

# **Отчёт по лабораторной работе №1**

**Простые модели компьютерной сети**

Гэинэ Андрей НФИбд-02-22

# Содержание

Цель работы	5
Задание	6
Выполнение лабораторной работы	7
Выводы	18

# Список иллюстраций

1	Файл шаблона . . . . .	7
2	Переменная nf . . . . .	8
3	Переменная f. Процедура finish . . . . .	8
4	Запуск процедуры finish . . . . .	9
5	Запуск программы . . . . .	9
6	Редактирование example1.tcl . . . . .	10
7	Запуск рограммы . . . . .	11
8	Редактирование example2.tcl . . . . .	12
9	Запуск рограммы . . . . .	13
10	Редактирование example3.tcl . . . . .	14
11	Запуск рограммы . . . . .	14
12	Код программы из Упражнения . . . . .	15
13	Движение пакетов по кратчайшему пути . . . . .	16
14	Движение пакетов в случае разрыва соединения . . . . .	16
15	Движение пакетов по кратчайшему пути . . . . .	17

## Список таблиц

# Цель работы

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

# Задание

1. Реализовать топологию сети, состоящую из двух узлов и одного соединения в NS-2.
2. Реализовать кольцевую топологию сети в NS-2.
3. Реализовать видоизмененную кольцевую топологию сети в NS-2.

# Выполнение лабораторной работы

Создал директорию и файл шаблона (рис. [-@fig:001])

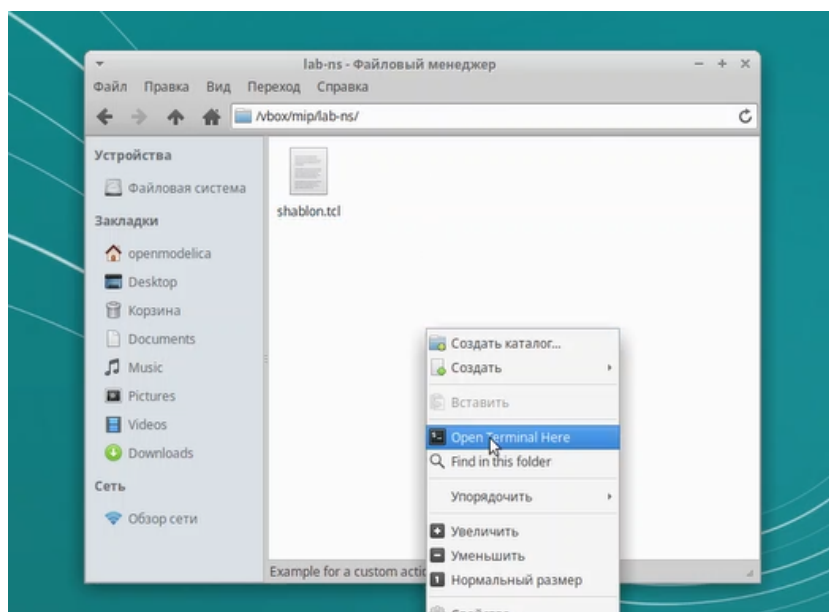


Рис. 1: Файл шаблона

Создал переменную `nf` и указал, что требуется открыть на запись `nam`-файл для регистрации выходных результатов моделирования (рис. [-@fig:002])

```

1  set ns [new Simulator]
2
3  set nf [open out.nam w]
4
5  $ns nametrace-all $nf
6

```

Рис. 2: Переменная nf

Создал переменную f и открыл на запись файл трассировки для регистрации всех событий модели. Написал процедуру finish (рис. [-@fig:003])

```

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
}

```

Рис. 3: Переменная f. Процедура finish

С помощью команды at указал планировщику событий, что процедуру finish следует запустить через 5 с после начала моделирования, после чего запустить симулятор ns (рис. [-@fig:004])



```

19
20 $ns at 5.0 "finish"
21
22 $ns run

```

Рис. 4: Запуск процедуры finish

Сохранил изменения и запустил программу (рис. [-@fig:005])

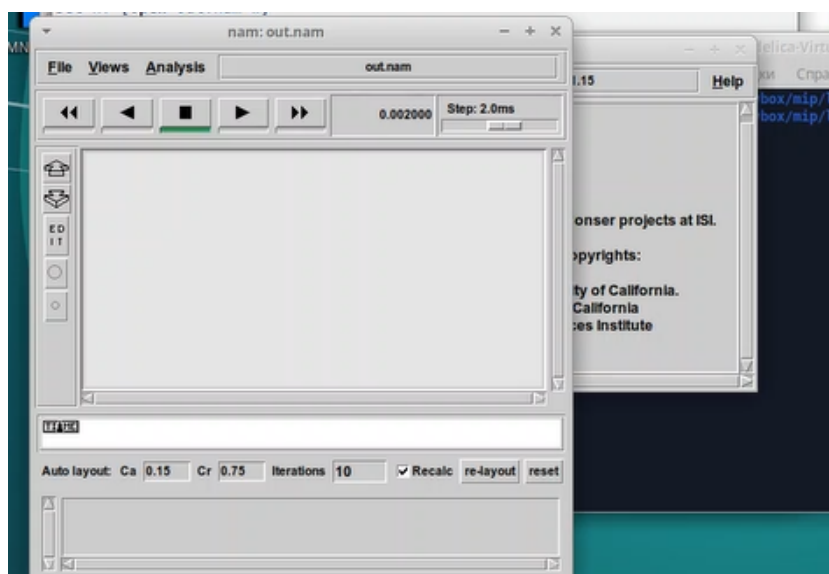
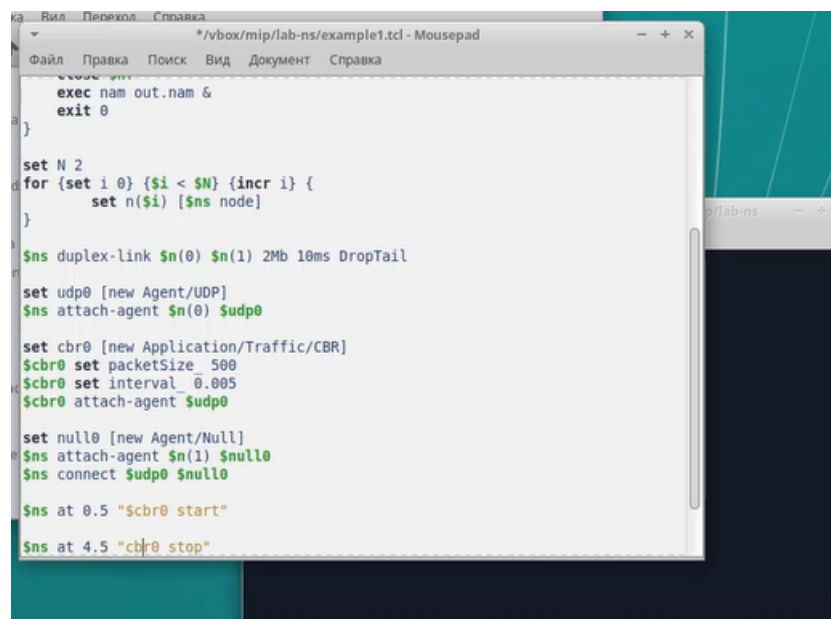


Рис. 5: Запуск программы

Создал example1.tcl. Написал программу (часть 1) (рис. [-@fig:006])



```
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"
```

Рис. 6: Редактирование example1.tcl

Запустил код программы example1.tcl. Просмотрел движение пакетов данных (рис. [-@fig:008])

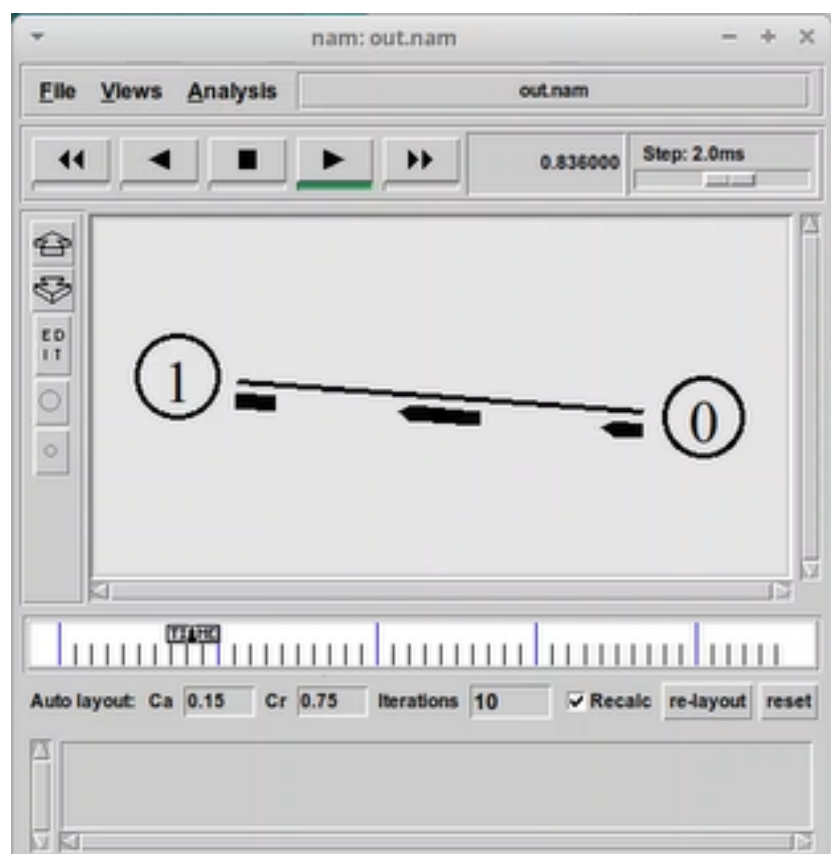


Рис. 7: Запуск программы

Создал новый файл example2.tcl. В нем создал 4 узла и 3 дуплексных соединения с указанием направления. (рис. [-@fig:009])

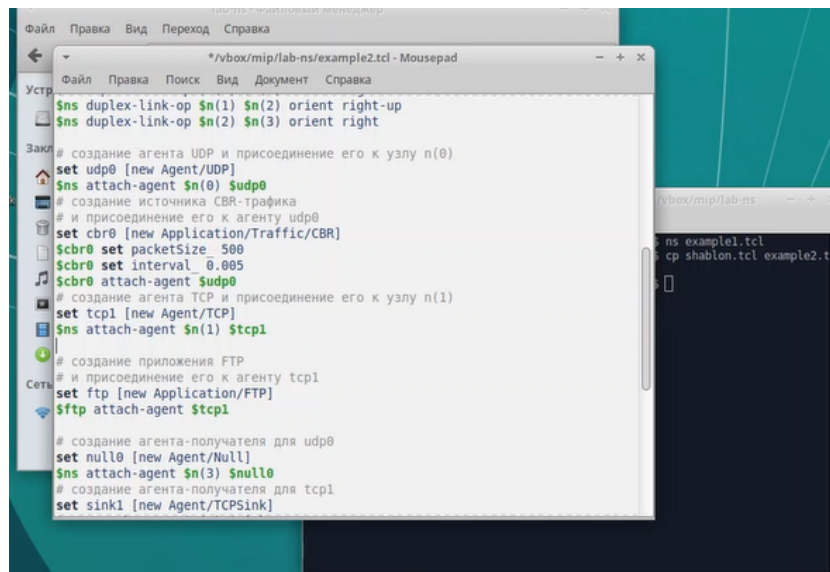


Рис. 8: Редактирование example2.tcl

Запустил код программы example2.tcl. Просмотрел движение пакетов данных (рис. [-@fig:011])

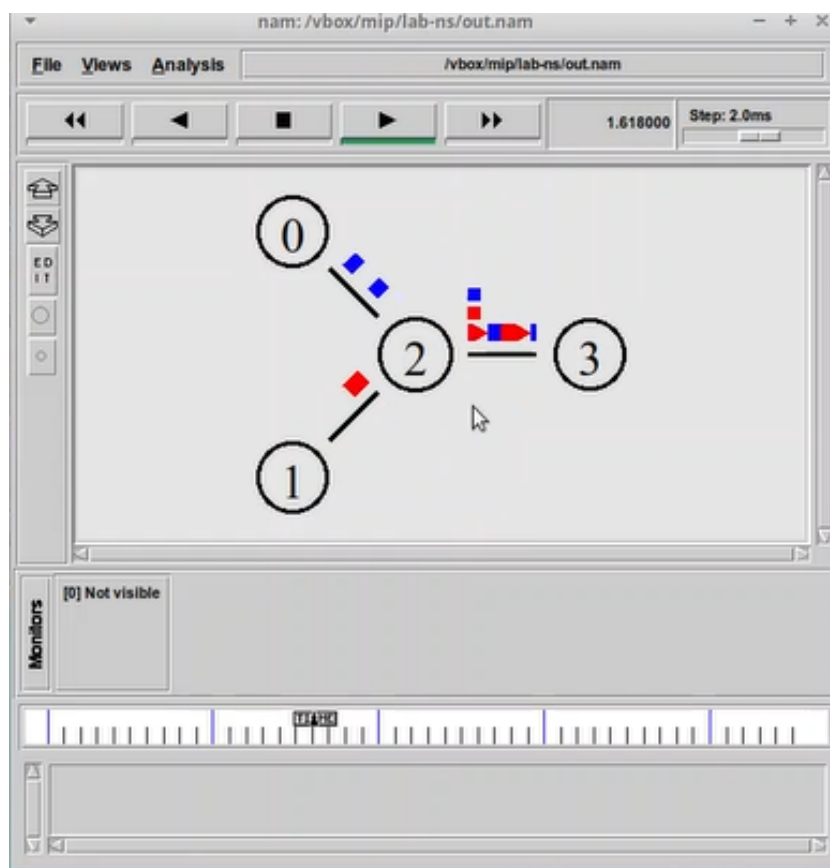


Рис. 9: Запуск программы

Создал новый файл example3.tcl. Написал первую часть программы (рис. [-@fig:012])

```

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

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"

```

Рис. 10: Редактирование example3.tcl

Запустил код программы example3.tcl. Просмотрел движение пакетов данных в случае разрыва соединения (рис. [-@fig:013])

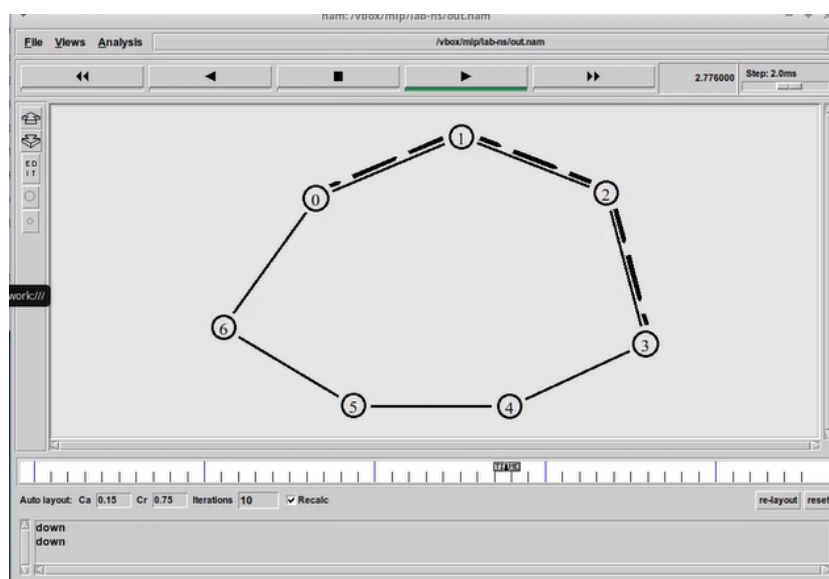
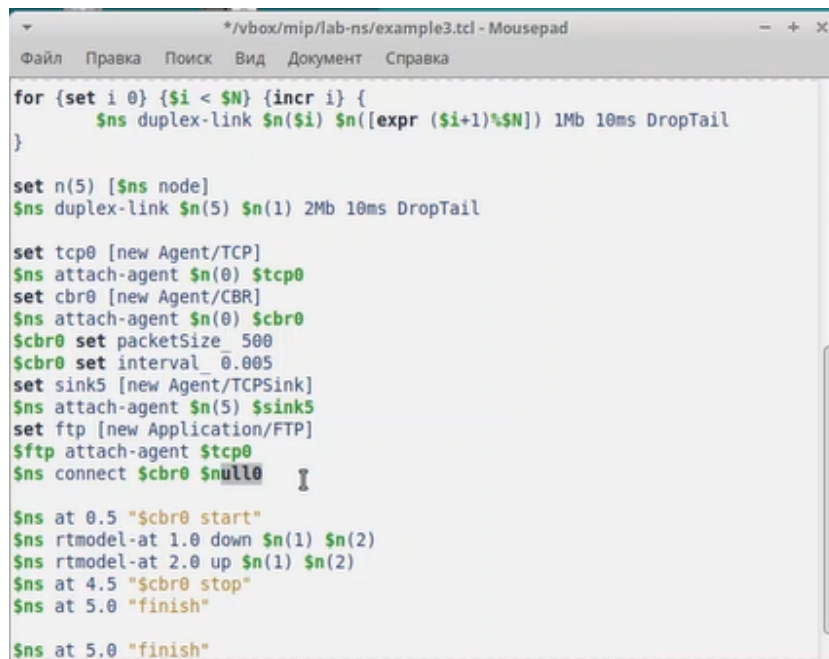


Рис. 11: Запуск программы

Написал код для программы из Упражнения (рис. [-@fig:015])

A screenshot of a text editor window titled "\*vbox/mip/lab-ns/example3.tcl - Mousepad". The window has a menu bar with "Файл", "Правка", "Поиск", "Вид", "Документ", and "Справка". The script content is as follows:

```
for {set i 0} {$i < $N} {incr i} {  
    $ns duplex-link $n($i) $n([expr ($i+1)%$N]) 1Mb 10ms DropTail  
}  
  
set n(5) [$ns node]  
$ns duplex-link $n(5) $n(1) 2Mb 10ms DropTail  
  
set tcp0 [new Agent/TCP]  
$ns attach-agent $n(0) $tcp0  
set cbr0 [new Agent/CBR]  
$ns attach-agent $n(0) $cbr0  
$cbr0 set packetSize 500  
$cbr0 set interval 0.005  
set sink5 [new Agent/TCPSink]  
$ns attach-agent $n(5) $sink5  
set ftp [new Application/FTP]  
$ftp attach-agent $tcp0  
$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. 12: Код программы из Упражнения

Запустил программу. Вначале пакеты идут по кратчайшему пути. (рис. [-@fig:016])

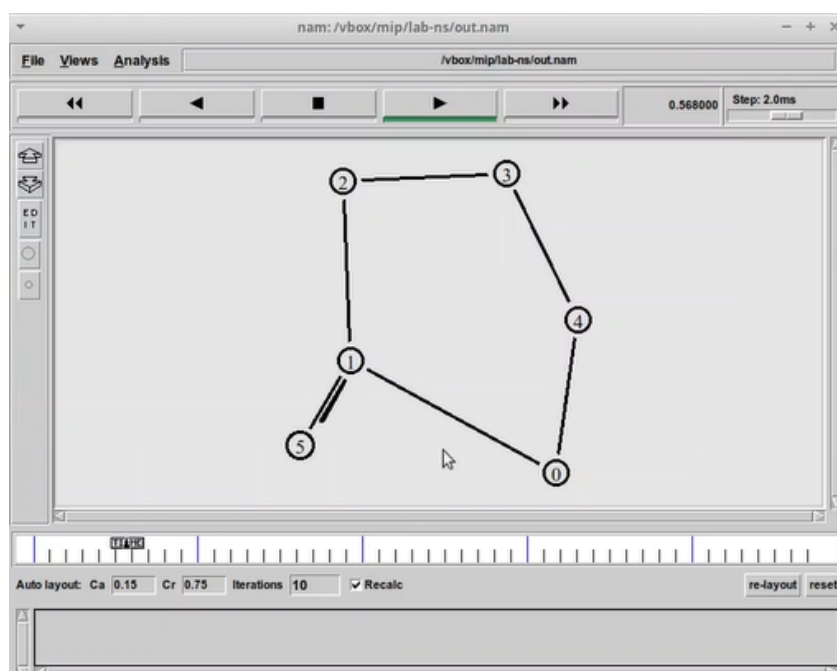


Рис. 13: Движение пакетов по кратчайшему пути

Движение пакетов в случае разрыва соединения. (рис. [-@fig:017])

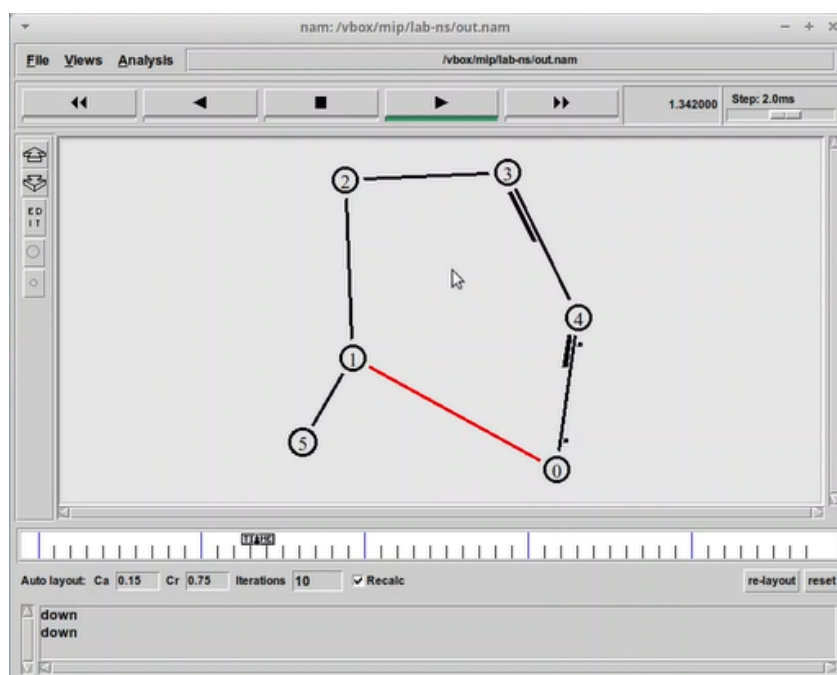


Рис. 14: Движение пакетов в случае разрыва соединения



Пакеты снова идут по кратчайшему пути (рис. [-@fig:018])

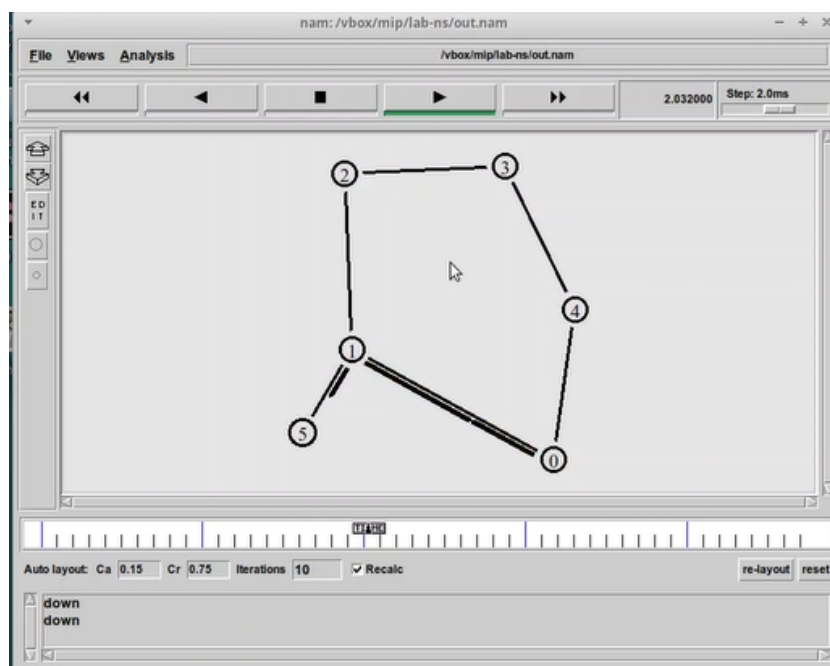


Рис. 15: Движение пакетов по кратчайшему пути

# Выводы

Приобрел навыки моделирования сетей передачи данных с помощью средств имитационного моделирования NS-2, а также анализа полученных результатов моделирования.