Отчёт по лабораторной работе №3

Моделирование стохастических процессов

Ибатулина Дарья Эдуардовна, НФИбд-01-22

Содержание

# 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

– 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 .

# 4 Выполнение лабораторной работы

## 4.1 Реализация модели M|M|1 на NS-2

Реализуем эту систему, описанную в теоретическом введении. Зададим параметры системы , размер очереди 100000, длительность эксперимента 1000.0. 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. В директории для лабораторных работ создаем файл, в котором будем описывать модель в NS-2. Запускаем его и получаем результат - теоретическая вероятность потери равна 0.0, теоретическая средняя длина очереди равна примерно 9 (рис. [[1](#fig:001)]).

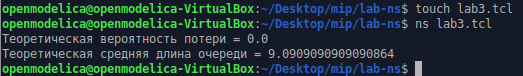


Figure 1: Модель M|M|1 на NS-2

Программа, задающая модель:

# 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

## 4.2 График в GNUplot

В каталоге для лабораторных работ создаю отдельный файл *graph\_plot* и записываем в него код (рис. [[2](#fig:002)]).

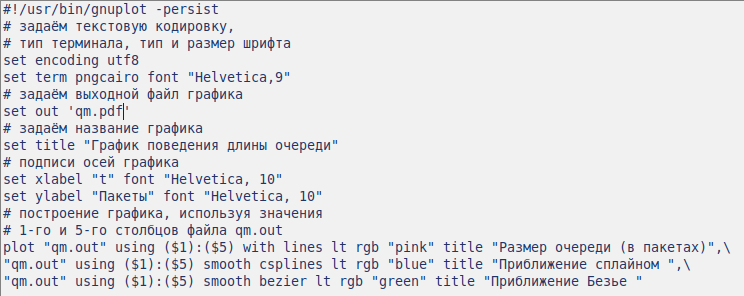


Figure 2: Программа для графика в GNUplot

Сделаю файл исполняемым. После компиляции файла с проектом, запускаю скрипт в созданном файле graph\_plot, который создаст файл qm.pdf (рис. [[3](#fig:003)]) с результатами моделирования (рис. [[4](#fig:004)].)

Figure 3: Передача файлу прав на исполнение и компиляция файла

Figure 3: Передача файлу прав на исполнение и компиляция файла

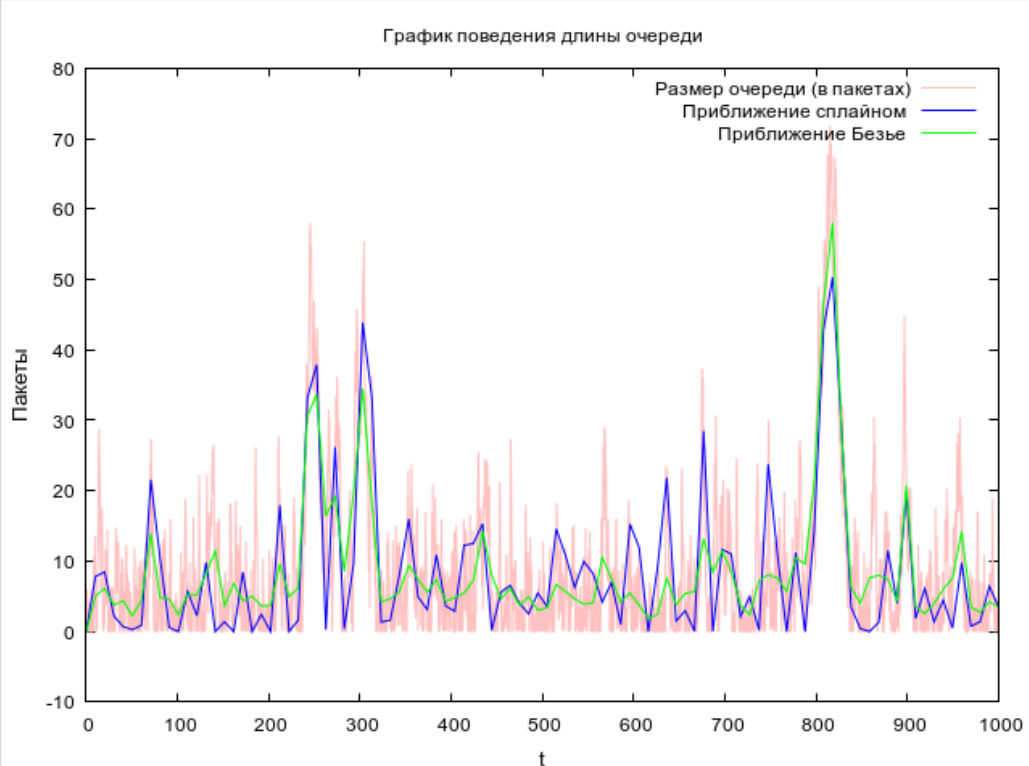


Figure 4: Результат моделирования

Программа для графика в GNUplot:

#!/usr/bin/gnuplot -persist  
# задаём текстовую кодировку,  
# тип терминала, тип и размер шрифта  
set encoding utf8  
set term pngcairo font "Helvetica,9"  
# задаём выходной файл графика  
set out 'qm.pdf'  
# задаём название графика  
set title "График поведения длины очереди"  
# подписи осей графика  
set xlabel "t" font "Helvetica, 10"  
set ylabel "Пакеты" font "Helvetica, 10"  
# построение графика, используя значения  
# 1-го и 5-го столбцов файла qm.out  
plot "qm.out" using ($1):($5) with lines  
 lt rgb "pink" title "Размер очереди (в пакетаx)",\  
 "qm.out" using ($1):($5) smooth csplines  
 lt rgb "blue" title "Приближение сплайном ",\  
 "qm.out" using ($1):($5) smooth bezier  
 lt rgb "green" title "Приближение Безье "

# 5 Выводы

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

# Список литературы

1. Королькова А.В., Кулябов Д.С. Руководство к лабораторной работе №3. Моделирование информационных процессов. - 2025. — 6 с.