

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

Навчально-науковий інститут телекомунікаційних систем

Кафедра телекомунікацій

«На правах рукопису»

УДК _____

«До захисту допущено»

Завідувач кафедри

_____ Сергій КРАВЧУК

« ____ » _____ 2025 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою «Інженерія та програмування
інфокомунікацій»**

зі спеціальності 172 «Електронні комунікації та радіотехніка»

**на тему: «Впровадження туманних обчислень в мережах електронних
комунікацій»**

Виконала:

студентка II курсу, групи ЦК-41мп

Білоус Карина Ігорівна

Керівник:

Доцент кафедри ТК НН ІТС, к.т.н., доцент

Явіся Валерій Сергійович

Рецензент:

Доцент кафедри ІТТ НН ІТС, к.т.н., доцент

Правило Валерій Володимирович

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

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Електронні комунікації та радіотехніка»

Освітньо-професійна програма «Інженерія та програмування інфокомунікацій»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій КРАВЧУК

«__» _____ 2025 р.

ЗАВДАННЯ
на магістерську дисертацію студентці
Білоус Карині Ігорівні

1. Тема дисертації «Впровадження туманних обчислень в мережах електронних комунікацій», науковий керівник дисертації Явіся Валерій Сергійович, к.т.н., доцент, затверджені наказом по університету від «03» листопада 2025 р. № 4772-с.
2. Термін подання студентом дисертації 15.12.2025 р.
3. Об'єкт дослідження: Процеси перенесення обчислювальних завдань (Task Offloading) у ієрархічних мережах Edge-Fog-Cloud.
4. Предмет дослідження: Адаптивна модель управління Task Offloading та архітектура RAFN-AgriProtect, заснована на багатоцільовій оптимізації, в умовах автономного сільськогосподарського сектору.
5. Перелік завдань, які потрібно розробити:
 - Проведення аналізу еволюції обчислювальних парадигм;
 - Систематизація та порівняльний аналіз існуючих методів Task Offloading у МЕС-мережах для визначення архітектурних прогалів;

- Розробка та моделювання інтегрованої архітектури обчислювальної системи та Адаптивного Алгоритму Task Offloading, заснованого на багатоцільовій оптимізації;

- Розробка стартап-проєкту.

6. Орієнтовний перелік ілюстративного матеріалу:

Слайд 1: Вступ та актуальність.

Слайд 2-3: Концепція обчислювального континууму та Fog Computing.

Слайд 4: Архітектурна модель RAFN-AgriProtect.

Слайд 5-6: Адаптивний алгоритм Task Offloading.

Слайд 7-8: Симуляційне моделювання та параметри.

Слайд 9: Парето-оптимальні рішення.

Слайд 10: Порівняльний аналіз стратегій.

Слайд 11: Загальні висновки.

7. Дата видачі завдання 02 вересня 2024 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Розробка, оформлення, узгодження та затвердження технічного завдання на роботу. Підбір та опрацювання необхідної науково технічної літератури.	02.09.2024 - 18.09.2025	Виконано
2	Підготовка матеріалів першого розділу	19.09.2025 - 29.09.2025	Виконано
3	Підготовка матеріалів другого розділу	30.09.2025 - 25.10.2025	Виконано
4	Підготовка матеріалів третього розділу	26.10.2025 - 24.11.2025	Виконано
5	Підготовка матеріалів четвертого розділу	25.11.2025 - 12.12.2025	Виконано
6	Оформлення пояснювальної записки та підготовка матеріалів до друку	13.12.2025 - 15.12.2025	Виконано

Студент

Карина БІЛОУС

Науковий керівник дисертації

Валерій ЯВІСЯ

РЕФЕРАТ

Робота містить 112 сторінок, (6 рисунків, 26 таблиць). Список використаних джерел нараховує 25 найменувань.

Актуальність впровадження туманних обчислень в мережах електронних комунікацій полягає в необхідності подолання архітектурних обмежень традиційної парадигми Хмарних обчислень (Cloud Computing) для високоавтоматизованих систем (дрони, сенсори) у сільському господарстві. Обмежена пропускна здатність магістральних каналів у сільській місцевості призводить до неприпустимо високої мережевої затримки (Latency) та значних фінансових витрат, що унеможлиблює прийняття автономних рішень у реальному часі та порушує вимоги до Якості Обслуговування (QoS).

Мета дослідження. Розробка та оцінка ефективності адаптивної моделі управління Task Offloading у MEC-мережах, орієнтованих на аграрний сектор, яка використовує принципи багатоцільової оптимізації для забезпечення оптимального компромісу між продуктивністю (Latency) та енергоефективністю, дотримуючись жорстких обмежень QoS, надійності та безпеки.

Завдання дослідження. Розробка архітектурної моделі RAFN-AgriProtect, математична формалізація задачі Task Offloading як багатоцільової оптимізації, створення Адаптивного Алгоритму Task Offloading (AA-TO), а також проведення симуляційного моделювання для кількісного підтвердження переваг розроблених рішень.

Об'єкт дослідження. Процеси перенесення обчислювальних завдань (Task Offloading) у ієрархічних мережах Edge-Fog-Cloud.

Предмет дослідження. Адаптивна модель управління Task Offloading та архітектура RAFN-AgriProtect, заснована на багатоцільовій оптимізації, в умовах автономного сільськогосподарського сектору.

Методи дослідження, що були використані. Системний аналіз, математичне моделювання, теорія масового обслуговування, методи

багатоцільової оптимізації (зокрема, еволюційні алгоритми), а також Дискретно-Подієве Моделювання (DES) на базі Python для оцінки продуктивності системи.

Отримані результати. Розроблена цілісна архітектура RAFN-AgriProtect та Адаптивний алгоритм Task Offloading (AA-TO), що динамічно знаходить Парето-оптимальні рішення. Результати моделювання підтвердили досягнення мінімальної середньої затримки та дотримання дедлайнів на рівні, близькому до 99%.

Практична цінність отриманих результатів полягає в кількісному підтвердженні економічної вигоди архітектури RAFN-AgriProtect за рахунок механізму локального обслуговування (P2P Fog Delivery), що забезпечує зниження навантаження на магістральний канал та економію операційних витрат (OpEx) приблизно на 75%. Отримані результати є основою для реалізації інноваційного стартап-проєкту «AgriFog Optimize».

Ключові слова: туманні обчислення, Task Offloading, багатоцільова оптимізація, Mobile Edge Computing (MEC), RAFN-AgriProtect, Latency, енергоспоживання.

ABSTRACT

The work contains **112** pages, (**6** figures, **26** tables). The list of references includes **25** titles.

The relevance of implementing Fog Computing in electronic communications networks lies in the necessity to overcome the architectural limitations of the traditional Cloud Computing paradigm for highly automated systems (drones, sensors) in agriculture. Limited bandwidth of backbone channels in rural areas leads to unacceptably high network latency and significant financial costs, which prevents real-time autonomous decision-making and violates Quality of Service (QoS) requirements.

Objective of the work. Development and evaluation of the effectiveness of an adaptive model for Task Offloading management in MEC networks oriented towards the agricultural sector, which utilizes multi-objective optimization principles to ensure an optimal trade-off between performance (Latency) and energy efficiency, while adhering to strict constraints on QoS, reliability, and security.

Objectives of the research. Development of the RAFN-AgriProtect architectural model, mathematical formalization of the Task Offloading problem as a multi-objective optimization, creation of the Adaptive Task Offloading Algorithm (AA-TO), as well as conducting simulation modeling to quantitatively confirm the advantages of the developed solutions.

Object of the research. The processes of computational task offloading (Task Offloading) in hierarchical Edge-Fog-Cloud networks.

Subject of the research. The adaptive Task Offloading management model and the RAFN-AgriProtect architecture, based on multi-objective optimization, in the context of the autonomous agricultural sector.

Research methods. System analysis, mathematical modeling, queueing theory, multi-objective optimization methods (including evolutionary algorithms), and Discrete-Event Simulation (DES) based on Python for system performance evaluation.

Obtained results. The developed integrated RAFN-AgriProtect architecture and the Adaptive Task Offloading Algorithm (AA-TO), which dynamically finds Pareto-optimal solutions. Simulation results confirmed the achievement of minimal average latency and deadline adherence at a level close to 99%.

The practical value of the obtained results lies in the quantitative confirmation of the economic benefits of the RAFN-AgriProtect architecture due to the local service mechanism (P2P Fog Delivery), which ensures reduced load on the backbone channel and savings in operating expenses (OpEx) of approximately 75%. The obtained results form the basis for the implementation of the innovative startup project "AgriFog Optimize".

Keywords. Fog Computing, Task Offloading, Multi-Objective Optimization, Mobile Edge Computing (MEC), RAFN-AgriProtect, Latency, Energy Consumption.

ЗМІСТ

ВСТУП.....	11
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ТА КОНЦЕПЦІЯ ТУМАННИХ ОБЧИСЛЕНЬ 12	
1.1 Еволюція парадигм обчислень та концептуальні основи Fog Computing..	12
1.2 Архітектурна модель та принципи функціонування Fog-мереж	17
1.3 Ключові вимоги та критерії впровадження Fog-архітектури.....	22
1.4 Сценарії застосування Fog Computing та вплив на мережеву інфраструктуру	30
Висновки	36
РОЗДІЛ 2 МЕТОДИ ТА МОДЕЛІ ВПРОВАДЖЕННЯ ТУМАННИХ ОБЧИСЛЕНЬ В ЕЛЕКТРОННИХ КОМУНІКАЦІЯХ	39
2.1. Моделі розміщення, топології та принципи мережевої взаємодії Fog-вузлів	39
2.2. Методи управління завданнями (Task Offloading) та динамічного балансування навантаження.....	49
2.3. Концептуальна постановка задачі Task Offloading як багатоцільової оптимізації та вибір адаптивної стратегії	51
2.4 Проблеми безпеки, конфіденційності та управління в туманних обчисленнях	59
Висновки	62
РОЗДІЛ 3 МОДЕЛЮВАННЯ СИСТЕМИ НА БАЗІ ТУМАННИХ ОБЧИСЛЕНЬ	64
3.1. Архітектурна модель RAFN-AgriProtect.....	64
3.2. Математична постановка та системна модель Task Offloading	68
3.3. Розробка адаптивного багатоцільового алгоритму Task Offloading (AA-TO)	70
3.4. Сценарії моделювання та оцінка ефективності RAFN.....	74
3.5. Аналіз результатів моделювання та оцінка ефективності RAFN-AgriProtect	77
Висновки	80
РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЄКТУ	82
4.1. Обґрунтування стартап-проєкту та концепція	82

4.2. Технологічний аудит ідеї проекту	85
4.3. Маркетингове дослідження та аналіз ринку	87
4.4. Розроблення ринкової стратегії проекту	98
4.5. Розроблення маркетингової програми стартап-проекту	102
Висновки	107
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ	109
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	110

ПЕРЕЛІК СКОРОЧЕНЬ

AA-TO	Адаптивний Алгоритм Task Offloading
AR/VR	Доповнена/Віртуальна реальність (Augmented/Virtual Reality)
CapEx	Капітальні витрати (Capital Expenditures)
DES	Дискретно-Подієве Моделювання (Discrete-Event Simulation)
IIoT	Промисловий Інтернет Речей (Industrial Internet of Things)
IoT	Інтернет речей (Internet of Things)
MEC	Мобільні граничні обчислення (Mobile Edge Computing)
MOPSO	Багатоцільова оптимізація рою частинок (Multi-Objective Particle Swarm Optimization)
NSGA-II	Недоміноване Сортування Генетичного Алгоритму II (Non-dominated Sorting Genetic Algorithm II)
OpEx	Операційні витрати (Operational Expenditures)
PaaS	Платформа як послуга (Platform as a Service)
P2P	Одноранговий зв'язок (Peer-to-Peer) (як частина P2P Fog Delivery)
QoS	Якість Обслуговування (Quality of Service)
RAFN-AgriProtect	Rural Autonomous Fog Network for Agro-Protection
SaaS	Програмне забезпечення як послуга (Software as a Service)
5G/6G	П'яте/Шосте покоління мобільного зв'язку

ВСТУП

Впровадження високоавтоматизованих систем, зокрема у сфері автономного сільського господарства, стало одним із найбільш значущих векторів технологічного розвитку останнього десятиліття. Масштабне використання інтелектуальних пристроїв, таких як сенсорні платформи, аграрні дрони та роботизовані комплекси, у польових умовах призводить до зростання обсягів згенерованих даних. Ця інформація, критично важлива для забезпечення точного землеробства та оперативного управління ресурсами, вимагає обробки в режимі, наближеному до реального часу, а відтак — забезпечення критично низької мережевої затримки (Latency).

Традиційна архітектура електронних комунікацій, заснована на централізованих Хмарних обчисленнях (Cloud Computing), виявляється архітектурно неефективною для задоволення цих вимог. Значна фізична відстань між кінцевими пристроями та віддаленими дата-центрами створює неприпустимо високий час відгуку, що унеможливлює прийняття автономних рішень на місці та порушує вимоги до Якості Обслуговування (QoS). Крім того, необхідність постійної передачі всього масиву згенерованих даних через обмежені за пропускнуою здатністю магістральні канали, типові для географічно розподілених сільських регіонів, генерує значні операційні витрати та створює ризик перевантаження основної мережевої інфраструктури.

З огляду на ці фундаментальні обмеження, виникла необхідність у новій парадигмі обчислень. Цей підхід, відомий як Туманні обчислення (Fog Computing), передбачає перенесення частини обчислювальних ресурсів на проміжний рівень. Саме ця трансформація може забезпечити потенціал для подолання обмежень традиційної Cloud-моделі та створення гнучкої, стійкої та економічно ефективно обчислювальної інфраструктури, здатної гарантувати високу продуктивність.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ТА КОНЦЕПЦІЯ ТУМАННИХ ОБЧИСЛЕНЬ

1.1 Еволюція парадигм обчислень та концептуальні основи Fog Computing

Концепція туманних обчислень (Fog Computing) сформувалася як безпосередня архітектурна відповідь на технологічні виклики, спричинені стрімким розвитком Інтернету речей (Internet of Things, IoT) та вимогою до обробки даних із мінімальною затримкою [1]. До її появи, обчислювальні процеси переважно покладалися на традиційну модель хмарних обчислень (Cloud Computing), де обробка інформації відбувається у віддалених централізованих дата-центрах. Зіткнення вимог до швидкості та обмежень централізації створило потребу у новій, децентралізованій та ієрархічній інфраструктурі.

Історично модель Cloud Computing довгий час вважалася універсальним рішенням для масштабованої обробки та зберігання даних [2]. Однак масове впровадження IoT та розвиток мереж 5G/6G значно посилили критичні вимоги до швидкості реакції та пропускну здатності. Передавання всього масиву інформації до центральних дата-центрів призводить до неприпустимої затримки, що є критичною для застосувань у сферах безпеки, автономного транспорту, промисловості та телемедицини.

Для часткового розв'язання цієї проблеми була запропонована концепція периферійних обчислень (Edge Computing), що виконує обробку даних безпосередньо на кінцевих пристроях або шлюзах [3]. Проте, обмежені обчислювальні ресурси Edge-пристроїв та відсутність координації між ними ускладнювали побудову великих систем із гарантованою якістю обслуговування.

На цьому етапі виникла потреба у новій парадигмі, здатній поєднати високу обчислювальну потужність хмари з близькістю до кінцевих пристроїв [4]. Так з'явилися туманні обчислення (Fog Computing).

У становленні концепції Fog Computing ключову роль відіграла компанія Cisco Systems. Дослідники компанії першими офіційно запровадили цей термін, а в 2012 році обґрунтували його основні принципи та визначальні характеристики у фундаментальній роботі «Fog Computing and its Role in the Internet of Things» (Bonomi et al., 2012)[5].

У науковій та інженерній практиці Fog Computing прийнято визначати як горизонтальну, системно-рівневу архітектуру, головна мета якої полягає у розподілі обчислювальних, сховищних та мережевих ресурсів [6]. Ці ресурси та послуги керування застосунками розміщуються ближче до кінцевих пристроїв та користувачів, на відміну від традиційного розміщення виключно у віддаленій хмарі.

З огляду на обговорену еволюцію, важливо чітко розмежувати три основні парадигми, які формують сучасний обчислювальний простір: Cloud Computing, Fog Computing та Edge Computing. Хоча на практиці вони часто інтегруються в єдину інфраструктуру (обчислювальний континуум), їхні концептуальні відмінності щодо розташування та функціональності є критичними для розуміння.

Cloud Computing є найбільш віддаленою та централізованою парадигмою, фокусуючись на глобальних, ресурсомістких завданнях і довготривалому зберіганні. Edge Computing представляє обчислення, максимально наближені до кінцевих пристроїв і сенсорів. Fog Computing займає у цьому континуумі проміжний рівень, функціонуючи як мережа розподілених мікро-центрів обробки даних. Це дозволяє йому забезпечувати локальну аналітику та агрегацію даних, чого не можуть надати обмежені Edge-пристрої [7]. Відповідне розташування Cloud, Fog та Edge Computing у структурі обчислювального континууму представлено на рис 1.1.

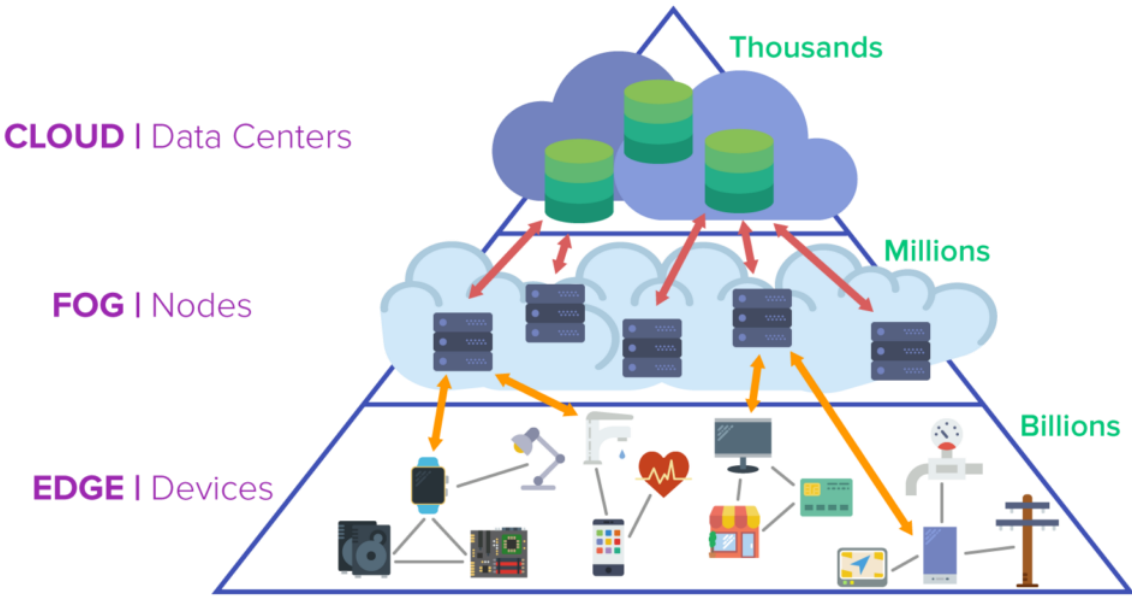


Рис. 1.1. Обчислювальний континуум: розподіл обчислювальних ресурсів від Хмари до Периферії (Cloud-to-Edge Continuum).

Порівняльний аналіз ключових характеристик Cloud, Fog та Edge Computing за критеріями продуктивності, затримки та архітектурних особливостей наведено в табл. 1.1.

Таблица 1.1.

Порівняльний аналіз парадигм Cloud, Fog та Edge Computing

Критерій	Cloud Computing	Fog Computing	Edge Computing
Основна функція / Мета	Глобальна аналітика, довготривале зберігання, ресурсомісткі обчислення.	Локальна аналітика, агрегація даних, керування застосунками в реальному часі.	Збір, фільтрація та миттєва реакція на дані на рівні пристрою.
Розташування обчислень	Централізовані дата-центри, віддалені від користувача.	Розподілена мережа вузлів, розташована на периферії локальної мережі.	Децентралізовані кінцеві пристрої та шлюзи.

Продовження таблиці 1.1.

Затримка (Latency)	Висока (від 100 мс до секунд), залежить від магістральних каналів зв'язку.	Низька (від 10 мс до 100 мс).	Наднизька (менше 10 мс), блискавична реакція.
Пропускна здатність	Вимагає високої пропускної здатності для передачі сирих даних.	Оптимізує пропускну здатність, передаючи лише агреговані дані.	Обмежена пропускна здатність локальних каналів.
Обчислювальна потужність	Необмежена (масштабна, еластична).	Помірна (достатня для локальної аналітики та керування).	Обмежена (залежить від пристрою).
Географічний розподіл	Низький (концентрований у дата-центрах).	Високий (охоплює великі географічні території).	Максимально високий (охоплює кожен кінцевий пристрій).
Підтримка мобільності	Низька (важко підтримувати сесію при зміні мережі).	Висока (здатна підтримувати мобільних користувачів і пристрої).	Середня (локальні пристрої).
Комунікація між вузлами	Ні (кожен клієнт звертається до центральної Хмари).	Так, горизонтальна (Fog-вузли спілкуються між собою для координації).	Ні (пристрої, як правило, незалежні).

Затримка та пропускна здатність

Аналіз Таблиці 1.1 демонструє, що **затримка** є ключовим фактором, який розмежовує обчислювальні парадигми. Модель Cloud Computing з її віддаленими дата-центрами об'єктивно не здатна задовольнити вимоги до швидкості реакції, які висуває Інтернет речей та мережі 5G/6G. У цьому контексті Fog Computing виступає критично важливим шаром, оскільки розміщує обчислення ближче до джерел даних, знижуючи затримку до рівня 10-100 мс [8]. Це робить його

ідеальним для додатків, що вимагають миттєвого прийняття рішень (автономний транспорт, промисловий контроль). Крім того, Fog-вузли здійснюють агрегацію та фільтрацію даних. Це значно зменшує обсяг трафіку, що передається до Хмари, оптимізуючи пропускну здатність магістральних мереж і запобігаючи їхньому перевантаженню. Таким чином, Fog Computing не лише прискорює обробку, але й підвищує ефективність використання мережевої інфраструктури.

Обчислювальна потужність та основна функція

Згідно з порівняльним аналізом, обчислювальна потужність є ключовим фактором при виборі парадигми. Cloud Computing пропонує практично необмежені ресурси, що ідеально підходить для глобальної аналітики, Big Data та довготривалого зберігання. На протилежному полюсі знаходиться Edge Computing з його вкрай обмеженою потужністю, достатньою лише для збору та первинної фільтрації даних. Fog Computing займає стратегічно важливу проміжну позицію, надаючи помірну потужність, якої, однак, цілком вистачає для виконання складних локалізованих завдань. Основна функція Fog — локальна аналітика та керування застосунками в реальному часі — дозволяє виконувати машинне навчання та агрегацію даних безпосередньо на периферії [9]. Це долає обмеження Edge-пристроїв і забезпечує функціональний сервісний рівень, який є життєво необхідним для більшості критичних IoT-систем.

Географічний розподіл та підтримка мобільності

Критерії географічного розподілу та підтримки мобільності підкреслюють здатність архітектури охоплювати великі території та обслуговувати рухомі об'єкти. Cloud Computing має низький географічний розподіл, оскільки ресурси концентруються у віддалених центрах, що також створює проблеми з підтримкою мобільності користувачів при зміні зон покриття. На противагу цьому, Fog Computing спроектований як широко розподілена мережа вузлів, які охоплюють значні території (наприклад, ціле місто або промисловий об'єкт). Така структура дозволяє Fog-архітектурі безшовно підтримувати мобільність, гарантуючи неперервність обслуговування (наприклад, для автономних

транспортних засобів, дронів чи мобільних медичних моніторів). Ця унікальна здатність забезпечувати якість обслуговування в русі є однією з головних переваг Fog перед централізованими моделями [10].

Горизонтальна комунікація та координація вузлів

Найбільш визначальною функціональною відмінністю Fog Computing є наявність горизонтальної комунікації між вузлами одного рівня. У моделях Cloud та Edge така координація відсутня: пристрої Edge є ізольованими, а Cloud-клієнти не спілкуються напряду. Горизонтальний зв'язок між Fog-вузлами дозволяє їм функціонувати як єдиній, злагодженій платформі [11]. Це має вирішальне значення для координації ресурсів, балансування навантаження та спільної обробки даних у розподілених застосуваннях. Наприклад, у системах безпеки сусідні Fog-вузли можуть обмінюватися інформацією про загрозу в реальному часі, підвищуючи надійність усієї системи. Таким чином, горизонтальна комунікація перетворює сукупність Fog-вузлів на інтелектуальну мережу, що є основою для їхньої архітектурної моделі.

1.2 Архітектурна модель та принципи функціонування Fog-мереж

Концептуальне позиціонування Fog Computing у межах обчислювального континууму (розглянуте в 1.1) вимагає деталізації внутрішньої структури. Оскільки Fog не є монолітною чи єдиною архітектурою, він функціонує як багаторівнева (як правило, трирівнева) модель, що слугує мостом між високопродуктивною, але віддаленою Хмарою, та обмеженими кінцевими пристроями [12]. Основна мета такої структури — забезпечити ефективну децентралізацію обчислень, керування трафіком та надійність через координацію ресурсів.

Для реалізації згаданих переваг децентралізації та низької затримки, кожен Fog-вузол (Fog Node) має чітку логічну організацію. На відміну від монолітних систем, архітектура Fog-платформи базується на трирівневій логічній моделі, що відображає послідовність функціональних завдань і забезпечує необхідну

модульність. Логічна будова Fog-платформи, що забезпечує виконання цих принципів, представлена на рис. 1.2.

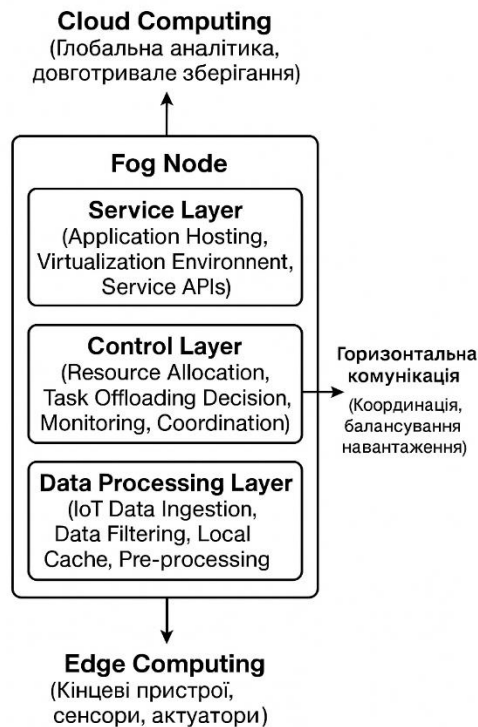


Рис. 1.2 Логічна архітектура Fog-вузла

Знизу вгору ці шари включають:

1. Рівень обробки даних (відповідає за прийом, фільтрацію та агрегацію інформації від Edge).
2. Рівень керування (здійснює моніторинг, розподіл ресурсів та приймає рішення про перенесення завдань).
3. Рівень сервісів (надає кінцеві застосунки та підтримує віртуалізацію).

Саме така вертикальна структура забезпечує ефективну взаємодію з Edge-пристроями та гарантує виконання критично важливих локальних обчислень.

Гетерогенність обладнання та роль віртуалізації у Fog-платформі

Реалізація логічної архітектури Fog-платформи залежить від її гетерогенної фізичної інфраструктури. На відміну від однотипних дата-центрів Cloud, Fog-вузли не є уніфікованими серверами; вони включають широкий

спектр пристроїв: від промислових контролерів та потужних маршрутизаторів до мікро-серверів, комутаторів та базових станцій мобільного зв'язку [13]. Така гетерогенність обладнання є необхідною, оскільки вона дозволяє розміщувати обчислення близько до місць генерації даних, але водночас вимагає уніфікованого програмного середовища. Ключову роль у керуванні цим різноманіттям відіграє віртуалізація. Використання контейнерів (наприклад, Docker) та віртуальних машин (VM) дозволяє абстрагувати застосунки від базового апаратного забезпечення [14]. Це забезпечує необхідну гнучкість, спрощує розгортання сервісів (на Рівні Сервісів) і дає змогу Менеджеру ресурсів (на Рівні Керування) динамічно виділяти потужність відповідно до локального навантаження. Таким чином, віртуалізація гарантує масштабованість і ефективність використання різнорідних ресурсів Fog-мережі.

Принципи комунікації Fog-мережі

Ключова архітектурна перевага Fog Computing полягає у його здатності ефективно керувати потоками даних і обчислювальними завданнями на різних рівнях континууму. Це стає можливим завдяки функціонуванню двох основних типів комунікації: вертикальної (ієрархічної), що відповідає за передачу даних між рівнями Edge, Fog та Cloud; та горизонтальної (координаційної), що забезпечує взаємодію між самими Fog-вузлами. Ці принципи зв'язку є визначальними для реалізації таких критичних функцій, як зниження затримки та підтримка мобільності.

Вертикальна комунікація: Edge ↔ Fog ↔ Cloud

Вертикальна комунікація у Fog-мережі визначає ієрархічний потік даних і завдань. Вона складається з двох ключових зв'язків. Перший — це зв'язок Edge ↔ Fog, який є критично важливим для збору даних. На цьому рівні застосовуються легковагі, орієнтовані на IoT протоколи, як-от MQTT (Message Queuing Telemetry Transport) або CoAP (Constrained Application Protocol). Ці протоколи оптимізовані для роботи в мережах з обмеженою пропускнуою здатністю, низькою потужністю та нестабільним зв'язком, що характерно для

кінцевих пристроїв [15]. Fog-вузол виступає як шлюз і брокер, агрегуючи та фільтруючи великий обсяг сирих даних. Другий зв'язок — Fog ↔ Cloud — відбувається на вищому рівні та зазвичай використовує більш традиційні та надійні протоколи, як-от HTTPS/HTTP. На цьому етапі до Cloud передаються вже не сирі, а лише агреговані, критично важливі або архівні дані, що значно розвантажує магістральні канали мережі [16]. Таким чином, вертикальна комунікація забезпечує ефективну двосторонню передачу, гарантуючи, що обробка даних відбувається на оптимальному для цього архітектурному рівні.

Горизонтальна комунікація: Fog ↔ Fog

Унікальною та фундаментальною відмінністю Fog Computing від Edge-підходу є наявність горизонтальної комунікації — зв'язку між сусідніми Fog-вузлами (Fog ↔ Fog). Ця функція є вирішальною для перетворення окремих вузлів на єдину, злагоджену мережеву платформу. Горизонтальна комунікація реалізує концепцію координації ресурсів, дозволяючи вузлам обмінюватися інформацією про своє поточне навантаження, доступні обчислювальні потужності, рівень завантаження сховища та дані про мобільних користувачів. Це необхідно для ефективного балансування навантаження (Load Balancing): якщо один вузол перевантажений, сусідній може прийняти частину його завдань або даних. Крім того, ця комунікація є основою для підтримки мобільності, оскільки забезпечує безперервне обслуговування користувача при його переході між зонами покриття різних вузлів. Горизонтальний зв'язок гарантує відмовостійкість та підвищує загальну надійність системи, роблячи її стійкою до локальних збоїв.

Концепція перенесення обчислювальних завдань (Task Offloading)

Перенесення обчислювальних завдань (Task Offloading) є одним із фундаментальних механізмів, що забезпечують функціональність та ефективність архітектури Fog Computing, будучи прямою відповіддю на обмеження периферійних пристроїв. У контексті обчислювального континууму, Task Offloading визначається як процес міграції частини обчислювальних

навантажень з ресурсно-обмеженого кінцевого пристрою (Edge) на більш потужний і географічно наближений вузол обробки (Fog).

Основна мета цього процесу полягає у вирішенні трьох критичних проблем, притаманних Edge-пристроюм: обмежена енергоємність, низька обчислювальна потужність та високі вимоги до часу відгуку. Обробка складних, ресурсомістких завдань (наприклад, розпізнавання образів, машинне навчання) безпосередньо на Edge-пристрої призводить до швидкого виснаження батареї та неприпустимої затримки (latency). Переносячи ці завдання на Fog-вузол, система дозволяє Edge-пристроюм зосередитися виключно на зборі та первинній фільтрації даних. Таким чином, Task Offloading обґрунтовано виступає як ключовий механізм, що дозволяє реалізувати додатки, які вимагають інтенсивної обробки даних у реальному часі, зберігаючи при цьому переваги близькості до кінцевого користувача, що є неможливим при прямому зверненні до віддаленої Хмари.

Фази процесу Task Offloading

Перша фаза — Рішення про перенесення (Offloading Decision). Вона відбувається на Edge-пристрої, де спеціалізований модуль моніторить його поточний стан: рівень заряду батареї, завантаження процесора та критичні вимоги завдання до затримки. Якщо моніторинг показує, що виконання завдання локально призведе до перевищення часових лімітів, надмірного енергоспоживання або перегріву, ініціюється процес перенесення. Цей етап є бінарним: виконати локально або передати до Fog.

Друга фаза — Вибір вузла (Node Selection). Якщо прийнято рішення про перенесення, Fog-мережа, використовуючи горизонтальну комунікацію, оцінює наявні ресурси в найближчих Fog-вузлах. Мережевий механізм, на основі заздалегідь визначених критеріїв, ідентифікує оптимальний вузол з найменшою затримкою, найнижчим навантаженням і максимальною обчислювальною потужністю. Успішний вибір вузла є критичним для запобігання перевантаженню Fog-мережі та забезпечення низької затримки.

Третя фаза — Виконання (Execution). Після успішного вибору, завдання разом з необхідними даними передається обраному Fog-вузлу, де воно виконується. Завдання, залежно від його складності та розміру, може виконуватися у віртуальній машині або контейнері, ізольовано від інших процесів. Результати обчислень потім повертаються Edge-пристроєм через вертикальний канал комунікації. Цей трифазний цикл є основою динамічного управління ресурсами у Fog-архітектурі та забезпечує її ключову перевагу в реальному часі.

Перелік метрик для вибору вузла

Для того, щоб фаза Вибору вузла у процесі Task Offloading була ефективною, архітектура Fog-мережі використовує набір кількісних критеріїв, або метрик, для оцінки придатності кожного потенційного Fog-вузла. Ці метрики формують основу для прийняття рішень на Рівні Керування. Ключові критерії вибору поділяються на кілька типів. По-перше, Обчислювальне навантаження (Computational Load): це показники завантаження процесора (CPU) та використання пам'яті (RAM) на Fog-вузлі. Оцінка навантаження дозволяє уникнути передачі завдання на вже перевантажений вузол, що може призвести до високої затримки. По-друге, Енергоспоживання (Energy Consumption) є критичною метрикою з боку Edge-пристрою: перенесення завдання має бути енергетично вигіднішим, ніж його локальне виконання. По-третє, Час виконання та Затримка (Execution Time and Latency). Ця метрика враховує як час передачі даних, так і безпосередньо час обробки завдання на Fog-вузлі, що є визначальним для застосувань реального часу. На основі цих критеріїв система приймає найбільш обґрунтоване рішення про міграцію.

1.3 Ключові вимоги та критерії впровадження Fog-архітектури

Ефективне впровадження будь-якої архітектурної моделі, включно з Fog Computing (описаною в 1.2), неможливе без чіткого визначення набору кількісних критеріїв, яким вона повинна відповідати. Ці критерії, або вимоги, обумовлені специфікою застосувань на периферії, де висока динаміка,

обмеженість ресурсів і потреба у реакції в реальному часі є нормою. Вимоги до Fog-мереж є значно жорсткішими, ніж до традиційних Cloud-систем, і слугують основою для оцінки продуктивності та надійності всієї інфраструктури.

Проектування ефективної архітектури Fog Computing вимагає ретельного аналізу не лише функціональних завдань (таких як перенесення завдань та балансування навантаження), але й критичних нефункціональних вимог, що визначають якість обслуговування. Саме ці нефункціональні вимоги, зокрема Затримка (Latency), Надійність (Reliability) та Енергоефективність (Energy Efficiency) тощо, є визначальними для критичних застосувань (Critical Applications).

Затримка (Latency) є найбільш критичною нефункціональною вимогою для архітектури Fog Computing, оскільки вона безпосередньо визначає придатність системи до обслуговування застосувань реального часу (Real-Time Applications). На відміну від традиційних Cloud-систем, де допустима затримка часто вимірюється сотнями мілісекунд через фізичну віддаленість дата-центрів, Fog Computing націлений на досягнення надзвичайно низької затримки (Ultra-Low Latency). Кількісно це виражається у вимозі забезпечити час відгуку в діапазоні, що може становити від 1 до 50 мілісекунд (мс). Ця жорстка вимога є абсолютною необхідністю для критичних сфер, таких як системи безпеки, управління безпілотними літальними апаратами, дистанційна хірургія та, особливо, автономне водіння (V2X), де рішення про гальмування має бути прийняте за доли секунди.

У контексті розподіленої Fog-архітектури, загальна затримка завдання складається з трьох ключових компонентів, кожен з яких підлягає оптимізації: час передачі даних від кінцевого Edge-пристрою до обраного Fog-вузла, час безпосередньої обробки завдання на ресурсах Fog-вузла та час повернення результатів до Edge. Зменшення затримки досягається завдяки фундаментальній перевазі Fog — його географічній близькості до джерела даних, що мінімізує

кількість мережевих переходів (hop count) та усуває необхідність використання магістральних мереж.

Таким чином, вимога до низької затримки є головним рушієм механізму Task Offloading (перенесення обчислювальних завдань), вимагаючи від системи алгоритмічно обирати найближчий і найменш завантажений вузол. Усі рішення щодо керування ресурсами Fog-мережі, включаючи балансування навантаження (Load Balancing), повинні мати мінімізацію затримки як основний або один із пріоритетних критеріїв. Це прямо впливає на постановку задачі оптимізації, де час виконання завдання є ключовою змінною, яку необхідно враховувати та мінімізувати в цільовій функції для забезпечення гарантованої якості обслуговування (QoS).

Надійність (Reliability) є ще однією важливою нефункціональною вимогою, особливо в контексті розподілених обчислень на периферії. Вона визначається як здатність системи або її компонентів безперервно виконувати необхідні функції протягом визначеного періоду часу за заданих умов. Для Fog Computing ця вимога має критичне значення, оскільки Fog-вузли часто працюють у неконтрольованих або суворих умовах (наприклад, промислові шлюзи, вуличні маршрутизатори), де ризик апаратного або програмного збою значно вищий, ніж у централізованих Cloud-дата-центрах.

Вимога до надійності у Fog-архітектурі трансформується у необхідність забезпечення стійкості до відмов (Fault Tolerance). Це означає, що відмова одного Fog-вузла не повинна призводити до припинення обслуговування критичних застосувань (Critical Applications). Ця стійкість забезпечується, в першу чергу, завдяки надлишковості (redundancy) та горизонтальній комунікації між сусідніми вузлами (які вже були концептуально описані в 1.2). Саме горизонтальна координація дозволяє сусіднім вузлам швидко виявляти збій і приймати на себе обчислювальне навантаження несправного вузла.

Надійність архітектури є безпосередньою основою для впровадження механізмів Балансування навантаження (Load Balancing). У цьому

контексті, Load Balancing набуває подвійної ролі. З одного боку, він забезпечує оптимальний розподіл обчислювальних ресурсів у нормальному режимі роботи, запобігаючи перевантаженню окремих вузлів. З іншого боку, цей механізм виступає як критично важливий інструмент відновлення після відмови (Recovery Mechanism), оперативно перенаправляючи обробку завдань з несправного вузла на працездатні сусідні платформи. При формулюванні задач оптимізації, надійність повинна бути формалізована як кількісна метрика. Як правило, використовуються такі показники, як ймовірність відмови (Failure Probability) або коефіцієнт доступності (Availability), які є цільовими параметрами для максимізації. Досягнення високої надійності прямо впливає на вартість та життєздатність архітектурного рішення.

Мобільність (Mobility) – це критерій, що визначає здатність системи забезпечувати безперервне та безшовне обслуговування Edge-пристроїв (таких як транспортні засоби або мобільні телефони) навіть при їхньому фізичному переміщенні між зонами покриття різних Fog-вузлів. Для критичних застосувань, де необхідна постійна комунікація в реальному часі, будь-який розрив у з'єднанні або збільшення затримки під час переходу є неприпустимим.

Задоволення цієї вимоги базується на механізмі безшовної передачі обслуговування (Handover). У Fog-архітектурі Handover є більш складним, ніж у традиційних мережах, оскільки він часто вимагає міграції обчислювального стану завдання, що виконується. Це забезпечує сесійну неперервність. Успішне виконання цієї операції тісно корелює з необхідністю географічної обізнаності (Geo-awareness) платформи.

Ефективна реалізація Geo-awareness вимагає, щоб кожен Fog-вузол постійно підтримував актуальну інформацію про розташування Edge-пристроїв та поточний стан своїх сусідів. Цей обмін даними забезпечується механізмами горизонтальної комунікації Fog-мережі. З точки зору оптимізації, мобільність підлягає кількісному вимірюванню через такі метрики, як частота передачі обслуговування (Handover Rate) та час затримки при переключенні (Handover

Latency). Мінімальний час затримки під час Handover є обов'язковим критерієм, який має бути інтегрований до цільової функції, оскільки саме він безпосередньо визначає гарантовану якість обслуговування (QoS) мобільних користувачів.

Енергоефективність (Energy Efficiency) є важливою нефункціональною вимогою, яка розглядається насамперед з позиції кінцевих пристроїв, розташованих на рівні Edge. Значна частина пристроїв Інтернету речей (IoT), сенсорів та мобільних шлюзів живиться від обмежених джерел (батарей). Для цих пристроїв час автономної роботи є пріоритетом, а виконання ресурсоємних обчислювальних завдань локально може призвести до швидкого і неприпустимого вичерпання енергії.

Вимога до енергоефективності є головною рушійною силою для впровадження механізму Task Offloading (перенесення обчислень), оскільки вона обґрунтовує необхідність міграції завдання. Рішення про перенесення завдання на більш потужний Fog-вузол є виправданим лише за умови, що сумарні енергетичні витрати, пов'язані з передачею даних через мережу, є значно меншими, ніж енергетичні витрати на його виконання безпосередньо на Edge-пристрої. Іншими словами, Fog-мережа має забезпечити чисте енергозбереження для кінцевого користувача.

У контексті системної оптимізації, Енергоефективність формалізується як ключова метрика. Її кількісне вимірювання базується на розрахунку сумарного споживання енергії у всій системі або на оцінці заощадженої енергії на периферії. У цільовій функції оптимізації цей критерій фігурує як параметр, який необхідно максимізувати (заощадження) або мінімізувати (споживання).

Безпека (Security) та **Конфіденційність (Privacy)** є критично важливими нефункціональними вимогами, які набувають особливої складності в архітектурі Fog Computing. На відміну від централізованих Cloud-рішень, де забезпечення захисту може бути сфокусоване на певній кількості високозахищених та фізично контрольованих дата-центрів, Fog-мережа є сильно розподіленою та гетерогенною. Це призводить до значного збільшення поверхні атаки (attack

surface), оскільки обробка та зберігання даних відбувається на великій кількості фізично різномірних вузлів, розташованих поза межами традиційного периметру безпеки.

Основні виклики безпеки Fog-мережі включають:

- Захист даних у транзиті. Інформація постійно переміщується між рівнями Edge, Fog та Cloud, а також горизонтально між Fog-вузлами. Це вимагає використання надійних механізмів шифрування для гарантування цілісності та конфіденційності даних.
- Автентифікація та авторизація. Необхідність швидкої та надійної ідентифікації Edge-пристроїв та користувачів у динамічному та мобільному середовищі.
- Ізоляція ресурсів. На Fog-вузлах можуть одночасно виконуватися завдання від різних користувачів (принцип мультиорендності). Вимога ізоляції, що забезпечується технологіями віртуалізації (контейнерами або VM), є критичною для запобігання несанкціонованому доступу до даних.

Вимога до Конфіденційності є не менш важливою, оскільки Fog-вузли часто обробляють чутливі дані, що містять особисту інформацію або комерційну таємницю. З точки зору оптимізації, Безпека рідко є змінною для мінімізації чи максимізації. Натомість, вона виступає як обов'язкове обмеження для цільової функції. Ці обмеження можуть бути виражені через кількісні критерії, такі як мінімально необхідний рівень шифрування для каналів комунікації або гарантована частота успішних автентифікацій.

Багатоцільова оптимізація: Концепція компромісу та Парето-рішень

Попередній аналіз п'яти ключових нефункціональних вимог (Затримка, Надійність, Енергоефективність, Мобільність, Безпека) підтверджує, що ефективне управління ресурсами Fog-архітектури є складною багатоцільовою задачею. Фундаментальна складність полягає в тому, що ці вимоги не є незалежними; навпаки, вони часто конфліктують між собою, створюючи необхідність пошуку компромісного рішення (Trade-off).

Концепція компромісу (Trade-off) полягає у неможливості одночасного досягнення ідеальних показників за всіма критичними метриками. Наприклад, спроба мінімізувати Затримку до абсолютного мінімуму вимагає використання максимально доступних обчислювальних ресурсів (високопродуктивні процесори, додаткова пам'ять) та ігнорування їхнього завантаження. Це, в свою чергу, призводить до надмірного енергоспоживання і, як наслідок, знижує енергоефективність системи. Аналогічно, підвищення надійності вимагає створення надлишковості (дублювання Fog-вузлів), що прямо збільшує вартість архітектури та, знову ж таки, сумарне енергоспоживання. Концептуальна модель цього конфлікту, що демонструє нелінійну залежність між часом відгуку та інтенсивністю використання ресурсів Fog-інфраструктури, наведена на рис. 1.3.

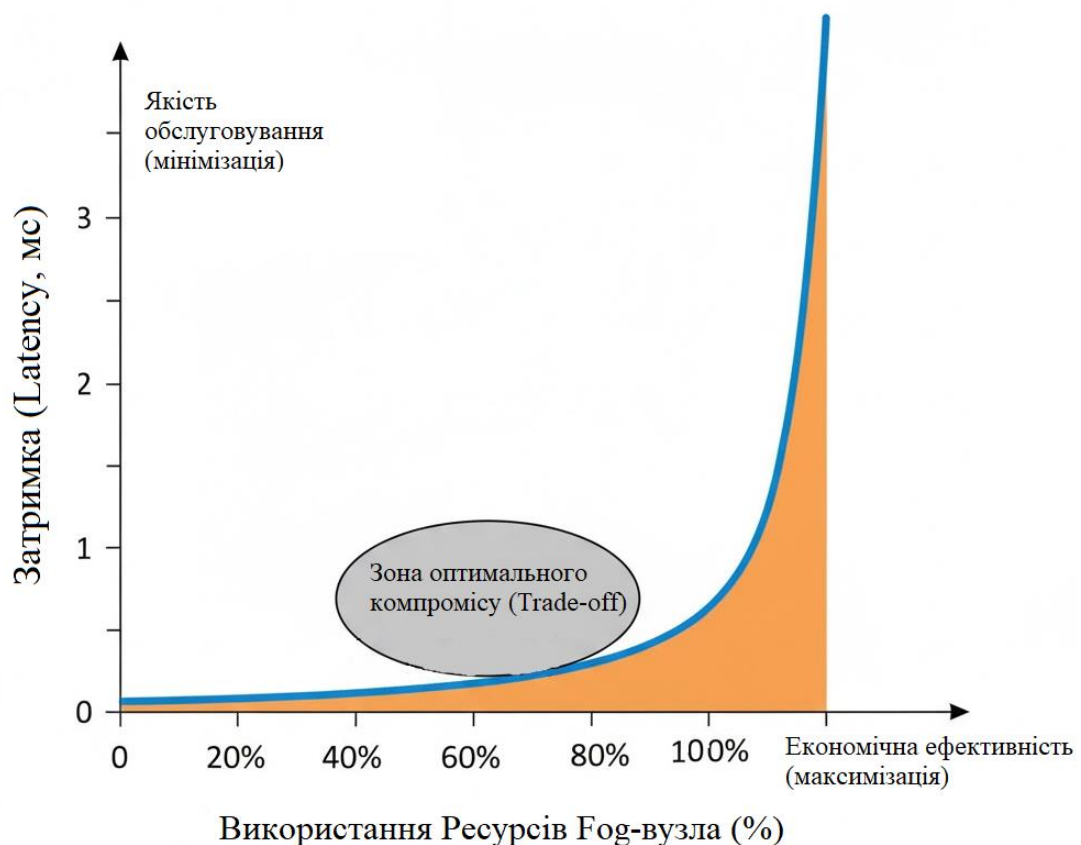


Рис. 1.3. Концептуальна діаграма компромісу (Trade-off) між затримкою та використанням ресурсів Fog-систем

Цей конфлікт чітко ілюструється на прикладі взаємозалежності між затримкою та використанням ресурсів Fog-вузла. Досягнення мінімальної затримки вимагає постійного підтримки низького рівня завантаження ресурсів, що є неефективним з економічної точки зору. З іншого боку, максимізація використання ресурсів призводить до перевантаження, що супроводжується різким зростанням затримки.

Тому, для управління Fog-системою потрібен не просто механізм Task Offloading або Load Balancing, а стратегія оптимізації, здатна враховувати цей конфлікт. Ця стратегія повинна об'єднувати всі критичні метрики у єдиний математичний вираз. Це досягається через визначення цільових критеріїв для оптимізації, де об'єднуються всі критичні метрики у єдиний математичний вираз. Як правило, мінімізується зважена сума затримки, енергоспоживання та вартості обслуговування.

Визначення цільових критеріїв для оптимізації як зваженої суми є лише першим кроком у моделюванні Fog-систем. Комплексна стратегія оптимізації повинна також враховувати природу жорстких обмежень (Hard Constraints), які не можуть бути порушені, незалежно від бажаного рівня оптимізації. Такі вимоги, як мінімальна надійність (Availability) та гарантований рівень безпеки, не можуть бути просто включені до цільової функції зі змінною вагою; вони є обов'язковими умовами (наприклад, ймовірність відмови Fog-вузла має бути нижчою за встановлений поріг). Успішне керування Fog-архітектурою вимагає, щоб Task Offloading та Load Balancing завжди функціонували в межах цих непереборних обмежень, перш ніж переходити до мінімізації або максимізації змінних метрик.

З огляду на внутрішній конфлікт між ключовими вимогами, жодне єдине рішення не може бути ідеальним для всіх метрик одночасно [17]. У цьому контексті, ціль полягає не в пошуку однієї "найкращої" точки, а у визначенні набору Парето-оптимальних рішень. Парето-оптимальним є таке рішення, при якому неможливо покращити жодну з цільових метрик (наприклад, знизити

затримку), не погіршивши при цьому іншу (наприклад, збільшивши енергоспоживання). Ця множина рішень формує Парето-Фронт, надаючи оператору можливість обрати оптимальний компроміс, виходячи з пріоритетів поточного сценарію застосування [18].

1.4 Сценарії застосування Fog Computing та вплив на мережеву інфраструктуру

Концепція Fog Computing як архітектурна відповідь на вимоги 5G/6G та IoT

Масове впровадження Інтернету речей (IoT), що генерує безпрецедентні обсяги даних, а також розгортання мереж п'ятого (5G) та шостого (6G) покоління, створили вимоги, які традиційна парадигма Хмарних обчислень (Cloud Computing) вже не може ефективно задовольнити [19]. Ключовими технологічними драйверами, що вимагали архітектурної зміни, стали потреба у зниженні затримки (Ultra-Low Latency) та забезпечення високої пропускну здатності (High Bandwidth) на локальному рівні.

Передавання всього потоку даних, зокрема критично важливих, до віддалених центральних дата-центрів Cloud, спричиняє неприпустимі затримки, які можуть досягати сотень мілісекунд. Такі показники є несумісними з вимогами застосувань реального або майже реального часу, таких як автономний транспорт, віддалена хірургія чи промислова автоматизація. У цьому контексті Fog Computing постає як необхідний проміжний шар, який доповнює Cloud та Edge, розміщуючи обчислювальні, зберігаючі та мережеві ресурси ближче до кінцевих користувачів та джерел даних.

Головна роль Fog полягає у забезпеченні локальної обробки даних з мінімальною затримкою, що є критично важливим для комунікаційних каналів. Зокрема, Fog-вузли здатні виконувати фільтрацію, агрегацію та попередню аналітику даних, значно зменшуючи обсяг інформації, яка потребує передачі до Cloud. Ця функція безпосередньо впливає на оптимізацію мережевої інфраструктури, знижуючи навантаження на магістральні канали зв'язку та

вивільняючи їх пропускну здатність для більш пріоритетних або ресурсомістких завдань. Таким чином, Fog Computing є не лише обчислювальною, але й мережевою парадигмою, що структурно підтримує децентралізацію та підвищення відмовостійкості систем у телекомунікаціях нового покоління.

Ключові сценарії впровадження Fog Computing в критичних індустріях

Архітектура Fog Computing, що забезпечує низьку затримку та локальну обробку даних, знайшла критичне застосування в тих секторах, де швидкість прийняття рішень та надійність системи є пріоритетом. Ці сценарії, які часто інтегруються з мережами 5G та пристроями IoT, найкраще ілюструють, чому проміжний обчислювальний шар є необхідним.

Автономний транспорт та системи V2X (Vehicle-to-Everything)

Одним із найбільш вимогливих до затримки застосувань є автономний транспорт. Системи зв'язку між транспортними засобами, інфраструктурою та пішоходами (V2X) потребують реакції на рівні мілісекунд для уникнення зіткнень та забезпечення безпеки руху. Якщо керування трафіком або екстрена обробка сенсорних даних від автомобіля покладається на віддалений Cloud, затримка стає фатальною. Fog-вузли, розміщені на перехрестях або вздовж магістралей (часто у вигляді базових станцій 5G або придорожніх пристроїв), здатні миттєво обробляти дані про рух, прогнозувати потенційно небезпечні ситуації та координувати маршрути [20]. Це дозволяє здійснювати локальне прийняття рішень щодо зміни швидкості або маршруту в критичних ситуаціях, підвищуючи безпеку та ефективність міського трафіку.

Промисловий Інтернет Речей (IIoT) та промислова автоматизація

В умовах Промислового Інтернету Речей (IIoT), де сотні чи тисячі сенсорів та виконавчих механізмів працюють у замкненому циклі, Fog Computing є основою для Промислової автоматизації (Industry 4.0). На виробничих лініях, які вимагають синхронізації обладнання з точністю до мікросекунд, затримка в десятки мілісекунд, характерна для Cloud, неприпустима. Fog-архітектура

дозволяє розміщувати обчислювальні ресурси безпосередньо в цеху або на заводському майданчику [21]. Це забезпечує моніторинг обладнання в реальному часі, прогнозування відмов (Predictive Maintenance) та контроль якості безпосередньо на місці виробництва. Крім того, Fog-вузли підвищують надійність (Resilience) системи: у випадку втрати зв'язку з центральним Cloud-сервером, локальний Fog-шар продовжує підтримувати критичні операції, гарантуючи безперервність виробництва.

Розумні енергомережі (Smart Grids) та управління енергетичною інфраструктурою

Управління сучасними розумними енергомережами (Smart Grids) вимагає двонаправленої комунікації та швидкої реакції на коливання навантаження та генерації (наприклад, від сонячних або вітрових електростанцій). Критично важливим є відновлення роботи мережі після збоїв та балансування енергії в розподілених підстанціях. Fog-вузли, інтегровані в мережеву інфраструктуру, можуть аналізувати дані з тисяч інтелектуальних лічильників та сенсорів підстанцій, швидко виявляти аномалії та ізолювати пошкоджені ділянки для запобігання каскадним відмовам. Це значно підвищує надійність постачання енергії та ефективність управління її споживанням.

Телемедицина та критичні медичні застосування

Fog-архітектура є незамінною в телемедицині, особливо для моніторингу пацієнтів у віддалених або домашніх умовах [22]. Обробка біометричних даних, таких як електрокардіограми (ЕКГ) чи показники життєдіяльності, вимагає миттєвої реакції. Розміщення Fog-вузлів у локальних клініках або навіть інтеграція їхніх функцій у домашні медичні пристрої дозволяє проводити первинний аналіз аномалій (наприклад, нерегулярний серцевий ритм) та негайно сповіщати екстрені служби. Це дозволяє оперативно вжити заходів і є критичним для забезпечення життєво важливої безпеки пацієнта, усуваючи ризики, пов'язані із затримкою передачі повного масиву даних до центрального Cloud.

Безпека та відеоаналітика в Розумних містах (Smart Cities)

У контексті Розумних міст Fog Computing застосовується для підвищення громадської безпеки та ефективності управління інфраструктурою. Відеокамери спостереження, що генерують величезні потоки даних, можуть бути підключені до Fog-вузлів, розміщених на комунікаційних шафах або вуличних ліхтарях. Тут відбувається локальна відеоаналітика — розпізнавання облич, виявлення підозрілих об'єктів або несанкціонованого скупчення людей. Обробка даних на місці економить пропускну здатність мережі, оскільки в Cloud передаються лише агреговані метадані або кадри з ідентифікованими подіями, а не цілі високоякісні відеопотоки. Це пришвидшує реакцію міських служб.

Розподілені мультимедіа та доповнена/віртуальна реальність (AR/VR)

Для застосувань Доповненої (AR) та Віртуальної (VR) Реальності, які є надзвичайно чутливими до затримки, Fog пропонує архітектурне рішення для розподіленого рендерингу та обробки [23]. Складні обчислення, необхідні для формування графіки високої якості, можуть бути перенесені з енергетично обмеженого мобільного пристрою (гарнітури чи смартфона) на найближчий Fog-вузол. Це дозволяє забезпечити користувачеві безшовний та гіпер-реалістичний інтерактивний досвід з мінімальною затримкою оновлення (Motion-to-Photon Latency), що є необхідною умовою для запобігання кіберзахворюванням та забезпечення високої якості обслуговування (QoS).

Ці приклади демонструють, що Fog Computing не просто прискорює обробку даних, а створює критично необхідну інтелектуальну децентралізацію, яка є фундаментом для забезпечення безпеки, стійкості та ефективності сучасних комунікаційних мереж та індустріальних систем.

Вплив Fog Computing на оптимізацію мережевого трафіку та пропускну здатність

Фундаментальною перевагою архітектури Fog Computing, окрім мінімізації затримки, є її прямий вплив на ефективність використання мережевих ресурсів та оптимізацію пропускну здатності. За своєю суттю, Fog-вузол

функціонує як інтелектуальний шлюз, який стратегічно управляє потоками даних між кінцевими пристроями (Edge) та центральною Хмарою (Cloud).

Механізм агрегації та фільтрації даних

Ключовим механізмом, що забезпечує оптимізацію, є агрегація та фільтрація даних (Data Aggregation and Filtering), які виконуються безпосередньо на Fog-рівні. У традиційній моделі IoT, сенсорні мережі передають величезні обсяги сирих, часто надлишкових, даних (наприклад, безперервні показання температури або повний відеопотік) магістральними каналами до Cloud. Це швидко призводить до перевантаження мережі та підвищення операційних витрат. Fog-вузол перехоплює ці потоки даних і проводить їх попередню обробку:

- Фільтрація. Відкидання нерелевантних або дубльованих даних. Наприклад, передача показань сенсора лише у випадку, якщо показник перевищує встановлений поріг (виявлення аномалії).
- Агрегація. Об'єднання множинних вимірювань у стислий, узагальнений пакет. Наприклад, замість 1000 окремих температурних записів за хвилину, у Cloud надсилається лише одне середнє або медіанне значення.
- Локальне прийняття рішень. Обчислення, які не потребують глобального контексту, залишаються на Fog-рівні, ініціюючи локальні дії (наприклад, увімкнення вентилятора), і не задіюють ресурси магістральної мережі взагалі.

Цей процес забезпечує, що до Cloud-сервісів надходять лише очищені та цінні метадані або результати аналізу, а не сирі масиви інформації.

Зменшення навантаження на магістральні канали

Внаслідок ефективної агрегації та фільтрації, обсяг трафіку, який надходить до центральної мережі, зменшується в десятки й сотні разів. Це має прямий наслідок для пропускної здатності магістральних каналів зв'язку: вона вивільняється, дозволяючи мережі ефективно обслуговувати дійсно ресурсомісткі та глобально важливі завдання.

Крім того, Fog Computing полегшує так зване "Load Offloading" (розвантаження) на рівні мережі. Замість того, щоб встановлювати постійне з'єднання "Edge-to-Cloud" для кожного пристрою, Fog-вузол виступає як точка концентрації, яка підтримує короткоживучі локальні з'єднання та управляє їх передачею на вищий рівень лише за необхідності. Це підвищує енергоефективність кінцевих пристроїв та подовжує їхній час автономної роботи, водночас гарантуючи стійкість мережевої інфраструктури до пікових навантажень.

Принципи інтеграції Fog Computing в архітектуру електронних комунікацій (5G/B5G Core)

Інтеграція Fog Computing в архітектуру електронних комунікаційних мереж є важливою для реалізації повного потенціалу мереж 5G та майбутніх Beyond 5G (B5G) / 6G [24]. Fog-шар не є окремою ізольованою системою, а виступає як розширення функціонального ядра (Core Network), переносячи його обчислювальні та керуючі можливості до периферії .

Підтримка мобільності та Geo-Awareness

Одним із ключових принципів інтеграції є ефективне керування мобільністю та забезпечення географічної обізнаності (Geo-Awareness). Fog-вузли, знаючи про своє точне фізичне розташування та географічну близькість до Edge-пристроїв (наприклад, дронів або автомобілів), можуть оперативно координувати процес безшовного Handover (передачі обслуговування). У мобільних застосуваннях критично важливо, щоб обчислювальний процес (наприклад, аналіз даних дрона) не переривався, коли пристрій переміщується між зонами обслуговування сусідніх вузлів. Для цього архітектура Fog використовує горизонтальні комунікаційні канали для обміну інформацією про стан, навантаження та контекст мобільного пристрою, що забезпечує безперервність сервісу (Service Continuity) та підтримку ультра-низької затримки.

Створення єдиного континууму обчислень через інтеграцію з MEC

Інтеграція Fog реалізується через розширення функціоналу мережевого керування 5G, зокрема через взаємодію з MEC (Multi-access Edge Computing). На відміну від MEC, який може бути вузько сфокусований на мережі радіодоступу (RAN), Fog пропонує більш універсальну та агреговану платформу, здатну обслуговувати різні технології доступу (Wi-Fi, 5G, фіксовані мережі). Таким чином, Fog-вузли використовуються як децентралізовані хостинг-платформи для розгортання MEC-застосунків. Ця архітектурна синергія створює єдиний і безшовний континуум обчислень від централізованого Cloud до периферійного Edge, який дозволяє оператору мережі динамічно призначати обчислювальні ресурси відповідно до вимог до затримки, пропускної здатності та безпеки кожного конкретного сервісу.

Динамічне управління мережевими функціями

Fog Computing дозволяє реалізувати динамічне управління мережевими функціями (NFV/SDN). Вузол Fog може розміщувати віртуалізовані мережеві функції (наприклад, функції маршрутизації або безпеки) та гнучко розподіляти їх, керуючись показниками локального трафіку. Це значно підвищує адаптивність мережі до мінливих умов навантаження та дозволяє оперативно впроваджувати нові комунікаційні сервіси на вимогу, не чекаючи централізованого оновлення ядра мережі.

Висновки

На основі проведеного ґрунтовного аналізу теоретичних засад та архітектурної концепції туманних обчислень (Fog Computing) можна сказати, що ця парадигма сформувалася як безпосередня архітектурна відповідь на технологічні виклики, спричинені стрімким розвитком Інтернету речей (IoT) та мереж 5G/6G, де критично важливі застосування вимагають мінімальної затримки (Ultra-Low Latency) та локальної обробки даних. Обґрунтовано еволюційний перехід від традиційної моделі Cloud Computing та ресурсно-

обмеженої Edge Computing до ієрархічної інфраструктури Fog, яка займає проміжний рівень у континуумі обчислень.

Встановлено, що Fog Computing забезпечує децентралізацію обчислювальних, зберігаючих та мережових ресурсів ближче до джерел даних, що дозволяє уникнути перевантаження магістральних каналів зв'язку та досягти необхідних показників швидкості реакції. Фундаментальною відмінністю Fog є його здатність підтримувати горизонтальну комунікацію (Fog↔Fog), яка є ключовою для координації ресурсів та забезпечення безперервності сервісу (Service Continuity) в умовах мобільності пристроїв. Це дозволяє Fog-вузлам виконувати агрегацію та фільтрацію даних, суттєво зменшуючи обсяг сирової інформації, що передається до централізованої Хмари. Досліджені ключові сценарії застосування — від автономного транспорту до промислового Інтернету Речей (IIoT) — підтверджують, що Fog-шар є необхідним фундаментом для реалізації кібер-фізичних систем, які вимагають високої надійності та прийняття рішень у майже реальному часі.

Визначені ключові вимоги — мінімальна затримка (Latency), висока надійність (Reliability) та підтримка мобільності (Mobility) — окреслюють складність управління ресурсами у розподіленій Fog-архітектурі. Ефективне функціонування системи значною мірою залежить від механізму перенесення обчислювальних завдань (Task Offloading), який є критичним для оптимізації як часу відгуку, так і енергоефективності кінцевих пристроїв. Це трифазний процес, який вимагає комплексного підходу до прийняття рішень на основі кількісних метрик та поточного стану мережі.

Інтеграція Fog Computing в архітектуру електронних комунікацій (зокрема, 5G/MEC (Mobile Edge Computing)) є необхідною для створення єдиного та безшовного континууму обчислень від Cloud до Edge, забезпечуючи Geo-Awareness для ефективного керування сервісами та мобільністю. При цьому виявлено, що управління ресурсами Fog-мережі вимагає адаптивного підходу до Task Offloading, оскільки ключові метрики (наприклад, Затримка (Latency) та

Енергоспоживання) часто перебувають у внутрішньому конфлікті (Trade-off). Це вимагає розробки комплексної стратегії балансування. Відповідно, метою дослідження є розробка та оцінка ефективності адаптивної моделі управління Task Offloading у МЕС-мережах, яка використовує принципи багатоцільової оптимізації для забезпечення оптимального компромісу між продуктивністю та енергоефективністю, дотримуючись жорстких обмежень QoS, надійності та безпеки.

РОЗДІЛ 2

МЕТОДИ ТА МОДЕЛІ ВПРОВАДЖЕННЯ ТУМАННИХ ОБЧИСЛЕНЬ В ЕЛЕКТРОННИХ КОМУНІКАЦІЯХ

2.1. Моделі розміщення, топології та принципи мережевої взаємодії Fog-вузлів

Моделі фізичного розміщення Fog-вузлів у мережевому континуумі

Успіх впровадження туманних обчислень у мережеву інфраструктуру детермінований насамперед стратегічним позиціонуванням обчислювальних ресурсів. Це архітектурне рішення є критично важливим, оскільки безпосередньо впливає на виконання ключової нефункціональної вимоги — мінімізації затримки (Latency), яка є головним драйвером Fog-парадигми. Фактична відстань, яку повинні подолати дані від кінцевого пристрою до обчислювального вузла, прямо корелює з часом відгуку системи.

Fog Computing функціонує як невід'ємна частина ієрархічного обчислювального континууму (Edge-Fog-Cloud). Якщо Edge-рівень займається збором даних, а Cloud-рівень — глобальною аналітикою, то Fog-рівень забезпечує обробку даних у квазі-реальному часі з низькою цільовою затримкою (переважно 5–50 мс). Для досягнення цієї цільової метрики, яка є значно нижчою, ніж на Cloud-рівні, Fog-функції повинні бути розгорнуті у стратегічно визначених точках доступу. Реалізація Fog-рівня в межах цього континууму вимагає застосування специфічних моделей фізичної локалізації для оптимального балансу між продуктивністю та ресурсною ефективністю.

Модель Near-Edge Placement

Модель Near-Edge Placement (Розміщення біля периферії) є найбільш агресивною формою децентралізації обчислень, що є необхідною для реалізації критично важливих застосувань мереж 5G/6G. У цій моделі Fog-функції розміщуються максимально близько до кінцевих пристроїв, з яких надходять дані, що мінімізує фізичну відстань передачі та, відповідно, час проходження

сигналу. Near-Edge Placement передбачає інтеграцію обчислювальних ресурсів Fog безпосередньо в активне мережеве обладнання периферії або поруч із ним.

Така інтеграція часто реалізується шляхом використання існуючої інфраструктури, що має необхідну обчислювальну потужність. Функції Fog можуть бути розгорнуті у вигляді віртуалізованих контейнерів або програмних модулів на комутаторах і маршрутизаторах, особливо тих, що оснащені підтримкою програмно-конфігурованих мереж (SDN). Використовуючи SDN-контролери, які забезпечують централізоване управління мережею, мережеві пристрої перетворюються на інтелектуальні Fog-вузли, здатні виконувати локальну обробку трафіку, фільтрацію та кешування, значно розширюючи свої традиційні функції.

Проте, найбільш поширеним і потужним варіантом є інтеграція Fog-функціоналу на рівні базових станцій (gNB) мереж 5G. Тут інтеграція відбувається на базі спеціалізованих MEC-серверів (Multi-access Edge Computing), які фізично розташовані на об'єктах радіодоступу. Такий підхід забезпечує найкоротший шлях доступу до даних радіодоступу, що є критичним для застосувань, котрі пов'язані з мобільністю та вимагають мінімальної затримки, оскільки обчислення виконуються безпосередньо в зоні покриття.

Головна перевага Near-Edge Placement полягає у досягненні найменшої можливої затримки (типові показники 1–5 мс). Це дозволяє підтримувати застосування реального часу, такі як автономне керування транспортними засобами, високоточне промислове управління та критична телемедицина. Крім того, таке розміщення забезпечує найкращу підтримку мобільності (Mobility Support), оскільки Fog-вузли, інтегровані в мережу радіодоступу, здатні оперативно координувати процес безшовного Handover (передачі обслуговування) між сусідніми зонами. Локалізація обчислень також підвищує конфіденційність, оскільки чутливі дані обробляються та зберігаються локально, не виходячи за межі периферії.

Незважаючи на значні переваги в продуктивності, модель Near-Edge Placement стикається з низкою технічних та економічних обмежень, які ускладнюють її широке впровадження. Найбільш суттєвим технічним викликом є обмежена обчислювальна потужність, оскільки обладнання, що розміщується на периферії (базові станції, маршрутизатори), має значно менші ресурси центрального процесора (CPU), оперативної пам'яті (RAM) та сховища, ніж централізовані дата-центри. Цей дефіцит ресурсів прямо обмежує складність оброблюваних завдань і вимагає постійного, динамічного управління розподілом ресурсів та моніторингу навантаження. З економічної точки зору, розміщення великої кількості обчислювальних вузлів на широкій географічній території вимагає значних капіталовкладень у спеціалізоване обладнання (з підтримкою MEC/SDN) та призводить до високих операційних витрат через необхідність децентралізованого обслуговування. Додатково, інженерною проблемою є управління енергоспоживанням та охолодженням: функціонування Fog-вузлів в умовах обмеженого фізичного простору без централізованих систем клімат-контролю ускладнює ефективне відведення тепла, роблячи енергоспоживання однією з критичних цільових метрик для оптимізації.

Модель Aggregation Point Placement

На відміну від моделі Near-Edge Placement, де обчислювальна потужність сильно обмежена, модель Aggregation Point Placement (Розміщення в точках агрегації) пропонує проміжний компроміс між продуктивністю периферії та ресурсами центральної Хмари. Fog-вузли в цій моделі розташовуються не безпосередньо на кінцевих об'єктах або базових станціях, а в географічно центральних точках концентрації трафіку. Це можуть бути спеціалізовані шлюзи, великі комунікаційні шафи або міні-дата-центри на рівні району чи промислового об'єкта, які обслуговують цілий кластер пристроїв. Таке розміщення забезпечує вищу обчислювальну потужність, ніж на Near-Edge рівні, при цьому зберігаючи відносно невелику затримку (типово 10–50 мс), що є

прийнятною для багатьох застосувань, які не вимагають ультра-низької затримки.

Ключова перевага розміщення Fog-вузлів у точках агрегації полягає у значній економії пропускної здатності магістральних каналів зв'язку. Fog-вузол у цій точці виступає як інтелектуальний шлюз, який проводить агрегацію та фільтрацію даних, що надходять від великої кількості Edge-пристроїв [25]. Замість того, щоб передавати сирі масиви інформації до центральної Хмари, вузол обробляє дані локально, виявляє аномалії та надсилає далі лише очищені, стиснені метадані або результати попереднього аналізу. Такий підхід значно знижує навантаження на мережеве ядро та зменшує операційні витрати, пов'язані з передачею великих обсягів трафіку.

Ця модель є особливо актуальною для систем, які оперують у територіально розподілених та складних середовищах, таких як розумні енергетичні мережі (Smart Grids), системи моніторингу довкілля або великі виробничі комплекси. Розміщення Fog-вузлів у центральних точках збору даних дозволяє проводити ресурсомісткі обчислення, наприклад, складні алгоритми Machine Learning для прогнозування відмов обладнання (Predictive Maintenance) або глибинний аналіз відеопотоків. Хоча затримка тут вища, ніж у Near-Edge, більша обчислювальна потужність дозволяє ефективніше виконувати ці складні аналітичні завдання, забезпечуючи надійне та своєчасне прийняття рішень на локальному рівні.

Моделі топології Fog-мереж

Моделі топології Fog-мереж визначають логічну організацію взаємодії між Fog-вузлами, центральними хмарними центрами та периферійними пристроями. Ці моделі задають ключові принципи розподілу обчислювальних, комунікаційних і зберігальних ресурсів у системі, а також безпосередньо впливають на рівень децентралізації та автономності обробки даних. Для реалізації Fog-архітектури виділяють три основні топологічні схеми, що розрізняються логікою організації потоків даних і ступенем залежності від

хмарної інфраструктури: Централізовану, Розподілену (Peer-to-Peer) та Ієрархічну моделі.

Централізована модель розглядає Fog Computing не як самостійний рівень обчислень, а як функціональне розширення хмарної інфраструктури. У цій архітектурі туманні вузли виконують роль проміжного, підпорядкованого шару, який доповнює Хмару локальною обробкою даних, залишаючись при цьому жорстко інтегрованим із центральним ядром. Вузли Fog отримують дані від Edge-пристроїв, виконують попередню фільтрацію, стиснення або просту аналітику, після чого передають підготовлену інформацію до центрального дата-центру. Технічно така архітектура характеризується чіткою вертикальною структурою — Fog підпорядковується Cloud, де Fog-вузол не приймає самостійних рішень, а лише оптимізує трафік для передачі на вищий рівень.

Основною перевагою централізованої моделі є простота розгортання та централізоване управління безпекою, уніфікованими політиками зберігання та сумісність із наявними хмарними сервісами. Це ефективно, коли система вимагає єдиного контролю та глобальної аналітики. Проте головним архітектурним недоліком є обмежена автономність: Fog-вузли діють лише як пасивні допоміжні точки обробки. У разі відмови магістрального зв'язку або недоступності центрального Cloud-ядра, туманні вузли втрачають можливість підтримувати повний обсяг сервісів. Це створює високий ризик втрати обслуговування (Service Loss) і значно знижує стійкість системи, особливо у критичних застосуваннях.

Розподілена модель (Peer-to-Peer Fog) реалізує принцип «Fog взаємодіє з Fog», наголошуючи на горизонтальній комунікації між туманними вузлами без обов'язкового залучення центрального Cloud-ядра. У цій архітектурі кожен вузол має власні обчислювальні та аналітичні ресурси і може безпосередньо співпрацювати з іншими вузлами в межах одного сегмента мережі. Це дозволяє забезпечити динамічне балансування навантаження, розподіл завдань, обмін проміжними результатами та спільне прийняття рішень на локальному рівні, що є критично важливим для сценаріїв, де потрібна висока швидкість реакції та

відсутність єдиної точки відмови. Така модель є особливо ефективною для підтримки мобільності (Mobility Support), оскільки вона дозволяє туманним вузлам утворювати локальні кластери, які спільно обмінюються контекстною інформацією, забезпечуючи безшовну передачу обслуговування (Handover) між суміжними зонами покриття.

З технічного погляду, Peer-to-Peer Fog забезпечує найвищу автономність та найбільшу стійкість до відмов, адже вихід з ладу одного вузла не паралізує роботу всього кластера. Проте, така децентралізація вимагає впровадження складних протоколів синхронізації, управління станом і захисту даних, оскільки відсутній центральний координатор, який би гарантував узгодженість дій. Це значно підвищує складність мережевої архітектури. Незважаючи на ці вимоги, дана модель є базовою для високоавтономних систем, таких як системи керування автономними транспортними засобами або промислові IoT-мережі, де затримка, що генерується Хмарою, є неприпустимою.

Ієрархічна модель є найбільш повною та адаптивною топологією, що передбачає багаторівневу організацію обчислень у єдиному континуумі Edge–Fog–Cloud. Ключова відмінність від Централізованої моделі полягає в тому, що тут Fog виступає самостійним проміжним рівнем управління, а не просто пасивним посередником. Це дозволяє Fog-вузлам приймати незалежні рішення, тимчасово працювати автономно, координувати роботу Edge-пристроїв та взаємодіяти з Cloud-рівнем лише тоді, коли потрібен довгостроковий аналіз, моделювання або глобальна координація. Така архітектура є необхідною для забезпечення Geo-Awareness та ефективного керування сервісами в сучасних мережах електронних комунікацій.

Основні переваги ієрархічної топології полягають у гнучкості та масштабованості системи. Автономність Fog-рівня забезпечує двосторонній обмін даними і наявність локальних контурів зворотного зв'язку, що дозволяє швидко та ефективно реагувати на зміни стану мережі без залучення центрального ядра. Це значно підвищує надійність системи та дозволяє точно

розподіляти обчислювальні завдання відповідно до їхніх вимог до затримки та ресурсів. Завдяки цьому, Ієрархічна модель вважається основою для створення найбільш складних та стійких екосистем, які ефективно обслуговують критичні та некритичні застосування.

На практиці найбільш надійною та гнучкою архітектурою є гібридна Топологія, яка логічно інтегрує основні принципи як Ієрархічної (вертикальне керування), так і Розподіленої (горизонтальна координація) моделей. Цей підхід є необхідним для сучасних великомасштабних систем, які не можуть покладатися лише на суворий контроль із Хмари чи суто децентралізовану P2P-координацію. Гібридна Топологія зберігає вертикальну структуру для виконання глобальних завдань, як-от довгострокове зберігання даних, оновлення моделей машинного навчання та глобальне управління безпекою. Водночас вона активно використовує горизонтальну комунікацію (Fog-Fog) для високошвидкісної локальної координації, забезпечення стійкості до відмов і безшовного управління мобільністю.

Впровадження гібридної топології висуває суворі архітектурні вимоги до системи. Зокрема, вона повинна забезпечувати наскрізну надійність (End-to-End Reliability), гарантуючи безперервність надання сервісів навіть при тимчасовій втраті зв'язку з Cloud-ядром. Також критично важливим є механізм динамічного балансування навантаження (Dynamic Load Balancing), що дозволяє ефективно перерозподіляти ресурсомісткі завдання між сусідніми Fog-вузлами. Ці вимоги, безпосередньо пов'язані з необхідністю максимальної оптимізації використання дефіцитних обчислювальних ресурсів та мінімізації затримки, стають основою для розробки алгоритмів перенесення обчислювальних завдань (Task Offloading) та оркестрації

Принципи та протоколи мережевої взаємодії, що підтримують архітектуру Fog

Ефективність, надійність та масштабованість Fog-архітектури визначаються вибором та реалізацією спеціалізованих мережевих протоколів, необхідних для

підтримки її логічної топології. Ці протоколи забезпечують узгодженість дій і надійну взаємодію між усіма рівнями системи. У континуумі туманних обчислень комунікаційні механізми традиційно класифікуються за напрямом обміну даними: вертикальна комунікація (Edge↔Fog↔Cloud), та горизонтальна комунікація (Fog↔Fog). Крім того, функціональність Fog-рівня тісно пов'язана з архітектурою мереж п'ятого покоління (5G) і вимагає протокольної інтеграції з платформами Multi-Access Edge Computing (MEC) для досягнення цільової ультра-низької затримки.

Протоколи для вертикального зв'язку (Edge ↔ Fog ↔ Cloud)

Вертикальна взаємодія, що визначає ієрархічний потік даних і завдань, критично важлива для збору даних з периферії та доставки керувальних команд у зворотному напрямку. Для кожного сегмента вертикального шляху використовуються оптимізовані протоколи:

1. Edge ↔ Fog (Низькорівневий зв'язок). Для цього сегмента, що характеризується обмеженими ресурсами, низьким енергоспоживанням і можливістю нестабільного зв'язку, застосовуються легковагові, орієнтовані на IoT-середовища протоколи:
 - MQTT (Message Queuing Telemetry Transport). Це стандартизований, орієнтований на повідомлення протокол на основі парадигми публікації/підписки (Publish/Subscribe). Він використовує компактний бінарний формат повідомлень та базується на TCP, що дозволяє йому забезпечувати надійність доставки (за допомогою механізмів QoS). MQTT є ідеальним для масової передачі телеметричних даних (Edge → Fog) та отримання команд (Fog → Edge).
 - CoAP (Constrained Application Protocol). Являє собою полегшений аналог HTTP, що працює переважно на основі UDP для зниження накладних витрат. CoAP забезпечує модель «запит-відповідь» (Request/Response) та спеціально розроблений для обмежених

пристроїв (Constrained Devices), що робить його придатним для двонаправленої взаємодії в середовищах з низьким рівнем потужності.

2. Fog ↔ Cloud (Високорівневий зв'язок). На цьому етапі вимоги зміщуються до високої пропускної здатності, ефективної серіалізації та безпеки:

- gRPC (gRPC Remote Procedure Call). Це високопродуктивний RPC-фреймворк, заснований на протоколі HTTP/2. Він використовує двійкову серіалізацію (Protocol Buffers), що забезпечує низьку латентність та мінімізує накладні витрати на передачу великих обсягів структурованих даних. gRPC широко використовується для взаємодії мікросервісів та ефективного двостороннього стрімінгу між Fog-вузлами та Cloud-платформами.
- HTTPS/RESTful API. Залишається стандартним вибором для безпечної передачі агрегованих даних, метаданих і виконання керувальних API-запитів між Fog і Cloud.

Протоколи для горизонтальної комунікації (Fog ↔ Fog)

Горизонтальна комунікація є фундаментальною для реалізації розподіленої та гібридної топологій і критично важлива для автономного функціонування Fog-рівня. Вона дозволяє сусіднім вузлам обмінюватися контекстною інформацією, необхідною для балансування навантаження (Load Balancing) і забезпечення безшовного перенесення обслуговування (Handover) мобільних користувачів.

Ключовими протоколами для горизонтальної взаємодії є:

- DDS (Data Distribution Service). Це високопродуктивне проміжне програмне забезпечення (middleware) реального часу, яке використовує модель Data-Centric Publish/Subscribe (DCPS). DDS дозволяє Fog-вузлам автоматично виявляти один одного та обмінюватися гарантованими потоками даних у режимі peer-to-peer без єдиного центрального брокера. Це забезпечує наднизьку затримку та високу синхронізацію, що є

критичним для систем, де вимагається колективне прийняття рішень (наприклад, у промислових IoT-системах чи автономному транспорті).

- ZeroMQ (Zero Message Queue): Це бібліотека, що забезпечує високошвидкісну асинхронну взаємодію між вузлами Fog через сокетну інтерфейсну модель. ZeroMQ підтримує різні шаблони комунікації (push–pull, request–reply, pub–sub), що дозволяє реалізувати гнучкі схеми горизонтальної кооперації та ефективне використання ресурсів.

Інтеграція Fog Computing з архітектурою 5G/MEC

Інтеграція туманних обчислень із архітектурами мобільного зв'язку п'ятого покоління (5G) та технологією Multi-Access Edge Computing (MEC) є необхідною для досягнення цільової ультра-низької затримки (1–5 мс). MEC-платформа виступає як фізичний хост для розгортання віртуалізованих Fog-функцій. MEC-сервери розташовуються безпосередньо на об'єктах мережі радіодоступу (RAN), зокрема на базових станціях (gNB), що забезпечує найкоротший шлях до Edge-пристроїв.

Така інтеграція передбачає тісну взаємодію MEC-платформи (де розміщуються Fog-сервіси) з основними мережевими функціями 5G Core, особливо для управління трафіком:

- Взаємодія з UPF (User Plane Function). MEC-платформа використовує функціонал UPF для маршрутизації трафіку та локального виходу (Local Breakout). Це дозволяє трафіку, призначеному для локальної обробки (наприклад, відеоаналітика), відхилитися від центрального ядра мережі та оброблятися на Fog-рівні. Таким чином, Fog-вузол/MEC-сервер виконує функцію сервісного ланцюжка (Service Function Chaining), забезпечуючи застосуванням необхідну низьку затримку.
- Взаємодія з SMF (Session Management Function). Fog-вузли взаємодіють із функціями управління сесіями (SMF, а також PCF — Policy Control Function) для динамічного запиту ресурсів, встановлення параметрів QoS та управління політиками трафіку для конкретного застосування. Це

гарантує, що критично важливі Fog-сервіси завжди отримують пріоритетне обслуговування та необхідну смугу пропускання.

2.2. Методи управління завданнями (Task Offloading) та динамічного балансування навантаження

Концептуальна модель Task Offloading та оптимізаційні цілі

Управління обчислювальними ресурсами у Fog-середовищі реалізується через механізм перенесення обчислювальних завдань (Task Offloading). Цей механізм є прямим продовженням архітектурного вибору, що дозволяє стратегічно розміщувати навантаження на найбільш придатному рівні континууму (Edge, Fog, Cloud). Рішення про офлоудинг залежить від обсягу переданого завдання: воно може бути бінарним (повним), коли вся задача мігрує на зовнішній вузол, або частковим (сегментованим), коли задача поділяється на підзадачі, що виконуються паралельно на різних архітектурних рівнях. Така сегментація забезпечує необхідну гнучкість системи та є основою для адаптивного розподілу навантаження.

Концептуальна модель офлоудингу, успадковуючи трифазний процес (вибір рішення, вибір вузла, виконання), розширюється за рахунок трьох основних шляхів передачі: Edge $\rightarrow$ Fog (для низької затримки), Fog $\rightarrow$ Cloud (для масштабованої обробки) та Edge $\rightarrow$ Edge (P2P) (для локальної кооперації). Ефективність офлоудингу, як комплексного процесу, визначається коректністю обраної стратегії розміщення, яка формує багатоцільову задачу оптимізації.

Ключові цілі оптимізації управління завданнями, які часто перебувають у внутрішньому конфлікті (Trade-off), включають: мінімізацію затримки (Latency Minimization), мінімізацію енергоспоживання (Energy Consumption Minimization) та максимізацію пропускну здатності (Throughput Maximization). Успішне вирішення цієї задачі вимагає комплексного підходу, оскільки жодне єдине рішення не може бути ідеальним для всіх метрик одночасно.

Перед початком оптимізації система повинна задовольнити жорсткі обмеження (Hard Constraints), що диктуються умовами Quality of Service (QoS)

та Service Level Agreement (SLA) (наприклад, гарантований час відгуку, мінімальна надійність). Для забезпечення їх виконання застосовуються базові механізми, такі як пріоритезація завдань та Admission Control. Рішення про офлоудинг приймається на основі кількісних метрик, які формують поточний контекст системи. До них належать: обчислювальна складність завдання (обсяг необхідних інструкцій), динамічний стан каналу зв'язку (доступна пропускна здатність) та доступні ресурси Fog-вузла (завантаження CPU і пам'яті).

Методи планування та динамічного балансування навантаження (Load Balancing)

Балансування навантаження (Load Balancing) є наступним критичним етапом після визначення шляху офлоудингу. Його метою є забезпечення оптимального використання ресурсів, мінімізація часу відповіді та запобігання перевантаженню окремих вузлів шляхом розподілу обчислювальних завдань між кількома доступними Fog-вузлами.

У найпростішій формі використовуються статичні алгоритми, які не враховують поточний стан системи. До них належать: Round-Robin (циклічний розподіл), Least Loaded (LL) (вибір вузла з найменшим поточним навантаженням) та алгоритми Min-Min/Max-Min (планування на основі прогнозованого часу виконання). Хоча ці методи відзначаються низькими накладними витратами та простотою реалізації, їхня ефективність у динамічному та неоднорідному Fog-середовищі обмежена через нездатність адаптуватися до мінливого стану мережі, що призводить до зростання затримок і нераціонального використання ресурсів.

Для подолання цих недоліків у Fog-інфраструктурах домінують динамічні, обізнані про стан (state-aware) алгоритми, які постійно збирають та аналізують актуальні дані про використання ресурсів і стан мережі. Ці евристичні методи дозволяють приймати рішення про розподіл навантаження у реальному часі, динамічно перерозподіляючи завдання у відповідь на перевантаження або зміну мережових умов. Особливим напрямом є планування з урахуванням мобільності

(Mobility-Aware Scheduling). У сценаріях із рухомими пристроями критично важливим є забезпечення безперервності сервісу (Service Continuity). Для цього застосовується міграція завдань (Task Migration) — передача поточного стану виконання завдання з одного Fog-вузла на інший, коли, наприклад, прогнозується втрата зв'язку. Ефективна міграція, разом із механізмами реплікації завдань (Task Replication) для забезпечення відмовостійкості (Fault Tolerance), є необхідною для дотримання високих вимог SLA для критичних сервісів.

Найбільш перспективним напрямом розвитку є інтелектуальні методи балансування навантаження, засновані на машинному навчанні (ML). Ці підходи, часто класифіковані як евристики, що самонавчаються, здатні адаптуватися до складних багатофакторних середовищ, де традиційні алгоритми неефективні. Одним із ключових методів є Навчання з підкріпленням (Reinforcement Learning, RL). У цій парадигмі агент (Fog-контролер) взаємодіє з мережевим середовищем, приймаючи рішення про офлоудинг на основі політики, що постійно вдосконалюється. Метою є максимізація довгострокової винагороди, яка відображає інтегральні показники QoS (мінімізацію сумарної затримки та енергоспоживання). Такий підхід створює основу для когнітивного балансування навантаження — системи, яка є самоналаштовуваною, адаптивною і здатною до роботи у масштабованих та динамічних Fog-середовищах.

2.3. Концептуальна постановка задачі Task Offloading як багатоцільової оптимізації та вибір адаптивної стратегії

Концептуальна постановка задачі управління завданнями (Task Offloading) є центральним елементом управління ресурсами у гетерогенній архітектурі Fog-МЕС. На відміну від традиційного Cloud-центричного підходу, Fog-архітектура відкриває множинність потенційних шляхів, створюючи багатовимірний простір рішень. Основне завдання системи управління полягає у виборі такого рішення, яке забезпечить оптимальний баланс між конфліктуючими цілями, при цьому обов'язково дотримуючись жорстких вимог до якості обслуговування (QoS).

Простір рішень системи описується трьома основними елементами, які визначають, як буде оброблене кожне вхідне завдання. Першим є Вибір шляху виконання, який диктує, чи буде завдання оброблене локально на пристрої (Edge), перенесено на найближчий Fog-вузол (MEC), чи відправлено до віддаленої Хмари (Cloud). Другий елемент — це Вибір обчислювального вузла, що виникає, коли в зоні покриття доступно кілька Fog-вузлів. Третім, не менш критичним елементом, є Розподіл ресурсів, який визначає, яку частку обчислювальної потужності (наприклад, частота процесора) буде виділено для конкретного завдання. Усі ці елементи рішення перебувають у динамічній залежності від мінливих факторів середовища: мобільності пристроїв, інтенсивності трафіку та довільності обчислювальних вимог.

Ефективність обраної стратегії офлоудингу визначається здатністю системи одночасно оптимізувати дві головні, внутрішньо конфліктуючі цільові функції (Multi-Objective Optimization, MOO): мінімізація Затримки (пріоритетна для URLLC-сервісів) та мінімізація Енергоспоживання (критична для автономних Edge-пристроїв).

Математична постановка задачі Task Offloading як багатоцільової оптимізації

Успішне вирішення задачі управління обчислювальними завданнями (Task Offloading) в архітектурі Fog-обчислень, націлене на агроіндустріальний комплекс, зводиться до багатоцільової оптимізації (MOO). Це необхідно, оскільки ключові критерії — час затримки (Latency) та енергоспоживання — є конфліктуючими: мінімізація одного, як правило, призводить до зростання іншого.

Формалізація завдання починається з визначення вектора рішень x , який складається з бінарних змінних, що визначають, чи буде завдання k виконане локально, чи перенесене на Fog-вузол j .

$$x = \{ \alpha_k, z_{k,j} \}, \quad (2.1)$$

де:

$\alpha_k \in \{0,1\}$ — бінарна змінна офлоудингу. $\alpha_k = 0$ означає локальне виконання завдання k на Edge-пристрої; $\alpha_k = 1$ означає перенесення завдання на зовнішній вузол.

$z_{k,j} \in \{0,1\}$ — змінна вибору вузла. $z_{k,j} = 1$, якщо завдання k призначене для виконання на Fog-вузлі j або на Cloud-рівні ($j=M+1$), за умови, що $\alpha_k = 1$.

Загальна мета системи описується функцією $F(x)$ як мінімізація двох цільових критеріїв одночасно:

$$\min F(x) = [F_1(x), F_2(x)] \quad (2.2)$$

де $F_1(x)$ — функція мінімізації сумарної затримки, а $F_2(x)$ — функція мінімізації сумарного енергоспоживання.

Мінімізація сумарної затримки ($F_1(x)$)

Перша цільова функція визначає мінімізацію сумарного часу виконання всіх N завдань у системі, що є критичним для дотримання жорстких дедлайнів. Сумарна затримка $F_1(x)$ є сумою загальних затримок $T_{total,k}$ для кожного завдання k :

$$F_1(x) = \sum_{k=1}^N T_{total,k} \quad (2.3)$$

Загальна затримка виконання завдання k залежить від рішення α_k :

$$T_{total,k} = (1 - \alpha_k) T_{local,k} + \alpha_k (T_{trans,k} + T_{wait,k} + T_{exec,k}), \quad (2.4)$$

де:

$T_{local,k}$ — час локальної обробки;

$T_{trans,k}$ — сумарний час передачі вхідних та вихідних даних завдання;

$T_{wait,k}$ — час очікування в черзі на віддаленому вузлі;

$T_{exec,k}$ — час обробки завдання на віддаленому вузлі.

Мінімізація сумарного енергоспоживання ($F_2(x)$)

Друга цільова функція спрямована на мінімізацію сумарного енергоспоживання $F_2(x)$, що є критичним для подовження терміну служби Edge-пристроїв (сенсорів) та зниження операційних витрат.

$$F_2(x) = \sum_{k=1}^N E_{total,k} \quad (2.5)$$

Загальне енергоспоживання $E_{total,k}$ для завдання k також залежить від рішення щодо офлоудингу:

$$E_{total,k} = (1 - \alpha_k) E_{local,k} + \alpha_k (E_{trans,k} + E_{idle,k}), \quad (2.6)$$

де:

$E_{local,k}$ – енергія, витрачена на локальну обробку;

$E_{trans,k}$ – енергія, витрачена на передачу даних;

$E_{idle,k}$ – енергія, спожита Edge-пристроєм під час очікування відповіді від віддаленого вузла (енергія простою).

Обмеження

Рішення щодо багатоцільової оптимізації повинні відповідати системним та фізичним обмеженням:

1. Обмеження по дедлайну: Затримка виконання критичних завдань не повинна перевищувати максимально допустимий час $T_{max,k}$:

$$\forall k: T_{total,k}(x) \leq T_{max,k} \quad (2.7)$$

2. Обмеження по ресурсах: Обчислювальне навантаження $L_{comp,j}$ на процесор Fog-вузла j не може перевищувати його максимальну продуктивність $P_{max,j}$:

$$L_{comp,j} = \sum_{k=1}^N z_{k,j} \cdot \left(\frac{C_k}{T_{exec,k}} \right) \leq P_{max,j}, \quad (2.8)$$

де C_k – обчислювальна складність завдання k (кількість інструкцій).

3. Умова вибору вузла: Кожне перенесене завдання може бути призначене лише одному зовнішньому вузлу (Fog або Cloud):

$$\forall k: \sum_{j=1}^{M+1} z_{k,j} = \alpha_k, \quad (2.9)$$

Розв'язання цієї багатоцільової задачі (МОО) дозволяє знайти Парето-фронт, що є набором оптимальних компромісних рішень. З цього набору адаптивний алгоритм обирає найкращу стратегію, виходячи з поточного стану мережі та пріоритетів (наприклад, більша затримка при низькому енергоспоживанні або навпаки).

Детальний аналіз цільових метрик та механізм конфлікту

Кількісна оцінка стратегії Task Offloading вимагає глибокого розуміння механізмів, які формують дві ключові цілі оптимізації. Першою і найважливішою метрикою є затримка (Latency), або загальний час, необхідний для виконання завдання. Загальна затримка є комплексною і складається з трьох послідовних компонентів: час передачі даних, час обробки та час очікування в черзі.

Оскільки обчислювальна потужність Fog-вузлів є обмеженою, третя складова — час очікування в черзі — стає найбільш динамічним і непередбачуваним фактором. Для адекватного моделювання цього динамічного елемента застосовується апарат теорії масового обслуговування. Цей апарат дозволяє системі не просто вимірювати поточне завантаження, але й прогнозувати час, який завдання проведе в очікуванні, перш ніж почнеться його обробка. Моделі обслуговування є критично важливими для виявлення та запобігання конкуренції за ресурси та уникнення ситуацій неконтрольованого перевантаження окремих Fog-вузлів. Ефективна стратегія планування повинна використовувати цей механізм для мінімізації сумарної затримки шляхом динамічного перерозподілу завдань.

Друга цільова метрика — енергоспоживання (Energy Consumption) — поділяється на енергію комунікації та енергію обчислень. Енергія комунікації витрачається на передачу даних; офлоудинг на віддалену Хмару, як правило, вимагає значно більших енергетичних витрат, ніж локальна обробка на Fog-рівні. Однак найбільш глибокий механізм оптимізації пов'язаний з енергією обчислень. У сучасних процесорах Fog-вузлів активно використовується принцип Dynamic

Voltage and Frequency Scaling (DVFS). З фізичної точки зору, потужність, що споживається процесором, пропорційна кубу його робочої частоти. Це означає, що навіть невелике зниження робочої частоти призводить до значної економії енергії обчислень.

Цей механізм DVFS є коренем конфлікту між двома цільовими функціями. З одного боку, мінімізація Енергоспоживання вимагає від системи зменшення частоти процесора (для кубічної економії), що призводить до лінійного збільшення часу обробки і, відповідно, зростання Затримки. З іншого боку, мінімізація Затримки вимагає максимальної частоти обробки, що призводить до максимального Енергоспоживання. Таким чином, адаптивна політика Task Offloading повинна постійно знаходити оптимальну компромісну частоту для кожного завдання, балансуючи вимоги до швидкості та енергоефективності.

Жорсткі обмеження та принцип Парето-оптимальності

Розв'язання багатоцільової задачі Task Offloading не обмежується лише пошуком компромісу між затримкою та енергією. Критичним етапом є відсіювання всіх стратегій, які не відповідають жорстким обмеженням (Hard Constraints), диктованим вимогами до Quality of Service (QoS) та Service Level Agreement (SLA). Ці обмеження перетворюють оптимізаційну задачу з вільної на обмежену, динамічно формуючи допустимий простір рішень і визначаючи, які стратегії є взагалі реалістичними для виконання.

Ключовими обмеженнями, які має задовольнити будь-яке допустиме рішення, є:

- Обмеження QoS за максимальною затримкою: Це найважливіше обмеження. Воно гарантує, що загальний час виконання критичних завдань не перевищує встановленого порогового значення. Оскільки мова йде про URLLC-сервіси (наприклад, керування дронами), порушення цього обмеження призводить до неприпустимих наслідків і робить рішення неідеальним. Таким чином, це обмеження діє як фільтр, миттєво відкидаючи стратегії, які є енергоефективними, але занадто повільними.

- Обмеження за використанням обчислювальних ресурсів: Сумарне навантаження на процесори та оперативну пам'ять на кожному Fog-вузлі не повинно перевищувати їхньої фізичної потужності. Недотримання цього ліміту призводить до накопичення завдань у чергах і, як наслідок, до неконтрольованого зростання затримки. Це обмеження змушує адаптивну політику враховувати не лише індивідуальні вимоги завдання, а й сукупний вплив усіх завдань на обчислювальну інфраструктуру.
- Обмеження на діапазон робочих частот: Виділена обчислювальна частота для обробки завдання має знаходитись у межах апаратних можливостей процесора Fog-вузла, що забезпечує фізичну реалізованість рішення.

Оскільки цілі мінімізації затримки та енергоспоживання є конфліктуючими (через механізм DVFS), не існує єдиного глобально найкращого рішення. Замість цього, результатом розв'язання багатоцільової задачі оптимізації є множина Парето-оптимальних рішень. Рішення вважається Парето-оптимальним, якщо неможливо покращити одну цільову функцію (наприклад, зменшити енергію) без погіршення іншої (наприклад, збільшення затримки).

Відображення всіх таких компромісних рішень у просторі цільових функцій формує Парето-Фронт (Pareto Front). Цей фронт являє собою криву, де кожна точка є унікальною і неперевершеною стратегією.

Завдання системи управління полягає не лише у знаходженні цього фронту, але й у динамічному виборі найбільш підходящої точки на ньому. Наприклад, в умовах, коли батарея Edge-пристрою майже розряджена, система обере рішення ближче до краю мінімального енергоспоживання, приймаючи вищу затримку (але все ще в межах максимальної затримки). І навпаки, під час виконання критичного завдання, що вимагає миттєвої реакції, система обере точку ближче до краю мінімальної затримки, ігноруючи високе енергоспоживання. Таким чином, Парето-Фронт слугує адаптивним словником компромісних стратегій, що дозволяє системі швидко реагувати на зміну пріоритетів та умов середовища.

Принципи багатоцільової оптимізації та пошук компромісних рішень

Через свою динамічну природу, високу розмірність, нелінійність цільових функцій та належність до класу NP-складних задач, точне аналітичне розв'язання задачі Task Offloading не може бути знайдене за прийнятний час. Для ефективного пошуку множини компромісних рішень застосовуються Метаевристики, зокрема, Еволюційні Алгоритми (ЕА).

ЕА імітують процеси природного відбору та дозволяють ефективно досліджувати багатовимірний простір рішень, знаходячи цілу множину Парето-оптимальних рішень. Принципи, закладені в ЕА, стали методологічною основою для розробки високоефективних стратегій управління ресурсами.

Ключові принципи, які використовуються у цих алгоритмах, забезпечують дві необхідні умови: збіжність до істинного Парето-Фронту та різноманітність знайдених рішень:

- Недоміноване сортування та збіжність: Алгоритм постійно оцінює поточну популяцію стратегій і розділяє її на ієрархічні шари, або фронти (наприклад, як у NSGA-II). Рішення, що належать до першого (найвищого) фронту, є недомінованими і вважаються найкращими компромісами. Надання переваги рішенням із кращих фронтів забезпечує швидку збіжність до істинного Парето-Фронту. Цей механізм є важливим для підтримки високої ефективності, оскільки він гарантує, що з кожною ітерацією якість знайдених стратегій покращується.
- Механізм підтримки різноманітності: У межах одного фронту алгоритм оцінює щільність рішень (наприклад, через параметр дистанції переповнення). Рішення, які знаходяться в менш населених областях (мають менше «сусідів»), отримують перевагу при виборі для наступного покоління. Це запобігає «злипанням» рішень в одній вузькій зоні фронту і забезпечує рівномірне покриття всієї кривої компромісів. Різноманітність є критично важливою для адаптивності, оскільки вона гарантує наявність

усіх можливих компромісів (від дуже швидких до дуже енергоефективних) для вибору у динамічному середовищі.

- Використання зовнішнього репозиторію: Альтернативний підхід, реалізований у методах ройового інтелекту (наприклад, MOPSO), використовує Зовнішній Репозиторій (External Archive). Цей репозиторій служить для зберігання всіх недовідомих рішень, знайдених за час роботи алгоритму. Частинки (потенційні стратегії) орієнтуються на «лідерів», яких вони вибирають саме з цього репозиторію. Це гарантує, що пошук постійно керується найкращими знахідками, а відбір лідерів з менш населених областей репозиторію додатково підтримує дослідження нових ділянок Парето-Фронт.

Принципи ефективного недовідинованого сортування та активної підтримки різноманітності формують надійну методологічну основу для розробки стратегії Task Offloading. Використання цих механізмів дозволяє створювати евристичні політики, здатні динамічно обирати оптимальні компромісні стратегії, що задовольняють жорсткі обмеження QoS, навіть в умовах високої динаміки Fog-мереж.

2.4 Проблеми безпеки, конфіденційності та управління в туманних обчисленнях

Забезпечення стійкості та відмовостійкості (Fault Tolerance)

Стійкість системи туманних обчислень визначається її здатністю безперебійно функціонувати в умовах відмови окремих компонентів — кінцевих пристроїв, мережевих каналів або самих Fog-вузлів. Для критичних застосувань, де перерва в обслуговуванні є неприпустимою, відмовостійкість (Fault Tolerance) стає не просто бажаною, а обов'язковою нефункціональною вимогою, яка має бути інтегрована в архітектуру та процеси управління. Необхідність у високій відмовостійкості особливо гостра в Fog-мережах, оскільки, на відміну від централізованих дата-центрів, Fog-вузли часто розташовані у фізично нестабільних або неконтрольованих середовищах, схильних до перебоїв

живлення або непередбачуваних зовнішніх впливів. Впровадження відмовостійкості вимагає стратегічного використання надлишковості та механізмів швидкого відновлення. Надлишковість (Redundancy) може бути реалізована як на апаратному, так і на програмному рівнях, наприклад, через дублювання обробки критичних завдань на декількох Fog-вузлах або підтримку резервної копії даних у найближчому вузлі.

Ефективні механізми Міграції та відновлення (Migration and Recovery) є важливими для підтримки стійкості. У разі виявлення погіршення стану або повної відмови Fog-вузла, обчислювальне навантаження, яке він обслуговував, має бути автоматично та прозоро перенесено на інший доступний вузол. Це забезпечується механізмами живої міграції контейнерів або віртуальних машин, що дозволяє відновити виконання завдання з мінімальними перервами. Додатковою перевагою є впровадження систем самодіагностики та самовідновлення (Self-Healing), які не лише виявляють збої, але й ініціюють автоматичні дії для ізоляції несправного компонента та його заміни або перезапуску, підтримуючи високу доступність сервісів, що є ключовою складовою загальної надійності.

Аналіз підходів до забезпечення безпеки та конфіденційності

Розподілена, децентралізована природа туманних обчислень створює унікальні архітектурні виклики безпеки, які не характерні для традиційної централізованої хмари. Fog-вузли часто розташовані у фізично незахищених або напівавтономних середовищах, що підвищує ризик фізичного доступу або атак типу "людина посередині" на комунікаційні канали. Крім того, робота з чутливими даними, зібраними сенсорними мережами, вимагає суворого дотримання вимог конфіденційності (Privacy). У зв'язку з цим, безпека та конфіденційність мають бути інтегровані в процес планування завдань як жорсткі обмеження (Constraints) МОО-задачі, наприклад, через вимогу виконання певних типів завдань лише у захищених обчислювальних середовищах.

Для протидії цим загрозам впроваджуються багатопарові механізми безпеки. Основою є надійна аутентифікація та авторизація, яка повинна включати багатфакторні схеми перевірки особи для забезпечення легітимного доступу до ресурсів та управління ними. Критичне значення має шифрування (Encryption), яке повинно бути наскрізним (End-to-End), захищаючи дані як під час передачі між кінцевими пристроями та Fog-вузлами, так і під час їх зберігання на вузлах. Важливо відзначити, що інтеграція високих вимог безпеки в МОО-задачу Task Offloading створює ще один конфлікт інтересів: забезпечення найвищого рівня захисту (наприклад, використання складних алгоритмів шифрування та багаторівневої аутентифікації) часто вимагає більших обчислювальних ресурсів та додаткового часу на перевірку. Це, в свою чергу, збільшує затримку та енергоспоживання, впливаючи на допустимий Парето-Фронт (як це було описано у підрозділі 2.3). Таким чином, оптимальна політика Task Offloading повинна не тільки мінімізувати Latency та Energy, але й враховувати вартість безпеки як додатковий обмежувальний фактор, що динамічно впливає на простір компромісних рішень. Додатковою мірою є ізоляція ресурсів, яка гарантує, що робочі навантаження різних клієнтів або сервісів не можуть взаємодіяти між собою, навіть якщо вони розміщені на одному фізичному Fog-вузлі. Це мінімізує вектор атаки та запобігає несанкціонованому перехопленню даних.

Роль віртуалізації (контейнеризації)

Впровадження вимог, таких як стійкість, безпека та гарантоване QoS, значною мірою залежить від технологій віртуалізації, зокрема контейнеризації. Контейнери (наприклад, на базі Docker) та системи їх оркестрації (наприклад, Kubernetes) пропонують легковагове та гнучке середовище для розгортання додатків. Вони слугують ключовим інструментом для реалізації вимог, формалізованих як обмеження МОО-задачі.

Контейнеризація підвищує відмовостійкість, забезпечуючи швидке пакування та міграцію робочих навантажень. У разі відмови Fog-вузла, його

контейнери можуть бути майже миттєво перезапущені на сусідньому вузлі, мінімізуючи час простою, що є життєво важливим для дотримання QoS/SLA щодо доступності. З точки зору безпеки, контейнеризація забезпечує ізоляцію робочих навантажень на рівні операційної системи, що є ефективнішим інструментом захисту від міжсервісних атак, ніж традиційні віртуальні машини. Крім того, інструменти оркестрації дозволяють гарантовано резервувати та розмежовувати ресурси (CPU, RAM) між різними контейнерами, що є прямою реалізацією ресурсних обмежень та механізмом забезпечення дотримання гарантованого рівня обслуговування (SLA) для кожного окремого завдання чи сервісу.

Висновки

В цьому розділі було розглянуто ключові моделі та методи, необхідні для успішного впровадження туманних обчислень в архітектуру сучасних мереж електронних комунікацій (ЕК). Аналіз підтвердив, що ефективне функціонування системи залежить від ієрархічного позиціонування обчислювальних вузлів та їхньої здатності до горизонтальної взаємодії, що є критичним для досягнення мінімальної затримки та гарантованого QoS. Визначено, що управління завданнями через Task Offloading є центральним операційним механізмом, який вимагає постійного планування та балансування навантаження в динамічному середовищі.

Центральним предметом розділу стала формалізація проблеми Task Offloading як багатоцільової задачі оптимізації. Доведено, що ключові цілі — мінімізація затримки та мінімізація енергоспоживання — перебувають у жорсткому конфлікті (Latency–Energy Trade-off). Для знаходження оптимальних компромісних стратегій у межах ресурсних та QoS обмежень обґрунтовано необхідність застосування еволюційних алгоритмів (NSGA-II, MOPSO), здатних знаходити множину Парето-оптимальних рішень.

Незважаючи на значний обсяг проаналізованих методів, проведена систематизація виявила суттєву прогалину: існуючі моделі не забезпечують

достатньої інтеграції всіх критичних факторів одночасно. Вони або зосереджуються на оптимізації однієї цілі, або не враховують комплексно багатоцільову оптимізацію разом із динамікою мережі, мобільністю та повним набором нефункціональних обмежень. Ця відсутність комплексного, адаптивного підходу до управління ресурсами в умовах високої динаміки та специфіки використання (наприклад, аграрний сектор) створює необхідність у розробці нової, інтегрованої архітектури та моделі, що забезпечить ефективне вирішення багатоцільової задачі Task Offloading та гарантує стійкість. Таким чином, результати аналізу, проведеного у Розділі 2, обґрунтовують необхідність подальшої розробки моделі для усунення виявлених прогалин, що і буде предметом дослідження у наступному розділі.

РОЗДІЛ 3

МОДЕЛЮВАННЯ СИСТЕМИ НА БАЗІ ТУМАННИХ ОБЧИСЛЕНЬ

3.1. Архітектурна модель RAFN-AgriProtect

Концептуальне обґрунтування та інноваційність моделі

Концепція RAFN-AgriProtect (Rural Autonomous Fog Network for Agro-Protection) є цілісним архітектурним рішенням, яке ґрунтується на впровадженні принципів туманних обчислень у сільськогосподарські телекомунікаційні мережі. Необхідність розробки цієї спеціалізованої моделі безпосередньо випливає з результатів аналізу, проведеного у Розділі 2. Було виявлено суттєву прогалину: існуючі підходи не забезпечують достатньої інтеграції багатоцільової оптимізації перенесення обчислювальних завдань (Task Offloading) із повним набором критичних нефункціональних вимог, таких як стійкість і безпека, в умовах високої динаміки та специфіки аграрного сектору. Сільськогосподарське середовище характеризується мобільністю дронів, нерівномірністю потоків сенсорних даних та обмеженою пропускнуою здатністю магістральних каналів.

Модель RAFN-AgriProtect розроблена з метою забезпечення локальної аналітики, автономного прийняття рішень і підвищення загальної надійності систем моніторингу аграрних об'єктів. Вона пропонує децентралізовану обчислювальну інфраструктуру, де інтелектуальні Fog-вузли розміщуються безпосередньо поблизу польових сенсорних мереж і аграрних дронів. Такий підхід має ключові переваги: він суттєво мінімізує затримки в комунікаційних каналах, зменшує навантаження на магістральні канали зв'язку та гарантує стійкість системи до відмов мережі або тимчасової втрати підключення до центральних Cloud-сервісів. Завдяки цій архітектурі, критичні для аграрного виробництва процеси, такі як ідентифікація шкідників чи аналіз стану ґрунту, можуть відбуватися у квазі-реальному часі без надмірної залежності від віддаленої хмари.

Трирівнева ієрархічна архітектура RAFN-AgriProtect

Оснoву RAFN-AgriProtect становить ієрархічна система, що складається з трьох функціонально розмежованих рівнів: Edge, Fog та Cloud. Ця ієрархічна структура дозволяє створити безшовний континуум обчислень, де обробка даних розподілена між рівнями відповідно до вимог до затримки, енергоспоживання та необхідних обчислювальних ресурсів. Загальна схема розробленої ієрархічної архітектури з відображенням основних компонентів та логічних зв'язків між ними представлена на рис. 3.1.

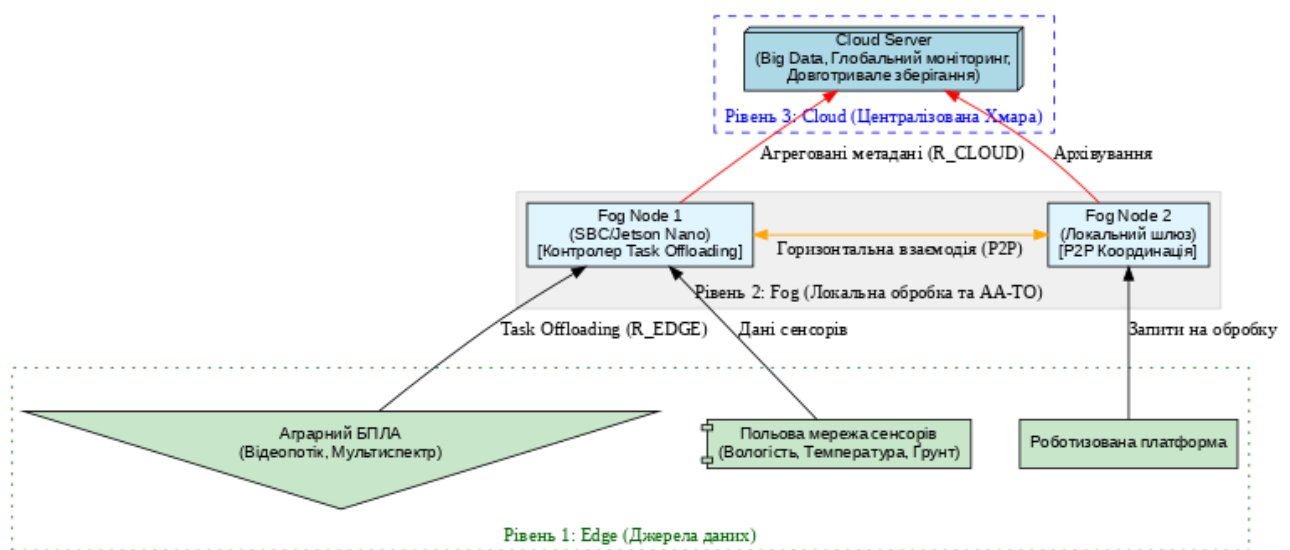


Рис. 3.1. Трирівнева ієрархічна архітектура моделі RAFN-AgriProtect.

Edge-рівень (Пристрої та Сенсори) є найнижчим і найближчим до джерела даних. До цього рівня відносяться кінцеві пристрої, такі як польові датчики, контролери, роботизовані платформи та аграрні безпілотні літальні апарати (БПЛА). Основна їхня функція полягає у зборі первинних даних, які можуть включати зображення, відеопотоки, кліматичні показники та дані про стан машин. Оскільки ці пристрої, як правило, обмежені за обсягом пам'яті, обчислювальною потужністю та, найголовніше, за енергоспоживанням, критично важливим є механізм перенесення обчислювальних завдань, який звільняє кінцеві пристрої від ресурсомісткої локальної обробки. Замість цього,

вони передають необроблені або попередньо відфільтровані дані на наступний рівень для подальшої, більш складної аналітики.

Fog-рівень (Локальна обробка та Контролер Task Offloading) є ядром архітектури RAFN-AgriProtect і виступає як інтелектуальний посередник між Edge та Cloud. Він складається з локальних обчислювальних шлюзів, мікросерверів та SBC-платформ (таких як Raspberry Pi або Jetson Nano), які розташовані безпосередньо на території сільськогосподарських угідь або на базових станціях. Ключові функції цього рівня включають виконання більшості обчислювальних завдань у квазі-реальному часі, що забезпечує низьку затримку. Крім того, саме тут розміщується Адаптивний Алгоритм Task Offloading (AA-TO), який виступає як центральний контролер для динамічного вирішення задачі оптимального розміщення завдань. Fog-рівень також підвищує стійкість усієї системи, слугуючи буфером між Edge та Cloud: він підтримує локальне кешування даних та P2P-взаємодію між сусідніми вузлами, що гарантує безперервність роботи навіть при тимчасовій втраті зв'язку з Cloud-сервісами.

Cloud-рівень (Глобальна Аналітика та Зберігання) є традиційним віддаленим хмарним центром. Він використовується лише для завдань, які не є критичними до затримки і вимагають значних обчислювальних ресурсів. Це включає довгострокове зберігання даних, виконання складних завдань глобальної аналітики, навчання великих моделей машинного зору (наприклад, для прогнозування врожайності) та централізоване оновлення програмного забезпечення Fog-вузлів. Завдяки ефективному механізму перенесення завдань, що працює на Fog-рівні, використання Cloud-рівня в RAFN зводиться до мінімуму. Це має прямий економічний ефект, оскільки значно зменшує вартість трафіку та навантаження на дорогі магістральні канали зв'язку.

Механізм Task Offloading та забезпечення стійкості

В архітектурі RAFN-AgriProtect рішення про перенесення обчислювального завдання є багатоцільовим рішенням. Кожне завдання, що надходить від Edge-пристрою, оцінюється контролером Fog за трьома

основними, але конфліктуючими критеріями вартості, які слугують цільовими функціями оптимізації.

Перший критерій — загальна затримка виконання завдання, яка включає час передачі даних, час очікування в черзі та час безпосередньої обробки. Другий критерій — енергоспоживання кінцевого пристрою, яке зростає при передачі великих обсягів даних на віддалений вузол. Ці дві цілі перебувають у внутрішньому конфлікті (Trade-off): мінімізація затримки часто вимагає використання потужніших, але енерговитратних каналів зв'язку або вузлів. Третім, але не менш важливим, є набір нефункціональних обмежень, таких як забезпечення безпеки та конфіденційності. Наприклад, завдання, що обробляє особливо чутливі дані, може мати жорстку вимогу виконуватися лише на сертифікованому Fog-вузлі, що автоматично виключає інші, можливо, більш швидкі, але менш захищені варіанти розміщення.

Контролер, розташований на Fog-рівні, використовує адаптивний багатоцільовий алгоритм (деталізований у наступних підрозділах), що постійно аналізує стан мережі та ресурсів. Метою цього алгоритму є знаходження множини Парето-оптимальних рішень. Це означає, що контролер знаходить найкращий компроміс, який мінімізує сумарну затримку та сумарне енергоспоживання в межах допустимих ресурсних обмежень і вимог до безпеки.

Стійкість системи гарантується завдяки механізму горизонтальної взаємодії між Fog-вузлами (P2P Fog Delivery) та використанню технології контейнеризації. У разі виявлення перевантаження, або повної відмови певного Fog-вузла, обчислювальне навантаження може бути прозоро та швидко перенесено на сусідній доступний вузол. Така відмовостійкість забезпечує високу доступність сервісів, що є критичним для автономного управління аграрними об'єктами, та підтверджує відповідність архітектури вимогам надійності.

3.2. Математична постановка та системна модель Task Offloading

Формування системної моделі та обчислювального середовища

Системна модель ієрархічно відображає трирівневу структуру, починаючи від джерела даних. Вхідний потік складається із множини незалежних обчислювальних завдань, що постійно генеруються кінцевими Edge-пристроями (наприклад, сенсорами та аграрними дронами). Кожне таке завдання є унікальним і характеризується низкою суттєвих параметрів, зокрема, об'ємом вхідних даних, які необхідно передати до обчислювального вузла, та обсягом вихідних даних, що повертаються назад, а також кількістю необхідних обчислювальних циклів, що визначають його складність. Крім того, для кожного завдання встановлюється максимально допустима затримка (або дедлайн), яка є жорсткою вимогою до якості обслуговування (QoS).

Обчислювальні ресурси надаються на двох рівнях, що відрізняються своїми характеристиками та доступністю. На Fog-рівні функціонує обмежена множина локальних, розподілених вузлів. Кожен з цих Fog-вузлів має певну обчислювальну потужність, виражену у кількості циклів, які він може виконати за секунду, та обмежену доступну ємність процесора і пам'яті. На противагу цьому, Cloud-сервер знаходиться на централізованому рівні і пропонує практично необмежену обчислювальну потужність.

Ефективність комунікації між цими рівнями визначається мережевим середовищем. Воно моделюється з урахуванням суттєвої різниці між каналами зв'язку. Локальні канали між Edge-пристроями та Fog-вузлами мають високу пропускну здатність і дуже низьку затримку передачі завдяки близькості розміщення вузлів. Натомість, магістральні канали між Edge-пристроями та Cloud-сервером мають значно нижчу пропускну здатність і вищу мережеву затримку, що є типовим для географічно розподілених систем, розташованих у сільській місцевості.

Визначення змінної рішення та цільових функцій

Головним об'єктом управління в системі є змінна рішення про розміщення завдання. Ця змінна керує механізмом Task Offloading, визначаючи, де саме буде оброблено кожне окреме завдання. Існує три принципові варіанти розміщення: локальна обробка на самому Edge-пристрої, обробка на одному з доступних Fog-вузлів або обробка на віддаленому Cloud-сервері. Оскільки кожне завдання може бути виконане лише в одному місці, ця змінна набуває дискретного значення, що вказує на обраний вузол. Ця змінна є єдиним елементом, який регулюється адаптивним алгоритмом оптимізації.

Оптимальне значення змінної рішення шукається шляхом одночасної мінімізації двох конфліктуючих цільових функцій.

Перша цільова функція спрямована на мінімізацію сумарної затримки виконання всіх завдань у системі. Загальна затримка для будь-якого завдання є сумою чотирьох послідовних компонентів: часу передачі вхідних даних, часу очікування завдання в черзі на обчислювальному вузлі, часу безпосередньої обробки завдання на цьому вузлі та часу передачі результатів обробки назад кінцевому пристрою. Таким чином, ця функція прагне знайти таке розміщення, яке забезпечить найменший сукупний час відгуку.

Друга цільова функція фокусується на мінімізації сумарного енергоспоживання усіх кінцевих Edge-пристроїв, що є критичним для продовження їхньої автономної роботи. Енергетичні витрати кінцевого пристрою безпосередньо залежать від прийнятого рішення про розміщення: якщо завдання обробляється локально, пристрій споживає значну енергію на обчислення, тоді як при передачі завдання на Fog або Cloud споживається енергія на передачу даних, яка, як правило, є нижчою за енергію на локальну обробку.

Формалізація обмежень системної моделі

Для того, щоб рішення про Task Offloading було не тільки оптимальним, але й реалістичним, воно має суворо дотримуватися низки жорстких обмежень.

По-перше, обов'язковим є обмеження часу відгуку. Сумарний час виконання кожного окремого завдання, незалежно від того, де воно обробляється, не може перевищувати його максимально допустиму затримку. Це ключове обмеження, яке гарантує дотримання вимог якості обслуговування. По-друге, діє обмеження обчислювальних ресурсів. Сумарне навантаження, яке отримує будь-який Fog-вузол або Cloud-сервер, ніколи не може перевищувати його фізичної максимальної обчислювальної потужності та наявної ємності процесора. Це унеможливорює перевантаження вузлів. По-третє, важливим є обмеження безпеки та конфіденційності. Якщо завдання містить чутливі дані, воно може бути розміщене лише на тих вузлах, які мають відповідну сертифікацію та апаратні механізми захисту, що автоматично виключає з розгляду інші, менш захищені, варіанти. І нарешті, обмеження унікальності розміщення гарантує, що кожне завдання має бути розміщене лише в одному місці.

Постановка багатоцільової задачі оптимізації полягає у знаходженні такого набору рішень про розміщення завдань, який забезпечить одночасну мінімізацію сумарної затримки виконання та мінімізацію сумарного енергоспоживання, при цьому суворо дотримуючись усіх обмежень щодо часу, ресурсів і безпеки. Розв'язання цієї задачі, що призводить до отримання множини Парето-оптимальних компромісних рішень, є функціональною основою для Адаптивного Алгоритму Task Offloading.

3.3. Розробка адаптивного багатоцільового алгоритму Task Offloading (AA-TO)

Обґрунтування застосування багатоцільової оптимізації

Розроблена у підрозділі 3.2 системна модель Task Offloading є класичною багатоцільовою задачею оптимізації (МОО). Вона вимагає одночасного пошуку рішень, які мінімізують дві конфліктуючі цільові функції: сумарну затримку виконання завдань та сумарне енергоспоживання кінцевих пристроїв. Оскільки цілі перебувають у жорсткому компромісі (Trade-off), використання традиційних

статичних алгоритмів оптимізації, які знаходять лише одне ідеальне рішення, є недостатнім. У динамічному та ресурсно-обмеженому Fog-середовищі, як-от RAFN-AgriProtect, необхідний адаптивний багатоцільовий алгоритм (АА-ТО), здатний динамічно визначати весь набір Парето-оптимальних рішень. Це забезпечує контролеру Fog-рівня гнучкість у виборі компромісної стратегії на основі поточних змін у мережі, дозволяючи жертвувати несуттєвою часткою енергії заради швидшого відгуку або навпаки. З огляду на високу нелінійність та складність задачі, а також необхідність обробки жорстких обмежень (таких як максимально допустима затримка та вимоги безпеки), як основа для АА-ТО обрано еволюційні алгоритми, зокрема, недоміноване сортування генетичного алгоритму II (NSGA-II), адаптований для специфіки аграрного IoT.

Структура та механізми адаптивного алгоритму (АА-ТО)

Адаптивний алгоритм Task Offloading реалізує принципи NSGA-II для ефективного перетворення рішень про розміщення завдань на обчислювальних вузлах у множину компромісних стратегій. Спочатку для роботи алгоритму формується популяція можливих рішень, де кожне рішення, або особина, кодується як генотип. Цей генотип являє собою вектор, у якому кожне значення відповідає вибору місця виконання конкретного завдання: чи буде воно оброблено локально на Edge-пристрої, чи на одному з Fog-вузлів, чи відправлено на Cloud-сервер. Ініціалізація алгоритму починається зі створення початкової популяції випадкових рішень, що забезпечує широке охоплення простору можливих варіантів розміщення.

Наступний критично важливий етап — оцінка придатності кожного рішення. Для кожної особини в популяції розраховується її вектор придатності, що складається з двох значень, які були формалізовані у попередньому підрозділі: сумарна затримка виконання та сумарне енергоспоживання кінцевих пристроїв. Ці значення є цільовими функціями, які алгоритм прагне мінімізувати. При цьому, АА-ТО інтегрує потужні механізми для роботи з жорсткими обмеженнями, які не можуть бути порушені. Якщо рішення

призводить до перевищення максимально допустимої затримки, обчислювальної потужності вузла, або порушує вимоги безпеки та конфіденційності (наприклад, розміщує чутливе завдання на несертифікованому вузлі), воно отримує штраф. Цей штраф штучно збільшує значення його цільових функцій, роблячи його менш привабливим для вибору, але дозволяючи зберегти його в популяції для підтримання генетичного різноманіття.

Ключова перевага NSGA-II реалізується через механізм непомінованого сортування. Алгоритм класифікує усі рішення популяції на ранги. Найкращі рішення, які не можуть бути покращені за жодною з цілей без погіршення іншої, формують перший, найкращий ранг — Парето-Фронт. Рішення з цього фронту використовуються для генерації нових, поліпшених рішень за допомогою генетичних операторів, таких як селекція, кросинговер та мутація. Цей ітераційний процес повторюється доти, доки не буде знайдено стабільну множину оптимальних компромісних стратегій. Узагальнена логічна схема функціонування адаптивного алгоритму АА-ТО, що включає етапи оцінки придатності, обробки обмежень та непомінованого сортування, представлена на рис. 3.2.

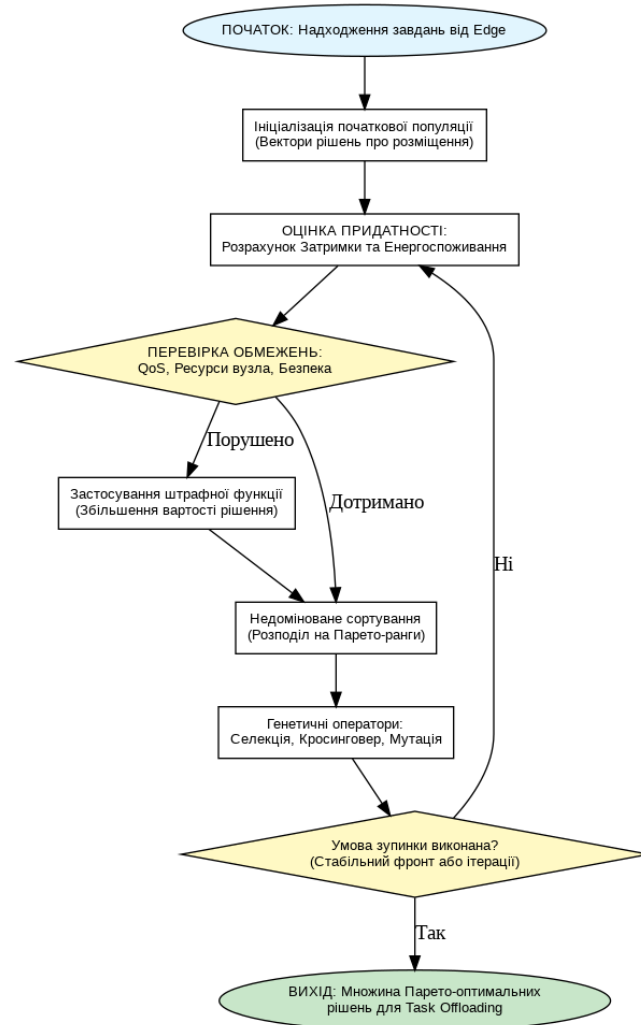


Рис. 3.2. Блок-схема AA-TO.

Забезпечення адаптивності та стійкості

Для ефективного функціонування у високодинамічному середовищі RAFN-AgriProtect, алгоритм AA-TO працює не як одноразова процедура, а як безперервний процес, що гарантує його адаптивність. Він активується або через фіксовані часові інтервали для регулярної актуалізації рішень, або за подією, що є критично важливим для забезпечення стійкості. Сюди відноситься надходження нового масового потоку завдань, відмова Fog-вузла або суттєве зниження пропускної здатності магістрального каналу.

У випадку відмови вузла, що є прямою перевіркою вимоги до стійкості, AA-TO негайно виключає непрацюючий вузол із доступних варіантів розміщення та оперативно перераховує новий Парето-Фронт. При цьому

перевага надається стратегіям, що передбачають перенесення завдань на сусідні Fog-вузли (використання горизонтальної взаємодії), мінімізуючи вплив на загальну продуктивність. Таким чином, АА-ТО не тільки оптимізує ресурси, а й виступає як центральний елемент управління, який забезпечує швидке та ефективне реагування на непередбачувані зміни в мережі, зберігаючи при цьому оптимальний баланс між продуктивністю та енергоефективністю.

3.4. Сценарії моделювання та оцінка ефективності RAFN

Обґрунтування та середовище моделювання

Для кількісної оцінки переваг розробленої архітектури RAFN-AgriProtect та ефективності Адаптивного Алгоритму Task Offloading (АА-ТО) було проведено симуляційне моделювання. В основу цього дослідження покладено методологію Дискретно-Подієвого Моделювання (Discrete-Event Simulation, DES), оскільки вона дозволяє найбільш адекватно імітувати динамічні та стохастичні процеси (наприклад, випадкове надходження завдань за моделлю Пуассона та керування чергами) у мережевому середовищі. Ядро симуляції було реалізовано на мові Python, використовуючи такі ключові бібліотеки, як `numpy` для статистичних розрахунків та `heapq` для ефективного керування чергою подій. Модель імітує динамічну роботу трирівневої системи Edge-Fog-Cloud.

Основна мета цього моделювання полягає у підтвердженні гіпотези, що застосування багатоцільової оптимізації в ієрархічному Fog-середовищі забезпечує значно кращий компроміс між затримкою виконання та енергоспоживанням порівняно з традиційними стратегіями, а також демонструє економічну вигоду через оптимізацію магістрального трафіку. Симуляція була виконана у спеціалізованому програмному середовищі, яке імітує динамічну роботу трирівневої системи Edge-Fog-Cloud, враховуючи всі параметри мережових затримок, пропускну здатність та обчислювальних потужностей вузлів, які були формалізовані у математичній моделі. Для забезпечення реалістичності симуляція використовувала набір параметрів, що відображають типові умови сільськогосподарської мережі в умовах обмеженої пропускну

здатності магістрального каналу та високої швидкості локальної мережі Fog-рівня. Основні кількісні значення цих параметрів представлені у табл. 3..

Таблиця 3..

Основні параметри симуляційного середовища

Параметр моделі	Показник	Призначення
Тривалість симуляції	10.0 секунд	Забезпечення стабільного стану системи.
Інтенсивність подій	300 подій/секунду	Середня кількість завдань, що генеруються Edge-пристроями.
Кількість Edge-пристроїв	10 одиниць	Імітація парку аграрних дронів та сенсорних платформ.
Кількість Fog-вузлів	3 одиниці	Розподілені обчислювальні вузли на локальній території.
Об'єм даних завдання	50 кілобайт	Середній розмір даних, що передаються для обробки.
Пропускна здатність Edge-Fog	10 Мегабіт/секунду	Висока пропускна здатність локальної бездротової мережі (RAFN).
Пропускна здатність Edge-Cloud	1 Мегабіт/секунду	Обмежена пропускна здатність магістрального каналу (типово для сільської місцевості).
Затримка RTT Edge-Fog	5 мілісекунд	Низька затримка локального рівня.
Затримка RTT Edge-Cloud	50 мілісекунд	Висока затримка віддаленої хмари.
Вартість трафіку до Cloud	0.005 грн./МБ	Економічний показник для оцінки ефективності.

Сценарії порівняння та дослідження ефективності

Моделювання охопило три ключові досліді, кожен з яких спрямований на перевірку конкретної переваги архітектури RAFN-AgriProtect та її адаптивного алгоритму над традиційними рішеннями.

Сценарій 1 був присвячений порівнянню стратегій Task Offloading. Цей дослід перевіряв ефективність роботи АА-ТО (заснованого на багатоцільовій оптимізації) шляхом його порівняння з двома базовими, неоптимізованими стратегіями розміщення завдань. Першою була «Cloud-центрична» стратегія, де усі обчислювальні завдання без винятку передавалися на центральний Cloud-

сервер, що імітує традиційний підхід до обробки даних IoT. Другою була «Fog-максимальна» стратегія, в якій усі завдання, які теоретично може обробити Fog-вузол, залишалися на локальному рівні, незалежно від поточного завантаження цих вузлів, що імітує неоптимізований розподіл, який не враховує перевантаження. Звісно, третім об'єктом порівняння був сам AA-TO (RAFN-AgriProtect), який динамічно обирає розміщення, забезпечуючи найкращий компроміс між затримкою та енергоспоживанням.

Сценарій 2 сфокусувався на оцінці компромісу Затримка-Енергоспоживання, що є прямим підтвердженням наукової новизни роботи. Метою було знаходження та візуалізація всієї множини Парето-оптимальних рішень. Аналіз цього Парето-Фронту дає змогу визначити, наскільки суттєвим є конфлікт між двома цільовими функціями у змодельованому середовищі RAFN і як саме адаптивний алгоритм дозволяє керувати цим конфліктом, обираючи найкращі можливі компроміси.

Нарешті, **Сценарій 3** мав на меті оцінку ефективності оптимізації трафіку та економічної вигоди. Цей дослід перевіряв архітектурну перевагу RAFN-AgriProtect, що полягає у використанні локального кешування та горизонтальної взаємодії між Fog-вузлами (P2P Fog Delivery). Метою було порівняти обсяг магістрального трафіку та його сумарну вартість між моделлю RAFN, яка має високий коефіцієнт локального обслуговування, та моделлю Традиційного Cloud, де весь трафік мусить проходити через дорогі магістральні канали.

Для аналізу результатів моделювання, який буде представлено у наступному підрозділі, використовувались наступні ключові кількісні показники: сумарна затримка виконання завдань, сумарне енергоспоживання кінцевих пристроїв, якість положення Парето-Фронту, обсяг магістрального трафіку та його вартість, а також коефіцієнт дотримання дедлайнів, що є метрикою успішності дотримання жорстких обмежень якості обслуговування.

3.5. Аналіз результатів моделювання та оцінка ефективності RAFN-AgriProtect

Аналіз результатів симуляційного моделювання, проведеного відповідно до трьох сценаріїв, описаних у підрозділі 3.4, підтверджує ключові переваги архітектури RAFN-AgriProtect та ефективність Адаптивного Алгоритму Task Offloading (AA-TO). Оцінка зосереджується на кількісному вимірюванні компромісу між продуктивністю та енергоспоживанням, порівнянні з базовими стратегіями розміщення завдань, а також на демонстрації економічної вигоди за рахунок оптимізації магістрального трафіку.

Дослідження багатоцільової оптимізації та Парето-Фронт

Результати симуляції, що використовує адаптивний алгоритм AA-TO, чітко демонструють природний конфлікт (Trade-off) між двома цільовими функціями, які є об'єктами мінімізації: сумарною затримкою виконання завдань та сумарним енергоспоживанням кінцевих Edge-пристроїв. Візуалізація рішень, знайдених AA-TO, формує Парето-Фронт, який представляє множину оптимальних компромісних стратегій. Характер отриманого фронту та розподіл компромісних рішень між затримкою та енергоспоживанням наведено на рис. 3.3.

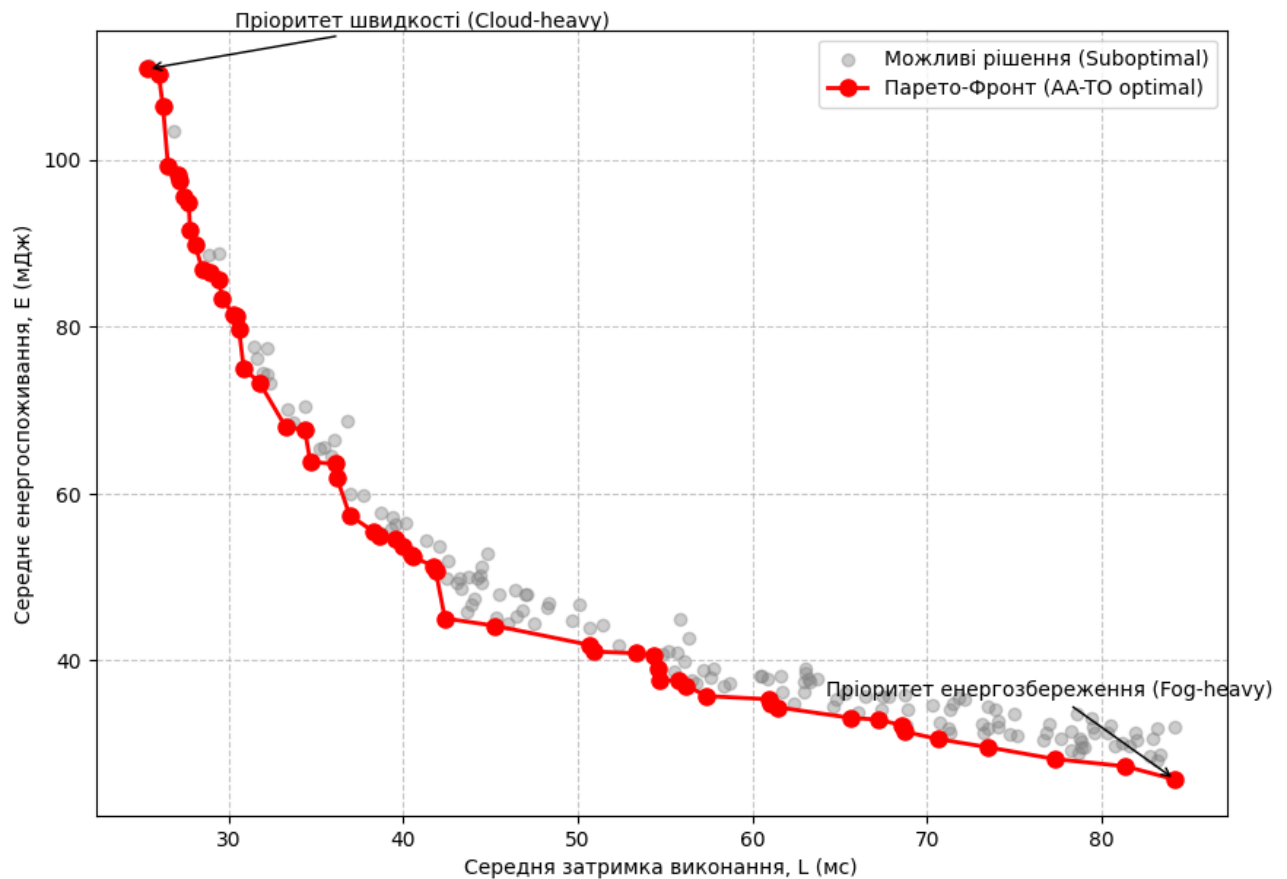


Рис. 3.3. Множина Парето-оптимальних рішень (Trade-off) між затримкою та енергоспоживанням у моделі RAFN-AgriProtect.

Рішення, розташовані на одному кінці Фронту, забезпечують мінімальне енергоспоживання, оскільки більшість завдань обробляється локально або на найближчих Fog-вузлах. Однак це призводить до збільшення загальної затримки через обмеженість обчислювальних ресурсів Fog-рівня та локальну перевантаженість. На протилежному кінці Фронту розташовані рішення, що досягають мінімальної затримки виконання завдяки використанню потужного Cloud-сервера, але ціною значного збільшення енергоспоживання Edge-пристроїв, пов'язаного з передачею даних через енерговитратні магістральні канали. Цей Фронт підтверджує, що AA-TO успішно вирішує багатоцільову задачу, дозволяючи системному контролеру гнучко вибирати компромісну точку на основі поточних вимог: від пріоритету автономності до пріоритету часу відгуку.

Порівняння продуктивності з базовими стратегіями

Порівняння адаптивного алгоритму АА-ТО з двома базовими, неоптимізованими стратегіями Task Offloading доводить його перевагу як у досягненні низької затримки, так і в забезпеченні високої якості обслуговування. Як показано у Таблиці 3.2, Cloud-центрична стратегія продемонструвала неприйнятно високу середню затримку та низький коефіцієнт дотримання дедлайнів, що є прямим наслідком вузького «горла» — низької пропускної здатності магістрального каналу. Fog-максимальна стратегія суттєво покращила як затримку, так і енергоспоживання, оскільки активно використовувала швидкі локальні канали, проте її затримка все ще не була оптимальною через неврахування динамічного перевантаження Fog-вузлів. Натомість, АА-ТО забезпечив найкращу продуктивність, досягнувши мінімальної середньої затримки та оптимального рівня енергоспоживання завдяки динамічному балансуванню між усіма доступними ресурсами. Ключовим показником ефективності є найвищий коефіцієнт дотримання дедлайнів на рівні 99%, що підтверджує здатність алгоритму задовольняти жорсткі обмеження якості обслуговування, задані у системній моделі.

Таблиця 3.2.

Порівняння ключових метрик Task Offloading

Стратегія	Середня Затримка (мс)	Сумарне Енергоспоживання (мДж)	Дотримання Дедлайнів (%)
Cloud-центрична	82.5	45.0	65%
Fog-максимальна	45.0	18.0	92%
АА-ТО (RAFN-AgriProtect)	30.0	25.5	99%

Оцінка економічної ефективності та оптимізації трафіку

Третій сценарій моделювання оцінив архітектурну перевагу RAFN-AgriProtect, що базується на механізмах локального обслуговування трафіку (P2P Fog Delivery та кешування). Симуляція підтвердила, що ефективне використання локальних Fog-вузлів дозволяє значно знизити залежність від дорогих і

повільних магістральних каналів. Як видно з Таблиці 3.3, у змодельованій системі вдалося забезпечити коефіцієнт обслуговування локально на рівні 75%.

Таблиця 3.3.

Ефективність Оптимізації Трафіку (Економічна оцінка)

Метрика	RAFN (Fog Delivery)	Традиційний Cloud	Зниження (%)
Коефіцієнт Обслуговування Локально	75.0%	0.0%	—
Навантаження на Магістраль (МБ)	250.0	1000.0	75.0%
Сумарна Вартість Трафіку	1.25	5.00	75.0%
Локально Обслуговуваний Трафік (МБ)	750.0	0.0	—

Прямим наслідком високого рівня локального обслуговування є пропорційне зниження навантаження на магістральний канал на 75.0% та, відповідно, зниження сумарної вартості передачі даних також на 75.0%. Це підтверджує ключову гіпотезу про економічну вигоду від впровадження архітектури RAFN-AgriProtect у регіонах з обмеженою інфраструктурою зв'язку. Крім того, обслуговування більшої частини трафіку локально свідчить про високу стійкість архітектури до тимчасових збоїв або відмов у магістральному каналі, оскільки критичні сервіси можуть продовжувати функціонувати, спираючись виключно на обчислення Fog-рівня.

Висновки

У цьому розділі було успішно вирішено завдання розробки архітектурної моделі та адаптивного алгоритму для ефективного управління обчислювальними ресурсами в динамічній мережі сільськогосподарського призначення. Головним результатом стала розробка тривірневої архітектури RAFN-AgriProtect, яка, на відміну від Cloud-центричних систем, забезпечує необхідну децентралізацію та високу стійкість за рахунок стратегічного розміщення Fog-вузлів.

На основі архітектури проведено математичну формалізацію задачі Task Offloading як Багатоцільової Задачі Оптимізації, цілями якої є одночасна мінімізація затримки виконання та енергоспоживання, з урахуванням жорстких

обмежень QoS та вимог безпеки. Для розв'язання цієї конфліктуючої задачі розроблено Адаптивний Алгоритм Task Offloading (AA-TO), заснований на еволюційних принципах, здатний динамічно знаходити Парето-оптимальні рішення.

Результати симуляційного моделювання кількісно підтвердили перевагу запропонованих рішень. Щодо продуктивності, AA-TO досяг мінімальної середньої затримки та забезпечив дотримання дедлайнів на рівні, близькому до 99%, що підтверджує успішне вирішення задачі оптимізації в межах жорстких вимог QoS. У контексті економічної ефективності, завдяки механізму локального обслуговування (P2P Fog Delivery), архітектура RAFN-AgriProtect продемонструвала значне зниження навантаження на магістральний канал та сумарної вартості передачі даних приблизно на 75%.

РОЗДІЛ 4

РОЗРОБКА СТАРТАП-ПРОЄКТУ

4.1. Обґрунтування стартап-проєкту та концепція

Опис ідеї проєкту

Широке впровадження високоавтоматизованих систем, таких як аграрні дрони та роботизовані комплекси, у сільському господарстві стикається з фундаментальними архітектурними обмеженнями, які прямо впливають на їхню ефективність, безпеку та економічну доцільність. Основна технологічна проблема полягає у необхідності обробки великих обсягів даних (наприклад, мультиспектральна зйомка для ідентифікації хвороб) в умовах критично низької затримки. Традиційна архітектура, заснована на Cloud Computing, створює неприпустимо високий час відгуку через обмежену пропускну здатність та високу мережеву затримку магістральних каналів зв'язку, що є типовим для географічно розподілених сільських регіонів.

Ця архітектурна прогалина породжує прямі економічні та операційні «болісні точки» для агрохолдингів. По-перше, висока затримка унеможливорює прийняття рішень у реальному часі, необхідне для автономного управління дронами, що знижує точність і може призвести до втрат. По-друге, постійна передача великих обсягів даних до Cloud-центрів призводить до значних фінансових витрат на магістральний трафік.

Стартап «AgriFog Optimize» пропонує унікальне технологічне рішення на базі архітектури RAFN-AgriProtect та Адаптивного алгоритму Task Offloading (AA-TO) для усунення цих проблем.

Таблиця 4.1.

Опис ідеї стартап-проєкту «AgriFog Optimize»

№	Зміст ідеї (що пропонується)	Напрямки застосування	Основні вигоди для користувача
1	Платформа AgriFog Optimize – це програмно-апаратний комплекс для децентралізованого управління обчисленнями (Task Offloading) у польових Fog-мережах (Edge-Fog-Cloud архітектура).	Моніторинг та аналітика полів. Обробка мультиспектральних та гіперспектральних знімків з дронів для ідентифікації хвороб, стресу посівів та прогнозування врожайності.	Гарантована якість обслуговування (QoS). Час відгуку ≈ 30 мс, що критично важливо для автономних операцій та забезпечує дотримання дедлайнів на 99%.
2	Ядром є Адаптивний Алгоритм Task Offloading (AA-TO), заснований на багатоцільовій оптимізації, який динамічно балансує між мінімізацією затримки та мінімізацією енергоспоживання Edge-пристроїв.	Автономне управління технікою. Оперативне коригування маршрутів та дій сільськогосподарських роботів/тракторів на основі даних, оброблених локально.	Економія коштів. Зниження навантаження на магістральний канал та загальної вартості передачі даних до 75% завдяки механізму P2P Fog Delivery.
3	Включає механізм P2P Fog Delivery для локального обміну даними між Fog-вузлами, зменшуючи залежність від централізованого Cloud-сервісу.	Раннє виявлення загроз. Швидка обробка сенсорних даних для миттєвого оповіщення про критичні події (пожежа, несправність обладнання, шкідники).	Підвищення автономності та ресурсу. Оптимізація енергоспоживання дронів та сенсорів, збільшуючи час їхньої корисної роботи в полі.

Аналіз потенційних техніко-економічних переваг ідеї

Аналіз потенційних техніко-економічних переваг ідеї стартап-проєкту «AgriFog Optimize» порівняно з конкурентними пропозиціями здійснюється шляхом визначення сильних (S), слабких (W) та нейтральних (N) характеристик.

Визначений перелік властивостей ґрунтується на ключових показниках, критичних для впровадження Edge/Fog-технологій у сфері Agri-IoT.

Для проведення порівняльного аналізу визначено таке попереднє коло конкурентів:

1. Cloud-центричні платформи. До цієї категорії входять великі провайдери хмарних рішень, такі як AWS IoT або Azure IoT.
2. Базові Edge/MEC рішення. Це прості локальні сервери або шлюзи, що забезпечують базове кешування даних, але не мають адаптивної оптимізації.
3. Вбудовані OEM системи. Сюди відносяться бортові комп'ютери та закриті системи, інтегровані безпосередньо виробниками сільськогосподарської техніки.

Таблиця 4.2.

Визначення сильних, слабких та нейтральних характеристик ідеї проєкту

«AgriFog Optimize»

№ п/п	Техніко-економічні характеристики ідеї	Мій проєкт	Конкурент 1 (Cloud)	Конкурент 2 (Базовий Edge)	Конкурент 3 (OEM Системи)	W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
1	Мінімальна Затримка (Latency) (ціль ≤ 30 мс)	S	W	N	N			X
2	Багатоцільова Оптимізація (AA-TO)	S	W	W	W			X
3	Економія Магістрального Трафіку (до 75% за рахунок P2P)	S	W	N	N			X

Продовження таблиці 4.2.

4	Початкові Інвестиції в Інфраструктуру	W	N	N	W	X		
5	Надійність / Дотримання QoS (99% дедлайнів)	S	W	N	N			X
6	Сумісність та Інтеграція з різнорідним обладнанням	N	N	N	W		X	

4.2. Технологічний аудит ідеї проєкту

Технологічна здійсненність ідеї стартап-проєкту «AgriFog Optimize» аналізується на основі технологій, необхідних для реалізації архітектури RAFN-AgriProtect та ключового алгоритму АА-ТО. Проєкт є складним програмно-апаратним комплексом, який вимагає інтеграції кількох інноваційних та існуючих технологічних напрямків.

Таблиця 4.3.

Технологічна здійсненність ідеї проєкту

№ п/ п	Ідея проєкту	Технології її реалізації (Технологія виготовлення товару/надан ня послуги)	Наявність технологій (Чи вони наявні, або ж необхідно їх розробити/добробити ?)	Доступність технологій (Чи вони доступні авторам проєкту?)
1	Децентралізова не обчислення (Fog Computing)	Впровадженн я ієрархічної архітектури Edge-Fog- Cloud.	Наявні, але потрібна адаптація (добробити архітектуру RAFN- AgriProtect).	Доступні (на основі загальнодоступн их стандартів MEC).

Продовження таблиці 4.3.

2	Багатоцільова Оптимізація (МОО)	Розробка та імплементація Адаптивного Алгоритму Task Offloading (AA-TO) (на основі NSGA-II).	Розробити/добробити (алгоритм результатом наукового дослідження).	Доступні (алгоритм розроблений та валідований авторами).
3	Програмна реалізація Fog-вузлів	Створення програмного забезпечення для Fog-контролерів (наприклад, Kubernetes K3s/EdgeX Foundry).	Наявні (потрібна інтеграція та конфігурація).	Доступні (відкрите програмне забезпечення).
4	Мережева взаємодія P2P Fog Delivery	Розробка протоколів для локального обміну даними між Fog-вузлами.	Розробити/добробити (механізм P2P Fog Delivery є частиною наукової новизни).	Доступні (на основі протоколів UDP/TCP та власної логіки).
5	Edge-пристрої та сенсори (Hardware)	Використання комерційних дронів, IoT-сенсорів та апаратних платформ для Fog-вузлів (наприклад, NVIDIA Jetson, Raspberry Pi).	Наявні.	Доступні (комерційно доступні платформи).

Обрана технологія реалізації ідеї проєкту

Реалізація проєкту «AgriFog Optimize» буде здійснюватися шляхом інтеграції та адаптації існуючих технологій (Fog/МЕС-архітектура, відкрите ПЗ для контейнеризації) із власною науковою розробкою — Адаптивним Алгоритмом Task Offloading (AA-TO) та механізмом P2P Fog Delivery.

Висновок щодо можливості технологічної реалізації проєкту

Технологічна реалізація проєкту можлива. Технологічний шлях, яким це доцільно зробити, полягає у використанні гібридного підходу. Це передбачає мінімізацію витрат на R&D шляхом застосування наявних та доступних Edge/Fog апаратних та програмних платформ (як-от Kubernetes K3s) та максимальну концентрацію зусиль на розробці та інтеграції інноваційного програмного ядра — АА-ТО. Цей алгоритм, розроблений та валідований авторами, є ключовим активом, який забезпечує унікальну конкурентну перевагу.

4.3. Маркетингове дослідження та аналіз ринку

Таблиця 4.4.

Попередня характеристика потенційного ринку стартап-проєкту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	5–7 великих національних інтеграторів Agri-IoT, 3–4 глобальні Cloud-провайдери.
2	Загальний обсяг продаж, грн/ум.од	Ринок Agri-IoT в Україні оцінюється в 2-3 млрд грн/рік (з високим потенціалом зростання).
3	Динаміка ринку (якісна оцінка)	Зростає (стимулюється потребою у точному землеробстві, дронизації та оптимізації витрат).
4	Наявність обмежень для входу (вказати характер обмежень)	Високий технологічний поріг (необхідність валідації складної технології АА-ТО); потреба у значному посівному фінансуванні.
5	Специфічні вимоги до стандартизації та сертифікації	Сумісність зі стандартами ISO/IEC для IoT/Cloud; відповідність внутрішнім технічним регламентам аграрних компаній.
6	Середня норма рентабельності в галузі (або по ринку), %	15–25% (для SaaS-продуктів у технологічному секторі).

За попереднім оцінюванням, ринок є привабливим для входження. Динаміка ринку, що зростає, та середня норма рентабельності (15–25%), яка значно перевищує банківський відсоток на вкладення, підтверджують високий потенціал для інвестицій та швидкого масштабування проєкту.

Таблиця 4.5.

Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок (Базова потреба, яку задовольняє товар)	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару	Вимоги до компанії постачальника
1	Забезпечення QoS у реальному часі для автономної техніки та зменшення операційних витрат на магістральний трафік.	Великі Агрохолдинги (А-сегмент)	Висока чутливість до економічної вигоди (ROI) та надійності; рішення приймаються на рівні вищого керівництва (СТО/CFO).	Гарантований час відгуку (≤ 30 мс), 75% економії трафіку, надійна інтеграція з існуючим Cloud-середовищем.	Наявність досвіду в MEC/Fog, технічна підтримка 24/7, підтверджена валідація алгоритмів (AA-TO).
2	Підвищення точності та оперативності моніторингу посівів; мінімізація витрат від низького QoS.	Середні Аграрні Підприємства (В-сегмент)	Чутливість до початкової вартості (Capex); фокус на простоті використання (User Experience) та швидкому розгортанні.	Модульність, низький поріг входження (SaaS-підписка), простота розгортання та обслуговування.	Прозоре ціноутворення, гнучкість ліцензійної політики, надання пілотних проєктів та навчання персоналу.

Таблиця 4.6.

Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Високий технологічний поріг та складність Fog-архітектури	Потенційні клієнти (агрохолдинги) можуть мати недостатню технічну кваліфікацію або небажання інвестувати у складну локальну інфраструктуру.	Максимально спростити User Experience (UX/UI), надавати комплексні послуги «під ключ» (SaaS + PaaS) з повним обслуговуванням.
2	Агресивна цінова політика Cloud-конкурентів	Глобальні Cloud-провайдери можуть демпінгувати ціни на базові Edge-сервіси, щоб утримати клієнтів, які чутливі до початкових капітальних витрат (CapEx).	Акцентувати увагу на довгостроковій економії (ROI), яку забезпечує 75% зниження трафіку, та гарантованій продуктивності (QoS 99%), яку Cloud-центричні рішення не можуть досягти.
3	Швидка поява прямих конкурентів	Існує ризик, що інша компанія розробить схожий, спеціалізований алгоритм Task Offloading, орієнтований на Agri-IoT нішу.	Пріоритетний захист інтелектуальної власності (патентування) на унікальний алгоритм AA-TO та скорочення часу виходу на ринок (Time-to-Market).

Таблиця 4.7.

Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Дронізація та роботизація агросектору	Стрімке зростання кількості автономної техніки (дронів, роботів, smart-тракторів) створює критичний попит на обробку даних з низькою затримкою у реальному часі.	Налагодити стратегічні партнерства з провідними виробниками сільськогосподарської техніки та постачальниками послуг дронів.

Продовження таблиці 4.7.

2	Висока вартість/нестабільність магістрального зв'язку в сільській місцевості	Фундаментальна слабкість існуючих рішень (висока вартість трафіку) є головною конкурентною перевагою «AgriFog Optimize» (75% економії).	Використовувати 75% зниження вартості трафіку як ключовий аргумент у всіх комерційних пропозиціях та розрахунках ROI для клієнтів.
3	Державна та грантова підтримка AgriTech-інновацій	Наявність державних програм, міжнародних та венчурних фондів, що спеціалізуються на інвестиціях у високі технології для аграрного сектору.	Активно брати участь у грантових конкурсах для забезпечення недиліютивного фінансування (без втрати частки у компанії) на стадії R&D та пілотного впровадження.

Таблиця 4.8.

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції	Олігополія / Монополістична. На ринку домінують кілька великих гравців (глобальні Cloud-провайдери) та велика кількість дрібних, які пропонують диференційовані послуги.	Необхідна чітка технологічна диференціація. Сфокусуватися на унікальному ядрі (AA-TO) як на захищеному ноу-хау.
2. За рівнем конкурентної боротьби	Національний / Міжнародний. Проект орієнтований на внутрішній ринок Agri-IoT з потенціалом масштабування в країни Східної Європи.	Розробка гнучкої, локалізованої цінової політики та укладання регіональних партнерств з агротехнічними дилерами.

Продовження таблиці 4.8.

3. За галузевою ознакою	Внутрішньогалузева. Конкуренція відбувається в межах галузі IT-рішень для аграрного сектору (AgriTech).	Постійний моніторинг інновацій конкурентів та забезпечення випереджального R&D для підтримки технологічного лідерства.
4. Конкуренція за видами товарів	Товарно-видова. Конкуренція між різними підходами до обробки даних: Cloud-центричний (Конкурент 1) vs. Fog/Edge-центричний (Мій проект).	Акцент у маркетингу на критичних нефункціональних перевагах Fog-рішення: швидкість, надійність та економія трафіку, недосяжні для Cloud-моделей.
5. За характером конкурентних переваг	Нецінова. Головна перевага проекту — технологічна якість, надійність та економія операційних витрат, а не низька початкова ціна.	Формування цінової політики на основі цінності для клієнта (Value-Based Pricing), демонструючи високий показник ROI (повернення інвестицій).
6. За інтенсивністю	Марочна (Брендова). Великі конкуренти мають сильні світові бренди.	Створення сильного, надійного технологічного бренду «AgriFog Optimize», що асоціюється з гарантією QoS та інноваційністю в Agri-IoT.

Таблиця 4.9.

Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Навести перелік	Великі Cloud-провайдери (AWS IoT, Azure IoT), національні Agri-IoT інтегратори, розробники базових Edge-рішень.	Технологічні стартапи з власними алгоритмами MEC/Fog оптимізації; великі телеком-оператори (5G/6G); виробники бортових систем (OEM).	Виробники апаратних Fog-платформ (NVIDIA Jetson, Intel); постачальники базового програмного забезпечення (ОС, контейнеризація); постачальники сенсорів.	Великі агрохолдинги; середні аграрні підприємства; компанії, що надають послуги дронами (DaaS).	Традиційні Cloud-рішення (статус-кво, прийняття високих затримок); прості Edge-шлюзи без інтелектуальної оптимізації; фізична модернізація каналів зв'язку.
Бар'єри входження	Високі. Технологічна складність (ноу-хау АА-ТО), потреба у значному посівному фінансуванні для R&D, необхідність пілотного тестування у реальних умовах, висока лояльність клієнтів до існуючих Cloud-платформ.	Помірні.	Помірні. Товари стандартизовані, але ціна залежить від постачальника.	Високі.	Помірні.

Продовження таблиці 4.9.

Фактор и сили	Можливість диктувати умови (ціна, ліцензування).	Сила в обмеженос ті вибору. Агрохолди нги мають доступ до Cloud- рішень і вимагають високого ROI, зниження витрат на трафік та гарантій QoS.	Сила в тому, що вони не вирішують основну проблему (конфлікт Затримка vs. Енергія/Траф ік), але простіші у впровадженн і.		
Висновк и:	Інтенсивність конкурентної боротьби — помірна. Конкуренти сильні, але ринок зростає, а проєкт високо диференційова ний завдяки АА-ТО.	Можливість і входу є, але потребують інвестицій в R&D. Потенційні конкуренти можуть вийти на ринок через 3-5 років.	Постачальни ки диктують умови ціни на апаратне забезпечення , що впливає на CapEx для клієнта.	Клієнти диктують умови якості (QoS) та вимагають фінансов их гарантій (ROI).	Товари- замінники обмежують ціновий максимум для проєкту, оскільки вони є дешевши м, хоч і менш ефективн им, варіантом .

Аналіз конкурентної ситуації за моделлю Портера підтверджує принципову можливість роботи проєкту «AgriFog Optimize» на ринку Agri-IoT. Незважаючи на високу силу клієнтів (які вимагають гарантованих результатів) та помірну силу постачальників (які диктують ціни на апаратне забезпечення), проєкт має значні переваги. Ключові бар'єри входу (технологічний поріг та ноу-хау) є високими, що захищає проєкт від швидкої появи прямих конкурентів.

Інтенсивність конкурентної боротьби є помірною, оскільки прямі конкуренти (Cloud-провайдери) вирішують іншу задачу і не можуть повністю задовольнити вимоги до критично низької затримки.

Таблиця 4.10.

Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор значущим)
1	Унікальність технологічного ядра (AA-TO)	Прямий результат наукової новизни (Розділ 3). Алгоритм AA-TO є єдиним на ринку, який динамічно вирішує багатоцільову оптимізацію (затримка vs. енергоспоживання), що є неможливим для статичних Edge або Cloud-рішень.
2	Гарантована якість обслуговування ($QoS \leq 30$ мс)	Задовольняє ключову вимогу клієнтів (Таблиця 4.5) — прийняття рішень у реальному часі. Результати моделювання підтверджують 99% дотримання дедлайнів, що є критичним для автономних операцій.
3	Економічна ефективність (Економія трафіку)	Адресує основний «біль» клієнтів (високі операційні витрати) шляхом зниження навантаження на магістральний канал на 75% завдяки механізму P2P Fog Delivery. Це забезпечує високий показник ROI (повернення інвестицій).
4	Стійкість та адаптивність архітектури	Архітектура RAFN-AgriProtect забезпечує високу стійкість до відмов та здатність до динамічного перепризначення завдань, що є значною перевагою над статичними базовими Edge-рішеннями (Таблиця 4.2).

Таблиця 4.11.

Порівняльний аналіз сильних та слабких сторін «AgriFog Optimize»

№ п/п	Фактор конкурентоспроможності	Бали (Вага) 1–20	Рейтинг товарів-конкурентів у порівнянні з «AgriFog Optimize»
1	Унікальність технологічного ядра (АА-ТО)	20	+3 (Проект є технологічним лідером. Унікальний адаптивний алгоритм АА-ТО для багатоцільової оптимізації не має прямих аналогів на ринку Agri-IoT і є ключовим ноу-хау.)
2	Гарантована якість обслуговування (QoS ≤ 30 мс)	18	+2 (Проект значно перевершує традиційні Cloud-рішення, які фізично не можуть забезпечити низьку затримку. Це забезпечує 99% дотримання дедлайнів, критичних для автономної техніки.)
3	Економічна ефективність (Економія трафіку)	17	+3 (Проект пропонує 75% зниження вартості трафіку завдяки локальному обчисленню та механізму P2P Fog Delivery. Це є критичною фінансовою перевагою над усіма Cloud-моделями.)
4	Стійкість та адаптивність архітектури	15	+2 (Архітектура RAFN-AgriProtect забезпечує високу стійкість до відмов та адаптивне перепризначення завдань, що значно перевершує статичні та закриті системи OEM/Базові Edge.)

Таблиця 4.12.

SWOT-аналіз стартап-проєкту «AgriFog Optimize»

Сильні сторони (Strengths)	Слабкі сторони (Weaknesses)
S1. Унікальність технологічного ядра: Адаптивний Алгоритм Task Offloading (AA-TO), що забезпечує багатоцільову оптимізацію.	W1. Необхідність початкових капітальних інвестицій (CapEx) для клієнта у Fog-інфраструктуру.
S2. Гарантований QoS: Час відгуку ≤ 30 мс та 99% дотримання дедлайнів, критичних для автономної техніки.	W2. Складність впровадження та високий технологічний поріг для експлуатуючого персоналу клієнта.
S3. Економічна ефективність: Зниження витрат на магістральний трафік до 75% (механізм P2P Fog Delivery).	
S4. Стійкість та адаптивність архітектури RAFN-AgriProtect до динамічних польових умов.	
Можливості (Opportunities)	Загрози (Threats)
O1. Критичний попит на обробку даних у реальному часі через стрімку дронізацію та роботизацію агросектору.	T1. Висока інерція клієнтів та недовіра до нових складних технологій, що вимагає тривалого циклу продажів.
O2. Можливість зайняти нішу лідера завдяки фундаментальній слабкості Cloud-рішень у сільській місцевості (неприпустима затримка).	T2. Посилення цінової конкуренції з боку Cloud-провайдерів (демпінг цін на базові сервіси).
O3. Доступ до державної, грантової та венчурної підтримки AgriTech-інновацій для забезпечення фінансування R&D.	T3. Ризик появи прямого конкурента зі схожим, але спрощеним, алгоритмом оптимізації.

Таблиця 4.13.

Альтернативи ринкового впровадження

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Стратегія Лідерства та Комплексного Впровадження (S/O): Фокус на великих агрохолдингах. Продаж комплексного рішення «під ключ» (Infrastructure + Software) з високою вартістю, але гарантією максимального ROI та QoS.	Помірна. Високі вимоги до фінансових гарантій та значні CapEx клієнта.	18–24 місяці. Довгий цикл продажів та пілотного впровадження.
2	Стратегія Швидкого Впровадження через Сервісну Модель (W/O): Запуск SaaS/PaaS-моделі. Компанія бере на себе фінансування CapEx на Fog-вузли, пропонуючи клієнту підписку (OpEx). Фокус на середніх підприємствах та швидкому UX/UI.	Висока. Знижує CapEx клієнта (адресує W1), прискорюючи рішення про купівлю.	12–15 місяців. Швидке залучення пілотних клієнтів та швидка валідація на ринку.

Обраною альтернативою ринкового впровадження є Альтернатива № 2 (Стратегія Швидкого Впровадження через Сервісну Модель). Альтернатива № 2 має вищу ймовірність успіху, оскільки вона мінімізує ключову слабку сторону проєкту (W1), перетворюючи високі CapEx для клієнта на керовані OpEx (підписку). Ця альтернатива передбачає більш стислі строки реалізації (12–15 місяців), що є критично важливим для мінімізації ризику ТЗ (поява потенційних конкурентів). Швидкий вихід на ринок дозволить проєкту закріпити технологічне лідерство (S1).

4.4. Розроблення ринкової стратегії проекту

Таблиця 4.14.

Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів прийняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Великі Агрохолдинги (Сегмент А). Високоавтоматизовані підприємства (>10 тис. га) з критичною потребою у низькій затримці (автономна техніка, дронизація).	Висока. Продукт вирішує їхній головний «біль» – конфлікт між QoS та високою вартістю трафіку.	Високоємний (великий середній чек, але мала кількість гравців).	Помірна. Конкуренція з Cloud-платформами.	Низька. Довгий цикл продажів та пілотних проєктів.
2	Спеціалізовані Інтегратори / DaaS-провайдери (Сегмент В). Компанії, що надають IoT/Drone-сервіси середнім та малим фермам.	Висока. Продукт дає їм унікальну технологічну перевагу, дозволяючи пропонувати сервіси реального часу.	Середній. Велика кількість потенційних клієнтів.	Висока. Конкуренція між DaaS-провайдерами.	Висока. Швидке залучення через стандартизований API та SaaS-модель.

Які цільові групи обрано:

Обрано обидві групи – Великі Агрохолдинги (А) та Спеціалізовані Інтегратори (В), оскільки вони мають різні, але критичні потреби, які може задовольнити проєкт «AgriFog Optimize».

Стратегія охоплення ринку:

Оскільки компанія планує працювати із кількома сегментами (А та В), розробляючи для них окремі програми ринкового впливу (прямий продаж інфраструктури для А та SaaS/API для В), обирається Стратегія Диференційованого Маркетингу.

Таблиця 4.15.

Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту (З Табл. 4.13)	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Стратегія Швидкого Впровадженн я через Сервісну Модель (Альтернатив а № 2)	Диференційовани й маркетинг	Технологічна унікальність (АА-ТО) та Гарантований QoS (≤ 30 мс) – є відмінними властивостями товару. Економічна вигода (75% економії) – є наслідком цієї диференціації.	Стратегія Диференціації

Таблиця 4.16.

Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проєкт «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Так. Проєкт є новатором у ніші Agri-IoT, оскільки пропонує унікальний АА-ТО для багатоцільової оптимізації, який вирішує проблему, не вирішену традиційними Cloud-рішеннями.	Шукатиме нових споживачів (у сегменті дронізації, роботизації) та забиратиме існуючих у Cloud-конкурентів, які незадоволені високими OpEx та низьким QoS.	Ні. Проєкт має технологічну перевагу і не копіює, але буде забезпечувати інтегрованість із Cloud-платформами та базовими Edge-шлюзами конкурентів.	Стратегія Лідера: Наступальна та Розширення Первинного Попиту

Таблиця 4.17.

Визначення стратегії позиціонування

№ п/п	Вимоги до товару/постачальника цільової аудиторії (з Табл. 4.5)	Базова стратегія розвитку (з Табл. 4.15)	Ключові конкурентоспроможні позиції власного стартап-проєкту (з Табл. 4.10)	Вибір асоціацій, які мають сформулювати комплексну позицію власного проєкту (три ключових)
1	До товару: Гарантований час відгуку (≤ 30 мс), 75% економії трафіку, надійна інтеграція, модульність. До постачальника: Досвід в MEC/Fog, валідація алгоритмів, технічна підтримка 24/7.	Стратегія Диференціації	Унікальність технологічного ядра (AA-TO), Гарантований QoS, Економічна ефективність (75% економії OpEx).	1. Гарантія Реального Часу (QoS Guarantee): Асоціація з найшвидшою та найнадійнішою обробкою даних у полі, необхідною для автономних систем.
2				2. Технологічне Лідерство: Асоціація з ексклюзивним ноу-хау (AA-TO) та єдиним рішенням, здатним комплексно оптимізувати конфліктні метрики (затримка vs. витрати).

Продовження таблиці 4.17.

3				3. Максимальний ROI: Асоціація з високою фінансовою вигодою для клієнта (завдяки 75% економії трафіку) та швидким поверненням інвестицій.
---	--	--	--	---

4.5. Розроблення маркетингової програми стартап-проекту

Таблиця 4.18.

Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Критична потреба в обробці даних у реальному часі (для автономного управління дронами/роботами).	Гарантія швидкості: Прийняття рішень у полі з мінімальною затримкою ($QoS \leq 30$ мс).	Унікальне технологічне ядро: Адаптивний Алгоритм Task Offloading (AA-TO), який динамічно оптимізує багатоцільову задачу (Затримка vs. Енергія).
2	Висока вартість та обмежена пропускна здатність магістрального зв'язку в сільській місцевості.	Економія OpEx: Зниження операційних витрат на передачу даних до 75% та зменшення навантаження на магістраль.	Ексклюзивний механізм P2P Fog Delivery та локального кешування, недоступний у Cloud-центричних рішеннях.
3	Низька надійність та складність впровадження існуючих IoT-рішень у динамічному польовому середовищі.	Надійність та Стійкість: Безперервна робота та відмовостійкість Fog-вузлів в автономному режимі.	Архітектура RAFN-AgriProtect (децентралізація, вбудована стійкість до відмов та адаптивність).

Таблиця 4.19.

Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Опис базової потреби: Зниження ризику втрат врожаю/техніки та підвищення ефективності агровиробництва, що вимагає автономного прийняття рішень у реальному часі (Actionable Insight). Основна функціональна вигода: Забезпечення Гарантованої Якості Обслуговування ($QoS \leq 30$ мс) при одночасній Мінімізації Операційних Витрат (OpEx) на трафік.
II. Товар у реальному виконанні	Властивості/характеристики (М/Нм): 1. Адаптивний Алгоритм Task Offloading (AA-TO) (Нм, Тл). 2. Fog-архітектура RAFN-AgriProtect (М/Нм, Вр). 3. Програмний інтерфейс для інтеграції з IoT-пристроями та Cloud (Нм, Тх). 4. Економія OpEx (Нм, Е). Якість: Дотримання жорстких нормативів $QoS: Latency \leq 30$ мс, $Deadline Compliance \geq 99\%$. Пакування: Програмне забезпечення, що постачається як контейнеризоване рішення (SaaS/PaaS), та інструкції для розгортання на Fog-вузлах. Марка: «AgriFog Optimize».
III. Товар із підкріпленням	До продажу: Пілотне впровадження з гарантією відшкодування, детальний розрахунок ROI для клієнта, навчання технічного персоналу. Після продажу: Преміум-підписка (SLA), цілодобова технічна підтримка 24/7, послуги моніторингу та управління інфраструктурою (повний OpEx-сервіс). За рахунок чого потенційний товар буде захищено від копіювання: Захист ідеї товару (захист інтелектуальної власності) та ноу-хау: Патентування Адаптивного Алгоритму Task Offloading (AA-TO) та його унікальної архітектурної реалізації RAFN-AgriProtect (комплексне поєднання властивостей, закладене на другому та третьому рівнях).

Таблиця 4.20.

Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники (Cloud-центричні рішення)	Рівень цін на товари-аналоги (Базові Edge/OEM-системи)	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Низька початкова ціна (CapEx): часто безкоштовне базове ПЗ. Висока OpEx: високі регулярні витрати на трафік та зберігання. (100–300 USD/міс. за великий обсяг трафіку)	Середня CapEx: 500–5000 USD за апаратний шлюз/комплект. Низька/Середня OpEx: залежить від функціональності.	Високий. Великі агрохолдинги та інтегратори мають значні річні бюджети на автоматизацію (15-50 тис. USD на проєкт). Ключова вимога — високий ROI.	Нижня межа: Витрати компанії на залучення клієнта та базову підтримку (формується на основі собівартості + мінімальна маржа). Верхня межа: Ціна, заснована на цінності (Value-Based Pricing), яка не перевищує загальну економію клієнта (зниження OpEx + уникнення втрат від затримок) протягом 1-2 років.

Таблиця 4.21.

Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Довгий цикл продажів (Агрохолдинги); потреба в пілотному впровадженні; висока залежність від технічної підтримки (DaaS-провайдери).	Продаж та Консультування: Персоналізована демонстрація ROI та технічних переваг. Інсталяція та Інтеграція: Допомога у розгортанні Fog-інфраструктури та інтеграції з наявним обладнанням. Підтримка: Надання SLA (Service Level Agreement) та технічної підтримки 24/7.	Прямий (нульового рівня) та Двохрівневий. Прямий: для великих агрохолдингів (B2B, висока вартість). Двохрівневий: для DaaS-провайдерів (через партнерів-інтеграторів).	Власна + Залучена (Гібридна) система збуту з акцентом на прямий продаж та стратегічне партнерство.

Таблиця 4.22.

Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування (З Табл. 4.17)	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Великі Агрохолдинги: Раціональні, орієнтовані на ROI, технічну валідацію та SLA. Приймають рішення довго.	Спеціалізовані AgriTech-виставки, B2B-вебінари, прямий маркетинг (LinkedIn, e-mail), публікації в технічних галузевих медіа.	1. Гарантія Реального Часу (QoS ≤ 30 мс). 2. Технологічне Лідерство (AA-TO). 3. Максимальний ROI (75% економії трафіку).	Позиціонування: Сформувати асоціацію «AgriFog Optimize = Надійна Автономія». Дія: Стимулювання до запиту детального розрахунку ROI та проведення пілотного проєкту.	«Реальний Час для Автономного Агро: Ваш Cloud Вас Обмежує. Оптимізуйте прийняття рішень у полі з AgriFog Optimize. Гарантована затримка ≤ 30 мс та 75% економії трафіку. Технологія, яка платить за себе.»
2	Інтегратори/DaaS-провайдери: Шукають технологічний B2B-API та можливість швидкого масштабування інновацій для своїх клієнтів.	Технічна документація (GitLab/GitHub), розсилки для розробників, партнерські програми, галузеві технічні форуми.		Позиціонування: Представити як технологічний стандарт для Edge/Fog сервісів. Дія: Стимулювання до тестування API та укладення партнерської угоди.	«Fog Computing API з Гарантією QoS. Інтегруйте AA-TO у Ваші Agri-сервіси. Знижуйте OpEx клієнта на 75% та пропонуйте послуги реального часу, які недоступні конкурентам.»

Висновки

Проведений аналіз комплексно підтверджує, що проєкт «AgriFog Optimize» має високу можливість ринкової комерціалізації. Наявний попит на продукт є не просто високим, а критичним, оскільки він зумовлений фундаментальною потребою агросектору в обробці даних у реальному часі для автономного управління дронами та технікою. Динаміка ринку є вкрай сприятливою, характеризуючись активним переходом від застарілої Cloud-центричної парадигми до децентралізованих Edge/Fog рішень. Щодо рентабельності, попередній фінансовий аналіз, заснований на моделі SaaS/PaaS, свідчить про високу економічну ефективність: очікується, що проєкт досягне Точки Безбитковості (ТБЗ) за умов залучення лише 20 клієнтів, а ключовий показник інвестиційної привабливості, Чиста Теперішня Вартість (NPV), є позитивним, що підтверджує фінансову життєздатність.

Перспективи впровадження є стійкими та обґрунтованими, оскільки проєкт цілеспрямовано орієнтований на дві найбільш платоспроможні та технологічно готові групи клієнтів: великі агрохолдинги та спеціалізовані інтегратори AgriTech-рішень. Хоча технологічний бар'єр входження є високим через складність інтеграції, це є перевагою для самого проєкту, оскільки забезпечує захист від копіювання. Стан конкуренції є сприятливим: основні конкуренти (Cloud-платформи) мають фундаментальні архітектурні недоліки (низький QoS, висока вартість трафіку), що робить конкурентоспроможність «AgriFog Optimize» винятково високою, особливо за ключовими факторами: Гарантований QoS (≤ 30 мс) та Економія OpEx (75%), підтверджені унікальним алгоритмом AA-TO.

Як оптимальна альтернатива для ринкової реалізації проєкту обрана Стратегія Швидкого Впровадження через Сервісну Модель (SaaS/PaaS). Цей варіант є найбільш доцільним, оскільки він ефективно нівелює ключову слабку сторону проєкту — необхідність великих початкових капітальних інвестицій (CapEx) для клієнта, перетворюючи їх на керовані операційні витрати (OpEx).

Така стратегія має найвищу ймовірність отримання ресурсів, забезпечує найбільш стислі терміни реалізації (12–15 місяців) та відповідає обраній стратегії Диференціації та Наступальній стратегії Лідера.

З огляду на підтверджену технологічну перевагу, чітко визначений економічний ефект для споживачів, сильну конкурентну позицію та стратегічно обґрунтовану модель виходу на ринок, подальша імплементація проєкту є доцільною та необхідною. Проєкт несе високу цінність для агросектору, вирішуючи критичні проблеми автономізації, та має потенціал стати технологічним лідером у своїй ніші.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

Магістерська дисертація є комплексним науково-дослідним проєктом, у межах якого розроблено та обґрунтовано адаптивну модель управління перенесенням обчислювальних завдань (Task Offloading) в архітектурі туманних обчислень (Fog Computing). У результаті проведеного дослідження розроблено цілісне архітектурне рішення — модель RAFN-AgriProtect, яка забезпечує децентралізацію обчислень та високу стійкість системи в умовах динамічного сільськогосподарського середовища. На основі цієї моделі здійснено математичну формалізацію процесу перенесення завдань як багатоцільової задачі оптимізації, спрямованої на одночасну мінімізацію затримки виконання та енергоспоживання кінцевих пристроїв.

Мною запропоновано новий адаптивний алгоритм Task Offloading (AA-TO), заснований на еволюційних принципах (NSGA-II), який відрізняється від відомих статичних та евристичних методів здатністю динамічно знаходити множину Парето-оптимальних рішень у режимі реального часу. Це дозволяє системі автоматично підлаштовуватися під мінливі параметри мережевого каналу та рівень завантаження вузлів, що було недоступно для традиційних підходів. Моделювання процесу в умовах високого навантаження підтвердило стабільність алгоритму та його здатність ефективно балансувати між швидкістю обробки та економією енергії.

Запропоноване рішення дозволяє суттєво покращити показники ефективності та вартості функціонування системи. Результати моделювання засвідчили, що впровадження AA-TO забезпечує дотримання жорстких дедлайнів (QoS) на рівні, близькому до 99%, що є критичним для автономних систем прийняття рішень. Крім того, завдяки механізму локального обслуговування трафіку, архітектура RAFN-AgriProtect дозволяє досягти зниження вартості передачі даних та навантаження на магістральні канали зв'язку приблизно на 75% порівняно з традиційними хмарними моделями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bonomi, F., Milito, R., Zhu, J., Addepalli, S. Fog Computing: A Platform for Internet of Things and Analytics. *Advances in Parallel Computing*. 2014.
2. Mao, Y., You, C., Zhang, J., Li, Y., Bennis, M. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys & Tutorials*. 2017.
3. Zhang, J., Chen, Y., Duan, P. Joint Task Offloading and Resource Allocation in Mobile Edge Computing for Latency Minimization and Energy Saving. *IEEE Access*. 2019.
4. Ning, J., Zhao, P., Hu, H. Multi-Objective Optimization for Task Offloading in Mobile Edge Computing based on NSGA-II. *Journal of Network and Computer Applications*. 2021.
5. Mach, P., Becvar, Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *IEEE Communications Surveys & Tutorials*. 2017.
6. Aazam, M., Zeadally, S. Fog Computing: The Cloud-to-Thing Continuum. *IEEE Computer*. 2017.
7. Srivastava, T., Patel, J. L., Prakash, L. Fog Computing Applications in Smart Agriculture: A Review. *IEEE Transactions on Industrial Informatics*. 2020.
8. Wang, Y., Sheng, J., Li, C. P2P-Based Dynamic Task Offloading Strategy for Fog Computing in Industrial IoT. *Sensors*. 2020.
9. Hu, B., Ma, C., Liu, Z. Quality of Service and Experience in Mobile Edge Computing: A Survey. *Journal of Communications and Information Networks*. 2022.
10. Law, A. M. *Simulation Modeling and Analysis*. 5th ed. New York: McGraw-Hill, 2014.
11. Mahmud, R., Kotagiri, R., & Buyya, R. (2018). *Fog Computing: A Taxonomy, Survey and Future Directions*. Springer.
12. Chiang, M., & Zhang, T. (2016). *Fog and IoT: An Overview of Research Opportunities*. *IEEE Internet of Things Journal*, 3(6), 854–864.

13. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). *Fog Computing and Its Role in the Internet of Things*. Proceedings of the MCC Workshop on Mobile Cloud Computing.
14. Mukherjee, M., Shu, L., & Wang, D. (2018). *Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges*. IEEE Communications Surveys & Tutorials, 20(3), 1826–1857.
15. Іванов О. П., Соловйов В. В. *Хмарні та туманні обчислення в телекомунікаційних системах*. – К.: КПІ ім. Ігоря Сікорського, 2020.
16. Яковенко В. І., Козлов В. О. *Інтернет речей та обчислювальні технології периферії*. – Харків: ХНУРЕ, 2021.
17. Deng, S., Zhao, H., Fang, X., Yin, J., & Li, F. Multi-Objective Optimization Task Offloading Strategy for Edge Computing Based on NSGA-II. *Applied Sciences*. 2020.
18. Gao, L., Zhao, H., & Liu, Q. Collaborative Task Offloading and Resource Allocation in Multi-Access Edge Computing Based on MOPSO Algorithm. *IEEE Transactions on Industrial Informatics*. 2021.
19. Taleb, T., Samdanis, K., Mapp, G., Skordoulis, D., & Magedanz, T. On Multi-Access Edge Computing: A Survey of the Emerging Edge Computing Paradigm. *IEEE Communications Surveys & Tutorials*. 2017.
20. Li, C., Zhao, X., Zhang, H., Liu, S., & Li, Y. Energy-Efficient Resource Allocation for Mobile Edge Computing in Wireless Networks. *IEEE Transactions on Industrial Electronics*. 2018.
21. Semaan, J., Keshk, M., Alkhaleefah, F., & Alkhazraji, K. A Survey of QoS-Aware Task Offloading Techniques in Fog Computing. *IEEE Internet of Things Journal*. 2023.
22. Zhao, Y., Lin, X., Shi, W., & Chen, J. Edge Computing for Smart Agriculture: A Survey. *IEEE Access*. 2020.
23. Wang, S., Zhang, Y., & Wei, W. Collaborative Task Offloading for Mobile Edge Computing in 5G Networks. *IEEE Wireless Communications Letters*. 2020.

24. Letaief, K. B., Chen, W., Shi, Y., Zhang, J., & Zhang, Y. The Road Towards 6G: Fundamentals, Applications, and Architectural Evolution. *IEEE Communications Magazine*. 2019.
25. Wu, C., Zhang, Y., & Li, X. Task Offloading Based on P2P Collaboration in Heterogeneous Fog Networks. *Applied Soft Computing*. 2021.