УДК 004.7 DOI: 10.31673/2518-7678.2019.048085

Звенігородський О.С., к.т.н.; **Кутовий С.О.,**
Прокопов С.В., к.т.н.; **Рижаков М.М.**

ЯКІСТЬ ОБСЛУГОВУВАННЯ ІНТЕРНЕТ РЕЧЕЙ В ПРОТОКОЛІ MQTT, ОСНОВНІ ОСОБЛИВОСТІ І ПРОЦЕДУРИ

Zvenigorodsky O.S., Kutovoy S.O., Prokopov S.V., Ryzhakov M.M. Service quality internet of things in the MQTT protocol, key features and procedures

The article discusses the Protocol MQTT (Message Queue Telemetry Transport) Internet of things, its features, applications, characteristic procedures. The different equal qualities of service in MQTT are explained. The analysis of information elements and messages. The justification for choosing the level of service that meets the requirements of network reliability and application logic is made. It is proved that objectively the routing paradigm shifts from routing at the transport layer to routing at the application when serviced by IOT. It is shown that the topicality of the topic is due to the rapid development of the "publisher-subscriber" architecture, for which the protocol is most characteristic. The relevance of the topic is due to the streaming development of the publisher-subscriber architecture, for which this protocol is the most characteristic.

Keywords: Internet of Things, IoT, MQTT, Subscriber Publisher, Broker, QoS, Quality of Service

Звенігородський О.С., Кутовий С.О., Прокопов С.В., Рижаков М.М. Якість обслуговування Інтернет речей в протоколі MQTT, основні особливості і процедури

У статті розглядається протокол MQTT (Message Queue Telemetry Transport) Інтернет речей, його особливості, варіанти застосування, характерні процедури. Пояснюються різні рівні якості обслуговування в MQTT**.** Проводиться аналіз інформаційних елементів і повідомлень. Проведено обґрунтування вибору рівня обслуговування, який відповідає вимогам надійності в мережі та логіці застосування. Доведено, що об'єктивно парадигма маршрутизації зміщується від маршрутизації на транспортному рівні до маршрутизації на прикладному при обслуговувані Іот. Показано, що актуальність теми обумовлена стрімким розвитком архітектури "видавець-підписчик", для якої найбільш характерним є даний протокол.

Ключові слова: Інтернет речі, Іot, MQTT, видавець-підписчик, брокер, QoS, Quality of Service

Звенигородский А.С., Кутовой С.О., Прокопов С.В., Рыжаков М.М. Качество обслуживания интернет вещей в протоколе MQTT, основные особенности и процедуры

В статье рассматривается протокол MQTT (Message Queue Telemetry Transport) Интернет вещей, его особенности, варианты применения, характерные процедуры. Объясняются различные уровни качества обслуживания в MQTT. Проводится анализ информационных элементов и сообщений. Проведено обоснование выбора уровня обслуживания, который отвечает требованиям надежности в сети и логике применения. Доказано, что объективно парадигма маршрутизации смещается от маршрутизации на транспортном уровне до маршрутизации на прикладном при обслуживании Іot. Показано, что актуальность темы обусловлена стремительным развитием архитектуры "издатель-подписчик", для которой наиболее харакерным является данный протокол.

Ключевые слова: Интернет вещи, IOT, MQTT, издатель-подписчик, брокер, QOS, Quality of Service

Вступ

В даний час, актуальними являються інформаційні технології, які використовують Інтернет речі (Іот). Дешевизна та доступність мікроконтролерів, таких як Arduino та Raspberry Pi, дозволяють їх використати для управління пристроями, що вимірюють дані сенсорів і надсилають їх через Інтернет.

Термін Інтернет речей був вперше використаний Кевіном Ештоном у 2009 році для підключення фізичних пристроїв через Інтернет [1]. Основна ідея дуже проста: фізичні пристрої можуть обмінюватися даними між собою або контролюватися іншими. Прикладами таких пристроїв можуть бути холодильник, автомобіль, будівля та інші.

В основному будь-який типовий електронний пристрій. Один з найпоширеніших випадків використання IoT – це збір, передача, аналіз та відображення даних датчиків. Результати можуть бути представлені веб-інформаційною панеллю з укрупненими значеннями або сигналом тривоги при перевищенні деякого порогу.

Сценарії застосувань IoT майже необмежені. Уявіть, що ваш будильник знає, що ваш поїзд на роботу запізнюється на 15 хвилин, і налаштує себе відповідно. Також ваша кавоварка автоматично включається на 15 хвилин пізніше, щоб приготувати вам гарячу чашку кави перед тим, як поїхати на роботу. Звучить, як майбутнє? Все, що вже сьогодні можливо. Ericsson прогнозує, що в 2020 році 50 мільярдів пристроїв підключено через Інтернет. Зв'язок між величезною кількістю пристроїв забезпечується інтернет-протоколом IPv6 та легкими протоколами зв'язку, такими як MQTT. В приведеній статті аналізуються проблеми застосування протоколу MQTT для збору, передачі, аналізу та відображення даних при використанні Інтернет речей.

Виклад основного матеріалу дослідження

Протокол MQTT був розроблений Енді Стенфордом-Кларком (IBM) та Арленом Ніппером (Eurotech; тепер Cirrus Link) у 1999 році для моніторингу нафтопроводу прокладеного через пустелю. Мета його розробки полягала в тому, щоб створити протокол, який володіє високою пропускною здатністю і використовує мало енергії акумулятора, оскільки пристрої були підключені по супутниковому каналу, і це було надзвичайно дорого на той час [2].

Протокол використовує архітектуру публікації/підписки на відміну від HTTP зі своєю парадигмою запит/відповідь – публікація/підписка, керується подіями та дозволяє передавати повідомлення клієнтам [3, 4]. Центральною точкою зв’язку є брокер MQTT, він відповідає за відправлення всіх повідомлень між відправниками та законними приймачами. Кожен клієнт, який публікує повідомлення брокеру, включає в тему. Тема - інформація про маршрутизацію брокера. Кожен клієнт, який хоче отримувати повідомлення, підписується на певну тему, а брокер доставляє клієнту всі повідомлення з відповідною темою. Тому клієнти не повинні знати один одного, вони спілкуються лише по темі. Ця архітектура дозволяє швидко масштабувати рішення без залежності між виробниками даних та споживачами даних. Архітектура MQTT представлена на рис. 1.

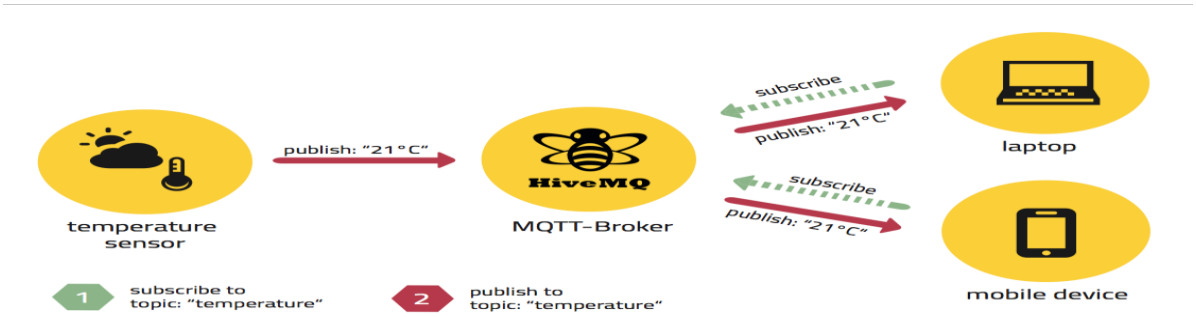

Рис. 1. MQTT архітектура

Відмінність MQTT від HTTP полягає в тому, що клієнту не потрібно перетягувати потрібну йому інформацію, але брокер передає інформацію клієнтові, якщо є щось нове. Тому кожен клієнт MQTT має постійно відкрите TCP-з'єднання з брокером. Якщо це з'єднання переривається будь-якими обставинами, брокер MQTT може зберігати всі повідомлення та надсилати їх клієнту, коли він знову в мережі [5].

Як згадувалося раніше, центральна концепція MQTT для відправки повідомлень – це теми. Тема – це простий рядок, який може мати більше рівнів ієрархії, які розділені косою рискою. Прикладною темою для надсилання даних про температуру вітальні може бути

будинок/вітальня/температура. З одного боку, клієнт може підписатись на точну тему або з іншого боку використовувати підстановку. Підписка на будинок/+/температура призведе до того, що всі повідомлення надсилатимуться раніше згадуваній темі будинок/вітальня/температура, а також будь-яка тема з довільним значенням у місці вітальні, наприклад, температура будинку та кухні/температура. Знак плюс – це однорівнева підказка яка дозволяє лише довільні значення для однієї ієрархії. Якщо вам потрібно підписатися на більш ніж один рівень, наприклад, на все піддерево, також існує багаторівнева підстановка (#). Це дозволяє підписатися на всі основні рівні ієрархії. Наприклад, house/# підписується на всі теми, починаючи з house.

Якість обслуговування (QoS) визначає надійність процесу доставки повідомлень у MQTT. Протокол MQTT надає три рівні QoS для доставки повідомлень: QoS 0, QoS 1 та QoS 2, схема яких приведено на рис. 2. Тому є можливість мати різні рівні QoS для публікації та підписки на повідомлення. Брокер MQTT, який використовується, може не підтримувати всі три рівні QoS. Наприклад, ThingSpeak MQTT підтримує лише QoS 0.

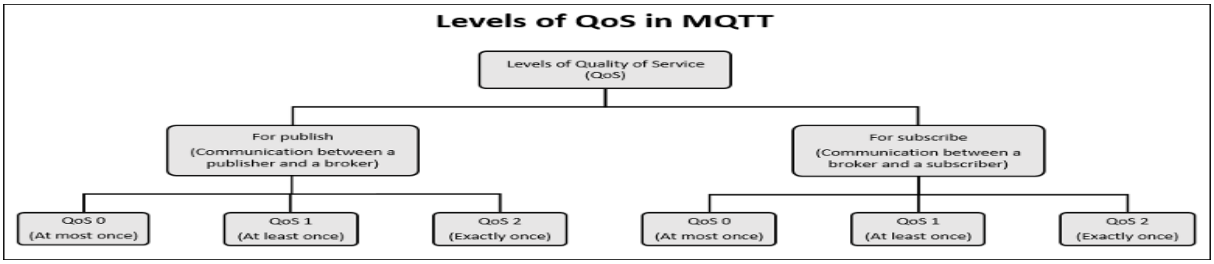

Рис. 2. Рівні QoS в MQTT

Коли мова йде про QoS в MQTT, то потрібно врахувати дві сторони доставки повідомлень:

1. Доставка повідомлень від видавчого клієнта брокеру.
2. Доставка повідомлень від брокера до клієнта, що підписався.

Ми розглянемо обидві сторони доставки повідомлень окремо, оскільки між ними є тонкі відмінності. Клієнт, який публікує повідомлення брокеру, визначає рівень QoS повідомлення, коли він надсилає повідомлення брокеру. Брокер передає це повідомлення клієнтам, що підписалися, використовуючи рівень QoS, який визначає кожен підписуючий клієнт під час підписки. Якщо клієнт, що підписався, визначає нижчий QoS, ніж клієнт-видавець, брокер передає повідомлення з нижчою якістю обслуговування.

QoS - ключова особливість протоколу MQTT. QoS надає клієнту можливість вибору рівня обслуговування, який відповідає його надійності в мережі та логіці застосування. Оскільки MQTT управляє повторною передачею повідомлень і гарантує доставку (навіть коли базовий транспорт не є надійним), QoS значно спрощує спілкування в ненадійних мережах.

Розглянемо докладніше, як реалізовано у протоколі MQTT кожен рівень QoS та як він функціонує.

Мінімальний рівень QoS дорівнює нулю. Цей рівень обслуговування гарантує найкращі зусилля. Гарантії доставки немає. Одержувач не підтверджує отримання повідомлення, і повідомлення не зберігається та повторно передається відправником. Рівень QoS 0 часто називають "передай та забудь" і надає ту саму гарантію, що і базовий протокол TCP.

Схема передачі повідомлення при QoS 0 приведена на рис. 3.

QoS рівень 1 гарантує, що повідомлення буде доставлено щонайменше один раз одержувачу. Відправник зберігає повідомлення, поки не отримає від приймача пакет PUBACK, який підтверджує отримання повідомлення. Можливо, щоб повідомлення було надіслане або доставлено кілька разів.

Схема передачі повідомлення при QoS 1 приведена на рис. 4.

Рис. 3. Схема передачі повідомлення при QoS 0

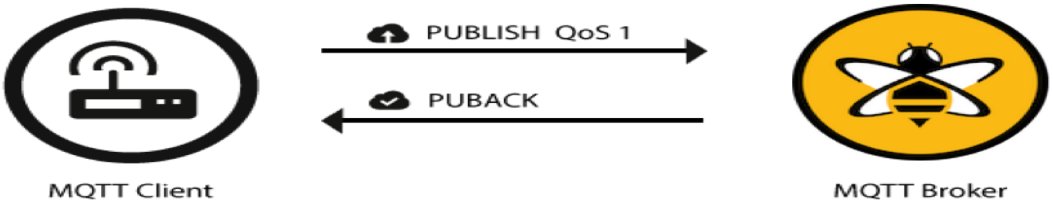

Рис. 4. Схема передачі повідомлення при QoS 1

Відправник використовує ідентифікатор пакету у кожному пакеті, щоб зіставити пакет PUBLISH з відповідним пакетом PUBACK. Якщо відправник не отримує пакет PUBACK у певний проміжок часу, відправник надсилає пакет PUBLACK. Коли одержувач отримує повідомлення з QoS 1, він може негайно обробити його. Наприклад, якщо одержувач є брокером, брокер надсилає повідомлення всім клієнтам, що підписалися, а потім відповідає з пакетом PUBACK.

Якщо клієнт-видавець знову надсилає повідомлення, він встановлює повторювану мітку (DUP). У QoS 1 ця мітка DUP використовується лише для внутрішніх цілей і не обробляється брокером чи клієнтом. Одержувач повідомлення надсилає PUBACK незалежно від мітки DUP.

QoS 2 - це найвищий рівень обслуговування в MQTT. Цей рівень гарантує, що кожне повідомлення буде отримане лише один раз передбачуваними одержувачами. QoS 2 - це найбезпечніший, проте і найповільніший рівень якості обслуговування. Гарантія надається щонайменше двома потоками запиту/відповіді між відправником та одержувачем. Відправник та одержувач використовують ідентифікатор пакета вихідного повідомлення PUBLISH для координації доставки повідомлення.

Схема передачі повідомлення при QoS 2 приведена на рис. 5.

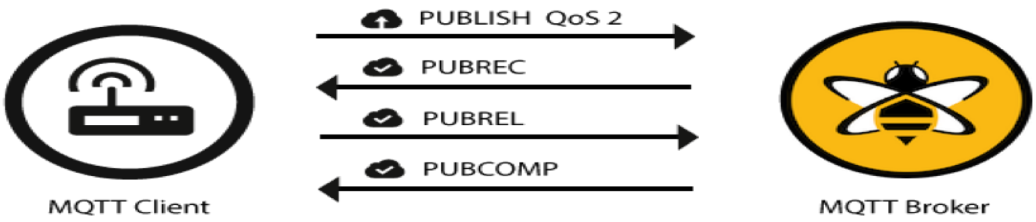

Рис. 5. Схема передачі повідомлення при QoS 2

Коли одержувач отримує від відправника пакет QoS 2 PUBLISH, він обробляє відповідне повідомлення про публікацію та відповідає на відправника пакетом PUBREC, який розпізнає

пакет PUBLISH. Якщо відправник не отримує пакет PUBREC від одержувача, він знову надсилає пакет PUBLISH з дублікатом (DUP), поки він не отримає підтвердження.

Як тільки відправник отримує пакет PUBREC від приймача, відправник може безпечно відкинути початковий пакет PUBLISH. Відправник зберігає пакет PUBREC від приймача і відповідає на PUBREL- пакет.

Після того як одержувач отримує пакет PUBREL, він може відкинути всі збережені стани і відповісти за допомогою пакету PUBCOMP (те саме стосується, коли відправник отримує PUBCOMP). Поки одержувач не завершить обробку і не відправить пакет PUBCOMP назад відправнику, одержувач зберігає посилання на ідентифікатор пакета вихідного пакету PUBLISH. Цей крок являється важливим для того, щоб уникнути обробки повідомлення вдруге. Після того як відправник отримує пакет PUBCOMP, ідентифікатор пакета опублікованого повідомлення стає доступним для повторного використання.

Коли потік QoS 2 завершений, обидві сторони впевнені, що повідомлення доставлено, а відправник має підтвердження доставки.

Якщо пакет втрачається на шляху, відправник зобов'язаний повторно передати повідомлення протягом розумного періоду часу. Це однаково справедливо, якщо відправник є клієнтом MQTT або брокером MQTT. Одержувач несе відповідальність за необхідність надсилати відповідь на кожне командне повідомлення відповідно.

Деякі аспекти застосування QoS являються не дуже очевидними на перший погляд. Ось кілька речей, про які слід пам’ятати, коли використовується QoS:

Пониження якості QoS. Як ми вже згадували, визначення QoS та рівні між клієнтом, який надсилає (публікує) повідомлення, і клієнтом, який отримує повідомлення, - це дві різні речі. Рівні якості цих двох взаємодій також можуть бути різними. Клієнт, який надсилає брокеру повідомлення PUBLISH, визначає QoS повідомлення. Однак, коли брокер доставляє повідомлення одержувачам (абонентам), брокер використовує QoS, який приймач (абонент) визначив під час підписки.

Ідентифікатор пакету, який MQTT використовує для QoS 1 та QoS 2, унікальний між конкретним клієнтом та брокером у межах взаємодії. Цей ідентифікатор не є унікальним для всіх клієнтів. Після завершення потоку ідентифікатор пакету стає доступним для повторного використання. Це повторне використання є причиною того, що ідентифікатор пакету не повинен перевищувати 65535. Нереально, що клієнт може відправити більше, ніж ця кількість повідомлень, не завершуючи взаємодії.

Приведемо кілька рекомендацій, які можуть допомогти у процесі прийняття рішень при виборі рівня обслуговування в MQTT. QoS, який підходить саме для рішення поставленого завдання по передачі даних Іот, суттєво залежить від вимог до якості використання.

Використання QoS 0 доцільно тоді, коли повністю або в основному забезпечено стабільний зв’язок між відправником і одержувачем. Класичний випадок використання для QoS 0 – це підключення тестового клієнта або переднього додатка до брокера MQTT через дротове з'єднання.

При цьому допускається випадок, коли періодично губляться кілька повідомлень. Втрата деяких повідомлень може бути прийнятною, якщо дані не так важливі або коли дані надсилаються через короткі проміжки часу

Також не висуваються вимоги для збереження черги з повідомленнями. Повідомлення стоять у черзі лише для відключених клієнтів, якщо вони мають QoS 1 або 2 та постійний сеанс .

Рекомендується використовувати QoS 1, коли потрібно отримувати кожне повідомлення. При цьому випадку використання необхідно враховувати можливість обробляти дублікати. Рівень QoS 1 - це найпоширеніший рівень обслуговування, оскільки він гарантує, що повідомлення надходить принаймні один раз, але дозволяє проводити багаторазові доставки. При цьому, дана заявка повинна адекватно реагувати на дублікати та мати можливість їх відповідно обробляти.

Використання QoS 2 обґрунтовано тоді, коли для робочої програми дуже важливо отримувати всі повідомлення рівно один раз. Це часто трапляється, якщо повторна доставка може завдати шкоди користувачам додатків або передплатним клієнтам. Тому необхідно

враховувати збільшення накладних витрат і часу на передачу даних при використанні QoS 2 для взаємодії IoT.

При передачі даних IoT усі повідомлення, надіслані за допомогою QoS 1 і QoS 2, ставлять у чергу для офлайн-клієнтів, поки клієнт знову не з'явиться. Однак ця черга можлива лише за умови постійного сеансу роботи клієнта.

Висновки

Таким чином, проведений у статті аналіз умов застосування протоколу MQTT для збору, передачі, аналізу та відображення даних при використанні IoT дозволив зробити наступні висновки.

1. Мінімальний об'єм службової інформації, наявність класів обслуговування і ієрархічна структура являються безперечними перевагами протоколу MQTT, що підтверджується великою різноманітністю як клієнтського, так і серверного програмного забезпечення, у тому числі відкритого програмного забезпечення.

2. Проте застосування у формі "видавець-підписчик" висуває вимоги до нового мережевого об'єкту – брокера, який по суті забезпечує маршрутизацію призначеної для користувача інформації.

3. Проведено аналіз та обґрунтовано вибір рівня обслуговування, який відповідає вимогам надійності в мережі та логіці застосування.

4. Таким чином, об'єктивно парадигма маршрутизації зміщується від маршрутизації на транспортному рівні до маршрутизації на прикладному при обслуговувані IoT.

Список використаної літератури

1. В. Гойхман. А. Савельева. Аналитический обзор протоколов Интернета вещей // Технологии и средства связи. – 2016. – № 4. С. 32–37.

2. «IBM Podcast: Piper, Diaz, Nipper – MQTT» [Електронний ресурс]: [Веб-сайт]. – Електронні дані, – Режим доступу: http://www.ibm.com/podcasts/software/ websphere/connectivity/piper_diaz_nipper_mq_tt_1118201 1.pdf. (дата звернення: 10.10.2019).

3. «Web-хостинг GitHub, раздел MQTT – libraries.» [Електронний ресурс] : [Веб-сайт]. – Електронні дані, – Режим доступу: https://github.com/mqtt/mqtt.github.io/wiki/libraries. (дата звернення: 11.10.2019).

4. «An Open Source MQTT v3.1/v3.1.1 Broker.» [Електронний ресурс] : [Веб-сайт]. – Електронні дані, – Режим доступу: https://mosquitto.org. (дата звернення: 11.10.2019).

5. «HiveMQ – Enterprise MQTT Broker.» [Електронний ресурс] : [Веб-сайт]. – Електронні дані, – Режим доступу: http://www.hivemq.com. (дата звернення: 10.10.2019).

Автори статті

Звенігородський Олександр Сергійович – кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук, Державний університет телекомунікацій, Київ, Україна.

Кутовий Сергій Олександрович– студент, Державний університет телекомунікацій, Київ, Україна.

Прокопов Сергій Васильович – кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук, Державний університет телекомунікацій, Київ, Україна.

Рижаков Микола Миколайович – аспірант, асистент кафедри комп'ютерних наук, Державний університет телекомунікацій, Київ, Україна.

Authors of the article

Zvenihorods′kyy Oleksandr Serhiyovych – candidate of Sciences (technical), associate professor, associate professor of Computer Science Department, State University of Telecommunications, Kyiv, Ukraine.

Kutovyy Serhiy Oleksandrovych – студент, Державний університет телекомунікацій, Київ, Україна.

Prokopov Serhiy Vasyl′ovych – candidate of Sciences (technical), associate professor, associate professor of Computer Science Department, State University of Telecommunications, Kyiv, Ukraine.

Ryzhakov Mykola Mykolayovych – postgraduate student, State University of Telecommunications, Kyiv, Ukraine.

Дата надходження в редакцію: 11.10.2019 р. Рецензент: д.т.н., с.н.с. М.П. Трембовецький