

Обзор и сравнение протоколов Интернета Вещей: MQTT и AMQP

Д.Р. Якупов

Аннотация — Разработка программных приложений для устройств Интернета Вещей (IoT) сопряжена с решением множества не простых задач, одной из которых является выбор подходящего протокола для обмена информацией. Устройства Интернета Вещей имеют ряд особенностей, таких как низкая производительность, небольшой объем памяти, короткое время автономной работы и т.д., что в свою очередь накладывает соответствующие ограничения на выбор протокола связи. В данной статье представлен обзор и сравнение двух популярных протоколов прикладного уровня для Интернета Вещей: Advanced Message Queuing Protocol (AMQP) и Message Queuing Telemetry Transport (MQTT). Данные протоколы стандартизированы консорциумом Organization for the Advancement of Structured Information Standards (OASIS) и имеют открытую модель лицензирования. Для обмена сообщениями протоколы MQTT и AMQP используют модель «публикация / подписка», а качестве транспорта для связи между устройствами поддерживают использование протокола TCP. Кроме того, в протоколах MQTT и AMQP, реализована поддержка различных уровней качества обслуживания сообщений, позволяющих управлять надежностью их доставки. В статье рассмотрены ключевые особенности протоколов MQTT и AMQP, их архитектура, преимущества и недостатки. В заключении представлен сравнительный анализ протоколов и результаты экспериментального сравнения их производительности.

Ключевые слова — Интернет Вещей (IoT), Advanced Message Queuing Protocol (AMQP), Message Queuing Telemetry Transport (MQTT).

I. ВВЕДЕНИЕ

Технологические достижения последних двух десятилетий положили начало эпохе Интернета Вещей (Internet of Things, IoT). Интернет Вещей позволяет миллиардам устройств подключаться через интернет и обмениваться информацией и ресурсами на глобальном уровне. Устройства Интернета Вещей могут быть любыми, от смартфонов до миниатюрных встроенных датчиков [6].

Интернет Вещей позволил достичь высокого уровня технологических достижений во многих областях, таких как промышленность, сельское хозяйство, строительство, здравоохранение и повседневная частная жизнь.

Интернет Вещей расширяет интеллектуальные,

вычислительные и коммуникационные возможности для многочисленных повседневных устройств, таких как навигационные системы в автомобилях, стиральные машины, холодильники, кофеварки, будильники и т.д. Он объединяет воедино разнородные и выполняющие разные функции устройства формируя таким образом цифровой мир [6].

Основная цель Интернета Вещей - создать устройства, способные достигать поставленной цели с минимальным вмешательством человека или вообще без него [6].

При этом устройства Интернета Вещей имеют ряд особенностей, таких как низкая производительность, небольшой объем памяти, короткое время автономной работы и т.д. Подключение к сети может выполняться с помощью проводных и беспроводных технологий, однако основной акцент делается на беспроводную связь.

Данные особенности накладывают соответствующие ограничения на используемые протоколы связи, в связи с чем выбор протокола при разработке программных приложений для устройств Интернета Вещей является не простой задачей.

В настоящей статье обсуждается архитектура, преимущества и недостатки двух популярных протоколов прикладного уровня, применяемых в приложениях для Интернета Вещей, а именно протоколов Message Queuing Telemetry Transport (MQTT) и Advanced Message Queuing Protocol (AMQP). Работа выполнена на основе изучения и анализа изданных в последние годы статей по данной тематике.

Протоколы MQTT и AMQP являются открытыми и в качестве транспорта для связи между устройствами поддерживают протокол TCP.

В разделах II и III рассмотрены основные характеристики протоколов MQTT и AMQP соответственно, а в разделе IV приведен их сравнительный анализ.

II. ПРОТОКОЛ MESSAGE QUEUING TELEMETRY TRANSPORT (MQTT)

A. Введение

Первая версия протокола Message Queuing Telemetry Transport (далее – MQTT) была разработана в 1999 году Энди Стэнфордом-Кларком (Andy Stanford-Clark), работавшем в компании IBM, и Арленом Ниппером (Arlen Nipper), работавшем в компании Eurotech [1].

MQTT - это клиент-серверный протокол обмена сообщениями, который достаточно упрощен, чтобы его

Статья получена 11 мая 2022.

Д.Р. Якупов – МГУ имени М.В. Ломоносова (email: dyakupov1@gmail.com)

могли поддерживать самые миниатюрные устройства с минимальным объемом доступных ресурсов, но при этом он достаточно надежен, чтобы гарантировать, что важные сообщения всегда дойдут до места назначения [2].

С помощью MQTT такие устройства, как интеллектуальные счетчики энергии, автомобили, поезда, спутниковые приемники и персональные медицинские устройства, могут взаимодействовать друг с другом и с другими системами или приложениями [2].

Протокол MQTT работает поверх протокола TCP, но может также использовать другие сетевые протоколы, обеспечивающие упорядоченные двунаправленные соединения без потерь [3].

Для беспроводных сенсорных сетей (wireless sensor networks) разработана специальная версия протокола MQTT for Sensor Networks (MQTT-SN), предназначенная для расширения протокола MQTT за пределы досягаемости TCP/IP инфраструктуры, например, сетей Zigbee [4].

В 2014 году протокол MQTT (версия 3.1.1) был стандартизован консорциумом Organization for the Advancement of Structured Information Standards (OASIS), а в 2019 году консорциумом OASIS была издана 5-я, на текущий момент последняя, версия стандарта [3].

В. Ключевые принципы MQTT

Протокол MQTT построен с использованием нескольких ключевых принципов, нацеленных на обеспечение доставки сообщений при одновременной минимизации их объема [2, 3]:

1) Публикация/подписка

Принцип публикации/подписки (publisher/subscribe) определяет механизм передачи сообщений между источником и получателем информации. В соответствии с ним источник публикует информацию, указывая тему (раздел) к которой она относится, а получатель подписывается на интересующие его темы сообщений.

Источник информации принято называть издателем (publisher), а получателя – подписчиком (subscriber). При реализации данного принципа издателю необязательно знать о местонахождении подписчика и наоборот, а также обеспечивается передача сообщений от одного издателя ко многим подписчикам.

2) Темы и подписки

В MQTT сообщения публикуются с указанием тем (topics), которые можно рассматривать как названия предметных областей. Для получения определенных сообщений подписчики подписываются на соответствующие темы. Подписка может быть задана с явным указанием конкретных тем, по которым необходимо получать сообщения, либо она может быть определена с использованием подстановочных знаков (например, «#») для получения сообщений по различным связанным темам.

3) Уровни качества обслуживания

В протоколе MQTT определено три уровня качества обслуживания (quality of service) для доставки сообщений, чем выше уровень качества, тем больше уровень усилий прилагает сервер для обеспечения доставки сообщений. Более высокие уровни качества

обеспечивают более надежную доставку сообщений, но они могут потреблять большую пропускную способность сети или приводить к задержкам в доставке сообщений.

4) Сохранение сообщений

В протоколе MQTT сервер сохраняет сообщение даже после отправки его всем текущим подписчикам соответствующей темы. В случае подписки на данную тему нового подписчика ему отправляются все ранее сохраненные сообщения.

5) Долговременные соединения и очистка сессии

При подключении клиента к серверу по протоколу MQTT, клиент устанавливает значение параметра очистки сессии (clean session flag). Данный параметр определяет необходимость удаления (очистки) подписок клиента при отключении его от сервера (завершения сессии). При значении true – все подписки клиента удаляются, в случае false – предполагается, что соединение является долговременным и все подписки клиента сохраняются после любого его отключения. При значении false – сообщения, поступившие в период отключения клиента и имеющие высокий уровень качества обслуживания, сохраняются и доставляются клиенту после восстановления соединения. Использование параметра очистки сессии не является обязательным.

6) Сообщения - завещания

Когда клиент подключается к серверу, он может сообщить серверу, что у него есть сообщение - завещание (will), которое должно быть опубликовано в определенной теме (-ax) в случае неожиданного отключения клиента. Сообщение - завещание особенно полезно при настройках оповещений или безопасности, где система должна незамедлительно узнавать, когда удаленный датчик потерял соединение с сетью.

С. Архитектура MQTT

Протокол MQTT построен на основе клиент-серверной архитектуры, которая включает в себя два основных компонента [3, 5, 6]:

1) Клиент (client) – программа или устройство (датчик, RFID-метка, мобильное устройство и т.д.), использующее протокол MQTT и выполняющее следующие основные функции:

- установка и закрытие сетевых соединений с сервером;

- публикация сообщений, которые могут представлять интерес для других клиентов протокола MQTT;

- подписка на сообщения, в получении которых заинтересован данный клиент;

- отмена подписки для прекращения получения сообщений, к которым у данного клиента утрачен интерес;

2) Сервер (server) – программа или устройство, использующее протокол MQTT и выступающее в качестве посредника между клиентами, публикующими сообщения, и клиентами, оформившими подписки.

Сервер, также называемый брокером (broker), выполняет следующие основные функции:

- устанавливает и закрывает сетевые соединения с

клиентами;

- принимает сообщения, опубликованные клиентами;
- обрабатывает запросы клиентов на подписки и отмены подписок;
- перенаправляет сообщения от клиентов в соответствии с оформленными подписками.

Под сообщением (application message) здесь понимаются данные приложений, передаваемые по сети посредством протокола MQTT. Каждое сообщение, передаваемое по протоколу MQTT, включает в себя:

- полезную нагрузку (payload data), т.е. непосредственно данные приложения;
- уровень качества обслуживания;
- тему сообщения;
- набор дополнительных свойств, характеризующих передаваемое сообщение.

Архитектура протокола MQTT представлена на рис. 1.

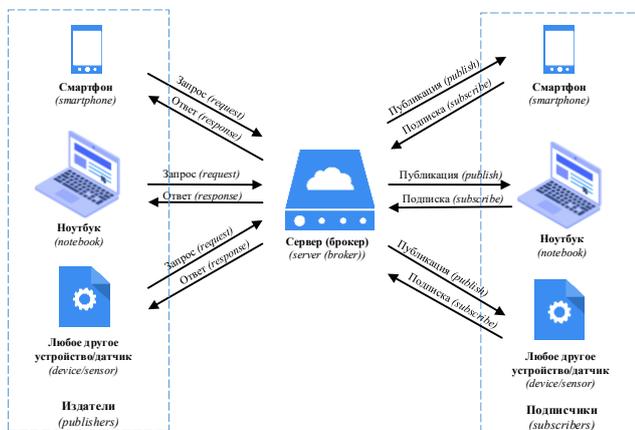


Рис. 1. Архитектура протокола MQTT

Протокол MQTT поддерживает три уровня качества обслуживания [3, 6]:

1) 0 – уровень («не более одного раза») – данный уровень качества подразумевает доставку сообщений в соответствии с возможностями используемой транспортной сети. Доставка сообщения не гарантируется, и оно может быть утеряно. Этот уровень качества может использоваться, например, при получении показаний с датчиков окружающей среды, когда потеря отдельного показания не является существенной, так вскоре после него будет отправлено новое текущее измеренное значение [3].

2) 1 – уровень («по крайней мере один раз») – данный уровень качества гарантирует доставку сообщения, но не гарантирует отсутствие получения дубликатов ранее принятых сообщений [3]. Этот уровень качества может использоваться в системах, в которых передаваемые датчиками показания являются кумулятивными, в таком случае сообщения, получаемые сервером более одного раза, должны иметь незначительный эффект [7]. Одним из способов исключить прием дубликатов может являться включение в каждое сообщение уникальной временной метки. Данный механизм, например, реализован в платформе Assetwolf, которая распознает повторяющиеся сообщения на основе метки времени и игнорирует их [7].

3) 2 – уровень («ровно один раз») – данный уровень гарантирует доставку сообщения и гарантирует отсутствие получения дубликатов ранее принятых сообщений. Этот уровень качества может использоваться, например, в системах выставления счетов, когда повторяющиеся или потерянные сообщения могут привести к неправильному начислению платежей.

Решение о том какой из уровней качества использовать влияет на производительность прикладного приложения, используемую ширину полосы пропускания сети и время автономной работы устройств [8].

Нулевой уровень является самым дешевым с точки зрения объема передаваемых данных, он подходит, когда используется надежное соединение между клиентом и сервером, либо потеря отдельных сообщений не является существенной [7].

Первый уровень более надежен, чем нулевой, в плане доставки сообщений, но он требует больше ресурсов, кроме того повторяющиеся сообщения могут вводить в заблуждение получателей информации [7].

Второй уровень обеспечит бесперебойную работу большинства прикладных приложений, но в тоже время он наиболее требователен к ресурсам.

В каждом конкретном случае рекомендуется проводить анализ возможности использования более низкого уровня качества обслуживания. Зачастую тщательный выбор полей в данных - использование временных меток и кумулятивных показаний - является простым способом избежать накладных расходов, связанных с использованием высокого уровня качества [7].

D. Преимущества и недостатки MQTT

Одним из главных преимуществ MQTT является его простота и легкость, т.е. в передаваемых данных объем служебной информации, используемой для функционирования протокола, сведен к минимуму. Благодаря этому он идеально подходит для удаленного мониторинга [6].

Протокол MQTT используется в промышленном Интернете Вещей для обеспечения надлежащего функционирования SCADA-систем (систем диспетчерского управления и сбора данных). MQTT предоставляет много серьезных преимуществ, таких как, например, сокращение потребления пропускной способности сети, что в свою очередь улучшает возможности по масштабируемости системы [6].

MQTT позволяет сократить до секунд частоту обновления информации, что позволяет эффективно осуществлять дистанционное зондирование и управление [6].

Протокол приносит много преимуществ в Интернет Вещей, в котором датчики и устройства используются, в том числе, для улучшения производственных процессов, протокол увеличивает эффективность распространения информации и обеспечивает три уровня качества обслуживания [6].

В тоже время у MQTT есть некоторые недостатки [6]:

- поскольку MQTT работает поверх протокола TCP

при увеличении каналов обмена сообщениями ему требуется отправка дополнительных пакетов-«рукопожатий» (handshaking), что приводит к увеличению времени установки нового канала связи;

- поскольку сервер (брокер) централизован, это влечет за собой проблемы с масштабируемостью.

III. ПРОТОКОЛ ADVANCED MESSAGE QUEUING PROTOCOL (AMQP)

A. Введение

Протокол Advanced Message Queuing Protocol (далее - AMQP) был разработан в 2003 году Джоном О'Хара (John O'Hara), работавшем в компании JPMorgan [13].

AMQP - это открытый стандартизованный протокол прикладного уровня, предназначенный для обеспечения взаимодействия между широким спектром различных приложений и систем, независимо от их внутреннего устройства. Первоначально он разрабатывался для финансового сектора с идеей предложить непатентованное решение способное управлять процессом обмена большим количеством сообщений, генерируемых в системе за короткий промежуток времени. Протокол AMQP позволяет различным платформам, реализованным на разных языках, обмениваться сообщениями, что может быть особенно полезно в гетерогенных системах [10], [13].

Протокол AMQP был реализован в двух разных вариантах: AMQP версия 0.9.1 [14] и AMQP версия 1.0 [15], каждая из которых имеет совершенно разную парадигму обмена сообщениями. Версия 1.0 в последующем была стандартизована OASIS, при этом в настоящее время старая версия протокола AMQP 0.9.1 по-прежнему популярна и используется многими брокерами сообщений [10]. В связи с этим в последующих разделах будут отдельно рассмотрены обе данные версии протокола AMQP.

B. AMQP версия 0.9.1

AMQP версии 0.9.1, также, как и MQTT использует модель публикации/подписки. Основной частью AMQP версии 0.9.1 является брокер (broker), который состоит из точек обмена (exchanges) и очередей сообщений (message queues) [15].

В отличие от MQTT, издатели (publishers) отправляют сообщение не напрямую в очередь сообщений подписчика (subscriber), а в точку обмена [15].

Точка обмена маршрутизирует входящее сообщение в соответствующую очередь сообщений, посредством сопоставления ключа привязки (binding key) очереди с ключом маршрутизации сообщения (routing key) [15].

Если в сообщении заинтересовано больше одного подписчика, сообщение дублируется и отправляется в другие очереди сообщений. Сообщение остается в очереди до тех пор, пока оно не будет получено подписчиком (рис. 2) [15].

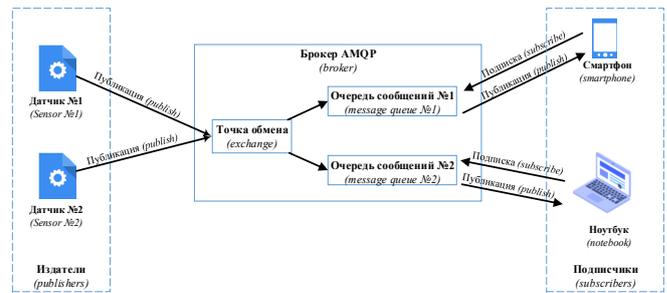


Рис. 2. Архитектура протокола AMQP версии 0.9.1

1) Очередь сообщений

Очередь сообщений хранит сообщения и распределяет их среди одного или нескольких подписчиков. Каждая очередь независима и имеет различные свойства, такие как [14]:

- имя очереди (name) – определяет имя очереди; если имя не указано сервер (брокер) самостоятельно выбирает и предоставляет его клиенту;
- эксклюзивность (exclusive) – определяет принадлежит ли очередь эксклюзивно только текущему соединению или нет; если очередь эксклюзивна - при закрытии соединения она удаляется;
- долговременность (durable) – определяет необходимость сохранения очереди сообщений при перезапуске сервера.

Обычно очереди сообщений именуются брокером, являются эксклюзивными и не долговременными. Например, эксклюзивная очередь сообщений хранит сообщения от различных издателей и отправляет их только одному подписанному клиенту [15].

2) Точка обмена

Точки обмена получают сообщения от издателей и направляют их в очереди сообщений в соответствии с заранее установленными критериями, которые называются привязками (bindings) [15].

Точки обмена не хранят сообщения, маршрутизация сообщений выполняется с помощью таблиц привязки.

Протокол определяет некоторые стандартные типы точек обмена, которых обычно достаточно для решения большинства задач. Кроме того, AMQP позволяет разработчикам создавать свои собственные экземпляры точек обмена. В дополнение к обмену сообщениями AMQP может действовать как интеллектуальный агент, который принимает сообщения и даже создает сообщения при необходимости [15].

Обычно точка обмена проверяет свойства сообщения, поле заголовка, содержимое основного текста и дополнительные данные из других источников, чтобы решить, куда оно должно быть направлено. Часто маршрутизация определяется ключевым полем, которое называется ключом маршрутизации (routing key) [15].

Ключ маршрутизации - это виртуальный адрес, который точка обмена использует для определения пункта назначения. В большинстве реализаций при маршрутизации типа «точка-точка» ключом маршрутизации является имя очереди сообщений. Для маршрутизации по темам - ключом маршрутизации является имя темы (также, как и в MQTT). Для более тонкой настройки ключ маршрутизации может быть

объединен с полями заголовка и/или содержимым сообщения [15].

3) Сообщение

Сообщение создается издателем с помощью клиентского API AMQP. После создания сообщения издатель определяет его содержимое и дополнительные свойства, а также маршрутную информацию. Далее сообщение направляется в точку обмена брокера [15].

Точка обмена анализирует и перенаправляет сообщение. Если сообщение невозможно перенаправить, точка обмена, в зависимости от установленных издателем параметров, либо удаляет его, либо возвращает издателю [15].

Одно и то же сообщение может находиться в разных очередях сообщений. Это может быть сделано путем копирования сообщения или с использованием ссылок. Скопированное сообщение полностью идентично исходному [15].

Когда сообщение поступает в очередь сообщений, брокер немедленно пытается передать сообщение подписчику. Если подписчик не готов к приему сообщения, сообщение сохраняется в очереди. В случае отсутствия подписчика брокер может вернуть сообщение обратно издателю [15].

4) Качество обслуживания

Аналогично MQTT протокол AMQP версии 0.9.1 поддерживает три уровня качества обслуживания (надежности доставки) сообщений [15]:

- «не более одного раза» («at most once») – без подтверждения доставки сообщения;
- «по крайней мере один раз» («at least once») – гарантирует, что сообщение будет доставлено подписчикам не менее одного раза;
- «только один раз» («exactly once») – гарантирует, что сообщение будет доставлено ровно один раз без дублирования.

AMQP предоставляет механизмы безопасности для защиты данных посредством использования TLS для шифрования сообщений. Для аутентификации обычно используется SASL (Simple Authentication and Security Layer) [10].

C. AMQP версия 1.0

AMQP версия 1.0 – это бинарный одноранговый (peer-to-peer) протокол для передачи сообщений между двумя клиентами [15].

Сеть AMQP версии 1.0 состоит из узлов (nodes) соединенных с помощью связей (links). Узлы также называются сущностями (entities), которые отвечают за хранение или доставку сообщений [15].

Связь можно рассматривать как однонаправленный маршрут между двумя узлами: источником (source) и получателем (target). Связь контролирует потоки исходящих сообщений источника и входящих сообщений для получателя (рис. 3) [15].

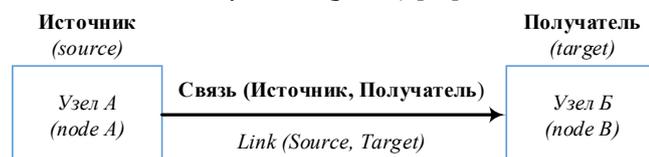


Рис. 3. Связь между узлами в протоколе AMQP версии 1.0

В протоколе AMQP версии 1.0 используются контейнеры. Каждый узел существует внутри контейнера, каждый контейнер может содержать много узлов. В качестве узлов могут быть, например, производители (producers), потребители (consumers) или очереди (queues). Очереди – это компоненты брокера, отвечающие за хранение и распространение сообщений. Производители и потребители – это компоненты клиентского приложения, генерирующие и обрабатывающие сообщения. Контейнерами могут быть брокеры или клиентские приложения (рис. 4) [15].

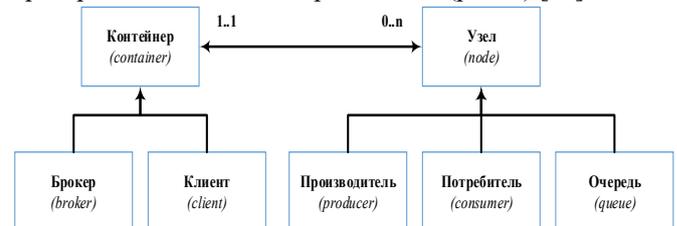


Рис. 4 Структура протокола AMQP версии 1.0

Протокол AMQP версии 1.0 не привязан к какому-либо конкретному механизму обмена сообщениями. По сравнению со старой версией 0.9.1, основанной на модели публикации/подписки с архитектурой, состоящей из точек обмена и очередей сообщений, новая версия протокола следует одноранговой парадигме и может использоваться без посредника между отправителем и получателем. Брокер используется только когда необходимо обеспечить механизм сохранения и пересылки сообщения, в остальных случаях возможен прямой обмен сообщениями [10].

Поддержка различных топологий повышает гибкость возможных решений на основе AMQP, позволяя использовать различные модели связи, такие как клиент-клиент, клиент-брокер и брокер-брокер. При этом, значительное количество инфраструктур по-прежнему используют старую версию AMQP 0.9.1 [10].

AMQP версии 1.0 работает поверх протокола TCP, и, также как MQTT, поддерживает три уровня качества обслуживания сообщений.

Протокол AMQP версии 1.0 предоставляет механизмы безопасности для защиты данных посредством использования TLS для шифрования сообщений. Для аутентификации обычно используется SASL (Simple Authentication and Security Layer) [10].

При всех своих преимуществах AMQP предъявляет относительно высокие требования к памяти и вычислительным ресурсам, что делает его довольно тяжелым протоколом, и является его самым большим недостатком при использовании в Интернете Вещей. Этот протокол лучше подходит для тех частей системы Интернета Вещей, которые не ограничены пропускной способностью и обладают большей вычислительной мощностью [10].

IV. СРАВНЕНИЕ ПРОТОКОЛОВ MQTT И AMQP

В таб. 1 приведено сравнение основных характеристик протоколов MQTT и AMQP, проведенное

на основе анализа литературы [5], [6], [9], [10], [12].

Таб. 1. Сравнение характеристик протоколов MQTT и AMQP

№	Характеристика	MQTT	AMQP
1	Год создания	1999	2003
2	Цель создания	уменьшить сетевую нагрузку	передача бизнес-сообщений между приложениями на разнородных платформах
3	Архитектура	клиент/брокер	1) клиент/брокер 2) клиент/сервер
4	Режим обмена сообщениями	асинхронный	асинхронный
5	Модель обмена сообщениями	публикация/подписка	1) публикация/подписка 2) запрос/ответ
6	Механизм управления подпиской	определение иерархической структуры тем сообщений	использование очереди и фильтрация сообщений
7	Размер заголовка сообщения	2 бит (MQTT-SN: 2 или 4 бита)	8 бит
8	Размер сообщения	до 256 МБ	не определен
9	Формат кодирования	бинарный	бинарный
10	Качество обслуживания (надежность доставки)	3 уровня	3 уровня
11	Транспортный протокол	TCP (MQTT-SN: TCP или UDP)	TCP, SCTP
12	Порт (по умолчанию)	1883/8883 (TLS/SSL)	5672, 5671 (TLS/SSL)
13	Безопасность	TLS/SSL	TLS/SSL, IPsec, SASL
14	Поддержка мобильными устройствами (Android, iOS)	да	да
15	Поддержка устройствами 6LoWPAN	да	в зависимости от реализации
16	Стандартизован	OASIS	OASIS
17	Модель лицензирования	открытая	открытая
18	Организационная поддержка	IBM, Facebook, Eurotech, Cisco, Red Hat, Software AG, Tibco, ITSO, M2Mi, Amazon Web Services (AWS), InduSoft, Fiorano и другие	Microsoft, JP Morgan, Bank of America, Barclays, Goldman Sachs, Credit Suisse и другие

В работе [11] авторы провели экспериментальное сравнение производительности протоколов Интернета Вещей для прикладного уровня, в том числе протоколов MQTT и AMQP. Авторами было проведено два эксперимента, в первом - оценивалась производительность протоколов при отсутствии сетевых сбоев при передаче данных, во втором - с учетом сетевых сбоев. Для получения усредненных значений и снижения помех при измерениях каждый эксперимент повторялся 30 раз.

В качестве метрик производительности протоколов авторы определили используемую ширину канала связи, средний размер передаваемого сообщения и процент

потерянных пакетов (при сетевых сбоях).

В первом эксперименте макет системы включал три датчика, шлюз, сервер и клиентскую рабочую станцию (рис. 5). Каждые 5 секунд датчики собирали данные об окружающей среде и через Wi-Fi-сеть отправляли их на центральный сервер. Эксперимент проводили в течение 5 минут, по результатам которого высчитывали средние значения каждой метрики.

Во втором эксперименте макет системы дополнительно для каждого датчика включал дублирующий маршрут связи (рис. 6). При возникновении ошибки при передаче сообщения приложение датчика перенаправляло данные по альтернативному маршруту. Эксперимент проводили в течение 10 минут, по результатам которого также высчитывали средние значения метрик.

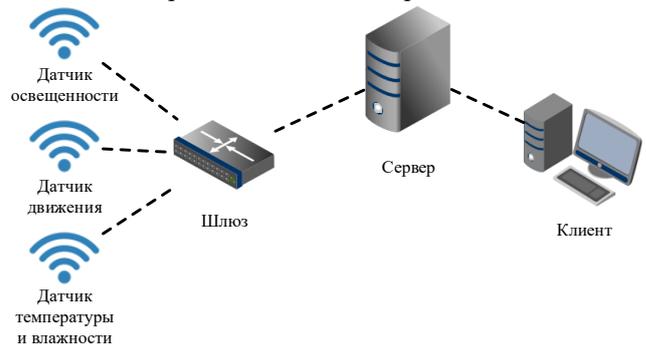


Рис. 5. Макет системы (первый эксперимент)

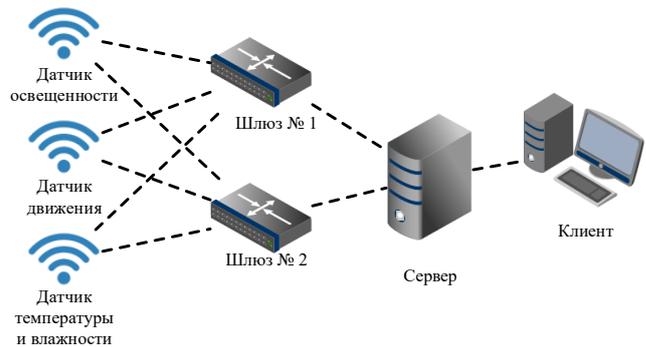


Рис. 6. Макет системы (второй эксперимент)

Время между сбоями моделировалось по формуле:

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

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

Время восстановления моделировалось по формуле:

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

где: $t_{\text{ср,длит.восст.}}$ – среднее время восстановления соединения (в рамках эксперимента принималось равным 2 секундам); rnd – случайное число от 0 до 1, полученное с использованием равномерного распределения.

В каждом эксперименте уровень качества обслуживания устанавливался таким, чтобы сообщение доставлялось по крайней мере один раз (для MQTT – 1 уровень, для AMQP – «at least once»).

Результаты первого эксперимента (рис. 7 и 8) показали, что протокол AMQP задействовал большую

ширину канала связи, чем протокол MQTT (в среднем 93,34 бит/с против 58,81 бит/с), средний размер сообщения у AMQP также был больше, чем у MQTT (в среднем 139,78 байт/пакет против 90,08 байт/пакет).

Результаты второго эксперимента показали (рис. 9, 10, 11), что протокол AMQP задействовал большую ширину канала связи, чем протокол MQTT (в среднем 100,37 бит/с против 59,48 бит/с), средний размер сообщения у AMQP составил 94,57 байт/пакет против 92,18 байт/пакет у MQTT.

У обоих протоколов используемая ширина канала увеличилась по сравнению с первым экспериментом, что связано с дополнительными повторными передачами дубликатов сообщений. Средний же размер сообщения у AMQP уменьшился, а у MQTT эквивалентен первому эксперименту. Уменьшение среднего размера сообщения у AMQP связано с тем, что AMQP передает служебные данные протокола только после восстановления соединения с сервером.

В части потери пакетов AMQP понес существенно большие потери, чем MQTT (в среднем 45,28% против 0,48%), несмотря на уменьшение размера пакета AMQP при повторных передачах.

Результаты проведенных экспериментов показывают, что протокол MQTT показывает лучшие результаты,

условиях ограниченности доступных ресурсов и нестабильности сети.

Однако авторы работы [11] отмечают, что они не проводили детальную настройку протоколов, все параметры протоколов, кроме уровня качества обслуживания, имели значения по умолчанию. При более тщательной настройке протоколов результаты производительности могут отличаться.

V. ЗАКЛЮЧЕНИЕ

В данной статье представлен краткий обзор характерных особенностей протоколов обмена сообщениями прикладного уровня: AMQP и MQTT, используемых в программных приложениях для устройств Интернета Вещей.

Рассмотрены преимущества и недостатки данных протоколов, и приведены результаты экспериментальных замеров их производительности. Результаты проведенных экспериментов показали, что при настройках по умолчанию протокол MQTT по сравнению с AMQP показывает лучшие результаты.

Настоящая статья будет полезна разработчикам при выборе оптимального протокола при разработке программ для устройств Интернета Вещей.

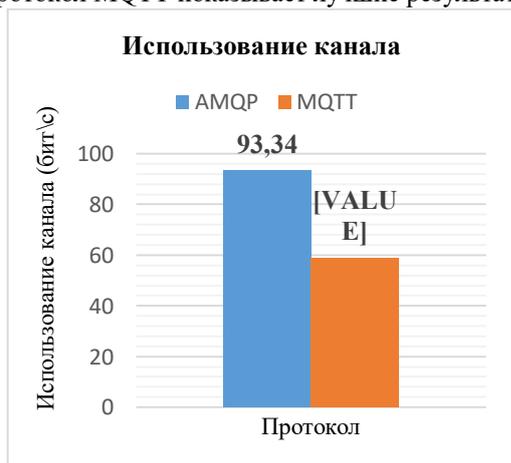


Рис. 7. Использование канала (первый эксперимент)

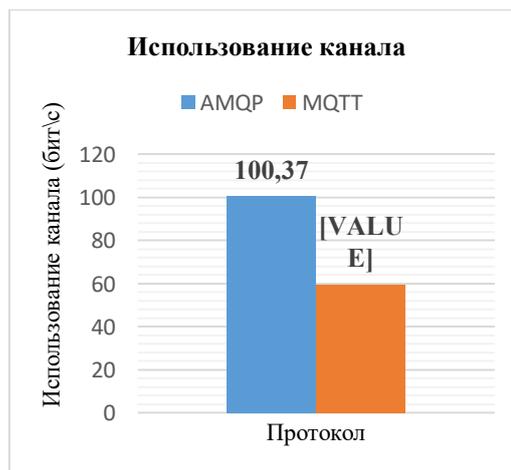


Рис. 9. Использование канала (второй эксперимент)

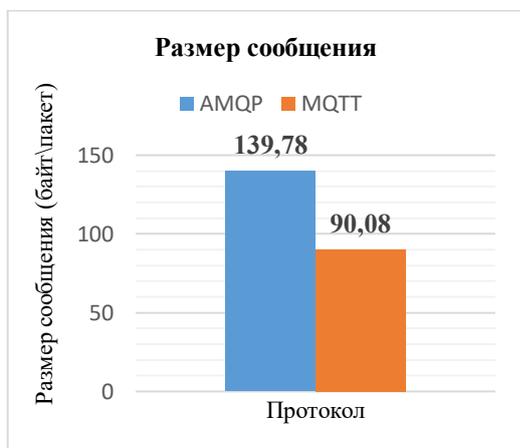


Рис. 8. Размер сообщения (первый эксперимент)

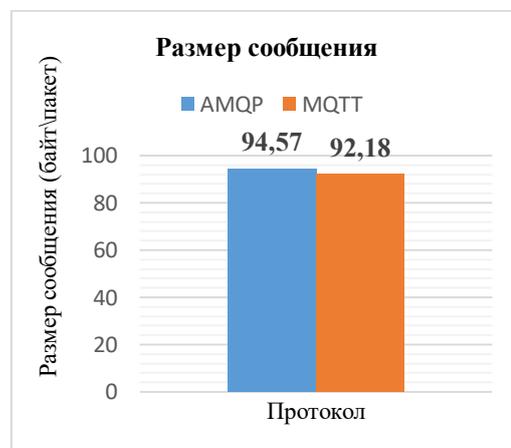


Рис. 10. Размер сообщения (второй эксперимент)

чем AMQP, в части использования пропускной способности сети, размеров передаваемых сообщений и потери пакетов, и представляется приемлемым выбором для использования в приложениях Интернета Вещей в

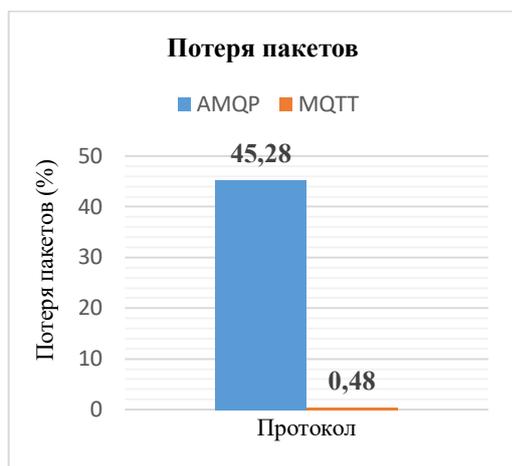


Рис. 11 Потеря пакетов (второй эксперимент)

БЛАГОДАРНОСТИ

Статья подготовлена в рамках курса по протоколам и стандартам Интернета вещей магистерской программы ПОВС [17]. Хотелось бы также отметить тех, кто способствовал появлению этого курса и учебной программы. Это, прежде всего, проф. Шнепс-Шнеппе М.А. и его работы по стандартизации [18, 19, 20], а также В.П. Куприяновский, развивший транспортное направление в системах IoT [21, 22].

БИБЛИОГРАФИЯ

- [1] "MQTT: The Standard for IoT Messaging. FAQ." <https://mqtt.org/faq/>. Retrieved: May, 2022.
- [2] V. Lampkin, W.T. Leong, L. Olivera, S. Rawat, N. Subrahmanyam, R. Xiang. "Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry. First Edition (September 2012)." IBM Redbooks. International Technical Support Organization (IBM).
- [3] A. Banks, E. Briggs, K. Borgendale, R. Gupta. "MQTT Version 5.0." 07 March 2019. OASIS Standard.
- [4] A. Stanford-Clark, H. Linh Truong. "MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2." 14 November 2013. IBM.
- [5] C. Bayılmış, M.A. Ebleme, Ü. Çavuşoğlu, K. Küçük, A. Sevin. "A survey on communication protocols and performance evaluations for Internet of Things." Digital Communications and Networks, 2022.
- [6] N. Bibi, F. Iqbal, S.M. Akhtar, R. Anwar, S. Bibi. "A Survey of Application Layer Protocols of Internet of Things." IJCSNS International Journal of Computer Science and Network Security, Vol. 21 No.11, November 2021.
- [7] "MQTT QoS: Understanding Quality of Service." <https://assetwolf.com/learn/mqtt-qos-understanding-quality-of-service>. Retrieved: May, 2022.
- [8] J.E. Luzuriaga, M. Perez, P. Boronat, J.C. Cano, C. Calafate, P. Manzoni. "Handling Mobility in IoT applications based on the MQTT protocol (Author Version)." Sixth International Conference on the Internet Technologies & Applications (ITA 2015). IEEE.
- [9] N. Naik. "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP." Computer Science. 2017 IEEE International Systems Engineering Symposium (ISSE).
- [10] J. Dizdarevic, F. Carpio, A. Jukan, X. Masip-Bruin. "A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration." ACM Computing Surveys, Vol. 1, No. 1. February 2019.
- [11] T. Moraes, B. Nogueira, V. Lira, E. Tavares. "Performance Comparison of IoT Communication Protocols." 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). October 2019.
- [12] A. Foster. "Messaging Technologies for the Industrial Internet and the Internet of Things Whitepaper." PrismTech Corp. 2014.
- [13] J. O'Hara. "Toward a Commodity Enterprise Middleware." ACM Queue, Vol. 5, Issue 4. May-June 2007, pp. 48–55.
- [14] "AMQP. Advanced Message Queuing Protocol. Protocol Specification Version 0-9-1." 13 November 2008.
- [15] F. Walcher, W. Kastner. "KNX to MQTT/AMQP." Computer Science. 2019.

- [16] "OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0." 29 October 2012. OASIS Standard.
- [17] Namiot, Dmitry, Manfred Sneps-Sneppe Ventspils, and Yousef Ibrahim Daradkeh. "On internet of things education." 2017 20th conference of open innovations association (FRUCT). IEEE, 2017.
- [18] Namiot, Dmitry, and Manfred Sneps-Sneppe. "On m2m software." International Journal of Open Information Technologies 2.6 (2014): 29-36.
- [19] Sneps-Sneppe, Manfred, and Dmitry Namiot. "About M2M standards and their possible extensions." 2012 2nd Baltic Congress on Future Internet Communications. IEEE, 2012.
- [20] Волков, А. А. О задачах создания эффективной инфраструктуры среды обитания / А. А. Волков, Д. Е. Намиот, М. А. Шнепс-Шнеппе // International Journal of Open Information Technologies. – 2013. – Т. 1. – № 7. – С. 1-10.
- [21] Куприяновская Ю. В. и др. Умный контейнер, умный порт, BIM, Интернет Вещей и блокчейн в цифровой системе мировой торговли //International Journal of Open Information Technologies. – 2018. – Т. 6. – №. 3. – С. 49-94.
- [22] Соколов И. А. и др. Прорывные инновационные технологии для инфраструктур. Евразийская цифровая железная дорога как основа логистического коридора нового Шелкового пути //International Journal of Open Information Technologies. – 2017. – Т. 5. – №. 9. – С. 102-118.

Overview and comparison of protocols Internet of Things: MQTT and AMQP

Dmitry Yakupov

Abstract — The development of software applications for Internet of Things (IoT) devices involves solving many difficult tasks, one of which is the choice of a suitable protocol for information exchange. Internet of Things devices have a number of features, such as low performance, small memory, short battery life, etc., which in turn imposes appropriate restrictions on the choice of communication protocol. This article provides an overview and comparison of two popular application layer protocols for the Internet of Things: Advanced Message Queuing Protocol (AMQP) and Message Queuing Telemetry Transport (MQTT). These protocols are standardized by the Organization for the Advancement of Structured Information Standards (OASIS) consortium and have an open licensing model. For messaging, the MQTT and AMQP protocols use the "publish/subscribe" model, and as a transport for communication between devices, they support the use of the TCP protocol. In addition, the MQTT and AMQP protocols support various levels of message quality of service, allowing you to manage the reliability of their delivery. The article discusses the key features of the MQTT and AMQP protocols, their architecture, advantages and disadvantages. In conclusion, a comparative analysis of the protocols and the results of an experimental comparison of their performance are presented.

Keywords— Internet of Things (IoT), Advanced Message Queuing Protocol (AMQP), Message Queuing Telemetry Transport (MQTT).

REFERENCES

- [1] "MQTT: The Standard for IoT Messaging. FAQ." <https://mqtt.org/faq/>. Retrieved: May, 2022.
- [2] V. Lampkin, W.T. Leong, L. Olivera, S. Rawat, N. Subrahmanyam, R. Xiang. "Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry. First Edition (September 2012)." IBM Redbooks. International Technical Support Organization (IBM).
- [3] A. Banks, E. Briggs, K. Borgendale, R. Gupta. "MQTT Version 5.0." 07 March 2019. OASIS Standard.
- [4] A. Stanford-Clark, H. Linh Truong. "MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2." 14 November 2013. IBM.
- [5] C. Bayılmış, M.A. Ebleme, Ü. Çavuşoğlu, K. Küçük, A. Sevin. "A survey on communication protocols and performance evaluations for Internet of Things." Digital Communications and Networks, 2022.
- [6] N. Bibi, F. Iqbal, S.M. Akhtar, R. Anwar, S. Bibi. "A Survey of Application Layer Protocols of Internet of Things." IJCSNS International Journal of Computer Science and Network Security, Vol. 21 No.11, November 2021.
- [7] "MQTT QoS: Understanding Quality of Service." <https://assetwolf.com/learn/mqtt-qos-understanding-quality-of-service>. Retrieved: May, 2022.
- [8] J.E. Luzuriaga, M. Perez, P. Boronat, J.C. Cano, C. Calafate, P. Manzoni. "Handling Mobility in IoT applications based on the MQTT protocol (Author Version)." Sixth International Conference on Internet Technologies & Applications (ITA 2015). IEEE.
- [9] N. Naik. "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP." Computer Science. 2017 IEEE International Systems Engineering Symposium (ISSE).
- [10] J. Dizdarevic, F. Carpio, A. Jukan, X. Masip-Bruin. "A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration." ACM Computing Surveys, Vol. 1, No. 1. February 2019.
- [11] T. Moraes, B. Nogueira, V. Lira, E. Tavares. "Performance Comparison of IoT Communication Protocols." 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). October 2019.
- [12] A. Foster. "Messaging Technologies for the Industrial Internet and the Internet of Things Whitepaper." PrismTech Corp. 2014.
- [13] J. O'Hara. "Toward a Commodity Enterprise Middleware." ACM Queue, Vol. 5, Issue 4. May-June 2007, pp. 48–55.
- [14] "AMQP. Advanced Message Queuing Protocol. Protocol Specification. Version 0-9-1." 13 November 2008.
- [15] F. Walcher, W. Kastner. "KNX to MQTT/AMQP." Computer Science. 2019.
- [16] "OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0." 29 October 2012. OASIS Standard.
- [17] Namiot, Dmitry, Manfred Sneps-Sneppe Ventspils, and Yousef Ibrahim Daradkeh. "On internet of things education." 2017 20th conference of open innovations association (FRUCT). IEEE, 2017.
- [18] Namiot, Dmitry, and Manfred Sneps-Sneppe. "On m2m software." International Journal of Open Information Technologies 2.6 (2014): 29-36.
- [19] Sneps-Sneppe, Manfred, and Dmitry Namiot. "About M2M standards and their possible extensions." 2012 2nd Baltic Congress on Future Internet Communications. IEEE, 2012.
- [20] Volkov, A. A. O zadachah sozdaniya jeffektivnoj infrastruktury sredy obitaniya / A. A. Volkov, D. E. Namiot, M. A. Shneps-Shneppe // International Journal of Open Information Technologies. – 2013. – T. 1. – # 7. – S. 1-10.
- [21] Kuprijanovskaja Ju. V. i dr. Umnyj kontejner, umnyj port, BIM, Internet Veshhej i blokchejn v cifrovoj sisteme mirovoj trgovli //International Journal of Open Information Technologies. – 2018. – T. 6. – #. 3. – S. 49-94.
- [22] Sokolov I. A. i dr. Proryvnye innovacionnye tehnologii dlja infrastruktur. Evrazijskaja cifrovaja zheleznaia doroga kak osnova logisticheskogo koridora novogo Shelkovogo puti //International Journal of Open Information Technologies. – 2017. – T. 5. – #. 9. – S. 102-118.