(nn) x $val(x) y $val(y) rp $val(rp)"
puts $tracefd "M 0.0 sc $val(sc) cp $val(cp) seed $val(seed)"
puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"
puts "Starting Simulation..."
$ns_ run
```

Analysing Simulation

Routing message Updates

s -t 0.001560381 -Hs 0 -Hd -1 -Ni 0 -Nx 350.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -NI **RTR**
-Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.255 -Id -1.255 -It **message** -Il 32 -If 0 -Ii 0 -Iv 32

r -t 0.002332881 -Hs 1 -Hd -1 -Ni 1 -Nx 500.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -NI **RTR**
-Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It **message** -Il 32 -If 0 -Ii
0 -Iv 32

s -t 0.036574082 -Hs 2 -Hd -1 -Ni 2 -Nx 650.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -NI **RTR**
-Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 2.255 -Id -1.255 -It **message** -Il 32 -If 0 -Ii 1 -Iv 32

r -t 0.037706582 -Hs 1 -Hd -1 -Ni 1 -Nx 500.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -NI **RTR**
-Nw --- -Ma 0 -Md ffffffff -Ms 2 -Mt 800 -Is 2.255 -Id -1.255 -It **message** -Il 32 -If 0 -Ii
1 -Iv 32

s -t 1.120124667 -Hs 1 -Hd -1 -Ni 1 -Nx 500.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -NI **RTR**
-Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.255 -Id -1.255 -It **message** -Il 32 -If 0 -Ii 2 -Iv 32

r -t 1.121437167 -Hs 2 -Hd -1 -Ni 2 -Nx 650.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -NI **RTR**
-Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It **message** -Il 32 -If 0 -Ii
2 -Iv 32

r -t 1.121437167 -Hs 0 -Hd -1 -Ni 0 -Nx 350.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -NI **RTR**
-Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It **message** -Il 32 -If 0 -Ii
2 -Iv 32

Analysing Simulation Packet Transmission

s -t 100.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 350.00 -Ny 500.00 -Nz 0.00 -Ne
-1.000000 -NI AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 2.0 -It cbr
-Il 200 -If 1 -Ii 30 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 3

r -t 100.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 350.00 -Ny 500.00 -Nz 0.00 -Ne
-1.000000 -NI RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 2.0 -It cbr
-Il 200 -If 1 -Ii 30 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 3

s -t 100.000000000 -Hs 0 -Hd 1 -Ni 0 -Nx 350.00 -Ny 500.00 -Nz 0.00 -Ne
-1.000000 -NI RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 2.0 -It cbr
-Il 220 -If 1 -Ii 30 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 3

r -t 100.005990500 -Hs 1 -Hd 1 -Ni 1 -Nx 500.00 -Ny 500.00 -Nz 0.00 -Ne
-1.000000 -NI RTR -Nw --- -Ma 13a -Md 1 -Ms 0 -Mt 800 -Is 0.0 -Id 2.0 -It
cbr -Il 220 -If 1 -Ii 30 -Iv 32 -Pn cbr -Pi 0 -Pf 1 -Po 3

f -t 100.005990500 -Hs 1 -Hd 2 -Ni 1 -Nx 500.00 -Ny 500.00 -Nz 0.00 -Ne
-1.000000 -NI RTR -Nw --- -Ma 13a -Md 1 -Ms 0 -Mt 800 -Is 0.0 -Id 2.0 -It
cbr -Il 220 -If 1 -Ii 30 -Iv 31 -Pn cbr -Pi 0 -Pf 1 -Po 3

r -t 100.011765000 -Hs 2 -Hd 2 -Ni 2 -Nx 650.00 -Ny 500.00 -Nz 0.00 -Ne
-1.000000 -NI AGT -Nw --- -Ma 13a -Md 2 -Ms 1 -Mt 800 -Is 0.0 -Id 2.0 -It
cbr -Il 220 -If 1 -Ii 30 -Iv 31 -Pn cbr -Pi 0 -Pf 2 -Po 3

Analysing Simulation

Node movement and Packet Drop

- **M 200.00000 1 (500.00, 500.00, 0.00), (500.00, 900.00), 2.00**
- **d -t 302.284948727 -Hs 0 -Hd 1 -Ni 0 -Nx 350.00 -Ny 500.00 -Nz 0.00
-Ne -1.000000 -Nl RTR -Nw CBK -Ma 13a -Md 1 -Ms 0 -Mt 800 -Is
0.0 -Id 2.0 -It cbr -Il 220 -If 1 -Ii 174 -Iv 32 -Pn cbr -Pi 98 -Pf 0 -Po 3**

Thank you !