

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
имени М. В. Ломоносова**

**Факультет вычислительной математики и кибернетики**

**Отчёт по заданию для спецкурса  
«Введение в сети передачи и обработки  
данных»**

**Оценка параметров качества сервиса (QoS) сетевого  
соединения**

**Студент: Китова Екатерина Денисовна  
Группа: 215**

Москва  
2025 год

# **Содержание**

<b>1 Постановка задачи</b>	<b>2</b>
<b>2 Введение</b>	<b>2</b>
<b>3 Реализация</b>	<b>3</b>
<b>4 Дополнительные задания</b>	<b>3</b>
<b>5 Результаты измерений</b>	<b>3</b>
5.1 Таблица результатов . . . . .	3
5.2 Графическая визуализация . . . . .	4
<b>6 Сравнение измеренных результатов с заданными параметрами сети</b>	<b>7</b>
<b>7 Анализ tshark</b>	<b>8</b>
<b>8 Анализ влияния параметров сети с использованием сетки значений</b>	<b>8</b>
<b>9 Заключение</b>	<b>8</b>

## 1 Постановка задачи

Необходимо разработать программу на C++ или Python, которая будет оценивать параметры качества сервиса (QoS) сетевого соединения, такие как:

- **Задержка (latency)** — измеряется с помощью утилиты `ping`.
- **Потери пакетов (packet loss)** — также измеряются с помощью `ping`.
- **Скорость (throughput)** — измеряется с помощью утилиты `iperf`.

Программа должна:

1. Ежесекундно измерять вышеуказанные параметры и собирать результаты для дальнейшего анализа.
2. Проводить статистический анализ собранных данных с вычислением среднего значения, медианы, 95-го процентиля, минимального и максимального значений, а также стандартного отклонения.
3. Выводить результаты анализа в виде таблицы на экран и сохранять их в файл (например, CSV или JSON).
4. Строить гистограммы для каждого измеряемого параметра с использованием библиотеки `matplotlib` или аналогичной.
5. Проводить эксперименты на эмуляторе сети Mininet, используя топологию из двух хостов (`h1` и `h2`) и одного коммутатора (`s1`). Параметры каналов должны быть настроены следующим образом: задержка 10 мс, потери пакетов 0,5%, пропускная способность 10 Мбит/с.
6. Запускать программу в течение одной минуты и собирать данные, после чего сравнивать полученные результаты с заданными параметрами сети.

Дополнительно, в программе должны быть реализованы:

1. Захват пакетов с фильтром "`tcp.analysis.retransmission`" для оценки повторных передач. Извлекаются временные метки, на основе которых строится линейный график повторных передач во времени.
2. Проведение эксперимента с перебором параметров сети (сетка значений задержки от 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 мс и потерь от 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0%) с построением тепловой карты зависимости пропускной способности от этих параметров.

## 2 Введение

В данном исследовании изучаются параметры QoS сетевого соединения, настроенного в Mininet с параметрами: задержка 10 мс, потери пакетов 0.5% и пропускная способность 10 Мбит/с. Необходимо собрать данные для анализа и оценить, насколько измеренные значения соответствуют ожидаемым, а также изучить влияние изменения параметров сети на пропускную способность.

### 3 Реализация

Эксперимент проводился в эмуляторе сети Mininet с топологией, состоящей из двух хостов (**h1** и **h2**) и одного коммутатора (**s1**). Программа реализована на Python 3.12 с использованием numpy, matplotlib, threading, time, statistics, csv, re, argparse и Mininet, OVSKernelSwitch, TCLink, setLogLevel, info, CLI из mininet. Код нашей программы и README находятся в архиве QOS.zip. Таблица результатов будет сохраняться с расширением CSV.

Тестирование проходило в виртуальной системе VirtualBox с ОС Ubuntu для ARM 24.04.2.

Измерения проводились следующим образом:

- **Ping:** Ежесекундно отправлял 1 пакет для измерения задержки и потерь.
- **Iperf:** Ежесекундно запускал тест пропускной способности.

Общее время эксперимента — 60 секунд.

### 4 Дополнительные задания

В рамках выполнения задания были реализованы следующие дополнительные задания:

1. **Анализ трафика с помощью tcpdump:** Захват пакетов с TCP Retransmission или TCP Fast Retransmission и построение графика повторных передач по времени из файла `tshark_output.txt`, если данные в нем присутствуют.
2. **Многопоточность:** Одновременное измерение задержки, потерь пакетов и пропускной способности с использованием двух потоков.
3. **Перебором параметров сети:** Измерение пропускной способности при варьировании задержки (2 мс, 4 мс, 6 мс, 8 мс, 10 мс, 12 мс, 14 мс, 16 мс, 18 мс, 20 мс) и потерь (0.2%, 0.4%, 0.6%, 0.8%, 1%, 1.2%, 1.4%, 1.6%, 1.8%, 2%) с построением тепловой карты. Для оптимизации работы этого случая, было принято решение проводить каждый эксперимент в течение 15 секунд. Так как при работе в течение 60 секунд каждого эксперимента были бы огромные накладные расходы и время работы скрипта ( $\approx 100$  минут).

## 5 Результаты измерений

### 5.1 Таблица результатов

Таблица 1: Результаты измерений QoS параметров (`qos_results.csv`)

Параметр QoS	Сред знач	Медиана	95%	Мин знач	Макс знач	Станд откл
Latency(ms)	45.9636	42.1	45.8	40.8	93.3	12.2449
Packet Loss(%)	0.0	0.0	0.0	0.0	0.0	0.0
Throughput(Mbits/sec)	7.909	8.76	9.06	4.67	9.07	1.4237

## 5.2 Графическая визуализация

Ниже приведены графики, построенные по результатам измерений:

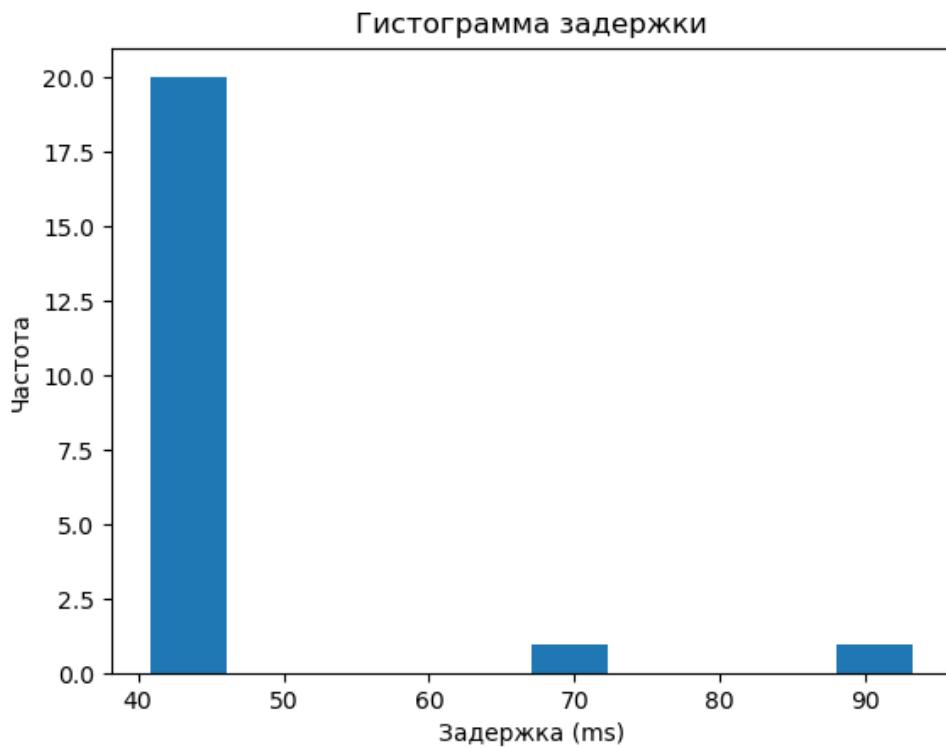


Рис. 1: Гистограмма задержки(latency\_histogram.png)

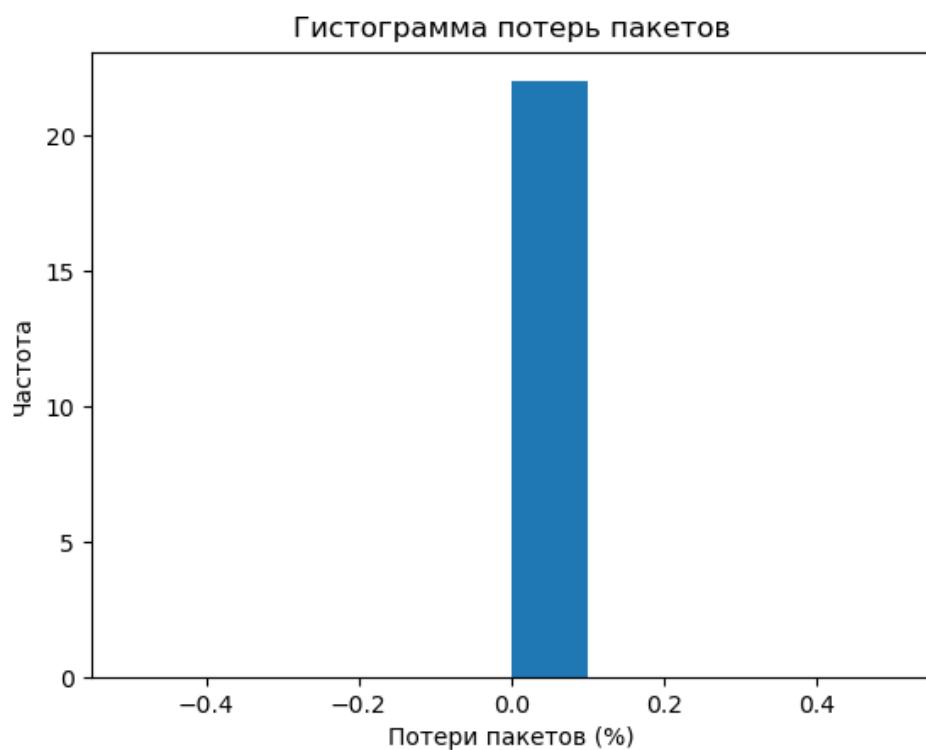


Рис. 2: Гистограмма потерь пакетов(`packet_loss_histogram.png`)

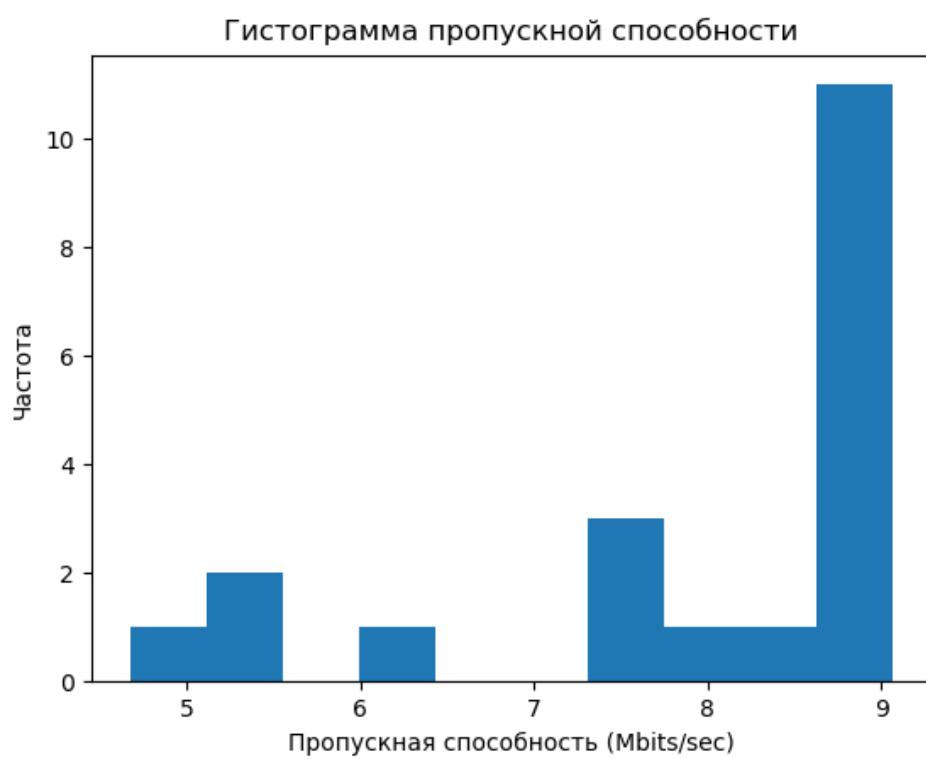


Рис. 3: Гистограмма пропускной способности(`throughput_histogram.png`)

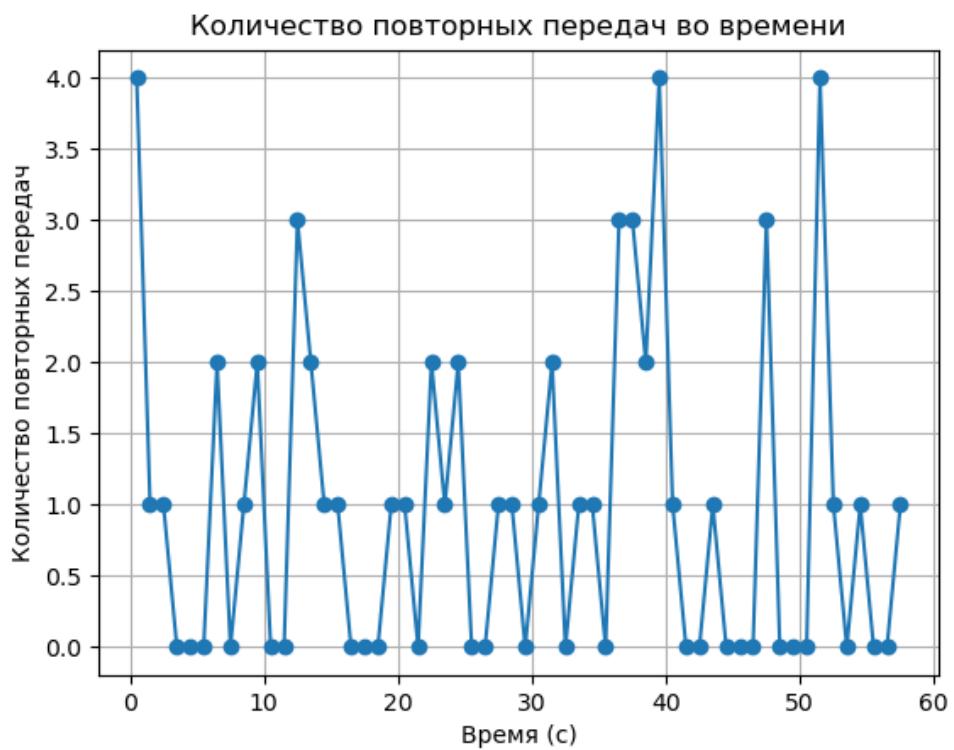


Рис. 4: Линейный график количества повторных передач во времени  
(retransmissions\_over\_time.png)

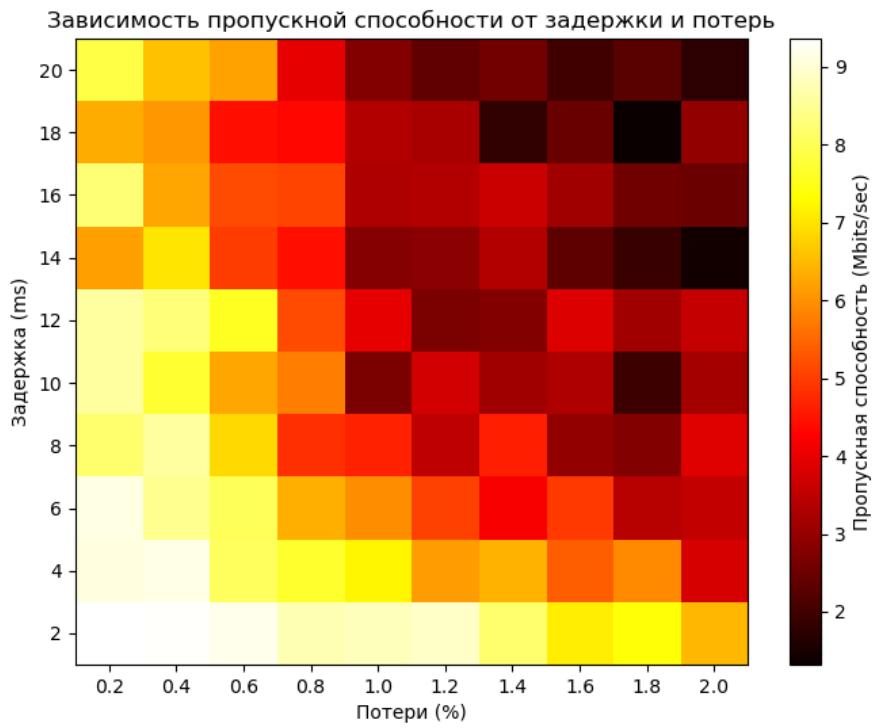


Рис. 5: Тепловая карта зависимости пропускной способности от задержки и потерь(throughput\_heatmap.png)

## 6 Сравнение измеренных результатов с заданными параметрами сети

В Mininet для каналов были заданы следующие параметры: задержка 10 мс, потери пакетов 0.5%, пропускная способность 10 Мбит/с. Сравним полученные результаты с этими настройками:

- **Задержка (Latency):**

Заданная задержка 10 мс относится к одностороннему времени. Однако, при измерении с использованием утилиты `ping` фиксируется round-trip time, которое примерно в два раза больше. Следовательно, полученные значения (медиана  $\approx 42$  мс, среднее  $\approx 46$  мс) соответствуют ожидаемому диапазону.

- **Потери пакетов (Packet Loss):**

При заданном уровне потерь 0.5% измерения показали 0% потерь, что также соответствует ожидаемому диапазону и показывает стабильность соединения.

- **Пропускная способность (Throughput):**

Заданная пропускная способность составляет 10 Мбит/с, однако измеренные значения оказались ниже (среднее  $\approx 7.9$  Мбит/с). Снижение пропускной способности связано с дополнительными накладными расходами, задержкой, потерей пакетов и особенностями эмуляции в виртуальной среде.

Таким образом, хотя измеренные значения не совпадают точно с номинальными, они находятся в ожидаемых пределах с учётом всех накладных расходов.

## 7 Анализ tshark

Анализ tshark позволил оценить наличие пакетов с повторными передачами (TCP Retransmission или TCP Fast Retransmission) из файла `tshark_output.txt` с помощью фильтра "`tcp.analysis.retransmission`". В нашем случае они присутствовали, но не так много из-за низких значений потерь пакетов.

## 8 Анализ влияния параметров сети с использованием сетки значений

На тепловой карте заметно, что при минимальных значениях задержки и потерь достигается максимальная пропускная способность. Если один из параметров начинает расти, при этом второй остается минимальных, пропускная способность остается достаточно высокой. Но при одновременном росте потерь и задержки, пропускная способность начинает значительно падать.

## 9 Заключение

Проведённое исследование QoS показало, что исследуемые параметры находятся в допустимых значениях. Потери пакетов практически не фиксировались, что указывает на стабильность соединения. Исследуя зависимость пропускной способности от задержки и потерь, мы убедились, что оптимальные параметры, то есть низкая задержка и минимальные потери, способствуют достижению максимальной пропускной способности. При ухудшении качества канала пропускная способность падает.