ORES
Научно-практический электронный журнал
Оригинальные исследования (ОРИС), *ISSN 2222-0402*
www.ores.su
original-research.ru

УДК 004.773

СРАВНЕНИЕ ПРОТОКОЛОВ MQTT И COAP, ИСПОЛЬЗУЕМЫХ ДЛЯ ПЕРЕДАЧИ СЕНСОРНЫХ ПАКЕТОВ

Аскеров Салех Теймур оглы,
Студент группы ИУК5-11М
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
askerovst1@student.bmstu.ru

Фёдоров Виктор Олегович,
Доцент кафедры ИУК5 «Системы обработки информации»
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
fedorov_vo@bmstu.ru

Романовский Илья Олегович,
Студент группы ИУК5-11М
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
romanovskiyio@student.bmstu.ru

Аннотация

В данной статье проводится сравнительный анализ производительности протоколов MQTT и CoAP, применяемых в системах Интернета вещей. Исследование фокусируется на экспериментальном тестировании передачи данных в различных сетевых сценариях, включая условия с высокой задержкой и потерей пакетов. Оценка проводится по таким критериям, как время отклика, коэффициент доставки сообщений и влияние уровней качества обслуживания на скорость передачи.

Ключевые слова: MQTT, CoAP, QoS, IoT, задержка передачи, надёжность доставки.

COMPARISON OF MQTT AND COAP PROTOCOLS USED TO TRANSMIT SENSOR PACKETS

Saleh T. Askerov,
Student of group IUK5-11M
Bauman Moscow State Technical University (Kaluga Branch)
askerovst1@student.bmstu.ru

Victor O. Fedorov,
Associate Professor of the Department of IUK5 "Information Processing Systems"

Bauman Moscow State Technical University (Kaluga Branch)
fedorov_vo@bmstu.ru

Ilya O. Romanovskiy,
Student of group IUK5-11M
Bauman Moscow State Technical University (Kaluga Branch)

ABSTRACT

This article provides a comparative analysis of the performance of the MQTT and CoAP protocols used in Internet of Things systems. The study focuses on experimental testing of data transmission in various network scenarios, including conditions with high latency and packet loss. The evaluation is based on criteria such as response time, message delivery rate, and the impact of service quality levels on transmission speed.

Keywords: MQTT, CoAP, QoS, IoT, transmission delay, delivery reliability.

Введение

В условиях повсеместного внедрения сенсорных сетей и систем Интернета вещей (IoT), эффективная передача телеметрии становится критически важной составляющей инфраструктуры дистанционной диагностики и мониторинга. Использование таких сетей обещает существенное повышение оперативности обнаружения неисправностей и снижение эксплуатационных затрат, однако обеспечение стабильного обмена данными в промышленных масштабах сопряжено с рядом технических вызовов. В реальных условиях эксплуатации нестабильные каналы связи, характеризующиеся потерей пакетов, высокой задержкой и ограниченной пропускной способностью, приводят к непредсказуемому поведению системы. Для инженеров это выражается в рисках пропуска критических сигналов, ложных срабатываниях и неоправданном усложнении архитектуры сбора данных. Особое внимание в данном контексте уделяется выбору протокола прикладного уровня — в частности, MQTT или CoAP — способного обеспечить надежную доставку сообщений в гетерогенной сетевой среде [3]. Цель данного исследования заключается в сравнительном анализе производительности протоколов MQTT и CoAP в различных сценариях сетевого взаимодействия для определения оптимального стека технологий в системах сбора сенсорных данных.

Обзор протоколов передачи данных для систем Интернета вещей

Протокол MQTT

MQTT (Message Queuing Telemetry Transport) представляет собой легковесный протокол обмена сообщениями, работающий поверх стека TCP/IP и использующий архитектуру «издатель-подписчик» [1]. В отличие от традиционной модели «клиент-сервер», где отправитель и получатель связаны напрямую, в MQTT взаимодействие осуществляется через центральный узел — брокер. Такой подход позволяет декуплировать компоненты системы во времени и пространстве, упрощая масштабирование.

Основные преимущества использования MQTT включают:

Гарантированная доставка (QoS): Протокол предлагает три уровня качества обслуживания: QoS 0 («не более одного раза»), QoS 1 («хотя бы один раз») и QoS 2 («ровно один раз»). Это позволяет гибко балансировать между надежностью доставки и нагрузкой на сеть в зависимости от критичности данных.

Эффективность при нестабильном соединении: Благодаря механизмам постоянного поддержания сессии (Keep Alive) и Last Will and Testament (LWT), MQTT эффективно работает в сетях с периодическими разрывами связи, оперативно уведомляя систему о статусе устройств.

При использовании MQTT важно учитывать, что:

Использование TCP обеспечивает надежность, но добавляет накладные расходы на установление соединения, что может увеличивать задержку и энергопотребление на этапе инициализации [3].

Централизованная архитектура требует надежного и производительного брокера, который может стать единой точкой отказа без должного резервирования.

В прикладных IoT-реализациях MQTT часто используют на оконечных устройствах на базе SoC ESP8266; обмен с сервером строится через брокер (например, Eclipse Mosquitto), а интеграция в систему управления может выполняться средствами типа OpenHAB. При оценке производительности MQTT обычно рассматривают зависимость времени отправки/приема от длины сообщения и уровня подтверждения доставки (QoS 1–2), а также измеряют сетевые показатели в Wi-Fi (время отклика по ICMP, потери пакетов и MQTT-сообщений), включая нагрузочные режимы с ростом очереди сообщений у брокера [7].

Протокол CoAP

CoAP (Constrained Application Protocol) представляет собой специализированный протокол передачи данных, разработанный для устройств с ограниченными ресурсами и работающий поверх транспортного протокола UDP [2]. Архитектурно CoAP следует модели REST, аналогичной HTTP, используя методы GET, POST, PUT и DELETE. Это обеспечивает простую интеграцию с веб-сервисами и позволяет устройствам взаимодействовать напрямую без постоянного посредника.

Ключевая особенность CoAP заключается в его минималистичности и способности работать в средах с высокой потерей пакетов и низкой пропускной способностью. Протокол использует асинхронный обмен сообщениями и компактный бинарный заголовок, что существенно снижает объем передаваемого трафика.

Основные характеристики CoAP:

Поддержка надежности поверх UDP: Несмотря на использование ненадежного транспорта, CoAP реализует собственный механизм подтверждения доставки через типы сообщений Confirmable (CON) и Non-confirmable (NON). Это позволяет приложению самостоятельно определять необходимость подтверждения для каждого пакета.

Встроенный механизм наблюдения: Расширение Observe позволяет клиентам подписываться на изменения ресурсов, что эмулирует поведение push-уведомлений без необходимости постоянного опроса сервера.

В сравнении с MQTT, CoAP характеризуется минимальным размером заголовка (4 байта против 2 байт у MQTT), что снижает энергопотребление и нагрузку на сеть для ограниченных устройств. По отношению к AMQP (тяжелые очереди сообщений) и HTTP (большие REST-заголовки) CoAP демонстрирует оптимальный баланс между легковесностью и поддержкой REST-методов (GET/POST/PUT/DELETE), что делает его предпочтительным для IoT-сетей с низкой пропускной способностью [8].

Экспериментальное исследование

Для проведения экспериментального исследования и исключения влияния неконтролируемых внешних факторов, таких как загрузка публичных каналов связи, был развернут изолированный тестовый стенд на базе виртуальной среды Ubuntu. Архитектура стенда включала MQTT-брокер Eclipse Mosquitto [4], работающий как системный сервис, и специально разработанный CoAP-сервер на языке Python, функционирующий в

пользовательском пространстве через UDP-сокеты. В качестве генератора нагрузки использовался скрипт, обеспечивающий последовательную отправку серий по 50 сообщений с полезной нагрузкой 200 байт и фиксацию метрик производительности для каждого пакета, что позволило оценить чистую эффективность протоколов и их устойчивость к контролируемой деградации сети.

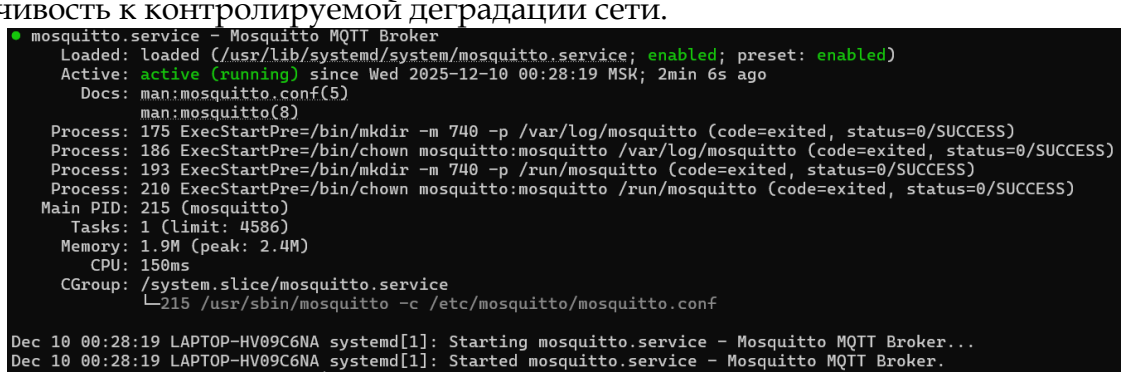

Рисунок 1 – Инициализация MQTT брокера

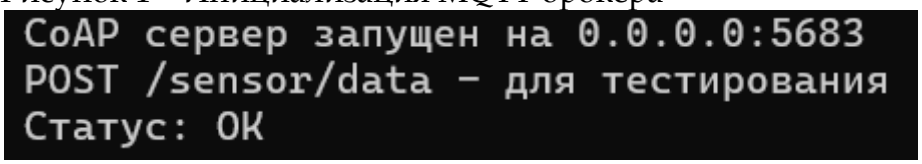

Рисунок 2 – Инициализация CoAP сервера

Моделирование сетевых условий

Ключевой особенностью эксперимента стало использование инструмента tc (Traffic Control) ядра Linux для эмуляции реальных проблем IoT-сетей. В отличие от программных задержек (time.sleep), tc работает на уровне сетевого интерфейса, задерживая или отбрасывая пакеты до того, как они попадут в приложение.

Были сконфигурированы следующие профили:

High RTT: tc qdisc add dev lo root netem delay 100ms — имитация спутникового канала или перегруженной 3G сети.

Packet Loss: tc qdisc add dev lo root netem loss 5% — имитация радиопомех или плохого покрытия.

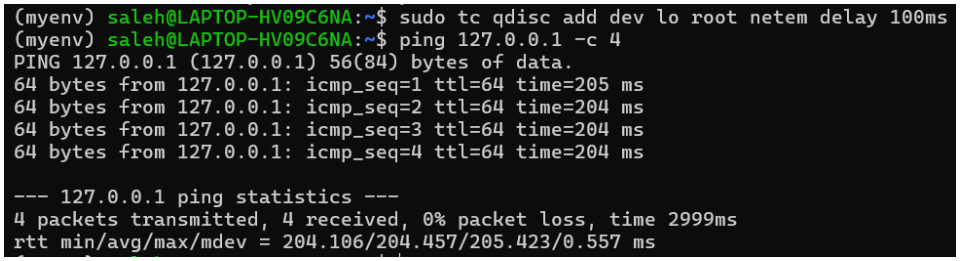

Рисунок 3 – Работа эмулятора

Результаты измерений

В рамках проведенной серии испытаний была выполнена передача и обработка массива из 800 контрольных сообщений. Для обеспечения достоверности и воспроизводимости результатов сбор телеметрии осуществлялся в автоматическом режиме с последующей сериализацией в формат JSON. Такой подход обеспечил машиночитаемую структуру данных для дальнейшего статистического анализа.

Во всех случаях была 100% доставка сообщений, остальные значение приведены на графиках.

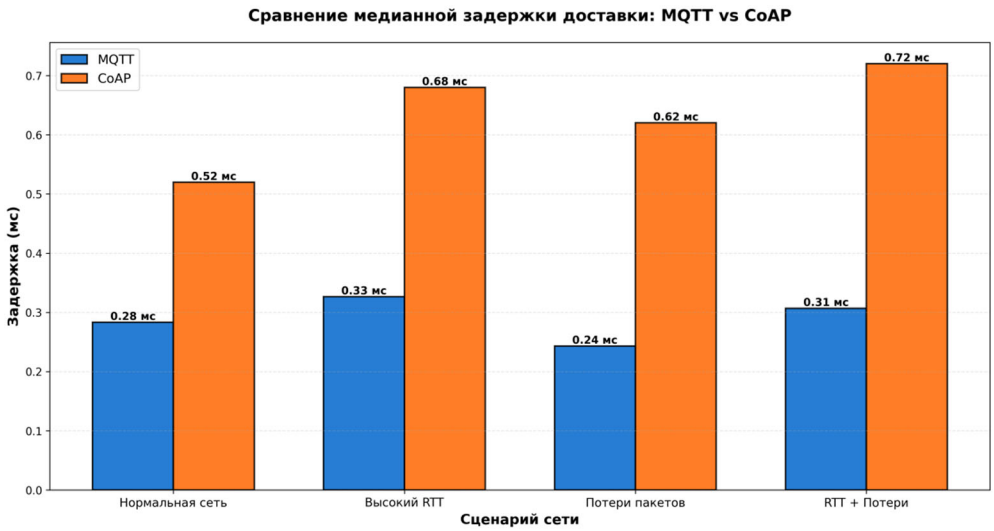

Рисунок 4 – Сравнение медианной задержки доставки

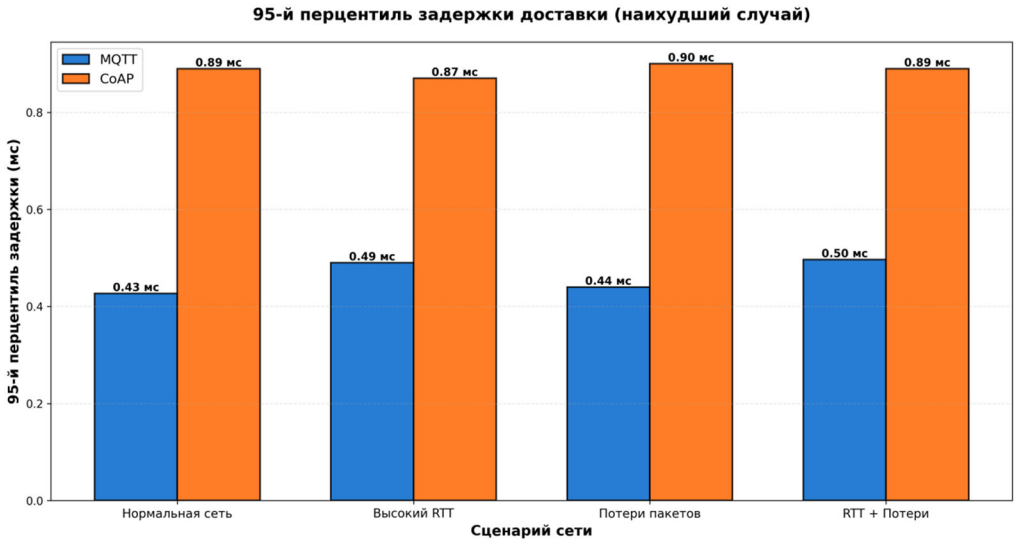

Рисунок 5 – Сравнение 95-го перцентиля задержки доставки

Анализ результатов

Зафиксированная разница в задержках на локальном интерфейсе (0.29 мс для MQTT против 0.64 мс для CoAP) нивелирует влияние среды передачи и демонстрирует чистую вычислительную стоимость стеков.

Двукратное преимущество MQTT (2.2x) обусловлено амортизацией накладных расходов: в рамках установленной TCP-сессии передача полезной нагрузки (Payload) происходит потоковым методом с минимальными затратами на инкапсуляцию заголовков. Реализация Mosquitto эффективно утилизирует механизмы ядра ОС (Kernel space) для буферизации и передачи данных.

В противовес этому, CoAP, функционирующий поверх UDP, вынужден обрабатывать каждую дейтаграмму независимо.

Полученные результаты согласуются с выводами о сравнении протоколов передачи данных в Интернете вещей [6], где MQTT демонстрирует преимущество в условиях стабильных TCP-сетей за счет QoS-механизмов, несмотря на большую задержку по сравнению с UDP-протоколами.

Критически важным результатом эксперимента является подтверждение 100% надежности доставки (Reliability) для обоих протоколов, даже в сценариях с эмуляцией 5% потерь пакетов.

MQTT достигает этого за счет встроенных механизмов транспортного уровня TCP (Retransmission), которые прозрачно для приложения восстанавливают потерянные сегменты.

CoAP обеспечивает аналогичный уровень надежности на прикладном уровне, используя механизм подтверждаемых сообщений (CON), который инициирует повторную отправку при отсутствии ACK-ответа.

Анализ стандартного отклонения задержек выявил более высокую временную детерминированность протокола MQTT в стабильных сетевых условиях. Постоянное соединение исключает необходимость повторных "рукопожатий", обеспечивая стабильный временной профиль передачи. CoAP продемонстрировал большую дисперсию задержек, что характерно для датаграммных протоколов, где каждый пакет маршрутизируется и обрабатывается изолированно, подвергаясь влиянию планировщика процессов ОС при каждом акте приёма-передачи.

Заключение

В ходе проведённого исследования был выполнен сравнительный анализ протоколов MQTT и CoAP применительно к задаче передачи телеметрии в шлюзовой системе умного дома. Эксперименты показали, что для полноценной и стабильной работы системы мониторинга наиболее подходящим протоколом является MQTT, который обеспечивает наилучший баланс между производительностью и надежностью.

Сравнение показателей выявило, что отказ от постоянного соединения хотя и допустим, но приводит к увеличению дисперсии задержек. MQTT же, благодаря потоковой природе TCP [5], обеспечивает надежную доставку данных без критических задержек даже в условиях эмуляции нестабильного Wi-Fi соединения, что критически важно для своевременного обновления показаний.

Также было установлено, что оба протокола способны гарантировать 100% доставку сообщений, однако MQTT достигает этого с меньшими накладными расходами на процессор шлюза. Это позволяет эффективнее использовать вычислительные ресурсы при масштабировании количества датчиков.

Таким образом, при разработке инфраструктуры умного дома рекомендуется использовать протокол MQTT в качестве основного стандарта транспорта — это позволяет добиться максимальной оперативности обнаружения изменений микроклимата, стабильности канала связи и минимизации рисков потери данных при сетевых сбоях.

Список литературы:

1. MQTT Version 5.0 — OASIS Standard. – URL: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html (дата обращения: 20.10.2025). – Текст: электронный.
2. RFC 7252: The Constrained Application Protocol (CoAP) — IETF Datatracker. – URL: https://datatracker.ietf.org/doc/html/rfc7252 (дата обращения: 22.10.2025). – Текст: электронный.
3. MQTT vs CoAP: IoT Protocols Compared — HiveMQ Blog. – URL: https://www.hivemq.com/blog/mqtt-vs-coap-iot-protocols-compared/ (дата обращения: 25.10.2025). – Текст: электронный.
4. Eclipse Mosquitto Documentation — Eclipse Foundation. – URL: https://mosquitto.org/documentation/ (дата обращения: 10.11.2025). – Текст: электронный.

5. TCP vs UDP: What's the Difference? — PCMag. – URL: https://www.pcmag.com/encyclopedia/term/tcp-vs-udp (дата обращения: 15.11.2025). – Текст: электронный.
6. Курмаев Т.И. Сравнение протоколов передачи данных в Интернете вещей // Международный научно-исследовательский журнал. – 2022. – № 1(115). – С. 68–74.
7. Корзухин С.В., Хайдарова Р.Р., Шматков В.Н. Конфигурируемые IoT-устройства на основе SoC-систем ESP8266 и протокола MQTT // Научно-технический вестник информационных технологий, механики и оптики. – 2020. – Т. 20, № 5. – С. 722–728.
8. Макшанский А.Р. Протоколы передачи данных Интернета вещей // Научный Лидер. – 2023. – № 7(105). – С. 10–17.

References:

1. MQTT Version 5.0 — OASIS Standard. – URL: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html (accessed: October 20, 2025). – Text: electronic.
2. RFC 7252: The Constrained Application Protocol (CoAP) — IETF Datatracker. – URL: https://datatracker.ietf.org/doc/html/rfc7252 (accessed: October 22, 2025). – Text: electronic.
3. MQTT vs CoAP: IoT Protocols Compared — HiveMQ Blog. – URL: https://www.hivemq.com/blog/mqtt-vs-coap-iot-protocols-compared/ (accessed: October 25, 2025). – Text: electronic.
4. Eclipse Mosquitto Documentation — Eclipse Foundation. – URL: https://mosquitto.org/documentation/ (accessed: November 10, 2025). – Text: electronic.
5. TCP vs UDP: What's the Difference? — PCMag. – URL: https://www.pcmag.com/encyclopedia/term/tcp-vs-udp (accessed: November 15, 2025). – Text: electronic.
6. Kurmaev T.I. Comparison of Data Transmission Protocols in the Internet of Things // International Research Journal. – 2022. – No. 1(115). – Pp. 68–74.
7. Korzukhin S.V., Khaydarova R.R., Shmatkov V.N. Configurable IoT Devices Based on ESP8266 SoC Systems and the MQTT Protocol // Scientific and Technical Bulletin of Information Technologies, Mechanics, and Optics. – 2020. – Vol. 20, No. 5. – Pp. 722–728.
8. Makshansky A.R. Internet of Things data transmission protocols // Scientific Leader. – 2023. – № 7(105). – Pp. 10-17.