

УДК 004.021:004.75

## ОЦЕНКА КАЧЕСТВА ПЕРЕДАЧИ ИНФОРМАЦИИ В СИСТЕМЕ ДИСПЕТЧЕРИЗАЦИИ НА БАЗЕ MQTT-АРХИТЕКТУРЫ



**В.Ф. Алексеев**  
доцент кафедры проектирования информационно-компьютерных систем БГУИР, кандидат технических наук, доцент



**Д.В. Лихачевский**  
декан факультета компьютерного проектирования БГУИР, кандидат технических наук, доцент



**Г.А. Пискун**  
доцент кафедры проектирования информационно-компьютерных систем БГУИР, кандидат технических наук, доцент



**И.В. Андриялович**  
заместитель декана факультета компьютерного проектирования БГУИР, соискатель

### **В.Ф. Алексеев**

Окончил Минский радиотехнический институт. Область научных интересов связана с разработкой методов и алгоритмов построения информационно-компьютерных систем, организацией учебного и научно-исследовательского процессов в техническом университете.

### **Д.В. Лихачевский**

Окончил Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов связана с исследованием проблем радиочастотной идентификации объектов, организацией учебного и научно-исследовательского процессов в техническом университете.

### **Г.А. Пискун**

Окончил Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов связана с разработкой методов и алгоритмов построения информационно-компьютерных систем, организацией учебного и научно-исследовательского процессов в техническом университете.

**Аннотация.** Выполнен анализ методов и алгоритмов оценки качества передачи информации, обеспечивающие качество бесперебойной передачи данных в системе диспетчеризации, построенной на базе MQTT-архитектуры.

Показано, что могут быть разработаны тесты для оценки работоспособности системы диспетчеризации на базе MQTT-архитектуры в различных условиях качества связи. Проведена оценка эффективности алгоритмов передачи информации в системе диспетчеризации на базе MQTT-архитектуры с различным уровнем качества обслуживания.

**Ключевые слова:** MQTT-архитектуры, оценка качества передачи информации, диспетчеризация, встроенные системы.

### **Введение.**

В основном процесс передачи информации осуществляется через Internet. Для соединения с сервером используются различные способы передачи информации. Важным критерием в этом является выбор сетевого протокола передачи данных. Одним из которых является MQTT протокол, который преимущественно используется во встраиваемых системах. Преимуществами данного протокола является то, что качество и скорость соединения не имеет особого значения

Одной из главных задач, выполняемых устройствами телемеханики и диспетчеризации, является бесперебойное отслеживания состояния объекта по различным показателям. Поэтому создание и внедрение новых систем диспетчеризации и телемеханики является актуальной научной

задачей. Использование достижений современной системотехники позволяет обеспечить более эффективное и безопасное управление процессом, а также достичь необходимой адаптивности управления в условиях изменения показателей, по которым необходимо производить отслеживание, а также изменение объекта отслеживания.

Надежное функционирование таких систем возможно при наличии достоверной и своевременной информации о состоянии устройств на объекте, телемеханики и связи.

Большое значение в этой ситуации приобретают вопросы, связанные с внедрением устройств и новых методов получения и обработки информации.

#### **Актуальность.**

Оценка качества передачи информации является одним из наиболее перспективных и актуальных направлений исследования для современных систем телемеханики и диспетчеризации. Это обусловлено тем, что при передаче данных стали предъявляться высокие требования к качеству бесперебойной передачи данных (банковская сфера, медицина, сельское хозяйство, производство и т.д.).

Рядом специалистов [1–6] выполнен анализ методов и алгоритмов качества передачи информации в системе диспетчеризации и телемеханики на базе MQTT-архитектуры.

Одним из недостатков подобных систем является плохое качество передачи информации, а также потеря необходимых данных во время передачи информации по сети.

#### **Анализ принципов работы протокола MQTT.**

*MQTT (MQ Telemetry Transport)* – протокол обмена сообщениями, который предоставляет сетевым клиентам с ограниченными ресурсами простой способ распространения телеметрической информации. Протокол, который использует шаблон связи «публикация/подписка», используется для связи между устройствами и играет важную роль в Интернете вещей.

Кроме того, поверх уровня *TCP* стоит уровень стандартной безопасности *TLS (Transport Layer Security)*, ранее известный как *SSL (Secure Sockets Layer)*. Порт 8883 обеспечивает безопасность связи, если адрес брокера работает с этим портом, то трафик передаётся с шифрованием.

Это протокол, разработанный конкретно для *IoT*. Пример *IoT* экосистемы с использованием протокола MQTT представлен на рисунке 1 [7].

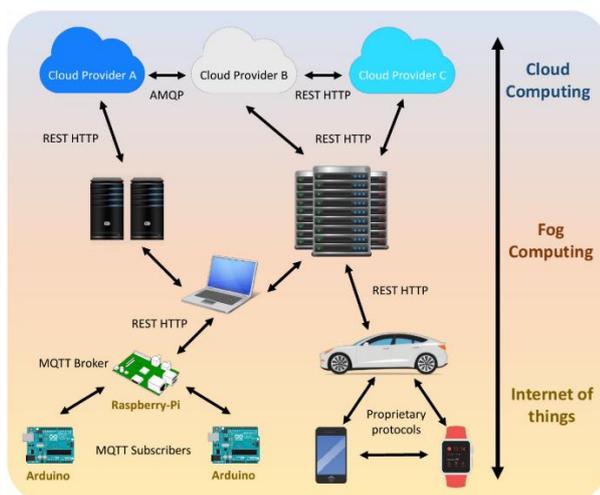


Рисунок 1. Пример *IoT* экосистемы с использованием протокола *MQTT*

Открытый и простой он предназначен для обмена информацией между разными устройствами и модулями. Упрощает соединение каналов связи быстро, качественно и своевременно. Отвечает за безопасность соединения, скорость передачи данных и практическое функционирование систем и программ. Защищает от всевозможных сбоев и неполадок, качественно выполняя свою работу.

MQTT позволяет устройствам интернета вещей с ограниченными ре-сурсами отправлять или публиковать информацию по заданной теме на сервер, который функционирует как посредник сообщений MQTT. Затем брокер передает информацию тем клиентам, которые ранее подписались на тему клиента. Для человека тема выглядит как иерархический путь к файлу. Клиенты могут

подписать на определенный уровень иерархии темы или использовать подстановочный знак для подписки на несколько уровней.

Протокол MQTT является хорошим выбором для беспроводных сетей, которые испытывают различные уровни задержки из-за случайных ограничений полосы пропускания или ненадежных соединений. В случае разрыва соединения подписывающего клиента с брокером брокер буферизует сообщения и отправляет их подписчику, когда он снова подключается. Если соединение между клиентом публикации и посредником будет отключено без предварительного уведомления, посредник может закрыть соединение и отправить подписчикам кэшированное сообщение с инструкциями от издателя.

#### **Сравнение протоколов взаимодействия во встраиваемых системах.**

Протокол MQTT – простой протокол обмена сообщениями, реализующий модель «публикации/подписки» (*publish/subscribe*) и предназначенный для связи компьютеризированных устройств, подключённых к локальной или глобальной сети, между собой и различными публичными или приватными веб-сервисами.

Протокол создавался, чтобы обеспечить открытость, простоту, минимальные требования к ресурсам и удобство внедрения.

В сети на базе протокола MQTT различают 3 объекта:

– издатель (*Publisher*) – MQTT-клиент, который при возникновении определенного события передает брокеру информацию о нём, публикуя соответствующие топики;

– брокер (*Broker*) – MQTT-сервер, который принимает информацию от издателей и передает ее соответствующим подписчикам, в сложных системах может выполнять также различные операции, связанные с анализом и обработкой поступивших данных. Разные брокеры могут соединяться между собой, если они подписываются на сообщения друг друга;

– подписчик (*Subscriber*) – MQTT-клиент, который после подписки к брокеру большую часть времени «слушает» его и постоянно готов к приему и обработке входящего сообщения на интересующие топики от брокера.

Протокол *CoAP* (*Constrained Application Protocol*) – протокол, разработанный Инженерным советом Интернета (*IETF, Internet Engineering Task Force*) и описан в документе *RFC 7252*. Протокол работает на прикладном уровне, и предназначен для передачи данных по линиям с ограниченной пропускной способностью. *CoAP* был разработан на основе протокола *HTTP*, представляет собой двоичную его версию, но не является слепым его сжатием. *CoAP* состоит из подмножества *HTTP* функциональных возможностей, которые были вновь разработаны с учетом низкой мощности и малого потребления энергии ограниченных встраиваемых устройств, например, такие как датчик уровня пыли в помещении. Кроме того, были изменены различные механизмы и добавлены некоторые новые возможности, чтобы протокол подходил для Интернета Вещей.

Так, в отличие от протокола *HTTP*, который является текстовым и использует *TCP*, *CoAP* – это бинарный протокол, который транспортируется через *UDP*, что уменьшает размер его служебных данных и повышает гибкость в моделях связи. *CoAP* организован в два слоя: слой транзакций и слой «*Request/Response*».

#### **Тестирование системы телемеханики и диспетчеризации с использованием различных уровней качества обслуживания.**

Для того, чтобы количественно оценить объем передаваемых данных при использовании протокола MQTT с различным параметром *QoS*, были проанализированы транзакции клиент-сервера и количество передаваемых байтов. В таблице 1 содержится информация о количестве байтов и пакетов, передаваемых за одну транзакцию. Транзакция начинается, когда клиент отправляет данные, и заканчивается, когда сервер получает данные или, в некоторых случаях, при получении клиентом подтверждения.

Таблица 1. Количество байт и пакетов, передаваемых за одну транзакцию

<i>QoS</i>	<i>MQTT QoS0</i>	<i>MQTT QoS1</i>	<i>MQTT QoS2</i>
Количество байт	75	135	255
Количество пакетов	1	2	4

Сообщение делится на две части: полезную информацию и служебную. Эти части влияют на затраты ресурса каналов и энергии батарей питания. Для улучшения эффективности требуется снижение служебной информации. В таблице 2 показано отношение служебной информации к полезной в процентах при передаче одного сообщения.

Таблица 2. Отношение полезной информации к служебной

<i>QoS</i>	<i>MQTT QoS0</i>	<i>MQTT QoS1</i>	<i>MQTT QoS2</i>
Полезная информация, %	16,8	16,5	16,5
Служебная информация, %	83,2	85,5	85,5

В *MQTT* с *QoS0* служебные поля в пакете занимают небольшой объем, поэтому при сеансе связи тратится малое количество энергии.

На рисунке 2 представлены результаты исследования величины задержки при передаче сообщений.

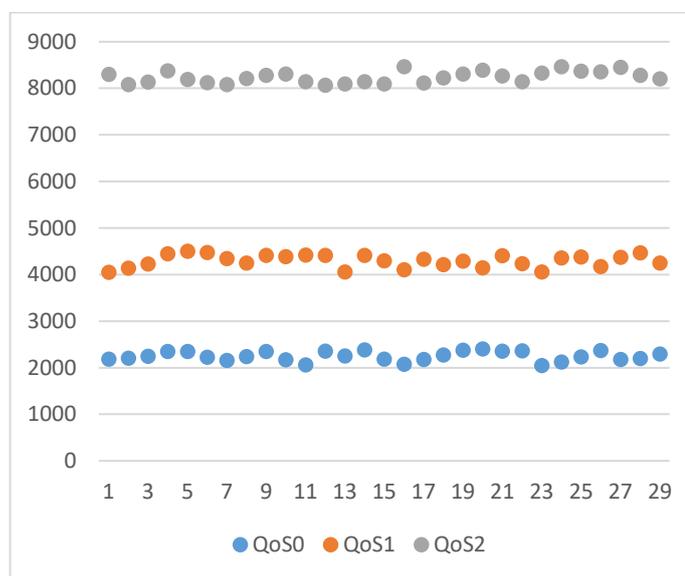


Рисунок 2. Величина задержки в миллисекундах для каждого сообщения

В таблице 3 представлен процент потерянных сообщений. Анализ данных показывает, что несмотря на то, что при использовании *QoS1* и *QoS2* значительно увеличивается задержка при отправке сообщений, уменьшается процент потерянных сообщений. Если не учитывать этот факт, можно понести значительные потери пакетов, что может быть критично в некоторых системах. В других системах, где не так важно каждое сообщение можно получить значительное увеличение времени на передачу сообщений, что может быть критично при большом количестве сообщений маленького размера

Таблица 3. Процент потерянных сообщений

Размер сообщения, <i>Byte</i>	Потеряно при <i>QoS0</i> , %	Потеряно при <i>QoS1</i> , %	Потеряно при <i>QoS2</i> , %
150	3,00	1,00	0,00
200	5,00	1,30	0,04
250	6,00	1,50	0,05
300	6,70	1,80	0,05

### Принципы достижения высокого качества связи при использовании протокола передачи данных MQTT.

Можно предложить набор принципов, которые помогают оптимизировать использование пропускной способности и данных:

1. Выбор правильного *QoS*. Одной из ключевых функций, предлагаемых MQTT, является качество обслуживания (*QoS*). Сообщения *QoS0* являются самыми простыми, их также называют сообщениями сработал и забыл. Эти сообщения не имеют подтверждения от брокера (но все еще есть подтверждение от уровня TCP) и, следовательно, не имеют гарантированной доставки. Сообщения типа *QoS1* имеют гарантию того, что они будут доставлены, хотя возможно, что они могут быть доставлены несколько раз. Сообщения *QoS1* включают два уровня обмена данными на уровне приложений. Сообщения типа *QoS2* имеют гарантированную доставку ровно один раз. Сообщения типа *QoS2* имеют максимальные накладные расходы.

Учитывая тот факт, что уровень *QoS* и служебные данные обратно пропорциональны, стратегия выбора уровня *QoS* для ваших сообщений довольно проста. Все высокочастотные данные (обычно данные в реальном времени) могут быть отправлены с использованием *QoS0*, поскольку потеря нескольких пакетов данных не может быть критической. Следует использовать *QoS1* для сообщений, которые требуют гарантированной доставки, в основном это события, команды.

При разработке любого приложения важно объективно понимать стоимость выбора *QoS*. В качестве примера были измерены данные, потребляемые сообщением, на разных уровнях *QoS*. Сообщение «HelloWorld» было опубликовано в теме «test\_test» с тремя различными уровнями *QoS*, а данные, которыми обменивались, были захвачены с помощью Wireshark (таблица 4).

Потребление данных снижается примерно до 50% при использовании *QoS1* в сравнении с *QoS2*. Точно так же *QoS0* использует на 40% меньше данных, чем *QoS1*.

Следовательно, правильно будет использовать *QoS0* для периодической информации, которую необходимо отправлять на сервер. Не следует использовать *QoS1* для команд с сервера, потому что существует возможность избыточной доставки команд. Хотя *QoS2* выглядит очевидным решением для команд с сервера, это не экономичный выбор.

Таблица 4. Количество байт, потребляемые сообщением, на разных уровнях *QoS*

<i>QoS</i>	MQTT <i>QoS0</i>	MQTT <i>QoS1</i>	MQTT <i>QoS2</i>
Количество байт	87	126	241

2. Минимизация сообщений *QoS2*. Накладные расходы сообщений *QoS2* сопоставимы с HTTP. У них на 50% больше накладных расходов, чем у сообщений *QoS1*. Сообщения *QoS1* можно использовать для замены сообщений *QoS2* в большинстве приложений. Проблема с сообщениями *QoS1* в том, что они могут доставляться несколько раз. Есть два способа избежать этого.

Первое решение применяется, когда вы контролируете реализацию клиентской библиотеки MQTT. В каждом пакете публикации MQTT есть поле «Идентификатор пакета». Это поле обычно увеличивается для каждого нового пакета, опубликованного брокером для клиента. Если вы получите такое же сообщение (в случае *QoS0*), то поле «Идентификатор пакета» останется прежним. Клиент может поддерживать список «идентификаторов пакетов», полученных в последних нескольких сообщениях (~10), и использовать его для определения, является ли новое полученное сообщение дубликатом.

Если у вас нет контроля над реализацией клиентской библиотеки MQTT, вы можете создать механизм виртуального идентификатора пакета (VPI). Полезная нагрузка каждого сообщения MQTT может быть настроена так, чтобы содержать VPI, этот VPI может увеличиваться отправителем сообщения всякий раз, когда публикуется новое сообщение. Клиент может вести список нескольких последних полученных VPI и использовать его для фильтрации повторяющихся сообщений.

3. Тщательный выбор названия тем. Поле заголовка содержит название темы в формате UTF8. Это означает, что длинное название темы составляет большую часть сообщения.

В том случае, если необходимо опубликовать (*QoS0*) информацию размером 20 байт по теме. Размер сообщения публикации будет включать в себя размер сообщения, размер заголовка, размер темы. Таким образом, размер темы занимают значительную часть сообщения. Следует выбирать стратегию именования так, чтобы в них содержалась только важная информация.

### **Заключение.**

Использование данных и пропускной способности является основным ограничением при проектировании периферийных устройств *IoT*. В отличие от сотовых данных для обычного использования сотовые данные корпоративного уровня с длительным временем безотказной работы обычно дороги. Следовательно, ограничения на потребление данных становятся более актуальными, если периферийное устройство полагается на сотовые данные. Неправильный дизайн приложений может перестать использовать преимущество низкого потребления данных, которое *MQTT* предлагает для систем *IoT*.

### **Список литературы**

- [1] Росляков, А.В. Интернет вещей: учебное пособие / А.В. Росляков, С.В. Ваняшин, А.Ю. Гребешков – Самара: ПГУТИ, 2015.— 200 с.
- [2] Марц, Натан Большие данные. Принципы и практика построения масштабируемых систем обработки данных в реальном времени: моногр. / Натан Марц , Джеймс Уоррен. – М.: Вильямс, 2016. – 368 с.
- [3] Тейлор, Джеймс Получение конкурентных преимуществ путем автоматизации принятия скрытых решений / Джеймс Тейлор , Нил Рэйдэн. – М.: Символ-плюс, 2009. – 448 с.
- [4] Фуругян, Меран Алгоритмы планирования вычислений и синтеза систем реального времени / Меран Фуругян. – М.: LAP Lambert Academic Publishing, 2012. – 232 с.
- [5] Хетагуров, Я. А. Проектирование автоматизированных систем обработки информации и управления (АСОИУ). Учебник / Я.А. Хетагуров. – М.: Бином. Лаборатория знаний, 2015. – 240 с.
- [6] Протокол MQTT. Особенности, варианты применения, основные процедуры MQTT Protocol: [Электронный ресурс]. URL: <http://tssonline.ru/articles2/fix-corp/protokol-mqtt-osobennosti-varianty-primeneniya-osnovnye-protsedury-mqtt-protocol>. (Дата обращения: 21.03.2022).
- [7] IoT, туман и облака: поговорим про технологии? [Электронный ресурс]. URL: <https://3-info.ru/post/2814> (Дата обращения: 21.03.2022).

## **QUALITY EVALUATION OF INFORMATION TRANSFER IN A DISPATCHING SYSTEM BASED ON MQTT ARCHITECTURE**

**V.F. ALEKSEEV**  
Associate Professor,  
Department of  
Information Computer  
Systems Design, PhD of  
Technical sciences,  
Associate Professor

**D.V. LIKHACHEVSKY**  
Dean of the Faculty of  
Computer Design of  
BSUIR,  
PhD of Technical  
Sciences, Associate  
Professor

**G.A. PISKUN**  
Associate Professor of  
the Department of  
Design of Information  
and Computer Systems  
of BSUIR, PhD of  
Technical Sciences,  
Associate Professor

**I.V. ANDRYALOVICH**  
Deputy Dean of the  
Faculty of Computer  
Design of BSUIR,  
postgraduate student of  
the Department of IP&E

*Department of Information and Computer Systems Design  
Faculty of Computer Engineering  
Belarusian State University of computer science and Radio Electronics, Republic of Belarus  
E-mail: alexvikt.minsk@gmail.com*

**Abstract.** The analysis of methods and algorithms for assessing the quality of information transfer, which ensure the quality of uninterrupted data transfer in a dispatch system built on the basis of the MQTT architecture, is carried out.

It is shown that tests can be developed to evaluate the operability of a dispatch system based on the MQTT architecture under various conditions of communication quality. The effectiveness of information transfer algorithms in a dispatch system based on the MQTT architecture with different levels of quality of service has been evaluated.

**Keywords:** MQTT architectures, information transfer quality assessment, scheduling, embedded systems.