МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

«Інформаційно-комунікаційна технологія налаштування протоколу Generic Routing Encapsulation у мережах Ethernet»

Завідувач випускаючої кафедри — **Довбиш А.С.**

Керівник роботи — **Великодний Д.В.**

Студент групи ІК.м-01 — **Ситніков В.О.**

СУМИ 2021

Сумський державний університет
(назва вузу)

Факультет *ЕлІТ* Кафедра *Комп'ютерних наук*
Спеціальність *«122 -Комп'ютерні науки»*

Затверджую:
зав.кафедрою _____
_____
"_____" _____ 20___р.

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Ситнікову Владиславу Олеговичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) *Інформаційно-комунікаційна технологія налаштування протоколу Generic Routing Encapsulation у мережах Ethernet*

затверджую наказом по інституту від "____" _____ 20 ___ р. № _____

2. Термін здачі студентом закінченого проекту (роботи) _____

3. Вхідні данні до проекту (роботи) _____
_____
_____
_____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз віртуальних приватних мереж. 2) Вибір способів для реалізації. Постановка задачі 3) Моделювання та проектування GRE мережі 4) Розробка інформаційного та програмного забезпечення

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 _____

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	*Аналіз віртуальних приватних мереж*		
2.	*Вибір способів для реалізації. Постановка задачі*		
3.	*Моделювання та проектування GRE мережі*		
4.	*Розробка інформаційного та програмного забезпечення*		
5.	*Оформлення пояснювальної записки до магістерської роботи*		

Студент – дипломник _____
(підпис)

Керівник проекту _____
(підпис)

РЕФЕРАТ

Записка: 61 стор., 37 рис., 1 табл., 5 додатків, 40 джерел.

Об’єкт дослідження — графічний інтерфейс налаштування протоколу GRE у мережах Ethernet.

Мета роботи — розробка графічного інтерфейсу, який допоможе спростити налаштування GRE мережі.

Методи дослідження — створення додатку для спрощення конфігурацій

Результати — у результаті роботи було створено мережу з GRE у симуляторі Cisco Packet Tracer. Проведено налаштування і перевірка на дієздатність мережі. Також було розроблено графічний інтерфейс у вигляді веб-сторінки , який дає змогу ввести початкові дані, а потім згенерувати код для обладнання автоматично. Також перевіряються вхідні дані, щоб не було допущено помилки. За допомогою інтерфейсу мережу можуть налаштувати як новачки так і професіонали. Проведено тестування веб-додатку з застосуванням віртуальної мережі GRE, і доведено працездатність та можливість використовувати додаток на справжньому устаткуванні.

ГРАФІЧНИЙ ІНТЕРФЕЙС, JAVASCRIPT, GRE ТУНЕЛЬ, CSS, HTML,

VUE.JS, CISCO PACKET TRACER

ЗМІСТ

ВСТУП

У наш час віртуальні приватні мережі стають все більш популярними. Це відбувається через наявність високої безпеки інфраструктури та економічність цього виду зв’язку.

Використовуючи ВПМ будь-який працівник зможе під’єднатися до ресурсів корпоративної мережі зі звичайного локального телефону, що дозволяє заощадити кошти. Збільшити безпеку підключення допомагають так звані тунельні з'єднання, що дозволяють здійснити вхід у мережу лише аутентифікованим користувачам.

Зазвичай, людям потрібно передати інформацію з одного офісу в інший, які знаходяться в різних містах. Звичайно, можна вирішити цю проблему, надавши віддалений доступ за допомогою телефонних ліній.

Мінусом такого рішення є те, що дзвінки до інших країн спричиняють великі витрати. Тому є ще один вихід під назвою GRE. Загальна інкапсуляція маршрутів створена для передачі групового та багатопротокольного трафіку між декількома майданчиками, зв’язок між якими забезпечується лише по IP.

Через людські фактори дуже висока ймовірність помилки при налаштуванні великої кількості мереж. Отже, було прийнято рішення створити такий додаток, за допомогою якого можна автоматично налаштовувати спочатку інтерфейси, а потім GRE. За зразок візьмемо роутер Cisco 2911.

Опрацювавши багато джерел, схожих додатків не було знайдено. Для створення графічного інтерфейсу будемо використовувати мови програмування JavaScript, мови розмітки HTML/CSS та ін.

1 АНАЛІЗ ВІРТУАЛЬНИХ ПРИВАТНИХ МЕРЕЖ

1.1 Огляд досліджень і публікацій

VPN — це загальна назва технологій, за допомогою яких можна забезпечити декілька мережевих з'єднань над іншою мережею.

Створення VPN пов'язують з послугою Centrex у телефонії. Згадка про Centrex з'явилася в США в середині XX століття. Вона означала загальну назву способу передачі послуг зв'язку юзерам декількох компаній на основі використання спільних пристроїв станції Private Branch Exchange (PBX). Пізніше, коли почали впроваджувати обладнання з програмним управлінням, цей термін набув більш глибокого змісту [1].

Основним плюсом Centrex було те, що, створюючи виділені корпоративні мережі, фірми та компанії економили дуже велику частину коштів, необхідних для придбання, ремонту та використання власних станцій. Абоненти утворюють між собою так звані ізольовані групи для юзерів Closed Users Group (CUG), які мають лімітований доступ всередині, незважаючи на те, що клієнти використовують прилади та ресурси мереж загального користування.

Для того щоб подолати обмеження Centrex було запропоновано створити VPN (Virtual Private Network), яка уособлює альянс CUG, об'єднаний в одне ціле і знаходиться на дистанції один від одного [2].

У кінці XX століття VPN почав активно використовуватися в компаніях надання доступу до інтернету в домашніх мережах. На ту пору ціни на зовнішній трафік були дуже великими, проте внутрішній був безкоштовним. Це давало право користувачам обмінюватися даними у своїх приватних мережах безлімітно. Через це VPN одержав високу оцінку у світі.

У наш час приватні мережі набувають все більшої популярності. Системи передачі сигналів застосовуються у багатьох сферах життєдіяльності:

- в охоронно-пожежних системах;

- у системах віддаленого банкінгу;
- у системах "розумний дім";
- у системах захисту передачі даних.

Зв'язки виконуються по мережах з малим, а ще частіше незнайомим рівнем довіри. Незважаючи на це ступінь довіри базової мережі не буде залежати від створеної логічної мережі. Цьому сприяють використання різних способів криптографії (аутентифікація, шифрування, інфраструктура відкритих ключів тощо) [3].

VPN забезпечує такі види зв'язку, як node-node, node-network and network-network, залежно від протоколів, що застосовуються та призначення (рис.1.1).

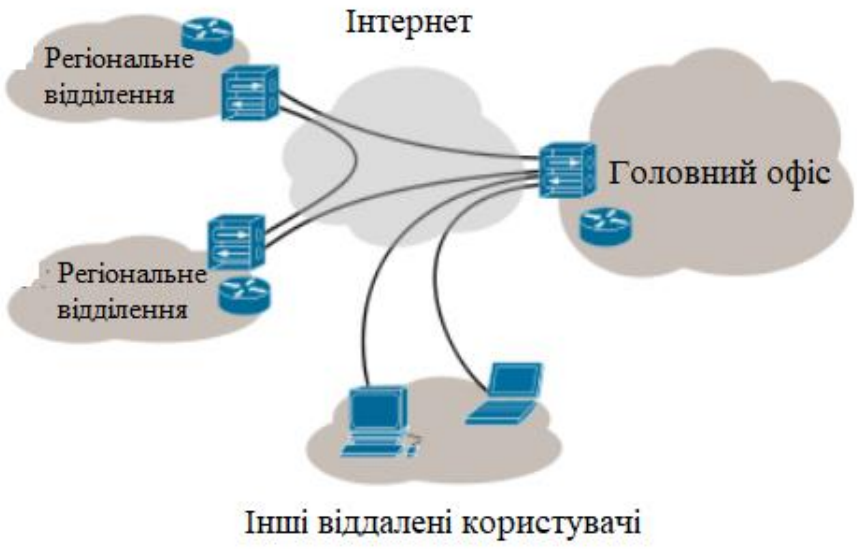

Рисунок 1.1 – VPN в інтернеті

Взагалі VPN налаштовується на рівнях не вище мережевого. Це відбувається тому, що використання криптографії дає право застосовувати транспортні протоколи в сталому вигляді.

Найчастіше, щоб створити віртуальну мережу вживається інкапсуляція PPP в деякий інакший - IP. Цей метод застосовується в протоколі тунелювання точка-точка (PPTP) та PPPoE (Ethernet), але і ці способи відрізняються один від

одного. Взагалі VPN застосовується не лише при створенні приватних мереж, а й для отримання доступу до Інтернету.

Якщо правильно використовувати та реалізовувати професійне програмне обладнання, то за допомогою VPN можна гарантувати підвищений рівень шифрування даних, які передаються, а також при точному налаштуванні всіх складників забезпечити секретність у мережі [4].

VPN утворюють дві частини:

- зовнішня - крізь неї віртуальна приватна мережа проходить інкапсульоване сполучення;
- внутрішня - підвладна або загальнодоступна, їх може бути декілька.

З'єднання ізольованого користувача з VPN виконується зі сприяння ПО, що підключається і до зовнішньої, і до внутрішньої мережі.

Якщо віддаленому юзеру треба приєднатися чи отримати можливість доступу з другою захищеною мережею, то сервер зобов'язує ідентифікуватися, а вже потім аутентифікуватися. Потім вдало пройшовши ці процедури, абонент отримує повний перелік привілегій, щоб працювати в мережі. Іншими словами виконується авторизація [5].

VPN надають життєво важливі послуги підприємствам, урядам, військовим організаціям і навіть приватним особам. VPN забезпечують безпечний доступ до вашої локальної мережі. Без VPN віддалений доступ до конфіденційної інформації був би неможливим. VPN дозволяють з’єднувати дві або більше мереж. Наприклад, філія банку, розташована в штаті Невада, і його штаб-квартира, розташована в Техасі, підключаються один до одного через Інтернет, щоб безпечно ділитися своїми ресурсами (такими як виписки з банку, записи про іпотеку тощо). За допомогою VPN обидва місця логічно з’єднані разом, використовуючи безпечний Інтернет як основу [6].

1.2 Класифікація VPN

Взагалі є декілька основних параметрів класифікації VPN (рис. 1.2).

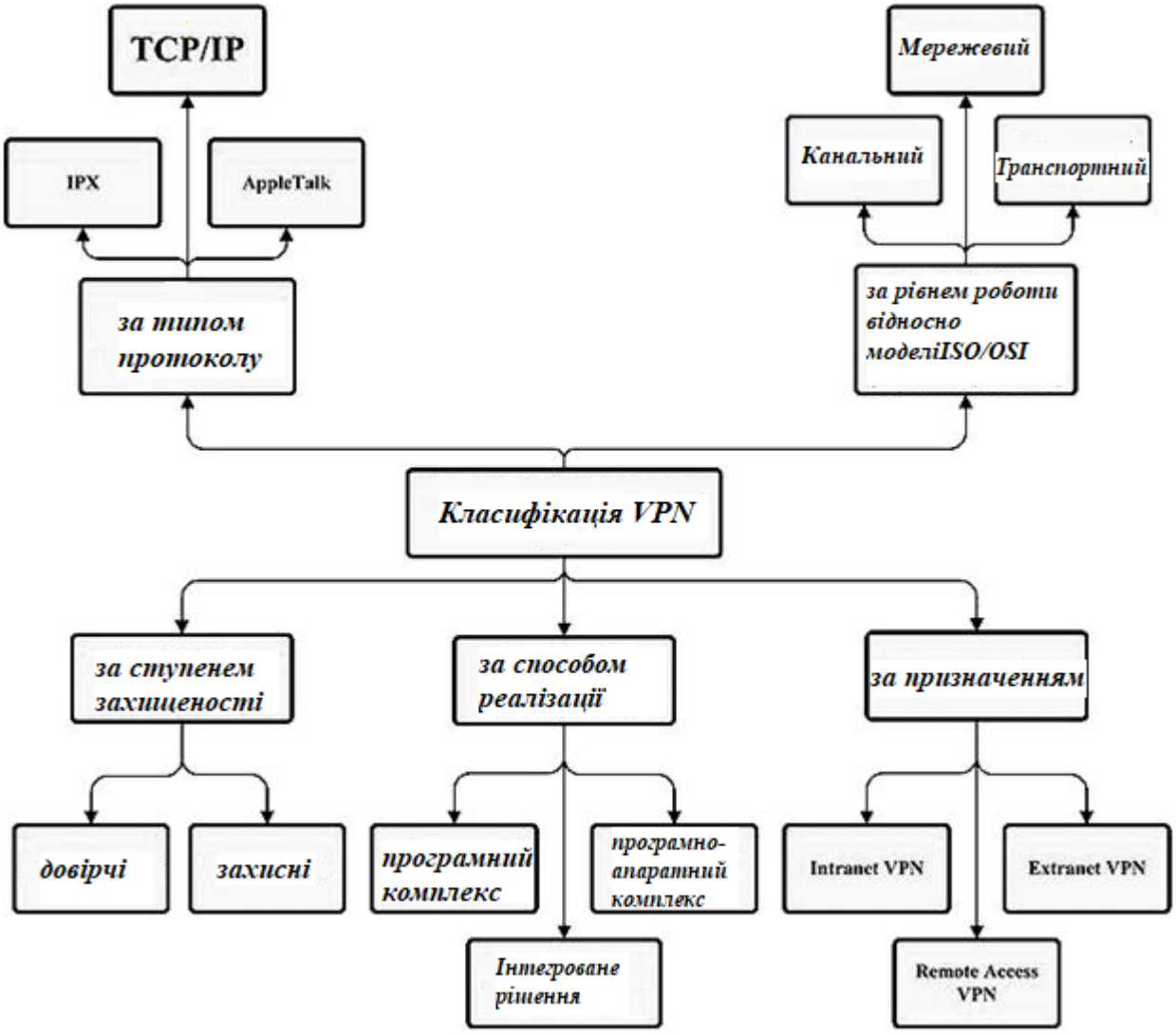

Рисунок 1.2 – Класифікація VPN

VPN має безліч своїх переваг. Розглянемо найголовніші з них:

- Дозволяє застарілим системам працювати віддалено, навіть якщо не має розробки для таких сценаріїв. Наприклад, файлові сервери SMB.
- Забезпечення другого рівня захисту від неправильно налаштованих, не виправлених або погано розроблених внутрішніх служб. Наприклад, веб-сайти внутрішньої мережі, які погано обслуговуються, або використовують застарілу криптографію.

- Захист внутрішніх мережевих серверів від зовнішніх неавтентифікованих зловмисників, обмеживши доступ до мережі для автентифікованих пристроїв. Наприклад, сховища файлів або бази даних.
- Захист пристроїв користувача від мережевих атак, забороняючи прямі підключення до та з локальної мережі. Наприклад, атаки ARP-спуфінгу або атака на відкриті мережеві інтерфейси на мобільному пристрої.
- Примусовий трафік між пристроєм і зовнішніми службами за допомогою внутрішніх засобів захисного моніторингу. Це захищає від різноманітних загроз. Наприклад, перевірка веб-вмісту на наявність шкідливого коду.
- Включення бізнес-моніторингу та/або фільтрації мережевого трафіку користувачів з юридичних причин, дисципліни та обов'язку піклуватися. Наприклад, блокування доступу до нелегальних веб-сайтів [7].

Захист даних під час передачі є одним з найважливіших аспектів безпеки, які слід враховувати при використанні мобільних пристроїв. Зловмисники, які мають доступ до незахищених даних (або даних із неналежним захистом), можуть перехопити та змінити дані, що може завдати шкоди. Тому важливо вирішити, які з перерахованих вище переваг мають відношення до вашої організації, перш ніж приймати рішення, яку технологію VPN використовувати та як її використовувати [8].

Більшість операційних систем мають вбудований клієнт VPN, який можна налаштувати на пристрої або керувати віддалено. Інтегровані клієнти, як правило, безкоштовні у використанні, надійно працюють і оновлюються автоматично, але також можуть мати відносно обмежені функціональні можливості. Наприклад, часто немає можливості налаштувати правила маршрутизації, винятки або розділене тунелювання [9].

Використання стороннього клієнта VPN збільшує ризик того, що інтеграція операційної системи буде поганою, і, отже, деякі дані можуть надсилатися за межі VPN. Це також збільшує кількість пакетів програмного забезпечення, які

необхідно оновлювати, додаючи ймовірність того, що деяке застаріле програмне забезпечення буде використовуватися.

Тільки трафік, який маршрутизується через VPN, буде захищений ним, тому ви можете примусово налаштувати маршрутизацію всього трафіку , щоб трафік користувача не проходив за межами цього з'єднання. Для цього знадобиться або сам клієнт VPN, або брандмауер клієнта, налаштований для запобігання встановлення з'єднань за межами VPN. Якщо примусова VPN не може підключитися, мережевий трафік з пристрою не буде можливим, якщо VPN не вимкнено. Використання додаткової віртуальної приватної мережі дозволяє користувачам вимкнути VPN і уникнути захисних служб моніторингу та аудиту. Це збільшує ризик атаки мобільних пристроїв через мережу та обходу користувачів обмежень корпоративної політики [10].

Щоб забезпечити переваги, VPN має встановити з'єднання, залишатися на зв'язку, поки пристрій використовується, і повторно під'єднатися, якщо з'єднання тимчасово втрачено. Як правило, існує три способи реалізації цієї поведінки: автоматичний, ініційований і ручний.

- Автоматичні VPN викликаються пристроєм щоразу, коли програмне забезпечення пристрою вимагає підключення до мережі.
- Запущені VPN з'являються щоразу, коли встановлюються певні мережеві з'єднання з визначеного списку (наприклад, сайти внутрішньої мережі).
- Ручні VPN вимагають, щоб користувач пристрою ініціював VPN, явно вирішивши виконати дію (наприклад, запустити програму та натиснути підключитися). Користувачеві, ймовірно, доведеться знову зробити цей крок, якщо під час або після використання з'єднання розірветься [11].

Залежно від того, чи є VPN примусовим чи необов'язковим, вибір між автоматичним, активованим і ручним може бути рішенням щодо зручності, а не безпеки. Якщо VPN примусовий, але ініціація здійснюється вручну, це буде поганою взаємодією з користувачем, оскільки пристрій не матиме підключення, доки користувач не запустить VPN вручну. І навпаки, якщо VPN є

необов’язковим, а ініціація здійснюється вручну, то існує ризик компромісу з боку локальних зловмисників у ненадійній мережі протягом періоду, коли пристрій не підключено.

Протоколи, які найбільш широко використовуються для VPN - це TLS та IPsec. Існує безліч інших, деякі з яких (наприклад, PPTP) вийшли з використання через проблеми безпеки. Інші протоколи, такі як Wireguard, стають популярнішими і здаються багатообіцяючими, але ще не перевірені в корпоративних контекстах [12].

IP Security (IPsec)

IP Security (IPSec) забезпечує стабільну, довготривалу базу для забезпечення безпеки мережевого рівня.

IPSec підтримує всі криптографічні алгоритми, які використовуються сьогодні, а також може вмістити новіші, потужніші алгоритми, коли вони стають доступними [13].

Протоколи IPSec вирішують такі основні проблеми безпеки:

- Аутентифікація походження даних - підтверджує, що кожна дейтаграма була створена заявленим відправником.
- Цілісність даних - перевіряє, що вміст дейтаграми не було змінено під час передачі навмисно або через випадкові помилки.
- Конфіденційність даних - приховує вміст повідомлення, як правило, за допомогою шифрування.
- Захист від повтору - гарантує, що зловмисник не зможе перехопити дейтаграму та відтворити її пізніше.
- Автоматизоване керування криптографічними ключами та асоціаціями безпеки - гарантує, що ваша політика VPN може використовуватися в розширеній мережі з незначною кількістю ручної настройки або без неї [14].

Служби безпеки IPsec пропонуються за допомогою двох виділених заголовків розширення, заголовка аутентифікації (AH) і інкапсулюючого корисного навантаження безпеки (ESP) , а також за допомогою процедур і протоколів керування криптографічними ключами. Заголовок AH був розроблений для забезпечення автентичності та цілісності IP-пакета. Він також надає додаткову службу захисту від повторів. Його наявність захищає від незаконної модифікації фіксованих полів IP, підробки пакетів і, за бажанням, від повторюваних пакетів. З іншого боку, ESPheader забезпечує інкапсуляцію даних із шифруванням, щоб переконатися, що тільки вузол призначення може прочитати корисне навантаження, передане IP-пакетом. ESP може також забезпечувати цілісність і автентичність пакетів, а також службу анти-відповіді. Ці два заголовки можна використовувати окремо або об’єднати для забезпечення бажаних функцій безпеки для IP-трафіку [15].

Це можна побачити на рисунку 1.3 та рисунку 1.4

Рисунок 1.3 – Заголовок аутентифікації

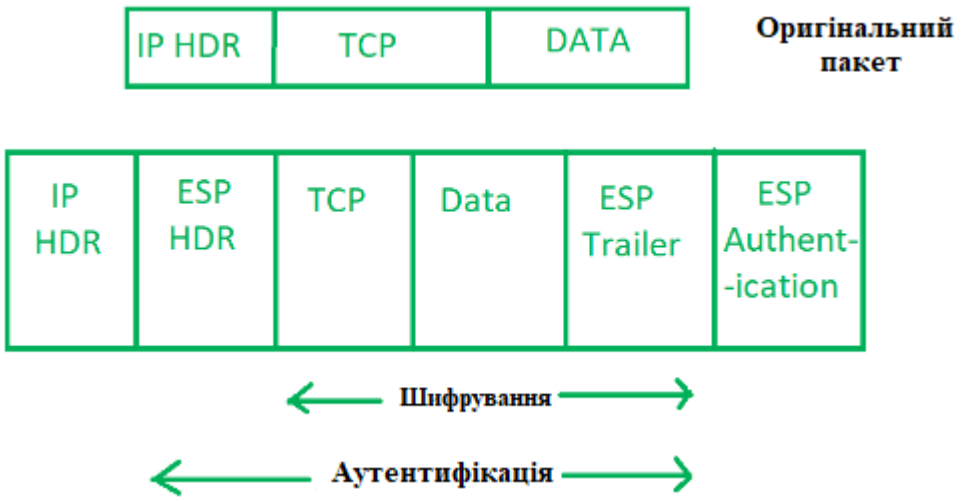

Рисунок 1.4 – Internet Key Exchange

І AH, і ESP використовують концепцію «безпекової асоціації» (SA) для узгодження алгоритмів безпеки, перетворень і параметрів, які спільно використовують відправник і одержувач захищеного потоку трафіку. Кожен IP-вузол керує набором SA, принаймні одним SA для кожного безпечного зв'язку. Зараз активні SA зберігаються в базі даних, відомій як база даних асоціації безпеки (SAD). Запис у SAD (тобто асоціація безпеки) однозначно ідентифікується трійкою, що складається з індексу параметрів безпеки (SPI), IP-адреси призначення та ідентифікатора протоколу безпеки (AH або ESP). Індекс параметрів безпеки (SPI) передається як у заголовках AH, так і в ESP, оскільки він використовується для вибору правильного SA, який буде застосовано для дешифрування та/або аутентифікації пакету. У одноадресних передачах SPI зазвичай вибирається вузлом призначення та надсилається назад відправнику, коли налаштовано зв'язок. У багатоадресних передачах SPI має бути загальним для всіх членів групи багатоадресної передачі. Кожен вузол повинен мати можливість вірно ідентифікувати правильний SA шляхом поєднання SPI з адресою багатоадресної адреси. Узгодження SA (і пов'язаного з ним SPI) є невід'ємною частиною протоколу для обміну ключами безпеки. Спеціальні вимоги безпеки визначаються на кожному вузлі, як правило, за допомогою впорядкованого списку правил (або політик), які формують безпеку вузла бази даних політики (SPD). Захист, який надається кожному вхідному/вихідному потоку трафіку, перевіряється/вирішується шляхом консультації з SPD. Загалом, пакети вибираються для одного з трьох режимів обробки на основі інформації заголовка IP та транспортного рівня, узгоджених з записами в SPD. Кожен пакет або надавав послуги безпеки IPsec, відкидаючи, або дозволяв обходити IPsec, на основі застосовних політик, знайдених у базі даних [16].

IPsec використовується для захисту конфіденційних даних, таких як фінансові транзакції, медичні записи та корпоративні комунікації, коли вони передаються по мережі. Він також використовується для захисту віртуальних приватних мереж (VPN), де тунелювання IPsec шифрує всі дані, що

надсилаються між 2 точками. IPsec може шифрувати дані прикладного рівня та забезпечувати безпеку для маршрутизаторів, які надсилають дані маршрутизації через загальнодоступний Інтернет. Також можна використовувати для аутентифікації без шифрування, наприклад, для автентифікації даних, отриманих від відомого відправника [17].

Шифрування в додатку або на транспортних рівнях моделі взаємозв'язку відкритих систем (OSI) може безпечно передавати дані без використання IPsec. На прикладному рівні шифрування виконує надійний протокол поширення гіпертексту (HTTPS). Перебуваючи на транспортному рівні, протокол безпеки транспортного рівня (TLS) забезпечує шифрування. Однак шифрування та аутентифікація на цих вищих рівнях збільшують ймовірність розкриття даних і перехоплення зловмисниками інформації протоколу [18].

IPsec аутентифікує та шифрує пакети даних, надіслані через мережі на основі IPv4 і IPv6. Заголовки протоколу IPsec знаходяться в заголовку IP пакета і визначають, як обробляються дані в пакеті, включаючи їх маршрутизацію та доставку по мережі. IPsec додає кілька компонентів до заголовка IP, включаючи інформацію безпеки та один або кілька криптографічних алгоритмів [19].

Протоколи IPsec використовують формат під назвою Запит на коментарі (RFC) для розробки вимог до стандартів безпеки мережі. Стандарти RFC використовуються в Інтернеті для надання важливої інформації, яка дозволяє користувачам і розробникам створювати, керувати та підтримувати мережу. Це продемонстровано на рисунку 1.5.

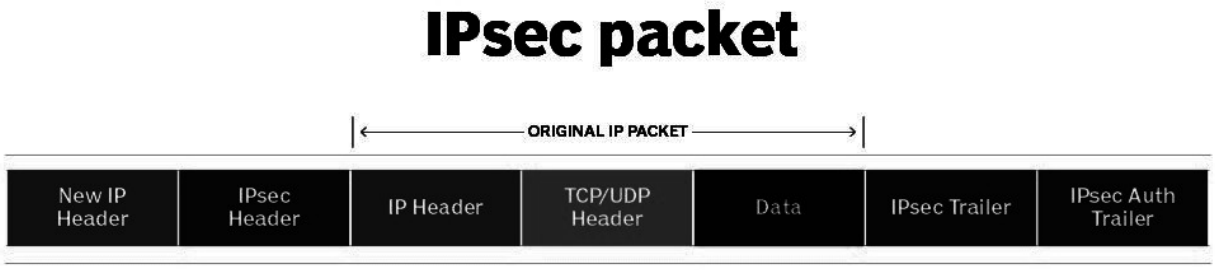

Рисунок 1.5 – Використання IPsec

Існує п’ять ключових кроків, пов’язаних із тим, як працює IPsec. Вони такі:

- Розпізнавання хоста. Процес IPsec починається, коли хост-система визнає, що пакет потребує захисту і має бути переданий за допомогою політик IPsec. Такі пакети вважаються «цікавим трафіком» для цілей IPsec, і вони запускають політики безпеки. Для вихідних пакетів це означає, що застосовуються відповідне шифрування та аутентифікація. Коли вхідний пакет визначається як цікавий, хост-система перевіряє, що він був належним чином зашифрований та автентифікований.
- Узгодження, або IKE Фаза 1. На другому кроці хости використовують IPsec для узгодження набору політик, які вони використовуватимуть для захищеного каналу. Вони також аутентифікуються один перед одним і встановлюють між собою безпечний канал, який використовується для узгодження того, як схема IPsec буде шифрувати або аутентифікувати дані, надіслані через нього. Цей процес переговорів відбувається в основному або агресивному режимі. У основному режимі хост, який ініціює сеанс, надсилає пропозиції із зазначенням бажаних алгоритмів шифрування та аутентифікації. Переговори тривають до тих пір, поки обидва хости не погодяться та не створять IKE SA, що визначає схему IPsec, яку вони використовуватимуть. Цей метод є більш безпечним, ніж агресивний режим, оскільки він створює безпечний тунель для обміну даними. В агресивному режимі хост-ініціатор не дозволяє проводити переговори і визначає IKE SA, який буде використовуватися. Прийняття хоста-відповідача автентифікує сеанс. За допомогою цього методу хости можуть швидше налаштувати ланцюг IPsec.
- Схема IPsec, або IKE Phase 2. Крок третій налаштовує ланцюг IPsec через безпечний канал, встановлений на IKE Phase 1. Хости IPsec узгоджують алгоритми, які будуть використовуватися під час поширення даних. Хости також погоджуються та обмінюються ключами кодування та декодування, які вони планують використовувати для трафіку до та з захищеної мережі. Хости

також обмінюються криптографічними одноразовими номерами, які є випадковими числами, які використовуються для аутентифікації сеансів.

- Передача IPsec. На четвертому кроці хости обмінюються фактичними даними через створений ними безпечний тунель. Налаштовані раніше IPsec SA використовуються для кодування та декодування пакетів.
- Припинення IPsec. Нарешті тунель IPsec припиняється. Зазвичай це відбувається після того, як раніше визначена кількість байтів пройшла через тунель IPsec або закінчився час очікування сеансу. Коли відбувається будь-яка з цих подій, хости спілкуються, і відбувається припинення. Після завершення роботи хости позбавляються приватних ключів, які використовуються під час трансляції даних [20].

IPsec зазвичай використовується для захисту VPN. У той час як VPN створює приватну мережу між комп'ютером користувача та сервером VPN, протоколи IPsec реалізують захищену мережу, яка захищає дані VPN від зовнішнього доступу.

Простіше кажучи, транспортний режим захищає дані під час їх передачі від одного пристрою до іншого, як правило, протягом одного сеансу. Крім того, тунельний режим захищає весь шлях даних, від точки A до точки B, незалежно від пристроїв між ними.

- Тунельний режим. Тунельний режим IPsec, який зазвичай використовується між захищеними мережевими шлюзами, дозволяє хостам за одним із шлюзів безпечно спілкуватися з хостами за іншим шлюзом. Наприклад, будь-які користувачі систем у філії підприємства можуть безпечно підключатися до будь-яких систем головного офісу, якщо філія та головний офіс мають захищені шлюзи, які діють як проксі-сервери IPsec для хостів у відповідних офісах. Тунель IPsec встановлюється між двома хостами шлюзу, але сам тунель переносить трафік з будь-яких хостів всередині захищених мереж. Тунельний режим корисний для встановлення механізму захисту всього

трафіку між двома мережами від різних хостів на обох кінцях. Це продемонстровано на рисунку 1.6.

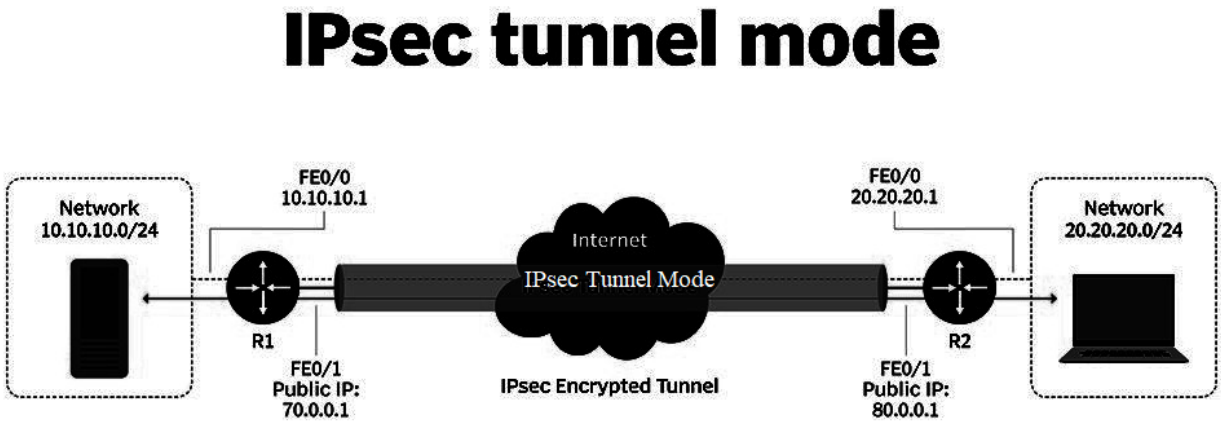

Рисунок 1.6 – IPsec tunnel mode

- Транспортний режим. Схема IPsec в транспортному режимі – це коли два хости встановлюють безпосередньо підключене IPsec VPN-з'єднання. Наприклад, цей тип ланцюга може бути налаштований, щоб дати можливість спеціалісту з підтримки віддалених інформаційних технологій (IT) увійти на віддалений сервер для виконання робіт з технічного обслуговування. Транспортний режим IPsec використовується у випадках, коли одному хосту потрібно взаємодіяти з іншим хостом. Два хости узгоджують ланцюг IPsec безпосередньо один з одним, і схема зазвичай розривається після завершення сеансу [21].

Безпека є однією з найшвидших сфер комп'ютерних мереж, оскільки вона має життєво важливе значення для захисту даних і забезпечення економічної експлуатації за допомогою електронної комерції. Безпека IP не є винятком із правила: нещодавно було визначено нові розширення для режимів автентифікації IKE та ISAKMP, які вирішують проблему захищеного віддаленого доступу та намагаються запровадити в IPsec деякі методи автентифікації на рівні користувача. Концептуальна модель мережевого середовища на основі політики безпеки визначається для IPsec разом із

протоколом, який має на меті забезпечити виявлення та вирішення політики для вузлів IPsec. До цих пір IPsec знайшов застосовність статичних мережевих конфігурацій і, загалом, мереж, які перебувають під загальним адмініструванням (VPN і внутрішньої мережі). Нові механізми, що вирішують питання управління політикою, роблять IPsec масштабованим у загальних випадках і можуть сприяти широкому розгортанню цієї технології безпеки в Інтернеті. Чиста перевага полягатиме в тому, що на рівні мережі вже буде доступний більший рівень безпеки, а отже, програми зможуть зосередитися на різних аспектах безпеки, таких як авторизація та недопущення відмови [22].

Open VPN

OpenVPN — це протокол підключення з відкритим кодом, який використовується для забезпечення безпечного тунелю між двома точками мережі. З точки зору неспеціаліста, це означає, що це надійна технологія, яка використовується багатьма віртуальними приватними мережами, щоб переконатися, що будь-які дані, надіслані через Інтернет, зашифровані та приватні. Коротше кажучи, це, можливо, найбезпечніший протокол VPN, який використовується сьогодні (хоча новий протокол Wireguard починає заперечувати це твердження) [23].

Коли ви підключаєтеся до Інтернету, особливо в загальнодоступній мережі, існує ризик передачі конфіденційних даних по мережі. Ось чому вважається найкращою практикою ніколи не входити у свій банк під час загальнодоступного Wi-Fi. З іншого боку, коли ви підключаєтеся до віртуальної приватної мережі, або «VPN», за допомогою протоколу OpenVPN, ваші дані захищені за допомогою надійного шифрування. Якщо хакер контролює вашу мережу, він не зможе пробити тунель безпеки.

Жоден інструмент або зашифроване підключення до Інтернету не може гарантувати вашу безпеку та конфіденційність, і OpenVPN нічим не

відрізняється. Однак є вагомі причини, чому це вважається одним з найбільш безпечних з’єднань, які ми розглянемо нижче [24].

OpenVPN був розроблений у 2001 році як відкритий проект, що означає, що будь-хто може використовувати та критикувати його код. Це породило спільноту програмістів, які регулярно тестують, покращують та оновлюють протокол.

Це основна перевага будь-якого програмного забезпечення з відкритим кодом. Оскільки це не запатентований код (тобто належить одній певній компанії), експерти з безпеки по всьому світу мають вільний доступ, щоб переконатися, що він підтримує безпеку. Подібно до того, як Вікіпедія отримує переваги від можливостей спільноти для перевірки фактів і виправлення будь-яких помилок, так і OpenVPN виграє від цієї спільноти спеціалістів із безпеки.

Ви можете вільно використовувати OpenVPN, оскільки він є відкритим вихідним кодом, що означає, що ви можете використовувати його, якщо дотримуєтеся умов ліцензійної угоди на програмне забезпечення. Але незважаючи на те, що код безкоштовний, варто зазначити, що він вимагає багато ручної настройки (тобто вимагає певних технічних знань). Немає безкоштовного додатка, який можна завантажити, або серверів по всьому світу, до яких ви матимете доступ.

Незважаючи на ці параметри, пересічний користувач в кінцевому підсумку використовує протокол через окремого постачальника VPN, який ліцензує програмне забезпечення та стягує з власні щомісячні платежі [25].

VPN розбиває це як два протоколи: OpenVPN UDP і OpenVPN TCP.

- OpenVPN UDP розшифровується як User Datagram Protocol (Протокол дейтаграм користувача) і включає правила, які дозволяють швидше підключитися. Найчастіше це буде з’єднання за замовчуванням просто тому, що воно забезпечить більшу швидкість Інтернету.

- OpenVPN TCP розшифровується як Transmission Control Protocol, який, як випливає з назви, підтримує більший контроль над передачею даних. Це призводить до меншої швидкості, але зазвичай є більш надійним з'єднанням.

OpenVPN використовує надійні шифри (ключі), що робить його надійним протоколом. Крім того, його можна налаштувати, щоб можна було змінювати параметри відповідно до своїх вимог – це те, що роблять багато служб VPN. Більшість експертів з безпеки вважають OpenVPN достатньо безпечним, щоб захистити від шпигунства з боку уряду [26].

Розглянемо функції, які забезпечують безпеку:

- OpenVPN адаптований: ключовою особливістю OpenVPN є те, що він дуже адаптований, і одна версія може відрізнятися від іншої. Тому він підходить для багатьох цілей. Ваш постачальник VPN може використовувати версію, відмінну від тієї, яку використовує інший постачальник.

- OpenVPN є відкритим вихідним кодом: коли програмне забезпечення є непатентованим, часто над ним працює ціла спільнота. Коли вони знаходять помилку, вони виправляють її, а також продовжують намагатися додати до неї нові функції. Це основна причина універсальності OpenVPN.

- OpenVPN підтримує кілька стандартів шифрування: OpenVPN підтримує кілька шифрів. Але як стандарт, OpenVPN реалізує 256-бітове шифрування, хоча воно не є обов'язковим (ви, можливо, бачили, що деякі постачальники VPN пропонують OpenVPN з 128-бітним шифруванням AES).

- OpenVPN є універсальним: він працює в кількох мережевих конфігураціях. Тож незалежно від того, як ваш постачальник VPN вирішить налаштувати свої сервери та підключення, OpenVPN підійде для них.

- OpenVPN не залежить від платформи: існують деякі протоколи, які залежать від пристрою. Наприклад, PPTP не працює на комп'ютерах Mac. OpenVPN, з іншого боку, може працювати на Windows, Mac, Android, iOS, Linux та інших платформах.

OpenVPN має кілька сторонніх плагінів і скриптів для покращення його функціональності. Поки Wireguard не набуде більшої популярності, OpenVPN залишатиметься стандартним безпечним протоколом підключення VPN [27].

Point-to-Point Tunneling Protocol (PPTP)

Протокол тунелювання «точка-точка» (PPTP) — це мережевий протокол, який забезпечує безпечну передачу даних від віддаленого клієнта до сервера приватного підприємства шляхом створення віртуальної приватної мережі (VPN) у мережах даних на основі TCP/IP. PPTP підтримує багатопротокольну віртуальну приватну мережу на вимогу через загальнодоступні мережі, такі як Інтернет. Мережева технологія PPTP є розширенням протоколу віддаленого доступу «точка-точка». PPTP також можна використовувати у приватних мережах LAN-to-LAN [28].

Важливою особливістю використання PPTP є його підтримка віртуальних приватних мереж за допомогою телефонних мереж загального користування (PSTN). PPTP спрощує та зменшує витрати на розгортання корпоративного рішення віддаленого доступу для віддалених або мобільних користувачів, оскільки забезпечує безпечний і зашифрований зв'язок через загальнодоступні телефонні лінії та Інтернет. PPTP усуває потребу в дорогих, виділених або приватних корпоративних серверах зв’язку, оскільки ви можете використовувати PPTP через лінії PSTN [29].

Як правило, у кожному розгортанні PPTP беруть участь три комп’ютери:

- клієнт PPTP
- сервер доступу до мережі
- сервер PPTP

Типове розгортання PPTP починається з віддаленого або мобільного клієнта PPTP, якому потрібен доступ до локальної мережі приватного підприємства за допомогою локального постачальника послуг Інтернету (ISP). Клієнти, які

використовують комп’ютери під керуванням Windows NT Server версії 4.0 або Windows NT Workstation версії 4.0, використовують Dial-up Networking і протокол віддаленого доступу PPP для підключення до провайдера [30].

Клієнт підключається до сервера доступу до мережі (NAS) на об'єкті ISP. Після підключення клієнт може надсилати та отримувати пакети через Інтернет. Сервер доступу до мережі використовує протокол TCP/IP для всього трафіку в Інтернет. Після того, як клієнт здійснив первинне з'єднання PPP з провайдером, другий дзвінок Dial-Up Networking здійснюється через існуюче з'єднання PPP. Дані, надіслані за допомогою цього другого з’єднання, мають форму IP-датаграм, які містять пакети PPP, які називаються інкапсульованими пакетами PPP. Другий виклик створює підключення до віртуальної приватної мережі (VPN) до сервера PPTP у локальній мережі приватного підприємства, це називається тунелем. Це показано на рисунку 1.7.

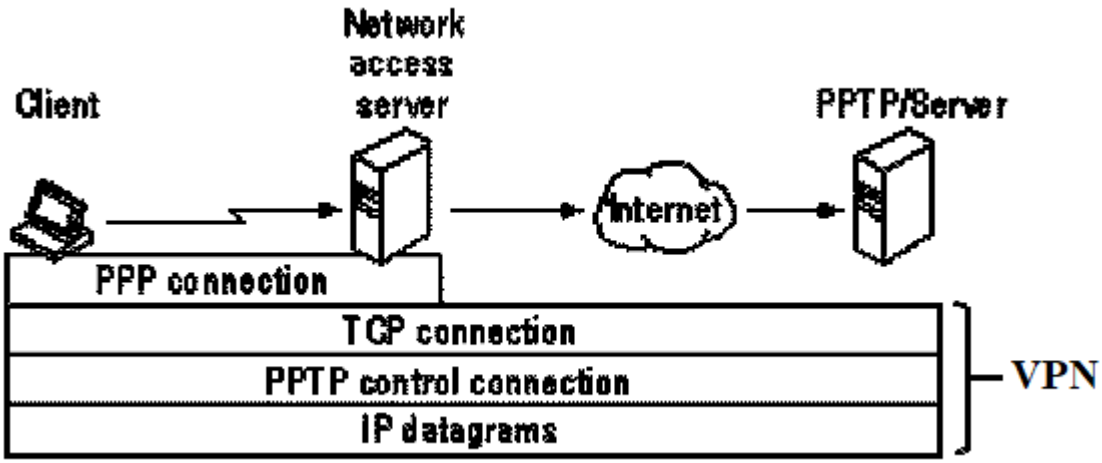

Рисунок 1.7 – Тунель PPTP

Тунелювання - це процес відправлення пакетів на комп'ютер у приватній мережі шляхом їх маршрутизації через іншу мережу, наприклад Інтернет. Інші мережеві маршрутизатори не можуть отримати доступ до комп’ютера, який знаходиться у приватній мережі. Однак тунелювання дозволяє мережі маршрутизації передавати пакет на комп’ютер-посередник, такий як сервер PPTP, який підключений як до мережі маршрутизації, так і до приватної

мережі. І клієнт PPTP, і сервер PPTP використовують тунелювання для безпечної маршрутизації пакетів на комп'ютер у приватній мережі за допомогою маршрутизаторів, які знають лише адресу сервера-посередника приватної мережі [30].

Коли сервер PPTP отримує пакет від мережі маршрутизації, він надсилає його через приватну мережу на комп'ютер призначення. Сервер PPTP робить це, обробляючи пакет PPTP, щоб отримати ім'я комп'ютера приватної мережі або інформацію про адресу в інкапсульованому пакеті PPP. Інкапсульований пакет PPP може містити багатопротокольні дані, такі як протоколи TCP/IP, IPX або NetBEUI. Оскільки сервер PPTP налаштований для зв'язку через приватну мережу за допомогою протоколів приватної мережі, він може читати багатопротокольні пакети [31].

На рисунку 1.8 проілюстровано підтримку кількох протоколів, вбудовану в PPTP. Пакет, надісланий від клієнта PPTP на сервер PPTP, проходить через тунель PPTP до кінцевого комп'ютера у приватній мережі.

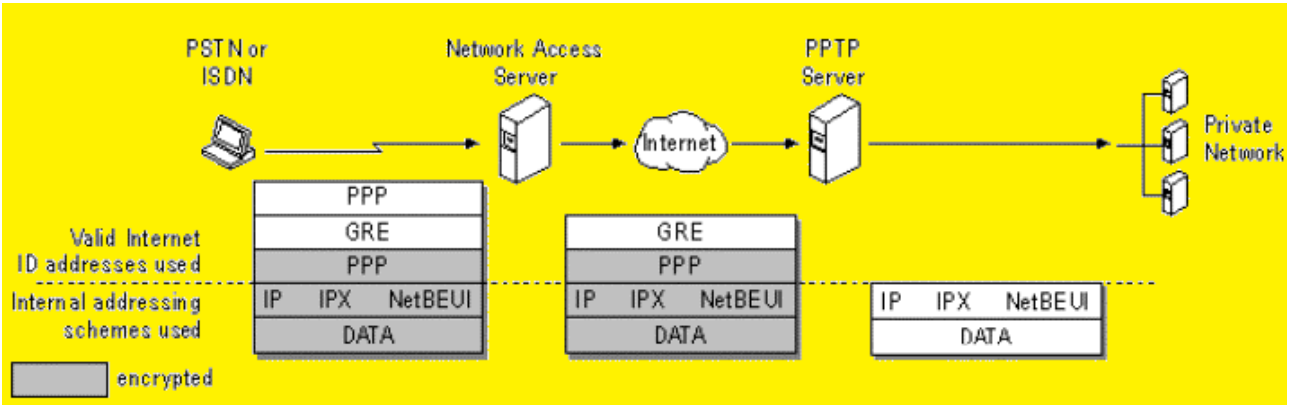

Рисунок 1.8 – Підключення клієнта PPTP до приватної мережі

PPTP інкапсулює зашифровані та стислі пакети PPP в IP-датаграми для передачі через Інтернет. Ці IP-дейтаграми маршрутизуються через Інтернет, поки не досягнуть сервера PPTP, який підключений до Інтернету та приватної мережі. Сервер PPTP розбирає дейтаграму IP на пакет PPP, а потім розшифровує пакет PPP за допомогою мережевого протоколу приватної мережі.

Комп’ютер, який підтримує мережевий протокол PPTP, наприклад, клієнт Microsoft, може підключитися до сервера PPTP двома способами:

- за допомогою сервера доступу до мережі Інтернет-провайдера, який підтримує вхідні з’єднання PPP
- за допомогою фізичного підключення до локальної мережі з підтримкою TCP/IP для підключення до сервера PPTP [31].

Клієнти PPTP, які використовують сервер доступу до мережі Інтернет-провайдера, повинні бути налаштовані з модемом і пристроєм VPN, щоб встановити окремі з’єднання з провайдером і сервером PPTP. Перше з'єднання - це комутований зв'язок за допомогою протоколу PPP через модем до постачальника послуг Інтернету. Друге з'єднання — це VPN-з'єднання за допомогою PPTP, через модем і з'єднання з провайдером, для тунелю через Інтернет до пристрою VPN на сервері PPTP. Друге з’єднання вимагає першого з’єднання, оскільки тунель між пристроями VPN встановлюється за допомогою модему та з’єднання PPP з Інтернетом.

Винятком з цієї вимоги щодо двох з'єднань є використання PPTP для створення віртуальної приватної мережі між комп'ютерами, фізично підключеними до локальної мережі приватної корпоративної мережі. У цьому сценарії клієнт PPTP вже підключений до мережі і використовує лише Dial-Up Networking з пристроєм VPN для створення підключення до сервера PPTP у локальній мережі [32].

Пакети PPTP від клієнта PPTP віддаленого доступу та клієнта PPTP локальної мережі обробляються по-різному. Пакет від клієнта віддаленого доступу розміщується на фізичному носії телекомунікаційного пристрою, а пакет від клієнта локальної мережі розміщується на фізичному носії мережевого адаптера, як показано на рисунку 1.9.

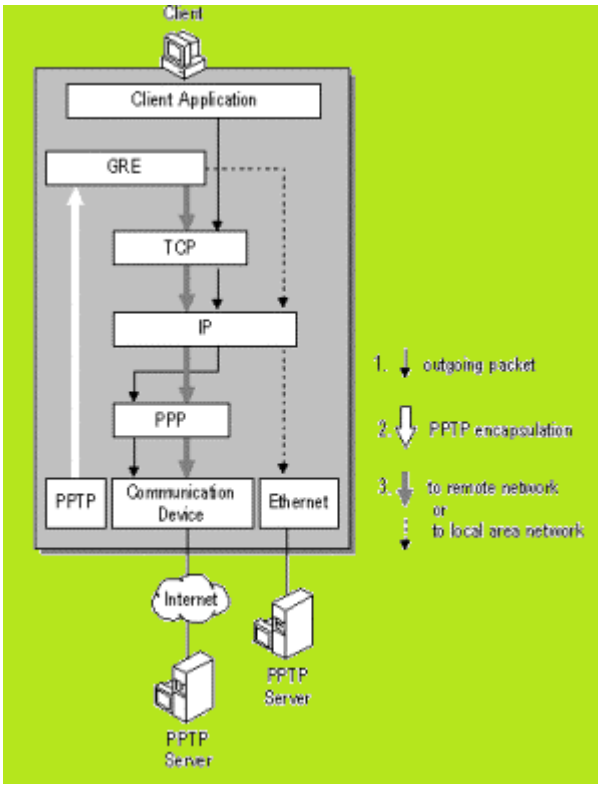

Рисунок 1.9 – Розміщення пакета PPTP на мережевому носії

Інтернет-провайдери використовують сервери доступу до мережі для підтримки клієнтів, які підключаються за допомогою протоколу, наприклад SLIP або PPP, для отримання доступу до Інтернету. Однак для підтримки клієнтів із підтримкою PPTP сервер доступу до мережі повинен надавати послугу PPP. Сервери доступу до мережі Інтернет-провайдерів розроблені та створені для розміщення великої кількості клієнтів, які підключаються до мережі [32].

Захищений зв’язок, створений за допомогою протоколу PPTP, зазвичай включає три процеси, кожен з яких вимагає успішного завершення попереднього процесу.

- PPP-з'єднання та зв'язок. Клієнт PPTP використовує PPP для підключення до провайдера за допомогою стандартної телефонної лінії або лінії ISDN. Це з’єднання використовує протокол PPP для приєднання та шифрування пакетів даних.

- Контрольне з'єднання PPTP. Використовуючи підключення до Інтернету, встановлене протоколом PPP, протокол PPTP створює контрольне з’єднання від клієнта PPTP до сервера PPTP в Інтернеті (використовує TCP для приєднання і називається тунелем PPTP).

- Тунелювання даних PPTP. Нарешті, протокол PPTP створює IP-датаграми, що містять зашифровані пакети PPP, які потім надсилаються через тунель PPTP на сервер PPTP. Сервер PPTP розбирає дейтаграми IP, розшифровує пакети PPP, а потім направляє розшифровані пакети до приватної мережі.

PPP — це протокол віддаленого доступу, який використовується PPTP для надсилання багатопротокольних даних через мережі на основі TCP/IP. PPP інкапсулює пакети IP, IPX і NetBEUI між кадрами PPP і відправляє інкапсульовані пакети, створюючи зв'язок «точка-точка» між комп'ютерами-відправником і одержувачем [33].

Більшість сеансів PPTP запускаються клієнтом, який підключається до сервера доступу до мережі Інтернет-провайдера. Протокол PPP використовується для створення комутованого з'єднання між клієнтом і сервером доступу до мережі і виконує такі три функції:

- Встановлює та припиняє фізичне з’єднання. Протокол PPP використовує послідовність, визначену в RFC 1661, для встановлення та підтримки з'єднань між віддаленими комп'ютерами.

- Аутентифікує користувачів. Клієнти PPTP аутентифікуються за допомогою протоколу PPP. Протокол PPP може використовувати автентифікацію з чистим текстом або зашифрованою Microsoft.

- Створює дейтаграми PPP, які містять зашифровані пакети IPX, NetBEUI або TCP/IP. PPP створює дейтаграми, які містять один або кілька зашифрованих пакетів даних TCP/IP, IPX або NetBEUI. Оскільки мережеві пакети зашифровані, весь трафік між клієнтом PPP і сервером доступу до мережі безпечний.

Безпека мережі від зловмисної активності може бути покращена, увімкнувши фільтрацію PPTP на сервері PPTP. Коли фільтрація PPTP увімкнена, сервер PPTP у приватній мережі приймає та маршрутизує лише пакети PPTP від автентифікованих користувачів. Це запобігає потраплянню всіх інших пакетів на сервер PPTP і приватну мережу. У поєднанні з шифруванням PPP це гарантує, що тільки авторизовані зашифровані дані входять або виходять з приватної локальної мережі.

PPTP можна використовувати з більшістю брандмауерів і маршрутизаторів, дозволяючи маршрутизувати трафік, призначений для порту 1723. Брандмауери забезпечують безпеку корпоративної мережі, суворо регулюючи дані, які надходять у приватну мережу з Інтернету. Сервер приймає пакети PPTP, передані в приватну мережу від брандмауера, і витягує пакет PPP з дейтаграми IP, розшифровує пакет і пересилає пакет на комп’ютер у приватній мережі [33].

1.3 Generic Routing Encapculation (GRE)

Тунелі загальної інкапсуляції маршрутизації (GRE) існують протягом досить тривалого часу. GRE був вперше розроблений компанією Cisco як засіб передачі інших маршрутизованих протоколів через переважно IP мережі. Деякі адміністратори мережі намагалися зменшити адміністративні витрати в ядрі своїх мереж, видаливши всі протоколи, крім IP як транспорту. Таким чином, протоколи без IP такі як IPX і AppleTalk тунелювалися через ядро IP через GRE. GRE додає новий заголовок до існуючого пакету. Ця концепція схожа на тунель IPsec режим. Оригінальний пакет передається через IP-мережу, і тільки новий зовнішній заголовок використовується для пересилання. Як тільки пакет GRE досягає кінця тунелю GRE, зовнішній заголовок видаляється, а внутрішній пакет знову відкривається. Сьогодні багатопротокольні мережі в основному зникли. Важко знайти сліди різноманітних протоколів, які раніше були у великій кількості в корпоративній та базовій інфраструктурі. В чистій IP-

мережі, GRE спочатку розглядався як марний застарілий протокол. Але зростання IPsec побачило відродження у використанні GRE в IP-мережах [34].

Початкова сила GRE полягала в тому, що в нього могло бути інкапсульовано все, що завгодно. Основне використання GRE полягало в перенесенні не IP-пакетів через IP-мережу; однак GRE також використовувався для передачі IP-пакетів через IP-хмару. Використовуючи таким чином оригінальний заголовок IP, похований всередині GRE заголовку і прихований від сторонніх очей. Загальні характеристики тунелю GRE наступні:

■ Тунель GRE схожий на тунель IPsec, оскільки вихідний пакет загорнутий у зовнішню оболонку.

■ GRE не має статусу і не пропонує механізмів контролю потоків.

■ GRE додає принаймні 24 байти накладних витрат, включаючи новий 20-байтовий IP-заголовок.

■ GRE є багатопротокольним і може тунелювати будь-який протокол рівня 3 OSI.

■ GRE дозволяє протоколам маршрутизації проходити через тунель.

■ GRE був необхідний для передачі багатоадресного IP-трафіку до випуску програмного забезпечення Cisco IOS 12.4(4)T.

■ GRE має відносно слабкі функції безпеки [34].

GRE використовується для інкапсуляції протоколу довільного рівня над іншим протоколом довільного рівня. Загалом, GRE дозволяє створити тунель за допомогою певного протоколу, який потім приховує вміст іншого, що передається всередині тунелю. Тунелювання забезпечує механізм транспортування пакетів одного протоколу в рамках іншого протоколу. Протокол, який перевозиться, називається пасажирським протоколом, а протокол, який використовується для перевезення пасажирського, називається транспортним протоколом. Generic Routing Encapsulation (GRE) є одним із доступних механізмів тунелювання, який використовує IP як транспортний протокол і може використовуватися для передачі багатьох різних протоколів

пасажирів. Тунелі поводяться як віртуальні прямі посилання, які мають дві кінцеві точки, визначені адресами джерела тунелю та адресами призначення тунелю на кожній кінцевій точці.

Структуру заголовка пакета GRE продемонстровано на рисунку 1.10.

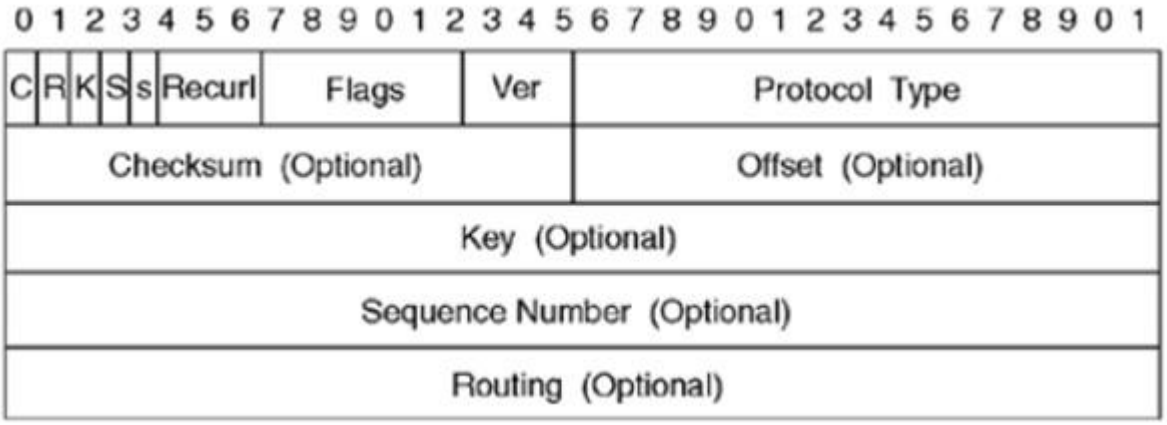

Рисунок 1.10 – Заголовок пакета GRE

Сам заголовок GRE містить 4 байти, які представляють мінімальний розмір заголовка GRE без додаткових опцій. Перша пара байтів (біти від 0 до 15) містить прапори, які вказують на наявність опцій GRE. Такі параметри, якщо вони активні, додають додаткові витрати до заголовка GRE. Друга пара байтів є полем протоколу і вказує тип даних, які передаються в GRE тунель. Параметр Присутня контрольна сума (біт 0) додає до заголовка GRE додаткове 4-байтове поле контрольної суми. Ця контрольна сума з’являється після поля протоколу в заголовку GRE, лише якщо контрольна сума присутня біт встановлений. Зазвичай цей параметр не потрібен, оскільки інші протоколи верхнього рівня надають можливості контрольної суми для виявлення пошкодження пакетів. Параметр Key Present (біт 2) додає в заголовок GRE додаткове 4-байтове ключове поле. Цей чіткий текст ключ знаходиться після поля контрольної суми. Ключ використовується для забезпечення базової аутентифікації, де кожен GRE кінцевої точки має ключ. Однак сам ключ відкривається в заголовку GRE. Через це вразливості, шифрування GRE зазвичай не використовується. Однак значення ключа можна використовувати для однозначного ідентифікування

кількох тунелів між двома кінцевими точками. Це буде схоже на IPsec SPI. Параметр Sequence Number (біт 3) додає необов'язкове поле порядкового номера з 4 байтами до GRE заголовку. Це значення послідовності відповідає ключовому параметру. Цей параметр використовується для правильної послідовності GRE пакету після прибуття. Як і параметр контрольної суми, він зазвичай не використовується, оскільки верхній рівень протоколу також пропонує цю функцію. Біти 13–15 вказують номер версії GRE. 0 представляє базовий GRE, а 1 показує, що використовується протокол тунелювання «точка-точка» (PPTP). Другі 2 байти заголовка GRE представляють поле Protocol. Ці 16 біт ідентифікують тип пакету, який передається всередині тунелю GRE. Ethertype 0x0800 вказує на IP. Це продемонстровано в таблиці 1.2.

Таблиця 1.1 Параметри заголовка GRE

Біт заголовка GRE	Варіант	Опис
0	Присутня контрольна сума	Додає 4-байтове поле контрольної суми до заголовка GRE після поля протоколу, якщо цей біт встановлений на 1
2	Key Present	Додає 4-байтовий ключ шифрування до заголовка GRE після поля контрольної суми, якщо цей біт встановлений на 1.
3	Sequence Number	Додає 4-байтовий порядковий номер до заголовка GRE після ключового поля, якщо цей біт встановлений на 1
13-15	Версія GRE	0 вказує на базовий GRE, тоді як 1 використовується для PPTP

GRE дозволяє використовувати протоколи, які зазвичай не підтримуються мережею, оскільки пакети загорнуті в інші , які використовують підтримувані протоколи [35].

Наприклад, припустимо, що компанії потрібно встановити з’єднання між локальними мережами (LAN) у двох різних офісах. Обидві локальні мережі використовують останню версію Інтернет-протоколу IPv6. Але для того, щоб потрапити з однієї офісної мережі в іншу, трафік повинен проходити через мережу, керовану третьою стороною, яка є дещо застарілою і підтримує лише старіший протокол IPv4. За допомогою GRE компанія може надсилати трафік через цю мережу, інкапсулюючи пакети IPv6 в пакети IPv4.

Інкапсуляція пакетів в інші пакети називається «тунелюванням». Тунелі GRE зазвичай налаштовуються між двома маршрутизаторами, причому кожен маршрутизатор діє як один кінець тунелю. Маршрутизатори налаштовані на надсилання та отримання пакетів GRE безпосередньо один одному. Будь-які маршрутизатори між цими двома не відкриють інкапсульовані пакети; вони лише посилаються на заголовки, що оточують інкапсульовані пакети, щоб переслати їх [36].

Але GRE створює віртуальний «тунель» через «гірську» мережу, щоб пропустити пакети даних. Подібно до того, як тунель створює шлях для автомобілів прямо через сушу, GRE (та інші протоколи тунелювання) створюють шлях для пакетів даних через мережу, яка їх не підтримує.

Усі дані, що надсилаються по мережі, розбиваються на менші частини, які називаються пакетами, і всі пакети мають дві частини: корисне навантаження та заголовок. Корисне навантаження — це фактичний вміст пакета, дані, що надсилаються. Заголовок містить інформацію про те, звідки надходить пакет і до якої групи пакетів він належить. Кожен мережевий протокол додає заголовок до кожного пакету [36].

GRE додає два заголовки до кожного пакету: заголовок GRE, який має довжину 4 байти, і заголовок IP, довжина якого 20 байтів. Заголовок GRE вказує тип протоколу, який використовується інкапсульованим пакетом. IP-заголовок інкапсулює заголовок і корисне навантаження вихідного пакета. Це означає, що пакет GRE зазвичай має два IP-заголовки: один для вихідного

пакета, а другий додається протоколом GRE. Лише маршрутизатори на кожному кінці тунелю GRE посилатимуться на оригінальний заголовок IP, що не є GRE.

Протокол GRE має ряд переваг, зокрема:

- Використання кількох протоколів через одну протокольну магістраль
- Надання обхідних шляхів для мереж з обмеженими стрибками
- Підключення несуміжних підмереж
- Менш вимогливий до ресурсів, ніж його альтернативи (наприклад, IPsec VPN)

GRE подібний до багатогранної служби доставки, яка може обробляти будь-які упаковки — від нейлонового мішка, наповненого водою, до репліки T-rex у натуральну величину — транспортуючи її від дверей до дверей. Він просто бере все, що ви йому даєте, кладе в одну з власних коробок, наклеює на нього адресну етикетку і доставляє туди, куди потрібно [37].

По суті, GRE робить те саме в процесі, який називається інкапсуляцією, приймає пакет даних, що надсилається, і «перепаковує» його. Це досягається шляхом додавання двох додаткових заголовків, один з яких ідентифікує його як пакет GRE, а другий – для надання нових IP-адрес джерела та призначення.

Для передачі інкапсульованого пакета необхідно встановити тунель GRE. Це віртуальне з’єднання «точка-точка» між двома мережами — безпечний прохід, який забезпечує пряму доставку «від дверей до дверей» «без запитань».

При наявності тунелю пакет GRE може переміщатися безпосередньо між двома своїми кінцевими точками. Те, що тунель є віртуальним, означає, що, навіть коли пакет проходить через інші маршрутизатори, немає взаємодії з його корисним навантаженням. Замість цього пакет просувається вперед, доки не досягне кінцевої точки тунелю, де зовнішній пакет деінкапсулюється, а корисне навантаження аналізується.

Тунель GRE встановлюється на рівні маршрутизатора і відрізняється залежно від типу обладнання або послуги, яку ви використовуєте. Зазвичай вам

потрібно буде налаштувати IP-адреси інтерфейсу тунелю та надати загальнодоступні IP-адреси для обох кінців тунелю GRE [37].

Ни рисунку 1.11 продемонстровано приклад тунелю, налаштованого між двома маршрутизаторами Cisco.

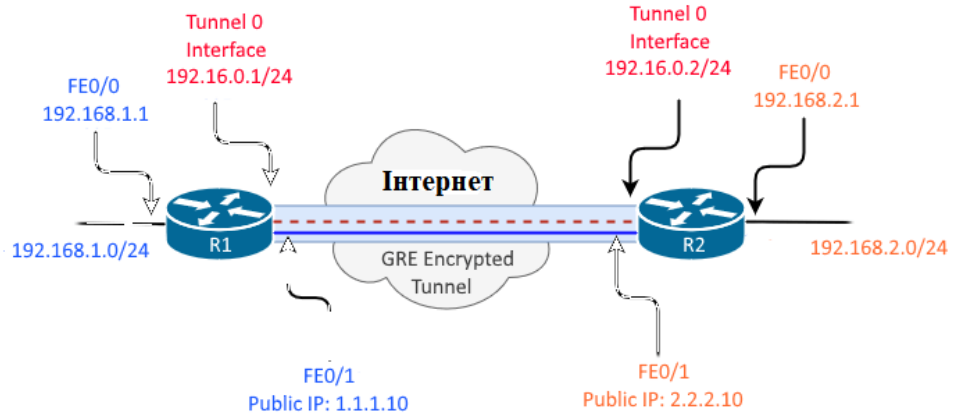

Рисунок 1.11 – Тунель GRE між маршрутизаторами

У більшості випадків при налаштуванні тунелю GRE, треба торкнутися налаштування маршрутизатора. Але для початку знадобляться дві частини інформації. Тунелі GRE покладаються на:

- IP-адреса відправника. Це не те саме, що зручна для людини веб-адреса (наприклад, www.starosta.com). IP-адреса складається з цифр і десяткових знаків.
- IP-адреса одержувача. Це означає, що знадобляться ті самі дані для вузла, до якого потрібно підключитися.

Пакетних правил багато і вони широко розповсюджені. Іноді впровадження тунелю GRE означає порушення деяких з цих протоколів. Наприклад, максимальна одиниця передачі (MTU) встановлюється на рівні пристрою, і вона визначає, наскільки великим може бути пакет. Максимальний розмір сегмента (MSS) вимірює розмір корисного навантаження.

Перейшовши до процесу інкапсуляції GRE, і змінюється розмір свого корисного навантаження. Але потім додається два заголовки, і це може

означати, що порушуються правила MTU. Чисті, переупаковані пакети можуть бути розбиті на частини, перш ніж вони будуть доставлені [38].

Протокол GRE дозволяє одному пристрою спілкуватися з іншим, минаючи відкриті системи фільтрації. Хакери люблять такі установки. Трохи попрацювавши, вони можуть налаштувати розподілені атаки відмови в обслуговуванні (DDoS), які призводять до збою ваших серверів і виводять користувача у автономний режим.

GRE можна використовувати для здійснення DDoS-атак, як і будь-який мережевий протокол. Одна з найбільших DDoS-атак зафіксована у вересні 2016 року. Вона була спрямована проти веб-сайту дослідника безпеки та здійснена за допомогою ботнету Mirai. Веб-сайт був переповнений пакетами, які використовували протокол GRE.

Під час атаки розподіленої відмови в обслуговуванні (DDoS) зловмисник намагається переповнити цільовий сервер або мережу небажаним мережевим трафіком — так само, як бомбардування ресторану фальшивими замовленнями на доставку, поки він не зможе надати послуги законним клієнтам.

Техніки маскування недоступні для тунелів GRE. Хакери повинні розкрити свої вихідні IP-адреси. Будь-який спеціаліст із безпеки, побачивши стільки трафіку, що надходить з одного джерела, просто заблокував би цю адресу від надсилання чогось іншого.Хакери обходять цю проблему, розгортаючи ботнети. Сотні тисяч заражених пристроїв, які зазвичай підключаються через покупку в Інтернеті, спрямовують свої пакети на ціль. Такий напад є масовим, і його дуже важко зупинити.

Підходи до захисту від DDoS-атаки, що випливають із GRE, такі ж, як і для будь-якої іншої атаки через ботнет. По суті, робота полягає в тому, щоб або збільшити свої можливості для обробки додаткового трафіку, або зменшити доступ, щоб зменшити кількість пакетів, які можуть надходити до юзера [38].

1.4 Вибір способів для реалізації

У наш час в інтернеті з’являється все більше сервісів та програмного забезпечення. Зараз мільйони комп’ютерів підключені до Інтернету, він перетворився на основний засіб комунікації і дозволяє користувачам взаємодіяти, не звертаючи уваги на відстань чи місцезнаходження. З Інтернетом асоціюється набір технологій, починаючи від мережевих протоколів і закінчуючи браузерами, які були розроблені для підтримки інтернет операцій. З огляду на це, в світі програмного забезпечення відбувається неймовірний розвиток веб-додатків різного типу [39].

Прості додатки мають не таке гарне становище, звичайно вони також мають розвиток, але не такий великий як веб-технології. Багато фірм використовують веб-додатки, в яких вбачають велику перспективу і застосовують їх замість звичайних додатків. Це також сприяє інвестиціям великих грошей, що виводить технології на більш високий рівень. Саме тому було прийнято рішення виконати поставлене завдання у вигляді веб-додатку, а саме графічного інтерфейсу.

Мова розмітки гіпертексту (HTML) – це мова комп’ютера, яка складається з більшості веб-сторінок та онлайн-додатків. Гіпертекст — це текст, який використовується для посилань на інші фрагменти тексту, тоді як мова розмітки — це серія маркування, яка повідомляє веб-серверам стиль і структуру документа.HTML не вважається мовою програмування, оскільки він не може створювати динамічні функції. Натомість за допомогою HTML веб-користувачі можуть створювати та структурувати розділи, абзаци та посилання за допомогою елементів, тегів та атрибутів [39].

CSS означає каскадні таблиці стилів. Коротше кажучи, CSS – це мова дизайну, яка робить веб-сайт привабливішим, ніж просто звичайний або ненадихаючий фрагмент тексту. Тоді як HTML значною мірою визначає

текстовий вміст, CSS визначає візуальну структуру, макет та естетику. HTML — це мова розмітки, а CSS — мова таблиць стилів.

JavaScript — це текстова мова програмування, яка використовується як на стороні клієнта, так і на стороні сервера, що дозволяє зробити веб-сторінки інтерактивними. Якщо HTML і CSS є мовами, які надають структуру і стиль веб-сторінкам, JavaScript надає веб-сторінкам інтерактивні елементи, які залучають користувача. Включення JavaScript покращує роботу веб-сторінки, перетворюючи її зі статичної сторінки в інтерактивну. JavaScript в основному використовується для веб-додатків і веб-браузерів. Але JavaScript також використовується за межами Інтернету в програмному забезпеченні, серверах та вбудованих апаратних елементах керування [40].

Ось деякі основні речі, для яких використовується JavaScript:

- Додавання інтерактивної поведінки на веб-сторінки. JavaScript дозволяє користувачам взаємодіяти з веб-сторінками. Практично немає обмежень на те, що ви можете робити за допомогою JavaScript на веб-сторінці.
- Створення веб- та мобільних додатків. Розробники можуть використовувати різні фреймворки JavaScript для розробки та створення веб- та мобільних додатків. Фреймворки JavaScript — це колекції бібліотек коду JavaScript, які надають розробникам попередньо написаний код для використання рутинних функцій і завдань програмування — буквально фреймворк для створення веб-сайтів або веб-додатків.
- Створення веб-серверів і розробка серверних додатків. Крім веб-сайтів і програм, розробники також можуть використовувати JavaScript для створення простих веб-серверів і розвитку внутрішньої інфраструктури за допомогою Node.js.
- Розробка ігор. Звичайно, також можна використовувати JavaScript для створення браузерних ігор. Це чудовий спосіб для початківців розробників практикувати свої навички JavaScript.

Vuetify — це повна структура інтерфейсу користувача, побудована на основі Vue.js. Мета проекту — надати розробникам інструменти, необхідні для створення багатих і привабливих користувацьких можливостей. На відміну від інших фреймворків, Vuetify розроблено з нуля так, щоб його було легко навчитися і опанувати завдяки сотням ретельно розроблених компонентів зі специфікації Material Design. Vuetify використовує мобільний підхід до дизайну, що означає, що ваша програма працює «з коробки» — на телефоні, планшеті чи настільному комп’ютері [40].

Для створення графічного інтерфейсу було застосовано HTML, CSS, JavaScript, тому що ці мови програмування є одними з найпопулярніших і неважкі у вивченні. А також використовується бібліотека Vuetify.

1.5 Постановка задачі

Розглянувши та проаналізувавши літературні довідники, мету даної роботи можна сформулювати так: треба створити графічний інтерфейс у вигляді веб-сторінки, який буде допомагати налаштовувати роутери та динамічну для роботи GRE мережі автоматично. При розробці згенерований код повинен легко переноситись у симулятор Cisco Packet Tracer та справжнє устаткування. Інтерфейс мусить бути зручним та зрозумілим для будь-яких юзерів, які навіть уявлення не мають як працювати с подібними додатками. Треба створити веб-сторінку, на якій вводяться початкові дані, далі генеруються налаштування, які потім копіюються та вносяться в симулятор. Це дозволить налаштувати маршрутизатор і в результаті саму мережу.

Постановка задачі:

1. Створення мережі в симуляторі Cisco Packet Tracer
2. Розробка графічного інтерфейсу
3. Перевірка та тестування веб-сторінки на справжньому устаткуванні та в симуляторі.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

2.1 Побудова віртуальної GRE мережі

Оскільки «епоха великих даних» набирає обертів, віртуалізація стає все більш ключовим інструментом, щоб зрозуміти трильйони рядків даних, які генеруються щодня. Технологія має не тільки промислове застосування, але й використовується для того, щоб вивчати та тестувати різні види програмного забезпечення.

У наш час існує багато інструментів для моделювання, тестування та діагностики комп’ютерних систем. Для того щоб не проводити експерименти на реальному обладнанні, використовують програмне забезпечення для мережевого моделювання. Наприклад, таким видом ПО є Cisco Packet Tracer.

Cisco Packet Tracer — це програмне забезпечення Cisco для моделювання. Його можна використовувати для створення складних мережевих типологій, а також для тестування та моделювання абстрактних мережевих концепцій. Він діє як ігровий майданчик для вивчення мереж, і дуже близький до того, що ви бачите в комп’ютерних мережах.

Особливості Cisco Packet Tracer:

- Cisco Packet Tracer підтримує багатокористувацьку систему, яка дозволяє багатьом користувачам підключати різні топології через комп'ютерну мережу. Інструктори також можуть створювати вправи для виконання студентами за допомогою Packet Tracer.
- Підтримує розширення функцій за допомогою додаткових програм, які використовують API для покращення можливостей Cisco Packet Tracer у таких областях, як навчальні програми та оцінювання, ігри, доступність та взаємодія з реальним обладнанням.
- Розширений фізичний режим переносить вас у віртуальну лабораторію, де ви можете імітувати кабельні пристрої на стійці. Оновлення ключових

навичок, таких як розміщення пристроїв (стійка й стек), перемикання живлення на пристрої, підключення кабелю від порту до порту пристрою (включаючи вибір кабелю та керування ним), усунення несправностей тощо.

- Його можна безкоштовно завантажити через обліковий запис Netacad.
- Дає змогу користувачам моделювати конфігурацію маршрутизаторів Cisco, і доступ до нього можна отримати з будь-якого місця в будь-який час.
- Мережевий контролер дає вам централізовану інформаційну панель, щоб бачити стан мережі, миттєво виявляти й діагностувати проблеми, а також направляти зміни конфігурації на всі керовані пристрої одночасно, незалежно від того, використовуєте ви його веб-інтерфейс або API. Ви також можете використовувати реальні програми на своєму комп'ютері для доступу до мережевого контролера та запуску власних сценаріїв автоматизації інфраструктури.
- До нього можна отримати доступ через необмежену кількість пристроїв.
- Забезпечує інтерактивне середовище, що розвивається самостійно [40].

Тому проаналізувавши всі плюси цього ПО, для побудови GRE мережі було вибрано Cisco Packet Tracer.

Спочатку було створено модель GRE мережі. Також проведено налаштування інтерфейсів роутерів, світчів та динамічної маршрутизації. При створенні моделі було використано роутери 2911 та світчі 3560-24PS.

По-перше утворюємо локальні мережі (рис. 2.1).

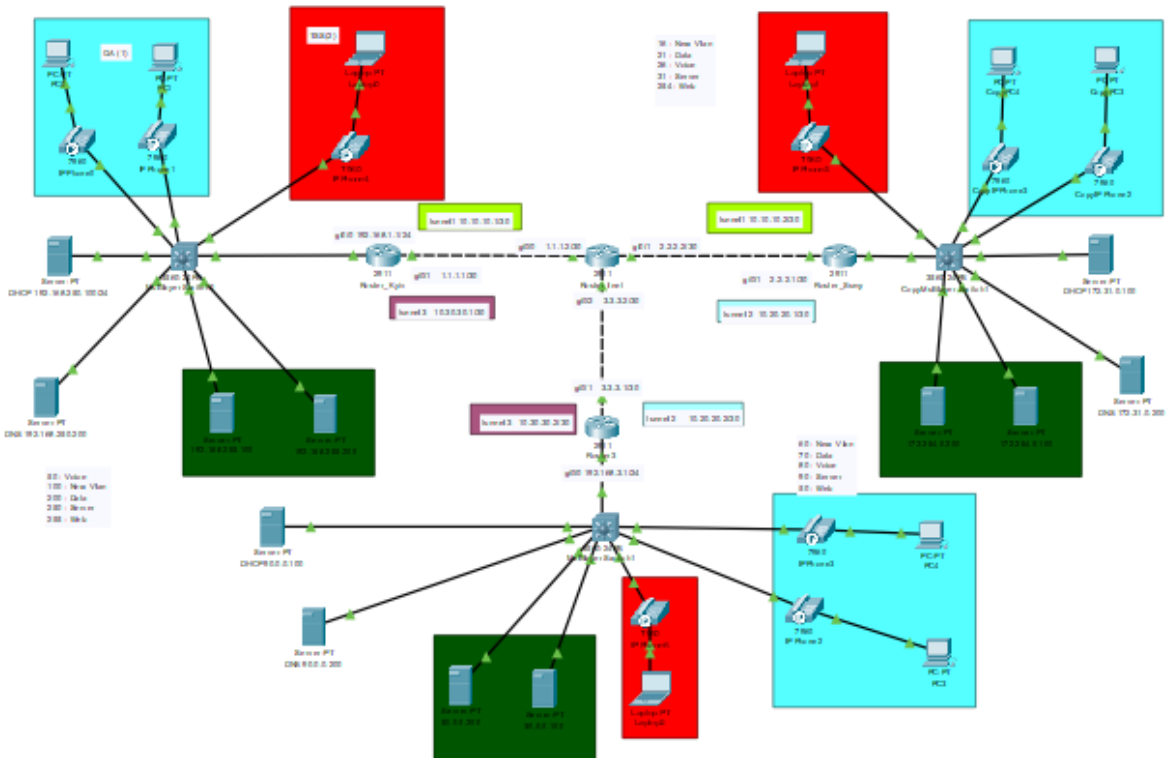

Рисунок 2.1 – Побудована GRE мережа

Спочатку налаштовуємо IP конфігурацію на ПК та ноутбуці першого офіса (рис. 2.2).

Рисунок 2.2 – Приклад IP-конфігурації

Потім на роутері “Kyiv” задаємо IP-адресу портів GigabitEthernet 0/0 та GigabitEthernet 0/1 (рис. 2.3).

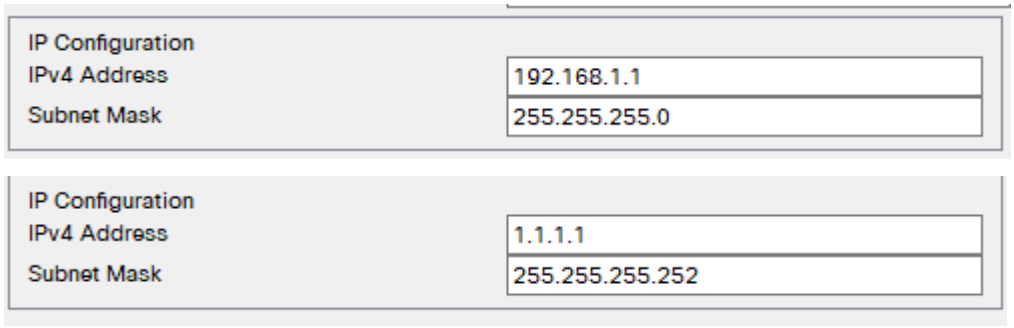

Рисунок 2.3 – Конфігурації GigabitEthernet 0/0 та 0/1 на роутері “Kyiv”

Далі налаштовуємо VoIP в мережі з vlan, та DHCP сервер. Для цього проводимо конфігурацію vlan на роутері “Kyiv” (рис. 2.4).

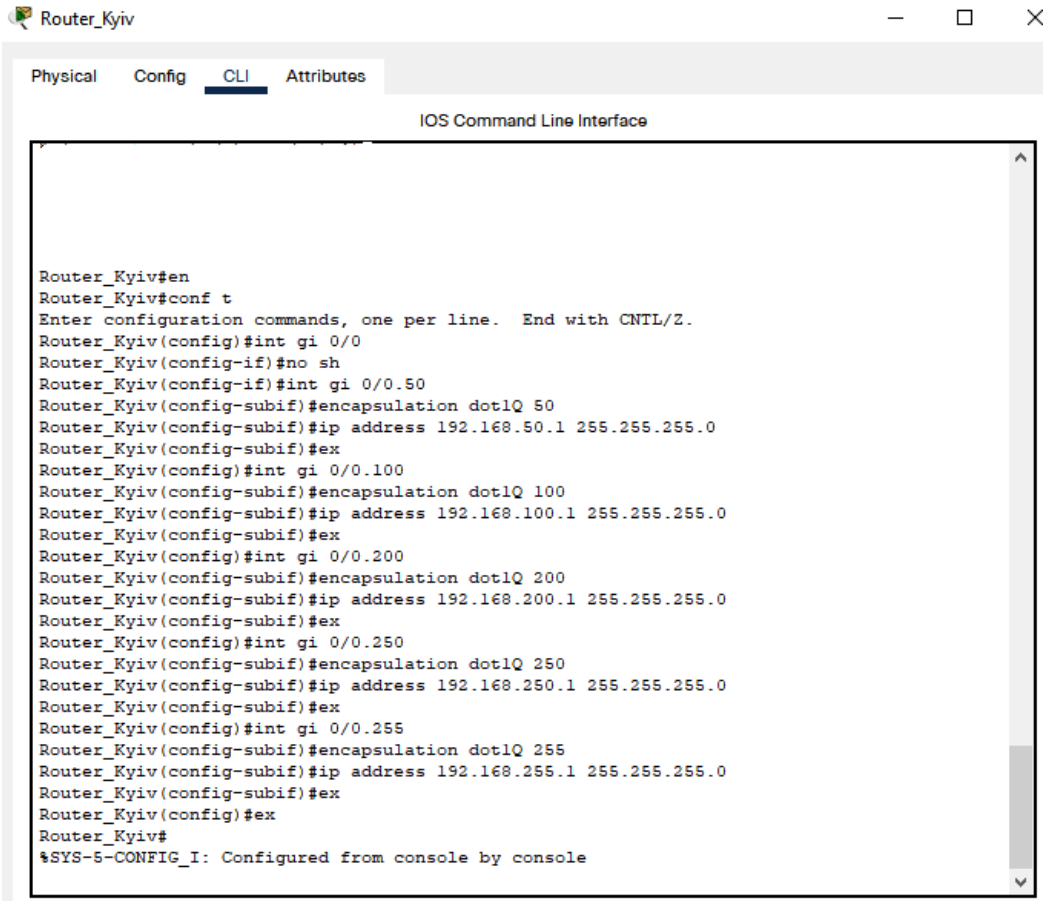

Рисунок 2.4 – Налаштування vlan на роутері “Kyiv”

Наступним кроком проводимо налаштування світчу для кожного vlan. Задаємо світчу vlan та вказуємо діапазон портів вільного доступу. Після цього налаштовуємо транковий порт від світча до маршрутизатора та приписуємо порт fa 0/5 до 250 vlan (для того щоб передавати данні між роутером та DHCP сервером) (рис. 2.5).

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int gi 0/1
Switch(config-if)#switchport trunk encapsulation dotlq
Switch(config-if)#switchport mode trunk
Switch(config-if)#ex
Switch(config)#int fa 0/5
Switch(config-if)#switchport access vlan 250
Switch(config-if)#ex
```

Рисунок 2.5 – Налаштування на Switch

Далі через екранні форми задаємо налаштування DHCP сервера та налаштовуємо його. Створюємо ще три пули адрес, що динамічно роздаються, для мереж 50,100 та 200 (рис. 2.6).

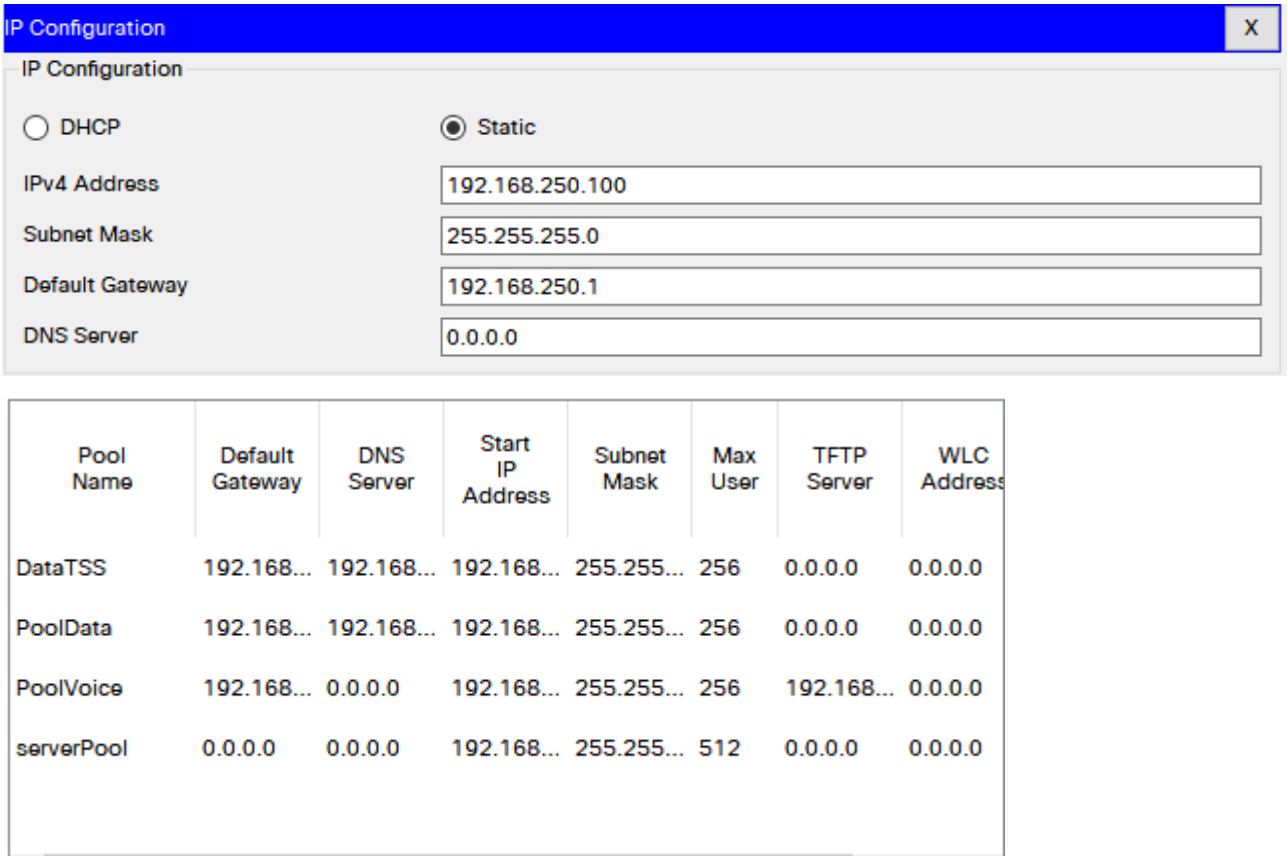

Рисунок 2.6 – Налаштування DHCP сервера

Далі тільки приписуючи на Switch той чи інший порт до відповідної VLAN автоматично можемо роздавати потрібні ip адреси. Наступним кроком налаштовуємо телефонний сервіс та задаємо номери (рис. 2.7).

```
Router_Kyiv>en
Router_Kyiv#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router_Kyiv(config)#ephone-dn 1
Router_Kyiv(config-ephone-dn)#number 10011
Router_Kyiv(config-ephone-dn)#ex
Router_Kyiv(config)#ephone-dn 2
Router_Kyiv(config-ephone-dn)#number 10012
Router_Kyiv(config-ephone-dn)#ex
Router_Kyiv(config)#ephone-dn 3
Router_Kyiv(config-ephone-dn)#number 10013
```

Рисунок 2.7 – Налаштування телефонії

Переходимо до DNS сервера та Web серверів. Спочатку проводимо налаштування на веб сервері. Для цього запускаємо конфігурацію HTTP та додаємо текстовий зміст сторінок, який буде відображатися в емуляторі браузера комп'ютера (рис. 2.8).

Рисунок 2.8 – Налаштування веб сервера

Далі переходимо до DNS сервера та задаємо доменні імена веб серверів (рис. 2.9).

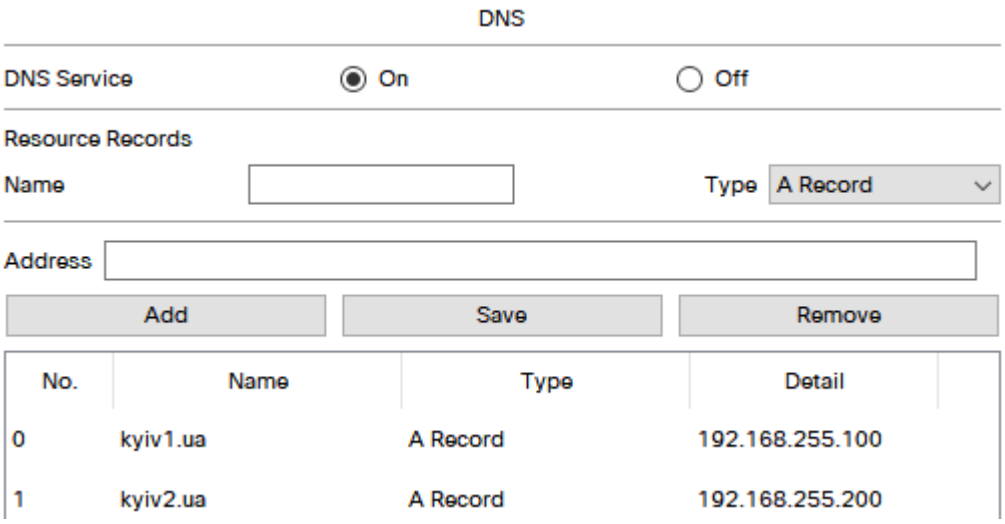

Рисунок 2.9 – Налаштування DNS сервера

Налаштування першого офісу можна вважати завершеним. Далі проводимо такі ж конфігурації для двох інших та переходимо до налаштування GRE тунелю.

Спочатку на Router_Kyiv створюємо тунель 11 та тунель 13 між роутерами Київ-Суми та Київ-Одеса та задаємо незнайомі мережі (рис. 2.10).

```
Router_Kyiv>en
Router_Kyiv#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router_Kyiv(config)#interface Tunnel 11

Router_Kyiv(config-if)#
%LINK-5-CHANGED: Interface Tunnel11, changed state to up

Router_Kyiv(config-if)#ip address 10.10.10.1 255.255.255.0
% 10.10.10.0 overlaps with Tunnel1
Router_Kyiv(config-if)#tunnel source GigabitEthernet 0/1
Router_Kyiv(config-if)#tunnel destination 2.2.2.1
Router_Kyiv(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel11, changed state to up

Router_Kyiv(config-if)#end
```

```
Router_Kyiv#en
Router_Kyiv#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router_Kyiv(config)#interface Tunnel 13

Router_Kyiv(config-if)#
%LINK-5-CHANGED: Interface Tunnel13, changed state to up

Router_Kyiv(config-if)#ip address 10.30.30.1 255.255.255.0
% 10.30.30.0 overlaps with Tunnel3
Router_Kyiv(config-if)#tunnel source GigabitEthernet 0/1
Router_Kyiv(config-if)#tunnel destination 3.3.3.1
Router_Kyiv(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel13, changed state to up
ex
Router_Kyiv(config)#ip route 192.168.2.0 255.255.255.0 10.10.10.2
Router_Kyiv(config)#ip route 192.168.3.0 255.255.255.0 10.30.30.2
Router_Kyiv(config)#ex
```

Рисунок 2.10 – Створення GRE тунелю 11 та 13

Потім подібні налаштування робимо для створення тунелів 11 та 12 на Router_Sumy між роутерами Суми-Київ та Суми-Одеса (рис. 2.11).

```
Router_Sumy>en
Router_Sumy#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router_Sumy(config)#interface Tunnel 11

Router_Sumy(config-if)#
%LINK-5-CHANGED: Interface Tunnel11, changed state to up

Router_Sumy(config-if)#ip address 10.10.10.2 255.255.255.0
% 10.10.10.0 overlaps with Tunnel1
Router_Sumy(config-if)#tunnel source GigabitEthernet 0/1
Router_Sumy(config-if)#tunnel destination 1.1.1.1
Router_Sumy(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel11, changed state to up
end
```

```
Router_Sumy#
%SYS-5-CONFIG_I: Configured from console by console
conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router_Sumy(config)#interface Tunnel 12

Router_Sumy(config-if)#
%LINK-5-CHANGED: Interface Tunnel12, changed state to up
ip address 10.20.20.1 255.255.255.0
% 10.20.20.0 overlaps with Tunnel2
Router_Sumy(config-if)#tunnel source GigabitEthernet 0/1
Router_Sumy(config-if)#tunnel destination 3.3.3.1
Router_Sumy(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel12, changed state to up
end
Router_Sumy#
%SYS-5-CONFIG_I: Configured from console by console
conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router_Sumy(config)#ip route 192.168.1.0 255.255.255.0 10.10.10.1
Router_Sumy(config)#ip route 192.168.3.0 255.255.255.0 10.20.20.2
Router_Sumy(config)#ex
```

Рисунок 2.11 – Створення GRE тунелю 11 та 12

І останнім етапом створюємо тунелі 12 та 13 на Router_Odesa між роутерами Одеса-Суми та Одеса-Київ (рис. 2.12).

```
Router_Odesa>en
Router_Odesa#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router_Odesa(config)#interface Tunnel 12

Router_Odesa(config-if)#
%LINK-5-CHANGED: Interface Tunnel12, changed state to up
ip address 10.20.20.2 255.255.255.0
% 10.20.20.0 overlaps with Tunnel2
Router_Odesa(config-if)#tunnel source GigabitEthernet 0/1
Router_Odesa(config-if)#tunnel destination 2.2.2.1
Router_Odesa(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel12, changed state to up
end

Router_Odesa#
%SYS-5-CONFIG_I: Configured from console by console
conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router_Odesa(config)#interface Tunnel 13

Router_Odesa(config-if)#
%LINK-5-CHANGED: Interface Tunnel13, changed state to up

Router_Odesa(config-if)#ip address 10.30.30.2 255.255.255.0
% 10.30.30.0 overlaps with Tunnel3
Router_Odesa(config-if)#tunnel source gi 0/1
Router_Odesa(config-if)#tunnel destination 1.1.1.1
Router_Odesa(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel13, changed state to up
end
Router_Odesa#
%SYS-5-CONFIG_I: Configured from console by console
conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router_Odesa(config)#ip route 192.168.1.0 255.255.255.0 10.10.10.1
Router_Odesa(config)#ip route 192.168.2.0 255.255.255.0 10.20.20.1
Router_Odesa(config)#ex
```

Рисунок 2.12 – Створення GRE тунелю 12 та 13

Після цього перевіряємо працездатність віртуальної мережі. Спочатку перевіряємо правильність пінгування між комп’ютерами на прикладі першого офісу (рис. 2.13).

```
C:\>ping 70.0.0.1

Pinging 70.0.0.1 with 32 bytes of data:

Reply from 70.0.0.1: bytes=32 time<1ms TTL=254
Reply from 70.0.0.1: bytes=32 time=11ms TTL=254
Reply from 70.0.0.1: bytes=32 time<1ms TTL=254
Reply from 70.0.0.1: bytes=32 time=10ms TTL=254

Ping statistics for 70.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 11ms, Average = 5ms

C:\>ping 172.21.0.1

Pinging 172.21.0.1 with 32 bytes of data:

Reply from 172.21.0.1: bytes=32 time=3ms TTL=254
Reply from 172.21.0.1: bytes=32 time=11ms TTL=254
Reply from 172.21.0.1: bytes=32 time=11ms TTL=254
Reply from 172.21.0.1: bytes=32 time=11ms TTL=254

Ping statistics for 172.21.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 11ms, Average = 9ms
```

Рисунок 2.13 – Перевірка ping між комп’ютерами

Далі переходимо до перевірки доступу до веб-сторінок (рис. 2.14).

Рисунок 2.14 – Перевірка доступу до веб-сторінок

Останнім етапом є тестування телефонії (рис. 2.15).

Рисунок 2.15 – Перевірка телефонії

У результаті віртуальна мережа успішно налаштована і без виникнення помилок.

3 РОЗРОБКА ГРАФІЧНОГО ІНТЕРФЕЙСУ

3.1 Інформаційне забезпечення та програмна реалізація

Для вирішення поставленої задачі було складено план реалізації:

— Визначити функціональність додатку

— Вибрати стилістичне оформлення

— Вибрати мови програмування для найкращого функціонування додатку

— Створити елементи графіки для інтерфейсу

— Перевірити інтерфейс

— Протестувати додаток на віртуальній GRE мережі

Створення графічного інтерфейсу починається з «кістяку». Для цього використаємо мови розмітки HTML. Далі створюється зовнішній вигляд додатку за допомогою мови каскадних стилів CSS, а також візуал для зручності використання. Застосовуючи фреймворк Vuetify це можна зробити швидше та простіше. За допомогою фреймворку, додаток можна використовувати на різних платформах, а також у різних версіях браузерів.

Графічний інтерфейс буде формуватися з таких елементів:

— Форма для конфігурації роутерів

— Форма для конфігурації світчів

— Форма для конфігурації GRE тунелів

— Форма налаштування DHCP

— Форма для конфігурації серверів vlan

— Вікно для формування готового коду на основі вхідних даних

Детальнішу інформацію та програмний код можна побачити в Додатках.

3.2 Використання графічного інтерфейсу

Додаток складається з «вікон» в яких знаходяться поля для конфігурації інтерфейсів, а також поле з результатом, куди записується згенерований код для налагодження GRE мережі (рис 3.1).

Рисунок 3.1 – Інтерфейс додатку

Вікно «Інтерфейс» складається з полів FastEthetnet, ip address та mask, які потрібні для конфігурації інтерфейсів портів роутерів (рис. 3.2).

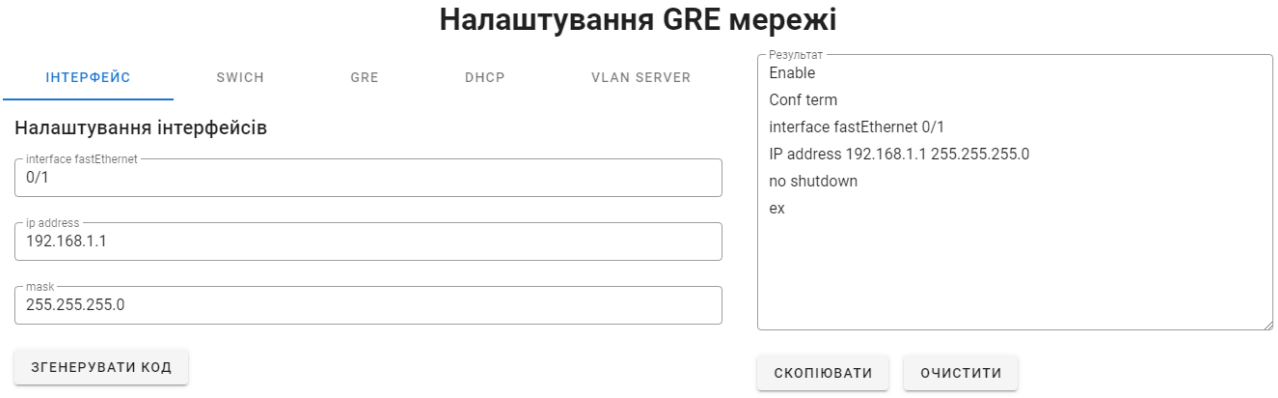

Рисунок 3.2 – Приклад згенерованого коду для інтерфейсів портів роутерів

Вікно «Switch» складається з полів FastEthernet, Vlan та кнопки для вибору режиму порту, які потрібні для конфігурації світчів (рис. 3.3).

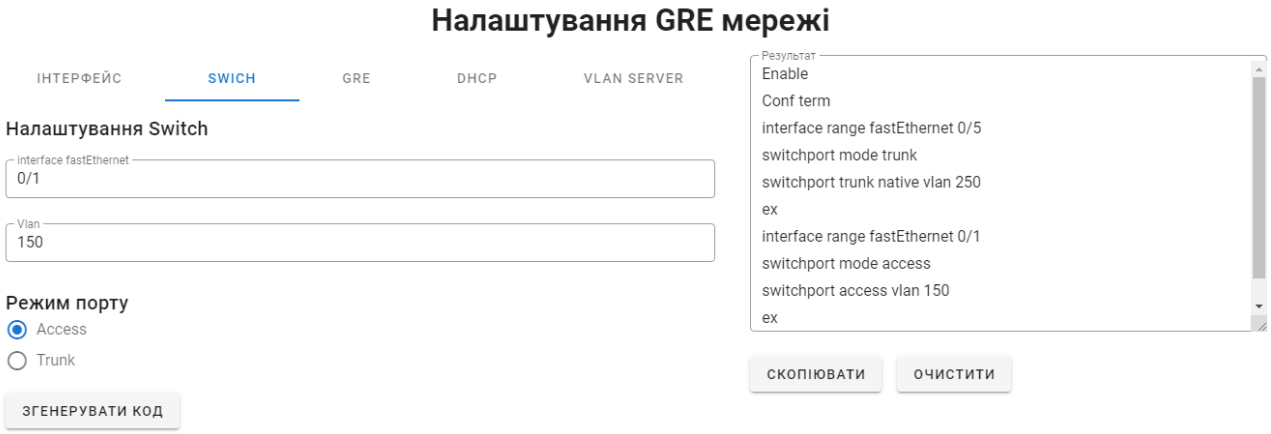

Рисунок 3.3 – Приклад згенерованого коду для конфігурації Switch

Вікно «GRE» складається з полів Interface Tunnel, IP адреса, маска, Tunnel source та Tunnel destination необхідних для налаштування GRE тунелю (рис. 3.4).

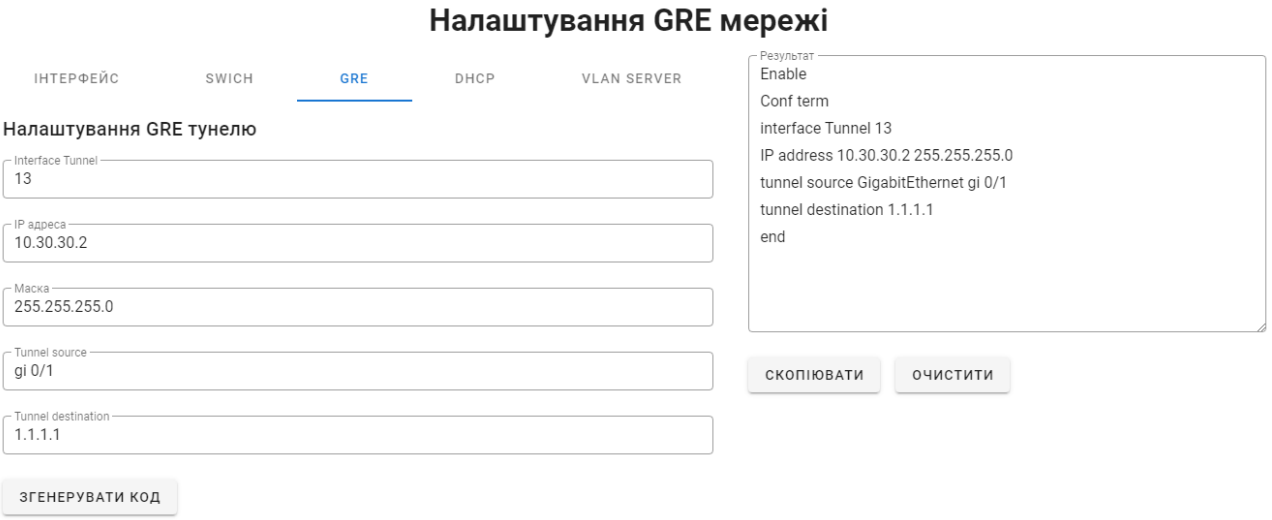

Рисунок 3.4 – Приклад згенерованого коду для конфігурації GRE тунелю

Вікно «DHCP» складається з 2 частин: полів з початком та кінцем діапазону ip адрес та конфігуряціями vlan. Щоб настроїти Vlan потрібно заповнити поля: номер vlan, ip address, mask та роутер за замовчуванням (рис. 3.5). Також є можливість додавання ще однієї мережі vlan, для цього є спеціальна кнопка.

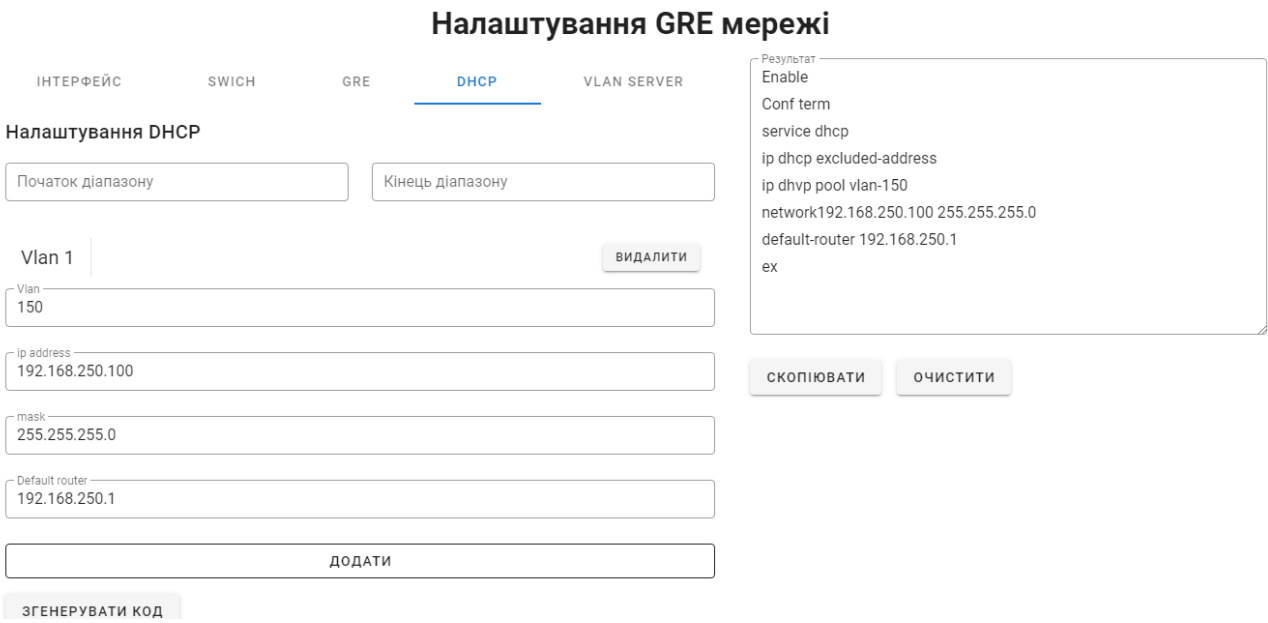

Рисунок 3.5 – Приклад згенерованого коду для конфігурації DHCP

Вікно «Vlan server» складається з полів FastEthernet, тип інкапсуляції, номер мережі vlan, ip адреси та маски, які потрібні для налаштування vlan серверу (рис. 3.6).

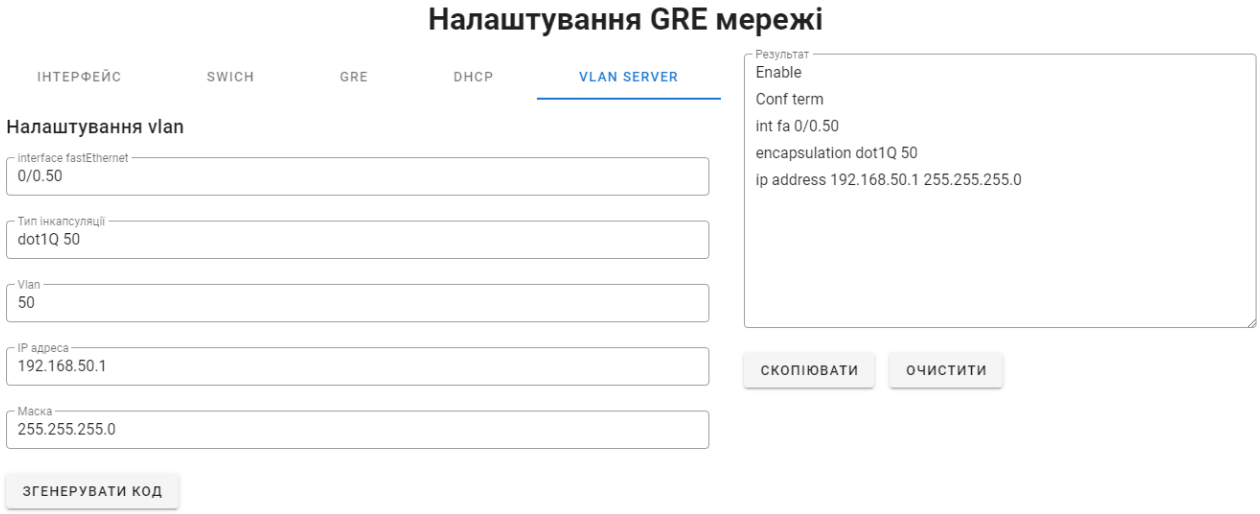

Рисунок 3.6 – Приклад згенерованого коду для конфігурації Vlan серверу

Всі поля повинні бути заповнені, а також перевіряються на правильність вхідних даних. Якщо поле заповнено з помилкою, та/або дані відсутні, то відображається повідомлення з описом помилки (рис. 3.7-8).

Рисунок 3.7 – Відсутні дані в формі

Рисунок 3.8 – Не коректний формат даних

Натиснувши кнопку «Згенерувати код» в спеціальне поле додається новий згенерований код. Потім програмою опрацьовуються вхідні дані та відбувається виведення потрібних команд у вікно справа. Якщо потрібно, можна скопіювати текст та/або очистити його за допомогою відповідних кнопок.

Для того щоб перевірити правильність роботи інтерфейсу було створено віртуальну GRE мережу в Cisco Packet Tracer 8.1. Це ПЗ розроблене для моделювання різних мереж, дозволяє проводити експерименти з поведінкою мережі та давати оцінку можливим ситуаціям. Приклад розробленої мережі показано на рис. 3.9.

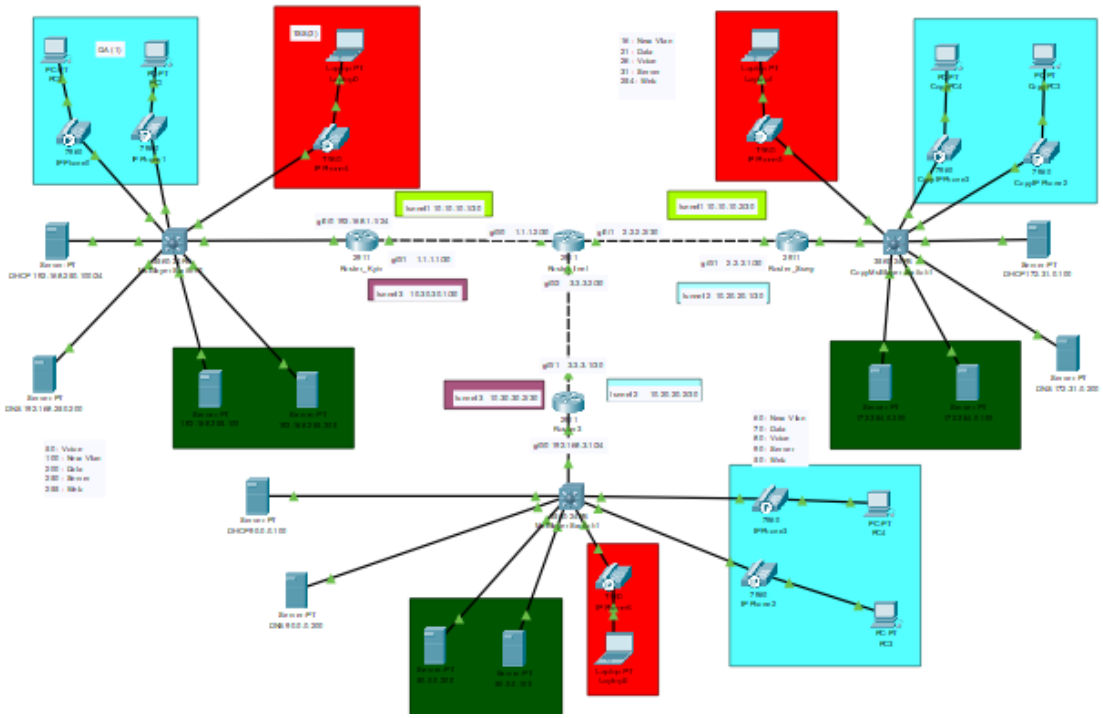

Рисунок 3.9 – Віртуальна GRE мережа

Спочатку за допомогою додатку потрібно скопіювати згенеровані команди та вставити їх в термінал роутеру або світча для конфігурації. Далі на рис. 3.10 приведено зразок налаштування Switch , використовуючи згенеровані команди.

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int gi 0/1
Switch(config-if)#switchport trunk encapsulation dotlq
Switch(config-if)#switchport mode trunk
Switch(config-if)#ex
Switch(config)#int fa 0/5
Switch(config-if)#switchport access vlan 250
Switch(config-if)#ex
```

Рисунок 3.10 – Зразок налаштування Switch за допомогою додатку

Після конфігурації всіх роутерів, світчів, а також GRE тунелю треба протестувати дієздатність системи. Для цього пінгуємо ПК двох різних офісів, якщо все налаштовано правильно в графі втрачених пакетів буде 0 (рис. 3.11).

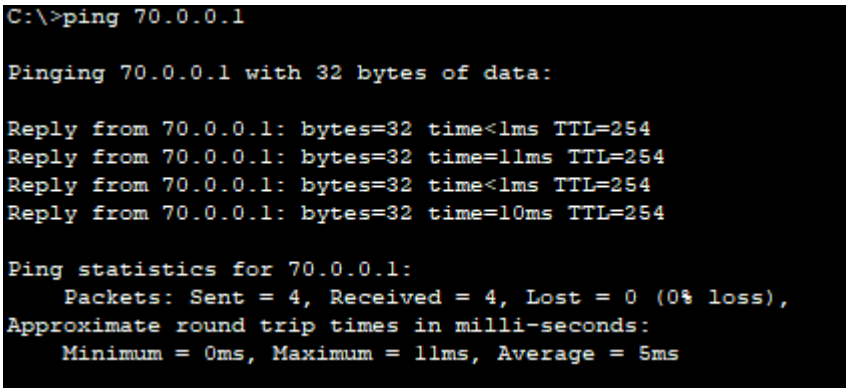

Рисунок 3.11 – Перевірка з’єднання між ПК

Тестування показало, що віртуальна GRE мережа побудована без помилок за допомогою графічного інтерфейсу.

ВИСНОВКИ

Висновки до кваліфікаційної магістерської роботи можна сформулювати наступним чином:

Було визначено принципи роботи відомих моделюючих мережевих симуляторів, до яких належить Cisco Packet tracer. Також популярними є GNS3, UNetLab та ін. Усі ці симулятори дозволяють отримати доступ до головних видів мережевого устаткування, до яких відносяться маршрутизатори та комутатори. Але мінусом нинішніх симуляторів є відсутність графічного інтерфейсу налаштування GRE мережі, що спричиняє складність та довготривалість процесу її конфігурування.

У межах роботи розроблено веб-орієнтовану інформаційну систему. Графічний інтерфейс якої надає можливість налаштування віртуальної GRE мережі. Під час роботи з додатком потрібно задати ключові параметри роутерів та обрати протокол динамічної маршрутизації, для якого треба одержати налаштування. Програма дає можливість легко переносити згенерований код конфігурацій роутера в налаштування справжнього мережевого устаткування.

Розроблений додаток дозволяє успішно настроїти мультисервісні мережі Ethernet як початківцям, без знання команд конфігурацій в Cisco, так і експертам у цій справі. Також ПЗ надає можливість автоматизованого налаштування маршрутизаторів та звільняє від здійснення рутинних операцій. Створений графічний інтерфейс можна використовувати як для налаштування GRE в симуляторах, так і на справжньому устаткуванні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Africa A. D. M., Lontoc C. R. V., Mendez R. J. M., Espiritu F. M. Application of computer systems in vpn networks, 2020.

2. Alshalan A., Pisharody S., Huang D. A Survey of Mobile VPN Technologies, 2018.

3. Awad A. I., Fairhurst M. Information Security: Foundations, Technologies and Applications, 2018. 406 p.

4. Belzer J. Encyclopedia of Computer Science and Technology, 2018. 200 p.

5. Besnard V., Brun M., Cerchio L. Software Technologies: Applications and Foundations, 2018.

6. Bhunia S., Tehranipoor M. Hardware security: A hands-on learning approach, 2018. 502 p.

7. Birudavolu S., Nag B. Business Innovation and ICT Strategies, 2019.

8. Carter J., O'Grady M., Rosen C. Higher Education Computer Science, 2018. 244 p.

9. Froehlich F. E., Kent A., Hall C. M. The Froehlich/Kent Encyclopedia of Telecommunications, 2021. 500 p.

10. Gunkel D. J. Hacking cyberspace, 2019. 256 p.

11. Hardle W. K., Lu H., Shen X. Handbook of Big Data Analytics, 2018.

12. Harvey M., Rhonda J., Eli M. The Telecommunications Revolution, 2018. 238 p.

13. Hercog D. Communication Protocols: Principles, Methods and Specifications, 2020.

14. Hills M. T. Telecommunications Switching Principles, 2020.

15. Huddleston R. Introduction to HTML and CSS, 2019.

16. Irizarry R. A. Introduction to Data Science, 2019. 743.

17. Jyothi K. K., Reddy B. I., Study on Virtual Private Network (VPN), VPN ’ s Protocols And Security, 2018.

18. Kang B. H., Balitanas M. O. Vulnerabilities of VPN using IPsec and defensive measures, 2018.

19. Martin S. Modern Telecommunications, 2018. 202 p.

20. Matin M. A., Telecommunication Networks - Trends and Developments, 2019. 287 p.

21. Meyer J. The Essential Guide to HTML, 2018.

22. Ogudo, K. A. Analyzing generic routing encapsulation (GRE) and IP Security (IPSec) tunneling protocols for secured communication over public networks, 2019.

23. Pooch U. W., Machuel D., McCahn J. Telecommunications and Networking, 2018. 577 p.

24. Ramakrishnan S. Cryptographic and Information Security, 2018. 986 c.

25. Ren W., Wang L., Choo K. K. R., Xhafa F. Security and privacy for big data, cloud computing and applications, 2019. 330 p.

26. Robillard M. P. Introduction to Software Design with Java, 2019. 297 p.

27. Serafin, F. Enabling modeling framework with surrogate modeling capabilities and complex networks, 2019.

28. Sholihah W., Rizaldi T., Novianty I. Information and communication system technology with VPN site-to-site IPsec, 2019.

29. Shunmuganathan S., Saravanan R. D., Palanichamy Y. Securing VPN from insider and outsider bandwidth flooding attack, 2020.

30. Suh S. C., Anthony T. Big data and visual analytics, 2018. 263 p.

31. Surasak T., Scott Huang C. H. Enhancing VoIP Security and Efficiency using VPN, 2019.

32. Susanto A., Meiryani The future of information technology, 2019. 401 p.

33. Tiller J. S. A technical guide to IPSec virtual private networks, 2018. 376 p.

34. Uddin M., Evan N., Alam M., Arefin M. Analysis of Generic Routing Encapsulation (GRE) over IP Security (IPSec) VPN Tunneling in IPv6 Network, 2021. 383 c.

35. Wu Z., Xiao M. Performance evaluation of VPN with different network topologies, 2019.

36. Yarali A. IOT: Platforms, connectivity, applications and services, 2018.

37. Анісімов А.В. Інформаційні системи та бази даних: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. / Анісімов А.В., Кулябко П.П. Київ. 2017. 110 с.

38. Бережна О. Б. Інформатика та комп'ютерна техніка. 1 частина : Навч. посіб. / О. Б. Бережна. Х. : ХНЕУ ім. С. Кузнеця, 2017. 164 с.

39. Бородкіна, І. Л. Web-технології та web-дизайн: застосування мови HTML для створення електронних ресурсів. Київ: Ліра-К, 2020. 210 с.

40. Климаш, М. М. Телекомунікаційні системи передавання інформації. Львів: Вид-во Львівської політехніки, 2018. 630 с.

ДОДАТКИ

Додаток А

Фрагмент коду з файлу App.vue: Відповідає за відображення базової структури програми та підключення всіх необхідних компонентів.

```
<template>
<v-app id="inspire">
  <h1 class="header text-center mt-4">Налаштування GRE мережі</h1>
  <v-row class="px-4">
   <v-col md="7">
    <v-tabs v-model="tab" grow>
     <v-tab>Інтерфейс</v-tab>
     <v-tab>Swich</v-tab>
     <v-tab>GRE </v-tab>
     <v-tab>DHCP</v-tab>
     <v-tab>Vlan Server</v-tab>
    </v-tabs>
    <v-tabs-items v-model="tab">
     <v-tab-item>
      <Interface @generateInterface='getData'></Interface>
     </v-tab-item>
     <v-tab-item>
      <SwitchComponent @generateSwitch='getData'></SwitchComponent>
     </v-tab-item>
     <v-tab-item>
      <GRE @generateGRE='getData'></GRE>
     </v-tab-item>
     <v-tab-item>
      <Dhcp @generateDHCP='getData'></Dhcp>
```

```
        </v-tab-item>
        <v-tab-item>
          <VlanServer @generateServer='getData'></VlanServer>
        </v-tab-item>
      </v-tabs-items>
    </v-col>
    <v-col>
      <v-textarea
        v-model="result"
        label="Результат"
        rows="10"
        required
        outlined
        dense
      ></v-textarea>
      <v-btn class="mr-4" @click="copy">Скопіювати</v-btn>
      <v-btn class="mr-4" @click="result = ''">Очистити</v-btn>
    </v-col>
  </v-row>
 </v-app>
</template>

<script>

import Interface from './components/Interface';
import SwitchComponent from './components/Switch';
import GRE from './components/GRE';
import Dhcp from './components/Dhcp';
import VlanServer from './components/VlanServer';

export default {
 name: 'App',
 data: () => ({
  tab: null,
```

```
    result: ""
  }),
  components: {
    Interface,
    SwitchComponent,
    GRE,
    Dhcp,
    VlanServer
  },
  methods: {
    getData(data) {
      if(this.result == "") {
        this.result = "Enable\nConf term\n";
      }
      this.result += data.text;
    },
    copy() {
      navigator.clipboard.writeText(this.result);
    }
  }
};
</script>
```

Додаток Б

Фрагмент коду з файлу Interface.vue: Відповідає за відображення конфігурацій інтерфейсів.

```
<template>
  <v-container>
    <h1 class="title mb-4">Налаштування інтерфейсів</h1>
    <v-form ref="form" v-model="valid" lazy-validation>
      <v-text-field
        v-model="data.fastEthernet"
        label="interface fastEthernet"
        outlined
        dense
        :rules="interfaceRules"
      ></v-text-field>
      <v-text-field
        v-model="data.ip"
        label="ip address"
        :rules="ipAddressRules"
        outlined
        dense
      ></v-text-field>
      <v-text-field
        v-model="data.mask"
        label="mask"
        :rules="maskRules"
        outlined
        dense
      ></v-text-field>
      <v-btn class="mr-4" :disabled="!valid" @click="generate">Згенерувати код</v-btn>
```

```
    </v-form>
  </v-container>
</template>

<script>
import validation from '../mixins/validation';

export default {
  mixins: [validation],
  name: 'Interface',
  data: () => ({
    valid: false,
    data: {
      fastEthernet: "",
      ip: "",
      mask: ""
    }
  }),
  methods: {
    generate() {
      var string = "interface fastEthernet "+this.data.fastEthernet+"\n"+
      "IP address "+this.data.ip+" "+this.data.mask+"\n"+
      "no shutdown"+"\n"+
      "ex"+"\n";
      this.$emit('generateInterface', {
        text: string,
      });
    }
  }
};
</script>
```

Додаток В

Фрагмент коду з файлу Switch.vue: відповідає за конфігурації Switch.

```
<template>
  <v-container>
    <h1 class="title mb-4">Налаштування Switch</h1>
    <v-form ref="form" v-model="valid" lazy-validation>
      <v-text-field
        v-model="data.fastEthernet"
        label="interface fastEthernet"
        :rules="interfaceRules"
        outlined
        dense
      ></v-text-field>
      <v-text-field
        v-model="data.vlan"
        label="Vlan"
        :rules="vlanRules"
        outlined
        dense
      ></v-text-field>
      <v-radio-group class="mt-0 pt-0" v-model="data.mode">
        <h1 class="title ">Режим порту</h1>
        <v-radio label="Access" value="access"></v-radio>
        <v-radio label="Trunk" value="trunk"></v-radio>
      </v-radio-group>
      <v-btn class="mr-4" :disabled="!valid" @click="generate">Згенерувати код</v-btn>
    </v-form>
  </v-container>
</template>
```

```
<script>
import validation from '../mixins/validation';

export default {
  name: 'Switch',
  mixins: [validation],
  data: () => ({
    valid: false,
    data: {
      fastEthernet: "",
      vlan: "",
      mode: "access"
    }
  }),
  methods: {
    generate() {
      if(this.data.mode == "access") {
        var string = "interface range fastEthernet "+this.data.fastEthernet+"\n"+
        "switchport mode access\n"+
        "switchport access vlan "+this.data.vlan+"\n"+
        "ex"+"\n";
      } else {
        var string = "interface range fastEthernet "+this.data.fastEthernet+"\n"+
        "switchport mode trunk\n"+
        "switchport trunk native vlan "+this.data.vlan+"\n"+
        "ex"+"\n";
      }
      this.$emit('generateSwitch', {
        text: string,
      });
    }
  }
};
</script>
```

Додаток Г

Фрагмент коду з файлу GRE.vue: відповідає за налаштування GRE тунелю.

```
<template>
  <v-container>
    <h1 class="title mb-4">Налаштування GRE тунелю</h1>
    <v-form ref="form" v-model="valid" lazy-validation>
      <v-text-field
      v-model="data.fastEthernet"
      label="interface Tunnel"
      :rules="interfaceRules"
      outlined
      dense
      ></v-text-field>
      <v-text-field
      v-model="data.ip"
      label="IP адреса"
      :rules="ipAddressRules"
      outlined
      dense
      ></v-text-field>
      <v-text-field
      v-model="data.mask"
      label="Маска"
      :rules="maskRules"
      outlined
      dense
      ></v-text-field>
      <v-text-field
      v-model="data.encapsulation"
```

```
            label="Tunnel source"
            :rules="interfaceRules"
            outlined
            dense
            ></v-text-field>
            <v-text-field
            v-model="data.vlan"
            label="Tunnel destination"
            :rules="ipAddressRules"
            outlined
            dense
            ></v-text-field>
            <v-btn class="mr-4" :disabled="!valid" @click="generate">Згенерувати код</v-btn>
        </v-form>
    </v-container>
</template>

<script>
import validation from '../mixins/validation';
export default {
  name: 'GRE',
  mixins: [validation],
  data: () => ({
    valid: false,
    data: {
        fastEthernet: "",
        encapsulation: "",
        ip: "",
        mask: "",
        vlan: ""
      }
  }),
  methods: {
    generate() {
```

```
        var string = "interface Tunnel " + this.data.fastEthernet +
        "\n" + "IP address " + this.data.ip + " " + this.data.mask +
        "\n" + "tunnel source GigabitEthernet " + this.data.encapsulation +
        "\n" + "tunnel destination" + this.data.vlan +
        "\n" + "end" + "\n";
        this.$emit('generateRoute', {
            text: string,
        });
    }
  }
};
</script>
```

Додаток Д

Фрагмент коду з файлу Dhcp.vue: відповідає за налаштування DHCP.

```
<template>
  <v-container>
    <h1 class="title mb-4">Налаштування DHCP</h1>
      <v-form ref="form" v-model="valid" lazy-validation>
        <v-row>
          <v-col class="py-0">
            <v-text-field
              v-model="startRange"
              label="Початок діапазону"
              :rules="ipAddressRules"
              outlined
              dense
            ></v-text-field>
          </v-col>
          <v-col class="py-0">
          <v-text-field
            v-model="finishRange"
            label="Кінець діапазону"
            :rules="ipAddressRules"
            outlined
            dense
          ></v-text-field>
          </v-col>
        </v-row>
        <div v-for="(item, index) in data" :key="index">
          <v-toolbar flat color="white">
            <v-toolbar-title>Vlan {{ index+1 }}</v-toolbar-title>
```

```
    <v-divider
    class="mx-4"
    inset
    vertical
    ></v-divider>
    <v-spacer></v-spacer>
<v-btn @click="deleteItem(item)" small>Видалити</v-btn>
</v-toolbar>
<v-text-field
    v-model="item.vlan"
    label="Vlan"
    :rules="vlanRules"
    outlined
    dense
></v-text-field>
<v-text-field
    v-model="item.ip"
    label="ip address"
    :rules="ipAddressRules"
    outlined
    dense
></v-text-field>
<v-text-field
    v-model="item.mask"
    label="mask"
    :rules="maskRules"
    outlined
    dense
></v-text-field>
<v-text-field
    v-model="item.defaultRouter"
    label="Default router"
    :rules="ipAddressRules"
    outlined
```

```
                    dense
                  ></v-text-field>
                </div>
                <v-btn block class="mb-4 mt-0" @click="add" outlined>Додати</v-btn>
            <v-btn class="mr-4" :disabled="!valid" @click="generate">Згенерувати код</v-btn>
            </v-form>
    </v-container>
</template>

<script>
import validation from '../mixins/validation';
export default {
  name: 'DHCP',
  mixins: [validation],
  data: () => ({
    valid: false,
    startRange: "",
    finishRange: "",
    data: [
        {
            vlan: "",
            ip: "",
            mask: "",
            defaultRouter: ""
        }
    ]
  }),
  methods: {
    add() {
      this.data.push({
            vlan: "",
            ip: "",
            mask: "",
            defaultRouter: ""
```

```
      })
    },
    deleteItem(item) {
      const index = this.data.indexOf(item)
      this.data.splice(index, 1)
    },
    generate() {
        var string = "service dhcp"+"\n"+
            "ip dhcp excluded-address "+this.startRange+" "+this.finishRange+"\n";
        this.data.map(item => {
            string += "ip dhvp pool vlan-"+item.vlan+"\n"+
        "network"+item.ip+" "+item.mask+"\n"+
        "default-router "+item.defaultRouter+"\n"+
        "ex"+"\n";
        })
        this.$emit('generateDHCP', {
            text: string,
        });
    }
  }
};
</script>
```