

# **Лабораторная работа №1**

**Дисциплина: Имитационное моделирование**

Пронякова Ольга Максимовна

# 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>
<b>4</b>	<b>Выводы</b>	<b>17</b>
	<b>Список литературы</b>	<b>18</b>

## Список иллюстраций

3.1	Заполнение файла shablon.tcl . . . . .	7
3.2	Заполнение файла tcl . . . . .	8
3.3	Результат работы программы . . . . .	9
3.4	Заполнение файла tcl . . . . .	11
3.5	Результат работы программы . . . . .	12
3.6	Заполнение файла tcl . . . . .	13
3.7	Результат работы программы . . . . .	14
3.8	Заполнение файла tcl . . . . .	15
3.9	Результат работы программы . . . . .	16

## **Список таблиц**

# 1 Цель работы

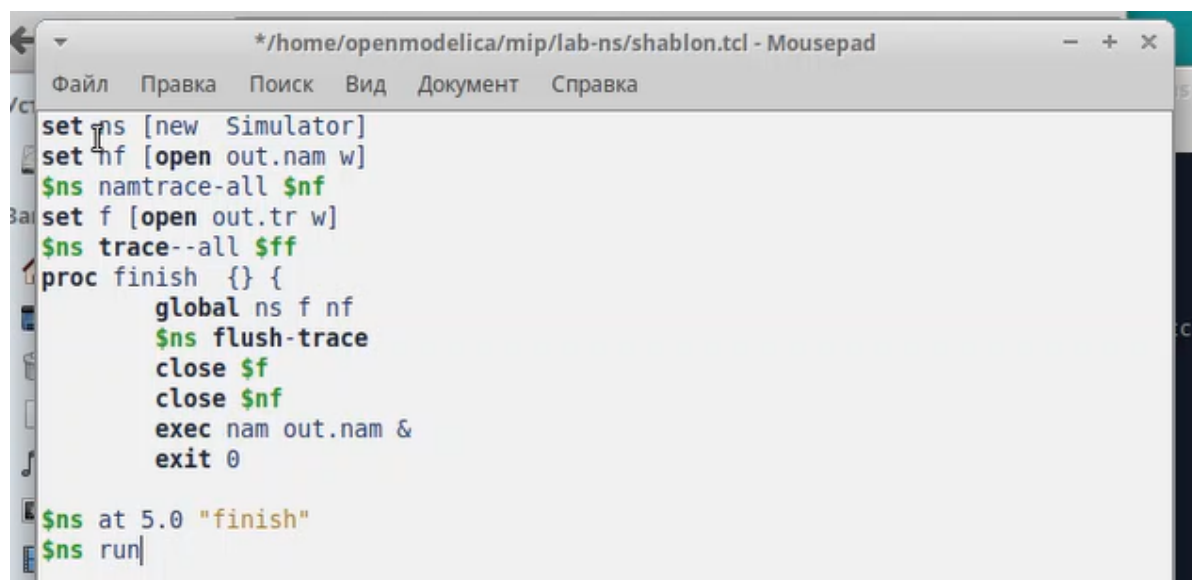
Приобретение навыков моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также анализ полученных результатов моделирования.

## 2 Задание

1. Шаблон сценария для NS-2
2. Простой пример описания топологии сети, состоящей из двух узлов и одного соединения
3. Пример с усложнённой топологией сети
4. Пример с кольцевой топологией сети
5. Упражнение

### 3 Выполнение лабораторной работы

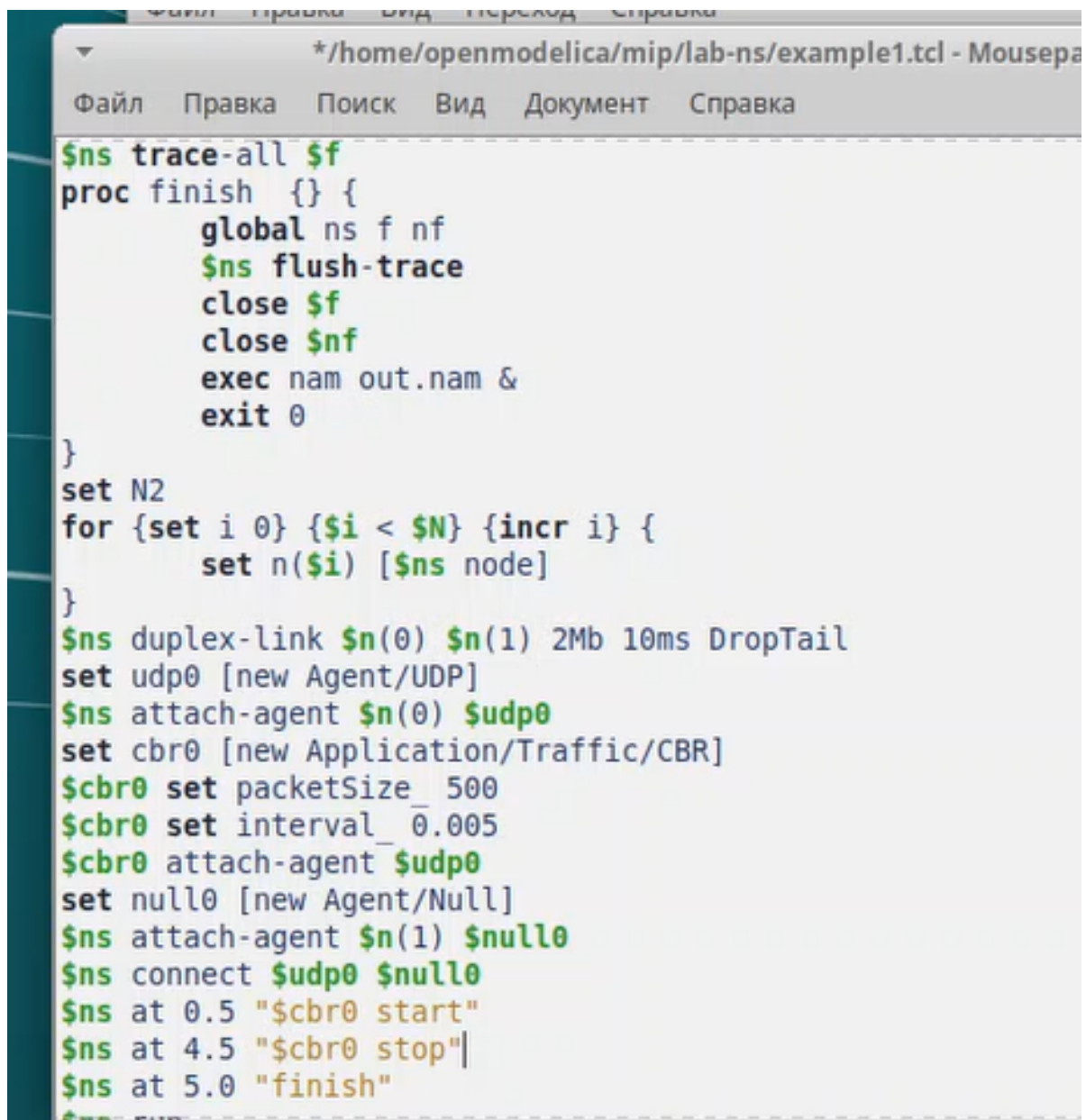
В своём рабочем каталоге создаю директорию `mip`, в которой будут выполняться лабораторные работы. Внутри `mip` создаю директорию `lab-ns`, а в ней файл `shablon.tcl`. Открываю на редактирование файла `shablon.tcl` и заполняю его по инструкции(рис.3.1).



```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
set f [open out.tr w]
$ns trace--all $ff
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: Заполнение файла `shablon.tcl`

Нужно 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. Копирую содержимое предыдущего файла в новый и заполняю по образцу(рис.3.2), (рис.3.3).

A screenshot of a text editor window titled `*/home/openmodelica/mip/lab-ns/example1.tcl - Mousepa`. The editor has a menu bar with options: `Файл`, `Правка`, `Поиск`, `Вид`, `Документ`, and `Справка`. The main text area contains a TCL script for network simulation. The script starts with `$ns trace-all $f`, followed by a `proc finish {} {}` procedure that flushes traces, closes files, and executes `nam out.nam`. It then sets `N2` and loops to create nodes `n(0)` to `n(N2)`. A duplex link is created between `n(0)` and `n(1)` with 2Mb buffer, 10ms delay, and DropTail queue. A UDP agent `udp0` is attached to `n(0)`, and a CBR application `cbr0` is attached to it with packet size 500 and interval 0.005. A null agent `null0` is attached to `n(1)`, and the UDP agent is connected to it. The simulation starts at 0.5s, stops at 4.5s, and finishes at 5.0s.

```
$ns trace-all $f
proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam out.nam &
    exit 0
}
set N2
for {set i 0} {$i < $N2} {incr i} {
    set n($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"
```

Рис. 3.2: Заполнение файла tcl



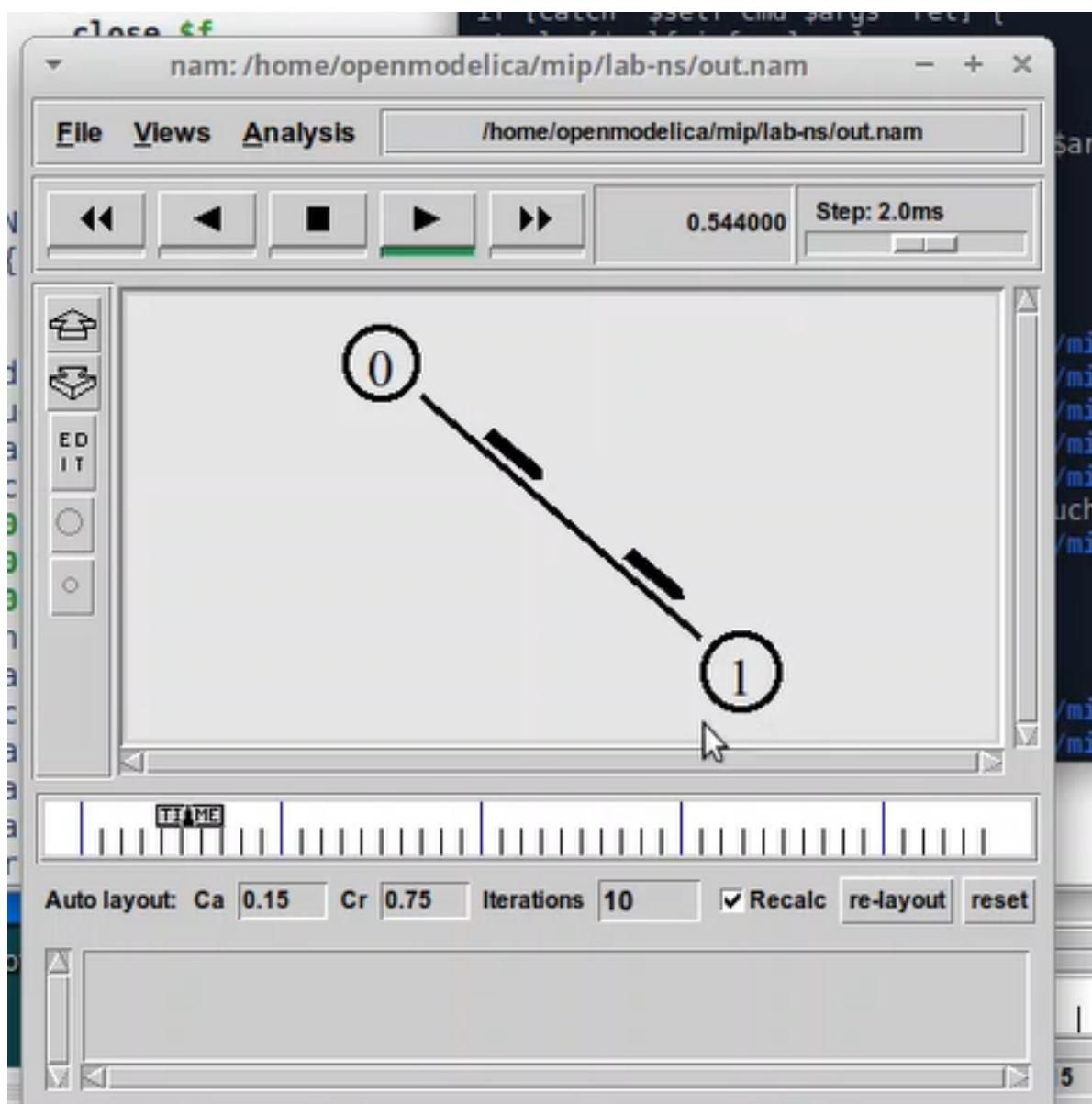
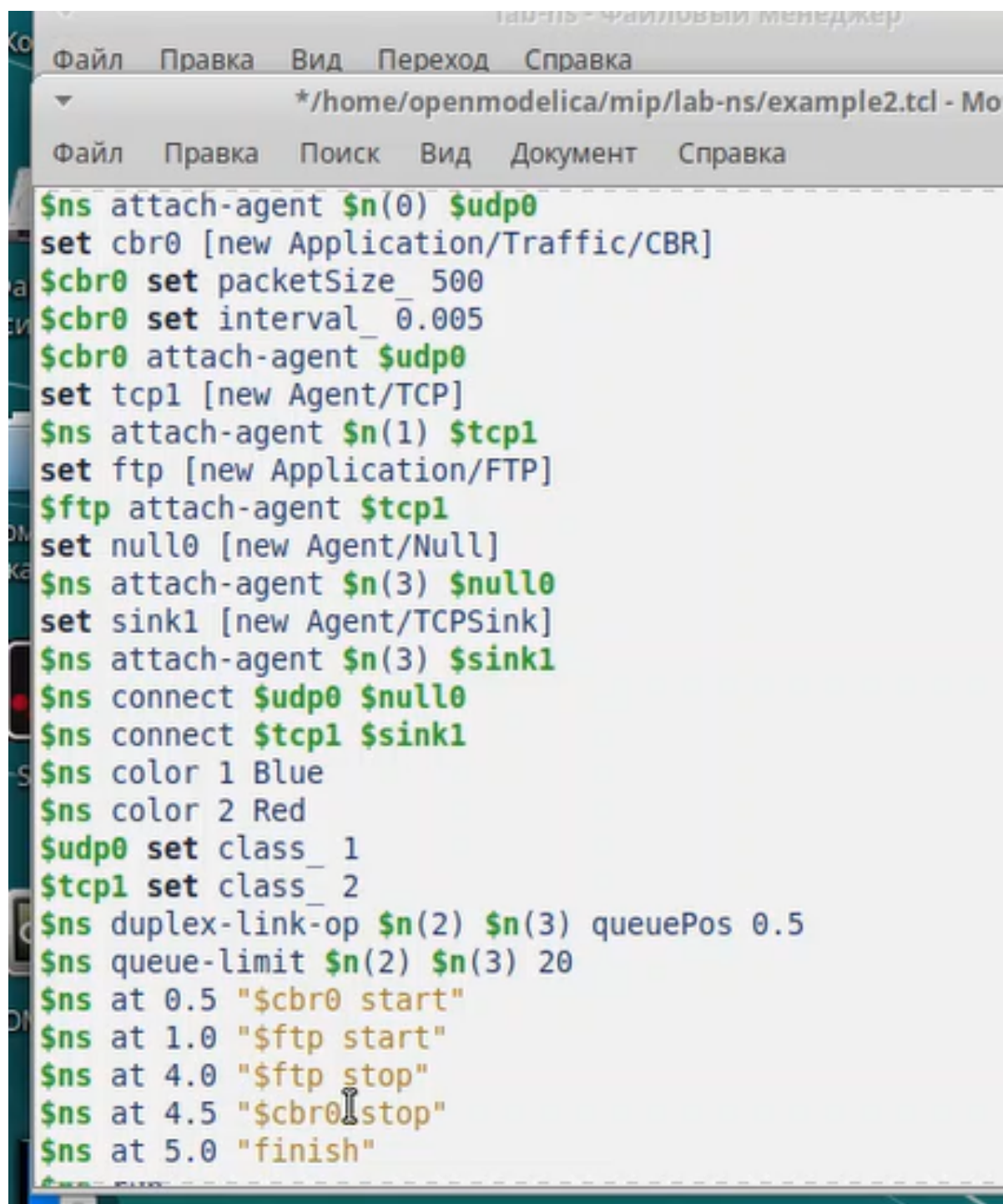


Рис. 3.3: Результат работы программы

- сеть состоит из 4 узлов
- между узлами n0 и n2, n1 и n2 установлено дуплексное соединение с пропускной способностью 2 Мбит/с и задержкой 10 мс;
- между узлами n2 и n3 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 n0 подключается к TCP-приёмнику на узле n3 (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

Копирую содержимое предыдущего файла в новый и заполняю по образцу(рис.3.4), (рис.3.5).



The image shows a screenshot of a text editor window, likely a file manager, displaying a TCL script. The window has a menu bar with options: 'Файл' (File), 'Правка' (Edit), 'Вид' (View), 'Переход' (Go), and 'Справка' (Help). The title bar indicates the file path: `*/home/openmodelica/mip/lab-ns/example2.tcl - Мо`. Below the menu bar, there is another set of menu options: 'Файл', 'Правка', 'Поиск' (Search), 'Вид', 'Документ' (Document), and 'Справка'. The main text area contains a TCL script for setting up a network simulation. The script defines agents (Cbr, TcP, Ftp, Null, Sink), connects them, sets colors, and schedules events. The script is as follows:

```
$ns attach-agent $n(0) $udp0
set cbr0 [new Application/Traffic/CBR]
$сbr0 set packetSize_ 500
$сbr0 set interval_ 0.005
$сbr0 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
$ns at 0.5 "$сbr0 start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$сbr0 stop"
$ns at 5.0 "finish"
```

Рис. 3.4: Заполнение файла tcl

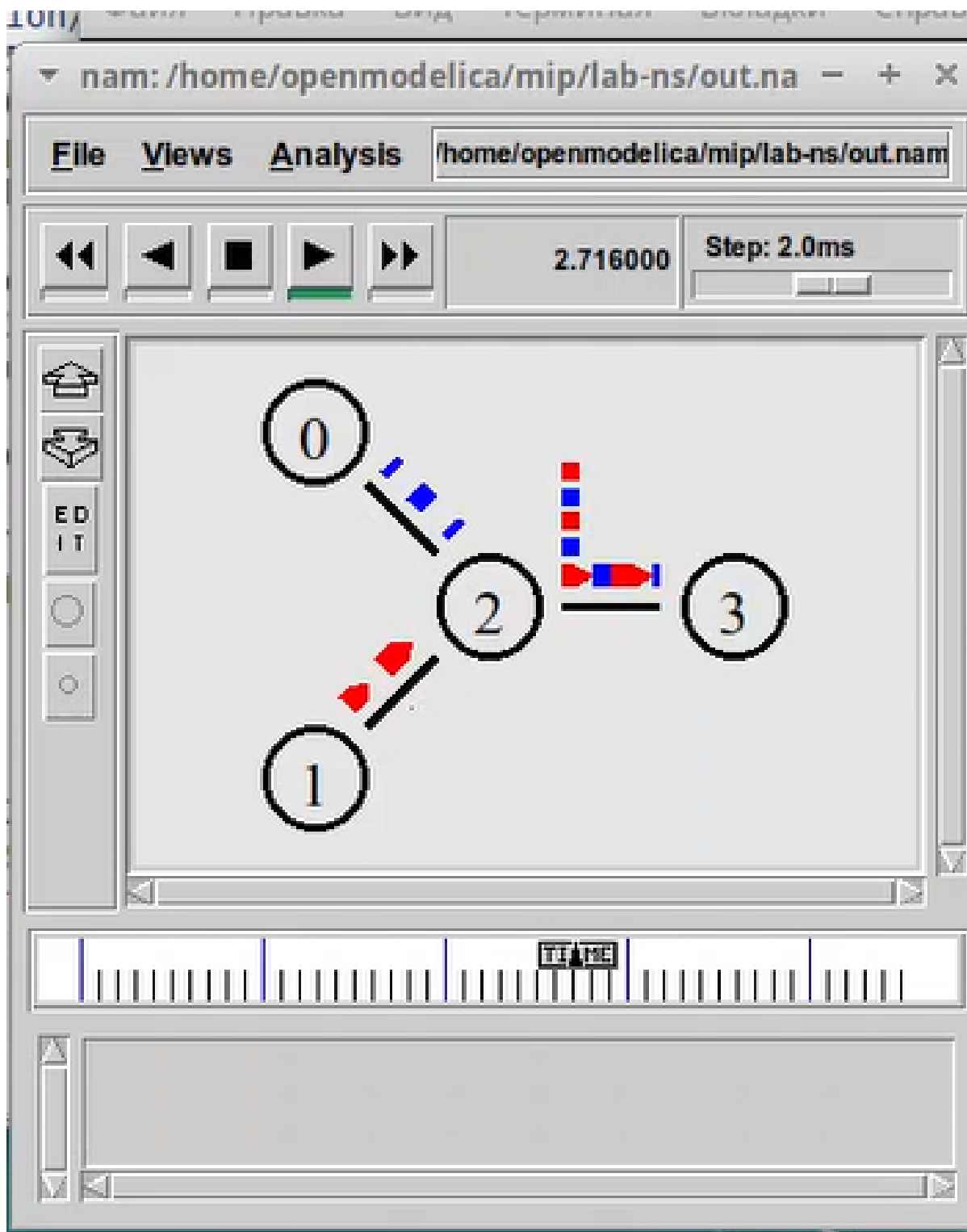
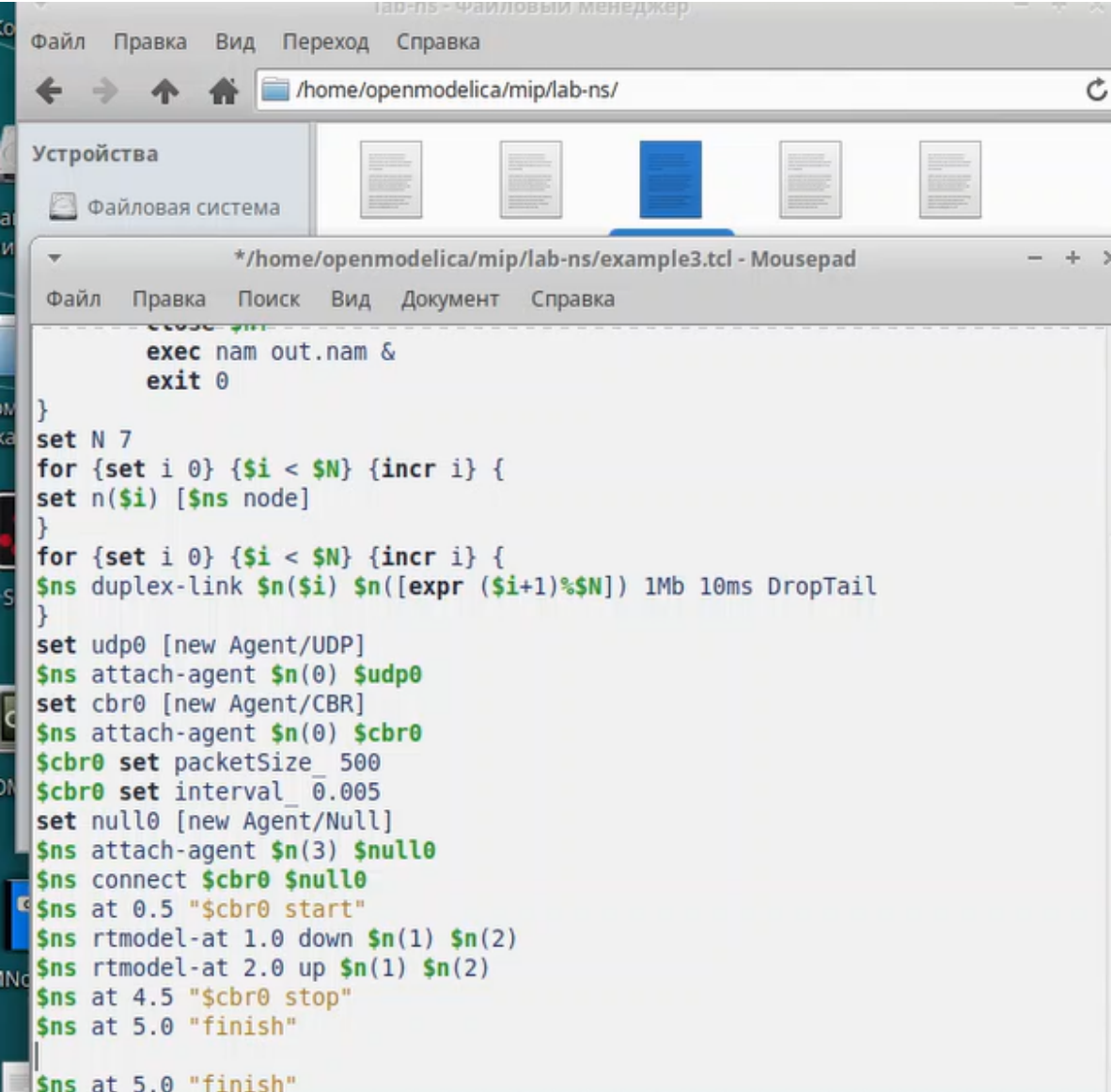


Рис. 3.5: Результат работы программы

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

Копирую содержимое предыдущего файла в новый и заполняю по образцу(рис.3.6), (рис.3.7).



The image shows a file manager window titled 'lab-ns - файловый менеджер' with the address bar set to '/home/openmodelica/mip/lab-ns/'. Below it, a text editor window titled '\* /home/openmodelica/mip/lab-ns/example3.tcl - Mousepad' displays the following tcl code:

```

close $f
exec nam out.nam &
exit 0
}
set N 7
for {set i 0} {$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 at 5.0 "finish"

```

Рис. 3.6: Заполнение файла tcl

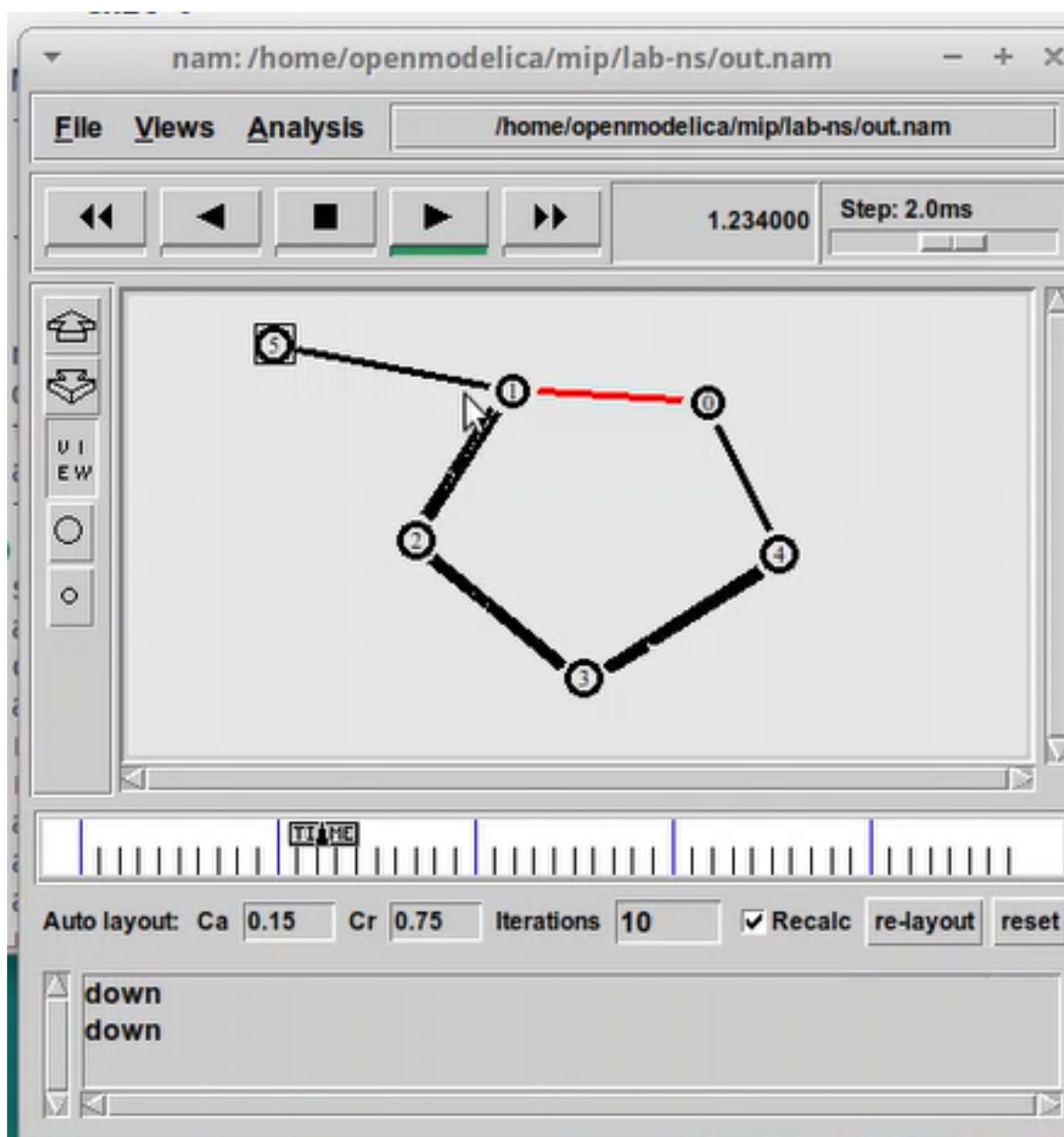


Рис. 3.7: Результат работы программы

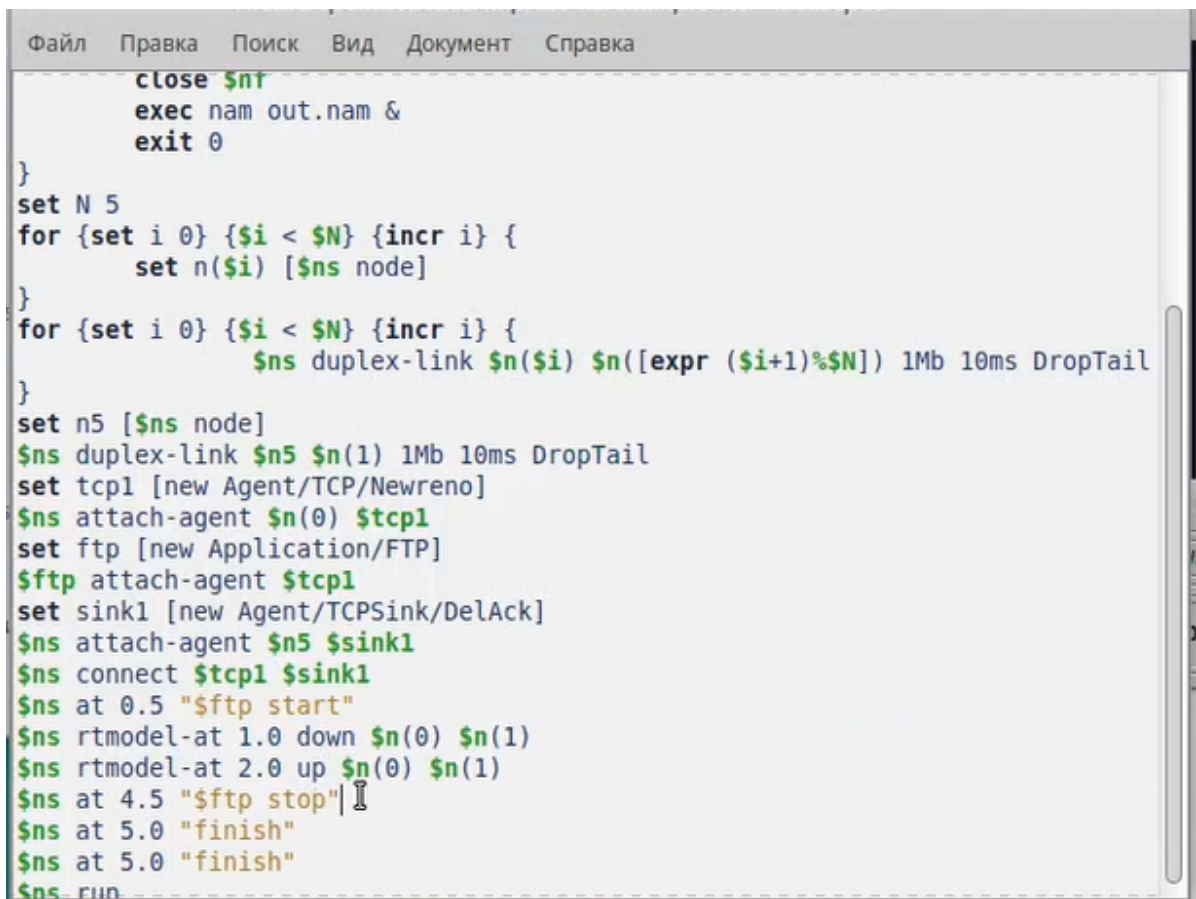
Внесите следующие изменения в реализацию примера с кольцевой топологией сети:

- передача данных должна осуществляться от узла  $n(0)$  до узла  $n(5)$  по кратчай- шему 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.

Копирую содержимое предыдущего файла в новый и добавляю корректировки (рис.3.8), (рис.3.9).



```

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 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"
$ns at 5.0 "finish"
$ns run

```

Рис. 3.8: Заполнение файла tcl

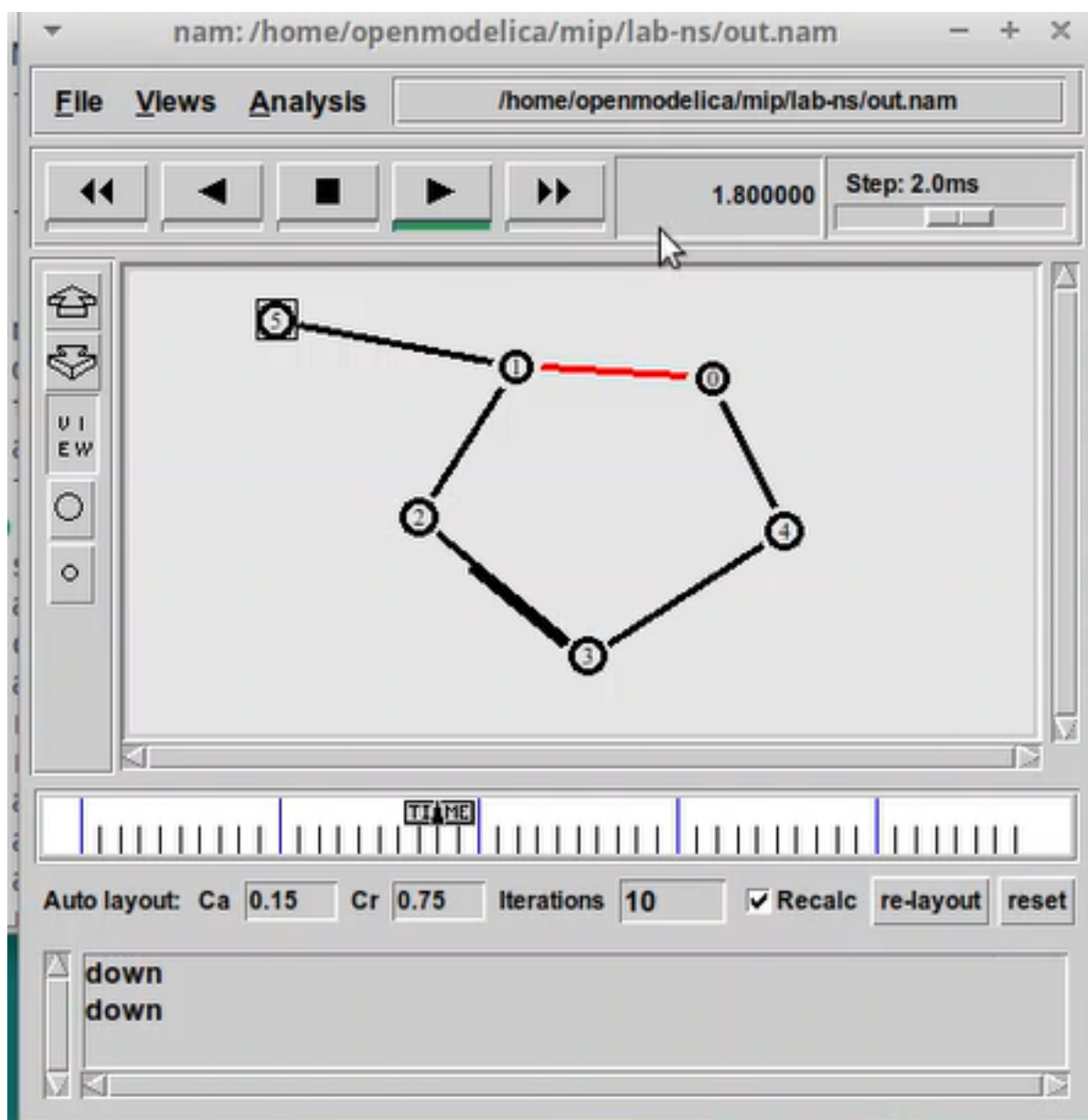


Рис. 3.9: Результат работы программы



## **4 Выводы**

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

## **Список литературы**