- Appendix A - Tcl/Tk Implementation Code for NS Simulator

A TCL/TK code is implemented for simulate the two protocols AODV and intelligent ant sense governed by two cases of network size 11 and 25 nodes. Control file is used to dedicate the nodes and their distributed area in the network. The following scenarios show the TCL/TK code files for AODV and Intelligent Ant Sense protocol with the process of load balance model.

A.1- Scenario 1

- **15 nodes TCL/TK Script file**

```
#############################################
#            Intelligent Ant Sense                    #
#            Copyright (c) 2011                       #
#############################################

# ======================================================
# Define options
# ======================================================
set val(chan)         Channel/WirelessChannel       ;# Channel Type
set val(prop)         Propagation/TwoRayGround      ;# radio-propagation model
set val(netif)        Phy/WirelessPhy/802_15_4
set val(mac)          Mac/802_15_4
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)           15                            ;# number of mobilenodes
set val(rp)           INTANTSENSE                   ;# routing protocol
set val(x)            50
set val(y)            50
set val(nam)          intelligentantsense.nam
set val(traffic)      cbr                 ;# cbr
```

```
# traffic Pause time

#set appTime1          2.0      ;# in seconds
set appTime2           2.0      ;# in seconds
set appTime3           2.0      ;# in seconds

#simulation time
set stopTime           100      ;# in seconds
# Initialize Global Variables
set ns_              [new Simulator]
set tracefd     [open ./intelligentantsense.tr w]
$ns_ trace-all $tracefd            ;
if { "$val(nam)" == "intelligentantsense.nam" } {
       set namtrace     [open ./$val(nam) w]
       $ns_ namtrace-all-wireless $namtrace $val(x) $val(y) ;
}
$ns_ puts-nam-traceall {# nam4wpan #}          ;# inform nam that this is a trace file for wpan (special handling needed)

Mac/802_15_4 wpanNam namStatus on          ;# default = off (should be turned on before other 'wpanNam' commands can work)

# For model 'TwoRayGround'
set dist(5m)  7.69113e-06
set dist(9m)  2.37381e-06
set dist(10m) 1.92278e-06
set dist(11m) 1.58908e-06
set dist(12m) 1.33527e-06
set dist(13m) 1.13774e-06
set dist(14m) 9.81011e-07
set dist(15m) 8.54570e-07
set dist(16m) 7.51087e-07
set dist(20m) 4.80696e-07
set dist(25m) 3.07645e-07
set dist(30m) 2.13643e-07
set dist(35m) 1.56962e-07
set dist(40m) 1.20174e-07
Phy/WirelessPhy set CSThresh_ $dist(15m)
```

```
Phy/WirelessPhy set RXThresh_ $dist(15m)
# set up topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
# Create God
set god_ [create-god $val(nn)]
set chan_1_ [new $val(chan)]
# configure node
$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 \
             -agentTrace OFF \
             -routerTrace OFF \
             -macTrace ON \
             -movementTrace OFF \
              #-energyModel "EnergyModel" \
              #-initialEnergy 1 \
              #-rxPower 0.3 \
              #-txPower 0.3 \
             -channel $chan_1_
for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0              ;# disable random motion
}
source ./control.scn
# Setup traffic flow between nodes
proc cbrtraffic { src dst interval starttime } {
  global ns_ node_
  set udp($src) [new Agent/UDP]                    ;#set udp [new Agent/UDP]
  eval $ns_ attach-agent \$node_($src) \$udp($src)          ;# $ns_ attach-agent
<node> <agent>
  set null($dst) [new Agent/Null]
  eval $ns_ attach-agent \$node_($dst) \$null($dst)
  set cbr($src) [new Application/Traffic/CBR]
```

```
  eval \$cbr($src) set packetSize_ 70
  eval \$cbr($src) set interval_ $interval
  eval \$cbr($src) set random_ 0
  eval \$cbr($src) set maxpkts_ 10000
  eval \$cbr($src) attach-agent \$udp($src)
  eval $ns_ connect \$udp($src) \$null($dst)
  $ns_ at $starttime "$cbr($src) start"
}
if { ("$val(traffic)" == "cbr")} {
  puts "\nTraffic: $val(traffic)"
  #Mac/802_15_4 wpanCmd ack4data on
  puts [format "Acknowledgement for data: %s" [Mac/802_15_4 wpanCmd
ack4data]]
  set lowSpeed 0.5ms
  set highSpeed 1.5ms
  Mac/802_15_4 wpanNam PlaybackRate $lowSpeed
  #$ns_ at [expr $appTime1+0.1] "Mac/802_15_4 wpanNam PlaybackRate
$highSpeed"
  $ns_ at $appTime2 "Mac/802_15_4 wpanNam PlaybackRate $lowSpeed"
  $ns_ at [expr $appTime2+0.1] "Mac/802_15_4 wpanNam PlaybackRate
$highSpeed"
  $ns_ at $appTime3 "Mac/802_15_4 wpanNam PlaybackRate $lowSpeed"
  $ns_ at [expr $appTime3+0.1] "Mac/802_15_4 wpanNam PlaybackRate
$highSpeed"
  #eval $val(traffic)traffic 19 6 0.2 $appTime1
  eval $val(traffic)traffic 10 4 0.2 $appTime2
  eval $val(traffic)traffic 3 2 0.2 $appTime3
  Mac/802_15_4 wpanNam FlowClr -p cbr -c tomato
  Mac/802_15_4 wpanNam FlowClr -p ARP -c green
  if { "$val(traffic)" == "cbr" } {
    set pktType cbr
  } else {
    set pktType exp
  }
  #Mac/802_15_4 wpanNam FlowClr -p $pktType -s 19 -d 6 -c blue
  Mac/802_15_4 wpanNam FlowClr -p $pktType -s 10 -d 4 -c green4
  Mac/802_15_4 wpanNam FlowClr -p $pktType -s 3 -d 2 -c cyan4
  #$ns_ at $appTime1 "$node_(19) NodeClr blue"
  #$ns_ at $appTime1 "$node_(6) NodeClr blue"
```

```
  #$ns_ at $appTime1 "$ns_ trace-annotate \"(at $appTime1) $val(traffic) traffic
from node 19 to node 6\""
  $ns_ at $appTime2 "$node_(10) NodeClr green4"
  $ns_ at $appTime2 "$node_(4) NodeClr green4"
  $ns_ at $appTime2 "$ns_ trace-annotate \"(at $appTime2) $val(traffic) traffic
from node 10 to node 4\""
  $ns_ at $appTime3 "$node_(3) NodeClr cyan3"
  $ns_ at $appTime3 "$node_(2) NodeClr cyan3"
  $ns_ at $appTime3 "$ns_ trace-annotate \"(at $appTime3) $val(traffic) traffic
from node 3 to node 2\""
}
# defines the node size in nam
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ initial_node_pos $node_($i) 2
}
# Tell nodes when the simulation ends
for {set i 0} {$i < $val(nn) } {incr i} {
   $ns_ at $stopTime "$node_($i) reset";
}
$ns_ at $stopTime "stop"
$ns_ at $stopTime "puts \"\nNS EXITING...\""
$ns_ at $stopTime "$ns_ halt"
proc stop {} {
   global ns_ tracefd val env
   $ns_ flush-trace
   close $tracefd
   set hasDISPLAY 0
   foreach index [array names env] {
      #puts "$index: $env($index)"
      if { ("$index" == "DISPLAY") && ("$env($index)" != "") } {
            set hasDISPLAY 1
      }
   }
   if { ("$val(nam)" == "intelligentantsense.nam") && ("$hasDISPLAY" == "1")
} {
       exec nam intelligentantsense.nam &
   }
}
puts "\nStarting Simulation..."
$ns_ run
```

- **15 nodes control.scn file**

```
# Nodes: 15
$node_(0) set X_ 32.178377307314
$node_(0) set Y_ 33.5556991827566
$node_(0) set Z_ 0.000000000000
$node_(1) set X_ 38.487311778721
$node_(1) set Y_ 32.782820874924
$node_(1) set Z_ 0.000000000000
$node_(2) set X_ 20.437938215931
$node_(2) set Y_ 22.771658830425
$node_(2) set Z_ 0.000000000000
$node_(3) set X_ 15.72998766842667
$node_(3) set Y_ 33.993389414657
$node_(3) set Z_ 0.000000000000
$node_(4) set X_ 26.146269526711
$node_(4) set Y_ 28.547450139045
$node_(4) set Z_ 0.000000000000
$node_(5) set X_ 26.476535726599
$node_(5) set Y_ 33.706396828920
$node_(5) set Z_ 0.000000000000
$node_(6) set X_ 33.810596917301
$node_(6) set Y_ 23.160042376073
$node_(6) set Z_ 0.000000000000
$node_(7) set X_ 19.995665842364
$node_(7) set Y_ 32.337008965783
$node_(7) set Z_ 0.000000000000
$node_(8) set X_ 14.741089465783
$node_(8) set Y_ 28.871428688349
$node_(8) set Z_ 0.000000000000
$node_(9) set X_ 38.725856848941
$node_(9) set Y_ 27.949644538277
$node_(9) set Z_ 0.000000000000
$node_(10) set X_ 14.346871605178
$node_(10) set Y_ 23.566174228621
$node_(10) set Z_ 0.000000000000
$node_(11) set X_ 10.828197591068
$node_(11) set Y_ 32.892175912553
$node_(11) set Z_ 0.000000000000
```

```
$node_(12) set X_ 20.519762507870
$node_(12) set Y_ 27.069997591097
$node_(12) set Z_ 0.000000000000
$node_(13) set X_ 26.697017351151
$node_(13) set Y_ 22.852149009306
$node_(13) set Z_ 0.000000000000
$node_(14) set X_ 31.315083056450
$node_(14) set Y_ 28.186742480258
$node_(14) set Z_ 0.000000000000

#movement process at 1.5m/s speed
$ns_ at 2.0 "$node_(0) setdest 25.81 46.655 1.5"
$ns_ at 2.0 "$node_(1) setdest 38.41 11.53 1.5"
$ns_ at 2.0 "$node_(3) setdest 35.0 31.28 1.5"
$ns_ at 2.0 "$node_(5) setdest 12.81 26.655 1.5"
$ns_ at 2.0 "$node_(9) setdest 16.73 25.75 1.5"
$ns_ at 2.0 "$node_(11) setdest 25.49 29.7 1.5"
$ns_ at 2.0 "$node_(10) setdest 35.82 36.38 1.5"
$ns_ at 2.0 "$node_(14) setdest 33.81 36.655 1.5"
```

A.2- Scenario 2

- **25 nodes TCL/TK Script file**

```
#############################################
#                   AODV                    #
#          Copyright (c) 2011               #
#############################################
 ====
# Define options
====

set val(chan)         Channel/WirelessChannel         ;# Channel Type
set val(prop)         Propagation/TwoRayGround        ;# radio-propagation model
set val(netif)        Phy/WirelessPhy/802_15_4
set val(mac)          Mac/802_15_4
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)           25                              ;# number of mobilenodes
set val(rp)           AODV                            ;# routing protocol
set val(x)            50
set val(y)            50
set val(nam)            aodv.nam
set val(traffic)   cbr                  ;# cbr

# traffic Pause time

set appTime1          2.0       ;# in seconds
set appTime2          2.0       ;# in seconds
set appTime3          2.0       ;# in seconds
```

```
# simulation time
set stopTime            100       ;# in seconds

# Initialize Global Variables
set ns_             [new Simulator]
set tracefd     [open ./aodv.tr w]
$ns_ trace-all $tracefd             ;
if { "$val(nam)" == "aodv.nam" } {
        set namtrace     [open ./$val(nam) w]
        $ns_ namtrace-all-wireless $namtrace $val(x) $val(y) ;
}
$ns_ puts-nam-traceall {# nam4wpan #}           ;# inform nam that this is a trace file for wpan (special handling needed)

Mac/802_15_4 wpanNam namStatus on           ;# default = off (should be turned on before other 'wpanNam' commands can work)
# For model 'TwoRayGround'
set dist(5m)  7.69113e-06
set dist(9m)  2.37381e-06
set dist(10m) 1.92278e-06
set dist(11m) 1.58908e-06
set dist(12m) 1.33527e-06
set dist(13m) 1.13774e-06
set dist(14m) 9.81011e-07
set dist(15m) 8.54570e-07
set dist(16m) 7.51087e-07
set dist(20m) 4.80696e-07
set dist(25m) 3.07645e-07
set dist(30m) 2.13643e-07
set dist(35m) 1.56962e-07
set dist(40m) 1.20174e-07
Phy/WirelessPhy set CSThresh_ $dist(15m)
Phy/WirelessPhy set RXThresh_ $dist(15m)
# set up topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
# Create God
set god_ [create-god $val(nn)]
set chan_1_ [new $val(chan)]
# configure node
```

```
$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 \
             -agentTrace OFF \
             -routerTrace OFF \
             -macTrace ON \
             -movementTrace OFF \
              #-energyModel "EnergyModel" \
              #-initialEnergy 1 \
              #-rxPower 0.3 \
              #-txPower 0.3 \
             -channel $chan_1_

for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0              ;# disable random motion
}
source ./control.scn
# Setup traffic flow between nodes
proc cbrtraffic { src dst interval starttime } {
  global ns_ node_
  set udp($src) [new Agent/UDP]                    ;#set udp [new Agent/UDP]
  eval $ns_ attach-agent \$node_($src) \$udp($src)          ;# $ns_ attach-agent
<node> <agent>
  set null($dst) [new Agent/Null]
  eval $ns_ attach-agent \$node_($dst) \$null($dst)
  set cbr($src) [new Application/Traffic/CBR]
  eval \$cbr($src) set packetSize_ 70
  eval \$cbr($src) set interval_ $interval
  eval \$cbr($src) set random_ 0
  eval \$cbr($src) set maxpkts_ 10000
  eval \$cbr($src) attach-agent \$udp($src)
  eval $ns_ connect \$udp($src) \$null($dst)
  $ns_ at $starttime "$cbr($src) start"
```

```
}

if { ("$val(traffic)" == "cbr")} {
  puts "\nTraffic: $val(traffic)"
  #Mac/802_15_4 wpanCmd ack4data on
  puts [format "Acknowledgement for data: %s" [Mac/802_15_4 wpanCmd ack4data]]
  set lowSpeed 0.5ms
  set highSpeed 1.5ms
  Mac/802_15_4 wpanNam PlaybackRate $lowSpeed
  $ns_ at [expr $appTime1+0.1] "Mac/802_15_4 wpanNam PlaybackRate $highSpeed"
  $ns_ at $appTime2 "Mac/802_15_4 wpanNam PlaybackRate $lowSpeed"
  $ns_ at [expr $appTime2+0.1] "Mac/802_15_4 wpanNam PlaybackRate $highSpeed"
  $ns_ at $appTime3 "Mac/802_15_4 wpanNam PlaybackRate $lowSpeed"
  $ns_ at [expr $appTime3+0.1] "Mac/802_15_4 wpanNam PlaybackRate $highSpeed"
  eval $val(traffic)traffic 19 6 0.2 $appTime1
  eval $val(traffic)traffic 10 4 0.2 $appTime2
  eval $val(traffic)traffic 3 2 0.2 $appTime3
  Mac/802_15_4 wpanNam FlowClr -p cbr -c tomato
  Mac/802_15_4 wpanNam FlowClr -p ARP -c green
  if { "$val(traffic)" == "cbr" } {
    set pktType cbr
  } else {
    set pktType exp
  }
  Mac/802_15_4 wpanNam FlowClr -p $pktType -s 19 -d 6 -c blue
  Mac/802_15_4 wpanNam FlowClr -p $pktType -s 10 -d 4 -c green4
  Mac/802_15_4 wpanNam FlowClr -p $pktType -s 3 -d 2 -c cyan4
  $ns_ at $appTime1 "$node_(19) NodeClr blue"
  $ns_ at $appTime1 "$node_(6) NodeClr blue"
  $ns_ at $appTime1 "$ns_ trace-annotate \"(at $appTime1) $val(traffic) traffic from node 19 to node 6\""
  $ns_ at $appTime2 "$node_(10) NodeClr green4"
  $ns_ at $appTime2 "$node_(4) NodeClr green4"
  $ns_ at $appTime2 "$ns_ trace-annotate \"(at $appTime2) $val(traffic) traffic from node 10 to node 4\""
  $ns_ at $appTime3 "$node_(3) NodeClr cyan3"
```

```
   $ns_ at $appTime3 "$node_(2) NodeClr cyan3"
   $ns_ at $appTime3 "$ns_ trace-annotate \"(at $appTime3) $val(traffic) traffic
from node 3 to node 2\""
}
# defines the node size in nam
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ initial_node_pos $node_($i) 2
}
# Tell nodes when the simulation ends
for {set i 0} {$i < $val(nn) } {incr i} {
   $ns_ at $stopTime "$node_($i) reset";
}
$ns_ at $stopTime "stop"
$ns_ at $stopTime "puts \"\nNS EXITING...\""
$ns_ at $stopTime "$ns_ halt"

proc stop {} {
   global ns_ tracefd val env
   $ns_ flush-trace
   close $tracefd
   set hasDISPLAY 0
   foreach index [array names env] {
      #puts "$index: $env($index)"
      if { ("$index" == "DISPLAY") && ("$env($index)" != "") } {
            set hasDISPLAY 1
      }
   }
   if { ("$val(nam)" == "aodv.nam") && ("$hasDISPLAY" == "1") } {
       exec nam aodv.nam &
   }
}
puts "\nStarting Simulation..."
$ns_ run
```

- **25 nodes control.scn file**

```
# Nodes: 25
$node_(0) set X_ 6.608377307314
$node_(0) set Y_ 40.446991827566
$node_(0) set Z_ 0.000000000000
$node_(1) set X_ 35.337311778721
$node_(1) set Y_ 33.582820874924
$node_(1) set Z_ 0.000000000000
$node_(2) set X_ 27.437938215931
$node_(2) set Y_ 30.091658830425
$node_(2) set Z_ 0.000000000000
$node_(3) set X_ 19.568766842667
$node_(3) set Y_ 39.453389414657
$node_(3) set Z_ 0.000000000000
$node_(4) set X_ 31.496269526711
$node_(4) set Y_ 20.257450139045
$node_(4) set Z_ 0.000000000000
$node_(5) set X_ 14.476535726599
$node_(5) set Y_ 34.646396828920
$node_(5) set Z_ 0.000000000000
$node_(6) set X_ 40.430596917301
$node_(6) set Y_ 35.320042376073
$node_(6) set Z_ 0.000000000000
$node_(7) set X_ 20.045665842364
$node_(7) set Y_ 32.537008965783
$node_(7) set Z_ 0.000000000000
$node_(8) set X_ 13.641089465783
$node_(8) set Y_ 28.871428688349
$node_(8) set Z_ 0.000000000000
$node_(9) set X_ 40.125856848941
$node_(9) set Y_ 27.049644538277
$node_(9) set Z_ 0.000000000000
$node_(10) set X_ 34.596871605178
$node_(10) set Y_ 39.816174228621
$node_(10) set Z_ 0.000000000000
$node_(11) set X_ 8.828197591068
$node_(11) set Y_ 34.402175912553
$node_(11) set Z_ 0.000000000000
$node_(12) set X_ 18.759762507870
```

```
$node_(12) set Y_ 25.069997591097
$node_(12) set Z_ 0.000000000000
$node_(13) set X_ 33.277017351151
$node_(13) set Y_ 26.522149009306
$node_(13) set Z_ 0.000000000000
$node_(14) set X_ 28.935083056450
$node_(14) set Y_ 37.156742480258
$node_(14) set Z_ 0.000000000000
$node_(15) set X_ 23.507724673396
$node_(15) set Y_ 23.566772846479
$node_(15) set Z_ 0.000000000000
$node_(16) set X_ 20.236128396148
$node_(16) set Y_ 45.484419961171
$node_(16) set Z_ 0.000000000000
$node_(17) set X_ 25.709636471608
$node_(17) set Y_ 44.353113573382
$node_(17) set Z_ 0.000000000000
$node_(18) set X_ 45.610349971557
$node_(18) set Y_ 29.787732692400
$node_(18) set Z_ 0.000000000000
$node_(19) set X_ 16.345058881159
$node_(19) set Y_ 22.614188426582
$node_(19) set Z_ 0.000000000000
$node_(20) set X_ 35.972132273531
$node_(20) set Y_ 20.620805728046
$node_(20) set Z_ 0.000000000000
$node_(21) set X_ 40.626765649851
$node_(21) set Y_ 22.028905860151
$node_(21) set Z_ 0.000000000000
$node_(22) set X_ 14.914806804685
$node_(22) set Y_ 24.094071765795
$node_(22) set Z_ 0.000000000000
$node_(23) set X_ 28.443020411336
$node_(23) set Y_ 24.367299319338
$node_(23) set Z_ 0.000000000000
$node_(24) set X_ 14.102836431793
$node_(24) set Y_ 39.394995544608
$node_(24) set Z_ 0.000000000000
```

```
#movement process at 1.5m/s speed

$ns_ at 2.0 "$node_(1) setdest 20.0 44.0 1.5"
$ns_ at 2.0 "$node_(2) setdest 34.0 43.0 1.5"
$ns_ at 2.0 "$node_(9) setdest 32.0 36.0 1.5"
$ns_ at 2.0 "$node_(7) setdest 26.0 46.0 1.5"
$ns_ at 2.0 "$node_(8) setdest 45.0 47.0 1.5"
$ns_ at 2.0 "$node_(4) setdest 37.0 30.0 1.5"
$ns_ at 2.0 "$node_(11) setdest 1.0 4.0 1.5"
$ns_ at 2.0 "$node_(12) setdest 21.0 34.0 1.5"
$ns_ at 2.0 "$node_(18) setdest 22.0 23.0 1.5"
$ns_ at 2.0 "$node_(19) setdest 49.0 45.0 1.5"
$ns_ at 2.0 "$node_(21) setdest 41.0 23.0 1.5"
$ns_ at 2.0 "$node_(23) setdest 49.0 49.5 1.5"
$ns_ at 2.0 "$node_(24) setdest 48.0 34.0 1.5"
```