РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

УТВЕРЖДАЮ
Заведующий кафедрой
прикладной информатики
и теории вероятностей
д.т.н., профессор
_____ К.Е. Самуйлов
«____»_____ 2021 г.

КУРСОВАЯ РАБОТА

на тему

«Численный анализ вероятностно-временных характеристик модели с нетерпеливым эластичным трафиком и минимальной скоростью»

по дисциплине «Компьютерный практикум по математическому моделированию»

Выполнила:

Студентка группы НКНбд-01-18

Студенческий билет №: 1032182530

_____ П.В. Азарцова

« 25 » ____декабря____ 2021 г.

Руководительница:

Доцент кафедры
прикладной информатики и теории
вероятностей, к.ф.-м.н.

_____ И.А. Кочеткова

Преподавательница:

Доцент кафедры
прикладной информатики и теории
вероятностей, к.ф.-м.н.

_____ А.В. Королькова

Москва 2021

Содержание

Список используемых сокращений

Русскоязычные сокращения

СМО	Система массового обслуживания
СУГБ	Система уравнений глобального баланса
СУЛБ	Система уравнений локального баланса

Англоязычные сокращения

QoS	Quality of Service

Введение

С каждым днем количество пользователей мобильных сетей в мире значительно увеличивается. В статистических данных можно увидеть, что на октябрь 2021 приходится около 4.88 миллиардов пользователей интернета по всему миру, что соответствует почти 62 процентам всего мирового населения. Количество пользователей растет со скоростью 4,8 процента в год, что составляет в среднем более 600 000 новых пользователей каждый день. [1]

Так как мобильные сети поддерживают всё более значительное число пользователей, запрашивающих разные услуги, в свою очередь предъявляющие различные требования качеству обслуживания QoS, обычное выделение ресурсов беспроводных сетей приводит к малоэффективному их использованию с относительно высокими затратами.

Актуальность данной работы обусловлена тем, что операторы заинтересованы в нахождении решений для более эффективного удовлетворения требований всё прибывающих пользователей, одним из которых является технология нарезки сетей (англ. network slicing). Концепция внедрения распределения ресурсов между виртуальными мобильными операторами является основой современных сетей пятого поколения, чем соответственно и обуславливается задача проектирования модели сегмента сети, включающей в себя различные характеристики предоставляемых услуг. [2]

Целью моей курсовой работы является изучение и анализ моносервисной модели сегмента системы массового обслуживания (СМО), а также разработка программного кода для решения задачи нарезки ресурсов и его реализация.

Основными задачами моей работы являются:

1. Формализация сегмента и численных характеристик моносервисной модели СМО с нетерпеливым эластичным трафиком и минимальным ограничением скорости с перераспределением радиоресурса.
2. Программная реализация на языке программирования Python задачи нарезки ресурсов в формализованной модели с интенсивностями переходов, решением в явном виде и анализом среднего числа заявок в состояниях.

Методом исследования является применение метода теории вероятности и теории массового облуживания.

Курсовая работа состоит из введения, двух разделов, заключения, списка используемой литературы и приложения. Во введении обоснована актуальность выполнения курсовой работы и приведена статистика, сформулированы цель и задачи, а также предоставлено описание разделов.

В первом разделе рассматриваются теория технологии нарезки ресурсов, предоставляются основные понятия, определения и характеристики, а также теория самой математической модели сегмента и её вероятностные характеристики.

Во втором разделе задаются уже определенные численные характеристики и на их основании проводятся вычисления и реализация программного кода с дальнейшим анализом.

В заключении подведены общие итоги курсовой работы.

1 Основная задача нарезки радиоресурсов беспроводной сети

Нарезка сети (network slicing) является одной из основных функций сетей следующего поколения. Вместо привычного разделения ресурсов, было внедрено сетевое сегментирование, т.е. разделение радиоресурса на слайсы (англ. slice) или сегменты, позволяющее операторам обеспечивать необходимую пользователям функциональность и параметры обслуживания.

1.1 Характеристика концепции нарезки радиоресурсов

Несмотря на широкое распространение, понятие радиоресурса семантически четко не определено.[3] Для технологии нарезки сети радиоресурс можно характеризовать в трёх измерениях: по мощности, частоте и времени. (Рисунок 1.1: Наглядное изображение радиоресурса)

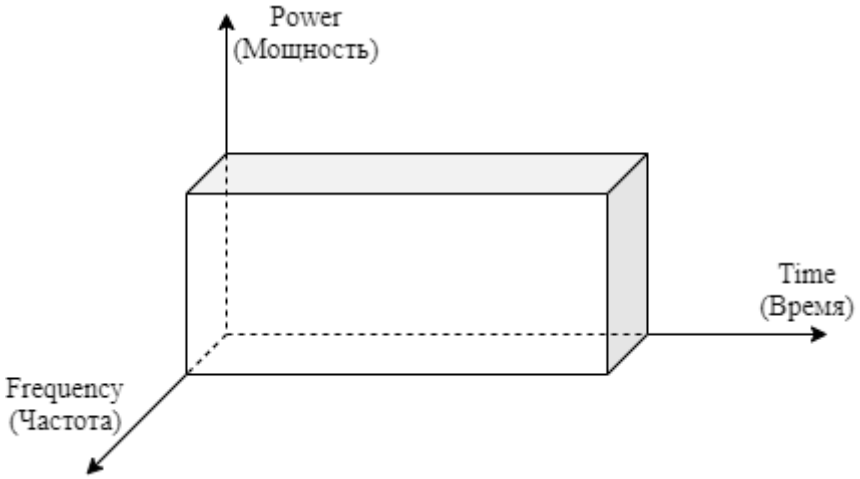

Рисунок 1.1.1: Наглядное изображение радиоресурса

По технологии нарезки сети базовый оператор обладает радиоресурсами и виртуально распределяет их другим операторам, между которыми они разделяются уже определенным образом, т.е. сами виртуальные операторы ресурсами не владеют. Между базовым и виртуальными операторами создается соглашение о качестве обслуживания, устанавливающее требования сети от каждого виртуального оператора базовому.

В самом простом случае разделение радиоресурсов по соглашению детерминировано (т.е. жестко фиксировано). А так как виртуальные операторы также предоставляют услуги своим пользователям, для которых в приоритете стоит

качество обслуживания на определенном достаточном для них уровне, соответственно и виртуальным операторам важнее не четко фиксированное распределение радиоресурсов, а постоянное выполнение потребностей своих пользователей. Гибким решением данного запроса может являться динамическое (с точки зрения занятия ресурса) распределение между виртуальными операторами. Эффективное динамическое распределение ресурса является одной из основных задач network slicing.

При динамическом распределении ресурсов, в системе присутствует контроллер, который и осуществляет саму функцию нарезки. Чтобы произошло перераспределение ресурсов между виртуальными операторами в соответствии с их требованиями, контроллер должен дать команду, которая станет дополнительной нагрузкой на сеть.

Следовательно, можно понять, что с одной стороны при наличии возможности непрерывно перераспределять ресурсы, например, в зависимости от изменяющихся количества пользователей или объема данных, преимущество будет заключаться в возможности гибко подстраиваться под любые предъявленные требования сети. Однако, с другой стороны, добавляется нагрузка сигнального трафика на сеть, из-за чего могут возникать задержки. Если же разделить ресурсы фиксировано, нагрузка сигнального трафика полностью исчезнет, но мы также потеряем и преимущество гибкости.

Выйти из данного положения может помочь гибридный вариант распределения ресурсов, в котором через какой-либо определенный промежуток времени контроллер проверяет состояние системы и принимает решение о перераспределении радиоресурсов. (Рисунок 1.2: Наглядное изображение динамического распределения ресурсов между двумя операторами с временным интервалом Δ) Отсюда и возникает оптимизационная задача того, какой взять интервал времени для наиболее эффективного перераспределения радиоресурсов в системе. [4]

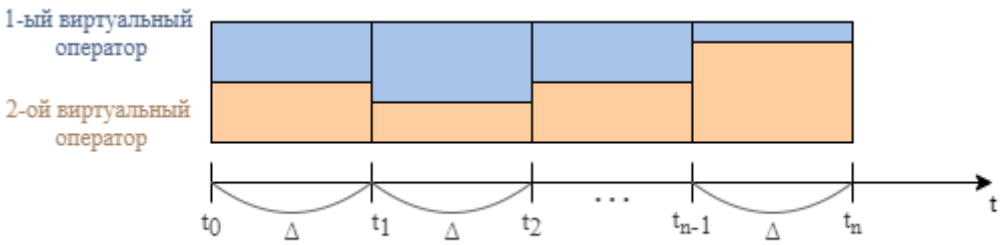

Рисунок 1.2: Наглядное изображение динамического распределения ресурсов между двумя операторами с временным интервалом Δ

1.2 Эластичный трафик

Обобщенно можно выделить два типа трафика: потоковый и эластичный. Потоковый определяется своей равномерностью, т.е. фиксированной скоростью и фиксированным временем обслуживания. Эластичный трафик определяется, наоборот, переменными скоростью и временем обслуживания, однако фиксированной длиной файла.

Чтобы подойти к оптимизационной задаче нарезки ресурсов, в данной курсовой рассмотрена конкретная задача с одной услугой – загрузка файлов. Исходя из этого, система будет с эластичным трафиком.

Помимо фиксированного размера файла эластичный трафик обычно характеризуется интенсивностями поступления запросов и минимально возможной скоростью, которая задается с целью соблюдения ограничения на время передачи файла, необходимого для эффективной работы системы. Если минимальное ограничение скорости не соблюдается, то трафик будет ожидать начала своего обслуживания.

2 Сегмент моносервисной модели сети с эластичным трафиком

Рассмотрим базовый вырожденный случай системы, когда существует лишь один виртуальный оператор и ему доступны все ресурсы. В представленном случае как таковой нарезки сетей не присутствует, но он является принципиально базовым для дальнейшего рассмотрения технологии.

2.1 Построение математической модели

Построим математическую моносервисную модель сегмента сети с нетерпеливым эластичным трафиком и минимальным ограничением по скорости.

В данной модели мы рассматриваем Пуассоновский входящий поток запросов с интенсивностью λ, экспоненциальный объем файла $S = \frac{1}{\mu^{-1}}$ при μ – интенсивности обслуживания запроса и минимальное ограничение скорости b. Если ограничение скорости b не соблюдается, то заявка отправляется в очередь, максимальная длина которой составляет R – максимальное число заявок, которые могут ожидать обслуживания, когда все приборы заняты. Также в модели есть приборы, которые являются ресурсом C, обеспечивающим некоторую максимально возможную скорость обслуживания. При существующей необходимости обеспечить минимальную скорость b, максимальное количество заявок, которые могут одновременно находиться на обслуживании (т.е. на приборе) будет $N = \left[\frac{C}{b}\right]$.

Причем если после отправки пользователем запроса на обслуживание оказывается, что на приборе меньше, чем N заявок, то запрос будет принят если же N заявок уже обслуживается, то пользователь будет ожидать в очереди не бесконечный промежуток времени, а при слишком долгом ожидании он покинет систему, так определяются дополнительные нетерпеливые заявки, покидающие очередь с интенсивностью γ. (Рисунок 2.1: Схема сегмента сети с нетерпеливым эластичным трафиком и минимальным ограничением по скорости)

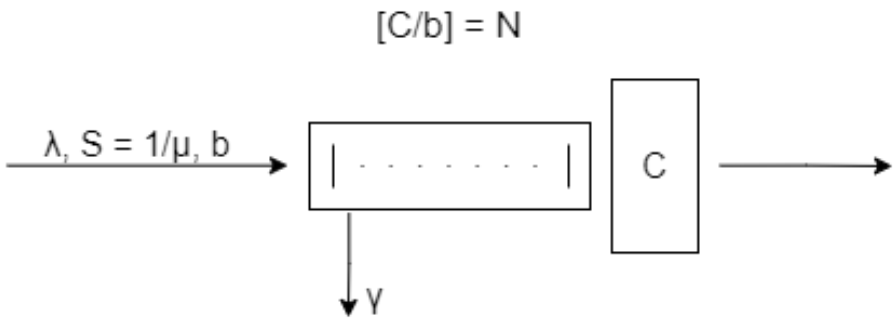

Рисунок 2.1: Схема-модель сегмента сети с нетерпеливым эластичным трафиком и минимальным ограничением по скорости

Итак, введем случайный процесс – число заявок в системе X(t). Пространство состояний исследуемого случайного процесса будет иметь вид:
$X = \{n: 0 \le n \le N + R\}$. Выведем граф интенсивностей переходов между состояниями для общего случая. (Рисунок 2.2: Граф интенсивностей переходов)

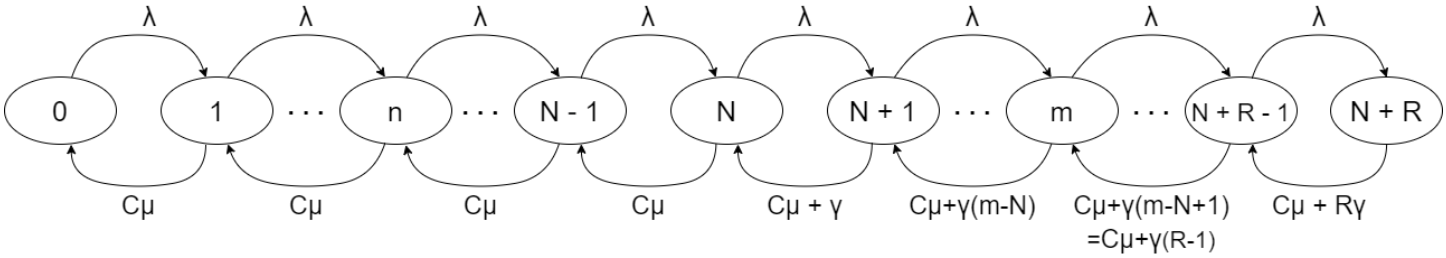

Рисунок 2.2: Граф интенсивностей переходов

Теперь необходимо составить систему уравнений глобального баланса в общем виде.

$$\begin{cases} \lambda p_0 = C\mu p_1 \\ (\lambda + C\mu)p_n = \lambda p_{n-1} + C\mu p_{n+1}, \qquad n = \overline{1 \dots N-1} \\ (\lambda + C\mu)p_N = \lambda p_{N-1} + (C\mu + \gamma)p_{N+1} \\ p_m\big(\lambda + C\mu + \gamma(m-N)\big) = \lambda p_{m-1} + \big(C\mu + \gamma(m-N+1)\big)p_{m+1}, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad m = \overline{N+1 \dots N+R-1} \\ \lambda p_{N+R-1} = (C\mu + R\gamma)p_{N+R} \end{cases}$$

А также составляем систему уравнений локального баланса в общем виде.

$$\begin{cases} \lambda p_{n-1} = C\mu p_n\,, & n = \overline{1 \dots N} \quad (1) \\ \lambda p_{m-1} = (C\mu + \gamma(m-N))p_m\,, & m = \overline{N+1 \dots N+R} \quad (2) \end{cases}$$

Из СУЛБ (1) получаем:

$$p_n = \left(\frac{\lambda}{C\mu}\right)^n p_0, \qquad n = \overline{1 \dots N} \qquad (3)$$

Из СУЛБ (2) получаем:

$$p_m = \lambda^{m-N} p_N \prod_{i=1}^{m-N} (C\mu + i\gamma)^{-1}\,, \qquad m = \overline{N+1 \dots N+R} \qquad (4)$$

Сводим получившиеся (3) и (4) и получаем:

$$p_N = \left(\frac{\lambda}{C\mu}\right)^N p_0 \;\Rightarrow\; p_m = \lambda^{m-N}\left(\frac{\lambda}{C\mu}\right)^N p_0 \prod_{i=1}^{m-N} (C\mu + i\gamma)^{-1},$$

$$\mathrm{m} = \overline{\mathrm{N}+1 \dots \mathrm{N}+\mathrm{R}}$$

Отсюда выводим стационарное распределение вероятностей:

$$p_n = \begin{cases} \left(\frac{\lambda}{C\mu}\right)^n p_0, & \mathrm{n} = \overline{0 \dots \mathrm{N}} \\ \lambda^{n-N}\left(\frac{\lambda}{C\mu}\right)^N p_0 \prod_{i=1}^{n-N} (C\mu + i\gamma)^{-1}\,, & \mathrm{n} = \overline{\mathrm{N}+1 \dots \mathrm{N}+\mathrm{R}} \end{cases}$$

Воспользовавшись условием нормировки, получили начальную вероятность.

Условие нормировки:

$$p_0(1+\frac{\lambda}{C\mu}+\left(\frac{\lambda}{C\mu}\right)^2+\cdots+\left(\frac{\lambda}{C\mu}\right)^N+\frac{\lambda\left(\frac{\lambda}{C\mu}\right)^N}{C\mu+\gamma}+\frac{\lambda^2\left(\frac{\lambda}{C\mu}\right)^N}{(C\mu+\gamma)(C\mu+2\gamma)}+\cdots$$
$$+\frac{\lambda^R\left(\frac{\lambda}{C\mu}\right)^N}{(C\mu+\gamma)\dots(C\mu+R\gamma)})=1$$

Начальная вероятность:

$$p_0=(\sum_{n=0}^{N}\left(\frac{\lambda}{C\mu}\right)^n+\left(\frac{\lambda}{C\mu}\right)^N\sum_{n=N+1}^{N+R}\frac{\lambda^{n-N}}{\prod_{i=1}^{n-N}(C\mu+i\gamma)})^{-1}$$

А также вывели некоторые численные характеристики.

Среднее число заявок, находящихся на обслуживании:

$$\overline{N}_{\text{обсл}}=\sum_{n=0}^{N}np_n+N\sum_{m=1}^{R}p_{N+m}$$

Среднее число заявок, находящихся в очереди:

$$\overline{N}_{\text{оч}}=\sum_{n=1}^{R}np_{N+n}$$

Среднее число заявок, находящихся в системе:

$$\overline{N}_{\text{сис}}=\sum_{n=0}^{N+R}np_n$$

Вероятность блокировки:

$$B=p_{N+R}$$

2.2 Численный анализ показателей системы

Изначально для наглядности возьмём небольшие значения.

Таблица 2.1: Исходные данные 1

Параметр	Характеристика	Заданное значение
C	Объём ресурса	4
R	Длина очереди (максимальное возможное чисто заявок в очереди)	2
b	Минимальная обеспечиваемая скорость	1
λ	Интенсивность поступления заявок	3
μ	Интенсивность обслуживания заявок	2
γ	Интенсивность ухода из системы нетерпеливых заявок	1
N	Максимальное число заявок на обслуживании ([C/b])	4

С использованием выведенных формул была написана программа, с помощью которой можно вывести все основные элементы модели. (Описание всех заданных функций, а также листинг программы представлены в приложении)

Пространство состояний и его размер: (Рисунок 2.3)

```
e = Elastic(C, R, b, λ, μ, γ)
print('Пространство состояний: ', e.X, 'Размер: ', e.len, sep='\n')
```

```
Пространство состояний:
[0, 1, 2, 3, 4, 5, 6]
Размер:
7
```

Рисунок 2.3: Пространство состояний и его размер

Начальная вероятность:

```
e.prob_zero()
```

```
0.6262901178881056
```

Рисунок 2.4: Начальная вероятность

Распределение вероятностей заданной системы:

```
Pn = e.prob_distribution()
for i in range (0, int(e.N)+R+1):
    print(f'p{i}=',Pn[i])
```

```
p0= 0.6262901178881056
p1= 0.23485879420803962
p2= 0.08807204782801485
p3= 0.03302701793550557
p4= 0.01238513172581459
p5= 0.004128377241938197
p6= 0.0012385131725814588
```

Рисунок 2.5: Распределение вероятностей

Вероятность определенного состояния:

```
Pn[3]
```

```
0.03302701793550557
```

Рисунок 2.6:Вероятность определенного состояния

Среднее число заявок, находящихся на обслуживании:

```
Q_ser = e.mean_quantity_service()
Q_ser
```

```
0.581092032231923
```

Рисунок 2.7: Среднее число заявок, находящихся на обслуживании

Среднее число заявок, находящихся в очереди:

```
Q_q = e.mean_quantity_queue()
Q_q
```

```
0.006605403587101114
```

Рисунок 2.8: Среднее число заявок, находящихся в очереди

Среднее число заявок, находящихся в системе:

```
Q_sys = e.mean_quantity_system()
Q_sys
```

```
0.5876974358190241
```

Рисунок 2.9: Среднее число заявок, находящихся в системе

Вероятность блокировки:

```
B = e.bloking_prob()
B
```

```
0.0012385131725814588
```

Рисунок 2.10:Вероятность блокировки

Задаем более подходящие исходные данные для построения графиков по численным характеристикам:

Таблица 2.2: Исходные данные 2

Параметр	**Характеристика**	**Заданное значение**
C	Объём ресурса	100
R	Длина очереди (максимальное возможное чисто заявок в очереди)	40
b	Минимальная обеспечиваемая скорость	4
λ	Интенсивность поступления заявок	8
μ	Интенсивность обслуживания заявок	6
γ	Интенсивность ухода из системы нетерпеливых заявок	2
N	Максимальное число заявок на обслуживании ([C/b])	25

При изменении λ от 0 до 1000 мы получаем следующий график колебаний среднего числа заявок в системе, на приборе и в очереди.(Рисунок 2.11: Колебание среднего числа заявок в зависимости от λ) Прослеживается логически очевидная корреляция увеличения числа заявок в каждом варианте при увеличении интенсивности поступления заявок. Также на графике можно заметить моменты полного заполнения, например, синий график

перестает менять значение относительно количества заявок, так как достигает своего предела в N = 25. Остальные действуют аналогично.

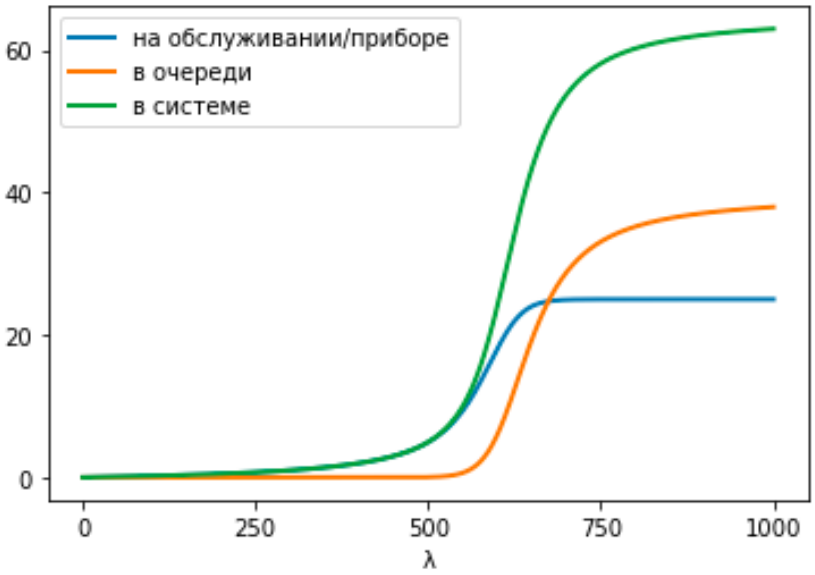

Рисунок 2.11: Колебание среднего числа заявок в зависимости от λ

При изменении μ от 1 до 60 мы получаем следующий график колебаний среднего числа заявок в системе, на приборе и в очереди.(Рисунок 2.12: Колебание среднего числа заявок в зависимости от μ) В данном случае, интенсивность изменяется от 1 , так как находится в знаменателе и до 61, потому что изменения незначительны на большем интервале. На графике можно заметить, что число заявок на приборе и в системе совпадают, и соответственно число приборов в очереди при этом будет равно 0. Данный пример иллюстрирует то, что $\bar{N}_{\text{сис}} = \bar{N}_{\text{обсл}} + \bar{N}_{\text{оч}}$.

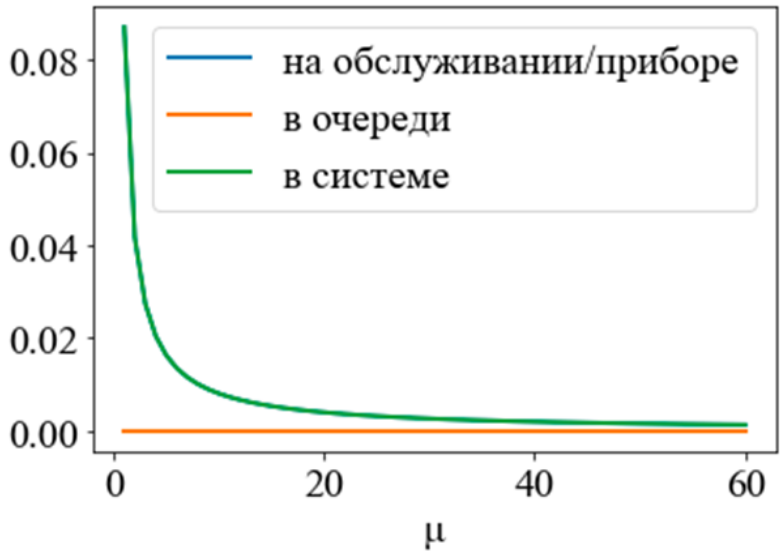

Рисунок 2.122: Колебание среднего числа заявок в зависимости от μ

При изменении γ от 0 до 1000 мы получаем следующий график колебаний среднего числа заявок в системе, на приборе и в очереди.(Рисунок 2.13: Колебание среднего числа заявок в зависимости от μ) В данном случае, можем наблюдать то, что числа заявок не зависят от интенсивности ухода заявок из очереди.

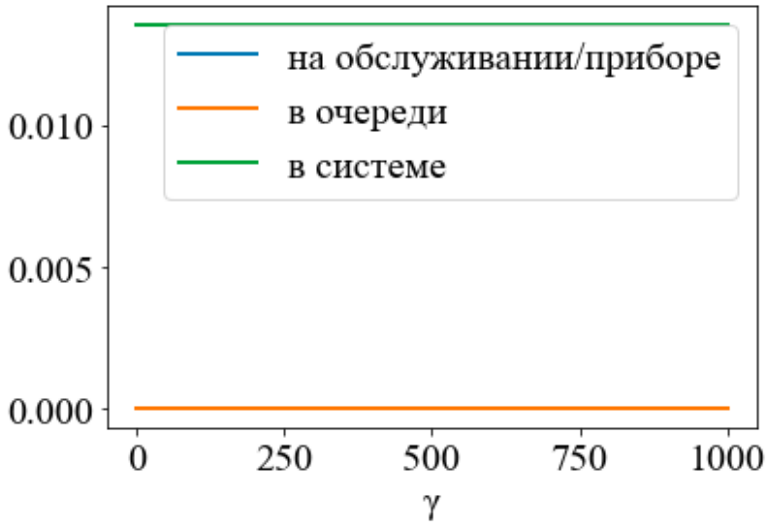

Рисунок 2.133: Колебание среднего числа заявок в зависимости от γ

Заключение

В процессе выполнения работы был проведен обзор математической модели сегмента сети с нетерпеливым эластичным трафиком и минимальным ограничением по скорости, получены и в общем виде, и с помощью реализации программы на Python такие элементы системы, как пространство состояний, его размер, граф интенсивностей переходов, распределение вероятностей, среднее число заявок, находящихся на обслуживании, в очереди и в системе, вероятность блокировки, а также выведены все формулы для соответствующих построений и нарисованы графики колебаний численных характеристик.

3 Список литературы по теме

1. Digital Around The World, Kepios, URL: https://datareportal.com/global-digital-overview (дата обращения: 24.12.2021)
2. Кочеткова И.А., Власкина А.С., Ву Н.Н., Шоргин В.С. Система массового обслуживания с управляемым по сигналам перераспределением приборов для анализа нарезки ресурсов сети 5G // Информатика и ее применения. – 2021. – Т. 15. – выпуск 3. – С.91-97
3. Шушков Александр Владимирович, Понаморев Алексей Валерьевичтом. Формализация понятия радиоресурса и направления повышения эффективности его использования // I-methods. – 2019. – Т. 11. – С. 3
4. Irina Kochetkova, Anastasia Vlaskina, Sofia Burtseva, Valeria Savich, Jiri Hosek // Analyzing the Effectiveness of Dynamic Network Slicing Procedure in 5G Network by Queuing and Simulation Models Internet of Things, Smart Spaces, and Next Generation Networks and Systems, 2020. – С. 71-85. – URL: https://doi.org/10.1007/978-3-030-65726-0_7

4 Приложение

Листинг программы на языке программирования на Pyhon:

1. Заданные функции

```
import math


class Elastic:
    def __init__(self, C, R, b, λ, μ, γ):
        self.C = C #объем ресурса
        self.R = R #макс число заявок в очереди
        self.b = b #мин обеспечиваемая скорость
        self.λ = λ #интенсивность поступления заявок
        self.μ = μ #интенсивность обслуживания заявок
        self.γ = γ #интенсивность ухода "нетерпеливых" заявок
        self.N = C/b #макс число заявок на обслуживании
        self.M = self.N + R
        #пространство состояний
        self.X = []
        for p in range(0, int(self.N)+self.R+1):
            self.X.append(p)
        #размер
        self.len = len(self.X)


        #вероятность начального состояния
    def prob_zero(self):
            P0_1 = sum((self.λ/(self.C*self.μ))**n for n in range(0, int(self.N)+1))
            P0_2 = 0
            for m in range(int(self.N)+1,int(self.M)+1):
                d=m-int(self.N)
                k=(self.λ**d)/math.prod((self.C*self.μ+i*self.γ) for i in range(1, m-
int(self.N)+1))
                P0_2+=k
            P0_2*=(self.λ/(self.C*self.μ))**(int(self.N))
```

```
            self.P0 = 1/(P0_1 + P0_2)
            return self.P0

        #Стационарное распределение вероятносей
    def prob_distribution(self):
        self.Pn = []
        for n in range(0, int(self.N)+1):
            self.Pn.append(((self.λ/(self.C*self.μ))**n)*self.P0)
        for n in range (int(self.N)+1, int(self.M)+1):
            Pn_2 = ((self.λ**(n-int(self.N)))*((self.λ/(self.C*self.μ))**(int(self.N)))*self.P0)/math.prod((self.C*self.μ+i*self.γ) for i in range(1, n-int(self.N)+1))
            self.Pn.append(Pn_2)
        return self.Pn

        #Среднее число заявок, находящихся на обслуживании
    def mean_quantity_service(self):
        Q_ser = sum(n*self.Pn[n] for n in range(0, int(self.N)+1)) + int(self.N)*sum(self.Pn[int(self.N)+i] for i in range (1, self.R+1))
        return Q_ser

        #Среднее число заявок, находящихся в очереди
    def mean_quantity_queue(self):
        Q_q = sum(n*self.Pn[int(self.N)+n] for n in range (1, self.R+1))
        return Q_q

        #Среднее число заявок, находящихся в системе
    def mean_quantity_system(self):
        Q_sys = sum(n*self.Pn[n] for n in range (0, int(self.N)+self.R+1))
        return Q_sys

    def bloking_prob(self):
        B = self.Pn[int(self.N)+int(self.R)]
        return B
```

2. Реализация программы и вывод графиков

1)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
from elastic import Elastic

C = 4
R = 2
b = 1
λ = 3
μ = 2
γ = 1

e = Elastic(C, R, b, λ, μ, γ)
print('Пространство состояний: ', e.X, 'Размер: ', e.len, sep='\n')

e.prob_zero()

Pn = e.prob_distribution()
for i in range (0, int(e.N)+R+1):
    print(f'p{i}=',Pn[i])

Pn[3]

Q_ser = e.mean_quantity_service()
Q_ser

Q_q = e.mean_quantity_queue()
Q_q

Q_sys = e.mean_quantity_system()
```

```
Q_sys

B = e.bloking_prob()
B
**********
#Задание новых исходных данных
C = 100
R = 40
b = 4
λ = 8
μ = 6
γ = 2

#изменение лямбды, те интенсивности поступления
I = range(0,1001)

#вычисление набора распределений через нулевую вероятность
zarr=[]
parr=[]
for i in I:
    e.λ = i
    zarr.append(e.prob_zero())
    e.P0 = zarr[i]
    parr.append(e.prob_distribution())
parr

Qser = []
Qq = []
Qsys = []
for i in I:
    e.Pn = parr[i]
    Qser.append(e.mean_quantity_service())
    Qq.append(e.mean_quantity_queue())
    Qsys.append(e.mean_quantity_system())
```

```
#построение графиков через сборку датафрейм
df = pd.DataFrame(I, columns=['λ'])
df['Qser'] = [Qser[i] for i in I]
df['Qq'] = [Qq[i] for i in I]
df['Qsys'] = [Qsys[i] for i in I]
print(df)

df.plot(x = 'λ', y = ['Qser', 'Qq', 'Qsys'],linewidth=2.0, linestyle='-' ,
      label=['на обслуживании/приборе','в очереди','в системе'])
plt.rcParams['font.size'] = '20'
plt.rcParams["font.family"]= 'Times New Roman'
plt.savefig('Quantities.png')

#изменение мю, те интенсивности обслуживания
I = range(1,61)

#далее повторение
zarr=[]
parr=[]
for i in I:
   e.μ = i
   zarr.append(e.prob_zero())
   e.P0 = zarr[i-1]
   parr.append(e.prob_distribution())
parr

Qser = []
Qq = []
Qsys = []
for i in I:
   e.Pn = parr[i-1]
   Qser.append(e.mean_quantity_service())
   Qq.append(e.mean_quantity_queue())
   Qsys.append(e.mean_quantity_system())
```

```
df = pd.DataFrame(I, columns=['μ'])
df['Qser'] = [Qser[i-1] for i in I]
df['Qq'] = [Qq[i-1] for i in I]
df['Qsys'] = [Qsys[i-1] for i in I]
print(df)
df.plot(x = 'μ', y = ['Qser', 'Qq', 'Qsys'],linewidth=2.0, linestyle='-' ,
      label=['на обслуживании/приборе','в очереди','в системе'])
plt.rcParams['font.size'] = '20'
plt.rcParams["font.family"]= 'Times New Roman'
plt.savefig('Quantities.png')

#изменение гаммы, те интенсивности ухода из очереди
I = range(0,101)

zarr=[]
parr=[]
for i in I:
   γ = i
   zarr.append(e.prob_zero())
   e.P0 = zarr[i]
   parr.append(e.prob_distribution())
parr

Qser = []
Qq = []
Qsys = []
for i in I:
   e.Pn = parr[i]
   Qser.append(e.mean_quantity_service())
   Qq.append(e.mean_quantity_queue())
   Qsys.append(e.mean_quantity_system())

df = pd.DataFrame(I, columns=['γ'])
```

```
df['Qser'] = [Qser[i] for i in I]
df['Qq'] = [Qq[i] for i in I]
df['Qsys'] = [Qsys[i] for i in I]
print(df)

df.plot(x = 'γ', y = ['Qser', 'Qq', 'Qsys'],linewidth=2.0, linestyle='-' ,
      label=['на обслуживании/приборе','в очереди','в системе'])
plt.rcParams['font.size'] = '20'
plt.rcParams["font.family"]= 'Times New Roman'
plt.savefig('Quantities.png')
```