

Лабораторная работа 1. Простые модели компьютерной сети

Абакумова О. М.

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Абакумова Олеся Максимовна
- Студентка
- Российский университет дружбы народов
- 1132220832@pfur.ru
- <https://github.com/omabakumova>



Цель работы

Цель работы

Приобретение навыков моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также анализ полученных результатов моделирования.

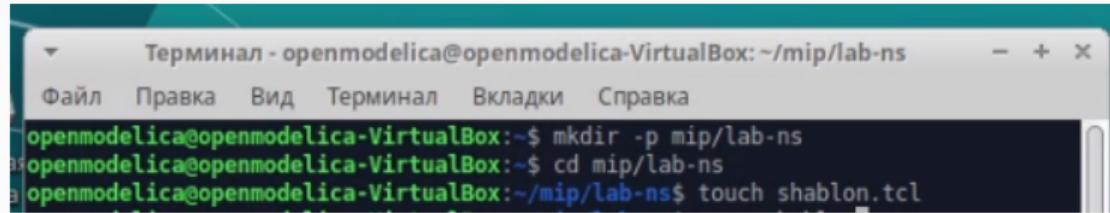
Задания

Задания

- Создать шаблон сценария для NS-2.
- Сделать реализацию простого примера описания топологии сети, состоящей из двух узлов и одного соединения.
- Сделать реализацию примера с усложненной топологией сети.
- Реализовать пример с кольцевой топологией сети.
- Выполнить упражнение.

Выполнение лабораторной работы

Шаблон сценария для NS-2



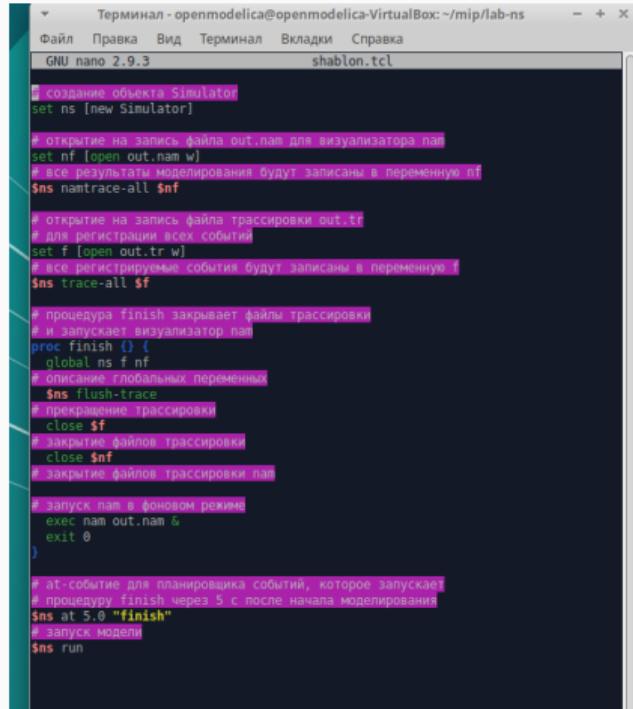
Терминал - openmodelica@openmodelica-VirtualBox:~/mip/lab-ns

Файл Правка Вид Терминал Вкладки Справка

```
openmodelica@openmodelica-VirtualBox:~$ mkdir -p mip/lab-ns
openmodelica@openmodelica-VirtualBox:~$ cd mip/lab-ns
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ touch shablon.tcl
```

Рис. 1: Создание шаблона

Шаблон сценария для NS-2



```
Терминал - openmodelica@openmodelica-VirtualBox: ~/mip/lab-ns
Файл Правка Вид Терминал Вкладки Справка
GNU nano 2.9.3                               shablon.tcl
# 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
    # описание глобальных переменных
    $ns flush-trace
    # прекращение трассировки
    close $f
    # закрытие файлов трассировки
    close $nf
    # закрытие файлов трассировки лям
    close $nam

    # запуск лам в фоновом режиме
    exec nam out.nam &
    exit 0
}

# ат-событие для планировщика событий, которое запускает
# процедуру finish через 5 с после начала моделирования
$ns at 5.0 "finish"
# запуск модели
$ns run
```

Рис. 2: Содержание шаблона

Шаблон сценария для NS-2

```
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ns shablon.tcl
```

Рис. 3: Запуск симуляции шаблона

Шаблон сценария для NS-2

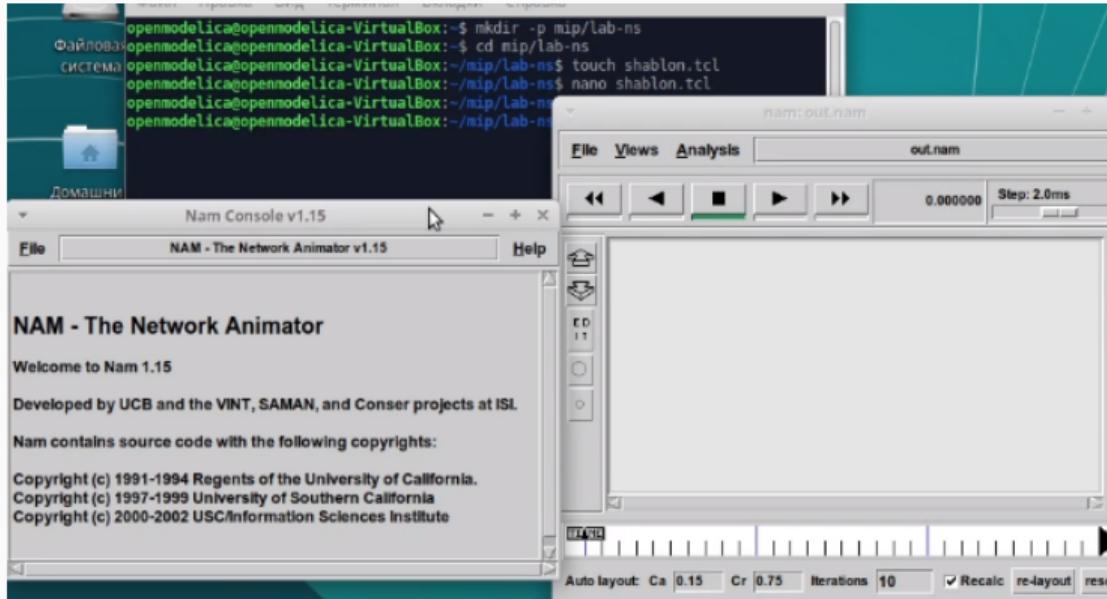


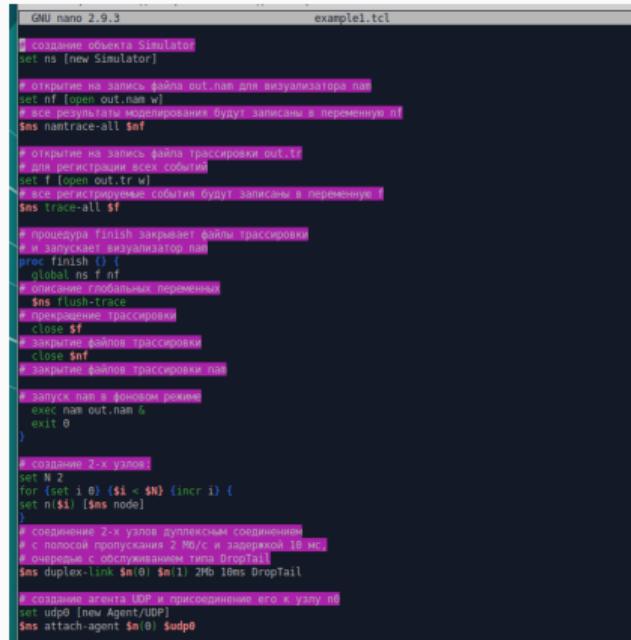
Рис. 4: Симуляция shablon.tcl

Простой пример описания топологии сети, состоящей из двух узлов и одного соединения

```
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ls shablon.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ cp shablon.tcl example1.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ █
```

Рис. 5: Копирование шаблона для example1.tcl

Простой пример описания топологии сети, состоящей из двух узлов и одного соединения



```
GNU nano 2.9.3
example1.tcl

# создание объекта Simulator
set ns [new Simulator]

# открытие на запись файла out.nam для визуализатора лав
set nf [open out.nam w]
# все результаты моделирования будут записаны в переменную $f
$ns namtrace-all $nf

# открытие языка записи файла трассировки out.tcl
# для регистрации всех событий
set f [open out.tr w]
# все зарегистрированные события будут записаны в переменную $f
$ns trace-all $f

# процедура finish закрывает файлы трассировки
# и возвращает визуализатор нам
proc finish {f nf} {
    global ns t
    # сохранение глобальных переменных
    $ns flush-trace
    # закрытие файла трассировки
    close $f
    # закрытие файлов трассировки
    close $nf
    # закрытие файлов трассировки нам
    exec nam out.nam &
    exit 0
}

# создание 2-х узлов
set N 2
for {set i 0} {$i < $N} {incr i} {
    set n[$i] [$ns node]
}

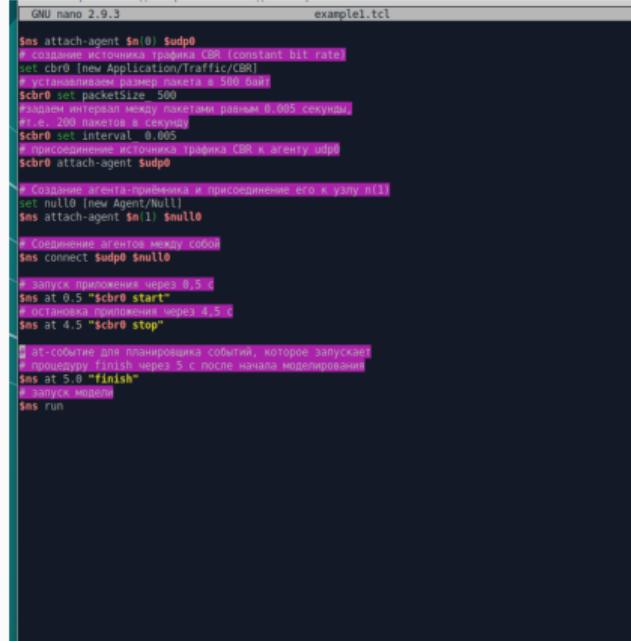
# соединение 2-х узлов дуплексным соединением
# с полосой пропускания 2 Мб/с и задержкой 10 мс
# очередь с обслуживанием типа DropTail
$ns duplex-link $n(0) $n(1) 2Mb 10ms DropTail

# создание агента UDP и присоединение его к узлу н0
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0

#
```

Рис. 6: Содержание example1.tcl

Простой пример описания топологии сети, состоящей из двух узлов и одного соединения



```
GNU nano 2.9.3                               example1.tcl

$ns attach-agent $n(0) $udp0
# создание источника трафика CBR (constant bit rate)
set cbr0 [new Application/Traffic/CBR]
# установлением 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-протокола и присоединение его к узлу n(1)
set null0 [new Agent/Null]
$ns attach-agent $n(1) $null0

# Создание соединения между $udp0 и $null0
$ns connect $udp0 $null0

# запуск приемника через 0.5 с
$ns at 0.5 "$cbr0 start"
# остановка приемника через 4.5 с
$ns at 4.5 "$cbr0 stop"

# стартование для таймерчика события, которое запускает
# процедуру finish через 5 с после начала моделирования
$ns at 5.0 "finish"
# запуск модели
$ns run
```

Рис. 7: Содержание example1.tcl(продолжение)

Простой пример описания топологии сети, состоящей из двух узлов и одного соединения

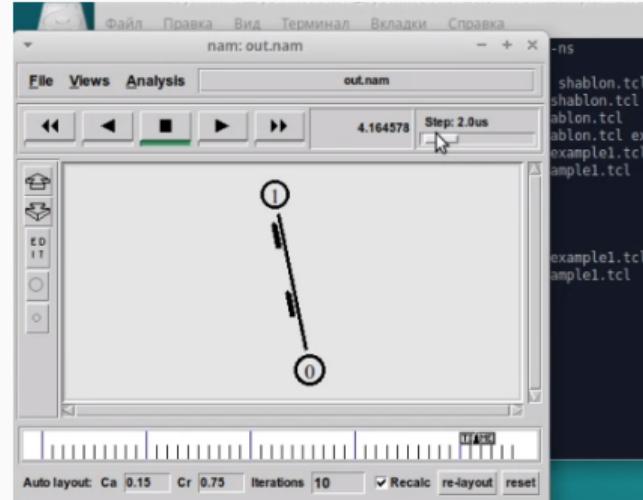


Рис. 8: Симуляция example1.tcl

Пример с усложнённой топологией сети

```
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ cp shablon.tcl example2.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$
```

Рис. 9: Копирование шаблона для example2.tcl

Пример с усложнённой топологией сети

```
GNU nano 2.9.3                                         example2.tcl

set N 4
for {set i 0} {$i < $N} {incr i} {
    set n[$i] [$ns node]
}
$ns duplex-link $n(0) $n(2) 2Mbps 10ms DropTail
$ns duplex-link $n(1) $n(2) 2Mbps 10ms DropTail
$ns duplex-link $n(3) $n(2) 2Mbps 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

# создание агента UDP и присоединение его к узлу n(0)
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
# создание источника трафика
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize 500
$cbr0 set interval 0.005
$cbr0 attach-agent $udp0
# передача трафика присоединение его к узлу n(1)
set tcp1 [new Agent/TCP]
$ns attach-agent $n(1) $tcp1
# создание приложения FTP
# и присоединение его к агенту tcp1
set ftp [new Application/FTP]
$ftp attach-agent $tcp1

# создание агента-получателя для udp0
set null0 [new Agent/Null]
$ns attach-agent $n(3) $null0
# создание агента-получателя для tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $n(3) $sink1

$ns connect $udp0 $null0
$ns connect $tcp1 $sink1

$ns color 1 Blue
$ns color 2 Red

$udp0 set class_1
$tcp1 set class_2

$ns duplex-link-op $n(2) $n(3) queuePos 0.5
$ns queue-limit $n(2) $n(3) 20

$ns at 0.5 "$cbr0 start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr0 stop"
```

Рис. 10: Содержание example2.tcl

Пример с усложнённой топологией сети

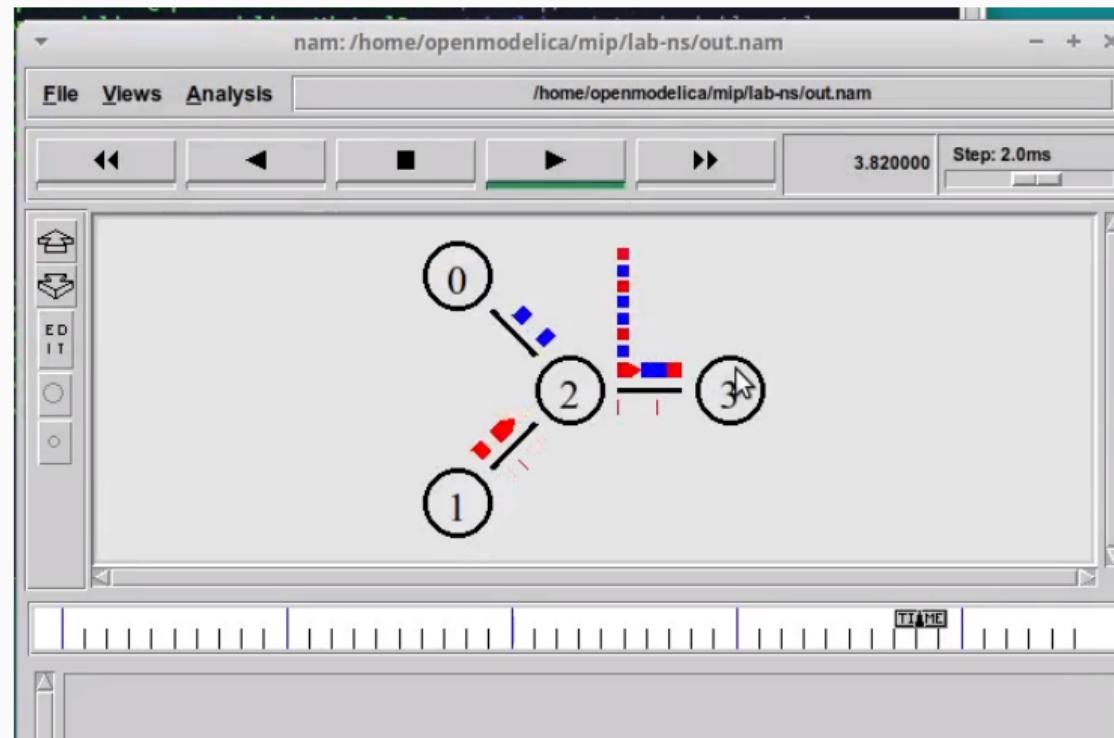


Рис. 11: Симуляция example2.tcl

Пример с кольцевой топологией сети

```
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ cp shablon.tcl example3.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$
```

Рис. 12: Копирование шаблона для example3.tcl

Пример с кольцевой топологией сети

```
GNU nano 2.9.3                                         example3.tcl

# запуск нам в фоновом режиме
exec nam out.nam &
exit 0
}

set N 7
for {set i 0} {($i < $N)} {incr i} {
set n[$i] [$ns node]

(or {set i 0} {($i < $N)} {incr i} {
$ns duplex-link $n[$i] $n [expr {($i+1)%$N}] 1Mb 10ms DropTail
}

set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0

set cbr0 [new Agent/CBR]
$ns attach-agent $n(0) $cbr0
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005

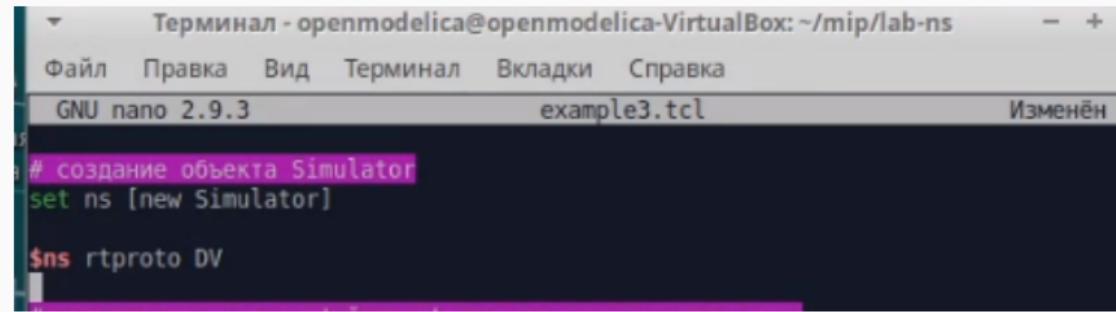
set null0 [new Agent/Null]
$ns attach-agent $n(3) $null0
$ns connect $cbr0 $null0

$ns at 0.5 "schr0 start"
$ns rtmodel-at 1.0 down $n(1) $n(2)
$ns rtmodel-at 2.0 up $n(1) $n(2)
$ns at 4.5 "schr0 stop"

# ат-событие для планировчика событий, которое запускает
# процедуру finish через 5 с после начала моделирования
$ns at 5.0 "finish"
#запуск модели
$ns run
```

Рис. 13: Содержание example3.tcl

Пример с кольцевой топологией сети



Терминал - openmodelica@openmodelica-VirtualBox: ~/mip/lab-ns

Файл Правка Вид Терминал Вкладки Справка

GNU nano 2.9.3 example3.tcl Изменён

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

$ns rtproto DV
```

Рис. 14: Добавление строчки в начало example3.tcl

Пример с кольцевой топологией сети

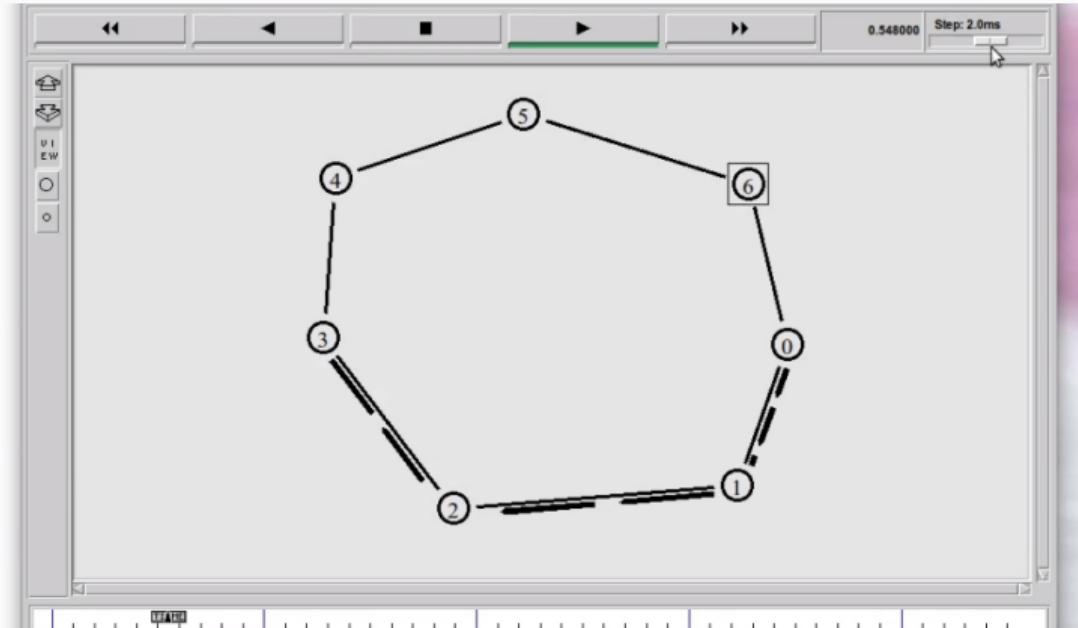


Рис. 15: Симуляция example3.tcl

Пример с кольцевой топологией сети

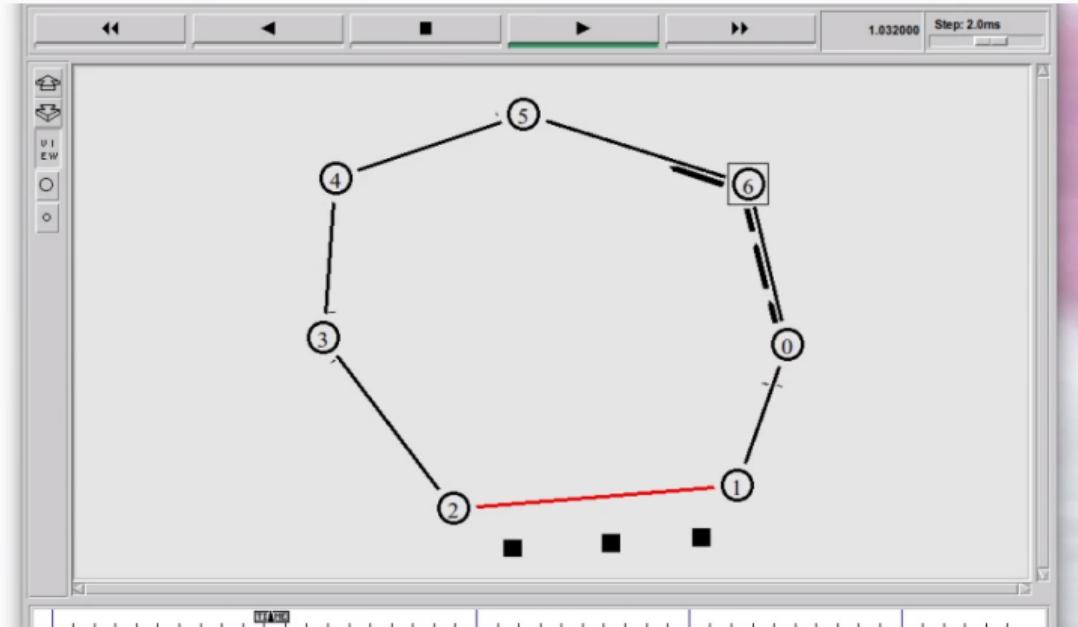


Рис. 16: Передача данных по сети с кольцевой топологией в случае разрыва соединения

Пример с кольцевой топологией сети

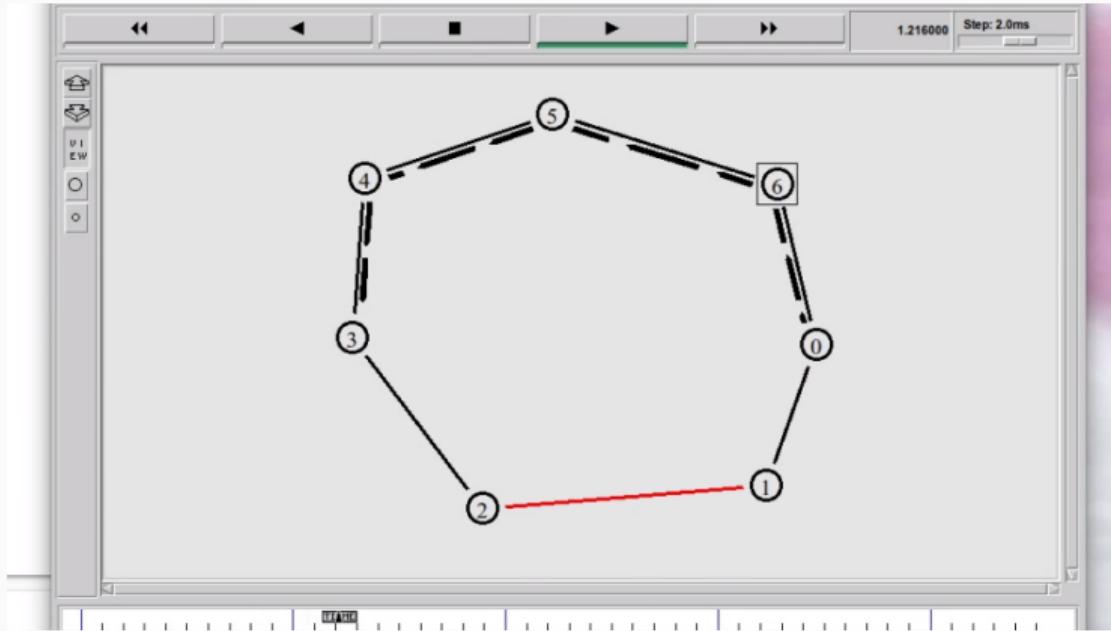


Рис. 17: Маршрутизация данных по сети с кольцевой топологией в случае разрыва соединения

Упражнение

```
set N 5
for {set i 0} {$i < $N} {incr i} {
    set n($i) [$ns node]
}

for {set i 0} {$i < $N} {incr i} {
    $ns duplex-link $n($i) $n([expr ($i+1)%$N]) 1Mb 10ms DropTail
}

set n5 [$ns node]
$ns duplex-link $n5 $n(1) 1Mb 10ms DropTail

set tcp1 [new Agent/TCP/Newreno]
$ns attach-agent $n(0) $tcp1

set ftp [new Application/FTP]
$ftp attach-agent $tcp1

set sink1 [new Agent/TCPSink/DelAck]
$ns attach-agent $n5 $sink1
$ns connect $tcp1 $sink1

$ns at 0.5 "$ftp start"
$ns rtmodel-at 1.0 down $n(0) $n(1)
$ns rtmodel-at 2.0 up $n(0) $n(1)
$ns at 4.5 "$ftp stop"
$ns at 5.0 "finish"
```

Рис. 18: Содержание example4.tcl

Упражнение

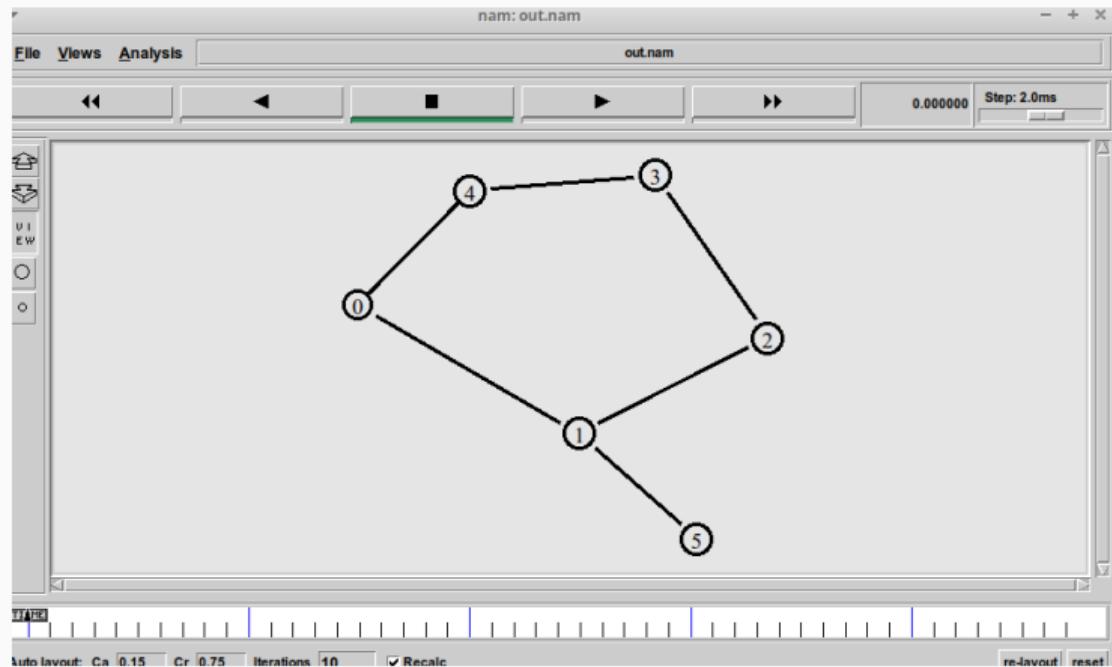


Рис. 19: Симуляция example4.tcl

Упражнение

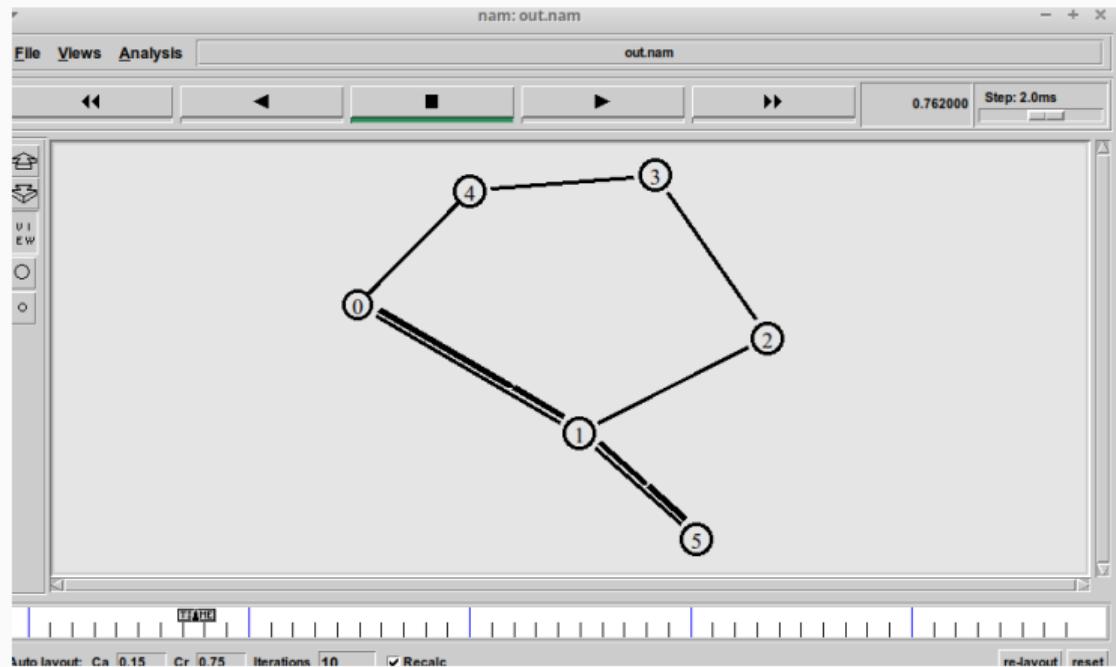


Рис. 20: Передача данных осуществляется по кратчайшему пути

Упражнение

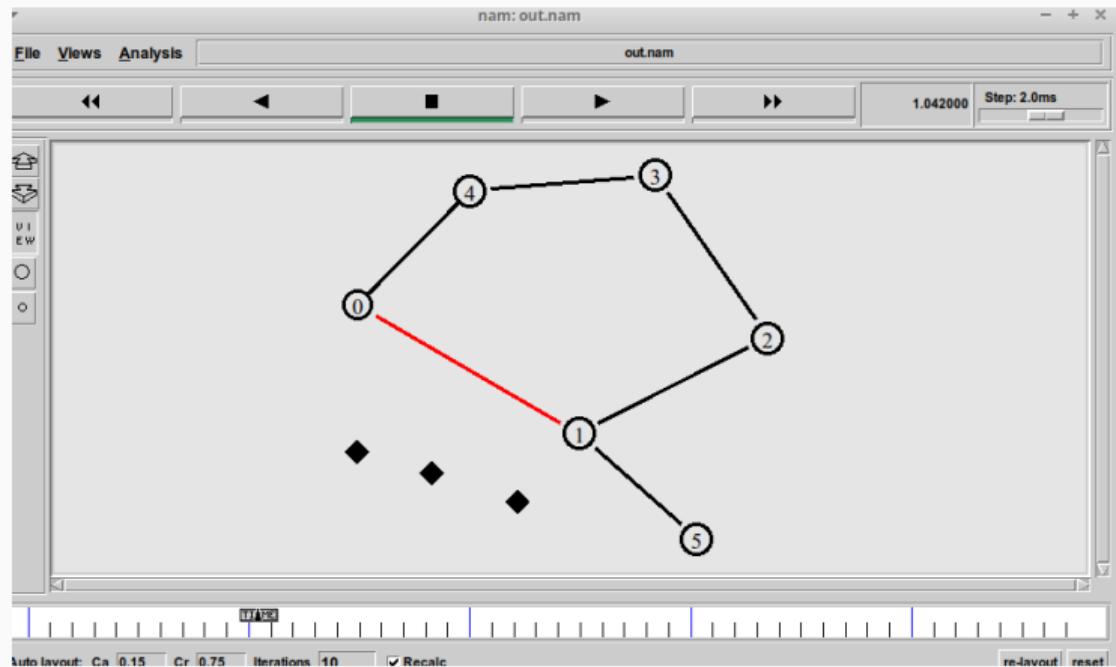


Рис. 21: Разрыв соединения между узлами

Упражнение

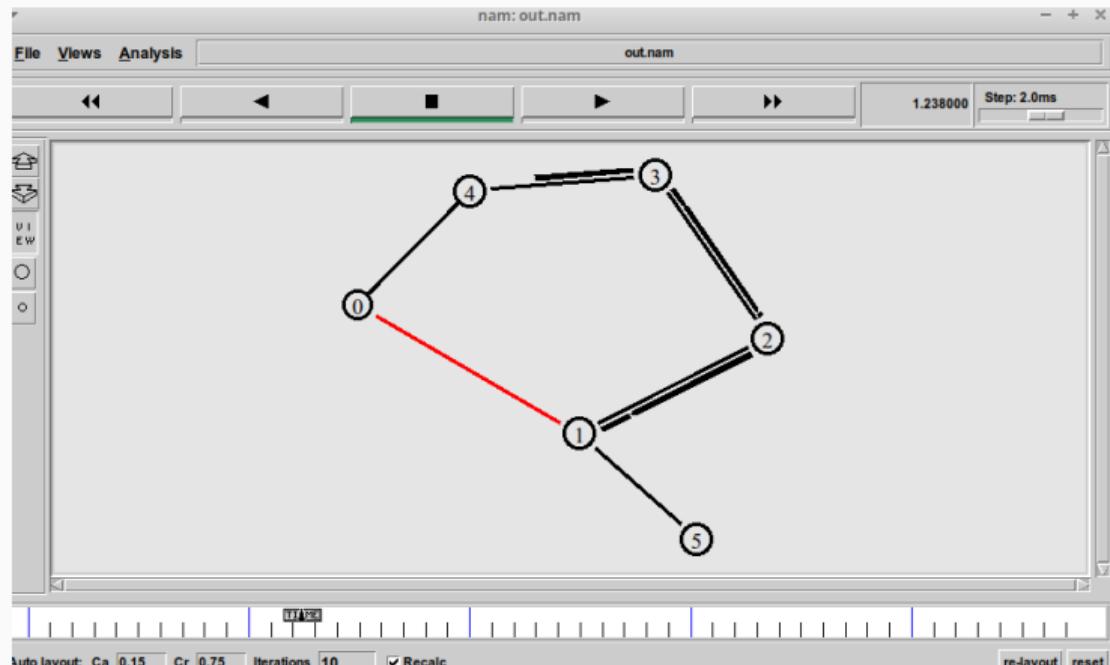


Рис. 22: Резервный маршрут передачи данных

Выводы

Выводы

Во время выполнения данной лабораторной работы я приобрела навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также провела анализ полученных результатов моделирования.