Научная статья
УДК 004.057.4; DOI: 10.61260/2218-13X-2024-4-141-150

ПОДХОДЫ К МАСШТАБИРОВАНИЮ ПРОТОКОЛА MQTT В СИСТЕМАХ МЕЖМАШИННОГО ВЗАИМОДЕЙСТВИЯ

✉**Котилевец Игорь Денисович.**
МИРЭА – Российский технологический университет, Москва, Россия
✉***ikotilevets@gmail.com***

Аннотация. Анализируются подходы к масштабированию протокола MQTT и его модификации MQTT-SN в системах межмашинного взаимодействия. Описываются архитектура протокола, модель публикации/подписки, уровни качества обслуживания, структура пакетов, проблемы, связанные с увеличением числа устройств и объема информации. Рассматриваются проблемы узких мест, такие как перегрузка брокеров и неравномерное распределение нагрузки. Проводится анализ возможностей масштабирования протокола и предложены архитектурные решения для повышения его производительности. Экспериментальная часть включает оценку производительности протокола через тесты на ширину канала, потери пакетов и задержки при различных нагрузках. Результаты показывают, что увеличение числа устройств приводит к росту задержек и потерь пакетов, что подчеркивает необходимость оптимизации. В заключение предлагаются рекомендации для улучшения масштабируемости.

Ключевые слова: MQTT, масштабируемость, межмашинное взаимодействие, протокол передачи данных, брокеры сообщений, модель публикация/подписка

Для цитирования: Котилевец И.Д. Подходы к масштабированию протокола MQTT в системах межмашинного взаимодействия // Науч.-аналит. журн. «Вестник С.-Петерб. ун-та ГПС МЧС России». 2024. № 4. С. 141–150. DOI: 10.61260/2218-13X-2024-4-141-150.

Scientific article

APPROACHES TO SCALING THE MQTT PROTOCOL IN MACHINE-TO-MACHINE INTERACTION SYSTEMS

✉**Kotilevets Igor D.**
MIREA – Russian technological university, Moscow, Russia
✉***ikotilevets@gmail.com***

Abstract. This article analyzes approaches to scaling the MQTT protocol and its modification, MQTT-SN, in machine-to-machine interaction systems. The architecture of the protocol, the publish/subscribe model, quality of service levels, packet structure, and issues related to the increasing number of devices and data volume are described. Problems such as broker overload and uneven load distribution are discussed. The analysis of scaling possibilities for the protocol is conducted, and architectural solutions are proposed to enhance its performance. The experimental part includes performance evaluation of the protocol through tests on bandwidth, packet loss, and latency under various loads. The results show that an increase in the number of devices leads to higher latencies and packet losses, highlighting the need for optimization. Recommendations for improving scalability are provided in conclusion.

Keywords: MQTT, scalability, machine-to-machine interaction, data transmission protocol, message brokers, publish/subscribe model

For citation: Kotilevets I.D. Approaches to scaling the MQTT protocol in machine-to-machine interaction systems // Scientific and analytical journal «Vestnik Saint-Petersburg university of State fire service of EMERCOM of Russia». 2024. № 4. P. 141–150. DOI: 10.61260/2218-13X-2024-4-141-150.

Введение

Парадигма публикации/подписки является одним из методов коммуникации в системах и сетях межмашинного взаимодействия (M2M). Протокол MQTT (Message Queuing Telemetry Transport) выступает одним из самых популярных протоколов для реализации этого подхода, благодаря своей простоте и низкому потреблению ресурсов. Однако с ростом числа подключенных устройств и объемов данных вопрос масштабируемости MQTT становится особенно актуальным.

Цель данной статьи – проанализировать возможности масштабирования MQTT и предложить архитектурные решения для повышения его эффективности в условиях роста нагрузки в сетях M2M.

Протокол MQTT основан на модели публикации/подписки, где участники делятся на издателей и подписчиков. Издатели передают данные брокеру – центральному элементу сети, который обрабатывает и распределяет сообщения подписчикам (рис. 1). Эта модель позволяет реализовать гибкую и эффективную передачу данных, особенно в условиях ограниченных ресурсов устройств M2M.

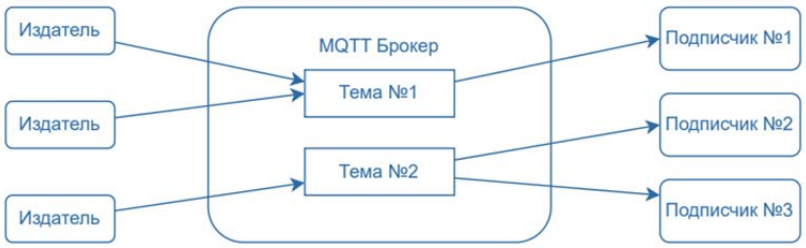

Рис. 1. **Архитектура протокола MQTT**

Протокол MQTT позволяет множеству подписчиков подключаться к различным темам. Это дает возможность получать разнообразные данные, не взаимодействуя с издателем. Темы представляют собой строки, закодированные в UTF-8, и имеют иерархическую структуру. Каждый уровень в дереве тем отделяется символом «/», что облегчает их организацию [1].

В этом протоколе есть три уровня качества обслуживания. Первый уровень не требует подтверждения доставки сообщений. Второй уровень подразумевает, что сервер должен подтвердить получение сообщения. Третий уровень гарантирует, что сообщение будет доставлено без дублирования.

Структура пакета MQTT включает три основные части. Фиксированный заголовок содержит тип сообщения, уровень качества обслуживания и информацию о дублировании. Переменный заголовок включает идентификатор пакета и флаги соединения. Кадр данных передает полезную информацию.

Протокол работает на базе TCP, но также может использовать другие сетевые протоколы. Это важно для обеспечения упорядоченных двусторонних соединений. Для беспроводных сенсорных сетей создана версия под названием MQTT-SN. Она расширяет возможности протокола, позволяя использовать его в сетях Zigbee или LoRaWAN [2, 3].

MQTT предлагает три уровня качества обслуживания для доставки сообщений. Более высокий уровень обеспечивает надежную доставку, но может увеличить пропускную способность или вызвать задержки. Протокол сохраняет сообщения на сервере даже после их отправки. Если к теме подключается новый подписчик, ему пересылаются ранее сохраненные сообщения.

При подключении клиента к серверу по протоколу можно установить параметр очистки сессии. Этот параметр определяет, будут ли удалены подписки клиента при отключении.

Клиент также может уведомить сервер о наличии сообщения-завещания. Если клиент отключается неожиданно, это сообщение будет опубликовано в указанной теме. Эта функция важна для систем оповещения и безопасности. Она позволяет быстро реагировать на потерю соединения с удалённым датчиком. Благодаря этим возможностям протокол MQTT подходит для различных приложений в области M2M [4–7].

Основной недостаток классической модели заключается в том, что брокер, обеспечивающий работу издателей и подписчиков, становится узким местом и потенциальной точкой отказа. В типичных клиент-серверных системах проблемы масштабирования решаются за счет добавления серверов, но в случае с MQTT брокеры играют роль посредников, что требует дополнительных архитектурных решений для обеспечения масштабируемости. Еще одной проблемой является низкая производительность брокеров при работе в однопоточном режиме. Большинство реализаций MQTT-брокеров работают в одном потоке, что делает невозможным использование всех доступных процессорных ядер на машине [8, 9].

Методы исследования

В данной статье использовался комплексный подход, включающий экспериментальный, симуляционный, аналитический и сравнительный методы для оценки производительности сети с использованием протокола MQTT и ZigBee. Экспериментальный метод включал проведение серии экспериментов для оценки производительности сети в зависимости от количества подключенных устройств. В ходе экспериментов измерялись метрики, такие как задержка, потеря пакетов и используемая пропускная способность. Симуляционный метод использовался для изучения влияния различных факторов, таких как количество устройств и наличие резервных каналов, на показатели сети. Моделирование позволило оценить поведение системы в различных сценариях и выявить проблемы. Аналитический метод включал анализ полученных данных для выявления закономерностей и проблем, связанных с увеличением нагрузки и числа устройств. Сравнительный метод использовался для сравнения метрик производительности до и после увеличения нагрузки, а также до и после внедрения резервных каналов. Сравнение данных позволило оценить эффективность резервных каналов в повышении производительности сети. Методы математического моделирования были использованы для моделирования времени сбоя и восстановления соединения. Применение формул позволило получить данные о поведении системы при сбоях и восстановлении, чтобы оценить надежность и устойчивость сети.

В проводимых экспериментах производительность протоколов измерялась на основе следующих метрик:

1. Используемая ширина канала связи.
2. Средний размер передаваемого сообщения.
3. Процент потерянных пакетов при сетевых сбоях.

Эксперимент 1 (без дублирующих маршрутов). Макет системы состоит из 3 000 датчиков, шлюза, сервера и рабочей станции. Датчики собирают данные об окружающей среде каждые 5 сек. и отправляют их на центральный сервер через ZigBee сеть. Эксперимент проводился в течение 10 мин, по результатам которого были рассчитаны средние значения каждой метрики.

Эксперимент 2 (с дублирующими маршрутами). Макет системы дополнительно включает дублирующие маршруты связи для каждых 100 датчиков. При возникновении ошибки при передаче сообщения приложение датчика перенаправляет данные по альтернативному маршруту. Эксперимент проводился в течение 10 мин, по результатам которого также были рассчитаны средние значения метрик.

Эксперимент 3 включает увеличение количества устройств и расчет оптимального значения шлюзов без резервных каналов с ними.

В каждом эксперименте уровень качества обслуживания устанавливался таким, чтобы сообщение доставлялось по крайней мере один раз. Формула рассчета:

$$t_{\text{с}} = -t_{\text{ср.дл.с}} * ln(1 - rnd),$$

где $t_{\text{ср.дл.с}}$ – среднее время между сбоями (в рамках эксперимента принималось равным 60 сек.); rnd – случайное число от 0 до 1, полученное с использованием равномерного распределения.

Время восстановления моделировалось по формуле:

$$t_{\text{в}} = -t_{\text{ср.дл.в}} * ln(1 - rnd),$$

где $t_{\text{ср.дл.в}}$ – среднее время восстановления соединения (в рамках эксперимента принималось равным 2 сек.).

Так как проводится моделирование необходимы следующие предположения: средний размер сообщения для одного датчика: 200 байт (данные об окружающей среде, например, температура, влажность и т.д.), ширина канала связи: 250 Кбит/с для одного шлюза при отправке сообщений через Zigbee.

Результаты исследования

В рамках первого эксперимента, включавшего 3 000 датчиков и один шлюз, за десятиминутный интервал было зафиксировано 360 000 сообщений. Объем переданных данных составил 72 МВ, что эквивалентно 576 Мбит. Средняя пропускная способность канала связи за этот период достигла 0,96 Мбит/с, при этом средний размер одного сообщения равнялся 200 байтам. Уровень потерь пакетов в данном эксперименте составил 5 %.

Во втором эксперименте с использованием резервирования для тех же 3 000 датчиков и одного шлюза, общее количество сообщений за 10 мин оставалось неизменным – 360 000 сообщений. Однако объем данных увеличился из-за применения резервных маршрутов для 10 % сообщений, что привело к приросту объема данных на 10 % из-за служебной информации, составив 7,92 МВ. Таким образом, общий объем данных достиг 79,92 МВ, что эквивалентно 639,36 Мбит, при этом средняя пропускная способность канала возросла до 1,07 Мбит/с. Средний размер сообщения с учетом передачи по резервному маршруту увеличился до 220 байт, а процент потерь пакетов снизился до 1 %.

Третий эксперимент включал увеличение количества устройств и шлюзов. В условиях с 5 000 датчиками и одним шлюзом средняя пропускная способность канала связи составила 1,6 Мбит/с, средний размер сообщения остался на уровне 200 байт, однако уровень потерь пакетов возрос до 10 %. При увеличении числа шлюзов до двух для тех же 5 000 датчиков пропускная способность снизилась до 0,8 Мбит/с на каждый шлюз, сохраняя средний размер сообщения в 200 байт, при этом процент потерянных пакетов уменьшился до 5 %. При увеличении количества датчиков до 7 000 и использовании двух шлюзов, средняя пропускная способность составила 1,12 Мбит/с на шлюз, а уровень потерь пакетов составил 8 %, в то время как средний размер сообщения остался неизменным – 200 байт. Наконец, при увеличении числа датчиков до 9 000 и использовании четырех шлюзов пропускная способность снизилась до 0,72 Мбит/с на каждый шлюз, сохраняя средний размер сообщения на уровне 200 байт, при этом процент потерь пакетов уменьшился до 2 %.

Все результаты показаны на рис. 2–4.

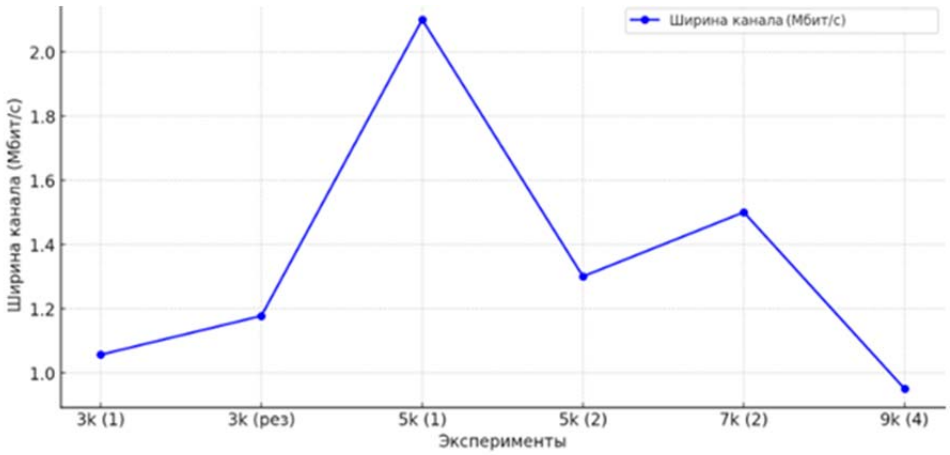

Рис. 2. **Ширина каналов в экспериментах**

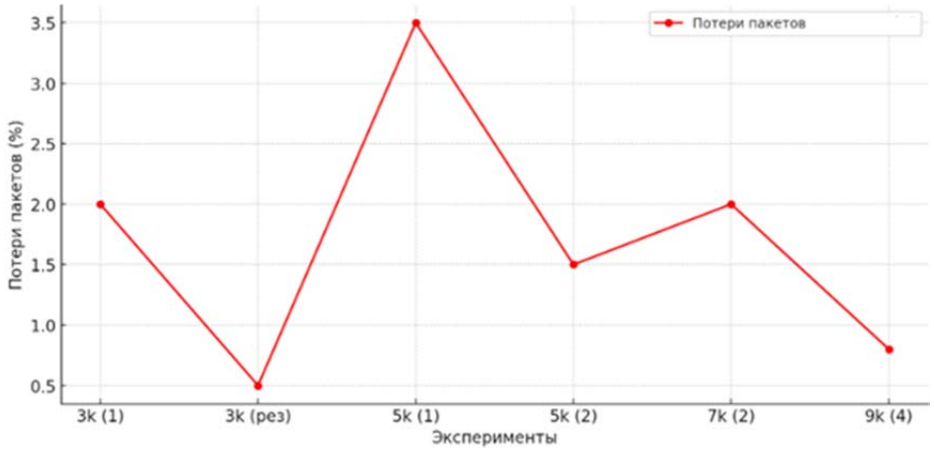

Рис. 3. **Процент потерянных пакетов в экспериментах**

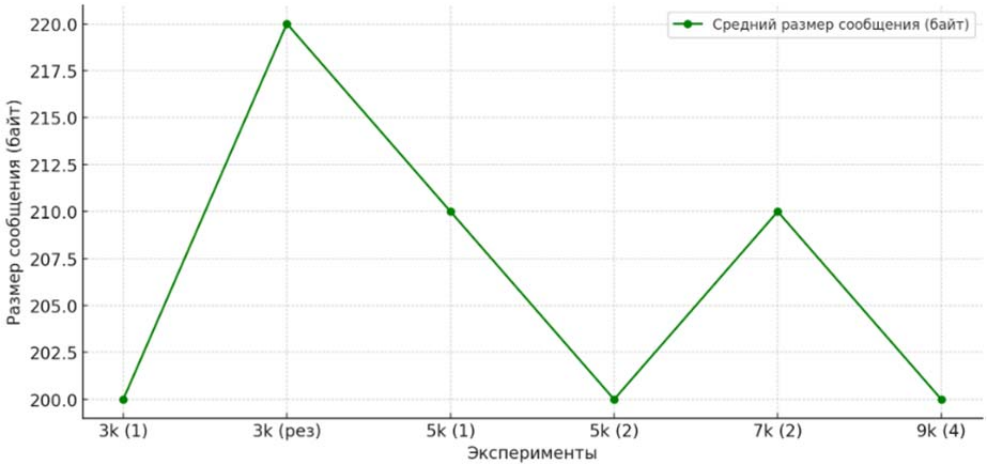

Рис. 4. **Средний размер сообщений в экспериментах**

Добавим в данные по третьему эксперименту резервные каналы. В таком случае ширина канала увеличится, так как будут дублироваться данные по резервному каналу, а потери пакетов уменьшатся из-за наличия альтернативных путей передачи. С другой стороны, средний размер сообщений останется прежним, так как резервирование не изменяет размер пакетов.

Исходные данные протокола без резервных каналов характеризуются следующими параметрами. Ширина канала связи составляет 1,6 Мбит/с, 0,8 Мбит/с, 1,12 Мбит/с и 0,72 Мбит/с. Потери пакетов составляют 10 %, 5 %, 8 % и 2 % соответственно. Средний размер сообщения равен 200 байтам.

При использовании резервных каналов наблюдается изменение ширины канала связи и процента потерянных пакетов. Например, при использовании 5 000 датчиков и одного шлюза ширина канала связи увеличивается с 1,6 Мбит/с до 1,76 Мбит/с, а процент потерянных пакетов уменьшается с 10 % до 5 %. Аналогичные изменения наблюдаются и в других сценариях: при использовании 5 000 датчиков и двух шлюзов, 7 000 датчиков и двух шлюзов, а также 9 000 датчиков и четырех шлюзов.

Теперь необходимо определить, при каком уровне нагрузки возникает проблема «бутылочного горлышка» в системе.

В эксперименте 4 участвуют 100 000 устройств, которые подключены к 10 шлюзам с распределением нагрузки. Каждый шлюз имеет резервный канал для обеспечения надежности связи. Пропускная способность сети составляет 250 кбит/с. Каждое устройство отправляет пакет данных размером 100 байт каждые 5 сек.

В ходе эксперимента проводится тестирование отказа, при котором один из шлюзов искусственно отключается на 50 % времени эксперимента. Это позволяет оценить поведение системы в условиях отказа и проверить эффективность резервных каналов в обеспечении надежности связи.

Далее устройства начинают передавать данные каждую одну секунду, увеличивая общую нагрузку на шлюзы, в процессе измеряются метрики производительности для каждого шлюза (задержки, потери пакетов и пропускная способность).

На рис. 5 представлены результаты эксперимента, демонстрирующие влияние увеличения числа устройств на показатели задержки и потерь пакетов при использовании брокера MQTT и сетей ZigBee.

Задержка возрастает существенно с увеличением количества устройств. При 10 000 устройствах задержка составляет 50 мс до увеличения нагрузки и 70 мс после увеличения нагрузки. При 100 000 устройствах задержка составляет 150 мс до увеличения нагрузки и 350 мс после увеличения нагрузки. При 1 000 000 устройствах задержка возрастает до 600 мс до увеличения нагрузки и 1 500 мс после увеличения нагрузки, что указывает на перегрузку сети.

Аналогично, потери пакетов также увеличиваются с ростом числа устройств. При 10 000 устройствах потери пакетов составляют 1 % до увеличения нагрузки и 2 % после увеличения нагрузки. При 100 000 устройствах потери пакетов составляют 3 % до увеличения нагрузки и 8 % после увеличения нагрузки. При 1 000 000 устройствах потери пакетов составляют 12 % до увеличения нагрузки и 25 % после увеличения нагрузки.

Эти результаты указывают на то, что сеть не может справиться с большим количеством устройств и нагрузкой, что приводит к увеличению задержки и потерь пакетов.

Как видно из экспериментов, с увеличением количества устройств возрастает и число используемых шлюзов, что может привести к ряду проблем. Одной из них является неравномерное распределение нагрузки даже при наличии нескольких шлюзов. Недостаточная оптимизация сессий может перегружать некоторые шлюзы, вызывая задержки в передаче данных.

Повышенная нагрузка на сеть при увеличении числа устройств также становится значительной проблемой. Ограниченная пропускная способность сети может вызывать задержки, потери пакетов и замедление обработки данных. При этом даже наличие резервных каналов не всегда позволяет избежать роста процента потерянных пакетов, особенно в условиях высокой сетевой нагрузки или частых сбоев.

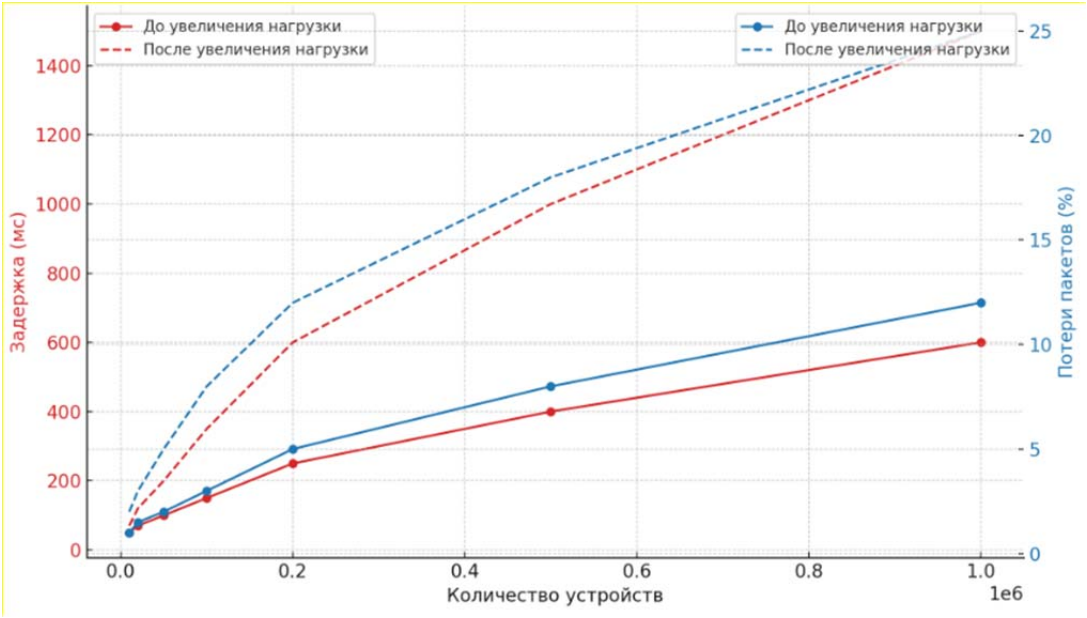

Рис. 5. **Эксперимент с «бутылочным горлышком»**
(красные линии показывают рост задержек до и после увеличения нагрузки;
синие линии отражают потери пакетов до и после увеличения нагрузки)

Еще одной потенциальной проблемой является перегрузка брокеров. В таких случаях большое число подключений может приводить к разрыву сессий и временной потере связи. Переход сессий между перегруженными брокерами вызывает дополнительные задержки, что также влияет на общую стабильность системы.

Когда объём данных становится слишком большим, задержки между сбором данных устройствами и их доставкой возрастают.

Даже наличие резервных каналов не всегда гарантирует достаточное количество ресурсов для обслуживания всех устройств в случае сбоев. Одновременный выход из строя нескольких шлюзов может привести к отключению части устройств от сети. При множественных отказах возникает риск появления «бутылочного горлышка», когда один из шлюзов или каналов становится перегруженным, снижая общую пропускную способность системы.

Кроме того, рост числа устройств в сети увеличивает количество потенциальных точек входа для кибератак. Защита всей инфраструктуры становится более сложной, и каждый шлюз и брокер должен обеспечивать надежную аутентификацию устройств и защиту данных. С увеличением числа подключённых устройств и соединений поддержание безопасности становится сложнее. Если устройства работают на батарейках или других ограниченных источниках питания, частые передачи данных и повышенная нагрузка могут быстро разряжать батареи. Это особенно важно для оборудования, которое установлено в труднодоступных местах и где замена батарей представляет проблему.

Чтобы управлять крупной и расширяющейся сетью требуются современные инструменты наблюдения и управления. При возникновении сбоев необходимо быстро реагировать и устранять их, используя сложные программы и системы, которые обеспечивают стабильную работу сети.

Для расширения возможностей протоколов MQTT и его версии MQTT-SN существует несколько методов. Каждый из них зависит от конкретных условий, например, количества устройств, числа брокеров, уровня задержек и стабильности соединений.

Разделение брокеров [10]. Этот метод предполагает разделение всей системы на несколько независимых брокеров. Каждый брокер обслуживает определённую группу подписчиков или тем. Это помогает снизить нагрузку на каждого брокера и улучшить работу

сети. Например, брокеры работают отдельно и связываются друг с другом только тогда, когда нужно передать данные в другой регион.

Автоматическое распределение сессий. В этом подходе соединения клиентов автоматически перераспределяются между брокерами в зависимости от текущей нагрузки. Когда новое устройство подключается, система выбирает брокера, который менее загружен. Это помогает сбалансировать нагрузку и избежать перегрузок. Если брокер начинает перегружаться, система может перенаправить устройства на другой сервер.

Разделение данных. Сообщения от устройств группируются по категориям, чтобы уменьшить нагрузку на сеть. Например, данные от сенсоров могут быть разделены по типу, и подписчики получают только ту информацию, которая им нужна. Это снижает объём трафика и снижает нагрузку на серверы, так как не все данные передаются всем подписчикам.

Маршрутизация на основе алгоритмов. Для управления потоками данных используются алгоритмы, которые выбирают лучшие пути передачи сообщений. Они учитывают скорость, доступность и загруженность сети. Алгоритмы могут меняться в зависимости от текущих условий и анализировать данные, чтобы оптимизировать передачу сообщений [11].

Использование ациклического направленного графа [12, 13]. В сетях MQTT и MQTT-SN можно использовать ациклический граф для маршрутизации данных. Каждый узел сети в таком случае будет связан с другим узлом в одном направлении, что позволит избежать возвратов и циклов, что, в свою очередь, поможет определить самые быстрые пути для передачи сообщений и повышает стабильность соединений. Как результат, в теории должны снизиться задержки и повысится надежность, так как можно обойти проблемные узлы. Кроме того, структурированная маршрутизация упрощает масштабирование всей системы.

Ациклический направленный граф особенно подходит для крупных распределенных сетей, где крайне важно минимизировать задержки и обеспечить надежность работы, особенно в условиях постоянного добавления новых устройств.

Эти методы помогают улучшить работу сети, распределить нагрузку и предотвратить перегрузки. Это важно для систем, которые работают с большим количеством подключений и устройств.

Заключение

Протокол MQTT и его модифицированная версия MQTT-SN подходят для передачи данных в условиях ограниченных возможностей. Однако с ростом числа устройств и объёма информации появляются серьёзные проблемы, влияющие на производительность и надёжность всей системы.

Проведенный эксперимент показал, что при повышении нагрузки сеть начинает испытывать задержки, и увеличивается количество потерянных пакетов. Это указывает на необходимость улучшения архитектуры и управления работой брокеров. Проблемы, такие как перегрузка и неравномерное распределение данных, требуют дополнительных решений. Одним из таких подходов может быть разделение сети на несколько независимых брокеров или использование горизонтального масштабирования. Также полезно рассмотреть внедрение технологий, таких как искусственный интеллект или использование направленных графов для организации инфраструктуры.

Для того чтобы улучшить надёжность и устойчивость систем M2M, важно разрабатывать новые алгоритмы, которые смогут эффективно управлять нагрузкой и поддерживать безопасность. Внедрение современных инструментов мониторинга также повысит стабильность и производительность сети. Если удастся оптимизировать протокол MQTT, его применение станет более широким и откроет возможности для использования в таких областях, как умные города, промышленные сети Интернета вещей и другие масштабные системы.

Список источников

1. MQTT Specification. URL: https://mqtt.org/mqtt-specification/ (дата обращения: 02.10.2024).

2. Banks A., Briggs E., Borgendale K., Gupta R. MQTT Version 5.0. OASIS Standard, 7 March 2019.

3. Stanford-Clark A., Truong H.L. MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2. IBM, 14 November 2013.

4. Якупов Д.Р. Обзор и сравнение протоколов интернета вещей: MQTT и AMQP // International Journal of Open Information Technologies. 2022. № 9.

5. Дикий Д.И., Артемьева В.Д. Протокол передачи данных MQTT в модели удаленного управления правами доступа для сетей Интернета // Научно-технический вестник информационных технологий, механики и оптики. 2019. № 1.

6. Костеннов Т.В. Об эффективности агрегации данных в сетях интернета вещей с использованием протокола MQTT // МСиМ. 2023. № 4 (68).

7. Костеннов Т.В. Сравнение протоколов связи для организации M2M-взаимодействий в SCADA-системах и системах промышленного интернета вещей // МСиМ. 2023. № 2 (66).

8. Курмаев Т.И. Сравнение протоколов передачи данных в интернете вещей // МНИЖ. 2022. № 1-1 (115).

9. Bevilaqua B., Spohn M. Self-Managed Federation of MQTT Brokers with Dynamic Topology Control // Journal of Computer Science. 2023. Vol. 19. P. 1398–1409. DOI: 10.3844/jcssp.2023.1398.1409.

10. Sharma S., Peddoju S.K. Efficient Multi-Broker Load Balancing in Event Driven Pub-Sub Networks // IEEE Transactions on Network and Service Management. 2024. Vol. 21. № 4. P. 3861–3873. DOI: 10.1109/TNSM.2024.3401484.

11. Sharma A., Babbar H. Towards Resilient IoT Security: An Analysis and Classification of Attacks in MQTT-based Networks // 2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT). IEEE, 2024. P. 122–125.

12. Kotilevets I.D., Ivanova I.A., Romanov I.O. Implementation of directed acyclic graph in blockchain network to improve security and speed of transactions // IFAC-PapersOnLine. 2018. Vol. 51. Iss. 30. P. 693–696. DOI: 10.1016/j.ifacol.2018.11.213.

13. Котилевец И.Д., Иванова И.А. Применение направленного ациклического графа в блокчейн сети для повышения безопасности и скорости транзакций в промышленности // Промышленные АСУ и контроллеры. 2019. № 1. P. 53–59.

References

1. MQTT Specification. URL: https://mqtt.org/mqtt-specification/ (data obrashcheniya: 02.10.2024).

2. Banks A., Briggs E., Borgendale K., Gupta R. MQTT Version 5.0. OASIS Standard, 7 March 2019.

3. Stanford-Clark A., Truong H.L. MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2. IBM, 14 November 2013.

4. Yakupov D.R. Obzor i sravnenie protokolov interneta veshchej: MQTT i AMQP // International Journal of Open Information Technologies. 2022. № 9.

5. Dikij D.I., Artem'eva V.D. Protokol peredachi dannyh MQTT v modeli udalennogo upravleniya pravami dostupa dlya setej Interneta // Nauchno-tekhnicheskij vestnik informacionnyh tekhnologij, mekhaniki i optiki. 2019. № 1.

6. Kostennov T.V. Ob effektivnosti agregacii dannyh v setyah interneta veshchej s ispol'zovaniem protokola MQTT // MSiM. 2023. № 4 (68).

7. Kostennov T.V. Sravnenie protokolov svyazi dlya organizacii M2M-vzaimodejstvij v SCADA-sistemah i sistemah promyshlennogo interneta veshchej // MSiM. 2023. № 2 (66).

8. Kurmaev T.I. Sravnenie protokolov peredachi dannyh v internete veshchej // MNIZH. 2022. № 1-1 (115).

9. Bevilaqua B., Spohn M. Self-Managed Federation of MQTT Brokers with Dynamic Topology Control // Journal of Computer Science. 2023. Vol. 19. P. 1398–1409. DOI: 10.3844/jcssp.2023.1398.1409.

10. Sharma S., Peddoju S.K. Efficient Multi-Broker Load Balancing in Event Driven Pub-Sub Networks // IEEE Transactions on Network and Service Management. 2024. Vol. 21. № 4. P. 3861–3873. DOI: 10.1109/TNSM.2024.3401484.

11. Sharma A., Babbar H. Towards Resilient IoT Security: An Analysis and Classification of Attacks in MQTT-based Networks // 2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT). IEEE, 2024. P. 122–125.

12. Kotilevets I.D., Ivanova I.A., Romanov I.O. Implementation of directed acyclic graph in blockchain network to improve security and speed of transactions // IFAC-PapersOnLine. 2018. Vol. 51. Iss. 30. P. 693–696. DOI: 10.1016/j.ifacol.2018.11.213.

13. Kotilevec I.D., Ivanova I.A. Primenenie napravlennogo aciklicheskogo grafa v blokchejn seti dlya povysheniya bezopasnosti i skorosti tranzakcij v promyshlennosti // Promyshlennye ASU i kontrollery. 2019. № 1. P. 53–59.

Информация о статье:
Статья поступила в редакцию: 20.10.2024; одобрена после рецензирования: 17.11.2024;
принята к публикации: 20.11.2024
Information about the article:
The article was submitted to the editorial office: 20.10.2024; approved after review: 17.11.2024;
accepted for publication: 20.11.2024

Сведения об авторах:
Котилевец Игорь Денисович, старший преподаватель кафедры КБ-14 «Цифровые технологии обработки данных» МИРЭА – Российского технологического университета (119454, Москва, пр. Вернадского, д. 78), email: ikotilevets@gmail.com, https://orcid.org/0000-0003-0433-3999, SPIN-код: 2544-2195

Information about the authors:
Kotilevets Igor D., senior lecturer of the department of KB-14 «Digital data processing technologies» of MIREA – Russian technological university (119454, Moscow, Vernadsky pr., 78), email: ikotilevets@gmail.com, https://orcid.org/0000-0003-0433-3999, SPIN: 2544-2195