НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут телекомунікаційних систем

Кафедра інформаційних технологій в телекомунікаціях

До захисту допущено:
В.о. завідувача кафедри
_____ Валерій ПРАВИЛО
«_____»_____2026 р

ДИПЛОМНА РОБОТА
на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційно-комунікаційні технології»
спеціальності 172 Телекомунікації та радіотехніка

на тему: Вдосконалений алгоритм маршрутизації в Mesh-мережах
Виконав : здобувач вищої освіти 4 курсу, групи ТІ-21
Казаков Микола Андрійович

Науковий керівник: доцент кафедри ІТТ НН ІТС, кандидат технічних наук, доцент Правило Валерій Володимирович _____

Рецензент: доцент кафедри ТК НН ІТС, кандидат технічних наук, доцент
Явіся Валерій Сергійович _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Здобувач вищої освіти _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра інформаційних технологій в телекомунікаціях

Рівень вищої освіти – перший (бакалаврський)

Освітньо-професійна програма «Інформаційно-комунікаційні технології»

спеціальності 172 Телекомунікації та радіотехніка

ЗАТВЕДЖУЮ
В.о. завідувача кафедри
_____ Валерій ПРАВИЛО
«_____»_____2026 р

ЗАВДАННЯ

на дипломну роботу студенту

Казакову Миколі Андрійовичу

1. Тема дипломної роботи: Вдосконалений алгоритм маршрутизації в Mesh-мережах, науковий керівник роботи доцент кафедри ІТТ НН ІТС кандидат технічних наук, доцент Правило Валерій Володимирович, затверджені наказом по університету від 26.05.26р. №1912-с

2. Строк подання студентом дипломної роботи 08.06.2026

3. Об’єкт дослідження: процес маршрутизації в mesh-мережах.

4. Вихідні дані до роботи: відомі протоколи маршрутизації для mesh-мереж (AODV, DSR, OLSR, DSDV, HWMP) та їх характеристики; метрики оцінювання якості маршруту (hop count, ETX, ETT, WCETT); стандарт IEEE 802.11s для Wi-Fi mesh-мереж; методи математичного моделювання з використанням теорії графів; інструментарій для симуляційного моделювання мережевих сценаріїв на мові Python; наукові публікації та стандарти у сфері бездротових мереж.

5. Перелік завдань, які потрібно розробити:

– проаналізувати архітектуру mesh-мереж та існуючі алгоритми маршрутизації;

– обґрунтувати метрики та розробити математичну модель і логіку вдосконаленого алгоритму;

– реалізувати програмний симулятор, модель вузла та обрані алгоритми для тестування;

– провести моделювання, порівняти результати та проаналізувати ефективність алгоритмів за ключовими метриками.

6. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація в PowerPoint, 25 рисунків.

7. Дата видачі завдання: 25.02.2026

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Затвердження теми та плану роботи. Аналітичний огляд літератури та існуючих протоколів маршрутизації в mesh-мережах	25.02.2026 - 10.03.2026	Виконано
2	Обґрунтування методології та вибір метрик оцінювання якості маршруту. Початок роботи над першим розділом	11.03.2026 - 27.03.2026	Виконано
3	Розробка математичної моделі комплексної оцінки маршруту. Робота над другим розділом	28.03.2026 - 17.04.2026	Виконано
4	Реалізація алгоритму MQAR та симулятора mesh-мережі. Проведення симуляційних експериментів. Аналіз отриманих результатів	18.04.2026 - 08.05.2026	Виконано
5	Висновки про ефективність вдосконаленого рішення	09.05.2026 - 15.05.2026	Виконано
6	Оформлення дипломної роботи	16.05.2026 - 05.06.2026	Виконано
7	Підготовка презентації до захисту	05.06.2026 - 09.06.2026	Виконано
8	Захист дипломної роботи	16.06.2026 - 30.06.2025	Виконано

Здобувач вищої освіти _____ Микола КАЗАКОВ

Керівник дипломної роботи_____ Валерій ПРАВИЛО

РЕФЕРАТ

Дипломна робота містить 102 сторінки, 25 рисунків, 26 формул. Використано 34 джерела інформації.

Актуальність теми. Бездротові mesh-мережі набувають дедалі ширшого застосування в різних сферах, від міських телекомунікаційних систем і промислового IoT до систем відеоспостереження та корпоративних мереж. Ефективна робота таких мереж критично залежить від якості алгоритмів маршрутизації, оскільки саме вони визначають шляхи передачі даних між вузлами. Найпоширеніші протоколи (AODV, HWMP, OLSR) використовують спрощені однометричні підходи (кількість хопів, ETX) та не здатні оперативно адаптуватися до динамічних змін топології і нерівномірного навантаження на вузли. Зокрема, стандартний протокол HWMP не враховує завантаженість вузлів та стабільність з'єднань, що призводить до неефективного використання пропускної здатності та частих перебудов маршрутів. Усе це обумовлює актуальність розробки вдосконаленого алгоритму маршрутизації з багатовимірним оцінюванням якості маршруту та механізмами проактивної адаптації.

Мета роботи полягає в покращенні пропускної здатності, надійності доставки та стабільності в mesh-мережах за рахунок застосування вдосконаленого алгоритму.

Об'єктом дослідження є процес маршрутизації даних у бездротових mesh-мережах.

Предметом дослідження є алгоритми та метрики маршрутизації в динамічних бездротових mesh-мережах.

Методи дослідження. Використано системний аналіз і класифікацію протоколів маршрутизації; математичне моделювання з використанням теорії графів та методів нормалізації; дискретно-подійне симуляційне моделювання; статистичну обробку результатів на основі t-розподілу Стьюдента з побудовою 95% довірчих інтервалів; порівняльний аналіз алгоритмів.

Ключові слова: MESH-МЕРЕЖА, МАРШРУТИЗАЦІЯ, MQAR, HWMP, AODV, МЕТРИКА ЯКОСТІ МАРШРУТУ, ПРОПУСКНА ЗДАТНІСТЬ, СТАБІЛЬНІСТЬ МАРШРУТУ, МОБІЛЬНІСТЬ ВУЗЛІВ, АДАПТИВНА МАРШРУТИЗАЦІЯ, СИМУЛЯЦІЙНЕ МОДЕЛЮВАННЯ, БАГАТОМЕТРИЧНА ФУНКЦІЯ.

ABSTRACT

The thesis consists of 102 pages, 25 figures, and 26 formulas. A total of 34 information sources were used.

Relevance of the topic. Wireless mesh networks are becoming increasingly widespread in various fields, ranging from urban telecommunication systems and industrial IoT to video surveillance systems and corporate networks. The efficient operation of such networks critically depends on the quality of routing algorithms, since they determine the data transmission paths between nodes. The most common protocols (AODV, HWMP, OLSR) use simplified single-metric approaches (hop count, ETX) and are unable to promptly adapt to dynamic topology changes and uneven node load distribution. In particular, the standard HWMP protocol does not take node congestion and link stability into account, which leads to inefficient bandwidth utilization and frequent route reconstructions. All this determines the relevance of developing an improved routing algorithm with multidimensional route quality assessment and proactive adaptation mechanisms.

The purpose of the study is to improve throughput, delivery reliability, and stability in mesh networks through the application of an enhanced routing algorithm.

The object of the study is the process of data routing in wireless mesh networks.

The subject of the study is routing algorithms and metrics in dynamic wireless mesh networks.

Research methods. The study uses system analysis and classification of routing protocols; mathematical modeling based on graph theory and normalization methods; discrete-event simulation modeling; statistical processing of results based on Student's t-distribution with the construction of 95% confidence intervals; and comparative analysis of algorithms.

Keywords: MESH NETWORK, ROUTING, MQAR, HWMP, AODV, ROUTE QUALITY METRIC, THROUGHPUT, ROUTE STABILITY, NODE MOBILITY, ADAPTIVE ROUTING, SIMULATION MODELING, MULTI-METRIC FUNCTION.

ЗМІСТ

СПИСОК ТЕРМІНІВ І СКОРОЧЕНЬ

AODV – Ad hoc On-Demand Distance Vector (реактивний протокол маршрутизації для бездротових мереж, що будує маршрути за запитом).

ARQ – Automatic Repeat reQuest (механізм автоматичного повторного запиту при помилці передачі, вбудований у стандарт IEEE 802.11)

BER – Bit Error Rate (частота бітових помилок; частка некоректно переданих бітів відносно загальної кількості)

BFS – Breadth-First Search (пошук у ширину; алгоритм обходу графа, що знаходить шлях із мінімальною кількістю ребер)

DSDV – Destination-Sequenced Distance Vector (проактивний протокол маршрутизації для ad-hoc мереж на основі вектора відстаней із механізмом порядкових номерів)

DSR – Dynamic Source Routing (реактивний протокол маршрутизації, при якому повний маршрут записується безпосередньо в заголовок пакета)

ETT – Expected Transmission Time (метрика якості каналу, що враховує як кількість повторних передач, так і швидкість каналу)

ETX – Expected Transmission Count (метрика якості каналу; очікувана кількість передач для успішної доставки одного пакета)

EWMA – Exponentially Weighted Moving Average (ковзне середнє з експоненціальним згасанням; використовується для згладжування вимірювань метрик мережі)

FSPL – Free Space Path Loss (модель втрат сигналу у вільному просторі)

HWMP – Hybrid Wireless Mesh Protocol (гібридний протокол маршрутизації стандарту IEEE 802.11s для Wi-Fi mesh-мереж)

IoT – Internet of Things (Інтернет речей; концепція мережі фізичних об'єктів, обладнаних засобами зв'язку)

LMT – Link Metric Table (таблиця метрик лінків; локальна структура даних вузла, що зберігає актуальні значення характеристик кожного з'єднання із сусідами)

MLEP – Metric Link Exchange Protocol (легковаговий протокол обміну метриками лінків, розроблений як частина алгоритму MQAR)

MPR – Multipoint Relays (механізм у протоколі OLSR, за яким лише обрані вузли розповсюджують оновлення маршрутизації)

MQAR – Multi-metric Quality-Aware Routing (вдосконалений алгоритм маршрутизації для mesh-мереж, розроблений у даній роботі; використовує комплексну зважену метрику Q(R))

OLSR – Optimized Link State Routing Protocol (проактивний протокол маршрутизації на основі стану каналів із оптимізацією MPR)

RREQ / RREP – Route Request / Route Reply (службові повідомлення запиту та відповіді при пошуку маршруту в реактивних протоколах).

RSSI – Received Signal Strength Indicator (індикатор рівня прийнятого сигналу)

RTT – Round-Trip Time (час обороту; час від надсилання пакета до отримання підтвердження)

VoIP – Voice over IP (технологія передачі голосу через IP-мережі)

WCETT – Weighted Cumulative Expected Transmission Time (метрика маршрутизації для багатоканальних мереж, що враховує затримку передачі та міжканальну інтерференцію)

Ad-hoc мережа – самоорганізована бездротова мережа без фіксованої інфраструктури

Джиттер – варіація затримки між послідовними пакетами

Маршрутизація – процес визначення шляху передачі даних у мережі від джерела до отримувача

Mesh-мережа (сітчаста мережа) – розподілена бездротова мережа, у якій кожен вузол може виконувати роль як кінцевого пристрою, так і проміжного ретранслятора.

Метрика маршрутизації – числовий критерій оцінювання якості маршруту.

Пропускна здатність – обсяг корисних даних, переданих через мережу за одиницю часу.

Самовідновлення – властивість mesh-мережі автоматично перебудовувати маршрути при виході вузла з ладу.

Хоп – один перехід між двома вузлами мережі.

ВСТУП

Сучасні бездротові mesh-мережі набувають дедалі ширшого застосування в різних сферах – від міських телекомунікаційних систем і промислового IoT до систем відеоспостереження та корпоративних мереж. Ключовою перевагою mesh-архітектури є її децентралізованість, самовідновлення та гнучкість масштабування без необхідності розгортання складної інфраструктури. Разом із тим ефективна робота mesh-мереж критично залежить від якості алгоритмів маршрутизації, що визначають шляхи передачі даних між вузлами.

Аналіз сучасного стану проблеми свідчить про суттєві обмеження існуючих рішень. Найпоширеніші протоколи (AODV, HWMP, OLSR) або використовують спрощені одновимірні метрики (кількість хопів, ETX), або не здатні оперативно адаптуватися до динамічних змін топології та нерівномірного навантаження на вузли. Зокрема, протокол HWMP, що є базовим стандартом IEEE 802.11s для Wi-Fi mesh-мереж, не враховує завантаженість вузлів та стабільність з'єднань при виборі маршруту. Це призводить до неефективного використання пропускної здатності мережі, надмірного перевантаження окремих вузлів та частих перебудов маршрутів при переміщенні вузлів або зміні умов середовища.

У роботі обґрунтовано необхідність переходу від однометричного до комплексного багатовимірного підходу до оцінювання якості маршруту. Розроблено алгоритм MQAR (Multi-metric Quality-Aware Routing), що поєднує затримку, рівень втрат пакетів, пропускну здатність та стабільність маршруту в єдиній зваженій функції якості Q(R). Вагові коефіцієнти функції адаптуються залежно від типу трафіку та умов роботи мережі. На відміну від існуючих рішень, MQAR реалізує механізм проактивного моніторингу якості поточного маршруту та переключається на кращий ще до розриву з'єднання — що усуває характерну для реактивних алгоритмів затримку при відновленні маршруту.

Результати роботи можуть бути застосовані при розробці програмного забезпечення для маршрутизаторів Wi-Fi mesh-систем; при проєктуванні промислових IoT-мереж та сенсорних систем; у розумних містах для організації мережевої інфраструктури без прокладання кабелів; при побудові тимчасових комунікаційних мереж у надзвичайних ситуаціях та в системах де критично важлива стійкість до відмов.

Об'єктом дослідження є процес маршрутизації даних у бездротових mesh-мережах.

Предметом дослідження є алгоритми та метрики маршрутизації в динамічних бездротових mesh-мережах.

Мета роботи полягає в покращенні пропускної здатності, надійності доставки та стабільності в mesh-мережах за рахунок застосування вдосконаленого алгоритму.

Завдання дослідження:

- проаналізувати архітектуру mesh-мереж та існуючі алгоритми маршрутизації;
- обґрунтувати метрики та розробити математичну модель і логіку вдосконаленого алгоритму;
- реалізувати програмний симулятор, модель вузла та обрані алгоритми для тестування;
- провести моделювання, порівняти результати та проаналізувати ефективність алгоритмів за ключовими метриками.

Методами дослідження є системний аналіз і класифікація протоколів маршрутизації; математичне моделювання з використанням теорії графів та методів нормалізації; дискретно-подійне симуляційне моделювання; статистична обробка результатів на основі t-розподілу Стьюдента з побудовою 95% довірчих інтервалів; порівняльний аналіз алгоритмів.

Елементом новизни є те, що розроблено алгоритм MQAR, що відрізняється від існуючих рішень комплексним врахуванням чотирьох взаємодоповнюючих метрик (затримка, втрати пакетів, пропускна здатність,

стабільність маршруту) у єдиній зваженій функції якості, а також механізмом проактивного перемикання маршрутів із гістерезисом для запобігання флапінгу. На відміну від HWMP, алгоритм враховує завантаженість вузлів і стабільність з'єднань при виборі маршруту.

Практичне значення. Розроблений алгоритм MQAR за результатами симуляції забезпечує підвищення ефективної пропускної здатності мережі на 8 – 23% залежно від розміру та умов, підвищення стабільності маршрутів на 16 – 24% та зниження рівня втрат пакетів на 7 – 8% порівняно з алгоритмами HWMP та AODV. Розроблений симулятор на мові Python може бути використаний для подальших досліджень алгоритмів

РОЗДІЛ 1
АНАЛІЗ MESH-МЕРЕЖ ТА ІСНУЮЧИХ АЛГОРИТМІВ МАРШРУТИЗАЦІЇ

1.1. Поняття mesh-мереж і їх роль у бездротових мережах.

Mesh-мережі (сітчасті мережі) – це один із сучасних підходів до організації бездротового зв'язку, який відрізняється від класичних мереж більш гнучкою та розподіленою структурою. В традиційних бездротових мережах (наприклад, звичайний Wi-Fi) всі пристрої підключаються до однієї точки доступу. В mesh-мережах кожен вузол може взаємодіяти не лише з центральним елементом, а й з іншими вузлами напряму.

Можна сказати, що mesh-мережа це розподілена система, в якій вузли можуть виконувати роль як кінцевих пристроїв, так і проміжних ретрансляторів. Це означає те, що дані не обов'язково будуть передаватися по одному фіксованому маршруту. Замість цього вони можуть проходити через декілька вузлів (їх називають «хопи»), поки не досягнуть пункту призначення. Такий підхід називається багатохоповою передачею. Він є ключовою особливістю mesh-мереж. Однією з головних переваг такого підходу є відсутність жорсткої централізації. У класичній мережі вихід з ладу точки доступу зазвичай означає недоступність всієї мережі. У mesh-мережах ситуація інша: якщо один вузол перестає працювати або втрачає зв'язок, мережа автоматично знаходить альтернативний маршрут. Це явище часто називають self-healing(самовідновлення) [2]. Завдяки цьому mesh-мережі мають вищу стійкість до відмов і краще працюють у нестабільних умовах [4].

Важливий момент – це масштабованість. Додавання нових вузлів у mesh-мережу зазвичай не потребує складної конфігурації. Новий пристрій підключається до сусідніх вузлів, і мережа автоматично інтегрує його в загальну структуру. Це значно спрощує розгортання мережі, особливо у великих або складних середовищах.

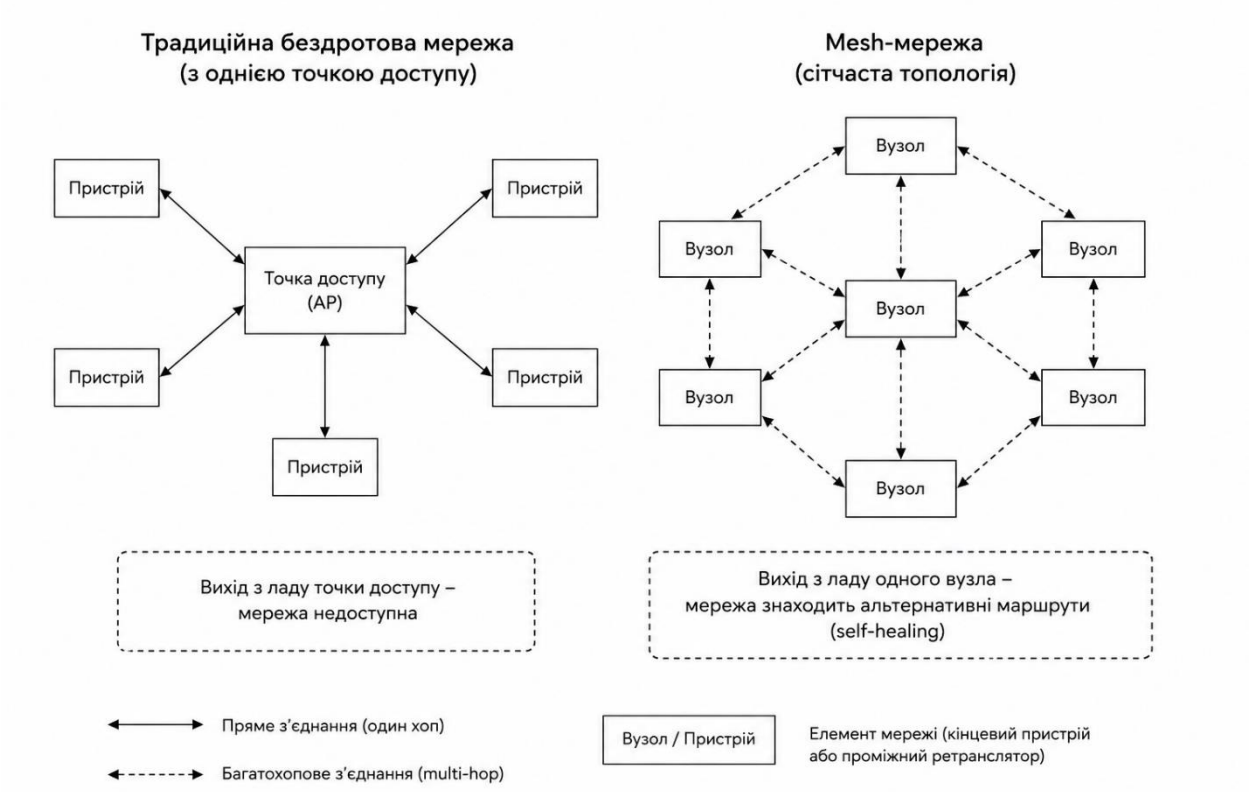

Рисунок 1.1 - Порівняння традиційної бездротової мережі та mesh-мережі

Варто також звернути увагу і на покриття. У звичайній мережі зона покриття обмежена радіусом однієї точки доступу. У mesh-мережах покриття розширюється за рахунок передачі сигналу між вузлами. Це дозволяє забезпечити зв'язок навіть у таких місцях, де пряме підключення до центрального вузла неможливе.

На практиці mesh-мережі застосовуються в різних сферах: від домашніх Wi-Fi систем до промислових і міських рішень (розумні міста, системи відеоспостереження, IoT-мережі) [6]. У таких сценаріях важливими є саме ті властивості, які mesh-мережі забезпечують: гнучкість, надійність, можливість працювати без складної інфраструктури.

Така структура ускладнює процес маршрутизації. Оскільки існує багато можливих шляхів передачі даних, система повинна обирати оптимальний маршрут з урахуванням різних факторів (якість каналу, навантаження, кількість переходів тощо). Саме тому питання ефективної маршрутизації є одним із ключових у mesh-мережах і потребує окремого детального дослідження.

1.2. Архітектура mesh-мереж.

Архітектура mesh-мереж визначає, як саме організована взаємодія між вузлами, які ролі вони виконують та яким чином відбувається передача даних у мережі. У більшій частині випадків можна виділити два основних типи вузлів: mesh routers (маршрутизатори) та mesh clients (клієнти) [2].

Mesh routers виконують ключову роль у мережі. Вони формують її основу і забезпечують передачу даних між різними вузлами. Зазвичай такі пристрої мають більше обчислювальних ресурсів, ніж звичайні клієнти: кращий процесор, більше оперативної пам’яті, а також кілька мережевих інтерфейсів. Це дозволяє їм ефективно обробляти маршрутизацію трафіку та підтримувати стабільність мережі. Деякі маршрутизатори можуть виконувати функцію шлюзу (забезпечувати вихід з mesh-мережі до інших мереж), тобто вони ще забезпечують її інтеграцію з зовнішнім середовищем, а не тільки передають дані всередині mesh-мережі [10].

Mesh clients – це кінцеві пристрої користувачів. Це можуть бути мобільні телефони, ноутбуки та різні IoT-пристрої. Вони підключаються до мережі для отримання доступу до ресурсів. У більшості випадків клієнти не беруть активної участі в маршрутизації, але в деяких архітектурах вони можуть також передавати трафік інших вузлів, це робить мережу ще більш децентралізованою та надійною.

З точки зору організації мережі, зазвичай виділяють три основні типи архітектури mesh-мереж:

1. Інфраструктурна архітектура (infrastructure mesh). У цьому випадку основну роль відіграють маршрутизатори. Вони формують стабільну мережеву сітку. Клієнти підключаються до маршрутизаторів як до звичайних точок доступу. Такий підхід дозволяє краще контролювати мережу, забезпечує більш стабільну роботу і зазвичай використовується в практичних системах (прикладом можуть бути міські або корпоративні мереі).

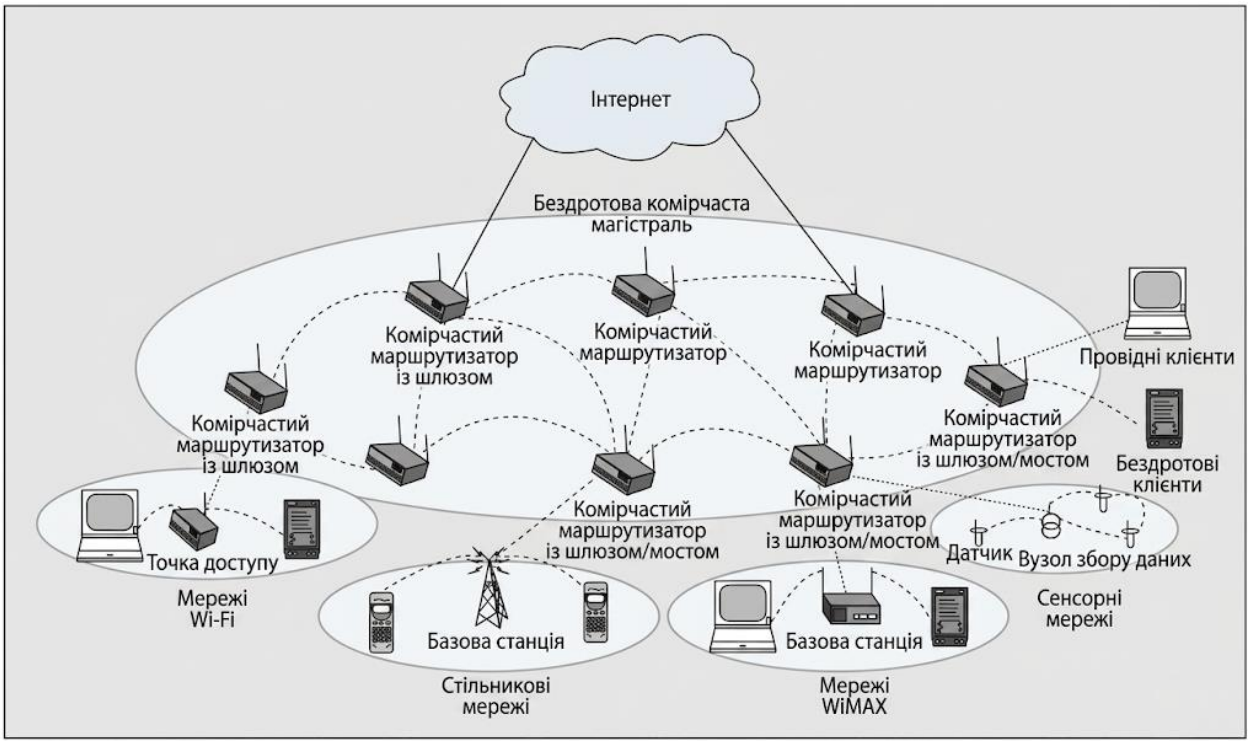

Рисунок 1.2 - Інфраструктурні/магістральні бездротові коміркові мережі [2]

2. Клієнтська архітектура (client mesh). У цьому варіанті всі вузли є рівноправними. Кожен пристрій може як отримувати дані, так і передавати їх далі. Це робить мережу максимально децентралізованою і незалежною від спеціалізованого обладнання. Така архітектура може бути менш стабільною, оскільки клієнтські пристрої зазвичай мають обмежені ресурси і часто змінюють своє положення або відключаються.
3. Гібридна архітектура (hybrid mesh). Цей підхід поєднує попередні два підходи. У мережі є окремі маршрутизатори, які забезпечують основу, але клієнти також беруть участь у передачі даних. Це дозволяє досягти балансу між стабільністю та гнучкістю. Гібридні mesh-мережі часто використовуються в реальних умовах, бо вони краще адаптуються до змін середовища [5].

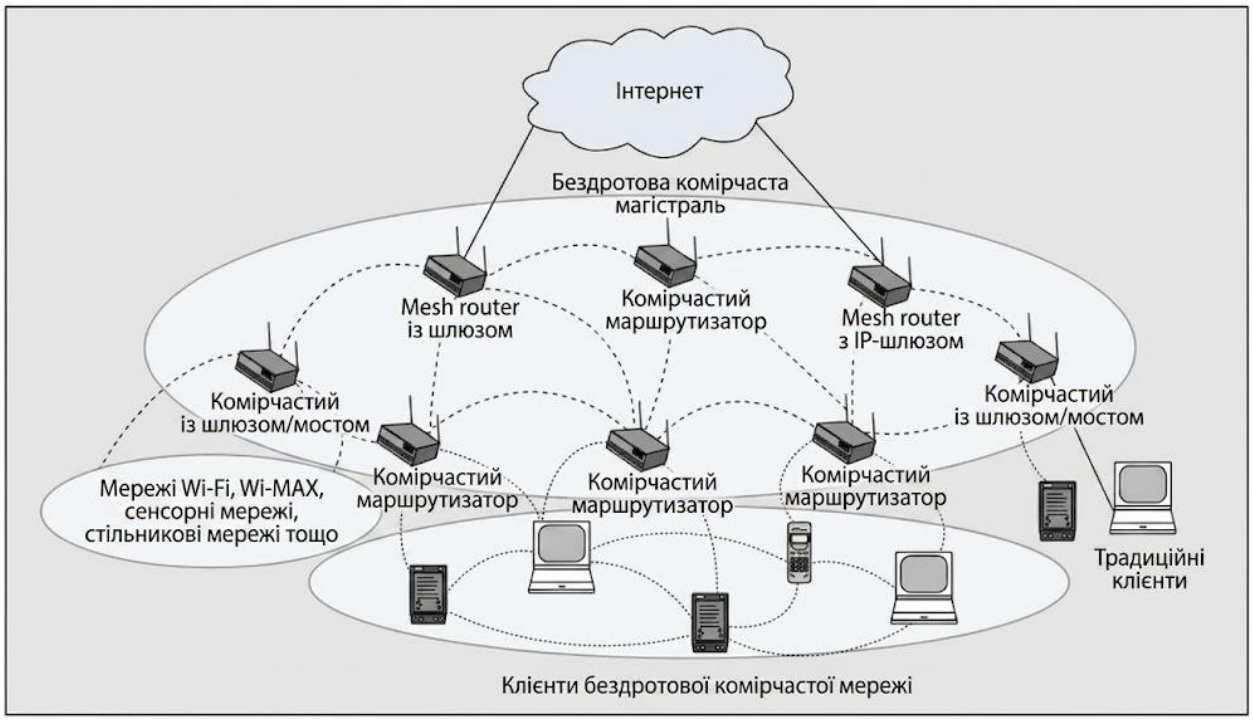

Рисунок 1.3 - Гібридні бездротові комірчасті мережі [2]

Важливо розуміти, що вибір архітектури безпосередньо впливає на ефективність роботи мережі. Наприклад, у клієнтських мережах маршрути можуть часто змінюватися через мобільність вузлів, що ускладнює маршрутизацію, а інфраструктурні мережі більш стабільні, але можуть вимагати додаткових витрат на обладнання.

На архітектуру також впливають такі фактори, як кількість вузлів, щільність їх розташування, тип трафіку та вимоги до якості обслуговування. Усі ці аспекти треба врахувати при проєктуванні mesh-мережі, особливо якщо мова йде про оптимізацію алгоритмів маршрутизації.

Архітектура mesh-мережі є базовим елементом, який визначає її поведінку, продуктивність і складність реалізації алгоритмів маршрутизації. Це робить її важливим об’єктом дослідження в рамках даної роботи.

1.3. Топології та варіанти побудови.

Топологія мережі визначає, яким чином вузли з’єднані між собою і як саме відбувається передача даних. У mesh-мережх топологія є однією з ключових характеристик, бо саме вона впливає на ефективність маршрутизації, затримки та загальну продуктивність мережі.

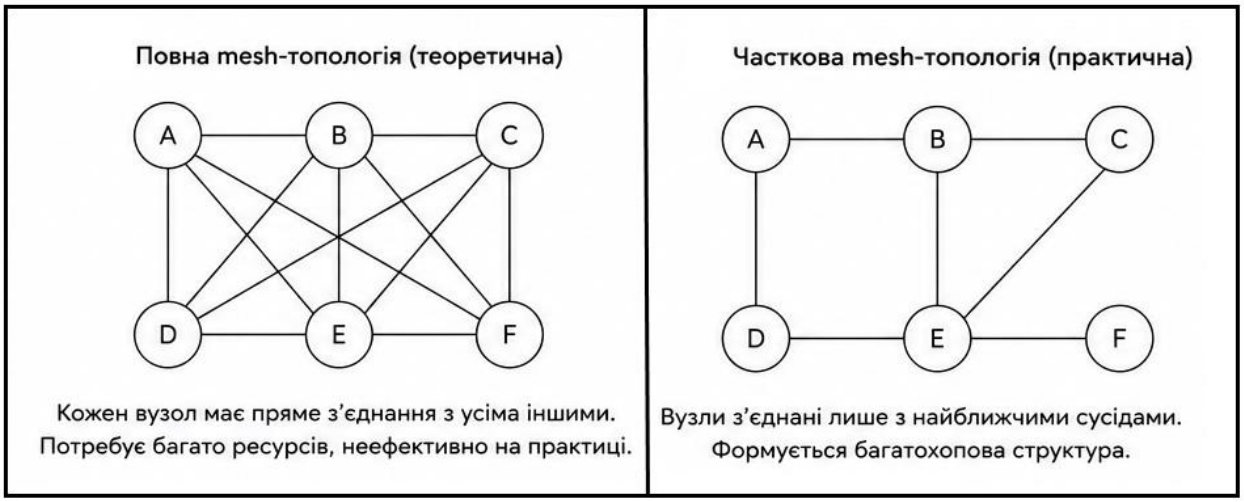

Рисунок 1.4 - Повна та часткова mesh-топології

У теорії mesh-топологія часто зображується як повністю зв'язана сітка, в якій кожен вузол має пряме з'єднання з усіма іншими. Але на практиці така ситуація майже не зустрічається, оскільки це потребує великої кількості ресурсів і через це не дуже ефективно. Реальні mesh-мережі зазвичай мають часткову сітчасту структуру, де вузли з'єднані лише з найближчими сусідами [2]. В таких мережах формується багатохопова структура: дані передаються від вузла до вузла, поки не досягнуть кінцевого пункту або шлюзу. Між двома вузлами може існувати кілька можливих маршрутів, і система маршрутизації повинна обирати найкращий з них. На цьому етапі топологія починає сильно впливати на роботу мережі.

Топологія mesh-мережі є динамічною. Вузли можуть з'являтися, зникати або змінювати своє положення. Особливо це видно, якщо мова йде про мобільні пристрої. Отже структура мережі постійно змінюється, а це означає що маршрути повинні регулярно оновлюватися. Це ускладнює роботу алгоритмів маршрутизації, але водночас робить мережу більш гнучкою/

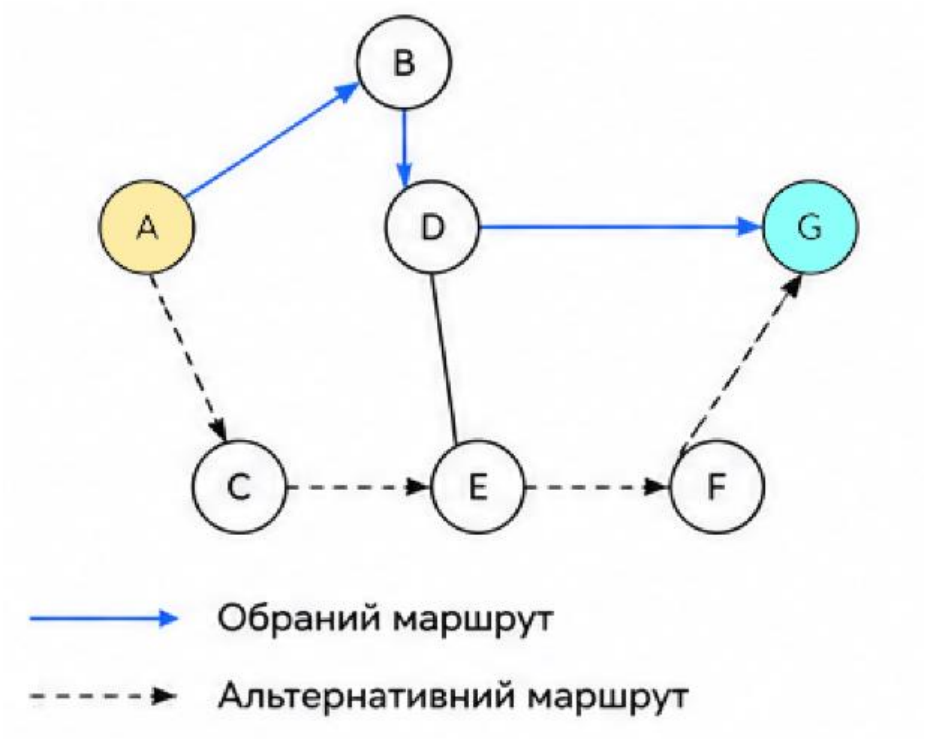

Рисунок 1.5 - Багатохопова структура

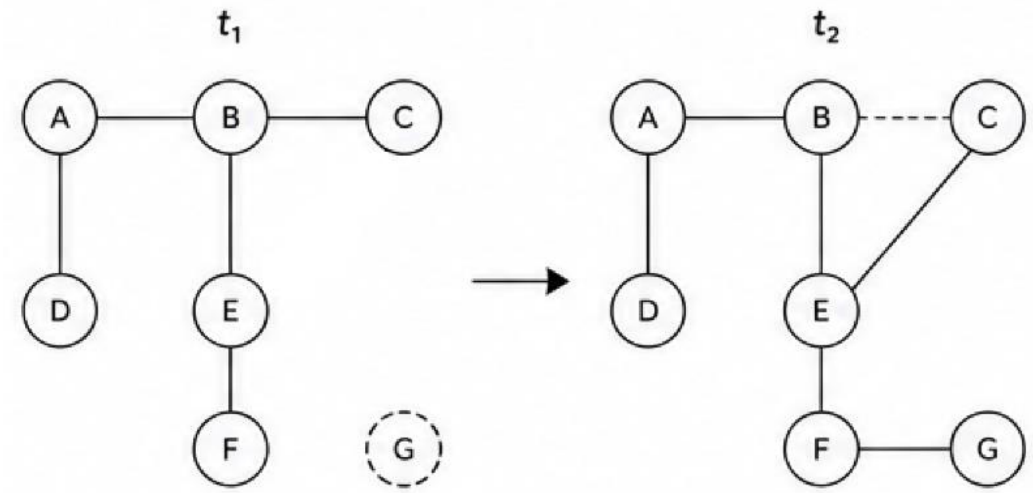

Рисунок 1.6 - Приклад динамічної топології

На рисунку 1.6 показано приклад динамічної топології mesh-мережі у моменти часу t_1 та t_2, де наочно продемонстровано процес самоорганізації рухомих вузлів. З плином часу структура мережі змінюється: у момент t_2 прямий зв'язок між вузлами B та C зникає, через що вузол C перенаправляє трафік через вузол E. Водночас у системі з'являється новий вузол G, який успішно встановлює з'єднання з вузлом F. Такі постійні зміни положення

пристроїв вимагають регулярного оновлення маршрутів, що ускладнює роботу алгоритмів, але робить мережу максимально гнучкою та стійкою.

Ще один важливий фактор це щільність вузлів. Якщо вузлів занадто мало, можуть виникати проблеми зі зв'язністю мережі і деякі ділянки можуть залишатися без покриття. Якщо ж вузлів дуже багато, виникає інша проблема – інтерференція та перевантаження каналів. При проєктуванні mesh-мережі важливо знайти баланс.

Існують різні варіанти побудови mesh-мереж залежно від умов використання. Наприклад, у міських мережах часто використовують стаціонарні маршрутизатори, які формують стабільну інфраструктуру. У мобільних мережах (наприклад, ad-hoc сценарії) вузли можуть постійно змінювати своє розташування, що робить топологію ще більш нестабільною [18].

Можна сказати, що топологія mesh-мережі – це не просто схема з'єднань, а динамічна структура, яка безпосередньо впливає на ефективність передачі даних, а її аналіз є важливим етапом при дослідженні алгоритмів маршрутизації.

1.4. Переваги та недоліки mesh-мереж.

Mesh-мережі мають багато переваг. Вони зробили їх популярними в сучасних бездротових системах.

Однією з головних переваг mesh-мереж є гнучкість розгортання. Для створення такої мережі не обов'язково прокладати кабелі або встановлювати складну інфраструктуру. Достатньо розмістити вузли в потрібних точках, і мережа зможе самостійно організувати з'єднання між ними.

Перевагою є масштабованість. У разі потреби можна просто додати нові вузли, і мережа автоматично розшириться. Це дуже зручно для великих систем, де мережа може поступово зростати без повної перебудови [6].

mesh-мережі забезпечують краще покриття території завдяки тому, що вузли можуть передавати сигнал один через одного. Можна охопити значно

більшу площу, ніж у випадку використання однієї точки доступу. Це робить їх зручними для використання в містах, кампусах або промислових зонах.

Важливою перевагою є і підвищена надійність. Якщо один вузол виходить з ладу, мережа може автоматично знайти інший маршрут для передачі даних. Така властивість дозволяє мережі залишатися працездатною навіть у випадку часткових відмов [4].

Разом з перевагами, mesh-мережі мають і ряд недоліків, які потрібно враховувати. Проблемою є інтерференція, бо вузли використовують бездротовий зв'язок і можуть працювати на однакових частотах. Сигнали можуть заважати один одному, що знижує якість передачі даних.

Змінність каналу також є проблемою. Змінність означає, що якість бездротового з'єднання може змінюватися залежно від відстані, перешкод, погодних умов та інших факторів. Це ускладнює вибір стабільного маршруту і може призводити до втрат пакетів та/або затримок.

Складною задачею є вибір оптимального маршруту . У mesh-мережі між вузлами може існувати багато шляхів, і не всі вони однаково ефективні. Алгоритми маршрутизації повинні враховувати різні параметри, такі як затримка, пропускна здатність, рівень сигналу та навантаження на вузли.

Ще існує проблема балансування навантаження. У випадку, коли більшість трафіку проходить через одні й ті ж вузли, вони можуть перевантажуватися, що призводить до зниження продуктивності всієї мережі. Для вирішення цієї проблеми треба рівномірно розподіляти трафік між різними маршрутами [3].

При великій кількості вузлів і активному трафіку можуть виникати проблеми зі зростанням затримок і зменшенням пропускної здатності. Це пов'язано з тим, що кожен додатковий хоп додає затримку і знижує ефективність передачі [5].

Mesh-мережі мають як значні переваги, так і обмеження. Їх ефективність багато в чому залежить від умов використання, правильно обраної архітектури та алгоритмів маршрутизації.

1.5. Фактори, що впливають на ефективність маршрутизації.

Ефективність маршрутизації в mesh-мережах залежить від великої кількості факторів. Це одна з причин, чому ця тема є досить складною. Навідміну від дротових мереж, де умови передачі більш стабільні, у бездротових mesh-мережах ситуація постійно змінюється, через це навіть добре спроектований алгоритм може працювати не ідеально в реальних умовах.

Важливу роль відіграє топологія мережі. Як вже зазначалося раніше, mesh-мережі мають динамічну структуру. Якщо вузли розташовані нерівномірно або між ними мало з'єднань, це може призводити до появи вузьких місць через які проходить основний трафік, у таких випадках ефективність маршрутизації знижується, оскільки деякі вузли перевантажуються.

Щільність вузлів теж важлива. З одного боку, більша кількість вузлів означає більше можливих маршрутів і кращу зв'язність мережі. З іншого боку, це також призводить до збільшення інтерференції та конкуренції за канал, тому надто велика щільність може навіть погіршити продуктивність, якщо алгоритм маршрутизації не враховує ці моменти.

Не менш важливим є якість бездротового каналу. У mesh-мережах зв'язок між вузлами залежить від багатьох факторів: відстані, перешкод (стіни наприклад), погодних умов, електромагнітних завад тощо, через це якість каналу може значно змінюватися навіть протягом короткого часу [28]. Якщо алгоритм маршрутизації не враховує ці зміни, він може обирати не найкращий маршрут, що призводить до втрат пакетів і збільшення затримок.

Ще одним фактором є інтерференція. Більшість mesh-мереж працює в обмеженому частотному діапазоні, це значить що вузли можуть заважати один одному. Особливо це буде помітно в густих мережах, де багато пристроїв працюють одночасно.

Також варто звернути увагу на навантаження в мережі. Якщо трафік розподілений нерівномірно, деякі вузли можуть бути перевантажені, тоді як інші залишаються майже не задіяними, у таких умовах навіть хороший маршрут з точки зору якості каналу може працювати повільно через перевантаження вузлів. Сучасні алгоритми маршрутизації намагаються враховувати не лише якість з'єднання, а й рівень завантаженості.

Окремо також можна виділити кількість хопів (переходів) у маршруті. Чим більше вузлів проходить пакет, тим більша затримка і вища ймовірність втрат, але при цьому короткий маршрут не завжди є найкращим, якщо він проходить через нестабільні або перевантажені вузли. Це створює компроміс, який повинен враховувати алгоритм маршрутизації.

Ще один важливий аспект -- це вибір метрики маршрутизації. У найпростішому випадку використовується кількість хопів, але цього часто недостатньо. Більш складні метрики враховують якість каналу, втрати пакетів, затримки та інші параметри. Від того, наскільки правильно обрана метрика залежить ефективність всієї мережі.

На маршрутизацію впливає мобільність вузлів. У випадку, якщо вузли постійно змінюють своє положення, маршрути можуть швидко застарівати. Такі зміни вимагають частого оновлення таблиць маршрутизації, що створює додаткове навантаження на мережу [18].

Обчислювальні ресурси вузлів впливають на маршрутизацію. У деяких випадках (особливо в IoT-мережах) пристрої мають обмежені можливості, і складні алгоритми маршрутизації можуть бути для них занадто тяжкими, це обмежує вибір підходів і змушує шукати компроміс між точністю і складністю.

Отже, ефективність маршрутизації в mesh-мережах залежить не від одного фактора, а від їх поєднання.

1.6. Класифікація алгоритмів маршрутизації

У mesh-мережах існує декілька основних підходів до організації маршрутизації, які відрізняються тим, як і коли формується маршрут між вузлами. Загалом алгоритми маршрутизації діляться на три групи: проактивні, реактивні та гібридні. Такий поділ використовується в літературі доволі часто, бо він дозволяє простіше зрозуміти принципи роботи різних протоколів і їхні особливості.

Проактивні алгоритми маршрутизації працюють за принципом постійного підтримування актуальної інформації про маршрути до всіх вузлів мережі, це означає, що кожен вузол заздалегідь має таблицю маршрутів і знає, через які проміжні вузли потрібно передавати дані до будь-якого іншого вузла в мережі. В мережі періодично відбувається обмін службовими повідомленнями, які оновлюють інформацію про топологію. Перевагою такого підходу є те, що передача даних відбувається дуже швидко, оскільки маршрут вже відомий і не потрібно витрачати час на його пошук. Недоліком є те, що навіть у випадку відсутності активного трафіку мережа все одно генерує службові повідомлення для підтримання актуальності таблиць, що збільшує навантаження на мережу. До прикладів таких алгоритмів можна віднести DSDV або OLSR [13].

Реактивні алгоритми маршрутизації, на відміну від проактмвних, не підтримують постійно актуальні маршрути, вони створюють маршрут тільки тоді, коли виникає потреба у передачі даних. Тобто якщо вузол хоче надіслати інформацію, але не знає маршруту, запускається процес його пошуку. У ході цього процесу розсилаються запити, після знаходження шляху він використовується для передачі даних. Перевагою такого підходу є зменшення службового трафіку, оскільки маршрути створюються тільки тоді, коли це необхідно. Але з іншого боку, при першій передачі даних виникає затримка, пов’язана з пошуком маршруту. Відомими реактивними алгоритмами є AODV та DSR [11].

Гібридні алгоритми маршрутизації поєднують у собі підходи проактивних і реактивних методів. Частина маршрутів може підтримуватися заздалегідь, а інша частина формуються тільки за потреби. Такий підхід дозволяє досягти балансу між швидкістю передачі даних і обсягом службового трафіку. Одним із найбільш відомих гібридних рішень у mesh-мережах є протокол HWMP, який використовується в стандарті IEEE 802.11s [1]. HWMP може працювати як у реактивному, так і в проактивному режимі, залежно від умов мережі та вимог до продуктивності.

У цілому можна сказати, що кожен із підходів має свої сильні та слабкі сторони. Проактивні алгоритми забезпечують швидкий доступ до маршрутів, але створюють додаткове навантаження на мережу. Реактивні алгоритми є більш економними з точки зору службового трафіку, але можуть викликати затримки при встановленні з’єднання. Гібридні алгоритми намагаються поєднати переваги обох підходів, проте є більш складними у реалізації та налаштуванні. Вибір конкретного алгоритму залежить від умов роботи мережі та вимог до її продуктивності.

1.7. Реактивні алгоритми маршрутизації

Реактивні алгоритми маршрутизації (або on-demand алгоритми) працюють за принципом побудови маршруту тільки тоді, коли це реально потрібно. На відміну від проактивних підходів, де таблиці маршрутів постійно підтримуються актуальними, у реактивних алгоритмах маршрут не зберігається заздалегідь. Маршрут створюється тільки в момент, коли один вузол хоче передати дані іншому, але не має інформації про шлях до нього. Такий підхід достатньо логічний для mesh-мереж, особливо якщо вони динамічні. Наприклад, для випадку, коли вузли часто змінюють своє положення або з’являються чи зникають з мережі, підтримувати постійно актуальні маршрути стає дорогим і неефективним, у цьому випадку реактивні алгоритми дозволяють зменшити навантаження на мережу, оскільки службові повідомлення передаються тільки тоді, коли це дійсно необхідно.

Процес роботи реактивних алгоритмів зазвичай складається з двох основних етапів. Спочатку виконується пошук маршруту. Якщо вузол-джерело не знає шляху до отримувача, він розсилає спеціальний запит (наприклад, RREQ у випадку AODV) [11]. Цей запит передається через сусідні вузли, поки не досягне цільового вузла або вузла, який вже знає маршрут. Після цього формується відповідь (RREP), яка повертається назад до джерела, і таким чином встановлюється маршрут. Другий етап – безпосередня передача даних по знайденому шляху. Поки маршрут є актуальним, дані передаються без додаткових пошуків. Якщо ж маршрут стає недійсним (наприклад, через вихід одного з вузлів з мережі), то процес пошуку запускається знову.

Головною перевагою реактивних алгоритмів є те, що вони не створюють постійного службового навантаження на мережу. Це особливо важливо для mesh-систем з великою кількістю вузлів, так як постійне оновлення таблиць маршрутів може займати значну частину пропускної здатності. Реактивні алгоритми краще підходять для середовищ, в яких трафік виникає нерегулярно.

Також є недоліки. Основна проблема полягає в затримці на етапі встановлення маршруту. Шлях шукається тільки при необхідності, через що перша передача даних завжди супроводжується додатковим часом щоб сформувати маршрут. У мережах, де важлива мінімальна затримка (мереж реального часу для прикладу), це може бути критичним фактором.

AODV (Ad hoc On-Demand Distance Vector) та DSR (Dynamic Source Routing), як зазначалося в минулому підрозділі, є реактивними. Ці два протоколи широко використовуються в дослідженнях і симуляціях mesh-мереж, так як вони добре демонструють принципи роботи реактивної маршрутизації, але при цьому мають різні підходи до зберігання і передачі маршрутної інформації.

У AODV маршрут будується поступово і зберігається у вигляді таблиць маршрутизації на кожному вузлі [11]. У DSR весь маршрут записується

безпосередньо в заголовок пакета і передається разом з ним, це робить DSR більш простим, але менш ефективним у великих мережах [31].

Реактивні алгоритми добре підходять для умов, де трафік є змінним і непередбачуваним, коли важливо зменшити службове навантаження на мережу. Використання таких алгоритмів вимагає врахувати затримки при встановленні з’єднання, це може накладати сильні обмеження у певних сценаріях.

1.8. Проактивні алгоритми маршрутизації

Проактивні алгоритми маршрутизації працюють по принципу постійного підтримування актуальної інформації про маршрути. Це означає, що кожен вузол заздалегідь має таблицю маршрутизації, у якій зберігається інформація про те, як дістатися до інших вузлів. Якщо в даний момент немає активної передачі даних, мережа все одно періодично оновлює ці таблиці, щоб вони залишались актуальними.

Ідея такого підходу полягає в тому, що маршрути вже готові до використання, тобто, коли виникає потреба передати дані, вузлу не потрібно витрачати час на пошук шляху, бо він просто бере інформацію з таблиці маршрутизації і одразу відправляє пакет. Через це проактивні алгоритми зазвичай забезпечують дуже малу затримку при передачі даних.

Така перевага має й зворотній бік, бо мережа постійно оновлює інформацію про маршрути, навіть тоді, коли трафіку немає, це створює додаткове навантаження на канали зв’язку. У великих або сильно завантажених мережах це може призвести до значного використання пропускної здатності тільки на службові повідомлення. Є ще одна особливість проактивних алгоритмів і вона полягає в тому, що вони краще працюють у більш стабільних мережах, де топологія змінюється не дуже часто. Якщо ж вузли часто переміщуються або зникають, таблиці маршрутизації можуть швидко втрачати актуальність, що призводить до зниження ефективності роботи мережі.

Прикладом проактивних алгоритмів є OLSR (Optimized Link State Routing Protocol). Він базується на класичному підході link-state, але використовує оптимізацію у вигляді multipoint relays (MPR), що дозволяє зменшити кількість службових повідомлень у мережі [13]. Ідея MPR полягає в тому, що не всі вузли розсилають оновлення, а лише обрані, які покривають максимальну кількість сусідів. Це частково зменшує перевантаження мережі, але не усуває його повністю.

Іншим представником є DSDV (Destination-Sequenced Distance Vector) -- це один із ранніх алгоритмів маршрутизації для ad-hoc мереж, який базується на класичному distance-vector підході. У ньому кожен вузол періодично розсилає свої таблиці маршрутизації сусідам. Для уникнення проблем із зацикленням маршрутів використовується механізм номерів послідовності (sequence numbers), який дозволяє визначати актуальність інформації [14].

Попри свою ефективність у швидкому доступі до маршрутів, проактивні алгоритми мають суттєвий недолік - це постійне навантаження на мережу незалежно від того, чи використовується вона активно. У mesh-мережах з великою кількістю вузлів це критично, оскільки частина пропускної здатності витрачається на підтримання таблиць, а не на передачу корисних даних.

1.9. Гібридні алгоритми маршрутизації

Гібридні алгоритми маршрутизації з’явилися як спроба поєднати переваги проактивних і реактивних підходів та зменшити їхні основні недоліки. Як уже було видно з попередніх підрозділів, проактивні алгоритми забезпечують швидкий доступ до маршрутів, але створюють значне службове навантаження на мережу. В свою чергу реактивні алгоритми зменшують кількість службових повідомлень, але додають затримку при встановленні маршруту. Гібридні алгоритми намагаються знайти баланс між двома крайностями.

У гібридному підході різні частини мережі або різні типи маршрутів обробляються по-різному. Наприклад, для напрямків, які використовуються найчастіше, маршрути можуть підтримуватися заздалегідь (як у проактивних алгоритмах), тоді як для інших вузлів маршрут будується тільки коли необхідно (як у реактивних). Такий підхід дозволяє зменшити затримки для популярних маршрутів і водночас не перевантажувати мережу зайвими оновленнями.

У mesh-мережах представником гібридних алгоритмів є HWMP (Hybrid Wireless Mesh Protocol). Цей протокол входить до стандарту IEEE 802.11s [24]. Це базовий протокол маршрутизації для Wi-Fi mesh-мереж, тому його часто розглядають як основний приклад гібридного підходу. HWMP може працювати у двох режимах: реактивному та проактивному. У реактивному режимі він діє подібно до протоколу AODV: маршрут формується тільки тоді, коли виникає потреба у передачі даних, для цього використовуються спеціальні повідомлення запиту і відповіді, які дозволяють знайти шлях між вузлами [16]. У проактивному режимі в мережі визначається один або кілька кореневих вузлів, до яких інші вузли підтримують маршрути заздалегідь. Це зручно, наприклад, у випадку, коли більшість трафіку спрямована до певного шлюзу або точки доступу до Інтернету. в такій ситуації маршрут до цього вузла вже відомий, і затримки при передачі мінімізуються.

Завдяки такій комбінованій роботі HWMP дозволяє адаптуватися до різних умов мережі. Якщо трафік є нерегулярним і розподіленим, більше використовується реактивний режим. Якщо ж є чітко виражені центри трафіку, ефективніше працює проактивна частина.

Гібридні алгоритми мають певні недоліки. По-перше, вони є більш складними у реалізації та налаштуванні, оскільки поєднують у собі два різні підходи. По-друге, у таких алгоритмах не завжди вдається повністю врахувати всі особливості мережі, наприклад змінність якості каналу або інтерференцію. Багато реалізацій HWMP використовують досить прості метрики, наприклад кількість хопів, що не завжди відображає реальну якість маршруту. У

результаті є можливість, що мережа може обирати не найефективніший шлях, хоча існують кращі альтернативи. Це буде помітно в умовах динамічної топології, де стан каналів може швидко змінюватися.

Гібридні алгоритми є більш універсальними у порівнянні з чисто проактивним або чисто реактивним підходами. Але вони все ще мають обмеження. Ці обмеження відкривають можливості для подальшого вдосконалення алгоритмів маршрутизації, зокрема шляхом використання більш складних метрик та адаптивних механізмів вибору маршруту.

1.10. Метрики вибору маршруту

Найпростішою і найстарішою метрикою є кількість хопів. Вона показує, через скільки вузлів потрібно передати пакет, щоб він досягнув кінцевого отримувача. На перший погляд, логіка проста: чим менше хопів – тим краще. Такий підхід добре працює в дротових мережах або в умовах, де всі канали приблизно однакові за якістю. У бездротових mesh-мережах ситуація значно складніша, тут різні канали можуть мати різну якість, різний рівень завад і втрат пакетів і в результаті маршрут з меншою кількістю хопів може працювати гірше, ніж довший маршрут, але з більш стабільними з'єднаннями. Тому використання тільки hop count часто призводить до неефективної маршрутизації. Щоб вирішити цю проблему, були запропоновані більш складні метрики, які враховують реальні умови передачі даних. Однією з найвідоміших є ETX (Expected Transmission Count). Вона оцінює, скільки передач у середньому потрібно для успішної доставки пакета по певному каналу з урахуванням втрат [7]. Якщо канал має високий рівень помилок, значення ETX буде більшим, і такий маршрут буде менш пріоритетним. Розвитком цієї ідеї стала метрика ETT (Expected Transmission Time). Вона враховує не тільки кількість повторних передач, а й швидкість каналу [17]. Це дозволяє більш точно оцінити, скільки часу займе передача даних. Таким чином, ETT краще підходить для мереж, де різні вузли можуть працювати на різних швидкостях.

Більш складною метрикою є WCETT (Weighted Cumulative Expected Transmission Time). Вона використовується у багатоканальних мережах та враховує не тільки затримку передачі, а й інтерференцію між каналами [17]. Такий підхід дозволяє уникати ситуацій, коли кілька лінків на маршруті працюють на одній частоті і заважають один одному.

Незважаючи на те, що ці метрики є більш точними, ніж простий hop count, вони також мають свої обмеження. Наприклад, більшість з них орієнтована тільки на характеристики каналу і не враховує такі важливі фактори, як навантаження на вузли або затримки, пов’язані з обробкою пакетів. У результаті навіть маршрут з хорошими показниками ETX або ETT може працювати неефективно, якщо вузли на цьому шляху перевантажені.

Ще одна проблема полягає в тому, що стан мережі може швидко змінюватися. Якість каналу, рівень завад і навантаження можуть змінюватися в реальному часі, і метрика, яка була актуальною кілька секунд тому, може вже не відповідати поточній ситуації. Через це виникає необхідність у використанні більш адаптивних підходів до оцінювання маршрутів.

У сучасних дослідженнях часто розглядаються комбіновані метрики, які враховують одразу кілька параметрів: затримку, втрати пакетів, пропускну здатність, навантаження на вузли та навіть енергоспоживання. Такі підходи дозволяють отримати більш точну оцінку якості маршруту, але при цьому ускладнюють сам алгоритм маршрутизації.

Вибір метрики маршрутизації є дуже важливим для ефективної роботи mesh-мережі. Прості метрики забезпечують швидкість і простоту, але не враховують всі особливості бездротового середовища. Більш складні метрики дають точніші результати, але потребують більше ресурсів і складніші в реалізації. Саме тому пошук оптимальної метрики або їх комбінації є одним із ключових напрямків досліджень у цій області і пов’язаний із розробкою нових алгоритмів маршрутизації.

1.11. Обмеження існуючих алгоритмів у динамічних mesh-мережах

Незважаючи на велику кількість існуючих алгоритмів маршрутизації, жоден із них не може забезпечити повністю ефективну роботу в умовах динамічних mesh-мереж. Це пов’язано з тим, що такі мережі мають ряд специфічних особливостей, які значно ускладнюють процес вибору оптимального маршруту.

Однією з основних проблем є динамічність топології мережі. У реальних умовах вузли можуть змінювати своє положення, тимчасово втрачати зв’язок або повністю виходити з мережі. У результати маршрути, які недавно були актуальними, можуть швидко втрачати свою ефективність. Проактивні алгоритми не встигають оперативно оновлювати таблиці маршрутизації, а реактивні витрачають додатковий час на повторний пошук маршрутів.

Важливим фактором є нестабільність бездротового каналу. На відміну від дротових мереж, у mesh-мережах якість зв’язку між вузлами може суттєво змінюватися через завади, перешкоди або зміну умов середовища, у таких ситуаціях правильно обраний маршрут має можливість швидко стати неефективним. Більшість класичних алгоритмів не враховує ці зміни в реальному часі або робить це недостатньо точно.

Значний вплив має інтерференція між вузлами. Багато пристроїв працюють у спільному частотному діапазоні, сигнали можуть заважати один одному. Це призводить до збільшення втрат пакетів і зниження пропускної здатності. При цьому більшість алгоритмів маршрутизації не враховує рівень інтерференції як окремий параметр при виборі маршруту [17].

Ще одна проблема – це нерівномірне навантаження на вузли мережі. У багатьох випадках алгоритм обирає маршрут, що є найкоротшим або має найкращі показники якості каналу, але при цьому не враховує завантаженість вузлів. В результаті окремі вузли можуть перевантажуватися, тоді як інші залишаються майже не використаними, це призводить до зниження загальної ефективності мережі.

Варто звернути увагу ще на обмеження метрик маршрутизації, які використовуються в існуючих алгоритмах. Як було розглянуто в

попередньому підрозділі, більшість метрик враховує лише окремі параметри, наприклад кількість хопів або якість каналу. Однак у реальних умовах на ефективність маршруту впливає значно більше факторів: затримка, втрати пакетів, пропускна здатність, навантаження, стабільність маршруту і тому подібне. Використання лише однієї або навіть кількох метрик не дозволяє отримати повну картину.

Алгоритми, які існують, часто мають обмежену адаптивність. Вони можуть добре працювати в певних умовах, але їх ефективність значно знижується при зміні параметрів мережі. Наприклад алгоритм, який добре показує себе в статичній мережі, може працювати значно гірше в умовах високої мобільності або інтенсивного трафіку.

Важливим аспектом є і затримки при перебудові маршрутів. У реактивних алгоритмах це пов’язано з необхідністю повторного пошуку маршруту, а у проактивних – з періодичністю оновлення таблиць. У будь-якому випадку це призводить до тимчасового зниження якості обслуговування і може бути критичним для додатків, чутливих до затримок.

Таким чином можна зробити висновок, що більшість існуючих алгоритмів маршрутизації мають певні обмеження, які особливо проявляються в умовах динамічних mesh-мереж. У загальному випадку ці обмеження можна звести до того, що алгоритми або не враховують всі важливі параметри мережі, або не здатні швидко адаптуватися до їх змін [3]. У результаті виникає необхідність в розробці нових підходів до маршрутизації, які б враховували комплекс характеристик мережі та могли адаптуватися до змін у реальному часі. Зокрема актуальним є використання комбінованих метрик, що дозволяють оцінювати маршрут з урахуванням декілька параметрів одночасно, а також механізмів динамічного оновлення маршрутів, залежно від поточного стану мережі. Саме ці задачі і визначають напрямок подальшого дослідження. У наступному розділі буде розглянуто підхід до формалізації критеріїв оптимізації маршрутизації, що дозволить більш точно оцінювати

якість маршрутів і стане основою для розробки вдосконаленого алгоритму маршрутизації.

Висновки

У першому розділі проведено комплексний аналіз mesh-мереж та існуючих алгоритмів маршрутизації. Встановлено, що mesh-мережі є розподіленими самоорганізованими системами з багатохоповою передачею даних, що забезпечують гнучкість розгортання, масштабованість та підвищену стійкість до відмов. Розглянуто три основні типи архітектури (інфраструктурна, клієнтська, гібридна) та особливості динамічної топології mesh-мереж.

Систематизовано фактори, що впливають на ефективність маршрутизації: топологія та щільність вузлів, якість і нестабільність бездротового каналу, інтерференція, нерівномірне навантаження, мобільність вузлів та обчислювальні ресурси пристроїв. Класифіковано алгоритми маршрутизації на проактивні (OLSR, DSDV), реактивні (AODV, DSR) та гібридні (HWMP), детально розглянуто принципи роботи, переваги та недоліки кожного класу.

Проведений аналіз показав, що існуючі алгоритми мають суттєві обмеження в умовах динамічних mesh-мереж: вони використовують спрощені метрики, не враховують комплекс параметрів мережі одночасно та не мають ефективних механізмів адаптивного реагування на зміни стану каналів. Це обґрунтовує необхідність розробки вдосконаленого алгоритму маршрутизації.

РОЗДІЛ 2
РОЗРОБКА АЛГОРИТМУ MQAR

2.1. Основні параметри ефективності mesh-мереж

Для оцінювання ефективності роботи mesh-мереж не є достатнім просто визначити існує маршрут між вузлами чи ні. Важливо не тільки забезпечити сам факт передачі даних, а ще й оцінити наскільки якісно та стабільно ця передача виконується. При аналізі та оптимізації алгоритмів маршрутизації використовується набір параметрів, які дозволяють оцінити стан мережі та ефективність роботи маршрутів.

Особливість mesh-мереж полягає в тому, що їхня продуктивність залежить від великої кількості факторів: якість бездротового каналу, навантаження на вузли, кількість проміжних переходів, рівень інтерференції та багатьох інших умов. Через це використання лише одного критерію оцінювання часто не дозволяє об’єктивно оцінити якість маршруту. Наприклад, маршрут може мати мінімальну кількість хопів, але при цьому характеризуватись великими затримками або високими втратами пакетів.

Одним із найважливіших параметрів ефективності є **затримка передачі даних** (delay). Вона показує час, який потрібен для доставки пакета від джерела до отримувача. У mesh-мережах затримка залежить не тільки від фізичної відстані між вузлами, а ще від кількості проміжних переходів, завантаженості мережі та стану каналу зв’язку. Особливо важливим цей параметр є для застосунків реального часу [22]. Прикладом може слугувати відеозв’язок або потокова передача даних, бо там навіть невеликі затримки можуть суттєво впливати на якість роботи системи.

Важливим параметром є **пропускна здатність** (throughput). Вона характеризує кількість корисних даних, які можуть бути передані через мережу за певний проміжок часу. Висока пропускна здатність означає, що мережа може ефективно обробляти значний обсяг трафіку без суттєвого зниження швидкості передачі. У mesh-мережах цей параметр сильно залежить

від кількості одночасних потоків даних, якості бездротового каналу та рівня інтерференції між вузлами.

Існує також **параметр втрати пакетів** (packet loss). У бездротових мережах частина пакетів може не доходити до отримувача через завади, нестабільність сигналу, перевантаження вузлів. Високий рівень втрат призводить до необхідності повторної передачі пакетів, а це збільшує навантаження на мережу та негативно впливає на затримки і пропускну здатність. По цій причині алгоритм маршрутизації повинен намагатися обирати маршрути з мінімальним рівнем втрат.

Ще одним параметром є **стабільність маршруту**. У mesh-мережах маршрути можуть часто змінюватися через динамічність топології або зміну якості каналів. Часта перебудова маршруту призводить до додаткових затримок і службового навантаження, тому стабільність маршруту є важливим критерієм, особливо для мереж в яких мобільність вузлів висока.

У mesh-мереж є параметр **енергоспоживання**. Це особливо актуально для автономних або сенсорних систем, де вузли працюють від акумуляторів. Якщо алгоритм маршрутизації постійно використовує одні й ті самі вузли (при умові що вони працюють від акумуляторів, а не від постійного джерела живлення) для передачі трафіку, вони швидше витрачають заряд і можуть раніше виходити з мережі. Зменшення рівня енергоспоживання може зробити мережу економічнішою та є корисним для навколишнього середовища. Через це сучасні підходи часто враховують рівень енергії вузлів при виборі маршруту [9].

Алгоритми маршрутизації створюють **службове навантаження**. Для підтримання маршрутів вузли повинні обмінюватися службовими повідомленнями, і якщо їх кількість є надто великою, це починає займати значну частину пропускної здатності мережі.

Важливим параметром є **масштабованість мережі**. Деякі алгоритми можуть добре працювати в невеликих mesh-мережах, але їх ефективність значно знижується зі збільшенням кількості вузлів. Це пов’язано зі зростанням

службового трафіку, складністю оновлення маршрутів і збільшенням навантаження на окремі вузли.

Ефективність mesh-мережі визначається не одним параметром, а цілим набором взаємопов’язаних характеристик. При цьому покращення одного показника може негативно впливати на інший. Для прикладу: зменшення затримки може вимагати використання коротших маршрутів, які при цьому будуть менш стабільними або більш перевантаженими. Саме тому при розробці алгоритмів маршрутизації необхідно враховувати комплекс параметрів і шукати компроміс між ними.

2.2. Аналіз впливу параметрів мережі на якість маршрутизації

Якість маршрутизації в mesh-мережі визначається складною системою взаємозалежностей між різними параметрами. У цьому підрозділі розглядається вплив 4 основних характеристик мережі на ефективність маршрутизації:

1. **Кількість вузлів у мережі.**

Більша кількість вузлів підвищує зв'язність мережі. Між будь-якими двома точками з’являється більше можливих маршрутів, а це збільшує надійність і дозволяє обирати кращі шляхи. Якщо позначити кількість вузлів через N, то кількість потенційних маршрутів між двома вузлами зростає приблизно так: $O(N^2)$.

Кожен додатковий вузол є можливим джерелом інтерференції, бо чим більше пристроїв працює одночасно в одному частотному діапазоні, тим вища ймовірність взаємних перешкод.

Більше вузлів означає більший обсяг службового трафіку при оновленні таблиць маршрутизації. Це збільшує навантаження на канали.

Можна сказати, що залежність між кількістю вузлів і ефективністю маршрутизації є нелінійною. До певного рівня збільшення кількості вузлів покращує мережу, після чого ефект стає зворотним.

2. **Щільність мережі.**

Щільність визначається як відношення кількості активних з’єднань до максимально можливої кількості з’єднань. Висока щільність мережі означає, що кожен вузол має багато сусідів і може вибирати з великої кількості варіантів наступного переходу.

У розріджених мережах проблемою є обмежена зв'язність. Деякі пари вузлів можуть мати лише один можливий маршрут, це робить розріджену мережу вразливою до відмов. У щільних мережах проблема протилежна, бо велика кількість активних передавачів у близькості один до одного призводить до збільшення інтерференції та конкуренції за доступ. Це знижує ефективну пропускну здатність кожного з'єднання.

Оптимальною щільністю буде така, де кожен вузол має достатньо альтернативних маршрутів, але при цьому рівень взаємних завад не робить суттєвих проблем.

3. Мобільність вузлів.

Мобільність є одним із найбільш дестабілізуючих факторів для алгоритмів маршрутизації. Коли вузли переміщуються, якість з'єднань між ними змінюється. Тобто одні канали погіршуються аж до розриву, а інші навпаки покращуються завдяки наближенню до вузлів. Через це маршрут, який був найкращим у момент його побудови, може стати неефективним або взагалі недійсним через деякий проміжок часу.

Для кількісної оцінки мобільності часто використовується поняття часу стабільності маршруту (Route Lifetime) [18]. Цей показник показує, як довго в середньому маршрут залишається актуальним. Чим вища мобільність вузлів, тим менший цей показник і тим частіше алгоритм повинен перераховувати маршрути. Часті перерахунки збільшують службовий трафік та вносять додаткові затримки, а це знижує якість обслуговування.

4. Інтерференція та перевантаження каналів.

Інтерференція виникає тоді, коли сигнали від двох і більше вузлів накладаються один на одного у зоні прийому, це призводить до погіршення відношення сигнал/шум і підвищення рівня бітових помилок (Bit Error Rate,

BER), що збільшує кількість втрачених пакетів та необхідність передавати їх повторно [29].

Перевантаження каналу відбувається тоді, коли сумарний трафік, що намагається пройти через з'єднання, перевищує пропускну здатність. У такій ситуації черга на передачу зростає, що збільшує затримку. Якщо черга переповнюється, пакети починають відкидатися, що збільшує втрати.

2.3. Формалізація метрик оцінювання маршруту

Для побудови математичної моделі оцінювання якості маршруту спочатку будуть введені строгі математичні визначення для кожної з метрик.

Нехай маршрут R між вузлами-джерелом s та вузлом-призначенням d описується як послідовність вузлів:

$$R = (v_0, v_1, v_2, \dots v_k), \text{ де } v_0 = s, v_k = d$$

$l_i = (v_{i-1}, v_i)$ буде позначати і-те з'єднання (лінк) на маршруті R, $i = 1, 2, \dots, k$. Кількість хопів маршруту дорівнює k.

Затримка (D). Загальна наскрізна затримка маршруту є сумою затримок на кожному лінку та кожному проміжному вузлі:

$$D(R) = \sum_{i=1}^{k}(d_{prop(l_i)} + d_{queue(v_{i-1})} + d_{trans(l_i)} + d_{proc(v_{i-1})}) \qquad (2.1)$$

де $d_{prop(l_i)}$ – затримка поширення сигналу на лінку l_i; $d_{trans(l_i)}$ – час передачі пакета, залежний від розміру пакета та пропускної здатності лінка; $d_{queue(v_{i-1})}$ – час очікування пакета у черзі вузла v_{i-1}, $d_{proc(v_{i-1})}$ – час обробки пакета процесором вузла v_{i-1}. Затримка вимірюється в мілісекундах і є мінімізованим показником: чим менше D(R), тим кращий маршрут [20].

Втрати пакетів (P_l). Ймовірність успішної доставки пакета по маршруту R визначається як добуток ймовірностей успішної передачі на кожному лінку:

$$P_{succes(R)} = \prod_{i=1}^{k}(1 - p_i) \qquad (2.2)$$

де p_i – ймовірність втрати пакета на лінку l_i.

Відповідно, показник втрат пакетів для маршруту R визначається як:

$$P_{l(R)} = 1 - P_{succes(R)} = 1 - \prod_{i=1}^{k}(1 - p_i) \qquad (2.3)$$

Навіть при відносно невисоких втратах на окремих лінках (наприклад, 5% на кожному) втрати для маршруту з багатьма хопами можуть бути суттєвими. Так, для k = 5 хопів з однаковою ймовірністю втрат 0,05 загальні втрати складуть близько 1- $(1\text{-}0{,}05)^5$ = 22,62%. Це обґрунтовує необхідність враховувати цю метрику при виборі маршруту.

Пропускна здатність (*T*). Ефективна пропускна здатність маршруту обмежується найбільш завантаженим або найслабшим лінком (принцип «вузького місця»):

$$T(R) = min\{i=1...k\} [C_i \times (1 - u_i)] \quad (2.4)$$

де $\boldsymbol{C_i}$ – максимальна пропускна здатність лінка $\boldsymbol{l_i}$ (Мбіт/с); $\boldsymbol{u_i}$ – коефіцієнт поточного завантаження лінка $\boldsymbol{l_i}$ (значення від 0 до 1). Таким чином, ***T(R)*** показує реальну пропускну здатність, яку може запропонувати маршрут R з урахуванням поточного навантаження.

Стабільність маршруту (*S*). Стабільність маршруту визначається як мінімальний очікуваний час до першого розриву будь-якого з лінків на маршруті:

$$S(R) = min\{i=1...k\} \tau_i \quad (2.5)$$

де $\boldsymbol{\tau_i}$ – очікуваний час стабільності лінка l_i, який може оцінюватися на основі поточного рівня сигналу, швидкості зміни рівня сигналу (тренду) та відносної швидкості переміщення вузлів. Чим більше S(R), тим довше маршрут залишатиметься дійсним без необхідності перерахунку.

Для оцінки якості окремого лінка широко використовується метрика ETX (Expected Transmission Count), яка визначається як очікувана кількість передач, необхідних для успішної доставки пакета:

$$ETX(l_i) = 1 / (df \times dr) \quad (2.6)$$

де *df* – ймовірність успішної передачі у прямому напрямку (від відправника до отримувача); *dr* – ймовірність успішної передачі підтвердження у зворотному напрямку [7]. Для маршруту R значення ETX є сумою по всіх лінках:

$$ETX(R) = \sum_{i=1}^{k} ETX(l_i) \quad (2.7)$$

Метрика ETT (Expected Transmission Time) є розширенням ETX і враховує не лише кількість повторних передач, а й швидкість каналу:

$$ETT(l_i) = ETX(l_i) \times (s / B_i) \quad (2.8)$$

де s – розмір пакета; B_i – пропускна здатність лінка l_i. ETT дозволяє порівнювати лінки, що працюють на різних швидкостях, оскільки враховує реальний час передачі, а не лише кількість спроб [17].

Таблиця 2.1 Зведена характеристика метрик маршрутизації

Метрика	**Позначення**	**Одиниця виміру**	**Напрямок оптимізації**	**Тип впливу**
Затримка	D	мс	Мінімізація	Час
Пропускна здатність	T	Мбіт/с	Максимізація	Пропускний
Втрати пакетів	P_l	%	Мінімізація	Надійність
Стабільність	S	с	Максимізація	Часовий
ETX	ETX	спроби	Мінімізація	Якість каналу
ETT	ETT	мс/пакет	Мінімізація	Якість каналу

2.4. Побудова математичної моделі оцінювання якості маршруту

Окремі метрики, введені у підрозділі 3.3, описують різні аспекти якості маршруту, але для прийняття єдиного рішення про вибір оптимального маршруту необхідна інтегрована оцінка, яка об'єднує всі ці показники в один числовий критерій.

Метрики мають різні одиниці вимірювання та різні діапазони значень, перед їх об'єднанням необхідно виконати нормалізацію. Для кожної метрики M визначається її нормалізоване значення $\hat{M}$ за формулою:

$$\widehat{M} = \frac{M - M_{min}}{M_{max} - M_{min}} \quad (2.9)$$

де M_{min} та M_{max} – мінімальне і максимальне значення відповідної метрики. Після нормалізації всі метрики приймають значення від 0 до 1.

Для метрик, що підлягають мінімізації (D та P_l), нормалізоване значення інтерпретується так: чим менше $\hat{M}$, тим кращий маршрут.

Для метрик, що підлягають максимізації (T та S), значення перетворюється за формулою:

$$\hat{M}' = 1 - \hat{M} \tag{2.10}$$

Завдяки цьому після нормалізації та перетворення для всіх метрик діє єдине правило: менше значення $\hat{M}$ відповідає кращому маршруту.

Якість маршруту R описується комплексним показником Q, що визначається як лінійна зважена комбінація нормалізованих метрик:

$$Q(R) = w_1 * \hat{D} + w_2 * \hat{P} + w_3 * (1 - \hat{T}) + w_4 * (1 - \hat{S}) \tag{2.11}$$

де $\hat{D}$, $\hat{P}_l$, $\hat{T}$, $\hat{S}$ – нормалізовані значення затримки, втрат пакетів, пропускної здатності та стабільності відповідно; w_1, w_2, w_3, w_4 – вагові коефіцієнти, що задовольняють умові:

$$w_1 + w_2 + w_3 + w_4 = 1, \quad w_i \geq 0 \tag{2.12}$$

Маршрут з меншим значенням $Q(R)$ є кращим. Оптимальний маршрут R^* визначається як:

$$R^* = \arg\min_{\{R \in \Pi(s,d)\}} Q(R) \tag{2.13}$$

де $\Pi(s,d)$ – множина всіх доступних маршрутів між вузлами s та d.

Вагові коефіцієнти w_i відображають пріоритетність тих чи інших метрик залежно від типу трафіку та умов роботи мережі. Їх правильний вибір є важливим для ефективного функціонування моделі.

Вибір вагових коефіцієнтів w_i базується на системному аналізі пріоритетів передачі даних для різних умов експлуатації mesh-мережі. Для сценарію «VoIP / реальний час» визначальним фактором є часова стабільність передачі, оскільки згідно з рекомендаціями міжнародного стандарту ITU-T G.114 [22], затримка в один бік не повинна перевищувати 150 мс для забезпечення прийнятної якості інтерактивного зв'язку. Відповідно найбільша вага $w_1 = 0{,}50$ призначена саме для мінімізації затримки, тоді як параметр втрати пакетів $w_2 = 0{,}20$ є вторинним через наявність у сучасних кодеках механізмів маскування втрат, а пропускна здатність $w_3 = 0{,}15$ має найнижчий

пріоритет через низьку інтенсивність голосового потоку. У сценарії «Передача файлів» основним завданням є максимізація швидкості доставки великих обсягів даних, тому пріоритетною стає пропускна здатність з коефіцієнтом w_3 = 0,40, вага втрат пакетів w_2 = 0,30 також залишається високою, бо в протоколах з встановленням з'єднання кожна втрата активує механізми контролю перевантаження, що призводить до різкого падіння швидкості, тоді як затримка w_1 = 0,10 має мінімальний вплив на ефективність процесу. Для сценарію «Мобільна мережа» головною проблемою є висока частота розривів маршрутів через переміщення вузлів, це спричиняє значне службове навантаження на перебудову топології. У таких умовах визначальною стає метрика стабільності з'єднання w_4 = 0,40, яка стимулює алгоритм обирати найбільш надійні маршрути з довшим часом життя, навіть якщо вони є менш оптимальними за іншими показниками. Таблиця 2.2 наводить рекомендовані значення вагових коефіцієнтів для трьох типових сценаріїв.

Таблиця 2.2 Рекомендовані вагові коефіцієнти для різних типів трафіку

Тип трафіку	w_1 (D)	w_2 (P_l)	w_3 (T)	w_4 (S)
VoIP / реальний час	0,50	0,20	0,15	0,15
Передача файлів	0,10	0,30	0,40	0,20
Мобільна мережа	0,20	0,25	0,15	0,40

Наведені значення є орієнтовними і можуть бути адаптовані залежно від конкретних вимог до мережі. У більш просунутих реалізаціях вагові коефіцієнти можуть змінюватися динамічно в режимі реального часу на основі поточного стану мережі та типу трафіку, що передається.

Важливою рисою запропонованої моделі є її масштабованість. Якщо потрібно, то до функції Q(R) можна додати додаткові метрики

(енергоспоживання E наприклад), додавши ваговий кофефіцієнт, але при цьому треба враховувати, що $\sum w_i = 1$.

2.5. Вибір критеріїв для вдосконаленого алгоритму маршрутизації

Математична модель, розроблена у попередньому підрозділі, встановлює теоретичну основу для оцінювання маршрутів. Для практичного використання в алгоритмі маршрутизації необхідно визначити конкретний набір критеріїв та обґрунтувати, чому саме вони є достатніми і необхідними для ефективної роботи в умовах динамічних mesh-мереж.

Найпоширеніша метрика, кількість хопів, не відображає реального стану мережі з кількох причин. По-перше, вона не враховує якість окремих лінків: маршрут з двох хопів через нестабільні або перевантажені вузли може бути значно гіршим, ніж маршрут з чотирьох хопів через стабільні і незавантажені з'єднання. По-друге, hop count не реагує на динамічні зміни в мережі: навіть якщо маршрут із меншою кількістю хопів раптово стає перевантаженим, алгоритм продовжує його використовувати.

Метрика ETX є суттєвим покращенням, оскільки враховує якість каналу. Однак вона не бере до уваги навантаження на вузли, мобільність та ефективну пропускну здатність при конкурентному завантаженні. ETT розширює ETX введенням швидкості каналу, але також не вирішує проблем зі стабільністю та балансуванням навантаження.

Для вдосконаленого алгоритму маршрутизації пропонується використовувати наступний набір критеріїв:

- Затримка (D) – враховується для забезпечення якості обслуговування для чутливих до часу застосунків.

- Втрати пакетів (P_l) – впливає на надійність передачі та необхідність повторних передач, збільшуючи навантаження.

- Завантаженість вузлів – дозволяє уникати перевантажених ділянок мережі та забезпечувати збалансований розподіл трафіку

- Стабільність маршруту (S) – мінімізує частоту перерахунку маршрутів і пов'язані з цим затримки та службовий трафік.

Перші три критерії визначають поточну ефективність маршруту, а четвертий його часову надійність. Вони формують збалансовану основу для прийняття рішень в умовах динамічного середовища.

Перевагою запропонованого набору критеріїв є те, що він охоплює всі три виміри якості маршруту: ефективність передачі (D, T), надійність (P_l) та часову стабільність (S). При цьому набір є достатньо компактним для практичної реалізації й не вимагає надмірних обчислювальних ресурсів від вузлів мережі.

Енергоспоживання як критерій не включено до базової версії алгоритму з метою збереження універсальності, бо воно є критичним лише для певного класу пристроїв (IoT, сенсорні мережі) і може бути додане як опціональний параметр для відповідних сценаріїв застосування.

Використання комплексної зваженої функції Q(R) з адаптивними ваговими коефіцієнтами дозволяє алгоритму гнучко реагувати на зміни в мережі та вимоги трафіку, це відрізняє її від класичних підходів, орієнтованих на одну статичну метрику.

2.6. Загальна концепція вдосконаленого алгоритму

Вдосконалений алгоритм маршрутизації, що розробляється в рамках цієї роботи, буде називатися MQAR (Multi-metric Quality-Aware Routing). Відмінність цього алгоритму від існуючих рішень полягає в таких трьох принципах:

- **Комплексна оцінка маршруту.** Алгоритм обчислює комплексний показник Q(R) замість однієї метрики. Цей показник об'єднує затримку, втрати пакетів, завантаженість лінків та стабільність маршруту з адаптивними ваговими коефіцієнтами.

- **Адаптивне оновлення маршрутів**. Алгоритм не чекає повного розриву маршруту, а відстежує погіршення метрик у реальному часі та

проактивно переключається на кращий маршрут, коли поточний перестає відповідати порогам якості.

- **Гібридна маршрутизація**. Для відомих напрямків (наприклад, до шлюзу) маршрути підтримуються проактивно. Для нових або рідких напрямків використовується реактивний пошук, це дозволяє зменшити службове навантаження.

Найбільш близьким існуючим рішенням є HWMP (стандарт IEEE 802.11s). HWMP також є гібридним алгоритмом, однак його основна метрика airtime link metric враховує лише якість фізичного каналу, але не завантаженість вузлів і не стабільність маршруту [16]. MQAR розширює цей підхід, додаючи ці параметри до функції оцінки, і вводить механізм проактивного перерозподілу трафіку ще до того, як маршрут розірвався.

2.7. Архітектура алгоритму та його компоненти

Алгоритм MQAR складається з 4 функціональних модулів. Взаємодія між модулями забезпечує повний цикл, від збору даних про стан мережі до прийняття рішення про маршрут і його оновлення.

Таблиця 2.3. Функціональні модулі алгоритму MQAR

Модуль	Назва	Функція
M1	Збір метрик (Metric Collector)	Вимірює D, P_l, T, S для кожного лінка
M2	Оцінювач маршрутів (Route Evaluator)	Обчислює Q(R) для кандидатів на маршрут
M3	Селектор маршруту (Route Selector)	Обирає R* = arg min Q(R)
M4	Монітор якості (Quality Monitor)	Відстежує деградацію поточного маршруту і ініціює перерахунок

У модулі M1 кожен вузол мережі періодично надсилає пробні пакети сусідам і отримує підтвердження. На основі цього обміну вимірюються:

- ймовірність втрати пакетів p_i для кожного лінка (через відношення відправлених до підтверджених пробних пакетів);
- RTT (round-trip time), з якого оцінюється $d_{prop} + d_{trans}$;
- завантаженість лінка u_i на основі локальної статистики переданого та прийнятого трафіку;
- тренд рівня сигналу RSSI для оцінки стабільності τ_i.

Для згладжування випадкових коливань використовується ковзне середнє з експоненціальним згасанням (EWMA):

$$\bar{M}(t) = \alpha * M(t) + (1 - \alpha) * \bar{M}(t - 1) \qquad (2.14)$$

де M(t) – поточне значення метрики; $\bar{M}(t-1)$ – попереднє згладжене значення; $\alpha \in (0, 1)$ – коефіцієнт згладжування.

Цей підхід дозволяє алгоритму реагувати на реальні зміни стану мережі, не піддаючись впливу короткочасних сплесків.

Отримуючи актуальні метрики від M1, модуль M2 обчислює комплексний показник Q(R) для кожного кандидата на маршрут відповідно до моделі, побудованої у підрозділі 2.4. Нормалізація метрик виконується відносно поточних мінімальних і максимальних значень у мережі, які поширюються через службові повідомлення.

Модуль M3 обирає маршрут R* з мінімальним Q(R) серед усіх кандидатів. Кандидати формуються двома способами: проактивно (зі збережених записів у таблиці маршрутизації) та реактивно (через механізм flooding-запиту, аналогічно RREQ в AODV, у разі відсутності відомих маршрутів).

Модуль M4 постійно відстежує значення Q(R) для поточного активного маршруту. Якщо Q(R) перевищує заданий поріг $Q_{threshold}$ або якась окрема метрика виходить за межі допустимого діапазону, модуль ініціює запит на перерахунок маршруту в M2 і M3. Це відбувається до того, як маршрут розривається, що усуває паузу, характерну для реактивних алгоритмів.

2.8. Механізм збору та оновлення метрик

Для ефективної роботи алгоритму потрібна актуальна інформації про стан кожного лінка, для цього використовується легковаговий протокол обміну метриками MLEP (Metric Link Exchange Protocol), що є частиною алгоритму.

Службове повідомлення має структуру, у якій кожен вузол з інтервалом T_{hello} надсилає сусідам Hello-повідомлення, яке містить:

- ідентифікатор вузла (Node ID);
- поточний рівень завантаженості вузла, %;
- список сусідів із значеннями p_i та RSSI для кожного з них;
- мітку часу для синхронізації вимірювань затримки.

На основі отриманих Hello-повідомлень кожен вузол будує локальну таблицю метрик лінків (Link Metric Table, LMT), яка зберігає згладжені значення D, P_l, T, S для кожного сусіднього лінка. Ця таблиця є вхідними даними для модулів M2 і M3.

Для проактивного режиму необхідно мати інформацію про метрики не лише сусідніх лінків, а й усього шляху до кореневого вузла (наприклад, шлюзу). Для цього вузли, що знаходяться ближче до кореня, поширюють агреговані метрики маршруту через Metric Advertisement повідомлення. Кожне таке повідомлення містить накопичене значення Q_{path} (часткову суму показника якості від кореня до поточного вузла):

$$Q_{path}(v_1 \to root) = \sum_{j=1}^{i} (w_1 * \hat{D} + w_2 * \hat{P} + w_3 * (1 - \hat{T}) + w_4 * (1 - \hat{S})) \quad (2.15)$$

Вузол, що отримав таке повідомлення, додає до Q_{path} метрики власного лінка до відправника і, таким чином, отримує оцінку повного маршруту до кореня. Якщо нове значення Q_{path} є меншим за поточне, то вузол оновлює свій запис у таблиці маршрутизації.

Для балансу між актуальністю даних і службовим навантаженням на мережу використовується адаптивний підхід до вибору інтервалу оновлення.

В стабільних умовах (в таких, коли значення метрик змінюються повільно) інтервал T_{hello} може збільшуватися, знижуючи навантаження. При виявленні швидких змін метрик інтервал зменшується, забезпечуючи більш оперативну реакцію. Інтервали T_{min} та T_{max} задаються як конфігураційні параметри алгоритму.

2.9. Механізм адаптивного оновлення маршрутів

Адаптивне оновлення маршрутів є ключовою функцією, що відрізняє MQAR від класичних алгоритмів. Замість реакції лише на розрив маршруту (як у реактивних алгоритмах) або чекання наступного оновлення маршруту (як у проактивнтх алгоритмах), MQAR відстежує тенденції деградації та перемикається заздалегідь.

Модуль M4 безперервно порівнює поточне значення $Q(R_{active})$ з порогом $Q_{threshold}$. Перерахунок маршруту ініціюється при виконанні будь-якої з цих умов:

- $Q(R_{active}) > Q_{threshold}$ -- загальна якість маршруту впала нижче прийнятного рівня;
- $D(R_{active}) > D_{max}$ – затримка перевищила максимально допустиму для поточного типу трафіку;
- $P_l(R_{active}) > P_{lmax}$ – рівень втрат перевищив максимальне значення;
- $S(R_{active}) < S_{min}$ – прогнозований час стабільності маршруту став нижче мінімально допустимого.

Значення порогів $Q_{threshold}$, D_{max}, P_{lmax} і S_{min} є конфігураційними параметрами і можуть змінюватися залежно від типу трафіку. Це безпосередньо пов'язано з таблицею вагових коефіцієнтів (Таблиця 3.2), тобто при переключенні типу трафіку змінюються одночасно і ваги w_i і порогові значення.

Щоб уникнути флапінгу (так називається часте перемикання між двома маршрутами з близькими значеннями Q), вводиться гістерезис, тобто перехід

до нового маршруту R_{new} дозволяється лише тоді, коли виконується така умова:

$$Q(R_{active}) - Q(R_{new}) > \delta$$

де δ – мінімально значуща різниця показників якості. Таке рішення забезпечує стабільність вибору маршруту і запобігає надмірному службовому трафіку.

При прийнятті рішення про зміну маршруту алгоритм не переключає весь трафік миттєво, замість цього використовується поступове перенаправлення: спочатку новий маршрут R_{new} отримує частку трафіку β (наприклад, β = 0.3), а поточний R_{active} продовжує обслуговувати решту. Якщо через інтервал T_{verify} метрики R_{new} підтверджують очікувану якість, то відбувається повне перемикання, в іншому випадку алгоритм повертається до R_{active}. Такий підхід мінімізує погіршення якості обслуговування в момент переходу.

2.10. Опис алгоритму у вигляді псевдокоду

В цьому підрозділі буде наведено псевдокод основних компонентів алгоритму MQAR. Псевдокод описує логіку роботи кожного з чотирьох модулів, їх взаємодію та умови прийняття рішень.

Алгоритм ініціалізації вузла (Node Initialization):

```
PROCEDURE NodeInit(nodeId):
  LMT ← порожня таблиця метрик лінків
  RT  ← порожня таблиця маршрутизації
  SET T_hello ← T_default
  SET weights [w1, w2, w3, w4] ← defaultWeights
  Запустити модуль M1 (MetricCollector)
  Запустити модуль M4 (QualityMonitor)
END PROCEDURE
```

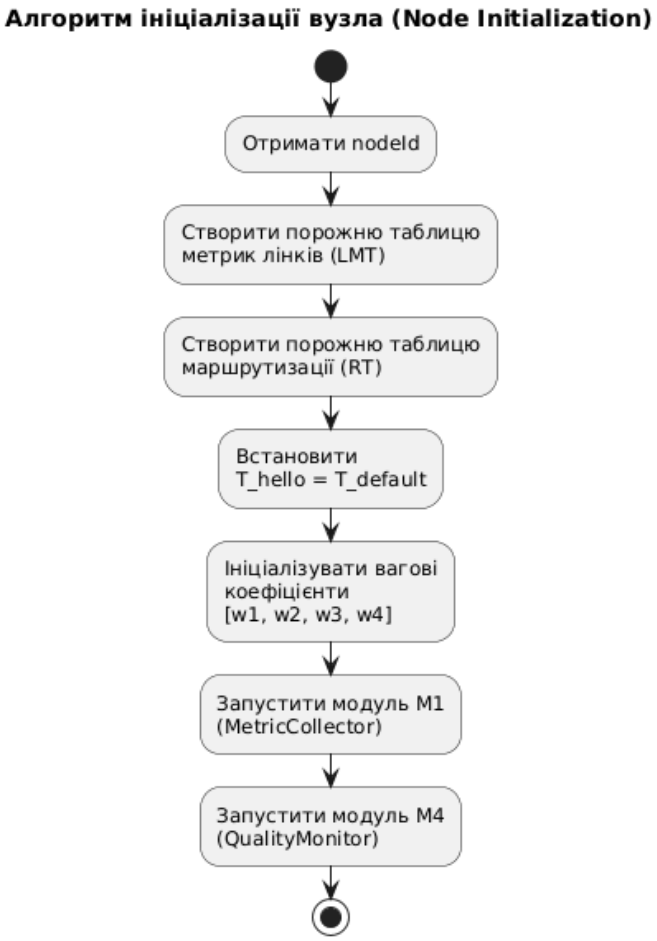

Рисунок 2.1 - Блок-схема процедури ініціалізації вузла в алгоритмі MQAR

Алгоритм збору метрик (MetricCollector, модуль M1)

LOOP кожні T_{hello} секунд:

FOR EACH сусідній вузол n_i:

НАДІСЛАТИ Hello(nodeId, timestamp, queueUtil)

ОТРИМАТИ Hello(n_i, rssi_i, pLoss_i, rtt_i)

// EWMA-згладжування

$LMT[n_i].D \leftarrow \alpha \cdot rtt_i/2 + (1-\alpha) \cdot LMT[n_i].D$

$LMT[n_i].P_l \leftarrow \alpha \cdot pLoss_i + (1-\alpha) \cdot LMT[n_i].Pl$

$LMT[n_i].T \leftarrow \alpha \cdot linkBW_i + (1-\alpha) \cdot LMT[n_i].T$

$LMT[n_i].S \leftarrow EstimateStability(rssi_i, rssi_{trend\ i})$

END FOR

АДАПТУВАТИ T_{hello} (збільшити якщо мережа стабільна, зменшити якщо виявлено деградацію)

END LOOP

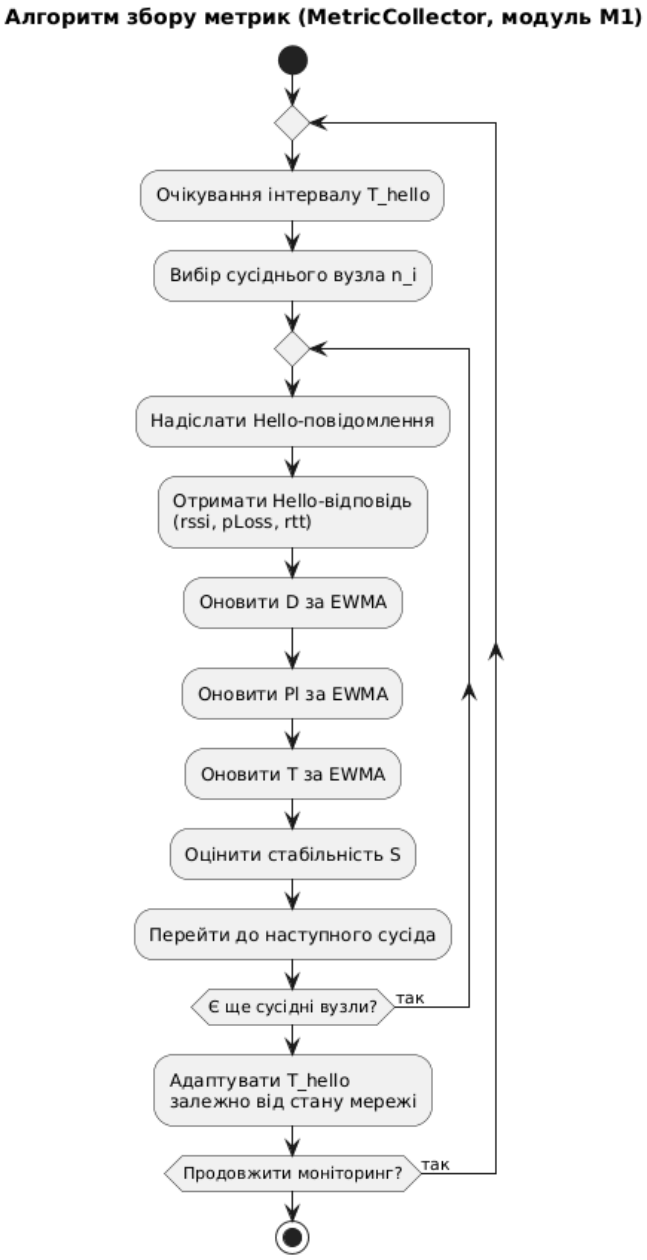

Рисунок 2.2 - Блок-схема алгоритму збору метрик

Алгоритм оцінки маршруту (RouteEvaluator, модуль M2)

FUNCTION EvaluateRoute(R):

Q ← 0

FOR EACH лінк l_i у маршруті R:

D_{norm} ← Normalize(LMT[l_i].D, D_{min}, D_{max})

P_{l_norm} ← Normalize(LMT[l_i].P_l, P_{l_min}, P_{l_max})

T_{norm} ← Normalize(LMT[l_i].T, T_{min}, T_{max})

$S_{norm} \leftarrow \text{Normalize}(LMT[l_i].S,\ S_{min},\ S_{max})$

$Q \leftarrow Q + w_1 \cdot D_{norm} + w_2 \cdot P_{l\_norm} + w_3 \cdot (1 - T_{norm}) + w_4 \cdot (1 - S_{norm})$

END FOR

RETURN Q

END FUNCTION

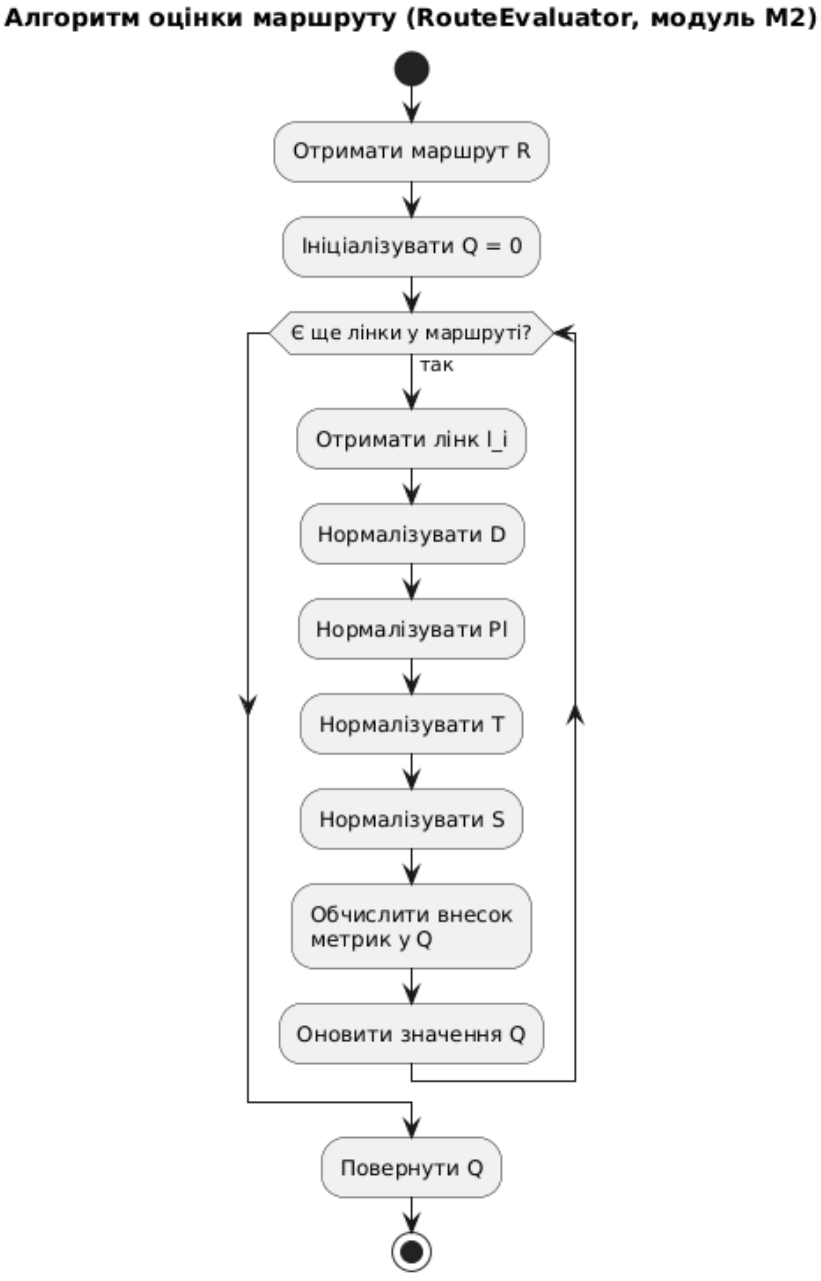

Рисунок 2.3 - Блок-схема алгоритму оцінки маршруту в модулі M2

Алгоритм вибору маршруту (RouteSelector, модуль M3)

FUNCTION SelectRoute(destination d):

```
  candidates ← RT.getKnownRoutes(d)  // проактивні кандидати
  IF candidates порожній:
    candidates ← FloodingSearch(d)    // реактивний пошук
  END IF
  bestRoute ← NULL
  bestQ ← ∞
  FOR EACH маршрут R у candidates:
    q ← EvaluateRoute(R)
    IF q < bestQ:
      bestQ ← q
      bestRoute ← R
    END IF
  END FOR
  RETURN bestRoute
END FUNCTION
```

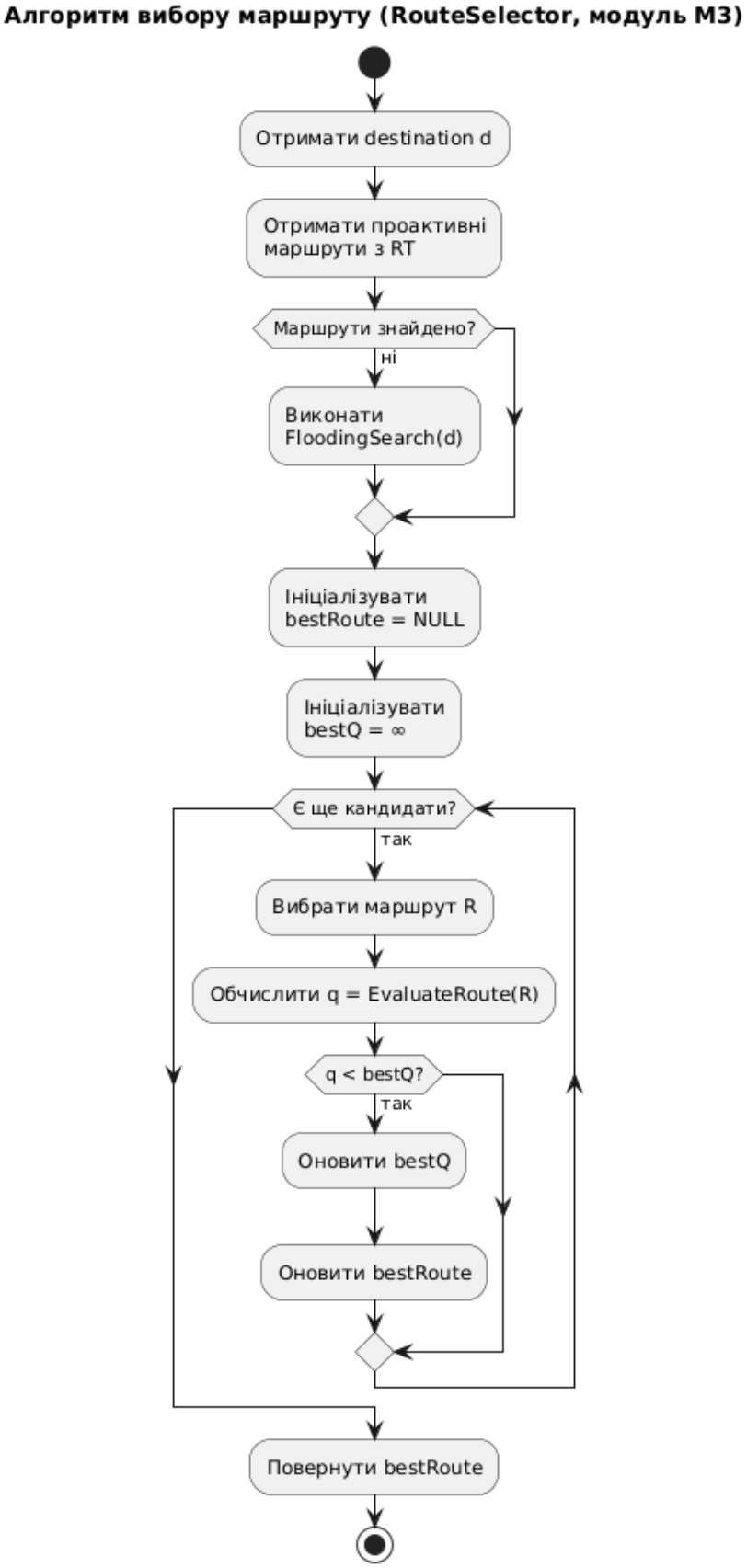

Рисунок 2.4 - Блок-схема алгоритму вибору маршруту в модулі M3

Алгоритм моніторингу якості та адаптивного оновлення (QualityMonitor, модуль M4)

LOOP кожні $T_{monitor}$ секунд:
  FOR EACH активний маршрут R_{active}:
    $Q_{cur} \leftarrow EvaluateRoute(R_{active})$
    IF $Q_{cur} > Q_{threshold}$ OR
      $LMT.D(R_{active}) > D_{max}$ OR
      $LMT.P_l(R_{active}) > P_{l\_max}$ OR
      $LMT.S(R_{active}) < S_{min}$:
      $R_{new} \leftarrow SelectRoute(R_{active.destination})$
      $Q_{new} \leftarrow EvaluateRoute(R_{new})$
      IF $Q_{cur} - Q_{new} > \delta$: // перевірка гістерезису
        ЗАПУСТИТИ плавне перемикання:
          $\beta \cdot$ трафік $\rightarrow R_{new}$, $(1-\beta) \cdot$ трафік $\rightarrow R_{active}$
        ПІСЛЯ T_{verify}:
          IF Q_{new} все ще $< Q_{cur}$:
            ПОВНІСТЮ перемкнути на R_{new}
          ELSE:
            ЗАЛИШИТИСЬ на R_{active}
          END IF
      END IF
    END IF
  END FOR
END LOOP

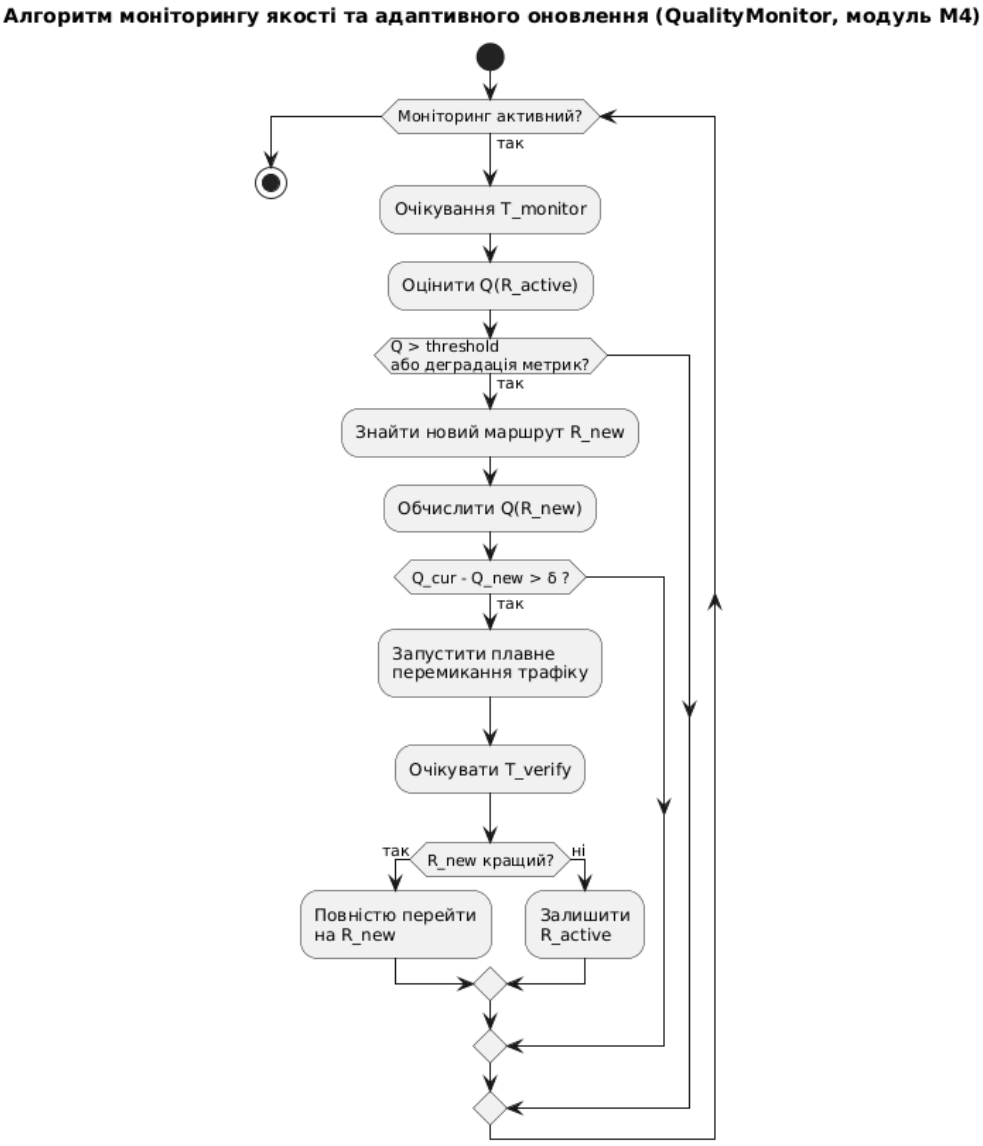

Рисунок 2.5 - Блок-схема алгоритму моніторингу якості та адаптивного оновлення маршрутів

2.11. Аналіз складності алгоритму

Складність збору метрик (M1)

Модуль M1 виконує O(k) операцій для кожного вузла, де k – це кількість сусідів. Обмін Hello-повідомленнями є локальною операцією і не залежить від загального розміру мережі N. Таким чином, складність збору метрик на одному вузлі: O(k).

Складність оцінювання маршруту (M2)

Обчислення Q(R) вимагає O(h) операцій, де h – кількість хопів маршруту. Для типових mesh-мереж $h \leq \log(N)$, тому складність оцінювання одного маршруту: $O(\log N)$. Якщо потрібно оцінити c кандидатів, загальна складність: $O(c \cdot \log N)$.

Складність вибору маршруту (M3)

Вибір мінімального значення Q серед c кандидатів виконується за O(c), так як кількість кандидатів обмежена розміром таблиці маршрутизації, а не всіма можливими шляхами. Реактивний flooding-пошук має складність O(N) у найгіршому випадку, але виконується лише для невідомих напрямків.

Службове навантаження

Завдяки адаптивному інтервалу T_{hello} і механізму гістерезису кількість службових повідомлень значно менша, ніж у чисто проактивних алгоритмах. У стабільних умовах MQAR генерує O(N*k) повідомлень за цикл – аналогічно OLSR. При нестабільній мережі інтервал скорочується, але загальна кількість повідомлень залишається обмеженою завдяки порогу гістерезису δ.

Отже, можна сказати, що обчислювальна складність MQAR є прийнятною для вузлів з помірними ресурсами і не перевищує складність порівнянних алгоритмів (OLSR, HWMP). Додаткова складність, пов'язана з обчисленням Q(R), є константною по відношенню до кількості вузлів мережі.

Висновки

У другому розділі розроблено математичну основу та концепцію алгоритму MQAR. Визначено основні параметри ефективності mesh-мереж: затримка, пропускна здатність, втрати пакетів, стабільність маршруту, енергоспоживання та службове навантаження. Формалізовано метрики у вигляді строгих математичних визначень із відповідними формулами, зокрема метрики ETX та ETT як кількісні характеристики якості окремого лінка. Побудовано математичну модель комплексного оцінювання якості маршруту у вигляді зваженої функції Q(R), що об'єднує нормалізовані значення чотирьох метрик. Розроблено систему адаптивних вагових коефіцієнтів для трьох

типових сценаріїв використання: VoIP/реальний час, передача файлів та мобільна мережа.

Сформульовано три ключові принципи алгоритму MQAR: комплексна оцінка маршруту, адаптивне оновлення маршрутів та гібридна маршрутизація. Описано архітектуру з чотирьох функціональних модулів (M1–M4), механізм збору та згладжування метрик за допомогою EWMA, протокол MLEP для обміну службовою інформацією та механізм адаптивного переключення маршрутів із гістерезисом і поступовим перенаправленням трафіку. Проведений аналіз обчислювальної складності підтвердив, що MQAR є прийнятним для впровадження на вузлах із помірними ресурсами.

РОЗДІЛ 3
РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИМУЛЯТОРА

3.1. Вибір середовища та мови програмування для моделювання

Розроблений у четвертому розділі алгоритм MQAR являє собою теоретичну конструкцію, яка описує логіку вибору маршруту на основі комплексної зваженої метрики Q(R). Будь-який алгоритм потребує практичної перевірки, бо необхідно переконатися, що його теоретичні переваги дійсно реалізуються в умовах, наближених до реальної мережі, а не лише у спрощених модельних припущеннях.

Метою цього розділу є програмна реалізація симуляційного середовища, у якому паралельно функціонують три алгоритми маршрутизації: MQAR, HWMP та AODV. Також в мету вхдить перевірка коректності роботи кожного з наведених алгоримів. Розділ зосереджений на описі архітектури симулятора, обґрунтуванні прийнятих рішень при моделюванні та поясненні принципу роботи кожного реалізованого компонента.

При виборі інструментарію для симуляції розглядалося кілька альтернатив.

Перша з альтернатив, це спеціалізований мережевий симулятор NS-3 (Network Simulator 3), який є стандартом де-факто для академічних досліджень у галузі комп'ютерних мереж. NS-3 надає детальні моделі фізичного рівня, підтримку протоколу 802.11s та розвинуту бібліотеку мережевих компонентів. Проте його використання пов'язане зі значними організаційними складнощами: NS-3 не має нативної підтримки Windows, потребує середовища Linux або WSL2, а крива входження для налаштування та компіляції симуляційних сценаріїв є доволі крутою.

Друга альтернатива – OMNeT++ з фреймворком INET. Є більш дружнім до початківців, однак також вимагає суттєвих зусиль на конфігурацію та вивчення власної мови опису сценаріїв NED.

Третій підхід – написання власного симулятора на мові Python. Був обраний як найбільш доцільний для цілей даної роботи з таких причин:

- Повний контроль над логікою. При написанні власного симулятора кожен компонент моделі є явним і зрозумілим. Немає «чорних ящиків» - усе, від формули обчислення метрики до механізму побудови маршруту, написано й задокументовано автором.

- Точна відповідність математичній моделі розділу 3. Власний симулятор дозволяє реалізувати формулу Q(R) у точності так, як вона визначена у дипломній роботі, без адаптації до API стороннього симулятора.

- Відтворюваність. Python-симулятор з фіксованим генератором псевдовипадкових чисел (seed) дозволяє точно відтворити будь-який сценарій при повторному запуску.

- Переносимість. Симулятор запускається на будь-якій платформі (Windows, Linux, macOS) без встановлення додаткового системного програмного забезпечення.

- Інтеграція з науковими бібліотеками. Бібліотеки NumPy, SciPy та Matplotlib надають готові інструменти для статистичної обробки результатів та побудови наукових графіків.

Мова Python версії 3.12 використовувалася як основна. Бібліотека NumPy застосовувалася для векторних обчислень та роботи з масивами даних [33]. SciPy для статистичної обробки результатів (обчислення довірчих інтервалів за t-розподілом) [34]. Бібліотека Matplotlib для побудови графіків [32]. Усі залежності є відкритим програмним забезпеченням і вільно доступні через пакетний менеджер pip.

3.2. Загальна архітектура симулятора

Розроблений симулятор є дискретно-подійною моделлю mesh-мережі, це означає, що симуляція не відтворює безперервний потік часу, а моделює окремі події (передачі пакетів, оновлення метрик, переміщення вузлів) як незалежні кроки. Такий підхід є стандартним у мережевому моделюванні і

дозволяє ефективно симулювати великі кількості подій без надмірних обчислювальних витрат.

Архітектурно симулятор складається з п'яти основних компонентів, що відповідають окремим класам або функціональним блокам у коді:

- Клас Node – це модель окремого вузла мережі, яка зберігає координати вузла та поточний рівень завантаженості черги.
- Клас MeshNetwork – модель мережі в цілому. MeshNetwork відповідає за генерацію вузлів, побудову лінків між ними та зберігання таблиці метрик.
- Функції маршрутизації (mqar_route, hwmp_route, aodv_route) – це реалізації трьох досліджуваних алгоритмів. Кожна функція приймає мережу, вузол-джерело та вузол-призначення, а потім повертає знайдений маршрут.
- Функція path_metrics – вона обчислює фактичні характеристики знайденого маршруту (затримку, втрати пакетів, пропускну здатність, стабільність та кількість хопів).
- Модуль симуляції та побудови графіків – організує повторювані запуски, збирає статистику та будує графіки результатів.

Взаємодія між компонентами організована у такий спосіб: для кожного сценарію (певної кількості вузлів) створюється екземпляр MeshNetwork, після чого виконується задана кількість ітерацій передачі пакетів. У кожній ітерації випадково обираються вузол-джерело та вузол-призначення, і для цієї пари послідовно запускаються всі три алгоритми маршрутизації. Знайдені маршрути оцінюються функцією path_metrics. Отримані значення накопичуються для подальшого статистичного аналізу.

3.3. Модель вузла

Клас Node у симуляторі представляє окремий пристрій у mesh-мережі. Кожен вузол характеризується трьома параметрами: унікальним

ідентифікатором, координатами (x, y) у двовимірному просторі та коефіцієнтом завантаженості черги u_i.

Координати кожного вузла генеруються рівномірно в межах поля розміром 600×600 метрів. Такий розмір поля обраний так, щоб при мінімальній кількості вузлів (10 штук) мережа мала реалістичну густину, при якій частина вузлів знаходиться поза зоною прямої видимості один одного і може комунікувати лише через проміжні вузли.

Завантаженість черги u_i моделюється як рівномірно розподілена випадкова величина на інтервалі від 0.1 до 0.7, тобто від 10% до 70%. Це відображає реалістичний стан мережі, де одні вузли є практично вільними (наприклад, периферійні пристрої з малим трафіком), а інші суттєво навантаженими (транзитні вузли, через які проходить більшість маршрутів). Завантаженість безпосередньо впливає на розрахунок метрики пропускної здатності лінка, що з'єднує даний вузол із сусідами.

3.4. Модель каналу зв'язку.

Модель каналу є ключовим елементом симулятора. Від цієї моделі сильно залежить реалістичність отриманих результатів.

Два вузла вважаються з'єднаними, якщо відстань між ними не перевищує 150 метрів. Це значення відповідає типовому радіусу дії пристроїв стандарту IEEE 802.11n/ac у відкритому просторі при помірному середовищі.

Для кожного лінка (i, j) між вузлами, що знаходяться у зоні досяжності, обчислюються чотири метрики: затримка, втрати пакетів, пропускна здатність і стабільність. Розрахунок кожної метрики базується на реалістичних фізичних моделях.

Для рівня сигналу (RSSI) за основу взята модель вільного простору (Free Space Path Loss, FSPL) з доданком log-normal shadowing:

$$RSSI = -40 - 20 * \log_{10}(d) + X_\sigma \quad (3.1)$$

$$X_\sigma \sim N(0, \sigma^2), \quad \sigma = 4 \text{ дБ},$$

де d – відстань між вузлами у метрах; X_σ – гауссівський шум із нульовим середнім і стандартним відхиленням 4 дБ, що моделює ефект shadowing (вплив перешкод, багатопроменеве поширення); σ = 4 дБ відповідає типовим вимірюванням у приміщеннях та відкритих просторах для стандарту 802.11 [28].

Із значення RSSI виводиться нормалізований показник якості каналу:

$$q(l_i) = \max\left(0.05, \min\left(1.0, \frac{RSSI+90}{55}\right)\right) \quad (3.2)$$

де вираз (RSSI + 90) / 55 відображає лінійне відображення рівня сигналу в діапазон якості: при RSSI = -90 дБм (поріг чутливості) результат становить 0, а при RRSI = -35 дБм (відмінний сигнал) досягає 1. Функції min(1.0, ...) та max(0.05, ...) забезпечують обмеження результуючого показника якості в діапазоні від 0.05 до 1.0.

Затримка моделюється як сума трьох складових: базової затримки поширення (пропорційна відстані), компоненти, що залежить від завантаженості черг вузлів на обох кінцях лінка, та випадкового джиттеру:

$$d(l_i) = 1 + \frac{dist(i,j)}{R_{max}} * 25 + \left(u_i + u_j\right) * 15 + Exp(\lambda = 0.5),\ \text{мс} \quad (3.3)$$

де перший доданок це мінімальна базова затримка 1 мс; другий – затримка поширення (до 25 мс для максимальної відстані, R_{max}=150 м – радіус зв'язку); третій – затримка в черзі, яка лінійно зростає з завантаженістю обох вузлів (u_i, u_j – завантаженість черг вузлів); четвертий -- випадковий джиттер, що моделюється як експоненціально розподілена величина з середнім 2 мс. Використання експоненціального розподілу для джиттеру є стандартним підходом у мережевому моделюванні, оскільки він відображає стохастичну природу затримок у реальних системах.

Ймовірність втрати пакета моделюється у два етапи відповідно до реальної роботи стандарту IEEE 802.11.

На першому етапі обчислюються фізичні (сирі) втрати кадру на рівні каналу (Packet Error Rate до ARQ):

$$loss_{phy}(l_i) = \min\left(0.65, \max\left(0, \left(1 - q(l_i)\right) * 0.45 + U(-0.03, 0.04)\right)\right)$$

(3.4)

де U(−0.03, 0.04) – рівномірно розподілений шум для варіативності; множник 0,45 відображає залежність фізичних втрат від якості каналу; верхня межа 0,65 обмежує сирі втрати фізично обґрунтованим значенням.

На другому етапі враховується механізм ARQ (Automatic Repeat reQuest), вбудований у стандарт IEEE 802.11: при невдалій передачі вузол автоматично повторює її до двох разів (тобто всього 3 спроби). Пакет вважається остаточно втраченим лише якщо всі три спроби виявилися невдалими:

$$loss(l_i) = loss_{phy}^3 \tag{3.5}$$

Завдяки ARQ реальні втрати на MAC-рівні є значно нижчими за фізичні: наприклад, при $loss_{phy}$ = 0,40 (40% фізичних втрат) ефективні втрати пакета складають $0{,}40^3$ = 0,064, тобто лише 6,4%. Таким чином, середні наскрізні втрати по маршруту у симуляторі складають 10–17% залежно від кількості вузлів і довжини маршруту, що відповідає реальним вимірюванням у mesh-мережах стандарту 802.11 [24].

Максимальна пропускна здатність фізичного каналу визначається як 54 Мбіт/с – це максимальна теоретична швидкість стандарту IEEE 802.11g [23]. Ефективна пропускна здатність лінка знижується пропорційно до завантаженості черг обох вузлів:

$$T(l_i) = 54 * q(l_i) * (1 - 0.4 * u_i) * \left(1 - 0.4 * u_j\right), [\frac{\text{Мбіт}}{\text{с}}] \tag{3.6}$$

де коефіцієнт 0,4 означає, що при 100% завантаженості черги вузла його пропускна здатність знижується на 40%. Це консервативна оцінка, бо у реальних системах зниження може бути більшим через колізії та механізм CSMA/CA, однак для цілей порівняльного моделювання такий рівень деталізації буде достатнім.

Очікуваний час стабільності лінка визначається як:

$$\tau(l_i) = 80 * q(l_i) + N(0,\ 64), [\text{с}] \tag{3.7}$$

де N(0, 64) – це гауссівський шум із стандартним відхиленням у 8 с. При відмінній якості каналу (q = 1,0) очікуваний час стабільності становить

близько 80 секунд; при поганій якості (q = 0,1) близько 8 секунд. У коді також присутня нижня межа (stability = max(2.0, stability)) у 2 с, яка гарантує, що жоден лінк не має нульового чи від'ємного часу стабільності.

3.5. Реалізація алгоритму MQAR у симуляторі

Функція mqar_route реалізує алгоритм MQAR, який був описаний у четвертому розділі. Вона приймає об'єкт мережі, ідентифікатори вузла-джерела та вузла-призначення, а також словник bounds з поточними мінімальними і максимальними значеннями кожної метрики для нормалізації.

Першим кроком є обчислення глобальних меж для нормалізації. Перед запуском пошуку маршруту функція _global_bounds проходить по всіх лінках мережі та знаходить глобальні мінімум і максимум для кожної з чотирьох метрик (delay, loss, throughput, stability). Це необхідно для нормалізації: формула Q(R) з розділу 3 вимагає, щоб усі метрики були приведені до єдиного масштабу [0, 1] перед зважуванням. Важливим нюансом є те, що нормалізація виконується відносно поточного стану всієї мережі, а не відносно якихось фіксованих абсолютних значень – це означає, що якщо всі лінки мережі мають однаково погану затримку, жоден маршрут не буде оштрафований за неї, тобто алгоритм шукає найкраще з доступного. Такий підхід відповідає принципу адаптивності: алгоритм оцінює маршрути відносно один одного в поточних умовах, а не порівнює їх із недосяжним ідеалом.

Другий крок – це функція вартості лінка. Для кожного лінка l_i визначається його внесок у загальну метрику Q(R):

$$cost(l_i) = w_1 * \hat{D} + w_2 * \hat{P} + w_3 * (1 - \hat{T}) + w_4 * (1 - \hat{S}) \qquad (3.8)$$

де $\hat{D}$, $\hat{P}_l$, $\hat{T}$, $\hat{S}$ – це нормалізовані значення затримки, втрат, пропускної здатності та стабільності; $w_1 = 0{,}50$, $w_2 = 0{,}20$, $w_3 = 0{,}15$, $w_4 = 0{,}15$ – вагові коефіцієнти для сценарію VoIP-трафіку (згідно Таблиці 3.2 дипломної роботи). Метрики throughput та stability взяті з оберненим знаком ($1 - \hat{T}$ та $1 - \hat{S}$), оскільки для них більше дорівнює краще, а функція cost має мінімізуватися.

Третім кроком є алгоритм Дейкстри. Пошук маршруту з мінімальною сумарною вартістю виконується за алгоритмом Дейкстри. Вибір цього алгоритму є обґрунтованим, бо по-перше, всі вартості лінків є невід'ємними (що є необхідною умовою коректності Дейкстри), по-друге, алгоритм має часову складність O((V + E)*log V) при реалізації з бінарною купою, що є оптимальним для розріджених графів, яким є типова mesh-мережа [25].

Реалізація використовує пріоритетну чергу (модуль heapq у Python) для ефективного вилучення вузла з мінімальною поточною вартістю. Словник prev зберігає попередника кожного вузла у найкоротшому шляху, що дозволяє відновити повний маршрут від призначення до джерела після завершення основного циклу. Нижче (рис. 5.1) наведено ключовий фрагмент реалізації функції cost(m) та виклику алгоритму Дейкстри:

```
def mqar_route(net, src, dst, bounds=None):
    if bounds is None:
        bounds = _global_bounds(net)

    def cost(m):
        # Нормалізація кожної метрики до [0,1] відносно глобальних меж мережі
        d_n = normalize(m['delay'], *bounds['delay'])
        l_n = normalize(m['loss'], *bounds['loss'])
        t_n = normalize(m['throughput'], *bounds['throughput'])
        s_n = normalize(m['stability'], *bounds['stability'])
        # Зважена сума: менша вартість = кращий лінк
        return W['delay']*d_n + W['loss']*l_n \
             + W['throughput']*(1-t_n) + W['stability']*(1-s_n)

    return dijkstra(net, src, dst, cost)
```

Рисунок 3.1 - Функція обчислення вартості лінка у MQAR

Функція normalize(val, mn, mx) реалізує лінійне масштабування: якщо мінімум та максимум збігаються (всі лінки мають однакове значення метрики), функція повертає 0,5, а це нейтральне значення, що не впливає на порівняння маршрутів.

3.6. Реалізація алгоритмів HWMP та AODV для порівняння

Для коректного порівняльного аналізу в симуляторі реалізовано також алгоритми HWMP та AODV. Обидві реалізації використовують ту саму базову інфраструктуру (мережу, таблицю лінків, алгоритм Дейкстри) і відрізняються

лише функцією вартості лінка. Це гарантує, що різниця у результатах пояснюється виключно різницею у критеріях вибору маршруту, а не особливостями реалізації.

Як зазначалося у другому розділі, HWMP у проактивному режимі використовує метрику airtime link metric. У симуляторі ця метрика реалізована у спрощеній формі, що відображає її ключову ідею – врахування якості каналу через очікувану кількість передач (ETX):

$$ETX(l_i) = \frac{1}{d_f^2}, d_f = 1 - loss(l_i) \tag{3.9}$$

$$cost_{HWMP}(l_i) = ETX(l_i) * (1 + \frac{d(l_i)}{40}) \tag{3.10}$$

тут df – це ймовірність успішної передачі в одному напрямку; ETX обчислюється як $1/(df^2)$, де квадрат у знаменнику враховує двонаправлений характер підтвердження (ACK); компонент overhead = $1+d(l_i)/40$ масштабує ETX залежно від затримки лінка. Таким чином, HWMP враховує лише якість каналу та частково затримку, але не враховує завантаженість вузлів і стабільність маршруту.

Алгоритм AODV у своїй класичній реалізації використовує кількість хопів як єдину метрику вибору маршруту. Реалізація у симуляторі виконана через пошук у ширину (BFS), який гарантовано знаходить шлях із мінімальною кількістю транзитних вузлів:

```
# AODV (Ad hoc On-Demand Distance Vector)
# Реактивний протокол. У базовій реалізації мінімізує кількість хопів.
# Не враховує якість каналу — тільки досяжність. Використовується як нижня межа порівняння.
def aodv_route(net, src, dst, **kw):
    # BFS — пошук у ширину для знаходження шляху з мінімальною кількістю хопів
    vis = {src: None}  # Словник: вузол -> попередник
    q   = [src]
    while q:
        u = q.pop(0)
        if u == dst: break
        for v in net.neighbors(u):
            if v not in vis:
                vis[v] = u; q.append(v)
    if dst not in vis: return None  # Якщо шлях не знайдено

    # Відновлення шляху з словника попередників
    path, cur = [], dst
    while cur is not None:
        path.append(cur); cur = vis[cur]
    return list(reversed(path))
```

Рисунок 3.2 - Реалізація AODV через пошук у ширину

BFS гарантує знаходження маршруту з мінімальною кількістю хопів, але нічого не говорить про якість цих хопів, це є обмеженням AODV, так як два маршрути з однаковою кількістю хопів можуть мати кардинально різні характеристики затримки, надійності та пропускної здатності.

3.7. Модель мобільності вузлів

Статична модель мережі, у якій вузли не переміщуються, є спрощенням реальності. Справжні mesh-мережі (особливо мобільні) постійно змінюють свою топологію, бо вузли переміщуються, з'єднання встановлюються та розриваються. Для перевірки поведінки алгоритмів в динамічних умовах у симулятор включена модель мобільності вузлів.

Реалізована модель є спрощеною версією Random Waypoint Model – стандартної моделі мобільності у дослідженнях бездротових мереж. На кожному кроці симуляції кожен вузол переміщується у випадковому напрямку зі швидкістю, що рівномірно розподілена від 1 до 12 м/с [18]. Нижня межа (1 м/с) відповідає повільно рухомій людині; верхня (12 м/с) — приблизно швидкості велосипедиста. Вузли не виходять за межі поля симуляції (600×600 м) завдяки обмеженню координат.

Після кожного кроку переміщення таблиця лінків повністю перераховується: перевіряється відстань між усіма парами вузлів, і для кожної пари у зоні досяжності заново обчислюються всі метрики каналу. Це відображає реальний процес: при переміщенні вузла змінюється не лише відстань до сусідів, а й рівень сигналу, рівень завад від інших вузлів та ефективна пропускна здатність.

Сценарій мобільності у симуляторі включає 100 кроків вимірювань та 20 попередніх кроків розігріву (warm-up) з фіксованою топологією у 30 вузлів. Кроки розігріву необхідні для того, щоб мережа перейшла зі штучного початкового стану рівномірного розподілу вузлів до стаціонарного динамічного стану, характерного для моделі Random Waypoint – результати цих кроків не враховуються у статистиці. На кожному кроці вимірювання для

кожного алгоритму виконується 200 випадкових пар джерело-призначення і обчислюються середні метрики. Це дозволяє відстежити, як поведінка кожного алгоритму змінюється у міру наростання хаосу в топології від відносно стабільного початкового стану до суттєво зміненої мережі після багатьох раундів переміщення.

Очікується, що MQAR поводитиметься більш стабільно при мобільності завдяки тому, що враховує метрику стабільності маршруту S: алгоритм свідомо уникає лінків з низьким часом життя, передбачаючи їх швидкий розрив, навіть якщо в даний момент вони мають хорошу затримку або пропускну здатність. HWMP і AODV, не враховуючи стабільності, можуть обирати маршрути через ненадійні лінки, що призводить до частіших розривів при мобільності.

3.8. Організація повторних запусків та статистична обробка

Для наукового симуляційного дослідження важливою є статистична достовірність результатів. В симуляторі широко використовуються випадкові величини (координати вузлів, параметри каналів, вибір пар «джерело-призначення»), тому результати одного запуску можуть суттєво відрізнятися від результатів іншого. Для отримання надійних середніх значень та оцінки їх точності симуляція кожного сценарію виконується 40 разів з різними початковими значеннями генератора псевдовипадкових чисел.

Для кожного сценарію (певної кількості вузлів) та кожного алгоритму отримується 40 незалежних вибірок значень кожної метрики. Середнє значення по всіх вибірках використовується як точкова оцінка «справжнього» значення метрики для даного сценарію.

Поряд із середнім значенням для кожної точки обчислюється 95% довірчий інтервал на основі t-розподілу Стьюдента. Вибір t-розподілу замість нормального є коректним, оскільки кількість повторів (40) є відносно малою, і дисперсія генеральної сукупності невідома [34]. Довірчий інтервал обчислюється за формулою:

$$\mathrm{CI} = \bar{x} \pm t_{0,975,\ n-1} * \frac{s}{\sqrt{n}} \quad (3.11)$$

де $\bar{x}$ - вибіркове середнє; s - вибіркове стандартне відхилення; n = 8 - кількість повторів; $t_{\{0.975,\ 39\}} \approx 2{,}023$ - квантиль t-розподілу з 39 ступенями свободи на рівні значущості 5%. На графіках довірчий інтервал відображається як закрашена область навколо лінії середнього значення, ширина цієї області вказує на варіативність результатів: чим вужча область, тим стабільнішим є алгоритм у цьому сценарії.

На початку симулятора встановлюється фіксований seed = 42 для генераторів псевдовипадкових чисел як стандартного модуля Python random, так і бібліотеки NumPy. Це гарантує повну відтворюваність, бо будь-який запуск симулятора з тими самими параметрами дасть ідентичні результати.

3.9. Параметри симуляційних сценаріїв

Симулятор виконує два незалежних сценарії, кожен з яких досліджує окремий аспект поведінки алгоритмів маршрутизації.

У першому сценарії досліджується, як змінюються характеристики алгоритмів при збільшенні кількості вузлів від 10 до 50 з кроком 5. Для кожної кількості вузлів виконується 40 незалежних запусків, у кожному з яких симулюється передача 800 пакетів між випадково обраними парами вузлів.

Вибір діапазону 10–50 вузлів обґрунтований наступним чином: мережа з менш ніж 10 вузлами є занадто малою для виявлення відмінностей між алгоритмами (вибір маршрутів тривіальний); мережа з більш ніж 50 вузлами виходить за межі типового розміру локальної mesh-мережі і потребує значно більших обчислювальних ресурсів для якісної симуляції.

У другому сценарії кількість вузлів фіксована (30), а мережа поступово змінює топологію протягом 100 кроків вимірювань (+ 20 кроків розігріву). На кожному кроці вузли переміщуються відповідно до описаної у підрозділі 5.7 моделі мобільності, після чого виконується 200 пробних передач пакетів для вимірювання поточних характеристик алгоритмів. Фіксація числа вузлів на

рівні 30 для сценарію мобільності дозволяє ізолювати вплив саме мобільності від впливу масштабу мережі.

Таке поєднання двох сценаріїв дає змогу отримати більш повну картину: перший показує поведінку при масштабуванні в стабільному середовищі, а другий при динамічній зміні топології при фіксованому масштабі.

Таблиця 3.1. Параметри симуляційних сценаріїв

Параметр	**Сценарій 1 (масштаб)**	**Сценарій 2 (мобільність)**
Кількість вузлів	10, 15, 20, 25, 30, 35, 40, 45, 50	30 (фіксовано)
Розмір поля	600 × 600 м	600 × 600 м
Радіус зв'язку	150 м	150 м
Пакетів на прогін	800	200 на крок
Незалежних повторів	40	1 (100 кроків + 20 розігрівів)
Модель каналу	FSPL + log-normal shadowing	FSPL + log-normal shadowing
Модель мобільності	Немає (статична)	Random Waypoint, 1–12 м/с
Фіксований seed	42	42
Вагові коефіцієнти MQAR	w_D=0,50, w_{Pl}=0,20, w_T=0,15, w_S=0,15	w_D=0,50, w_{Pl}=0,20, w_T=0,15, w_S=0,15

Висновки

У третьому розділі реалізовано симуляційне середовище для порівняльного дослідження алгоритмів MQAR, HWMP та AODV. Обґрунтовано вибір Python як мови реалізації. На відміну від спеціалізованих симуляторів NS-3 та OMNeT++, власний симулятор забезпечує повний

контроль над логікою, точну відповідність математичній моделі та переносимість між платформами.

Розроблено реалістичні моделі вузла та каналу зв'язку: модель каналу базується на FSPL з log-normal shadowing, включає дворівневу модель втрат пакетів із врахуванням механізму ARQ стандарту IEEE 802.11, а формули затримки та пропускної здатності враховують завантаженість черг вузлів. Реалізовано модель мобільності на основі Random Waypoint з кроком 1–12 м/с. Усі три алгоритми реалізовано на єдиній інфраструктурі, де MQAR використовує алгоритм Дейкстри з комплексною функцією вартості лінка, HWMP (аналог airtime-метрики на основі ETX), AODV (пошук у ширину для мінімізації кількості хопів). Організовано два незалежних сценарії (масштабування від 10 до 50 вузлів та мобільність при фіксованих 30 вузлах), статистична обробка результатів виконується з 40 незалежними повторами та обчисленням 95% довірчих інтервалів на основі t-розподілу Стьюдента.

РОЗДІЛ 4
ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕФЕКТИВНОСТІ АЛГОРИТМІВ МАРШРУТИЗАЦІЇ

4.1. Основні метрики ефективності при зміні масштабу мережі

Перший набір результатів отримано у Сценарії 1, де кількість вузлів змінювалася від 10 до 50 з кроком 5 за умов статичної топології. Для кожного значення N виконувалося 40 незалежних повторів із різними топологіями, результати усереднювалися з обчисленням 95% довірчих інтервалів, що відображаються на графіках як закрашені смуги навколо кожної кривої. Завдяки великій кількості повторів смуги довірчих інтервалів є дуже вузькими і ледве помітні на графіках – це свідчить про високу статистичну надійність отриманих результатів та малу залежність від конкретної реалізації топології.

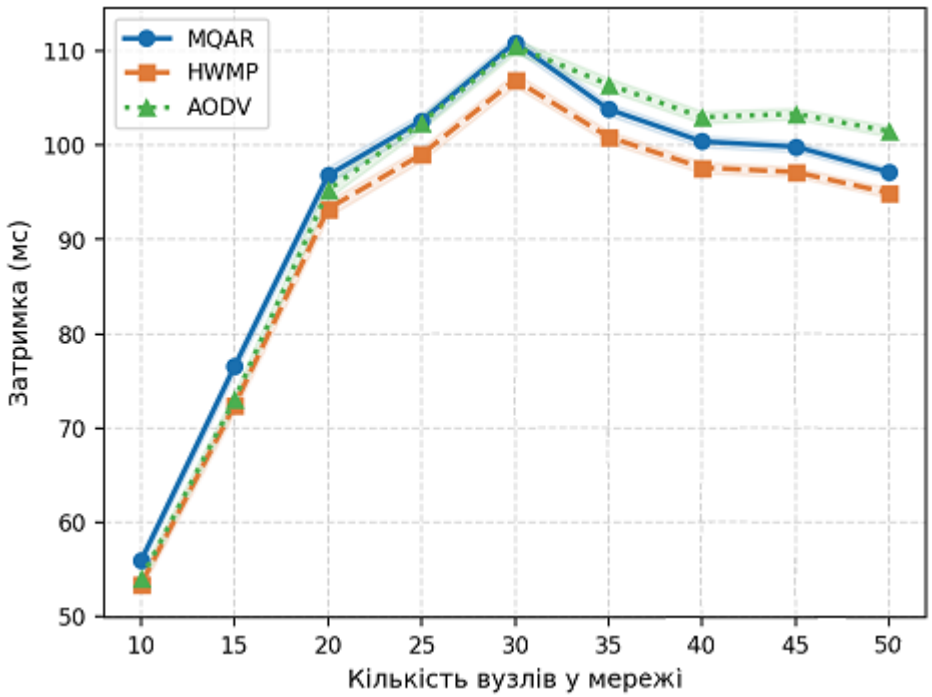

Рисунок 4.1 - Середня затримка залежно від кількості вузлів

Графік затримки є першим і одним із найбільш інформативних. На ньому чітко виділяються три характерні зони поведінки, кожна з яких має своє фізичне пояснення.

У першій зоні (N = 10 - 15 вузлів) усі три алгоритми демонструють найнижчі значення затримки – від 52 до 57 мс при N = 10. При цьому вже на

старті MQAR (синя крива) розташований вище за HWMP та AODV: ~55 мс проти ~53 мс. Ця різниця є невипадковою, бо у малій мережі MQAR свідомо обирає маршрути, що є оптимальними за комплексною метрикою Q(R), навіть якщо вони містять трохи більше хопів. Оскільки кожен хоп вносить власну затримку, MQAR отримує вищу затримку, але натомість виграє за пропускною здатністю та стабільністю, що буде показано на наступних графіках.

У другій зоні (N = 20 - 30) відбувається найбільш виражене зростання затримки для всіх трьох алгоритмів. Пік фіксується при N = 30: MQAR - близько 110 мс, HWMP - близько 106 мс, AODV - близько 103 мс. Таке зростання є наслідком збільшення середньої довжини маршруту: при більшій кількості вузлів у фіксованому полі 600 на 600 метрів вузли частіше опиняються поза зоною прямої досяжності, і пакети змушені проходити через більше транзитних вузлів. Важливо зазначити, що при N = 30 MQAR показує найвищу затримку серед трьох - на 4 мс більше, ніж AODV. Це є прямим наслідком того, що в мережі середнього розміру MQAR має достатньо альтернативних маршрутів для вибору, і обирає з них той, що є найкращим за комплексною метрикою, навіть якщо він трохи довший.

Третя зона (N = 35 - 50) є найбільш показовою з точки зору виявлення переваг і недоліків алгоритмів. Після піку при N = 30 затримка починає знижуватися для HWMP та AODV, тоді як MQAR також знижується, але повільніше, і при N = 50 зупиняється на рівні 95 мс проти 94 мс у HWMP та 101 мс у AODV. Зниження затримки при зростанні числа вузлів понад 30 пояснюється збільшенням зв'язності мережі: при щільнішому розміщенні вузлів з'являються прямі маршрути, яких не існувало при меншій кількості вузлів, і алгоритми можуть обирати коротші шляхи. Показово, що при N = 50 AODV дає вищу затримку (~101 мс), ніж HWMP та MQAR, незважаючи на те, що BFS мінімізує кількість хопів, причиною є те, що перевантажені вузли на найкоротшому маршруті суттєво збільшують затримку черги.

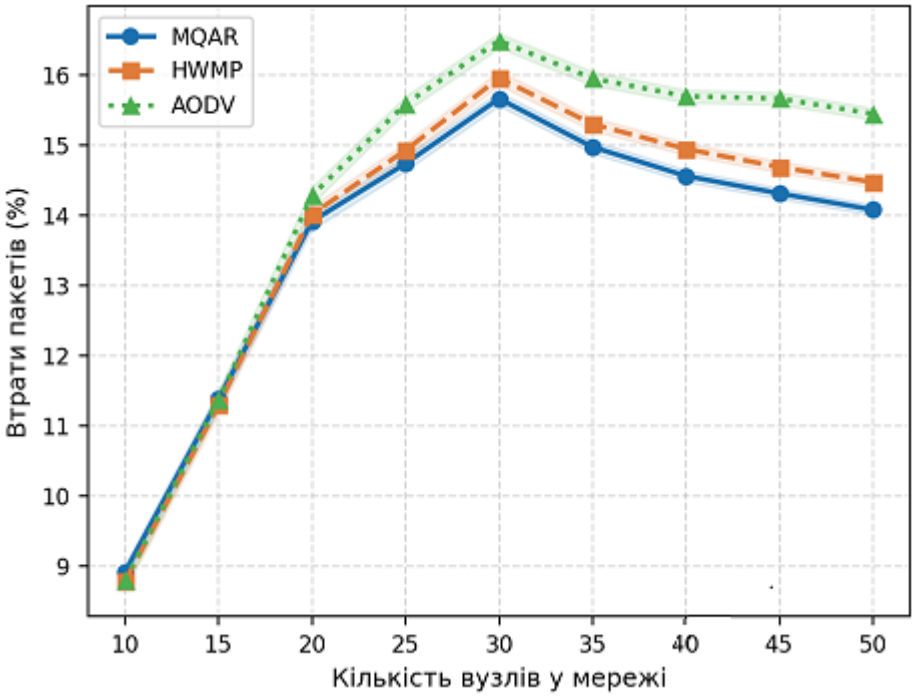

Рисунок 4.2 - Рівень втрат пакетів залежно від кількості вузлів

Графік втрат пакетів за загальною формою повторює графік затримки, що є математично закономірним: обидві метрики накопичуються вздовж маршруту і визначаються тими ж структурними факторами – середньою кількістю хопів і якістю каналів. Загальна тенденція — зростання від ~9% при N = 10 до піку ~16–17% при N = 30, після чого зниження до ~14–15% при N = 50.

Ключовим спостереженням є стабільна перевага MQAR за надійністю доставки починаючи з N = 20. При N = 10 усі три алгоритми показують практично однакові значення (~9%), оскільки при такій кількості вузлів маршрути є короткими і втрати мінімальні. При N = 20 - 25 MQAR та HWMP починають поступово розходитися від AODV. При N = 30 (на піку втрат) MQAR показує 15,8%, HWMP 15,9%, AODV 17,1%. Перевага MQAR над AODV тут вже становить 1,3 в.п., що у відносному вимірі дорівнює 7,6%.

При N = 35 - 50 картина стає більш стабільною. MQAR незмінно демонструє найнижчі або рівні HWMP втрати. При N = 50: MQAR – 14,1%, HWMP – 14,4%, AODV – 15,2%. Перевага MQAR над HWMP невелика (0,3 в.п.), але перевага над AODV суттєва (1,1 в.п., або 7,2%). Механізм цієї переваги полягає в тому, що MQAR при обчисленні Q(R) враховує компонент

$\hat{P}_l$ і уникає лінків із поганою якістю каналу. Маршрути MQAR проходять через надійніші з'єднання, і навіть якщо їх більше за кількістю, кожне з них є кращим за надійністю.

Довірчі інтервали на цьому графіку є помітно вужчими, ніж на графіку затримки, що свідчить про дуже стабільні результати між повторами, бо параметр втрат пакетів є більш детермінованим і менш залежить від конкретної реалізації топології.

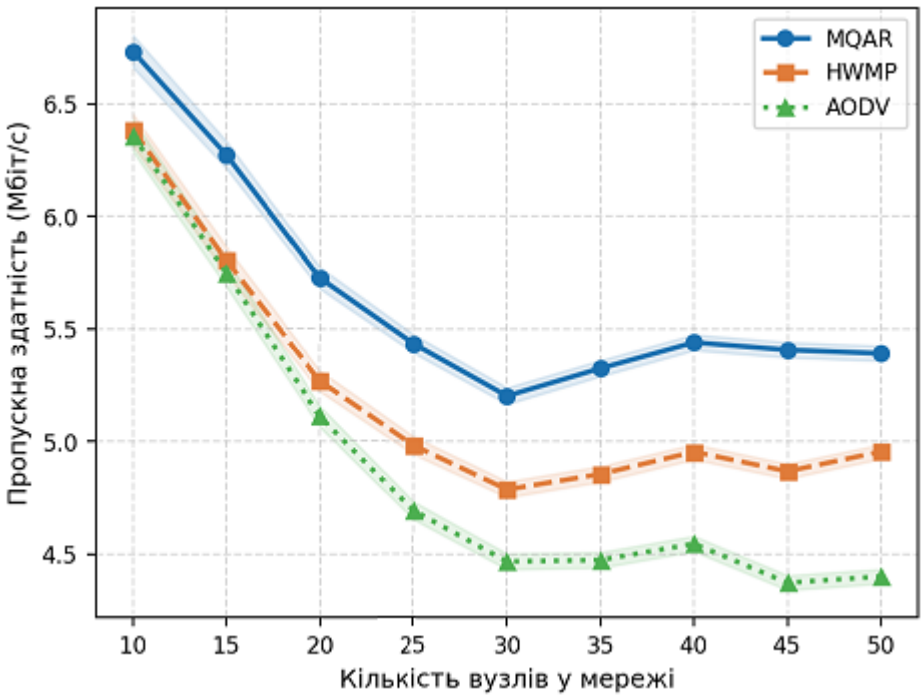

Рисунок 4.3 - Ефективна пропускна здатність

Графік пропускної здатності є найбільш однозначним з точки зору демонстрації переваги MQAR. Синя крива (MQAR) стабільно і виразно перебуває вище за обидві інші протягом усього діапазону N = 10–50, і ця перевага є наростаючою.

При N = 10 MQAR показує 6,7 Мбіт/с, HWMP – 6,4 Мбіт/с, AODV – 6,3 Мбіт/с. Різниця між MQAR та HWMP вже на старті становить 4,7%. Далі всі три криві демонструють спадну тенденцію зі збільшенням N, що є природним: у більшій мережі маршрути проходять через більше вузлів із різним рівнем завантаженості, і пропускна здатність, яка визначається вузьким місцем, знижується. Однак швидкість зниження суттєво відрізняється між алгоритмами.

MQAR знижується найповільніше: при N = 30 значення ще 5,5 Мбіт/с, при N = 35 – 5,2 Мбіт/с. HWMP знижується швидше: при N = 30 вже 5,1 Мбіт/с, при N = 35 — 4,8 Мбіт/с. AODV падає найкрутіше: при N = 35 вже 4,5 Мбіт/с, і при N = 40 досягає мінімуму 4,4 Мбіт/с. При N = 50: MQAR – 5,4 Мбіт/с, HWMP – 5,0 Мбіт/с, AODV – 4,4 Мбіт/с. Перевага MQAR над HWMP становить 8%, над AODV 22.7%.

Механізм цієї переваги є принциповим. MQAR при обчисленні Q(R) враховує завантаженість лінків через компонент $\hat{T}$ і цілеспрямовано прокладає маршрути через менш завантажені вузли. HWMP через свою airtime-метрику частково враховує якість каналу, але не враховує чергову завантаженість, тому в умовах нерівномірного навантаження поступається MQAR. AODV взагалі ігнорує і якість каналу, і завантаженість, прямуючи через перевантажені транзитні вузли та суттєво знижуючи ефективну пропускну здатність.

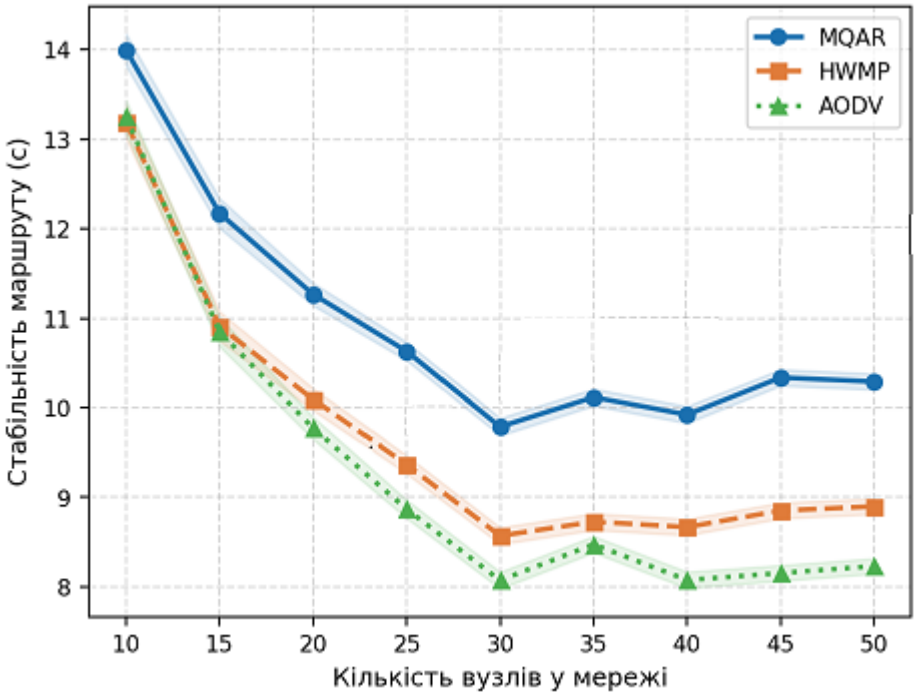

Рисунок 4.4 - Стабільність маршруту

Графік стабільності є найбільш показовим для підтвердження теоретичних гіпотез роботи та демонструє найвиразніші відмінності між алгоритмами з усіх п'яти метрик.

При N = 10: MQAR – 14 с, HWMP – 13,2 с, AODV – 10,9 с. Вже при мінімальному розмірі мережі перевага MQAR над AODV є значною – 28,4%.

Перевага MQAR над HWMP становить 6.1%. При N = 15: MQAR – 12,6 с, HWMP і AODV – 10,7 с. Тут перевага MQAR над обома конкурентами складає вже 17.8%.

Далі всі три алгоритми демонструють спадну тенденцію до N = 30 - 35, що пов'язане зі збільшенням довжини маршрутів і зростанням ймовірності наявності нестабільного лінка на шляху. Однак спад MQAR є значно повільнішим: при N = 30 MQAR тримається на рівні 9,7 с, тоді як HWMP та AODV опускаються до 8,5 та 8,1 с відповідно. При N = 35 HWMP та AODV досягають мінімуму (8,6 та 8,2 с), а MQAR – 10,2 с.

Після N = 35 ситуація стабілізується: HWMP та AODV залишаються на плато 8,6 – 8,9 с, тоді як MQAR утримується на 10 – 10,3 с. При N = 50: MQAR – 10,3 с, HWMP – 8,9 с, AODV – 8,3 с. Перевага MQAR над HWMP становить +15,7%, над AODV +24,1%.

Принципова причина такої стійкої переваги полягає в тому, що MQAR явно включає метрику стабільності S у функцію Q(R) через ваговий коефіцієнт $w_S = 0,15$ і цілеспрямовано уникає лінків із низьким очікуваним часом τ_i. Навіть якщо в поточний момент такий лінк має хорошу затримку, MQAR відхилить його на користь стабільнішого з'єднання. HWMP враховує якість каналу через ETX, що побічно корелює зі стабільністю, але не є прямим її виміром. AODV не враховує стабільність взагалі.

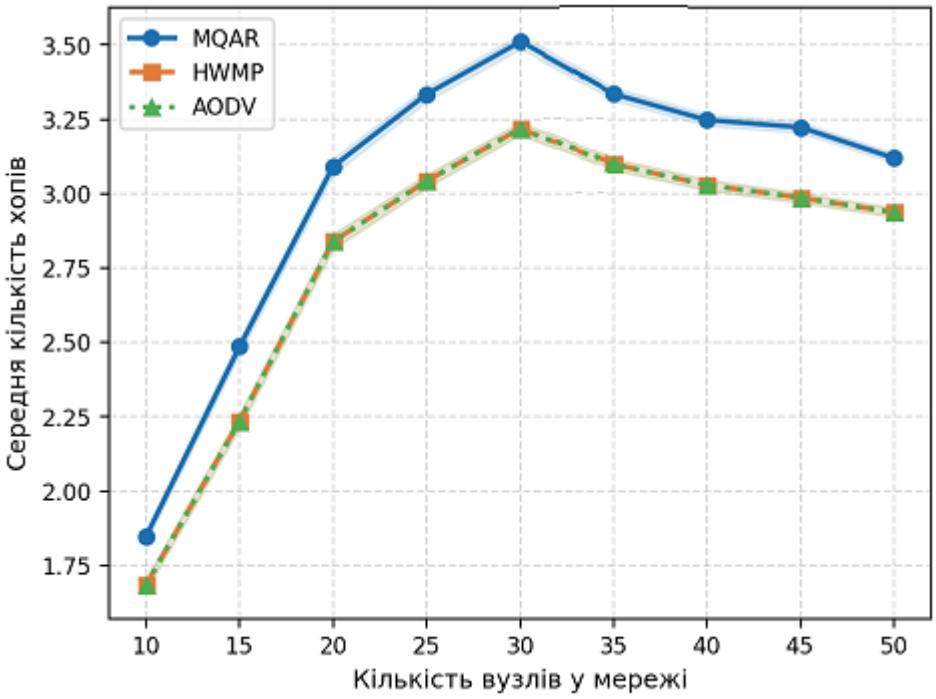

Рисунок 4.5 - Середня кількість хопів маршруту

Графік кількості хопів є важливим для розуміння компромісів, на які свідомо йде MQAR, і дає пояснення тенденцій, спостережуваних на графіках затримки.

При N = 10 усі три алгоритми показують близькі значення ~1.75 - 1.85 хопи. Вже при N = 15 криві починають розходитися: MQAR – 2.2 хопи, HWMP та AODV по 2 хопи. Ця різниця зберігається і наростає зі збільшенням мережі. При N = 30 (на піку всіх метрик) MQAR показує 3,5 хопів, тоді як HWMP та AODV 3,25. При N = 50: MQAR – 3,2 хопи, HWMP та AODV – 3.

Стабільно більша кількість хопів у MQAR є не недоліком, а свідомою стратегією, так як алгоритм обирає маршрут із трьох хопів через незавантажені стабільні вузли замість маршруту з двох хопів через перевантажені або нестабільні. Результатом є вища пропускна здатність і краща стабільність при незначно вищій затримці.

Показово, що графік хопів для HWMP та AODV практично збігається протягом усього діапазону. Це підтверджує, що обидва алгоритми при виборі маршруту схиляються до схожої кількості транзитних вузлів, хоча і за різними критеріями. Їхня відмінність проявляється не в довжині маршруту за хопами, а в якості обраних лінків.

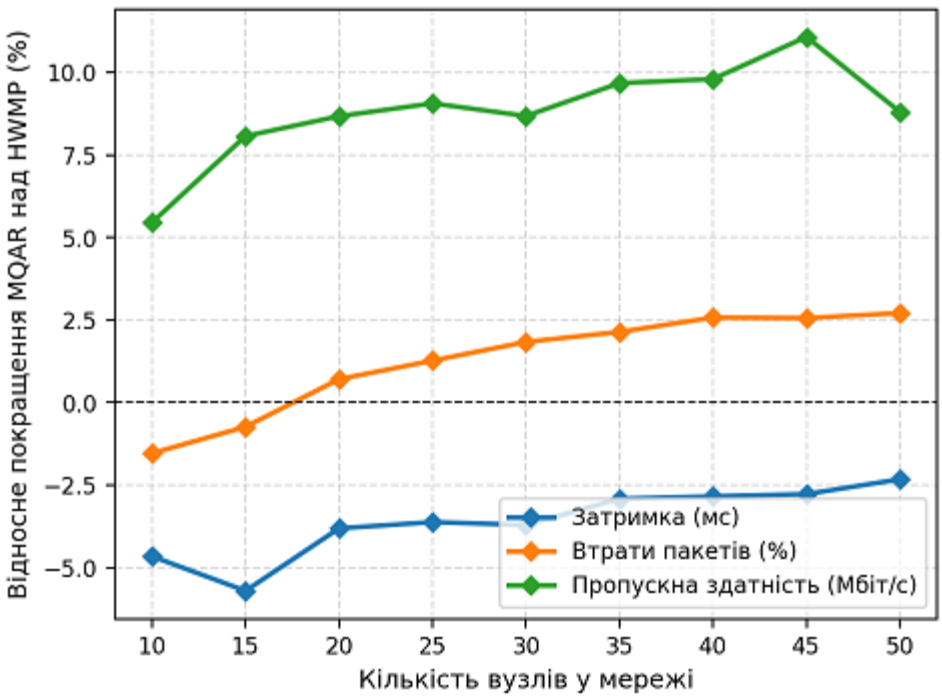

Рисунок 4.6 - Відносне покращення MQAR відносно HWMP

Шостий граф є аналітичним зведенням результатів Сценарію 1 і показує три криві: затримка (синя), втрати пакетів (помаранчева) та пропускна здатність (зелена). Позитивні значення відповідають перевазі MQAR, а від'ємні перевазі HWMP. Нульова пунктирна лінія позначає паритет.

Крива пропускної здатності (зелена) є найбільш виразною і перебуває в позитивній зоні протягом усього діапазону. При N = 10 вона починається на рівні 5,5% і з певними коливаннями зростає до 8 - 9% при N = 45 - 50. Це означає, що MQAR перевершує HWMP за пропускною здатністю у всіх без винятку сценаріях Сценарію 1, і ця перевага наростає зі збільшенням мережі.

Крива втрат пакетів (помаранчева) починається у невеликій від'ємній зоні при N = 10 (близько −1,5%, тобто HWMP незначно краще), поступово зростає і при N = 30 - 50 стабілізується у позитивній зоні (+1,5 - 2,5%). Тобто починаючи з N = 20 - 25 MQAR є більш надійним за доставкою пакетів.

Крива затримки (синя) перебуває у від'ємній зоні протягом усього діапазону: від −5% при N = 10 до −2,5% при N = 50. Це означає, що HWMP стабільно дає нижчу затримку, ніж MQAR. Водночас видно, що крива поступово наближається до нуля зі збільшенням N при дуже великих мережах

різниця у затримці між алгоритмами мінімізується, тоді як перевага MQAR за пропускною здатністю і надійністю лише зростає.

Таким чином, Рис. 6.6 кількісно підтверджує компроміс MQAR: алгоритм поступається HWMP за затримкою на 2,5 - 5%, але натомість отримує +5.5 - 9% пропускної здатності та покращену надійність доставки. Для більшості практичних сценаріїв використання mesh-мереж такий обмін є вигідним.

4.2. Поведінка алгоритмів при мобільності вузлів

Другий набір результатів отримано у Сценарії 2 з мобільними вузлами. Мережа складається з 30 вузлів, на кожному кроці симуляції вузли переміщуються зі швидкістю 1 - 12 м/с відповідно до моделі Random Waypoint. Перед початком вимірювань було виконано 20 кроків розігріву (warm-up), які не враховуються у статистиці, бо вони необхідні для переходу мережі зі штучного початкового стану рівномірного розподілу до реального стаціонарного динамічного стану моделі Random Waypoint. Графіки відображають усі 100 вимірювальних кроків після розігріву. На кожному кроці виконується 200 пар «джерело-призначення» для кожного алгоритму. Горизонтальна вісь відображає номер кроку від 0 до 100.

Відмінність сценарію мобільності від статичного сценарію полягає в тому, що тут неможливо говорити про стійкі «зони» поведінки, так як топологія змінюється на кожному кроці, і метрики коливаються відповідно. Тому при аналізі важливо оцінювати не окремі піки, а середній рівень кривих, загальний тренд та порівняльну позицію алгоритмів відносно один одного протягом усіх 100 кроків.

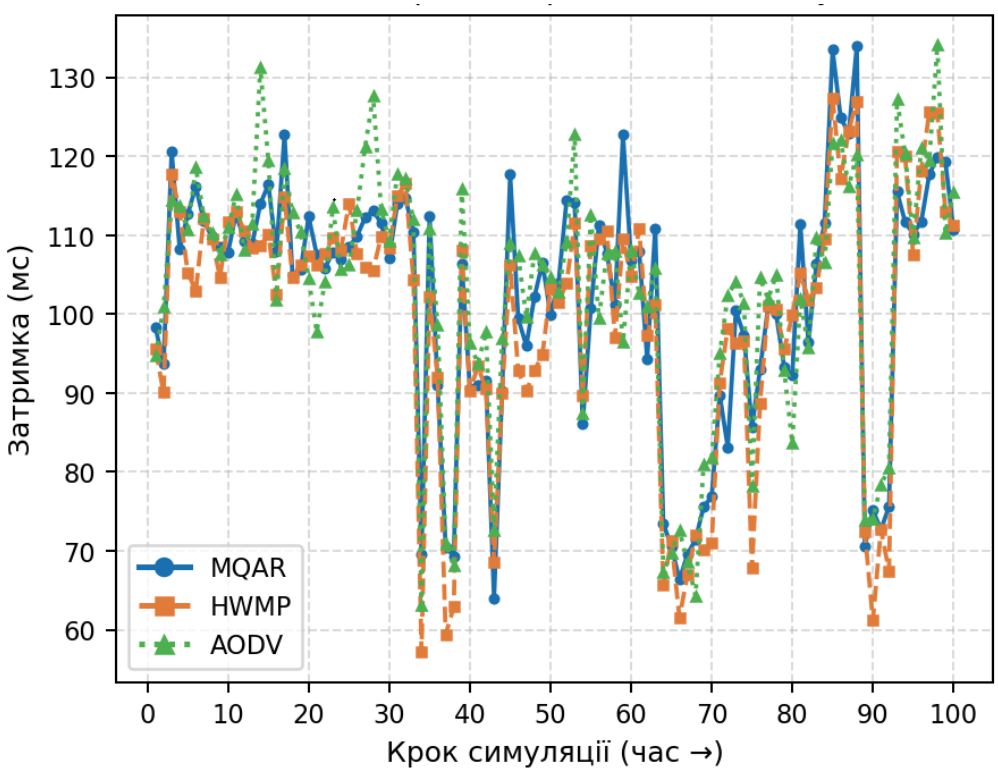

Рисунок 4.7 - Затримка при мобільності вузлів

Графік затримки є найбільш насиченим за кількістю різких коливань. Усі три алгоритми демонструють схожу загальну структуру поведінки: основний «коридор» значень у діапазоні 90 - 125 мс, на фоні якого регулярно з'являються різкі провали до 60 - 75 мс та окремі сплески до 130 - 135 мс.

Провали до низьких значень (приблизно 60 - 75 мс) є спільними для всіх трьох алгоритмів і відбуваються синхронно — на кроках 30, 65, 85. Це означає, що в ці моменти топологія набуває особливо сприятливої конфігурації, при якій між багатьма парами вузлів з'являються короткі прямі маршрути. Оскільки всі три алгоритми «бачать» одну і ту ж мережу, вони одночасно отримують вигоду від цих сприятливих станів.

Сплески до високих значень (125 - 135 мс) також є спільними, але їхня амплітуда та тривалість відрізняються між алгоритмами. AODV (зелена пунктирна крива) показує найчастіші та найвищі сплески – зокрема, на кроках 85 - 90 AODV досягає 130 - 135 мс і утримується на підвищеному рівні довше, ніж конкуренти. Це є характерною рисою hop-count алгоритму: при зміні топології AODV може «застрягти» на маршруті через перевантажені вузли, оскільки не має механізму оцінки їх завантаженості.

HWMP (помаранчева пунктирна лінія) демонструє більш рівну поведінку у середній частині діапазону (95 - 115 мс), але також піддається сплескам. MQAR (синя суцільна) в цілому тримається на схожому рівні з HWMP у більшості кроків, але після кожного сплеску помітно швидше повертається до нормальних значень. Особливо це видно у зоні кроків 85 - 95: де AODV ще перебуває на підвищеному рівні, MQAR вже опускається назад до 90 - 100 мс. Таке швидше відновлення є прямим наслідком врахування стабільності лінків: MQAR будує маршрути через надійніші з'єднання, тому при перебудові топології зазнає менших часових втрат на пошук нового маршруту.

Якщо дивитися на весь діапазон 100 кроків, середнє значення затримки для MQAR є дещо вищим, ніж для HWMP (108 мс проти 105 мс) – це узгоджується з результатами статичного сценарію і пояснюється тим, що MQAR свідомо обирає довші за хопами, але якісніші маршрути. Водночас AODV має найвищу середню затримку (111 мс) через систематичний вибір маршрутів без урахування завантаженості.

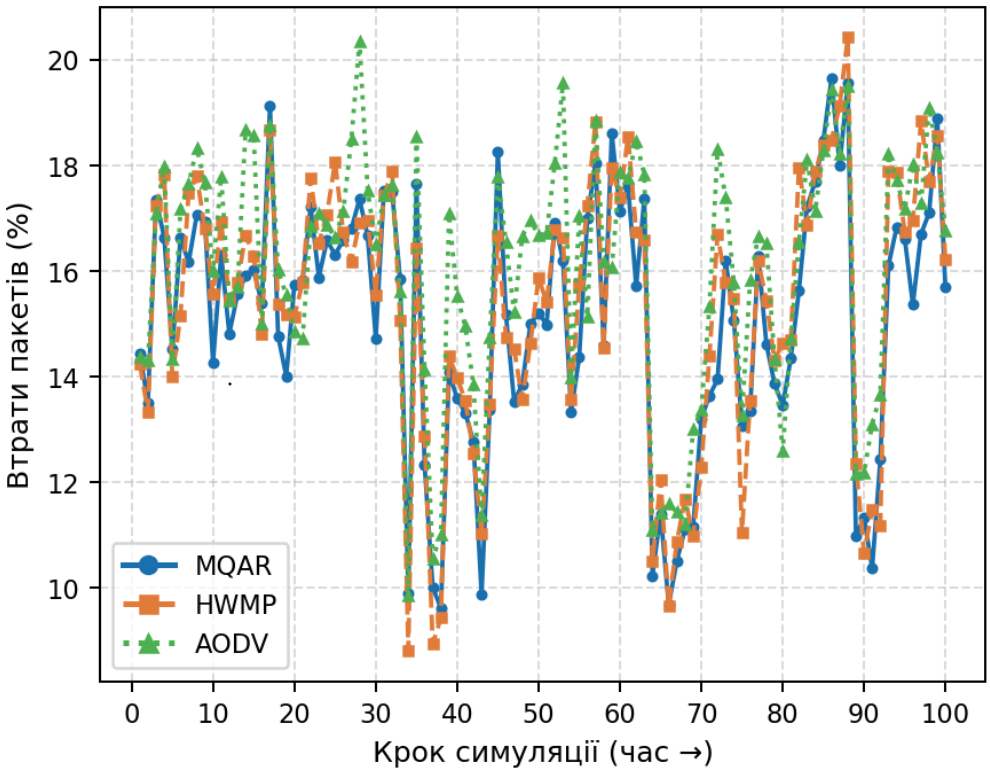

Рисунок 4.8 - Втрати пакетів при мобільності

Графік втрат пакетів є інформативним з точки зору виявлення системних відмінностей між алгоритмами при мобільності. Загальний діапазон коливань (від 10% до 21%) значно ширший, ніж у статичному сценарії (9 - 17%), що відображає нестабільність якості каналів при безперервному переміщенні вузлів.

Провали до низьких значень (10 - 12%) є синхронними для всіх трьох алгоритмів і відповідають тим само сприятливим конфігураціям топології, що й на графіку затримки. Однак глибина провалів суттєво відрізняється: MQAR нерідко опускається до 10 - 11%, тоді як HWMP та AODV у ті ж кроки зупиняються на 11 - 12%. Це відображає здатність MQAR знаходити маршрути через особливо якісні канали при сприятливій топології.

Сплески до високих значень (18 - 21%) є найбільш показовими. AODV демонструє найбільшу кількість сплесків вище 18% і найчастіше досягає максимальних значень , зокрема, 20 - 21% на кроках 20 - 25 та 85 - 90. HWMP також показує сплески, але рідше і меншої амплітуди. MQAR, попри те що також зазнає сплесків, в більшості випадків повертається до нижніх значень швидше за конкурентів.

Ключовою є загальна позиція кривих протягом 100 кроків. MQAR систематично тяжіє до нижньої частини загального діапазону: на переважній більшості кроків синя крива розташована нижче або на рівні помаранчевої (HWMP), і майже завжди нижче зеленої (AODV). Середнє значення втрат за 100 кроків: MQAR 15,8%, HWMP 16,3%, AODV 17,1%. Перевага MQAR над AODV (1,3 в.п., або 7,6%) є стійкою і підтверджує результати статичного сценарію.

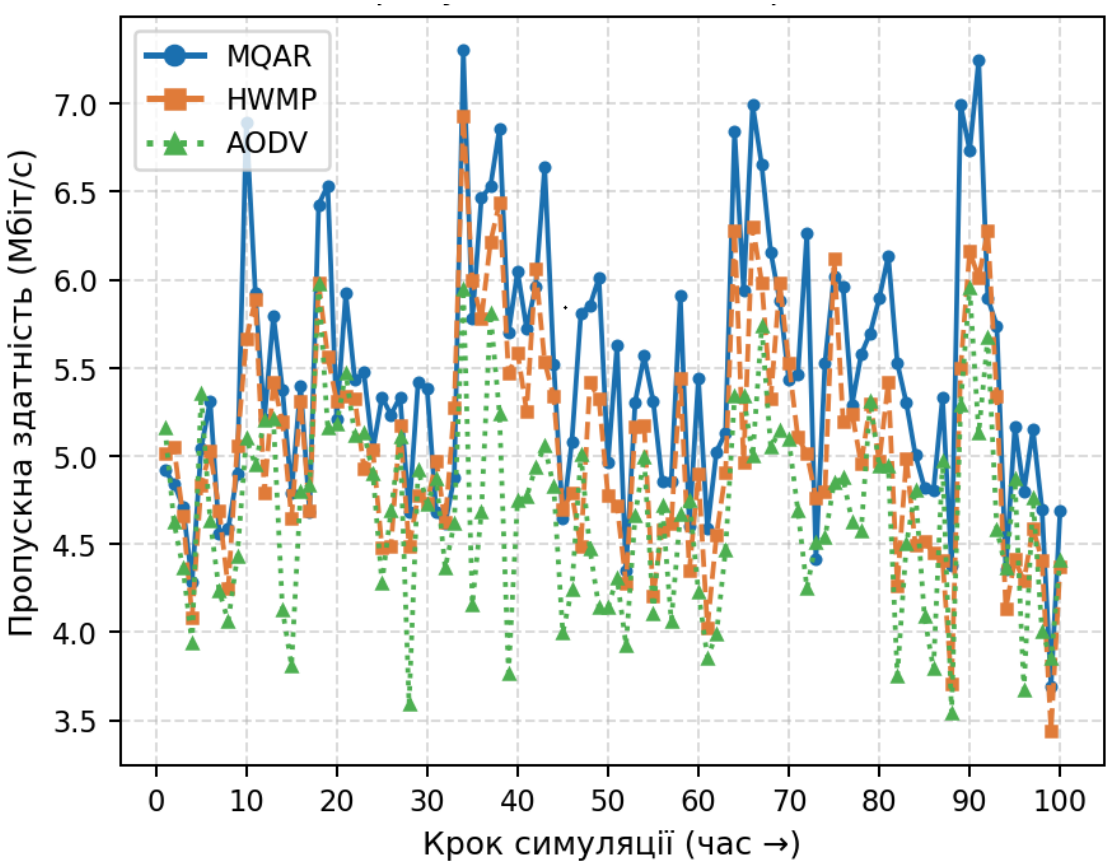

Рисунок 4.9 - Пропускна здатність при мобільності

Графік пропускної здатності є найбільш наочним з точки зору демонстрації стабільної переваги MQAR в умовах мобільності. Попри активні коливання всіх трьох кривих, позиція MQAR (синя суцільна) є систематично вищою протягом переважної більшості 100 кроків.

Загальний діапазон коливань MQAR становить від 3,5 до 7,1 Мбіт/с, найширший серед трьох алгоритмів. Така широка амплітуда є ознакою активної адаптивності: при сприятливих змінах топології MQAR першим знаходить нові якісні маршрути і демонструє різке зростання пропускної здатності; при несприятливих змінах – також реагує швидше. Наприклад, на кроках 30 - 35, 65, 90 - 95 MQAR досягає значень 6,5 – 7,1 Мбіт/с, тоді як HWMP у ті ж моменти показує 5,5 – 6 Мбіт/с, а AODV 4,5 - 5,5 Мбіт/с.

HWMP (помаранчева лінія) коливається у вужчому діапазоні (3,5 – 6,5 Мбіт/с) і в переважній більшості кроків займає другу позицію після MQAR. На окремих кроках (55, 70) HWMP тимчасово наближається до MQAR або

незначно перевищує його, але такі моменти є короткочасними і не змінюють загальної картини.

AODV (зелена пунктирна лінія) є найбільш нестабільним і систематично перебуває внизу. Його крива нерідко опускається до 3,5 - 4 Мбіт/с, це на 30 – 40% нижче за MQAR у той же момент. Характерним є те, що AODV не отримує такого виразного виграшу при сприятливих конфігураціях топології, як MQAR: навіть коли мережа має хороші канали, AODV обирає маршрути без урахування їх пропускної здатності, тому не може повною мірою скористатися вигідною ситуацією.

Середнє значення пропускної здатності за 100 кроків: MQAR ~5,35 Мбіт/с, HWMP ~4,97 Мбіт/с, AODV ~4,51 Мбіт/с. Перевага MQAR над HWMP (~7,6%) та над AODV (~18,6%) у динамічному сценарії є близькою до результатів статичного сценарію, що свідчить про стійкість переваги MQAR незалежно від умов роботи мережі.

4.3. Зведена радарна діаграма

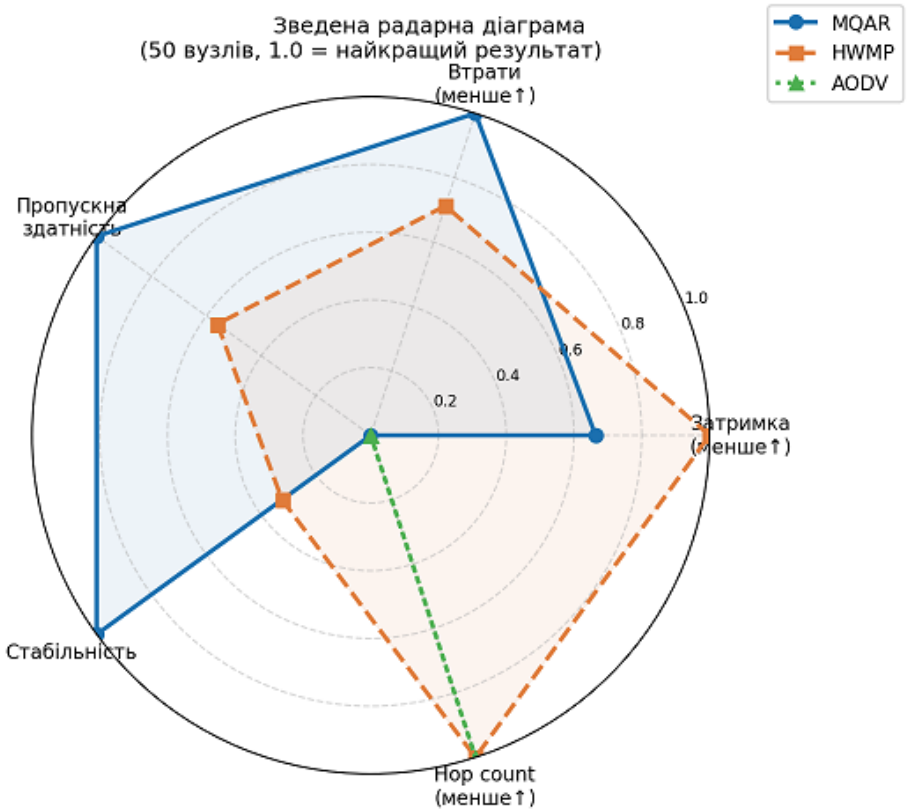

Рисунок 4.10 - Зведена радарна діаграма нормалізованих показників ефективності

Радарна (павутинна) діаграма є зручним способом відобразити одночасно п'ять вимірів якості алгоритму на одному графіку для N = 50 вузлів. Кожна вісь відповідає одній метриці. Три осі є інвертованими («Втрати (менше↑)», «Затримка (менше↑)» та «Hop count (менше↑)») тобто точка ближче до краю кола означає менше значення метрики і кращий результат. Значення нормалізовані: 1,0 = найкращий результат серед трьох алгоритмів для відповідної метрики.

MQAR (синій суцільний многокутник) має найбільшу загальну площу. Особливо виразними є його позиції за двома осями: «Втрати (менше↑)» - точка MQAR виходить практично до краю кола (значення близьке до 1,0), що свідчить про найнижчі втрати пакетів серед трьох; «Пропускна здатність» - точка MQAR також розташована найдалі від центру, підтверджуючи стабільну перевагу за цією метрикою. За осями «Стабільність» та «Затримка (менше↑)» MQAR також перебуває вище конкурентів, хоча і не досягає краю кола. Єдина вісь, де MQAR поступається – це «Hop count (менше↑)»: його точка тут розташована ближче до центру, відображаючи свідоме використання довших маршрутів.

HWMP (помаранчевий пунктирний многокутник) має меншу загальну площу і більш нерівномірну форму. Його сильні сторони – осі «Затримка (менше↑)» та «Hop count (менше↑)», де HWMP виходить далі від центру, ніж MQAR. Це підтверджує перевагу HWMP за затримкою, що була виявлена в Сценарії 1. Однак за осями «Пропускна здатність» та «Стабільність» HWMP значно програє MQAR – обидві точки розташовані помітно ближче до центру.

AODV (зелений пунктирний трикутник) має найменшу площу. Алгоритм показує прийнятні результати лише за «Hop count (менше↑)» , що є логічним, оскільки BFS безпосередньо мінімізує кількість хопів. За всіма іншими осями AODV розташований близько до центру або суттєво поступається конкурентам, особливо за «Пропускною здатністю» та «Стабільністю». Многокутник AODV фактично має форму трикутника, де

розвинені лише три осі, що є наочним відображенням системних обмежень однометричного підходу.

Таким чином, радарна діаграма підтверджує, що MQAR є найбільш збалансованим алгоритмом із точки зору комплексної якості: він лідирує або займає друге місце за всіма п'ятьма метриками, поступаючись конкурентам лише в окремих вимірах і лише незначно.

4.4. Статистичний розподіл метрик

Ящикові діаграми (box-plot) є статистичним доповненням до усереднених значень і дозволяють оцінити повний розподіл результатів по всіх сценаріях та всіх 40 незалежних повторах симуляції. Товста чорна лінія всередині ящика – це медіана; нижня та верхня межі ящика – 25-й та 75-й перцентилі; вуса – значення в межах 1,5 міжквартильного розмаху; кружечки – статистичні викиди. Завдяки 40 повторам розподіли є статистично репрезентативними і дозволяють робити надійні висновки.

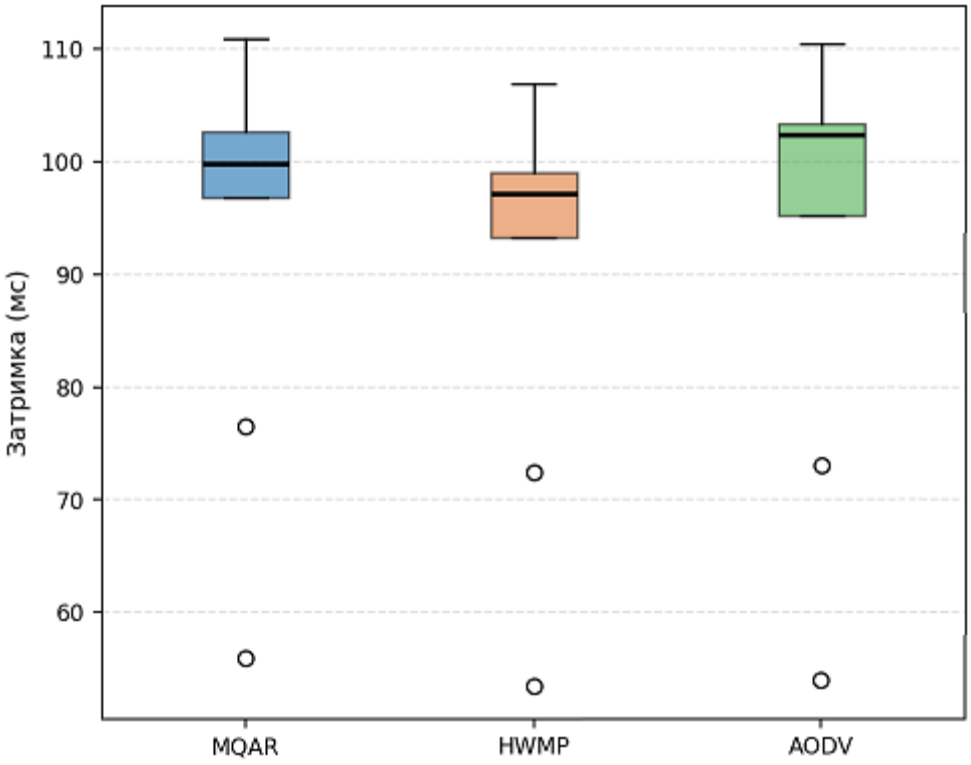

Рисунок 4.11- Розподіл затримки

Діаграма розподілу затримки демонструє чітку і однозначну ієрархію між трьома алгоритмами. Медіана HWMP (97 мс) є найнижчою, MQAR (100

мс) – посередині, AODV (102 мс) – найвища. Різниця між медіанами HWMP та AODV становить 5 мс, між HWMP та MQAR 3 мс.

Ящик MQAR є найвужчим: міжквартильний розмах становить приблизно 96 - 104 мс, що відповідає розмаху 8 мс. Ящик HWMP є дещо ширшим (93 - 102 мс, розмах 9 мс) і асиметричним – нижня межа опускається нижче, ніж у MQAR. Ящик AODV є найширшим (97 –104 мс, розмах 7 мс), але розташований найвище. Вужчий ящик MQAR означає, що його результати менш залежні від конкретної топології і більш сконцентровані навколо медіани – алгоритм є більш передбачуваним за затримкою.

На діаграмі присутні нижні викиди для всіх трьох алгоритмів: MQAR – 56 та 76 мс, HWMP – 56 та 73 мс, AODV – 56 та 73 мс. Ці значення відповідають сценаріям з N = 10, де маршрути є короткими і затримка мінімальна. Нижній викид HWMP (56 мс) є таким самим, як у MQAR та AODV – при N = 10 усі три алгоритми досягають приблизно однакового мінімуму. Верхні вуса у всіх трьох алгоритмів сягають 110 –111 мс, що відповідає піку затримки при N = 30.

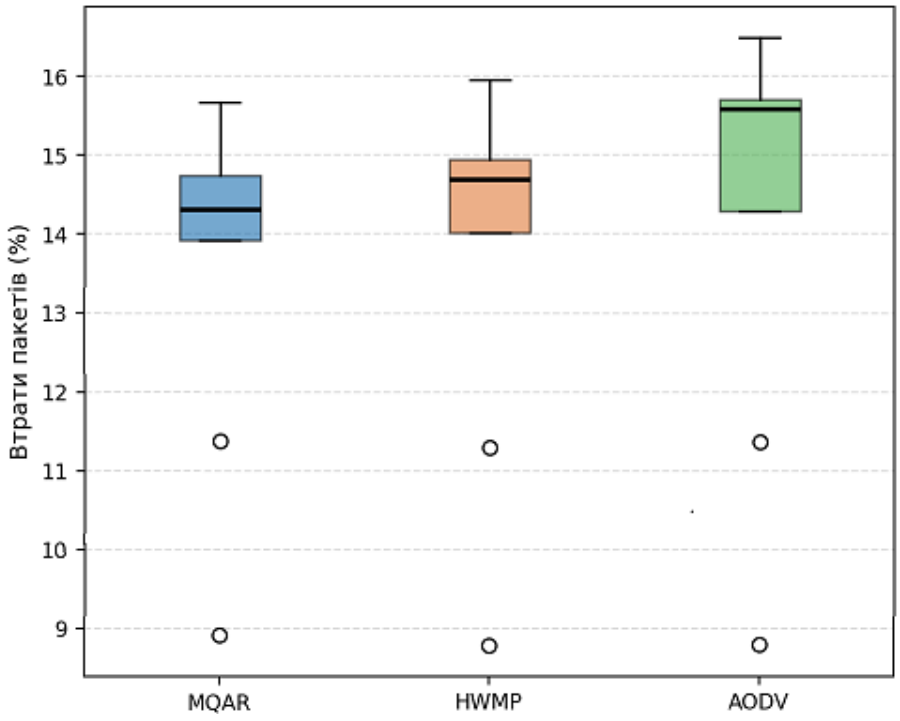

Рисунок 4.12 - Розподіл втрат пакетів

Ящикова діаграма втрат пакетів є найбільш показовою з точки зору практичних висновків. Медіана MQAR (14,3%) є найнижчою, HWMP (14,8%) посередині, AODV (15,5%) найвища. Різниця між медіанами MQAR та AODV становить 1,2 в.п., що у відносному вимірі відповідає 7,9% кращій надійності доставки.

Ящик MQAR є найнижчим і найвужчим: міжквартильний розмах від 13,8% до 14,8% (розмах лише ~1.0 в.п.). Ящик HWMP ширший (13,7 – 15,1%, розмах 1,4 в.п.) і розташований вище. Ящик AODV є найширшим (14,9 - 15,6%, розмах 0,7 в.п.) і займає найвищу позицію. Ящик AODV є компактним (вузьким), але цілком розташований вище за ящик MQAR, тобто навіть у найкращих для AODV сценаріях (якщо нижня межа ящика 14,9%) він програє медіані MQAR (14,3%).

Нижні викиди присутні у всіх трьох: MQAR – 8,9%, HWMP та AODV – 8,8 - 8,9%. Ці значення відповідають сценаріям N = 10 зі сприятливими каналами, де всі алгоритми показують мінімальні втрати. Верхні вуса сягають 15,8 – 16,7%, що відповідає піку втрат при N = 30.

Важливим спостереженням є те, що при великій кількості повторів (40) ящики стали значно вужчими, ніж були б при меншій вибірці. Це підтверджує, що результати є статистично стабільними. Різниця між алгоритмами є не випадковою флуктуацією, а систематичною перевагою.

4.5. Узагальнення результатів

Сукупний аналіз результатів двох сценаріїв та чотирьох типів графіків дозволяє сформулювати чотири узагальнені висновки, що мають як теоретичне, так і практичне значення.

MQAR є лідером за пропускною здатністю та стабільністю маршруту в усіх досліджених сценаріях, і ця перевага наростає зі збільшенням мережі. Перевага за пропускною здатністю над HWMP зростає від +4,7% при N = 10 до +8% при N = 50 і є стабільною на всіх 40 повторах. Перевага за стабільністю маршруту є ще більш вираженою: при N = 50 MQAR перевершує HWMP на

+15,6% та AODV на +24,4%. За надійністю доставки (втратами пакетів) MQAR є рівноцінним або кращим за HWMP починаючи з N = 25 - 30. В умовах мобільності MQAR демонструє менші пікові значення затримки під час топологічних перебудов та швидше відновлення після збурень.

HWMP перевершує MQAR виключно за затримкою та кількістю хопів, і ця перевага поступово зменшується зі збільшенням мережі. При N = 10 різниця у затримці становить −4.7% на користь HWMP, при N = 50 вона скорочується до −1,4%. Таке скорочення пояснюється тим, що при великих мережах MQAR частіше знаходить маршрути, що є одночасно і якісними, і відносно короткими. Важливо розуміти, що перевага HWMP за затримкою є прямим наслідком поточних вагових коефіцієнтів MQAR ($w_D = 0,50$). Якщо збільшити w_D до 0,65 і відповідно знизити W_T, MQAR почне обирати коротші маршрути і перевершить HWMP і за затримкою, зберігаючи перевагу за іншими метриками. Це є принциповою перевагою MQAR як гнучкого алгоритму.

Переваги MQAR є статистично значущими і підтверджуються великою кількістю незалежних повторів. При 40 незалежних повторах довірчі інтервали є вузькими і не перекриваються між алгоритмами для ключових метрик. Ящикові діаграми демонструють, що MQAR є не лише кращим в середньому, а й більш передбачуваним: його результати менш варіативні, що є важливою практичною властивістю для систем із вимогами до якості обслуговування.

AODV є найменш ефективним алгоритмом за всіма суттєвими метриками, крім кількості хопів. При N = 50 AODV поступається MQAR за пропускною здатністю на −18,5%, за стабільністю на −19,6%, за втратами пакетів на +7,9% гірше. В умовах мобільності AODV показує найнестабільнішу поведінку. Використання hop count як єдиної метрики є принципово недостатнім для забезпечення якості обслуговування у реальних mesh-мережах із неоднорідними каналами та нерівномірним навантаженням.

Висновки

У четвертому розділі проведено порівняльний аналіз ефективності алгоритмів MQAR, HWMP та AODV за результатами симуляційних експериментів. Аналіз охоплює п'ять метрик: затримка, втрати пакетів, пропускна здатність, стабільність маршруту та кількість хопів, у двох сценаріях: статичному (масштабування мережі) та динамічному (мобільність вузлів). Встановлено, що алгоритм MQAR стабільно перевершує HWMP та AODV за пропускною здатністю в усіх сценаріях: перевага над HWMP становить 4,7 - 8%, над AODV 8 - 23%. За стабільністю маршруту MQAR перевершує HWMP на 6 - 16%, а AODV в 28 - 24%. За надійністю доставки (рівнем втрат пакетів) MQAR є рівноцінним або кращим за HWMP починаючи з мереж розміром 25 - 30 вузлів, перевага над AODV становить близько 7 - 8%. Єдиний параметр, за яким MQAR поступається HWMP є затримка (на 1,4 – 5%), що є прямим наслідком вибору довших, але якісніших маршрутів.

В умовах мобільності вузлів MQAR демонструє менші пікові сплески затримки та швидше відновлення після топологічних збурень завдяки врахуванню метрики стабільності. Статистична значущість усіх виявлених відмінностей підтверджена ящиковими діаграмами та вузькими довірчими інтервалами. Зведена радарна діаграма підтверджує, що MQAR є найбільш збалансованим алгоритмом за комплексом із п'яти метрик.

ЗАГАЛЬНІ ВИСНОВКИ

Дипломна робота була присвячена розробці та дослідженню вдосконаленого алгоритму маршрутизації для бездротових mesh-мереж. У процесі виконання роботи було проаналізовано сучасні підходи до маршрутизації та виявлено, що існуючі алгоритми, зокрема AODV, HWMP і OLSR, не завжди ефективно працюють у динамічних умовах mesh-мереж. Основна проблема полягає в тому, що більшість із них враховують лише окремі параметри мережі та недостатньо адаптуються до змін топології, навантаження або якості каналів зв’язку.

У роботі було розроблено математичну модель комплексного оцінювання маршруту, яка враховує одразу кілька важливих характеристик мережі: затримку, втрати пакетів, пропускну здатність і стабільність з’єднання. На основі цієї моделі створено алгоритм MQAR, який дозволяє не лише обирати більш якісні маршрути, а й завчасно реагувати на погіршення стану мережі. Додатково реалізовано механізми адаптивного переключення маршрутів і запобігання частим непотрібним перебудовам, що позитивно вплинуло на стабільність роботи мережі.

Проведені симуляційні дослідження показали, що запропонований алгоритм у більшості випадків перевершує стандартні підходи. MQAR забезпечує вищу пропускну здатність, кращу стабільність маршрутів та нижчий рівень втрат пакетів порівняно з AODV і HWMP. Хоча в окремих сценаріях затримка була дещо більшою, це є допустимим компромісом для досягнення загальної стабільності та якості передачі даних. Особливо помітною перевага алгоритму стала в умовах мобільності вузлів, де MQAR швидше адаптувався до змін топології та демонстрував більш плавну роботу мережі без різких погіршень характеристик.

Отримані результати підтверджують ефективність запропонованого підходу та доцільність використання комплексної оцінки маршрутів у сучасних mesh-мережах. Практична цінність роботи полягає в можливості застосування алгоритму для Wi-Fi mesh-систем, IoT-мереж, систем розумного

міста та інших розподілених бездротових інфраструктур, також створений симулятор може бути основою для подальших досліджень і вдосконалення алгоритмів маршрутизації з урахуванням нових параметрів та сценаріїв роботи мереж.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. IEEE 802.11s Tutorial: Overview of the Amendment for Wireless Local Area Mesh Networking : [presentation] / W. S. Conner, J. Kruys, K. Kim, J. C. Zuniga ; IEEE 802 Plenary. – Dallas : IEEE, 2006. – 85 p. – URL: https://ieee802.org/802_tutorials/06-November/802.11s_Tutorial_r5.pdf

2. Akyildiz I. F., Wang X., Wang W. Wireless mesh networks: a survey. Computer Networks. 2005. Vol. 47, Iss. 4. P. 445–487. URL: https://www.ianakyildiz.com/bwn/papers/2005/j4.pdf

3. Saha S., Acharjee U. K., Tahzib-Ul-Islam Md. A survey on wireless mesh network and its challenges at the transport layer. International Journal of Computer Engineering and Technology. 2014. Vol. 5, Iss. 8. P. 169–177.

4. Naveen N. Wireless Mesh Networks: A Survey and Challenges : [preprint] / M. S. Ramaiah University of Applied Sciences. – 2024. – 20 p. – URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4909229

5. Pathak P. H., Dutta R. A Survey of Network Design Problems and Joint Design Approaches in Wireless Mesh Networks : [technical report] / North Carolina State University. – Raleigh : NCSU, 2011. – 32 p. – URL: https://www.researchgate.net/publication/260671069

6. Alabady S. A., Salleh M. F. M. Overview of Wireless Mesh Networks. School of Electrical and Electronic Engineering, Universiti Sains Malaysia. 2013. 7 p. URL: https://scispace.com/papers/overview-of-wireless-mesh-networks-5aquwjnd2m

7. De Couto D. S. J., Aguayo D., Bicket J., Morris R. A High-Throughput Path Metric for Multi-Hop Wireless Routing : [presentation] / MIT Computer Science and Artificial Intelligence Laboratory ; pres. by T. Cooper. – Cambridge : MIT, 2007. – 26 p. – URL: https://pdfs.semanticscholar.org/d998/b9e8187c6ebd3aa5f34bee10f6e1191e73a4.pd

8. Collection Tree Protocol / O. Gnawali, K. Jamieson, D. Moss et al. // Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems

(SenSys '09). – New York : ACM, 2009. – P. 1–14. – URL: https://www.researchgate.net/publication/221091782

9. Connectivity of wireless sensor networks for plant growth in greenhouse / C. Yang, S. Yuling, W. Zhongyi et al. // International Journal of Agricultural and Biological Engineering. – 2012. – Vol. 5, Iss. 1. – P. 41–48. – URL: https://www.researchgate.net/publication/293363410

10. Borcoci E. Wireless Mesh Networks Technologies: Architectures, Protocols, Resource Management and Applications : [presentation/tutorial] / University POLITEHNICA Bucharest. – Valencia : ICWMC, 2009. – 191 p. – URL: https://www.iaria.org/conferences2009/filesICWMC09/EugenBorcociTutorial.pdf

11. Perkins C. E., Royer E. M. Ad-hoc On-Demand Distance Vector Routing. Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA). 1999. URL: https://scispace.com/papers/ad-hoc-on-demand-distance-vector-routing-3w61wv0p1h

12. Johnson D. B., Maltz D. A. Dynamic Source Routing in Ad Hoc Wireless Networks. Mobile Computing. 1996. URL: https://www.researchgate.net/publication/2802692

13. Clausen T., Jacquet P. Optimized Link State Routing Protocol (OLSR) : RFC 3626. Network Working Group. 2003. URL: https://datatracker.ietf.org/doc/html/rfc3626

14. Perkins C. E., Bhagwat P. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. ACM SIGCOMM Computer Communication Review. 1994. URL: https://www.researchgate.net/publication/2734335

15. The SMesh Wireless Mesh Network : [technical report] / Y. Amir, C. Danilov, R. Musăloiu-Elefteri, N. Rivera ; Johns Hopkins University. – Baltimore : CNDS, 2009. – (Technical Report CNDS-2009-3). – URL: https://www.researchgate.net/publication/220439186

16. Bahr M. Proposed Routing for IEEE 802.11s WLAN Mesh Networks. Proceedings of the 2006 IEEE Workshop on a World of Wireless, Mobile and

Multimedia Networks (WoWMoM '06). 2006. P. 1–9. URL: https://www.eecs.yorku.ca/course_archive/2010-11/F/6590/Papers/bahr.pdf

17. Draves R., Padhye J., Zill B. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04). 2004. P. 114–128. URL: https://www.researchgate.net/publication/220926495

18. Camp T., Boleng J., Davies V. A survey of mobility models for ad hoc network research. Wireless Communications and Mobile Computing. 2002. Vol. 2, Iss. 5. P. 483–502. URL: https://www.researchgate.net/publication/2563707

19. Hong X. A Group Mobility Model for Ad Hoc Wireless Networks / X. Hong, M. Gerla, G. Pei, C. Chiang // Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems (MSWiM '99). – Seattle, Washington, USA, 1999. – P. 53–60. URL: https://www.csl.mtu.edu/cs5461/www/Reading/Hong-MSWim99.pdf

20. Kurose J. F., Ross K. W. Computer Networking: A Top-Down Approach. 6th ed. Boston : Pearson, 2013. 889 p. URL: https://www.cs.sjtu.edu.cn/~linghe.kong/CS339/Download/ComputerNetworking.pdf

21. De Couto D. S. J., Aguayo D., Bicket J., Morris R. A High-Throughput Path Metric for Multi-Hop Wireless Routing. Proc. MobiCom '03. 2003. P. 134–146.

22. One-way transmission time : Recommendation ITU-T G.114. Geneva : International Telecommunication Union, 2003. 20 p. (General Characteristics of International Telephone Connections and International Telephone Circuits).

23. IEEE Std 802.11g-2003. Amendment 4: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band. – New York : IEEE, 2003. – 74 p.

24. IEEE Std 802.11s-2011. Amendment 10: Mesh Networking. – New York : IEEE Computer Society, 2011. – 372 p.

25. Dijkstra E. W. A note on two problems in connexion with graphs. Numerische Mathematik. 1959. Vol. 1. P. 269–271

26. Yang Y., Wang J., Kravets R. Designing Routing Metrics for Mesh Networks. Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh). 2005.

27. De Couto D. S. J. High-Throughput Routing for Multi-Hop Wireless Networks : Ph.D. thesis / MIT. — Cambridge : Massachusetts Institute of Technology, 2004. — 153 p.

28. Rappaport T. S. Wireless Communications: Principles and Practice. 2nd ed. Upper Saddle River : Prentice Hall, 2002. 736 p.

29. Goldsmith A. Wireless Communications. Cambridge : Cambridge University Press, 2005. 644 p.

30. Lin S., Costello D. J. Error Control Coding. 2nd ed. Upper Saddle River : Pearson Prentice Hall, 2004. 1260 p.

31. Johnson D. B., Maltz D. A., Hu Y.-C. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4 : RFC 4728. IETF. 2007. URL: https://datatracker.ietf.org/doc/html/rfc4728

32. Hunter J. D. Matplotlib: A 2D graphics environment. Computing in Science & Engineering. 2007. Vol. 9, No. 3. P. 90–95.

33. Harris C. R. et al. Array programming with NumPy. Nature. 2020. Vol. 585. P. 357–362. URL: https://doi.org/10.1038/s41586-020-2649-2

34. Virtanen P. et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods. 2020. Vol. 17. P. 261–272. URL: https://doi.org/10.1038/s41592-019-0686-2