

ЕКСПЕРИМЕНТАЛЬНИЙ АНАЛІЗ ЗАТРИМОК ТА ПРОДУКТИВНОСТІ ПРОТОКОЛІВ ПЕРЕДАЧІ ДАНИХ У CLOUD–EDGE СИСТЕМАХ

¹ Львівський національний університет імені Івана Франка, факультет електроніки та комп'ютерних технологій, кафедра радіофізики та комп'ютерних технологій, Львів, Україна

Анотація

Досліджено характеристики затримок і стабільності протоколів передавання даних у сценарії «cloud → edge» для потреб онлайн-лабораторій та віддалених експериментальних установок. Порівняно gRPC (unary), gRPC bidirectional streaming, HTTP, MQTT, NATS і CoAP за круговою затримкою (RTT), перцентиллями p50/p95/p99, втратами повідомлень та використанням процесора на граничному вузлі. Експерименти виконано на Raspberry Pi Zero 2 W (DietPi) у Wi-Fi мережі 2,4 ГГц із клієнтом на ПК. Отримані статистичні профілі протоколів використано як вхідні параметри для адаптивного вибору протоколу в протокольному шарі системи.

Ключові слова: edge computing, cloud–edge, онлайн-лабораторії, протоколи зв'язку, кругова затримка, gRPC, MQTT, CoAP.

Abstract

This work evaluates latency and reliability of data transmission protocols in a cloud-to-edge setting relevant to online laboratories. We benchmark gRPC (unary), gRPC bidirectional streaming, HTTP, MQTT, NATS, and CoAP using round-trip time (RTT) with p50/p95/p99 percentiles, message loss, and edge CPU utilization. The edge endpoint is a Raspberry Pi Zero 2 W (DietPi) connected via 2.4 GHz Wi-Fi; the benchmark client runs on a PC. The resulting statistical profiles are then used as inputs for an adaptive protocol-selection mechanism.

Keywords: edge computing, cloud–edge, online laboratories, communication protocols, RTT, gRPC, MQTT, CoAP.

Вступ

Онлайн-лабораторії та віддалені експериментальні установки працюють у розподіленому середовищі, де хмарні сервіси поєднуються з граничними (edge) вузлами [2]. Для таких систем характерні різні потоки даних — керуючі команди, телеметрія та результати експериментів — які мають різні вимоги до затримки, надійності та пропускної здатності. Тому практично важливо мати адаптивний протокольний шар, який обирає протокол з урахуванням QoS та обмежень ресурсів edge-вузлів [1]

Метою роботи є експериментально оцінити поведінку поширених протоколів передавання даних у реальному cloud–edge сценарії та отримати статистичні профілі (RTT, хвостова затримка, втрати), придатні як вхідні дані для алгоритмів адаптивного вибору протоколу.

Методика експерименту

Оцінювання виконано у сценарії «клієнт (cloud) → сервіс (edge)» для протоколів gRPC (unary), gRPC bidirectional streaming, HTTP, MQTT, NATS та CoAP [3]. Для кожної комбінації швидкості надсилання повідомлень і розміру корисного навантаження клієнт надсилав повідомлення з унікальними ідентифікаторами та очікував відповіді з тим самим ідентифікатором; кругову затримку (RTT) вимірювали на стороні клієнта як час між надсиланням і отриманням відповіді.

Граничним вузлом був Raspberry Pi Zero 2 W під керуванням DietPi, підключений через Wi-Fi 2,4 ГГц; клієнтський застосунок і бенчмарк виконувались на персональному комп'ютері. Окрім середнього RTT, обчислювалися перцентилі p50/p95/p99, а втрати визначалися як різниця між кількістю надісланих і отриманих відповідей. Додатково фіксувалось середнє використання процесора на edge-вузлі.

Результати та обговорення

Експериментальні дослідження, виконані на платформі Raspberry Pi Zero 2 W, показали, що

ефективність протоколів передачі даних істотно залежить від розміру повідомлення та інтенсивності навантаження (рис. 1). Для малих payload (256–1024 Б) найменші значення середньої затримки забезпечували gRPC bidirectional streaming, NATS та gRPC, тоді як MQTT характеризувався стабільно вищою затримкою, а CoAP — зростанням нестабільності зі збільшенням частоти передавання. Для payload 4100 Б найкращі показники затримки зберігали NATS, gRPC-bidi та gRPC, тоді як HTTP і MQTT починали деградувати за високої інтенсивності, а CoAP втратив працездатність уже в усьому цьому діапазоні навантажень. Для великих повідомлень (8200 Б) у всіх протоколів спостерігалось різке зростання затримки та втрат, що свідчить про наближення до меж продуктивності обчислювальної платформи; водночас відносно кращу стійкість у цій зоні демонстрували gRPC-bidi при 100 пов./с та NATS у режимах підвищеного навантаження. Як видно з рис. 1, жоден із досліджених протоколів не є однаково ефективним у всьому робочому діапазоні, а отже адаптивний вибір протоколу може бути доцільним насамперед у змінних умовах функціонування, коли параметри навантаження не є наперед фіксованими і потребують динамічного узгодження з поточним станом системи.

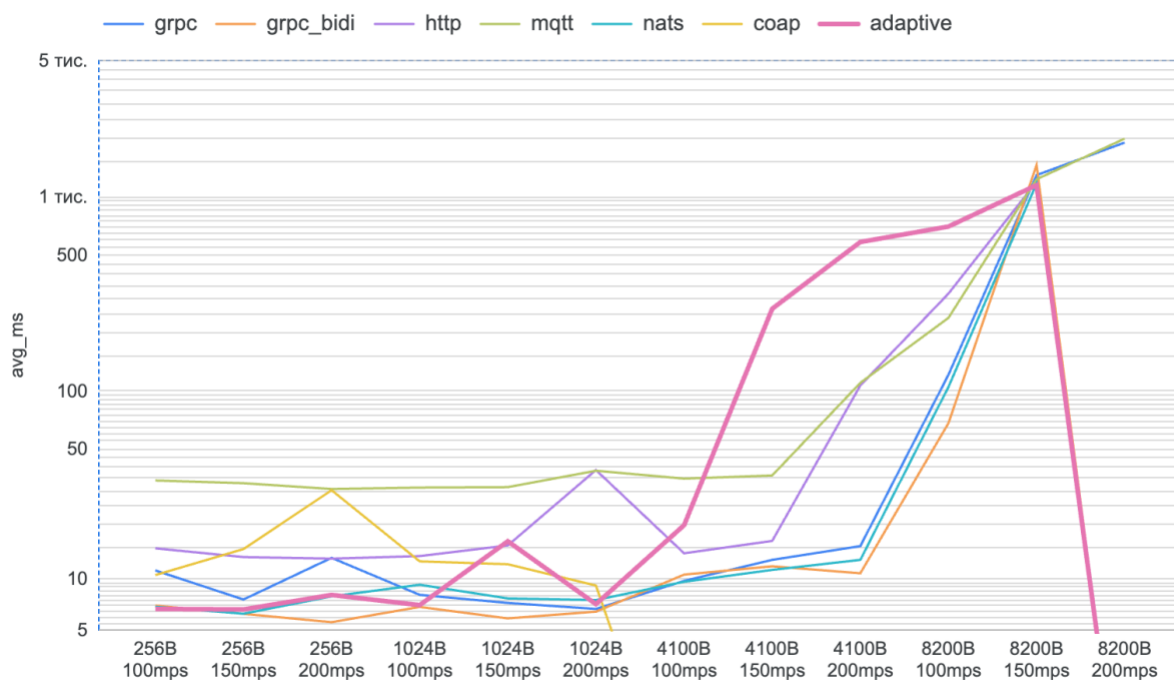


Рис. 1. Порівняння середньої затримки досліджуваних протоколів залежно від розміру повідомлення та інтенсивності навантаження на Raspberry Pi Zero 2 W

Отримані емпіричні оцінки RTT, перцентилів та втрат формують компактний статистичний профіль кожного протоколу й режиму навантаження. Такий профіль може бути використаний у вартісній функції алгоритму адаптивного вибору протоколу, який мінімізує затримку та втрати за обмежень edge-вузлів. Як наступний крок розглядається перехід від евристичного підходу до контекстного бандита.

Висновки

Результати дослідження підтверджують доцільність адаптивного вибору протоколу в edge/IoT-системах за змінного навантаження та різних розмірів повідомлень, оскільки жоден із протестованих протоколів не є однаково ефективним у всьому робочому діапазоні. CoAP найкраще працює з малими повідомленнями, MQTT є доцільним для великих повідомлень за вимоги мінімізації втрат, gRPC bidi забезпечує найменшу затримку для малих payload, а NATS виявляється найефективнішим для великих пакетів і burst-навантажень. Водночас для однорідних і наперед відомих сценаріїв статичний вибір протоколу є кращим за адаптивний, оскільки не потребує етапу навчання і не спричиняє додаткових втрат. Отже, адаптивний підхід найбільш виправданий у динамічних середовищах, тоді як у стабільних умовах доцільніше використовувати попередньо відібраний протокол.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Nazarevych, R., & Bolesta, I. (2025). Software of Internet-accessible semiconductor laboratory. *Information Systems and Networks*, 17, 146–159. <https://doi.org/10.23939/sisn2025.17.146>
2. Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30–39. <https://doi.org/10.1109/MC.2017.9>
3. Thangavel, D., Ma, X., Valera, A., Tan, H. X., & Tan, C. K. Y. (2014). Performance evaluation of MQTT and CoAP via a common middleware. In *Proceedings of the 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)* (pp. 1–6). <https://doi.org/10.1109/ISSNIP.2014.6827678>

Назаревич Роман Андрійович — аспірант, кафедра радіофізики та комп'ютерних технологій, Львівський національний університет імені Івана Франка, Львів, e-mail: roman.nazarevych@gmail.com , roman.nazarevych@lnu.edu.ua

Науковий керівник: **Болеста Іван Михайлович** — д-р фізико-математичних наук, професор, Львівський національний університет імені Івана Франка, м. Львів

Nazarevych Roman A. — PhD student, Department of Radiophysics and Computer Technologies, Ivan Franko National University of Lviv, Lviv, email: roman.nazarevych@gmail.com , roman.nazarevych@lnu.edu.ua

Supervisor: **Bolesta Ivan M.** — Doctor of Physical and Mathematical Sciences, Professor, Ivan Franko National University of Lviv, Lviv.