

# EEC 687/787 Mobile Computing (Spring, 2008)

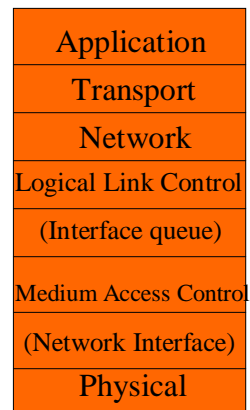
## Ns-2 Laboratory #2

Chansu Yu

Cleveland State University

## 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.
- ❑ RAgent (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



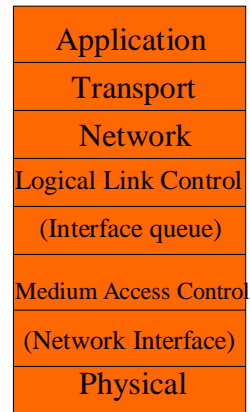
## Mobile Node Modules (Continued)

### ❑ 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



3

*c.yu91@csuohio.edu*

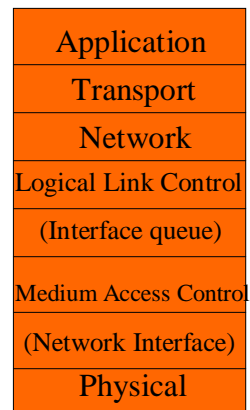
## Mobile Node Modules (Continued)

### ❑ 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

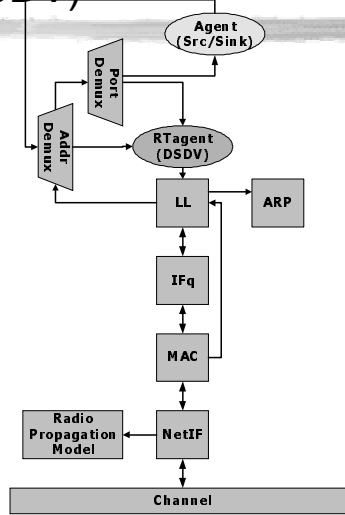
- Uses Friss-space attenuation( $1/r^2$ ) at near distance and Two ray ground ( $1/r^4$ ) at far distance
- Decides whether the packet can be received by the mobilenode with given distance, transmit power and wavelength
- Implements Omni Directional Antenna module which has unity gain for all direction



4

*c.yu91@csuohio.edu*

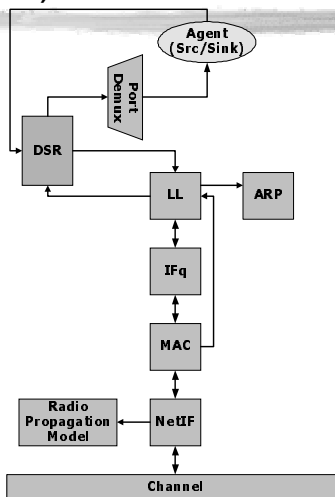
## Wireless Simulation in ns-2 (Mobile Node Diagram - DSDV)



5

*c.yu91@csuohio.edu*

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



6

*c.yu91@csuohio.edu*

## Run the Simulation

- Download ns-2 for the impatient  
(<http://icapeople.epfl.ch/aad/teaching/ns/ns.html>)

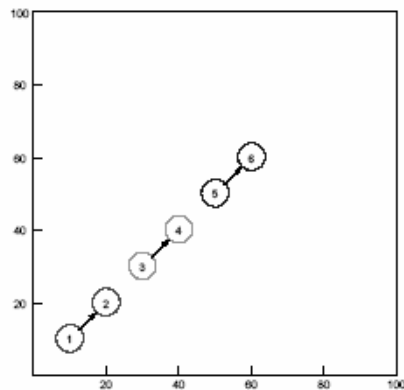
➤ [ex6sta.tcl](#)

➤ [fil.awk](#), [fil2.awk](#), [fil4.awk](#), [fil6.awk](#)

7

*c.yu91@csuohio.edu*

## Mobile Network Simulation with ns-2



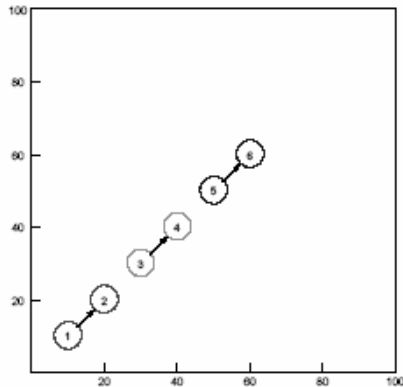
- 6 nodes
- Located (10,10), (20,20), ...
- CBR (Constant bit rate) traffic based on UDP 1→2, 3→4, 5→6

8

*c.yu91@csuohio.edu*

# Mobile Network Simulation with ns-2

```
set ns_ [new Simulator]
set opt(nn) 6
```



- 6 nodes

```
set WT1 [$ns_ node $1]
```

...

```
set WT6 [$ns_ node $6]
```

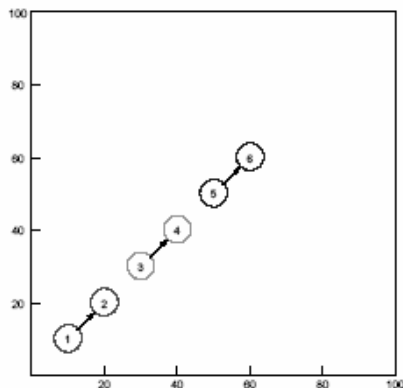
=>

```
for {set i 1} {$i<=$opt(nn)} {incr i} {
    set WT($i) [$ns_ node $i]
}
```

9

# Mobile Network Simulation with ns-2

- Located (10,10), (20,20), ...



```
$WT1 set X_ 10
```

```
$WT1 set Y_ 10
```

```
$WT1 set Z_ 0.0
```

...

```
$WT6 set X_ 60
```

```
$WT6 set Y_ 60
```

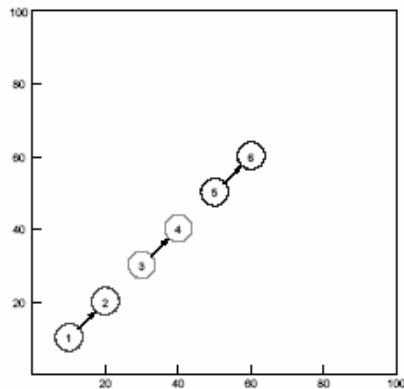
```
$WT6 set Z_ 0.0
```

=>

```
for {set i 1} {$i<=$opt(nn)} {incr i} {
    $WT($i) set X_ [expr 10*$i]
    $WT($i) set Y_ [expr 10*$i]
    $WT($i) set Z_ 0.0
}
```

}10

## Mobile Network Simulation with ns-2



- CBR (Constant bit rate) traffic based on UDP 1→2, 3→4, 5→6

```
set udp1 [new Agent/UDP]
$ns_ attach-agent $WT1 $udp1
```

```
set sink1 [new Agent/Null]
$ns_ attach-agent $WT2 $sink1
```

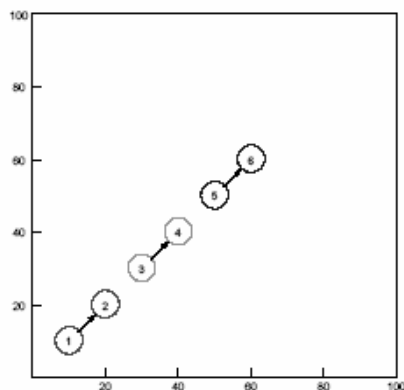
```
$ns_ connect $udp1 $sink1
```

...

11

c.yu91@csuohio.edu

## Mobile Network Simulation with ns-2



- CBR (Constant bit rate) traffic based on UDP 1→2, 3→4, 5→6

```
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 1000
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
```

```
$ns_ at 20.0 "$cbr1 start"
```

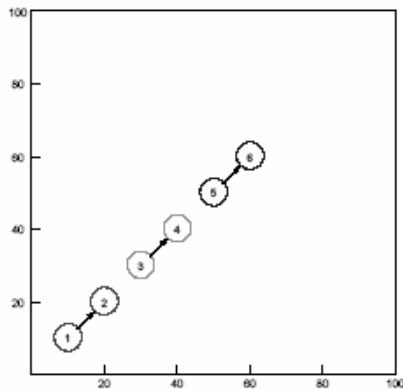
```
$ns_ at 150.0 "$cbr1 stop"
```

...

12

c.yu91@csuohio.edu

# Mobile Network Simulation with ns-2



•CBR (Constant bit rate) traffic based on UDP 1→2, 3→4, 5→6

```
for {set i 1} {$i <= $opt(nn)} {incr i 2} {
  set udp($i) [new Agent/UDP]
  $ns_ attach-agent $WT($i) $udp($i)

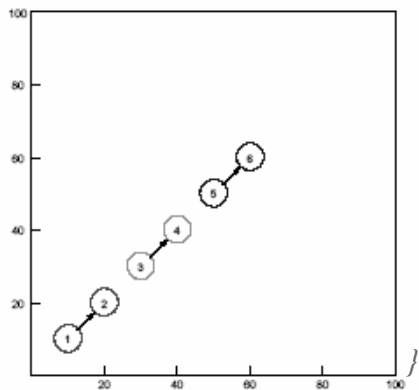
  set sink($i) [new Agent/Null]
  $ns_ attach-agent $WT([expr $i+1]) $sink($i)

  $ns_ connect $udp($i) $sink($i)
}
```

13

c.yu91@csuohio.edu

# Mobile Network Simulation with ns-2



•CBR (Constant bit rate) traffic based on UDP 1→2, 3→4, 5→6

```
set cbr($i) [new Application/Traffic/CBR]
$cbr($i) set packetSize_ 1000
$cbr($i) set interval_ 0.005
$cbr($i) attach-agent $udp($i)
```

```
$ns_ [expr 20.0*$i] "$cbr($i) start"
$ns_ at $opt(stop) "$cbr($i) stop"
```

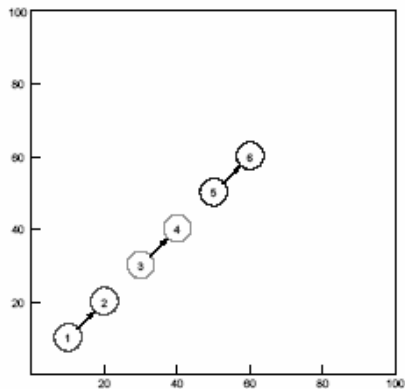
set opt(stop) 150

14

c.yu91@csuohio.edu

# Mobile Network Simulation with ns-2

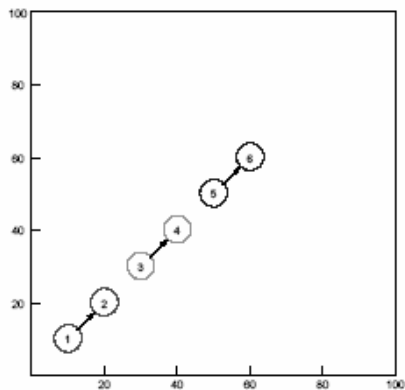
- Node configuration



```
$ns_ node-config
-adhocRouting DumbAgent \
-llType LL \
-macType Mac/802_11 \
-ifqType Queue/DropTail/PriQueue \
-ifqLen 50 \
-antType Antenna/OmniAntenna \
-propType Propagation/FreeSpace \
-phyType Phy/WirelessPhy \
-channelType Channel/WirelessChannel \
-topoInstance $topo \
```

# Mobile Network Simulation with ns-2

- Topology configuration



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

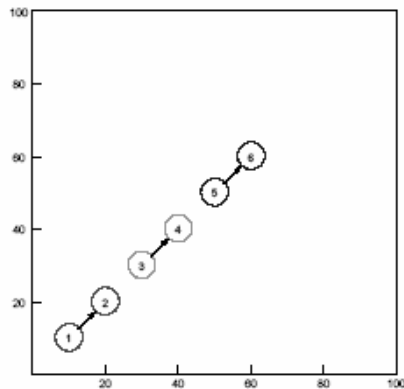
- GOD (General operations director)

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

```
set opt(x) 100
set opt(y) 100
```

# Mobile Network Simulation with ns-2

- Node configuration (cont'd)



```
$ns_ node-config  
...  
-agentTrace ON \  
-routerTrace OFF \  
-macTrace OFF \  
-movementTrace OFF
```

```
set tracefd [open $opt(tr) w]  
$ns_ trace-all $tracefd
```

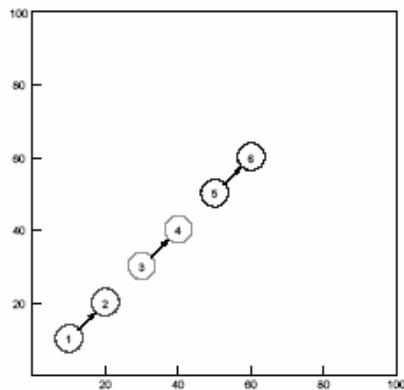
*set opt(tr) out.tr*

17

*c.yu91@csuohio.edu*

# Mobile Network Simulation with ns-2

- End procedure & Start ns2 simulation



```
for {set i} {$i <= $opt(nn)} {incr i} {  
  $ns_ at $opt(stop).0000010 "$WT($i) reset";  
}
```

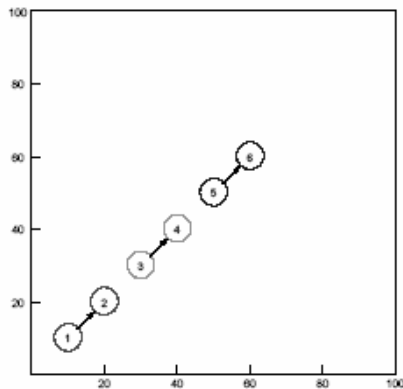
```
$ns_ at $opt(stop).2 "$ns_ halt"
```

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

18

*c.yu91@csuohio.edu*

# Mobile Network Simulation with ns-2



- Adding finish procedure

```
$ns_ at $opt(stop).1 "finish"  
$ns_ at $opt(stop).2 "$ns_ halt"  
  
proc finish {} {  
  for {set i 2} {$i <=6 } {incr i 2} {  
    exec rm -f out$i.xgr  
    exec awk -f fil$i.awk out.tr > out$i.xgr  
  }  
  exec xgraph out2.xgr out4.xgr out6.xgr &  
  puts "Finishing ns.."  
  exit 0  
}
```

19

c.yu91@csuohio.edu

## Summarize the Results

- Ns2 manual, “Trace and Monitoring support,” Ch. 26
- Gets two files (if specified)
- out-test.tr
  - use awk script to summarize the results
- nam-out-test.nam
  - What is this?

20

c.yu91@csuohio.edu

## Visualize the Simulation Runs (nam file)

- nam-out-test.nam
  - Nam: network animator
  
- Try this
  - `nam ../edu/rts-cts/rts-cts-data-ack.nam`
  
- Not really useful, though

21

*c.yu91@csuohio.edu*

## Summarize the Results (trace file)

- Trace file
  
- Linux commands to quickly get results
  
- Awk script to summarize results

22

*c.yu91@csuohio.edu*

## Trace File Format

Nodes are numbered  
as 0, 1, 2, ... inside ns2.  
I.e., node WT1 is node 0.

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

<b>r</b> :receive event,	<b>100.381997477</b> :time stamps,
<b>_1_</b> :node 1,	<b>AGT</b> :trace generated by agent,
<b>82</b> :event(pkt) id,	<b>cbr</b> : cbr packet,
<b>1060</b> :packet size,	
<b>13a(hex)</b> :expected duration of pkt transmission (not working),	
<b>1</b> :sender mac id,	<b>0</b> :transmitter mac id,
<b>800</b> :pkt type IP (806 for ARP),	<b>0:0</b> : sender address:port#
<b>1:0</b> : receiver address:port#,	<b>32</b> : TTL
<b>1</b> : next hop address,	<b>[32 0]</b> : TCP sequence #, ack #

23

c.yu91@csuohio.edu

## Summarize Trace File

### Using simple Linux commands

cat, grep, wc, |, >, >>, etc.

eg. Calculate **packet delivery ratio** from a trace file (out.tr)

```
cat out.tr | grep AGT | grep cbr | grep ^s | wc -l  
cat out.tr | grep AGT | grep cbr | grep ^r | wc -l
```

```
cat out.tr | grep AGT | grep cbr | grep ^s | grep _0_ | wc -l  
cat out.tr | grep AGT | grep cbr | grep ^r | grep _1_ | wc -l
```

### Simple programming

shell, awk, etc.

### Advanced programming

C/C++, Java, VB, etc.

Base 0

24

c.yu91@csuohio.edu

## Lab Exercise #2 (A)

- Overall PDR?
  
- PDR for WT1-WT2 connection?
- PDR for WT3-WT4 connection?
- PDR for WT5-WT6 connection?
  
- Are they very different? If yes, why?

25

*c.yu91@csuohio.edu*

## awk

- Calling format
  - `awk '/pattern-to-match/ {program to run}' trace-file`  
eg. `awk '$1 == "s" {print}' aaa.tr`
  - `awk -f awk-script trace-file`
- Characteristics
  - Flexible (C style)
  - Simple (no pointers, no references)
  - Powerful
    - Float calculation
    - Automatic data type assignment and check
    - Branch/Loop control
    - Function call

26

*c.yu91@csuohio.edu*

## awk (cont.)

### □ Script structure

#### ➤ Initialization

```
BEGIN { ... }
```

#### ➤ Body

```
{ ... }
```

Important: Every row in the trace file is scanned by the commands in the body part one time, just ONE time.

#### ➤ Summarization

```
END { ... }
```

27

c.yu91@csuohio.edu

## An Example Of awk Script (fil2.awk)

```
BEGIN{ sim_end = 200;
  i=0;
  while (i<=sim_end) { sec[i]=0; i+=1; }
}

{ if ($1=="r" && $7=="cbr" && $3=="_1_") {
  sec[int($2)] += $8;
};
}

END{ i=0;
  while (i<=sim_end) {
  print i " " sec[i]*8;
  i+=1;
};
}
```

*Every line in the trace file  
will be processed using the  
awk commands in the body.*

***What is \$8?  
What does it show?  
What should be changed  
for fil4.awk & fil6.awk?***

28

c.yu91@csuohio.edu

## Working with awk Scripts

```
awk -f fil2.awk out.tr > out2.xgr
```

```
awk -f fil4.awk out.tr > out4.xgr
```

```
awk -f fil5.awk out.tr > out6.xgr
```

```
xgraph out2.xgr out4.xgr out6.xgr &
```



29

*c.yu91@csuohio.edu*

## Lab Exercise #2 (B)

- cp fil.awk pdr.awk

- Modify pdr.awk to calculate PDR

- I.e., “awk -f pdr.awk out.tr” prints PDR.

30

*c.yu91@csuohio.edu*

## Another awk Script: How to get Delay?

```
BEGIN {
    nSentPackets = 0 ;
    nReceivedPackets = 0 ;
    rTotalDelay = 0.0 ;
}

{
    strEvent = $1 ;           rTime = $2 ;
    strAgt = $4 ;           idPacket = $6 ;
    strType = $7 ;
```

31

*c.yu91@csuohio.edu*

## Another awk Script: How to get Delay?

```
if ( strAgt == "AGT" && strType == "cbr" ) {
    if ( strEvent == "s" ) {
        nSentPackets += 1 ;
        rSentTime[ idPacket ] = rTime ;
    }
    if ( strEvent == "r" ) {
        nReceivedPackets += 1 ;
        rReceivedTime[ idPacket ] = rTime ;
        rTotalDelay += rReceivedTime[ idPacket ] - rSentTime[ idPacket ] ;
    }
}

END {
    rTime = rEndTime - rStartTime ;
    rPacketDeliveryRatio = nReceivedPackets / nSentPackets * 100 ;
    if ( nReceivedPackets != 0 ) rAverageDelay = rTotalDelay / nReceivedPackets ;
    printf( "AverageDelay: %15.5f PacketDeliveryRatio: %10.2f\n", rAverageDelay,
    rPacketDeliveryRatio ) ;
}
```

32

*c.yu91@csuohio.edu*

## Lab Exercise #2 (C)

- Create delay.awk
- cp delay.awk delay2.awk
- Modify delay2.awk to measure the average delay for the first traffic stream (WT1->WT2)
- Average delay is calculated over the entire simulation time. Develop delay\_all.awk to measure the average delay only when there exist three traffic streams (time=60~150)

33

*c.yu91@csuohio.edu*

## Lab Report #2

- A
  - Overall PDR?
  - PDR for WT1-WT2, WT3-WT4, and WT5-WT6 connection?
  - Are they very different? If yes, why?
- B
  - Send pdr.awk
- C
  - Send delay.awk and result
  - Send delay2.awk with result
  - Send delay\_all.awk with result
- What you have learned

34

*c.yu91@csuohio.edu*

## ns Tutorials

---

- ❑ NS website <http://www.isi.edu/nsnam/ns/>
- ❑ NS Manual [http://www.isi.edu/nsnam/ns/doc/ns\\_doc.pdf](http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf)
- ❑ Marc Greis's Tutorial <http://www.isi.edu/nsnam/ns/tutorial>
- ❑ <http://www.cs.virginia.edu/~cs757/slidespdf/cs757-ns2-tutorial-exercise1.pdf>
- ❑ <http://nile.wpi.edu/NS/>
- ❑ [http://nesl.ee.ucla.edu/courses/ee206a/2002s/guest\\_presentations/GP02\\_Park\\_ns2.ppt](http://nesl.ee.ucla.edu/courses/ee206a/2002s/guest_presentations/GP02_Park_ns2.ppt)
- ❑ [http://www.ece.ubc.ca/~elec565/ns2\\_tutorial.ppt](http://www.ece.ubc.ca/~elec565/ns2_tutorial.ppt)