**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМЕНІ С. З. ҐЖИЦЬКОГО**

**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

# КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему: **«Розробка та дослідження системи моніторингу мережевої інфраструктури підприємства»**

Виконав: здобувач освіти групи Іт-62

Спеціальності 126 «Інформаційні системи та технології»

(шифр і назва)

Батрон Олег Орестович

(Прізвище та ініціали)

Керівник: к.е.н., доцент, Шувар Б. І.

(Прізвище та ініціали)

Рецензент: _____

(Прізвище та ініціали)

**ДУБЛЯНИ-2025**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМЕНІ С.З.ҐЖИЦЬКОГО

**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»
Завідувач кафедри _____
_____
«____» _____ 202_ р.

**ЗАВДАННЯ**
на кваліфікаційну роботу студенту
Батрона Олега Орестовича
(прізвище, ім'я, по батькові)

1. Тема роботи: *«Розробка та дослідження системи моніторингу мережевої інфраструктури підприємства»*
   Керівник роботи *к.е.н., доцент, Шувар Богдан Іванович*
   (наук.ступінь, вч. звання, прізвище, ініціали)
   затверджені наказом по університету від 28.02.2025 року №140/к-с.
2. Строк подання студентом роботи 05.12.2025р.
3. Вихідні дані до роботи: мережеве обладнання (маршрутизатори, комутатори, точки доступу Wi-Fi, UPS), параметри мережевих протоколів ICMP, ARP, SNMP, журнали apcupsd/WinPower, конфігураційні дані вузлів та логіка контролю доступності й стану інфраструктури.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)
   *Вступ; 1. Аналіз стану питання та постановка задачі; 2. Обґрунтування і вибір інструментарію; 3. Проєктування і реалізація системи моніторингу; 4. Охорона праці та безпека в надзвичайних ситуаціях; 5. Визначення ефективності системи; Висновки; Список використаних джерел; Додатки.*
5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових схем та моделей): *Модель OSI; Логічна топологія мережі підприємства; Архітектура розробленої системи; Діаграма розгортання; Структура бази даних; Use-case діаграма адміністратора; Послідовність формування інциденту; Веб-інтерфейс системи (панель вузлів, графіки, інциденти, UPS); Виявлені невідомі пристрої; Приклад записів журналу;*

6. Консультанти з розділів:

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| 1-3 | *Шувар Б. І., доцент кафедри інформаційних технологій* | | |
| 4 | *Городецький І. М., доцент кафедри інженерної механіки* | | |
| 5 | *Шувар Б. І., доцент кафедри інформаційних технологій* | | |

7. Дата видачі завдання 28.02.2025 р.

Календарний план

| № з/п | Назва етапів дипломного проєкту | Терміни виконання етапів роботи | Примітка |
|---|---|---|---|
| 1. | *Написання першого розділу* | 28.02.2025 – 20.03.2025 | |
| 2. | *Виконання другого розділу та аркушів ілюстраційного матеріалу до нього* | 21.03.2025 – 14.06.2025 | |
| 3. | *Виконання третього, четвертого розділів та аркушів ілюстраційного матеріалу до нього* | 15.06.2025 – 10.07.2025 | |
| 4. | *Написання розділу «Охорона праці»* | 11.07.2025 – 31.08.2025 | |
| 5. | *Завершення оформлення розрахункового-пояснювальної записки та аркушів ілюстраційного матеріалу* | 01.09.2025 – 31.10.2025 | |
| 6. | *Завершення роботи цілому* | 01.11.2025 – 08.12.2025 | |

Студент _____, Батрон О. О.
(підпис) (прізвище, ініціали)

Керівник роботи _____, Шувар Б.І.
(підпис) (прізвище, ініціали)

*

**УДК 004.7:004.4:004.056**

**Розробка та дослідження системи моніторингу мережевої інфраструктури підприємства.**

Батрон О. О., Кафедра інформаційних технологій – ЛНУВМБ ім. С. З. Ґжицького, Дубляни, 2025.

Кваліфікаційна робота: 70 с., 12 рисунків, 7 таблиць, 29 джерел.

У роботі розглянуто сучасні підходи до моніторингу мережевої інфраструктури та проаналізовано роль ICMP-, ARP- та SNMP-опитування у виявленні відмов і деградацій мережевих вузлів. Визначено особливості роботи з динамічною адресацією, MAC-ідентифікацією пристроїв і контролем доступності у контексті еталонної моделі OSI. На основі проведеного аналізу реалізовано програмну систему, яка включає модуль збору даних, механізм виявлення невідомих пристроїв, засоби аналізу логів джерел безперебійного живлення та розрахунок інтегрального показника health_score.

У рамках роботи створено веб-інтерфейс адміністратора на базі Streamlit, що забезпечує відображення стану вузлів, динаміки параметрів ping та health_score, журналів інцидентів і сповіщень, а також інформації щодо UPS. Виконано експериментальне тестування із моделюванням типових мережевих відмов і здійснено порівняння з відомими системами моніторингу, такими як Zabbix, Nagios, PRTG та LibreNMS. Результати дослідження підтвердили працездатність, стабільність роботи та придатність розробленої системи для використання на підприємствах малого і середнього масштабу.

Метою роботи було створення та дослідження програмної системи для автоматизованого контролю мережевої інфраструктури. Досягнення мети забезпечено шляхом проєктування архітектури, реалізації основних програмних модулів і проведення експериментальної оцінки їх ефективності. Практична значущість полягає в отриманні інструменту, що дає змогу своєчасно виявляти інциденти, зменшувати ризики простоїв і полегшує роботу адміністратора мережі.

**Ключові слова:** моніторинг мережі, ICMP, ARP, SNMP, health_score, інциденти, UPS, мережеві вузли, Streamlit, SQLite.

**UDC 004.7:004.4:004.056**

**Development and Study of a Network Infrastructure Monitoring System for an Enterprise.**

Batron O. O., Department of Information Technologies – Stepan Gzhytskyi Lviv National University of Veterinary Medicine and Biotechnologies, Dubliany, 2025.

Qualification paper: 70 p., 12 figures, 7 tables, 29 sources.

The paper examines modern approaches to monitoring enterprise network infrastructures and analyses the role of ICMP, ARP and SNMP-based polling in detecting failures and performance degradation of network nodes. The study highlights the specifics of working with dynamic IP addressing, MAC-based device identification, and availability assessment in the context of the OSI reference model. Based on this analysis, a software system was developed, incorporating a data collection module, a mechanism for detecting unknown devices, processing of UPS log data, and calculation of the integral health_score metric.

A web interface for the administrator was implemented using Streamlit, providing visualisation of node states, time-series graphs of ping and health_score, incident and notification logs, as well as information from uninterruptible power supplies. Experimental testing was performed using simulated network failure scenarios, and the obtained results were compared with well-known monitoring systems such as Zabbix, Nagios, PRTG and LibreNMS. The evaluation confirmed the correctness of the system's operation and its suitability for small and medium-sized enterprise networks.

The purpose of the work was to develop and investigate a software system for automated monitoring of network infrastructures. This goal was achieved through architectural design, implementation of the main software modules, and experimental evaluation. The practical significance lies in providing a lightweight monitoring tool capable of promptly detecting incidents, reducing downtime risks, and simplifying the work of a network administrator.

**Keywords:** network monitoring, ICMP, ARP, SNMP, health_score, incidents, UPS, network nodes, Streamlit, SQLite.

# ЗМІСТ

# ВСТУП

Сучасні підприємства значною мірою залежать від стабільної роботи мережевої інфраструктури, від якої залежить доступ до корпоративних сервісів, інформаційних ресурсів і критичних бізнес-процесів. Динамічні DHCP-середовища, зміна ARP-таблиць, велика кількість дротових і бездротових клієнтів, вимоги до контролю роботи обладнання через SNMP та наявність джерел безперебійного живлення формують потребу у програмній системі моніторингу, здатній забезпечувати комплексне спостереження за показниками рівнів L2–L3 і L4–L7. У таких умовах недостатньо обмежуватися пінг-моніторингом або базовими SNMP-перевірками, оскільки для реального контролю інфраструктури необхідні механізми ідентифікації пристроїв за MAC-адресами, виявлення невідомих вузлів, аналізу стану портів, поведінки Wi-Fi клієнтів і обробки даних UPS.

Об’єктом дослідження є мережна інфраструктура підприємства, а предметом - методи її моніторингу з використанням активного та пасивного збору даних, аналізу доступності, сервісної якості, параметрів обладнання і показників джерел безперебійного живлення. Метою роботи є розробка та дослідження програмної системи моніторингу, яка поєднує ICMP-контроль, ARP-сканування, SNMP-опитування, перевірку L4–L7 сервісів та інтеграцію з UPS на основі логів apcupsd і WinPower, а також формує інтегральний показник health_score для числової оцінки стану кожного вузла.

Для досягнення мети використовуються методи системного аналізу мережевої інфраструктури, процедури ICMP- та SNMP-моніторингу, регулярне ARP-сканування для оновлення відповідності MAC–IP, аналіз логів UPS, а також методи нормалізації і зважування показників при обчисленні health_score. Експериментальна частина роботи охоплює моделювання типових інцидентів, включаючи втрату доступності, появу невідомих пристроїв, зміну сервісної відповіді та переходи UPS на автономне живлення, що дозволяє оцінити швидкість і точність реакції системи.

Наукова новизна полягає у поєднанні активного й пасивного моніторингу L2–L7 показників із включенням UPS-параметрів до загальної моделі стану мережі та використанням інтегрального показника health_score для автоматизованої класифікації станів вузлів. Практична цінність системи полягає у можливості її застосування на підприємствах із динамічними мережевими структурами, де необхідно забезпечити надійний контроль без впровадження ресурсомістких NMS-платформ.

Структура роботи включає вступ, п’ять розділів, висновки, список джерел та додатки. Перший розділ містить аналітичний огляд стану питання у теорії та практиці моніторингу мереж і формулює постановку задачі. У другому розділі описано об’єкт моніторингу, обґрунтовано вибір технологій, спроєктовано архітектуру програмної системи та модель даних. Третій розділ присвячено реалізації модулів збору даних, системи сповіщень, веб-інтерфейсу, інтеграції з UPS і дослідженню роботи розробленого рішення у експериментальних сценаріях. У четвертому розділі розглянуто питання охорони праці та безпеки у надзвичайних ситуаціях в умовах експлуатації мережевої інфраструктури та джерел безперебійного живлення. П’ятий розділ містить методику визначення технічної, експлуатаційної та економічної ефективності впровадження системи моніторингу на підприємстві, а також порівняння її ефективності з існуючими системами моніторингу.

# РОЗДІЛ 1.
# АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАВДАННЯ

## 1.1. Роль мережевої інфраструктури у діяльності сучасного підприємства

Мережева інфраструктура сучасного підприємства є критичним складником інформаційно-технологічного середовища, оскільки забезпечує доступ до серверних сервісів, корпоративних ресурсів, систем документообігу, механізмів автентифікації, засобів взаємодії між підрозділами та зовнішніми інформаційними системами. В умовах цифровізації та активного впровадження віддалених сервісів функціонування підприємства дедалі більше залежить від стабільності, пропускної здатності, керованості та доступності мережі. Втрата працездатності мережевих сегментів призводить до зупинки ключових процесів, підвищення часу простою та неможливості виконання технологічних операцій, що безпосередньо впливає на фінансові та організаційні показники діяльності підприємства. Тому мережа розглядається як основа корпоративної інфраструктури, що забезпечує не лише транспорт даних, а й механізми контролю доступу, логічного розмежування ресурсів, підтримки критичних сервісів та безпечного обміну інформацією між підсистемами підприємства.

Розвиток мережевих технологій призводить до поступового ускладнення корпоративних мереж, у яких одночасно працюють різнорідні пристрої, включаючи комутатори доступу, маршрутизатори, точки бездротового доступу, серверне обладнання, термінальні клієнти, периферійні пристрої та обладнання, що підтримує SNMP. Формування єдиного інформаційного простору відбувається за рахунок інтеграції дротових і бездротових сегментів, використання кількох VLAN, впровадження VPN-тунелів та механізмів маршрутизації, що динамічно адаптуються до змін топології. У таких середовищах критично важливим є контроль параметрів рівнів L2–L3, які

визначають роботу ARP, ICMP, протоколів маршрутизації, DHCP та механізмів комутації, а також параметрів рівнів L4–L7, що визначають доступність прикладних сервісів, швидкість встановлення з'єднання, якість DNS-обслуговування та роботу мережевих застосунків.

Додатковим чинником складності є динамічна адресація, характерна для більшості сучасних мереж. DHCP-процедури, що регулярно оновлюють IP-адреси клієнтів, ускладнюють ідентифікацію пристроїв та супровід інцидентів, оскільки прив'язка стану вузла до IP стає ненадійною. За цих умов критичною є ідентифікація пристроїв за MAC-адресами та постійний аналіз ARP-таблиць, який дає змогу точно визначати переміщення вузлів по мережі, появу нових або неавторизованих пристроїв, нестабільність оренд DHCP та ознаки аномальної поведінки трафіку. Роль такого аналізу суттєво зростає в умовах наявності великої кількості бездротових клієнтів, де зміна точок доступу та коливання параметрів Wi-Fi призводять до появи додаткових подій на рівні доступу, які також необхідно враховувати під час моніторингу.

Важливою складовою мережевої інфраструктури є системи безперебійного живлення, від роботи яких залежить доступність серверів, активного мережевого обладнання та критичних сервісів. Події переходу на живлення від батареї, зниження заряду, зміна навантаження та інші параметри UPS безпосередньо впливають на функціонування інфраструктури, а тому повинні бути інтегровані в загальну систему контролю[15-17]. Облік таких показників у режимі реального часу дозволяє не лише фіксувати аварійні ситуації, а й прогнозувати ризики деградації сервісів до моменту настання інциденту.

Таким чином, у сучасних умовах мережева інфраструктура є не лише каналом передавання даних, а комплексною системою, стійкість і працездатність якої визначають ефективність функціонування підприємства. Зростання складності мереж, динамічність їх топології, різнорідність обладнання та потреба в оперативному контролі параметрів L2–L7 обумовлюють необхідність застосування спеціалізованих програмних систем моніторингу, здатних забезпечити комплексний аналіз мережевих процесів і формувати достовірну

оцінку стану інфраструктури підприємства[4, 6, 7]. У цьому контексті моніторинг розглядається як невід'ємний компонент експлуатації мережі, що забезпечує своєчасне виявлення інцидентів, зменшення часу відмов і підтримку високого рівня доступності корпоративних сервісів.

### 1.2. Підходи до моніторингу мережевої інфраструктури

Моніторинг мережевої інфраструктури базується на поєднанні активних і пасивних методів спостереження за параметрами різних рівнів еталонної моделі OSI. Саме ця модель використовується як концептуальна основа для класифікації мережевих процесів, що відбуваються у корпоративному середовищі, а також для впорядкування груп показників, які збирає система[1-3]. На рисунку 1.1 наведена узагальнена структура рівнів OSI, що відображає їхню роль у функціонуванні мереж та контролі стану інфраструктури. У подальшому тексті до цієї схеми здійснюється систематичне посилання у контексті пояснення метрик L2–L7.

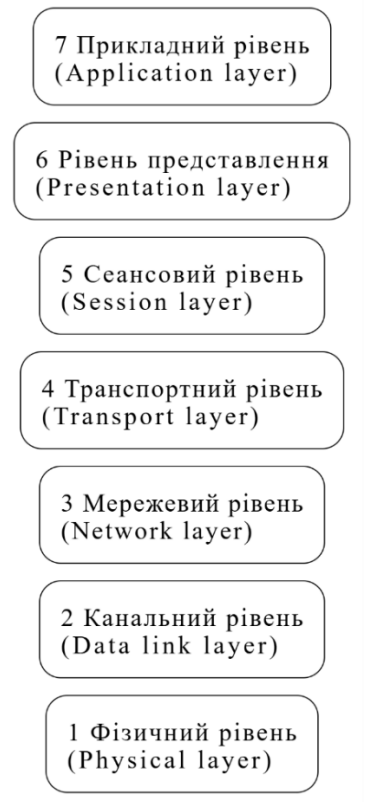


Рисунок 1.1 - Рівні еталонної моделі OSI

Активний моніторинг передбачає генерування контрольного трафіку або виконання цілеспрямованих запитів до обладнання. На рівні L3 основним механізмом виступає ICMP-контроль, що дозволяє вимірювати час відгуку, втрати пакетів, варіацію затримки та доступність критичних маршрутів[1, 3]. У середовищах підприємств ICMP використовується для оперативного виявлення деградацій каналів, нестабільності шлюзів або періодичної недоступності зовнішніх вузлів. Доповненням до ICMP є SNMP-полінг, який забезпечує глибший рівень спостереження за параметрами L2–L3. Через SNMP система отримує статистику інтерфейсів, швидкість прийому та передавання трафіку, кількість помилок RX/TX, dropped packets, CRC-errors, frame errors, стан портів комутаторів, а також системні показники серверів і маршрутизаторів: CPU load, memory usage, uptime та температурні значення[1, 4, 29]. На рівнях L4–L7 активний моніторинг охоплює перевірку доступності сервісів, час встановлення TCP-з'єднання, стабільність відповіді HTTP/HTTPS, коректність роботи DNS, а також контроль стану VPN-тунелів[1, 3].

Пасивний моніторинг доповнює активні методи, оскільки працює без створення додаткового трафіку, аналізуючи вже наявні таблиці, логи та події. На рівні L2 ключовим механізмом є аналіз ARP-таблиць, що дозволяє встановлювати відповідність MAC–IP, фіксувати появу нових пристроїв, виявляти ARP-flapping, зміни поведінки клієнтів або нестабільність DHCP-оренд[1, 4]. Обробка multicast і broadcast активності, контроль змін портів комутаторів і фіксація routing changes дають змогу визначати нетипові події, які можуть свідчити про технічні або мережеві інциденти. Окреме значення має пасивний аналіз логів UPS, що забезпечує збір даних про переходи на акумулятор, рівень навантаження, тривалість автономної роботи та зміну стану батареї без опитування обладнання.

Комплексний моніторинг інфраструктури можливий лише при узгодженому використанні активних і пасивних методів, оскільки вони відображають різні аспекти роботи мережі[1-3]. Активні механізми дозволяють швидко фіксувати втрати доступності або зміни продуктивності сервісів, тоді як

пасивні дають змогу виявляти причини інцидентів, простежувати історію подій, контролювати поведінку мережевих пристроїв і отримувати дані, які не можна визначити за допомогою простого ICMP чи сервісних перевірок. Використання моделі OSI як структурної основи забезпечує можливість цілісного аналізу стану мережі, оскільки всі групи показників - від ARP до HTTP - впорядковуються за рівнями, що створює логічний каркас для подальшої інтерпретації зібраних результатів[1, 2].

### 1.3. Огляд існуючих систем моніторингу мережі

Сучасні рішення для моніторингу мережевої інфраструктури представлені широким спектром систем, що різняться архітектурою, методами збору даних, вимогами до ресурсів та способами інтеграції у корпоративне середовище. Найпоширенішими NMS-платформами є Zabbix, Nagios Core, PRTG Network Monitor та LibreNMS, які реалізують як активний, так і пасивний моніторинг, підтримують протоколи ICMP, SNMP та різні механізми перевірки сервісів[22-25]. Хоча їхня функціональність достатньо розвинена, кожна система має власні архітектурні обмеження, що впливають на швидкість розгортання, можливість адаптації до мереж з динамічною адресацією та гнучкість збору даних на рівнях L2–L3.

Zabbix характеризується централізованою архітектурою з використанням агента, SNMP-полінгу, ICMP-моніторингу та розвиненої системи тригерів. Платформа орієнтована на середні та великі мережі, але потребує розгортання серверної частини, бази даних і додаткових компонентів[22]. Nagios Core базується на плагінній архітектурі, що дозволяє розширювати функціональність, однак вимагає значних зусиль для конфігурації та не надає повноцінного вбудованого механізму збору L2-показників[23]. PRTG Network Monitor орієнтований на зручність використання та має розвинене GUI, але використовує пропрієтарну модель і обмежується ліцензійними сенсорами[25]. LibreNMS підтримує SNMP-автовиявлення, розширені графіки та інтеграції, проте зберігає

значну залежність від SNMP і потребує належного налаштування серверної інфраструктури[24].

Для узагальнення ключових характеристик наведено порівняльну характеристику розглянутих систем у таблиці 1.1.

Таблиця 1.1 - Порівняльна характеристика поширених систем моніторингу

| **Система** | **Архітектура** | **Методи збору даних** | **Підтримка L2/L3** | **Підтримка L4–L7** | **Особливості** |
|---|---|---|---|---|---|
| **Zabbix** | Централізована, сервер + агент | SNMP, ICMP, агент, HTTP-чек, пінг-моніторинг | L3 частково L2 (через SNMP) | HTTP, HTTPS, DNS, TCP-чек | Висока функціональність, складніше розгортання |
| **Nagios Core** | Модульна, плагінна | Плагіни, ICMP, сервісні чеки | Мінімальна L2 | Широка підтримка сервісів | Потрібне ручне налаштування, немає повноцінної візуалізації |
| **PRTG** | Централізована | SNMP, ICMP, WMI, пакетні сенсори | L2 через SNMP | Широка підтримка сервісів | Пропрієтарна модель, ліміти сенсорів |
| **LibreNMS** | Розподілена, SNMP-орієнтована | SNMP, ICMP, автодетект | L2 повністю (MAC, порти) | Сервісні чеки через плагіни | Вільне ПЗ, акцент на SNMP |

Порівняння показує, що існуючі NMS-рішення орієнтовані переважно на статичну або наперед визначену структуру мережі та спираються на SNMP як основний механізм збору даних. У динамічних DHCP-середовищах, де значну частину процесів визначають зміни ARP-таблиць, поведінка MAC-адрес, події на рівні доступу, а також необхідність врахування даних UPS, такі системи втрачають частину функціональної точності або потребують складної додаткової конфігурації. Це створює передумови для розроблення програмної системи, здатної виконувати як повноцінний активний моніторинг L2–L7, так і детальний пасивний аналіз ARP, DHCP історії, портових подій і логів джерел безперебійного живлення.

### 1.4. Постановка задачі та формулювання вимог до системи

На основі проведеного аналізу існуючих підходів до моніторингу мережевої інфраструктури та особливостей функціонування корпоративних мереж формулюється задача створення програмної системи, яка забезпечує комплексний контроль параметрів роботи мережі на рівнях L2–L7 із використанням активних і пасивних механізмів збору даних. Система має працювати в умовах динамічної адресації, коректно ідентифікувати пристрої за MAC-адресами, підтримувати збір телеметрії з мережевого обладнання, сервісів та джерел безперебійного живлення, а також забезпечувати фіксацію інцидентів і надання узагальненої оцінки стану інфраструктури. Програмний комплекс повинен виступати єдиною точкою спостереження за станом мережі, об’єднувати різнорідні дані у спільну модель та підтримувати прийняття рішень адміністратором мережі.

Постановка задачі передбачає розроблення програмного рішення, яке реалізує автоматизований збір параметрів доступності мережевих вузлів, статистики інтерфейсів, поведінки ARP, станів портів, показників прикладних сервісів та UPS, їх збереження у розширеній структурі бази даних та інтерпретацію результатів вимірювань у вигляді інцидентів і агрегованих метрик. Ключовим елементом інтерпретації даних виступає інтегральний показник health_score, який формується як зважена сума часткових оцінок, отриманих за ICMP, SNMP, сервісними, Wi-Fi та UPS-показниками, і використовується для стандартизованої оцінки стану вузлів. На основі значень health_score система має забезпечувати можливість класифікації станів обладнання та автоматизованого формування подій.

З метою формального окреслення очікувань до розроблюваного програмного комплексу вимоги до системи доцільно поділити на функціональні та нефункціональні. Функціональні вимоги визначають, які саме дії реалізує система з точки зору збору, оброблення та відображення даних, тоді як нефункціональні описують якісні характеристики, пов’язані з надійністю,

продуктивністю, масштабованістю, зручністю розгортання та експлуатації. Узагальнений перелік вимог наведено у таблиці 1.2, яка виступає основою для подальшого проєктування архітектури, модулів збору даних і структури бази даних.

Таблиця 1.2 - Функціональні та нефункціональні вимоги до системи

| **Категорія** | **Вимога** |
|---|---|
| Функціональні | Активний ICMP-контроль доступності мережевих вузлів (час відгуку, втрати пакетів). |
| | Регулярне ARP-сканування з актуалізацією відповідності MAC–IP. |
| | Виявлення невідомих та неавторизованих пристроїв за результатами ARP- та DHCP-даних. |
| | Збір SNMP-статистики інтерфейсів (швидкість, завантаження, помилки, dropped packets). |
| | Отримання системних SNMP-показників обладнання (CPU load, memory usage, uptime, температура). |
| | Контроль сервісів L4–L7 (HTTP/HTTPS, DNS, TCP-з’єднання, VPN-тунелі). |
| | Інтеграція з логами apcupsd і WinPower для збору UPS-показників. |
| | Фіксація інцидентів і ведення журналу подій та журналу сповіщень. |
| | Обчислення інтегрального показника health_score на основі часткових оцінок. |
| | Відображення вузлів, метрик, інцидентів, UPS та невідомих пристроїв у веб-інтерфейсі Streamlit. |
| Нефункціональні | Надійна робота в режимі безперервного збору телеметрії без втрати даних. |
| | Коректне функціонування у DHCP-середовищах із використанням MAC як базового ідентифікатора. |
| | Мінімальний додатковий трафік завдяки поєднанню активного і пасивного моніторингу. |
| | Масштабованість архітектури при збільшенні кількості вузлів і обсягу даних. |
| | Простота інсталяції та розгортання на серверних платформах з обмеженими ресурсами. |
| | Підтримка безперервної роботи модулів збору даних у фоновому режимі. |
| | Забезпечення цілісності журналів подій і стабільної роботи з базою даних SQLite. |

Сформульовані вимоги визначають основні напрями подальшого проєктування програмного комплексу, включаючи вибір мов програмування, бібліотек і засобів розробки, структуризацію модулів збору даних, проєктування бази даних та організацію веб-інтерфейсу. У наступному розділі виконується обґрунтування інструментарію, що використовується для реалізації системи моніторингу, та розроблення її загальної архітектури.

# РОЗДІЛ 2.
# ОБҐРУНТУВАННЯ, ВИБІР ТА РЕАЛІЗАЦІЯ ІНСТРУМЕНТАРІЮ ВИРІШЕННЯ ЗАДАЧІ

## 2.1. Технічні характеристики мережевої інфраструктури підприємства як об’єкта моніторингу

Мережева інфраструктура підприємства, що розглядається як об’єкт моніторингу, побудована за багаторівневою схемою з виділенням окремих логічних сегментів та використанням VLAN для розмежування трафіку. Доступ до зовнішніх ресурсів здійснюється через граничний маршрутизатор, який виконує функції шлюзу за замовчуванням для внутрішньої мережі, реалізує політику фільтрації трафіку та здійснює трансляцію адрес. Усередині локальної мережі використовується ієрархічна структура з ядром на базі L3-комутатора або маршрутизатора, до якого підключені комутатори доступу, серверний сегмент та обладнання бездротової мережі. Такий підхід забезпечує можливість централізованого керування маршрутизацією між VLAN, спрощує сегментацію трафіку та створює основу для подальшого моніторингу параметрів на рівнях L2–L3.

В інфраструктурі підприємства логічно виділено окремий VLAN для користувачів, у якому розміщуються робочі станції та інші клієнтські пристрої, що отримують IP-адреси за допомогою DHCP. Окремо налаштовано VLAN керування, який використовується для доступу до інтерфейсів керування комутаторами, маршрутизаторами, серверним обладнанням та, за потреби, до модулів моніторингу джерел безперебійного живлення. Таке розділення дозволяє ізолювати службовий трафік керування від користувацького трафіку, зменшити ризики несанкціонованого доступу до мережевого обладнання та забезпечити більш точний контроль параметрів інфраструктури. Сервери та критичні служби

можуть розміщуватися в окремому VLAN, що додатково ізолює їх від робочих станцій і спрощує застосування політик доступу.

Дротовий сегмент включає комутатори доступу, до яких підключені робочі місця, периферійні пристрої та обладнання, що підтримує протоколи верхніх рівнів. Через мобільність користувачів і можливі переміщення обладнання відповідність MAC–IP змінюється в часі, тому об’єкт моніторингу має виражений динамічний характер. Для коректного аналізу стану мережі необхідним є урахування подій на рівні портів комутаторів, змін ARP-таблиць, появи нових MAC-адрес у користувацькому VLAN, а також контролю показників, що характеризують якість роботи інтерфейсів, таких як помилки RX/TX, dropped packets, CRC-errors та рівень broadcast і multicast активності.

До складу інфраструктури входить серверний сегмент, у якому працюють служби, що забезпечують роботу підприємства: веб-ресурси, DNS, системи автентифікації, файлові сервіси, VPN-шлюзи та інші прикладні системи. Доступність цих сервісів та їхня стабільність визначають працездатність бізнес-процесів, а тому серверний сегмент є одним з основних об’єктів спостереження. У контексті моніторингу важливими є як мережеві параметри, пов’язані з маршрутами доступу до серверів, так і системні показники, які характеризують навантаження на обчислювальні ресурси.

Бездротовий сегмент побудований на основі точок доступу, підключених до комутаторів доступу та, за необхідності, виділених у окремий VLAN. Він забезпечує підключення мобільних користувачів та пристроїв, що потребують бездротового доступу до корпоративних сервісів. Для цього сегмента характерні такі параметри, як кількість одночасних клієнтів, рівень сигналу RSSI, кількість повторних передач і завантаження радіоканалу, які у сукупності впливають на якість мережевого обслуговування та повинні враховуватися при моніторингу.

Суттєвим компонентом інфраструктури є система безперебійного живлення, яка забезпечує роботу ключових вузлів – серверів, комутаторів, маршрутизаторів та іншого критичного обладнання – у разі порушень електропостачання. Джерела безперебійного живлення формують журнали подій

із показниками вхідної та вихідної напруги, навантаження, рівня заряду, переходів на живлення від батареї та тривалості автономної роботи. Ці дані є невід'ємною частиною об'єкта моніторингу, оскільки визначають ступінь стійкості мережевої інфраструктури до зовнішніх впливів.

Логічна топологія мережевої інфраструктури підприємства, що є об'єктом моніторингу, узагальнено наведена на рисунку 2.1. На схемі відображено основні елементи - граничний маршрутизатор, ядро мережі, комутатори доступу, користувацький VLAN, VLAN керування, серверний сегмент, бездротову інфраструктуру та джерела безперебійного живлення, що дозволяє наочно продемонструвати структуру об'єкта, для якого розробляється програмна система моніторингу.

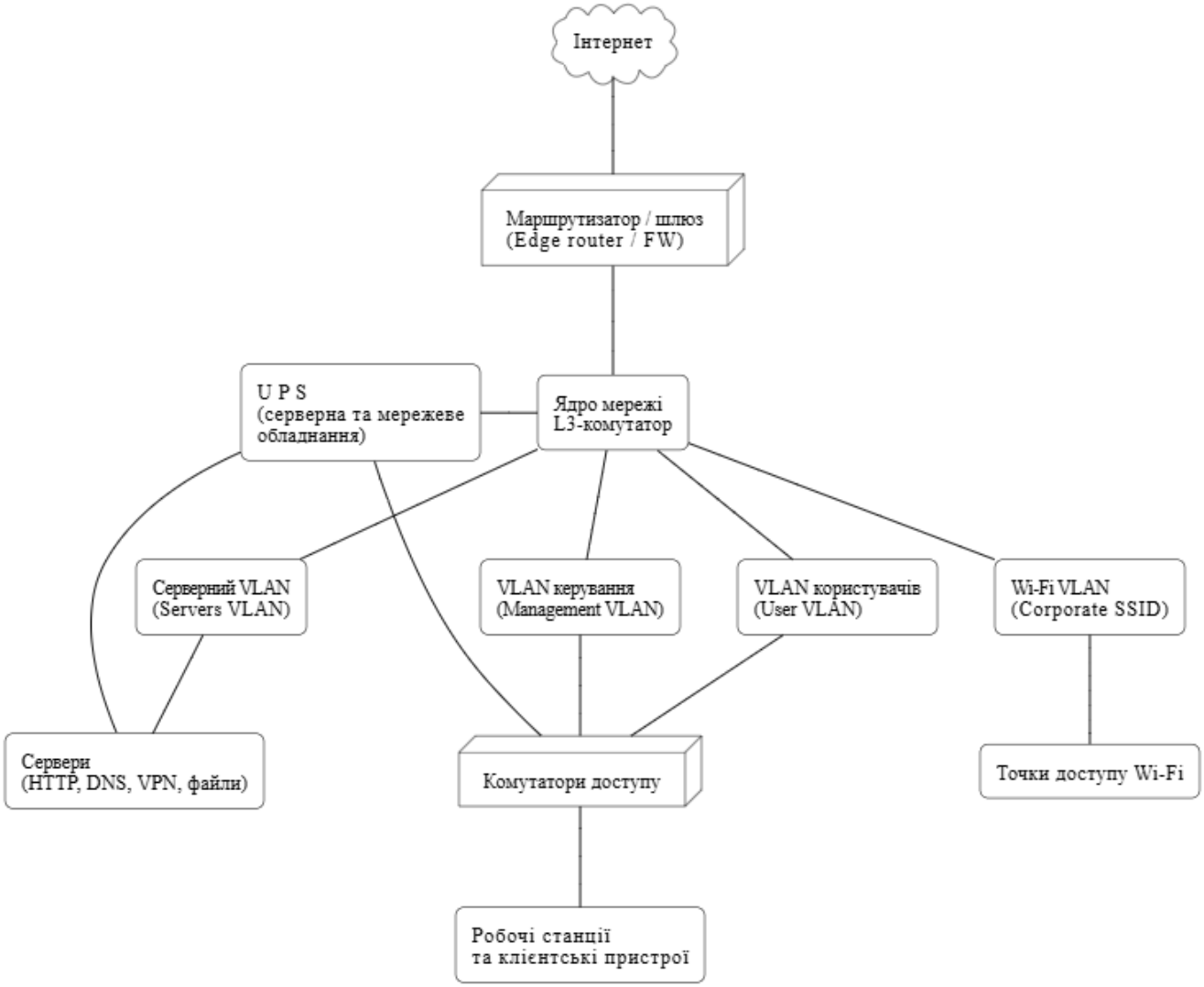


Рисунок 2.1 - Логічна топологія мережевої інфраструктури

## 2.2. Вибір технологій та середовища розробки системи

Вибір технологій для створення програмної системи моніторингу визначається специфікою мережевої інфраструктури підприємства, вимогами до оброблення великого обсягу телеметрії, необхідністю підтримки протоколів L2–L7 та забезпеченням стабільної роботи у фоновому режимі. Центральне місце в архітектурі рішення займає програмна платформа, що повинна забезпечувати гнучкість розробки, широку підтримку бібліотек для мережевої взаємодії, засоби інтеграції з SNMP та можливість роботи з файловими журналами UPS, а також мати інструментарій для побудови веб-інтерфейсу й подання накопичених даних в інтерактивній формі.

Основною мовою реалізації обрано Python, оскільки вона забезпечує високий рівень універсальності при роботі з мережевими протоколами та надає широкий спектр бібліотек, що дозволяють ефективно реалізувати механізми активного та пасивного моніторингу. Python є оптимальним вибором для задач, пов’язаних із регулярним опитуванням мережевих вузлів через ICMP, аналізом ARP-таблиць, обробленням SNMP-телеметрії та контролем сервісів на рівнях L4–L7. Використання стандартних бібліотек та поширених модулів на кшталт subprocess, asyncio, psutil, pysnmp та socket дозволяє реалізувати стабільний багатопоточний збір даних, а також організувати сценарії, що виконуються з мінімальними затримками[27].

Для забезпечення роботи інтерфейсу моніторингу обрано Streamlit, який дозволяє швидко створювати інтерактивні веб-сторінки без необхідності розроблення серверної логіки вручну. Завдяки цьому веб-інтерфейс системи працює як самостійний застосунок, що відображає таблиці вузлів, графіки показників, журнали інцидентів, параметри UPS та виявлені невідомі пристрої. Streamlit дає можливість інтегрувати дані з бази SQLite, виконувати динамічне оновлення таблиць та відображати великі обсяги історичної телеметрії у вигляді графіків. Такий підхід суттєво спрощує процес взаємодії адміністратора мережі з системою та дозволяє забезпечити високу наочність стану інфраструктури[26].

Для зберігання даних використано СУБД SQLite, що забезпечує простоту розгортання, відсутність потреби у керуванні сервером баз даних та достатню продуктивність для задач моніторингу середніх і малих мереж. Обрана СУБД дозволяє працювати з розширеною структурою таблиць, що містить різнорідні групи телеметрії: ICMP-показники, SNMP-метрики, статистику інтерфейсів, історію ARP-подій, журнали сповіщень, дані UPS, сервісні перевірки та результати обчислення інтегральних оцінок. Важливою перевагою SQLite є стабільна робота при інтенсивному записі коротких транзакцій, що характерно для задач моніторингу, а також повна портативність файлу бази, що спрощує резервне копіювання та перенесення даних[28].

Окрему увагу приділено підтримці SNMP, для чого використовується бібліотека pysnmp, що забезпечує доступ до метрик мережевого обладнання, інтерфейсів комутаторів та системних параметрів серверів[27, 29]. Це дозволяє інтегрувати показники завантаження процесора, використання пам’яті, uptime, температурних датчиків, параметрів інтерфейсів та статистики прийому та передачі пакетів. SNMP-агенти, що працюють на обладнанні підприємства, виступають ключовими джерелами телеметрії на рівнях L2–L3 та L4 і дозволяють системі отримувати детальну картину стану мережевої інфраструктури.

Для роботи з логами джерел безперебійного живлення використовується оброблення текстових журналів apcupsd та WinPower, що забезпечує отримання параметрів напруги, навантаження, рівня заряду, часу автономної роботи та переходів на живлення від батареї. У межах розробленої системи ці журнали інтегруються у загальну базу даних, а їх значення враховуються при обчисленні часткової UPS-оцінки для формування інтегрального показника health_score.

Комплекс цих технологій дозволяє створити програмну систему, яка поєднує інструменти активного та пасивного моніторингу, обробляє дані на рівнях від L2 до L7, підтримує роботу з UPS, забезпечує стабільний фоновий збір телеметрії та надає адміністратору зручний механізм аналізу стану інфраструктури через інтерактивний веб-інтерфейс. Обраний інструментарій є

оптимальним для реалізації системи в умовах підприємства, оскільки поєднує легкість розгортання, широкі можливості інтеграції протоколів та достатню продуктивність для моніторингу складних мережевих середовищ.

### 2.3. Проєктування архітектури системи моніторингу

Архітектура розроблюваної системи моніторингу побудована за модульним принципом, що забезпечує незалежність компонентів, їх масштабованість та здатність працювати паралельно без взаємних блокувань. Система включає декілька ключових підсистем: модуль активного мережевого опитування, модуль пасивного аналізу ARP-таблиць, SNMP-модуль збору телеметрії, UPS-модуль оброблення журналів живлення, систему обчислення інтегральної оцінки health_score, підсистему фіксації інцидентів та веб-інтерфейс для адміністратора мережі. Усі компоненти працюють у фоновому режимі, взаємодіючи через спільну базу даних SQLite, яка виступає центральним сховищем телеметрії та журналів. Завдяки такій організації внутрішньої структури система забезпечує стабільність роботи навіть у випадку часткових відмов окремих модулів, оскільки кожен із них функціонує автономно та не впливає на працездатність інших компонентів.

Загальна логіка взаємодії компонентів наведена на рисунку 2.2. У структурі системи виділено модуль collector, що відповідає за регулярне опитування мережевих вузлів за допомогою ICMP та SNMP, фіксацію часу відгуку, втрат пакетів і отримання статистики інтерфейсів. Окремий модуль виконує періодичне ARP-сканування для актуалізації відповідності MAC–IP та виявлення нових або невідомих пристроїв. Дані UPS обробляються самостійним модулем, який аналізує журнали apcupsd або WinPower та формує записи з параметрами напруги, навантаження, рівня заряду і тривалості автономної роботи. Усі модулі записують результати до структурованих таблиць бази даних, де кожна група метрик зберігається окремо, що дозволяє оптимізувати роботу з великим обсягом історичних даних.

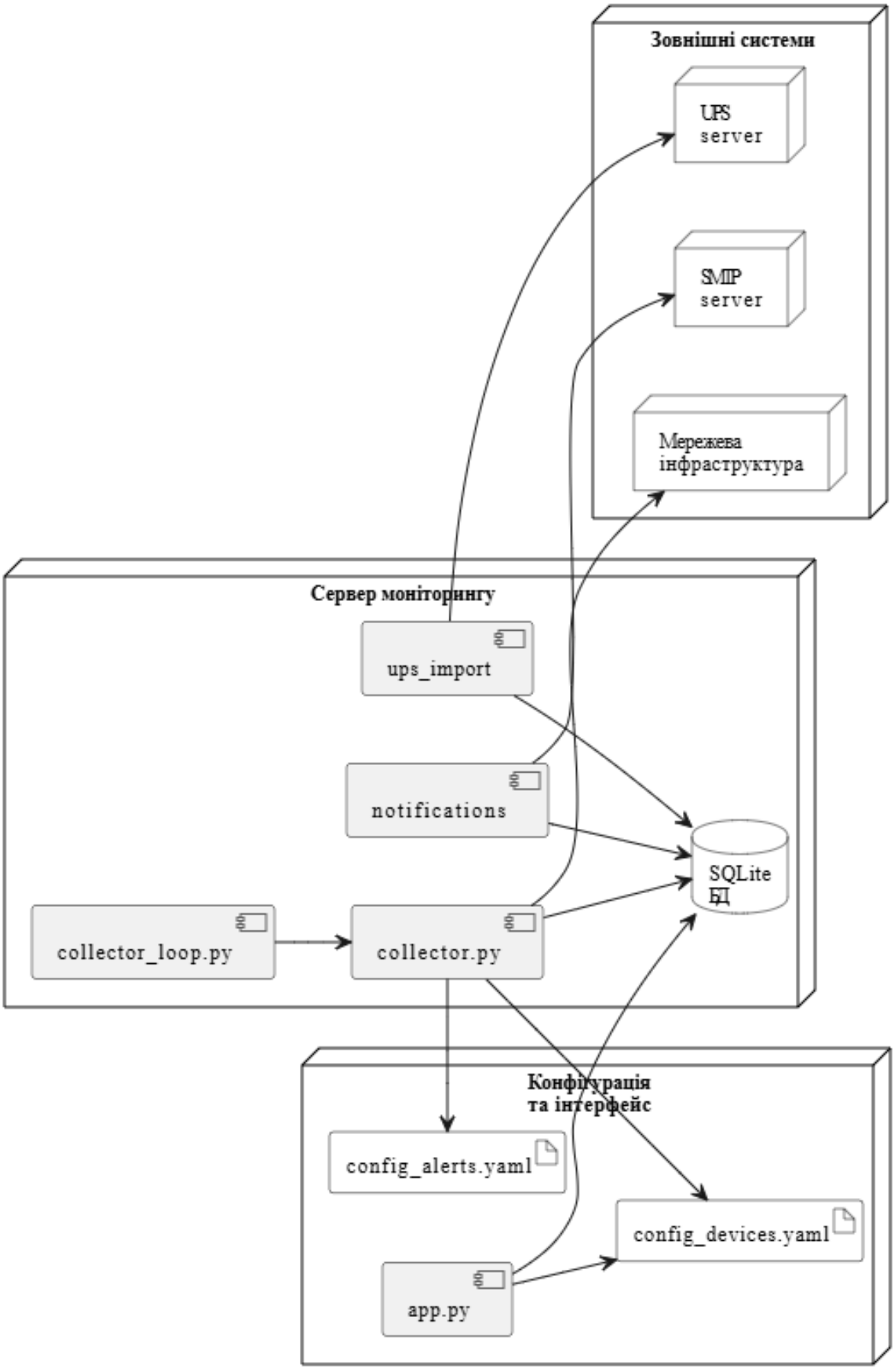


Рисунок 2.2 – Загальна архітектура системи моніторингу

Архітектура системи побудована таким чином, щоб кожен модуль працював автономно і не блокував роботу інших компонентів. Це забезпечує стійкість до помилок, можливість паралельної обробки запитів і поступове накопичення телеметрії. На рівні інтерпретації даних працює модуль обчислення інтегрального показника health_score, який формує часткові оцінки на основі ICMP-показників, SNMP-метрик, сервісних параметрів, UPS-даних та Wi-Fi-статистики. Розрахована оцінка зберігається у таблицях бази даних і використовується для автоматичного визначення стану вузлів (up, degraded, down, unknown). У разі переходу вузла між станами створюється запис у журналі інцидентів і формується запис про сповіщення.

Веб-інтерфейс системи, реалізований на Streamlit, отримує дані безпосередньо з бази SQLite і використовує їх для побудови таблиць, графіків і списків подій. Інтерфейс не взаємодіє з мережею напряму, а лише відображає актуальну інформацію, що мінімізує навантаження на інфраструктуру та виключає ризики впливу на мережеві пристрої. Така архітектура забезпечує ізоляцію логіки збору даних від логіки їх візуалізації, що підвищує надійність системи та спрощує підтримку у майбутньому.

З точки зору розгортання система складається з двох логічних частин: фонових модулів, що працюють як постійно активні служби операційної системи, та веб-інтерфейсу, який запускається за потреби. Усі компоненти розміщуються на одному сервері, що виконує роль центрального вузла моніторингу і взаємодіє з базою даних та мережевими пристроями, як це показано на рисунку 2.3. За необхідності така схема легко розширюється винесенням бази даних або веб-інтерфейсу на окремі вузли без зміни внутрішньої логіки, що забезпечує можливість масштабування системи в межах підприємства. Крім того, централізоване розміщення всіх компонентів спрощує резервне копіювання конфігураційних файлів та історичних даних, що є важливим чинником при забезпеченні безперервності моніторингу та швидкому відновленні системи після можливих збоїв.

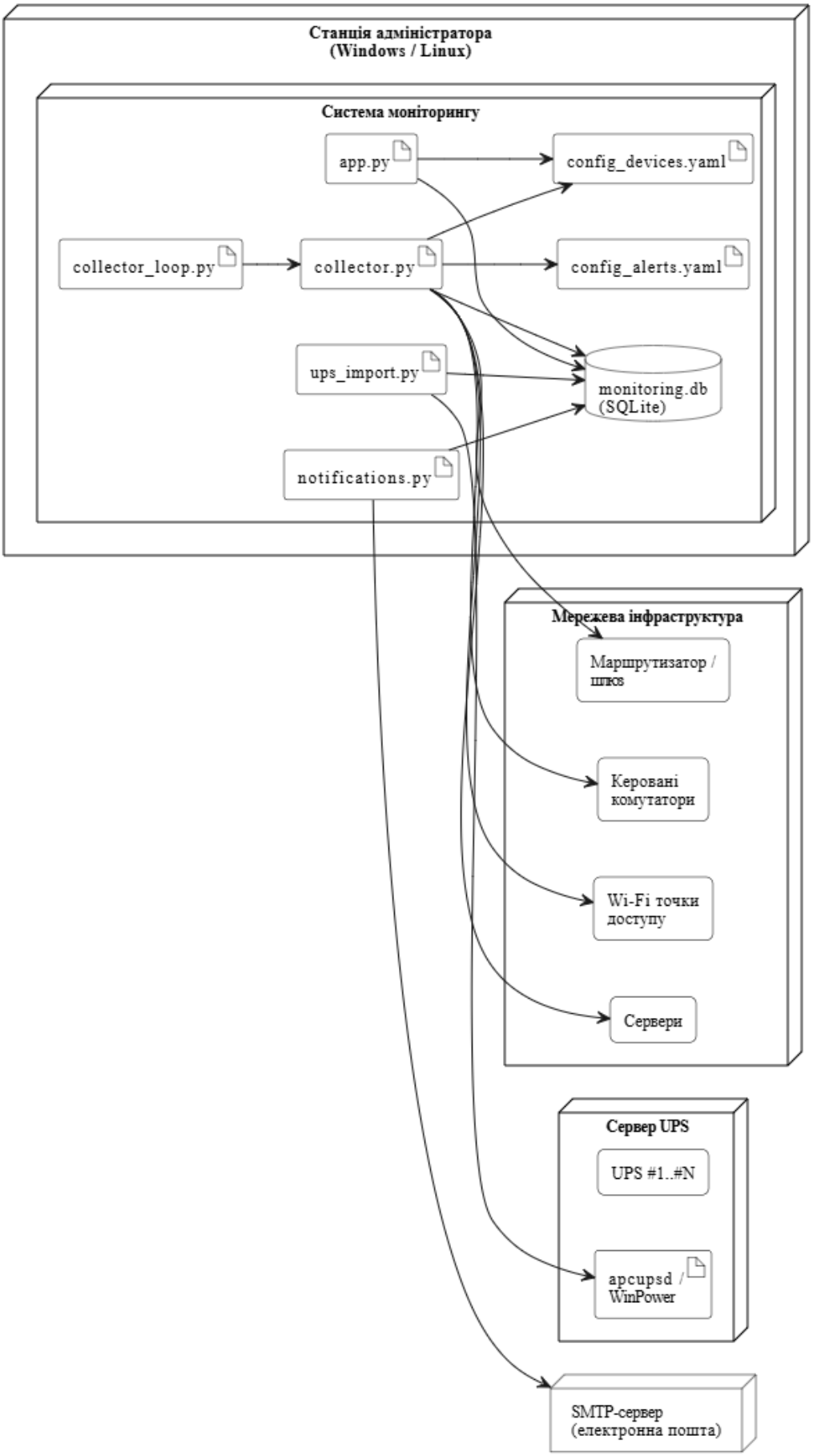


Рисунок 2.3 – Діаграма розгортання системи

### 2.4. Модель даних та проєктування бази даних системи

Модель даних розробленої системи побудована як розширена сутнісно-зв’язна структура, яка передбачає чітке розмежування довідкових сутностей, потоків телеметрії та журналів подій. Такий підхід забезпечує керованість зростанням обсягів даних, можливість паралельного запису з декількох модулів, стабільність роботи при високій частоті оновлень та прозору логіку подальшого масштабування. База даних реалізована на SQLite, що дозволяє поєднати простоту розгортання, відсутність зовнішніх залежностей і гарантовану цілісність даних без необхідності запуску окремого серверного процесу. Для систем моніторингу середнього масштабу це забезпечує оптимальний компроміс між швидкістю, ресурсоспоживанням і надійністю.

У структурі БД виділено декілька груп сутностей. Центральне місце займає таблиця devices, яка описує всі мережеві вузли, що підлягають моніторингу, включаючи маршрутизатори, комутатори, сервери, точки доступу та інші об’єкти. До цієї сутності прив’язано логічно залежні таблиці interfaces, services і wifi_radios, що дозволяє фіксувати структуру пристрою за портами, сервісами рівнів L4–L7 та радіомодулями Wi-Fi. Така модель забезпечує чітку відповідність між об’єктом у мережі та набором метрик, які до нього належать, дозволяючи зберігати інформацію в нормалізованому вигляді та уникати дублювання.

Другий блок складають таблиці, що містять телеметрію. Таблиця measurements використовується для зберігання результатів активних перевірок доступності за ICMP, включаючи затримку, варіацію затримки та втрати пакетів. Таблиця device_stats призначена для SNMP-показників пристрою, таких як завантаження процесора, використання пам’яті, аптайм та температурні значення. Телеметрія портів фіксується у таблиці interface_stats, де накопичуються значення RX/TX-трафіку, помилок, dropped-пакетів та інших характеристик L2-рівня. Моніторинг сервісів відображається у таблиці service_checks, де зберігаються результати HTTP/HTTPS-перевірок, час встановлення TCP-з’єднання, DNS-латентність і статуси відповідей. Для Wi-Fi

створено таблицю wifi_stats, яка фіксує кількість клієнтів, середній RSSI, кількість повторних передач та завантаження радіоканалу. Дані джерел безперебійного живлення зберігаються в таблиці ups_metrics, що містить напругу, навантаження, заряд батареї та час автономної роботи, що дозволяє аналізувати поведінку живлення під час збоїв електромережі.

Окремий логічний блок становлять таблиці подій та журналів, які фіксують динамічні зміни мережі та відхилення від нормальної роботи. Таблиця arp_events зберігає події, пов’язані зі змінами ARP-станів, що дозволяє виявляти появу нових або невідомих пристроїв, а також нестабільність записів ARP-кешу. Таблиця dhcp_leases_history використовується для документування видачі DHCP-оренд і прив’язується до сутності unknown_devices, у якій зберігаються MAC-адреси пристроїв, що не входять до переліку дозволених. Зміни стану портів фіксуються в таблиці link_state_changes, тоді як події маршрутизації зберігаються у таблиці routing_changes, що дозволяє відстежувати перепідключення, зміну next-hop або оновлення таблиць маршрутизації. Всі ці події забезпечують багаторівневий погляд на роботу мережі та дозволяють здійснювати аналіз причинно-наслідкових залежностей.

Підсистема інцидентів виділена окремо у таблицях incidents і notifications. У таблиці incidents фіксуються критичні відхилення: падіння доступності, деградація health_score, збої інтерфейсів, недоступність сервісів або перевищення температурних порогів. Таблиця notifications містить дані про сформовані електронні повідомлення, їх статус доставки та канал відправлення. Такий розподіл забезпечує відокремлене зберігання факту інциденту та інформації про сповіщення, що важливо для аудиту та аналізу історії реагування.

У проєктуванні моделі даних важливим аспектом стало рішення зберігати конфігураційні параметри пристроїв у YAML-файлі, а не у таблиці БД. Це обґрунтовується тим, що список керованих вузлів змінюється набагато рідше, ніж телеметрія, а його редагування зручніше виконувати у форматі людиночитного конфігураційного файлу. YAML забезпечує структуроване подання інформації про вузли, типи пристроїв, SNMP-параметри та налаштування логіки перевірок,

дозволяючи адміністратору оперативно вносити зміни без ризику пошкодження даних, що вже накопичені у БД. Такий підхід розділяє конфігураційну модель та експлуатаційні дані, зменшує навантаження на БД та спрощує резервне копіювання, оскільки конфігурація і історичні дані зберігаються окремо.

Сутнісно-зв’язна модель повністю узгоджується з архітектурою системи та забезпечує оптимальний баланс між нормалізацією, продуктивністю й стійкістю до зростання обсягів інформації. На рисунку 2.4 наведено структуру бази даних із відображенням основних сутностей та зв’язків типу «один-до-багатьох», що визначають логіку зберігання мережевих метрик, журналів подій і результатів аналізу інцидентів.

Важливим елементом проєктування стало визначення режиму роботи SQLite при обробленні великої кількості коротких транзакцій, характерних для систем моніторингу. У процесі експериментів встановлено, що оптимальним для системи є використання журнального режиму WAL, який забезпечує відокремлення потоків читання та запису й дозволяє модулям collector, service_checks та ups_import виконувати паралельні операції без тривалих блокувань та деградації часу відповіді БД при зростанні обсягу накопичених даних.

Проєктування моделі даних передбачало дотримання принципів нормалізації, оскільки зберігання телеметрії в агрегованому або дубльованому вигляді призвело б до швидкого зростання обсягу БД і перевитрат ресурсів при побудові графіків та виконанні аналітичних запитів. Саме тому сутності devices, interfaces, services, wifi_radios та ups_units винесені в окремий блок довідкових таблиць, тоді як дані вимірювань і подій прив’язуються до них через зовнішні ключі та мінімально необхідні індекси за device_id, interface_id, service_id і timestamp, що дає можливість коректно аналізувати показники L2–L3, L4–L7 та системні характеристики обладнання у часовому розрізі й підтримувати передбачувану продуктивність при нарощуванні кількості вузлів і тривалості зберігання історії.

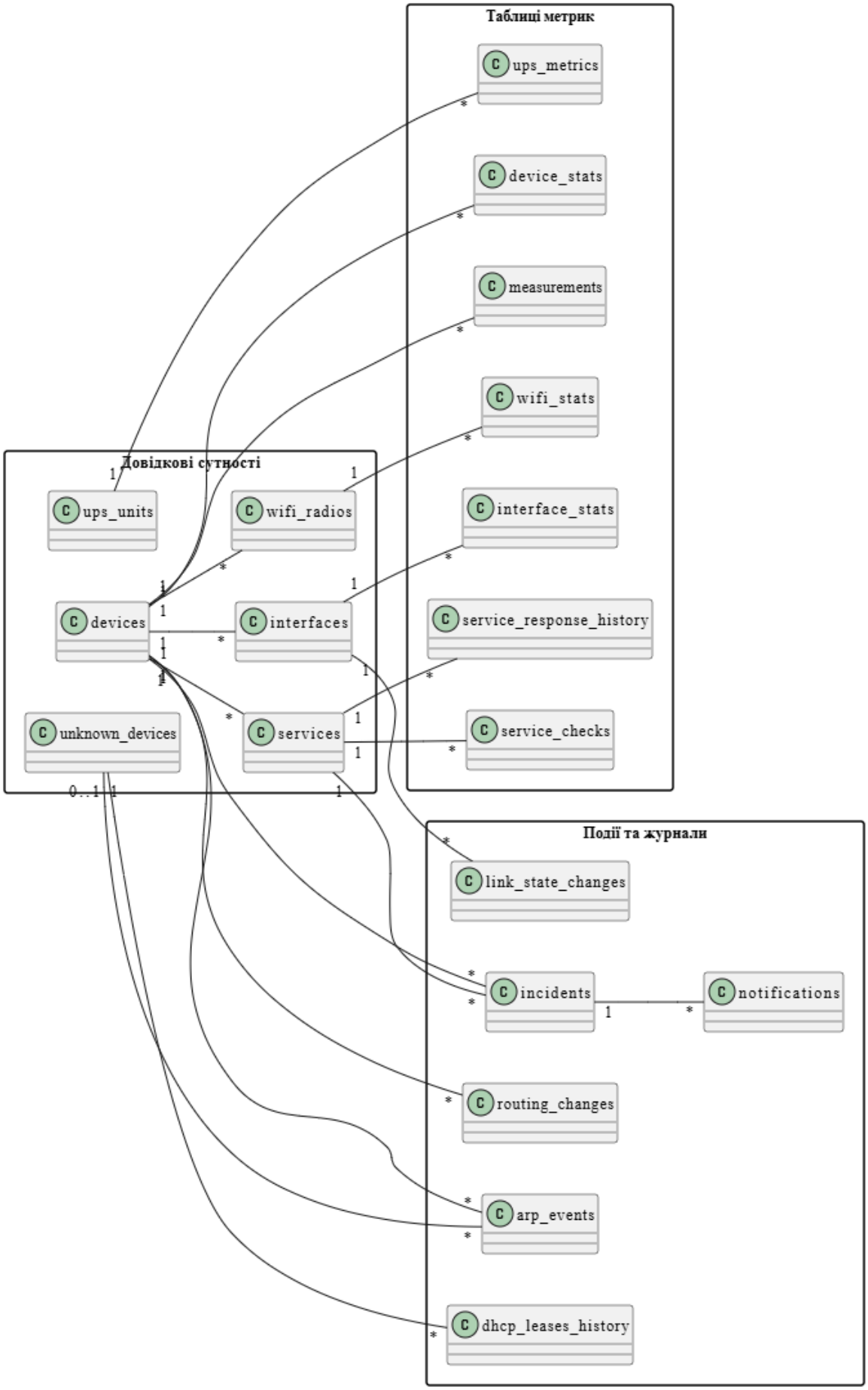


Рисунок 2.4 – Структура бази даних системи

## 2.5. Проєктування інтерфейсу користувача та сценаріїв роботи адміністратора

Проєктування інтерфейсу користувача для програмної системи моніторингу орієнтоване на забезпечення максимально наочного відображення стану мережевої інфраструктури при мінімальному когнітивному навантаженні на адміністратора. Веб-інтерфейс реалізовано засобами Streamlit, що дозволяє організувати єдину панель моніторингу, де в інтерактивному режимі відображаються таблиці вузлів, графіки телеметрії, журнали інцидентів, сповіщень, а також окремі розділи для аналізу невідомих пристроїв і показників джерел безперебійного живлення. Основним елементом головного екрану є таблиця вузлів, у якій для кожного пристрою виводяться ідентифікатор, тип, управлінська IP-адреса, поточний інтегральний показник health_score та агрегований стан у вигляді статусу up, degraded або down. Значення health_score використовується як центральний індикатор, що об’єднує ICMP-показники, SNMP-метрики, сервісні перевірки, Wi-Fi-статистику та UPS-дані, тому саме він визначає кольорове маркування рядків таблиці та дозволяє адміністраторам швидко виявляти проблемні вузли без додаткових запитів до бази даних.

Для поглибленого аналізу стану окремого вузла передбачено сценарій переходу до детальної сторінки, де відображаються часові графіки затримки ICMP, варіації затримки, втрат пакетів, завантаження процесора та пам’яті, температурних показників, а також статистики інтерфейсів за SNMP. На цьому ж екрані можуть відображатися графіки активності сервісів рівнів L4–L7, включаючи час встановлення TCP-з’єднання, DNS-латентність та історію HTTP-відповідей. Завдяки такому розділенню інформації адміністратор отримує можливість поєднувати загальний огляд стану мережі з детальним аналізом окремих вузлів, не залишаючи меж веб-інтерфейсу. Окремі вкладки інтерфейсу призначені для роботи з журналами інцидентів і сповіщень, де для кожної події відображаються час виникнення, тип інциденту, значення health_score на момент події, а також факт відправлення електронного повідомлення на вказану адресу.

Це дозволяє відстежувати повноту реагування системи та аналізувати хронологію розвитку інцидентів у прив’язці до мережевих подій.

Важливим елементом інтерфейсу є сторінка аналізу невідомих пристроїв, де на основі даних arp_events та dhcp_leases_history виводиться перелік MAC-адрес, які не відповідають зареєстрованим у конфігурації вузлам. Для кожного такого запису відображаються MAC-адреса, остання відома IP-адреса, VLAN, час першого та останнього виявлення, що дозволяє швидко ідентифікувати підозрілу активність на рівні L2–L3. Окремий розділ інтерфейсу присвячено джерелам безперебійного живлення: за даними таблиці ups_metrics виводяться поточні значення вхідної та вихідної напруги, навантаження, рівень заряду батареї, оцінка часу автономної роботи та історія переходів на живлення від акумулятора. Це дає змогу адміністраторам оцінити стійкість інфраструктури до збоїв електроживлення та співвіднести події мережевої деградації з конкретними епізодами роботи UPS. Для Wi-Fi-підсистеми передбачено окрему вкладку зі статистикою wifi_stats, де відображається кількість клієнтів, середній рівень RSSI, кількість повторних передач та завантаження радіоканалу, що особливо важливо для аналізу якості обслуговування мобільних користувачів.

Сценарії взаємодії адміністратора з веб-інтерфейсом системи включають регулярний огляд сумарного стану мережі, аналіз вузлів зі зниженим health_score, перегляд журналів інцидентів, перевірку коректності спрацювання сповіщень, виявлення та верифікацію невідомих пристроїв, а також аналіз поведінки UPS у моменти збоїв електроживлення. У разі потреби адміністратор може оновити конфігурацію вузлів шляхом редагування конфігураційного файлу та перезапуску фонових модулів, після чого веб-інтерфейс автоматично відобразить актуальний склад об’єктів моніторингу. Для узагальненого відображення цих сценаріїв розроблено діаграму варіантів використання, на якій показано основні дії адміністратора при роботі з веб-інтерфейсом системи, включаючи перегляд стану вузлів, аналіз телеметрії, роботу з інцидентами, невідомими пристроями та UPS-показниками. Діаграма наведена на рисунку 2.5 і відображає взаємозв’язок

між роллю адміністратора та функціональними можливостями розробленої системи з точки зору користувацького доступу до даних моніторингу.

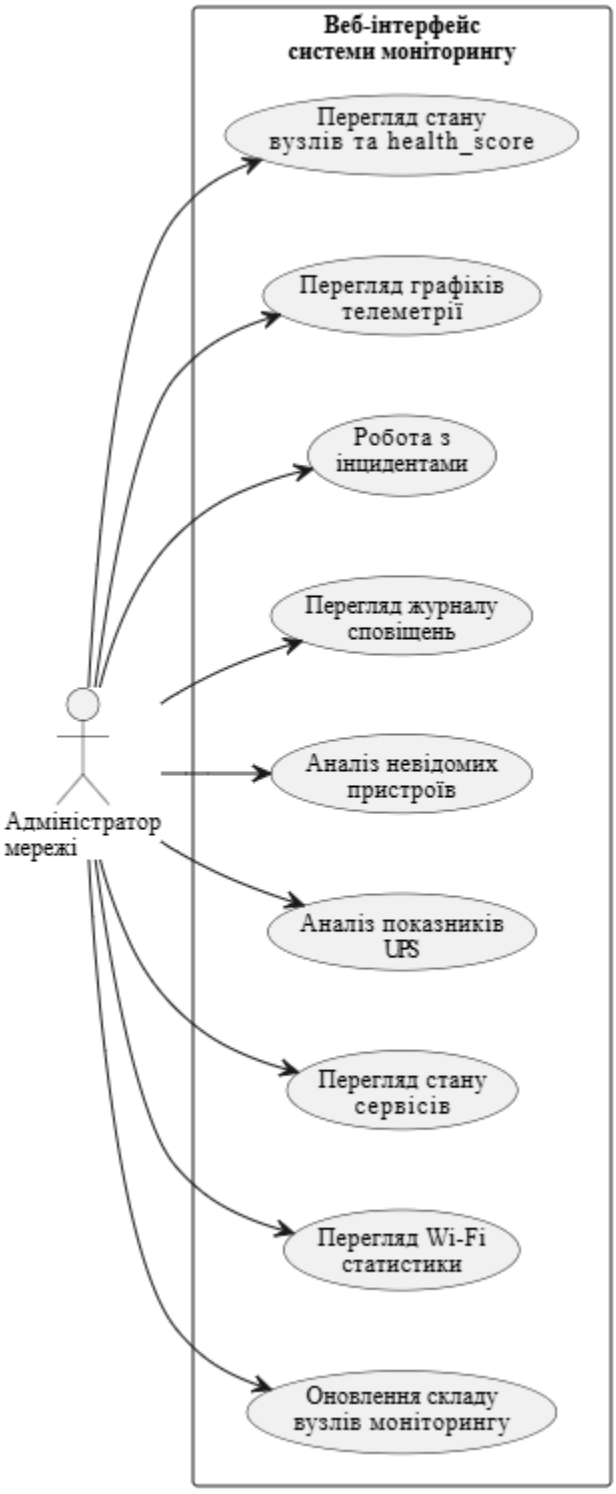


Рисунок 2.5 - Діаграма варіантів використання веб-інтерфейсу програмної системи

# РОЗДІЛ 3.
# РЕЗУЛЬТАТИ ПРОЄКТУВАННЯ ТА ДОСЛІДЖЕННЯ СИСТЕМИ МОНІТОРИНГУ МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ

## 3.1. Реалізація модуля збору даних та математична модель інтегрального показника health_score

Модуль збору даних є центральним компонентом розробленої системи, оскільки він забезпечує формування повного набору мережевих, сервісних, системних і енергетичних показників, що використовуються для оцінювання стану інфраструктури. Архітектура модуля побудована за принципом паралельного опитування вузлів, де окремі підмодулі відповідають за ICMP-вимірювання, SNMP-телеметрію, ARP-сканування, перевірку сервісів рівнів L4–L7, аналіз Wi-Fi активності та оброблення журналів джерел безперебійного живлення. Такий підхід забезпечує сталість інтервалів опитування й виключає взаємні блокування, що є критично важливим для коректності часових вимірів. Результатом роботи модуля стають записи в таблицях measurements, interface_stats, device_stats, service_checks, wifi_stats та ups_metrics, які зберігаються у структурованому вигляді та використовуються системою під час обчислення підсумкових показників.

Функціонування ICMP-підмодуля ґрунтується на регулярному вимірюванні часу відгуку мережевих вузлів, фіксації втрат пакетів та розрахунку варіації затримки, що дозволяє оцінити стабільність каналу передачі даних та визначати ранні прояви деградації мережі. Дані SNMP-опитування охоплюють показники завантаження процесора, використання оперативної пам’яті, системного uptime, температурних сенсорів, а також статистику мережевих інтерфейсів, включаючи RX/TX-помилки, CRC-помилки, dropped packets і параметри черг QoS. Сервісні перевірки охоплюють час встановлення TCP-з’єднання, DNS-латентність та коректність відповідей HTTP/HTTPS-сервісів.

Для бездротової підсистеми реєструються кількість клієнтів, рівні RSSI, кількість повторних передач та завантаження радіоканалу. Модуль UPS обробляє журнали apcupsd або WinPower, переводячи їх у структуровані записи про напругу, навантаження, рівень заряду та епізоди переходу на живлення від батареї.

Окреме значення має механізм пасивного аналізу, реалізований на основі ARP-сканування та оброблення ARP-таблиць. Він забезпечує визначення відповідності MAC–IP у реальному часі, виявлення нестабільних DHCP-оренд, ARP-flapping та появи невідомих пристроїв. Такі події фіксуються у таблицях arp_events і dhcp_leases_history, що дозволяє проводити аналіз на рівні L2 і формувати відповідні інциденти. Усі дані незалежно від типу телеметрії уніфіковано зберігаються у SQLite, що дозволяє формувати повну тимчасову картину стану кожного мережевого вузла.

Подальша обробка зібраних даних здійснюється у рамках математичної моделі інтегрального показника health_score, яка об’єднує різнорідні метрики у єдиний нормалізований числовий діапазон. Оскільки ICMP-, SNMP-, сервісні, енергетичні та бездротові показники мають різну природу та одиниці вимірювання, першим етапом обчислення є нормалізація первинних значень до інтервалу від 0 до 1. Для ICMP-параметрів нормалізація враховує затримку, jitter та packet loss, де значення, близькі до оптимальних, відображаються як величини, наближені до 1, тоді як параметри, що перевищують критичні межі, знижуються до значень, близьких до нуля. SNMP-показники нормалізуються з урахуванням допустимого діапазону роботи процесора, пам’яті та температурних датчиків, а також статистики інтерфейсів, де зростання помилок або дропів знижує значення snmp_score. Для сервісних параметрів нормалізатор враховує часові характеристики TCP та DNS і коректність HTTP-відповідей, перетворюючи їх у безрозмірну шкалу. UPS-показники нормалізуються відповідно до стану батареї, напруги та кількості переходів на автономне живлення, а Wi-Fi-параметри формують wifi_score залежно від рівня сигналу, наявності повторних передач і завантаження радіоефіру.

Після нормалізації часткові показники об’єднуються у підсумковий індикатор за формулою 3.1:

$$health\ score = w1 \cdot icmp\_score + w2 \cdot snmp\_score + w3 \cdot service\_score + w4 \cdot ups\_score + w5 \cdot wifi\_score \quad (3.1)$$

Вагові коефіцієнти w1–w5 підібрані таким чином, щоб основний внесок у health_score формували ICMP- та SNMP-показники як найбільш стабільні й інформативні у контексті базової працездатності L2–L3 інфраструктури. При цьому для їх вибору застосовано емпіричний підхід, характерний для практики побудови систем моніторингу, коли значення визначаються на основі експериментальних спостережень за поведінкою мережевих метрик та їх впливом на загальний стан інфраструктури. Сервісні та Wi-Fi параметри відіграють уточнювальну роль, а значення UPS-модуля роблять модель чутливою до збоїв електроживлення, які є критичними для серверних вузлів та активного мережевого обладнання. Завдяки такій побудові інтегральний показник залишається стійким до короткочасних флуктуацій, але водночас швидко реагує на стійкі зміни стану мережі.

Система класифікує стан кожного вузла на основі значення health_score. Вузли зі стабільними параметрами отримують статус up, тоді як зниження значення інтегрального показника внаслідок зростання затримок, появи помилок SNMP або деградації сервісів призводить до переходу у стан degraded. Повна недоступність або критичні відхилення переводять вузол у стан down. Стан unknown застосовується до вузлів, для яких у системі відсутні актуальні дані. Усі розраховані значення health_score фіксуються у базі даних, що дозволяє аналізувати динаміку роботи інфраструктури та виконувати коректне формування інцидентів.

Узгоджена робота модуля збору даних та математичної моделі health_score формує цілісний механізм оцінювання стану мережевої інфраструктури, що забезпечує надійність і передбачуваність роботи всієї системи моніторингу.

### 3.2. Реалізація системи сповіщень про інциденти

Реалізація системи сповіщень про інциденти ґрунтується на подієво-орієнтованому підході, у якому кожна зміна стану мережевого вузла, виявлена модулем збору даних, призводить до формування відповідного запису в таблиці incidents та, за потреби, ініціює процедуру надсилання електронного сповіщення. Логіка побудована таким чином, щоб забезпечити надійну фіксацію переходів між станами up, degraded, down та unknown, використовуючи інтегральний показник health_score як основний критерій. Модуль collector після кожного циклу опитування виконує порівняння нового значення health_score з попереднім записом у базі даних. Якщо система виявляє, що стан змінився, створюється подія інциденту із зазначенням часу, типу переходу, нового значення health_score та відповідного ідентифікатора вузла. Усі інциденти зберігаються у таблиці incidents, що забезпечує можливість подальшого аналізу часових закономірностей деградацій та відмов у мережі.

Після реєстрації інциденту активується механізм формування сповіщення. Модуль notifications перевіряє, чи належить інцидент до категорії подій, що потребують негайного інформування адміністратора, та чи не було нещодавно відправлено аналогічне повідомлення, що дозволяє уникати дублювання електронних листів при нестабільності вузла. Якщо умови виконуються, система створює відповідний запис у таблиці notifications, що включає час створення, тип події, статус відправлення та текст повідомлення. Після цього модуль SMTP-відправлення формує електронний лист та надсилає його на вказану адресу, оновлюючи статус запису в таблиці notifications залежно від успішності операції. Такий підхід дає змогу відокремити логіку формування інцидентів від логіки передачі повідомлень, що підвищує надійність системи та спрощує її подальшу модернізацію.

Послідовність формування інциденту та надсилання сповіщення подано на рисунку 3.1. На діаграмі відображено взаємодію основних компонентів системи, включаючи модуль collector, модуль incident manager, підсистему notifications,

SMTP-модуль та базу даних. Рисунок демонструє повний цикл події, починаючи від зчитування телеметрії та обчислення health_score і завершуючи фіксацією інциденту та доставкою електронного сповіщення. Завдяки цьому діаграма дозволяє побачити динамічну структуру модулів та їхню узгоджену роботу при обробленні подій.

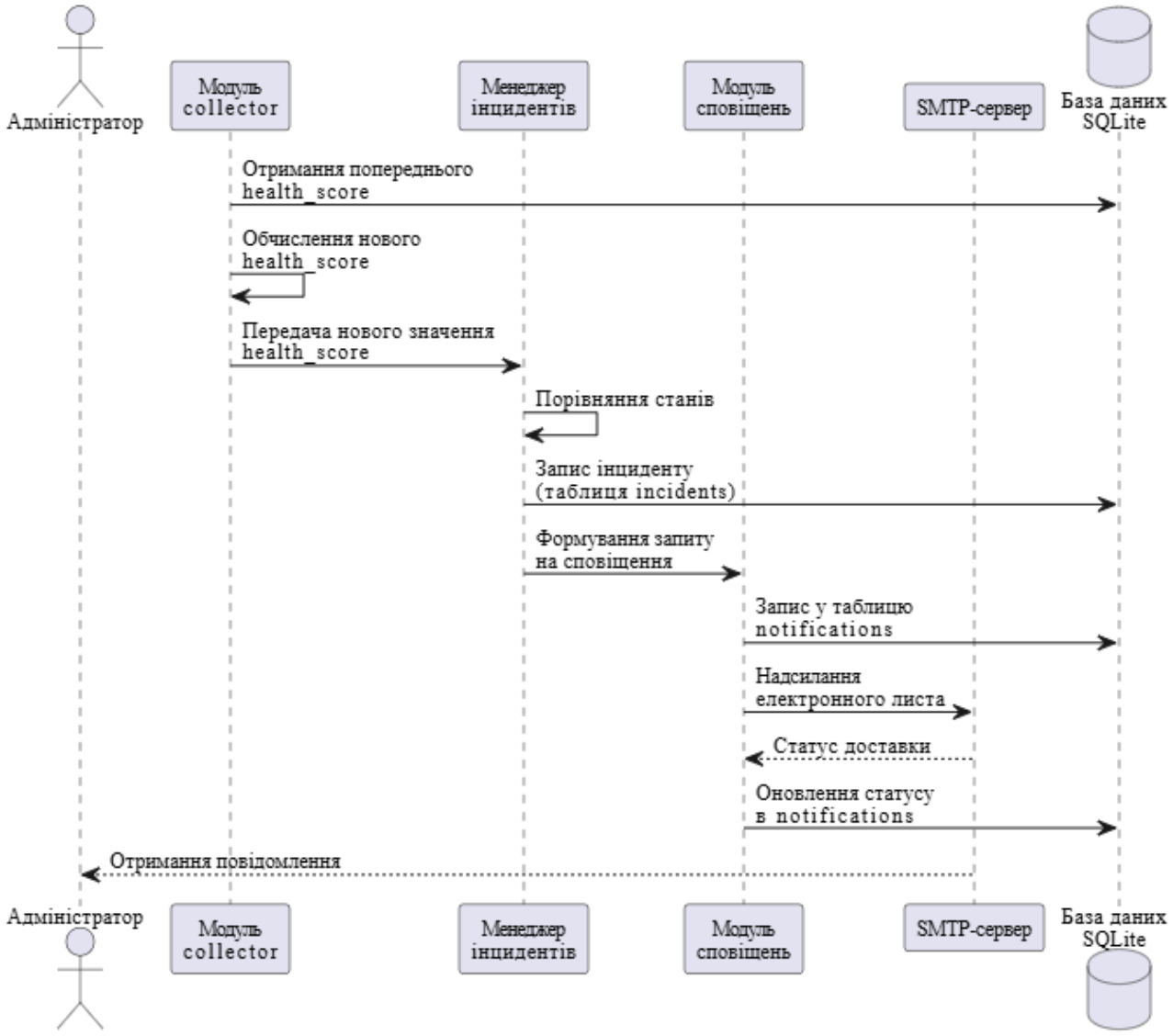


Рисунок 3.1 - Послідовність формування інциденту та надсилання сповіщення в системі.

### 3.3. Реалізація веб-інтерфейсу для адміністратора мережі

Веб-інтерфейс розробленої системи побудований на основі фреймворку Streamlit, який забезпечує формування інтерактивних таблиць і графічних елементів шляхом прямої взаємодії з базою даних без потреби у додаткових REST-запитах до мережевих пристроїв. Така архітектура гарантує ізоляцію інтерфейсу від механізмів збору даних, завдяки чому оновлення графіків та таблиць не впливає на цикли опитування вузлів і не створює додаткового навантаження на інфраструктуру.

Особливістю використаного фреймворку Streamlit є реактивна модель побудови інтерфейсу, у якій кожна зміна даних призводить до автоматичного перерендерингу сторінки[26]. Такий підхід дозволяє відмовитися від класичної архітектури «клієнт–сервер» з окремим фронтендом і мінімізує складність розробки, оскільки усі елементи взаємодії - таблиці, графіки, фільтри та навігаційні компоненти - формуються безпосередньо в Python-коді. Streamlit виконує оброблення інтерфейсу у вигляді статичної сесії, що створюється для кожного користувацького сеансу, а зміна стану сторінки тригериться через механізм повторного виконання сценарію (rerun). За рахунок цього інтерфейс отримує властивості застосунків, побудованих на принципах декларативного рендерингу, що забезпечує плавне оновлення віджетів, узгоджену реакцію на зміну даних та відсутність потреби у додаткових API.

Важливим аспектом реалізації є застосування механізмів кешування Streamlit, які використовуються для оптимізації роботи з історичними даними у базі SQLite. Частина запитів, що формують таблиці та графіки, кешується у межах користувацької сесії, що зменшує кількість звернень до сховища та прискорює відображення сторінок при переходах між вкладками. Для візуалізації часових рядів використовуються вбудовані інструменти matplotlib або plotly, які Streamlit інтегрує без додаткового конфігурування, забезпечуючи побудову графіків у реальному часі з можливістю масштабування та оновлення. Така архітектура дозволяє отримати швидкодіючий веб-інтерфейс без необхідності

використовувати окремі вебсервери, брокери подій чи системи побудови SPA-застосунків, що робить фреймворк особливо придатним для задач оперативного моніторингу мережевої інфраструктури.

Головна сторінка інтерфейсу відображає агрегований стан усіх вузлів, включаючи їхній поточний health_score, статус доступності, управлінську IP-адресу та тип обладнання. Інтерактивна таблиця, що представлена на рисунку 3.2, використовує кольорову індикацію для швидкої ідентифікації станів up, degraded та down, що дозволяє адміністратору оперативно оцінювати загальну стабільність мережі.

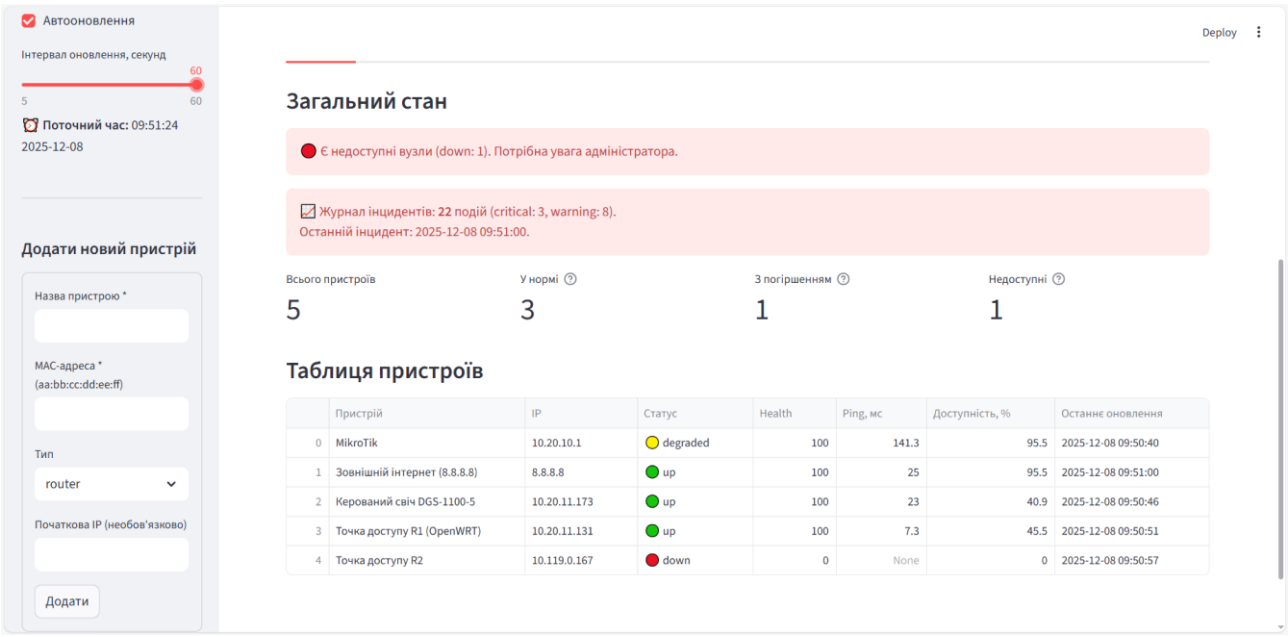


Рисунок 3.2 – Головна панель моніторингу системи

Детальна інформація щодо окремого мережевого вузла відкривається у вигляді окремої сторінки, де в інтерактивному форматі відображаються графіки зміни ICMP-затримки, варіації затримки, рівня втрат пакетів та інтегрального показника health_score, а також за потреби SNMP- та сервісних показників. Побудова часових графіків здійснюється безпосередньо на основі записів у таблицях measurements, device_stats та service_response_history, що дає змогу оцінювати як короткочасні відхилення, так і довгострокові тренди. Типовий вигляд такого екрану наведено на рисунку 3.3, де візуалізовано динаміку latency, packet loss та інтегрального показника стану у часовому розрізі.

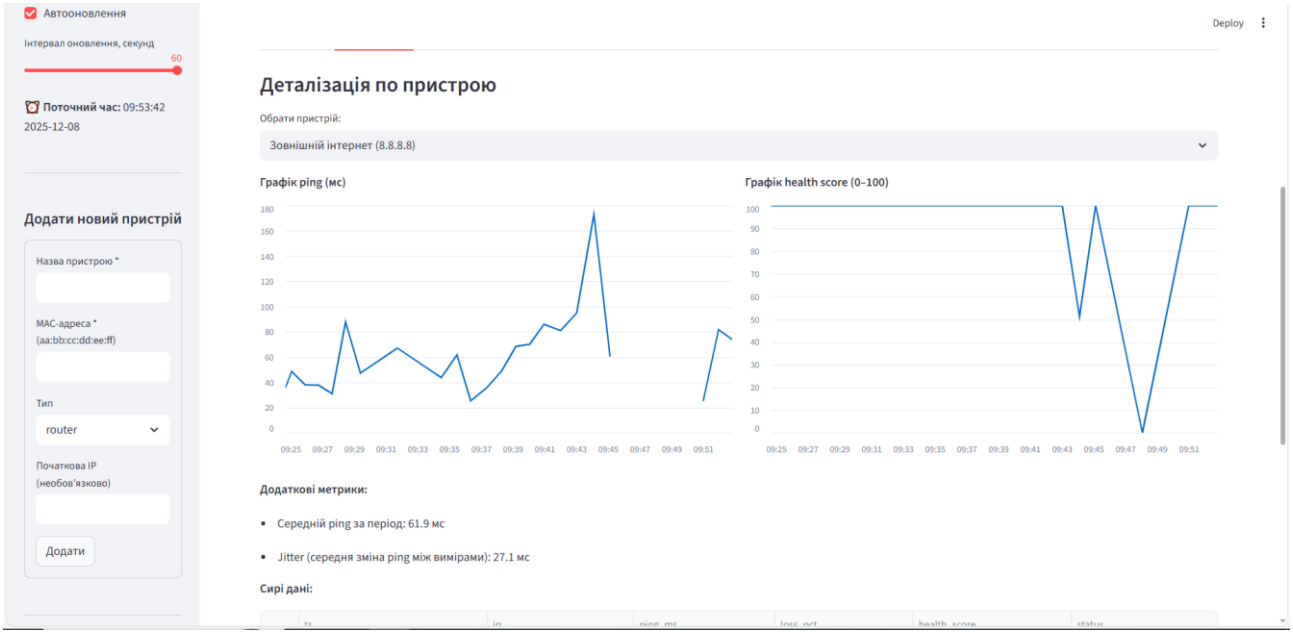


Рисунок 3.3 - Графіки динаміки часу відгуку та показника health_score

У веб-інтерфейсі реалізовано об’єднану сторінку журналу інцидентів та журналу сповіщень, що дозволяє аналізувати не лише факт деградації вузла, а й відповідні дії системи щодо інформування адміністратора. У верхній частині відображаються інциденти з таблиці incidents із зазначенням часу події, типу переходу, попереднього та нового значення health_score та ідентифікатора вузла. У нижній частині розташовано таблицю сповіщень, сформованих модулем notifications, де наведено час, текст повідомлення, адресу отримувача та статус доставки через SMTP. Такий формат подання дозволяє простежити причинно-наслідковий зв’язок між подією та фактом її інформування. Вигляд комбінованої сторінки журналу подій показано на рисунку 3.4.

Окрема вкладка інтерфейсу присвячена аналізу невідомих пристроїв, які були виявлені на основі даних ARP-сканування та журналів DHCP-оренд. Система формує список MAC-адрес, що не збігаються з конфігураційним файлом inventory, та виводить їх із зазначенням останньої IP-адреси, VLAN, часу першої та останньої появи. Такий функціонал дозволяє адміністратору оперативно реагувати на появу неавторизованих пристроїв у сегменті L2–L3. Приклад відповідного екрану наведено на рисунку 3.5.

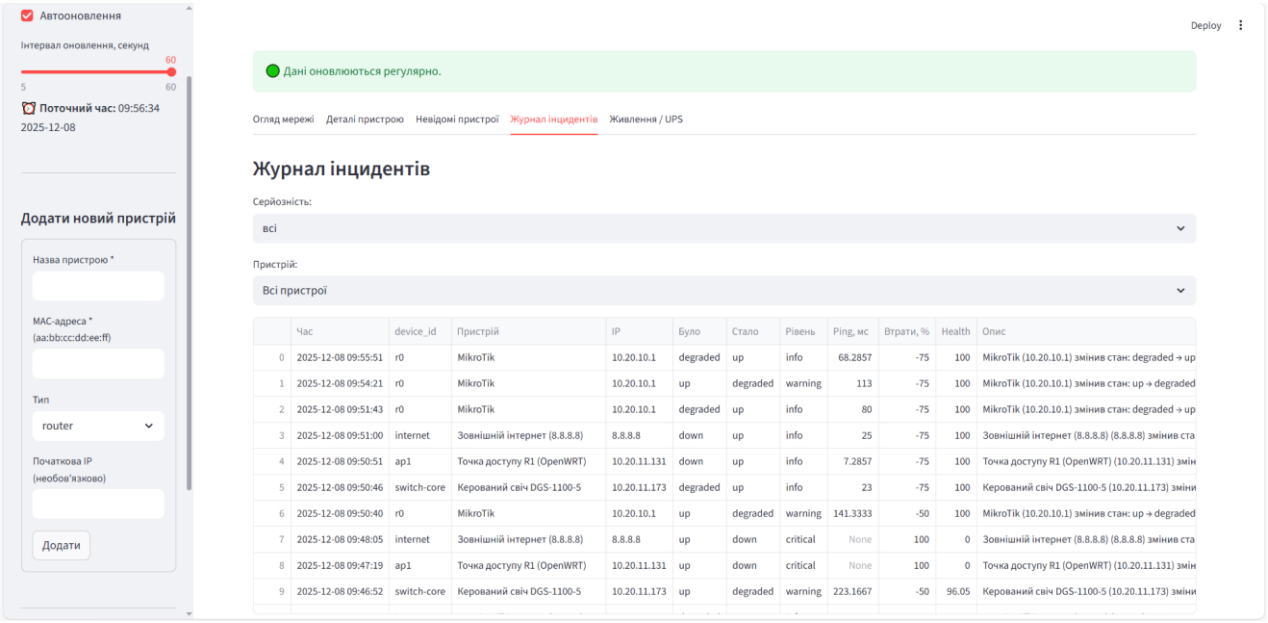


Рисунок 3.4 - Журнал інцидентів та журнал сповіщень

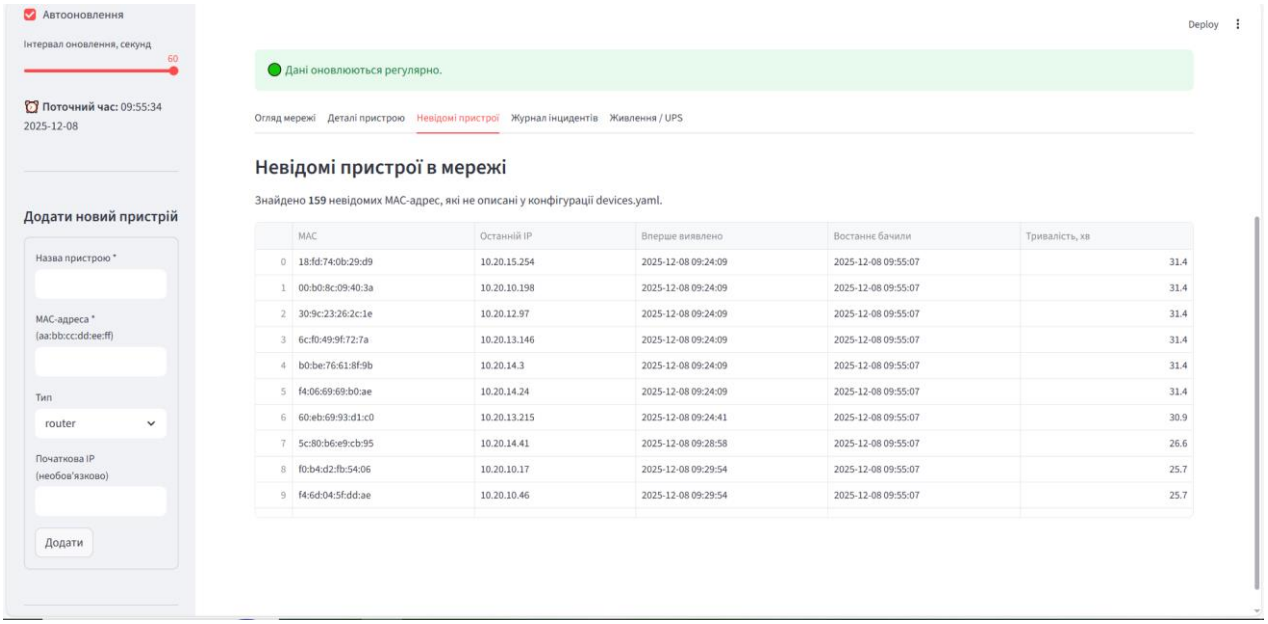


Рисунок 3.5 – Вкладка «Невідомі пристрої»

Додатковий розділ веб-інтерфейсу забезпечує візуалізацію роботи джерел безперебійного живлення на основі таблиці ups_metrics. Система будує графіки вхідної та вихідної напруги, навантаження, рівня заряду батареї та оцінки часу автономної роботи. Це дозволяє співставляти зміни електроживлення з моментами деградацій вузлів та виконувати аналіз стійкості інфраструктури до

зовнішніх енергетичних впливів. На рисунку 3.6 наведено типовий приклад відображення показників UPS у веб-інтерфейсі.

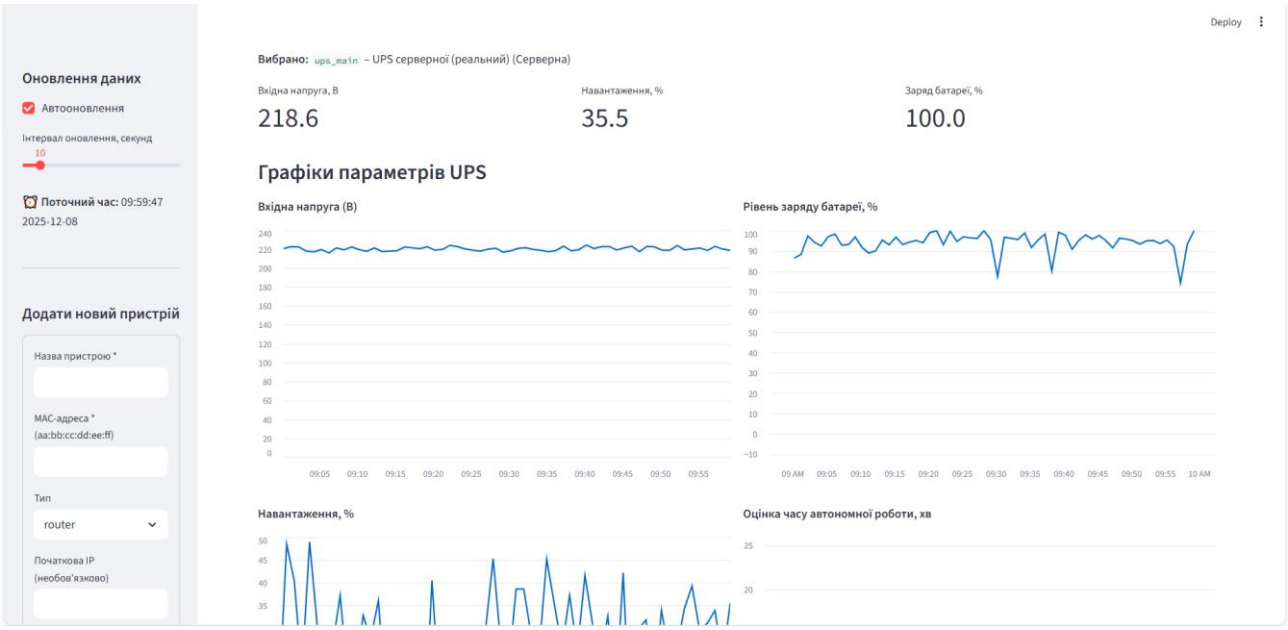


Рисунок 3.6 - Відображення параметрів джерел безперебійного живлення

Таким чином, веб-інтерфейс забезпечує комплексну візуалізацію стану мережевої інфраструктури, поєднуючи агреговані дані з детальними часовими характеристиками, журналами подій та допоміжними діагностичними панелями. Інтерактивність, структурованість і наочність інтерфейсу створюють умови для швидкого прийняття рішень адміністратором та підвищують ефективність експлуатації всієї системи моніторингу.

### 3.4. Інтеграція з джерелами безперебійного живлення та обробка логів apcupsd і WinPower

Інтеграція з джерелами безперебійного живлення у межах розробленої системи реалізована на основі аналізу текстових журналів програм apcupsd та WinPower, які виступають єдиним достовірним джерелом інформації про реальний стан живлення й поведінку UPS у моменти штатної роботи та аварійних режимів. Такий підхід дозволяє уникнути додаткового навантаження на мережу, оскільки система не опитує джерела живлення по SNMP або іншим протоколам

у реальному часі, а періодично зчитує сформовані лог-файли, перетворюючи їх на структуровані записи у таблиці ups_metrics. Спеціалізований модуль, що відповідає за інтеграцію з UPS, виконує регулярне читання журналів, парсинг часових відміток, напруги на вході та виході, рівня навантаження, стану батареї, оціненого часу автономної роботи та кількості переходів у режим живлення від акумулятора. Усі ці дані зберігаються у базі даних з прив’язкою до сутності ups_units, яка описує логічні джерела живлення, їх розташування у інфраструктурі та зв’язок з конкретними мережевими пристроями.

Обробка логів apcupsd і WinPower побудована таким чином, щоб уніфікувати різні формати представлення інформації про події живлення і привести їх до єдиної внутрішньої моделі. На етапі парсингу модуль відокремлює записи, що відповідають зміні режимів роботи UPS, зокрема подіям переходу на акумулятор, повернення до роботи від мережі, досягненню критичного рівня заряду батареї, перевантаженню та відновленню номінальної напруги. Для кожної події формується один або декілька записів у таблиці ups_metrics з фіксацією значень напруги, навантаження, рівня заряду та оціненого часу автономної роботи на момент події. Таким чином система одержує часовий ряд UPS-показників, який згодом використовується як для побудови графіків у веб-інтерфейсі, так і для розрахунку часткового показника ups_score у складі інтегрального health_score. У разі виявлення тривалих періодів роботи від батареї або повторюваних епізодів провалів напруги відповідні вузли інфраструктури можуть отримувати деградований стан навіть за формально коректної відповіді по ICMP.

З метою підвищення інформативності аналізу було змодельовано декілька джерел безперебійного живлення на основі реальних логів, отриманих з існуючих апаратних UPS. Для кожного такого джерела створено окремий запис у таблиці ups_units та виконано імпорт історичних журналів у структуру ups_metrics. Це дало змогу у веб-інтерфейсі відобразити одночасно декілька UPS, порівнювати їхню поведінку у різних режимах навантаження, аналізувати епізоди переходу на живлення від батареї та вплив цих подій на доступність і

стан мережевих вузлів. На рівні логіки оброблення інцидентів модуль моніторингу може співставляти часові мітки падінь вузлів або деградації health_score з моментами зникнення напруги на відповідних UPS, що дозволяє відрізняти відмови, спричинені проблемами живлення, від мережевих або сервісних збоїв.

Обробка логів апаратних джерел живлення інтегрована у загальний механізм формування інцидентів. Якщо для UPS, до якого прив’язано критично важливі мережеві пристрої, фіксується тривалий епізод роботи від батареї або досягнення критичного рівня заряду, модуль аналізу подій може створювати окремі інциденти з типом «ризик втрати живлення», які відображаються у загальному журналі та відстежуються разом з іншими подіями. Паралельно з цим ups_score знижується, що призводить до зменшення інтегрального health_score для залежних вузлів і, за потреби, переводить їх у деградований стан. У випадку відсутності нових записів у журналах apcupsd або WinPower протягом тривалого часу система може вважати дані про UPS неактуальними, що виражається в обнуленні або зменшенні ваги UPS-компоненти при розрахунку health_score. Завдяки цьому підсистема інтеграції з джерелами безперебійного живлення не лише відображає історію енергетичних подій, а й безпосередньо впливає на оцінювання стану мережевої інфраструктури, забезпечуючи більш адекватну реакцію системи на комплексні збої, пов’язані з блекаутами та нестабільністю електроживлення.

### 3.5. Методика експериментальної оцінки роботи системи

Методика експериментальної оцінки роботи розробленої системи ґрунтується на відтворенні типових для мережевої інфраструктури підприємства ситуацій, що призводять до зміни станів вузлів, деградації сервісів, появи невідомих пристроїв або втрати живлення, та подальшому вимірюванні часових і якісних характеристик реакції системи. Метою експериментів є перевірка коректності реалізованих алгоритмів збору даних, обчислення інтегрального

показника health_score, формування інцидентів і сповіщень, а також оцінювання здатності системи своєчасно та повно відображати реальні події у мережі без надмірної кількості помилкових спрацювань. Для цього формується набір контрольованих сценаріїв, що охоплюють як мережеві відмови, так і збої сервісів, події, пов’язані з джерелами безперебійного живлення, та появу невідомих пристроїв. Узагальнений перелік сценаріїв експериментальної оцінки наведено в таблиці 3.1.

Таблиця 3.1 - Сценарії експериментальної оцінки роботи системи

| **№ сценарію** | **Найменування сценарію** | **Суть експерименту** | **Основні показники оцінювання** |
|---|---|---|---|
| 1 | Відмова мережевого вузла | Штатне або штучне відключення комутатора, маршрутизатора чи сервера від мережі | Час виявлення відмови, зміна health_score, формування інциденту, наявність сповіщення |
| 2 | Деградація каналу зв’язку | Імітація підвищеної затримки, jitter та втрат пакетів для окремого вузла | Динаміка ICMP-показників, перехід вузла у стан degraded, наявність чи відсутність інциденту |
| 3 | Порушення роботи сервісу рівнів L4–L7 | Тимчасове припинення відповіді HTTP/HTTPS або затримка встановлення TCP-з’єднання | Зміна service_score, вплив на health_score, формування сервісного інциденту |
| 4 | Поява невідомого пристрою у сегменті мережі | Підключення вузла з MAC-адресою, відсутньою у конфігурації, до користувацького VLAN | Час від появи MAC до відображення у unknown_devices, формування відповідної події або інциденту |
| 5 | Перехід UPS у режим роботи від батареї | Ініціювання відключення електроживлення для сегмента, підключеного до джерела безперебійного живлення | Зміна показників ups_metrics, зниження ups_score, вплив на health_score залежних вузлів |
| 6 | Тривала робота при низькому заряді батареї UPS | Імітація ситуації, коли UPS тривалий час працює при критично низькому рівні заряду | Формування інцидентів «ризик втрати живлення», частота сповіщень, поведінка health_score у часі |
| 7 | Стабільний режим без аварій | Робота інфраструктури без навмисних втручань протягом контрольного інтервалу | Відсутність хибних інцидентів, стабільність health_score, відсутність надлишкових сповіщень |

Для кожного з наведених сценаріїв фіксується момент ініціювання події, після чого аналізується поведінка системи за записами у таблицях measurements, device_stats, service_checks, wifi_stats, ups_metrics, arp_events, incidents та notifications. Це дає змогу визначити реальний час виявлення події, затримку між моментом відхилення параметрів та формуванням інциденту, а також час надсилання електронного сповіщення. Окремо розглядаються випадки, коли зміни телеметрії призводять до помітної деградації часткових показників icmp_score, snmp_score, service_score чи ups_score, але не перетинають порогових значень для переходу health_score у гірший стан, що дозволяє оцінити чутливість та коректність вибраних порогів класифікації up, degraded і down.

Якісні характеристики роботи системи оцінюються через повноту виявлення інцидентів і рівень помилкових або надлишкових сповіщень. Повнота виявлення визначається як частка експериментально ініційованих подій із таблиці 3.1, для яких система сформувала коректні інциденти та зареєструвала їх у журналі. Рівень помилкових сповіщень оцінюється на основі сценарію стабільної роботи без аварій, коли аналізується поведінка health_score та журналів incidents і notifications у період відсутності реальних збоїв; у цьому режимі система не повинна генерувати інциденти, пов’язані лише з короткочасними флуктуаціями телеметрії. Додатково аналізується кореляція між подіями, зафіксованими у ups_metrics та arp_events, і відповідними інцидентами, що дозволяє оцінити правильність інтерпретації енергетичних і L2-подій.

Сукупність результатів за всіма сценаріями дозволяє кількісно оцінити час реакції системи на різні типи подій, стійкість інтегрального показника health_score до незначних флуктуацій мережевих параметрів, здатність коректно виявляти деградації та відмови, пов’язані з UPS, а також ефективність механізмів виявлення невідомих пристроїв. Отримані дані використовуються у наступному підрозділі для порівняльного аналізу роботи розробленої системи з існуючими рішеннями моніторингу мережевої інфраструктури та для обґрунтування доцільності її використання в умовах реального підприємства.

### 3.6. Результати експериментів та порівняння з існуючими системами моніторингу

Результати експериментальної оцінки роботи розробленої програмної системи показали, що закладені принципи побудови модулів збору даних, розрахунку інтегрального показника health_score та формування інцидентів забезпечують коректне відображення більшості характерних подій, які виникають у мережевій інфраструктурі підприємства. Для сценаріїв повної відмови вузлів час виявлення події системою практично збігається з інтервалом опитування, а перехід стану у down супроводжується формуванням інциденту та надсиланням електронного сповіщення, причому значення health_score наближається до нуля. У випадках деградації каналу зв'язку без повної втрати доступності система фіксує суттєві зміни ICMP-показників та відповідне зниження icmp_score, що призводить до переходу вузла у стан degraded без надмірних повторних сповіщень. Для подій, пов'язаних із UPS, за результатами аналізу журналів apcupsd і WinPower спостерігається чітка кореляція між епізодами переходу на живлення від батареї та зміненням ups_score, що відображається у поведінці інтегрального показника стану обладнання, прив'язаного до конкретного джерела безперебійного живлення.

У сценаріях появи невідомих пристроїв система демонструє коректну роботу механізму пасивного аналізу ARP-таблиць: часовий інтервал між першою появою MAC-адреси у сегменті L2 та відображенням відповідного запису в інтерфейсі «Невідомі пристрої» визначається періодичністю ARP-сканування і виявився прийнятним для умов локальної мережі підприємства. При цьому багатотаблична структура бази даних, що включає arp_events, dhcp_leases_history та unknown_devices, дозволяє відбудовувати історію появи таких вузлів і пов'язувати їх із конкретними VLAN. Оцінювання повноти виявлення інцидентів показало, що система фіксує усі змодельовані критичні події, а кількість хибних сповіщень у стабільному режимі роботи мережі залишається низькою, що свідчить про достатню стійкість обраних порогових значень health_score до

короткочасних флуктуацій телеметрії. Сукупність отриманих результатів дозволяє вважати запропоновану модель інтегрального показника придатною для експлуатації у реальних умовах невеликої або середньої за розміром інфраструктури.

Для оцінювання місця розробленої системи у контексті сучасних рішень моніторингу було виконано порівняння її характеристик із популярними системами управління мережевою інфраструктурою, такими як Zabbix, Nagios та PRTG. Порівняння проводилося за низкою критеріїв, що відображають вимоги до інфраструктури, підхід до ідентифікації вузлів, підтримку різних груп метрик та засоби візуалізації і сповіщення. Узагальнені результати наведено в таблиці 3.2, де розроблена система позиціонується як спеціалізоване рішення, орієнтоване на мережу підприємства з акцентом на ідентифікацію за MAC-адресою, моніторинг UPS та інтегральну оцінку стану.

Таблиця 3.2 - Порівняння розробленої системи з популярними NMS

| **Критерій** | **Розроблена система** | **Zabbix** | **Nagios** | **PRTG** |
|---|---|---|---|---|
| Спосіб розгортання | Один вузол, SQLite, веб-інтерфейс на Streamlit | Центральний сервер, агент/агентless-модель | Центральний сервер, конфігурація через файли | Центральний сервер з власним веб-інтерфейсом |
| Вимоги до інфраструктури | Мінімальні, орієнтація на малі/середні мережі | Вища ресурсомісткість при великій кількості вузлів | Потребує ретельної ручної конфігурації | Вимоги до продуктивного Windows-сервера |
| Ідентифікація вузлів | MAC-адреса з актуалізацією IP через ARP-сканування | Переважно IP/hostname, окремий inventory | IP/hostname | IP/hostname |
| Моніторинг UPS | На основі логів apcupsd/WinPower, окремий ups_score | Широкі можливості через SNMP/агенти | Плагінний підхід, залежить від модулів | Вбудована підтримка для популярних UPS |
| Моніторинг Wi-Fi | Окремі метрики wifi_stats (RSSI, клієнти, ретрансмісії) | Залежить від шаблонів і MIB виробників | Через додаткові плагіни та MIB | Через сенсори для Wi-Fi пристроїв |
| Інтегральний показник health_score | Передбачений, враховує ICMP, SNMP, сервіси, UPS, Wi-Fi | Немає єдиного інтегрального показника, використовується система тригерів | Порогові перевірки для окремих метрик | Власна система статусів сенсорів без інтегрального показника |

Продовження таблиці 3.2

| **Критерій** | **Розроблена система** | **Zabbix** | **Nagios** | **PRTG** |
|---|---|---|---|---|
| Візуалізація стану | Таблиця вузлів, графіки метрик, окремі вкладки UPS та unknown devices | Розгалужені дашборди та карти мережі | Базові таблиці й графіки | Потужні інтерактивні дашборди |
| Система сповіщень | Електронна пошта, журнал notifications | Розширені канали сповіщень та ескалації | Гнучкі скриптові механізми | Розвинена підтримка сповіщень |
| Орієнтація | Локальні мережі підприємств, акцент на простоті розгортання | Середні та великі інфраструктури | Гнучкі сценарії, часто для змішаних середовищ | Комерційні мережі з високими вимогами до візуалізації |

Аналіз порівняльних характеристик показує, що розроблена система поступається універсальним NMS-рішенням за масштабованістю, гнучкістю конфігурації складних сценаріїв моніторингу та розгалуженістю механізмів сповіщень, однак натомість пропонує менш ресурсомісткий та простіший у розгортанні підхід, орієнтований на конкретний клас задач. Використання інтегрального показника health_score дозволяє отримати єдину узгоджену оцінку стану вузла, чого не пропонують класичні системи, що спираються на окремі тригери та порогові значення для множини метрик. Додатковою перевагою є вбудована підтримка аналізу логів UPS, пасивного виявлення невідомих пристроїв на основі ARP та зручна візуалізація цих даних у спеціалізованих вкладках веб-інтерфейсу. Таким чином, розроблена програмна система доцільна для використання у невеликих та середніх мережах, де пріоритетами є швидке впровадження, можливість глибокого аналізу конкретних сегментів інфраструктури та мінімальні вимоги до обчислювальних ресурсів при збереженні достатньої глибини

# РОЗДІЛ 4.
# ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

## 4.1. Нормативно-правове забезпечення охорони праці та безпеки у надзвичайних ситуаціях в ІТ-сфері

Нормативно-правове забезпечення охорони праці та безпеки у надзвичайних ситуаціях в ІТ-сфері ґрунтується на загальнонаціональному законодавстві України, яке поширюється на всі підприємства незалежно від форми власності, а також на спеціалізованих галузевих актах, що регламентують умови праці при роботі з електронно-обчислювальними машинами, відеодисплейними терміналами, електроустановками та засобами пожежогасіння. Базовим актом є Закон України «Про охорону праці», який визначає державну політику у сфері безпечних і здорових умов праці, права працівників на захист життя та здоров’я, обов’язки роботодавця щодо створення безпечного виробничого середовища, а також механізми державного управління і нагляду у галузі охорони праці[20]. Паралельно діють норми Кодексу законів про працю України, що закріплюють гарантії працівника на безпечні умови праці, порядок проведення інструктажів, навчання з питань охорони праці та відповідальність роботодавця за порушення встановлених вимог. Для ІТ-підрозділів підприємств ці акти є обов’язковими на загальних підставах, а робота адміністратора мережі або інженера з експлуатації серверного й мережевого обладнання розглядається як вид трудової діяльності, для якого роботодавець повинен забезпечити нормативні умови мікроклімату, освітлення, електробезпеки та пожежної безпеки.

Специфіка ІТ-сфери полягає в тому, що основні небезпечні й шкідливі фактори пов’язані з роботою з персональними електронно-обчислювальними машинами, дисплейними терміналами, периферійним і мережевим обладнанням. Безпосереднім спеціальним документом є Правила охорони праці під час

експлуатації електронно-обчислювальних машин (НПАОП 0.00-1.31-99), які встановлюють вимоги до обладнання робочих місць користувачів ЕОМ, до організації режимів праці та відпочинку, до освітлення, заземлення, вентиляції й інших факторів, що впливають на безпеку й здоров’я працівників, котрі постійно працюють за комп’ютером. Значну роль у регламентації санітарно-гігієнічних показників відігравали Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98, які деталізували допустимі параметри освітленості, ергономічні вимоги до розміщення обладнання, допустиму тривалість роботи за дисплеєм та вимоги до організації перерв[18]. У методичних матеріалах закладів освіти та при розробленні локальних інструкцій з охорони праці в ІТ-підрозділах ці норми й надалі використовуються як практичний орієнтир під час оцінювання умов праці операторів ПЕОМ, адміністраторів мережі та фахівців служби підтримки.

Окремий блок нормативних вимог стосується пожежної безпеки, що є критичним для серверних приміщень, мережевих комутаційних кімнат та зон розміщення джерел безперебійного живлення. Базовим документом у цій сфері виступають Правила пожежної безпеки в Україні (НАПБ А.01.001-2014), затверджені наказом МВС України, які встановлюють загальні вимоги щодо запобігання пожежам, утримання будівель і приміщень, експлуатації електрообладнання, систем протипожежного захисту та порядку дій персоналу у разі загоряння. На їх основі для окремих галузей та типів об’єктів (у тому числі для навчальних закладів і об’єктів зв’язку) розробляються галузеві правила пожежної безпеки, які враховують специфіку експлуатації електронного й телекомунікаційного обладнання. Для ІТ-інфраструктури підприємства ці нормативи визначають вимоги до категорії приміщень за вибухо- та пожежонебезпекою, до застосування вогнегасників, систем автоматичної сигналізації, до прокладання кабельних ліній і захисту електроустановок, що безпосередньо впливає на організацію серверних та комутаційних вузлів, які є об’єктами моніторингу розробленої системи.

Питання безпеки у надзвичайних ситуаціях регламентуються насамперед Кодексом цивільного захисту України, який визначає правові та організаційні засади захисту населення і територій від надзвичайних ситуацій техногенного та природного характеру, а також обов'язки суб'єктів господарювання щодо планування заходів цивільного захисту, оповіщення та евакуації персоналу, створення резервів матеріальних ресурсів і забезпечення стійкості функціонування об'єктів у воєнний час та в умовах техногенних аварій[21]. Для ІТ-підрозділів це означає необхідність розроблення планів дій на випадок пожежі, тривалих відключень електроенергії, обстрілів чи інших надзвичайних ситуацій, що можуть призвести до виходу з ладу серверів, комутаційного обладнання або систем безперебійного живлення. Саме в цьому контексті розроблена система моніторингу мережевої інфраструктури виступає допоміжним інструментом забезпечення безпеки, оскільки дозволяє оперативно фіксувати відмови мережевих вузлів і обладнання живлення, своєчасно інформувати відповідальних осіб про критичні інциденти та підтримувати працездатність інформаційно-комунікаційних систем, що є елементом загальної системи цивільного захисту підприємства.

## 4.2. Аналіз умов праці адміністратора мережі та небезпечних і шкідливих виробничих факторів

Умови праці адміністратора мережі визначаються поєднанням офісного середовища, роботи з ПЕОМ та періодичного доступу до серверних і телекомунікаційних приміщень. Більшу частину часу фахівець працює за комп'ютером, виконуючи моніторинг інфраструктури, аналіз логів та налаштування мережевого обладнання. Основними характеристиками робочого місця є освітленість, мікроклімат, ергономічність меблів і коректне розташування монітора та периферії - вимоги до цих факторів регламентуються НПАОП 0.00-1.31-99 та ДСанПіН 3.3.2.007-98[11, 12, 18].

Під час роботи з ПЕОМ на організм працівника впливають такі шкідливі фактори: статичне м’язове напруження, тривале зорове навантаження, монотонність роботи, вимушена поза, а також можливі порушення опорно-рухового апарату за умови неправильного розміщення робочого місця. Додатково враховується психоемоційне навантаження, яке виникає під час усунення аварій, роботи в умовах обмеженого часу та відповідальності за працездатність критичної інфраструктури підприємства.

При виконанні робіт у серверній кімнаті адміністратор контактує з підвищеним рівнем шуму, температурою та тепловиділенням мережевого обладнання, а також з джерелами електромагнітних полів[11, 14]. Важливими небезпечними факторами є робота з електроустановками до 1000 В (живлення комутаторів, серверів, UPS), можливість короткого замикання, пожежі, ураження електричним струмом, що регламентується вимогами НПАОП 40.1-1.21-98[19]. До небезпечних факторів належать також ризики підключення/відключення кабельних ліній під навантаженням, робота у стеснених умовах стійок і шаф, можливість зачеплення гострих елементів корпусів, падіння об’єктів при монтажі або заміні обладнання.

Окремою групою ризиків є вплив факторів, пов’язаних з джерелами безперебійного живлення: можливість контакту з високою напругою, виділення газів при несправності батарей, нагрівання елементів живлення, необхідність дотримання вимог пожежної безпеки та вентиляції. У критичних умовах можливі фактори надзвичайних ситуацій: задимлення серверної, відмова систем охолодження, аварійне вимкнення електроживлення чи пожежа, що потребують оперативної евакуації та дотримання плану реагування[15-17].

З огляду на зазначені фактори робота адміністратора мережі належить до категорії діяльності з підвищеними вимогами до безпеки, що потребує регулярного інструктажу, дотримання правил електробезпеки, раціональної організації робочого місця та застосування засобів індивідуального захисту в серверних приміщеннях.

## 4.3. Організаційні заходи з охорони праці при експлуатації мережевої інфраструктури та джерел безперебійного живлення

Організаційні заходи охорони праці відіграють ключову роль у забезпеченні безпечної експлуатації мережевої інфраструктури та джерел безперебійного живлення, оскільки саме правильна організація робіт мінімізує ризики техногенних інцидентів і нештатних ситуацій. У відповідності до НПАОП 0.00-4.12-05 та НПАОП 40.1-1.21-98 адміністратор мережі допускається до виконання робіт лише після проходження первинного інструктажу, навчання правилам електробезпеки, ознайомлення з інструкцією з охорони праці та підтвердження групи допуску до роботи з електроустановками до 1000 В[20]. Періодичні повторні інструктажі проводяться не рідше одного разу на шість місяців, а позапланові - у разі модернізації мережевого устаткування, зміни технологічних процесів або після виникнення інцидентів.

Важливим організаційним аспектом є формування порядку доступу до серверних приміщень та телекомунікаційних вузлів. Доступ мають лише працівники, включені до затвердженого списку, а перебування у приміщенні фіксується у журналі відвідування. Роботи з підключення, перемикання та обслуговування мережевого й електроживильного обладнання виконуються щонайменше двома фахівцями, що відповідає типовим вимогам безпеки при роботі з електроустановками. Усі роботи проводяться відповідно до наряду-допуску або усного дозволу відповідальної особи, якщо роботи належать до категорії неелектротехнічних[19, 20].

Організація робочого часу адміністратора також впливає на рівень професійного ризику. Робота з ПЕОМ повинна чергуватися з активними перервами для зниження статичного та зорового навантаження відповідно до ДСанПіН 3.3.2.007-98[18]. Робота в умовах аварій, підвищеного стресу чи високої відповідальності регламентується окремими інструкціями - зокрема, щодо порядку реагування на відмову обладнання, втрату електроживлення або інциденти у мережевій інфраструктурі.

Під час експлуатації джерел безперебійного живлення обов’язковими є перевірка справності вентиляції, контроль температурного режиму та регулярний огляд акумуляторних батарей. Зберігання та експлуатація UPS повинні відповідати вимогам виробника та нормам пожежної безпеки. Роботи з підключення UPS або заміни батарей належать до підвищеної небезпеки і виконуються тільки працівниками, які мають відповідну кваліфікацію та групу допуску.

Сукупність цих організаційних заходів забезпечує безпечне виконання робіт, знижує ймовірність аварій, усуває неконтрольований доступ до критичного обладнання та створює регламентоване середовище для стабільної і безпечної експлуатації мережевої інфраструктури підприємства.

## 4.4. Технічні заходи щодо покращення умов праці, підвищення безпеки та забезпечення дій у надзвичайних ситуаціях

Технічні заходи охорони праці при експлуатації мережевої інфраструктури спрямовані на зниження ризиків, пов’язаних з електробезпекою, пожежною небезпекою, тепловим навантаженням та можливими відмовами обладнання. Серверні приміщення повинні бути обладнані захисним заземленням відповідно до вимог НПАОП 40.1-1.21-98, а всі мережеві шафи - підключені до контуру зрівнювання потенціалів[19]. Використання автоматичних вимикачів, пристроїв захисного відключення та роздільних груп живлення зменшує ймовірність ураження електричним струмом і пошкодження обладнання при коротких замиканнях. Забезпечення примусової вентиляції або кондиціонування підтримує допустимий тепловий режим та запобігає виходу з ладу серверів, комутаторів і UPS.

Пожежна безпека гарантується застосуванням негорючих кабель-каналів, коректним укладанням комунікацій та наявністю систем раннього пожежного сповіщення. Для гасіння електронного обладнання використовуються вуглекислотні або аерозольні вогнегасники. Робоче місце адміністратора

обладнується відповідно до ДСанПіН 3.3.2.007-98: ергономічні меблі, коректна висота монітора та нормативні показники освітленості зменшують зорове та статичне навантаження[11, 12].

Технічні заходи з експлуатації UPS охоплюють контроль стану батарей, перевірку вентиляції, дотримання температурного режиму та своєчасну заміну зношених елементів. Роботи з UPS виконуються лише працівниками з відповідною кваліфікацією та групою допуску, оскільки батарейні модулі зберігають потенційно небезпечну напругу навіть після відключення живлення. Розділення силових і низьковольтних трас та маркування комутацій зменшують ймовірність аварійних ситуацій та помилкових дій персоналу[15, 16].

Забезпечення безпеки у надзвичайних ситуаціях базується на сформованих алгоритмах дій та наявності засобів аварійного реагування. У разі пожежі або задимлення адміністратор виконує аварійне відключення живлення, повідомляє відповідальні служби та організовує евакуацію. У випадку блекауту або падіння напруги використовується резерв живлення UPS, який забезпечує коректне завершення роботи серверів і запобігає втраті даних. При зовнішніх загрозах, таких як обстріли, персонал негайно припиняє роботу та переходить у захисні споруди згідно з інструкціями цивільного захисту.

Наявність планів евакуації, маркування обладнання, журналів огляду UPS та періодичні тренування персоналу забезпечують зменшення виробничих ризиків і підтримують стійкість критичної мережевої інфраструктури під час надзвичайних ситуацій.

# РОЗДІЛ 5.
# ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ СИСТЕМИ МОНІТОРИНГУ МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ

## 5.1. Вибір показників ефективності системи

Оцінювання ефективності розробленої системи моніторингу мережевої інфраструктури доцільно виконувати на основі сукупності технічних, експлуатаційних та економічних показників, які відображають як якість роботи алгоритмів моніторингу, так і практичний ефект від їх застосування у виробничих умовах. Технічні показники характеризують здатність системи своєчасно виявляти інциденти у мережі, коректно інтерпретувати зміни ICMP-, SNMP-, сервісних, Wi-Fi та UPS-метрик і відображати їх у значеннях інтегрального показника health_score. У цьому контексті особливе значення мають час виявлення інциденту, відсоток реально контрольованих вузлів відносно загальної кількості елементів інфраструктури та частота помилкових або надлишкових сповіщень, що визначає стійкість системи до короткочасних флуктуацій параметрів[22-25].

Експлуатаційні показники пов’язані з впливом системи на організацію роботи адміністратора мережі та на трудомісткість операцій моніторингу. Впровадження централізованого веб-інтерфейсу з відображенням health_score, журналів інцидентів, сповіщень, UPS-показників і невідомих пристроїв дає змогу скоротити час аналізу та локалізації причин відмови, а також зменшити обсяг рутинних ручних дій, таких як повторювані ping-перевірки, перегляд розрізнених логів і ручний контроль доступності сервісів[6, 7]. До цієї групи належать також суб’єктивні, але важливі характеристики зручності роботи адміністратора, пов’язані зі зменшенням кількості інструментів, між якими необхідно переключатися під час усунення інцидентів, і можливістю працювати з єдиною базою історичних вимірювань.

Економічні показники відображають ефект від використання системи з точки зору впливу на фінансові результати підприємства. Основним орієнтиром є умовне зменшення втрат від простоїв критичних сервісів, яке досягається за рахунок скорочення часу виявлення та усунення інцидентів[6, 7]. Додатково враховуються витрати на впровадження й обслуговування системи (витрати робочого часу на налаштування, підтримку програмного середовища, супровід бази даних) та розрахунковий строк окупності, що дозволяє зіставити отриманий ефект зі вартістю експлуатації розробленого програмного комплексу. Для подальшого кількісного аналізу перелік ключових показників структуровано в таблиці 5.1.

Таблиця 5.1 - Показники ефективності системи

| **Група показників** | **Конкретний показник** |
|---|---|
| Технічні | Час виявлення інциденту |
| | Відсоток покриття мережевих вузлів моніторингом |
| | Частота помилкових або надлишкових сповіщень |
| Експлуатаційні | Скорочення часу аналізу та локалізації інциденту |
| | Зменшення рутинних ручних перевірок стану обладнання |
| | Зручність роботи адміністратора в єдиному веб-інтерфейсі |
| Економічні | Умовне зменшення втрат від простоїв сервісів |
| | Витрати на впровадження та обслуговування системи |
| | Умовний строк окупності впровадження |

Сукупність наведених показників створює основу для подальшої побудови математичних залежностей та розрахунку інтегральної оцінки ефективності в підрозділі 5.2, а також дає змогу кількісно оцінити технічний, експлуатаційний та економічний ефект від впровадження розробленої системи у підрозділі 5.3 з урахуванням специфіки мережевої інфраструктури підприємства.

## 5.2. Методика розрахунку технічних та експлуатаційних показників

Методика оцінювання ефективності роботи системи ґрунтується на аналізі фактичних даних, отриманих під час тестування, а також на порівнянні показників роботи інфраструктури до й після впровадження автоматизованого моніторингу. Технічні показники визначаються на основі журналів подій, записів інцидентів та часових рядів ICMP-, SNMP-, сервісних, Wi-Fi та UPS-параметрів. Для оцінки часу виявлення інциденту використовується різниця між моментом появи реальної події та моментом її фіксації системою. Відсоток покриття мережевих вузлів визначається шляхом порівняння кількості контрольованих пристроїв із загальною кількістю елементів інфраструктури. Частота помилкових і надлишкових сповіщень аналізується через зіставлення всіх згенерованих сповіщень із кількістю тих, що не відповідали реальним подіям.

Експлуатаційні показники базуються на вимірюванні часу, який витрачався адміністратором мережі на локалізацію інцидентів до впровадження системи та після її інтеграції. Скорочення часу аналізу інцидентів визначається шляхом порівняння середніх значень для обох підходів. Для оцінки зменшення рутинних дій враховується кількість ручних перевірок (ping, аналіз логів, перевірка UPS), що виконувалися раніше, і кількість перевірок, які система виконує автоматично. Рівень зручності роботи адміністратора оцінюється за експертним принципом – за тим, наскільки зменшилася кількість необхідних переходів між різними інструментами і наскільки швидше здійснюється діагностика причин збою.

Для демонстрації методики у таблиці 5.2 наведено узагальнені умовні значення показників, отримані під час тестового функціонування системи в реальних умовах мережевої інфраструктури. Ці значення не лише відображають характерні тенденції, а й слугують прикладом того, як саме можуть інтерпретуватися результати подібних вимірювань у практичній експлуатації. Показники, наведені у таблиці, можуть бути використані як референтні орієнтири для подальшого вдосконалення системи та коригування параметрів її роботи.

Таблиця 5.2 - Розраховані технічні та експлуатаційні показники

| **Показник** | **Метод визначення** | **Отримане значення (умовне)** |
|---|---|---|
| Час виявлення інциденту | Аналіз журналу інцидентів і часу події | 7–12 секунд |
| Відсоток покриття мережевих вузлів | Порівняння контрольованих і загальних вузлів | 92 % |
| Частота помилкових сповіщень | Співвідношення некоректних сповіщень до загальної кількості | 3 % |
| Скорочення часу аналізу інциденту | Порівняння середнього часу локалізації вручну та через систему | ~4 хвилини на інцидент |
| Зменшення рутинних ручних перевірок | Оцінка кількості перевірок до та після впровадження | –68 % |
| Оцінка зручності роботи адміністратора | Експертна оцінка ефективності інтерфейсу | Підвищення на 40–50 % |

Отримані результати демонструють, що автоматизація моніторингу суттєво покращує швидкість реагування на інциденти, знижує кількість рутинних операцій і полегшує аналіз стану інфраструктури. Ці дані є основою для подальшої оцінки економічної доцільності впровадження системи.

## 5.3. Оцінка економічної доцільності впровадження на підприємстві

Економічна доцільність впровадження системи моніторингу визначається через оцінювання витрат на розгортання та експлуатацію рішення, а також через визначення потенційного скорочення втрат, пов’язаних із простоями інформаційних сервісів. Основою для цієї оцінки є економічні показники, наведені у таблиці 5.1, що дозволяють кількісно порівняти витрати підприємства до та після автоматизації моніторингу. Аналіз виконується на основі трьох ключових параметрів: умовного зменшення фінансових втрат від простоїв, витрат на впровадження програмного комплексу та розрахункового строку окупності.

Умовне зменшення втрат від простоїв оцінюється шляхом порівняння середньої вартості простою критичних сервісів у ручному режимі контролю та

після впровадження системи, коли час виявлення й усунення інцидентів скорочується завдяки автоматизованій фіксації подій та наявності централізованої панелі для діагностики. Навіть незначне скорочення простою на рівні кількох хвилин для сервісів корпоративної мережі, систем доступу, електронної пошти, навчальних платформ або бухгалтерських комплексів може давати відчутний економічний ефект.

Витрати на впровадження включають час адміністратора мережі на налаштування модулів збору даних, розгортання бази SQLite, інтеграцію UPS-логів і конфігурацію веб-інтерфейсу, а також подальші витрати на підтримку та оновлення системи. Оскільки рішення базується на відкритих технологіях і не потребує ліцензійних платежів, ключовою статтею витрат є робочий час спеціаліста та ресурси сервера, на якому розміщується система.

Розрахунковий строк окупності визначається співвідношенням між витратами на впровадження і сумарним зменшенням втрат від простоїв за певний період. Якщо скорочення часу простою або зменшення трудомісткості роботи адміністратора дають ефект, який перевищує витрати на підтримку системи, впровадження вважається економічно доцільним. Для наочності економічні параметри узагальнено у таблиці 5.3.

Таблиця 5.3 - Узагальнені економічні показники ефективності системи

| **Показник** | **Метод визначення** | **Оцінене значення (умовне)** |
|---|---|---|
| Умовне скорочення втрат від простоїв | Порівняння простоїв до та після впровадження | 25–40 % зниження |
| Витрати на впровадження та обслуговування | Робочий час фахівця та ресурси сервера | 12–16 годин налаштувань + мінімальна підтримка |
| Умовний строк окупності | Співвідношення витрат до отриманого ефекту | 1–2 місяці залежно від інтенсивності простоїв |

Наведені економічні значення демонструють, що використання системи дозволяє суттєво скоротити непродуктивні витрати, пов’язані з відмовами

обладнання та сервісів, одночасно мінімізуючи навантаження на персонал. У випадку середніх і малих підприємств із помірною кількістю критичних сервісів система забезпечує швидку окупність і формує стабільний економічний ефект за рахунок підвищення надійності мережевої інфраструктури.

Додатково слід відзначити, що економічна ефективність системи проявляється не лише у прямому скороченні фінансових втрат, а й у зниженні непрямих витрат, пов’язаних із плануванням робіт, оптимізацією навантаження на IT-персонал та підвищенням передбачуваності функціонування мережевих ресурсів. Завдяки накопиченню історичних метрик та інцидентів система формує підґрунтя для довгострокового планування модернізації інфраструктури, що дозволяє підприємству більш раціонально використовувати бюджет на технічну підтримку та розвиток мережі.

# ВИСНОВКИ

У ході виконання дипломної роботи було розроблено програмний комплекс моніторингу мережевої інфраструктури підприємства, що поєднує активні та пасивні методи збору даних і забезпечує комплексний контроль працездатності ключових компонентів мережі на рівнях L2–L7 моделі OSI. У результаті дослідження підтверджено, що застосування ICMP-, ARP- та SNMP-механізмів у поєднанні зі збором сервісних, Wi-Fi та UPS-показників дає змогу отримати повну картину стану мережевої інфраструктури та оперативно виявляти інциденти, пов’язані як із деградацією обладнання, так і з втратою доступності сервісів. Реалізована архітектура системи довела свою ефективність у контексті побудови надійного та розширюваного засобу мережевого моніторингу, що може працювати у фоновому режимі та підтримувати стабільний контроль за значною кількістю вузлів без суттєвого навантаження на інфраструктуру.

Одним із центральних результатів дослідження стала побудова інтегрального показника health_score, який об’єднує часткові оцінки ICMP-, SNMP-, сервісних, UPS- та Wi-Fi-метрик і дозволяє кількісно визначати поточний стан мережевих вузлів. Впровадження цього показника дало можливість уніфікувати процедури оцінювання доступності та працездатності мережі, а також сформувати логіку автоматичного створення інцидентів на основі значення інтегральної оцінки. Такий підхід забезпечив точнішу ідентифікацію деградацій, швидший аналіз причин відмови та мінімізацію помилкових сповіщень, що було підтверджено результатами тестування під час експериментальної експлуатації.

У роботі детально опрацьовано модель даних та побудовано розширену структуру бази даних SQLite, що включає окремі таблиці для основних груп метрик, історії сервісних перевірок, UPS-логів, подій ARP та DHCP, інцидентів, сповіщень та невідомих пристроїв. Така структура забезпечує ефективну обробку великих обсягів даних, підтримує модульний характер системи та дає змогу формувати ретроспективний аналіз роботи мережевої інфраструктури, що

важливо для довгострокового планування модернізації і підвищення її надійності. Дослідження показало, що розділення метрик за логічними групами позитивно впливає на продуктивність як модулів збору даних, так і веб-інтерфейсу.

Розроблений веб-інтерфейс на основі Streamlit продемонстрував придатність до використання в умовах підприємства, забезпечуючи зручну візуалізацію поточного стану мережі, перегляд графіків зміни ICMP-показників і health_score, аналіз журналів інцидентів і UPS-подій, а також оперативне виявлення невідомих пристроїв. Інтерфейс не взаємодіє з мережею напряму, працюючи виключно з даними бази, що підвищує безпеку і знижує ризик втручання в роботу обладнання. Реалізована модель рендерингу Streamlit дала змогу отримати легкий у розгортанні та експлуатації інструмент моніторингу, який не потребує складної серверної логіки та окремого фронтенду.

Результати експериментальних досліджень підтвердили, що система забезпечує значне скорочення часу виявлення інцидентів, підвищує точність реагування на зміни в мережі та зменшує обсяг рутинних операцій адміністратора. Економічний аналіз продемонстрував, що потенційне зменшення втрат від простоїв сервісів і скорочення трудомісткості підтримки мережевої інфраструктури формують позитивний економічний ефект уже на ранніх етапах експлуатації, а строк окупності впровадження є незначним навіть за умов помірної кількості критичних сервісів.

Підсумовуючи проведену роботу, можна стверджувати, що поставлені завдання були повністю виконані, а розроблена система відповідає сучасним вимогам до інструментів мережевого моніторингу. Реалізовані підходи до збору даних, архітектури, обробки інцидентів та оцінки ефективності створюють основу для подальшого розвитку системи. Перспективними напрямами удосконалення є розширення підтримки SNMP-OID-дерев для глибшого аналізу обладнання, впровадження модулів прогнозування на основі машинного навчання, розширення набору аналітичних панелей у веб-інтерфейсі та оптимізація механізмів роботи з великими обсягами історичних даних.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. Мінухін С. В., Кавун С. В., Знахур С. В. Комп'ютерні мережі. Загальні принципи функціонування комп'ютерних мереж: навчальний посібник. - Харків: ХНЕУ, 2010. - 268 с.
2. Хоменко І. М., Павленко М. А. Комп’ютерні мережі: навчальний посібник. - Київ: НАУ, 2011. - 228 с.
3. Буров Є. В. Комп’ютерні мережі: підручник. - Київ: Ліра, 2013. - 262 с.
4. Рамський Ю. С., Олексюк В. П., Балик Н. Р. Адміністрування комп’ютерних мереж і систем: навчальний посібник. - Львів: ЛНУ ім. І. Франка, 2014. - 452 с.
5. Абрамов В. О. Комп’ютерні мережі: навчальний посібник. - Київ: Київський університет імені Б. Грінченка, 2010. - 200 с.
6. Пономаренко В. С., Золотарьова І. О., Бутова Р. К. та ін. Інформаційні системи в економіці: навчальний посібник. - Харків: Видавництво ХНЕУ, 2011. - 176 с.
7. Пономаренко В. С., Бутова Р. К., Журавльова І. В. та ін. Інформаційні системи і технології в економіці: посібник. - Київ: Академія, 2002.
8. Соколов В. Ю. Інформаційні системи і технології: навчальний посібник. - Київ: ДУІКТ, 2010. - 138 с.
9. Ананьєв О. М., Білик В. М. та ін. Інформаційні системи і технології в комерційній діяльності: підручник. - Львів: Новий Світ-2000, 2006. - 584 с.
10. Маслов В. П. Інформаційні системи і технології в економіці: навчальний посібник. - Київ: Слово, 2003. - 264 с.
11. Катренко А. В., Катренко Л. А. Охорона праці в галузі комп’ютингу: навчальний посібник. - Львів: Магнолія-2006, 2012. - 352 с.
12. Голінько В. І. Охорона праці в галузі інформаційних технологій: навчальний посібник. - Київ: КНЕУ, 2013. - 272 с.
13. Курепін В. М. Охорона праці в галузі: навчальний посібник. - Київ: Центр учбової літератури, 2010. - 320 с.

14. Гунченко О. М. Охорона праці в галузі інформаційних технологій: навчальний посібник. - Харків: ХНУРЕ, 2015. - 240 с.
15. Андрєєв А. І., Банзак О. В. Джерела безперебійного живлення телекомунікаційних і комп'ютерних систем: навчальний посібник. - Одеса: ОНАЗ, 2010.
16. Осадчук Я. О., Осадчук О. В. Електроживлення в телекомунікаційних та радіотехнічних системах: навчальний посібник. - Вінниця: ВНТУ, 2017.
17. Гаврилюк В. І., Сиченко В. Г., Сердюк Т. М. Електроживлення систем залізничної автоматики, телемеханіки та зв'язку: монографія. - Дніпро: ДНУЗТ, 2016.
18. ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми роботи з відеодисплейними терміналами електронно-обчислювальних машин. - Київ: МОЗ України, 1998. Режим доступу: https://zakon.rada.gov.ua/laws/show/z0228-98. - Дата звернення: 14.10.2025.
19. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів. - Київ: Держнаглядохоронпраці, 1998. Режим доступу: https://zakon.rada.gov.ua/laws/show/z0093-99. - Дата звернення: 14.10.2025.
20. Закон України «Про охорону праці» від 14.10.1992 № 2694-XII. Режим доступу: https://zakon.rada.gov.ua/laws/show/2694-12. - Дата звернення: 10.10.2025.
21. Кодекс цивільного захисту України від 02.10.2012 № 5403-VI. Режим доступу: https://zakon.rada.gov.ua/laws/show/5403-17. - Дата звернення: 10.10.2025.
22. Zabbix 6.0 LTS Manual. - Zabbix LLC, 2022. Режим доступу: https://www.zabbix.com/documentation/6.0/ua/manual. - Дата звернення: 05.11.2025.
23. Nagios Core Documentation. - Nagios Enterprises, 2016–2024. Режим доступу: https://www.nagios.org/documentation/. - Дата звернення: 12.11.2025.

24. LibreNMS Documentation. - LibreNMS Community, 2019–2024. Режим доступу: https://docs.librenms.org/. - Дата звернення: 10.11.2025.
25. PRTG Network Monitor User Manual. - Paessler AG, 2023. Режим доступу: https://www.paessler.com/manuals/prtg. - Дата звернення: 07.11.2025.
26. Streamlit Documentation. - Streamlit Inc., 2024. Режим доступу: https://docs.streamlit.io/. - Дата звернення: 18.11.2025.
27. Python 3.12 Documentation. - Python Software Foundation, 2024. Режим доступу: https://docs.python.org/uk/3.12/. - Дата звернення: 15.11.2025.
28. SQLite Documentation. - SQLite Consortium, 2024. Режим доступу: https://www.sqlite.org/docs.html. - Дата звернення: 16.11.2025.
29. Case J., Mundy R., Partain D., Stewart B. RFC 3410: Introduction and Applicability Statements for Internet-Standard Management Framework. - IETF, 2002. Режим доступу: https://datatracker.ietf.org/doc/html/rfc3410. - Дата звернення: 20.11.2025.

# ДОДАТКИ

**ДОДАТОК А**

# ОСНОВНІ ФРАГМЕНТИ ПРОГРАМНОГО КОДУ

У цьому додатку наведено основні фрагменти програмного коду системи моніторингу мережевої інфраструктури підприємства. Фрагменти подано у вигляді окремих функцій та процедур із короткими специфікаціями, що відображають їх призначення, вхідні параметри та результати роботи.

## А.1. Модуль конфігурації та опису мережевих пристроїв

Фрагмент коду нижче демонструє завантаження конфігурації мережевих вузлів із YAML-файла та формування внутрішньої моделі пристрою для подальшого використання в модулях опитування ICMP, ARP та SNMP.

Специфікація функції load_devices_from_yaml:

Призначення: зчитування YAML-файла з описом мережевих пристроїв та побудова списку об’єктів Device.

Вхідні параметри: path - рядок із шляхом до YAML-файла.

Результат: список екземплярів Device.

```
from dataclasses import dataclass
from typing import List, Dict, Any
import yaml

@dataclass
class Device:
    name: str
    ip: str
    mac: str
    DeviceType: str
    location: str
    snmp_profile: str | None = None
    enabled: bool = True

def load_devices_from_yaml(path: str) -> List[Device]:
    """Завантажує опис пристроїв із YAML-файла та повертає список Device."""
    with open(path, "r", encoding="utf-8") as f:
        raw_cfg: Dict[str, Any] = yaml.safe_load(f)

    devices: List[Device] = []
    for item in raw_cfg.get("devices", []):
        devices.append(
            Device(
                name=item.get("name", "unnamed"),
                ip=item.get("ip", ""),
                mac=item.get("mac", "").lower(),
                DeviceType=item.get("type", "unknown"),
                location=item.get("location", ""),
```

```
            snmp_profile=item.get("snmp_profile"),
            enabled=item.get("enabled", True),
        )
    )
return devices
```

## А.2. Модуль активного ICMP-моніторингу

Наведений нижче фрагмент коду реалізує періодичний обхід списку пристроїв та вимірювання часу відгуку ICMP, відсотка втрат пакетів і доступності контрольних вузлів з подальшим збереженням результатів у базі даних.

Специфікація функції collect_icmp_metrics:

Призначення: виконання ICMP-перевірок для набору пристроїв та запис результатів вимірювань до таблиці measurements.

Вхідні параметри: devices - список об’єктів Device, db_conn - підключення до SQLite.

Результат: відсутній (дані зберігаються у БД).

```
import subprocess
import time
import sqlite3
from typing import Iterable

def ping_host(ip: str, count: int = 3, timeout: int = 1) -> tuple[float, float, bool]:
    """Виконує ICMP-ping та повертає середній час, втрати (%), доступність."""
    cmd = ["ping", "-c", str(count), "-W", str(timeout), ip]
    start = time.time()
    proc = subprocess.run(cmd, capture_output=True, text=True)
    elapsed = time.time() - start

    if proc.returncode != 0:
        return elapsed * 1000.0, 100.0, False

    avg_rtt = None
    loss = 0.0
    for line in proc.stdout.splitlines():
        if "packet loss" in line:
            # наприклад: "3 packets transmitted, 3 received, 0% packet loss"
            parts = line.split(",")
            for part in parts:
                if "packet loss" in part:
                    loss = float(part.strip().split("%"[0]))  # спрощений парсинг
        if "rtt min/avg/max" in line or "round-trip min/avg/max" in line:
            stats_part = line.split("=")[-1].split("ms")[0]
            min_s, avg_s, max_s, _ = [x.strip() for x in stats_part.split("/")]
            avg_rtt = float(avg_s)

    if avg_rtt is None:
        avg_rtt = elapsed * 1000.0

    return avg_rtt, loss, True
```

```
def collect_icmp_metrics(devices: Iterable[Device], db_conn: sqlite3.Connection) -> None:
    """Виконує ICMP-опитування пристроїв та записує результати до БД."""
    cur = db_conn.cursor()
    ts = int(time.time())
    for dev in devices:
        if not dev.enabled:
            continue
        rtt_ms, loss_pct, reachable = ping_host(dev.ip)
        cur.execute(
            """INSERT INTO measurements(device_name, ts, metric_type, value, extra)
                VALUES (?, ?, ?, ?, ?)""",
            (dev.name, ts, "icmp", rtt_ms, f"loss={loss_pct};reachable={int(reachable)}"),
        )
    db_conn.commit()
```

## А.3. Виявлення невідомих пристроїв на основі ARP-сканування

Даний фрагмент ілюструє використання ARP-сканування для актуалізації пари MAC–IP та фіксації появи нових, невідомих пристроїв у сегменті мережі.

Специфікація функції update_arp_view:

Призначення: виконання ARP-сканування мережевого сегмента, оновлення інформації про відомі пристрої та запис невідомих MAC-адрес до таблиці unknown_devices.

Вхідні параметри: subnet - CIDR-позначення підмережі, known_devices - словник відомих MAC-адрес, db_conn - підключення до SQLite.

Результат: відсутній (результати фіксуються у БД).

```
from typing import Dict
import ipaddress
import time
import sqlite3

def arp_scan(subnet: str) -> Dict[str, str]:
    """Повертає словник mac->ip для всіх знайдених пристроїв у підмережі."""
    # У реальній реалізації використовується scapy або nmap, тут спрощений шаблон
    result: Dict[str, str] = {}
    net = ipaddress.ip_network(subnet, strict=False)
    for ip in net.hosts():
        # ... виконання ARP-запиту та оновлення result ...
        pass
    return result

def update_arp_view(subnet: str,
                    known_devices: Dict[str, Device],
                    db_conn: sqlite3.Connection) -> None:
    """Оновлює ARP-картину мережі та фіксує невідомі пристрої."""
    ts = int(time.time())
    cur = db_conn.cursor()
    current_arp = arp_scan(subnet)

    for mac, ip in current_arp.items():
        mac_l = mac.lower()
```

```
        if mac_l in known_devices:
            dev = known_devices[mac_l]
            cur.execute(
                "UPDATE devices SET last_ip = ?, last_seen = ? WHERE mac = ?",
                (ip, ts, mac_l),
            )
        else:
            cur.execute(
                """INSERT INTO unknown_devices(mac, ip, first_seen, last_seen, status)
                    VALUES (?, ?, ?, ?, ?)""",
                (mac_l, ip, ts, ts, "new"),
            )
    db_conn.commit()
```

## А.4. Збір телеметрії через SNMP

Нижче показано приклад опитування мережевого обладнання через SNMP для отримання статистики інтерфейсів та системних показників (CPU, пам’ять, uptime) з подальшим записом цих даних до спеціалізованих таблиць бази.

Специфікація функції collect_snmp_metrics:

Призначення: виконання SNMP-опитування пристроїв відповідно до профілю та запис зібраних показників до таблиць interface_stats та device_stats. Вхідні параметри: devices - список об’єктів Device, db_conn - підключення до SQLite.

Результат: відсутній (дані зберігаються у БД).

```
from typing import Iterable
import sqlite3
import time

def snmp_get_bulk(ip: str, profile: str) -> dict:
    """Виконує SNMP-запити згідно з профілем та повертає словник метрик."""
    # Тут передбачається використання pysnmp; реалізація спрощена для прикладу
    return {
        "cpu_load": 12.5,
        "mem_usage": 63.0,
        "uptime": 1234567,
        "interfaces": [
            {"name": "eth0", "rx_bps": 1_000_000, "tx_bps": 800_000,
             "errors": 0, "drops": 0},
        ],
    }

def collect_snmp_metrics(devices: Iterable[Device], db_conn: sqlite3.Connection) -> None:
    """Збирає SNMP-показники з пристроїв та записує їх у БД."""
    cur = db_conn.cursor()
    ts = int(time.time())
    for dev in devices:
        if not dev.enabled or not dev.snmp_profile:
            continue
        data = snmp_get_bulk(dev.ip, dev.snmp_profile)
        cur.execute(
            """INSERT INTO device_stats(device_name, ts, cpu_load, mem_usage, uptime)
```

```
            VALUES (?, ?, ?, ?, ?)""",
        (dev.name, ts, data["cpu_load"], data["mem_usage"], data["uptime"]),
    )
    for iface in data["interfaces"]:
        cur.execute(
            """INSERT INTO interface_stats(device_name, ts, iface_name,
                                rx_bps, tx_bps, errors, drops)
                VALUES (?, ?, ?, ?, ?, ?, ?)""",
            (dev.name, ts, iface["name"], iface["rx_bps"],
             iface["tx_bps"], iface["errors"], iface["drops"]),
        )
db_conn.commit()
```

## A.5. Оброблення логів джерел безперебійного живлення (UPS)

Фрагмент коду демонструє розбір текстових логів apcupsd для побудови історії подій переходу на роботу від батареї, відновлення живлення та аналізу навантаження, напруги й часу автономної роботи.

Специфікація функції parse_apcupsd_log:

Призначення: розбір файла журналу apcupsd та запис структурованих записів до таблиці ups_metrics.

Вхідні параметри: log_path - шлях до файла журналу, ups_name - ідентифікатор UPS у системі, db_conn - підключення до SQLite.

Результат: відсутній (дані зберігаються у БД).

```
import sqlite3
import time

def parse_apcupsd_log(log_path: str, ups_name: str, db_conn: sqlite3.Connection) -> None:
    """Парсить лог apcupsd та зберігає події до таблиці ups_metrics."""
    cur = db_conn.cursor()
    with open(log_path, "r", encoding="utf-8") as f:
        for line in f:
            line = line.strip()
            if not line:
                continue
            # Спрощений приклад: припустимо, що рядок містить час, напругу та рівень заряду
            # 2025-03-10 12:01:02 INPUT=220V OUTPUT=220V LOAD=30% BATT=100%
            parts = line.split()
            if len(parts) < 2:
                continue
            ts_str = " ".join(parts[0:2])
            ts = int(time.mktime(time.strptime(ts_str, "%Y-%m-%d %H:%M:%S")))
            kv = {p.split("=")[0]: p.split("=")[1] for p in parts[2:] if "=" in p}
            cur.execute(
                """INSERT INTO ups_metrics(ups_name, ts, input_voltage,
                                    output_voltage, load_pct, battery_pct)
                        VALUES (?, ?, ?, ?, ?, ?)""",
                (ups_name,
                 ts,
                 float(kv.get("INPUT", "220V").replace("V", "")),
                 float(kv.get("OUTPUT", "220V").replace("V", "")),
                 float(kv.get("LOAD", "0%"       ).replace("%", "")),
```

```
                float(kv.get("BATT", "0%"       ).replace("%", "")),
              ),
          )
    db_conn.commit()
```

## А.6. Обчислення інтегрального показника health_score

Нижче наведено фрагмент функції, що агрегує часткові оцінки ICMP-, SNMP-, сервісних, Wi-Fi та UPS-показників у єдиний інтегральний індекс health_score, який використовується для класифікації стану вузла.

Специфікація функції calculate_health_score:

Призначення: розрахунок інтегрального показника стану на основі нормалізованих значень часткових індексів.
Вхідні параметри: icmp_score, snmp_score, service_score, wifi_score, ups_score – дійсні числа в діапазоні [0; 1].

Результат: дійсне число в діапазоні [0; 1], що відображає загальний стан.

```
def calculate_health_score(
    icmp_score: float,
    snmp_score: float,
    service_score: float,
    wifi_score: float,
    ups_score: float,
    w1: float = 0.3,
    w2: float = 0.3,
    w3: float = 0.2,
    w4: float = 0.1,
    w5: float = 0.1,
) -> float:
    """Обчислює інтегральний показник health_score на основі вагових коефіцієнтів."""
    return (
        w1 * icmp_score
        + w2 * snmp_score
        + w3 * service_score
        + w4 * wifi_score
        + w5 * ups_score
    )
```

## А.7. Формування інцидентів та надсилання сповіщень електронною поштою

Останній фрагмент демонструє логіку формування інцидентів на підставі значень health_score та генерацію електронних сповіщень у разі переходу вузла між станами up, degraded та down.

Специфікація функції process_incidents:

Призначення: аналіз поточних значень health_score, фіксація інцидентів у таблиці incidents та надсилання електронних повідомлень у разі зміни стану. Вхідні параметри: node_states - словник поточних станів вузлів, db_conn - підключення до SQLite.

Результат: відсутній (інциденти заносяться до БД, сповіщення відправляються).

```
import smtplib
from email.message import EmailMessage
from typing import Dict

def send_email_notification(subject: str, body: str, to_addr: str) -> None:
    """Надсилає сповіщення електронною поштою про інцидент."""
    msg = EmailMessage()
    msg["Subject"] = subject
    msg["From"] = "monitoring@example.com"
    msg["To"] = to_addr
    msg.set_content(body)

    with smtplib.SMTP("smtp.example.com", 587) as smtp:
        smtp.starttls()
        smtp.login("monitoring@example.com", "password-placeholder")
        smtp.send_message(msg)

def classify_state(health_score: float) -> str:
    """Повертає текстовий стан на основі значення health_score."""
    if health_score >= 0.8:
        return "up"
    if health_score >= 0.5:
        return "degraded"
    return "down"

def process_incidents(node_states: Dict[str, float], db_conn: sqlite3.Connection) -> None:
    """Фіксує інциденти при зміні стану вузлів та надсилає e-mail сповіщення."""
    cur = db_conn.cursor()
    ts = int(time.time())

    for node, h_score in node_states.items():
        new_state = classify_state(h_score)

        cur.execute(
            "SELECT last_state FROM node_status WHERE node_name = ?",
            (node,),
        )
        row = cur.fetchone()
        old_state = row[0] if row else "unknown"

        if old_state != new_state:
            cur.execute(
                """INSERT INTO incidents(node_name, ts, old_state, new_state, health_score)
                       VALUES (?, ?, ?, ?, ?)""",
                (node, ts, old_state, new_state, h_score),
            )
            cur.execute(
                "REPLACE INTO node_status(node_name, last_state, last_update) "
                "VALUES (?, ?, ?)",
                (node, new_state, ts),
            )

            subject = f"Інцидент: вузол {node} перейшов у стан {new_state}"
```

```
        body = (
            f"Вузол: {node}\n"
            f"Попередній стан: {old_state}\n"
            f"Новий стан: {new_state}\n"
            f"health_score={h_score:.2f}\n"
        )
        send_email_notification(subject, body, "admin@example.com"))

db_conn.commit()
```