

ДОДАТОК А

Програмний код сценарію моделювання

```
# Параметри моделі
#
set val(chan)          Channel/WirelessChannel ;# клас каналу
set val(prop)          Propagation/TwoRayGround ;# модель
розповсюдження сигналу
set val(netif)         Phy/WirelessPhy/802_15_4 ;# клас мережевого
інтерфейсу
set val(mac)           Mac/802_15_4 ;# клас MAC
set val(ifq)           Queue/DropTail/PriQueue ;# клас черги
інтерфейсу
set val(ll)            LL/LL802_15_4 ;# клас рівню LL
set val(ant)           Antenna/OmniAntenna ;# клас антени
set val(rp)            NOAH ;# клас агенту
маршрутизації
set val(ifqlen)        3 ;# розмір черги
інтерфейсу
set val(x)             10 ;# X розмір поля
розташування
set val(y)             10 ;# Y розмір поля
розташування
set val(assocStart)    0.6 ;# час початку
асоціації
set val(assocTime)     1.3 ;# час асоціації
одного вузл

set val(beacon_enabled) 1 ;# включити режим beacon-enabled
set val(BO)            "5" ;# beacon order
set val(SO)            "5" ;# superframe order
set val(GTS_setting)   0x8E ;# параметри GTS: 8 слотів

Agent/NOAH set be_random_ 0 ;# відключити jitter NOAH

# Параметри командної строки
set argNN [lindex $argv 0] ;# кількість вузлів
set argSizeBack [lindex $argv 1] ;# розмір пакету від
Клієнта
set argIntervalBack [lindex $argv 2] ;# інтервал пакетів від
Клієнта
set argSize [lindex $argv 3] ;# розмір пакету від
Сервера

set val(nn) [expr {$argNN != "" ? $argNN : 2}] ;# кількість
вузлів >= 2

# Час початку та закінчення відправки трафіка
set val(operationStart) [expr $val(assocStart) + $val(assocTime)
* $val(nn)]
```

```

set val(stop)                [expr $val(operationStart) + 10.1]

# Параметри трафіка від Сервера до Клієнтів
set val(bmsg-interval) 0.012
;# інтервал, с
set val(bmsg-size)      [expr {$argSize != "" ? $argSize : 120}]
;# розмір, байт
set val(bmsg-start)     $val(operationStart)
;# час початку
set val(bmsg-stop)      [expr $val(stop) - 0.1]
;# час останову

# Параметри трафіка від Клієнтів до Сервера
set val(pois-interval) [expr {$argIntervalBack != "" ?
$argIntervalBack : 3}]
set val(pois-size)      [expr {$argSizeBack != "" ? $argSizeBack
: 10}]
set val(pois-rate)      250
set val(pois-start)     $val(operationStart)
set val(pois-stop)      [expr $val(stop) - 0.1]

#
# Створення моделі
#

set namtracename      backtraffic_test.nam      ;# назва trace-файлу

# Створення стандартних об'єктів середовища моделювання
set ns                [new Simulator]
set tracefd           [open backtraffic_test.tr w]
set namtrace           [open $namtracename w]

$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

$ns puts-nam-traceall {# nam4wpan #}

Mac/802_15_4 wpanCmd verbose on
Mac/802_15_4 wpanNam namStatus on

# Параметри моделі розповсюдження
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 CStresh_ $dist(40m)
Phy/WirelessPhy set RXThresh_ $dist(40m)

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

create-god $val(nn)

# Завантаження параметрів MobileNode
$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 [new $val(chan)] \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF \
    -rxPower 35.28e-3 \
    -txPower 31.32e-3 \
    -idlePower 712e-6 \
    -sleepPower 144e-9

# Створення об'єктів MobileNode
for {set i 0} {$i < $val(nn)} { incr i } {
    set mnode_($i) [$ns node]
}

# Функція, що встановлює флаг GTS_delivery_ на рівні LL вузла
proc setNodeGTS {node gts} {
    set mac [ [set node] getMac 0 ]
    puts $mac
    set ll [$mac up-target]
    puts $ll
    $ll set GTS_delivery_ $gts
}

if {[info exists val(GTS_setting)]} {
    # Сервер веде передачу протягом CFP в інтервалах GTS
    setNodeGTS $mnode_(0) 1
}

```

```

# Функція, що встановлює розмір черги інтерфейсу окремого
об'єкту MobileNode
proc setNodeIfqLen {node qlen} {
    set mac [ [set node] getMac 0]
    puts $mac
    set ll [$mac up-target]
    puts $ll
    set ifq [$ll down-target]
    $ifq set limit_ $qlen
}

# Розмір черги інтерфейсу Сервера = 1000
setNodeIfqLen $mnode_(0) 1000

# Створення таблиці маршрутизації для NOAH на Сервері
set cmd "[ $mnode_(0) set ragent_ routing $val(nn) 0 0"
for {set to 1} {$to < $val(nn)} {incr to} {
    set hop $to
    set cmd "$cmd $to $hop"
}
eval $cmd

# Створення таблиць маршрутизації для NOAH на Клієнтах
for {set i 1} {$i < $val(nn)} {incr i} {
    set cmd "[ $mnode_($i) set ragent_ routing $val(nn)"
    for {set to 0} {$to < $val(nn)} {incr to} {
        if {$to == $i} {
            set hop $to
        } else {
            set hop 0
        }
        set cmd "$cmd $to $hop"
    }
    eval $cmd
}

# Розташування вузлів Клієнтів випадково на площині
for {set i 1} {$i < $val(nn)} {incr i} {
    $mnode_($i) set X_ [ expr {$val(x) * rand()} ]
    $mnode_($i) set Y_ [ expr {$val(y) * rand()} ]
    $mnode_($i) set Z_ 0
}

# Розташування Сервера посередині площини
$mnode_(0) set X_ [ expr {$val(x)/2} ]
$mnode_(0) set Y_ [ expr {$val(y)/2} ]
$mnode_(0) set Z_ 0.0
$mnode_(0) label "Sink"

for {set i 0} {$i < $val(nn)} {incr i} {
    $ns initial_node_pos $mnode_($i) 10
}

```

```

#
# Об'єднання вузлів у мережу
#

# Старт координатора
$ns at 0.0 "$mnode_(0) sscs startPANCoord $val(beacon_enabled)
$val(BO) $val(SO) "

if {[info exists val(GTS_setting)]} {
    # Встановлення параметрів GTS
    $ns at $val(assocStart) "$mnode_(0) sscs MLME_GTS_indication
0 [expr {$val(GTS_setting)}]"
}

# Підключення Клієнтів до Сервера
for {set i 1} {$i < $val(nn)} { incr i } {
    set t [expr $val(assocStart) + $val(assocTime) * ($i - 1)]
    $ns at $t "$mnode_($i) sscs startDevice $val(beacon_enabled)
0 0 $val(BO) $val(SO) "
}

#
# Налаштування генераторів трафіку
#

# Створення агентів ширококомовного трафіку
for {set i 0} {$i < $val(nn)} { incr i } {
    set agent($i) [new Agent/Broadcastbase]
    $mnode_($i) attach $agent($i) 250
    $agent($i) set fid_ $i
    set game($i) [new Application/BroadcastbaseApp]
    $game($i) set bsize_ $val(bmsg-size)
    $game($i) set bmsg-interval_ $val(bmsg-interval)
    $game($i) set propagate_ 0
    $game($i) attach-agent $agent($i)
}

# Запуск та останів відправки ширококомовного трафіку Сервером у
запланований час
$ns at $val(bmsg-start) "$game(0) start "
$ns at $val(bmsg-stop) "$game(0) stop "

# Функція налаштування пуасоновського генератора трафіку
proc poissontraffic { src dst } {
    global ns val mnode_
    set udp($src) [new Agent/UDP]
    eval $ns attach-agent \"$mnode_($src) \"$udp($src)
    set null($dst) [new Agent/Null]
    eval $ns attach-agent \"$mnode_($dst) \"$null($dst)
    set expl($src) [new Application/Traffic/Exponential]
    eval \"$expl($src) set packetSize_ \"$val(pois-size)
    eval \"$expl($src) set burst_time_ 0

```

```

    eval \${expl($src)} set idle_time_ [expr \${val(pois-
interval)*1000.0-\${val(pois-size)*8/\${val(pois-rate)]ms      ;#
idle_time + pkt_tx_time = interval
    eval \${expl($src)} set rate_ \${val(pois-rate)}k
    eval \${expl($src)} attach-agent \${udp($src)}
    eval $ns connect \${udp($src)} \${null($dst)}
    $ns at \${val(pois-start)} "\${expl($src)} start"
    $ns at \${val(pois-stop)} "\${expl($src)} stop"
}

# Створення генераторів трафіку від Клієнтів до Сервера
for {set i 1} {$i < \${val(nn)}} { incr i } {
    poissontraffic $i 0
}

# Планування, коли зупинити роботу вузлів
for {set i 0} {$i < \${val(nn)} } { incr i } {
    $ns at \${val(stop)} "\${mnode_($i)} reset;"
}

# Планування останову моделі
$ns at \${val(stop)} "$ns nam-end-wireless \${val(stop)}"
$ns at \${val(stop)} "stop"
$ns at [expr \${val(stop)} + 0.01] "puts \"end simulation\"; $ns
halt"
proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
}

# Запуск моделювання
$ns run

```