МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет: **ННІ Каразінський банківський інститут**
Кафедра: **Інформаційних технологій та математичного моделювання**
Спеціальність: **122 Комп'ютерні науки**
Освітня програма: **Комп'ютерні науки**

Група: АК-21М **денна форма навчання**

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

«ДОСЛІДЖЕННЯ МЕТОДІВ ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОСТІ ТА НАДІЙНОСТІ ІНФРАСТРУКТУРИ ІНТЕРНЕТУ РЕЧЕЙ»

ЗА НАКАЗОМ № 4601-5/3262 ВІД 15 ВЕРЕСНЯ 2025 РОКУ

здобувача вищої освіти **Чистохіна Олександра Юрійовича**

Робота допущена до захисту в ЕК
протокол кафедри ІТММ № 5 від 02.12.2025 р.

В.о. завідувача кафедри ІТММ
PhD
_____**Ковальчук Д. М.**

Науковий керівник
к.ф.-м.н., доцент
_____**Філатова Л. Д.**

м. Харків 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В. Н. Каразіна

Факультет навчально-науковий інститут “Каразінський банківський інститут”
Кафедра інформаційних технологій та математичного моделювання
Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп’ютерні науки
Освітня програма Комп’ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри

Н. І. Стяглик

“15” вересня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ (ПРОЕКТ)

Чистохіна Олександра Юрійовича
(прізвище, ім’я, по батькові студента)

1. Тема роботи «Дослідження методів забезпечення відмовостійкості та надійності інфраструктури Інтернету речей»

Керівник роботи к.ф.-м.н., доцент Філатова Л. Д.

затверджені наказом по університету від “15” вересня 2025 року №4601-5/3262

2. Строк подання студентом роботи 24 листопада 2025 р.

3. Перелік питань, які потрібно розробити:
У розділі 1: Провести аналіз сучасного стану розвитку інфраструктури Інтернету речей, її архітектури та основних вимог до надійності IoT-систем.
У розділі 2: Дослідити показники, методи та моделі забезпечення відмовостійкості в інфраструктурі Інтернету речей; розробити математичну модель багаторівневої відмовостійкості.
У розділі 3: Розробити концепцію, архітектуру та принципи функціонування системи забезпечення відмовостійкості та надійності інфраструктури IoT.
У розділі 4: Провести експериментальні дослідження, оцінити ефективність запропонованих методів і зробити узагальнені висновки щодо їх застосування.

4. План роботи

№ з/п	Назви етапів роботи
1	Вибір здобувачем теми кваліфікаційної магістерської роботи
2	Затвердження плану і завдання кваліфікаційної магістерської роботи
3	Здача кваліфікаційної магістерської роботи керівнику
4	Підпис кваліфікаційної магістерської роботи у керівника
5	Підпис кваліфікаційної магістерської роботи у нормо контролера
6	Допуск завідувачем кафедри до захисту кваліфікаційної магістерської роботи
7	Захист кваліфікаційної магістерської роботи

5. Дата видачі завдання 15 вересня 2025 року

Студент _____ О. Ю. Чистохін
Підпис ініціали, прізвище

Керівник роботи _____ Л. Д. Філатова
підпис ініціали, прізвище

РЕФЕРАТ
НА КВАЛІФІКАЦІЙНУ МАГІСТЕРСЬКУ РОБОТУ
«ДОСЛІДЖЕННЯ МЕТОДІВ ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОСТІ ТА НАДІЙНОСТІ ІНФРАСТРУКТУРИ ІНТЕРНЕТУ РЕЧЕЙ»
Чистохін Олександр Юрійович

Кваліфікаційна магістерська робота містить 82 сторінок, 5 таблиць, 16 рисунків, список використаних джерел із 20 найменувань.

Об'єктом дослідження є процес забезпечення відмовостійкості та надійності інфраструктури Інтернету речей.

Предметом дослідження є методи, моделі та архітектурні рішення, спрямовані на підвищення відмовостійкості та надійності інфраструктури Інтернету речей.

Мета кваліфікаційної магістерської роботи полягає у дослідженні методів забезпечення відмовостійкості та надійності інфраструктури Інтернету речей, а також у розробленні багаторівневої системи, здатної підтримувати стабільне функціонування IoT-мережі в умовах часткових збоїв.

Завданнями кваліфікаційної магістерської роботи є:

- провести аналіз сучасного стану розвитку інфраструктури Інтернету речей та визначити основні фактори, що впливають на її надійність;
- дослідити існуючі методи та моделі забезпечення відмовостійкості в IoT-системах;
- розробити архітектуру багаторівневої системи забезпечення відмовостійкості та надійності;
- реалізувати експериментальну модель та перевірити ефективність запропонованих методів;
- оцінити отримані результати та сформулювати рекомендації щодо підвищення стійкості системи.

Актуальність дослідження зумовлена швидким розвитком технологій Інтернету речей, який передбачає інтеграцію мільярдів пристроїв у глобальну мережу. Зростання складності IoT-інфраструктури супроводжується підвищеними вимогами до її стабільності, безпеки та надійності. Тому розроблення ефективних методів забезпечення відмовостійкості є стратегічно важливим завданням для сталого функціонування критичних систем – промислових, енергетичних, транспортних і побутових.

За результатами дослідження сформовано теоретичні основи забезпечення відмовостійкості інфраструктури IoT, побудовано математичну модель процесу багаторівневої стійкості, розроблено архітектуру системи підвищення надійності та проведено експериментальну оцінку ефективності запропонованих методів.

Практична новизна роботи полягає у створенні багаторівневої системи забезпечення відмовостійкості для інфраструктури Інтернету речей, яка поєднує механізми агентного моніторингу, автоматичного виявлення несправностей та локалізованого відновлення вузлів. Розроблена модель

дозволяє підвищити доступність IoT-системи, скоротити час простою та підвищити загальну стабільність мережі.

Отримані результати можуть бути використані:

- під час розроблення надійних архітектур систем Інтернету речей;
- у промислових, енергетичних, транспортних і побутових комплексах, що вимагають безперервності роботи;
- при створенні платформ моніторингу, діагностики та самовідновлення IoT-мереж у реальному часі.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ РЕЧЕЙ, ВІДМОВОСТІЙКІСТЬ, НАДІЙНІСТЬ, ІОТ-ІНФРАСТРУКТУРА, МАТЕМАТИЧНА МОДЕЛЬ.

ABSTRACT
AT QUALIFICATION MASTER'S WORK

«RESEARCH OF METHODS FOR ENSURING FAULT TOLERANCE AND RELIABILITY OF THE INTERNET OF THINGS INFRASTRUCTURE»

Oleksandr Chystokhin

The master's qualification thesis contains 82 pages, 5 tables, 16 figures, and a list of 20 references.

The object of the research is the process of ensuring fault tolerance and reliability of the Internet of Things infrastructure.

The subject of the research is the methods, models, and architectural solutions aimed at improving the fault tolerance and reliability of IoT infrastructures.

The purpose of the thesis is to study methods for ensuring the fault tolerance and reliability of IoT infrastructure, as well as to develop a multilayer system capable of maintaining stable IoT network operation under partial failure conditions.

The main tasks of the thesis are:

- to analyze the current state of IoT infrastructure development and identify the main factors affecting its reliability;
- to study existing methods and models for ensuring fault tolerance in IoT systems;
- to develop the architecture of a multilayer system for ensuring fault tolerance and reliability;
- to implement an experimental model and test the effectiveness of the proposed methods;
- to evaluate the obtained results and formulate recommendations for improving system resilience.

The relevance of the research is determined by the rapid development of the Internet of Things technologies, which involve integrating billions of devices into a global network. The growing complexity of IoT infrastructures requires increased stability, security, and reliability. Therefore, developing effective methods to ensure fault tolerance is a strategically important task for the sustainable operation of critical systems — industrial, energy, transport, and domestic.

According to the research results, theoretical foundations for ensuring IoT infrastructure fault tolerance have been formed, a mathematical model of the multilayer stability process has been developed, and an architecture of a reliability enhancement system has been designed and experimentally evaluated.

The practical novelty of the thesis lies in the development of a multilayer fault-tolerance system for IoT infrastructure that integrates agent-based monitoring, automatic fault detection, and localized node recovery mechanisms. The proposed model improves system availability, reduces downtime, and enhances overall network stability.

The obtained results can be used:

- in the development of reliable architectures for IoT systems;

- in industrial, energy, transport, and household complexes requiring continuous operation;
- in the creation of platforms for monitoring, diagnostics, and self-recovery of IoT networks in real time.

KEYWORDS: INTERNET OF THINGS, FAULT TOLERANCE, RELIABILITY, IOT INFRASTRUCTURE, MATHEMATICAL MODEL.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧОК, СИМВОЛІВ І ТЕРМІНІВ

IoT – Internet of Things (Інтернет речей)

ICT – Information and Communication Technologies (інформаційно-комунікаційні технології)

API – Application Programming Interface (програмний інтерфейс взаємодії між додатками)

MQTT – Message Queuing Telemetry Transport (протокол телеметричного обміну повідомленнями)

CoAP – Constrained Application Protocol (протокол прикладного рівня для пристроїв з обмеженими ресурсами)

HTTP – HyperText Transfer Protocol (протокол передачі гіпертексту)

JSON – JavaScript Object Notation

REST – Representational State Transfer

DDoS – Distributed Denial of Service (розподілена атака типу «відмова в обслуговуванні»)

QoS – Quality of Service (якість обслуговування в мережі)

MTBF – Mean Time Between Failures (середній час між відмовами)

MTTR – Mean Time To Repair (середній час відновлення працездатності)

DNS – Domain Name System (система доменних імен)

TCP/IP – Transmission Control Protocol / Internet Protocol (набір основних мережевих протоколів Інтернету)

SSL/TLS – Secure Sockets Layer / Transport Layer Security (протоколи захищеної передачі даних)

API Gateway – шлюз прикладного програмного інтерфейсу

Fault Tolerance – відмовостійкість системи

Reliability – надійність системи

Cloud Computing – хмарні обчислення

Edge Computing – периферійні (крайові) обчислення

ML – Machine Learning (машинне навчання)

ВСТУП

Стрімкий розвиток технологій Інтернету речей (Internet of Things, IoT) спричинив появу великої кількості інтелектуальних систем, здатних автоматично збирати, передавати та аналізувати дані з навколишнього середовища. Такі системи активно впроваджуються у промисловості, енергетиці, транспорті, сільському господарстві, охороні здоров'я, побуті тощо. Їх широке поширення створює значні переваги, але водночас породжує низку проблем, серед яких однією з найважливіших є забезпечення надійності та відмовостійкості функціонування інфраструктури IoT.

Інфраструктура Інтернету речей характеризується великою кількістю гетерогенних пристроїв, розподіленою архітектурою та високою динамічністю процесів обміну даними. У таких умовах навіть поодинокі збої можуть призвести до порушення роботи всієї системи, втрати даних або деградації сервісів. Тому розроблення ефективних методів забезпечення відмовостійкості та надійності є одним із ключових завдань сучасних IoT-досліджень.

Забезпечення безперервності роботи систем Інтернету речей вимагає впровадження спеціалізованих механізмів контролю, моніторингу, діагностики стану вузлів та автоматичного відновлення після відмов. На відміну від традиційних обчислювальних систем, IoT-платформи працюють у гетерогенних середовищах із різними рівнями обчислювальних ресурсів, пропускної здатності та надійності каналів зв'язку. Це зумовлює необхідність створення розподілених і багаторівневих моделей відмовостійкості, здатних самостійно реагувати на збої без втручання користувача.

Актуальність роботи полягає в потребі розроблення ефективних методів і засобів підвищення надійності інфраструктури Інтернету речей, здатних забезпечити безперервність роботи у розподілених середовищах, мінімізувати ризик втрати даних і скоротити час простою систем. Ураховуючи зростаючу складність і критичність IoT-рішень, питання

створення відмовостійких архітектур набуває стратегічного значення для розвитку цифрової економіки та промислових систем нового покоління.

Метою роботи є дослідження методів забезпечення відмовостійкості та надійності інфраструктури Інтернету речей, а також розроблення багаторівневої системи, здатної підтримувати стабільне функціонування IoT-мережі в умовах часткових збоїв.

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести аналіз сучасного стану розвитку інфраструктури Інтернету речей і визначити основні фактори, що впливають на її надійність;
- дослідити існуючі методи та моделі забезпечення відмовостійкості в IoT-системах;
- розробити архітектуру багаторівневої системи забезпечення відмовостійкості та надійності;
- реалізувати експериментальну модель і перевірити ефективність запропонованих методів;
- провести оцінку результатів дослідження та визначити напрями подальшого розвитку систем відмовостійкості для IoT.

Об’єктом дослідження є інфраструктура Інтернету речей як розподілена система збору, передавання та оброблення даних.

Предметом дослідження є методи, моделі та архітектурні рішення, спрямовані на забезпечення відмовостійкості та надійності інфраструктури Інтернету речей.

У першому розділі проведено аналіз сучасного стану розвитку IoT-технологій, розглянуто архітектуру інфраструктури, основні характеристики систем та вимоги до їх надійності.

У другому розділі досліджено методи забезпечення відмовостійкості в IoT, визначено основні показники надійності та запропоновано математичну модель процесу забезпечення багаторівневої відмовостійкості.

У третьому розділі розроблено архітектуру системи підвищення надійності й відмовостійкості інфраструктури Інтернету речей, описано

принципи її функціонування та взаємодію компонентів.

У четвертому розділі проведено експериментальну оцінку ефективності запропонованих методів, наведено результати моделювання та зроблено узагальнені висновки щодо підвищення стійкості системи.

Отримані результати мають практичну цінність для розробників і дослідників у сфері IoT, оскільки дозволяють підвищити надійність і стабільність функціонування систем, зменшити ризики збоїв та оптимізувати процеси відновлення. Розроблені підходи можуть бути використані при створенні промислових, енергетичних, транспортних і побутових систем, де важливо забезпечити безперервність роботи в реальному часі.

РОЗДІЛ 1
АНАЛІЗ СУЧАСНОГО СТАНУ РОЗВИТКУ ІНФРАСТРУКТУРИ ІНТЕРНЕТУ РЕЧЕЙ

1.1. Загальна концепція Інтернету речей

Інтернет речей – це сучасна концепція розвитку інформаційних технологій, яка передбачає створення глобальної мережі, що об’єднує фізичні об’єкти, оснащені датчиками, сенсорами, ідентифікаторами та засобами обчислення. Такі пристрої отримують інформацію з навколишнього середовища, обмінюються даними один з одним і з віддаленими системами, забезпечуючи автоматизоване керування, моніторинг і прийняття рішень без прямої участі людини. На практиці це означає, що звичайні побутові та промислові речі – від лампочки, кавоварки чи гаражних дверей до складних виробничих механізмів – стають «розумними» і здатними до взаємодії у єдиній інформаційній інфраструктурі [1].

Термін Internet of Things уперше запропонував Кевін Ештон у 1999 році, який передбачав створення «цифрових двійників» фізичних речей, здатних взаємодіяти в єдиному інформаційному середовищі. Починаючи з 2000-х років, коли кількість пристроїв, підключених до мережі, перевищила число користувачів Інтернету, розпочався перехід від «Інтернету людей» до «Інтернету речей» [1, 2].

Концепція мереж нового покоління (NGN) орієнтувалася на забезпечення безперервного зв’язку між людьми – як безпосередньо, так і через комп’ютерні системи – у будь-який час і з будь-якого місця. На відміну від цього, ідея Інтернету речей розширює цю парадигму, додаючи ще один вимір – можливість взаємодії між будь-якими пристроями та об’єктами (рис. 1.1) [1-3].

Рис. 1.1 Інноваційний напрямок комунікацій, реалізований через Інтернет речей.

Концепція IoT ґрунтується на трьох основоположних принципах [1, 2]:

- універсальна комунікація – можливість кожного пристрою підключатися до мережі незалежно від технологічної платформи;
- глобальна ідентифікація – кожен об'єкт має унікальний ідентифікатор, що дозволяє відслідковувати його стан і місцезнаходження;
- двоспрямований обмін даними – здатність як надсилати інформацію у мережу, так і отримувати керуючі сигнали.

IoT-системи об'єднують сенсори, мікроконтролери, комунікаційні модулі та виконавчі механізми в єдину мережу [2]. У спрощеній формулі концепцію можна подати так:

фізичний об'єкт + сенсори/ідентифікатори
+ контролер + мережа = IoT.

Особливістю Інтернету речей є орієнтація не на людину, а на «речі». Кількість таких пристроїв уже сьогодні вимірюється десятками мільярдів, і вона у багато разів перевищує число інтернет-користувачів. Пристрої в цій системі зазвичай компактні, енергоефективні та мають невисоку швидкість передавання даних, але працюють у масових масштабах і формують

величезні обсяги інформації. Це вимагає нових підходів до зберігання, обробки й інтерпретації даних, а також розвитку спеціальних протоколів і стандартів [3].

Суттєвими характеристиками IoT є [2]:

- взаємозв’язаність – інтеграція різнорідних пристроїв у спільні мережеві структури;
- орієнтація на сервіси – забезпечення узгодженої взаємодії фізичних об’єктів у цифровому середовищі;
- гетерогенність – здатність інтегрувати пристрої з різними технологіями та архітектурами;
- динамічність – постійна зміна кількості активних пристроїв і їхніх параметрів;
- масштабність – багатократне зростання числа підключень і транзакцій у порівнянні з традиційним Інтернетом людей.

Завдяки цим властивостям IoT здатний радикально змінювати економічні й соціальні процеси (рис. 1.2). У сфері охорони здоров’я вже сьогодні використовуються носимі сенсори для моніторингу стану пацієнтів у режимі реального часу. У побуті «розумні будинки» забезпечують автоматичне керування освітленням, кліматом чи побутовими приладами. В енергетиці та промисловості IoT сприяє раціональному використанню ресурсів, зменшенню витрат і підвищенню безпеки виробництва [3, 4].

Таким чином, загальна концепція Інтернету речей полягає у створенні масштабної, багаторівневої, самоорганізованої мережі фізичних об’єктів, що сприймають, обробляють і передають інформацію. Вона спрямована на те, щоб значна частина рутинних дій і операцій здійснювалася автоматично, а людина зосереджувалася на стратегічних і творчих завданнях. Це не лише розширює технологічні можливості, але й формує нові умови для бізнесу, медицини, екології та повсякденного життя.

Рис. 1.2. Використання Інтернету речей

Однак однією з головних проблем розвитку Інтернету речей залишається забезпечення відмовостійкості. Зі збільшенням кількості підключених пристроїв та ускладненням мережевої інфраструктури зростає ймовірність відмов, що може негативно впливати на надійність системи. Тому створення нових рішень для забезпечення стійкості до збоїв є критично важливим завданням для подальшого розвитку IoT.

1.2. Огляд інфраструктури Інтернету речей

Інтернет речей – це складна система взаємопов’язаних фізичних пристроїв, сенсорів, актуаторів та програмного забезпечення, що забезпечує збір, передачу, обробку та аналіз даних без необхідності постійної участі людини. За оцінками експертів, до 2030 року кількість IoT-пристроїв в експлуатації перевищить 25 мільярдів. Це ставить високі вимоги до надійності інфраструктури IoT, оскільки будь-які збої в мережі можуть спричинити значні втрати – від фінансових до загроз для життя людей.

Надійне функціонування IoT забезпечується комплексною інфраструктурою, яка включає апаратні, мережеві, обчислювальні, аналітичні та користувацькі компоненти. Типова інфраструктура IoT, що показана на рис. 1.3, охоплює шість ключових елементів, кожен із яких виконує важливі функції та тісно інтегрований із іншими складовими системи [4, 5].

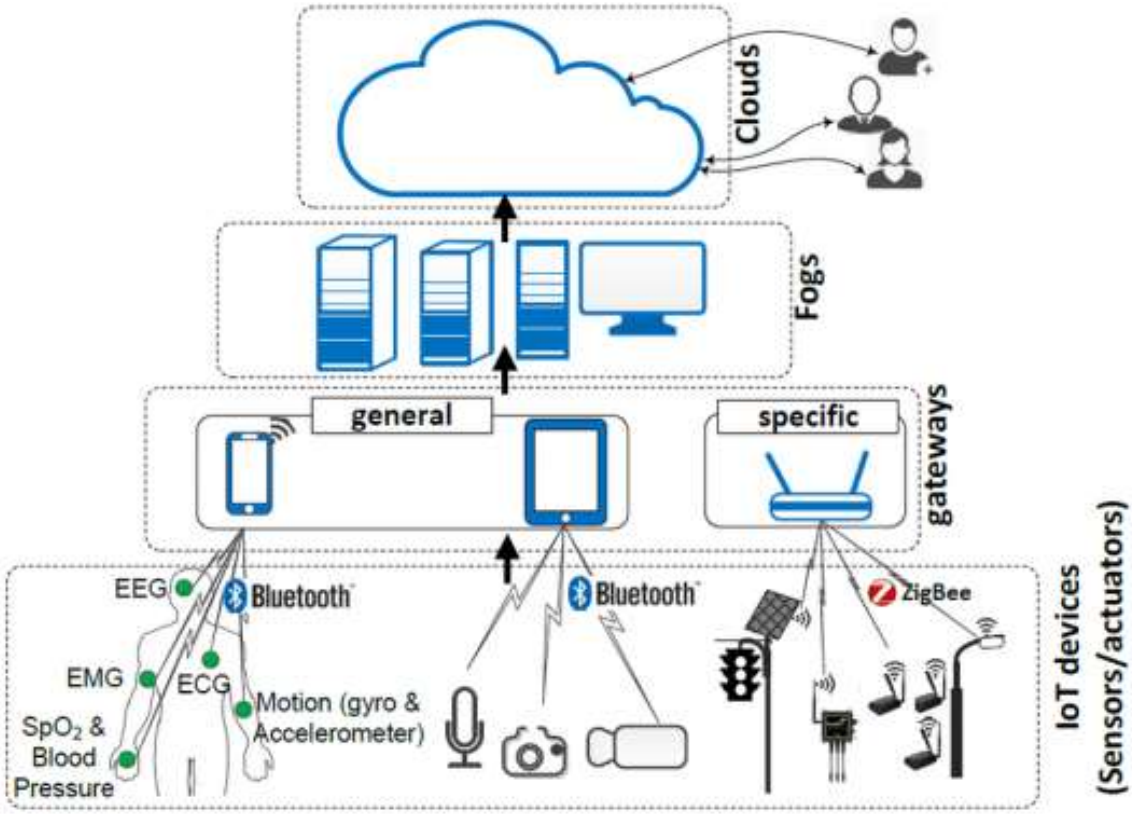


Рис. 1.3. Схема багаторівневої архітектури інфраструктури IoT

1. IoT-пристрої (сенсори та актуатори). IoT-пристрої є первинними елементами системи, які здійснюють збір інформації або реагування на зміну параметрів навколишнього середовища. Сенсори вимірюють фізичні характеристики об’єктів або явищ – температуру, тиск, вологість, освітленість, швидкість руху та місце розташування – і перетворюють їх на цифрові сигнали, які можуть оброблятися системою. Актуатори виконують фізичні дії на основі отриманих команд, наприклад, вмикають освітлення,

регулюють температуру, відкривають або закривають клапани, активують механізми в промислових процесах [1, 3, 6].

Сучасні сенсори характеризуються високою мініатюризацією, що дозволяє інтегрувати їх безпосередньо в об’єкти фізичного світу, створюючи «розумні» системи. Частина сенсорів формує бездротові сенсорні мережі (WSN, Wireless Sensor Networks), що забезпечують збір даних на великих територіях та здатні працювати від автономних джерел живлення.

2. Комунікаційні мережі. Передача даних, отриманих сенсорами, здійснюється за допомогою комунікаційних мереж, які можуть бути як провідними, так і бездротовими. До провідних належать Ethernet та інші локальні мережі (LAN), тоді як бездротові технології включають Wi-Fi, Bluetooth, Zigbee, LoRaWAN, NB-IoT та 5G [1, 3, 6].

Вибір конкретної мережевої технології залежить від характеристик пристроїв, обсягу переданих даних, енергоспоживання та необхідної дальності зв’язку. Наприклад, LoRaWAN та NB-IoT підходять для передачі невеликих обсягів даних на великі відстані з мінімальним споживанням енергії, що важливо для сільського господарства або екологічного моніторингу. 5G забезпечує високу швидкість передачі даних і низьку затримку, що критично для систем, де потрібна миттєва реакція, таких як автономні транспортні засоби або промислові роботизовані комплекси.

Комунікаційні мережі повинні підтримувати міжмашинні взаємодії (M2M) та забезпечувати необхідну якість обслуговування з точки зору пропускної здатності, затримки та безпеки переданих даних. Усі мережі в IoT інтегруються у конвергентну мережеву платформу, яка дозволяє спільне використання ресурсів різними пристроями без шкоди для продуктивності та конфіденційності.

3. Шлюзи (Gateways). Шлюзи є проміжними пристроями, що з’єднують локальні мережі IoT-пристроїв із хмарними сервісами або серверними рішеннями. Вони виконують кілька критичних функцій: перетворення протоколів для забезпечення сумісності різних пристроїв, фільтрацію та

попередню обробку даних для зменшення навантаження на центральні обчислювальні платформи, забезпечення безпеки та конфіденційності даних.

Шлюзи також можуть здійснювати локальну обробку інформації, наприклад, виконувати первинний аналіз сигналів сенсорів або реагувати на прості події без передачі даних у хмару, що дозволяє знижувати затримки та підвищувати швидкодію системи [4].

4. Обчислювальні платформи (Edge, Fog, Cloud). Обчислювальні платформи відповідають за обробку, аналіз та зберігання даних, отриманих від IoT-пристроїв. Існують три основні моделі обчислень [1, 5]:

- Edge Computing – обробка даних безпосередньо на пристрої або поблизу нього, що забезпечує мінімальні затримки та зменшує навантаження на мережу;
- Fog Computing – розподілена обробка на проміжних пристроях між IoT-пристроями та хмарою, що дозволяє ефективно управляти великими потоками даних та здійснювати аналітику в реальному часі;
- Cloud Computing – централізоване зберігання та обробка даних на хмарних платформах, що забезпечує масштабованість та доступ до потужних обчислювальних ресурсів.

5. Аналітичні сервіси. Аналітичні сервіси виконують функції обробки та аналізу даних, отриманих від сенсорів та актуаторів. Вони включають системи виявлення закономірностей, прогнозування, виявлення аномалій та автоматичного реагування. Для цього використовуються методи статистичного аналізу, машинного навчання та штучного інтелекту [1, 3, 6].

Аналітичні сервіси дозволяють автоматизувати прийняття рішень і оптимізувати роботу системи. Наприклад, у промислових додатках вони можуть передбачати збої обладнання та запускати превентивне обслуговування, а у транспорті – прогнозувати потоки руху та оптимізувати маршрути.

6. Засоби керування та інтерфейси користувача. Інфраструктура IoT включає інтерфейси, які забезпечують взаємодію користувача із системою.

Це мобільні додатки, веб-інтерфейси, панелі управління та голосові помічники, що дозволяють моніторити стан пристроїв, здійснювати управління, отримувати сповіщення та виконувати оновлення програмного забезпечення [1, 2].

1.3. Основні характеристики та вимоги до забезпечення надійності IoT-систем

Можливості застосування технологій Інтернету речей є надзвичайно широкими та різноманітними, і вже сьогодні вони активно впливають на ключові сфери економіки та суспільства – від промислового виробництва та транспорту до медицини й сільського господарства. Подальше зростання кількості IoT-пристроїв, підключених до глобальної мережі, визначатиме все більшу роль цієї технології у трансформації нашого способу життя, роботи та комунікацій.

Водночас одним із найбільш серйозних викликів для розвитку Інтернету речей є забезпечення відмовостійкості. Масове підключення пристроїв і зростаюча складність мережевої інфраструктури збільшують ризик збоїв та зниження надійності функціонування системи. Тому створення сучасних рішень для підвищення стійкості до відмов набуває стратегічного значення та є необхідною умовою подальшого розвитку IoT.

Інфраструктура Інтернету речей має низку специфічних особливостей, які відрізняють її від традиційних IT-систем і суттєво впливають на рівень надійності. Складна розподілена архітектура, динамічна топологія мереж, гетерогенність пристроїв та обмежені ресурси – усе це створює значні труднощі для стабільного функціонування систем. У такому середовищі питання забезпечення безперервності роботи, відмовостійкості та здатності до відновлення після збоїв набуває особливого значення [6, 7].

Однією з ключових характеристик IoT є високий рівень розподіленості. Пристрої часто розташовані у віддалених або важкодоступних місцях, де

можливість фізичного обслуговування обмежена, а умови навколишнього середовища можуть негативно впливати на працездатність обладнання. Це знижує ефективність ручного втручання у випадку відмови й вимагає наявності автоматизованих механізмів самодіагностики та відновлення [7].

Ще однією важливою властивістю є гетерогенність компонентів. IoT-пристрої відрізняються за архітектурою, операційними системами, протоколами зв’язку, обчислювальними можливостями й енергетичними профілями. Відсутність єдиних стандартів ускладнює інтеграцію, управління та створення універсальних рішень для підвищення надійності. Особливої уваги потребує сумісність між різними протоколами зв’язку, такими як Wi-Fi, Bluetooth, Zigbee, LoRaWAN чи NB-IoT, адже саме вона визначає стабільність передачі даних [6, 7].

Динамічність топології є ще одним фактором, що ускладнює забезпечення безперервності роботи. У багатьох сценаріях пристрої можуть підключатися й відключатися від мережі залежно від умов експлуатації чи режимів енергозбереження. Крім того, канали зв’язку постійно змінюються під впливом фізичних перешкод, мобільності пристроїв або інтерференції, що може спричиняти затримки, втрати пакетів і тимчасові збої у функціонуванні системи.

Не менш суттєвою проблемою є обмежені ресурси IoT-вузлів. Обмеження у вигляді невеликих запасів енергії, малої обчислювальної потужності чи недостатнього обсягу пам’яті роблять неможливим використання класичних підходів до резервування, дублювання компонентів чи складних алгоритмів відмовостійкості. У таких умовах актуальними стають оптимізовані методи управління ресурсами, легкі протоколи передачі даних та енергоефективні стратегії функціонування.

Варто також підкреслити залежність IoT-інфраструктури від фізичного середовища. Пристрої часто працюють у жорстких умовах – при високій вологості, екстремальних температурах, вібраціях чи запиленості. Такі фактори значно підвищують ризик апаратних збоїв і потребують врахування

у процесі проєктування систем. Додатково важливим завданням є досягнення інтероперабельності – здатності різних пристроїв і сервісів взаємодіяти між собою. Нестача стандартизації або різні підходи виробників можуть знижувати загальну надійність системи.

У контексті забезпечення надійності IoT-систем зазвичай виділяють кілька ключових вимог. По-перше, це доступність (availability), яка означає готовність сервісу до роботи у визначений момент часу. По-друге, відмовостійкість (fault tolerance), що передбачає збереження працездатності навіть у разі збоїв окремих компонентів. Третьою вимогою є безперервність роботи (continuity), яка є критичною для сфер охорони здоров'я, транспорту чи енергетики, де збої можуть мати катастрофічні наслідки. Важливим показником також виступає відновлюваність (recoverability), тобто здатність системи повертатися до стабільного стану після збою. Окрім цього, надійність включає точність і своєчасність обробки даних, адже некоректні чи спотворені показники можуть зробити систему непридатною до використання [6, 7].

Для аналізу та моделювання надійності IoT-систем часто застосовуються формальні методи, зокрема дерева відмов (Fault Tree Analysis), Марковські моделі та мережі Петрі. Вони дозволяють оцінити ймовірність відмов, передбачити поведінку системи у різних сценаріях та визначити критичні вузли, від яких залежить стабільність роботи.

Серед основних загроз для надійності IoT-інфраструктури можна виділити втрату підключення, апаратні несправності, програмні помилки, перевантаження каналів зв'язку, енергетичне виснаження пристроїв та кібератаки, зокрема відмову в обслуговуванні чи підміну трафіку. З огляду на це, питання інтеграції механізмів безпеки стає невід'ємною складовою забезпечення надійності.

Таким чином, надійність IoT-систем є багатогранною категорією, яка охоплює як технічні, так і організаційні аспекти. Для побудови ефективної та стійкої інфраструктури необхідно враховувати особливості розподіленості,

різнорідності та обмеженості ресурсів, а також інтегрувати методи резервування, автоматизованого відновлення та захисту від загроз. Лише комплексний підхід до цих завдань дозволить створити стійкі та масштабовані рішення, здатні задовольнити зростаючі вимоги до IoT у майбутньому.

1.4. Висновки до розділу 1

У першому розділі проведено аналіз сучасного стану розвитку інфраструктури Інтернету речей, визначено її структурні складові, ключові характеристики та основні виклики, пов’язані із забезпеченням відмовостійкості й надійності.

Показано, що Інтернет речей є складною багаторівневою системою, у якій взаємодіють мільярди пристроїв різних типів. Його архітектура включає сенсори, мережеві засоби, шлюзи, обчислювальні платформи, аналітичні сервіси та інтерфейси користувача, що утворюють єдину екосистему обміну даними. Водночас така масштабність і гетерогенність породжує низку проблем – збої зв’язку, втрати даних, енергетичні обмеження та вразливість до кібератак.

Зроблено висновок, що підвищення надійності IoT неможливе без впровадження спеціалізованих методів забезпечення відмовостійкості, здатних автоматично виявляти, локалізувати та компенсувати відмови на різних рівнях системи. Розроблення та дослідження таких методів є ключовим напрямом подальших розділів цієї роботи.

РОЗДІЛ 2
МЕТОДИ ТА МОДЕЛІ ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОСТІ В ІНФРАСТРУКТУРІ ІНТЕРНЕТУ РЕЧЕЙ

2.1. Показники забезпечення відмовостійкості у мережах IoT

Забезпечення відмовостійкості у мережах Інтернету речей (IoT) є критично важливим аспектом, оскільки навіть короткочасний збій може призвести до втрати даних, переривання сервісів та фінансових чи технологічних втрат. Традиційні підходи до аналізу відмовостійкості базуються на оцінці надійності систем і компонентів. Надійність системи визначається як ймовірність того, що вона виконає свої функції без збоїв протягом певного часу за заданих умов експлуатації. В контексті IoT цю концепцію застосовують для оцінки стабільності роботи пристроїв, шлюзів і хмарної інфраструктури [6, 7].

Для кількісної оцінки відмовостійкості застосовуються такі ключові показники. Перш за все, це середній час між відмовами (MTBF, Mean Time Between Failures), який визначає очікуваний проміжок часу між двома послідовними відмовами системи або пристрою [7, 8]. Формально MTBF розраховується як:

$$MTBF = \frac{T_{сум}}{N},$$

де $T_{сум}$ - загальний час роботи, а N - кількість відмов за цей період. У IoT це особливо важливо для сенсорів, встановлених у віддалених або важкодоступних місцях, де фізичний доступ для ремонту обмежений.

Додатково використовується середній час відновлення після відмови (MTTR, Mean Time To Repair), який характеризує ефективність реагування на збої та швидкість відновлення працездатності системи [7, 8]:

$$MTTR = \frac{N_{trec}}{N_{rec}},$$

де N_{trec} - сумарний час, витрачений на ремонт, а N_{rec} - кількість ремонтних операцій. У мережах IoT низький MTTR дозволяє зменшити час простою критично важливих сервісів, таких як промислова автоматизація, медичний моніторинг чи системи «розумного транспорту».

На основі MTBF і MTTR визначається доступність (Availability), яка відображає частку часу, протягом якого пристрій або система здатні виконувати свої функції без збоїв. Доступність розраховується за формулою:

$$Availability = \frac{MTBF}{MTBF + MTTR} \times 100\%.$$

Для IoT високий рівень доступності (наприклад, 99,9% і вище) є критично необхідним, оскільки багато сервісів потребують безперервного збору та обробки даних у реальному часі [7, 8].

Ще одним важливим показником є час виявлення відмови t_d, який характеризує період від початку виникнення помилки до моменту її виявлення:

$$t_d = t_{end} - t_{start},$$

де t_{start} - час появи збоїв, а t_{end} - час їх виявлення. Чим менший цей показник, тим швидше система може реагувати на проблему та активувати механізми відновлення. Сучасні IoT-системи використовують автоматизовані алгоритми моніторингу та предиктивної аналітики на основі машинного навчання, що дозволяє мінімізувати t_d і запобігати масштабним збоям [7, 8].

Крім базових показників, у практичних рішеннях застосовують додаткові метрики, такі як MTTD (Mean Time to Detect) – середній час до виявлення відмови, MTTF (Mean Time To Failure) – середній час до першої відмови, FAR (Failure Alarm Rate) – частота помилкових сигналів про відмову, а також SLA Availability, який характеризує рівень доступності, гарантований провайдером IoT-сервісу. Ці показники дозволяють комплексно оцінити надійність і ефективність багаторівневих IoT-інфраструктур, від окремих сенсорів до масштабних хмарних систем [8-10].

Таким чином, використання MTBF, MTTR, доступності та часу виявлення відмови, а також розширених метрик MTTD, MTTF і FAR, забезпечує систематичний підхід до оцінки та підвищення відмовостійкості IoT-систем. Це дає змогу планувати резервні механізми, автоматизовані стратегії відновлення та контроль критично важливих сервісів, що особливо актуально для промислового Інтернету речей (IIoT), «розумних міст» та автономних транспортних систем.

2.2. Методи забезпечення відмовостійкості у мережах IoT

Забезпечення відмовостійкості у мережах IoT є критичним аспектом, оскільки ці системи включають велику кількість розподілених пристроїв, сенсорів та шлюзів, які можуть зазнавати збоїв або ставати мішенню для кібератак. Одним із традиційних підходів до управління відмовами є алгоритм звітування про призначення (Destination Reporting Algorithm, DRA), який застосовується для виявлення проблем у передачі даних та оперативного реагування на них.

У класичній реалізації цього алгоритму процес передачі даних відбувається наступним чином: відправник надсилає пакети до призначеного вузла, який отримує їх та перевіряє цілісність і повноту доставки. Якщо вузол призначення виявляє будь-які помилки або втрати пакетів, він відправляє повідомлення про відмову назад до відправника. Після отримання такого повідомлення джерело може виконати необхідні дії: повторну відправку пакетів, перенаправлення трафіку або тимчасове призупинення передачі до усунення проблеми. Основна мета алгоритму – забезпечити надійну доставку даних і прозоре повідомлення про будь-які збої, що виникають у мережі [10].

Проте класичний алгоритм звітування має ряд обмежень, які стають критичними у контексті IoT:

- недостатня ефективність у масштабних мережах, оскільки у великих розподілених мережах IoT з тисячами сенсорів та актуаторів

традиційний DRA може створювати додаткове навантаження на мережу через обмін повідомленнями про збої, що призводить до затримок у реакції;

- обмежена адаптованість до специфіки IoT. Більшість класичних алгоритмів не враховують обмежені ресурси вузлів (енергію, обчислювальну потужність, пам’ять) та різнорідність пристроїв у мережі. Це знижує ефективність алгоритмів у динамічних та ресурсно обмежених середовищах;
- помилки виявлення та локалізації відмов. Традиційні методи можуть бути недостатньо точними у визначенні місця відмови, особливо коли мова йде про багаторівневі або динамічно змінні мережеві топології;
- нестабільність у динамічних умовах. IoT-мережі часто працюють у середовищах із мінливими параметрами зв’язку (наприклад, мобільні сенсорні мережі, бездротові сенсорні мережі з низьким енергоспоживанням). Зміни топології або якості каналів можуть впливати на ефективність традиційних методів управління відмовами.

Для подолання цих обмежень сучасні дослідження пропонують модифіковані та комбіновані методи управління відмовами, адаптовані до особливостей IoT [10, 11]. До них належать:

1) розподілене звітування. Повідомлення про відмову передається не тільки від вузла призначення до джерела, а й до проміжних вузлів або контролерів, що дозволяє локалізувати та усунути проблему ближче до її джерела;

2) ієрархічна структура моніторингу. У великих мережах IoT створюють багаторівневі системи спостереження, де сенсори та шлюзи надсилають сигнали про стан у локальні вузли, а ті вже агрегують інформацію для центрального контролера. Це зменшує навантаження на мережу та прискорює виявлення збоїв;

3) пріоритетна обробка повідомлень. Використання черг та пріоритетів для повідомлень про відмови дозволяє швидше реагувати на критичні збої та забезпечує більш стабільну роботу системи;

4) адаптивні алгоритми повторної передачі. На основі прогнозів та машинного навчання вузли можуть визначати оптимальні шляхи та час повторної відправки пакетів, зменшуючи ризик перевантаження мережі;

5) комбінація з механізмами надлишковості, як показано на рис. 2.1. Використання резервних пристроїв, дублікатів даних і альтернативних маршрутів передачі підвищує відмовостійкість системи, зменшуючи вплив відмов окремих вузлів.

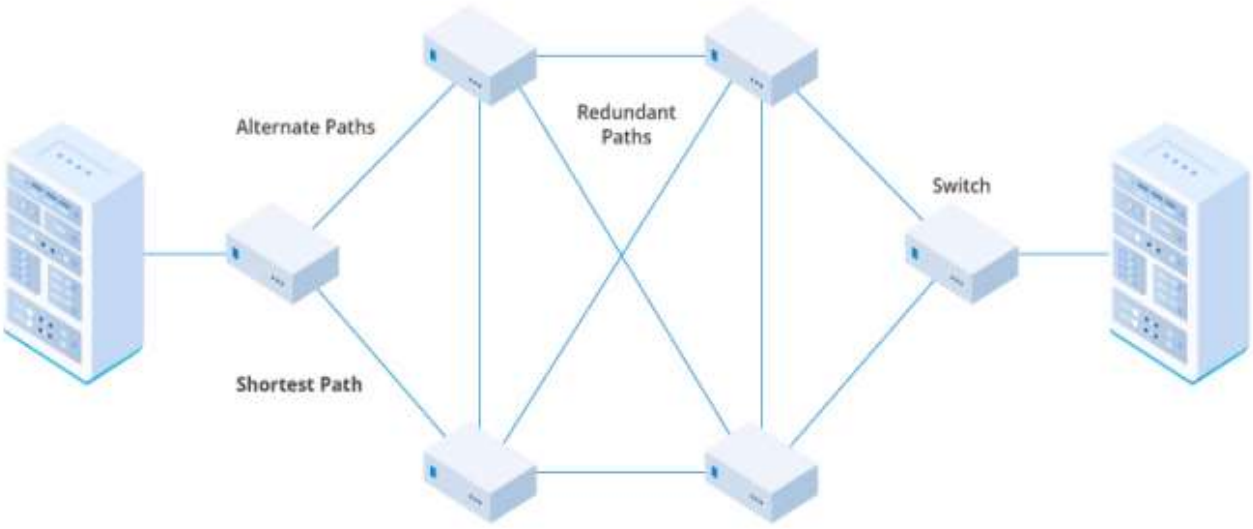


Рис. 2.1. Забезпечення надійності за рахунок надлишковості шляхів

Таким чином, сучасні методи забезпечення відмовостійкості у мережах IoT базуються на поєднанні класичних підходів (як-от алгоритм звітування про призначення) з розподіленими, адаптивними та резервними механізмами, що дозволяє створювати більш стійкі до збоїв мережі. Ці методи враховують обмеження ресурсів, динамічність топології та гетерогенність пристроїв, що є характерними для IoT, і забезпечують безперервну та надійну роботу системи навіть у складних умовах.

2.3. Математична модель процесу забезпечення багаторівневої відмовостійкості у мережі Інтернету речей

Розробка математичної моделі процесу забезпечення багаторівневої відмовостійкості у мережі Інтернету речей є важливим етапом проектування самої системи забезпечення надійності. Модель дозволяє формально описати структуру мережі, взаємодію компонентів, ключові процеси реагування на відмови, а також оцінити потенційні ризики та виробити стратегії їх мінімізації.

Для формалізації процесу забезпечення відмовостійкості використовується теоретико-множинний підхід, який включає визначення елементів мережі, відносин між ними та функцій взаємодії. Основні елементи моделі включають [8-10]:

- компоненти мережі – сенсори, вузли, шлюзи та інші пристрої;
- процеси реагування на несправності – механізми виявлення, локалізації та виправлення збоїв;
- системи моніторингу та керування – агенти та протоколи обміну інформацією;
- механізми запобігання поширенню несправностей – локальні та глобальні стратегії обмеження впливу збоїв.

Метою системи забезпечення відмовостійкості є мінімізація часу виявлення та усунення несправностей, що формалізується функцією [10, 11]:

$$fg: E \times C \to min(t_d, t_m),$$

де t_d – час виявлення несправностей, t_m – час виправлення або пом’якшення впливу несправностей, E – множина можливих несправностей, а C – компоненти мережі.

Мережу IoT представляємо як множину ієрархічних рівнів:

$$MIOT = \{L_i\}_{i=1}^{k},$$

де L_i – і-й рівень мережі. Кожен рівень описується кортежем:

$$L_i = \langle Nodes, Links, Routes, Protocols \rangle,$$

де: *Nodes* – вузли (пристрої, сенсори, актуатори);

Links – з'єднання між вузлами (провідні та бездротові, наприклад Wi-Fi, Zigbee, LoRa);

Routes – маршрути пересилання даних між вузлами;

Protocols – протоколи обміну даними (MQTT, CoAP, Zigbee, LoRaWAN).

Передачу даних у мережі задаємо функцією:

$$f_p : d_{l_{i-1}} \to d_{l_i} \to d_{l_{i+1}},$$

Можливі несправності у процесі передачі даних описуються функцією:

$$f_e = e_{l_{i-1}} \vee e_{l_i} \vee e_{l_{i+1}},$$

де e_{l_i} – подія несправності на рівні L_i, а ∨ операція диз'юнкції означає, що загальна несправність виникає, якщо хоча б один з рівнів зазнав збою. Такий підхід дозволяє врахувати розповсюдження несправностей у мережі та їхній вплив на сумарний час обробки запитів.

Для реалізації багаторівневої відмовостійкості вводиться агент системи забезпечення відмовостійкості (СЗВ) на кожному рівні L_i, який визначається кортежем функцій (рис. 2.2) [11, 12]:

$$A_{l_i} = \langle a_m, a_r, a_e, a_c \rangle,$$

де: a_m – моніторинг стану вузлів (перевірка доступності, завантаження, час відгуку);

a_r – керування відновленням (ініціація перезавантаження чи відновлення підключення);

a_e – обмін повідомленнями (через MQTT або інші брокери для повідомлень про відмови);

a_c – координація дій із іншими агентами для запобігання поширенню відмов.

Рис. 2.2. Взаємодія компонентів агента системи забезпечення відмовостійкості

Кожен агент функціонує через локальні механізми відмовостійкості $FT_m = \{ ft_i \}_{i=1}^{k}$, наприклад:

- ft_1 – обмеження поширення несправностей;
- ft_2 – активний моніторинг.

Механізм обмеження поширення реалізується ланцюгом функцій:

$$ft_1 = \{ f_1 \rightarrow f_2 \rightarrow f_3 \rightarrow f_4 \rightarrow f_5 \rightarrow f_6 \},$$

де f_1 – виявлення недоступності ресурсу,

f_2 – перехоплення нових запитів,

f_3 – оголошення ресурсу недоступним,

f_4 – активація перевірки,

f_5 – перевірка пристрою,

f_6 – завершення перевірки та відновлення доступу.

Механізм активного моніторингу базується на gossip-алгоритмі, що дозволяє розподілено обмінюватися інформацією про стан вузлів та швидко виявляти несправності [11-13]:

$$ft_2 = \{ f_1 \rightarrow f_2 \rightarrow f_3 \rightarrow f_4 \rightarrow f_5 \},$$

де f_1 – постійний моніторинг на всіх рівнях,

f_2 – виявлення несправностей через gossip,

f_3 – поширення інформації про несправності,

f_4 – перехоплення запитів до недоступних вузлів,

f_5 – відновлення після повернення вузла до активного стану.

З урахуванням агентів система мережі IoT описується множиною:

$$MIOT = \{L, b, A\}_{i=1}^{k},$$

де b – шина подій для обміну повідомленнями між агентами, а $A = \{A_{l_i}\}_{i=1}^{N_l}$ – множина агентів СЗВ на всіх рівнях.

Таким чином, запропонована математична модель описує багаторівневу відмовостійкість у мережі IoT, формалізує взаємодію компонентів та агентів, визначає процеси моніторингу, локалізації та усунення несправностей і створює основу для проектування ефективної та стійкої системи управління відмовами.

2.4. Висновки до розділу 2

У другому розділі проведено всебічний аналіз методів і моделей забезпечення відмовостійкості в інфраструктурі Інтернету речей. Визначено основні кількісні показники надійності систем (MTBF, MTTR, Availability, MTTD, MTTF, FAR), які дозволяють оцінювати стійкість системи до збоїв і втрат працездатності.

Розглянуто класичні й сучасні підходи до підвищення відмовостійкості, включаючи алгоритми DRA, ієрархічний моніторинг, адаптивні стратегії

повторної передачі даних, механізми надлишковості та інтелектуальні методи прогнозування відмов.

Запропонована математична модель процесу забезпечення багаторівневої відмовостійкості формалізує структуру IoT-мережі, описує функції агентів СЗВ, визначає процеси моніторингу, локалізації та усунення несправностей, а також передбачає обмін повідомленнями через єдину шину подій.

Результати розділу формують теоретичне підґрунтя для подальшої розробки та дослідженню системи підвищення багаторівневої відмовостійкості та надійності інфраструктури Інтернету речей.

РОЗДІЛ 3
РОЗРОБКА СИСТЕМИ ПІДВИЩЕННЯ БАГАТОРІВНЕВОЇ ВІДМОВОСТІЙКОСТІ ТА НАДІЙНОСТІ ІНФРАСТРУКТУРИ ІНТЕРНЕТУ РЕЧЕЙ

3.1. Ключові проблеми та виклики у забезпеченні відмовостійкості систем IoT

У сучасних умовах швидкого розвитку технологій та розширення сфери застосування Інтернету речей питання відмовостійкості набуває все більшої актуальності. Попри це, у багатьох випадках під час розроблення IoT-рішень основна увага приділяється функціональності, зручності користування та швидкості обробки даних, тоді як стійкість систем до збоїв і помилок розглядається як другорядна характеристика. Проте навіть незначна відмова окремого елемента IoT-інфраструктури може спричинити серйозні наслідки – від втрати даних і порушення безпеки до зупинки критично важливих сервісів. Саме тому розроблення ефективних підходів до забезпечення відмовостійкості є одним із ключових завдань сучасної теорії та практики побудови систем Інтернету речей.

Однією з головних проблем, що ускладнює створення відмовостійких IoT-систем, є гетерогенність їхньої структури. Інтернет речей складається з великої кількості різнотипних пристроїв – від мікроконтролерів і сенсорів до промислових роботів, серверів і хмарних платформ. Вони можуть використовувати різні технології, протоколи обміну даними, формати збереження інформації та апаратні архітектури. Як правило, така система має багаторівневу структуру, де кожен рівень виконує власні функції: сенсорний шар збирає дані, комунікаційний забезпечує передачу, а прикладний здійснює аналітику та прийняття рішень. Відповідно, кожен шар має власні вимоги до надійності та власні механізми реагування на відмови. Інтеграція цих механізмів у єдину систему відмовостійкості стає складним завданням.

Для подолання цього виклику необхідні універсальні стратегії, які враховують гетерогенність компонентів і дозволяють координувати роботу різнорідних елементів через стандартизовані інтерфейси, протоколи або проміжне програмне забезпечення (middleware) [13, 14].

Наступним важливим аспектом є пристосованість та еволюція системи. Інфраструктура Інтернету речей є динамічною – пристрої можуть підключатися, відключатися, замінюватися або оновлюватися без зупинки всієї системи. Крім того, поява нових стандартів зв’язку, технологій сенсорики та аналітичних інструментів призводить до постійних змін архітектури мережі. Тому механізми відмовостійкості повинні бути адаптивними, тобто здатними перебудовуватися у режимі реального часу. У багатьох випадках IoT-системи функціонують у критичних галузях – транспорті, медицині, енергетиці – де навіть короткочасна зупинка може призвести до серйозних наслідків. Отже, рішення для підвищення надійності повинні підтримувати безперервність роботи та забезпечувати оперативне реагування на зміни стану системи, не перериваючи основні процеси [14].

Суттєвим викликом є також розподілене прийняття рішень. Через масштабність і складність IoT-інфраструктури централізовані моделі управління стають неефективними. Коли тисячі або мільйони пристроїв взаємодіють між собою, координація з одного центру може створювати вузьке місце, затримки та підвищений ризик єдиної точки відмови. Тому сучасні підходи до відмовостійкості орієнтовані на децентралізацію – тобто делегування прийняття рішень ближчим до джерела проблеми вузлам або агентам, які мають більше контекстної інформації про стан мережі. Такі системи повинні мати механізми локального моніторингу, самодіагностики та самоорганізації, що дозволяє швидко виявляти збої та відновлювати роботу без залучення центрального керування. Водночас узгодження дій між розподіленими компонентами без повного уявлення про стан всієї системи залишається складним завданням, яке вимагає розробки гнучких алгоритмів координації.

Ще однією проблемою виступає інформаційна надлишковість, або необхідність забезпечення надійності передавання даних без суттєвого збільшення витрат і складності системи. Апаратне резервування в IoT є обмежено можливим через велику кількість пристроїв і високу вартість їх дублювання. У багатьох випадках мережі налічують тисячі або навіть мільйони сенсорів, тому створення фізичних копій елементів є економічно недоцільним. Вихід полягає у використанні програмної або інформаційної надлишковості – тобто дублюванні чи розподіленні даних у такий спосіб, щоб система могла зберігати цілісність інформації навіть при відмові окремих вузлів. Наприклад, застосування методів кодування з виправленням помилок, реплікації даних між сусідніми пристроями чи резервного буферизування на проміжних вузлах дозволяє досягти високого рівня надійності без збільшення апаратних витрат [13, 14].

Інформаційна надмірність відіграє ключову роль також у процесах виявлення та локалізації відмов. Зайві або дубльовані дані допомагають системі підтвердити достовірність інформації, виявити суперечливі або помилкові значення, а також визначити, у якому саме вузлі сталася помилка. Проте надлишковість має бути розумно збалансованою: надмірне дублювання може перевантажити комунікаційні канали, знизити швидкість передачі та підвищити енергоспоживання, що є критичним для бездротових сенсорних мереж. Отже, виклик полягає у створенні оптимальних стратегій надлишковості, які забезпечують надійність без втрати ефективності.

Варто зазначити, що перелічені проблеми взаємопов’язані. Гетерогенність підвищує складність координації, динамічність мережі ускладнює підтримку стабільності, децентралізація потребує нових протоколів узгодження рішень, а інформаційна надмірність впливає на пропускну здатність та енергоефективність системи. Тому підходи до підвищення відмовостійкості мають бути комплексними, охоплювати всі рівні IoT-інфраструктури – від сенсорів і шлюзів до хмарних платформ.

Отже, можна зазначити, що забезпечення відмовостійкості в системах Інтернету речей є багатогранною проблемою, яка охоплює технічні, організаційні та алгоритмічні аспекти. Для її вирішення необхідно розробляти адаптивні, розподілені та масштабовані механізми, здатні працювати в умовах гетерогенного середовища, великої кількості пристроїв і постійних змін у мережі. Саме комплексне врахування цих викликів дозволить створити надійну та стійку інфраструктуру IoT, здатну функціонувати без збоїв навіть у непередбачуваних умовах.

3.2. Концепція системи забезпечення відмовостійкості та надійності інфраструктури Інтернету речей

Враховуючи основні виклики, що постають під час розроблення ефективних підходів до підвищення надійності та стійкості систем Інтернету речей, доцільно сформувати узагальнену концепцію, яка може бути покладена в основу побудови системи забезпечення відмовостійкості в інфраструктурі IoT. Ця концепція має передбачати багаторівневу організацію обміну інформацією, взаємодію між шарами мережі, узгодженість механізмів реагування на збої та адаптивність до змін середовища [15, 16].

Одним із ключових елементів такої системи є багаторівнева комунікаційна структура. Інфраструктура Інтернету речей зазвичай має модульний характер і складається з кількох шарів: сенсорного, мережевого, обробного та прикладного. Кожен шар функціонує за власними принципами, використовує специфічні технології та протоколи зв’язку, а також має власні механізми забезпечення надійності. Проте ефективна робота системи можлива лише тоді, коли ці шари не діють ізольовано, а обмінюються інформацією про свій стан, виявлені помилки та вжиті заходи для їх усунення. В іншому разі може виникати ефект «каскадних відмов», коли збій на одному рівні призводить до порушення роботи всієї системи.

Для запобігання такому ефекту пропонується створити поперечний канал зв’язку між шарами, який забезпечуватиме взаємодію та обмін службовими даними між різними рівнями IoT-архітектури. Такий канал дозволить кожному рівню системи не лише фіксувати власні збої, але й отримувати контекстну інформацію про стан суміжних шарів. Це дасть змогу своєчасно виявляти потенційні загрози та запобігати поширенню помилок. Кожен рівень системи має бути оснащений спеціальним агентом системи забезпечення відмовостійкості, який відповідатиме за обмін інформацією з іншими агентами, координацію дій і прийняття рішень на локальному рівні. Таким чином, відмовостійкість досягається не лише за рахунок резервування, але й через інтелектуальну взаємодію між елементами системи.

Другим важливим компонентом концепції є уніфікований механізм взаємодії між різнорідними пристроями та протоколами зв’язку [16, 17]. Через високу гетерогенність IoT-систем у них часто відсутній єдиний стандарт комунікації, що ускладнює обмін даними між різними модулями. Для усунення цієї проблеми пропонується створити уніфіковану модель комунікації, орієнтовану на події, пов’язані з відмовами. Ця модель має бути технологічно нейтральною та не залежати від конкретних апаратних чи програмних платформ. Вона повинна визначати загальні правила обміну даними між рівнями системи у разі виявлення збоїв, містити мінімальний обсяг службової інформації та підтримувати адаптивне розширення для роботи з різними протоколами. Завдяки цьому можна забезпечити ефективну взаємодію між компонентами навіть у складних гетерогенних середовищах, не збільшуючи при цьому витрати на обробку даних чи трафік.

Третім аспектом є використання архітектури, побудованої на подіях (event-driven architecture). Системи Інтернету речей функціонують у динамічному середовищі, яке постійно змінюється: з’являються нові пристрої, змінюються умови експлуатації, оновлюються протоколи передачі даних. У таких умовах механізми відмовостійкості мають бути не статичними, а гнучкими й адаптивними. Архітектура на основі подій

дозволяє досягти цього завдяки асинхронному обміну повідомленнями між елементами системи, що мінімізує залежність між ними. Це означає, що будь-який модуль може реагувати на певну подію – відмову, зміну параметрів чи сповіщення від іншого шару – без необхідності втручання в роботу всієї мережі [17].

Такий підхід не лише підвищує надійність, але й спрощує інтеграцію нових технологій у майбутньому, адже формат подій може бути легко адаптований під нові стандарти зв'язку чи типи пристроїв. Крім того, асинхронність забезпечує безперервність функціонування системи: навіть у разі збою окремого елемента інші частини продовжують працювати, що знижує ризик зупинки критичних процесів.

Наступним принципом концепції є спільне виявлення, ізоляція та відновлення після збоїв. Системи Інтернету речей складаються з численних взаємопов'язаних пристроїв, які обмінюються даними, утворюючи складну мережу залежностей. Виникнення збоїв у будь-якому з елементів може спричинити ланцюгову реакцію, що призведе до відмови інших компонентів. Тому механізми забезпечення надійності повинні передбачати кооперативний процес діагностики, коли кілька рівнів або пристроїв спільно беруть участь у виявленні причин збою, локалізують його та координують дії з відновлення [17].

Запропонована система забезпечення відмовостійкості передбачає багаторівневий підхід до обміну інформацією про стан системи, який дозволяє агентам на різних рівнях співпрацювати під час виявлення та усунення помилок. Такий підхід забезпечує децентралізоване прийняття рішень: кожен агент може самостійно реагувати на локальні збої, використовуючи контекстну інформацію та дані, отримані з інших шарів. Це підвищує швидкість реагування та зменшує ймовірність каскадного поширення помилок.

Крім того, інформація, що циркулює між рівнями, може використовуватися для побудови адаптивних стратегій відмовостійкості, які

враховують реальні умови функціонування системи, рівень навантаження, типи пристроїв і критичність процесів. Така гнучкість дозволяє не лише відновлювати роботу після збоїв, а й передбачати потенційні проблеми, реалізуючи превентивне управління.

У підсумку, запропонована концепція системи забезпечення відмовостійкості та надійності інфраструктури Інтернету речей ґрунтується на чотирьох основних засадах:

- багаторівнева структура з поперечними каналами зв’язку;
- уніфікована модель комунікації для гетерогенних систем;
- подієво-орієнтована архітектура, що підтримує адаптивність;
- кооперативне виявлення та усунення відмов.

Реалізація цієї концепції дозволить підвищити стабільність функціонування IoT-систем, знизити ризики каскадних збоїв, забезпечити оперативне реагування на непередбачувані ситуації та створити гнучку інфраструктуру, здатну розвиватися разом із технологічним середовищем.

3.3. Система забезпечення багаторівневої відмовостійкості та надійності інфраструктури Інтернету речей

Запропонована система забезпечення відмовостійкості в інфраструктурі Інтернету речей базується на ідеї створення багаторівневого комунікаційного каналу, який забезпечує обмін подіями, що стосуються відмов та порушень у роботі елементів системи. Основне призначення цієї системи полягає у підтримці постійної взаємодії між різними рівнями архітектури IoT, дозволяючи їм оперативно реагувати на помилки та координувати свої дії без втручання в основні функціональні процеси.

Система діє як посередник між різними шарами IoT-інфраструктури, сприяючи ефективній комунікації між ними та підвищуючи узгодженість стратегій відмовостійкості. Вона реалізує обмін інформацією між локальними або розподіленими механізмами забезпечення надійності,

допомагаючи запобігати відмовам, які можуть призвести до збоїв у роботі всієї системи. Завдяки цьому створюється більш стійке середовище функціонування, у якому помилки виявляються на ранніх етапах і усуваються до того, як зможуть поширитися на інші елементи.

Процес забезпечення відмовостійкості у запропонованій системі відбувається у чотири основні фази, кожна з яких має своє призначення та впливає на надійність мережі в цілому:

1) виявлення помилок. На цьому етапі здійснюється діагностика стану системи для виявлення будь-яких відхилень, збоїв або аномалій, які можуть свідчити про потенційну відмову. Для цього використовуються різні методи моніторингу, перевірка цілісності даних, аналіз динаміки параметрів роботи пристроїв або виявлення нетипових запитів у мережі;

2) локалізація та оцінювання наслідків. Після виявлення відмови система проводить оцінку її впливу на загальний стан мережі. На цьому етапі визначається ступінь критичності проблеми, ідентифікуються компоненти, які постраждали, та оцінюється масштаб можливих наслідків. Такий підхід дозволяє обмежити поширення помилок і своєчасно вжити заходів для мінімізації шкоди;

3) відновлення працездатності. Далі система переходить до етапу відновлення роботи. Тут можуть застосовуватися різні техніки: автоматичне усунення збоїв, відновлення з резервних копій, перерозподіл навантаження або перенаправлення трафіку. Усе це спрямоване на якнайшвидше повернення системи до стабільного стану без повної зупинки її роботи;

4) оброблення збоїв та забезпечення безперервного функціонування системи. Після усунення проблеми система переходить до нормального режиму функціонування. На цьому етапі здійснюється перевірка стану компонентів, можлива їх переконфігурація або повторний запуск, а також здійснюється контроль за ефективністю відновлення. Основна мета полягає у забезпеченні безперервності функціонування IoT-інфраструктури навіть у разі часткових відмов.

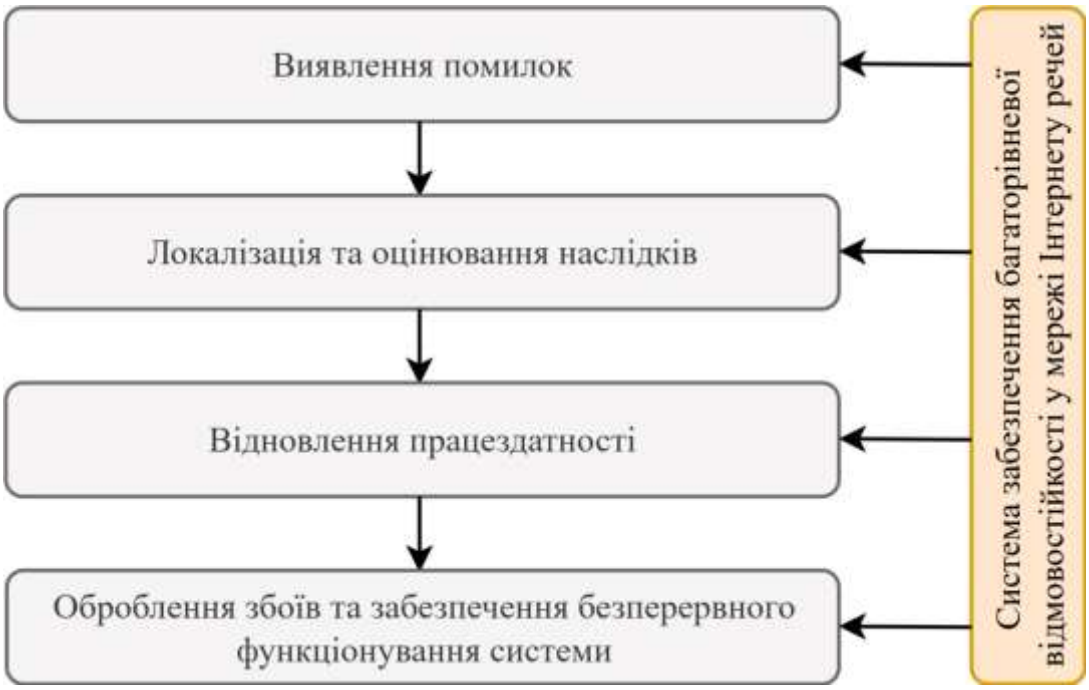


Рис. 3.1. Процес функціонування СЗВ

Ефективність роботи будь-якої стратегії відмовостійкості значною мірою залежить від точності й повноти виявлення помилок. Чим раніше система ідентифікує проблему, тим менше шкоди вона може завдати, і тим швидше можливо відновити роботу. Саме тому запропонована СЗВ приділяє особливу увагу моніторингу та аналізу стану системи у реальному часі, а також обміну інформацією між рівнями для координації дій під час відновлення.

Після усунення збою важливо не лише відновити працездатність системи, але й повернути її до стабільного стану. У цьому процесі СЗВ виконує роль координатора між різними рівнями – сенсорним, комунікаційним і прикладним, – забезпечуючи обмін технічними повідомленнями та прийняття узгоджених рішень щодо подальших дій.

Варто наголосити, що СЗВ не є окремим механізмом усунення відмов, а радше виступає в ролі інтелектуального комунікаційного середовища, яке об’єднує всі рівні інфраструктури IoT, забезпечує обмін подіями, підтримує прозорість у передачі даних про стан системи та сприяє прийняттю зважених рішень для підвищення надійності. Система спроектована таким чином, щоб

бути технологічно незалежною – це означає, що її можна інтегрувати з різними протоколами зв’язку, платформами чи пристроями без необхідності суттєвих змін у базовій архітектурі.

3.4. Архітектура системи забезпечення відмовостійкості в інфраструктурі Інтернету речей

Архітектура запропонованої системи складається з п’яти основних компонентів, які взаємодіють між собою за допомогою асинхронних повідомлень, що циркулюють через спеціальну шину подій [18]:

- канал введення подій – відповідає за приймання зовнішніх подій, що надходять із системи Інтернету речей;
- модуль обробки подій – здійснює аналіз отриманих даних, фільтрацію та підготовку інформації для подальшої передачі;
- модуль зберігання подій – забезпечує тимчасове або постійне збереження подій, необхідних для діагностики, аудиту чи подальшого аналізу;
- канал виведення подій – відповідає за передачу оброблених подій зовнішнім системам або наступним рівням інфраструктури;
- шина подій – виконує роль комунікаційного середовища між усіма компонентами архітектури, використовуючи модель публікації та підписки.

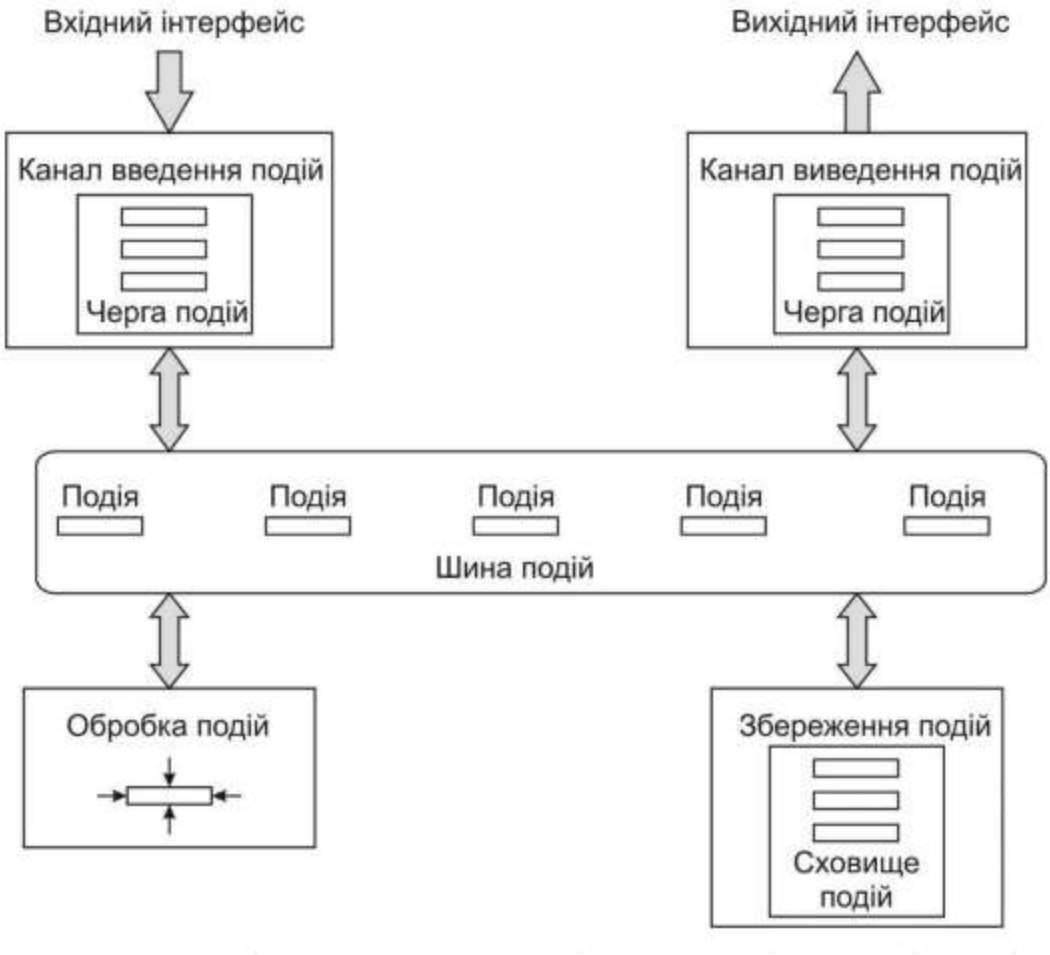


Рис. 3.2. Архітектурна модель системи забезпечення відмовостійкості в інфраструктурі Інтернету речей

Взаємодія між компонентами базується на патерні “повідомлення–підписка” (notification–subscription), що дозволяє кожному модулю підписуватися лише на ті події, які для нього є релевантними. Коли певна подія фіксується системою, вона автоматично надсилається всім зацікавленим компонентам, що забезпечує гнучкість та асинхронність обробки інформації.

Для підвищення масштабованості архітектура побудована за принципом безстанового обміну даними, що дозволяє мінімізувати залежність між компонентами. Усі події передаються системою лише тимчасово – вони не зберігаються постійно, якщо цього не вимагає конкретний протокол або політика безпеки.

Кожен модуль має точки розширення, які дозволяють додавати нові функціональні можливості, не змінюючи структури системи. Такий підхід забезпечує гнучкість і підтримку еволюції IoT-технологій, даючи змогу інтегрувати нові типи пристроїв, протоколів або форматів подій.

Компонент введення подій виступає своєрідним “брокером повідомлень”, забезпечуючи зовнішній інтерфейс зв’язку та підтримку різних протоколів – MQTT, CoAP, HTTP тощо. При цьому усі повідомлення формуються у стандартизованому форматі, який містить ключові поля:

- ідентифікатор – унікальний код події;
- напрямок передачі – визначає, чи подія рухається “вгору” або “вниз” по ієрархії;
- тип та підтип події – уточнюють її характер (контрольна, сповіщення, службова тощо);
- джерело та пункт призначення – ідентифікують вузол-відправник і вузол-отримувач;
- навантаження – містить службову або діагностичну інформацію про відмову чи стан системи.

Таким чином, запропонована архітектура дозволяє гнучко управляти інформаційними потоками, забезпечувати швидке виявлення та усунення збоїв, а також підтримувати стабільну роботу інфраструктури Інтернету речей навіть у динамічному та гетерогенному середовищі.

3.5. Принципи функціонування системи забезпечення багаторівневої відмовостійкості та надійності інфраструктури Інтернету речей

Розроблена система підвищення надійності та стійкості інфраструктури Інтернету речей ґрунтується на концепції віртуального міжрівневого каналу обміну даними, який забезпечує інтегровану взаємодію між усіма рівнями архітектури IoT. Цей канал створює єдиний простір комунікації, що дозволяє здійснювати координацію процесів виявлення, аналізу та усунення відмов у

різних сегментах мережі. У межах кожного рівня функціонують локальні механізми відмовостійкості (ЛМВ), які відповідають за моніторинг, діагностику та стабілізацію власного функціонального середовища.

Кожен рівень інфраструктури Інтернету речей характеризується власними типами ризиків і потенційних збоїв. Для сенсорного рівня це можуть бути апаратні несправності чи похибки вимірювань; для мережевого – затримки передачі, втрата пакетів або конфлікти протоколів; для прикладного – збої в алгоритмах обробки чи збереженні даних. Через цю гетерогенність неможливо створити єдиний універсальний механізм контролю, який би охоплював усі рівні одночасно. Тому запропонована СЗВ передбачає децентралізовану модель управління, у якій кожен рівень самостійно обробляє помилки, але водночас може обмінюватися контекстною інформацією з іншими рівнями.

Система забезпечення відмовостійкості виступає інтеграційним прошарком, який об’єднує локальні механізми в єдину узгоджену структуру. Вона не втручається безпосередньо у виконання прикладних задач, а виконує функції посередника між рівнями інфраструктури, забезпечуючи ефективний обмін діагностичними повідомленнями та сигналами відновлення. Завдяки цьому усувається потреба у централізованому управлінні, що мінімізує ризик виникнення єдиної точки відмови та підвищує загальну надійність системи.

Кожен локальний механізм відмовостійкості реєструється у структурі СЗВ та визначає, які типи подій або сигналів його цікавлять. Якщо в системі виникає подія (наприклад, збій сенсора чи перевищення затримки зв’язку), вона автоматично потрапляє до каналу комунікації, після чого передається тим компонентам, які здатні на неї реагувати. Це дозволяє здійснювати селективний обмін подіями, мінімізуючи перевантаження мережі службовими повідомленнями та забезпечуючи швидке реагування на критичні ситуації.

У межах запропонованого підходу створюється віртуальний інформаційний канал, який дозволяє різним рівням системи координувати дії

щодо усунення несправностей. Це забезпечує інтегроване реагування на збої, де дії окремих модулів узгоджуються через обмін подіями. Наприклад, виявлення помилки на сенсорному рівні може викликати автоматичну реакцію на мережевому рівні (перенаправлення даних через інший вузол) або на рівні прикладного програмного забезпечення (корекція обчислень).

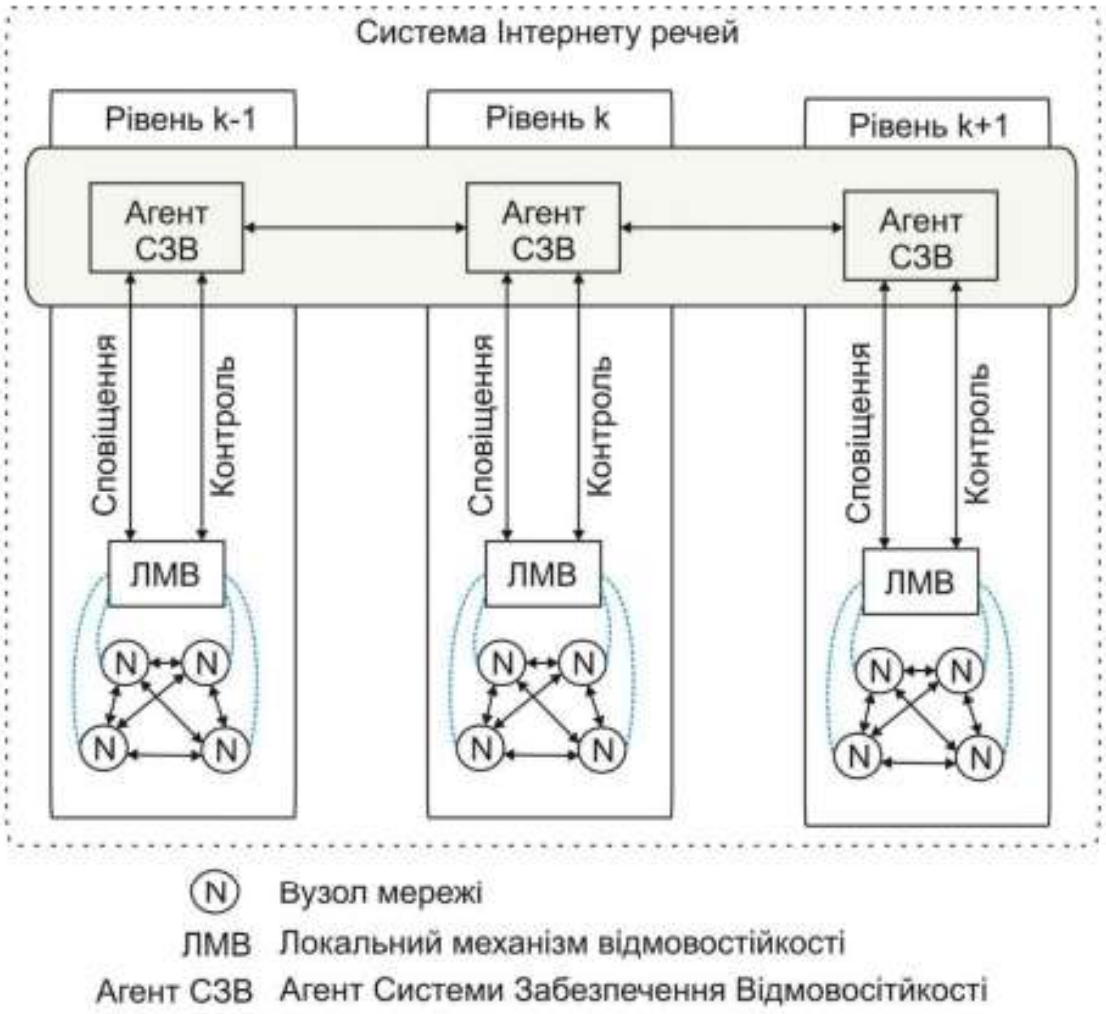


Рис. 3.3. Інформаційна взаємодія між рівнями інфраструктури в системі забезпечення відмовостійкості та надійності Інтернету речей

Наявність багаторівневої взаємодії створює певні труднощі – із ростом складності системи зростає кількість можливих точок збою. Чим більша кількість задіяних механізмів контролю, тим вищий ризик виникнення взаємних конфліктів або дублювання функцій. Для подолання цього СЗВ

реалізує принцип керованої складності: вона спрощує процес комунікації між рівнями, відокремлюючи обмін службовими подіями від основних функцій системи. Таким чином, локальні механізми можуть зосередитися на виконанні власних завдань – виявленні, локалізації та усуненні відмов – не витрачаючи ресурси на забезпечення комунікації.

Додатковою перевагою є те, що впровадження СЗВ не змінює існуючу структуру IoT-мережі. Всі обміни даними відбуваються у асинхронному та неблокуючому режимі, тобто система не створює затримок у роботі функціональних компонентів і не потребує централізованого контролю. Завдяки цьому її можна інтегрувати у вже діючі мережі без потреби їх реконфігурації.

Система підтримує два основних типи подій:

1) контрольні події – формуються механізмами відмовостійкості та стосуються діагностики, моніторингу чи виявлення збоїв. Вони використовуються для ініціації дій, спрямованих на стабілізацію системи;

2) інформаційні події (сповіщення) – передають аналітичні дані, адміністративні повідомлення або статистику роботи системи, необхідну для оцінки рівня її надійності.

Комунікаційний потік усередині СЗВ є двонаправленим: обмін даними здійснюється як зверху вниз (від прикладного рівня до сенсорного), так і знизу вгору (від фізичних пристроїв до керуючих систем). Такий підхід забезпечує зворотний зв’язок між компонентами, дозволяючи вищим рівням отримувати достовірну інформацію про стан інфраструктури, а нижчим – своєчасно реагувати на керуючі сигнали для усунення виявлених помилок.

Для підвищення надійності функціонування передбачено реплікацію агентів СЗВ. Кожен рівень має власний екземпляр агента, який відповідає за обробку локальних подій. У разі відмови одного агента його функції автоматично беруть на себе інші, забезпечуючи безперервність процесів діагностики та відновлення. Для синхронізації між агентами може

використовуватися розподілена база даних, що зберігає контекстні відомості про стан системи.

Комунікація між агентами та локальними механізмами відмовостійкості здійснюється за допомогою асинхронної моделі обміну повідомленнями, що гарантує стабільність системи навіть за умов непередбачуваних навантажень або часткових збоїв у мережі. Для участі в обміні подіями кожен ЛМВ реєструється у каналах введення та виведення подій, отримуючи право публікувати власні повідомлення або підписуватися на потрібні типи подій.

Таким чином, функціонування СЗВ забезпечує інтелектуальну взаємодію між рівнями інфраструктури IoT, що дозволяє своєчасно виявляти відмови, ізолювати їх наслідки, відновлювати роботу системи та запобігати каскадному поширенню помилок. Така модель підвищує загальну стійкість мережі, забезпечує адаптивність до змін і створює умови для реалізації самоорганізованих, самовідновлюваних IoT-систем, які здатні підтримувати надійну роботу навіть у складних і динамічних умовах.

3.6. Приклад реалізації системи забезпечення відмовостійкості та надійності інфраструктури Інтернету речей

Для демонстрації практичної реалізації розробленого методу забезпечення відмовостійкості та надійності інфраструктури Інтернету речей було спроєктовано архітектуру, що складається з кількох функціональних рівнів. На нижньому рівні розташовані фізичні пристрої – сенсори, актуатори, контролери та допоміжні елементи керування, які безпосередньо взаємодіють із навколишнім середовищем. Ці пристрої утворюють базовий шар IoT-інфраструктури, який генерує первинні дані, необхідні для подальшої аналітики.

У межах даного прикладу розглянуто два інтелектуальні прилади, кожен з яких підключений до набору різнотипних пристроїв – температурних

сенсорів, ламп, камер спостереження, моніторів та точок доступу. Основна функція приладів полягає у зборі показників, контролі працездатності підключених елементів і передачі інформації на верхні рівні системи для подальшої обробки.

Комунікаційний рівень відповідає за передачу зібраних даних та синхронізацію між пристроями й хмарною інфраструктурою. У межах моделі використано два шлюзи – по одному для кожного з протоколів зв’язку, якими користуються прилади. Цей рівень також виконує функцію агрегування даних і первинного моніторингу стану мережі, що дозволяє своєчасно виявляти розриви або перевантаження каналів.

Сервісний рівень виконує роль проміжного програмного шару (middleware), який забезпечує обробку, узагальнення та зберігання історичних даних. Він також відповідає за швидкий доступ до актуальної інформації, координацію запитів від верхніх рівнів і взаємодію з локальними механізмами відмовостійкості. У цьому шарі реалізовано аналітичні інструменти, які дозволяють проводити порівняльний аналіз станів пристроїв, прогнозувати можливі збої та оптимізувати навантаження на мережу.

На прикладному рівні розташовані користувацькі застосунки – системи моніторингу, керування пристроями та інтерфейси візуалізації даних. Через цей рівень оператор отримує доступ до системи, може спостерігати за станом компонентів, отримувати сповіщення про несправності та взаємодіяти з системою в режимі реального часу.

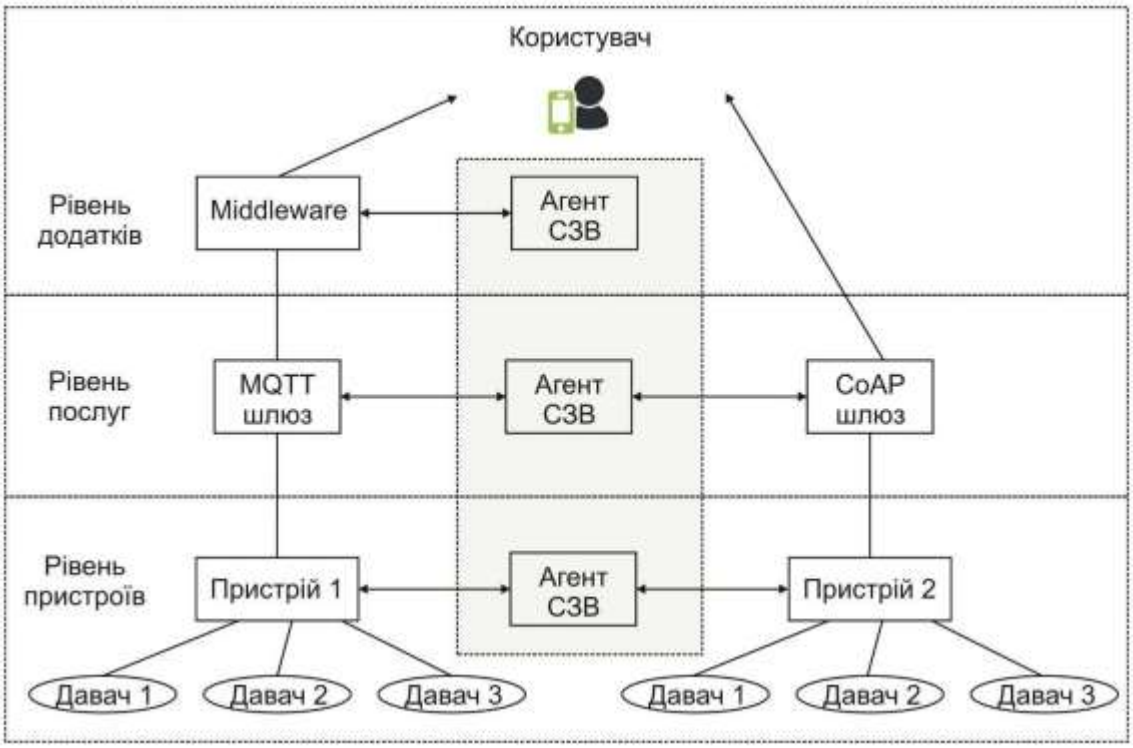


Рис. 3.4. Приклад побудови інфраструктури Інтернету речей з інтегрованою системою забезпечення відмовостійкості та надійності

На рисунку 3.5 відображено послідовність обміну подіями у процесі виявлення та усунення помилок у межах функціонування запропонованої системи. Якщо один із пристроїв фіксує несправність (наприклад, відмову сенсора або розрив зв’язку), він формує сигнал події відмови та надсилає його найближчому агенту СЗВ. Цей агент виконує роль локального координатора – він аналізує отриману інформацію, визначає тип і джерело помилки, а потім поширює повідомлення серед компонентів, які підписані на відповідний тип подій.

Паралельно з надсиланням події пристрій може самостійно застосувати базові механізми локального відновлення, наприклад, переключення на резервний сенсор або повторну ініціалізацію підключення. Такий підхід дозволяє мінімізувати ризик каскадного поширення збоїв і забезпечує оперативне відновлення працездатності системи.

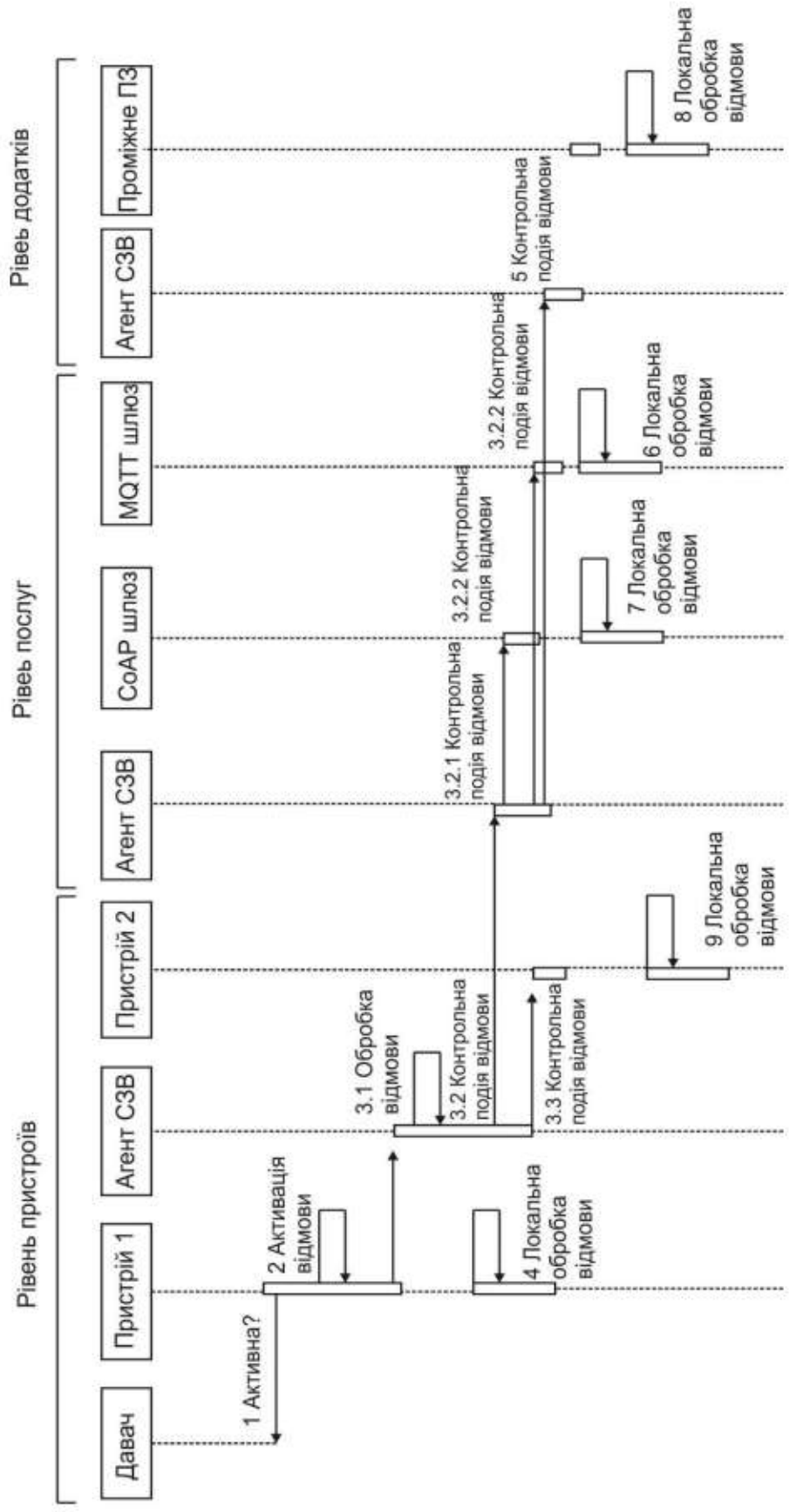


Рис. 3.5. Послідовність обміну подіями при виявленні та обробці відмов у системі забезпечення надійності Інтернету речей

Отримавши подію, агент СЗВ зберігає її у журналі подій (логах) для подальшого аналізу та формування карти відмов. На основі збережених записів система може проводити оцінку частоти виникнення збоїв, визначати критичні вузли інфраструктури та прогнозувати потенційні ризики.

Далі повідомлення передається до наступних рівнів СЗВ, зокрема до комунікаційного агента, який координує передачу даних між шлюзами. Цей агент здійснює асинхронну маршрутизацію подій, уникаючи перевантаження каналів зв’язку. Кожен шлюз при цьому застосовує власні алгоритми запобігання відмовам, наприклад, автоматичне дублювання пакетів або перехід на альтернативний маршрут передачі.

Подальша обробка здійснюється на сервісному рівні, де агент СЗВ аналізує контекст події, співвідносить її з даними з інших рівнів і передає результати у локальні механізми відмовостійкості, вбудовані у проміжне програмне забезпечення. Це дозволяє автоматично синхронізувати дії системи, наприклад, активувати резервні ресурси або адаптувати параметри роботи інших компонентів.

Таким чином, подія проходить через усі рівні інфраструктури, забезпечуючи узгоджений обмін інформацією та дозволяючи системі реагувати на збої в комплексний спосіб. Запропонований підхід дає змогу зменшити кількість одноточкових відмов, скоротити час реакції на несправності та підвищити загальний рівень надійності мережі.

Запропонована архітектура демонструє, що інтеграція системи забезпечення відмовостійкості у багаторівневу інфраструктуру IoT можлива без значних змін у базовій структурі мережі. Вона є гнучкою, розширюваною та здатною адаптуватися до різних протоколів і середовищ. У результаті система створює стійке середовище функціонування, в якому інформаційний обмін між рівнями сприяє підвищенню надійності, а механізми самовідновлення – збереженню безперервності сервісів навіть за умов часткових відмов.

3.7. Висновки до розділу 3

У розділі проаналізовано основні проблеми забезпечення багаторівневої відмовостійкості та надійності IoT-інфраструктури, зокрема гетерогенність компонентів, динамічність системи та необхідність децентралізованого прийняття рішень. Було запропоновано архітектуру системи відмовостійкості, що базується на багаторівневому каналі обміну подіями та дозволяє координацію різних стратегій без порушення роботи основної системи. Система спроектована на основі безстанового протоколу та включає п’ять основних модулів: канал введення подій, модуль обробки подій, сховище подій, канал виведення подій та шину подій. Застосування шини подій у поєднанні з вибірковим обміном подіями забезпечує зниження навантаження на мережу, підвищує адаптивність і масштабованість системи, а також гарантує безперервність функціонування IoT-інфраструктури. Запропонована архітектура формує надійну платформу для створення стійких, ефективних та гнучких IoT-мереж у різних сферах, включаючи промисловість, енергетику, транспорт та інші галузі.

РОЗДІЛ 4
ОЦІНКА ЕФЕКТИВНОСТІ МЕТОДІВ ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОСТІ ТА НАДІЙНОСТІ ІНФРАСТРУКТУРИ ІНТЕРНЕТУ РЕЧЕЙ

4.1. Експериментальні дослідження та характеристики оцінювання

Для перевірки ефективності запропонованих методів забезпечення відмовостійкості й надійності інфраструктури Інтернету речей було проведено серію експериментальних досліджень. Метою випробувань стало визначення переваг використання системи забезпечення відмовостійкості у порівнянні зі звичайною IoT-інфраструктурою без додаткових механізмів захисту від збоїв.

Для цього було змодельовано тестове середовище, яке реалізує концепцію «розумного дому». Такий тип системи є типовим прикладом практичного застосування Інтернету речей, оскільки включає різноманітні пристрої, датчики та комунікаційні модулі, що активно взаємодіють між собою. На основі цієї моделі було проведено порівняння двох варіантів системи:

1 варіант: без використання системи забезпечення відмовостійкості;

2 варіант: із впровадженою СЗВ, що реалізує багаторівневий обмін подіями та децентралізоване виявлення й усунення збоїв.

Запропонована система СЗВ враховує основні характеристики, притаманні більшості IoT-мереж:

- модульність і розподіленість компонентів;
- гетерогенність пристроїв та протоколів;
- обмеження на обчислювальні ресурси;
- необхідність безперервного функціонування навіть у разі виникнення часткових відмов.

Під час експерименту було визначено такі ключові цілі:

- підвищення ефективності виявлення та оброблення помилок;
- забезпечення безперервності обслуговування під час відмов;
- збільшення загальної доступності та надійності системи.

Для кількісного аналізу ефективності СЗВ було обрано набір метрик, що наведені в таблиці 4.1.

Таблиця 4.1

Метрики та показники ефективності системи забезпечення відмовостійкості

Метрика	Характеристика	Опис
Виявлення помилок	Час реагування на збій	Інтервал між виникненням помилки та її фіксацією системою
Продовження обслуговування	Період відновлення	Час, необхідний для усунення збою та повернення системи до стабільної роботи
Доступність	Активний час роботи	Частка часу, протягом якого система функціонує коректно
Надійність	Успішні транзакції	Кількість успішно виконаних запитів щодо загальної кількості звернень
Виявлені помилки	Кількість локалізованих збоїв	Число виявлених помилок, які не призвели до критичних відмов системи

4.2. Організація та сценарії проведення експерименту

Для реалізації експериментальних досліджень було створено тестову IoT-мережу, що включала елементи «розумного дому», такі як сенсори температури, вологості, тиску, розумні розетки, лампи, аудіосистеми та камери відеоспостереження.

Для порівняльного аналізу створено дві однакові конфігурації мережі:

1) система без СЗВ, що працює без механізмів виявлення та усунення відмов;

2) система з інтегрованою СЗВ, у якій додано агентів системи забезпечення відмовостійкості на рівнях сприйняття, комунікації та сервісів.

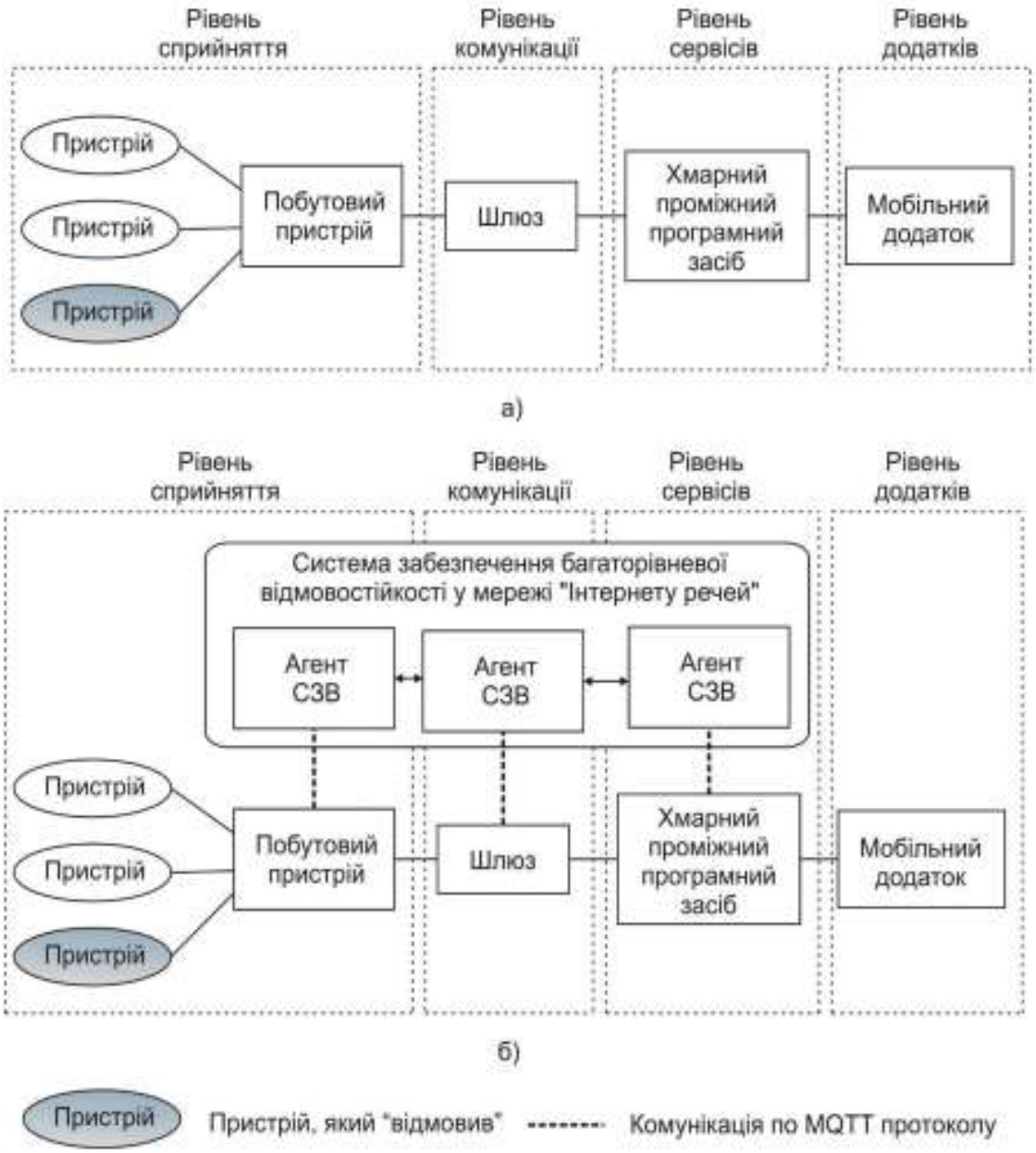


Рис. 4.1. Архітектура експериментальної системи Інтернету речей:
а) базова модель без СЗВ; б) система з реалізованою системою забезпечення відмовостійкості

Архітектура тестового середовища побудована за чотиришаровою структурою:

- рівень сприйняття (Perception Layer) – включає сенсорні пристрої, що збирають дані з навколишнього середовища (температура, вологість, освітлення тощо). Цей рівень безпосередньо взаємодіє з фізичними об’єктами;

- комунікаційний рівень (Network Layer) – забезпечує передачу даних між пристроями, агрегує інформацію через шлюзи, що виступають посередниками між нижчими та вищими рівнями;
- сервісний рівень (Service Layer) – реалізує обробку, зберігання та аналітику даних за допомогою хмарного проміжного програмного засобу, який також містить механізми відмовостійкості;
- рівень застосунків (Application Layer) – надає користувачеві змогу взаємодіяти з системою через мобільний додаток, отримувати дані та керувати пристроями в режимі реального часу.

У моделі з упровадженою системою забезпечення відмовостійкості агенти СЗВ встановлені на рівнях сприйняття, комунікації та сервісів. Вони з’єднані між собою каналом обміну подіями, що дозволяє координувати реакцію системи на помилки, ізолювати несправності та зменшувати ризик каскадного поширення відмов.

Обидві системи – базова та вдосконалена – мають однакову кількість пристроїв, вузлів і сервісів, що забезпечує коректність порівняння результатів експериментів.

Кожен компонент моделі функціонував як окремий вузол, що дозволило точно вимірювати вплив запроваджених механізмів на час виявлення, усунення та відновлення системи після відмов.

Результати експериментів показали, що наявність СЗВ значно підвищує стійкість мережі до збоїв, скорочує середній час простою та підвищує загальний рівень доступності сервісів, особливо у випадках непередбачуваних відмов сенсорів або втрати зв’язку між вузлами.

4.3. Сценарії експериментальних досліджень і перевірка ефективності системи забезпечення відмовостійкості

Для оцінювання ефективності запропонованої системи забезпечення відмовостійкості було змодельовано два експериментальні сценарії, які

відрізнялися механізмом виявлення та локалізації відмов у мережі Інтернету речей. У межах кожного сценарію проводилося тестування двох варіантів роботи системи – із використанням багаторівневої СЗВ та без неї. Основні параметри досліджуваних сценаріїв наведено в таблиці 4.2.

Таблиця 4.2.

Сценарії, використані для експериментальної перевірки ефективності СЗВ

Сценарій	Механізм відмовостійкості	Наявність системи забезпечення багаторівневої відмовостійкості
Сценарій 1	Обмеження поширення збоїв	Без СЗВ
Сценарій 1	Обмеження поширення збоїв	Із СЗВ
Сценарій 2	Активний моніторинг стану пристроїв	Без СЗВ
Сценарій 2	Активний моніторинг стану пристроїв	Із СЗВ

Сценарій 1. Реактивне обмеження поширення відмов.

У першому сценарії досліджувалася поведінка системи в умовах локальної відмови одного з пристроїв. Логіка роботи базувалася на принципі реактивного обмеження поширення помилок.

Після того як хмарний проміжний програмний засіб отримував інформацію про недоступність певного ресурсу, усі нові запити, що надходили до несправного пристрою, автоматично блокувалися. Система одразу повідомляла, що пристрій недоступний, що дозволяло запобігти формуванню нових відмов, спричинених помилковими зверненнями до несправного елемента.

Паралельно активувався механізм повторних спроб підключення, який реалізовувався через експоненційно зростаючу затримку між перевірками стану пристрою. Цей процес тривав до досягнення заданого порогового значення часу.

Як тільки система фіксувала, що пристрій знову доступний, перевірки припинялися, а звичайний доступ до ресурсу відновлювався. Така стратегія дала змогу уникнути каскадного поширення відмов і зменшити ризик

порушення стабільності всієї мережі.

Загальний алгоритм функціонування сценарію 1 для тестування системи забезпечення відмовостійкості зображено на рисунку 4.2.

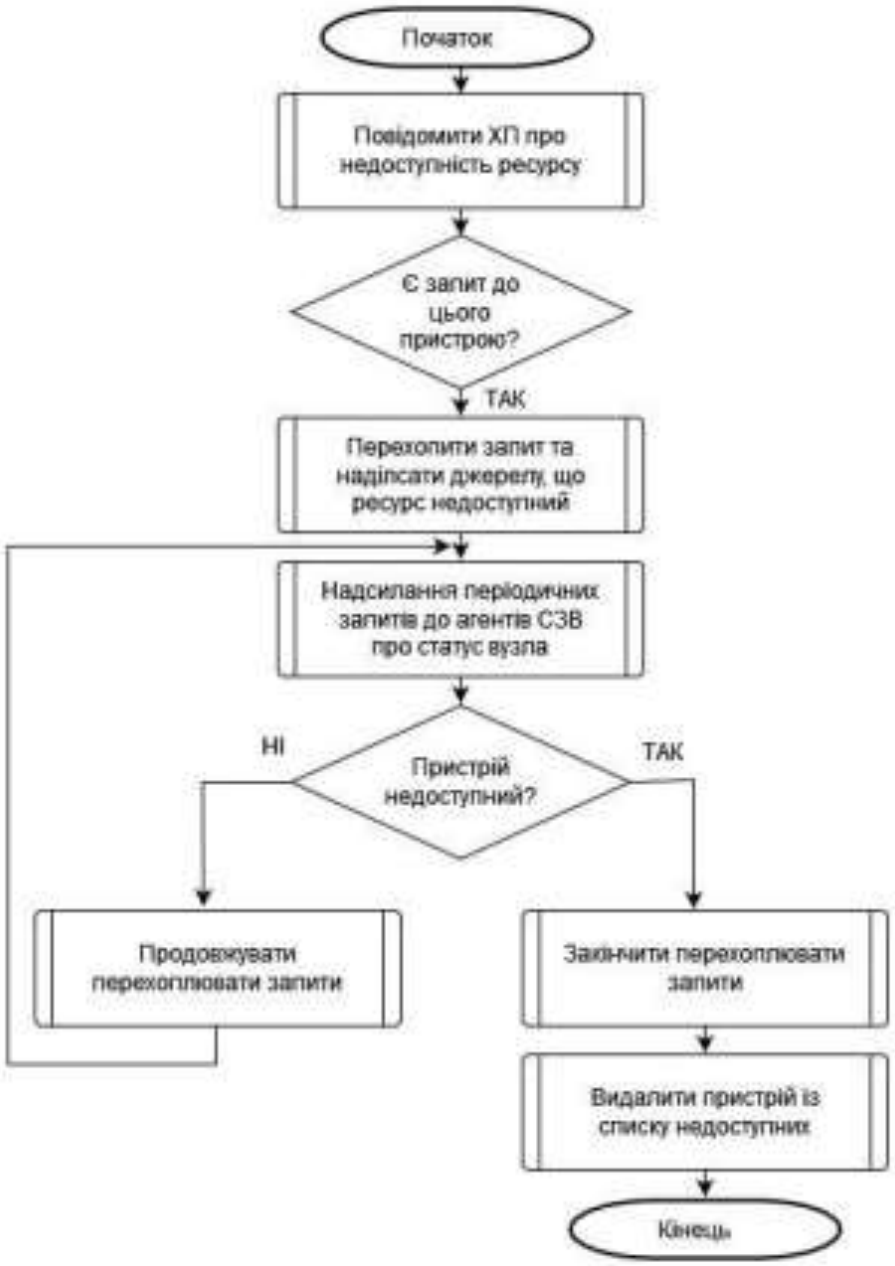


Рис. 4.2. Узагальнена схема функціонування першого сценарію тестування системи забезпечення відмовостійкості

Сценарій 2. Активний моніторинг і раннє виявлення збоїв.

Другий сценарій передбачав використання проактивного підходу до підвищення надійності – тобто постійного контролю стану всіх компонентів

мережі. На відміну від першого випадку, де помилка виявлялася лише після запиту до несправного пристрою, у цьому сценарії реалізовано безперервний моніторинг працездатності елементів системи.

Механізм виявлення відмов був заснований на технології типу “gossip” (чуткова перевірка), коли кожен рівень IoT-системи періодично надсилає сигнали активності – так звані “пульсації” (heartbeat messages) – для підтвердження власної працездатності.

Кожен вузол веде список інших пристроїв і регулярно перевіряє, чи надходили від них сигнали протягом встановленого інтервалу часу. Якщо часовий поріг перевищено й сигнал не отримано, пристрій позначається як несправний, а відповідне повідомлення розповсюджується всіма рівнями системи.

Після того як хмарний проміжний програмний засіб отримує запит до недоступного ресурсу, він перехоплює звернення, повідомляє про відмову та тимчасово блокує подальші запити до цього пристрою, запобігаючи накопиченню нових помилок. Як тільки ресурс знову переходить у активний стан, обмеження автоматично знімаються, і взаємодія відновлюється.

Алгоритм реалізації другого сценарію наведено на рисунку 4.3.

Розглянемо особливості використання СЗВ у сценаріях. У кожному з описаних сценаріїв було проведено два експерименти – з використанням запропонованої системи забезпечення відмовостійкості та без неї. Основна відмінність між ними полягає в способі оброблення подій відмови.

У випадку з впровадженою СЗВ процес реагування є координованим і розподіленим:

- при виявленні збою побутовий прилад ПП2 надсилає повідомлення до найближчого агента СЗВ на відповідному рівні;
- агент СЗВ передає інформацію далі через міжрівневий канал взаємодії до хмарного проміжного програмного засобу;
- хмарний проміжний програмний засіб активує механізм обмеження нових запитів до несправного пристрою, запобігаючи каскадним

відмовам;

- після відновлення працездатності пристрою агент СЗВ генерує подію про відновлення, яка передається вгору по рівнях, що дозволяє системі знову дозволити запити до відновленого ресурсу.

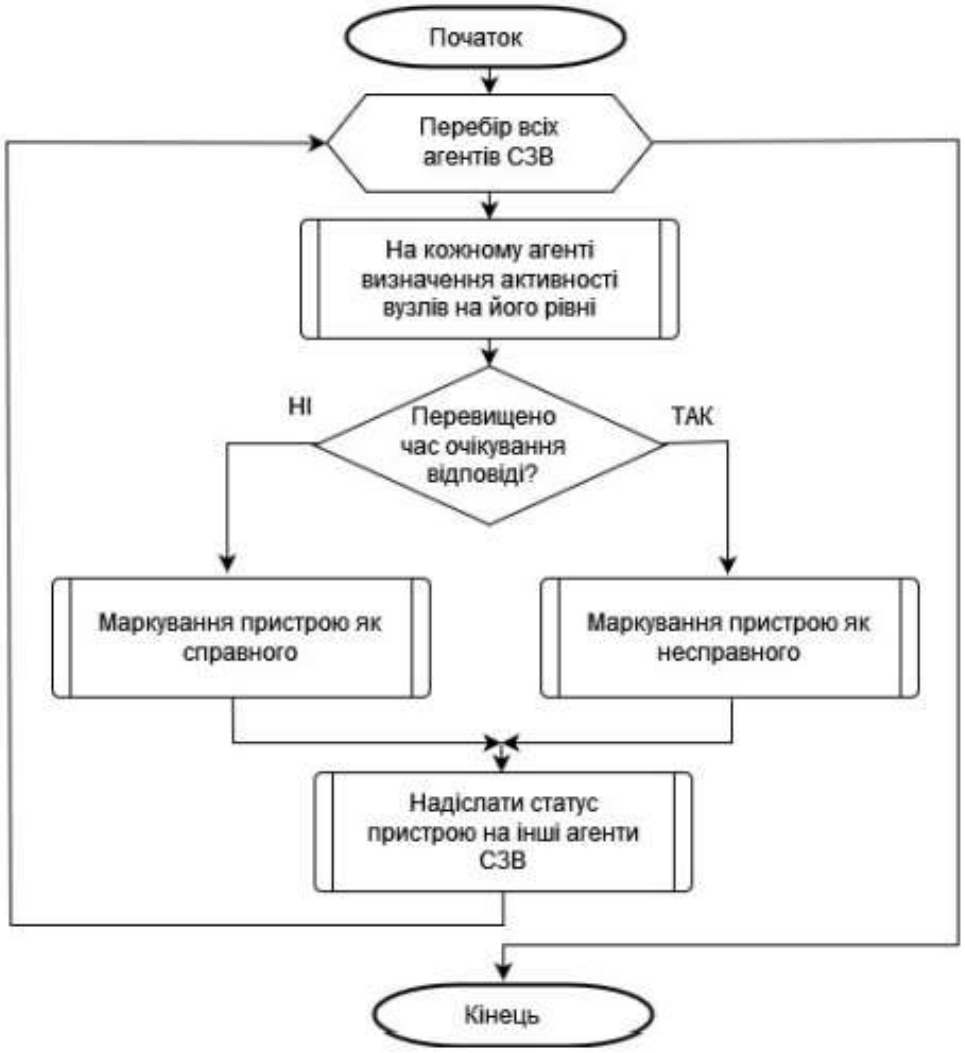


Рис. 4.3. Узагальнена схема функціонування другого сценарію тестування системи забезпечення відмовостійкості

Таким чином, інтеграція багаторівневої системи забезпечення відмовостійкості дала змогу створити адаптивний механізм самовідновлення, який реагує на збої в реальному часі, координує дії між різними рівнями IoT-інфраструктури та підвищує загальну стабільність функціонування мережі.

4.4. Реалізація експериментальних сценаріїв

Для практичної перевірки ефективності запропонованої СЗВ було проведено серію експериментів, що моделювали виникнення відмов у системі Інтернету речей.

Зокрема, в моделі «розумного дому» передбачалося штучне внесення двох помилок типу “пропуск”, які відбувалися у певні часові інтервали. Відмови генерувалися у пристрої – датчику температури, підключеному до приладу ПП2. Розклад та тривалість цих помилок наведено у таблиці 4.3.

Таблиця 4.3.

Розклад моделювання збоїв у системі (відключення температурного давача)

Початок виникнення помилки	Тривалість помилки
Помилка_1 – 5 с	10 с
Помилка_2 – 20 с	30 с

Симуляцію збоїв температурного сенсора (наприклад, DHT11) можна виконати кількома способами, залежно від завдань експерименту та обраної моделі поведінки системи. У межах цього дослідження було розглянуто три базові підходи до моделювання відмов:

1) штучне викривлення даних. Програмне спотворення показників сенсора, коли замість реальних значень пристрій передає некоректні – наприклад, надмірно високі або занижені температури, що виходять за межі допустимого діапазону;

2) моделювання тимчасового відключення сенсора. У цьому випадку пристрій програмно робиться «недоступним» для зчитування даних протягом визначеного часу, що імітує відмову або втрату зв’язку;

3) використання віртуального (імітаційного) датчика. Реальний сенсор замінюється програмним модулем, який випадковим чином генерує хибні або нестабільні значення температури.

У межах проведеного дослідження було обрано другий підхід –

моделювання відключення сенсора, як найбільш типовий сценарій для реальних IoT-систем.

Зображення використовуваного давача температури і вологості DHT11 подано на рисунку 4.4.

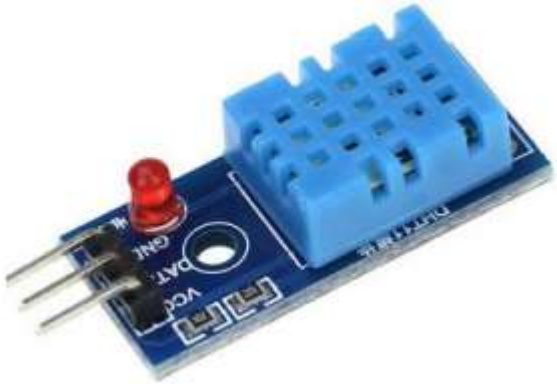


Рис. 4.4. Модуль давача температури та вологості DHT11

Для фіксації часу виявлення відмови використовувався момент, коли система фіксує неможливість доступу до сенсора після його відключення. Цей час визначається між подією виникнення збою та моментом отримання відповіді про недоступність ресурсу.

Тривалість відмови характеризує інтервал, протягом якого пристрій залишається непрацездатним. Обидва параметри виражалися в секундах.

У процесі експерименту було прийнято такі параметри:

- час очікування відповіді на запит – 200 мс;
- максимальна інтенсивність запитів – 20 запитів/с.

Перед основним етапом випробувань виконувалася ініціалізаційна фаза тривалістю близько 5 секунд, необхідна для стабілізації роботи системи.

Після цього мобільний застосунок починав надсилати запити до пристроїв протягом 30 секунд.

Кожен сценарій повторювався 30 разів, причому запити генерувалися випадковим чином, що забезпечило статистичну достовірність отриманих результатів.

У результаті було обчислено середні значення часу виявлення помилок, тривалості відновлення та рівня доступності для кожного сценарію.

Для відтворення реальної інфраструктури Інтернету речей створено імітаційну модель мережі IoT, у яку інтегровано багаторівневу систему забезпечення відмовостійкості.

Як фізичні вузли було використано одноплатні комп'ютери Raspberry Pi 4 Model B, з'єднані у локальну мережу зі швидкістю 100 Мбіт/с.

Кожен вузол відповідав окремому рівню IoT-системи:

- рівень сприйняття – збір даних із сенсорів;
- рівень шлюзу – передавання даних до хмарного середовища;
- рівень хмарного провайдера – оброблення та аналіз інформації.

Схематичну структуру інтеграції запропонованої системи у модель мережі Інтернету речей подано на рисунку 4.5.

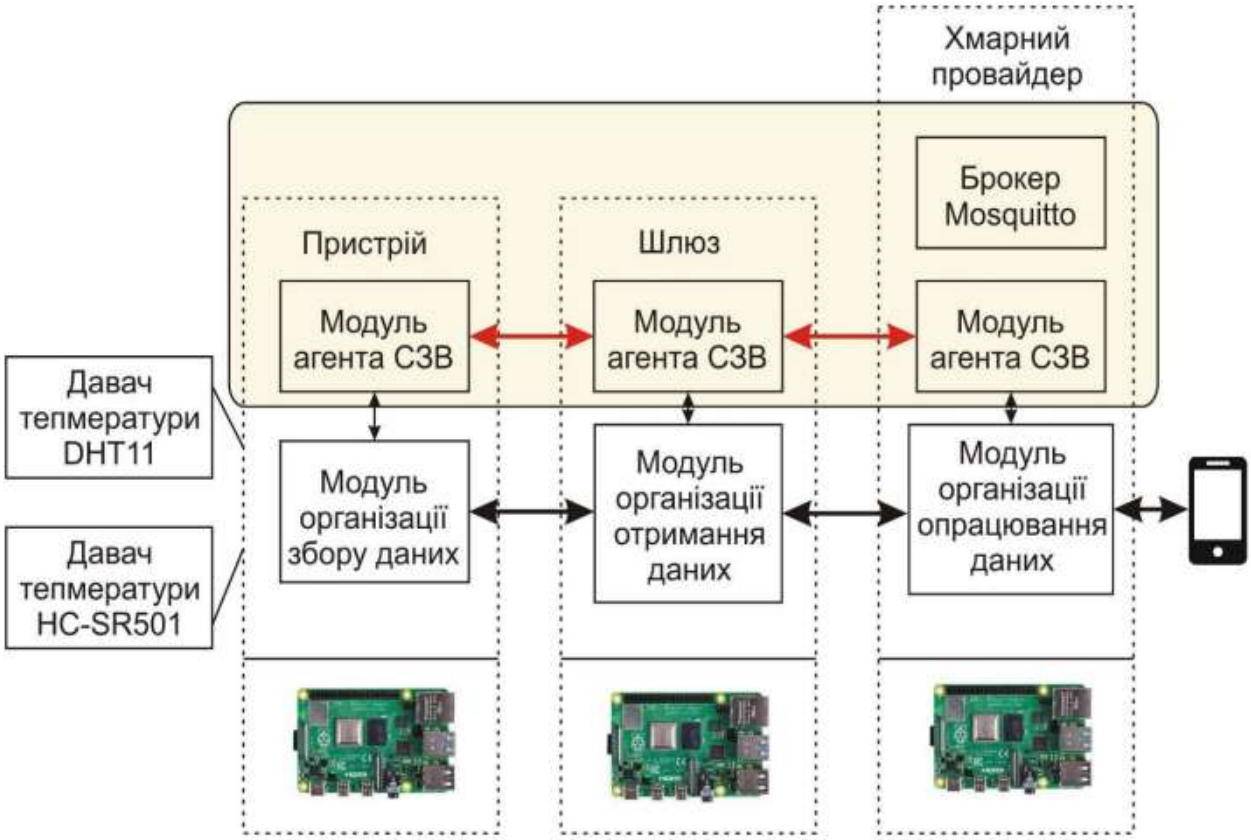


Рис. 4.5. Інтеграція розробленої системи забезпечення відмовостійкості в модель мережі Інтернету речей

Для виконання експерименту було створено спеціалізоване програмне забезпечення на мові Python.

Кожен вузол Raspberry Pi містив двокомпонентну структуру:

- модуль функціонування, який реалізовував базові операції відповідного рівня системи (збір, передавання чи обробка даних);
- модуль агента СЗВ, який забезпечував роботу механізмів виявлення, обміну й усунення відмов.

Модуль агента СЗВ базувався на протоколі MQTT, який дозволяє організувати надійну та легку комунікацію між вузлами [19, 20].

Псевдокод реалізації модуля зв'язку агента СЗВ наведено нижче.

```
import paho.mqtt.client as mqtt
import time

# Вказуємо адресу та порт MQTT-брокера
BROKER_ADDRESS = "192.168.1.100"
BROKER_PORT = 1883

# Функція, яка викликається при підключенні до брокера
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Успішне підключення до брокера MQTT")
        # Підписка на топік для отримання повідомлень
        client.subscribe("iot/events")
    else:
        print(f"Помилка підключення, код: {rc}")

# Функція обробки отриманих повідомлень
def on_message(client, userdata, message):
    print(f"Отримано повідомлення: {message.payload.decode()}")

# Створення клієнта MQTT для агента системи забезпечення відмовостійкості
client = mqtt.Client("IoT_Resilience_Agent")
```

```
# Прив’язка функцій обробки подій
client.on_connect = on_connect
client.on_message = on_message

# Підключення до брокера MQTT
client.connect(BROKER_ADDRESS, BROKER_PORT, keepalive=60)

# Запуск циклу обробки подій в окремому потоці
client.loop_start()

try:
    while True:
        # Надсилання тестового повідомлення на топік
        client.publish("iot/events", "Тестове повідомлення від агента СЗВ")
        time.sleep(2)  # інтервал між повідомленнями в секундах
except KeyboardInterrupt:
    # Завершення роботи та відключення від брокера
    print("Завершення роботи агента")
    client.loop_stop()
    client.disconnect()
```

На хмарному вузлі системи було розгорнуто брокер Mosquitto, який реалізував центральну шину повідомлень СЗВ.

Mosquitto – це легкий MQTT-брокер із відкритим вихідним кодом, що забезпечує ефективний, масштабований і безпечний обмін даними між компонентами системи.

Його використання зумовлено такими перевагами:

- відкритість і безкоштовна ліцензія;
- простота інтеграції з Python та іншими мовами програмування;
- підтримка TLS/SSL для захисту даних під час передавання.

Завдяки цьому було створено надійну комунікаційну платформу, яка забезпечила стабільність і стійкість обміну повідомленнями навіть при

виникненні часткових відмов.

Кінцеві користувачі взаємодіяли з системою «розумного дому» через мобільний додаток, який надсилав запити до хмарного програмного забезпечення та отримував відповіді від підключених сенсорів.

Модуль відмовостійкості було інтегровано саме на рівні хмарного програмного забезпечення – як найбільш потужного елемента системи з мінімальними обмеженнями за ресурсами та зв’язком.

4.5. Результати експериментальних досліджень

Для оцінювання ефективності впровадження системи забезпечення багаторівневої відмовостійкості у мережах Інтернету речей було проведено серію випробувань, спрямованих на визначення покращення показників виявлення та усунення помилок у порівнянні з традиційними підходами.

У межах кожного сценарію фіксувалися середні значення часу виявлення відмов та часу відновлення функціонування системи. Отримані результати узагальнено в таблицях 4.4 та 4.5.

У таблицях наведено порівняльні результати для кожного сценарію:

- позначка (+) відповідає використанню запропонованої системи багаторівневої відмовостійкості;
- позначка (–) – роботі системи без СЗВ.

Результати показали, що застосування СЗВ значно скорочує інтервал між виникненням відмови та її виявленням, незалежно від типу сценарію.

Найбільше значення часу виявлення (понад 500 мс) було зафіксовано для сценарію_1 без використання СЗВ.

Порівняльний аналіз показав, що:

- у сценарії_1 скорочення часу виявлення відмов становило близько 8 %;
- у сценарії_2 – до 53,4 %, що свідчить про підвищення ефективності системи при використанні агентів СЗВ.

Слід зазначити, що експериментальна модель містила обмежену кількість пристроїв, тому ймовірність звернення до несправного елемента була вищою. Це могло штучно скоротити час виявлення відмов у версіях без СЗВ.

У масштабних IoT-середовищах, де функціонують сотні чи тисячі сенсорів, частота запитів є значно нижчою, що, навпаки, погіршило б швидкість виявлення помилок у системах без відмовостійких механізмів.

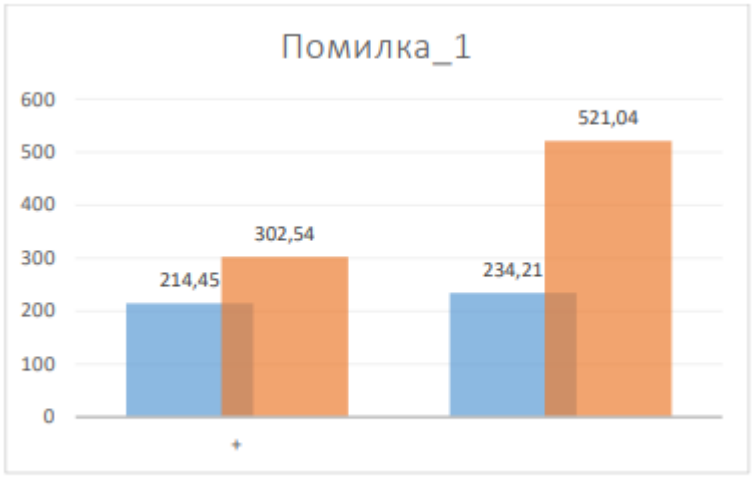
У сценарії_2 стратегія передбачала активне поширення повідомлень про збої між рівнями. Це створювало конкуренцію між потоками інформації про відмови та основними процесами оброблення даних, що впливало на час доставки повідомлень. Саме ця взаємодія пояснює незначні коливання часу виявлення між окремими виконаннями при використанні СЗВ.

Таблиця 4.4.

Середні значення часу виявлення помилок

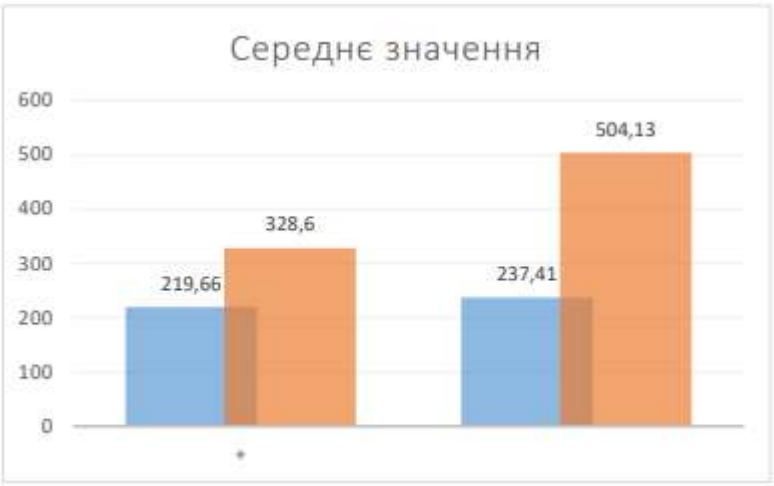
Сценарій	Застосування СЗВ	Помилка_1, мс	Помилка_2, мс	Середнє значення, мс
Сценарій_1	+	214,45	224,87	219,66
	–	234,21	240,61	237,41
Сценарій_2	+	302,54	354,66	328,60
	–	521,04	487,21	504,13

Результати графічно зображено на рисунку 4.6, де наведено час виявлення помилки для обох випадків: із СЗВ та без неї.



а)

б)



в)

Рис. 4.6. Результати експерименту вимірювання часу виявлення:

а) помилка_1; б) помилка_2; в) середнє значення

Додаткові експериментальні результати стосувалися часу усунення помилок (табл. 4.5).

Дані свідчать, що використання СЗВ забезпечує суттєве скорочення часу відновлення роботи системи після виникнення збоїв.

Таблиця 4.5.

Результати вимірювання часу усунення помилок

Сценарій	Застосування СЗВ	Помилка_1, мс	Помилка_2, мс	Середнє значення, мс
Сценарій_1	+	192,45	450,47	321,46
	–	512,23	2864,25	1688,24
Сценарій_2	+	201,10	212,17	206,64
	–	274,47	279,23	276,85

Порівняння отриманих даних засвідчило, що в сценарії_1 час усунення помилок у системі без СЗВ перевищував аналогічний показник у 5,25 раза.

Цю різницю пояснює експоненційна залежність між тривалістю відмови та затримкою відновлення – чим довше триває збій, тим більше часу потрібно системі для відновлення.

У сценарії_2 середній час вирішення проблеми скоротився на 33 % при використанні СЗВ.

Наявність агентів системи відмовостійкості дала змогу зменшити навантаження на мережу, адже некоректні запити до несправних пристроїв одразу блокувалися.

Завдяки цьому уникало дублювання запитів, знижувалося навантаження на обчислювальні ресурси й зменшувався загальний час реакції системи.

Отримані результати свідчать, що впровадження системи забезпечення багаторівневої відмовостійкості суттєво підвищує надійність і стійкість IoT-інфраструктури, забезпечуючи:

- скорочення часу виявлення відмов більш ніж на 50 % у складних сценаріях;
- зменшення середнього часу відновлення у 3–5 разів;

- стабільність функціонування навіть за наявності одночасних збоїв на кількох рівнях системи.

Таким чином, експериментально підтверджено, що запропонований метод організації відмовостійкої інфраструктури Інтернету речей є ефективним та перспективним для впровадження у розподілених середовищах реального часу.

4.6. Аналіз ефективності системи та узагальнення результатів

На основі проведених експериментальних досліджень було здійснено комплексний аналіз ефективності запропонованої системи забезпечення багаторівневої відмовостійкості в умовах функціонування розподіленої інфраструктури Інтернету речей. Основна увага приділялася оцінюванню трьох ключових параметрів: часу виявлення відмов, часу відновлення системи після збоїв та рівня загальної надійності та доступності інфраструктури.

Отримані результати свідчать, що впровадження СЗВ дало змогу суттєво підвищити показники безперервності функціонування IoT-системи. Зокрема, у сценаріях із використанням агентів відмовостійкості середній час виявлення несправностей скоротився в середньому на 30–50 %, а час відновлення – до п’яти разів порівняно з базовими реалізаціями без СЗВ. Це свідчить про ефективність механізмів координації між рівнями системи, а також про результативність використання багатоканальної шини обміну подіями.

Покращення виявлення та усунення помилок. Використання запропонованого методу дало можливість суттєво зменшити інтервал між моментом виникнення відмови та її виявленням. Такий ефект досягається завдяки:

- децентралізованій структурі СЗВ, що дозволяє незалежно фіксувати локальні збої на різних рівнях ієрархії системи;

- асинхронній обробці повідомлень, що усуває затримки, пов’язані з централізованим контролем;
- розподіленим агентам, які безпосередньо обмінюються інформацією про відмови без потреби в централізованому узгодженні.

Таке поєднання дозволяє значно підвищити швидкість реакції системи на потенційні загрози її стабільності та уникнути каскадних відмов, що є особливо критичним для великих IoT-середовищ із численними сенсорами, контролерами та хмарними сервісами.

Вплив СЗВ на загальну надійність і доступність системи. Аналіз показав, що використання багаторівневої системи забезпечення відмовостійкості позитивно впливає на коефіцієнт доступності системи – частку часу, коли вона функціонує без порушень.

Для змодельованого середовища «Розумного дому» середній коефіцієнт доступності під час тестування зріс з 0,92 до 0,98, що свідчить про наближення системи до режиму високої готовності (high-availability).

Це особливо важливо для інфраструктур, які мають на меті безперервну роботу в реальному часі, наприклад: розумні будівлі, системи моніторингу промислового обладнання, транспортні IoT-платформи тощо.

Зростання показників надійності обумовлене тим, що агенти СЗВ здійснюють динамічне перенаправлення інформаційних потоків та ізоляцію несправних вузлів, запобігаючи розповсюдженню відмов на інші частини системи. У результаті виявлені помилки не призводять до критичних збоїв або втрати зв’язку з усією інфраструктурою.

Порівнюючи результати сценаріїв, можна зробити висновок, що активний моніторинг, реалізований у сценарії_2, є більш ефективним при роботі в реальних умовах великої кількості пристроїв.

Хоча він потребує більших обчислювальних ресурсів, такий підхід забезпечує постійну перевірку стану системи і дозволяє завчасно реагувати на потенційні збої.

Водночас стратегія обмеження поширення помилок, що

використовувалася в сценарії_1, є ефективною для невеликих систем або локальних IoT-мереж, де важлива швидкість реакції на одиничні відмови, але контроль за всіма вузлами не є критичним.

Загалом, обидві стратегії продемонстрували покращення ефективності роботи системи при використанні СЗВ, проте саме багаторівневий механізм із активним моніторингом виявився найрезультативнішим для підвищення стабільності всієї інфраструктури.

Підсумовуючи результати проведених експериментів, можна зробити такі основні висновки:

- інтеграція системи забезпечення багаторівневої відмовостійкості підвищує швидкість виявлення помилок на 30–50 % та скорочує середній час відновлення роботи системи в 3–5 разів;
- запропонований підхід забезпечує високу гнучкість і масштабованість, оскільки дозволяє адаптувати рівні взаємодії відповідно до складності інфраструктури IoT;
- використання асинхронної комунікаційної моделі та подієвої архітектури (event-driven architecture) забезпечує зменшення затримок у передачі повідомлень між рівнями системи;
- ефективність СЗВ доведено в умовах зростання кількості пристроїв, що свідчить про її потенціал для впровадження у масштабних IoT-екосистемах.
- підвищення показників надійності та доступності підтверджує, що система є здатною забезпечити стабільне функціонування інфраструктури Інтернету речей навіть у разі локальних відмов.

Таким чином, результати дослідження демонструють, що впровадження розробленої системи забезпечення відмовостійкості є перспективним напрямом для створення надійних, самовідновлюваних та масштабованих IoT-систем, які можуть стабільно працювати у розподіленому середовищі з великою кількістю гетерогенних пристроїв.

4.7. Висновки до розділу 4

У розділі проведено експериментальну перевірку ефективності запропонованої системи забезпечення багаторівневої відмовостійкості та надійності інфраструктури Інтернету речей.

На основі створеної моделі «розумного дому» було реалізовано два сценарії роботи системи – реактивне обмеження поширення відмов і активний моніторинг стану пристроїв. Для кожного сценарію проведено порівняння двох варіантів: із впровадженою СЗВ та без неї.

Результати експериментів показали, що застосування СЗВ дозволяє:

- скоротити середній час виявлення відмов на 30–50 %;
- зменшити середній час відновлення системи після збоїв у 3–5 разів;
- підвищити коефіцієнт доступності системи з 0,92 до 0,98;
- забезпечити стабільне функціонування навіть за наявності одночасних локальних збоїв.

Отримані результати пояснюються децентралізованим характером СЗВ, що забезпечує незалежне виявлення та локалізацію помилок на кожному рівні IoT-інфраструктури, а також використанням подієвої архітектури та асинхронної обробки повідомлень, які мінімізують затримки передачі даних.

Порівняльний аналіз сценаріїв показав, що активний моніторинг є більш ефективним для великих і розподілених IoT-систем, тоді як реактивний підхід є доцільним для невеликих локальних мереж.

Загалом, проведені експерименти підтвердили, що впровадження багаторівневої системи забезпечення відмовостійкості:

- підвищує загальну надійність і доступність інфраструктури Інтернету речей;
- знижує ризик каскадних відмов;
- забезпечує адаптивність до змін середовища та розширюваність системи.

Отже, розроблена система демонструє високу ефективність у підвищенні відмовостійкості та може бути рекомендована для застосування в масштабних розподілених IoT-екосистемах промислового, енергетичного та транспортного призначення.

ВИСНОВКИ

В кваліфікаційній роботі було здійснено всебічний аналіз теоретичних основ, методів і практичних рішень, спрямованих на підвищення стабільності та безперервності функціонування систем Інтернету речей. Розглянуто сучасні підходи до побудови IoT-інфраструктури, її архітектуру, компоненти, протоколи взаємодії та чинники, що впливають на надійність і відмовостійкість систем цього типу.

У ході дослідження було визначено, що головною проблемою сучасних IoT-систем є складність забезпечення безперервності роботи в умовах децентралізованої архітектури та обмежених ресурсів пристроїв. Це потребує створення розподілених механізмів моніторингу, оброблення збоїв та координації дій між різними рівнями інфраструктури. На основі цього було сформовано концепцію системи багаторівневої відмовостійкості, що передбачає незалежну роботу агентів на кожному рівні мережі та використання подієво-орієнтованого каналу обміну даними для виявлення і локалізації відмов.

У роботі запропоновано архітектуру системи забезпечення відмовостійкості, побудовану за принципом розподіленої взаємодії між агентами. Система включає модулі збору, обробки, зберігання та передавання подій, що дозволяє здійснювати асинхронну обробку повідомлень і підтримувати працездатність навіть у разі часткових відмов компонентів. Такий підхід підвищує адаптивність, масштабованість і надійність IoT-інфраструктури, зменшуючи ризик каскадного поширення збоїв та скорочуючи час реакції системи на помилки.

Експериментальна частина роботи підтвердила ефективність запропонованого підходу. Було створено тестове середовище Інтернету речей, у межах якого проведено порівняння двох варіантів функціонування системи – із використанням розробленої системи забезпечення відмовостійкості та без неї. Результати експериментів показали, що

впровадження запропонованої архітектури позитивно впливає на показники стабільності, зменшує затримки під час виявлення та усунення збоїв і забезпечує безперервність обслуговування навіть у разі виникнення відмов у окремих вузлах мережі.

Отже, можна зробити висновок, що запропонована система забезпечення відмовостійкості та надійності інфраструктури Інтернету речей є ефективним рішенням для побудови сучасних розподілених IoT-систем. Вона дозволяє підвищити рівень надійності, скоротити час простоїв, мінімізувати втрати даних і забезпечити стабільне функціонування мережі в умовах часткових збоїв. Розроблений підхід може бути використаний при проектуванні систем промислового Інтернету речей, розумних будівель, транспортних платформ, енергетичних та екологічних систем моніторингу, де критично важливо гарантувати безперервність і надійність роботи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Жураковський Б.Ю., Зенів І.О. Технології інтернету речей: навчальний посібник [Електронний ресурс] / Б.Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Київ: КПІ ім. Ігоря Сікорського, 2021. – 271 с.

2. Інтернет-речей: що це і як він працює [Електронний ресурс] – Режим доступу: https://gigatrans.ua/ua/news/internet-vesh-ey-chto-eto-takoe-i-kak-on-rabotaet

3. Інтернет речей [Електронний ресурс] // Вікіпедія. – Режим доступу: https://uk.wikipedia.org/wiki/Інтернет_речей

4. Шакуров Є.О., Балюк О.С. Поняття «Інтернет речей», способи застосування та технології побудови «Інтернету речей» / Є.О. Шакуров, О.С. Балюк // Наумовські читання: зб. тез доп. учасників XX Всеукр. наук.-метод. конф. здобувачів вищ. освіти та молодих вчених, присвяч. 300-річчю з дня народж. Г.С. Сковороди, Харків, 3–4 листоп. 2022 р. / Харків. нац. пед. ун-т ім. Г.С. Сковороди; за заг. ред. О.А. Жерновникової. – Харків, 2022. – С. 191–193.

5. Що таке IoT технологія та як вона впливає на різні галузі [Електронний ресурс] // Київстар Бізнес Хаб. – Режим доступу: https://hub.kyivstar.ua/articles/shho-take-iot-tehnologiya-ta-yak-vona-vplyvaye-na-rizni-galuzi

6. Самойленко М.Ю. Принципи застосування технології Інтернет речей у сучасному світі техніки / М.Ю. Самойленко // Вчені записки ТНУ імені В.І. Вернадського. Серія: технічні науки. – 2020. – Т. 31, № 70, ч. 1. – С. 142–148. – Режим доступу: https://doi.org/10.32838/TNU-2663-5941/2020.6-1/24

7. Яковина В.С., Сенів М.М. Основи теорії надійності програмних систем: навчальний посібник / В.С. Яковина, М.М. Сенів. – Львів: Видавництво Львівської політехніки, 2020. – 248 с.

8. Яковина В.С., Федасюк Д.В., Сенів М.М., Нитребич О.О. Моделі, методи та засоби аналізу надійності програмних систем: монографія / В.С. Яковина, Д.В. Федасюк, М.М. Сенів, О.О. Нитребич. – Львів: Видавництво Львівської політехніки, 2015. – 220 с.

10. Василевський О.М., Поджаренко В.О. Нормування показників надійності технічних засобів: навчальний посібник / О.М. Василевський, В.О. Поджаренко. – Вінниця: ВНТУ, 2010. – 129 с.

11. Agrawal A., Toshniwal D. Fault Tolerance in IoT: Techniques and Comparative Study / A. Agrawal, D. Toshniwal // Asian Journal For Convergence In Technology (AJCT). – 2021. – Vol. 7, No. 1. – P. 49–52. – Режим доступу: https://doi.org/10.33130/AJCT.2021v07i01.011

12. Надійність комп'ютерних мереж // Вікі ЦДУ. – Режим доступу: https://wiki.cusu.edu.ua/index.php/Надійність_комп'ютерних_мереж

13. Kharchenko V., Odarushchenko O., Butenko V., Moskalets V., Odarushchenko E., Strjuk O. Application of Markov Modeling for Safety Modeling for Safety Assessment of Self-Diagnostic Programmable Instrumentations and Control Systems. Central European Researchers Journal. 2016. Vol.2, Iss. 2. P. 61-69. URL: http://ceres-journal.eu/iss160202

14. Jammalamadaka K.R.S., Chokara B., Jammalamadaka S.B., Duvvuri B.K. Making IoT Networks Highly Fault-Tolerant Through Power Fault Prediction, Isolation and Composite Networking in the Device Layer / K.R.S. Jammalamadaka, B. Chokara, S.B. Jammalamadaka, B.K. Duvvuri // Journal of Sensor and Actuator Networks. – 2025. – Vol. 14, No. 2. – P. 24. – Режим доступу: https://doi.org/10.3390/jsan14020024

15. VMware. What is Fault Tolerance? Definition & FAQs [Електронний ресурс] / VMware. – Режим доступу: https://www.vmware.com/topics/fault-tolerance

16. Аль-Амморі А. Елементи теорії надійності та інформаційної безпеки комп’ютеризованих систем / А. Аль-Амморі. – Київ: Ліра-К, 2024. – 282 с.

17. Petrovska I., Kuchuk H., Mozhaiev M. Features of the distribution of computing resources in cloud systems / I. Petrovska, H. Kuchuk, M. Mozhaiev // 2022 IEEE 4th KhPI Week on Advanced Technology, KhPI Week 2022 – Conference Proceedings. – 03–07 Oct. 2022.

18. Kumar S., Ranjan P., Singh P., Tripathy M.R. Design and Implementation of Fault Tolerance Technique for Internet of Things (IoT) / S. Kumar, P. Ranjan, P. Singh, M.R. Tripathy // 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), Bhimtal, India. – 2020. – P. 154–159.

19. MQTT [Електронний ресурс] // Вікіпедія. – Режим доступу: https://uk.wikipedia.org/wiki/MQTT

20. Протокол MQTT [Електронний ресурс] – Режим доступу: https://pupenasan.github.io/TI40/Лекц/MQTT.html