

Лабораторная работа 3

Моделирование стохастических процессов

Хамдамова Айжана

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	11
	Список литературы	12

Список иллюстраций

4.1	скрипт	8
4.2	вывод	8
4.3	код	9
4.4	вывод	9
4.5	gm_plot	10

Список таблиц

1 Цель работы

Провести моделирование системы массового обслуживания (СМО).

2 Задание

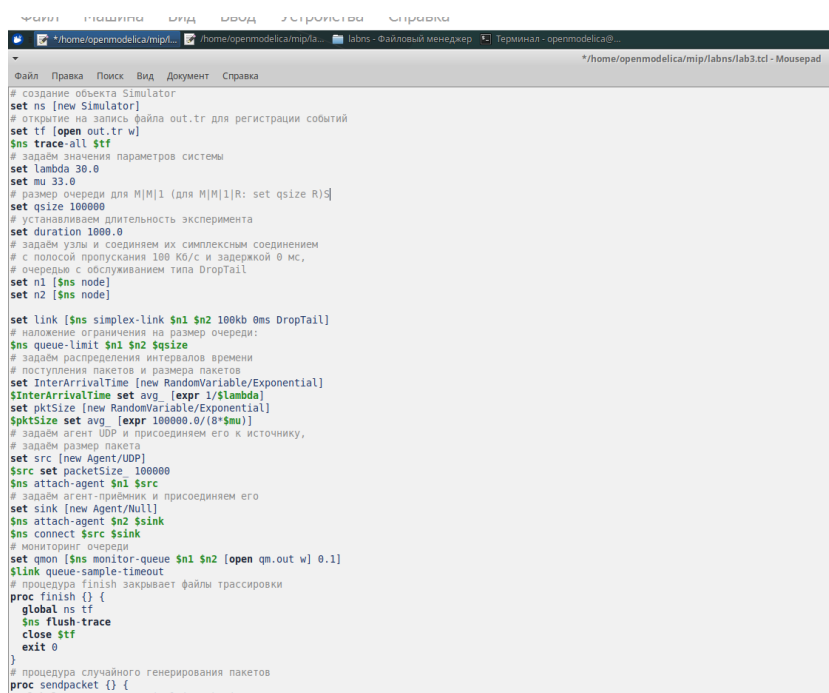
1. Реализовать модель $M|M|1$
2. Посчитать загрузку системы и вероятность потери пакетов;
3. Построить график изменения размера очереди.

3 Теоретическое введение

$M|M|1$ - это однолинейная СМО с накопителем бесконечной ёмкости. Поступающий поток заявок — пуассоновский с интенсивностью λ . Времена обслуживания заявок — независимые в совокупности случайные величины, распределённые по экспоненциальному закону с параметром μ . Реализуем эту систему. Зададим параметры системы $\lambda=30, \mu=33$, размер очереди 100000, длительность эксперимента 100000. Далее задаем узлы, между которыми будут идти пакеты, и соединяем их симплексным соединением с полосой пропускания 100 Кб/с и задержкой 0 мс, очередью с обслуживанием типа DropTail. Наложим ограничения на размер очереди. Источником трафика ставим UDP-агент, приемником Null-агент. Также осуществим мониторинг очереди. Процедура finish закрывает файлы трассировки. Процедура sendpack – случайно генерирует пакеты по экспоненциальному распределению. Также в данной сценарии рассчитывается по формулам загрузка система и вероятность потери пакетов.

4 Выполнение лабораторной работы

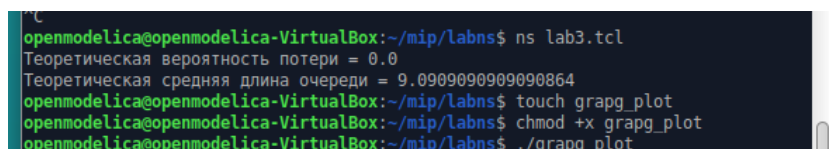
Запустив программу ниже(рис. [4.1]), получим значения загрузки системы и вероятности потери пакетов (рис. [4.2]).



```
# 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,
# очередью 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 распределение 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
$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 qn.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
```

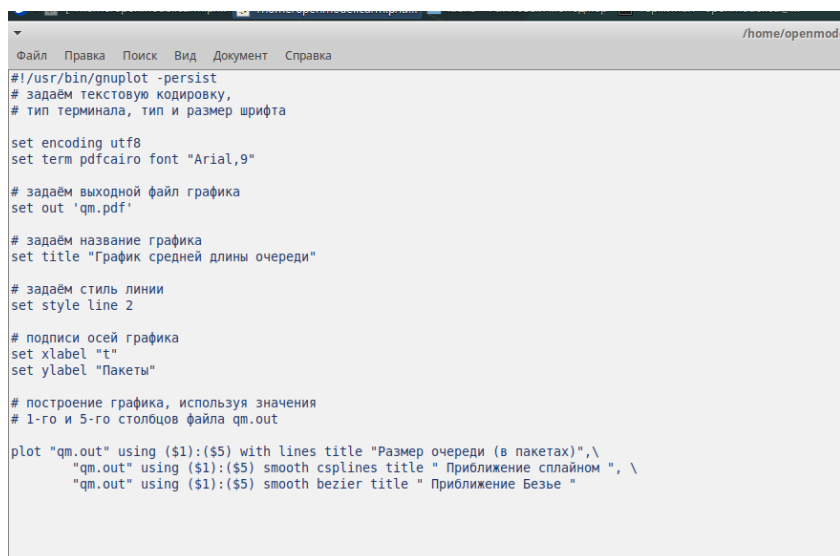
Рис. 4.1: скрипт



```
openmodelica@openmodelica-VirtualBox:~/mip/labns$ ns lab3.tcl
Теоретическая вероятность потери = 0.0
Теоретическая средняя длина очереди = 9.0909090909090864
openmodelica@openmodelica-VirtualBox:~/mip/labns$ touch grapg_plot
openmodelica@openmodelica-VirtualBox:~/mip/labns$ chmod +x grapg_plot
openmodelica@openmodelica-VirtualBox:~/mip/labns$ ./grapg_plot
```

Рис. 4.2: вывод

В каталоге с проектом создадим отдельный файл, например, `graph_plot` touch `graph_plot`. Откроем его на редактирование и добавим следующий код(рис. [4.3]), обращая внимание на синтаксис GNUplot.

A screenshot of a text editor window with a menu bar (Файл, Правка, Поиск, Вид, Документ, Справка) and a title bar showing the path /home/openmodelica. The editor contains a Gnuplot script with the following content:

```
#!/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

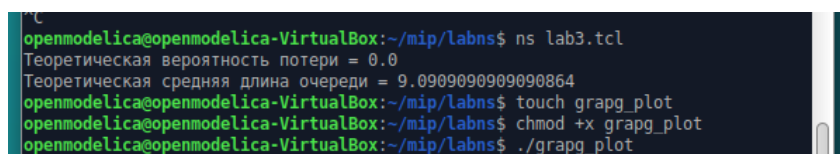
# подписи осей графика
set xlabel "t"
set ylabel "Пакеты"

# построение графика, используя значения
# 1-го и 5-го столбцов файла qm.out

plot "qm.out" using ($1):($5) with lines title "Размер очереди (в пакетах)", \
      "qm.out" using ($1):($5) smooth csplines title " Приближение сплайном ", \
      "qm.out" using ($1):($5) smooth bezier title " Приближение Безье "
```

Рис. 4.3: код

Сделаем файл исполняемым. После компиляции файла с проектом, запустим скрипт в созданном файле `graph_plot` (рис. [4.4]), который создаст файл `qm.png` с результатами моделирования (рис. [4.5]).

A screenshot of a terminal window with the prompt openmodelica@openmodelica-VirtualBox:~/mip/labns\$. The terminal shows the following commands and their outputs:

```
openmodelica@openmodelica-VirtualBox:~/mip/labns$ ns lab3.tcl
Теоретическая вероятность потери = 0.0
Теоретическая средняя длина очереди = 9.090909090909090864
openmodelica@openmodelica-VirtualBox:~/mip/labns$ touch grapg_plot
openmodelica@openmodelica-VirtualBox:~/mip/labns$ chmod +x grapg_plot
openmodelica@openmodelica-VirtualBox:~/mip/labns$ ./grapg_plot
```

Рис. 4.4: вывод

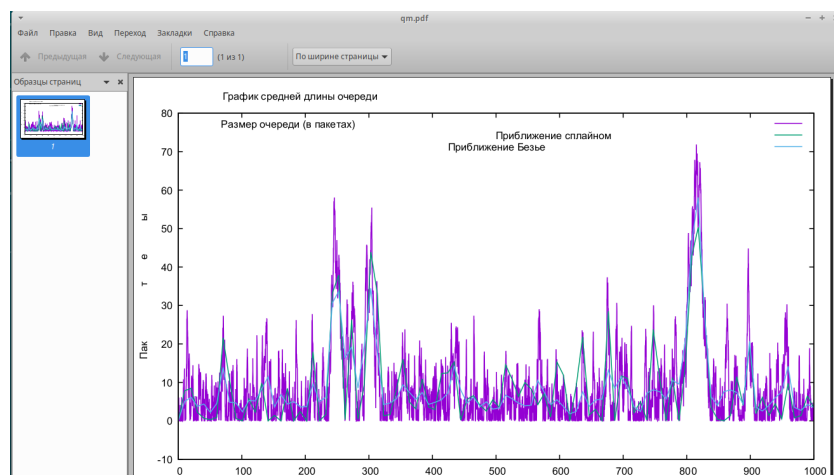


Рис. 4.5: gm_plot

На данном графике изображен размер очереди в пакетах, а также его приближение сплайном и Безье.

5 Выводы

В процессе выполнения данной лабораторной работы я провела моделирование системы массового обслуживания (СМО).

Список литературы