

Лабораторная работа 3. Моделирование стохастических процессов

Ильин Никита Евгеньевич

Содержание

1	Цель работы	5
2	Ход работы	6
3	Выводы	11
4	Библиография	12

List of Figures

2.1	Рис 1. Код программы(1)	7
2.2	Рис 2. Код программы(2)	8
2.3	Рис 3. Результат работы программы	8
2.4	Рис 4. Код программы GNUplot	9
2.5	Рис 5. График №1 - Результат выполнения программы	10

List of Tables

1 Цель работы

Научиться разрабатывать сценарий, реализующий модель стохастических процессов

2 Ход работы

1. Реализую модель на NS-2, с помощью кода, представленного в задании

```

1 # создание объекта Simulator
2 set ns [new Simulator]
3 # открытие на запись файла out.tr для регистрации событий
4 set tf [open out.tr w]
5 $ns trace-all $tf
6 # задаём значения параметров системы
7 set lambda 30.0
8 set mu 33.0
9 # размер очереди для M/M/1 (для M/M/1/R: set qsize R)
10 set qsize 100000
11 # устанавливаем длительность эксперимента
12 set duration 1000.0
13 # задаём узлы и соединяем их симплексным соединением
14 # с полосой пропускания 100 Кб/с и задержкой 0 мс,
15 # очередью с обслуживанием типа DropTail
16 set n1 [$ns node]
17 set n2 [$ns node]
18 set link [$ns simplex-link $n1 $n2 100kb 0ms DropTail]
19 # наложение ограничения на размер очереди:
20 $ns queue-limit $n1 $n2 $qsize
21 # задаём распределения интервалов времени
22 # поступления пакетов и размера пакетов
23 set InterArrivalTime [new RandomVariable/Exponential]
24 $InterArrivalTime set avg_ [expr 1/$lambda]
25 set pktSize [new RandomVariable/Exponential]
26 $pktSize set avg_ [expr 100000.0/(8*$mu)]
27 # задаём агент UDP и присоединяем его к источнику,
28 # задаём размер пакета
29 set src [new Agent/UDP]
30 $src set packetSize_ 100000
31 $ns attach-agent $n1 $src
32 # задаём агент-приёмник и присоединяем его
33 set sink [new Agent/Null]
34 $ns attach-agent $n2 $sink
35 $ns connect $src $sink
36 # мониторинг очереди
37 set qmon [$ns monitor-queue $n1 $n2 [open qm.out w] 0.1]
38 $link queue-sample-timeout
39 # процедура finish закрывает файлы трассировки
40 proc finish {} {
41     global ns tf
42     $ns flush-trace
43     close $tf
44     exit 0
45 }
46 # процедура случайного генерирования пакетов
47 proc sendpacket {} {
48     global ns src InterArrivalTime pktSize
49     set time [$ns now]
50     $ns at [expr $time + [$InterArrivalTime value]] "sendpacket"
51     set bytes [expr round ([$pktSize value])]
52     $src send $bytes
53 }
54 # планировщик событий

```

Figure 2.1: Рис 1. Код программы(1)

```

54 # планировщик событий
55 $ns at 0.0001 "sendpacket"
56 $ns at $duration "finish"
57 # расчет загрузки системы и вероятности потери пакетов
58 set rho [expr $lambda/$mu]
59 set ploss [expr (1-$rho)*pow($rho,$qsize)/(1-pow($rho,($qsize+1)))]
60 puts "Теоретическая вероятность потери = $ploss"
61 set aveq [expr $rho*$rho/(1-$rho)]
62 puts "Теоретическая средняя длина очереди = $aveq"
63 # запуск модели
64 $ns run

```

Figure 2.2: Рис 2. Код программы(2)

2. Запускаю файл:

```
ns lab03.tcl
```

```

openmodelica@nkt1lyn:~/mip/lab-ns/lab03$ ns lab03.tcl
Теоретическая вероятность потери = 0.0
Теоретическая средняя длина очереди = 9.0909090909090864

```

Figure 2.3: Рис 3. Результат работы программы

2. Создаю файл graph в каталоге с проектом. Добавляю код, предоставленный в задании, соблюдая синтаксис.

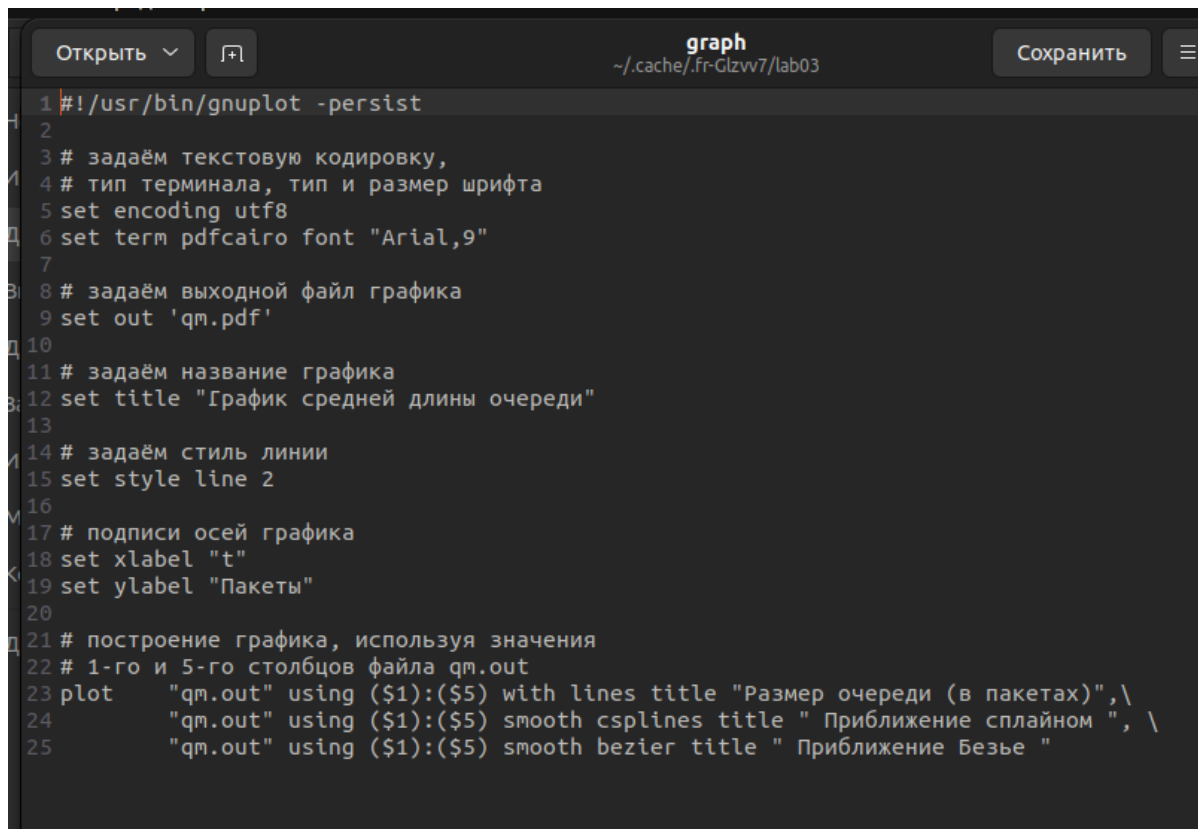
Создание файла:

```
touch graph
```

Делаю файл исполняемым:

```
chmod +x graph
```

Добавляю в файл код программы:



```
1#!/usr/bin/gnuplot -persist
2
3# задаём текстовую кодировку,
4# тип терминала, тип и размер шрифта
5set encoding utf8
6set term pdfcairo font "Arial,9"
7
8# задаём выходной файл графика
9set out 'qm.pdf'
10
11# задаём название графика
12set title "График средней длины очереди"
13
14# задаём стиль линии
15set style line 2
16
17# подписи осей графика
18set xlabel "t"
19set ylabel "Пакеты"
20
21# построение графика, используя значения
22# 1-го и 5-го столбцов файла qm.out
23plot "qm.out" using ($1):($5) with lines title "Размер очереди (в пакетах)", \
24      "qm.out" using ($1):($5) smooth csplines title " Приближение сплайном ", \
25      "qm.out" using ($1):($5) smooth bezier title " Приближение Безье "
```

Figure 2.4: Рис 4. Код программы GNUplot

3. Запускаю программу с помощью команды

`./graph`

Результат выполнения программы:

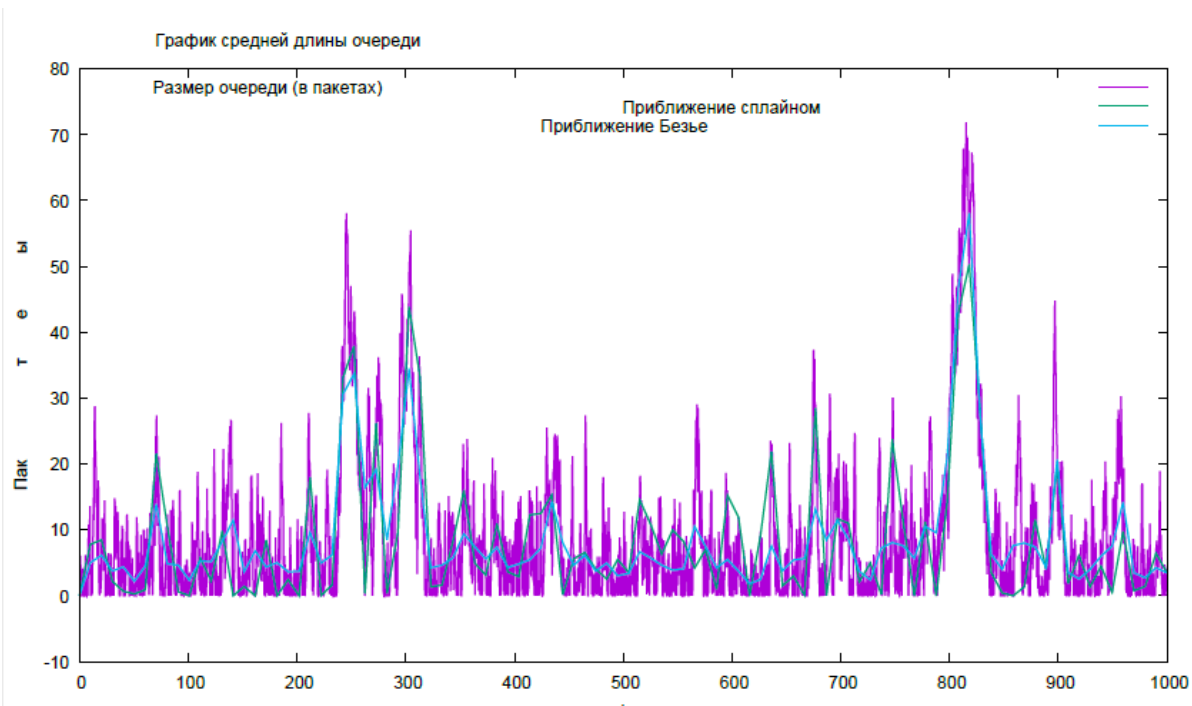


Figure 2.5: Рис 5. График №1 - Результат выполнения программы

3 Выводы

Получены навыки работы с ns2 в связке с GNUpot

4 Библиография

1. Методические материалы курса