

# **Отчёт по лабораторной работе 1**

**Простые модели компьютерной сети**

Наталья Андреевна Сидорова

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>14</b>
	<b>Список литературы</b>	<b>15</b>

# Список иллюстраций

2.1	Код шаблона . . . . .	6
2.2	Код первой сети . . . . .	7
2.3	Первая модель . . . . .	8
2.4	Код второй сети . . . . .	9
2.5	Заполненная очередь . . . . .	9
2.6	Код третьей сети . . . . .	10
2.7	Обход пакетов . . . . .	11
2.8	Код упражнения . . . . .	12
2.9	Моя топология . . . . .	13

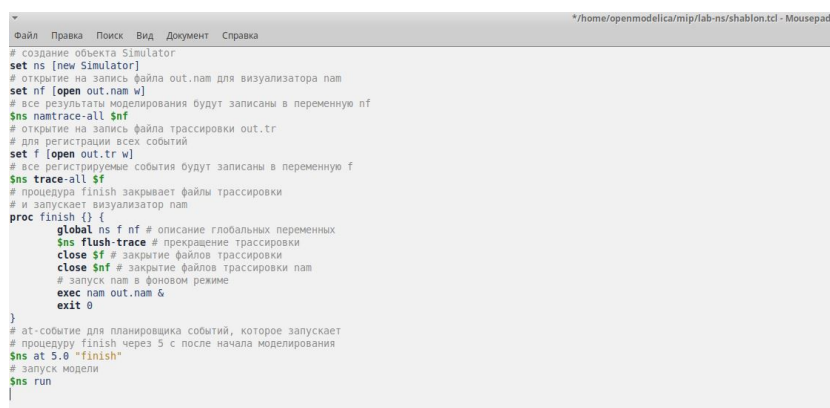
## **Список таблиц**

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

Я создала файл шаблона для построения нескольких моделей компьютерной сети (рис. 2.1).



```

# создание объекта Simulator
set ns [new Simulator]
# открытие на запись файла out.nam для визуализатора nam
set nf [open out.nam w]
# все результаты моделирования будут записаны в переменную nf
$ns namtrace-all $nf
# 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 # описание глобальных переменных
    $ns flush-trace # прекращение трассировки
    close $f # закрытие файлов трассировки
    close $nf # закрытие файлов трассировки лог
    # запуск nam в фоновом режиме
    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. 2.1: Код шаблона

Я создала сеть, состоящую из двух узлов и одного соединения (рис. 2.2).

```

# создание 2-х узлов:
set N 2
for {set i 0} {$i < $N} {incr i} {
    set n($i) [$ns node]
}
# соединение 2-х узлов дуплексным соединением
# с полосой пропускания 2 Мб/с и задержкой 10 мс,
# очередь с обслуживанием типа DropTail
$ns duplex-link $n(0) $n(1) 2Mb 10ms DropTail

# создание агента UDP и присоединение его к узлу n0
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
# создание источника трафика CBR (constant bit rate)
set cbr0 [new Application/Traffic/CBR]
# устанавливаем размер пакета в 500 байт
$cbr0 set packetSize_ 500
# задаем интервал между пакетами равным 0.005 секунды,
# т.е. 200 пакетов в секунду
$cbr0 set interval_ 0.005
# присоединение источника трафика CBR к агенту udp0
$cbr0 attach-agent $udp0

# Создание агента-приёмника и присоединение его к узлу n(1)
set null0 [new Agent/Null]
$ns attach-agent $n(1) $null0
# Соединение агентов между собой
$ns connect $udp0 $null0
# запуск приложения через 0,5 с
$ns at 0.5 "$cbr0 start"
# остановка приложения через 4,5 с
$ns at 4.5 "$cbr0 stop"

```

Рис. 2.2: Код первой сети

Запустила в программе данную модель (рис. 2.3).

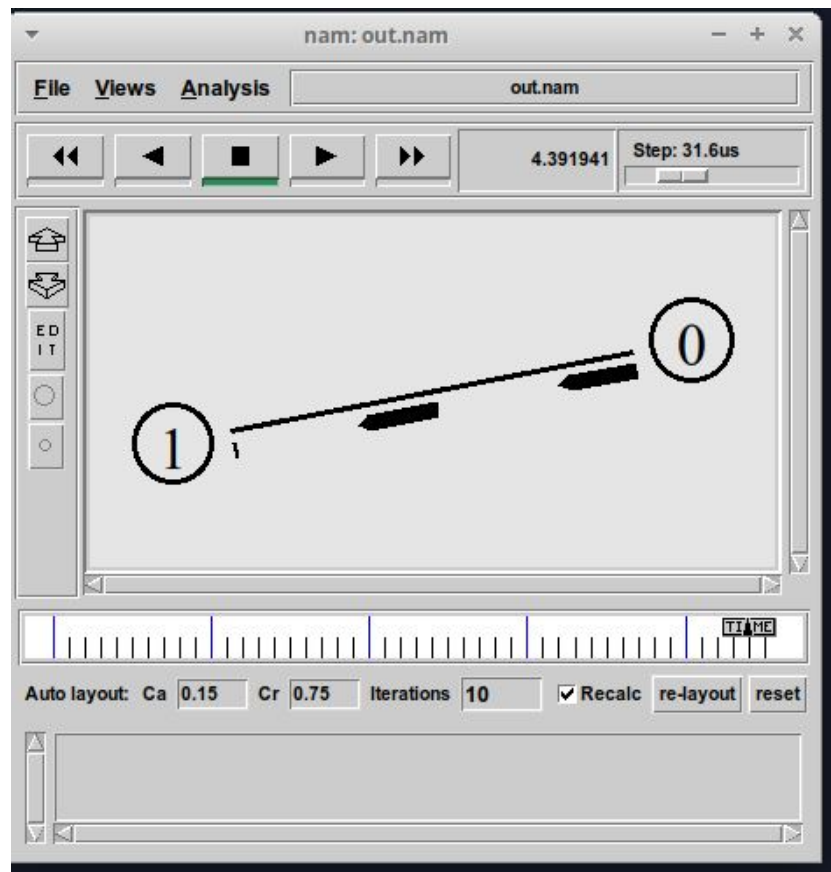


Рис. 2.3: Первая модель

Создала усложненную топологию сети с разветвлением. Из за него в приемнике возникает очередь, когда она переполняется то пакеты сбрасываются. (рис. 2.4).



```

* /home/openmodelica/mip/lab-ns/example2.tcl - Mousepad
Файл Правка Поиск Вид Документ Справка
tcl {set i 0; while { $i < 300 } {
  set n($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

# создание агента UDP и присоединение его к узлу n(0)
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
# создание источника CBR-трафика
# и присоединение его к агенту udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize 500
$cbr0 set interval 0.005
$cbr0 attach-agent $udp0
# создание агента TCP и присоединение его к узлу n(1)
set tcp1 [new Agent/TCP]
$ns attach-agent $n(1) $tcp1
# создание приложения FTP
# и присоединение его к агенту tcp1
set ftp [new Application/FTP]
$ftp attach-agent $tcp1
# создание агента-получателя для udp0
set null0 [new Agent/Null]
$ns attach-agent $n(3) $null0
# создание агента-получателя для tcp1
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"

# ат-событие для планировщика событий, которое запускает
# процедуру finish через 5 с после начала моделирования
$ns at 5.0 "finish"
# запуск модели
$ns run
}

```

Рис. 2.4: Код второй сети

Запустила и увидела переполнение очереди (рис. 2.5).

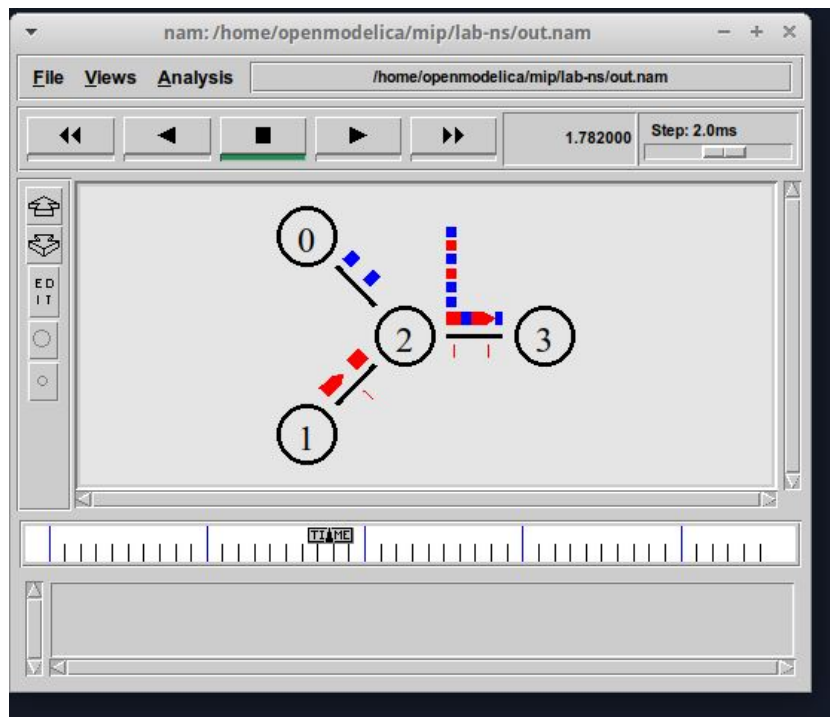


Рис. 2.5: Заполненная очередь

Создала кольцевую топологию сети, в ней пакеты идут по кратчайшему пути, если соединение разрывается то пакеты идут по более длинному пути. (рис. 2.6).

```

Файл  Правка  Поиск  Вид  Документ  Справка  /home/openmodelica/mip/lab-ns/example3.tcl - Mousepad
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 {} {
    # описание глобальных переменных
    global ns f nf
    # прекращение трассировки
    $ns flush-trace
    # закрытие файлов трассировки
    close $f
    # закрытие файлов трассировки nam
    close $nf
    # запуск nam в фоновом режиме
    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
$ns set packetSize 500
$ns 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"

# 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. 2.6: Код третьей сети

Запустила и увидела что при разрыве связи пакеты действительно идет по длинному маршруту (рис. 2.7).

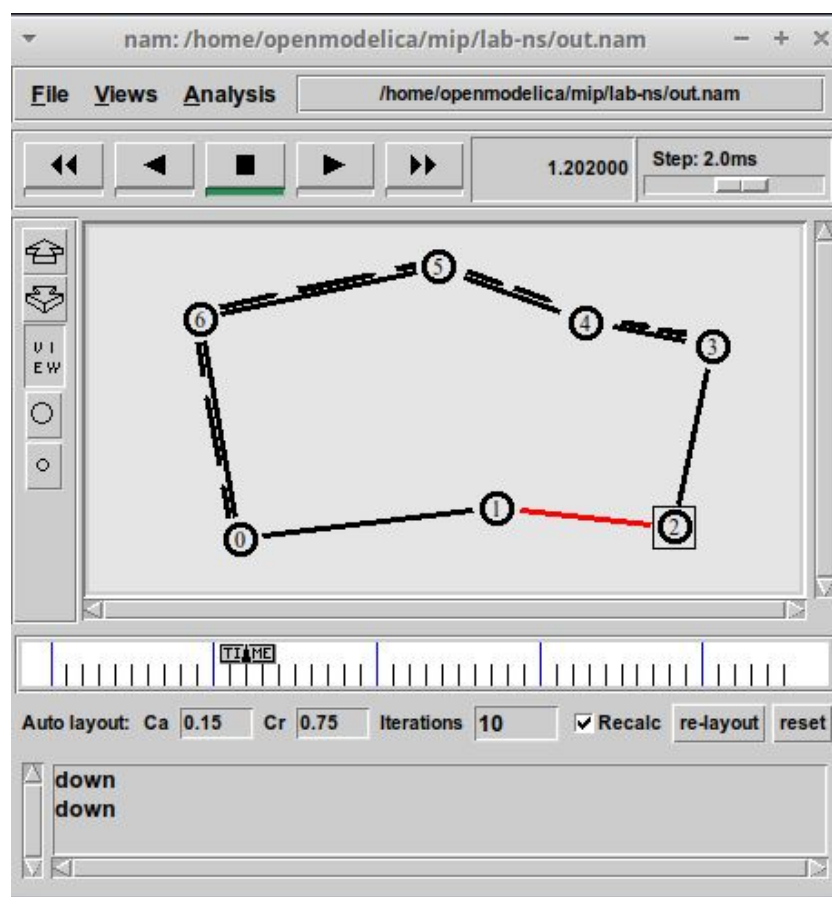


Рис. 2.7: Обход пакетов

Сделала упражнение с новыми требованиями (рис. 2.8).

```
Файл Правка Поиск Вид Документ Справка /home/openmodelica/mip/lab-ns/example64.tcl - Mousepad
# все результаты моделирования будут записаны в переменную nf
$ns namtrace-all $nf
# 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 {} {
    # описание глобальных переменных
    global ns f nf
    # прекращение трассировки
    $ns flush-trace
    # закрытие файлов трассировки
    close $f
    # закрытие файлов трассировки nam
    close $nf
    # запуск nam в фоновом режиме
    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 "sftp 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 "sftp stop"
$ns at 5.0 "finish"
```

Рис. 2.8: Код упражнения

Получившаяся топология (рис. 2.9).

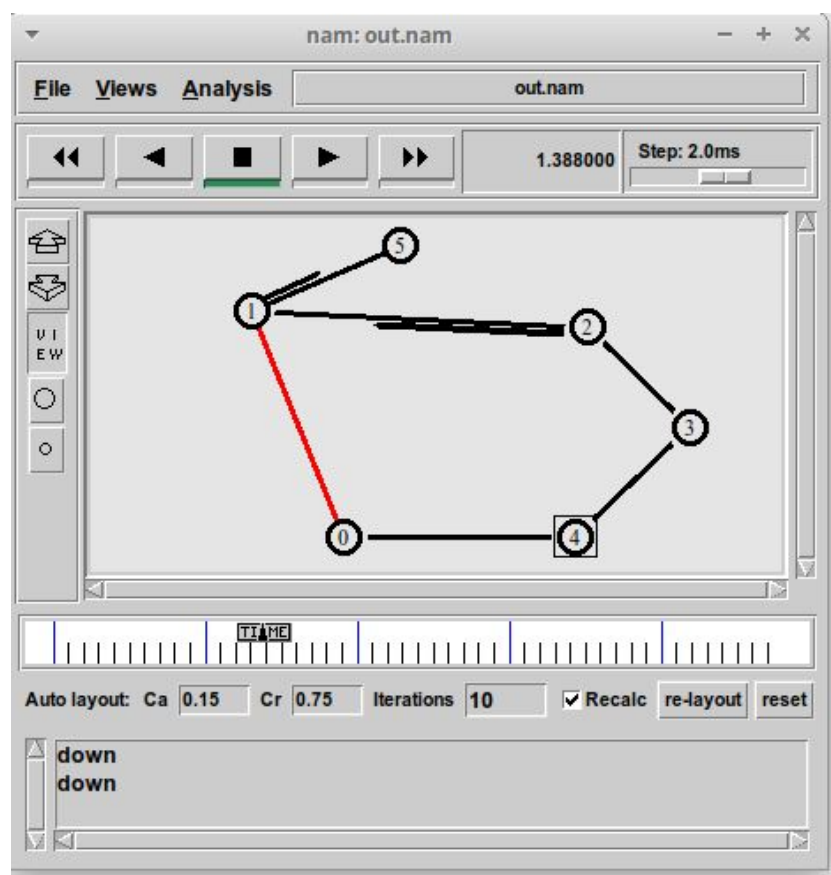


Рис. 2.9: Моя топология

## 3 Выводы

При разветвленной топологии в узле приема могут образовываться очереди, которые сбрасываются при переполнении. При замкнутой топологии пакеты идут по кратчайшему маршруту, но если соединение прервалось то пакеты пойдут по запасному пути.

## **Список литературы**