

# **Отчёт по лабораторной работе №1**

**Имитационное моделирование**

**Екатерина Канева, НФИбд-02-22**

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.0.1	Шаблон . . . . .	7
3.0.2	Два узла . . . . .	7
3.0.3	Три узла, усложнённая топология . . . . .	11
3.0.4	Кольцевая топология . . . . .	17
3.0.5	Доработка схемы . . . . .	23
<b>4</b>	<b>Выводы</b>	<b>32</b>

# Список иллюстраций

3.1	Шаблон. . . . .	7
3.2	Пример 1. . . . .	8
3.3	Схема 1. . . . .	9
3.4	Начало работы схемы 1. . . . .	10
3.5	Конец работы схемы 1. . . . .	11
3.6	Пример 2. . . . .	13
3.7	Схема 2. . . . .	14
3.8	Начало работы схемы 2. . . . .	15
3.9	Передача из нового узла (схема 2) и потеря пакетов из очереди. . .	16
3.10	Конец работы схемы 2. . . . .	17
3.11	Пример 3. . . . .	18
3.12	Схема 3. . . . .	19
3.13	Начало работы схемы 3. . . . .	20
3.14	Разрыв соединения между 1 и 2 и передача по альтернативному пути. . . . .	21
3.15	Восстановление соединения и передача по кратчайшему пути. . .	22
3.16	Конец работы схемы 3. . . . .	23
3.17	Схема для упражнения. . . . .	24
3.18	Упражнение. . . . .	25
3.19	Схема упражнения. . . . .	26
3.20	Начало работы схемы упражнения. . . . .	27
3.21	Начало передачи пакетов. . . . .	28
3.22	Разрыв соединения между 1 и 0 и передача по альтернативному пути. . . . .	29
3.23	Восстановление соединения и передача по кратчайшему пути. . .	30
3.24	Конец работы схемы 3. . . . .	31

## **Список таблиц**

# **1 Цель работы**

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

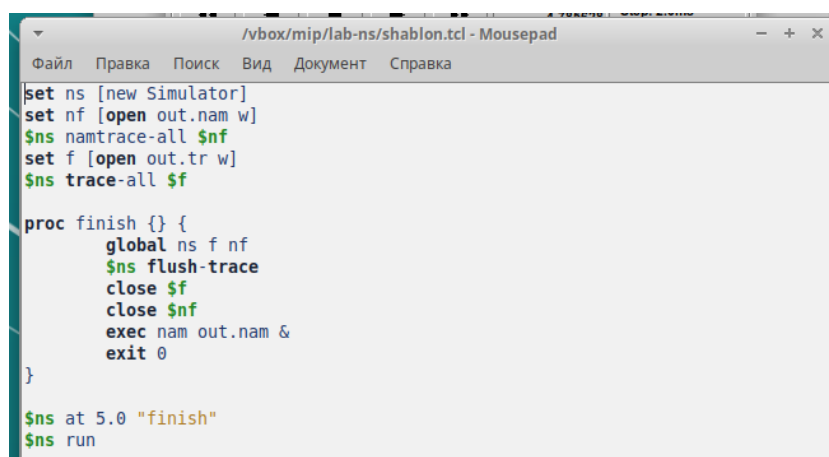
## 2 Задание

- Создать шаблон для выполнения заданий.
- Смоделировать сеть из двух узлов.
- Смоделировать сеть из трёх узлов.
- Смоделировать кольцевую сеть из 7 узлов.
- Смоделировать кольцевую сеть из 5 узлов и 1 некольцевого узла.

## 3 Выполнение лабораторной работы

### 3.0.1 Шаблон

Сначала я создала шаблон для выполнения заданий согласно описанию из лабораторной работы (рис. 3.1):



```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
set f [open out.tr w]
$ns trace-all $f

proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam out.nam &
    exit 0
}

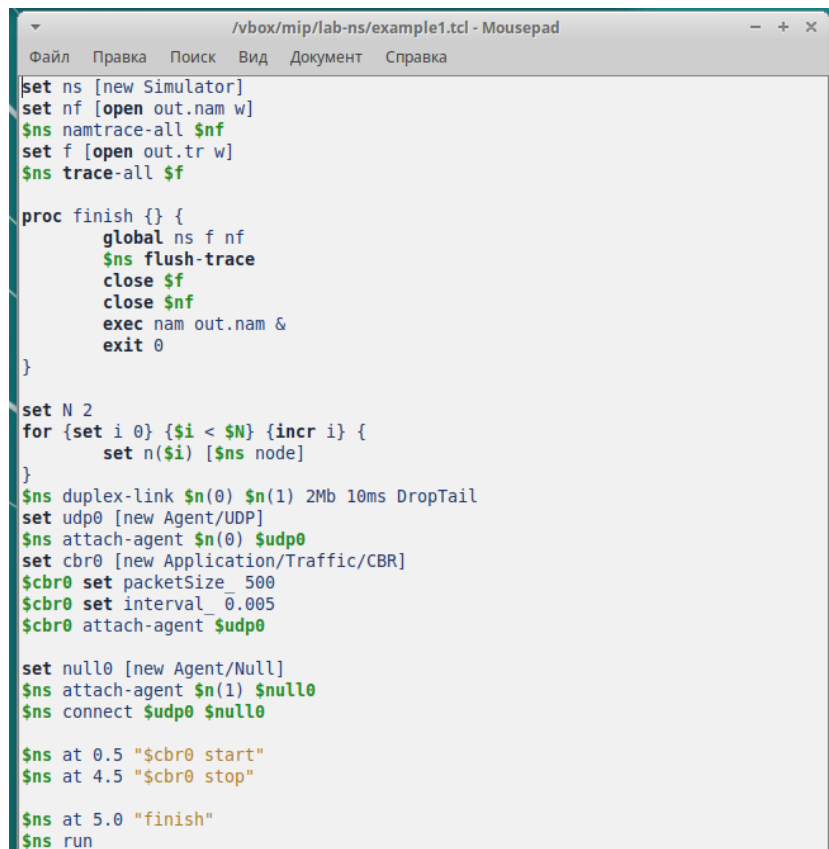
$ns at 5.0 "finish"
$ns run
```

Рис. 3.1: Шаблон.

### 3.0.2 Два узла

**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.

Для рассмотрения этого примера я создала следующий файл example1.tcl (рис. 3.2):



```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
set f [open out.tr w]
$ns trace-all $f

proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    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
```

Рис. 3.2: Пример 1.

Получилась следующая схема (рис. 3.3):



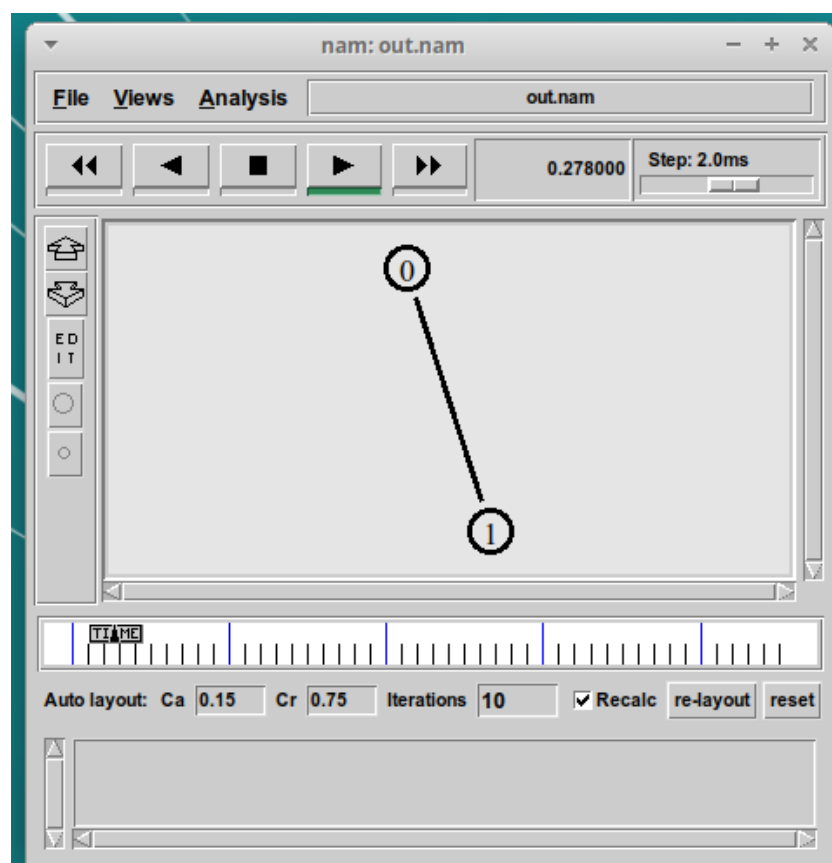


Рис. 3.3: Схема 1.

Она работала (рис. 3.4 и 3.5):

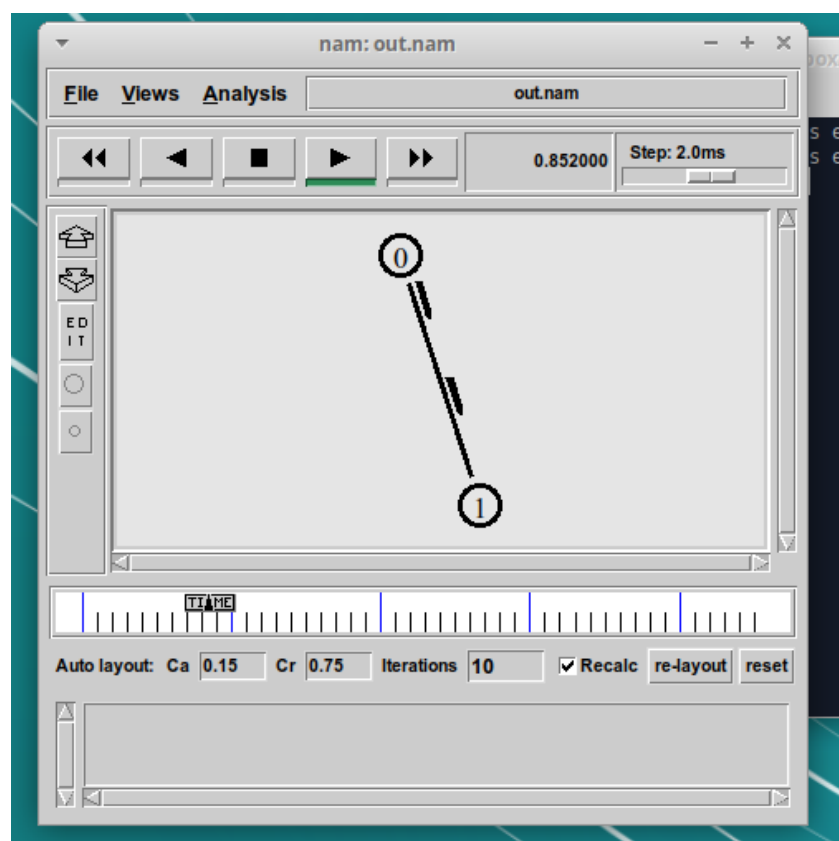


Рис. 3.4: Начало работы схемы 1.

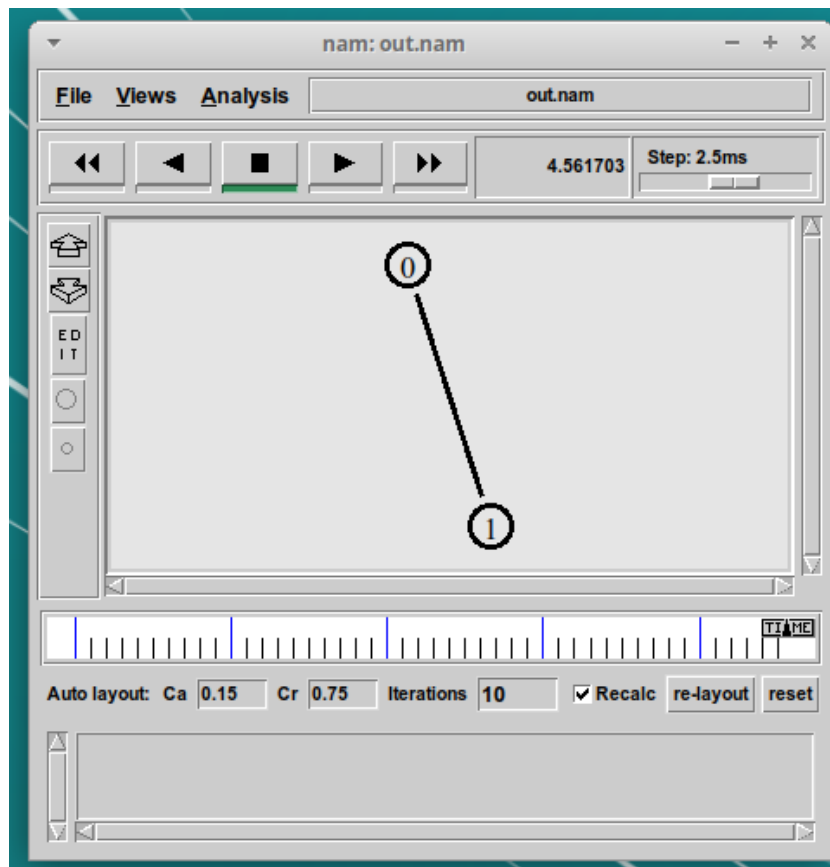


Рис. 3.5: Конец работы схемы 1.

### 3.0.3 Три узла, усложнённая топология

**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 n1, подключён к null-агенту на узле n3 (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.

Для рассмотрения этого примера я создала следующий файл `example2.tcl` (рис. 3.6):

```

/vbox/mip/lab-ns/example2.tcl - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка

set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
set f [open out.tr w]
$ns trace-all $f

proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam out.nam &
    exit 0
}

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

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 tcp1 [new Agent/TCP]
$ns attach-agent $n(1) $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 $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"

$ns at 5.0 "finish"
$ns run

```

Рис. 3.6: Пример 2.

Получилась следующая схема (рис. 3.7):

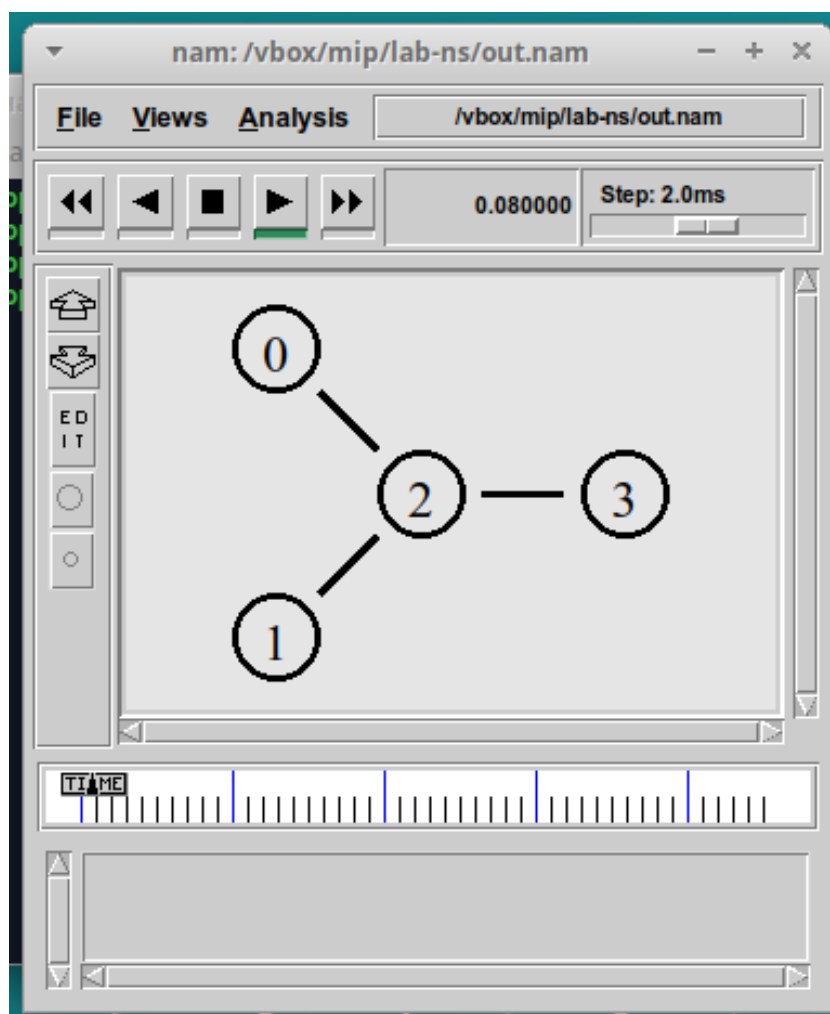


Рис. 3.7: Схема 2.

Она работала (рис. 3.8, 3.9 и 3.10):

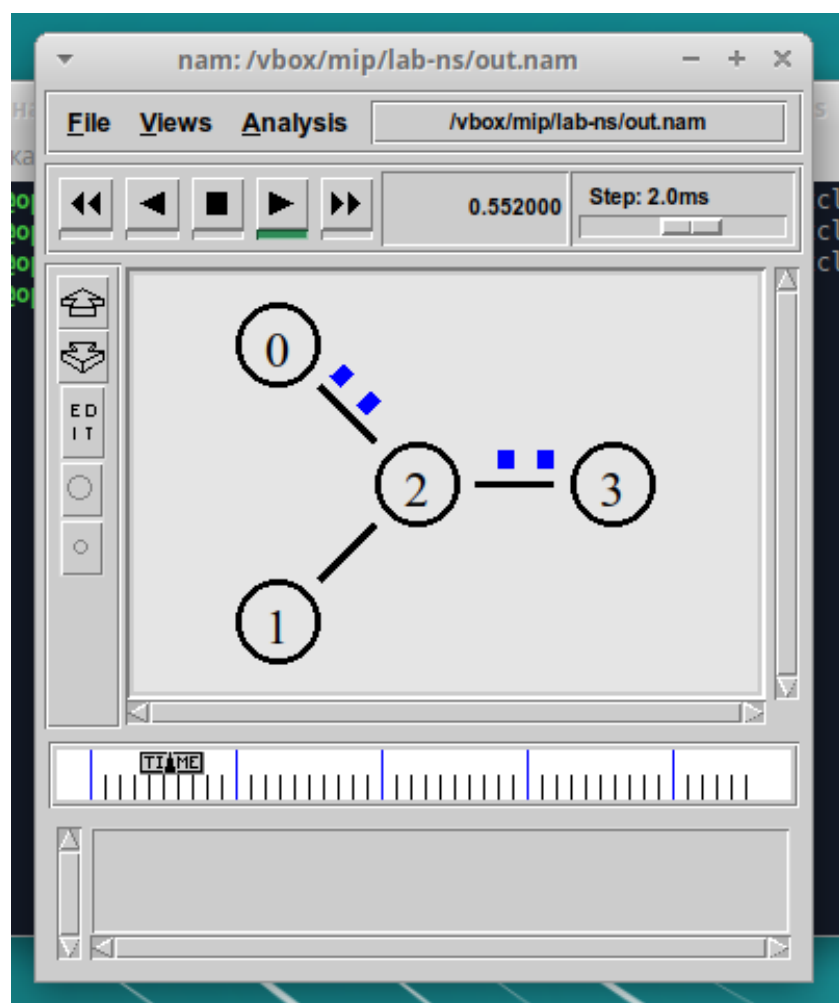


Рис. 3.8: Начало работы схемы 2.

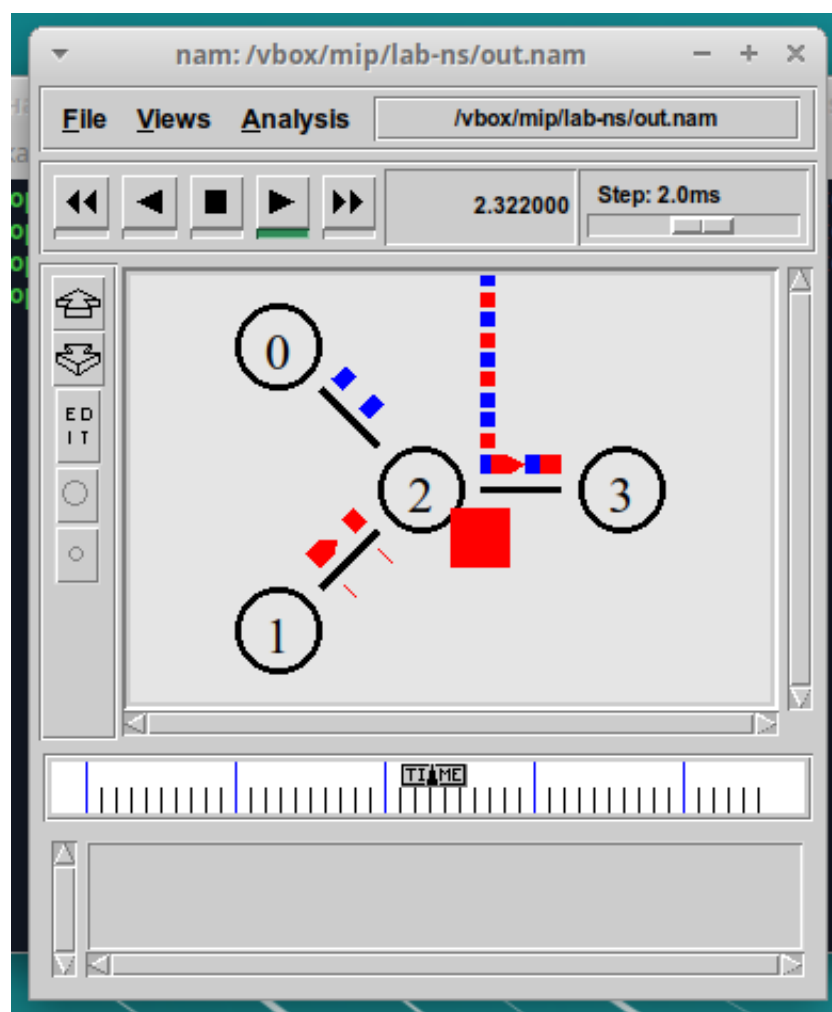


Рис. 3.9: Передача из нового узла (схема 2) и потеря пакетов из очереди.



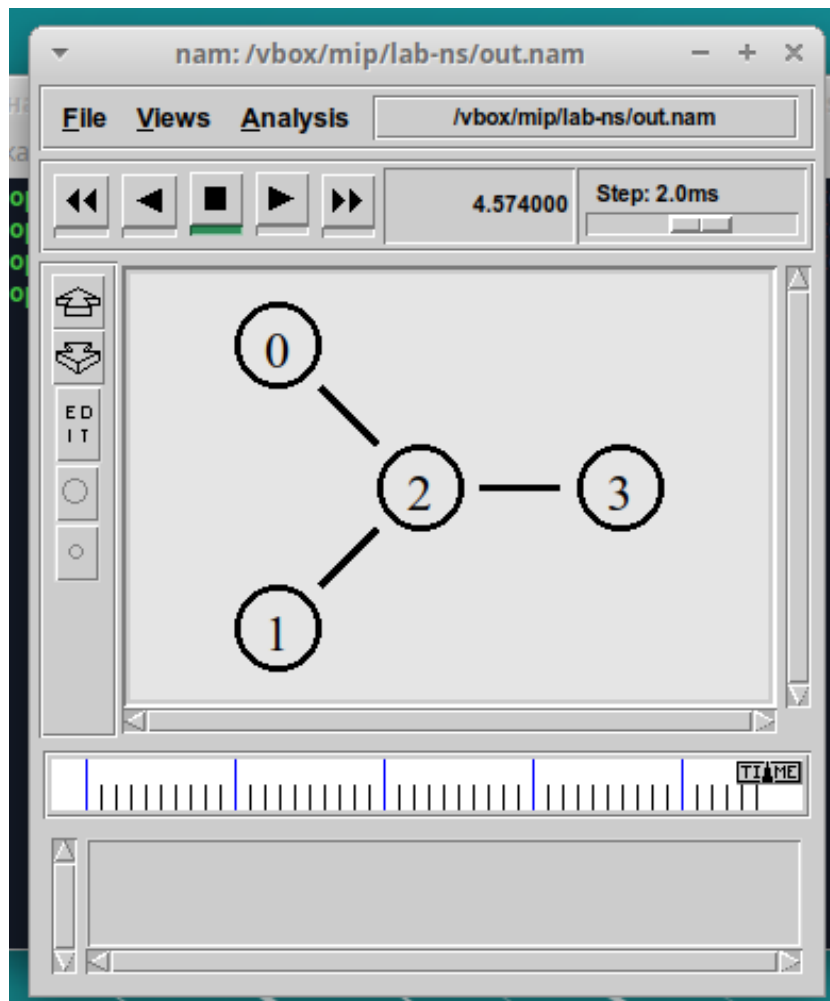


Рис. 3.10: Конец работы схемы 2.

### 3.0.4 Кольцевая топология

**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.

Для рассмотрения этого примера я создала следующий файл `example3.tcl` (рис. 3.11):

```

/vbox/mip/lab-ns/example3.tcl - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка

set ns [new Simulator]
$ns rtproto DV
set nf [open out.nam w]
$ns namtrace-all $nf
set f [open out.tr w]
$ns trace-all $f

proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam out.nam &
    exit 0
}

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 rtmodel-at 1.0 down $n(1) $n(2)
$ns rtmodel-at 2.0 up $n(1) $n(2)
$ns at 4.5 "$cbr0 stop"
$ns at 5.0 "finish"

$ns run

```

Рис. 3.11: Пример 3.

Получилась следующая схема (рис. 3.12):

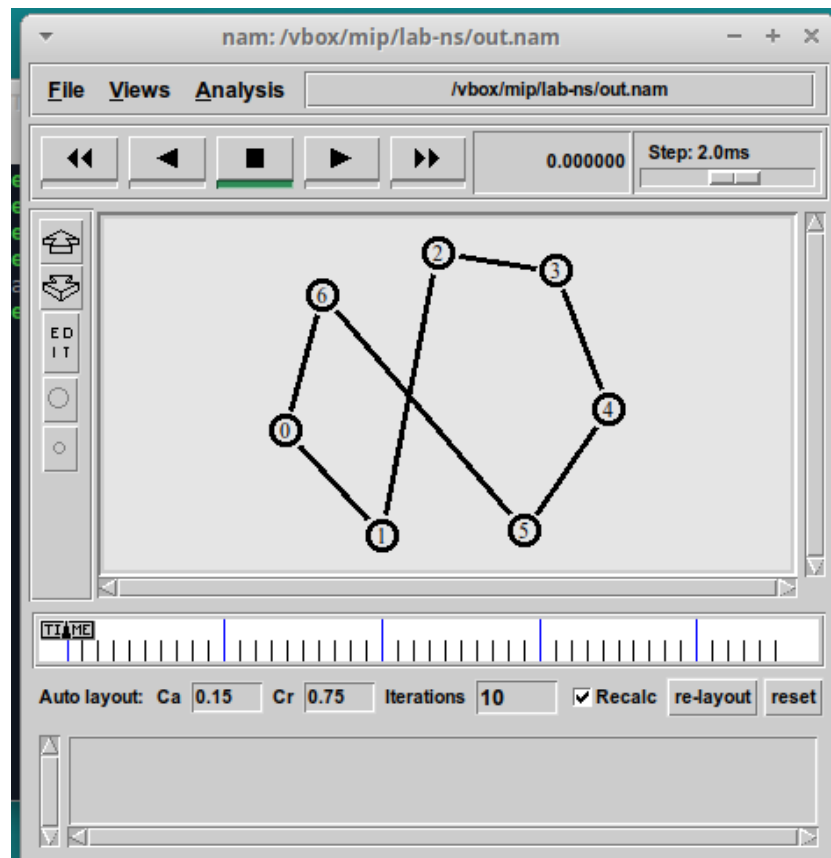


Рис. 3.12: Схема 3.

Она работала (рис. 3.13, 3.14, 3.15 и 3.16):

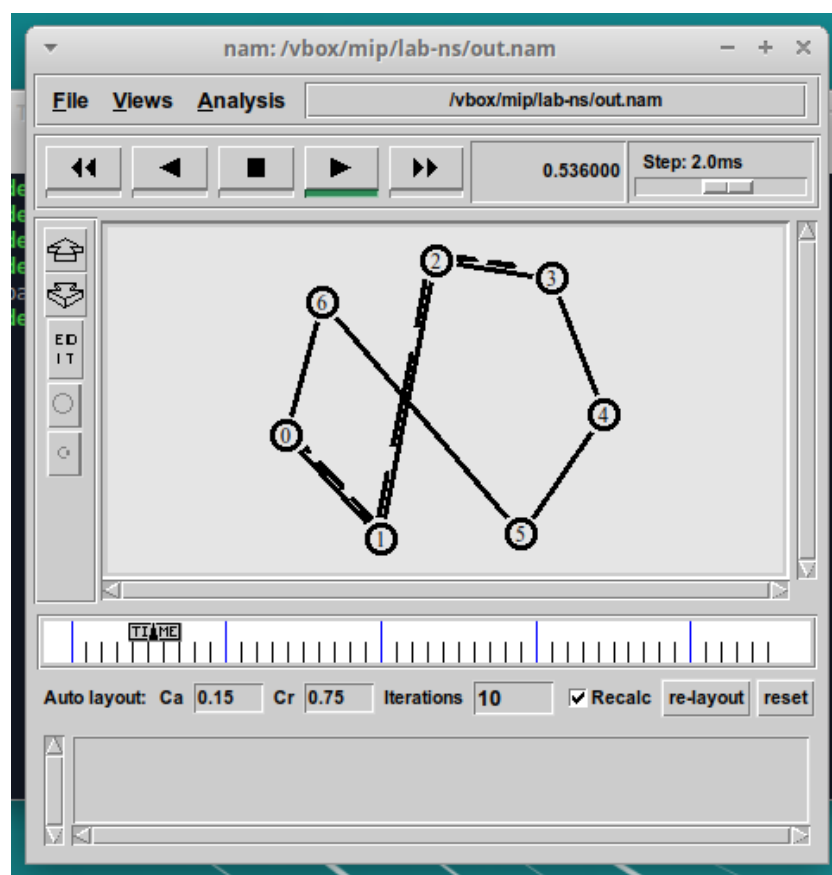


Рис. 3.13: Начало работы схемы 3.

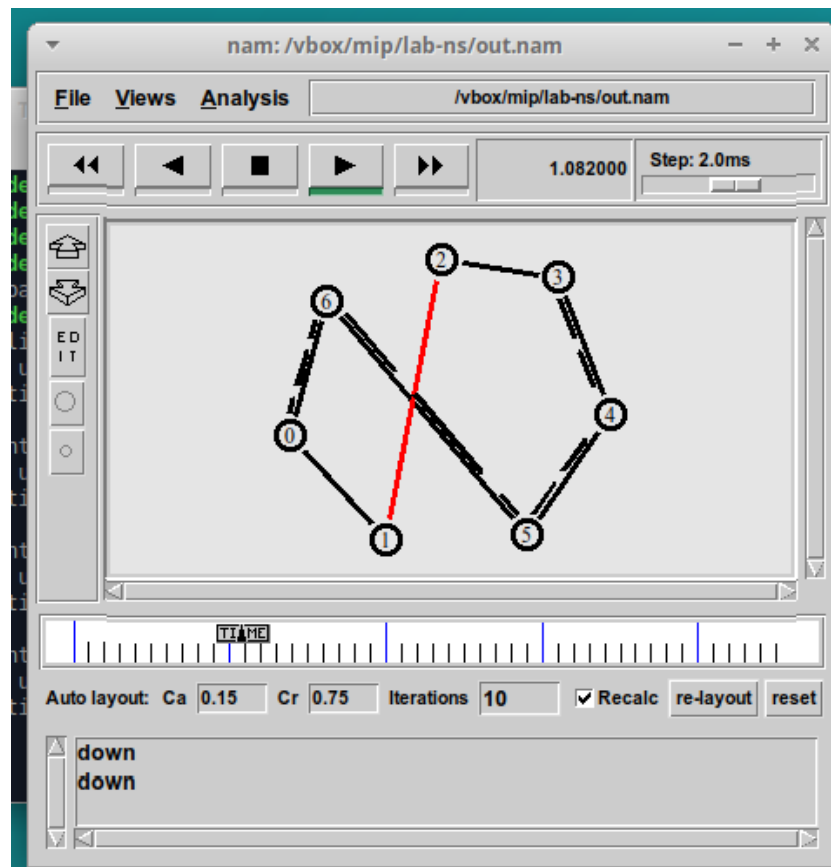


Рис. 3.14: Разрыв соединения между 1 и 2 и передача по альтернативному пути.

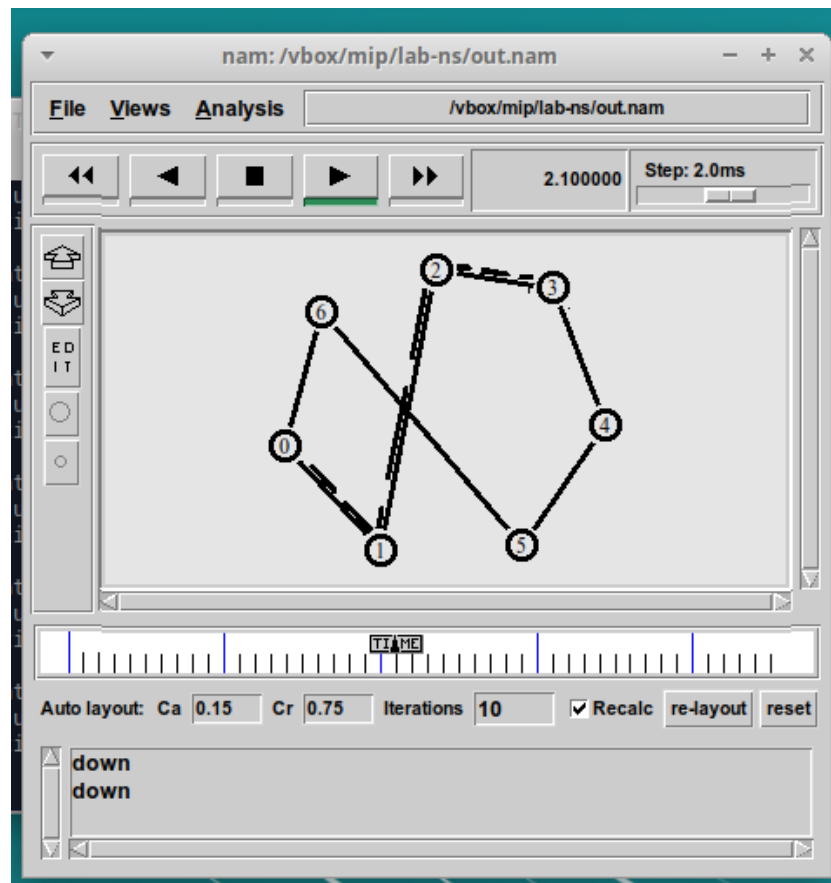


Рис. 3.15: Восстановление соединения и передача по кратчайшему пути.

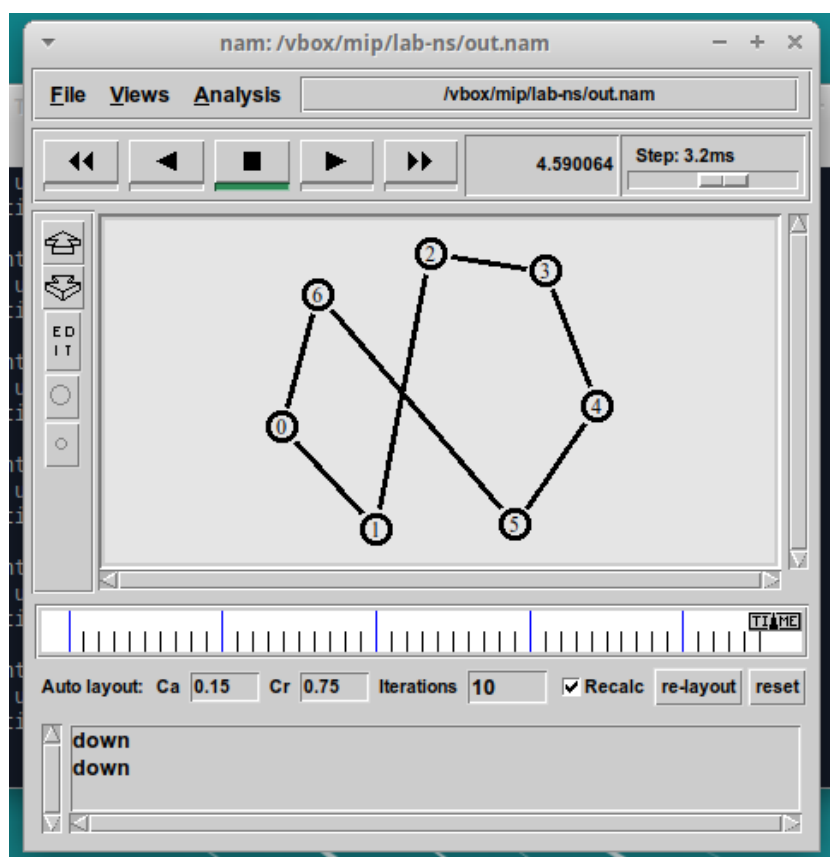


Рис. 3.16: Конец работы схемы 3.

### 3.0.5 Доработка схемы

**Упражнение.** Внесите следующие изменения в реализацию примера с кольцевой топологией сети: - топология сети должна соответствовать представленной на рис. 3.17:

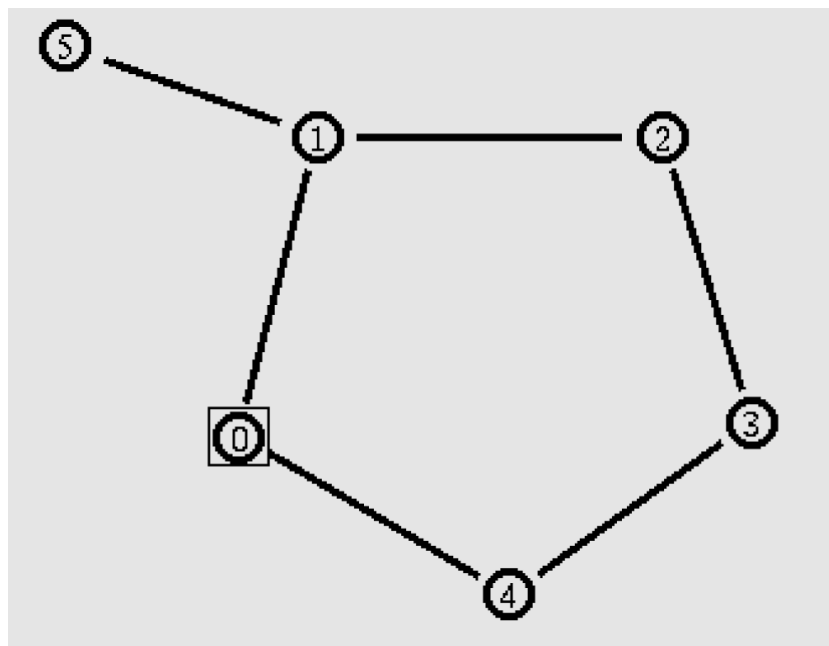


Рис. 3.17: Схема для упражнения.

- 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.

Для выполнения этого упражнения я создала следующий файл `task.tcl` (рис. 3.18):



```

/vbox/mip/lab-ns/task.tcl - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка

set ns [new Simulator]
$ns rtproto DV
set nf [open out.nam w]
$ns namtrace-all $nf
set f [open out.tr w]
$ns trace-all $f

proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam out.nam &
    exit 0
}

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

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

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

$ns run

```

Рис. 3.18: Упражнение.

Получилась следующая схема (рис. 3.12):

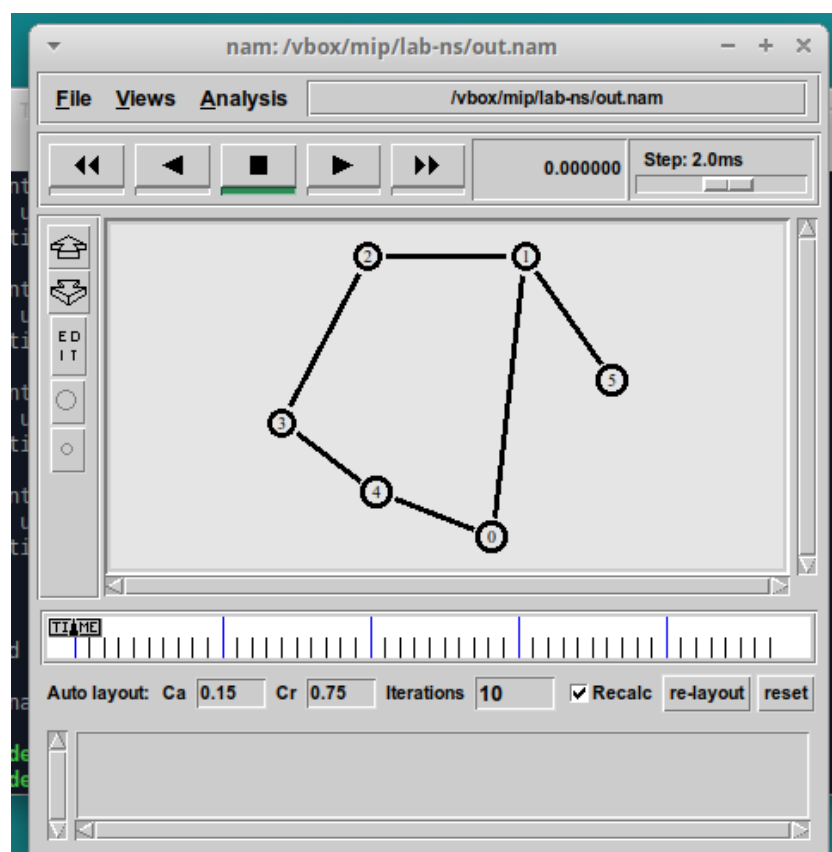


Рис. 3.19: Схема упражнения.

Она работала (рис. 3.20, 3.21, 3.22, 3.23 и 3.24):

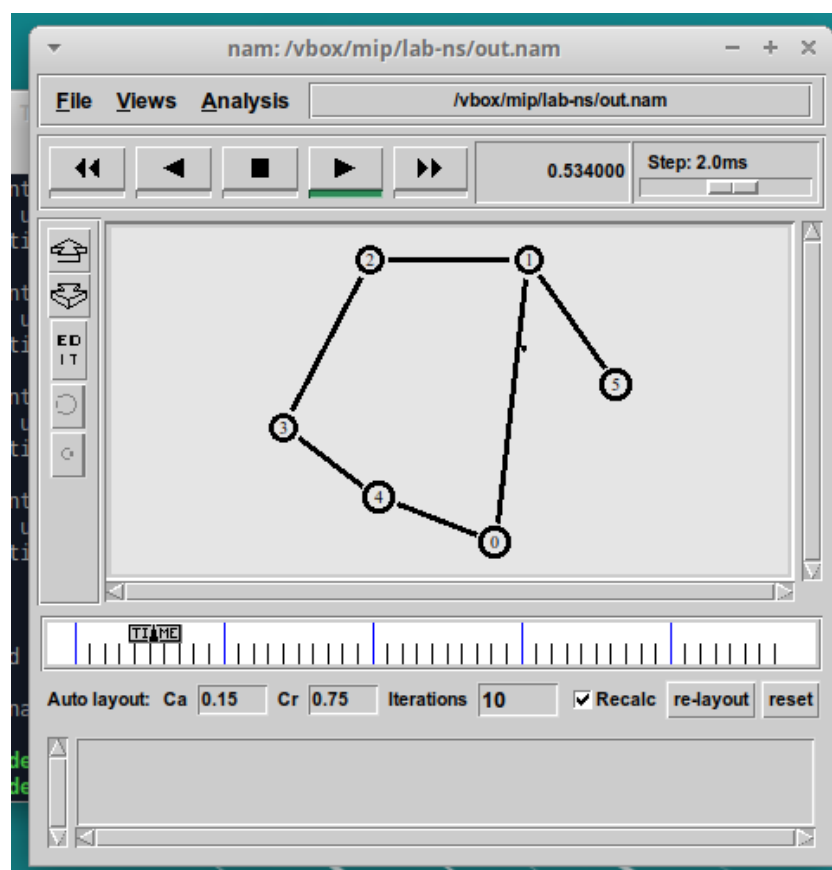


Рис. 3.20: Начало работы схемы упражнения.

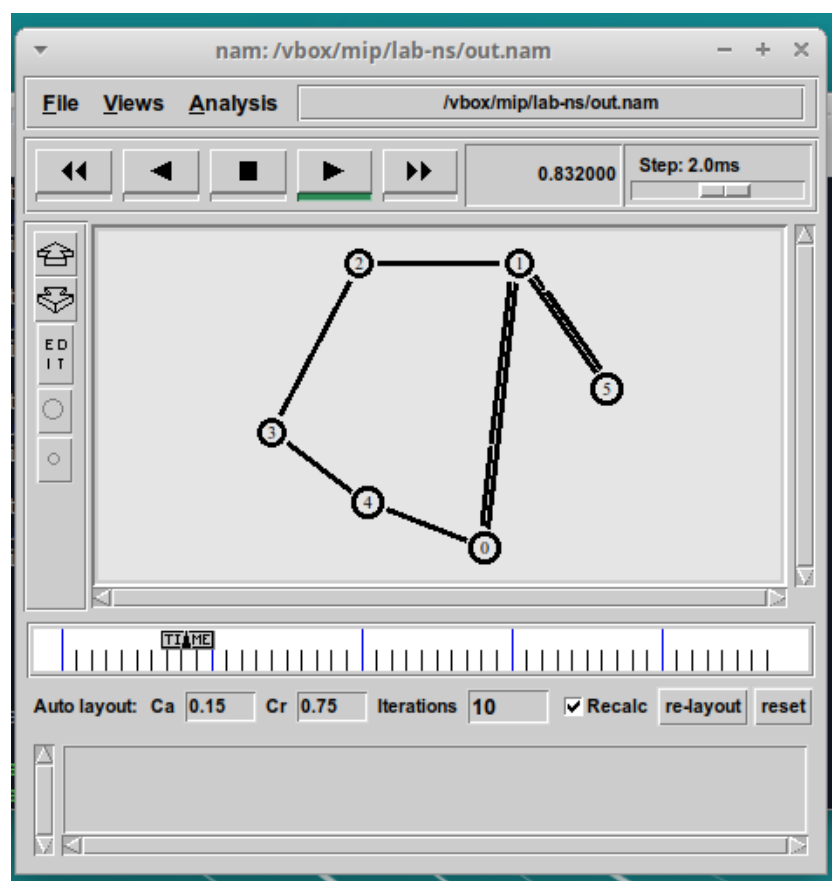


Рис. 3.21: Начало передачи пакетов.

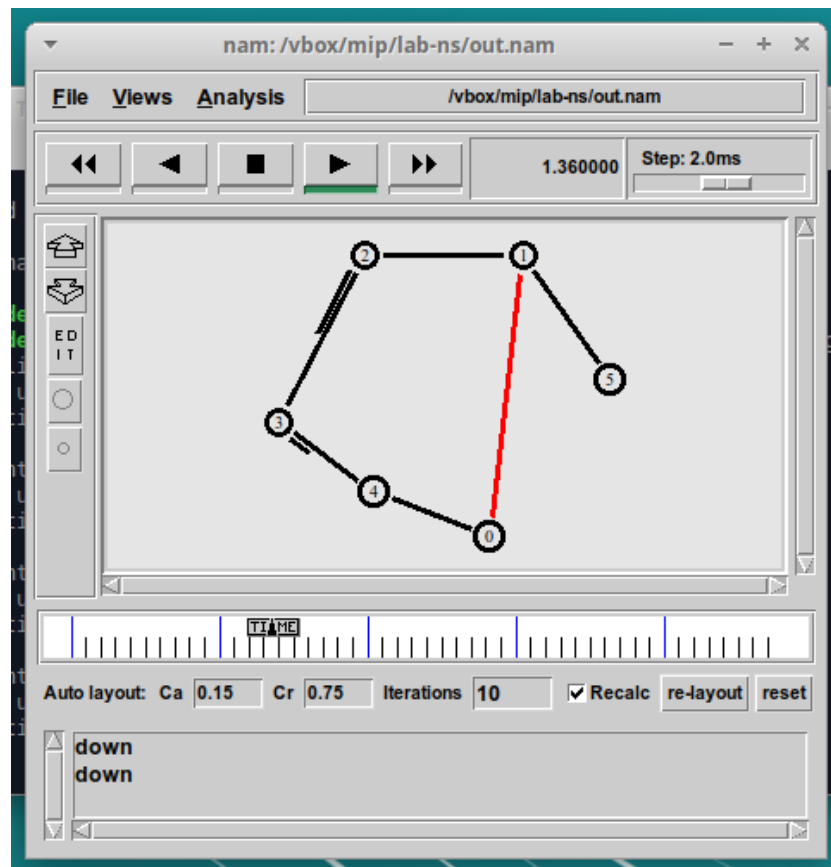


Рис. 3.22: Разрыв соединения между 1 и 0 и передача по альтернативному пути.

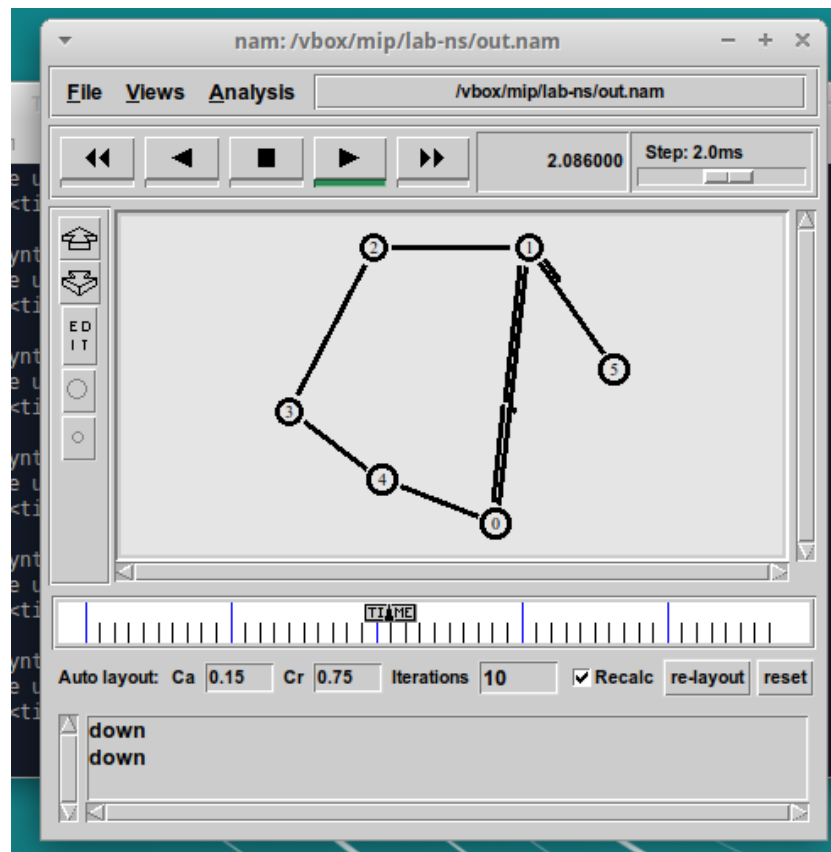


Рис. 3.23: Восстановление соединения и передача по кратчайшему пути.

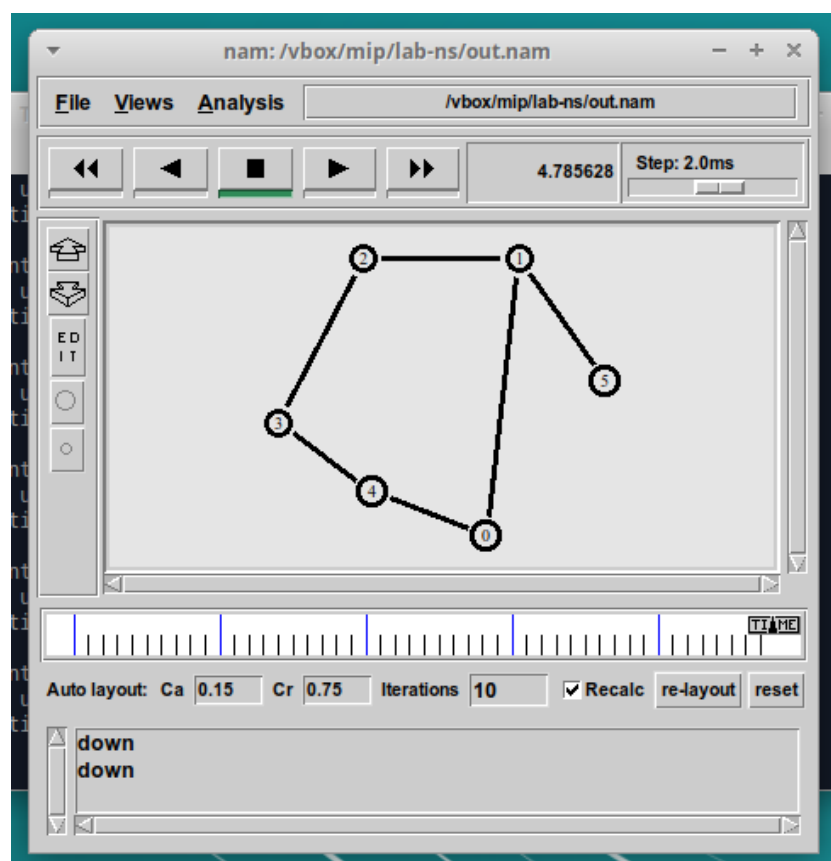


Рис. 3.24: Конец работы схемы 3.

## **4 Выводы**

Приобрели навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, проанализировали полученные результаты моделирования и доработали схему.