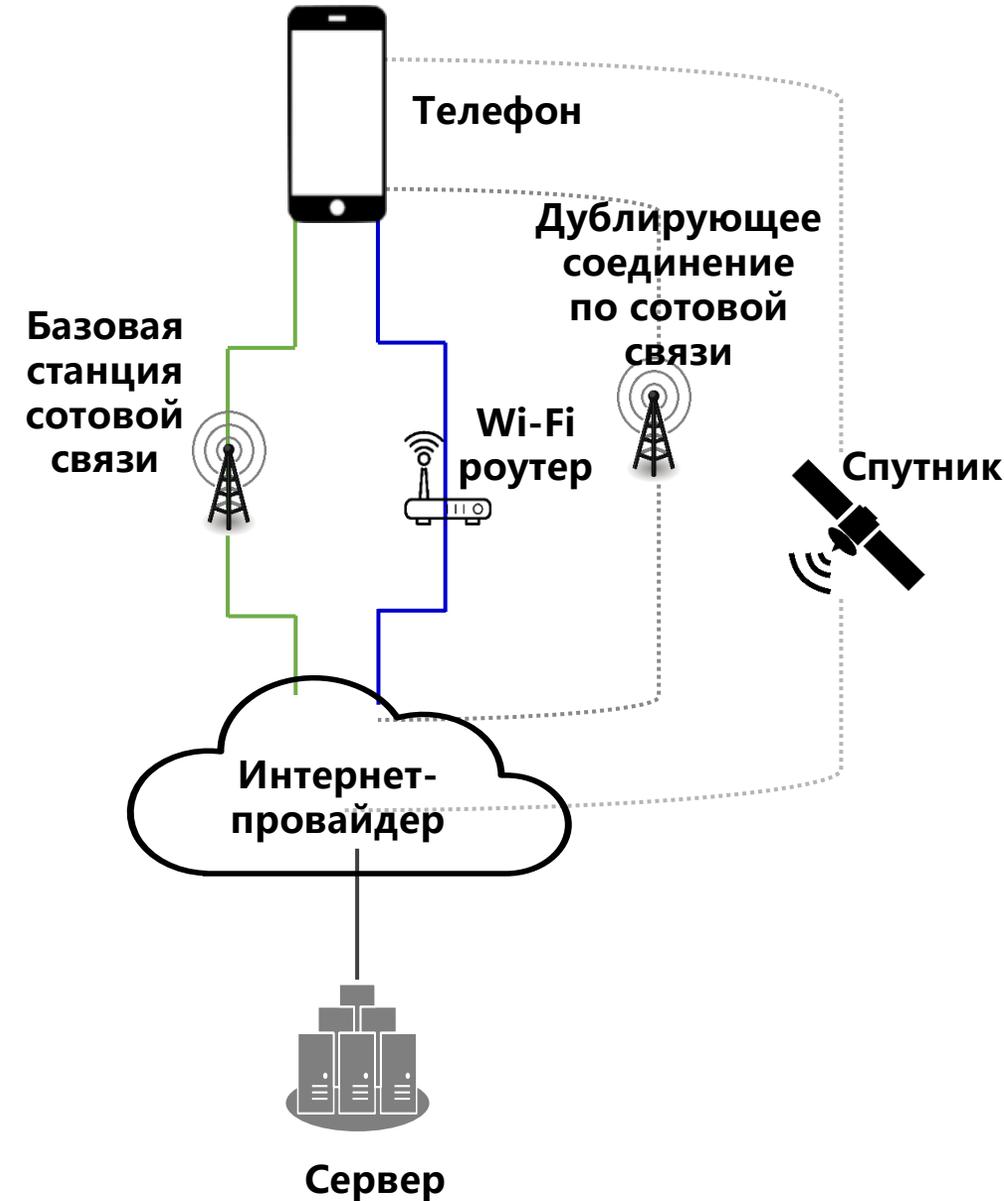


Управление потоками в многоканальных гетерогенных сетях с использованием систем обучения с подкреплением

Д.Москвитин ,
Аспирант ИТМО,
2025

Актуальность задачи

- **Многопутевая передача** — одновременная передача по нескольким сетевым путям.
- **Самый общий случай** — одновременная работа Wi-Fi и сотовой связи (LTE/5g).
- **Отсутствие алгоритмов многокритериальной оптимизации под мобильные устройства:**
 - Минимизация задержки (latency).
 - Максимизация пропускной способности (throughput).
 - Снижение энергопотребления (energy efficiency).
 - Увеличение надежности передачи данных (reliability).



Существующие решения

Статья/Решение	Достоинства	Недостатки
MPTCP: A Multipath Transport Protocol	Улучшает пропускную способность, надежность, имеет обратную совместимость.	Высокая сложность реализации, неспособность противодействовать неупорядоченной доставки пакетов.
Multipath Scheduling for Video Streaming over MPTCP	Оптимизировано для потокового видео, динамическая адаптация.	Высокие вычислительные затраты, мониторинг на базе фиксированных временных интервалов.
BALIA: Balanced Linked Adaptation for MPTCP	Улучшает пропускную способность и скорость реагирования.	Требует настройки, плохо адаптируется к динамическими средами.
Energy-Efficient Multipath Scheduling for Mobile Devices	Минимизирует потребление энергии, балансирует трафик. Компромисс между энергоэффективностью и производительностью.	Экспериментальный. Не адаптирован к мобильным устройствам. Оптимизирует только передачу файлов.
Multipath QUIC: Design and Evaluation	Декларирует расширение протокола QUIC для многоканальной передачи данных .	Ограниченное развертывание, требуются изменения в QUIC.
XLINK: QoE-Driven Multi-Path QUIC Transport in Large-scale Video Services	Уменьшает задержку (minRTT), поддерживает плавную передачу обслуживания. Широко применяется в индустрии. Оптимизирован под потоковый тип передачи данных	Не принимает во внимание мобильный характер передачи данных - оптимизация энергопотребления и характера передачи данных (сообщение, файл, потоковое видео)
Reinforcement Learning-Based Multipath QUIC Scheduler	Адаптируется к динамическим условиям, оптимизирует потоковую передачу мультимедиа.	Высокие накладные расходы на обучение/выводы, ограниченная обобщаемость, мониторинг на базе фиксированных временных интервалов. Не принимает во внимание характер передачи данных (сообщение, файл, потоковое видео)

Научная новизна

1. Применение глубокого обучения для многопутевой передачи данных в гетерогенных сетях:

- В отличие от традиционных подходов, которые используют статические правила или эвристики для выбора пути, в проекте предлагается использовать алгоритмы **глубокого обучения** (например, Deep Reinforcement Learning, DRL) для адаптивного управления передачей данных.

2. Мониторинг сети на основе временных и объемных интервалов:

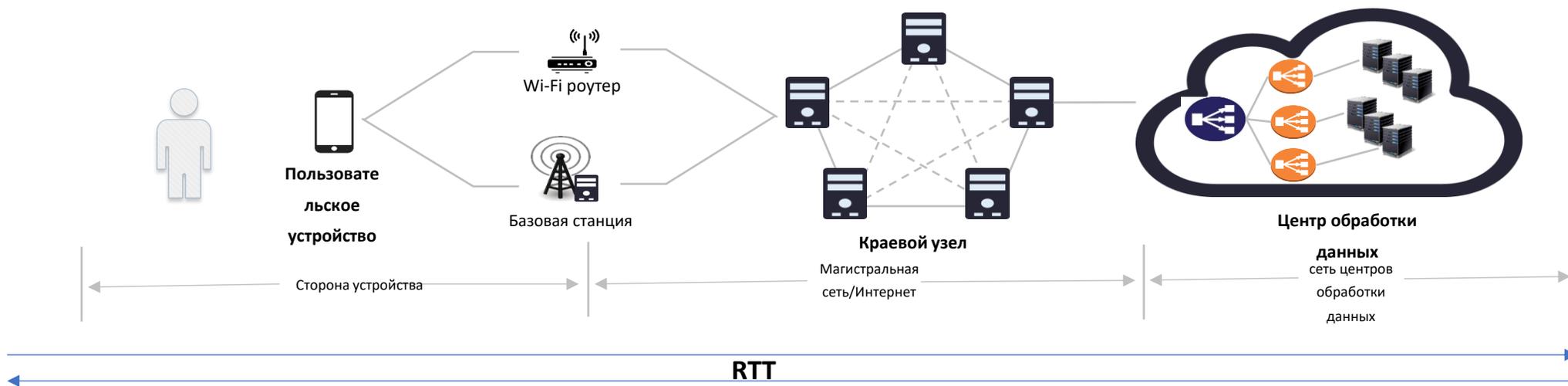
- В проекте предлагается **новый подход к мониторингу сети**, который сочетает анализ данных как по временным интервалам (например, каждые N мс), так и по объемным интервалам (например, после передачи каждых M КБ данных).

3. Разработка гибридных моделей:

- В проекте предлагается использовать **гибридные модели**, которые сочетают глубокое обучение с традиционными методами (например, minRTT или Redundant Scheduling (ReMP)). Это позволяет использовать преимущества обоих подходов:
 - Глубокое обучение для адаптации к динамическим условиям.
 - Традиционные методы для обеспечения стабильности и простоты реализации.

Введение: контроль перегрузки E2E и диспетчеризация пакетов

- Контроль перегрузки — это контрольная деятельность всех сетевых устройств.
- Его можно реализовать в разных масштабах и на разных уровнях.
- Наше внимание сосредоточено на сквозном контроле перегрузок в сетевых путях.

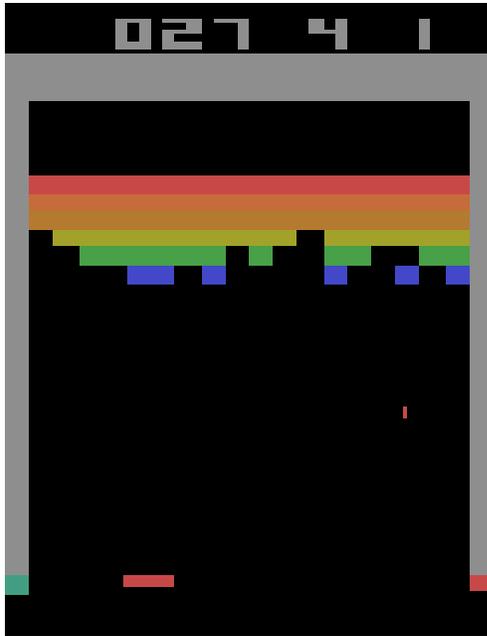


- Модуль управления перегрузкой E2E обычно называется алгоритмом отправки.
- Расположен на уровне устройства и работает со следующими метриками: время приема-передачи (RTT), пропускная способность, уровень потерь.
- Цель – **передача данных с максимальной скоростью и минимальными потерями.**

Обучение с подкреплением | Обзор

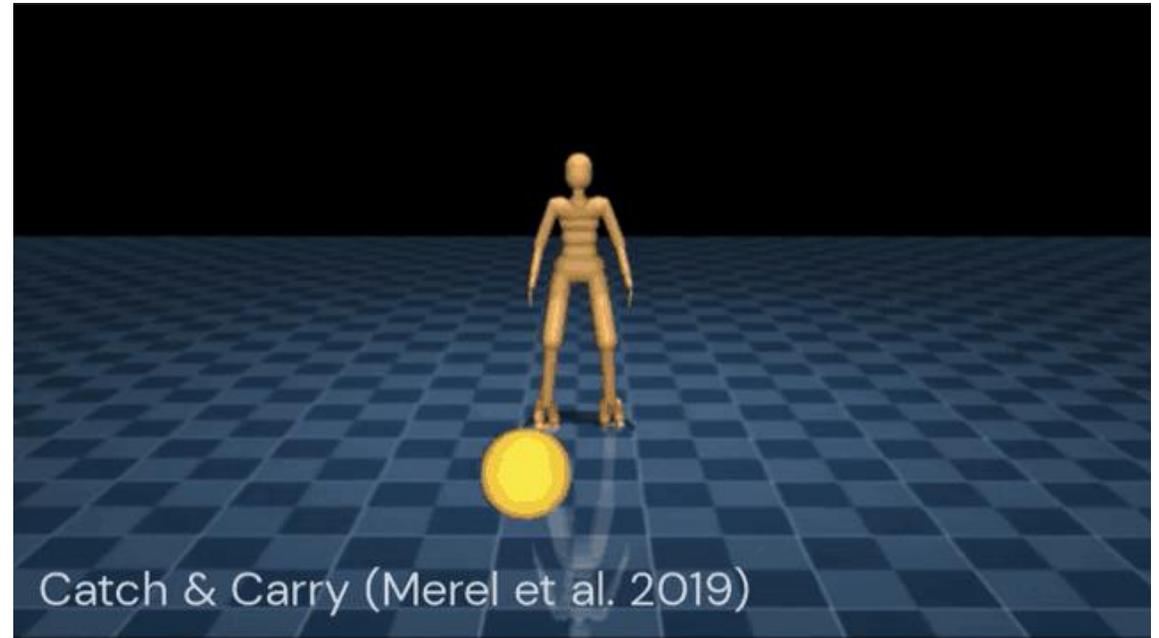
Дискретное пространство действий

- **DQN** (Мних и др., 2015)
- **Двойной DQN** (ван Хасселт и др., 2015 г.)
- ...

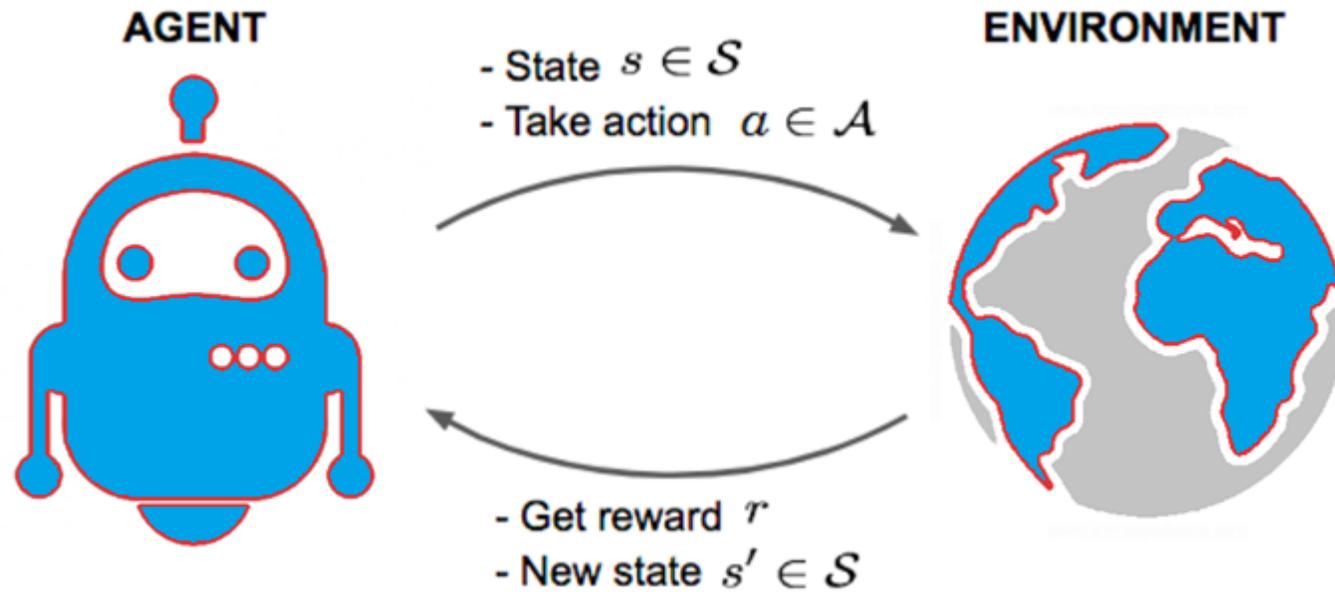


Пространство непрерывного действия

- **DDPG** (Лилликрап и др., 2015)
- **TD3** (Фудзимото и др., 2018)
- **SAC** (Хаарноя и др., 2018)



Обучение с подкреплением | Обзор



- Агент взаимодействует с окружающей средой
 - ...используя **действия**
 - наблюдает за **состоянием** окружающей среды
- Агент получает обратную связь (**вознаграждение**)
 - на основе его действий
 - вознаграждение является скаляром

- Агент хочет максимизировать **суммарную** (дисконтированную) **награду**

$$R = \sum_t \gamma^t r_t, \quad \gamma \in [0, 1]$$

- **Политика:** стратегия, которую использует агент для выбора действий

$$\pi(a | s) = P(\text{take action } a \mid \text{in state } s)$$

Среда обучения с подкреплением | проектирование функций

Задержка и потеря пакетов

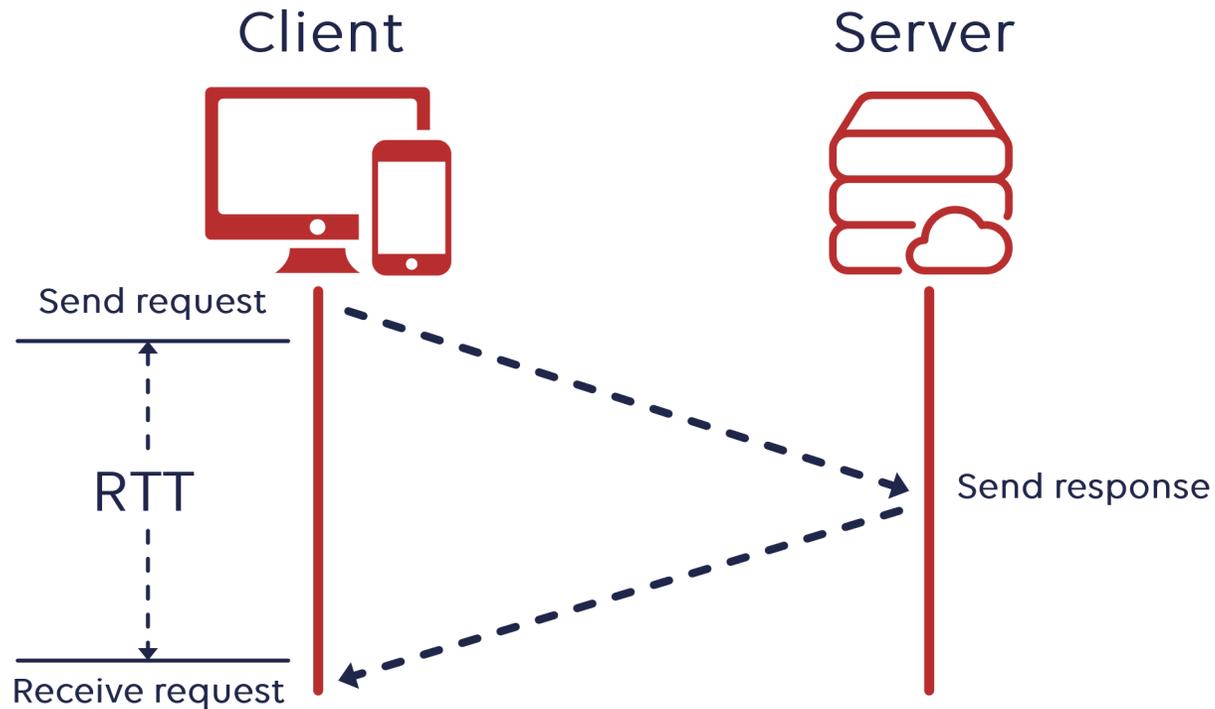


Пустое место в буфере.

Если буфер переполнен, входящие пакеты будут потеряны (**потеря пакетов**).

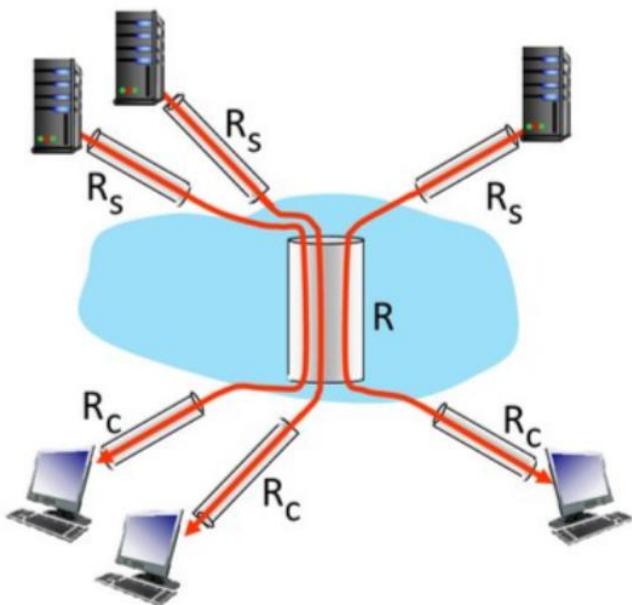
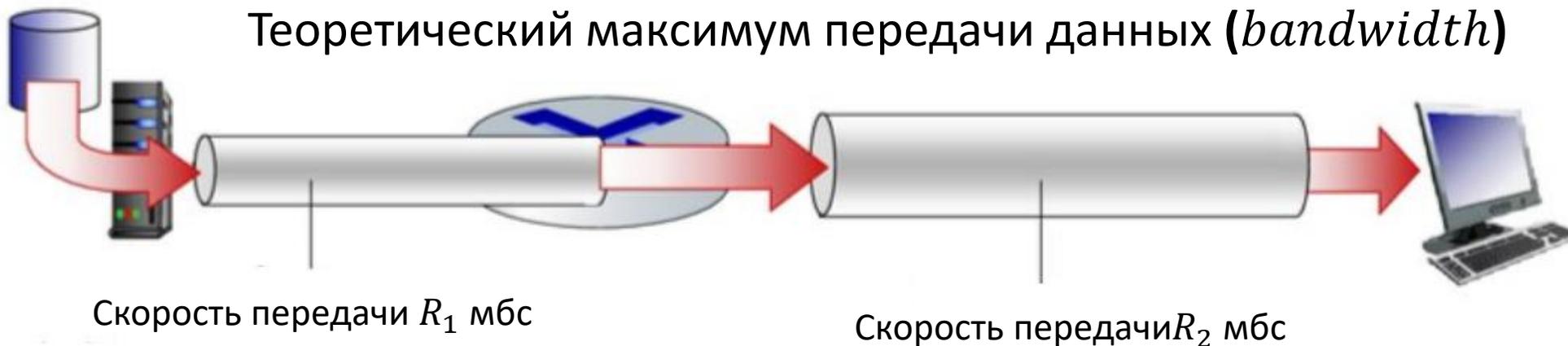
Среда обучения с подкреплением | проектирование функций

Время приема-передачи (RTT)



- Мера задержки
- Играет решающую роль в приложениях реального времени
- Оказывают огромное влияние на скорость передачи файлов.

Среда обучения с подкреплением | проектирование функций



- Для каждого соединения

$$bandwidth = \min(R_s, R_c, R/3)$$

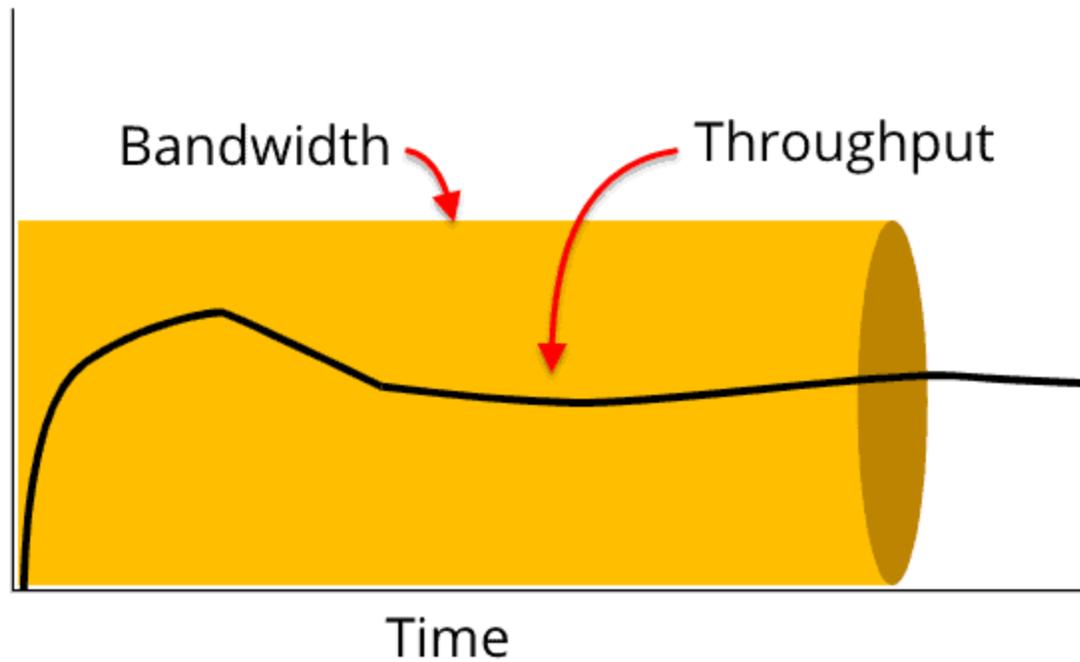
- В реальной жизни

$$bandwidth = \min(R_s, R_c)$$

- Кроме того, в реальной жизни не существует равномерного распределения пропускной способности узкого места .

пример: 3 соединения имеют общее узкое место R

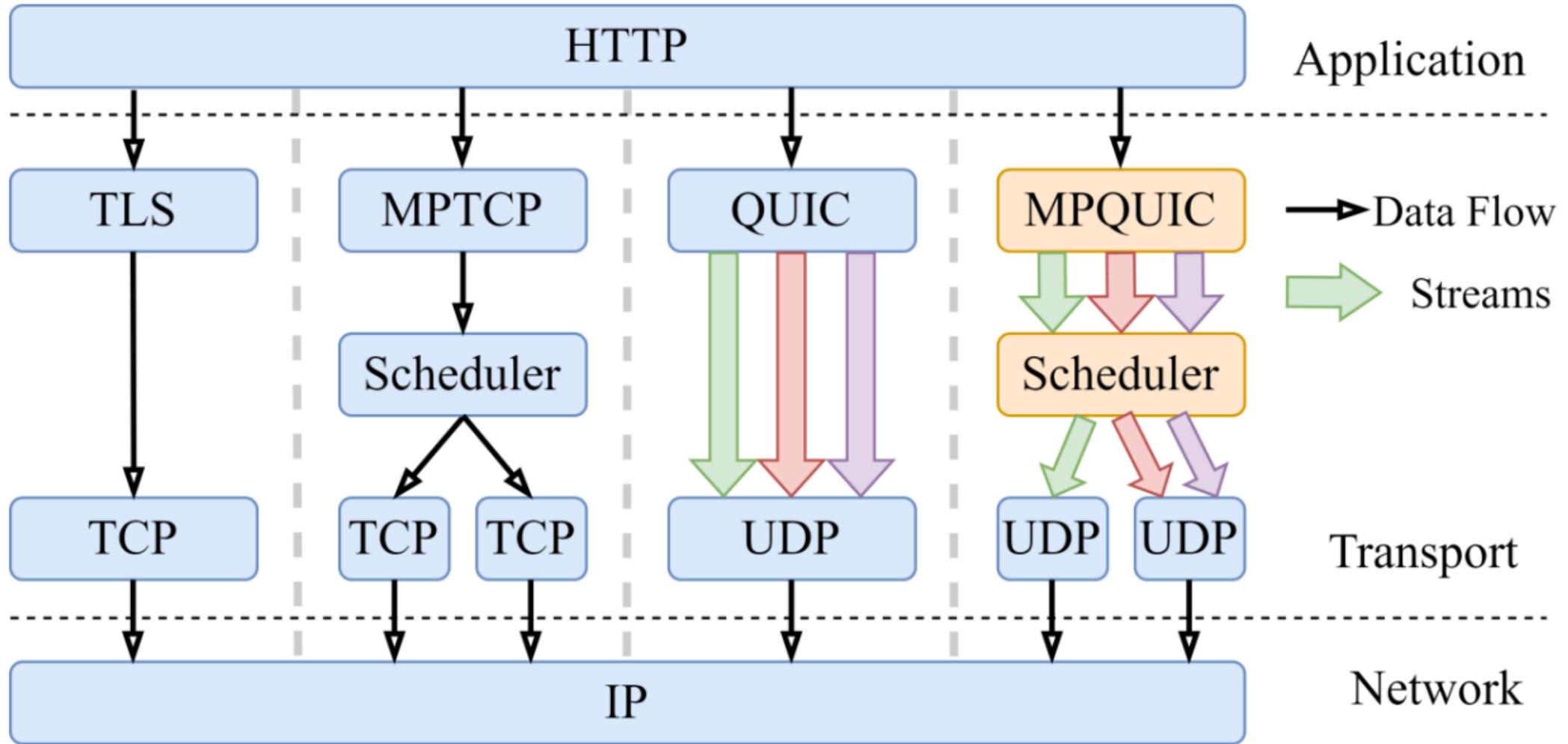
Реальная скорость передачи данных (Throughput)



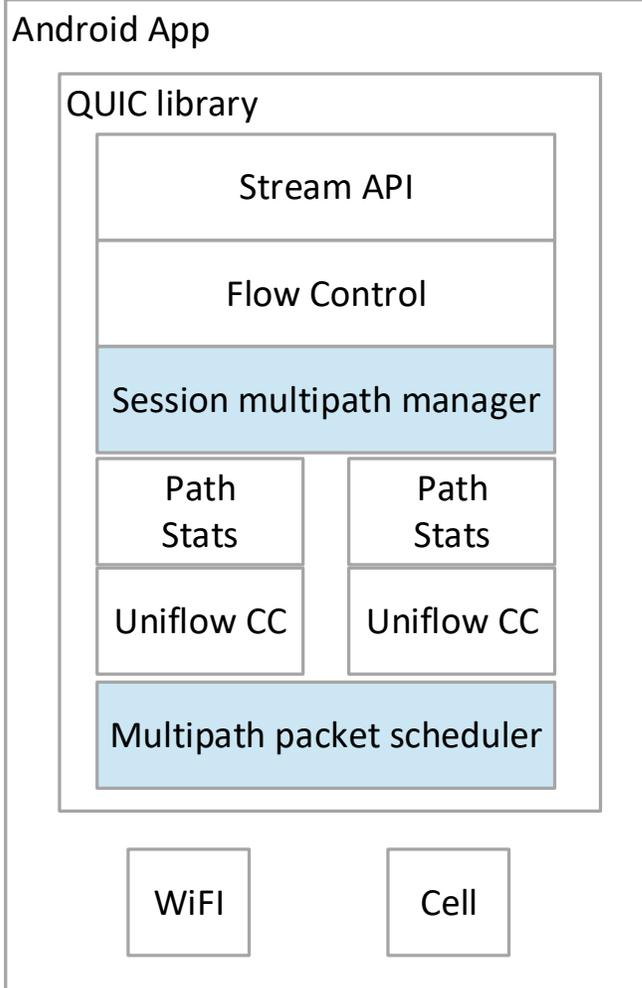
- Скорость доставки данных по каналу связи
- **Скорость доставки данных** — это верхняя граница **пропускной способности**.
- Эффективность использования полосы пропускания:

$$\text{efficiency} = \frac{\text{throughput}}{\text{bandwidth}}$$

Многоканальный QUIC | Сравнение стеков



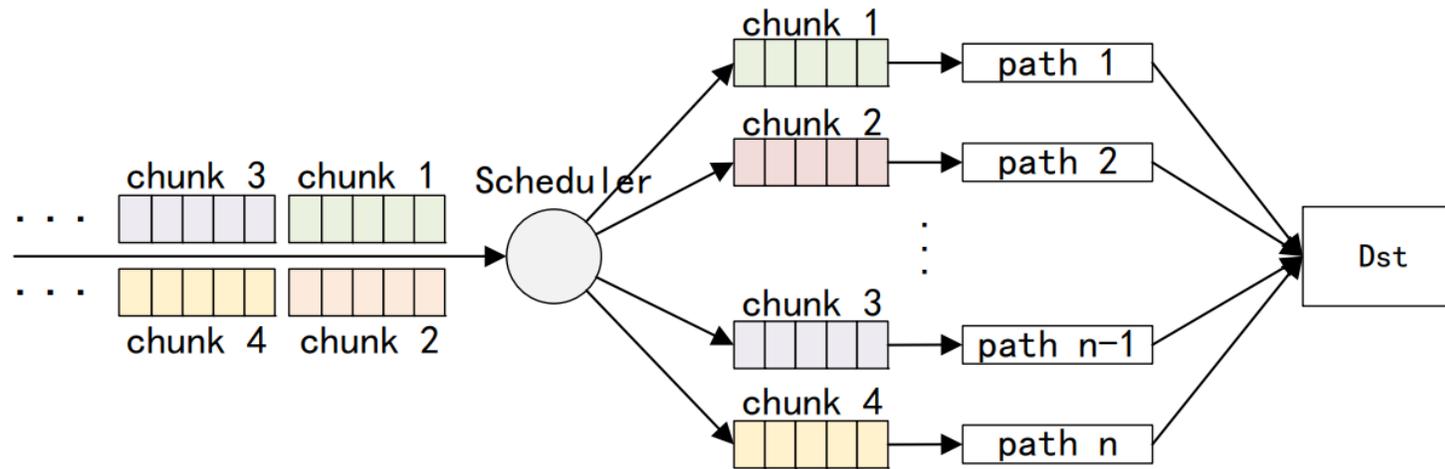
QUIC: компоненты базовой архитектуры



- **API потока** – потоки в QUIC предоставляют легкую, упорядоченную абстракцию потока байтов для приложения. Потоки могут быть однонаправленными или двунаправленными.
- **Управление потоком** . Чтобы гарантировать, что буферы приемника не переполнятся, QUIC использует два механизма управления потоком: i) основанный на потоке; ii) управление потоком на уровне соединения.
- **Контроль перегрузки на основе пути/единого потока** – контроль перегрузки на основе пути, также известный как BBR, Cubic и т. д.
- **Отправитель/получатель пакетов** – низкоуровневый интерфейс для взаимодействия со стекем сети платформы. В настоящее время мы фокусируемся на алгоритме с одним путем и возможностью миграции соединения.
- **Менеджер сеансов многопутевого управления** – модуль, который управляет операциями инициации пути, состояния пути и закрытия пути, обрабатывая следующие кадры QUIC: `NEW_CONNECTION_ID`, `PATH_CHALLENGE` и `PATH_RESPONSE`.
- **Планировщик многопутевых пакетов** – основной компонент решения, обрабатывает входящие пакеты для отправки, запрашивает текущее состояние сети, выполняет ту или иную политику планирования, выбранную агентом на основе машинного обучения/обучения с обратной связью, и в конечном итоге решает, по какому пути следует направить пакет.

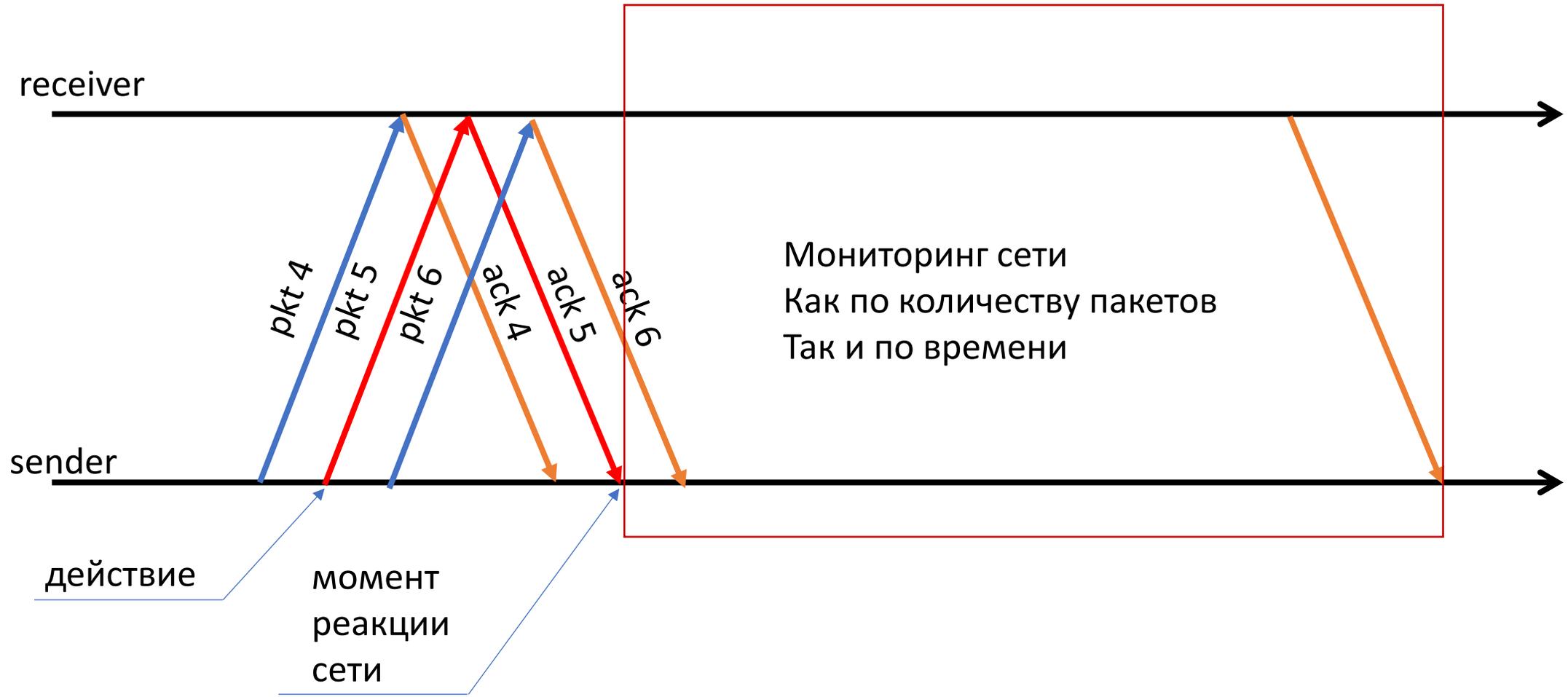
Планировщики многопутевого распространения

- Принцип многопутевого планирования данных:



- Планировщик работает для каждого пакета — важно иметь оптимальный алгоритм, чтобы избежать проблем с производительностью.
- Каждый пакет генерируется только после выбора пути.
- Контроль перегрузки на выбранном пути имеет решающее значение при принятии решения о том, отправлять ли пакет прямо сейчас.

Обучение с подкреплением | Смещение награды

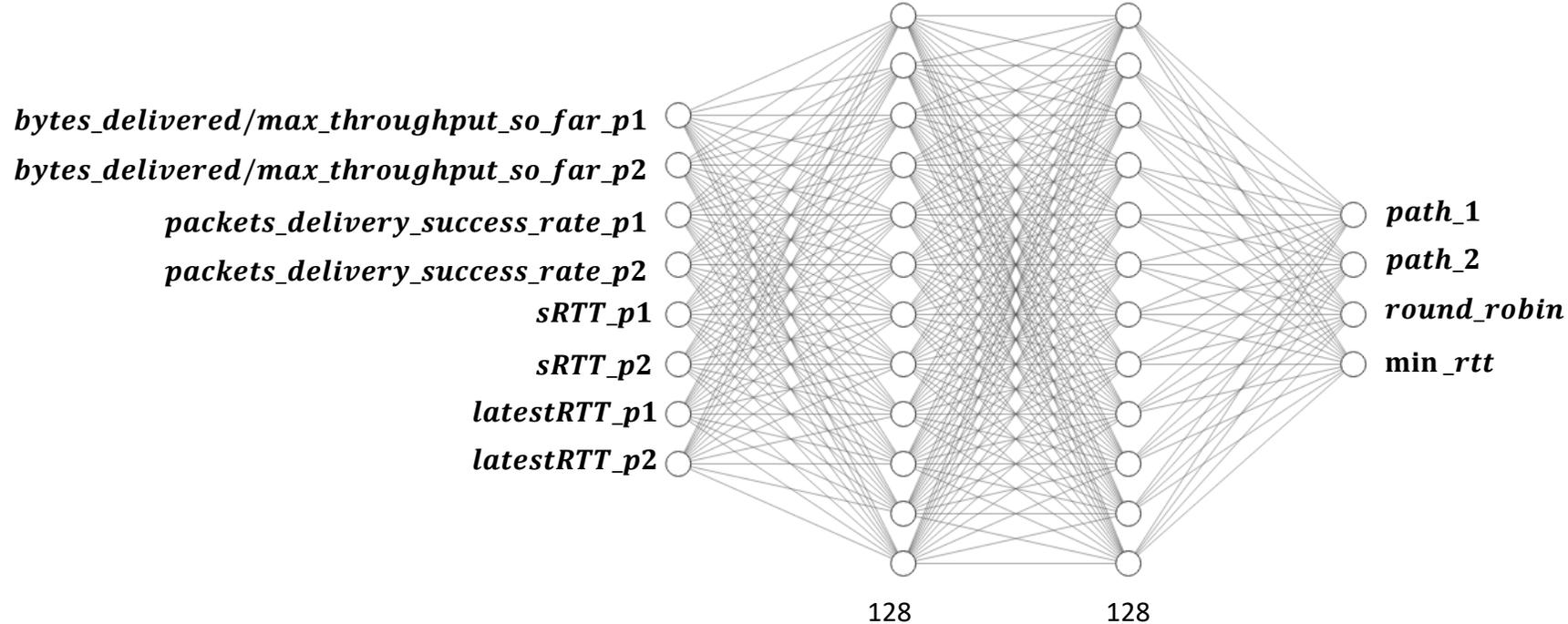


Дискретное пространство действий: приближение Q-функции

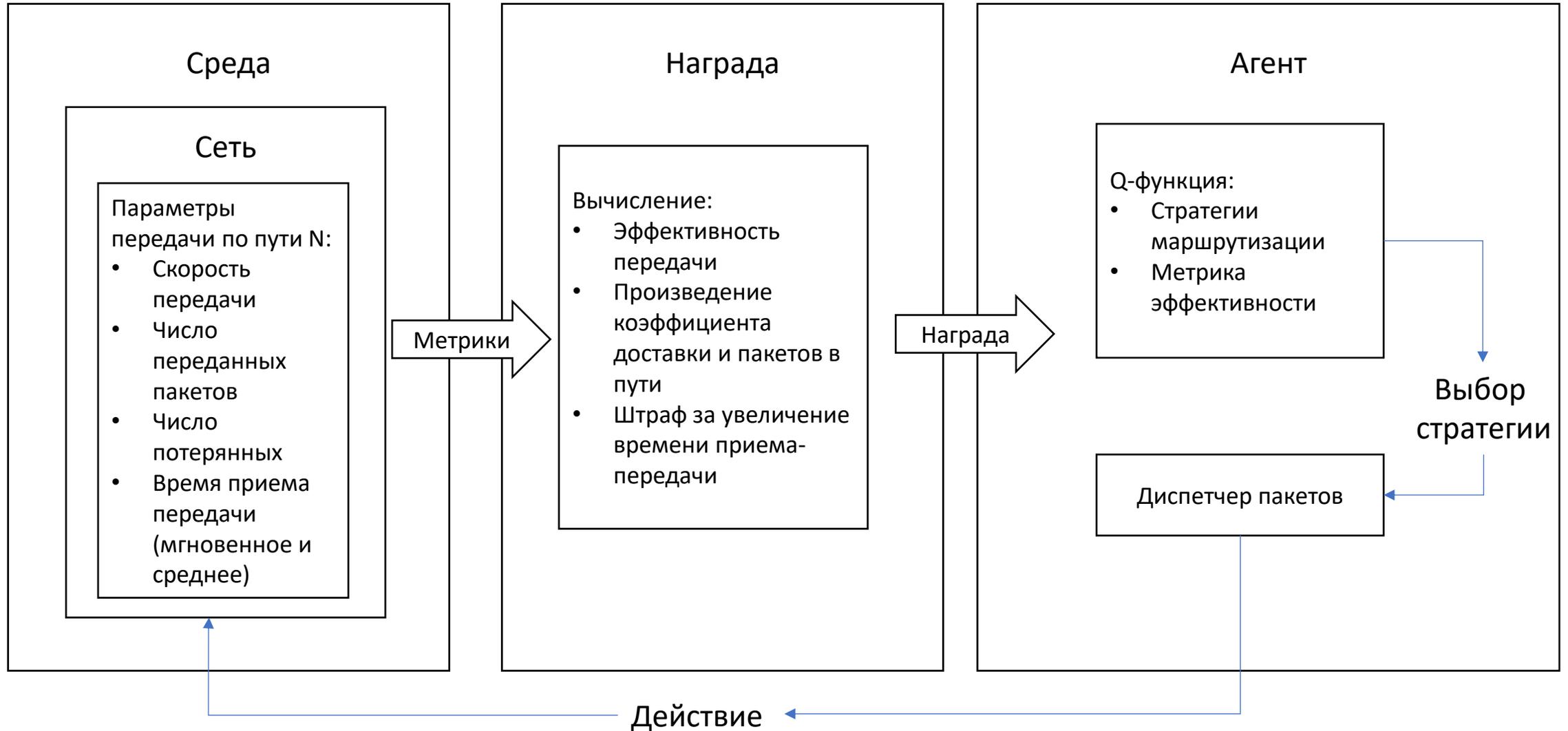
$$\mathcal{A}_{ddqn} = \{path_1, path_2, round_robin, min_rtt\}, a(t) = \underset{a}{\operatorname{argmax}} Q(s, a)$$

$$s(t) = [bytes_delivered/max_throughput_so_far \quad packets_delivery_success_rate \quad sRTT/maxRTT \quad latestRTT/maxRTT \quad]^T \in [0, 1]^{10}.$$

$$r(t) = k_1 \cdot bytes_delivered/max_throughput_so_far - k_2 \cdot bytes_in_flight \cdot packets_delivery_success_rate - k_3 \cdot sRTT$$



Архитектура решения



Окружающая среда | Агент на базе обучения с подкреплением



Параметры сети:

Асимметричная конфигурация пути:

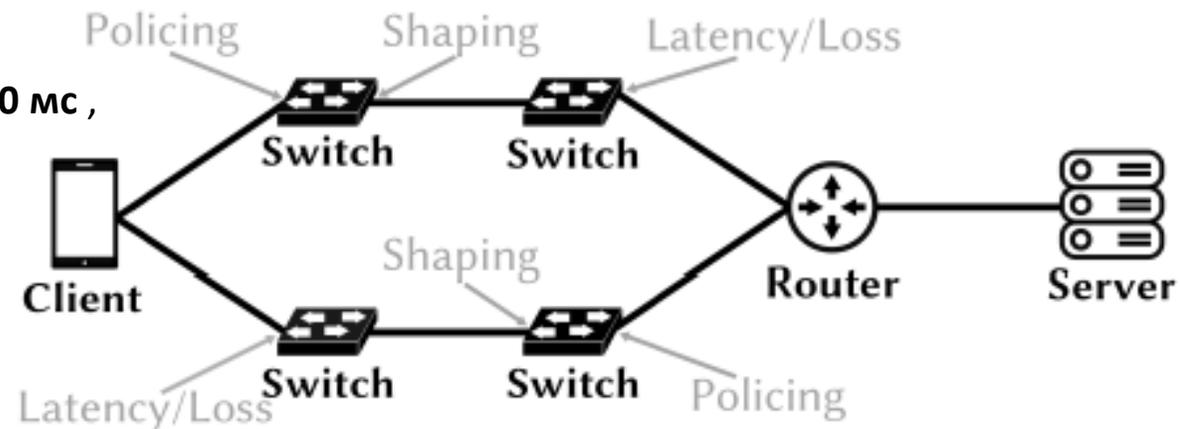
Путь с высокой пропускной способностью: **50 Мбит/с** Задержка: **100 мс**,

Путь с низкой пропускной способностью: **4 Мбит/с** Задержка **20 мс**

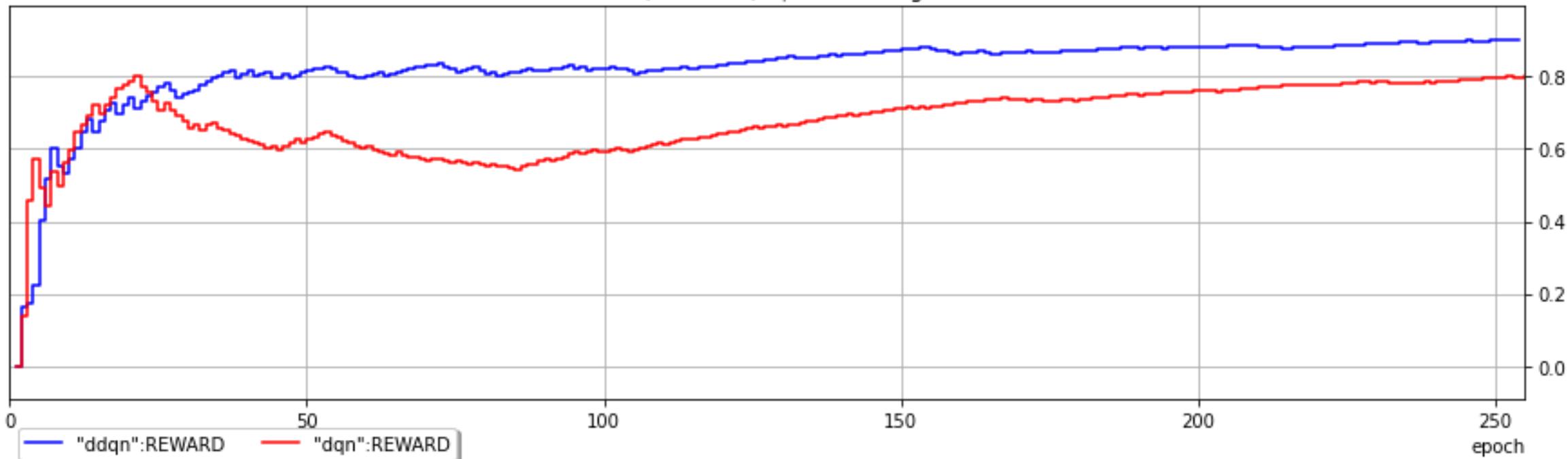
Симметричная конфигурация пути:

Путь 1: **4 Мбит/с** Задержка **10 мс**

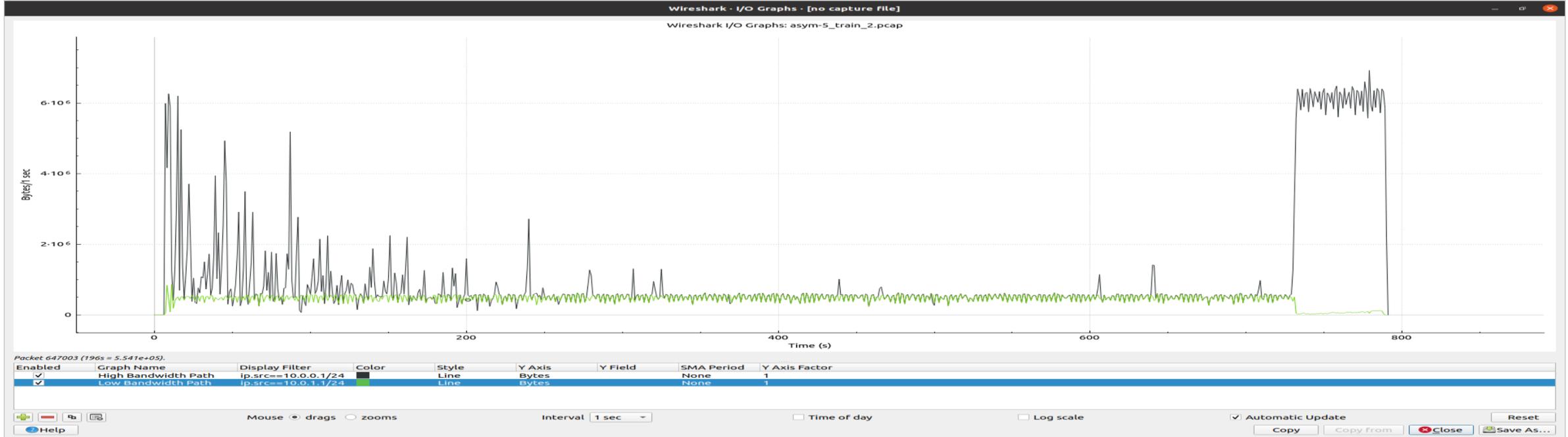
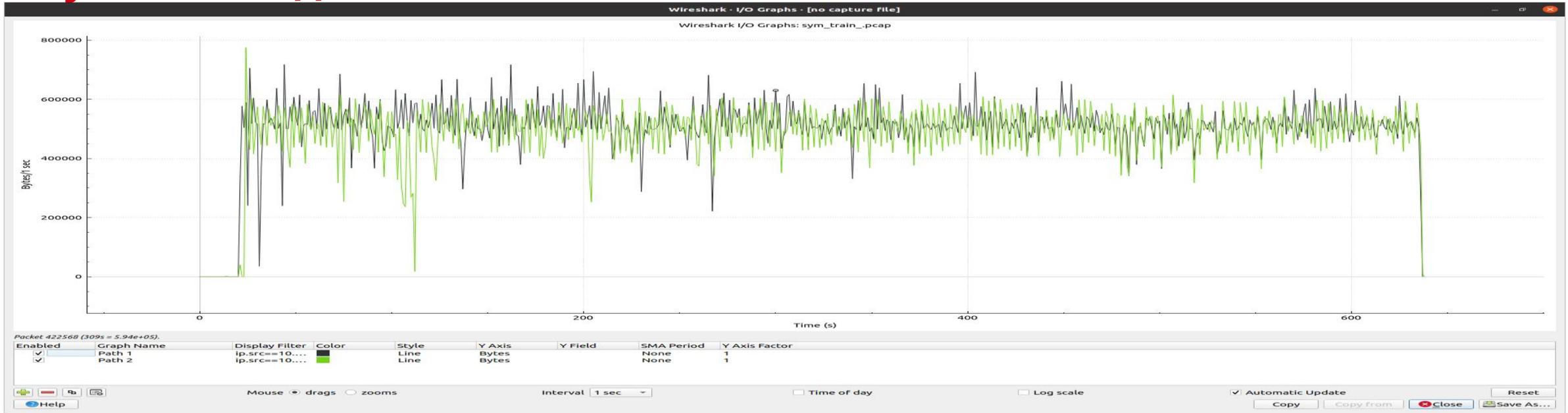
Путь 2: **4 Мбит/с** Задержка **40 мс**



DQN vs DDQN | Reward avg



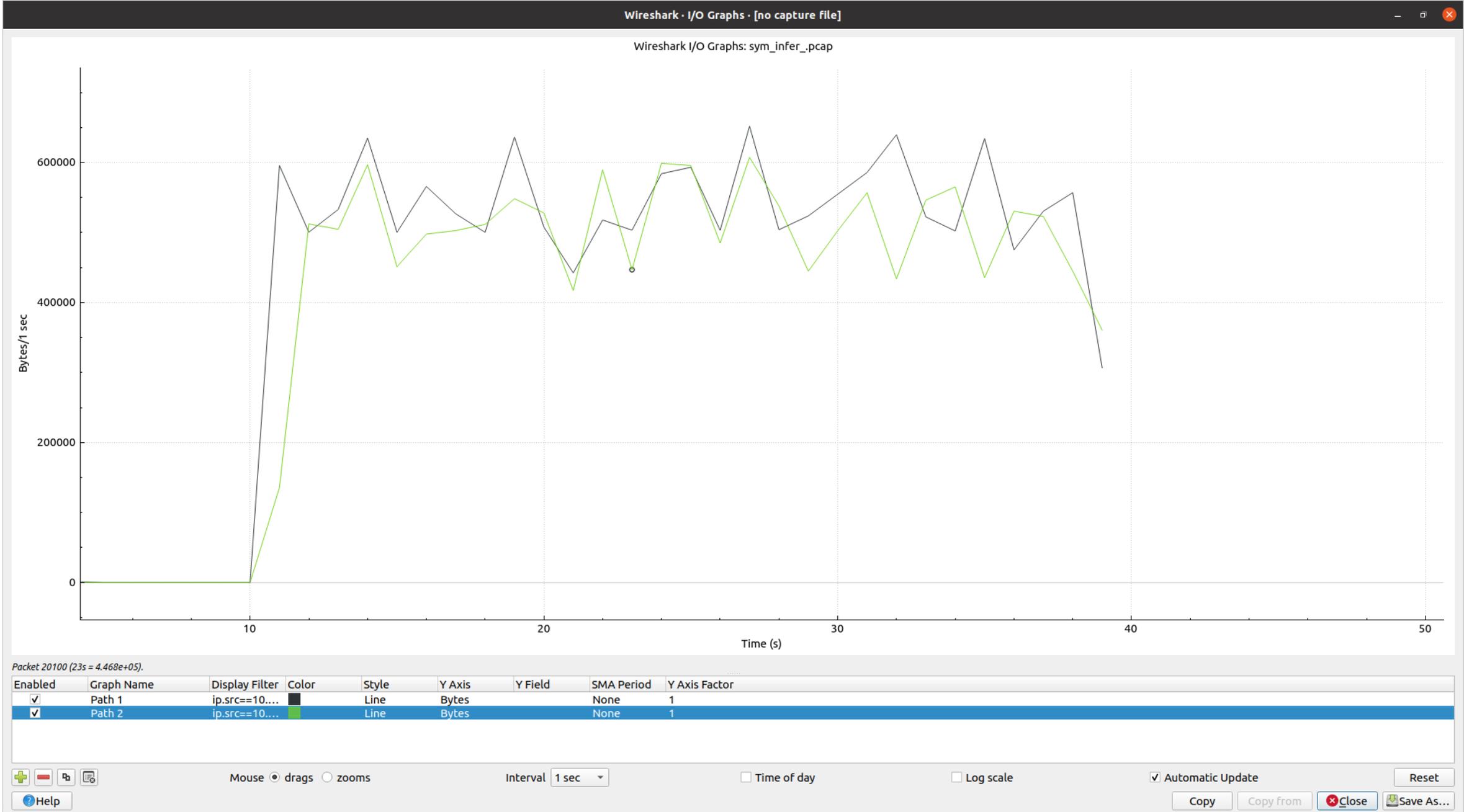
Обучение модели



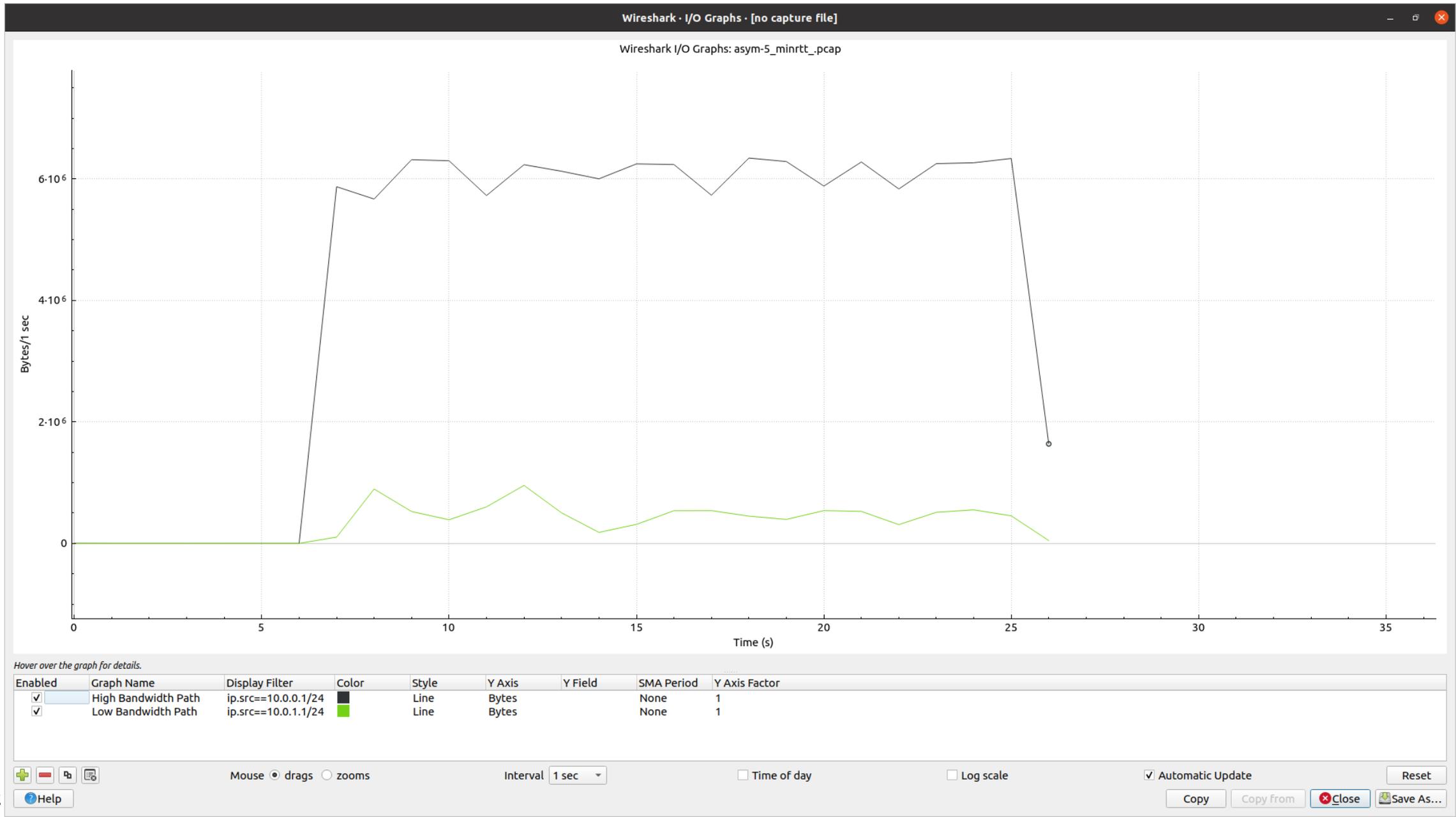
minRTT: симметричный канал



Инференс модели: симметричный канал



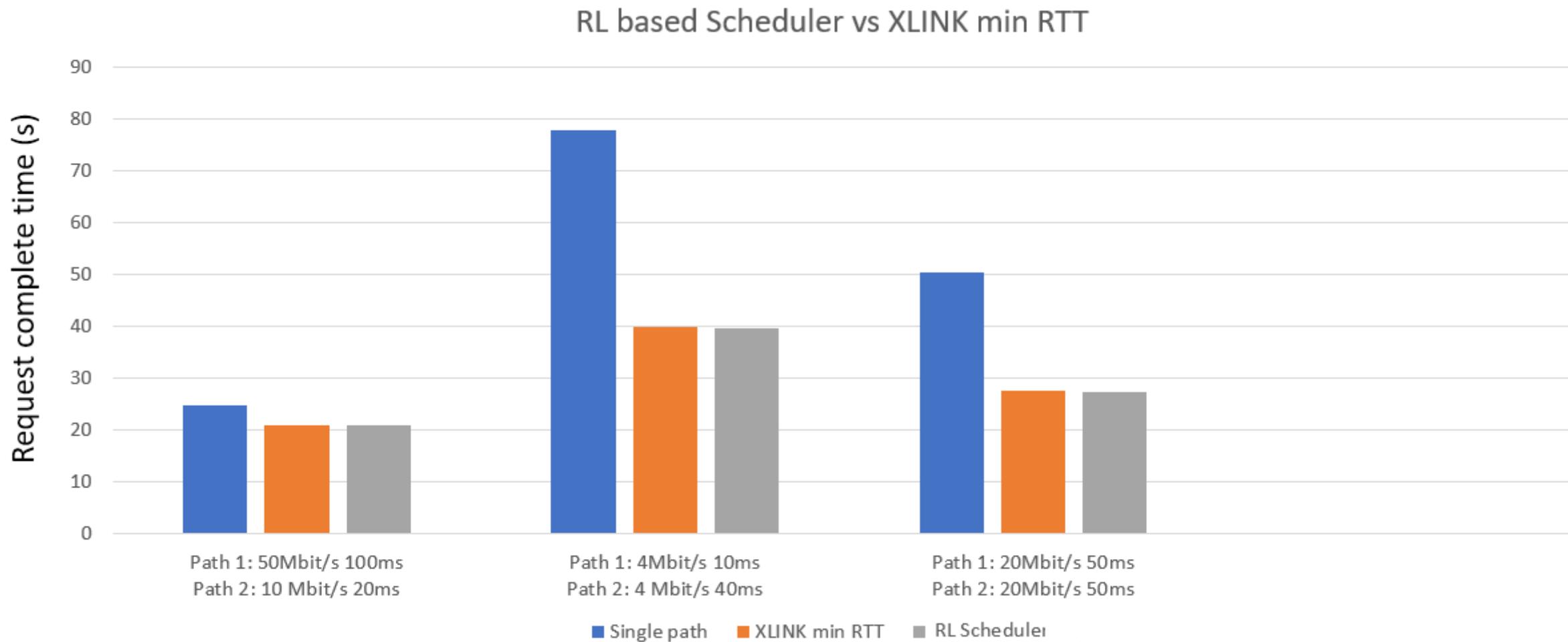
minRTT: асимметричный канал



Инференс модели: асимметричный канал



Многопутевая передача данных | Результат теста из среды minitopo



Выводы

Эксперименты показали, что оба алгоритма (DQN и DDQN) демонстрируют высокую эффективность в нахождении оптимальной политики маршрутизации, адаптируясь к динамическим изменениям в сети и учитывая такие параметры, как задержки, потери пакетов и загрузку каналов.

Однако, несмотря на их преимущества, такие как способность обучаться в сложных и нестационарных условиях, **DQN и DDQN не смогли превзойти по производительности классический алгоритм minRTT (Minimum Round-Trip Time)**, который продолжает оставаться эталоном в задачах минимизации задержек. Это может быть связано с несколькими факторами:

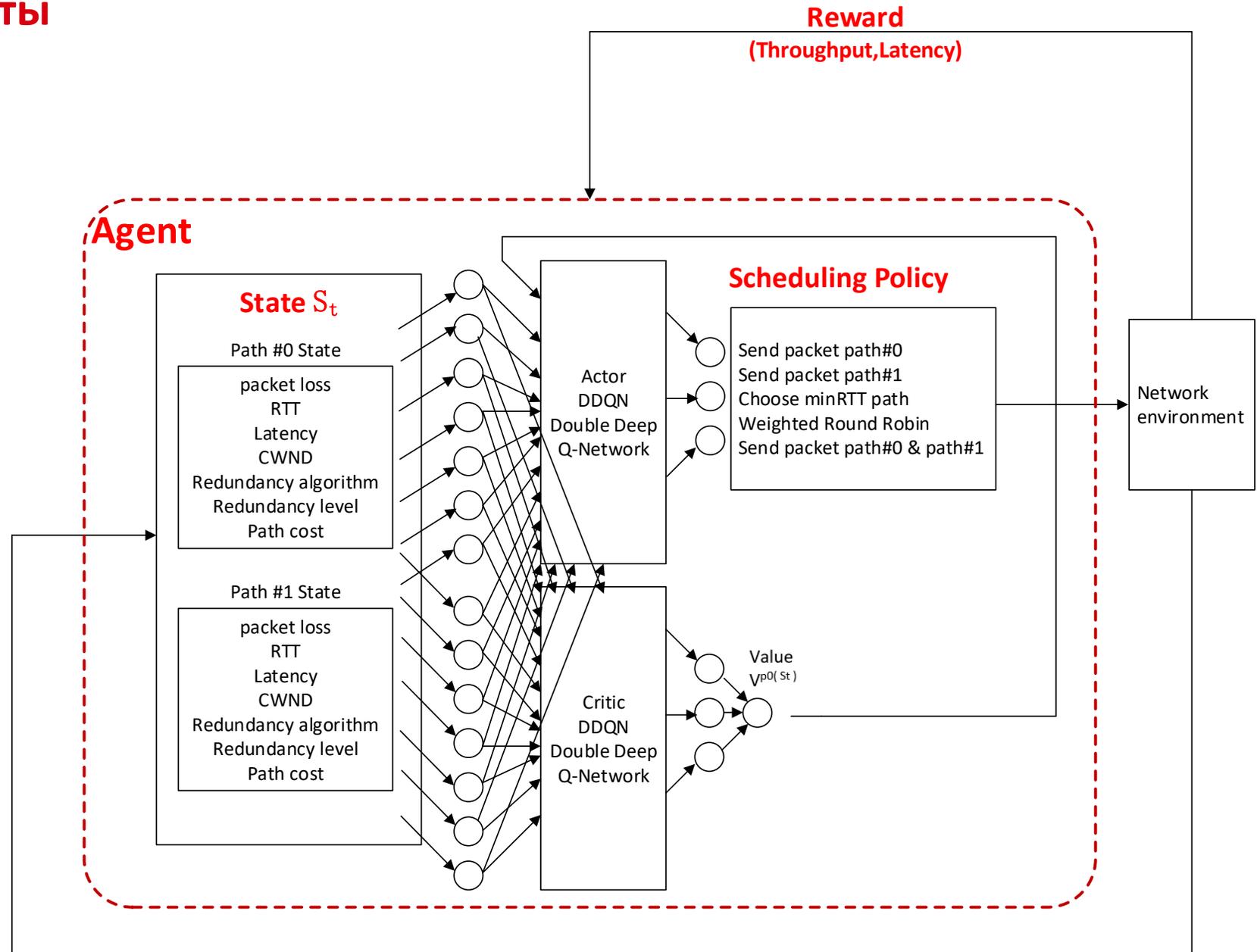
1. Сложность среды: В реальных сетевых условиях динамика изменений может быть настолько высокой, что даже мощные алгоритмы глубокого обучения не всегда успевают адаптироваться достаточно быстро.

2. Ограничения данных: Для эффективного обучения DQN и DDQN требуется значительный объем данных, что может быть затруднительно в реальных сетях с ограниченными возможностями сбора информации.

Дальнейшее развитие работы

Дальнейшие исследования в этом направлении могут быть сосредоточены на:

- **Улучшении сходимости алгоритмов:** Использование более сложных архитектур нейронных сетей или гибридных подходов, сочетающих глубокое обучение с традиционными методами (Actor-Critic).
- **Оптимизации обучения:** Применение методов ускоренного обучения, таких как transfer learning или federated learning, для уменьшения времени и ресурсов, необходимых для обучения.
- **Интеграции с существующими алгоритмами:** Комбинирование эвристических алгоритмов и глубокого обучения для создания более эффективных решений.



Спасибо за внимание!