

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

дисциплина: Моделирование информационных процессов

Студент:

Ибрагимов Улугбек

Группа:

НФИбд-02-20

МОСКВА

2023 г.

Цель	3
Задачи	4
Выполнение лабораторной работы.....	5
Анализ результатов	8
Вывод.....	9

Цель

Приобретение навыков моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также анализ полученных результатов моделирования.

Задачи

1. Произвести моделирование стохастических процессов на примере СМО $M|M|n$ в R.
2. Построить график в GNUplot

Выполнение лабораторной работы

1. Напишем скрипт выполнения модели:

```
# 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 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

```

2. Построим график в GNUpot. Для этого создадим файл и напишем в нем следующий скрипт. После сделаем данный файл исполняемым.

```

#!/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 "

```

3. Скомпилируем модель ns-2, после этого запустим вышесказанный скрипт, на выходе получаем файл pdf с результатами моделирования (рис. 1):

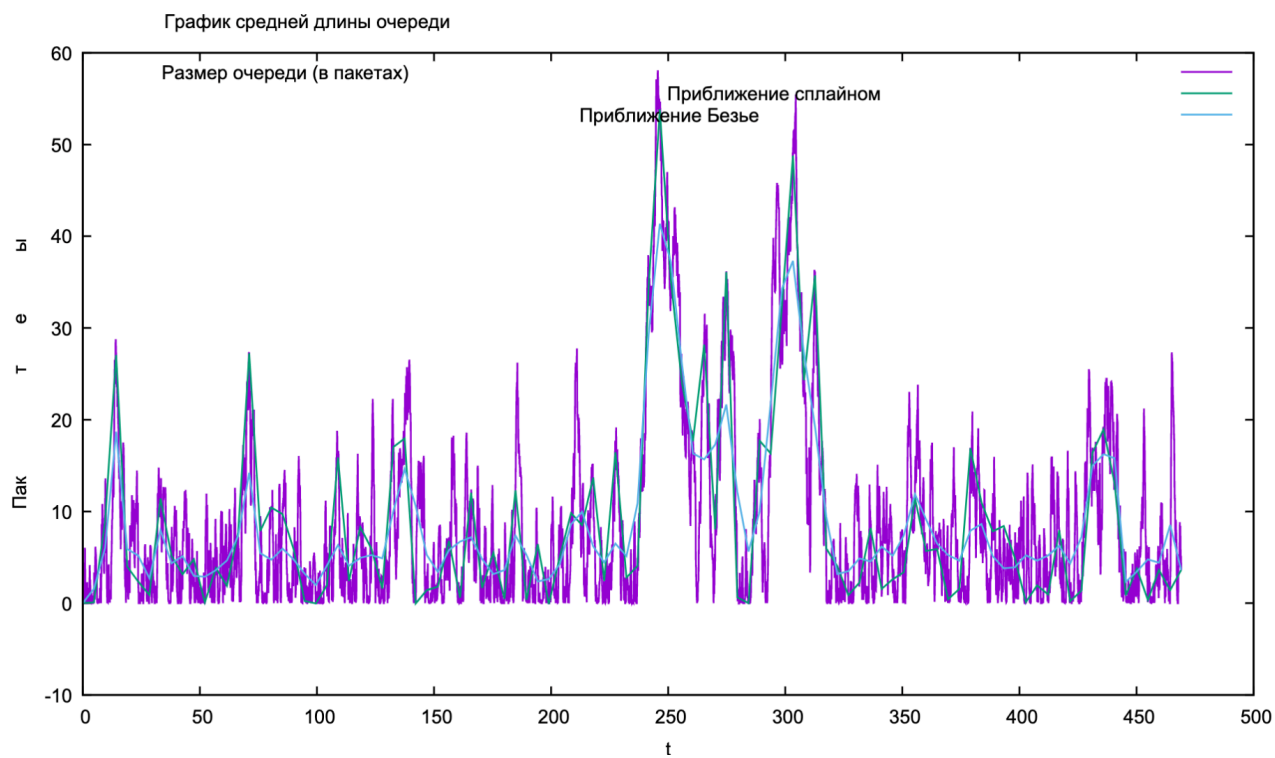


Рис. 1. График поведения длины очереди

Анализ результатов

В рамках выполнения работы произошли некоторые незначительные проблемы, но все они решены. В остальном работа выполнена без непредвиденных проблем в соответствии с руководством: ошибок и сбоев не произошло.

Стоит отметить, что написание скриптов NS-2 происходи крайне комфортно, синтаксис, объекты интуитивно понятны. Также сами скрипты довольно быстро выполняются.

Синтаксис GNUpot также понятен и не вызывает проблем.

Вывод

В результате выполнения работы, были получены и улучшены практические навыки моделирования на NS-2. Также были созданы, настроены и смоделирована модель стохастических процессов. В добавок были приобретены навыки построения графиков при помощи средства GNUpot.