

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет програмних систем та прикладної математики
Кафедра системного програмування і спеціалізованих комп'ютерних
систем**

«На правах рукопису»
УДК 004.7:621.396

До захисту допущено:
Завідувач кафедри
Віталій РОМАНКЕВИЧ
«__» _____ 2026 р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-науковою програмою «Системне програмування та
спеціалізовані комп'ютерні системи»
зі спеціальності 123 «Комп'ютерна інженерія»
на тему: «Метод адаптивного керування параметрами радіопротоколу
для підвищення надійності передачі даних
у low-power IoT системах»**

Виконав:
студент II курсу, групи KB-41мн
Статечний Сергій Валерійович _____

Науковий керівник:
ст. викладач кафедри СПіСКС, д. ф.
Радченко Костянтин Олександрович _____

Консультант з нормоконтролю:
доцент кафедри СПіСКС, с.н.с., к.т.н.
Боярінова Юлія Євгенівна _____

Рецензент:
д. ф., асистент кафедри ОТ
Коренко Дмитро Володимирович _____

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів
без відповідних посилань.
Студент _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет програмних систем та прикладної математики
Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-наукова програма «Системне програмування та спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

«__» _____ 2025 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Статечному Сергію Валерійовичу

1. Тема дисертації «Метод адаптивного керування параметрами радіопротоколу для підвищення надійності передачі даних у low-power IoT системах», науковий керівник дисертації Радченко Костянтин Олександрович, ст. викладач кафедри СПіСКС, д. ф., затверджені наказом по університету від «3» квітня 2026 р. № 1386-с
2. Термін подання студентом дисертації 18 травня 2026 р.
3. Об'єкт дослідження: процес обміну даними між вузлами low-power IoT мережі в умовах змінного стану радіоканалу.
4. Вихідні дані: апаратні плати NUCLEO-WL55JC1 (3 шт.); специфікація LoRa; результати попередніх досліджень.
5. Перелік завдань, які потрібно розробити:
 - 5.1. Проаналізувати сучасні методи адаптивного керування параметрами LoRa та визначити їх обмеження.
 - 5.2. Сформулювати багатокритеріальну функцію оцінки якості зв'язку $Q(t)$.
 - 5.3. Розробити метод контекстно-залежної адаптації Q-ADR на основі функції $Q(t)$.
 - 5.4. Реалізувати метод на платформі STM32WL55 (вузли Master, Slave, Interferer).
 - 5.5. Реалізувати алгоритми Fixed, ADR і Q-ADR для порівняльного дослідження.
 - 5.6. Провести експериментальне дослідження за критеріями PDR та енергоефективності.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу:
 - 6.1. Презентація магістерської дисертації.
7. Орієнтовний перелік публікацій:
 - 7.1. Радченко К. О., Статечний С. В., Крайносвіт А. А. Адаптивний метод покращення надійності та енергоефективності радіопротоколу для Low power IoT систем // Прикладна математика та комп'ютинг. ПМК, 2025 : Вісімнадцята наук. конф. магістрантів та аспірантів, [Київ], 19-21 лист.

2025 р. : зб. тез доп. / ФПМ КПІ ім. Ігоря Сікорського. - Київ, 2025. - С. 334-339. URI: <https://ela.kpi.ua/handle/123456789/78493>

7.2. Радченко, К., & Статечний С. (2026). Сучасні підходи до контекстно-залежної адаптації параметрів LoRaWAN у енергообмежених IoT-системах. КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ: ОСВІТА, НАУКА, ВИРОБНИЦТВО, (62), 311-320. <https://doi.org/10.36910/6775-2524-0560-2026-62-35>

7.3. Радченко К.О., Статечний С.В. Адаптивні інтелектуальні системи для динамічного керування навантаженням вебсерверів // Інтегровані інтелектуальні робототехнічні комплекси (ІРТК-2025). Вісімнадцята міжнародна науково-практична конференція 20-21 травня 2025 р., Київ, Україна. – К.: KAI, 2025. – С. 375 – 377. (збірка тез). <https://drive.google.com/file/d/1SEJm-haqlvY8xvpGDhzqCxxZLdRhE6yz/view>

8. Дата видачі завдання 25 жовтня 2025 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання	Примітка
1	Аналіз літератури за темою магістерської дисертації	03.11.2025	виконано
2	Розробка та узгодження завдання із керівником	08.12.2025	виконано
3	Робота над першим розділом магістерської дисертації	09.02.2026	виконано
4	Розробка методу адаптивного керування (Q-ADR)	12.03.2026	виконано
5	Робота над другим розділом магістерської дисертації	02.04.2026	виконано
6	Програмно-апаратна реалізація на STM32WL55	17.04.2026	виконано
7	Робота над третім розділом магістерської дисертації	25.04.2026	виконано
8	Експериментальне дослідження ефективності методу	29.05.2026	виконано
9	Робота над четвертим розділом магістерської дисертації	03.05.2026	виконано
10	Попередній розгляд магістерської дисертації на кафедрі	06.05.2026	виконано

Студент
Науковий керівник

Сергій СТАТЕЧНИЙ
Костянтин РАДЧЕНКО

РЕФЕРАТ

Актуальність теми. Стрімкий розвиток технологій Інтернету речей (IoT) призвів до масового впровадження бездротових пристроїв, що функціонують у складі мереж класу LPWAN (Low-Power Wide Area Networks). Одним із найпоширеніших рішень є технологія LoRa, яка забезпечує передавання даних на великі відстані за низького енергоспоживання. Ефективність функціонування таких систем наряду залежить від параметрів передавання даних, а саме: коефіцієнта розширення спектра (SF), потужності сигналу (TxPower) та частоти виходу в ефір. У реальних умовах експлуатації стан радіоканалу та навантаження мережі можуть швидко змінюватися, що породжує конфлікт між надійністю передачі даних та енергоефективністю системи. Стандартний механізм адаптації Adaptive Data Rate (ADR), який реалізований в протоколі LoRaWAN є прикладом спроби вирішити дану проблему. Однак через відсутність урахування комплексного стану мережі та загальну інертність він лише частково вирішує проблему. Це створює передумови для розроблення методу контекстно-залежної адаптації, який забезпечує одночасне врахування рівня сигналу, надійності доставки та стабільності каналу.

Мета роботи. Підвищення надійності передавання даних з урахуванням енергоефективності функціонування low-power IoT систем шляхом розроблення методу адаптивного керування параметрами радіопротоколу на основі багатокритеріальної оцінки якості зв'язку.

Об'єкт дослідження. Процес обміну даними між вузлами low-power IoT мережі в умовах змінного стану радіоканалу.

Предмет дослідження. Контекстно-залежні методи адаптації параметрів радіопротоколу на основі функції оцінки якості зв'язку $Q(t)$.

Методи дослідження. У роботі застосовано методи системного та порівняльного аналізу для узагальнення сучасних підходів адаптації параметрів LoRa-мереж. Для формалізації багатокритеріальної функції оцінки якості зв'язку $Q(t)$ застосовано методи математичного моделювання. Реалізацію запропонованого методу на платформі STM32WL55 виконано з використанням

модульного програмування мовою C та підходів проектування вбудованих систем. Ефективність розроблених алгоритмів оцінено за допомогою натурних експериментів і обробки результатів за критеріями PDR та енергоефективності.

Наукова новизна. У результаті виконання роботи отримано такі наукові результати. Удосконалено метод адаптації параметрів передавання у LoRa-системах. На відміну від стандартного механізму ADR, розроблений алгоритм базується на багатокритеріальній функції оцінки якості зв'язку $Q(t)$, що враховує одночасно рівень сигналу (RSSI, SNR), надійність доставки (PDR) та стабільність якості каналу (дисперсію SNR за вікно спостереження) з ваговими коефіцієнтами, що дозволяє ефективніше реагувати на зміну стану радіоканалу та навантаження мережі. Дістала подальшого розвитку концепція контекстно-залежної адаптації параметрів радіопротоколу, яка, на відміну від відомих рішень, забезпечує одночасну узгоджену зміну SF і TxPower на основі інтегрованої оцінки стану мережі, що скорочує кількість повторних передач і зменшує енергоспоживання вузлів за умов динамічного радіосередовища.

Практична цінність. Отримані результати мають практичну цінність завдяки можливості застосування розробленого методу при проектуванні енергообмежених IoT-систем на базі технології LoRa. Реалізований на платформі STM32WL55 програмно-апаратний прототип методу демонструє підвищення надійності передавання даних та зниження енергоспоживання у порівнянні зі стандартним механізмом ADR за умов динамічного радіосередовища. Запропонований підхід може бути впроваджений у системах віддаленого моніторингу довкілля, промислового IoT та масштабованих сенсорних мережах, де критичним є балансування між надійністю зв'язку та строком автономної роботи пристроїв.

Особистий внесок магістранта. Усі основні наукові та практичні результати дисертації одержані автором особисто. У спільних публікаціях здобувач брав участь в аналізі сучасних методів адаптації параметрів LoRa-мережі. Також він узагальнив обмеження існуючих підходів і сформулював концепцію контекстно-залежної адаптації параметрів передавання, яка базується

на комплексній оцінці стану радіоканалу. У межах магістерської дисертації автор самостійно розробив формалізацію багатокритеріальної функції оцінки якості зв'язку $Q(t)$ та на її основі запропонував алгоритм Q-ADR. Крім того, автор спроектував і реалізував програмно-апаратну архітектуру цього методу на платформі STM32WL55 та провів експериментальне дослідження ефективності запропонованого методу з порівнянням із фіксованими параметрами та стандартним ADR. Науковий керівник здійснював постановку задачі, загальне керівництво дослідженням і надавав консультативну допомогу.

Апробація результатів дисертації. Основні положення і результати роботи були представлені та обговорювались на XVIII науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг – ПМК'2025» (м. Київ, 19–21 листопада 2025 р.).

Публікації. За результатами дослідження опубліковано дві наукові праці: одну статтю у фаховому науковому виданні категорії «Б» («Комп'ютерно-інтегровані технології: освіта, наука, виробництво», 2026, № 62, с. 311–320) та одні тези доповіді на XVIII науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг – ПМК'2025» (м. Київ, 2025, с. 334–339).

Структура та обсяг роботи. Магістерська дисертація складається зі вступу, чотирьох розділів, висновків до кожного розділу та загальних висновків по роботі, списку використаних джерел та додатків.

У вступі подано загальну характеристику роботи, обґрунтовано актуальність обраного напрямку досліджень, сформульовано мету і задачі, визначено об'єкт і предмет дослідження, відображено наукову новизну та практичну цінність отриманих результатів.

У першому розділі проведено аналіз сучасних методів адаптивного керування параметрами LoRa-мереж, розглянуто стандартний механізм ADR та його обмеження, проаналізовано підходи з динамічним керуванням потужністю передавання, методи на основі машинного навчання та контекстно-залежної адаптації, визначено напрям удосконалення.

У другому розділі обґрунтовано вибір метрик якості каналу зв'язку, сформульовано багатокритеріальну функцію оцінки стану зв'язку $Q(t)$, розроблено модель контекстно-залежної адаптації параметрів передавання, визначено методику експериментального дослідження.

У третьому розділі описано програмно-апаратну реалізацію запропонованого методу на платформі STM32WL55 з використанням трьох вузлів (Master, Slave, Interferer), наведено архітектуру протоколу обміну та реалізацію трьох варіантів алгоритмів керування – фіксовані параметри (Fixed), стандартний ADR і запропонований Q-ADR.

У четвертому розділі наведено результати експериментального дослідження ефективності запропонованого методу за критеріями PDR та енергоефективності у різних сценаріях радіоумов, представлено порівняльний аналіз з існуючими рішеннями.

У висновках підсумовано результати проведеної роботи.

Робота викладена на 97 сторінках, містить 13 рисунків, 8 таблиць, посилання на список використаних джерел із 20 найменувань.

Ключові слова: LoRa, LPWAN, IoT, адаптивне керування, Adaptive Data Rate, контекстно-залежна адаптація, багатокритеріальна оптимізація, енергоефективність, надійність зв'язку, STM32WL55.

ABSTRACT

Topic relevance. The rapid development of Internet of Things (IoT) technologies has led to mass deployment of wireless devices operating within Low-Power Wide Area Networks (LPWAN). One of the most widespread solutions is LoRa technology, which enables long-range data transmission at low energy consumption. The performance of such systems directly depends on transmission parameters, namely the spreading factor (SF), the transmitter power (TxPower), and the transmission frequency. Under real-world operating conditions the state of the radio channel and network load may change rapidly, which gives rise to a conflict between transmission reliability and energy efficiency. The standard Adaptive Data Rate (ADR) mechanism implemented in the LoRaWAN protocol is one example of an attempt to resolve this issue. However, due to the lack of awareness of the comprehensive network state and its general inertia, it solves the problem only partially. This creates the prerequisites for the development of a context-aware adaptation method that simultaneously takes into account signal level, delivery reliability and channel stability.

Aim of the work. Improving data transmission reliability while taking into account the energy efficiency of low-power IoT systems through the development of a method for adaptive control of radio protocol parameters based on a multi-criteria assessment of link quality.

Object of research. The process of data exchange between nodes of a low-power IoT network under variable radio channel conditions.

Subject of research. Context-aware methods for adaptation of radio protocol parameters based on the link quality assessment function $Q(t)$.

Research methods. The work applies methods of systems and comparative analysis to summarise current approaches to parameter adaptation in LoRa networks. Mathematical modelling methods are applied for the formalisation of the multi-criteria link quality assessment function $Q(t)$. The implementation of the proposed method on the STM32WL55 platform is carried out using modular programming in the C language and embedded systems design techniques. The effectiveness of the developed

algorithms is evaluated through field experiments and processing of results against the criteria of PDR and energy efficiency.

Scientific novelty. The following scientific results have been obtained in the work. The method of transmission parameter adaptation in LoRa systems has been improved. Unlike the standard ADR mechanism, the developed algorithm is based on a multi-criteria link quality assessment function $Q(t)$, which simultaneously takes into account signal level (RSSI, SNR), delivery reliability (PDR) and channel quality stability (SNR variance over observation window) with weighting coefficients, allowing more effective response to changes in the radio channel state and network load. The concept of context-aware adaptation of radio protocol parameters has received further development which, unlike known solutions, provides a coordinated simultaneous adjustment of SF and TxPower based on an integrated assessment of the network state, which reduces the number of retransmissions and decreases node energy consumption under dynamic radio environment conditions.

Practical value. The obtained results have practical value owing to the possibility of applying the developed method in the design of energy-constrained IoT systems based on LoRa technology. The hardware-software prototype of the method implemented on the STM32WL55 platform demonstrates improved data transmission reliability and reduced energy consumption compared to the standard ADR mechanism under dynamic radio environment conditions. The proposed approach can be applied in remote environmental monitoring systems, industrial IoT and scalable sensor networks where balancing link reliability against device autonomy is critical.

Personal contribution of the master's student. All major scientific and practical results of the dissertation were obtained by the author personally. In the joint publications, the author participated in the analysis of current methods for parameter adaptation in LoRa networks. The author also generalised the limitations of existing approaches and formulated the concept of context-aware adaptation of transmission parameters based on a comprehensive assessment of the radio channel state. Within the master's dissertation, the author personally developed the formalisation of the multi-criteria link quality assessment function $Q(t)$ and, on its basis, proposed the Q-ADR

algorithm. In addition, the author designed and implemented the hardware-software architecture of this method on the STM32WL55 platform and conducted experimental research on the effectiveness of the proposed method, comparing it against fixed parameters and the standard ADR. The supervisor performed the problem statement, provided overall research guidance and consultative assistance.

Approbation of the dissertation results. The main provisions and results of the work were presented and discussed at the 18th Scientific Conference of Master's and PhD Students "Applied Mathematics and Computing – AMC'2025" (Kyiv, 19–21 November 2025).

Publications. Two scientific works have been published on the basis of the research results: one paper in a professional scientific journal of category "B" ("Computer-Integrated Technologies: Education, Science, Production", 2026, No. 62, pp. 311–320) and one set of conference theses at the 18th Scientific Conference of Master's and PhD Students "Applied Mathematics and Computing – AMC'2025" (Kyiv, 2025, pp. 334–339).

Structure and scope of the work. The master's dissertation consists of an introduction, four chapters, conclusions to each chapter and general conclusions to the work, a list of references and appendices.

The introduction provides a general description of the work, substantiates the relevance of the chosen research direction, formulates the aim and tasks, defines the object and subject of research, and reflects the scientific novelty and practical value of the obtained results.

The first chapter analyses approaches to adaptive control of data transmission parameters in LoRa networks, examines the standard ADR mechanism and its limitations, reviews approaches with dynamic transmission power control, methods based on machine learning and context-aware adaptation, and identifies the direction for improvement.

The second chapter substantiates the choice of channel quality metrics, formulates the multi-criteria link quality assessment function $Q(t)$, develops a model

of context-aware adaptation of transmission parameters and defines the methodology of the experimental study.

The third chapter describes the hardware-software implementation of the proposed method on the STM32WL55 platform with the use of three nodes (Master, Slave, Interferer), presents the architecture of the exchange protocol and the implementation of the three control algorithm variants – fixed parameters (Fixed), the standard ADR and the proposed Q-ADR.

The fourth chapter presents the results of the experimental study of the effectiveness of the proposed method against the criteria of PDR and energy efficiency under different radio condition scenarios, with a comparative analysis against existing solutions.

The conclusions summarise the results of the conducted work.

The work is presented on 97 pages, contains 13 figures, 8 tables and references to a list of 20 sources.

Keywords: LoRa, LPWAN, IoT, adaptive control, Adaptive Data Rate, context-aware adaptation, multi-criteria optimisation, energy efficiency, link reliability, STM32WL55.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	14
ВСТУП	16
РОЗДІЛ 1 АНАЛІЗ ПІДХОДІВ ДО АДАПТИВНОГО КЕРУВАННЯ	
ПАРАМЕТРАМИ ПЕРЕДАЧІ ДАНИХ У LORA-МЕРЕЖАХ.....	18
1.1 Технологія LoRa та протокол LoRaWAN	18
1.2 Стандартний механізм Adaptive Data Rate і його обмеження	21
1.3 Альтернативні підходи до адаптивного керування	25
1.3.1 Підходи для сценаріїв з динамічними радіоумовами.....	25
1.3.2 Підходи для сценаріїв з високою щільністю мережі.....	26
1.3.3 Підходи з розширеною оцінкою стану каналу.....	27
1.3.4 Підходи з адаптацією на стороні вузла.....	27
1.4 Порівняльний аналіз існуючих підходів.....	28
1.5 Висновки до розділу	30
РОЗДІЛ 2 РОЗРОБКА МЕТОДУ АДАПТИВНОГО КЕРУВАННЯ	
ПАРАМЕТРАМИ РАДІОПРОТОКОЛУ З ВИКОРИСТАННЯМ	
БАГАТОКРИТЕРІАЛЬНОЇ ФУНКЦІЇ ЯКОСТІ КАНАЛУ	32
2.1 Постановка задачі та обґрунтування напряду досліджень.....	32
2.2 Вибір метрик якості каналу.....	34
2.2.1 RSSI як індикатор бюджету радіоканалу	35
2.2.2 SNR як показник якості декодування	35
2.2.3 PDR як прямий показник надійності.....	36
2.2.4 Дисперсія SNR як показник стабільності каналу	37
2.2.5 Метрики, не включені до функції $Q(t)$	39
2.3 Композиція функції якості каналу $Q(t)$	40
2.4 Логіка прийняття рішень Q-ADR	44
2.4.1 Порогова класифікація стану каналу	44
2.4.2 Ескалація параметрів за деградації якості.....	45
2.4.3 Деескалація параметрів за відновлення якості	47

	13
2.4.4 Механізм блокування та захист від коливань	48
2.5 Висновки	49
РОЗДІЛ 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ МЕТОДУ Q-ADR НА	
ПЛАТФОРМІ STM32WL55	51
3.1 Архітектура експериментальної установки.....	51
3.2 Апаратна платформа	52
3.3 Архітектура прошивки	55
3.4 Реалізація протоколу обміну даними	58
3.4.1 Цикл master–slave та фази обміну	58
3.4.2 Формат кадру	60
3.4.3 Механізм SET_PARAMS та обробка відмов.....	62
3.5 Реалізація методу Q-ADR.....	63
3.5.1 Структури даних	64
3.5.2 Обчислення функції якості $Q(t)$	64
3.5.3 Реалізація логіки прийняття рішень.....	66
3.6 Реалізація еталонів для порівняння.....	67
3.7 Інфраструктура експерименту	70
3.7.1 Генератор завад	70
3.7.2 Логування подій	73
3.7.3 Автоматизація проведення експериментів.....	74
3.8 Висновки	75
РОЗДІЛ 4 АНАЛІЗ РЕЗУЛЬТАТІВ ЕКСПЕРИМЕНТАЛЬНОГО	
ДОСЛІДЖЕННЯ МЕТОДУ Q-ADR.....	77
4.1 Методика експерименту	77
4.2 Експериментальні сценарії.....	79
4.3 Дослідження адаптації за імпульсних LoRa-завад	80
4.4 Дослідження адаптації за неперервних FSK-завад.....	84
4.5 Висновки	88
ВИСНОВКИ.....	91
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	95

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ACK – acknowledgement (підтвердження отримання пакета)

ACK delay – затримка підтвердження отримання пакета

ADR – Adaptive Data Rate (механізм адаптивного керування параметрами передавання у LoRaWAN)

BER – Bit Error Rate (рівень бітових помилок)

BW – Bandwidth (ширина смуги пропускання)

CR – Coding Rate (швидкість кодування)

CSS – Chirp Spread Spectrum (модуляція з лінійним частотним перетворенням)

DR – Data Rate (індекс швидкості передачі в LoRaWAN)

EIRP – Equivalent Isotropically Radiated Power (ефективна ізотропно випромінювана потужність)

ETSI – European Telecommunications Standards Institute (Європейський інститут телекомунікаційних стандартів)

HAL – Hardware Abstraction Layer (рівень апаратних абстракцій для уніфікованої роботи з периферією мікроконтролера)

Interferer – окремий вузол, що генерує радіозавади

IoT – Internet of Things (Інтернет речей)

ISM – Industrial, Scientific, Medical (неліцензовані радіодіапазони для промислового, наукового та медичного використання)

LoRa – Long Range (технологія модуляції фізичного рівня на базі CSS)

LoRaWAN – Long Range Wide Area Network (мережний MAC-протокол на базі LoRa)

LPWAN – Low-Power Wide Area Network (мережа з низьким енергоспоживанням і великим радіусом дії)

Master – головний вузол у point-to-point радіопротоколі

PDR – Packet Delivery Ratio (частка успішно доставлених пакетів)

PER – Packet Error Rate (частка пакетів з помилками декодування)

PHY – Physical Layer (фізичний рівень моделі OSI)

Q-ADR – Quality-based Adaptive Data Rate (запропонований метод адаптації на основі функції якості $Q(t)$)

RSSI – Received Signal Strength Indicator (індикатор потужності сигналу на вході приймача)

RX – receive (приймання)

SDR – Software Defined Radio (програмно-кероване радіо)

SF – Spreading Factor (коефіцієнт розширення спектра)

Slave – підлеглий вузол у point-to-point радіопротоколі

SNR – Signal-to-Noise Ratio (відношення сигнал/шум)

TX – transmit (передавання)

TxPower – Transmission Power (потужність передавача)

UART – Universal Asynchronous Receiver-Transmitter (універсальний асинхронний приймач-передавач)

ВСТУП

Кількість підключених до інтернету пристроїв у світі вже давно перевищила кількість населення планети. За оцінками IoT Analytics, у 2024 році в активному використанні перебувало понад 18 мільярдів пристроїв Інтернету речей (Internet of Things, IoT), а до 2030 року прогнозується подвоєння цього показника [1]. Значна частина цих пристроїв, зокрема, сенсори, лічильники, актуатори, працює у складі мереж класу Low-Power Wide Area Network (LPWAN). Такі мережі поєднують у собі дві властивості, які важко об'єднати в традиційних радіотехнологіях: великий радіус дії (одиниці кілометрів у міській забудові, до 15 км на відкритій місцевості) та низьке енергоспоживання (роки автономної роботи від однієї батареї).

Однією з найпоширеніших технологій у цьому класі є LoRa (Long Range), побудована на запатентованій модуляції Chirp Spread Spectrum (CSS) і працює у неліцензованих діапазонах ISM (Industrial, Scientific, Medical) 433, 868 та 915 МГц залежно від регіону [2]. Поверх фізичного рівня LoRa працює мережний протокол LoRaWAN (Long Range Wide Area Network), що визначає правила доступу до середовища, безпеку та керування пристроями. Саме LoRaWAN використовують у більшості публічних і приватних LPWAN-мереж – від комунальних лічильників і моніторингу повітря до агросенсорики та систем безпеки [4].

Робота LoRa-вузла визначається кількома ключовими параметрами фізичного рівня, які впливають на дальність, надійність і енергоспоживання передачі. Це коефіцієнт розширення спектра (Spreading Factor, SF), потужність передавача (TxPower), ширина смуги (Bandwidth, BW) та швидкість кодування (Coding Rate, CR). Збільшення SF підвищує чутливість приймача та дальність зв'язку, але водночас збільшує час перебування в ефірі (time-on-air) і, відповідно, енергоспоживання та ймовірність колізій. Зменшення TxPower економить енергію, але робить сигнал вразливішим до завад. Між надійністю передачі та енергоефективністю існує компроміс, що особливо проявляється у щільних мережах та за змінних радіоумов.

Стандартний механізм Adaptive Data Rate (ADR), описаний у специфікації LoRaWAN, є найпоширенішим інструментом цього керування. Він динамічно адаптує SF, TxPower і кількість повторних передач залежно від показника відношення сигнал/шум (Signal-to-Noise Ratio, SNR), що оцінюється на стороні мережного сервера за останніми uplink-пакетами [3]. Однак цей механізм має суттєві обмеження. Алгоритм спирається лише на SNR, ігноруючи PDR, затримки підтверджень (ACK delay) та поточне навантаження мережі. Інерційність також висока: щонайменше 20 uplink-пакетів потрібно для зміни конфігурації, що за динамічних умов каналу обертається тривалою роботою з неоптимальними параметрами передачі. У мобільних вузлах та за сильної інтерференції стандартний ADR взагалі може не досягати стабільної конфігурації [3].

Ситуація ускладнюється тим, що в реальних розгортаннях стан радіоканалу та навантаження мережі змінюються постійно: з'являються нові вузли, переміщуються джерела завад, змінюється рівень фонового шуму. Це створює потребу у механізмах адаптації, які працюють із сукупною оцінкою стану мережі та реагують швидше за стандартний ADR. Запропоновано низку альтернативних підходів – від методів машинного навчання (ML) до модифікацій з урахуванням колізій. Кожен з них має свої обмеження за складністю реалізації, обчислювальними витратами або вимогами до інфраструктури [5].

Метою даної магістерської дисертації є розроблення методу контекстно-залежної адаптації параметрів радіопротоколу, який на відміну від стандартного ADR використовує багатокритеріальну оцінку стану зв'язку і реалізується безпосередньо на стороні кінцевого пристрою без необхідності залучення мережного сервера. Це робить метод придатним для застосування у спрощених розгортаннях LoRa peer-to-peer без повного стека LoRaWAN, що часто зустрічається у промислових IoT-системах з власними протоколами обміну.

РОЗДІЛ 1 АНАЛІЗ ПІДХОДІВ ДО АДАПТИВНОГО КЕРУВАННЯ ПАРАМЕТРАМИ ПЕРЕДАЧІ ДАНИХ У LORA-МЕРЕЖАХ

1.1 Технологія LoRa та протокол LoRaWAN

LPWAN-технології відрізняються від стільникових мереж (4G/5G) і мереж ближнього радіуса (Wi-Fi, Bluetooth) поєднанням великого радіусу дії з низьким енергоспоживанням кінцевих пристроїв. На сьогодні провідні позиції в цьому сегменті займають три технології: LoRa з протоколом LoRaWAN, NB-IoT (Narrowband IoT) та Sigfox [6]. NB-IoT працює у ліцензованому спектрі стільникових операторів і забезпечує гарантовану якість обслуговування через інфраструктуру 4G/5G, але потребує абонентської плати та має вищі піки споживання струму під час передачі. Sigfox використовує ультравузькосмугову модуляцію у неліцензованому спектрі та централізовану інфраструктуру одного оператора, що обмежує гнучкість розгортання приватних мереж. LoRa, на відміну від обох, працює у неліцензованому спектрі та допускає як публічні (через мережі типу The Things Network), так і повністю автономні приватні мережі без зовнішнього оператора. Саме це робить її найбільш гнучкою для дослідницьких і промислових сценаріїв, де власник системи бажає контролювати весь стек від кінцевого вузла до сервера.

Технологія LoRa визначає лише фізичний рівень передачі – модуляційну схему та параметри радіоканалу. Її основою є модуляція CSS, запатентована компанією Semtech [7]. На відміну від класичної FSK-модуляції, CSS використовує сигнали з лінійною зміною частоти у часі (chirp), що забезпечує стійкість до інтерференції та доплерівського зсуву частоти, характерного для рухомих вузлів. Перевагою CSS є здатність декодувати сигнал навіть тоді, коли потужність шуму перевищує потужність сигналу. Для SF12 цей запас сягає 20 дБ нижче рівня шуму [2]. Більшість інших схем модуляції в неліцензованому спектрі такої стійкості не мають.

Поведінку LoRa-вузла визначають чотири взаємопов'язані параметри фізичного рівня:

- Spreading Factor (SF) – коефіцієнт розширення спектра, що набуває значення від 7 до 12. Кожне збільшення SF на одиницю подвоює тривалість передачі одного символу і покращує бюджет лінії приблизно на 2.5 дБ. Це збільшує дальність зв'язку, але водночас подовжує час перебування в ефірі (time-on-air) та збільшує енергоспоживання передачі.
- Bandwidth (BW) – ширина смуги пропускання, типово 125 або 250 кГц у європейському діапазоні 868 МГц. Ширша смуга пришвидшує передачу, але знижує чутливість приймача приблизно на 3 дБ при кожному подвоєнні BW.
- Coding Rate (CR) – швидкість прямої корекції помилок (Forward Error Correction, FEC), набуває значення 4/5, 4/6, 4/7 або 4/8. Чим більший знаменник (тобто 4/8 проти 4/5), тим більше надлишкових бітів додається до пакета, що підвищує стійкість до пакетних помилок ціною додаткового time-on-air.
- TxPower – потужність передавача. У європейському діапазоні 868 МГц нормами ETSI обмежена до +14 дБм EIRP при дотриманні правила максимального duty cycle 1% на канал.

Комбінація цих параметрів формує компроміс між дальністю, пропускнуою спроможністю та енергоспоживанням, який і є основним об'єктом адаптивного керування. Перехід від SF7 до SF12 збільшує time-on-air пакета приблизно у 32 рази [2], що відповідно множить енергоспоживання передачі, але дозволяє приймати сигнал з місць, де SF7 декодувати неможливо.

LoRa-радіо працює у неліцензованих ISM-діапазонах. Регіональна сегментація відрізняється: 868 МГц у Європі, 915 МГц у США та частині Азії, 433 МГц у деяких інших країнах [6]. Регуляторні обмеження неліцензованого спектру (duty cycle 1% у Європі, специфічні правила Listen-Before-Talk у деяких діапазонах США) формують верхню межу частоти передач і змушують враховувати ці обмеження при проєктуванні протоколу обміну даними.

LoRaWAN – це специфікація MAC-рівня, яка працює поверх фізичного рівня LoRa, розроблена та підтримувана альянсом LoRa Alliance [4]. Стандарт визначає правила доступу до середовища, формати кадрів, процедури активації пристроїв (Over-The-Air Activation, OTAA, та Activation By Personalization, ABP), механізми безпеки на основі AES-128, а також ключовий для цієї роботи механізм адаптивного керування ADR, детально розглянутий у підрозділі 1.2.

Архітектурно LoRaWAN-мережа має чотирирівневу зіркоподібну структуру (рисунк 1.1):

- кінцеві пристрої (end-devices) – сенсори, лічильники, актуатори, що передають дані через LoRa-радіо;
- шлюзи (gateways) – приймають радіопакети від усіх вузлів у радіусі дії та ретранслюють їх через IP-мережу;
- мережний сервер (Network Server) – центральний компонент, що обробляє метадані пакетів, відповідає за дедуплікацію (один пакет може бути прийнятий кількома шлюзами одночасно), маршрутизацію, керування активацією та виконує алгоритм ADR;
- сервер застосунків (Application Server) – отримує розшифровані корисні дані для подальшої обробки бізнес-логікою.

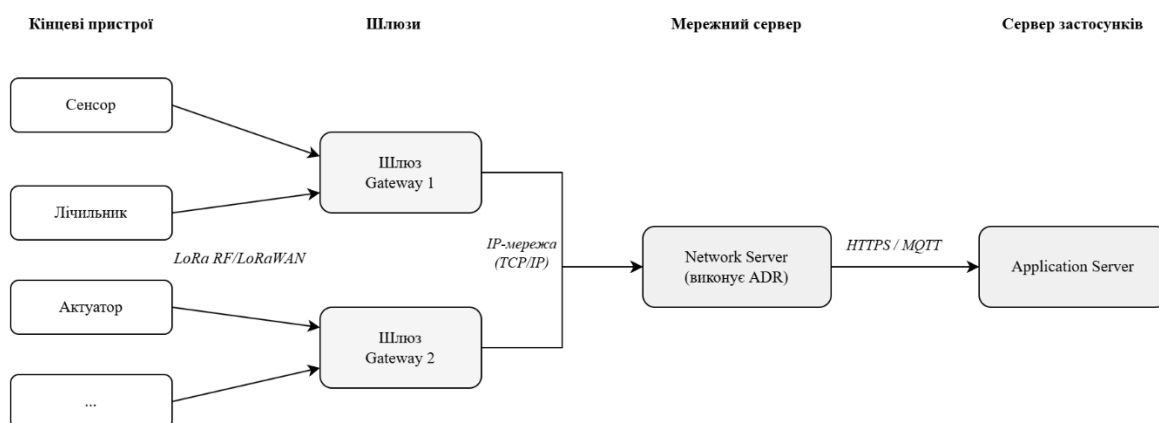


Рисунок 1.1 – Архітектура LoRaWAN-мережі

Ключовою особливістю архітектури з точки зору адаптивного керування є те, що рішення про зміну параметрів вузла приймає мережний сервер, а не сам вузол. Сервер має повну статистику від усіх шлюзів, бачить історію передач і може приймати рішення централізовано. Вузол лише виконує отримані команди MAC-рівня і підтверджує їх. Така архітектура спрощує реалізацію вузла, але створює залежність від наявності сервера та робить адаптацію недоступною у сценаріях без серверної інфраструктури. Це характерно, зокрема, для LoRa peer-to-peer мереж з власним протоколом обміну, поширених у промисловому IoT.

Стандарт визначає три класи кінцевих пристроїв за моделлю приймання downlink-повідомлень. Клас А – найбільш енергоефективний: пристрій відкриває два короткі вікна приймання (RX1 і RX2) лише після власної передачі, в інший час радіо вимкнене. Клас В додає періодичні синхронізовані вікна приймання за рахунок маяків (beacons) від шлюзів, що збільшує середнє споживання, але дозволяє серверу ініціювати передачу до вузла з обмеженою затримкою. Клас С тримає радіо у режимі приймання постійно, окрім моментів власної передачі – найвищі затримки реакції на команди, але і найбільше споживання, що робить цей клас непридатним для автономних пристроїв з батарейним живленням. Більшість датчиків і лічильників, які формують основну масу LoRaWAN-вузлів, працюють у класі А саме через жорсткі вимоги до часу автономної роботи [4].

Описана архітектура LoRaWAN добре працює в типових сценаріях стаціонарного моніторингу, наприклад для лічильників комунальних послуг, агрокліматичних сенсорів чи моніторингу повітря. У таких випадках канал зазвичай стабільний, а централізоване прийняття рішень на сервері не спричиняє проблем із затримками. Однак саме в умовах динамічних радіоумов та для сценаріїв без сервера обмеження архітектури проявляються найвиразніше.

1.2 Стандартний механізм Adaptive Data Rate і його обмеження

Стандартний механізм ADR визначений у специфікації LoRaWAN [4] як основний інструмент керування параметрами передавання у всіх класах кінцевих

пристроїв. Хоча специфікація не визначає конкретні алгоритми і залишає їх на розсуд оператора, на практиці закріпився підхід від Semtech, який використовують у більшості публічних мереж, зокрема в The Things Network [9]. У цьому підрозділі розглянуто роботу цього алгоритму та проаналізовано його обмеження для динамічних радіоумов і сценаріїв без серверної інфраструктури.

Алгоритм ADR розподілений між мережним сервером і кінцевим пристроєм, але основна логіка прийняття рішень виконується саме на сервері (NS-managed ADR). Робота алгоритму починається після того, як вузол виставляє прапорець $ADR=1$ у заголовку uplink-кадру. Сервер починає накопичувати статистику SNR у вікні з N останніх пакетів від цього вузла, де типове значення $N=20$ [3, 4]. Після наповнення вікна обчислюється запас за відношенням сигнал/шум за формулою (1.1):

$$SNR_margin = SNR_max - SNR_req - device_margin \quad (1.1)$$

де SNR_max – максимальне значення SNR у вікні; SNR_req – мінімальне SNR, необхідне для демодуляції на поточному SF (визначається чутливістю приймача за таблицею демодуляції: від -7.5 дБ для SF7 до -20 дБ для SF12 [7]); $device_margin$ – додатковий запас безпеки, типово 10 дБ [3, 9].

Далі обчислюється кількість кроків адаптації $NStep$ як цілочисельне ділення SNR_margin на 3 (3 дБ – крок зміни параметрів). За позитивного $NStep$ сервер послідовно зменшує SF (до мінімального SF7), а коли SF досягає мінімуму – зменшує TxPower на 2 дБм за кожен залишковий крок до мінімуму потужності (+2 дБм для EU868). За негативного $NStep$ TxPower збільшується назад до регіонального максимуму (+14 дБм для EU868). Нові параметри передаються вузлу командою MAC-рівня, на яку вузол відповідає підтвердженням. Сторона вузла активується лише в окремому випадку: якщо за певний час від сервера не надходить downlink-кадрів, end-device-managed ADR самостійно знижує швидкість передачі (збільшує SF), припускаючи погіршення каналу – ця гілка алгоритму не належить до NS-side логіки і використовується як механізм відновлення зв'язку. Послідовність кроків NS-managed ADR показана на рисунку 1.2. Граничні значення SF (SF7 для EU868) і TxPower (від +2 до +14

дБм для EU868) визначаються регіональним планом частот, алгоритм оперує цими дискретними рівнями за таблицями специфікації [4].

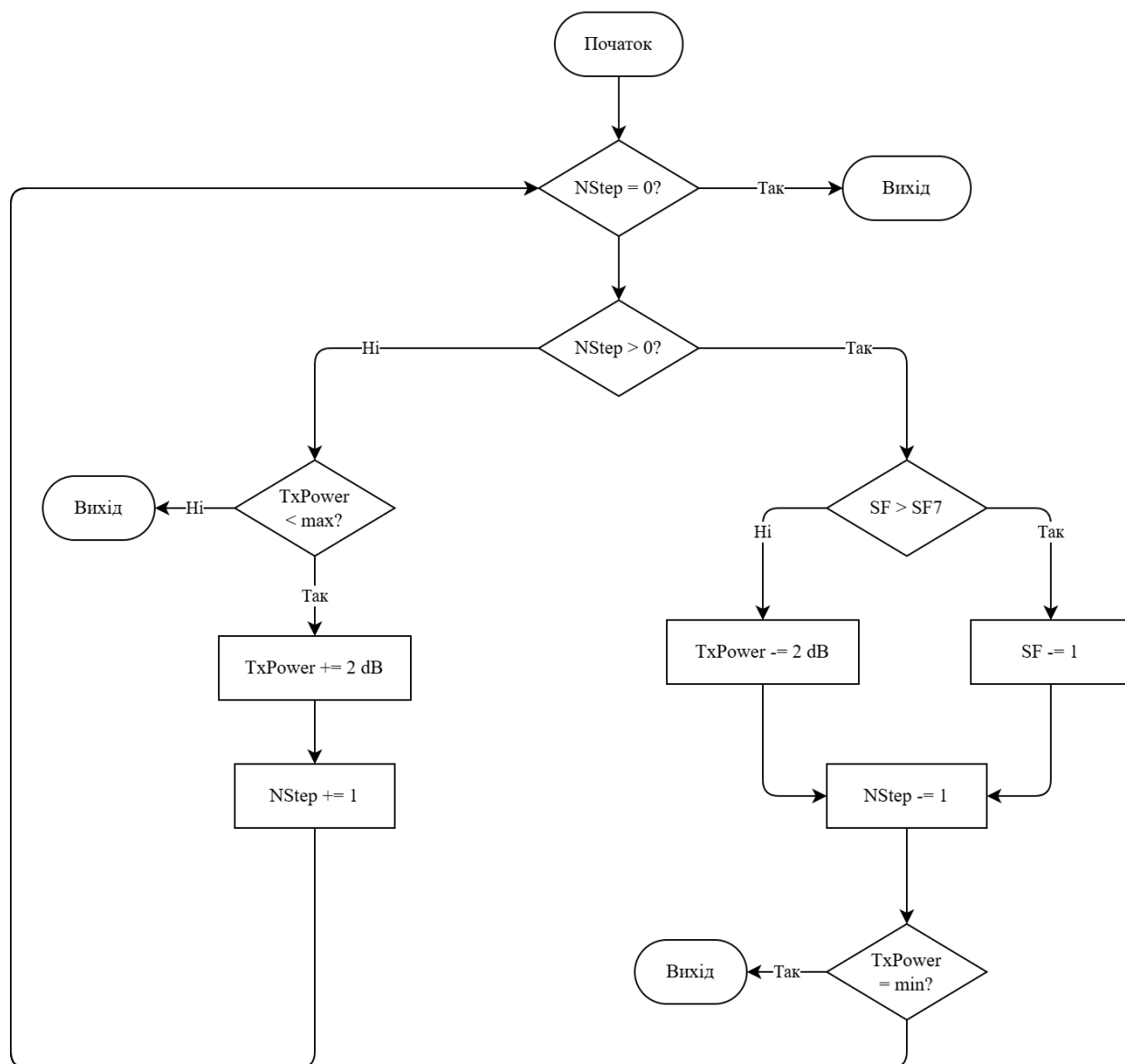


Рисунок 1.2 – Блок-схема стандартного алгоритму ADR

Описана архітектура ефективно працює в типових сценаріях стаціонарного моніторингу, але має низку обмежень, що проявляються за динамічних умов експлуатації.

Одним із основних обмежень є інерційність прийняття рішень. Алгоритм виконується тільки після накопичення вікна з 20 пакетів, що при типовому періоді передачі 30–300 секунд відповідає від 10 хвилин до 1.5 годин між послідовними адаптаціями [3]. У роботі [3] показано, що для рухомих пристроїв або в змінному радіосередовищі затримка призводить до роботи з неоптимальними параметрами, оскільки канал встигає змінитися ще до завершення обробки попереднього вікна. Автори того самого дослідження зазначають, що частота запуску алгоритму довільна та фактично не прив'язана до реальної динаміки каналу [3]. Окремим випадком є пристрої, що вже працюють на DR0 (SF12), де алгоритм у деяких реалізаціях може взагалі не виконуватися, оскільки нижче знижувати швидкість нікуди.

Іншою проблемою є однокритеріальність оцінки. ADR використовує як показник стану каналу лише максимальне значення SNR за вікно. Інші важливі характеристики ігноруються: частка успішно доставлених пакетів (PDR), затримки підтверджень (ACK delay), щільність трафіку в сусідніх вузлах. В огляді [9] показано, що оцінка каналу лише за SNR створює занадто оптимістичну картину і не враховує втрати через колізії. Це підтверджується експериментами [8]: у щільних мережах PDR може знижуватися до 50% навіть за високого SNR, тоді як стандартний ADR цього не враховує. Модифікація алгоритму з адаптацією Coding Rate [10] частково підвищує надійність, однак проблема однокритеріальності лишається актуальною.

Ще одним фактором неефективності є параметр `device_margin`. Це статичний запас (типово 10 дБ), що додається до `SNR_req` при обчисленні `margin`. Запас потрібен для компенсації коливань каналу між адаптаціями, але оскільки він налаштовується на етапі розгортання і не змінюється у часі, алгоритм консервативно тримає його навіть тоді, коли канал стабільно хороший. Дослідження [9] показує, що оптимальне значення `device_margin` може значно відрізнятися для різних сценаріїв розгортання, а нездатність ADR підлаштовувати цей параметр динамічно є додатковим джерелом енергетичних втрат.

Окремо варто відзначити централізацію логіки ADR на мережному сервері. NS-managed ADR за визначенням потребує наявності повного стека LoRaWAN з network server, application server та шлюзами, що обмінюються через IP-мережу. Це обмежує застосування алгоритму в сценаріях без серверної інфраструктури. Зокрема, йдеться про LoRa peer-to-peer мережі з власним протоколом обміну, поширені у промисловому IoT та закритих відомчих системах. Перенесення логіки ADR на вузол ускладнюється тим, що він не має доступу до об'єднаної статистики від кількох шлюзів і змушений приймати рішення лише на основі власних спостережень.

Сукупність цих обмежень показує, що стандартний ADR залишається відносно повільним, однокритеріальним і прив'язаним до серверної інфраструктури. Для типових сценаріїв стаціонарного моніторингу його можливостей достатньо, але динамічні радіоумови, мобільні вузли, P2P-розгортання та сценарії з високою щільністю трафіку потребують інших підходів. Альтернативні методи адаптації, що пропонуються в літературі для подолання цих обмежень, розглянуто у наступному підрозділі.

1.3 Альтернативні підходи до адаптивного керування

Стандартний ADR має три основні слабкості: він повільно реагує на зміни, дивиться лише на SNR і працює тільки на сервері. Для кожної з цих проблем у літературі є свої рішення. Нижче розглянуто чотири групи підходів – згруповані за тим, який саме сценарій експлуатації вони покривають.

1.3.1 Підходи для сценаріїв з динамічними радіоумовами

ADR погано працює при швидкій зміні стану радіоканалу. Насамперед це актуально для мобільних вузлів та умов з тимчасовими завадами і нерівномірним покриттям у міському середовищі. Поки алгоритм збирає 20 пакетів, реальний стан каналу вже інший, і обрана конфігурація стає неоптимальною.

Одним з варіантів вирішення даної проблеми є спрощення алгоритму, щоб він приймав рішення на основі останніх кількох пакетів або навіть одного. У

роботі [11] описано підхід ADR-Lite, який приймає рішення без накопичення довгої історії. Час реакції менший, але дисперсія параметрів вища. У симуляціях для мобільних end-devices ADR-Lite вдвічі скорочує час до стабільної конфігурації, у статичних сценаріях ефекту майже немає.

Інший варіант – машинне навчання для прогнозування стану каналу. Алгоритм аналізує тенденції і коригує параметри проактивно, не чекаючи появи проблеми. У [12] використано регресійні моделі, тоді як у [13] застосовано навчання з підкріпленням, що дозволяє формувати політику адаптації безпосередньо під час роботи мережі. У симуляціях такі методи зменшують PER (Packet Error Rate) на 15–30% порівняно з ADR. Найбільший ефект саме при мобільності.

Ключовою слабкістю ML-підходу є обчислювальна складність. ML-моделі і RL-агенти потребують пам'яті на історію спостережень і нетривіальних обчислень кожен ітерацію. На мікроконтролерах класу Cortex-M0+/M4 з кількома десятками кілобайтів RAM повноцінна реалізація проблематична. ADR-Lite без проблем реалізується на embedded залізі, але дивиться лише на SNR і не помічає колізій чи навантаження мережі.

1.3.2 Підходи для сценаріїв з високою щільністю мережі

У щільних мережах основні втрати спричиняють не шуми, а колізії. Кілька вузлів виходять в ефір одночасно в результаті чого шлюз не декодує жоден пакет. Стандартний ADR колізій не бачить тому загублений через колізію пакет для нього просто не існує.

У [8] цей ефект показано через симуляції в LoRaSim. У мережах з понад тисячею вузлів у радіусі шлюза PDR падає до 50% і нижче, при тому що SNR залишається прийнятним. Запропонована авторами модифікація ADR використовує оцінку колізій як додатковий вхід, що піднімає PDR на 15–25%.

Інший напрямок це зміна самого способу доступу до середовища. У [15] досліджено CSMA та slotted-варіанти замість стандартного ALOHA. У мережах з понад двома тисячами вузлів slotted-протокол підвищує пропускну здатність у 1.5–2 рази, при цьому самі параметри передавання не змінюються.

Обмеженням цих підходів є вузьке спрямування, вони працюють тільки з колізіями. Якщо канал погіршується через інтерференцію або рух, регулювання інтервалів передач не допомагає. Slotted-протоколи додатково потребують синхронізації між вузлами, а це складно реалізувати без централізованого арбітра.

1.3.3 Підходи з розширеною оцінкою стану каналу

Третій напрямок адресує однокритеріальність ADR. Замість одного SNR пропонується дивитися на кілька метрик одночасно. Це дозволяє відрізнити ситуації, які для ADR виглядають однаково: стабільний канал з низьким навантаженням і канал з тим же SNR, але великою часткою колізій.

У [14] описано ML-підхід, що навчається оптимальної конфігурації SF, BW і CR з урахуванням SNR, RSSI та статистики попередніх передач разом. Multi-параметрична модель краще відповідає реальному стану каналу, особливо при нестабільній інтерференції. Робота [10] показує схожий ефект через додавання Coding Rate і статистики колізій, надійність передачі при цьому зростає на 10–20%.

У [16] окремо вивчена ортогональність LoRa-сигналів з різними SF. Виявилось, що припущення про повну ортогональність, на яке неявно спирається ADR, у щільних мережах не виконується. Це означає, що навіть правильно обрані параметри можуть давати гірший результат, ніж очікується теоретично.

Обмеження тут двояке. ML-моделі знов потребують більше ресурсів, ніж є в типового IoT-пристрою. Більшість пропозицій передбачають виконання логіки на network server, що для P2P-сценаріїв не підходить.

1.3.4 Підходи з адаптацією на стороні вузла

Остання група підходів спрямована на подолання архітектурної прив'язки ADR до серверу. У P2P-розгортаннях, ad-hoc мережах і edge-IoT повний LoRaWAN-стек не використовується, тож канонічного ADR там немає в принципі. Жоден з підходів 1.3.1–1.3.3 не працює в таких умовах без істотної переробки.

У [17] описано SlidingChange, це адаптація параметрів LoRa без участі серверу. Алгоритм використовує ковзне вікно з останніх N пакетів для згладжування коливань і приймає рішення прямо на пристрої. Якість адаптації порівнянна зі стандартним ADR, серверна інфраструктура не потрібна. Слабкістю SlidingChange як і раніше є однокритеріальність. Алгоритм працює тільки з SNR і не враховує статистики доставлення чи навантаження.

Робота [18] описує LoRa-мережу без шлюзів і серверу. Вузли утворюють mesh-топологію і передають пакети напряму через інші вузли. Автори зосереджені переважно на маршрутизації пакетів між вузлами, а не на адаптації параметрів зв'язку. Конфігурація вузла спирається тільки на показники сигналу, без врахування статистики доставлення чи затримок підтверджень. Це залишає відкритим питання, як поєднати локальне виконання адаптації з оцінкою кількох метрик одночасно.

Обидва підходи знімають серверну прив'язку, але роблять це за рахунок меншої інформаційної бази. Багатокритеріального аналізу, який є перевагою серверних рішень з 1.3.3, ні в SlidingChange, ні в P2P-мережі немає.

1.4 Порівняльний аналіз існуючих підходів

Розглянуті у попередньому підрозділі альтернативні методи адаптації відрізняються за принципом роботи, набором використовуваних метрик і архітектурною прив'язкою. Для систематизованого порівняння у таблиці 1.1 зведено ключові характеристики стандартного ADR і п'яти альтернативних підходів за критеріями, які мають значення для подальшого вибору методу: місце виконання логіки, набір метрик оцінки, параметри адаптації, періодичність прийняття рішень та реалізованість на стороні кінцевого вузла.

Зведене порівняння показує, що серверні підходи (стандартний ADR, ADR-Lite, RL/ML-методи, Multi-criteria моделі) можуть оперувати ширшим набором даних, але потребують повного стека LoRaWAN і не реалізуються на стороні вузла. SlidingChange є єдиним підходом, який працює локально на

пристрої, проте обмежений однією метрикою (SNR). Anti-collision методи покращують доступ до середовища, але адаптують лише часові параметри передавання, не реагуючи на зміни якості каналу.

Таблиця 1.1 – Порівняльний аналіз підходів до адаптивного керування параметрами LoRa

Критерій	Стандартний ADR	ADR-Lite	RL/ML-методи	Anti-collision	Multi-criteria	Sliding Change
Місце виконання	мережний сервер	мережний сервер	мережний сервер	змішане	мережний сервер	вузол
Метрики оцінки	SNR	SNR останнього пакета	SNR, історія, контекст мобільності	оцінка колізій	SNR, RSSI, частка втрат пакетів	SNR (ковзне вікно)
Параметри адаптації	SF, TxPower	SF, TxPower	SF, TxPower, CR	інтервали передач	SF, BW, CR	SF, TxPower
Періодичність прийняття рішень	після N пакетів (вікно)	на кожен пакет	за подією або таймером	за спостереженням каналу	після N пакетів	після кожного пакета (ковзне вікно)
Складність реалізації на вузлі	не реалізується	не реалізується	обмежено	частково (на рівні MAC)	обмежено	реалізується

Жоден з оглянутих підходів не поєднує одночасно локальне виконання на вузлі, оцінку за кількома метриками та адаптацію параметрів передавання. Саме цей пробіл і визначає напрям подальшого дослідження.

1.5 Висновки до розділу

У цьому розділі проведено аналіз сучасних підходів до адаптивного керування параметрами передачі даних у LoRa-мережах. Розглянуто фізичний рівень LoRa та архітектуру протоколу LoRaWAN, детально описано стандартний механізм Adaptive Data Rate і його обмеження, а також систематизовано альтернативні методи адаптації за чотирма напрямками: швидкою реакцією на динамічні радіоумови, зменшенням колізій у щільних мережах, розширеною оцінкою стану каналу та локальною адаптацією на стороні вузла.

Аналіз показав, що стандартний механізм ADR має низку принципів обмежень. Алгоритм повільно реагує на зміни. Для прийняття рішення йому потрібно накопичити вікно з близько 20 пакетів, а це 10 хвилин і більше при типовому періоді передачі. Він також однокритеріальний, оскільки спирається лише на максимальне значення SNR і не враховує частку успішно доставлених пакетів, затримки підтверджень або щільність трафіку. Крім того, логіка ADR централізована на мережному сервері та потребує повного стека LoRaWAN, що робить її недоступною у LoRa peer-to-peer розгортаннях, поширених у промисловому IoT.

Огляд альтернативних підходів показав, що для кожного з обмежень стандартного ADR у літературі запропоновані часткові рішення. Підходи зі спрощеним вікном адаптації (ADR-Lite) та методи прогнозування на основі машинного навчання прискорюють реакцію на зміни каналу, але потребують серверної інфраструктури або обчислювальних ресурсів, що перевищують можливості типових IoT-вузлів. Anti-collision механізми зменшують втрати у щільних мережах, але адаптують лише часові параметри передавання. Multi-criteria моделі дозволяють врахувати кілька метрик одночасно, проте виконуються на сервері. Edge-підходи на кшталт SlidingChange знімають серверну прив'язку, але повертаються до однокритеріальної логіки.

Порівняльний аналіз, наведений у таблиці 1.1, підтверджує, що жоден з оглянутих методів не поєднує одночасно локальне виконання на стороні вузла, оцінку каналу за кількома метриками та адаптацію основних параметрів

передавання SF і TxPower. Ця комбінація вимог особливо критична для сценаріїв LoRa peer-to-peer без серверної інфраструктури в енергообмежених IoT-пристроях.

На основі цього аналізу новий метод адаптивного керування має відповідати таким вимогам:

- виконуватися безпосередньо на стороні кінцевого вузла без залучення мережного сервера;
- оцінювати стан зв'язку на основі кількох метрик одночасно – рівня сигналу, частки доставлених пакетів і стабільності каналу;
- реагувати на зміни швидше за стандартний ADR без накопичення довгої історії;
- зберігати обчислювальні вимоги в межах ресурсів типових IoT-мікроконтролерів;
- узгоджено адаптувати параметри SF і TxPower на основі інтегрованої оцінки стану каналу.

РОЗДІЛ 2 РОЗРОБКА МЕТОДУ АДАПТИВНОГО КЕРУВАННЯ ПАРАМЕТРАМИ РАДІОПРОТОКОЛУ З ВИКОРИСТАННЯМ БАГАТОКРИТЕРІАЛЬНОЇ ФУНКЦІЇ ЯКОСТІ КАНАЛУ

2.1 Постановка задачі та обґрунтування напрямку досліджень

Раніше було сформульовано п'ять вимог, яким має задовольняти метод адаптивного керування параметрами радіопротоколу для low-power IoT систем, що працюють у динамічному радіосередовищі. Ці вимоги задають напрям подальшого дослідження. У цьому розділі задача формалізується, обирається набір вимірюваних метрик каналу, формулюється цільова функція оцінки стану каналу та правила прийняття рішень. Окремо описано базові алгоритми, з якими порівнюється запропонований метод у наступних розділах.

Адаптивне керування у цій роботі стосується двох параметрів фізичного рівня – SF і TxPower. Обидва параметри є дискретними. SF набуває значень від 7 до 12, TxPower у регіоні EU868 – від 2 дБм до 22 дБм з кроком 2 дБ [7]. Інші параметри радіоканалу (смуга пропускання, швидкість кодування, центральна частота) залишаються незмінними протягом сеансу. Їхня динамічна зміна не є частиною ADR і у літературі розглядається окремо як предмет специфічних розширень стандарту [10]. Робочий діапазон SF, що використовується безпосередньо у запропонованій реалізації, додатково уточнюється у підрозділі 2.4.

Задачу адаптивного керування у формалізованому вигляді можна сформулювати так. Нехай $S(t)$ – стан радіоканалу в момент часу t , що включає сукупність вимірюваних характеристик якості зв'язку: рівень сигналу, відношення сигнал/шум, частку успішних доставок та інші показники, опис яких наводиться у наступному підрозділі. Нехай $p(t)$ – вектор поточних параметрів передавача:

$$p(t) = (SF(t), (TxPower(t))). \quad (2.1)$$

Тоді завдання адаптивного керування полягає у визначенні функції F , яка за станом каналу повертає такий новий вектор параметрів, що максимізує надійність доставки при мінімально можливих витратах енергії на передавання:

$$p(t + \Delta t) = F(S(t)). \quad (2.2)$$

У запропонованому методі функція F шукається у формі, що базується на скалярній функції якості $Q(S(t)) \in [0, 1]$, яка інтегрує множину метрик стану каналу в одне значення. Деталі побудови Q та правил формування p на її основі наведено у наступних підрозділах.

Сформульована задача має дві принципові особливості, що визначають вибір підходу до її розв'язання.

Першою особливістю є конфлікт критеріїв надійності та енергоефективності. Підвищення TxPower покращує надійність доставки, але енергоспоживання зростає пропорційно. Аналогічно зі SF: збільшення цього параметра підвищує чутливість приймача й стійкість зв'язку до завад, але час передачі пакета зростає експоненціально. Це збільшує енергоспоживання й одночасно зменшує пропускну здатність каналу [2]. Метод адаптації не може максимізувати обидва критерії одночасно. Задача зводиться до пошуку прийняттого компромісу, який залежить від поточного стану каналу. У стабільному каналі компроміс зсувається у бік енергоефективності, у деградованому – у бік надійності.

Друга особливість полягає у відсутності централізованого спостерігача. У канонічному LoRaWAN рішення про адаптацію приймається мережним сервером на основі даних від кількох шлюзів [4]. У цій роботі розглядається point-to-point реалізація без сервера. Це означає, що алгоритм адаптації виконується безпосередньо на кінцевому вузлі, без зовнішньої інфраструктури, на обмежених обчислювальних ресурсах. Така постановка є характерною для приватних промислових і дослідницьких розгортань LoRa [17] і безпосередньо корелює з висновками попереднього розділу щодо обмежень класичного ADR.

З огляду на вимоги, сформульовані раніше, та зазначені особливості задачі, обрано такий напрям дослідження. Для оцінки стану каналу застосовується

багатокритеріальна функція якості $Q(t)$, що інтегрує кілька різнопланових метрик в одну скалярну величину. Такий підхід відрізняється від класичного ADR, що оперує однією метрикою (SNR margin), та від ML-методів, які вимагають значних обчислювальних ресурсів і навчальних вибірок [12, 13]. Інтегрована скалярна оцінка дозволяє приймати рішення на основі однозначного порівняння з пороговими значеннями, зберігаючи водночас інформативність вхідних даних від кількох метрик. Концепцію інтеграції кількох індикаторів каналу у єдину функцію якості розглянуто у статті [5]. Там її обґрунтовано як перспективний напрям контекстно-залежної адаптації.

Запропонований метод отримав назву Q-ADR (Quality-based Adaptive Data Rate). Назва вказує на спорідненість з ADR і водночас на принципову відмінність – використання функції $Q(t)$ замість однієї метрики.

2.2 Вибір метрик якості каналу

У попередньому підрозділі задача адаптивного керування зведена до пошуку функції F , що за станом каналу $S(t)$ повертає новий вектор параметрів передавача. Сам стан $S(t)$ поки залишається у загальному вигляді – як сукупність вимірюваних характеристик каналу. Тепер визначимо, які саме характеристики потрапляють до цього набору і як кожна з них використовується далі функцією $Q(t)$.

Вибір метрик підпорядковується двом міркуванням. Кожна метрика має передавати окремий аспект стану каналу і не дублювати решту – інакше у функції $Q(t)$ кілька внесків будуть одне й те саме, тільки з різними вагами. Окрім цього, метрика має бути доступною безпосередньо на вузлі, без додаткової інфраструктури – це впливає з вимоги локального виконання алгоритму. За цими міркуваннями обрано чотири метрики: рівень прийнятого сигналу RSSI, відношення сигнал/шум SNR, частку успішно доставлених пакетів PDR та дисперсію SNR за вікно спостереження. Інші можливі метрики обговорено у кінці підрозділу.

2.2.1 RSSI як індикатор бюджету радіоканалу

RSSI (Received Signal Strength Indicator) – це показник потужності сигналу на вході приймача, виміряний у дБм. У LoRa-модулях RSSI обчислюється апаратно за результатами демодуляції преамбули кожного прийнятого пакета. Типовий діапазон значень становить від -30 дБм для дуже близьких передавачів до -130 дБм на межі чутливості приймача [7].

З точки зору адаптивного керування RSSI безпосередньо показує, наскільки сильно сигнал перевищує мінімальну чутливість приймача – тобто який запас має сигнал над порогом демодуляції. Високий RSSI означає, що сигнал стійкий до можливих завад та завмирань, без ризику втрати пакетів. Низький RSSI вказує на роботу близько до межі демодуляції, де навіть невелика деградація умов одразу призведе до помилок. У контексті адаптації SF і TxPower RSSI є основним індикатором того, наскільки доцільно знижувати потужність передавача без шкоди для надійності.

Однією метрикою RSSI обмежитися не можна. Високий рівень сигналу ще не гарантує його декодованість: за наявності інтерференції від інших радіопристроїв у тому ж діапазоні RSSI може залишатися високим, бо приймач реєструє сумарну енергію в каналі, але корисний сигнал при цьому декодується невірно. Це особливо актуально для неліцензованих ISM-діапазонів, де співіснують десятки різнотипних радіосистем [16]. Тому RSSI у функції $Q(t)$ використовується як необхідна, але недостатня умова якості каналу – у комбінації з іншими показниками.

2.2.2 SNR як показник якості декодування

SNR (Signal-to-Noise Ratio) – це відношення потужності корисного сигналу до потужності шуму, виміряне у дБ. У LoRa-приймачах SNR обчислюється під час демодуляції преамбули, разом з RSSI, і повертається у метаданих кожного прийнятого пакета. Типовий діапазон значень становить від $+10$ дБ для сильного

сигналу до -20 дБ для слабкого сигналу, що декодується тільки завдяки властивостям CSS-модуляції [7].

SNR є ключовою метрикою для LoRa саме через CSS-модуляцію. На відміну від FSK та інших класичних схем, які потребують позитивного SNR для коректного декодування, CSS дозволяє відновлювати сигнал нижче рівня шуму. Для SF7 поріг декодування становить -7.5 дБ, а для SF12 знижується до -20 дБ [2]. Така властивість і дає LoRa її головну перевагу – здатність декодувати пакети навіть в каналах з високим рівнем шуму, де інші технології взагалі не працюють. Збільшення SF на одиницю покращує бюджет каналу приблизно на 2.5 дБ ціною подвоєння часу передачі пакета.

З точки зору адаптивного керування саме SNR безпосередньо вказує, який SF доцільно вибрати. Якщо SNR має помітний запас над порогом для поточного SF, можна знизити SF, що зменшить час передачі та енергоспоживання. Якщо ж SNR близький до порогу декодування, треба підіймати SF, інакше зростає ризик помилок. Це і є логіка, на якій базується стандартний механізм ADR – обчислення SNR margin та перерахунок його в кроки зміни SF або TxPower.

Проте лише SNR недостатньо. SNR показує стан декодування одного пакета і не бачить статистики втрат. Якщо SNR залишається високим, але пакети все одно губляться через колізії чи імпульсні завади, ADR на це не зреагує тому, що для нього канал виглядає прийнятним.

2.2.3 PDR як прямий показник надійності

PDR (Packet Delivery Ratio) – це частка успішно доставлених пакетів від загальної кількості переданих за певний інтервал. На відміну від RSSI і SNR, які характеризують умови отримання одного пакета, PDR оцінює канал з точки зору результату передачі. Це робить його найбільш близьким до прикладної мети адаптивного керування, якою є надійна доставка даних.

У запропонованому методі PDR обчислюється на стороні передавача за ковзним вікном з останніх N передач за формулою (2.3):

$$PDR = N_{ack} / N, \quad (2.3)$$

де N_{ack} – кількість пакетів у вікні, на які отриманий ACK; N – розмір вікна спостереження.

Розмір вікна N – компроміс між реактивністю і стабільністю оцінки. Менші значення дають швидку реакцію на зміну каналу, але збільшують дисперсію значень PDR через статистичну невизначеність на малих вибірках. Більші значення згладжують флуктуації, але затримують виявлення нових тенденцій. У запропонованій реалізації обрано $N = 10$ пакетів. За тривалості циклу master–slave 4 секунди це відповідає вікну спостереження близько 40 секунд, що дозволяє виявляти зміни стану каналу швидше за стандартний ADR і водночас зберігати достатню статистичну значущість оцінки PDR.

PDR як метрика має кілька властивостей, що роблять її незамінною у багатокритеріальній оцінці. Вона безпосередньо враховує колізії та інші втрати, невидимі для SNR. У сценаріях з високою щільністю мережі або зовнішніми завадами від інших радіосистем у тому ж діапазоні SNR може залишатися високим. Коли пакети все ж приходять, вони приходять з хорошим сигналом. Але загальна частка доставлених пакетів все одно падає. RSSI у такій ситуації теж буде високим. Саме PDR фіксує цей ефект першим [8, 9].

PDR не залежить від абсолютних значень потужності сигналу. RSSI і SNR є інженерними характеристиками радіоканалу, вони мають сенс лише у контексті конкретного приймача та порогових значень. PDR є безрозмірною величиною у діапазоні $[0, 1]$, що однаково інтерпретується незалежно від конкретної реалізації. Це робить даний параметр зручним для нормалізації у складі функції $Q(t)$.

Ключова слабкість даної метрики це інерційність та необхідність накопичення статистики. На відміну від RSSI і SNR, що доступні відразу після отримання пакета, осмислене значення PDR потребує хоча б кількох передач у вікні. У стартовому стані вузла, коли вікно ще не наповнене, PDR не може використовуватися як надійний індикатор.

2.2.4 Дисперсія SNR як показник стабільності каналу

Перші три метрики (RSSI, SNR, PDR) описують рівень якості каналу – наскільки добре він зараз працює. Однак для адаптивного керування важлива не лише поточна якість, але й її стабільність у часі. Канал з постійним $SNR = +5$ дБ і канал, що коливається між $+12$ дБ і -2 дБ із середнім тим самим $+5$ дБ, мають однакові середні характеристики, але поведуться абсолютно по-різному. У першому випадку можна впевнено знизити SF або TxPower. У другому ці зміни ризиковані, короткий провал каналу одразу призведе до втрати пакетів.

Стандартний механізм ADR цю різницю не помічає, бо оперує лише максимальним SNR за вікно [3]. Запропонований метод використовує дисперсію SNR як окрему метрику стану каналу. Дисперсія обчислюється за тим самим вікном, що й PDR, і визначається стандартною формулою (2.4):

$$D_{SNR} = 1/N * \sum_i (SNR_i - SNR_{avg})^2, \quad (2.4)$$

де N – розмір вікна спостереження; SNR_i – значення SNR для i -го пакета у вікні; SNR_{avg} – середнє значення SNR за вікно.

Низька дисперсія означає, що канал стабільний і поточне значення SNR можна вважати репрезентативним. Висока дисперсія вказує на нестабільність та непередбачуваність каналу. У такій ситуації будь-яке рішення про оптимізацію параметрів (зниження TxPower або SF) є ризикованим. Воно може бути правильним для поточного моменту, але неправильним через секунду, коли канал просяде.

З точки зору практичної інтерпретації висока дисперсія SNR характерна для трьох типових ситуацій:

- імпульсні завади – інший радіопристрій періодично виходить в ефір через, що відбуваються колізії пакетів і значення SNR коливається;
- рух вузла – переміщення передавача або приймача змінює шлях поширення сигналу, що дає різні значення SNR від пакета до пакета;
- нестабільне середовище – відбиття від рухомих об'єктів (людей, машин), завмирання через атмосферні явища, температурні зміни характеристик радіо тощо.

В усіх трьох випадках середній SNR залишається помірним, а дисперсія сигналізує про те, що канал ненадійний. Алгоритм, що використовує дисперсію у складі функції $Q(t)$, не знижує параметри передачі при нестабільному каналі. Він тримає вищу потужність і більший SF, доки канал не стабілізується.

Із всіх чотирьох метрик дисперсія SNR є найменш стандартною для адаптивних алгоритмів LoRa. У роботі [17] алгоритм SlidingChange спирається на ковзне середнє SNR і коливання не враховує. У методі EARN [10] додано адаптацію Coding Rate, але дисперсія знов не використовується. Окрема метрика дисперсії дозволяє розрізняти стабільно хороший канал і канал, який в середньому хороший, але нестабільний. Саме ця різниця визначає, чи можна агресивно знижувати параметри передачі.

2.2.5 Метрики, не включені до функції $Q(t)$

Обрані чотири метрики покривають три різнопланових аспекти каналу: рівень сигналу, результат передачі та стабільність у часі. У літературі є й інші метрики, але кожна з них або дублює обрані, або непридатна для нашого сценарію.

PER – це частка пакетів з помилками декодування у вікні. У сценарії з ACK-підтвердженням розрізнити, чи пакет не дійшов взагалі, чи дійшов з помилками, неможливо. Пакет або підтверджений, або ні. PER не дає нової інформації понад PDR.

Затримка ACK (ACK delay) у деяких методах [9, 12] використовується як індикатор завантаженості мережі. У стандартному LoRaWAN затримка пов'язана з очікуванням RX-вікон та можливою конкуренцією за шлюз. У нашому point-to-point протоколі ситуація інша: slave відповідає одразу після отримання, а час відповіді детермінований і повністю визначається поточним SF. Окрема метрика ACK delay тут немає сенсу.

Потужність завад (interference power) – це окремий вимір рівня завад у каналі. LoRa-радіо не надає прямого виміру цієї величини. Приймач реєструє лише сумарну потужність на вході (RSSI) і не розрізняє корисний сигнал від завад. Побічно цю інформацію можна отримати визначивши розрив між RSSI та

SNR. Цей зв'язок вже закладений у функцію $Q(t)$ через одночасне використання обох метрик.

Втрати на бітовому рівні (BER, Bit Error Rate) – нижчий за пакет рівень оцінки. LoRa-приймач BER не вимірює. При CSS-демодуляції помилкові біти призводять до невдалого декодування всього пакета, що уже покрито PDR.

Таким чином, обрані чотири метрики не лише доповнюють одна одну за змістом, але й покривають усі індикатори каналу, доступні у LoRa-вузлі без додаткової апаратури або модифікації радіо.

2.3 Композиція функції якості каналу $Q(t)$

У попередньому підрозділі було обрано чотири метрики стану каналу, а саме: RSSI, SNR, PDR та дисперсію SNR. Кожна з них передає окремий аспект якості зв'язку. Проте для прийняття рішень алгоритму потрібна єдина величина, з якою можна однозначно порівнювати порогові значення. Цей підрозділ описує, як чотири різнопланові метрики об'єднуються у скалярну функцію якості $Q(t)$.

Обрані метрики мають різну природу і різні шкали вимірювання. RSSI вимірюється у дБм і набуває від'ємних значень від -130 дБм і вище залежно від умов розгортання. SNR вимірюється у дБ і залежить від умов поширення та обраного коефіцієнта розширення. Нижньою межею нормалізації обрано -20 дБ, яке є мінімально допустимим значенням SNR для декодування при найбільшому коефіцієнті розширення, передбаченому стандартом LoRa (SF12). Порогові значення SNR на пряму залежать від SF: від -7.5 дБ для SF7 до -20 дБ для SF12. Нижче -20 дБ декодування є неможливим для жодного SF, визначеного стандартом LoRa. PDR є безрозмірною величиною у діапазоні $[0, 1]$. Дисперсія SNR має розмірність дБ² і теоретично не обмежена зверху. Безпосередньо підсумувати ці величини неможливо, оскільки вони непорівнянні ні за розмірністю, ні за діапазоном. Тому перед інтеграцією кожен метрику потрібно нормалізувати, тобто привести до спільної безрозмірної шкали $[0, 1]$, де 0 відповідає найгіршому стану каналу, а 1 – найкращому. Сирі метрики зберігають

свої усталені позначення ($RSSI$, SNR , PDR , D_{SNR} для дисперсії), а їхні нормалізовані відповідники позначаються окремими символами R , S , P та V .

У формулах використовується функція $clamp(x; a; b)$, яка повертає a якщо $x < a$, b якщо $x > b$, і x в усіх інших випадках – тобто обмежує значення заданим діапазоном.

Рівень сигналу приводиться до шкали $[0, 1]$ лінійним відображенням робочого діапазону $RSSI$. Нижньою межею обрано значення -130 дБм, яке близьке до чутливості приймача та нижче якого декодування практично неможливе. Верхньою межею обрано -50 дБм – рівень, вище якого подальше зростання потужності вже не впливає на надійність, бо канал і так має значний запас. Нормалізований рівень сигналу R обчислюється за формулою (2.5):

$$R = clamp\left(\frac{RSSI_{avg} + 130}{80}; 0; 1\right), \quad (2.5)$$

де $RSSI_{avg}$ – середній рівень сигналу за ковзне вікно спостереження.

Функція $clamp$ потрібна тому, що реальні значення $RSSI$ можуть виходити за межі обраного діапазону. Сигнал, сильніший за -50 дБм, отримує $R = 1$ – подальше посилення не несе додаткової інформації для алгоритму. Сигнал, слабший за -130 дБм, отримує $R = 0$.

Відношення сигнал/шум приводиться до шкали $[0, 1]$ аналогічно. Нижня межа нормалізації -20 дБ обґрунтована вище. Верхньою межею обрано $+10$ дБ, рівень, що відповідає каналу з великим запасом якості. Нормалізоване відношення сигнал/шум S обчислюється за формулою (2.6):

$$S = clamp\left(\frac{SNR_{avg} + 20}{30}; 0; 1\right), \quad (2.6)$$

де SNR_{avg} – середнє відношення сигнал/шум за ковзне вікно спостереження.

Частка успішних доставок PDR уже є безрозмірною величиною у діапазоні $[0, 1]$ і не потребує нормалізації. Це одна з причин, чому PDR зручна для багатокритеріальної оцінки – вона входить до функції $Q(t)$ безпосередньо. Надалі частка успішних доставок позначається символом P .

Дисперсія SNR потребує не лише нормалізації, але й інверсії шкали. Для трьох попередніх метрик більше значення означає кращий стан каналу. Для

дисперсії навпаки: висока дисперсія означає нестабільний, непередбачуваний канал, тобто гірший стан. Тому нормалізована дисперсія V будується так, щоб низька дисперсія давала V близьке до 1, а висока близьке до 0. Обчислення виконується за формулою (2.7):

$$V = 1 - \text{clamp}\left(\frac{D_{SNR}}{V_{max}}; 0; 1\right), \quad (2.7)$$

де D_{SNR} – дисперсія SNR за ковзне вікно спостереження; V_{max} – параметр масштабування, що задає рівень дисперсії, вище якого канал вважається повністю нестабільним.

Параметр V_{max} визначає чутливість метрики. У роботі прийнято значення цього параметра, що дорівнює 300. Це означає, що дисперсія SNR на рівні 300 дБ² і вище дає $V = 0$, а проміжні значення масштабуються лінійно. Вибір цього значення забезпечує, що нормалізована дисперсія зберігає ненульове значення навіть за помірної нестабільності каналу і не насичується передчасно.

Чотири нормалізовані метрики об'єднуються у скалярну функцію якості зваженою сумою за формулою (2.8):

$$Q(t) = \alpha R + \beta S + \gamma P + \delta V, \quad (2.8)$$

де R, S, V – нормалізовані значення RSSI, SNR та дисперсії SNR; P – частка успішних доставок PDR; $\alpha, \beta, \gamma, \delta$ – вагові коефіцієнти, що задають внесок кожної метрики.

Вагові коефіцієнти підпорядковані умові нормування (2.9):

$$\alpha + \beta + \gamma + \delta = 1. \quad (2.9)$$

Завдяки цьому функція $Q(t)$ є опуклою комбінацією нормалізованих метрик і гарантовано набуває значень у діапазоні $[0, 1]$. Значення $Q = 1$ відповідає ідеальному стану каналу за всіма метриками одночасно, $Q = 0$ – повній деградації каналу.

Значення вагових коефіцієнтів визначено на основі фізико-інженерної важливості кожної метрики для оцінки стану каналу LoRa незалежно від результатів проведених експериментів. Прийнято такі значення: $\alpha = 0.10$, $\beta = 0.40$, $\gamma = 0.30$, $\delta = 0.20$.

Найбільшу вагу присвоєно SNR ($\beta = 0.40$). Відношення сигнал/шум є найпрямішою метрикою здатності CSS-модулятора декодувати пакет: пороги декодування LoRa визначаються саме у SNR і становлять від -7.5 дБ для SF7 до -20 дБ для SF12. Саме SNR безпосередньо вказує, який запас має канал над межею декодування, тому ця метрика є основною для вибору SF.

Другою за вагою є PDR ($\gamma = 0.30$) – прямий показник результату передачі. PDR має найбільшу прикладну значущість, бо вимірює саме надійність доставки даних. Проте вага PDR менша за вагу SNR через інерційність метрики. PDR підтверджує деградацію постфактум, коли пакети вже втрачено, тоді як SNR попереджає про погіршення умов раніше.

Дисперсія SNR отримала вагу $\delta = 0.20$. Це метрика другого порядку, оскільки вона характеризує не сам рівень сигналу, а його стабільність у часі. Дисперсія дозволяє відрізнити рівномірно деградований канал від каналу зі спорадичними завадами, що важливо для прийняття рішень, але є допоміжним, а не основним індикатором якості.

Найменшу вагу присвоєно RSSI ($\alpha = 0.10$). За наявності зовнішньої інтерференції RSSI може вводити в оману, приймач реєструє сумарну потужність у каналі, і високий RSSI не гарантує декодованість корисного сигналу. RSSI залишається корисним як індикатор бюджету радіоканалу, але його внесок до інтегральної оцінки свідомо обмежений.

Окремої уваги потребує ситуація, коли пакет не доставлено – отримано тайм-аут замість підтвердження ACK. Для такого пакета значення RSSI і SNR недоступні оскільки приймач не зафіксував жодного сигналу. Проте просто пропустити втрачений пакет при обчисленні середніх не можна – це спотворило б оцінку, оскільки саме втрати є найгіршими подіями для каналу.

У запропонованому методі застосовується імпуація найгіршим значенням. Для кожного втраченого пакета у ковзне вікно спостереження підставляються значення $\text{RSSI} = -130$ дБм і $\text{SNR} = -20$ дБ – величини, що відповідають межі непрацездатності каналу. Середні значення, з яких

обчислюються R і S , а також дисперсія, з якої обчислюється V , визначаються за повним вікном з урахуванням цих підставлених значень.

Такий підхід забезпечує, що втрата пакета проявляється у функції $Q(t)$ одразу через три канали. По-перше, напряму знижується PDR. По-друге, знижуються R і S , бо середні значення тягнуться вниз підставленими величинами -130 і -20 . По-третє, змінюється дисперсія SNR – при поодиноких втратах підставлені значення збільшують розкид відносно реальних вимірів і знижують V . При глибокій деградації вирішальним чинником стає P , який падає прямо пропорційно до кількості втрат у вікні. Завдяки цьому одна втрата пакета дає узгоджений сигнал деградації за кількома метриками одночасно, а не лише за PDR. Це робить функцію $Q(t)$ чутливою до втрат навіть тоді, коли решта пакетів у вікні має формально прийнятні показники. Окремим граничним випадком є повна відсутність підтверджень у ковзному вікні. За відсутності будь-яких вимірювань $Q(t)$ покладається рівним нулю безпосередньо, що відповідає стану повної деградації каналу.

2.4 Логіка прийняття рішень Q-ADR

Функція якості $Q(t)$, побудована у підрозділі 2.3, дає інтегральну оцінку стану каналу одним числом у діапазоні $[0, 1]$. Проте сама по собі ця оцінка не змінює параметрів передавача – вона лише вхідний сигнал для логіки прийняття рішень. Цей підрозділ описує, як алгоритм Q-ADR на основі значення Q обирає, чи змінювати коефіцієнт розширення SF і потужність передавача TxPower, у який бік і на скільки. Логіка складається з порогової класифікації стану каналу, правил ескалації та деескалації параметрів і механізму блокування, що захищає систему від надмірно частих перемикань.

2.4.1 Порогова класифікація стану каналу

Прийняття рішень будується не на безперервній шкалі Q , а на дискретній класифікації. Неперервне значення Q відображається у один із чотирьох станів каналу за допомогою трьох порогових значень. Такий підхід спрощує логіку,

замість того щоб реагувати на кожну дрібну зміну Q , алгоритм діє лише при переході через поріг:

$$\text{стан}(Q) = \begin{cases} \text{критичний, якщо } Q < Q_{crit} \\ \text{низький, якщо } Q_{crit} \leq Q < Q_{low} \\ \text{робочий, якщо } Q_{low} \leq Q \leq Q_{high} \\ \text{високий, якщо } Q > Q_{high} \end{cases} \quad (2.10)$$

де $Q_{crit} = 0.40$ – критичний поріг, нижче якого канал вважається непрацездатним; $Q_{low} = 0.60$ – нижній поріг робочого діапазону; $Q_{high} = 0.90$ – верхній поріг робочого діапазону.

Кожен стан має власну інтерпретацію і власну реакцію. *Критичний стан* ($Q < Q_{crit}$) означає, що канал утратив працездатність – частка успішних доставок упала приблизно до 40 % і нижче, і потрібне негайне втручання. *Низький стан* ($Q_{crit} \leq Q < Q_{low}$) відповідає погіршеному, але ще робочому каналу, який потребує підсилення параметрів. *Робочий стан* ($Q_{low} \leq Q \leq Q_{high}$) – це цільовий режим: канал працює стабільно, і будь-яка зміна параметрів є зайвою. *Високий стан* (Q більше Q_{high}) означає, що канал має надлишковий запас якості, а отже частину ресурсів передавача можна заощадити без шкоди для надійності.

Проміжок між нижнім і верхнім порогами, тобто робочий стан, відіграє роль зони гістерезису. У цій зоні алгоритм не виконує жодних дій, навіть якщо Q коливається. Ширина зони у 0.30 одиниці Q обрана так, щоб дрібні природні коливання якості каналу не спричиняли безперервного перемикання параметрів. Без такої зони алгоритм реагував би на кожну незначну флуктуацію, що призводило б до нестабільної поведінки і зайвих витрат енергії на перемикання.

2.4.2 Ескалація параметрів за деградації якості

Перед описом правил ескалації слід уточнити робочий діапазон коефіцієнта розширення. Стандарт LoRa передбачає значення SF від 7 до 12. Однак у запропонованій реалізації робочий діапазон обмежено значеннями SF від 7 до 10. Причиною обмеження є ефірний час кадру. При SF11 і SF12 тривалість передачі одного пакета перевищує цикл опитування майстра, що становить 4 секунди, а отже ці два значення непридатні для обраного режиму

роботи. Таким чином, алгоритм оперує чотирма значеннями SF (7, 8, 9, 10) і потужністю передавача TxPower у діапазоні від 2 до 22 дБм із кроком 2 дБ. Верхня межа 22 дБм відповідає апаратному максимуму радіопідсистеми мікроконтролера STM32WL55 у режимі високої потужності та використовується у рамках лабораторного дослідження. Для польового розгортання у регіоні EU868 верхня межа TxPower обмежується нормами ETSI (+14 дБм EIRP) і встановлюється на рівні платформи незалежно від алгоритму адаптації.

Ескалація – це підсилення параметрів передавача у відповідь на погіршення якості каналу. Вона активується, коли Q опускається нижче порога Q_{low} , тобто канал переходить у низький або критичний стан. Ключовим принципом ескалації є правило *TxPower-first*: спершу вичерпується запас потужності передавача, і лише потім підвищується коефіцієнт розширення.

Підставою для такого порядку є вартість двох механізмів підсилення. Підвищення TxPower на один крок збільшує енергоспоживання передавача, але не змінює тривалості кадру. Підвищення SF подвоює ефірний час пакета, що збільшує і енергію на передачу, і затримку, і ймовірність колізій. Тому потужність є дешевшим ресурсом, і її доцільно використовувати першою.

У низькому стані ($Q_{crit} \leq Q < Q_{low}$) алгоритм діє так. Якщо потужність передавача ще не досягла максимуму, вона підвищується на один крок 2 дБ, а коефіцієнт розширення залишається незмінним. Якщо потужність уже на максимумі, але коефіцієнт розширення ще не досяг верхньої межі робочого діапазону, на один крок підвищується SF. Якщо ж вичерпано обидва ресурси, алгоритм не виконує дій, оскільки підсилювати більше нічим.

Окремо обробляється критичний стан ($Q < Q_{crit}$). Тут правило *TxPower-first* не застосовується, оскільки канал утратив працездатність, і поступове підсилення лише потужності було б надто повільним. Натомість алгоритм одночасно підвищує і коефіцієнт розширення на один крок, і потужність передавача на один крок (за умови, що для обох параметрів є запас). Одночасна зміна обох параметрів дає максимально швидкий вихід із критичного стану. Підвищення SF додає запас на декодування, а підвищення TxPower – запас за

рівнем сигналу. Якщо коефіцієнт розширення вже на максимумі робочого діапазону, ескалація переходить на правило ескалації низького стану, тобто підвищується лише потужність до досягнення апаратної межі.

2.4.3 Деескалація параметрів за відновлення якості

Деескалація – це зворотний процес, тобто зниження параметрів передавача, коли якість каналу відновилося і має надлишковий запас. Вона активується, коли Q піднімається вище порога Q_{high} , тобто канал переходить у високий стан. Мета деескалації – знизити енергоспоживання і скоротити ефірний час, не жертвуючи надійністю.

Порядок деескалації зворотний до порядку ескалації. Спершу знижується коефіцієнт розширення, і лише коли він досяг мінімального значення – знижується потужність передавача. Логіка та сама, що й при ескалації, але у дзеркальному відображенні. Коефіцієнт розширення є дорожчим ресурсом за ефірним часом, тому його доцільно знижувати першим, щоб якнайшвидше скоротити тривалість кадру.

Проте деескалація потенційно небезпечніша за ескалацію оскільки помилкове зниження параметрів у каналі, що лише здається стабільним, призведе до втрати пакетів. Тому деескалація захищена додатковою умовою – SNR-запобіжником. Перш ніж знизити будь-який параметр, алгоритм перевіряє, чи нормалізоване відношення сигнал/шум S перевищує верхній поріг Q_{high} . Якщо S нижче або дорівнює цьому порогу, деескалація блокується, навіть якщо саме Q перевищує Q_{high} .

Потреба у SNR-запобіжнику пов'язана з особливістю функції Q . Канал може бути деградованим, але стабільним. Наприклад, при неперервній інтерференції рівень сигнал/шум залишається стабільно низьким. Стабільність означає малу дисперсію, тобто високе V . Воно у свою чергу здатне підняти Q вище Q_{high} , хоча канал насправді перебуває на межі працездатності. У такій ситуації лише сигнал/шум показує реальний ризик, і SNR-запобіжник блокує зниження параметрів.

2.4.4 Механізм блокування та захист від коливань

Навіть із зоною гістерезису між порогами зміни параметрів потребують додаткового захисту від надмірної частоти перемикань. Після кожної зміни алгоритму потрібен час, щоб оцінити її результат. Ковзне вікно ще містить переважно старі виміри, і ухвалене негайно наступне рішення спиралося б на застарілі дані.

Для цього застосовується механізм блокування. Після будь-якої зміни параметрів алгоритм робить паузу тривалістю 3 пакети, протягом якої нові рішення не ухвалюються. За цей час до ковзного вікна потрапляє кілька вимірів при нових параметрах. Цього достатньо, щоб помітити, чи піде Q у бажаному напрямку, але без зайвої затримки реакції на подальші зміни каналу. Тривалість 3 пакети однакова для ескалації і деескалації, що забезпечує симетричну обережність алгоритму при погіршенні і при відновленні якості каналу.

Окремий захист передбачено для виходу з критичного стану. Після критичної ескалації, коли коефіцієнт розширення було підвищено для подолання різкої деградації, діє додаткове обмеження тривалістю 30 пакетів. Протягом цього періоду коефіцієнт розширення захищений від зниження. Навіть якщо Q швидко піднімається вище Q_{high} , алгоритм утримує SF на досягнутому рівні. При цьому зниження потужності передавача залишається дозволеним. Сенс обмеження – не дати алгоритму одразу повернути коефіцієнт розширення назад і знову потрапити у деградований стан, якщо різке зростання Q виявилося нестійким.

Ще один випадок блокування пов'язаний із відмовою механізму встановлення параметрів. Якщо серія спроб передати нові параметри підлеглому вузлу вичерпана без підтвердження, активне блокування скорочується до 2 пакетів (якщо воно перевищувало це значення), після чого алгоритм може повторити спробу адаптації. Скорочення замість встановлення фіксованої тривалості зумовлене тим, що під час невдалих спроб уже пройшов певний час, протягом якого ковзне вікно частково оновилося, тож утримувати повний lockout недоцільно. Це запобігає безперервним невдалим перемиканням, коли канал

тимчасово не дозволяє навіть доставити команду зміни параметрів, але водночас дозволяє якомога швидше відреагувати на стабілізацію каналу.

Сукупно описані правила – порогова класифікація, ескалація за принципом *TxPower-first*, дзеркальна деескалація із SNR-запобіжником та механізм блокування утворюють завершену логіку прийняття рішень

2.5 Висновки до розділу

У розділі сформульовано задачу адаптивного керування параметрами радіопротоколу для low-power IoT систем у динамічному радіосередовищі. Об'єктом адаптації обрано два параметри фізичного рівня: коефіцієнт розширення SF і потужність передавача TxPower. Задачу формалізовано через пошук відображення стану каналу у вектор параметрів передавача, що максимізує надійність доставки при мінімально можливих витратах енергії. Як підхід до її розв'язання запропоновано метод, що базується на скалярній багатокритеріальній функції якості каналу $Q(t)$ із значеннями у діапазоні $[0, 1]$. Метод отримав назву Q-ADR (Quality-based Adaptive Data Rate).

Відібрано чотири метрики, що становлять основу функції якості: рівень сигналу RSSI, відношення сигнал/шум SNR, частку успішних доставок PDR та дисперсію SNR. Для кожної з метрик окремо обґрунтовано її роль у характеристиці стану каналу: RSSI як показник бюджету радіоканалу, SNR як показник якості декодування, PDR як прямий вимір надійності, дисперсія SNR як індикатор стабільності каналу у часі. Інші метрики, що зустрічаються у літературі, розглянуто і свідомо виключено зі складу Q з обґрунтуванням у кожному випадку.

Побудовано функцію якості каналу як зважену суму нормалізованих метрик у формі $Q(t) = \alpha R + \beta S + \gamma P + \delta V$, де нормалізовані величини R , S , P та V приведено до спільної шкали $[0, 1]$. Вагові коефіцієнти прийнято такими: $\alpha = 0.10$, $\beta = 0.40$, $\gamma = 0.30$, $\delta = 0.20$. Розподіл ваг відображає ієрархію інформативності метрик з пріоритетом SNR як безпосереднього індикатора декодованості та PDR

як підсумкового показника надійності. Окремо опрацьовано випадок утрачених пакетів через імпутацію найгіршим значенням. Для кожного тайм-ауту у ковзне вікно спостереження підставляються $RSSI = -130$ дБм і $SNR = -20$ дБ. Завдяки цьому одна втрата пакета впливає на Q одночасно через три канали – безпосередньо через PDR, через середні значення метрик і через дисперсію.

Розроблено логіку прийняття рішень, що відображає поточне значення Q у дискретний стан каналу за допомогою трьох порогів: $Q_{crit} = 0.40$, $Q_{low} = 0.60$ та $Q_{high} = 0.90$. Для ескалації параметрів за деградації якості сформульовано правило *TxPower-first*: спершу вичерпується запас потужності передавача, і лише потім підвищується коефіцієнт розширення. Для критичного стану передбачено одночасне підвищення обох параметрів. Деескалація за відновлення якості виконується у дзеркальному порядку та захищена SNR-запобіжником, що блокує зниження параметрів у каналах з хибною стабільністю. Передбачено механізм блокування на 3 пакети після кожної зміни, додатковий захист SF тривалістю 30 пакетів після критичної ескалації та коротке блокування на 2 пакети у випадку відмови підтвердження нових параметрів. Робочий діапазон у реалізації обмежено значеннями SF від 7 до 10 і TxPower від 2 до 22 дБм з кроком 2 дБ.

Розроблений метод Q-ADR описано на рівні алгоритму. Для оцінки його практичної застосовності і порівняння зі стандартним механізмом ADR та конфігурацією з фіксованими параметрами метод необхідно реалізувати на цільовій апаратній платформі та провести експериментальне дослідження у контрольованих умовах. Опис програмно-апаратної реалізації Q-ADR на мікроконтролері STM32WL55 наведено у наступному розділі.

РОЗДІЛ 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ МЕТОДУ Q-ADR НА ПЛАТФОРМІ STM32WL55

3.1 Архітектура експериментальної установки

Раніше розроблений метод Q-ADR описано на рівні алгоритму, тобто у формі правил перетворення стану каналу у параметри передавача. Перевірка ефективності таких правил вимагає експериментального дослідження, у якому алгоритм взаємодіє з реальною радіочастотною обстановкою. Симуляційні засоби, такі як NS-3 або LoRaSim, обмежуються спрощеними моделями каналу і не відтворюють низку практично важливих явищ – тимчасові завади від сусідніх радіопристроїв, відбиття від поверхонь, нелінійності апаратної частини приймача. Тому для оцінювання Q-ADR обрано підхід натурного експерименту на фізичній платформі.

Експериментальна установка проєктувалась так, щоб одночасно забезпечувати три властивості: повторюваність умов запуску, спостережуваність поведінки алгоритму у реальному часі та керованість зовнішніх завад. Повторюваність потрібна для коректного порівняння в однакових умовах трьох алгоритмів – запропонованого Q-ADR, стандартного ADR і конфігурації з фіксованими параметрами. Спостережуваність забезпечується детальним логуванням кожного пакета та рішення алгоритму. Керованість зовнішніх завад досягається через окремий вузол, що виконує роль *interferer* (генератор завад) за заданим сценарієм. Основні дані для оцінювання алгоритму збираються самою установкою через логування на головному (*master*) і підлеглому (*slave*) вузлах. Незалежний контроль за фізичним станом ефіру здійснюється програмно-керованим радіо HackRF One, що дозволяє у реальному часі верифікувати факт випромінювання від *interferer* та коректність частотних параметрів передачі.

Топологію установки організовано як трирольову систему. Перший вузол виконує роль *master*: він є джерелом пакетів даних, отримувачем підтверджень і єдиним місцем виконання алгоритму адаптації. Другий вузол виконує роль *slave*: він приймає пакети від *master* і відправляє у відповідь підтвердження АСК.

Третій вузол виконує роль *interferer*. Він не бере участі у корисному обміні даними, а лише генерує радіозавади у тому самому частотному діапазоні. Такий поділ ролей дозволяє чітко розмежувати функції. Пара *master* і *slave* формують point-to-point з'єднання, на якій тестується алгоритм адаптації. В свою чергу *interferer* імітує зовнішнє джерело завад, типово для LoRa-мереж у середовищах з багатьма радіопристроями.

У фізичному вимірі *master* і *slave* розміщені у типовому офісному радіосередовищі з присутністю стін, перегородок і фонові активності інших радіопристроїв на сусідніх частотах. Вузол *interferer* розташовано між парою *master–slave*, щоб його сигнал створював завади однаково для обох напрямків передачі – як від *master* до *slave*, так і назад при передачі підтвердження.

Установка є лабораторною, призначеною для дослідницьких цілей. У ній регуляторні обмеження окремих регіонів не застосовуються, що дозволяє використовувати повний апаратний діапазон потужності передавача до +22 дБм, як зазначено у підрозділі 2.4. У реальній експлуатації системи верхня межа потужності встановлюється на рівні платформи відповідно до чинних норм регіону незалежно від алгоритму адаптації.

3.2 Апаратна платформа

Вибір платформи зумовлений властивостями, що були описані раніше: повторюваність умов запуску, спостережуваність поведінки алгоритму та керованість зовнішніх завад. Платформа має забезпечувати ідентичність радіопідсистеми на всіх трьох вузлах, достатній динамічний діапазон потужності передавача для роботи методу Q-ADR та програмне зчитування значень RSSI і SNR для кожного прийнятого пакета. Розглянуто мікроконтролер, налагоджувальну плату, антенну систему та конфігурацію радіоканалу.

Як обчислювальну платформу обрано мікроконтролер STM32WL55JC компанії STMicroelectronics [20]. Принципова особливість цього мікроконтролера – інтегрована радіочастотна частина на базі трансивера

SX126x. Завдяки інтеграції в установці відсутня шина SPI між обчислювальним ядром і радіо, тому усувається додаткове джерело випадкових затримок і спотворень команд керування передавачем. Крім того, радіопідсистема підтримує режим високої потужності HP PA з вихідним рівнем до +22 дБм, що відповідає верхній межі параметра TxPower для алгоритмів адаптації. Низькоспоживальний профіль із режимами Stop і Standby робить платформу придатною для батарейного живлення, типового для IoT-сценаріїв, на які орієнтований запропонований метод.

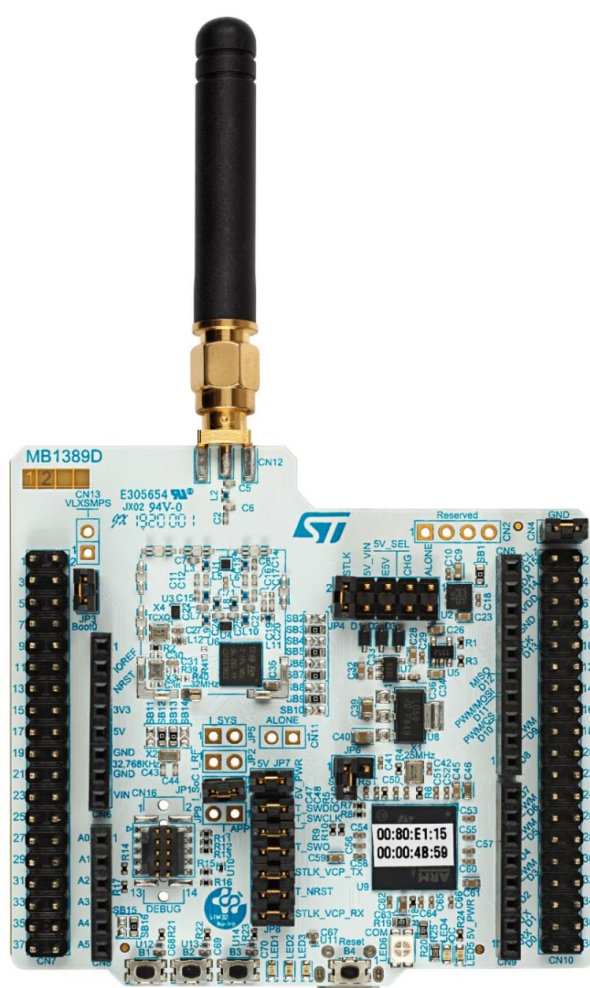


Рисунок 3.1 – Зовнішній вигляд плати NUCLEO-WL55JC1 [19]

Як апаратний носій використано налагоджувальну плату NUCLEO-WL55JC1 [19]. Плата містить мікроконтролер STM32WL55JC з ядром Arm Cortex-M4 на частоті 48 МГц, 256 КБ Flash-пам'яті та 64 КБ RAM. На платі

інтегровано програматор та модуль налагодження ST-LINK, а також інтерфейс UART, що використовується для логування у реальному часі. В установці задіяно три ідентичні плати для кожної з ролей: *master*, *slave* та *interferer*. Ідентичність апаратної частини гарантує, що відмінності у поведінці вузлів зумовлені лише прошивкою та алгоритмом, а не фізичними розбіжностями платформи. Зовнішній вигляд плати показано на рисунку 3.1.

На кожній платі встановлено штатну антену з підсиленням приблизно +2 дБі, що йде у комплекті з NUCLEO-WL55JC1 [19]. Ідентичність антен забезпечує симетрію радіоканалу між *master* і *slave* та однаковий вплив завад від *interferer* на обидва напрямки передачі.

Параметри радіоінтерфейсу для пари *master* – *slave* наведено у таблиці 3.1. Центральна частота 868.0 МГц обрана у межах діапазону EU868, призначеного для неліцензованих застосувань. Ширина смуги 125 кГц відповідає типовій конфігурації LoRaWAN [4]. Швидкість кодування 4/5 встановлена як мінімальна, що дозволяє оцінювати PDR переважно як характеристику стану каналу, а не наслідок дії корекції помилок.

Таблиця 3.1 – Параметри радіоінтерфейсу для пари *master* – *slave*

Параметр	Значення
Центральна частота	868.0 МГц
Ширина смуги	125 кГц
Швидкість кодування	4/5
Преамбула	8 символів
Слово синхронізації	0x1424
Розмір корисного навантаження	32 байти
Початковий SF	7
Початкова потужність передавача	10 дБм

Слово синхронізації 0x1424 ізолює мережу від трафіку публічних LoRaWAN-пристроїв, які використовують стандартне значення 0x3444 [4]. Розмір корисного навантаження 32 байти обрано таким, що вміщує заголовок протоколу і службові поля з резервом на майбутні розширення.

Початкові значення параметрів передавання встановлено як $SF=7$ і $TxPower=10$ дБм. $SF=7$ відповідає мінімуму робочого діапазону, окресленого у підрозділі 2.4, тому алгоритм має повний простір для ескалації коефіцієнта розширення при погіршенні каналу. Потужність 10 дБм лежить у середині діапазону від 2 дБм до 22 дБм, тому її можна адаптувати в обидва боки. Перед кожним експериментальним сценарієм початкові значення скидаються для забезпечення відтворюваності результатів.

Interferer реалізований на тій самій платі NUCLEO-WL55JC1, що й *master* та *slave*. Апаратно вузли однакові, відмінність полягає лише у прошивці. В експериментах було задіяно два сценарії постановки завад. Перший сценарій імітує роботу суміжних LoRa-пристроїв за допомогою ідентичної модуляції. Другий сценарій моделює сторонній шум у робочому діапазоні шляхом використання FSK-модуляції. Часові характеристики завад, паузи між ними та рівні потужності для обох сценаріїв будуть описані далі.

3.3 Архітектура прошивки

Прошивка вузлів написана мовою C за допомогою STM32CubeIDE та бібліотек HAL від STMicroelectronics. Архітектура проєкту передбачає суворе розмежування функцій між радіомодулем, протоколом, системою логування та блоком адаптації. Такий модульний підхід дозволяє гнучко замінювати алгоритми керування (Q-ADR, стандартний ADR або фіксовану конфігурацію) без переписування загального коду. Це забезпечує чисті умови для подальших експериментів, оскільки всі досліджувані алгоритми працюють на базі єдиного стеку протоколів та однакових налаштувань радіоефіру

В основі програмного забезпечення вузлів лежить концепція рівневої архітектури. Базовий (нижній) рівень сформовано з немодифікованих стандартних бібліотек STMicroelectronics: HAL для керування периферією мікроконтролера (GPIO, UART, таймери) та SubGHz Radio API для взаємодії із вбудованим sub-GHz трансивером. Прямий доступ до цього рівня обмежено проміжним модулем абстракції радіо *lora_app.c*. Він інкапсулює низькорівневі Semtech-специфічні функції та надає вищим рівням уніфікований інтерфейс для приймання й передачі пакетів. Завдяки такій ізоляції протокольний та прикладний рівні оперують виключно логічними кадрами LoRa, ролями вузлів і командами керування, будучи повністю незалежними від апаратних нюансів платформи.

Точкою входу прикладного рівня є модуль *node.c*, який виконує функцію диспетчера конфігурації пристрою. Усі три типи вузлів (*master*, *slave* та *interferer*) компілюються з єдиної кодової бази. Конкретний функціонал визначається на етапі збирання проєкту за допомогою макросів препроцесора *ROLE_MASTER*, *ROLE_SLAVE* або *ROLE_INTERFERER*, які задаються в налаштуваннях середовища STM32CubeIDE окремо для кожної плати. Такий підхід унеможливорює випадкову зміну логіки роботи пристрою під час виконання експерименту, а також гарантує використання спільної реалізації радіотракту та єдиних протокольних структур даних.

Реалізації ролей винесено в окремі модулі. Модуль *master.c* реалізує головний вузол з циклом опитування і викликами алгоритму адаптації. Модуль *slave.c* реалізує підлеглий вузол з прийманням пакетів та надсиланням підтверджень. Модуль *interferer.c* реалізує генератор завад з двома режимами роботи. Кожен з цих модулів є самодостатнім: він взаємодіє з нижніми рівнями через *lora_app.c* та з рівнем логування, але не звертається до коду інших ролей. Модуль *node.c* переадресовує події радіо (приймання, передача, тайм-аути) до відповідної реалізації ролі через стандартний набір функцій-обробників, спільний для всіх трьох ролей.

Алгоритм адаптації параметрів виокремлено в окремий модуль *algo.c*. Замість трьох різних копій модуля *master* (по одній на кожен досліджуваний алгоритм) реалізовано єдиний *master*, що отримує результат рішення від *algo.c* через єдиний інтерфейс. Цей інтерфейс складається з функції *Algo_OnPacket*. Вона приймає поточні значення SF та TxPower, останні виміри RSSI і SNR, ознаку доставки пакета й режим алгоритму (*ALGO_FIXED*, *ALGO_ADR* або *ALGO_QADR*), а повертає рекомендовані нові значення SF та TxPower. Активний алгоритм обирається на етапі компіляції через значення макроса *DEFAULT_ALGO_MODE*, що передається у *Algo_OnPacket* кожним викликом. Завдяки цьому всі три алгоритми оперують одними й тими ж структурами даних, отримують один і той самий потік подій від радіо й логуються однаково. Така ізоляція є необхідною умовою методологічної коректності експерименту. Будь-яка різниця у результатах між трьома прогонами не може бути пояснена різницею у протоколі чи інфраструктурі логування, а пояснюється лише різницею у самому алгоритмі.

Загальну структуру прошивки з міжмодульними залежностями наведено на рисунку 3.2.

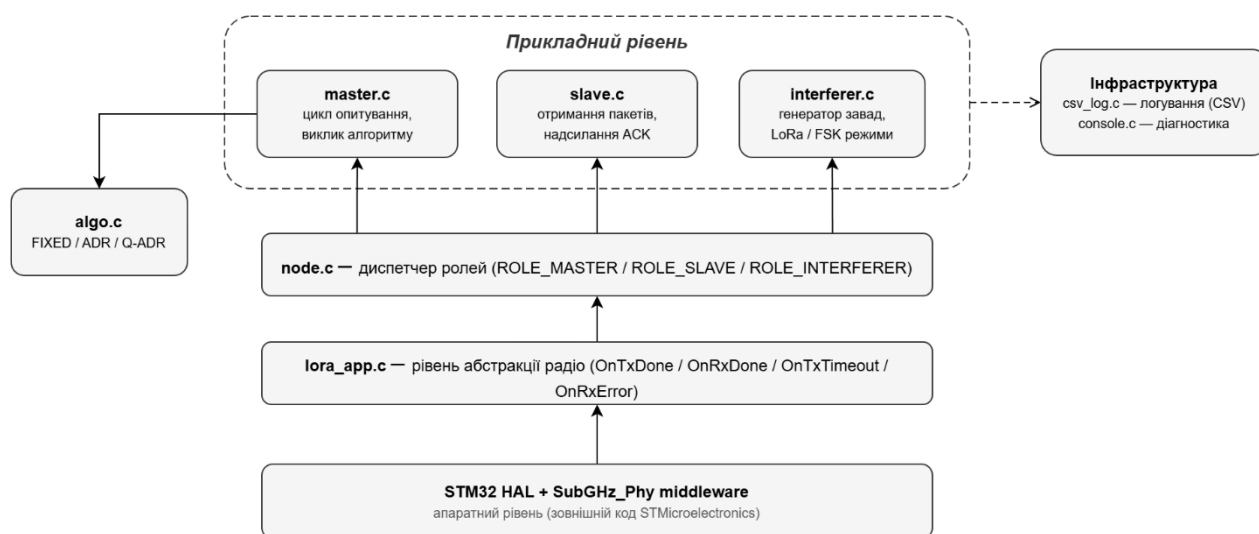


Рисунок 3.2 – Структура модулів прошивки вузла

Інфраструктурну частину прошивки складають два модулі. Модуль *csv_log.c* виконує детальне логування у форматі CSV кожної протокольної події і кожного рішення алгоритму через інтерфейс UART на швидкості 115200 біт/с. Кожен запис містить часову мітку, ідентифікатор пакета, поточні значення SF та TxPower, виміри каналу та результат рішення алгоритму. Це забезпечує повну відновлюваність картини експерименту під час фінального аналізу даних. Модуль *console.c* реалізує мінімальний інтерфейс логування для діагностичних команд і ручного запуску фаз експерименту під час етапу попереднього налагодження.

3.4 Реалізація протоколу обміну даними

Протокол обміну між *master* і *slave* є фундаментальним рівнем, над яким будується алгоритм адаптації. Він забезпечує надійну доставку пакетів з підтвердженням, передавання параметрів від *master* до *slave* під час адаптації та обробку відмов каналу. Цей підрозділ описує реалізацію протоколу у трьох аспектах: цикл обміну та фази передачі, формат кадру і механізм встановлення параметрів з обробкою помилок.

3.4.1 Цикл master–slave та фази обміну

Обмін між *master* і *slave* побудовано як циклічний процес з фіксованим періодом 4 секунди. У кожному циклі *master* ініціює одну спробу передачі пакета даних, після чого очікує підтвердження ACK від *slave*. Тривалість одного циклу обрано як компроміс між частотою адаптації та ефірним часом. Зазначений інтервал є достатнім для передавання та приймання пакетів навіть за максимального коефіцієнта розширення спектра SF10, але водночас дозволяє оцінювати реакцію алгоритму у прийнятному часовому масштабі експерименту.

Кожен цикл послідовно проходить три етапи: передачу кадру даних від *master* до *slave*, очікування підтвердження ACK з відповідним тайм-аутом та паузу для синхронізації часової структури циклу. Така структура забезпечує

однаковий темп обміну незалежно від поточних параметрів радіо, що спрощує порівняння поведінки трьох алгоритмів адаптації в експерименті.

Тривалість фази передачі визначається часом перебування кадру в ефірі (time-on-air), що залежить від коефіцієнта розширення SF, ширини смуги, швидкості кодування та розміру пакета. Тайм-аут очікування ACK обрано індивідуально для кожного SF на основі очікуваного часу обробки пакета підлеглим вузлом і передавання підтвердження у зворотний бік. Фаза паузи у реалізації окремо не контролюється – вона визначається як залишок 4-секундного циклу після завершення TX і отримання ACK (або спрацювання тайм-ауту). Конкретні значення тривалостей фаз для робочого діапазону SF7–SF10 наведено у таблиці 3.2.

Таблиця 3.2 – Тривалості фаз циклу master–slave для діапазону SF

SF	Time-on-air, мс	Тайм-аут ACK, мс	Пауза, мс
7	72	300	3556
8	134	400	3332
9	247	700	2806
10	453	1200	1894

З таблиці видно, що при переході від SF7 до SF10 time-on-air зростає приблизно у шість разів (з 72 до 453 мс), а тайм-аут очікування ACK – у чотири рази (з 300 до 1200 мс). Зі свого боку, пауза скорочується від майже 3,6 секунди при SF7 до менш ніж 1,9 секунди при SF10. Саме обмеження паузи знизу не дозволяє включити SF11 і SF12 у робочий діапазон, як було зазначено раніше.

У штатному режимі цикл завершується успішним прийманням ACK на стороні *master*. У разі виникнення тайм-ауту *master* переходить безпосередньо до фази паузи без дублювання поточного кадру, оскільки механізм повторних спроб передбачений виключно для команди *SET_PARAMS*.

3.4.2 Формат кадру

Кадр LoRa у прошивці має фіксований розмір 32 байти, що відповідає значенню константи `LORA_PAYLOAD_LEN` з *protocol.h*. Фіксований розмір спрощує обробку у приймачі і виключає необхідність читання заголовка LoRa з полем довжини. Кадр даних складається з двох основних частин: загального заголовка завдовжки 10 байтів, який присутній у пакетах усіх типів, та корисного навантаження обсягом 22 байти, формат якого визначається кодом команди.

Заголовок містить службові поля, необхідні для адресації та контексту передачі. Структура заголовка наведена у таблиці 3.3.

Таблиця 3.3 – Структура заголовка кадру

Поле	Зміщення	Розмір, байт	Опис
node_id	0	1	Ідентифікатор вузла-відправника
algo_mode	1	1	Режим алгоритму (FIXED/ADR/Q-ADR)
packet_id	2	2	Порядковий номер пакета
cmd	4	1	Код команди (PROBE/ACK/SET_PARAMS)
snr	5	1	SNR на стороні відправника, дБ
sf	6	1	Поточний коефіцієнт розширення
tx_power	7	1	Поточна потужність передавача, дБм
rssr	8	2	RSSI на стороні відправника, дБм

Поля *sf*, *tx_power*, *snr* і *rssr* передають метричну інформацію про стан каналу з боку відправника. Завдяки цьому *master* отримує безпосереднє свідчення про якість каналу від підлеглого вузла, а не лише вимірює його зі свого боку, що дає повну картину для обчислення функції якості $Q(t)$.

Корисне навантаження кадру (22 байти) поділяється на дві частини: спільний для всіх команд блок `payload` довжиною 4 байти і блок навантажувальних даних довжиною 18 байт. Структура блоку `payload` реалізована як C-union *RadioPayload_t*, що дозволяє інтерпретувати одну й ту саму ділянку пам'яті різними способами залежно від коду команди. Варіанти структури `payload` для трьох команд протоколу наведено у таблиці 3.4.

Таблиця 3.4 – Структура поля `payload` для трьох команд протоколу

Команда	Код	Поля <code>payload</code>	Призначення
CMD_PROBE	0x01	<code>dummy[4]</code> : 4 байти доповнення	Передача пакета даних від <i>master</i> до <i>slave</i>
CMD_ACK	0x02	<code>slave_rssi</code> (int16), <code>slave_snr</code> (int8), <code>rx_count</code> (uint8)	Підтвердження від <i>slave</i> з метриками каналу
CMD_SET_PARAMS	0x03	<code>new_sf</code> (uint8), <code>new_txpwr</code> (int8), <code>pad[2]</code>	Передача нових параметрів від <i>master</i> до <i>slave</i>

Команда CMD_PROBE відповідає звичайному пакету даних. Блок `payload` в даній команді не несе службової інформації, а лише служить заповнювачем фіксованої довжини. Команда CMD_ACK передає від *slave* до *master* безпосередні виміри каналу з боку приймача: рівень сигналу `slave_rssi`, відношення сигнал/шум `slave_snr` та кумулятивну кількість успішно прийнятих пакетів `rx_count`. Поле `rx_count` резервується у структурі кадру для незалежної верифікації PDR з боку *slave* під час аналізу логів і не використовується самим алгоритмом адаптації. Команда CMD_SET_PARAMS передає від *master* до *slave* нові значення коефіцієнта розширення `new_sf` і потужності передавача `new_txpwr`, які *slave* має застосувати для наступного циклу обміну.

3.4.3 Механізм SET_PARAMS та обробка відмов

Окремим викликом протоколу є передача нових параметрів адаптації від *master* до *slave*. Це нетривіальна задача, оскільки після успішного обчислення нових значень SF і TxPower на стороні *master* обидва вузли повинні узгоджено перейти на нову конфігурацію. Якщо *slave* не отримає команду SET_PARAMS, він продовжить очікувати пакети на старих параметрах, тоді як *master* вже передає на нових – відбудеться втрата зв'язку. Тому передача SET_PARAMS реалізована із захистом від невдачі.

Для забезпечення стійкості процесу конфігурування та гарантованої доставки службових повідомлень в архітектурі протоколу закладено кілька рівнів завадостійкості. Основу цього захисту становить механізм автоматичних повторних спроб: за відсутності підтвердження ACK на запит SET_PARAMS у поточному циклі, *master* здійснює до 10 повторних передач кадру поспіль. Щоб унеможливити появу циклічних колізій із періодичними зовнішніми завадами, кожна наступна спроба супроводжується випадковим часовим зсувом у межах від 0 до 127 мс.

Ефективність такої серії повторів додатково підсилюється завдяки тимчасовому форсуванню вихідної потужності передавача. Незалежно від поточного рівня TxPower, який на даний момент визначено алгоритмом адаптації для звичайних даних, трансляція службової команди SET_PARAMS завжди виконується із граничною потужністю апаратної платформи (+22 дБм). Симетрично цей підхід застосовується і на стороні *slave*: підтвердження ACK на отриману команду SET_PARAMS також відсилається із потужністю +22 дБм. Таке рішення максимізує енергетичний бюджет радіолінії саме для критично важливих пакетів, не порушуючи при цьому фонові розрахунки та загальну логіку роботи алгоритму адаптації. Після завершення процедури узгодження TxPower на обох вузлах автоматично повертається до робочого значення.

Якщо ліміт у 10 спроб вичерпано без отримання підтвердження, система переходить до сценарію обробки критичного збою. Вузол *master* скасовує поточну спробу адаптації, відновлює попередню стабільну конфігурацію SF і

TxPower та сигналізує про невдачу через виклик функції *Algo_OnParamChangeFailed*. Це дозволяє оперативно скоротити блокування адаптаційних алгоритмів і прискорити наступну спробу підлаштування параметрів, як тільки умови каналу стабілізуються. Описана взаємодія гарантує цілісність методу Q-ADR навіть за тимчасової неможливості взаємної синхронізації вузлів.

Окремий випадок становить невдала зміна коефіцієнта розширення SF. У цій ситуації стан вузлів може непередбачувано розсинхронізуватися. Вузол *slave* міг отримати SET_PARAMS і застосувати новий SF після успішної передачі підтвердження, тоді як саме це підтвердження не дійшло до *master* через завади у зворотному напрямку. У такому випадку *master* вважатиме зміну невдалою і залишиться на старому SF, а *slave* перейде на новий, і зв'язок втратиться. Для запобігання цьому передбачено два компенсуючих засоби.

На стороні *master* після вичерпання повторів відправляється додатковий пакет SET_PARAMS з поточним (старим) значенням SF, що служить командою повернення для *slave*. На стороні *slave* запускається rollback-таймер тривалістю 60 секунд: якщо за цей час не прийде жодного пакета від *master*, *slave* самостійно повертається до попередньої конфігурації. Спільна дія цих двох механізмів гарантує, що пара *master-slave* протягом обмеженого часу повертається до синхронного стану.

3.5 Реалізація методу Q-ADR

Метод Q-ADR описано у Розділі 2 на рівні правил перетворення стану каналу у параметри передавача. Цей підрозділ описує реалізацію цих правил у модулі *algo.c* прошивки *master*. Інтерфейс модуля та його місце в архітектурі прошивки описано у підрозділі 3.3.

Тут розглядаються три аспекти: структури даних для зберігання історії вимірів, алгоритм обчислення функції якості $Q(t)$ та логіка прийняття рішень про зміну параметрів передавача.

3.5.1 Структури даних

Стан алгоритму зберігається у статичній структурі модуля *algo.c*. Структура містить три кільцеві буфери фіксованого розміру `QADR_WINDOW = 10` для зберігання історії вимірів за останні десять циклів обміну: буфер значень RSSI, буфер значень SNR і буфер ознак успішної доставки. Поряд із буферами зберігаються лічильник поточної позиції запису та лічильник фактичної заповненості вікна.

Буфер значень RSSI використовує тип `int16_t`, що покриває робочий діапазон LoRa-приймача від -130 дБм до -30 дБм. Буфер значень SNR має тип `int8_t`, оскільки робочий діапазон становить від -20 дБ до $+10$ дБ. Буфер ознак доставки реалізовано як масив булевих значень: для кожного слота вікна він маркує, чи був успішно отриманий ACK у відповідному циклі, чи спрацював тайм-аут.

Запис нового виміру у вікно виконує допоміжна функція `QadrPush`. Вона реалізує класичну схему кільцевого буфера зі сталою складністю $O(1)$: новий вимір переписує елемент у поточній позиції, лічильник позиції інкрементується за модулем `QADR_WINDOW`, а лічильник заповненості зростає до 10 і далі лишається на цьому значенні. Кільцевий буфер обрано замість простого масиву зі зсувом для уникнення непотрібного копіювання даних на кожному кроці.

Окремо від кільцевого буфера в модулі оголошено два лічильники блокувань. Перший зберігає значення звичайного `lockout` для тимчасової паузи після зміни параметрів. Другий зберігає значення затримки зниження SF після критичної ескалації. Також зберігаються кешовані значення останнього обчисленого Q , дисперсії SNR, PDR і трьох нормалізованих метрик. Кешування використовується одночасно для логування у CSV-форматі і для перевірки SNR-запобіжника на наступній ітерації функції рішень.

3.5.2 Обчислення функції якості $Q(t)$

Обчислення значення функції якості за поточним вмістом ковзного вікна виконує функція `QadrComputeQ`. Вона реалізує формулу (2.8) і логіку імпутації

втрачених пакетів з підрозділу 2.3. Обчислення розбито на два проходи по ковзному вікну, після яких іде етап нормалізації та зважена сума.

На першому проході підраховуються суми значень RSSI і SNR за десятима слотами. Для слотів з тайм-аутом, тобто тих, де ACK не був отриманий, у суми підставляються константи $RSSI_TIMEOUT = -130$ дБм і $SNR_TIMEOUT = -20$ дБ, тобто значення на межі непрацездатності каналу. Одночасно ведеться лічильник успішних ACK для подальшого обчислення PDR. Через імпутацію втрата одного пакета знижує Q одразу за трьома каналами: безпосередньо через PDR, через зміщення середніх значень RSSI і SNR та через збільшення дисперсії SNR.

Окремим граничним випадком є повна відсутність підтверджень у вікні. У такій ситуації обчислювати середні значення немає сенсу, тож функція повертає безпосередньо $Q = 0$, без подальших розрахунків. Усі кешовані метрики також обнуляються для коректного логування.

Після обчислення середніх виконується другий прохід для дисперсії SNR. Та сама логіка імпутації застосовується і тут: для слотів з тайм-аутом у формулу підставляється $SNR_TIMEOUT$ замість відсутнього виміру. Дисперсія обчислюється як популяційна, тобто ділиться на повний розмір вікна $QADR_WINDOW$, а не на $(W - 1)$ чи на кількість фактичних ACK. Завдяки цьому часткова деградація вікна одразу проявляється у компоненті V як зростання розкиду між реальними вимірами і підставленими штрафними значеннями.

Останнім етапом є нормалізація трьох метрик (середнього RSSI, середнього SNR і дисперсії) до спільної шкали $[0, 1]$ за формулами (2.5)–(2.7) і зважена сума за формулою (2.8). Допоміжна функція $ClampF$ обмежує проміжні значення заданим діапазоном на випадок виходу вимірів за межі робочого діапазону. Вагові коефіцієнти $\alpha, \beta, \gamma, \delta$ зі значеннями 0.10, 0.40, 0.30, 0.20 задано як константи модуля. Константа $V_MAX = 300$ задає рівень дисперсії, вище якого канал вважається повністю нестабільним.

Усі обчислення виконуються у float завдяки апаратній підтримці операцій з плаваючою комою у ядрі Arm Cortex-M4. Повний цикл обчислення Q

складається з двох проходів по десяти слотах та одного зваженого сумування. На частоті 48 МГц це займає десятки мікросекунд, що несуттєво на тлі періоду циклу *master* 4 секунди.

3.5.3 Реалізація логіки прийняття рішень

Дерево рішень з підрозділу 2.4 реалізує функція RunQADR. Порядок виконання прямий: декремент лічильника затримки зниження SF, обчислення Q, перевірка блокування, класифікація стану каналу за пороговими значеннями, виконання відповідної дії та встановлення наступного блокування.

Лічильник затримки декрементується на кожній ітерації незалежно від того, чи активне поточне блокування *lockout*. Через це 30-пакетне обмеження на зниження SF після критичної ескалації відраховується від моменту самої ескалації, а не від моменту виходу з блокування.

Перед прийняттям рішення алгоритм перевіряє наявність активного блокування *lockout*. Якщо лічильник додатний, рішення не виконується: лічильник зменшується на одиницю, функція повертає ознаку відсутності змін. Цей механізм захищає алгоритм від рішень на основі застарілих вимірів одразу після зміни параметрів.

Класифікація стану каналу виконується трьома послідовними порівняннями значення Q з пороговими константами $Q_CRITICAL = 0.40$, $Q_LOW = 0.60$ та $Q_HIGH = 0.90$. Критичний стан перевіряється першим, оскільки ця гілка має пріоритет над звичайною ескалацією.

У критичній гілці, коли $Q < Q_CRITICAL$, алгоритм одночасно підвищує SF на один крок і потужність передавача на 2 дБ. Гілка активується лише за наявності запасу до апаратного максимуму $SF_MAX = 10$. Підвищення потужності виконується лише за наявності запасу до апаратної межі $TX_MAX = 22$ дБм. Одночасно лічильник затримки встановлюється у значення $QADR_CRIT_HYSTERESIS = 30$ пакетів. Це блокує зворотне зниження SF до того, як підтвердиться стабільність каналу протягом наступних 30 циклів обміну.

Гілка низького стану, коли $Q_CRITICAL \leq Q < Q_LOW$, реалізує звичайну ескалацію за принципом *TxPower-first*, який був описаний раніше у Розділі 2.

Спершу у послідовних циклах підвищується потужність до апаратного максимуму $TX_MAX = 22$ дБм. І лише після вичерпання запасу потужності підвищується коефіцієнт розширення до $SF_MAX = 10$. Якщо обидва параметри вже досягли межі, алгоритм залишає поточну конфігурацію без змін.

Деескалація за відновлення якості, коли $Q > Q_HIGH$, виконується у дзеркальному порядку: спочатку зменшується SF , потім $TxPower$. Однією з умов деескалації є SNR -запобіжник. Перед будь-яким зниженням параметрів алгоритм перевіряє, чи нормалізоване значення SNR за поточне вікно перевищує поріг Q_HIGH . Якщо ні, деескалація блокується, навіть коли загальне Q вище порогу. Це захищає від помилкової деескалації у каналах з прихованою деградацією, де PDR і дисперсія тимчасово сприятливі, а абсолютний рівень сигналу залишається граничним. Перевірка використовує значення нормалізованого SNR , яке щойно обчислила функція $QadrComputeQ$ і зберегла у кешованій змінній модуля, без потреби повторного обчислення.

Окремий механізм передбачено для виходу з критичного стану. Поки лічильник затримки додатний, зниження SF блокується навіть за умови $Q > Q_HIGH$ і виконаному SNR -запобіжнику. Зниження потужності у цій ситуації залишається дозволеним. Це запобігає поверненню коефіцієнта розширення на попередній рівень одразу після короткочасного зростання Q , яке може виявитися нестійким на каналах з повторюваною деградацією.

3.6 Реалізація еталонів для порівняння

Для оцінювання методу Q-ADR у порівнянні з альтернативними підходами в системі реалізовано два алгоритми-еталони: режим з фіксованими параметрами передавача та стандартний механізм ADR за специфікацією Semtech. Обидва алгоритми реалізовано як окремі гілки публічного інтерфейсу модуля *algo.c*, описаного у підрозділі 3.3.

Найпростішим з еталонів є режим з фіксованими параметрами (ALGO_FIXED). Функція рішень для цього режиму повертає ознаку відсутності

змін незалежно від поточного стану каналу. Початкові значення $SF = 7$ і $TxPower = 10$ дБм, встановлені на старті прошивки *master*, лишаються незмінними протягом усього експерименту. Режим виконує роль контрольної конфігурації для оцінювання адаптивних алгоритмів.

Другий еталон – стандартний ADR за специфікацією Semtech, що реалізовано в режимі ALGO_ADR. Алгоритм базується на формулі NStep, яка визначає потрібну кількість кроків зміни параметрів за поточним рівнем відношення сигнал/шум у каналі. Він є найпоширенішим механізмом адаптації в LoRa-мережах і слугує природним базовим рівнем для порівняння.

Стан алгоритму ADR зберігається у статичній структурі модуля *algo.c* і містить кільцевий буфер останніх 20 значень SNR (константа $ADR_WINDOW = 20$), лічильник послідовних тайм-аутів та допоміжні поля для керування буфером. Розмір вікна 20 пакетів відповідає рекомендації Semtech і помітно перевищує вікно методу Q-ADR (10 пакетів). Більший розмір вікна забезпечує стандартному ADR високу стійкість до випадкових флуктуацій якості, але одночасно сповільнює реакцію алгоритму на швидкі зміни каналу.

Логіку прийняття рішень розбито на дві гілки відповідно до події. При успішному отриманні ACK обчислюється максимальне значення SNR у вікні. Від нього віднімається пороговий SNR для поточного SF (значення наведені у таблиці 3.5) та запас у 10 дБ (константа $ADR_MARGIN_DB = 10$). Отримана різниця ділиться на 3 і округлюється донизу. Це і є значення NStep. Якщо NStep додатне, алгоритм виконує деескалацію: спочатку послідовно знижує SF до мінімуму $SF_MIN = 7$, після чого знижує TxPower крок за кроком. Якщо NStep від'ємне, виконується протилежна послідовність ескалації. Якщо NStep нульове, параметри лишаються без змін.

Значення взято зі специфікації Semtech [7] для робочого діапазону SF, обраного в підрозділі 2.4.2. За цим діапазоном чутливість декодування зростає на 2.5 дБ при кожному збільшенні SF на одиницю, що відображає логарифмічну природу енергетичного виграшу від розширення спектра. У реалізації значення

таблиці округлено до цілих відповідно до типу `int8_t`, в якому зберігається порогова чутливість для кожного SF.

Таблиця 3.5 – Порогові значення SNR для декодування LoRa-кадрів

Коефіцієнт розширення (SF)	Мінімальний SNR для декодування, дБ
7	-7.5
8	-10.0
9	-12.5
10	-15.0

Окрема гілка спрацьовує у разі тайм-ауту замість ACK. Кожний тайм-аут інкрементує лічильник послідовних втрат. Коли лічильник досягає порогу `ADR_ACK_LIMIT = 8`, алгоритм форсовано виконує одну ескалаційну дію: підвищує `TxPower` на крок, а у випадку досягнення апаратного максимуму – підвищує SF. Після кожної форсованої ескалації вміст SNR-буфера очищується для запобігання прийняттю рішень на основі застарілих вимірів за попередніх параметрів. Лічильник тайм-аутів скидається після кожної успішної доставки ACK або після форсованої ескалації.

Метод Q-ADR і стандартний ADR відрізняються у кількох принципових аспектах. Стандартний ADR оперує лише однією метрикою якості каналу, а саме рівнем SNR, і не враховує абсолютний рівень сигналу RSSI, фактичну успішність доставки PDR та стабільність каналу через дисперсію SNR. Тайм-аут пакета у ADR не впливає на NStep-формулу і фактично ігнорується алгоритмом до накопичення восьми послідовних втрат, тоді як Q-ADR одразу знижує функцію якості через імпутацію найгіршими значеннями. Деескалація у ADR не має додаткового захисту, тож при підвищенні SNR алгоритм може спустити параметри до мінімуму навіть у каналах з прихованою деградацією. Q-ADR блокує таку деескалацію через SNR-запобіжник. Стандартний ADR також не передбачає окремого захисту від повторного попадання у деградований стан

після ескалації, тоді як Q-ADR блокує зниження коефіцієнта розширення протягом 30 пакетів після критичної ескалації. Окрім того, ADR використовує вікно у 20 пакетів проти 10 у Q-ADR, що подвоює мінімальний час до першого рішення про адаптацію. Описані відмінності становлять основу для експериментального дослідження ефективності трьох реалізованих алгоритмів.

3.7 Інфраструктура експерименту

Цей підрозділ описує допоміжну інфраструктуру експериментальної установки, що не входить безпосередньо до алгоритмів адаптації, але є необхідною для проведення повторюваних експериментів і подальшого аналізу даних. Розглядаються три аспекти: реалізація генератора завад на третьому вузлі мережі, формат і механізм логування протокольних подій та програмна автоматизація проведення експериментальних прогонів.

3.7.1 Генератор завад

Третій вузол експериментальної установки виконує роль *interferer*, тобто генератора керованих радіозавад у тому самому частотному діапазоні, у якому працює пара *master-slave*. Він реалізований на ідентичній платі NUCLEO-WL55JC1, що і два інші вузли, але з прошивкою, скомпільованою з макросом `ROLE_INTERFERER`. Сам *interferer* не бере участі у протокольному обміні. Його єдиною функцією є періодичне випромінювання радіокадрів за заданим часовим профілем.

У роботі реалізовано два сценарії постановки завад. Перший сценарій (`LORA_RANDOM`) імітує роботу суміжних LoRa-вузлів за допомогою ідентичної модуляції на тих самих параметрах, що і робоча пара. У цьому режимі *interferer* передає кадри тривалістю 400 мс, після чого витримує паузу випадкової тривалості від 100 до 600 мс перед наступним кадром. Випадкова пауза наближає поведінку до реалістичного середовища з декількома незалежними LoRa-пристроями, що працюють у тому самому каналі. Цей сценарій впливає

переважно на нижні значення SF, де time-on-air короткий і часові вікна між кадрами *master* можуть припадати на паузи між імпульсами завад.

Другий сценарій (FSK_CONTINUOUS) моделює сторонній шум у робочому діапазоні через використання FSK-модуляції замість LoRa. На відміну від LoRa, FSK не використовує розширення спектра і має значно більшу спектральну щільність потужності на однаковій смузі частот. У цьому режимі *interferer* передає імпульси тривалістю 5 секунд з паузою у 100 мс між ними, тобто майже неперервне випромінювання. Цей сценарій впливає переважно на високі значення SF, де time-on-air *master* перевищує 200 мс і повністю потрапляє в імпульс завад.

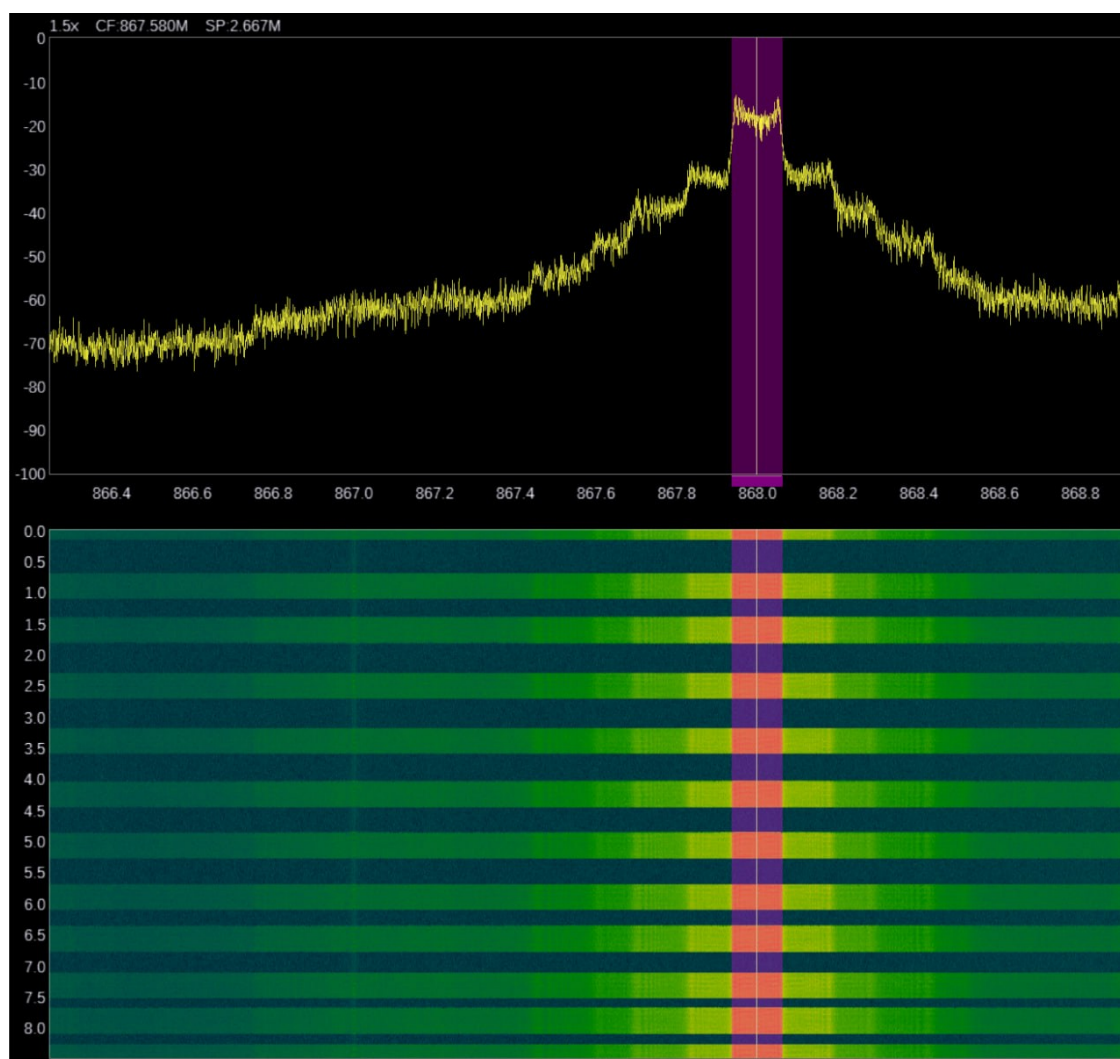


Рисунок 3.3 – Спектр HackRF з активним *interferer* вузлом у режимі LORA_RANDOM

В обох сценаріях *interferer* передає на потужності +16 дБм. Це значення обрано як середнє між робочою потужністю *master* і покликане імітувати завади, які за рівнем сигналу можуть бути порівнянними з корисним сигналом, але не повністю його перекривати. Перевірка фактичного випромінювання у заданих режимах здійснюється незалежно через програмно-кероване радіо HackRF One, що візуалізує спектр у діапазоні 868 МГц у реальному часі. Скріншоти спектру у двох режимах наведено на рисунках 3.3 і 3.4.

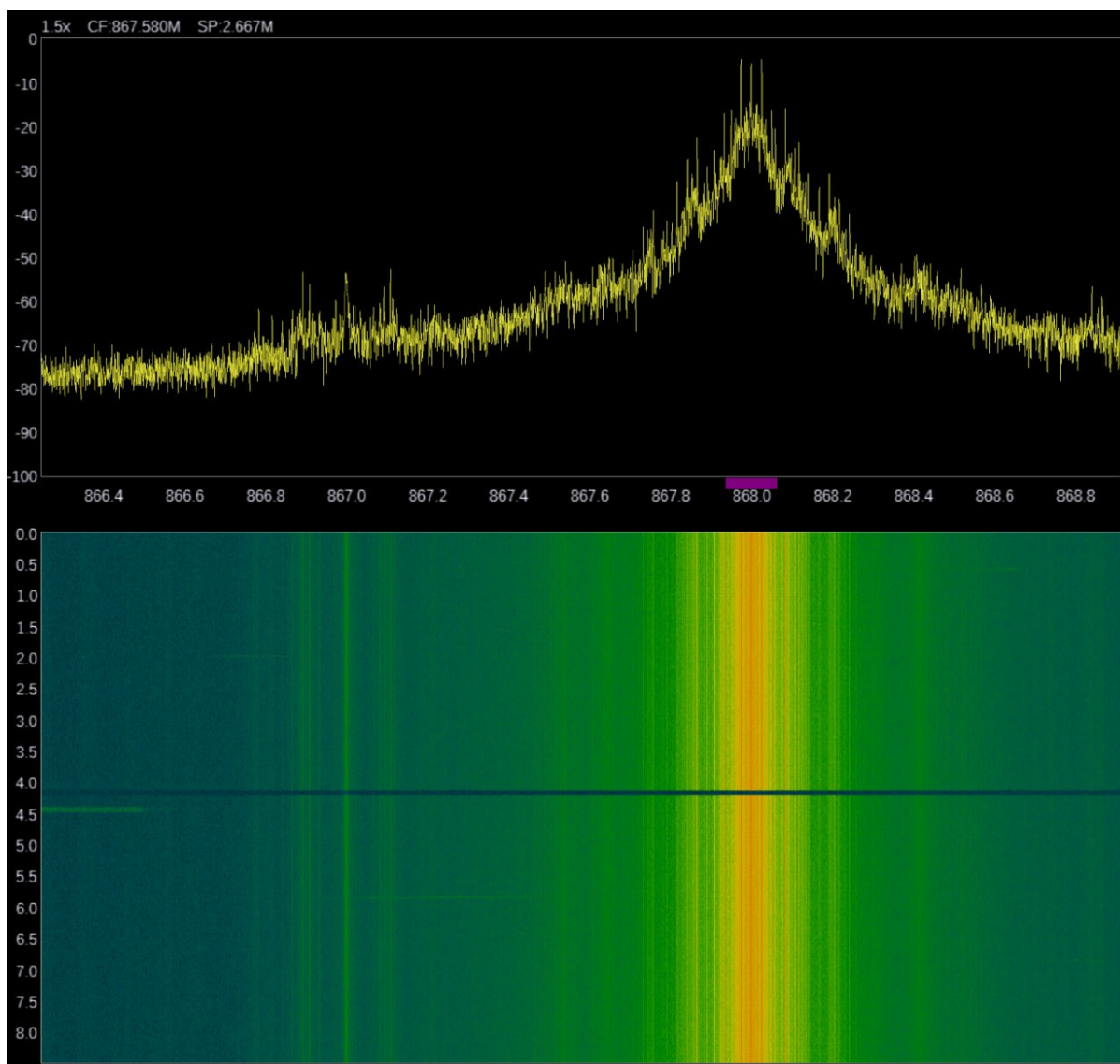


Рисунок 3.4 – Спектр HackRF з активним *interferer* вузлом у режимі FSK_CONTINUOUS

3.7.2 Логування подій

Усі три алгоритми адаптації використовують ту саму інфраструктуру логування для забезпечення можливості порівняння експериментальних результатів. Логування виконує модуль *csv_log.c*, який було згадано при описі архітектури проєкту. Кожен вузол виводить записи через інтерфейс UART на швидкості 115200 біт/с, після чого кожен рядок одразу зберігається у вихідний текстовий файл. Кожен рядок лога являє собою одну подію у форматі CSV з фіксованою послідовністю полів. Перший рядок файлу містить імена стовпців і записується одноразово під час старту прошивки. Кожна подальша подія додається як окремий рядок, що дозволяє завантажувати лог у аналітичні інструменти без попередньої обробки. Структуру полів лог-файлу головного вузла наведено у таблиці 3.6.

Таблиця 3.6 – Структура запису у лог-файлі master

Поле	Тип	Опис
pkt_id	uint16	Порядковий номер пакета
event	enum	Тип події (ACK / TIMEOUT / PARAM_CHANGE)
sf	uint8	Поточний коефіцієнт розширення
txpwr	int8	Поточна потужність передавача, дБм
rssr	int16	RSSI прийнятого пакета на стороні master, дБм
snr	int8	SNR прийнятого пакета на стороні master, дБ
slave_rssi	int16	RSSI на стороні slave, дБм
slave_snr	int8	SNR на стороні slave, дБ
q	float	Поточне значення функції якості $Q(t)$
snr_var	float	Дисперсія SNR за ковзним вікном
pdr_w	float	Частка успішних доставок за ковзним вікном (PDR)
algo	enum	Режим алгоритму (FIXED / ADR / Q-ADR)

Поля *q*, *snr_var* і *pdr_w* заповнюються тільки в режимі Q-ADR. У режимах Fixed і ADR відповідні внутрішні змінні модуля *algo.c* не обчислюються, тож ці поля містять нульові значення. Поле *algo* з режимом алгоритму є однаковим для всіх рядків у межах одного прогону і дозволяє розрізняти прошивки під час пакетної обробки логів. Така уніфікована структура дозволяє читати лог-файли всіх трьох алгоритмів одним і тим же парсером.

Окремо логуються події зміни параметрів (PARAM_CHANGE), що дозволяє точно визначити моменти ескалації і деескалації у часі. У полях *sf* і *txprwr* такого запису зберігаються старі і нові значення параметрів. Решта числових полів запису не заповнюється, оскільки на момент зміни параметрів актуальні виміри радіоканалу ще не отримано. Це робить можливим ретроспективний аналіз поведінки алгоритму без необхідності повторно запускати експеримент.

Аналогічний лог-файл ведеться і на стороні *interferer*. Він фіксує події початку і завершення імпульсів завад, що дозволяє точно зіставити моменти активності завад з втратами пакетів у лог-файлі *master*.

3.7.3 Автоматизація проведення експериментів

Для забезпечення повторюваності експериментальних умов і скорочення часу проведення кожного прогону розроблено скрипт автоматизації на мові Python. Скрипт підключається до плат *master* і *interferer* через інтерфейси USB-UART, які відображаються в системі як окремі віртуальні COM-порти. Кожній платі присвоюється конкретний COM-порт у конфігураційному файлі, що жорстко прив'язує роль вузла до конкретної фізичної плати. Підлеглий вузол *slave* в експериментальній установці розташовано відносно далеко від інших плат, тому до автоматизованого керування він не підключається.

Конфігурація експерименту визначається у текстовому файлі формату JSON, де задаються тривалість трьох фаз прогону (базова, з активною завадою і фаза відновлення), імена COM-портів для кожної плати та тайм-аут верифікації стартової конфігурації прошивок. Сценарії експериментів, тобто вибір алгоритму адаптації *master* і режиму *interferer*, описуються в окремому JSON-

файлі. Запуск прогону зводиться до однієї команди в терміналі з вказанням імені сценарію, після чого скрипт зчитує конфігурацію, верифікує готовність плат після чого починає одночасне читання логів з UART-портів.

Запис логів виконується двома незалежними потоками, по одному на кожен порт. Кожен потік приймає рядки CSV і пише у відповідний вихідний файл (*master.log* та *interferer.log*) у директорії результатів прогону. Така ізоляція потоків і файлів забезпечує цілісність даних навіть за коротких затримок у читанні з одного з портів. По завершенні останньої фази прогону скрипт надсилає команду зупинки, закриває файли логів і звітує про статистику прогону.

3.8 Висновки до розділу

У цьому розділі було описано програмно-апаратну реалізацію методу Q-ADR на платформі STM32WL55 та інфраструктуру експериментальної установки, необхідну для його перевірки.

Модуль *algo.c* реалізує всі три досліджувані алгоритми за спільним публічним інтерфейсом. Це забезпечує методологічну коректність порівняння: будь-яка різниця в експериментальних результатах між трьома прошивками пояснюється виключно різницею в алгоритмі адаптації, а не в протоколі чи інфраструктурі. Реалізація Q-ADR включає кільцевий буфер на 10 пакетів, обчислення функції якості $Q(t)$ у два проходи по буферу з імпутацією тайм-аутів найгіршим значенням і дерево рішень з трьома пороговими порівняннями. Стандартний ADR реалізовано за специфікацією Semtech з кільцевим буфером на 20 пакетів і формулою NStep. Режим з фіксованими параметрами не приймає рішень і слугує контрольною конфігурацією.

Реалізовано протокол обміну *master-slave* з 4-секундним циклом та окремими тайм-аутами ACK для кожного значення SF. Передавання SET_PARAMS захищено серією повторних спроб з випадковим часовим зсувом і тимчасовим форсуванням потужності передавача до апаратного максимуму.

Інфраструктурну частину утворюють три компоненти. Генератор завад на третьому вузлі реалізує два сценарії постановки завад – LORA_RANDOM з ідентичною модуляцією до робочої пари та FSK_CONTINUOUS з широкосмуговою FSK-модуляцією. Логування подій у форматі CSV охоплює всі протокольні дії та рішення алгоритму, що забезпечує повну відновлюваність картини експерименту. Програмний скрипт автоматизації керує запуском плат, верифікацією стартової конфігурації і паралельним записом логів через USB-UART.

Наукова новизна технічних рішень виявляється на двох рівнях. На рівні оцінювання стану каналу застосовано багатокритеріальну функцію $Q(t)$, яка інтегрує чотири метрики (RSSI, SNR, PDR та дисперсію SNR) у єдиний скалярний показник, що обчислюється безпосередньо на стороні кінцевого вузла. На рівні логіки прийняття рішень інтегровано три механізми захисту, непередбачені у стандартному ADR:

- одночасне підвищення SF та потужності передавача в критичному стані каналу для швидкого виходу із зони низької якості зв'язку;
- SNR-запобіжник, що блокує деескалацію параметрів у каналах із хибною стабільністю, де інтегральний показник Q є високим, але рівень сигнал/шум залишається граничним;
- затримка зниження коефіцієнта розширення спектра на 30 пакетів після критичної ескалації для запобігання повторному потраплянню в деградований стан.

Описана програмно-апаратна реалізація та інфраструктура експерименту утворюють основу для кількісного порівняння методу Q-ADR з реалізованими еталонами у наступному розділі.

РОЗДІЛ 4 АНАЛІЗ РЕЗУЛЬТАТІВ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ МЕТОДУ Q-ADR

4.1 Методика експерименту

У цьому розділі наведено результати експериментальної перевірки методу Q-ADR. Перевірка ґрунтується на порівнянні трьох реалізованих алгоритмів адаптації, а саме Fixed, ADR і Q-ADR, у двох сценаріях зовнішніх радіозавад. Метою експерименту є кількісне оцінювання ефективності кожного алгоритму за надійністю доставки і енергоспоживанням передавача в умовах активних радіозавад.

Кожен експериментальний прогон має трифазну часову структуру тривалістю 640 секунд: фаза нормальної роботи (160 с), фаза з активною завадою (300 с) і фаза відновлення (180 с). Під час першої фази *master* і *slave* обмінюються пакетами без зовнішніх завад. Це дозволяє зафіксувати стартові характеристики каналу і дати алгоритмам можливість стабілізуватись. У фазі завад вмикається *interferer* з заданим режимом постановки завад, який працює до її завершення. У фазі відновлення *interferer* вимикається, і алгоритм повертає параметри передачі до енергоефективніших значень за умов знову чистого каналу. Така трифазна структура дозволяє оцінити три ключові аспекти поведінки алгоритму: стартову поведінку у чистому каналі, реакцію на деградацію умов і швидкість відновлення після зникнення завад.

Для забезпечення статистичної достовірності результатів експериментальне дослідження виконано у три незалежні прогони для кожного з шести сценаріїв. Графіки 4.1–4.6 наведено для одного представницького прогону, обраного за критерієм найближчого до середніх показників. Числові значення метрик у тексті та підсумкових таблицях обчислено як середнє арифметичне за трьома прогонами. Міжпрогонна варіація не перевищує 2–5% за всіма метриками PDR, EE та сумарним енергоспоживанням, що підтверджує стабільність результатів і достатність вибраної кількості прогонів для виявлення характеристик алгоритмів адаптації.

Для оцінювання ефективності алгоритмів обрано наступні метрики:

- Packet Delivery Ratio (PDR) – частка пакетів *master*, на які отримано підтвердження ACK від *slave*; обчислюється окремо для кожної з трьох фаз і для прогону в цілому;
- середній коефіцієнт розширення і середня потужність передавача – обчислюються за фазами і відображають вибір параметрів алгоритмом;
- Energy Efficiency (EE) – енергоефективність як відношення кількості доставлених пакетів до сумарної енергії передачі, виражена в пакетах на мДж;

Енергія передачі окремого пакета обчислюється за спрощеною моделлю, що враховує тільки активну передачу без споживання приймача і процесора, за формулою (4.1):

$$E_{TX} = \frac{T_{ToA} * P_{TX}}{1000}, \quad (4.1)$$

де E_{TX} – енергія передачі одного пакета, мДж; T_{ToA} – тривалість пакета в ефірі для поточного коефіцієнта розширення SF, мс (за таблицею 4.1); P_{TX} – потужність передавача в лінійних одиницях, мВт; отримана з налаштованого значення у дБм стандартним перетворенням.

Таблиця 4.1 – Тривалість пакета в ефірі для робочого діапазону SF

SF	T_{ToA} , мс
7	72
8	134
9	247
10	453

Підсумкова метрика Energy Efficiency обчислюється як відношення кількості пакетів, на які отримано підтвердження ACK, до сумарної енергії всіх передач за прогон, за формулою (4.2):

$$EE = \frac{N_{ACK}}{\sum_{i=1}^{N_{TX}} E_{TX,i}}, \quad (4.2)$$

де EE – енергоефективність алгоритму, пакетів/мДж; N_{ACK} – кількість пакетів з отриманим АСК (доставлених); N_{TX} – загальна кількість передач за прогон, що включає як успішні (АСК), так і завершені тайм-аутом.

Більше значення EE відповідає вищій ефективності алгоритму у термінах кількості доставлених пакетів на одиницю спожитої енергії, що безпосередньо переноситься на час життя пристрою від обмеженого джерела живлення. Однак ця метрика розглядається разом з PDR, оскільки алгоритм з низькою потужністю передачі може отримувати формально високе значення EE через малий знаменник, не забезпечуючи достатнього рівня надійності доставки. Тому остаточна оцінка ефективності алгоритму будується на комбінації EE з PDR, що дозволяє виявляти випадки пасивної поведінки алгоритму, які не вирішують поставлену задачу адаптації.

4.2 Експериментальні сценарії

Експериментальне порівняння виконано у шести сценаріях, утворених перетином трьох досліджуваних алгоритмів адаптації (Fixed, ADR, Q-ADR) і двох режимів роботи генератора завад (LORA_RANDOM і FSK_CONTINUOUS).

Конфігурація радіоінтерфейсу *master* є однаковою для всіх сценаріїв і відповідає таблиці 3.1: частота 868 МГц, смуга 125 кГц, швидкість кодування 4/5. Початкові параметри передавача ($SF = 7$, $TxPower = 10$ дБм) також однакові для всіх алгоритмів. Це гарантує однаковий стан системи перед початком кожного прогону.

Два режими постановки завад обрано як такі, що демонструють принципово різну природу впливу на робочий канал. У режимі LORA_RANDOM завади мають імпульсну природу з нерегулярним часовим профілем і впливають насамперед на нижні значення SF , де *time-on-air master* є коротким. У режимі FSK_CONTINUOUS завади є майже неперервними і впливають на всі значення

SF з різним ступенем перекриття. Ці два режими забезпечують різні умови для тестування адаптивних алгоритмів. Перший становить стрес-тест для нижніх значень SF, де адаптація вимушена піднімати SF або потужність, щоб уникнути колізій з імпульсами завад. Другий становить стрес-тест для всіх значень параметрів, де ефективність алгоритму визначається його здатністю знайти оптимальну точку у двовимірному просторі параметрів (SF, TxPower).

4.3 Дослідження адаптації за імпульсних LoRa-завад

У сценарії LORA_RANDOM генератор завад працює в режимі імпульсних LoRa-сигналів з випадковими часовими інтервалами. Цей режим створює стрес-тест для нижніх значень SF, де *time-on-air master* є коротким, а ймовірність зіткнення з імпульсом завади помірна. Динаміку надійності доставки трьох алгоритмів у часі наведено на рис. 4.1.

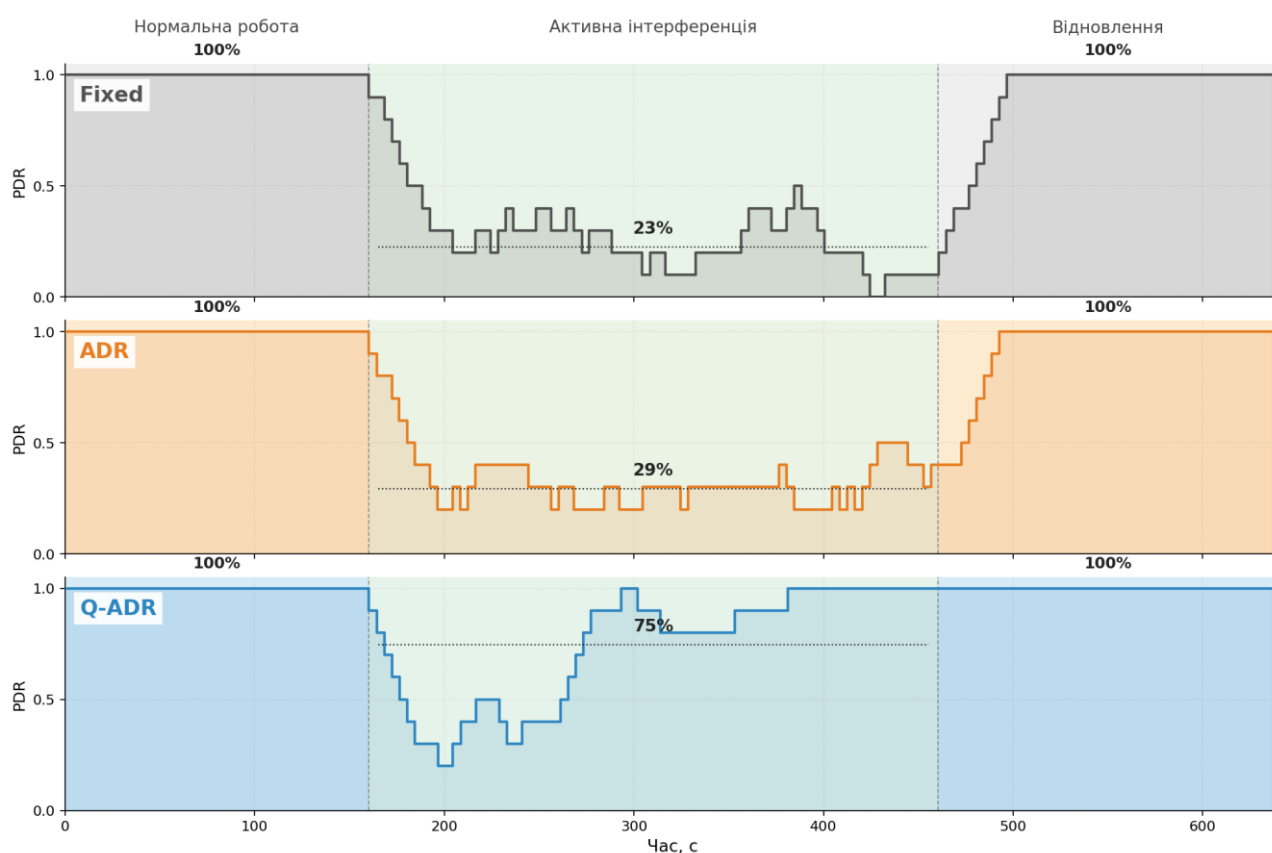


Рисунок 4.1 – Динаміка надійності доставки трьох алгоритмів адаптації у сценарії LORA_RANDOM

Під час фази нормальної роботи (0–160 с) усі три алгоритми забезпечують $PDR = 100\%$, що підтверджує справність установки та коректну роботу пари *master–slave* за відсутності зовнішнього впливу. Після ввімкнення інтерферера на $t = 160$ с PDR алгоритмів Fixed і ADR протягом наступних 40 секунд плавно сповзає до 20–30% і утримується на цьому рівні до завершення фази активної інтерференції на $t = 460$ с. PDR алгоритму Q-ADR у перші секунди дії завад також знижується до значень близько 20%, проте, на відміну від ADR, через приблизно 30 секунд після появи завад починає поступово зростати завдяки активній адаптації параметрів передавача і досягає значення $PDR = 100\%$ ще під час активної інтерференції, приблизно на $t = 400$ с. У фазі відновлення (460–640 с) всі три алгоритми повертають PDR до 100% за один цикл обміну, оскільки на момент зняття завад параметри Q-ADR залишаються надлишковими для чистого каналу, а Fixed і ADR взагалі не змінювалися. Видиме зростання кривої на рис. 4.1 у цій фазі триває близько 40 секунд і відповідає тривалості ковзного вікна спостереження PDR .

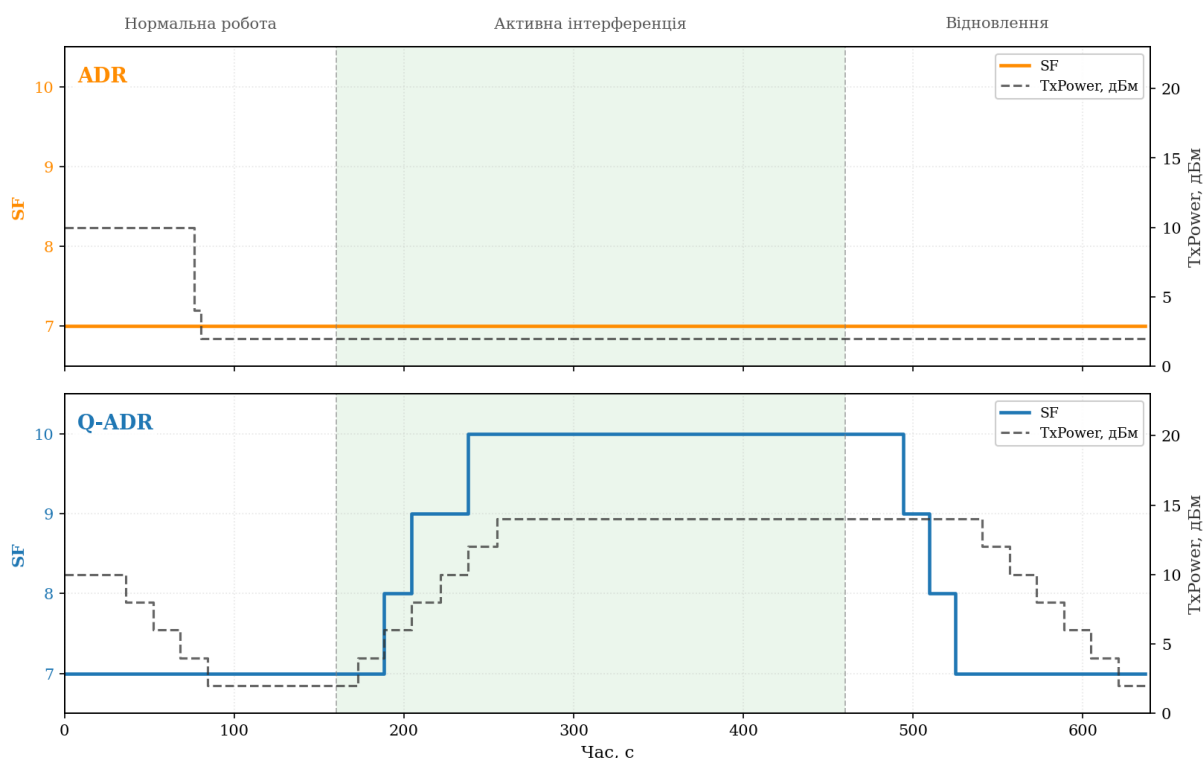


Рисунок 4.2 – Динаміка параметрів передавача SF і TxPower для алгоритмів ADR і Q-ADR у сценарії LORA_RANDOM

Поведінку адаптивних алгоритмів щодо вибору параметрів передавача наведено на рис. 4.2. Алгоритм Fixed на цьому рисунку не показаний, оскільки його параметри ($SF = 7$, $TxPower = 10$ дБм) залишаються незмінними протягом усього експерименту.

Аналіз рис. 4.2 розкриває принципово різну поведінку двох методів. Алгоритм ADR у початковій фазі виконав послідовну деескалацію потужності з $TxPower = 10$ дБм до $TxPower = 2$ дБм, оскільки канал був чистим і значення SNR утримувалось на рівні близько 13 дБ. Після ввімкнення інтерференції алгоритм не зреагував жодним чином: SF залишився рівним 7, а $TxPower$ – рівним 2 дБм, що становить мінімально дозволене значення для цього радіомодуля. За час фази активної інтерференції алгоритм не виконав жодної адаптаційної події. Така поведінка є наслідком архітектурної особливості ADR Semtech, який оперує тільки за SNR і запускається лише після наповнення вікна з 20 пакетів стабільно низьким значенням SNR. Імпульсна природа LORA_RANDOM не призводить до стабільного зниження SNR, оскільки декодовані пакети зберігають високе значення SNR, а недекодовані – не реєструються у вікні ADR як такі, що несуть інформацію про якість каналу. Окрема гілка стандартного ADR передбачає примусове підвищення параметрів після 8 послідовних тайм-аутів, але в цьому сценарії така умова не спрацьовує: 22 успішні доставки серед 75 спроб щоразу скидають лічильник до досягнення порогу.

Алгоритм Q-ADR продемонстрував принципово іншу поведінку. Через 13 секунд після появи завад алгоритм виконав першу адаптацію – підвищив $TxPower$ з 2 до 4 дБм. На 28-й секунді інтерференції відбулась друга критична адаптація з одночасним підвищенням SF з 7 до 8 та $TxPower$ з 4 дБм до 6 дБм. Протягом наступних 70 секунд алгоритм послідовно піднімав параметри до конфігурації $SF = 10$, $TxPower = 14$ дБм, яка забезпечила стабільне декодування пакетів попри активність інтерферера. Усього за час фази активної інтерференції запропонований алгоритм виконав 6 змін параметрів передавання, тоді як ADR – жодної. У фазі відновлення Q-ADR виконав 9 деескалаційних кроків і повернувся до конфігурації $SF = 7$, $TxPower = 2$ дБм.

Підсумкові значення надійності доставки та енергоефективності у фазі активної інтерференції наведено на рис. 4.3.

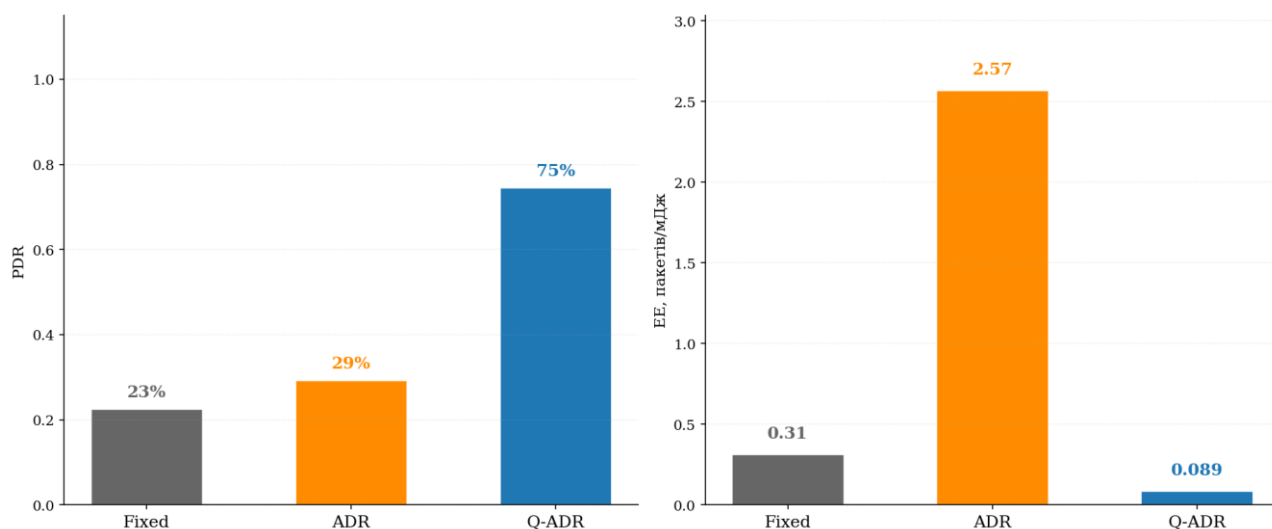


Рисунок 4.3 – Підсумкові показники PDR та енергоефективності EE під час фази активної інтерференції у сценарії LORA_RANDOM

За показником PDR алгоритм Q-ADR з результатом 74.7% значно випереджає Fixed (22.7%) і ADR (29.3%), забезпечуючи майже у 3 рази вищу надійність доставки порівняно з ADR. За показником EE формальне лідерство займає алгоритм ADR зі значенням 2.57 пакетів/мДж, що у 8.3 рази перевищує Fixed (0.31) і у 28.9 рази – Q-ADR (0.089). Ця формальна перевага ADR за енергоефективністю пояснюється тим, що алгоритм передавав на мінімальній потужності 2 дБм протягом усіх 300 секунд активної інтерференції без жодної спроби адаптації.

Однак низький рівень енергоспоживання ADR поєднується з втратою 71% переданих пакетів, що неприйнятно для більшості IoT-систем з вимогою надійності доставки понад 70%. Алгоритм Q-ADR обирає протилежну стратегію: ціною підвищеного енергоспоживання забезпечує надійність 74.7%. Саме Q-ADR є придатним вибором для типових сценаріїв IoT, оскільки задача алгоритму адаптації полягає не в мінімізації енергоспоживання за будь-яку ціну, а в

підтриманні необхідного рівня надійності доставки за обмежених енергетичних ресурсів.

Цей результат виявляє ключове обмеження одновимірних метрик енергоефективності. Алгоритм, який лише мінімізує потужність без адаптації до каналу, формально отримує найвищий ЕЕ, але не забезпечує надійності зв'язку. Тому остаточна оцінка у роботі ґрунтується на комбінації PDR та ЕЕ з пріоритетом PDR як індикатора виконання основного завдання системи.

4.4 Дослідження адаптації за неперервних FSK-завад

У сценарії FSK_CONTINUOUS генератор завад працює в режимі неперервної FSK-передачі, спектр якої повністю перекриває робочу смугу 125 кГц. На відміну від імпульсного режиму LORA_RANDOM, тут завада діє практично безперервно і впливає на всі значення SF з різним ступенем перекриття. Динаміку надійності доставки трьох алгоритмів у часі наведено на рис. 4.4.

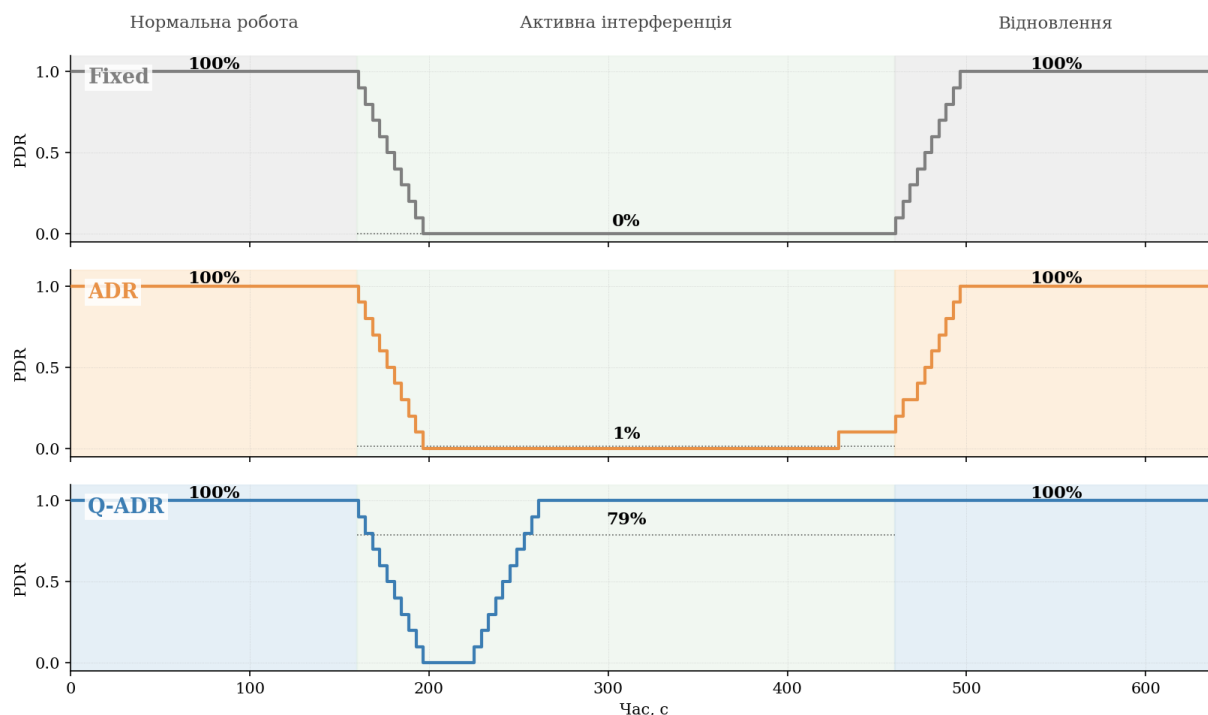


Рисунок 4.4 – Динаміка надійності доставки трьох алгоритмів адаптації у сценарії FSK_CONTINUOUS

Під час фази нормальної роботи (0–160 с) усі три алгоритми забезпечують $PDR = 100\%$, що підтверджує справність установки. Після ввімкнення інтерферера на $t = 160$ с PDR алгоритмів Fixed і ADR протягом наступних 36 секунд плавно сповзає до нуля і утримується на цьому рівні до завершення фази активної інтерференції на $t = 460$ с. У Fixed жоден з 75 пакетів не був успішно доставлений. Алгоритм ADR доставив лише один пакет із 75 ($PDR = 1.3\%$), причому це сталося наприкінці фази, після того як алгоритм підняв TxPower до 18 дБм, що дало короточасний підйом кривої PDR ближче до $t = 430$ с. PDR алгоритму Q-ADR у перші 37 секунд після появи завад також сповзає до нуля, проте, на відміну від попередніх двох, через приблизно 65 секунд після появи завад починає сходинково зростати завдяки активній адаптації параметрів передавача і досягає значення $PDR = 100\%$ приблизно на $t = 260$ с. Стабільний рівень $PDR = 100\%$ алгоритм Q-ADR утримує близько 200 секунд із 300 до завершення фази активної інтерференції. У фазі відновлення (460–640 с) усі три алгоритми повертають PDR до 100% за один цикл обміну.

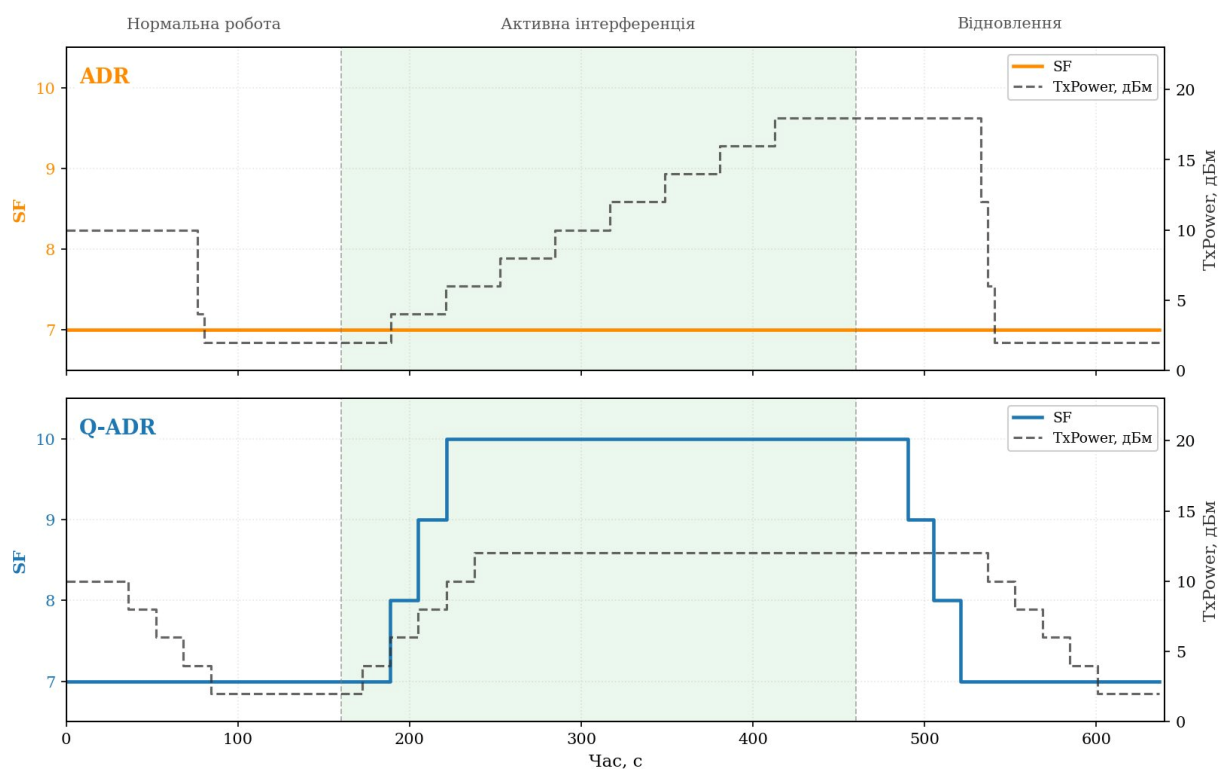


Рисунок 4.5 – Динаміка параметрів передавача SF і TxPower для алгоритмів ADR і Q-ADR у сценарії FSK_CONTINUOUS

Поведінку адаптивних алгоритмів щодо вибору параметрів передавача наведено на рис. 4.5. Алгоритм Fixed на цьому рисунку не показаний, оскільки його параметри ($SF = 7$, $TxPower = 10$ дБм) залишаються незмінними протягом усього експерименту.

На відміну від сценарію LORA_RANDOM, у режимі неперервних FSK-завад алгоритм ADR активно реагує на погіршення каналу. Неперервна перешкода призводить до серії послідовних тайм-аутів, що активує механізм форсованої ескалації після 8 послідовних втрат пакетів. Як видно з рис. 4.5, протягом фази активної інтерференції ADR виконав 8 послідовних адаптаційних подій, поступово піднявши $TxPower$ від 2 до 18 дБм. Однак SF залишився рівним 7 протягом усієї фази. Це є наслідком ієрархії дій у механізмі форсованої ескалації: алгоритм спочатку послідовно підвищує $TxPower$ на 2 дБм за кожні 8 тайм-аутів і лише після досягнення апаратного максимуму потужності починає підвищувати SF . За 300-секундну фазу активної інтерференції $TxPower$ встиг піднятися лише до 18 дБм зі стартового значення 2 дБм і не досяг апаратної межі радіомодуля, через що умови для запуску SF -ескалації так і не виникли. У результаті ADR обмежився підвищенням потужності, яке для безперервних завад є недостатнім: навіть на $TxPower = 18$ дБм при $SF = 7$ запас за SNR залишається недостатнім для надійного декодування. Як наслідок, із 75 спроб передавання успішно доставленим виявився лише один пакет, причому під самий кінець фази, після того як потужність зросла до 18 дБм.

Алгоритм Q-ADR продемонстрував принципово іншу стратегію адаптації. Через 13 секунд після появи завад алгоритм виконав першу адаптацію – підвищив $TxPower$ з 2 до 4 дБм. Однак уже на 29-й секунді інтерференції відбулась критична адаптація з одночасним підвищенням SF з 7 до 8 та $TxPower$ з 4 до 6 дБм. Протягом наступних 50 секунд алгоритм послідовно піднімав параметри до конфігурації $SF = 10$, $TxPower = 12$ дБм, яка забезпечила стабільне декодування пакетів попри активність *interferer* вузла. Усього за час фази активної інтерференції запропонований алгоритм виконав 5 змін параметрів передавання. Це менше, ніж 8 у ADR, проте саме завдяки залученню SF як

другого виміру адаптації Q-ADR досяг стабільної доставки на 65-й секунді після початку завад. У фазі відновлення Q-ADR виконав 8 деескалаційних кроків і повернувся до конфігурації $SF = 7$, $TxPower = 2$ дБм.

Підсумкові значення надійності доставки та енергоефективності у фазі активної інтерференції наведено на рис. 4.6.

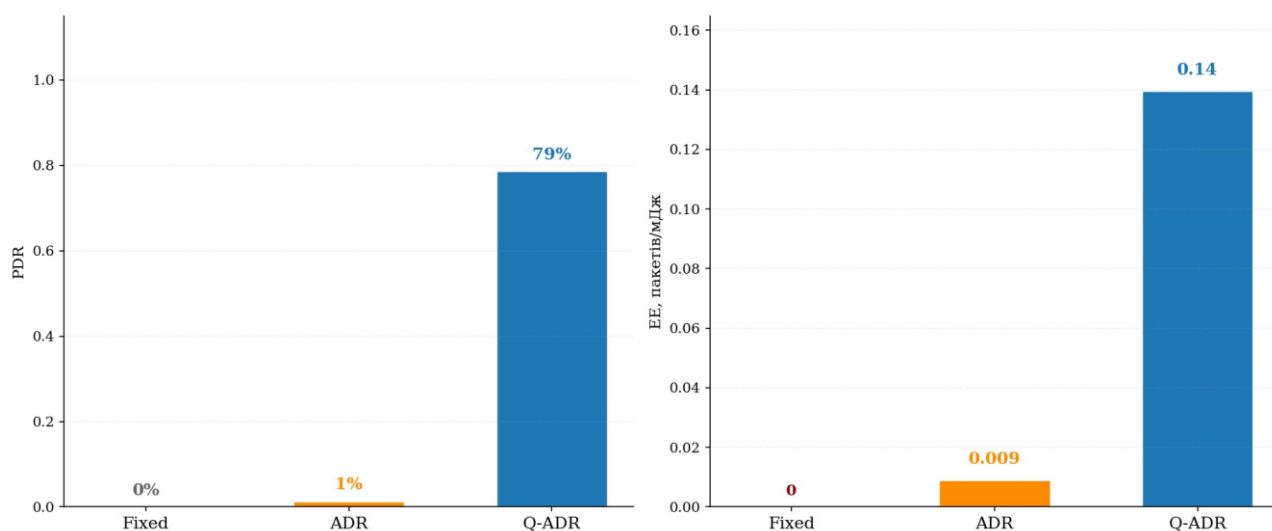


Рисунок 4.6 – Підсумкові показники PDR та енергоефективності EE під час фази активної інтерференції у сценарії FSK_CONTINUOUS

За показником PDR алгоритм Q-ADR з результатом 78.7% значно випереджає Fixed (0%) і ADR (1.3%). На відміну від сценарію LORA_RANDOM, перевага Q-ADR підтверджується і за показником EE: 0.14 пакетів/мДж проти 0.009 у ADR і 0 у Fixed. У режимі безперервних FSK-завад Q-ADR є кращим за обома критеріями одночасно.

Така поведінка має чітке фізичне пояснення. ADR піднімає $TxPower$ до 18 дБм, однак цього виявляється недостатньо для забезпечення необхідного запасу SNR при $SF = 7$ в умовах безперервних FSK-завад: декодування пакетів практично не відбувається, і витрачена на передавання енергія не перетворюється у доставлені пакети. Q-ADR натомість залучає підвищення SF як основний інструмент компенсації. Кожне збільшення SF на одиницю розширює запас за чутливістю приймача приблизно на 2.5 дБ. У сукупності з

помірним підвищенням TxPower до 12 дБм це забезпечує стабільне декодування пакетів у конфігурації $SF = 10$. У результаті майже вся витрачена Q-ADR енергія конвертується у доставлені пакети, що дає одночасно високий PDR і прийнятні енергетичні витрати на доставлений пакет.

Результати цього сценарію підтверджують ключову перевагу двовимірної адаптації параметрів передавача (SF і TxPower) над одновимірною (тільки TxPower) у сценаріях з безперервними завадами. Нездатність ADR оперативно піднімати SF робить його неефективним за таких умов. Коли завади знижують SNR нижче порогу декодування для $SF = 7$, доставка пакетів неможлива без переходу на вищий SF .

4.5 Висновки до розділу

У цьому розділі проведено експериментальне дослідження методу Q-ADR у порівнянні з двома алгоритмами-еталонами – режимом з фіксованими параметрами (Fixed) та стандартним ADR за специфікацією Semtech. Експерименти виконано на платформі STM32WL55 у двох якісно різних типах радіозавад: імпульсних LoRa-сигналах (LORA_RANDOM) і безперервних FSK-перешкодах (FSK_CONTINUOUS). Підсумкове порівняння трьох алгоритмів за обома сценаріями наведено на рис. 4.7.

За показником надійності доставки метод Q-ADR демонструє стабільну перевагу над обома еталонами в обох сценаріях. У сценарії LORA_RANDOM Q-ADR забезпечує $PDR = 74.7\%$, тоді як Fixed і ADR залишаються на рівні 22.7% і 29.3% відповідно. У сценарії FSK_CONTINUOUS перевага ще більш виражена: 78.7% проти 0% у Fixed і 1.3% у ADR. Це підтверджує, що запропонований метод здатний підтримувати надійний зв'язок як за коротких імпульсних інтерференцій, так і за постійних завад широкого спектра.

Варто зазначити, що метод Q-ADR досягає стабільної доставки швидше у сценарії з безперервною FSK-завадою, ніж у сценарії з імпульсною LoRa-завадою. Це зумовлено тим, що стабільно низьке значення $Q(t)$ за безперервної

завади дозволяє алгоритму швидко досягти стаціонарного стану, тоді як за імпульсної завади $Q(t)$ осцилює між високими і низькими значеннями синхронно з паузами між імпульсами генератора, що сповільнює збіжність алгоритму.

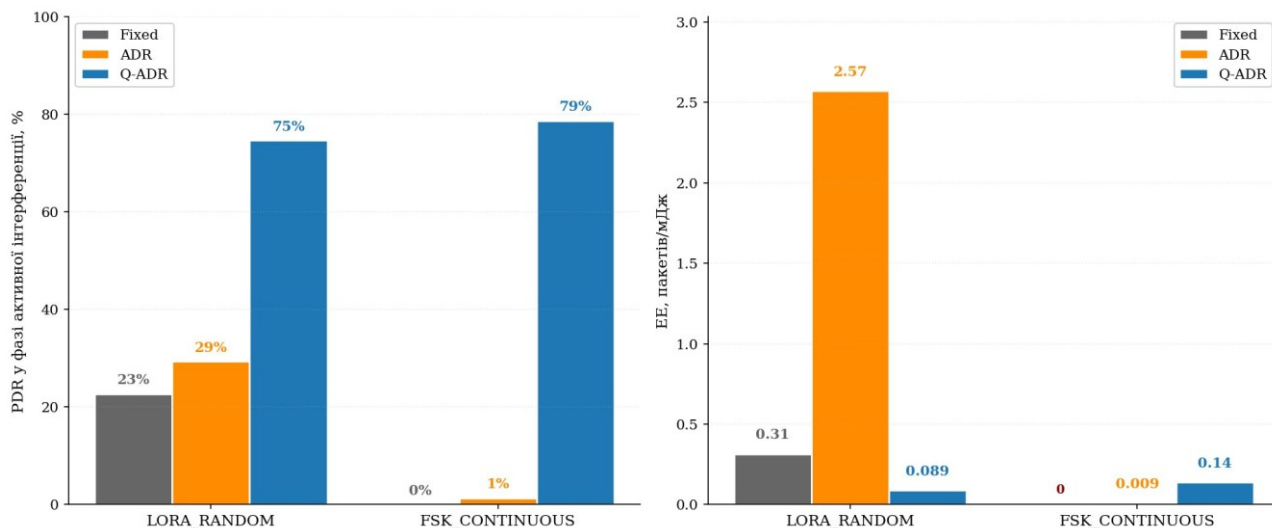


Рисунок 4.7 – Підсумкові показники PDR та енергоефективності EE трьох алгоритмів адаптації у сценаріях LORA_RANDOM і FSK_CONTINUOUS

За показником енергоефективності картина є складнішою. У сценарії LORA_RANDOM формальне лідерство за EE належить ADR (2.57 пакетів/мДж) проти 0.31 у Fixed і 0.089 у Q-ADR. Однак ця перевага є наслідком пасивної поведінки ADR, який не виконав жодної адаптаційної події протягом фази інтерференції і працював на мінімальній потужності 2 дБм. У сценарії FSK_CONTINUOUS, де ADR змушений активно нарощувати потужність до 18 дБм, його EE падає до 0.009 пакетів/мДж, тоді як Q-ADR досягає 0.14 пакетів/мДж. Це показує, що одновимірний показник енергоефективності може формально віддавати перевагу алгоритму, який економить енергію за рахунок невиконання основної функції підтримання надійного зв'язку. Тому остаточна оцінка ефективності у роботі ґрунтується на спільному розгляді PDR та EE з пріоритетом PDR як індикатора виконання основного завдання комунікаційної системи.

Отримані результати підтверджують переваги двовимірної адаптації параметрів передавача (SF і TxPower) на основі багатокритеріальної функції якості $Q(t)$ над одновимірною адаптацією (тільки TxPower), реалізованою у стандартному ADR. У сценарії LORA_RANDOM Q-ADR виконав 6 адаптаційних подій із залученням SF і досяг конфігурації SF = 10, TxPower = 14 дБм, що дозволило відновити зв'язок ще під час фази інтерференції. У сценарії FSK_CONTINUOUS аналогічна двовимірна стратегія дала стабільне декодування за рахунок підвищення SF з 7 до 10 при помірному рівні TxPower = 12 дБм. На противагу, ADR не зміг оперативно піднятися до конфігурації, придатної для декодування пакетів за безперервних FSK-завад.

Кількісні результати експерименту підтверджують гіпотезу про те, що ефективна адаптація параметрів передавання в умовах змінного радіосередовища потребує одночасного врахування декількох метрик якості каналу і двовимірної зміни параметрів передавача. Метод Q-ADR забезпечує цю функціональність на енергообмеженій IoT-платформі STM32WL55 без додаткових апаратних або обчислювальних витрат, що робить його придатним для практичного застосування в LPWAN-системах різного призначення.

ВИСНОВКИ

У магістерській дисертації розв'язано актуальне науково-практичне завдання підвищення надійності передавання даних з урахуванням енергоефективності функціонування low-power IoT систем шляхом розроблення методу контекстно-залежної адаптації параметрів радіопротоколу на основі багатокритеріальної оцінки якості зв'язку. Мета роботи, що полягала у розробленні та експериментальній перевірці запропонованого методу, досягнута у повному обсязі.

У першому розділі дисертації проведено аналіз сучасних підходів до адаптивного керування параметрами передавання у LoRa-мережах. Розглянуто фізичний рівень технології LoRa та архітектуру протоколу LoRaWAN, детально описано стандартний механізм Adaptive Data Rate і три його ключові обмеження: інерційність (потреба у накопиченні вікна з 20 пакетів), однокритеріальність (адаптація лише за SNR без врахування PDR, затримок підтверджень та навантаження мережі) та централізованість (вимога повного стека LoRaWAN з мережним сервером). Виконано систематичний огляд альтернативних методів за чотирма напрямками: швидкою реакцією на динамічні радіоумови, зменшенням колізій у щільних мережах, розширеною оцінкою стану каналу та локальною адаптацією на стороні вузла. Встановлено, що жоден з оглянутих методів не поєднує одночасно локального виконання на стороні кінцевого вузла, оцінки каналу за кількома метриками та узгодженої адаптації основних параметрів передавання SF і TxPower.

У другому розділі сформульовано задачу адаптивного керування параметрами радіопротоколу і розроблено метод Q-ADR (Quality-based Adaptive Data Rate). Побудовано скалярну функцію якості каналу $Q(t) \in [0, 1]$ як зважену суму чотирьох нормалізованих метрик: рівня сигналу RSSI ($\alpha = 0.10$), відношення сигнал/шум SNR ($\beta = 0.40$), частки успішних доставок PDR ($\gamma = 0.30$) і дисперсії SNR ($\delta = 0.20$). Розподіл ваг відображає ієрархію інформативності метрик з пріоритетом SNR як безпосереднього індикатора декодованості та PDR як підсумкового показника надійності. Сформульовано логіку прийняття рішень

на основі трьох порогів ($Q_{crit} = 0.40$, $Q_{low} = 0.60$, $Q_{high} = 0.90$) з механізмами захисту від хибних спрацьовувань: SNR-запобіжником, що блокує деескалацію у каналах з хибною стабільністю; затримкою зниження SF на 30 пакетів після критичної ескалації; блокуванням на 3 пакети після кожної зміни параметрів.

У третьому розділі розроблено програмно-апаратну реалізацію методу Q-ADR на платформі STM32WL55. Реалізацію виконано у вигляді модуля адаптації зі спільним публічним інтерфейсом для трьох алгоритмів: запропонованого Q-ADR, стандартного ADR за специфікацією Semtech та режиму з фіксованими параметрами. Такий підхід забезпечує методологічну коректність порівняння: будь-яка різниця у результатах між трьома прошивками пояснюється виключно відмінністю в алгоритмі адаптації, а не у протоколі чи інфраструктурі. Створено експериментальну установку з трьох вузлів NUCLEO-WL55JC1 (master, slave, interferer) з керованим генератором радіозавад у двох режимах: імпульсних LoRa-сигналів (LORA_RANDOM) та неперервних FSK-перешкод (FSK_CONTINUOUS). Розроблено програмну інфраструктуру, що забезпечує повне логування подій радіопротоколу у форматі CSV та автоматизоване проведення експериментів.

У четвертому розділі проведено експериментальне дослідження ефективності методу Q-ADR у порівнянні з двома алгоритмами-еталонами в двох якісно різних типах радіозавад. Кожен сценарій включав три фази: нормальну роботу (160 с), активну інтерференцію (300 с) і відновлення (180 с), що забезпечило відтворюваність експерименту та можливість окремої оцінки динаміки відновлення зв'язку.

За результатами проведених експериментів встановлено, що метод Q-ADR забезпечує суттєву перевагу за надійністю доставки в обох сценаріях. У сценарії з імпульсними LoRa-завадами Q-ADR досяг $PDR = 74.7\%$ проти 29.3% у стандартного ADR та 22.7% у режиму з фіксованими параметрами, що відповідає підвищенню надійності у 2.55 рази порівняно з ADR. У сценарії з неперервними FSK-завадами перевага виявилася ще більш виражена: Q-ADR забезпечив $PDR = 78.7\%$, тоді як ADR та режим з фіксованими параметрами

залишилися практично непрацездатними з результатами 1.3 % і 0 % відповідно. За показником енергоефективності у сценарії з неперервними завадами Q-ADR досяг 0.14 пакетів/мДж проти 0.009 у стандартного ADR, тобто перевершив еталон одночасно за обома критеріями оцінки.

Виявлено принципову перевагу двовимірної адаптації параметрів передавача (SF і TxPower) над одновимірною (тільки TxPower), реалізованою у стандартному ADR. У сценарії з імпульсними завадами Q-ADR виконав 6 адаптаційних подій із залученням SF і досяг конфігурації SF = 10, TxPower = 14 дБм, що дозволило відновити надійний зв'язок під час фази інтерференції. У сценарії з неперервними завадами аналогічна двовимірна стратегія забезпечила стабільне декодування за рахунок підвищення SF з 7 до 10 при помірному рівні TxPower = 12 дБм. На противагу, стандартний ADR не зміг оперативно піднятися до конфігурації, придатної для декодування пакетів за неперервних завад. Достовірність отриманих результатів забезпечується повним логуванням усіх подій радіопротоколу у форматі CSV, що дозволяє відтворити числові показники безпосередньо з первинних даних експерименту.

У результаті виконання роботи отримано такі наукові результати. Удосконалено метод адаптації параметрів передавання у LoRa-системах. На відміну від стандартного механізму ADR, розроблений алгоритм базується на багатокритеріальній функції оцінки якості зв'язку $Q(t)$, що враховує одночасно рівень сигналу (RSSI, SNR), надійність доставки (PDR) та стабільність якості каналу (дисперсію SNR) з ваговими коефіцієнтами, що дозволяє ефективніше реагувати на зміну стану радіоканалу та навантаження мережі. Дістала подальшого розвитку концепція контекстно-залежної адаптації параметрів радіопротоколу, яка, на відміну від відомих рішень, забезпечує одночасну узгоджену зміну SF і TxPower на основі інтегрованої оцінки стану мережі, що скорочує кількість повторних передач і зменшує енергоспоживання вузлів за умов динамічного радіосередовища.

Практичне значення одержаних результатів полягає у можливості застосування розробленого методу при проєктуванні енергообмежених IoT-

систем на базі технології LoRa. Метод Q-ADR виконується безпосередньо на стороні кінцевого пристрою без необхідності залучення мережного сервера, що робить його придатним для застосування у LoRa peer-to-peer розгортаннях, поширених у промислових IoT-системах з власними протоколами обміну. Програмно-апаратний прототип методу, реалізований на платформі STM32WL55, демонструє підвищення надійності передавання даних та збереження прийнятного рівня енергоспоживання порівняно зі стандартним механізмом ADR за умов динамічного радіосередовища. Запропонований підхід може бути впроваджений у системах віддаленого моніторингу довкілля, промислового IoT та масштабованих сенсорних мережах, де критичним є балансування між надійністю зв'язку та строком автономної роботи пристроїв.

Серед перспективних напрямків подальших досліджень слід відзначити адаптацію методу для масштабованих мереж з великою кількістю вузлів, де додаткового значення набувають ефекти колізій і конкуренції за радіоканал, а також дослідження поведінки Q-ADR у мобільних сценаріях, де стан каналу змінюється внаслідок переміщення вузлів або джерел завад. Окремий перспективний напрямок становить оптимізація вагових коефіцієнтів функції $Q(t)$ методами автоматизованого підбору параметрів на основі історичних даних експлуатації мережі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. IoT Analytics. State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally. URL: <https://iot-analytics.com/number-of-connected-iot-devices-2024/>
2. Augustin A., Yi J., Clausen T., Townsley W. M. A Study of LoRa: Long Range and Low Power Networks for the Internet of Things. *Sensors*. 2016. Vol. 16, No. 9. P. 1466. DOI: 10.3390/s16091466.
3. Finnegan J., Farrell R., Brown S. Analysis and Enhancement of the LoRaWAN Adaptive Data Rate Scheme. *IEEE Internet of Things Journal*. 2020. Vol. 7, No. 8. P. 7171–7180. DOI: 10.1109/JIOT.2020.2982745.
4. LoRaWAN L2 1.0.4 Specification (TS001-1.0.4). LoRa Alliance, 2020. URL: <https://resources.lora-alliance.org/technical-specifications/ts001-1-0-4-lorawan-l2-1-0-4-specification>
5. Радченко, К., & Статечний С. (2026). Сучасні підходи до контекстно-залежної адаптації параметрів LoRaWAN у енергообмежених IoT-системах. КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ: ОСВІТА, НАУКА, ВИРОБНИЦТВО, (62), 311-320. <https://doi.org/10.36910/6775-2524-0560-2026-62-35>
6. Mekki K., Bajic E., Chaxel F., Meyer F. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express*. 2019. Vol. 5, No. 1. P. 1–7. DOI: 10.1016/j.icte.2017.12.005.
7. Semtech Corporation. AN1200.22 LoRa Modulation Basics. Application Note, Revision 2. 2015.
8. Slabicki M., PremSankar G., Di Francesco M. Adaptive configuration of LoRa networks for dense IoT deployments. *У: NOMS 2018 – IEEE/IFIP Network Operations and Management Symposium*. 2018. P. 1–9. DOI: 10.1109/NOMS.2018.8406255.
9. Kufakunesu R., Hancke G. P., Abu-Mahfouz A. M. A survey on adaptive data rate optimization in LoRaWAN: Recent solutions and major challenges. *Sensors*. 2020. Vol. 20, No. 18. P. 5044. DOI: 10.3390/s20185044.

10. Park J., Park K., Bae H., Kim C.-K. EARN: Enhanced ADR with coding rate adaptation in LoRaWAN. *IEEE Internet of Things Journal*. 2020. Vol. 7, No. 12. P. 11873–11883. DOI: 10.1109/JIOT.2020.3005881.
11. Serati R., Teymuri B., Anagnostopoulos N. A., Rasti M. ADR-Lite: A Low-Complexity Adaptive Data Rate Scheme for the LoRa Network. 2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). 2022. P. 296–301. DOI: 10.1109/WiMob55322.2022.9941614.
12. Cuomo F., Garlisi D., Martino A., Martino A. Predicting LoRaWAN Behavior: How Machine Learning Can Help. *Computers*. 2020. Vol. 9, No. 3. P. 60. DOI: 10.3390/computers9030060.
13. Park G., Lee W., Joe I. Network resource optimization with reinforcement learning for low power wide area networks. *EURASIP Journal on Wireless Communications and Networking*. 2020. Vol. 2020. P. 176. DOI: 10.1186/s13638-020-01783-5.
14. Sandoval R. M., Garcia-Sanchez A.-J., Garcia-Haro J. Optimizing and Updating LoRa Communication Parameters: A Machine Learning Approach. *IEEE Transactions on Network and Service Management*. 2019. Vol. 16, No. 3. P. 884–895. DOI: 10.1109/TNSM.2019.2927759.
15. Beltramelli L., Mahmood A., Österberg P., Gidlund M. LoRa beyond ALOHA: An Investigation of Alternative Random Access Protocols. *IEEE Transactions on Industrial Informatics*. 2021. Vol. 17, No. 5. P. 3544–3554. DOI: 10.1109/TII.2020.3023341.
16. Benkhelifa F., Bouazizi Y., McCann J. A. How orthogonal is LoRa modulation? *IEEE Internet of Things Journal*. 2022. Vol. 9, No. 20. P. 19928–19944. DOI: 10.1109/JIOT.2022.3173060.
17. Silva E. F., Figueiredo L. M., de Oliveira L. A., Chaves L. J., de Oliveira A. L., Rosário D., Cerqueira E. Adaptive Parameters for LoRa-Based Networks Physical-Layer. *Sensors*. 2023. Vol. 23, No. 10. P. 4597. DOI: 10.3390/s23104597.

18. Berto R., Napoletano P., Savi M. A LoRa-Based Mesh Network for Peer-to-Peer Long-Range Communication. *Sensors*. 2021. Vol. 21, No. 13. P. 4314. DOI: 10.3390/s21134314.
19. STMicroelectronics. NUCLEO-WL55JC: STM32 Nucleo-64 development board with STM32WL55JCI MCU, SMPS, supports Arduino and morpho connectivity. Product page. URL: <https://www.st.com/en/evaluation-tools/nucleo-wl55jc.html>
20. STMicroelectronics. STM32WL55JC datasheet: multiprotocol LPWAN dual-core 32-bit Arm Cortex-M4/M0+ MCUs, LoRa, (G)FSK, (G)MSK, BPSK, up to 256-Kbyte Flash memory, 64-Kbyte SRAM. DS13293. Rev. 5. 2025. URL: <https://www.st.com/resource/en/datasheet/stm32wl55jc.pdf>