

Отчёт по лабораторной работе

Лабораторная №3 по имитационному моделированию

Дзахмишев Камбулат Заурович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выполнение лабораторной работы	8
5	Выполнение лабораторной работы	9
6	Выполнение лабораторной работы	10
7	Выполнение лабораторной работы	11
8	Выполнение лабораторной работы	12
9	Выполнение лабораторной работы	13
10	Выполнение лабораторной работы	14
11	Выводы	15
	Список литературы	16

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	Теор. вероятность потери и средняя длина очереди	10
7.1	Список	11
8.1	graph_plot	12
9.1	Файл qm появился	13
10.1	Содержимое графика qm	14

Список таблиц

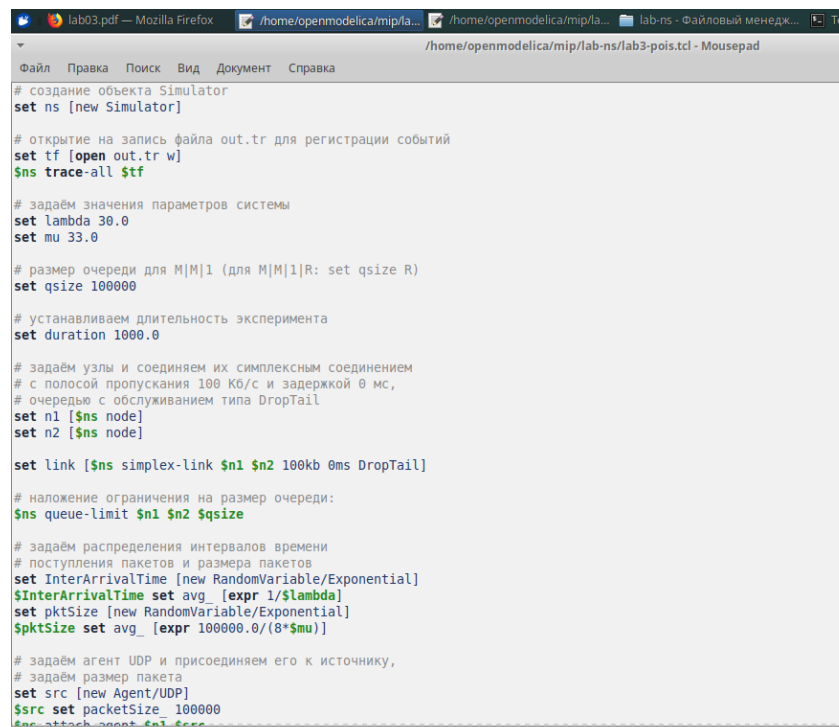
1 Цель работы

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.tr для регистрации событий
set tf [open out.tr w]
$ns trace-all $tf

# задаём значения параметров системы
set lambda 30.0
set mu 33.0

# размер очереди для M|M|1 (для M|M|1|R: set qsize R)
set qsize 100000

# устанавливаем длительность эксперимента
set duration 1000.0

# задаём узлы и соединяем их симплексным соединением
# с полосой пропускания 100 Кб/с и задержкой 0 мс,
# очередь с обслуживанием типа DropTail
set n1 [$ns node]
set n2 [$ns node]

set link [$ns simplex-link $n1 $n2 100kb 0ms DropTail]

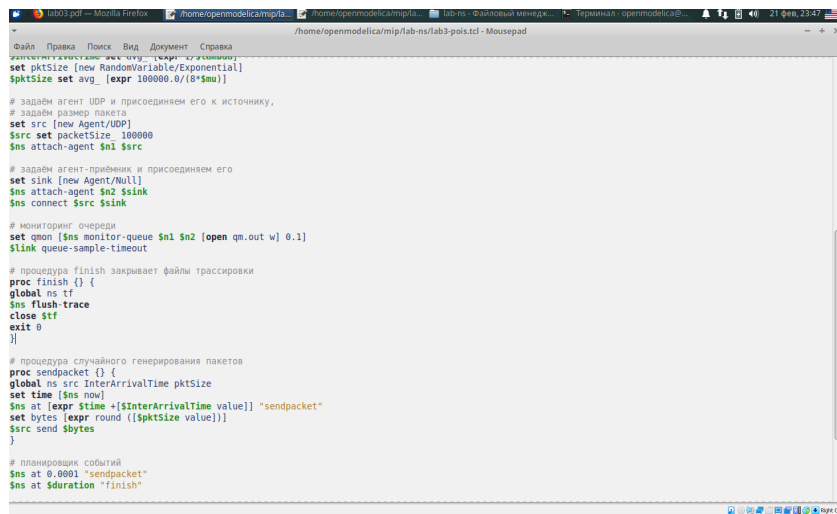
# наложение ограничения на размер очереди:
$ns queue-limit $n1 $n2 $qsize

# задаём распределения интервалов времени
# поступления пакетов и размера пакетов
set InterArrivalTime [new RandomVariable/Exponential]
$InterArrivalTime set avg_ [expr 1/$lambda]
set pktSize [new RandomVariable/Exponential]
$pktSize set avg_ [expr 100000.0/(8*$mu)]

# задаём агент UDP и присоединяем его к источнику,
# задаём размер пакета
set src [new Agent/UDP]
$src set packetSize 100000
$src attach-agent $n1 $src
```

Рис. 3.1: Создание модели по приведённому коду.

4 Выполнение лабораторной работы



```
set pktSize [new RandomVariable/Exponential]
$pktSize set avg_ [expr 100000.0/(8*$mu)]

# задаём агент UDP и присоединяем его к источнику,
# задаём размер пакета
set src [new Agent/UDP]
$src set packetSize 100000
$ns attach-agent $n1 $src

# задаём агент-приёмник и присоединяем его
set sink [new Agent/Null]
$ns attach-agent $n2 $sink
$ns connect $src $sink

# мониторинг очереди
set qmon [$ns monitor-queue $n1 $n2 [open qm.out w] 0.1]
$link queue-sample-timeout

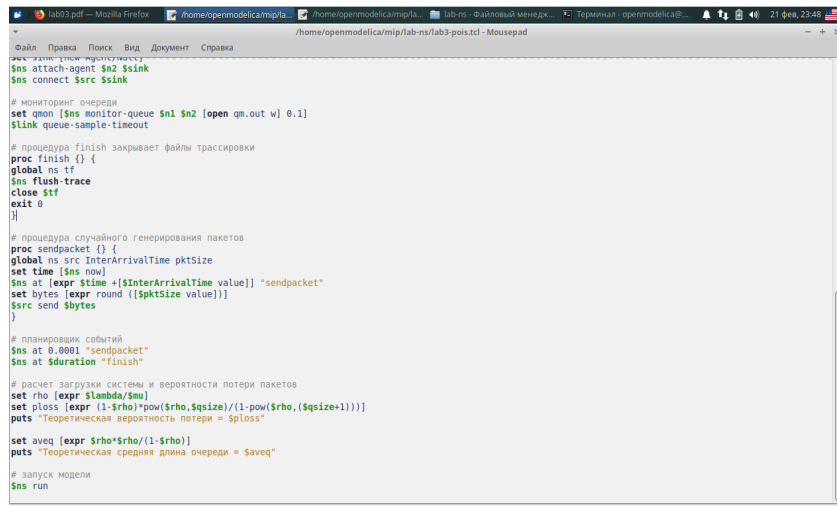
# процедура finish закрывает файлы трассировки
proc finish {} {
    global ns tf
    $ns flush-trace
    close $tf
    exit 0
}

# процедура случайного генерирования пакетов
proc sendpacket {} {
    global ns src InterArrivalTime pktSize
    set time [$ns now]
    $ns at [expr $time + [$InterArrivalTime value]] "sendpacket"
    set bytes [expr round ([$pktSize value])]
    $src send $bytes
}

# планирование событий
$ns at 0.0001 "sendpacket"
$ns at $duration "finish"
```

Рис. 4.1: Создание модели по приведённому коду.

5 Выполнение лабораторной работы



```
#!/usr/bin/env ns3
# -*- coding: utf-8 -*-
# Создание модели по приведённому коду

# attach-agent
$ns attach-agent $n2 $sink
$ns connect $src $sink

# мониторинг очереди
set qmon [$ns monitor-queue $n1 $n2 [open qm.out w] 0.1]
$link queue-sample-timeout

# процедура finish закрывает файлы трассировки
proc finish {} {
    global ns tf
    $ns flush-trace
    close $tf
    exit 0
}

# процедура случайного генерирования пакетов
proc sendpacket {} {
    global ns src InterArrivalTime pktSize
    set time [$ns now]
    $ns at [expr $time + [$InterArrivalTime value]] "sendpacket"
    set bytes [expr round ([[$pktSize value]])]
    $src send $bytes
}

# планировщик событий
$ns at 0.0001 "sendpacket"
$ns at $duration "finish"

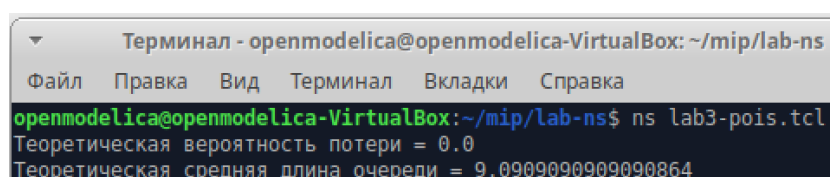
# расчет загрузки системы и вероятности потери пакетов
set rho [expr $lambda/$mu]
set ploss [expr (1-$rho)^pow($rho,$qsize)/(1-pow($rho,($qsize+1)))]
puts "Теоретическая вероятность потери = $ploss"

set aveq [expr $rho*$rho/(1-$rho)]
puts "Теоретическая средняя длина очереди = $aveq"

# запуск модели
$ns run
```

Рис. 5.1: Создание модели по приведённому коду.

6 Выполнение лабораторной работы



```
Терминал - openmodelica@openmodelica-VirtualBox: ~/mip/lab-ns
Файл  Правка  Вид  Терминал  Вкладки  Справка
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ns lab3-pois.tcl
Теоретическая вероятность потери = 0.0
Теоретическая средняя длина очереди = 9.0909090909090864
```

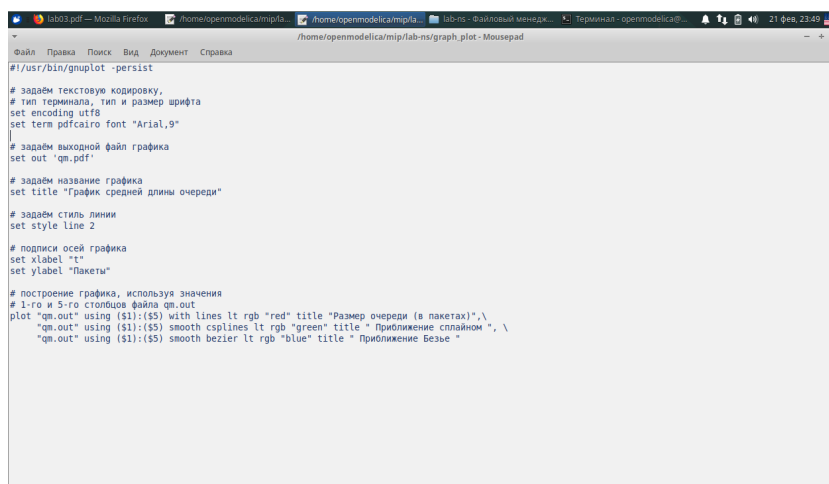
Рис. 6.1: Теор. вероятность потери и средняя длина очереди

7 Выполнение лабораторной работы

```
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ touch graph_plot
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ls
all.q          example_lab2.tcl  lab3-pois.tcl  shablon.tcl  WindowVsTimeReno
example1.tcl   example_V.tcl    out.nam        temp.a
example2.tcl   exapmle_lab2.2.tcl out.tr         temp.q
example3.tcl   graph plot       qm.out         temp.queue
```

Рис. 7.1: Список

8 Выполнение лабораторной работы



```
#!/usr/bin/gnuplot -persist
# задаём текстовую кодировку,
# тип терминала, тип и размер шрифта
set encoding utf8
set term pdfcairo font "Arial,9"
# задаём выходной файл графика
set out 'qm.pdf'
# задаём название графика
set title "График средней длины очереди"
# задаём стиль линии
set style line 2
# подписи осей графика
set xlabel "t"
set ylabel "Пакеты"
# построение графика, используя значения
# 1-го и 5-го столбцов файла qm.out
plot "qm.out" using ($1):($5) with lines lt rgb "red" title "Размер очереди (в пакетах)", \
      "qm.out" using ($1):($5) smooth csplines lt rgb "green" title "Приближение сплайном ", \
      "qm.out" using ($1):($5) smooth bezier lt rgb "blue" title "Приближение Безье "
```

Рис. 8.1: graph_plot

9 Выполнение лабораторной работы

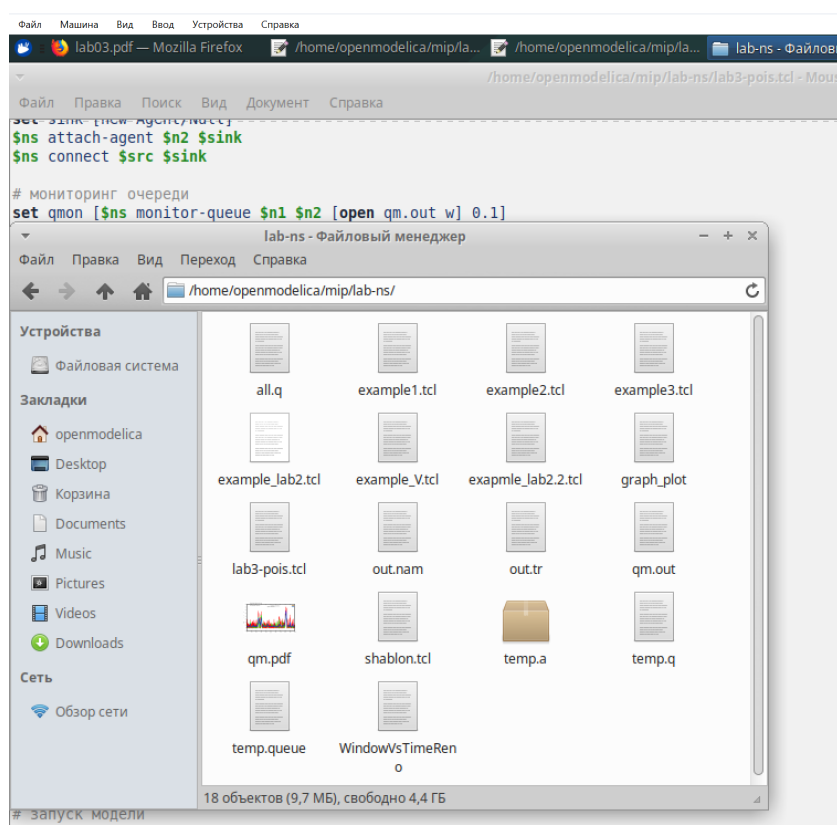


Рис. 9.1: Файл qm появился

10 Выполнение лабораторной работы

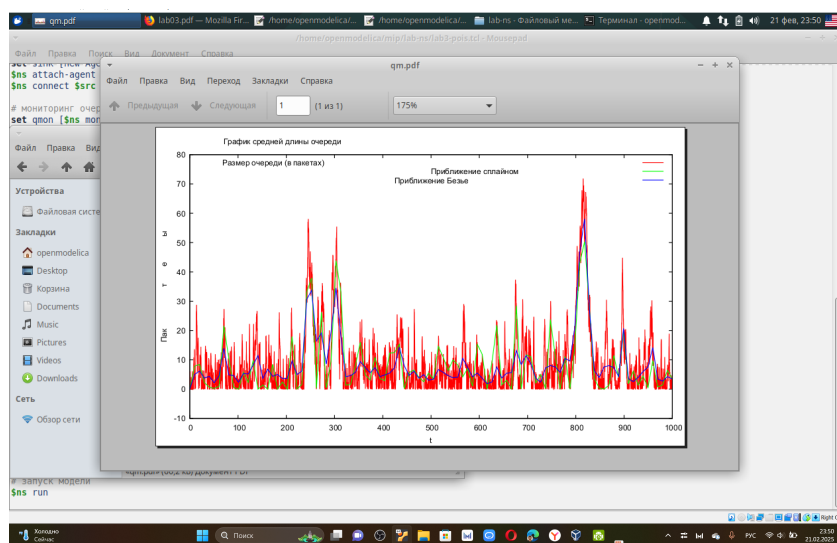


Рис. 10.1: Содержимое графика qtm

11 Выводы

В ходе данной лабораторной работы составил график задачи по вычислению средней длины очереди.

Список литературы