УДК 004.7 DOI: 10.31673/2786-8362.2026.012589

Яковець В.П.; Гоббязов А.С.;
Стародубцев Я.О.

МЕТОДИ ПОБУДОВИ МУЛЬТИПРОТОКОЛЬНОЇ ІНФРАСТРУКТУРИ ІНТЕРНЕТУ РЕЧЕЙ

Yakovets V.P., Hobbiazov A.S., Starodubtsev Ya.O. Methods for building a multi-protocol Internet of Things infrastructure. The article discusses approaches to integrating heterogeneous sensor networks into multi-protocol information systems (MPIS). Methods of integration at three levels are analyzed: sensors, gateways, and application level. It proposes a classification of architectural patterns, QoS mapping rules, and mechanisms for maintaining end-to-end security during transmission between TLS and DTLS, as well as recommendations for placing functions in Edge, Fog, and Cloud to optimize latency and energy efficiency. The analysis concludes that combined solutions provide expanded functionality and increased network resilience when using modular gateways, adapters, and a registry of schemes. The article formulates practical recommendations for designing MPIs, defines criteria for selecting technology combinations, and outlines directions for further research in validating QoS mappings and energy policies when integrating IoT networks into a unified multi-protocol infrastructure.

Keywords: Internet of Things, multi-protocol networks, sensor networks, Edge/Fog architecture, CoAP, MQTT

Яковець В.П., Гоббязов А.С., Стародубцев Я.О. Методи побудови мультипротокольної інфраструктури Інтернету речей. У статті розглянуто підходи до інтеграції гетерогенних сенсорних мереж у мультипротокольні інформаційні системи (МПС). Проаналізовано методи об'єднання на трьох рівнях: сенсорів, шлюзів та прикладного рівня. Запропоновано класифікацію архітектурних патернів, правила мапінгу QoS і механізми збереження наскрізної безпеки при трансляції між TLS та DTLS, а також рекомендації щодо розміщення функцій в Edge, Fog та Cloud для оптимізації затримки та енергоефективності. В результаті аналізу виведено, що комбіновані рішення забезпечують розширення функціональності та підвищення стійкості мережі за умови застосування модульних шлюзів, адаптерів і реєстру схем. Стаття формулює практичні рекомендації для проєктування МПІ, визначає критерії вибору композиції технологій та окреслює напрями подальших досліджень у валідації QoS-мапінгів і енергетичних політик при інтеграції IoT-мереж в єдину мультипротокольну інфраструктуру.

Ключові слова: Інтернет речей, мультипротокольні мережі, сенсорні мережі, Edge/Fog-архітектура, CoAP, MQTT

Вступ

Архітектура мереж Інтернету речей традиційно розбивається на чотири функціональні рівні: (i) рівень «розумних» об'єктів із сенсорами; (ii) рівень шлюзів і зовнішніх глобальних мереж (зокрема Internet/Fog-зв'язок); (iii) сервісний (Information/Service) рівень; та (iv) рівень прикладних рішень для галузей (енергетика, транспорт, медицина тощо). Цю чотирирівневу модель широко використовують для опису інтеграції IoT, Fog і Cloud систем, оскільки вона відображає розподіл ресурсів і ролей від обмежених в обчислювальній потужності пристроїв до потужних хмарних сервісів.

Нижній рівень, сенсори та «Smart» пристрої, виконує ключову функцію зв'язку фізичного і цифрового рівнів: збору вимірів, первинної обробки та їхньої локальної експозиції у вигляді URI, теми або контент-імені. У практичних реалізаціях сенсорні вузли публікують дані в локальні брокери або безпосередньо до шлюзів.

Другий рівень (шлюзи і конвергентна мережева інфраструктура) функціонує як місце інтеграції гетерогенних мереж і протоколів: тут виконуються протокольні мапінги (наприклад HTTP до CoAP), брокерні мости (MQTT до AMQP або локальний рівень до хмарного MQTT), семантична нормалізація та Edge-обробка. Різні протоколи оптимально розміщуються у різних підсистемах (пристрої – легкі UDP/CoAP або MQTT-SN; Fog – брокери та адаптери; Cloud – TCP-орієнтовані сервіси), що зменшує накладні витрати та покращує відмовостійкість системи.

Сервісний і прикладний рівні реалізують інформаційні послуги, аналітику, моніторинг і бізнес-логіку. У проєктуванні цих шарів важливо приховати від додатків гетерогенність нижчих шарів через уніфікований API або семантичний Middleware. Вибір протоколу на рівні додатків істотно впливає на параметри передачі: CoAP зазвичай показує нижчі затримки та менше споживання пропускної здатності та енергії для невеликих корисних навантажень, тоді як TCP-орієнтовані протоколи (HTTP, MQTT, DDS) можуть давати кращу поведінку при більших навантаженнях або в умовах високої завантаженості мережі – отже, оптимальний вибір залежить від сценарію застосування.

Аналіз характеристик сенсорних мереж дозволяє сформулювати базовий набір параметрів для проєктування: діапазон використовуваних частот, швидкість передачі даних, структура пакетів (і потреба у фрагментації), методи ідентифікації/реєстрації пристроїв і авторизації, механізми захисту даних, дальність дії, енергоефективність, методи модуляції та політики управління частотним ресурсом і ущільнення каналів [1]. Важливу роль у визначенні обсягу контрольного трафіку, розміру FIB/PIT/таблиць маршрутизації та оперативної пам'яті вузлів грають топологія (однострибкова чи багатострибкова) і стратегія ретрансляції (наскрізна чи «крок-за-кроком»). Зокрема, дослідження порівняння NDN, CoAP і MQTT-SN показують відмінності в потребах по пам'яті, уникненні фрагментації та схемах іменування, що впливає на архітектуру сенсорних мереж.

Аналіз останніх досліджень. В [1] пропонується та оцінюється мультипротокольна система безпеки для хмарних IoT-архітектур (поєднання TLS, DTLS і MQTT), яка підвищує захист, масштабованість та ресурсну ефективність порівняно з однопротокольними рішеннями. В [2] пропонується IoT-фреймворк для інтелектуального моніторингу та керування будівлями – інтегруючи сенсорні мережі, шари зв'язку та інтелектуальні рішення (SVM, Q-learning, Deep reinforcement learning, DNN), підтверджує переваги DNN (точність 85,2%) у прогнозуванні та динамічному регулюванні (зниження енерговитрат >30%) і рекомендується подальша інтеграція Edge-обчислень і розподіленої архітектури для підвищення реального часу та масштабованості. В [3] пропонується модульна архітектура IoT-шлюзу MIGS із чотирма компонентами (Management, Southbound, Northbound, Cache), де Southbound використовує ізоляцію інстансів і незалежні потоки для роботи з гетерогенними протоколами (Modbus, MQTT, OPC UA). Cache відділяє збір даних від передачі для збереження цілісності при мережевих затримках, а веб-модуль забезпечує runtime-управління; автори також формалізують модель затримок (враховуючи вплив Python GIL і накладні витрати серіалізації) і теоретично та функціонально підтверджують ефективність рішення для паралельної багатопротокольної комунікації, надійної передачі даних і масштабованості. В [4] описується розробка та впровадження розподіленої IoT-системи розумного паркування з ультразвуковими та інфрачервоними сенсорами на базі NodeMCU ESP8266, передачею даних через MQTT до AWS IoT Core з зберіганням у NoSQL і обробкою LSTM-моделями для прогнозування попиту та оптимального розподілу, із RESTful API для клієнтських додатків, адмін-дашбордами, механізмами кібербезпеки (TLS 1.2, RBAC) і модульною масштабованістю для інтеграції в інфраструктуру Smart City. В [5] оцінюється п'ять класифікаторів (RF, лінійний і RBF SVM, CNN, CNN-LSTM) на наборі даних MQTT-IoT-IDS2020 для виявлення атак у середовищі IoT, показується лідерство RF (99,9% точності) і пропонується XMID-MQTT – підхід із застосуванням XAI (SHAP, LIME) для інтерпретованого пояснення рішень моделей та їх безпечнішої інтеграції в IoT-IDS. В [6] пропонується енергоефективний фреймворк для інтеграції блокчейну з MEC в IoT: MCHID для багатострибкового відвантаження, параметризований MDP для спільної оптимізації та HTRPO для навчання гібридних дій, що знижують енергоспоживання і накладні витрати. В [7] досліджуються проблеми багатодоменної сумісності в IoT-шлюзах і аналізується Middleware, використання блокчейну для підвищення безпеки, протоколи MQTT і CoAP, а також роль Fog computing і Edge computing у зниженні затримок. В [8] пропонується протокол PPAP для квантово-стійкої та приватної автентифікації в децентралізованих системах з практичною ефективністю, низькими накладними витратами та оптимізацією пошуку доступу через TAPM. В [9] пропонується

легкий багатострибковий протокол маршрутизації LMRP для мереж краю на основі Power Line Communication, реалізований у масштабованій багаторівневій архітектурі, що демонструє мінімум 32.62% зниження втрат шляху і 76.3% приріст ефективності маршрутизації.

Постановка завдання. Дослідити та формалізувати підходи до інтеграції гетерогенних сенсорних мереж у мультипротокольні інформаційні системи – визначити архітектурні патерни, правила трансляції протоколів і політики мапінгу сервісних показників.

Метою роботи є розробка практичних рекомендацій для побудови мультипротокольної інфраструктури (МПІ), яка забезпечує інтероперабельність монопротокольних сенсорних мереж на рівнях сенсорів, шлюзів і додатків, з урахуванням вимог до затримки, надійності, енергоефективності та безпеки.

Виклад основного матеріалу дослідження

Сучасні стандарти безпроводових сенсорних мереж реалізують різні діапазони значень базових параметрів – частотний спектр, швидкість передавання, радіус покриття, енергоспоживання та протокольні механізми управління доступом. Ці характеристики не розподілені рівномірно між стеками: наприклад, протоколи, орієнтовані на канали великої дальності і низьку пропускну здатність (типу LoRaWAN), навмисне знижують швидкість і підвищують дальність, тоді як IEEE 802.15.4-сумісні стеки (наприклад, ZigBee) працюють у вищих частотних діапазонах з більшими швидкостями та зручніші для Mesh-топологій з багатострибковою маршрутизацією. Наслідком цих відмінностей є специфічні вимоги до фрагментації, таблиць маршрутизації та енергетичного профілю вузлів [2].

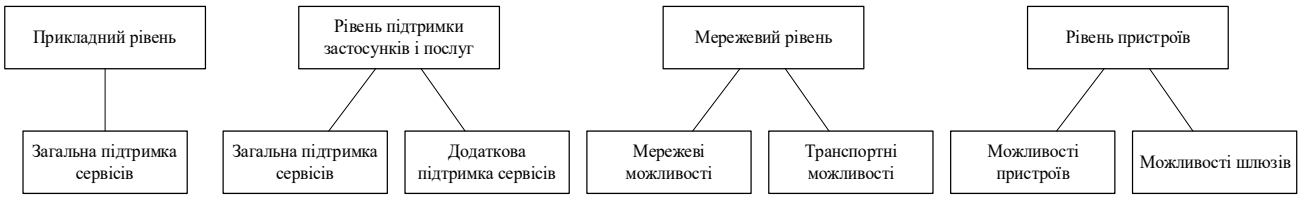

Рис. 1. Типова модель Інтернету речей

Порівняльний аналіз реалізацій показує, що для кожного стеку взаємодії існує свій характерний набір значень базових показників – тобто кожен протокол «займає» певну область у просторі параметрів (частота, швидкість, дальність та енергія). На практиці це призводить до того, що багато розгортань сенсорних мереж є монопротокольними: вибір стандарту на етапі проєктування визначає експлуатаційні властивості та можливі сценарії застосування мережі. Така картина підтверджується як систематичними оглядами протоколів, так і вимірювальними дослідженнями на тест-бенчах.

У типовій монопротокольній топології роль серверної компоненти (серверу застосунків) зосереджує функції сервісного та прикладного рівнів: централізована або розподілена обробка та зберігання даних, управління збиранням інформації від сенсорів, надання інтерфейсів користувачам і зовнішнім застосункам, а також виконання аналітики та оркестрації процесів. Саме на цьому рівні реалізуються політики агрегації, нормалізації форматів і гарантування доступу до даних, тобто саме сервер застосунків часто «маскує» від кінцевих користувачів гетерогенність нижчих шарів [3].

Розширити експлуатаційні характеристики і сферу застосування IoT-систем можна шляхом поєднання кількох монопротокольних мереж (МПМ) у мультипротокольну інфраструктуру (МПІ). Практичні шляхи такої еволюції включають: інтеграцію через Fog-шлюзи та проксі (протокольна трансляція між HTTP та CoAP, брокерні мости для MQTT та AMQP), семантичну нормалізацію на Middleware-рівні та локальну Edge-обробку для зниження трафіку до хмари. Комбінування МПМ у МПІ дає змогу зберегти переваги окремих стеків (дальність LoRaWAN, низьку затримку IEEE 802.15.4-mesh тощо) та розширити функціонал мережі за рахунок реплікації, кешування та агрегації на рівні шлюзів і сервісів.

Корисність мультипротокольних рішень залежить від топології та режиму роботи: у однострибкових сценаріях IP-орієнтовані протоколи часто забезпечують нижчі накладні витрати і вищу швидкість, натомість у багатохопових, нестійких середовищах підходи з кешуванням і відновленням сигналу з урахуванням переприйомів (наприклад ICN/NDN-парадигма) демонструють кращу стійкість і енергоефективність. Відтак перехід від МПМ до МПІ повинен супроводжуватись проектуванням правил мапінгу QoS, схем аутентифікації та політик енергетичного менеджменту між шарами.

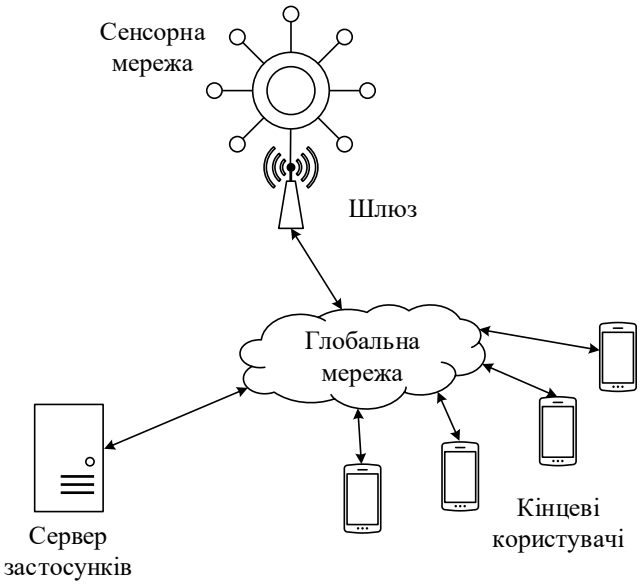

Рис. 2. Топологія монопротокольної мережі IoT

Інтеграція IoT-мереж на прикладному рівні. Архітектура інтеграції на рівні додатків розглядається як один із вищих рівнів об'єднання мультипротокольних сенсорних підмереж, при якому незалежні сенсорні мережі передають свої потоки даних через відповідні шлюзи у центральний мультипротокольний сервер застосунків, що виступає одночасно точкою збору, нормалізації та експозиції даних для кінцевих користувачів і сервісів.

Топологічно така інтеграція має просту структуру: кілька монопротокольних сенсорних мереж пов'язані зі своїми шлюзами, а ті – з єдиним мультипротокольним сервером застосунків, який функціонує як головний «об'єднуючий» елемент. У спрощених схемах опускають проміжні глобальні мережі, проте на практиці роль проміжних шарів (Fog) і мережева інфраструктура впливають на вибір протоколів і на продуктивність кінцевого рішення.

Перевага підходу на рівні додатків полягає в мінімальній потребі змін у вже розгорнутих апаратно-програмних компонентах сенсорів і шлюзів: на їхньому боці зазвичай не потрібно оновлювати прошивку чи прапори мережевого стеку, оскільки трансформація і уніфікація виконується централізовано сервером застосунків [4]. Водночас практичні дослідження показують, що такий централізований механізм має істотний недолік – при підключенні нової мережі або зміни формату/семантики повідомлень часто потрібна суттєва доробка інтегрованого прикладного програмного забезпечення, що ускладнює еволюцію системи та уповільнює адаптацію нових технологій.

Щоб зменшити ступінь втручання у центральний застосунок і підвищити гнучкість інтеграції, у практиці застосовують проміжні «адаптери застосунків» (Application Adapters) або енергоефективні Middleware-компоненти, які перед прийомом даних на мультипротокольний сервер виконують трансформацію форматів, нормалізацію схем та базову валідацію. У такій архітектурі сервер кожної мережі перетворюється на адаптер – дані проходять попередню обробку і вже у уніфікованому вигляді надходять у загальний застосунок. Огляд літератури підкреслює роль подібних Middleware рішень і Edge-передобробки для зменшення навантаження на центральні сервіси та для підвищення сумісності протоколів.

Загальний недолік варіанту інтеграції на рівні додатків – лінійне зростання кількості мережевих елементів і складності операцій при масштабуванні: кількість сенсорів і шлюзів зростає пропорційно числу об’єднуваних монопротокольних мереж, причому ці елементи часто не є взаємозамінними. Це веде до підвищення експлуатаційних витрат, збільшення ризиків відмов і зниження загальної надійності системи, якщо не впроваджено додаткові механізми оркестрації, моніторингу і автоматичного оновлення адаптерів/схем. Сучасні праці рекомендують комбінувати центральний сервер із набором модульних адаптерів, реєстром схем (Schema Registry) і CI/CD-процедурами для оновлення інтеграційних компонентів, щоби мінімізувати ці ризики.

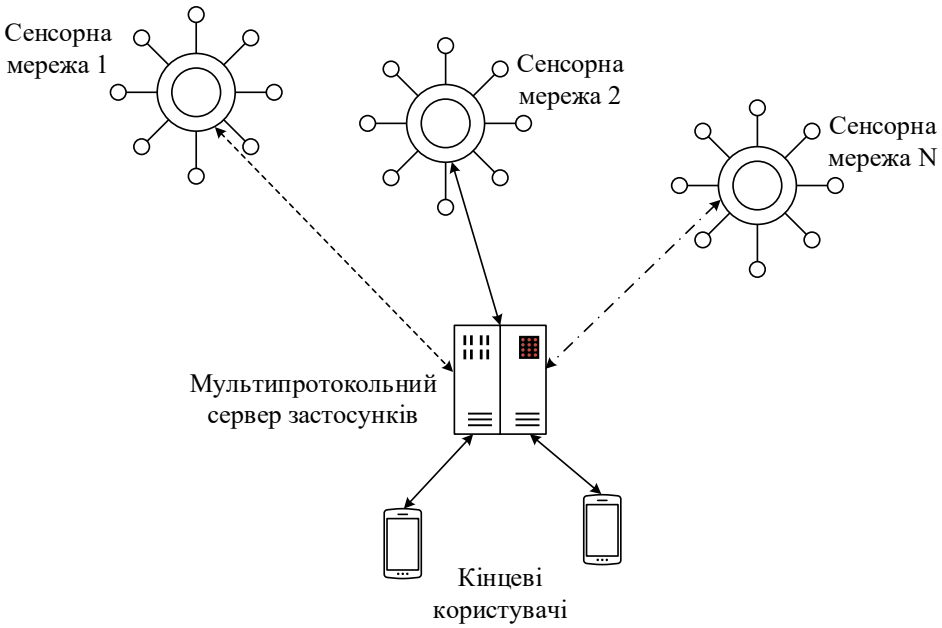

Рис. 3. Топологія мультипротокольної мережі IoT

Першим і найпоширенішим методом інтеграції розрізнених IoT-мереж на рівні додатків є реалізація RESTful-проксі на межі між веб-сервісами та пристроями, обмеженими за обчислювальною здатністю, що виконує HTTP/CoAP-переклад. Така схема дозволяє зберегти REST-семантику для хмарних додатків і одночасно використовувати легкий CoAP на кінцевих вузлах, однак практична реалізація вимагає уважної проєктної роботи з безпеки та обробки розривів наскрізних гарантій.

Другий підхід спирається на модель «брокерних мостів», де локальні та віддалені брокери публікацій/підписок зв’язуються мостами або адаптерами (наприклад, від MQTT до MQTT та MQTT до AMQP або брокерні мости між Fog і Cloud). Ця стратегія зберігає Pub/Sub-семантику та дозволяє реалізувати кешування і офлайн-підписки на рівні Fog-вузлів, проте вимагає визначення політик мапінгу QoS, ідентифікації тем і поведінки при дублюванні повідомлень. Для систем із вимогами до надійності та збереження стану брокерних мостів забезпечує гнучку реплікацію, але створює додаткову інфраструктуру з визначенням стану.

Третій клас рішень – семантичні шлюзи і Middleware-платформи, які виконують не просто протокольну трансляцію, а узгодження моделі даних: нормалізацію структур повідомлень, перерахунок одиниць виміру, версіонування схем та додавання семантичних анотацій. Такі шлюзи виступають рівнем абстракції між heterogenous IoT-стеком і бізнес-додатками, що спрощує інтероперабельність на рівні сенсорних ресурсів і «віртуальних пристроїв», але підвищує складність розробки й супроводу (потрібні реєстри схем і механізми міграції схем).

Альтернативна парадигма інтеграції на рівні додатків полягає у використанні інформаційно-центрованих підходів (ICN/NDN) через шлюзи, які містять IP-додатки і NDN-мережі. Дослідження показують, що у складних багатострибкових середовищах NDN-підходи забезпечують кращу стійкість, ефективне кешування та балансування потоків, а відповідні

шлюзові рішення (наприклад MQTT до CCN, CoAP над ICN) дають змогу зберегти сумісність із існуючою інфраструктурою.

Інтеграція IoT на рівні шлюзів. Мультипротокольний шлюз у контексті сенсорних мереж визначається як проміжний вузол, що забезпечує інтероперабельність гетерогенних монопротокольних підмереж шляхом обрізки та трансформації їхніх потоків у єдиний формат обміну даними та уніфікований інтерфейс для серверу застосунків. Такий шлюз реалізує набір функцій: протокольну трансляцію (наприклад між HTTP та CoAP або MQTT до AMQP), нормалізацію форматів (CBOR/JSON/Protobuf), локальне кешування і базову попередню обробку (агрегація, фільтрація), а також механізми безпечного завершення сеансів і управління авторизацією.

Зі сторони продуктивності та стійкості, шлюз виступає критичним елементом архітектури: він зменшує кількість кінцевих змін на сенсорах і дозволяє централізовано оптимізувати політики QoS та маршрутизації трафіку до Fog/Cloud шарів. Поведінка прикладних протоколів у різних топологіях (однострибкових чи багатострибкових) безпосередньо впливає на дизайн шлюзу – зокрема, на стратегії повторних передач, кешування та управління чергами. У складних багатострибкових сценаріях підходи з реплікацією з урахуванням переприйомів (наприклад, NDN-парадигми) показують кращу надійність порівняно з класичними наскрізними протоколами, що слід враховувати при проєктуванні шлюзових механізмів [5].

Головний операційний виграш від мультипротокольних шлюзів – можливість зменшити загальну кількість шлюзів в інфраструктурі за рахунок їхньої багатофункціональності, що приводить до зниження експлуатаційних витрат і, за певних умов, до підвищення загальної надійності системи (менше точок адміністрування та менше апаратури для обслуговування). Крім того, при підключенні нових сенсорних мереж зміни часто обмежуються конфігурацією або розширенням логіки саме в новому шлюзі, а не в усій інсталяції сенсорів, що спрощує розгортання нових технологій.

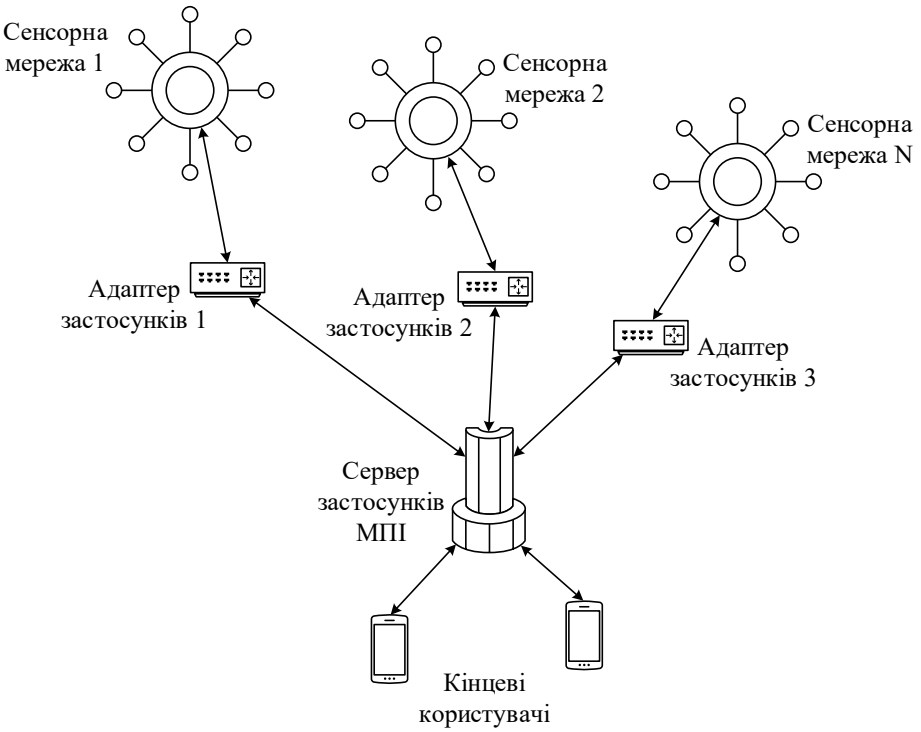

Рис. 4. Топологія мультипротокольної мережі IoT з використанням адаптерів

Водночас мультипротокольний шлюз є «вузьким місцем» з кількох причин. По-перше, функціональна універсальність збільшує апаратні та програмні вимоги до шлюзу (CPU, пам’ять, енергетичне заживлення), а також підвищує складність розробки та тестування адаптерів для кожного нового стеку або формату. По-друге, централізація трансляції і

термінування захищених каналів (між DTLS та TLS) може ускладнювати збереження наскрізних властивостей безпеки та створює додаткові ризики, якщо не впроваджено механізмів захищеного зберігання ключів і коротко-живучих даних аутентифікації. Також підвищується відповідальність шлюзу за QoS-мапінг (наприклад, відтворення гарантій MQTT QoS у CoAP або навпаки).

Інтеграція IoT на рівні сенсорів. На ринку пристроїв для IoT уже з'являються апаратні платформи (сенсорні блоки), які інтегрують кілька радіо- та протокольних стеків у межах одного апаратного модуля — наприклад платформи, що поєднують IEEE 802.11 (Wi-Fi) і IEEE 802.15.4 (ZigBee) на одній платі. Така апаратна консолідація дозволяє сенсору працювати одночасно або перемикатися між різними мережевими доменами залежно від умов оточення і вимог прикладного завдання, що відкриває можливості для гнучких топологічних схем і полегшує інтеграцію гетерогенних МПМ у мультипротокольну інфраструктуру [6].

Кожен універсальний сенсорний блок містить прошивку та драйвери, що забезпечують взаємодію з відповідними монопротокольними шлюзами, при цьому можливі режимі одно- або багатоканальної передачі даних залежно від зовнішніх умов (наявність завад у певному діапазоні, перешкоди) або внутрішніх станів (рівень заряду батареї, працездатність трансиверу). У практичних реалізаціях це означає наявність у прошивці логіки для вибору інтерфейсу, менеджера енергетичних режимів і механізмів уникнення дублювання переданих повідомлень; на серверній стороні треба передбачити можливість приймання потоків із різних протоколів та їхню нормалізацію.

Використання кількох стеків у межах одного структурного елемента має суттєві інженерні компроміси. По-перше, мультипротокольність збільшує апаратні та енергетичні вимоги: одночасна підтримка декількох трансиверів або часті переключення між режимами призводять до підвищеного енергоспоживання та, відповідно, до необхідності більшого джерела живлення або частішої заміни/підзарядки батарей. По-друге, для автономної та ефективної роботи необхідне складне керуюче ПЗ у сенсорі, здатне в реальному часі оцінювати якість каналів, пріоритезувати трафік і вибирати оптимальні маршрути передачі. Відсутність таких механізмів знижує корисність мультипротокольної реалізації в умовах обмежених ресурсів [7].

Крім технічних витрат, мультипротокольні сенсори ускладнюють операційні процедури: реєстрація та управління ідентичностями пристроїв у різних протокольних доменах вимагає узгоджених схем іменування, а також додаткових змін у серверних компонентах – іноді потрібна модернізація застосункового серверу або введення проміжних адаптерів для коректної інтерпретації даних. Отже, перехід до мультипротокольних сенсорів зазвичай супроводжується роботою з Middleware і реєстрами схем для забезпечення сумісності та безперебійної інтеграції в існуючу IT-інфраструктуру.

Позитивний ефект від застосування універсальних сенсорів полягає у підвищенні адаптивності та функціональної надійності мережі: сенсор, що підтримує кілька стандартів, може автоматично переключатися на найпридатніший канал у змінних умовах (наприклад, при появі інтерференції або відсутності одного з шлюзів), що підвищує стійкість мережі та розширює можливі сценарії застосування. Для практичної інженерії рекомендується поєднувати мультипротокольні апаратні рішення з модульною прошивкою (clear network abstraction), локальними енергетичними політиками та набором енергоефективних адаптерів на боці серверів/шлюзів, щоб мінімізувати необхідні зміни в інфраструктурі та знизити ризики при масштабуванні [9].

Висновки

Сучасний підхід до розширення застосувань IoT полягає не в уніфікації під один протокол, а в інтелектуальному поєднанні монопротокольних підмереж через набір шлюзів і сервісів, що забезпечують протокольну трансляцію, семантичну нормалізацію та управління ключовими експлуатаційними властивостями (затримка, надійність, енергія). Така архітектура дозволяє сконструювати мультипротокольну інфраструктуру (МПІ), яка комбінує сильні сторони окремих стандартів і тим самим розширює спектр реальних сценаріїв застосування IoT.

Вибір конкретного методу інтеграції мереж IoT на рівні додатків має базуватися на вимогах до затримки, гарантій доставки, обмежень пристроїв та на характеристиках мережевої топології (однострибкової чи багатострибкової). При проєктуванні обов'язково слід формалізувати політики мапінгу QoS і автентифікації, реалізувати моніторинг шлюзів/брокерів та передбачити механізми для еволюції схем даних і високої доступності компонентів.

З позиції розвитку мереж IoT інтеграція на рівні шлюзів є компромісним, але перспективним варіантом: вона поєднує гнучкість приєднання гетерогенних сенсорних підмереж із можливістю централізованого контролю та оптимізації трафіку, проте вимагає уважного інженерного підходу до модульності, безпеки та масштабування шлюзових компонентів.

Інтеграція на рівні сенсорів є перспективним напрямом для підвищення гнучкості та стійкості IoT-систем, але вимагає ретельного балансування між апаратною універсальністю, енергоефективністю та складністю керування. Практичні впровадження повинні супроводжуватися тестуванням у реальних умовах (однострибкової чи багатострибкової топології), оцінкою життєвого циклу живлення та планом оновлення серверної частини для прийому та нормалізації багатопротокольних потоків.

Список використаної літератури:

1. Hemantkumar Balasaheb Jadhav. Securing Cloud-Based IoT Architectures: A Multi-Protocol Approach. Computer Fraud and Security. 2024. P. 41–46. URL: https://doi.org/10.52710/cfs.33

2. Tsang T., Ka Hei Y. IOT-BASED SMART BUILDING MONITORING AND CONTROL STRATEGIES WITH INTELLIGENCE DEEP NEURAL NETWORKS. International Journal of Computer Science and Information Technology. 2025. Vol. 17, no. 6. P. 01–16. URL: https://doi.org/10.5121/ijcsit.2025.17601.

3. MIGS: A Modular Edge Gateway with Instance-Based Isolation for Heterogeneous Industrial IoT Interoperability / Y. Ai et al. Sensors. 2026. Vol. 26, no. 1. P. 314. URL: https://doi.org/10.3390/s26010314.

4. IoT-Enabled Cloud-Integrated Smart Parking System with Real-Time Monitoring and AI-Based Space Optimization for Next-Gen Mobility / S. Deepan Kumar et al. Advances in Design, Materials, Manufacturing, and Surface Engineering (ADMMS'26), Chennai, India, 6 February 2026. 400 Commonwealth Drive, Warrendale, PA, United States, 2026. URL: https://doi.org/10.4271/2026-28-0113.

5. XMID-MQTT: explaining machine learning-based intrusion detection system for MQTT protocol in IoT environment / H. Zeghida et al. International Journal of Information Security. 2025. Vol. 24, no. 3. URL: https://doi.org/10.1007/s10207-025-01036-w.

6. Novel Optimization Framework for Energy-Efficiency-based Resource Allocation and Multi-Hop Offloading in Blockchain-Enhanced IoT / X. Xue et al. IEEE Internet of Things Journal. 2025. P. 1. URL: https://doi.org/10.1109/jiot.2025.3562420.

7. Kambala G. Review on Multi-Domain Interoperability in IoT Gateways: A Cross-Platform Approach to Web and Software Integration for Smart Ecosystems. International Journal of Scientific Research and Management (IJSRM). 2024. Vol. 12, no. 12. P. 1845–1853. URL: https://doi.org/10.18535/ijsrm/v12i12.ec08.

8. Zamil A. K., Jasim A. D. A Multi-Factor Quantum-Resistant and Privacy-Preserving Authentication Protocol for Decentralized Systems. Mesopotamian Journal of CyberSecurity. 2025. Vol. 5, no. 3. P. 1272–1291. URL: https://doi.org/10.58496/mjcs/2025/066.

9. Dhafer S. Y., Hardik J. Lightweight Multi-Hop Routing Protocols for Efficient Resource Utilization in Edge-Enabled PLC IoT Networks. Journal of Nonlinear Analysis and Optimization. 2025. Vol. 16, no. 1. P. 1493-1510.

Автори статті

Яковець Всеволод – аспірант, старший викладач, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0002-3866-8017

Гоббязов Андрій – аспірант, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0005-6605-8808

Стародубцев Ярослав – аспірант, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID:0009-0007-5787-3661

Authors of the article

Yakovets Vsevolod – postgraduate, senior lecturer, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0002-3866-8017

Hobbiazov Andrii – postgraduate, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0005-6605-8808

Starodubtsev Yaroslav – postgraduate, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID:0009-0007-5787-3661

Надійшла до редакції: 11.02.2026
Прийнята до друку: 16.03.2026
Опубліковано: 25.05.2026