

# Моделирование беспроводной сети в ns-2



Генеральский Д., 22 ноября 2014г.

В данном примере будет разобрано построение простого двухузлового беспроводного сценария. Подвижные узлы движутся по площади, площадь которой  $500*500$  м<sup>2</sup>. Узлы начинают свое движение с противоположных концов. Они движутся навстречу друг другу в первой половине симуляции и в противоположных направлениях во второй. Между ними при этом устанавливается TCP соединение. Пакеты пересылаются между узлами, когда они находятся в зоне радиопокрытия. Когда они расходятся на большое расстояние, пакеты начинают теряться.

Подвижный узел состоит из следующих компонентов: Канальный Уровень, Интерфейс Очереди, MAC уровень, беспроводной канал. Перед началом симуляции необходимо определить их параметры. Кроме того, необходимо задать и другие параметры, такие как тип антенны, модель радиораспространения, тип протокола маршрутизации ad-hoc, используемый подвижными узлами и т.д. Их краткое описание указано в комментариях:

```
# =====  
# Define options  
# =====  
  
set val(chan)    Channel/WirelessChannel ;# channel type  
set val(prop)    Propagation/TwoRayGround ;# radio-propagation model  
set val(ant)     Antenna/OmniAntenna    ;# Antenna type  
set val(ll)      LL                      ;# Link layer type  
set val(ifq)     Queue/DropTail/PriQueue ;# Interface queue type  
set val(ifqlen)  50                      ;# max packet in ifq  
set val(netif)   Phy/WirelessPhy        ;# network interface type  
set val(mac)     Mac/802_11              ;# MAC type  
set val(rp)      DSDV                    ;# ad-hoc routing protocol  
set val(mn)      2                      ;# number of mobilenodes
```

Теперь можно перейти к главной части программы:

```
set ns_ [new Simulator]
```

Затем создадим файл трассировки и используем для него функцию trace-all:

```
set tracefd [open simple.tr w]
```

```
$ns_ trace-all $tracefd
```

Затем создадим объект топологии, который хранит пути передвижения подвижных узлов внутри заданной границы:

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

Так как в данном примере подвижные узлы передвигаются внутри топологии 500 на 500 м, то зададим объекту координаты X и Y:

```
$topo load_flatgrid 500 500
```

Топография разбивается на сетки, и значения разрешения сетки по умолчанию равно 1. Другое значение может быть задано третьим параметром предыдущей функции.

Далее создадим объект God:

*create-god \$val(m)*

God (General Operations Director) – это объект, который используется для хранения глобальной информации о состоянии среды, сети или узлов, которые хотелось бы иметь, но которые не должны быть известны любому участнику симуляции. Фактически, God хранит общее число подвижных узлов и таблицу с наименьшим числом скачков, требуемых для перехода от одного узла к другому.

Из-за простоты данного примера нам не нужна информация о скачках, однако поскольку God объект объявлен внутри MAC объектов на подвижных узлах, мы должны создать God.

Далее необходимо создать подвижные узлы. Для начала, необходимо сконфигурировать узлы. Конфигурирование узлов API (Applications Programming Interface – интерфейс взаимодействия между транспортным агентом и приложением) состоит из определения типа адресации, типа протокола маршрутизации, канального уровня, MAC уровня. Например:

*(parameter examples)*

```
# $ns_ node-config -addressingType flat or hierarchical or expanded
```

```
# -adhocRouting DSDV or DSR or TORA
```

```
# -llType LL
```

```
# -macType Mac/802_11
```

```
# -propType "Propagation/TwoRayGround"
```

```
# -ifqType "Queue/DropTail/PriQueue"
```

```
# -ifqLen 50
```

```
# -phyType "Phy/WirelessPhy"
```

```
# -antType "Antenna/OmniAntenna"
```

```
# -channelType "Channel/WirelessChannel"
```

```
# -topoInstance $topo
```

```
# -energyModel "EnergyModel"
```

```
# -initialEnergy (in Joules)
```

```
# -rxPower (in W)
```

```
# -txPower (in W)
```

```
# -agentTrace ON or OFF
```

```
# -routerTrace ON or OFF
```

```
# -macTrace ON or OFF
```

```
# -movementTrace ON or OFF
```

Все значения параметров по умолчанию нулевые, кроме: addressingType: flat.

Конфигурация API для создания подвижных узлов в данном случае выглядит следующим образом:

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

```

Далее создадим два подвижных узла:

```

for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns_node]
    $node_($i) random-motion 0 ;# disable random motion
}

```

Рандомное передвижение отключено, так как ниже следует задать параметры передвижения (скорость и направление).

Зададим начальные координаты:

```

#
# Provide initial (X,Y, for now Z=0) co-ordinates for node_(0) and node_(1)
#
$node_(0) set X_ 5.0
$node_(0) set Y_ 2.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 390.0
$node_(1) set Y_ 385.0
$node_(1) set Z_ 0.0

```

Узел 0 стартует из точки с координатами (5,2), а узел 1 – (390, 385).

Произведем некоторые передвижения узла:

```

#
# Node_(1) starts to move towards node_(0)
#
$ns_at 50.0 "$node_(1) setdest 25.0 20.0 15.0"
$ns_at 10.0 "$node_(0) setdest 20.0 18.0 1.0"

# Node_(1) then starts to move away from node_(0)
$ns_at 100.0 "$node_(1) setdest 490.0 480.0 15.0"

```

Так `$ns_at 50.0 "$node_(1) setdest 25.0 20.0 15.0"` означает, что при достижении времени симуляции 50 мс, узел 1 начнет двигаться в точку с координатами (25, 20) со скоростью 15 м/с.

Далее установим поток трафика между узлами:

```
# TCP connections between node_(0) and node_(1)
```

```
set tcp [new Agent/TCP]
```

```
$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"
```

Эти строки устанавливают TCP соединение между узлами с источником TCP трафика, находящимся в узле 0.

Затем необходимо определить время окончания симуляции и дать команду подвижным узлам перезагрузиться, которая перезагрузит их внутренние сетевые компоненты.

```
#
```

```
# Tell nodes when the simulation ends
```

```
#
```

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

```
    $ns_ at 150.0 "$node_($i) reset";
```

```
}
```

```
$ns_ at 150.0001 "stop"
```

```
$ns_ at 150.0002 "puts \"NS EXITING...\" ; $ns_ halt"
```

```
proc stop {} {
```

```
    global ns_ tracefd
```

```
    close $tracefd
```

```
}
```

На 150-й секунде симуляция прекратится. Узлы перезагрузятся в это время и вызовется переменная \$ns\_ halt на 150.0002 секунде, немного позже перезагрузки узлов.

Остается только добавить команду начала симуляции:

```
puts "Starting Simulation..."
```

```
$ns_ run
```

Результатом симуляции такой модели будет создание файла трассировки simple.tr. Если мы обратимся к графе AgentTrace/RouterTrace, мы увидим DSDV (DSDV - Destination Sequence Distance Vector – протокол маршрутизации, при котором узлы, находящиеся в пределах зоны радиопокрытия друг друга, обмениваются сообщениями маршрутизации) сообщения маршрутизации и TCP пакеты, принимаемые и отправляемые маршрутизатором и агентом в узле 0 и 1.

```
s 0.029290548 _1_RTR --- 0 message 32 [0 0 0 0] ----- [1:255 -1:255 32 0]
```

```
s 1.119926192 _0_RTR --- 1 message 32 [0 0 0 0] ----- [0:255 -1:255 32 0]
```

Мы видим, что на 10 секунде поток TCP пакетов отправляется из узла 0. Первоначально оба узла слишком удалены друг от друга, и эти пакеты удаляются из узла 0, но не достигают узла 1. Примерно на 81-й секунде между узлами начинается обмен информацией маршрутизации, и примерно на 100-й секунде можно увидеть первый пакет, принятый агентом узла 1, который отправляет подтверждение ACK обратно узлу 0 и таким образом, устанавливается TCP соединение.

```
r 100.001596889 _1_ AGT --- 3824 tcp 1060 [13a 1 0 800] ----- [0:0 1:0 32 1] [1908 0] 1 0
s 100.001596889 _1_ AGT --- 3855 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [1908 0] 0 0
r 100.001596889 _1_ RTR --- 3855 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [1908 0] 0 0
s 100.001596889 _1_ RTR --- 3855 ack 60 [0 0 0 0] ----- [1:0 0:0 32 0] [1908 0] 0 0
r 100.003680943 _0_ AGT --- 3851 ack 60 [13a 0 1 800] ----- [1:0 0:0 32 0] [1906 0] 1 0
s 100.003680943 _0_ AGT --- 3856 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [1926 0] 0 0
r 100.003680943 _0_ RTR --- 3856 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [1926 0] 0 0
s 100.003680943 _0_ RTR --- 3856 tcp 1060 [0 0 0 0] ----- [0:0 1:0 32 1] [1926 0] 0 0
r 100.006105016 _0_ AGT --- 3853 ack 60 [13a 0 1 800] ----- [1:0 0:0 32 0] [1907 0] 1 0
s 100.006105016 _0_ AGT --- 3857 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [1927 0] 0 0
r 100.006105016 _0_ RTR --- 3857 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [1927 0] 0 0
s 100.006105016 _0_ RTR --- 3857 tcp 1060 [0 0 0 0] ----- [0:0 1:0 32 1] [1927 0] 0 0
r 100.016129071 _1_ AGT --- 3825 tcp 1060 [13a 1 0 800] ----- [0:0 1:0 32 1] [1909 0] 1 0
s 100.016129071 _1_ AGT --- 3858 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [1909 0] 0 0
r 100.016129071 _1_ RTR --- 3858 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [1909 0] 0 0
s 100.016129071 _1_ RTR --- 3858 ack 60 [0 0 0 0] ----- [1:0 0:0 32 0] [1909 0] 0 0
r 100.026133146 _1_ AGT --- 3827 tcp 1060 [13a 1 0 800] ----- [0:0 1:0 32 1] [1910 0] 1 0
```

Однако как только узлы начинают удаляться друг от друга, соединение разрывается примерно на 116-й секунде, и пакеты начинают пропадать.

```
s 116.002082380 _1_ AGT --- 6430 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [3202 0] 0 0
r 116.002082380 _1_ RTR --- 6430 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [3202 0] 0 0
s 116.002082380 _1_ RTR --- 6430 ack 60 [0 0 0 0] ----- [1:0 0:0 32 0] [3202 0] 0 0
r 116.005616463 _0_ AGT --- 6396 ack 60 [13a 0 1 800] ----- [1:0 0:0 32 0] [3185 0] 1 0
s 116.005616463 _0_ AGT --- 6431 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [3205 0] 0 0
r 116.005616463 _0_ RTR --- 6431 tcp 1040 [0 0 0 0] ----- [0:0 1:0 32 0] [3205 0] 0 0
s 116.005616463 _0_ RTR --- 6431 tcp 1060 [0 0 0 0] ----- [0:0 1:0 32 1] [3205 0] 0 0
```

Подробное содержание файла трассировки было изложено в примере 1.

#### Список литературы:

1. <http://www.isi.edu/nsnam/ns/tutorial/index.html>
2. [http://www.isi.edu/nsnam/ns/doc/ns\\_doc.pdf](http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf)

