

# Доклад по проекту QoS

---

## Описание проекта

Проект создан для измерения параметров Quality of service, таких как:

- задержка (с помощью `ping`)
- потери пакетов (с помощью `ping`)
- скорость соединения (с помощью `iperf`)

Скрипты на программе `python` запускают необходимые программы каждую секунду в течение заданного времени, анализируют данные и строят графики с помощью библиотеки `matplotlib` в директории `output`

Для каждого параметра QoS программа выводит в виде таблицы значения со следующими столбцами:

- Параметр QoS
- Среднее значение
- Медиана
- 95-й процентиль
- Минимальное значение
- Максимальное значение
- Стандартное отклонение

## Перед запуском

Перед началом запуска скриптов необходимо создать виртуальное окружение для `python` и установить необходимые модули.

```
python -m venv ./.venv
source .venv/bin/activate
pip3 install -r requirements.txt
```

а также для выполнения `sudo`-команд из под виртуального пространства:

```
PYTHON3_VENV_PATH=$(which python3)
sudo $PYTHON3_VENV_PATH some_script
```

## Структура проекта

Проект состоит из следующих основных скриптов:

`create.py`

Скрипт создаёт сеть **mininet** с двумя хостами и одним коммутатором. Параметры для каждого канала:

- Задержка: 10 мс
- Потери пакетов: 0,5%
- Пропускная способность: 10 Мбит/с

После выполнения для h2 команду **iperf -s** и включает для h1 виртуальное окружение **python** (должно быть создано изначально). *Необходимы права root для создания сети.*

### **Пример запуска**

```
sudo $PYTHON3_VENV_PATH create.py
```

### **main.py**

Скрипт запускает перечисленные выше программы и сохраняет данные в output директорию (создаёт её при отсутствии). По умолчанию запускает на 60 секунд, но можно передать время с помощью аргумента командной строки. Хост по умолчанию - **10.0.0.2** (базовый ip для h2 в сети mininet. см ниже)

### **Пример запуска в сети mininet**

```
h1 python3 main.py <duration>
```

### **extras/extra.py**

Скрипт выполняет роль аналогичную **main.py**, но:

1. все измерения задержки, потерь и скорости производятся в мультипоточном режиме.
2. в отдельные файлы вынесены классы отвечающие за свои подзадачи
3. добавлена возможность анализа повторных передач с помощью **tshark**. (при запуске **grip** указывается соответствующий для этого параметр)

### **Пример запуска в сети mininet**

```
h1 python3 extras/extra.py
```

### **extras/extra\_create.py**

Скрипт анализирует пропускную способность сети для каждой комбинации параметров задержки и потери пакетов создавая различные сети mininet. Выводит собранные данные в виде тепловой карты. *Необходимы права root для создания сети.*

Список параметров (пропускная способность остаётся равной 10 mbit/s):

- Задержка (latency): 2 мс, 4 мс, 6 мс, 8 мс, 10 мс, 12 мс, 14 мс, 16 мс, 18 мс, 20 мс.
- Потери пакетов (packet loss): 0,2%, 0,4%, 0,6%, 0,8%, 1%, 1,2%, 1,4%, 1,6%, 1,8%, 2%.

### **Пример запуска**

**! Скрипт выполняется ~10 минут !**

```
sudo $PYTHON3_VENV_PATH extras/extra_create.py
```

## Собранные данные

Задержка

![[output/latency\_(ms)\_histogram.png]]

Потеря пакетов

![[output/packet\_loss\_histogram.png]]

Пропускная способность

![[output/throughput\_(mbs)\_histogram.png]]

Количество повторных пакетов во времени

![[output/retransmissions\_plot.png]]

Тепловая карта скорости от задержки и потери пакетов

![[output/heatmap.png]]

## Заключение

В ходе выполнения проекта была создана система для измерения параметров QoS, которая позволяет анализировать задержку, потери пакетов и пропускную способность сети. Разработанные скрипты обеспечивают автоматизацию процесса сбора данных, их анализа и визуализации.

Система успешно протестирована в различных условиях, включая вариации задержки и потерь пакетов. Полученные результаты демонстрируют возможность использования разработанного инструмента для оценки качества сетевых соединений и выявления узких мест в сети.

Дальнейшее развитие проекта может включать:

- Добавление поддержки дополнительных параметров QoS.
- Интеграцию с другими инструментами анализа сетей.
- Оптимизацию производительности скриптов для работы в реальном времени.

## Используемые материалы

Работа с mininet

- [https://asvk.cs.msu.ru/wp-content/uploads/2023/04/2022-10-10-CN\\_lecture\\_1.pdf](https://asvk.cs.msu.ru/wp-content/uploads/2023/04/2022-10-10-CN_lecture_1.pdf) (с 20ой страницы)

## Работа с ipref

- <https://yamadharma.github.io/ru/teaching/simulation-networks/lab/mininet-iperf-intro/>
- <https://stackoverflow.com/questions/63907768/use-iperf-and-ping-at-the-same-time-mininet>