 # aodv.tcl
Ad hoc On-Demand Distance Vector Routing
 # A 3-node example for ad-hoc simulation with aodv

 # 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   ( use to assign
                                                        #the buffering capacity of wireless
interface)
set val(nn)           3                          ;       # number of mobilenodes
set val(rp)           AODV                       ;              # routing protocol
set val(x)            500                        ;     # X dimension of topography
set val(y)            400                        ;     # Y dimension of topography
set val(stop)         150                        ;              # time of simulation end

set ns_               [new Simulator]
```

#newtrace (new format of trace file for wireless)
 # for using this write as below

```
$ns_ use-newtrace

set tracefd       [open simple.tr w]
set windowVsTime2 [open win.tr w]
set namtrace      [open simple.nam w]
```

 # trace-all $filename causes trace objects to be pushed on all links. If you only want to trace one link, there's no need for this overhead. Saving is about 14 KB/link.

```
$ns_ trace-all $tracefd


$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
```

 # Topographyis the study of Earth's surface shape and features or those of planets, #moons, and asteroids
 # set up topography object
#Create and configure topography object (Used for mobile scenario)

```
set topo       [new Topography]
```

#The load_flatgrid object is used to specify a 2-D terrain. Support is available for simulation of 3D terrains for more realistic depiction of scenarios.

```
$topo load_flatgrid $val(x) $val(y)
```

 # GOD or General Operations Director is a ns-2 simulator object, which is used to store global information about the state of the environment, network, or nodes that an omniscient observer would have, but that should not be made known to any participant in the simulation

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

 # configure the 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) \
                -channelType $val(chan) \
                -topoInstance $topo \
                -agentTrace ON \
                -routerTrace ON \
                -macTrace ON \
                -movementTrace ON
```

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

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

#By default, a node is specified as a unicast node. If a multicast protocol is desired, a #separate clause has to be specified during simulator initialization-

 # set ns [new Simulator -multicast on]

 # Provide initial location of mobilenodes

```
$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
```

 # Generation of movements

```
$ns_ at 10.0 "$node_(0) setdest 250.0 250.0 3.0"
$ns_ at 15.0 "$node_(1) setdest 45.0 285.0 5.0"
$ns_ at 110.0 "$node_(2) setdest 480.0 300.0 5.0"

# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/Newreno]
$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"

# Printing the window size
proc plotWindow {tcpSource file}
{
global ns_
set time 0.01
set now [$ns_ now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns_ at  [expr $now+$time] "plotWindow $tcpSource $file"
}
$ns_ at 10.1 "plotWindow $tcp $windowVsTime2"

# Define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 30 defines the node size for nam
$ns_ initial_node_pos $node_($i) 30
}
```

```
# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns_ at $val(stop) "$node_($i) reset";
}

# ending nam and the simulation
$ns_ at $val(stop) "$ns_ nam-end-wireless $val(stop)"
$ns_ at $val(stop) "stop"
$ns_ at 150.01 "puts \"end simulation\" ; $ns_ halt"
proc stop {} {
    global ns_ tracefd namtrace
    $ns_ flush-trace
    close $tracefd
    close $namtrace
}

$ns_ run
```