

# Маршрутизация в компьютерной сети с применением глубокого обучения

Панчишин Иван Романович, группа М41381с

2021-06-11

## Запуск экспериментов

Склонировать репозиторий:

```
git clone https://github.com/vpunch/routesim
```

Установить пакет и зависимости в тестовое окружение:

```
python -m venv routesim-env
. routesim-env/bin/activate

cd routesim/src
python setup.py develop
pip install -r ../requirements.txt

pip install ipykernel
python -m ipykernel install --user --name=routesim-env
```

Открыть ноутбук:

```
cd ../notebooks
jupyter notebook
```

Запустить симуляцию при помощи файла: `run_simulation.ipynb`.

## Задача 1

Для реализации проекта был выбран язык программирования Python. Для построения графов используется библиотека `networkx`, для симуляции компьютерной сети — `simpy`.

Компьютерная сеть представлена в виде графа, где в каждом узле находится маршрутизатор.

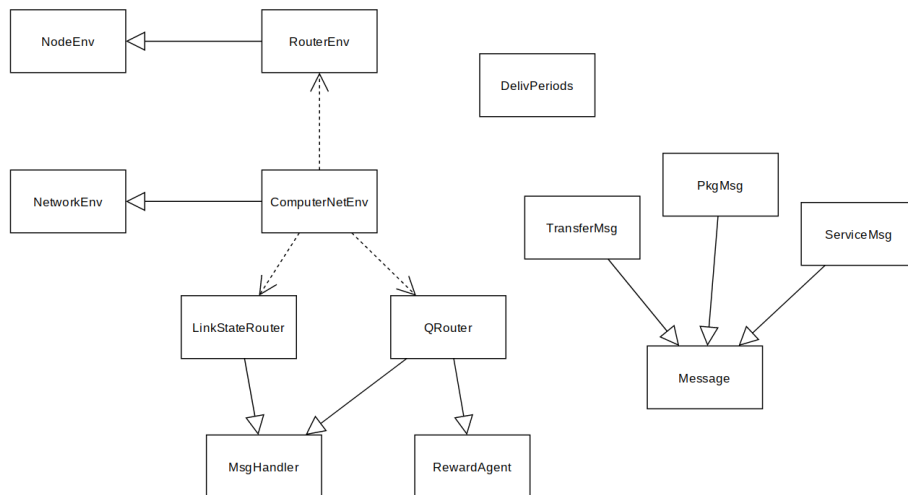


Рис. 1: Диаграмма классов

2 основных типа сообщений:

- Сообщение с пакетом (**PkgMsg**), перемещение которого по сети исследуется.
- Служебное сообщение (**ServiceMsg**), которым может быть анонс состояний в алгоритме link-state или награда в алгоритме Q.

Класс **DelivPeriods** используется для агрегации данных во время симуляции сети.

## Задача 2

Алгоритм реализован в классе **LinkStateRouter**.

## Задача 3

В терминах обучения с подкреплением, маршрутизатор является агентом, сеть — окружением, выбор соседнего узла — действием.

Алгоритм реализован в классе **QRouter**.

## Задача 4

Эксперименты проводились в сети, которую можно представить следующим графом:

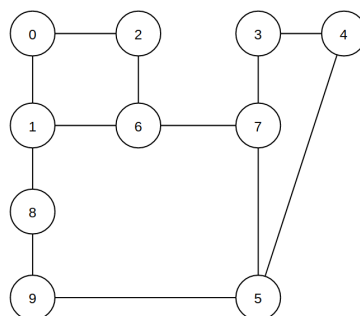


Рис. 2: Структура сети

Симуляция сети выполнялась не в реальном времени, поэтому вместо секунд далее будут указываться шаги.

Каждое ребро графа является соединением между двумя роутерами и имеет вес, который является пропускной способностью этого соединения.

- Каждое соединение имеет пропускную способность 124 байт/шаг,
- Каждый пакет имеет размер 1024 байт.
- Обработка каждого пакета роутером длится 5 шагов.

### Адаптация к изменению нагрузки на сеть

Сценарий отправки пакетов:

- 100 пакетов с задержкой 10 шагов между любыми узлами,
- затем 500 пакетов с задержкой 10 шагов от узлов 0, 1, 2, 6 к узлам 3, 4, 5, 7.
- затем 1500 пакетов с задержкой 5 шагов между теми же узлами,
- затем 500 пакетов каждые 10 шагов между теми же узлами.

Результат симуляции приведен на рисунке 3.

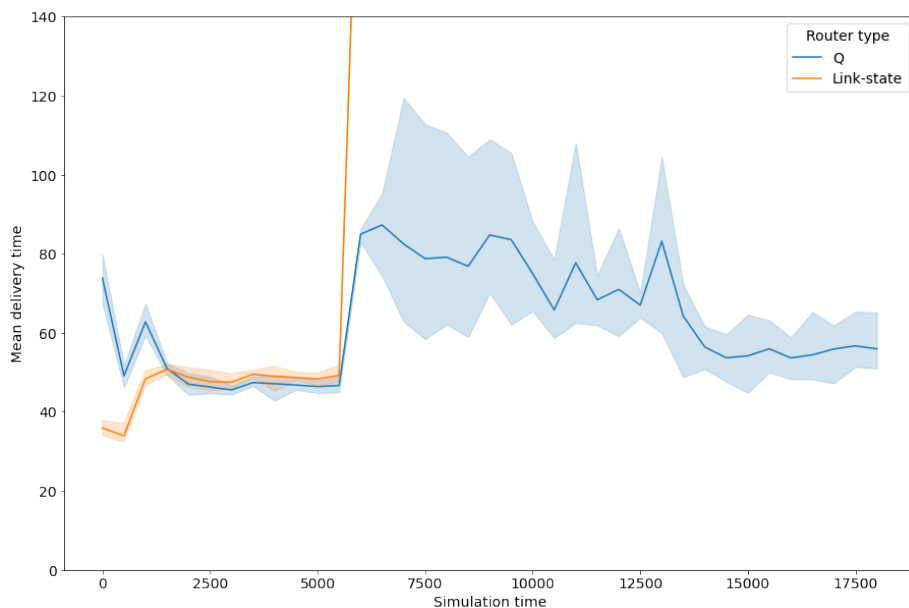


Рис. 3: Изменение нагрузки

С алгоритмом link-state наблюдается быстрый рост среднего времени доставки пакета в момент повышения нагрузки на сеть. Это связано с тем, что алгоритм выбирает один и тот же кратчайший путь между парами узлов, основываясь только на пропускной способности соединений, но не учитывая их загруженность. Алгоритм Q учитывает время передачи пакета, что позволяет изменять путь маршрутизации между парой узлов.

## Обрыв соединений

Сценарий отправки пакетов:

- 100 пакетов с задержкой 10 шагов между любыми узлами,
- затем 3500 пакетов с задержкой 10 шагов от узлов 0, 1, 2, 6 к узлам 3, 4, 5, 7.

Соединения (6, 7), (0, 1), (4, 5) последовательно обрываются спустя каждые 500 пакетов, начиная со 101 пакета, затем восстанавливаются в том же порядке, с тем же интервалом.

Результат симуляции приведен на рисунке 4.

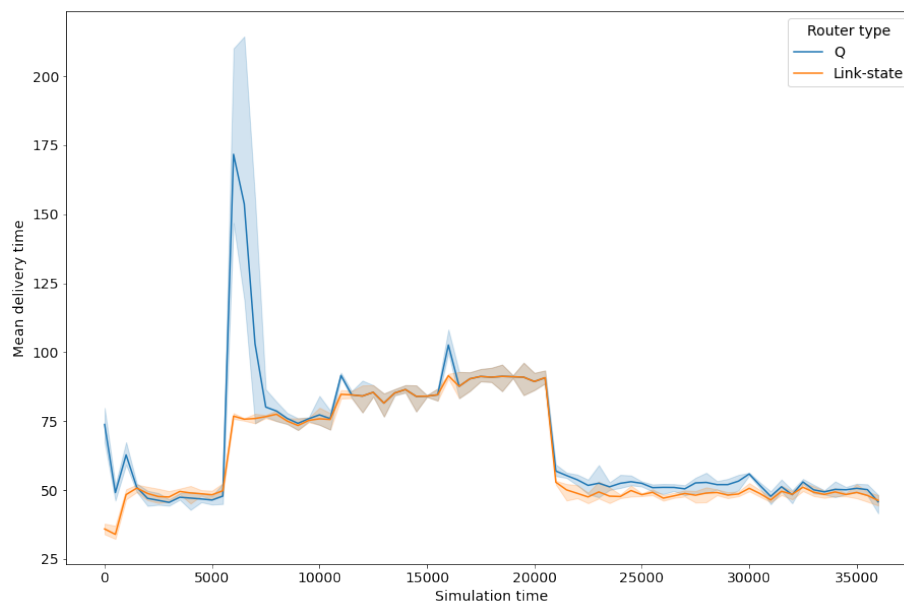


Рис. 4: Обрыв соединений

В алгоритме link-state маршрутизатор обновляет карту сети при каждом изменении в своих сетевых интерфейсах. Так как служебные сообщения передаются моментально, алгоритм мгновенно адаптируется к обрыву соединения. Алгоритму Q требуется время для обучения, поэтому на графике наблюдается скачки среднего времени доставки в момент обрывов и длительный возврат к оптимальной работе в момент восстановления соединений.