ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система віддаленого керування живленням комп'ютера засобами технології Wake-on-LAN.»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 Комп'ютерна інженерія
освітньо-професійної програми «Комп'ютерна інженерія»

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Артем ТИМЧЕНКО
(підпис)

Виконав: здобувач вищої освіти групи КІД-41
Артем ТИМЧЕНКО

Керівник: Борис КОНДРАТЮК

Рецензент: _____
Ім'я, ПРІЗВИЩЕ

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерна інженерія

Ступінь вищої освіти Бакалавра

Спеціальність 123 Комп'ютерна інженерія

Освітньо-професійна програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
_____Наталія ЛАЩЕВСЬКА
«_____» _____2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Тимченко Артем Віталійович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Система віддаленого керування живленням комп'ютера засобами технології Wake-on-LAN
керівник кваліфікаційної роботи Кондратюк Борис Олександрович викладач,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «20» лютого 2026р. № 51
2. Строк подання кваліфікаційної роботи « » _____ 2026 р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література з мережевих технологій та віддаленого адміністрування комп'ютерів; документація щодо технології Wake-on-LAN; матеріали з побудови клієнт-серверних систем;
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   1. Аналіз сучасного стану технологій віддаленого керування живленням комп'ютерів у мережевих середовищах.
   2. Огляд принципів роботи Wake-on-LAN та особливостей передачі magic packet у локальних і глобальних мережах.
   3. Реалізація клієнт-серверної моделі системи віддаленого ввімкнення комп'ютерів.
   4. Тестування та оцінювання ефективності роботи розробленої системи в різних мережевих умовах.
   5. Узагальнення отриманих результатів і формування рекомендацій щодо практичного впровадження системи.
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання « » _____ 2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	1.03-5.03.2026	Виконано
2	Аналіз науково-технічної літератури з питань віддаленого керування комп'ютерами, мережевих технологій та принципів роботи Wake-on-LAN	7.03-20.03.2026	Виконано
3	Вивчення архітектури систем віддаленого керування та методів передачі magic packet у мережі	21.03-30.03.2026	Виконано
4	Аналіз сучасних рішень для дистанційного ввімкнення комп'ютерів у локальних і глобальних мережах	1.04-6.04.2026	Виконано
5	Проєктування структури системи віддаленого керування живленням: модулі, взаємозв'язки, UML-діаграми	7.04-20.04.2026	Виконано
6	Реалізація прототипу системи Wake-on-LAN та проведення тестування її працездатності	21.04-3.05.2026	Виконано
7	Розробка презентаційних матеріалів	4.05-8.05.2026	Виконано
8	Передзахист кваліфікаційної роботи	11.05-14.05.2026	Виконано
9	Здача роботи в деканат	18.05-21.05.2026	Виконано

Здобувач вищої освіти _____ Артем ТИМЧЕНКО
(підпис)

Керівник
кваліфікаційної роботи _____ Борис КОНДРАТЮК
(підпис)

РЕФЕРАТ

Текстова частина бакалаврської роботи: 59 сторінки, 17 рисунків, 14 таблиць, 31 джерело.

Об'єкт дослідження – процеси віддаленого керування живленням комп'ютерів у мережевих середовищах.

Предмет дослідження – методи та засоби реалізації технології Wake-on-LAN для віддаленого ввімкнення комп'ютерів через мережу.

Мета роботи – розробка програмної системи віддаленого керування живленням комп'ютера з використанням технології Wake-on-LAN, що забезпечує можливість дистанційного ввімкнення пристроїв через локальну або глобальну мережу.

Методи дослідження:

1. Аналіз принципів роботи технології Wake-on-LAN.
2. Дослідження мережевих протоколів передачі даних.
3. Проєктування архітектури системи віддаленого керування.
4. Реалізація сервера для відправки magic packet.
5. Розробка клієнтського інтерфейсу користувача.
6. Тестування працездатності системи в різних мережевих умовах.

Галузь використання – корпоративні комп'ютерні мережі, системне адміністрування, інформаційна інфраструктура підприємств, віддалене керування комп'ютерами та автоматизація процесів ввімкнення і керування живленням пристроїв.

WAKE-ON-LAN, MAGIC PACKET, ВІДДАЛЕНЕ КЕРУВАННЯ, МЕРЕЖА, NODE.JS, PYTHON, UDP, КЛІЄНТ-СЕРВЕР, АВТОМАТИЗАЦІЯ

ЗМІСТ

ВСТУП 8
1 ТЕОРЕТИЧНІ ОСНОВИ ВІДДАЛЕНОГО КЕРУВАННЯ ЖИВЛЕННЯМ ПК 10
1.1 Принципи роботи технології Wake-on-LAN 10
1.2 Мережеві протоколи та особливості передачі magic packet 15
1.3 Аналіз існуючих рішень для віддаленого керування ПК 19
Висновок до розділу 1 27
2 ПРОЄКТУВАННЯ СИСТЕМИ ВІДДАЛЕНОГО КЕРУВАННЯ 28
2.1 Формування функціональних та нефункціональних вимог 28
2.2 Розробка архітектури системи (клієнт – сервер – мережа)у 30
2.3 Проєктування бази даних та логіки роботи системи 33
Висновок до розділу 2 37
3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ WAKE-ON-LAN 38
3.1 Реалізація сервера для відправки magic packet 38
3.2 Розробка клієнтського інтерфейсу користувача 45
3.3 Тестування системи та перевірка працездатності 53
Висновок до розділу 3 58
ВИСНОВКИ 59
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ ..... **Ошибка! Закладка не определена.**
Додаток А 60
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ 60

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій та широкого використання комп'ютерних мереж особливого значення набуває ефективне управління обчислювальними ресурсами. У корпоративних середовищах, освітніх установах і центрах обробки даних виникає необхідність забезпечення віддаленого доступу до комп'ютерів, їх оперативного адміністрування та оптимізації енергоспоживання.

Одним із ефективних підходів до вирішення цих завдань є використання технології Wake-on-LAN, яка дозволяє дистанційно вмикати комп'ютери за допомогою спеціального мережевого пакета. Це забезпечує можливість централізованого керування пристроями, зменшення витрат на електроенергію, а також підвищення зручності обслуговування інформаційної інфраструктури.

Незважаючи на наявність окремих програмних рішень, існує потреба у створенні гнучких, масштабованих та зручних у використанні систем, які враховують особливості сучасних мережевих середовищ, включаючи роботу як у локальних, так і в глобальних мережах. Особливо актуальним є питання інтеграції таких систем у клієнт-серверні архітектури та забезпечення їх стабільної роботи в різних умовах.

Розробка системи віддаленого керування живленням комп'ютера засобами технології Wake-on-LAN є актуальним завданням, що має як теоретичне, так і практичне значення у сфері інформаційних технологій.*Об'єкт дослідження* – процеси віддаленого керування живленням комп'ютерів у мережевих середовищах.

Предмет дослідження – методи та засоби реалізації технології Wake-on-LAN для віддаленого ввімкнення комп'ютерів через мережу.

Мета роботи – розробка програмної системи віддаленого керування живленням комп'ютера з використанням технології Wake-on-LAN, що забезпечує можливість дистанційного ввімкнення пристроїв через локальну або глобальну мережу.

Методологічною основою дослідження є вивчення і систематизація сучасних

підходів до віддаленого керування комп’ютерами, а також аналіз мережевих технологій, що забезпечують дистанційне ввімкнення пристроїв. Опрацьовано джерела, присвячені реалізації технології Wake-on-LAN, принципам передачі мережевих пакетів, особливостям роботи протоколу UDP та функціонуванню клієнт-серверних систем у комп’ютерних мережах.

У дослідженні застосовано методи системного аналізу, структурного моделювання інформаційних процесів у мережі, а також експериментального тестування. Проведено аналіз існуючих рішень для віддаленого керування живленням комп’ютерів з акцентом на ефективність передачі magic packet, швидкість реакції системи та надійність роботи в різних мережевих умовах. Особливу увагу приділено питанням забезпечення стабільної роботи системи як у локальних, так і в глобальних мережах.

Практичний етап роботи передбачає створення прототипу системи віддаленого керування живленням комп’ютера на основі технології Wake-on-LAN із використанням клієнт-серверної архітектури. Реалізовано сервер для відправки magic packet, клієнтський інтерфейс для керування пристроями, а також механізми взаємодії між компонентами системи. Система протестована на працездатність у різних мережевих умовах, визначено рівень її ефективності та надійності. Отримані результати порівнюються з існуючими рішеннями, що використовуються у сфері віддаленого адміністрування комп’ютерів.

1 ТЕОРЕТИЧНІ ОСНОВИ ВІДДАЛЕНОГО КЕРУВАННЯ ЖИВЛЕННЯМ ПК

1.1 Принципи роботи технології Wake-on-LAN

У комп’ютерних мережах важливе місце займають технології віддаленого керування, які дозволяють адміністраторам ефективно взаємодіяти з обчислювальними ресурсами незалежно від їх фізичного розташування. Однією з таких технологій є Wake-on-LAN (WoL), яка забезпечує можливість дистанційного ввімкнення комп’ютера через мережу без необхідності фізичного доступу до пристрою.

Wake-on-LAN — це мережева технологія, що дозволяє перевести комп’ютер зі стану вимкнення або енергозбереження в активний режим роботи шляхом надсилання спеціального мережевого сигналу. Основою функціонування цієї технології є передача так званого «magic packet» — спеціально сформованого пакета даних, який містить MAC-адресу цільового пристрою та розпізнається мережевим адаптером комп’ютера навіть у стані його вимкнення [1].

Принцип роботи Wake-on-LAN базується на тому, що мережева карта комп’ютера залишається частково активною навіть після вимкнення системи, отримуючи живлення від блоку живлення. У такому режимі вона здатна приймати мережеві пакети та аналізувати їх вміст. У разі отримання magic packet, який містить 16-кратне повторення MAC-адреси пристрою, мережева карта генерує сигнал на материнську плату, що призводить до запуску комп’ютера.

Для її коректної роботи необхідна підтримка Wake-on-LAN як на апаратному, так і на програмному рівнях. Зокрема, мережева карта та материнська плата повинні підтримувати відповідний режим, а в налаштуваннях BIOS/UEFI має бути активована функція WoL. Крім того, операційна система також повинна дозволяти обробку відповідних мережевих сигналів.

Рис. 1.1. демонструє послідовність взаємодії між користувачем, клієнтським застосунком, серверною частиною та цільовим пристроєм у процесі передачі

спеціального мережевого пакета.

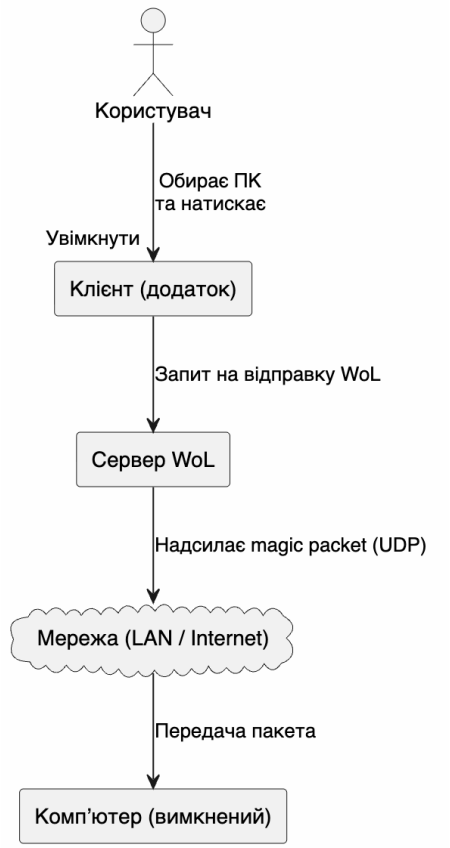


Рисунок 1.1 – Схема роботи технології Wake-on-LAN

Magic packet, виступає сигналом для віддаленого ввімкнення комп'ютера. Він спеціально сформований мережевий пакет, який передається по мережі та містить унікальний ідентифікатор цільового пристрою, а саме його MAC-адресу [2].

Структура є досить простою, але водночас ефективною. Він складається з початкової синхронізаційної послідовності байтів, що містить шість байтів зі значенням FF (у шістнадцятковому форматі), після чого слідує 16-кратне повторення MAC-адреси комп'ютера, який необхідно увімкнути. Така структура дозволяє мережевому адаптеру швидко ідентифікувати пакет серед інших мережевих даних навіть у режимі низького енергоспоживання.

Передача magic packet зазвичай здійснюється за допомогою протоколу UDP, використовуючи широкомовну (broadcast) адресу мережі, що забезпечує доставку пакета всім пристроям у підмережі. Незважаючи на те, що комп'ютер перебуває у

вимкненому стані, його мережева карта залишається активною та постійно прослуховує мережу на наявність відповідного сигналу. У разі отримання пакета, який відповідає MAC-адресі пристрою, мережевий адаптер генерує сигнал для материнської плати, ініціюючи процес увімкнення комп’ютера.

Слід зазначити, що magic packet не містить жодних механізмів автентифікації або шифрування, що може створювати потенційні ризики безпеки [3]. Саме тому у практичних реалізаціях часто використовуються додаткові заходи захисту, такі як обмеження доступу до мережі, використання VPN або фільтрація пакетів.

Для детального розуміння принципу формування сигналу, що використовується у технології Wake-on-LAN, доцільно розглянути структуру magic packet, який передається мережею для ініціювання ввімкнення комп’ютера (рис. 1.2).

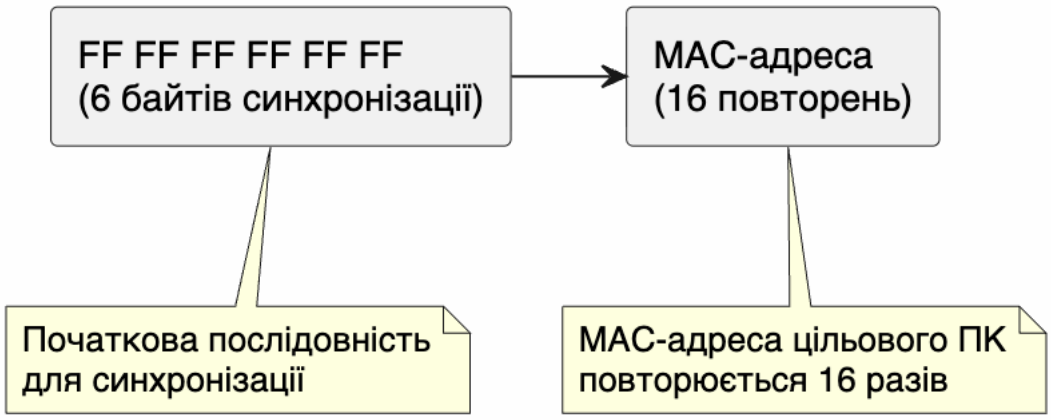


Рис. 1.2 – Структура magic packet у технології Wake-on-LAN

Magic packet має чітко визначену структуру, що складається з двох основних частин: синхронізаційної послідовності та повторюваної MAC-адреси цільового пристрою. Початкові шість байтів зі значенням FF дозволяють мережевому адаптеру визначити початок пакета, після чого здійснюється перевірка відповідності MAC-адреси. Багаторазове повторення адреси підвищує надійність розпізнавання пакета навіть за наявності мережевих перешкод або втрати частини даних.

Як видно з таблиці 1.1, структура magic packet є досить простою та не містить

складних заголовків або службової інформації. Основний акцент зроблено на багаторазовому повторенні MAC-адреси, що дозволяє мережевому адаптеру коректно ідентифікувати пакет навіть у випадку часткових втрат даних або наявності перешкод у мережі. Невеликий розмір пакета сприяє швидкій передачі та мінімальному навантаженню на мережу.

Таблиця 1.1 – Структура magic packet у технології Wake-on-LAN

№	Поле пакета	Розмір	Опис
1	Синхронізаційна послідовність	6 байтів (FF FF FF FF FF FF)	Використовується для визначення початку пакета мережевим адаптером
2	MAC-адреса пристрою	6 байтів	Унікальний ідентифікатор мережевого інтерфейсу комп’ютера
3	Повторення MAC-адреси	16 × 6 байтів (96 байтів)	MAC-адреса повторюється 16 разів для надійного розпізнавання пакета
4	Загальний розмір пакета	102 байти	Повна довжина magic packet (6 + 96 байтів)

Для комплексної оцінки технології Wake-on-LAN доцільно розглянути її основні переваги та недоліки, що дозволяє визначити ефективність використання цієї технології в сучасних комп’ютерних мережах [4]. Як видно з табл. 1.2, технологія Wake-on-LAN має низку суттєвих переваг, серед яких варто відзначити простоту реалізації, низьке навантаження на мережу та можливість ефективного віддаленого керування комп’ютерами. Це робить її особливо корисною для системних адміністраторів у корпоративних середовищах. Водночас існують і певні обмеження, зокрема відсутність вбудованих механізмів захисту та складність використання у глобальних мережах через мережеві обмеження. Тому при практичному впровадженні технології доцільно передбачати додаткові заходи безпеки, такі як використання VPN або обмеження доступу до мережі.

Таблиця 1.2 – Переваги та недоліки технології Wake-on-LAN

№	Переваги	Недоліки
1	Можливість віддаленого ввімкнення комп’ютера	Відсутність вбудованих механізмів безпеки
2	Зниження енергоспоживання	Обмежена робота через Інтернет (NAT, firewall)
3	Простота реалізації	Потреба налаштування BIOS/UEFI
4	Мінімальне навантаження на мережу	Залежність від апаратної підтримки
5	Зручність для адміністрування	Можливість несанкціонованого доступу
6	Швидка реакція системи	Працює переважно в локальній мережі

Однією особливостей технології Wake-on-LAN є здатність комп’ютера реагувати на спеціальний мережевий сигнал навіть у вимкненому стані. Це досягається завдяки взаємодії мережевого адаптера та системи базового вводу-виводу (BIOS/UEFI), які забезпечують апаратну підтримку функції віддаленого ввімкнення.

Мережева карта комп’ютера, підключена до джерела живлення, залишається частково активною навіть після вимкнення основних компонентів системи. У такому режимі вона продовжує прослуховувати мережу, аналізуючи вхідні пакети даних. При отриманні magic packet мережевий адаптер виконує перевірку його структури, зокрема відповідності MAC-адреси. У разі успішного розпізнавання пакета формується спеціальний сигнал, який передається на материнську плату.

BIOS або UEFI, у свою чергу, відповідає за обробку цього сигналу та ініціалізацію процесу запуску комп’ютера. За умови, що функція Wake-on-LAN активована в налаштуваннях системи, BIOS/UEFI дозволяє приймати сигнал від мережевого адаптера та запускає процедуру подачі живлення на основні компоненти комп’ютера, включаючи процесор, оперативну пам’ять та інші пристрої [5].

Для коректної роботи Wake-on-LAN необхідне попереднє налаштування як апаратної, так і програмної складових. Зокрема, у BIOS/UEFI має бути увімкнено відповідну опцію (наприклад, «Wake on LAN», «Power On by PCI-E» або

аналогічну), а також у параметрах операційної системи повинні бути активовані функції дозволу пробудження мережевим адаптером.

Процес віддаленого ввімкнення комп'ютера складається з кількох основних етапів: формування запиту користувачем, передача magic packet через мережу та його обробка мережевим адаптером цільового пристрою. Після успішного розпізнавання пакета відбувається генерація сигналу пробудження, що ініціює запуск комп'ютера. Такий підхід забезпечує простоту реалізації та ефективність застосування технології у різних мережевих середовищах.



Рисунок 1.3 – Принцип роботи технології Wake-on-LAN

1.2 Мережеві протоколи та особливості передачі magic packet

Передача magic packet у технології Wake-on-LAN здійснюється з використанням мережевих протоколів, які забезпечують доставку даних від джерела до цільового пристрою. Одним із протоколів, що застосовується для

реалізації цієї технології, є протокол UDP (User Datagram Protocol), який належить до транспортного рівня моделі TCP/IP.

UDP є простим безз'єднувальним протоколом, що не потребує встановлення попереднього з'єднання між відправником і отримувачем. На відміну від TCP, який гарантує доставку даних, UDP працює за принципом «відправив і забув», тобто не забезпечує підтвердження отримання пакета, повторної передачі у разі втрати або контролю порядку доставки. Саме ця особливість робить UDP швидким і ефективним для передачі невеликих пакетів даних, таких як magic packet, де важлива швидкість, а не гарантована доставка.

У технології Wake-on-LAN використання UDP є доцільним, оскільки magic packet має невеликий розмір і не потребує складної обробки або встановлення з'єднання. Передача пакета зазвичай здійснюється на стандартні порти, такі як 7 або 9, із використанням широкомовної адреси (broadcast), що дозволяє доставити пакет усім пристроям у межах локальної мережі. Мережева карта цільового комп'ютера аналізує отримані пакети та реагує лише на той, що містить відповідну MAC-адресу.

Окрім UDP, у процесі передачі magic packet задіюються й інші мережеві протоколи нижчих рівнів, зокрема протокол IP, який відповідає за адресацію та маршрутизацію пакетів у мережі, а також протоколи канального рівня, такі як Ethernet, що забезпечують фізичну доставку даних у межах локальної мережі. У випадку використання широкомовної передачі пакетів застосовується спеціальна broadcast-адреса, яка дозволяє одночасно надіслати пакет усім вузлам мережі.

Передача magic packet у глобальних мережах, зокрема через Інтернет, ускладнюється через використання технологій NAT (Network Address Translation) та мережевих екранів (firewall), які можуть блокувати широкомовні пакети або змінювати їх маршрутизацію. У таких випадках для забезпечення працездатності Wake-on-LAN застосовуються додаткові механізми, наприклад перенаправлення портів (port forwarding), використання статичних IP-адрес або організація захищених каналів зв'язку через VPN [6].

Використання протоколу UDP у поєднанні з іншими мережевими технологіями забезпечує ефективну передачу magic packet, що є основою функціонування систем віддаленого керування живленням комп'ютерів на базі Wake-on-LAN.

Як видно з табл. 1.3, протокол UDP є більш придатним для реалізації технології Wake-on-LAN, оскільки забезпечує швидку передачу даних без необхідності встановлення з'єднання та мінімізує накладні витрати. У той же час використання TCP у даному випадку є недоцільним через складність встановлення з'єднання та наявність механізмів контролю, які не є необхідними для передачі невеликого за обсягом magic packet. Таким чином, саме UDP виступає оптимальним вибором для реалізації систем віддаленого ввімкнення комп'ютерів.

Таблиця 1.3 – Порівняння протоколів UDP та TCP у контексті Wake-on-LAN

№	Критерій	UDP	TCP
1	Тип з'єднання	Безз'єднувальний	З'єднувальний
2	Швидкість передачі	Висока	Нижча через встановлення з'єднання
3	Гарантія доставки	Відсутня	Гарантована
4	Контроль помилок	Мінімальний	Повний контроль
5	Накладні витрати	Низькі	Високі
6	Складність реалізації	Проста	Складніша
7	Підходить для WoL	Так (основний протокол)	Ні (надлишковий для цієї задачі)

Передача magic packet у технології Wake-on-LAN тісно пов'язана з особливостями адресації в мережі, зокрема використанням широкомовної передачі (broadcast), підмереж (subnet) та стандартних портів. У більшості випадків пакет надсилається на широкомовну адресу мережі, наприклад 192.168.1.255, що дозволяє доставити його одночасно всім пристроям у межах конкретної підмережі. Такий підхід є ефективним, оскільки комп'ютер, який перебуває у вимкненому стані, не має активної IP-адреси, але його мережева карта здатна обробляти пакети

на рівні MAC-адрес. Передача зазвичай здійснюється через UDP із використанням стандартних портів, таких як 7 або 9, які традиційно застосовуються для службових або тестових мережевих функцій і добре підходять для передачі коротких сигналів, зокрема magic packet.

У межах локальної мережі процес передачі пакета відбувається відносно просто та ефективно, оскільки всі пристрої знаходяться в одній підмережі та можуть безпосередньо обмінюватися даними. Після формування magic packet клієнт або сервер надсилає його на broadcast-адресу, внаслідок чого всі мережеві вузли отримують цей пакет. Мережеві адаптери кожного пристрою перевіряють вміст пакета, однак реагує лише той комп'ютер, MAC-адреса якого відповідає зазначеній у пакеті. Завдяки цьому забезпечується вибірковість спрацювання без необхідності індивідуального адресування кожного пристрою, що значно спрощує реалізацію технології у локальних мережах.

Разом із тим, передача magic packet у глобальних мережах, зокрема через Інтернет, супроводжується рядом технічних труднощів, пов'язаних із використанням технології NAT (Network Address Translation) та засобів мережевого захисту, таких як firewall. NAT виконує трансляцію внутрішніх IP-адрес у зовнішню, що ускладнює маршрутизацію пакетів до конкретного пристрою в локальній мережі [7]. Крім того, широкомовні пакети, як правило, не маршрутизуються через Інтернет, що робить стандартний механізм Wake-on-LAN непридатним без додаткових налаштувань. Firewall, у свою чергу, може блокувати вхідні UDP-пакети, особливо якщо вони надходять на нестандартні або незахищені порти, що також ускладнює реалізацію віддаленого ввімкнення комп'ютера.

Для подолання зазначених обмежень застосовуються додаткові підходи, зокрема налаштування перенаправлення портів (port forwarding) на маршрутизаторі, використання статичних ARP-записів або організація захищених з'єднань через VPN, які дозволяють імітувати перебування пристрою в локальній мережі. Таким чином, хоча технологія Wake-on-LAN найбільш ефективно працює у межах локальної мережі, її використання в глобальному середовищі також можливе за умови належної конфігурації мережевої інфраструктури.

Використання технології Wake-on-LAN у локальній мережі є значно простішим та ефективнішим, оскільки забезпечується пряма передача широкомовних пакетів без необхідності додаткових налаштувань. У той же час реалізація цієї технології через Інтернет супроводжується рядом труднощів, зокрема відсутністю підтримки broadcast-пакетів, впливом NAT та обмеженнями з боку мережевих екранів. Це вимагає застосування додаткових механізмів, таких як перенаправлення портів або використання VPN, що ускладнює впровадження системи, але водночас розширює можливості її використання (табл. 1.4).

Таблиця 1.4 – Порівняння передачі Wake-on-LAN у локальній мережі та через Інтернет

№	Критерій	Локальна мережа (LAN)	Глобальна мережа (Internet)
1	Тип адресації	Broadcast (широкомовна)	Unicast або через relay
2	Складність реалізації	Низька	Висока
3	Необхідність налаштувань	Мінімальна	Потрібне налаштування NAT, port forwarding
4	Швидкість передачі	Висока	Залежить від мережі
5	Надійність	Висока	Нижча через мережеві обмеження
6	Підтримка broadcast	Повністю підтримується	Не підтримується
7	Вплив firewall	Мінімальний	Значний (може блокувати пакети)
8	Додаткові технології	Не потрібні	VPN, relay-сервери

1.3 Аналіз існуючих рішень для віддаленого керування ПК

У процесі дослідження технології Wake-on-LAN важливим етапом є аналіз уже існуючих програмних рішень, які застосовуються для віддаленого ввімкнення комп’ютерів. Такий аналіз дозволяє не лише оцінити сучасний стан практичної реалізації цієї технології, але й виявити сильні та слабкі сторони наявних інструментів, що надалі слугує основою для обґрунтування доцільності розробки власної системи. Серед найбільш відомих рішень, які використовуються для

передачі magic packet і дистанційного запуску комп’ютерів, доцільно виділити WakeMeOnLan, Depicus WOL та TeamViewer із підтримкою Wake-on-LAN. Кожен із цих інструментів реалізує одну й ту саму базову ідею, але відрізняється підходом до налаштування, функціональними можливостями, зручністю використання та сферою практичного застосування.

Одним із інструментів є WakeMeOnLan, розроблений компанією NirSoft. Це утиліта, орієнтована насамперед на середовище Windows, яка дозволяє виявляти комп’ютери в локальній мережі, зберігати інформацію про їх IP- та MAC-адреси, а також надсилати Wake-on-LAN-пакети для віддаленого ввімкнення вибраних машин. Серед важливих можливостей цієї програми слід відзначити підтримку командного рядка: утиліта може працювати не лише через графічний інтерфейс, а й за допомогою параметра /wakeup, що дає змогу інтегрувати її у сценарії автоматизації та адміністративні скрипти. На офіційному сайті NirSoft також зазначається, що WakeMeOnLan дозволяє виконувати надсилання пакета за ім’ям комп’ютера, IP-адресою або MAC-адресою, якщо відповідна інформація збережена у конфігурації програми [8].

WakeMeOnLan, програма не перевантажена складними функціями, має зрозумілий інтерфейс, не потребує значних системних ресурсів і добре підходить для базових задач системного адміністрування в межах локальної мережі. Крім того, підтримка командного рядка робить це рішення зручним для напівавтоматичного або сценарного запуску, наприклад у корпоративному середовищі, де необхідно за розкладом вмикати робочі станції перед початком робочого дня. Ще однією сильною стороною є орієнтація саме на Wake-on-LAN без додаткового функціонального навантаження, що робить утиліту легкою у впровадженні та зрозумілою навіть для користувача з базовими навичками адміністрування.

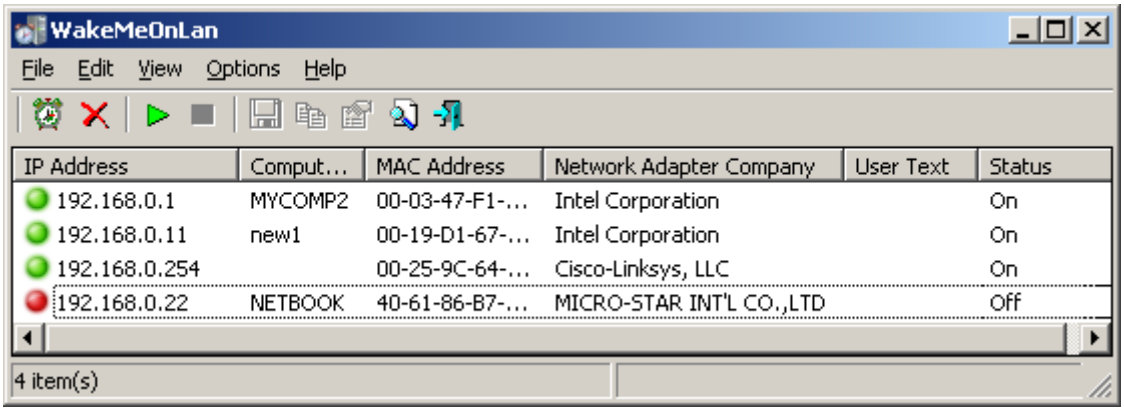


Рисунок 1.4 – Інтерфейс WakeMeOnLan

Разом із тим WakeMeOnLan має і низку обмежень. Передусім це рішення найбільш ефективне саме в локальній мережі, де можна швидко виявити пристрої та надіслати їм широкомовний пакет. Утиліта не пропонує повноцінної серверної архітектури, централізованого веб-інтерфейсу, розвинених механізмів автентифікації користувачів або журналювання подій на рівні багатокористувацької системи. Фактично вона є локальною адміністративною утилітою, а не комплексною інформаційною системою для віддаленого керування живленням ПК. Саме тому її використання є доречним у невеликих або середніх локальних мережах, але менш зручним у тих випадках, коли потрібно забезпечити централізоване керування великою кількістю пристроїв, доступ через браузер або детальний облік дій користувачів. Така оцінка є узагальненням функціональних можливостей, наведених на офіційному сайті, і показує, що інструмент орієнтований радше на точкове адміністрування, ніж на побудову повноцінної системи.

Іншим відомим рішенням є Depicus WOL, яке представлено у вигляді декількох варіантів реалізації: графічної програми для Windows, утиліти командного рядка, вебсервісів, а також мобільних застосунків і допоміжних засобів для роботи з Wake-on-LAN [9]. На офіційному сайті Depicus зазначено наявність GUI-версії для Windows, безкоштовної командної утиліти wolcmd, сервісу для відправлення magic packet через Інтернет, а також допоміжних інструментів, пов’язаних із forwarding та моніторингом WoL. Окремо вказано, що GUI-версія

дозволяє використовувати не лише IP-адресу, а й FQDN, а командна версія підтримує запуск із параметрами, зокрема MAC-адресою, IP, subnet і портом.

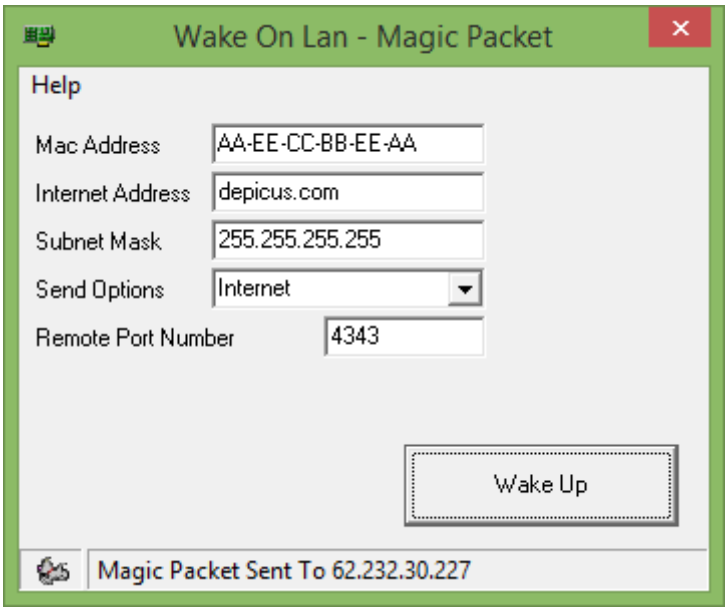


Рисунок 1.5 – Інтерфейс Depicus WOL

Сильна сторона Depicus WOL полягає в його гнучкості. Це рішення пропонує декілька форматів використання: від простого GUI-інструмента до командного рядка й сервісів для роботи через Інтернет. Така багатоваріантність робить його придатним для різних сценаріїв — від одиничного ввімкнення домашнього ПК до побудови більш складного адміністративного процесу із використанням зовнішнього сервісу або скриптів. Крім того, на сайті Depicus прямо наголошується на підтримці Wake-on-LAN як у локальній мережі, так і через Інтернет, що розширює сферу практичного застосування цього рішення.

Однак і це рішення не позбавлене недоліків. Попри наявність кількох способів реалізації, Depicus WOL залишається набором окремих утиліт, а не цілісною клієнт-серверною системою з єдиним інформаційним простором. Для практичного використання через Інтернет користувачу все одно необхідно враховувати особливості маршрутизації, налаштування портів, broadcast-адрес, а також мережеві обмеження з боку NAT і firewall. Іншими словами, інструмент полегшує відправлення magic packet, але не усуває фундаментальних мережевих проблем, які виникають при віддаленому доступі поза межами локальної мережі.

Також слід відзначити, що інтерфейси таких інструментів орієнтовані переважно на технічно підготовленого користувача і не завжди забезпечують зручне централізоване керування списком пристроїв, ролями користувачів, журналами подій та інтеграцією з корпоративними сервісами. Це дозволяє зробити висновок, що Depicus WOL є зручним технічним інструментом, але не повністю відповідає вимогам до масштабованої системи керування живленням комп'ютерів.

Більш комплексним за функціональністю рішенням є TeamViewer із вбудованою підтримкою Wake-on-LAN. На відміну від попередніх інструментів, TeamViewer не обмежується лише надсиланням magic packet, а пропонує повноцінну екосистему для віддаленого доступу, адміністрування та підтримки пристроїв [10]. В офіційній документації TeamViewer вказано, що Wake-on-LAN може бути налаштований двома основними способами: через інший пристрій TeamViewer у тій самій локальній мережі або через публічну адресу й UDP-порт. Компанія також підкреслює, що ця можливість дозволяє вмикати вимкнені або сплячі пристрої з будь-якого місця за умови належної конфігурації цільового комп'ютера та мережі.

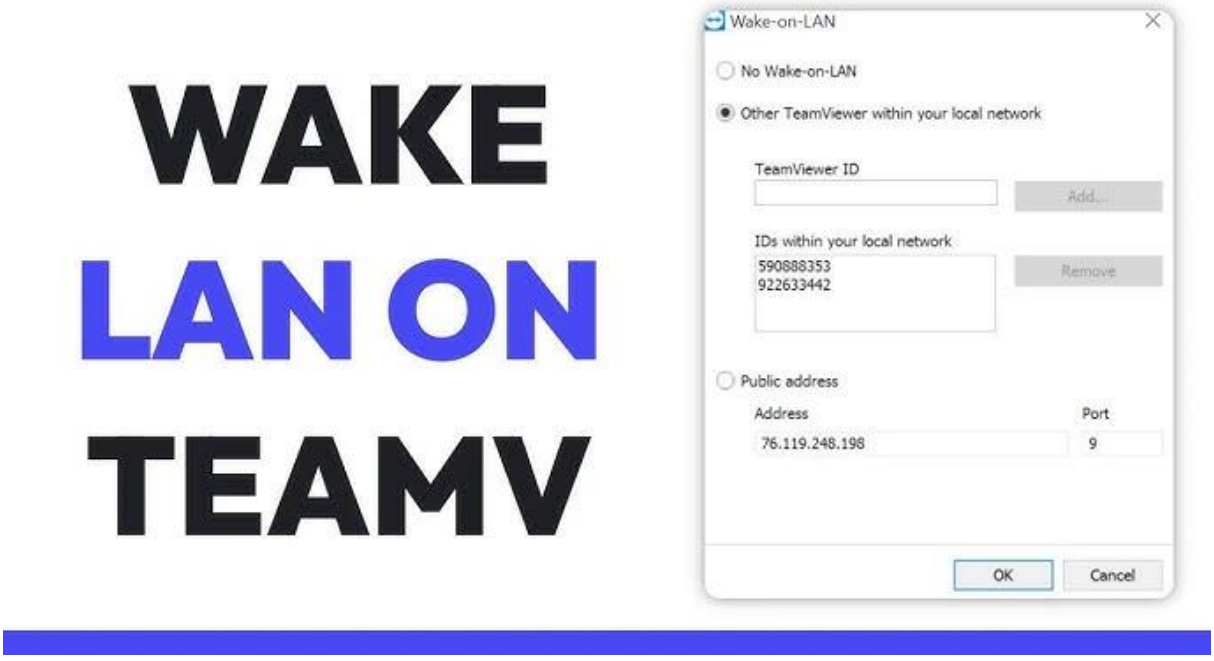


Рисунок 1.5 – Інтерфейс TeamViewer із підтримкою Wake-on-LAN

Переваги TeamViewer полягають у тому, що користувач отримує не просто

інструмент для надсилання magic packet, а комплексне середовище для подальшого віддаленого підключення до комп’ютера після його ввімкнення. Це суттєво підвищує практичну цінність рішення, особливо для служб технічної підтримки, системних адміністраторів і підприємств, де важливо не лише запустити пристрій, а й одразу перейти до його налаштування, обслуговування або діагностики. Крім того, TeamViewer підтримує різні варіанти ініціації пробудження, включаючи використання іншого активного комп’ютера в тій самій мережі або звернення через публічну адресу, що робить систему більш адаптивною до різних умов експлуатації.

Водночас саме комплексність TeamViewer є і його певним недоліком у контексті досліджуваної теми. Для простого надсилання Wake-on-LAN-пакета це рішення може виявитися надлишковим, оскільки потребує встановлення та конфігурації додаткового програмного забезпечення, прив’язки пристроїв, а інколи й орієнтації на екосистему самого сервісу. Також Wake-on-LAN у TeamViewer не є абсолютно автономною мережею функцією, а виступає складовою більшого продукту віддаленого доступу. Через це користувач опиняється залежним від конкретної платформи, її налаштувань і сценаріїв роботи. Для дипломного проєкту, метою якого є розробка власної системи віддаленого керування живленням комп’ютера, така залежність від сторонньої комерційної екосистеми може розглядатися як обмеження, особливо якщо потрібно створити окреме, легке, контрольоване й адаптоване під конкретні потреби рішення.

Порівнюючи зазначені інструменти, можна зробити висновок, що кожен із них вирішує задачу віддаленого ввімкнення комп’ютера лише частково або в межах певного сценарію. WakeMeOnLan є зручним і легким інструментом для локального адміністрування, але не забезпечує розвиненої серверної логіки та засобів централізованого керування. Depicus WOL є більш гнучким у плані способів надсилання magic packet і підтримки різних платформ, однак теж не формує цілісної системи з єдиним інтерфейсом, базою даних і механізмами розмежування доступу [11]. TeamViewer, навпаки, пропонує потужне інтегроване середовище, але є значно складнішим, більш залежним від зовнішньої платформи та орієнтованим

не лише на Wake-on-LAN, а на віддалений доступ загалом. Така багатофункціональність є перевагою з позиції готового комерційного продукту, але не завжди відповідає завданню побудови спеціалізованої системи саме для керування живленням комп’ютерів.

В табл. 1.5, кожне з розглянутих рішень має свої особливості та сферу ефективного застосування. WakeMeOnLan є зручним інструментом для роботи в локальній мережі, однак має обмежений функціонал і не підтримує централізоване керування. Depicus WOL забезпечує більшу гнучкість і дозволяє працювати як у локальних, так і в глобальних мережах, проте також не реалізує повноцінної системної архітектури. TeamViewer, у свою чергу, пропонує найширші можливості, включаючи централізоване керування та віддалений доступ, але є складнішим у налаштуванні та залежним від зовнішньої платформи.

Таблиця 1.5 – Порівняння інструментів Wake-on-LAN

№	Критерій	WakeMeOnLan	Depicus WOL	TeamViewer (WoL)
1	Тип рішення	Локальна утиліта	Набір інструментів	Комплексна система
2	Інтерфейс	Графічний + CLI	GUI + CLI + Web	Графічний (desktop/web)
3	Підтримка LAN	Так	Так	Так
4	Підтримка через Інтернет	Обмежена	Частково	Повна
5	Складність налаштування	Низька	Середня	Висока
6	Централізоване керування	Ні	Ні	Так
7	Наявність БД пристроїв	Локально	Обмежено	Так
8	Безпека	Низька	Середня	Висока
9	Залежність від платформи	Низька	Низька	Висока (екосистема TeamViewer)
10	Основне призначення	Простий WoL у LAN	Гнучкий WoL (LAN + Internet)	Віддалений доступ + WoL

Проовження таблиці 1.5

11	Масштабованість	Низька	Середня	Висока
12	Придатність для дипломної системи	Частково	Частково	Обмежено (через складність і залежність)

Саме тому можна стверджувати, що існуючим рішенням часто не вистачає тих характеристик, які є необхідними для інформаційної системи віддаленого керування живленням комп’ютера. Зокрема, мова йде про наявність єдиного клієнт-серверного підходу, централізованого зберігання даних про пристрої, журналювання операцій, зручного веб-інтерфейсу, можливості гнучкого розширення функціональності та адаптації до конкретної організації чи мережевої інфраструктури. У частині випадків наявні інструменти є занадто простими і виконують лише одну операцію, а в інших — навпаки надмірно складними й залежними від сторонніх платформ. Отже, розробка власної системи віддаленого керування живленням комп’ютера засобами технології Wake-on-LAN є обґрунтованою, оскільки вона дає змогу поєднати простоту використання, спеціалізовану функціональність, контроль над архітектурою системи та можливість її подальшого вдосконалення відповідно до практичних потреб користувача.

Висновок до розділу 1

Розглянуто теоретичні основи технології віддаленого керування живленням комп’ютерів на базі Wake-on-LAN, а також проаналізовано принципи її функціонування та особливості застосування у сучасних мережевих середовищах. Визначено, що Wake-on-LAN є ефективним інструментом для дистанційного увімкнення комп’ютерів, який базується на передачі спеціального мережевого пакета – magic packet, що розпізнається мережевим адаптером навіть у вимкненому стані пристрою.

Розглянуто структуру magic packet, особливості його формування та передачі через мережу з використанням протоколу UDP. Встановлено, що ключову роль у процесі відіграють широкомовні (broadcast) пакети, параметри підмережі та правильне налаштування портів, що забезпечують доставку сигналу до цільового пристрою. Також було проаналізовано механізм реагування комп’ютера на отриманий пакет, зокрема взаємодію мережевого адаптера та BIOS/UEFI.

Окрему увагу приділено аналізу існуючих програмних рішень, таких як WakeMeOnLan, Depicus WOL та TeamViewer, що дозволило визначити їх основні переваги та недоліки. Встановлено, що хоча зазначені інструменти забезпечують базову функціональність, вони мають обмеження у сфері централізованого управління, масштабованості, безпеки та розширених можливостей моніторингу.

Також було досліджено особливості передачі magic packet у локальних мережах та через Інтернет, включаючи проблеми, пов’язані з NAT, firewall та маршрутизацією пакетів. Визначено, що найбільш ефективною технологія є у межах локальної мережі, тоді як для віддаленого доступу необхідні додаткові налаштування або використання проміжних серверів.

2 ПРОЄКТУВАННЯ СИСТЕМИ ВІДДАЛЕНОГО КЕРУВАННЯ

2.1 Формування функціональних та нефункціональних вимог

На етапі проєктування системи віддаленого керування живленням комп’ютерів на основі технології Wake-on-LAN важливим завданням є визначення функціональних вимог, які описують основні можливості та поведінку системи. Саме функціональні вимоги визначають, які операції повинна виконувати система та які сервіси надавати користувачу.

Розроблювана система повинна забезпечувати можливість віддаленого ввімкнення комп’ютерів шляхом використання технології Wake-on-LAN. Користувач повинен мати змогу обирати цільовий пристрій зі списку, ініціювати надсилання magic packet та отримувати інформацію про результат виконання операції. Крім того, система має реалізовувати механізми збереження інформації про пристрої, а також вести журнал подій для контролю та аналізу виконаних дій [12].

Однією з функцій системи є можливість ввімкнення комп’ютера за його MAC-адресою, що забезпечує точну ідентифікацію пристрою в мережі. Для зручності користування передбачається формування списку пристроїв, який містить необхідну інформацію про кожен комп’ютер, зокрема MAC-адресу, IP-адресу та назву. Система також повинна реалізовувати механізм відправки magic packet через мережу із використанням протоколу UDP.

Забезпечення логування дій користувача, що дозволяє відстежувати історію виконаних операцій, включаючи час надсилання пакета та результат виконання. Додатково система повинна підтримувати функцію перевірки статусу пристрою, що дає змогу визначити, чи був комп’ютер успішно ввімкнений.

Функціональні вимоги охоплюють основні аспекти роботи системи, включаючи взаємодію з користувачем, обробку мережевих запитів та контроль результатів виконання операцій. Реалізація зазначених функцій забезпечує повноцінну роботу системи віддаленого керування живленням комп’ютерів і

створює основу для подальшого розширення її можливостей (табл. 2.1).

Таблиця 2.1 – Функціональні вимоги системи

№	Функція	Опис
1	Ввімкнення ПК по MAC	Віддалене ввімкнення комп’ютера за допомогою MAC-адреси
2	Список пристроїв	Збереження та відображення переліку комп’ютерів
3	Відправка magic packet	Формування та передача WoL-пакета через мережу
4	Логування	Фіксація дій користувача та результатів виконання
5	Перевірка статусу	Визначення стану пристрою (увімкнений/вимкнений)

Окрім функціональних характеристик, важливим етапом проєктування є визначення нефункціональних вимог, які описують якісні властивості системи, її продуктивність, надійність, безпеку та зручність використання. Саме ці вимоги визначають рівень ефективності системи в реальних умовах експлуатації та впливають на її практичну цінність.

Розроблювана система повинна забезпечувати високу швидкість виконання основних операцій, зокрема мінімальний час відгуку при відправці magic packet та отриманні результату. Це є критично важливим для оперативного керування пристроями, особливо у випадках, коли система використовується в корпоративному середовищі.

Надійність системи передбачає стабільну роботу навіть за умов змінного мережевого навантаження або часткових збоїв. Система повинна коректно обробляти помилки передачі даних та забезпечувати повторне виконання операцій у разі необхідності.

Важливим аспектом є безпека, оскільки технологія Wake-on-LAN не має вбудованих механізмів захисту. У зв’язку з цим система повинна обмежувати доступ до функцій керування, забезпечувати автентифікацію користувачів та захист від несанкціонованого використання.

Масштабованість системи визначає її здатність працювати з різною кількістю пристроїв без втрати продуктивності. Система повинна підтримувати розширення списку комп’ютерів і адаптуватися до зростання мережевої інфраструктури.

Зручність використання є важливою характеристикою, яка визначає ефективність взаємодії користувача із системою. Інтерфейс повинен бути інтуїтивно зрозумілим, забезпечувати швидкий доступ до основних функцій та не вимагати спеціальних технічних знань для роботи.

Нефункціональні вимоги визначають ключові якісні характеристики системи, що забезпечують її ефективну та стабільну роботу. Дотримання цих вимог дозволяє створити надійну, безпечну та зручну у використанні систему віддаленого керування живленням комп’ютерів (табл. 2.2).

Таблиця 2.2 – Нефункціональні вимоги системи

№	Вимога	Опис
1	Швидкість	Мінімальний час відправки пакета та реакції системи
2	Надійність	Стабільна робота та обробка помилок
3	Безпека	Захист доступу та автентифікація користувачів
4	Масштабованість	Можливість роботи з великою кількістю пристроїв
5	Зручність	Простий та інтуїтивний інтерфейс користувача

2.2 Розробка архітектури системи (клієнт – сервер – мережа)у

На основі сформованих функціональних та нефункціональних вимог було виконано проєктування архітектури системи віддаленого керування живленням комп’ютерів із використанням технології Wake-on-LAN. Архітектура системи визначає основні компоненти, їх взаємодію та логіку обробки запитів користувача.

Розроблювана система реалізується за клієнт-серверною моделлю, де користувач взаємодіє з інтерфейсом застосунку, який забезпечує передачу команд на серверну частину. Сервер, у свою чергу, відповідає за обробку запитів, формування magic packet та його передачу до цільового пристрою через мережу.

Загальна структура системи передбачає наявність кількох основних компонентів: користувацького інтерфейсу, модуля керування пристроями, мережевого модуля для передачі пакетів, а також модуля збереження даних. Такий підхід забезпечує модульність системи, що спрощує її розширення та підтримку.

Модуль користувача (UI) –забезпечує взаємодію користувача із системою. Через інтерфейс користувач має можливість переглядати список пристроїв, додавати нові комп’ютери, ініціювати їх ввімкнення та переглядати журнал подій. Інтерфейс повинен бути інтуїтивно зрозумілим і забезпечувати швидкий доступ до основних функцій.

Модуль керування пристроями – відповідає за збереження та обробку інформації про комп’ютери. Він містить дані про MAC-адреси, IP-адреси, назви пристроїв та інші параметри. Також модуль забезпечує вибір цільового пристрою для надсилання команди ввімкнення.

Мережевий модуль реалізує ключову функцію системи — формування та передачу magic packet. Він використовує протокол UDP для надсилання широкомовного пакета у мережу, що дозволяє досягти цільового пристрою навіть у вимкненому стані. Саме цей модуль забезпечує безпосередню реалізацію технології Wake-on-LAN [13].

Модуль логування відповідає за фіксацію всіх дій користувача та результатів виконання операцій. У журналі зберігається інформація про час надсилання пакета, обраний пристрій та статус виконання. Це дозволяє здійснювати контроль роботи системи та аналізувати її ефективність.

Модуль перевірки статусу, забезпечує визначення стану комп’ютера після відправки magic packet. Це може реалізовуватись через ping-запити або інші мережеві механізми перевірки доступності пристрою.

У процесі роботи системи користувач через інтерфейс ініціює команду ввімкнення комп’ютера. Запит передається до модуля керування пристроями, який визначає параметри цільового пристрою. Далі мережевий модуль формує magic packet і надсилає його через мережу. Після цього модуль перевірки статусу здійснює перевірку доступності комп’ютера, а результати фіксуються у журналі подій (табл. 2.3).

Таблиця 2.3 – Модулі системи та їх функції

№	Модуль	Функції
1	Модуль користувача	Інтерфейс, взаємодія з користувачем
2	Модуль керування пристроями	Збереження та обробка інформації про ПК
3	Мережевий модуль	Формування та відправка magic packet
4	Модуль логування	Запис подій та результатів
5	Модуль перевірки статусу	Перевірка доступності пристрою

Для кращого розуміння структури системи та взаємодії її основних компонентів доцільно представити узагальнену архітектурну схему. Вона відображає процес передачі команд від користувача до цільового пристрою та демонструє роль кожного модуля в системі (рис. 2.1).

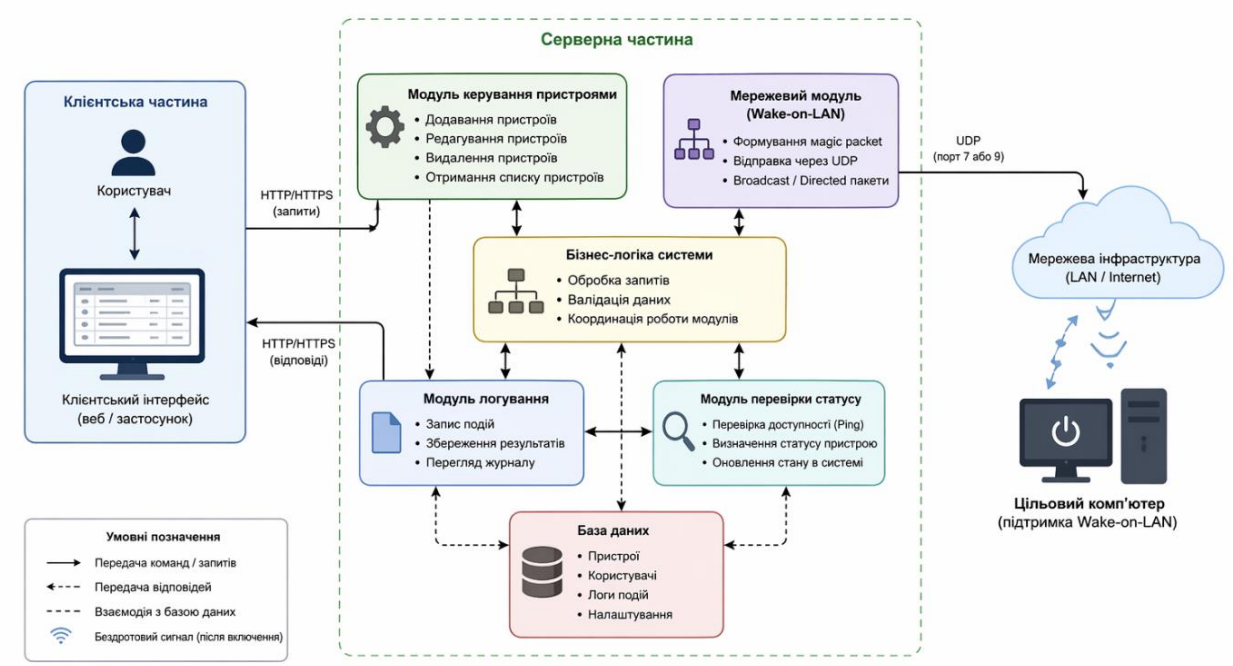


Рисунок 2.1 – Архітектурна схема системи віддаленого керування живленням комп’ютерів на основі технології Wake-on-LAN

Розроблена архітектура системи забезпечує чітке розмежування функцій між основними модулями та ефективну взаємодію між ними. Використання модульного підходу дозволяє підвищити гнучкість системи, спростити її масштабування та забезпечити зручність подальшого розвитку.

2.3 Проєктування бази даних та логіки роботи системи

У межах системи Wake-on-LAN збереження даних виконується у реляційній базі даних, яка містить інформацію про користувачів, керовані пристрої та історію надсилання magic packet. Такий підхід дозволяє забезпечити централізоване керування, аудит дій та подальше масштабування рішення.

Мінімальний склад сутностей:

1. users - облікові записи операторів системи.
2. devices - перелік комп'ютерів/вузлів, що можуть бути пробуджені.
3. wake_logs - журнал подій надсилання пакетів і результатів виконання.

Логіка роботи має такий цикл: користувач автентифікується, обирає пристрій, система перевіряє коректність MAC-адреси та мережевих параметрів, формує magic packet, надсилає його через UDP broadcast і фіксує результат у журналі [16, 17].

SQL-фрагмент структури БД:

```
CREATE TABLE users (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  username TEXT NOT NULL UNIQUE,
  password_hash TEXT NOT NULL,
  role TEXT NOT NULL DEFAULT 'operator'
);

CREATE TABLE devices (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  name TEXT NOT NULL,
  mac_address TEXT NOT NULL UNIQUE,
  ip_address TEXT,
  broadcast_ip TEXT NOT NULL,
  udp_port INTEGER NOT NULL DEFAULT 9,
  owner_id INTEGER,
```

```
  FOREIGN KEY (owner_id) REFERENCES users(id)
);

CREATE TABLE wake_logs (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  device_id INTEGER NOT NULL,
  requested_by INTEGER NOT NULL,
  requested_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  status TEXT NOT NULL,
  details TEXT,
  FOREIGN KEY (device_id) REFERENCES devices(id),
  FOREIGN KEY (requested_by) REFERENCES users(id)
);
```

Приклад реалізації ключового функціоналу надсилання magic packet (Python):

```
import socket
def send_magic_packet(mac: str, broadcast_ip: str, port: int = 9):
    mac = mac.replace(":", "").replace("-", "")
    if len(mac) != 12:
        raise ValueError("Некоректна MAC-адреса")

    data = bytes.fromhex("FF" * 6 + mac * 16)
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as sock:
        sock.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
        sock.sendto(data, (broadcast_ip, port))
```

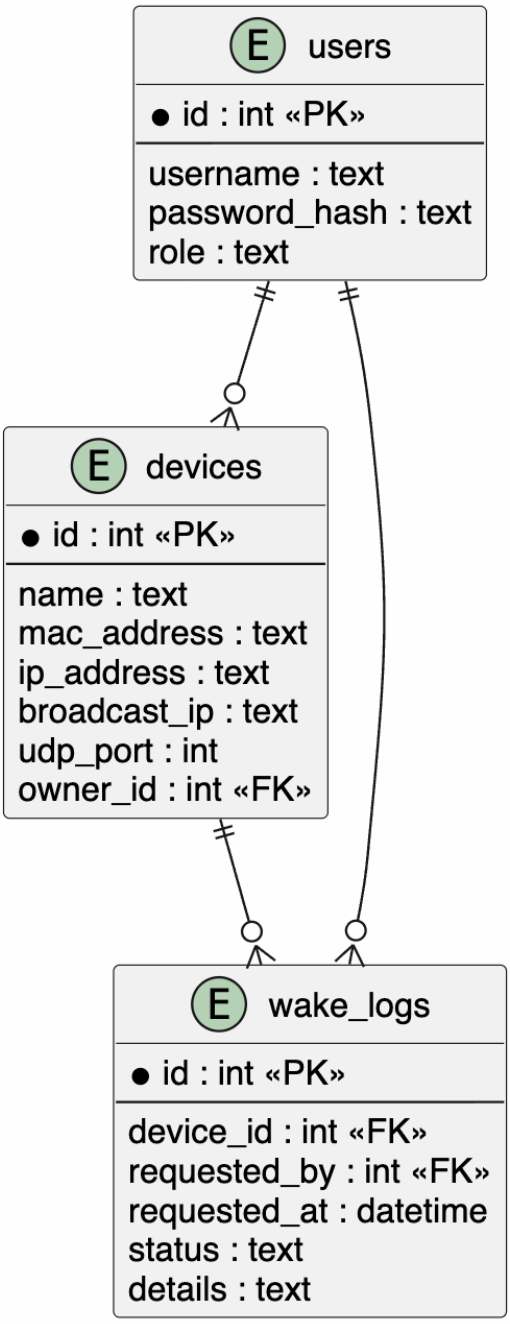


Рисунок 2.1 – ER-структура бази даних

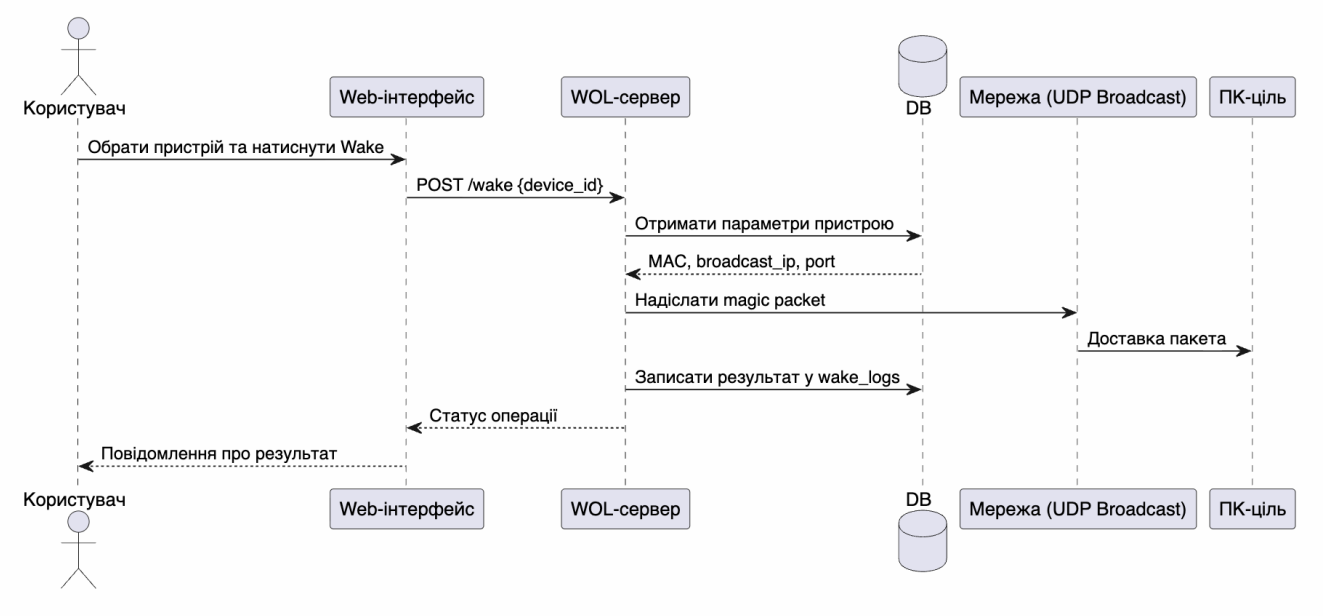


Рисунок 2.2 – UML-діаграма послідовності надсилання пакета пробудження

Отже, на рисунку 2.2 показано послідовність взаємодії між користувачем, серверною частиною, базою даних і цільовим пристроєм під час виконання операції пробудження. Це демонструє, що процес Wake-on-LAN в системі реалізується як керований сценарій із фіксацією кожного ключового етапу.

Додатково слід зазначити, що практична цінність такої послідовності полягає у можливості контролю надійності виконання запиту. Якщо пакет не був доставлений або пристрій не перейшов у робочий стан, система не лише повертає користувачу повідомлення про помилку, а й зберігає діагностичну інформацію в журналі подій. Це спрощує технічний супровід і дозволяє аналізувати типові причини збоїв (некоректна MAC-адреса, недоступний broadcast-сегмент, блокування UDP-портів мережевою політикою) [18].

Окрему увагу в логіці підсистеми приділено попередній валідації параметрів пристрою. Перед надсиланням magic packet перевіряється формат MAC-адреси, наявність коректної broadcast-адреси та відповідність порту значенням, допустимим у конфігурації мережі. Такий підхід знижує кількість хибних запитів і мінімізує непотрібне навантаження на мережеву інфраструктуру.

Крім того, журнал wake_logs виконує не лише облікову, але й аналітичну функцію. На основі накопичених записів можна оцінювати частоту звернень до

окремих вузлів, часові піки активності, відсоток успішних пробуджень і стабільність роботи системи у різних умовах експлуатації. У перспективі це створює основу для впровадження автоматизованих правил, наприклад повторної спроби надсилання пакета або сповіщення адміністратора при серії невдалих запусків [18].

Висновок до розділу 2

Сформовано цілісну систему вимог до майбутнього рішення віддаленого керування живленням ПК. Визначено функціональні можливості системи, критерії зручності для користувача, вимоги до безпеки та стабільності роботи в мережевому середовищі. Це дало змогу чітко зафіксувати межі проєкту, узгодити очікувані результати та закласти основу для подальшої формалізації всіх процедур керування.

Побудована архітектура підтвердила доцільність клієнт-серверного підходу для реалізації Wake-on-LAN у локальних і сегментованих мережах. Розподіл ролей між інтерфейсом користувача, серверною логікою та мережевим модулем забезпечує керованість системи, масштабованість і спрощує супровід. Структурний опис взаємодії компонентів показав, що обране технічне рішення дозволяє ефективно організувати передачу команд, контроль доступу та обробку помилок.

Запропонована модель даних і логіка обробки запитів забезпечують цілісність інформації про пристрої, користувачів і результати операцій пробудження. Передбачене журналювання дій створює підґрунтя для аудиту, діагностики та подальшого аналізу надійності системи в різних мережевих умовах. Таким чином, розділ 2 формує завершену проєктну основу, достатню для переходу до поглибленого аналітичного обґрунтування функціонування системи у наступному розділі.

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ WAKE-ON-LAN

3.1 Реалізація сервера для відправки magic packet

Технологія Wake-on-LAN базується на простому, але надійному механізмі: мережевий адаптер ПК, який перебуває у стані сну або вимкнення (за наявності чергового живлення), аналізує вхідний трафік і реагує на спеціальний шаблон байтів. Цей шаблон називається magic packet і має строго визначену структуру, що робить його легко детектованим на рівні мережевого контролера [19].

Практична цінність такого підходу полягає в тому, що сервер керування не потребує встановлення повноцінного агентного ПЗ на вимкненій машині: достатньо знати MAC-адресу цільового адаптера, broadcast-адресу сегмента мережі та UDP-порт. Водночас результативність пробудження залежить не лише від коректності пакета, а й від налаштувань BIOS/UEFI, енергозбереження ОС, політик комутаторів і маршрутизаторів.

Структура magic packet та принцип його розпізнавання

Magic packet складається з:

1. Префікса синхронізації: FF FF FF FF FF FF (6 байт).
2. MAC-адреси цільового пристрою, повтореної 16 разів поспіль (16 × 6 = 96 байт).

Загальна довжина корисного навантаження: 6 + 96 = 102 байти (без урахування Ethernet/IP/UDP-заголовків).

На рисунку 3.1 буде показано логічну побудову magic packet та послідовність повторення MAC-адреси.

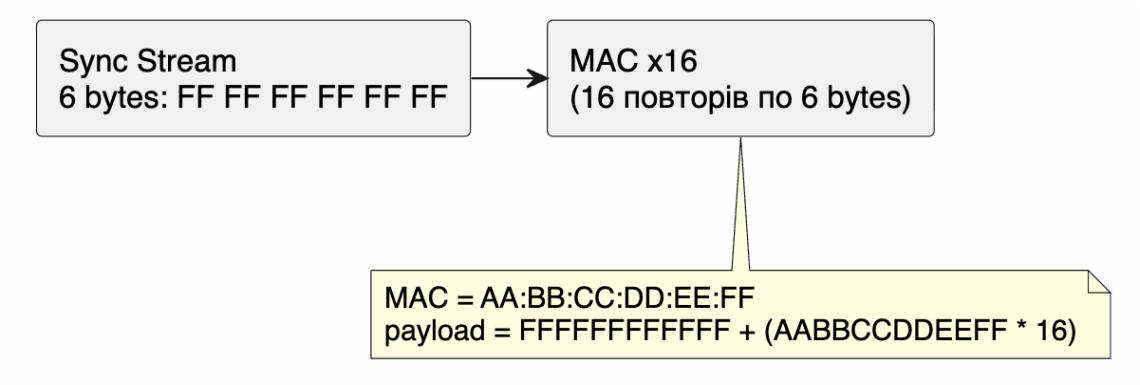


Рисунок 3.1 – Структура magic packet

Отже, на рисунку 3.1 показано, що пакет має фіксований формат без службових полів авторизації, тому ключовим фактором є точність формування байтової послідовності та її доставка в правильний мережевий сегмент.

Для прикладу, якщо MAC-адреса пристрою D8:5E:D3:11:22:33, початок payload матиме вигляд: FF FF FF FF FF FF D8 5E D3 11 22 33 D8 5E D3 11 22 33 ... (повтор 16 разів) [20].

Практичні вхідні параметри для формування пакета

Перед формуванням пакета система повинна зібрати й валідувати технічні параметри. Це мінімізує частку помилок ще до моменту надсилання.

Таблиця 3.1

Мінімальні параметри для успішного Wake-on-LAN

Параметр	**Приклад**	**Призначення**	**Типові**
MAC-адреса	D8:5E:D3:11:22:33	Ідентифікація мережевого адаптера	Неправильний формат, зайві символи
Broadcast IP	192.168.1.255	Доставка пакета всім вузлам сегмента	Вказано unicast IP замість broadcast
UDP порт	9	Транспорт WOL-повідомлення	Закритий порт ACL/Firewall

Продовження таблиці 3.1

Інтерфейс відправника	eth0 / en0	Вибір мережі при багатьох NIC	Відправка не в той VLAN/підмережу
Стан BIOS/UEFI	Wake on PCI-E: Enabled	Дозвіл апаратного пробудження	Вимкнено WOL у firmware
Налаштування ОС цілі	Allow wake from magic packet	Підтримка пробудження драйвером	Відключено в драйвері NIC

За практикою експлуатації, найбільша частка невдач виникає не через код формування пакета, а через некоректні мережеві передумови: помилкову broadcast-адресу, відсутність маршруту до потрібного сегмента або заборону directed broadcast на маршрутизаторі.

Алгоритм формування пакета на сервері керування

На рисунку 3.2 буде показано покроковий алгоритм перевірки параметрів, побудови payload та відправки UDP-пакета [21].

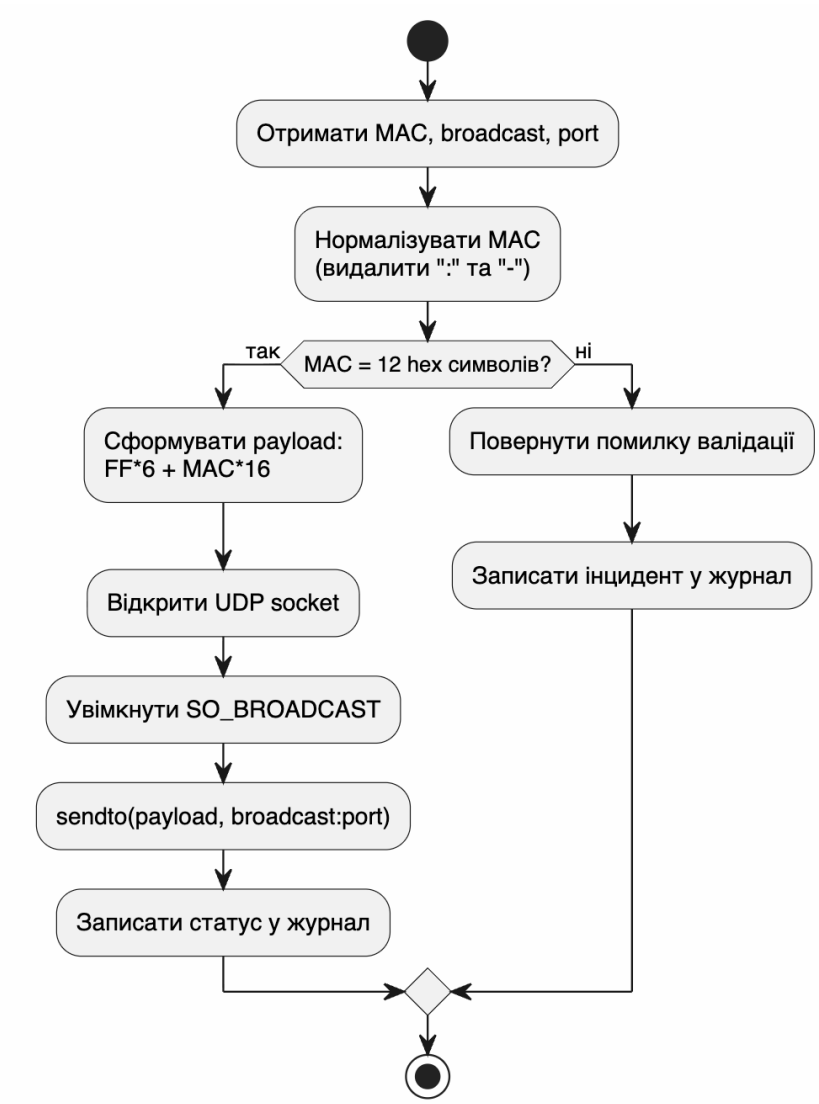


Рисунок 3.2 – Алгоритм формування та надсилання magic packet

Отже, на рисунку 3.2 показано, що надійність реалізації забезпечується двома рівнями контролю: синтаксичною валідацією MAC-адреси та обов’язковим журналюванням результату кожної спроби відправки.

Нижче наведено практичний приклад на Python із валідацією, який можна включати у серверну частину [21]:

```
import re
import socket
MAC_RE = re.compile(r"^[0-9A-Fa-f]{12}$")
```

```
def build_magic_packet(mac: str) -> bytes:
    normalized = mac.replace(":", "").replace("-", "").strip()
    if not MAC_RE.match(normalized):
        raise ValueError("Некоректний формат MAC-адреси")
    return bytes.fromhex("FF" * 6 + normalized * 16)
def send_wol(mac: str, broadcast_ip: str, port: int = 9, timeout: float = 1.0) -> None:
    payload = build_magic_packet(mac)
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as sock:
        sock.settimeout(timeout)
        sock.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
        sock.sendto(payload, (broadcast_ip, port))
```

Для стеку Node.js логіка аналогічна:

```
import dgram from "node:dgram";
function buildMagicPacket(mac) {
  const normalized = mac.replace(/[:-]/g, "").toUpperCase();
  if (!/^[0-9A-F]{12}$/.test(normalized)) throw new Error("Bad MAC");
  const sync = "FF".repeat(6);
  const body = normalized.repeat(16);
  return Buffer.from(sync + body, "hex");
}
export function sendWol(mac, broadcastIp, port = 9) {
  const payload = buildMagicPacket(mac);
  const socket = dgram.createSocket("udp4");
  socket.bind(() => {
    socket.setBroadcast(true);
    socket.send(payload, port, broadcastIp, () => socket.close());
  });
}
```

Процес надсилання в мережі та контроль доставки

На рисунку 3.3 буде показано послідовність взаємодії компонентів під час надсилання пакета в локальному сегменті [22].

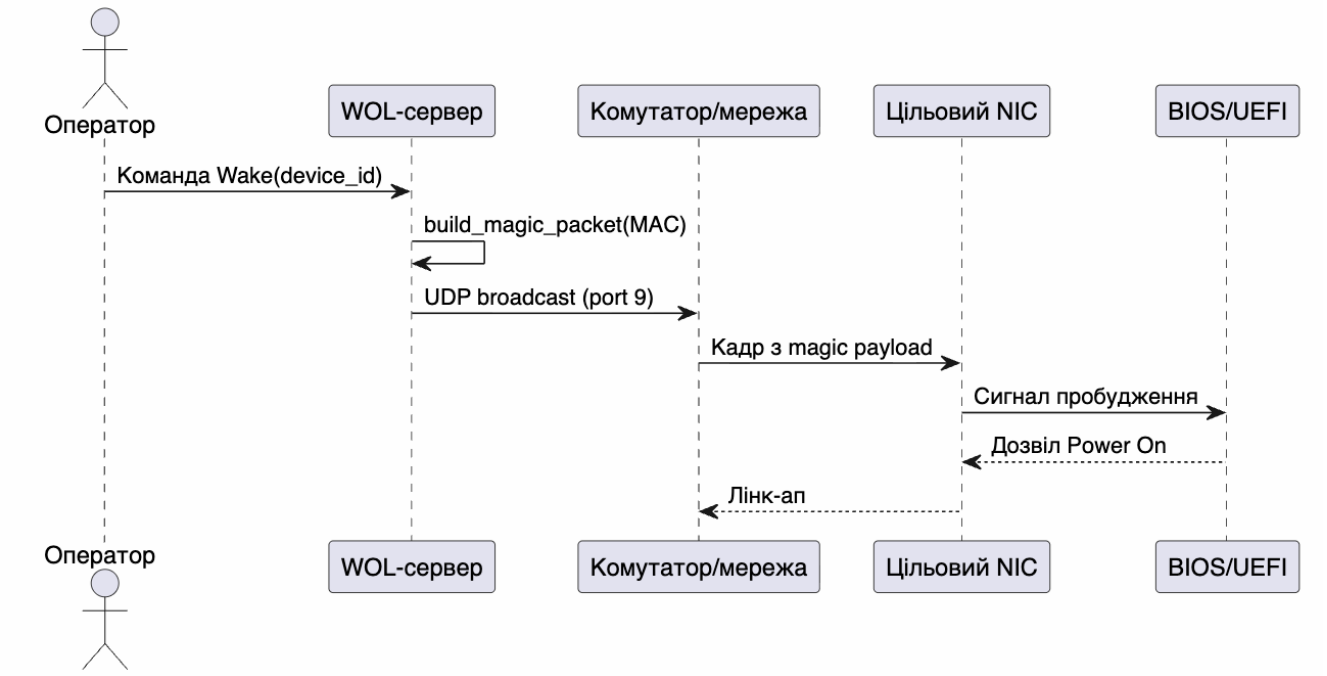


Рисунок 3.3 – Послідовність надсилання magic packet

Отже, на рисунку 3.3 показано, що сервер не “вмикає” ПК напряму; він лише доставляє мережевий тригер, а фінальне пробудження виконується апаратно-програмним контуром NIC + BIOS/UEFI.

У практичному розгортанні рекомендовано:

1. Використовувати directed broadcast конкретної підмережі (наприклад, 192.168.10.255), а не 255.255.255.255.
2. Фіксувати порт (7 або 9) і не змінювати його без потреби.
3. Застосовувати повторну відправку 2-3 рази з інтервалом 200-500 мс для “шумних” мереж.
4. Розділяти корпоративні VLAN і явно дозволяти WOL-трафік у правилах ACL.

Верифікація та діагностика (практичний контроль)

Для перевірки факту відправки використовують Wireshark із фільтром:

udp.port == 9

або

udp.port == 7

Ознаки коректного пакета:

1. UDP datagram містить payload довжиною 102 байти.
2. Перші 6 байтів payload дорівнюють FF.
3. Далі MAC-адреса повторюється 16 разів без розривів [23].

Таблиця 3.2 – Критерії приймання результату в підсистемі WOL

Критерій	Метод перевірки	Очікуваний результат
Коректність формату MAC	Regex/парсер на сервері	Запит проходить валідацію
Відправка пакета	Логи сервера + Wireshark	Є UDP-пакет із payload 102 байти
Досягнення сегмента	Capture у VLAN цілі	Пакет присутній у цільовій мережі
Реакція пристрою	Ping/Agent/ARP після 10-60 с	Вузол стає доступним
Аудит події	Запис у wake_logs	Зафіксовано success або failed

Якщо пакет видно в мережі, але ПК не прокидається, послідовність діагностики така:

1. Перевірити BIOS/UEFI (Wake on LAN, Power on by PCI-E).
2. Перевірити драйвер NIC в ОС (дозвіл wake only by magic packet).
3. Перевірити фізичне живлення і режим S3/S4/S5.
4. Перевірити MAC саме того адаптера, який має WOL-підтримку.
5. Перевірити, чи не відключене чергове живлення на материнській платі [23, 24].

Практичні обмеження при роботі через маршрутизовані мережі

Для міжмережевого WOL існують типові бар’єри:

– Блокування directed broadcast на L3-пристроях (з міркувань безпеки).

– NAT не має активного ARP-зв’язку для вимкненого вузла.

– Мережеві політики SOC/Firewall блокують UDP broadcast.

Робочі підходи для практичної реалізації:

– Локальний WOL-проксі в кожній підмережі (найбільш керований варіант).

– VPN-доступ до внутрішнього сегмента з відправкою WOL “зсередини”.

– Комбінування WOL із out-of-band керуванням (Intel AMT/IPMI) для критичних серверних зон.

Безпека процесу надсилання magic packet

Оскільки сам пакет не має вбудованої автентифікації, безпека досягається на рівні системи керування:

– Автентифікація користувачів (RBAC, MFA для адміністраторів).

– Підпис або токенізація API-запитів на wake.

– Rate limiting (захист від флуду).

– Журналювання IP-джерела, користувача, часу та цільового MAC.

– Сегментація доступу: оператор бачить лише “свої” пристрої.

3.2 Розробка клієнтського інтерфейсу користувача

Клієнтський інтерфейс системи Wake-on-LAN повинен забезпечувати швидке виконання типової дії “обрати пристрій -> надіслати команду пробудження -> отримати зрозумілий статус”. На відміну від загального адміністрування мережі, цей сценарій має високу частоту повторення, тому структура UI орієнтується на мінімальну кількість кроків, передбачуваність елементів керування та швидку інтерпретацію результату.

Обґрунтування структури інтерфейсу базується на трьох принципах:

1. Операційна простота: критична дія (Wake) виконується за 1-2 кліки.
2. Контрольованість: кожен запит має видимий стан і слід у журналі.
3. Безпека: доступ до дій і пристроїв обмежується ролями користувачів.

Функціонально-рольова модель інтерфейсу

У системі доцільно передбачити щонайменше дві ролі:

1. Оператор - перегляд “своїх” пристроїв, запуск пробудження, перегляд власної історії.
2. Адміністратор - повний доступ: пристрої, користувачі, політики, журнал подій.

На рисунку 3.4 буде показано UML use-case структуру клієнтського інтерфейсу та розподіл доступних дій між ролями [25].



Рисунок 3.4 – UML діаграма варіантів використання клієнтського інтерфейсу

На рисунку 3.4 показано рольову модель інтерфейсу, де базовий сценарій пробудження доступний обом ролям, а адміністративні дії виділені в окремий

контур безпеки.

Інформаційна архітектура екранних форм

Раціональна структура клієнтського UI для цього класу задач:

1. Екран авторизації.
2. Головна панель (Dashboard) зі статусами доступності.
3. Список пристроїв з фільтрами, сортуванням та кнопкою Wake.
4. Журнал подій з деталізацією результату.
5. Налаштування (для адміністраторів).

На рисунку 3.5 буде показано рекомендовану структуру навігації між екранними формами та основними переходами користувача.

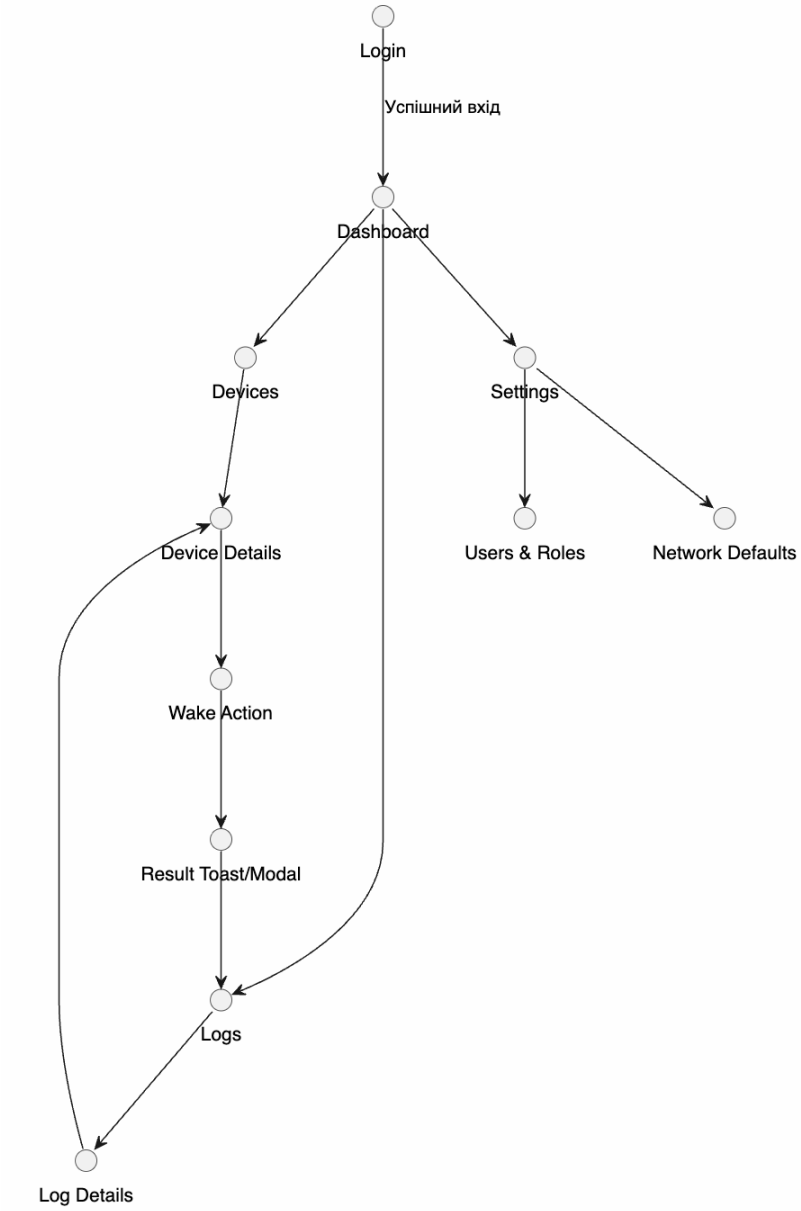


Рисунок 3.5 – Структура навігації клієнтського інтерфейсу

На рисунку 3.5 показано, що навігація орієнтована на “короткий операційний шлях”: з головного екрана користувач переходить до списку пристроїв і виконує пробудження без зайвих проміжних сторінок [26].

Обґрунтування складу елементів на ключовому екрані

“Пристрої” Екран “Пристрої” є центральним, тому доцільно включити:

1. Поле швидкого пошуку (за ім’ям, MAC, IP, локацією).
2. Фільтри (група, VLAN/сегмент, власник, останній статус).
3. Таблицю з ключовими колонками.
4. Контекстні дії (Wake, Details, Ping, History).
5. Індикатори стану останніх запитів.

Таблиця 3.3 – Рекомендовані поля таблиці “Пристрої”

Поле	Призначення	Формат відображення
Назва пристрою	Людинозрозумілий ідентифікатор	PC-Accounting-07
MAC-адреса	Ключ для формування magic packet	D8:5E:D3:11:22:33
Broadcast IP	Цільовий сегмент доставки	192.168.10.255
UDP порт	Параметр відправки	9
Останній результат	Швидка оцінка працездатності	Success/Failed + колір
Час останньої спроби	Актуальність подій	2026-04-20 14:35
Власник/група	Контекст доступу	IT-Support

Щоб зменшити операційні помилки, кнопка Wake повинна бути доступною лише за наявності валідної конфігурації пристрою. Для невалідних записів доцільно показувати неактивну кнопку з поясненням причини (наприклад, “Невірна MAC-адреса”) [26, 27].

Послідовність взаємодії UI з сервером під час Wake-запиту

Перед діаграмою: на рисунку 3.6 буде показано UML-послідовність обробки клієнтського запиту від натискання кнопки до відображення результату.

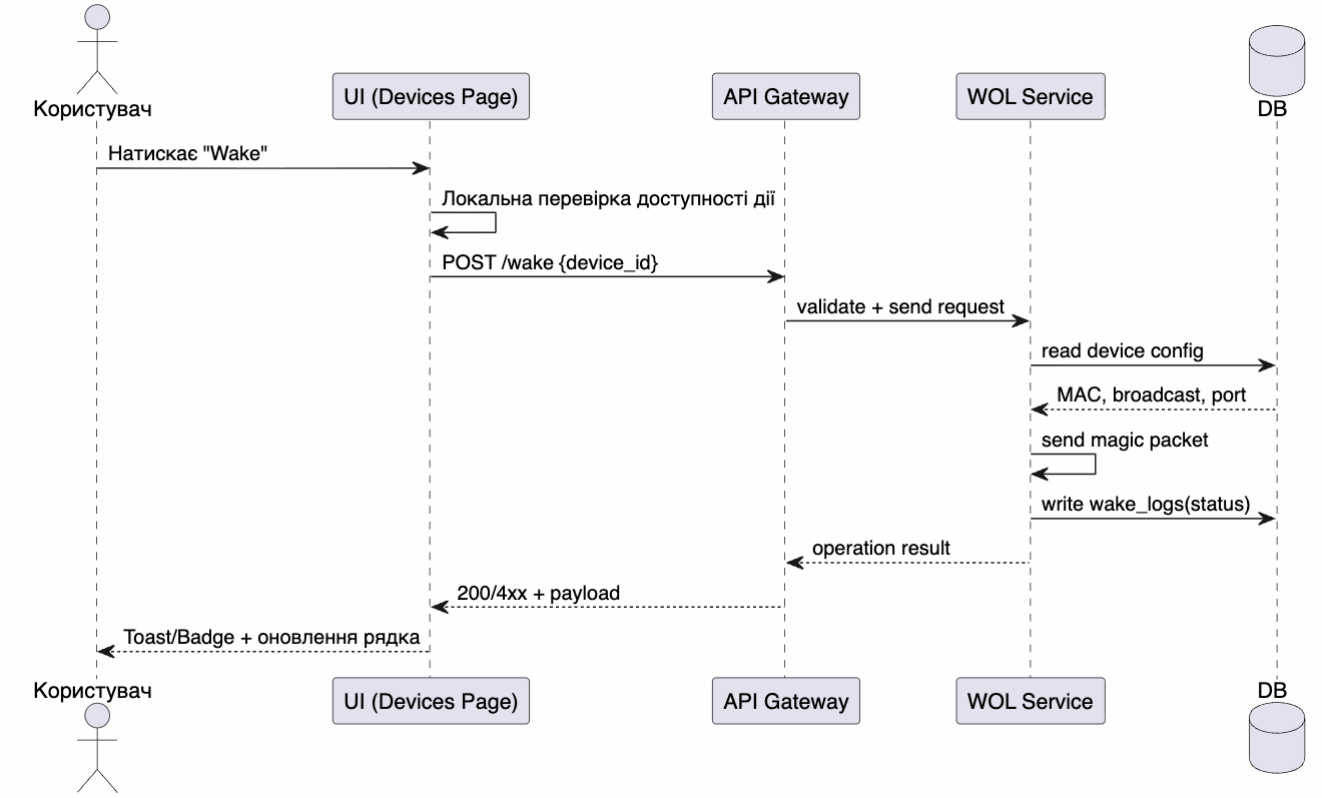


Рисунок 3.6 – Послідовність обробки Wake-запиту в клієнтському інтерфейсі

На рисунку 3.6 показано, що UI не виконує мережеву логіку напряму, а працює через API-контур; це ізолює критичні дії, спрощує аудит і підвищує безпеку.

Таблиця 3.4 – Основні користувацькі сценарії та очікувана поведінка UI

Сценарій	Дія користувача	Реакція інтерфейсу
Швидке пробудження	Натискає Wake у рядку пристрою	Показує Sending..., потім статус
Помилкова MAC	Відкриває картку з некоректними даними	Блокує Wake, пояснює причину

Продовження таблиці 3.4

Відсутній доступ	Оператор відкриває чужий пристрій	Повертає Access denied
Масова робота	Вибір кількох пристроїв, Bulk Wake	Показує прогрес і підсумок
Аналіз інциденту	Перехід у Logs	Фільтрує за device/time/status

Вимоги до візуальної ієрархії та UX-поведінки

Для задач операційного керування особливо важливі читабельність і швидкість сприйняття:

1. Статус успіху: зелений маркер + текст Success.
2. Статус помилки: червоний маркер + короткий діагностичний код.
3. Проміжний стан: нейтральний індикатор Sending....
4. Кнопка Wake в одній і тій самій позиції у всіх рядках.
5. Час події в єдиному форматі (YYYY-MM-DD HH:mm).

Також обов’язкові UX-правила:

1. Оптимістичний відгук UI не замінює фактичний результат API.
2. Повідомлення про помилку має містити рекомендацію (“Перевірте MAC/Broadcast”).
3. Доступність з клавіатури (Tab, Enter, контраст, aria-label).
4. Підтримка мобільної/звуженої ширини через адаптивну таблицю (card-view).

Компонентна модель клієнтської частини

Перед діаграмою: на рисунку 3.7 буде показано компонентну структуру frontend-рішення, яку доцільно використовувати для підтримуваного й масштабованого інтерфейсу [28].

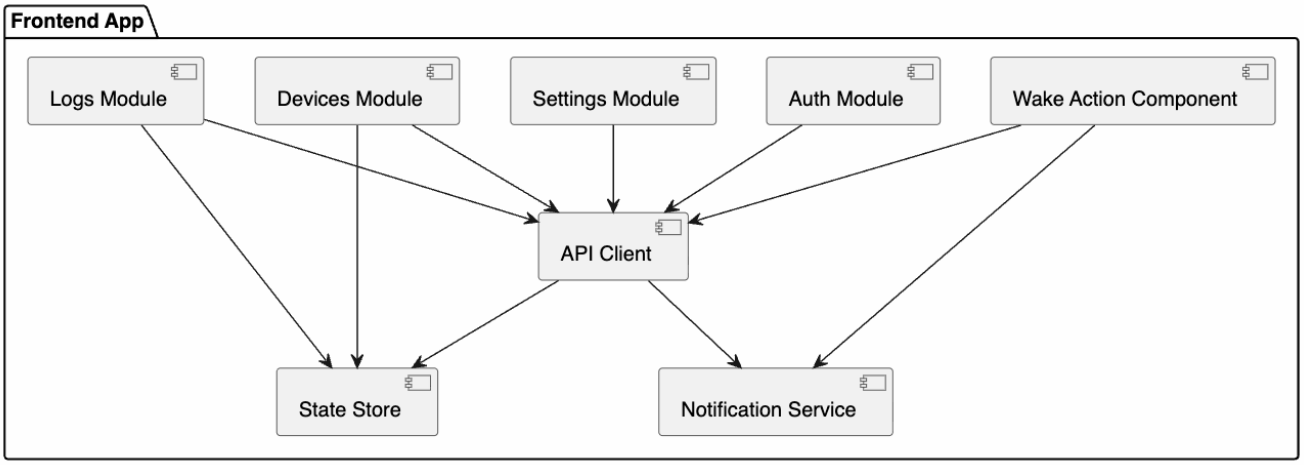


Рисунок 3.7 – UML діаграма компонентів клієнтського інтерфейсу

На рисунку 3.7 показано модульний підхід, де Wake Action Component відокремлений від інших екранів, а спільні сервіси (API Client, State Store, Notification Service) уніфікують поведінку інтерфейсу.

```
async function wakeDevice(deviceId) {
  setRowState(deviceId, { status: "sending" });
  try {
    const res = await fetch("/api/wake", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({ device_id: deviceId })
    });
    const payload = await res.json();
    if (!res.ok) {
      setRowState(deviceId, { status: "failed", details: payload.message });
      showToast(`Помилка: ${payload.message}`, "error");
      return;
    }
    setRowState(deviceId, { status: "success", details: payload.message });
    showToast("Команду Wake успішно надіслано", "success");
```

```
    refreshLogs();
  } catch (e) {
    setRowState(deviceId, { status: "failed", details: "Network error" });
    showToast("Мережева помилка під час надсилання запиту", "error");
  }
}
```

Цей фрагмент демонструє обов’язкові елементи операційного UI:

- Проміжний стан у рядку пристрою.
- Обробку позитивного і негативного сценарію.
- Синхронізацію з журналом подій після дії.

Для обґрунтування структури важливо встановити вимірювані KPI:

Таблиця 3.5 – Базові UX/KPI-показники підсистеми інтерфейсу

Метрика	Цільове значення	Коментар
Час до виконання Wake	до 5 секунд	Від кліку до відображення результату
Кількість кліків у базовому сценарії	1-2	Для вже відомого пристрою
Частка хибних запусків через UI	< 2%	За рахунок валідації і підказок
Час пошуку пристрою	до 10 секунд	Через фільтр/пошук
Задоволеність операторів	> 80%	За результатами опитування

Структура клієнтського інтерфейсу для системи Wake-on-LAN має бути орієнтована на повторювану операційну дію з мінімальним порогом помилки. Запропонована модель поєднує рольовий контроль доступу, короткий сценарій виконання команди, чітку індикацію станів та глибоку трасованість подій через журнал.

3.3 Тестування системи та перевірка працездатності

Перевірка працездатності системи Wake-on-LAN має виконуватися не як одноразова дія “пакет відправлено”, а як комплексна методика, що охоплює коректність конфігурації, факт доставки magic packet, реакцію цільового пристрою, стабільність повторних запусків і стійкість до змін мережевих параметрів. Такий підхід дозволяє отримати об’єктивну оцінку готовності системи до практичної експлуатації в реальному IT-середовищі.

Методика повинна враховувати, що успішний WOL-запуск залежить від багатьох факторів: налаштувань BIOS/UEFI, параметрів драйвера мережевого адаптера, правил VLAN/ACL, стану UDP-маршрутизації, якості каналу, затримок і особливостей політик інформаційної безпеки. Відповідно, результати тестування мають фіксуватися не лише у форматі “success/failed”, а з деталізацією причин і контексту виконання [29, 30].

Мета, об’єкт і критерії перевірки

Мета методики: підтвердити, що система стабільно виконує операцію пробудження пристроїв у типових та граничних мережевих умовах, а також забезпечує відтворюваний журнал результатів для технічного аналізу.

Об’єкт перевірки:

1. Серверний модуль формування і відправки magic packet.
2. Клієнтський інтерфейс ініціації Wake-запиту.
3. База даних журналювання подій.
4. Мережева інфраструктура доставки broadcast/direct-broadcast трафіку.
5. Цільові робочі станції в енергозберігаючих режимах.

Критерії приймання:

1. Коректність формування payload (102 байти, правильна сигнатура).
2. Успішна доставка пакета в цільовий сегмент.
3. Пробудження вузла в допустимий час.
4. Повне журналювання результату і діагностичних ознак.

5. Відсутність критичних відмов при серіях повторних запусків.

На рис. 3.8 буде показано загальний цикл перевірки працездатності системи від підготовки стенду до аналізу результатів і повторних прогонів [31].

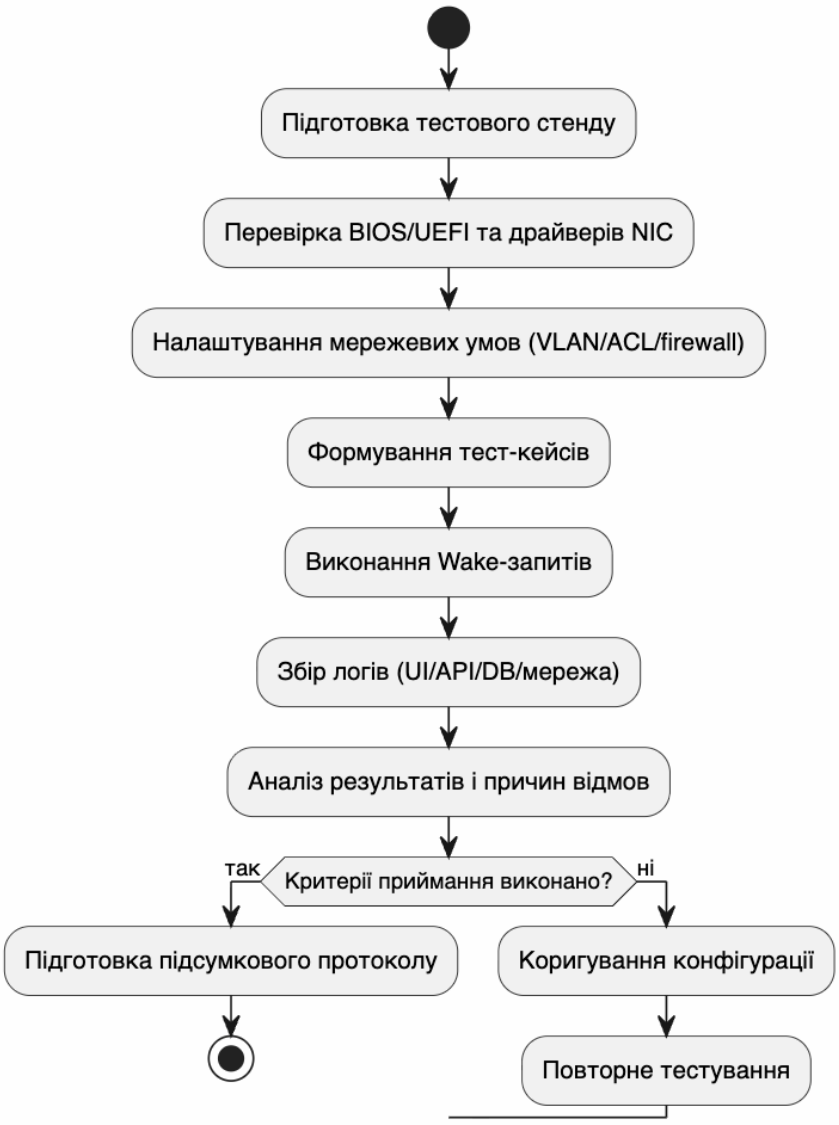


Рисунок 3.8 – Загальний цикл методики перевірки працездатності

На рисунку 3.8 показано ітераційний характер методики, у якій невдала перевірка не завершує процес, а запускає цикл діагностики і повторного підтвердження.

Класифікація мережевих умов для випробувань

Для об’єктивності результати мають охоплювати різні режими роботи

мережі. Мінімальний набір умов:

Таблиця 3.6 – Набір мережевих умов для тестування WOL

ID	Мережевий сценарій	Характеристика середовища	Очікування
N1	Локальний сегмент (L2)	Відправник і ціль в одній підмережі	Найвищий % успіху
N2	Інший VLAN через L3	Потрібен directed broadcast/proxy	Успіх за правильної маршрутизації
N3	VPN-підключення оператора	Команда з віддаленого доступу	Збереження працездатності API
N4	Висока затримка/втрати	Емуляція навантаженої мережі	Допустиме зростання часу реакції
N5	Обмеження ACL/Firewall	Часткове блокування UDP	Контрольована відмова з логом
N6	Масові одночасні запити	Серія на 10-50 пристроїв	Стабільність без деградації

Передумови і підготовка тестового стенду

До старту випробувань потрібно забезпечити:

- Увімкнений Wake on LAN у BIOS/UEFI.
- Актуальні драйвери NIC та дозволи на пробудження.
- Фіксовані MAC-адреси і коректні broadcast-адреси в БД.
- Синхронізований час на сервері й клієнтах (NTP).
- Наявність засобів мережевого моніторингу (Wireshark/tcpdump).
- Виділений журнал тестових прогонів у БД (wake_logs із позначкою test run).

Перед діаграмою: на рисунку 3.9 буде показано логічну конфігурацію

тестового стенду для перевірки системи в різних сегментах мережі [31].

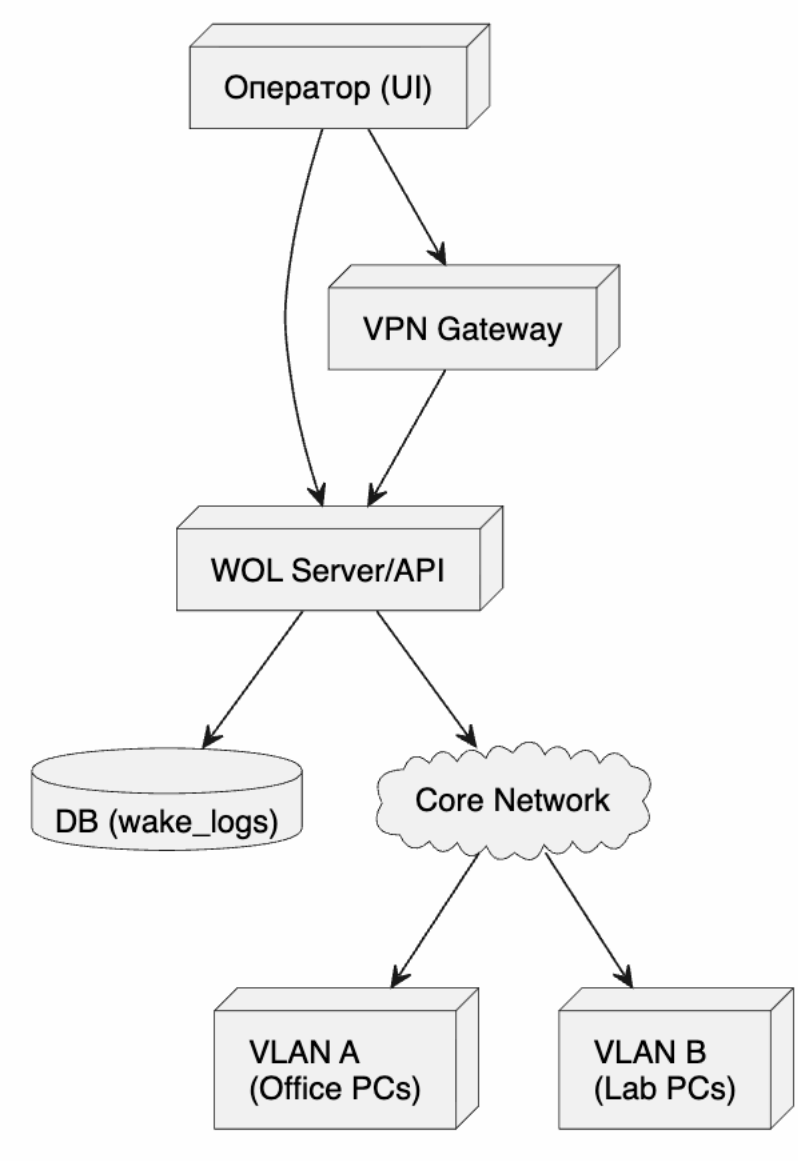


Рисунок 3.9 – Логічна схема тестового стенду

На рис. 3.9 показано стенд, у якому можна відпрацювати як базовий локальний сценарій, так і складні випадки з міжсегментною взаємодією та віддаленим доступом через VPN.

На рисунку 3.10 буде показано потік формування діагностичного запису від моменту запиту до збереження в БД.

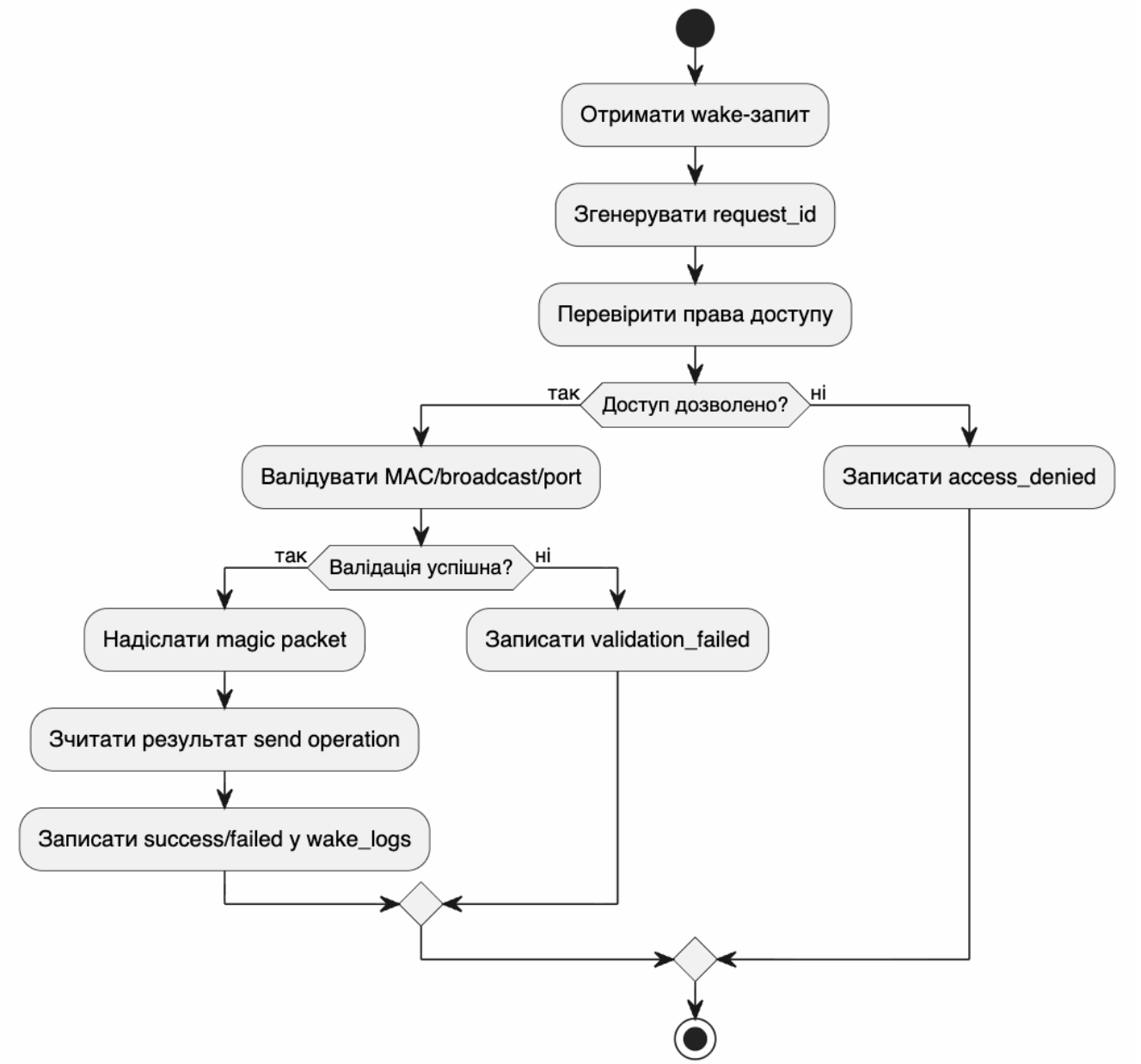


Рисунок 3.10 – Потік журналювання подій Wake-запиту

На рис. 3.10 показано, що незалежно від підсумку запиту система зберігає уніфікований трасувальний запис, що критично важливо для технічного аудиту та подальшого аналізу інцидентів.

Висновок до розділу 3

У третьому розділі виконано комплексне аналітичне обґрунтування функціонування системи Wake-on-LAN, у межах якого деталізовано механізм формування й надсилання magic packet, визначено критичні параметри коректної доставки пакета та встановлено практичні умови, за яких забезпечується стабільне пробудження цільових пристроїв. Узагальнення теоретичних і прикладних аспектів підтвердило, що результативність технології залежить не лише від правильності байтової структури пакета, а й від узгодженої роботи мережевої інфраструктури, налаштувань BIOS/UEFI та політик доступу.

Обґрунтована структура клієнтського інтерфейсу показала доцільність побудови операційно орієнтованого UI з коротким сценарієм виконання команди, прозорою індикацією станів і обов’язковим журналюванням подій. Запропонований підхід забезпечує зручність роботи оператора, зменшує ймовірність помилкових дій та створює умови для контрольованого адміністрування системи в багатосегментних мережах.

Сформована методика перевірки працездатності в різних мережевих умовах дозволяє відтворювано оцінювати якість роботи системи за вимірюваними критеріями успішності, часу реакції, стабільності та повноти аудиту. Отримані в розділі результати формують достатню теоретико-практичну основу для підсумкового узагальнення ефективності запропонованого рішення та підтверджують його готовність до подальшого застосування в реальних умовах експлуатації.

ВИСНОВКИ

У кваліфікаційній роботі комплексно проаналізовано проблематику віддаленого керування живленням комп’ютерів у сучасних мережевих середовищах та обґрунтовано актуальність застосування технології Wake-on-LAN для підвищення керованості IT-інфраструктури.

У теоретичній частині розглянуто принципи роботи Wake-on-LAN, структуру magic packet, особливості мережевої доставки в локальних і маршрутизованих сегментах, а також ключові обмеження, пов’язані з налаштуваннями BIOS/UEFI, мережевих адаптерів і політик безпеки. Це дозволило сформувати цілісне уявлення про умови коректного функціонування технології в реальних сценаріях використання.

У проєктній частині розроблено систему вимог до рішення, визначено архітектурну модель взаємодії клієнтського інтерфейсу, серверної логіки та мережевого контуру, а також запропоновано структуровану модель даних для зберігання відомостей про пристрої, користувачів і результати виконаних операцій. Такий підхід забезпечує масштабованість, контрольованість і зручність супроводу системи.

У межах аналітичного опрацювання функціонування системи виокремлено критичні фактори успішного пробудження: коректність MAC-адресації, правильний вибір broadcast-адреси та порту, доступність мережевого маршруту, узгодженість прав доступу й повнота журналювання.

Підсумково можна стверджувати, що поставлену мету роботи досягнуто: проаналізовано предметну область, розглянуто теоретичні та практичні аспекти технології, розроблено проєктні й методичні засади функціонування системи та виокремлено умови її надійної експлуатації. Отримані результати можуть бути використані як методична й прикладна база для впровадження рішень віддаленого керування живленням у навчальних, офісних і корпоративних мережа

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wake-on-LAN [Електронний ресурс] // Wikipedia. – Режим доступу: https://en.wikipedia.org/wiki/Wake-on-LAN .
2. ManageEngine. What is Wake-on-LAN? [Електронний ресурс]. – Режим доступу: https://www.manageengine.com/products/oputils/tech-topics/what-is-wake-on-lan.html .
3. OneUptime. How to Configure Broadcast Forwarding for Wake-on-LAN [Електронний ресурс]. – Режим доступу: https://oneuptime.com/blog/post/2026-03-20-configure-broadcast-forwarding-wake-on-lan/view .
4. SuperOps. What is Wake-on-LAN and how it works [Електронний ресурс]. – Режим доступу: https://superops.com/tech-hub/what-is-wake-on-lan.
5. Wireshark Wiki. WakeOnLAN Protocol [Електронний ресурс]. – Режим доступу: https://wiki.wireshark.org/WakeOnLAN.
6. Devolutions. Wake-on-LAN Documentation [Електронний ресурс]. – Режим доступу: https://docs.devolutions.net/rdm/kb/knowledge-base/wake-on-lan/ .
7. Ubuntu Documentation. Wake-on-LAN [Електронний ресурс]. – Режим доступу: https://doc.ubuntu-fr.org/wakeonlan .
8. DD-WRT Wiki. Wake-on-LAN Setup [Електронний ресурс]. – Режим доступу: https://wiki.dd-wrt.com/wiki/index.php/WOL .
9. Lifewire. How to Set Up and Use Wake-on-LAN [Електронний ресурс]. – Режим доступу: https://www.lifewire.com/wake-on-lan-4149800.
10. Tanenbaum A. Computer Networks. – 5th ed. – Pearson, 2011. – 960 p.
11. Kurose J., Ross K. Computer Networking: A Top-Down Approach. – 7th ed. – Pearson, 2017. – 864 p.
12. Stallings W. Data and Computer Communications. – 10th ed. – Pearson, 2013. – 800 p.
13. Comer D. Internetworking with TCP/IP. – Vol. 1. – Pearson, 2015. – 672 p.

14. Fielding R. Architectural Styles and the Design of Network-based Software Architectures (REST). – University of California, 2000.

15. Bass L., Clements P., Kazman R. Software Architecture in Practice. – 3rd ed. – Addison-Wesley, 2012. – 624 p.

16. Microsoft Learn. Wake on LAN behavior – Windows Client [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/troubleshoot/windows-client/setup-upgrade-and-drivers/wake-on-lan-feature.

17. Microsoft Learn. Wake On LAN for Surface devices [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/surface/wake-on-lan-for-surface-devices.

18. Intel Support. Enabling Wake-on-LAN [Електронний ресурс]. – Режим доступу: https://www.intel.com/content/www/us/en/support/articles/000059062/ethernet-products/intel-killer-ethernet-products.html.

19. Intel Support. Wake on Lan (WOL) Does Not Work on Intel Ethernet Controller I225-V [Електронний ресурс]. – Режим доступу: https://www.intel.com/content/www/us/en/support/articles/000060275/ethernet-products/gigabit-ethernet-controllers-up-to-2-5gbe.html.

20. Intel Support. Wake on Lan Not Working on Intel Ethernet Connection I219-V [Електронний ресурс]. – Режим доступу: https://www.intel.com/content/www/us/en/support/articles/000092746/ethernet-products/gigabit-ethernet-controllers-up-to-2-5gbe.html.

21. Intel Support. Unable to Enable Wake-On LAN for Intel i210-AT on Intel NUC 9 Extreme Kit [Електронний ресурс]. – Режим доступу: https://www.intel.com/content/www/us/en/support/articles/000096664/intel-nuc.html.

22. MikroTik Documentation. Wake on LAN – RouterOS [Електронний ресурс]. – Режим доступу: https://help.mikrotik.com/docs/spaces/ROS/pages/120782862/Wake+on+LAN.

23. Cisco. Troubleshoot Wake on LAN Feature on Catalyst 9500 Series Switches [Електронний ресурс]. – Режим доступу:

https://www.cisco.com/c/en/us/support/docs/switches/catalyst-9500-switch/223266-troubleshoot-wake-on-lan-feature-on.html.

24. ASUS Support. [Motherboard] How to set and enable WOL (Wake On LAN) function in BIOS [Електронний ресурс]. – Режим доступу: https://www.asus.com/support/faq/1045950/.

25. ASUS Support. [Notebook] How to enable Wake on LAN (WOL) feature [Електронний ресурс]. – Режим доступу: https://www.asus.com/us/support/faq/1049115/.

26. ASUS Support. [Commercial DT/AIO/MiniPC] How to enable Wake on LAN (WOL) feature [Електронний ресурс]. – Режим доступу: https://www.asus.com/support/faq/1048459/.

27. Debian Wiki. WakeOnLan [Електронний ресурс]. – Режим доступу: https://wiki.debian.org/WakeOnLan.

28. Wireshark Documentation. Display Filter Reference: User Datagram Protocol [Електронний ресурс]. – Режим доступу: https://www.wireshark.org/docs/dfref/u/udp.html.

29. Wireshark Documentation. Wireshark User’s Guide [Електронний ресурс]. – Режим доступу: https://www.wireshark.org/docs/wsug_html/.

30. NIST. The NIST Cybersecurity Framework (CSF) 2.0 [Електронний ресурс]. – Режим доступу: https://doi.org/10.6028/NIST.CSWP.29.

31. Python Software Foundation. socket — Low-level networking interface (Python 3.12) [Електронний ресурс]. – Режим доступу: https://docs.python.org/3.12/library/socket.html.

Додаток А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ
(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій
Кафедра Комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

На тему: «Система віддаленого керування живленням комп'ютера засобами технології Wake-on-LAN»

Виконав студент групи КІД-41
Тимченко Артем Віталійович

Керівник кваліфікаційної роботи:
Кондратюк Борис Олександрович, викладач кафедри

Київ – 2026

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи

Розробка програмної системи віддаленого керування живленням комп'ютера з використанням технології Wake-on-LAN, що забезпечує можливість дистанційного ввімкнення пристроїв через локальну або глобальну мережу.

Об'єкт дослідження

Процеси віддаленого керування живленням комп'ютерів у мережевих середовищах.

Предмет дослідження

Методи та засоби реалізації технології Wake-on-LAN для віддаленого ввімкнення комп'ютерів через мережу в локальних і глобальних мережевих середовищах.

Актуальність

3

Актуальність даного дослідження полягає у розробці та теоретичному обґрунтуванні моделі централізованого управління, яка вирішує проблему фізичної прив'язки фахівця до обладнання. Запропонована архітектура дозволяє дистанційно та масово запускати комп'ютери для проведення регламентних робіт у неробочий час (вночі або на вихідних). Це дає можливість повністю автоматизувати процеси встановлення критичних оновлень операційної системи, розгортання нового програмного забезпечення, антивірусного сканування та резервного копіювання даних. У результаті повністю усувається необхідність фізичної присутності адміністратора біля кожного системного блока, підвищується загальна безпека мережі, а головне — технічне обслуговування відбувається непомітно і не зупиняє робочий процес користувачів».

НАУКОВІ ЗАВДАННЯ ТА МЕТОДИ ДОСЛІДЖЕННЯ

4

Наукові завдання:

1. Аналіз сучасного стану технологій віддаленого керування живленням ПК у мережевих середовищах

2. Огляд принципів роботи Wake-on-LAN та особливостей передачі magic packet у LAN/WAN

3. Аналіз існуючих рішень для дистанційного ввімкнення комп'ютерів (WakeMeOnLan, Depicus, TeamViewer)

4. Формування функціональних та нефункціональних вимог до системи

5. Проєктування архітектури системи: клієнт – сервер – мережевий контур

6. Реалізація серверного модуля відправки magic packet (Node.js/Python, UDP)

7. Розробка клієнтського інтерфейсу та тестування в різних мережевих умовах

Методи дослідження:

- Аналіз принципів роботи Wake-on-LAN та мережевих протоколів
- Проєктування архітектури клієнт-серверної системи
- Реалізація сервера надсилання magic packet (UDP)
- Розробка клієнтського інтерфейсу користувача
- Тестування в локальній мережі, через роутер та VPN

ТЕХНОЛОГІЯ WAKE-ON-LAN: ПРИНЦИПИ РОБОТИ ТА MAGIC PACKET

5



Рисунок 1.3 – Принцип роботи технології Wake-on-LAN

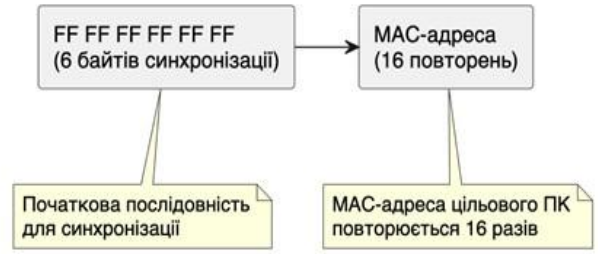


Рисунок 2 – Схема роботи технології Wake-on-LAN

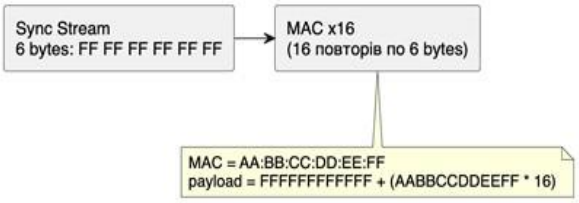


Рисунок 1.2 – Структура magic packet: 6 байт синхронізації + MAC×16

6

МЕРЕЖЕВІ ПРОТОКОЛИ ТА ПОРІВНЯННЯ ІНСТРУМЕНТІВ WAKE-ON-LAN

Таблиця 1 – Порівняння протоколів UDP та TCP у контексті Wake-on-LAN

Критерій	UDP (порт 7/9)	TCP	Висновок
Швидкість доставки	Висока (без встановлення з'єднання)	Нижча (handshake)	UDP — оптимальний вибір для WOL
Надійність	Без підтвердження	Гарантована доставка	Для WOL достатньо UDP
Broadcast-підтримка	Повна підтримка	Відсутня	Лише UDP дозволяє broadcast
Ресурси ПК	Мінімальні (ПК вимкнений)	Потрібна активна TCP-стека	UDP — єдиний варіант для сплячого ПК

Таблиця 2 – Порівняння інструментів Wake-on-LAN

Інструмент	Тип	ОС	WAN	Переваги	Недоліки
WakeMeOnLan	Desktop	Windows	Ні	Простий, безкоштовний	Лише LAN, немає Web UI
Depicus WOL	Desktop/Web	Cross	Так	Web-інтерфейс, MAC-пошук	Платні функції
TeamViewer WOL	Desktop	Cross	Так	Інтеграція RDP, VPN-підтримка	Комерційний, важкий
Розроблена система	Web/API	Cross	Так	Open-source, клієнт-сервер, журнал	Потребує сервера

АРХІТЕКТУРА СИСТЕМИ ТА ВИМОГИ

7

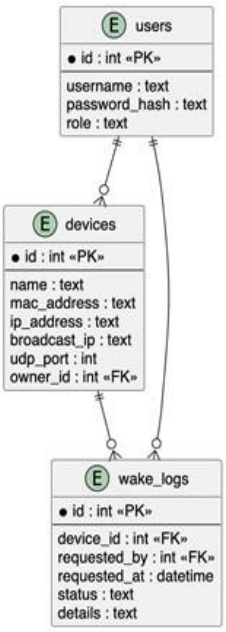


Рисунок 2.1 – Архітектурна схема системи Wake-on-LAN (клієнт – сервер – мережа)

Функціональні вимоги:

• Відправка magic packet за MAC-адресою пристрою

• Зберігання списку пристроїв у базі даних

• Підтримка локальної мережі та Інтернету (через VPN)

• Журналювання всіх wake-запитів (статус, час, деталі)

• Управління користувачами та ролями (адміністратор/оператор)

Нефункціональні вимоги:

• Надійність доставки пакета при коректному налаштуванні

• Веб-інтерфейс (кросбраузерність, адаптивність)

• Безпека: автентифікація, контроль доступу

СТРУКТУРА БАЗИ ДАНИХ ТА UML-ДІАГРАМИ

8



Рисунок 3.4 – Use Case: сценарії роботи WOL

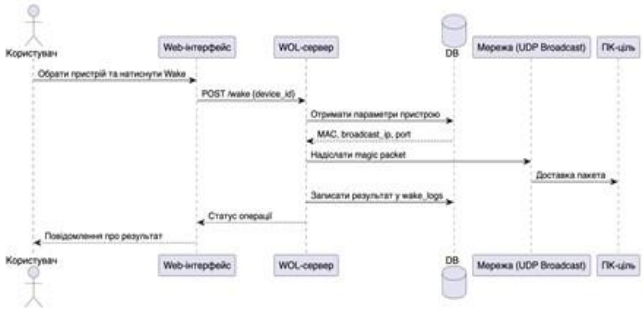


Рисунок 2.2 – UML-діаграма послідовності надсилання пакета пробудження

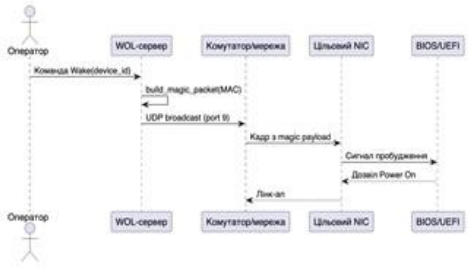


Рисунок 3.3 – Алгоритм формування та надсилання magic packet

9

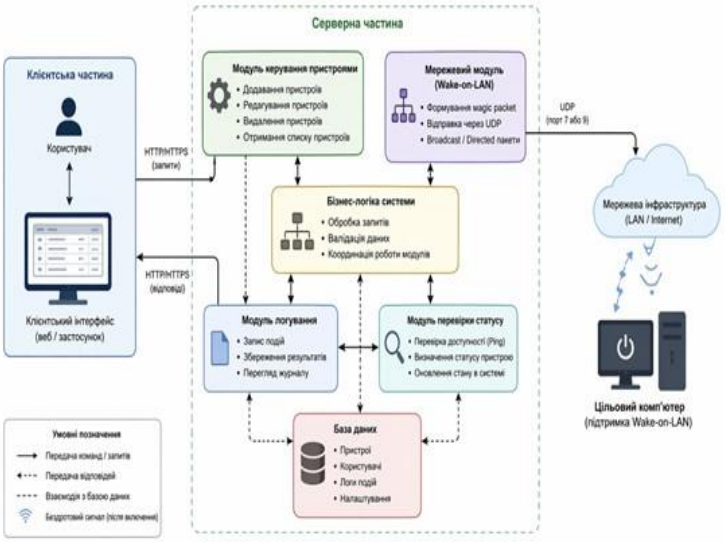


Рисунок 2.1 – ER-структура бази даних (users, devices, wake_logs)

10

РОЗРОБЛЕНО ЛГІЧНУ СТРУКТУРУ СЕРВЕРНОГО МОДУЛЯ ВІДПРАВКИ MAGIC PACKET



Рисунок 3.2 – Алгоритм обробки wake-запиту на серверному рівні

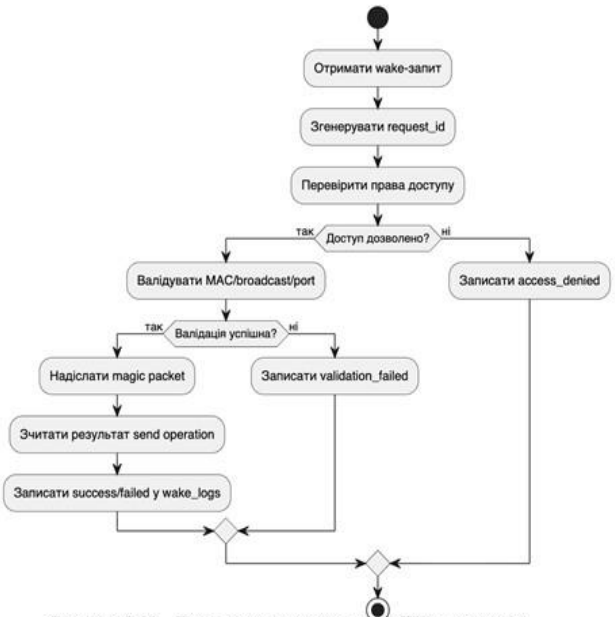


Рисунок 3.10 – Потік журналювання подій Wake-запиту до wake_logs

КЛІЄНТСЬКИЙ ІНТЕРФЕЙС ТА СТРУКТУРА НАВІГАЦІЇ

11

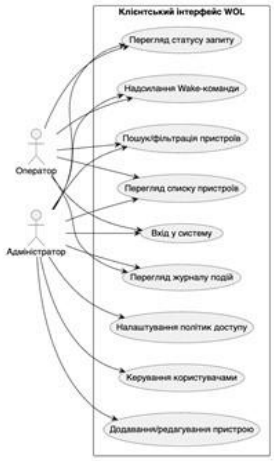


Рисунок 3.4 – UML Use Case клієнтського інтерфейсу

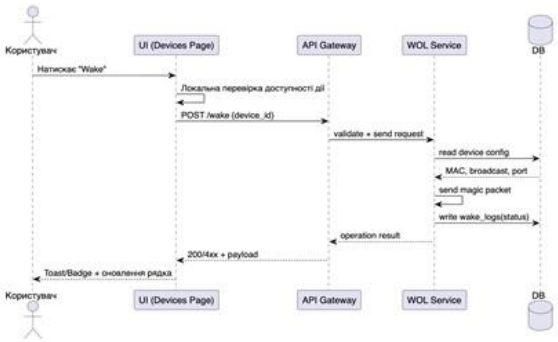


Рисунок 3.6 – UML-діаграма компонентів клієнтського інтерфейсу

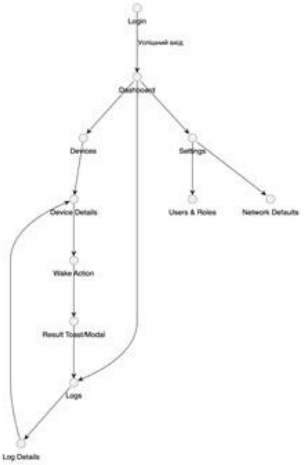


Рисунок 3.5 – Структура навігації клієнтського інтерфейсу

ТЕСТУВАННЯ СИСТЕМИ В РІЗНИХ МЕРЕЖЕВИХ УМОВАХ

12

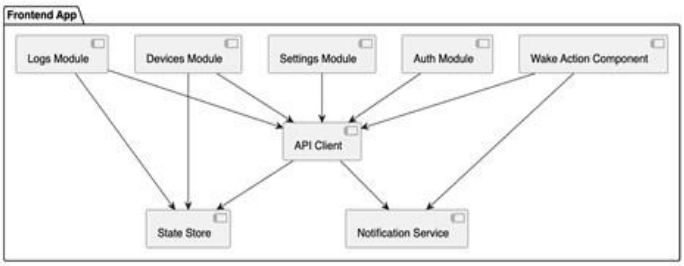


Рисунок 3.7 – Загальний цикл методики перевірки працездатності WOL

Таблиця 3 – Набір мережевих умов для тестування Wake-on-LAN

Умова	Broadcast-адреса	Порт	Очікуваний результат
Локальна мережа (той же сегмент)	192.168.x.255	9 (або 7)	Пробудження < 5 с
Різні сегменти (через роутер)	Directed broadcast	9	Залежить від ip directed-broadcast
VPN-тунель	IP цільового ПК / broadcast VPN	9	Пробудження при коректному маршруті



Рисунок 3.8 – Логічна схема тестового стенду (LAN / VPN / WAN)

ВИСНОВКИ

У результаті проведеного дослідження було створено діючу систему для віддаленого керування живленням комп'ютерів. Ми проаналізували технологію Wake-on-LAN та принципи доставки мережевих пакетів через протокол UDP, що дозволило побудувати надійний алгоритм пробудження ПК. Проведено порівняльний аналіз запропонованої системи з існуючими альтернативними варіантами віддаленого адміністрування. Було спроєктовано сучасну архітектуру, яка включає зручний клієнтський веб-інтерфейс, серверну частину та базу даних. Для забезпечення надійності та безпеки впроваджено систему рольового доступу та автоматичне збереження історії всіх дій адміністратора. Теоретичне обґрунтування та розроблена архітектура підтверджують можливість успішної реалізації такого підходу в умовах сучасних комп'ютерних мереж.