

Е. Е. Малохвій, Г. І. Молчанов, Ю. В. Паржин

Національний технічний університет “Харківський політехнічний інститут”, Харків, Україна

ДОСЛІДЖЕННЯ ПРОТОКОЛІВ ПЕРЕДАЧІ ДАНИХ В УМОВАХ ІНТЕРНЕТУ РЕЧЕЙ

Анотація. Предметом дослідження в статті є протоколи передачі даних в умовах Інтернет речей. Мета роботи – порівняння протоколів Інтернет речей на базі демонстраційного стенда. У якості апаратної платформи розглядається модуль WeMos D1, який використовує технологію передачі даних Wi-Fi, що є найбільш поширеною на сьогоднішній день для пристроїв Інтернет речей. В ході експерименту визначаються затримки, втрати пакетів в залежності від їх розміру, відношення службової інформації до корисної за одну транзакцію при передачі даних з використанням протоколів: MQTT, CoAP, HTTP/2; з метою визначення, в яких пристроях наведені протоколи найбільш ефективні.

Ключові слова: Інтернет речей, протокол, MQTT, CoAP, HTTP/1.1, HTTP/2

Вступ

Багато технологій кардинально змінили стиль життя протягом ХХІ ст.: радіозв'язок, телебачення, комп'ютери, мобільні телефони та Інтернет є незамінними в сучасному світі. Особливого поширення набула технологія Інтернет речей, основною концепцією якої є можливість підключати усілякі об'єкти «речей» до мережі, обробляти інформацію, що надходить з навколишнього середовища, обмінюватися нею і виконувати різні дії залежно від отриманої інформації. Інтернет речей вважають наступним етапом технічної революції [1]. Важливими функціями цієї концепції є полегшення повсякденного життя, підвищення ефективності та якості роботи, енергозощадження тощо.

В даний час Інтернетом речей охоплюється величезний спектр галузей, починаючи від промисловості і закінчуючи продуктами харчування [2]. Прикладом впровадження Інтернет речей є система «розумний будинок». Однією із функцій «розумного будинку» є контроль параметрів навколишнього середовища, залежно від чого здійснюється регулювання життєво важливих показників, таких як температура та вологість, в приміщенні. У зимовий період нагріваючі прилади залежно від температури повітря ззовні, вітру, часу доби без втручання людини регулюють інтенсивність опалення, що дає змогу значно зменшити споживання енергоносіїв. А це дозволяє створити максимально комфортні умови проживання людей при максимально економному споживанні ресурсів.

Дедалі частіше дані від «речей» передаються в мережу зв'язку загального користування на хмарні сервіси. Для передачі даних використовуються протоколи, яких зараз налічується близько двадцяти п'яти [3]. Серед існуючих протоколів Інтернету речей найбільшого поширення набули протоколи: MQTT, CoAP, HTTP/2, які використовуються для збору та передачі даних між пристроями та іншими учасниками інфраструктури Інтернет речей [4]. Однак у кожного протоколу є свої особливості функціонування, які проявляються при стресових режимах експлуатації мережі. З цієї причини, вибір і використання будь-якого протоколу стали нагальною проблемою, оскільки завантаження мережі з кожним

роком все зростає. У статті розглянуто і проаналізовано особливості функціонування протоколів: MQTT, CoAP, HTTP/2 на базі власного демонстраційного стенду. В результаті аналізу запропоновані рекомендації по використанню конкретних протоколів для різних типів пристроїв.

Виклад основного матеріалу

MQTT (*Message Queuing Telemetry Transport*) — простий протокол обміну повідомленнями, який реалізує модель «публікація / підписка» (*publish / subscribe*). Протокол створювався для забезпечення відкритості, простоти, мінімальних вимог к ресурсам та зручності впровадження [5].

Протокол створювався, щоб забезпечити відкритість, простоту, мінімальні вимоги до ресурсів та зручність. MQTT базується поверх TCP/IP та працює з моделлю «клієнт / сервер», де кожен датчик є клієнтом та підключений до серверу, який виступає у ролі брокера. Протокол MQTT потребує обов'язкової наявності брокера, який керує розподілом даних підписникам. Усі пристрої та актуатори надсилають дані тільки брокеру та отримують дані лише від нього. У мережі на базі протоколу MQTT розрізняють 3 об'єкти:

Видавець (*Publisher*) — MQTT, який під час виникнення відповідної події передає брокеру інформацію про неї, публікуючи відповідні топіки;

Брокер (*Broker*) — MQTT-сервер, який отримує інформацію від видавців та передає її відповідним підписникам, окрім цього в складних системах може виконувати різні операції, пов'язані з аналізом та обробкою отриманих даних. Різні брокери можуть з'єднатися між собою, якщо вони підписуються на повідомлення одне одного;

Підписчик (*Subscriber*) — MQTT-клієнт, який після підписки до брокера переважно «слухає» його та постійно готовий до отримання та обробки вхідного повідомлення на необхідні йому топіки від брокера.

Тобто, коли один клієнт видавець передає повідомлення М на певну тему Т, усі клієнти, які підписалися на тему Т, отримують це повідомлення М. Наприклад, три клієнта підключені до брокера, клієнт Б та С підписуються на топик «temperature». В певний момент часу клієнт А передає значення «30» на топик

«temperature», миттєво після отримання, брокер передає це повідомлення до підписників.

В MQTT передбачається три вибори надійності обміну повідомленнями, які забезпечуються трьома рівнями якості обслуговування (QoS, Quality of Service):

QoS0 — повідомлення передається тільки один раз та не потребує підтвердження;

QoS1 — повідомлення надсилається мінімум один раз та потребує підтвердження;

QoS2 — для доставки повідомлення використовується механізм чотирьох етапного рукоштовування.

Крім цього, поверх рівня TCP присутній рівень безпеки TLS (Transport Layer Security), раніш відомий як SSL (Secure Socket Layer). Порт 8883 забезпечує безпеку зв'язку, якщо адреса брокера працює з цим портом, то трафік передається з шифруванням.

CoAP (*Constrained Application Protocol*) — протокол, розроблений IETF (*Internet Engineering Task Force*) та описан в документі RFC 7252 [6]. Протокол працює на прикладному рівні, та призначений для передачі даних по лініях з обмеженою пропускну здатністю. CoAP був розроблений на основі протоколу HTTP, представляє собою його бінарну версію, але не є його сліпою копією. CoAP складається з підмножини HTTP функціональних можливостей, які були розроблені з врахуванням низької потужності та споживання енергії вбудованих пристроїв. Крім цього, були змінені різні механізми та додані деякі нові можливості, щоб протокол підходив для Інтернет речей.

На відміну, від протоколу HTTP, який є текстовим та використовує TCP, CoAP — бінарний протокол, який транспортується через UDP, що зменшує розміри службових даних та збільшує гнучкість в моделях зв'язку.

CoAP організований в два рівня: рівень транзакцій та рівень «Request / Response». Рівень транзакції обробляє єдиний обмін повідомленнями між кінцевими точками. Повідомлення обміну на цьому рівні можуть бути чотирьох типів:

«Confirmable» — вимагає підтвердження;

«Non-confirmable» — не вимагає підтвердження;

квітування — підтверджує отримання «Confirmable» повідомлення;

«Reset» — вказує на те, що «Confirmable» повідомлення було отримано, але контекст, потребує обробки.

Перерахованими повідомленнями CoAP забезпечує механізм власної надійності. Коли отримувачу приходиться «Confirmable» повідомлення, він завжди повертає підтвердження. Сторона, яка надіслала повідомлення, в свою чергу, знову надсилає повідомлення, якщо підтвердження не повертається протягом певного періоду часу, заданого за замовчуванням та постійно зростає, доки отримувач не надіслав підтвердження отримання. Крім цього, це дає можливість асинхронного зв'язку, який є ключовою вимогою до Інтернет речей. Але деякі повідомлення не потребують підтвердження. Це особливо вірно по відношенню до повідомлень, які регулярно повторюються, наприклад однакові значення від датчику.

Тоді в якості більше легкої альтернативи, повідомлення передається з меншою надійністю, відмічене як «Non-confirmable». Це повідомлення завжди несе запит або відповідь та не повинне бути порожнім. А також, рівень транзакції забезпечує підтримку багатоадресної передачу та контролю перезавантаження. Таким чином CoAP — приклад передачі «один до одного», проте як зазначалося раніше, протокол підтримує багатоадресне надсилання, що можливо завдяки розташуванню над IPv6. Це може бути використано, наприклад, для виявлення пристроїв, та обміну даними через між мережеві екрани. Рівень «Request / Response» представляє модель взаємодії «запит / відповідь» або «клієнт / сервер» для маніпулювання ресурсами та передачі. В CoAP підтримується шифрування, але без TCP стандартний TLS не може бути використаний для забезпечення безпеки зв'язку. Тому в CoAP використовується DTLS (Datagram Transport Layer Security), більш нова похідна TLS, яка за рахунок додатків дозволяє йому працювати на верхині свого UDP транспортного протоколу.

Протокол HTTP/2 — оновлений протокол HTTP версії 2, який був розроблений IETF та описан в документі RFC 7540 [7]. Він повністю сумісний зі своїм попередником — HTTP/1.1, який в свою чергу не досить підходить для «речей», оскільки витрачає багато оперативної пам'яті та буферного простору. Крім цього, він витрачає більше енергії, із-за чого вистає проблема батарейного живлення. Нова версія протоколу HTTP/2 створена, щоб вирішити проблеми попередника.

HTTP/2 є бінарним протоколом, тобто потік ділиться на фрейми, які мають фіксовано структуру та розмір. Немає необхідності в повному зчитуванні для пошуку границь повідомлення. HTTP/2 надсилає бінарні фрейми, які розрізняють декілька типів, але у всіх однакова будова: тип, довжина, прапори, ідентифікатор потоку та корисне навантаження фрейму. Бінарний формат допомагає зменшити розмір кінцевого пакету. В специфікації HTTP/2 існує десять різних типів фреймів, але найбільш важливими, які зв'язують з HTTP/1.1 є: DATA (дані) та HEADERS (заголовки).

Одне з'єднання включає в себе протокол мультиплексування, тобто протокол дозволяє мультиплексування декількох з'єднань в одне з'єднання TCP. Мультиплексування потоків означає, що пакети з багатьох потоків змішані в рамках одного з'єднання. Два або більше окремих потоки даних збираються в один, а потім на іншій стороні розділяються. Ідентифікатор потоку прив'язує кожен фрейм, передаючи поверх HTTP/2. Потік — це логічна асоціація, незалежна двостороння послідовність фреймів, якими обмінюються клієнт з сервером всередині HTTP/2 з'єднання. В одному з'єднанні можливо розмістити багато одночасно відкритих потоків від будь-якої сторони, клієнта або сервера. Потоки можуть бути встановлені та використані в односторонньому порядку, та можуть бути зачинені з будь-якої сторони. Завжди важливий порядок потоків, в якому надсилаються фрейми. Отримувач або клієнт обробляє їх в порядку отримання.

Протоколи MQTT та CoAP передбачаються для зв'язку шлюзу к серверу. На даний час, численна кількість протоколів використовується для цих цілей, але наведені вище протоколи отримали найбільшу розповсюдженість під час розробки Інтернет речей. Також можливо ефективне використання CoAP та MQTT, коли необхідно надсилати короткі повідомлення. Протокол HTTP/2 більш передбачає використання для «веб речей» (WoT, *Web of Things*) [8]. Отже, визначимо основні відмінності між протоколами MQTT, CoAP та HTTP/2 в табл 1 [9].

Для даної статті був розроблений демонстраційний стенд. Реалізація протоколів: MQTT, CoAP та HTTP/2 була представлена з використанням open-source проєктів, а саме mosquitto, node-http2 та lib-coap, відповідно, які були інтегровані за допомогою спеціалізованого ПО для подальшого проведення експерименту. Також була використана програма WireShark для перехвату та аналізу трафіку.

В експерименті, кожні дві секунди клієнт надсилав одне повідомлення на сервер. Наприклад, повідомлення з поточною температурою повітря ({"temp": "25"}), яка вимірюється за допомогою датчика BME280 від компанії Bosch [10]. Цей показник передається від датчика через пристрій на сервер. Відповідно з використанням протоколів: MQTT, CoAP, HTTP/2, між ними на транспортному рівні забезпечується з'єднання на основі TCP або UDP протоколів.

Для того, щоб кількісно оцінити об'єм передачі даних під час використання протоколів, було проаналізовано транзакції клієнт-сервера та кількість переданих байтів. табл. 2 демонструє кількість байтів та пакетів, які передаються за одну транзакцію для MQTT, CoAP та HTTP/2. Транзакція починається, коли клієнт надсилає дані, та закінчується, коли сервер отримує дані або, у деяких випадках, під час отримання клієнтом підтвердження.

Виходячи з показників в табл. 2, транзакція HTTP/2, хоч і використовує технологію стиснення заголовків HPACK, все одно включає значно більше байтів та пакетів. Протоколи MQTT та CoAP мають коротку довжину заголовку. Але CoAP на транспортному рівні використовує UDP, тому має менший розмір пакета, на відміну від MQTT. Після інкапсуляції в заголовках рівня TCP та UDP, MAC пакет цих

протоколів може бути переданий в один кадр MAC, який має розмір 80 байтів.

Повідомлення ділиться на дві частини: корисну інформацію та службову. Ці частини впливають на витрати ресурсів каналу та енергії батарейного живлення. Для покращення ефективності потребується зниження службової інформації. В табл. 3 наведено відношення службової інформації до корисної в відсотках під час передачі одного повідомлення.

Протоколи MQTT та CoAP з QoS0 — службові поля в пакеті займають незначний об'єм, тому під час сеансу зв'язку витрачається мала кількість енергії. Інтернет речей зазвичай передає дані через радіоефір, в нашому випадку через Wi-Fi, тому проблема енергоспоживання буде важлива для збільшення життєвого циклу пристрою. Для протокола CoAP можливо встановити режим роботи сервера, коли при необхідності оновлення даних створюється запит та датчик надсилає нові значення. Таким чином, вдається уникнути постійної передачі даних та неефективної витрати енергії. Такий варіант підходить для пристроїв з обмеженими ресурсами.

На рис. 1 наведені результати експериментального дослідження величини затримки під час передачі повідомлення. Видно, що у протокола CoAP спостерігається стабільно невелика затримка. CoAP використовує протокол транспортного рівня UDP, який дозволяє швидко оброблювати дані та передає пакет невеликої довжини, що зменшує надлишковість в каналі передачі.

Проте, виникає ситуація, коли з'єднання клієнт-сервер не стабільне. Зазвичай, така ситуація виникає, коли за період часу — декілька годин клієнт надсилає дані з повідомленнями різної довжини. На рис. 2 наведено залежність втрат пакетів в залежності від їх розміру. На графіку видно, що під час збільшення розміру повідомлення, відсоток втрат пакетів зростає. Варто зазначити, що вірогідність втрати у протокола CoAP більше, ніж у протоколів MQTT та HTTP/2, що зумовлено використанням транспортного протоколу UDP, який не гарантує доставку повідомлення. Також варто врахувати, що протокол MQTT ми розглядаємо для трьох різних типів надійності (QoS0, QoS1, QoS2). Згідно типам надійності бачимо, що чим більшу ступінь надійності ми присвоюємо повідомленню (QoS2), тем менша вірогідність втрати.

Таблиця 1 – Основні відміни в протоколах MQTT, CoAP та HTTP/2

Протокол	MQTT	CoAP	HTTP/2
Транспортний рівень	TCP	UDP	TCP
Безпека	TSL/SSL	DTLS	TLS/SSL
Обмін повідомленнями	Публікація/Підписка	Запит/Відповідь	Запит/Відповідь
Надійність	3 типи: QoS0, QoS1, QoS2	2 типи: Confirmable, Non-confirmable	немає

Таблиця 2 – Байти, передані за одну транзакцію клієнт-сервер

Протокол	MQTT (QoS0)	MQTT (QoS1)	MQTT (QoS2)	CoAP	HTTP/2
Байти за одну транзакцію	75	135	255	162	1149
Пакети за одну транзакцію	1	2	4	2	10

Таблиця 3 – Відношення корисної інформації до службової в одному повідомленні

Протокол	MQTT (QoS0)	MQTT (QoS1)	MQTT (QoS2)	CoAP	HTTP/2
Корисна інформація, %	16,8	16,5	16,5	15,3	10,9
Службова інформація, %	83,2	85,5	85,5	84,7	89,1

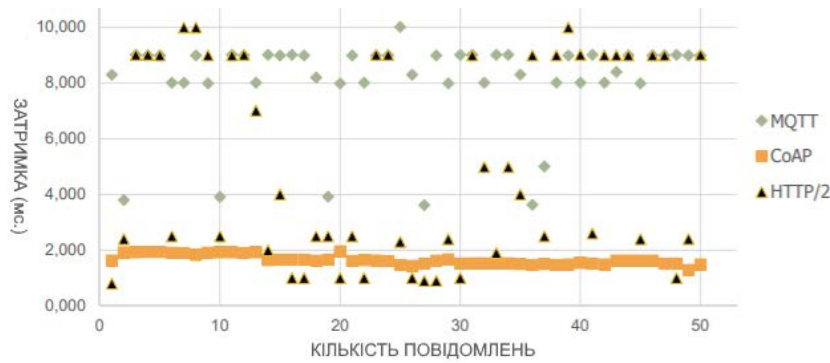


Рис. 1. Затримки під час надсилання даних від клієнта до сервера

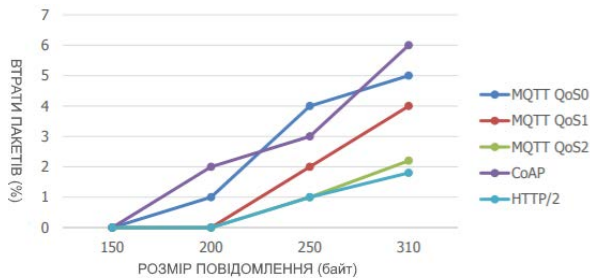


Рис. 2. Графік впливу розміру повідомлення на втрати пакетів

Висновок

У статті розглянуто найбільш популярні протоколи Інтернету речей: MQTT, CoAP, HTTP/2, які використовуються для відправки інформації від датчика до хмарного сервісу. В ході дослідження виявлено, що для протоколів MQTT та CoAP характерні менші накладні витрати на передачу даних (у зв'язку з невеликою кількістю службового трафіку) і меншої смуги пропускання, у порівнянні з протоколом HTTP/2. Дані протоколи добре адаптовані для мало-

потужних пристроїв Інтернет речей на базі мікроконтролерів. Для своєї роботи протокол MQTT не вимагає постійного з'єднання між клієнтом і сервером, також, як і протокол CoAP, на відміну від протоколу HTTP/2.

Експериментальні результати продемонстрували, що ефективність розглянутих протоколів залежить від різних умов мережі зв'язку. Найбільш оптимальним є протокол MQTT, в якому можливо задавати параметри, що відповідають за надійність доставки повідомлень. Протокол HTTP/2 за умови, що мережа зв'язку працює стабільно буде більш оптимальним для «веб речей» — інформація доставляється швидко і може бути відображена в мобільних додатках або на персональних комп'ютерах.

У зв'язку з цим, в даний час вибір протоколів для пристроїв в залежності від умов експлуатації допомагає вирішити задачу економії ресурсів як енергоспоживання, так і гарантованої доставки. Особливо це актуально у зв'язку з розповсюдженням технології Інтернет речей та збільшенням загальної кількості «речей».

СПИСОК ЛІТЕРАТУРИ

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. No 1. pp. 21–24.
3. Koucheryavy A. E. Self-Organizing Networks and New Services // *Elektrosvyaz*'. 2009. No 1. pp. 19–23.
4. Perry Lea. IoT and Edge Computing for Architects 2nd edition / Lea Perry – Packt Publishing, 2020. – 632 p.
5. Gaston C. Hillar. MQTT Essentials – A Lightweight IoT Protocol / C. Hillar Gaston – Packt Publishing, 2017. – 280 p.
6. RFC 7252 – The Constrained Application Protocol (CoAP) – Режим доступу: <https://tools.ietf.org/html/rfc7252>, 07.02.2020.
7. RFC 7540 – Hypertext Transfer Protocol Version 2 – Режим доступу: <https://tools.ietf.org/html/rfc7540>, 07.02.2020.
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. – Режим доступу: <https://electronicdesign.com/iot/mqtt-and-coap-underlying-protocols-iot>
10. Malokhvii E. Library for Bosch Sensortec BME280: combined temperature, pressure, humidity sensor. – Режим доступу: <https://github.com/malokhvii-eduard/arduino-bme280>

Received (Надійшла) 05.01.2022

Accepted for publication (Прийнята до друку) 09.03.2022

Research of data transmission protocols in the conditions of the Internet of Things

E. Malokhvii, H. Molchanov

Abstract. The subject of research in the article is the are the protocols of data transmission in the conditions of the Internet of Things. The purpose of the work is to compare protocols of the Internet of Things on the basis of a demonstration stand. The WeMos D1 module is considered as a hardware platform, which uses Wi-Fi data transmission technology, which is the most common today for Internet of Things devices. During the experiment, are determined delays, packet loss depending on their size, the ratio of service information to useful information per transaction. All measurements are made using protocols: MQTT, CoAP, HTTP/2; in order to determine in which kind of devices the protocols are most effective.

Keywords: Internet of Things, protocol, MQTT, CoAP, HTTP/1.1, HTTP/2.