

УДК 004.728.5:621.398

<https://doi.org/10.31854/2307-1303-2026-14-1-35-56>

EDN: YKGWDF

## Анализ протоколов передачи данных в задачах управления промышленными роботами

Белов М. А., Кузьмина Е. А., Филина Н. Р., Мутханна А. С. А.

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация

**Постановка задачи.** Развитие концепции промышленного Интернета вещей (IIoT) в рамках парадигмы «Индустрия 4.0» сопровождается увеличением числа подключенных устройств, ростом требований к скорости передачи данных и жесткими ограничениями по задержкам. В условиях реального времени даже миллисекундные задержки могут приводить к нарушению синхронизации роботизированных систем, остановке производственных линий и экономическим потерям. При этом широко используемые транспортные протоколы TCP и UDP имеют существенные ограничения: первый обеспечивает надежность ценой высокой задержки и значительного объема служебных данных, второй гарантирует минимальные задержки, но не поддерживает контроль доставки, что критично для производственных процессов. Таким образом, актуальной задачей становится поиск транспортного протокола, совмещающего надежность, низкую задержку и устойчивость к нестабильным каналам связи. **Цель работы** заключается в проведении сравнительного анализа транспортных протоколов TCP, UDP и QUIC в условиях, приближенных к промышленным сценариям, с последующей оценкой их производительности и метрики размера серии при обеспечении требований реального времени. **Методы.** В работе использованы инструментарий сетевого мониторинга Wireshark и разработанный программный комплекс на языке Python с библиотеками pyshark и matplotlib. Исследование охватывает анализ PCAP-файлов (файлов захвата сетевых пакетов), построение временных рядов, вычисление ключевых метрик (скорость передачи команд, доля потерь пакетов, размер серии команд) и их визуализацию. Элементами новизны являются: комплексное сопоставление трех транспортных протоколов применительно к задачам управления роботизированными манипуляторами, где критичны миллисекундные задержки; использование QUIC – относительно нового протокола, ранее не исследованного в прикладных сценариях IIoT; разработка программного инструмента, обеспечивающего автоматизированный анализ и классификацию сетевого трафика в промышленных условиях. **Результат:** эксперименты показали, что TCP обеспечивает нулевые потери, но неприемлемо низкую скорость передачи команд (0,58 команд/с); UDP демонстрирует на порядок большую скорость (5,8 команд/с), но при этом сопровождается потерями на уровне 3,88 %, что делает его непригодным для высокоточных операций; QUIC показывает наилучшие результаты: скорость 18,16 команд/с при потерях менее 1 % и трехкратное увеличение среднего размера серии команд. **Теоретическая значимость:** полученные результаты подтверждают перспективность применения протокола QUIC в сетях IIoT. Его использование позволяет не только обеспечить минимальные задержки и высокую надежность передачи данных, но и существенно снизить нагрузку на сеть за счет мультиплексиро-

### Библиографическая ссылка на статью:

Белов М. А., Кузьмина Е. А., Филина Н. Р., Мутханна А. С. А. Анализ протоколов передачи данных в задачах управления промышленными роботами // Информационные технологии и телекоммуникации. 2026. Т. 14. № 1. С. 35–56. DOI: 10.31854/2307-1303-2026-14-1-35-56. EDN: YKGWDF

### Reference for citation:

Belov M. A., Kuzmina E. A., Filina N. R., Muthanna A. S. A. Analysis of Communication Protocols for Industrial Robot Control Systems // Telecom IT. 2026. Vol. 14. Iss. 1. PP. 35–56. (in Russian). DOI: 10.31854/2307-1303-2026-14-1-35-56. EDN: YKGWDF

вания потоков. *Практическая реализация* может быть связана с внедрением QUIC в системы управления робототехническими комплексами, сенсорными сетями и производственными линиями, что соответствует требованиям к масштабируемым и безопасным коммуникациям будущего поколения.

**Ключевые слова:** промышленный Интернет вещей, промышленная автоматизация, транспортный протокол, TCP, UDP, QUIC, робототехнические системы, низкая задержка, надежная передача данных, мультиплексирование потоков, защита данных

## Введение

С развитием технологий «Индустрии 4.0» и активным внедрением промышленного Интернета вещей (ПоТ, аббр. от англ. Industrial Internet of Things) актуализируются вопросы обеспечения стабильности, отказоустойчивости и эффективности сетевой коммуникации на производстве. ПоТ представляет собой инфраструктуру, объединяющую киберфизические системы, датчики, исполнительные механизмы, контроллеры и интерфейсы «человек – машина» в единую сеть для реализации мониторинга и управления технологическими процессами в режиме реального времени.

В современных условиях ключевыми сетевыми требованиями в среде ПоТ являются:

- сверхнизкая задержка передачи данных (1–10 мс для управления приводами в реальном времени);
- высокая пропускная способность при одновременной передаче управляющих и телеметрических потоков;
- гарантированная доставка и устойчивость к потере пакетов;
- интеграция механизмов криптографической защиты при передаче критически важных данных.

Наиболее распространенными решениями транспортного уровня в сетях ПоТ остаются протокол управления передачей (TCP, аббр. от англ. Transmission Control Protocol) и протокол пользовательских дейтаграмм (UDP, аббр. от англ. User Datagram Protocol). TCP обеспечивает гарантированную доставку, однако характеризуется высокими задержками из-за механизмов подтверждения и повторных передач. В свою очередь, UDP минимизирует накладные расходы, но не проверяет целостность доставленных данных, что ограничивает его применение в высокоточных производственных процессах.

Для преодоления данных ограничений был разработан транспортный протокол нового поколения QUIC (аббр. от англ. Quick UDP Internet Connections – быстрые мультиплексированные соединения через UDP), стандартизированный IETF (аббр. от англ. Internet Engineering Task Force – Инженерный совет по развитию интернета) [1] и сочетающий механизмы восстановления потерь, мультиплексирование независимых потоков, встроенное шифрование на базе протокола Transport Layer Security (TLS 1.3) и ускоренное установление соединения (0-RTT). Данные архитектурные свойства делают его перспективным решением для промышленных приложений, чувствительных к скорости отклика и сохранности управляющих команд.

Таким образом, в отрасли складывается проблемная ситуация: существует острая необходимость в обеспечении отказоустойчивой, масштабируемой и безопасной передачи данных, однако традиционные протоколы не способны одновременно удовлетворить строгие требования к задержке, скорости и целостности доставки. Для разрешения данного противоречия в работе предложен экспериментальный подход, включающий сравнительный анализ протоколов TCP, UDP и QUIC в контексте управления промышленными робототехническими системами. Эксперименты выполнялись с использованием робота-манипулятора (PM) DOBOT Magician, а анализ сетевого трафика производился средствами Wireshark и разработанным программным модулем на Python, который обрабатывает PCAP-файлы и визуализирует конкретные метрики сетевого взаимодействия: пропускную способность, уровень потерь пакетов и размер серии команд  $b$ .

Результаты исследования позволяют объективно оценить применимость транспортных протоколов в системах IoT и определить оптимальное решение для повышения эффективности промышленных сетей. Полученные данные доказывают, что протокол QUIC обеспечивает наилучший баланс между скоростью, безопасностью и надежностью, выступая перспективным стандартом для коммуникационной инфраструктуры «Индустрии 4.0».

### Постановка задачи

Для формальной постановки и решения задачи в работе введены обозначения, представленные в таблице 1.

Таблица 1. Принятые обозначения

Обозначение	Содержание
СУ	система управления, формирующая и передающая управляющие команды
СС	совокупность активного сетевого оборудования и каналов передачи данных, обеспечивающих двустороннюю доставку пакетов управляющих команд и сигналов
PM	робот-манипулятор DOBOT Magician, исполнительная часть системы
$S_{DC}$	размер одной управляющей команды, байт
$L$	объем служебной информации (заголовки, метаданные), байт
$b$	размер серии, число команд в одном пакете
$\Omega_1, \Omega_2$	пространственные пределы допустимых координат манипулятора (внешняя и внутренняя сферы)
$x, y, z$	координаты рабочей точки манипулятора
$r_1, r_2$	радиусы внутренней и внешней сфер, мм

На вербальном уровне задача оптимизации и оценки передачи управляющих команд в исследовании декомпозируется следующим образом:

– формирование программно-аппаратного стенда с клиент-серверной архитектурой, позволяющей тестировать протоколы TCP, UDP, QUIC в контролируемой локальной сети и варьировать сетевые параметры (задержка, пропускная способность, вероятность потерь);

– определение допустимой рабочей области РМ через границы сфер  $\Omega_1, \Omega_2$  и координат  $x, y, z$ ;

– организация передачи команд сериями размера  $b$  и формализация механизма подтверждения / повторной передачи (замкнутый цикл: СУ  $\rightarrow$  пакет  $b \rightarrow$  буфер РМ  $\rightarrow$  подтверждение / выполнение  $\rightarrow$  запрос следующего пакета или повтор при ошибке);

– экспериментальное исследование и сравнительная оценка эффективности передачи команд для каждого протокола по таким метрикам, как пропускная способность (команд в секунду), уровень потерь пакетов и размер серии команд  $b$  (среднее число команд на пакет).

На формальном уровне задача исследования имеет следующий вид.

Дано:

– множество компонентов экспериментального стенда: СУ, СС, РМ;  
– транспортные протоколы TCP, UDP, QUIC, каждый с заданными параметрами: пропускная способность, задержка, вероятность потери пакета;

– рабочая область манипулятора, определяемая допустимыми координатами  $x \in (-120, 320)$ ;  $y \in (-300, 320)$ ;  $z > 0$ ; радиусы внутренней и внешней сфер  $r_1 = 180$  мм;  $r_2 = 300$  мм;

– оператор передачи серии команд, обеспечивающий: формирование пакета из  $b$  команд, передачу через сетевой протокол (TCP, UDP или QUIC), подтверждение или повтор при ошибке исполнения команды.

Требуется оценить:

– влияние выбора транспортного протокола на средний размер серии команд и итоговую пропускную способность системы управления.

## Фундаментальные принципы сетевого взаимодействия в ПоТ

Современные архитектуры ПоТ и технологии межмашинного взаимодействия базируются на распределенных вычислениях, включающих уровни физических устройств, периферии и облака. На уровне устройств располагаются датчики, исполнительные механизмы, промышленное оборудование и роботы, которые формируют телеметрию с помощью проводных (например, Modbus, Industrial Ethernet) и беспроводных интерфейсов (Wi-Fi, LoRa, 5G). Периферийные вычисления выполняют первичную фильтрацию и снижают нагрузку на облачные системы, разделяясь на три подуровня: дальний (Far Edge) обеспечивает мгновенную реакцию с миллисекундными задержками; средний (Mid Edge) занимается агрегацией данных с задержками в секунды; ближний (Near Edge) решает ресурсоемкие аналитические задачи [4].

Межмашинное взаимодействие в ПоТ реализуется через несколько моделей коммуникации: периодическая телеметрия, событийно-ориентированные сообщения и синхронизация по расписанию. Исследования характеристик трафика показывают наличие долгосрочных корреляций и самоподобия в промышленных телеметрических потоках. Так, оценки коэффициента Хёрста для трафика, генерируемого системами аддитивного производства, подтверждают его самоподобный

характер: для промышленной системы Trumpf TruPrint 1000 этот показатель составляет 0,83, а для системы 3D Systems ProJet 4500 достигает 0,96. Это указывает на выраженные временные корреляции и диктует необходимость внедрения механизмов управления очередями и динамического выделения полосы пропускания.

Свойства трафика ПоТ существенно отличаются от традиционного интернет-трафика в силу разнообразия источников и сценариев применения. Источники данных можно классифицировать на датчики и исполнительные механизмы, корпоративные бизнес-приложения, мультимедийные системы (например, видеонаблюдение) и внешние веб-сервисы. Эмпирические исследования показывают, что трафик от 3D Systems ProJet 4500 описывается гамма-распределением с параметрами  $\alpha = 92,86$  и  $\lambda = 50\ 000$ , что отражает изменчивость межпакетных интервалов. Потоки бизнес-приложений (например, 1С Bitrix) могут подчиняться распределению Вейбулла – Гнеденко с параметрами  $\alpha = 36,85$ ;  $c = 0,00019$ . Платформы видеонаблюдения (например, Ivideon) демонстрируют характеристики трафика, описываемые гамма-распределением ( $\alpha = 8,99$ ;  $\lambda = 11\ 111$ ) со средним межпакетным интервалом около 1,05 мс. В свою очередь, открытые веб-сервисы (например, метео-API Open Weather Maps) вносят низкоинтенсивный, но высоко вариативный трафик, подчиняющийся экспоненциальному закону распределения ( $\lambda \approx 1112,82$ ; интервал 1,06 мс), что в совокупности усложняет прогнозирование нагрузки и контроль доступа [1].

Интероперабельность в таких сетях остается важнейшей проблемой из-за гетерогенности протоколов и форматов данных. Устаревшее оборудование на базе Modbus часто должно взаимодействовать с современными системами, использующими OPC UA или MQTT. Для этого предлагаются семантические шлюзы, обеспечивающие преобразование между протоколами и унификацию данных. На практике входящие пакеты Modbus TCP преобразуются во внутренний формат ИСФ (*аббр. от англ. Industrial Internet of Things Conversion Format*) для стандартизации структуры данных, после чего семантическое преобразование адаптирует форматы полезной нагрузки (например, трансляция между CSV, JSON и XML), а конечное преобразование конвертирует ИСФ в целевые прикладные протоколы, такие как CoAP или HTTP [2].

Киберфизические системы объединяют физические процессы с вычислительными моделями, интегрируя датчики, исполнительные механизмы, сетевую инфраструктуру и программное обеспечение в системы, которые воспринимают окружающую среду, анализируют данные и формируют управляющие воздействия с динамической обратной связью. Они применяются в различных отраслях – от промышленной автоматизации и транспорта до медицины и энергетики [3, 4]. Промышленный РМ является классическим примером такой системы: он получает управляющие команды по сети, обрабатывает обратную связь от датчиков в реальном времени и координирует свои действия с другими устройствами производственной линии.

Реализация CPS предъявляет строгие требования к синхронизации компонентов, из-за чего сетевые инфраструктуры ПоТ должны удовлетворять условиям, значительно превышающим возможности традиционных информационно-

технологических сетей. К ключевым сетевым требованиям относятся: сверхнизкая задержка (для управления приводами в реальном времени требуется 1–10 мс), достаточная пропускная способность, высокая надежность в условиях электромагнитных помех и нестабильного питания, масштабируемость до тысяч устройств и энергоэффективность для автономных узлов. Поскольку детерминированность поведения является фундаментальным условием корректного функционирования синхронизированных задач управления, технология TSN (*аббр. от англ. Time-Sensitive Networking* – сети с высокими требованиями к задержкам) стала ключевым решением для коммуникаций в реальном времени, обеспечивая ограниченные задержки и предсказуемую доставку в промышленных сценариях. Наряду с транспортными характеристиками, критически важной является комплексная защита инфраструктуры: стратегии обеспечения безопасности должны быть многоуровневыми и включать шифрование данных, аутентификацию устройств, сегментацию сети и управление жизненным циклом систем для минимизации киберугроз.

Требования масштабируемости и энергоэффективности стимулируют использование легковесных протоколов обмена сообщениями и энергосберегающих стратегий взаимодействия для устройств с ограниченными ресурсами. Главными ограничениями узлов IoT остаются лимиты вычислительных мощностей, доступной энергии и скорости обработки данных. Для решения этих проблем применяются специализированные прикладные протоколы: например, MQTT, использующий легковесные заголовки для снижения энергопотребления, и CoAP, который целенаправленно функционирует поверх транспортного протокола UDP. Главная причина отказа от TCP на уровне таких устройств – стремление исключить избыточные издержки транспортного уровня и снизить потребление пропускной способности. Однако использование базового UDP для снижения нагрузки неизбежно ведет к пропорциональному снижению надежности доставки. Необходимость одновременно обеспечить минимальное энергопотребление и гарантированную доставку команд обуславливает актуальность поиска универсальных транспортных протоколов нового поколения [5].

### Обзор связанных исследований

Эффективность QUIC научно подтверждается и непосредственно в задачах облачных систем управления, где контроллер отделен от физического объекта управления сетью. Сравнительный анализ производительности QUIC (HTTP/3) и TCP (HTTP/1.1) в публичных облаках и 5G-сетях доказал, что при высоких частотах опроса (например, с интервалом выборки 5 мс), когда сетевая задержка превышает этот интервал, QUIC однозначно превосходит традиционные решения. Мультиплексирование потоков устраняет критическую для TCP проблему межпоточной блокировки, гарантируя более короткие и предсказуемые задержки отклика, а также генерируя меньший служебный трафик на каждый запрос [6].

Экспериментальная оценка QUIC применительно к задачам реального времени показала, что датаграммный режим протокола, отключающий механизм повторной передачи, обеспечивает однонаправленную задержку, сопоставимую

с UDP. В частности, на физическом стенде с периодом передачи 10 мс порог надежности 99,9 % достигается при задержке менее 1 мс, что укладывается в нормативные требования для критических приложений реального времени. Данные метрики подтверждают, что протокол способен удовлетворить строгие требования высокоточных контуров управления РМ [7].

### Анализ транспортных альтернатив для доставки команд в роботизированных системах

В рамках научно-исследовательских и прикладных разработок, проводимых в Санкт-Петербургском государственном университете телекоммуникаций им. проф. М. А. Бонч-Бруевича, была создана экспериментальная архитектура для дистанционного управления РМ, основанная на использовании транспортного протокола TCP. Соответствующее решение реализовано в виде зарегистрированного программного обеспечения (свидетельство № 2024686344 от 07.11.2024) и отражено во вкладе Российской Федерации в рабочую группу ITU-T SG11 в рамках проекта Q.TSN.

Модельная сеть состоит из трех уровней: управляющий сервер, клиент (прокси-контроллер) и исполнительное устройство – РМ. Управляющие команды передаются сериями (в формате JSON-структур) по протоколу TCP, что позволяет обеспечить гарантированную доставку, контроль целостности данных и синхронность их выполнения. В рамках проводимого эксперимента использование подобной архитектуры направлено на достижение высокой точности реализации последовательных пространственных перемещений, что необходимо для корректной оценки базовых параметров контура управления манипулятором перед тестированием альтернативных протоколов.

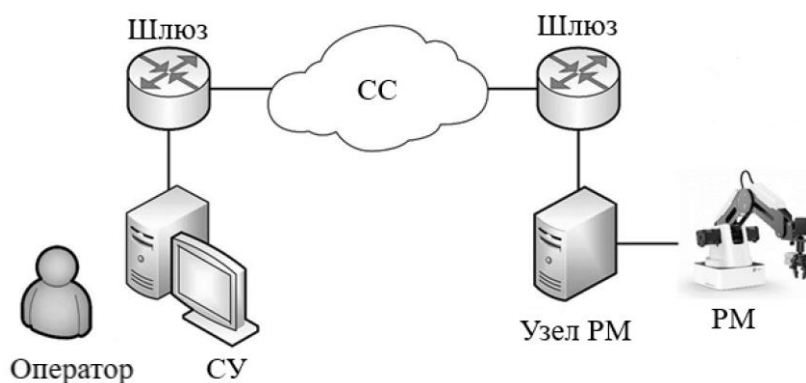


Рис. 1. Структура модельной сети сегмента РМ

Однако экспериментальная оценка выявила ряд фундаментальных ограничений TCP в сценариях высокочастотной передачи команд. Механизмы подтверждения доставки создают накопленные задержки, служебные заголовки занимают до 40–60 % от объема пакета, а увеличение сетевой нагрузки приводит к снижению стабильности, особенно в условиях эмуляции деградации параметров канала связи. При частотах отправки команд 10–20 Гц формируются оче-

реди, движения РМ теряют синхронизацию, возникают ошибки выполнения движений, что подтверждается протоколами измерений. Эти ограничения демонстрируют узость применения ТСР в распределенных системах управления реального времени и телеприсутствия.

Для преодоления выявленных ограничений в рамках эксперимента проводится разработка и оценка альтернативных протоколов UDP и QUIC. Первый из них обеспечивает минимальные задержки для приложений с высокой плотностью команд, а второй сочетает скорость UDP с надежностью ТСР, встроенным шифрованием и мультиплексированием потоков. Практическая реализация включает тестирование механизмов доставки команд, оптимизацию использования сетевых ресурсов и проведение сравнительной оценки по показателям пропускной способности, уровня потерь пакетов, размера серии команд и встроенной безопасности. На основе полученных данных формулируются рекомендации для эксплуатационных сценариев, таких как телеприсутствие и высокочастотное управление.

Платформа DOBOT поддерживает различные методы доставки команд, каждый из которых имеет различное соотношение задержки, надежности, масштабируемости и сложности интеграции. Традиционное последовательное соединение (COM через USB) обеспечивает минимальную задержку для локального управления, но архитектурно не масштабируется для распределенных систем. Сетевые решения прикладного уровня, такие как WebSocket, MQTT и REST API, позволяют организовать гибкое масштабируемое взаимодействие, однако их применение для высокочастотного управления роботизированными комплексами ограничено из-за вносимых задержек и отсутствия жестких временных гарантий доставки на транспортном уровне [8].

Сравнительный анализ доступных методов доставки команд для платформы DOBOT, представленный в таблице 2, наглядно демонстрирует компромиссы между различными методами доставки команд.

Таблица 2. Сравнение альтернативных методов доставки команд для платформы DOBOT

Метод доставки	Тип соединения	Задержка	Надежность доставки	Масштабируемость	Сложность внедрения
Serial COM	USB / UART	Низкая	Высокая	Отсутствует	Низкая
WebSocket API	TCP / WebSocket	Средняя	Средняя	Высокая	Средняя
MQTT	TCP / MQTT	Средняя	Средняя	Высокая	Средняя
REST API	HTTP	Высокая	Низкая	Средняя	Низкая
ROS	TCP / UDP (ROS Stack)	Средняя	Высокая	Средняя	Высокая
TCP	TCP / IP	Высокая	Высокая	Ограниченная	Условная
UDP	UDP / IP	Низкая	Низкая	Средняя	Средняя
QUIC	UDP + TLS	Низкая	Высокая	Высокая	Средне-высокая

Хотя интеграция на основе протоколов прикладного уровня (MQTT, WebSocket) или профильных фреймворков (ROS) обеспечивает высокую масштабируемость, их зависимость от механизмов компенсации потерь на верхних уровнях и чувствительность к межпоточной блокировке приводят к непредсказуемым задержкам. В свою очередь, анализ на транспортном уровне ясно выявляет противоречие: классический TCP гарантирует надежность в ущерб оперативности, а базовый UDP минимизирует задержки ценой потери целостности данных.

Таким образом, ни один из рассмотренных подходов не является универсальным, что обуславливает необходимость прямой эмпирической оценки транспортных протоколов TCP, UDP и QUIC в тестовой среде DOBOT. Дальнейший анализ сосредоточен на объективных аппаратных метриках: скорости передачи команд, уровне потерь пакетов и достигаемом размере серии команд  $b$ .

### Моделирование сетевого взаимодействия и управление роботизированным манипулятором

В рамках данного исследования была разработана архитектура программно-аппаратного стенда, предназначенная для передачи команд на РМ в контролируемой локальной сети. Основная цель системы заключается в обеспечении гибкости выбора транспортного уровня, возможности тестирования различных реализаций протоколов (TCP, UDP и QUIC) и получения репрезентативных экспериментальных данных в стандартизированных условиях.

Архитектура стенда состоит из двух персональных компьютеров (обозначаемых как PC#1 и PC#2) и РМ DOBOT Magician. Такое разделение обеспечивает клиент-серверную структуру, позволяя осуществлять удаленный доступ, симметричную обработку команд и четкое логическое разделение источника команд (сервер), промежуточного узла (клиент) и физического исполнителя (робот).

DOBOT Magician – компактный РМ с четырьмя степенями свободы, предназначенный для манипуляции объектами, взаимодействию с сенсорами и точному позиционированию (рисунок 2). Программируемость и стандартные интерфейсы (USB, UART) делают его пригодным для прикладного и экспериментального тестирования, включая оценку новых протоколов передачи команд.

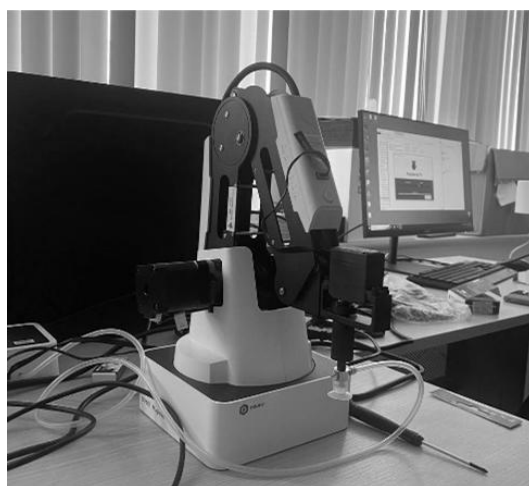


Рис. 2. РМ DOBOT Magician

В используемой конфигурации РМ имеет следующие характеристики:

- радиус рабочей области: 340 мм;
- количество осей: 4;
- точность: позиционирования – до 0,2 мм, повторяемости – до 0,02 мм;
- максимальная нагрузка: 500 г;
- конечные выключатели защищают РМ от повреждений при достижении им конечной точки в своем движении;
- подключение: через USB-порт;
- программное обеспечение: DOBOT Studio, DOBOT Blockly;
- питание внешнее: 100–240 В, 50/60 Гц;
- энергопотребление: 200 Вт;
- температурный режим работы: 10–60°.

Для проведения экспериментов был разработан программный комплекс на языке Python (версия 3.11). Выбор протокола обусловлен наличием стабильных библиотек взаимодействия с манипулятором через DOBOTAPI. Разработанное программное обеспечение реализует технологическую операцию типа «взять – переложить». В ходе работы манипулятор осуществлял непрерывное движение, базирующееся на потоковой генерации управляющих микрокоманд. В качестве опорного каркаса использовалась базовая траектория из 50 контрольных точек. Для обеспечения строгой методологической идентичности экспериментальных условий эта траектория была сгенерирована случайным образом в пределах допустимой рабочей области однократно до начала тестирования и сохранена в виде статического массива данных. Именно этот заранее сформированный эталонный набор координат загружался алгоритмом при тестировании каждого из оцениваемых транспортных протоколов. Таким образом, итоговое количество переданных команд за сеанс связи напрямую и объективно отражает пропускную способность протокола при непрерывном удержании управляющего потока.

Схема сетевой модели представлена на рисунке 3. В данной конфигурации РС#2 выступает сервером, иницирующим и координирующим передачу команд. Он подключается к Ethernet-коммутатору, обеспечивающему связь с клиентом (РС#1), который управляет манипулятором через USB. Такое разделение позволяет независимо изменять сетевые параметры, такие как задержка, пропускная способность и вероятность потери пакетов, моделируя различные условия сети.

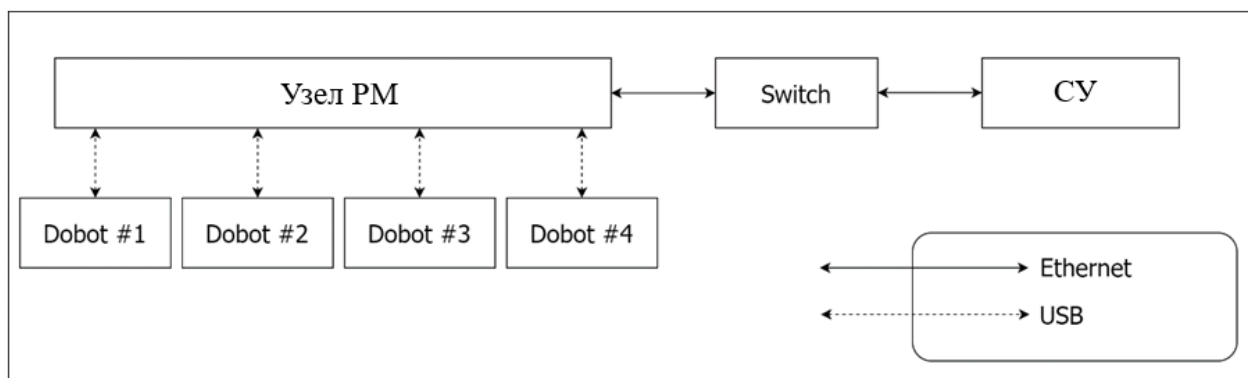


Рис. 3. Схема сетевой модели с участием DOBOT Magician

Выбор Ethernet обусловлен его широким распространением, предсказуемым поведением и возможностями мониторинга. Данная модель может масштабироваться до многопользовательских конфигураций с использованием Wi-Fi, 5G или VPN-каналов, однако настоящее исследование сосредоточено на стабильной проводной инфраструктуре.

Для генерации случайных координат была рассмотрена схема пределов перемещений манипулятора DOBOT Magician, которая представляет собой пространство, вычисляемое через разницу двух полусфер, включая ограничения передвижения самого манипулятора: из четырех полуплоскостей, изображенных на рисунке 4, в допустимые точки положения РМ входит любая комбинация  $x, y, z$ , находящаяся в пределах  $x \in (-120, 320)$ ;  $y \in (-300, 320)$ ;  $z > 0$  и удовлетворяющая нижеописанным условиям.

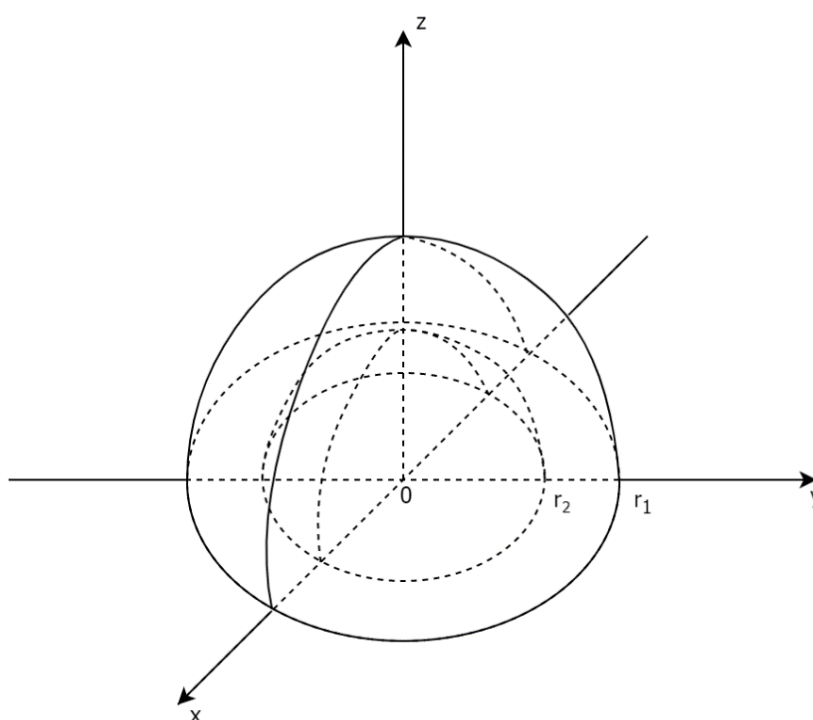


Рис. 4. Схема пределов перемещений DOBOT Magician

Условие включения сгенерированной координаты в массив допустимых точек можно выразить следующим образом:

$$r_1^2 < x^2 + y^2 + z^2 < r_2^2,$$

где  $r_1 = 180$ ;  $r_2 = 300$  – радиусы внутренней и внешней сфер соответственно, рассчитанные на основе известных координатных плоскостей манипулятора;  $x, y, z$  – координаты точки. Пусть сфера с радиусом  $r_1$  обозначается как  $\Omega_1$ , а сфера с радиусом  $r_2$  – как  $\Omega_2$ .

В таблице 3 представлен фрагмент массива сгенерированных управляющих команд. Каждая строка содержит пять параметров: пространственные координаты  $x, y, z$ , угол поворота захвата  $r$  и бинарный сигнал состояния вакуумного

захвата  $g$ . Таким образом, передаваемый пакет данных обеспечивает полную конфигурацию всех степеней свободы и исполнительных органов манипулятора в каждой точке траектории.

Таблица 3. Фрагмент полученных координат

$x; y; z; r; g$
151; 2; 136; 15; 0
173; -170; 42; 15; 0
198; 139; -13; 15; 0
178; 105; 97; 15; 0
187; 128; 5; 15; 0
...

В сценариях управления роботами передача команд от СУ к РМ осуществляется по каналу связи с ненулевой вероятностью ошибок. Вследствие этого возникает задача снижения избыточного трафика, обусловленного повторной передачей команд, при одновременном увеличении общей пропускной способности системы управления. Взаимодействие между СУ, СС и РМ организовано по модели замкнутого цикла (рисунок 5).

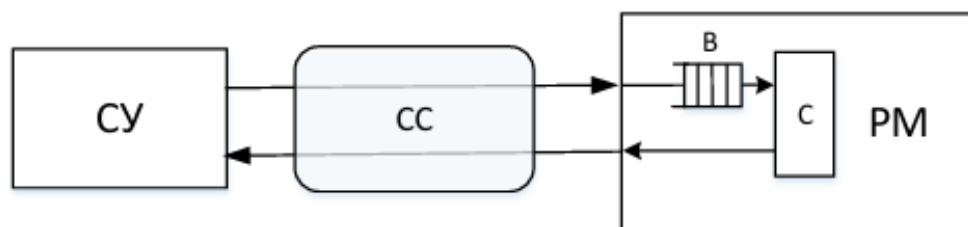


Рис. 5. Модель взаимодействия между СУ и РМ

Под СС в данном контексте понимается совокупность активного сетевого оборудования и каналов передачи данных, обеспечивающих двустороннюю доставку пакетов управляющих команд и сигналов подтверждения. Взаимодействие компонентов осуществляется по следующему алгоритму:

- 1) СУ формирует пакет из  $b$  команд и передает его в буфер РМ;
- 2) встроенный контроллер (С) РМ извлекает команды из буфера, после чего РМ подтверждает успешное получение пакета и последовательно выполняет команды;
- 3) после успешного выполнения пакета РМ запрашивает следующий набор команд;
- 4) в случае сбоя выполнения буфер очищается, и СУ повторно передает пакет.

Размер серии команд  $b$  в предложенной модели не является жестко заданной величиной. В ходе эксперимента алгоритм управления динамически адаптировал количество команд в одном отправляемом пакете в зависимости от успешности выполнения предыдущих операций (оценка рассинхронизации координат и потеря в сети). Таким образом, приводимые далее значения размера серии (например, 1,11 для TCP или 3,005 для QUIC) представляют собой не расчетные константы, а усредненные эмпирические показатели, к которым система автоматически сходилась в реальных сетевых условиях для каждого конкретного протокола.

Группировка нескольких команд в одну серию (пакет) позволяет существенно сократить долю служебной информации  $L$  в общем объеме сетевого трафика. Эффективность передачи напрямую зависит от обеспечиваемого транспортным протоколом размера серии  $b$ , соблюдая баланс между скоростью доставки и минимизацией служебного трафика. Данный подход целесообразен для использования в системах реального времени, где требуется регулярное управление исполнительными механизмами.

### Тестирование и анализ результатов

Мониторинг и анализ сетевого трафика представляют собой ключевой этап оценки производительности коммуникационных протоколов. В качестве основного инструмента был выбран Wireshark.

Для автоматизации обработки данных, полученных с помощью Wireshark, была разработана специализированная программная система на Python 3.11. Основными задачами этого инструмента являются анализ файлов захвата пакетов (PCAP, аббр. от англ. Packet Capture), визуализация временных рядов и расчет ключевых показателей, таких как скорость передачи пакетов, коэффициент потеря пакетов и размер серии команд. Реализация основана на библиотеках `pyshark` и `matplotlib`, которые обеспечивают эффективные механизмы разбора PCAP-файлов и построения графических представлений.

Разработанная функция `parse_pcap` отвечает за извлечение данных и реализует потоковую обработку сетевых пакетов, устраняя необходимость загрузки всего файла в оперативную память. Этот подход особенно важен при работе с большими наборами данных, типичными для промышленных сетей, где генерируются тысячи пакетов в секунду. Алгоритм классифицирует трафик по протоколам (TCP, UDP, QUIC) и извлекает размер полезной нагрузки, что позволяет оценить средний размер серии команд.

Визуализация результатов выполняется с помощью метода `plot_time_series`, который отображает динамику соотношения команд и размеров пакетов в виде линейных графиков. Кроме того, программное обеспечение формирует сводные таблицы с общими статистическими показателями, такими как общее время передачи, количество пакетов и количество команд, что упрощает сравнительный анализ.

Разработанный инструмент автоматизирует процесс извлечения метрик из PCAP-файлов, исключая необходимость ручного разбора и визуальной фильтрации пакетов в интерфейсе Wireshark. Это существенно ускоряет сравнительный

анализ больших массивов сетевого трафика и минимизирует вероятность ошибок, связанных с человеческим фактором, при обработке данных. Кроме того, его функциональность может быть расширена за счет интеграции алгоритмов машинного обучения для прогнозирования аномалий в сети, что является перспективным направлением для дальнейших исследований.

Тестирование проводилось в контролируемых условиях, моделирующих реальные промышленные сценарии. В частности, для эмуляции помех вносилось искусственное затухание на уровне более 23 дБ, а высокая плотность соединений имитировалась путем параллельной рассылки данных на несколько целевых сетевых портов конечных устройств.

Критерии оценки включали:

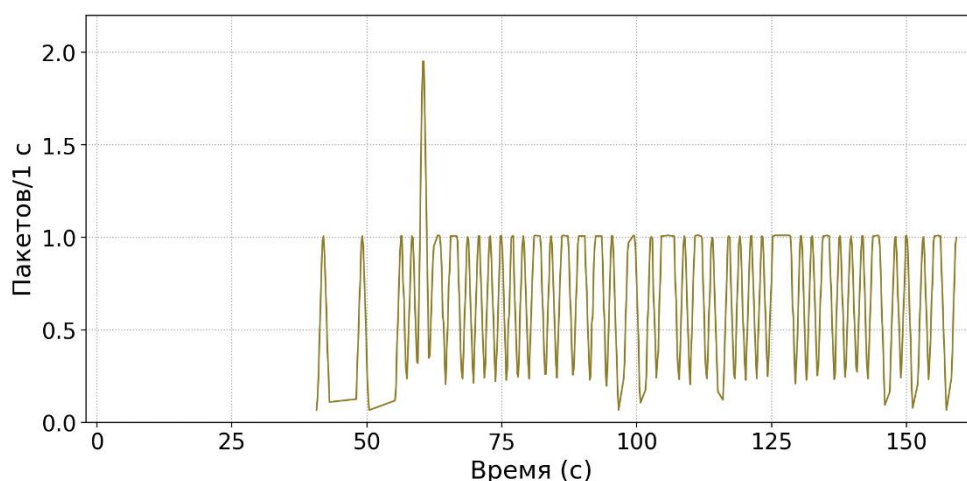
- пропускную способность (количество команд в секунду);
- надежность (коэффициент потерь пакетов);
- размер серии команд  $b$ ;
- безопасность (наличие шифрования и аутентификации).

Каждый протокол тестировался в идентичных условиях для обеспечения объективного сравнения. Под идентичностью условий в данном случае понимается использование единого аппаратно-программного стенда, генерация одинаковой базовой траектории (50 контрольных точек) и фиксированные физические параметры сети. Наблюдаемые различия в итоговом количестве успешно переданных пакетов и команд за сеанс обусловлены исключительно внутренней архитектурой самих протоколов (наличием задержек квитирования, скоростью обработки очередей и эффективностью алгоритмов агрегации данных). Результаты фиксировались с помощью Wireshark и обрабатывались скриптом на Python, что позволило выполнить детальный статистический анализ и визуализировать динамику передачи.

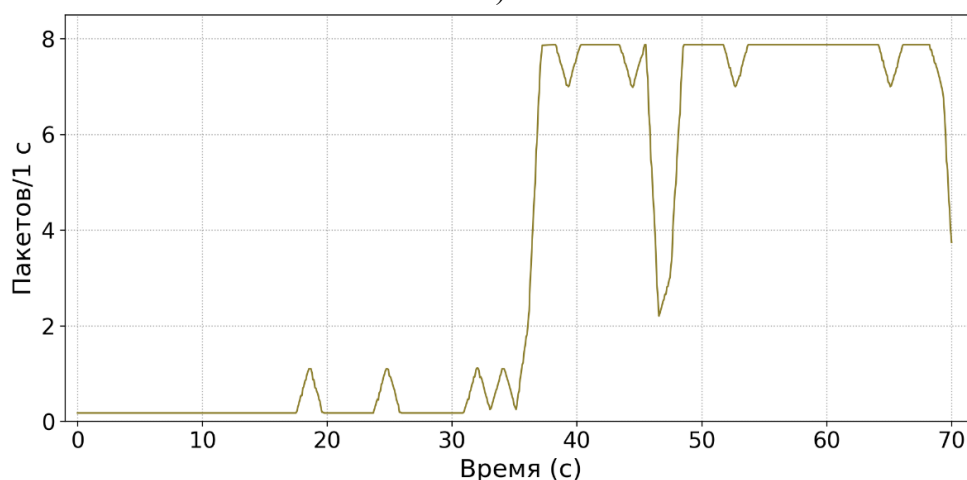
## Сравнительный анализ протоколов TCP, UDP и QUIC

*Протокол TCP* с установлением соединения обеспечивает гарантированную доставку данных с помощью механизмов подтверждения и повторной передачи потерянных пакетов. Однако механизмы подтверждения и повторной передачи порождают значительный объем служебного трафика. Как показано на рисунке 6, скорость передачи команд по TCP составила 0,58 команд/с, что значительно меньше, чем у UDP и QUIC. Низкая пропускная способность в основном обусловлена трехсторонним рукопожатием, буферизацией на стороне клиента и необходимостью синхронной обработки каждой команды.

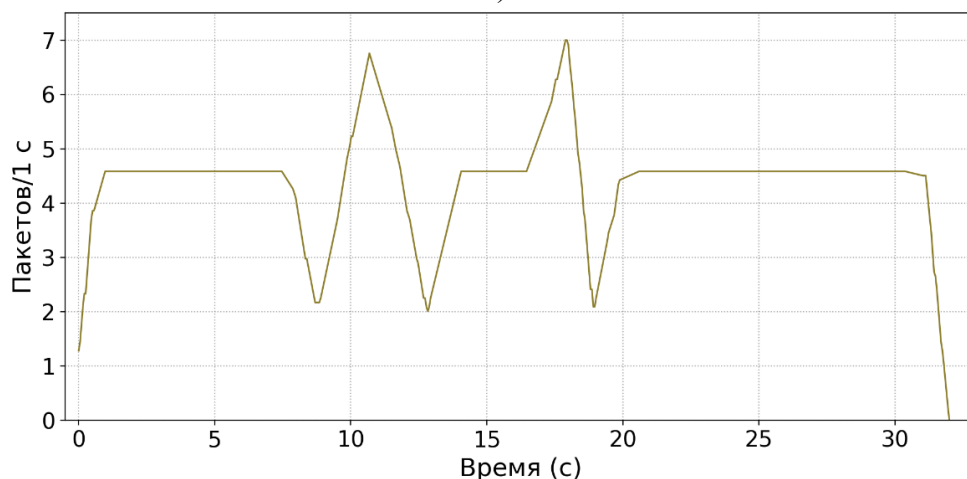
Для оценки динамики трафика использовались графики ввода / вывода, отображающие интенсивность передачи пакетов в единицу времени (пакетов/с). Это позволяет визуализировать равномерность потока данных и выявить периоды простоя или перегрузки канала.



а)



б)



в)

Рис. 6. График ввода / вывода для реализации TCP (а), UDP (б), QUIC (в)

График полезной нагрузки (рисунок 7а) демонстрирует малый размер серии: среднее значение составило 1,11 (размер пакетов варьировался от 1,0 до 5,0). Такая неэффективность объясняется относительно большим размером заголовка TCP (20 байт), что становится критичным при передаче коротких команд, типичных для управления роботом. Например, одна управляющая команда, включающая координаты  $x$ ,  $y$ ,  $z$ , угол поворота  $r$  и состояние захвата  $g$ , требует 77 байт

полезной нагрузки. При этом суммарный служебный трафик протоколов составляет значительную долю трафика, что подчеркивает необходимость оптимизации размера серии команд. Сводные результаты измерений для всех трех протоколов представлены в таблице 4.

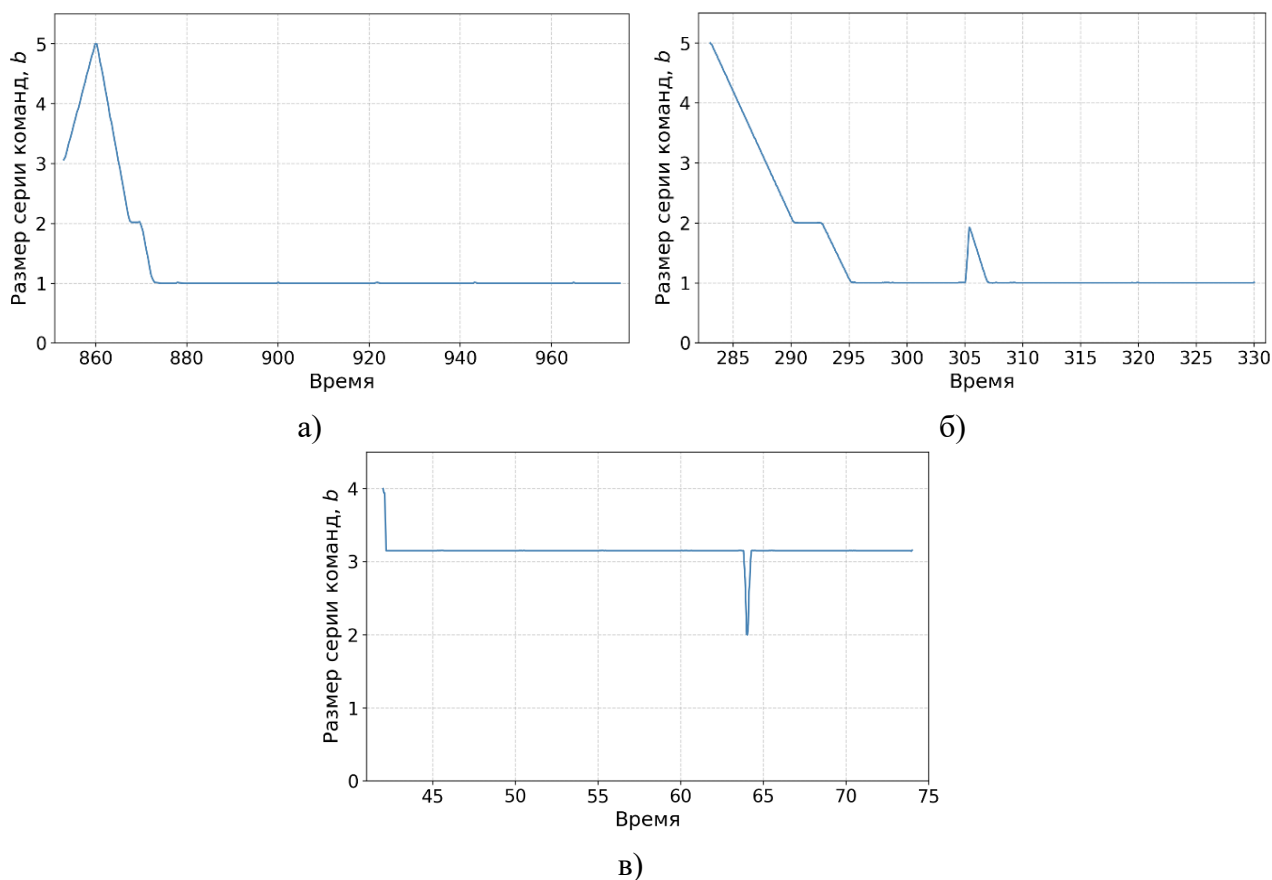


Рис. 7. График полезной нагрузки для реализации TCP (а), UDP (б), QUIC (в)

Таблица 4. Сравнительная статистика сетевого трафика для протоколов TCP, UDP, QUIC

Параметр	Значение		
	TCP	UDP	QUIC
Общее количество пакетов	59	247	193
Общее время, с	113,63	43,65	31,93
Средняя скорость передачи пакетов, пакетов/с	0,51	5,66	6
Общее количество команд	66	253	580
Средний размер серии команд $b$	1,11	1,02	3,005
Средняя скорость передачи команд, команд/с	0,58	5,8	18,16
Зафиксированный уровень потерь пакетов, %	—	3,88	< 1
Безопасность	—	—	TLS 1.3

Эксперимент также выявил проблему с очередями на стороне клиента. При высоких темпах генерации команд сервером задержки подтверждений приводят к буферизации на клиенте, увеличивая время отклика системы. В промышленных условиях такая задержка отклика может вызывать десинхронизацию производственных линий и аварийные остановки.

*Протокол UDP* в отличие от TCP исключает фазы установления соединения и подтверждений, что обеспечивает минимальную задержку. Скорость передачи команд по UDP достигла 5,66 команд/с, что в 10 раз превышает производительность TCP. Однако отсутствие встроенных механизмов восстановления потерянных пакетов оказалось существенным недостатком: уровень потерь пакетов составил 3,88 %, что неприемлемо для систем управления в реальном времени.

Резкие колебания на графике ввода / вывода (рисунок 6б) свидетельствуют о нестабильной работе канала. Например, между 15-й и 20-й секундами наблюдалось заметное падение скорости передачи из-за перегрузки сети. График полезной нагрузки (рисунок 7б) подтвердил малый размер серии со средним значением  $b = 1,02$  (близко к показателям TCP).

UDP демонстрировал преимущества в сценариях с избыточным управлением, где потеря отдельных команд не критична. Например, при транспортировке объекта РМ может продолжать движение на основе последних полученных координат, игнорируя отсутствующие пакеты. Однако в высокоточных задачах, таких как лазерная гравировка, даже незначительная потеря данных приводит к дефектам на выходе.

*Протокол QUIC* показал наибольшую эффективность среди протестированных протоколов. Скорость передачи достигла 18,16 команд/с при уровне потерь пакетов менее 1 %, что объясняется использованием мультиплексирования потоков и адаптивных алгоритмов управления перегрузкой. График ввода / вывода (рисунок 6в) демонстрирует стабильную работу без резких просадок скорости передачи: это свидетельствует об эффективном восстановлении пакетов. Скорость передачи пакетов варьировалась от 4 до 7 пакетов/с, в среднем составляя 6 пакетов/с. Отсутствие сбоев пропускной способности подтверждает способность QUIC эффективно справляться с потерями.

График полезной нагрузки (рисунок 7в) показывает среднее значение 3,005 команд на пакет, что в три раза выше, чем у TCP. Это улучшение достигается за счет агрегирования данных из нескольких потоков в рамках одного соединения. Например, команды управления движением, телеметрия и диагностические данные могут передаваться одновременно, минимизируя служебный трафик протокола.

Ключевое преимущество QUIC заключается в интеграции TLS 1.3, обеспечивающей сквозное шифрование. Это исключает риск перехвата критически важных команд, таких как параметры движения или конфигурации оборудования. Кроме того, QUIC поддерживает миграцию соединений, позволяя роботам переключаться между точками доступа без прерывания сеансов связи.

Сравнительный анализ трех транспортных протоколов выявил их сильные и слабые стороны применительно к промышленным сценариям. TCP, несмотря на высокую надежность, продемонстрировал неприемлемо низкую пропускную способность и значительный объем служебного трафика, что делает его непригодным для систем реального времени. UDP обеспечивает высокую скорость передачи, однако отсутствие встроенных механизмов восстановления потерянных пакетов ограничивает его использование в критически важных приложениях.

Протокол QUIC, объединяющий преимущества UDP и TCP, показал наилучшие результаты по всем критериям оценки. Его возможности по мультиплексированию потоков данных, адаптации к сетевой динамике и встроенной безопасности делают его оптимальным выбором для промышленной автоматизации. Например, в системах управления конвейерными линиями QUIC обеспечивает синхронизацию десятков роботов с задержками менее 1 мс, что соответствует требованиям приложений тактильного интернета.

Интеграция QUIC в сети IoT открывает широкие перспективы повышения эффективности и надежности производственных процессов. Одним из ключевых преимуществ протокола является минимальная задержка передачи. Механизм 0-RTT handshake (установление соединения без предварительного обмена пакетами) позволяет устанавливать соединения за один обмен пакетами, а мультиплексирование потоков обеспечивает параллельную передачу данных без межпоточной блокировки, существенно снижая задержки. Это особенно важно в приложениях, требующих высокой точности и оперативности, например при управлении роботизированными манипуляторами, где задержки даже в миллисекунды могут вызвать десинхронизацию или замедление производственного процесса.

Помимо снижения задержки, QUIC демонстрирует высокую устойчивость к нестабильным сетевым условиям, что делает его особенно подходящим для промышленных сред с сильными электромагнитными помехами. Заводы со сварочным оборудованием и тяжелой техникой часто создают сетевую нестабильность; тем не менее QUIC обеспечивает надежную доставку данных благодаря эффективным механизмам восстановления потерянных пакетов и поддержке динамической миграции соединений. Например, при автоматическом переключении устройства с Wi-Fi на 5G соединение сохраняется, и коммуникация продолжается без прерывания.

Энергоэффективность устройств также является важным фактором. В отличие от TCP, простой механизм управления перегрузкой в QUIC снижает вычислительные требования к периферийным устройствам, увеличивая автономность мобильных роботов, беспроводных сенсоров и промышленных модулей с батарейным питанием.

## Заключение

В ходе проведенного исследования была успешно решена научно-практическая задача повышения эффективности доставки команд для систем дистанционного управления промышленными РМ. Главной целью работы являлся поиск оптимального количества команд в серии  $b^*$ , обеспечивающего максимум целевой функции эффективности передачи данных и наивысшую пропускную способность канала связи в условиях IoT.

Для достижения поставленной цели была разработана архитектура замкнутого цикла, в которой алгоритм управления непрерывно оценивал уровень рассинхронизации координат манипулятора и вероятность сетевых потерь, динамически

адаптируя размер отправляемой серии. Это позволило провести объективное эмпирическое сравнение транспортных протоколов TCP, UDP и QUIC в строго идентичных методологических условиях.

По результатам анализа сетевого трафика установлено следующее:

Во-первых, классические транспортные протоколы продемонстрировали низкую эффективность инкапсуляции данных. Для протокола UDP из-за отсутствия встроенных механизмов восстановления потерь на транспортном уровне доля утерянных пакетов составила 3,88 %, что вынудило алгоритм динамической адаптации снизить средний размер серии до одной команды на пакет для минимизации риска критической рассинхронизации. Протокол TCP обеспечил надежную доставку (0 % потерь), однако из-за проблемы межпоточной блокировки, буферизации на стороне клиента и высокого служебного трафика средний размер серии составил всего 1,11 команд на пакет, а итоговая пропускная способность упала до неприемлемых 0,58 команд/с.

Во-вторых, протокол QUIC показал самую высокую эффективность по сравнению с другими тестируемыми протоколами в сценариях высокоточного управления IoT. Благодаря механизмам независимого мультиплексирования потоков и асинхронного восстановления потерь, математический аппарат системы смог безопасно уплотнить полезную нагрузку, доведя средний размер серии до 3 команд на пакет.

В-третьих, использование QUIC гарантирует криптографическую безопасность контура управления за счет обязательной интеграции стандарта TLS 1.3, защищающего управляющие телеметрические данные от перехвата и злонамеренной модификации.

Таким образом, в результате проведенных экспериментов доказано, что внедрение протокола QUIC совместно с математически обоснованным механизмом динамической адаптации размера серии является оптимальным решением для распределенных киберфизических систем и роботизированных производств концепции «Индустрия 4.0», требующих сверхнизких задержек, высокой пропускной способности и отказоустойчивости. Дальнейшие исследования целесообразно направить на интеграцию данного транспортного протокола с технологиями периферийных вычислений и сетями связи 6G.

## Литература

1. Киричек Р. В., Кулик В. А. Исследование и генерация трафика промышленного Интернета Вещей // Труды учебных заведений связи. 2019. Т. 5. № 3. С. 27–36. DOI: 10.31854/1813-324X-2019-5-3-27-36. EDN: JQBTYU
2. Кулик В. А., Вахитов С. А., Киричек Р. В. Модель семантического преобразования пакетов для гетерогенного шлюза промышленного интернета вещей // Электросвязь. 2020. № 3. С. 49–54. DOI: 10.34832/ELSV.2020.4.3.007. EDN: ANXFTE
3. Lee E. A. Cyber Physical Systems: Design Challenges // Proceedings of the 11<sup>th</sup> IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC, 05–07 May 2008, Orlando, USA). 2008. PP. 363–369. DOI: 10.1109/ISORC.2008.25.

4. Rajkumar R., Lee I., Sha L., Stankovic J. Cyber-Physical Systems: The Next Computing Revolution // Proceedings of the 47<sup>th</sup> Design Automation Conference (13–18 June 2010, Anaheim, USA). 2010. PP. 731–736. DOI: 10.1145/1837274.1837461. EDN: OCIABJ

5. Rani D., Gill N. S. Review of Various IoT Standards and Communication Protocols // International Journal of Engineering Research and Technology. 2019. Vol. 12. Iss. 5. PP. 647–657.

6. Peng H., Tärneberg W., Fitzgerald E., Kihl M. Performance Evaluation of QUIC Vs. TCP for Cloud Control Systems // Proceedings of the 31<sup>st</sup> International Conference on Software, Telecommunications and Computer Networks (SoftCOM, 21–23 September 2023, Split, Croatia). 2023. DOI: 10.23919/SoftCOM58365.2023.10271592

7. Amet M., Thomas L., Song Y.-Q. A Performance Evaluation of QUIC in Real-Time Networks // Proceedings of the 32nd International Conference on Real-Time Networks and Systems (RTNS, 6–8 November 2024, Porto, Portugal). 2024. PP. 255–265. DOI: 10.1145/3696355.3699698

8. Silva D. R. C., Oliveira G. M. B., Silva I., Ferrari P., Sisinni E. Latency Evaluation for MQTT and WebSocket Protocols: An Industry 4.0 Perspective // IEEE Symposium on Computers and Communications (ISCC, 25–28 June 2018, Natal, Brazil). IEEE, 2018. PP. 1233–1238.

**Статья поступила 8 октября 2025 г.  
Одобрена после рецензирования 17 февраля 2026 г.  
Принята к публикации 22 апреля 2026 г.**

### **Информация об авторах**

*Белов Максим Алексеевич* – студент 1-го курса института магистратуры (группа МСТ-2561) Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. E-mail: belov.ma@sut.ru

*Кузьмина Екатерина Алексеевна* – выпускник института магистратуры Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. E-mail: kuzmina.ea@sut.ru

*Филина Надежда Романовна* – аспирант кафедры сетей связи и передачи данных Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. E-mail: filina.nr@sut.ru

*Мутханна Аммар Салех Али* – доктор технических наук, профессор кафедры сетей связи и передачи данных Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. E-mail: muthanna.asa@sut.ru

<https://doi.org/10.31854/2307-1303-2026-14-1-35-56>  
EDN: YKGWDF

## Analysis of Communication Protocols for Industrial Robot Control Systems

 M. A. Belov,  E. A. Kuzmina,  N. R. Filina ,  A. S. A. Muthanna

The Bonch-Bruевич Saint Petersburg State University of Telecommunications,  
St. Petersburg, 193232, Russian Federation

**Purpose.** The development of the Industrial Internet of Things (IIoT) within the framework of the “Industry 4.0” paradigm is accompanied by an increase in the number of connected devices, growing demands for data transmission rate, and stringent latency constraints. In real-time control systems, even millisecond-level delays may lead to desynchronization of robotic mechanisms, production line interruptions, and economic losses. Widely used transport protocols, TCP and UDP, exhibit significant limitations: the former provides reliability at the cost of increased latency and overhead, while the latter ensures minimal delay without delivery control, which is critical for industrial processes. The purpose of this paper is to perform a comparative analysis of TCP, UDP, and QUIC transport protocols under conditions close to industrial scenarios and to evaluate their applicability for robotic control and reliable IIoT communications. **Methods.** Network analysis was conducted using the Wireshark packet analyzer and a custom-developed Python-based software complex incorporating the pyshark and matplotlib libraries. The study included PCAP file analysis, construction of time series, calculation of key metrics (command transmission rate, packet loss ratio, and command series size  $b$ ), and their visualization. The elements of **novelty** consist in the comprehensive comparison of three transport protocols in the context of robotic manipulator control, where millisecond-level delays are critical; the application of QUIC, a relatively new protocol not previously examined in applied IIoT scenarios; and the development of a software tool enabling automated traffic analysis and classification in industrial environments. **Results.** Experimental results show that TCP provides zero packet loss but an unacceptably low command rate (0.58 commands/s). UDP achieves an order-of-magnitude higher rate (5.8 commands/s) but suffers from 3.88 % packet loss, rendering it unsuitable for high-precision operations. QUIC demonstrates the best performance: 18.16 commands/s with less than 1 % packet loss and a threefold increase in the average command series size  $b$ . **Theoretical relevance.** The obtained results confirm the prospects of using the QUIC protocol in IIoT industrial networks. Its adoption ensures minimal latency and high transmission reliability while significantly reducing network load through stream multiplexing. **Practical implementation** may involve integration of QUIC into robotic control systems, sensor networks, and production lines, aligning with the requirements for scalable and secure next-generation communications.

**Key words:** Industrial Internet of Things, transport protocol, TCP, UDP, QUIC, industrial robotics, latency, data transmission reliability, network efficiency

### References

1. Kirichek R., Kulik V. Industrial Internet of Things Traffic Research and Generation // Proceedings of Telecommunication Universities. 2019. Vol. 5. Iss 3. PP. 27–36. DOI: 10.31854/1813-324X-2019-5-3-27-36. EDN: JQBTYU
2. Kulik V. A., Vakhitov S. A., Kirichek R. V. Semantic Packet Conversion Model for Industrial Internet of Things Heterogeneous Gateway // Elektrosvyaz. 2020. Iss. 3. PP. 49–54. DOI: 10.34832/ELSV.2020.4.3.007. EDN: AHXFTF
3. Lee E. A. Cyber Physical Systems: Design Challenges // Proceedings of the 11<sup>th</sup> IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC, 05–07 May 2008, Orlando, USA). 2008. PP. 363–369. DOI: 10.1109/ISORC.2008.25.

4. Rajkumar R., Lee I., Sha L., Stankovic J. Cyber-Physical Systems: The Next Computing Revolution // Proceedings of the 47<sup>th</sup> Design Automation Conference (13–18 June 2010, Anaheim, USA). 2010. PP. 731–736. DOI: 10.1145/1837274.1837461. EDN: OCIABJ

5. Rani D., Gill N. S. Review of Various IoT Standards and Communication Protocols // International Journal of Engineering Research and Technology. 2019. Vol. 12. Iss. 5. PP. 647–657.

6. Peng H., Tärneberg W., Fitzgerald E., Kihl M. Performance Evaluation of QUIC Vs. TCP for Cloud Control Systems // Proceedings of the 31<sup>st</sup> International Conference on Software, Telecommunications and Computer Networks (SoftCOM, 21–23 September 2023, Split, Croatia). 2023. DOI: 10.23919/SoftCOM58365.2023.10271592

7. Amet M., Thomas L., Song Y.-Q. A Performance Evaluation of QUIC in Real-Time Networks // Proceedings of the 32nd International Conference on Real-Time Networks and Systems (RTNS, 6–8 November 2024, Porto, Portugal). 2024. PP. 255–265. DOI: 10.1145/3696355.3699698

8. Silva D. R. C., Oliveira G. M. B., Silva I., Ferrari P., Sisinni E. Latency Evaluation for MQTT and WebSocket Protocols: An Industry 4.0 Perspective // IEEE Symposium on Computers and Communications (ISCC, 25–28 June 2018, Natal, Brazil). IEEE, 2018. PP. 1233–1238.

### Information about Authors

*Belov Maksim* – 1<sup>st</sup> Year Master’s Student (The Bonch-Bruevich Saint Petersburg State University of Telecommunications). E-mail: belov.ma@sut.ru

*Kuzmina Ekaterina* – Master of Science (The Bonch-Bruevich Saint Petersburg State University of Telecommunications). E-mail: kuzmina.ea@sut.ru

*Filina Nadezhda* – Postgraduate at the Department of Communication and Data Transmission (The Bonch-Bruevich Saint Petersburg State University of Telecommunications). E-mail: filina.nr@sut.ru

*Muthanna Ammar* – D. Sc. in Engineering, Professor at the Department of Communication and Data Transmission (The Bonch-Bruevich Saint Petersburg State University of Telecommunications). E-mail: muthanna.asa@sut.ru