

# Лабораторная работа №1. Простые модели компьютерной сети

Дисциплина: Имитационное моделирование

---

Ганина Т. С.

11 февраля 2025

Группа НФИбд-01-22

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Ганина Таисия Сергеевна
- Студентка 3го курса, группа НФИбд-01-22
- Фундаментальная информатика и информационные технологии
- Российский университет дружбы народов
- Ссылка на репозиторий гитхаба `tsganina`

## Вводная часть

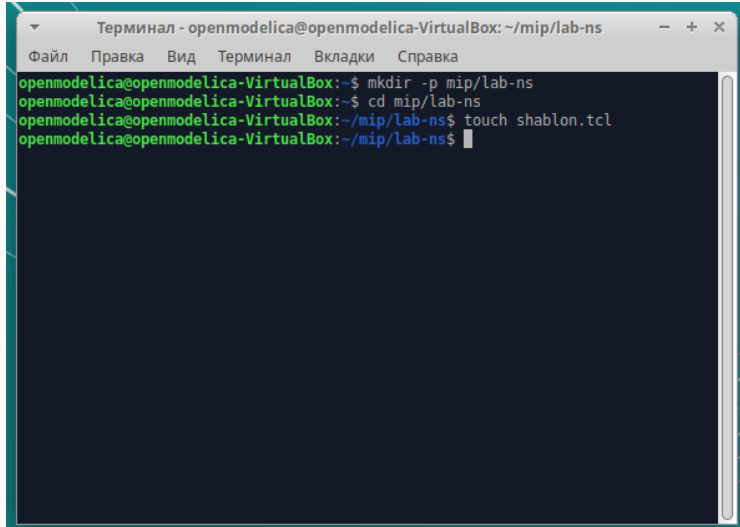
---

Целью данной работы является приобретение навыков моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также анализ полученных результатов моделирования.

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

Записать шаблон сценария для NS-2

---

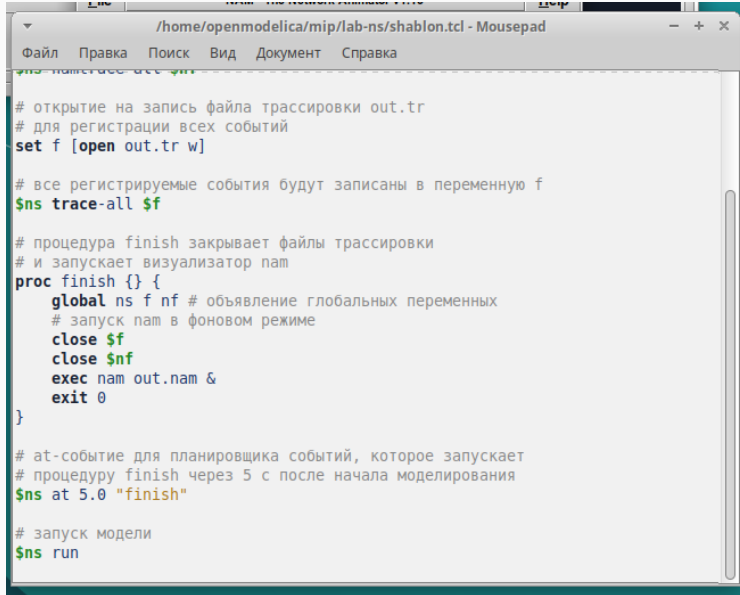


The image shows a terminal window titled "Терминал - openmodelica@openmodelica-VirtualBox: ~/mip/lab-ns". The window has a menu bar with "Файл", "Правка", "Вид", "Терминал", "Вкладки", and "Справка". The terminal content shows the following commands and their outputs:

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

Рис. 1: Создание файлов





The image shows a screenshot of a text editor window titled "/home/openmodelica/mip/lab-ns/shablon.tcl - Mousepad". The window has a menu bar with "Файл", "Правка", "Поиск", "Вид", "Документ", and "Справка". The main text area contains a TCL script for running an OpenModelica simulation. The script includes comments in Russian and TCL commands for file handling, variable registration, and simulation execution.

```
# 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 # объявление глобальных переменных
    # запуск nam в фоновом режиме
    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
```

Рис. 2: Заполнение файла шаблона

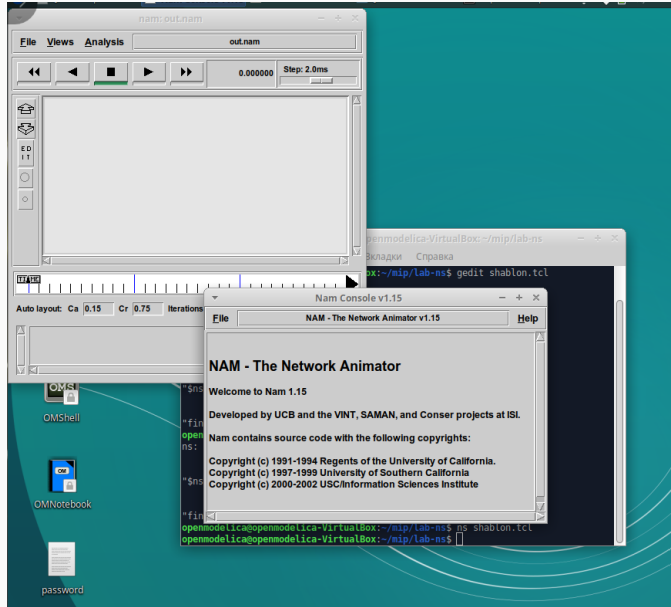


Рис. 3: Результат выполнения файла шаблона, приложение

Создать простой пример описания топологии сети, состоящей из двух узлов и одного соединения.

---

```
/home/openmodelica/mip/lab-ns/example1.tcl - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка

28 # создание 2-х узлов:
29 set N 2
30 for {set i 0} {$i < $N} {incr i} {
31   set n($i) [new node]
32 }
33 # соединение 2-х узлов дуплексным соединением
34 # с полосой пропускания 2 Мб/с и задержкой 10 мс,
35 # очередь с обслуживанием типа DropTail
36 $ns duplex-link $n(0) $n(1) 2Mb 10ms DropTail
37
38
39 # создание агента UDP и присоединение его к узлу n0
40 set udp0 [new Agent/UDP]
41 $ns attach-agent $n(0) $udp0
42 # создание источника трафика CBR (constant bit rate)
43 set cbr0 [new Application/Traffic/CBR]
44 # устанавливаем размер пакета в 500 байт
45 $cbr0 set packetSize_ 500
46 #задаем интервал между пакетами равным 0.005 секунды,
47 #т.е. 200 пакетов в секунду
48 $cbr0 set interval_ 0.005
49 # присоединение источника трафика CBR к агенту udp0
50 $cbr0 attach-agent $udp0
51
52 # Создание агента-приёмника и присоединение его к узлу n(1)
53 set null0 [new Agent/Null]
54 $ns attach-agent $n(1) $null0
55
56 # Соединение агентов между собой
57 $ns connect $udp0 $null0
58
59 # запуск приложения через 0,5 с
60 $ns at 0.5 "$cbr0 start"
61 # остановка приложения через 4,5 с
62 $ns at 4.5 "$cbr0 stop"
63 |
64
65 # at-событие для планировщика событий, которое запускает
66 # процедуру finish через 5 с после начала моделирования
67 $ns at 5.0 "finish"
```

Рис. 4: Код программы

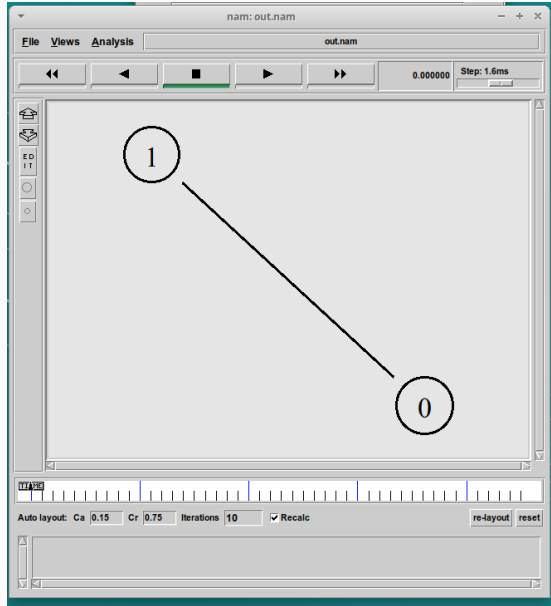


Рис. 5: Аниматор nam

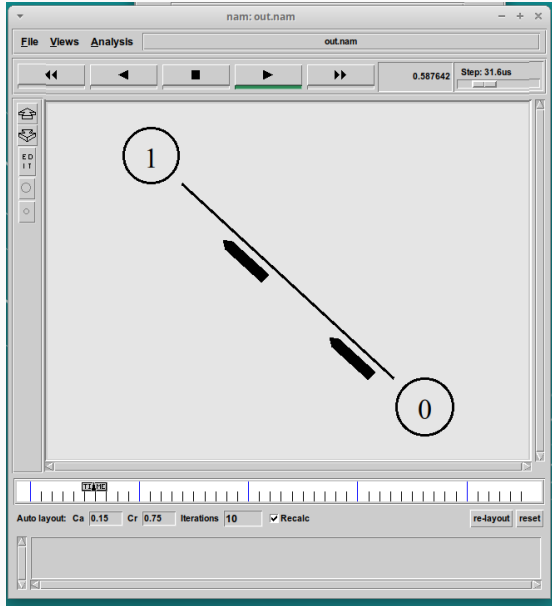
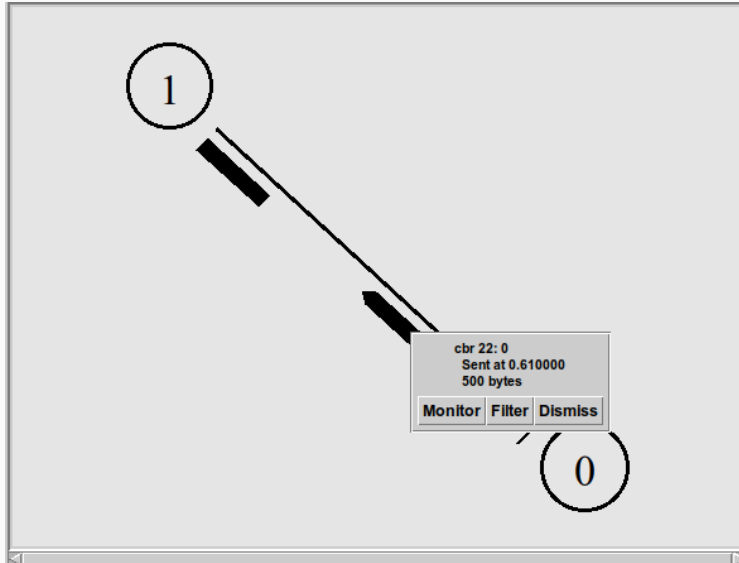


Рис. 6: Визуализация простой модели сети с помощью nam



**Рис. 7:** Можно осуществлять наблюдение за отдельным пакетом, щёлкнув по нему в окне net

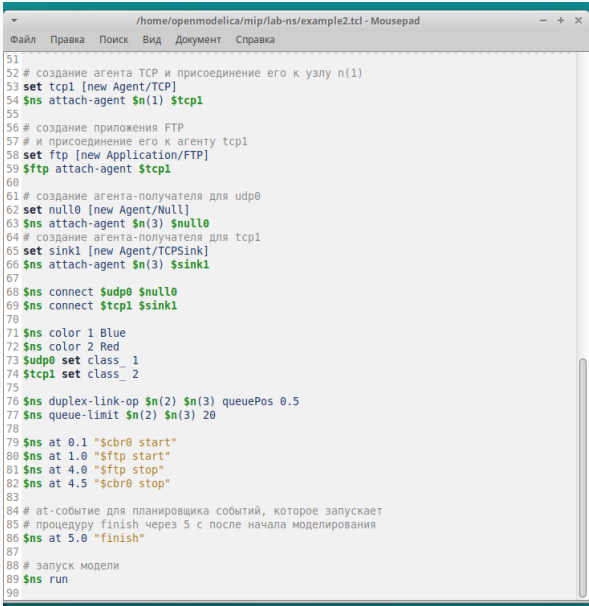
Создать пример с усложнённой  
топологией сети.

---



```
/home/openmodelica/mip/lab-ns/example2.tcl - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка
27
28 set N 4
29 for {set i 0} {$i < $N} {incr i} {
30   set n($i) [$ns node]
31 }
32
33 $ns duplex-link $n(0) $n(2) 2Mb 10ms DropTail
34 $ns duplex-link $n(1) $n(2) 2Mb 10ms DropTail
35 $ns duplex-link $n(3) $n(2) 1.7Mb 20ms DropTail
36
37 $ns duplex-link-op $n(0) $n(2) orient right-down
38 $ns duplex-link-op $n(1) $n(2) orient right-up
39 $ns duplex-link-op $n(2) $n(3) orient right
40
41 # создание агента UDP и присоединение его к узлу n(0)
42 set udp0 [new Agent/UDP]
43 $ns attach-agent $n(0) $udp0
44
45 # создание источника CBR-трафика
46 # и присоединение его к агенту udp0
47 set cbr0 [new Application/Traffic/CBR]
48 $cbr0 set packetSize_ 500
49 $cbr0 set interval_ 0.005
50 $cbr0 attach-agent $udp0
51
52 # создание агента TCP и присоединение его к узлу n(1)
53 set tcp1 [new Agent/TCP]
54 $ns attach-agent $n(1) $tcp1
55
56 # создание приложения FTP
57 # и присоединение его к агенту tcp1
58 set ftp [new Application/FTP]
59 $ftp attach-agent $tcp1
60
61 # создание агента-получателя для udp0
62 set null0 [new Agent/Null]
63 $ns attach-agent $n(3) $null0
64 # создание агента-получателя для tcp1
65 set sink1 [new Agent/TCPSink]
66 $ns attach-agent $n(3) $sink1
```

Рис. 8: Код программы



```
51
52 # создание агента TCP и присоединение его к узлу n(1)
53 set tcp1 [new Agent/TCP]
54 $ns attach-agent $n(1) $tcp1
55
56 # создание приложения FTP
57 # и присоединение его к агенту tcp1
58 set ftp [new Application/FTP]
59 $ftp attach-agent $tcp1
60
61 # создание агента-получателя для udp0
62 set null0 [new Agent/Null]
63 $ns attach-agent $n(3) $null0
64 # создание агента-получателя для tcp1
65 set sink1 [new Agent/TCPSink]
66 $ns attach-agent $n(3) $sink1
67
68 $ns connect $udp0 $null0
69 $ns connect $tcp1 $sink1
70
71 $ns color 1 Blue
72 $ns color 2 Red
73 $udp0 set class_ 1
74 $tcp1 set class_ 2
75
76 $ns duplex-link-op $n(2) $n(3) queuePos 0.5
77 $ns queue-limit $n(2) $n(3) 20
78
79 $ns at 0.1 "$cbr0 start"
80 $ns at 1.0 "$ftp start"
81 $ns at 4.0 "$ftp stop"
82 $ns at 4.5 "$cbr0 stop"
83
84 # at-событие для планировщика событий, которое запускает
85 # процедуру finish через 5 с после начала моделирования
86 $ns at 5.0 "finish"
87
88 # запуск модели
89 $ns run
90
```

Рис. 9: Продолжение кода

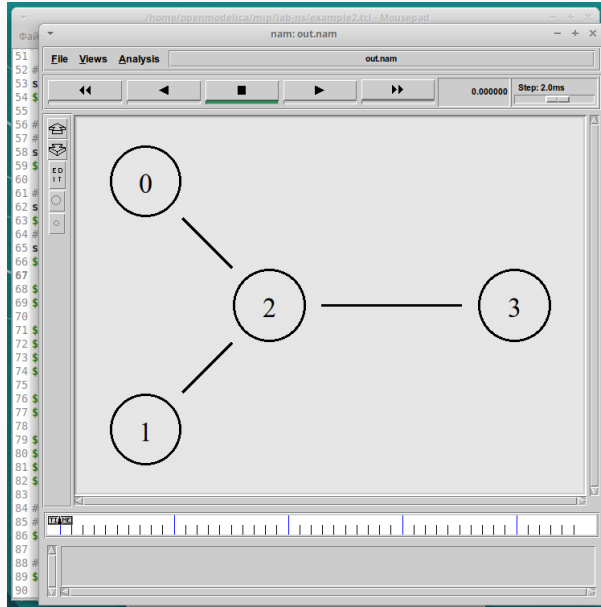


Рис. 10: До запуска анимации, вид на топологию

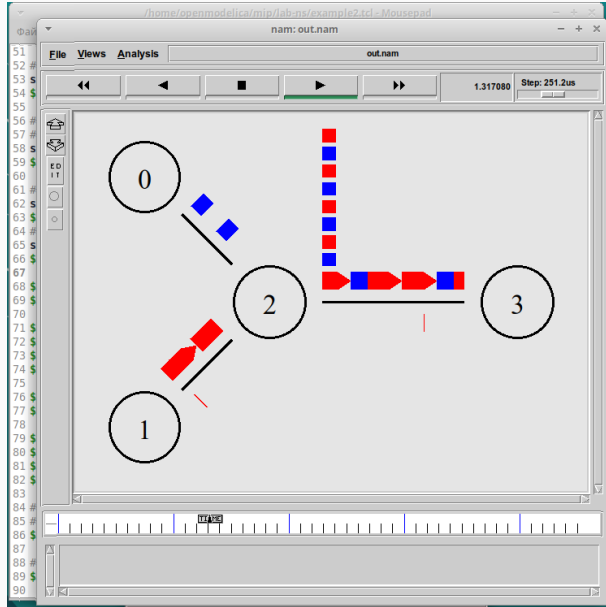


Рис. 11: Выполнение моделирование, анимация

Создать пример с кольцевой  
топологией сети

---

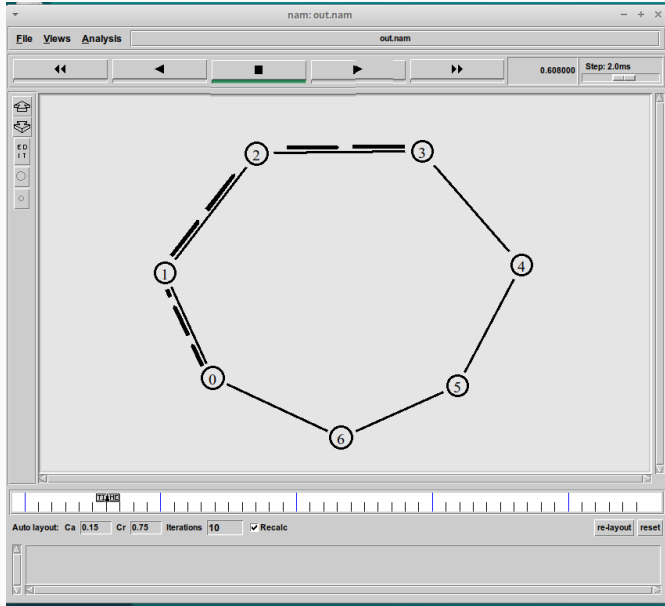


Рис. 12: Кратчайший маршрут до разрыва соединения

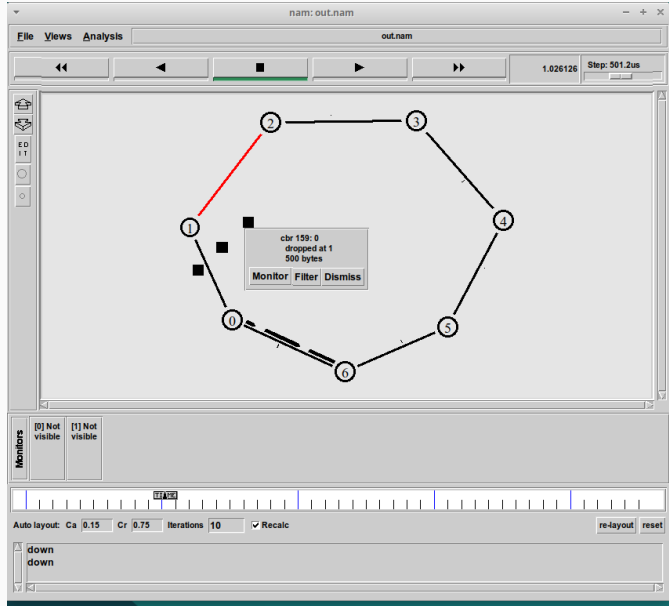


Рис. 13: Момент разрыва соединения

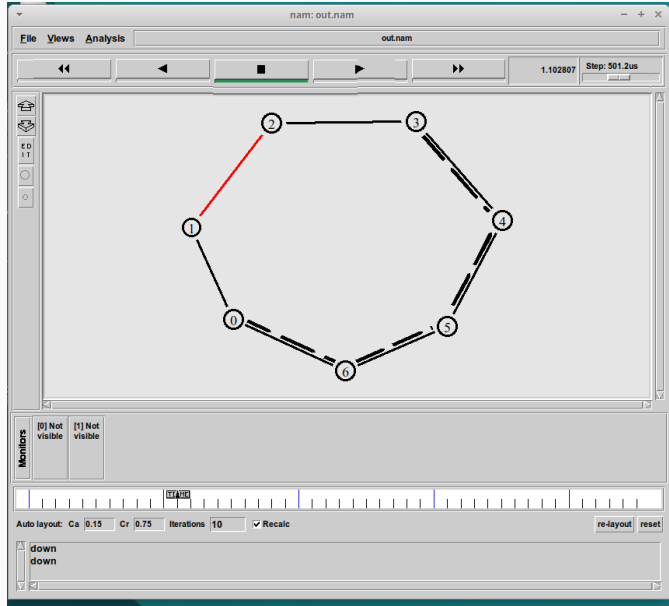
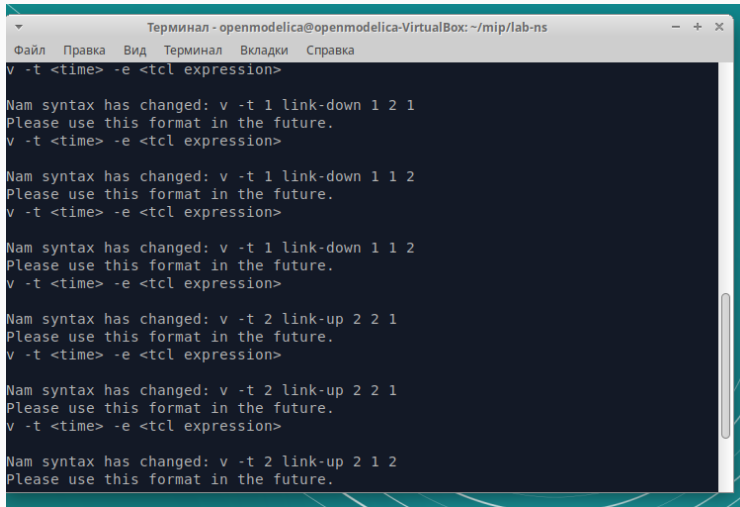


Рис. 14: Резервный маршрут через узлы  $n(6)$ ,  $n(5)$  и  $n(4)$





```
Терминал - openmodelica@openmodelica-VirtualBox: ~/mip/lab-ns
Файл  Правка  Вид  Терминал  Вкладки  Справка
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 1 link-down 1 2 1
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 1 link-down 1 1 2
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 1 link-down 1 1 2
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 2 link-up 2 2 1
Please use this format in the future.
v -t <time> -e <tcl expression>

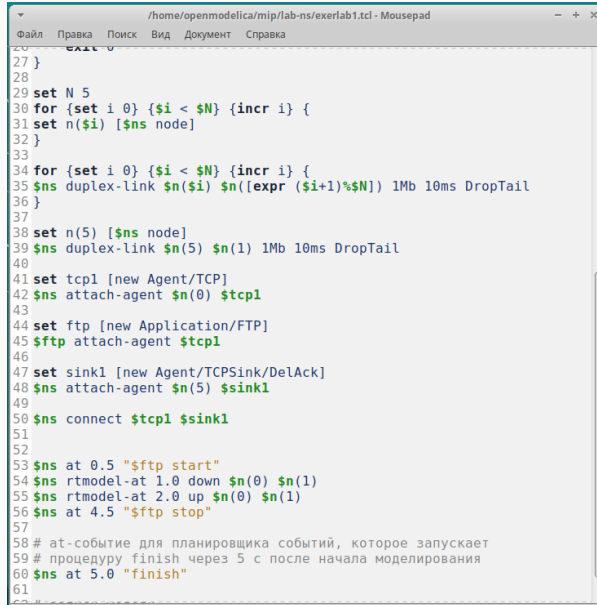
Nam syntax has changed: v -t 2 link-up 2 2 1
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 2 link-up 2 1 2
Please use this format in the future.
```

Рис. 15: Сообщения в терминале о разрыве и восстановлении

## Упражнение

---



```
26 exit 0
27 }
28
29 set N 5
30 for {set i 0} {$i < $N} {incr i} {
31 set n($i) [$ns node]
32 }
33
34 for {set i 0} {$i < $N} {incr i} {
35 $ns duplex-link $n($i) $n([expr ($i+1)%$N]) 1Mb 10ms DropTail
36 }
37
38 set n(5) [$ns node]
39 $ns duplex-link $n(5) $n(1) 1Mb 10ms DropTail
40
41 set tcp1 [new Agent/TCP]
42 $ns attach-agent $n(0) $tcp1
43
44 set ftp [new Application/FTP]
45 $ftp attach-agent $tcp1
46
47 set sink1 [new Agent/TCPSink/DelAck]
48 $ns attach-agent $n(5) $sink1
49
50 $ns connect $tcp1 $sink1
51
52
53 $ns at 0.5 "$ftp start"
54 $ns rtmodel-at 1.0 down $n(0) $n(1)
55 $ns rtmodel-at 2.0 up $n(0) $n(1)
56 $ns at 4.5 "$ftp stop"
57
58 # at-событие для планировщика событий, которое запускает
59 # процедуру finish через 5 с после начала моделирования
60 $ns at 5.0 "finish"
61
62 }
```

Рис. 16: Вид скрипта упражнения

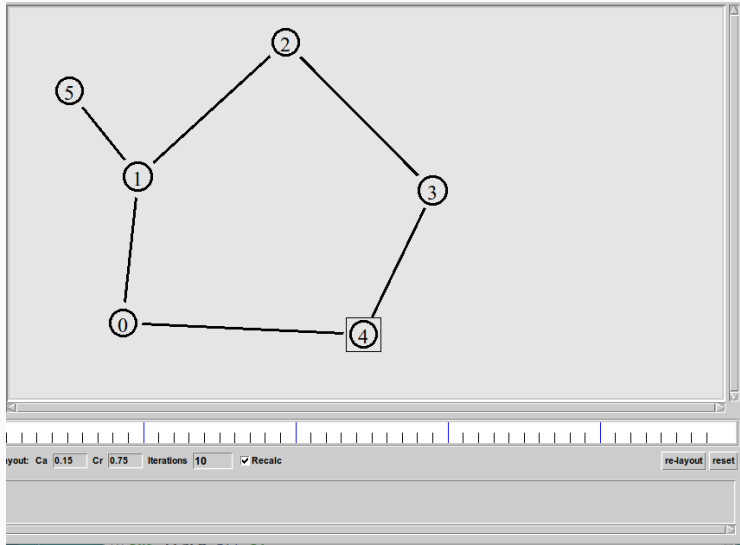


Рис. 17: Топология сети

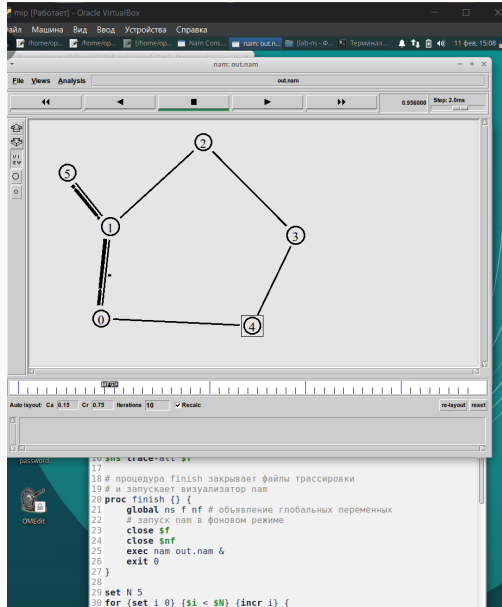


Рис. 18: Передача данных до разрыва соединения между 0 и 1 узлами

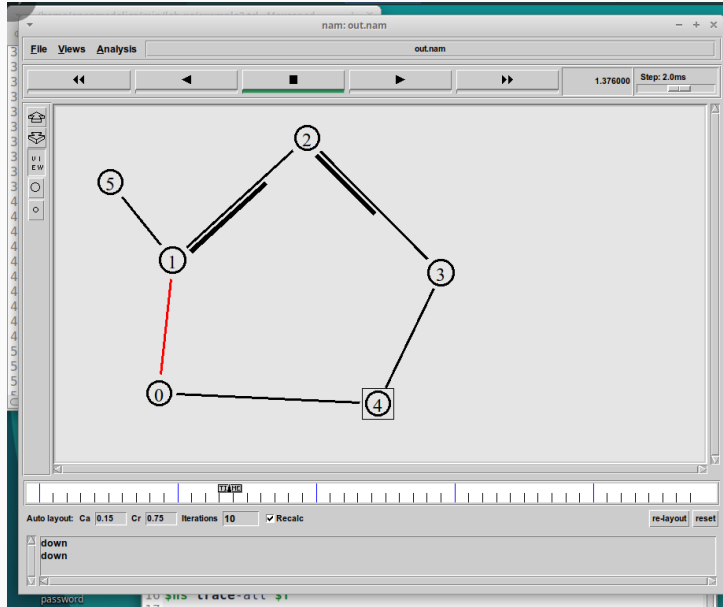


Рис. 19: Передача после разрыва соединения и обновления маршрута

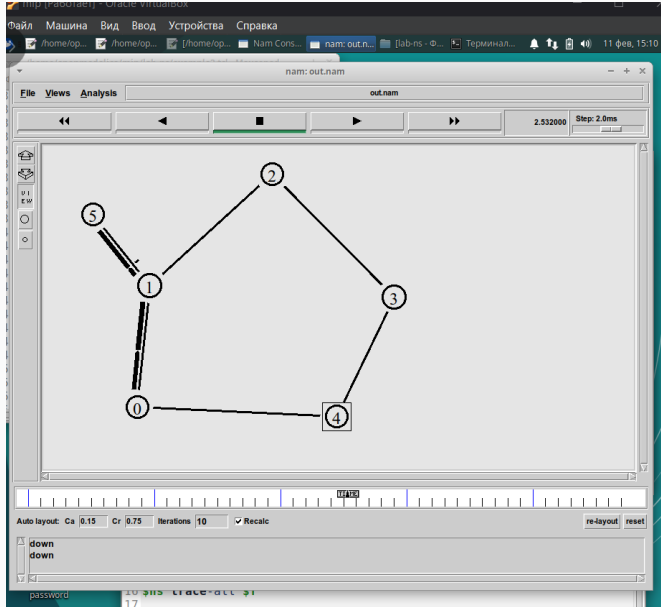


Рис. 20: Передача данных после восстановления соединения между 0 и 1 узлами

## Результаты

---



В ходе данной работы я приобрела практические навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также выполнила анализ полученных результатов моделирования.