МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

«Графічний інтерфейс налаштування VPN доступу до ресурсів корпоративної мережі»

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

Керівник роботи **Великодний Д.В.**

Студент групи ІН-61 **Ситніков В. О.**

Звіт захищено з оцінкою _____ «_____» _____2020 р.

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерних наук

Затверджую _____
Зав. кафедрою Довбиш А.С.
«_____» _____2020 р.

ЗАВДАННЯ
до випускної роботи

Студента четвертого курсу, групи ІН-61 спеціальності «Інформатика», денної форми навчання Ситнікова Владислава Олеговича.

Тема: «Графічний інтерфейс налаштування VPN доступу до ресурсів корпоративної мережі»

Затверджена наказом по СумДУ
№_____ від _____2020 р.

Зміст пояснювальної записки: 1) аналіз предметної області; 2) моделювання та проектування; 3) огляд методів для розробки додатку; 4) розробка графічного інтерфейсу для налаштування VPN доступу корпоративної мережі; 5) аналіз результатів.

Дата видачі завдання «_____» _____2020 р.
Керівник випускної роботи _____ Великодний Д.В.
Завдання прийняв до виконання _____ Ситніков В.О.

РЕФЕРАТ

Записка: 66 стр., 32 рис., 6 додатків, 24 джерела.

Об'єкт дослідження — графічний інтерфейс налаштування VPN доступу до ресурсів корпоративної мережі.

Мета роботи — розробити графічний інтерфейс, який повинен спрощувати налаштування VPN доступу.

Результати — у результаті роботи було побудовану віртуальну приватну мережу, з використанням програмного забезпечення Cisco Tacket Tracer. Проведення налаштування VPN мережі та перевірка працездатності. Також було створено графічний інтерфейс у вигляді веб-додатку для налаштування VPN мережі. Додаток надає можливість вводити дані приватної мережі і автоматично генерувати коди для налаштування пристроїв. Також відбувається перевірка вхідних даних для запобігання допущення помилок, що надає можливість без зусиль налаштовувати мережу як початківцям так і професіоналам. Протестувавши роботу додатку з використанням віртуальної приватної мережі було доведено її працездатність і готовність до використння а реальних пристроях.

ГРАФІЧИЙ ІНТЕРФЕЙС, VPN ДОСТУП, HTML, CSS, JAVASCRIPT, VUE.JS, CISCO PACKET TRACER

ЗМІСТ

ВСТУП

Останнім часом інтерес до віртуальних приватних мереж у світі телекомунікацій зріс. Це пов’язано з необхідністю дешевого підключення користувачів до віддалених офісів та через Інтернет, адже це зменшує витрати на обслуговування мережі компанії.

Зазвичай людям потрібно отримати доступ до інформації, яка знаходиться та зберігається на домашніх комп'ютерах або на комп’ютерах компанії. Звичайно, дану задачу можна вирішити, якщо надати віддалений доступ до неї за допомогою модемів та телефонних ліній.

Недоліком цього рішення є те, що для здійснення дзвінків з іншої країни це коштує чималих грошей. Є ще одне рішення під назвою VPN. Перевага технології VPN полягає в тому, що віддалений доступ здійснюється не по телефонній лінії, а через Інтернет, що дешевше і краще. Я вважаю, що технологія VPN має перспективу бути широко розповсюдженою по всьому світу.

Оскільки при великих значеннях, коли занадто багато мереж VPN, дуже висока ймовірність того, що може виникнути помилка через людські фактори, що є дуже трудомістким процесом. Отже, було вирішено розробити сервіс для автоматичного налаштування спершу інтерфейсів, а потім і серверу VPN на роутерах. В якості гарного прикладу візьмемо роутер CISCO 2911.

Проаналізувавши мережу Internet, мною не було виявлено подібних сервісів. Для розробки сервісу будемо використовувати засоби та мови розмітки HTML/CSS, мови програмування JavaScript, фреймворк Vuetify.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Розберемо детальніше загрози інформаційної безпеки. Під даним типом загрози мова йде про потенційну подію, що може будь-яким чином завдати шкоди. В майбутньому під загрозою обробки інформації ми будемо мати на увазі потенційний вплив на мережу, оскільки це може прямо чи опосередковано зашкодити її безпеці.

У наш час різні типи загроз широко розповсюджені та часто використовуються в протиправних діях. Можливий перелік інформації про безпеку повинен бути розглянутий, з метою вивчення та використання повного і необхідного обладнання захисту.

Перелік загроз, оцінка ймовірності реалізації та модель правопорушника є основою для аналізу ризиків реалізації загрози та сформування вимог систем захисту. Виявляючи можливість загроз, рекомендується класифікувати ці загрози для аналізу за різними причинами. Кожен класифікаційний символ відображає символ із узагальненими системними вимогами.

Оскільки на інформацію, що зберігається та обробляється в мережі MODERN, впливає безліч факторів, необхідно класифікувати загрози інформаційній безпеці, що унеможливлює формалізацію завдання опису всіх загроз. Тому з міркувань безпеки користувацька система не визначає повний рівень усієї загрози, а швидше визначає рівень загрози «гачок».

Наступні функції орієнтири можуть зберігати класифікацію можливих загроз безпеці. Розглянемо таблицю 1.1.

Таблиця 1.1 Перелік загроз та причин виникнення

№	Назва класифікації	Перелік
1.	За природою виникнення	Природна загроза або стихійне лихо, яке має фактичний вплив на фізичний процес.
		Техногенні загрози безпеці, реальна поведінка особи.
2.	За ступенем навмисності	Загрози, які виникають через ті чи інші помилки чи недбалість, також пов’язані з загрозою некомпетентності.
		Загрози навмисної протидії, яскравим прикладом є протизаконні дії зловмисників.
3.	За безпосереднім джерелом загроз	Природне середовище, наприклад стихійні лиха, магнітні бурі та ін.
		Проблеми з персоналом, наприклад, підкуп працівників та розголошення конфіденційної інформації.
		Дозволене програмне та апаратне забезпечення для публікації даних, вибраних в операційній системі.
		Несанкціоноване програмне забезпечення, наприклад зараження комп'ютерів вірусами, що мають руйнівні функції.
4.	Відповідно за положенням до джерела загрози	Поза межами контрольованої зони мережі, наприклад, як перехоплення передачі через канал зв'язку, електромагнітні перешкоди, що випромінюються іншими електромагнітними обладнаннями, та перехоплення інших різних звуків, що роздаються.

Продовження таблиці 1.1

№	Назва класифікації	Перелік
4.		Можна використовувати VPN, але ресурси можуть бути невірно реалізовані.
5.	За ступенем активності користувача	Незалежно від активності розширення паролів для шифрування даних.
		Тільки під час обробки даних, приклад загроз виконанню та поширенню програмних вірусів.
6.	За ступенем загального впливу	Загрози копіювання пасивного типу, використовують без зміни даних на файли які не завжди структурно збільшуються.
		Активні загрози, такі як впровадження троянських коней та подібного типу вірусів.
7.	Доступ користувача або програми до ресурсів	Загрози, які виникають під час доступу до мережевих ресурсів, наприклад, загроза несанкціонованого доступу.
		Загрози, які виникають, коли доступ до ресурсів дозволений, наприклад, загрози несанкціонованого або неправильного використання мережевих ресурсів.
8.	За способом доступу до загальних ресурсів	Загрози, що використовують стандартні шляхи доступу до ресурсів, такі як незаконний доступ до паролів та інших даних про розмежування, додатково блокуються зареєстрованими користувачами.

Продовження таблиці 1.1

№	Назва класифікації	Перелік
8.		Загрози реалізуються за допомогою прихованих нестандартних шляхів доступу до ресурсів, таких як несанкціонований доступ до мережевих ресурсів за допомогою недокументованих функцій операційної системи.
9.	За поточним місцем розташування інформації, яка зберігається та опрацьовується в загальній мережі	Загрозливий доступ до інформації, що зберігається на зовнішніх пристроях зберігання даних, таких як несанкціоноване копіювання конфіденційної інформації з жорсткого диска.
		Існує загроза доступу до розповсюдженої інформації в затримці зв'язку, наприклад, незаконно підключитися до контактної особи або змінити передане повідомлення з вхідної відстані звіту про помилку, незаконне з'єднання із затриманим зв’язком з метою безпосередньої заміни помилкової розмови законного користувача на інформацію про помилку, введену пізніше.
		Загроза, коли отримують доступ до інформації, що зберігається в пам'яті, наприклад, зчитування залишкової інформації з пам'яті та програм, що отримують доступ до областей системної пам'яті.

Продовження таблиці 1.1

№	Назва класифікації	Перелік
9.	За поточним місцем розташування інформації, яка зберігається та опрацьовується в загальній мережі	Загроза, коли отримують доступ до інформації, що відображається на терміналі або надрукована на принтері, наприклад, записаної інформації, що відображається на прихованій відеокамері [1].

Як вже згадувалося раніше, небезпека для мережі класифікується як випадкова та навмисна. Аналіз проектування, виготовлення експлуатаційного досвіду показує, що ця інформація піддається різному випадковому впливу протягом життєвого циклу мережі VPN та на всіх етапах роботи та функціонування.

Причинами різноманітних навмисних чи випадкових впливів при експлуатації можуть бути:

- Надзвичайні ситуації внаслідок стихійних лих та відключення електроенергії.
- Відмови та несправності обладнання
- Помилка програмного забезпечення.
- Погана робота сервісного персоналу та користувачів.
- Втручання навколишнього середовища в лінії зв'язку через вплив на зовнішні об’єкти.

Помилки програмного забезпечення – це найпоширеніші помилки на комп’ютерах. Серверне програмне забезпечення, робочі станції, маршрутизатори тощо, пишуть люди, де вони фактично роблять помилки. Чим вища складність програмного забезпечення для обману, тим більшою є можливість виявлення помилок та загроз [2].

Більшість з них не становитиме ніякої небезпеки, а деякі можуть спричинити серйозні негативні наслідки, такі як отримання зловмисного

контролю над сервером, неефективний сервер, несанкціоноване використання ресурсів, таких як використання комп’ютера як платформи атаки та тощо. За допомогою пакетів послуг, які регулярно випускаються виробниками програмного забезпечення, такі помилки легко усуваються. Своєчасна установка таких програмних пакетів має вирішальне значення для захисту інформації.

Навмисні погрози, пов'язані з умисними діями злочинця, можуть бути виконані різними службовцями, відвідувачами, конкурентами, найманцями тощо. Злочинців можуть спонукати різні мотиви: незадоволення своєю професією, чисті матеріальні вигоди, цікавість, конкуренція, твердження своїх побажань за будь-яку ціну тощо.

Несанкціонований доступ – найпоширеніший та найрізноманітніший вид зловживань комп'ютером. Суть NMS полягає в тому, що вона порушує правила обмеження доступу, встановлені відповідно до політики безпеки, прийнятої організацією, й користувачі чи злочинці отримують доступ до об'єктів.

NSD буде використовувати будь-які помилки в системі безпеки, які можуть виникнути через необґрунтований вибір методів захисту, неправильне встановлення та налаштування. NSD може бути реалізований як внутрішньо, так і в спеціально створеному апаратному та програмному забезпеченні [3].

Несанкціонований доступ до основних каналів, через які зловмисники можуть отримати доступ до мережевих компонентів та більш ефективно красти, змінювати та/або знищувати інформацію.

Наприклад:

– Створення каналів доступу до інформації (кінцеві користувачі, оператори, системні адміністратори; засоби відображення та інформаційні документи; канали зв'язку) під час використання винуватців та закони користувачів, що перевищують їх повноваження; технологічні пульти управління.

– Лінія зв'язку між обладнанням.

– Непряме електромагнітне випромінювання обладнання та ліній зв'язку.

– Мережі електроживлення, заземлення та інше.

За допомогою різноманітних методів та технологій несанкціонованого доступу до мережі VPN можна докладно описати такі поширені та пов’язані з ними порушення як, перехоплення паролів, незаконне використання привілеїв.

Паролі перехоплюються спеціально розробленими програмами. При спробі зайти в систему програма перехоплювача буде імітувати форму авторизації на екрані дисплея, і дані форми авторизації будуть негайно передані власнику програми перехоплювача, після чого повідомлення про помилку буде виведене та повернене в операційну систему [4].

1.2 Способи створення захищених віртуальних каналів

Будь-який з двох віртуальних вузлів мережі, які утворюють захищений тунель між ними, може належати до кінцевої точки або проміжної точки безпечного потоку повідомлень. Тому існує багато способів формування захищеного віртуального каналу. На рисунку 1.1 показаний приклад захищеного віртуального каналу.

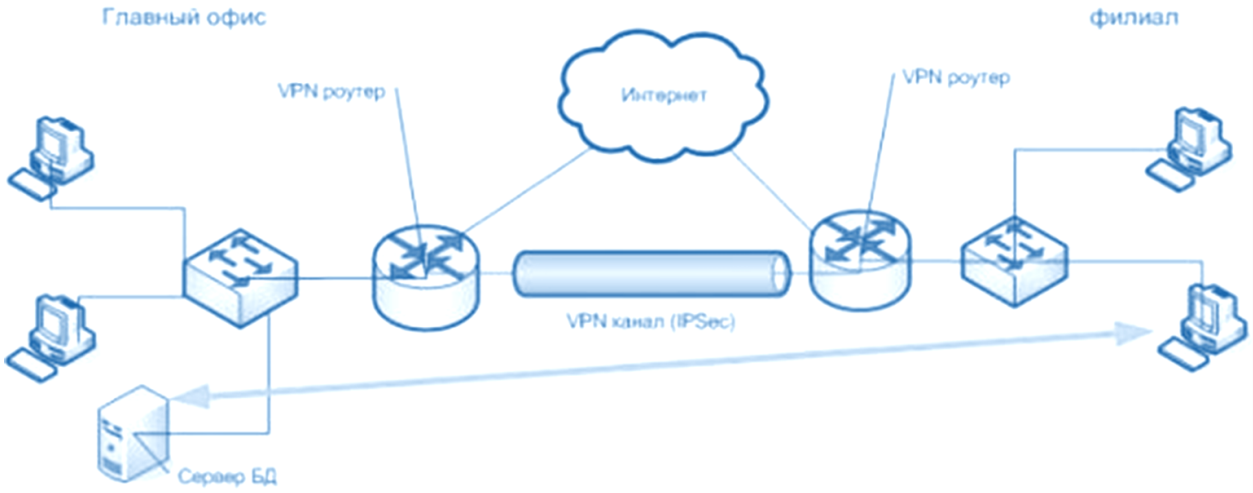

Рисунок 1.1 – Захищений віртуальний канал [5]

З точки зору безпеки, кращим є варіант, коли кінцева точка захищеного тунелю збігається з кінцевою точкою захищеного потоку повідомлень. У цьому випадку канал повністю захищений уздовж усього шляху пакета повідомлень. Однак цей варіант призводить до децентралізованого управління та надмірних витрат на ресурси. Кожен клієнтський комп'ютер локальної мережі потребує встановлення захищеного тунелю, що ускладнює централізоване управління доступом до комп'ютерних ресурсів і не завжди є економічно обґрунтованим. У великій мережі індивідуальне управління кожним клієнтським комп'ютером для налаштування його функцій захисту є дуже трудомістким процесом.

Тому, якщо вам не потрібно захистити трафік в локальній мережі, що належить до віртуальної мережі, рекомендується вибрати маршрутизатор локальної мережі як кінцеву точку захищеного тунелю. Якщо потік повідомлень також повинен бути захищений всередині локальної мережі, комп'ютером, який представляє одну із сторін захищеної взаємодії, він повинен служити кінцевою точкою тунелю в мережі. Під час доступу до локальної мережі віддаленого користувача, його комп'ютер також повинен бути кінцевою точкою захищеного віртуального каналу.

Існує також варіант, який менш безпечний, але більш корисний. За цієї опції робочі станції та сервери локальної мережі та віддалені комп’ютери не беруть участі у створенні захищених тунелів, які прокладаються лише у загальнодоступних мережах з комутацією пакетів, наприклад Інтернет. Кінцевими точками таких тунелів зазвичай є Інтернет-провайдери або межуючі маршрутизатори чи брандмауери. Через віддалений доступ до локальної мережі може бути створений тунель між сервером віддаленого доступу постачальника послуг Інтернету та межуючи Інтернет-провайдером або маршрутизатором чи брандмауером локальної мережі. Під час підключення до локальної мережі утворюється лише тунель між Інтернет-провайдером або маршрутизатором чи брандмауером локальної мережі.

Причиною вище описаного варіанту віртуальної мережі є те, що через зловмисників це є більш сприйнятливим до атак з мереж, що комутуються в пакеті, таких як Інтернет, ніж через телефонні мережі або спеціальні канали зв'язку. Віртуальна мережа, побудована за допомогою цієї опції, має гарну масштабованість та керованість. Для клієнтських комп'ютерів та серверів у мережі WAN захищений тунель повністю прозорий, а програмне забезпечення для цих вузлів залишається незмінним. Однак, оскільки деякий користувацький трафік не захищений каналами загального зв’язку, цей варіант значно знижує безпеку інформаційної взаємодії. Крім того, значна частина роботи над створенням безпечного тунелю залежить від постачальників, яким потрібно довіряти та платити.

Операція створення віртуального тунелю виконується компонентами віртуальної мережі, а віртуальні компоненти працюють на вузлах, що утворюють тунель. Ці компоненти часто називають ініціатором та термінатором тунелю. Ініціатор тунелю інкапсулює або вбудовує пакет даних у новий пакет даних, який містить новий заголовок з інформацією про відправника та приймача поруч із вихідними даними [6].

Незважаючи на те, що всі тунельні пакети є IP-пакетами, інкапсульовані пакети можуть належати до будь-якого типу протоколів, включаючи пакети без маршрутизації, такі як NetBEUI. Маршрут між ініціатором і тунельним термінатором визначається звичайною IP-мережею маршрутизації, яка може бути немережевою мережею. Тунельний термінатор виконує процес зворотної інкапсуляції - видалення нового заголовка і пересилання кожного вихідного пакета до місця призначення в локальному стеку протоколу або локальної мережі. Сама інкапсуляція не впливає на безпеку пакетів повідомлень, що передаються через тунель VPN. Але інкапсуляція дозволяє повністю зашифрувати інкапсульовані пакети даних.

Конфіденційність інкапсульованого пакету даних забезпечується за допомогою шифрування капсульованого пакету даних та цілісності та достовірності приміток до випуску. Оскільки існує багато методів захисту

даних шифрування, важливо, щоб ініціатор тунелю та термінатор використовували один і той же метод і мали можливість координувати цю інформацію один з одним.

1.2.1 Класифікація VPN мереж

Загалом, можна класифікувати рішення VPN за кількома основними параметрами, як показано на рис. 1.2.

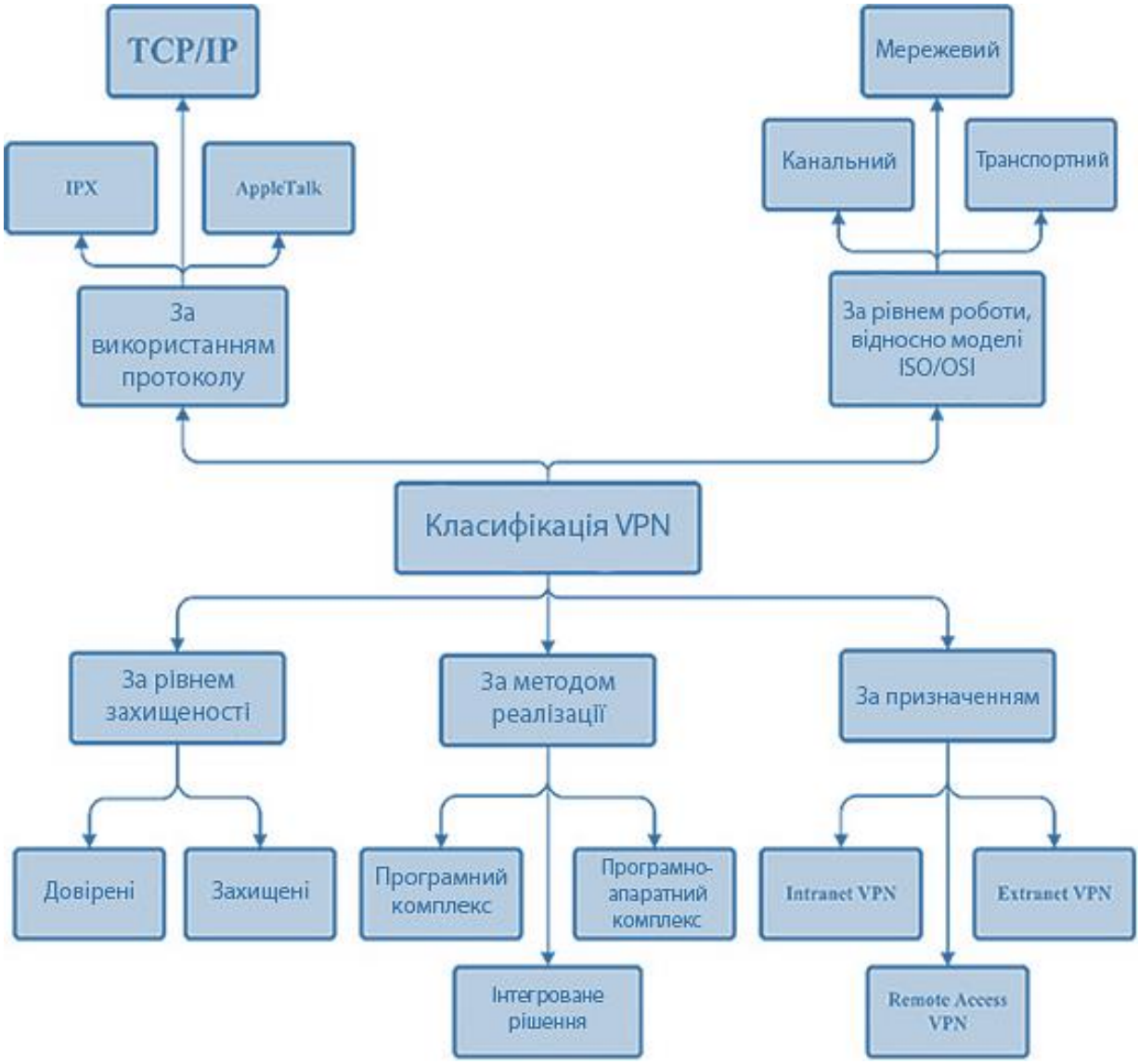

Рисунок 1.2 – Класифікація VPN [7]

Розглянемо таблицю з описом додаткової інформації про деякі типи VPN (табл.1.2).

Таблиця 1.2 Додатковий опис про VPN

Назва	Опис
Безпечний VPN	Найпоширеніша версія спеціалізованої приватної мережі. Можна створити захищену підмережу на основі ненадійної мережі (як правило, Інтернету). Приклади захищених VPN включають: IPSec, OpenVPN та PPTP.
Довірена мережа VPN	Він використовується тоді, коли середовище передачі можна вважати надійним, і йому потрібно лише створити віртуальну підмережу в більшій мережі. Питання безпеки застаріло. Приклади таких рішень VPN включають: MPLS та L2TP. Якщо бути точним, ці протоколи перетворюють завдання безпеки в інші завдання, наприклад, L2TP, який зазвичай використовується спільно з IPSec.

Мережа VPN реалізується за допомогою спеціального набору програмного чи апаратного забезпечення. Дана реалізація забезпечує високу продуктивність та зазвичай має високу безпеку.

Мережа VPN як програмне рішення. Вони використовують персональні комп’ютери зі спеціальним програмним забезпеченням, яке забезпечує функціональність VPN.

VPN-мережа з інтегрованим рішенням. Функція VPN забезпечує складну фільтрацію мережевого трафіку, організацію мережевого екрану та якість обслуговування [8].

1.2.2 Класифікація VPN по типу технічної реалізації

Існує багато варіантів встановлення VPN. Вибираючи рішення, потрібно враховувати коефіцієнти продуктивності побудованої VPN мережі. Наприклад, якщо маршрутизатор все ще працює на потужності свого процесора, додавання тунелю VPN та застосування шифрування/розшифровки можуть перешкодити роботу всієї мережі, оскільки маршрутизатор не може обробляти простий трафік, не кажучи вже про VPN.

Досвід показує, що для створення VPN найкраще використовувати спеціальне обладнання, але якщо існують обмеження щодо методу, можна орієнтуватися на чисті програмні рішення. Розглянемо деякі варіанти побудови VPN. На основі першої версії брандмауера. Більшість сторонніх брандмауерів підтримують тунелювання даних та шифрування. Всі такі продукти базуються на тому, що трафік, що проходить через брандмауер, шифрується [9].

Модуль шифрування доданий до самого програмного забезпечення брандмауера. Недоліком цього методу є те, що продуктивність залежить від обладнання, на якому працює брандмауер. Використовуючи брандмауер на базі ПК, пам’ятайте, що це рішення можна застосувати лише до невеликих мереж, які передають невелику кількість інформації.

На малюнку 1.3 показано принципову схему побудови VPN на основі брандмауера.

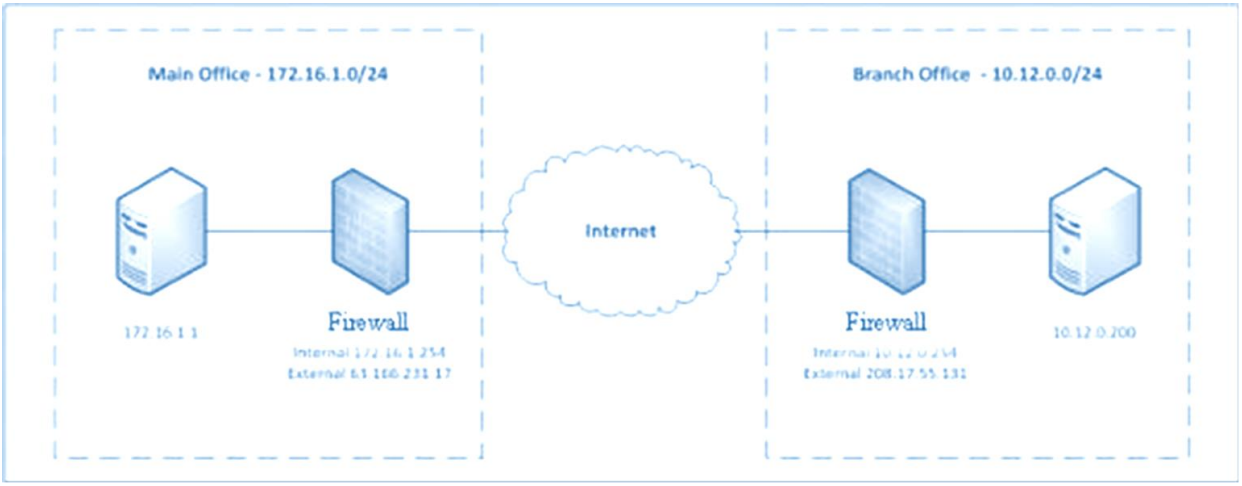

Рисунок 1.3 – Схема побудови VPN на базі брандмауэрів [10]

Другий варіант – побудувати мережу на основі маршрутизатора. Інший спосіб побудови VPN – це використання захищеного тунелю маршрутизатора (рис. 1.4). Оскільки вся інформація, що надсилається з локальної мережі, проходить через маршрутизатор, рекомендується розмістити на маршрутизаторі завдання шифрування.

Системне обладнання Cisco – приклад пристрою, який створює VPN на маршрутизаторі. Починаючи з програмного забезпечення IOS 11.3, маршрутизатори Cisco підтримують L2TP та IPSec. Окрім простого шифрування та передачі інформації, Cisco також підтримує інші функції VPN, такі як ідентифікація тунелів та обмін ключами [11].

Необов'язковий модуль шифрування ESA може бути використаний для покращення продуктивності маршрутизатора. Крім того, Cisco також випустив спеціальний пристрій VPN, який називається маршрутизатором доступу до VPN, який можна використовувати невеликим та середнім компаніям та великим організаціям.

Рисунок 1.4 – VPN на базі маршрутизаторів [12]

Ще один варіант на основі програмного забезпечення. Наступний спосіб створення VPN – це чисте програмне рішення. У цьому рішенні використовується спеціалізоване програмне забезпечення, яке працює на спеціалізованому комп'ютері і в більшості випадків діє як проксі-сервер. Комп'ютер із цим програмним забезпеченням може знаходитися за брандмауером.

Прикладом такого рішення є програмне забезпечення Digital AltaVista Tunnel. Під час використання цього програмного забезпечення клієнт підключиться до тунельного сервера, підтвердить його автентифікацію та обміняти ключі. Шифрування засноване на 56-бітному або 128-бітному ключі, отриманому в процесі з'єднання.

Крім того, зашифрований пакет даних інкапсулюється в інші пакети даних IP і потім надсилається на сервер. Крім того, програмне забезпечення кожні 30 хвилин генерує новий ключ, що значно покращує безпеку з'єднання.

Перевага тунелю AltaVista полягає в тому, що він простий в установці та експлуатації. Недоліками цієї системи можна вважати нестандартну архітектуру (власний алгоритм обміну ключами) та низьку продуктивність.

Мережеве рішення на базі системи Microsoft Windows NT. Для створення VPN Microsoft використовує протокол PPTP, інтегрований у Windows NT. Слід зазначити, що вартість цього рішення набагато нижча, ніж

вартість інших рішень. VPN на основі Windows NT використовує базу даних користувачів NT, що зберігається на первинному контролері домену (PDC).

Під час підключення до сервера PPTP користувачі отримуватимуть авторизацію за допомогою протоколів PAP, CHAP або MS-CHAP. Передані пакети даних інкапсульовані в пакети даних GRE / PPTP. Шифрування пакетів використовує нестандартний протокол шифрування Microsoft "точка-точка" і отримує 40 або 128-бітний ключ під час з'єднання.

Недоліками цієї системи є відсутність перевірок цілісності даних та неможливість змінити ключ під час з'єднання. Перевагою є легка інтеграція з Windows і низька вартість.

1.2.3 Протокол PPTP

Протокол тунелювання «точка-точка», розроблений Microsoft за підтримки інших компаній, є розширенням протоколу (PPP) та використовується для створення захищеного віртуального каналу, коли віддалені користувачі отримують доступ до локальної мережі через Інтернет. У випадку, коли віддалений комп'ютер безпосередньо підключений до загальнодоступної мережі, а також, коли він підключений до загальнодоступної мережі через телефонну лінію постачальника, на рівні зв’язку моделі OSI встановлюється тунель.

Протокол подано в інженерну цільову групу, як конкурент стандартного протоколу. Стандартний протокол використовується для створення захищеного каналу під час доступу віддалених користувачів до локальної мережі через загальнодоступну мережу, головним чином через Інтернет. RTDR набув статусу проекту Інтернет-стандарту, але, хоча він був широко прийнятий, стандарт ще не затверджений. Наразі робоча група IETF розглядає можливість прийняття протоколу тунелювання рівня 2 як стандарт, який об'єднує найкращих партнерів протоколу PPTP та подібних протоколів переадресації рівня 2, передбачених Cisco Systems.

Для віддалених користувачів, підключених до Служби віддаленого доступу (RAS) в локальній мережі через загальнодоступну мережу IP, PPTP тунельні пакети повідомлень для імітації часу перебування користувача у внутрішній мережі. Дані, що проходять через тунель, передаються за допомогою стандартного протоколу віддаленого доступу PPP, який використовується не тільки для підключення віддаленого комп'ютера користувача до RAS постачальника Інтернет-послуг в протоколі PPTP, але і для зв'язку з RAS локальної мережі через тунель [13].

Для пакетних даних впливатимуть IP-пакети, що містять капсульовані пакети PPP. У свою чергу, капсульовані пакети даних PPP включають зашифровані капсульовані пакети вихідних даних (IP, IPX або NetBEUI) для зв'язку між віддаленим комп'ютером та локальною мережею.

Програмний пакет, що циркулює в сесії RTTP, має таку структуру, як показано на малюнку 1.5. Розглянемо певний перелік:

– Заголовки посилальних шарів, що використовуються в Інтернеті, такі як заголовки кадрів Ethernet.

– IP-заголовок.

– Заголовок GRE (загальна інкапсуляція маршрутизації).

– Вихідний пакет PPP, що включає пакет IP, IPX або NetBEUI.

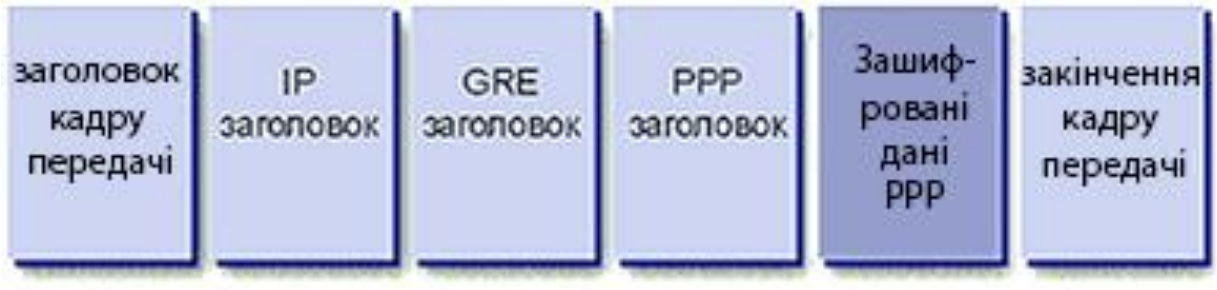

Рисунок 1.5 – Структура даних для пересилання

Крім того, PPTP інкапсулює PPP-кадри в загальний пакет інкапсуляції маршрутизації (GRE), що належить мережевому рівню. GRE інкапсулює

мережевий шар. Однак GRE не може налаштувати сеанси та захистити дані від шкідливих користувачів. Це використовує функцію PPTP для створення лише поля PPTP, де обмеження капсуляції захистом GRE є обмеженим

Після інкапсуляції кадру PPP у кадр у заголовку GRE він буде інкапсульований у кадр у заголовку IP. IP-заголовок – це адреса відправника та одержувача пакету даних. Нарешті, PPTP додає заголовок PPP та End.

Система відправника посилає повідомлення через тунель. Система-одержувач видаляє всі службові заголовки, залишаючи тільки дані PPP.

Оскільки вся ідея віддаленого доступу полягає в підключенні клієнтського комп'ютера до серверного комп'ютера через мережу, клієнт використовує службу віддаленого доступу Windows NT (RAS) для ініціювання PPTP-з'єднання для встановлення PPP-зв’язку з Інтернетом, провайдером. Потім, коли сервер, підключений до Інтернету, використовується для активації PPP-з'єднання і виступає в ролі сервера RAS, клієнт буде використовувати RAS для встановлення другого з'єднання.

На цей раз у полі номера телефону вказується IP-адреса (або ім'я домену), і клієнт використовує порт VPN для підключення замість COM-порту (у цьому процесі порти VPN налаштовуються на клієнті та сервері (установка PPTP). Дозволяє вказати IP та розпочати передачу запитів на сервер на початку сеансу. Клієнт чекає, коли сервер підтвердить ім'я користувача та пароль та відповідь на повідомлення для встановлення з'єднання.

У цей момент канал PPTP починає працювати, і клієнт може почати передавати пакети даних сервера через тунель. Оскільки вони можуть бути пакетами IPX та NetBEUI, сервер може виконувати звичні для них процедури безпеки. Основою зв'язку PPTP є послідовність керуючих повідомлень, які керують підключенням до PPTP-встановлення та обслуговування тунелів [14].

Повне з'єднання PPTP складається лише з одного TCP / IP-з'єднання, яке вимагає передачі команд, щоб тримати його відкритим у міру просування транзакції. Протокол PPTP визначає дві схеми його реалізації. Перше рішення призначене для підтримки захищеного каналу між сервером віддаленого

доступу ISP та маршрутизатором кордону корпоративної мережі, як показано на рис. 1.6.

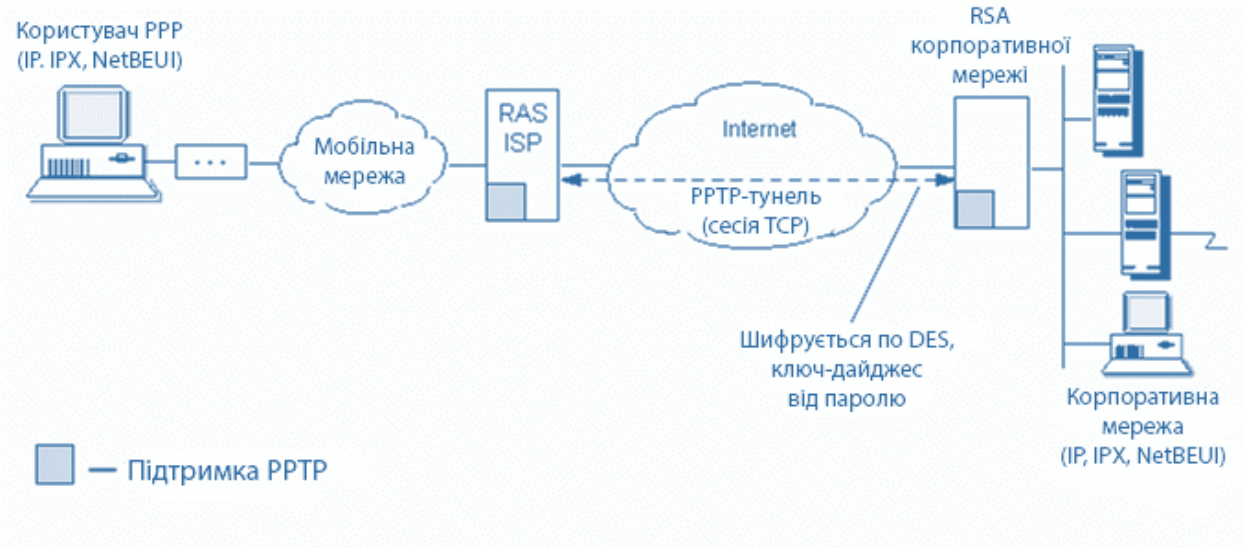

Рисунок 1.6 – Канал постачальника захищеного мережевого маршрутизатора на основі протоколу PPTP [15]

У цьому варіанті віддалений комп'ютер не потребує підтримки PPTP. Він використовує стандартний протокол PPP для зв'язку з сервером віддаленого доступу RAS, встановленим в ISP, і спочатку аутентифікується з постачальником. ISP RAS повинен підтримувати PPTP.

На ім'я користувача RAS ISP повинен знайти в базі облікових даних користувачів IP-адреса маршрутизатора, що є маршрутизатором корпоративної мережі даного користувача. З цим маршрутизатором RAS ISP встановлює сесію по протоколу PPTP. Протокол PPTP визначає деяку кількість службових повідомлень, якими обмінюються взаємодіючі сторони, службові повідомлення передаються по протоколу TCP.

RAS ISP передає ідентифікатор компанії до мережевого маршрутизатора компанії, а потім маршрутизатор використовує протокол CHAP для повторної аутентифікації користувача. Якщо користувач передає другу аутентифікацію (прозору для нього), провайдер RAS надсилає йому повідомлення про PPP, і

користувач починає надсилати свої дані в RAS ISP через IP, IPX або NetBIOS і пакує їх у рамки PPP.

RAS ISP інкапсулює кадр PPP в пакет даних IP, вказує адресу призначення адреси маршрутизатора межі та вказує власну IP-адресу як адресу джерела. PPP-пакети шифруються за допомогою ключа, що зберігається в облікових записах ISP RAS для аутентифікації CHAP.

Внутрішній веб-сервер компанії також не потребує підтримки PPTP, оскільки маршрутизатор кордону витягує кадри PPP з IP-пакетів та надсилає їх по мережі в необхідному форматі (IP, IPX або NetBIOS).

1.2.4 Протоколи DNS та DHC

Система доменних імен (загально відома як DNS) та протокол конфігурації динамічного хоста (також відомий як DHCP) являють собою дві важливі області TCP / IP в мережі. DNS відповідає за перетворення імен хостів до IP-адрес, а DHCP відповідає за призначення унікальних динамічних IP-адрес та відповідних масок підмережі та шлюзів за замовчуванням для роботи комп'ютерів у певній серверній мережі.

Оскільки це динамічна адресація, комп'ютер може мати різну IP-адресу кожного разу, коли він підключається до мережі, до якої належить, без втручання адміністратора UNIX. Використовуючи цю функцію DHCP, кожному новому комп'ютеру, доданому до мережі, автоматично буде призначена унікальна IP-адреса. Сервер DHCP значно спрощує налаштування мережі та використовується в більшості точок бездротового доступу та провідних маршрутизаторів Ethernet.

У мережі DHCP-сервер управляє інформацією про пул IP та шлюзом за замовчуванням, даними DNS та іншою інформацією про конфігурацію клієнтської мережі. Коли новий комп'ютер підключений до мережі, що підтримує DHCP, він відправить запит на сервер DHCP, щоб запитати всю необхідну інформацію [16].

Коли запит надійде до сервера DHCP, він надасть новому комп'ютеру нову IP-адресу та оренду - часовий інтервал, протягом якого комп'ютер може використовувати IP-адресу та інші дані конфігурації. Весь процес буде здійснюватися відразу після запуску нового комп’ютера, і щоб успішно завершити процес, процес повинен бути завершений перед початком зв'язку на основі IP з іншими хостами в мережі. Розглянемо рисунок 1.7.

Рисунок 1.7 – Схема взаємодії по протоколу DHCP [17]

Залежно від конфігурації DHCP-сервер може працювати трьома способами. Розглянемо таблицю з більш детальною інформацією по даній темі (табл.1.3).

Таблиця 1.3 Конфігурації DHCP-серверів

№	Назва	Опис
1.	Динамічний розподіл	Коли сервер DHCP налаштований на використання динамічного розподілу, він вказує, що він використовує політику оренди. Якщо IP-адреси, виділені з наявних пулів, більше не використовуються, помилка буде передана назад до пулу, який зробив пул доступним для інших користувачів.

Продовження таблиці 1.3

№	Назва	Опис
2.	Автоматичний розподіл	Метод автоматичного розподілу дуже схожий на метод динамічного розподілу - після підключення клієнта сервер DHCP негайно надасть йому IP-адресу з пулу IP. Однак, використовуючи автоматичний розподіл, сервер DHCP зберігає базу даних попередніх IP-грантів і намагається надати клієнту останню IP-адресу, яку він використав, якщо така є.
3.	Статичний розподіл	За допомогою статичного розподілу DHCP-сервер зберігатиме всі MAC-адреси локальної мережі клієнта в базі даних і лише надає клієнту IP-адресу, якщо його MAC-адреса знаходиться в базі даних. Таким чином, ви можете гарантувати, що клієнт буде отримувати одну і ту ж IP-адресу щоразу.

Великі мережі зазвичай діляться в підмережах, щоб запобігти затримці, спричиненій занадто великою кількістю пристроїв, які намагаються отримати доступ до середовища передачі. У цих випадках мережа все ще може використовувати лише один сервер DHCP, але для кожної підмережі потрібен ретрансляційний пристрій.

Структура пакетів усіх типів повідомлень DHCP містить поле GIADDR, яке заповнює реле. Це власна адреса ретранслятора, тому коли повідомлення, отримане від клієнтського ретранслятора, буде перенаправлено на сервер DHCP, сервер буде знати, куди надсилати відповідь та діапазон адреси, призначеної на повторювачі підмережі. [7].

1.3 Вибір засобів реалізації

Останнім часом список сервісів і ресурсів які виросли в інтернеті безліч. Інтернет перетворився з одноманітних статичних сторінок в потужний інструмент інтерактивності і спілкування з кінцевими користувачами. У зв'язку з цим веб-додатки в даний час набули небувалу популярність, тому що вони пропонують масу важливих переваг, які відсутні в звичайних додатках. Дивлячись на все, що відбувається в індустрії програмного забезпечення відбувається великий розвиток веб-додатків як окремої ланки у всьому ланцюжку даного напрямку, і це не рахуючи швидкого приходу таких технологій як CSS3 і HTML5 [18].

З звичайними додатками все не так райдужно, вони звичайно розвиваються, але не так бурхливо як веб-технології. Велика безліч компаній переходять зі звичайних додатків на веб-додатки, саме тому що бачать в них майбутнє, готові користуватися ними і перейти на них з звичайних додатків, а за цим слідує і капіталовкладення, що дає потужний поштовх технологій. Саме тому поло вирішено реалізувати поставлену задачу у вигляді веб-додатку.

HTML – це стандартизована мова розмітки документів у Всесвітній павутині. Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML). Мова HTML інтерпретується браузерами; отриманий в результаті інтерпретації форматований текст відображається на екрані монітора комп'ютера або мобільного пристрою [19].

CSS – це формальна мова, яка використовується для опису зовнішнього вигляду документів, написаних мовою розмітки. Розробники веб-сайтів використовують CSS для встановлення шрифтів, їх макета на сторінці та інших принципів того, як виглядають блоки веб-сторінок. Основним завданням створення CSS є стилістичне відображення контенту, тим самим збільшуючи доступність документа та забезпечує більшу гнучкість та можливість керування відображенням веб-сторінок та зменшення складності

їх структури. CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript [20].

JavaScript – динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript є об'єктно-орієнтованою мовою програмування, але оскільки вона використовує прототипи (швидко реалізують основні функції для аналізу системи), порівняно з традиційними об'єктно-орієнтованими, існує багато функціональних мов програмування для прямого виконання мовного коду. Крім того, JavaScript має перелік атрибутів, притаманних іншим функціональним мовам, а саме: функції як об'єкти, об'єкти як списки та анонімні функції.

Хоча мова JavaScript по синтаксису схожа на мову C, вона все ж має ряд відмінностей:

– Використовуються об'єкти зі здатністю самоаналізу.
– Функції як об'єкти класу.
– Автоматичне приведення типів.
– Анонімні функції.

Як недоліки мови JavaScript можна виділити наступні:

– Немає можливості регулювати області.
– Відсутність інтерфейсу як такого.
– використання стандартних інтерфейсів доступу до Web-серверів і баз даних [21].

Vuetify є бібліотекою для розробки інтерфейсів і активно розробляється з 2016 року. Vuetify розроблений точно у відповідності зі специфікацією Material Design. Кожен компонент розроблений вручну, щоб надати найкращі

інструменти для користувача інтерфейсу. Розробка не зупиняється на основних компонентах, описаних в специфікації Google.

Для розробки додатку було обрано мови програмування HTML, CSS і JavaScript так як вони є най популярнішими мовами для веб розробки і досить легкі у вивченні. Для спрощення розробки було обрану бібліотеку Vuetify.

1.4 Постановка задачі

Проаналізувавши літературні дані, мету роботи можна сформулювати наступним чином: необхідно розробити веб-орієнтовану інформаційну систему, графічний інтерфейс якої буде дозволяти автоматично налаштувати інтерфейси роутерів та динамічну для підтримки роботи VPN мережі. Розробка повинна забезпечувати зручне перенесення згенерованого коду налаштувань в симулятор Cisco packet tracer та реальне обладнання Cisco. Програмне забезпечення повинно дозволяти початківцям успішно налаштовувати VPN мережі, не вимагаючи на початковому етапі знання команд конфігурації роутерів Cisco. Інтерфейс має бути зрозумілим навіть користувачу, що не має спеціальних навичок та досвіду у роботі з подібними інтерфейсами. Для створення графічного інтерфейсу необхідно створити веб-сторінку, на якій можна буде ввести вхідні дані та в результаті отримати налаштування, які можна скопіювати та внести в симулятор Cisco packet tracer. В результаті буде отримано налаштований маршрутизатор, а далі й мережу в цілому. Постановка задачі:

1. Конфігурація мережі на базі роутерів Cisco та симулятора Cisco packet tracer.
2. Розробка графічного інтерфейсу налаштування протоколів динамічної маршрутизації.
3. Тестування розробленої веб-орієнтованої інформаційної системи в симуляторі Cisco packet tracer та на реальному обладнанні Cisco.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

2.1 Побудова віртуальної VPN мережі

В даний час технології віртуалізації є невід'ємною складовою IT-світу. Крім промислового застосування технології віртуальних машин, що дозволяє скоротити сукупну вартість володіння IT-інфраструктурою, про що вже кілька років пишуть всі кому не лінь, технологія також широко використовується для вивчення або тестування системного, мережевого і прикладного програмного забезпечення.

Мережеві інженери і адміністратори використовують різні інструменти для проектування, моніторингу або аналізу комп'ютерних систем. Що б не експериментувати на реальних мережах (що може спричинити збій або виходом з ладу мережевої інфраструктури) системні адміністратори для цього використовують інструменти мережевого моделювання. Однією з таких програм моделювання являється Cisco packet tracer.

Cisco Packet Tracer розроблений компанією Cisco і рекомендований використовуватися при вивченні телекомунікаційних мереж і мережевого устаткування, а також для проведення уроків з лабораторних робіт у вищих закладах.

Основні можливості Packet Tracer:

- Дружній графічний інтерфейс (GUI), що сприяє до кращого розуміння організації мережі, принципів роботи пристрою;
- Можливість змоделювати логічну топологію: робочий простір для того, щоб створити мережі будь-якого розміру на CCNA-рівні складності;
- моделювання в режимі real-time (реального часу);
- режим симуляції;
- Багатомовність інтерфейсу програми: що дозволяє вивчати програму на своїй рідній мові [22].

- вдосконалене зображення мережевого обладнання зі здатністю додавати / видаляти різні компоненти;
- наявність Activity Wizard дозволяє мережевим інженерам, студентам і викладачам створювати шаблони мереж і використовувати їх в подальшому.
- проектування фізичної топології: доступне взаємодія з фізичними пристроями, використовуючи такі поняття як місто, будівля, стійка і т.д .;

Проаналізувавши переваги даного програмного забезпечення для побудови віртуальної VPN мережі було обрано саме Cisco Packet Tracer.

В першу чергу було змодельовано VPN мережу. Також налаштовані інтерфейси роутерів, свічів та протоколи динамічної маршрутизації. Для моделювання використовувались роутери Cisco 2811 і свічі 2960.

Першими кроками створюємо локальні сеті

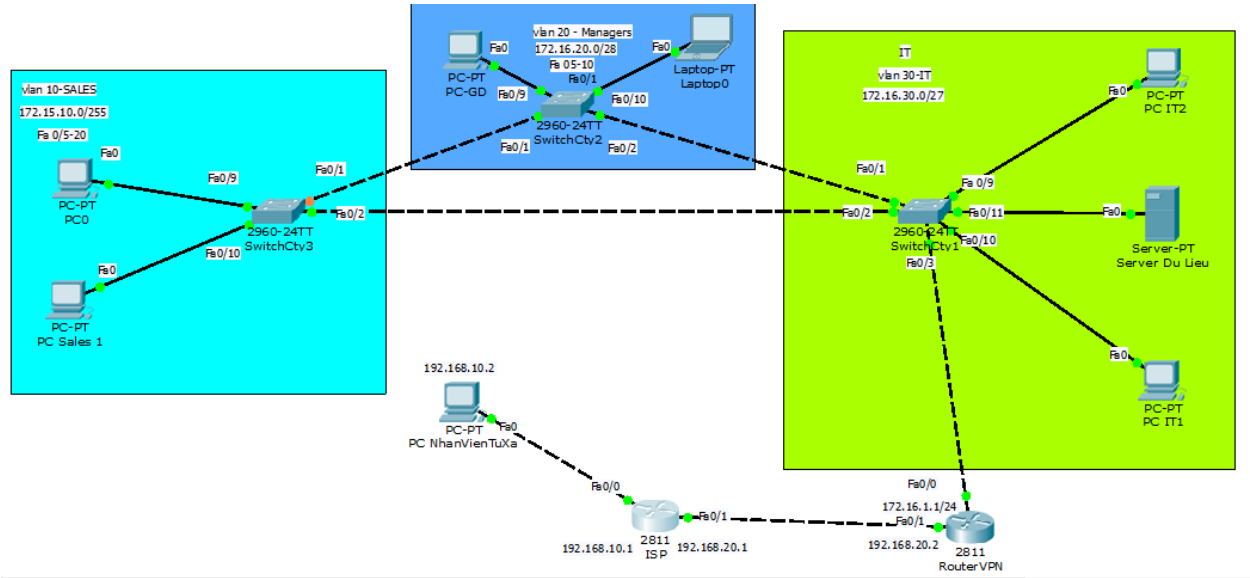

Рисунок 2.1 – Побудована VPN мережа

В першу чергу налаштовуємо IP конфігурацію на ПК клієнта:

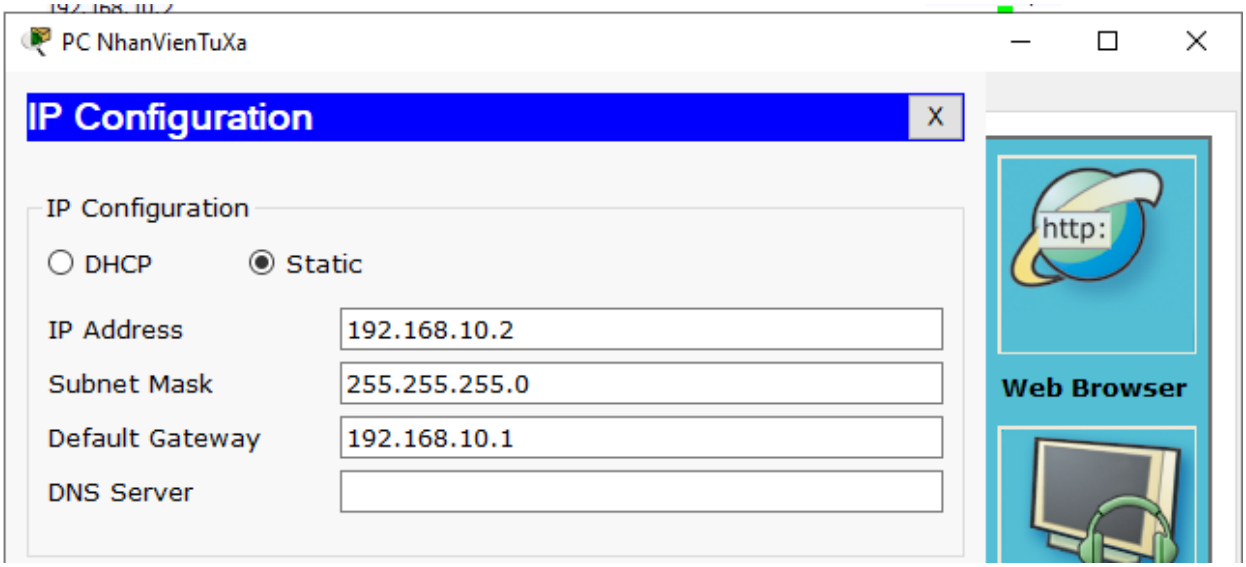

Рисунок 2.2 – Конфігурації IP

Потім на роутері “ISP” задаємо IP адреси на порти FastEthernet0/0 та FastEthernet 0/1.

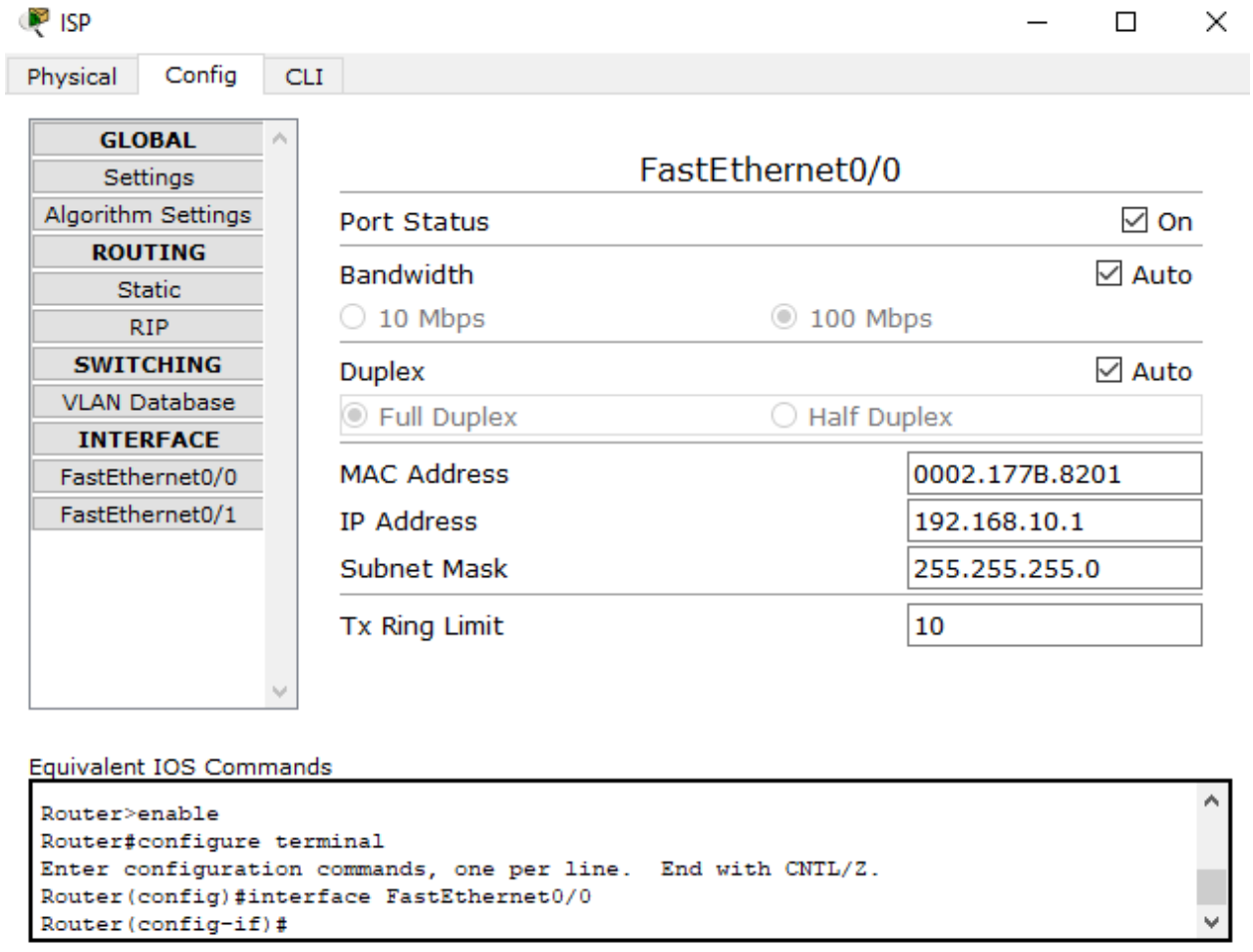

Рисунок 2.3 – Конфігурації FastEthernet0/0 на роутері “ISP”

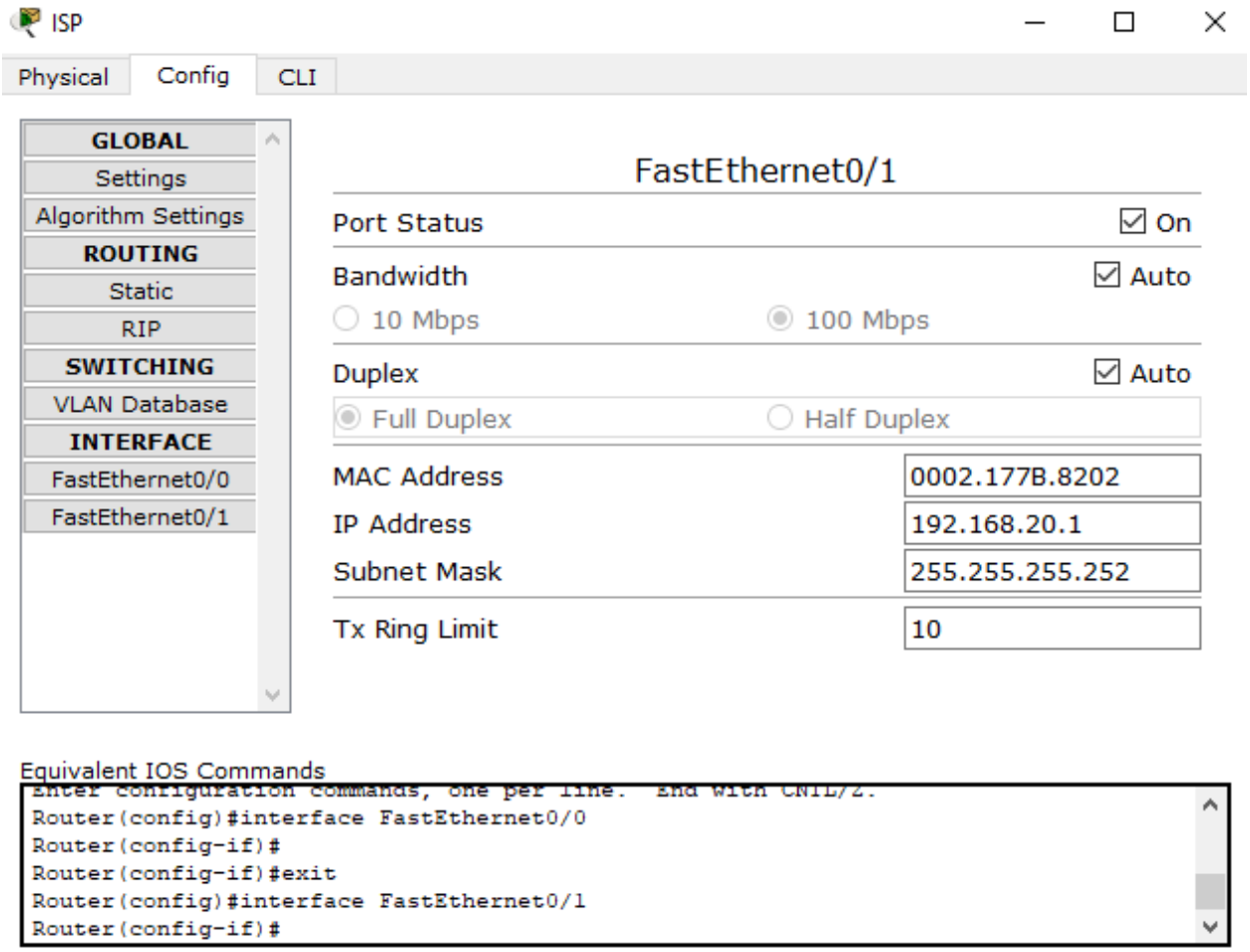

Рисунок 2.4 – Конфігурації FastEthernet0/1 на роутері “ISP”

На роутері «RouterVPN» задаємо лише FastEthernet0/1 та надсилаємо конверт з ПК клієнта до роутеру «RouterVPN», для перевірки підключення і збереження адрес роутерів.

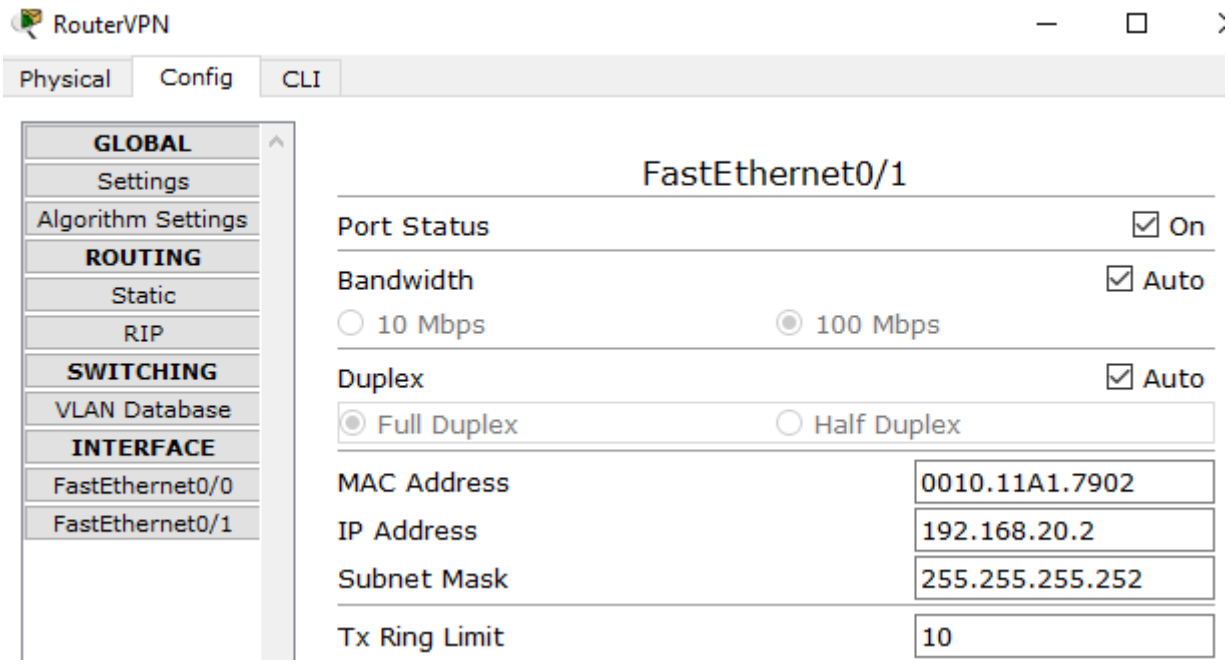

Рисунок 2.5 – Конфігурації FastEthernet0/1 на роутері “RouterVPN”

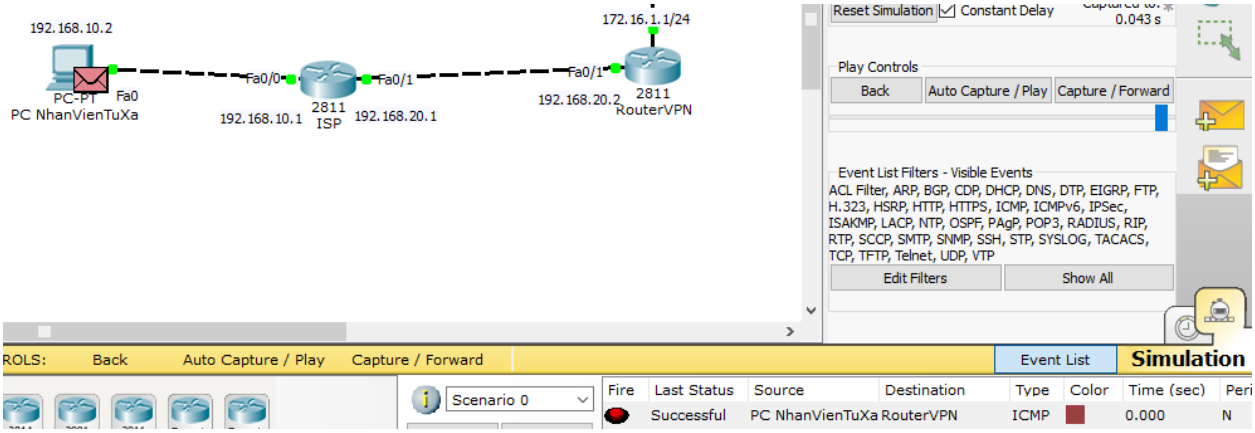

Рисунок 2.6 – Перевірка роботи роутеру

Наступним кроком буде налаштування свічів (Switches) для кожного (vlan). Задаємо кожному свічу по vlan та вказуємо діапазон портів вільного доступу.

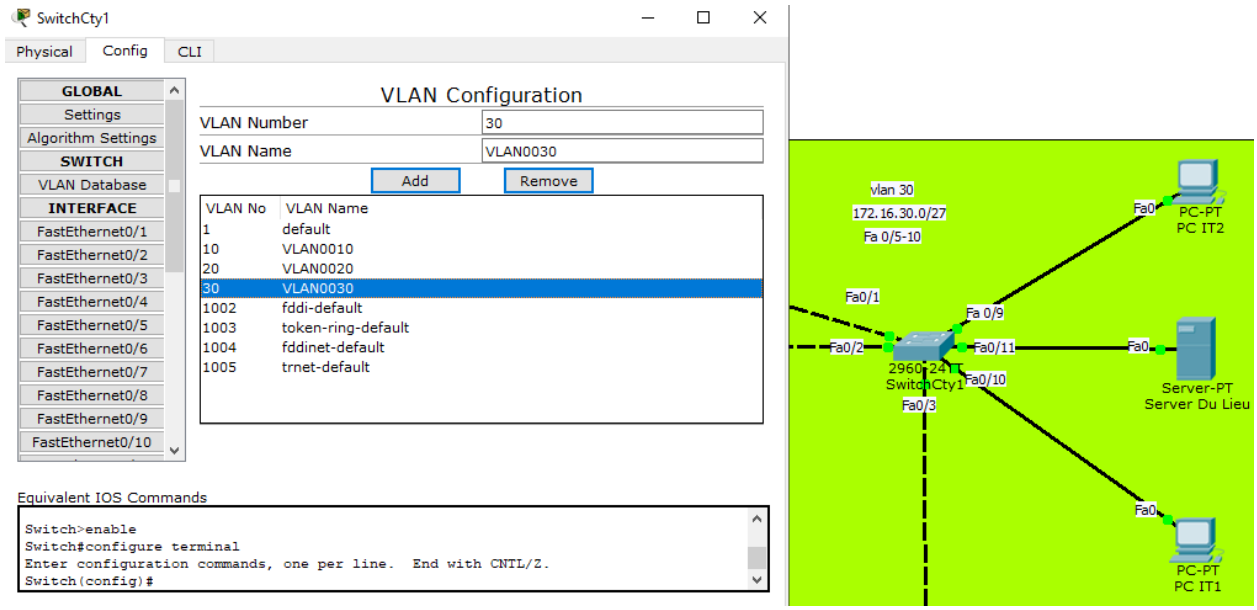

Рисунок 2.7 – Налаштування Switch

Потім на роутері «RouterVPN» додаємо IP адреси вільних vlan і маршрути передачі даних.

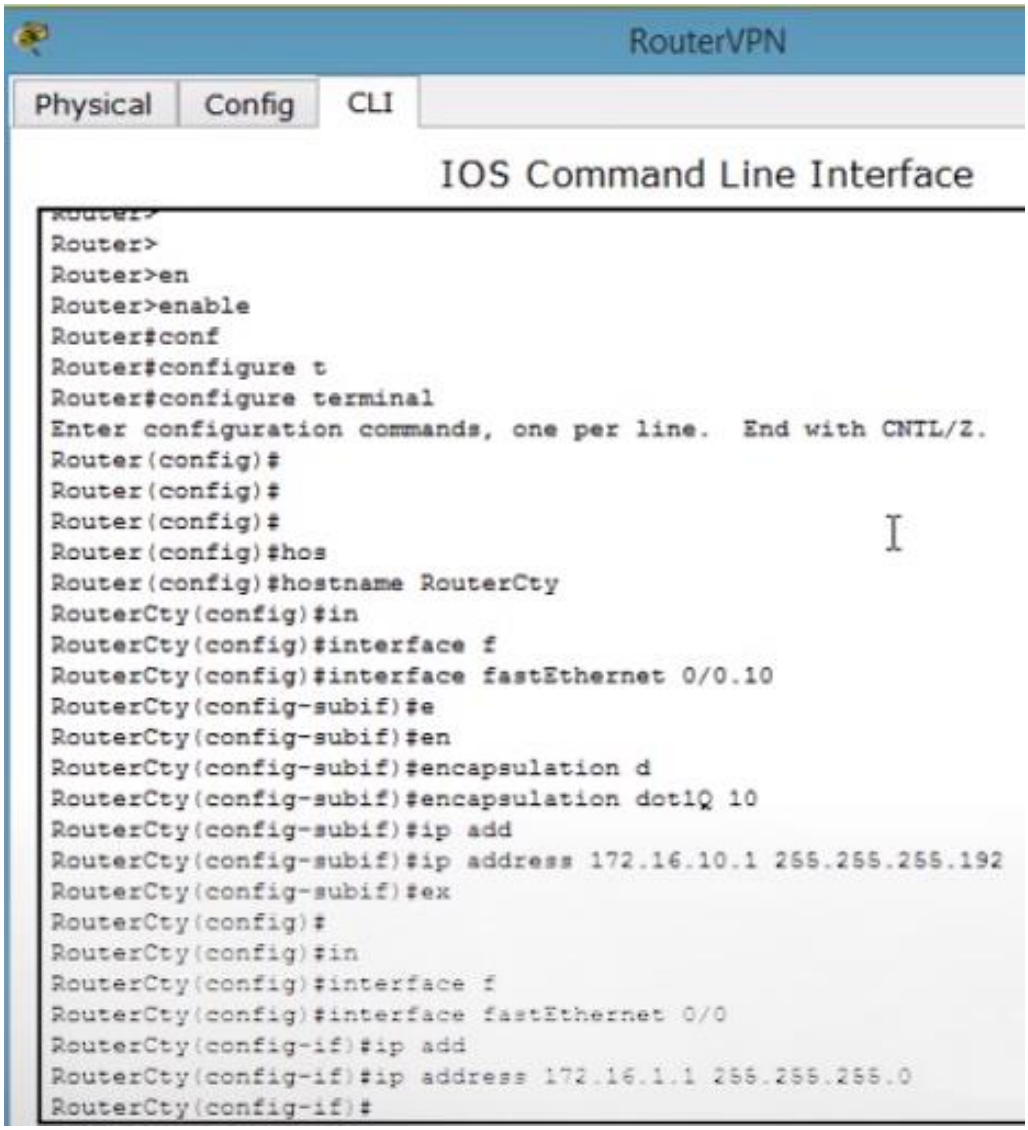

Рисунок 2.8 – Налаштування vlan на «RouterVPN»

Після того як налаштували маршрути для vlan потрібно налаштувати DHCP на роутері «RouterVPN».

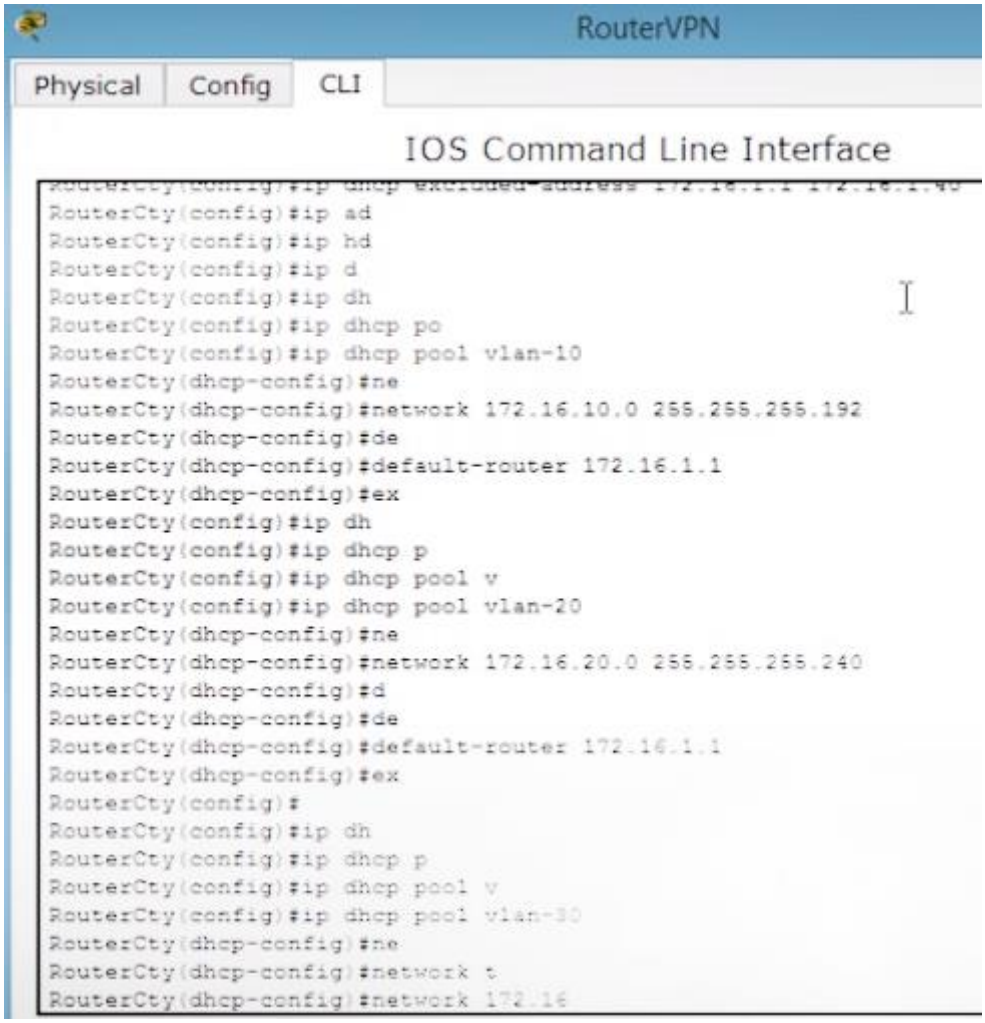

Рисунок 2.9 – Налаштування DHCP на «RouterVPN»

Після налаштування DHCP вмикаємо на кожному ПК в IP Configuration і перемикаємо режим на DHCP.

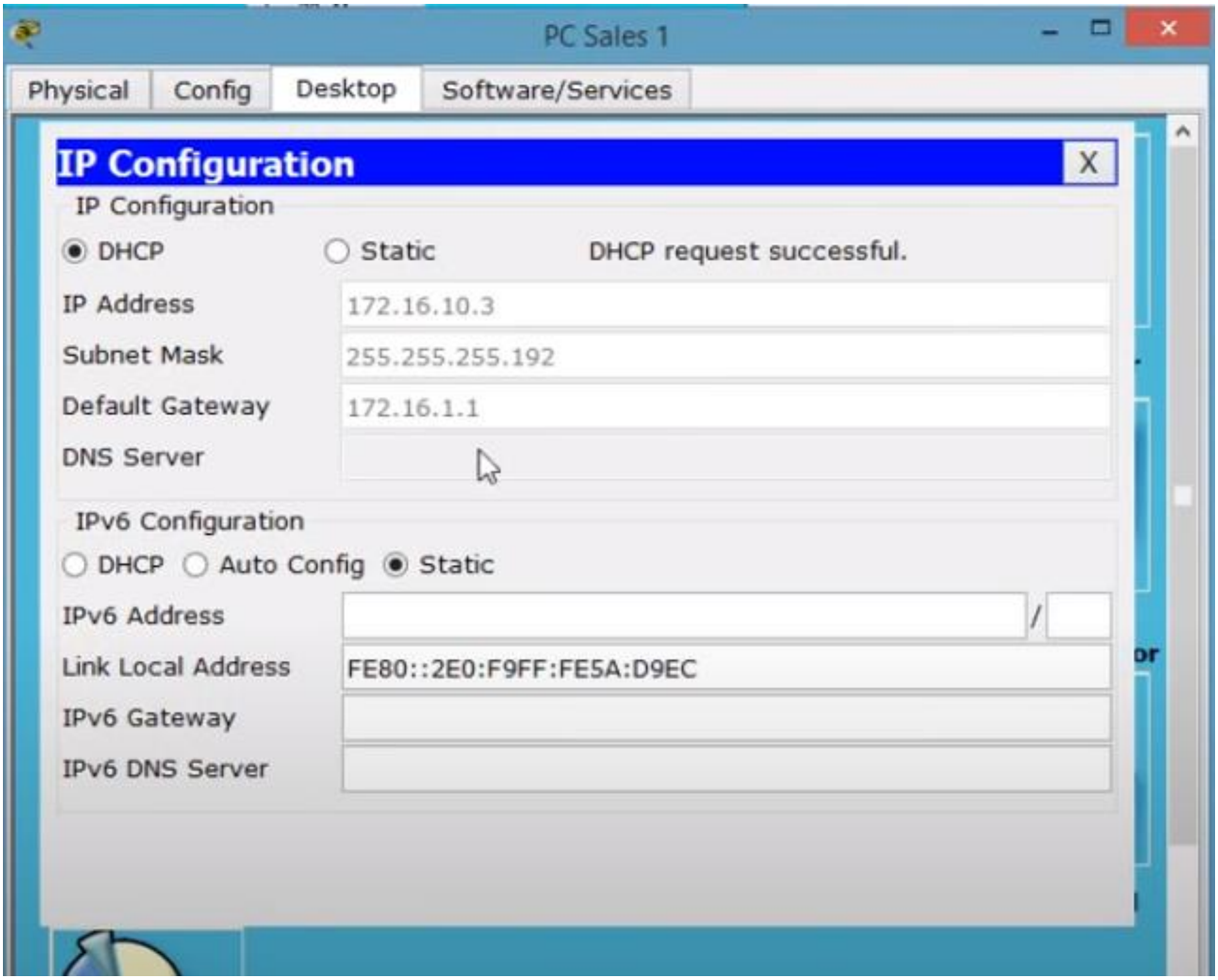

Рисунок 2.10 – Налаштування DHCP на ПК

Останнім налаштуванням на «RouterVPN» буде задання клієнтських даних для доступу до віртуальної мережі.

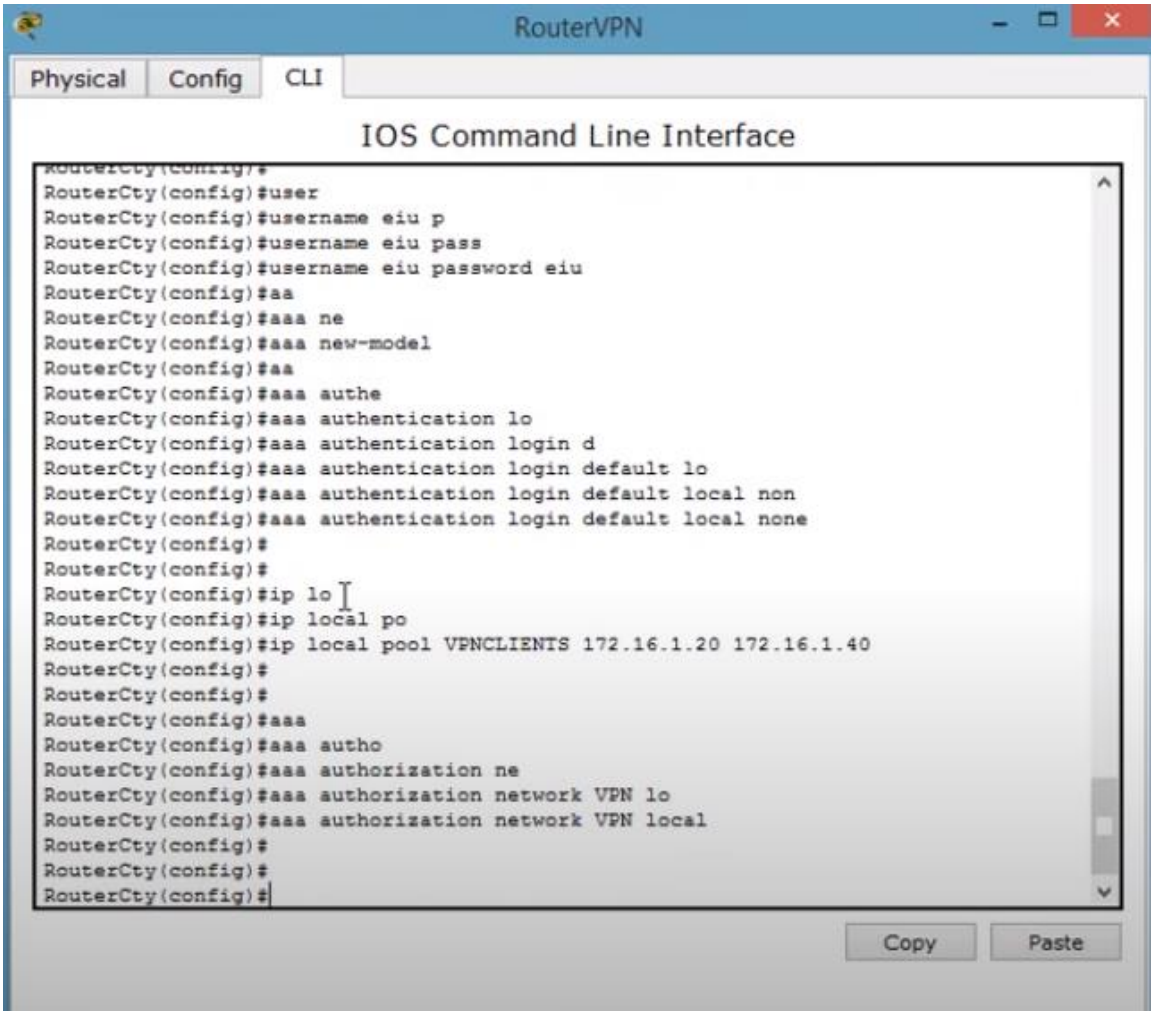

Рисунок 2.11 – Налаштування доступу

Тепер щоб ввімкнути VPN достатньо зайти в конфігурацію VPN та вести клієнтські дані.

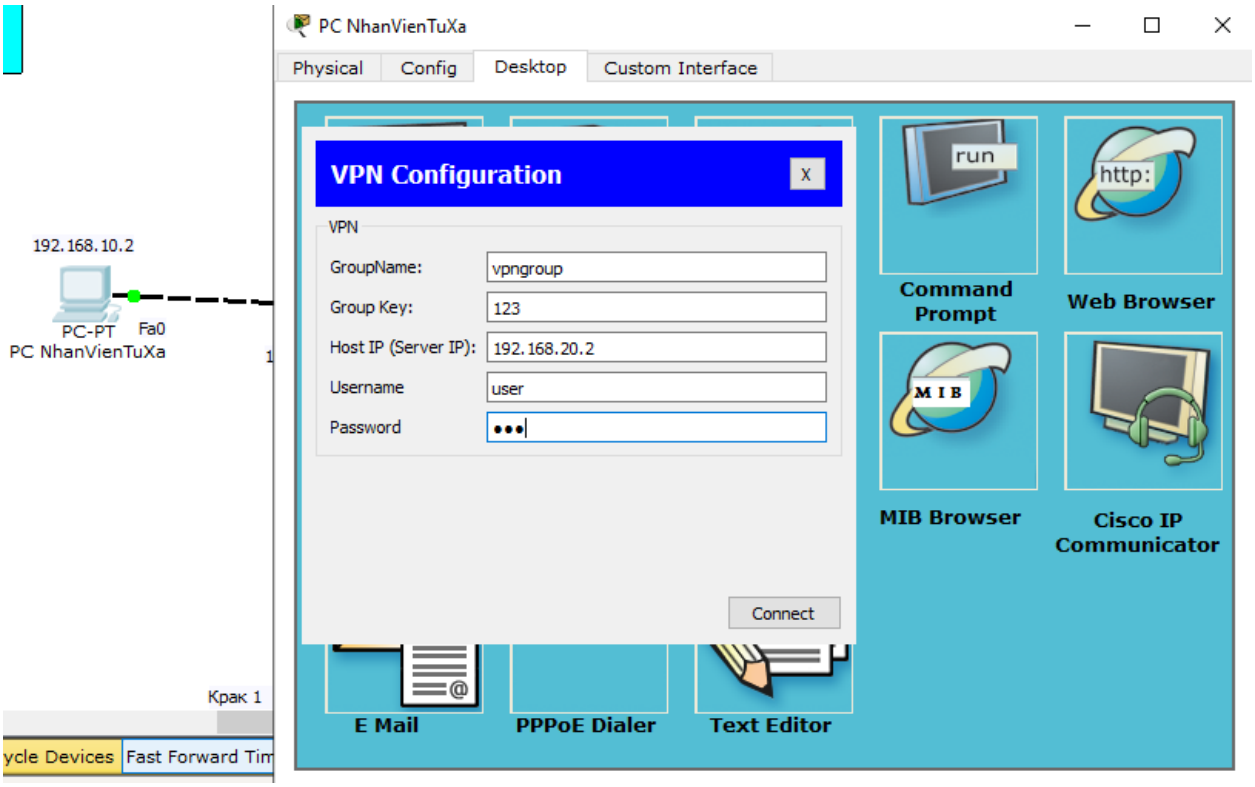

Рисунок 2.12 – Включення VPN на ПК клієнта

3 РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ

Для реалізації поставленої задачі було потрібно вирішити наступні проблеми:

– Розробити простий та інтуїтивний інтерфейс.
– Реалізувати перевірку правильності вхідних даних.

3.1 Програмна реалізація

Для вирішення поставленої завдання було визначено ряд задач:

– Визначення функцій сайту;
– Вибір стилістичного оформлення;
– Вибір мови програмування, яка задовольняє функції сайту;
– Створення графічних елементів сайту;
– Верстка сайту.
– Тестування на побудованій віртуальній VPN мережі.

Написання сайту починається із створення «скелету» з використанням мови розмітки HTML. Після чого за допомогою мови каскадних стилів CSS формується зовнішній вигляд веб-сайту і надається візуальна форма елементів для зручного користування. Із використанням фреймворку Vuetify даний етап виконується в рази швидше і легше. Також даний фреймворк надає сайту адаптивність під різні платформи, а також кросбраузерність, що надає можливість користуватися сайтом використовуючи різні версії браузерів.

Інтерфейс сайту буде складатись із таких елементів:

– Форма для налаштування інтерфейсів роутерів;
– Форма для налаштування мережевих комутаторів switch;
– Форма для налаштування VPN;
– Форма для налаштування DHCP;
– Форма для налаштування політики користувачів;

– Можливість формувати на базі вхідних даних готовий код і відображати на екран.

Більш детальну інформацію та код можна розглянути у Додатках.

3.2 Використання програмного додатку

Додаток складається з вкладок які містять форми для налаштування інтерфейсів мережі, поле результату в яке буде записуватись згенерований код для налаштування VPN мережі (рис. 3.1).

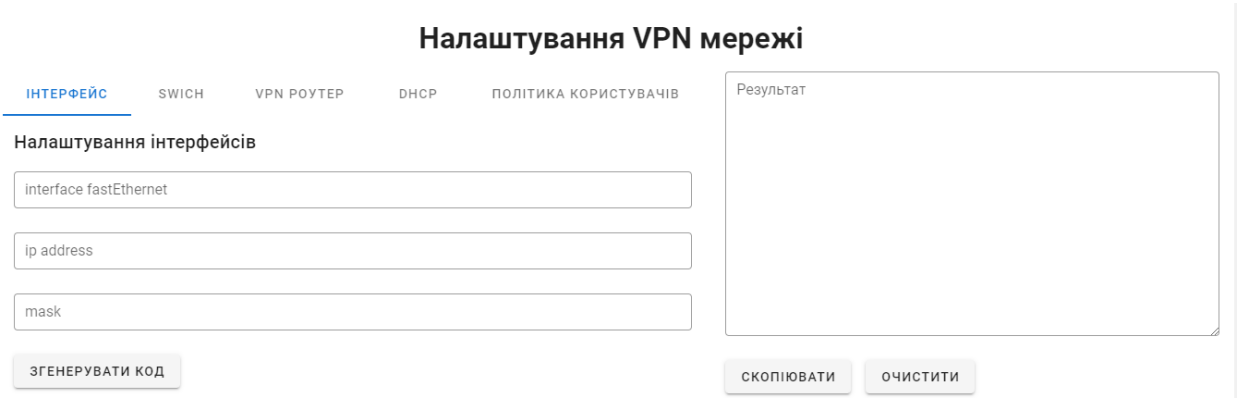

Рисунок 3.1 – Інтерфейс додатку

Вкладка «Інтерфейс» містить форму з полями fastEthernet, ip адреса і маска які необхідні для налаштування інтерфейсів портів роутерів (рис. 3.2).

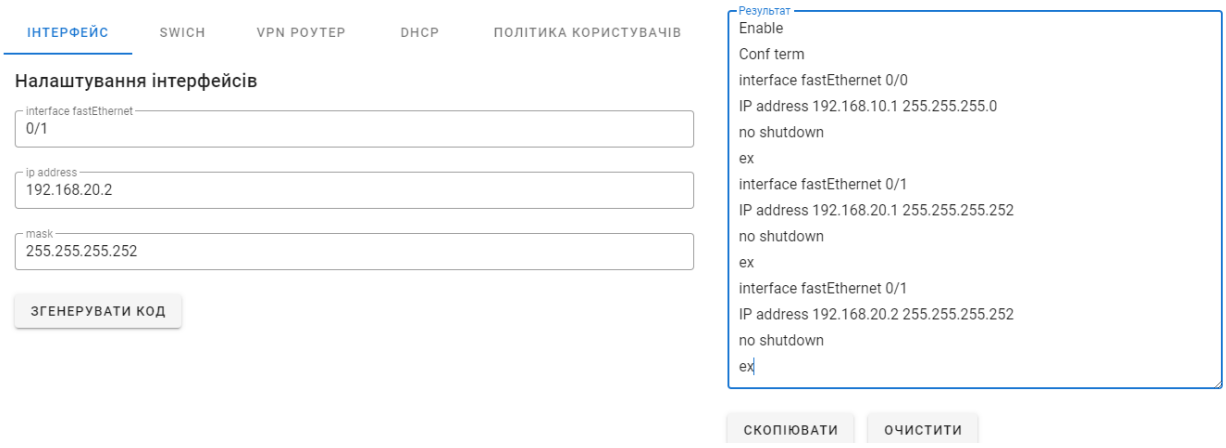

Рисунок 3.2 – Приклад згенерованого коду для налаштування інтерфейсу

Вкладка «Switch» містить форму з полями fastEthernet, Vlan і кнопки для перемикання режиму порту які необхідні для налаштування комутатора Switch (рис. 3.3).

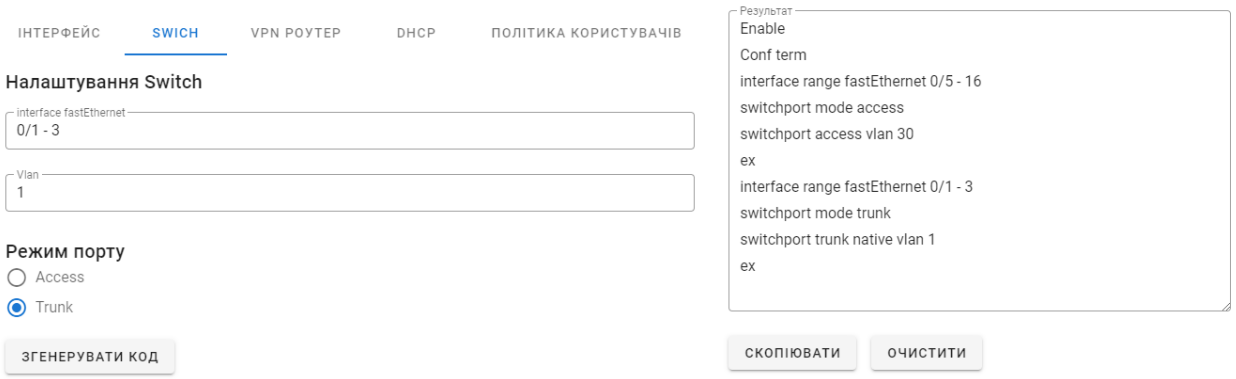

Рисунок 3.3 – Приклад згенерованого коду для налаштування Switch

Вкладка «VPN роутер» містить форму з полями fastEthernet, тип інкапсуляції, Vlan ip адресу і маску мережі які необхідні для налаштування комутатора VPN на головному роутері (рис. 3.4).

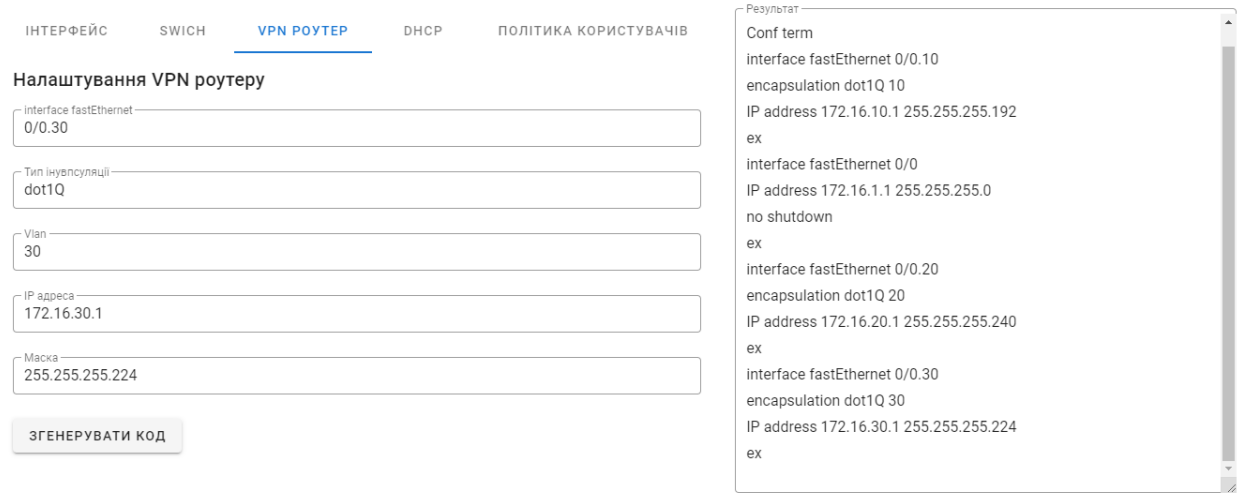

Рисунок 3.4 – Приклад згенерованого коду для налаштування VPN роутера

Вкладка «DHCP» складається із двух частин, а саме поля з діапазоном ip адрес і налаштуваннями для Vlan [9]. Для налаштування Vlan необхідно

заповнити поля: номер віртуальної локальної мережі, ip адресу, маску і роутер за замовчуванням (рис. 3.5). За необхідністю можна додати мережу Vlan натиснувши відповідну кнопку.

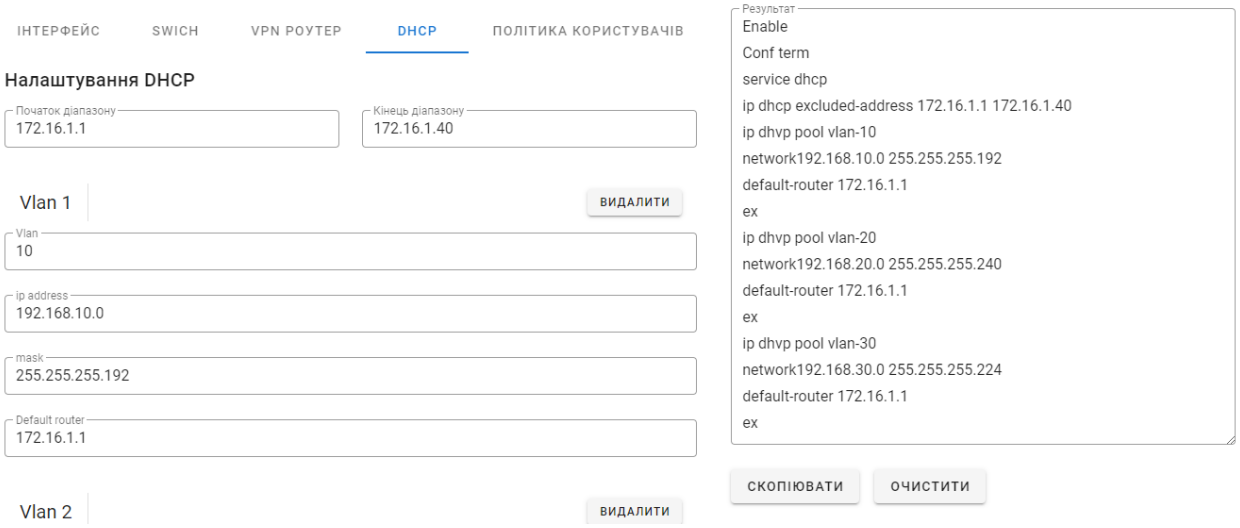

Рисунок 3.5 – Приклад згенерованого коду для налаштування DHCP

Вкладка «Політика користувачів» складається із двух частин, а саме поля для налаштування pool: ім’я та пароль користувача, назва моделі, назва pool і діапазон ip адрес. Друга частина має поля для налаштування групової політики: назва групи, маску мережі і тип шифрування (рис. 3.6-7).

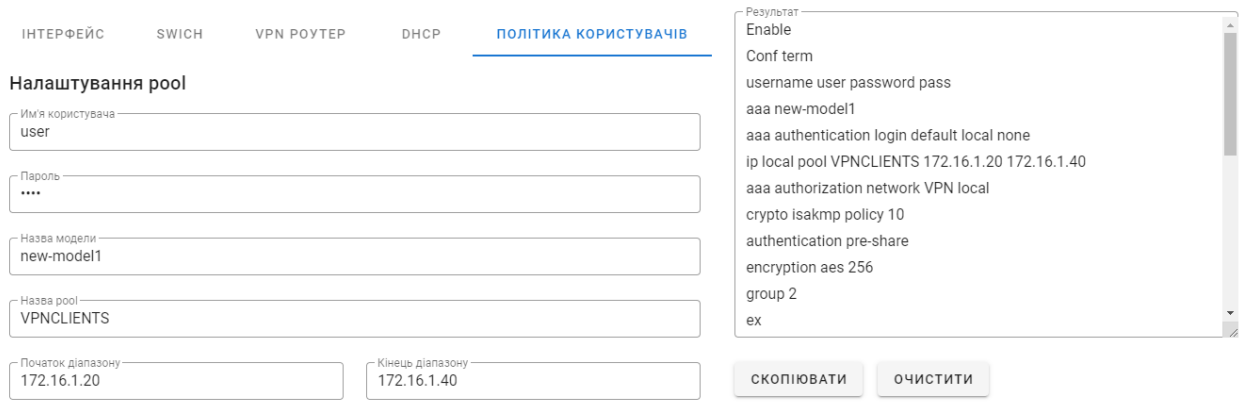

Рисунок 3.6 – Приклад згенерованого коду для налаштування Pool

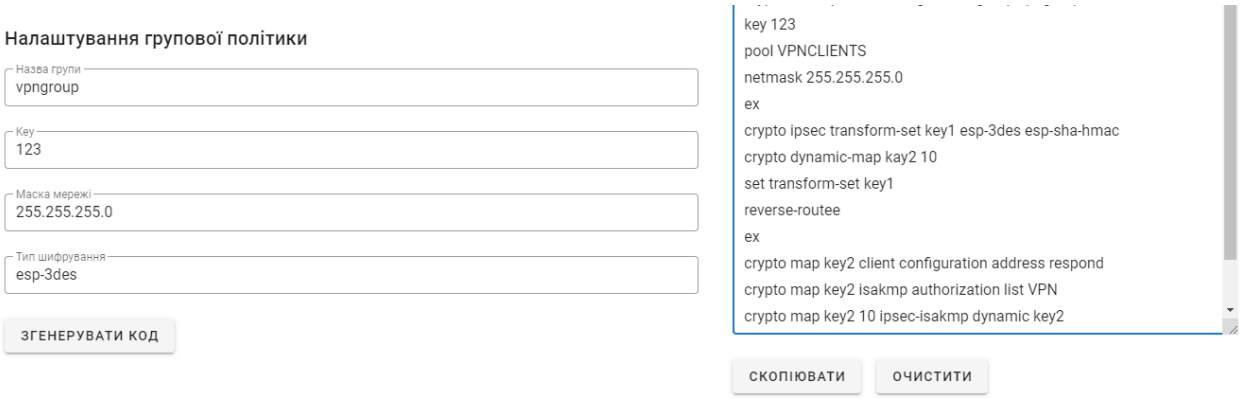

Рисунок 3.7 – Приклад згенерованого коду для налаштування групової політики

Всі поля є обов'язковими для заповнення і перевіряються на правильність вхідний даних. У випадку помилки або відсутності даних відображається повідомлення з описом помилки (рис. 3.8-9).

Рисунок 3.8 – Відсутність даних в формі

Рисунок 3.9 – Не коректні вхідні дані

При кожному натисканні на кнопку «Згенерувати код» якщо корректно заповненні всі дані в поле ркзвльтату додасться новий згенерований код. Далі програма опрацює вхідні данні та виведе необхідні команди у відповідне вікно. Користувач може за необхідності скопіювати текст з поля результату у буфер обміну, або ж очистити його відповідними конпками.

Для перевірки працездатності додатку було розроблено віртуальну VPN мережу в програмному забезпеченні Cisco Packet Tracer 6.0. Cisco Packet Tracer – це багатофункціональна програма для моделювання мережі, яка дозволяє

експериментувати з поведінкою мережі та оцінювати можливі ситуації. Як складова частина всебічного навчального середовища, Packet Tracer забезпечує функції моделювання, візуалізації, створення, автентифікації та співпраці та полегшує викладання складних технічних принципів. Приклад побудованої мережі показаний на рис. 3.10.

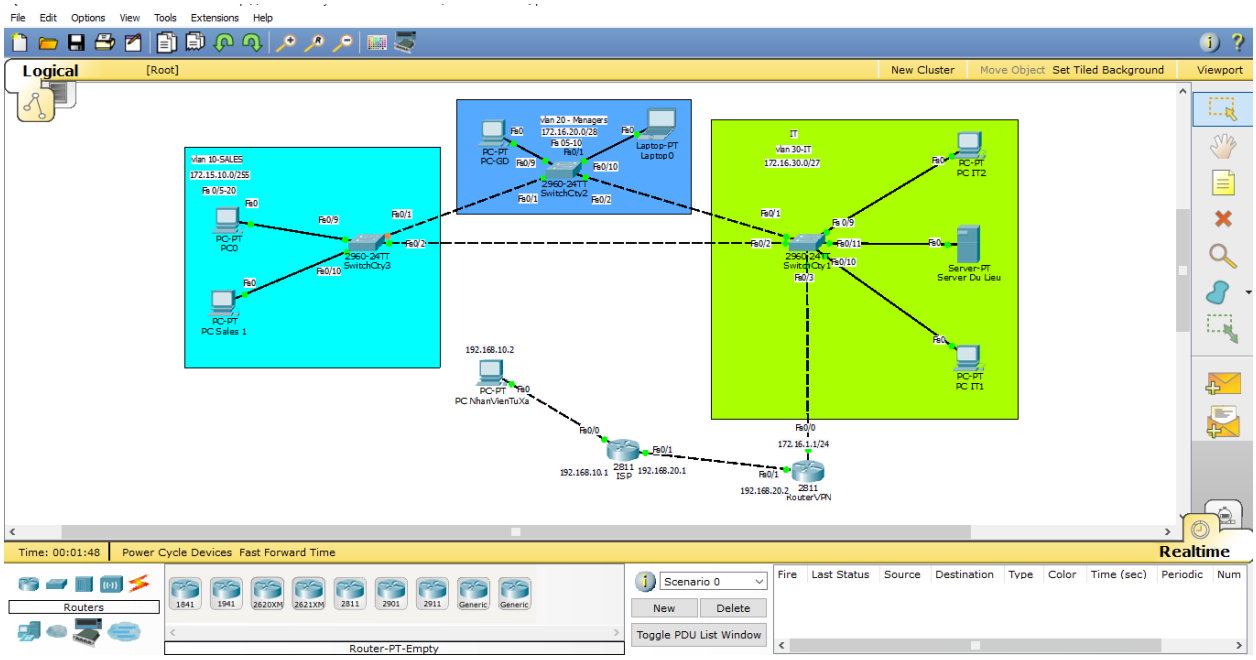

Рисунок 3.10 – Віртуальна VPN мережа

Скопіювавши згенеровані команди за допомогою додатку необхідно їх вставити в термінал роутеру, або switch для налаштування. Далі наведено приклад налаштування switch з використанням згенерованих команд (рис. 3.11).

```
Switch>Enable
Switch#Conf term
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface range fastEthernet 0/5 - 16
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 30
Switch(config-if-range)#ex
Switch(config)#interface range fastEthernet 0/1 - 3
Switch(config-if-range)#switchport mode trunk
Switch(config-if-range)#switchport trunk native vlan 1
Switch(config-if-range)#ex
Switch(config)#
```

Рисунок 3.11 – Налаштування switch

Після налаштування всіх роутерів та switch потрібно перевірити працездатність системи для цього необхідно надіслати пакети даних на кожен з комп'ютерів, у випадку успішного налаштування відобразиться повідомлення «successful» (рис. 3.12). А також перевірити конфігурації VPN (рис. 3.13).

Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic
	Successful	PC IT1	PC Sales 1	ICMP		0.000	N
	Successful	PC IT2	Laptop0	ICMP		0.000	N
	Successful	PC0	PC-GD	ICMP		0.000	N

Рисунок 3.12 – Перевірка з'єднання з комп'ютерами мережі

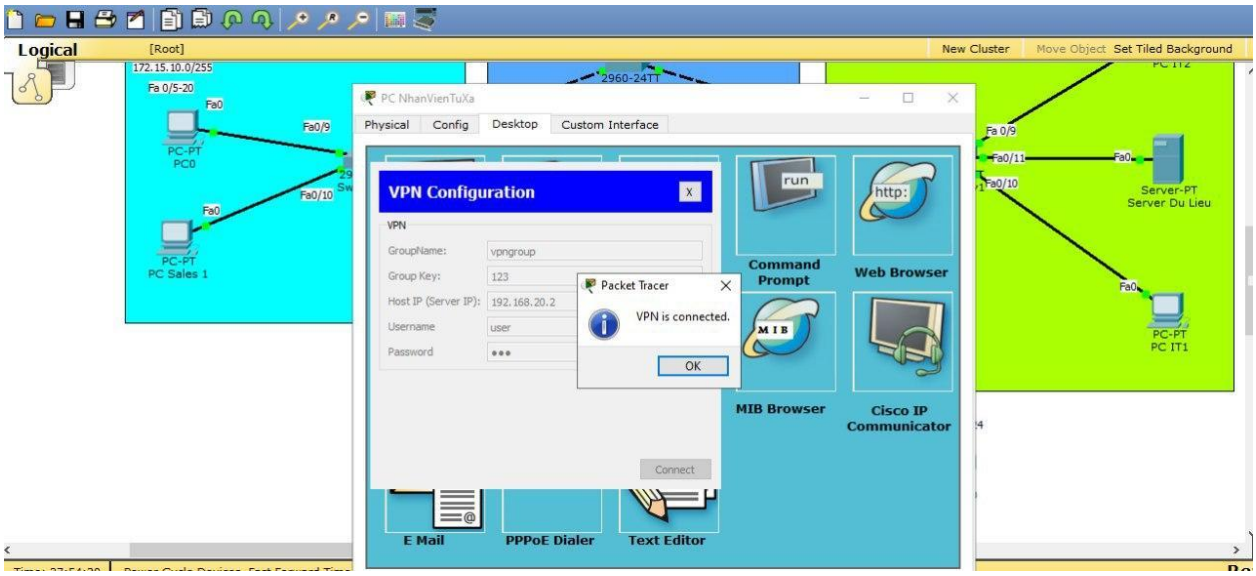

Рисунок 3.13 – Перевірка налаштування VPN

В результаті віртуальна VPN мережа була успішно налаштована за досить короткий час і без виникнення помилок.

ВИСНОВКИ

Висновки до роботи можна сформулювати наступним чином: З’ясовано принципи роботи популярних мережевих симуляторів, одним з яких є Cisco Packet Tracer. Популярними мережевими симуляторами є також GNS3, UNetLab. Кожний з цих симуляторів представляє доступ до основних видів мережевого обладнання, таких як комутатори, маршрутизатори. Але недоліком сучасних симуляторів є відсутність графічного інтерфейсу для налаштування VPN, що робить процес її конфігурування довготривалим та складним для початківців.

В рамках роботи розроблена веб-орієнтована інформаційна система. Графічний інтерфейс якої дозволяє налаштувати віртуальну приватну мережу. При роботі з програмою необхідно задати ключові параметри роутера (ip-адреси інтерфейсів), а також вибрати протокол динамічної маршрутизації, для якого потрібно отримати налаштування. Система дозволяє зручно перенести згенерований код налаштувань роутера в налаштування реального мережевого обладнання. Розроблене програмне забезпечення дозволяє початківцям успішно налаштовувати мультисервісні мережі Ethernet, не вимагаючи на початковому етапі знання команд конфігурації роутерів Cisco, а також дає можливість автоматизувати процес налаштування роутерів та позбавляє від виконання рутинних операцій. Даний інтерфейс може використовуватись як для налаштування VPN в симуляторах, так і на реальному обладнанні.

Мета досягнута, поставлені завдання виконані: досліджена предметна область та обрано стратегію розробки. Всі цілі та завдання були виконані.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. А. Сергеев "Основи локальних комп'ютерних мереж", 2016. – 184с.

2. В. Оліфер, Н. Оліфер "Комп'ютерні мережі. Принципи, технології, протоколи", 2016. - 992с.

3. Misra Kundan "OSS for Telecom Networks", 2017. – 340с.

4. Джеймс Куроуз, Кіт Росс "Комп'ютерні мережі", 2016. – 520с.

5. Web application control [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/figure/The-Web-application-control-solution-based-on-SD_fig1_241171063. (дата звернення: 17.04.2020).

6. Олексій Сергєєв "Основи локальних комп'ютерних мереж", 2016. – 266с.

7. Віртуальні приватні мережі. Основні поняття VPN [Електронний ресурс] – Режим доступу до ресурсу: https://present5.com/virtualnye-chastnye-seti-1-osnovnye-ponyatiya-vpn/. (дата звернення: 17.04.2020).

8. Протоколи і методи реалізації VPN мереж [Електронний ресурс] – Режим доступу до ресурсу: https://ua-referat.com/Протоколи_і_методи_реалізації_VPN_мереж. (дата звернення: 17.04.2020).

9. Мережі VPN та проблеми їх захисту [Електронний ресурс] – Режим доступу до ресурсу: https://studfile.net/preview/8099727/. (дата звернення: 17.04.2020).

10. Класифікація VPN за типом технічної реалізації [Електронний ресурс] – Режим доступу до ресурсу: https://studbooks.net/2239928/informatika/klassifikatsiya_tipu_tehnicheskoy_realizatsii. (дата звернення: 17.04.2020).

11. Таненбаум, Д. Уэзеролл "Комп'ютерні мережі" 5 вид. 2016. – 960с.

12. Створення і використання віртуальних приватних мереж [Електронний ресурс] – Режим доступу до ресурсу: https://works.doklad.ru/view/K8qeIZiPnjY.html. (дата звернення: 17.04.2020).

13. Основи роботи віртуальних приватних мереж [Електронний ресурс] – Режим доступу до ресурсу: - http://vlasti.net/news/246593. (дата звернення: 19.04.2020).

14. Set Up a Remote Access Tunnel? [Електронний ресурс] – Режим доступу до ресурсу: https://www.cisco.com/c/en/us/support/docs/smb/routers/cisco-rv-series-small-business-routers/smb624-set-up-a-remote-access-tunnel-client-to-gateway-for-vpn-clie.html. (дата звернення: 20.04.2020).

15. VPN маршрутизатор [Електронний ресурс] – Режим доступу до ресурсу: http://www.oszone.net/3604/. (дата звернення: 20.04.2020).

16. Что такое DHCP? [Електронний ресурс] – Режим доступу до ресурсу: https://lanmarket.ua/entsiklopediya/telekommunikatsionnye-tekhnologii/dhcp.html. (дата звернення: 20.04.2020).

17. Як працює DHCP [Електронний ресурс] – Режим доступу до ресурсу: http://www.oszone.net/829/. (дата звернення: 20.04.2020).

18. Що таке сайт? Інтернет сайт? Види сайтів [Електронний ресурс] – Режим доступу до ресурсу: http://moolkin.ru/chto-takoe-sayt-internet-sayt-vidy-saytov/. (дата звернення: 20.04.2020).

19. Джон Дукетт. HTML and CSS design and build websites, 2016. – 420 с.

20. Що таке CSS сайтів [Електронний ресурс] – Режим доступу до ресурсу: http://phpist.com.ua/css/5-whatcss/. (дата звернення: 22.04.2020).

21. Хавербеке Марейн. Виразний JavaScript. 2 видання, 2015. – 745 с.

22. Джо Хабракен "Маршрутизатор Cisco. Практичне застосування", 2018. – 312с.

23. Побудова локальної обчислювальної мережі на основі VPN-технології [Електронний ресурс] – Режим доступу до ресурсу:

https://ukrbukva.net/page,2,94066-Postroenie-lokal-noiy-vychislitel-noiy-seti-na-osnove-VPN-tehnologii.html. (дата звернення: 19.04.2020).

24. How can I reach the VPN clients from my network? [Електронний ресурс] – Режим доступу до ресурсу: https://openvpn.net/faq/how-can-i-reach-the-vpn-clients-from-my-network/. (дата звернення: 20.04.2020).

ДОДАТКИ

Додаток А

Фрагмент коду з файлу App.vue: відповідає за відображення базової структури додатку і підключення всіх необхідних компонентів.

```
<template>
  <v-app id="inspire">
    <h1 class="header text-center mt-4">Налаштування VPN 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>VPN роутер</v-tab>
          <v-tab>DHCP</v-tab>
          <v-tab>Політика користувачів</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>
            <RouterVPN @generateRoute='getData'></RouterVPN>
          </v-tab-item>
          <v-tab-item>
            <Dhcp @generateDHCP='getData'></Dhcp>
          </v-tab-item>
          <v-tab-item>
            <GroupPolicy @generatePolicy='getData'></GroupPolicy>
          </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 RouterVPN from './components/RouterVPN';
import Dhcp from './components/Dhcp';
import GroupPolicy from './components/GroupPolicy';

export default {
  name: 'App',
  data: () => ({
    tab: null,
    result: ''
  }),
  components: {
    Interface,
    SwitchComponent,
    RouterVPN,
    Dhcp,
    GroupPolicy
  },
  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 налаштувань 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 налаштування комутаторів 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 RouterVPN.vue: відповідає за налаштування головного роутеру VPN.

```
<template>
  <v-container>
    <h1 class="title mb-4">Налаштування VPN роутеру</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.encapsulation"
      label="Тип інувпсуляції"
      :rules="vlanRules"
      outlined
      dense
      ></v-text-field>
      <v-text-field
      v-model="data.vlan"
      label="Vlan"
      :rules="vlanRules"
      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-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: 'RouterVPN',
      mixins: [validation],
      data: () => ({
        valid: false,
        data: {
            fastEthernet: "",
            encapsulation: "",
            ip: "",
            mask: "",
            vlan: ""
          }
      }),
      methods: {
        generate() {
            var string = "interface fastEthernet "+this.data.fastEthernet+"\n"+
            "encapsulation "+this.data.encapsulation+" "+this.data.vlan+"\n"+
            "IP address "+this.data.ip+" "+this.data.mask+"\n"+
            "ex"+"\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>
```

Додаток Е

Фрагмент коду з файлу GroupPolicy.vue: відповідає за налаштування групової політики.

```
<template>
  <v-container>
    <v-form ref="form" v-model="valid" lazy-validation>
      <h1 class="title mb-4">Налаштування pool</h1>
    <v-text-field
      v-model="poolData.username"
      label="Им'я користувача"
      outlined
      dense
      :rules="interfaceRules"
    ></v-text-field>

    <v-text-field
      v-model="poolData.password"
      label="Пароль"
      type="password"
      outlined
      dense
      :rules="interfaceRules"
    ></v-text-field>

    <v-text-field
      v-model="poolData.model"
      label="Назва модели"
      outlined
      dense
      :rules="interfaceRules"
    ></v-text-field>

    <v-text-field
      v-model="poolData.pool"
      label="Назва pool"
      outlined
      dense
      :rules="interfaceRules"
    ></v-text-field>
      <v-row>
        <v-col class="py-0">
          <v-text-field
```

```
            v-model="poolData.startRange"
            label="Початок діапазону"
            :rules="ipAddressRules"
            outlined
            dense
          ></v-text-field>
      </v-col>
      <v-col class="py-0">
      <v-text-field
        v-model="poolData.finishRange"
        label="Кінець діапазону"
        :rules="ipAddressRules"
        outlined
        dense
      ></v-text-field>
      </v-col>
  </v-row>
  <h1 class="title mb-4">Налаштування групової політики</h1>
<v-text-field
  v-model="groupData.name"
  label="Назва групи"
  outlined
  dense
  :rules="interfaceRules"
></v-text-field>
<v-text-field
  v-model="groupData.key"
  label="Key"
  outlined
  dense
  :rules="interfaceRules"
></v-text-field>
<v-text-field
  v-model="groupData.mask"
  label="Маска мережі"
  :rules="maskRules"
  outlined
  dense
></v-text-field>
<v-text-field
  v-model="groupData.cript"
  label="Тип шифрування"
  outlined
  dense
  :rules="interfaceRules"
```

```
            ></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,
          poolData: {
            username: "",
            password: "",
            model: "",
            pool: "",
            startRange: "",
            finishRange: "",
          },
          groupData: {
            name: "",
            key: "",
            mask: "",
            cript: ""
          }
        }),
        methods: {
          generate() {
            var string = "username "+this.poolData.username+" password
"+this.poolData.password+" \n"+
            "aaa "+this.poolData.model+" \n"+
            "aaa authentication login default local none\n"+
            "ip local pool "+this.poolData.pool+" "+this.poolData.startRange+"
"+this.poolData.finishRange+"\n"+
            "aaa authorization network VPN local\n"+
            "crypto isakmp policy 10\n"+
            "authentication pre-share\n"+
            "encryption aes 256\n"+
            "group 2\n"+
            "ex\n"+
            "crypto isakmp client configuration group "+this.groupData.name+"\n"+
            "key "+this.groupData.key+"\n"+
```

```
      "pool "+this.poolData.pool+"\n"+
      "netmask "+this.groupData.mask+"\n"+
      "ex\n"+
      "crypto ipsec transform-set key1 "+this.groupData.cript+" esp-sha-hmac\n"+
      "crypto dynamic-map kay2 10\n"+
      "set transform-set key1\n"+
      "reverse-routee\n"+
      "ex\n"+
      "crypto map key2 client configuration address respond\n"+
      "crypto map key2 isakmp authorization list VPN\n"+
      "crypto map key2 10 ipsec-isakmp dynamic key2\n"+
      "ex\n";
      this.$emit('generatePolicy', {
        text: string,
      });
    }
  }
};
</script>
```