Лабораторная работа №3

Моделирование стохастических процессов

Игнатенкова Варвара Николаевна

Содержание

[1 Цель работы 1](#_Toc192294681)

[2 Задание 1](#_Toc192294682)

[3 Выполнение лабораторной работы 1](#_Toc192294683)

[4 Выводы 4](#_Toc192294684)

# 1 Цель работы

Провести моделирование системы массового обслуживания (СМО).

# 2 Задание

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

# 3 Выполнение лабораторной работы

– это однолинейная СМО с накопителем бесконечной ёмкости. Поступающий поток заявок — пуассоновский с интенсивностью . Времена обслуживания заявок — независимые в совокупности случайные величины, распределённые по экспоненциальному закону с параметром .

Реализуем эту систему. Зададим параметры системы , 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.

set ns [new Simulator]

set tf [open out.tr w]

$ns trace-all $tf

set lambda 30.0

set mu 33.0

set qsize 100000

set duration 1000.0

set n1 [$ns node]

set n2 [$ns node]

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

$ns queue-limit $n1 $n2 $qsize

set InterArrivalTime [new RandomVariable/Exponential]

$InterArrivalTime set avg\_ [expr 1/$lambda]

set pktSize [new RandomVariable/Exponential]

$pktSize set avg\_ [expr 100000.0/(8\*$mu)]

set src [new Agent/UDP]

$src set packetSize\_ 100000

$ns attach-agent $n1 $src

set sink [new Agent/Null]

$ns attach-agent $n2 $sink

$ns connect $src $sink

set qmon [$ns monitor-queue $n1 $n2 [open qm.out w] 0.1]

$link queue-sample-timeout

proc finish {} {

global ns tf

$ns flush-trace

close $tf

exit 0

}

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

}

$ns at 0.0001 "sendpacket"

$ns at $duration "finish"

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"

$ns runЗапустив эту программу, получим значения загрузки системы и вероятности потери пакетов (рис. 1).

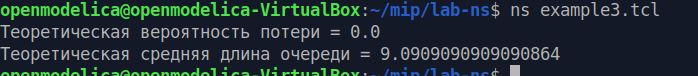


Рис. 1: Результат выполнения программы

В каталоге с проектом создадим отдельный файл, например, graph\_plot touch graph\_plot. Откроем его на редактирование и добавим следующий код, обращая внимание на синтаксис GNUplot (рис. 2).

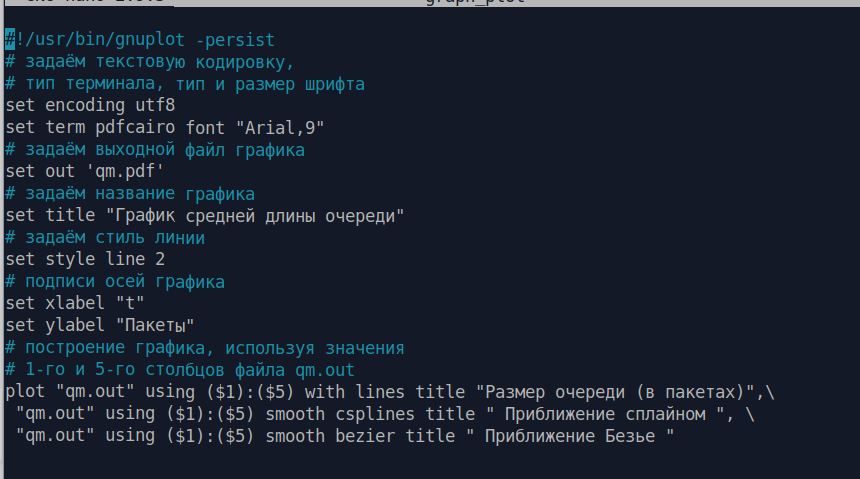


Рис. 2: Листинг программы для отрисовки графика поведения длины очереди в пакетах

Сделаем файл исполняемым. После компиляции файла с проектом, запустим скрипт в созданном файле graph\_plot (рис. 3), который создаст файл qm.png с результатами моделирования (рис. 4).



Рис. 3: Запуск программы отрисовки графика

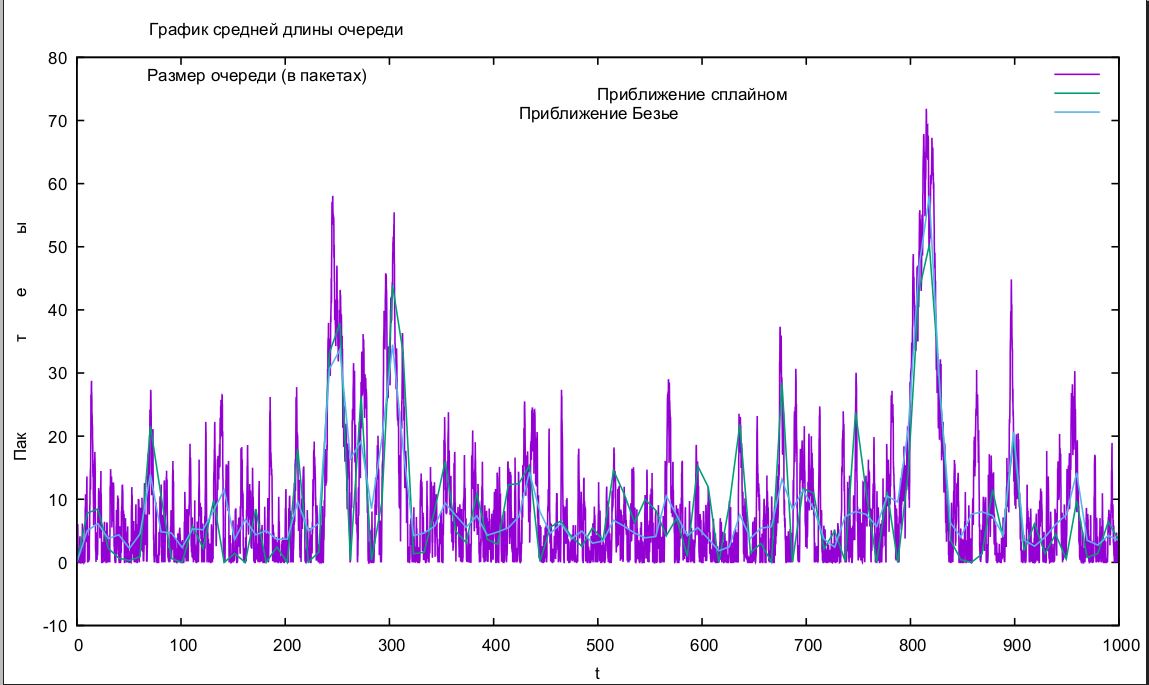


Рис. 4: График поведения длины очереди

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

# 4 Выводы

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