# Parameters Settings

set val(chan) Channel/WirelessChannel ; # Channel Model: Wireless

set val(prop) Propagation/Shadowing ; # Propragation Model: Shadowing(Shadowing/TwoRayGround/FreeSpace)

set val(netif) Phy/WirelessPhy/802\_15\_4

set val(mac) Mac/802\_15\_4

set val(ifq) Queue/DropTail/PriQueue ; # Queue Model: Drop at Tail

set val(ll) LL ; # Link Layer Model

set val(ant) Antenna/OmniAntenna ; # Antenna Model

set val(ifqlen) 100 ; # Max Number of Queue

set val(nn) 56 ; # Number of Nodes

set val(rp) AODV ; # Routing Protocol: AODV (DSR/ZBR/AOMDV/AODV)

set val(x) 100 ; # Center Position of nam

set val(y) 100

set val(tr) zigbee.tr ; # Tracing File

set val(nam) zigbee.nam ; # Nam File

set val(traffic) cbr ; # Data Flow: cbr (cbr/poisson/ftp)

set val(trInterval) 0.01 ; # Time Interval between Packets

set val(startInterval) 0.5 ; # Start Time

set stopTime 100 ; # Stop Time

# Input

proc getCmdArgu {argc argv} {

global val

for {set i 0} {$i < $argc} {incr i} {

set arg [lindex $argv $i]

if {[string range $arg 0 0] != "-"} continue

set name [string range $arg 1 end]

set val($name) [lindex $argv [expr $i+1]]

}

}

getCmdArgu $argc $argv

set ns\_ [new Simulator]

set tracefd [open ./$val(tr) w]

$ns\_ trace-all $tracefd

if { "$val(nam)" == "zigbee.nam" } {

set namtrace [open ./$val(nam) w]

$ns\_ namtrace-all-wireless $namtrace $val(x) $val(y)

}

$ns\_ puts-nam-traceall {# ZigBee #

}

Mac/802\_15\_4 wpanCmd verbose on

Mac/802\_15\_4 wpanNam namStatus on

set topo [new Topography]

$topo load\_flatgrid $val(x) $val(y)

set god\_ [create-god $val(nn)]

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) \

-topoInstance $topo \

-agentTrace OFF \

-routerTrace OFF \

-macTrace ON \

-movementTrace OFF \

-energyModel "EnergyModel" \

-initialEnergy 1000 \

-rxPower 35.28e-3 \

-txPower 31.32e-3 \

-idlePower 712e-6 \

-sleepPower 144e-9 \

-channel $chan\_1\_

for {set i 0} {$i < $val(nn) } {incr i} {

set node\_($i) [$ns\_ node]

$node\_($i) random-motion 0

}

# Topology Input

source ./zigbee\_topology.scn

# Start Coordinate/Normal Nodes

$ns\_ at 0.0 "$node\_(0) NodeLabel \"PAN Coor\""

$ns\_ at 0.0 "$node\_(0) sscs startCTPANCoord"

for {set i 1} {$i < 52} {incr i} {

$ns\_ at [expr $i\*($val(startInterval))] "$node\_($i) sscs startCTDevice"

}

for {set i 52} {$i < 56} {incr i} {

$ns\_ at [expr $i\*($val(startInterval))] "$node\_($i) sscs startCTDevice"

}

# Runing Speed of Nam

Mac/802\_15\_4 wpanNam PlaybackRate 5ms

$ns\_ at 30.0 "Mac/802\_15\_4 wpanNam PlaybackRate 10.0ms"

$ns\_ at 40.0 "Mac/802\_15\_4 wpanNam PlaybackRate 100.0ms"

$ns\_ at 50.0 "puts \"\nTransmitting data ...\n\""

# Setup UDP and CBR

proc cbrtraffic { src dst interval starttime } {

global ns\_ node\_

set udp\_($src) [new Agent/UDP]

eval $ns\_ attach-agent \$node\_($src) \$udp\_($src)

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\_ 100

eval \$cbr\_($src) set interval\_ $interval

eval \$cbr\_($src) set random\_ 0

#eval \$cbr\_($src) set maxpkts\_ 1000

eval \$cbr\_($src) attach-agent \$udp\_($src)

eval $ns\_ connect \$udp\_($src) \$null\_($dst)

$ns\_ at $starttime "$cbr\_($src) start"

}

# ACK

if {$val(rp) == "ZBR"} {

# 0=No ACK; 1=ACK at failure (default); 2=ACK at success/failure

Mac/802\_15\_4 wpanCmd callBack 2;

}

if { ("$val(traffic)" == "cbr") } {

puts "\nTraffic: $val(traffic)"

puts [format "Acknowledgement for data: %s" [Mac/802\_15\_4 wpanCmd ack4data off]]

# Start CBR

for {set i 1} {$i < 56} {incr i} {

$val(traffic)traffic 0 $i $val(trInterval) [expr 30+$i])

}

# Color of Nodes in Nam

Mac/802\_15\_4 wpanNam FlowClr -p AODV -c green

Mac/802\_15\_4 wpanNam FlowClr -p ARP -c tomato

Mac/802\_15\_4 wpanNam FlowClr -p MAC -c navy

}

# Size of Nodes in Nam

for {set i 0} {$i < $val(nn)} {incr i} {

$ns\_ initial\_node\_pos $node\_($i) 8

}

# Reset of Nodes

for {set i 0} {$i < $val(nn) } {incr i} {

$ns\_ at $stopTime "$node\_($i) reset";

}

# Stop Simulation

$ns\_ at $stopTime "stop"

$ns\_ at $stopTime "puts \"\nNS EXITING...\n\""

$ns\_ at $stopTime "$ns\_ halt"

# Stop Function

proc stop {} {

#global ns\_ tracefd starttime(1) val env

global ns\_ tracefd appTime 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 { ("$val(nam)" == "zigbee.nam") && ("$hasDISPLAY" == "1") } {

exec nam zigbee.nam &

}

}

# Start Simulation

puts "\nStarting Simulation..."

$ns\_ run