

Лаб - 1 Простые модели компьютерной сети

Адабор Кристофер Твум

Российский университет дружбы народов, Москва, Россия

Информация

- Адабор Кристофер Твум
 - студент
 - Российский университет дружбы народов
 - 1032225824@pfur.ru
- <https://github.com/Adabor/-1->

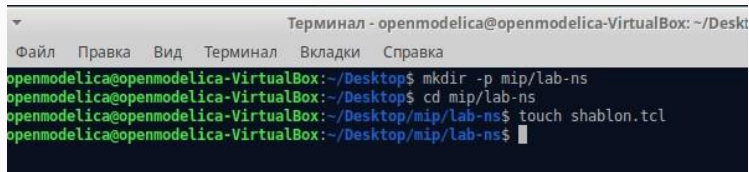


Приобрести навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также проанализировать полученные результаты моделирования.

-
1. Создать шаблон сценария для NS-2;

2. Выполнить простой пример описания топологии сети, состоящей из двух узлов и одного соединения;
3. Выполнить пример с усложнённой топологией сети;
4. Выполнить пример с кольцевой топологией сети;
5. Выполнить упражнение.

Выполнение лабораторной работы



```
Терминал - openmodelica@openmodelica-VirtualBox: ~/Desk
Файл  Правка  Вид  Терминал  Вкладки  Справка
openmodelica@openmodelica-VirtualBox:~/Desktop$ mkdir -p mip/lab-ns
openmodelica@openmodelica-VirtualBox:~/Desktop$ cd mip/lab-ns
openmodelica@openmodelica-VirtualBox:~/Desktop/mip/lab-ns$ touch shablon.tcl
openmodelica@openmodelica-VirtualBox:~/Desktop/mip/lab-ns$
```

Рис. 1: Создание директорий и файла

Шаблон сценария для NS-2

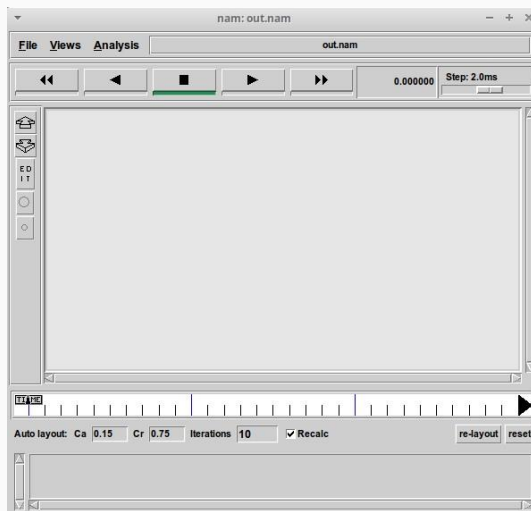
```
# создание объекта Simulator
set ns [new Simulator]

# открытие на запись файла out.nam для визуализатора nam
set nf [open out.nam w]

# все результаты моделирования будут записаны в переменную nf
$ns namtrace-all $nf

# открытие на запись файла трассировки out.tr
# для регистрации всех событий
set f [open out.tr w]
# все регистрируемые события будут записаны в переменную f
$ns trace-all $f
|
# процедура finish закрывает файлы трассировки
# и запускает визуализатор nam
proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam out.nam &
    exit 0
}
# at-событие для планировщика событий, которое запускает
# процедуру finish через 5 с после начала моделирования
$ns at 5.0 "finish"
# запуск модели
$ns run
```

Шаблон сценария для NS-2



запуск шаблона

сценария для NS-2

```

}
# создание 2-х узлов:
set N 2
for {set i 0} {$i < $N} {incr i} {
    set n($i) [$ns node]
}
# соединение 2-х узлов дуплексным соединением
# с полосой пропускания 2 Мб/с и задержкой 10 мс,
# очередь с обслуживанием типа DropTail
$ns duplex-link $n(0) $n(1) 2Mb 10ms DropTail

# создание агента UDP и присоединение его к узлу n0
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0

# создание источника трафика CBR (constant bit rate)
set cbr0 [new Application/Traffic/CBR]
# устанавливаем размер пакета в 500 байт
$cbr0 set packetSize_ 500

# задаем интервал между пакетами равным 0.005 секунды,
# т.е. 200 пакетов в секунду
$cbr0 set interval_ 0.005

# присоединение источника трафика CBR к агенту udp0
$cbr0 attach-agent $udp0

# Создание агента-приёмника и присоединение его к узлу n(1)
set null0 [new Agent/Null]
$ns attach-agent $n(1) $null0

# Соединение агентов между собой
$ns connect $udp0 $null0

# запуск приложения через 0,5 с
$ns at 0.5 "$cbr0 start"
# остановка приложения через 4,5 с
$ns at 4.5 "$cbr0 stop"

```

Рис. 4: Пример описания топологии сети, состоящей из двух узлов и одного соединения

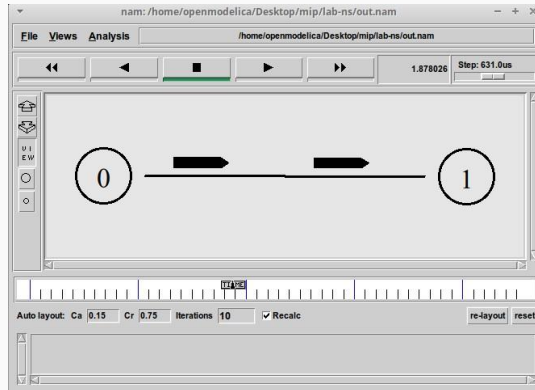
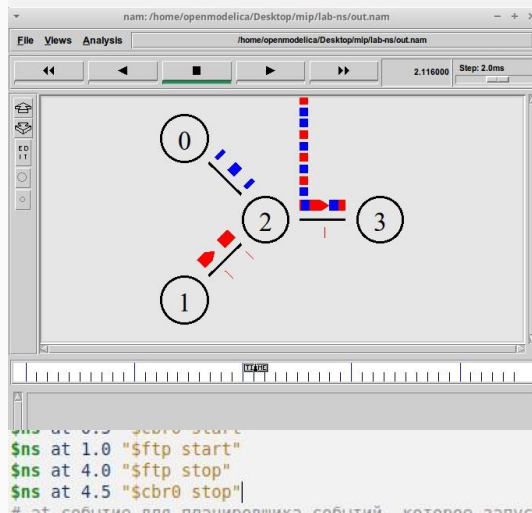


Рис. 5: Визуализация простой модели сети с помощью nam

```
set N 4
for {set i 0} {$i < $N} {incr i} {
  set n($i) [$ns node]
}
$ns duplex-link $n(0) $n(2) 2Mb 10ms DropTail
$ns duplex-link $n(1) $n(2) 2Mb 10ms DropTail
$ns duplex-link $n(3) $n(2) 2Mb 10ms DropTail
$ns duplex-link-op $n(0) $n(2) orient right-down
$ns duplex-link-op $n(1) $n(2) orient right-up
$ns duplex-link-op $n(2) $n(3) orient right
```

Рис. 6: Визуализация простой модели сети с помощью nam

Пример с усложнённой топологией сети



с. 8: Описание

усложненной топологии сети Рис. 9: Описание усложненной топологии сети

Пример с кольцевой топологией сети

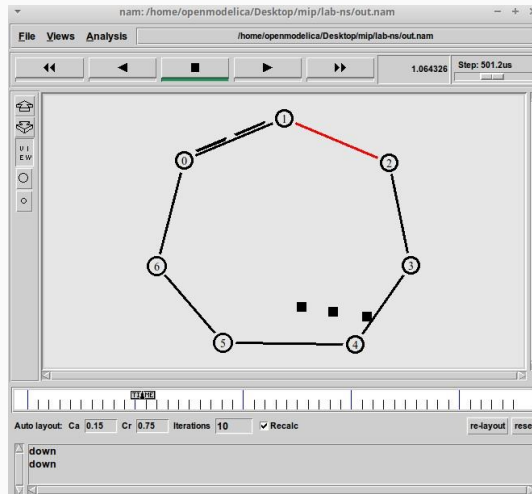


Рис. 10: Описание кольцевой топологии сети и динамической маршрутизацией пакетов Рис. 11: Передача данных по кратчайшему пути сети с кольцевой топологией Рис. 12: Передача данных по сети

с кольцевой топологией в случае разрыва соединения

Добавив в начало скрипта после команды создания объекта Simulator:

```
$ns rtpproto DV
```

увидим, что сразу после запуска в сети отправляется небольшое количество маленьких пакетов, используемых для обмена информацией, необходимой для маршрутизации между узлами.

Пример с кольцевой топологией сети

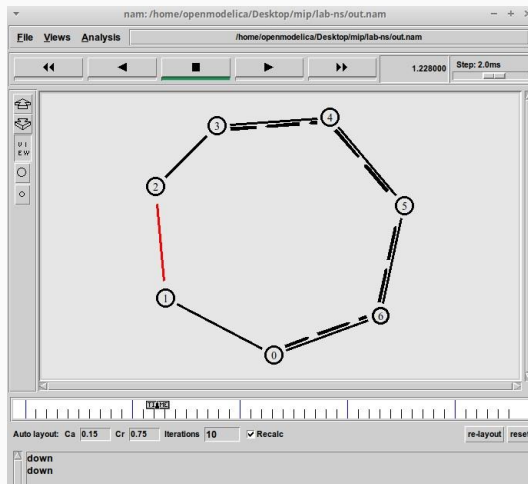


Рис. 13: Маршрутизация данных по сети с кольцевой топологией в случае разрыва соединения

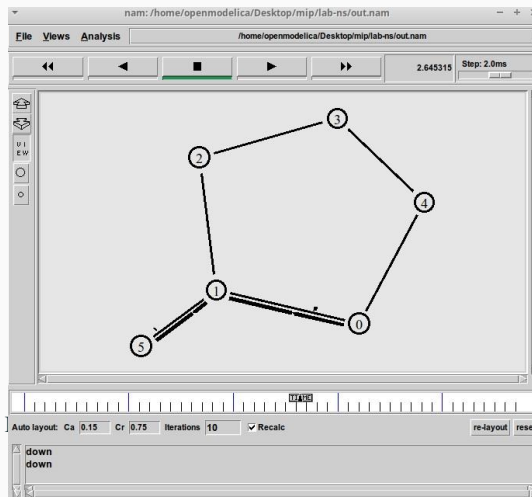


Рис. 14:

и сети Рис. 15:

Передача данных по изменённой кольцевой топологии сети Рис. 16: Передача данных по сети в случае разрыва соединения Рис. 17: Передача данных после восстановления соединения

В процессе выполнения данной лабораторной работы я приобрела навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также проанализировала полученные результаты моделирования.