

ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ INFORMATION TECHNOLOGY, COMPUTER SCIENCE AND MANAGEMENT



УДК 004.738.5

Оригинальное эмпирическое исследование

<https://doi.org/10.23947/2687-1653-2026-26-2-2228>

Эффективность и перспективы экспериментального протокола Quick User Datagram Protocol Internet Connections

Д. Рахмани^{id}, С.П. Сухарев^{id}✉

Московский технический университет связи и информатики, г. Москва, Российская Федерация

✉ s.p.suharev@mtuci.ru

EDN: CNWWGW

Аннотация

Введение. Обеспечение и повышение доступности веб-ресурсов в сети Интернет представляет собой актуальную задачу для разработчиков информационных систем. Критическую роль в доступности страниц по протоколу Hypertext Transfer Protocol¹ (HTTP) играет версия протокола и его реализация на транспортном уровне. Протокол Quick User Datagram Protocol Internet Connections² (QUIC), разработанный компанией Google, позволяет добиться прироста скорости загрузки ресурсов за счёт применения протокола User Datagram Protocol³ (UDP) в HTTP/3. Однако QUIC имеет статус экспериментального, а в существующих исследованиях основное внимание уделяется теоретическим аспектам или общим показателям в глобальной сети. При этом остаются недостаточно изученными: одновременное сопоставление трёх версий протокола в единых контролируемых условиях, практические сложности конфигурирования и влияние алгоритмов контроля перегрузок на прикладные метрики; трудозатраты на внедрение и настройку; усилия по настройке и количественная оценка выигрыша в контролируемых условиях. Указанные пробелы формируют разрыв между теоретическими ожиданиями и практической реализацией. Поэтому целью данного исследования явилась экспериментальная оценка прикладной эффективности HTTP/3 (QUIC) в контролируемых условиях на едином тестовом стенде, включающая сопоставление HTTP/1.1, HTTP/2, HTTP/3, анализ влияния алгоритмов контроля перегрузок CUBIC и BBR и документирование процедуры конфигурирования HTTP/3-сервера.

Материалы и методы. Для исследования был развёрнут тестовый стенд на основе виртуального сервера под управлением операционной системы (ОС) Linux и веб-сервера nginx с поддержкой HTTP/1.1, HTTP/3 (QUIC), а также алгоритмов контроля перегрузок CUBIC и Bottleneck Bandwidth and Round-trip propagation time⁴ (BBR). В качестве клиента использовался браузер Google Chrome в сети 4G. Производительность оценивалась по метрике Time to First Byte⁵ (TTFB), скорости загрузки файлов и времени полной загрузки веб-страницы. Замеры выполнялись многократно с использованием Chrome DevTools⁶ и клиентских скриптов. В статье подробно описан процесс настройки сервера для работы с HTTP/3.

Результаты исследований. Эксперименты показали, что применение HTTP/3 (QUIC) сокращает время до первого байта на 23,06 % и ускоряет полную загрузку страницы на 9,5 % по сравнению с HTTP/1.1. Теоретическая модель прогнозировала снижение TTFB на 71,43 % за счёт объединённого рукопожатия QUIC и TLS 1.3. Выявленное расхождение обусловлено особенностями обработки UDP-трафика операторами, экспериментальным статусом реализации и нестабильностью мобильного канала. При скачивании крупных файлов алгоритмы CUBIC и

¹ Протокол передачи гипертекста (англ.).

² Быстрые подключения к интернету по протоколу пользовательских датаграмм (англ.).

³ Протокол пользовательских датаграмм (англ.).

⁴ Пропускная способность узкого места и время круговой задержки распространения (англ.).

⁵ Время до первого байта (англ.).

⁶ Инструменты разработчика (англ.).

BBR обеспечили сопоставимую среднюю скорость ($\approx 13,12$ и $12,75$ МБ/с соответственно). Однако BBR за первые три секунды передал на 18,2 % больше данных, демонстрируя более быстрый выход на рабочую скорость и стабильный профиль передачи.

Обсуждение. Практические результаты частично расходятся с теоретическими оценками: снижение задержки оказалось меньше прогнозируемого из-за особенностей реализации Transport Layer Security⁷ (TLS), обработки UDP-трафика провайдерами и характеристик оборудования. Показано, что преимущества QUIC/HTTP/3 наиболее заметны при множественных коротких запросах и высоких задержках. Преимущество BBR перед CUBIC реализуется не на длительных передачах, а при загрузке множества небольших ресурсов страницы — типичного сценария веб-взаимодействий. Для повышения достоверности оценки производительности планируется расширение экспериментов с различными сетевыми условиями, реализациями протокола и географически распределёнными клиентами.

Заключение. Исследование подтвердило преимущество HTTP/3 (QUIC): TTFB снизился на 23,06 %, время загрузки страницы — на 9,5 %. При этом теоретическая модель прогнозировала более существенное сокращение, что указывает на влияние особенностей реализации и сетевой среды. Сравнение CUBIC и BBR выявило преимущество BBR при передаче файлов малого объёма. Несмотря на сложность конфигурирования HTTP/3, переход оправдан для сервисов со значительным количеством ресурсов. Ограничения эксперимента требуют расширения исследований в различных сетевых сценариях.

Ключевые слова: QUIC, HTTP/3, протоколы транспортного уровня, алгоритмы контроля перегрузок, BBR, CUBIC, TTFB, производительность веб-приложений

Благодарности. Авторы выражают искреннюю признательность сотрудникам кафедры «Сетевые информационные технологии и сервисы» Московского технического университета связи и информатики за поддержку в проведении исследования. Особая благодарность выражается редакционной команде журнала за внимательное отношение к рукописи, оперативное сопровождение подготовки публикации и организацию процесса рецензирования.

Для цитирования. Рахмани Д., Сухарев С.П. Эффективность и перспективы экспериментального протокола Quick User Datagram Protocol Internet Connections. *Advanced Engineering Research (Rostov-on-Don)*. 2026;26(2):2228. <https://doi.org/10.23947/2687-1653-2026-26-2-2228>

Original Empirical Research

Efficiency and Prospects of the Experimental QUIC Protocol

Jahed Rahmani , Serafim P. Sukharev  

Moscow Technical University of Communication and Informatics, Moscow, Russian Federation

 s.p.suharev@mtuci.ru

Abstract

Introduction. Ensuring and improving the availability of web resources on the Internet is a crucial task for developers of information systems. A critical role in the page accessibility via the Hypertext Transfer Protocol (HTTP) is played by the protocol version and its transport-level implementation. The QUIC (Quick User Datagram Protocol Internet Connections) protocol, developed by Google, provides an increase in resource loading speed through the use of User Datagram Protocol (UDP) in HTTP/3. However, QUIC has an experimental status, and existing research primarily focuses on theoretical aspects or general performance metrics in the global network. At the same time, the following aspects remain insufficiently studied: simultaneous comparison of three protocol versions under unified controlled conditions, practical complexities of configuration and the effect of congestion control algorithms on application-level metrics, labor costs for implementation and configuration, tuning efforts, and quantifying gains under controlled condition.

These gaps create a disconnect between theoretical expectations and practical implementation. Therefore, the objective of this study is to experimentally evaluate the applied performance of HTTP/3 (QUIC) under controlled conditions on a unified testbed, including a comparison of HTTP/1.1, HTTP/2, and HTTP/3, an analysis of the impact of the CUBIC and BBR congestion control algorithms, and documentation of the HTTP/3 server configuration procedure.

⁷ Безопасность транспортного уровня (англ.).

Materials and Methods. A testbed was deployed based on a virtual server running the Linux operating system (OS) and the nginx web server supporting HTTP/1.1, HTTP/3 (QUIC), and congestion control algorithms CUBIC and Bottleneck Bandwidth and Round-trip propagation time (BBR). The Google Chrome browser over a 4G network was used as the client. Performance was evaluated using the Time to First Byte (TTFB) metric, file download speed, and total web page load time. Measurements were performed multiple times using Chrome DevTools and client-side scripts. The paper provides a detailed description of the server configuration process for enabling HTTP/3.

Results. The experiments showed that using HTTP/3 (QUIC) reduced the time to first byte by 23.06% and accelerated full page load by 9.5% compared to HTTP/1.1. The theoretical model predicted a TTFB reduction of 71.43% due to the combined QUIC and TLS 1.3 handshake. The observed discrepancy was attributed to the specifics of UDP traffic processing by internet service providers, the experimental status of the implementation, and mobile channel instability. When downloading large files, the CUBIC and BBR algorithms provided comparable average speeds (≈ 13.12 MB/s and 12.75 MB/s, respectively). However, BBR transmitted 18.2% more data within the first three seconds, demonstrating faster ramp-up to operational speed and a more stable transfer profile.

Discussion. Practical results partially differed from theoretical estimates: the observed latency reduction was lower than expected due to Transport Layer Security (TLS) implementation features, UDP traffic processing by Internet providers, and hardware characteristics. It is shown that the advantages of QUIC/HTTP/3 are most noticeable under conditions of multiple short requests and high latency. The advantage of BBR over CUBIC is realized not in long-duration transfers, but when loading numerous small page resources — a typical web interaction scenario. To improve the reliability of performance evaluation, further experiments are planned under various network conditions, protocol implementations, and geographically distributed clients.

Conclusion. The study confirmed the advantages of HTTP/3 (QUIC): TTFB decreased by 23.06%, and page load time by 9.5%. However, the theoretical model predicted a greater reduction, indicating the influence of implementation and network environment factors. The comparison of CUBIC and BBR revealed the advantage of BBR when transferring small-sized files. Despite the complexity of HTTP/3 configuration, the transition is justified for services with a significant number of resources. The experimental limitations indicate the need for further studies under different network scenarios.

Keywords: QUIC, HTTP/3, transport layer protocols, congestion control algorithms, BBR, CUBIC, TTFB, web performance

Acknowledgments. The authors sincerely acknowledge the Department of Network Information Technologies and Services, Moscow Technical University of Communications and Informatics, for their support in conducting this research. Special thanks are expressed to the journal editorial team for their attentive attitude to the manuscript, prompt support in the preparation of the publication, and organization of the peer-review process.

For Citation. Rahmani J, Sukharev SP. Efficiency and Prospects of the Experimental QUIC Protocol. *Advanced Engineering Research (Rostov-on-Don)*. 2026;26(2):2228. <https://doi.org/10.23947/2687-1653-2026-26-2-2228>

Введение. В настоящее время доступные веб-ресурсы представляют собой ценную базу знаний и неотъемлемую часть функционирования организаций различных направлений. Развитие рынка интернет-рекламы выступает ключевым фактором роста всей рекламной индустрии в Российской Федерации [1]. Обеспечение комфортного взаимодействия конечного пользователя с информационными системами в сети Интернет является приоритетной задачей для компаний любого масштаба. Исследования демонстрируют, что даже незначительное повышение производительности веб-страниц способно оказать существенное положительное влияние на финансовые показатели бизнеса [2, 3].

Передача данных в World Wide Web (WWW) осуществляется посредством протокола HTTP, имеющего три основные версии [4]. Поддержка современных протоколов веб-серверами способна благоприятно воздействовать на пользовательский опыт при работе с сайтами [5]. HTTP/3 базируется на протоколе QUIC [6], который разрабатывался как альтернатива Transmission Control Protocol⁸ (TCP), применявшемуся в предыдущих версиях HTTP [7], с целью повышения скорости и безопасности интернет-соединений.

Анализ существующей литературы позволяет зафиксировать как достигнутые результаты, так и существенные противоречия между ними. В [8] представлены результаты масштабного развёртывания QUIC в инфраструктуре Google: задержка ответов поискового сервиса сократилась на 8,0 % для настольных платформ и на 3,6 % для мобильных пользователей, частота прерываний видео воспроизведения снизилась на 18,0 % и 15,3 % соответственно. Авторы фиксируют, что выигрыш нарастает с увеличением метрики Round-Trip Time⁹ (RTT) и потерь пакетов, однако оказывается ниже теоретически ожидаемого — в том числе из-за ограничений реализации и спецификации сетевого оборудования операторов. Это означает, что перенос выводов на развёртывания умеренного масштаба требует самостоятельной экспериментальной проверки.

⁸ Протокол управления передачей (англ.).

⁹ Время круговой задержки (англ.).

В [9] сравнили QUIC и TCP в лабораторной среде с локально установленным сервером и констатировали минимальные различия между HTTP/2 и HTTP/3 при многопоточной передаче. В другой статье, [10], напротив, в аналогичном сценарии с уровнем потерь 12 % зафиксировали пятикратное преимущество HTTP/3 над HTTP/2. Данное противоречие обусловлено различием сетевых условий и остаётся неразрешённым применительно к штатному развёртыванию без симулированных нарушений сети. При этом работы не включают HTTP/1.1 как отправную точку для сравнений и не содержат данных по метрике TTFB.

В работе [11] были исследованы клиент-серверные протоколы с позиций отказоустойчивости и безопасности в реальных сетевых условиях и зафиксировали, что фактический прирост производительности при переходе на QUIC варьируется в зависимости от конфигурации промежуточного сетевого оборудования. Данный вывод согласуется с наблюдением авторов [8] о расхождении между теоретически ожидаемым и практически достигаемым выигрышем от протокола.

В обзорной работе о перспективах HTTP/3 и QUIC [12] показано, что совмещённое рукопожатие QUIC и встроенное шифрование TLS 1.3 формируют теоретическую основу для существенного снижения задержек. Однако учёные указывают на сложности практического развёртывания, обусловленные незрелостью поддержки протокола серверным программным обеспечением, а конкретная конфигурационная процедура в работе не приводится.

В [13] задокументировано, что 19,38 % запросов в крупном мобильном веб-сервисе приходится на фазу медленного старта, в течение которой статически заданный начальный размер окна перегрузок не соответствует реальной полосе пропускания. Тем самым исследование подтверждает, что механизм медленного старта алгоритма CUBIC создаёт измеримые задержки при передаче небольших файлов, из которых обычно состоят веб-страницы, что служит основанием для изучения BBR как альтернативного решения.

В статье [14] обоснован алгоритм BBR, использующий оценку пропускной способности канала и задержки распространения вместо потерь пакетов в качестве сигнала перегрузки. Развёртывание BBR в магистральной сети Google B4 обеспечило рост пропускной способности в 2–25 раз по сравнению с CUBIC. Вместе с тем в публикации анализируются агрегированные показатели пропускной способности, тогда как динамика скорости передачи во времени (характерная для CUBIC пилообразная форма графика в сравнении с более стабильным профилем BBR) экспериментально не визуализируется.

В работе [15] документируется, что в реализациях QUIC алгоритм CUBIC остаётся используемым по умолчанию, тогда как BBR предоставляется как опциональный. Проведённые авторами эксперименты в симуляторе ns-3 показывают, что BBR обеспечивает более высокую пропускную способность по сравнению с CUBIC при высоких задержках канала, однако при низкой полосе пропускания уступает ему по числу потерянных пакетов. При этом динамика скорости передачи во времени в работе не анализируется, что оставляет открытым вопрос о стабильности скоростного профиля каждого из алгоритмов.

Приведённый обзор позволяет выделить три взаимосвязанных пробела. В имеющихся работах HTTP/1.1, HTTP/2 и HTTP/3 не сопоставляются одновременно в единых контролируемых условиях: большинство исследований ограничено сравнением HTTP/2 и HTTP/3. Сведения о производительности протоколов существенно расходятся в зависимости от условий тестирования, а оценка в штатных условиях без прокси-серверов и искусственно вносимых потерь пакетов в эмулируемой сетевой среде представлена недостаточно. При этом практическая процедура конфигурирования стандартного веб-сервера для HTTP/3 в научной литературе не задокументирована. Таким образом, научный пробел состоит в отсутствии работ, в которых в рамках единой методологической схемы одновременно: сопоставляются HTTP/1.1, HTTP/2 и HTTP/3 в контролируемых условиях; документируется процедура конфигурирования стандартного веб-сервера для поддержки HTTP/3; оценивается влияние алгоритмов контроля перегрузок CUBIC и BBR на прикладные метрики веб-доступа. Восполнение указанного пробела определяет цель и постановку задач данного исследования.

Целью данного исследования являлась экспериментальная оценка прикладной эффективности HTTP/3 (QUIC) в контролируемых условиях на едином тестовом стенде путём сопоставления HTTP/1.1, HTTP/2 и HTTP/3, анализа влияния алгоритмов контроля перегрузок CUBIC и BBR и документирования процедуры конфигурирования HTTP/3-сервера.

Для достижения поставленной цели были решены следующие задачи:

- развернуть тестовый стенд с поддержкой HTTP/1.1, HTTP/2 и HTTP/3;
- зафиксировать последовательность конфигурирования веб-сервера для работы с HTTP/3;
- измерить и сопоставить TTFB и время полной загрузки страницы для разных версий HTTP;
- сравнить поведение алгоритмов CUBIC и BBR при передаче файла большого объема;
- интерпретировать полученные результаты с учётом теоретических ожиданий и опубликованных исследований.

Материалы и методы

Оборудование. Для проведения экспериментов был подготовлен тестовый стенд, состоящий из арендованной в датацентре IXcellerate в Москве виртуальной машины с характеристиками: 1 Virtual Central Processing Unit¹⁰ (vCPU) 2.4 ГГц, 2 ГБ оперативной памяти под управлением операционной системы Ubuntu 22.04.5 Long-Term Support¹¹ (LTS) и личной рабочей станции под Windows 11. Оба устройства имеют выход в интернет. У виртуальной машины есть статический Internet Protocol address¹² (IP-адрес) в сети интернет с доменным именем serafimdev.com.

План исследования. В рамках работы было запланировано провести три эксперимента: сравнение скорости установления соединения с использованием HTTP/1.1, HTTP/2, HTTP/3; сравнение скорости загрузки тестовой страницы через протоколы HTTP/1.1, HTTP/2, HTTP/3; сравнение скорости передачи информации с использованием протоколов контроля перегрузок CUBIC и BBR.

Реализация экспериментов осуществлялась в следующей последовательности. На виртуальной машине был установлен и настроен веб-сервер nginx с тремя поддоменами, каждый из которых обслуживал трафик по одной версии протокола: HTTP/1.1, HTTP/2 и HTTP/3. Для оценки скорости установления соединения на каждом поддомене измерялась метрика TTFB. Затем на сервер был загружен шаблон веб-страницы, состоящий из 32 файлов, и произведены замеры времени её полной загрузки. Для эксперимента по сравнению алгоритмов контроля перегрузок на сервер был размещён файл объёмом 3 ГБ; скачивание выполнялось последовательно при активном алгоритме CUBIC (конфигурация по умолчанию) и после переключения на BBR, в обоих случаях с записью профиля скорости передачи. Последующий анализ результатов исследования включал сравнение полученных данных с теоретическими значениями, полученными на основе приведённой ниже модели.

При использовании HTTP/1.1 с шифрованием TLS 1.2 полное время до отправки первого байта данных складывается из нескольких этапов инициализации соединения (таблица 1), каждый из которых занимает некоторое количество круговых запросов между узлами сети. Таким образом, ожидаемое время инициализации защищённого соединения через HTTP/1.1 получается по формуле (1).

Таблица 1

Этапы инициализации HTTP/1.1 соединения

Этап	Назначение	Вклад в время инициализации
1	Three-way Handshake ¹³ протокола TCP	1,5 RTT
2	Обмен сообщениями ClientHello/ServerHello с согласованием шифров для протокола TLS [16]	1 RTT
3	Передача сертификата и сессионных ключей для протокола TLS [16]	1 RTT

$$T_{TCP} = 1,5 RTT + 1 RTT + 1 RTT = 3,5 RTT. \quad (1)$$

Протокол QUIC объединяет транспортное и криптографическое рукопожатие на основе TLS 1.3, сокращая суммарные затраты до 1 RTT. По формуле (2) получаем теоретический выигрыш по метрике TTFB.

$$\mu = \frac{T_{TCP} - T_{QUIC}}{T_{TCP}} \cdot 100 \% = \frac{3,5 RTT - 1 RTT}{3,5 RTT} \approx 71,43 \%. \quad (2)$$

Инструменты. Все измерения производили с использованием мобильного интернета 4G. На виртуальной машине был установлен nginx версии 1.27.4, поскольку поддержка HTTP/3 соединений в веб-сервер nginx добавлена с версии 1.25.0¹⁴.

Для проведения экспериментов было сконфигурировано три поддомена: http1.serafimdev.com, http2.serafimdev.com, http3.serafimdev.com с сертификатами от Let's Encrypt для протоколов HTTP/1.1, HTTP/2, HTTP/3 соответственно. При конфигурации использовалось программное обеспечение (ПО) автоматизации работы с сертификатами certbot.

Корректность выполненных на сервере настроек была проверена через инструменты разработчика браузера Google Chrome. При обмене данных по протоколу HTTP/3 в столбце протокол должны отображаться значения h3 (рис. 1). Данные инструменты также предоставляют информацию для оценки производительности веб-страницы и скоростей загрузки её компонентов и применялись для измерения времени полной загрузки страницы.

¹⁰ Виртуальное ядро процессора (англ.).

¹¹ Долгосрочная поддержка (англ.).

¹² Адрес интернет-протокола (англ.).

¹³ Трёхстороннее рукопожатие (англ.).

¹⁴ Mukhin S. Как включить HTTP/3 в NGINX. SergeyMukhin.com. URL: <https://sergeymukhin.com/blog/kak-vklyuchit-http3-v-nginx> (дата обращения: 11.10.2025).

Ресурс	Статус	Протокол
 images.html	200	h3
 1.jpg	200	h3
 2.jpg	200	h3
 3.jpg	200	h3
 4.jpg	200	h3
 5.jpg	200	h3

Рис. 1. Протокол HTTP/3 в инструментах разработчика

Метрика TTFB была определена при помощи скрипта на языке программирования JavaScript, который выполнял запросы к серверу с записью значений метрики. Аналогичный подход был применён и для сбора данных, необходимых для визуализации графиков скоростей загрузки ресурсов при различных алгоритмах контроля перегрузки. При проведении измерений было важно отключить кэширование запросов в инструментах разработчика, чтобы исключить влияние локального кэша на результаты. Обработка собранных данных была выполнена с использованием библиотек `pumpru` и `matplotlib` для Python. Для интеграции нестандартного алгоритма управления перегрузкой BBR в ядро операционной системы — в данном исследовании использовалась Ubuntu 22.04.5 LTS — было необходимо загрузить соответствующий модуль — `sudo modprobe tcp_bbr` и изменить значение переменной — `sudo sysctl -w net.ipv4.tcp_congestion_control=bbr`.

Процедуры. Интернет-соединение часто бывает нестабильно, поэтому был применен статистический анализ результатов во избежание влияния выбросов на результаты исследования. В экспериментах по измерению времени инициализации соединения и скорости загрузки страницы замеры для каждого из протоколов были проведены по 20 раз. Затем был применен метод межквартильного размаха для выявления выбросов и среднее арифметическое для агрегации оставшихся измерений в финальный результат. Верхняя и нижняя границы выбросов определялись по формулам (3) и (4).

$$\text{Нижняя граница} = Q1 - 1,5 \times IQR; \quad (3)$$

$$\text{Верхняя граница} = Q3 + 1,5 \times IQR, \quad (4)$$

где $Q1$ — 25-й перцентиль, $Q3$ — 75-й перцентиль, $IQR = Q3 - Q1$.

При анализе скорости передачи для замеров использовался модуль JavaScript XMLHttpRequest, предоставляющий количество загруженных байт раз в неравное количество миллисекунд. Из-за маленького промежутка между измерениями и нестабильности сети необработанные данные содержали большое количество выбросов. Для получения более наглядных графиков было применено сглаживание результатов при помощи скользящей средней с окном в 31 измерение.

Результаты исследования

Процедура конфигурирования HTTP/3-сервера. Документация процесса конфигурации тестового стенда является самостоятельной задачей исследования. Для организации доступа к веб-контенту по HTTP/3 необходим сервер с операционной системой и доступ к нему. В исследовании применялся сервер с операционной системой Ubuntu 22.04.5 LTS с удаленным доступом по SSH. В процессе настройки была выявлена ключевая сложность при организации HTTP/3 веб-сервера — необходимость использовать `mainline`¹⁵ — репозиторий `nginx` вместо стандартного.

Стандартный репозиторий Ubuntu 22.04.5 LTS содержит `nginx` версии 1.18, тогда как поддержка HTTP/3 появилась начиная с версии 1.25.0. Для установки актуальной версии необходимо добавить основной репозиторий с последними версиями программного обеспечения в систему, используя команды, показанные на рис. 2.

```
curl https://nginx.org/keys/nginx_signing.key | gpg --dearmor \
| sudo tee /usr/share/keyrings/nginx-archive-keyring.gpg >/dev/null

echo "deb [signed-by=/usr/share/keyrings/nginx-archive-keyring.gpg] \
http://nginx.org/packages/mainline/ubuntu `lsb_release -cs` nginx" \
| sudo tee /etc/apt/sources.list.d/nginx.list
```

Рис. 2. Добавление mainline-репозитория nginx в список источников пакетов операционной системы

¹⁵ Основной репозиторий (англ.).

Приведённые инструкции записывают ключ, которым подписываются пакеты из нового репозитория в системное хранилище доверенных ключей: /usr/share/keyrings и добавляют адрес mainline-репозитория nginx в список источников пакетов /etc/apt/sources.list.d с указанием пути к подписи пакетов. После выполнения этих шагов при установке nginx через менеджер пакетов apt будет использоваться репозиторий, содержащий все последние версии сервиса.

Поскольку шифрование встроено в протокол QUIC, наличие сертификата и доменного имени является обязательным требованием при использовании HTTP/3. На предыдущих версиях протокола HTTP шифрование через TLS является отдельным слоем, что добавляет задержку при установлении соединения, но позволяет организовать незащищённый обмен данными. HTTP без шифрования имеет применение в сетях, где шифрование уже обеспечено через Virtual Private Network¹⁶ (VPN). Таким образом, использование HTTP/3 в таких сетях приводит к необходимости настройки сертификатов. На используемом тестовом стенде были выпущены сертификаты от компании Let's Encrypt при помощи программного обеспечения по автоматизации работы с сертификатами — certbot.

Помимо этого, в конфигурационный файл сайта необходимо внести изменения (рис. 3). Ключевыми в конфигурации являются строки 3, 10, 12, 16. Строка 12 содержит инструкцию, разрешающую использование протокола HTTP/3. На строчке 16 указывается заголовок Alt-Svc. При наличии этого заголовка в ответе сервера клиент переходит на использование HTTP/3 (если поддерживается). Поэтому на строчках 2 и 3 указана поддержка стандартных TCP соединений и UDP через QUIC. Впоследствии значение заголовка кэшируется и обмен заголовками по TCP не происходит. Так как QUIC работает на основе TLS версии 1.3, эта версия явно указывается на строке 10.

```

1  server {
2      listen 443 ssl;
3      listen 443 quic reuseport;
4
5      server_name http3.serafimdev.com;
6
7      ssl_certificate /etc/letsencrypt/live/http3.serafimdev.com/fullchain.pem;
8      ssl_certificate_key /etc/letsencrypt/live/http3.serafimdev.com/privkey.pem;
9
10     ssl_protocols TLSv1.3;
11
12     http3 on;
13     quic_gso on;
14     quic_retry on;
15
16     add_header Alt-Svc 'h3=":443"; ma=86400';
17
18     root /var/www/http3.serafimdev.com;
19     index index.html;
20
21     location / {
22         try_files $uri $uri/ =404;
23     }
24 }
25
26 server {
27     listen 80;
28     server_name http3.serafimdev.com;
29     return 301 https://$host$request_uri;
30 }

```

Рис. 3. Конфигурационный файл nginx для HTTP/3 сайта

TTFB. TTFB — метрика, показывающая, сколько времени проходит от отправки запроса до начала передачи ресурса. Для проведения замеров использовался JavaScript скрипт. Полученные значения приведены в таблице 2.

Таблица 2

Замеры TTFB по разным версиям протокола HTTP

HTTP/1,1, мс	HTTP/2, мс	HTTP/3, мс
20,5	80,3	69,4
28,3	28,4	26,8
38,3	24,4	29,1
27,0	30,4	16,3
41,3	29,1	21,1
24,8	25,7	17,5

¹⁶ Виртуальная частная сеть (англ.).

HTTP/1,1, мс	HTTP/2, мс	HTTP/3, мс
30,1	33,2	18,4
26,7	30,4	23,2
30,9	46,0	22,4
24,7	15,6	23,1
29,2	26,0	24,1
33,3	23,6	19,7
25,9	26,1	24,3
34,7	26,3	23,7
33,3	25,1	23,5
33,3	32,5	25,4
25,9	26,5	26,4
29,4	19,7	22,6
25,0	26,9	24,5
22,7	28,6	15,8

Были получены значения:

- HTTP/1.1: выбросов нет, среднее — 29,27 мс;
- HTTP/2: есть 3 выброса, среднее — 27,23 мс;
- HTTP/3: есть 1 выброс, среднее — 22,52 мс.

$$\mu = \frac{29,27 - 22,52}{29,27} \cdot 100 \% \approx 23,06 \%. \quad (5)$$

Согласно расчётам по формуле (5), эксперимент показал, что переход с HTTP/1.1 на HTTP/3 с протоколом QUIC обеспечил сокращение времени до первого байта (TTFB) на 23,06 %.

Время загрузки страницы. Для исследования влияния использования протокола QUIC на скорость загрузки обычной веб страницы в открытых источниках был найден шаблон страницы, состоящий из 32 файлов, всего 728 килобайт. Полученные значения представлены в таблице 3.

Таблица 3

Время загрузки страницы по разным версиям протокола HTTP

HTTP/1,1, мс	HTTP/2, мс	HTTP/3, мс
1140	1060	781
826	794	699
1160	953	630
726	851	733
731	940	622
757	814	652
746	846	669
815	908	978
791	747	764
834	832	917
709	1060	725
830	868	601
951	744	890
1020	785	643
677	725	906
806	711	739
1100	695	647
743	922	687
775	750	652
739	658	678

Были получены значения:

- HTTP/1.1: есть 3 выброса, среднее — 792,7 мс;
- HTTP/2: выбросов нет, среднее — 833,15 мс;
- HTTP/3: есть 1 выброс, среднее — 717,63 мс.

$$\mu = \frac{792,7 - 717,63}{792,7} \cdot 100 \% \approx 9,5 \%. \quad (6)$$

Расчёт по формуле (6) показывает сокращение времени загрузки страницы на 9,5 % при использовании HTTP/3 и QUIC по сравнению с HTTP/1.1 и TCP.

Скорость передачи. Для эксперимента использовался файл размером 3 Гб передача которого осуществлялась по протоколу HTTP/2. Выбор HTTP/2 для данного эксперимента обусловлен методологическими соображениями: целью сравнения являлась изолированная оценка влияния алгоритма контроля перегрузок на профиль скорости передачи, без наложения дополнительных эффектов, вносимых транспортным протоколом. Протокол HTTP/3 (QUIC) реализует собственный стек управления перегрузками поверх UDP, тогда как HTTP/2 использует стандартный TCP-стек ядра операционной системы, в котором переключение между алгоритмами CUBIC и BBR осуществляется непосредственно через параметры ядра Linux без изменения логики протокола прикладного уровня. Было произведено скачивание файла с применением алгоритмов контроля перегрузок CUBIC и BBR. На рис. 4 и 5 приведены начальные фрагменты из графиков скорости загрузки при использовании CUBIC и BBR соответственно.

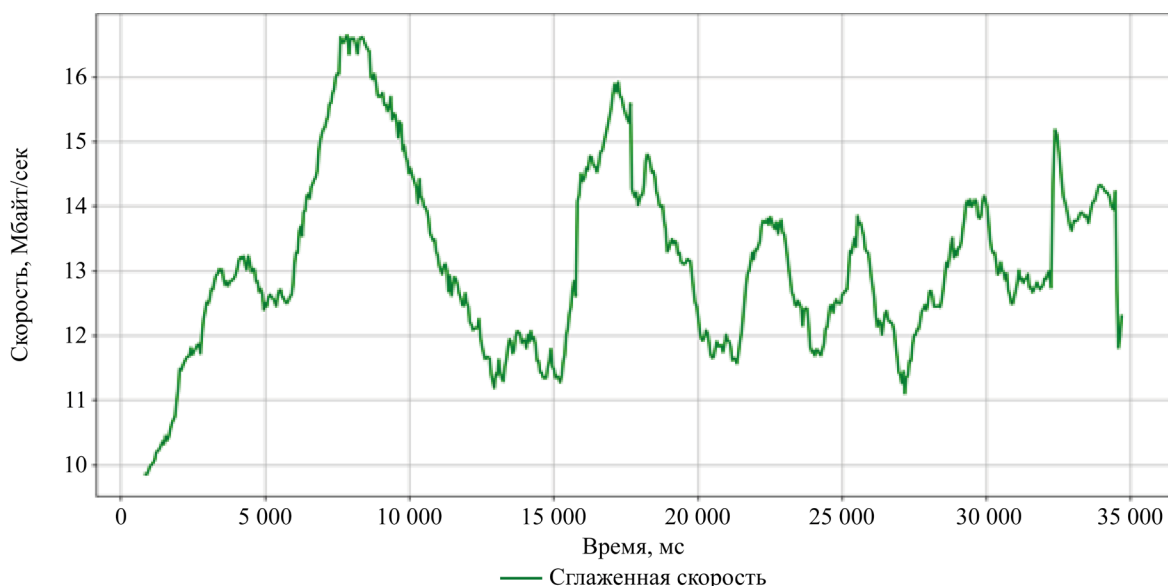


Рис. 4. Скорость скачивания при использовании CUBIC

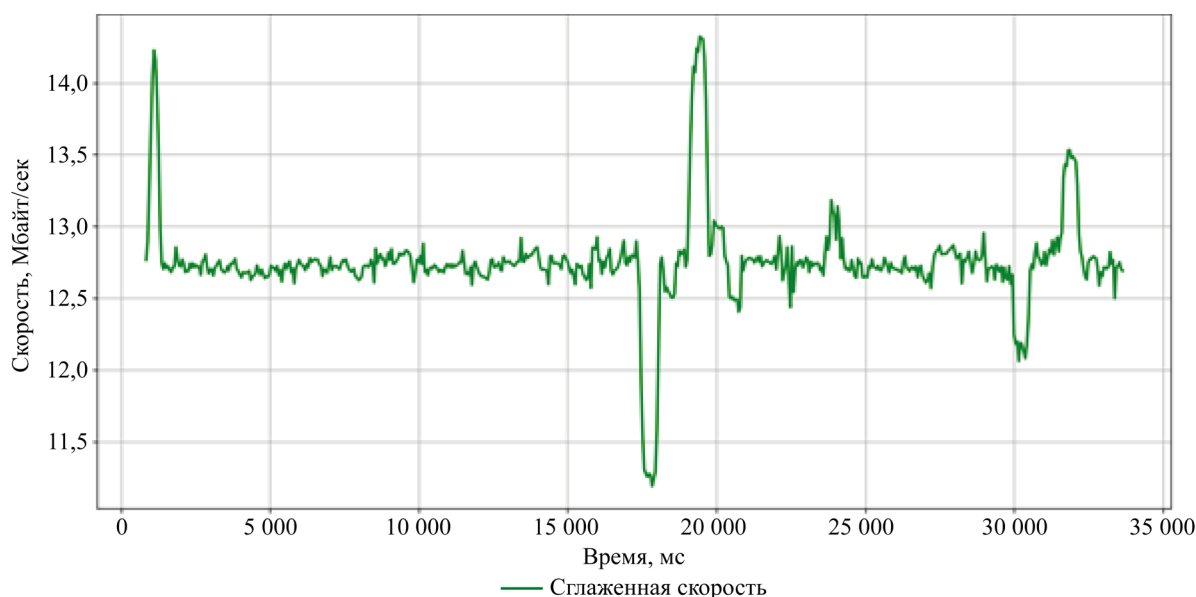


Рис. 5. Скорость скачивания при использовании BBR

При работе алгоритма BBR скорость большую часть времени держится на одном уровне. Средние значения скорости скачивания получились 13,12 МБ/с для CUBIC и 12,75 МБ/с для BBR.

Площадь под кривой графика отражает количество переданной информации и вычисляется по формуле (7). За первые 3 секунды передачи с использованием CUBIC было передано 23,1568 МБ данных, а с использованием BBR — 27,3622 МБ.

$$S = \int_0^{3000} v(t) dt \approx \sum_{i=0}^{3000} \frac{v_i + v_{i+1}}{2} (t_{i+1} - t_i). \quad (7)$$

Обсуждение. Полученные результаты показывают, что практический выигрыш от внедрения HTTP/3 (QUIC) подтверждается экспериментально, однако его величина оказывается существенно ниже теоретически ожидаемой, что требует интерпретации с учётом особенностей сетевой среды, программной реализации и методики измерений.

Сложности конфигурирования и операционные риски. Описанная процедура настройки сопряжена с рядом практических трудностей, которые необходимо учитывать при планировании производственного развёртывания HTTP/3.

Первая сложность связана с невозможностью использования стандартных образов виртуальных машин и контейнеров при автоматизации развёртывания. Большинство официальных образов Ubuntu и популярных контейнерных образов nginx на базе Alpine Linux или Debian включают версию nginx из дистрибутивного репозитория, которая не поддерживает HTTP/3. При использовании таких образов в системах оркестрации (Kubernetes, Docker Compose) каждый узел требует либо ручного добавления mainline-репозитория и переустановки nginx, либо сборки собственного образа с нужной версией. Это усложняет автоматизацию и увеличивает операционные издержки на сопровождение инфраструктуры [17].

Вторая сложность обусловлена механизмом согласования протокола через заголовок Alt-Svc. Первое соединение клиента с сервером всегда устанавливается по TCP (HTTP/1.1 или HTTP/2), и лишь получив в ответе заголовки Alt-Svc, браузер переключается на QUIC при последующих запросах. Если между клиентом и сервером присутствует промежуточный узел — балансировщик нагрузки, обратный прокси или CDN-узел — который не поддерживает QUIC или блокирует UDP-трафик на порту 443, заголовок Alt-Svc либо не доходит до клиента, либо клиент не может установить UDP-соединение. Такое поведение затрудняет диагностику и создаёт риск скрытого отсутствия эффекта от внедрения протокола. Согласно базовой спецификации QUIC [18], при недоступности UDP-транспорта клиент должен откатиться к TCP-соединению без явной сигнализации об ошибке, что делает отсутствие HTTP/3 в реальном трафике трудно диагностируемым без явной инструментальной проверки. Для верификации факта использования HTTP/3 необходима явная проверка через инструменты разработчика браузера или серверные логи с фиксацией версии протокола.

Необходимым условием установления соединения по протоколу HTTP/3 на стороне клиента является наличие браузера с включённой поддержкой QUIC. В браузере Google Chrome поддержка протокола активирована по умолчанию начиная с версии 87, однако в корпоративных средах она может быть отключена групповыми политиками. При отключённой поддержке QUIC браузер не будет устанавливать соединения по HTTP/3 вне зависимости от корректности серверной конфигурации и наличия заголовка Alt-Svc.

Интерпретация результатов TTFB. Объединённое рукопожатие QUIC и TLS 1.3, регламентированное в RFC 9001 [19] и опирающееся на спецификацию TLS 1.3 [20], теоретически сокращает инициализацию до 1 RTT, однако практическая эффективность этого механизма чувствительна к особенностям реализации криптографического стека на конкретном оборудовании и в конкретной версии программного обеспечения. Снижение TTFB на 23,06 % при переходе с HTTP/1.1 на HTTP/3 экспериментально подтверждает преимущество объединённого QUIC-рукопожатия, устраняющего дополнительный RTT, характерный для раздельной инициализации TCP и TLS. Вместе с тем измеренный прирост существенно уступает теоретически ожидаемому значению 71,43 %, что объясняется совокупностью факторов.

Обработка UDP-пакетов на промежуточном сетевом оборудовании провайдера зачастую менее оптимизирована по сравнению с TCP-трафиком: маршрутизаторы и межсетевые экраны традиционно настроены под TCP-профиль нагрузки, а часть операторов применяет дополнительную инспекцию или троттлинг UDP-потоков [8, 11]. Реализации QUIC в nginx и Google Chrome на момент проведения эксперимента сохраняют экспериментальный статус, что проявляется в неоптимальных значениях параметров стека и дополнительных издержках на уровне приложения. Мобильный канал 4G вносит нестабильную переменную составляющую задержки, которая нивелирует часть выигрыша от сокращения числа RTT при установлении соединения. Аналогичное расхождение между теорией и практикой зафиксировано в масштабном развёртывании QUIC компанией Google, где реальное снижение задержки поиска составило 8,0 % для настольных и 3,6 % для мобильных пользователей вместо теоретически ожидаемых значений [8], а также подтверждено в отечественных исследованиях реальных сетевых условий [11]. Полученный результат согласуется и с данными компании ITSumma¹⁷, зафиксировавшей снижение TTFB на 12,4 %, что в совокупности свидетельствует о системном характере разрыва между теоретической и практической эффективностью протокола на текущем этапе его развития.

¹⁷ ITSumma. HTTP/3 и QUIC: будущее быстрого интернета. Хабр. URL: <https://habr.com/ru/companies/itsumma/articles/497520> (дата обращения: 11.02.2026)

Интерпретация времени загрузки страницы. Ускорение полной загрузки тестовой страницы на 9,5 % при использовании HTTP/3 по сравнению с HTTP/1.1 объясняется прежде всего мультиплексированием запросов в рамках одного QUIC-соединения и устранением Head-of-Line¹⁸ блокировки на транспортном уровне. При загрузке тестовой страницы, состоящей из 32 файлов, HTTP/3 параллельно обрабатывает все запросы без взаимной блокировки потоков, тогда как HTTP/1.1 ограничен числом параллельных TCP-соединений.

Примечательна аномалия в результатах HTTP/2: его среднее время загрузки (833,15 мс) превысило аналогичный показатель HTTP/1.1 (792,7 мс). Данный эффект является артефактом применённой методологии статистической обработки, а не показателем реальной неэффективности протокола. В выборке HTTP/1.1 метод межквартильного размаха исключил три значения, превышающих верхнюю границу, что существенно снизило итоговое среднее. Средние значения времени загрузки для HTTP/1.1 и HTTP/2 по всем двадцати измерениям без удаления выбросов составляют 843,8 мс и 833,15 мс соответственно, что подтверждает большую эффективность HTTP/2 в скорости загрузки по сравнению с HTTP/1.1. В данных HTTP/2 выбросы по верхней границе отсутствовали, это означает, что протокол обеспечил более равномерное распределение времён отклика за счёт мультиплексирования запросов и приоритизации потоков в рамках одного TCP-соединения. Таким образом, сравнение очищенных средних некорректно отражает реальное соотношение протоколов: HTTP/2 демонстрирует более высокую стабильность и меньшую вариативность, тогда как итоговое среднее значение после удаления выбросов оказалось статистически ниже у HTTP/1.1.

Интерпретация CUBIC и BBR. Средние скорости передачи файла объёмом 3 ГБ при использовании CUBIC (13,12 МБ/с) и BBR (12,75 МБ/с) оказались сопоставимы: разница в 3 % находится в пределах погрешности измерений при нестабильном мобильном канале. Данный результат предсказуем для длительной передачи одного файла: на протяжении сотен секунд оба алгоритма достигают стационарного режима загрузки канала, и различия в стратегии инициализации перестают определять итоговую среднюю скорость.

Принципиальное различие между алгоритмами проявляется в профиле передачи. CUBIC демонстрирует характерный пилообразный профиль скорости (рис. 4): она монотонно возрастает от стартовых значений менее 10 МБ/с до рабочего уровня примерно за четыре секунды, после чего циклически снижается в моменты потерь пакетов. BBR, использующий оценку пропускной способности Bandwidth-Delay Product¹⁹ (BDP), с первых секунд удерживает скорость на стабильном уровне.

При работе с WWW большая часть запрашиваемых файлов имеют небольшой размер. Предположим, что размер одного файла с ресурсами для веб-страницы составляет 63 КБ, начальный размер окна для медленного старта — 10 килобайт (настройка по умолчанию в Linux-системах), максимальный размер окна — 64 КБ. При медленном старте в алгоритме CUBIC для увеличения окна перегрузки используется экспонента [21].

$$A = \left\lceil \log_2 \frac{cwnd_{max}}{cwnd_0} \right\rceil = \left\lceil \log_2 \frac{64}{10} \right\rceil = 3. \quad (8)$$

Результат вычислений по формуле (8) свидетельствует о том, что для передачи файла необходимо увеличить размер окна перегрузки в три раза, на что потребуется 3 RTT. Если бы передача с самого начала велась с окном перегрузки равным 64, то можно было бы провести передачу за 1 RTT.

Именно этот стартовый период определяет прикладную значимость BBR для веб-сервисов. Типичный ресурс страницы — скрипт, таблица стилей, шрифт — передаётся за одно-два RTT и укладывается в фазу медленного старта CUBIC, в течение которой алгоритм ещё не вышел на рабочее окно перегрузки. BBR в этом сценарии обеспечивает немедленный выход на оптимальную скорость, что суммарно сокращает время параллельной загрузки множества мелких ресурсов. Таким образом, преимущество BBR перед CUBIC реализуется не на длинных передачах, а при загрузке ресурсов страницы, сопровождающейся множественными короткими запросами сразу после установления соединения.

Количественным подтверждением преимущества BBR в начальной фазе передачи служит сравнение площадей под кривыми графиков скорости, вычисленных по формуле (7) методом трапеций. За первые три секунды передачи алгоритм CUBIC обеспечил передачу 23,16 МБ данных, тогда как BBR — 27,36 МБ, что на 18,2 % больше. Данный результат согласуется с результатами экспериментов в работе [15], где была зафиксирована более высокая пропускная способность BBR при высоких значениях RTT, характерных в том числе для межконтинентальных соединений.

Перспективным направлением дальнейших исследований является экспериментальная проверка преимущества BBR непосредственно в веб-сценарии: измерение времени полной загрузки страницы, состоящей из множества небольших ресурсов, при использовании алгоритмов CUBIC и BBR в сочетании с каждой из версий протокола HTTP. Подобный эксперимент позволит количественно оценить, в какой мере ускоренный выход BBR на рабочую скорость транслируется в сокращение времени загрузки страницы в реальном веб-сценарии, а также определить, при каких сочетаниях версии протокола и алгоритма управления перегрузками прикладной эффект проявляется наиболее отчетливо.

¹⁸ Блокировка начала очереди (англ.).

¹⁹ Произведение пропускной способности сети на время задержки (англ.).

Ограничения исследования и практические рекомендации. Полученные результаты обладают практической ценностью для оценки выгоды перехода на HTTP/3, однако ряд ограничений экспериментального стенда сужает область применимости выводов.

Конфигурация стенда включала единственную виртуальную машину с 1 vCPU и единственного клиента, подключённого через мобильный канал 4G. Мобильный канал привносит нестабильную переменную задержку и непостоянную полосу пропускания, что увеличивает разброс замеров и затрудняет изоляцию эффекта протокола от эффекта канала. Производительность транспортных протоколов существенно зависит от сетевой топологии, значений RTT, уровня потерь пакетов и политик промежуточного оборудования, поэтому перенос результатов на иные сетевые сценарии требует самостоятельной экспериментальной проверки [22, 23].

Малый объём выборки (20 замеров на протокол) обеспечивает достаточную точность для первичной оценки, но недостаточен для построения статистически устойчивых доверительных интервалов. Отсутствие географически распределённых клиентов не позволяет оценить влияние высоких RTT — условий, при которых преимущество QUIC теоретически наиболее выражено.

На основании полученных данных можно сформулировать следующие практические рекомендации. Переход на HTTP/3 целесообразен прежде всего для сервисов с большим числом небольших ресурсов на странице, аудиторией с высокими задержками (мобильные пользователи, удалённые регионы) и требованиями к минимизации TTFB. Для сервисов с преобладающей передачей крупных файлов через стабильные проводные соединения практический выигрыш от перехода будет минимальным. Алгоритм BBR рекомендуется к включению совместно с HTTP/3: при сопоставимой средней скорости на длинных передачах он обеспечивает стабильный профиль и сокращает время выхода на рабочую скорость, что критично именно в веб-сценарии множественных коротких запросов.

Для повышения достоверности оценки производительности в последующих исследованиях обосновано сравнение альтернативных серверных реализаций QUIC (помимо nginx), а также анализ влияния прокси-серверов на метрики производительности при использовании QUIC [10].

Заключение. Проведённое исследование подтвердило, что внедрение протокола QUIC даёт практический выигрыш в скорости веб-взаимодействий. Таким образом, было получено сокращение метрики TTFB на 23,06 % и снижение времени загрузки тестовой веб-страницы на 9,5 %. Приведённая в статье модель прогнозирует снижение TTFB не на 23,06 %, а на 71,43 %. Такое расхождение показывает сильное влияние экспериментального статуса протокола на его распространённость на сетевом оборудовании и критичность иных факторов, рассмотренных в статье, для полномасштабного внедрения протокола.

В работе проведено сравнение алгоритмов контроля перегрузок CUBIC (стандартный) и BBR (наиболее распространённый в HTTP/3), а также представлено теоретическое обоснование преимущества BBR при передаче мало-размерных файлов, обусловленного более высокой начальной скоростью и отсутствием механизма медленного старта. Аналогичные достоинства BBR зафиксированы и в зарубежных исследованиях. В частности, в статье [15] отмечается эффективность данного алгоритма в каналах с большим RTT (межконтинентальные соединения) благодаря моделированию пропускной способности канала связи.

Протокол QUIC в сочетании с HTTP/3 на текущем этапе развития сопряжён с определёнными сложностями внедрения, однако обладает значительным потенциалом для построения высокопроизводительных информационных систем, что подтверждается результатами других исследований [24], в том числе применительно к системам с требованиями к надёжной доставке трафика реального времени [25].

Процесс настройки HTTP/3-сервера требует загрузки дополнительных модулей в ядро операционной системы. Это негативно сказывается на распространённости протокола, однако при наличии в организации достаточного количества квалифицированных специалистов для поддержки подобной инфраструктуры переход на HTTP/3 способен обеспечить ощутимые преимущества. Выигрыш по TTFB и отсутствие Head-of-Line блокировок теоретически позволят существенно ускорить загрузку ресурсов страницы. Особенно заметным прирост производительности может оказаться на страницах, содержащих множество небольших ресурсов.

Помимо вопросов производительности, переход на QUIC и HTTP/3 порождает специфические вызовы для инфраструктуры информационной безопасности: встроенное шифрование затрудняет глубокую инспекцию пакетов (DPI) и функционирование систем обнаружения вторжений, что детально рассмотрено в [26] и требует отдельного учёта при планировании производственного внедрения.

Ограничения исследования включают упрощённую конфигурацию тестового стенда (одна виртуальная машина и мобильный 4G-канал), а также небольшую выборку замеров. Полученные результаты обладают прикладной ценностью для оценки целесообразности перехода на HTTP/3, однако нуждаются в подтверждении на более широком спектре сценариев, поскольку производительность транспортных протоколов существенно зависит от сетевой топологии, задержек и потерь. Тема алгоритмов контроля перегрузок может быть раскрыта подробнее в последующих публикациях — в частности, представляется целесообразным сопоставление различных алгоритмов в сочетании с разными версиями протокола HTTP. В современных информационных системах широко применяются прокси-серверы для управления трафиком; перспективным направлением является исследование влияния подобных промежуточных узлов на метрики производительности при использовании QUIC [10].

Список литературы / References

1. Горохова П.А. Российский рынок рекламы: состояние, структура, тенденции и перспективы развития. *Государственное и муниципальное управление. Ученые записки*. 2020;(1):297–302. <https://doi.org/10.22394/2079-1690-2020-1-1-297-302>
- Gorokhova PA. Russian Advertising Market: State, Structure, Trends and Development Prospects. *State and Municipal Management. Scholar Notes*. 2020;(1):297–302. <https://doi.org/10.22394/2079-1690-2020-1-1-297-302>
2. Arapakis I, Park S, Pielot M. Impact of Response Latency on User Behaviour in Mobile Web Search. In: *Proc. ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR '21)*. Canberra, Australia: ACM; 2021. P. 279–283. <https://doi.org/10.1145/3406522.3446038>
3. Рахмани Д., Баранов М.Д., Кузьмин Д.А. Разработка метода оптимизации веб-приложений с целью повышения производительности. *International Journal of Humanities and Natural Sciences*. 2025;102(3–1):237–241. <https://doi.org/10.24412/2500-1000-2025-3-1-237-241>
- Rahmani J, Baranov MD, Kuzmin DA. Development of a Web Application Optimization Method to Improve Performance. *International Journal of Humanities and Natural Sciences*. 2025;102(3–1):237–241. <https://doi.org/10.24412/2500-1000-2025-3-1-237-241>
4. Wendroth J, Jaeger B. A Brief Overview on HTTP. In: *Proc. Seminar on Innovative Internet Technologies and Mobile Communications (IITM)*. Munich: Technical University of Munich; 2022. P. 59–69. https://doi.org/10.2313/NET-2022-11-1_11
5. Рахмани Д., Рогов И.Д. Тенденции развития сетевых технологий в 2022 году. В: *Труды XVI Международной отраслевой научно-технической конференции «Технологии информационного общества», Москва, 02–03 марта 2022 года*. Москва; Издательский дом Медиа паблишер. 2022. С. 30–31.
- Rahmani J, Rogov ID. Trends in the Development of Network Technologies in 2022. In: *Proc. XVI International Industry Scientific and Technical Conference*. Moscow: Izdatel'skii dom Media publisher; 2022. P. 30–31. (In Russ.)
6. Bishop M (ed). *HTTP/3. RFC 9114*. AMS LLC: RFC Editor; 2022. <https://doi.org/10.17487/RFC9114>
7. Amet M, Thomas L, Ye-Qiong Song. A Performance Evaluation of QUIC in Real-Time Networks. In: *Proc. 32nd International Conference on Real-Time Networks and Systems (RTNS 2024)*. New York, NY: ACM; 2024. P. 255–265. <https://doi.org/10.1145/3696355.3699698>
8. Langley A, Riddoch A, Wilk A, Vicente A, Krasic C, Zhang D, et al. The QUIC Transport Protocol: Design and Internet-Scale Deployment. In: *Proc. ACM SIGCOMM Conference*. New York, NY: ACM; 2017. P. 183–196. <https://doi.org/10.1145/3098822.3098842>
9. Fan Liu, Crowley P. Security and Performance Characteristics of QUIC and HTTP/3. In: *Proc. 10th ACM Conference on Information-Centric*. New York, NY: ACM; 2023. p. 124–126. <https://doi.org/10.1145/3623565.3623757>
10. Fan Liu, Farkiani B, Dehart J, Parwatikar J, Crowley P. Performance Comparison of HTTP/3 and HTTP/2 with Proxy Integration. *arXiv preprint*. 2024. <https://doi.org/10.48550/arXiv.2409.16267>
11. Андреянов Н.С., Рябиков А.Ю. Исследование клиент-серверных протоколов для повышения отказоустойчивости и безопасности соединения. *International Journal of Humanities and Natural Sciences*. 2024;97(10–1):128–132. <https://doi.org/10.24412/2500-1000-2024-10-1-128-132>
- Andreyanov NS, Ryabikov AY. Research on Client-Server Protocols to Improve Fault Tolerance and Connection Security. *International Journal of Humanities and Natural Sciences*. 2024;97(10–1):128–132. <https://doi.org/10.24412/2500-1000-2024-10-1-128-132>
12. Смирнов И.А., Михайлов М.Ю., Лаптева М.Г. HTTP/3 и протокол QUIC: будущее высокопроизводительных веб-сервисов. *Вестник науки*. 2025;2(6):1780–1788.
- Smirnov IA, Mikhailov MYu, Lapteva MG. HTTP/3 and the QUIC Protocol: Future of High-Performance Web Services. *Vestnik Nauki*. 2025;2(6):1780–1788.
13. Jia Zhang, Haixuan Tong, Enhuan Dong, Xin Qian, Mingwei Xu, Xiaotian Li, et al. Cold Start or Hot Start? Robust Slow Start in Congestion Control with A Priori Knowledge for Mobile Web Services. In: *Proc. ACM Web Conference*. New York, NY: ACM; 2024. P. 2870–2878. <https://doi.org/10.1145/3589334.3645393>
14. Cardwell N, Yuchung Cheng, Gunn CS, Hassas Yeganeh S, Jacobson V. BBR: Congestion-Based Congestion Control. *Communications of the ACM*. 2017;60(2):58–66. <https://doi.org/10.1145/3009824>
15. Yi Han, Mengjie Zuo, Huijun Yuan, Yi Zhong, Zhenhui Yuan, Ting Bi. A QoS-Based Fairness-Aware BBR Congestion Control Algorithm Using QUIC. *Mathematical Problems in Engineering*. 2022;2022:7222030. <https://doi.org/10.1155/2022/7222030>
16. Dierks T, Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.2. *RFC 5246*. New York: IETF; 2008. 101 p. <https://doi.org/10.17487/RFC5246>
17. Voronova AG. Typification of Projects for the Transition to Cloud Services. *Advanced Engineering Research (Rostov-on-Don)*. 2024;24(3):274–282. <https://doi.org/10.23947/2687-1653-2024-24-3-274-282>
18. Iyengar J, Thomson M (eds). *QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000*. AMS LLC: RFC Editor; 2021. <https://doi.org/10.17487/RFC9000>
19. Thomson M, Turner S (eds). *Using TLS to Secure QUIC. RFC 9001*. AMS LLC: RFC Editor; 2021. <https://doi.org/10.17487/RFC9001>

20. Rescorla E. *The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446*. New York, NY: IETF; 2018. <https://doi.org/10.17487/RFC8446>
21. Rhee I, Xu L, Ha S, Zimmermann A, Eggert L, Scheffenegger R. *CUBIC for Fast and Long-Distance Networks. RFC 8312*. AMS LLC: RFC Editor; 2018. <https://doi.org/10.17487/RFC8312>
22. Linets G.I., Voronkin R.A., Slyusarev G.V., Govorova S.V. Optimization Problem for Probabilistic Time Intervals of Quasi-Deterministic Output and Self-Similar Input Data Packet Flow in Telecommunication Networks. *Advanced Engineering Research (Rostov-on-Don)*. 2024;24(4):424-432. <https://doi.org/10.23947/2687-1653-2024-24-4-424-432>
23. Samoylenko V.V. Concept of a Multilevel Network Infrastructure for Monitoring Agricultural Facilities Based on Wireless Sensor Networks. *Advanced Engineering Research (Rostov-on-Don)*. 2025;25(4):371-382. <https://doi.org/10.23947/2687-1653-2025-25-4-2238>
24. Gupta A, Bartos R. Improving Web Content Delivery with HTTP/3 and Non-Incremental EPS. In: *Proc. Conference on Networking and Internet Architecture*. Durham, NH: University of New Hampshire; 2024. P. 1–10. <https://doi.org/10.48550/arXiv.2404.13460>
25. Антонова В.М., Бужин И.Г., Гречишкина Н.А., Корочкин Е.Д., Кузнецов Н.А. Транспортный протокол QUIC как способ надёжной передачи трафика реального времени. *Информационные процессы*. 2025;25(3):490–500. https://doi.org/10.53921/18195822_2025_25_3_490
- Antonova VM, Buzhin IG, Grechishkina NA, Korochkin ED, Kuznetsov NA. QUIC Transport Protocol as Reliable Transmission Method for Real-Time Traffic. *Information Processes*. 2025;25(3):490–500. https://doi.org/10.53921/18195822_2025_25_3_490
26. Селиванов М.А. Сетевые протоколы нового поколения и их анализ с точки зрения информационной безопасности. *Молодой исследователь Дона*. 2022;(1(34):56–62.
- Selivanov MA. New Generation Network Protocols and Their Analysis from the Point of View of Information Security. *Young Researcher of the Don*. 2022;(1):56–62.

Об авторах:

Джахед Рахмани, старший преподаватель кафедры «Сетевые информационные технологии и сервисы» Московского технического университета связи и информатики (111024, Российская Федерация, г. Москва, ул. Авиамоторная, 8 а), [SPIN-код](#), [ORCID](#), j.rahmani@mtuci.ru

Серафим Павлович Сухарев, техник НИО «Центр искусственного интеллекта и перспективных проектов» Московского технического университета связи и информатики (111024, Российская Федерация, г. Москва, ул. Авиамоторная, 8 а), [SPIN-код](#), [ORCID](#), [ResearchGate](#), [ResearcherID](#), s.p.suharev@mtuci.ru

Заявленный вклад авторов:

Д. Рахмани: разработка концепции и методологии, научное руководство, написание рукописи, редактирование.

С.П. Сухарев: курирование данных, формальный анализ, проведение исследования, разработка программного обеспечения, визуализация, написание черновика рукописи.

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the Authors:

Jahed Rahmani, Senior Lecturer of the Department of Network Information Technologies and Services, Moscow Technical University of Communications and Informatics (8a, Aviamotornaya Str., Moscow, 111024, Russian Federation), [SPIN-code](#), [ORCID](#), j.rahmani@mtuci.ru

Serafim P. Sukharev, Technician of the Research and Innovation Department “Center for Artificial Intelligence and Advanced Projects”, Moscow Technical University of Communications and Informatics (8a, Aviamotornaya Str., Moscow, 111024, Russian Federation), [SPIN-code](#), [ORCID](#), [ResearchGate](#), [ResearcherID](#), s.p.suharev@mtuci.ru

Claimed Contributorship:

J Rahmani: conceptualization, methodology, supervision, writing & editing.

SP Sukharev: data curation, formal analysis, investigation, software, visualization, writing – original draft preparation.

Conflict of Interest Statement: the authors declare no conflict of interest.

All authors have read and approved the final version of manuscript.

Поступила в редакцию / Received 02.03.2026

Поступила после рецензирования / Reviewed 02.04.2026

Принята к публикации / Accepted 09.04.2026