

# **Отчёт по лабораторной работе**

**Лабораторная №2 по имитационному моделированию**

**Дзахмишев Камбулат Заурович**

# 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>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
<b>6</b>	<b>Выполнение лабораторной работы</b>	<b>10</b>
<b>7</b>	<b>Выполнение лабораторной работы</b>	<b>11</b>
<b>8</b>	<b>Выполнение лабораторной работы</b>	<b>12</b>
<b>9</b>	<b>Выполнение лабораторной работы</b>	<b>13</b>
<b>10</b>	<b>Выполнение лабораторной работы</b>	<b>14</b>
<b>11</b>	<b>Выполнение лабораторной работы</b>	<b>15</b>
<b>12</b>	<b>Выводы</b>	<b>16</b>
	<b>Список литературы</b>	<b>17</b>

# Список иллюстраций

3.1	Создание модели по приведённому коду. . . . .	7
4.1	Создание модели по приведённому коду. . . . .	8
5.1	Создание модели по приведённому коду. . . . .	9
6.1	Создание модели по приведённому коду. . . . .	10
7.1	Создание модели по приведённому коду. . . . .	11
9.1	График агента Reno . . . . .	13
10.1	График агента NewReno . . . . .	14
11.1	График агента Vegas . . . . .	15

## **Список таблиц**

# 1 Цель работы

Описание моделируемой сети: – сеть состоит из 6 узлов; – между всеми узлами установлено дуплексное соединение с различными пропускной способностью и задержкой 10 мс (см. рис. 2.4); – узел r1 использует очередь с дисциплиной RED для накопления пакетов, максимальный размер которой составляет 25; – TCP-источники на узлах s1 и s2 подключаются к TCP-приёмнику на узле s3; – генераторы трафика FTP прикреплены к TCP-агентам.

## **2 Задание**

Требуется разработать сценарий, реализующий модель согласно рис. 2.4, построить в Xgraph график изменения ТСП-окна, график изменения длины очереди и средней длины очереди

### 3 Выполнение лабораторной работы



```
# 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:
proc finish {} {
    global tchan_
    set awkCode {
        {
            if ($1 == "Q" && NF>2) {
                print $2, $3 >> "temp.q";
                set end $2
            }
            else if ($1 == "a" && NF>2)
                print $2, $3 >> "temp.a";
        }
    }
    set f [open temp.queue w]
    puts $f "TitleText: Reno"
    puts $f "Device: Postscript"
    puts $f "0.Color: Blue"
    puts $f "1.Color: Green"
    if { [info exists tchan_] } {
        close $tchan_
    }
    exec rm -f temp.q temp.a
    exec touch temp.a temp.q
    exec awk $awkCode all.q
```

Рис. 3.1: Создание модели по приведённому коду.

## 4 Выполнение лабораторной работы

```
exec awk $awkCode all.q
puts $f "\"queue_Reno
exec cat temp.q >@ $f
puts $f \"\\n\\\"ave_queue_Reno
exec cat temp.a >@ $f
close $f

exec xgraph -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeReno &
exec xgraph -bb -tk -x time -y queue temp.queue &
exit 0
}
# Формирование файла с данными о размере окна TCP:
proc plotWindow {tcpSource file} {
global ns
set time 0.01
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file"
}
# Узлы сети:
set N 5
for {set i 1} {$i < $N} {incr i} {
set node_(s$i) [$ns node]
}
set node_(r1) [$ns node]
set node_(r2) [$ns node]

# Соединения:
$ns duplex-link $node_(s1) $node_(r1) 10Mb 2ms DropTail
$ns duplex-link $node_(s2) $node_(r1) 10Mb 3ms DropTail
$ns duplex-link $node_(r1) $node_(r2) 1.5Mb 20ms RED
$ns queue-limit $node_(r1) $node_(r2) 25
$ns queue-limit $node_(r2) $node_(r1) 25
$ns duplex-link $node_(s3) $node_(r2) 10Mb 4ms DropTail
$ns duplex-link $node_(s4) $node_(r2) 10Mb 5ms DropTail
# Агенты и приложения:
set tcp1 [$ns create-connection TCP/Reno $node_(s1) TCPSink $node_(r2) 0.1
```

Рис. 4.1: Создание модели по приведённому коду.



## 5 Выполнение лабораторной работы

```
# Соединения:
$ns duplex-link $node_(s1) $node_(r1) 10Mb 2ms DropTail
$ns duplex-link $node_(s2) $node_(r1) 10Mb 3ms DropTail
$ns duplex-link $node_(r1) $node_(r2) 1.5Mb 20ms RED
$ns queue-limit $node_(r1) $node_(r2) 25
$ns queue-limit $node_(r2) $node_(r1) 25
$ns duplex-link $node_(s3) $node_(r2) 10Mb 4ms DropTail
$ns duplex-link $node_(s4) $node_(r2) 10Mb 5ms DropTail

# Агенты и приложения:
set tcp1 [$ns create-connection TCP/Reno $node_(s1) TCPSink $node_(s3) 0]
$tcp1 set window_ 15
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]
$tcp2 set window_ 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]

# Мониторинг размера окна TCP:
set windowVsTime [open WindowVsTimeReno w]
puts $windowVsTime "0.Color: White"
puts $windowVsTime "\"Window_size\""
set qmon [$ns monitor-queue $node_(r1) $node_(r2) [open qm.out w] 0.1];
[$ns link $node_(r1) $node_(r2)] queue-sample-timeout;

# Мониторинг очереди:
set redq [[$ns link $node_(r1) $node_(r2)] queue]
set tchan_ [open all.q w]
$redq trace curq_
$redq trace ave_
$redq attach $tchan_

# Добавление at-событий:
$ns at 0.0 "$ftp1 start"
$ns at 1.1 "plotWindow $tcp1 $windowVsTime"
$ns at 3.0 "$ftp2 start"
$ns at 10 "finish"

# запуск модели
$ns run
```

Рис. 5.1: Создание модели по приведённому коду.

## 6 Выполнение лабораторной работы

```
/home/openmodelica/mip/lab-ns/exapmle_lab2.2.tcl - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка
puts at [expr $now+$time] "plotWindow $tcpSource $file"
}
# Узлы сети:
set N 5
for {set i 1} {$i < $N} {incr i} {
  set node_($i) [$ns node]
}
set node_(r1) [$ns node]
set node_(r2) [$ns node]

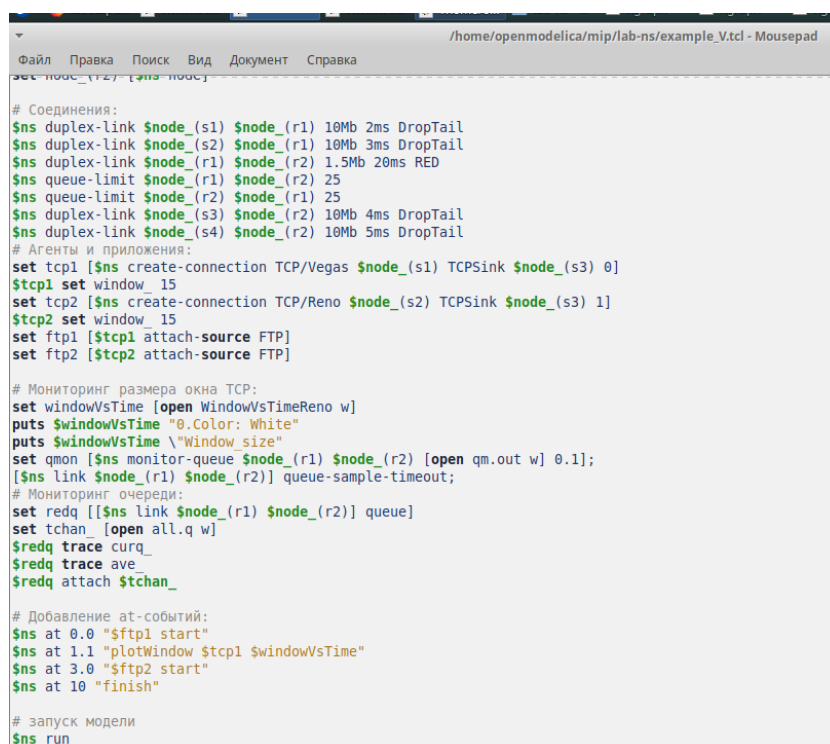
# Соединения:
$ns duplex-link $node_(s1) $node_(r1) 10Mb 2ms DropTail
$ns duplex-link $node_(s2) $node_(r1) 10Mb 3ms DropTail
$ns duplex-link $node_(r1) $node_(r2) 1.5Mb 20ms RED
$ns queue-limit $node_(r1) $node_(r2) 25
$ns queue-limit $node_(r2) $node_(r1) 25
$ns duplex-link $node_(s3) $node_(r2) 10Mb 4ms DropTail
$ns duplex-link $node_(s4) $node_(r2) 10Mb 5ms DropTail

# Агенты и приложения:
set tcp1 [$ns create-connection TCP/Newreno $node_(s1) TCPSink $node_(s3) 0]
$tcp1 set window 15
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]
$tcp2 set window 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]

# Мониторинг размера окна TCP:
set windowVsTime [open WindowVsTimeReno w]
puts $windowVsTime "0.Color: White"
puts $windowVsTime "\nWindow size"
set qmon [$ns monitor-queue $node_(r1) $node_(r2) [open qm.out w] 0.1];
[$ns link $node_(r1) $node_(r2)] queue-sample-timeout;
# Мониторинг очереди:
set redq [$ns link $node_(r1) $node_(r2)] queue]
set tchan [open all.q w]
$redq trace curq_
$redq trace ave_
$redq attach $tchan_
```

Рис. 6.1: Создание модели по приведённому коду.

## 7 Выполнение лабораторной работы



The image shows a screenshot of a text editor window titled "/home/openmodelica/mip/lab-ns/example\_V.tcl - Mousepad". The editor contains a NetSim configuration script. The script is divided into several sections: connections, agents and applications, TCP window size monitoring, queue monitoring, and event scheduling. The script uses various NetSim commands like `duplex-link`, `queue-limit`, `set`, `link`, `monitor-queue`, `link`, `redq`, `trace`, and `attach` to configure a network simulation. The script ends with a `run` command to execute the model.

```
# Соединения:
$ns duplex-link $node_(s1) $node_(r1) 10Mb 2ms DropTail
$ns duplex-link $node_(s2) $node_(r1) 10Mb 3ms DropTail
$ns duplex-link $node_(r1) $node_(r2) 1.5Mb 20ms RED
$ns queue-limit $node_(r1) $node_(r2) 25
$ns queue-limit $node_(r2) $node_(r1) 25
$ns duplex-link $node_(s3) $node_(r2) 10Mb 4ms DropTail
$ns duplex-link $node_(s4) $node_(r2) 10Mb 5ms DropTail

# Агенты и приложения:
set tcp1 [$ns create-connection TCP/Vegas $node_(s1) TCPSink $node_(s3) 0]
$tcp1 set window_ 15
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]
$tcp2 set window_ 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]

# Мониторинг размера окна TCP:
set windowVsTime [open WindowVsTimeReno w]
puts $windowVsTime "0.Color: White"
puts $windowVsTime \ "Window size"
set qmon [$ns monitor-queue $node_(r1) $node_(r2) [open qm.out w] 0.1];
[$ns link $node_(r1) $node_(r2)] queue-sample-timeout;

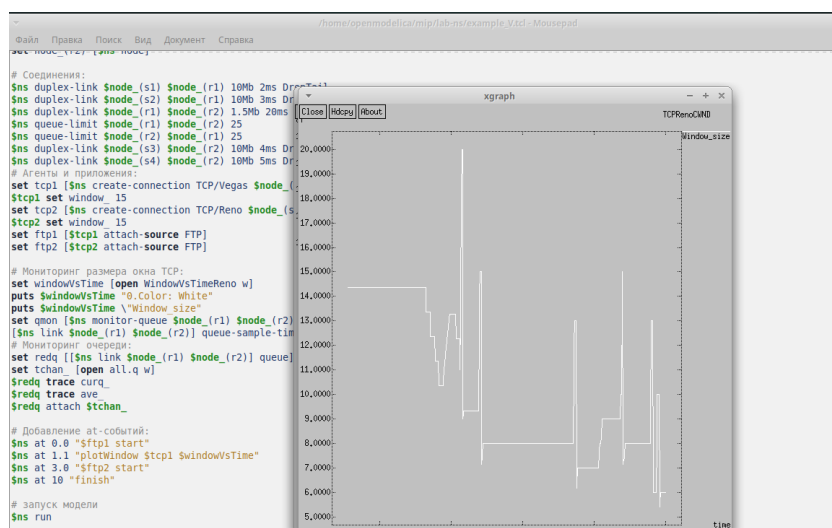
# Мониторинг очереди:
set redq [$ns link $node_(r1) $node_(r2)] queue]
set tchan_ [open all.q w]
$redq trace curq_
$redq trace ave_
$redq attach $tchan_

# Добавление at-событий:
$ns at 0.0 "$ftp1 start"
$ns at 1.1 "plotWindow $tcp1 $windowVsTime"
$ns at 3.0 "$ftp2 start"
$ns at 10 "finish"

# запуск модели
$ns run
```

Рис. 7.1: Создание модели по приведённому коду.

## 8 Выполнение лабораторной работы



## 9 Выполнение лабораторной работы

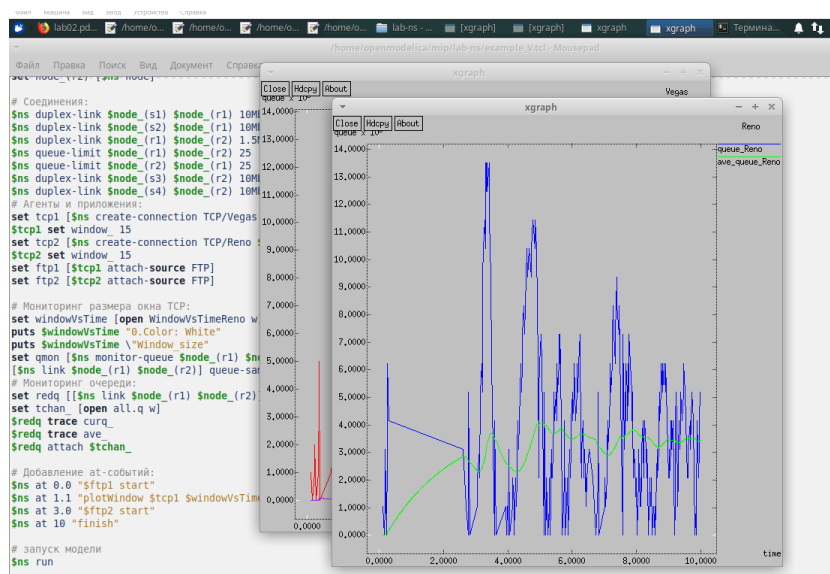


Рис. 9.1: График агента Reno

## 10 Выполнение лабораторной работы

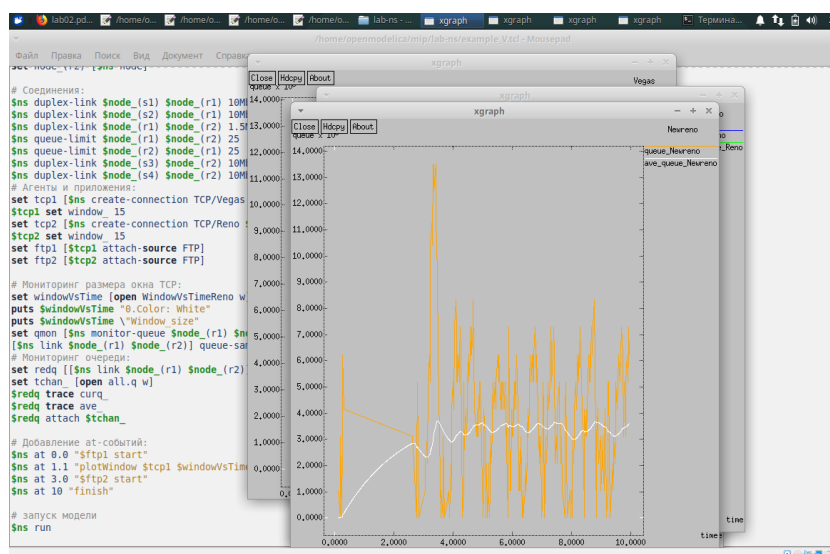


Рис. 10.1: График агента NewReno

# 11 Выполнение лабораторной работы

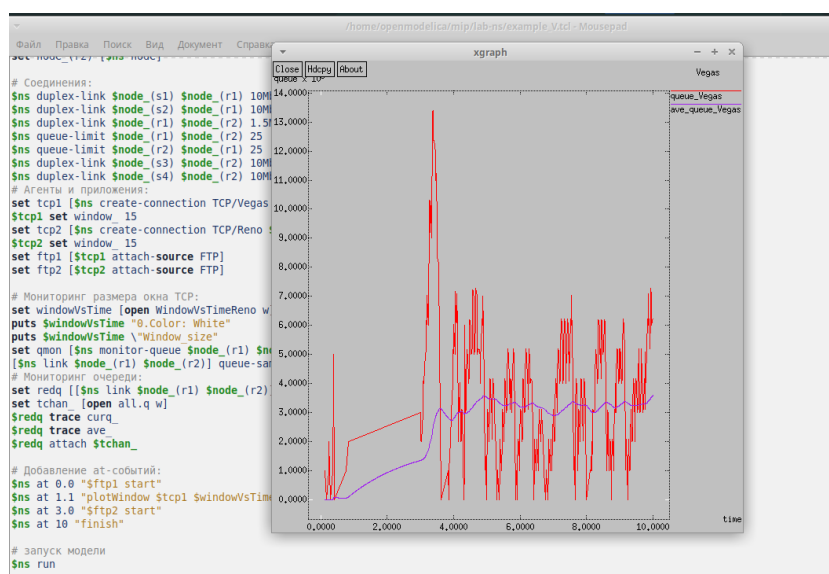


Рис. 11.1: График агента Vegas

## **12 Выводы**

**В ходе данной лабораторной работы составил графики трафика разных агентов и провел сравнительный анализ каждого из них, а также поменял цвета отображаемых показателей и заголовки.**



## **Список литературы**