ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ
СПбГУТ))) Март 2016 Том 4 № 1 ISSN 2307-1303 http://www.itt.sut.ru
TELECOM IT
March 2016 Vol. 4 Iss. 1 ISSN 2307-1303 http://www.itt.sut.ru

ИССЛЕДОВАНИЕ ПРОТОКОЛОВ ВЗАИМОДЕЙСТВИЯ ИНТЕРНЕТА ВЕЩЕЙ НА БАЗЕ ЛАБОРАТОРНОГО СТЕНДА

В. Д. Фам[1], Л. О. Юльчиева[1], Р. В. Киричек[1]

[1] СПбГУТ, Санкт-Петербург, 193232, Российская Федерация
Адрес для переписки: ruslan.stk@gmail.com

Информация о статье
УДК 004.725.7
Язык статьи – русский.
Поступила в редакцию 20.01.16, принята к печати 25.02.16.

Ссылка для цитирования: Фам В. Д., Юльчиева Л. О., Киричек Р. В. Исследование протоколов взаимодействия интернета вещей на базе лабораторного стенда // Информационные технологии и телекоммуникации. 2016. Том 4. № 1. С. 55–67.

Аннотация
В статье приводится исследование протоколов Интернета Вещей на базе лабораторного стенда. В качестве аппаратной платформы рассматривается модуль NodeMCU, который использует технологию передачи данных WiFi, являющуюся наиболее распространенной на сегодняшний день для устройств Интернета Вещей. В ходе эксперимента определяются задержки, потери пакетов в зависимости от их размера, отношение служебной информации к полезной за одну транзакцию при передаче данных с использованием протоколов MQTT, CoAP, HTTP/2, в целях определения, в каких устройствах данные протоколы будут наиболее эффективны.

Ключевые слова
Интернет Вещей, протокол, HTTP/2, MQTT, CoAP.

RESEARCH OF PROTOCOLS OF INTERACTION OF THE INTERNET OF THINGS ON THE BASIS OF THE LABORATORY BENCH

V-D. Pham[1], L. Yulchieva[1], R. Kirichek[1]

[1] SPbSUT, St. Petersburg, 193232, Russian Federation
Corresponding author: ruslan.stk@gmail.com

Article info
Article in Russian.
Received 20.01.16, accepted 25.02.16.

For citation: Pham V-D., Yulchieva L., Kirichek R.: Research of Protocols of Interaction of the Internet of Things on the Basis of the Laboratory Bench // Telecom IT. 2016. Vol. 4. Iss. 1. pp. 55–67 (in Russian).

Abstract
The article presents a study of the Internet of Things protocols based on the laboratory bench. As a hardware platform is considered NodeMCU module, which uses the WiFi data transmission technology, is the most common today for the Internet of Things devices. During experiment time delays, losses of packets depending on their size, the attitude of the control footing towards the useful were received for one transaction in case of data transfer with use of the protocols MQTT, CoAP, HTTP/2, for the purpose of determination in what devices these protocols will be most effective.
Keywords
Internet of Things, protocol, HTTP/2, MQTT, CoAP.

Введение

Под Интернетом Вещей понимается совокупность разнообразных приборов, датчиков (сенсоров), исполнительных устройств (актуаторов), объединенных в сеть посредствам любых доступных каналов связи, использующих различные протоколы взаимодействия между собой и единственный протокол доступа – IP к глобальной сети Интернет [1].

В настоящее время Интернетом Вещей охватывается огромный спектр отраслей, начиная от промышленности и заканчивая продуктами питания [2]. Данные от Вещей передаются в сеть связи общего пользования на облачные сервера. Для передачи данных используются протоколы, которых в настоящий момент насчитывается около двадцати пяти [3]. Среди существующих протоколов Интернета Вещей наибольшее распространение получили протоколы MQTT, CoAP, HTTP/2, которые используются для сбора и передачи данных между устройствами и серверной инфраструктурой [4]. Однако у каждого протокола есть свои особенности функционирования, которые проявляются при стрессовых режимах работы сети. По этой причине, выбор и использование какого-либо протокола стали насущной проблемой, так как загрузка сети с каждым годом все возрастает. В докладе рассмотрены и проанализированы особенности функционирования протоколов MQTT, CoAP, HTTP/2 на базе модельной сети Лаборатории Интернета Вещей СПбГУТ. В результате анализа предложены рекомендации по использованию конкретных протоколов для разных типов Интернет устройств.

1 Протоколы взаимодействия интернета вещей

1.1 Протокол MQTT

MQTT (*Message Queuing Telemetry Transport*) – 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 [5].

Протокол создавался, чтобы обеспечить открытость, простоту, минимальные требования к ресурсам и удобство внедрения. MQTT располагается поверх TCP/IP и работает с моделью «клиент/сервер», где каждый датчик является клиентом и подключен к серверу, который является брокером. Протокол MQTT требует обязательного наличия брокера, который управляет распределением данных подписчикам. Все устройства или актуаторы посылают данные только брокеру и принимают данные тоже только от него. В сети на базе протокола MQTT различают 3 объекта (рис. 1)[1]:

- 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.

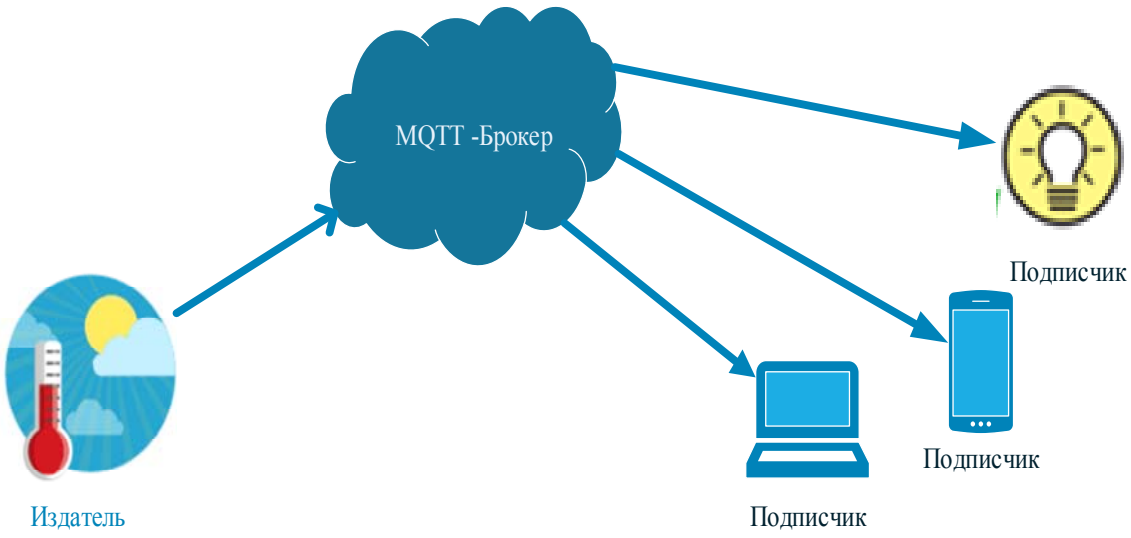

Рис. 1. Основная структура работы протокола MQTT

То есть, когда один клиент, так называемый издатель, передает сообщение *M* на определенную тему *T*, то все клиенты, которые подписываются тему *T*, получают это сообщение *M*. Например, три клиента подключены к брокеру, клиенты *Б*

[1] Mosquitto. URL: http://mosquitto.org

и *C* подписываются на топик «temperature». В какое-то время, когда клиент *A* передает значение «30» на топик «temperature», сразу после получения, брокер передает это сообщение к подписавшимся клиентам (рис. 2).

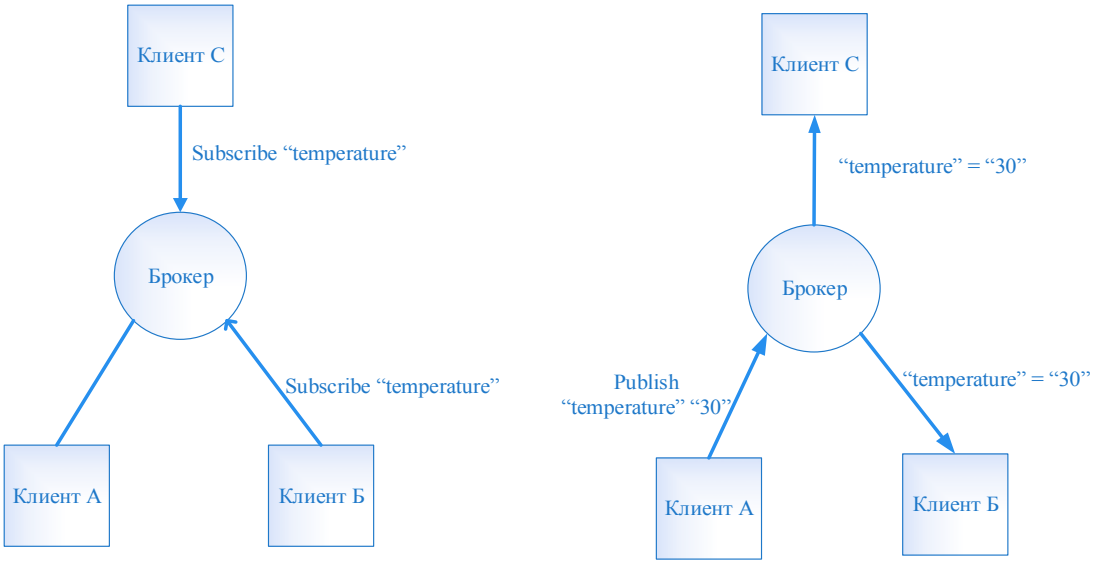

Рис. 2. Обмен сообщением между клиентами через брокера

В MQTT предусматривается 3 выбора надежности обмена сообщениями, которые обеспечиваются тремя уровнями качества обслуживания (*QoS, Quality of Service*)[2]:

- QoS0 – сообщение передается только один раз и не требует подтверждение;
- QoS1 – сообщение отправляется минимум один раз и требует подтверждение;
- QoS2 – для доставки сообщения используется механизм четырехэтапного рукопожатия.

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.

1.2 Протокол COAP

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 функци-

[2] MQTT.org. Mqtelemetry transport. URL: http://mqtt.org

ональных возможностей, которые были вновь разработаны с учетом низкой мощности и малого потребления энергии ограниченных встраиваемых устройств, например, такие как датчик уровня пыли в помещении [6]. Кроме того, были изменены различные механизмы и добавлены некоторые новые возможности, чтобы протокол подходил для Интернета Вещей. Стеки протоколов HTTP и COAP показаны на рис. 3.

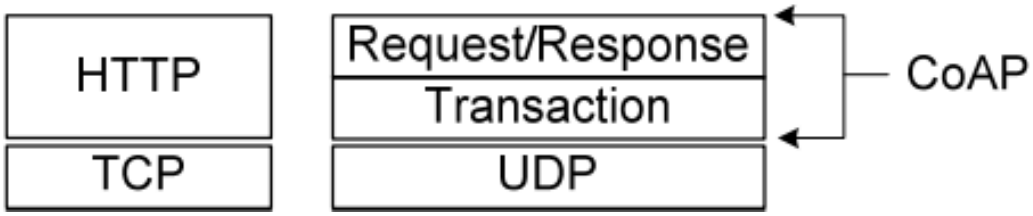

Рис. 3. Стеки протоколов HTTP и COAP (прикладной и транспортный уровни)

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».

Слой транзакций

Обрабатывает единый обмен сообщениями между конечными точками. Сообщения обмена на этом слое могут быть четырех типов:

- «Confirmable» – требует подтверждение;
- «Non-confirmable» – не требует подтверждение;
- квитирование – подтверждает получение «Confirmable» сообщения;
- «Reset» – указывает на то, что «Confirmable» сообщение было получено, но контекст, подлежащий обработке, отсутствует.

Данными сообщениями CoAP обеспечивает механизм собственной надежности.

Когда получателю приходит «Confirmable» сообщение, он всегда возвращает подтверждение. Передающая сторона, в свою очередь, повторно отправляет сообщение, если подтверждение не возвращается в течение определенного периода времени, заданного по умолчанию и с каждым разом возрастающего экспоненциально, пока получатель не посылает сообщение подтверждения приема. Кроме того, это дает возможность асинхронной связи, которая является ключевым требованием для Интернета Вещей.

Но некоторые сообщения не требуют подтверждения. Это особенно верно в отношении сообщений, которые повторяются регулярно, например, повторные показания от датчика. Тогда в качестве более легкой альтернативы, сообщение может быть передано с меньшей надежностью, помеченное как «Non-confirmable». Данное сообщение всегда несет либо запрос, либо ответ и не должно быть пустым.

Слой транзакций также обеспечивает поддержку многоадресной передачи и контроля перегрузки. По своей сути CoAP – пример передачи «один к одному», но тем не менее он поддерживает и многоадресную рассылку, что возможно бла-

годаря расположению над IPv6. И это может быть использовано, например, для обнаружения устройств, или обмена данными через межсетевые экраны.

Слой «Request / Response»

Представляет модель взаимодействия «запрос/ответ» (клиент/сервер) для манипулирования ресурсами и передачи. Сенсорный узел обычно является сервером, но может реализовывать оба стиля общения, что сокращает время разработки и ресурсы, необходимые на устройствах.

В CoAP поддерживается шифрование, но без TCP стандартный TLS не может быть использован для обеспечения безопасности связи. Поэтому в CoAP используется DTLS (Datagram Transport Layer Security), более новая производная TLS, которая за счет дополнений позволяет ему работать на вершине своего UDP транспортного протокола.

1.3 Протокол HTTP/2

Протокол HTTP/2 – это обновленный протокол HTTP версии 2, который был разработан IETF и описан в документе RFC 7540. Он полностью совместим со своим предшественником – HTTP/1.1, который в свою очередь не хорошо подходит для встраиваемых устройств, так как тратит много оперативной памяти и буферного пространства [7]. Кроме этого, он затрачивает больше энергии, из-за чего встает проблема с батарейным питанием. Новая версия протокола HTTP/2 создана, чтобы справляться с данными проблемами[3]. Конечно, он не будет столь же эффективным и идеальным для встраиваемых устройств, как протокол, специально предназначенный для этого, но при проектировании протокола HTTP/2 была рассмотрена данная возможность. Для крупных веб-серверов с большим количеством клиентов целесообразно снижать затраты памяти и пропускной способности. В отличии от протокола HTTP/1.1, для начала, протокол HTTP/2 использует очень эффективную технологию сжатия HPACK (в отличии от GZIP). Эта технология работает путем присвоения имен заголовков и значений записей в таблицах. Затем, используется только необходимый номер записи. HPACK позволяет уменьшить размер заголовка HTTP, за счет чего эффективно используется пропускная способность канала, и также уменьшается время необходимое для обработки анализа, не жертвуя исходным синтаксисом HTTP.

Одним из недостатков HTTP/1.1 является тот факт, что, когда HTTP-сообщение отправлено с заголовком Content-Length определённой длины, просто так его остановить не представляется возможным. Конечно, зачастую можно (но не всегда) разорвать TCP-соединение, но ценой повторного согласования нового TCP-соединения. Гораздо лучше просто отменить отправку и начать новое сообщение. Это может быть достигнуто отправкой HTTP/2-фрейма RST_STREAM, который таким образом предотвратит растрату полосы пропускания и необходимость разрыва каких-либо соединений.

HTTP/2 является бинарным протоколом, т. е. поток делится на фреймы, имеющие фиксированную структуру и размер. Нет необходимости в парсинге для по-

[3] GáborMolnár.node-http2. URL: https://github.com/molnarg/node-http2

иска границ сообщений. HTTP/2 отправляет бинарные фреймы, коих различают несколько типов, но у всех одинаковое строение: тип, длина, флаги, идентификатор потока и полезная нагрузка фрейма. Бинарный формат помогает уменьшить размер отправляемого пакета. В спецификации HTTP/2 существует десять различных типов фреймов, но наиболее важными, которые связывают с HTTP/1.1 являются: DATA (данные) и HEADERS (заголовки).

Одно соединение TCP включает в себя протокол мультиплексирования, т. е. протокол позволяет мультиплексирование нескольких соединений в одно соединение TCP. Мультиплексирование потоков означает, что пакеты множества потоков смешаны в рамках одного соединения. Два или больше отдельных «поезда» данных собираются в один состав, а затем разделяются на другой стороне. Идентификатор потока привязывает каждый фрейм, передаваемый поверх HTTP/2. Поток – это логическая ассоциация, независимая двухсторонняя последовательность фреймов, которыми обмениваются клиент с сервером внутри HTTP/2-соединения. В одном соединении можно содержать множество одновременных открытых потоков от любой из сторон – клиента или сервера. Потоки могут быть установлены и использованы в одностороннем порядке, и могут быть закрыты любой из сторон. Всегда важен и порядок потоков, в втором отправляется фреймы. Получатель или клиент обрабатывает фреймы в порядке их получения.

Протоколы CoAP и MQTT предполагаются для связи шлюза к серверу. В настоящее время многочисленное количество протоколов используется для этих целей, но данные протоколы получили наибольшее распространение при разработке Интернет Вещей. Также возможно эффективное использование CoAP и MQTT, когда необходимо отправлять короткие сообщения. Протокол HTTP/2 больше предполагает использование для Веб Вещей (WoT, WEB of Things) [8]. Структурируем основные различия между протоколами CoAP, MQTT и HTTP/2 и представим в таблице 1 [9].

Таблица 1.

Основные различия в протоколах CoAP, MQTTи HTTP/2

Протокол	**MQTT**	**CoAP**	**HTTP/2**
Транспортный уровень	TCP	UDP	TCP
Безопасность	TLS/SSL	DTLS	TLS/SSL
Обмен сообщениями	Издатель/Подписчик	Запрос/Ответ	Запрос/Ответ
Надежность	3 типа: QoS0, QoS1, QoS2	2 типа: Confirmable, Non-Confirmable	–

2 Экспериментальная часть

2.1 Постановка задачи исследования

Цель: Исследование протоколов взаимодействия прикладного уровня для Интернета Вещей с использованием специализированного аппаратного и программного обеспечения.

Для данного исследования был разработан лабораторный стенд в лаборатории Интернета Вещей СПбГУТ. Реализация протоколов MQTT, HTTP/2 и CoAP была представлена с использованием открытых исходных кодов, а именно Mosquitto, node-http2и libcoap, соответственно, которые были интегрированы с помощью специализированного программного обеспечения для дальнейшего проведения экспериментов. Также была использована программа WireShark для перехвата и анализа трафика[4].

Основа аппаратного обеспечения – датчик температуры и влажности DHT11[5], WiFi модуль NodeMCU[6] и одноплатный компьютер Intel Galileo 2-го поколения, также точка доступа WiFi на базе стандарта IEEE 802.11n модель TLWR841N, к одному порту которой подключено мониторинговый ответвитель TAP NetOptics для перехвата пакетов отправления (информация о состоянии соединения). В качестве сервера/брокера использовался настольный компьютер с операционной системой Ubuntu 14.04 LTS, с предустановленной программой WireShark для перехвата и анализа трафика. Структурная схема стенда приведена на рис. 4.

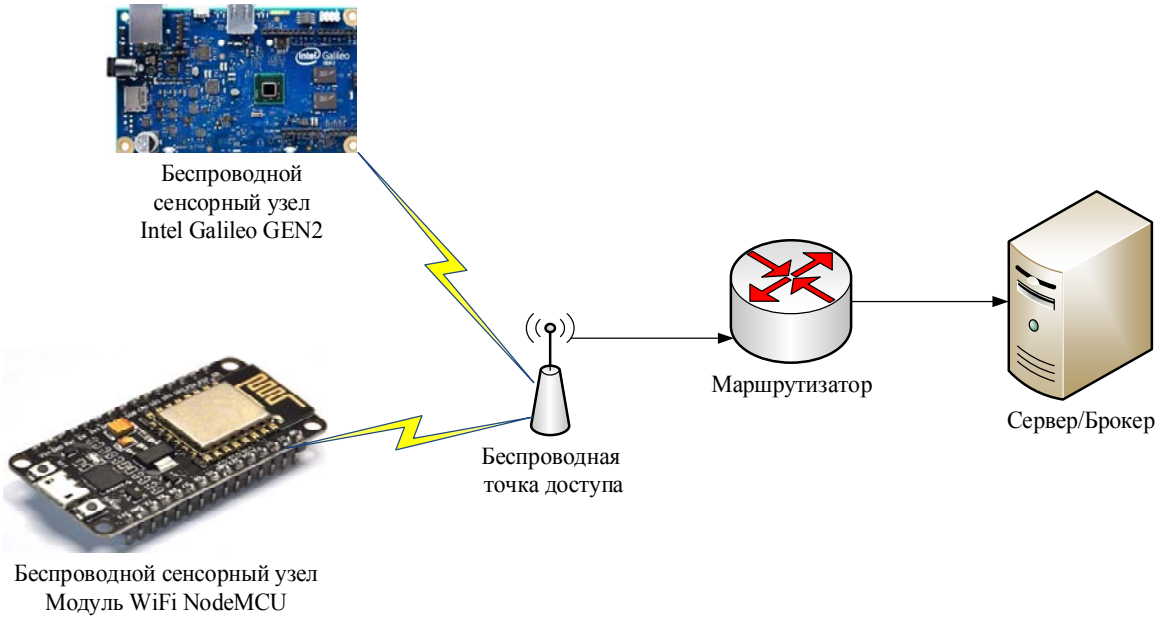

Рис. 4. Структура экспериментального стенда

Схема типовой интернет вещи, для сбора показаний влажности и температуры представлена на рис. 5.

В ходе эксперимента последовательно запускались режим работы клиента для различных протоколов на модуле NodeMCU и одноплатном компьютере Intel Galileo Gen 2.

[4] Wireshark. org. Wireshark. 2016. URL: http://www.wireshark.org
[5] Arduino. URL: http://www.arduino.cc
[6] NodeMCU. URL: https://nodemcu.readthedocs.org/en/dev

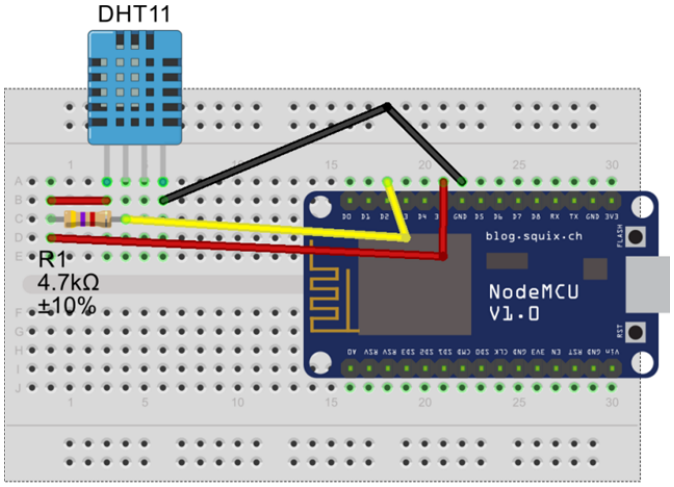

Рис. 5. Схема типовой интернет вещи

2.2 Описание хода эксперимента

Режим работы сервера запускался на персональном компьютере, на котором также была запущена программа WireShark для перехвата и анализа трафика.

Каждые две секунды клиент отправлял одно сообщение на сервер. Например, сообщение *{"temp":"25"}*, означающее, что датчик считал показание температуры равно 25^{o} С. Это показание передается от датчика на сервер. В соответствии с используемым протоколом: MQTT, CoAP или HTTP/2, между ними на транспортном уровне обеспечивается соединение на основе TCP или UDP протокола.

2.3 Анализ результатов экспериментов

Для того, чтобы количественно оценить объем передаваемых данных при использовании протоколов, были проанализированы транзакции клиент-сервера и количество передаваемых байтов. Таблица 2 показывает количество байтов и пакетов, передаваемых за одну транзакцию для MQTT, CoAP и HTTP/2. Транзакция начинается, когда клиент отправляет данные, и заканчивается, когда сервер получает данные или, в некоторых случаях, при получении клиентом подтверждения.

Таблица 2.

Байты, переданные за одну транзакцию клиент-сервера

Протокол	MQTT – QoS0	MQTT – QoS1	MQTT – QoS2	CoAP	HTTP/2
Байты за одну транзакцию	75	135	255	162	1149
Пакеты за одну транзакцию	1	2	4	2	10

Как показано в таблице 2, транзакция HTTP/2, хоть и использует технологию сжатия заголовка HPACK, все равно включает значительно больше байтов и пакетов. Протоколы же MQTTи CoAP имеют короткую длину заголовка. Но CoAP

на транспортном уровне использует UDP, поэтому имеет меньший размер пакета, в отличии от MQTT. После инкапсуляции в заголовках уровня TCP и UDP, MAC пакет этих протоколов может быть передан в один кадр MAC, который имеет размер 80 байтов.

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 3 показано отношение служебной информации к полезной в процентах при передаче одного сообщения.

Протоколы CoAP и MQTT с Qos0 служебные поля в пакете занимают небольшой объем, поэтому при сеансе связи тратится малое количество энергии. Интернет Вещи в основном передают данные по радиоэфиру, в нашем случае по WiFi, поэтому проблема энергопитания будет важна для увеличения жизненного цикла устройства. Для протокола CoAP можно установить такой режим работы сервера, когда при необходимости обновления данных создается запрос и датчик посылает новые значения. Таким образом, удается избежать постоянной передачи данных и неэффективной затраты энергии. Такой вариант подходит для устройств с ограниченным ресурсом.

Таблица 3.

Отношение полезной информации к служебной в одном сообщении

Protocol	**MQTT – QoS0**	**MQTT – QoS1**	**MQTT – QoS2**	**CoAP**	**HTTP/2**
Полезная информация, %	16,8	16,5	16,5	15,3	10,9
Служебная информация, %	83,2	85,5	85,5	84,7	89,1

На рис. 6 представлены результаты экспериментального исследования величины задержки при передаче сообщений.

Видно, что у протокола CoAP наблюдается стабильная небольшая здержка. CoAP использует протокол транспортного уровня UDP, который позволяет быстро обрабатывать данные и передает пакеты небольшой длины, что снижает избыточность в канале передачи. Однако, возникают ситуации, когда соединение клиент-сервер не стабильное. Обычно, такая ситуация возникает, когда за период времени – несколько часов клиент отправляет данные с сообщений разной длины. На рис. 7 представлена зависимость потерь пакетов в зависимости от их размера.

На графике видно, что при увеличении размера сообщений, процент потерь пакетов возрастает. Стоит отметить, что вероятность потерь у протокола CoAP больше, чем у протоколов MQTT и HTTP/2, что обусловлено использованием транспортного протокола UDP, который не гарантирует доставку сообщений. Также стоит учесть, что протокол MQTT мы рассматриваем для 3 различных типов качества (QoS0, QoS1, QoS2). Согласно типам качества видим, что чем большую степень надежности мы присваиваем сообщению (QoS2), тем меньше вероятность потери.

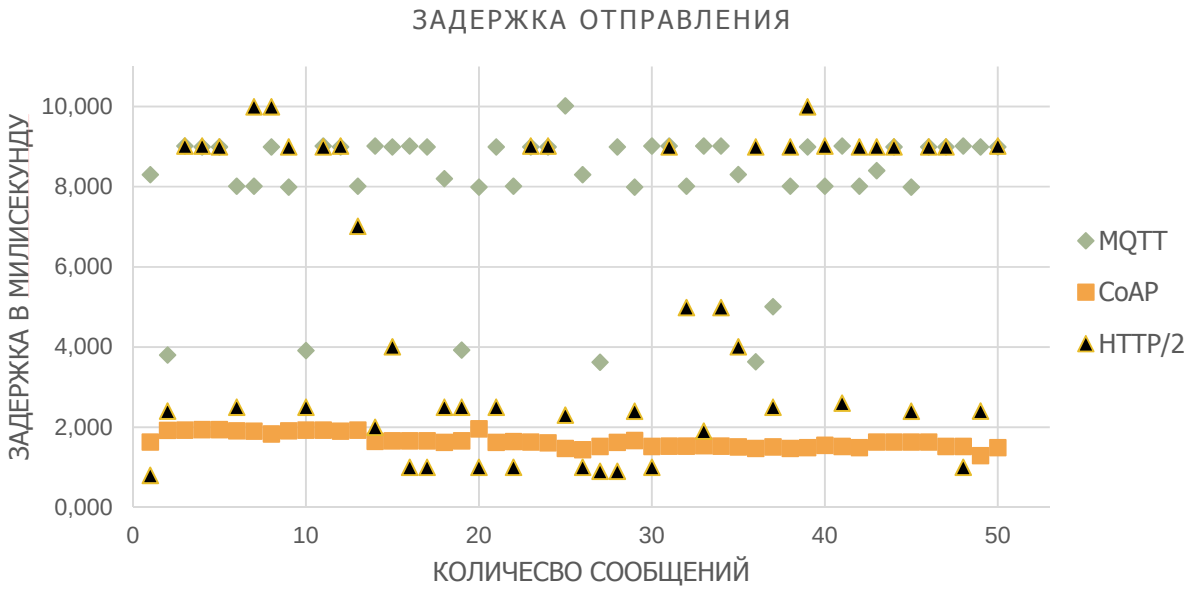

Рис. 6. Задержка при отправке данных от клиента до сервера

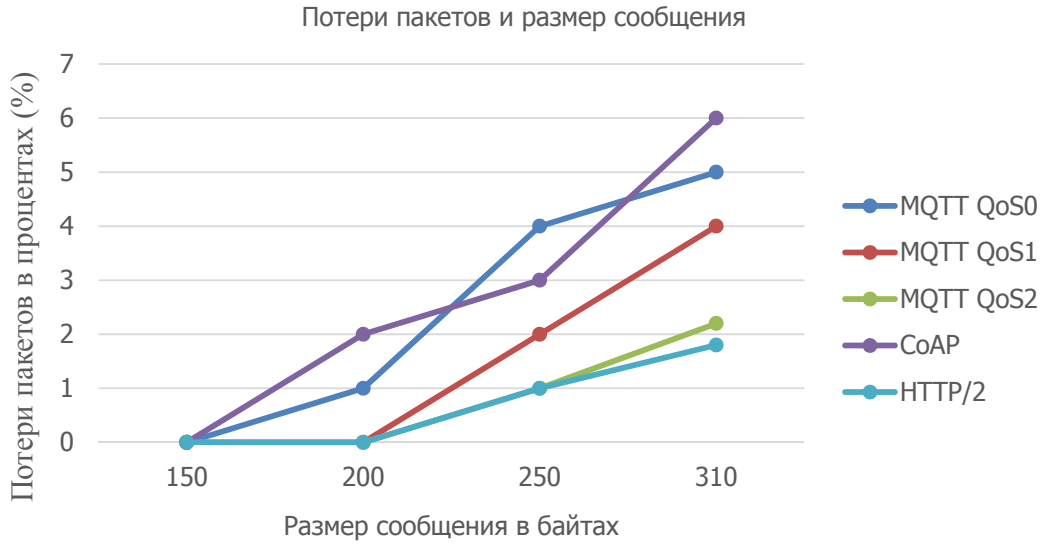

Рис. 7. График влияние размера сообщения на потери пакетов

Заключение

В статье рассмотрены наиболее популярные протоколы Интернета Вещей CoAP, MQTT, HTTP/2, которые используются для отправки информации от датчика до облачного сервера. В ходе исследования выявлено, что для протоколов MQTT и CoAP характерны меньшие накладные расходы на передачу данных (в связи с небольшим количеством служебного трафика) и меньшей полосы пропускания, нежели чем у протокола HTTP/2. Данные протоколы хорошо адаптированы для маломощных устройств Интернета Вещей на базе микроконтроллеров. Для своей ра-

боты протокол MQTT не требует постоянного соединения между клиентом и сервером, также, как и протокол CoAP, чего не сказать о HTTP/2. Экспериментальные результаты показали, что эффективность рассмотренных протоколов зависит от различных условий сети связи. Наиболее оптимальным является протокол MQTT, в котором возможно задавать параметры, отвечающие за надежность доставки сообщений. Протокол HTTP/2 при условии, что сеть связи работает стабильно будет более подходящим для Веб Вещей - информация доставляется быстро и может быть визуализирована в мобильных приложениях или на персональных компьютерах. В связи с этим, в настоящее время правильный выбор протоколов для различных Интернет Вещей помогает решить задачу экономии ресурсов как энергопотребления, так и гарантированной доставки. Особенно это актуально в связи с увеличением количества устройств Интернета Вещей. Так по данным J'son & Partners Consulting, несмотря на текущие проблемы в экономике, к 2018 году рынок Интернета Вещей в России достигнет уровня в 32 млн. подключенных устройств.

Литература

1. Росляков А. В., Ваняшин С. В., Гребешков А. Ю., Самсонов М. Ю. Интернет вещей / под ред. А. В. Рослякова. Самара : ПГУТИ, ООО «Издательство Ас Гард», 2014. 340 с.

2. Кучерявый А. Е. Интернет Вещей // Электросвязь. 2013. № 1. С. 21–24.

3. Кучерявый А. Е. Самоорганизующиеся сети и новые услуги // Электросвязь. 2009. № 1. С. 19–23.

4. Waher P. Learning Internet of Things. URL: http://rutracker.org/forum/viewtopic.php?t=5078184. – 2015.

5. Light R. Mosquitto-an open source mqtt v3.1 broker. URL: http://mosquitto.org.

6. Shelby Z., Hartke K., Bormann C., The Constrained Application Protocol (CoAP). Universitaet Bremen TZI, June 2014. URL: https://tools.ietf.org/html/rfc7252.

7. Belshe M., Peon R., Thomson M. Hypertext Transfer Protocol Version 2 (HTTP/2). URL: https://tools.ietf.org/html/rfc7540. May 2015.

8. Castellani A. P., et al. Architecture and Protocolsfor the Internet of Things: A Case Study. In Proceedings of First International Workshop on the Web of Things (WoT). – 2010.

9. Philip N. MQTT and CoAP: Underlying Protocols for the IoT. URL: http://electronicdesign.com/iot/mqtt-and-coap-underlying-protocols-iot.

References

1. Roslyakov A. V., Vanyashin S. V., Grebeshkov A. Yu., Samsonov M. Yu. The Internet of Things // Edited by A. V. Roslyakov. Samara: PSUTI, Ltd «Publishing House Asgard», 2014. 340 p.

2. Koucheryavy A. E. The internet of Things // Elektrosvyaz'. 2013. № 1. pp. 21–24.

3. Koucheryavy A. E. Self-Organizing Networks and New Services // Elektrosvyaz'. 2009. № 1. pp. 19–23.

4. Waher P. Learning Internet of Things. URL: http://rutracker.org/forum/viewtopic.php?t=5078184. – 2015.

5. Light R. Mosquitto-an open source mqtt v3.1 broker. URL: http://mosquitto.org.

6. Shelby Z., Hartke K., Bormann C., The Constrained Application Protocol (CoAP). Universität Bremen TZI, June 2014. URL: https://tools.ietf.org/html/rfc7252.

7. Belshe M., Peon R., Thomson M. Hypertext Transfer Protocol Version 2 (HTTP/2). URL: https://tools.ietf.org/html/rfc7540. May 2015.

8. Castellani A. P., et al. Architecture and Protocols for the Internet of Things: A Case Study. In Proceedings of First International Workshop on the Web of Things (WoT). – 2010.

9. Philip N. MQTT and CoAP: Underlying Protocols for the IoT. URL: http://electronicdesign.com/iot/mqtt-and-coap-underlying-protocols-iot

Фам Ван Дай – студент, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, daipham93@gmail.com

Юлчиева Лолита Олимжоновна – студент, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, lola_9@mail.ru

Киричек Руслан Валентинович – кандидат технических наук, доцент, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, ruslan.stk@gmail.com

Pham Van-Dai – student, SPbSUT, St. Petersburg, 193232, Russian Federation, daipham93@gmail.com

Yulchieva Lolita – student, SPbSUT, St. Petersburg, 193232, Russian Federation, lola_9@mail.ru

Kirichek Ruslan – Ph.D., assistant professor, SPbSUT, St. Petersburg, 193232, Russian Federation, ruslan.stk@gmail.com