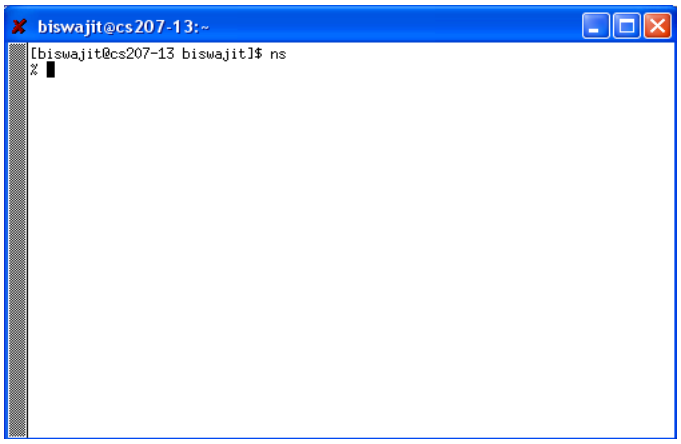
Using Ns2

Step :1 (Make sure that Ns2 is working from your login)
If you write ns, it should show % (as shows in the screen). Use Xwin32 for graphical support.



Step :2 (Write tcl script for your application)
You need to save this tcl script (filename.tcl) in your account, you can do ftp to transfer your tcl script to the Ns2 server. To know more about the syntax of tcl use this link http://hegel.ittc.ukans.edu/topics/tcltk/index.html

Step :3 (Understanding a tcl script for "Mobile ad hoc network")
a. Following is a tcl script for mobile ad hoc network (note: This tcl script is not created by me, but it works fine and a very good example to start with). Please note one important thing in this tcl script "$ns_ use-newtrace", this will give detailed trace file.

b. To get a detailed explanation of running a wireless simulation use this link http://www.isi.edu/nsnam/ns/tutorial/nsscript5.html

```
# ======================================================================
# 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(ifqlen)         50                         ;# max packet in ifq
set val(nn)             27                         ;# number of
                                                     #mobile nodes

set val(rp)             AODV   ;# this is for AODV routing protocol,
```

```
                                #you can use DSDV, TORA, etc.

set val(x)              500     # Area
set val(y)              500

proc usage {} {
    puts {cbr_mobile: Usage> ns simple_manet.tcl [manet
<DSR,AODV,TORA,OLSR,NRLOLSR,others> }
    puts {PARAMETERS NEED NOT BE SPECIFIED... DEFAULTS WILL BE USED}
    exit
}

set state flag
foreach arg $argv {
        switch -- $state {
                flag {
                switch -- $arg {
                        manet   {set state manet}
                        help    {usage}
                        default {error "unknown flag $arg"}
                }
                }
                manet   {set state flag; set val(rp) $arg}
        }
}

# ======================================================================
# Main Program
# ======================================================================

#
# Initialize Global Variables
#

set ns_         [new Simulator]

$ns_ use-newtrace # Please use this trace command to get the latest
                  #format in you trace file

# Create trace file
set tracefd     [open simple_manet.tr w]

# Write into trace file
$ns_ trace-all $tracefd

#Create Nam file
set namtrace [open simple_manet.nam w]

# Write into nam file
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

# Set up topography object
set topo       [new Topography]

$topo load_flatgrid 500 500
```

```
# Create God

create-god $val(nn)

#Create the specified number of mobilenodes [$val(nn)] and "attach"
#them to the channel.

# Configure node using the parameters specified in "define options"
set chan_1_ [new $val(chan)]

        $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_1_ \
                         -topoInstance $topo \
                         -agentTrace ON \
                         -routerTrace ON \
                         -macTrace OFF \
                         -movementTrace ON

        # Creating nodes using above parameters
         for {set i 0} {$i < $val(nn) } {incr i} {
                 set node_($i) [$ns_ node]
                 $node_($i) random-motion 1
                                 ;# enable random motion
         }



# Provide initial (X,Y, for now Z=0) co-ordinates for mobilenodes
# Here node_(0) is the source.
$node_(0) set X_ 50.0
$node_(0) set Y_ 50.0

$node_(1) set X_ 1.0
$node_(1) set Y_ 1.0
$node_(2) set X_ 1.0
$node_(2) set Y_ 100.0
$node_(3) set X_ 1.0
$node_(3) set Y_ 200.0
$node_(4) set X_ 1.0
$node_(4) set Y_ 300.0
$node_(5) set X_ 1.0
$node_(5) set Y_ 400.0
$node_(6) set X_ 100.0
$node_(6) set Y_ 1.0
$node_(7) set X_ 100.0
$node_(7) set Y_ 100.0
$node_(8) set X_ 100.0
$node_(8) set Y_ 200.0
$node_(9) set X_ 100.0
$node_(9) set Y_ 300.0
```

```
$node_(10) set X_ 100.0
$node_(10) set Y_ 400.0
$node_(11) set X_ 200.0
$node_(11) set Y_ 1.0
$node_(12) set X_ 200.0
$node_(12) set Y_ 100.0
$node_(13) set X_ 200.0
$node_(13) set Y_ 200.0
$node_(14) set X_ 200.0
$node_(14) set Y_ 300.0
$node_(15) set X_ 200.0
$node_(15) set Y_ 400.0
$node_(16) set X_ 300.0
$node_(16) set Y_ 1.0
$node_(17) set X_ 300.0
$node_(17) set Y_ 100.0
$node_(18) set X_ 300.0
$node_(18) set Y_ 200.0
$node_(19) set X_ 300.0
$node_(19) set Y_ 300.0
$node_(20) set X_ 300.0
$node_(20) set Y_ 400.0
$node_(21) set X_ 400.0
$node_(21) set Y_ 1.0
$node_(22) set X_ 400.0
$node_(22) set Y_ 100.0
$node_(23) set X_ 400.0
$node_(23) set Y_ 200.0
$node_(24) set X_ 400.0
$node_(24) set Y_ 300.0
$node_(25) set X_ 400.0
$node_(25) set Y_ 400.0

# Node_(26) is the data collection point
$node_(26) set X_ 450.0
$node_(26) set Y_ 450.0

# Set the color of the nodes
$ns_ at .01 "$node_(1) color blue"
$ns_ at .01 "$node_(26) color yellow"

#Set destination format is "setdest <x> <y> <speed>"

# node_(0) is the phenominon.
$ns_ at 0.01 "$node_(0) setdest 50.0 50.0 0.0"
$ns_ at 5.0 "$node_(0) setdest 350.0 350.0 0.0"
$ns_ at 6.0 "$node_(0) setdest 1.0 350.0 0.0"
$ns_ at 7.0 "$node_(0) setdest 50.0 50.0 0.0"

$ns_ at 0.01 "$node_(1) setdest 1.0 1.0 0.0"
$ns_ at 0.01 "$node_(2) setdest 1.0 100.0 0.0"
$ns_ at 0.01 "$node_(3) setdest 1.0 200.0 0.0"
$ns_ at 0.01 "$node_(4) setdest 1.0 300.0 0.0"
$ns_ at 0.01 "$node_(5) setdest 1.0 400.0 0.0"
$ns_ at 0.01 "$node_(6) setdest 100.0 1.0 0.0"
$ns_ at 0.01 "$node_(7) setdest 100.0 100.0 0.0"
$ns_ at 0.01 "$node_(8) setdest 100.0 200.0 0.0"
```

```
$ns_ at 0.01 "$node_(9) setdest 100.0 300.0 0.0"
$ns_ at 0.01 "$node_(10) setdest 100.0 400.0 0.0"
$ns_ at 0.01 "$node_(11) setdest 200.0 1.0 0.0"
$ns_ at 0.01 "$node_(12) setdest 200.0 100.0 0.0"
$ns_ at 0.01 "$node_(13) setdest 200.0 200.0 0.0"
$ns_ at 0.01 "$node_(14) setdest 200.0 300.0 0.0"
$ns_ at 0.01 "$node_(15) setdest 200.0 400.0 0.0"
$ns_ at 0.01 "$node_(16) setdest 300.0 1.0 0.0"
$ns_ at 0.01 "$node_(17) setdest 300.0 100.0 0.0"
$ns_ at 0.01 "$node_(18) setdest 300.0 200.0 0.0"
$ns_ at 0.01 "$node_(19) setdest 300.0 300.0 0.0"
$ns_ at 0.01 "$node_(20) setdest 300.0 400.0 0.0"
$ns_ at 0.01 "$node_(21) setdest 400.0 1.0 0.0"
$ns_ at 0.01 "$node_(22) setdest 400.0 100.0 0.0"
$ns_ at 0.01 "$node_(23) setdest 400.0 200.0 0.0"
$ns_ at 0.01 "$node_(24) setdest 400.0 300.0 0.0"
$ns_ at 0.01 "$node_(25) setdest 400.0 400.0 0.0"
$ns_ at 0.01 "$node_(26) setdest 450.0 450.0 0.0"


#Set Udp, cbr agent and attach those with nodes

set udp1 [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp1
$udp1 set class_ 0

#$udp1 set fid_ 2
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$cbr1 set packetSize_ 1000
$cbr1 set interval_ 0.02

# Attach null agent for sink
set null1 [new Agent/Null]
$ns_ attach-agent $node_(26) $null1
$ns_ connect $udp1 $null1

# Start the traffic generator
$ns_ at 2.0 "$cbr1 start"

# Setting the time to stop the simulation
$ns_ at 160.0 "stop"
$ns_ at 160.01 "puts \"NS EXITING...\" ; $ns_ halt"

proc stop {} {
    global ns_ tracefd namtrace
    $ns_ flush-trace
    close $tracefd
    close $namtrace
}

puts "Starting Simulation..."

# Run the simulation
$ns_ run
```

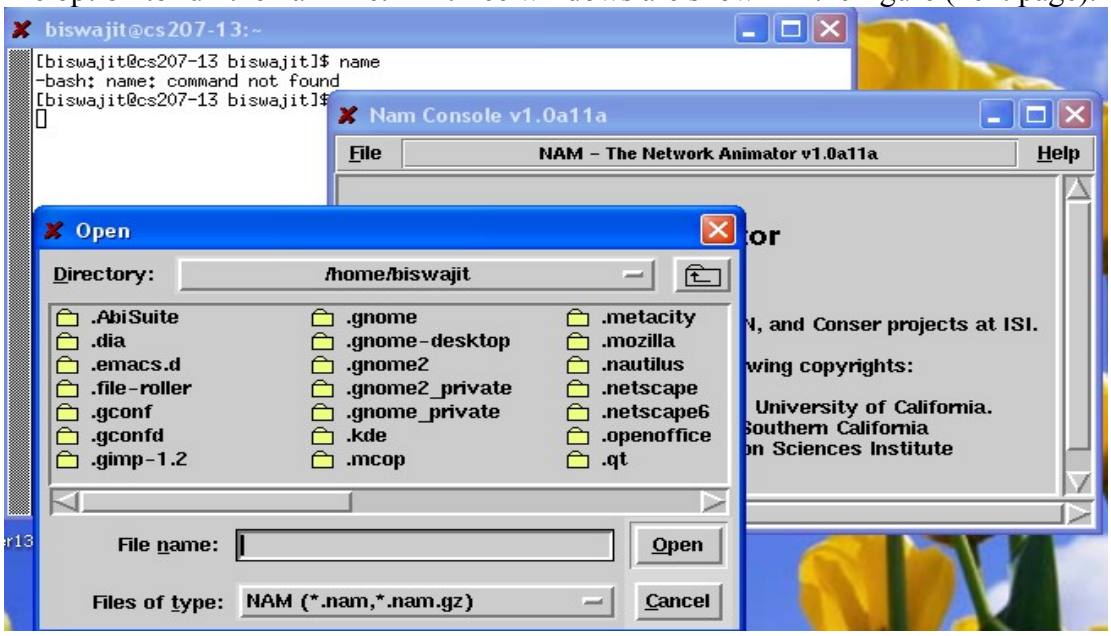
Step 4 (To analyze the trace file or to run nam)

a. An example of the trace format is shown below:

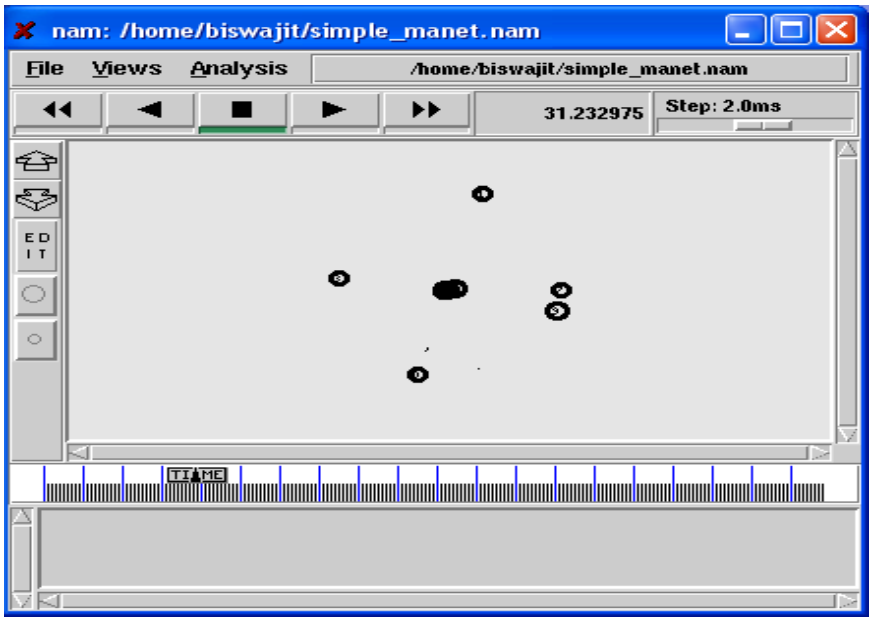
```
s -t 0.267662078 -Hs 0 -Hd -1 -Ni 0 -Nx 5.00 -Ny 2.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.255 -Id -1.255
-It
message -Il 32 -If 0 -Ii 0 -Iv 32
s -t 1.511681090 -Hs 1 -Hd -1 -Ni 1 -Nx 390.00 -Ny 385.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.255 -Id -1.255
-It
message -Il 32 -If 0 -Ii 1 -Iv 32
s -t 10.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 5.00 -Ny 2.00 -Nz 0.00 -Ne
-1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 1.0 -It
tcp -Il 1000 -If
2 -Ii 2 -Iv 32 -Pn tcp -Ps 0 -Pa 0 -Pf 0 -Po 0
r -t 10.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 5.00 -Ny 2.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 1.0 -It
tcp -Il 1000 -If
2 -Ii 2 -Iv 32 -Pn tcp -Ps 0 -Pa 0 -Pf 0 -Po 0
r -t 100.004776054 -Hs 1 -Hd 1 -Ni 1 -Nx 25.05 -Ny 20.05 -Nz 0.00 -Ne
-1.000000 -Nl AGT -Nw --- -Ma a2 -Md 1 -Ms 0 -Mt 800 -Is 0.0 -Id 1.0 -It
tcp -Il 1020 -If 2 -Ii 21 -Iv 32 -Pn tcp -Ps 0 -Pa 0 -Pf 1 -Po 0
s -t 100.004776054 -Hs 1 -Hd -2 -Ni 1 -Nx 25.05 -Ny 20.05 -Nz 0.00 -Ne
-1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.0 -Id 0.0 -It
ack -Il 40
-If 2 -Ii 22 -Iv 32 -Pn tcp -Ps 0 -Pa 0 -Pf 0 -Po 0
```

b. Explanation of the trace file can be found at
http://www.isi.edu/nsnam/ns/doc/node186.html

C. To run .nam file, write nam in your login, it will pop up Nam Console, in that click on file option to run the nam file. All three windows are shown in the figure (next page).



The nam window looks like this:-



<u>How to analyze the routing of AODV using the trace file?</u>

If you run the above tcl script, you will get a trace file part of which looks like this:

```
M 0.01000 0 (50.00, 50.00, 0.00), (50.00, 50.00), 0.00 –
```

This shows the initial and final position a node, and it's not moving as speed is set at 0.

```
s -t 2.000000000 -Hs 1 -Hd -2 -Ni 1 -Nx 1.00 -Ny 1.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.255 -Id -1.255
-It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 26
-Pds 0 -Ps 1 -Pss 4 -Pc REQUEST
```

-----In this, node 1 is broadcasting a request to establish a path. Hd -2 means no destination hop found yet.

```
r -t 2.001360231 -Hs 0 -Hd -2 -Ni 0 -Nx 50.00 -Ny 50.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255
-Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb
1 -Pd 26 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST
```

------This says node 2 receives a broadcasted request.

```
f -t 2.150459666 -Hs 0 -Hd 13 -Ni 0 -Nx 50.00 -Ny 50.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 13a -Md 0 -Ms 1 -Mt 800 -Is 1.0 -Id 26.0
-It cbr -Il 1020 -If 0 -Ii 4 -Iv 29 -Pn cbr -Pi 4 -Pf 1 -Po 0
```

--Message is forwarded by node 0.

```
s -t 2.140000000 -Hs 1 -Hd 0 -Ni 1 -Nx 1.00 -Ny 1.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.0 -Id 26.0 -It
cbr -Il 1020 -If 0 -Ii 7 -Iv 30 -Pn cbr -Pi 7 -Pf 0 -Po 0
r -t 2.140822742 -Hs 0 -Hd 0 -Ni 0 -Nx 50.00 -Ny 50.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 13a -Md 0 -Ms 1 -Mt 800 -Is 1.0 -Id 26.0
-It cbr -Il 1020 -If 0 -Ii 2 -Iv 30 -Pn cbr -Pi 2 -Pf 1 -Po 0
f -t 2.140822742 -Hs 0 -Hd 13 -Ni 0 -Nx 50.00 -Ny 50.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 13a -Md 0 -Ms 1 -Mt 800 -Is 1.0 -Id 26.0
-It cbr -Il 1020 -If 0 -Ii 2 -Iv 29 -Pn cbr -Pi 2 -Pf 1 -Po 0
r -t 2.150459666 -Hs 0 -Hd 0 -Ni 0 -Nx 50.00 -Ny 50.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 13a -Md 0 -Ms 1 -Mt 800 -Is 1.0 -Id 26.0
-It cbr -Il 1020 -If 0 -Ii 4 -Iv 30 -Pn cbr -Pi 4 -Pf 1 -Po 0
f -t 2.150459666 -Hs 0 -Hd 13 -Ni 0 -Nx 50.00 -Ny 50.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 13a -Md 0 -Ms 1 -Mt 800 -Is 1.0 -Id 26.0
-It cbr -Il 1020 -If 0 -Ii 4 -Iv 29 -Pn cbr -Pi 4 -Pf 1 -Po 0
```

---From the hop source and destination we can find that, data delivery is done in the route 1-0-13