

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Графічний інтерфейс налаштування локальних мереж Ethernet та
Wi-Fi Сумського закладу загальної середньої освіти I-III ступенів №26»
здобувача групи ІН – 94-1 Ніколаєнка Павла Михайловича

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

Павло НІКОЛАСНКО

_____ (підпис)

Керівник,
кандидат фізико-математичних наук,
старший викладач кафедри
комп'ютерних наук

Дмитро ВЕЛИКОДНИЙ

_____ (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН – 94-1 Ніколаєнка Павла Михайловича.

1. Тема роботи: «Графічний інтерфейс налаштування локальних мереж Ethernet та Wi-Fi Сумського закладу загальної середньої освіти І-ІІІ ступенів №26»
затверджую наказом по СумДУ від _____
2. Термін здачі здобувачем кваліфікаційної роботи до 09 червня 2023 року
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Інформаційно-аналітичний огляд огляд технологій та протоколів. 2) Створення моделі локальної мережі у середовищі Cisco Packet Trace. 3) Створення графічного інтерфейсу для налаштування локальної мережі.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Завдання прийняв до виконання _____ (підпис)

Керівник _____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд мережевих технологій Ethernet та Wi-Fi</i>		
3	<i>Створення моделі локальної мережі у середовищі Cisco Packet Trace та графічного інтерфейсу для налаштування мережі</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти _____ (підпис)

Керівник _____ (підпис)

АНОТАЦІЯ

Записка: стор. 56, рис. 29, табл. 0, додаток 1, джерел 11.

Обґрунтування актуальності теми роботи – Тема кваліфікаційної роботи є актуальною, оскільки присвячена практичному застосуванню для налагодження мережевого обладнання.

Об'єкт дослідження – огляд мережевих технологій Ethernet та Wi-Fi

Мета роботи – моделювання локальної мережі, створити графічний інтерфейс для автоматизованого налаштування мережевого обладнання.

Методи дослідження – моделювання мережі в cisco packet tracer, розробка інтерфейсу для налаштування комутаторів.

Результати – здійснено моделювання роботи апаратно-програмного забезпечення локальної мережі школи №26 за допомогою програмного забезпечення Cisco Packet Tracer. Розроблено графічний інтерфейс конфігурації мережі для спрощення налаштування.

КОМП'ЮТЕРНА МЕРЕЖА,
VLAN, CISCO, PACKET TRACER,
ГРАФІЧНИЙ ІНТЕРФЕЙС

ЗМІСТ

ВСТУП	4
1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ ОГЛЯД	5
1.1 Огляд мережевих технологій Ethernet та Wi-Fi.....	5
1.2 Опис протоколів мережевого рівня та транспортного рівня	8
1.3 Огляд існуючих програм для налаштування мереж	12
1.4 Постановка задачі	14
2 МОДЕЛЮВАННЯ МЕРЕЖІ В CISCO PACKET TRACER	16
2.1 Моделювання топології	16
2.2 Налаштування мережевого обладнання.....	20
2.3 Створення VLAN та налаштування безпеки мережі	26
3 ГРАФІЧНИЙ ІНТЕРФЕЙС НАЛАШТУВАННЯ МЕРЕЖ	31
3.1 Розробка графічного інтерфейсу для налаштування мережі	31
3.2 Функціональні вимоги до інтерфейсу	39
3.3 Розробка інтерфейсу	40
3.4 Тестування розробленого інтерфейсу	43
ВИСНОВКИ	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	46
ДОДАТОК	47

ВСТУП

Використання мережі Інтернет є надзвичайно важливим у сучасному світі. Інтернет забезпечує швидкий та зручний доступ до великої кількості інформації з усього світу, що дає можливість знайти відповіді на будь-які запитання та розв'язати складні завдання.

Крім того, Інтернет є невід'ємною частиною бізнесу та економіки, де він забезпечує ефективний спосіб комунікації між різними організаціями, а також дозволяє проводити трансакції та здійснювати операції з будь-якої точки світу.

Мережа Інтернет також є незамінним джерелом розваг та спілкування. Люди можуть спілкуватись з друзями та родиною, відвідувати соціальні мережі, дивитись фільми, слухати музику та грати в ігри.

Крім того, мережа Інтернет є важливим інструментом для освіти та навчання. Завдяки Інтернету студенти та учні можуть знайти різноманітну інформацію про свої предмети, виконувати домашні завдання, здавати тести та іспити онлайн. Використання мережі Інтернет має безліч переваг та відкриває безмежні можливості для розвитку та прогресу у різних сферах життя людей.

За останні роки період епідемії, військовий стан, вимоги сучасності потребують використання великої кількості комп'ютерної та мобільної техніки. Цей розвиток викликає розширення мережевих технологій у підприємствах, закладах освіти, тощо. Впровадження або модернізація технологій значною мірою впливатиме на робочий процес, розвиток суспільства та комунікацію. Ці процеси потребують відповідального підходу до роботи з комп'ютерними мережами. Рішення які приймаються стосовно них повинні відповідати сучасним вимогам та спільно зважати на вже існуючі варіанти.

1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ ОГЛЯД

1.1 Огляд мережевих технологій Ethernet та Wi-Fi

У сучасному світі, де швидкість та надійність мережевого з'єднання є ключовими факторами для ефективної роботи, налаштування мереж Ethernet та Wi-Fi стає все більш важливим завданням. Задля досягнення максимальної продуктивності, користувачі мереж повинні мати доступ до інтуїтивно зрозумілого та легко використовуваного інтерфейсу налаштування.

Для забезпечення роботи мережі, необхідно правильно налаштувати її компоненти, зокрема, налаштувати параметри мережевих пристроїв, таких як маршрутизатори, комутатори та точки доступу Wi-Fi [1, 2].

Однак, процес налаштування мереж може бути складним та вимагати від операторів мережі великої кількості часу та знань. У зв'язку з цим, для спрощення та автоматизації процесу налаштування мереж було розроблено різноманітні програмні засоби, зокрема графічні інтерфейси.

Кваліфікаційна робота бакалавра присвячена аналізу та розробці графічного інтерфейсу налаштування мереж Ethernet та Wi-Fi. У процесі дослідження, були проаналізовані сучасні інструменти налаштування мереж та з'ясовано їх переваги та недоліки. На основі цього аналізу було розроблено новий графічний інтерфейс, який відповідає вимогам сучасних користувачів.

У кваліфікаційній роботі розглянуто технології мереж та виконано розробку графічного інтерфейсу налаштування мереж Ethernet та Wi-Fi на прикладі локальної мережі школи. У роботі описано основні функції графічного інтерфейсу налаштування, його архітектуру та інтерфейсні елементи. Також надано інструкції з використання інтерфейсу та налаштування мережі за допомогою цього інструменту.

Метою роботи є поліпшення процесу налаштування мереж та забезпечення їх більш ефективної роботи. З метою досягнення даної мети було визначено наступні завдання:

1. Провести аналіз існуючих методів налаштування мереж Ethernet та Wi-Fi.
2. Розробити графічний інтерфейс налаштування мереж Ethernet та Wi-Fi.
3. Моделювати роботу графічного інтерфейсу на програмному забезпеченні Cisco Packet Tracer.
4. Проаналізувати результати моделювання та зробити висновки щодо ефективності запропонованого рішення.

Дана робота має практичну цінність, оскільки дозволить поліпшити процес налаштування мереж Ethernet та Wi-Fi в шкільній мережі, зменшити кількість помилок та збільшити ефективність роботи мережі в цілому. Результати дослідження та розробки можуть бути корисними для шкільних адміністраторів, які відповідають за налаштування мереж та забезпечення їх безперебійної роботи.

Розробка та впровадження мереж є актуальною проблемою для будь-якої компанії або установи, оскільки мережеві технології стали необхідністю в сучасному світі. У зв'язку з цим, існує безліч різних технологій, які дозволяють побудувати мережу, забезпечуючи надійний та ефективний обмін даними.

Однією з найпоширеніших технологій є Ethernet, що є стандартом для провідних локальних мереж [2]. Ця технологія базується на передачі даних за допомогою кабелю, що забезпечує високу швидкість передачі даних та малу кількість помилок. Стандарт Ethernet змінювався з часом, щоб забезпечити підтримку високошвидкісних мереж, таких як 10GbE [3].

Іншою поширеною мережевою технологією є Wi-Fi, що забезпечує бездротовий доступ до мережі. Вона базується на стандартах IEEE 802.11 і забезпечує швидку передачу даних у мережі без необхідності використовувати кабель [3]. Wi-Fi дозволяє підключатися до мережі з будь-якої точки, що робить її зручною для використання у мобільних пристроях, таких як смартфони та планшети.

Для побудови мережі, що забезпечує як провідний, так і бездротовий доступ, зазвичай використовують комбіновані рішення, які використовують Ethernet для провідного з'єднання та Wi-Fi для бездротового доступу [1,2,3]. Це забезпечує максимальну гнучкість та зручність при використанні мережі.

Технологія Ethernet була розроблена в 1970-х роках і поступово стала стандартом для провідних мереж [3]. Ця технологія передачі даних використовує кабелі для передачі сигналів між пристроями в мережі. Розроблено кілька типів кабелів, що використовуються в Ethernet, таких як коаксіальний кабель, вита пара, оптичний кабель тощо. Поширеним стандартом Ethernet став 10Base-T, який використовує виту пару кабелю із чотирма провідниками. Цей стандарт дозволяє передавати дані зі швидкістю до 10 Мбіт/с.

У 1990-х роках були розроблені стандарти для бездротового зв'язку, які отримали назву Wi-Fi. Wi-Fi використовує радіохвилі для передачі даних між пристроями в мережі. За допомогою Wi-Fi можна підключати пристрої до мережі без використання кабелів. До Wi-Fi використовуються різні стандарти, такі як 802.11b, 802.11g, 802.11n, 802.11ac тощо, які дозволяють передавати дані з різною швидкістю та на різну відстань [4,5].

Однак, мережі Ethernet та Wi-Fi мають свої особливості та обмеження, які варто враховувати при їх використанні. Наприклад, мережі Ethernet мають обмеження на довжину кабелю та кількість пристроїв у мережі, тоді як мережі Wi-Fi залежать від перешкод в середовищі та можуть мати обмеження на швидкість передачі даних. Однак, з правильною конфігурацією та налагодженням мережі, ці технології можуть забезпечити швидку та стабільну передачу даних в мережі.

Розглянемо більш детально мережеву технологію Ethernet. Це технологія передачі даних, яка використовується для створення локальних мереж. Ethernet забезпечує передачу даних від одного пристрою до іншого за допомогою фізичного з'єднання. У мережі Ethernet кожен пристрій має свій

унікальний MAC-адрес, який використовується для ідентифікації пристрою в мережі [5].

Технологія Wi-Fi є бездротовою мережевою технологією, що дозволяє підключати пристрої до мережі за допомогою радіохвиль. Wi-Fi дозволяє підключати пристрої до мережі без потреби в фізичному з'єднанні, що забезпечує більшу мобільність і гнучкість. Кожна точка доступу Wi-Fi має свій унікальний ідентифікатор SSID, який дозволяє підключатися до неї [6].

Обидві технології мають свої переваги та недоліки, і вибір технології залежить від конкретних вимог та потреб користувачів. У деяких випадках доцільно використовувати Ethernet, а в інших випадках Wi-Fi. Для прикладу, Ethernet зазвичай має вищу швидкість передачі даних, але його використання може бути обмеженим з точки зору фізичної мережі, тоді як Wi-Fi може бути використаний для підключення пристроїв, які знаходяться на великій відстані один від одного.

1.2 Опис протоколів мережевого рівня та транспортного рівня

Мережевий рівень та транспортний рівень є ключовими складовими будь-якої мережі.

На мережевому рівні існує безліч протоколів, які забезпечують пересилання даних між різними мережевими вузлами. Один з найпоширеніших протоколів мережевого рівня - це протокол IP (Internet Protocol), який використовується для адресації та маршрутизації даних у мережі Інтернет [2]. Протокол IP дозволяє встановлювати зв'язок між двома вузлами мережі та передавати дані між ними.

На транспортному рівні найбільш поширеними є протоколи TCP (Transmission Control Protocol) та UDP (User Datagram Protocol) [2]. Протокол TCP забезпечує надійну передачу даних між двома вузлами мережі, контролюючи пересилання даних та перевіряючи їх доставку. Протокол UDP

використовується, коли надійність передачі даних не є критичною, і дозволяє передавати дані без попередньої встановленої з'єднання.

Розуміння протоколів мережевого та транспортного рівнів є важливим для розробки та налагодження мережевих додатків та систем, що передбачають обмін даними між вузлами мережі.

Протокол IP (Internet Protocol) - це протокол мережевого рівня, який використовується для передачі даних між різними мережами. Він забезпечує маршрутизацію даних з однієї мережі в іншу, використовуючи адресацію IP-адресами.

IP-адреса складається з 32 біт (IPv4) або 128 біт (IPv6) ідентифікаторів, які унікальні для кожного пристрою в мережі. IP-адреса складається з двох частин: ідентифікатора мережі та ідентифікатора хоста [2].

Крім того, протокол IP може бути використаний разом з іншими протоколами, такими як TCP (Transmission Control Protocol) і UDP (User Datagram Protocol), які працюють на рівні транспортного протоколу. TCP забезпечує надійну передачу даних, а UDP - швидку передачу без підтвердження доставки.

Протокол IP забезпечує доставку даних до призначеного місця, але не гарантує надійність передачі, тому деякі дані можуть бути втрачені або повторно надіслані. Це може бути особливо важливо для додатків, які вимагають надійної доставки даних, таких як відео - та голосові додатки.

Крім того, протокол IP може бути використаний в різних мережевих технологіях, таких як Ethernet, Wi-Fi, DSL та інші. IP-адреса дозволяє різним пристроям взаємодіяти між собою в мережі незалежно від технології передачі даних.

Усі ці функції протоколу IP роблять його важливим елементом інфраструктури Інтернету та мережі загалом.

Протокол TCP (Transmission Control Protocol) - це протокол транспортного рівня, який забезпечує передачу даних між програмами на різних вузлах мережі з надійністю та контролем потоку [2].

TCP забезпечує надійну передачу даних, використовуючи підтвердження та перевірку цілісності даних. Якщо один з пакетів не прийшов, він буде повторно відправлений. Крім того, TCP забезпечує контроль потоку, який регулює швидкість передачі даних, щоб уникнути перевантаження мережі та втрати даних.

Протокол TCP використовується в багатьох додатках, які вимагають надійної передачі даних, таких як веб-браузери, електронна пошта та файлообмінні програми. Якщо прикладна програма використовує TCP для передачі даних, то TCP розбиває дані на пакети та відправляє їх через мережу. На іншому кінці мережі TCP збирає пакети та забезпечує правильну послідовність передачі даних.

Протокол TCP використовується разом з протоколом IP для передачі даних через мережу Інтернет. TCP/IP стек є стандартом де-факто для мережеских з'єднань та використовується в більшості комп'ютерних мереж.

Протокол ICMP (Internet Control Message Protocol) є одним з основних протоколів мережевого рівня в мережах з протоколом IP [1]. Він використовується для передачі повідомлень про помилки, стан мережі та для взаємодії з іншими мережевими пристроями.

ICMP повідомляє про помилки, що виникають при обробці пакетів IP. Прикладом таких помилок можуть бути: недоступність мережі або хосту, переповнення буфера, неправильний формат пакета і т.д. Такі повідомлення є важливими для виявлення та виправлення проблем в мережі.

Протокол ICMP також використовується для взаємодії з іншими мережевими пристроями [1]. Наприклад, з його допомогою можна відправляти запити на пінгування (ping) до інших пристроїв для перевірки доступності та швидкості відповіді. Також ICMP може використовуватись для відправки повідомлень про перенаправлення пакетів від маршрутизатора до маршрутизатора з метою оптимізації шляху та зменшення трафіку.

Протокол ICMP не є протоколом, що забезпечує доставку повідомлень, тому він не має можливості перевіряти доставку повідомлення або

запитувати про його статус. Також він не є частиною стандартного TCP/IP стеку протоколів, а є окремим протоколом.

Усі повідомлення ICMP складаються з заголовку та даних. Заголовок ICMP містить тип повідомлення, код помилки, контрольну суму та інші поля, що необхідні для обробки повідомлення. Дані ICMP можуть містити додаткову інформацію, наприклад, IP-адресу, яка є об'єктом запиту чи повідомлення про помилку [1].

Загальні типи повідомлень ICMP включають повідомлення про помилки можуть містити таку інформацію, як:

- Код помилки: це число вказує на конкретний тип помилки, що сталася. Наприклад, код помилки 1 вказує на те, що пакет не може бути доставлений, оскільки мережа недоступна.

- Послідовний номер пакету: це номер пакету, що викликав помилку.

- Інші поля: деякі повідомлення про помилки можуть містити додаткову інформацію, таку як IP-адреса пристрою, який відправив повідомлення.

Повідомлення про помилки ICMP є важливим інструментом для діагностики проблем мережі та розробки програмного забезпечення, яке працює з мережевими пристроями.

Протокол DHCP (Dynamic Host Configuration Protocol) - це мережевий протокол, який дозволяє автоматично налаштовувати параметри IP-адреси, маски підмережі, шлюзу за замовчуванням, DNS-серверів та інших параметрів мережевого з'єднання [1]. Це дозволяє комп'ютерам і іншим пристроям отримувати необхідні мережеві налаштування з централізованого DHCP-сервера замість ручної настройки на кожному пристрої окремо.

DHCP-клієнти запитують DHCP-сервери, щоб отримати необхідні налаштування. Якщо DHCP-сервер відповідає позитивно, він надає клієнту потрібні параметри мережі. Якщо в мережі наявний більше одного DHCP-сервера, то відповідь від одного з них вважається вибраною, а відповіді від інших серверів відхиляються.

Протокол DHCP використовує UDP-порти 67 та 68 для комунікації між DHCP-серверами та DHCP-клієнтами відповідно. Клієнти можуть отримувати налаштування на будь-якому протоколі мережевого рівня, включаючи Ethernet, Wi-Fi, Token Ring тощо.

DHCP також може використовуватися для надання статичних IP-адрес та інших налаштувань для пристроїв, які повинні мати сталу адресу, наприклад, серверів або принтерів.

Протокол DHCP забезпечує багато переваг, зокрема зменшення трудомісткості та забезпечення послуг віддаленого адміністрування.

1.3 Огляд існуючих програм для налаштування мереж

На сьогоднішній день існує велика кількість програм для налаштування мереж, які дозволяють користувачам швидко та ефективно налаштовувати різноманітні параметри мережі. Основними типами програм для налаштування мереж є консольні та графічні інтерфейси.

Консольні інтерфейси передбачають введення команд в консоль або термінал, які дозволяють користувачам налаштовувати мережеві параметри. Такі програми зазвичай мають велику кількість параметрів та вимагають певного рівня знань для їх коректного використання.

Графічні інтерфейси є більш інтуїтивно зрозумілими для користувачів та надають більш простий та зручний спосіб налаштування мережевих параметрів [2]. Такі програми зазвичай мають графічний інтерфейс, який дозволяє користувачам налаштовувати параметри мережі за допомогою миші та клавіатури.

До популярних програм для налаштування мереж належать:

1. Cisco Network Assistant [7,9]: це безкоштовна програма для налаштування мережевих пристроїв Cisco, що надає графічний інтерфейс користувача і дозволяє керувати різними настройками мережі.

2. SolarWinds Network Configuration Manager [11]: ця програма дозволяє автоматизувати процес налаштування мережевих пристроїв, зберігати та контролювати налаштування, виконувати аудит налаштувань мережі, стежити за змінами налаштувань та багато іншого.

3. OpenNMS [10]: це безкоштовна відкрита програма для моніторингу та управління мережами, що дозволяє відслідковувати стан мережевих пристроїв, виконувати аудит налаштувань, стежити за змінами та діагностувати проблеми мережі.

4. ManageEngine OpManager [10]: ця програма надає централізоване управління мережею, що дозволяє відслідковувати стан мережевих пристроїв, аналізувати трафік, виконувати аудит налаштувань та багато іншого.

5. Paessler PRTG Network Monitor [10]: ця програма дозволяє моніторити стан мережевих пристроїв, виконувати аудит налаштувань та відстежувати зміни в мережі, забезпечуючи повний контроль над мережевою інфраструктурою.

Ці програми забезпечують широкий функціонал для налаштування та моніторингу мереж, та є ефективними інструментами для роботи зі складними мережами.

Існує багато програм для налаштування мереж, які мають різний рівень складності та функціоналу. Одні з найпоширеніших програм цього типу - це програми від виробників обладнання мережі, такі як Cisco Network Assistant та Cisco Prime Infrastructure [7,8,9]. Вони призначені для налаштування та керування мережею на різних рівнях, від окремих пристроїв до всієї мережі в цілому.

Крім того, існують такі програми, як PuTTY та SecureCRT, які забезпечують доступ до командного рядка пристроїв мережі через протоколи SSH або Telnet [1,2]. Ці програми використовуються для виконання консольних команд на окремих пристроях мережі. Основне призначення цих програм - забезпечити користувачам можливість виконувати команди на

віддаленому сервері або мережевому пристрої. Вони реалізують наступні основні функції:

- з'єднання з віддаленими системами, що дає можливість отримати доступ до командного рядка цих систем для виконання різних завдань і налаштувань;
- шифрування трафіку;
- керування сесіями: PuTTY та SecureCRT зберігають історію з'єднань, дозволяючи користувачам зручно керувати різними сесіями і швидко переключатися між ними;
- підтримують можливість зберігати конфігурацію підключень для легкого повторного використання;
- виконувати налаштування терміналу;
- функції передачі файлів.

Також наявні програми, які забезпечують більш високорівневий інтерфейс для налаштування мережі, такі як Network Configuration Manager та SolarWinds Network Configuration Manager. Вони дозволяють налаштовувати та керувати мережею з одного місця, зокрема встановлювати параметри мережевих пристроїв, налаштовувати безпеку та відстежувати події в мережі [2].

Окрім вищезгаданих програм, існує багато інших програм, які забезпечують налаштування та керування мережами, в залежності від їх функціональності та призначення. Важливо вибрати програму, яка найбільше відповідає потребам користувача та вимогам його мережі.

1.4 Постановка задачі

Постановка задачі є важливим етапом дослідження та написання кваліфікаційної роботи. Вона дозволяє чітко сформулювати цілі та об'єктиви роботи, а також визначити проблеми, які потрібно вирішити в рамках даної роботи на тему "Графічний інтерфейс налаштування мереж Ethernet та Wi-Fi

на прикладі локальної мережі школи №26". Основні моменти котрих торкається постановка задачі:

1. Детальний аналіз інформаційних та аналітичних матеріалів, пов'язаних з мережевими технологіями Ethernet та Wi-Fi, а також з використанням графічних інтерфейсів для налаштування мереж.

2. Вивчення основних протоколів мережевого рівня та транспортного рівня, зокрема протоколів IP, TCP та UDP, їх функцій та взаємодії.

3. Розробка графічного інтерфейсу мережі Ethernet та Wi-Fi, який буде забезпечувати зручний та інтуїтивно зрозумілий спосіб роботи з мережею.

4. Використання програмного забезпечення Cisco Packet Tracer для моделювання та тестування розробленого графічного інтерфейсу налаштування мереж.

5. Проведення експериментів та аналіз результатів для перевірки працездатності та ефективності розробленого графічного інтерфейсу.

6. Порівняння розробленого графічного інтерфейсу з існуючими програмами для налаштування мереж та оцінка його переваг та недоліків.

В рамках роботи ми маємо на меті створити зручний та ефективний графічний інтерфейс для налаштування мереж Ethernet та Wi-Fi в локальній мережі школи.

2 МОДЕЛЮВАННЯ МЕРЕЖІ В CISCO PACKET TRACER

2.1 Моделювання топології

Для початку роботи з моделювання топології, було розглянуто реальну мережу школи №26.

Даний заклад має досить маленьку мережу, як для закладу освіти, та має у своєму складі один маршрутизатор та три комутатори, два з яких виконують комутацію у класах з інформатики, а третій – комутацію інших класів у яких передбачено підключення до мережі. Wi-Fi роутери розміщено у класах самостійно керівниками кабінетів. У кожному кабінеті розміщено пристрої від різних виробників та різних потужностей. Для полегшення моделювання та роботи з обладнанням кількість роутерів у моделі зменшено.

На основі проведеного аналізу потреб та враховуючи наявне обладнання, було обрано фізичну топологію "дерево" для мережі школи. У даній топології використовуються маршрутизатор Cisco 1941, три комутатори Cisco 2960, сервер, 16 комп'ютерів та один роутер.

Ця фізична топологія використовує ієрархічну структуру, де маршрутизатор Cisco 1941 виступає як головний маршрутизатор, що забезпечує зв'язок з зовнішніми мережами та координує роботу всієї мережі в школі. Три комутатори Cisco 2960 розташовуються на різних поверхах або в різних приміщеннях школи і забезпечують підключення до комп'ютерів і сервера.

Роутер використовується для забезпечення з'єднання пристроїв по протоколу Wi-Fi.

Крім того, сервер, який виконує функцію DNS-сервера, встановлений у мережі школи. DNS-сервер відповідає за перетворення доменних імен на IP-адреси, що дозволяє комп'ютерам та іншим пристроям в мережі знайти

необхідні ресурси за їх назвами. Він виконує важливу роль у забезпеченні правильної адресації та маршрутизації мережевих запитів.

Ця фізична топологія "дерево" дозволяє ефективно організувати мережеву інфраструктуру школи, забезпечуючи стабільний зв'язок між всіма пристроями, легке масштабування та керування мережею.

Маршрутизатор Cisco 1941 відіграє важливу роль у мережі [7], особливо якщо використовується статична маршрутизація. Основні аспекти його ролі в такій мережі включають:

1. Визначення маршрутів: Маршрутизатор 1941 дозволяє адміністратору мережі вручну визначати маршрути до різних мереж та підмереж. Це означає, що кожен маршрутизатор має явно налаштовані маршрути, які вказують, які мережі можна досягти через які інтерфейси.
2. Керування трафіком: Завдяки статичній маршрутизації, адміністратор має повний контроль над тим, як розподіляється мережевий трафік. Він може налаштувати маршрутизатор таким чином, щоб трафік передавався через певні шляхи в мережі, що дозволяє оптимізувати шляхи передачі даних та уникнути зайвого навантаження на деякі мережеві сегменти.
3. Надійність мережі: Використання статичної маршрутизації дозволяє забезпечити високу надійність мережі. Оскільки маршрути вручну встановлюються і не змінюються автоматично, немає ризику некоректного оновлення маршрутних таблиць та несправностей, пов'язаних з динамічною маршрутизацією.

Комутатори Cisco 2960 виконують кілька важливих функцій у мережі, що допомагають управляти, керувати та забезпечувати передачу даних між різними пристроями [7]. Основні функції, які вони виконують у даній мережі, включають:

1. Пересилання пакетів. Комутатори 2960 забезпечують розподіл трафіку між підключеними пристроями. Вони приймають пакети з одного порту і пересилають їх на інші порти, визначаючи оптимальний шлях для кожного

пакета. Це дозволяє ефективно передавати дані в мережі і зменшує навантаження на мережевий трафік.

2. Управління VLAN. Комутатори 2960 підтримують віртуальні локальні мережі (VLAN), що дозволяють логічно розділяти мережу на окремі сегменти. Вони дозволяють налаштовувати VLAN, привласнювати порти до певних VLAN і контролювати комунікацію між ними. Це допомагає встановити гнучкі мережеві політики та забезпечити безпеку та ефективність мережі [7].

3. Керування пропускною здатністю. Комутатори 2960 мають високу пропускну здатність, що дозволяє передавати дані з великою швидкістю між підключеними пристроями. Вони підтримують швидкісні Ethernet-порти, що дозволяють реалізувати швидку передачу даних в мережі.

У даній мережі (рис. 2.1) для підключення обладнання до комутаторів Cisco 2960, використовуються мережеві кабелі Ethernet. Обладнання, таке як комп'ютери, принтери або сервери, підключаються до відповідних портів на комутаторах за допомогою Ethernet-кабелів.

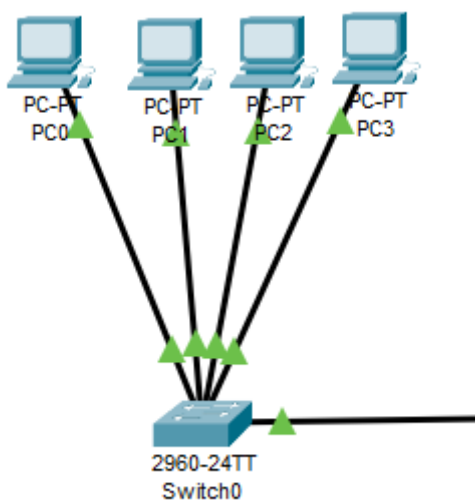


Рисунок 2.1 – Приклад комутації комутатора Cisco 2960

Для підключення обладнання використовуються роз'єми Ethernet, які зазвичай мають форм-фактор RJ-45. Кабель Ethernet підключається до порту на комутаторі та до мережевого порту обладнання, таким чином створюється з'єднання для передачі даних у мережі.

Підключення обладнання до комутаторів дозволяє забезпечити комунікацію та обмін даними між різними пристроями у мережі. Кількість підключеного обладнання може бути регульована в залежності від наявних портів на комутаторі та потреб мережі.

Проте, важливо врахувати, що для належної конфігурації та налаштування мережі рекомендується звернутися до фахівців з мережевого адміністрування, які зможуть забезпечити оптимальну роботу мережі.

У результаті розробки моделі локальної мережі розроблена топологія, яка включає маршрутизатор 1941, три комутатори 2960, сервер, 16 комп'ютерів та один роутер. Вибрана топологія "дерево" для побудови мережі, що надає структуроване та ефективне з'єднання між пристроями. Маршрутизатор Cisco 1941 виконує ключову роль у мережі. Він забезпечує маршрутизацію даних між різними мережевими сегментами, що дозволяє забезпечити комунікацію між пристроями у мережі та доступ до зовнішніх мереж. Комутатори Cisco 2960 відповідають за забезпечення комутації даних в мережі [7]. Вони забезпечують підключення пристроїв до мережі, передачу даних між пристроями та контроль рівня швидкості передачі даних. Сервер, встановлений у якості DNS-сервера, виконує функцію розподілу доменних імен та перетворення доменних імен в IP-адреси. Це дозволяє пристроям в мережі знаходити й звертатися до інших пристроїв за допомогою зрозумілих для них доменних імен.

Таким чином, подано повну картину (рис. 2.2) розробки топології мережі, включаючи ролі різних пристроїв і їх функціональні можливості. Це сприяє налагодженню ефективної та надійної мережевої інфраструктури, яка задовольняє потреби шкільної локальної мережі.

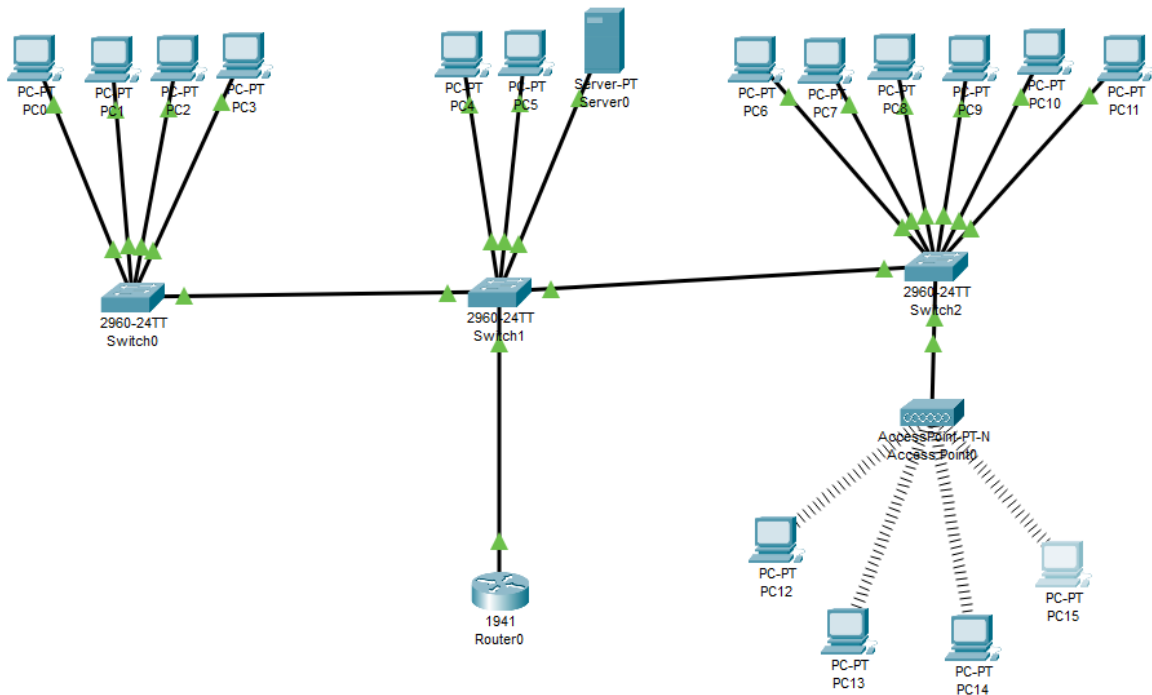


Рисунок 2.2 – Комутація моделі локальної мережі

2.2 Налаштування мережевого обладнання

Налаштування обладнання є невід’ємним кроком у функціонуванні та існуванні мережі.

Почати варто з комплектування комп’ютерів. У комп’ютерах для яких передбачено підключення до мережі по Wi-Fi необхідно замінити LAN роз’єм на Wi-Fi антену (рис. 2.3). Для цього відкриваємо комп’ютер, вимикаємо його, виймаємо старий модуль та змінюємо його на Wi-Fi антену. Після цього вмикаємо комп’ютер. Дану операцію повторюємо для всіх комп’ютерів для яких передбачено підключення по Wi-Fi.

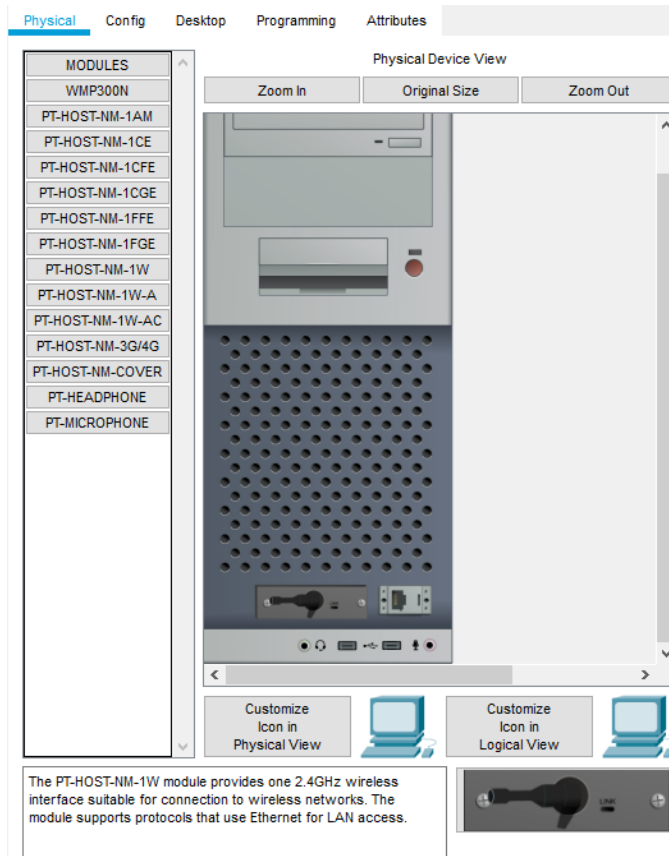


Рисунок 2.3 – Заміна модуля на комп'ютері

Наступним кроком потрібно встановити IP-адреси. Кожному комп'ютеру у першому класі (рис. 2.4) присвоїти значення по формулі $N0.0.0.n$, де N – номер класу, а n – номер пристрою (наприклад, 10.0.0.2 – другий комп'ютер у першому класі). Сервер, так, як він третій пристрій у другому класі буде мати адресу 20.0.0.3). Маску підмережі встановлено 255.255.255.0, Gateway – $N0.0.0.254$, DNS – 20.0.0.3 (рис. 2.5-2.7).

IP Configuration

Interface: FastEthernet0

IP Configuration

DHCP Static

IP Address: 10.0.0.1

Subnet Mask: 255.255.255.0

Default Gateway: 10.0.0.254

DNS Server: 20.0.0.3

IPv6 Configuration

DHCP Auto Config Static

IPv6 Address: /

Link Local Address: FE80::201:43FF:FE3C:CB84

IPv6 Gateway: /

IPv6 DNS Server: /

Рисунок 2.4 – Приклад налаштування комп’ютера у першому класі

IP Configuration

Interface: FastEthernet0

IP Configuration

DHCP Static

IP Address: 20.0.0.1

Subnet Mask: 255.255.255.0

Default Gateway: 20.0.0.254

DNS Server: 20.0.0.3

IPv6 Configuration

DHCP Auto Config Static

IPv6 Address: /

Link Local Address: FE80::240:BFF:FE06:BB10

IPv6 Gateway: /

IPv6 DNS Server: /

Рисунок 2.5 – Приклад налаштування комп’ютера у другому класі

The image shows a screenshot of a network configuration window titled "IP Configuration". The window is divided into three main sections:

- IP Configuration:** This section has two radio buttons: "DHCP" (unselected) and "Static" (selected). Below these are four text input fields: "IP Address" (20.0.0.3), "Subnet Mask" (255.255.255.0), "Default Gateway" (20.0.0.254), and "DNS Server" (20.0.0.3).
- IPv6 Configuration:** This section has three radio buttons: "DHCP" (unselected), "Auto Config" (unselected), and "Static" (selected). Below these are four text input fields: "IPv6 Address" (empty), "Link Local Address" (FE80::2D0:FFFF:FEB1:DE30), "IPv6 Gateway" (empty), and "IPv6 DNS Server" (empty).
- 802.1X:** This section has a checkbox "Use 802.1X Security" (unchecked). Below it is a dropdown menu for "Authentication" set to "MD5", and two text input fields for "Username" and "Password" (both empty).

Рисунок 2.6 – Налаштування IP Configuration на сервері

У третьому класі IP-адреси встановлюємо по формулі 30.0.0.10n, де n – номер комп'ютера.

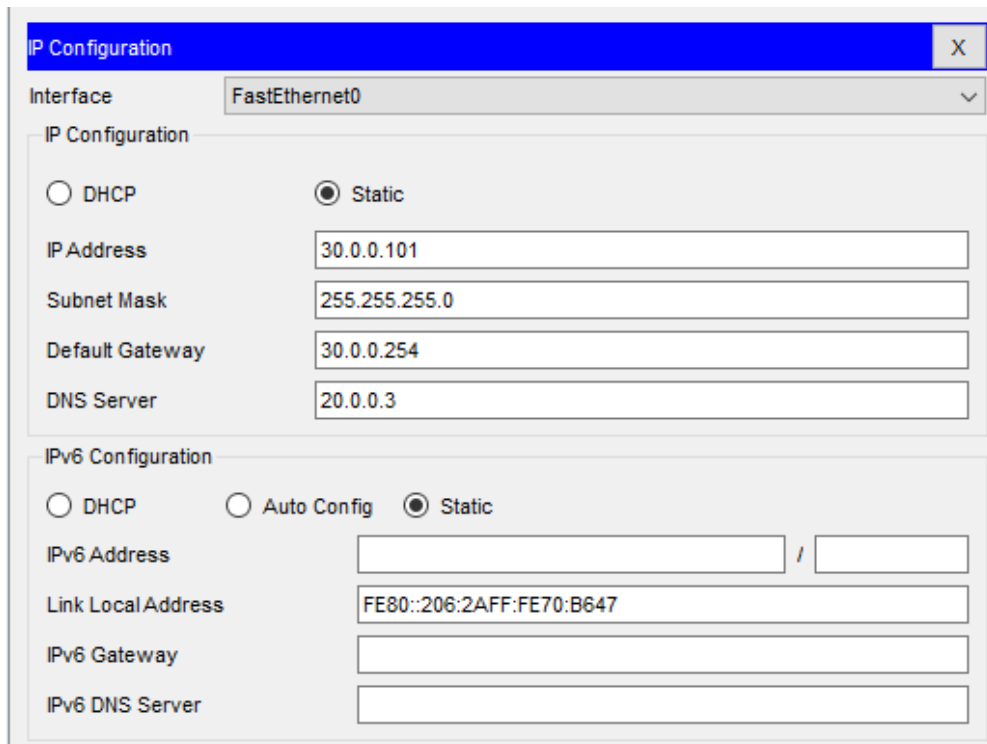


Рисунок 2.7 – Приклад налаштування комп’ютера у третьому класі

Налаштування комп’ютерів для яких передбачено підключення до мережі по Wi-Fi потрібно задати ім’я мережі “School” та WPA2-Personal – school2023 (рис. 2.8).

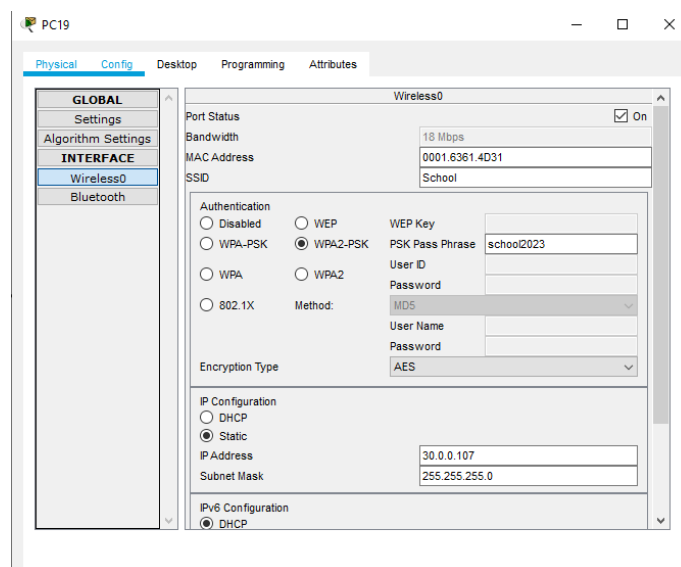


Рисунок 2.8 – Приклад налаштування комп’ютера з Wi-Fi підключенням

Наступним кроком потрібно налаштувати конфігурацію на сервері (рис. 2.9).

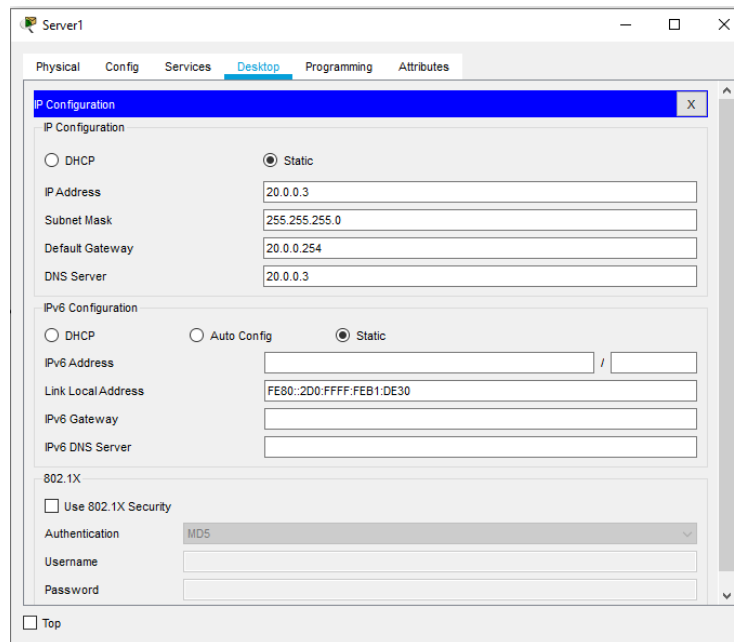


Рисунок 2.9 – Налаштування сервер

Також сервер повинен виконувати функції DNS-сервера (рис. 2.10). Для цього потрібно налаштувати відповідні сервіси. У меню "Services" обираємо DNS та вмикаємо його. Задаємо IP-адресу та назву нашого сайту.

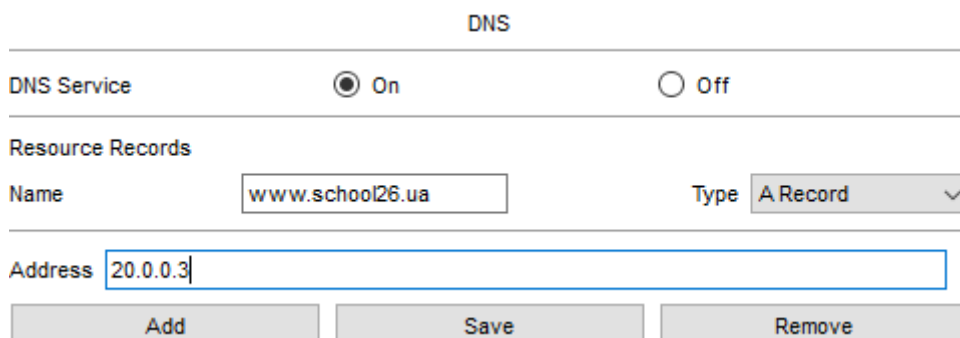


Рисунок 2.10 – Налаштування сервісу DNS

Далі у списку повинен з'явитися сайт з доменним ім'ям та його IP-адресою (рис. 2.11).

No.	Name	Type	Detail
0	www.school26.ua	A Record	20.0.0.3

Рисунок 2.11 – Додане доменне ім'я

2.3 Створення VLAN та налаштування безпеки мережі

На етапі створення VLAN ми маємо змогу на одному фізичному інтерфейсі розділити класи на віртуальні підмережі.

Для початку на кожному комутаторі необхідно у інтерфейсі командного рядка задати команди:

```
Switch>en
Switch#conf t
Switch(config)#vlan 10
Switch(config-vlan)#name School1
Switch(config-vlan)#end
```

Розглянемо детально наведені команди:

En – відкриває розширений доступ до конфігурації.

Conf t – відкриває термінал налаштувань.

Vlan 10 – створює віртуальну мережу з індексом 10.

Name School1 – задає ім'я VLAN.

End – завершує налаштування.

Виконуємо налаштування на комутаторі у другому класі та прописуємо наступні команди:

```
Switch>en
Switch#conf t
Switch(config)#vlan 10
Switch(config-vlan)#name School1
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name School2
Switch(config-vlan)#exit
Switch(config)#vlan 30
Switch(config-vlan)#name School3
Switch(config-vlan)#exit
Switch(config)#end
```

Виконуємо налаштування на комутаторі у третьому класі та прописуємо наступні команди:

```
Switch>en
Switch#conf t
Switch(config)#vlan 30
Switch(config-vlan)#name School3
Switch(config-vlan)#end
```

Встановлюємо на першому комутаторі VLAN 10 на всі порти до яких є підключення.

На другому комутаторі потрібно встановити порт, до якого підключено комутатор із першого класу VLAN – 10, з третього класу VLAN – 30, а 2 комп'ютера та сервер другого класу VLAN – 20. Інтерфейс, який з'єднує маршрутизатор та комутатор встановлюємо у режим Trunk.

На третьому комутаторі потрібно встановити на всі порти VLAN 30.

Після цього виконуємо налаштування маршрутизатора для роботи з VLAN. Переходимо у вкладку CLI та прописуємо там наступні команди:

```
Router>en
Router#conf t
Router(config)#int gig 0/0.10
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 10.0.0.254 255.255.255.0
Router(config-subif)#exit
Router(config)#int gig 0/0.20
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 20.0.0.254 255.255.255.0
Router(config-subif)#exit
Router(config)#int gig 0/0.30
Router(config-subif)#encapsulation dot1Q 30
Router(config-subif)#ip address 30.0.0.254 255.255.255.0
Router(config-subif)#end
```

Розглянемо деякі команди:

- Int gi 0/0.10 – ця команда підключає віртуальний інтерфейс для роботи з різними VLAN. Цифра після точки – номер VLAN.
- Encapsulation dot1Q 10 – команда налаштування VLAN у sub. Номер після dot1Q – номер VLAN.
- ip address 10.0.0.254 255.255.255.0 – IP адреса виходу пакетів інформації.

Наступним кроком виконуємо налаштування безпеки мережі.

Виконуємо наступні налаштування:

```
Router>en
Router#clock set 11:40:00 16 May 2023
Router#conf t
Router(config)#ip domain name ssh.dom
```

```
Router(config)#crypto key generate rsa
Router(config)#service password-encryption
Router(config)#username Pavlo privilege 15 password 8 school2023
Router(config)#aaa new-model
Router(config)#line vty 0 4
Router(config-line)#transport input ssh
Router(config-line)#logging synchronous
Router(config-line)#exec-timeout 60 0
Router(config-line)#exit
Router(config)#exit
Router#copy running-config startup-config
```

Виконуємо налаштування SSH. Для цього заходимо у маршрутизатор та пишемо команди.

Розглянемо кожну команду:

- `clock set 11:40:00 16 May 2023` – встановлення точного часу для генерації ключа.
- `ip domain name ssh.dom` – вказуємо ім'я домена.
- `crypto key generate rsa` – генерує RSA ключ.
- `service password-encryption` – активуємо шифрування паролей у конфігураційному файлі.
- `username Pavlo privilege 15 password 8 school2023` – створюємо користувача з ім'ям Pavlo та паролем school2023 та рівнем привілеїв 15.
- `aaa new-model` – активуємо протокол AAA.
- `line vty 0 4` – заходимо у режим конфігурування термінальних ліній з 0 по 4.
- `transport input ssh` – вказуємо середовищем доступу через мережу за замовчуванням SSH.
- `logging synchronous` – активуємо автоматичний підйом рядка після відповіді системи на виконані зміни.

- exec-timeout 60 0 – вказуємо час таймаута до автоматичного закриття SSH сесії у 60 хвилин.
- copy running-config startup-config – зберігаємо конфігураційний файл у енергонезалежну пам'ять.

Наступним кроком виконаємо налаштування захисту портів на кожному комутаторі. Для цього відкриваємо комутатор та пишемо наступні команди:

```
Switch>en
Switch#conf t
Switch(config)#interface range fastEthernet 0/X-Y
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport port-security
Switch(config-if-range)#switchport port-security maximum K
Switch(config-if-range)#switchport port-security mac-address sticky
Switch(config-if-range)#switchport port-security violation shutdown
Switch(config-if-range)#end
```

Розглянемо дані команди:

- Interface range fastEthernet 0/X-Y – вибір діапазону інтерфейсів (X – перший потрібний порт, Y – останній потрібний порт);
- switchport mode access – переводимо порти у режим access;
- switchport port-security – вмикаємо захист портів;
- switchport port-security maximum K – обмежуємо число MAC-адрес на інтерфейсі (K – число портів);
- switchport port-security mac-address sticky – вибираємо спосіб дослідження MAC-адрес комутатором;
- switchport port-security violation shutdown – задаємо тип реагування на перевищення числа дозволених MAC-адрес.

3 ГРАФІЧНИЙ ІНТЕРФЕЙС НАЛАШТУВАННЯ МЕРЕЖ

3.1 Розробка графічного інтерфейсу для налаштування мережі

Для управління мережею школи №26, необхідно змінювати її налаштування: надати доступ до мережі, вимкнути його, перевірити працездатність зв'язку, доступ до мережі Інтернет, тощо. Під час подібних операцій конфігурування мережевого обладнання вводяться однотипні команди, які досить часто повторюються. Дана робота є однотипною тим самим сбиває увагу від глобального налаштування. Через це виникає ряд помилок, які у подальшому важко шукати та виправляти Крім того необхідно візуально продемонструвати подібні операції з підключенням або відключенням комп'ютерів від мережі. Для виконання подібних операцій слід вводити відповідні команди, але з суворим дотриманням їх синтаксису та послідовності.

У більшості випадків для адміністратора – вчителя інформатики або іншого вчителя/працівника закладу, який повинен виконати дані налаштування доступу мало часу або компетенцій у налаштуванні мережевого обладнання, керування мережевими сервісами. Таким чином, виникла необхідність створити програму, яка буде брати на себе рутинну задачу введення команд із дотриманням синтаксису та відповідною візуалізацією. Причому, управління програмою повинно бути інтуїтивно зрозумілим та не відрізнятись від роботи інших програм, з якими зазвичай працює користувач.

На рисунку 3.1 наведено структурну схему роботи програми. Програма передбачає створення необхідних команд з відповідним синтаксисом та наступною передачею цих команд до терміналу комутатора.



Рисунок 3.1 – Структурна схема роботи програми

Інтерфейс програми наведено на рисунку 3.2:

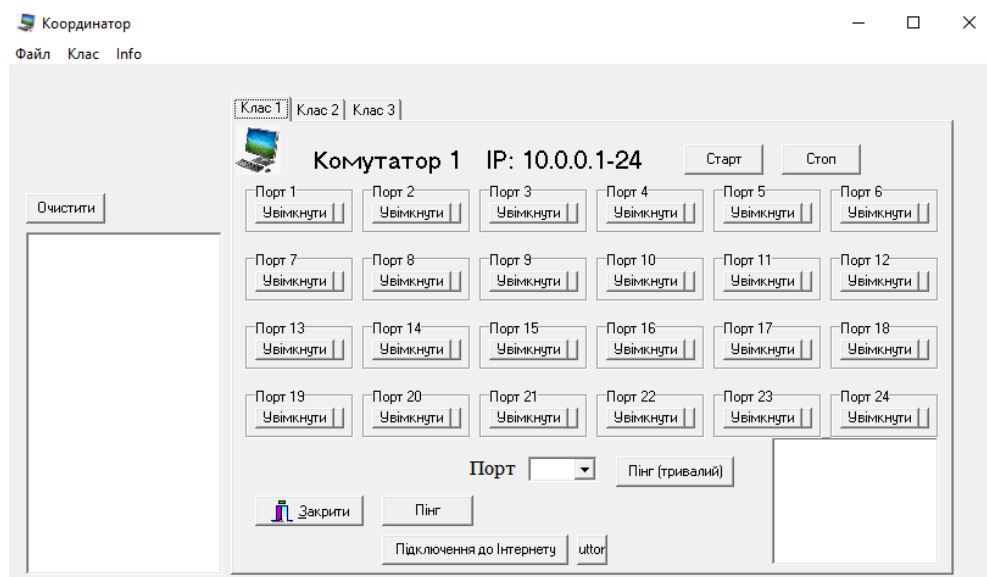


Рисунок 3.2 – Інтерфейс програми

На головному екрану програми містяться наступні елементи:

- меню програми;
- вкладки: “Клас 1”, “Клас 2”, “Клас 3”;
- відповідні підписи: назви комутатора та відповідний діапазон IP адрес;

- кнопки “Старт”, “Стоп”;
- пронумеровані блоки відповідних портів;
- контейнер для формування пакету команд;
- область вибору номеру порта;
- кнопки для проведення діагностики підключення;
- контейнер для відображення результату діагностики;
- кнопка завершення роботи програми.

Управління програмою можна здійснювати як за допомогою головного меню програми так і за допомогою візуальних елементів.

Розглянемо більш детально роботу нашої програми.

Головне меню програми складається із трьох пунктів:

- Файл;
- Клас;
- Info.

Пункт меню “Файл” має підпункти, які наведені на рисунку 3.3:

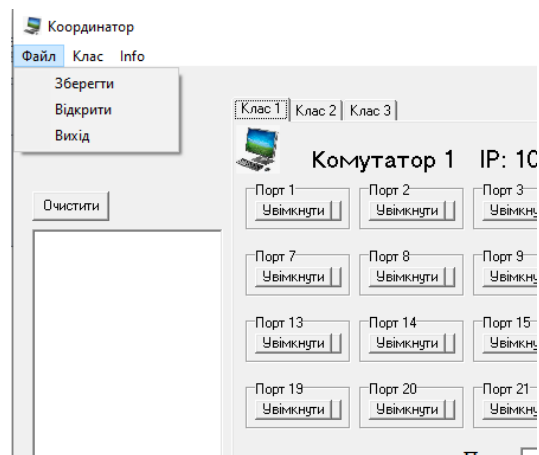


Рисунок 3.3 – Пункт меню “Файл”

Даний пункт передбачає підпункти:

- збереження набору команд у файлі з подальшим їх використанням у майбутньому. Назвати файл команд можна у відповідному

діалоговому вікні у зручній для користувача формі, наприклад: “Підключити праву сторону ЕОМ у другому класі” (рис. 3.4).

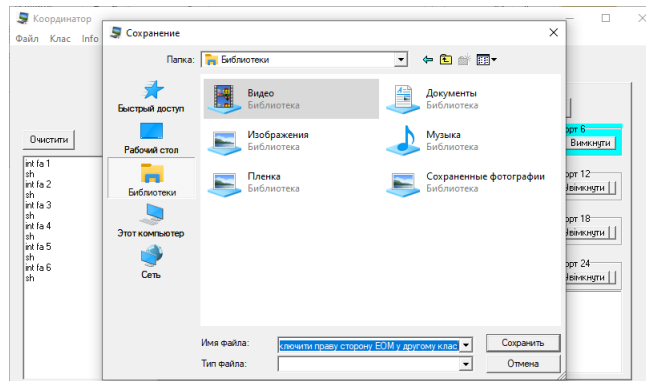


Рисунок 3.4 – Пункт меню “Збереження”

– Завантаження раніше збереженого файлу налаштувань (рис. 3.5).

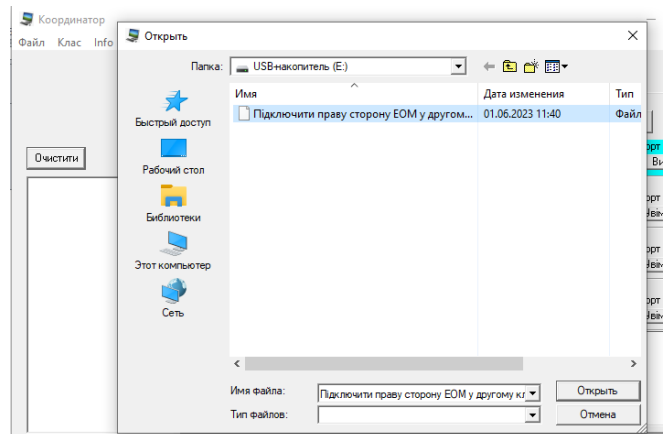


Рисунок 3.5 – Пункт меню “Завантаження”

Таким чином, маємо можливість завчасно створити файли-прототипи для виконання певних налаштувань класів, тощо.

– Останній пункт “Вихід” передбачає завершення роботи програми. Пункт меню “Клас” має підпункти, які наведені на рисунку 3.6:

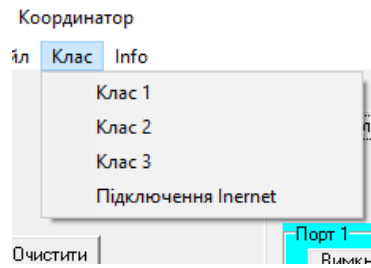


Рисунок 3.6 – Пункт меню “Клас”

Даний пункт дає можливість виконати необхідний вибір класу та виконати перевірку підключення до Інтернету. При виборі відповідного пункту – активується відповідна вкладка з візуальними параметрами, наприклад вкладка “Клас 1” (рис. 3.7):

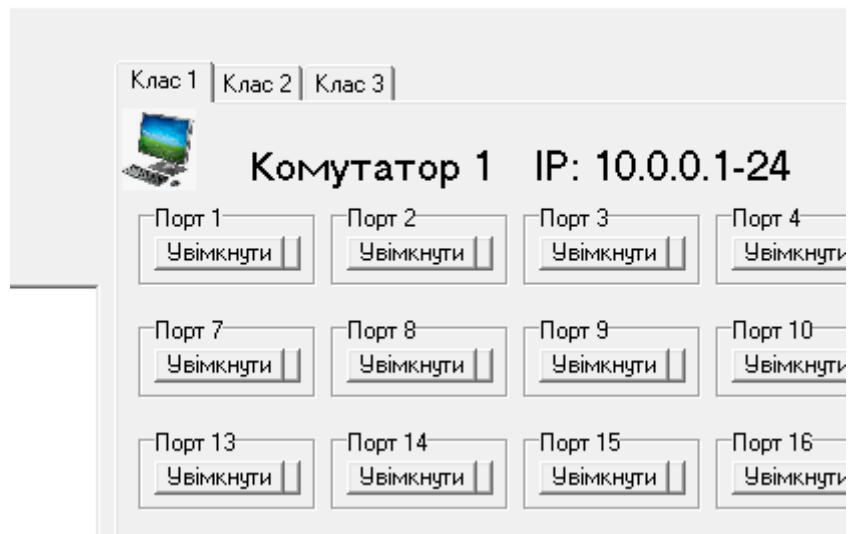


Рисунок 3.7 – Активована вкладка “Клас 1”

Робота пункту “Підключення до Інтернет” відображена на прикладі тестування доступу до сервера Google і відображена на рисунку 3.8:

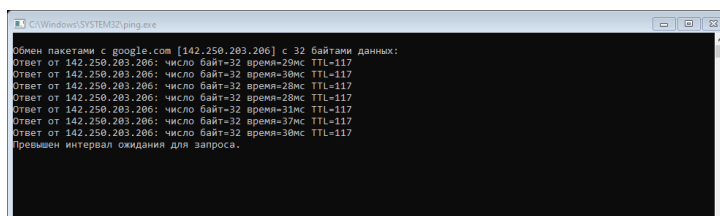


Рисунок 3.8 – Виконання пункту “Підключення до Інтернет”

Пункт меню “Info” виводить інформацію про автора програмного продукту (рис. 3.9):

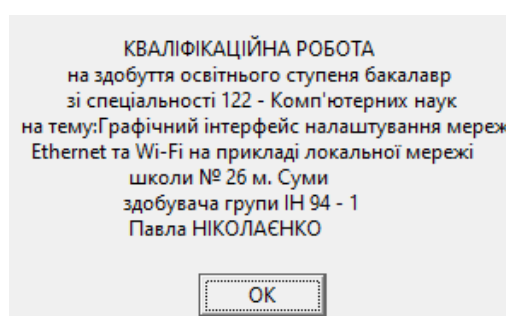


Рисунок 3.9 – Виконання пункту “Info”

Роботу графічного інтерфейсу програми продемонстровано на рисунку 3.10:

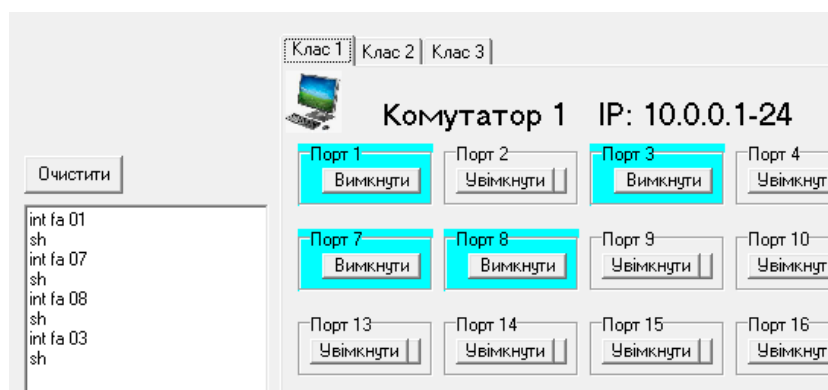


Рисунок 3.10 – Фрагмент програми роботи програми

На рисунку відображені візуальні компоненти. Компоненти, які були включені, відображені іншим кольором. При натиску на відповідні компоненти формуються команди, які записуються у контейнер. При

повторному натиску на включений компонент формується відповідна команда із записом її у контейнері та змінюється колір візуального компонента на сірий.

Таким чином, користувач наочно відслідковує виконані ним операції.

Контейнер має кнопку “Очистити”, що дозволяє очистити його і виконати процес налаштування заново.

Натискаючи на кнопки “Старт” та “Стоп” користувач має можливість виконати налаштування параметрів Telnet, таким чином виконати генерацію команд для налаштування та через буфер обміну передати їх на пристрій. Перелік команд та її призначення і синтаксис був розглянутий у розділі 2 даної кваліфікаційної роботи.

Нижня частина екрану програми (рис. 3.11) відповідає за перевірку зв’язку з комп’ютером та Інтернетом.



Рисунок 3.11 – Фрагмент інтерфейсу програми для перевірки зв’язку

Однією із важливих етапів роботи адміністратора є виявлення доступності до відповідного комп’ютера та мережі Інтернет. Для встановлення зв’язку з необхідним елементом мережі, користувачу пропонується зробити вибір номеру порта (рис. 3.12) і виконати перевірку.

Це передбачено зробити двома шляхами: натиском на кнопку “Пінг (тривалий)” (рис. 3.13) та натиском на кнопку “Пінг” (рис. 3.14).



Рисунок 3.12 – Вибір порта та конфігурування IP адреси

```
Pinging 10.0.0.10 with 32 bytes of data:
```

Рисунок 3.13 – Фрагмент виконання програми для перевірки зв’язку у консольному режимі

Результати виконання команди поміщаються у контейнер, який знаходиться в правій стороні програми (рис. 3.14).

```
Ping statistics for 10.0.0.10:  
Packets: Sent = 4,  
Received = 4, Lost = 0 (0%  
loss).
```

Рисунок 3.14 – Результат виконання програми для перевірки зв’язку

Для перевірки доступності до мережі Інтернет використовується сервер Google, який працює, в нашій програмі, на такому ж принципі, який описано вище для перевірки зв’язку із відповідним портом. Приклади виконання наведені на рисунку 3.15 та 3.16.

```
Approximate round trip
times in milli-seconds:
  Minimum = 19ms,
  Maximum = 24ms, Average
= 21ms
```

Рисунок 3.15 – Результат виконання для перевірки зв'язку з Інтернет

```
Pinging google.com [142.250.186.206] with 32 bytes of data:
Reply from 142.250.186.206: bytes=32 time=20ms TTL=119
Reply from 142.250.186.206: bytes=32 time=21ms TTL=119
Reply from 142.250.186.206: bytes=32 time=19ms TTL=119
Reply from 142.250.186.206: bytes=32 time=20ms TTL=119
```

Рисунок 3.16 – Результат виконання для перевірки зв'язку з Інтернет
(консольний режим)

3.2 Функціональні вимоги до інтерфейсу

Створений нами інтерфейс програми максимально простий та дружній. Були використані стандартні, без зайвого прикрашення компоненти, застосовані звичні, використовувані аналогічними програмами прийоми управління.

Нами було передбачено створення меню програми та візуальні компоненти для управління. З метою, щоб не перевантажувати інтерфейс програми використані кнопки були програмно приховані, а візуально елементи програми були виділені іншим кольором.

Такий підхід спрощує використання програми і унеможлиблює появу помилкових ситуацій.

Нами було передбачено очищення контейнера із командами, та наповнення його заново.

Були реалізовані підходи по збереженню та зчитуванню раніше створених конфігурацій, що в свою чергу підвищує експлуатаційні можливості програми.

Для реалізації спрощеної роботи були передбачені кнопки, які “приховуючи” роботу програми, виводять лише результати виконання (рис. 3.14 та 3.15). Також передбачено виконання команд в консольному режимі та перегляду їх результатів (рис. 3.13 та 3.16) .

При помилкових діях користувача, виводяться відповідні інформаційні повідомлення (рис. 3.17):

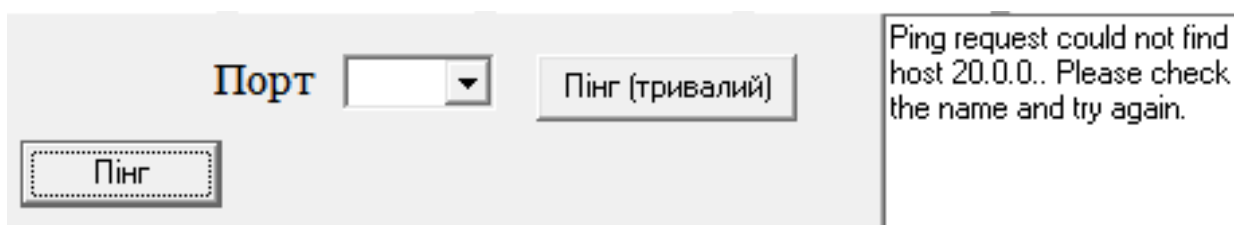


Рисунок 3.17 – Помилкова ситуація

Таким чином, створений нами інтерфейс, відповідає критеріям програм, використовує інтуїтивно зрозумілі для користувача функції, стандартні кнопки керування програмою, що призводить до ефективного використання розробленої програми.

3.3 Розробка інтерфейсу

Для розробки інтерфейсу була використана мова програмування Delphi.

Нами був створений метод, який реалізує “підключення” та “відключення” елемента порта:

```
procedure TForm1.Button148Click(Sender: TObject);
begin //порт X увімкнути
    groupBox69.Color:=clAqua;
    text_comand1:='int fa X' ;
```

```

text_comand2:='sh';
memo2.Lines.Add(text_comand1);
memo2.Lines.Add(text_comand2);
button148.Hide;
button147.Show;
end;

procedure TForm1.Button147Click(Sender: TObject);
begin //порт X вимнути
  groupBox69.Color:=clBtnFace;
  text_comand1:='int fa X' ;
  text_comand2:='no sh';
  memo2.Lines.Add(text_comand1);
  memo2.Lines.Add(text_comand2);
  button147.Hide;
  button148.Show;
end;

```

У наведеному прикладі елемент **X** визначає номер порта. Властивості **Hide** та **Show** – відповідно приховують та виводять відповідні елементи інтерфейсу.

Для збереження набору команд і подальшого його завантаження були написані методи:

```

procedure TForm1.Pthtunb1Click(Sender: TObject);
begin
  if SaveDialog1.Execute then
  begin
    Memo2.Lines.SaveToFile(SaveDialog1.FileName);
    ShowMessage('Файл збережено');
  end;
end;

```

```

procedure TForm1.N3Click(Sender: TObject);
begin
    if OpenFileDialog1.Execute then
    begin
        memo2.Lines.LoadFromFile(OpenDialog1.FileName);
        ShowMessage('Файл завантажено');
    end;
end;

```

Результатом виконання даних фрагментів коду є завантаження або збереження набору команд із/в контейнер. Після виконання відповідно на екрані з'являються діагностичні повідомлення про виконання операції.

Для організації виведення результатів виконання команд був створений метод:

```

procedure TForm1.Button106Click(Sender: TObject);
begin
    Memo1.Clear;
    ExecuteCommand('ping
30.0.0.'+intToStr(strToInt(combobox3.Text)+100), Memo1);
end;

```

Для реалізації цього метода було написано процедуру ExecuteCommand з параметрами передачі сформованої команди та переведенням їх повідомлень у відповідний контейнер.

Формування IP адреси було побудоване за принципом вибраного комутатора та додавання до неї вибраного номеру порта. Це реалізовано у наступному кодї:

```

procedure TForm1.ComboBox3Change(Sender: TObject);
begin
    Application.processMessages;
    label6.Caption:= '30.0.0.'+intToStr(strToInt(combobox3.Text)+100);

```

end;

Де метод `Application.processMessages` “змушує” програму відразу виконувати дану операцію, що в свою чергу створює необхідну нам адресу.

За допомогою глобальних змінних `text_comand1`, `text_comand2` було досягнуто мети передачі вказаних команд маршрутизатора у контейнер із подальшим їх опрацюванням.

Повний код програми наведено у додатку.

3.4 Тестування розробленого інтерфейсу

Після запуску програми у контейнері (рис. 3.18) формується список команд:

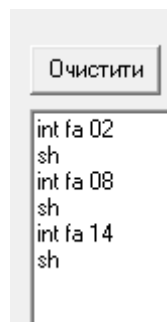


Рисунок 3.18 – Список команд

Даний список відповідає синтаксису команд для маршрутизатора. Копіюючи його у Telnet було виконано перевірку на правильність виконання, що дало вірний результат.

Виконуючи пункти збереження та завантаження файлу команд ми отримали також правильний результат.

При тестуванні відповідних перевірок ліній зв'язку, що наведені на рисунках: 3.8, 3.13, 3.14, 3.15, 3.16 та 3.17, а також при формуванні списку

команд (рис. 3.10, 3.12) ми бачимо правильні повідомлення, що в свою чергу стверджує працездатність нашої програми.

Список команд, сформованих у контейнері, використовуючи відповідні пункти меню, можемо зберегти у файлі і при необхідності завантажити його до програми. Таким чином, використовуючи створену програму можна підготувати конфігурації файлів – команд і в подальшому їх використовувати.

В ході виконання кваліфікаційної роботи розроблена програма дозволила значно підвищити швидкість налаштування мережевого обладнання, спростити конфігурацію базових параметрів. Дане програмне забезпечення стане корисне новачкам, які навчаються налаштовувати комп'ютерні мережі, та може стати у нагоді досвідченим адміністраторам при конфігурації великої кількості мережевого обладнання.

ВИСНОВКИ

В ході виконання кваліфікаційної бакалаврської роботи "Графічний інтерфейс налаштування мереж Ethernet та Wi-Fi на прикладі локальної мережі школи №26" було розглянуто та проаналізовано велику кількість джерел інформації, що дають можливість зробити наступні висновки: для закладів освіти використання локальних мереж, особливо Ethernet та Wi-Fi, на сьогоднішній день, є невід'ємною складовою освітнього процесу; саме технології Ethernet та Wi-Fi забезпечують надійність швидкість та безпеку передачі даних.

Кваліфікаційна робота має практичну цінність, оскільки дозволить поліпшити процес налаштування мереж Ethernet та Wi-Fi в шкільній мережі, зменшити кількість помилок та збільшити ефективність роботи мережі в цілому. Результати дослідження та розробки можуть бути корисними для шкільних адміністраторів, які відповідають за налаштування мереж та забезпечення їх безперебійної роботи.

Виконані: моделювання топології, створення VLAN та налаштування безпеки мережі, створено програму із графічним інтерфейсом, перевірено її працездатність та правильність у роботі.

Подано повну картину розробки топології мережі, включаючи ролі різних пристроїв і їх функціональні можливості. Це сприяє налагодженню ефективної та надійної мережевої інфраструктури, яка задовольняє потреби шкільної локальної мережі.

Створений інтерфейс програми максимально простий та інтуїтивно зрозумілий. що забезпечує максимальну зручність у її роботі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дейтел Г. Мережі комп'ютерів: принципи, технології, протоколи: підручник / Г. Дейтел, П. Дейтел, К. Саджівський. – Київ: Навчальна книга - Богдан, 2018.
2. Жураковський Б.Ю., Зенів І.О. Комп'ютерні мережі. Частина 1. Навчальний посібник / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 336 с.
3. Карпенко О. Мережні технології та системи: підручник / О. Карпенко, Г. Пасічник, А. Глинський. - Київ: Ліра-К, 2019.
4. Ключко М. С. Мережні технології: підручник / М. С. Ключко, О. А. Дем'яненко. – Київ: Центр учбової літератури, 2019.
5. Сидоренко О. Мережеві технології: навчальний посібник / О. Сидоренко, В. Кобзарьов, А. Брежнева. – Київ: Видавництво "Право", 2018.
6. Шаповал О. Мережеві технології: підручник / О. Шаповал, А. Горобець. – Київ: Центр учбової літератури, 2019.
7. Cisco Packet Tracer [Електронний ресурс] – Режим доступу до ресурсу: <https://www.netacad.com/>.
8. Cisco Press [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ciscopress.com/>.
9. Cisco DevNet [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.cisco.com/>.
10. Cisco Blogs [Електронний ресурс] – Режим доступу до ресурсу: <https://blogs.cisco.com/>.
11. IEEE Xplore [Електронний ресурс] – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/Xplore/home.jsp>.

ДОДАТОК

Код програми:

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls, ExtCtrls, jpeg, ComCtrls,
  ShellAPI, Buttons, Menus;
type
  TForm1 = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    TabSheet3: TTabSheet;
    Label1: TLabel;
    Image1: TImage;
    Button2: TButton;
    Memo1: TMemo;
    GroupBox1: TGroupBox;
    Button4: TButton;
    Button5: TButton;
    Memo2: TMemo;
    Button6: TButton;
    GroupBox2: TGroupBox;
    Button7: TButton;
    Button8: TButton;
    GroupBox3: TGroupBox;
    Button9: TButton;
    Button10: TButton;
    ...
  MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    Info1: TMenuItem;
    Phtunb1: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N11: TMenuItem;
    N21: TMenuItem;
    N31: TMenuItem;
```



```

OpenDialog1: TOpenDialog;
SaveDialog1: TSaveDialog;
Inernet1: TMenuItem;
procedure Button2Click(Sender: TObject);
procedure ComboBox1Change(Sender: TObject);
procedure Button53Click(Sender: TObject);
procedure Button155Click(Sender: TObject);
procedure N11Click(Sender: TObject);
procedure N21Click(Sender: TObject);
procedure N31Click(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure Info1Click(Sender: TObject);
procedure Pthtunb1Click(Sender: TObject);
procedure N3Click(Sender: TObject);
procedure Inernet1Click(Sender: TObject);

```

```

private
  { Private declarations }
public
  { Public declarations }
end;

```

```

var
  Form1: TForm1;
  text_comand1, text_comand2:string;

```

implementation

```
{ $R *.dfm }
```

```

procedure ExecuteCommand(const Command: string; Memo: TMemo);
const
  ReadBuffer = 2400;
var
  Security: TSecurityAttributes;
  ReadPipe, WritePipe: THandle;
  StartInfo: TStartupInfo;
  ProcessInfo: TProcessInformation;
  Buffer: PAnsiChar;
  BytesRead: DWORD;
  Apprunning: DWORD;
begin
  with Security do

```

```

begin
  nlength := SizeOf(TSecurityAttributes);
  binherithandle := true;
  lpsecuritydescriptor := nil;
end;

if CreatePipe(ReadPipe, WritePipe, @Security, 0) then
begin
  Buffer := AllocMem(ReadBuffer + 1);
  FillChar(StartInfo, SizeOf(TStartupInfo), #0);
  StartInfo.cb := SizeOf(TStartupInfo);
  StartInfo.hStdOutput := WritePipe;
  StartInfo.hStdInput := ReadPipe;
  StartInfo.dwFlags := STARTF_USESTDHANDLES or
STARTF_USESHOWWINDOW;
  StartInfo.wShowWindow := SW_HIDE;

  if CreateProcess(nil, PChar(Command), @Security, @Security, true,
  NORMAL_PRIORITY_CLASS, nil, nil, StartInfo, ProcessInfo) then
  begin
    repeat
      Apprunning := WaitForSingleObject(ProcessInfo.hProcess, 100);
      Application.ProcessMessages;
    until (Apprunning <> WAIT_TIMEOUT);

    while True do
    begin
      BytesRead := 0;
      ReadFile(ReadPipe, Buffer[0], ReadBuffer, BytesRead, nil);
      Buffer[BytesRead] := #0;
      OemToAnsi(Buffer, Buffer);
      Memo.Lines.Add(String(Buffer));
      if BytesRead < ReadBuffer then
        Break;
      end;
    end;
  end;

  FreeMem(Buffer);
  CloseHandle(ProcessInfo.hProcess);
  CloseHandle(ProcessInfo.hThread);
  CloseHandle(ReadPipe);
  CloseHandle(WritePipe);
end;
end;

```

```
procedure TForm1.Button3Click(Sender: TObject);
begin
  Memo1.Clear;
  ExecuteCommand('ping google.com', Memo1);
end;
```

```
procedure TForm1.Button4Click(Sender: TObject);
begin //порт 1
  groupBox1.Color:=clAqua;
  text_comand1:='int fa 01' ;
  text_comand2:='sh';
  memo2.Lines.Add(text_comand1);
  memo2.Lines.Add(text_comand2);
  button4.Hide;
  button5.Show;
end;
```

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  groupBox1.Color:=clBtnFace;
  text_comand1:='int fa 01' ;
  text_comand2:='no sh';
  memo2.Lines.Add(text_comand1);
  memo2.Lines.Add(text_comand2);
  button5.Hide;
  button4.Show;
end;
```

```
//-----
procedure TForm1.FormCreate(Sender: TObject);
begin
  memo2.Clear;

end;
```

```
procedure TForm1.Button6Click(Sender: TObject);
begin
  memo2.Clear;
end;
```

```
procedure TForm1.Button8Click(Sender: TObject);
begin //порт 2 увімкнути
  groupBox2.Color:=clAqua;
```

```

text_comand1:='int fa 02' ;
text_comand2:='sh';
memo2.Lines.Add(text_comand1);
memo2.Lines.Add(text_comand2);
button8.Hide;
button7.Show;
end;

procedure TForm1.Button7Click(Sender: TObject);
begin //порт 2 вимкнути
  groupBox2.Color:=clBtnFace;
  text_comand1:='int fa 02' ;
  text_comand2:='no sh';
  memo2.Lines.Add(text_comand1);
  memo2.Lines.Add(text_comand2);
  button7.Hide;
  button8.Show;
end;

...
procedure TForm1.ComboBox1Change(Sender: TObject);
begin
  Application.ProcessMessages;
  label2.Caption:= '10.0.0.'+combobox1.Text;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  Memo1.Clear;
  ExecuteCommand('ping 10.0.0.'+combobox1.Text, Memo1);
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  WinExec(PAnsiChar(AnsiString('ping google.com -t')), SW_SHOW);
end;

procedure TForm1.Button53Click(Sender: TObject);
begin
  WinExec(PAnsiChar(AnsiString('ping 10.0.0.'+combobox1.Text+' -t')),
  SW_SHOW);
end;
//-----komutator 2
procedure TForm1.Button56Click(Sender: TObject);
begin //порт 1 увімкнути

```

```

    groupBox25.Color:=clAqua;
    text_comand1:='int fa 1' ;
    text_comand2:='sh';
    memo2.Lines.Add(text_comand1);
    memo2.Lines.Add(text_comand2);
    button56.Hide;
    button55.Show;
end;
...
procedure TForm1.Button105Click(Sender: TObject);
begin
    WinExec(PAnsiChar(AnsiString('ping 20.0.0.'+combobox2.Text+' -t')),
    SW_SHOW);
end;

procedure TForm1.Button54Click(Sender: TObject);
begin
    Memo3.Clear;
    ExecuteCommand('ping 20.0.0.'+combobox2.Text, Memo3);
end;

procedure TForm1.Button104Click(Sender: TObject);
begin
    Memo1.Clear;
    //ExecuteCommand('8.8.8.8', Memo1);
    ExecuteCommand('ping google.com', Memo3);
end;

procedure TForm1.Button103Click(Sender: TObject);
begin
    WinExec(PAnsiChar(AnsiString('ping google.com -t')), SW_SHOW);
end;

procedure TForm1.ComboBox2Change(Sender: TObject);
begin
    Application.processMessages;
    label4.Caption:= '20.0.0.'+combobox2.Text;
end;
//***** komutator 3

procedure TForm1.Button108Click(Sender: TObject);
begin //порт 1 увімкнути
    groupBox49.Color:=clAqua;

```

```

    text_comand1:='int fa 1' ;
    text_comand2:='sh';
    memo2.Lines.Add(text_comand1);
    memo2.Lines.Add(text_comand2);
    button108.Hide;
    button107.Show;
end;
...
procedure TForm1.ComboBox3Change(Sender: TObject);
begin
    Application.processMessages;
    label6.Caption:= '30.0.0.'+intToStr(strToint(combobox3.Text)+100);
end;

procedure TForm1.Button157Click(Sender: TObject);
begin
    WinExec(PAnsiChar(AnsiString('ping
30.0.0.'+intToStr(strToint(combobox3.Text)+100)+' -t')), SW_SHOW);
end;

procedure TForm1.Button106Click(Sender: TObject);
begin
    Memo1.Clear;
    ExecuteCommand('ping 30.0.0.'+intToStr(strToint(combobox3.Text)+100),
Memo1);
end;

procedure TForm1.Button156Click(Sender: TObject);
begin
    Memo1.Clear;
    ExecuteCommand('ping google.com', Memo4);
end;

procedure TForm1.Button155Click(Sender: TObject);
begin
    WinExec(PAnsiChar(AnsiString('ping google.com -t')), SW_SHOW);
end;

procedure TForm1.N11Click(Sender: TObject);
begin
    PageControl1.ActivePage := TabSheet1;
end;

procedure TForm1.N21Click(Sender: TObject);

```

```

begin
    PageControl1.ActivePage := TabSheet2;
end;

procedure TForm1.N31Click(Sender: TObject);
begin
    PageControl1.ActivePage := TabSheet3;
end;

procedure TForm1.N4Click(Sender: TObject);
begin
    close;
end;

procedure TForm1.Info1Click(Sender: TObject);
var strLine1, strLine2, strLine3, strLine4, strLine41, strLine42, strLine5,
    strLine6 : String;
begin
    strLine1 := '                КВАЛІФІКАЦІЙНА РОБОТА ';
    strLine2 := '                на здобуття освітнього ступеня бакалавр ';
    strLine3 := '                зі спеціальності 122 - Комп'ютерних наук';
    strLine4 := ' на тему:Графічний інтерфейс налаштування мереж';
    strLine41:= ' Ethernet та Wi-Fi на прикладі локальної мережі ';
    strLine42:= Сумського закладу загальної середньої освіти I-III
                ступенів №26м. Суми ';
    strLine5 := '                здобувача групи ІН 94 - 1 ';
    strLine6 := '                Павла НІКОЛАЄНКО ';
    ShowMessage(strLine1 + sLineBreak + strLine2 + sLineBreak +
                strLine3 + sLineBreak + strLine4+sLineBreak +
strLine41+sLineBreak+ strLine42+sLineBreak+ strLine5 +sLineBreak
+strLine6);
end;

procedure TForm1.Pthtunb1Click(Sender: TObject);

begin
    if SaveDialog1.Execute then // Відкрити діалогове вікно збереження
    файлу
    begin
        Memo2.Lines.SaveToFile(SaveDialog1.FileName);
        ShowMessage('Файл збережено');
    end;
end;
end;

```

```
procedure TForm1.N3Click(Sender: TObject);
begin
    if OpenDialog1.Execute then // Відкрити діалогове вікно збереження
    файлу
    begin
        memo2.Lines.LoadFromFile(OpenDialog1.FileName);
        ShowMessage('Файл завантажено');
    end;
end;
```

```
procedure TForm1.Inernet1Click(Sender: TObject);
begin
    WinExec(PAnsiChar(AnsiString('ping google.com -t')), SW_SHOW);
end;
```

end.