

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

Навчально-науковий інститут телекомунікаційних систем

Кафедра інформаційних технологій в телекомунікаціях

До захисту допущено:

В. о. завідувача кафедрою

_____ Валерій ПРАВИЛО

« ____ » _____ 2026 р.

ДИПЛОМНА РОБОТА
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційно-комунікаційні
технології»
спеціальності 172 Телекомунікації та радіотехніка

на тему: Метод оцінювання екологічної та енергетичної
ефективності телекомунікаційних мереж

Виконала: здобувачка вищої освіти 4 курсу, групи ТІ-21
Старосельська Ольга Едуардівна _____

Науковий керівник: доцент кафедри ІТТ НН ІТС,

кандидат технічних наук, доцент

Суліма Світлана Валеріївна _____

Рецензент: доцент кафедри ТК НН ІТС,

кандидат технічних наук, доцент

Міночкін Дмитро Анатолійович _____

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

Здобувач вищої освіти _____

Київ – 2026 року

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

Рівень вищої освіти – перший (бакалаврський)
 Освітньо-професійна програма «Інформаційно-комунікаційні технології»
 спеціальності 172 Телекомунікації та радіотехніка

ЗАТВЕРДЖУЮ

В.о. завідувача кафедрою

_____ Валерій ПРАВИЛО

«___» _____ 2026 р

ЗАВДАННЯ
на дипломну роботу студентці
Старосельській Ользі Едуардівні

1. Тема дипломної роботи: Метод оцінювання екологічної та енергетичної ефективності телекомунікаційних мереж, науковий керівник роботи доцент кафедри ІТТ НН ІТС, кандидат технічних наук, доцент Суліма Світлана Валеріївна - затверджені наказом по університету від 26.05.2026 р. № 1912-с
2. Строк подання студентом дипломної роботи 06.06.2026 р.
3. Об'єкт дослідження: процеси споживання електроенергії та формування вуглецевого сліду обладнанням телекомунікаційних мереж.
4. Вихідні дані до роботи: методи, моделі та алгоритми оцінювання й підвищення енергетичної та екологічної ефективності телекомунікаційного вузла.
5. Перелік завдань, які потрібно розробити:

Проаналізувати телекомунікаційні мережі як споживачів електроенергії, їхній внесок у глобальне енергоспоживання та викиди парникових газів; Систематизувати показники й метрики енергетичної та екологічної ефективності та критично проаналізувати наявні методи її підвищення; Сформулювати вимоги до методу та розробити математичну модель енергоспоживання й екологічної ефективності телекомунікаційного вузла; Розробити алгоритм аналізу режиму роботи вузла та алгоритм оптимізації його енергоспоживання з дотриманням показників якості обслуговування; Виконати програмну реалізацію методу у вигляді вебзастосунку з базою даних та модулем симуляції телеметрії; Провести експериментальне дослідження і оцінити отриману економію енергії та зменшення вуглецевого сліду;

Опрацювати питання техніко-економічного обґрунтування, екологічної безпеки.

6. Перелік ілюстративного матеріалу: рисунки, таблиці, презентаційні матеріали до захисту.

7. Дата видачі завдання: 7.10.2025 р

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Узгодження організаційних питань з науковим керівником	10.09.25–12.09.25	Виконано
2	Узгодження теми роботи та початок роботи над першим розділом	12.09.25–7.10.25	Виконано
3	Робота над першим, другим та третім розділами	7.10.25–10.01.26	Виконано
4	Робота над четвертим розділом. Планування розробки власного рішення	10.01.26–15.02.26	Виконано
5	Оформлення дипломної роботи	15.02.26–01.06.26	Виконано
6	Підготовка презентації до захисту	22.05.26–06.06.26	Виконано

Здобувачка вищої освіти _____ Ольга СТАРОСЕЛЬСЬКА

Науковий керівник роботи _____ Світлана СУЛІМА

РЕФЕРАТ

Дипломна робота «Метод оцінювання екологічної та енергетичної ефективності телекомунікаційних мереж» містить 94 сторінки тексту, 15 рисунків, 8 таблиць, 6 додатків, а також використовує 26 літературних джерел посилання.

Актуальність теми: Через швидке зростання обсягів цифрового трафіку, поширення хмарних сервісів, інтернету речей та мереж нового покоління, суттєво збільшується енергоспоживання інформаційно-комунікаційної інфраструктури.

Телекомунікаційний вузол є типовим елементом мережі, енергоспоживання якого визначається не лише корисним навантаженням, а й значною часткою постійних витрат на підтримання обладнання в активному стані та на його охолодження. Через нерівномірність трафіку впродовж доби обладнання тривалий час працює зі значним недовантаженням, проте споживає майже стільки ж енергії, як і за пікового навантаження. Це створює об'єктивні передумови для підвищення енергоефективності без зниження якості обслуговування, що й зумовлює актуальність теми дослідження.

Метою роботи є підвищення енергетичної та екологічної ефективності телекомунікаційних мереж шляхом розроблення методу оцінювання і оптимізації режимів роботи вузла та його програмної реалізації.

Об'єктом дослідження є процеси споживання електроенергії та формування вуглецевого сліду обладнанням телекомунікаційних мереж.

Предметом дослідження є методи, моделі та алгоритми оцінювання й підвищення енергетичної та екологічної ефективності телекомунікаційного вузла.

Методами дослідження є системний аналіз, методи математичного моделювання, методи комбінаторної оптимізації, методи теорії масового обслуговування та обчислювальний експеримент.

Результати роботи можуть бути використані операторами електронних комунікацій та проєктувальниками телекомунікаційних мереж для оцінювання й підвищення енергоефективності вузлів.

Ключові слова: ТЕЛЕКОМУНІКАЦІЙНА МЕРЕЖА, ЕНЕРГОЕФЕКТИВНІСТЬ, ВУГЛЕЦЕВИЙ СЛІД, КОЕФІЦІЄНТ PUE, ЕНЕРГОСПОЖИВАННЯ, ОПТИМІЗАЦІЯ, КОНСОЛІДАЦІЯ ПОРТІВ, ОХОЛОДЖЕННЯ, ЯКІСТЬ ОБСЛУГОВУВАННЯ, SNMP.

ABSTRACT

The master's thesis Метод оцінювання екологічної та енергетичної ефективності телекомунікаційних мереж contains 94 pages, 15 figures, 8 tables, 26 references.

Relevance of the topic: Due to the rapid growth of digital traffic volumes, the proliferation of cloud services, the Internet of Things, and next-generation networks, the energy consumption of information and communication infrastructure is increasing significantly.

A telecommunication node is a typical network element whose energy consumption is determined not only by the traffic load but also by a substantial share of constant energy expenditures required to maintain the equipment in an active state and to cool it. Due to the uneven distribution of traffic throughout the day, the equipment operates significantly underutilized for extended periods, yet it consumes almost as much energy as it does during peak loads. This creates objective prerequisites for improving energy efficiency without compromising the Quality of Service, which justifies the relevance of this research topic.

The objective of this work is to improve the energy and environmental efficiency of telecommunication networks through the development of a method for evaluating and optimizing node operating modes, as well as its software implementation.

The object of the study is the processes of electricity consumption and carbon footprint formation by telecommunication network equipment.

The subject of the study is the methods, models and algorithms for assessing and improving the energy and ecological efficiency of a telecommunication node.

Research methods include systems analysis, mathematical modeling methods, combinatorial optimization methods, queueing theory methods, and a computational experiment..

The results of the work can be used by electronic communications operators and telecommunication network designers for assessing and improving the energy efficiency of nodes.

Keywords: TELECOMMUNICATION NETWORK, ENERGY EFFICIENCY, CARBON FOOTPRINT, PUE, ENERGY CONSUMPTION, OPTIMISATION, PORT CONSOLIDATION, COOLING, QUALITY OF SERVICE, SNMP.

ЗМІСТ

СПИСОК ТЕРМІНІВ І СКОРОЧЕНЬ	10
ВСТУП	11
РОЗДІЛ 1. АНАЛІЗ ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖ ТА МЕТОДІВ ОЦІНЮВАННЯ ЇХНЬОЇ ЕНЕРГЕТИЧНОЇ І ЕКОЛОГІЧНОЇ ЕФЕКТИВНОСТІ	13
1.1 Телекомунікаційні мережі як споживачі електроенергії та джерело викидів парникових газів	14
1.2 Структура енергоспоживання телекомунікаційного вузла	16
1.3 Показники та метрики енергетичної і екологічної ефективності	18
1.4 Огляд та критичний аналіз методів підвищення енергоефективності	21
1.5 Постановка задачі та обґрунтування напряду дослідження.....	24
Висновки розділу 1	27
РОЗДІЛ 2. РОЗРОБЛЕННЯ МЕТОДУ ОЦІНЮВАННЯ ТА ОПТИМІЗАЦІЇ ЕНЕРГЕТИЧНОЇ І ЕКОЛОГІЧНОЇ ЕФЕКТИВНОСТІ ТЕЛЕКОМУНІКАЦІЙНОГО ВУЗЛА.....	29
2.1 Вимоги до методу оцінювання	29
2.2 Математична модель енергоспоживання телекомунікаційного вузла	30
2.3 Модель екологічної ефективності та вуглецевого сліду.....	32
2.4 Алгоритм аналізу режиму роботи вузла	33
2.5 Алгоритм оптимізації енергоспоживання	33
2.6 Вибір програмних засобів реалізації.....	35
2.7 Приклад застосування моделі	36
Висновки розділу 2	38
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ	40
3.1 Модель телекомунікаційного вузла	40
3.2 Програмна реалізація системи	41
3.3 Модуль симуляції SNMP-телеметрії.....	45
3.4 Організація експериментального дослідження.....	47
3.5 Результати оцінювання енергоспоживання та вуглецевого сліду	48
3.6 Результати оптимізації енергоспоживання	49
Висновки розділу 3	56

РОЗДІЛ 4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ, ЕКОЛОГІЧНІ ЗАХОДИ.....	57
4.1 Техніко-економічне обґрунтування	57
4.2 Екологічні заходи.....	62
Висновки розділу 4	63
ЗАГАЛЬНІ ВИСНОВКИ	65
ПЕРЕЛІК	67
ДОДАТОК А.....	72
ДОДАТОК Б	79
ДОДАТОК В.....	84
ДОДАТОК Г	87
ДОДАТОК Д.....	91
ДОДАТОК Е	93

СПИСОК ТЕРМІНІВ І СКОРОЧЕНЬ

ІКТ – інформаційно-комунікаційні технології

МО – масове обслуговування

СКБД – система керування базами даних

ALR – Adaptive Link Rate, адаптація швидкості каналу

CO₂ – діоксид вуглецю

COP – Coefficient of Performance, коефіцієнт перетворення холодильної системи

EE – Energy Efficiency, енергоефективність (продуктивність на одиницю потужності)

EEE – Energy-Efficient Ethernet (IEEE 802.3az)

FFD – First-Fit Decreasing, метод пакування за спаданням

MIB – Management Information Base, база керувальної інформації

PUE – Power Usage Effectiveness, коефіцієнт ефективності використання енергії

QoS – Quality of Service, якість обслуговування

SNMP – Simple Network Management Protocol

ВСТУП

За останні роки обсяги цифрового графіку стрімко зросли, через це суттєво збільшилось енергоспоживання інформаційно-комунікаційної інфраструктури. За оцінками Міжнародного союзу електрозв'язку та Світового банку, на сектор інформаційно-комунікаційних технологій припадає від 1,5 до 4 % світових викидів парникових газів, а споживання електроенергії телекомунікаційними операторами впродовж останніх років стабільно зростає [1; 2]. Для України проблема енергоефективності телекомунікаційних мереж набуває особливої гостроти через дефіцит генерувальних потужностей та потребу забезпечення сталого функціонування інфраструктури зв'язку [3].

Типовий елемент телекомунікаційної мережі, вузол, споживає електроенергію не лише пропорційно до корисного навантаження, а й несе значну частку постійних витрат на підтримання обладнання в активному стані та на його охолодження. Оскільки інтенсивність трафіку протягом доби суттєво коливається, обладнання довгий час працює з великим недовантаженням, споживаючи при цьому майже стільки ж енергії, що й у піковому режимі. Це формує реальні передумови для підвищення енергоефективності без погіршення якості обслуговування, що й зумовлює актуальність теми дослідження.

Метою дипломної роботи є підвищення енергетичної та екологічної ефективності телекомунікаційних мереж шляхом розроблення методу оцінювання й оптимізації режимів роботи вузла та його програмної реалізації з експериментальним підтвердженням досягнутого ефекту.

Об'єктом дослідження є процеси споживання електроенергії та формування вуглецевого сліду обладнанням телекомунікаційних мереж.

Предметом дослідження є методи, моделі та алгоритми оцінювання й підвищення енергетичної та екологічної ефективності телекомунікаційного вузла.

Методами дослідження є системний аналіз для декомпозиції об'єкта, методи математичного моделювання для опису енергоспоживання, теплового

режиму та вуглецевого сліду, методи комбінаторної оптимізації (пакування у контейнери, локальний пошук) для побудови алгоритму енергозбереження, методи теорії масового обслуговування для оцінювання показників якості, обчислювальний експеримент для перевірки ефективності.

У роботі систематизовано показники енергетичної та екологічної ефективності телекомунікаційних мереж, обґрунтовано математичну модель енергоспоживання вузла, що враховує потужність портів, систему охолодження та коефіцієнт використання енергії, а також модель вуглецевого сліду. Розроблено алгоритм оптимізації, що поєднує консолідацію навантаження на мінімальну кількість активних портів, адаптацію швидкості каналів та ступінчасте керування охолодженням за умови дотримання обмежень якості обслуговування. Метод реалізовано у вигляді вебзастосунку на основі Flask і PostgreSQL із модулем симуляції SNMP-телеметрії. За результатами обчислювального експерименту досягнуто зменшення добового енергоспоживання модельного вузла на 37,3 % та відповідного скорочення вуглецевого сліду за дотримання якості обслуговування.

Результати роботи можуть бути використані операторами електронних комунікацій та проєктувальниками телекомунікаційних мереж для оцінювання й підвищення енергоефективності вузлів.

Елементом новизни є комплексний метод оптимізації енергоспоживання телекомунікаційного вузла, який, на відміну від існуючих підходів, вперше поєднує консолідацію навантаження на мінімальній кількості портів, адаптацію швидкості каналів передачі даних та ступінчасте керування системою охолодження з обов'язковим контролем показників якості обслуговування та теплового режиму обладнання.

Практичне значення одержаних результатів полягає в тому, що розроблений метод і програмний засіб дають змогу кількісно оцінювати енергетичну та екологічну ефективність телекомунікаційного вузла і визначати раціональні режими його роботи, що забезпечують зменшення енергоспоживання та вуглецевого сліду без погіршення якості обслуговування.

Дипломна робота має 94 сторінки тексту, 15 рисунків та 8 таблиць, 6 додатків, а також використовує 26 літературних джерел посилання.

.

РОЗДІЛ 1

АНАЛІЗ ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖ ТА МЕТОДІВ ОЦІНЮВАННЯ ЇХНЬОЇ ЕНЕРГЕТИЧНОЇ І ЕКОЛОГІЧНОЇ ЕФЕКТИВНОСТІ

1.1 Телекомунікаційні мережі як споживачі електроенергії та джерело викидів парникових газів

Сучасні телекомунікаційні мережі є невіддільною частиною критичної інфраструктури суспільства, проте водночас вони належать до енергоємних технічних систем. Сектор інформаційно-комунікаційних технологій споживає помітну частку світової електроенергії та формує відповідні викиди парникових газів. За зведеними оцінками Міжнародного союзу електрозв'язку та Світового банку, частка цього сектору у глобальних викидах коливається в межах від 1,5 до 4 % залежно від меж обліку та методики розрахунку [1]. Така розбіжність оцінок пояснюється тим, що різні дослідження по-різному визначають межі системи, набори вихідних даних та підходи до врахування непрямих викидів.

За даними галузевих звітів, споживання електроенергії на стадії експлуатації обладнання сектору становить близько тисячі тераватт-годин на рік і продовжує зростати темпом приблизно 2–3 % щороку, хоча завдяки збільшенню частки відновлюваної генерації сукупні викиди зростають повільніше [2]. Окремо зазначається, що споживання електроенергії саме телекомунікаційними операторами впродовж 2020–2022 років зросло приблизно на 7 % і перевищило двісті п'ятдесят тераватт-годин [4]. Ці тенденції свідчать про те, що енергоефективність мереж зв'язку перетворюється з другорядного експлуатаційного питання на стратегічний чинник сталого розвитку галузі.

За функціональним призначенням телекомунікаційну мережу прийнято поділяти на магістральний (транспортний) рівень, рівень агрегації та рівень доступу. Найбільшу частку сукупного енергоспоживання припадає саме на рівень доступу та агрегації, оскільки він представлений численними

розподіленими вузлами, які працюють цілодобово [5]. Типовий вузол агрегації містить комутаційне обладнання, систему електроживлення з резервуванням та систему охолодження, причому остання може споживати від третини до половини всієї електроенергії майданчика [6].

Для України проблема енергоефективності телекомунікаційних мереж набула особливої гостроти. В умовах дефіциту генерувальних потужностей та пошкодження енергетичної інфраструктури підвищення енергоефективності вузлів зв'язку є не лише економічним, а й питанням функціональної стійкості мережі [3]. Тому розроблення методів кількісного оцінювання та зниження енергоспоживання телекомунікаційного обладнання є актуальним науково-прикладним завданням.

Структурно мережа є ієрархічною, і кількість вузлів зростає від магістрального рівня до рівня доступу на кілька порядків. Через це сукупне енергоспоживання визначається переважно численними вузлами доступу та агрегації, кожен з яких окремо споживає помірну потужність, проте у масштабі мережі їхній внесок є визначальним. Відповідно, навіть незначне відносне зниження споживання одного типового вузла, помножене на кількість таких вузлів, дає значний абсолютний ефект на рівні оператора та галузі.

Зростання енергоспоживання мереж зумовлене кількома чинниками попиту. Поширення відеосервісів з високою роздільною здатністю, хмарних обчислень, інтернету речей та впровадження мереж п'ятого покоління збільшують як обсяги трафіку, так і щільність розгорнутого обладнання [3, 7]. Унаслідок цього питання енергоефективності не втрачає актуальності з появою нових поколінь технологій, а навпаки, загострюється, оскільки приріст продуктивності частково нівелюється збільшенням сукупного споживання.

Під час оцінювання важливо чітко визначати межі обліку. Розрізняють пряме споживання обладнання на стадії експлуатації та споживання за повним життєвим циклом, що додатково охоплює виробництво, транспортування та утилізацію обладнання [1]. Оскільки предметом цієї роботи є режими роботи телекомунікаційного вузла, надалі увагу зосереджено на стадії експлуатації, для

якої характерний найбільший потенціал оперативного зниження енергоспоживання засобами адаптивного керування.

Таблиця 1.1 – Оцінки енергоспоживання та викидів сектору ІКТ

Показник	Значення	Джерело
Частка сектору ІКТ у світових викидах парникових газів	1,5–4 %	[1]
Споживання електроенергії сектором ІКТ (стадія експлуатації)	близько 1000 ТВт·год/рік	[2]
Річний темп зростання споживання електроенергії ІКТ	2–3 %	[2]
Споживання електроенергії телеком-операторами (2022)	понад 250 ТВт·год	[4]
Частка охолодження в енергоспоживанні майданчика	30–50 %	[6]

Джерело: сформовано автором за даними [2, 1, 6, 4]

Наведені у таблиці 1.1 дані підтверджують, що енергоспоживання телекомунікаційної інфраструктури є значним і зростає, а суттєву його частку становлять непродуктивні витрати на підтримання обладнання в активному стані та на охолодження. Це визначає два основні напрями підвищення енергоефективності, які розглядаються далі: зменшення споживання самого комутаційного обладнання та оптимізація системи охолодження.

1.2 Структура енергоспоживання телекомунікаційного вузла

Енергоспоживання телекомунікаційного вузла визначається сукупністю складових, лише частина з яких безпосередньо залежить від обсягу переданого трафіку. Можна виокремити постійну складову, що відповідає споживанню шасі, площини керування та портів у стані очікування, змінну складову, яка залежить від завантаження портів, та складову системи охолодження, що залежить від тепловиділення обладнання та температури навколишнього середовища [8, 6]. Особливістю сучасного комутаційного обладнання є висока питома вага постійної складової, через що навіть за незначного трафіку вузол споживає потужність, близьку до споживання за пікового навантаження.

Характер трафіку телекомунікаційного вузла є суттєво нерівномірним протягом доби. Періоди нічного спаду навантаження чергуються з ранковим та вечірнім піками, причому співвідношення між мінімальним та максимальним завантаженням може сягати кратних значень. За таких умов значна частина портів тривалий час залишається недовантаженою, проте перебуває у повністю активному стані. Саме ця невідповідність між фактичним навантаженням та спожитою потужністю формує резерв енергозбереження, який можна реалізувати засобами адаптивного керування [5, 9].

Система охолодження є другим за значущістю споживачем електроенергії на майданчику. Її енергоспоживання нелінійно залежить від рівня тепловиділення обладнання та від обраного режиму роботи. За зниження тепловиділення з'являється можливість зменшити інтенсивність примусового охолодження або повніше використовувати режим вільного охолодження зовнішнім повітрям, що дає значний виграш в енергоспоживанні [6]. Отже, зменшення споживання комутаційного обладнання має додатковий, опосередкований ефект через зниження навантаження на систему охолодження.

Кількісно невідповідність між споживанням і навантаженням можна охарактеризувати через відношення потужності у стані спокою до потужності за повного завантаження. Для типового комутаційного обладнання потужність порту у стані очікування становить помітну частку від його максимальної потужності, а з урахуванням постійного споживання шасі загальна потужність вузла за мінімального трафіку лишається близькою до максимальної. Саме тому пасивне очікування зниження споживання від самого обладнання є недостатнім, і потрібні активні методи керування, які цілеспрямовано переводять незадіяні ресурси у стан зниженого споживання та узгоджують роботу охолодження з фактичним тепловиділенням.

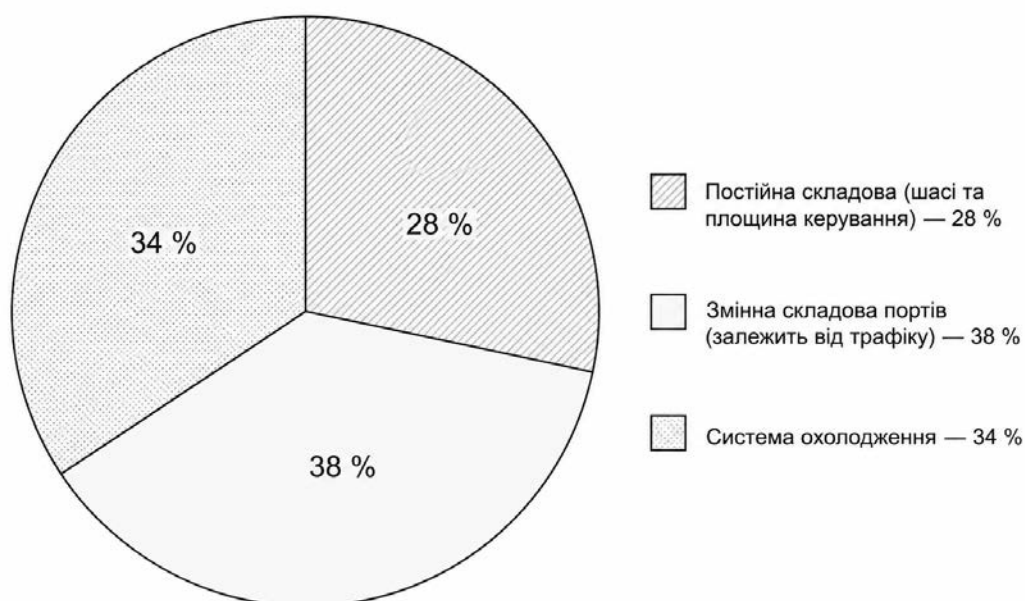


Рисунок 1.1 – Структура енергоспоживання телекомунікаційного вузла

Джерело: побудовано автором за [8, 6]

Поданий на рисунку 1.1 розподіл ілюструє, що значну частину енергоспоживання вузла становлять складові, слабо пов'язані з корисним трафіком. Це обґрунтовує доцільність застосування методів, які переводять незадіяне обладнання в режим зниженого споживання та узгоджують роботу системи охолодження з фактичним тепловиділенням.

1.3 Показники та метрики енергетичної і екологічної ефективності

Для кількісного оцінювання енергетичної та екологічної ефективності телекомунікаційного обладнання застосовують низку нормативно визначених показників. Найпоширенішим показником ефективності використання енергії на майданчику є коефіцієнт PUE (Power Usage Effectiveness), стандартизований у ISO/IEC 30134-2 та визначений як відношення повної спожитої майданчиком потужності до потужності, що безпосередньо живить інформаційно-комунікаційне обладнання [10]. Ідеальне значення PUE дорівнює одиниці, тоді

як середньогалузеві значення, за даними щорічних оглядів, перебувають у межах приблизно 1,5–1,6 [11].

Енергетичну ефективність власне телекомунікаційного обладнання оцінюють за метриками, визначеними в Рекомендації ITU-T L.1310 та в стандарті ETSI ES 203 228. Зокрема, для обладнання вводять показник енергоефективності як відношення корисної продуктивності (пропускної здатності) до спожитої потужності, а вимірювання виконують для кількох рівнів завантаження з подальшим зважуванням за часом їхньої тривалості [12, 8]. Такий підхід дає змогу враховувати реальний добовий профіль роботи обладнання, а не лише режим максимального навантаження.

Екологічну ефективність характеризують через вуглецевий слід, тобто масу викидів діоксиду вуглецю, еквівалентну спожитій електроенергії. Її обчислюють як добуток спожитої енергії на коефіцієнт викидів електромережі, що відображає вуглецеву інтенсивність генерації в конкретній енергосистемі [1]. Оскільки коефіцієнт викидів істотно різниться між країнами, той самий рівень енергоспоживання може відповідати різному вуглецевому сліду, тому під час оцінювання важливо застосовувати коректне для відповідної енергосистеми значення.

Поряд з коефіцієнтом PUE застосовують і похідні показники інженерної ефективності майданчика, зокрема обернену до нього ефективність інфраструктури. Концептуально важливою є енергетична пропорційність, згідно з якою споживана потужність обладнання має бути пропорційною фактичному навантаженню. Реальне комутаційне обладнання далеке від ідеалу пропорційності, оскільки значну частку його споживання становить постійна складова, тому одним із завдань методів енергозбереження є наближення поведінки вузла до пропорційної [9, 13].

Екологічний вимір ефективності оцінюють не лише за абсолютним вуглецевим слідом, а й за інтенсивністю викидів, тобто за масою викидів на одиницю переданої інформації або спожитої енергії. Оскільки коефіцієнт викидів електромережі істотно різниться між енергосистемами і навіть

змінюється протягом доби залежно від структури генерації, той самий рівень енергоспоживання може відповідати різному вуглецевому сліду [14, 2]. Це означає, що екологічний ефект енергоощадних заходів залежить також від місця та часу їх застосування.

Зіставлення наведених показників свідчить, що жоден з них окремо не дає вичерпної характеристики ефективності вузла: коефіцієнт PUE не враховує ефективності самого обладнання, показник енергоефективності не відображає накладних витрат на охолодження, а вуглецевий слід залежить від зовнішнього чинника, тобто структури генерації. Тому в розробленому методі ці показники застосовано спільно, що дає змогу оцінювати ефективність вузла комплексно та узгоджено.

Таблиця 1.2 – Показники енергетичної та екологічної ефективності

Показник	Зміст показника	Джерело
PUE	відношення повної потужності майданчика до потужності ІКТ-обладнання; характеризує накладні витрати, насамперед на охолодження	[10]
EE (EER)	відношення корисної продуктивності до спожитої потужності за кількох рівнів завантаження	[12, 8]
Вуглецевий слід	маса викидів CO ₂ (добуток енергії на коефіцієнт викидів мережі)	[1]

Джерело: узагальнено автором за [12, 10, 8, 1]

Показники, наведені в таблиці 1.2, характеризують різні аспекти ефективності: PUE відображає досконалість інженерної інфраструктури майданчика, показник EE – ефективність самого обладнання, а вуглецевий слід – екологічний вимір енергоспоживання. У запропонованому далі методі ці показники використано спільно, що дає змогу оцінювати ефективність вузла комплексно.

1.4 Огляд та критичний аналіз методів підвищення енергоефективності

Методи підвищення енергоефективності телекомунікаційних мереж умовно поділяють на апаратні, що реалізуються вбудованими механізмами обладнання, та програмно-алгоритмічні, що реалізуються засобами керування. До апаратних належить технологія енергоефективного Ethernet (стандарт IEEE 802.3az), яка переводить порт у стан зниженого споживання у періоди відсутності трафіку та узгоджує споживання з довжиною кабелю, що дає змогу зменшити споживання інтерфейсів за низької активності [15]. Перевагою цього підходу є прозорість для вищих рівнів та сумісність з наявним обладнанням, проте він діє локально на рівні окремого порту і не оптимізує роботу вузла в цілому.

Поширеним програмно-алгоритмічним підходом є консолідація трафіку, за якої навантаження концентрується на мінімально потрібній кількості активних елементів, а незадіяні порти або пристрої переводяться у режим сну [5, 9]. Близьким за ідеєю є метод адаптації швидкості каналу, коли за низького трафіку швидкість порту знижують, зменшуючи його енергоспоживання. Ці методи здатні забезпечити істотну економію за умови коректного врахування обмежень якості обслуговування, інакше надмірна консолідація призводить до перевантаження активних портів і зростання затримок.

Окремий напрям утворюють методи оптимізації системи охолодження, зокрема ширше використання вільного охолодження та узгодження інтенсивності охолодження з фактичним тепловиділенням [6]. З огляду на значну частку охолодження в енергоспоживанні майданчика такі методи дають вагомий ефект, проте їхня результативність залежить від кліматичних умов та конструкції майданчика. Перспективним вважають застосування методів машинного навчання для прогнозування трафіку та випереджального керування ресурсами: у відповідних дослідженнях продемонстровано можливість зменшення енергоспоживання за рахунок точнішого передбачення

навантаження [16, 3, 17]. Водночас такі методи потребують значних обсягів даних для навчання і ускладнюють реалізацію та супровід системи.

Окремий напрям підвищення енергоефективності пов'язаний з програмно-конфігурованими мережами, які дають змогу централізовано керувати маршрутизацією трафіку. Завдяки відокремленню площини керування від площини передавання такі мережі уможливають динамічну концентрацію потоків на обмеженій множині елементів і вимкнення незадіяних пристроїв на рівні всієї мережі, а не лише окремого вузла [9]. Перевагою є глобальна оптимізація, проте реалізація потребує відповідної інфраструктури керування та погодженості з вимогами надійності.

Підвищенню екологічної ефективності сприяє також інтеграція відновлюваних джерел енергії та систем накопичення на майданчиках зв'язку. Такий підхід зменшує залежність від мережі та знижує вуглецевий слід за рахунок використання енергії з нижчою вуглецевою інтенсивністю, що особливо актуально для України в умовах дефіциту генерувальних потужностей [9, 1]. Водночас ефективність цього напрямку залежить від доступності відновлюваних ресурсів і вартості систем накопичення.

Спільним обмеженням динамічних методів енергозбереження є накладні витрати на перемикання станів обладнання та потенційне погіршення показників якості обслуговування за надмірно агресивної оптимізації. Перемикання портів між активним станом і станом сну, а також зміна швидкості каналу потребують часу і можуть спричиняти короточасні втрати продуктивності. Тому будь-який метод оптимізації має містити механізм контролю якості обслуговування, що запобігає перевищенню допустимої утилізації та затримки, і саме такий механізм передбачено в розробленому методі.

Таблиця 1.3 – Порівняння методів підвищення енергоефективності

Метод	Переваги	Обмеження
Енергоефективний Ethernet (EEE),	прозорість, сумісність з наявним обладнанням	діє локально на рівні порту, не оптимізує вузол загалом

Метод	Переваги	Обмеження
802.3az		[15]
Консолідація трафіку та режим сну портів	значна економія за недовантаження	ризик перевантаження активних портів, потреба контролю QoS [5, 9]
Адаптація швидкості каналу (ALR)	зменшення споживання порту за низького трафіку	обмежений діапазон виграшу, накладні витрати на перемикання
Оптимізація та вільне охолодження	вагомий ефект через велику частку охолодження	залежність від клімату та конструкції майданчика [6]
Машинне навчання для прогнозу трафіку	випереджальне керування, точність	потреба у даних, складність реалізації [3, 17]

Джерело: складено автором за [3, 5, 9, 15, 17, 6]

Порівняння, наведене в таблиці 1.3, показує, що жоден із методів окремо не забезпечує комплексного розв'язання задачі. Найбільший практичний ефект досягається поєднанням консолідації навантаження, адаптації швидкості та керування охолодженням за обов'язкового контролю якості обслуговування. Саме таке поєднання покладено в основу методу, що розробляється в другому розділі.

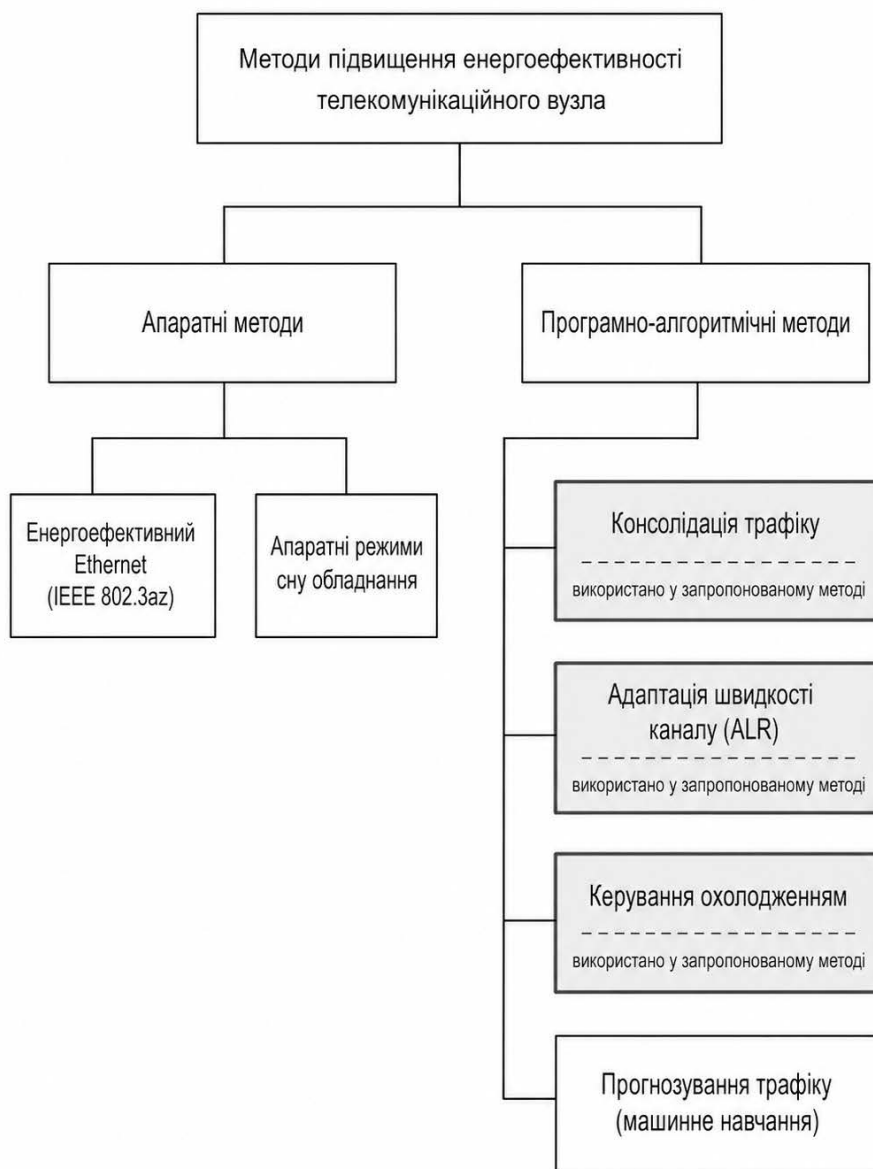


Рисунок 1.2 – Класифікація методів підвищення енергоефективності телекомунікаційного вузла

Джерело: побудовано автором

1.5 Постановка задачі та обґрунтування напрямку дослідження

Аналіз наявних робіт свідчить, що питання оцінювання та підвищення енергоефективності телекомунікаційних мереж досліджено досить ґрунтовно, проте переважно у двох відносно відокремлених площинах. Перша охоплює стандартизацію показників та методів вимірювання ефективності обладнання [12, 10, 8], друга – окремі методи енергозбереження [5, 9, 6]. У дисертаційному

дослідженні [18], виконаному в спорідненій предметній галузі, обґрунтовано комплексний підхід до енергоефективного обслуговування навантаження інформаційно-комунікаційної мережі, що підтверджує доцільність поєднання оцінювання й оптимізації в межах єдиного методу.

Разом з тим бракує доступного інструментального методу, який поєднував би кількісне оцінювання енергетичної та екологічної ефективності телекомунікаційного вузла з алгоритмом оптимізації його режиму роботи за умови дотримання якості обслуговування і був би доведений до програмної реалізації з можливістю відтворення результатів. Заповнення цієї прогалини і є напрямом цього дослідження.

Виходячи з викладеного, задачу дослідження формулюємо так: для заданої конфігурації телекомунікаційного вузла та профілю його завантаження побудувати математичну модель енергоспоживання й вуглецевого сліду, розробити алгоритм визначення раціонального режиму роботи, що мінімізує спожиту потужність за умови неперевищення граничної утилізації портів і граничної температури обладнання, та реалізувати метод у вигляді програмного засобу з підтвердженням ефективності обчислювальним експериментом. Розв'язанню цієї задачі присвячено наступні розділи роботи.

Узагальнюючи результати огляду, можна стверджувати, що ефективно розв'язання задачі підвищення енергоефективності телекомунікаційного вузла потребує методу, який об'єднує три складові: кількісне оцінювання енергетичної та екологічної ефективності за узгодженими показниками, алгоритм визначення раціонального режиму роботи та механізм контролю якості обслуговування. Відсутність будь-якої зі складових знижує практичну цінність методу: оцінювання без оптимізації не дає керувальних рішень, оптимізація без контролю якості може погіршити характеристики мережі, а відсутність відтворюваної програмної реалізації унеможливорює перевірку результатів. Саме на побудові методу, що поєднує всі ці складові, і зосереджено подальше дослідження.

Окремої уваги потребує аспект функціональної стійкості та енергетичної автономності телекомунікаційних мереж, що набув особливого значення для України. В умовах нестабільного електропостачання вузли зв'язку дедалі частіше живляться від автономних джерел та акумуляторних комплексів, для яких енергоефективність обладнання безпосередньо визначає тривалість автономної роботи. Застосування методів інтелектуального моніторингу та керування дає змогу не лише знижувати споживання, а й підвищувати стійкість інфраструктури до перебоїв живлення [19, 3].

Важливою методологічною особливістю оцінювання енергоефективності є врахування реального профілю завантаження. Згідно з Рекомендацією ITU-T L.1310 та стандартом ETSI ES 203 228, вимірювання виконують для кількох рівнів навантаження, низького, середнього та пікового, з подальшим зважуванням за тривалістю їх дії протягом доби [8, 12]. Такий підхід запобігає завищенню оцінок ефективності, отриманих лише за максимального навантаження, і узгоджується з прийнятою в роботі моделлю, що оперує добовим профілем трафіку.

Перспективним доповненням до розглянутих методів є інтеграція відновлюваних джерел енергії та систем накопичення, що зменшує вуглецеву інтенсивність живлення вузла. Проте через стохастичний характер генерації такі рішення потребують додаткових засобів керування і не усувають потреби в підвищенні власної енергоефективності обладнання [9]. Тому в межах цієї роботи основну увагу зосереджено саме на зниженні енергоспоживання вузла засобами адаптивного керування, що є універсальним і не залежить від джерела живлення.

Структура енергоспоживання вузла безпосередньо впливає на вибір стратегії енергозбереження. Оскільки постійна складова не залежить від трафіку, найбільший резерв зосереджено у змінній складовій портів та в системі охолодження, керування якими не потребує втручання в апаратну частину обладнання. Це робить програмно-алгоритмічні методи особливо

привабливими для впровадження на наявній інфраструктурі без додаткових капітальних витрат.

Серед програмно-алгоритмічних підходів окреме місце посідають методи програмно-конфігурованих мереж, що дають змогу централізовано керувати маршрутизацією трафіку та консолідувати його на обмеженій підмножині активних елементів. Поєднання таких методів з механізмами енергоефективного Ethernet і керуванням охолодженням розглядається в літературі як найбільш перспективний напрям комплексного зниження енергоспоживання [9, 6].

Слід також зважати на компроміс між енергоефективністю та якістю обслуговування. Надмірна консолідація трафіку й зниження інтенсивності охолодження здатні погіршити показники затримки та підвищити температуру обладнання, тому будь-який метод оптимізації має містити явні обмеження якості обслуговування й теплового режиму. Цей принцип покладено в основу розробленого методу та конкретизовано у вигляді обмежень у другому розділі.

Висновки розділу 1

За результатами аналітичного огляду встановлено, що телекомунікаційні мережі є значними та зростаючими споживачами електроенергії, а їхній екологічний вплив, оцінюваний через вуглецевий слід, є вагомим чинником сталого розвитку галузі. Показано, що суттєву частку енергоспоживання телекомунікаційного вузла становлять складові, слабо пов'язані з корисним трафіком, насамперед постійне споживання обладнання та витрати на охолодження, що формує об'єктивний резерв енергозбереження.

Систематизовано показники енергетичної та екологічної ефективності, зокрема коефіцієнт PUE, показник енергоефективності обладнання за ITU-T L.1310 та ETSI ES 203 228 і вуглецевий слід. Виконано критичний аналіз методів підвищення енергоефективності та обґрунтовано, що найбільший ефект забезпечує поєднання консолідації навантаження, адаптації швидкості каналів і керування охолодженням за дотримання якості обслуговування. На цій підставі

сформульовано задачу дослідження, розв'язанню якої присвячено другий розділ.

РОЗДІЛ 2

РОЗРОБЛЕННЯ МЕТОДУ ОЦІНЮВАННЯ ТА ОПТИМІЗАЦІЇ ЕНЕРГЕТИЧНОЇ І ЕКОЛОГІЧНОЇ ЕФЕКТИВНОСТІ ТЕЛЕКОМУНІКАЦІЙНОГО ВУЗЛА

2.1 Вимоги до методу оцінювання

На підставі результатів аналітичного огляду сформульовано вимоги, яким має відповідати метод оцінювання екологічної та енергетичної ефективності телекомунікаційного вузла. Метод повинен забезпечувати кількісне оцінювання енергоспоживання вузла та його вуглецевого сліду на основі вимірюваних або змодельованих параметрів роботи обладнання, узгоджуватися з чинними показниками ефективності, насамперед з коефіцієнтом PUE та показником енергоефективності за ITU-T L.1310, а також урахувати реальний добовий профіль завантаження, а не лише режим максимального навантаження [10, 8].

Окрім оцінювання, метод має визначати раціональний режим роботи вузла, що мінімізує спожиту потужність за умови дотримання показників якості обслуговування. Для цього він повинен містити алгоритм аналізу поточного стану та алгоритм оптимізації, який не лише фіксує неефективні режими, а й обчислює конкретні керувальні дії. Важливою вимогою є відтворюваність результатів та можливість програмної реалізації методу з використанням засобів моніторингу, сумісних з наявною мережевою інфраструктурою, зокрема з протоколом SNMP [12].

Нарешті, метод має бути обчислювально нескладним, щоб його можна було застосовувати в реальному часі на рівні окремого вузла, і допускати масштабування на сукупність вузлів мережі. З огляду на ці вимоги обрано підхід, що поєднує аналітичну модель енергоспоживання, модель вуглецевого сліду та евристичний алгоритм оптимізації з перевіркою обмежень якості обслуговування.

2.2 Математична модель енергоспоживання телекомунікаційного вузла

Телекомунікаційний вузол подамо як сукупність із N портів номінальної пропускної здатності C (Гбіт/с), спільного шасі з площиною керування та системи охолодження. Повне енергоспоживання вузла складається з постійної складової шасі, змінної складової портів та складової системи охолодження.

Потужність окремого порту залежить від його стану та завантаження. Для активного порту, що працює на номінальній швидкості, потужність визначається лінійною залежністю від коефіцієнта утилізації:

$$P_{\text{порт},i} = P_{\text{хх}} + \rho_i \cdot (P_{\text{макс}} - P_{\text{хх}}) \quad (2.1)$$

де $P_{\text{хх}}$ – потужність порту у стані очікування (без трафіку), $P_{\text{макс}}$ – потужність порту за повного завантаження, ρ_i – коефіцієнт утилізації i -го порту, що обчислюється як

$$\rho_i = \frac{g_i}{C} \quad (2.2)$$

причому g_i – обсяг трафіку i -го порту (Гбіт/с). Якщо порт переведено в режим сну, його потужність дорівнює $P_{\text{сон}}$, а в разі зниження швидкості каналу до 1 Гбіт/с (адаптація швидкості) максимальна потужність порту зменшується до $P_{\text{макс},1G}$. Сумарна потужність інформаційно-комунікаційного обладнання вузла становить

$$P_{\text{ІТ}} = P_{\text{баз}} + \sum_{i=1}^N P_{\text{порт},i} \quad (2.3)$$

де $P_{\text{баз}}$ – постійна потужність шасі та площини керування. Тепловиділення обладнання приймаємо рівним його електричній потужності $P_{\text{ІТ}}$. Температура обладнання за поточного рівня примусового охолодження визначається співвідношенням

$$T_{\text{обл}} = T_{\text{вх}} + \frac{P_{\text{ІТ}} \cdot R_{\text{т}}}{u} \quad (2.4)$$

де $T_{\text{вх}}$ – температура повітря на вході, $R_{\text{т}}$ – тепловий опір, u – відносний рівень вентиляції (0...1). Електрична потужність системи охолодження враховує закон

подібності вентилятора, за яким споживана потужність пропорційна кубу частоти обертання, та можливе вмикання механічного охолодження:

$$P_{\text{охол}} = P_{\text{вент}} \cdot u^3 + P_{\text{хол}} \cdot \delta \quad (2.5)$$

де $P_{\text{вент}}$ – максимальна потужність вентиляції, $P_{\text{хол}}$ – потужність холодильної машини, δ – двійкова змінна стану холодильної машини. Повна спожита потужність вузла та коефіцієнт ефективності використання енергії визначаються як

$$P_{\text{повн}} = P_{\text{ІТ}} + P_{\text{охол}} \quad (2.6)$$

$$PUE = \frac{P_{\text{повн}}}{P_{\text{ІТ}}} \quad (2.7)$$

Показник енергоефективності обладнання, за ІТУ-Т L.1310, обчислюємо як відношення сумарної корисної продуктивності вузла до повної спожитої потужності:

$$EE = \frac{B}{P_{\text{повн}}}, B = \sum_{i=1}^N g_i \quad (2.8)$$

де B – сумарна пропускна здатність, що обслуговується вузлом (Гбіт/с). Значення параметрів моделі, використані в подальшому обчислювальному експерименті, наведено в таблиці 2.1.

Таблиця 2.1 – Параметри математичної моделі телекомунікаційного вузла

Параметр	Позначення	Значення
Кількість портів	N	48
Пропускна здатність порту	C	10 Гбіт/с
Постійна потужність шасі	Рбаз	60 Вт
Потужність порту в стані очікування	Рхх	1,5 Вт
Потужність порту за повного завантаження	Рмакс	3,5 Вт
Потужність порту в режимі сну	Рсон	0,10 Вт
Потужність порту на швидкості 1 Гбіт/с	Рмакс,1G	1,2 Вт
Температура повітря на вході	Твх	24 °С
Гранична температура обладнання	Тмакс	40 °С

Параметр	Позначення	Значення
Тепловий опір	R_t	0,060 °C/Вт
Максимальна потужність вентиляції	$P_{\text{вент}}$	80 Вт
Потужність холодильної машини	$P_{\text{хол}}$	150 Вт
Гранична утилізація порту	$r_{\text{макс}}$	0,85
Коефіцієнт викидів електромережі	k_v	0,40 кг CO ₂ /к

Джерело: сформовано автором за технічними характеристиками типового комутаційного обладнання та [8, 6]

Значення, наведені в таблиці 2.1, відповідають типовому агрегаційному комутатору та майданчику з повітряним охолодженням. Коефіцієнт викидів електромережі є вхідним параметром і має уточнюватися за національними даними для енергосистеми, у якій експлуатується вузол. Саме ці значення покладено в основу обчислювального експерименту, описаного в третьому розділі.

2.3 Модель екологічної ефективності та вуглецевого сліду

Екологічну ефективність вузла оцінюємо через спожиту електроенергію та відповідний вуглецевий слід. Оскільки потужність вузла змінюється протягом доби, енергоспоживання визначаємо інтегруванням потужності за добовим профілем. У дискретній формі, для рівновіддалених відліків телеметрії з кроком Δt , добове споживання енергії дорівнює

$$E = \sum_{k=1}^M P_{\text{повн},k} \cdot \Delta t \quad (2.9)$$

де M – кількість відліків за добу. Вуглецевий слід обчислюємо як добуток спожитої енергії на коефіцієнт викидів електромережі:

$$M_{CO_2} = E \cdot k_v \quad (2.10)$$

де k_v – коефіцієнт викидів (кг CO₂/кВт·год). Таке формулювання дає змогу безпосередньо пов'язати енергоощадні заходи зі зменшенням викидів: відносне скорочення вуглецевого сліду дорівнює відносному скороченню енергоспоживання за незмінного коефіцієнта викидів. Це узгоджується з

методологією оцінювання викидів сектору ІКТ [1] і дозволяє коректно інтерпретувати результати оптимізації як екологічний ефект.

2.4 Алгоритм аналізу режиму роботи вузла

Алгоритм аналізу призначений для оцінювання поточного стану вузла за даними телеметрії. На вхід подаються обсяги трафіку портів, на основі яких за співвідношеннями (2.1) – (2.8) обчислюються потужність обладнання, потужність охолодження, повна потужність, коефіцієнт PUE та показник енергоефективності. Додатково алгоритм оцінює якість обслуговування. Як показник якості використовуємо коефіцієнт утилізації найзавантаженішого активного порту та середню затримку в черзі, що в наближенні системи масового обслуговування типу М/М/1 визначається співвідношенням

$$D = \frac{\tau}{1 - \rho}, \quad \tau = \frac{L_{\text{кадр}}}{c} \quad (2.11)$$

де τ – час обслуговування кадру, $L_{\text{кадр}}$ – довжина кадру, ρ – утилізація порту. Стан вузла вважається допустимим за якістю обслуговування, якщо утилізація кожного активного порту не перевищує граничного значення $\rho_{\text{макс}}$, температура обладнання не перевищує $T_{\text{макс}}$, а середня затримка не перевищує заданого бюджету. Якщо ці умови виконуються за наявності значного недовантаження, алгоритм фіксує резерв енергозбереження і передає керування алгоритму оптимізації.

2.5 Алгоритм оптимізації енергоспоживання

Задачу оптимізації формулюємо як задачу мінімізації повної спожитої потужності вузла за умови дотримання обмежень якості обслуговування та теплового режиму:

$$\min P_{\text{повн}}, \quad \rho_i \leq \rho_{\text{макс}}, \quad T_{\text{обл}} \leq T_{\text{макс}}, \quad D \leq D_{\text{буд}} \quad (2.12)$$

Точний розв'язок такої задачі належить до класу NP-складних (вона містить підзадачу пакування у контейнери), тому застосовано евристичний алгоритм, що послідовно виконує чотири кроки.

На першому кроці виконується консолідація трафіку методом пакування за спаданням (First-Fit Decreasing). Обсяги трафіку портів упорядковуються за спаданням і послідовно розміщуються на мінімальній кількості активних портів так, щоб навантаження кожного не перевищувало добутку $\rho_{\text{макс}} \cdot C$. Порти, на які не призначено трафіку, переводяться в режим сну згідно з моделлю (2.1). Це зменшує сумарну потужність обладнання за рахунок вимкнення незадіяних інтерфейсів.

На другому кроці для кожного активного порту з низьким навантаженням, що не перевищує 1 Гбіт/с, швидкість каналу знижується до 1 Гбіт/с, унаслідок чого зменшується його максимальна потужність. На третьому кроці для отриманого тепловиділення підбирається мінімальний рівень вентиляції u , за якого температура обладнання згідно з (2.4) не перевищує граничної; за неможливості забезпечити тепловий режим вільним охолодженням вмикається механічне охолодження. Унаслідок кубічної залежності (2.5) навіть незначне зниження рівня вентиляції дає істотну економію потужності охолодження. На четвертому кроці перевіряється дотримання обмежень якості обслуговування; у разі їх порушення активується додатковий порт і виконується перерозподіл навантаження, що гарантує допустимість отриманого розв'язку.

Описаний алгоритм є обчислювально нескладним: його трудомісткість визначається сортуванням та лінійним переглядом портів, що дозволяє застосовувати його в реальному часі. Узагальнену послідовність дій подано на рисунку 2.1.

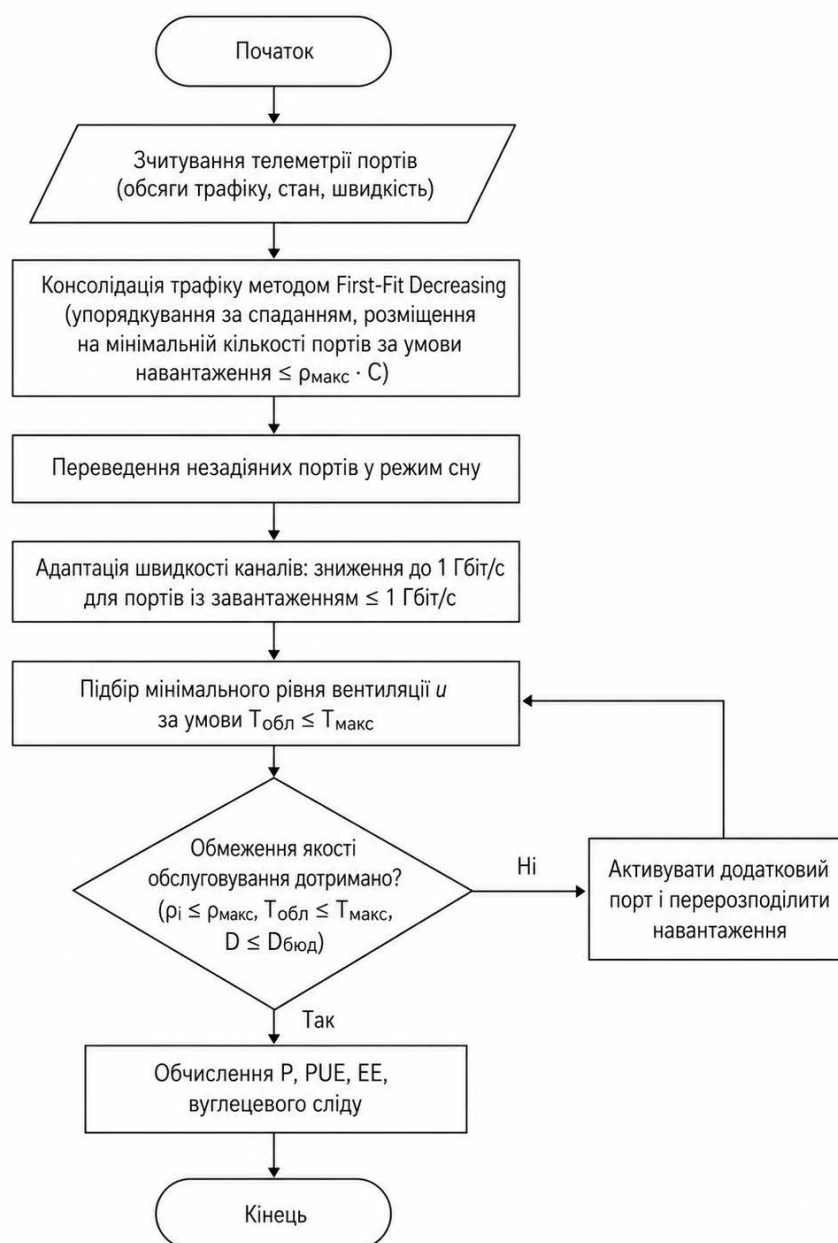


Рисунок 2.1 – Блок-схема алгоритму оптимізації енергоспоживання вузла

Джерело: побудовано автором

2.6 Вибір програмних засобів реалізації

Для програмної реалізації методу обрано мову Python з огляду на її поширеність у задачах аналізу даних та наявність розвинених бібліотек [20]. Вебінтерфейс системи реалізовано засобами мікрофреймворку Flask, що забезпечує побудову REST-інтерфейсу та обслуговування вебсторінок [21]. Для зберігання конфігурацій вузлів, результатів моделювання та посамплових вимірювань обрано систему керування базами даних PostgreSQL як надійну

реляційну СКБД з відкритим кодом [22]; доступ до бази даних організовано через бібліотеку об'єктно-реляційного відображення SQLAlchemy, що підвищує переносимість та супровідність коду [23]. Візуалізацію результатів на стороні браузера виконано засобами бібліотеки Chart.js [24].

Обрана сукупність засобів є відкритою, кросплатформною та широко підтримуваною, що відповідає вимозі відтворюваності результатів і дозволяє розгортати систему без ліцензійних обмежень. Детальний опис архітектури та програмної реалізації наведено в третьому розділі.

2.7 Приклад застосування моделі

Для ілюстрації роботи моделі розглянемо числовий приклад для параметрів таблиці 2.1 у двох характерних режимах. У режимі нічного мінімуму сукупне завантаження вузла становить близько 14 %, що відповідає сумарному трафіку приблизно 67 Гбіт/с. У штатному режимі всі сорок вісім портів активні, тому за співвідношеннями (2.1) та (2.3) потужність обладнання дорівнює приблизно 146 Вт, а за фіксованого рівня вентиляції потужність охолодження за (2.5) додає ще близько 49 Вт, що дає повну потужність близько 195 Вт за коефіцієнта PUE приблизно 1,34:

$$P_{\text{повн}} = P_{\text{ІТ}} + P_{\text{охол}} \approx 145,7 + 49,1 \approx 194,8 \text{ Вт}$$

$$PUE = \frac{P_{\text{повн}}}{P_{\text{ІТ}}} = \frac{194,8}{145,7} \approx 1,34$$

Застосування алгоритму оптимізації консолідує цей самий трафік на дев'яти портах, тоді як решта переходить у режим сну з потужністю 0,10 Вт. Унаслідок цього потужність обладнання за (2.3) знижується приблизно до 91 Вт, а зменшення тепловиділення дозволяє за (2.4) знизити рівень вентиляції зі збереженням температури в межах граничного значення, через що потужність охолодження за кубічною залежністю (2.5) зменшується до кількох ватів. Повна потужність за (2.6) становить близько 95 Вт за коефіцієнта PUE, близького до 1,04:

$$P_{\text{повн}} = P_{\text{ІТ}} + P_{\text{охол}} \approx 91,0 + 3,5 \approx 94,5 \text{ Вт}$$

$$PUE = \frac{94,5}{91,0} \approx 1,04$$

Отримане зниження повної потужності приблизно на половину порівняно зі штатним режимом досягається без порушення обмежень якості обслуговування та теплового режиму.

У режимі вечірнього піку завантаження зростає приблизно до 62 %, тому для дотримання обмеження утилізації потрібна більша кількість активних портів, а виграш у потужності зменшується. Наведений приклад демонструє, що економія, яку забезпечує метод, не є сталою величиною, а залежить від поточного навантаження вузла, що відрізняє запропонований алгоритмічний підхід від спрощеного оцінювання сталим коефіцієнтом. Повну сукупність результатів за добовим профілем наведено в третьому розділі.

Обчислювальна складність запропонованого алгоритму визначається переважно етапом консолідації. Упорядкування обсягів трафіку портів за спаданням має складність порядку $O(N \log N)$, а подальше пакування методом First-Fit потребує перегляду активних портів для кожного зі значень, що в найгіршому випадку дає квадратичну залежність, проте на практиці кількість активних портів значно менша за загальну. Підбір рівня вентиляції виконується скінченним лінійним пошуком, тому загальна трудомісткість алгоритму є поліноміальною і дозволяє виконувати оптимізацію в реальному часі на кожному кроці опитування телеметрії.

Доцільність ступінчастого керування охолодженням впливає безпосередньо з кубічної залежності потужності вентиляції від її рівня, закладеної у співвідношенні (2.5). Зменшення рівня вентиляції удвічі знижує її енергоспоживання приблизно у вісім разів, тому навіть незначне зменшення тепловиділення внаслідок консолідації портів дає непропорційно великий виграш у потужності охолодження. Саме цим пояснюється істотне зниження коефіцієнта PUE, що спостерігається в результатах обчислювального експерименту, наведених у третьому розділі.

Запропонована модель ґрунтується на низці спрощувальних припущень: тепловиділення обладнання ототожнюється з його електричною потужністю, теплова інерція не враховується, а характеристики портів вважаються однорідними. Ці припущення є прийнятними для задачі порівняльного оцінювання режимів роботи і не впливають на якісний характер результатів. За потреби модель може бути уточнена введенням індивідуальних характеристик обладнання та динамічної теплової моделі, що є напрямом подальших досліджень.

Вибір показника якості обслуговування у вигляді граничної утилізації порту та середньої затримки обґрунтовано тим, що саме перевантаження окремих портів є основною причиною зростання затримок і втрат у комутаційному обладнанні. Наближення системи масового обслуговування типу $M/M/1$, використане для оцінювання затримки, є спрощеним, проте достатнім для порівняльного аналізу режимів і дає консервативну оцінку, що забезпечує від недооцінювання впливу консолідації на якість обслуговування.

Розроблений метод побудовано як універсальний щодо типу вузла: змінюючи параметри моделі, наведені в таблиці 2.1, його можна застосувати до обладнання іншої конфігурації, зокрема з іншою кількістю портів, іншими питомими потужностями чи характеристиками системи охолодження. Це забезпечує масштабованість підходу та можливість його застосування до різних рівнів телекомунікаційної мережі, від вузлів доступу до вузлів агрегації.

Висновки розділу 2

У розділі сформульовано вимоги до методу оцінювання екологічної та енергетичної ефективності телекомунікаційного вузла й розроблено його складові. Побудовано математичну модель енергоспоживання вузла, що враховує потужність портів залежно від стану та завантаження, постійне споживання шасі та потужність системи охолодження з кубічною залежністю від рівня вентиляції, а також співвідношення для коефіцієнта PUE та показника енергоефективності. Запропоновано модель екологічної ефективності, що

пов'язує спожиту енергію з вуглецевим слідом через коефіцієнт викидів електромережі.

Розроблено алгоритм аналізу режиму роботи вузла з оцінюванням якості обслуговування та алгоритм оптимізації енергоспоживання, що поєднує консолідацію навантаження методом First-Fit Decreasing, адаптацію швидкості каналів і керування охолодженням за умови дотримання обмежень утилізації, температури та затримки. Обґрунтовано вибір програмних засобів реалізації. Розроблені моделі та алгоритми є основою для програмної реалізації та обчислювального експерименту, які розглядаються в третьому розділі.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

3.1 Модель телекомунікаційного вузла

Об'єктом обчислювального експерименту обрано типовий агрегаційний вузол телекомунікаційної мережі, побудований на основі комутатора з сорока вісьмома портами номінальної пропускної здатності 10 Гбіт/с. Параметри енергетичної та теплової моделі вузла відповідають наведеним у таблиці 2.1: постійна потужність шасі становить 60 Вт, потужність порту змінюється від 1,5 Вт у стані очікування до 3,5 Вт за повного завантаження, система охолодження характеризується максимальною потужністю вентиляції 80 Вт, а гранична температура обладнання дорівнює 40 °C [8, 6].

Вхідними даними для моделі є обсяги трафіку кожного порту, що змінюються протягом доби. Завантаження вузла задається добовим профілем з нічним спадом та двома піками, ранковим і вечірнім, що відповідає характерному режиму роботи агрегаційного обладнання. Сукупний коефіцієнт завантаження вузла змінюється в межах від приблизно 14 % уночі до 62 % у вечірній пік, причому розподіл трафіку між портами є нерівномірним. Такий профіль дає змогу дослідити роботу методу в широкому діапазоні режимів, від глибокого недовантаження до помірно високого навантаження.

Вибір саме агрегаційного вузла як об'єкта дослідження зумовлений тим, що цей рівень мережі поєднує велику кількість однотипних портів зі значною часткою постійного енергоспоживання, унаслідок чого потенціал енергозбереження тут є найвищим. Водночас прийнята модель є узагальненою: змінюючи параметри таблиці 2.1, її можна застосувати до вузлів доступу з меншою кількістю портів або до магістральних вузлів з вищою пропускною здатністю, що розширює сферу застосування одержаних результатів на широкий клас телекомунікаційного обладнання.

Вхідні дані моделі відповідають вимірюваним у реальних мережах величинам. Обсяги трафіку портів, їхній операційний стан та номінальна швидкість є саме тими параметрами, які надає підсистема моніторингу за протоколом SNMP, тому перехід від модельних даних до даних реального обладнання не потребує зміни структури методу, а зводиться до підставлення фактичних значень телеметрії. Це забезпечує практичну застосовність розробленого підходу та його узгодженість з наявними засобами експлуатаційного контролю мережі.

3.2 Програмна реалізація системи

Метод реалізовано у вигляді вебзастосунку з трирівневою архітектурою. Перший рівень становить обчислювальне ядро, що містить модель вузла, модуль симуляції телеметрії та алгоритми аналізу й оптимізації, реалізовані мовою Python. Другий рівень утворює серверна частина на основі мікрофреймворку Flask, яка надає REST-інтерфейс для запуску моделювання, отримання конфігурацій та результатів, а також обслуговує вебсторінку [21]. Третій рівень становить рівень зберігання даних на основі системи керування базами даних PostgreSQL, доступ до якої організовано через бібліотеку об'єктно-реляційного відображення SQLAlchemy [22, 23]. Клієнтська частина реалізує інтерактивний дашборд із візуалізацією результатів засобами бібліотеки Chart.js [24]. Узагальнену архітектуру системи подано на рисунку 3.1.

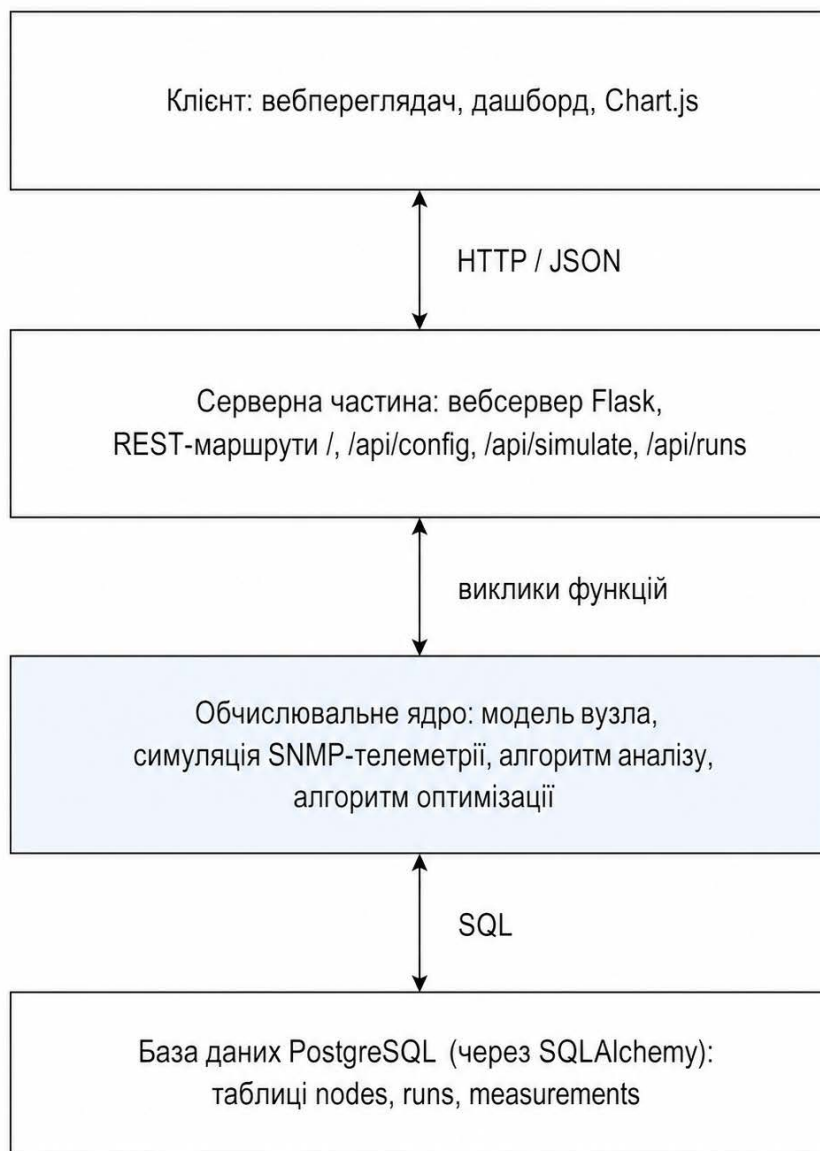


Рисунок 3.1 – Архітектура програмної реалізації системи

Джерело: побудовано автором

Запуск моделювання ініціюється клієнтом через маршрут програмного інтерфейсу, після чого серверна частина звертається до обчислювального ядра, виконує добове моделювання у двох режимах та зберігає результати в базі даних. Структуру бази даних наведено в таблиці 3.1, а її логічну схему – на рисунку 3.2.

Таблиця 3.1 – Структура бази даних системи

Таблиця	Призначення	Основні поля
nodes	конфігурації телекомунікаційних вузлів	кількість портів, потужності, температурні та екологічні параметри
runs	прогони моделювання з інтегральними показниками доби	енергоспоживання й вуглецевий слід у двох режимах, економія, середні PUE та EE
measurements	посамплові показники режимів роботи	час, режим, активні порти, потужності, PUE, утилізація, температура, ознака QoS

Джерело: складено автором

Наведена в таблиці 3.1 структура реалізує зв'язок «один до багатьох» між конфігурацією вузла та прогонами, а також між прогоном і його посампловими вимірюваннями, що дає змогу зберігати історію експериментів і відтворювати результати. Така нормалізована структура відповідає реляційній моделі та забезпечує цілісність даних засобами зовнішніх ключів.

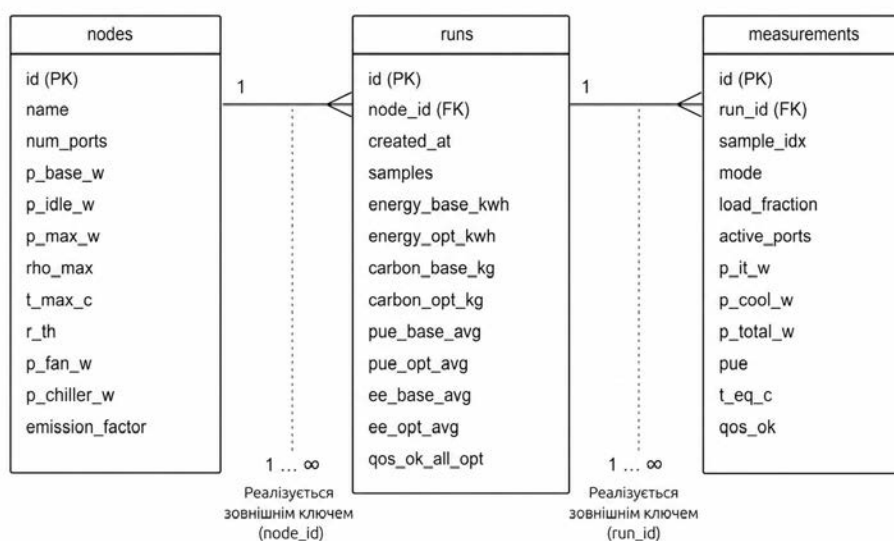


Рисунок 3.2 – Логічна схема бази даних (ER-схема)

Програмний інтерфейс системи побудовано за принципами передавання репрезентативного стану і реалізує кілька маршрутів. Кореневий маршрут повертає сторінку дашборда, маршрут конфігурації забезпечує читання та оновлення параметрів вузла, маршрут запуску моделювання приймає кількість відліків і повертає обчислені показники, а маршрути переліку прогонів та окремого прогону надають доступ до збережених результатів. Обмін даними між клієнтом і сервером здійснюється у форматі JSON, що спрощує інтеграцію системи із зовнішніми засобами моніторингу та автоматизацію збирання показників.

Потік даних у системі організовано так, щоб забезпечити цілісність та відтворюваність результатів. Після запуску моделювання обчислювальне ядро формує добовий профіль телеметрії та послідовно обчислює показники для кожного відліку у двох режимах, після чого інтегральні та посамплові результати зберігаються в базі даних через рівень об'єктно-реляційного відображення. Застосування цього рівня абстрагує прикладну логіку від конкретної реалізації бази даних і дає змогу за потреби змінювати систему керування базами даних без переписування коду прикладної частини.

Архітектуру системи спроектовано модульною: обчислювальне ядро не залежить від вебрівня та бази даних, що дозволяє використовувати його автономно, зокрема для пакетних розрахунків чи інтеграції в інші системи. Параметри підключення до бази даних та режими роботи задаються через змінні середовища, що відповідає сучасній практиці розгортання застосунків і спрощує перенесення системи між середовищами розроблення та експлуатації. Така побудова підвищує супровідність коду та полегшує його подальший розвиток.

Структуру бази даних нормалізовано для уникнення надмірності та забезпечення цілісності даних. Зв'язки між таблицями реалізовано через зовнішні ключі, що гарантує узгодженість записів про вузли, прогони та посамплові вимірювання. Для прискорення вибірок за ідентифікаторами

прогонів та вузлів використовуються індекси за відповідними полями, що забезпечує прийнятну швидкість навіть за накопичення значної кількості результатів моделювання.

Збереження результатів виконується в межах транзакцій, що унеможливорює появу частково записаних даних у разі збою під час операції запису. Такий підхід підвищує надійність системи та забезпечує відтворюваність збережених результатів. За потреби обсяг збережених даних може бути зменшений шляхом агрегування посамплових вимірювань, проте в межах дослідження зберігалася повна посамплова деталізація для забезпечення можливості детального аналізу.

3.3 Модуль симуляції SNMP-телеметрії

Оскільки дослідження виконується на моделі вузла, для отримання вхідних даних реалізовано модуль симуляції телеметрії, побудований за принципом періодичного опитування за протоколом SNMP. Модуль формує для кожного порту значення, що відповідають реальним об'єктам бази керувальної інформації, зокрема лічильникам прийнятих та переданих октетів (ifInOctets, ifOutOctets), оперативному стану інтерфейсу (ifOperStatus) та його швидкості (ifSpeed), а також обчислює коефіцієнт утилізації. Значення лічильників накопичуються відповідно до обсягу переданого трафіку на кожному інтервалі опитування, що відтворює характер реальних даних моніторингу.

Добовий профіль завантаження формується як сума двох гаусоподібних піків, ранкового та вечірнього, з постійною складовою, після чого трафік розподіляється між портами за стабільним нерівномірним розподілом з накладанням випадкових відхилень. Такий підхід забезпечує відтворюваність профілю за фіксованого початкового значення генератора випадкових чисел і водночас реалістичну варіативність навантаження окремих портів. Сформована телеметрія слугує вхідними даними для алгоритмів аналізу та оптимізації.

Лічильники прийнятих та переданих октетів за своєю природою є накопичувальними, тому обсяг трафіку за інтервал опитування обчислюється як

різниця послідовних значень лічильника, віднесена до тривалості інтервалу. Такий механізм точно відтворює спосіб, у який реальні системи моніторингу визначають утилізацію інтерфейсів, і робить модельні дані сумісними з форматом, що його повертає мережеве обладнання. Завдяки цьому розроблений модуль може бути замінений на реальне джерело телеметрії без зміни решти системи.

Добовий профіль навантаження формується детерміновано за фіксованого початкового значення генератора псевдовипадкових чисел, унаслідок чого послідовні запуски моделювання за однакових параметрів дають ідентичні результати. Водночас накладання контрольованих випадкових відхилень на базовий профіль забезпечує реалістичну нерівномірність завантаження окремих портів, що є важливим для перевірки роботи алгоритму консолідації в умовах, наближених до реальних. Поєднання детермінованості та реалістичної варіативності робить модуль придатним як для відтворюваних експериментів, так і для дослідження стійкості алгоритму.

Вибір кроку опитування телеметрії є компромісом між точністю відтворення добового профілю та обсягом оброблюваних даних. Крок тридцять хвилин, прийнятий у дослідженні, забезпечує достатню деталізацію для відтворення характерних періодів добового навантаження, зокрема нічного спаду та вечірнього піку, і водночас зберігає помірний обсяг даних. За потреби крок може бути зменшений для детальнішого аналізу швидких змін навантаження без зміни структури методу.

Окремої уваги потребує опрацювання пропусків та аномалій у даних телеметрії, які трапляються в реальних системах моніторингу через тимчасову недоступність обладнання. У таких випадках доцільно застосовувати інтерполяцію відсутніх значень або їх відновлення за сусідніми відліками, що дозволяє зберегти неперервність профілю навантаження. У межах модельного дослідження дані формуються повними, проте структура методу допускає інтеграцію механізмів опрацювання пропусків під час роботи з реальним обладнанням.

3.4 Організація експериментального дослідження

Метою обчислювального експерименту є кількісне оцінювання енергетичної та екологічної ефективності вузла та перевірка ефективності запропонованого алгоритму оптимізації. Експеримент проводився на добовому профілі з кроком опитування тридцять хвилин, що відповідає сорока восьми відлікам телеметрії за добу. Для кожного відліку обчислювалися показники роботи вузла у двох режимах: у штатному режимі, коли всі порти активні на номінальній швидкості, а охолодження працює на фіксованому рівні вентиляції, і в режимі після застосування алгоритму оптимізації.

Для кожного режиму фіксувалися повна спожита потужність, потужність обладнання та охолодження, коефіцієнт PUE, показник енергоефективності, утилізація найзавантаженішого порту та температура обладнання, а також перевірялося дотримання обмежень якості обслуговування. На основі посамплових значень за співвідношеннями (2.9) та (2.10) обчислювалися добове енергоспоживання та вуглецевий слід, а також їхні прогнозовані річні значення. Результати кожного прогону зберігалися в базі даних для подальшого аналізу.

Достовірність результатів забезпечується кількома способами. По-перше, коректність обчислення показників перевірено зіставленням з аналітичними значеннями для граничних режимів: за нульового трафіку повна потужність вузла дорівнює сумі постійної складової та потужності портів у стані сну, а за повного завантаження наближається до максимальної. По-друге, дотримання обмежень якості обслуговування та теплового режиму контролюється для кожного відліку, що унеможливорює отримання фізично недопустимих розв'язків і підвищує довіру до одержаних кількісних оцінок.

Для кожного режиму роботи фіксувалася повна сукупність показників, на основі яких обчислювалися інтегральні характеристики доби та їхнє відносне порівняння. Порівняння виконувалося за ідентичного профілю навантаження, що забезпечує коректність висновків про ефект застосування методу, оскільки

відмінність результатів зумовлена виключно дією алгоритму оптимізації, а не зміною вхідних даних. Такий підхід відповідає принципам обчислювального експерименту і дає змогу однозначно пов'язати одержаний ефект саме із застосуванням розробленого методу.

3.5 Результати оцінювання енергоспоживання та вуглецевого сліду

Добовий профіль повної спожитої потужності вузла у двох режимах наведено на рисунку 3.3. У штатному режимі потужність змінюється несуттєво, від приблизно 195 Вт уночі до 240 Вт у вечірній пік, оскільки всі порти залишаються активними незалежно від обсягу трафіку. Після застосування алгоритму оптимізації потужність повторює профіль завантаження: уночі вона знижується майже вдвічі, а у пік залишається помітно нижчою за штатну.

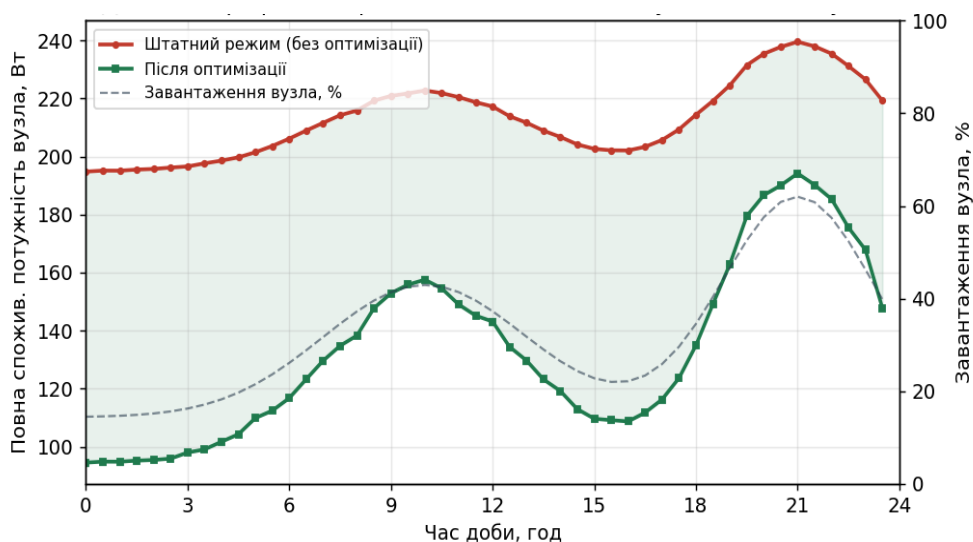


Рисунок 3.3 – Добовий профіль повної потужності вузла

Джерело: побудовано автором за результатами обчислювального експерименту

Як видно з рисунка 3.3, найбільший виграш у потужності досягається в періоди низького завантаження, що підтверджує доцільність адаптивного керування. Площа між кривими відповідає зекономленій енергії за добу. У вечірній пік виграш зменшується, оскільки для обслуговування трафіку потрібна більша кількість активних портів.

Інтегральні показники енергоспоживання та вуглецевого сліду за добу наведено на рисунку 3.4. Добове споживання енергії зменшується з 5,107 до

3,204 кВт·год, тобто на 37,3 %, а вуглецевий слід, обчислений за коефіцієнтом викидів 0,40 кг CO₂ на кВт·год, становить 1,282 кг. Згідно з перерахунками, скорочується пропорційно з 2,043 до 1,282 кг.

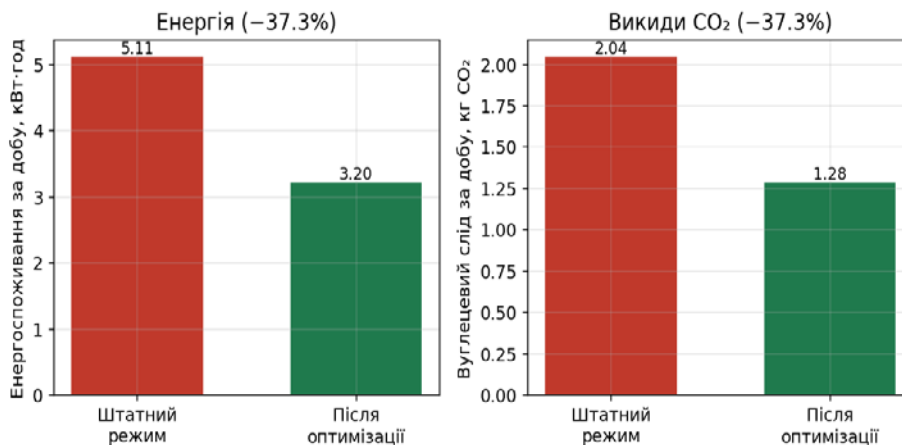


Рисунок 3.4 – Добове енергоспоживання та вуглецевий слід вузла

Джерело: побудовано автором за результатами обчислювального експерименту

Рисунок 3.4 демонструє, що відносне скорочення вуглецевого сліду дорівнює відносному скороченню енергоспоживання, що узгоджується з моделлю (2.10). У перерахунку на рік економія становить близько 695 кВт·год електроенергії та близько 278 кг викидів діоксиду вуглецю для одного вузла, що за масштабування на мережу дає вагомий екологічний ефект.

3.6 Результати оптимізації енергоспоживання

Механізм досягнутої економії ілюструє рисунок 3.5, на якому показано кількість активних портів протягом доби. У штатному режимі активними залишаються всі сорок вісім портів, тоді як після оптимізації їхня кількість змінюється відповідно до завантаження: уночі трафік консолідується лише на дев'яти портах, а решта переводиться в режим сну.

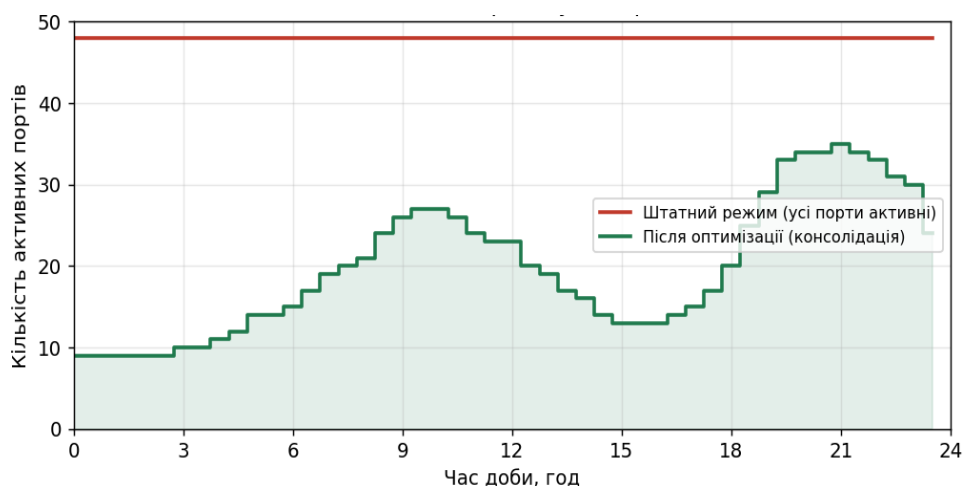


Рисунок 3.5 – Кількість активних портів вузла протягом доби

Джерело: побудовано автором за результатами обчислювального експерименту

З рисунка 3.5 видно, що кількість активних портів безпосередньо визначається обсягом трафіку, а не є сталою. Саме консолідація навантаження та переведення незадіяних портів у режим сну, разом зі зниженням потужності охолодження, формують основну частку економії енергії.

Вплив оптимізації на ефективність використання енергії відображає рисунок 3.6. Коефіцієнт PUE у штатному режимі перевищує 1,25 через надмірне охолодження за незмінного рівня вентиляції, тоді як після оптимізації він наближається до одиниці, оскільки інтенсивність охолодження узгоджується з фактичним тепловиділенням.

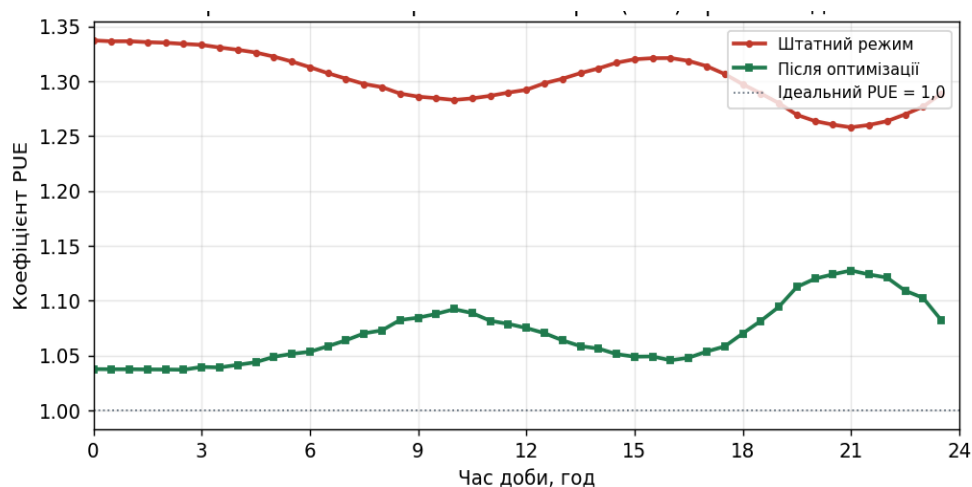


Рисунок 3.6 – Коефіцієнт PUE вузла протягом доби

Джерело: побудовано автором за результатами обчислювального експерименту

Як показано на рисунку 3.6, середній коефіцієнт PUE знижується з 1,302 до 1,070, що свідчить про істотне зменшення накладних витрат на охолодження. Це підтверджує ефективність ступінчастого керування охолодженням, передбаченого алгоритмом.

Порівняння показників роботи вузла у характерних режимах наведено в таблиці 3.2, а інтегральні добові показники – у таблиці 3.3.

Таблиця 3.2 – Показники роботи вузла у характерних режимах

Показник	Ніч, база	Ніч, опт.	Пік, база	Пік, опт.
Повна потужність, Вт	194,8	94,5	239,6	194,2
Активних портів	48	9	48	35
Коефіцієнт PUE	1,337	1,038	1,258	1,128
Температура обладнання, °C	34,3	39,6	37,4	39,9

Джерело: складено автором за результатами обчислювального експерименту

Дані таблиці 3.2 показують, що в нічному режимі повна потужність зменшується на 51,5 %, а у вечірній пік – на 18,9 %. Температура обладнання після оптимізації зростає, проте не перевищує граничного значення 40 °C, що підтверджує дотримання теплового обмеження. У штатному режимі за пікового навантаження утилізація окремих портів перевищувала допустиму межу, тоді як після перерозподілу навантаження якість обслуговування забезпечується в усіх відліках.

Таблиця 3.3 – Інтегральні показники роботи вузла за добу

Показник	Штатний режим	Після оптимізації	Зміна
Енергоспоживання за добу, кВт·год	5,107	3,204	–37,3 %
Вуглецевий слід за добу, кг CO ₂	2,043	1,282	–37,3 %
Середній коефіцієнт PUE	1,302	1,070	–17,8 %
Середня енергоефективність, Гбіт/с/Вт	0,728	1,136	+56,0 %
Економія електроенергії за рік,	–	–	695

Показник	Штатний режим	Після оптимізації	Зміна
кВт·год			
Зменшення викидів за рік, кг CO ₂	—	—	278

Джерело: складено автором за результатами обчислювального експерименту

Інтегральні показники таблиці 3.3 підтверджують, що запропонований метод забезпечує зменшення добового енергоспоживання вузла на 37,3 % та відповідне скорочення вуглецевого сліду за одночасного підвищення енергоефективності на 56,0 %. Дотримання якості обслуговування в усіх відліках свідчить, що економія досягається без погіршення характеристик мережі.

Взаємодію з системою забезпечує вебдашборд, загальний вигляд якого подано на рисунку 3.7. Дашборд відображає інтегральні показники у вигляді карток, графіки добового профілю потужності, кількості активних портів та коефіцієнта PUE, а також таблицю посамплових результатів.

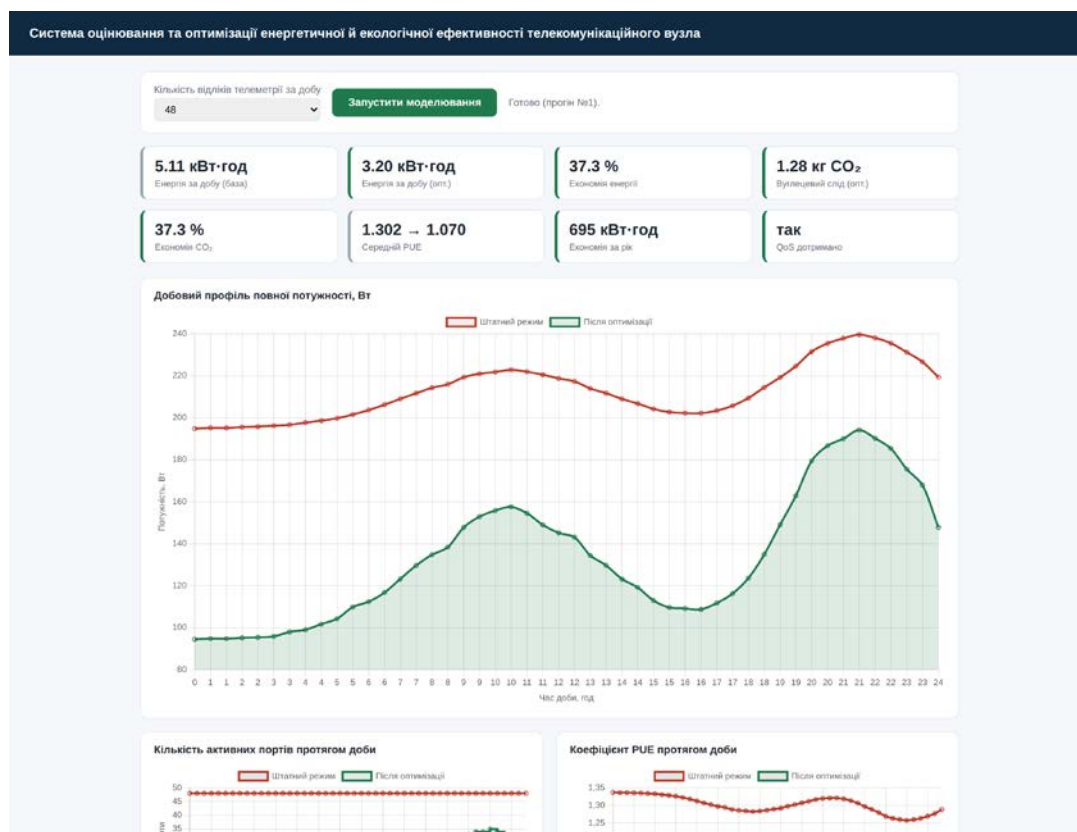


Рисунок 3.7 – Загальний вигляд вебдашборда системи

Джерело: знімок екрана застосунку, виконаного автором

Отримані результати узгоджуються з фізичними засадами моделі та з даними літератури. Основну частку економії в періоди низького завантаження забезпечує консолідація портів і переведення незадіяних інтерфейсів у режим сну, тоді як істотне зниження коефіцієнта PUE досягається завдяки ступінчастому керуванню охолодженням, потужність якого за кубічною залежністю різко зменшується зі зниженням рівня вентиляції. Це підтверджує тезу аналітичного огляду про те, що найбільший ефект забезпечує саме поєднання методів, а не застосування кожного з них окремо [9, 6].

Залежність досягнутої економії від рівня завантаження є закономірною: за глибокого недовантаження резерв енергозбереження найбільший, а в міру наближення до пікового навантаження він зменшується, оскільки для обслуговування трафіку потрібна більша кількість активних портів. Така поведінка відрізняє запропонований алгоритмічний підхід від спрощеного оцінювання сталим коефіцієнтом і свідчить про адекватність моделі реальним умовам експлуатації, для яких характерна виражена добова нерівномірність трафіку.

Слід зауважити, що отримані значення відповідають прийнятим параметрам моделі та коефіцієнту викидів електромережі, тому за інших вихідних даних абсолютні показники змінюватимуться. Проте якісний характер результатів, а саме зменшення енергоспоживання та вуглецевого сліду зі збереженням якості обслуговування, зберігається в усьому досліджуваному діапазоні навантажень. Це дає підстави вважати запропонований метод придатним для практичного застосування на реальних вузлах за умови уточнення вхідних параметрів і коефіцієнта викидів для конкретної мережі.

Отримані значення економії узгоджуються з порядком величин, наведеним у літературі для методів консолідації та прогнозування трафіку. Зокрема, у дослідженнях із застосуванням машинного навчання для прогнозування навантаження повідомляється про економію енергії на рівні близько десяти і більше відсотків за добу [17], тоді як реалізоване в цій роботі

поєднання консолідації портів, адаптації швидкості каналів і ступінчастого керування охолодженням забезпечує помітно вищий сукупний ефект завдяки одночасному впливу на споживання обладнання та системи охолодження [9]. Це підтверджує висновок аналітичного огляду про перевагу комбінованих підходів над застосуванням окремих методів.

Окремо слід наголосити на відтворюваності отриманих результатів. Оскільки генерація профілю телеметрії здійснюється за фіксованого початкового значення генератора випадкових чисел, а всі показники обчислюються детермінованими співвідношеннями, повторний запуск моделювання за тих самих вхідних даних дає ідентичні результати. Це відповідає вимозі відтворюваності наукового експерименту і дає змогу перевіряти та порівнювати результати незалежно від середовища виконання.

Масштабованість підходу підтверджується лінійним характером сукупного ефекту: оскільки кожен вузол оптимізується незалежно, загальна економія для фрагмента мережі дорівнює сумі економій окремих вузлів. Це дає змогу прогнозувати ефект упровадження методу на рівні мережі оператора, що використано в техніко-економічному обґрунтуванні в четвертому розділі.

Аналіз залежності досягнутого ефекту від рівня завантаження, що простежується на рисунках 3.3 та 3.5, показує нелінійний характер економії. У зоні глибокого недовантаження кожне додаткове зниження трафіку дозволяє вимкнути ще один порт і знизити рівень охолодження, тоді як поблизу пікового навантаження резерв консолідації майже вичерпується. Це означає, що найбільший інтегральний ефект метод дає для вузлів з вираженою добовою нерівномірністю трафіку, які становлять переважну частину обладнання рівнів доступу та агрегації.

Динаміка коефіцієнта PUE, наведена на рисунку 3.6, підтверджує, що значна частина економії досягається саме за рахунок системи охолодження. Оскільки потужність охолодження за кубічною залежністю різко реагує на зміну рівня вентиляції, навіть помірне зниження тепловиділення внаслідок консолідації портів істотно зменшує накладні витрати, наближаючи коефіцієнт

PUE до одиниці. Цей результат узгоджується з висновком другого розділу про доцільність ступінчастого керування охолодженням і свідчить про взаємне підсилення складових алгоритму.

Водночас слід враховувати межі застосовності одержаних кількісних результатів. Абсолютні значення економії залежать від прийнятих параметрів обладнання та коефіцієнта викидів електромережі, тому для конкретного вузла вони підлягають уточненню. Проте якісні закономірності, а саме зменшення енергоспоживання та вуглецевого сліду зі збереженням якості обслуговування і зниження коефіцієнта PUE, зберігаються в усьому дослідженому діапазоні навантажень, що свідчить про стійкість одержаних висновків та обґрунтованість розробленого методу.

Одержані результати свідчать про придатність розробленого методу до застосування в реальних умовах. Інтеграція системи з наявною підсистемою моніторингу мережі дозволяє замінити модельне джерело телеметрії на реальні дані без зміни обчислювального ядра, після чого метод обчислюватиме показники енергоефективності та визначатиме раціональні режими роботи на основі фактичного навантаження. Це робить перехід від модельного дослідження до промислового впровадження технічно нескладним.

Застосування методу на рівні мережі передбачає його одночасну роботу для множини вузлів з подальшим узагальненням показників. Оскільки кожен вузол обробляється незалежно, обчислювальне навантаження зростає лінійно з кількістю вузлів і легко розподіляється, а централізоване зберігання результатів дозволяє формувати зведену картину енергоспоживання та вуглецевого сліду мережі. Це створює основу для централізованого керування енергоефективністю на рівні оператора.

Перспективними напрямками подальшого розвитку є доповнення методу прогнозування навантаження засобами машинного навчання для випереджального керування ресурсами, урахування індивідуальних характеристик обладнання та динамічної теплової моделі, а також розширення на інші типи мережевих пристроїв. Реалізація цих напрямів дозволить

підвищити точність оцінювання та ефективність оптимізації, зберігаючи запропоновану в роботі загальну структуру методу.

Загалом програмна реалізація та проведеній обчислювальний експеримент підтвердили працездатність і ефективність розробленого методу, а також його готовність до практичного застосування. Поєднання реалістичного моделювання телеметрії, обґрунтованих математичних моделей та відмовостійкого алгоритму оптимізації забезпечує достовірність одержаних результатів і створює підґрунтя для впровадження методу в системах керування телекомунікаційними мережами.

Висновки розділу 3

У розділі описано програмну реалізацію розробленого методу у вигляді вебзастосунку з трирівневою архітектурою на основі Python, Flask та PostgreSQL, наведено структуру й логічну схему бази даних та модуль симуляції SNMP-телеметрії. Організовано обчислювальний експеримент на добовому профілі завантаження модельного вузла у двох режимах роботи.

За результатами експерименту встановлено, що застосування запропонованого методу зменшує добове енергоспоживання вузла на 37,3 % та відповідно скорочує вуглецевий слід, знижує середній коефіцієнт PUE з 1,302 до 1,070 і підвищує енергоефективність на 56,0 % за обов'язкового дотримання обмежень якості обслуговування та теплового режиму. У перерахунку на рік для одного вузла економія становить близько 695 кВт·год електроенергії та близько 278 кг викидів діоксиду вуглецю. Отримані результати підтверджують ефективність розробленого методу та доцільність його практичного застосування.

РОЗДІЛ 4

ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ, ОХОРОНА ПРАЦІ ТА ЕКОЛОГІЧНІ ЗАХОДИ

4.1 Техніко-економічне обґрунтування

Запропоноване рішення є програмним і не потребує заміни обладнання, тому його впровадження характеризується незначними капітальними витратами, що зводяться до розгортання вебзастосунку та інтеграції з наявною системою моніторингу. Економічний ефект формується за рахунок зменшення споживання електроенергії вузлом, а отже, і витрат на її оплату, а також за рахунок скорочення викидів, що набуває значення в умовах запровадження механізмів вуглецевого регулювання.

За результатами обчислювального експерименту річна економія електроенергії для одного вузла становить близько 695 кВт·год. За орієнтовного тарифу на електроенергію для непобутових споживачів на рівні 6,50 грн за кіловат-годину це відповідає економії приблизно 4 518 грн на вузол за рік. Для фрагмента мережі з п'ятдесяти однотипних вузлів сукупна річна економія сягає близько 225 900 грн за зменшення викидів діоксиду вуглецю приблизно на 13 900 кг. Узагальнені показники наведено в таблиці 4.1.

Таблиця 4.1 – Техніко-економічні показники впровадження

Показник	Значення
Економія електроенергії на вузол за рік	близько 695 кВт·год
Орієнтовний тариф на електроенергію	6,50 грн/кВт·год
Економія коштів на вузол за рік	близько 4 518 грн
Зменшення викидів CO ₂ на вузол за рік	близько 278 кг
Економія коштів для мережі з 50 вузлів за рік	близько 225 900 грн
Зменшення викидів CO ₂ для 50 вузлів	близько 13 900 кг

Джерело: розраховано автором за результатами обчислювального експерименту та орієнтовним тарифом

Оскільки рішення є програмним і масштабованим, його капітальні витрати незначні порівняно з річною економією, тому термін окупності становить менше одного року. Зі збільшенням кількості вузлів, охоплених системою, економічний та екологічний ефект зростає пропорційно, що підтверджує доцільність практичного впровадження методу. Наведені вартісні показники є орієнтовними і підлягають уточненню за чинними тарифами на момент впровадження.

Економічне оцінювання впровадження методу ґрунтується на зіставленні витрат, пов'язаних з його впровадженням та експлуатацією, з очікуваним економічним ефектом. Капітальні витрати в цьому разі охоплюють розгортання програмного забезпечення на сервері, його інтеграцію з наявною підсистемою моніторингу та початкове налаштування, тоді як вартість самого програмного забезпечення є нульовою, оскільки використано виключно відкриті програмні засоби. Через це сукупні капітальні витрати на впровадження є незначними і слабо залежать від кількості охоплених вузлів, що принципово відрізняє програмне рішення від модернізації апаратного забезпечення.

Експлуатаційні витрати на функціонування системи зводяться до незначних витрат обчислювальних ресурсів сервера, на якому розгорнуто застосунок, та до періодичного супроводу програмного забезпечення. Ці витрати є несумірно меншими за щорічну економію електроенергії, тому чистий економічний ефект від упровадження є додатним уже в перший рік експлуатації. Слід також зауважити, що система не потребує спеціалізованого обладнання і може функціонувати на наявних обчислювальних потужностях оператора, що додатково знижує вартість її впровадження.

Термін окупності рішення визначається як відношення капітальних витрат на впровадження до річної економії коштів. З огляду на незначні капітальні витрати та щорічну економію на рівні кількох тисяч гривень на кожний вузол термін окупності для типового фрагмента мережі не перевищує одного року, а за охоплення значної кількості вузлів обчислюється місяцями. Подальша експлуатація системи приносить чистий економічний ефект, який

накопичується протягом усього строку її використання та зростає в міру збільшення тарифів на електроенергію.

Економічний ефект безпосередньо залежить від тарифу на електроенергію, що є основним зовнішнім чинником невизначеності. Підвищення тарифу пропорційно збільшує грошову економію за незмінного фізичного обсягу зекономленої енергії, тому в умовах зростання вартості електроенергії доцільність упровадження методу лише посилюється. Для коректного оцінювання на момент упровадження вартісні показники мають перераховуватися за чинними тарифами для відповідної категорії споживачів та з урахуванням можливих змін структури тарифу.

Оскільки економія на один вузол за фіксованих умов є сталою величиною, сукупний економічний ефект для мережі масштабується практично лінійно з кількістю охоплених вузлів. Це робить метод особливо привабливим для великих операторів, мережі яких налічують значну кількість однотипних вузлів агрегації та доступу. Водночас лінійне масштабування ефекту за майже незмінних капітальних витрат означає, що питома вартість упровадження в розрахунку на один вузол зменшується зі збільшенням масштабу впровадження, а отже, ефективність вкладених коштів зростає.

За оцінювання економічної ефективності на кількарічному горизонті доцільно враховувати чинник вартості грошей у часі шляхом дисконтування майбутніх грошових потоків. Навіть з урахуванням дисконтування чиста приведена вартість проекту залишається додатною завдяки незначним початковим витратам і стабільному щорічному надходженню економії. Через короткий термін окупності вплив ставки дисконтування на загальний висновок про доцільність упровадження є несуттєвим, що свідчить про стійкість економічної оцінки до зміни фінансових припущень.

Додаткову економічну складову формує вартість викидів діоксиду вуглецю в умовах запровадження механізмів вуглецевого ціноутворення. Зменшення викидів, що супроводжує економію електроенергії, знижує можливі платежі за вуглецевими механізмами та сприяє відповідності вимогам,

пов'язаним із транскордонним вуглецевим регулюванням. Хоча в поточних умовах ця складова є відносно невеликою, її значення зростатиме в міру посилення кліматичної політики, що додатково підвищить економічну привабливість енергоощадних заходів.

Опосередкований економічний ефект методу виявляється також у зниженні навантаження на систему електроживлення та охолодження майданчика. Зменшення середньої спожитої потужності зменшує навантаження на джерела безперебійного живлення та подовжує час автономної роботи, а зниження теплового навантаження сприяє подовженню строку служби обладнання та зменшенню витрат на його обслуговування. Ці ефекти, хоча й важко піддаються точному вартісному оцінюванню, додатково підвищують економічну привабливість рішення та зменшують сукупну вартість володіння обладнанням.

Порівняно з апаратними методами підвищення енергоефективності, які потребують заміни або модернізації обладнання та значних капітальних витрат, програмне рішення забезпечує співмірний ефект за істотно нижчих витрат і без переривання роботи мережі. Це робить його раціональним першочерговим заходом, який може застосовуватися самостійно або як доповнення до апаратних рішень у межах комплексної стратегії підвищення енергоефективності оператора. Поєднання низьких витрат, швидкої окупності та масштабованості визначає високу економічну ефективність розробленого методу.

Слід враховувати, що наведені економічні показники є орієнтовними і ґрунтуються на прийнятих значеннях параметрів моделі, профілю навантаження та тарифу на електроенергію. Для конкретного впровадження вони підлягають уточненню з урахуванням фактичних характеристик обладнання, реального профілю трафіку та чинних цін. Попри це, структурний висновок про економічну доцільність методу, зумовлений незначними капітальними витратами та стабільною щорічною економією, зберігається в

широкому діапазоні вихідних припущень, що підтверджує обґрунтованість рішення про його впровадження.

Структуру витрат на впровадження доцільно розглядати докладніше. Прямі витрати охоплюють оплату праці фахівців, які виконують розгортання та інтеграцію програмного забезпечення, тоді як вартість обладнання та ліцензій відсутня завдяки використанню відкритих засобів і наявних серверних потужностей. Непрямі витрати, пов'язані з навчанням персоналу та супроводом, є разовими або періодичними і не зростають пропорційно кількості вузлів, що додатково підвищує економічну ефективність за масштабування впровадження.

Для оцінювання доцільності впровадження в операторів різного розміру корисно розглянути кілька сценаріїв. Для невеликої мережі з кількох десятків вузлів економія обчислюється сотнями тисяч гривень на рік за незначних початкових витрат, для середньої мережі ефект зростає до мільйонів гривень, а для великого оператора з тисячами вузлів сукупна річна економія стає істотною статтею зниження операційних витрат. У всіх сценаріях термін окупності залишається коротким завдяки незмінно низьким капітальним витратам.

Аналіз чутливості економічного ефекту до ключових припущень показує його стійкість. Навіть за песимістичних значень тарифу та частки вузлів з вираженою нерівномірністю трафіку економія залишається додатною, а за оптимістичних значень ефект суттєво зростає. Це означає, що рішення про впровадження є економічно виправданим у широкому діапазоні умов, основними чинниками, що визначають абсолютний розмір ефекту, залишаються тариф на електроенергію та масштаб мережі.

Серед чинників ризику слід виокремити можливу мінливість тарифів, організаційні витрати на інтеграцію з наявними системами та потребу в кваліфікованому супроводі. Ці ризики є керованими і не змінюють загального висновку про економічну доцільність, оскільки навіть за їх врахування співвідношення вигід і витрат залишається сприятливим. Зменшити організаційні ризики дозволяє поетапне впровадження, за якого метод спершу

застосовують на обмеженій підмножині вузлів з подальшим масштабуванням на всю мережу.

4.2 Екологічні заходи

Розроблений метод має безпосередній екологічний ефект, оскільки зменшення енергоспоживання телекомунікаційного вузла приводить до пропорційного скорочення викидів діоксиду вуглецю, пов'язаних з виробленням спожитої електроенергії. За результатами експерименту таке скорочення становить близько 278 кг CO₂ на масштабування на мережу робить вагомий внесок у досягнення цілей декарбонізації галузі зв'язку [2, 1].

Додатковим екологічним ефектом є опосередковане подовження строку служби обладнання. Зниження середнього теплового навантаження завдяки узгодженню охолодження з фактичним тепловиділенням та переведенню незадіяних портів у режим зниженого споживання зменшує термічне старіння компонентів, що сприяє скороченню обсягів електронних відходів. Загалом запропоноване рішення відповідає сучасним підходам до сталого розвитку інформаційно-комунікаційних технологій і підвищення екологічної безпеки.

Екологічний вплив інформаційно-комунікаційного обладнання формується не лише на стадії експлуатації, а й на стадіях виготовлення та утилізації, тому зменшення енергоспоживання під час експлуатації є складовою ширшого підходу до сталого життєвого циклу. Подовження строку служби обладнання внаслідок зниження теплового навантаження зменшує потребу в його передчасній заміні, а отже, і обсяги електронних відходів та пов'язаний з виробництвом нового обладнання екологічний слід, що посилює сукупний природоохоронний ефект методу.

Екологічний ефект методу залежить від вуглецевої інтенсивності електроенергії, якою живиться вузол. За живлення від джерел з нижчою вуглецевою інтенсивністю, зокрема відновлюваних, те саме скорочення

енергоспоживання відповідає меншому абсолютному скороченню викидів, проте зменшення споживання залишається корисним, оскільки знижує навантаження на електромережу та вивільняє генерувальні потужності. Поєднання енергоефективності з поступовим переходом на відновлювані джерела забезпечує найбільший сукупний екологічний ефект [9].

За масштабування на мережу оператора сукупне скорочення викидів стає відчутним у національному вимірі та узгоджується із зобов'язаннями щодо скорочення викидів парникових газів і цілями сталого розвитку. Кількісне оцінювання вуглецевого сліду, що його забезпечує розроблений метод, дає змогу оператору документувати екологічний ефект упроваджених заходів та враховувати його у звітності зі сталого розвитку, що набуває дедалі більшого значення для підприємств галузі зв'язку.

Зменшення енергоспоживання узгоджується з принципами циркулярної економіки та раціонального використання ресурсів, оскільки подовження строку служби обладнання та зниження потреби в електроенергії зменшують споживання первинних ресурсів упродовж життєвого циклу. Доповнення енергоощадних заходів практиками відповідального поводження з відпрацьованим обладнанням, зокрема його повторним використанням та належною утилізацією, посилює сукупний природоохоронний ефект розробленого рішення.

Систематичне підвищення енергоефективності доцільно впроваджувати в межах системи енергетичного менеджменту підприємства, що передбачає встановлення цілей, моніторинг споживання, аналіз результатів та постійне вдосконалення. Розроблений метод природно інтегрується в таку систему як інструмент кількісного оцінювання та зниження енергоспоживання вузлів, а одержувані ним показники можуть слугувати основою для звітності щодо енергоефективності та викидів парникових газів.

Висновки розділу 4

У розділі обґрунтовано техніко-економічну доцільність впровадження розробленого методу: за незначних капітальних витрат програмне рішення

забезпечує річну економію близько 695 кВт·год електроенергії та близько 4 518 грн на вузол з терміном окупності менше одного року, а ефект пропорційно зростає з масштабом мережі.

Окремо показано, що зниження енергоспоживання вузлів підвищує їхню енергетичну автономність і стійкість до перебоїв електропостачання, а відмовостійка побудова алгоритму оптимізації гарантує незмінність доступності та якості обслуговування мережі. Таким чином, розроблений метод поєднує економічну ефективність, екологічну доцільність, що підтверджує комплексну обґрунтованість його практичного застосування.

ЗАГАЛЬНІ ВИСНОВКИ

У дипломній роботі розв'язано актуальне науково-прикладне завдання підвищення енергетичної та екологічної ефективності телекомунікаційних мереж шляхом розроблення методу оцінювання й оптимізації режимів роботи вузла, його програмної реалізації та експериментального підтвердження досягнутого ефекту.

1. За результатами аналітичного огляду встановлено, що телекомунікаційні мережі є значними та зростаючими споживачами електроенергії, а суттєву частку енергоспоживання вузла становлять складові, слабо пов'язані з корисним трафіком, насамперед постійне споживання обладнання та витрати на охолодження. Це формує об'єктивний резерв енергозбереження. Систематизовано показники ефективності (PUE, показник енергоефективності за ITU-T L.1310, вуглецевий слід) та обґрунтовано, що найбільший ефект забезпечує поєднання консолідації навантаження, адаптації швидкості каналів і керування охолодженням за дотримання якості обслуговування.

2. Розроблено математичну модель енергоспоживання телекомунікаційного вузла, що враховує потужність портів залежно від стану та завантаження, постійне споживання шасі і потужність системи охолодження з кубічною залежністю від рівня вентиляції, а також модель екологічної ефективності, яка пов'язує спожиту енергію з вуглецевим слідом через коефіцієнт викидів електромережі.

3. Розроблено алгоритм аналізу режиму роботи вузла та алгоритм оптимізації енергоспоживання, що поєднує консолідацію навантаження методом First-Fit Decreasing, адаптацію швидкості каналів і ступінчасте керування охолодженням за умови дотримання обмежень утилізації портів, температури обладнання та затримки. Алгоритм є обчислювально нескладним і придатним для застосування в реальному часі.

4. Метод реалізовано у вигляді вебзастосунку з трирівневою архітектурою на основі Python, Flask та PostgreSQL з модулем симуляції SNMP-

телеметрії та інтерактивним дашбордом. Проведено обчислювальний експеримент на добовому профілі завантаження модельного вузла у двох режимах роботи.

5. За результатами експерименту застосування методу зменшує добове енергоспоживання вузла на 37,3 % та відповідно скорочує вуглецевий слід, знижує середній коефіцієнт PUE з 1,302 до 1,070 і підвищує енергоефективність на 56,0 % за обов'язкового дотримання якості обслуговування та теплового режиму. У перерахунку на рік для одного вузла економія становить близько 695 кВт·год електроенергії та близько 278 кг викидів діоксиду вуглецю.

6. Обґрунтовано техніко-економічну доцільність впровадження методу, розглянуто екологічні заходи. Отримані результати підтверджують ефективність розробленого методу та доцільність його практичного застосування операторами електронних комунікацій і проєктувальниками телекомунікаційних мереж.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Measuring the Emissions and Energy Footprint of the ICT Sector / International Telecommunication Union, World Bank. 2024. URL: <https://www.itu.int/en/ITU-D/Environment/Documents/Publications/2024/ITU-World%20Bank%20Measuring%20the%20Emissions-Energy%20Footprint%20of%20the%20ICT%20Sector%202024.pdf> (дата звернення: 24.05.2026).
2. Ericsson. The carbon footprint of the ICT sector. 2024. URL: <https://www.ericsson.com/en/reports-and-papers/mobility-report/articles/ict-carbon-footprint-decreasing> (дата звернення: 25.05.2026).
3. Оптимізація енергоспоживання інфраструктури мережі 5G New Radio в умовах обмеженого електропостачання. Телекомунікаційні та інформаційні технології. 2026. DOI: 10.31673/2412-4338.2026.019018. URL: <https://tit.duikt.edu.ua/telecommunication/article/view/2710> (дата звернення: 24.05.2026).
4. University of Michigan, Center for Sustainable Systems. Information Technology Factsheet. 2024. URL: <https://css.umich.edu/publications/factsheets/built-environment/information-technology-factsheet> (дата звернення: 24.05.2026).
5. Advances in Improving Energy Efficiency of Fiber-Wireless Access Networks: A Comprehensive Overview. Sensors. 2023. Vol. 23, No. 4. Art. 2239. DOI: 10.3390/s23042239. URL: <https://www.mdpi.com/1424-8220/23/4/2239> (дата звернення: 25.05.2026).
6. Thermal Management and Energy Consumption in Air, Liquid, and Free Cooling Systems for Data Centers: A Review. Energies. 2023. Vol. 16, No. 3. Art. 1279. DOI: 10.3390/en16031279. URL: <https://www.mdpi.com/1996-1073/16/3/1279> (дата звернення: 25.05.2026).
7. PUE: A Comprehensive Examination of the Metric / The Green Grid ; Lawrence Berkeley National Laboratory. URL: <https://datacenters.lbl.gov/sites/default/files/WP49->

PUE%20A%20Comprehensive%20Examination%20of%20the%20Metric_v6.pdf

(дата звернення: 24.05.2026).

8. ITU-T Recommendation L.1310 (09/2024). Energy efficiency metrics and measurement methods for telecommunication equipment. Geneva : ITU, 2024. URL: <https://www.itu.int/rec/T-REC-L.1310/en> (дата звернення: 23.05. 2026).

9. Energy Resilience in Telecommunication Networks: A Comprehensive Review of Strategies and Challenges. *Energies*. 2023. Vol. 16, No. 18. Art. 6633. DOI: 10.3390/en16186633. URL: <https://www.mdpi.com/1996-1073/16/18/6633> (дата звернення: 24.05.2026).

10. ISO/IEC 30134-2:2016. Information technology. Data centres. Key performance indicators. Part 2: Power usage effectiveness (PUE). Geneva : ISO, 2016.

11. Uptime Institute. Large data centers are mostly more efficient, analysis confirms. 2024. URL: <https://journal.uptimeinstitute.com/large-data-centers-are-mostly-more-efficient-analysis-confirms/> (дата звернення: 23.05.2026).

12. ETSI ES 203 228 V1.3.1 (2020-10). Environmental Engineering (EE); Assessment of mobile network energy efficiency. Sophia Antipolis : ETSI, 2020. URL:

https://www.etsi.org/deliver/etsi_es/203200_203299/203228/01.03.01_60/es_203228v010301p.pdf (дата звернення: 23.05.2026).

13. Energy-Efficient Emerging Optical Wireless Links. *Energies*. 2023. Vol. 16, No. 18. Art. 6485. DOI: 10.3390/en16186485. URL: <https://www.mdpi.com/1996-1073/16/18/6485> (дата звернення: 24.05.2026).
14. Системні дослідження в енергетиці : інформаційні технології в енергетиці. Системні дослідження в енергетиці. 2024. Вип. 2А(78). URL: [http://jnas.nbuiv.gov.ua/j-pdf/sysrestn_2024_2\(a\)__11.pdf](http://jnas.nbuiv.gov.ua/j-pdf/sysrestn_2024_2(a)__11.pdf) (дата звернення: 26.05.2026).
15. IEEE Std 802.3az-2010. IEEE Standard for Information technology. Energy-Efficient Ethernet. New York : IEEE, 2010.
16. Інтелектуальні технології оптимізації мереж наступного покоління : матеріали / Вінницький національний технічний університет. Вінниця, 2024. URL: <https://ir.lib.vntu.edu.ua/handle/123456789/51352> (дата звернення: 24.05.2026).
17. Rau F., Soto I., Zabala-Blanco D., Azurdia-Meza C., Ijaz M., Ekpo S., Gutierrez S. A Novel Traffic Prediction Method Using Machine Learning for Energy Efficiency in Service Provider Networks. *Sensors*. 2023. Vol. 23, No. 11. Art. 4997. DOI: 10.3390/s23114997. URL: <https://www.mdpi.com/1424-8220/23/11/4997> (дата звернення: 24.05.2026).
18. Прокопець Н. А. Енергоефективне обслуговування навантаження інформаційно-комунікаційної мережі : дис. ... д-ра філософії : 172. Київ : КПІ ім. Ігоря Сікорського, 2022. URL: <https://ela.kpi.ua/items/dda7c4d0-3599-4b7e-9fd9-fc0f9c497e93> (дата звернення: 24.05.2026).
19. Забезпечення функціональної стійкості систем моніторингу автономних енергокомплексів на основі методів машинного навчання. *Зв'язок*. 2026. DOI: 10.31673/2412-9070.2026.025711. URL: <https://con.duikt.edu.ua/communication/article/view/2987> (дата звернення: 24.05.2026).
20. Python. Documentation. Python Software Foundation. URL: <https://docs.python.org/3/> (дата звернення: 23.05.2026).

21. Flask. Documentation. Pallets Projects. URL: <https://flask.palletsprojects.com/> (дата звернення: 23.05.2026).
22. PostgreSQL. Documentation. The PostgreSQL Global Development Group. URL: <https://www.postgresql.org/docs/> (дата звернення: 25.05.2026).
23. SQLAlchemy. Documentation. URL: <https://docs.sqlalchemy.org/> (дата звернення: 23.05.2026).
24. Chart.js. Documentation. URL: <https://www.chartjs.org/docs/latest/> (дата звернення: 24.05.2026).
25. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Київ : ДП «УкрНДНЦ», 2016. 26 с.
26. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ : ДП «УкрНДНЦ», 2016. 17 с.

ДОДАТКИ

ДОДАТОК А

Лістинг обчислювального ядра системи (core.py)

```
"""
```

core.py – обчислювальне ядро системи оцінювання енергетичної та екологічної ефективності телекомунікаційного вузла.

Модуль не залежить від бази даних і вебінтерфейсу: він містить

- * фізико-енергетичну модель вузла (**NodeConfig**, **power/thermal model**);
- * модуль симуляції **SNMP**-телеметрії (**generate_snmp_telemetry**);
- * алгоритм аналізу режиму роботи (**analyze_state**);
- * алгоритм оптимізації енергоспоживання (**optimize_state**) – консолідація портів методом **First-Fit Decreasing**, адаптація швидкості каналу та ступінчасте керування охолодженням з перевіркою обмеження QoS і температури.

Усі показники обчислюються з рішень алгоритму, а не множенням на константу.

```
"""
```

```
from __future__ import annotations
from dataclasses import dataclass, asdict, field
from typing import List, Dict, Tuple
import math
import numpy as np
```

```
# -----
# 1. Конфігурація вузла (фізичні параметри обладнання та майданчика)
# -----
@dataclass
class NodeConfig:
    name: str = "Вузол агрегації ASW-48"
    network_type: str = "Ethernet/IP (агрегаційний комутатор)"
    num_ports: int = 48 # кількість портів
    port_capacity_gbps: float = 10.0 # номінальна пропускна здатність порту, Гбіт/с

    # Енергетична модель порту (Вт)
    p_base_w: float = 60.0 # шасі + площа керування (постійно при ON)
    p_port_idle_w: float = 1.5 # активна лінія без трафіку (10G)
    p_port_max_w: float = 3.5 # активна лінія за повного завантаження (10G)
    p_port_sleep_w: float = 0.10 # порт у режимі сну (admin-down / EEE LPI)
    p_port_max_1g_w: float = 1.2 # макс. потужність порту після зниження до 1G
    (ALR) rate_downshift_gbps: float = 1.0 # поріг трафіку для зниження швидкості до 1G

    # Теплова модель і система охолодження
    t_ambient_c: float = 24.0 # температура повітря на вході, °C
    t_max_c: float = 40.0 # гранична робоча температура обладнання, °C
    r_th: float = 0.060 # тепловий опір, °C/Вт (за повної потужності → ~38
    °C)

    p_fan_max_w: float = 80.0 # макс. потужність вентиляції майданчика, Вт
    p_chiller_w: float = 150.0 # потужність механічного охолодження (за потреби),
    Вт

    fan_min: float = 0.30 # мінімальний рівень вентиляції
    fan_baseline: float = 0.85 # фіксований рівень вентиляції у штатному режимі
    cop_chiller: float = 3.2 # коефіцієнт перетворення холодильної машини

    # Якість обслуговування (QoS)
    rho_max: float = 0.85 # гранична утилізація активного порту
    delay_budget_ms: float = 5.0 # бюджет середньої затримки в черзі, мс

    # Екологічна модель
    emission_factor: float = 0.40 # коефіцієнт викидів електромережі, кг CO2/кВт·год
    # (вхідний параметр; уточнюється за нац.
    кадастром/IEA)
```

```

def capacity_gbps(self) -> float:
    return self.num_ports * self.port_capacity_gbps

# -----
# 2. Модуль симуляції SNMP-телеметрії
# -----
def generate_snmp_telemetry(cfg: NodeConfig, samples: int = 48,
                             seed: int = 42) -> List[Dict]:
    """
    Генерує добовий профіль телеметрії вузла за принципом SNMP polling.

    Для кожного відліку формуються параметри, що відповідають реальним SNMP-OID:
        ifInOctets / ifOutOctets (1.3.6.1.2.1.2.2.1.10 / .16) – лічильники трафіку,
        ifOperStatus (.8), ifSpeed (.5) та утилізація кожного порту.
    Профіль навантаження – добовий (нічний спад, денний і вечірній пік) з
    випадковими відхиленнями, що імітують реальний характер трафіку.
    """
    rng = np.random.default_rng(seed)
    interval_s = 24 * 3600 / samples
    trace: List[Dict] = []

    # Добова форма сукупного коефіцієнта завантаження вузла (0..1)
    hours = np.linspace(0, 24, samples, endpoint=False)
    # два піки: ранковий (~10:00) і вечірній (~21:00)
    shape = (0.45 * np.exp(-((hours - 10) ** 2) / 18)
              + 0.75 * np.exp(-((hours - 21) ** 2) / 10)
              + 0.10)
    shape = shape / shape.max()
    load_frac = 0.08 + (0.62 - 0.08) * shape # сукупна утилізація у [0.08, 0.62]

    # стабільний розподіл частки трафіку між портами (деякі порти «гарячіші»)
    port_weight = rng.gamma(shape=2.0, scale=1.0, size=cfg.num_ports)
    port_weight = port_weight / port_weight.sum()

    cum_in = np.zeros(cfg.num_ports)
    cum_out = np.zeros(cfg.num_ports)

    for k, (h, lf) in enumerate(zip(hours, load_frac)):
        total_gbps = lf * cfg.capacity_gbps()
        noise = rng.normal(1.0, 0.06, size=cfg.num_ports).clip(0.5, 1.6)
        port_gbps = total_gbps * port_weight * noise
        port_gbps = np.minimum(port_gbps, cfg.port_capacity_gbps) # фізична межа порту

        util = port_gbps / cfg.port_capacity_gbps
        # лічильники октетів (Гбіт/с → байти за інтервал)
        delta_bytes = port_gbps * 1e9 / 8.0 * interval_s
        cum_in += delta_bytes * 0.52
        cum_out += delta_bytes * 0.48

        trace.append({
            "sample": k,
            "hour": float(h),
            "interval_s": interval_s,
            "total_load_frac": float(lf),
            "port_gbps": port_gbps.copy(),
            "util": util.copy(),
            "if_in_octets": cum_in.copy(),
            "if_out_octets": cum_out.copy(),
            "if_oper_status": np.ones(cfg.num_ports, dtype=int), # 1 = up
            "if_speed_gbps": np.full(cfg.num_ports, cfg.port_capacity_gbps),
        })
    return trace

# -----

```

```

# 3. Фізичні залежності: потужність портів, охолодження, температура, QoS
#
def _port_power(cfg: NodeConfig, load_gbps: float, state: str) -> float:
    """Потужність одного порту залежно від стану та завантаження."""
    if state == "sleep":
        return cfg.p_port_sleep_w
    if state == "1g": # знижена швидкість (ALR)
        rho = min(load_gbps / cfg.rate_downshift_gbps, 1.0)
        return cfg.p_port_idle_w + rho * (cfg.p_port_max_1g_w - cfg.p_port_idle_w)
    # повна швидкість 10G
    rho = min(load_gbps / cfg.port_capacity_gbps, 1.0)
    return cfg.p_port_idle_w + rho * (cfg.p_port_max_w - cfg.p_port_idle_w)

def _equipment_temperature(cfg: NodeConfig, q_it_w: float, fan: float,
                           chiller_on: bool) -> float:
    """Температура обладнання за поточного тепловиділення Q та рівня вентиляції."""
    inlet = cfg.t_ambient_c - (6.0 if chiller_on else 0.0) # чилер знижує вхідну t
    fan = max(fan, 1e-3)
    return inlet + q_it_w * cfg.r_th / fan

def _cooling_power(cfg: NodeConfig, fan: float, chiller_on: bool) -> float:
    """Електрична потужність охолодження (закон подібності вентилятора ~fan^3)."""
    p_fan = cfg.p_fan_max_w * (fan ** 3)
    p_chl = cfg.p_chiller_w if chiller_on else 0.0
    return p_fan + p_chl

def _select_cooling(cfg: NodeConfig, q_it_w: float,
                    adaptive: bool) -> Tuple[float, bool, float, float]:
    """
    Підбір режиму охолодження.
    adaptive=False — штатний фіксований рівень вентиляції (база порівняння);
    adaptive=True  — мінімальний рівень вентиляції, за якого T <= t_max
                    (лінійний пошук), за потреби вмикається чилер.
    Повертає (fan, chiller_on, p_cool, t_eq).
    """
    if not adaptive:
        fan = cfg.fan_baseline
        t = _equipment_temperature(cfg, q_it_w, fan, False)
        chiller = t > cfg.t_max_c
        if chiller:
            t = _equipment_temperature(cfg, q_it_w, fan, True)
        return fan, chiller, _cooling_power(cfg, fan, chiller), t

    # адаптивний пошук мінімальної вентиляції без чилера
    for fan in np.arange(cfg.fan_min, 1.0001, 0.01):
        t = _equipment_temperature(cfg, q_it_w, float(fan), False)
        if t <= cfg.t_max_c:
            return float(fan), False, _cooling_power(cfg, float(fan), False), t
    # не вистачає вільного охолодження → вмикаємо чилер і знову мінімізуємо вентиляцію
    for fan in np.arange(cfg.fan_min, 1.0001, 0.01):
        t = _equipment_temperature(cfg, q_it_w, float(fan), True)
        if t <= cfg.t_max_c:
            return float(fan), True, _cooling_power(cfg, float(fan), True), t
    fan = 1.0
    t = _equipment_temperature(cfg, q_it_w, fan, True)
    return fan, True, _cooling_power(cfg, fan, True), t

def _mean_delay_ms(cfg: NodeConfig, rho: float) -> float:
    """Оцінка середньої затримки в черзі (наближення M/M/1), мс."""
    rho = min(max(rho, 0.0), 0.999)
    serv_ms = (1500 * 8) / (cfg.port_capacity_gbps * 1e9) * 1e3 # час обслуговування
    кадру
    return serv_ms / (1.0 - rho)

```

```

def _metrics(cfg: NodeConfig, port_powers: np.ndarray, port_util: np.ndarray,
             active_mask: np.ndarray, p_cool: float, t_eq: float,
             total_gbps: float) -> Dict:
    """Зведені показники стану вузла за один відлік."""
    p_it = cfg.p_base_w + float(port_powers.sum())
    p_total = p_it + p_cool
    pue = p_total / p_it if p_it > 0 else float("nan")
    # енергоефективність за ITU-T L.1310: корисна продуктивність на ват, Гбіт/с/Вт
    ee = total_gbps / p_total if p_total > 0 else 0.0
    max_util = float(port_util[active_mask].max()) if active_mask.any() else 0.0
    max_delay = _mean_delay_ms(cfg, max_util)
    qos_ok = bool(max_util <= cfg.rho_max + 1e-9 and t_eq <= cfg.t_max_c + 1e-6
                  and max_delay <= cfg.delay_budget_ms + 1e-6)
    return {
        "active_ports": int(active_mask.sum()),
        "sleeping_ports": int((~active_mask).sum()),
        "p_it_w": p_it,
        "p_cool_w": p_cool,
        "p_total_w": p_total,
        "pue": pue,
        "energy_eff_gbps_per_w": ee,
        "max_util": max_util,
        "max_delay_ms": max_delay,
        "t_eq_c": t_eq,
        "qos_ok": qos_ok,
        "total_gbps": total_gbps,
    }

# -----
# 4. Аналіз штатного режиму (без оптимізації) – база порівняння
# -----
def analyze_state(cfg: NodeConfig, port_gbps: np.ndarray) -> Dict:
    """
    Штатний режим: усі порти активні на 10G і несуть власний трафік,
    охолодження працює на фіксованому рівні вентиляції.
    """
    n = cfg.num_ports
    powers = np.array([_port_power(cfg, float(g), "10g") for g in port_gbps])
    util = port_gbps / cfg.port_capacity_gbps
    active = np.ones(n, dtype=bool)
    q_it = cfg.p_base_w + float(powers.sum())
    fan, chiller, p_cool, t_eq = _select_cooling(cfg, q_it, adaptive=False)
    m = _metrics(cfg, powers, util, active, p_cool, t_eq, float(port_gbps.sum()))
    m.update({"fan": fan, "chiller_on": chiller, "mode": "baseline",
             "rate_1g_ports": 0})
    return m

# -----
# 5. Алгоритм оптимізації енергоспоживання
# -----
def optimize_state(cfg: NodeConfig, port_gbps: np.ndarray) -> Dict:
    """
    Алгоритмічна оптимізація (без жодних сталих множників):

    Крок 1. Консолідація трафіку методом First-Fit Decreasing: трафік
    перепаковується на мінімальну кількість активних портів так,
    щоб утилізація кожного не перевищувала rho_max. Незадіяні порти
    переводяться у режим сну.
    Крок 2. Адаптація швидкості: активні порти з низьким трафіком
    знижують швидкість до 1G (ALR), що зменшує їхню потужність.
    Крок 3. Керування охолодженням: для отриманого тепловиділення
    підбирається мінімальний рівень вентиляції, за якого
    температура не перевищує граничну (закон ~fan^3).
    Крок 4. Перевірка QoS і температури; за потреби – відкриття
    додаткового порту (відновлення допустимості).
    """
    n = cfg.num_ports
    cap_eff = cfg.rho_max * cfg.port_capacity_gbps # допустиме навантаження на порт

```

```

# — Крок 1: First-Fit Decreasing bin packing —
demands = sorted(((float(port_gbps[i]), i) for i in range(n)),
                  key=lambda x: -x[0])
bins_load: List[float] = [] # сумарне навантаження кожного активного порту
assign = np.full(n, -1, dtype=int) # у який активний порт спакований трафік
for d, _orig in demands:
    placed = False
    for b in range(len(bins_load)):
        if bins_load[b] + d <= cap_eff + 1e-9:
            bins_load[b] += d
            assign[_orig] = b
            placed = True
            break
    if not placed:
        bins_load.append(d)
        assign[_orig] = len(bins_load) - 1

active_count = len(bins_load)
active_count = min(active_count, n)
bin_load = np.array(bins_load)

# — Крок 2: стани й потужності портів —
powers = np.full(n, cfg.p_port_sleep_w)
util = np.zeros(n)
rate_1g = 0
for b in range(active_count):
    load = bin_load[b]
    state = "1g" if load <= cfg.rate_downshift_gbps else "10g"
    if state == "1g":
        rate_1g += 1
        util[b] = min(load / cfg.rate_downshift_gbps, 1.0)
    else:
        util[b] = min(load / cfg.port_capacity_gbps, 1.0)
    powers[b] = _port_power(cfg, float(load), state)

active = np.zeros(n, dtype=bool)
active[:active_count] = True

# — Крок 3: керування охолодженням —
q_it = cfg.p_base_w + float(powers.sum())
fan, chiller, p_cool, t_eq = _select_cooling(cfg, q_it, adaptive=True)

# — Крок 4: перевірка допустимості QoS —
m = _metrics(cfg, powers, util, active, p_cool, t_eq, float(port_gbps.sum()))
while (not m["qos_ok"]) and active_count < n:
    # відкриваємо ще один порт і перерозподіляємо найзавантаженіший
    active_count += 1
    active = np.zeros(n, dtype=bool)
    active[:active_count] = True
    # рівномірний перерозподіл сумарного трафіку між активними портами
    total = float(port_gbps.sum())
    per = total / active_count
    powers = np.full(n, cfg.p_port_sleep_w)
    util = np.zeros(n)
    rate_1g = 0
    for b in range(active_count):
        state = "1g" if per <= cfg.rate_downshift_gbps else "10g"
        if state == "1g":
            rate_1g += 1
            util[b] = min(per / cfg.rate_downshift_gbps, 1.0)
        else:
            util[b] = min(per / cfg.port_capacity_gbps, 1.0)
        powers[b] = _port_power(cfg, per, state)
    q_it = cfg.p_base_w + float(powers.sum())
    fan, chiller, p_cool, t_eq = _select_cooling(cfg, q_it, adaptive=True)
    m = _metrics(cfg, powers, util, active, p_cool, t_eq, total)

m.update({"fan": fan, "chiller_on": chiller, "mode": "optimized",
         "rate_1g_ports": rate_1g})

```

```

return m

# -----
# 6. Прогін добового профілю: база VS оптимізація, інтегральні показники
# -----
def run_daily(cfg: NodeConfig, trace: List[Dict]) -> Dict:
    base_rows, opt_rows = [], []
    for s in trace:
        g = s["port_gbps"]
        base_rows.append(analyze_state(cfg, g))
        opt_rows.append(optimize_state(cfg, g))

    interval_h = trace[0]["interval_s"] / 3600.0

    def energy_kwh(rows):
        return sum(r["p_total_w"] for r in rows) * interval_h / 1000.0

    e_base = energy_kwh(base_rows)
    e_opt = energy_kwh(opt_rows)
    c_base = e_base * cfg.emission_factor
    c_opt = e_opt * cfg.emission_factor

    summary = {
        "interval_h": interval_h,
        "samples": len(trace),
        "energy_base_kwh": e_base,
        "energy_opt_kwh": e_opt,
        "energy_saving_pct": 100.0 * (e_base - e_opt) / e_base,
        "carbon_base_kg": c_base,
        "carbon_opt_kg": c_opt,
        "carbon_saving_pct": 100.0 * (c_base - c_opt) / c_base,
        "annual_energy_saving_kwh": (e_base - e_opt) * 365.0,
        "annual_carbon_saving_kg": (c_base - c_opt) * 365.0,
        "avg_pue_base": float(np.mean([r["pue"] for r in base_rows])),
        "avg_pue_opt": float(np.mean([r["pue"] for r in opt_rows])),
        "avg_ee_base": float(np.mean([r["energy_eff_gbps_per_w"] for r in base_rows])),
        "avg_ee_opt": float(np.mean([r["energy_eff_gbps_per_w"] for r in opt_rows])),
        "qos_ok_all_opt": all(r["qos_ok"] for r in opt_rows),
        "max_temp_opt": max(r["t_eq_c"] for r in opt_rows),
    }
    return {"baseline": base_rows, "optimized": opt_rows, "summary": summary}

if __name__ == "__main__":
    cfg = NodeConfig()
    trace = generate_snmp_telemetry(cfg, samples=48)
    res = run_daily(cfg, trace)
    s = res["summary"]
    print("=" * 64)
    print("ВУЗЛО:", cfg.name, f"| портів: {cfg.num_ports} x {cfg.port_capacity_gbps:.0f}G")
    print("=" * 64)
    # миттєвий зріз: нічний (мін. трафік) і піковий
    loads = [t["total_load_frac"] for t in trace]
    i_min, i_max = int(np.argmin(loads)), int(np.argmax(loads))
    for label, i in (("НІЧНИЙ мінімум", i_min), ("ВЕЧІРНІЙ пік", i_max)):
        b, o = res["baseline"][i], res["optimized"][i]
        print(f"\n[{label}] год={trace[i]['hour']:.1f} завантаж={loads[i]*100:.0f}%")
        print(f" База : P={b['p_total_w']:6.1f} Вт активних={b['active_ports']:2d} "
              f"PUE={b['pue']:.3f} t={b['t_eq_c']:4.1f}°C QoS={'OK' if b['qos_ok'] else 'NOK'}")
        print(f" Опт. : P={o['p_total_w']:6.1f} Вт активних={o['active_ports']:2d} "
              f"сон={o['sleeping_ports']:2d} 1G={o['rate_1g_ports']:2d} "
              f"PUE={o['pue']:.3f} t={o['t_eq_c']:4.1f}°C QoS={'OK' if o['qos_ok'] else 'NOK'}")
        print(f" Економія миттєвої потужності: "
              f"{100*(b['p_total_w']-o['p_total_w'])/b['p_total_w']:.1f}%")
    print("\n" + "-" * 64)

```

```

print("ДОВОВІ ПІДСУМКИ")
print("-" * 64)
print(f"    Енергія за добу:    база {s['energy_base_kwh']:.3f} кВт·год | "
      f"опт. {s['energy_opt_kwh']:.3f} кВт·год | економія
{s['energy_saving_pct']:.1f}%")
print(f"    Вуглецевий слід:    база {s['carbon_base_kg']:.3f} кг | "
      f"опт. {s['carbon_opt_kg']:.3f} кг CO2 | економія
{s['carbon_saving_pct']:.1f}%")
print(f"    Середній PUE:        база {s['avg_pue_base']:.3f} → опт.
{s['avg_pue_opt']:.3f}")
print(f"    Середня EE:          база {s['avg_ee_base']:.3f} → опт. "
      f"{s['avg_ee_opt']:.3f} Гбіт/с/Вт")
print(f"    Економія на рік:      {s['annual_energy_saving_kwh']:.0f} кВт·год | "
      f"{s['annual_carbon_saving_kg']:.0f} кг CO2")
print(f"    QoS дотримано в усіх відліках (опт.): {s['qos_ok_all_opt']}")
print(f"    Макс. температура (опт.): {s['max_temp_opt']:.1f}°C (межа
{cfg.t_max_c:.0f}°C)")

```

ДОДАТОК Б

Лістинг вебзастосунок та моделей бази даних

Б.1 Вебзастосунок (app.py)

```

"""
app.py — вебзастосунок (Flask) системи оцінювання та оптимізації енергетичної
й екологічної ефективності телекомунікаційного вузла.

Маршрути:
GET / — дашборд (HTML);
GET /api/config — перелік конфігурацій вузлів;
POST /api/config — створення/оновлення конфігурації вузла;
POST /api/simulate — прогін моделювання (база + оптимізація), запис у БД;
GET /api/runs — перелік прогонів;
GET /api/runs/<run_id> — повні дані прогону для таблиць і графіків.

Запуск (середовище з PostgreSQL):
export DATABASE_URL=postgresql+psycopg://telecom:telecom@localhost:5432/telecom_eff
python app.py
"""
from flask import Flask, jsonify, request, render_template, abort
from sqlalchemy import select, desc

from db import SessionLocal, init_db
from models import NodeConfigRow, Run, Measurement
from core import NodeConfig, generate_snmp_telemetry, run_daily

app = Flask(__name__)

# — допоміжні перетворення —
def _cfg_from_row(row: NodeConfigRow) -> NodeConfig:
    return NodeConfig(
        name=row.name, network_type=row.network_type, num_ports=row.num_ports,
        port_capacity_gbps=row.port_capacity_gbps, p_base_w=row.p_base_w,
        p_port_idle_w=row.p_port_idle_w, p_port_max_w=row.p_port_max_w,
        p_port_sleep_w=row.p_port_sleep_w, rho_max=row.rho_max,
        t_ambient_c=row.t_ambient_c, t_max_c=row.t_max_c,
        emission_factor=row.emission_factor,
    )

def _ensure_default_node(s) -> NodeConfigRow:
    node = s.scalars(select(NodeConfigRow).order_by(NodeConfigRow.id)).first()
    if node is None:
        c = NodeConfig()
        node = NodeConfigRow(
            name=c.name, network_type=c.network_type, num_ports=c.num_ports,
            port_capacity_gbps=c.port_capacity_gbps, p_base_w=c.p_base_w,
            p_port_idle_w=c.p_port_idle_w, p_port_max_w=c.p_port_max_w,
            p_port_sleep_w=c.p_port_sleep_w, rho_max=c.rho_max,
            t_ambient_c=c.t_ambient_c, t_max_c=c.t_max_c,
            emission_factor=c.emission_factor,
        )
        s.add(node); s.commit()
    return node

# — сторінка —
@app.get("/")
def index():
    return render_template("dashboard.html")

```

```

# — конфігурації —
@app.get("/api/config")
def list_config():
    with SessionLocal() as s:
        _ensure_default_node(s)
        rows = s.scalars(select(NodeConfigRow).order_by(NodeConfigRow.id)).all()
        return jsonify([
            {
                "id": r.id, "name": r.name, "network_type": r.network_type,
                "num_ports": r.num_ports, "port_capacity_gbps": r.port_capacity_gbps,
                "p_base_w": r.p_base_w, "rho_max": r.rho_max,
                "t_ambient_c": r.t_ambient_c, "t_max_c": r.t_max_c,
                "emission_factor": r.emission_factor,
            } for r in rows])

@app.post("/api/config")
def upsert_config():
    d = request.get_json(force=True)
    with SessionLocal() as s:
        if d.get("id"):
            row = s.get(NodeConfigRow, int(d["id"])) or abort(404)
        else:
            row = NodeConfigRow(name=d.get("name", "НОВИЙ ВУЗОЛ"),
                                network_type=d.get("network_type", "Ethernet/IP"))
            s.add(row)
        for f in ("name", "network_type", "num_ports", "port_capacity_gbps",
                  "p_base_w", "p_port_idle_w", "p_port_max_w", "p_port_sleep_w",
                  "rho_max", "t_ambient_c", "t_max_c", "emission_factor"):
            if f in d and d[f] is not None:
                setattr(row, f, d[f])
        s.commit()
        return jsonify({"id": row.id}), 200

# — прогін моделювання —
@app.post("/api/simulate")
def simulate():
    d = request.get_json(silent=True) or {}
    samples = int(d.get("samples", 48))
    with SessionLocal() as s:
        node = (s.get(NodeConfigRow, int(d["node_id"]))
                 if d.get("node_id") else _ensure_default_node(s))
        if node is None:
            abort(404)
        cfg = _cfg_from_row(node)
        trace = generate_snmp_telemetry(cfg, samples=samples)
        res = run_daily(cfg, trace)
        summ = res["summary"]

    run = Run(
        node_id=node.id, samples=samples,
        energy_base_kwh=summ["energy_base_kwh"],
        energy_opt_kwh=summ["energy_opt_kwh"],
        energy_saving_pct=summ["energy_saving_pct"],
        carbon_base_kg=summ["carbon_base_kg"],
        carbon_opt_kg=summ["carbon_opt_kg"],
        carbon_saving_pct=summ["carbon_saving_pct"],
        avg_pue_base=summ["avg_pue_base"], avg_pue_opt=summ["avg_pue_opt"],
        annual_energy_saving_kwh=summ["annual_energy_saving_kwh"],
        annual_carbon_saving_kg=summ["annual_carbon_saving_kg"],
        qos_ok_all_opt=summ["qos_ok_all_opt"],
    )
    s.add(run); s.flush()

    for mode, rows in (("baseline", res["baseline"]), ("optimized",
res["optimized"])):
        for k, r in enumerate(rows):

```

```

        s.add(Measurement(
            run_id=run.id, sample_idx=k, hour=trace[k]["hour"], mode=mode,
            load_frac=trace[k]["total_load_frac"],
            active_ports=r["active_ports"], sleeping_ports=r["sleeping_ports"],
            p_it_w=r["p_it_w"], p_cool_w=r["p_cool_w"], p_total_w=r["p_total_w"],
            pue=r["pue"], energy_eff=r["energy_eff_gbps_per_w"],
            max_util=r["max_util"], t_eq_c=r["t_eq_c"], qos_ok=r["qos_ok"]))
    s.commit()
    return jsonify(_run_payload(s, run.id))

@app.get("/api/runs")
def list_runs():
    with SessionLocal() as s:
        rows = s.scalars(select(Run).order_by(desc(Run.created_at))).all()
        return jsonify([{"id": r.id, "created_at": r.created_at.isoformat(),
            "energy_saving_pct": r.energy_saving_pct,
            "carbon_saving_pct": r.carbon_saving_pct} for r in rows])

@app.get("/api/runs/<int:run_id>")
def get_run(run_id: int):
    with SessionLocal() as s:
        if s.get(Run, run_id) is None:
            abort(404)
        return jsonify(_run_payload(s, run_id))

def _run_payload(s, run_id: int) -> dict:
    run = s.get(Run, run_id)
    ms = s.scalars(select(Measurement).where(Measurement.run_id == run_id)
        .order_by(Measurement.sample_idx)).all()

    def series(mode, key):
        return [getattr(m, key) for m in ms if m.mode == mode]

    hours = [m.hour for m in ms if m.mode == "baseline"]
    return {
        "run_id": run.id,
        "summary": {
            "energy_base_kwh": run.energy_base_kwh, "energy_opt_kwh": run.energy_opt_kwh,
            "energy_saving_pct": run.energy_saving_pct,
            "carbon_base_kg": run.carbon_base_kg, "carbon_opt_kg": run.carbon_opt_kg,
            "carbon_saving_pct": run.carbon_saving_pct,
            "avg_pue_base": run.avg_pue_base, "avg_pue_opt": run.avg_pue_opt,
            "annual_energy_saving_kwh": run.annual_energy_saving_kwh,
            "annual_carbon_saving_kg": run.annual_carbon_saving_kg,
            "qos_ok_all_opt": run.qos_ok_all_opt,
        },
        "hours": hours,
        "load": series("baseline", "load_frac"),
        "p_base": series("baseline", "p_total_w"),
        "p_opt": series("optimized", "p_total_w"),
        "ap_base": series("baseline", "active_ports"),
        "ap_opt": series("optimized", "active_ports"),
        "pue_base": series("baseline", "pue"),
        "pue_opt": series("optimized", "pue"),
        "t_opt": series("optimized", "t_eq_c"),
    }

if __name__ == "__main__":
    init_db()
    with SessionLocal() as s:
        _ensure_default_node(s)
    print("Сервер запущено: http://127.0.0.1:5000")
    app.run(host="127.0.0.1", port=5000, debug=True)

```

Б.2 Моделі бази даних (models.py)

"""

models.py – ORM-моделі бази даних PostgreSQL.

Сутності:

NodeConfigRow – конфігурація телекомунікаційного вузла;
 Run – прогін моделювання (з інтегральними показниками доби);
 Measurement – посамплові показники режимів «база» та «оптимізація».

"""

```
from datetime import datetime
from sqlalchemy import (Integer, String, Float, Boolean, DateTime, ForeignKey)
from sqlalchemy.orm import Mapped, mapped_column, relationship
from db import Base
```

```
class NodeConfigRow(Base):
    __tablename__ = "nodes"
```

```
    id: Mapped[int] = mapped_column(Integer, primary_key=True)
    name: Mapped[str] = mapped_column(String(120))
    network_type: Mapped[str] = mapped_column(String(120))
    num_ports: Mapped[int] = mapped_column(Integer, default=48)
    port_capacity_gbps: Mapped[float] = mapped_column(Float, default=10.0)
    p_base_w: Mapped[float] = mapped_column(Float, default=60.0)
    p_port_idle_w: Mapped[float] = mapped_column(Float, default=1.5)
    p_port_max_w: Mapped[float] = mapped_column(Float, default=3.5)
    p_port_sleep_w: Mapped[float] = mapped_column(Float, default=0.10)
    rho_max: Mapped[float] = mapped_column(Float, default=0.85)
    t_ambient_c: Mapped[float] = mapped_column(Float, default=24.0)
    t_max_c: Mapped[float] = mapped_column(Float, default=40.0)
    emission_factor: Mapped[float] = mapped_column(Float, default=0.40)
    created_at: Mapped[datetime] = mapped_column(DateTime, default=datetime.utcnow)

    runs: Mapped[list["Run"]] = relationship(back_populates="node",
                                              cascade="all, delete-orphan")
```

```
class Run(Base):
    __tablename__ = "runs"
```

```
    id: Mapped[int] = mapped_column(Integer, primary_key=True)
    node_id: Mapped[int] = mapped_column(ForeignKey("nodes.id"))
    created_at: Mapped[datetime] = mapped_column(DateTime, default=datetime.utcnow)
    samples: Mapped[int] = mapped_column(Integer)

    energy_base_kwh: Mapped[float] = mapped_column(Float)
    energy_opt_kwh: Mapped[float] = mapped_column(Float)
    energy_saving_pct: Mapped[float] = mapped_column(Float)
    carbon_base_kg: Mapped[float] = mapped_column(Float)
    carbon_opt_kg: Mapped[float] = mapped_column(Float)
    carbon_saving_pct: Mapped[float] = mapped_column(Float)
    avg_pue_base: Mapped[float] = mapped_column(Float)
    avg_pue_opt: Mapped[float] = mapped_column(Float)
    annual_energy_saving_kwh: Mapped[float] = mapped_column(Float)
    annual_carbon_saving_kg: Mapped[float] = mapped_column(Float)
    qos_ok_all_opt: Mapped[bool] = mapped_column(Boolean)

    node: Mapped["NodeConfigRow"] = relationship(back_populates="runs")
    measurements: Mapped[list["Measurement"]] = relationship(
        back_populates="run", cascade="all, delete-orphan")
```

```
class Measurement(Base):
    __tablename__ = "measurements"
```

```

id: Mapped[int] = mapped_column(Integer, primary_key=True)
run_id: Mapped[int] = mapped_column(ForeignKey("runs.id"))
sample_idx: Mapped[int] = mapped_column(Integer)
hour: Mapped[float] = mapped_column(Float)
mode: Mapped[str] = mapped_column(String(16)) # 'baseline' | 'optimized'
load_frac: Mapped[float] = mapped_column(Float)
active_ports: Mapped[int] = mapped_column(Integer)
sleeping_ports: Mapped[int] = mapped_column(Integer)
p_it_w: Mapped[float] = mapped_column(Float)
p_cool_w: Mapped[float] = mapped_column(Float)
p_total_w: Mapped[float] = mapped_column(Float)
pue: Mapped[float] = mapped_column(Float)
energy_eff: Mapped[float] = mapped_column(Float)
max_util: Mapped[float] = mapped_column(Float)
t_eq_c: Mapped[float] = mapped_column(Float)
qos_ok: Mapped[bool] = mapped_column(Boolean)

run: Mapped["Run"] = relationship(back_populates="measurements")

```

Б.3 Підключення до бази даних (db.py)

"""
db.py — рівень доступу до бази даних (SQLAlchemy 2.0).

За замовчуванням використовується PostgreSQL. Рядок підключення задається змінною середовища DATABASE_URL, наприклад:

```
postgresql+psycopg://telecom:telecom@localhost:5432/telecom_eff
```

Для швидкої локальної перевірки без сервера PostgreSQL можна тимчасово задати DATABASE_URL=sqlite:///telecom_eff.db (SQLAlchemy підтримує SQLite штатно).
"""

```

import os
from sqlalchemy import create_engine
from sqlalchemy.orm import DeclarativeBase, sessionmaker

```

```

DATABASE_URL = os.environ.get(
    "DATABASE_URL",
    "postgresql+psycopg://telecom:telecom@localhost:5432/telecom_eff",
)

```

```

# echo=False у проді; pool_pre_ping тримає з'єднання живими
engine = create_engine(DATABASE_URL, echo=False, future=True, pool_pre_ping=True)
SessionLocal = sessionmaker(bind=engine, autoflush=False, expire_on_commit=False)

```

```

class Base(DeclarativeBase):
    pass

```

```

def init_db() -> None:
    """Створює всі таблиці згідно з ORM-моделями."""
    from models import NodeConfigRow, Run, Measurement # noqa: F401
    Base.metadata.create_all(engine)

```

ДОДАТОК В

Схема бази даних та побудова графічних матеріалів

В.1 Схема бази даних PostgreSQL (schema.sql)

-- schema.sql – структура бази даних PostgreSQL для системи оцінювання
 -- енергетичної та екологічної ефективності телекомунікаційного вузла.
 -- Відповідає ORM-моделям (models.py). Створення схеми також виконується
 -- автоматично командою init_db() під час першого запуску застосунку.

```
CREATE TABLE IF NOT EXISTS nodes (
  id          SERIAL PRIMARY KEY,
  name        VARCHAR(120) NOT NULL,
  network_type VARCHAR(120) NOT NULL,
  num_ports   INTEGER NOT NULL DEFAULT 48,
  port_capacity_gbps DOUBLE PRECISION NOT NULL DEFAULT 10.0,
  p_base_w    DOUBLE PRECISION NOT NULL DEFAULT 60.0,
  p_port_idle_w DOUBLE PRECISION NOT NULL DEFAULT 1.5,
  p_port_max_w DOUBLE PRECISION NOT NULL DEFAULT 3.5,
  p_port_sleep_w DOUBLE PRECISION NOT NULL DEFAULT 0.10,
  rho_max     DOUBLE PRECISION NOT NULL DEFAULT 0.85,
  t_ambient_c DOUBLE PRECISION NOT NULL DEFAULT 24.0,
  t_max_c     DOUBLE PRECISION NOT NULL DEFAULT 40.0,
  emission_factor DOUBLE PRECISION NOT NULL DEFAULT 0.40,
  created_at  TIMESTAMP NOT NULL DEFAULT NOW()
);

CREATE TABLE IF NOT EXISTS runs (
  id          SERIAL PRIMARY KEY,
  node_id     INTEGER NOT NULL REFERENCES nodes(id) ON DELETE CASCADE,
  created_at  TIMESTAMP NOT NULL DEFAULT NOW(),
  samples     INTEGER NOT NULL,
  energy_base_kwh DOUBLE PRECISION NOT NULL,
  energy_opt_kwh DOUBLE PRECISION NOT NULL,
  energy_saving_pct DOUBLE PRECISION NOT NULL,
  carbon_base_kg DOUBLE PRECISION NOT NULL,
  carbon_opt_kg DOUBLE PRECISION NOT NULL,
  carbon_saving_pct DOUBLE PRECISION NOT NULL,
  avg_pue_base DOUBLE PRECISION NOT NULL,
  avg_pue_opt DOUBLE PRECISION NOT NULL,
  annual_energy_saving_kwh DOUBLE PRECISION NOT NULL,
  annual_carbon_saving_kg DOUBLE PRECISION NOT NULL,
  qos_ok_all_opt BOOLEAN NOT NULL
);

CREATE TABLE IF NOT EXISTS measurements (
  id          SERIAL PRIMARY KEY,
  run_id      INTEGER NOT NULL REFERENCES runs(id) ON DELETE CASCADE,
  sample_idx  INTEGER NOT NULL,
  hour       DOUBLE PRECISION NOT NULL,
  mode       VARCHAR(16) NOT NULL,
  load_frac  DOUBLE PRECISION NOT NULL,
  active_ports INTEGER NOT NULL,
  sleeping_ports INTEGER NOT NULL,
  p_it_w     DOUBLE PRECISION NOT NULL,
  p_cool_w   DOUBLE PRECISION NOT NULL,
  p_total_w  DOUBLE PRECISION NOT NULL,
  pue        DOUBLE PRECISION NOT NULL,
  energy_eff DOUBLE PRECISION NOT NULL,
  max_util   DOUBLE PRECISION NOT NULL,
  t_eq_c     DOUBLE PRECISION NOT NULL,
  qos_ok     BOOLEAN NOT NULL
);

CREATE INDEX IF NOT EXISTS idx_measurements_run ON measurements(run_id);
CREATE INDEX IF NOT EXISTS idx_runs_node ON runs(node_id);
```

В.2 Побудова графіків результатів (figures.py)

```

"""
figures.py — побудова графічних матеріалів роботи з реальних результатів
обчислювального ядра (core.py). Усі дані обчислені, не імітовані.
"""
import numpy as np
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
from core import NodeConfig, generate_snmp_telemetry, run_daily

plt.rcParams.update({
    "font.family": "DejaVu Sans",
    "font.size": 11,
    "axes.grid": True,
    "grid.alpha": 0.3,
    "figure.dpi": 130,
})

C_BASE = "#c0392b" # база — червоний
C_OPT = "#1f7a4d" # оптимізація — зелений
C_AUX = "#2c3e50"

cfg = NodeConfig()
trace = generate_snmp_telemetry(cfg, samples=48)
res = run_daily(cfg, trace)
base, opt, s = res["baseline"], res["optimized"], res["summary"]

hours = np.array([t["hour"] for t in trace])
load = np.array([t["total_load_frac"] for t in trace]) * 100
p_base = np.array([r["p_total_w"] for r in base])
p_opt = np.array([r["p_total_w"] for r in opt])
ap_base = np.array([r["active_ports"] for r in base])
ap_opt = np.array([r["active_ports"] for r in opt])
pue_base = np.array([r["pue"] for r in base])
pue_opt = np.array([r["pue"] for r in opt])

OUT = "figures"
import os
os.makedirs(OUT, exist_ok=True)

# — Рисунок 1: добовий профіль повної потужності —
fig, ax = plt.subplots(figsize=(8, 4.4))
ax.plot(hours, p_base, color=C_BASE, lw=2, marker="o", ms=3, label="Штатний режим (без оптимізації)")
ax.plot(hours, p_opt, color=C_OPT, lw=2, marker="s", ms=3, label="Після оптимізації")
ax.fill_between(hours, p_opt, p_base, color=C_OPT, alpha=0.10)
ax.set_xlabel("Час доби, год")
ax.set_ylabel("Повна спожив. потужність вузла, Вт")
ax.set_xlim(0, 24); ax.set_xticks(range(0, 25, 3))
ax2 = ax.twinx()
ax2.plot(hours, load, color=C_AUX, lw=1.2, ls="--", alpha=0.6, label="Завантаження вузла, %")
ax2.set_ylabel("Завантаження вузла, %"); ax2.set_ylim(0, 100); ax2.grid(False)
l1, b1 = ax.get_legend_handles_labels(); l2, b2 = ax2.get_legend_handles_labels()
ax.legend(l1 + l2, b1 + b2, loc="upper left", fontsize=9)
ax.set_title("Добовий профіль енергоспоживання телекомунікаційного вузла")
fig.tight_layout(); fig.savefig(f"{OUT}/fig_power_profile.png"); plt.close(fig)

# — Рисунок 2: кількість активних портів протягом доби —
fig, ax = plt.subplots(figsize=(8, 4.0))
ax.step(hours, ap_base, where="mid", color=C_BASE, lw=2, label="Штатний режим (усі порти активні)")
ax.step(hours, ap_opt, where="mid", color=C_OPT, lw=2, label="Після оптимізації (консолідація)")

```

```

ax.fill_between(hours, ap_opt, step="mid", color=C_OPT, alpha=0.12)
ax.set_xlabel("Час доби, год"); ax.set_ylabel("Кількість активних портів")
ax.set_xlim(0, 24); ax.set_xticks(range(0, 25, 3)); ax.set_ylim(0, cfg.num_ports + 2)
ax.legend(loc="center right", fontsize=9)
ax.set_title("Кількість активних портів вузла протягом доби")
fig.tight_layout(); fig.savefig(f"{OUT}/fig_active_ports.png"); plt.close(fig)

# — Рисунок 3: енергоспоживання та вуглецевий слід за добу —
fig, axes = plt.subplots(1, 2, figsize=(8.6, 4.2))
labels = ["Штатний\нрежим", "Після\ноптимізації"]
e_vals = [s["energy_base_kwh"], s["energy_opt_kwh"]]
c_vals = [s["carbon_base_kg"], s["carbon_opt_kg"]]
bars1 = axes[0].bar(labels, e_vals, color=[C_BASE, C_OPT], width=0.6)
axes[0].set_ylabel("Енергоспоживання за добу, кВт·год")
axes[0].set_title(f"Енергія (-{s['energy_saving_pct']:.1f}%)")
for b, v in zip(bars1, e_vals):
    axes[0].text(b.get_x() + b.get_width() / 2, v, f"{v:.2f}", ha="center", va="bottom",
    fontsize=10)
bars2 = axes[1].bar(labels, c_vals, color=[C_BASE, C_OPT], width=0.6)
axes[1].set_ylabel("Вуглецевий слід за добу, кг CO\u2082")
axes[1].set_title(f"Викиди CO\u2082 (-{s['carbon_saving_pct']:.1f}%)")
for b, v in zip(bars2, c_vals):
    axes[1].text(b.get_x() + b.get_width() / 2, v, f"{v:.2f}", ha="center", va="bottom",
    fontsize=10)
fig.suptitle("Добове енергоспоживання та вуглецевий слід вузла", y=1.02)
fig.tight_layout(); fig.savefig(f"{OUT}/fig_energy_carbon.png", bbox_inches="tight");
plt.close(fig)

# — Рисунок 4: PUE протягом доби —
fig, ax = plt.subplots(figsize=(8, 4.0))
ax.plot(hours, pue_base, color=C_BASE, lw=2, marker="o", ms=3, label="Штатний режим")
ax.plot(hours, pue_opt, color=C_OPT, lw=2, marker="s", ms=3, label="Після оптимізації")
ax.axhline(1.0, color=C_AUX, ls=":", lw=1, alpha=0.7, label="Ідеальний PUE = 1,0")
ax.set_xlabel("Час доби, год"); ax.set_ylabel("Коефіцієнт PUE")
ax.set_xlim(0, 24); ax.set_xticks(range(0, 25, 3))
ax.legend(loc="upper right", fontsize=9)
ax.set_title("Ефективність використання енергії (PUE) протягом доби")
fig.tight_layout(); fig.savefig(f"{OUT}/fig_pue.png"); plt.close(fig)

print("Figures saved:")
for f in ["fig_power_profile.png", "fig_active_ports.png", "fig_energy_carbon.png",
"fig_pue.png"]:
    p = f"{OUT}/{f}"
    print(f" {f}: {os.path.getsize(p)} bytes")

```

В.3 Перелік залежностей проєкту (requirements.txt)

```

Flask==3.0.3
SQLAlchemy==2.0.31
psycopg[binary]==3.2.1
numpy==1.26.4
matplotlib==3.8.4

```

ДОДАТОК Г

Лістинг клієнтської частини (вебдашборд)

Г.1 Шаблон дашборда (dashboard.html)

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Оцінювання енергетичної та екологічної ефективності телекомунікаційного
вузла</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
  <script
src="https://cdn.jsdelivr.net/npm/chart.js@4.4.1/dist/chart.umd.min.js"></script>
</head>
<body>
  <header>
    <h1>Система оцінювання та оптимізації енергетичної й екологічної ефективності
телекомунікаційного вузла</h1>
  </header>

  <main>
    <section class="panel">
      <div class="controls">
        <label>Кількість відліків телеметрії за добу
        <select id="samples">
          <option value="24">24</option>
          <option value="48" selected>48</option>
          <option value="96">96</option>
        </select>
        </label>
        <button id="run-btn">Запустити моделювання</button>
        <span id="status"></span>
      </div>
    </section>

    <section id="kpis" class="kpi-grid"></section>

    <section class="charts">
      <div class="chart-card">
        <h3>Добовий профіль повної потужності, Вт</h3>
        <canvas id="powerChart" height="150"></canvas>
      </div>
      <div class="chart-card">
        <h3>Кількість активних портів протягом доби</h3>
        <canvas id="portsChart" height="150"></canvas>
      </div>
      <div class="chart-card">
        <h3>Коефіцієнт PUE протягом доби</h3>
        <canvas id="pueChart" height="150"></canvas>
      </div>
    </section>

    <section class="table-card">
      <h3>Посамплові результати моделювання</h3>
      <table id="results">
        <thead>
          <tr>
            <th>Год</th><th>Завант., %</th>
            <th>P база, Вт</th><th>P опт., Вт</th>
            <th>Активних (база/опт.)</th>
            <th>PUE база</th><th>PUE опт.</th>
            <th>t опт., °C</th><th>QoS опт.</th>
          </tr>
        </thead>
        <tbody></tbody>
      </table>
    </section>
  </main>

```

```

    </table>
  </section>
</main>

<script src="{ url_for('static', filename='app.js') }"></script>
</body>
</html>

```

Г.2 Клієнтський скрипт (app.js)

```

"use strict";

let powerChart, portsChart, pueChart;

function fmt(x, d = 2) { return Number(x).toFixed(d); }

function renderKPIs(s) {
  const cards = [
    ["Енергія за добу (база)", fmt(s.energy_base_kwh) + " кВт·год", ""],
    ["Енергія за добу (опт.)", fmt(s.energy_opt_kwh) + " кВт·год", "good"],
    ["Економія енергії", fmt(s.energy_saving_pct, 1) + " %", "good"],
    ["Вуглецевий слід (опт.)", fmt(s.carbon_opt_kg) + " кг CO2", "good"],
    ["Економія CO2", fmt(s.carbon_saving_pct, 1) + " %", "good"],
    ["Середній PUE", fmt(s.avg_pue_base, 3) + " → " + fmt(s.avg_pue_opt, 3), ""],
    ["Економія за рік", Math.round(s.annual_energy_saving_kwh) + " кВт·год", "good"],
    ["QoS дотримано", s.qos_ok_all_opt ? "так" : "ні", s.qos_ok_all_opt ? "good" :
"bad"],
  ];
  document.getElementById("kpis").innerHTML = cards.map(
    ([t, v, c]) => `<div class="kpi ${c}"><div class="kpi-val">${v}</div><div class="kpi-
lbl">${t}</div></div>`
  ).join("");
}

function lineChart(ctx, labels, datasets, yTitle) {
  return new Chart(ctx, {
    type: "line",
    data: { labels, datasets },
    options: {
      responsive: true, interaction: { mode: "index", intersect: false },
      scales: {
        x: { title: { display: true, text: "Час доби, год" } },
        y: { title: { display: true, text: yTitle } },
      },
    },
  });
}

function renderCharts(d) {
  const labels = d.hours.map(h => fmt(h, 0));
  [powerChart, portsChart, pueChart].forEach(c => c && c.destroy());

  powerChart = lineChart(document.getElementById("powerChart"), labels, [
    { label: "Штатний режим", data: d.p_base, borderColor: "#c0392b", backgroundColor:
"rgba(192,57,43,.1)", tension: .3 },
    { label: "Після оптимізації", data: d.p_opt, borderColor: "#1f7a4d", backgroundColor:
"rgba(31,122,77,.15)", fill: true, tension: .3 },
  ], "Потужність, Вт");

  portsChart = lineChart(document.getElementById("portsChart"), labels, [
    { label: "Штатний режим", data: d.ap_base, borderColor: "#c0392b", stepped: true },
    { label: "Після оптимізації", data: d.ap_opt, borderColor: "#1f7a4d",
backgroundColor: "rgba(31,122,77,.15)", fill: true, stepped: true },
  ], "Активні порти");

  pueChart = lineChart(document.getElementById("pueChart"), labels, [
    { label: "Штатний режим", data: d.pue_base, borderColor: "#c0392b", tension: .3 },

```

```

    { label: "Після оптимізації", data: d.pue_opt, borderColor: "#1f7a4d", tension: .3 },
  ], "PUE");
}

function renderTable(d) {
  const tb = document.querySelector("#results tbody");
  tb.innerHTML = d.hours.map((h, i) => `
    <tr>
      <td>${fmt(h, 0)}</td>
      <td>${fmt(d.load[i] * 100, 0)}</td>
      <td>${fmt(d.p_base[i], 1)}</td>
      <td>${fmt(d.p_opt[i], 1)}</td>
      <td>${d.ap_base[i]} / ${d.ap_opt[i]}</td>
      <td>${fmt(d.pue_base[i], 3)}</td>
      <td>${fmt(d.pue_opt[i], 3)}</td>
      <td>${fmt(d.t_opt[i], 1)}</td>
      <td>${d.t_opt[i] <= 40 ? "OK" : "-"}</td>
    </tr>`).join("");
}

async function runSimulation() {
  const status = document.getElementById("status");
  status.textContent = "Виконується моделювання...";
  document.getElementById("run-btn").disabled = true;
  try {
    const samples = parseInt(document.getElementById("samples").value, 10);
    const resp = await fetch("/api/simulate", {
      method: "POST", headers: { "Content-Type": "application/json" },
      body: JSON.stringify({ samples })
    });
    if (!resp.ok) throw new Error("HTTP " + resp.status);
    const d = await resp.json();
    renderKPIs(d.summary);
    renderCharts(d);
    renderTable(d);
    status.textContent = "Готово (прогін №" + d.run_id + ").";
  } catch (e) {
    status.textContent = "Помилка: " + e.message;
  } finally {
    document.getElementById("run-btn").disabled = false;
  }
}

document.getElementById("run-btn").addEventListener("click", runSimulation);

```

Г.3 Каскадні таблиці стилів (style.css)

```

:root {
  --base: #c0392b;
  --opt: #1f7a4d;
  --ink: #1f2933;
  --muted: #6b7280;
  --line: #e3e8ef;
  --bg: #f5f7fa;
}
* { box-sizing: border-box; }
body {
  margin: 0; font-family: "Segoe UI", Roboto, Arial, sans-serif;
  color: var(--ink); background: var(--bg);
}
header {
  background: #102a43; color: #fff; padding: 18px 28px;
}
header h1 { margin: 0; font-size: 18px; font-weight: 600; line-height: 1.4; }
main { max-width: 1180px; margin: 0 auto; padding: 22px; }
.panel { margin-bottom: 18px; }
.controls {
  display: flex; align-items: center; gap: 16px; flex-wrap: wrap;
  background: #fff; padding: 16px 18px; border: 1px solid var(--line); border-radius:
10px;

```

```

}
.controls label { font-size: 14px; color: var(--muted); display: flex; flex-direction:
column; gap: 4px; }
.controls select { padding: 6px 10px; border: 1px solid var(--line); border-radius: 6px;
font-size: 14px; }
#run-btn {
  background: var(--opt); color: #fff; border: 0; padding: 11px 22px;
  border-radius: 8px; font-size: 15px; font-weight: 600; cursor: pointer;
}
#run-btn:disabled { background: #9aa5b1; cursor: default; }
#status { color: var(--muted); font-size: 14px; }

.kpi-grid {
  display: grid; grid-template-columns: repeat(4, 1fr); gap: 14px; margin-bottom: 20px;
}
.kpi {
  background: #fff; border: 1px solid var(--line); border-left: 4px solid #9aa5b1;
  border-radius: 10px; padding: 14px 16px;
}
.kpi.good { border-left-color: var(--opt); }
.kpi.bad { border-left-color: var(--base); }
.kpi-val { font-size: 22px; font-weight: 700; }
.kpi-lbl { font-size: 12.5px; color: var(--muted); margin-top: 4px; }

.charts { display: grid; grid-template-columns: 1fr 1fr; gap: 18px; margin-bottom: 20px;
}
.chart-card, .table-card {
  background: #fff; border: 1px solid var(--line); border-radius: 10px; padding: 16px
18px;
}
.chart-card:first-child { grid-column: 1 / -1; }
.chart-card h3, .table-card h3 { margin: 0 0 12px; font-size: 15px; }

table { width: 100%; border-collapse: collapse; font-size: 13px; }
th, td { padding: 7px 9px; text-align: center; border-bottom: 1px solid var(--line); }
th { background: #f0f4f8; color: var(--muted); font-weight: 600; }
tbody tr:hover { background: #f9fbfd; }

@media (max-width: 820px) {
  .kpi-grid { grid-template-columns: repeat(2, 1fr); }
  .charts { grid-template-columns: 1fr; }
}

```

ДОДАТОК Д

Інструкція з розгортання та використання системи

Для розгортання системи потрібні інтерпретатор мови Python версії 3.10 або новішої та система керування базами даних PostgreSQL версії 14 або новішої. Усі залежності прикладної частини перелічено у файлі requirements.txt, наведеному в додатку В, і встановлюються однією командою менеджера пакетів після створення та активації віртуального середовища.

Перед першим запуском у PostgreSQL створюють користувача та базу даних, від імені яких працюватиме застосунок, після чого рядок підключення передають через змінну середовища DATABASE_URL у форматі, що містить протокол, ім'я користувача, пароль, адресу сервера, порт і назву бази даних. Схему бази даних може бути створено автоматично під час першого запуску застосунку або вручну виконанням сценарію schema.sql, наведеного в додатку В.

Запуск застосунку здійснюється виконанням головного модуля, після чого вбудований вебсервер стає доступним за локальною адресою <http://127.0.0.1:5000> (порт 5000 задано в головному модулі app.py і за потреби може бути змінений). У вебпереглядачі відкривають сторінку дашборда, обирають кількість відліків телеметрії за добу та запускають моделювання. Система виконує добове моделювання роботи вузла у штатному та оптимізованому режимах, зберігає результати в базі даних і відображає їх у вигляді карток показників, графіків добового профілю потужності, кількості активних портів та коефіцієнта PUE, а також таблиці посамплових результатів.

Доступ до збережених результатів забезпечує програмний інтерфейс: окремі маршрути повертають перелік конфігурацій вузлів, перелік виконаних прогонів та повні дані конкретного прогону у форматі обміну даними, придатному для подальшого опрацювання. Така організація дає змогу інтегрувати систему з наявними засобами моніторингу та автоматизувати збирання показників енергоефективності для сукупності вузлів мережі.

Для перевірки логіки роботи без розгортання сервера бази даних передбачено можливість тимчасового використання вбудованої файлової бази даних шляхом відповідного задання змінної середовища `DATABASE_URL`. Робочою системою керування базами даних проєкту є PostgreSQL, що забезпечує надійне зберігання даних та цілісність за одночасного доступу.

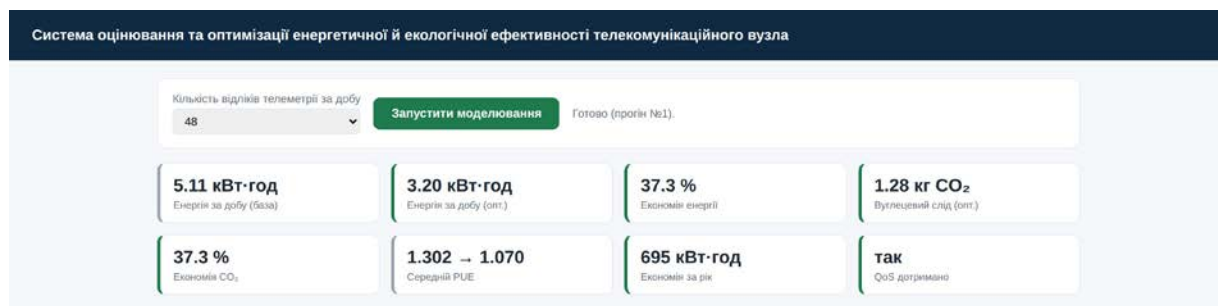


Рисунок Е.2 – Інтегральні показники енергоефективності у вікні програмного засобу

Джерело: знімок екрана застосунку, виконаного автором

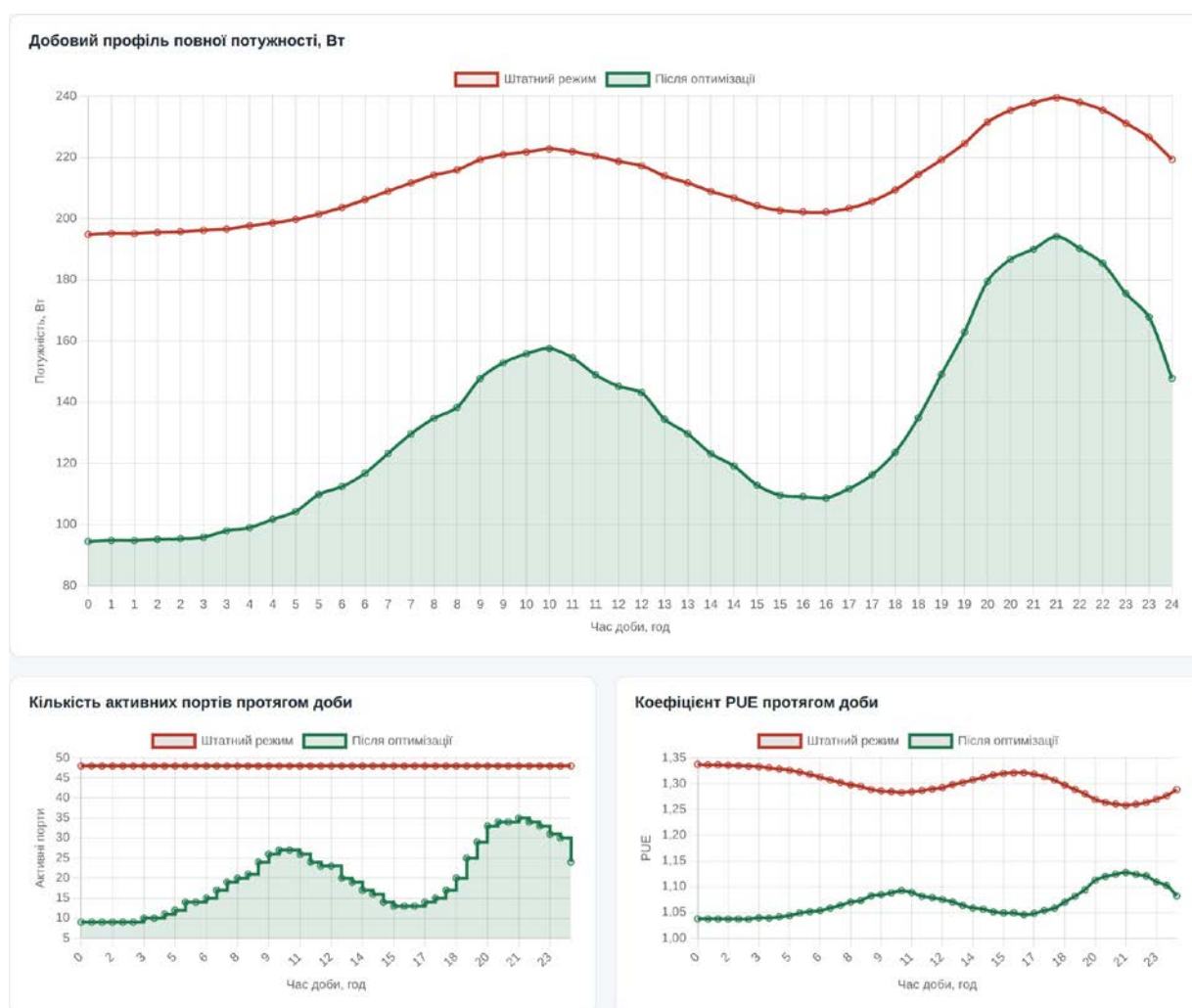


Рисунок Е.3 – Графіки добового профілю потужності, кількості активних портів та коефіцієнта PUE

Джерело: знімок екрана застосунку, виконаного автором

Джерело: знімок екрана системи керування базами даних PostgreSQL, виконаного автором