Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

ТЕХНОЛОГІЇ ПОШУКУ НЕСПРАВНОСТЕЙ В СУЧАСНИХ КОМП'ЮТЕРНИХ МЕРЕЖАХ ЗА ВИКОРИСТАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

Виконав студент 2 курсу, групи 1КІ-24м
спеціальності 123 — Комп'ютерна інженерія
_____ Балух Б. А.
Керівник к.т.н., проф. каф. ОТ
_____ Захарченко С. М.
"12" грудня 2025 р.
Опонент доктор філософії, доц. каф. МБІС
_____ Салієва О. В.
"12" грудня 2025 р.

Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О. Д.
_____
"15" 12 2025 р.

ВНТУ 2025

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

Галузь знань — Інформаційні технології

Освітній рівень — магістр

Спеціальність — 123 Комп'ютерна інженерія

Освітньо-професійна програма — Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д.т.н., проф. Азаров О. Д.

"25" вересня 2025 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

Студенту Балуху Богдану Анатолійовичу

1 Тема роботи "Технології пошуку несправностей в сучасних комп'ютерних мережах за використання великих мовних моделей" керівник роботи Захарченко С. М. к.т.н., проф. каф. ОТ, затверджено наказом вищого навчального закладу від 24.09.2025 року № 313.

2 Строк подання студентом роботи: 04.12.2025 р.

3 Вихідні дані до роботи: типи комп'ютерних мереж для пошуку та усунення несправностей — локальні мережі (LAN), глобальні мережі (WAN); типи маршрутизації в мережах — статична, динамічна; інструменти моделювання комп'ютерних мереж — програмне забезпечення візуального моделювання та симуляції мережі передачі даних "Cisco Packet Tracer"; метод пошуку та усунення несправностей — траблшутінг із використанням великих

мовних моделей; метод взаємодії із LLM — prompt-engineering; моделі для дослідження ефективності пошуку та усунення несправностей — GPT-5, Gemini, Claude, LLaMA, Mistral.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз теоретичних основ діагностики та усунення несправностей у комп'ютерних мережах, дослідження методів використання LLM, як інструменту для траблшутінгу в комп'ютерних мережах, експериментальні дослідження ефективності LLM для пошуку несправностей у комп'ютерних мережах, економічна частина.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): блок-схема алгоритму традиційного пошуку несправностей у комп'ютерних мережах, блок-схема алгоритму побудови промптів для мережевої діагностики.

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Захарченко С. М., к.т.н., проф. каф. ОТ		
4	Ратушняк О. Г., к.т.н., доц. каф. ЕПВМ		
Нормоконтроль	Швець С. І., асист. каф. ОТ		

7 Дата видачі завдання: 25.09.2025

8 Календарний план виконання МКР приведений в таблиці 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів МКР	Строк виконання	Примітка
1	Постановка задачі роботи	08.09.2025	виконано
2	Аналіз принципів побудови комп'ютерних мереж, мережевих протоколів та технологій з метою визначення типових проблем	10.09.2025 - 15.09.2025	виконано
3	Вивчення архітектури LLM, їхнього потенціалу в мережевій діагностиці та методів prompt-engineering	16.09.2025 - 23.09.2025	виконано
4	Розробка методики проведення експериментальних досліджень	24.09.2025 - 25.09.2025	виконано
5	Проведення експериментів, оцінка ефективності LLM та формування алгоритму побудови ефективних промптів	26.09.2025 - 11.10.2025	виконано
6	Розрахунок економічної частини	15.10.2025	виконано
7	Оформлення матеріалів до захисту МКР	16.10.2025	виконано
8	Перевірка якості виконання магістерської роботи та усунення недоліків	25.10.2025	виконано
9	Підписи супроводжувальних документів у нормоконтролера, керівника, опонента	03.11.2025	виконано
10	Перевірка на антиплагіат та ШІ	05.11.2025	виконано
11	Попередній захист роботи	10.11.2025	виконано

Студент _____ Балух Б. А.

Керівник _____ к.т.н., проф. каф. ОТ Захарченко С. М.

АНОТАЦІЯ

УДК 004.7

Балух Б. А. Технології пошуку несправностей у сучасних комп’ютерних мережах із використанням великих мовних моделей. Магістерська кваліфікаційна робота зі спеціальності 123 — Комп’ютерна Інженерія, Вінниця: ВНТУ, 2025 — 138 с. На укр. мові. Бібліогр.: 41 назв; рис.: 84; табл. 19.

У роботі досліджено застосування великих мовних моделей для траблшутінгу в комп’ютерних мережах. Проведено аналіз існуючих методів діагностики та усунення типових несправностей у локальних і глобальних мережах. Розроблено алгоритм prompt-engineering для ефективної взаємодії із нейронними мережами. Проведено порівняння різних LLM (GPT-5, Gemini, Claude, LLaMA, Mistral) за критеріями точності, швидкодії та зручності використання. Для проєктування експериментальних комп’ютерних мереж було використано програмне забезпечення візуального моделювання та симуляції мережі передачі даних “Cisco Packet Tracer”. Результати роботи демонструють можливості використання LLM для автоматизації пошуку та усунення помилок у мережевій інфраструктурі, що підвищує ефективність її адміністрування.

Ключові слова: комп’ютерна мережа, велика мовна модель, prompt-engineering, траблшутінг, маршрутизація, LAN, WAN.

ABSTRACT

УДК 004.7

Balukh B. A. Fault finding technologies in modern computer networks using large language models. Master's thesis in the specialty 123 — Computer Engineering, Vinnytsia: VNTU, 2025. In Ukrainian language. Bibliographer: 41 titles; fig.: 84; tabl. 19.

The work investigates the use of large language models for troubleshooting in computer networks. An analysis of existing methods for diagnosing and eliminating typical faults in local and global networks is carried out. A prompt-engineering algorithm is developed for effective interaction with neural networks. A comparison of different LLMs (GPT-5, Gemini, Claude, LLaMA, Mistral) is carried out according to the criteria of accuracy, speed and ease of use. To design experimental computer networks, the software for visual modeling and simulation of the data transmission network "Cisco Packet Tracer" was used. The results of the work demonstrate the possibilities of using LLM to automate the search and elimination of errors in the network infrastructure, which increases the efficiency of its administration.

Keywords: computer network, large language model, prompt-engineering, troubleshooting, routing, LAN, WAN.

ЗМІСТ

ВСТУП

Актуальність теми дослідження полягає в тому, що в сучасному світі практично кожна галузь людської діяльності вимагає використання комп'ютерних мереж. Наукові установи, промислові підприємства, торгові компанії, фінансові установи, освітні заклади та багато інших організацій потребують надійного та ефективного функціонування комп'ютерних мереж для забезпечення безперебійної роботи всієї організації. Водночас, розуміння складності та мінливості мережевих технологій, а також необхідність системного підходу до їх проєктування та управління вимагає спеціалізованих знань та навичок. Зростання обсягів передавання інформації, поява нових сервісів, потреба у високій надійності та захисті даних зумовлюють необхідність розробки та аналізу ефективних рішень для організації та адміністрування комп'ютерних мереж. Одним із ключових завдань є своєчасне виявлення та локалізація несправностей, що впливають на продуктивність і безпеку мережевих сервісів. Традиційні методи пошуку помилок потребують значних трудових і часових витрат, що робить актуальним застосування сучасних технологій штучного інтелекту, зокрема великих мовних моделей, здатних автоматизувати аналіз, прогнозування та усунення несправностей у комп'ютерних мережах.

Метою роботи є розробка методології та алгоритму ефективного використання великих мовних моделей для пошуку і діагностики несправностей у сучасних комп'ютерних мережах.

Для досягення цієї мети необхідно вирішити такі **задачі:**

— провести аналіз сучасного стану та тенденцій розвитку технологій діагностики мережевих несправностей;

— дослідити можливості застосування великих мовних моделей для траблшутінгу в комп'ютерних мережах;

— експериментально оцінити ефективність різних LLM для пошуку та усунення типових проблем, які можуть виникати при моделюванні

комп’ютерних мереж;

— розробити алгоритм конструювання запитів до LLM для ефективного траблшутінгу в комп’ютерних мережах.

Для досягнення поставленої мети використовуються такі **методи дослідження:**

— системний аналіз;

— методи моделювання комп’ютерних мереж;

— методи обробки природної мови (NLP);

— методи машинного навчання;

— експериментальні дослідження;

— порівняльний аналіз.

Об’єктом дослідження є процеси пошуку та діагностики несправностей у комп’ютерних мережах.

Предметом дослідження є методи та інструменти використання великих мовних моделей для аналізу й локалізації несправностей у мережах.

Новизна полягає в тому, що вперше було запропоновано методологію використання великих мовних моделей для пошуку та діагностики несправностей у комп’ютерних мережах, яка ґрунтується на алгоритмі побудови ефективних запитів (prompt-engineering). Це дозволяє систематизувати процес взаємодії з LLM і застосовувати їх, як інтелектуальний інструмент траблшутінгу. Також вперше було проведено порівняльний аналіз ефективності декількох сучасних мовних моделей (GPT-5, Gemini, Claude, LLaMA, Mistral) у вирішенні практичних мережевих задач.

Практичне значення роботи полягає в тому, що розроблений алгоритм конструювання запитів до великих мовних моделей може бути використаний для підвищення ефективності роботи мережевих адміністраторів та інженерів під час пошуку і усунення несправностей. Результати дослідження можуть бути застосовані у навчальному процесі під час підготовки фахівців з комп’ютерних мереж, а також у практичній діяльності підприємств для оптимізації процесів

моніторингу та обслуговування мережевої інфраструктури.

Апробація результатів роботи здійснена в доповіді на LIV Всеукраїнській науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії (2025) та на конференції "Молодь в науці: дослідження, проблеми, перспективи (МН - 2026)".

Матеріали роботи доповідались та опубліковувались [1, 2]:

Захарченко С. М., Балух Б. А. ТЕХНОЛОГІЇ ПОШУКУ НЕСПРАВНОСТЕЙ В СУЧАСНИХ КОМП'ЮТЕРНИХ МЕРЕЖАХ ЗА ВИКОРИСТАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ. Конференція ВНТУ: LIV Всеукраїнська науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2025). Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/24354

Захарченко С. М., Балух Б. А. ДІАГНОСТИКА НЕСПРАВНОСТЕЙ КОНФІГУРАЦІЇ VLAN ТА VTP ЗА ВИКОРИСТАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ. Конференція ВНТУ: Молодь в науці: дослідження, проблеми, перспективи (МН - 2026). Режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2026/paper/view/25875

1 ТЕОРЕТИЧНІ ОСНОВИ ДІАГНОСТИКИ ТА УСУНЕННЯ НЕСПРАВНОСТЕЙ У КОМП'ЮТЕРНИХ МЕРЕЖАХ

1.1 Поняття і класифікація комп'ютерних мереж

Комп'ютерна мережа — це система комп'ютерів та інших пристроїв, які з'єднані каналами зв'язку (кабельними чи бездротовими) для обміну даними, спільного використання ресурсів, таких як принтери та програмне забезпечення, а також для спілкування між користувачами. Прикладами мереж є локальні мережі (LAN) в офісах чи школах та глобальні мережі, як-от Інтернет, що охоплює весь світ.

Спочатку головним завданням об'єднання комп'ютерів у мережу було забезпечення спільного використання ресурсів. Завдяки мережевій взаємодії користувачі та програми могли отримувати доступ до ресурсів інших комп'ютерів. До таких ресурсів зазвичай відносять:

— периферійні пристрої (принтери, сканери, накопичувачі тощо);

— дані;

— обчислювальні ресурси (процесорний час, оперативна пам'ять тощо).

Під час проєктування комп'ютерної мережі важливо визначити її тип, адже це впливає на вибір обладнання, протоколів передачі даних та вимоги до кваліфікації персоналу, який забезпечуватиме її роботу. Класифікація мереж здійснюється за різними критеріями, серед яких: територіальне охоплення, спосіб організації взаємодії комп'ютерів, топологія, рівень доступності тощо [3].

За масштабом поширення комп'ютерні мережі поділяють на два основні типи: локальні та глобальні. Проте з розвитком технологій з'явилися й додаткові класи, серед яких є регіональні, персональні та мережі для з'єднання систем зберігання даних.

Локальні мережі (LAN, Local Area Network) охоплюють комп'ютери та пристрої в межах обмеженої території. Вони зазвичай належать до однієї організації, яка й відповідає за їхнє налаштування та обслуговування. Швидкість

передачі даних у LAN зазвичай становить від 100 Мбіт/с до 1 Гбіт/с. Типові приклади — мережі підприємств, банківських установ, навчальних закладів тощо.

Глобальні мережі (WAN, Wide Area Network) використовуються для об’єднання локальних мереж, розташованих на великих відстанях. Наприклад, компанія з офісами у різних містах чи країнах потребує зв’язку між ними. Такі послуги надають телекомунікаційні провайдери, і користувачі повинні оплачувати доступ. Це зумовлює особливі вимоги до WAN-протоколів: підтримка автентифікації, різна пропускна здатність, можливість забезпечення якості обслуговування (QoS).

Регіональні мережі (MAN, Metropolitan Area Network) займають проміжне місце між LAN і WAN. Вони охоплюють комп’ютери та мережі в межах одного міста чи регіону та зазвичай будуються на основі високошвидкісних міських магістралей.

Персональні мережі (PAN, Personal Area Network) призначені для взаємодії пристроїв, що знаходяться на невеликій відстані один від одного. Найпоширеніший приклад — Bluetooth, який використовується для з’єднання побутової техніки, клавіатур, гарнітур тощо.

Мережі зберігання даних (SAN, Storage Area Network) створюються для підключення зовнішніх пристроїв зберігання (дискових масивів, оптичних приводів) до серверів так, щоб операційна система розпізнавала їх, як локальні ресурси. Це дозволяє кільком серверам одночасно користуватися спільним дисковим простором, що є особливо важливим для технологій віртуалізації та кластеризації [4, 5].

Використання клієнт-серверної архітектури передбачає наявність одного чи кількох спеціалізованих комп’ютерів, які виконують певні функції та надають послуги іншим учасникам мережі. Такі пристрої називаються серверами (від англ. Server — “той, хто обслуговує”). Одним із ключових завдань серверів є керування доступом до мережі та її ресурсів. Це завдання реалізують, наприклад, Active Directory від Microsoft, ZENworks від Micro Focus та інші рішення. У

корпоративних мережах також часто застосовуються файлові, поштові та сервери друку. Важливо зазначити, що клієнт-серверна архітектура лежить в основі мережі Інтернет, адже більшість її ресурсів зберігаються саме на веб-серверах.

При об’єднанні у мережу більше ніж двох робочих станцій необхідно визначитися з конфігурацією фізичних з’єднань, тобто топологією. Топологія мережі — це структура, у якій вершинами графа виступають кінцеві вузли (робочі станції) та комутаційне обладнання (маршрутизатори), а ребрами — фізичні або логічні канали зв’язку. Розрізняють такі основні типи топологій:

— повнозв’язна топологія;

— комірчата (частково зв’язна) топологія;

— кільцева топологія;

— зіркоподібна топологія;

— ієрархічна зірка (дерево);

— загальна шина.

Для повноз’язної топології кожна станція напряму з’єднується з усіма іншими. Такий варіант надзвичайно ресурсозатратний, адже для зв’язку n вузлів потрібно $n(n-1)/2$ дуплексних ліній, тому у великих мережах використовується дуже рідко.

Комірчата (частково зв’язна) топологія — утворюється шляхом видалення частини з’єднань із повнозв’язної схеми. Застосовується переважно у глобальних мережах, де вузлами виступають магістральні маршрутизатори, а з’єднанням — виділені фізичні або віртуальні канали.

В кільцевій топології дані передаються по кільцю від одного комп’ютера до іншого. Перевагою є резервування каналів (кожен вузол доступний двома шляхами), можливість організувати зворотний зв’язок. Недолік — потреба у спеціальних механізмах при відмові одного вузла. Використовувалася у LAN Token Ring та FDDI, а також у деяких варіантах SAN.

У зіркоподібній топології кожен комп’ютер з’єднаний із центральним

багатовходовим пристроєм, який розподіляє інформаційні потоки. Така схема лежить в основі сучасних локальних мереж, де центральним елементом є комутатор або точка доступу.

В основі топології ієрархічної зірки (дерева) лежить з’єднання декількох “зірок” у вигляді ієрархічної структури. У результаті утворюється деревоподібна схема з одним “коренем” і розгалуженими “гілками”. Перевагами такої топології є добра масштабованість та структурованість. До недоліків можна віднести залежність від вузлів верхнього рівня та складність адміністрування у великих масштабах.

Загальна шина є окремим різновидом зіркоподібної топології. У цьому випадку центральним елементом виступає пасивний кабель або концентратор (hub), до якого під’єднується кілька комп’ютерів. Основними перевагами такої топології є низька вартість та простота підключення нових пристроїв. Водночас головним недоліком виступає обмежена продуктивність, адже у певний момент часу передавати дані може лише один комп’ютер [6, 7].

На рисунку 1.1 наведено структурні схеми основних топологій комп’ютерних мереж.

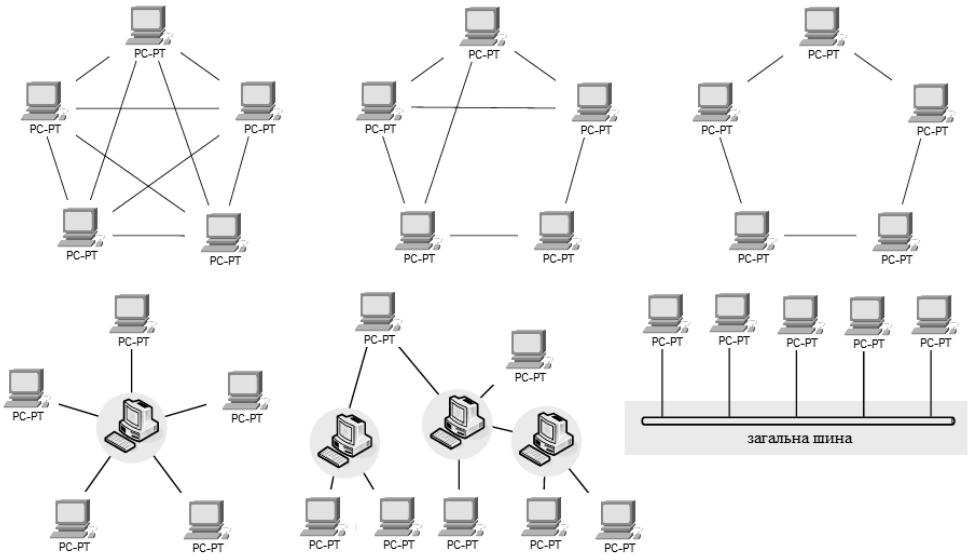


Рисунок 1.1 — Основні топології комп’ютерних мереж

В таблиці 1.1 наведено порівняльний огляд основних топологій комп'ютерних мереж та проблеми, які можуть виникати під час їхнього моделювання (наприклад, в середовищі Cisco Packet Tracer).

Таблиця 1.1 — Топології комп'ютерних мереж та проблеми моделювання

Топологія	Характеристика	Проблеми моделювання
Повнозв'язна	Надійна, але ресурсозатратна	Велика кількість з'єднань, плутанина в інтерфейсах, ризик появи маршрутних петель без STP
Комірчата	Гнучка, але складна для контролю	Проблема зі статичними маршрутами, можлива ізоляція вузлів
Кільцева	Резервування каналів, але вразлива при відмові одного вузла	Ризик появи маршрутних петель без STP, складність діагностики
Зіркоподібна	Проста, централізована	Відмова центрального вузла порушує працездатність усієї мережі
Дерево	Масштабована, структурована, але складна для адміністрування у великих масштабах	Проблеми із налаштуванням VLAN та trunk-портів
Загальна шина	Низька вартість, простота, але обмежена продуктивність	Важко визначити джерело трафіку

1.2 Основні принципи маршрутизації та типові проблеми її конфігурування

1.2.1 Алгоритми маршрутизації та їх класифікація

Маршрутизація — процес визначення (вибору) шляху проходження інформації від джерела до адресата. Основною метою маршрутизації є забезпечення оптимального шляху проходження інформації — її мінімально

можливої затримки і максимальної пропускної спроможності мережі.

Основні цілі маршрутизації полягають у мінімізації (максимізації) значень обраних показників якості обслуговування (швидкості передачі, середньої затримки, джитера, втрат пакетів тощо), а також у забезпеченні збалансованого завантаження мережі, її канальних і буферних ресурсів. Тому основними завданнями, які належать до галузі маршрутизації, є: контроль і збір інформації про стан мережі (її топології, завантаження мережних ресурсів тощо), розрахунок шуканих шляхів (маршрутів) і реалізація маршрутних рішень.

Алгоритм маршрутизації реалізується у складі програмного забезпечення мережного рівня, яке відповідає за вибір вихідного каналу для передавання пакета. Важливо розрізняти поняття маршрутизації та пересилання: пересилання означає обробку вхідних пакетів і визначення для них вихідної лінії згідно з таблицею маршрутизації, тоді як маршрутизація забезпечує формування та оновлення цих таблиць за допомогою відповідних алгоритмів.

Алгоритм вибору маршруту має бути коректним, простим, надійним (адекватно реагувати на зміни топології та трафіку), стійким, справедливим та оптимальним. За принципом роботи алгоритми маршрутизації поділяють на два типи:

— неадаптивні;

— адаптивні.

Неадаптивні не враховують поточний стан мережі чи її топологію, маршрути визначаються заздалегідь і завантажуються в маршрутизатори під час запуску мережі. Адаптивні ж змінюють маршрути залежно від змін у топології чи завантаженості каналів.

Адаптивні алгоритми можуть відрізнятися:

— джерелами отримання інформації;

— моментами внесення змін (через фіксовані інтервали часу, при зміні навантаження або при зміні структури мережі);

— критеріями оптимізації (наприклад, відстань, кількість проміжних

вузлів чи очікуваний час доставки пакета).

Метод вибору найкоротшого шляху широко застосовується завдяки своїй простоті. Його суть полягає в тому, що підмережу подають у вигляді графа: вузли відповідають маршрутизаторам, а дуги — лініям зв’язку. Для визначення маршруту між двома маршрутизаторами алгоритм шукає найкоротший шлях у цьому графі. Критерієм “найкоротшості” може бути кількість проміжних ділянок, фізична довжина каналу, середній розмір черги та час затримки або ж пропускна здатність каналу.

Найпоширенішим алгоритмом пошуку найкоротшого шляху є алгоритм Дейкстри, який полягає у призначенні кожному вузлу “мітки”, що показує відстань до нього від вузла-відправника за найкоротшим на даний момент відомим шляхом. На початку роботи алгоритму всі вузли вважаються недосяжними, але поступово мітки оновлюються, відображаючи оптимальні маршрути. Коли стає зрозуміло, що мітка дійсно відповідає найкоротшому шляху, вона фіксується остаточно й більше не змінюється.

Для прикладу розглянемо зважений ненаправлений граф, зображений на рисунку 1.2, де ваги ребер відповідають відстаням між маршрутизаторами. Потрібно визначити найкоротший шлях від вузла А до вузла Г.

Спершу виділяємо вузол А (позначаємо його чорним колом) і перевіряємо всі вузли, з’єднані з ним, вказуючи біля них відстань до А. Якщо в процесі з’являється коротший шлях до будь-якого вузла, його мітка оновлюється, а разом з нею змінюється і вказівка на вузол, через який проходить найкоротший маршрут. Це дозволяє відновити повний шлях після завершення алгоритму.

Після обробки всіх сусідніх вузлів А вибираємо вузол із найменшою відстанню та позначаємо його як постійний. У нашому прикладі таким вузлом стає Б, який тепер стає новим робочим вузлом.

Далі алгоритм повторюється для вузла Б: досліджуються всі його сусіди. Якщо сума відстані від Б до сусіднього вузла та відстані від А до Б (значення мітки вузла Б) виявляється меншою за поточну мітку сусіднього вузла, це означає, що знайдено коротший шлях, і мітка вузла оновлюється.

Після перевірки всіх сусідніх вузлів тимчасові мітки аналізуються по всьому графу, і вузол з найменшою міткою позначається як постійний і стає новим робочим вузлом [8].

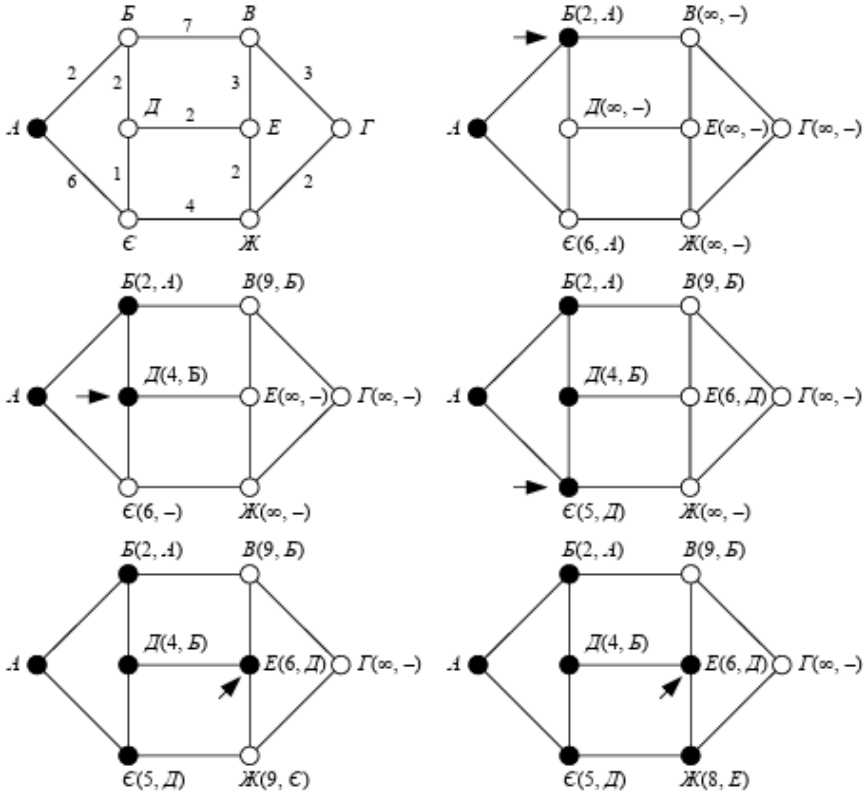


Рисунок 1.2 — Приклад знаходження найкоротшого маршруту від вузла А до вузла Г за використання алгоритму Дейкстри

Далі більш детально розглянемо найпоширеніші алгоритми маршрутизації:

— маршрутизація за вектором відстаней;

— маршрутизація з урахуванням стану лінії (каналу);

— гібридна маршрутизація.

Алгоритм маршрутизації за вектором відстаней (Distance Vector) використовується для визначення оптимальних маршрутів у мережі на основі інформації, що надходить від сусідніх маршрутизаторів. Кожен маршрутизатор підтримує таблицю, де для кожного віддаленого вузла вказана відстань до нього та наступний вузол на шляху. Маршрутизатори обмінюються цими таблицями зі

своїми сусідами, і після отримання нових даних вони оновлюють свої записи, перевіряючи, чи можна знайти коротший шлях через сусіда. Цей процес повторюється періодично або при зміні топології мережі. Основна перевага такого алгоритму полягає у простоті його реалізації, тоді як недоліком є відносно повільне поширення інформації про зміни та можливість виникнення проблем “лічильника до нескінченності” при розриві зв’язку.

Алгоритм маршрутизації з урахуванням стану лінії або каналу (Link State) працює на основі детальної інформації про топологію мережі. Кожен маршрутизатор повідомляє іншим про стан своїх безпосередніх з’єднань, включаючи наявність каналу, його пропускну здатність і затримки. На основі цих даних кожен маршрутизатор будує повну карту мережі та самостійно розраховує оптимальні шляхи до всіх вузлів, використовуючи алгоритми пошуку найкоротшого шляху, наприклад, алгоритм Дейкстри. Такий підхід дозволяє швидко реагувати на зміни топології та завантаженості каналів, забезпечує високу точність маршрутизації та стійкість мережі, проте потребує більшого обсягу пам’яті та обчислювальних ресурсів порівняно з алгоритмами за вектором відстаней.

Гібридний алгоритм маршрутизації поєднує принципи алгоритмів за вектором відстаней і з урахуванням стану каналу. У такому підході маршрутизатор не зберігає повної карти всієї мережі, як у випадку зі станом лінії, але й не обмежується лише простим обміном таблицями відстаней із сусідами. Замість цього він підтримує інформацію про топологію у своєму оточенні та періодично або за потреби обмінюється з іншими маршрутизаторами відомостями про стан з’єднань. Це дозволяє швидше знаходити зміни в мережі та ефективніше будувати маршрути, ніж у класичному методі за вектором відстаней, але з меншими витратами ресурсів, ніж у випадку повного контролю топології.

1.2.2 Проблеми конфігурування статичної та динамічної маршрутизації

Маршрутизатори отримують дані про віддалені мережі динамічно за

допомогою протоколів маршрутизації або вручну — за допомогою статичних маршрутів. У багатьох випадках маршрутизатори одночасно використовують протоколи динамічної маршрутизації і статичні маршрути. Статичні маршрути дуже поширені, при цьому вони не вимагають такої ж кількості обчислень і операцій, як протоколи динамічної маршрутизації [9].

Переваги статичної маршрутизації в порівнянні із динамічною маршрутизацією:

— статичні маршрути не оголошуються по мережі, таким чином, вони безпечніші;

— статичні маршрути використовують вужчу смугу пропускання, ніж протоколи динамічної маршрутизації;

— шлях, використовуваний статичним маршрутом для відправки даних, відомий.

Головними недоліками статичної маршрутизації є тривале початкове налаштування і подальше обслуговування, помилки при налаштуванні, особливо у великих мережах, необхідність втручання адміністратора для внесення змін в дані маршруту, а також недостатні можливості масштабування для зростаючих мереж, обслуговування при цьому стає досить трудомістким.

До типових проблем конфігурування статичної маршрутизації належать:

— помилки під час визначення мережевої адреси або маски підмережі;

— некоректне визначення вихідного інтерфейсу чи адреси наступного переходу;

— створення “routing loops” у разі некоректного дублювання маршрутів;

— відсутність резервних маршрутів, що призводить до повної втрати зв’язку при відмові каналу;

— складність підтримки актуальності таблиць у великих мережах із частими змінами топології;

— конфлікти між статичними і динамічними маршрутами (наприклад,

коли статичний маршрут має вищий пріоритет і перекриває динамічний).

Найбільш ефективними але і, мабуть, найскладнішими є способи динамічної (адаптивної) маршрутизації. При динамічний маршрутизації вміст таблиць маршрутів змінюється залежно від стану і завантаження каналів передачі даних і вузлів комутації. Для адаптації до зміни навантаження кожний вузол комутації повинен мати певну інформацію про стан мережі передачі даних і в першу чергу про її топологію, інтенсивність потоків даних і затримки (черги) у вузлах комутації. Ця інформація відстежується спеціальними керуючими пакетами, що ними обмінюються вузли комутації. Якість маршрутизації значною мірою залежить від оперативності відновлення керуючої інформації. Найбільш оптимальна маршрутизація досягається наявністю інформації про миттєвий стан мережі та її завантаження. Проте це, зазвичай призводить до значного збільшення потоку керуючих пакетів у мережі передачі даних і до зниження її ефективності.

Динамічна маршрутизація є досить складним процесом, що включає:

— формування маршрутів, яке здійснюється за допомогою алгоритмів маршрутизації шляхом упорядкування у кожному вузлі комутації таблиць маршрутів пакетів;

— реалізацію маршрутів, тобто керування пакетами при прямуванні їх підмережею зв’язку до місця призначення за допомогою спеціальних протоколів мережевого рівня;

— контроль стану мережі, у тому числі аналіз топології мережі, структури протоків і затримок у вузлах комутації;

— передачу інформації про стан мережі, яку використовують для коригування таблиць маршрутів.

Протоколи маршрутизації можуть працювати тільки з пакетами, які належать до одного з маршрутизованих протоколів. Протокол RIP (Routing Information Protocol) — один з перших протоколів внутрішньої маршрутизації, що застосовувалися в Інтернеті (1982 р.). Перша версія протоколу RIP описана в

специфікації RFC 1058, а друга (RIP ver. 2) — в специфікації RFC 2453. В цьому протоколі метрикою є кількість пересилань між вузлами (hops). Ця метрика не забезпечує облік пропускної здатності, надійності та завантаженості трактів передачі, а також характеристик трафіка користувачів.

RIP — це протокол дистанційно-векторної маршрутизації, що ґрунтується на використанні вектора відстаней (Distance Vector). Він не дозволяє забезпечити функціонування широкомасштабних мереж через обмеженість числа пересилань (hops) до 15. Для пошуку шляху використовується алгоритм Беллмана-Форда. Цей протокол застосовувався для перших маршрутизаторів Інтернету, побудованих на міні-ЕОМ типу Honeywell 516 (8-розрядні мікропроцесори типу Intel 8080 або Zilog Z80).

Робота протоколу RIP заснована на широкомовному розсиланні повідомлень про коректування маршрутів. Періодичність розсилання оновлень — 30 с. Розсилаються повні копії таблиць маршрутизації незалежно від того, змінені вони чи ні. RIP схильний до утворення петель у розрахованих маршрутах.

У RIPv2 допускається балансування навантаження на шляхах з рівною “довжиною” (вартістю).

Протокол IGRP (Interior Gateway Routing Protocol) і його розширена версія EIGRP (Enhanced Interior Gateway Routing Protocol) розроблені в корпорації Cisco. І якщо протокол IGRP — це дистанційно-векторний протокол, то EIGRP — гібридний протокол.

Протокол EIGRP розроблено на початку 1990-х рр. і є спробою поєднати переваги дистанційно-векторних протоколів і протоколів стану каналів. У ньому залишилися від IGRP без зміни максимальне число пересилань (225) і типи використовуваних метрик. Для усунення петель у маршрутах і оперативному реагуванні на зміни в мережі використовується алгоритм дифузійного оновлення DUAL (Distributed Update Algotithm). EIGRP не використовує періодичні оновлення про стан мережі, а використовує інкрементні оновлення (incremental updates).

Протокол OSPF (Open Shortest Path First) належить до класу протоколів

стану каналів (Link State Protocol). Дослівний переклад — першим обирається найкоротший шлях. “Open” — специфікація протоколу вільно поширюється, на відміну, наприклад, від специфікації протоколу EIGRP. RFC 2328 — основний діючий документ по OSPF.

Окрім того, протокол OSPF дозволяє визначити для будь-якої мережі значення метрики залежно від типу послуги TOS (Type of Service). В OSPF підтримуються метрики пропускної здатності та затримки. Метрика, що оцінює пропускну здатність каналу, визначається, наприклад, компанією Cisco, як кількість секунд, необхідних для передачі 100 Мбіт. Метрика затримки — час у мікросекундах, необхідний маршрутизатору для обробки, постановки в чергу та передачі пакетів. Для кожної з метрик протокол OSPF будує окрему таблицю маршрутизації. Стандартний порядок розрахунку метрики, що оцінює показники надійності, затримки й вартості, поки не визначений. Цей порядок визначається адміністратором.

У цьому протоколі також закладено можливість балансування навантаження на шляхах як з однаковою вартістю, так і з різною, розподіл трафіка відбувається пропорційно метриці шляху.

До типових проблем конфігурування динамічної маршрутизації належать:

— неправильне визначення межі зон або автономних систем;

— проблеми із класовою адресацією у RIPv1;

— розбіжності параметрів (несумісність таймерів Hello/Dead у OSPF, різні значення AS number у EIGRP);

— некоректна робота механізмів аутентифікації між маршрутизаторами;

— надлишкова кількість оголошених маршрутів, що може призвести до “routing table overflow”;

— помилки при налаштуванні протоколів пріоритетів і метрик;

— поява маршрутних петель (routing loops) через відсутність фільтрації або неправильні політики поширення маршрутів;

— збільшене навантаження на процесор маршрутизатора у великих мережах, спричинене постійними обчисленнями маршрутів і обміном службовими повідомленнями;

— конфлікти між різними протоколами маршрутизації у випадках багатопротокольного середовища [10-13].

В таблиці 1.2 наведено порівняльний огляд статичної та динамічної маршрутизації в комп'ютерних мережах та проблеми, які можуть виникати під час конфігурування (наприклад, в середовищі Cisco Packet Tracer).

Таблиця 1.2 — Маршрутизація та проблеми її конфігурування

Тип маршрутизації	Характеристика	Проблеми конфігурування
Статична	Проста, надійна, але вимагає постійного втручання адміністратора та не має достатніх можливостей для масштабування	Помилки у введенні мережевих адрес, масок підмереж, відсутність зворотніх маршрутів
RIP (динамічна)	Проста у налаштуванні, але застаріла	Проблеми із класовою адресацією у RIPv1
EIGRP (динамічна)	Поєднує переваги дистанційно-векторних протоколів та протоколів із урахуванням стану каналу	Невідповідність номерів автономних систем, проблеми із аутентифікацією
OSPF (динамічна)	Масштабована, універсальна	Невідповідність ідентифікаторів маршрутизаторів (Router ID), відсутність узгодження Hello/Dead таймерів, проблеми із аутентифікацією

1.3 Типові проблеми конфігурування популярних мережевих протоколів та технологій

1.3.1 Проблеми конфігурування віртуальних локальних мереж

Віртуальна локальна мережа (VLAN, Virtual Local Area Network) є

логічним сегментом мережі, який дозволяє об’єднувати пристрої з різних фізичних локальних мереж у єдину групу. У великих корпоративних мережах VLAN використовуються для оптимізації управління трафіком та підвищення продуктивності мережі. Правильне налаштування VLAN дозволяє групувати пристрої, які найчастіше обмінюються даними, і зменшувати навантаження на маршрутизатори, оскільки буде можливість перенаправлювати більшу частину трафіку через комутатори. На рисунку 1.3 зображена структурна схема загальної концепції віртуальних локальних мереж.

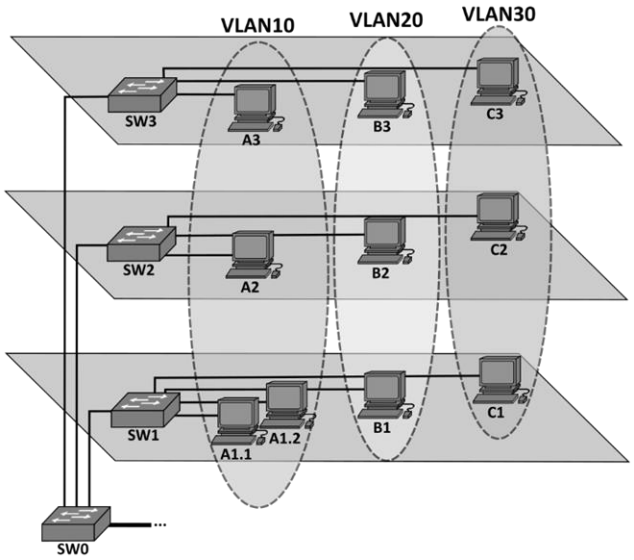


Рисунок 1.3 — Структурна схема концепції VLAN

Основні проблеми конфігурування VLAN включають:

— неправильне призначення портів до VLAN, що може ізолювати пристрої або перешкоджати їх взаємодії;

— некоректна маршрутизація між VLAN (Inter-VLAN routing), що ускладнює обмін даними між сегментами;

— відсутність узгодження конфігурації VLAN між комутаторами, що призводить до втрати зв’язку або некоректної передачі трафіку;

— неправильне налаштування trunk-портів, через що міжсегментний трафік не проходить через мережу [14].

1.3.2 Проблеми конфігурування адресних служб (DHCP, NAT)

Динамічний протокол конфігурації хостів (DHCP, Dynamic Host Configuration Protocol) є мережевим сервісом, що забезпечує автоматичний розподіл IP-адрес та параметрів мережевої конфігурації між клієнтськими пристроями. Завдяки цьому адміністратор звільняється від необхідності самостійного введення параметрів, що є важливим у великих корпоративних мережах. DHCP підтримує динамічне призначення адрес, їхнє повторне використання після завершення оренди, а також може надавати додаткові параметри, зокрема адреси шлюзу за замовчуванням, DNS-серверів та маски підмережі. На рисунку 1.4 зображено структурну схему роботи DHCP [15].

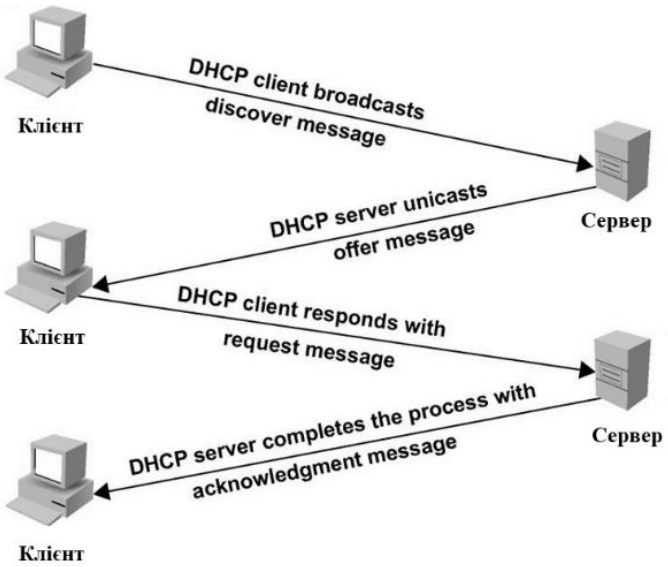


Рисунок 1.4 — Структурна схема роботи DHCP

Трансляція мережевих адрес (NAT, Network Address Translation) — це технологія, яка дозволяє декільком пристроям у приватній мережі спільно використовувати одну публічну IP-адресу для доступу до Інтернету. Використання NAT забезпечує економію адресного простору IPv4, підвищує рівень безпеки мережі шляхом приховування внутрішньої структури та дозволяє кільком клієнтам одночасно виходити до глобальної мережі через одну публічну адресу. На рисунку 1.5 зображено структурну схему роботи NAT [16].

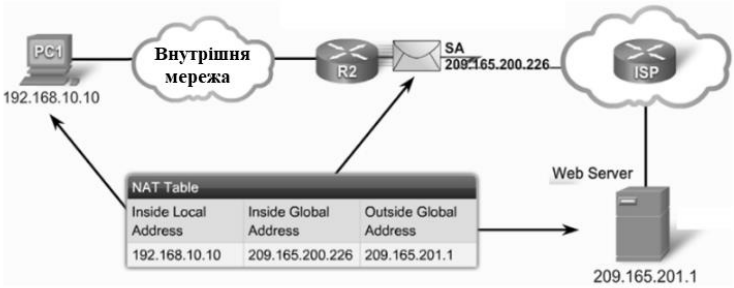


Рисунок 1.5 — Структурна схема роботи NAT

Попри значну користь, під час налаштування DHCP і NAT можуть виникати певні труднощі. Для DHCP типовими труднощами є неправильне визначення діапазону адрес, що може призвести до їх накладання чи конфлікту з уже статично призначеними IP-адресами. Крім того, помилки в конфігурації шлюзу або DNS-серверів здатні спричинити недоступність зовнішніх ресурсів для клієнтів. У великих мережах можливими є проблеми з часом оренди адрес: занадто короткий період створює надмірне навантаження на сервер, а занадто довгий може зумовити нестачу вільних адрес.

При конфігуруванні NAT поширеними є труднощі з коректною маршрутизацією трафіку, особливо за використання статичної трансляції для серверів у внутрішній мережі. Також можливими є конфлікти між правилами трансляції, які можуть спричинити блокування доступу до окремих сервісів. Використання PAT (Port Address Translation) може обмежувати функціональність деяких додатків, що потребують відкритих портів або працюють з протоколами, які несумісні з трансляцією.

1.3.3 Проблеми конфігурування технологій глобальних мереж (на прикладі Frame Relay)

Технології глобальних мереж (WAN) забезпечують взаємодію локальних сегментів на великих відстанях та є основою корпоративних і міжмережевих комунікацій. Однією з класичних технологій передавання даних у глобальних мережах є Frame Relay, яка була розроблена як ефективний метод

мультиплексування каналів із використанням віртуальних з'єднань. Frame Relay дозволяє організовувати логічні канали між різними вузлами без необхідності створення фізичного каналу для кожної пари абонентів, що забезпечує оптимальне використання пропускної здатності.

Фізично мережа Frame Relay складається із сукупності комунікаційних вузлів, зв'язаних фізичними каналами, та пристроїв доступу. Структурна схема мережі Frame Relay зображена на рисунку 1.6. В мережі FR розрізняють два типи пристроїв:

— термінальні пристрої DTE (Data Terminal Equipment), які є зовнішніми модулями для доступу до мережі, наприклад, робочі станції, термінали, мультиплексори, маршрутизатори, мости тощо;

— мережні пристрої DCE (Data Circuit-terminating Equipment), які є комунікаційними вузлами у складі мережі, що призначені для синхронізації та формування каналу між кінцевими взаємодійними пристроями; зазвичай, це комутатори кадрів [17].

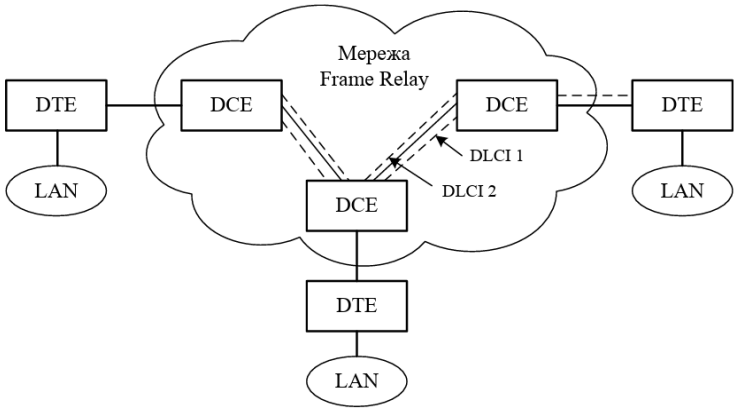


Рисунок 1.6 — Структурна схема мережі Frame Relay

Попри функціональні переваги, конфігурування технологій WAN, зокрема Frame Relay, пов'язане з низкою проблем. По-перше, складність налаштування виникає через необхідність правильної ідентифікації DLCI (Data Link Connection Identifier), які відповідають за відображення логічних каналів. Невірне призначення DLCI призводить до втрати доступності вузлів або неправильного

маршрутизаційного відображення.

По-друге, Frame Relay передбачає використання різних режимів інкапсуляції (Cisco чи IETF). Якщо обидва кінці з'єднання налаштовані з різними параметрами інкапсуляції, встановлення віртуального каналу стає неможливим. Це створює типову проблему сумісності в гетерогенних мережах.

По-третє, у процесі конфігурування можуть виникати труднощі з підтримкою протоколів маршрутизації поверх Frame Relay. Наприклад, у випадку використання багатоточкових з'єднань можливі проблеми з доставкою широкомовного або мультикаст-трафіку, що безпосередньо впливає на коректність роботи протоколів динамічної маршрутизації (OSPF, EIGRP).

Крім того, проблеми можуть виникати при невірному налаштуванні параметрів комутованої інфраструктури провайдера, оскільки від неї залежить надання гарантованої пропускної здатності (CIR — Committed Information Rate). У разі некоректного планування смуги пропускання мережа може зазнавати затримок, втрат кадрів або перевантаження [18].

1.4 Традиційні методи пошуку несправностей у мережах

Розглянемо класичні інструменти діагностики несправностей у комп'ютерних мережах, що дозволяють оцінити доступність вузлів, визначити маршрут проходження мережевого трафіку, а також здійснити аналіз роботи мережевого обладнання. До таких інструментів належать команди "ping", "traceroute" (для Cisco IOS), "debug", механізми логування та системи моніторингу, зокрема використанням протоколу SNMP.

Команда "ping" (Packet Internet Groper) є базовим інструментом для перевірки доступності вузла в мережі на основі протоколу ICMP (Internet Control Message Protocol). Вона надсилає ICMP Echo Request до заданої IP-адреси та очікує ICMP Echo Reply. Основні завдання команди "ping":

— перевірка наявності мережевого з'єднання між двома вузлами;

— вимірювання часу відгуку (RTT — Round Trip Time);

— визначення відсотку втрати пакетів;

— оцінка стабільності каналу.

Команда “traceroute” використовується для визначення маршруту, яким проходять пакети від джерела до призначення. В її основі лежить використання поля TTL (Time To Live) у заголовку IP-пакета. Traceroute поступово збільшує значення TTL, отримуючи ICMP-повідомлення від кожного проміжного маршрутизатора. Це дозволяє відобразити список вузлів, через які проходить трафік. Основні завдання команди “traceroute”:

— локалізація вузла, де виникає затримка або обрив;

— виявлення маршрутів у випадку асиметричної маршрутизації;

— аналіз продуктивності каналів у глобальних мережах [19].

Команда “debug” у Cisco IOS призначена для детальної діагностики процесів, що відбуваються на мережевому обладнанні. Вона дозволяє в режимі реального часу відстежувати роботу протоколів маршрутизації, процеси встановлення з’єднань, обробку пакетів та інші події. Особливості використання “debug”:

— вивід інформації відбувається безпосередньо в консольний сеанс;

— може створювати значне навантаження на процесор маршрутизатора або комутатора, особливо у великих мережах;

— зазвичай застосовується для точкового аналізу конкретних протоколів чи інтерфейсів;

— після закінчення діагностики обов’язково слід вимкнути відлагодження командою “undebug all” або “u all” [20].

На рисунку 1.7 зображено фрагмент схеми комп’ютерної мережі (із використанням адрес IPv4 та маршрутизацією EIGRP) в середовищі Cisco Packet Tracer, для якої буде проведено перевірку працездатності за допомогою команд “ping” та “traceroute”.

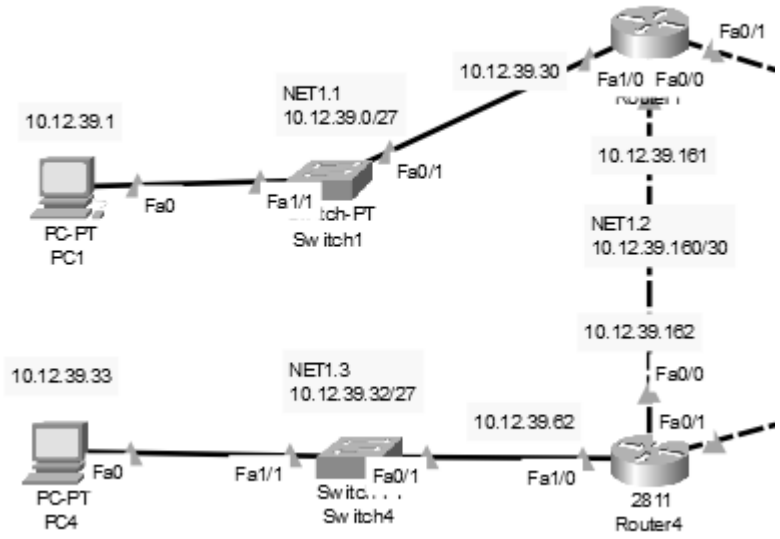


Рисунок 1.7 — Фрагмент схеми комп’ютерної мережі

Далі на маршрутизаторі Router1 у привілейованому режимі послідовно дамо команди “ping” та “traceroute”, як показано на рисунку 1.8.

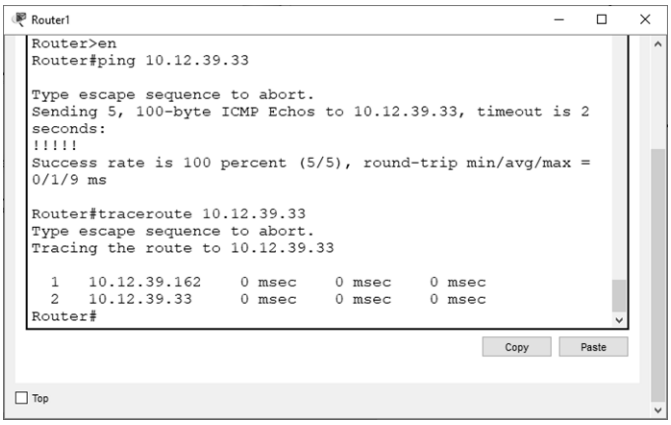


Рисунок 1.8 — Результат роботи команд “ping” та “traceroute”

Далі на маршрутизаторі Router1 у привілейованому режимі дамо команду “debug eigrp packets”, яка дозволить відслідковувати всі EIGRP-пакети, що проходять через інтерфейси маршрутизатора, включаючи Hello, Update, Query, Reply, ACK. Вона корисна для глибокої діагностики роботи протоколу та виявлення проблем із встановленням сусідських відносин чи обміном маршрутами. Результат роботи цієї команди показано на рисунку 1.9.

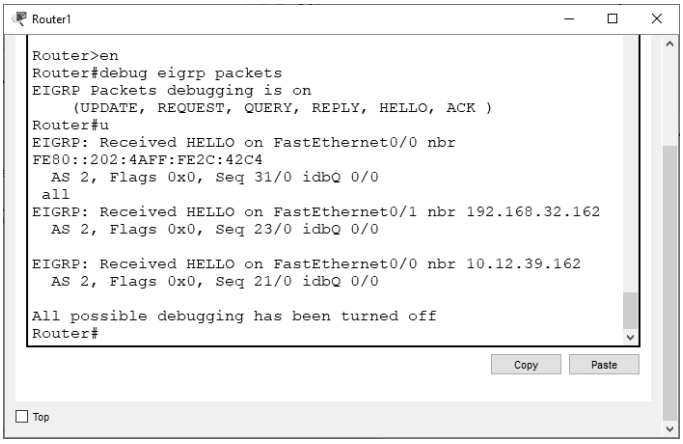


Рисунок 1.9 — Результат роботи команди “debug eigrp packets”

Розглянемо протокол SNMP, який є протоколом управління мережею і дозволяє централізовано відстежувати стан мережевого обладнання та кінцевих пристроїв, збирати статистику і реагувати на події. SNMP широко використовується для моніторингу комутаторів, маршрутизаторів, серверів та інших мережевих елементів. Основними компонентами SNMP є: Manager, Agent, MIB (Management Information Base).

Manager — програмне забезпечення, що запитує інформацію та керує мережевими пристроями.

Agent — програмний модуль на пристрої, який збирає інформацію та передає її менеджеру.

MIB — база даних, що містить об’єкти, доступні для моніторингу та управління.

До переваг SNMP можна віднести централізоване управління великою кількістю пристроїв; збір статистики по трафіку, завантаженості інтерфейсів, працездатності сервісів; можливість налаштування сповіщень (traps) у разі критичних подій, наприклад, відмови інтерфейсу або перевантаження процесора.

До проблем при використанні SNMP можна віднести неправильне налаштування версії протоколу (SNMPv1, v2c, v3) та ключів доступу, що може блокувати доступ менеджера до агентів; обмеження на кількість запитів або частоту опитувань, що може створювати навантаження на мережу; відсутність узгодження MIB-об’єктів між пристроями та програмним забезпеченням

менеджера [21].

Розглянемо схему комп’ютерної мережі, що зображена на рисунку 1.10. Вона призначена для демонстрації практичного використання протоколу SNMP для моніторингу та управління мережевими пристроями.

Рисунок 1.10 — Схема мережі для демонстрації роботи протоколу SNMP

На схемі зображено мережу з п’яти пристроїв:

— PC-PT (MIB_Browser) — клієнт для роботи з SNMP, на якому запущено програму MIB Browser;

— SNMP_Router — маршрутизатор, налаштований для обслуговування SNMP-запитів;

— Router1 — проміжний маршрутизатор;

— Server0 — сервер, підключений до мережі.

SNMP_Router налаштований на дві спільноти (community strings):

— “testro” — доступ тільки для читання (Read-Only);

— “testrw” — доступ для читання та запису (Read-Write).

Це дозволяє контролювати, хто може просто отримувати дані пристрою і хто може змінювати конфігураційні параметри через SNMP.

Налаштуємо MIB Browser для доступу до маршрутизатора, використовуючи IP-адресу: 10.0.0.1, стандартний SNMP-порт: 161 та визначені параметри автентифікації: спільнота для читання “testro”, спільнота для запису “testrw” і протокол SNMP версії v3. Ці налаштування показані на рисунку 1.11. Основна мета полягає у перевірці можливості встановлення з’єднання між MIB Browser та маршрутизатором із коректним розмежуванням прав доступу.

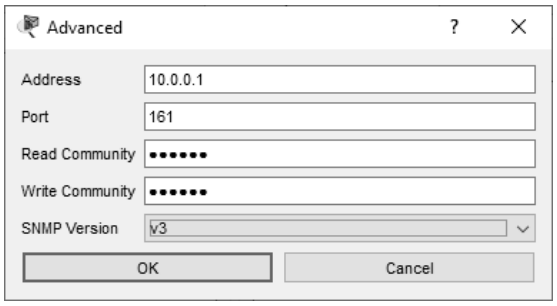


Рисунок 1.11 — Налаштування MIB Browser

Після встановлення з’єднання розгортаємо дерево MIB та вибираємо об’єкт .sysDescr, що відповідає за системний опис пристрою. Виконання операції Get дозволяє надіслати SNMP-запит і отримати у відповідь рядок з інформацією про маршрутизатор, зокрема версію IOS та загальний опис обладнання. На рисунку 1.12 наведено результат надсилання SNMP-запиту.

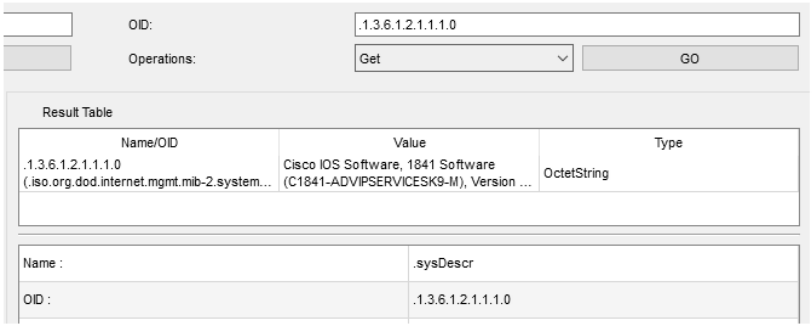


Рисунок 1.12 — Результат надсилання SNMP-запиту

Далі виконуємо операцію Set для об’єкта .sysContact. У полі значення вводимо новий рядок типу OctetString, наприклад, ім’я адміністратора “ADMIN”, що зображено на рисунку 1.13.

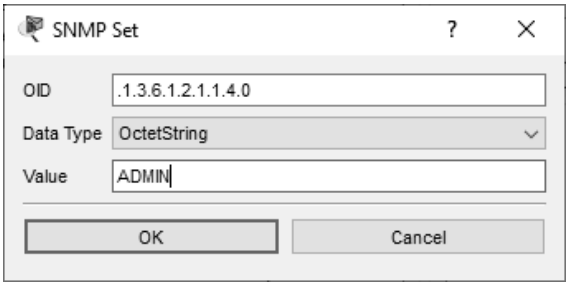


Рисунок 1.13 — Встановлення значення типу OctetString

Якщо доступ здійснюється через спільноту “testrw”, SNMP-запит на зміну має бути успішно застосований до конфігурації маршрутизатора. Для перевірки результату знову використовуємо операцію Get для об’єкта .sysContact. У випадку коректної роботи SNMP-механізму MIB Browser відобразить оновлене значення, що підтверджує можливість як читання, так і модифікації параметрів пристрою через SNMP. Результати коректної роботи протоколу наведено на рисунку 1.14.

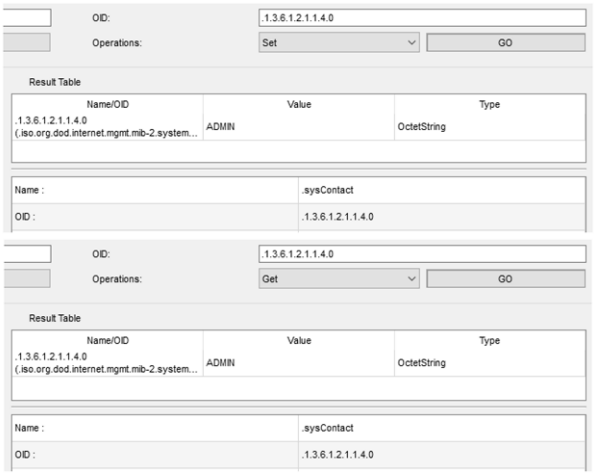


Рисунок 1.14 — Результат коректної роботи SNMP

Отже, SNMP забезпечує централізований механізм доступу до конфігураційних даних маршрутизатора, дозволяє як отримувати системну інформацію, так і змінювати параметри обладнання безпосередньо через MIB. Рівень доступу визначається налаштованими спільнотами, де спільнота для читання обмежує роботу лише моніторингом, а спільнота для запису надає можливість модифікації параметрів. Використання SNMP v3 гарантує захищений обмін із автентифікацією та контролем прав користувачів, що наближає роботу до вимог реальних корпоративних мереж. У середовищі Cisco Packet Tracer цей механізм можна відпрацювати, перевіряючи як операції читання, так і зміни значень у MIB, що дозволяє імітувати практичні сценарії адміністрування. Коректна конфігурація MIB Browser разом із налаштуванням спільнот забезпечує можливість локалізувати проблеми й змінювати параметри

пристрою віддалено, без необхідності прямого доступу до CLI маршрутизатора.

1.5 Сучасні тенденції автоматизації траблшутінгу

Сучасні тенденції у сфері автоматизації траблшутінгу тісно пов’язані з розвитком систем централізованого моніторингу, які дозволяють не лише відслідковувати працездатність мережевої інфраструктури, але й автоматизувати процеси виявлення несправностей. Серед найбільш поширених інструментів можна виокремити Zabbix, Nagios, PRTG Network Monitor та Wireshark.

Zabbix — це потужна система моніторингу з відкритим кодом, яка підтримує як серверну, так і агентську архітектуру. Основними її компонентами є Zabbix Server (ядро, що збирає дані та обробляє тригери), агенти, що встановлюються на вузли, а також веб-інтерфейс для адміністрування. Zabbix підтримує SNMP, IPMI, JMX та інші протоколи, що дозволяє здійснювати моніторинг мережевого обладнання, серверів і прикладних сервісів. Завдяки тригерам адміністратор може налаштовувати автоматичні реакції на інциденти, наприклад, відправку повідомлень або запуск скриптів для виправлення ситуації. Система також має вбудовані засоби візуалізації: дашборди, графіки та карти мережі, що спрощує аналіз несправностей у реальному часі. Приклад інтерфейсу системи мережевого моніторингу Zabbix наведено на рисунку 1.15.

Рисунок 1.15 — Приклад інтерфейсу системи Zabbix

Nagios — це одна з найстаріших систем моніторингу, яка побудована на модульній архітектурі з використанням плагінів. Основний компонент — Nagios

Core, який відповідає за обробку конфігурацій та управління перевірками. Плагіни забезпечують перевірку різних служб і протоколів: від доступності хоста до роботи веб-сервера чи бази даних. Система дозволяє налаштовувати оповіщення з багатоступеневою ескалацією інцидентів, що робить її корисною для оперативного реагування. Проте для більш сучасних можливостей візуалізації чи масштабованості потрібні додаткові модулі, такі як Nagios XI або сторонні веб-надбудови. Nagios більше орієнтований на точковий контроль і традиційний підхід до адміністрування. Приклад інтерфейсу системи мережевого моніторингу Nagios наведено на рисунку 1.16.

Рисунок 1.16 — Приклад інтерфейсу системи Nagios

PRTG Network Monitor — це комерційна система моніторингу від Paessler AG, яка вирізняється простотою налаштування і широким набором вбудованих сенсорів. Сенсор у PRTG — це об’єкт, який контролює один параметр, наприклад, пропускну здатність інтерфейсу, доступність порту чи завантаження процесора. Архітектурно PRTG складається з ядра серверу та проб (probes), які здійснюють безпосередній збір даних. Система має зручний веб-інтерфейс і мобільні застосунки, підтримує SNMP, WMI, NetFlow, sFlow і API-запити. Вбудовані карти та звіти дозволяють створювати візуальні схеми інфраструктури. Особливістю є також автоматичне виявлення пристроїв і мереж, що знижує трудовитрати при розгортанні. Приклад інтерфейсу системи мережевого моніторингу PRTG Network Monitor наведено на рисунку 1.17.

Рисунок 1.17 — Приклад інтерфейсу системи PRTG

Wireshark — це не система централізованого моніторингу, а аналізатор мережевого трафіку. Він функціонує на рівні захоплення пакетів і дозволяє вивчати деталі роботи протоколів. Основним компонентом є графічний інтерфейс із фільтрами, які дозволяють виділяти потрібні пакети, наприклад тільки TCP або DNS-запити. Wireshark підтримує сотні протоколів і може декодувати структуру пакетів аж до полів заголовків. У практиці траблшутінгу він використовується для глибокого аналізу проблем — наприклад, визначення причин затримок, помилок у TCP-сесіях чи некоректної роботи служб. У поєднанні з іншими системами моніторингу Wireshark виступає інструментом деталізації інцидентів, коли необхідно вийти за межі загальних показників доступності чи навантаження. Приклад системи аналізу мережевого трафіку Wireshark наведено на рисунку 1.18.

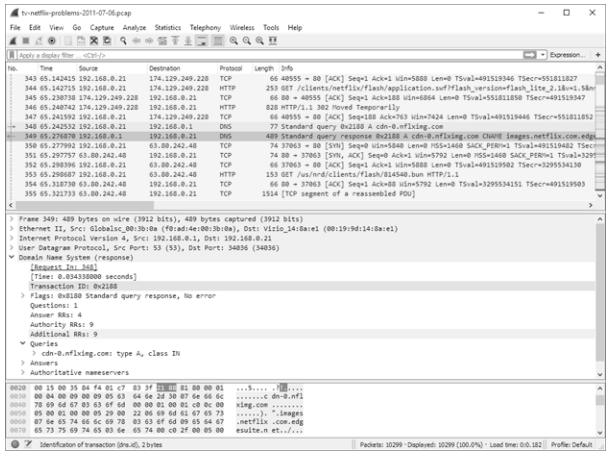


Рисунок 1.18 — Приклад інтерфейсу системи Wireshark

Отже, Zabbix та PRTG представляють собою комплексні рішення для централізованого моніторингу із розвиненими можливостями автоматизації та візуалізації. Nagios більше орієнтований на базовий контроль із використанням плагінів та ручного налаштування, тоді як Wireshark надає глибокий аналіз трафіку на пакетному рівні. Разом ці інструменти можуть використовуватися як багаторівнева система для автоматизованого виявлення та усунення несправностей. На основі проведеного аналізу сучасних тенденцій у сфері автоматизації траблшутінгу, було створено порівняльну характеристику, результати якої було занесено до таблиці 1.3.

Таблиця 1.3 — Порівняльна характеристика відомих систем для автоматизації траблшутінгу

Система / Характеристика	Zabbix	Nagios	PRTG	Wireshark
Централізований моніторинг	+	+	+	—
Підтримка SNMP	+	+	+	+
Підтримка NetFlow/sFlow	+	—	+	+
Вбудовані засоби візуалізації	+	—	+	+
Автоматичне виявлення пристроїв	+	—	+	—
Модульність/плагіни	+	+	—	—
Глибокий аналіз пакетів	—	—	—	+
Безкоштовність/відкритий код	+	+	—	+

Отже, Zabbix і PRTG є найбільш комплексними рішеннями для централізованого моніторингу, Nagios підходить для базового контролю з плагінами, а Wireshark орієнтований на низькорівневий аналіз мережевого трафіку [22].

2 ВЕЛИКІ МОВНІ МОДЕЛІ, ЯК ІНСТРУМЕНТ ДЛЯ ТРАБЛШУТІНГУ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

2.1 Архітектура і принципи роботи великих мовних моделей (LLM)

У сучасних інформаційних системах великі мовні моделі (LLM, Large Language Models) стали ключовим інструментом для автоматизації обробки природної мови. Вони відкрили нові можливості у сфері аналізу тексту, генерації контенту, підтримки користувачів та інтелектуального пошуку. На відміну від класичних підходів у галузі обробки природної мови (NLP), які базувалися на статистичних методах або відносно невеликих нейронних мережах, LLM відзначаються масштабністю архітектури та здатністю до узагальнення на основі колосальних обсягів даних. Їхня поява стала можливою завдяки розвитку обчислювальної інфраструктури, алгоритмів глибинного навчання та доступності великих корпусів текстової інформації.

LLM — це штучні нейронні мережі, спеціально спроєктовані для роботи з текстовими даними у природній мові. Вони здатні прогнозувати наступне слово чи символ у послідовності, що дозволяє їм генерувати цілісний зв'язний текст, відповідати на запитання, виконувати переклад та інші завдання. Відмінністю LLM від традиційних моделей NLP є їхній масштаб: якщо класичні мовні моделі оперували тисячами чи мільйонами параметрів, то сучасні LLM містять мільярди, а іноді й трильйони параметрів, що забезпечує значно вищу якість відтворення мовних закономірностей. Крім того, LLM навчаються на багатомовних та тематично різнорідних корпусах, що включають терабайти даних із книг, наукових статей, веб-ресурсів та програмного коду. Завдяки цьому вони демонструють здатність до узагальнення знань і можуть ефективно застосовуватися у завданнях, які раніше вимагали окремих спеціалізованих моделей.

Основою сучасних великих мовних моделей є архітектура трансформера. Ключовим компонентом трансформера є механізм self-attention, який дозволяє моделі враховувати контекст усього вхідного речення при обробці кожного

токена. Математично робота механізму уваги описується формулою (2.1) [23].

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2.1)$$

де Q(query), K(key), V(value) — матриці, отримані з вхідних векторів токенів; d_k — розмірність ключів.

На рисунку 2.1 наведена схема архітектури трансформера, яка відображає блоки Multi-Head Attention, нормалізацію та Feed Forward мережі, організовані у шари [24].

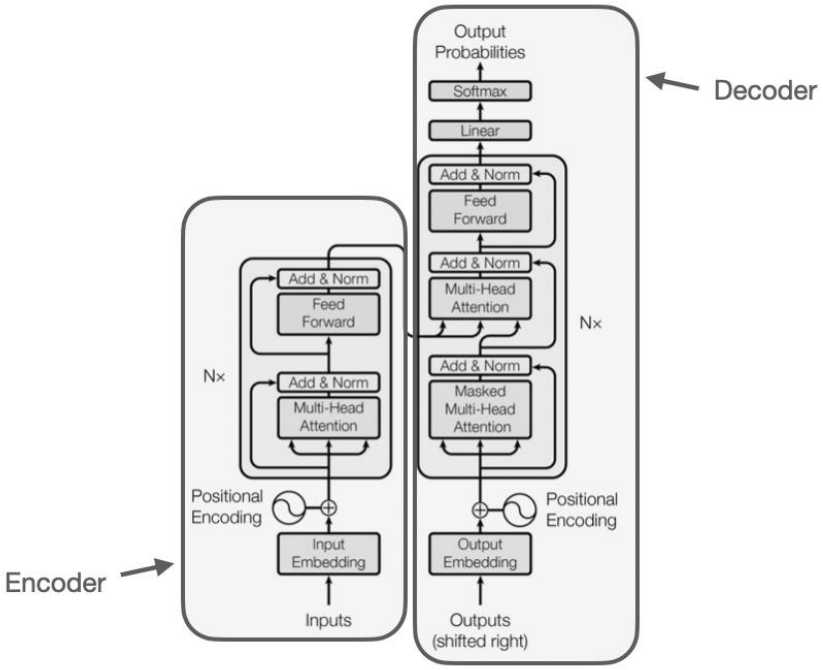


Рисунок 2.1 — Схема архітектури трансформера

Процес функціонування LLM можна описати як поетапну обробку текстових даних. Спочатку вхідний текст проходить токенізацію, після чого кожен токен перетворюється на вектор у багатовимірному просторі за допомогою embeddings. Далі вектори послідовно проходять через багатошарові блоки трансформера, де на кожному кроці застосовується self-attention та нелінійні перетворення.

Вихід моделі визначається як ймовірнісний розподіл наступного токена у послідовності. Для навчання застосовується функція крос-ентропійної втрати, що наведена у формулі (2.2) [25].

$$L = -\sum_{i=1}^{N}(y_i \, log \, \widehat{y_i}), \tag{2.2}$$

де y_i — істинний розподіл (one-hot);

$\widehat{y_i}$ — прогнозована ймовірність моделі для кожного токена.

Навчання LLM потребує потужних обчислювальних ресурсів: сучасні моделі тренуються на GPU або TPU-кластерах із тисячами ядер одночасно. Для розподіленого навчання використовують техніки data parallelism (розподіл даних) та model parallelism (розподіл моделі між пристроями).

Математично оновлення параметрів у розподіленому середовищі можна виразити формулою (2.3) [26].

$$\theta^{t+1} = \theta^t - \eta \frac{1}{K}\sum_{k=1}^{K} \nabla_\theta L_k, \tag{2.3}$$

де θ — вектор параметрів моделі, тобто всі ваги та зсуви нейронної мережі;

t — ітерація (крок) оптимізації, тобто поточний час або номер оновлення параметрів;

η — це швидкість навчання (learning rate), скаляр, що визначає, наскільки змінюються параметри на кожному кроці;

K — кількість обчислювальних пристроїв, що беруть участь у розподіленому навчанні;

$\nabla_\theta L_k$ — градієнт втрат по параметрах на k-му пристрої.

На рисунку 2.2 наведена схема pipeline тренування LLM, яка включає: текстові дані, токенізацію, функцію втрат, блоки трансформера, оновлення параметрів через оптимізацію, розподілене навчання на GPU/TPU [27].

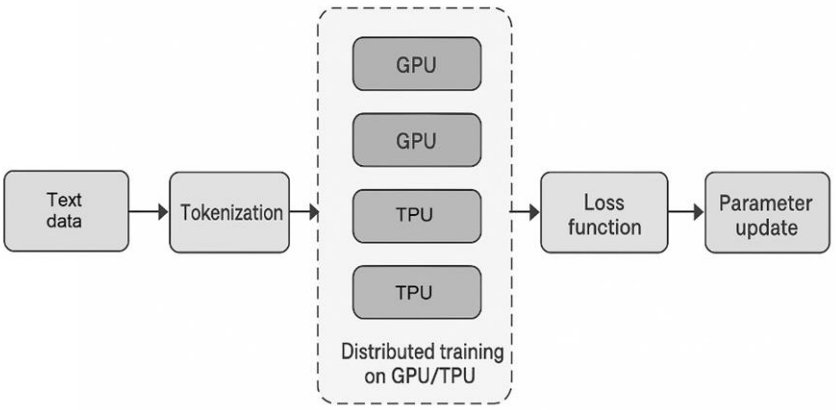


Рисунок 2.2 — Схема pipeline тренування LLM

2.2 Огляд можливостей існуючих LLM у сфері мережевих технологій

У контексті траблшутінгу в комп’ютерних мережах доцільно зосередитись на кількох найбільш популярних великих мовних моделях, які мають потенціал стати ефективними інструментами для автоматизації пошуку та усунення несправностей. Для аналітичного огляду було обрано GPT-5, Gemini, Claude, LLaMA та Mistral, оскільки саме ці системи репрезентують різні підходи до архітектури та використання в практичних середовищах. Подальші експериментальні дослідження будуть спрямовані на оцінку їхніх можливостей у вирішенні конкретних завдань, пов’язаних із діагностикою та конфігуруванням мережевої інфраструктури [28].

Варто чітко розмежовувати LLM, як мовні моделі та прикладні системи, що створені на її основі. LLM — це, власне, модель (наприклад, GPT-5, Gemini, Claude, LLaMA, Mistral), яка містить архітектуру трансформера, параметри, навчальні алгоритми та корпус даних. Вона є “рушієм” обробки тексту. Прикладні системи (інтерфейси) — це чат-боти, віртуальні асистенти чи бізнес-рішення, що використовують LLM у своїй роботі. У контексті траблшутінгу в комп’ютерних мережах безпосередня взаємодія відбуватиметься не із LLM, як математичними конструкціями, а із прикладними системами, які реалізують доступ до їхніх можливостей. Прикладами таких систем є: ChatGPT, Google Gemini, Claude.ai, Perplexity AI, Le Chat. Це пояснюється тим, що великі мовні

моделі існують у вигляді високорівневих архітектур з мільярдами параметрів, але без спеціалізованих інтерфейсів вони не можуть бути ефективно використані для вирішення прикладних задач. Чат-боти та асистенти, що працюють на основі LLM, виступають у ролі прикладного шару. Вони реалізують механізми авторизації, мережевої взаємодії, зручного користувацького інтерфейсу, а також забезпечують інтеграцію з інструментами, які адміністратор реально застосовує для траблшутінгу. Тобто, інженер комунікує, наприклад із ChatGPT, як із системою, що вміє інтерпретувати запит, перетворювати його у відповідний формат для LLM і повертати результат у зрозумілому вигляді [29].

GPT-5 є найновішою ітерацією мовної моделі компанії OpenAI, яка розвиває архітектуру трансформерів із суттєвим масштабуванням параметрів і вдосконаленими механізмами оптимізації. Модель працює на корпусах даних, що охоплюють як загальну інформацію, так і спеціалізовані домени, включно з технічною документацією, кодом та матеріалами з комп'ютерних мереж. Завдяки цьому GPT-5 здатна опрацьовувати, як природномовні запити, так і запити, що містять конфігураційні команди, фрагменти конфігураційних файлів чи повідомлення системного журналу. На рисунку 2.3 зображено еволюцію усіх GPT-моделей від попередніх поколінь до сучасної реалізації GPT-5.

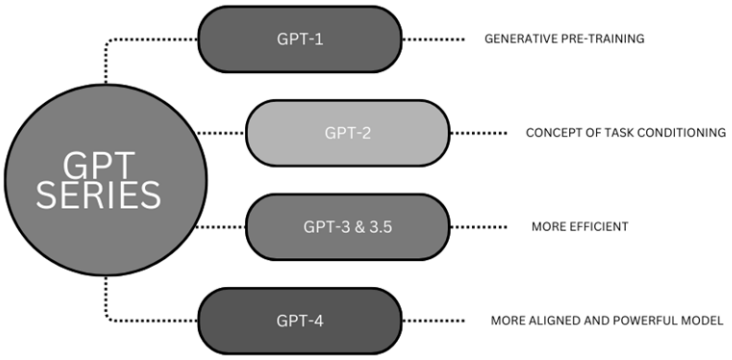


Рисунок 2.3 — Еволюція GPT-моделей

У контексті траблшутінгу в комп'ютерних мережах GPT-5 може аналізувати мережеві топології, знаходити потенційні помилки у конфігураційних файлах маршрутизаторів та комутаторів та пропонувати

варіанти їх усунення. Ключовою характеристикою є здатність інтегрувати знання про комп'ютерні мережі і застосовувати їх у вигляді покрокових інструкцій. Модель GPT-5 доступна через інтерфейс ChatGPT та API OpenAI. У реальних сценаріях адміністратор може використовувати GPT-5 для швидкої перевірки коректності команд в операційній системі Cisco IOS, генерації конфігурацій для певних топологій, пояснення логів із SNMP чи syslog тощо [30].

Gemini — це сімейство великих мовних моделей, розроблене Google DeepMind, яке поєднує класичну архітектуру трансформерів із вдосконаленою багатомодальністю (MoE, Mixture-of-Expert). MoE — це нова парадигма машинного навчання, яка передбачає поєднання прогнозів кількох спеціалізованих моделей, відомих як "експерти", для отримання остаточного прогнозу. На рисунку 2.4 наведено діаграму високого рівня MoE, яка ілюструє, як вхідні дані проходять через експертів [31].

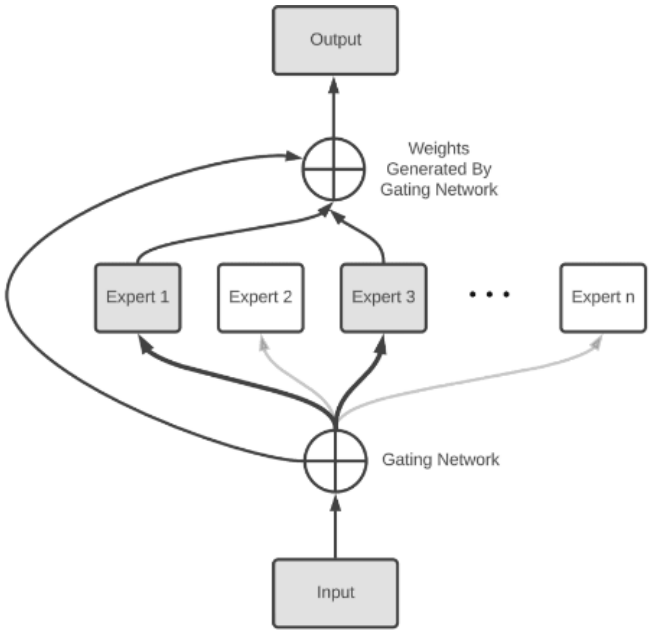


Рисунок 2.4 — Діаграма MoE

Gemini спроєктований для нативної роботи з різними типами даних — текстовими логами, кодом, мережевими схемами, графіками та навіть відеопотоками. Завдяки цьому він може аналізувати як текстові описи топологій, так і візуальні зображення схем у середовищах моделювання. Практична

реалізація Gemini здійснюється через платформу Google Gemini, яка є прикладним інтерфейсом до моделі.

У задачах траблшутінгу Gemini є корисним для багатоканального аналізу — він може одночасно обробляти текстові конфігурації та результати моніторингу. Це робить його потужним інструментом у сценаріях, коли потрібно комплексно оцінити роботу мережевої інфраструктури та допомогти виявити першопричини збоїв та проблем конфігурування.

Claude — це лінійка великих мовних моделей, розроблена компанією Anthropic, яка суттєво відрізняється від інших сучасних LLM своєю архітектурною концепцією, орієнтованою на безпечність та керованість. Якщо GPT та Gemini роблять акцент на масштабуванні параметрів і багатомодальності, то Claude з самого початку проєктувався як модель із вбудованими механізмами “constitutional AI” [32].

З точки зору архітектури, Claude реалізує класичний трансформерний стек, проте має модифіковані механізми уваги, оптимізовані для надзвичайно довгих контекстів — до сотень тисяч токенів. Це важлива перевага для траблшутінгу в комп’ютерних мережах, де необхідно аналізувати великі обсяги конфігураційних файлів, логи з численних пристроїв тощо

На рисунку 2.5 зображено ілюстрацію підходу, який був застосований у моделях Claude. Він показує, як різні методи навчання впливають на баланс між корисністю (helpfulness) та безпечністю/нешкідливістю (harmlessness) моделей.

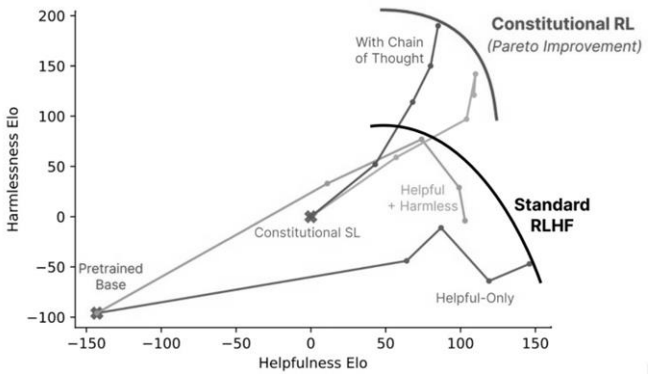


Рисунок 2.5 — Ілюстрація підходу “constitutional AI”

У вигляді прикладного інтерфейсу Claude доступний через чат Claude.ai. Архітектурні особливості, такі як контрольована поведінка та наддовгий контекст, роблять цю модель важливою для тих завдань, де потрібно балансувати між аналітикою, точністю та відповідальністю у прийнятті рішень.

LLaMA (Large Language Model Meta AI) від Meta — це серія відкритих LLM, розроблених із прицілом на академічні дослідження й оптимізацію продуктивності. Її архітектура є варіацією трансформера, але з низкою оптимізацій, які відрізняються від вже розглянутих LLM.

Ключова відмінність полягає у використанні Grouped Query Attention (GQA) та Multi-Head Attention з оптимізованим розподілом ключів, що зменшує вимоги до пам’яті під час інференсу. Це робить LLaMA більш ефективною у сценаріях, де потрібні швидкі відповіді на менш потужних GPU. На рисунку 2.6 наведено схему, яка демонструє архітектурний принцип GQA [33].

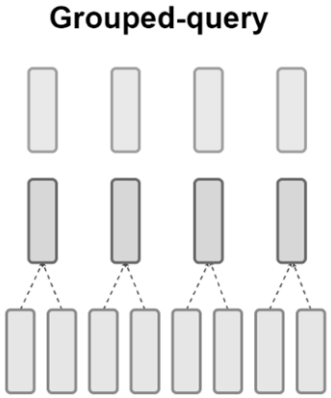


Рисунок 2.6 — Принцип GQA

Другою суттєвою рисою є обмеження параметрів контекстного вікна: початкові версії LLaMA працювали з вікном 2048 токенів, пізніші моделі (LLaMA 2, LLaMA 3) значно його розширили, проте залишаються орієнтованими на відносно вузькі завдання порівняно з GPT-5 чи Gemini, які від початку проєктувались, як мультимодальні універсали. Це означає, що LLaMA краще підходить для досліджень та fine-tuning у вузьких доменах, зокрема для траблшутінгу в комп’ютерних мережах.

Щоб отримати доступ до LLaMA, потрібен прикладний інтерфейс, який

реалізує діалог із користувачем і підключається до самої моделі. Наприклад, Perplexity AI (частково) інтегрує LLaMA як одну з моделей для генерації відповідей.

Mistral — це сімейство відкритих мовних моделей, яке з'явилося у 2023 році й відразу заявило про себе, як про архітектуру, сфокусовану на максимально ефективному інференсі та масштабованості в умовах обмежених ресурсів. На відміну від розглянутих LLM, Mistral від початку проєктувався для високошвидкісної роботи з довгими контекстами і модульного використання.

Ключова архітектурна відмінність Mistral — застосування Sliding Window Attention (SWA). На відміну від класичного механізму self-attention у трансформерах, який масштабується квадратично від довжини контексту, SWA працює за принципом ковзного вікна, обмежуючи кількість токенів, які беруть участь у взаємодії. Наприклад, якщо розмір вікна дорівнює 3, то на першому шарі токен оброблятиме 3 сусідні токени, але на другому шарі токен вже оброблятиме 5 токенів, оскільки один з його токенів вже обробляв 2 вище. Це показано на рисунку 2.7. Зелена лінія вказує на вікно уваги на першому шарі, а червона лінія — на другому шарі. Це різко знижує обчислювальну складність і дозволяє ефективно обробляти довгі послідовності без катастрофічного росту витрат пам'яті та часу. Така властивість робить Mistral потужним інструментом в контексті аналізу логів мережевого обладнання, де розміри вхідних даних можуть бути великими, але ключові залежності часто локальні [34].

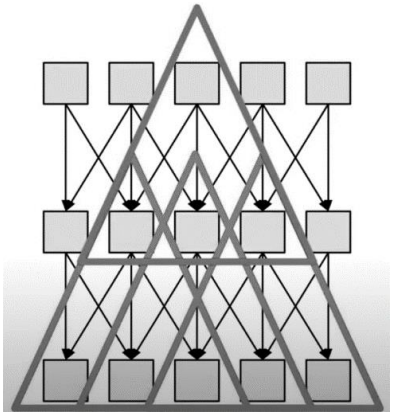


Рисунок 2.7 — Зображення вікон SWA

Архітектурно Mistral також спирається на ефективні активаційні функції (SwiGLU) та RMSNorm, подібно до LLaMA, але із сильнішим акцентом на inference-оптимізації. На рівні тренування Mistral підтримує паралельність по токенах та експертах, що робить її добре пристосованою до масштабних розподілених систем.

На основі проведеного аналізу можливостей існуючих LLM у сфері мережевих технологій, було створено порівняльну характеристику, результати якої було занесено до таблиці 2.1.

Таблиця 2.1 — Порівняльна характеристика можливостей існуючих LLM

Характеристика \ LLM	GPT-5	Gemini	Claude	LLaMA	Mistral
Підтримка довгого контексту	+	+	+	—	+
Мультимодальність (текст, код, медіа)	+	+	—	—	—
Оптимізація для inference	—	—	+	+	+
Відкритість/доступність моделей	—	—	—	+	+
Орієнтація на безпечні відповіді	+	+	+	—	—
Придатність для роботи з логами/моніторингом	+	+	+	—	+

Отже, GPT-5 і Gemini — найбільш універсальні, але закриті й важкі для кастомізації. Claude має переваги у безпеці й роботі з довгим контекстом, проте менш гнучкий у fine-tuning. LLaMA — найбільш відкрита і проста для досліджень модель, але без підтримки мультимодальності й довгого контексту. А Mistral (Large 2) виділяється ефективністю та модульністю.

2.3 Prompt-engineering, як метод взаємодії з моделлю для вирішення практичних завдань

У контексті роботи з великими мовними моделями ключовим елементом взаємодії виступає prompt — текстовий запит, який формулює завдання для

моделі. Саме від змісту та структури запиту залежить, наскільки релевантною і корисною буде відповідь. Prompt можна розглядати як своєрідний інтерфейс між користувачем і моделлю, який замінює традиційне програмування інструкцій явними командами. Якщо у класичному програмуванні розробник створює алгоритми у вигляді чітких операторів і структур керування, то в prompt-engineering інструкції задаються у вигляді природної мови, але з урахуванням певних шаблонів і технік, що дозволяють моделі правильно інтерпретувати задачу. Це створює гнучкий спосіб керування LLM без написання коду, хоча сам принцип нагадує формування правил або сценаріїв. На рисунку 2.8 наведено схему, що відображає різницю між традиційним програмуванням та prompt-engineering [35].

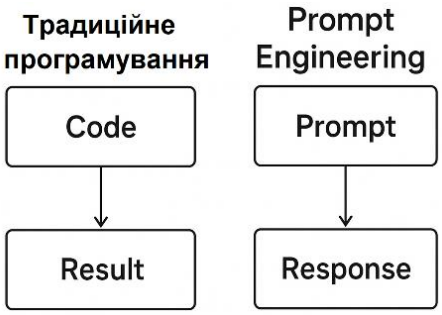


Рисунок 2.8 — Різниця між традиційним програмуванням та prompt-engineering

Prompt може мати різні форми залежно від мети. Найбільш поширеними є інструктивні запити, що задають моделі чітке завдання у вигляді інструкції; рольові, які змушують модель діяти в певному контексті або від імені певного спеціаліста; контекстні, що містять додаткову інформацію, наприклад, вихідні дані або приклади; а також ланцюгові (chain-of-thought), у яких модель ведеться через послідовність логічних кроків. Окремим випадком є few-shot і zero-shot підходи: у першому модель отримує приклади розв’язання задачі для наслідування, у другому приклади не надаються, і LLM мусить інтерпретувати задачу лише з інструкції.

Методи оптимізації промптів у роботі з LLM полягають у підвищенні

точності та стабільності результатів шляхом ретельного формулювання вхідних інструкцій. Ключову роль відіграє правильне використання контексту. Якщо модель отримує вихідні умови задачі у вигляді структурованої інформації — таблиць, списків або чітко сформованих текстових блоків — то її здатність інтерпретувати завдання значно зростає, що є суттєвою перевагою при використанні prompt-engineering для автоматизації траблшутінгу в комп'ютерних мережах. Відмінність від звичайних запитів полягає у тому, що не просто ставиться питання, а створюється середовище, яке підказує моделі логіку очікуваної відповіді. Це особливо важливо у випадках, коли проблема містить багато змінних або передбачає кілька можливих сценаріїв розв'язання [36].

У траблшутінгу в комп'ютерних мережах prompt-engineering можна розглядати як спосіб перетворення неструктурованих діагностичних повідомлень у вхідні запити, які LLM здатна інтерпретувати й порівняти з відомими патернами відмов. Формування промпту дозволяє уточнити, яку саме інформацію необхідно аналізувати: від простого відбору критичних рядків syslog до побудови контексту з кількох джерел. Мова запиту може бути адаптована так, щоб модель розпізнавала в логах сигнатури втрати маршрутів, ознаки нестабільності STP, симптоми некоректної трансляції адрес у NAT тощо. Техніка уточнення інструкцій у промпті забезпечує можливість спрямувати модель на виявлення конкретних кореляцій, наприклад, між повідомленнями про втрату сусіда в OSPF і подальшою зміною таблиці маршрутизації. Завдяки ретельному формулюванню можна змусити модель не лише класифікувати події, а й будувати діагностичний ланцюжок, що відображає логіку інженера: від первинного симптому до можливого кореневого джерела проблеми. У випадку, наприклад, із DHCP це може означати, що LLM отримує контекст про топологію, політику розподілу адрес та конфігураційні параметри, після чого виявляє закономірності, що свідчать про системну помилку. Таким чином, prompt-engineering у цій сфері виконує роль метамови, яка перетворює хаотичний масив логів у структуроване завдання з діагностики, спрямовуючи модель на

визначення найбільш ймовірних технічних причин збою. Блок-схему алгоритму традиційного пошуку несправностей у комп’ютерних мережах наведено в Додатку В.

2.4 Методика проведення експериментальних досліджень

У цьому підрозділі формується методологічна основа, що визначає спосіб проведення експериментальних досліджень ефективності застосування великих мовних моделей для траблшутінгу в комп’ютерних мережах. Основна мета полягає в тому, щоб забезпечити відтворюваність експериментів і надати чіткі критерії для оцінки якості результатів, отриманих при взаємодії з LLM у процесі діагностики та усунення мережевих проблем.

Для симуляції та моделювання мережевих інфраструктур використовується середовище Cisco Packet Tracer, яке забезпечує можливість побудови мережевих топологій різного рівня складності, моделювання конфігурацій маршрутизаторів та комутаторів Cisco за допомогою імітованого інтерфейсу командного рядка. Cisco Packet Tracer використовує інтерфейс користувача з перетягуванням, що дозволяє користувачам додавати та видаляти імітовані мережеві пристрої на свій розсуд. Загалом це середовище дозволяє створювати контрольовані експериментальні сценарії, що є необхідною умовою для порівняння ефективності роботи різних мовних моделей.

Набір сценаріїв включає типові ситуації, які відображають найпоширеніші проблеми в комп’ютерних мережах, які можуть виникати під час їхнього моделювання чи адміністрування: проблеми побудови топологій комп’ютерних мереж, симуляція помилок під час конфігурування статичної та динамічної маршрутизацій, симуляція помилок конфігурації VLAN і протоколу STP, симуляція помилок під час налаштування служб адресації DHCP та NAT, симуляція проблем конфігурування технологій глобальних мереж (на прикладі Frame Relay). Для кожної з цих ситуацій формується набір промптів, що репрезентують реалістичний формат запиту адміністратора до системи,

включаючи як прості інструктивні запити, так і складні ланцюгові запити з елементами few-shot навчання.

Оцінка ефективності роботи моделей здійснюється за кількома формалізованими критеріями. Точність діагностики визначається як відношення кількості правильно визначених помилок до загальної кількості помилок, що були закладені в сценарії. Це виражається формулою (2.4) [37, 38].

$$Accuracy = \frac{N_{correct}}{N_{total}} \times 100\%\,, \tag{2.4}$$

де $N_{correct}$ — кількість правильно визначених помилок;

N_{total} — загальна кількість помилок.

Час отримання відповіді обчислюється як різниця між моментом подачі запиту та моментом формування завершеної відповіді системою і виражається формулою (2.5).

$$T_{response} = t_{out} - t_{in}\,, \tag{2.5}$$

де t_{out} — час отримання відповіді;

t_{in} — час відправлення промпту.

Узагальнена метрика ефективності може бути сформована як середньозважене значення за всіма критеріями і виражається формулою (2.6) [39].

$$E = \omega_1 \cdot Accuracy + \omega_2 \cdot \left(1 - \frac{T_{response}}{T_{max}}\right), \tag{2.6}$$

де ω_1, ω_2 — вагові коефіцієнти, що визначають пріоритетність критеріїв;

T_{max} — максимально допустимий час відповіді.

Таким чином, методика поєднує в собі як технічний аналіз (через симуляцію мережевих сценаріїв у Cisco Packet Tracer), так і формалізовану оцінку результатів на основі кількісних показників. Це дозволяє забезпечити об’єктивність висновків щодо придатності різних великих мовних моделей до використання у сфері мережевого траблшутінгу та формування алгоритму побудови промптів для ефективної мережевої діагностики.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Практичні сценарії тестування

3.1.1 Симуляція проблем моделювання топології дерева

В ході проведення першого експерименту для дослідження ефективності траблшутінгу в комп’ютерних мережах за використання великих мовних моделей, розглядається побудова та аналіз роботи комп’ютерної мережі, структурованої за принципом ієрархічної топології дерева. Такий підхід широко застосовується в корпоративних мережах, оскільки він забезпечує логічний поділ рівнів доступу, агрегації, а також дозволяє реалізувати масштабованість і відмовостійкість. На рисунку 3.1 наведено структурну схему комп’ютерної мережі, яка побудована за цією топологією.

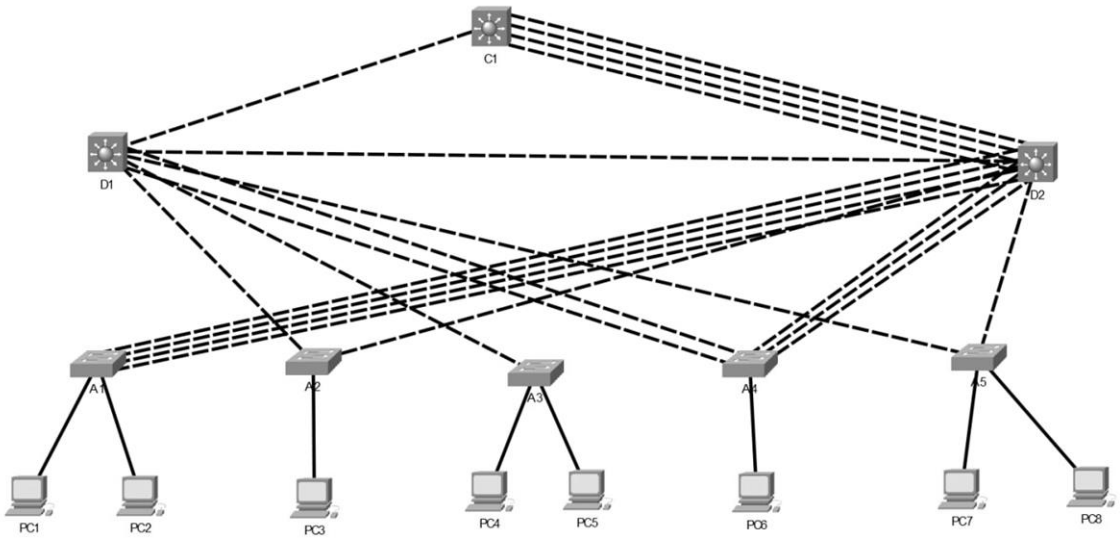


Рисунок 3.1 — Структурна схема ієрархічної мережі

Для моделювання було використано середовище Cisco Packet Tracer. Мережа побудована на базі комутаторів Layer 2 та Layer 3. У топології передбачено як одинарні канали зв’язку між комутаторами, так і паралельні канали, які об’єднані за допомогою механізму EtherChannel із застосуванням протоколу узгодження PAgP.

На логічному рівні в мережі активовано протокол Spanning Tree Protocol (STP), що дозволяє уникнути утворення маршрутних петель і широкомовних

штормів.

Типи зв’язків між комутаторами наведені в таблиці 3.1., де G — Gigabit Ethernet, F — Fast Ethernet, “-“ — зв’язок відсутній. Цифра позначає кількість паралельних каналів, а літера — протокол агрегації: P — Port Aggregation Protocol (PAgP), відсутність літери — статичне налаштування агрегації. Для побудови STP-дерева потрібно призначити кореневий комутатор, що визначає формування оптимального дерева шляхів у мережі. Назва комутатора також наведена в таблиці 3.1.

Таблиця 3.1 — Типи зв’язків між комутаторами

Root	Зв’язки між комутаторами												
	C1-D1	C1-D2	D1-D2	D1-A1	D2-A1	D1-A2	D2-A2	D1-A3	D2-A3	D1-A4	D2-A4	D1-A5	D2-A5
D2	G	4F	G	-	4F	F	F	F	-	2F/P	3F	F	F

Метою цього експериментального сценарію є моделювання некоректних режимів роботи, які можуть виникати у випадках неправильної конфігурації PAgP або STP. Зокрема, буде розглянуто ситуації, коли EtherChannel формується з порушеннями параметрів узгодження, а також випадки, коли STP не забезпечує блокування надлишкових каналів, що призводить до утворення маршрутних петель. Такі проблеми мають критичне значення для роботи мережі, адже вони викликають перевантаження широкомовним трафіком, втрату доступності ресурсів і нестабільність комутації на другому рівні моделі OSI.

Подальший аналіз передбачає застосування методик prompt-engineering для різних великих мовних моделей, які будуть використані як інструмент інтелектуальної діагностики. Це дозволить оцінити ефективність LLM ідентифікувати джерело проблеми, запропонувати коректні кроки конфігурації та сприяти швидшому усуненню збоїв у складних топологіях.

На рисунку 3.2 зображена зібрана схема мережі в середовищі Cisco Packet Tracer із зазначенням назв інтерфейсів та відповідними кольорами портів

(зелений або оранжевий). Зелений — порт у робочому стані Forwarding (активний, передає і приймає трафік). Оранжевий — порт заблокований протоколом STP з метою уникнення петель.

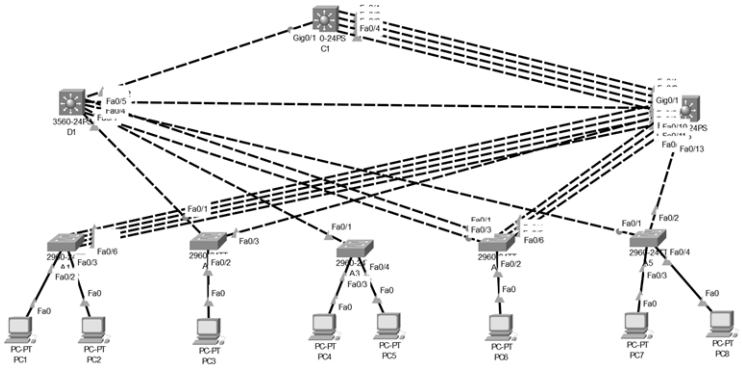


Рисунок 3.2 — Зібрана схема мережі

Спочатку потрібно призначити комутатор D2 на роль кореневого комутатора в дереві STP. Кореневий комутатор є центральною точкою для всієї мережі STP, оскільки на основі його положення STP будує деревоподібну структуру (математичний граф) із вершиною у вигляді кореневого комутатора. Вибір кореневого комутатора забезпечує, що інші комутатори можуть визначити, які порти мають бути активними для передавання трафіку, а які — заблокованими, для уникнення петель. Вибір кореневого комутатора може здійснюватись автоматично на основі Bridge Priority та MAC-адреси комутатора, або ж його можна призначити власноруч. Після цього налаштування за допомогою команди “show spanning-tree” переглянемо статус протоколу STP на комутаторі D2, що зображено на рисунку 3.3.

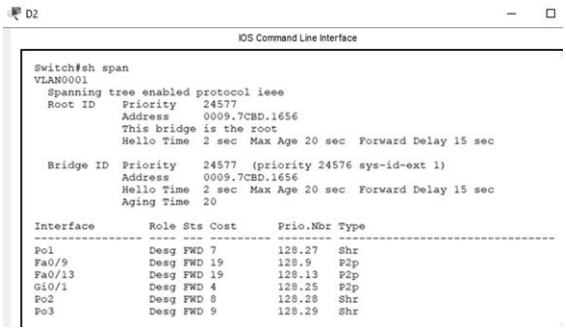


Рисунок 3.3 — Статус протоколу STP на комутаторі D2

Можемо побачити запис “This bridge is the root”, що свідчить про те, що комутатор є кореневим.

Далі потрібно налаштувати агрегацію каналів в мережі центрального офісу. Для цього скористаємось технологією EtherChannel, що була розроблена компанією Cisco Systems. Агрегація каналів потрібна для об’єднання кількох фізичних каналів у один логічний канал для підвищення пропускної здатності та надійності з’єднання між комутаторами.

Агрегацію потрібно налаштовувати між тими комутаторами, між якими є більше ніж одне фізичне з’єднання. На рисунку 3.4 наведено приклад налаштованої агрегації портів із використанням протоколу PAgP між Layer 2 комутатором A4 та Layer 3 комутатором D1.

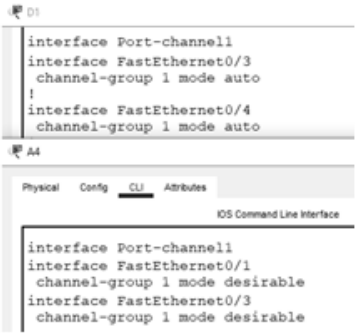


Рисунок 3.4 — Приклад налаштувань PAgP

Перевіримо налаштування за допомогою команди “show etherchannel summary” на комутаторі D1, що зображено на рисунку 3.5.

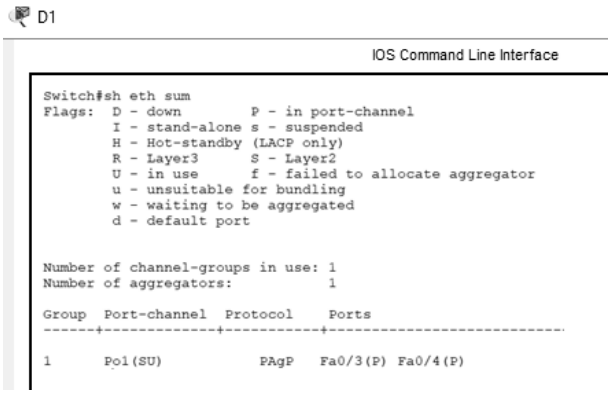


Рисунок 3.5 — Перевірка агрегації

Тепер після проведених коректних налаштувань STP та PAgP у наведеній схемі, симулюємо проблеми, які можуть виникнути в реальних мережах у разі некоректної конфігурації протоколів.

Спочатку для всіх комутаторів було примусово відключено роботу STP. Це призвело до того, що механізм блокування надлишкових шляхів між комутаторами повністю зник, а отже, у топології почали формуватися петлі на канальному рівні. Наслідком цього є виникнення широкомовних штормів, неконтрольоване дублювання кадрів тощо. Це критично впливає на працездатність усієї мережі. Приклад вимкнення STP наведено на рисунку 3.6.

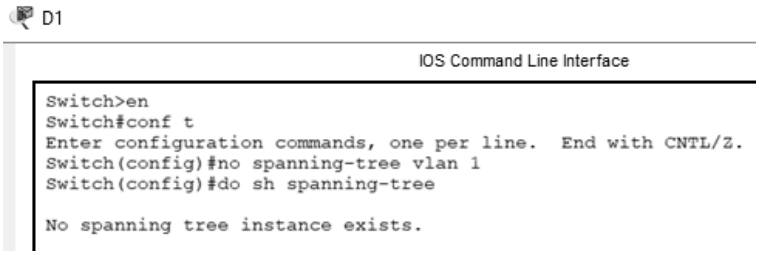


Рисунок 3.6 — Вимкнення STP

Запустимо режим симуляції для демонстрації відсутності роботи протоколу STP, що зображено на рисунку 3.7.

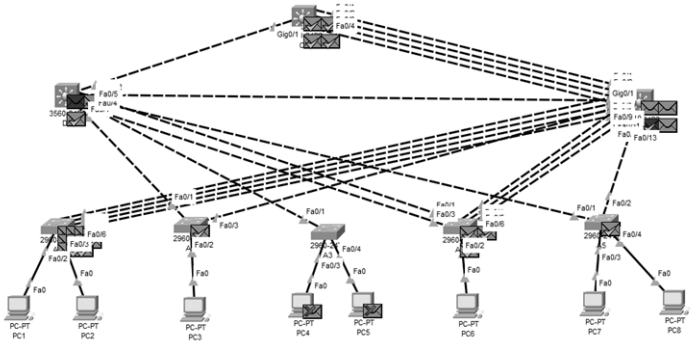


Рисунок 3.7 — Демонстрація відсутності роботи STP в режимі симуляції

В результаті можна побачити, що у графічному інтерфейсі Cisco Packet Tracer оранжева індикація усіх портів комутаторів змінилась на зелену (бо блокування портів відсутнє). Також службові пакети CDP (Cisco Discovery Protocol) та DTP (Dynamic Trunking Protocol) починають безконтрольно

циркулювати мережею та множитись, що демонструє наявність широкомовного шторму.

Друга проблема була змодельована у сегменті з агрегацією каналів. Для комутатора A4 на інтерфейсах, які об’єднані в EtherChannel, було встановлено режим “active”, тобто використано протокол LACP. Водночас на комутаторі D1 для відповідних портів залишився попередній режим “auto”, що відповідає роботі з PAgP. У результаті між комутаторами утворилася конфліктна конфігурація: одна сторона очікує встановлення агрегації через LACP, а інша — через PAgP. Оскільки ці протоколи несумісні, група портів не об’єдналася в коректний логічний канал, і на канальному рівні з’явилися помилки.

Зміна PAgP на LACP спровокує те, що трафік не використовуватиме агрегований канал, балансування навантаження не працюватиме, і трафік може йти по одному фізичному каналу, що створює перевантаження і непрогнозовану поведінку за наявності резервних шляхів. Зміна PAgP на LACP на комутаторі A4 наведена на рисунку 3.8.

A4

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#no int port-channel 1
Switch(config)#int fa0/1
Switch(config-if)#channel-group 1 mode active
Switch(config-if)#no shut

Switch(config-if)#int fa0/3
Switch(config-if)#channel-group 1 mode active
Switch(config-if)#no shut
```

Рисунок 3.8 — Зміна PAgP на LACP

Результат зміни протоколів агрегації каналів на комутаторі A4 наведено на рисунку 3.9.

A4

```
Switch(config-if)#do sh eth sum
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------
+----------------------------------------------

1      Po1(SD)           LACP   Fa0/1(I) Fa0/3(I)
```

Рисунок 3.9 — Результат зміни PAgP на LACP

Після навмисного внесення некоректних змін у конфігурацію мережі та відтворення характерних симптомів (зростання обсягу службового трафіку, поява нестабільності у роботі агрегації каналів) наступним етапом першого експерименту стає застосування методології prompt-engineering для діагностики виявлених збоїв за допомогою розглянутих LLM. Для цього однаковий текстовий запит, сформований на основі симптомів, буде надіслано кільком мовним моделям. Кожна з моделей генерує власний варіант діагностики та рекомендацій.

Приклад структурованого текстового запиту для цього випадку (prompt): “Я моделюю комп’ютерну мережу в Cisco Packet Tracer, яка складається з декількох комутаторів і кінцевих пристроїв. Після первинного налаштування мережа працювала без проблем. Однак після деяких змін у конфігурації ситуація значно погіршилася. У симуляційному режимі я бачу дуже велику кількість службових кадрів, які постійно передаються між комутаторами. Це переважно пакети CDP і DTP. Також інтерфейси комутаторів у візуальному середовищі мають лише зелений колір. Між інтерфейсами двох комутаторів налаштований EtherChannel, який об’єднує два фізичних канали у один логічний, однак чомусь він має обмежену пропускну здатність. Мені потрібно зрозуміти, яка саме проблема відбувається з мережею, чому утворюється така кількість службового трафіку та чому я маю обмежену пропускну здатність між двома комутаторами. Поясни, які можливі причини цього, які саме налаштування могли призвести до такої ситуації, і що варто перевірити в конфігураціях, щоб усунути проблему. Дай коротку та структуровану відповідь у вигляді таблиці.”.

Наведений текстовий запит є ефективним і добре структурованим, оскільки він містить всі необхідні деталі для діагностики проблеми. Запит включає опис конкретних ознак, що дає достатньо інформації для LLM, щоб зробити висновок про можливі причини несправностей.

На рисунку 3.10 наведено результати діагностики проблем із STP та агрегацією каналів у розглянутій мережевій інфраструктурі. Імітаційний

текстовий запит було подано до кількох LLM. Дані відповіді демонструють варіанти інтерпретації та стратегії пошуку несправностей, запропоновані кожною моделлю, а саме: GPT-5, Gemini (2.5 Flash), Claude (Sonnet 4), LLaMA (3.3) та Mistral (Large 2).

Симптом	Можлива причина	Що перевірити / зробити
Велика кількість службових кадрів (CDP, DTP)	Порти працюють у dynamic desirable/auto режимі, DTP активно намагається узгодити trunk	Явно налаштувати порти як trunk або access, за потреби вимкнути DTP
Всі інтерфейси зелені (немає блокування)	Відсутній або вимкнений STP, тому петлі не контролюються	Перевірити, чи ввімкнено STP на VLAN; перевірити root bridge
EtherChannel працює з обмеженнями	Можливе використання різних протоколів агрегації (PAgP ↔ LACP), некоректний режим	Переконатися, що обидва кінці EtherChannel налаштовані на однаковий протокол і режим
Часткова втрата зв'язку між кінцевими пристроями	Топологічні петлі або некоректні VLAN/Trunk налаштування	Перевірити VLAN assignment, trunk дозволені VLAN, відсутність петель

GPT-5

Аспект проблеми	Опис	Можлива причина	Рекомендовані дії
Службовий трафік	Велика кількість кадрів CDP та DTP.	**Мережева петля (loop).** Відсутність протоколу STP для блокування надлишкових шляхів.	Перевірити статус **STP** на всіх комутаторах. Переконатися, що він увімкнений та працює коректно.
EtherChannel	Обмежена пропускна здатність, попри наявність двох фізичних каналів.	**Невідповідність протоколів агрегації** на комутаторах (наприклад, **LACP** на одному, а **PAgP** на іншому).	Перевірити конфігурацію EtherChannel на обох комутаторах і переконатися, що вони використовують **однаковий протокол.**
Загальні симптоми	Часткова втрата зв'язку з кінцевими пристроями, усі індикатори зеленого кольору.	**Широкомовний шторм (broadcast storm)**, що є наслідком мережевої петлі. Це перевантажує мережу і призводить до втрати пакетів.	Усунути першопричини: **вимкнення STP** та **неправильно налаштований EtherChannel**. Після виправлення мережа повинна стабілізуватися.

Gemini

Категорія	Симптом/ Проблема	Можлива причина	Команди для перевірки	Рекомендовані дії
Службовий трафік	Велика кількість CDP/DTP кадрів	Петля в мережі через відсутність STP	`show spanning-tree` `show spanning-tree summary`	Увімкнути STP на всіх комутаторах
Топологія мережі	Постійна передача службових кадрів	Широкомовний шторм	`show mac address-table` `show interfaces status`	Перевірити наявність петель
EtherChannel	Обмежена пропускна здатність	Неузгодженість протоколів агрегації (LACP vs PAgP)	`show etherchannel summary` `show etherchannel port-channel`	Синхронізувати протоколи на обох кінцях
Інтерфейси	Тільки зелений колір портів	Відсутність блокування портів STP	`show spanning-tree interface` `show interfaces trunk`	Налаштувати правильну STP топологію
Підключення	Часткова втрата зв'язку	Нестабільна мережева топологія	`show cdp neighbors` `show logging`	Стабілізувати L2 топологію
Загальна діагностика	Нестабільна робота мережі	Комбінація проблем STP + EtherChannel	`show version` `show running-config`	Послідовно усунути проблеми STP, потім EtherChannel

Claude

Проблема	Можливі причини	Що перевірити та усунути
Некоректна агрегація каналів (EtherChannel)	Використання різних протоколів: PAgP на одному комутаторі, LACP на іншому (несумісність)	Упевнитися, що обидва кінці використовують один протокол: або PAgP, або LACP
	Невідповідність налаштувань швидкості, дуплексу, номерів каналів	Перевірити параметри швидкості, дуплексу і правильність номерів каналів
Вимкнений протокол Spanning Tree (STP)	Відсутність контролю петель у мережі, що викликає безперервну циркуляцію службових кадрів	Увімкнути або коректно налаштувати STP на всіх комутаторах
Великий обсяг службового трафіку	Петлі по мережі через вимкнений STP; робота DTP, що генерує кадри CDP і DTP	Перевірити та оптимізувати налаштування DTP, відключити непотрібні протоколи
Часткова втрата зв'язку кінцевих пристроїв	Нестабільність мережі через петлі і проблеми з агрегацією каналів	Відновити STP і вирівняти налаштування EtherChannel
Невідповідність параметрів портів	Різні швидкості, дуплексні режими на фізичних інтерфейсах, що об'єднані в EtherChannel	Перевірити і узгодити швидкість та дуплекс на всіх фізичних лінках

LLaMA

Параметр	Що перевірити
STP	Чи вимкнено STP на всіх комутаторах? Якщо так, увімкнути (spanning-tree vlan 1)
EtherChannel	Чи узгоджені протоколи (PAgP/LACP) і режими (active/passive/on) на обох сторонах?
CDP/DTP	Чи вимкнено CDP/DTP на інтерфейсах, де вони не потрібні? (no cdp enable, switchport nonegotiate)
VLAN/Trunk	Чи правильно налаштовані транки та VLAN на всіх інтерфейсах?
Навантаження на канали	Чи рівномірно розподіляється трафік між фізичними каналами EtherChannel?

Mistral

Рисунок 3.10 — Результати мережевої діагностики за використання різних LLM

На основі отриманих відповідей можна зробити висновок, що моделі Gemini (2.5 Flash) та Claude (Sonnet 4), надають найбільш повну діагностику для наведеного експериментального випадку. Модель Gemini (2.5 Flash) точно ідентифікувала основні джерела проблеми — відсутність протоколу STP та некоректну конфігурацію EtherChannel, пояснивши механізм виникнення

широкомовного шторму. Відповідь Claude (Sonnet 4), хоч і менш структурована, але є найбільш практично орієнтованою, оскільки включає конкретні команди для діагностики в Cisco OS.

Моделі GPT-5 та LLaMA (3.3) продемонстрували здатність правильно визначати першопричини несправності, але їхні відповіді були менш систематизованими. GPT-5 запропонувала розширений, але дещо надлишковий перелік можливих проблем. Відповідь LLaMA (3.3) не надала достатньо розгорнутого пояснення взаємозв'язку між симптомами та причинами.

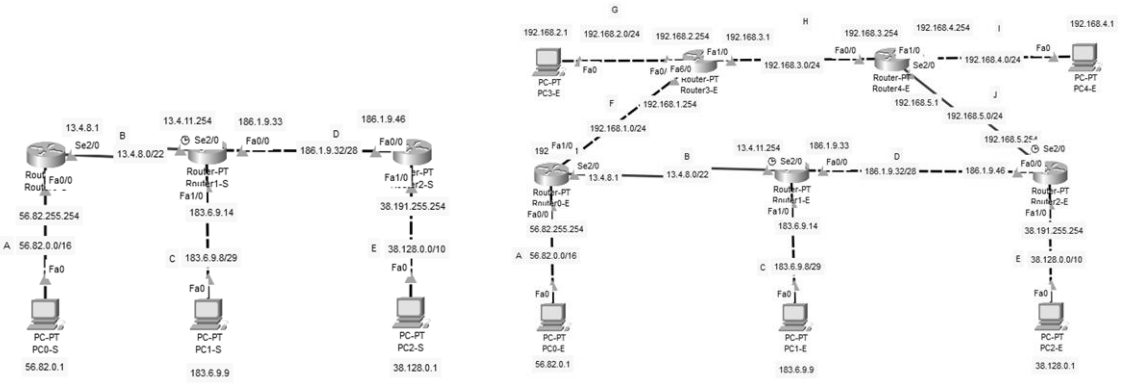
Модель Mistral (Large 2) застосувала унікальний підхід, сформулювавши свою відповідь у вигляді контрольного списку запитань, що є ефективним для досвідчених фахівців, але може бути недостатньо інформативним для менш досвідчених мережевих інженерів.

3.1.2 Симуляція проблем конфігурування маршрутизації

В ході проведення другого експерименту для дослідження ефективності траблшутінгу в комп'ютерних мережах за використання великих мовних моделей, розглядається п'ять подібних топологій мережі, для яких реалізовано кілька варіантів маршрутизації: статична, динамічна (RIP, EIGRP, OSPF).

Мережі спроєктовано таким чином, щоб саме відмінності в конфігурації маршрутизації (а не у фізичній топології та адресному просторі) визначали поведінку мережі. Для прикладу із конфігурацією статичної та динамічної маршрутизації RIP було побудовано дві мережі із однаковою топологією та із однаковим адресним простором. Схеми із динамічною маршрутизацією EIGRP та OSPF є однаковими між собою за топологією та адресним простором і є подібними до схем із статичною маршрутизацією та маршрутизацією RIP. Однак, до цих схем було додано по два додаткових маршрутизатора і хоста, щоб продемонструвати поведінку протоколів у більш розгалужених умовах.

На рисунку 3.11 зображено дві побудовані мережеві топології, для яких налаштовано різні варіанти маршрутизації.



а) статична та RIP б) EIGRP та OSPF

Рисунок 3.11 — Побудовані мережеві топології

Спочатку проведемо огляд технічно-коректних початкових налаштувань кожної з конфігурацій маршрутизації, після цього симулюємо проблеми, які можуть виникнути в реальних умовах у разі некоректної конфігурації статичної та динамічної маршрутизації.

На рисунку 3.12 наведено коректні налаштування статичної маршрутизації для маршрутизаторів Router0-S, Router1-S, Router2-S, відповідно.

Network Address
183.6.9.8/29 via 13.4.11.254
186.1.9.32/28 via 13.4.11.254
38.128.0.0/10 via 186.1.9.46

Network Address
56.82.0.0/16 via 13.4.8.1
38.128.0.0/10 via 186.1.9.46

Network Address
183.6.9.8/29 via 186.1.9.33
13.4.8.0/22 via 186.1.9.33
56.82.0.0/16 via 13.4.8.1

Рисунок 3.12 — Коректні налаштування статичної маршрутизації

Для прикладу переглянемо таблицю маршрутизації на маршрутизаторі Router0-S за допомогою команди “show ip route” (Лістинг 3.1).

Лістинг 3.1 — Таблиця маршрутизації на Router0-S

```
13.0.0.0/22 is subnetted, 1 subnets
C       13.4.8.0 is directly connected, Serial2/0
    38.0.0.0/10 is subnetted, 1 subnets
S       38.128.0.0 [1/0] via 186.1.9.46
    56.0.0.0/16 is subnetted, 1 subnets
```

```
C       56.82.0.0 is directly connected, FastEthernet0/0
    183.6.0.0/29 is subnetted, 1 subnets
S       183.6.9.8 [1/0] via 13.4.11.254
    186.1.0.0/28 is subnetted, 1 subnets
S       186.1.9.32 [1/0] via 13.4.11.254
```

Можемо побачити, що після проведеного налаштування в таблиці з’явились коректні записи статичних маршрутів.

Далі налаштуємо динамічну маршрутизацію RIP в іншій спроєктованій схемі мережі. На рисунку 3.13 наведено коректні налаштування динамічної маршрутизації RIP для маршрутизаторів Router0-R, Router1-R, Router2-R, відповідно.

```
router rip
 version 2
 network 13.0.0.0
 network 56.0.0.0
 no auto-summary
```

```
router rip
 version 2
 network 13.0.0.0
 network 183.6.0.0
 network 186.1.0.0
 no auto-summary
```

```
router rip
 version 2
 network 38.0.0.0
 network 186.1.0.0
 no auto-summary
```

Рисунок 3.13 — Коректні налаштування динамічної маршрутизації RIP

Для прикладу переглянемо таблицю маршрутизації на маршрутизаторі Router0-R за допомогою команди “show ip route” (Лістинг 3.2).

Лістинг 3.2 — Таблиця маршрутизації на Router0-R

```
13.0.0.0/22 is subnetted, 1 subnets
C       13.4.8.0 is directly connected, Serial2/0
    38.0.0.0/10 is subnetted, 1 subnets
R       38.128.0.0 [120/2] via 13.4.11.254, 00:00:16, Serial2/0
    56.0.0.0/16 is subnetted, 1 subnets
C       56.82.0.0 is directly connected, FastEthernet0/0
    183.6.0.0/29 is subnetted, 1 subnets
R       183.6.9.8 [120/1] via 13.4.11.254, 00:00:16, Serial2/0
    186.1.0.0/28 is subnetted, 1 subnets
R       186.1.9.32 [120/1] via 13.4.11.254, 00:00:16, Serial2/0
```

Можемо побачити, що після проведеного налаштування в таблиці з’явились коректні записи маршрутів RIP.

Далі налаштуємо динамічну маршрутизацію EIGRP в іншій спроєктованій схемі мережі. На рисунку 3.14 наведено коректні налаштування динамічної маршрутизації EIGRP для маршрутизатора Router0-E (до прикладу).

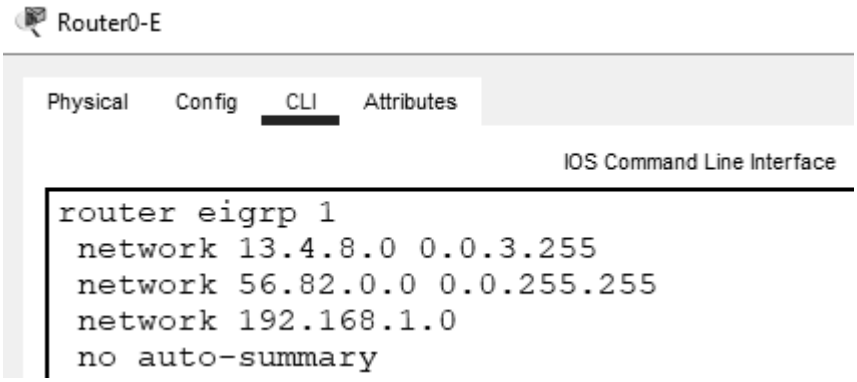


Рисунок 3.14 — Коректні налаштування EIGRP на Router0-E

Далі налаштуємо динамічну маршрутизацію OSPF в іншій спроєктованій схемі мережі. На рисунку 3.15 наведено коректні налаштування динамічної маршрутизації OSPF для маршрутизатора Router0-O (до прикладу) та його таблиця маршрутизації.

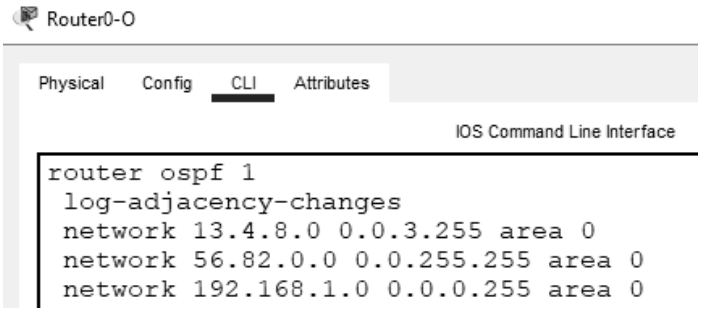


Рисунок 3.15 — Коректні налаштування OSPF на Router0-O та його таблиця маршрутизації

Тепер після визначення базових коректних налаштувань розпочинається етап навмисного внесення конфігураційних помилок, які моделюють типові збої у роботі різних протоколів маршрутизації.

У схемі зі статичною маршрутизацією у визначенні статичного маршруту приберемо перший статичний запис на маршрутизаторі Router0-S. Це призведе

до його відсутності у таблиці маршрутизації, що спричинить втрату зв’язку між окремими сегментами. Перевіримо це за допомогою команди “ping”, що зображено на рисунку 3.16.

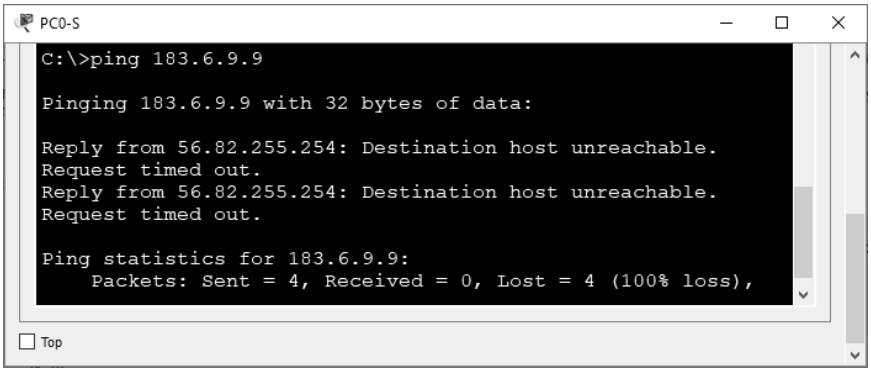


Рисунок 3.16 — Втрата зв’язку між хостами PC0-S та PC1-S

У розробленій схемі мережі використовуються маски змінної довжини (VLSM). Протокол маршрутизації RIP має кілька версій: version 1 та version 2. При налаштуванні динамічної маршрутизації RIP на маршрутизаторах в середовищі Cisco Packet Tracer за замовчуванням буде застосовуватись RIPv1. RIPv1 використовує класову маршрутизацію, а це означає, що він не може відправляти інформацію про маски підмереж під час надсилання оновлень іншим маршрутизаторам. Це обмеження унеможливлює наявність підмереж різного розміру в одному класі мережі. Тому для коректної роботи усі підмережі в класі мережі повинні мати однаковий розмір. Отже, для симуляції проблеми під час конфігурування протоколу RIP на маршрутизаторі Router2-R буде залишено version 1 замість переходу до version 2, що зображено на рисунку 3.17.

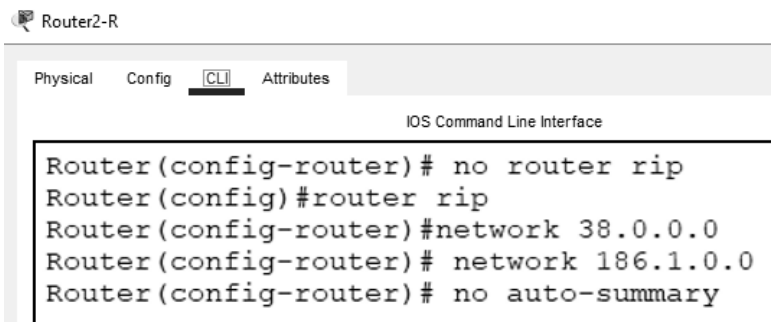


Рисунок 3.17 — Симуляція проблем із RIP на маршрутизаторі Router2-R

В цьому випадку зв’язок із хостом PC2-R між хостами PC0-R та PC1-R буде втрачено, оскільки маршрутизатор Router2-R не зможе відправляти інформацію про маски підмереж під час надсилання оновлень іншим маршрутизаторам.

У випадку із протоколом EIGRP маршрутизатор Router1-E буде виведений з процесу маршрутизації (процес EIGRP на ньому буде вимкнено за допомогою команди “no router eigrp 1”), що порушить оптимальний маршрут передачі пакетів до хоста PC2-E та призведе до втрати прямого зв’язку із хостом PC1-E. Унаслідок цього пакети замість прямого шляху проходитимуть через два додаткових маршрутизатори (Router3-E та Router4-E), утворивши довший та менш ефективний маршрут із підвищеними затримками та неефективним використанням пропускної здатності. Це дозволить на практиці продемонструвати особливості обчислення метрик та побудови альтернативних маршрутів у EIGRP. Переглянемо запис про маршрут до підмережі із адресою 38.128.0.0 на Router0-E до вимкнення EIGRP на Router1-E і після, що зображено на рисунку 3.18.

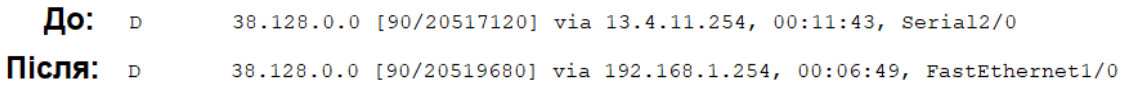


Рисунок 3.18 — Порівняння записів у таблиці маршрутизації

Обидва записи мають однакову адміністративну відстань EIGRP (AD = 90), тобто рішення про вибір шляху базується виключно на порівнянні внутрішніх метрик EIGRP. Перший маршрут (через 13.4.11.254 по Serial2/0) був обраний як найкращий до моменту відключення, він мав нижчу метрику (20517120). Після відключення EIGRP на Router1-E цей шлях перестав рекламуватися безпосередньо, і маршрутизатор отримав альтернативний шлях через іншого сусіда (192.168.1.254) через інтерфейс FastEthernet1/0 — він став найкращим доступним і був записаний у таблицю, хоча його метрика (20519680) є більшою, що свідчить про меншу ефективність цього маршруту.

Переконаємось, що пакет буде проходити саме через нового сусіда (Router3-E). Для цього можна передати пакет в режимі симуляції від PC0-E до PC2-E, або скористатись командою “tracert”, що зображено на рисунку 3.19.

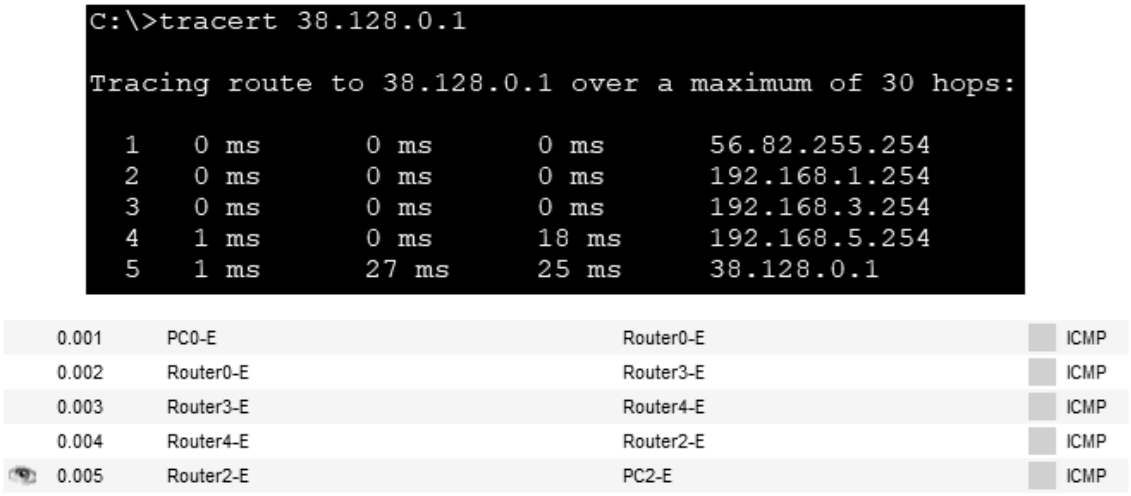


Рисунок 3.19 — Демонстрація проходження пакета за альтернативним маршрутом

У конфігурації з OSPF буде внесено цілеспрямовану зміну на маршрутизаторі Router3-O, де для всіх вказаних мереж замість базової area 0 буде вказано area 1. При цьому на всіх інших маршрутизаторах конфігурація залишиться незмінною, і вони продовжать працювати в area 0.

Маршрутизатори, які очікують встановлення суміжності з Router3-O в area 0, не зможуть сформувати з ним OSPF adjacency через невідповідність area id. У результаті LSDB Router3-O не синхронізується з базами інших маршрутизаторів, а анонси маршрутів з його сегментів не будуть поширені в area 0. Це фактично ізолює підмережі Router3-O від решти мережі, оскільки зв’язок між OSPF-зонами не існуватиме без спеціально налаштованого ABR, якого в даному випадку немає. Після проведених змін переглянемо вміст таблиці сусідів на маршрутизаторі Router0-O (Лістинг 3.3).

Лістинг 3.3 — Демонстрація вмісту таблиці сусідів на Router0-O

```
Neighbor ID     Pri   State           Dead Time   Address         Interface
186.1.9.33        0   FULL/  -        00:00:36    13.4.11.254     Serial2/0
```

Сусід Router3-O не відображається в таблиці сусідів Router0-O. Таким чином, кінцеві пристрої, підключені до Router3-O, втратять можливість обмінюватися даними з іншими сегментами мережі, а топологія буде мати “розрив” між областями. Цей експеримент дозволяє наочно продемонструвати критичну залежність працездатності OSPF від правильного призначення area id.

Після навмисного внесення некоректних змін у конфігурацію різних протоколів маршрутизації, наступним етапом другого експерименту стає застосування методології prompt-engineering для діагностики виявлених збоїв за допомогою розглянутих LLM.

Приклад структурованого текстового запиту для цього випадку (prompt): “Я змоделював чотири комп’ютерні мережі, кожна з яких працює із різними типами маршрутизації: статична, динамічна (RIP, EIGRP, OSPF). В процесі конфігурації я помітив дивні симптоми. У схемі зі статичною мережею одна підмережа стала недоступною. У мережі з RIP один із маршрутизаторів не обмінюється маршрутами із іншими. У мережі з EIGRP пакет проходить не за оптимальним маршрутом через конкретний маршрутизатор, а обирає маршрут через інші додаткові маршрутизатори, що створює затримки. У конфігурації з OSPF маршрутизатори в одній топології перестали формувати сусідство між собою. На одному із маршрутизаторів я отримую лог: mismatch area ID, from backbone area must be virtual-link but not found. Поясни, які можливі причини описаних проблем, які саме налаштування могли призвести до такої ситуації, і що варто перевірити в конфігураціях, щоб усунути проблему. Дай коротку та структуровану відповідь у вигляді таблиці.”.

Наведений текстовий запит є ефективним і добре структурованим, оскільки він містить всі необхідні деталі для діагностики проблеми. Запит включає опис конкретних ознак, що дає достатньо інформації для LLM, щоб зробити висновок про можливі причини несправностей.

Імітаційний текстовий запит було подано до кількох розглянутих LLM. На рисунку 3.20 наведено результати діагностики проблем.

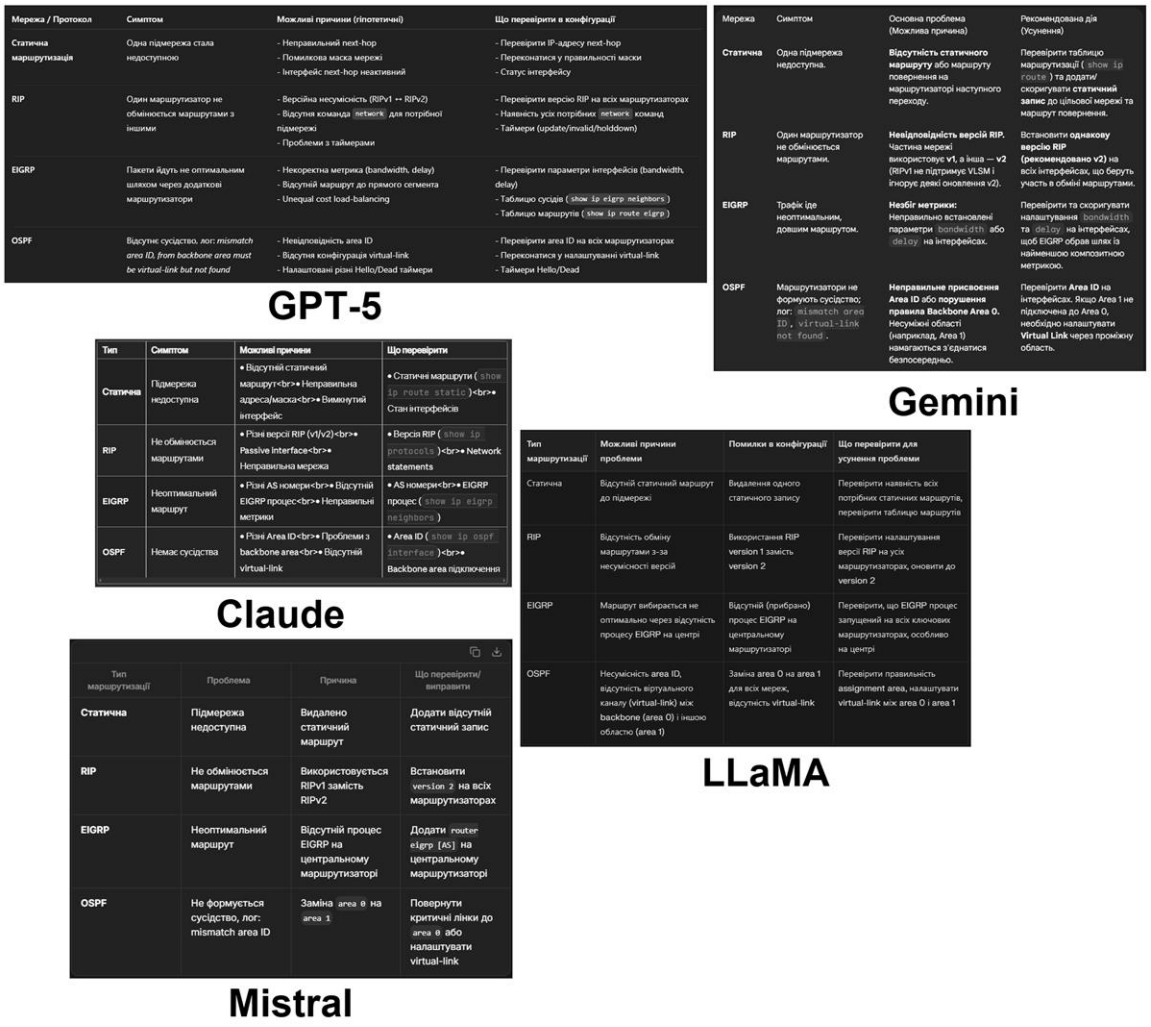


Рисунок 3.20 — Результати мережевої діагностики за використання різних LLM

На основі отриманих відповідей можна зробити висновок, що наведені LLM проявляють різний рівень ефективності у діагностиці мережевих проблем, пов’язаних із конфігурацією статичної та динамічної маршрутизації.

Найбільш деталізованою та структурованою виявилася відповідь моделі GPT-5, яка надала систематизований опис симптомів, можливих причин та кроків для перевірки, використавши коректну термінологію та команди діагностики для наведеного експерименту. Такий підхід найбільше відповідає вимогам до практичної роботи мережевого інженера, оскільки дозволяє не лише ідентифікувати ймовірну проблему, але й одразу отримати алгоритм перевірки конфігурації.

Модель Gemini (2.5 Flash), хоч і продемонструвала коректність виявлення основних причин, але подає інформацію у більш узагальненому вигляді. Відповідь містить опис проблем і рекомендацій без надмірної деталізації, що спрощує сприйняття, але знижує практичну цінність для діагностики складних випадків. Модель відзначається чіткою структурою викладу та лаконічністю, що робить її результат придатним для швидкої попередньої оцінки ситуації.

Claude (Sonnet 4) вирізняється тим, що подає відповідь із використанням реальних прикладів CLI-команд. Це може розглядатися як перевага, адже інженер отримує прямі інструкції для перевірки. Проте подана інформація є надмірно спрощеною, а деякі формулювання носять гіпотетичний характер, що обмежує можливість точного діагностування у складних сценаріях.

Відповіді моделей LLaMA (3.3) та Mistral (Large 2) виглядають найменш інформативними з точки зору практичного застосування. Вони зосереджуються на загальних причинах та базових рекомендаціях, однак бракує детального пояснення механізмів виникнення проблем та конкретних інструментів діагностики. У результаті такі відповіді можуть бути корисними лише на початкових етапах аналізу, але не забезпечують достатньої глибини для повного усунення несправностей.

3.1.3 Симуляція проблем конфігурування VLAN

В ході проведення третього експерименту для дослідження ефективності траблшутінгу в комп’ютерних мережах за використання великих мовних моделей, розглядається побудова та аналіз роботи комп’ютерної мережі, що побудована на комутаторах.

Для дослідження було змодельовано комп’ютерну мережу, що зображена на рисунку 3.21, яка складається з чотирьох комутаторів та маршрутизатора. За допомогою маршрутизатора реалізовано inter-VLAN маршрутизацію за принципом Router-on-a-Stick. Така архітектура дає змогу ефективно розділити мережу на логічні сегменти, забезпечуючи оптимальний розподіл трафіку.

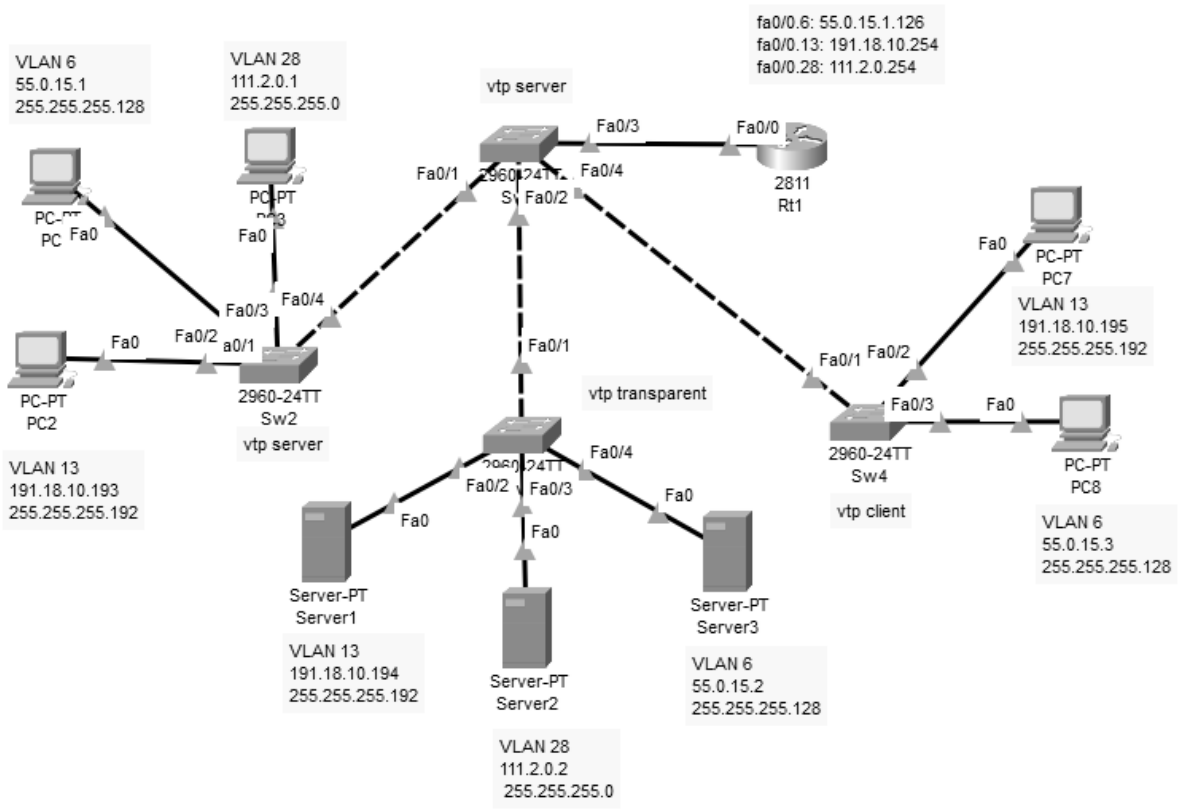


Рисунок 3.21 — Змодельована схема мережі

У мережі створено три VLAN: VLAN 13 (мережа 191.18.10.192/26), VLAN 6 (мережа 55.0.15.0/25) та VLAN 28 (мережа 111.2.0.0/24). Для централізованого керування віртуальними локальними мережами використовується протокол VTP (VLAN Trunking Protocol), при цьому ролі комутаторів розподілені наступним чином: Sw1 — VTP Server, Sw2 — VTP Server, Sw3 — VTP Transparent, Sw4 — VTP Client.

Спочатку проведемо огляд технічно-коректних початкових налаштувань комутаторів та маршрутизатора Rt1, після цього буде проведено симуляцію можливих проблем, що можуть виникати при конфігуруванні VLAN та протоколу VTP у наведеній комп’ютерній мережі.

Спочатку створюємо відповідні VLAN на усіх чотирьох комутаторах за допомогою команд: “vlan 6”, “vlan 13”, “vlan 28” та даємо їм імена. Приклад конфігураційного файлу із налаштуванням віртуальних мереж VLAN 6, VLAN 13, VLAN 28 на комутаторі Sw2 наведено на рисунку 3.22.

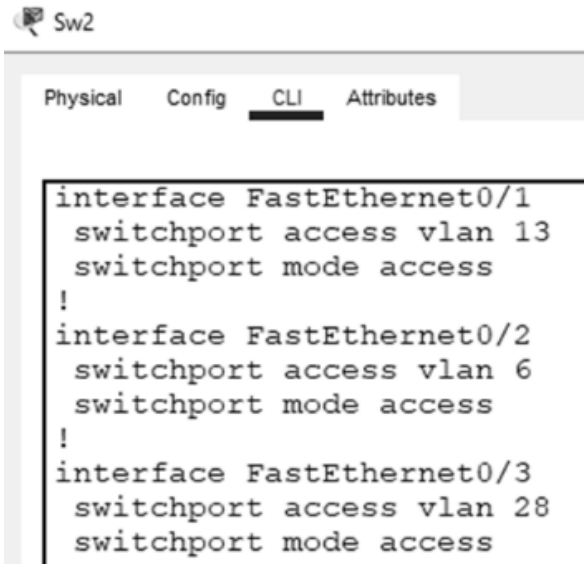


Рисунок 3.22 — Коректні налаштування VLAN на комутаторі Sw2

Перевіряємо коректність налаштувань VLAN за допомогою команди “show vlan brief”, що зображено на рисунку 3.23.

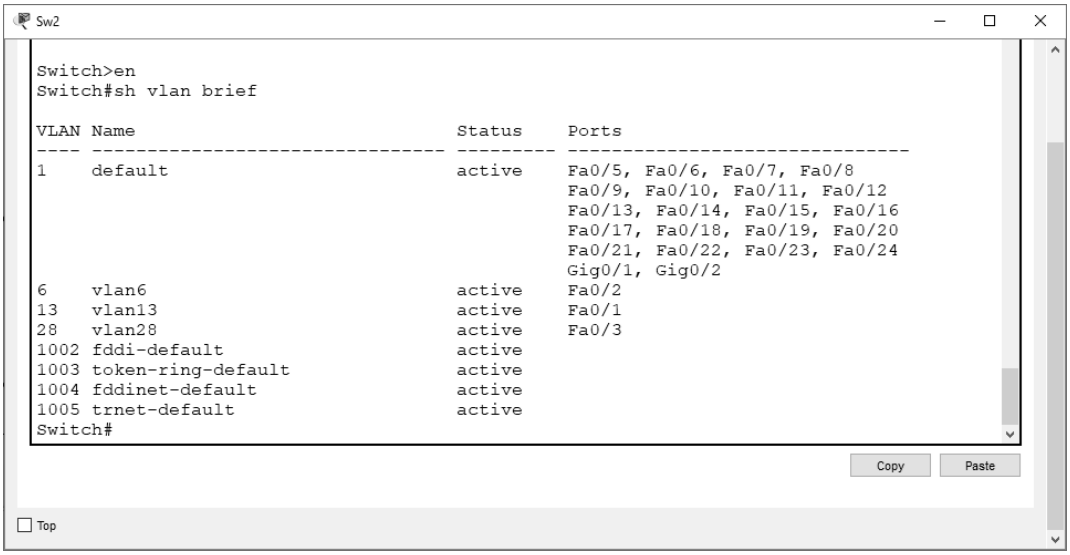


Рисунок 3.23 — Перевірка коректності налаштувань створених VLAN

Аналогічно налаштовуємо VLAN на комутаторах Sw3 та Sw4. Далі потрібно налаштувати інтерфейси комутатора Sw1. До цього комутатора відсутні під’єднання кінцевих пристроїв, отже достатньо налаштувати інтерфейси у trunk-режимі. Приклад конфігураційного файлу із коректним налаштуванням trunk-портів на комутаторі Sw1 наведено на рисунку 3.24.

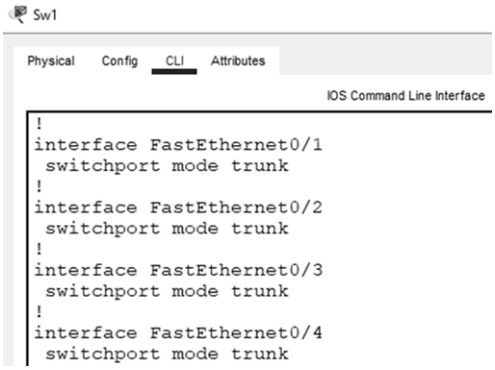


Рисунок 3.24 — Переведення інтерфейсів у trunk-режим

Тепер можна встановити зв’язок усередині кожного VLAN. Але між різними VLAN зв’язку не буде. Для того, щоб організувати зв’язок і між вузлами, що належать різним VLAN на маршрутизаторі Rt1 слід налаштувати маршрутизацію. Щоб налаштувати інтерфейс маршрутизатора, через який він з’єднується із комутатором Sw1, необхідно зробити у ньому декілька підінтерфейсів. Налаштування inter-VLAN маршрутизації на Rt1 наведено на рисунку 3.25.

Рисунок 3.25 — Налаштування inter-VLAN маршрутизації

Кожен підінтерфейс на маршрутизаторі логічно відповідає певному VLAN і виконує роль шлюзу для пристроїв цього сегмента. Використання команди “encapsulation dot1Q” дає змогу маршрутизатору коректно ідентифікувати, до якого VLAN належить кадр, що надходить із комутатора, адже стандарт IEEE

802.1Q визначає спосіб маркування кадрів спеціальним тегом VLAN. Завдяки цьому один фізичний інтерфейс маршрутизатора фактично розділяється на кілька віртуальних каналів, кожен з яких обслуговує власну підмережу. Вміст конфігураційного файлу маршрутизатора Rt1 наведено в Додатку Г.

Після проведених коректних налаштувань, перевіримо працездатність усієї мережі в режимі симуляції, шляхом передавання ICMP-пакетів від кінцевого пристрою PC1 до усіх інших, що зображено на рисунку 3.26.

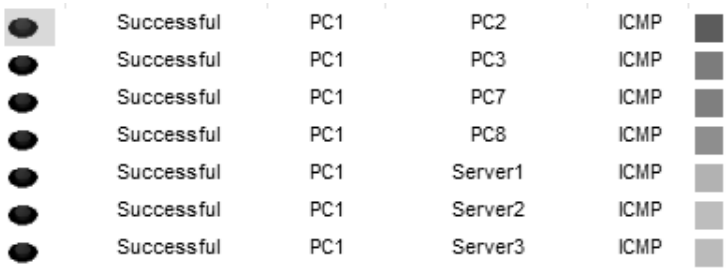


Рисунок 3.26 — Перевірка працездатності мережі в режимі симуляції

Далі розглянемо коректні налаштування VTP в мережі. На рисунку 3.27 наведено приклад налаштування VTP-сервера на комутаторі Sw1.

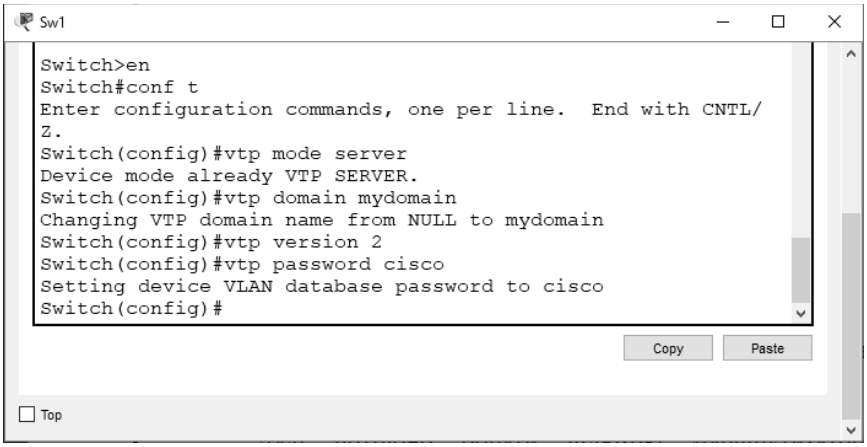


Рисунок 3.27 — Приклад налаштування VTP-сервера

Аналогічно налаштовуємо VTP на решті комутаторів, вказавши тип: vtp server, vtp client, vtp transparent. Перевіряємо налаштування за допомогою команди “show vtp status” на комутаторі Sw3 (vtp transparent), що зображено на рисунку 3.28.

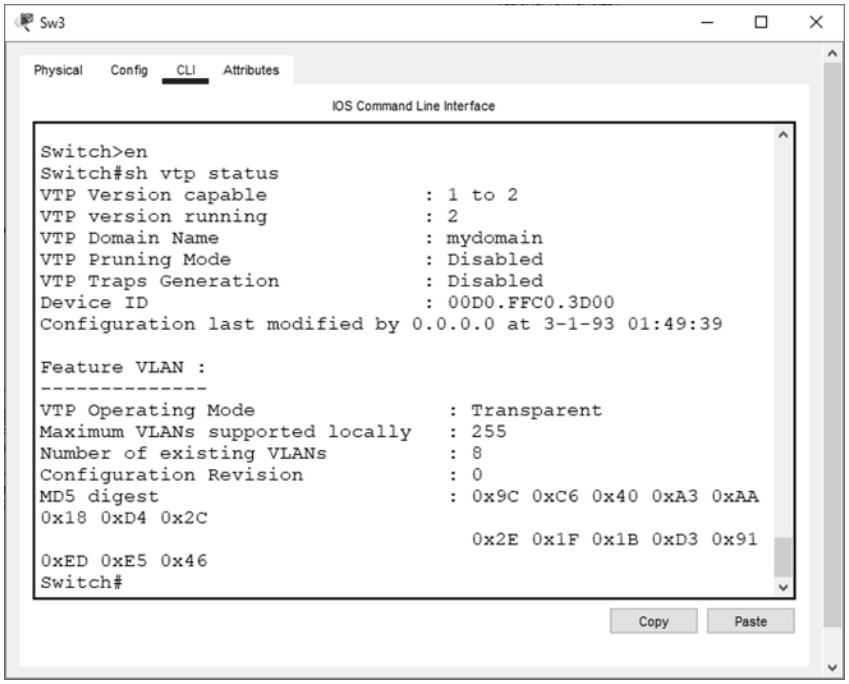


Рисунок 3.28 — Перевірка налаштувань VTP

Налаштування VTP у побудованій мережі забезпечує централізоване керування віртуальними локальними мережами, спрощуючи процес їхнього створення, зміни та видалення. У цій конфігурації один комутатор виступає сервером, інший працює як клієнт, ще один виконує роль додаткового сервера, а четвертий налаштований у прозорому режимі. Такий підхід дозволяє поширювати інформацію про VLAN з одного центру на інші пристрої мережі, уникаючи необхідності дублювання налаштувань на кожному комутаторі окремо.

Тепер після визначення базових коректних налаштувань VLAN та VTP для належного функціонування мережі на основі комутаторів, розпочинається етап навмисного внесення конфігураційних помилок, які моделюють типові збої у роботі VLAN та VTP.

Найбільш показовим буде випадок, пов’язаний із некоректною маршрутизацією між різними VLAN через неправильне створення підінтерфейсів на маршрутизаторі, що призведе до повної ізоляції сегментів мережі. Наприклад, внесемо зміни до конфігурації на маршрутизаторі Rt1, що зображено на рисунку 3.29.

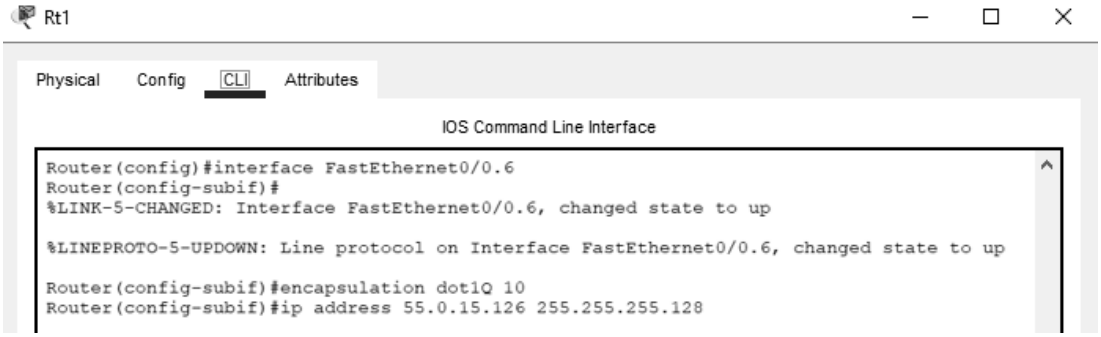


Рисунок 3.29 — Симуляція проблем із inter-VLAN маршрутизацією

У наведеному прикладі підінтерфейс маршрутизатора створюється для обробки трафіку з VLAN 6, але замість коректного VLAN ID використовується інкапсуляція з тегом 10. Це призведе до розриву зв’язку із іншими VLAN, оскільки маршрутизатор фактично не розпізнає трафік цього VLAN, хоч і працездатність мережі у межах одного VLAN 6 буде залишатись незмінною.

Інший клас проблем стосується безпосередньо конфігурації VLAN на комутаторах. Наприклад, видалення VLAN на одному з серверів VTP призведе до автоматичного поширення змін по всій доменній інфраструктурі й спричинить масове порушення зв’язку. Симуляція цієї проблеми наведена на рисунку 3.30.

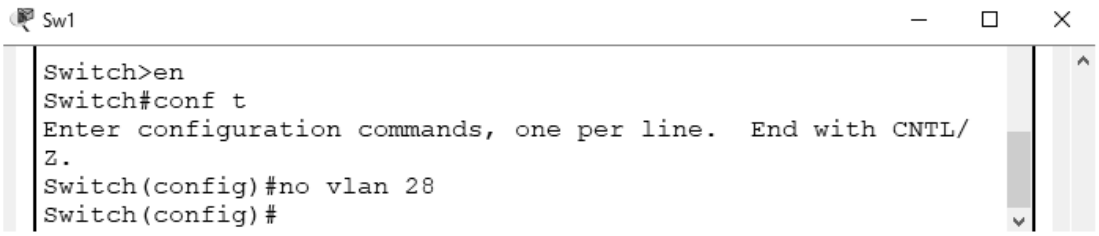


Рисунок 3.30 — Видалення VLAN 28 на Sw1

Результатом цієї конфігураційної помилки стане те, що VLAN 28 зникне на усіх комутаторах, які виконують ролі VTP-клієнтів та VTP-серверів. У випадку створеної мережі — це комутатори Sw1 (сервер), Sw2 (сервер), Sw4 (клієнт). На комутаторі Sw3 усе залишиться без змін, оскільки він має роль VTP Transparent і не синхронізує інформацію про конфігурацію VLAN з іншими

комутаторами. Для прикладу, за допомогою команди “show vlan brief” переконаємось, що інформація про VLAN 28 зникла із комутатора Sw4, що зображено на рисунку 3.31.

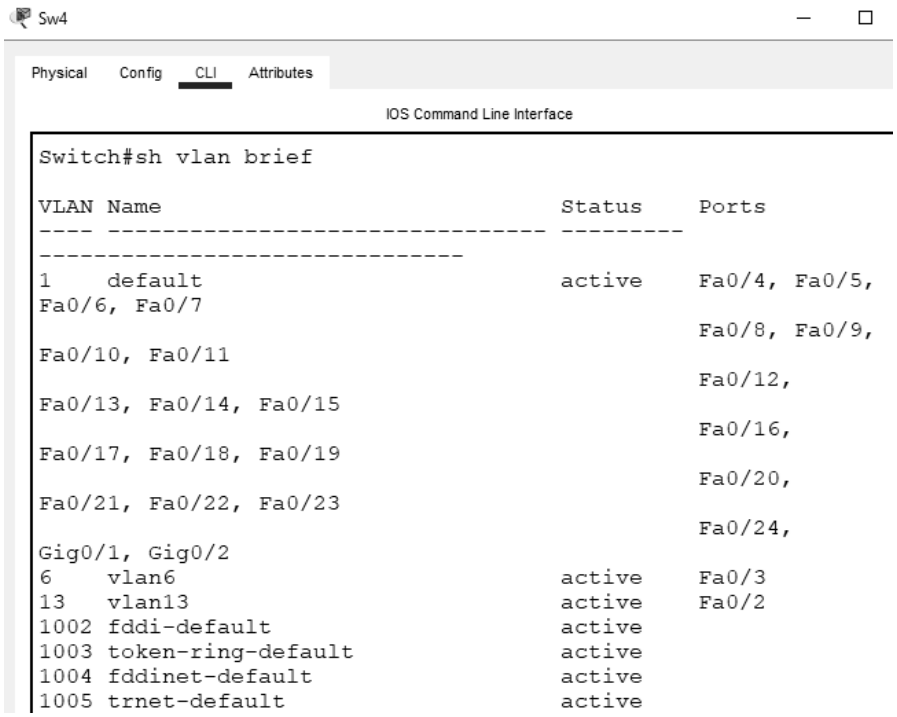


Рисунок 3.31 — Демонстрація відсутності VLAN 28 на комутаторі Sw4

Окрему групу становлять проблеми із VTP. Наприклад, критичним випадком є зміна VTP password на одному із комутаторів, що призведе до припинення синхронізації інформації про VLAN між цим комутатором та всіма іншими комутаторами, на яких використовується інший пароль. На рисунку 3.32 наведено приклад зміни VTP password на комутаторі Sw2.

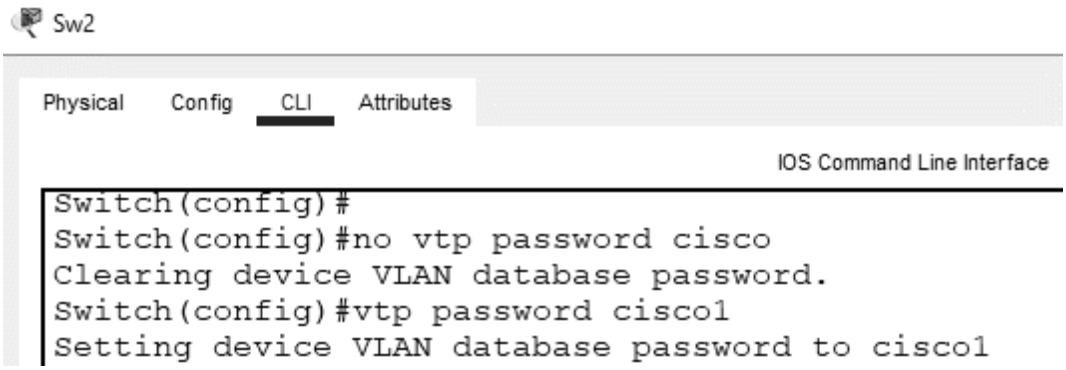


Рисунок 3.32 — Зміна VTP password на комутаторі Sw2

Після навмисного внесення некоректних змін у конфігурацію VLAN та VTP у спроєктованій комп'ютерній мережі, наступним етапом третього експерименту стає застосування методології prompt-engineering для діагностики виявлених проблем за допомогою розглянутих LLM.

Приклад структурованого текстового запиту для цього випадку (prompt): "Я змоделював комп'ютерну мережу, яка працює на основі комутаторів із використанням VLAN 6, 13, 28. Inter-VLAN маршрутизація працює за допомогою Router-on-a-Stick. В мережі працює VTP. Під час налаштування я зіштовхнувся із кількома проблемами. У мене є кілька підінтерфейсів на маршрутизаторі, які мають забезпечувати зв'язок між різними сегментами, однак частина комп'ютерів, підключених до VLAN 6, виявилася повністю ізольованою. Зв'язок між пристроями присутній лише в межах одного VLAN 6. Пристрої в інших VLAN мають між собою зв'язок. Також в мережі на кількох комутаторах зникли налаштування VLAN 28, після внесення змін на одному із комутаторів. Ще одна проблема полягає в тому, що деякі комутатори перестали отримувати оновлення про VLAN від свого VTP-сервера. Поясни, які можливі причини описаних проблем, які саме налаштування могли призвести до такої ситуації, і що варто перевірити в конфігураціях, щоб усунути проблему. Дай коротку та структуровану відповідь у вигляді таблиці.".

Наведений текстовий запит є ефективним і добре структурованим, оскільки він подає проблему у вигляді реалістичного сценарію налаштування мережі, де одночасно описуються кілька симптомів, які не дають прямої підказки щодо конкретної помилки. Завдяки чіткому викладу умов із зазначенням використання VLAN, Router-on-a-Stick та VTP, він створює достатній контекст для аналізу різними LLM, але при цьому не розкриває справжніх причин збоїв.

Імітаційний текстовий запит було подано до кількох розглянутих LLM. На рисунку 3.33 наведено результати діагностики проблем. Проведений аналіз дав змогу порівняти не лише коректність отриманих рекомендацій, але й здатність моделей узагальнювати інформацію та пропонувати системний підхід до пошуку несправностей.

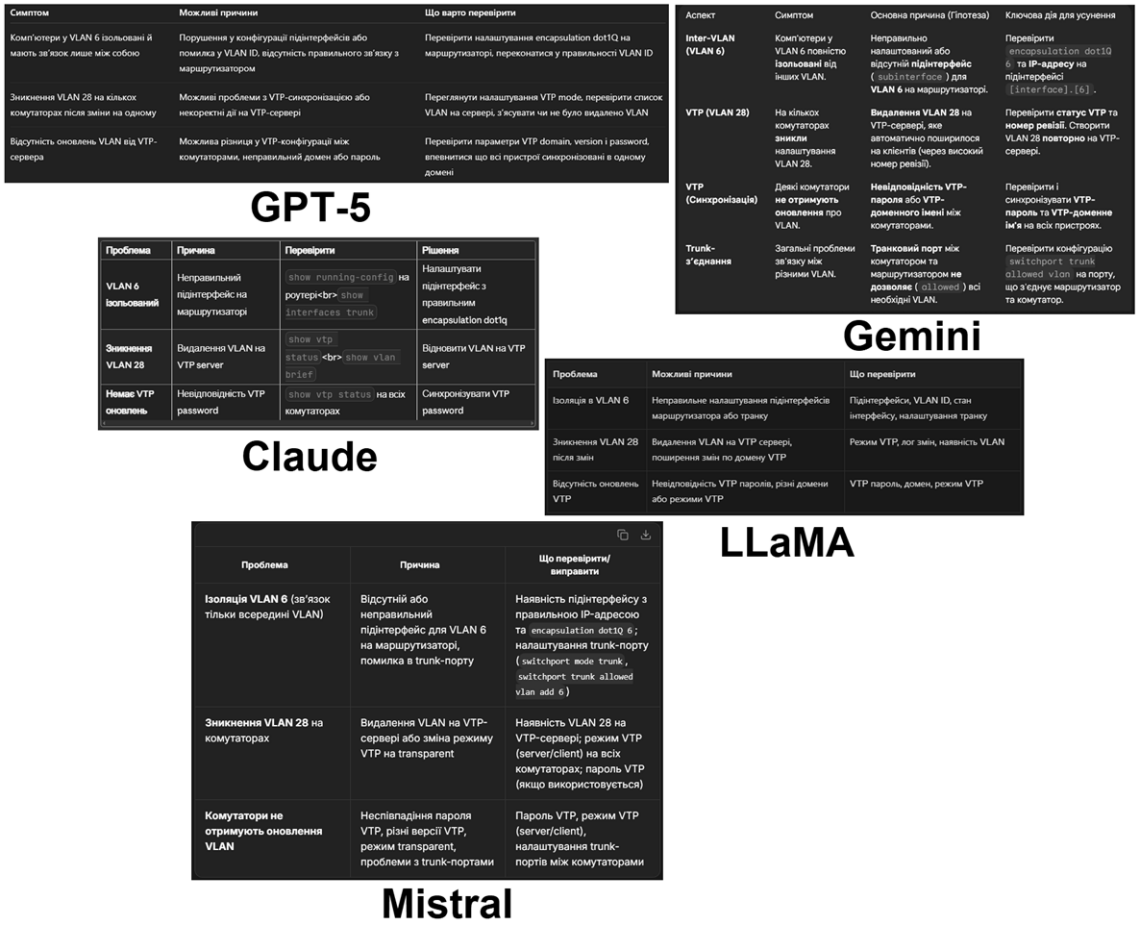


Рисунок 3.33 — Результати мережевої діагностики за використання різних LLM

На основі отриманих відповідей можна зробити висновок, що наведені LLM проявляють різний рівень ефективності у діагностиці мережевих проблем, пов’язаних із конфігуруванням VLAN та VTP.

Найбільш комплексним і систематизованим виявився результат моделі GPT-5, яка запропонувала чітку структуру “симптоми — можливі причини — що перевірити” та зберіг баланс між гіпотетичними поясненнями й конкретними кроками діагностики. Такий підхід дозволяє швидко співвіднести спостережувані збої із потенційними конфігураційними помилками, що робить відповідь максимально корисною в умовах практичного тестування.

Модель Gemini (2.5 Flash) надала детальний опис аспектів і симптомів, які спостерігаються у мережі, поєднавши їх із гіпотезами та можливими кроками вирішення. Її відповідь відзначається добрим рівнем систематизації, але

водночас містить елементи надмірного теоретизування, що може ускладнити швидке практичне застосування.

Модель Claude (Sonnet 4) продемонструвала дещо інший стиль, акцентуючи увагу на конкретних командах діагностики, які слід виконати на мережевому обладнанні. Ця відповідь має прикладний характер, однак є менш збалансованою, оскільки одразу пропонує інструменти для перевірки без достатньо розгорнутого пояснення природи проблеми. Такий підхід ефективний для досвідченого інженера, але менш інформативний для користувача, який прагне зрозуміти причинно-наслідкові зв’язки у конфігурації.

LLaMA (3.3) обмежився базовим переліком проблем і загальних напрямків перевірки. Така відповідь є коректною, але досить поверховою: вона вказує на типові помилки (невірний VLAN ID, видалення VLAN, різниця в параметрах VTP), проте не заглиблюється у діагностичні процедури чи практичні команди, що знижує її прикладну цінність у контексті реального практичного застосування.

3.1.4 Симуляція проблем конфігурування адресних служб

В ході проведення четвертого експерименту для дослідження ефективності траблшутінгу в комп’ютерних мережах за використання великих мовних моделей, розглядається виявлення характерних помилок, які виникають у процесі конфігурування адресних служб DHCP та NAT, а також відпрацювання методик їхнього діагностування та усунення.

Для дослідження було змодельовано комп’ютерну мережу, яка охоплює кілька підмереж з різними підходами до призначення адрес, що дозволяє комплексно дослідити як динамічне, так і статичне налаштування параметрів. Для симуляції роботи NAT мережу було умовно розділено на локальну частину та глобальну (Інтернет). Схема мережі зображена на рисунку 3.34. Окрему увагу під час моделювання приділено створенню ситуацій, у яких NAT може працювати некоректно — як через конфігураційні помилки, так і через конфлікти між внутрішніми та зовнішніми адресами.

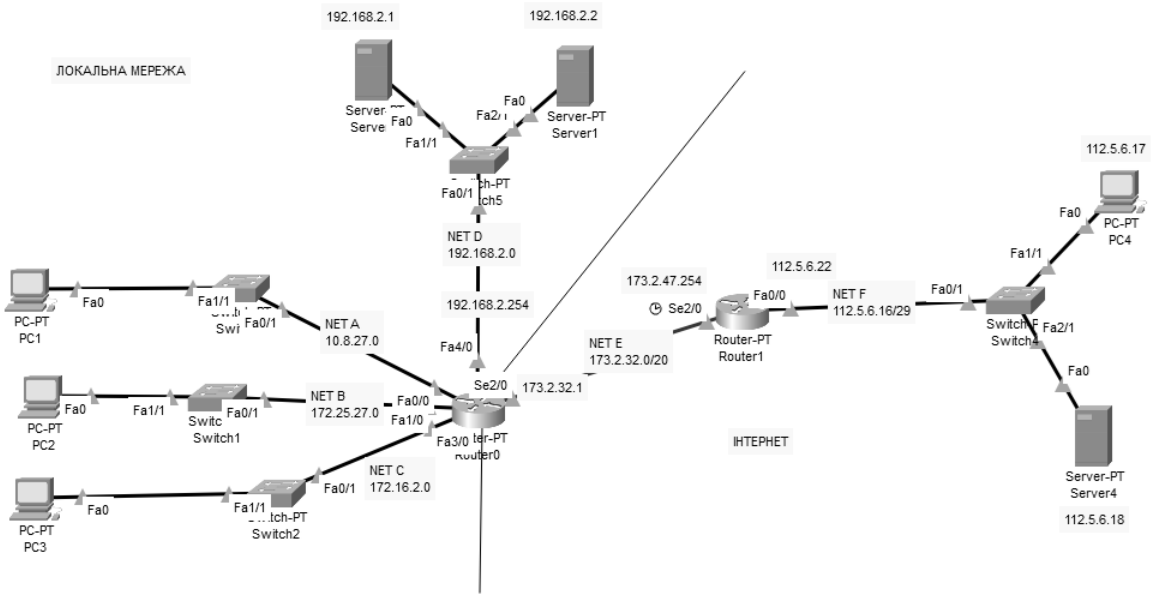


Рисунок 3.34 — Змодельована схема мережі

У побудованій топології окремі сегменти локальної мережі (NET A, NET B, NET C) використовують DHCP на базі маршрутизатора Router0 для динамічного призначення адрес хостам: PC1, PC2, PC3, тоді як для Server1 та Server2 адреси призначені статично. Для глобальної частини мережі налаштування хостів PC4 та Server4 також є статичними. Адреси підмереж: NET A (10.8.27.0/24), NET B (172.25.27.0/24), NET C (172.16.2.0/24), NET D (192.168.2.0/24), NET E (173.2.32.0/20), NET F (112.5.6.16/29)

Маршрутизатор Router0 виконує роль основного вузла, відповідального за динамічний розподіл адрес, а також забезпечує вихід трафіку із локальної частини мережі в глобальну через Router1 за допомогою налаштованого статичного маршруту за замовчуванням.

Спочатку проведемо огляд технічно-коректних початкових налаштувань маршрутизатора Router0, після цього буде проведено симуляцію можливих проблем, що можуть виникати під час конфігурування DHCP та NAT у наведеній комп’ютерній мережі.

На рисунку 3.35 наведено частину конфігураційного файлу маршрутизатора Router0 із активізацією DHCP-сервісу для динамічного призначення адрес хостам в мережах NET A, NET B, NET C.

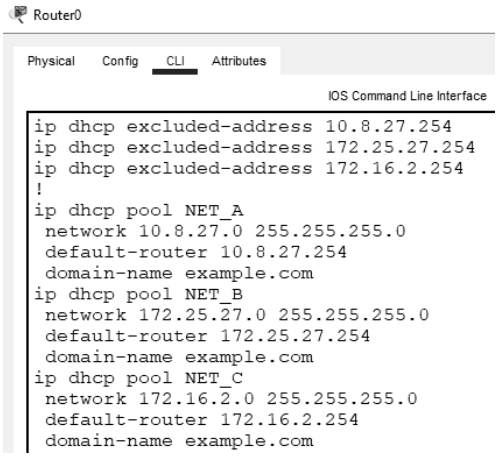


Рисунок 3.35 — Конфігурація DHCP-сервісу на Router0

Команди “ip dhcp excluded-address” вказують, які саме IP-адреси не повинні видаватися DHCP-сервером клієнтам. В цьому випадку — це адреси інтерфейсів маршрутизатора Router0. Після цього було створено три окремі DHCP-пули для трьох різних локальних мереж. Кожен пул визначає діапазон адрес і параметри, які будуть надаватися клієнтам у відповідній мережі. Вміст конфігураційного файлу маршрутизатора Router0 наведено в Додатку Д.

Перевіримо, чи надходить DHCP-запит від кінцевого пристрою PC1 до маршрутизатора Router0, що наведено на рисунку 3.36

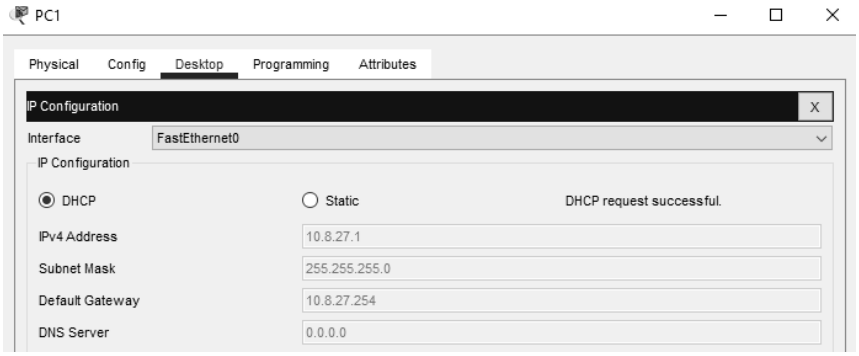


Рисунок 3.36 — Перевірка працездатності DHCP-сервісу

Отже, DHCP-сервіс на основі Router0 було налаштовано коректно, про що свідчить успішний DHCP-запит від кінцевого пристрою до маршрутизатора.

Далі коректно налаштуємо статичний NAT для серверів Server1 та Server2. Це потрібно для того, аби вони мали постійні, незмінні зовнішні IP-адреси, за

якими до них можна отримати доступ із глобальної частини мережі. Статичний NAT відображає кожну внутрішню IP-адресу на єдину зовнішню IP-адресу.

Приклад налаштування статичної трансляції адрес для серверів наведено на рисунку 3.37. Для глобальних адрес серверів було обрано дві вільні адреси із NET E.

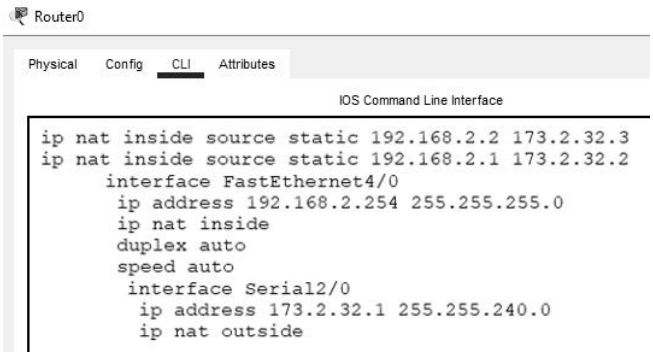


Рисунок 3.37 — Коректне налаштування статичного NAT

Перевіримо працездатність трансляції, відправивши пакет з PC4 на Server1. Для цього переглянемо вміст пакету в режимі симуляції (зокрема IP-адресу відправника) до і після проходження пакета через маршрутизатор, що реалізує NAT, що наведено на рисунку 3.38.

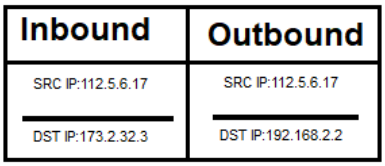


Рисунок 3.38 — Вміст пакету в режимі симуляції

Також переглянемо таблицю трансляцій на Router0 (Лістинг 3.4). При цьому, під час відправлення ICMP-пакету за допомогою команди “ping”, потрібно вказувати зовнішню IP-адресу Server1.

Лістинг 3.4 — Демонстрація вмісту таблиці трансляцій

```
Pro  Inside global     Inside local       Outside local      Outside global
icmp 173.2.32.3:10     192.168.2.2:10     112.5.6.17:10      112.5.6.17:10
icmp 173.2.32.3:11     192.168.2.2:11     112.5.6.17:11      112.5.6.17:11
```

```
icmp 173.2.32.3:12    192.168.2.2:12    112.5.6.17:12     112.5.6.17:12
icmp 173.2.32.3:9     192.168.2.2:9     112.5.6.17:9      112.5.6.17:9
---  173.2.32.2       192.168.2.1       ---               ---
---  173.2.32.3       192.168.2.2       ---               ---
```

Далі коректно налаштуємо динамічний NAT для мережі NET A. В пул зовнішніх адрес введені усі вільні адреси з мережі NET E, що зображено на рисунку 3.39. Динамічний NAT відображає одну приватну IP-адресу на одну публічну IP-адресу із пулу доступних публічних адрес. Зіставлення встановлюється, коли внутрішній пристрій ініціює з'єднання.

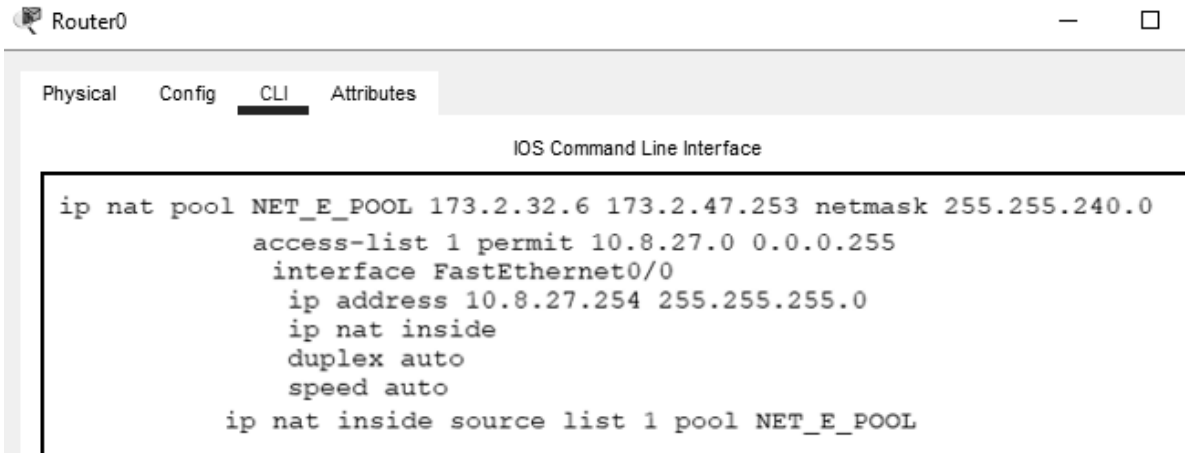


Рисунок 3.39 — Коректне налаштування динамічного NAT

Перевіримо працездатність трансляції, відправивши пакет з PC1 на Server4. Після цього переглянемо таблицю трансляцій на Router0 (Лістинг 3.5).

Лістинг 3.5 — Перевірка працездатності динамічного NAT

```
Pro  Inside global    Inside local      Outside local     Outside global
icmp 173.2.32.6:1     10.8.27.1:1       112.5.6.18:1      112.5.6.18:1
icmp 173.2.32.6:2     10.8.27.1:2       112.5.6.18:2      112.5.6.18:2
---  173.2.32.2       192.168.2.1       ---               ---
---  173.2.32.3       192.168.2.2       ---               ---
```

Далі коректно налаштуємо динамічний NAT із перекриттям для мережі NET B. В пул зовнішніх адрес введені дві перші адреси з мережі NET E, що зображено на рисунку 3.40. Зіставлення адрес встановлюється, коли внутрішній пристрій ініціює з'єднання.

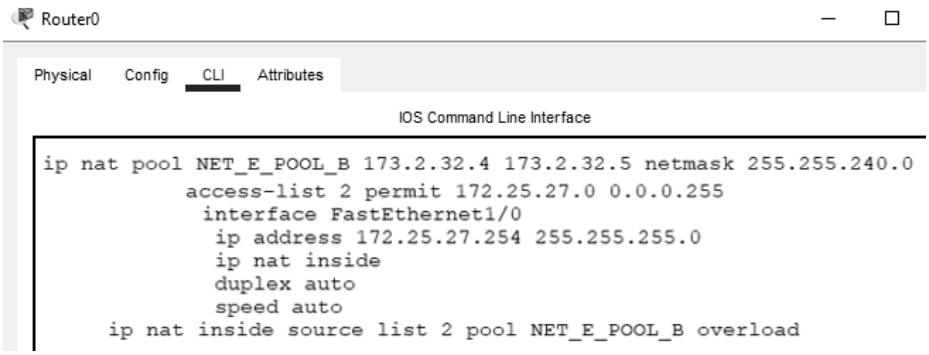


Рисунок 3.40 — Коректне налаштування динамічного NAT із перекриттям

Далі коректно налаштуємо PAT для мережі NET C, що зображено на рисунку 3.41. Це дозволяє кінцевому пристрою внутрішньої мережі використовувати одну або кілька глобальних IP-адрес через підключення до глобальної частини мережі, завдяки тому, що кожне з'єднання ідентифікується не лише IP-адресою, а й номером порту.

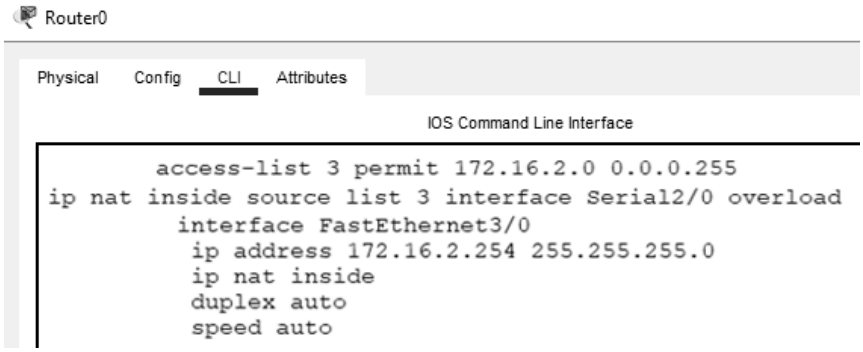


Рисунок 3.41 — Коректне налаштування PAT

Тепер після визначення базових коректних налаштувань DHCP та NAT в наведеній комп'ютерній мережі, розпочинається етап навмисного внесення конфігураційних помилок, які моделюють типові збої у роботі DHCP та NAT.

Одним із найтиповіших сценаріїв порушення коректної роботи DHCP є неправильне визначення виключених адрес, коли маршрутизатор отримує можливість видавати клієнтам адресу шлюза, що призводить до конфлікту і втрати доступності мережі для окремих хостів. Симуляція цієї проблеми наведена на рисунку 3.42.

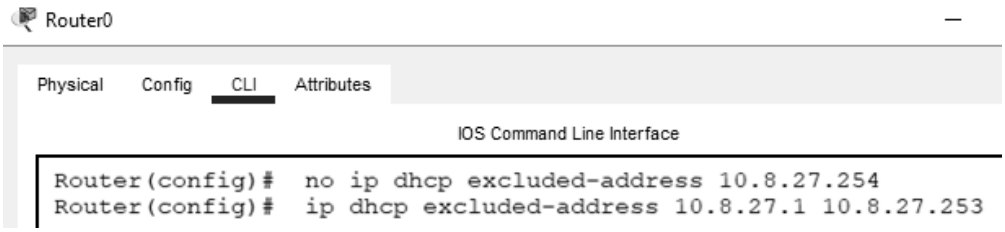


Рисунок 3.42 — Симуляція проблеми відсутності доступних адрес в пулі

Таким чином, практично весь пул адрес від 10.8.27.1 до 10.8.27.253 опиниться у списку заборонених для роздачі, і DHCP-сервер зможе призначити клієнтам лише одну адресу — 10.8.27.254, яка вже використовується інтерфейсом маршрутизатора як шлюз за замовчуванням. Наслідком цього буде те, що всі хости в мережі NET A залишаться без належної IP-конфігурації і згенерують автоматичні адреси з діапазону APIPA (169.254.0.0/16).

Для налаштування NAT типовою є помилка неправильного призначення NAT inside та outside на інтерфейсах. Наприклад, замінимо усі внутрішні інтерфейси на outside, а зовнішній — на inside, що наведено на рисунку 3.43.

Рисунок 3.43 — Симуляція помилки із визначенням NAT inside/outside

У цьому випадку критичним є неправильне співставлення адрес, що призведе до втрати доступності усіх хостів з боку Інтернету, хоч всередині локальної частини мережі вони залишатимуться досяжними.

Також для налаштування NAT типовою помилкою є некоректне

налаштування ACL, які співставляються із NAT-пулами. Наприклад, змінимо номер access-list 2 на access-list 4, що зображено на рисунку 3.44.

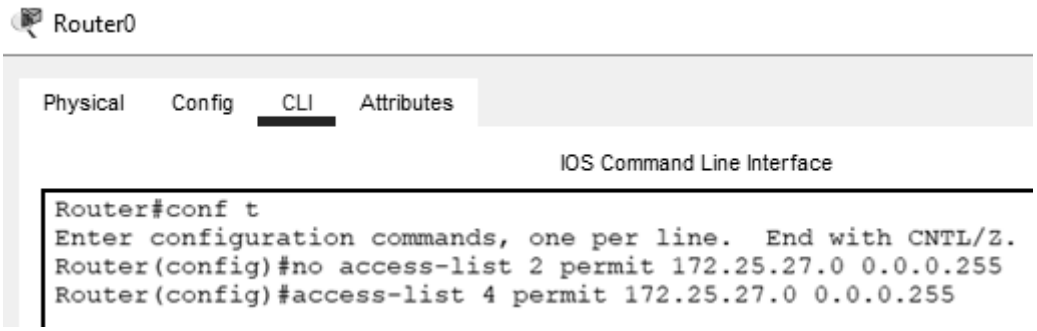


Рисунок 3.44 — Симуляція помилки із некоректним ACL

Оскільки access-list 2 більше не існує, а NET_E_POOL_B співставляється саме із ним, то маршрутизатор не зможе визначити, який трафік підлягає трансляції, що призведе до втрати можливості ініціювати з'єднання із пристроями, що знаходяться в глобальній частині мережі.

Після навмисного внесення некоректних змін у конфігурацію DHCP та NAT у спроєктованій комп'ютерній мережі, наступним етапом четвертого експерименту стає застосування методології prompt-engineering для діагностики виявлених проблем за допомогою розглянутих LLM.

Приклад структурованого текстового запиту для цього випадку (prompt): "Я змоделював комп'ютерну мережу, в якій використовується DHCP для автоматичної конфігурації робочих станцій у кількох підмережах та реалізований NAT різних типів (динамічний, динамічний з перекриттям, PAT) для забезпечення доступу до зовнішніх ресурсів. Під час налаштування я зіштовхнувся з низкою проблем, що проявляються у вигляді дивних симптомів. Один із клієнтів не може отримати IP-адресу динамічно, натомість генерується автоматична адреса з діапазону APIPA. Окрім цього, після реалізації трансляції адрес усі внутрішні пристрої втратили можливість виходу в Інтернет, хоча пінг до шлюзу доступний і локальна маршрутизація працює. Поясни, які можливі причини описаних проблем, які саме налаштування могли призвести до такої ситуації, і що варто перевірити в конфігураціях, щоб усунути проблему. Дай

коротку та структуровану відповідь у вигляді таблиці.”.

Імітаційний текстовий запит було подано до кількох розглянутих LLM. На рисунку 3.45 наведено результати діагностики проблем.

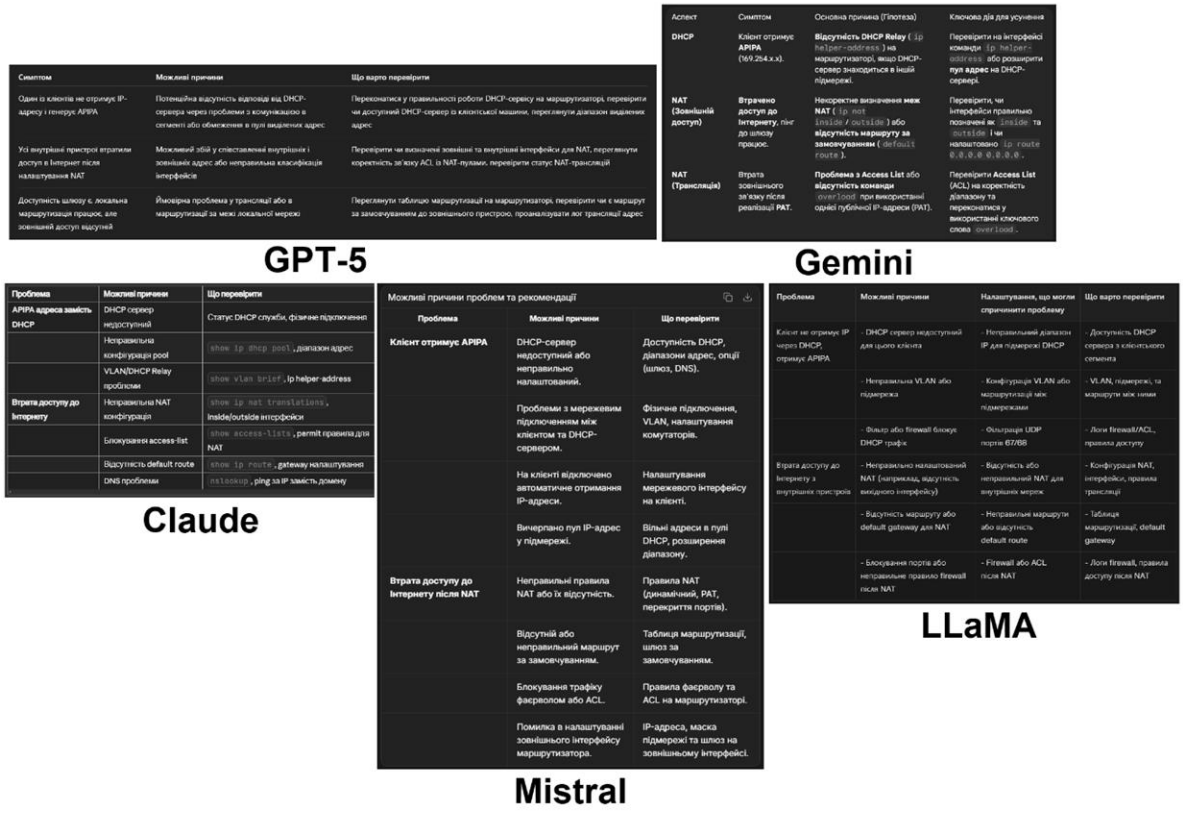


Рисунок 3.45 — Результати мережевої діагностики за використання різних LLM

На основі отриманих відповідей можна зробити висновок, що наведені LLM проявляють різний рівень ефективності у діагностиці мережевих проблем, пов’язаних із конфігурацією DHCP та NAT.

Модель GPT-5 поєднує достатній рівень деталізації з чіткою структурою подання матеріалу. Відповідь є компактною, однак охоплює ключові аспекти — від перевірки стану DHCP-сервера та правильності налаштування інтерфейсів до діагностики NAT і маршрутизації.

Модель Gemini (2.5 Flash) орієнтується на докладний опис окремих етапів діагностики, але його відповідь виявилася надмірно розгорнутою й менш компактною, що ускладнює сприйняття інформації в умовах практичного

застосування.

Модель Claude (Sonnet 4) пропонує відповідь у більш фрагментованій і формалізованій формі і робить акцент на відокремленні кожного симптому від його можливої причини. Такий підхід є корисним для систематизації даних, але водночас спостерігається тенденція до занадто прямолінійних рішень, що звужує поле аналізу.

Модель LLaMA (3.3) демонструє значну розгалуженість викладу з акцентом на деталізацію та розширене пояснення. Проте через надмірну кількість факторів і варіантів рішення відповідь виглядає менш цілеспрямованою, ніж у GPT-5. Такий підхід можна розглядати як сильну сторону для навчальних цілей.

Модель Mistral (Large 2), навпаки, прагне до максимальної стислості й узагальненості. Рішення подано у вигляді компактних формулювань без глибокого пояснення механізмів виникнення проблем. Попри це, у відповіді чітко відображається причинно-наслідковий зв’язок між неправильною конфігурацією NAT, відсутністю доступу до Інтернету та особливостями DHCP.

3.1.5 Симуляція проблем конфігурування Frame Relay

В ході проведення п’ятого експерименту для дослідження ефективності траблшутінгу в комп’ютерних мережах за використання великих мовних моделей, розглядається виявлення характерних помилок, які виникають у процесі конфігурування протоколу канального рівня Frame Relay в умовах топології Partial Mesh.

Для дослідження було змодельовано комп’ютерну мережу, що зображена на рисунку 3.46, в якій реалізовано підхід емуляції глобальної мережі. Для емуляції було використано елемент Cloud-PT, який надає середовище Cisco Packet Tracer. Використання цього елементу разом із протоколом Frame Relay відтворює модель, коли локальні маршрутизатори взаємодіють не безпосередньо, а через умовного віртуального провайдера послуг. У цьому випадку “хмара” імітує обладнання оператора, яке забезпечує віртуальні канали

на базі комутації пакетів у глобальному середовищі.

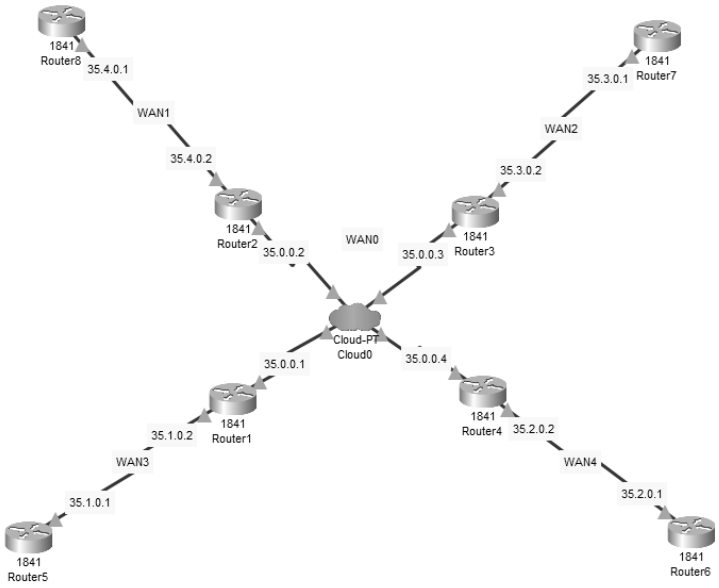


Рисунок 3.46 — Змодельована схема мережі

Побудована мережа із використанням Frame Relay відображає поєднання архітектур Hub and Spoke та Full Mesh, що дозволяє відтворити різні сценарії взаємодії між маршрутизаторами та проаналізувати вплив некоректних налаштувань на працездатність усієї системи. Центральним вузлом виступає маршрутизатор Router1, який виконує роль концентратора у класичній топології Hub and Spoke, тоді як маршрутизатори Router3 і Router4 з'єднані між собою за принципом Full Mesh.

Для кожного сегмента мережі було виділено окремий адресний простір: WAN0 із мережею 35.0.0.0, WAN1 із мережею 35.1.0.0, WAN2 із мережею 35.2.0.0, WAN3 із мережею 35.3.0.0 та WAN4 із мережею 35.4.0.0. Взаємодія між усіма іншими вузлами (Router5, Router6, Router7, Router8), забезпечується за допомогою статичної маршрутизації.

Окрім побудови основної інфраструктури, в мережі було реалізовано різні варіанти автентифікації для протоколу канального рівня PPP (Point-to-Point Protocol). Для WAN1 та WAN3 використано автентифікацію PAP, тоді як для WAN2 застосовано автентифікацію за допомогою CHAP.

Спочатку проведемо огляд технічно-коректних початкових налаштувань

технології глобальних мереж Frame Relay та автентифікації для PPP між маршрутизаторами. Після цього буде проведено симуляцію можливих конфігураційних проблем, які можуть виникати під час налаштування Frame Relay та автентифікації.

Для коректного налаштування Frame Relay спочатку потрібно налаштувати DLCI для з'єднання між вузлами. Для цього потрібно перейти до елемента Cloud-PT та здійснити прив'язку локальних ідентифікаторів DLCI до відповідних фізичних інтерфейсів, що зображено на рисунку 3.47.

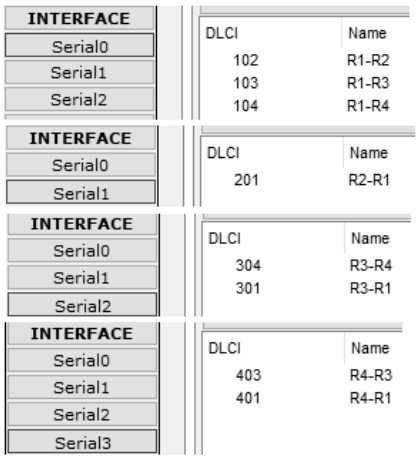


Рисунок 3.47 — Налаштування DLCI

Далі потрібно створити mapping DLCI на інтерфейсах, щоб було відомо, куди потрібно пересилати кадри. Frame Relay комутатор потребує таблиці, яка зіставляє DLCI з відповідними портами, що зображено на рисунку 3.48.

Frame Relay

Serial0	R1-R2	<-->	Serial0	R1-R2
Port	Sublink		Port	Sublink

	From Port	Sublink	To Port	Sublink
1	Serial0	R1-R2	Serial1	R2-R1
2	Serial0	R1-R3	Serial2	R3-R1
3	Serial2	R3-R4	Serial3	R4-R3
4	Serial3	R4-R1	Serial0	R1-R4

Рисунок 3.48 — Таблиця зіставлення DLCI із портами

Далі на маршрутизаторах потрібно встановити інкапсуляцію Frame Relay

на відповідних інтерфейсах, через які маршрутизатори об’єднуються за допомогою глобальної мережі. На рисунку 3.49 наведено налаштування для маршрутизаторів: Router2, Router3, Router4.

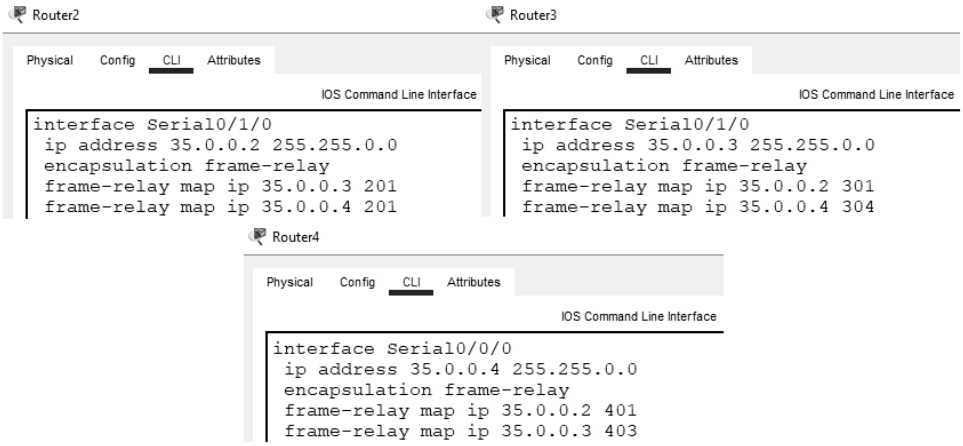


Рисунок 3.49 — Налаштування інкапсуляції Frame Relay на маршрутизаторах

На маршрутизаторі Router1 достатньо лише дати команду “encapsulation frame-relay”, оскільки він є центральним вузлом в топології Hub and Spoke, через який обов’язково буде проходити увесь трафік з інших маршрутизаторів (окрім Router3 та Router4, для яких трафік буде проходити за принципом Full Mesh). Вміст конфігураційного файлу маршрутизатора Router1 наведено в Додатку Е.

Далі налаштуємо статичну маршрутизацію на кожному маршрутизаторі для мережевої взаємодії між віддаленими вузлами мережі (Router5, Router6, Router7, Router8). На рисунку 3.50 наведено фрагмент конфігураційного файлу із записом про налаштування статичної маршрутизації для Router8.

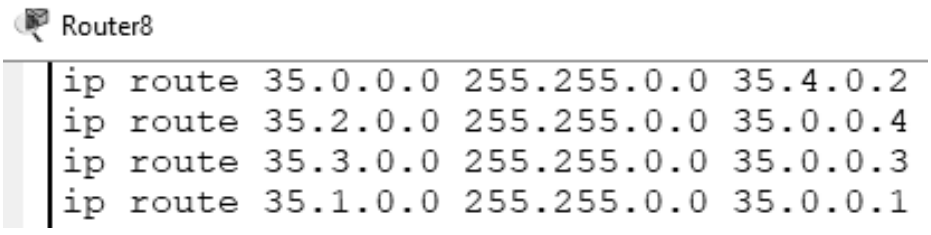


Рисунок 3.50 — Приклад налаштування статичної маршрутизації

Далі перевіримо працездатність мережі за допомогою команди “traceroute”, яку ми дамо на маршрутизаторах Router7 та Router8 для демонстрації шляху проходження пакета до маршрутизатора Router6, що зображено на рисунку 3.51.

Router7

```
Router>en
Router#traceroute 35.2.0.1
Type escape sequence to abort.
Tracing the route to 35.2.0.1

  1   35.3.0.2        9 msec    6 msec    5 msec
  2   35.0.0.4        23 msec   19 msec   15 msec
  3   35.2.0.1        18 msec   26 msec   20 msec
```

Router8

```
Router>en
Router#traceroute 35.2.0.1
Type escape sequence to abort.
Tracing the route to 35.2.0.1

  1   35.4.0.2        6 msec    0 msec    5 msec
  2   35.0.0.1        19 msec   9 msec    17 msec
  3   35.0.0.4        18 msec   31 msec   21 msec
  4   35.2.0.1        24 msec   30 msec   19 msec
```

Рисунок 3.51 — Демонстрація шляху проходження пакета до Router6 у двох випадках

Можемо переконатись, що у випадку із Router7, пакет проходить менший шлях, оскільки між Router3 та Router4 налаштований зв’язок за принципом Full Mesh. У випадку із Router8, пакет має обов’язково проходити через Router1, який є центральним вузлом топології Hub and Spoke. Отже, зв’язок в побудованій мережі працює за принципом топології Partial Mesh із використанням комутації за допомогою Frame Relay.

Далі проведемо коректні налаштування автентифікації між маршрутизаторами за допомогою PPP із PAP (Password Authentication Protocol) та CHAP (Challenge-Handshake Authentication Protocol). На рисунку 3.52 наведено фрагмент конфігураційного файлу Router7 (із налаштуванням автентифікації PAP) та конфігураційного файлу Router8 (із налаштуванням автентифікації CHAP).

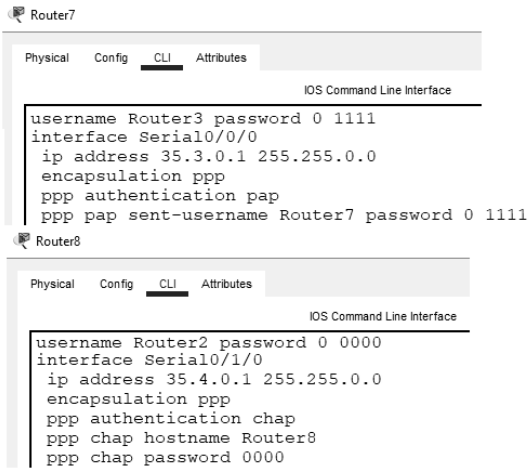


Рисунок 3.52 — Приклади налаштування автентифікації PAP та CHAP

Автентифікація між маршрутизаторами за допомогою PAP чи CHAP у протоколі PPP використовується для підтвердження достовірності сторін перед тим, як з'єднання буде вважатися встановленим. Це є важливим в умовах WAN, коли маршрутизатори підключаються через середовище, яке контролює провайдер, і є ризик несанкціонованого доступу. PAP реалізує просту автентифікацію шляхом передачі логіна і пароля у відкритому вигляді під час встановлення PPP-сесії. CHAP працює більш захищено, оскільки замість прямої передачі пароля використовує алгоритм перевірки через хешування.

Перевірка працездатності автентифікації здійснюється кількома способами. По-перше, якщо автентифікація між маршрутизаторами була пройдена успішно, то інтерфейси між ними перейдуть у стан "up". По-друге, можна використати команду "show interface" для перегляду стану PPP на інтерфейсі, що зображено на рисунку 3.53 (приклад для Router8).

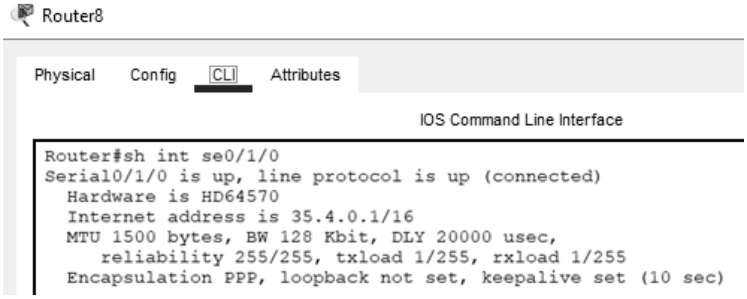


Рисунок 3.53 — Перевірка працездатності автентифікації

Тепер після визначення базових коректних налаштувань Frame Relay та автентифікації за допомогою PPP в наведеній комп'ютерній мережі, розпочинається етап навмисного внесення конфігураційних помилок, які моделюють типові збої у роботі цих протоколів.

Найбільш показовим для Frame Relay є сценарій некоректного налаштування зіставлення DLCI із портами. Наприклад, на Router3 і Router4 для налаштувань інтерфейсів відмінимо команди "frame-relay map ip" та на елементі Cloud-PT видалимо запис DLCI з фізичними портами (R3-R4 та R4-R3), що зруйнує механізм логічної адресації і призведе до втрати зв'язку між Router3 та Router4. Після цього передамо ICMP-пакет від Router6 до Router7 в режимі симуляції і переглянемо PDU Information в момент проходження пакета через Cloud-PT, що зображено на рисунку 3.54.

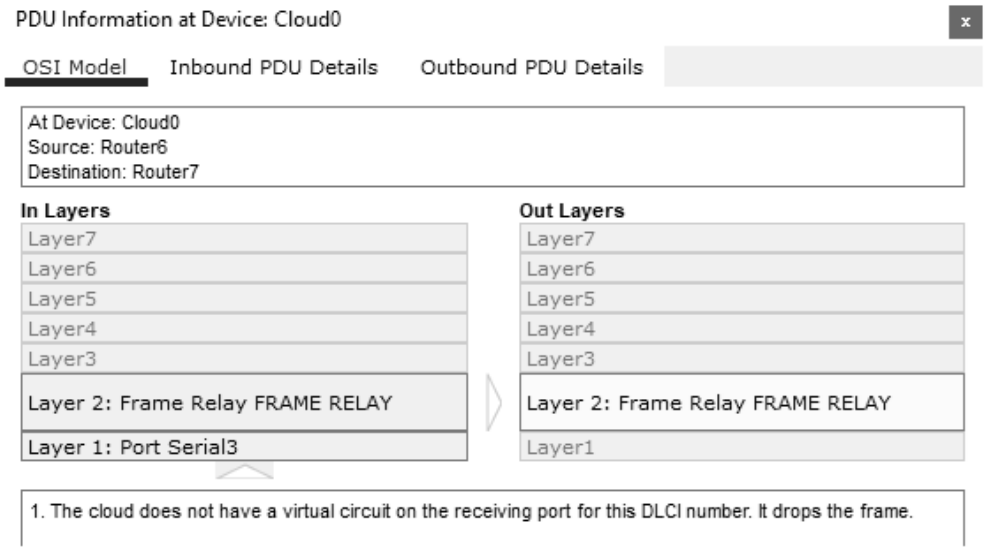


Рисунок 3.54 — Демонстрація порушення працездатності мережі через помилки зіставлення DLCI із портами

У випадку із автентифікацією найбільш показовим випадком буде відміна інкапсуляції PPP на одному із маршрутизаторів, що призведе до втрати зв'язку між парою маршрутизаторів, для яких з обох боків налаштована автентифікація. В цьому випадку протокол лінії зв'язку маршрутизатора перейде в режим "down" і не зможе встановити з'єднання із сусідом, що зображено на рисунку 3.55.

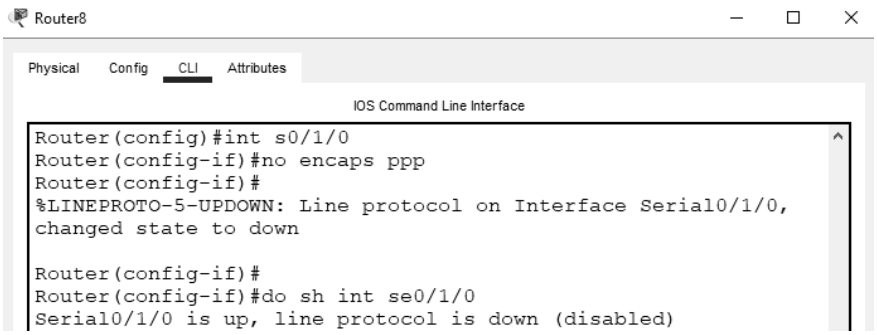


Рисунок 3.55 — Демонстрація порушення автентифікації CHAP на маршрутизаторі Router8

Після навмисного внесення некоректних змін у конфігурацію Frame Relay та автентифікації у спроєктованій мережі, наступним етапом п’ятого експерименту стає застосування методології prompt-engineering для діагностики виявлених проблем за допомогою розглянутих LLM.

Приклад структурованого текстового запиту для цього випадку (prompt): “Я працюю над побудовою мережі, що складається із маршрутизаторів (Router1 — Router8) із використанням Frame Relay у топології Partial Mesh, де між деякими маршрутизаторами (Router3, Router4) є прямі з’єднання, а решта взаємодіє через центральний маршрутизатор (Router1). Чотири маршрутизатори беруть безпосередню участь у комутації через Frame Relay. Інші чотири маршрутизатори — є віддаленими. Для зв’язку між віддаленими маршрутизаторами в мережі було налаштовано статичну маршрутизацію. Під час тестування конфігурації зіткнувся з тим, що зв’язку між Router3 та Router4 немає (вони з’єднані через Frame Relay безпосередньо). В усіх інших випадках — взаємодія між маршрутизаторами відбувається без проблем. Також в моїй мережі налаштована автентифікація за допомогою PPP. Однак, на маршрутизаторі Router8 (CHAP), який з’єднаний із Router2 безпосередньо, чомусь отримую помилку: Serial0/1/0 is up, line protocol is down (disabled). Через що я не маю зв’язку із Router2. Поясни, які можливі причини описаних проблем, які саме налаштування могли призвести до такої ситуації, і що варто перевірити в конфігураціях, щоб усунути проблему. Дай коротку та структуровану відповідь

у вигляді таблиці.”.

Імітаційний текстовий запит було подано до кількох розглянутих LLM. На рисунку 3.56 наведено результати діагностики проблем.

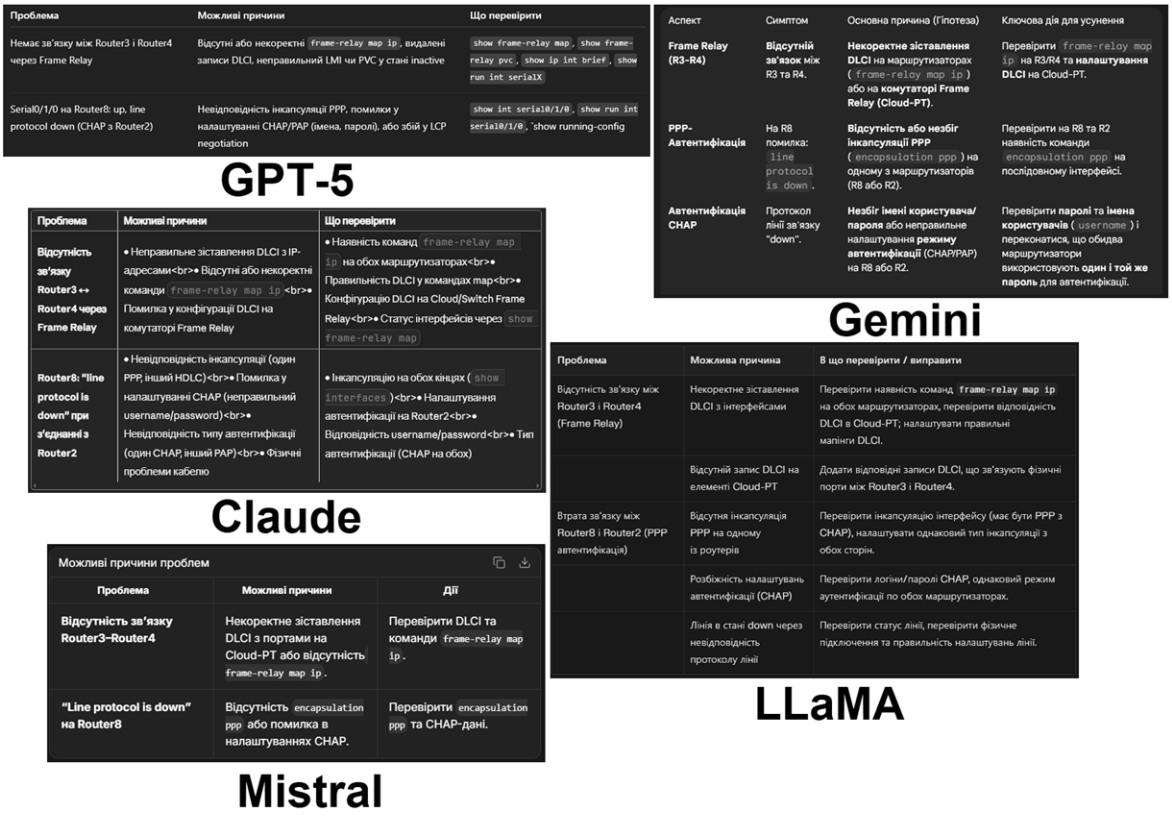


Рисунок 3.56 — Результати мережевої діагностики за використання різних LLM

На основі отриманих відповідей можна зробити висновок, що наведені LLM проявляють різний рівень ефективності у діагностиці мережевих проблем, пов’язаних із конфігурацією Frame Relay та автентифікації.

Модель GPT-5 дала збалансовану відповідь, поєднавши симптоми, можливі причини та конкретні команди перевірки. Це дозволяє не лише описати проблему, а й створює покроковий алгоритм дій адміністратора, що наближує результат до реальної практики усунення конфігураційних проблем.

Модель Gemini (2.5 Flash) демонструє схожий підхід, однак у його відповіді спостерігається надмірна деталізація, що знижує зручність використання через високий рівень фрагментації інформації.

У випадку моделі Claude (Sonnet 4) спостерігається влучне відображення ключових причин, однак бракує повноти, особливо щодо інтерпретації діагностичних команд та відображення взаємозв’язку між симптомами і конфігураційними похибками.

Відповіді LLaMA (3.3) виглядають дещо спрощеними, вони концентруються на окремих причинах, але не створюють цілісної діагностичної картини, що знижує їхню практичну цінність у складних сценаріях.

Модель Mistral (Large 2) демонструє найбільш узагальнену відповідь, яка придатна для базового діагностування, але відсутні конкретні вказівки щодо перевірки та виправлення конфігурацій.

3.2 Порівняльний аналіз результатів роботи різних LLM

Після проведення п’яти експериментальних досліджень, було отримано емпіричний масив даних, представлений відповідями п’яти великих мовних моделей (GPT-5, Gemini (2.5 Flash), Claude (Sonnet 4), LLaMA (3.3), Mistral (Large 2)). Кожен експеримент був ініційований структурованим запитом, що містив опис симульованих конфігураційних несправностей, а вихідні дані LLM були представлені у формі діагностичних таблиць.

В цьому підрозділі буде розраховано кількісні оцінки ефективності розглянутих LLM у завданнях мережевої діагностики для кожного експериментального сценарію. Для розрахунку узагальненої метрики ефективності скористаємось формулою (2.6). За вагові коефіцієнти приймаємо значення омега: $\omega_1=0.5$, $\omega_2=0.3$. Оскільки акцент робиться на точності діагностики, трохи менше — на часі відповіді. Для нормалізації було обрано максимальний час $T_{max} = 10$ с.

Далі визначимо значення двох параметрів для розглянутих LLM в усіх експериментальних випадках згідно із формулою (2.4) та формулою (2.5) на основі якісних характеристик ефективності та часу відповіді. Після цього занесемо дані до таблиці 3.2.

Таблиця 3.2 — Значення параметрів для оцінки ефективності

№	LLM	Accuracy	$T_{response}$
1	GPT-5	0.75	8
	Gemini (2.5 Flash)	0.9	6
	Claude (Sonnet 4)	0.85	7
	LLaMA (3.3)	0.65	7
	Mistral (Large 2)	0.7	5
2	GPT-5	0.9	7
	Gemini (2.5 Flash)	0.85	6
	Claude (Sonnet 4)	0.8	7
	LLaMA (3.3)	0.65	6
	Mistral (Large 2)	0.7	5
3	GPT-5	0.88	7
	Gemini (2.5 Flash)	0.85	6
	Claude (Sonnet 4)	0.8	7
	LLaMA (3.3)	0.7	6
	Mistral (Large 2)	0.68	5
4	GPT-5	0.88	6
	Gemini (2.5 Flash)	0.85	8
	Claude (Sonnet 4)	0.8	6
	LLaMA (3.3)	0.75	8
	Mistral (Large 2)	0.7	5
5	GPT-5	0.9	7
	Gemini (2.5 Flash)	0.88	8
	Claude (Sonnet 4)	0.82	7
	LLaMA (3.3)	0.7	6
	Mistral (Large 2)	0.65	5

Далі розрахуємо загальну ефективність (E), як середньозважене значення за всіма критеріями для розглянутих LLM в усіх експериментальних випадках згідно із формулою (2.6). Після цього занесемо дані до таблиці 3.3.

Таблиця 3.3 — Загальна ефективність LLM для усіх експериментальних випадків

№	GPT-5	Gemini (2.5 Flash)	Claude (Sonnet 4)	LLaMA (3.3)	Mistral (Large 2)
1	0.435	0.57	0.515	0.415	0.5
2	0.54	0.545	0.49	0.445	0.5
3	0.53	0.545	0.49	0.47	0.49
4	0.56	0.485	0.52	0.435	0.5
5	0.54	0.5	0.5	0.47	0.475

Розрахуємо середню ефективність (E_{avg}) для кожної LLM. Для цього додамо розраховані значення ефективності для кожного експерименту та поділимо на кількість експериментів. Після цього занесемо дані до таблиці 3.4.

Таблиця 3.4 — Середня ефективність кожної LLM для усіх проведених експериментів

LLM	Середня ефективність (E_{avg})
GPT-5	0.521
Gemini (2.5 Flash)	0.529
Claude (Sonnet 4)	0.503
LLaMA (3.3)	0.447
Mistral (Large 2)	0.493

На основі розрахунків загальної та середньої ефективності LLM, можна сформулювати висновки щодо їхньої продуктивності у сфері траблшутінгу в комп’ютерних мережах.

Найвищий середній показник ефективності продемонструвала модель Gemini (2.5 Flash), що свідчить про її здатність забезпечувати найвищий рівень коректності відповідей серед досліджених моделей. Цей результат позиціонує Gemini, як найбільш ефективний інструмент підтримки прийняття рішень для мережевого інженера.

Модель GPT-5 продемонструвала близьке до Gemini (2.5 Flash) значення середньої ефективності, продемонструвавши високу узгодженість результатів у більшості експериментальних сценаріїв, зокрема при складних конфігураційних умовах.

Моделі Claude (Sonnet 4) та Mistral (Large 2) показали середній рівень ефективності. Вони здатні забезпечити прийнятну точність, проте схильні до варіативності результатів залежно від структури вхідного запиту.

Найнижчий рівень ефективності продемонструвала LLaMA (3.3), що може свідчити про дещо обмежену здатність даної моделі до глибокого контекстного аналізу в завданнях траблшутінгу в комп'ютерних мережах.

3.3 Формування алгоритму побудови промптів для мережевої діагностики

У результаті проведення п'яти експериментів, було визначено оптимальну структуру запитів (промптів) до великих мовних моделей (LLM), що забезпечує найвищу точність та стабільність результатів під час виконання мережевої діагностики.

На основі аналізу ефективності відповідей різних LLM було сформовано узагальнений алгоритм побудови промптів, який поєднує формалізований технічний опис проблеми із контекстною деталізацією параметрів мережевої конфігурації:

— опис контексту середовища;

— визначення симптомів або проблеми;

— уточнення конфігураційних параметрів;

— визначення цілі відповіді;

— опціональне уточнення рівня деталізації.

Спочатку потрібно вказати тип мережевої інфраструктури, роль мережевих пристроїв та логічні зв'язки між ними. Це забезпечить моделі базовий контекст і дозволить коректно інтерпретувати подальші команди.

Далі потрібно описати спостережувану проблему, наприклад: "немає

зв’язку між вузлами мережі NET A та NET B”. Це дозволить LLM обрати релевантний набір діагностичних кроків.

Далі потрібно подати ключові фрагменти конфігурацій, що можуть бути джерелом помилки (команди Cisco IOS, IP-адреси, VLAN ID, ACL, DLCI тощо). Це підвищить точність інтерпретації і зменшить ймовірність хибних припущень.

Далі потрібно задати очікуваний формат результату, наприклад: “наведи можливі причини і способи усунення, створи послідовність перевірок”. Це спрямує модель на практичний результат, а не на загальний опис теорії.

В кінці запиту потрібно вказати, чи потрібна коротка діагностика, чи детальна інструкція. Що забезпечить баланс між глибиною аналізу і компактністю відповіді.

Запропонований алгоритм базується на принципах контекстного підкріплення та інструкційної структуризації, що є оптимальним для LLM у задачах траблшутінгу в комп’ютерних мережах. Блок-схему алгоритму побудови промптів для мережевої діагностики наведено в Додатку Ж.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання наукового ефекту

Основними та важливими ознаками наукового ефекту науково-дослідної роботи є новизна роботи, рівень її теоретичного опрацювання, перспективність, рівень розповсюдження результатів, можливість реалізації. Науковий ефект НДР на тему “Технології пошуку несправностей в сучасних комп’ютерних мережах за використання великих мовних моделей”.

Впродовж написання роботи було обрано експертів для оцінки та визначення рівня ступеня новизни з Вінницького національного технічного університету, кафедра обчислювальної техніки: к.т.н., проф. Захарченко С.М., к.т.н., доц. каф. Богомолов С.В., к.т.н., доц. Тарновський М.Г.

Значення зазначених критеріїв у балах, що дозволяє кількісно оцінити значущість та унікальність проведеної роботи, представлені у таблицях 4.1 та 4.2.

Таблиця 4.1 — Показники ступеня новизни науково-дослідної роботи виставлені експертами

Ступінь новизни	Характеристика ступеня новизни	Значення ступеня новизни, бали		
		Експерти (ПІБ)		
		Захарченко С.М.	Богомолов С.В.	Тарновський М.Г.
Принципово нова	Робота якісно нова за постановкою задачі і ґрунтується на застосуванні оригінальних методів дослідження. Результати дослідження відкривають новий напрям в цій галузі науки і техніки. Отримано принципово нові факти, закономірності; розроблено нову теорію. Створено принципово новий пристрій, спосіб, метод	0	0	0
Нова	Отримано нову інформацію, яка суттєво зменшує невизначеність наявних значень (по-новому або вперше пояснено відомі факти, закономірності, впроваджено нові поняття, розкрито структуру змісту). Проведено суттєве вдосконалення, доповнення і уточнення раніше досягнутих результатів	59	58	63

Продовження таблиці 4.1

Ступінь новизни	Характеристика ступеня новизни	Значення ступеня новизни, бали		
		Експерти (ПІБ)		
		Захарченко С.М.	Богомолов С.В.	Тарновський М.Г.
Відносно нова	Робота має елементи новизни в постановці задачі і методах дослідження. Результати дослідження систематизують і узагальнюють наявну інформацію, визначають шляхи подальших досліджень; вперше знайдено зв'язок (або знайдено новий зв'язок) між явищами. В принципі, відомі положення поширено на велику кількість об'єктів, в результаті чого знайдено ефективне рішення. Розроблено більш прості способи для досягнення відомих результатів. Проведено часткову раціональну модифікацію (з ознаками новизни)	0	0	0
Традиційна	Робота виконана за традиційною методикою. Результати дослідження мають інформаційний характер. Підтверджені або поставлені під сумнів відомі факти та твердження, які потребують перевірки. Знайдено новий варіант рішення, який не дає суттєвих переваг в порівнянні з існуючим	0	0	0
Не нова	Отримано результат, який раніше зафіксований в інформаційному полі, та не був відомий авторам	0	0	0
Середнє значення балів експертів		60		

Залежно від середнього значення визначених експертних балів ступінь новизни характеризується як відносно нова, тобто отримана нова інформація, яка систематизує та узагальнює наявну інформацію.

Таблиця 4.2 — Показники рівня теоретичного опрацювання науково-дослідної роботи виставлені експертами

Характеристика рівня теоретичного опрацювання	Значення показника рівня теоретичного опрацювання, бали		
	Експерти (ПІБ)		
	Захарченко С.М.	Богомолов С.В.	Тарновський М.Г.
Відкриття закону, розробка теорії	0	0	0
Глибоке опрацювання проблеми: багатоаспектний аналіз зв'язків, взаємозалежності між фактами з наявністю пояснень, наукової систематизації з побудовою евристичної моделі або комплексного прогнозу	0	0	0
Розробка способу (алгоритму, програми), пристрою, отримання нової речовини	57	60	63

Продовження таблиці 4.2

Характеристика рівня теоретичного опрацювання	Значення показника рівня теоретичного опрацювання, бали		
	Експерти (ПІБ)		
	Захарченко С.М.	Богомолов С.В.	Тарновський М.Г.
Елементарний аналіз зв'язків між фактами та наявною гіпотезою, класифікація, практичні рекомендації для окремого випадку тощо	0	0	0
Опис окремих елементарних фактів, викладення досвіду, результатів спостережень, вимірювань тощо	0	0	0
Середнє значення балів експертів	60		

Згідно отриманого середнього значення балів експертів рівень теоретичного опрацювання науково-дослідної роботи характеризується як глибоке опрацювання проблеми: багатоаспектний аналіз зв'язків, взаємозалежності між фактами з наявністю пояснень, наукової систематизації з побудовою евристичної моделі або комплексного прогнозу.

Показник, який характеризує рівень наукового ефекту, визначаємо за формулою [40]:

$$E_{нау} = 0{,}6\ k_{нов} + 0{,}4\ k_{теор}\,, \qquad (4.1)$$

де $k_{нов}$, $k_{теор}$ — показники ступеня новизни та рівня теореного опрацювання науково-дослідної роботи, де $k_{нов} = 60$, $k_{теор} = 60$ балів;

0,6 та 0,4 — питома вага (значимість) показників ступеня новизни та рівня теоретичного опрацювання науково-дослідної роботи.

$$E_{нау} = 0{,}6 \cdot 60 + 0{,}4 \cdot 60 = 60 \text{ балів}$$

Визначення характеристики показника $E_{нау}$ ипроводиться на основі висновків експертів виходячи з граничних значень, які наведені в таблиці 4.3.

Таблиця 4.3 — Граничні значення показника наукового ефекту

Досягнутий рівень показника	Кількість балів
Високий	70…100
Середній	50…69
Достатній	15…49
Низький (помилкові дослідження)	1…14

Відповідно до визначеного рівня наукового ефекту проведеної науково-дослідної роботи на тему “Технології пошуку несправностей в сучасних комп’ютерних мережах за використання великих мовних моделей”, даний рівень становить 60 балів і відповідає статусу — середній рівень. Тобто у даному випадку можна вести мову про потенційну фактичну ефективність науково-дослідної роботи.

4.2 Прогнозування витрат на виконання науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи групуються за такими статтями: витрати на оплату праці, витрати на соціальні заходи, матеріали, паливо та енергія для науково-виробничих цілей, витрати на службові відрядження, програмне забезпечення для наукових робіт, інші витрати, накладні витрати.

1. Основна заробітна плата кожного із дослідників $З_О$, якщо вони працюють в наукових установах бюджетної сфери визначається за формулою:

$$З_О = \frac{М}{Т_р} \cdot t \text{ (грн)} \qquad (4.2)$$

де M — місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

T_P — число робочих днів в місяці; приблизно $T_P \approx 21...23$ дні;

t — число робочих днів роботи дослідника.

Зведемо сумарні розрахунки до таблиці 4.4.

Таблиця 4.4 — Заробітна плата дослідника в науковій установі бюджетної сфери

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату грн.
1. Мережевий інженер	30000	1428,6	5	7143
2. Технічний аналітик	28000	1333,3	21	28000
3. Тестувальник (QA-інженер)	25000	1190,5	20	23810
Всього				58952

2. Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт розраховують за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i, \quad (4.3)$$

де C_i — погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i — час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{М_м \cdot К_і \cdot К_с}{Т_р \cdot t_{зм}}, \quad (4.4)$$

де $М_М$ — розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), грн;

$К_і$ — коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду;

$К_с$ — мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати;

$Т_р$ — середня кількість робочих днів в місяці, приблизно Тр = 21…23 дні;

$t_{зм}$ — тривалість зміни, год.

Витрати на основу заробітну плату робітників занесено до таблиці 4.5.

Таблиця 4.5 — Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Погодинна тарифна ставка, грн	Величина оплати на робітника, грн
1. Підготовчі роботи (постановка задачі)	3	2	52,4	157,1
3. Інтеграційні роботи	6	4	71,4	428,6
4. Проведення експериментальних досліджень	5	5	81,0	404,8
5. Розробка алгоритму	10	4	71,4	714,3
Всього				1704,8

3. Розрахунок додаткової заробітної плати робітників

Додаткова заробітна плата $З_д$ всіх розробників та робітників, які приймали устають в розробці нового технічного рішення розраховується як 10 - 12 % від основної заробітної плати робітників.

На даному підприємстві додаткова заробітна плата начисляється в розмірі 11% від основної заробітної плати.

$$З_д = (З_о + З_р) \cdot \frac{Н_{доп}}{100\%} \quad (4.5)$$

$$З_д = 0{,}11 \cdot (58952 + 1704{,}8) = 6672{,}29 \text{ (грн)}$$

4. Нарахування на заробітну плату $Н_{ЗП}$ дослідників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою (4.10):

$$Н_{ЗП} = (З_о + З_р + З_д) \cdot \frac{\beta}{100} \text{ (грн)} \quad (4.6)$$

де $З_о$ — основна заробітна плата розробників, грн.;

$З_д$ — додаткова заробітна плата всіх розробників та робітників, грн.;

$З_р$ — основну заробітну плату робітників, грн.;

β — ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % .

Дана діяльність відноситься до бюджетної сфери, тому ставка єдиного внеску на загальнообов'язкове державне соціальне страхування буде складати 22%, тоді:

$$Н_{ЗП} = (58952 + 1704{,}8 + 6672{,}29) \cdot \frac{22}{100} = 14812{,}47 \text{ (грн)}$$

5. Сировина та матеріали

До статті "Сировина та матеріали" належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби й предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за прямим призначенням згідно з нормами їх витрачання, а також витрачені придбані напівфабрикати, що підлягають монтажу або виготовленню й додатковій обробці в цій організації, чи дослідні зразки, що виготовляються виробниками за документацією наукової організації.

Витрати на матеріали (М) у вартісному вираженні розраховуються окремо для кожного виду матеріалів за формулою:

$$М = \sum_{i=1}^{n} H_j \cdot Ц_j \cdot К_j - \sum_{i=1}^{n} В_j \cdot Ц_{вj}, \quad (4.7)$$

де Нj — норма витрат матеріалу j-го найменування, кг;

n — кількість видів матеріалів;

Цj — вартість матеріалу j-го найменування, грн/кг;

Кj — коефіцієнт транспортних витрат, (Кj = 1,1 … 1,15);

Вj — маса відходів j-го найменування, кг;

Цвj — вартість відходів j-го найменування, грн/кг.

Проведені розрахунки зведені в таблицю 4.6.

Таблиця 4.6 — Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, шт	Вартість витраченого матеріалу, грн
Папір	180	0,5	90
Ручка	18	1	18
Блокнот	40	1	40
Флешка	280	1	280
З врахуванням коефіцієнта транспортування			470,8

6. Програмне забезпечення для наукових (експериментальних) робіт

Балансову вартість програмного забезпечення розраховують за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (4.8)$$

де $Ц_{iпрг}$ — ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.i}$ — кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i — коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, (K_i = *1,10…1,12);*

k — кількість найменувань програмних засобів.

Отримані результати необхідно занести до таблиці 4.7.

Таблиця 4.7 — Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Windows 10 Pro	1	6590	6590
Microsoft Office 365	1	2600	2600
Підписка на LLM (ChatGPT Plus)	2	840	1680
Всього з врахуванням налагодження			11957

7. Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.9)$$

де $Ц_б$ — балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ — термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ — строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

Проведені розрахунки необхідно занести до таблиці 4.8.

Таблиця 4.8 — Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
1. Персональний комп'ютер	28000	2	1	1166,67
2. Wi-Fi маршрутизатор	3200	2	1	133,33
3. Зовнішній SSD-накопичувач (1-2 ТБ)	2800	2	1	116,67
Всього				1416,67

8. До статті "Паливо та енергія для науково-виробничих цілей" відносяться витрати на всі види палива й енергії, що безпосередньо використовуються з технологічною метою на проведення досліджень.

$$В_е = \sum_{i=1}^{n} \frac{W_{yt} \cdot t_i \cdot Ц_е \cdot К_{впі}}{\eta_i} \quad (4.10)$$

де W_{yt} — встановлена потужність обладнання на певному етапі розробки, кВт;

t_i — тривалість роботи обладнання на етапі дослідження, год;

$Ц_е$ — вартість 1 кВт-години електроенергії, грн;

$К_{впі}$ — коефіцієнт, що враховує використання потужності, $К_{впі} < 1$;

η_i — коефіцієнт корисної дії обладнання, $\eta_i < 1$.

Для написання магістерської роботи використовується персональний комп'ютер для якого розрахуємо витрати на електроенергію.

$$В_е = \frac{0{,}5 \cdot 180 \cdot 12{,}69 \cdot 0{,}5}{0{,}8} = 713{,}81$$

9. Витрати за статтею “Службові відрядження” розраховуються як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{св} = (З_о + З_р) \cdot \frac{Н_{св}}{100\%}, \quad (4.11)$$

де $Н_{св}$ — норма нарахування за статтею “Службові відрядження”.

$$В_{св} = 0,2 \cdot (58952 + 1704,8) = 12131,43$$

10. Накладні (загальновиробничі) витрати $В_{нзв}$ охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Накладні (загальновиробничі) витрати $В_{нзв}$ можна прийняти як (100…150)% від суми основної заробітної плати розробників та робітників, які виконували дану МКНР, тобто:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.12)$$

де $Н_{нзв}$ — норма нарахування за статтею “Інші витрати”.

$$В_{нзв} = (58952 + 1704,8) \cdot \frac{100}{100\%} = 60657,14 \text{ грн}$$

Сума всіх попередніх статей витрат дає витрати, які безпосередньо стосуються даного розділу МКНР:

В = 58952 + 1704,8 + 6672,29 + 14812,47 + 470,8 + 11957 + 1416,67 + 713,81 + 12131,43 + 60657,14 = 169488,75 грн.

Прогнозування загальних втрат ЗВ на виконання та впровадження результатів виконаної МКНР здійснюється за формулою:

$$ЗВ = \frac{В}{\eta}, \quad (4.13)$$

де η — коефіцієнт, який характеризує стадію виконання даної НДР.

Оскільки, робота знаходиться на стадії науково-дослідних робіт, то коефіцієнт β = 0,3.

Звідси:

$$ЗВ = \frac{169488,75}{0,3} = 564962,51 \text{ грн.}$$

4.3 Оцінювання важливості та наукової значимості науково-дослідної роботи

Оцінювання та доведення ефективності виконання науково-дослідної роботи фундаментального чи пошукового характеру є достатньо складним процесом і часто базується на експертних оцінках, тому має вірогідний характер.

Для обґрунтування доцільності виконання науково-дослідної роботи на тему “Технології пошуку несправностей в сучасних комп’ютерних мережах за використання великих мовних моделей” використовується спеціальний комплексний показник, що враховує важливість, результативність роботи, можливість впровадження її результатів у виробництво, величину витрат на роботу.

Комплексний показник K_P рівня науково-дослідної роботи може бути розрахований за формулою:

$$K_P = \frac{I^n \cdot T_C \cdot R}{B \cdot t} \quad (4.14)$$

де I — коефіцієнт важливості роботи. I = 3;

n — коефіцієнт використання результатів роботи; n = 0, коли результати роботи не будуть використовуватись; n = 1, коли результати роботи будуть

використовуватись частково; n = 2, коли результати роботи будуть використовуватись в дослідно-конструкторських розробках; n = 3, коли результати можуть використовуватись навіть без проведення дослідно-конструкторських розробок. n=2;

T_C — коефіцієнт складності роботи, T_C = 3;

R — коефіцієнт результативності роботи; якщо результати роботиплануються вище відомих, то R = 4; якщо результати роботи відповідають відомому рівню, то R = 3; якщо нижче відомих результатів, то R = 4;

B — вартість науково-дослідної роботи, тис. грн. B = 564 тис. грн;

t — час проведення дослідження. t = 0,08 років.

$$K_P = \frac{I^n \cdot T_C \cdot R}{B \cdot t} = \frac{3^2 \cdot 3 \cdot 4}{564 \cdot 0{,}08} = 2{,}39$$

Якщо $K_P > 1$, то науково-дослідну роботу на тему “Технології пошуку несправностей в сучасних комп’ютерних мережах за використання великих мовних моделей” можна вважати ефективною з високим науковим, технічним і економічним рівнем.

4.4 Висновки до розділу

У результаті було розраховано витрати на проведення науково-дослідної роботи, які складають 564962,51 грн. Відповідно до проведеного аналізу та розрахунків рівень науково-економічного ефекту проведеної науково-дослідної роботи на тему “Технології пошуку несправностей в сучасних комп’ютерних мережах за використання великих мовних моделей” є середнім, а дослідження актуальними, рівень доцільності виконання науково-дослідної роботи $K_P > 1$, що свідчить про потенційну ефективність з високим науковим, технічним і економічним рівнем.

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи було проведено комплексне дослідження теоретичних, аналітичних та практичних аспектів діагностики та усунення несправностей у комп’ютерних мережах за використання великих мовних моделей.

У першому розділі розглянуто теоретичні основи функціонування комп’ютерних мереж, класифікацію їх типів і принципи маршрутизації. Проаналізовано основні алгоритми маршрутизації, типові проблеми під час конфігурування статичної та динамічної маршрутизації, а також труднощі при конфігуруванні таких мережевих технологій, як VLAN, DHCP, NAT і Frame Relay. Окрему увагу приділено традиційним методам пошуку несправностей і сучасним тенденціям автоматизації траблшутінгу. Це забезпечило формування системного уявлення про можливі джерела помилок у комп’ютерних мережах і дозволило визначити напрями, де автоматизація може підвищити ефективність діагностики.

У другому розділі досліджено архітектуру та принципи роботи великих мовних моделей (LLM), їх можливості у сфері мережевих технологій та методи взаємодії з ними за допомогою prompt-engineering. Це дозволило сформувати методику проведення експериментів, спрямованих на оцінку потенціалу LLM у вирішенні практичних завдань мережевої діагностики, а також забезпечило основу для побудови власного алгоритму ефективного промптингу для траблшутінгу.

У третьому розділі проведено серію експериментальних досліджень, що охопили основні сценарії діагностики: моделювання конфігураційних проблем топології дерева (STP, PAgP), статичної та динамічної маршрутизації, VLAN, адресних служб (DHCP, NAT) та технології Frame Relay. Порівняльний аналіз результатів роботи різних мовних моделей підтвердив, що LLM здатні ефективно підтримувати процеси виявлення, ізоляції та усунення несправностей у комп’ютерних мережах, за умови правильного формулювання промптів. У

результаті сформовано алгоритм побудови промптів для мережевої діагностики, що дозволило стандартизувати підхід до використання LLM у процесі траблшутінгу та забезпечило підвищення точності рекомендацій.

У четвертому розділі визначено рівень науково-економічного ефекту та проведено розрахунок витрат на проведення науково-дослідної роботи, що підтвердило потенційну ефективність з високим науковим, технічним і економічним рівнем.

Проведене дослідження підтвердило доцільність інтеграції великих мовних моделей у процес технічної підтримки та діагностики комп'ютерних мереж, що створює перспективи подальшої автоматизації аналітичних процесів і створення інтелектуальних систем підтримки мережевого адміністрування.

Результати виконання магістерської кваліфікаційної роботи оформлені згідно з вимогами [41].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Захарченко С. М., Балух Б. А. ТЕХНОЛОГІЇ ПОШУКУ НЕСПРАВНОСТЕЙ В СУЧАСНИХ КОМП'ЮТЕРНИХ МЕРЕЖАХ ЗА ВИКОРИСТАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ. Конференція ВНТУ: LIV Всеукраїнська науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2025). Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/24354

2. Захарченко С. М., Балух Б. А. ДІАГНОСТИКА НЕСПРАВНОСТЕЙ КОНФІГУРАЦІЇ VLAN ТА VTP ЗА ВИКОРИСТАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ. Конференція ВНТУ: Молодь в науці: дослідження, проблеми, перспективи (МН - 2026). Режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2026/paper/view/25875

3. Азаров О. Д. Комп'ютерні мережі / О. Д. Азаров, С. М. Захарченко, О. В. Кадук та ін. — Вінниця : ВНТУ, 2013. — 370 с. ISBN 978-966-641-543-4

4. Комп'ютерні мережі: [навчальний посібник] / А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, В. В. Пасічник. — Львів: "Магнолія 2006", 2013. — 256 с. ISBN 978-617-574-087-3

5. Буров Є. В. Комп'ютерні мережі: підручник / Євген Вікторович Буров. — Львів: "Магнолія 2006", 2010. — 262 с. ISBN 966-8340-69-8

6. Кулаков Ю. О. Комп'ютерні мережі / Ю. О. Кулаков, Г. М. Луцький. — К. : Юніор, 2005. — 397 с. ISBN 966-7323-27-7

7. Stallings William. Data and Computer Communication / William Stallings. — 1999. — 810 p. ISBN-10: 0130843709.

8. DSA Dijkstra's Algorithm [Електронний ресурс] — Режим доступу до ресурсу: https://www.w3schools.com/dsa/dsa_algo_graphs_dijkstra.php

9. Азаров О. Д., Захарченко С. М., Яремчук Ю. Є., Дубінін В. М.. Основи роботи та адміністрування мережних операційних систем. Основи роботи та адміністрування мережних операційних систем [Текст] : навчальний посібник з дисципліни "Корпоративні та загальнодоступні мережі" / О. Д.

Азаров, С. М. Захарченко, Є. В. Яремчук, В. М. Дубінін. — Вінниця : ВДТУ. — 2001. — 114 с.

10. Routing Information Protocol (RIP) [Електронний ресурс] — Режим доступу до ресурсу: https://www.geeksforgeeks.org/computer-networks/routing-information-protocol-rip/

11. How OSPF protocol implements Dijkstra Algorithm [Електронний ресурс] — Режим доступу до ресурсу: https://medium.com/@kp-the-great/how-ospf-protocol-implements-dijkstra-algorithm-53c390199ee8

12. EIGRP Diffusing Update Algorithm (DUAL) [Електронний ресурс] — Режим доступу до ресурсу: https://study-ccna.com/eigrp-diffusing-update-algorithm-dual/

13. Захарченко С. М., Трояновська Т. І., Бойко О. В.. Основи побудови захищених мереж на базі обладнання компанії Cisco : навчальний посібник. Основи побудови захищених мереж на базі обладнання компанії Cisco, навчальний посібник / Захарченко С. М., Трояновська Т. І., Бойко О. В. Вінниця, ВНТУ, 2017. — 135 с.

14. Stallings William. Computer Networking with Internet Protocols and Technology / William Stallings. — 2004. — 640 p. ISBN 10 9780131410985.

15. Комп’ютерні мережі : навчальний посібник / О. С. Городецька, В. А. Гикавий, О. В. Онищук. − Вінниця : ВНТУ, 2017. − 129 с.

16. Простір приватних IPv4-адрес [Електронний ресурс] — Режим доступу до ресурсу: https://the-purple.team/nat_basics/

17. Frame Relay Glossary [Електронний ресурс] — Режим доступу до ресурсу: https://www.cisco.com/c/en/us/support/docs/wan/frame-relay/47202-87.html

18. Comprehensive Guide to Configuring and Troubleshooting Frame Relay [Електронний ресурс] — Режим доступу до ресурсу: https://www.cisco.com/c/en/us/support/docs/wan/frame-relay/16563-12.html

19. Understand the Ping and Traceroute Commands [Електронний ресурс] — Режим доступу до ресурсу: https://www.cisco.com/c/en/us/support/docs/ios-nx-

os-software/ios-software-releases-121-mainline/12778-ping-traceroute.html

20. Cisco IOS Debug Command Reference — Commands I through L [Електронний ресурс] — Режим доступу до ресурсу: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/debug/command/db-i1-cr-book/db-i1.html

21. William Stallings. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. 3-rd ed., Addison-Wesley, 1999. — 619 p.

22. Моніторинг мультисервісних мереж. Комп'ютерний практикум: навч. посіб. для студентів спеціальності 121 — "Інженерія програмного забезпечення" денної форми навчання / Укладач: Федорова Н.В.; КПІ ім. Ігоря Сікорського. — Київ: КПІ ім. Ігоря Сікорського, 2020. — 105 с.

23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. In Advances in Neural Information Processing Systems (NeurIPS 2017), 30, pp. 5998-6008. — 11 p.

24. How do Transformers work? [Електронний ресурс] — Режим доступу до ресурсу: https://huggingface.co/learn/llm-course/chapter1/4

25. Mean Squared Error vs Cross Entropy Loss Function [Електронний ресурс] — Режим доступу до ресурсу: https://vitalflux.com/mean-squared-error-vs-cross-entropy-loss-function/

26. Optimizers: A Deep Dive into Gradient Descent, Adam, and Beyond [Електронний ресурс] — Режим доступу до ресурсу: https://medium.com/@ilyurek/optimizers-a-deep-dive-into-gradient-descent-adam-and-beyond-e6a1d00bc9b0

27. Adaptable and Precise: Enterprise-Scenario LLM Function-Calling Capability Training Pipeline [Електронний ресурс] — Режим доступу до ресурсу: https://arxiv.org/abs/2412.15660

28. Introduction to Large Language Models [Електронний ресурс] — Режим доступу до ресурсу: https://developers.google.com/machine-learning/resources/intro-llms

29. LLM-Chatbots — An introduction to the new world of bots

[Електронний ресурс] — Режим доступу до ресурсу: https://sophiehundertmark.medium.com/llm-chatbots-an-introduction-to-the-new-world-of-bots-485db17da7b2

30. OpenAI. ChatGPT API Documentation [Електронний ресурс] — Режим доступу до ресурсу: https://platform.openai.com/docs/

31. Mixture of Experts: How an Ensemble of AI Models Decide As One [Електронний ресурс] — Режим доступу до ресурсу: https://deepgram.com/learn/mixture-of-experts-ml-model-guide

32. Constitutional AI: Harmlessness from AI Feedback [Електронний ресурс] — Режим доступу до ресурсу: https://www.anthropic.com/research/constitutional-ai-harmlessness-from-ai-feedback

33. What is Grouped Query Attention (GQA)? [Електронний ресурс] — Режим доступу до ресурсу: https://klu.ai/glossary/grouped-query-attention

34. Sliding Window Attention [Електронний ресурс] — Режим доступу до ресурсу: https://www.deepchecks.com/glossary/sliding-window-attention/

35. Sahoo, Pranab; Singh, Ayush Kumar; Saha, Sriparna; Jain, Vinija; Mondal, Samrat; Chadha, Aman. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. arXiv preprint arXiv:2402.07927, 5 February 2024. — 9 p.

36. Haochen Li; Jonathan Leung; Zhiqi Shen. Towards Goal-oriented Prompt Engineering for Large Language Models: A Survey. arXiv preprint arXiv:2401.14043, 25 January 2024. — 15 p.

37. Etalon: Holistic Performance Evaluation Framework for LLM Inference Systems [Електронний ресурс] — Режим доступу до ресурсу: https://arxiv.org/abs/2407.07000

38. Saini H., Laskar M. T. R., Chen C., Mohammadi E., Rossouw D. LLM Evaluate: An Industry-Focused Evaluation Tool for Large Language Models. Proceedings of the 31st International Conference on Computational Linguistics: Industry Track, COLING 2025 — Abu Dhabi, UAE, January 2025. — 294 p.

39. Evaluation of Large Language Models: Review of Metrics, Applications,

and Methodologies [Електронний ресурс] — Режим доступу до ресурсу: https://www.researchgate.net/publication/390486356_Evaluation_of_Large_Language_Models_Review_of_Metrics_Applications_and_Methodologies

40. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. — Вінниця : ВНТУ, 2021. — 42 с.

41. Методичні вказівки до виконання магістерських кваліфікаційних робіт студентами спеціальності 123 “Комп’ютерна інженерія”. / Укладачі О.Д. Азаров, О.В. Дудник, С.І. Швець − Вінниця : ВНТУ, 2023. − 57 с.

ДОДАТОК А

Технічне завдання
Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ
Завідувач кафедри ОТ
д.т.н., проф. Азаров О. Д.

_____

“ 3 “ жовтня 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи
“Технології пошуку несправностей в сучасних комп'ютерних мережах за використання великих мовних моделей”

Науковий керівник: к.т.н., проф.
_____Захарченко С. М.

Виконав: студент гр. 1КІ-24м
_____Балух Б.А.

ВНТУ 2025

1 Підстава для виконання магістерської кваліфікаційної роботи (МКР)

1.1 Актуальність полягає в тому, що зростання складності та масштабів комп’ютерних мереж у всіх сферах діяльності потребує високого рівня надійності, безпеки та ефективного управління. Своєчасне виявлення й усунення несправностей є критично важливим для стабільної роботи мережевої інфраструктури. Традиційні методи діагностики потребують значних ресурсів, тому актуальним є застосування сучасних технологій штучного інтелекту, зокрема великих мовних моделей, які дозволяють автоматизувати процеси аналізу, прогнозування та усунення мережевих проблем.

1.2 Наказ про затвердження теми МКР №313 від 24.09.2025 р.

2 Мета і призначення МКР

2.1 Метою роботи є розробка методології та алгоритму ефективного використання великих мовних моделей для пошуку і діагностики несправностей у сучасних комп’ютерних мережах.

2.2 Призначення розробки — забезпечення інтелектуальної підтримки мережевого адміністрування шляхом автоматизації процесів аналізу, пошуку та діагностики несправностей у комп’ютерних мережах за використанням великих мовних моделей, що підвищує ефективність обслуговування, зменшує час реагування на збої та покращує стабільність роботи мережевої інфраструктури.

3 Вихідні дані для виконання МКР

3.1 Теоретичний аналіз основ діагностики та усунення несправностей у комп’ютерних мережах.

3.2 Вивчення великих мовних моделей, як інструменту для траблшутінгу в комп’ютерних мережах.

3.3 Проведення експериментальних досліджень.

3.4 Виконання економічних розрахунків для доведення значущості наукового дослідження.

4 Технічні вимоги до виконання МКР

Технічні вимоги:

— аналіз теоретичних основ діагностики та усунення несправностей у комп’ютерних мережах;

— розробка методики prompt-engineering для ефективної взаємодії з LLM під час діагностики;

— проведення експериментальних досліджень із моделюванням типових мережевих проблем;

— порівняльний аналіз результатів роботи різних великих мовних моделей;

— формування алгоритму побудови промптів для автоматизації мережевої діагностики.

5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в Таблиці А.1

Таблиця А.1 — Етапи МКР

№ з/п	Назва етапу	Термін виконання		Очікувані результати
		початок	закінчення	
1	Постановка задачі роботи	08.09.2025	08.09.2025	Задачі дослідження
2	Аналіз принципів побудови комп’ютерних мереж, мережевих протоколів та технологій з метою визначення типових проблем	10.09.2025	15.09.2025	Розділ 1
3	Вивчення архітектури LLM, їхнього потенціалу в мережевій діагностиці та методів prompt-engineering	16.09.2025	23.09.2025	Розділ 2
4	Розробка методики проведення експериментальних досліджень	24.09.2025	25.09.2025	Розділ 2

Продовження таблиці А.1

№ з/п	Назва етапу	Термін виконання		Очікувані результати
		початок	закінчення	
5	Проведення експериментів, оцінка ефективності LLM та формування алгоритму побудови ефективних промптів	26.09.2025	11.10.2025	Розділ 3
6	Розрахунок економічної частини	15.10.2025	15.10.2025	Розділ 4
7	Оформлення матеріалів до захисту МКР	16.10.2025	16.10.2025	ПЗ, графічний матеріал і презентація
8	Перевірка якості виконання магістерської роботи та усунення недоліків	25.10.2025	25.10.2025	Оформлені документи
9	Підписи супроводжувальних документів у нормоконтролера, керівника, опонента	03.11.2025	03.11.2025	Оформлені документи
10	Перевірка на антиплагіат та ШІ	05.11.2025	05.11.2025	Оформлені документи

6 Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами.

7 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

8 Вимоги до оформлювання та порядок виконання МКР

8.1 При оформлюванні МКР використовуються:

— ДСТУ 3008: 2015 “Звіти в сфері науки і техніки. Структура та правила оформлювання”;

— ДСТУ 8302: 2015 “Бібліографічні посилання. Загальні положення та правила складання”;

— ГОСТ 2.104-2006 “Єдина система конструкторської документації. Основні написи”;

— методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — “Комп’ютерна інженерія”;

— документи на які посилаються у вище вказаних.

8.2 Порядок виконання МКР викладено в “Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21”.

ДОДАТОК Б

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Технології пошуку несправностей в сучасних комп'ютерних мережах за використання великих мовних моделей

Тип роботи: магістерська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра обчислювальної техніки, ФІТКІ, 1КІ-24м
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КП1) 13 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- ✓ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак академічного плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту.
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки академічного плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. Робота до захисту не приймається.

Експертна комісія:

Азаров О. Д., д.т.н., зав. каф. ОТ _____
(прізвище, ініціали, посада) (підпис)

Мартинюк Т. Б., д.т.н., проф. каф. ОТ _____
(прізвище, ініціали, посада) (підпис)

Особа, відповідальна за перевірку _____ Захарченко С. М.
(підпис) (прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник _____ Захарченко С. М., к.т.н., проф. каф. ОТ
(підпис) (прізвище, ініціали, посада)

Здобувач _____ Балух Б. А.
(підпис) (прізвище, ініціали)

ДОДАТОК В

Блок-схема алгоритму традиційного пошуку несправностей у комп’ютерних мережах

Рисунок В.1 — Блок-схема алгоритму традиційного пошуку несправностей у комп’ютерних мережах

ДОДАТОК Г

Конфігураційний файл маршрутизатора Rt1

```
Current configuration : 875 bytes
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
hostname Router
ip cef
no ipv6 cef
license udi pid CISCO2811/K9 sn FTX1017B65W-
spanning-tree mode pvst
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
interface FastEthernet0/0.6
 encapsulation dot1Q 6
 ip address 55.0.15.126 255.255.255.128
interface FastEthernet0/0.13
 encapsulation dot1Q 13
 ip address 191.18.10.254 255.255.255.192
interface FastEthernet0/0.28
 encapsulation dot1Q 28
 ip address 111.2.0.254 255.255.255.0
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
 shutdown
interface Vlan1
 no ip address
 shutdown
ip classless
ip flow-export version 9
line con 0
line aux 0
line vty 0 4
 login
end
```

ДОДАТОК Д

Конфігураційний файл маршрутизатора Router0

```
Current configuration : 2203 bytes
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
hostname Router
ip dhcp excluded-address 10.8.27.254
ip dhcp excluded-address 172.25.27.254
ip dhcp excluded-address 172.16.2.254
ip dhcp pool NET_A
 network 10.8.27.0 255.255.255.0
 default-router 10.8.27.254
 domain-name example.com
ip dhcp pool NET_B
 network 172.25.27.0 255.255.255.0
 default-router 172.25.27.254
 domain-name example.com
ip dhcp pool NET_C
 network 172.16.2.0 255.255.255.0
 default-router 172.16.2.254
 domain-name example.com
no ip cef
no ipv6 cef
interface FastEthernet0/0
 ip address 10.8.27.254 255.255.255.0
 ip nat inside
 duplex auto
 speed auto
interface FastEthernet1/0
 ip address 172.25.27.254 255.255.255.0
 ip nat inside
 duplex auto
 speed auto
interface Serial2/0
 ip address 173.2.32.1 255.255.240.0
 ip nat outside
interface FastEthernet3/0
 ip address 172.16.2.254 255.255.255.0
 ip nat inside
 duplex auto
 speed auto
interface FastEthernet4/0
 ip address 192.168.2.254 255.255.255.0
 ip nat inside
 duplex auto
 speed auto
interface FastEthernet5/0
```

```
 no ip address
 duplex auto
 speed auto
 shutdown
interface FastEthernet6/0
 no ip address
 duplex auto
 speed auto
 shutdown
interface Serial7/0
 no ip address
 clock rate 2000000
 shutdown
interface Serial8/0
 no ip address
 clock rate 2000000
 shutdown
interface FastEthernet9/0
 no ip address
 duplex auto
 speed auto
 shutdown
ip nat pool NET_E_POOL 173.2.32.6 173.2.47.253 netmask 255.255.240.0
ip nat pool NET_E_POOL_B 173.2.32.4 173.2.32.5 netmask 255.255.240.0
ip nat inside source list 1 pool NET_E_POOL
ip nat inside source list 2 pool NET_E_POOL_B overload
ip nat inside source list 3 interface Serial2/0 overload
ip nat inside source static 192.168.2.2 173.2.32.3
ip nat inside source static 192.168.2.1 173.2.32.2
ip classless
ip route 0.0.0.0 0.0.0.0 173.2.47.254
ip flow-export version 9
access-list 1 permit 10.8.27.0 0.0.0.255
access-list 2 permit 172.25.27.0 0.0.0.255
access-list 3 permit 172.16.2.0 0.0.0.255
line con 0
line aux 0
line vty 0 4
 login
end
```

ДОДАТОК Е

Конфігураційний файл маршрутизатора Router1

```
Current configuration : 842 bytes
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
hostname Router
no ip cef
no ipv6 cef
username Router5 password 0 1111
spanning-tree mode pvst
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
 shutdown
interface Serial0/0/0
 ip address 35.0.0.1 255.255.0.0
 encapsulation frame-relay
interface Serial0/1/0
 ip address 35.1.0.2 255.255.0.0
 encapsulation ppp
 ppp authentication pap
 ppp pap sent-username Router1 password 0 1111
 clock rate 2000000
interface Vlan1
 no ip address
 shutdown
ip classless
ip flow-export version 9
line con 0
line aux 0
line vty 0 4
 login
end
```

ДОДАТОК Ж

Блок-схема алгоритму побудови промптів для мережевої діагностики

Рисунок Ж.1 — Блок-схема алгоритму побудови промптів для мережевої діагностики