

Отчёт по лабораторной работе 3

Моделирование стохастических процессов

Наталья Андреевна Сидорова

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Выполнение лабораторной работы | 5 |
| 2 | Выводы | 9 |
| | Список литературы | 10 |

Список иллюстраций

| | | |
|-----|--------------------------------------|---|
| 1.1 | Код реализации | 5 |
| 1.2 | Код реализации часть 2 | 6 |
| 1.3 | Вывод результата в консоль | 6 |
| 1.4 | Код построения графика | 7 |
| 1.5 | Нужные файлу права | 7 |
| 1.6 | Получившийся график | 8 |

Список таблиц

1 Выполнение лабораторной работы

Реализовала на NS-2 модель системы массового обслуживания (рис. 1.1).

```
# 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
$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
```

Рис. 1.1: Код реализации

Вторая часть кода (рис. 1.2).

```

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
$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
}

# 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. 1.2: Код реализации часть 2

Теоретическая вероятность потери получилась равная 0, теоретическая средняя длина очереди получилась 9,09 (рис. 1.3).

```

openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ns lab3.tcl
Теоретическая вероятность потери = 0.0
Теоретическая средняя длина очереди = 9.0909090909090864
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$

```

Рис. 1.3: Вывод результата в консоль

Создала график в GNUplot, на нем изображен размер очереди в пакетах, приближение сплайном и приближение Безье (рис. 1.4).

```
#!/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. 1.4: Код построения графика

Сделала файл исполняемым, дав необходимые права (рис. 1.5).

```
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ touch graph_plot
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ chmod +x graph_plot
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ./graph_plot
```

Рис. 1.5: Нужные файлу права

Вывод графика (рис. 1.6).

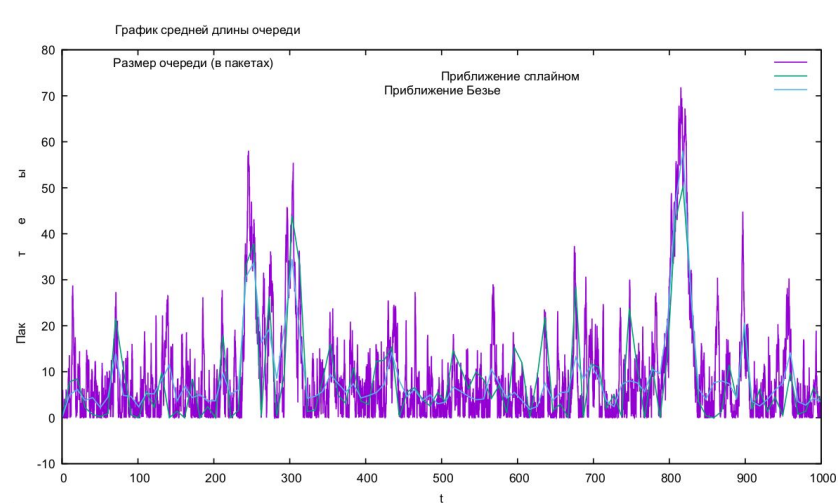


Рис. 1.6: Получившийся график

2 Выводы

В этой лабораторной я создала модель системы массового обслуживания.

Список литературы