



УНИВЕРСИТЕТ ИТМО

Глубокие самообучающиеся агенты для мультиагентной системы маршрутизации

Мухутдинов Дмитрий, группа М3438

Научный руководитель: Фильченков А. А., к.ф.-м.н., доцент кафедры КТ

Рецензент: Тарасов В. Б., к.т.н., МГТУ им. Н.Э. Баумана



Задача маршрутизации

Поиск пути наименьшей стоимости из точки А в точку Б в распределенной сети

- Сетевой роутинг
- Транспортная логистика
- Управление конвейерными системами
- Автоматическое управление городским трафиком



Существующие решения

- Link-state
 - Open Shortest Path First (OSPF)
 - IS-IS
- Distance-vector
 - RIP
 - IGRP
- Прочие
 - AntNet
 - ...



Проблемы

- Примерно все алгоритмы маршрутизации заточены под компьютерные сети
- В других задачах существуют свои, более сложные условия
 - Мотоцикл проедет сквозь пробку, а автомобиль - нет
 - Чемоданы бизнес-класса хочется доставить первыми
 - ...



Задача

- Построить алгоритм, способный адаптироваться под более широкий класс условий



Идея

- Обучение с подкреплением
- Нейросети в качестве обучающихся агентов
- Q-routing (Boyan & Littman, 1994)

Постановка задачи в терминах RL

- Рассматриваем пакет в сети как обучающегося агента, взаимодействующего с сетью как с средой
- Полное состояние среды неизвестно, состояние текущего роутера — *наблюдение* пакета
- Действие — переход к одному из соседей
- Q-learning:

$$Q(o_t, a_t) \leftarrow r_t + \gamma \cdot \max_{a \in \mathcal{A}_{o_{t+1}}} Q(o_{t+1}, a)$$

- Принцип аналогичный Q-routing:

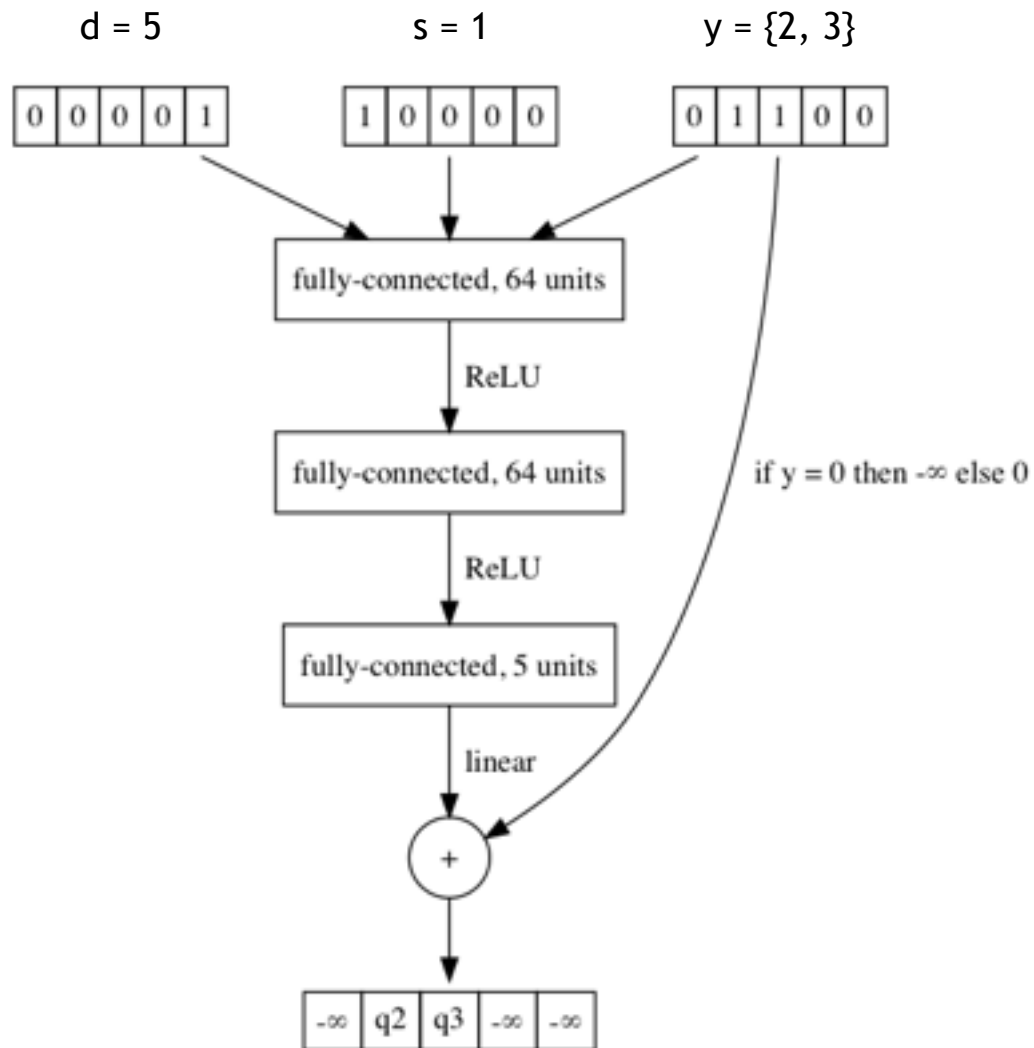
$$Q_x(d, y) \leftarrow (t_{start} - t_{finish}) + \max_{z \in \{V | (y, z) \in E\}} Q_y(d, z)$$



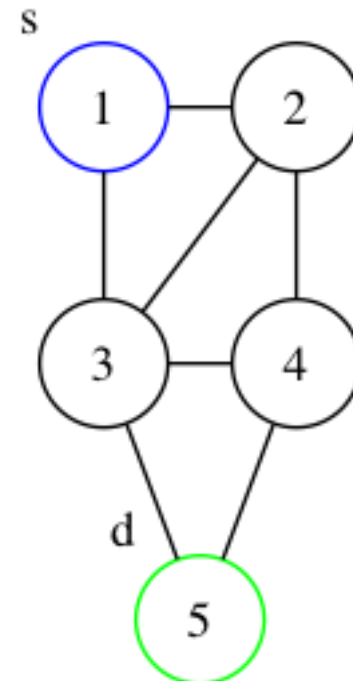
Базовая архитектура нейросети

Вход нейросети $o = (d, s, y_1 \dots y_n, o')$, где:

- d — узел назначения, s — текущий узел, $y_1 \dots y_n$ — номера соседей, o' — любая дополнительная информация
- $a_1 \dots a_n$ — оценки $Q(s, a)$ для всех узлов сети ($-\infty$ для узлов, не являющихся соседями)
- Кодлируем номера унитарным кодом, чтобы избежать корреляции результатов
- Используем RMSProp для оптимизации



Пример сети для работы в
графе из 5 вершин



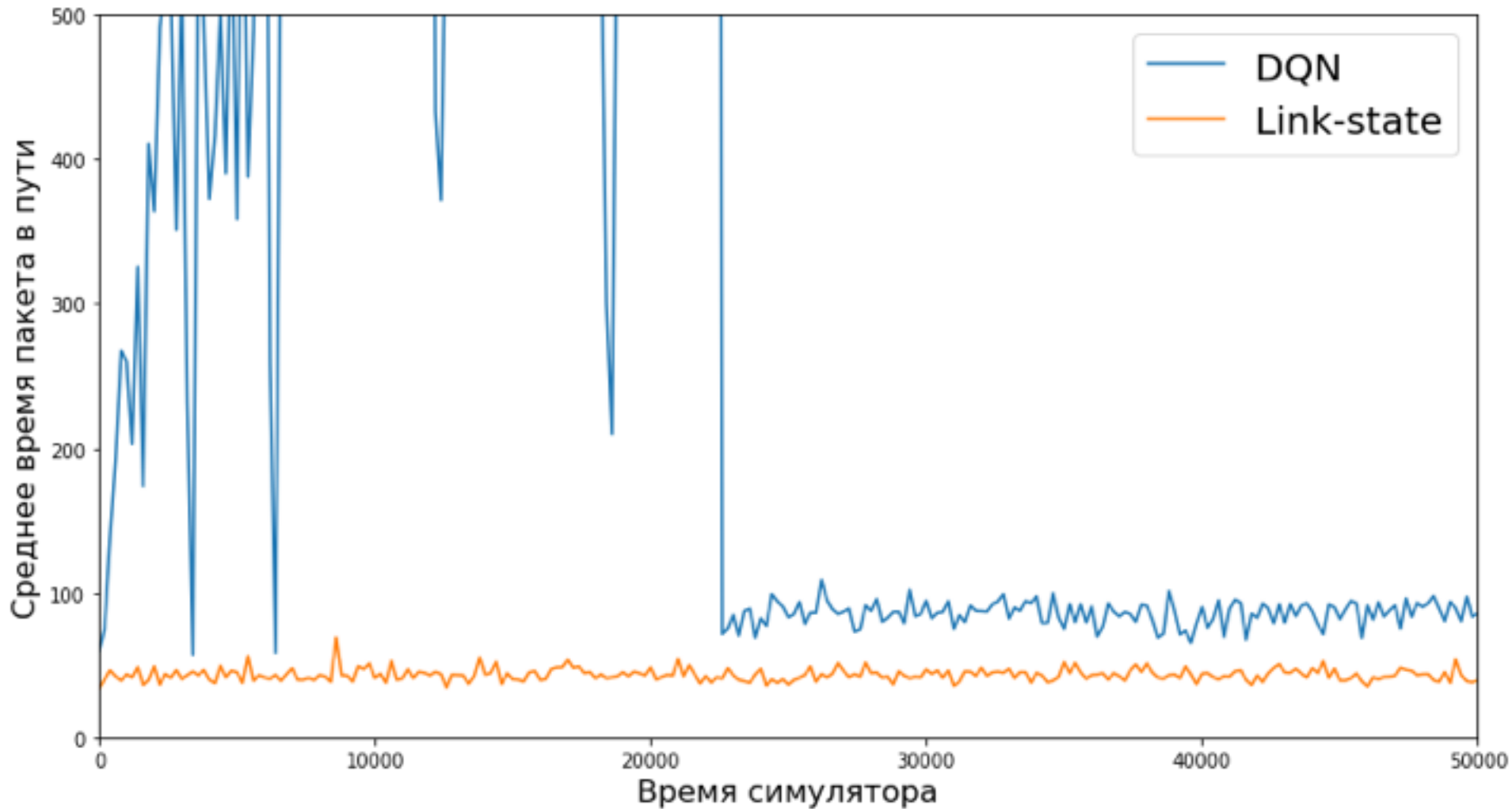


Проблема нестационарности (1/3)

- Q-обучение, вообще говоря, не сходится к оптимальному решению с нейросетями
- Другие агенты меняют свое поведение — среда нестационарна, вследствие чего experience replay (Mnih et al, 2015) не работает
- Обучение с нуля не сходится к оптимальному решению



Запуск RL без инициализации





Проблема нестационарности (2/3)

Предобучение сети (bootstrapping)

- Соберем данные работы алгоритма кратчайших путей
- Обучаем одну нейросеть на данных от всех роутеров
- Предобученная сеть повторяет работу алгоритма кратчайших путей



Проблема нестационарности (3/3)

Отказ от experience replay во время работы

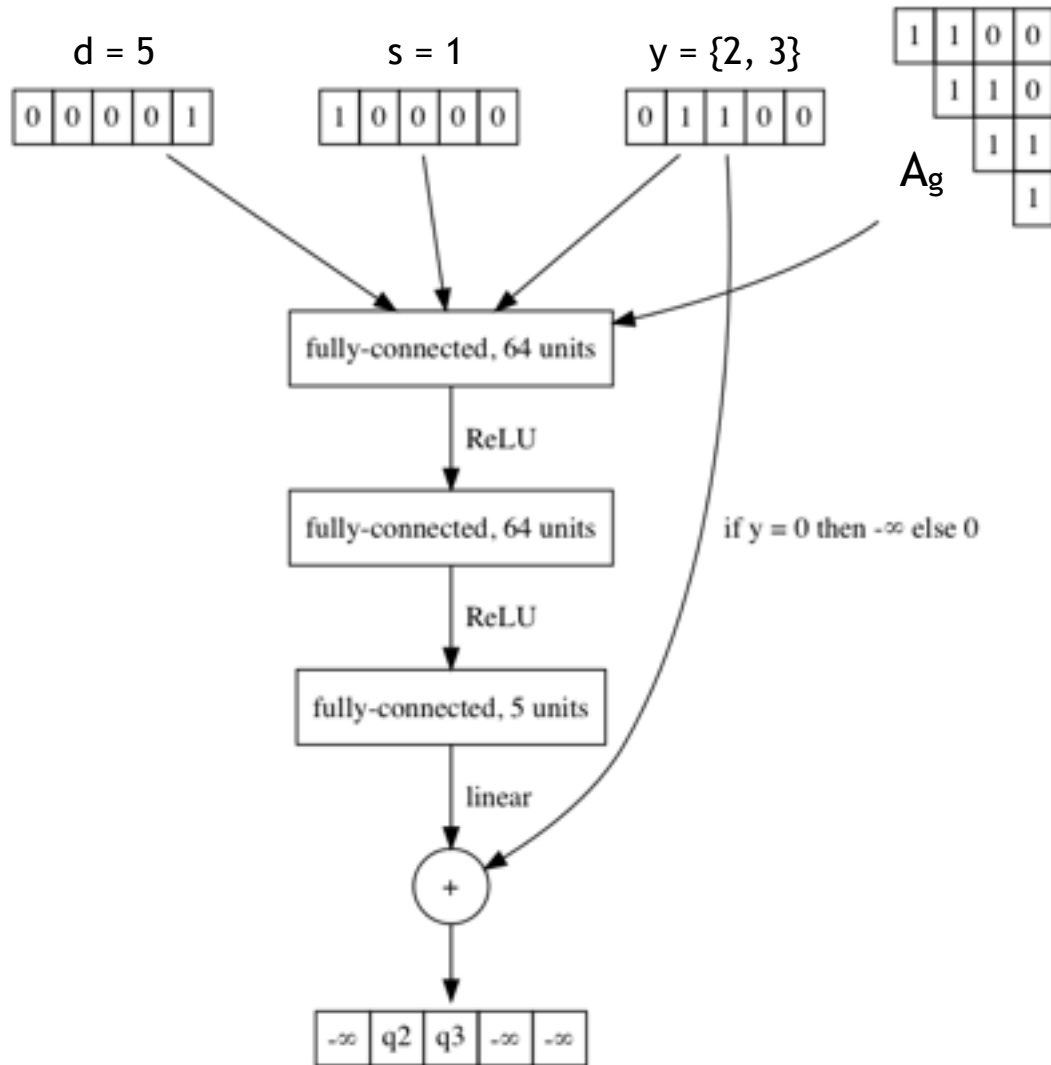
- Условия работы в сети меняются
- Старый опыт перестает быть актуальным
- Если обращать на него внимание, адаптивность к изменению нагрузки страдает



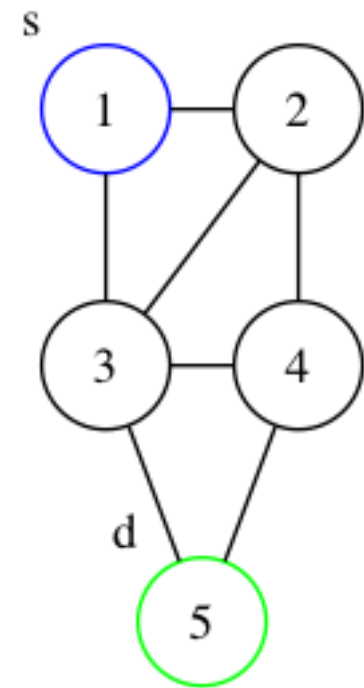
Расширение наблюдаемого состояния

Какую информацию полезно учесть при роутинге?

- Топологию сети полезно знать
- Используем link-state протокол, чтобы иметь информацию о топологии
- Подаем матрицу смежности A_g на вход нейросети



Сеть с матрицей смежности для того же графа



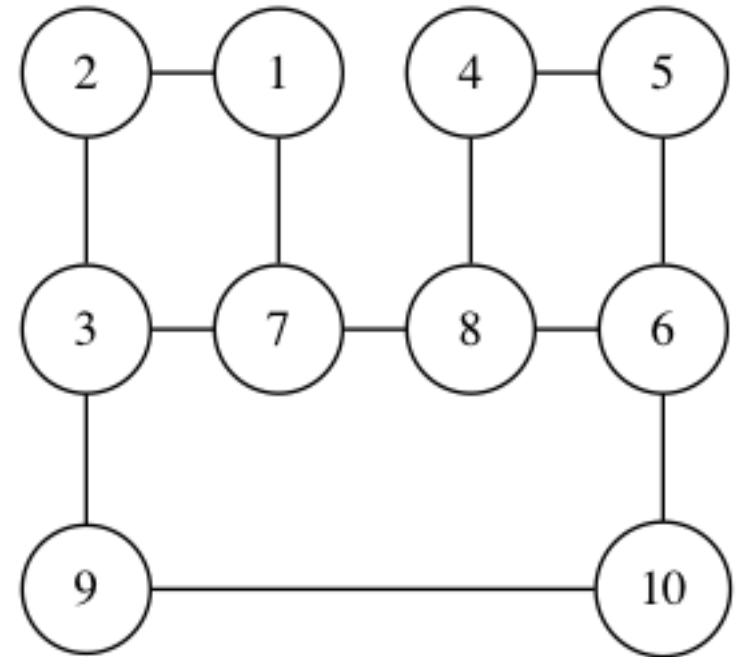


Компромисс между исследованием и использованием

- Алгоритм застревает в локальном максимуме
- Особенно заметно при снижении нагрузки с пиковой на обычную
- Будем не всегда выбирать наиболее выгодное с текущей точки зрения действие
- Берем softmax от выходов нейросети
- Выбираем случайное действие из распределения, заданного результатом softmax

Эксперименты

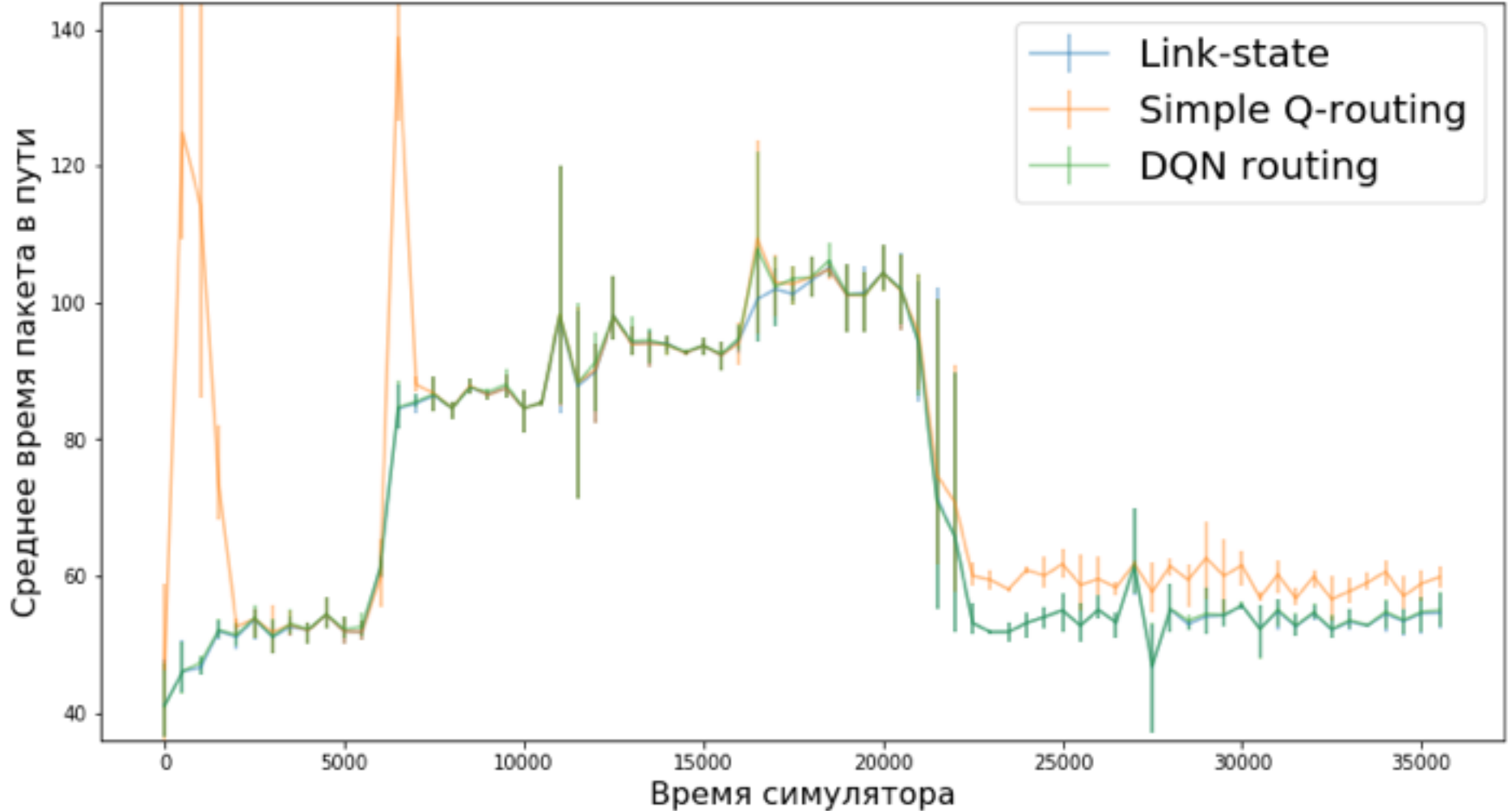
- Симуляция времени работы сети
- Служебные сообщения передаются бесплатно
- Производим по три запуска сценария для каждого алгоритма с разными random seed и усредняем



Граф для проведения тестов

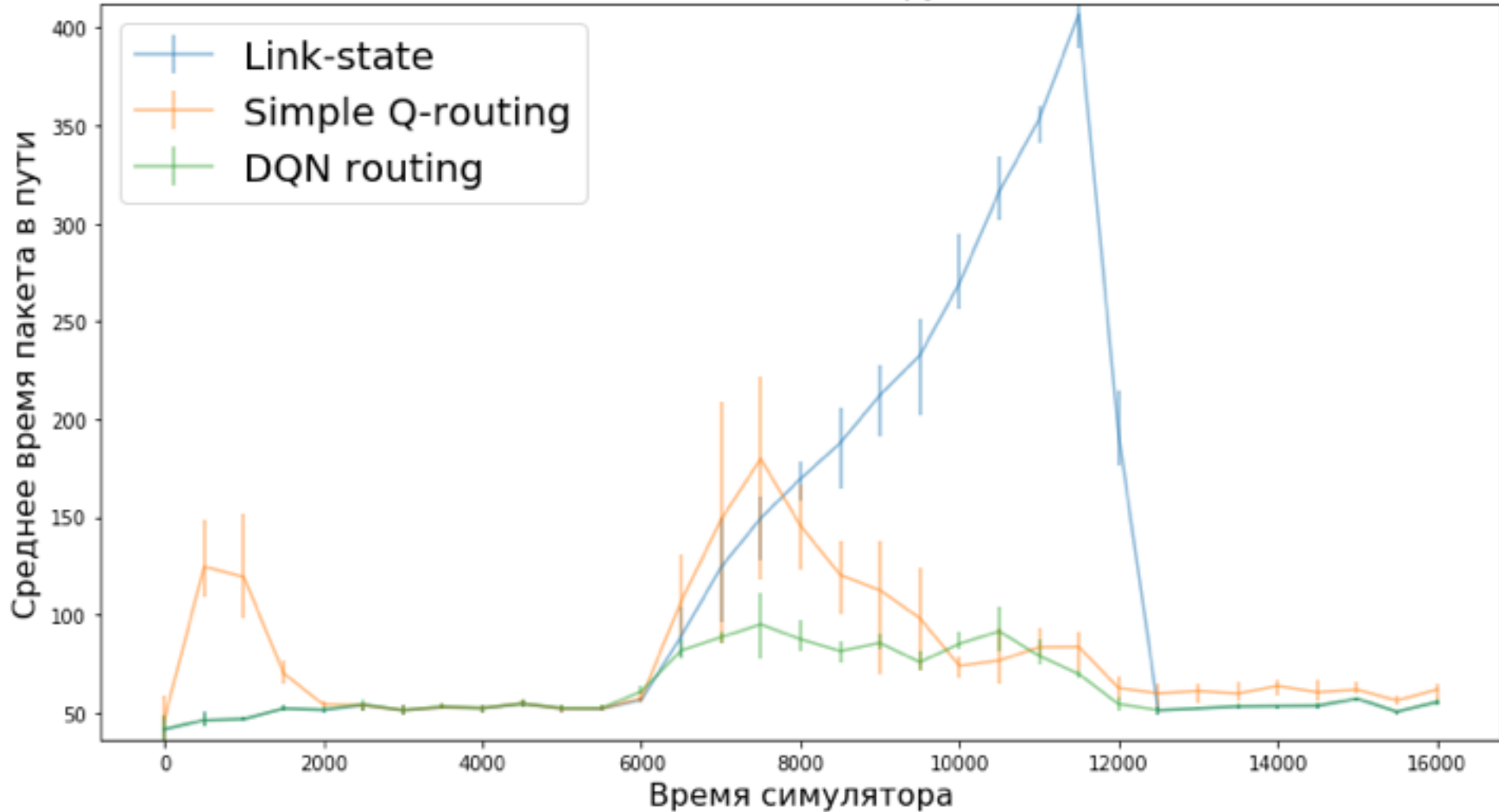


Обрыв трех ребер одно за другим с последующим восстановлением в обратном порядке



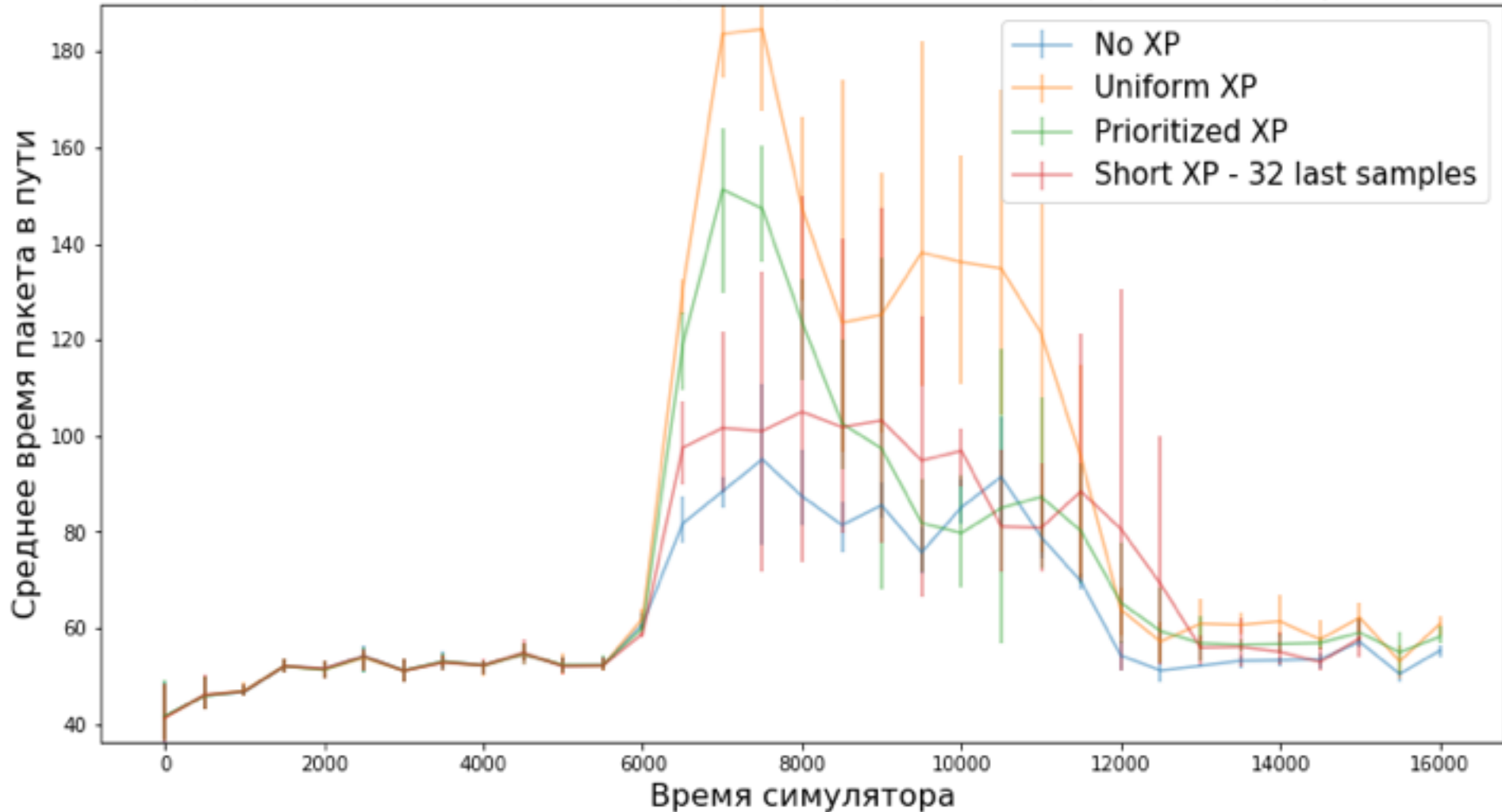


Резкий скачок нагрузки





Сравнение адаптивности алгоритма с применением Experience Replay и без





Идеи продолжения работы

- Эксперименты по расширению состояния
- Эксперименты в усложненной среде (e.g. имитация багажных конвейеров)
- Применение рекуррентных нейросетей
- Стабилизация experience replay методом отпечатков (Foester & Nardelli, 2017)
- Векторное представление номеров узлов
- Вторичное предобучение в полных условиях среды с использованием метода A3C (Mnih et al., 2016)

Спасибо за внимание!