Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)
Кафедра Інфокомунікаційної інженерії імені В.В. Поповського
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Дослідження процесів управління телекомунікаційними мережами із застосуванням засобів автоматизації
(тема)

Виконав:
студент 2 курсу, групи ТСМм-22-1
Мамон Р.В.
(прізвище, ініціали)

Спеціальність: 172 Телекомунікації та радіотехніка
(код і повна назва спеціальності)
Тип програми: освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма: Телекомунікаційні системи та мережі
(повна назва освітньої програми)

Керівник: проф. кафедри ІКІ ім. В.В. Поповського
Єременко О.С.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ Лемешко О.В.
(підпис) (прізвище, ініціали)

2024р.

Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)
Кафедра Інфокомунікаційної інженерії імені В.В. Поповського
(повна назва)
Рівень вищої освіти другий (магістерський)
Спеціальність 172 Телекомунікації та радіотехніка
(код і повна назва)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Телекомунікаційні системи та мережі
(повна назва)

ЗАТВЕРДЖУЮ
Зав. кафедри_____
(підпис)
«____» _____2024р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Мамону Роману Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження процесів управління телекомунікаційними мережами із застосуванням засобів автоматизації
затверджена наказом по університету від «08» березня 2024 р. №210 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 15.06.2024 р.
3. Вихідні дані до роботи: існуючі засоби автоматизації; пропрієтарне програмне забезпечення щодо управління пристроями телекомунікаційних мереж; інтегроване середовище розробки програмного забезпечення; завдання для автоматизації у вигляді вебзастосунку – опитування порту маршрутизатора мережі.
4. Перелік питань, що потрібно опрацювати в роботі:
1) Провести аналіз та порівняння існуючих засобів автоматизації.
2) Розглянути питання щодо розробки нових засобів автоматизації завдань управління в телекомунікаційних системах і мережах.
3) Провести огляд інструментів та мов програмування, що використовуються під час автоматизації мереж.
4) Виконати програмну реалізацію завдань управління та моніторингу в телекомунікаційній мережі.
5) Провести кількісну оцінку ефективності автоматизації завдання у розробленому програмному забезпеченні.

5. Перелік графічного матеріалу із зазначенням креслень, плакатів, комп'ютерних ілюстрацій: Демонстраційний матеріал у вигляді ppt-презентації.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		(підпис)	(дата)
Основна частина	професор Єременко Олександра Сергіївна		12.06.2024

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	15.02.2024	Виконано
2	Збір матеріалів для дослідження	28.02.2024	Виконано
3	Розробка 1 розділу	19.03.2024	Виконано
4	Розробка 2 розділу	02.04.2024	Виконано
5	Розробка 3 розділу	12.04.2024	Виконано
6	Розробка 4 розділу	23.05.2024	Виконано
7	Оформлення кваліфікаційної роботи	10.06.2024	Виконано

Дата видачі завдання 15 лютого 2024 року

Студент _____ Мамон Р.В.
(підпис) (прізвище, ініціали)

Керівник роботи _____ проф. Єременко О.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 62 с., 36 рис., 2 табл., 3 додатки, 25 джерел.

ТЕЛЕКОМУНІКАЦІЙНА МЕРЕЖА, УПРАВЛІННЯ, АВТОМАТИЗАЦІЯ, ІНТЕРФЕЙС.

Об’єкт дослідження – процес управління телекомунікаційними мережами із застосуванням засобів автоматизації.

Предмет дослідження – засоби автоматизованого управління телекомунікаційними мережами.

Мета роботи – аналіз шляхів підвищення ефективності управління телекомунікаційними мережами із застосуванням засобів автоматизації.

Методи досліджень – емпіричний аналіз, формалізація, розробка та порівняння.

В роботі проведено аналіз трендів і стратегій мережної автоматизації, а також визначено переваги автоматизації мереж. Виконано огляд типів мережної автоматизації, приділено увагу завданням управління в телекомунікаційних мережах, що підлягають автоматизації. Розглянуто особливості, переваги та недоліки застосування інтерфейсу командного рядка для мережної автоматизації. Наведено приклади практичного використання програмного забезпечення для налаштування телекомунікаційного обладнанная відомих виробників. Проаналізовано інструменти та мови програмування, такі як Perl та Python, для створення скриптів і програмного забезпечення. Детально описано програмну реалізацію завдань управління та моніторингу мережі, а також оцінено ефективність автоматизації у разі використання розробленого вебзастосунку. Робота демонструє важливість автоматизації для підвищення ефективності та надійності телекомунікаційних мереж.

ABSTRACT

The report contains: 62 p., 36 fig., 2 table, 3 appendices, 25 sources.

TELECOMMUNICATIONS NETWORK, MANAGEMENT, AUTOMATION, INTERFACE.

The object of research is the process of telecommunication network management using automation means.

The subject of research is the means of automated management of telecommunication networks.

The work aims to analyze ways to improve the efficiency of telecommunications network management using automation means.

Research methods – empirical analysis, formalization, development, and comparison.

The work analyses the trends and strategies of network automation and identifies the benefits of network automation. The types of network automation are reviewed, and attention is paid to the management tasks in telecommunication networks that are subject to automation. The features, advantages, and disadvantages of using the command line interface for network automation are considered. Examples of practical use of software from third-party manufacturers for configuring telecommunications equipment are given. Tools and programming languages such as Perl and Python are analyzed for creating scripts and software. The software implementation of network management and monitoring tasks is described in detail, and the effectiveness of automation is evaluated when using the developed web application. The work demonstrates the importance of automation in improving the efficiency and reliability of telecommunication networks.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення
ТКМ – телекомунікаційна мережа
AAR – Application Aware Routing
ADSL – Asymmetric Digital Subscriber Line
AI – Artificial Intelligence
AJAX – Asynchronous JavaScript and XML
API – Application Programming Interface
CGI – Common Gateway Interface
CLI – Command-Line Interface
CORBA – Common Object Request Broker Architecture
DOM – Document Object Model
FTTH – Fiber To The Home
GUI – Graphical User Interface
HTML – HyperText Markup Language
HTTP – HyperText Transfer Protocol
IBN – Intent-Based Network
IP – Internet Protocol
JSON – JavaScript Object Notation
LAN – Local Area Network
ML – Machine Learning
MPLS – Multiprotocol Label Switching
MTTR – Mean Time To Repair
NOC – Network Operations Center
OT – Operational Technology
SDN – Software Defined Network
SMB – Small and Midsize Business
SMPP – Short Message Peer-to-Peer
SNMP – Simple Network Management Protocol
SSH – Secure Shell
SSL – Secure Sockets Layer
vEPC – virtual Evolved Packet Core

VLAN – Virtual Local Area Network
VNF – Virtual Network Functions
VPN – Virtual Private Network
WAN – Wide Area Network
XML – eXtensible Markup Language
Telnet – TELetype NETwork

ВСТУП

У сучасному світі технологій та зв’язку телекомунікаційні системи і мережі відіграють ключову роль у забезпеченні безперервного зв’язку та обміну інформацією. Швидкі та надійні телекомунікаційні мережі стають все більш важливими для різноманітних галузей. Однак зростаюча складність цих мереж та потреба у швидкій реакції на змінні умови пов’язані з постійним прагненням до оптимізації та ефективності. Саме тут виникає необхідність у використанні засобів автоматизації, які спрямовані на полегшення управління та підтримку телекомунікаційних систем і мереж.

Автоматизація управління телекомунікаційними мережами (ТКМ) – це процес використання програмного забезпечення для автоматизації численних завдань управління ТКМ з метою постійного підвищення її ефективності та функціональності [1 – 4]. На сьогодні забезпечення швидкого, гнучкого та злагодженого управління є головною вимогою як у традиційних мережах, так і різноманітних архітектурах програмно-конфігурованих мереж [5 – 7]. Зі свого боку автоматизація на основі API дозволила замінити ручні інструкції під час використання командного рядка для налаштування кожного мережного пристрою. API можна викликати безпосередньо або через мову програмування, наприклад Python, Java, Perl тощо [8 – 15]. Водночас сценарії є лише одним із аспектів автоматизації мереж.

Таким чином, дослідження процесу управління ТКМ із застосуванням засобів автоматизації представляється важливою практичною задачей.

Метою роботи є аналіз шляхів підвищення ефективності управління телекомунікаційними мережами із застосуванням засобів автоматизації.

Для вирішення поставленої задачі, в першому розділі кваліфікаційної роботи розглянуто основні тенденції та стратегії мережної автоматизації. Проаналізовано переваги, які надає автоматизація, а також різні типи мережної автоматизації. Детально розглянуто завдання управління, які підлягають автоматизації в телекомунікаційних мережах.

Другий розділ присвячено особливостям розробки нових засобів автоматизації. Розглянуто переваги та недоліки використання інтерфейсу командного рядка, а також наведено приклади практичного використання

програмного забезпечення для налаштування телекомунікаційного обладнання відомих виробників, таких як Huawei та Ericsson.

У третьому розділі розглядаються інструменти та мови програмування, що використовуються для автоматизації мереж. Проаналізовано архітектуру програмного забезпечення, що розробляється у межах кваліфікаційної роботи, обґрунтоваано використані технології та вибір мови програмування для написання скриптів. Розглянуто характеристики Perl та Python у застосуванні до мережної автоматизації, а також приклади практичного застосування скриптів.

Четвертий розділ описує програмну реалізацію завдань управління та моніторингу в телекомунікаційній мережі. Детально розглянуто ядро програмного забезпечення, клієнтську та серверну частини вебзастосунку.

У п’ятому розділі проведено оцінку ефективності автоматизації завдань управління мережею. Розглянуто якісну характеристику розробленого програмного забезпечення, а також здійснено кількісну оцінку ефективності автоматизації типових завдань управління та конфігурації мережних пристроїв.

Окремі результати роботи доповідались на трьох Міжнародних науково-технічних конференціях та апробовані в лабораторній мережі компанії Nokia Bell NV (м. Антверпен, Бельгія).

1 АНАЛІЗ ТА ПОРІВНЯННЯ ІСНУЮЧИХ ЗАСОБІВ АВТОМАТИЗАЦІЇ

1.1 Аналіз трендів і стратегій мережної автоматизації

Автоматизація мереж – це процес автоматизації розгортання, конфігурації, тестування/перевірки та експлуатації мережних пристроїв, які можуть бути фізичними (маршрутизатори, комутатори, точки доступу, брандмауери тощо) або віртуальними (загальнодоступні хмарні мережі, віртуальні машини, контейнери, віртуальні мережні функції тощо) [1 – 4]. Автоматизація мережі може бути застосована до широкого спектру мережних послуг підприємств і постачальників послуг (сервіс-провайдерів), відповідні приклади наведено нижче.

У межах корпоративних мереж автоматизація може бути застосована в наступному [1]:

– мереж центрів обробки даних, включаючи порти доступу, порти-канали, маршрутизацію, якість обслуговування, брандмауери та політики балансування навантаження, гіпервізори, віртуальні машини та контейнери тощо;

– користувачів публічних хмар, віртуальних мереж, підмереж, груп безпеки, балансувальників навантаження, обчислювальних сервісів, сервісів зберігання даних і безсерверних функцій;

– кампусних мереж та мереж OT (Operational Technology), включаючи проводовий та безпроводовий доступ, контроль доступу до мережі, макро- та мікро-сегментацію;

– обладнання для глобальних мереж (Wide Area Networks, WAN), накладених віртуальних приватних мереж, маршрутизації з урахуванням вимог програм (Application Aware Routing, AAR), якість обслуговування та політики маршрутизації.

Зі свого боку автоматизація на рівні постачальників послуг містить [1]:

– стаціонарний ADSL і оптоволоконний доступ до будинків (Fiber To The Home, FTTH);

– політики доступу, безпеки та маршрутизації віртуальних приватних мереж (Virtual Private Network, VPN) для малого та середнього бізнесу (Small and Midsize Business, SMB) і підприємств;

– віртуальні мережні функції (Virtual Network Functions, VNF), включаючи маршрутизатори, брандмауери, балансувальники навантаження і віртуальні компоненти virtual Evolved Packet Core (vEPC);

– VPN-з'єднання від підприємства до публічної хмари, включаючи приграничні хмарні маршрутизатори.

Автоматизація мережі поширюється на весь життєвий цикл послуги, включаючи введення в експлуатацію мережі в день 0, конфігурацію в день 1 і експлуатацію та оптимізацію в день 2. Варіанти використання автоматизації на кожному етапі життєвого циклу можуть відрізнятися. Серед прикладів можна навести наступні.

1) День 0: розгортання пристроїв мережі без дотику (plug&play), яке іноді називають "вбудовуванням" (onboarding). Розгортання сертифікатів безпеки пристрою. Розгортання віртуальної машини та підключення до мережі.

2) День 1: Конфігурація та управління мережею з використанням шаблонів пристроїв, визначених командами мережних інженерів або прописаних контролером мережі. Конфігурація сегментації мережі на основі аналітики поведінки трафіку.

3) День 2: Оновлення/зміни конфігурацій мережі. Виявлення та усунення вразливостей безпеки на основі AI/ML. Планування потужностей. Управління життєвим циклом обладнання та програмного забезпечення.

1.2 Переваги автоматизації мереж

Відомо, що зміни у мережі, які виконується вручну, часто призводять до неузгодженості мереж, що є результатом конфігурацій, побудованих різними інженерами, погано задокументованих операційних журналів або людських помилок [5 – 7]. Така неузгодженість конфігурації часто призводить до простою мережі, зниження швидкості обслуговування та збільшення операційних витрат.

Автоматизація мережі може усунути вищезгадані недоліки ручного адміністративного втручання, надаючи наступні переваги [1].

Рисунок 1.1 – Ключові переваги автоматизації мереж

1) Швидка реакція на нові вимоги. Підвищуючи рівень автоматизації мереж, IT-команди можуть зосередити свої зусилля на створенні цінності для бізнесу. Автоматизація дозволяє знизити вартість операцій, одночасно покращуючи рівень обслуговування та швидкість реагування на нові вимоги до послуг/застосунків і розгортання.

2) Висока надійність. Завдяки заміні ручної роботи автоматизованими завданнями, пристрої мережі конфігуруються більш послідовно і менш схильні до людських помилок. Завдяки автоматизації підвищується доступність мережі.

3) Зниження операційних витрат. Автоматизація допомагає зменшити операційні витрати, вона усуває ручні, трудомісткі та повторювані завдання. Забезпечення на основі політик і кероване відновлення підвищують час безвідмовної роботи мережі і скорочують час, що витрачається на управління мережними операціями.

4) Стандартизовані конфігурації. Для того, щоб досягти ефективності автоматизації мережі, шаблони мережних конфігурацій повинні бути

стандартизовані. Наявність проєктних планів зі стандартизованими шаблонами і чітко визначеною логікою обслуговування усуває двозначність для інженерів-програмістів, які розробляють програмне забезпечення для автоматизації.

5) Швидке внесення змін до мережі. Час на внесення змін до мережі скорочується, коли завдання виконуються і перевіряються інструментом автоматизації, а не мережним адміністратором.

6) Спрощення роботи віддалених IT-команд. Контролери мережної автоматизації оптимізовані для віддаленого доступу, вони надають інформаційну панель з чіткими та організованими робочими процесами, що полегшує віддалене управління.

Крім того, більш високі рівні автоматизації, такі як системи автоматизації з замкнутим циклом і самокеровані мережі, надають організаціям всі попередні і додаткові переваги, представлені нижче [1].

1) Розширений інтелект. Постійна і ретельна оцінка зібраних даних. Виявлення відхилень та інформування про них.

2) Динамічна оптимізація/виправлення. Постійна підтримка бажаного стану мережі. Використання машинного навчання для вибору найкращого способу реалізації бажаного стану і прийняття автоматизованих коригувальних дій для підтримки цього стану.

3) Спрощене усунення несправностей. Діяльність з усунення несправностей спрощується, коли вона повністю або частково автоматизована. Крім того, автоматизація уможливлює AIOps шляхом автоматичного збору та аналізу даних.

4) Більше розуміння мережних процесів. Багато інструментів автоматизації мережі, зокрема контролери мережі, підвищують видимість вашої інфраструктури та надають інформацію про мережу. Команди мережних адміністраторів можуть вживати проактивних заходів для усунення проблем у мережі до того, як вони вплинуть на діяльність компанії. Аналітика мережі AI/ML скорочує час, що витрачається на управління мережними операціями, і покращує якість обслуговування користувачів.

5) Посилення безпеки. Інструменти автоматизації безпеки покращують поточне розуміння потоків застосунків для виявлення потенційних загроз. Зменшення поверхні атаки шляхом автоматизованої мікросегментації за допомогою індивідуального аналізу та рекомендацій на основі AI/ML. Автоматичне виявлення та забезпечення дотримання нормативних вимог. Краще відстеження стану безпеки.

1.3 Огляд типів мережної автоматизації

Коли мова йде про автоматизовані інженерні технології, виділяють три основні типи автоматизації мереж, які необхідно враховувати під час їхньої реалізації на практиці [2].

1) Програмне забезпечення для автоматизації мереж. Технологія програмної автоматизації, також відома як інтелектуальна мережна автоматизація, є, мабуть, одним з найскладніших типів мережної автоматизації. Програмні засоби автоматизації пропонують користувачам шаблони, які позбавляють необхідності застосовувати команди сценаріїв вручну. Зазвичай вони можуть автоматизувати ширший спектр мережних завдань, ніж сценарії, що полегшує інженерам усунення несправностей, пов'язаних з мережею.

2) Автоматизація мережі на основі сценаріїв. Це тип технології автоматизації мережної інженерії, який використовує сценарії для автоматизації мережних завдань, що включають певні тригери і типові процеси. Цей тип технології автоматизації мережі часто вважається найпростішим і найпоширенішим, оскільки він використовує застарілі мови, такі як Tcl і Perl, які переважно використовуються через їх знайомство з автоматизацією мережних завдань. Інші популярні мови програмування з відкритим вихідним кодом, такі як Ruby, Python, Ansible, Bash і Go, також використовуються в автоматизації мережі на основі сценаріїв завдяки своїй універсальності.

3) Автоматизація мережі на основі намірів. Останній з трьох основних типів – це мережна автоматизації на основі намірів (Intent-Based Network, IBN). Ця технологія мережної інженерії використовує підхід на основі моделей для автоматизації мережних завдань і є найсучаснішим типом автоматизації мереж. Автоматизація на основі намірів використовує штучний інтелект (Artificial Intelligence, AI) і машинне навчання (Machine Learning, ML) для автоматичної зміни способу реалізації політики мережі після розуміння намірів бізнесу і користувача. Це означає, що якщо адміністратори мережі встановлюють рівні обслуговування для програм і користувачів, і ці рівні не досягаються, то мережа автоматично налаштовується так, щоб відновити рівні критично важливих для бізнесу програм якнайкраще.

Ці типи технологій мережної автоматизації можуть підтримувати будь-яку мережу, яку може контролювати оператор зв'язку або API. До них відносяться наступні [1 – 4]:

- глобальні мережі (WAN);
- локальні мережі (LAN);
- хмарні мережі;
- безпроводові мережі;
- мережі центрів обробки даних тощо.

1.4 Завдання управління в телекомунікаційних мережах, що підлягають автоматизації

Приведемо декілька прикладів мережних завдань, які підлягають автоматизації [1 – 4]:

- забезпечення пристроїв;
- управління конфігурацією;
- управління несправностями;
- управління продуктивністю;
- управління безпекою;
- балансування навантаження;
- управління відповідністю нормативним вимогам;
- управління змінами;
- аварійне відновлення.

Забезпечення високого рівня якості обслуговування, необхідного для цифрових бізнес-операцій, може бути складним завданням через невідповідності, неправильні налаштування та нестабільність мережі, які можуть виникнути внаслідок ручного налаштування мережі. Впровадити найкращі практики можна, стандартизувавши операції адміністрування мережі за допомогою автоматизації. Команди з експлуатації мережі можуть швидко і легко зменшити середній час усунення несправностей (MTTR), пов’язаних з перебоями в обслуговуванні. Щоб мінімізувати час очікування процесів, а також максимізувати продуктивність і витрати, навантаження програм повинно бути рівномірно розподілене по всій інфраструктурі. Низька продуктивність застосунків і затримка в обході відмов при виникненні системних збоїв можуть бути наслідком ручного балансування навантаження. Так, наприклад, автоматизувавши балансувальник навантаження, можна швидше вносити безперервні зміни та обхід відмов, що підвищить продуктивність та надійність роботи програм.

2 ПИТАННЯ ЩОДО РОЗРОБКИ НОВИХ ЗАСОБІВ АВТОМАТИЗАЦІЇ, ЯКІ МОЖУТЬ БУТИ ВИКОРИСТАНІ ДЛЯ ЗАВДАНЬ УПРАВЛІННЯ В ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМАХ І МЕРЕЖАХ

2.1 Особливості, переваги та недоліки застосування інтерфейсу командного рядка для мережної автоматизації

Інтерфейс командного рядка – це інтерфейс, в якому користувач використовує текстові команди для взаємодії з системою. Він також відомий як символьний або консольний інтерфейс користувача. До появи графічного інтерфейсу це був основний спосіб взаємодії користувача з операційною системою комп'ютера та програмами [3].

На сьогоднішній день існує багато систем, які використовують інтерфейс командного рядка або як єдиний метод взаємодії, або в поєднанні з графічним інтерфейсом. Для тих систем, що використовують CLI, характерні такі особливості [3].

1) Можливості написання сценаріїв. Скрипти – це комбінація команд, які можна використовувати для автоматизації деяких поширених дій, що повторюються. Сценарії дозволяють користувачам автоматизувати команди, які можуть бути складними при використанні графічного інтерфейсу.

2) Використання конвеєрних команд, коли вихідні дані однієї команди є вхідними даними для іншої команди. Конвеєр здебільшого включає більше однієї команди, де одна залежить від результату іншої.

3) Історія команд. Системи CLI зберігають історію виконання команд, яку можна використовувати для перегляду у разі виникнення проблем. Користувач також може згадати команду і використати її замість того, щоб вводити весь рядок знову.

4) Інтерфейс командного рядка є текстовим або символьним інтерфейсом, який не має вікон, піктограм, меню та вказівників, на відміну від інтерфейсу графічного інтерфейсу.

5) Використання тільки клавіатури. Більшість систем на основі CLI не використовують вказівні пристрої, такі як миші, оскільки більша частина роботи пов'язана з введенням команд за допомогою клавіатури.

6) CLI характеризується високою швидкістю виконання порівняно з графічним інтерфейсом. Це пов'язано з тим, що він більш точний і вимагає менше ресурсів для виконання команди.

Графічний інтерфейс та інтерфейс командного рядка є найпоширенішими типами системних інтерфейсів. Нижче наведено порівняння цих двох типів користувацьких інтерфейсів.

Таблиця 2.1 – Переваги та недоліки використання GUI та CLI для мережної автоматизації [3]

Критерій	GUI	CLI
Швидкість	Відносно повільний	Команди виконуються швидко
Навчальна крива	Інтерфейс простий в опануванні. Користувачі можуть навчитися користуватися ним самостійно.	Команди складні для вивчення і вимагають засвоєння багатьох команд.
Зручність для користувача	Зручний у використанні, оскільки містить звичайні іконки, які користувачі можуть розпізнати.	Не зручний для користувача, оскільки користувачеві потрібно знати всі команди.
Компоненти	Основними компонентами графічного інтерфейсу є Windows Icons Menus and Pointer (WIMP)	Інтерфейс є текстовим.
Основні користувачі	Призначений для використання користувачами всіх рівнів – від простих кінцевих користувачів до системних адміністраторів.	Широко використовуються і надають перевагу досвідченим користувачам, як-от програмісти та системні адміністратори.

2.2 Приклади практичного використання програмного забезпечення для налаштування телекомунікаційного обладнанная відомих виробників

Нижче наведено огляд деяких функцій існуючого ПЗ таких вендорів, як Huawei та Ericsson.

2.2.1 Приклад налаштування обладнання радіорелейної станції Huawei RTN310

Після запуску і авторизації в програмі startweblct.bat потрібно відкрити вікно налаштування обладнання RTN310 кліком на кнопці «NE Explorer» (рис. 2.1).

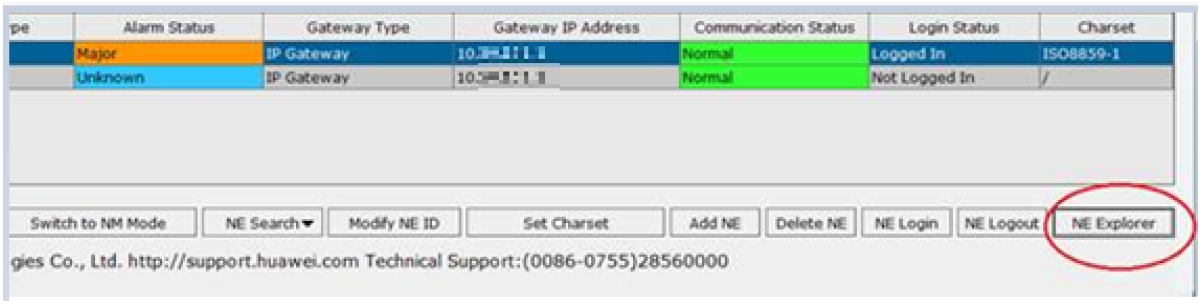


Рисунок 2.1 – Вхід до налаштування RTN310

Вікно програми управління обладнанням наведено на рис. 2.2:

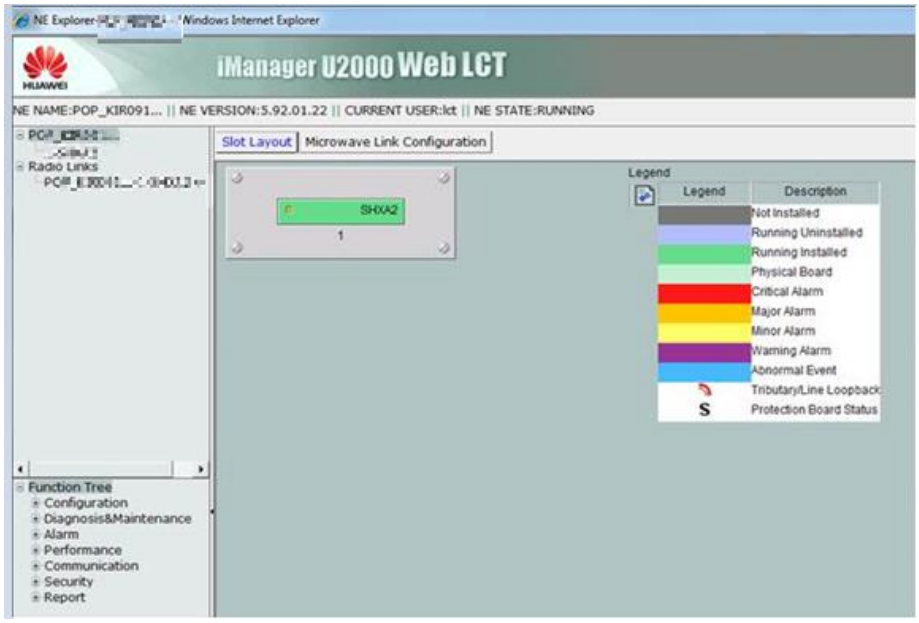


Рисунок 2.2 – Вікно програми управління обладнанням Huawei RTN310

Далі в меню програми «Communication/Communication Parameters», інженер має можливість встановити параметр Gateway IP Address (рис. 2.3).

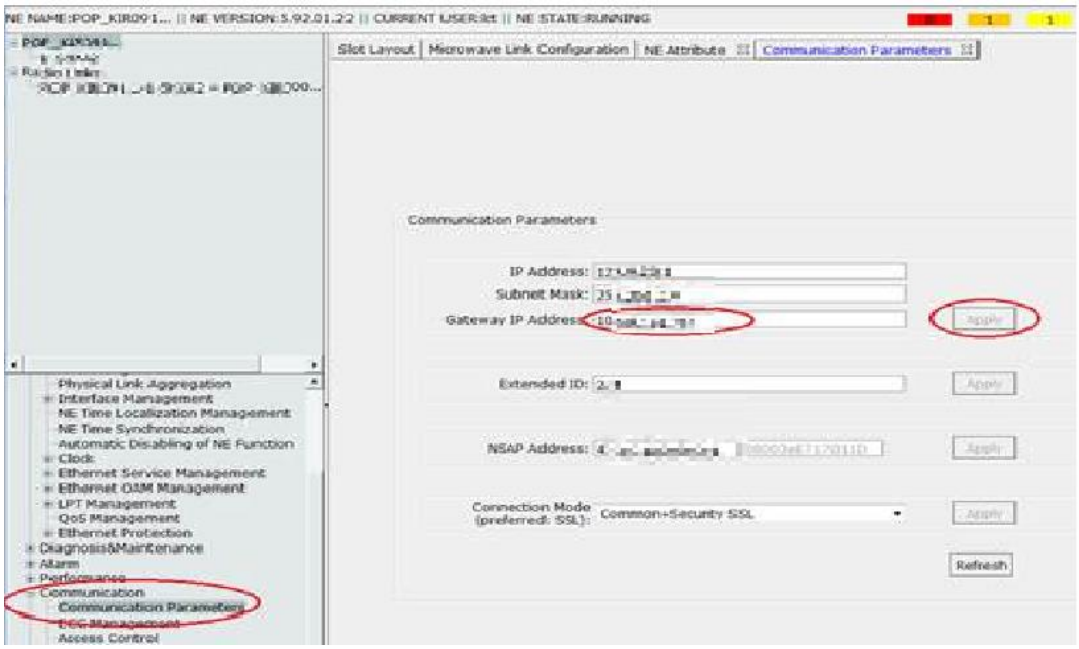


Рисунок 2.3 – Gateway IP Address

Потім необхідно встановити номер керуючого VLAN в меню Communication/ DCN Management/ Bandwidth Management (рис. 2.4).

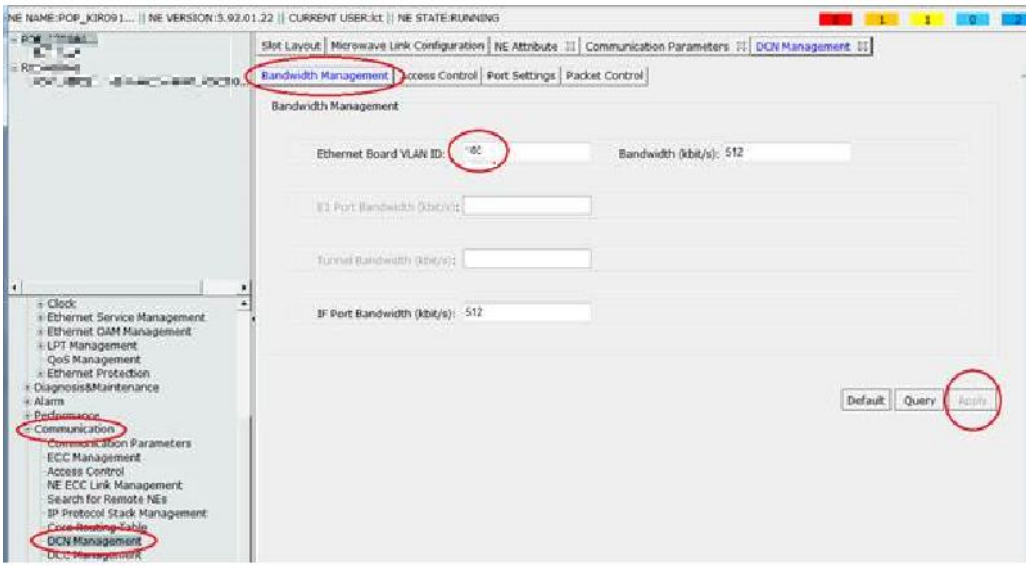


Рисунок 2.4 – Керування VLAN

В «Communication/ DCN Management/ Access Control» – потрібно перевести Port 3 в «enabled» і прописати два параметри: IP-address та Subnet Mask (рис. 2.5).

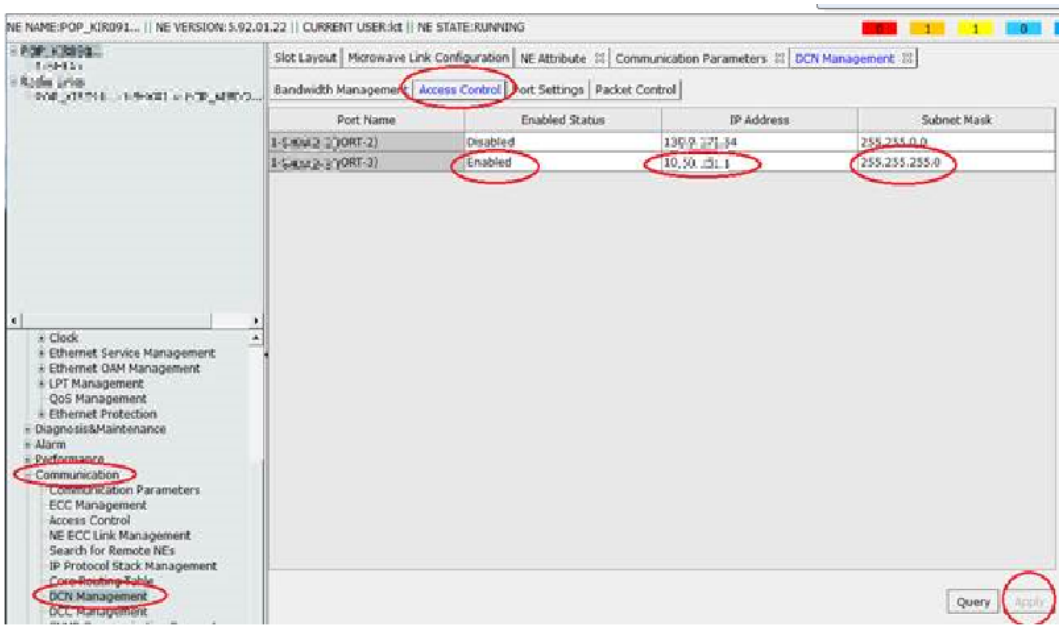


Рисунок 2.5 – IP-address та Subnet Mask

В останню чергу в «Configuration / NE Attribute» необхідно прописати три параметри: ID, Extended ID, Name (рис. 2.6). Для зміни параметрів ID та Extended ID використовується кнопка «Modify NE ID».

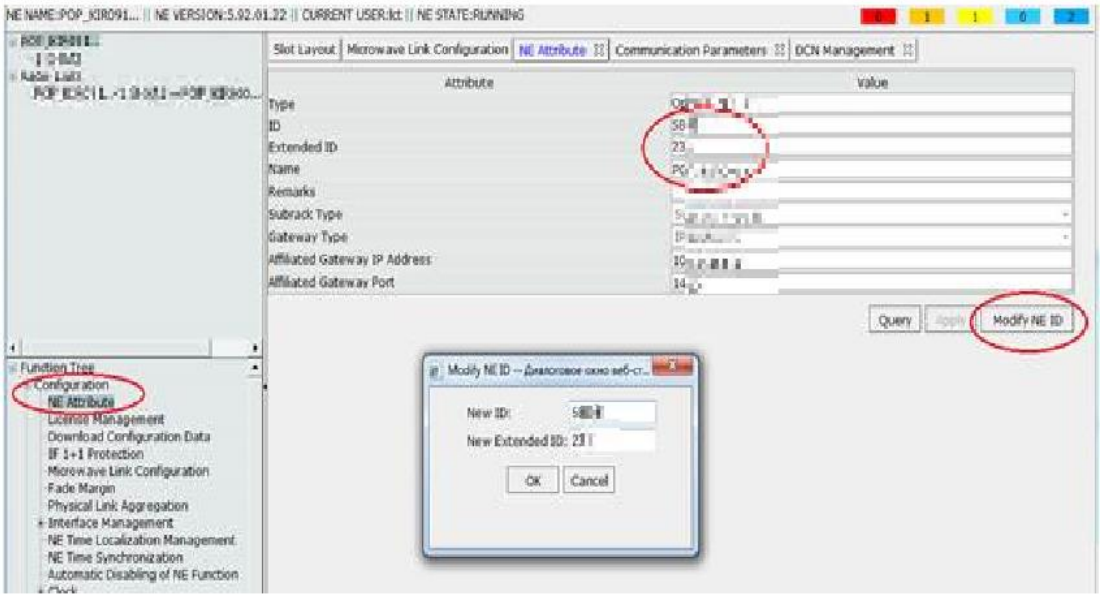


Рисунок 2.6 – ID, Extended ID, Name

Також в «Configuration/ Ethernet Service Management/ E-LAN Service» треба виділити рядок auto_configuration і видали її. Для цього натиснути кнопку «Delete» в правій стороні вікна (рис. 2.7).

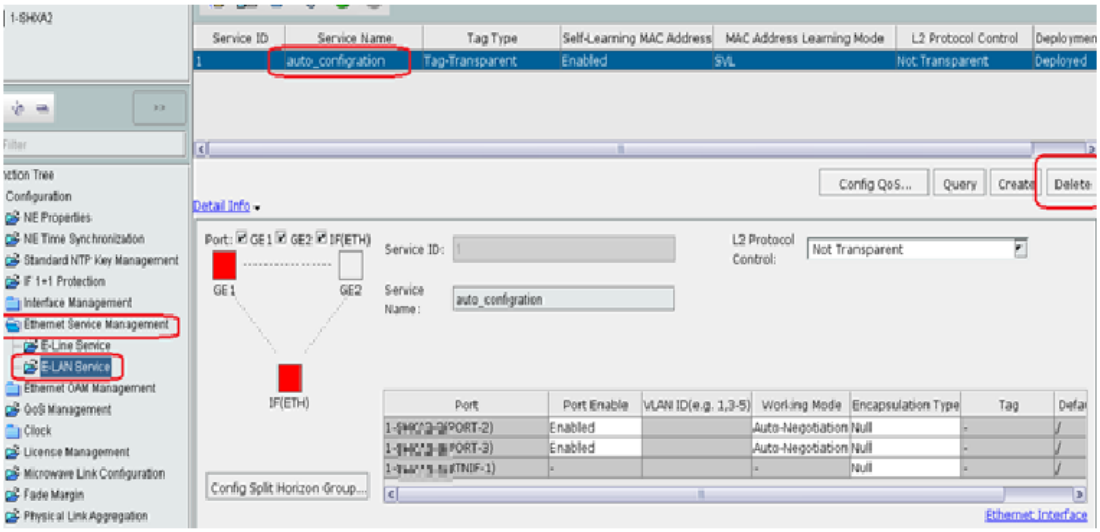


Рисунок 2.7 – E-LAN Service

Щоб застосувати зміни, потрібно перезавантажити обладнання, а саме правою кнопкою обрати «Warm reset» або «Cold reset» (рис. 2.8) та підтвердити цей запит двічі.

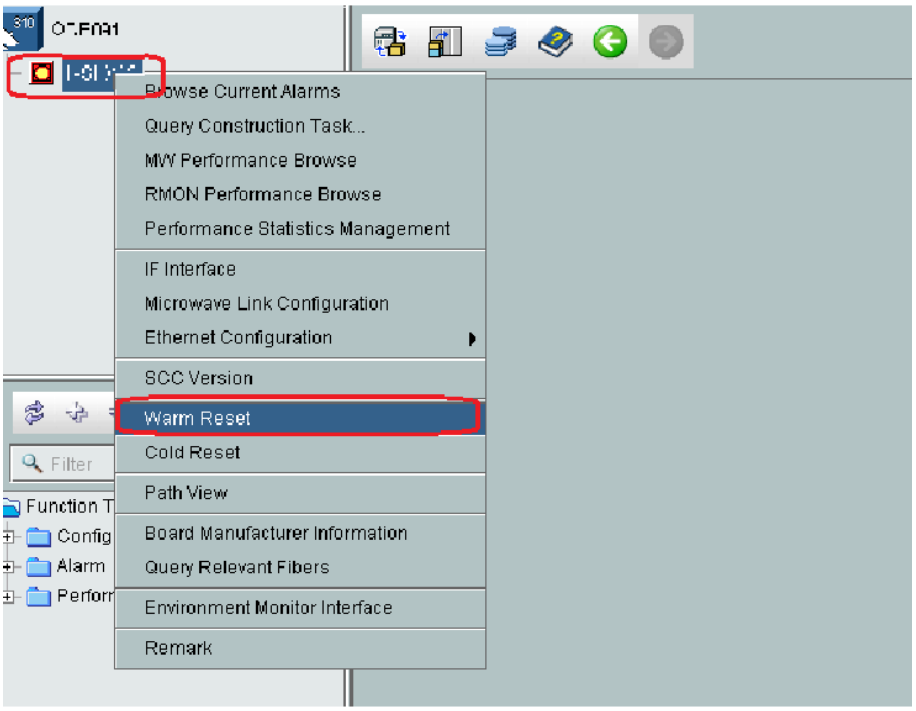


Рисунок 2.8 – Виконання Reset

2.2.2 Приклад налаштування обладнання радіорелейної станції Ericsson ML6366

Для доступу до налаштування IP, Mask та Gateway потрібно авторизуватись по https та в контекстному меню обрати пункти Configure – Basic NE (рис. 2.9).

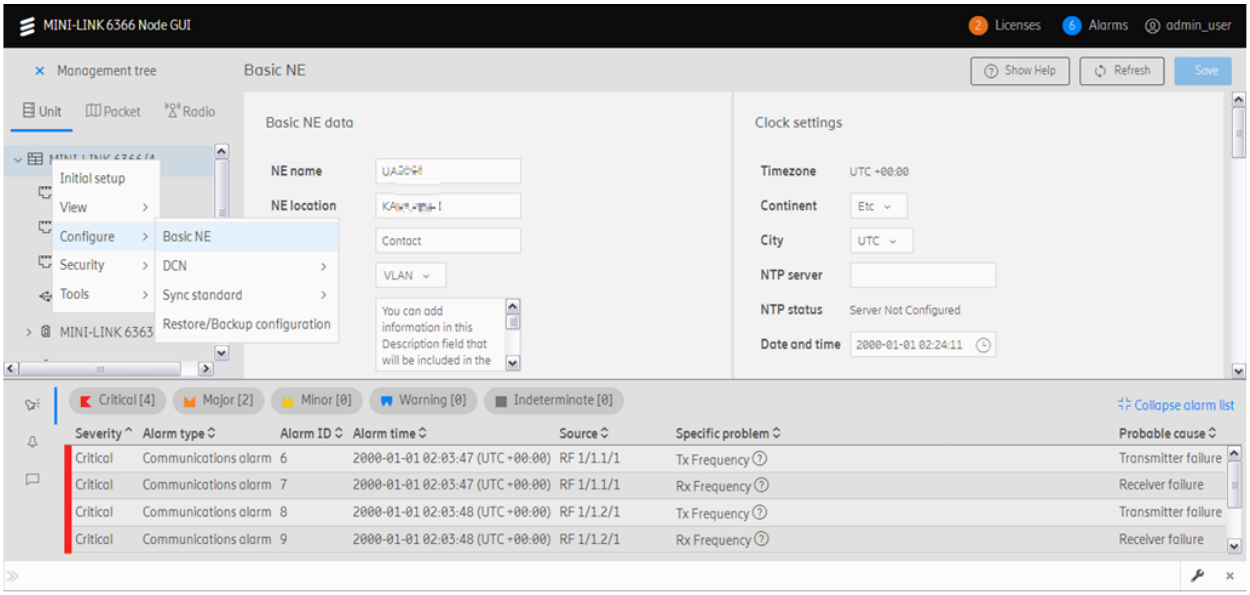


Рисунок 2.9 – Basic NE

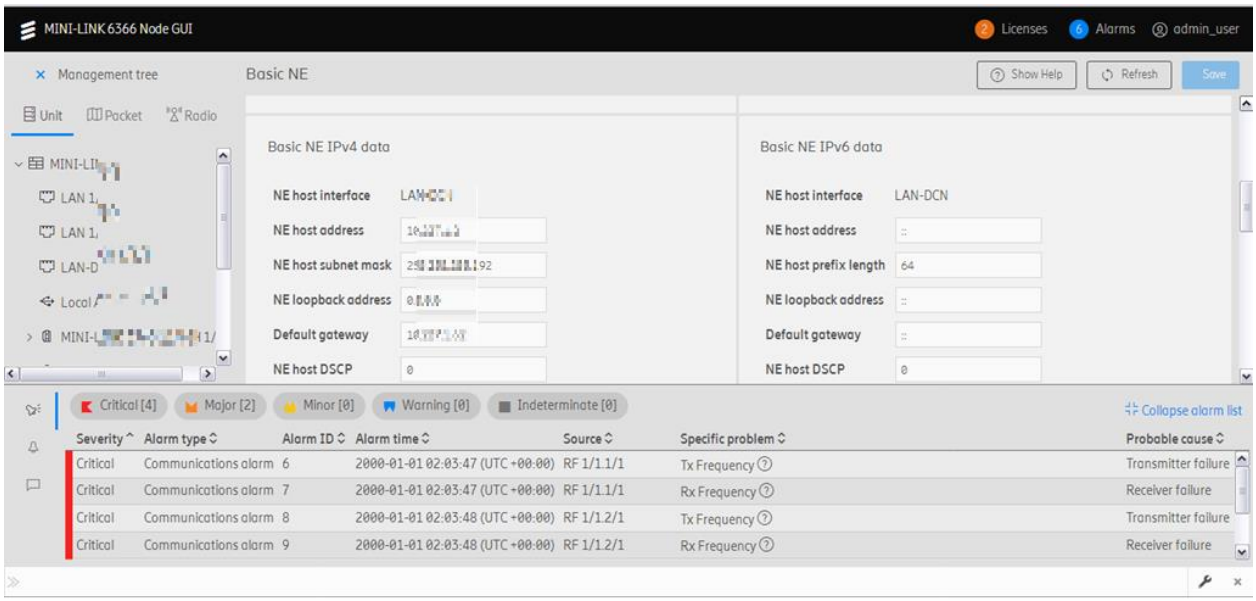
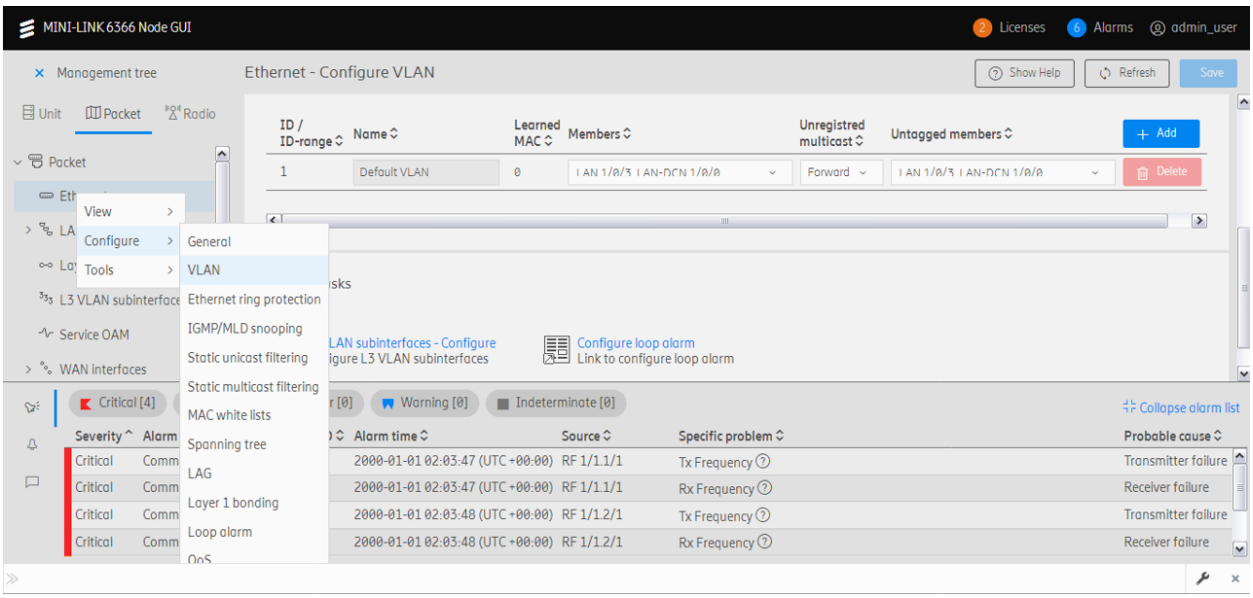
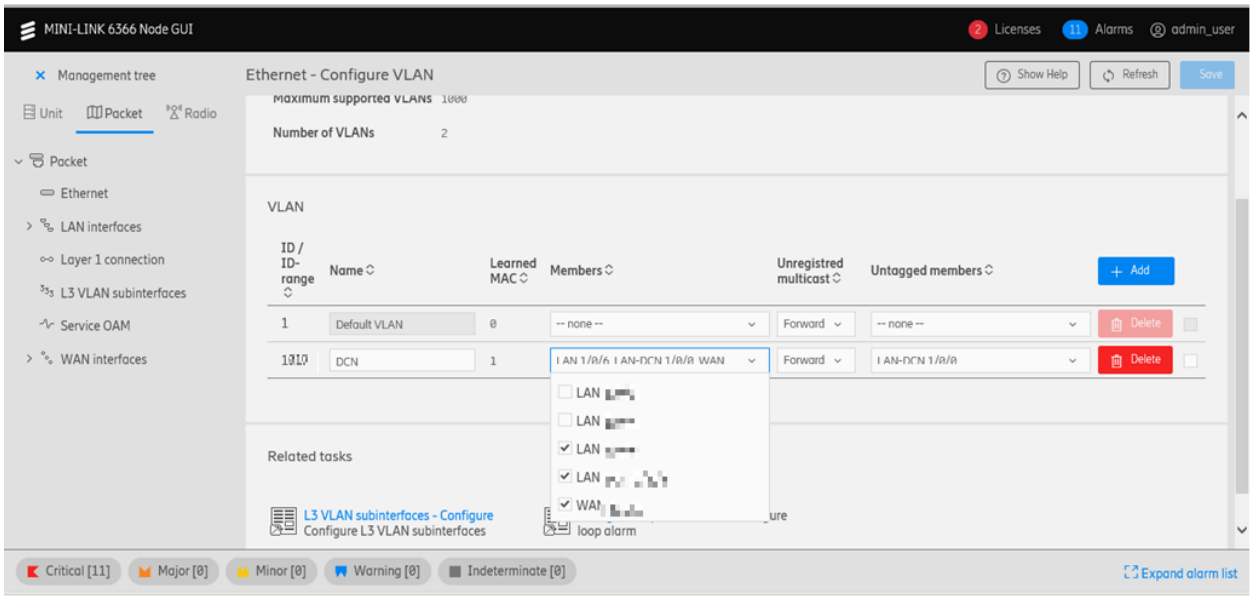


Рисунок 2.10 – Налаштування IP, Mask та Gateway

Наступні снепшоти представляють налаштування VLAN керування.



а)



б)

Рисунок 2.11 – Налаштування VLAN

Як висновок можна зазначити, що представлене ПЗ має інтуїтивно зрозумілий і зручний інтерфейс. Водночас використання GUI може обмежити можливості автоматизації процесів у випадках, коли, наприклад, потрібно швидко налаштувати декілька інтерфейсів обладнання [3, 8 – 10]. Це може бути досягнуто програмуванням однотипних операцій та їх виконанням в CLI за наявності одночасного віддаленого доступу до необхідного обладнання.

Деякі зображення параметрів «розмиті» з міркувань безпеки, оскільки приклади налаштувань були взяті з діючого обладнання операторів зв’язку.

3 ОГЛЯД ІНСТРУМЕНТІВ ТА МОВ ПРОГРАМУВАННЯ, ЩО ВИКОРИСТОВУЮТЬСЯ ПІД ЧАС АВТОМАТИЗАЦІЇ МЕРЕЖ

Серед найбільш популярних інструментів і мов програмування для атоматизації мереж можна виділити наступні [11].

1) Ansible являє собою ефективну платформу автоматизації, яка дозволяє користувачам визначати інфраструктуру у вигляді коду за допомогою простих плейбуків, що базуються на YAML. Зазвичай використовується для управління конфігурацією, розгортання додатків та оркестрування завдань автоматизації мережі.

2) Cisco DNA Center є централізованою платформою для управління спільнотою та автоматизації, яка надає можливості для роботи в мережі, що ґрунтуються насамперед на раціональному підході. Вона дозволяє автоматизувати забезпечення спільноти, забезпечення покриття та гарантії на всіх пристроях мережі Cisco.

3) Juniper Networks Junos Automation пропонує набір засобів автоматизації для роботи з пристроями Junos, включаючи сценарії Junos OS, сценарії Junos Automation і Junos PyEZ (бібліотека Python для автоматизації Junos).

4) JavaScript регулярно використовується для засобів і фреймворків автоматизації мережі, а також для розробки користувацьких сценаріїв автоматизації та програм, які взаємодіють з мережними API та інтернет-інтерфейсами.

5) Python є однією з найпоширеніших мов програмування в мережній автоматизації завдяки своїй простоті, зрозумілості та великим бібліотекам мережних функцій.

Проте, далі піде мова саме про ті інструменти і мови програмування, які було обрано як стек в даній роботі для розробки відповідних рішень автоматизації:

– скрипту опитування портів;

– комплексного рішення використання скриптів з графічним вебінтерфейсом.

3.1 Архітектура розробленого програмного забезпечення та використані технології

В даній роботі створено, протестовано та надано функціональний опис програмного забеспечення (ПЗ), що дозволяє оператору мережі виконувати виробничі завдання на обладнанні 7250 Interconnect Router фірми Nokia, що представляє собою маршрутизатор IP/MPLS, як із застосуванням командного рядку (CLI), так і з використанням графічного вебінтерфейсу (GUI).

На рисунку 3.1 наведено діаграму роботи ПЗ та застосовані технології.

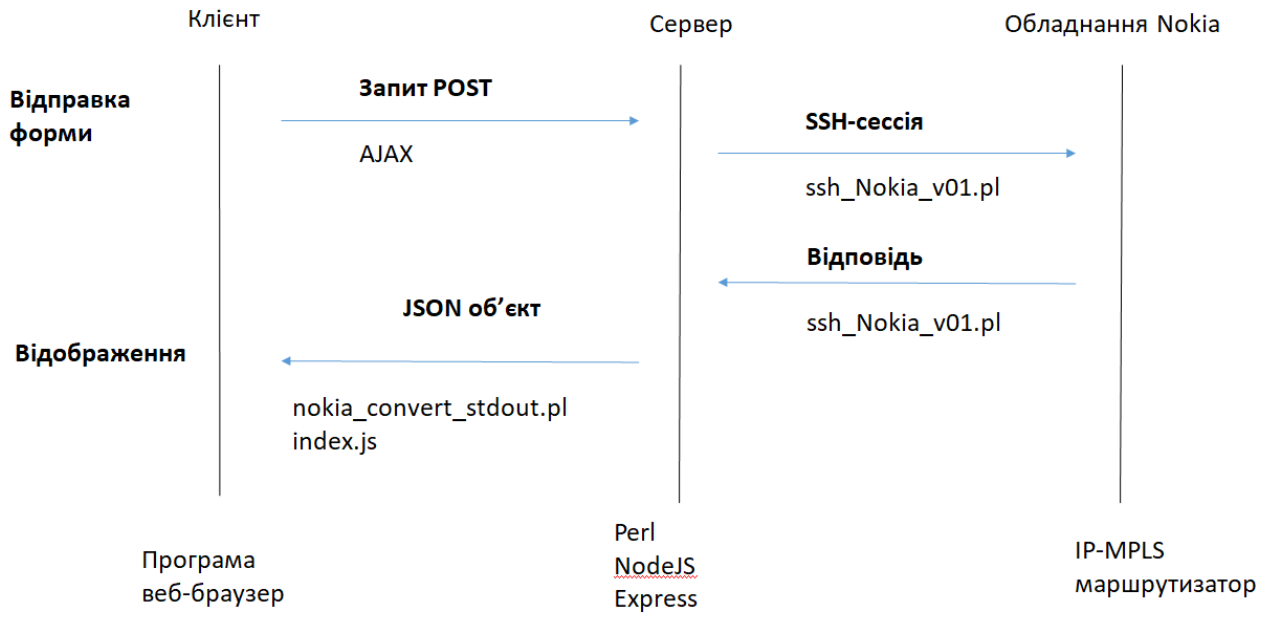


Рисунок 3.1 – Діаграма роботи ПЗ та застосовані технології

3.2 Вибір мови програмування для написання скрипту та ядра програмного забезпечення

Під час розробки архітектури розроблюваного ПЗ постало питання вибору мови програмного забезпечення. В процесі аналізу існуючих мов було обрано два можливі варінти: Perl та Python [12 – 15].

Perl – це високорівнева, універсальна, інтерпретована, динамічна мова програмування, що добре підходить для мережної автоматизації. Вона надає потужні можливості обробки тексту і має велику колекцію сторонніх модулів, зокрема для мережної та системної інтеграції. Скрипти Perl можуть автоматизувати

такі завдання, як моніторинг продуктивності мережі, управління конфігураціями та обробка файлів журналів, що робить її цінним інструментом для системних адміністраторів та мережних інженерів.

Perl часто порівнюють з Python, іншою популярною мовою сценаріїв для мережної автоматизації. Perl традиційно користується популярністю завдяки своїм можливостям, згаданим вище. Однак Python здобув популярність завдяки своїй читабельності та простоті використання, а також фреймворкам, таким як Django, для веброзробки. Обидві мови мають свої сильні сторони, і вибір між ними часто зводиться до конкретних вимог проєкту.

3.2.1 Характеристика Perl у застосуванні до мережної автоматизації

Серед основних можливостей мови Perl можна виділити наступні [12 – 15].

1) Perl пропонує процедурне програмування зі змінними, виразами, блоками коду, підпрограмами тощо.

2) Має багато вбудованих функцій для підтримки обробки тексту та функцій операційної системи.

3) Завдання управління даними можна вирішувати за допомогою асоціативних масивів.

4) Це дуже виразна мова, тому навіть для великих програм код, написаний на Perl, є коротким.

5) Perl, який зараз відноситься до своєї останньої версії, Perl 5 – це мова CGI-скриптів, яка може бути використана в мережному програмуванні, системному адмініструванні тощо.

6) У Perl 5 додано функції для підтримки структур даних, об’єктно-орієнтованого програмування тощо.

7) Код, написаний на Raku, який спочатку був відомий як Perl 6, можна викликати з програми на Perl, і навпаки.

Водночас основні переваги Perl зазначаються таким чином [12 – 15].

1) Легше кодувати, оскільки не доводиться турбуватися про пробіли.

2) Дозволяє користувачеві писати один і той самий код у різних стилях.

3) Має вбудовані функції для обробки операцій на рівні операційної системи.

4) Дозволяє легко ідентифікувати змінні завдяки використанню перед ними символів '@', '%' тощо.

5) Операції, пов'язані з введенням/виведенням, виконуються набагато швидше за допомогою Perl.

6) За допомогою Perl можна легко створювати звіти.

7) Мова має потужні опції порівняння рядків, які допомагають писати швидкі та короткі коди.

Області застосування Perl, окрім зазначених на рис. 3.2, включають в себе [12 – 15]:

– використання для написання CGI-скриптів у великих проєктах, таких як Bugzilla, Splash, RT і т.д.; деяких дуже завантажених веб-сайтах, таких як IMDb, Live Journal, Slashdot і т.д.

– застосування як мови системного програмування в Debian (дистрибутив Linux).

– використання як мови сценаріїв для зв’язування системи та інтерфейсів разом, які в іншому випадку не є сумісними; обробки великих обсягів даних для таких завдань, як генерація звітів і т.д.

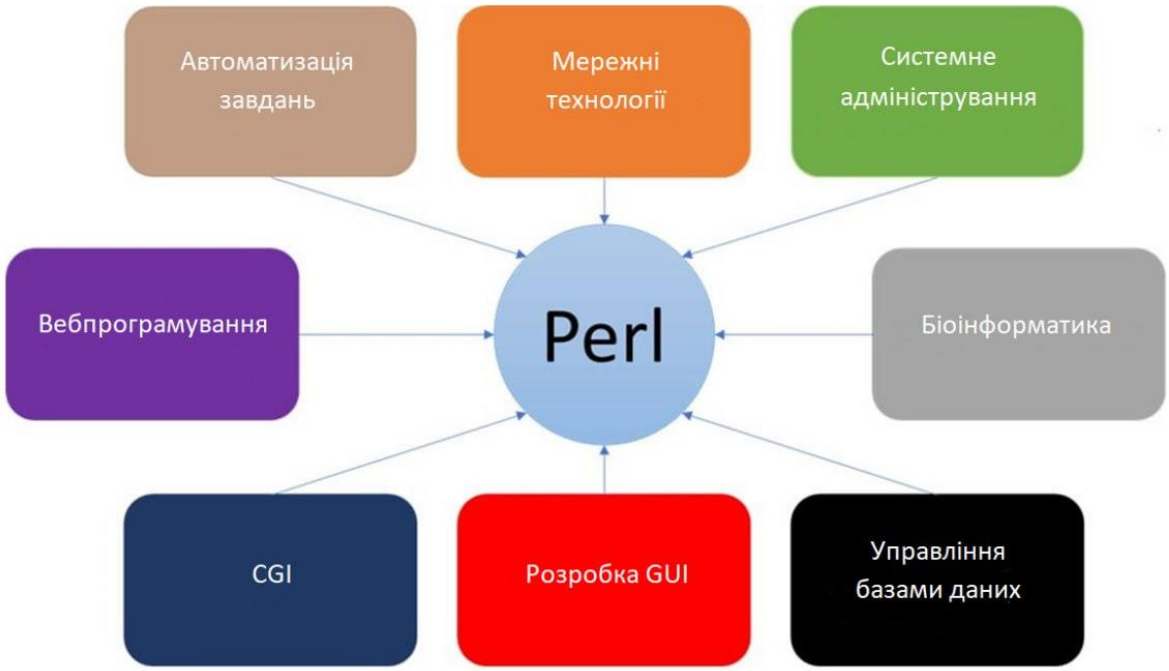


Рисунок 3.2. – Області застосування мови Perl

Отже, Perl – це універсальна та потужна мова програмування, яка вирізняється своєю розвиненою екосистемою та багатоцільовими можливостями. Як мова-оболонка, вона дозволяє безпосередньо взаємодіяти з операційною

системою, що робить її ідеальною для завдань системного адміністрування. Синтаксис Perl використовує дужки та цикли для функцій, забезпечуючи знайому структуру для тих, хто має досвід роботи з подібними мовами. Сильна сторона мови полягає в її пристосованості до різних потреб, чи то для швидких скриптів, чи то для масштабних додатків, що робить її найкращим вибором для багатьох розробників.

3.2.2 Характеристика Python у застосуванні до мережної автоматизації

Водночас до основних можливостей мови Python можна віднести наступні [12 – 15].

1) Її легко зрозуміти, вивчити та опанувати.

2) Налагоджувати код на Python легко завдяки простоті коду.

3) Код на Python можна запускати на різних операційних системах та апаратному забезпеченні.

4) Python дозволяє створювати складні коди, необхідні в робототехніці, штучному інтелекті тощо.

5) Python надає багато готових бібліотек, що полегшує кодування.

6) У Python можлива інтеграція баз даних з MySQL, Oracle тощо.

7) Python можна інтегрувати з іншими мовами програмування, такими як C, C++, Java тощо.

8) Забезпечує автоматичне «збирання сміття» (garbage collection).

Також зазначимо переваги Python для різних завдань автоматизації [12 – 15].

1) Ця мова має елегантний і чистий синтаксис, що робить її гарним вибором для початківців, які зацікавлені у вивченні мови програмування.

2) Кожен рядок коду не потрібно закінчувати символом ';' завдяки використанню пробілів та відступів.

3) З її допомогою можна легко створювати великі додатки та вебсайти.

4) Вона має вражаючу підтримку бібліотек, що робить сфери її використання широкими – наприклад, машинне навчання, великі дані, вебпрограмування, десктопні застосунки тощо.

5) Більші програми можна писати з меншою кількістю рядків коду.

6) Python є об’єктно-орієнтованою мовою програмування.

7) Perl є кращою мовою для написання CGI-скриптів, але Django і web2py також є популярними рішеннями.

8) Python має велику спільноту користувачів, які є активними та підтримують її, оскільки це мова з відкритим вихідним кодом.

Python став провідною мовою для автоматизації мереж завдяки своїй простоті та великої кількості бібліотек, доступних для автоматизації мережних завдань. За допомогою Python мережні інженери можуть легко писати сценарії складних завдань, керувати конфігураціями пристроїв та взаємодіяти з API для мережного програмування. Такі бібліотеки, як Netmiko, Paramiko та Ansible, дозволяють автоматизувати повторювані завдання на різних мережних пристроях, заощаджуючи час та зменшуючи кількість людських помилок. Зручність Python для читання та широка підтримка спільноти ще більше сприяють його популярності у сфері мережної автоматизації.

3.2.3 Приклад практичного застосування скрипту для автоматизації мережних завдань

Під час дослідження можливостей мов програмування було розроблено програму getDistance.py мовою Python, яка призначена для пошуку номерів базових станцій (Base Transceiver Station, BTS), розташованих на території – колі заданного радіусу, у мережах мобільного зв’язку. Основне завдання програми полягає у вимірюванні відстані між двома точками на сфері за формулою гаверсинусів [16 – 18], що було реалізовано за допомогою модуля haversine. Для інсталяції данного модуля необхідно виконати наступну команду:

```
pip install haversine
```

У випадку, що розглядається, двом точкам відповідають координати центра кола, які задаються, та координати базових станцій. Відповідність номерів базових станцій до їх географічних координат (у форматі десяткових градусів) зберігається в текстовому файлі coordinates.log.

Для корректної роботи програми необхідно задати два параметри:

– географічні координати центра кола (ключі -lat та -lon);
– його радіус, що вимірюється в км (ключ -r).

Обчислення відстаней між цими точками відбувається в циклі for. Далі перевіряється умова того, що отримана відстань менша за радіус кола. Таким

чином, можливо отримати (та роздрукувати в STDOUT) список номерів потрібних базових станцій BTS.

У подальшому такий список може бути використаний для автоматизації таких задач, як, наприклад:

– перевірка стану BTS;
– віддалене завантаження програмного забезпечення BTS;
– перезавантаження, вимкнення BTS тощо.

Отже, за умови наступних вихідних даних:

– координати центра кола дорівнюють LAT = 49.9805, LON = 36.2525;
– радіус кола – 3 км,

отримуємо наступний результат виконання програми (рис. 3.3).

```
python .\getDistance.py -lat 49.9805 -lon 36.2525 -r 3
BTS#    Distance (km)

1009    2.561245679607431
1028    0.7955474176666661
1035    1.491009177040893
1070    2.9918124622661995
1087    2.9228360429872455
1142    2.454094683531889
1146    1.9910209432844208
1236    1.5433990521839978
1293    2.0067559007189435
1371    1.864457710714527
1405    2.5020730036080683
3758    2.0061093009218
3759    1.9803179481687634
3901    0.8754090859594472
3947    0.7955474176666661
5602    2.0061093009218
5751    1.3221775076414382
5953    2.7110218678846665
602     2.002289227888395
6028    0.7572912932257677
6070    2.9918124622661995
6087    2.901778742693847
6146    1.9803179481687634
6293    1.998107837361528
640     0.6678466225740406
751     1.4037953661158136
775     0.8754090859594472
848     1.4975183518866366
953     2.72422614242918
```

Рисунок 3.3 – Результат обчислення номерів базових станцій і відповідних відстаней

Таким чином, розроблений на мові Python програмний застосунок був апробований для обчислення номерів базових станцій і відстаней до них, і довів свою працездатність для застосування у мережах мобільного зв’язку. Зі свого боку формула гаверсинусів стала математичною основою застосунку.

Зі свого боку працездатність скриптів і програмного забезпечення, створеного за допомогою мови Perl, буде продемонстровано у наступному розділі, оскільки саме цю мову було обрано для подальшого використання та розробки.

3.3 Технології клієнтської частини застосунку

3.3.1 Особливості використання AJAX

В процесі розробки вебзастосунку було реалізовано клієнт-серверну архітектуру взаємодії. Водночас для забеспечення відображення результатів використовувались технології AJAX та платформи Node.js мови програмування JavaScript.

AJAX – це технологія, що використовується у вебпрограмуванні для створення швидких і динамічних вебсторінок [19]. Вона дозволяє асинхронно оновлювати вміст вебсторінки, обмінюючись невеликими обсягами даних з сервером у фоновому режимі, без необхідності перезавантажувати всю сторінку. Це призводить до більш плавного та інтерактивного користувацького сприйняття. AJAX зазвичай використовується в мережних і вебзастосунках для отримання даних з сервера після завантаження сторінки, відправки даних на сервер у фоновому режимі і динамічного оновлення частин вебсторінки.

AJAX працює за допомогою об’єкта XMLHttpRequest в JavaScript, який дозволяє веб-сторінкам надсилати та отримувати дані з сервера асинхронно, не втручаючись у відображення та поведінку існуючої сторінки. Далі наведено спрощений опис того, як працює AJAX [19].

1) На вебсторінці відбувається подія (наприклад, завантаження сторінки або натискання кнопки).

2) Об’єкт XMLHttpRequest створюється за допомогою JavaScript.

3) Об’єкт XMLHttpRequest надсилає запит на веб-сервер.

4) Сервер обробляє запит.

5) Сервер надсилає відповідь у зворотньому напрямку на вебсторінку.

6) Відповідь зчитується JavaScript.

7) Відповідні дії (наприклад, оновлення сторінки) виконуються JavaScript.

За допомогою AJAX оновлюється лише необхідна частина вебсторінки, що може покращити продуктивність та зручність вебпрограм за рахунок зменшення обсягу даних, що передаються між клієнтом та сервером, та пришвидшення часу відгуку.

3.3.2 Формати надсилання та отримання структурованих даних

AJAX може працювати з декількома форматами даних при обміні даними між вебсервером і клієнтом [19]:

– прості текстові дані (Plain Text), які можуть включати слова або речення;

– мова розмітки XML, яка визначає набір правил для кодування документів у форматі, придатному як для читання людиною, так і для машинного зчитування;

– полегшений формат обміну даними JSON, який легко читати і писати людиною, а також легко аналізувати і генерувати за допомогою комп’ютера;

– стандартна мова розмітки для документів HTML, призначених для відображення у браузері.

Ці формати забезпечують гнучкість у структуруванні та передачі даних в операціях AJAX, що дозволяє розробникам вибирати найбільш підходящий формат для своїх конкретних потреб.

Розглянемо більш детально формат JSON, оскільки саме він планується для використання (рис. 3.1).

JSON – це формат обміну даними, який легко читати і записувати, а також легко аналізувати і генерувати [20]. Він базується на підмножині мови JavaScript, але є незалежним від мови, оскільки парсери доступні для багатьох мов.

Для мережної автоматизації JSON має кілька наступних переваг.

1) Формат зрозумілий і простий, що дозволяє легко зрозуміти структуру і дані.

2) JSON менш багатослівний, ніж інші формати, такі як XML, а це означає, що його можна швидко аналізувати і генерувати, заощаджуючи час і ресурси.

3) JSON підтримується багатьма мовами програмування, що робить його універсальним вибором для інструментів і скриптів мережної автоматизації.

4) Формат JSON дозволяє представляти складні структури даних, включаючи вкладені масиви та об’єкти.

5) JSON є стандартом де-факто в мережній індустрії, підтримується багатьма API та інструментами для конфігурації.

Ці особливості роблять JSON популярним вибором для конфігурації пристроїв, обміну даними між системами та автоматизації завдань управління мережею.

Припустимо, необхідно налаштувати інтерфейс мережного пристрою. В цьому випадку можна використовувати JSON для визначення параметрів конфігурації, які можуть виглядати так, як показано на рис. 3.4.

```
{
  "interface": {
    "name": "GigabitEthernet0/1",
    "description": "Uplink to Router",
    "status": "up",
    "ipv4_address": "192.168.1.1",
    "ipv4_mask": "255.255.255.0"
  }
}
```

Рисунок 3.4 – Приклад JSON-об’єкта для опису параметрів інтерфейсу

У сценарії мережної автоматизації за допомогою цього JSON-об’єкта можна проаналізувати параметри конфігурації, а потім використати їх для налаштування інтерфейсу пристрою за допомогою API або інструменту керування конфігурацією, наприклад, Ansible.

Ця структура JSON зрозуміла і проста, її можна легко змінити або розширити, щоб включити додаткові налаштування або інтерфейси.

3.3.3 Використання JavaScript/DOM для відображення інформації та взаємодії з нею

JavaScript є мовою програмування, яка дозволяє створювати інтерактивні веб-сторінки. Це невід'ємна частина вебзастосунків, що дозволяє користувачам взаємодіяти з вебсторінками в режимі реального часу без необхідності перезавантаження сторінки.

DOM (Document Object Model) – це програмний інтерфейс для вебдокументів. Вона представляє сторінку так, щоб програми могли змінювати структуру, стиль і зміст документа. DOM представляє документ у вигляді дерева об'єктів; JavaScript може взаємодіяти з цими об'єктами для модифікації HTML і CSS, таким чином динамічно оновлюючи зовнішній вигляд і вміст вебсторінки.

Отже, JavaScript використовує DOM для динамічного відображення та взаємодії з інформацією на вебсторінці, що дозволяє оновлювати вміст, створювати візуальні ефекти, перевіряти форми та обробляти події користувача, такі як кліки або натискання клавіш.

3.4 Технології серверної частини застосунку

Зі свого боку Node.js є ефективним середовищем виконання JavaScript, побудованим на JavaScript-движку V8 для браузера Chrome. Він призначений для створення масштабованих мережних застосунків і може обробляти багато з'єднань одночасно, що робить його чудовим вибором для веброзробки, особливо в контексті мережної автоматизації.

У мережній автоматизації Node.js можна використовувати для створення вебінструментів та інтерфейсів для управління мережними завданнями. Його неблокуюча архітектура, керована подіями, дозволяє йому виконувати такі завдання, як читання з бази даних або запис до неї, виклик API або ефективне управління файловими системами, що є загальними вимогами в мережній автоматизації.

Розгалужена екосистема модулів Node.js, npm, включає такі бібліотеки, як netmiko-js та node-red, які спеціально розроблені для задач мережної автоматизації. Ці бібліотеки надають функції для автоматизації взаємодії з мережними пристроями, зокрема отримання статусу пристрою, оновлення конфігурацій або навіть написання сценаріїв складних мережних змін.

Використовуючи Node.js для мережної автоматизації, розробники можуть створювати надійні та ефективні вебзастосунки, які спрощують процеси управління мережею, зменшують кількість ручних операцій та підвищують загальну експлуатаційну ефективність.

3.5 Характеристика пропрієтарного програмного забезпечення Network Services Platform компанії Nokia

Для подальшого порівняння розроблюваного ПЗ з існуючими аналогами приведемо характеристику пропрієтарного програмного забезпечення Network Services Platform компанії Nokia [21].

Сервісна платформа управління мережею (NSP) допомагає автоматизувати IP, оптичні та радіорелейні мережі, щоб спростити роботу, швидко реагувати на попит, що швидко змінюється, отримувати максимальну віддачу від ресурсів і забезпечувати максимальну продуктивність і надійність обслуговування.

NSP забеспечує надійність управління за допомогою [21]:

– повного набору готових до використання функцій, які допомотають операторам і інженерним командам охопити всі випадки керування мережею, контролю та оптимізації;

– відкритої програмної платформи, яка дозволяє мережевим інженерам і розробникам створювати та адаптувати власні сценарії використання автоматизації та визначати налаштовані послуги, мережі та операції пристроїв.

Задачи, що охоплює NSP (рис. 3.5) [22]:

– конфігурація обладнання;
– надання мережних послуг;
– забезпечення надійності мережі;
– обчислення шляху;
– координація IP-оптики;
– мережна аналітика;
– поділ мережі на сегменти (slicing).

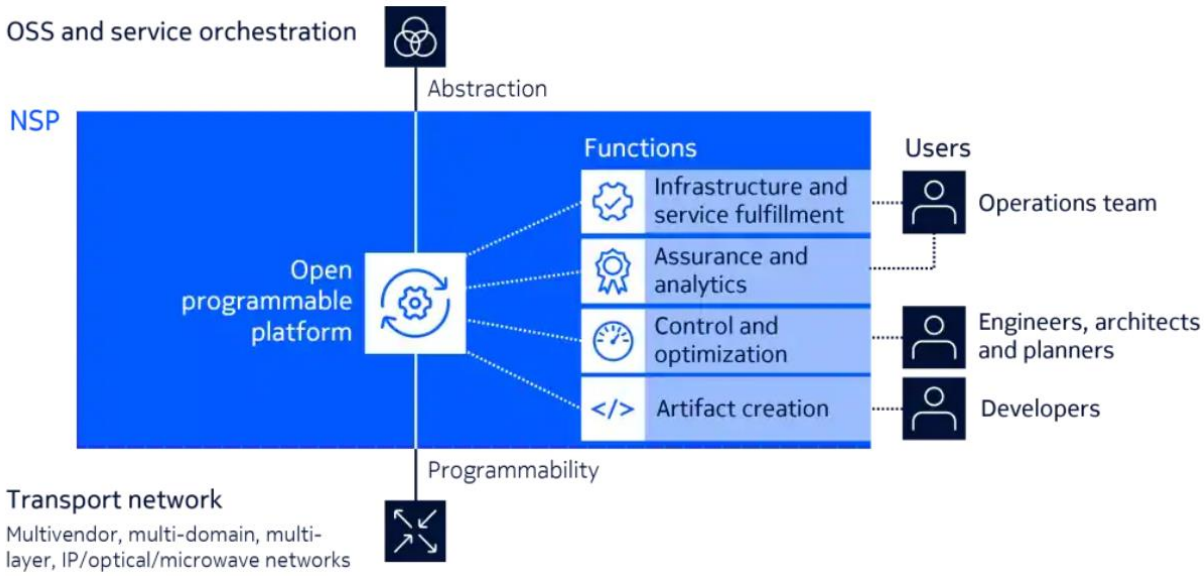


Рисунок 3.5 – Сервісна платформа управління мережею компанії Nokia [21, 22]

Платформа Nokia 7250 IXR підтримує консольний порт для підключення терміналів до пристрою. Також підтримується Ethernet-порт керування. Це дозволяє налаштовувати параметри з системної консолі по протоколам Telnet або SSH.

Приклад базового процесу налаштування:

– визначити слот шасі;

– вказати тип лінійної картки або тип IMM (якщо застосовно; має бути дозволений тип картки);

– визначити слот MDA (якщо є);

– указати MDA (якщо застосовно; має бути дозволений тип MDA);

– визначити конкретний порт для налаштування.

```
A:IXR6>config>card# info
----------------------------------------------
        card-type imm36-100g-qsfp28
        no shutdown
----------------------------------------------
A:IXR6>config>card#


A:IXR-R6>config>card# info
----------------------------------------------
        mda 5
            mda-type a4-choc3/12
            no shutdown
        exit
----------------------------------------------
A:IXR-R6>config>card#
```

Рисунок 3.6 – Приклад налаштування карти [21, 22]

Також наведемо приклад налаштування параметрів порту доступу Ethernet. Послуги мають бути налаштовані на портах доступу, що використовуються для корисного трафіку. SAP на порту налаштовується в режимі доступу. Коли порт налаштовано для режиму доступу, можна вказати тип інкапсуляції, щоб розрізняти служби на порті. Після того, як порт налаштовано для режиму доступу, на ньому можна налаштувати кілька служб.

```
*A:ZBG>config>port# info
----------------------------------------------
        ethernet
            mode access
            access
                egress
                exit
            exit
            encap-type dot1q
            mtu 9800
        exit
        no shutdown
----------------------------------------------
*A:ZBG>
```

Рисунок 3.7 – Приклад конфігурації порту доступу Ethernet

Таким чином, штатне ПЗ дозволяє здійснювати або високорівневе керування великими мережами за допомогою GUI NSP, або керування окремим маршрутизатором, але тільки в консольному режимі.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАВДАНЬ УПРАВЛІННЯ ТА МОНІТОРИНГУ В ТЕЛЕКОМУНІКАЦІЙНІЙ МЕРЕЖІ

4.1 Ядро програмного забезпечення

Зазначимо, що повну структуру каталогів проєкту наведено у додатку А. Скрипт ssh_Nokia_v01.pl (рис. 4.1) приймає в якості параметрів командного рядка IP-адресу маршрутизатора, ім’я користувача, номер порту, що опитується та здійснює їх перевірку в циклі for для масиву ARGV. Повний лістинг вихідного коду ssh_Nokia_v01.pl наведено у додатку Б.

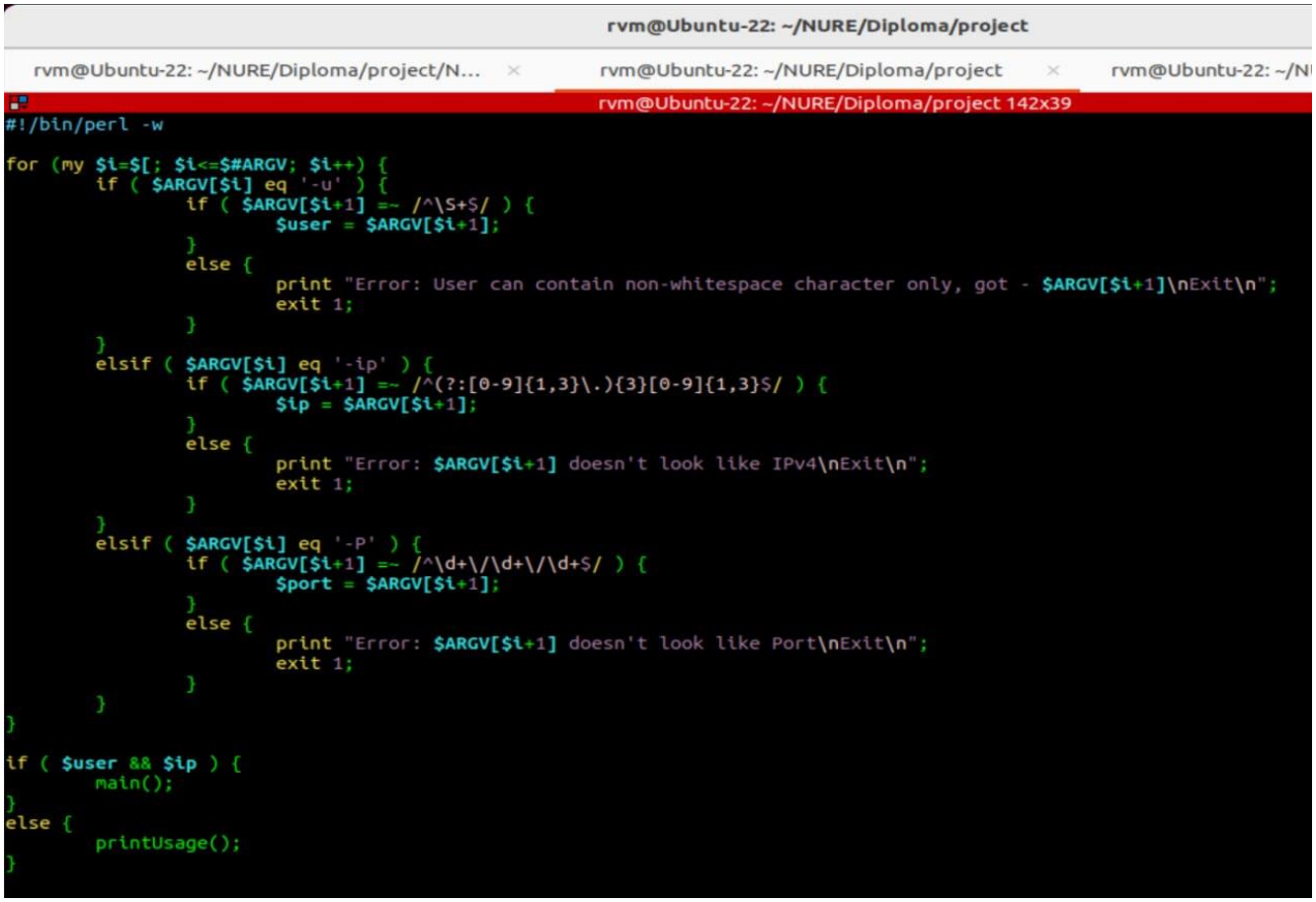


Рисунок 4.1 – Скрипт, що описує ключову функціональність застосунку

Далі скрипт здійснює перевірку доступності порту 22 на обладнанні та виконує команду show port <port> detail на маршрутизаторі, підключившись до нього за допомогою протоколу SSH (рис. 4.2).

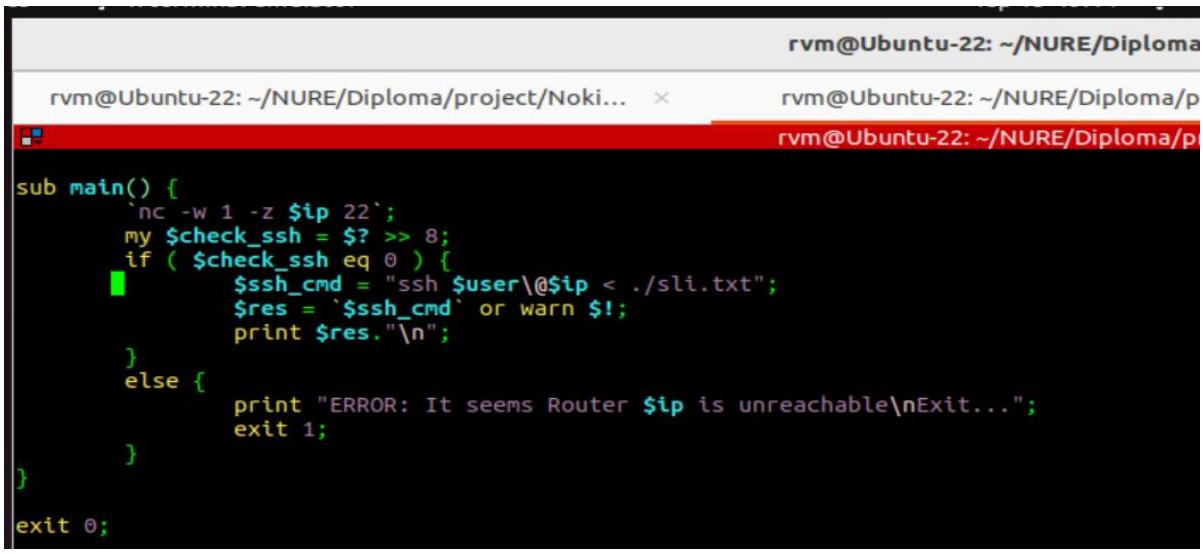


Рисунок 4.2 – Головна функція застосунку щодо перевірки доступності порту 22 на обладнанні

Приведемо приклад виконання цього скрипту за допомогою наступної команди в CLI:

./ssh_Nokia_v01.pl -ip 192.168.1.1 -u admin -P 1/1/1

Результат виконання цієї команди наведено на рисунку 4.3. Проте, повний результат виконання програми ssh_Nokia_v01.pl наведено в Додатку В.

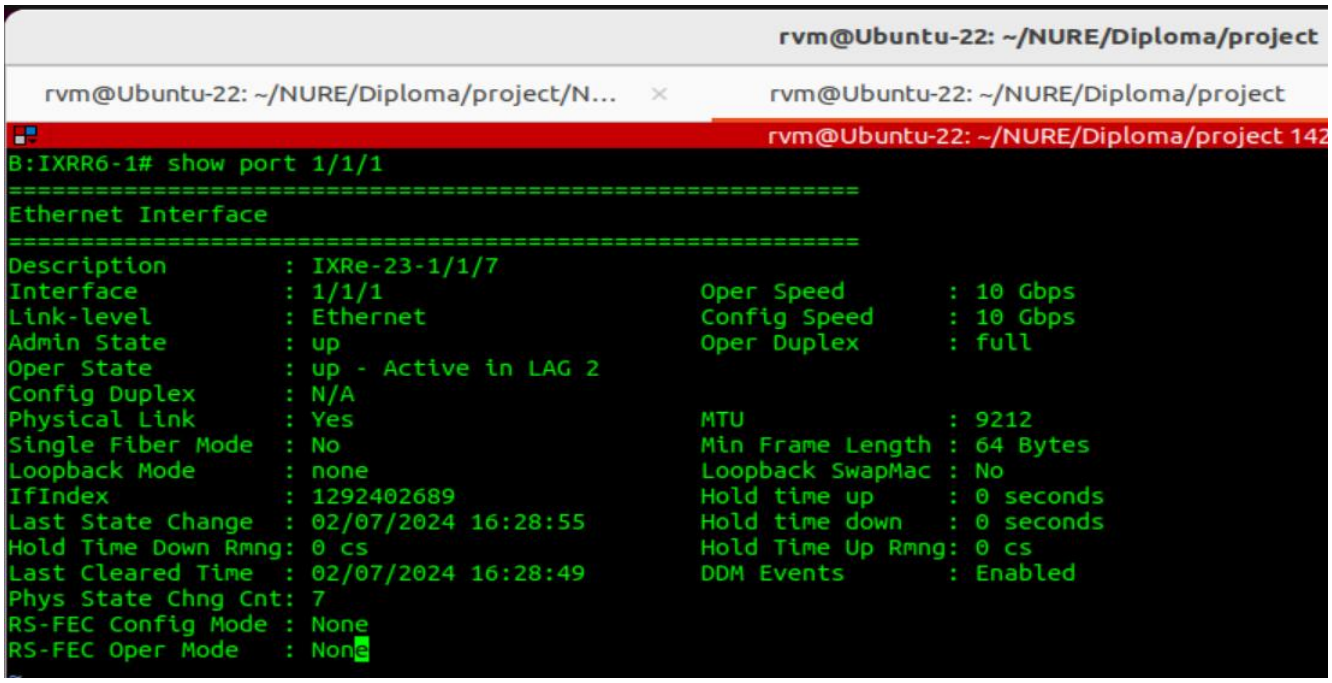


Рисунок 4.3 – Частковий результат виконання програми ssh_Nokia_v01.pl

Для подальшої обробки та відображення результату роботи даного скрипта у вікні браузера дані записуються в файл nokia_raw_data.txt. Наступним кроком скрипт nokia_convert_stdout.pl виконує аналіз файлу nokia_raw_data.txt і запис результату в файл render_obj.txt (рис. 4.4).

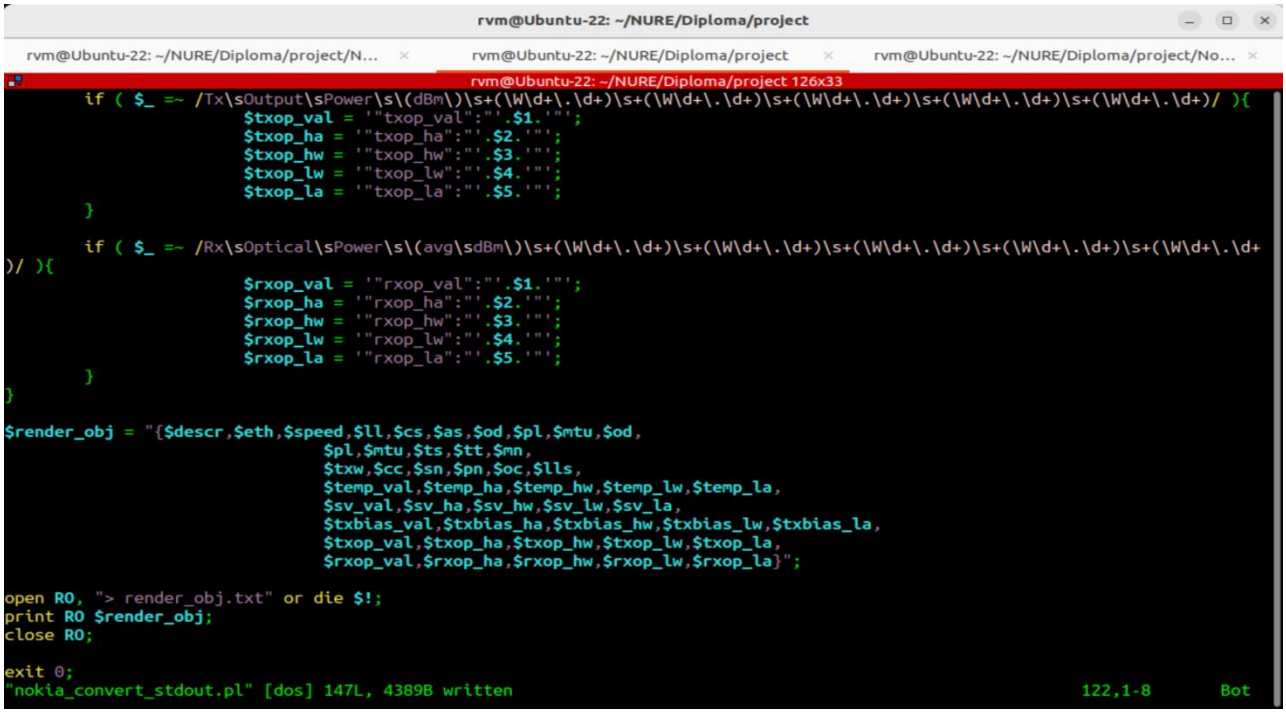


Рисунок 4.4 – Фрагмент програми обробки отриманних даних

Для написання вищезазначенних скриптів було обрано інтерпретатор Perl, так як він дозволяє ефективно виконувати команди CLI та, водночас, використовувати потужний механізм розбору текстових даних за допомогою регулярних виразів - perlre.

4.2 Клієнтська частина вебзастосунку

Для забеспечення відображення результатів в даній роботі було реалізовано схему взаємодії клієнт-сервер (рис. 3.1) на основі технології AJAX та платформи Node.js мови програмування Javascript.

У каталозі public було створено файли index.html та main.html з html-кодом сторінок, що включають js-файли клієнтської частини застосунку login.js та script.js відповідно (рис. 4.5 – 4.8).

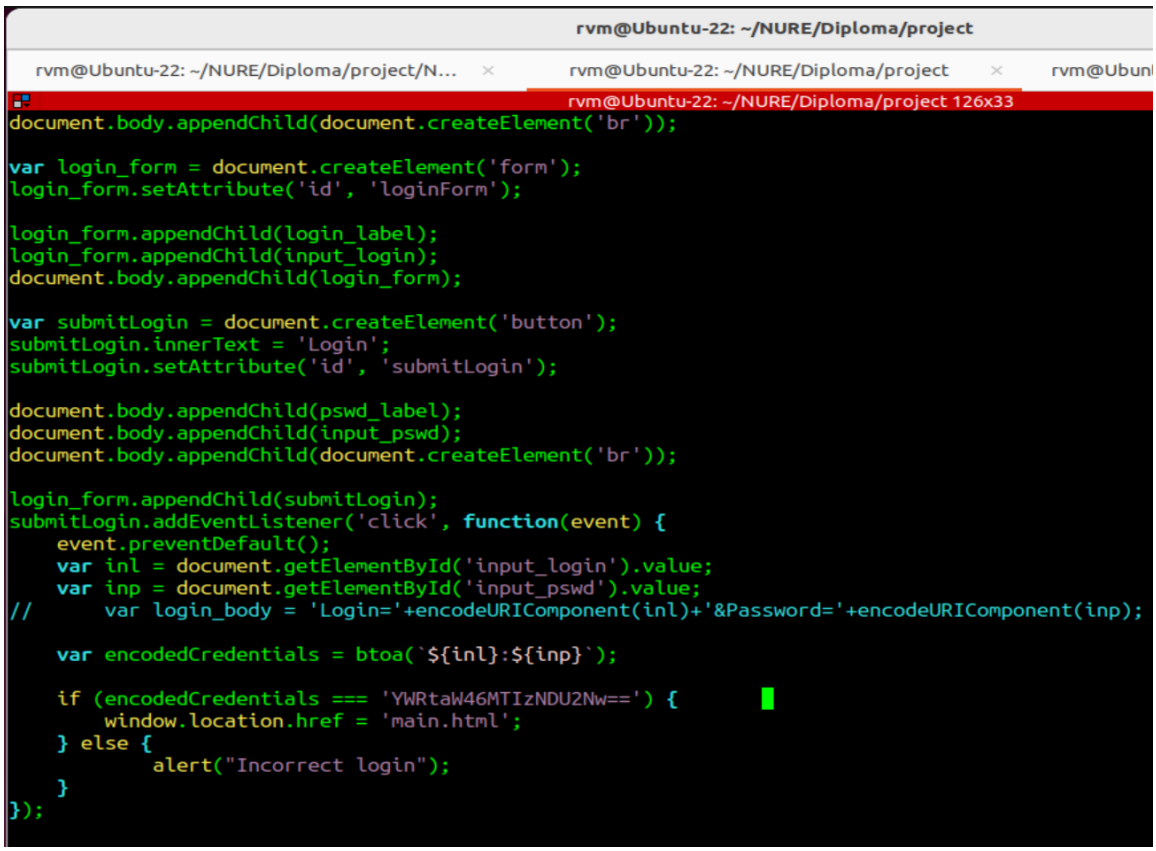


Рисунок 4.5 – Фрагмент вихідного коду файлу login.js

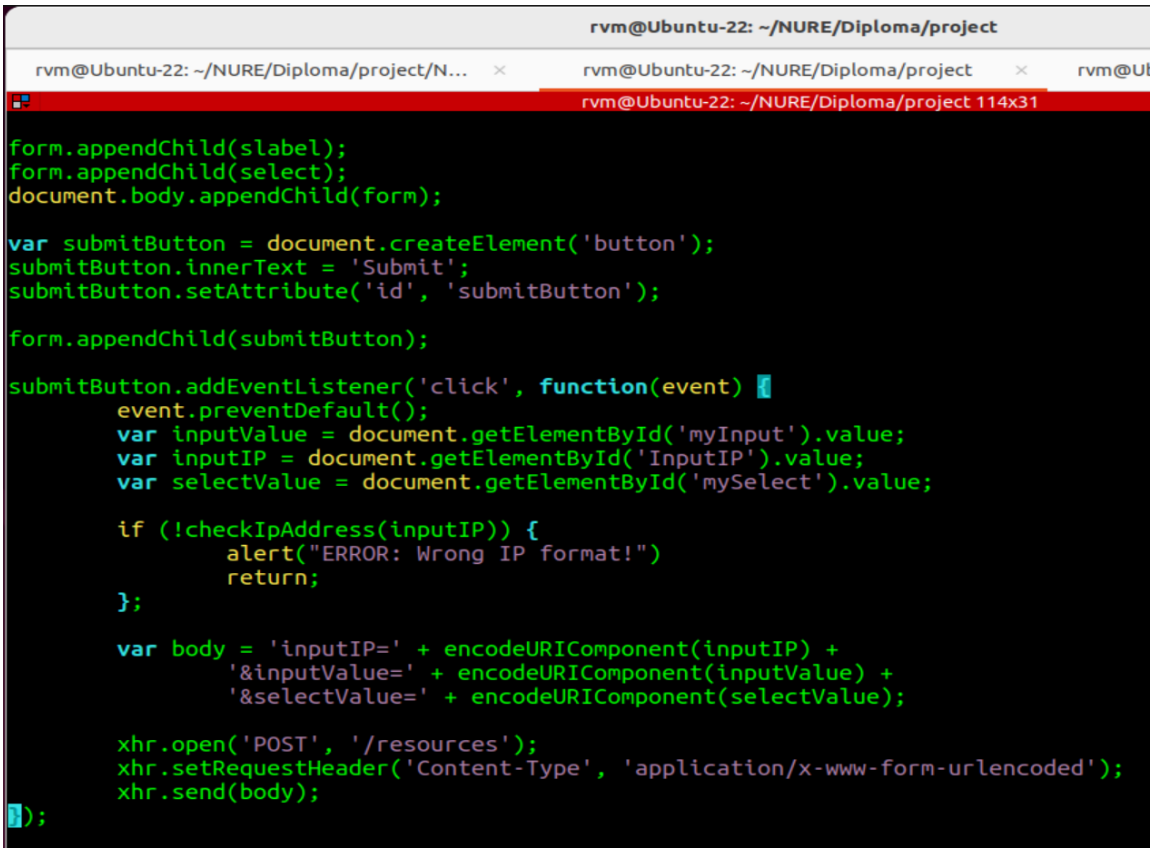


Рисунок 4.6 – Фрагмент файлу клієнтської частини script.js, що відправляє запити (request) на сервер

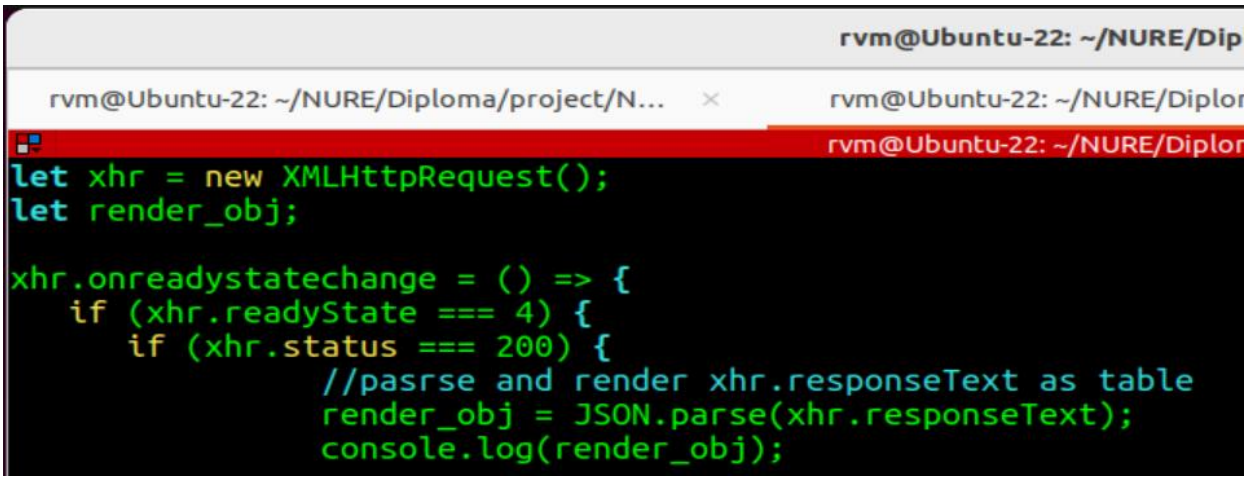


Рисунок 4.7 – Фрагмент файлу клієнтської частини script.js, що обробляє відповідь сервера (response)

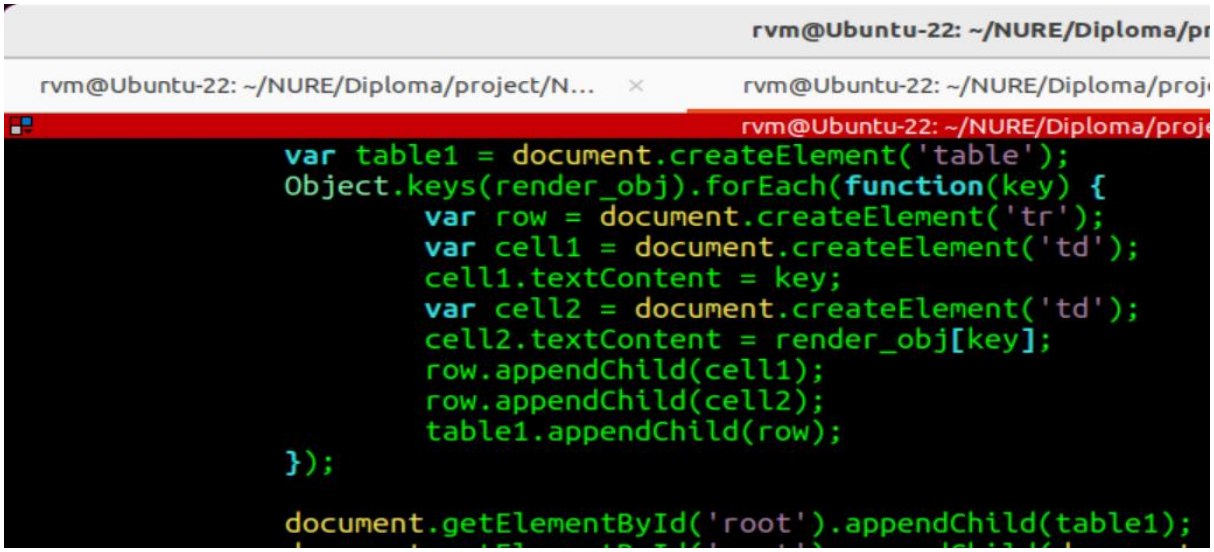


Рисунок 4.8 – Фрагмент файлу клієнтської частини script.js, що створює елементи DOM

Скрипт login.js, показаний на рис. 4.5, відповідає за перевірку облікових даних (рис. 4.9), що були введені користувачем.

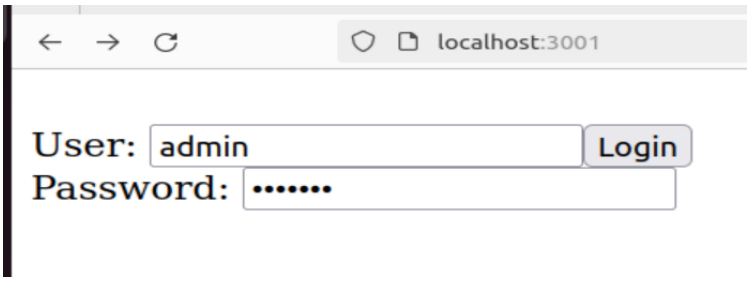


Рисунок 4.9 – Сторінка автентифікації

У разі введеня некоректних даних автентифікації буде сгенеровано відповідне повідомлення (рис. 4.10).

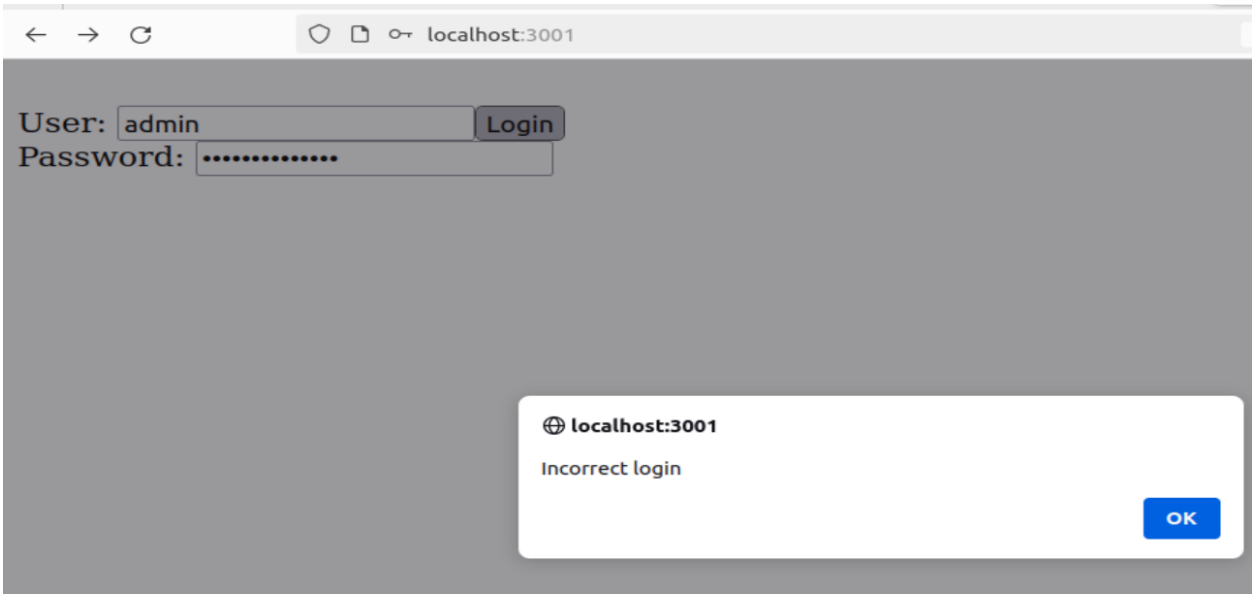


Рисунок 4.10 – Повідомлення під час провірки автентифікації

В разі успішної автентифікації користувачу буде доступна головна сторінка (рис. 4.11).

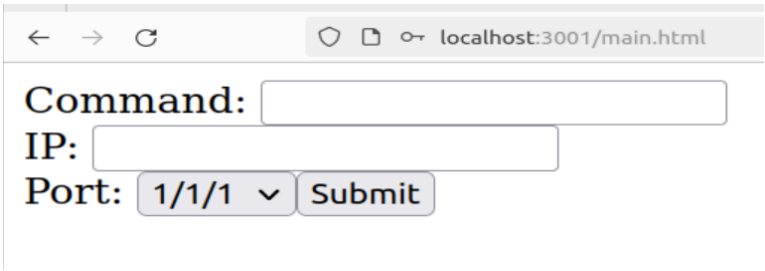


Рисунок 4.11 – Сторінка доступу до меню виконання команд

Після введення необхідних даних (рис. 4.12) користувач відправляє команду на виконання та отримує результат (рис. 4.13).

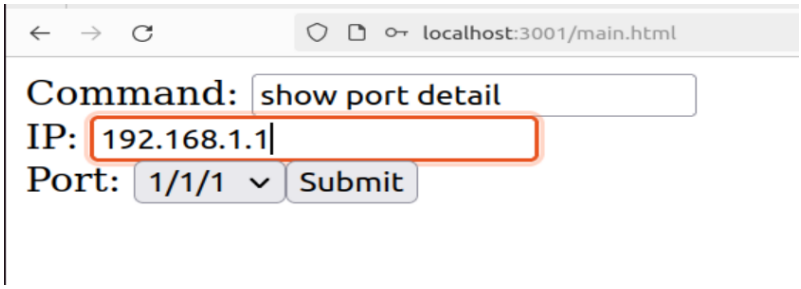


Рисунок 4.12 – Форма введення даних команди

Description	IXRX1-4-1/1/c4/1
Interface	1/1/1
Oper_Speed	10 Gbps
Link-level	Ethernet
Config_Speed	10 Gbps
Admin_State	up
Oper_Duplex	full
Physical_Link	Yes
MTU	9212
Transceiver_Status	operational
Transceiver_Type	SFP
Model_Number	3HE04823AAAA01 NOK IPU3ANKEAA
TX_Laser_Wavelength	1310 nm
Connector_Code	LC
Serial_Number	HM205100090679
Part_Number	RTXM228-401-C83
Optical_Compliance	10GBASE-LR
Link_Length_support	10km for SMF
temp_val	+41.6
temp_ha	+80.0
temp_hw	+70.0
temp_lw	+0.0
temp_la	-10.0
sv_val	3.34
sv_ha	3.63
sv_hw	3.47
sv_lw	3.14
sv_la	2.97
txbias_val	38.9
txbias_ha	80.0
txbias_hw	75.0
txbias_lw	15.0
txbias_la	10.0
txop_val	-1.39
txop_ha	2.50
txop_hw	0.50
txop_lw	-8.20
txop_la	-10.20
rxop_val	-2.45
rxop_ha	2.50
rxop_hw	0.50
rxop_lw	-14.40
rxop_la	-16.40

Рисунок 4.13 – Результат роботи кліентскої частини ПЗ

4.3 Серверна частина вебзастосунку

Серверна частина реалізована на платформі Node.js із використанням фреймфорку express та модулів fs і body-parser (рис. 4.14).

Рисунок 4.14 – Фрагмент файлу серверної частини index.js, що підключає необходібні фреймворки та модулі

Маршутизація HTTP-запитів здійснюється за допомогою метода post. Запис даних в локальний файл на сервері забеспечює функія writeFileSync (рис. 4.15).

Рисунок 4.15 – Фрагмент файлу серверної частини index.js, що забеспечує маршрутизацію кліентських викликів та запуск скрипту ssh_Nokia_v01.pl

Після старту сервер прослуховує порт 3001 (рис. 4.16).

```
[nodemon] starting `node index.js`
Server has been started at localhost:3001
```

Рисунок 4.16 – Повідомлення про старт Node.JS сервера

Виконання локальних програм на сервері можливо завдяки модулю node:child_process (рис. 4.17).

```
rvm@Ubuntu-22: ~/NURE/Diploma/project
rvm@Ubuntu-22: ~/NURE/Diploma/project/N...  rvm@Ubuntu-22: ~/NURE/Diploma/project  rvm@Ubuntu-22: ~/NURE/Di
rvm@Ubuntu-22: ~/NURE/Diploma/project 114x31
let rd_file = __dirname + "/../nokia_raw_data.txt";
const { spawn } = require('child_process');
const rd = spawn( __dirname + '/../ssh_Nokia_v01.pl',['-u', 'admin', '-ip', router_ip]);
rd.stdout.on('data', (data) => {
  console.log(`stdout: ${data}`);
  fs.writeFileSync(rd_file, data);
});
```

Рисунок 4.17 – Фрагмент файлу серверної частини index.js, що застосовує модуль node:child_process

Необхідно зазначити, что розроблене ПЗ було протестовано на лабораторній мережі компанії Nokia Bell NV, розташованій в м. Антверпен, Бельгія (реальні IP-адреси та облікові дані було змінено з міркувань безпеки).

5 ОЦІНКА ЕФЕКТИВНОСТІ АВТОМАТИЗАЦІЇ ЗАВДАНЬ ЩОДО УПРАВЛІННЯ МЕРЕЖЕЮ

5.1 Якісна характеристика розробленого програмного забезпечення

Отже, розроблене в ході виконання роботи програмне забезпечення:

– дозволяє виконувати команди оболонки, маючи мінімальні навички роботи з CLI;

– дозволяє працювати, використовуючи вебінтрефейс (GUI);

– має невилику вартість у порівнянні зі штатним пропрієтарним ПЗ.

Розроблене ПЗ призначено для використання персоналом ланки обслуговування та диспетчерських центрів.

Зазвичай інженерному персоналу сектору експлуатації мережі в процесі роботи доводиться звертатись за домогою до персоналу центрів управління (таких, як MSC, BSC тощо). Але такі задачи, як визначення статусу портів, змінення адміністративного стану портів, вимірювання параметрів ліній та інших, що не потребують поглибленних знань управління мережею, можна виконувати «на місцях» (виносні концетратори обладнання, приміщення вузлів агрегації тощо).

Наприклад, на те, щоб отримати дані про статус порта маршрутизатора необхідно зв’язатись з інженером центру управління, пояснити задачу, зачекати на відповідь, проаналізувати отримані дані. Залежно від завантаженності персоналу, вирішення однієї задачи може зайняти від 3 до 15 хвилин.

У випадку використання ПЗ, що представлено в даній кваліфікаційній роботі, необхідно (маючи мережний доступ до обладнання) відкрити в програмі браузера сторінку localhost:3001 (або запустити консольний скрипт), пройти автентифікацю, ввести IP-адресу маршрутизатора та запустити потрібну команду на виконання. Результат виконання можна проаналізувати беспосередньо на моніторі свого пристороюю. Необхідний час – 2 хвилини.

Такий підхід до розподілення задач усуває необхідність обміну данами між секторами служб експлуатації мережі та може прискорити виконання експлуатаційних завдань.

5.2 Кількісна оцінка ефективності розробленого програмного забезпечення

5.2.1 Аналіз ефективності автоматизації типових завдань управління та конфігурації мережних пристроїв

Оскільки мережі стають дедалі складнішими, діяльність, необхідна для їх підтримки та управління ними, стає експоненціально складнішою та більш трудомісткою. Через постійний тиск забезпечення відповідності мереж зростаючим вимогам щодо управління, такі завдання, як оновлення програмного забезпечення, міграція та забезпечення нових елементів мережі в ефективний і відповідний вимогам спосіб, є критично важливими, а мережна автоматизація необхідна для прискорення та підтримки мережних операцій [23].

В [23] показано, яким чином можна виміряти цінність автоматизації та розрахувати, використовуючи просту модель, її ефективність у межах відсоткового скорочення часу/зусиль.

Під часом виконання завдання розуміється кількість часу, необхідного для виконання конкретного завдання.

Водночас наскрізний час (End-to-End Time) – це кількість часу, необхідного для завершення повного процесу конкретного завдання від створення завдання до закриття завдання, наприклад, збір даних, попередня перевірка, сама зміна, пост-перевірка, закриття завдання і повідомлення про зміну через системи обміну повідомленнями.

У таблиці 5.1 наведено порівняння часу на ручне та автоматизоване виконання типових задач мережного управління та конфігурації [23].

Для прикладу розглянемо завдання введення в експлуатацію нового пристрою в мережі. Підключення пристрою включає конфігурацію та активацію нового пристрою в мережі. Це вимагає послідовного застосування стандартних конфігурацій для досягнення цільових показників якості більшості мережних операторів. Це критично важливе завдання для швидкого розшиення мережі, надання послуг та активації нових користувачів, а також покращення якості обслуговування. Крім того, необхідне належне управління для застосування конфігурацій на етапі Day 0 для мережного підключення, конфігурацій на етапі Day 1 для операційних налаштувань та управління конфігурацією пристрою з часом.

Таблиця 5.1 – Порівняння часу на ручне та автоматизоване виконання типових задач мережного управління та конфігурації [23]

Завдання	Час ручного виконання (хвилин)	Час автоматизованного виконання (хвилин)
Оновлення ПЗ	45	5
Конфігурація та активація нового пристрою	240	15
Управління конфігурацією	12	2
Управління SD-WAN Branch	210	15
Оновлення DNS	15	1
Управління балансувальником навантаження	29	1
Зміни конфігурації міжмережного екрана	29	1
Конфігурація віртуальної приватної хмари	10	3
Конфігурація VLAN	15	1
Керування політикою міжмережного екрана	25	3

Отже, у разі виконання цього завдання вручну отримаємо наступне: 1000 пристроїв, що вводяться в експлуатацію 1 раз на рік, наразі потребують 4 години людських зусиль і охоплюють 20 днів (28 800 хвилин) загальної тривалості (підготовка, планування, попередня перевірка, введення в експлуатацію, пост-перевірка). Водночас застосування автоматизованого підходу скорочує цей процес до 15 хвилин людських зусиль протягом 5 днів (7 200 хвилин).

Таким чином, автоматизація конфігурації та активації нового пристрою в мережі дозволить скоротити час на розгортання нових клієнтів або додавання нових послуг, а також підвищити рівень утримання клієнтів, оскільки будуть перевершені їхні очікування щодо обслуговування.

5.2.2 Кількісна оцінка ефективності автоматизації завдання у розробленому програмному забезпеченні

Проведемо кількісну оцінку ефективності автоматизації завдання опитування порту у розробленому програмному забезпеченні.

Нехай ручне виконання команди «show port» на одному маршрутизаторі займає 3 хвилини. Якщо інженеру потрібно виконати це завдання на 20 маршрутизаторах, це займе більше часу:

$$20 \text{ роутерів} \cdot 3 \text{ хв/роутер} = 60 \text{ хв.}$$

У випадку автоматизованного виконання команди на всіх 20 маршрутизаторах, загальний час складатиметься з таких компоментів:

– 5 хвилин налаштування;

– 10 сек виконання для кожного маршрутизатора;

– 3 хв 20 с для 20 маршрутизаторів.

Відповідно до наведених вихідних даних отримаємо економія часу та ефективність у відсотках:

$$\Delta = 60 \text{ хв} - (5 + 3 \text{ хв } 20 \text{ сек}) = 51 \text{ хв } 40 \text{ с;}$$

$$\Delta_{\%} = \frac{60 \text{ хв} - (5 + 3 \text{ хв } 20 \text{ сек})}{60 \text{ хв}} \cdot 100\% = 86\%.$$

Отже, у цьому сценарії автоматизація економить більш ніж 50 хвилин (або 86% часу) лише за один раунд виконання команди «show port» на 20 маршрутизаторах.

Також потрібно враховувати людський фактор. У разі ручного виконання великої кількості команд існує значна вірогідніть помилок, тобто час роботи потенційно може збільшитись.

ВИСНОВКИ

У кваліфікаційній роботі вирішено важливу задачу щодо дослідження процесу управління ТКМ із застосуванням засобів автоматизації.

З цією метою проведений аналіз та порівняння існуючих засобів автоматизації показав, що автоматизація є необхідним інструментом для сучасних телекомунікаційних мереж. Автоматизація дозволяє знизити операційні витрати, підвищити ефективність та надійність роботи мереж, а також швидко адаптуватися до змін у мережному середовищі. Існуючі рішення охоплюють широкий спектр технологій і підходів, що дає змогу обрати оптимальні варіанти для конкретних завдань.

Аналіз питання розробки нових засобів автоматизації виявив важливість правильного вибору інструментів для управління мережами. Використання інтерфейсу командного рядка (CLI) має свої переваги та недоліки, однак на практиці ще залишається важливим методом налаштування телекомунікаційного обладнання. Приклади налаштування обладнання таких виробників, як Huawei та Ericsson, демонструють практичну цінність і ефективність автоматизованих підходів.

Огляд інструментів та мов програмування показав, що для автоматизації мереж часто використовуються мови програмування Perl та Python. Ці мови відзначаються широкими можливостями для інтеграції з різними мережними пристроями. Використання AJAX, JavaScript/DOM для клієнтської частини та різноманітних серверних технологій забезпечує ефективне створення вебзастосунків для управління мережею.

Програмна реалізація завдань управління та моніторингу продемонструвала, що розроблене програмне забезпечення дозволяє ефективно моніторити телекомунікаційну мережу. Ядро програмного забезпечення, клієнтська та серверна частини вебзастосунку забезпечують інтегровану платформу для автоматизації певних процесів управління мережею.

Оцінка ефективності автоматизації показала, що впровадження автоматизованих засобів управління значно покращує якість і продуктивність роботи мережі. Кількісна оцінка підтвердила, що автоматизація типових завдань управління та конфігурації мережних пристроїв суттєво знижує час на їх виконання та мінімізує людські помилки.

Загальний підсумок щодо отриманих результатів можна сформулювати наступним чином.

1) Пропрієтарне програмне забезпечення вимагає від операторів зв'язку придбання дорогої ліцензії на його використання.

2) Використання GUI надає можливість автоматичної перевірки синтаксису команд до їхнього відправлення на виконання, що дозволяє «відфільтровувати» помилкові команди, тим самим прискорюючи робочий процес.

3) Використання CLI забезпечує можливість написання сценаріїв для прискорення обробки однотипних завдань та їх виконання безпосередньо у командному рядку.

4) Можливість розширення використання розробленого у межах роботи програмного забезпечення надасть можливість програмування нових модулів для роботи з різним обладнанням, що дозволить уніфікувати інтерфейс і скоротити час і витрати (окрім SSH можна використовувати такі протоколи зв'язку з обладнанням, як Telnet, CORBA, SNMP, SMPP, HTTP та ін.), наприклад, на навчання персоналу NOC (Network Operations Center).

Отже, дослідження підтвердило, що автоматизація процесів управління телекомунікаційними мережами є критично важливою для забезпечення їх ефективної та надійної роботи. Вибір відповідних інструментів та технологій для автоматизації дозволяє значно підвищити продуктивність і якість обслуговування мереж. Подальші дослідження та розвиток у цій сфері повинні зосередитися на інтеграції новітніх технологій, вдосконаленні існуючих рішень та створенні більш гнучких і масштабованих систем управління.

Окремі результати роботи доповідались на трьох Міжнародних науково-технічних конференціях [16, 24, 25]. Розроблене програмне забезпечення було протестовано на лабораторній мережі компанії Nokia Bell NV, розташованій в м. Антверпен, Бельгія (реальні IP-адреси та облікові дані було змінено з міркувань безпеки).

Загалом виконані завдання кваліфікаційної роботи підтвердили важливість та ефективність автоматизації в сучасних телекомунікаційних системах і мережах та сприяли більш глибокому розумінню її ролі та можливостей у сфері мережного управління.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Pinto I., Lacunza A. A. Network Automation Trends and Strategy. Cisco Systems Inc., 2021. 17 p. URL: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/network-automation-strategy-wp.pdf (дата звернення: 01.03.2024).

2. 7 benefits of network automation technology to power your brand. URL: https://www.hamilton-barnes.com/resources/blog/7-benefits-of-network-automation-technology-to-power-your-brand/ (дата звернення: 01.03.2024).

3. Features, Advantages, and Disadvantages Command line interface (CLI). URL: https://www.knowcomputing.com/features-advantages-and-disadvantages-command-line/ (дата звернення: 01.03.2024).

4. Networking Automation Software: Pros & Cons. URL: https://www.auvik.com/franklyit/blog/best-network-automation-software/ (дата звернення: 01.03.2024).

5. Лемешко О. В., Єременко О. С., Невзорова О. С. Потокові моделі та методи маршрутизації в інфокомунікаційних мережах: відмовостійкість, безпека, масштабованість. Харків: ХНУРЕ, 2020. 308 с. DOI: https://doi.org/10.30837/978-966-659-282-1.

6. Лемешко О. В., Єременко О. С., Євдокименко М. О., Шаповалова А. С., Слейман Б. Моделювання та оптимізація процесів безпечної та відмовостійкої маршрутизації в телекомунікаційних мережах : монографія. М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. Харків : ХНУРЕ, 2022. 198 с. DOI: https://doi.org/10.30837/978-966-659-378-1.

7. Rak J., Hutchison D. (eds) Guide to Disaster-Resilient Communication Networks. Computer Communications and Networks. Springer, Cham, 2020. 813 p. DOI: https://doi.org/10.1007/978-3-030-44685-7.

8. Network automation tools comparison in code examples: Terraform, Ansible, and Python SDK. URL: https://codilime.com/blog/network-automation-tools-comparison/ (дата звернення: 01.03.2024).

9. Compare 8 network automation tools. URL: https://www.techtarget.com/searchnetworking/feature/The-8-leading-options-in-network-automation-tools (дата звернення: 01.03.2024).

10. Cloud-native automation for Telecom networks. URL: https://cloud.google.com/telecom-network-automation (дата звернення: 01.03.2024).

11. What is Network Automation? URL: https://www.geeksforgeeks.org/what-is-network-automation/ (дата звернення: 01.03.2024).

12. Perl vs. Python: Which is Better for Scripting? URL: https://devtechnosys.com/insights/tech-comparison/perl-vs-python/ (дата звернення: 01.04.2024).

13. Python vs Perl: A Detailed Comparison for Scripting in 2024. URL: https://www.halfnine.com/blog/post/python-vs-perl (дата звернення: 01.04.2024).

14. Perl vs Python: Showdown of the Best Programming Languages 2024. URL: https://www.redswitches.com/blog/perl-vs-python/ (дата звернення: 01.04.2024).

15. Perl Vs Python: What Are The Key Differences. URL: https://www.softwaretestinghelp.com/perl-vs-python/ (дата звернення: 01.04.2024).

16. Мамон Р. В., Биковець О. І. Застосування засобів автоматизації в управлінні телекомунікаційними мережами. Проблеми електромагнітної сумісності перспективних безпроводових мереж зв'язку (EMC-2022) : матеріали восьмої Міжнародної науково-технічної конференції, 24 – 25 листопада 2022 р. Харків, ХНУРЕ, 2022. С. 76 – 77. URI: https://openarchive.nure.ua/handle/document/21186.

17. haversine. PyPI. URL: https://pypi.org/project/haversine/ (дата звернення: 11.04.2024).

18. Формула гаверсинуса – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Формула_гаверсинуса (дата звернення: 11.04.2024).

19. AJAX Introduction. URL:https://www.w3schools.com/asp/asp_ajax_intro.asp (дата звернення: 01.03.2024).

20. Introducing JSON. URL: https://www.json.org/json-en.html (дата звернення: 01.05.2024).

21. Nokia Documentation Center. URL: https://documentation.nokia.com/pybin/doc_ctr.py (дата звернення: 01.03.2024).

22. Network Services Platform. Automate, manage and control IP and optical networks. URL: https://www.nokia.com/networks/ip-networks/network-services-platform (дата звернення: 01.05.2024).

23. Top 10 Network Automation Use Cases. URL: https://www.itential.com/resource/ebook/top-10-network-automation-use-cases/ (дата звернення: 01.05.2024).

24. Персіков М. А., Лемешко В. О., Мамон Р. В. Практичне використання Network DevOps для автоматизації управління в інфокомунікаціях. XV Міжнародна науково-технічна конференція студентів та аспірантів «Перспективи розвитку інформаційно-телекомунікаційних технологій та систем» ПРІТС 2023: Збірник тез конференції. К.: КПІ ім. Ігоря Сікорського, 2023. С. 358. URI: https://openarchive.nure.ua/handle/document/22764.

25. Nedostup D., Solomianyi M., Mamon R. End-to-End Network Resilience, Security, and QoS in SD-WAN. Інформатика, Математика, Автоматика IMA :: 2023: матеріали та програма Міжнародної наукової конференції молодих учених (м. Суми, 24-28 квітня 2023 р.). Суми : СумДУ. 2023. С. 39. URI: https://openarchive.nure.ua/handle/document/22988.