1. Set up a 2-node wireless network. Analyze TCP performance for this scenario with DSDV as routing protocols.

Aim: To understand and design the two node wireless network. Also here we understand the working of DSDV routing protocol and analyze its performance.

Theory:

Theory:

- Define the options for the wireless network.(Choose routing protocol as DSDV, No. of nodes 2)
- Create a simulator object.
- Setup trace and nam and call methods traceall,namtrace-all-wireless
- Create topology, God
- Configure and then create nodes.
- Define node movement.
- Generate TCP traffic and Termination of simulated events
- Start the simulation

Save the following program as 001.tcl

#Setting the Default Parameters

```
#####################################################################
#           Setting the Default Parameters                          #
#####################################################################
set val(chan)           Channel/WirelessChannel
set val(prop)           Propagation/TwoRayGround
set val(netif)          Phy/WirelessPhy
set val(mac)            Mac/802_11
set val(ifq)                Queue/DropTail/PriQueue
set val(ll)                 LL
set val(ant)            Antenna/OmniAntenna
set val(x)                  500
set val(y)                  500
set val(ifqlen)         50
set val(nn)                 2
set val(stop)           20.0
set val(rp)             DSDV
```

```
set ns_ [new Simulator]

set tracefd     [open 001.tr w]
$ns_ trace-all $tracefd

set namtrace [open 001.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

set prop [new $val(prop)]

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

create-god $val(nn)

#Node Configuration

        $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) \
                         -channelType $val(chan) \
                         -topoInstance $topo \
                         -agentTrace ON \
                         -routerTrace ON \
                         -macTrace ON

#Creating Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
     set node_($i) [$ns_ node]
    $node_($i) random-motion 0
     }

#Initial Positions of Nodes
```

```
for {set i 0} {$i < $val(nn)} {incr i} {
        $ns_ initial_node_pos $node_($i) 40
}

#Topology Design

$ns_ at 1.1 "$node_(0) setdest 310.0 10.0 20.0"
$ns_ at 1.1 "$node_(1) setdest 10.0 310.0 20.0"

#Generating Traffic

  set tcp0 [new Agent/TCP]
    set sink0 [new Agent/TCPSink]
  $ns_ attach-agent $node_(0) $tcp0
     $ns_ attach-agent $node_(1) $sink0
  $ns_ connect $tcp0 $sink0
  set ftp0 [new Application/FTP]
  $ftp0 attach-agent $tcp0
  $ns_ at 1.0 "$ftp0 start"
     $ns_ at 18.0 "$ftp0 stop"

#Simulation Termination

for {set i 0} {$i < $val(nn) } {incr i} {
   $ns_ at $val(stop) "$node_($i) reset";
}
$ns_ at $val(stop) "puts \"NS EXITING...\" ; $ns_ halt"

puts "Starting Simulation..."

$ns_ run
```

2. Set up 3-node wireless network with node N1 between N0 and N2. As the nodes N0 and N2 moves towards each other they exchange packets. As they move out of each other's range they drop some packets. Analyze TCP performance for this scenario with AODV, DSDV and DSR as routing protocols.

Aim: To understand and design the three node wireless network. Here we implement the working of DSDV, AODV, DSR routing protocol and analyze their performance.

Theory:

- Define the options for the wireless network.(Choose routing protocol as DSDV,AODV,DSR(one at a time) ,No. of nodes 3)
- Create a simulator object.
- Setup trace and nam and call methods traceall,namtrace-all-wireless
- Create topology, God
- Configure and then create nodes.
- Define node movement i.e n0 and n2 moves towards each other and after some time move away from each other.
- Generate TCP traffic and Termination of simulated events
- Start the simulation

Save the following program as 002.tcl

#Setting the Default Parameters

```
set val(chan)           Channel/WirelessChannel
set val(prop)           Propagation/TwoRayGround
set val(netif)          Phy/WirelessPhy
set val(mac)          Mac/802_11
 #set val(ifq)          CMUPriQueue
set val(ifq)        Queue/DropTail/PriQueue
set val(ll)               LL
set val(ant)          Antenna/OmniAntenna
set val(x)                  500
set val(y)                  400
set val(ifqlen)           50
set val(nn)               3
set val(stop)           60.0
```

```
 set val(rp)             AODV


set ns_ [new Simulator]

set tracefd     [open 002.tr w]
$ns_ trace-all $tracefd

set namtrace [open 002.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

set prop        [new $val(prop)]

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

create-god $val(nn)

#Node Configuration

        $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) \
                    -channelType $val(chan) \
                         -topoInstance $topo \
                         -agentTrace ON \
                         -routerTrace ON \
                         -macTrace ON

#Creating Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0
    }
```

```
#Initial Positions of Nodes
$node_(0) set x_ 5.0
$node_(0) set y_ 5.0
$node_(0) set z_ 0.0

$node_(1) set x_ 490.0
$node_(1) set y_ 285.0
$node_(1) set z_ 0.0

$node_(2) set x_ 150.0
$node_(2) set y_ 240.0
$node_(2) set z_ 0.0


for {set i 0} {$i < $val(nn)} {incr i} {
        $ns_ initial_node_pos $node_($i) 40
}

#Topology Design

$ns_ at 0.0 "$node_(0) setdest 450.0 285.0  30.0"
$ns_ at 0.0 "$node_(1) setdest 200.0 285.0 30.0"
$ns_ at 0.0 "$node_(2) setdest 1.0 285.0 30.0"


$ns_ at 25.0 "$node_(0) setdest 300.0 285.0 10.0"
$ns_ at 25.0 "$node_(2) setdest 100.0 285.0 10.0"


$ns_ at 40.0 "$node_(0) setdest 490.0 285.0  5.0"
$ns_ at 40.0 "$node_(2) setdest 1.0 285.0 5.0"


#Generating Traffic
  set tcp0 [new Agent/TCP]
    set sink0 [new Agent/TCPSink]
  $ns_ attach-agent $node_(0) $tcp0
     $ns_ attach-agent $node_(2) $sink0
  $ns_ connect $tcp0 $sink0
  set ftp0 [new Application/FTP]
```

```
$ftp0 attach-agent $tcp0
$ns_ at 10.0 "$ftp0 start"


#Simulation Termination

for {set i 0} {$i < $val(nn) } {incr i} {
   $ns_ at $val(stop) "$node_($i) reset";
   }
   $ns_ at $val(stop) "puts \"NS EXITING...\" ; $ns_ halt"

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

Repeat the simulation for AODV and DSR Routing protocols.

Output: Demonstration of wireless network, Analysis of routing protocols.

3. Set up a 6-node wireless network; analyze TCP performance when nodes are static and mobile.

Aim: To simulate 6 node wireless network. Here we analyze performance of nodes when they are static and mobile.

Theory:

Theory:

- Define the options for the wireless network.(Choose No. of nodes 6)
- Create a simulator object.
- Setup trace and nam and call methods traceall,namtrace-all-wireless
- Create topology, God
- Configure and then create nodes.
- Define node movement.
- Generate TCP traffic and Termination of simulated events
- Start the simulation

To make nodes static and mobile follow the setup below

Desktop>ns-2.35>ns-2.35all_in_one>indeputils>lib>cmu-scen-gen>cbrgen.tcl

nodes: 6, max conn: 8, send rate: 0.0, seed: 1.0

Desktop>ns-2.35>ns-2.35all_in_one>indeputils>lib>cmu-scen-gen>setdest>setdest.h

nodes: 6, pause: 2.00, max speed: 40.00, max x: 200.00, max y: 300.00

Save the following program as 003.tcl

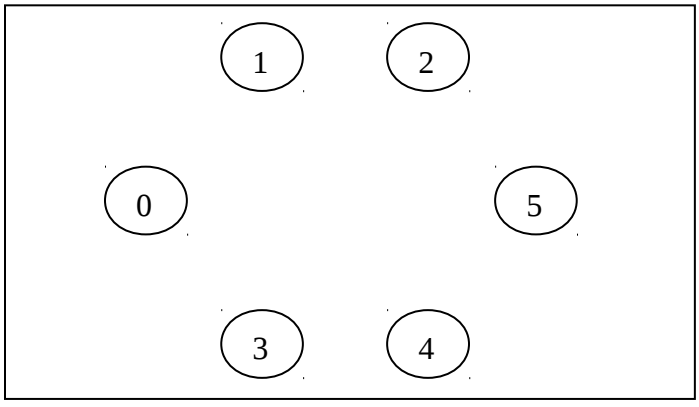
```
set val(chan)           Channel/WirelessChannel
set val(prop)           Propagation/TwoRayGround
set val(netif)          Phy/WirelessPhy
set val(mac)          Mac/802_11
set val(ifq)            Queue/DropTail/PriQueue
set val(ll)             LL
set val(ant)          Antenna/OmniAntenna
set val(x)                  500
set val(y)                  500
set val(ifqlen)          50
set val(nn)              25
set val(stop)           100.0
    set val(rp)          AODV
 #set val(sc)             "mob-25-50"
set val(cp)            "tcp-25-8"

set ns_ [new Simulator]

set tracefd     [open 003.tr w]
$ns_ trace-all $tracefd

set namtrace [open 003.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)


set prop        [new $val(prop)]

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

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

```
    #Node Configuration

      $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) \
                  -channelType $val(chan) \
                       -topoInstance $topo \
                       -agentTrace ON \
                       -routerTrace ON \
                       -macTrace ON

#Creating Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0
}


     for {set i 0} {$i < $val(nn) } { incr i } {
              set xx [expr rand()*500]
              set yy [expr rand()*400]
              $node_($i) set X_ $xx
              $node_($i) set Y_ $yy

          }


#Initial Positions of Nodes

for {set i 0} {$i < $val(nn)} {incr i} {
        $ns_ initial_node_pos $node_($i) 40
}


#puts "Loading scenario file..."
```

```
#source $val(sc)
puts "Loading connection file..."
source $val(cp)


#Simulation Termination

for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at $val(stop) "$node_($i) reset";
    }
    $ns_ at $val(stop) "puts \"NS EXITING...\" ; $ns_ halt"

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

Expected Output: Setup of wireless network and Performance analysis of static and mobile nodes.

4. **Write a TCL script to simulate the following scenario. Consider six nodes, (as shown in the figure below) moving within a flat topology of 700m x 700m. The initial positions of nodes are: n0(150, 300), n1(300, 500), n2(500, 500), n3 (300, 100), n4(500, 100) and n5(650, 300) respectively. A TCP connection is initiated between n0 (source) and n5 (destination) through n3 and n4 i.e., the route is 0-3-4-5. At time t = 3 seconds, the FTP application runs over it. After time t = 4 seconds, n3(300,100) moves towards n1(300, 500) with a speed of 5.0m/sec and after some time the path breaks. The data is then transmitted with a new path via n1 and n2 i.e., the new route is 0-1-2-5. The simulation lasts for 60 secs. In the above said case both the routes have equal cost. Use DSR as the routing protocol and the IEEE 802.11 MAC protocol.**



Aim: To simulate wireless network with six nodes. Here we implement the working of DSR routing protocol, IEEE 802.11 and analyze performance when a specific path breaks.

Theory:

- Define the options for the wireless network.(Choose routing protocol as DSR,No. of nodes 6)
- Create a simulator object.
- Setup trace and nam and call methods traceall,namtrace-all-wireless
- Create topology, God
- Configure and then create nodes.
- Define node movement.
- Generate TCP traffic and Termination of simulated events
- Start the simulation

Save the following program as 004tcl

```
set val(chan)         Channel/WirelessChannel
set val(prop)         Propagation/TwoRayGround
set val(netif)        Phy/WirelessPhy
set val(mac)        Mac/802_11
#set val(ifq)         Queue/DropTail/PriQueue
set val(ifq)        CMUPriQueue
set val(ll)           LL
set val(ant)        Antenna/OmniAntenna
set val(x)                700
set val(y)                700
set val(ifqlen)        50
set val(nn)            6
set val(stop)         60.0
   set val(rp)         DSR


set ns_ [new Simulator]
```

```
set tracefd     [open 004.tr w]
$ns_ trace-all $tracefd

set namtrace [open 004.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

set prop        [new $val(prop)]

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

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

    #Node Configuration

      $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) \
                  -channelType $val(chan) \
                       -topoInstance $topo \
                       -agentTrace ON \
                       -routerTrace ON \
                       -macTrace ON

#Creating Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0
}
#Initial Positions of Nodes



$node_(0) set X_ 150.0
```

```
$node_(0) set Y_ 300.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 300.0
$node_(1) set Y_ 500.0
$node_(1) set Z_ 0.0

$node_(2) set X_ 500.0
$node_(2) set Y_ 500.0
$node_(2) set Z_ 0.0

$node_(3) set X_ 300.0
$node_(3) set Y_ 100.0
$node_(3) set Z_ 0.0

$node_(4) set X_ 500.0
$node_(4) set Y_ 100.0
$node_(4) set Z_ 0.0

$node_(5) set X_ 650.0
$node_(5) set Y_ 300.0
$node_(5) set Z_ 0.0
for {set i 0} {$i < $val(nn)} {incr i} {
        $ns_ initial_node_pos $node_($i) 40
}

#Topology Design
$ns_ at 1.0 "$node_(0) setdest 160.0 300.0 2.0"
$ns_ at 1.0 "$node_(1) setdest 310.0 150.0 2.0"
$ns_ at 1.0 "$node_(2) setdest 490.0 490.0 2.0"
$ns_ at 1.0 "$node_(3) setdest 300.0 120.0 2.0"
$ns_ at 1.0 "$node_(4) setdest 510.0 90.0 2.0"
$ns_ at 1.0 "$node_(5) setdest 640.0 290.0 2.0"

$ns_ at 4.0 "$node_(3) setdest 300.0 500.0 5.0"

#Generating Traffic
  set tcp0 [new Agent/TCP]
    set sink0 [new Agent/TCPSink]
  $ns_ attach-agent $node_(0) $tcp0
```

```
    $ns_ attach-agent $node_(5) $sink0
$ns_ connect $tcp0 $sink0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns_ at 5.0 "$ftp0 start"
    $ns_ at 60.0 "$ftp0 stop"


#Simulation Termination

for {set i 0} {$i < $val(nn) } {incr i} {
   $ns_ at $val(stop) "$node_($i) reset";
   }
   $ns_ at $val(stop) "puts \"NS EXITING...\" ; $ns_ halt"

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

Output: Demonstration of wireless network, Analysis of DSR routing protocol.

5. Simulate simple ESS and with transmitting nodes in wireless LAN by simulation and determine the performance with respect to transmission of packet.

Solution:

Aim: To simulate wireless network. Here we implement LAN, analyze performance wrt packet transmission.

Theory:

- Define the options for the wireless network.
- Create a simulator object.
- Setup trace and nam and call methods traceall,namtrace-all-wireless
- Create topology, God
- Configure and then create nodes.
- Set up the LAN
- Generate TCP traffic and Termination of simulated events
- Start the simulation

Save the following program as 005.tcl.

```
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 ns [new Simulator]

set f [open 005.tr w]
$ns trace-all $f
set nf [open 005.nam w]
$ns namtrace-all-wireless $nf 100 100


$ns node-config       -llType $val(ll) \
                              -macType $val(mac) \
                              -ifqType $val(ifq) \
                               -ifqLen $val(ifqlen) \
                               -antType $val(ant) \
                               -propType $val(prop) \
                               -phyType $val(netif) \
                               -channelType $val(chan) \

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]

set lan [$ns newLan "$n0 $n1 $n2 $n3 $n4 $n5" 0.5Mb 30ms LL Queue/DropTail
Mac/802_11 Channel/WirelessChannel]


set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
```

```
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0

set sink0 [new Agent/TCPSink]
$ns attach-agent $n5 $sink0

$ns connect $tcp0 $sink0

$ns at 5.000000 "$ftp0 start"

proc finish {} {
global ns f nf
$ns flush-trace
close $f
close $nf
puts "running nam..."
exec nam 005.nam &
exit 0
}
$ns at 60.0000 "finish"
$ns run
```

Expected output: Animated 6 node structure is displayed. We need to see the trace file to understand what has happened to the data flow and packet dropped.

6. Set up a wireless network with mobile nodes, induce 1 to 10% error to the network using uniform error model. Plot the congestion window for TCP connections. Write your observation on TCP performance as error increases in the network.

Solution:
Aim: To simulate wireless network with mobile nodes and induce error to analyze performance..
Theory:

- Define the options for the wireless network.
- Create a simulator object.

- Setup trace and nam and call methods traceall,namtrace-all-wireless
- Create topology, God
- Create uniform error procedure and induce the interval at which the error is to be generated.
- Configure and then create nodes.
- Set up the LAN
- Generate TCP traffic and Termination of simulated events

Start the simulation

Save the following program as 006.tcl.

```
set val(chan)          Channel/WirelessChannel
set val(prop)          Propagation/TwoRayGround
set val(netif)         Phy/WirelessPhy
set val(mac)         Mac/802_11
set val(ifq)           Queue/DropTail/PriQueue
set val(ll)            LL
set val(ant)         Antenna/OmniAntenna
set val(x)                500
set val(y)                500
set val(ifqlen)         50
set val(nn)             5
set val(stop)          50.0
    set val(rp)         AODV


set ns_ [new Simulator]

set tracefd     [open 006.tr w]
$ns_ trace-all $tracefd

set namtrace [open 006.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

set prop        [new $val(prop)]

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

create-god $val(nn)
```

```
    #Node Configuration

      $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) \
                  -channelType $val(chan) \
                       -topoInstance $topo \
                       -agentTrace ON \
                       -routerTrace ON \
                       -macTrace ON \
                    -IncomingErrProc "uniformErr" \
                       -OutgoingErrProc "uniformErr"

proc uniformErr {} {
set err [new ErrorModel]
$err unit pkt
$err set rate_ 0.01
return $err
}

#Creating Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0
}
#Initial Positions of Nodes

for {set i 0} {$i < $val(nn)} {incr i} {
        $ns_ initial_node_pos $node_($i) 40
}
#Topology Design

$ns_ at 1.0 "$node_(0) setdest 10.0 10.0 50.0"
$ns_ at 1.0 "$node_(1) setdest 10.0 100.0 50.0"
```

```
$ns_ at 1.0 "$node_(4) setdest 50.0 50.0 50.0"
$ns_ at 1.0 "$node_(2) setdest 100.0 100.0 50.0"
$ns_ at 1.0 "$node_(3) setdest 100.0 10.0 50.0"

#Generating Traffic
  set tcp0 [new Agent/TCP]
    set sink0 [new Agent/TCPSink]
  $ns_ attach-agent $node_(0) $tcp0
     $ns_ attach-agent $node_(2) $sink0
  $ns_ connect $tcp0 $sink0
  set ftp0 [new Application/FTP]
  $ftp0 attach-agent $tcp0
  $ns_ at 1.0 "$ftp0 start"
     $ns_ at 50.0 "$ftp0 stop"

       set tcp1 [new Agent/TCP]
    set sink1 [new Agent/TCPSink]
  $ns_ attach-agent $node_(1) $tcp1
     $ns_ attach-agent $node_(2) $sink1
  $ns_ connect $tcp1 $sink1
  set ftp1 [new Application/FTP]
  $ftp1 attach-agent $tcp1
  $ns_ at 1.0 "$ftp1 start"
     $ns_ at 50.0 "$ftp1 stop"

#Simulation Termination

for {set i 0} {$i < $val(nn) } {incr i} {
   $ns_ at $val(stop) "$node_($i) reset";
   }
   $ns_ at $val(stop) "puts \"NS EXITING...\" ; $ns_ halt"

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

Expected output: Animated nodes structure is displayed. We need to see the performance of the network with a varying error rate.

7. Set up a wireless network with mobile nodes; provide energy to the nodes using Energy Model. Assume initial energy as 100. Analyze the performance of TCP.

Aim: To simulate wireless network with mobile nodes and setting energy to analyze performance.

Theory:

- Define the options for the wireless network.(Choose initial energy to 100)
- Create a simulator object.
- Setup trace and nam and call methods traceall,namtrace-all-wireless
- Create topology, God
- Configure and then create nodes.(various parameters of energy model are to be set)
- Set up the LAN
- Generate TCP traffic and Termination of simulated events

Start the simulation

Save the following program as 007.tcl.

```
set val(chan)           Channel/WirelessChannel
set val(prop)           Propagation/TwoRayGround
set val(netif)          Phy/WirelessPhy
set val(mac)          Mac/802_11
#set val(ifq)           Queue/DropTail/PriQueue
 set val(ifq)          CMUPriQueue
set val(ll)             LL
set val(ant)          Antenna/OmniAntenna
set val(x)                  500
set val(y)                  500
set val(ifqlen)          50
set val(nn)              25
set val(stop)           100.0
    set val(rp)          DSR
set val(energymodel)    EnergyModel
set val(initialenergy)  100
  set val(sc)            "mob-25-test"
 set val(cp)            "tcp-25-8"

set ns_ [new Simulator]
```

```
set tracefd     [open 007.tr w]
$ns_ trace-all $tracefd

set namtrace [open 007.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

set prop        [new $val(prop)]

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

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

    #Node Configuration

      $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) \
                  -channelType $val(chan) \
                       -topoInstance $topo \
                       -agentTrace ON \
                       -routerTrace ON \
                       -macTrace ON \
                    -energyModel $val(energymodel) \
                       -idlePower 0.005 \
                       -rxPower 1.0 \
                       -txPower 5.0 \
                  -sleepPower 0.0001 \
                  -transitionPower 0.002 \
                  -transitionTime 0.005 \
                       -initialEnergy $val(initialenergy)

#Creating Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
```

```
    $node_($i) random-motion 0
}

#Initial Positions of Nodes

for {set i 0} {$i < $val(nn)} {incr i} {
        $ns_ initial_node_pos $node_($i) 40
}


puts "Loading scenario file..."
source $val(sc)
puts "Loading connection file..."
source $val(cp)

#Simulation Termination

for {set i 0} {$i < $val(nn) } {incr i} {
   $ns_ at $val(stop) "$node_($i) reset";
   }
   $ns_ at $val(stop) "puts \"NS EXITING...\" ; $ns_ halt"

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

Expected output: Animated nodes structure is displayed. We need to see the performance of the network with a varying Initial energy.