УДК 621.39

ПРОТОКОЛИ РІВНЯ ЗАСТОСУНКІВ В МЕРЕЖАХ IoT

Григоренко О.Г., Дідковська Н.А.
Навчально-науковий інститут телекомунікаційних систем КПІ ім. Ігоря Сікорського, Україна
E-mail: olenagri@ukr.net, didkovska.natalia@lll.kpi.ua

APPLICATION LAYER PROTOCOLS IN IoT NETWORKS

The main application protocols in the Internet of Things networks are analyzed and their features are defined, which should meet the functional requirements and basic characteristics of IoT devices that exchange information.

Проаналізовані основні протоколи рівня застосунків в мережах Інтернету речей та зазначені їх особливості, що повинні відповідати вимогам функціонування і основним характеристикам пристроїв IoT, які обмінюються інформацією.

Останнім часом кількість пристроїв Інтернету Речей (IoT) збільшується з вражаючою швидкістю, що відкриває безліч перспектив для подальшого розвитку людства. Це потребує ефективних протоколів на рівні застосунків, сервісів, інфраструктури, що відповідають потребам пристроїв IoT, які характеризуються обмеженими ресурсами: мале споживання енергії, малий обсяг пам'яті та трафіку, що генерується, простота та інші. Дану мережу можна порівняти з лабіринтом, адже Інтернет речей – це велика кількість протоколів, які визначають, як пристрої спілкуються, обмінюються даними і взаємодіють у мережі. Протоколи рівня застосунків в мережах IoT визначають способи, за якими пристрої збирають, обробляють та передають дані, відкриваючи безмежні можливості для покращення нашого повсякденного життя та розвитку промисловості. Нижче проаналізовані особливості та характеристики протоколів рівня застосунків, що найчастіше використовуються у Інтернеті речей, зазначені їх переваги та недоліки, проведено порівняння.

MQTT (Message Queuing Telemetry Transport) — це відкритий стандартний протокол обміну повідомленнями, реалізований OASIS і рекомендований Міжнародною організацією зі стандартизації (ISO/IEC 20922). Він розроблений як надзвичайно легкий засіб для обміну повідомленнями за допомогою публікації/підписки на сервері-брокері в мережах M2M (machine-to-machine), ідеально підходить для підключення віддалених пристроїв з невеликим кодовим простором і мінімальною пропускною здатністю мережі, що відповідає характеристикам пристроїв IoT. Сьогодні MQTT використовується в найрізноманітніших галузях, таких як автомобілебудування, виробництво, телекомунікації, нафтогазова промисловість тощо. Переваги даного протоколу включають в себе [1]:

1. MQTT дозволяє обмінюватися повідомленнями між пристроєм у хмару та з хмари на пристрій. Це спрощує трансляцію повідомлень групам речей.
2. Масштабування MQTT для підключення до мільйонів пристроїв IoT.

3. Надійна доставка повідомлень.

4. Підтримка трьох класів якості обслуговування QoS.

5. Підтримка ненадійних мереж: Багато пристроїв IoT підключаються через ненадійні стільникові мережі. Підтримка MQTT для постійних сесій скорочує час на повторне з'єднання клієнта з брокером.

6. Безпека протоколу: MQTT дозволяє легко шифрувати повідомлення за допомогою TLS і автентифікувати клієнтів за допомогою сучасних протоколів автентифікації, таких як OAuth. Є можливість налаштування процесу відстеження всіх станів підключення клієнта, включаючи облікові дані безпеки та сертифікати.

В якості недоліків можна відмітити, що безпека протоколу була скомпроментована у 2020 р. [6] (CVE-2020-13849). Також наявність брокера, через який відбувається обмін повідомленнями, є “слабким” місцем мережі при перериванні обслуговування. Проте вбудована черга брокерів MQTT може в такій ситуації забезпечити буферизацію для обмежених пристроїв IoT, які не мають можливості зробити це самостійно.

HTTP (Hypertext Transfer Protocol) - протокол передачі гіпертексту є основою Інтернету і використовується для завантаження веб-сторінок за допомогою гіпертекстових посилань [2]. Призначений для передачі інформації між мережевими пристроями на прикладному рівні моделі OSI та працює поверх інших рівнів стека TCP/IP. Типовий потік через HTTP передбачає, що клієнтська машина робить запит до сервера, який потім надсилає повідомлення у відповідь. HTTP, як правило, розроблений таким чином, щоб бути простим і зручним для читання людиною, навіть з урахуванням додаткової складності, яка притаманна HTTP/2, шляхом інкапсуляції повідомлень у фрейми. Зрозумілість HTTP-повідомлень полегшує тестування для розробників і знижує складність для новачків. З'єднання контролюється на транспортному рівні, і тому принципово виходить за рамки HTTP. Він не вимагає, щоб базовий транспортний протокол був заснований на з'єднанні, лише щоб був надійним, або не втрачав повідомлення (як мінімум, представляючи помилку в таких випадках). Тому HTTP працює поверх TCP. [3]

CoAP (The Constrained Application Protocol) — це легкий протокол, призначений для використання з пристроями з низьким енергоспоживанням, обчислювальною потужністю, пам'яттю та часом автономної роботи,такими як пристрої IoT, і обмеженими пропускною здатністю мережами. Визначає протокол веб-передачі на основі репрезентативного стану (REST) на додаток до функцій HTTP, це надає простіший спосіб обміну даними між клієнтами та серверами через HTTP. CoAP спирається на архітектуру клієнт-сервер без збереження стану. Він використовується в мобільних додатках і програмах соціальних мереж і усуває неоднозначність за допомогою методів HTTP get, post, put і delete. CoAP надає набір методів для виявлення, маніпулювання та моніторингу ресурсів, а також підтримку асинхронного зв'язку та кешування. CoAP працює поверх протоколу UDP і зазвичай використовує порт 5683 для незахищеного зв'язку та порт 5684 для безпечного зв'язку за допомогою DTLS

(Datagram Transport Layer Security). [4]

Основними перевагами протоколу є:

- зменшення накладних витрат на зв’язок за рахунок поблочного транспортування ресурсів, коли при обміні даними між клієнтом і сервером непотрібно оновлення всіх даних,
- гнучкість взаємодії з HTTP через проксі,
- безпека на основі DTLS.

Служба розподілу даних (DDS-Data Distribution Service) - це протокол проміжного програмного забезпечення (middleware) та стандарт API для зв'язку. Використовує механізм публікації/підписки для комунікацій M2M у реальному часі. Він об'єднує компоненти системи разом, забезпечуючи низьку затримку передачі даних, надзвичайну надійність, розширену безпеку та масштабовану архітектуру, необхідну для бізнес-додатків та критично важливих додатків Інтернету речей [5]. На відміну від інших протоколів публікації/підписки, таких як MQTT, DDS покладається на архітектуру без посередників і використовує багатоадресну розсилку, щоб забезпечити відмінну якість обслуговування (QoS) і високу надійність своїх програм. Його архітектура публікації/підписки без брокерів добре відповідає обмеженням реального часу для комунікацій IoT та M2M. DDS підтримує 23 політики QoS, за допомогою яких розробник може врахувати різноманітні критерії зв’язку, такі як безпека, терміновість, пріоритет, довговічність, надійність тощо [7].

Таблиця 1. Порівняння протоколів.

	MQTT	CoAP	DDS	HTTP
Клієнт-серверна архітектура	так	так	так	так
Механізм публікації/підписки	так	так	так	ні
Запит/відповідь	ні	так	ні	так
Транспортний протокол	TCP	UDP	TCP/ UDP	TCP
Безпека	SSL	DTLS	SSL/DTLS	SSL
QoS	так	так	так	ні
Розмір заголовку (байт)	2	4	-	-

Підсумовуючи можна сказати, що проаналізовані протоколи відповідають вимогам обмеження ресурсів в пристроях і мережах IoT і знаходять застосування в різноманітних сценаріях і середовищах.

Література

1. Головний сайт MQTT https://mqtt.org/
2. https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/
3. https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview
4. https://cqr.company/ru/wiki/protocols/constrained-application-protocol-coap/
5. https://www.twinoakscomputing.com/datasheets/DDS-Brochure.pdf
6. Vaccari, I., Aiello, M., & Cambiaso, E. (2020). SlowITe, a novel denial of service attack affecting MQTT. Sensors, 20(10), 2932.
7. http://www.omg.org/