Математические структуры и моделирование 2023. № 4 (68). С. 100–110

УДК 004.75
DOI 10.24147/2222-8772.2023.4.100-110

ОБ ЭФФЕКТИВНОСТИ АГРЕГАЦИИ ДАННЫХ В СЕТЯХ ИНТЕРНЕТА ВЕЩЕЙ С ИСПОЛЬЗОВАНИЕМ ПРОТОКОЛА MQTT

Т.В. Костеннов
аспирант, e-mail: timofey.kostennov@gmail.com

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. Высокие темпы развития отрасли интернета вещей и промышленного интернета вещей приводят к увеличению количества устройств и передаваемых данных в проектируемых и используемых системах. В данной работе предложен подход, направленный на уменьшение использования сетевых ресурсов, основанный на агрегации данных на шлюзах-агрегаторах. Проведены эксперименты, позволяющие оценить предложенные подходы по параметру использования сетевых ресурсов.

Ключевые слова: IoT, MQTT, интернет вещей, агрегация данных.

Введение

С общим развитием технологий развивается и сфера интернета вещей (Internet of Things). Этот термин зачастую используют для описания систем, концепция которых основана на связи между некими физическими объектами, обладающими возможностью получать данные о внешней среде или как-либо влиять на неё. Начавшись как идея всеобъемлющего внедрения радиочастотных меток, сегодня интернет вещей представлен различными устройствами и протоколами [6].

Технологии взаимодействия между устройствами в рамках сетей интернета вещей и промышленного интернета вещей обычно обозначаются термином «M2M» (Machine-to-Machine, машинное взаимодействие). Концепция M2M предполагает отсутствие или минимальное участие человека. Быстрый рост сферы интернета вещей и применение технологий и протоколов в промышленности создают новые вызовы и проблемы для разработчиков подобных систем. По данным аналитических отчетов, в ближайшие годы индустрия интернета вещей продолжит свое развитие. Однако использование традиционных подходов к проектированию может привести к возникновению проблем при улучшении или расширении существующих систем. Одной из множества существующих проблем является ограниченная пропускная способность телекоммуникационных сетей, используемых системами интернета вещей.

Существует несколько решений, позволяющих тем или иным способом меньше использовать сетевые ресурсы. Одним из них является метод агрегации данных

при использовании протоколов прикладного уровня. Суть метода заключается в агрегации данных от конечных устройств на узлах передачи и пересылке к следующему устройству единым сообщением вместо многочисленных пересылок единичного объёма данных от каждого устройства. Предлагается подробнее рассмотреть несколько реализаций данного решения и оценить их эффективность с помощью экспериментов.

В сетях интернета вещей широкое распространение получили различные протоколы прикладного уровня, такие как MQTT, CoAP, AMQP. В экспериментах данной статьи будет рассмотрен протокол MQTT как протокол со средними требованиями к пропускной способности сети и средним размером сообщения [2].

1. О размере накладных расходов на пересылку сообщения

Общий размер сообщения зависит от всего стека технологий, применяемого для его пересылки. Для передачи информации современные информационные сети наиболее часто используют стек протоколов IP/TCP. Пакеты данных протоколов инкапсулируются в поле полезной нагрузки Ethernet. Рассмотрим подробнее каждый протокол, чтобы подсчитать средний размер итогового сообщения.

2. Ethernet

Ethernet – это протокол, работающий на канальном уровне согласно сетевой модели OSI. Использует в качестве метода управления доступом множественный доступ с контролем несущей и обнаружением коллизий (CSMA/CD, Carrier Sense Multiple Access with Collision Detection). Большую часть времени сетевые устройства, подключённые к сети, находятся в режиме ожидания передачи данных по каналу связи. Когда канал свободен, то отсутствует постоянная составляющая («несущая»). В таком состоянии любое сетевое устройство, подготовившее кадр для передачи, может начать его передачу. Переданный кадр поступает на приёмники остальных устройств, подключённых к сети. Сетевое устройство распознает свой адрес, копирует этот кадр в свой приемный буфер. После успешной передачи кадра все сетевые устройства, подключённые к шине, включая устройство-отправитель, должны выдержать межпакетную паузу IPG (Inter Packet Gap), равную времени передачи 96 бит (12 байт) данных. Если же во время передачи кадра возникла коллизия, то передача должна быть немедленно прекращена. Коллизия происходит в том случае, когда информация о том, что рабочая станция начала передачу, не достигла других станций в сети. Стандарт IEEE 802.3 определяет структуру единичного кадра протокола Ethernet со скоростью 10,100 Мбит/с следующим образом (см. табл. 1) [4].

Минимальный размер полезной нагрузки определяется требованием к минимальному размеру информационного кадра. Информационный кадр составляют поля с 3 по 8. Для стандартов Ethernet (10 Mbps) и Fast Ethernet (100 Mbps) он составляет 64 байта.

Следовательно, кадр содержит в себе от 42 (в случае, если тег 802.1Q присутствует) до 46 (в случае его отсутствия) байт полезных данных для сетевого уровня. Однако на их передачу физическая линия задействуется на время, необходимое для

Таблица 1. Структура Ethernet кадра

№	Поле	Размер (байт)
1	Преамбула	7
2	Разделитель фрейма	1
3	MAC-адрес получателя	6
4	MAC-адрес источника	6
5	Тег 802.1Q	4
6	Тип Ethernet или длина	2
7	Полезная нагрузка (минимум)	46
8	CRC-последовательность	4
9	Межпакетная пауза	12

передачи 84 байт. В рамках проводимого эксперимента тег 802.1Q отсутствует, что позволяет использовать минимум 46 байт для передачи данных при использовании линии связи в эквиваленте передачи 84 байт.

3. IP

IP – это протокол сетевого уровня согласно модели OSI [5]. Исходя из спецификации протокол IP не обеспечивает гарантию доставки и не отвечает за установление соединения. Заголовок пакета протокола IP имеет следующий вид (см. табл. 2).

Таблица 2. Структура заголовка протокола IP

Байт	1	2	3	4
4	Версия и длина	Тип сервиса	Полная длина	
8	Идентификатор		Флаги и указатель фрагмента	
12	Время жизни	Протокол	Контрольная сумма	
16	Адрес отправителя			
20	Адрес получателя			

4. TCP/UDP

TCP/UDP – это протоколы транспортного уровня, согласно модели OSI. TCP, в отличие от протокола UDP, позволяет гарантировать целостность передачи данных и устранять дублирование. В зависимости от необходимости использования механизма гарантии доставки на транспортном уровне допускается использование лю-

бого из двух протоколов. В случае использования протокола UDP реализация механизма гарантии доставки производится с помощью настроек режима качества обслуживания (QoS, Quality of Service) в протоколе MQTT [7]. Заголовок пакета UDP состоит из 8 байт и имеет следующий вид (см. табл.3) [11].

Таблица 3. Структура заголовка протокола UDP

1 байт	2 байта	3 байта	4 байта
Порт отправителя		Порт получателя	
Длина сообщения		Контрольная сумма	

Заголовок протокола TCP минимально состоит из 20 байт и имеет следующий вид (см. табл. 4) [10].

Таблица 4. Структура заголовка протокола TCP

Байт	1	2	3-4
0	Порт отправителя		Порт получателя
4	Порядковый номер		
8	Номер подтверждения		
12	Длина заголовка и флаги		Размер окна
16	Контрольная сумма		Указатель важности
20	Опции и смещение		

В рамках проводимого эксперимента предлагается использовать протокол UDP.

5. Message Queuing Telemetry Transport (MQTT)

MQTT – это легковесный сетевой протокол обмена сообщениями по принципу «издатель – подписчик» (publisher/subscriber). MQTT разрабатывался как способ поддержания связи между устройствами в сетях с ограниченной пропускной способностью или непредсказуемой связью. Впервые протокол MQTT был опубликован консорциумом OASIS (Organization for the Advancement of Structured Information Standards) в октябре 2014 г. Данный стандарт находится в открытом доступе [8]. MQTT – один из вариантов для разработчиков, которые создают приложения и устройства с надёжной функциональностью и широкой совместимостью с подключёнными к интернету устройствами и приложениями, включая браузеры, смартфоны и устройства IoT [3].

Протокол MQTT построен по шаблону «издатель – подписчик». В соответствии с данным шаблоном отправители, именуемые издателями, не связаны напрямую с получателями, именуемыми подписчиками, и, как правило, разделены. Разделение может быть организовано в различных плоскостях:

- Издатель и подписчик не общаются напрямую – разделение в пространстве.
- Издатель и подписчик могут быть включены в разное время – разделение во времени.
- Издатель и подписчик не приостанавливают выполнение операций в процессе публикации или получения информации – разделение в синхронизации.

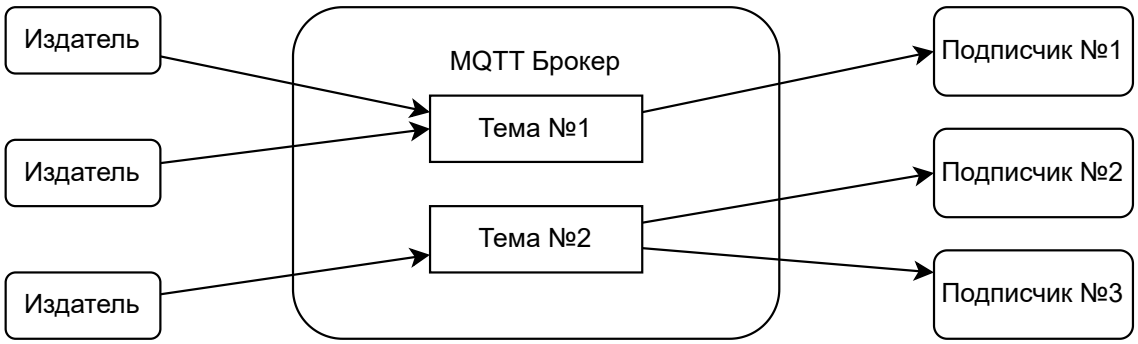

Рис. 1. Архитектура протокола MQTT

Координацией и передачей сообщений управляет брокер (см. рис. 1). Упрощённо процесс обмена данными в такой системе можно описать следующим образом:

1. Издатель отправляет данные с указанием темы, к которой относятся эти данные, брокеру.
2. Брокер отправляет данные подписчикам, имеющим подписку на указанную тему.

Любое количество подписчиков может быть подписано на любое количество тем и получать с помощью этого механизма различные данные без необходимости контактировать напрямую с издателем. Тема представляет собой строку из символов с кодировкой UTF-8. Структура тем имеет формат дерева, что облегчает их организацию. Различные уровни дерева тем разделяются с помощью символа «/» [7].

Протокол поддерживает три уровня надёжности доставки (качества обслуживания):

- 0 – без подтверждения доставки сообщения;
- 1 – с подтверждением доставки;
- 2 – с подтверждением доставки без дублирования.

Пакет протокола состоит из трёх частей (см. табл. 5):

1. Фиксированного заголовка, содержащего в себе информацию о типе сообщения, необходимости дублирования, качестве обслуживания, длине сообщения.

2. Переменного заголовка, содержащего в себе идентификатор пакета, название и версию протокола, флаги соединения.

3. Кадра данных, содержащего в себе полезные данные.

Таблица 5. Структура заголовка протокола MQTT

Бит	7	6	5	4	3	2	1	0
1 байт	Тип сообщения				Вспомогательные флаги			
2 байта	Длина сообщения							

6. Дизайн эксперимента

Проведены два эксперимента, в каждом из которых рассмотрены два различных подхода к агрегации и упаковке данных в системах интернета вещей. В первом эксперименте предлагалось сравнить затраты пропускной способности сети при использовании 5 датчиков, 5 шлюзов-агрегаторов и 1 сервера и при использовании 5 датчиков, 1 шлюза-агрегатора и 1 сервера. В каждой из двух систем датчики передают показания шлюзам-агрегаторам один раз в 5 отсчётов времени. Шлюзы-агрегаторы передают данные серверу один раз в 5 отсчётов времени.

Во втором эксперименте предлагалось сравнить затраты пропускной способности сети между системами, использующими 5 датчиков, 1 шлюз-агрегатор и 1 сервер, где датчики передают информацию агрегатору раз в 1, 3, 5, 7 и 9 отсчётов времени соответственно. Шлюз-агрегатор первой системы передаёт данные серверу по мере поступления данных от датчиков, в то время как шлюз-агрегатор второй системы передаёт данные раз в t_{TT} отсчётов времени

$$t_{TT} = t_{MSD}, \tag{1}$$

где t_{TT} – период для передачи данных, t_{MSD} – наибольший период передачи данных среди всех датчиков.

Единичным отсчётом времени для каждого из экспериментов принята 1 секунда. Одно измерение датчика представляется 2 байтами данных.

Все эксперименты проведены не менее 10 раз для обеспечения выборки не менее чем 10 повторений для получения средних значений с приблизительно нормальным распределением. Это позволяет уменьшить влияние возможных шумов измерений.

Оба эксперимента предполагают использование датчиков, собирающих информацию о внешней среде, подключённых к устройствам – шлюзам-агрегаторам, и пересылку данных от шлюзов-агрегаторов к серверу с использованием Ethernet соединения.

В каждом из экспериментов данные, получаемые шлюзом-агрегатором от датчиков, упаковывались в сообщение протокола MQTT, затем – в сообщение протокола UDP, затем – в сообщение протокола IP и передавались через Ethernet.

Размер сообщения для одной передачи через Ethernet-соединение вычисляется по формуле

$$b = d_s + c_{MQTT} + c_{UDP} + c_{IP}, \tag{2}$$

где b – размер сообщения, d_s – данные от датчиков, c_{MQTT} – заголовок протокола MQTT, c_{UDP} – заголовок протокола UDP, c_{IP} – заголовок протокола IP. Если размер сообщения не превышает минимальный размер данных, передаваемых через протокол Ethernet, размер сообщения принимается равным 46 байтам.

В первой системе первого эксперимента каждому датчику соответствует свой шлюз-агрегатор (см. рис. 2). Датчики опрашиваются каждые 5 секунд. Каждые 5 секунд каждый шлюз-агрегатор отсылает данные серверу. Спустя 600 секунд рассчитывается среднее значения метрики. Это позволяет отфильтровать возможные шумы измерений.

Во второй системе первого эксперимента всем датчикам соответствует единственный в системе шлюз-агрегатор (см. рис. 3). Датчики так же, как и в первом эксперименте, опрашиваются каждые 5 секунд. Каждые 5 секунд шлюз-агрегатор отсылает данные серверу.

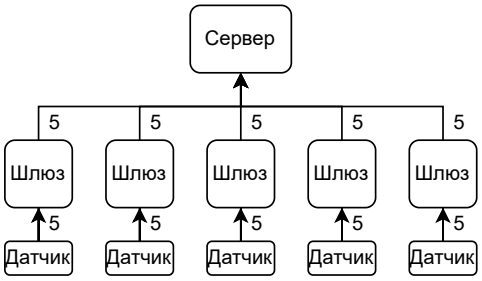

Рис. 2. Структурная схема сети первой системы первого эксперимента

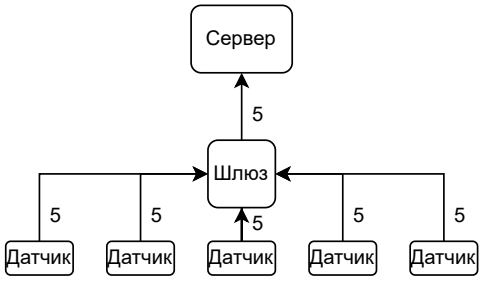

Рис. 3. Структурная схема сети второй системы первого эксперимента

В первой системе второго эксперимента каждому датчику соответствует единственный в системе шлюз-агрегатор (см. рис. 4). Каждый из 5 датчиков имеет своё время опроса, равное 1, 3, 5, 7 и 9 секундам соответственно. Шлюз-агрегатор отсылает данные серверу по мере их получения. Спустя 600 секунд рассчитывается среднее значения метрики.

Во второй системе второго эксперимента каждому датчику соответствует единственный в системе шлюз-агрегатор (см. рис. 5). Каждый из 5 датчиков имеет своё время опроса, равное 1, 3, 5, 7 и 9 секундам соответственно. Шлюз-агрегатор отсылает данные серверу каждые 9 секунд. Спустя 600 секунд рассчитывается среднее значения метрики.

Для протокола MQTT каждый из шлюзов-агрегаторов логически представлял собой «издателя», а сервер – «подписчика». Ожидаемые затраты пропускной способности для каждого из экспериментов были вычислены по формуле

$$bc = ((b + 38) * dtc)/t, \tag{3}$$

где bc – ожидаемые затраты пропускной способности, b – размер сообщения для

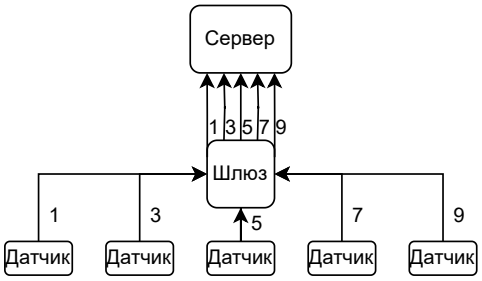

Рис. 4. Структурная схема сети первой системы второго эксперимента

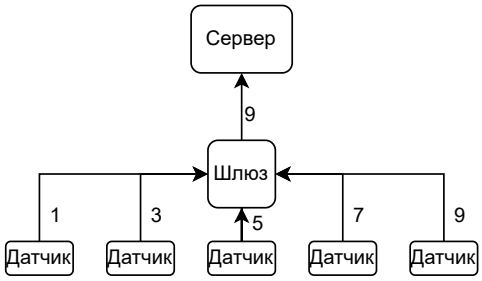

Рис. 5. Структурная схема сети второй системы второго эксперимента

передачи через Ethernet соединение, dtc – количество передач данных, t – период передачи данных (см. табл. 6).

Таблица 6. Ожидаемые затраты пропускной способности

Эксперимент	Затраты пропускной способности, Б\с
1	84
2	16,8
3	150,13
4	9,33

7. Используемые инструменты

В эксперименте были использованы цифровые датчики температуры DS18B20. Датчики подключены к SoC-системам ESP8266, представляющим шлюзы-агрегаторы с использованием интерфейса 1-Wire. Такие системы используются для проектирования устройств интернета вещей [9]. Сервер использует платформу Raspberry Pi 4 и Raspbian Kernel 6.1. Подключение шлюзов-агрегаторов к серверам осуществляется с помощью интегральной схемы Ethernet контроллера W5500 через интерфейс SPI.

Для реализации брокера протокола MQTT использованы инструменты с открытым исходным кодом Mosquitto [1]. Существуют и другие реализации, однако используемые в данном эксперименте имеют открытый исходный код и подробную документацию. Для сбора данных о работе сети используется инструмент tcpdump.

В обоих экспериментах настройки качества обслуживания протокола MQTT установлены в состояние «At most once» для большей прозрачности эксперимента. Использование протокола UDP одновременно с отключением настройки качества обслуживания означает отсутствие в эксперименте механизма гарантии доставки сообщений. Предполагается, что в рамках эксперимента сбои соединений будут отсутствовать и все сообщения достигнут адресатов.

8. Результаты экспериментов

Как было описано ранее, было проведено два эксперимента для оценки затрат пропускной способности сети с использованием разных топологий сети интернета вещей.

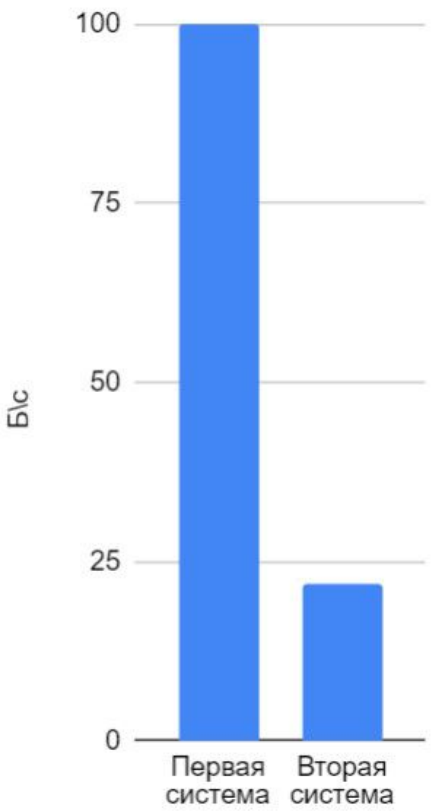

Рис. 6. Структурная схема сети первой системы второго эксперимента

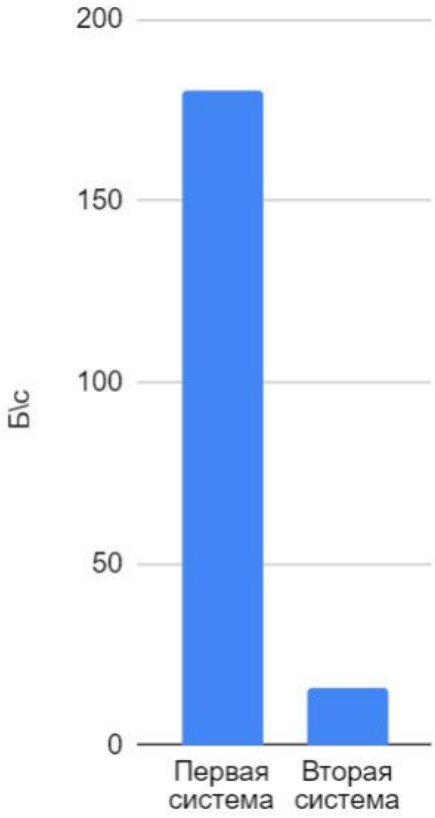

Рис. 7. Структурная схема сети второй системы второго эксперимента

9. Первый эксперимент

С точки зрения затрат пропускной способности сети (см. рис. 6) наибольшее среднее значение, а значит, и наибольшее использование сетевых ресурсов имеет первая система со значением 99,96 Б/с. Вторая система, имеющая такой же период передачи данных, но содержащая в 5 раз меньшее количество шлюзов-агрегаторов, а соответственно, и в 5 раз меньшее количество пересылок данных, имеет затраты пропускной способности сети на уровне 21,94 Б/с. Разница между значениями составляет 78,02 Б/с, или примерно 78 %. Ожидаемая на основании расчётов разница составляет 80 %. Погрешность вносят передачи данных для установки соединения и физическая реализация каналов связи.

10. Второй эксперимент

С точки зрения затрат пропускной способности сети (см. рис. 7) наибольшее среднее значение, а значит, и наибольшее использование сетевых ресурсов имеет первая система со значением 185,29 Б/с. Вторая система, имеющая такое же количество шлюзов-агрегаторов, но передающая данные не по мере поступления, а при накоплении данных от всех датчиков, имеет затраты пропускной способности сети на уровне 15,96 Б/с. Разница между значениями составляет 78,02 Б/с, или примерно 91 %. Ожидаемая на основании расчётов разница составляет примерно 93 %. Погрешность вносят передачи данных для установки соединения и физическая реализация каналов связи.

11. Выводы

Исходя из результатов обоих экспериментов можно сказать, что различные методы агрегации данных в сетях интернета вещей являются эффективным средством для снижения используемых сетевых ресурсов. Уменьшение количества соединений для передачи данных при небольшом размере полезной нагрузки приводит к тому, что большая часть передаваемых данных не несёт в себе пользы для работы системы, а выступает служебной информацией для обеспечения соединения. Увеличивая часть с полезной нагрузкой в таких сообщениях, можно добиться снижения доли служебных данных в соединениях.

Однако замену нескольких устройств шлюзов-агрегаторов одним следует использовать с осторожностью, поскольку такой подход повышает опасность выхода из строя всей системы из-за отказа одного устройства.

Если же целью ставится ещё большая оптимизация использования ресурсов сети и уплотнение данных, следует обратить внимание на более низкоуровневые или специализированные протоколы, позволяющие использовать меньше служебной информации в организации межмашинных связей в сетях интернета вещей.

Данные эксперименты проводились без углублённой настройки каждого протокола под различные возможные задачи и без модификаций для использования служебных полей в передаче полезных данных, поскольку для обеспечения совместимости в системах интернета вещей зачастую используются стандартные протоколы.

12. Заключение

Грамотное использование сетевых ресурсов является одним из важных аспектов в проектировании, построении и эксплуатации систем интернета вещей. Рост отрасли за последние годы создаёт новые вызовы для разработчиков протоколов и стандартов связи. Оптимизация передачи данных в сетях интернета вещей позволяет использовать большее количество устройств в рамках одного и того же объёма сетевых ресурсов. В данной статье рассмотрено теоретическое предложение об уменьшении использования сетевых ресурсов с помощью агрегации данных и использования в передаче большей доли полезных ресурсов. Были проведены эксперименты, в результате которых теоретические предположения были подтверждены.

Литература

1. 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.
2. Костеннов Т.В. Сравнение протоколов связи для организации M2M-взаимодействий в SCADA-системах и системах промышленного интернета вещей // Математические структуры и моделирование. 2023. № 2 (66). С. 91–102.
3. Ли П. Архитектура интернета вещей / перевод с английского М.А. Райтман. М. : ДМК Пресс, 2019. 454 с.
4. IEEE 802.3 ETHERNET. URL: `https://www.ieee802.org/3/` (дата обращения 24.10.2023).
5. Internet Protocol, Protocol Specification. URL: `https://www.ietf.org/rfc/rfc791.txt` (дата обращения 24.10.2023).
6. Ashton K. That ‘Internet of Things’ Thing. In the real world, things matter more than ideas. // RFID Journal. 2009. URL: `http://www.rfidjournal.com/articles/view?4986` (дата обращения 24.10.2023).
7. MQTT Specification. URL: `https://mqtt.org/mqtt-specification/` (дата обращения 11.11.2023).
8. OASIS Standard – MQTT Version 3.1.1. URL: `http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html` (дата обращения 24.10.2023).
9. Light R.A. Mosquitto: server and client implementation of the MQTT protocol // Journal of Open Source Software. 2017. Vol. 2, No. 13. P. 265.
10. Transmission Control Protocol. URL: `https://datatracker.ietf.org/doc/html/rfc793` (дата обращения 24.10.2023).
11. User Datagram Protocol. URL: `https://datatracker.ietf.org/doc/html/rfc768` (дата обращения 24.10.2023).

THE EFFECTIVENESS OF DATA AGGREGATION IN INTERNET OF THINGS NETWORKS USING THE MQTT PROTOCOL

T.V. Kostennov
Ph.D. Student, e-mail: timofey.kostennov@gmail.com

Dostoevsky Omsk State University, Omsk, Russia

Abstract. The high pace of development of the Internet of Things and Industrial Internet of Things industry leads to an increase in the number of devices and transmitted data in designed and used systems. This paper proposes an approach aimed at reducing the use of network resources, based on data aggregation on aggregator gateways. Experiments were carried out to evaluate the proposed approaches in terms of the use of network resources.

Keywords: IoT, MQTT, internet of things, data aggregation.

Дата поступления в редакцию: 24.11.2023