

Лабораторная работа №1

Простые модели компьютерной сети

Кадров Виктор Максимович

11 апреля 2025

Российский университет дружбы народов имени Патриса Лумумбы, Москва, Россия

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

1. Создать шаблон сценария для NS-2.
2. Рассмотреть простой пример описания топологии сети, состоящей из двух узлов и одного соединения.
3. Рассмотреть пример с усложнённой топологией сети.
4. Рассмотреть пример с кольцевой топологией сети
5. Выполнить упражнение

В своём рабочем каталоге создадим директорию `mip`, в которой будут выполняться лабораторные работы. Внутри `mip` создадим директорию `lab-ns`, а в ней файл `shablon.tcl`.

```
142 mkdir -p mip/lab-ns
143 cd mip/lab-ns/
144 ls
145 touch shablon.tcl
146 nano shablon.tcl
147 ns shablon.tcl
```

Рис. 1: Создание директорий и файла

Получившийся шаблон можно использовать в дальнейшем в большинстве разрабатываемых скриптов NS-2, добавляя в него до строки `$ns at 5.0 "finish"` описание объектов и действий моделируемой системы.

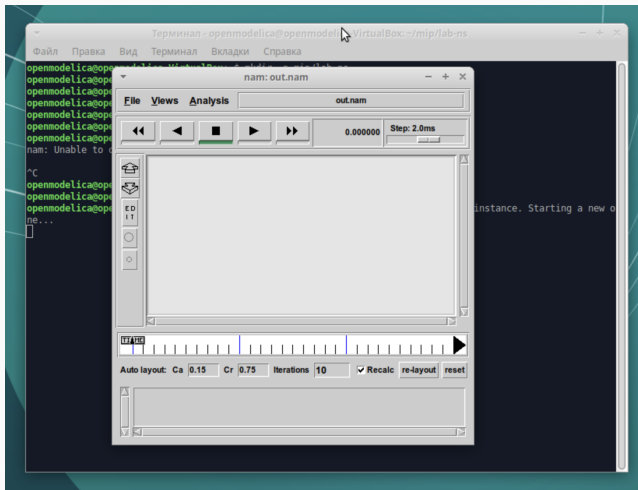
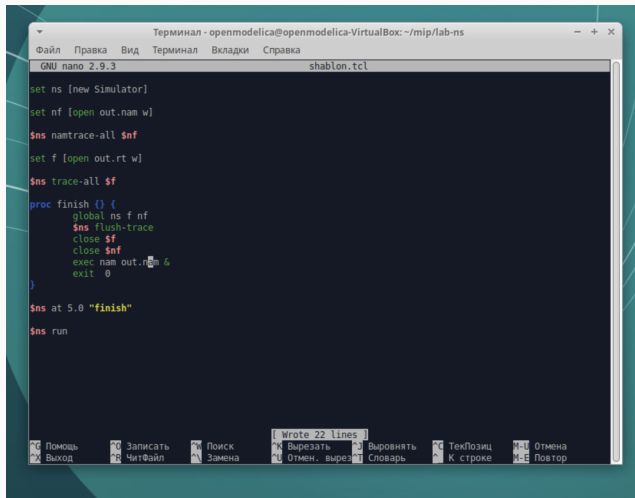


Рис. 2: Скрипт шаблона



The image shows a terminal window titled "Терминал - openmodelica@openmodelica-VirtualBox: ~/mip/lab-ns". The window contains the GNU nano 2.9.3 text editor editing a file named "shablon.tcl". The script defines a simulator, sets up output files, enables tracing, and defines a "finish" procedure. The simulation is currently at 5.0 seconds.

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
set f [open out.rt w]
$ns trace-all $f
proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam out.nam &
    exit 0
}
$ns at 5.0 "finish"
$ns run
```

The bottom status bar of the nano editor shows "I wrote 22 lines" and various keyboard shortcuts for editing and navigation.

Рис. 3: Симуляция шаблона

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

Постановка задачи. Требуется смоделировать сеть передачи данных, состоящую из двух узлов, соединённых дуплексной линией связи с полосой пропускания 2 Мб/с и задержкой 10 мс, очередью с обслуживанием типа DropTail. От одного узла к другому по протоколу UDP осуществляется передача пакетов, размером 500 байт, с постоянной скоростью 200 пакетов в секунду.

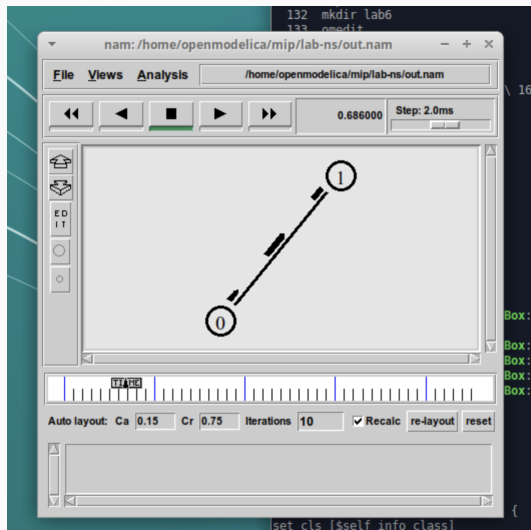
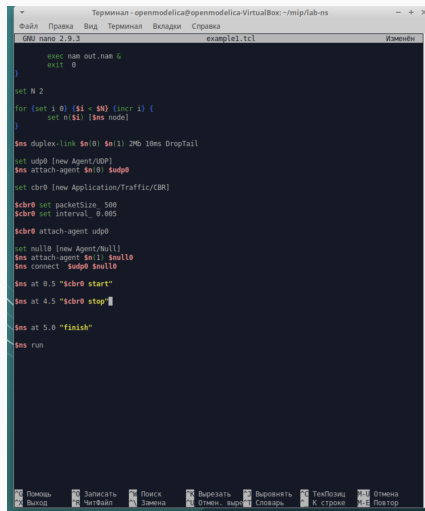


Рис. 4: Визуализация простой модели сети с помощью nam

Скопируем содержимое созданного шаблона в новый файл `example1.tcl` и откроем на редактирование. Создадим агенты для генерации и приёма трафика. Создадим агент UDP и присоединим к узлу `n0`. К агенту присоединяем приложение. В данном случае — это источник с постоянной скоростью (Constant Bit Rate, CBR), который каждые 5 мс посылает пакет $R = 500$ байт. Далее создадим Null-агент, который работает как приёмник трафика, и прикрепим его к узлу `n1`. Соединим агенты между собой. Для запуска и остановки приложения CBR добавляются `at`-события в планировщик событий.

Скрипт сети из двух узлов и одного соединения



The image shows a terminal window titled "Терминал - openmodelica@openmodelica-VirtualBox: ~/mip/lab-ns". The terminal is running the nano text editor, editing a file named "example1.tcl". The script content is as follows:

```
GNU nano 2.9.3 example1.tcl

exec nam out.nam &
exit 0
}

set N 2

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

$ns duplex-link $n(0) $n(1) 2Mb 10ms DropTail

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

set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize 500
$cbr0 set interval 0.005

$cbr0 attach-agent udp0

set null0 [new Agent/Null]
$ns attach-agent $n(1) $null0
$ns connect $udp0 $null0

$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"

$ns at 5.0 "finish"

$ns run
```

The terminal window includes a menu bar with options: "Файл", "Правка", "Вид", "Терминал", "Вкладки", and "Справка". At the bottom, there is a toolbar with icons and labels for various editing functions: "Помощь", "Записать", "Поиск", "Вырезать", "Выровнять", "ТекПозиц", "Отмена", "Выход", "ЧитФайл", "Замена", "Отмен. выре", "Словарь", "К. строке", and "Повтор".

Рис. 5: Скрипт сети из двух узлов и одного соединения

Пример с усложненной топологией сети

Постановка задачи. Описание моделируемой сети: – сеть состоит из 4 узлов (n_0 , n_1 , n_2 , n_3); – между узлами n_0 и n_2 , n_1 и n_2 установлено дуплексное соединение с пропускной способностью 2 Мбит/с и задержкой 10 мс; – между узлами n_2 и n_3 установлено дуплексное соединение с пропускной способностью 1,7 Мбит/с и задержкой 20 мс; – каждый узел использует очередь с дисциплиной DropTail для накопления пакетов, максимальный размер которой составляет 10; – TCP-источник на узле n_0 подключается к TCP-приёмнику на узле n_3 (по-умолчанию, максимальный размер пакета, который TCP-агент может генерировать, равняется 1KByte) – TCP-приёмник генерирует и отправляет ACK пакеты отправителю и откидывает полученные пакеты; – UDP-агент, который подсоединён к узлу n_1 , подключён к null-агенту на узле n_3 (null-агент просто откидывает пакеты); – генераторы трафика ftp и cbr прикреплены к TCP и UDP агентам соответственно; – генератор cbr генерирует пакеты размером 1 Кбайт со скоростью 1 Мбит/с; – работа cbr начинается в 0,1 секунду и прекращается в 4,5 секунды, а ftp начинает работать в 1,0 секунду и прекращает в 4,0 секунды.

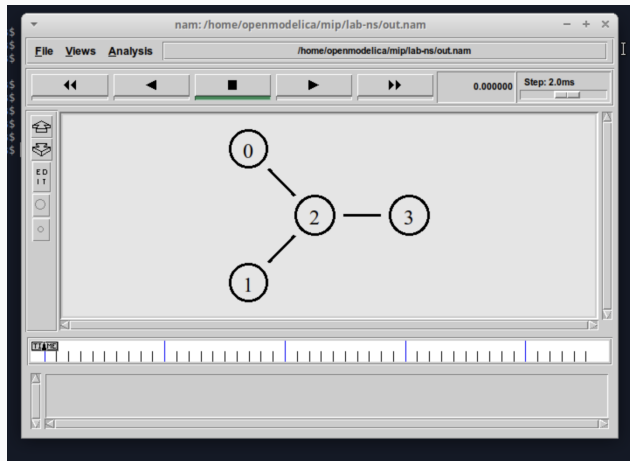


Рис. 6: Модель с усложненной топологией сети без симуляции

Модель с усложненной топологией сети. Симуляция

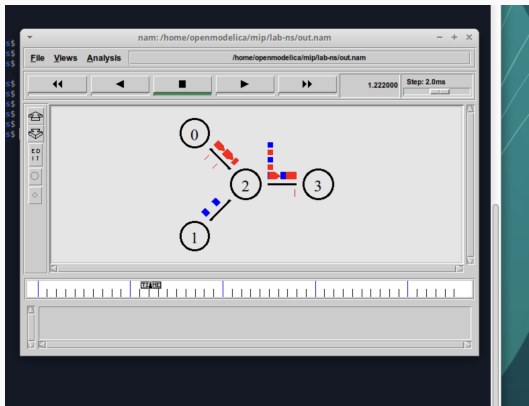
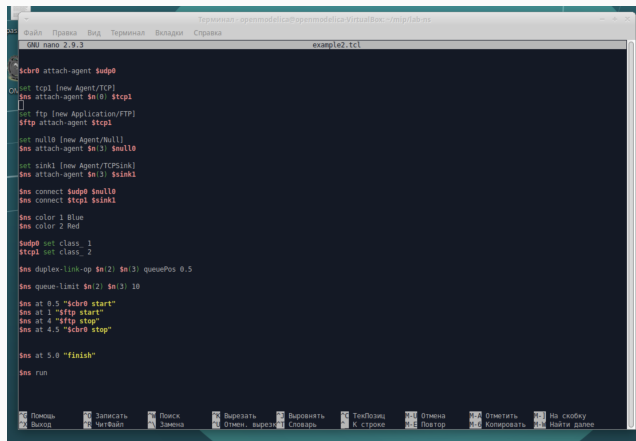


Рис. 7: Модель с усложненной топологией сети. Симуляция

Скопируем содержимое созданного шаблона в новый файл и откроем `example2.tcl` на редактирование. Создадим 4 узла и 3 дуплексных соединения с указанием направления. Создадим агент UDP с прикрепленным к нему источником CBR и агент TCP с прикрепленным к нему приложением FTP. Создадим агенты-получатели. Соединим агенты `udp0` и `tcp1` и их получателей. Зададим описание цвета каждого потока. Отслеживание событий в очереди. Наложим ограничения на размер очереди. Добавим `at`-события.

Скрипт модели с усложненной топологией сети



```
Terминал - openmodelica@openmodelica-VirtualBox: ~/mip/lab-ns
GNU nano 2.9.3 example2.tcl

$scr0 attach-agent $sudp0

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

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

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

set sink1 [new Agent/TCPSink]
$ns attach-agent $n(3) $sink1

$ns connect $sudp0 $null0
$ns connect $tcp1 $sink1

$ns color 1 Blue
$ns color 2 Red

$sudp0 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) 10

$ns at 0.5 "$scr0 start"
$ns at 1 "$ftp start"
$ns at 4 "$ftp stop"
$ns at 4.5 "$scr0 stop"

$ns at 5.0 "finish"

$ns run
```

Рис. 8: Скрипт модели с усложненной топологией сети

Постановка задачи. Требуется построить модель передачи данных по сети с кольцевой топологией и динамической маршрутизацией пакетов: – сеть состоит из 7 узлов, соединённых в кольцо; – данные передаются от узла $n(0)$ к узлу $n(3)$ по кратчайшему пути; – с 1 по 2 секунду модельного времени происходит разрыв соединения между узлами $n(1)$ и $n(2)$; – при разрыве соединения маршрут передачи данных должен измениться на резервный.

Скопируем содержимое созданного шаблона в новый файл и откроем `example3.tcl` на редактирование. Опишем топологию моделируемой сети. Далее соединим узлы так, чтобы создать круговую топологию. Каждый узел, за исключением последнего, соединяется со следующим, последний соединяется с первым. Для этого в цикле использован оператор `%`, означающий остаток от деления нацело. Зададим передачу данных от узла $n(0)$ к узлу $n(3)$. Данные передаются по кратчайшему маршруту от узла $n(0)$ к узлу $n(3)$, через узлы $n(1)$ и $n(2)$

Передача данных по кратчайшему пути сети с кольцевой топологией

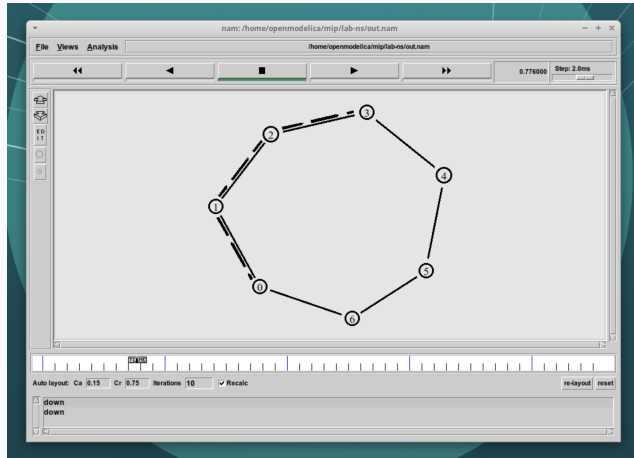


Рис. 9: Передача данных по кратчайшему пути сети с кольцевой топологией

Передача данных по сети с кольцевой топологией в случае разрыва соединения

Добавим команду разрыва соединения между узлами $n(1)$ и $n(2)$ на время в одну секунду, а также время начала и окончания передачи данных. Передача данных при кольцевой топологии сети в случае разрыва соединения.

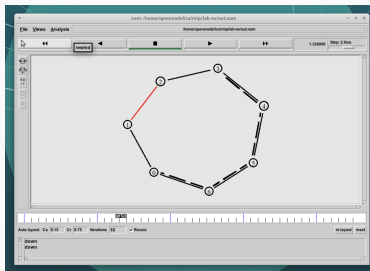
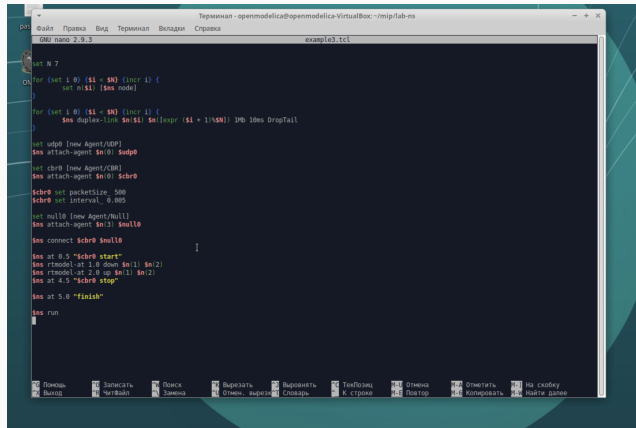


Рис. 10: Передача данных по сети с кольцевой топологией в случае разрыва соединения

Добавив в начало скрипта после команды создания объекта Simulator, увидим, что сразу после запуска в сети отправляется небольшое количество маленьких пакетов, используемых для обмена информацией, необходимой для маршрутизации между узлами. Когда соединение будет разорвано, информация о топологии будет обновлена, и пакеты будут отсылаться по новому маршруту через узлы $n(6)$, $n(5)$ и $n(4)$.

Скрипт модели с кольцевой топологией



```
Терминал - openmodelica@openmodelica-VirtualBox: ~/nip/lab-ns
GNU nano 2.9.3 example3.tcl

set N 7
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 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

$ns at 0.5 "$cbr0 start"
$ns rmodel-at 1.0 down $n(1) $n(2)
$ns rmodel-at 2.0 up $n(1) $n(2)
$ns at 4.5 "$cbr0 stop"

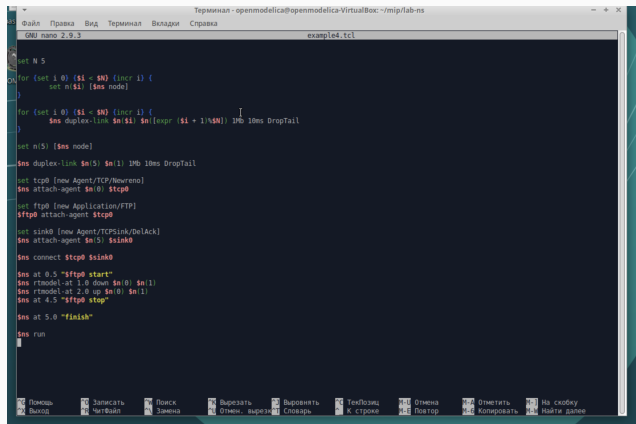
$ns at 5.0 "finish"

$ns run
```

Рис. 11: Скрипт модели с кольцевой топологией

Упражнение. Внесите следующие изменения в реализацию примера с кольцевой топологией сети: – повторить топологию сети, предоставленную в файле с заданиями к лабораторной работе; – передача данных должна осуществляться от узла $n(0)$ до узла $n(5)$ по кратчайшему пути в течение 5 секунд модельного времени; – передача данных должна идти по протоколу TCP (тип Newreno), на принимающей стороне используется TCPSink-объект типа DelAck; поверх TCP работает протокол FTP с 0,5 до 4,5 секунд модельного времени; – с 1 по 2 секунду модельного времени происходит разрыв соединения между узлами $n(0)$ и $n(1)$; – при разрыве соединения маршрут передачи данных должен измениться на резервный, после восстановления соединения пакеты снова должны пойти по кратчайшему пути.

Скрипт модели из упражнения



```
Терминал - openmodelica@openmodelica-VirtualBox: ~/mir/lab-ns
GNU nano 2.9.3 example4.tcl

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}] 1Mb 10ms DropTail
}

set n(5) [$ns node]
$ns duplex-link $n(5) $n(1) 1Mb 10ms DropTail

set tcp0 [new Agent/TCP/Newreno]
$ns attach-agent $n(0) $tcp0

set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0

set sink0 [new Agent/TCPSink/DelAck]
$ns attach-agent $n(5) $sink0

$ns connect $tcp0 $sink0

$ns at 0.5 "$ftp0 start"
$ns rtmodel-at 1.0 down $n(0) $n(1)
$ns rtmodel-at 2.0 up $n(0) $n(1)
$ns at 4.5 "$ftp0 stop"

$ns at 5.0 "finish"

$ns run
```

Рис. 12: Скрипт модели из упражнения

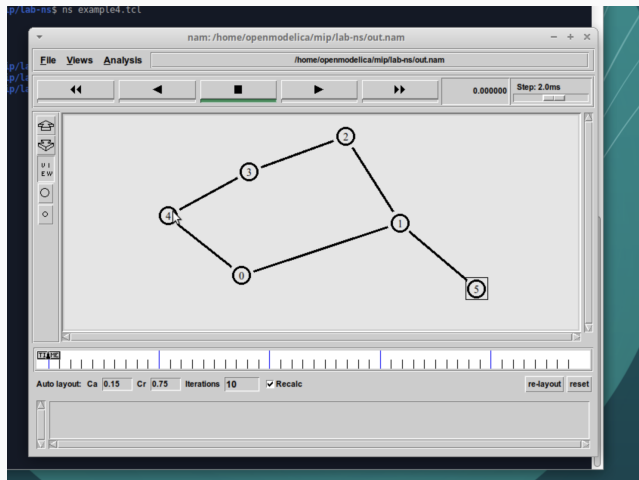


Рис. 13: Топология сети из упражнения

Движение по кратчайшему пути.

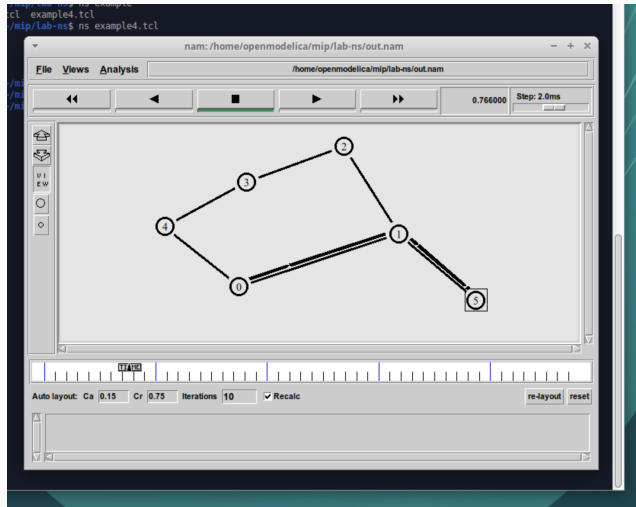


Рис. 14: Движение по кратчайшему пути

Разрыв связи между узлами.

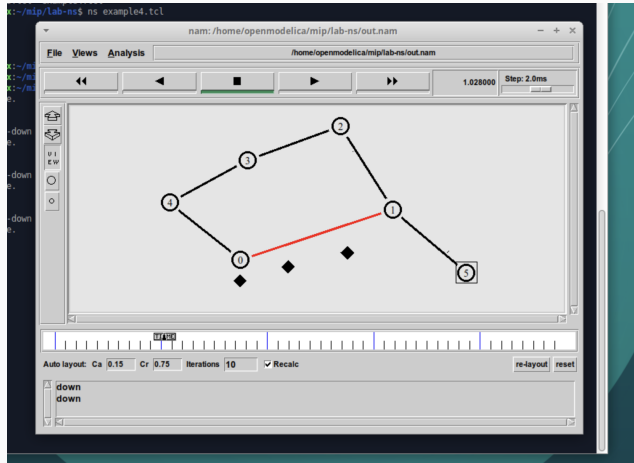


Рис. 15: Разрыв связи между узлами

Движение по длинному пути.

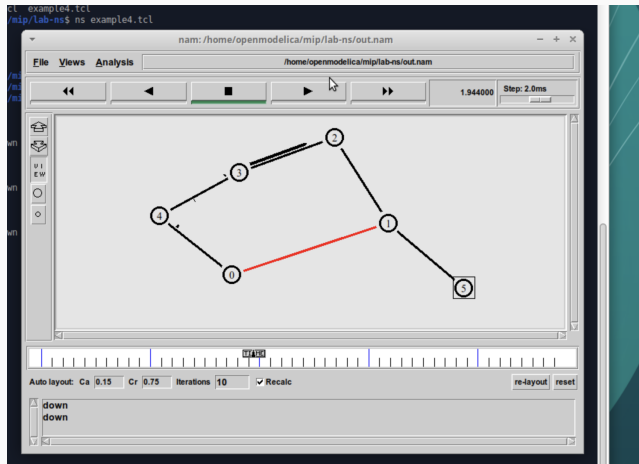


Рис. 16: Движение по длинному пути

Выводы

Мы приобрели навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также провели анализ полученных результатов моделирования.