МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

«Графічний інтерфейс налаштування набору технологій DMVPN (Multipoint GRE+NHRP) over IPsec»

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

Керівник роботи — **Великодний Д.В.**

Студента групи ІН м.-92 — **Ульянік О.М.**

СУМИ 2020

Сумський державний університет

(назва вузу)

Факультет *ЕлІТ* Кафедра *Комп'ютерних наук*

Спеціальність *122 «Комп'ютерні науки»*

Затверджую:

зав. кафедри _____

"_____" _____ 20___р.

ЗАВДАННЯ

НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Ульяніка Олександра Миколайовича

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Графічний інтерфейс налаштування набору технологій DMVPN (Multipoint GRE+NHRP) over IPsec
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) Літературний огляд. 2) Моделювання технології Dmvpn Over Ipsec та аналіз пакетів з використанням Емулятора GNS3. 3) Створення схеми Gre Over Ipsec в програмному середовищі*

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	Літературний огляд		
2	Моделювання технології Dmvpn Over Ipsec та аналіз пакетів з використанням Емулятора GNS3		
3	Створення схеми Gre Over Ipsec в програмному середовищі		
4	Оформлення кваліфікаційної магістерської роботи		

Студент – дипломник _____

(підпис)

Керівник проекту _____

(підпис)

РЕФЕРАТ

Записка: 81 стор., 59 рис., 29 джерело, 2 додатки

Мета роботи — розробка програмного забезпечення, графічний інтерфейс якого буде дозволяти налаштувати на маршрутизаторах конфігурацію захищених мереж з використанням протоколів DMVPN over IPsec.

Об'єкт дослідження — налаштування DMVPN over IPsec в локальній та глобальній мережах Ethernet на базі маршрутиризаторів CISCO.

Предмет дослідження — набір протоколів для забезпечення захисту даних IPsec та набір протоколів DMVPN.

Методи дослідження — моделювання в графічному емуляторі мереж GNS3 та аналіз трафіку в сніфері WireShark.

Результати — Побудована складна система, веб-інтерфейс якої надає змогу створити налаштування для маршрутиризатора, налаштувати на роутерах конфігурацію складних мереж DMVPN over IPsec. Розроблений проект дає змогу з великою швидкістю перенести отриманий код налаштувань роутера у термінал фізичного обладнання Cisco. Створений інтерейс має інформаційну сторінку щодо введених команд. Крім того програма дозволяє перевірити правильність введених даних. Основна частина програми реалізована на мові програмування JavaScript.

ЗАХИЩЕНІ МЕРЕЖІ, VPN, НАБІР ПРОТОКОЛІВ IPSEC,
ТЕХНОЛОГІЯ DMVPN, ВЕБ-ОРІЄНТОВАНА СИСТЕМА,
GNS3, JAVASCRIPT, CISCO, NHRP, MULTIPOINT GRE

ЗМІСТ

ВСТУП

Проблема передачі інформації через незахищені мережі набула доволі гострого характеру. Захистити наш трафік можна за умови використання механізмів його тунелювання та шифрування, що в більшості випадків дозволяє повністю позбавитись від втрати переданої інформації.

Для більш зрозумілого та швидкого налаштування в більшості випадків використовують мережеві симулятори, які допомагають швидко побудувати складні та безпечні комп'ютерні мережі без реального обладнання. Проте в більшості симуляторах на сьогоднішній день для налаштування треба вміти використовувати термінал (налаштування за допомогою базових екранних форм), що значно ускладнює процес створення комп'ютерної мережі.

Саме тому за основу налаштування «Графічного інтерфейсу налаштування набору технологій DMVPN (Multipoint GRE+NHRP) over IPsec» стало створення зрозумілого для всіх та швидкого інструменту для налаштування технології DMVPN на роутерах фірми Cisco. Набір технологій DMVPN дозволяє забезпечити створення динамічного тунельованого зв'язку між віддаленими сегментами мережі, імітуючи пряме з`єднання роутерів. А для безпеки передачі інформації використано набір протоколів IPsec, який дозволяє забезпечити надійність передачі даних. Поєднання протоколів DMVPN та IPsec забезпечує постійний та шифрований канал зв`язку.

Графічний інтерфейс програми допомагає користувачу як і користувачам без досвіду, так і фахівцям у цій галузі налаштувати технологію DMVPN over IPsec. Для початку роботи з інтерфейсом необхідно лише задати IP-адреси інтерфейсів роутерів (Hub та Spoke), які потрібні для налаштування технології.

Графічний інтерфейс був реалізований на мові JavaScript, а веб форма побудована використовуючи веб-програмування: HTML, CSS. Це робить інтерфейс крос платформним та портативним

1 ЛІТЕРАТУРНИЙ ОГЛЯД

1.1 Технологія VPN та її основні типи

Як відомо, VPN — це віртуальна приватна мережа, яка дозволяє користувачеві безпечно та приватно підключатися до приватної мережі через Інтернет. VPN створює зашифроване з'єднання, відоме як тунель VPN, і весь Інтернет-трафік та зв'язок проходять через цей захищений тунель. Таким чином, збереження даних користувачів у безпеці та конфіденційності. Іншими словами технологія дозволяє забезпечити одне або кілька мережевих з'єднань (логічну мережу) поверх іншої мережі (наприклад, Інтернет) [1, 2].

Канали віртуальної приватної мережі, так само як і виділені канали, з'єднують окремі мережі клієнта в єдину ізольовану мережу [3].

Існує два основних типи VPN, які описані нижче:

1. Віддалений доступ VPN (Remote access)

Віддалений доступ VPN дозволяє користувачеві підключатися до приватної мережі та віддалено отримувати доступ до її послуг та ресурсів. Зв'язок між користувачем та приватною мережею відбувається через Інтернет, і з'єднання є безпечним та приватним.

Під час подорожі корпоративний працівник використовує VPN для підключення до приватної мережі своєї компанії та віддаленого доступу до файлів та ресурсів у приватній мережі.

Домашні або приватні користувачі VPN в основному використовують служби VPN, щоб обійти регіональні обмеження в Інтернеті та отримати доступ до заблокованих веб-сайтів. Користувачі, які усвідомлюють безпеку Інтернету, також використовують послуги VPN для підвищення своєї безпеки та конфіденційності в Інтернеті.

2. Site-to-Site VPN

VPN Site-to-Site також називається VPN від маршрутизатора до маршрутизатора і в основному використовується в корпораціях. Компанії, офіси яких знаходяться в різних географічних місцях, використовують

мережеву VPN для підключення мережі одного офісу до мережі в іншому офісі.

Оскільки VPN Site-to-Site базується на зв'язку маршрутизатора до маршрутизатора, у цьому типі VPN один маршрутизатор виконує роль клієнта VPN, а інший маршрутизатор — як VPN-сервер. Зв'язок між двома маршрутизаторами починається лише після перевірки автентичності між ними [2, 4].

Також необхідно відмітити, що Site-to-Site VPN-мережі можна розгорнути двома способами [3]:

- Інтранет VPN;
- Екстранет VPN.

Інтранет VPN пов'язує корпоративні штаб-квартири, віддалені офіси та філії через спільну інфраструктуру, використовуючи виділені з'єднання. Компанії користуються тими ж політиками, що і приватна мережа, включаючи безпеку, якість обслуговування (QoS), керованість та надійність.

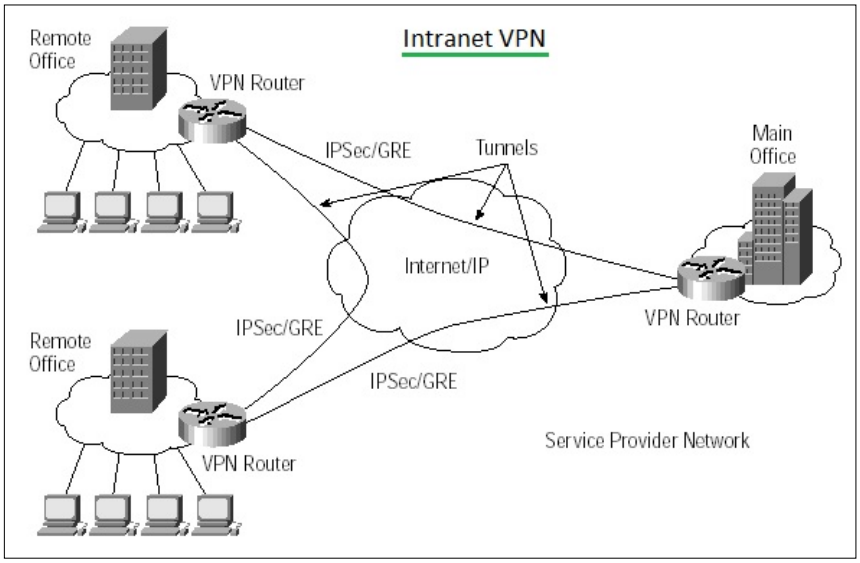

Рисунок 1.1— Інтранет VPN[5]

Екстранет — це контрольована приватна мережа, яка надає доступ партнерам, постачальникам та постачальникам або уповноваженому набору клієнтів - як правило, до підмножини інформації, доступної з інтрамережі організації. Екстранет схожий на DMZ, оскільки він надає доступ до

необхідних послуг для уповноважених осіб, не надаючи доступу до всієї мережі організації.

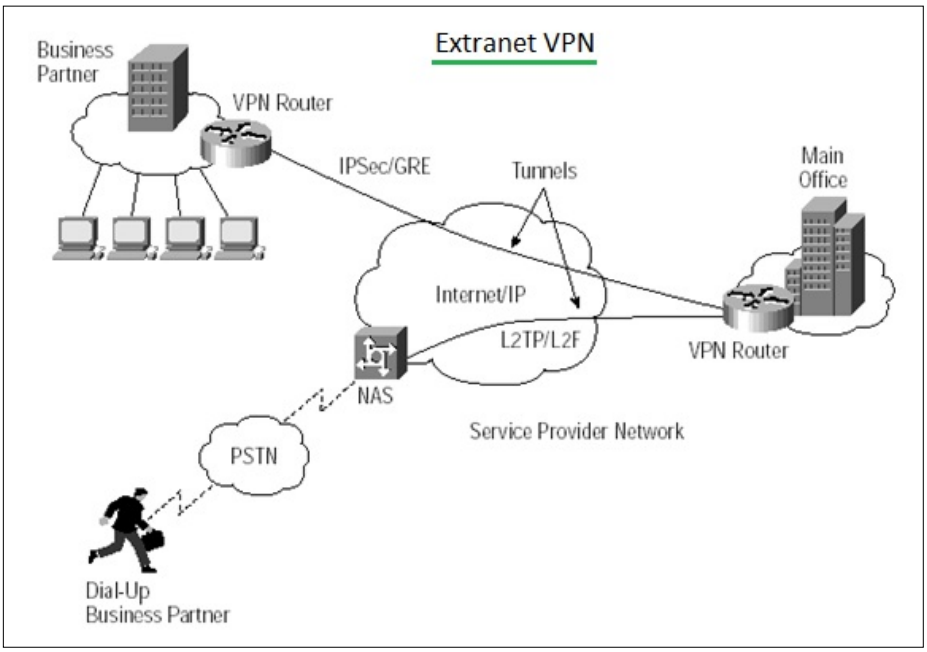

Рисунок 1.2— Екстранет VPN[5]

Служби VPN в інтрамережі та екстрамережі на основі IPSec, GRE та мобільних IP створюють безпечні тунелі в мережі IP (рис.1.1 та рис.1.2). Ці технології використовують галузеві стандарти для встановлення безпечних з'єднань "точка-точка" в топології сітки, яка накладається на IP-мережу постачальника послуг або Інтернет. Вони також пропонують можливість визначити пріоритети заявок. Однак архітектура IPSec включає стандарт, запропонований IETF для шифрування на основі IP, і забезпечує зашифровані тунелі від точки доступу до інтрамережі або екстрамережі та через неї [6].

Саме до Site-to-Site VPN і відноситься технологія Dynamic Multipoint VPN (DMVPN) про яку і піде далі мова.

1.2 Технологія DMVPN.Особливості функціонування

Для безпечної передачі даних можна використовувати доволі багато варіантів тунелів. Наприклад, можна шифрувати дані за допомогою GRE over IPSec та протоколів динамічної маршрутизації. Однак виникає проблема масштабованості нашої мережі.

Припустимо у нас є мережа з рисунку 1.3. Для налаштування нам знадобиться лише 3 тунелі.

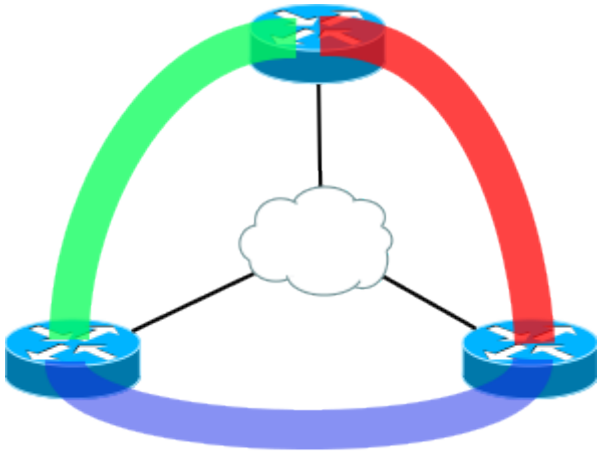

Рисунок 1.3— Невелика мережа на 3 тунелі [7]

Однак глянемо на мережу з рисунка 1.4. Для побудови тунелів тут знадобиться набагато більше часу.

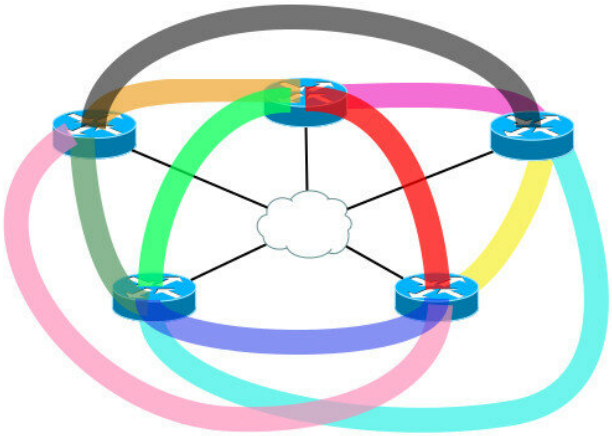

Рисунок 1.4— Велика мережа на 10 тунелів [7]

Крім того, якщо не використовувати Full-Mesh, а звернутися до топології Hub-and-Spoke з однієї центральною точкою, то з'являється інша проблема - трафік між будь-якими філіями буде проходити через центральний вузол [7].

Саме технологія DMVPN дозволяє вирішити ці проблеми. Дане рішення використовується, коли потрібна висока масштабованість і легкість настройки при підключенні філій до головного офісу.

DMPVN одне з найбільш масштабованих і ефективних рішень VPN підтримуваних компанією Cisco. В основному воно використовується при топології Hub-and-Spoke, де ви хотіли б бачити прямі VPN тунелі Spoke-to-Spoke в доповлення до звичайних Spoke-to-Hub тунелях. Це означає, що філії зможуть спілкуватися один з одним безпосередньо, без необхідності проходження трафіку через Hub.

Якщо вам необхідно підключити багато маршрутиризаторів до головного офісу, то DMPVN буде ідеальним вибором. Крім того, DMPVN підтримує не тільки Hub-and-Spoke, але і Full-Mesh топологію, так як всі сайти мають між собою зв'язність без необхідності настройки статичних VPN тунелів між сайтами [8].

В основі DMVPN лежать кілька технологій:

- Multipoint GRE-тунелі.

Multipoint GRE (mGRE) – це технологія, яка дозволяє термінувати на собі кілька GRE-тунелів. Технологія mGRE-тунелю дозволяє одному GRE-інтерфейсу підтримувати декілька IPsec-тунелів, що значно спрощує складність налаштувань великих мереж, в порівнянні з звичайним GRE-тунелем.

- Протокол NHRP (Next Hop Resolution Protocol) про який пізніше;
- Протоколи динамічної чи статичної маршрутизації;
- Профілі IPsec (IPsec profiles) [9].

Крім того у DMVPN розрізняють три фази або версії: першу, другу і третю. Пізніше розглянемо докладніше кожну з них [10].

1.3 Протокол NHRP

У комп’ютерній мережі Next Hop Resolution Protocol (NHRP) - це протокол або метод, який можна використовувати для того, щоб комп’ютер, що надсилає дані на інший комп’ютер, міг засвоїти найбільш прямий шлях (найменшу кількість стрибків) до комп’ютера-одержувача. NHRP - це

протокол клієнт-сервер. Серверна сторона називається NHS або концентратором (hub), тоді як клієнт називається NHC або spoke.

Якщо комп’ютер-одержувач знаходиться в одній підмережі, використання NHRP повідомляє комп’ютеру-відправнику, що комп’ютер-одержувач локальний, і він може надсилати наступні пакети даних безпосередньо на комп’ютер-одержувач, використовуючи свою адресу підмережі, а не свою глобальну мережеву адресу.

Якщо комп'ютер-одержувач знаходиться не в тій самій підмережі, використання NHRP повідомляє комп'ютеру-відправнику комп'ютер у підмережі, маршрутизатор якого забезпечує найбільш прямий шлях до комп'ютера-одержувача, і відправник тепер може пересилати наступні пакети даних на цей маршрутизатор [11–13].

Cisco Dynamic Multipoint VPN (DMVPN) базується на NHRP, і frr nhrpd реалізує цей сценарій.

Як результат у мережі DMVPN:

- Головний маршрутизатор (Hub) працює як next-hop-сервер, а маршрутизатори spoke виступають клієнтами у цій мережі.
- Hub-маршрутизатор стає базою даних для створенного NHRP, в якій зберігаються відповідності між всіма фізичними адресами та адресами тунелів для всіх підключених маршрутизаторів (spoke).
- Для кожного spoke за замовчуванням hub-маршрутизатор відповідно вказаний як NHS. Крім того вказується зв’язок між фізичною інтерфейсом (адресою)та адресою побудованого тунелю до hub-маршрутизатора.
- При новому чи повторному підключенні кожен spoke-маршрутизатор реєструється на хабові і, за необхідністю, може отримувати від сервера детальну інформацію щодо адрес інших spoke-маршрутизаторів для побудови тунелів між споками, не спілкуючись із хабом в подальшому.
- Network ID – це унікальна характеристика для кожного маршрутизатора (за аналогією - OSPF area). Різні інтерфейси можуть брати участь в різних

NHRP сесіях; це просто розмежувач того, що сесія NHRP на одному інтерфейсі відрізняється. Відповідно, зовсім не обов'язково, щоб Network ID збігалися на різних маршрутизаторах [3, 9, 11].

1.4 Профіль IPsec. Опис та архітектура даного набору протоколів

Завдяки вже розглянутим протоколам можна побудувати тунель між маршрутиризаторами однак як і із протоколом GRE тунель буде не зашифрований. Для того щоб забезпечити надійність даних використовується набір протоколів IPsec.

IPSec (англ. Internet Protocol Security) - це набір протоколів для забезпечення конфіденційних, який дає можливість захистити IP мережу, за хешування та шифрування даних. IPSec дозволяє аутентифікацію сторін на мережевому рівні, перевірку інікальності адресату, правильність отриманої інформації, конфіденційність даних (шифрування) і захист від повторного відтворення [14, 15].

IPsec працює завдяки двум протоколам налаштування - Authentication Header (AH) і Encapsulating Security Payload (ESP) IP Authentication Header (AH) забезпечує цілісність без встановлення з'єднання, аутентифікацію джерела даних і додаткову службу захисту від повтору а також протокол IKE для аутентифікації та узгодження параметрів для сторін IPSec.

AH використовує хеш-алгоритм для обчислення значення хеша як для корисного навантаження, так і для заголовка пакета, забезпечуючи цілісність пакета. Однак це викликає дуже специфічну проблему. AH не працюватиме через NAT-пристрій. NAT змінює IP-заголовок пакета під час перекладу, але значення хеша не змінюється. Таким чином, приймаючий пристрій буде вважати, що пакет був змінений при передачі і відхилив пакет.

Протокол Encapsulating Security Payload (ESP) може забезпечувати конфіденційність (шифрування) і обмежену конфіденційність трафіку. Він також може забезпечувати підключення. Він також може забезпечити

цілісність без встановлення з'єднання, аутентифікацію джерела даних і службу захисту від повтору. (Один або інший набір цих служб безпеки повинен

застосовуватися щоразу, коли викликається ESP.) ESP виконує функції конфіденційності, аутентифікації і цілісності. Таким чином, ESP виконує шифрування і за своєю суттю більш безпечний, ніж AH. ESP вводить в пакет як додатковий заголовок, так і трейлер. ESP також використовує алгоритм хешування для цілісності даних. Однак хеш не включає IP-заголовок пакета, і, таким чином, ESP буде (зазвичай) працювати через NAT-пристрій.

Обидва AH і ESP є транспортними засобами для контролю доступу на основі розподілу криптографічних ключів та управління потоками трафіку протоколів безпеки, однак AH необхідний для перевірки цілісності з'єднання між точками (переданих даних) а ESP використовується для шифрування переданої інформації, а також обмеження потоку конфіденційного трафіку. [15, 16].

Для автоматичного обміну ключами за замовчуванням використовується Протокол управління ключами в Інтернеті (Internet Key Management Protocol - IKMP), інакше званий Обмін ключами в Інтернеті (Internet Key Exchange - IKE).

IKE містить дві фази узгодження ключів. Тобто у першій фазі відбувається організація безпечного каналу між сторонами для другої фази IKE, а вже у другій - узгодження і обмін ключами, встановлення SA, простіше кажучи, узгодження роботи учасників захищеного з'єднання або ж IPsec-тунелю. Перша фаза налаштовується одним із двох можливих режимів: основний режим (англ. Main Mode) або агресивний режим (англ. Aggressive Mode). Різниця між ними в рівні захищеності і швидкості роботи. При використанні основного режиму ми захищаємо всю інформацію, передану між вузлами, хоч і більше повільно. Агресивний режим же використовується для прискорення роботи однак ми не шифруємо всі заголовки та наповнення пакету. Його рекомендується використовувати коли основною задачею стає швидкість з’єднання. У другій фазі використовується єлише один режим:

швидкий режим (англ. Quick Mode), який не робить аутентифікації вузлів, виходячи з того, що це було зроблено в першій фазі. В результаті, ця фаза забезпечує обмін ключами, за допомогою яких відбувається шифрування даних [17, 18].

IPSec використовує методи шифрування та налаштування, які можна описати наступними фазами:

1. Починається процесс налаштування IPSec. Відповідно до політики захисту IPSec шифрується трафік, який вже передається між узгодженими сторонами IPSec, і як в результаті починає IKE-процес.
2. Перша фаза IKE. А значить відбувається налаштування аутентифікації сторін IPSec та параметри асоціацій захисту IKE. Після цього створюється захищений канал передачі для налаштування другої фази IKE.
3. Друга фаза IKE. Відбувається фінальне налаштування IKE-процесу, що призводить до налаштування параметрів захисту передачі пакетів технологією IPSec та отримумання інформації щодо пристроїв, які спілкуються між собою.
4. Передача даних. Відбувається передача пакетів між вже зв’язаними IPSec пристроями, використовуючи налаштовані ключі та параметри шифрування, які були занесені в асоціації захисту IPSec.
5. Завершення роботи тунелю IPSec. Після завершення сеансу зв’язку асоціації захисту IPSec очищаються, якщо відбулось перевищення назначеного часу їх існування [19].

1.5 Фази DMVPN

Як вже було сказано DMVPN має три фази, які ми зараз і розглянемо.

У першій фазі допускається динамічне підключення spoke-маршрутизаторів до hub, при цьому взаємодія між мережами, розташованими за spoke, ведеться через центр - через hub-маршрутизатор. На першій фазі

NHRP mGRE використовує NHRP для інформування концентратора про появу spoke. Спочатку налаштовується кожен spoke з IP-адресою концентратора як сервер NHS. Spoke можуть дістатися лише до концентратора і дістатися лише до інших spoke-мереж через концентратор. Тобто в першій фазі неможливо пряму взаємодію між spoke-маршрутизаторами. Всі spoke-маршрутизатори в цій фазі використовують тільки point-to-point тунелі [10, 20].

Перевагою Фази 1 є спрощена конфігурація маршрутизатора концентратора, яка не вимагає статичного відображення NHRP для кожної нового spoke.

Недоліком фази 1 NHRP є неможливість встановлення швидкісних тунелів «spoke-to-spoke». Фаза 2 NHRP вирішує цю проблему та дозволяє використовувати тунелі, що говорять із spoke.

Друга фаза включає в себе оптимізацію шляху проходження трафіку, що передається між spoke-пристроями, за рахунок динамічного побудови тунелів між кінцевими маршрутизаторами, яке стає можливим, якщо spoke-маршрутизатори мають повну інформацію про всі префікси в мережі.

- Налаштування hub-маршрутизатора повинна задовольняти трьом основним правилам:
- Агрегація мереж або не повинна проводитися зовсім, або не повинна приховувати реального розташування префіксів;
- На hub-пристрої повинна бути відключена розщеплення горизонту;
- Hub-маршрутизатор не повинен підміняти адреса next-hop, отриманий від spoke-пристроїв [9, 20].

Перевагою використання другогої фази є оптимізація шляхів передачі трафіку між spoke. До недоліків можна віднести зростання таблиць маршрутизації на spoke-обладнання. Для оптимізації таблиці маршрутиризації і використовується фаза три.

Третья фаза позбавлена недоліків перших двох фаз, але при цьому надає їм переваги: трафік передає оптимальний шлях, при цьому таблиця

маршрутизації не повинна включати в себе всі можливі префікси мереж. Це вдалось завдяки тому, щоб змістити місця в таблиці маршрутизації лише тех префіксів, які реально використовуються в даний момент часу. Запис у таблиці маршрутизації з'являється лише після того, як з'являється трафік, призначений для відповідних одержувачів. Це стає можливим завдяки використанню опцій redirects та shortcuts [9, 20]. При використанні третьої фази немає необхідності виявляти суммування маршрутів (як це було в першій фазі) і не виникає проблем з адресами маршрутизаторів наступного хопу (як у другому фазі). Однак третя фаза потребує більше умов та налаштувань ніж попередні фази.

Треба пам’ятати що ці версії можуть працювати окремо одна від одної, тобто не завжди є сенс налаштовувати третю фазу якщо можна обійтись першою.

1.6 Постановка задачі

Налаштування тунелів доволі часта задача як для людей, починаючих вивчати інформаційні та телекомунікаційні технології проектування, так і для професіоналів у цій галузі. Окрім того доволі часто трафік, який проходить через ці тунелі, треба зашифрувати інформацію, що передається, а потім перевірити, щоб отримати наочне уявлення про різницю між шифрованим та незашифрованим трафіком. Однак, на це можна втратити доволі багато часу, що буває дуже критично. Як результат, з’явилася ідея спробувати полегшити та пришвидшити налаштування технології DMVPN over IPsec. А отже, за мету роботи було взято наступне: перевірити захищеність даних під час використання DMVPN протоколу, DMVPN over IPsec протоколів, а також розробити веб-орієнтовану інформаційну систему із графічним інтерфейсом, який дозволяє автоматично та швидко налаштувати набір протоколів GRE over IPsec на інтерфейсах маршрутизаторів. Окрім того, програма дозволятиме зручне та швидке перенесення згенерованих налаштувань у симулятор чи

емулятор реального обладнання, який підтримує технологію DMVPN або навіть на реальне обладнання Cisco.

Першочергово треба зрозуміти, чи потрібно налаштовувати DMVPN over IPsec замість DMVPN. Крім того треба зрозуміти чи не заважає надбудова у вигляді шифрування для швидкої передачі пакетів, адже якщо швидкість передачі через мережу до іншого комп'ютера або сервера сильно зменшиться, треба буде відмовитись від шифрування та шукати альтернативу цій технології.

Створена веб-орієнтована інформаційну система повинна бути швидкою, зрозумілою для людей без знань команд конфігурації роутерів. Навіть людина без досвіду роботи з таким інтерфейсом повинна мати змогу з ним працювати та не відчувати дискомфорт.

На даний момент я бачу графічний інтерфейс у вигляді у веб-сторінки, на якій треба ввести лише стартові налаштування роутерів (IP-адресу та маску), а в результаті отримати налаштування для цих роутерів, які можна швидко прочитати, скопіювати в буфер обміну та ввести на необхідний реальний чи віртуальний маршрутизатор. Отже, отримуємо швидко налаштований маршрутизатор, а потім і мережу в цілому.

Постановка задачі:

1. Конфігурація базової мережі та протоколу DMVPN в емуляторі GNS3.
2. Налаштування набору протоколів DMVPN over IPsec в емуляторі GNS3.
3. Порівняння захищеності мережі під час використання протоколу DMVPN та DMVPN over IPsec та часу передачі пакету в цій мережі.
4. Перевірка можливості налаштування якості обслуговування (QoS) поверх технології DMVPN over IPsec в емуляторі GNS3.
5. Розробка графічного інтерфейсу налаштування набору протоколів DMVPN over IPsec.
6. Тестування розробленої веб-орієнтованої програми в емуляторі GNS3, на реальному обладнанні.

2 МОДЕЛЮВАННЯ ТЕХНОЛОГІЇ DMVPN OVER IPSEC ТА АНАЛІЗ ПАКЕТІВ З ВИКОРИСТАННЯМ ЕМУЛЯТОРА GNS3

2.1 Програмні засоби для моделювання та аналізу роботи комп'ютерних мереж

2.1.1 Емулятор комп'ютерних мереж GNS3

На жаль симулятор комп'ютерних мереж Cisco Packet Tracer немає підтримки багато точкових тунелів GRE (multipoint GRE tunnels). Як результат, в цьому симуляторі немає підтримки DMVPN взагалі, що змушує нас відмовитись нас від нього на користь емулятора GNS3.

Graphical Network Simulator-3 (скорочений до GNS3) — це мережевий графічний симулятор мережі графічним інтерфейсом (рис. 2.1), який створює віртуальні мережі [21, 22].

GNS3 дозволяє реалізувати мережі для робітників різних професій. Крім того він дозволяє експерементувати із реальними прошивками роутерів без шкоди для обладнання. Що дозволить спочатку спробувати схему на емульованому роутері,а потім- на реальному.

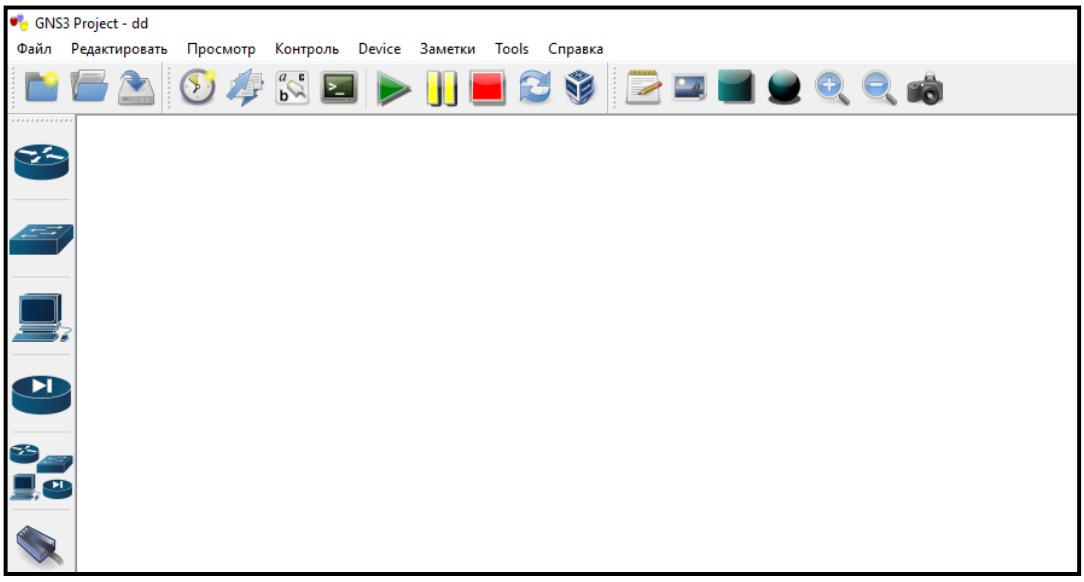

Рисунок 2.1 — Стартове вікно GNS3

Особливо це критично, коли взагалі немає реального обладнання,яке коштує значних грошей. Ми можемо навіть поєднувати як емульоані прошивки роутерів так і реальне обладнання. Це досягається завдяки повному копіюванню розгортування прошивки на маршрутизаторах, навіть робота з процессором та пам'яттю [22]. Для такого детального копіювання використовуеться можливості емулятору Dynamips.

Перш за все GNS3 і ціниться за можливість перевірити схему на роботоздатність. Він дозволяє емулювати майже будь-яку схему будь то локальна чи глобальна мережа. У нашому випадку з Cisco, емулятор спочатку створює модель роутера а вже потім запускає всередині його реальну операційну систему. Для цього нам потрібен лише образ маршрутизатора, який ми будемо відкривати в Graphical Network Simulator-3. Не маючи на руках реального обладння, нам вдається працювати із прошивкою цього пристрою. На відміну від емулятору, симулятори не дають змоги повних можливостей реального обладнання. Як вже було раніше зазначений - симулятор мереж Cisco Packet Tracer [22, 23].

Перевагами GNS 3 є:

1. Емуляція пристрою від запуску прошивки до налаштування будь-якої розробленої технології для даної прошивки. А це значить, що запустивши віртуальний роутер у програмі, ми маємо всі ті ж можливості налаштування, що і на реальному обладнанні. Наприклад, Cisco Packet Tracer дасть змогу лише перегляну та налаштувати прописані технології а не технології нашого віртуального роутьеру[22].
2. Ми можемо реально візуалізувати мережу, адже ми можемо створити віртуальному схему із роутерами від різних виробників, як це часто буває у реальних схемах. Це не знадобиться в нашій роботі, однак може статися в нагоді при роботі із реальним обладнанням [22].
3. Перевагою емулятору над симулятором також можна віднести можливість створення повноцінних серверів та віртуальних машин. Це дає змогу перевірити роботоздатність складної серверної мережі або ж

подивитись функціонування мережі на віртуальному комп'ютері. Це дає можливість навіть вийти у справжній інтернет з віртуальної машини в VirtualBox[22].

Проте, GNS3 має недоліки:

1. Основною проблемою нашого емулятору є неможливість емуляції комутатори. Це спричинено тим, що на даний момент неможливо в повній мірі емулювати мікросхеми, які працюють швидше звичайного процесору [24].
2. Для побудови мереж необхідна доволі сильна система із великими потужностями,зокрема потрібен сильний ЦП та багато ОЗУ.
3. Третій недолік — недостатньо інформативний графічний інтерфейс. Дозволяє тільки подивитися створену топологію, тобто з'єднання та опис пристроїв, чого може бути недостатньо.

2.1.2 Аналізатор мережевого трафіку WireShark

Wireshark — це сніфер, який дозволяє перехоплювати пакети в створеній мережі. Ця програма дозволяє навіть переглянути вміст пакету та його заговки. Ці та багато інших можливостей дозволяють аналізувати трафік, і як результат надійність створеної мережі. Раніше такі інструменти були або дуже дорогими, власними, або обома. Однак із появою Wireshark це змінилося. Wireshark доступний безкоштовно, з відкритим кодом і є одним з найкращих аналізаторів пакетів, доступних сьогодні та став широко прийнятий як галузевий стандарт. [24, 25].

Wireshark дозволяє грамотно відсортувати перехоплені пакети(рис.2.2) і надати інформацію щодо їх протоколу, що буває дуже зручно.

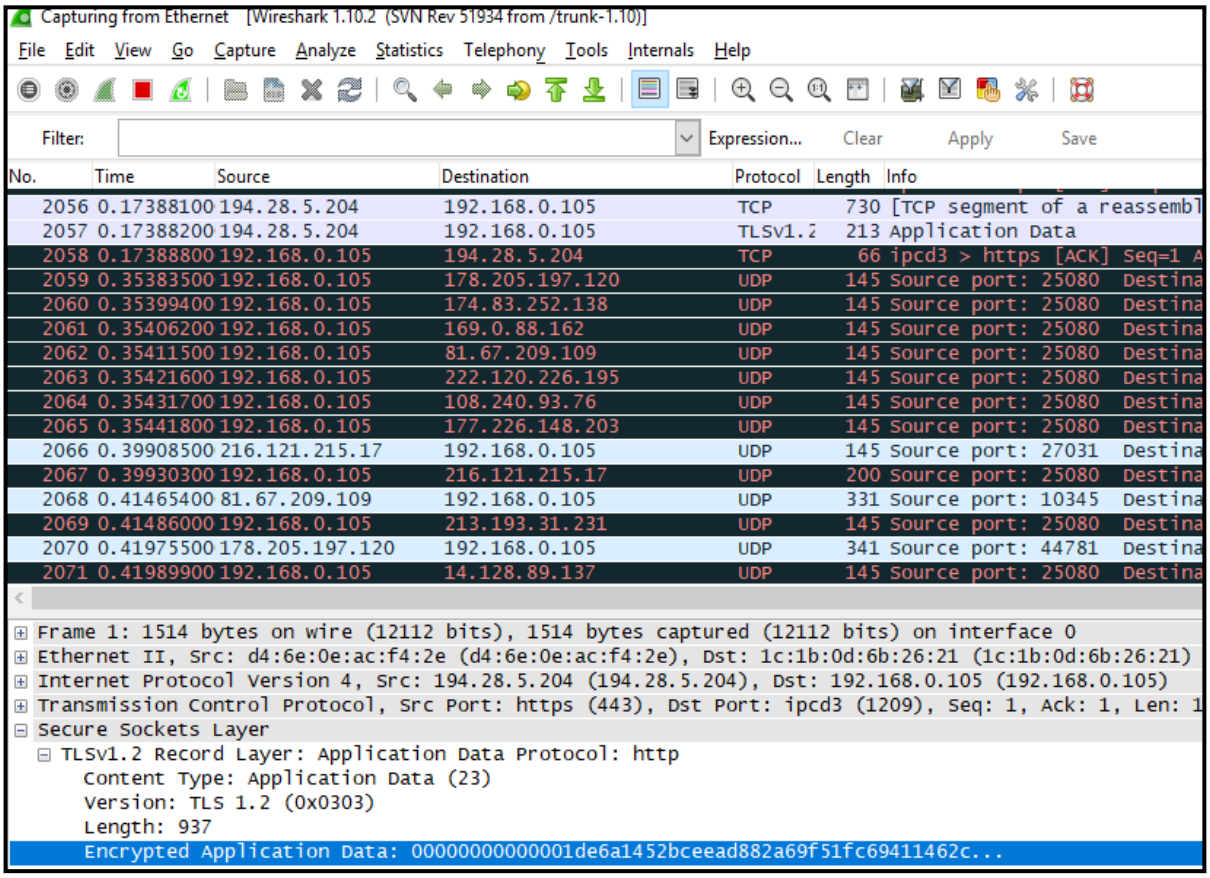

Рисунок 2.2 — Вікно перехоплення пакетів Wireshark

Wireshark може захоплювати трафік із різних типів мережевих носіїв, включаючи Ethernet, бездротову локальну мережу, Bluetooth, USB та багато іншого. Всі пакети, які були перехоплені, можна збергти собі на локальне сховище та відкрити у подальшому[25]. Wireshark може зберігати захоплені пакети у багатьох форматах, включаючи ті, що використовуються іншими програмами захоплення.

Wireshark дозволяє працювати майже із всіма загальновідомими протоколами в мережі. Зокрема можна дізнатися детальну інформацію щодо програми у відкриваючиму списку. Wireshark дозволяє фільтрувати журнал перед початком захоплення або під час аналізу, тому ви можете звузити і обнулити те, що ви шукаєте в мережевому трафіці. Наприклад, ми можемо встановити фільтр, щоб бачити трафік TCP між двома IP-адресами, що нам і буде потрібно у цій роботі. Також можна встановити лише для показу пакетів,

надісланих з одного комп'ютера. Фільтри Wireshark - одна з основних причин, через яку він став стандартним інструментом для аналізу пакетів. [24, 26].

Ці пакети можна відфільтрувати, вибрати лише необхідні протоколи, наприклад, та згодом відправити іншим у різних форматах [24, 26].

Wireshark - це проект програмного забезпечення з відкритим кодом, який випускається під загальною публічною ліцензією GNU (GPL). Ви можете вільно використовувати Wireshark на будь-якій кількості вподобаних комп’ютерів, не турбуючись про ліцензійні ключі, збори тощо. Крім того, весь вихідний код знаходиться у вільному доступі під GPL. Через це людям дуже легко додавати нові протоколи до Wireshark, або як плагіни, або вбудовані у джерело [26].

2.2 Створення та тестування офісів (без технології DMVPN)

Для основи візьмемо візьмемо один із наведених прикладів у розділі. Тобто використаємо схему із трьох офісів, де один із них буде хабом а два інших - spoke. У кожному офісі налаштована своя локальна сітка. Спочатку треба зв’язати ці роутери. Це можна зробити через Switch, щоб всі роутери дивились в одну мережу, а можна через ще один роутер, який буде виступати ніби реальним Інтернетом. Це трохи складніше однак краще демонструє можливості технології DMVPN.

Отже, створюємо базову топологію трьох офісів (поки що без IP та налаштувань) та додаємо роутер WAN, щоб наші офіси не дивились в одну локальну мережу (рис.2.3)). В подальшому, ця схема ляже в основу мого графічного інтерфейсу.

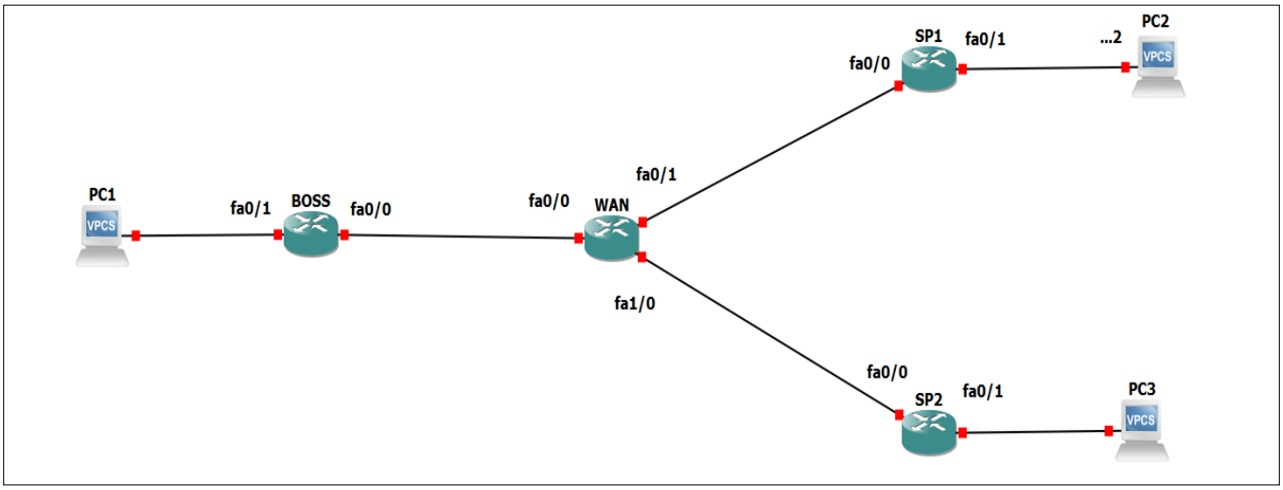

Рисунок 2.3 — Майбутня топологія офісів

Далі налаштовуємо IP-адреси на віртуальних персональних комп'ютерах (PC1, PC2, PC3) та роутерах (BOSS, WAN, SP1, SP2) (різні офіси поки що з'єднуємо за допомогою будь-якого протоколу динамічної маршрутизації. Я обрав протокол EIGRP, який треба буде дещо змінити про що пізніше (рис. 2.4)).

```
!
router eigrp 100
 network 10.10.1.0 0.0.0.255
 network 192.168.1.0
!
```

Рисунок 2.4 — налаштований протокол EIGRP на роутері BOSS

Після цього додамо IP-адреси наших мереж, щоб орієнтуватися та перевіримо нашу схему на працездатність за допомогою команди «ping» (рис.2.5).

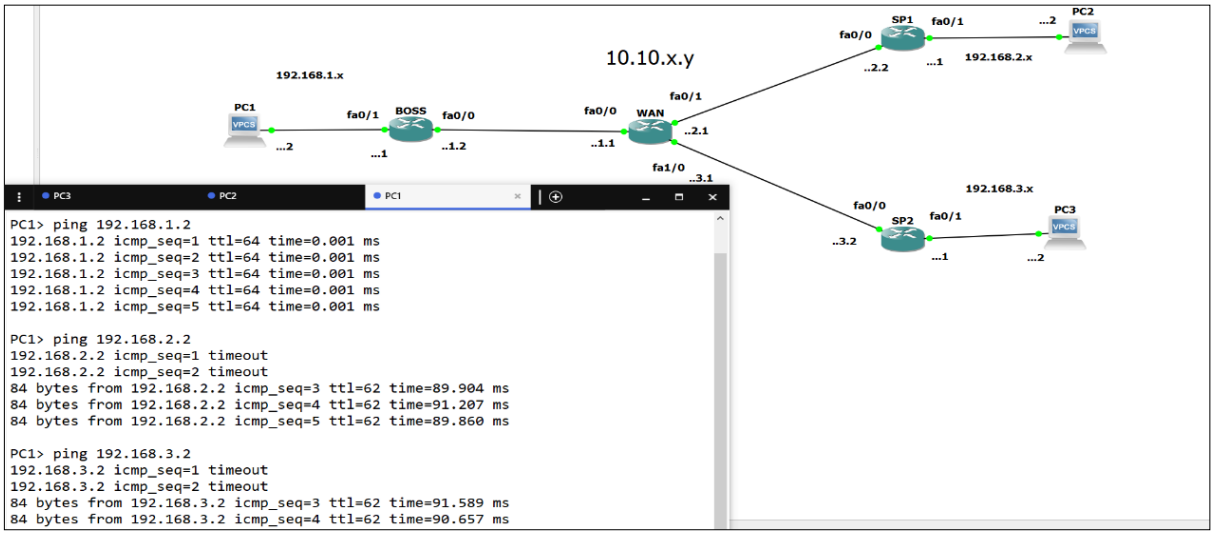

Рисунок 2.5 — Робочі 3 офіси

Якщо пакети ICMP проходять, то це означає що офіси зможуть спілкуватись між собою. Як бачимо пакети надходять, а значить схема працює. Як результат, ця схема підходить для подальшого налаштування нашого протоколу та програмної реалізації DMVPN.

Ми переконались, що схема дійсно працює і можна почати налаштовувати технологію.

2.3 Налаштування GRE over IPsec та аналіз пакетів за допомогою сніфера WireShark

Спочатку схематично проведемо майбутні тунелі. Як ми бачимо із рисунка 2.6 ми побудуємо лише 2 тунелі, але якби ми використовували технологію GRE, то знадобилось 3 тунелі. І чим більша топологія, тим більша різниця у кількість тунелів була б.

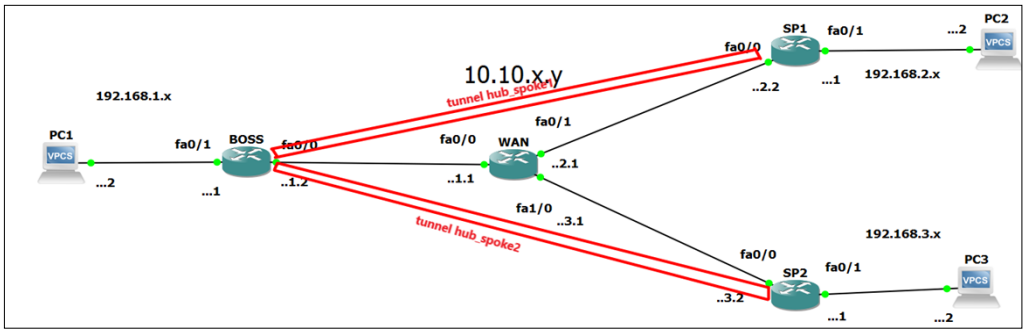

Рисунок 2.6 — Майбутні тунелі в мережі

Що ж, почнемо налаштовувати тунелі у багатоточковому режимі. Відразу будемо піднімати і протокол NHRP. Спочатку налаштуємо наш хаб (рис.2.7), а потім і spoke (SP1 (рис. 2.8) та SP2 (рис. 2.9)). Як бачимо налаштування на споках майже не відрізняється, на відміну від тих випадків коли б довелось будувати звичайний тунель GRE.

```
interface Tunnel0
 ip address 192.168.0.1 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication DMVPN
 ip nhrp map multicast dynamic
 ip nhrp network-id 1
 ip tcp adjust-mss 1360
 no ip split-horizon eigrp 100
 tunnel source FastEthernet0/0
 tunnel mode gre multipoint
 tunnel key 1234
 !
!
```

Рисунок 2.7 — налаштування тунелю на хабі

```
interface Tunnel0
 ip address 192.168.0.2 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication DMVPN
 ip nhrp map multicast dynamic
 ip nhrp map multicast 10.10.1.2
 ip nhrp map 192.168.0.1 10.10.1.2
 ip nhrp network-id 1
 ip nhrp nhs 192.168.0.1
 ip tcp adjust-mss 1360
 tunnel source FastEthernet0/0
 tunnel mode gre multipoint
 tunnel key 1234
 !
```

Рисунок 2.8 — налаштування тунелю на Spoke1(SP1)

```
interface Tunnel0
 ip address 192.168.0.3 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication DMVPN
 ip nhrp map multicast dynamic
 ip nhrp map multicast 10.10.1.2
 ip nhrp map 192.168.0.1 10.10.1.2
 ip nhrp network-id 1
 ip nhrp nhs 192.168.0.1
 ip tcp adjust-mss 1360
 tunnel source FastEthernet0/0
 tunnel mode gre multipoint
 tunnel key 1234
 !
```

Рисунок 2.9 — налаштування тунелю на Spoke2 (SP2)

Перевіримо працездатність створеного тунелю DMVPN за допомогою команди за замовчуванням «sh ip route» в консолі на роутері(рис. 2.10).

```
S*    0.0.0.0/0 [1/0] via 10.10.1.1
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.10.1.0/24 is directly connected, FastEthernet0/0
L        10.10.1.2/32 is directly connected, FastEthernet0/0
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected, Tunnel0
L        192.168.0.1/32 is directly connected, Tunnel0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, FastEthernet0/1
L        192.168.1.1/32 is directly connected, FastEthernet0/1
```

Рисунок 2.10 — перевірка тунелю на інтерфейсі роутера

Також ми бачимо, що технологія DMVPN працює.

Однак поки що ми поки що не можемо спілкуватися між локальними мережами. Це спричинене тим, що ми не розповіли протоколу динамічної маршрутизації про наш тунель. Отже просто додаємо мережу тунелю в наш протокол EIGRP (рис. 2.11). Аналогічно на інших роутерах.

```
router eigrp 100
 network 192.168.0.0
 network 192.168.1.0
!
```

Рисунок 2.11 — Повторне налаштування протоколу EIGRP

Після зміни протоколу динамічної маршрутизації, знову використаємо команду «sh ip route» в консолі на одному із роутерів. Як бачимо, нам невистачає по одній адресі на кожному споку (рис. 2.12). Це відбувається через специфіку протоколу маршрутизації EIGRP, а саме EIGRP Split Horizon Rule.

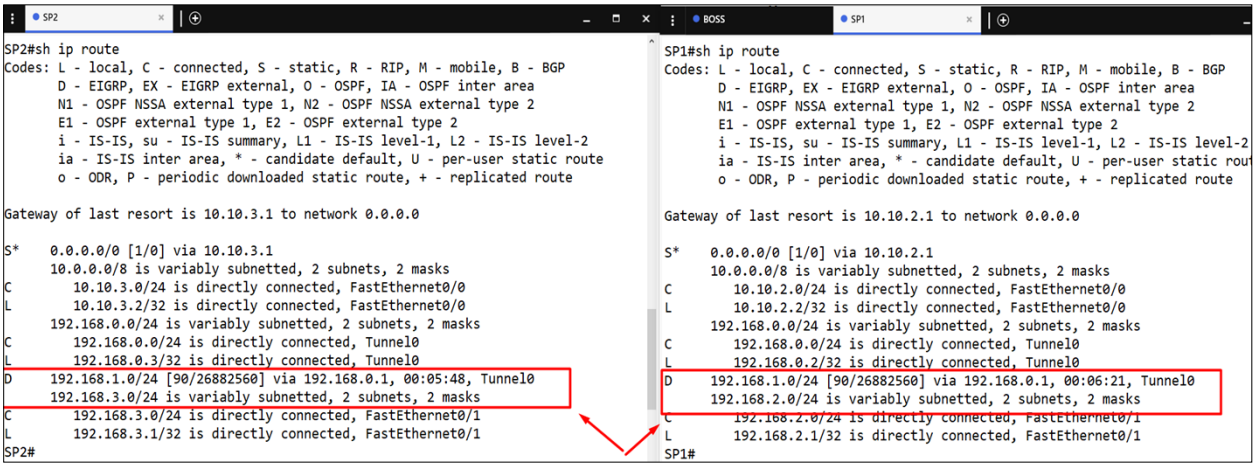

Рисунок 2.12 — Проблема EIGRP Split Horizon Rule

Правило розділеного горизонту забороняє маршрутизатору рекламувати маршрут через інтерфейс, який сам маршрутизатор використовує для досягнення пункту призначення. У DMVPN маршрутизатор концентратора вивчає маршрути з одного споку через інтерфейс Tunnel0 і повинен рекламувати його на іншому споку через той же інтерфейс Tunnel0. Таким чином, для того, щоб DMVPN працював з EIGRP, розділений горизонт повинен бути відключений на інтерфейсі тунелю за допомогою команди [27].

Що ж налаштуємо та спробуємо ще раз команду «sh ip route» (рис. 2.13).

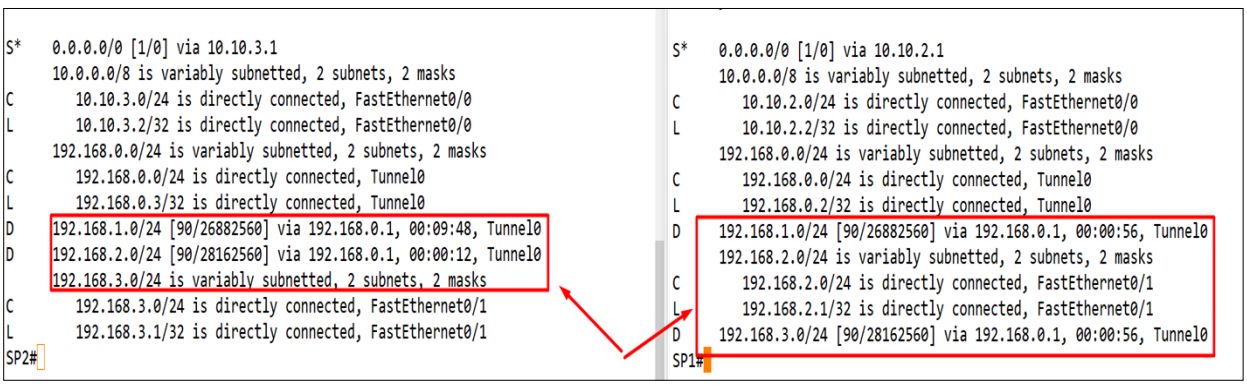

Рисунок 2.13 — Вимкнений EIGRP Split Horizon Rule

Тепер за допомогою команди «sh dmvpn» подивимось чи працює технологія DMVPN (рис. 2.14).

```
BOSS#sh dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface: Tunnel0, IPv4 NHRP Details
Type:Hub, Total NBMA Peers (v4/v6): 2

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1      10.10.2.2      192.168.0.2    UP 00:05:23     D     192.168.0.2/32

    1      10.10.3.2      192.168.0.3    UP 00:05:53     D     192.168.0.3/32
```

Рисунок 2.14 — Маршрути DMVPN на хабі

Після цього у передаємо пакет та подивимось чи можуть спілкуватись роутери через тунель (рис. 2.15).

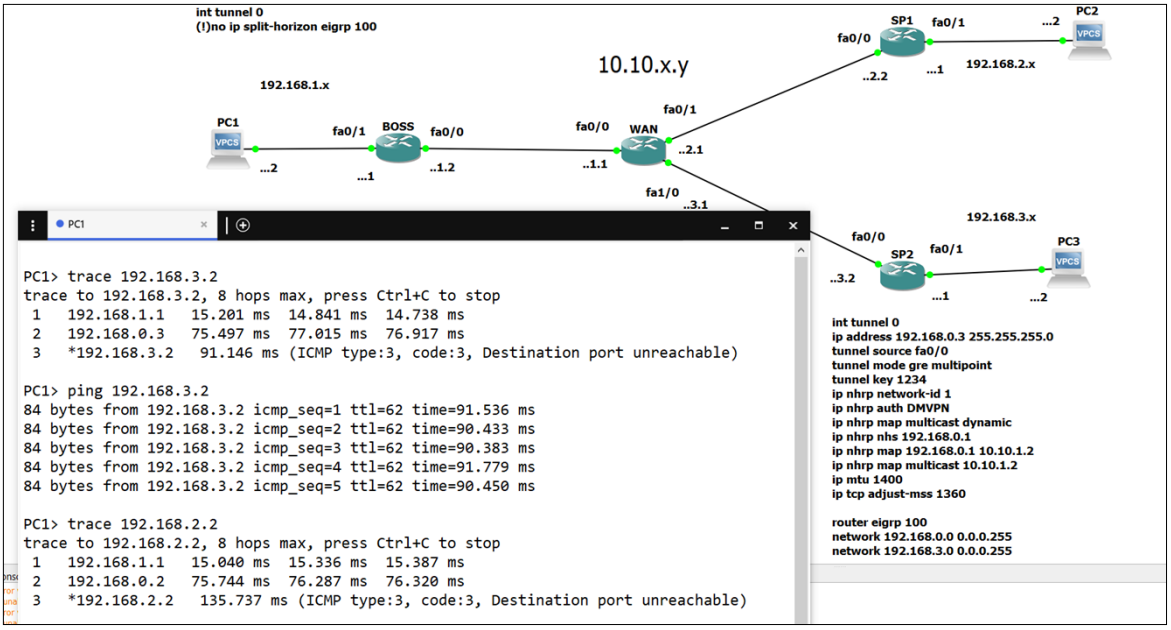

Рисунок 2.15 — Ping та traceroute з PC1

Що ж дійсно, віртуальні комп’ютери змогли зв’язатись між собою через тунель.

Відразу налаштуємо якість обслуговування (QoS) для нашої мережі. За основу я візьму розмежування основного трафіку та голосового, якому дам 10 відсотків від максимально можливої пропускної здатності. Звісно, можна налаштовувати і набагато складніші політики якості обслуговування, однак зараз наша мета дослідити як це працює в технології DMVPN.

Налаштовуємо політику якості на хабі (рис. 2.16).

```
class-map match-all voip
 match protocol rtp
!
!
policy-map child
 class voip
    priority percent 10
  set dscp ef
 class class-default
  set dscp default
    queue-limit 1000 packets
    bandwidth remaining percent 90
policy-map spoke-qos
 class class-default
    shape average 100000000
  service-policy child
!
```

Рисунок 2.16 — Політика якості на хабі

Застосовуємо політики до вже готового тунелю DMVPN (рис. 2.17), а на споках лише налаштовуємо DMVPN інтерфейс (рис. 2.18), який ви ми будемо застосувати до нього. На різних споках можна використовувати різні групи політик, якщо вони налаштовані на хабі.

```
interface Tunnel0
 ip address 192.168.0.1 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication DMVPN
 ip nhrp map multicast dynamic
 ip nhrp map group QoS service-policy output spoke-qos
 ip nhrp network-id 1
 ip tcp adjust-mss 1360
 no ip split-horizon eigrp 100
 tunnel source FastEthernet0/0
 tunnel mode gre multipoint
 tunnel key 1234
 !
!
```

Рисунок 2.17 — налаштування якості на тунелі хабу

```
interface Tunnel0
 ip address 192.168.0.2 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication DMVPN
 ip nhrp group QoS
 ip nhrp map multicast dynamic
 ip nhrp map multicast 10.10.1.2
 ip nhrp map 192.168.0.1 10.10.1.2
 ip nhrp network-id 1
 ip nhrp nhs 192.168.0.1
 ip tcp adjust-mss 1360
 tunnel source FastEthernet0/0
 tunnel mode gre multipoint
 tunnel key 1234
 !
```

Рисунок 2.18 — Налаштування якості на тунелі споуку

За допомогою команди «show dmvpn detail» побачимо чи налаштувалися політики якості обслуговування (рис. 2.19).

```
BOSS#sh dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding
        UpDn Time --> Up or Down Time for a Tunnel
==============================================================================

Interface Tunnel0 is up/up, Addr. is 192.168.0.1, VRF ""
   Tunnel Src./Dest. addr: 10.10.1.2/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect ""
   Interface State Control: Disabled
Type:Hub, Total NBMA Peers (v4/v6): 2

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1      10.10.2.2     192.168.0.2    UP 01:05:29    D     192.168.0.2/32
NHRP group: QoS
 Output QoS service-policy applied: spoke-qos

    1      10.10.3.2     192.168.0.3    UP 01:05:59    D     192.168.0.3/32
NHRP group: QoS
 Output QoS service-policy applied: spoke-qos
```

Рисунок 2.19 — Маршрути DMVPN на хабі

Тепер настав час подивитись пакети, які передаються та зрозуміти чи можна їх перехопити та прочитати. Для цього використаємо сніфер WireShark (рис. 2.20).

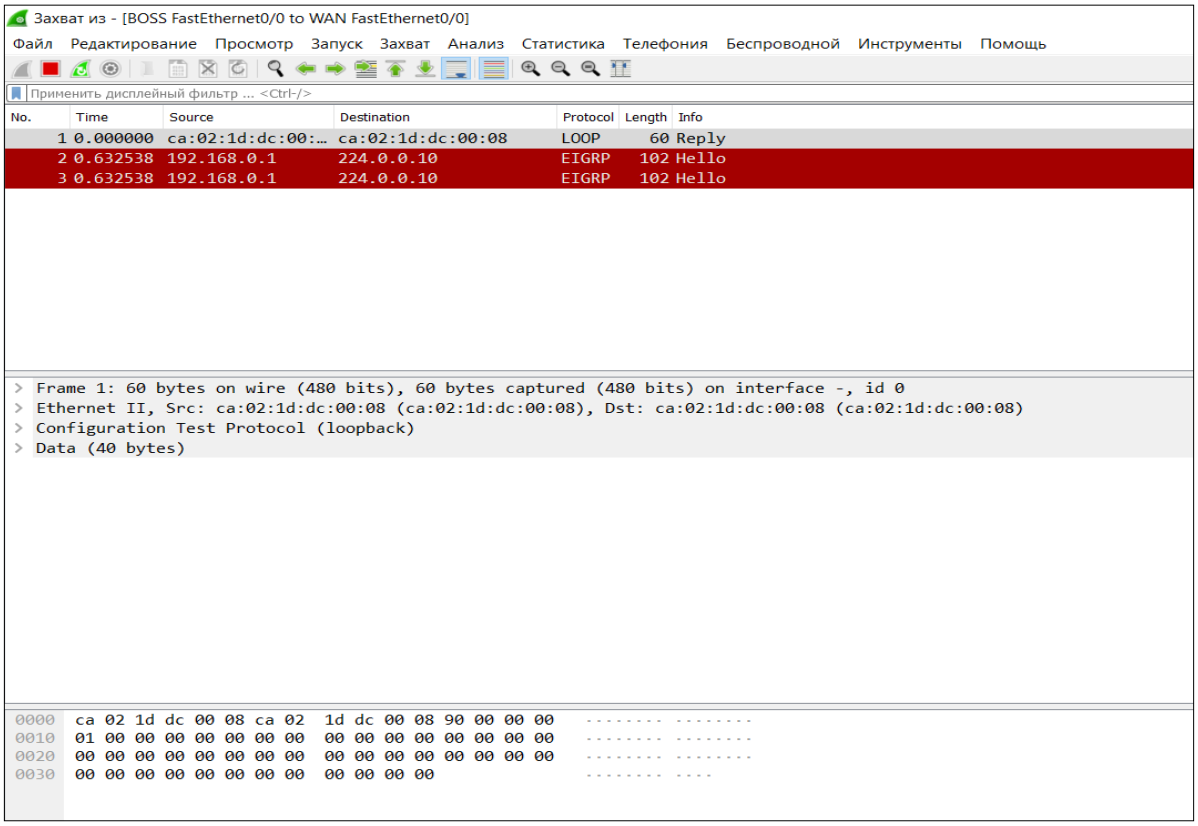

Рисунок 2.20 — Сніфер WireShark

Спочатку перевіримо налаштовану якість обслуговування, для цього нам потрібні пакети із протоколом NHRP (рис. 2.21).

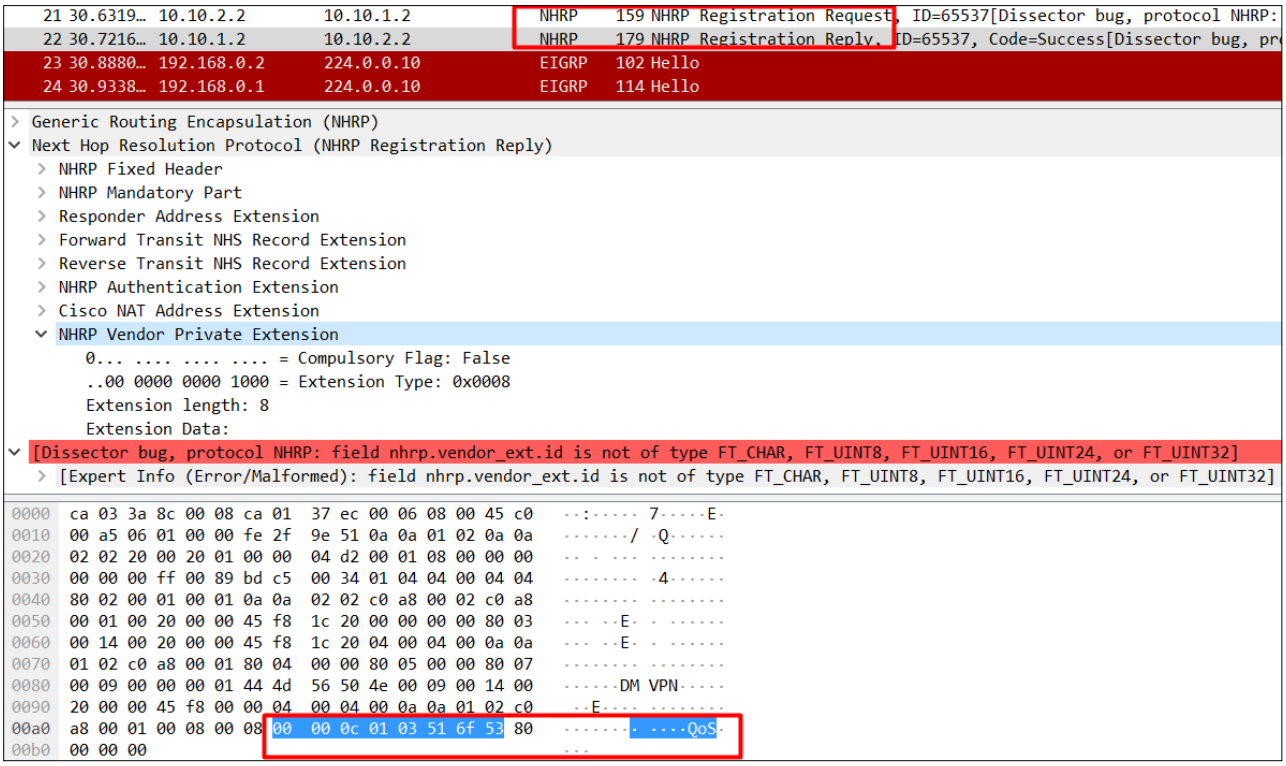

Рисунок 2.21 — Налаштована якість обслуговування

Як видно із рисунка 2.21 наші маршрутизатори почали працювати за створеними правилами якості обслуговування.

Тепер відправимо команду «ping», але UDP щоб точно знати тип пакету. Тепер подивимось, чи зможемо ми прочитати заголовок та дані пакету за допомогою сніферу (рис. 2.22).

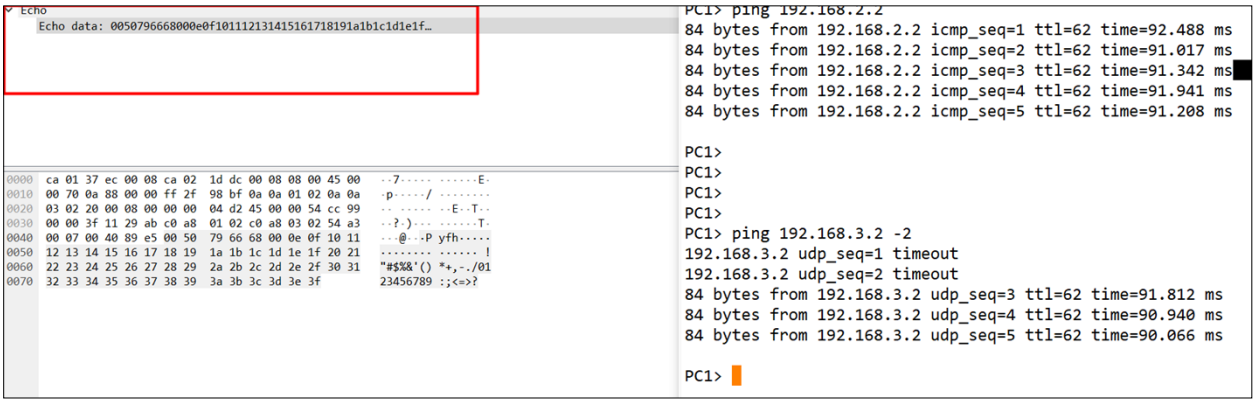

Рисунок 2.22 — Перехоплений пакет та інформація в ньому

Як бачимо, нам видно і заголовок, і повністю всі дані, що передаються під цим заголовком, а отже дані треба буде захищати протоколом шифрування. Нам на допомогу приходить набір протоколів IPsec, який є частиною технології DMVPN.

2.4 Налаштування DMVPN over IPsec та аналіз пакетів за допомогою симулятора GNS3 та сніферу Wireshark

Ми бачимо, що налаштований «голий» зовсім не захищає передачу даних. Тепер же скористаємось можливістю шифрування передачі пакетів. Для цього використаємо надійний протокол Advanced Encryption Standard (AES) - симетричний алгоритм блочного шифрування [1,2]. Цей тип шифрування, як і ще десяток, є доступний завдяки набору протоколів IPsec. Так як схема вже готова, нам потрібно лише правильно налаштувати шифрування наших пакетів.

Налаштуємо для цього спочатку команди на хабі (рис. 2.23), а потім на споках (рис. 2.24). Треба зазначити, що аутентифікація по pre-shared key хоч і однаковий для налаштування на всіх вузлах є не зовсім надійний, так як за замовчуванням використовується аутентифікація за сертифікатами. Через це з'являється небезпека, що встановити IPSec-сесію з хабом, знаючи ключ, може будь-який пристрій. Альтернативою стають сертифікати, які дещо складно буде продемонструвати без віртуальної машини в GNS3.

Перевіримо набрані команди на роутері за допомогою консолі та команди «show dmvpn detail» (рис. 2.25).

```
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key PASSWORD address 0.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN_IPSEC
 set transform-set AES256-SHA
!
```

Рисунок 2.23 — результат набору команд на хабі

```
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key PASSWORD address 0.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN_IPSEC
 set transform-set AES256-SHA
!
```

Рисунок 2.24 — результат набору команд на споках

```
BOSS#sh dmvpn detail
# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1      10.10.2.2     192.168.0.2    UP 00:07:30    D     192.168.0.2/32
NHRP group: QoS
 Output QoS service-policy applied: spoke-qos

    1      10.10.3.2     192.168.0.3    UP 00:01:00    D     192.168.0.3/32
NHRP group: QoS
 Output QoS service-policy applied: spoke-qos



Crypto Session Details:
--------------------------------------------------------------------------------

Interface: Tunnel0
Session: [0x683FC82C]
  IKE SA: local 10.10.1.2/500 remote 10.10.2.2/500 Active
          Capabilities:(none) connid:1001 lifetime:23:52:28
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 10.10.2.2
  IPSEC FLOW: permit 47 host 10.10.1.2 host 10.10.2.2
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 106 drop 0 life (KB/Sec) 4485668/3148
        Outbound: #pkts enc'ed 108 drop 0 life (KB/Sec) 4485668/3148
   Outbound SPI : 0xA84966C6, transform : esp-256-aes esp-sha-hmac
    Socket State: Open
```

Рисунок 2.25 — перевірка налаштованого шифрування

Після цього знову у програмі Wireshark спробуємо знову переглянути пакети. Тепер вже доведеться розшифровувати пакет, щоб прочитати в ньому ієрархію заголовків та його зміст (рис. 2.26).

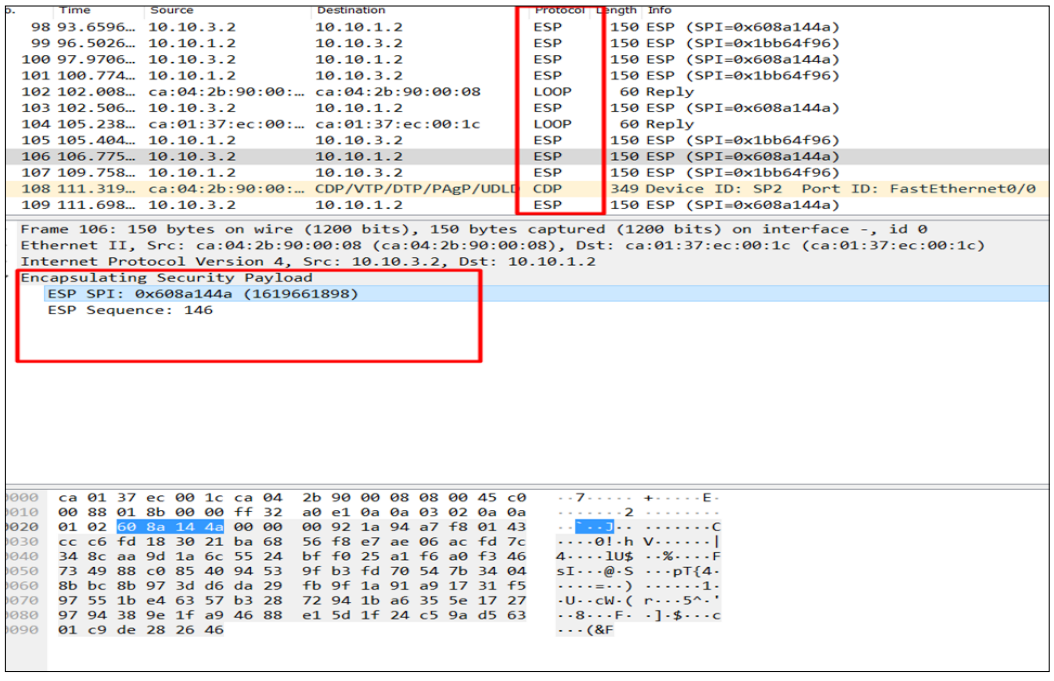

Рисунок 2.26 — Дані із пакета в програмі Wireshark після використання IPsec протоколу

Тепер всі протоколи передачі данних мають заголовок ESP (Encapsulating Security Payload), а самі пакети мають зашифровані тідзаголовки та основні дані із пакету також зашифровані.

Отже, дійсно IPsec протоколи допомагають захистити дані в пакеті, але ця технологія має і свої недоліки. Подивимось тепер, чи сильно алгоритм шифрування вплинув на швидкість передачі даних (рис. 2.27). порівнюючи із результатом на рисунку 2.15 ми бачимо, що час відгуку збільшився приблизно на декілька мілісекунд що не критично в нашому випадку.

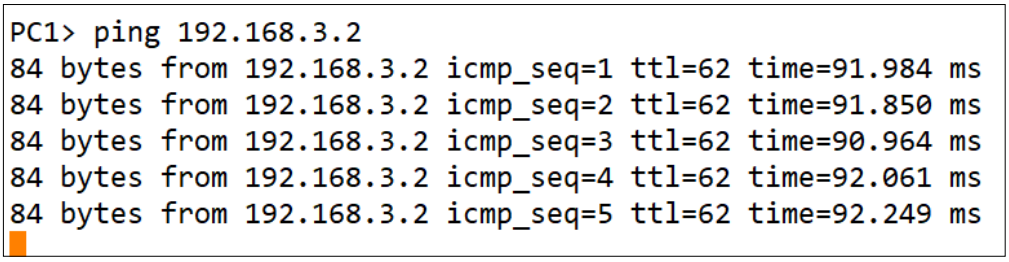

Рисунок 2.27 — Аналіз пакетів при використанні IPsec протоколу

Якщо ж є бажання змінити тип шифрування, можна розглянути AES з іншою бітністю. Або ж використати інший алгоритм, зокрема 3DES, який вже на сьогоднішній день він вже є застарілим. На сьогоднішній день AES є самим надійним алгоритмом шифрування.

2.5 Налаштування різних фаз технології DMVPN за допомогою симулятора GNS3

Тепер, коли технологія DMVPN over IPsec налаштована, треба спробувати сконфігурувати її різні фази та знайти найкращу для нашого випадку. Треба сказати, що на даний, що на даний повноцунно працює лише перша фаза технології, хоча на споках вже встановлена властивість gre multipoint (при базовому налаштуванню фази 1 на споках пунктом призначення пишеться справжня IP-адреса тунелю хаба). Це призводить до того, що при набиранні команд «show dmvpn» (рис.2.28) та «trace 192.168.3.1» (рис. 2.29) на роутері споука SP1 бачимо, що ми все рівно побудували статичний маршрут.

```
SP1#sh dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding
        UpDn Time --> Up or Down Time for a Tunnel
==============================================================================

Interface: Tunnel0, IPv4 NHRP Details

IPv4 NHS: 192.168.0.1 RE
Type:Spoke, Total NBMA Peers (v4/v6): 1

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- -----  -----------------
    1      10.10.1.2     192.168.0.1    UP 00:00:01    S    192.168.0.1/32
```

Рисунок 2.28 — Show dmvpn на даний момент

```
SP1#trace 192.168.3.1

Type escape sequence to abort.
Tracing the route to 192.168.3.1

  1 192.168.0.1 68 msec 24 msec 60 msec
  2 192.168.0.3 100 msec 104 msec 88 msec
```

Рисунок 2.29 — Маршрут від SP1 до SP2

А це означає, що наші пакети через хаб а вже потім йде на інший спок, в нашому випадку - SP2 (рис. 2.30).

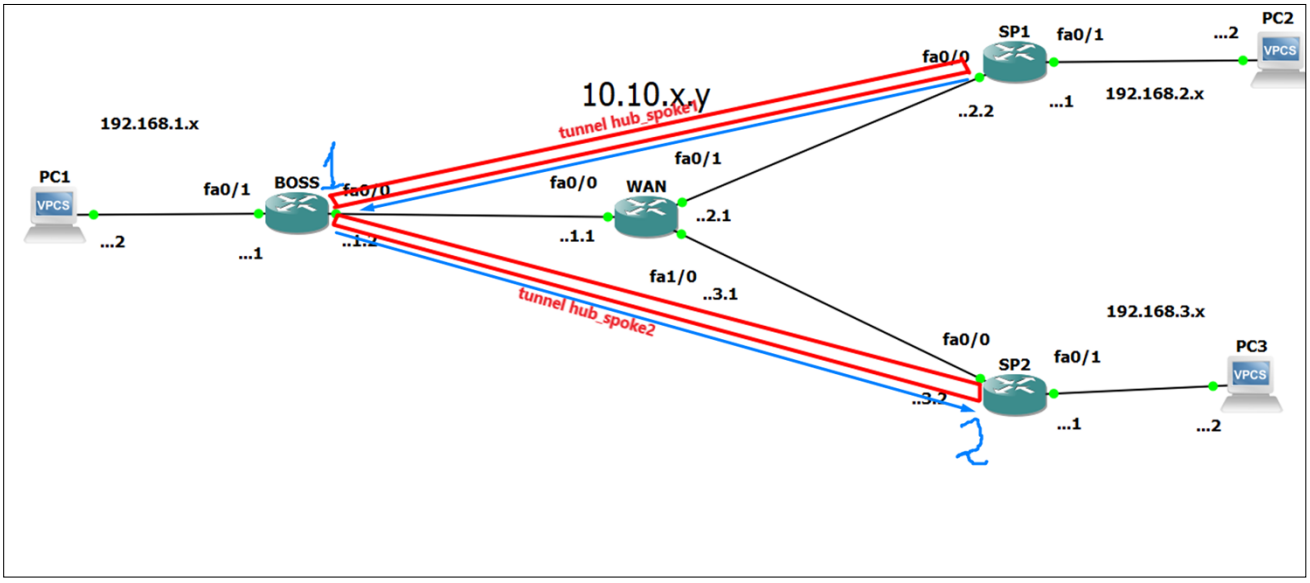

Рисунок 2.30 — Маршрут передачі пакетів між споками

Це знову ж таки спричинено протоколом динамічної маршрутизації EIGRP. Додаємо налаштування на хабі (потрібне лише на другій фазі) (рис. 2.31).

```
interface Tunnel0
 ip address 192.168.0.1 255.255.255.0
 no ip redirects
 ip mtu 1400
 no ip next-hop-self eigrp 100
 ip nhrp authentication DMVPN
 ip nhrp map multicast dynamic
 ip nhrp map group QoS service-policy output spoke-qos
 ip nhrp network-id 1
 ip tcp adjust-mss 1360
 no ip split-horizon eigrp 100
 tunnel source FastEthernet0/0
 tunnel mode gre multipoint
 tunnel key 1234
 tunnel protection ipsec profile DMVPN_IPSEC
 !
```

Рисунок 2.31 — налаштування другої фази

Знову перевіряємо атрибут нашого тунелю (рис. 2.32).

```
SP1#sh dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface: Tunnel0, IPv4 NHRP Details

IPv4 NHS: 192.168.0.1 RE
Type:Spoke, Total NBMA Peers (v4/v6): 2

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1      10.10.1.2     192.168.0.1    UP 00:04:36    S     192.168.0.1/32

    1      10.10.3.2     192.168.0.3    UP 00:03:50    D     192.168.0.3/32


SP1#trace 192.168.3.1

Type escape sequence to abort.
Tracing the route to 192.168.3.1

  1 192.168.0.3 56 msec 56 msec 56 msec
```

Рисунок 2.32 — тунель між споками

Як бачимо, тепер наші споки спілкувати напряму, що зберігає час на передачу пакету. Це стало можливим завдяки тому, що після запитів споків вони знають IP-адреси NBMA один одного і мають прямий тунель DMVPN (рис. 2.33).

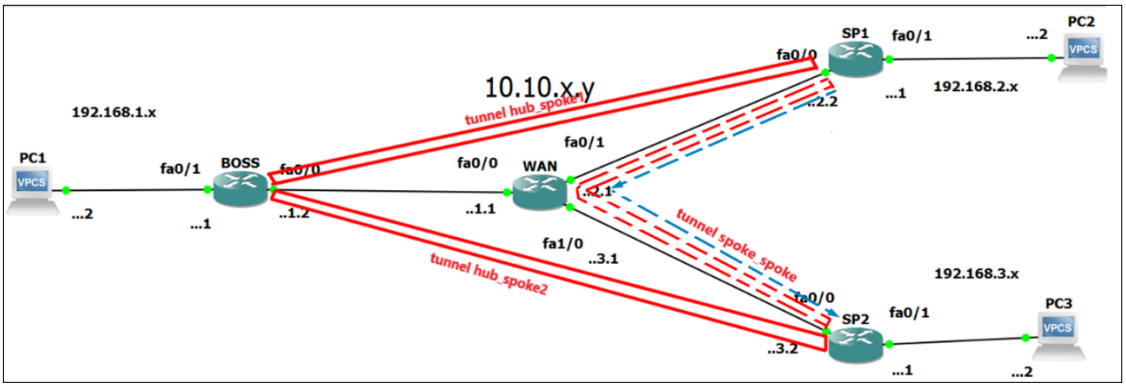

Рисунок 2.33 — Новий маршрут передачі пакетів між споками

Тепер у нас повністю налаштована фаза 2. Перейдемо до фази 3. Тут все значно легше для налаштування. Фаза 3 дозволяє підсумовувати оновлення протоколу маршрутизації від нашого хабу до споків. Споки більше не потребують індивідуального маршруту з next hop`ом IP тунельної IP-адреси віддаленого спікера для мереж, що стоять за всіма іншими споками. Налаштовуємо наш хаб (рис. 2.34) та споки (на спокові команда «ip nhrp redirect» не обов’язкова) (рис. 2.35).

```
interface Tunnel0
 ip address 192.168.0.1 255.255.255.0
 no ip redirects
 ip mtu 1400
 no ip next-hop-self eigrp 100
 ip nhrp authentication DMVPN
 ip nhrp map multicast dynamic
 ip nhrp map group QoS service-policy output spoke-qos
 ip nhrp network-id 1
 ip nhrp holdtime 300
 ip nhrp redirect
 ip tcp adjust-mss 1360
 no ip split-horizon eigrp 100
 tunnel source FastEthernet0/0
 tunnel mode gre multipoint
 tunnel key 1234
 tunnel protection ipsec profile DMVPN_IPSEC
 !
```

Рисунок 2.34 — Налаштування третьої фази на хабі

```
!
interface Tunnel0
 ip address 192.168.0.3 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication DMVPN
 ip nhrp group QoS
 ip nhrp map multicast dynamic
 ip nhrp map 192.168.0.1 10.10.1.2
 ip nhrp map multicast 10.10.1.2
 ip nhrp network-id 1
 ip nhrp nhs 192.168.0.1
 ip nhrp shortcut
 ip nhrp redirect
 ip tcp adjust-mss 1360
 tunnel source FastEthernet0/0
 tunnel mode gre multipoint
 tunnel key 1234
 tunnel protection ipsec profile DMVPN_IPSEC
!
```

Рисунок 2.35 — Налаштування третьої фази на споках

Для перевірки скористаємось командами «show dmvpn» та «show ip route» (рис.2.36).

```
SP2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 10.10.3.1 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.10.3.1
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.10.3.0/24 is directly connected, FastEthernet0/0
L        10.10.3.2/32 is directly connected, FastEthernet0/0
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected, Tunnel0
L        192.168.0.3/32 is directly connected, Tunnel0
D     192.168.1.0/24 [90/26882560] via 192.168.0.1, 00:06:14, Tunnel0
D  %  192.168.2.0/24 [90/28162560] via 192.168.0.2, 00:06:14, Tunnel0
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/24 is directly connected, FastEthernet0/1
L        192.168.3.1/32 is directly connected, FastEthernet0/1
SP2#
SP2#sh dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface: Tunnel0, IPv4 NHRP Details
Type:Spoke, NHRP Peers:2,

 # Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     1 10.10.1.2           192.168.0.1    UP 00:02:11     S
     2 10.10.2.2           192.168.0.2    UP 00:02:30     D
                           192.168.0.2    UP 00:02:30   DT2
```

Рисунок 2.36 — Перевірка фази 3

Як бачимо, все чудово працює однак потрібно пам’ятати, що роутери спочатку встановити зв'язок між собою (можна зробити командою «ping» «traceroute»).Також було виявлено, що на деяких роутерах може не працювати третя фаза (проблема з прошивками).

Що ж тепер можна приступити до створення графічного інтерфейсу для нашої схеми.

2.6 Мова програмування JavaScript для створення графічного інтерфейсу налаштування технології

Так як графічний інтерфейс буде у вигляді веб-додатку, для його розробки використана мова розмітки HTML, мови стилів CSS та мови програмування JavaScript, про яку дещо детальніше.

Так як користувач буде вз’яємодіяти із інтерфейсом, нам потрібно вміти аналізувати введені дані від нього та видавати результат. Це можливо зробити досягти завдяки так званим «скриптам» у мові JavaScript. Крім того нам буде необхідний лише браузер, що дозволить користуватися графічним інтерфейсом на великій кількості девайсів. Ці скрипти починають виконуються вже під час завантаження сторінки і виконується на боці клієнта, а отже нам не буде потрібен навіть доступ в мережу Інтернет. Як результат, ми можемо працювати в одному сеансі без перезавантаження сторінки. Це призводить до того, що для роботи потрібен лише інтерпретатор, що дозволяє запустити програму навіть без браузера. В нашому випадку, під час відправлення результатів можна перевірити правильність введених IP-адресів та масок, якщо значення виходять за ці рамки відповідають очікуваним, заборонити відправлення даних або повідомити про це [28].

JavaScript дає можливість доступу до використання необхідних додатків. Основною властивістю мови програмування є можливість створення сценаріїв.

Сучасний JavaScript - це «безпечний» мову програмування. Він не надає низькорівневий доступ до пам'яті або процесору, тому що спочатку був створений для браузерів, які не потребують цього.

Можливості JavaScript сильно залежать від оточення, в якому він працює. Наприклад, Node.JS підтримує функції читання / запису довільних файлів, виконання мережевих запитів і т.д.

У браузері для JavaScript є все, що пов'язано з маніпулюванням веб-сторінками, взаємодією з користувачем і веб-сервером.

Наприклад, в браузері JavaScript може:

- Додавати новий HTML-код на сторінку, змінювати існуючий вміст, модифікувати стилі.
- Реагувати на дії користувача, клацання миші, перемістити вказівник, натискання клавіш, міняти стилі елементів, міняти сторінку, писати на ній текст, додавати і видаляти теги.
- Дозволяє перевірити значення полів форм HTML до відправки на сервер.
- Отримувати і встановлювати куки, задавати питання відвідувачеві, показувати повідомлення.
- Запам'ятовувати дані на стороні клієнта («local storage»).
- Багато мов можуть бути «транспілірovані» в JavaScript для надання додаткових функцій.
- JavaScript дозволяє робити повідомлення (alert),які покажуть застереження або допоможуть знайти помилку при програмуванні або ж при введенні даних [28, 29].

Підсумовуючи можна сказати, що можливості не оновлювати веб-сторінку та працювати без доступу в Інтернет (потрібен лише сам браузер) стали вирішальними у виборі мови програмування. Зазвичай JavaScript підтримується всіма поширеними браузерами і включений за замовчуванням, але бувають випадки, коли Javascript вимкнений у вашому веб-браузері. В такому випадку нам потрібно лише дозволити виконувати скрипти у нашому браузері, щоб програма запрацювала корректно [28, 29].

3 СТВОРЕННЯ СХЕМИ GRE OVER IPSEC В ПРОГРАМНОМУ СЕРЕДОВИЩІ

3.1 Розробка графічного інтерфейсу налаштування DMVPN over IPsec

При створенні схеми у емуляторі GNS3 великим було витрачего багато часу на рутинні задачі, такі як включення інтерфейсів, перевірка правильності введених IP-адресу, тощо. Крім того, емулятор мереж не має графічного інтерфейсу налаштування, що збільшує складність налаштування схеми. Через цезадача розробки веб-орієнтованого графічного інтерфейсу може стати дуже зручним інструментом для автоматичної конфігурації динамічної маршрутизації.

Проект, як вже було зазначено, створений за допомогою мови програмування JavaScript. Повний код написаного веб-додатку, написаного на javascript, html та css можна знайти у додатку Б.

Інтерфейс веб-орієнтованого графічного інтерфейсу створювався інтуїтивно зрозумілим для користувачів, які не знайомі із командами налаштування інтерфейсів. За основу була взята вже створена схема у эмуляторі мереж GNS3 (рис. 3.1).

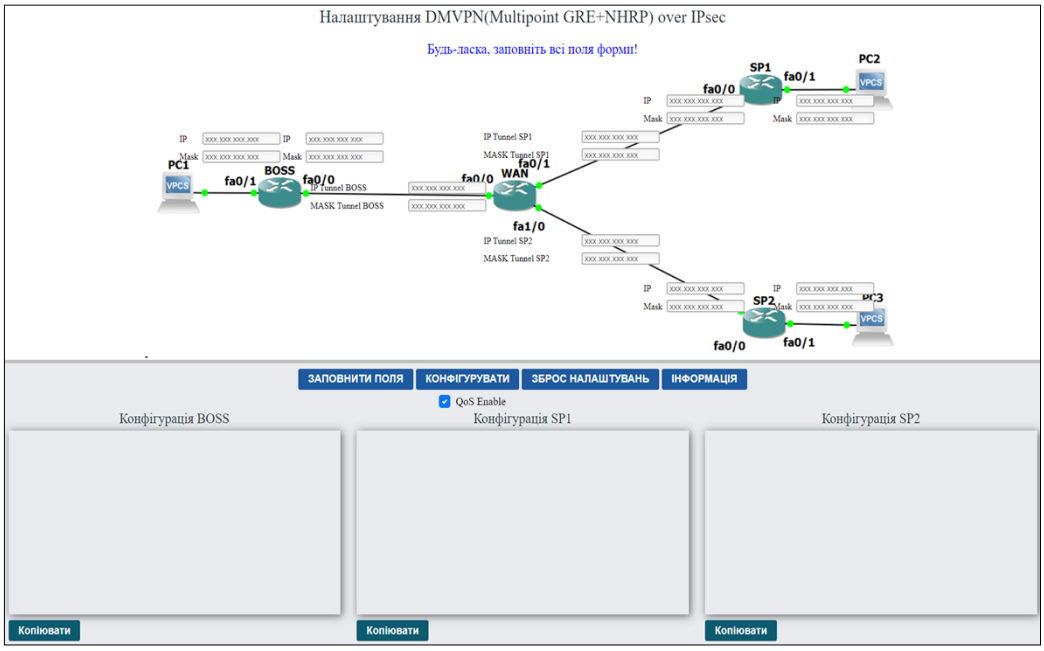

Рисунок 3.1 – Інтерфейс веб-орієнтованого графічного інтерфейсу

Після відкриття проекту на рис 3.1 побачимо схему налаштування тенології DMVPN, пусті поля для заповнення (IP та маски наших роутерів, а також пусті поля для майбутніх тунелів на офісах), чотири кнопки для роботи з програмою, один чекбокс для включення чи виключення команд якості обслуговування та три поля для отримання конфігурації на головному офісі і на двох споках (за аналогією можна налаштувати безліч офісів споків), а також три кнопки «Копіювати».

Веб-додаток перевіряє введені IP-адреси та маски у відповідні поля на валідність. Якщо була здійснена помилка при наборі, з’явиться повідомлення про неправильно введений IP чи маску та не дозволить згенерувати код налаштування маршрутизатора для схеми (рис. 3.2).

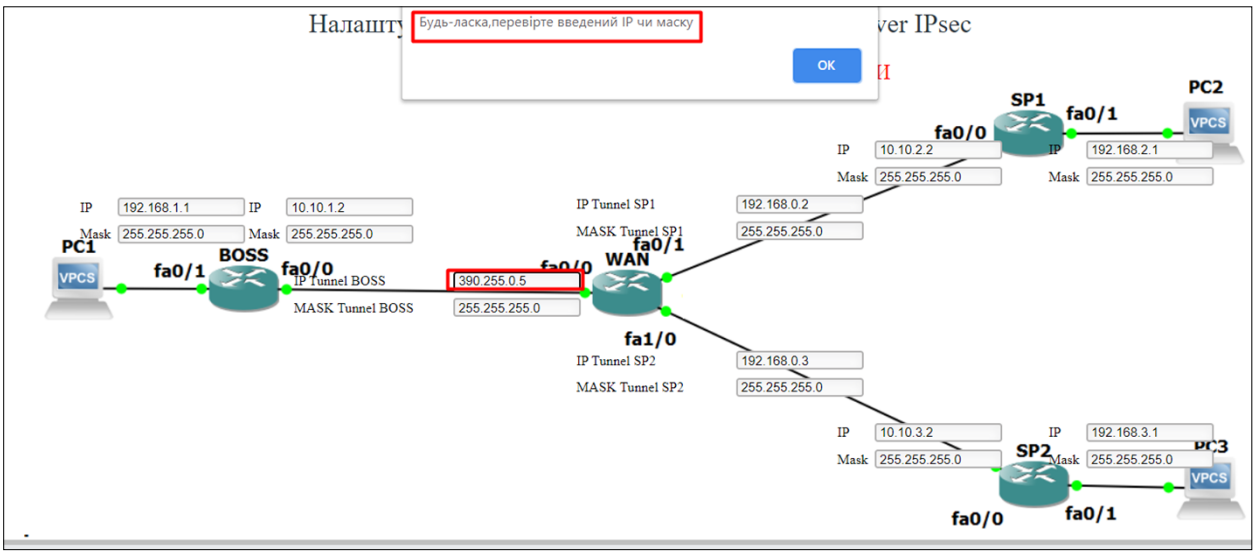

Рисунок 3.2 – Валідація на невірний формат IP адреси або маски

Якщо ігнорувати попередження та продовжити, натиснуши кнопку «Конфігурація», то відразу покажеться повідомлення, що IP чи маска – неправильна чи неповна, а після цього відбудеться затирання вже створених конфігурації, щоб не виводити неправильну конфігурацію (Рис. 3.2). Окрім цього створене поле інформації, що дозволяє прочитати чи є знайдені помилки у конфігурації. (Треба заповнити поле, всі поля відповідають валідації, одне чи більше полів не відповідають валідації (Рис. 3.3)).

Однак, якщо потрібно згенерувати саме з неправильною IP- адресою чи маскою, треба ще раз натиснути «Конфігурація», однак в такому випадку работа схеми не гарантується.

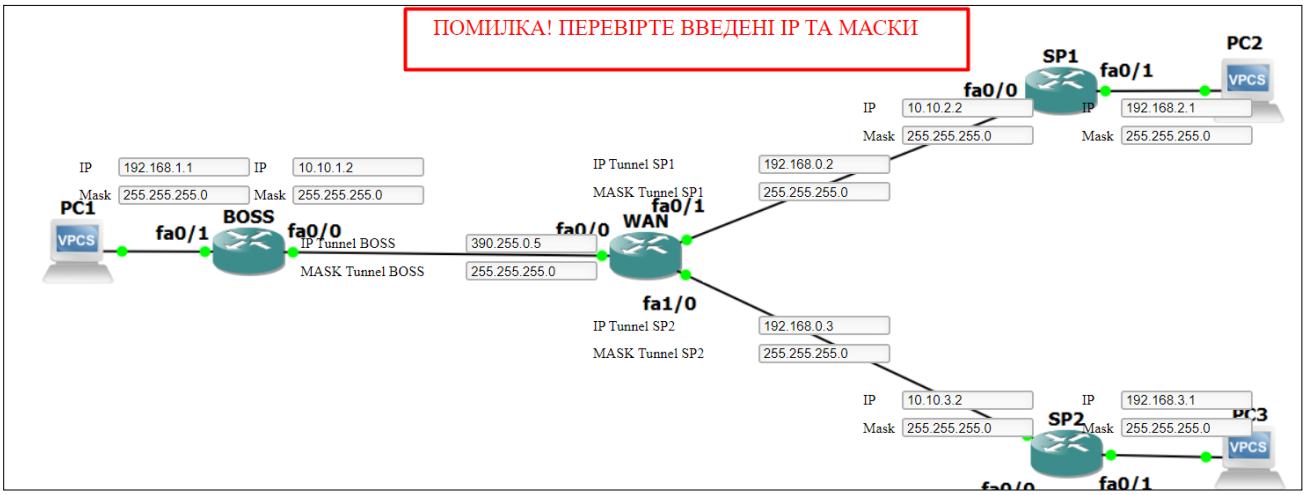

Рисунок 3.3 – Приклад роботи текстового полю

Також була додана перевірка IP-адресів на унікальність. Це означає що на одному роутері не буде однакових адрес як і на тунельних інтерфейсах маршрутизаторів (рис. 3.4).

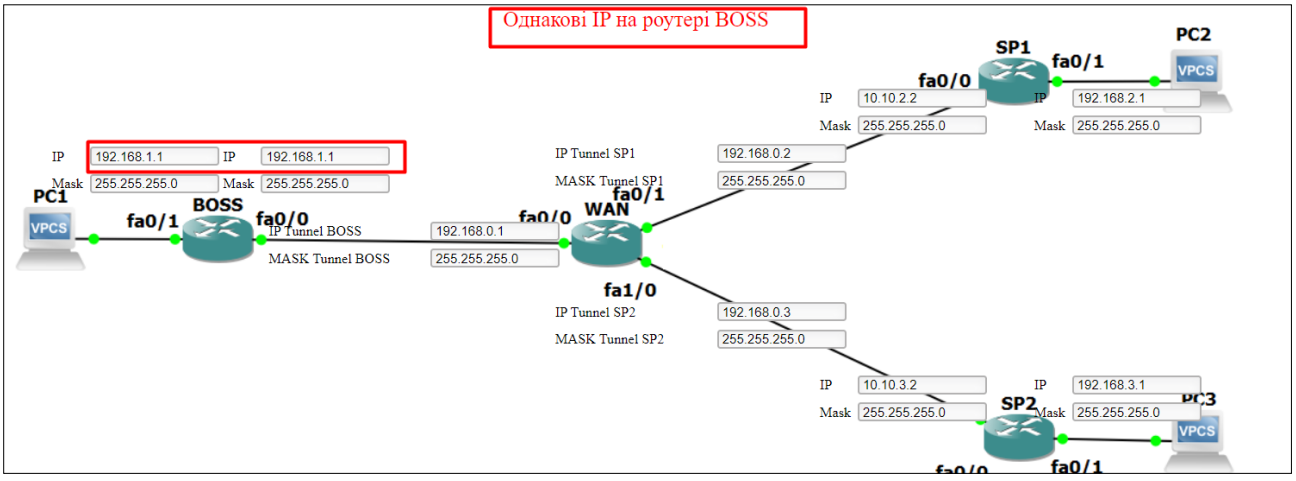

Рисунок 3.4 – Перевірка IP-адрес на унікальність

Проте однакові адреси можуть бути на внутрішніх інтерфейсах роутерів (рис. 3.5).

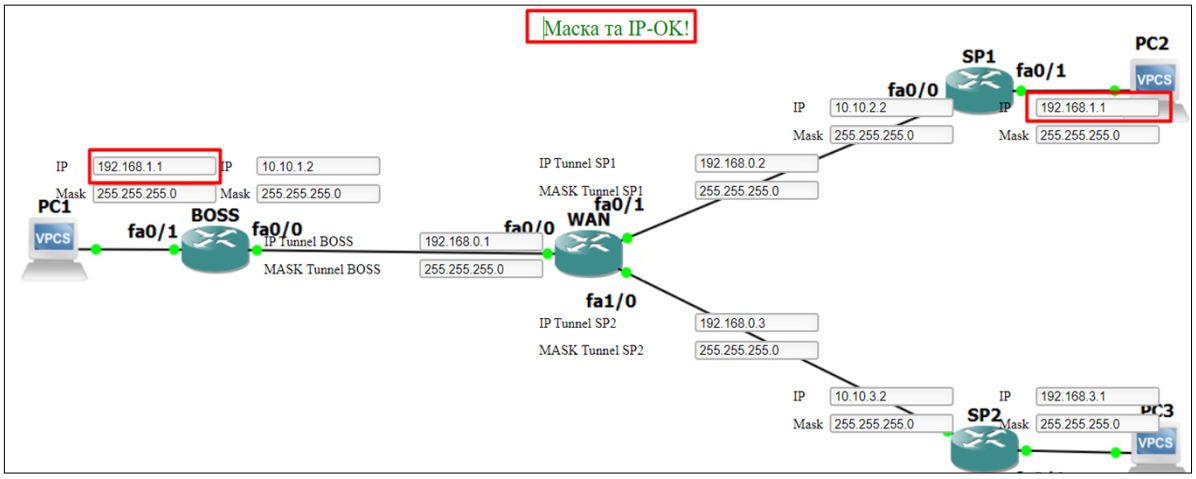

Рисунок 3.5 – Перевірка IP-адрес на унікальність

Щоб уникнути складностей із набором IP-адрес та масок, можна скористатись кнопкою «Заповнити поля» (рис. 3.6).

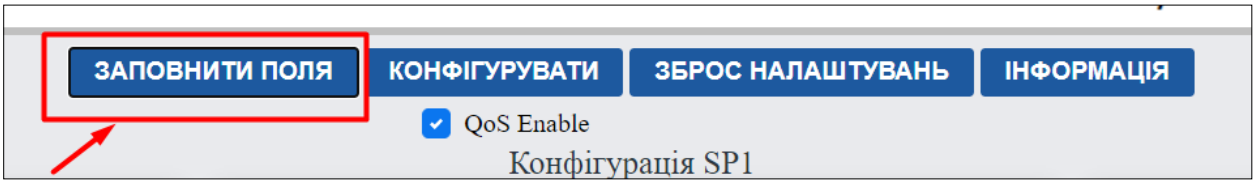

Рисунок 3.6 – Місцезнаходження кнопки «Заповнити поля»

Коли всі поля правильно заповнені, можна перейти до генерації команд для офісів. Треба відмітити, що є можливість не створювати налаштування для якості обслуговування. Для цього треба натиснути на чекбокс «QoS Enable» (за замовчуванням команди активні) (рис. 3.7).

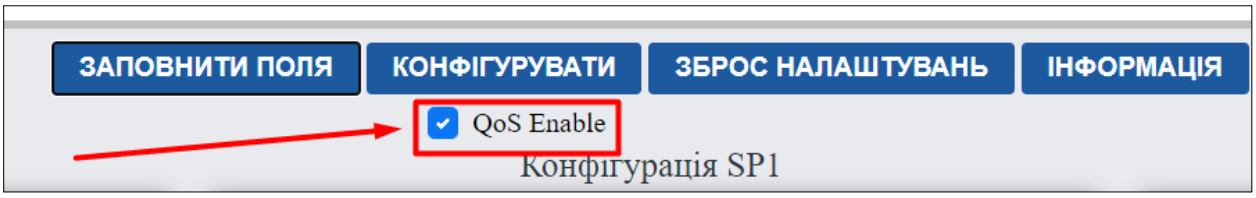

Рисунок 3.7 –Розташування чекбоксу «QoS Enable»

Після цього треба натиснути кнопку «Конфігурувати» та звернути увагу, що у текстових полях з’явились команди для налаштування маршрутизаторів (рис. 3.8).

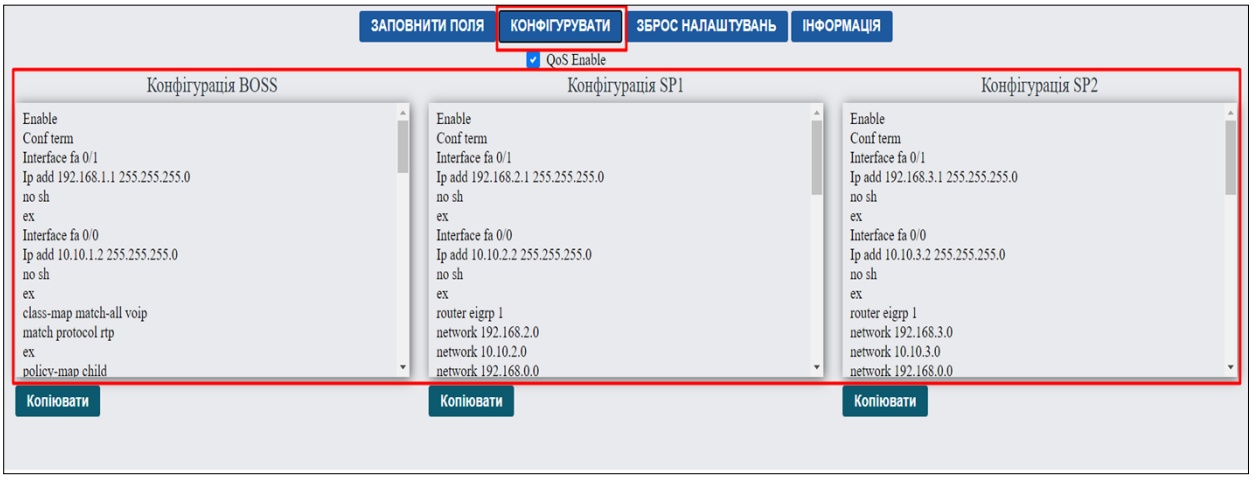

Рисунок 3.8 – Згенерований код для всіх трьох офісів

Конфігурація для роутерів доволі велика, тому для швидкого копіювання додані кнопки «Копіювати» для кожного блоку окремо (рис. 3.9).

Після натискання на кнопку, всі команди копіюються до буферу обміну. Після цього користувач ці команди можна записати у будь-який редактор, на віртуальний роутер чи навіть на реальне обладнання Cisco (якщо воно підтримує технологію DMVPN).

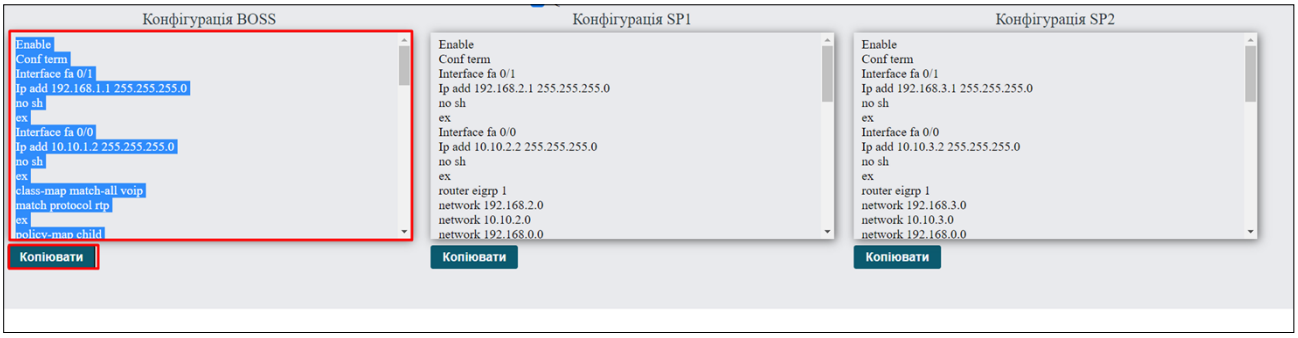

Рисунок 3.9 – Робота кнопки «Копіювати»

Якщо ж не зрозуміле налаштування технології, чи є питання щодо використаних команд налаштування, завжди можна звернутись до довідки натиснуши кнопку «Інформація» (рис. 3.10).

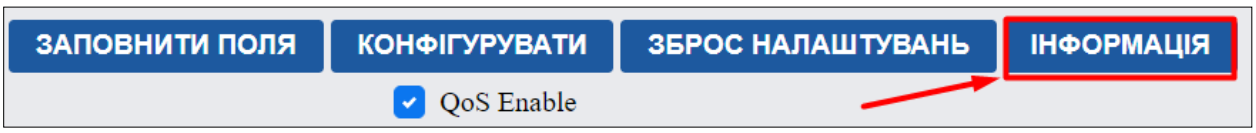

Рисунок 3.10 – Кнопка «Інформація»

Кнопка дозволяє перейти на сторінку з прикладом налаштування DMVPN. На цій сторінці знаходяться команди для налаштування використаної технології, пояснення майже кожної використаної команди, а також показана схема з заданими IP-адресами, для якої вони підходять (рис. 3.11).

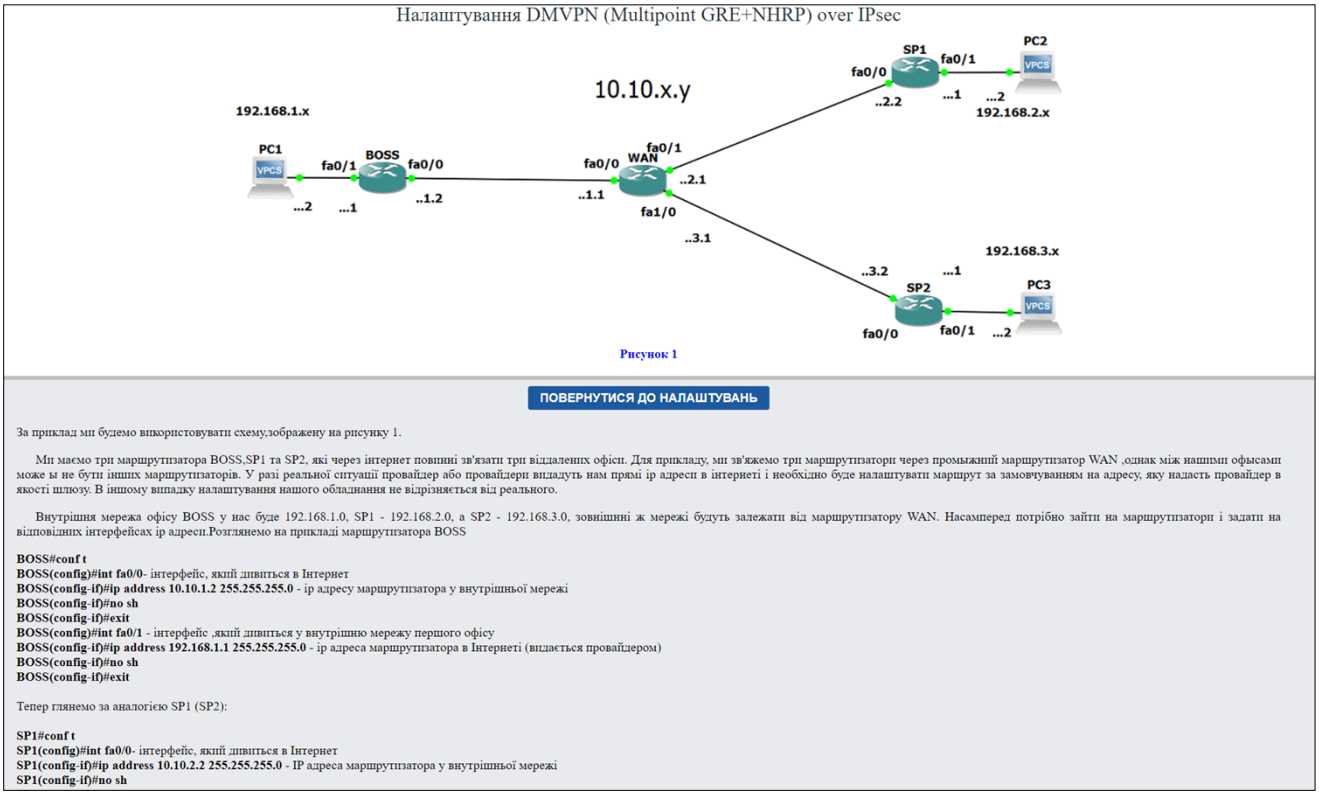

Рисунок 3.11 – Пояснювальна записка з прикладом налаштування технології DMVPN over IPsec

Сторінка дозволяє користувачу розібратися або згадати команди для налаштування DMVPN. Як вже відомо, технологія DMVPN може використовуватися без шифрування та якості обслуговування, тому інформація про ці технології дещо відокремлена одна від одної для кращого розуміння необхідності тієї чи іншої команди.

Для швидкого повернення налаштування схема створено дві кнопки «Повернутись до налаштувань» спочатку та вкінці сторінки (рис. 3.12).

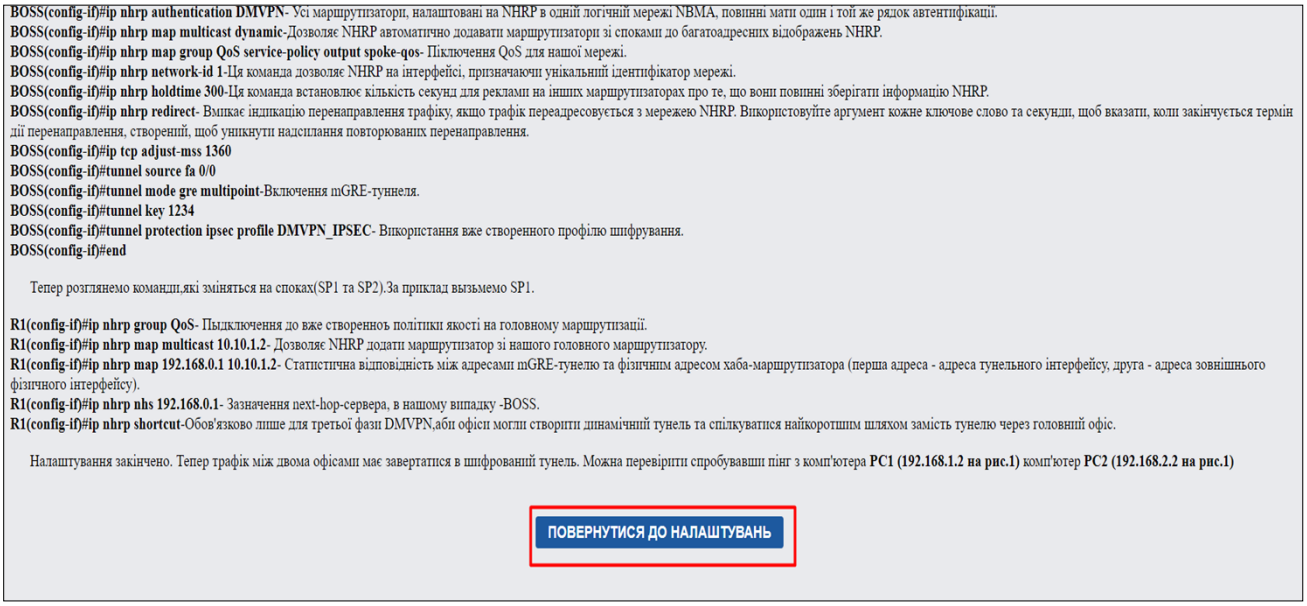

Рисунок 3.12 – кнопка для переходу на сторінку налаштування DMVPN (Multipoint GRE+NHRP) over IPsec

Необхідність кнопки полягає у тому, щоб мати можливість перейти до налаштування схеми, якщо веб-інтерфейс був відкритий із сторінки інформації.

3.2 Тестування створеного графічного інтерфейсу налаштування DMVPN over IPsec

Для перевірки працездатності розробленої веб-орієнтованої програми потрібно її протестувати в эмуляторі GNS3 або краще на реальному обладнанні. Ми обираємо перший варіант для нашого випадку.

Спершу в веб-інтерфейсі вводимо всі IP та маски, необхідні для програми (для пришвидшення перевірки нашого налаштування використаємо значення, які я створив за замовчуванням). (рис. 3.13).

Спершу натискаємо цього «Заповнити поля». Після переконання в тому, що поля заповнені переходимо до отримання результату. Натискаємо кнопку «Конфігурувати». В моєму випадку будемо налаштовувати технологію разом із якістю обслуговуванням (QoS). Після цього бачимо, що код налаштувань DMVPN over IPsec згенерований (рис. 3.14).

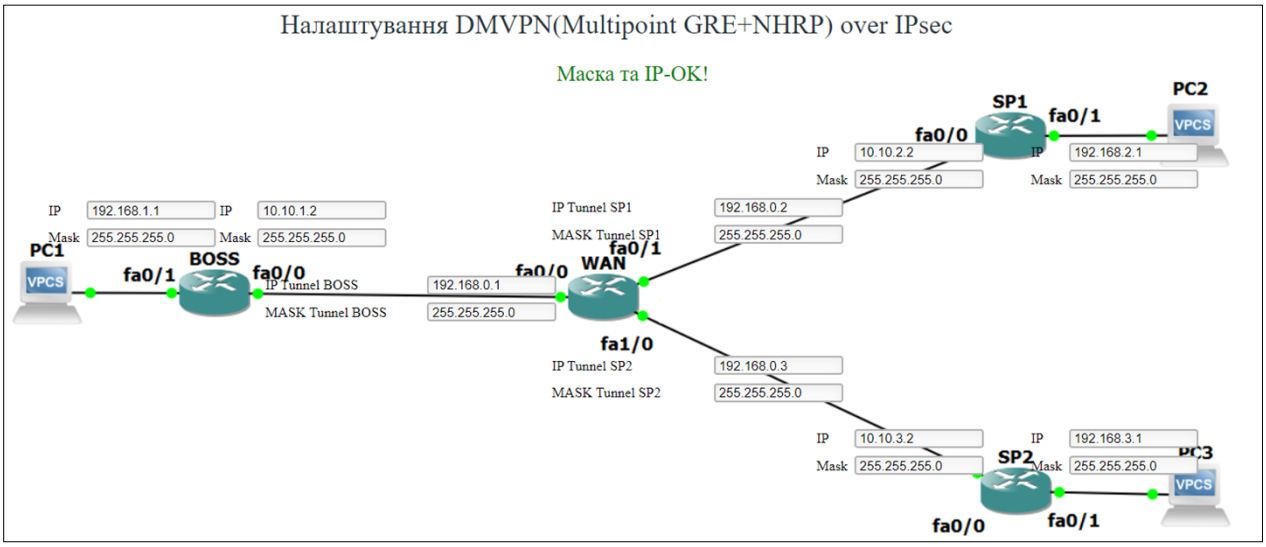

Рисунок 3.13 – Заповнені поля програми

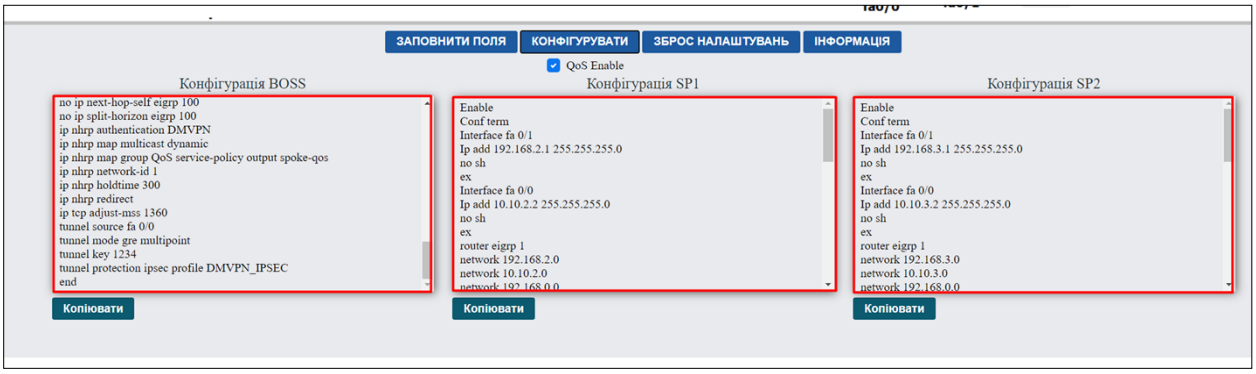

Рисунок 3.14 – Згенерований код налаштувань DMVPN over IPsec у графічному інтерфейсі

Копіюємо налаштування за допомогою кнопки «Копіювати» та вставляємо до вікна налаштувань маршрутизаторів в симуляторі GNS3 на головному офісі (рис. 3.15). За основу були використані роутери серії c7200 компанії Cisco з версією c7200-adventerprisek9-mz.152-4.M7.

Нажаль технологія у повному обсязі підтримується лише роутерами компанії Cisco. Крім того треба розуміти, що старі прошивки роутерів можуть не підтримувати дєякі можливості технології як наприклад третю фазу протоколу DMVPN. ЯК результат, для можливості функціонування потрібна лише підримка набору технології DMVPN over IPsec.

```
BOSS(config)#router eigrp 1
BOSS(config-router)#network 192.168.1.0
BOSS(config-router)#network 10.10.1.0
BOSS(config-router)#network 192.168.0.0
BOSS(config-router)#ex
BOSS(config)#ip route 0.0.0.0 0.0.0.0 fastEthernet0/0
%Default route without gateway, if not a point-to-point interface, may impact perfo
rmance
BOSS(config)#crypto isakmp policy 1
BOSS(config-isakmp)#authentication pre-share
BOSS(config-isakmp)#ex
BOSS(config)#crypto isakmp key PASSWORD address 0.0.0.0 0.0.0.0
BOSS(config)#crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac
BOSS(cfg-crypto-trans)#mode transport
BOSS(cfg-crypto-trans)#ex
BOSS(config)#crypto ipsec profile DMVPN_IPSEC
BOSS(ipsec-profile)#set transform-set AES256-SHA
BOSS(ipsec-profile)#ex
BOSS(config)#interface tunnel 0
BOSS(config-if)#ip address 192.168.0.1 255.255.255.0
BOSS(config-if)#no ip redirects
BOSS(config-if)#ip mtu 1400
BOSS(config-if)#no ip next-hop-self eigrp 100
BOSS(config-if)#no ip split-horizon eigrp 100
BOSS(config-if)#ip nhrp authentication DMVPN
BOSS(config-if)#ip nhrp map multicast dynamic
BOSS(config-if)#ip nhrp map group QoS service-policy output spoke-qos
BOSS(config-if)#ip nhrp network-id 1
BOSS(config-if)#ip nhrp holdtime 300
BOSS(config-if)#ip nhrp redirect
BOSS(config-if)#ip tcp adjust-mss 1360
BOSS(config-if)#tunnel source fa 0/0
BOSS(config-if)#tunnel mode gre multipoint
BOSS(config-if)#tunnel key 1234
BOSS(config-if)#tunnel protection ipsec profile DMVPN_IPSEC
BOSS(config-if)#end
*Nov 20 12:06:39.115: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
BOSS(config-if)#end
```

Рисунок 3.15 – Вікно консолі на роутері BOSS

Тепер перевіримо правильність введене команд в консоль на роутерах. Зробимо це за допомогою команди «sh run» на головному офісі (рис. 3.16).

З рисунка 3.16 ми бачимо, що якість безпеки, шифрування та налаштування інтерфейсів працюють однак сусідів для нашого тунелю ми не бачимо. Для початку треба не забути налаштувати проміжний роутер WAN (якщо такий є). та інші споки. На проміжний роутер налаштувати потрібно лише IP-адреси а споки ми беремо із нашого графічного інтерфейсу.

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key PASSWORD address 0.0.0.0
!
!
crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN_IPSEC
 set transform-set AES256-SHA
!
!
!
!
!
!
!
!
interface Tunnel0
 ip address 192.168.0.1 255.255.255.0
 no ip redirects
 ip mtu 1400
 no ip next-hop-self eigrp 100
 no ip split-horizon eigrp 100
 ip nhrp authentication DMVPN
 ip nhrp map multicast dynamic
 ip nhrp map group QoS service-policy output spoke-qos
 ip nhrp network-id 1
 ip nhrp holdtime 300
 ip nhrp redirect
 ip tcp adjust-mss 1360
 tunnel source FastEthernet0/0
 tunnel mode gre multipoint
 tunnel key 1234
 tunnel protection ipsec profile DMVPN_IPSEC
!
interface FastEthernet0/0
 ip address 10.10.1.2 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 --More--
```

Рисунок 3.16 – Команда «sh run» на роутері BOSS

Тепер налаштуємо (рис. 3.17) та перевіримо (рис. 3.18) споки. За аналогією налаштовуємо другий спок в симуляторі GNS3 за допомогою розробленого програмного забезпечення.

```
%Default route without gateway, if not a point-to-point interface, may impact performance
SP1(config)#crypto isakmp policy 1
SP1(config-isakmp)#authentication pre-share
SP1(config-isakmp)#ex
SP1(config)#crypto isakmp key PASSWORD address 0.0.0.0 0.0.0.0
SP1(config)#crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac
SP1(cfg-crypto-trans)#mode transport
SP1(cfg-crypto-trans)#ex
SP1(config)#crypto ipsec profile DMVPN_IPSEC
SP1(ipsec-profile)#set transform-set AES256-SHA
SP1(ipsec-profile)#ex
SP1(config)#interface tunnel 1
SP1(config-if)#ip add 192.168.0.2 255.255.255.0
SP1(config-if)#no ip redirects
SP1(config-if)#ip mtu 1400
SP1(config-if)#ip nhrp authentication DMVPN
SP1(config-if)#ip nhrp group QoS
SP1(config-if)#ip nhrp map multicast dynamic
SP1(config-if)#ip nhrp map multicast 10.10.1.2
SP1(config-if)#ip nhrp map 192.168.0.1 10.10.1.2
SP1(config-if)#
*Nov 20 12:08:37.895: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Nov 20 12:08:38.191: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to upip nhrp network-id 1
SP1(config-if)#ip nhrp nhs 192.168.0.1
SP1(config-if)#ip nhrp shortcut
SP1(config-if)#ip nhrp redirect
SP1(config-if)#ip tcp adjust-mss 1360
SP1(config-if)#tunnel source fa 0/0
SP1(config-if)#tunnel mode gre multipoint
SP1(config-if)#
*Nov 20 12:08:38.827: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel1, changed state to down
*Nov 20 12:08:38.975: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
*Nov 20 12:08:39.191: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to uptunnel key 1234
SP1(config-if)#tunnel protection ipsec profile DMVPN_IPSEC
SP1(config-if)#end
*Nov 20 12:08:40.111: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel1, changed state to up
SP1(config-if)#end
```

Рисунок 3.17 – Вікно консолі на роутері SP1

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key PASSWORD address 0.0.0.0
!
!
crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN_IPSEC
 set transform-set AES256-SHA
!
!
!
!
!
!
!
!
interface Tunnel1
 ip address 192.168.0.2 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication DMVPN
 ip nhrp group QoS
 ip nhrp map multicast dynamic
 ip nhrp map multicast 10.10.1.2
 ip nhrp map 192.168.0.1 10.10.1.2
 ip nhrp network-id 1
 ip nhrp nhs 192.168.0.1
 ip nhrp shortcut
 ip nhrp redirect
 ip tcp adjust-mss 1360
 tunnel source FastEthernet0/0
 tunnel mode gre multipoint
 tunnel key 1234
 tunnel protection ipsec profile DMVPN_IPSEC
!
interface FastEthernet0/0
 ip address 10.10.2.2 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.2.1 255.255.255.0
```

Рисунок 3.18 – Команда «sh run» на роутері SP1

Із рисунків видно, що піднятий тунель, а також налаштований тип шифрування, якість обслуговування, піднятий тунель, тощо.

Тепер переконаємось, що схема дійсно працює та передає пакети. Тепер будемо використовувати вже відомий сніфер WireShark, який допоможе показати, що мережа дійсно, а й зробили шифровану передачу даних (рис. 3.19). Для цього перехопимо пакет в нашій схемі та подивимось його вміст.

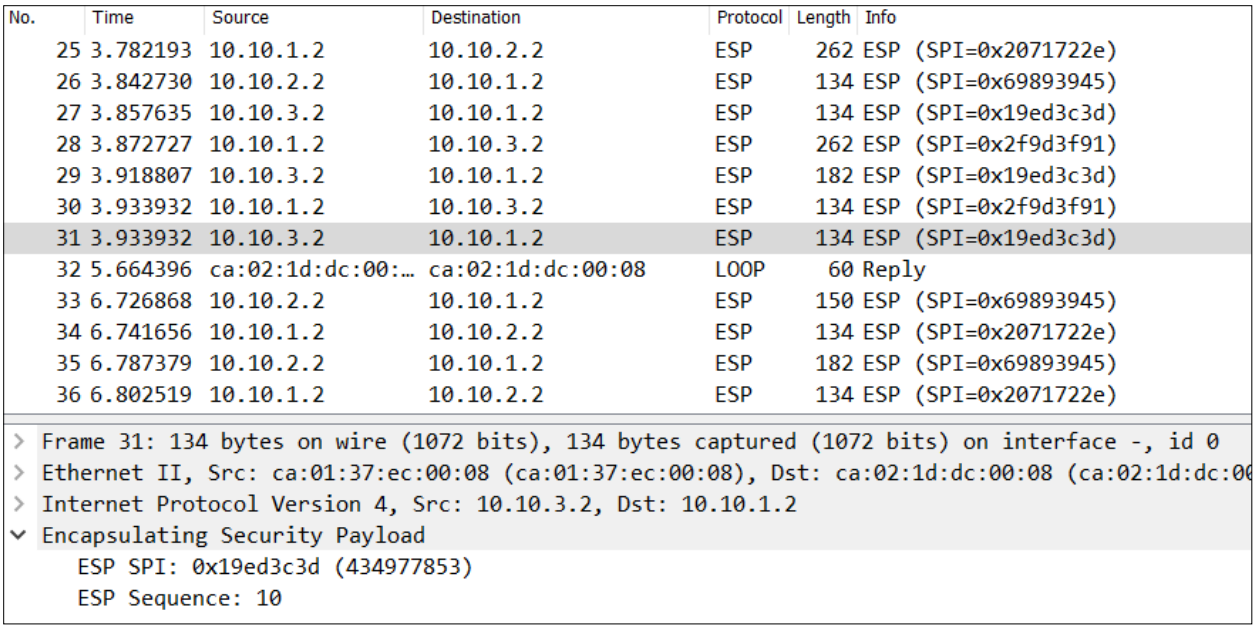

Рисунок 3.19 – Перевірка схеми за допомогою сніфера Wireshark

Як видно із скріншоту, через центральний Роутер WAN передаються зашифровані пакети. З цього можна зробити висновок, що працездатність нашої схеми підтверджена та налаштована навіть із шифруванням.

Для створення схеми від введення IP-адрес та масок на веб-інтерфейсі до перевірки працездатності схеми за допомогою сніферу Wireshark було витрачено приблизно 15 хвилин. Це приблизно втричі швидше, аніж набирати ці команди самому хоча б для одного споку. Отже можна стверджувати, що графічний інтерфейс значно пришвидшує налаштування DMVPN over IPsec а значить він виконує свою головну функцію.

ВИСНОВКИ

Підводячи підсумки кваліфікаційної магістерської роботи можна зробити висновки щодо ефективності технології DMVPN (Multipoint GRE+NHRP) over IPsec. Перш за все треба відмітити, що технологія DMVPN без грошових затрат дозволяє об'єднати величезну кількість офісів, що дозволяє впоратись із зростанням масштабованості мережі краще ніж інші VPN протоколи, зокрема той же чистий GRE тунель. І все ж, виникає проблема із захистом наших пакетів,які можуть перехопити та трафіком, який ніяк не регулюється. В такому випадку нам на допомогу приходять набір протоколів шифрування IPsec та набір політик якості обслуговування відповідно. Звісно, відбудеться невелике зростання «пінгу» та зменшення каналу передачі даних, однак ми отримаємо велику захищену та відмовостійку мережу.

Проте, налаштування цього набору технології є доволі важкою та часозатратною в емуляторах як, наприклад, GNS3 або на живому обладнанні. Основною проблемою стає відсутність графічного інтерфейсу для набору команд та так необхідного пояснення почерговості, що значно ускладнює конфігурування технології, особливо для початківців у цій сфері.

Як результат, для вирішення цієї проблеми я розробив веб-орієнтований графічний інтерфейс програми, яка дозволяє без особливих часових витрат налаштувати VPN технологію DMVPN (Multipoint GRE+NHRP) over IPsec з якістю обслуговування (QoS). Для правильної роботи інтерфейсу необхідно лише ввести IP-адреси та маски інтерфейсів та тунелів. Проект дає змогу швидко скопіювати та перенести отриманий код налаштувань маршрутизатора на інтерфейс віртуально чи реального роутера CISCO, або подивитись інформацію про ту чи іншу команду в довідці.

Отже, результатом роботи став веб-інтерфейс, який дозволяє успішно налаштовувати технологію користувачу, не маючи досвіду роботи з нею. Як результат, зроблений інтерфейс дозволяє пришвидшити роботу на справжньому обладнанні.

СПИСОК ЛІТЕРАТУРИ

1. Andrew G. Mason. Guide to Virtual Private Networks via the Internet between WMO Information System Centres. Indianapolis, Ind., 2016. 60 p.
2. VPN One Click. Types of VPN and types of VPN Protocols [Electronic resource]. 2016. URL: https://www.vpnoneclick.com/types-of-vpn-and-types-of-vpn-protocols/.
3. Н. Олифер С.О. Компьютерные сети принципы, технологии,протоколы. 5th-е издани ed. / ed. Питер. Питер, 2016. 920 p.
4. Frankel S. et al. Guide to IPsec VPNs. 2019. 308 p.
5. Master. Types of VPNs [Electronic resource]. 2019. URL: https://e-tutes.com/lesson12/types-of-vpns/.
6. Master. The Intranet VPN [Electronic resource] // Underst. Virtual. 2019. URL: https://e-tutes.com/lesson12/the-intranet-vpn/.
7. Марат eucariot. DMVPN сети для самих маленьких [Electronic resource]. 2016. URL: https://linkmeup.gitbook.io/sdsm/7.-vpn/5.-dmvpn/0.-teoriya-i-praktika.
8. мерион нетворкс. НАСТРОЙКА DMVPN НА ОБОРУДОВАНИИ CISCO [Electronic resource]. 2018. URL: https://wiki.merionet.ru/seti/10/nastrojka-dmvpn-na-oborudovanii-cisco/.
9. Наташа Самойленко. Настройка DMVPN на маршрутизаторах Cisco [Electronic resource]. 2018. URL: http://xgu.ru/wiki/Настройка_DMVPN_на_маршрутизаторах_Cisco#.D0.A4.D0.B0.D0.B7.D1.8B_DMVPN.
10. Cisco. Настройка резервных интернет-провайдеров в луче DMVPN с помощью функции VRF-Lite [Electronic resource]. 2015. URL: https://www.cisco.com/c/ru_ru/support/docs/security-vpn/dynamic-multi-point-vpn-dmvpn/119022-configure-dmvpn-00.html.
11. Cansever D. NHRP Protocol Applicability Statement [Electronic resource]. URL: https://tools.ietf.org/html/rfc2333.

12. Luciani J. et al. Performance of Encryption Techniques Using Dynamic Virtual Protocol Network Technology // IEEE. 2018.
13. FRR. NHRP — FRR latest documentation [Electronic resource]. 2017.
14. Баумана М. из Н. библиотеки им. Н.Э. IPsec (IP Security) [Electronic resource]. 2017. URL: https://ru.bmstu.wiki/IPsec_(IP_Security).
15. InamdarAmjad, Bartlett G. IKEv2 IPsec Virtual Private Networks: Understanding and Deploying IKEv2, IPsec VPNs, and FlexVPN in Cisco IOS. 2016. 648 p.
16. LanMarket. IPsec [Electronic resource]. 2015. URL: http://xgu.ru/wiki/IPsec.
17. Университет ИТМО. Методическое пособие. IPSec. [Electronic resource]. 2016. URL: https://dfe.karelia.ru/koi/posob/security/index.html#8.
18. Як працює ipsec. Технології використовувані в IPSEC. Входимо у внутрішню мережу [Electronic resource]. 2019. URL: https://beasthackerz.ru/uk/audio/kak-rabotaet-ipsec-tehnologii-ispolzuemye-v-ipsec-vhodim-vo-vnutrennyuyu.html.
19. D-link. Как настроить IPSec VPN failover? [Electronic resource]. 2020. URL: https://www.dlink.ru/ru/faq/85/575.html.
20. Lapukhov P. DMVPN EXPLAINED [Electronic resource]. 2019. URL: https://blog.ine.com/2008/08/02/dmvpn-explained.
21. Демянович В. GNS3 - ГРАФИЧЕСКИЙ СИМУЛЯТОР СЕТИ, МАРШРУТИЗАТОРОВ CISCO [Electronic resource]. 2014. URL: https://elims.org.ua/blog/gns3-graficheskij-simulyator-seti-marshrutizatorov-cisco/.
22. Александр Хан. GNS3 - Графический Сетевой Симулятор [Electronic resource]. 2014. URL: http://www.ciscolab.ru/labs/40-gns3-graficheskiy-setevoy-simulyator.html.
23. Cisco Packet Tracer [Electronic resource]. 2017. URL: http://j0secuerv0.blogspot.com/2017/12/cisco-packet-tracer-packet-tracer-cisco.html.
24. losst.ru. КАК ПОЛЬЗОВАТЬСЯ WIRESHARK ДЛЯ АНАЛИЗА

ТРАФИКА [Electronic resource]. 2016. URL: https://losst.ru/kak-polzovatsya-wireshark-dlya-analiza-trafika.

25. CHRIS HOFFMAN. How to Use Wireshark to Capture, Filter and Inspect Packets. 2017.
26. PETTERS J. How to Use Wireshark: Comprehensive Tutorial + Tips [Electronic resource] // 9/18/2020. 2020. URL: https://www.varonis.com/blog/how-to-use-wireshark/.
27. DMVPN Phase 2 [Electronic resource]. 2017. URL: http://www.amolak.net/dmvpn-phase-2/.
28. С. О. Петрішенко. Петрішенко С. О. Мова JavaScript та її можливості [Electronic resource]. 2014. URL: https://sites.google.com/site/webtehnologiietawebdizajn/mova-javascript-ta-ieie-mozlivosti.
29. Docs M. web. Обзор JavaScript [Electronic resource] // javascript.ru. 2020. URL: https://learn.javascript.ru/intro.

ДОДАТОК

Додаток А

Налаштування на хабі (роутер BOSS)

```
BOSS(config)#int fa0/0
BOSS(config-if)#ip add 10.10.1.2 255.255.255.0
BOSS(config-if)#no sh
BOSS(config-if)#ex
BOSS(config)#int fa0/1
BOSS(config-if)#ip add 192.168.1.1 255.255.255.0
BOSS(config-if)#no sh
BOSS(config-if)#ex
BOSS(config)#router eigrp 100
BOSS(config-router)#network 192.168.0.0 0.0.0.255
BOSS(config-router)#network 192.168.1.0 0.0.0.255
BOSS(config-router)#ex
BOSS(config)#ip route 0.0.0.0 0.0.0.0 10.10.1.1
BOSS(config)#class-map match-all voip
BOSS(config-cmap)#match protocol rtp
BOSS(config-cmap)#ex
BOSS(config)#policy-map child
BOSS(config-pmap)#class voip
BOSS(config-pmap-c)#priority percent 10
BOSS(config-pmap-c)#set dscp ef
BOSS(config-pmap-c)#ex
BOSS(config-pmap)#class class-default
BOSS(config-pmap-c)#queue-limit 1000 packets
BOSS(config-pmap-c)#bandwidth remaining percent 90
BOSS(config-pmap-c)#ex
BOSS(config-pmap)#ex
BOSS(config)#policy
BOSS(config)#policy-map spoke-qos
BOSS(config-pmap)#class class-default
BOSS(config-pmap-c)#shape average 100000000
BOSS(config-pmap-c)#service-policy child
BOSS(config-pmap-c)#ex
BOSS(config-pmap)#ex
BOSS(config)#crypto isakmp policy 1
BOSS(config-isakmp)#authentication pre-share
BOSS(config-isakmp)#ex
BOSS(config)#crypto isakmp key PASSWORD address 0.0.0.0 0.0.0.0
BOSS(config)#crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac
BOSS(cfg-crypto-trans)#mode transport
BOSS(cfg-crypto-trans)#ex
BOSS(config)#crypto ipsec profile DMVPN_IPSEC
BOSS(ipsec-profile)#set transform-set AES256-SHA
BOSS(ipsec-profile)#ex
BOSS(config)#int tunnel 0
BOSS(config-if)# ip address 192.168.0.1 255.255.255.0
BOSS(config-if)# no ip redirects
BOSS(config-if)# ip mtu 1400
BOSS(config-if)# no ip next-hop-self eigrp 100
BOSS(config-if)# ip nhrp authentication DMVPN
BOSS(config-if)# ip nhrp map multicast dynamic
BOSS(config-if)# ip nhrp map group QoS service-policy output spoke-qos
BOSS(config-if)# ip nhrp network-id 1
BOSS(config-if)# ip nhrp holdtime 300
BOSS(config-if)# ip nhrp redirect
```

```
BOSS(config-if)# ip tcp adjust-mss 1360
BOSS(config-if)# no ip split-horizon eigrp 100
BOSS(config-if)# tunnel source FastEthernet0/0
BOSS(config-if)# tunnel mode gre multipoint
BOSS(config-if)# tunnel key 1234
BOSS(config-if)# tunnel protection ipsec profile DMVPN_IPSEC
BOSS(config-if)#ex
```

Налаштування на роутері WAN

```
WAN#conf t
WAN(config)#int fa0/0
WAN(config-if)#ip add 10.10.1.1 255.255.255.0
WAN(config-if)#no sh
WAN(config-if)#ex
WAN(config)#int fa 0/1
WAN(config-if)#ip add 10.10.2.1 255.255.255.0
WAN(config-if)#no sh
WAN(config-if)#ex
WAN(config)#int fa 1/0
WAN(config-if)#ip add 10.10.3.1 255.255.255.0
WAN(config-if)#no sh
WAN(config-if)#
```

Налаштування на споуку (SP1)

```
SP1(config)#int fa0/0
SP1(config-if)#ip add 10.10.2.2 255.255.255.0
SP1(config-if)#no sh
SP1(config-if)#ex
SP1(config)#int fa0/1
SP1(config-if)#ip add 192.168.2.1 255.255.255.0
SP1(config-if)#no sh
SP1(config-if)#ex
SP1(config)#router eigrp 100
SP1(config-router)#network 192.168.0.0 0.0.0.255
SP1(config-router)#network 192.168.2.0 0.0.0.255
SP1(config-router)#ex
SP1(config)#ip route 0.0.0.0 0.0.0.0 10.10.2.1
SP1(config)#crypto isakmp policy 1
SP1(config-isakmp)#authentication pre-share
SP1(config-isakmp)#ex
SP1(config)#crypto isakmp key PASSWORD address 0.0.0.0 0.0.0.0
SP1(config)#crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac
SP1(cfg-crypto-trans)#mode transport
SP1(cfg-crypto-trans)#ex
SP1(config)#crypto ipsec profile DMVPN_IPSEC
SP1(ipsec-profile)#set transform-set AES256-SHA
SP1(ipsec-profile)#ex
SP1(config)#int tunnel 0
SP1(config-if)# ip address 192.168.0.2 255.255.255.0
SP1(config-if)# no ip redirects
SP1(config-if)# ip mtu 1400
SP1(config-if)# ip nhrp authentication DMVPN
SP1(config-if)# ip nhrp group QoS
SP1(config-if)# ip nhrp map multicast dynamic
SP1(config-if)# ip nhrp map multicast 10.10.1.2
SP1(config-if)# ip nhrp map 192.168.0.1 10.10.1.2
SP1(config-if)# ip nhrp network-id 1
SP1(config-if)# ip nhrp nhs 192.168.0.1
SP1(config-if)# ip nhrp shortcut
SP1(config-if)# ip nhrp redirect
SP1(config-if)# ip tcp adjust-mss 1360
SP1(config-if)# tunnel source FastEthernet0/0
SP1(config-if)# tunnel mode gre multipoint
```

```
SP1(config-if)# tunnel key 1234
SP1(config-if)# tunnel protection ipsec profile DMVPN_IPSEC
SP1(config-if)#ex
```

Налаштування на споуку (SP2)

```
SP1(config)#int fa0/0
SP1(config-if)#ip add 10.10.3.2 255.255.255.0
SP1(config-if)#no sh
SP1(config-if)#ex
SP1(config)#int fa0/1
SP1(config-if)#ip add 192.168.3.1 255.255.255.0
SP1(config-if)#no sh
SP1(config-if)#ex
SP1(config)#router eigrp 100
SP1(config-router)#network 192.168.0.0 0.0.0.255
SP1(config-router)#network 192.168.3.0 0.0.0.255
SP1(config-router)#ex
SP1(config)#ip route 0.0.0.0 0.0.0.0 10.10.3.1
SP1(config)#crypto isakmp policy 1
SP1(config-isakmp)#authentication pre-share
SP1(config-isakmp)#ex
SP1(config)#crypto isakmp key PASSWORD address 0.0.0.0 0.0.0.0
SP1(config)#crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac
SP1(cfg-crypto-trans)#mode transport
SP1(cfg-crypto-trans)#ex
SP1(config)#crypto ipsec profile DMVPN_IPSEC
SP1(ipsec-profile)#set transform-set AES256-SHA
SP1(ipsec-profile)#ex
SP1(config)#int tunnel 0
SP1(config-if)# ip address 192.168.0.3 255.255.255.0
SP1(config-if)# no ip redirects
SP1(config-if)# ip mtu 1400
SP1(config-if)# ip nhrp authentication DMVPN
SP1(config-if)# ip nhrp group QoS
SP1(config-if)# ip nhrp map multicast dynamic
SP1(config-if)# ip nhrp map multicast 10.10.1.2
SP1(config-if)# ip nhrp map 192.168.0.1 10.10.1.2
SP1(config-if)# ip nhrp network-id 1
SP1(config-if)# ip nhrp nhs 192.168.0.1
SP1(config-if)# ip nhrp shortcut
SP1(config-if)# ip nhrp redirect
SP1(config-if)# ip tcp adjust-mss 1360
SP1(config-if)# tunnel source FastEthernet0/0
SP1(config-if)# tunnel mode gre multipoint
SP1(config-if)# tunnel key 1234
SP1(config-if)# tunnel protection ipsec profile DMVPN_IPSEC
SP1(config-if)# ex
```

Додаток Б

practika.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js"></script>
<script
src="https://cdn.rawgit.com/zenorocha/clipboard.js/master/dist/clipboard.min.
js"></script>
<script src="js/clipboard.js"></script>
<script src="js/jquery.min.js"></script>
<script src="js/script.js"></script>
            <script type='text/javascript'>
            function validate(value) {
  if  (/^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-
9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-
9]|[01]?[0-9][0-9]?)$/.test(value)) {
        document.getElementById('message').className = "mes1";
        document.getElementById('message').innerHTML = "Маска та IP-OK!";
      var ip_BOSS1 = document.getElementById('IP_BOSS_0_1');
    var ip_BOSS2 = document.getElementById('IP_BOSS_0_0');
      var ip_SP1_1 = document.getElementById('IP_SP1_0_0');
      var ip_SP1_2 = document.getElementById('IP_SP1_0_1');
      var ip_SP2_1 = document.getElementById('IP_SP2_0_0');
      var ip_SP2_2 = document.getElementById('IP_SP2_0_1');
    var ip_tunnel1=document.getElementById('tunnel1');
      var ip_tunnel2=document.getElementById('tunnel2');
      var ip_tunnel3=document.getElementById('tunnel3');
    if    (ip_BOSS1.value    ===    ip_BOSS2.value    ||    ip_BOSS1.value    ===
ip_tunnel1.value ||ip_BOSS2.value === ip_tunnel1.value ){
            document.getElementById('message').className = "mes2";
            document.getElementById('message').innerHTML   ="Однакові   IP   на
роутері BOSS";
      }
        else    if(ip_SP1_1.value    ===ip_SP1_2.value    ||ip_SP1_1.value    ===
ip_tunnel2.value ||ip_SP1_2.value === ip_tunnel2.value){
            document.getElementById('message').className = "mes2";
            document.getElementById('message').innerHTML   ="Однакові   IP   на
роутері SP1";
        }
        else    if(ip_SP2_1.value    ===ip_SP2_2.value    ||ip_SP2_1.value    ===
ip_tunnel3.value ||ip_SP2_2.value === ip_tunnel3.value){
            document.getElementById('message').className = "mes2";
            document.getElementById('message').innerHTML   ="Однакові   IP   на
роутері SP2";
        }
        else  if(ip_BOSS2.value  ===  ip_tunnel2.value  ||  ip_BOSS2.value  ===
ip_tunnel3.value || ip_BOSS2.value === ip_tunnel3.value || ip_BOSS2.value ===
ip_SP1_1.value || ip_BOSS2.value === ip_SP2_1.value){
            document.getElementById('message').className = "mes2";
            document.getElementById('message').innerHTML   ="Однакові   IP   з
інтерфейсом BOSS fa0/0";
        }
        else if(ip_tunnel1.value === ip_tunnel2.value || ip_tunnel1.value ===
ip_tunnel3.value || ip_tunnel1.value === ip_SP1_1.value || ip_tunnel1.value ===
ip_SP2_1.value){
            document.getElementById('message').className = "mes2";
```

```
document.getElementById('message').innerHTML ="Однакові IP з інтерфейсом BOSS
tunnel";
        }
        else if(ip_tunnel2.value === ip_tunnel3.value || ip_tunnel2.value ===
ip_SP1_1.value || ip_tunnel2.value === ip_SP2_1.value){
            document.getElementById('message').className = "mes2";
            document.getElementById('message').innerHTML   ="Однакові   IP   з
інтерфейсом SP1 tunnel";
        }
         else if(ip_tunnel3.value  === ip_SP1_1.value || ip_tunnel3.value ===
ip_SP2_1.value){
         document.getElementById('message').className = "mes2";
         document.getElementById('message').innerHTML    ="Однакові    IP    з
інтерфейсом SP2 tunnel";
        }
        else if(ip_SP1_1.value  === ip_SP2_1.value){
         document.getElementById('message').className = "mes2";
         document.getElementById('message').innerHTML    ="Однакові    IP    на
інтерфейсах SP1 та SP2(fa0/0) ";
        }
        else
        {
        document.getElementById('message').className = "mes1";
        document.getElementById('message').innerHTML = "Маска та IP-OK!";
        }
    return(true);}
  alert("Будь-ласка,перевірте введений IP чи маску");
      document.getElementById('message').className = "mes2";
                  document.getElementById('message').innerHTML      ="ПОМИЛКА!
ПЕРЕВІРТЕ ВВЕДЕНІ IP ТА МАСКИ";
                  var                    Past_in_block_BOSS                    =
document.getElementById('BOSS_RESULT1');
                  Past_in_block_BOSS.innerHTML = " ";
                  var                     Past_in_block_SP1                     =
document.getElementById('block_result_SP1');
              Past_in_block_SP1.innerHTML = " ";
                  var                      block_result_SP2                     =
document.getElementById('block_result_SP2');
              block_result_SP2.innerHTML = " ";
  return (false);
}
function def()
      {       document.getElementById('message').className = "mes1";
                document.getElementById('message').innerHTML  =  "Маска  та  IP-
OK!";}
      function cl()
      {
            document.getElementById('message').className = "mes";
            document.getElementById('message').innerHTML     =     "Будь-ласка,
заповніть всі поля форми!";
            Past_in_block_BOSS.innerHTML = " ";
            Past_in_block_SP1.innerHTML = " ";
            block_result_SP2.innerHTML = " ";
      }
</script>
<link rel="stylesheet" href="css/style.css">
</head>
<body>
<div  class="title_prog">Налаштування  DMVPN(Multipoint  GRE+NHRP)  over  IPsec
</div>
<div class="map">
      <p class='mes' id="message" >Будь-ласка, заповніть всі поля форми!</p>
            <div class="block_router">
```

```
                <div class="block_router_gi_0_0">
                      <div    class="label">IP    Tunnel    BOSS</div><input
id="tunnel1"    placeholder='xxx.xxx.xxx.xxx'  onchange='validate(this.value)'
type="text" name="" > <br>
                      <div    class="label">MASK    Tunnel    BOSS</div><input
id="mask_tunnel1"                                  placeholder='xxx.xxx.xxx.xxx'
onchange='validate(this.value)' type="text" name="" > <br>
                </div>
            </div>
            <div class="block_tunnel_SP1">
                <div class="block_router_gi_0_0">
                      <div    class="label">IP    Tunnel    SP1</div><input
id="tunnel2"    placeholder='xxx.xxx.xxx.xxx'  onchange='validate(this.value)'
type="text" name="" > <br>
                      <div    class="label">MASK    Tunnel    SP1</div><input
id="mask_tunnel2"                                  placeholder='xxx.xxx.xxx.xxx'
onchange='validate(this.value)' type="text" name="" > <br>
                </div>
            </div>
            <div class="block_tunnel_SP2">
                <div class="block_router_gi_0_0">
                      <div    class="label">IP    Tunnel    SP2</div><input
id="tunnel3"    placeholder='xxx.xxx.xxx.xxx'  onchange='validate(this.value)'
type="text" name="" > <br>
                      <div    class="label">MASK    Tunnel    SP2</div><input
id="mask_tunnel3"                                  placeholder='xxx.xxx.xxx.xxx'
onchange='validate(this.value)' type="text" name="" > <br>
                </div>
            </div>
            <div class="GRE_OVER_SP1">
                <div class="int_et_0_0">
                      <div  class="label">IP  </div><input  id="IP_BOSS_0_1"
placeholder='xxx.xxx.xxx.xxx'    onchange='validate(this.value)'    type="text"
name="" > <br>
                      <div class="label">Mask </div><input id="MASK_BOSS_0_1"
placeholder='xxx.xxx.xxx.xxx'       onchange='validate(this.value)'type="text"
name="" >
                </div>
                <div class="int_et_0_1">
                      <div  class="label">IP  </div><input  id="IP_BOSS_0_0"
placeholder='xxx.xxx.xxx.xxx'    onchange='validate(this.value)'    type="text"
name="" > <br>
                      <div class="label">Mask </div><input id="MASK_BOSS_0_0"
placeholder='xxx.xxx.xxx.xxx'    onchange='validate(this.value)'    type="text"
name="" >
                </div>
            </div>
            <div class="block_SP1">
                <div class="int_SP1_0_0">
                      <div  class="label">IP  </div><input  id="IP_SP1_0_0"
placeholder='xxx.xxx.xxx.xxx'     onchange='validate(this.value)'  type="text"
name="" > <br>
                      <div class="label">Mask </div><input id="MASK_SP1_0_0"
placeholder='xxx.xxx.xxx.xxx'       onchange='validate(this.value)'type="text"
name="" >
                </div>
                <div class="int_et_0_1">
                      <div  class="label">IP  </div><input  id="IP_SP1_0_1"
placeholder='xxx.xxx.xxx.xxx'     onchange='validate(this.value)'  type="text"
name="" > <br>
                      <div class="label">Mask </div><input id="MASK_SP1_0_1"
placeholder='xxx.xxx.xxx.xxx'     onchange='validate(this.value)'  type="text"
name="">
```

```
                  </div>
            </div>
            <div class="block_SP2">
                  <div class="int_SP2_0_0">
                        <div   class="label">IP   </div><input   id="IP_SP2_0_0"
placeholder='xxx.xxx.xxx.xxx'     onchange='validate(this.value)'   type="text"
name="" > <br>
                        <div class="label">Mask </div><input id="MASK_SP2_0_0"
placeholder='xxx.xxx.xxx.xxx'       onchange='validate(this.value)'type="text"
name="" >
                  </div>
                  <div class="int_SP2_0_1">
                        <div   class="label">IP   </div><input   id="IP_SP2_0_1"
placeholder='xxx.xxx.xxx.xxx'     onchange='validate(this.value)'   type="text"
name="" > <br>
                        <div class="label">Mask </div><input id="MASK_SP2_0_1"
placeholder='xxx.xxx.xxx.xxx'     onchange='validate(this.value)'   type="text"
name="">
                  </div>
            </div>
            <div class="str0"></div>
            <div class="str1"></div>
      </div>
<div class="bg_block_kontr">
      <div class="block_kontr">
            <div class="block_button_ger">
                  <div   class="block_button_conf"><input   class="button_fill"
type="button" onclick="def()" value="Заповнити поля" ></div>
                  <div   class="block_button_conf"><input   class="button_conf"
type="button"                                                value="Конфігурувати"
onclick="validate(tunnel1.value)&&validate(mask_tunnel1.value)&&validate(tunn
el2.value)&&validate(mask_tunnel2.value)&&validate(tunnel3.value)&&validate(m
ask_tunnel3.value)&&validate(IP_BOSS_0_1.value)&&validate(MASK_BOSS_0_1.value
)&&validate(IP_BOSS_0_0.value)&&validate(MASK_BOSS_0_0.value)&&validate(IP_SP
1_0_1.value)&&validate(MASK_SP1_0_1.value)&&validate(IP_SP1_0_0.value)&&valid
ate(MASK_SP1_0_0.value)&&validate(IP_SP2_0_0.value)&&validate(MASK_SP2_0_0.va
lue)validate(IP_SP2_0_1.value)&&validate(MASK_SP2_0_1.value)" ></div>
                  <div                          class="block_button_conf"><input
class="button_delete_setting"    type="button"    onclick="cl()"    value="Зброс
налаштувань" ></div>
                  <div                          class="block_button_conf"><input
class="button_enter_info"           type="button"           value="Інформація"
onClick='location.href="info.html"' ></div>
            </div>
                  <div class="qos">
            <input  type="checkbox"  checked="checked"  class="custom-checkbox"
id="happy" name="happy" value="yes">
                  <label for="happy">QoS Enable</label>
            </div>
        <div class="left_block_asa">
                  <div class="title_BOSS_RESULT1">Конфігурація BOSS</div>
                  <div id="BOSS_RESULT1" class="BOSS_RESULT1"></div>
                  <input id="button_copy_BOSS" class="button_copy_BOSS" data-
clipboard-target="#BOSS_RESULT1" type="button" value="Копіювати" >
            </div>
            <div class="right_block_asa">
                  <div class="title_block_result_SP1">Конфігурація SP1</div>
                  <div id="block_result_SP1" class="block_result_SP1"></div>
                  <input  id="button_copy_SP1"  class="button_copy_SP1"  data-
clipboard-target="#block_result_SP1" type="button" value="Копіювати" >
            </div>
          <div class="fin_block_SP2">
                  <div class="title_block_result_SP1">Конфігурація SP2</div>
```

```
                <div id="block_result_SP2" class="block_result_SP2"></div>
                <input  id="button_copy_SP2"  class="button_copy_SP2"  data-
clipboard-target="#block_result_SP2" type="button" value="Копіювати" >
            </div>
      </div>

</div>
</body>
</html>
```

info.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js"></script>
<script
src="https://cdn.rawgit.com/zenorocha/clipboard.js/master/dist/clipboard.min.
js"></script>
<script src="js/clipboard.js"></script>
<script src="js/jquery.min.js"></script>
<script src="js/script.js"></script>
<link rel="stylesheet" href="css/style.css">
</head>
<body>
<div   class="title_prog">Налаштування   DMVPN   (Multipoint   GRE+NHRP)   over
IPsec</div>
      <div class="map1"></div>
      <font                          style="color:blue"><center><strong>Рисунок
1</strong></center></font>
      <br>
      <div class="bg_block_kontr">
      <div class="block_kontr">
      <center><div    class="block_button_conf"><input    class="button_fill"
type="button"   onClick='location.href="DMVPN   (Multipoint   GRE+NHRP)   over
IPsec.html"'  value="Повернутися до налаштувань" ></div></center><br>
За приклад ми будемо використовувати схему,зображену на рисунку 1.<br>
<p>Ми маємо три маршрутизатора BOSS,SP1 та SP2, які через інтернет повинні
зв'язати три віддалених офіси. Для прикладу, ми  зв'яжемо три маршрутизатори
через промыжний маршрутизатор WAN ,однак між нашими офысами може ы не бути інших
маршрутизаторів. У разі реальної ситуації провайдер або провайдери видадуть нам
прямі ip адреси в інтернеті і необхідно буде налаштувати маршрут за
замовчуванням на адресу, яку надасть провайдер в якості шлюзу. В іншому випадку
налаштування нашого обладнання не відрізняється від реального.</p>
<p>Внутрішня мережа офісу BOSS у нас буде 192.168.1.0, SP1 - 192.168.2.0, а SP2
- 192.168.3.0, зовнішнні ж мережі будуть залежати від маршрутизатору WAN.
Насамперед потрібно зайти на маршрутизатори і задати на відповідних інтерфейсах
ip адреси.Розглянемо на прикладі маршрутизатора BOSS</p>
<b>BOSS#conf t <br>
BOSS(config)#int fa0/0</b>- інтерфейс, який дивиться в Інтернет<br>
<b>BOSS(config-if)#ip  address  10.10.1.2  255.255.255.0</b>  -  ip  адресу
маршрутизатора у внутрішньої мережі <br>
<b>BOSS(config-if)#no sh<br>
BOSS(config-if)#exit<br>
BOSS(config)#int fa0/1</b> - інтерфейс ,який дивиться у внутрішню мережу першого
офісу <br>
<b>BOSS(config-if)#ip  address  192.168.1.1  255.255.255.0</b>  -  ip  адреса
маршрутизатора в Інтернеті (видається провайдером)<br>
<b>BOSS(config-if)#no sh<br>
BOSS(config-if)#exit</b><br>
```


Тепер глянемо за аналогією SP1 (SP2):

SP1#conf t
SP1(config)#int fa0/0- інтерфейс, який дивиться в Інтернет
SP1(config-if)#ip address 10.10.2.2 255.255.255.0 - IP адреса маршрутизатора у внутрішньої мережі
SP1(config-if)#no sh
SP1(config-if)#exit
SP1(config)#int fa0/1 - інтерфейс ,який дивиться у внутрішню мережу другого офісу
SP1(config-if)#ip address 192.168.2.1 255.255.255.0 - IP адреса маршрутизатора в Інтернеті (видається провайдером)
SP1(config-if)#no sh
SP1(config-if)#exit

SP2#conf t
SP2(config)#int fa0/0- інтерфейс, який дивиться в Інтернет
SP2(config-if)#ip address 10.10.3.2 255.255.255.0 - IP адреса маршрутизатора у внутрішньої мережі
SP2(config-if)#no sh
SP2(config-if)#exit
SP2(config)#int fa0/1 - інтерфейс ,який дивиться у внутрішню мережу другого офісу
SP2(config-if)#ip address 192.168.3.1 255.255.255.0 - IP адреса маршрутизатора в Інтернеті (видається провайдером)
SP2(config-if)#no sh
SP2(config-if)#exit

 Тепер можна налаштувати наш тунель в офісах, однак спочатку налаштуємо технології шифрування а також якість обслуговування (QoS).IPSec – протокол захисту мережевого трафіку на IP-рівні тунелі в чистому вигляді цього не можуть,а суть якості обслуговування інших методів полягає у пріоритетному наданні ресурсів мережі трафіку чутливих протоколів за рахунок протоколів, яким не потрібна висока якість обслуговування. Для їх налаштування використовуємо такі команди:
На роутері BOSS:
BOSS(config)#class-map match-all voip- Використовується для завдання класу трафіка та критеріїв цього класу.
BOSS(config-cmap)#match protocol rtp- Налаштування розпізнавання мережевих додатків (NBAR) відповідно до трафіку протоколу реального часу (RTP).
BOSS(config-cmap)#ex
BOSS(config)#policy-map child- Створює або змінює дочірню політику. Входить у режим конфігурації карти політики.
BOSS(config-pmap)#class voip- Призначає вказаний вами клас трафіку на карті політики. Входить у режим конфігурації класу map-policy.
BOSS(config-pmap-c)#set dscp ef- Команда match у модульному інтерфейсі QoS для відповідності значенням DSCP.
BOSS(config-pmap-c)#ex
BOSS(config-pmap)#class class-default- Налаштовуємо клас для якості обслуговування.
BOSS(config-pmap-c)#set dscp default
BOSS(config-pmap-c)#queue-limit 1000 packets- Налаштовуємо поточну глибину цієї черги та налаштований ліміт черги.
BOSS(config-pmap-c)#bandwidth remaining percent 90- Обсяг інформації, який спроможний пропустити канал за певний тимчасовий проміжок.(В нашому випадку 90 відсодків).
BOSS(config-pmap-c)#ex
BOSS(config-pmap)#ex
BOSS(config)#policy-map spoke-qos
BOSS(config-pmap)#class class-default

BOSS(config-pmap-c)#shape average 100000000- Максимум того, що може отримати клієнт.
BOSS(config-pmap-c)#service-policy child- Присвоюваємо дочірну політику.
BOSS(config-pmap-c)#ex
BOSS(config-pmap)#ex
 Після цього перейдемо до налаштувань набору протоколів IPsec (зокрема можна спочатку "підняти" IPsec, а вже потім - DMVPN).Налаштування буде однакове на всіх офісах, однак краще налаштовувати шифрування, використовуючи сертифікати, що значно збільшує надійність мережі. Наведемо приклад конфігурації і опишемо докладніше:

BOSS(config)#crypto isakmp policy 1- Створити політику isakmp.
BOSS(config-isakmp)#authentication pre-share- Налаштування політики з аутентифікацією по pre-shared key.
BOSS(config-isakmp)#ex
BOSS(config)#crypto isakmp key PASSWORD address 0.0.0.0 0.0.0.0- Необхідно вказувати шаблонний адреса (wildcard address).
BOSS(config)#crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac- Налаштування тунельних інтерфейсів dynamic crypto map.
BOSS(cfg-crypto-trans)#mode transport- Переведення IPsec перевести в транспортний режим.
BOSS(cfg-crypto-trans)#ex
BOSS(config)#crypto ipsec profile DMVPN_IPSEC- Створення IPSec-профіль.
BOSS(ipsec-profile)#set transform-set AES256-SHA
BOSS(ipsec-profile)#ex

Так як офіси розділені додатково проміжними маршрутизаторами, треба використати протоколи динамічної маршрутиризації, наприклад eigrp. Що ж піднімемо протокол динамічної маршрутизації.
BOSS(config)#router eigrp 100
BOSS(config-router)#network 192.168.1.0
BOSS(config-router)#network 192.168.0.0

По аналогії, робимо і для інших офісів.
Що ж тепер можна налаштувати тунель з піднятою технологією DMVPN. Почнемо з маршрутизатора BOSS:

BOSS(config)#interface tunnel 0- Піднімаємо інтерфейсу тунелю
BOSS(config-if)#ip address 192.168.0.1 255.255.255.0
BOSS(config-if)#no ip redirects- Відключає переадресацію протоколу керуючих повідомлень
BOSS(config-if)#ip mtu 1400- В цьому випадку ip mtu буде забезпечувати простір для mGRE + IPSec, в разі ж більш низьких значеннях MTU по шляху, IP MTU буде підкориговано динамічно. Значення 1 400 рекомендовано, тому що воно покриває більшість можливих комбінацій mGRE + IPSec.
BOSS(config-if)#no ip next-hop-self eigrp 100- EIGRP, за замовчуванням, встановлює значення IP наступного переходу як власне для маршрутів, які він рекламує, навіть коли рекламує ці маршрути на тому самому інтерфейсі, з якого маршрутизатор їх дізнався. Щоб змінити це значення за замовчуванням, ви повинні використовувати команду no ip next-hop-self eigrp, щоб доручити EIGRP використовувати отримане значення наступного переходу при рекламі цих маршрутів.
BOSS(config-if)#no ip split-horizon eigrp 100- Вимкнення розділеного горизонту для EIGRP аби дати змогу спілкуватися нашим всім офісам спілкуватися один з одним.
BOSS(config-if)#ip nhrp authentication DMVPN- Усі маршрутизатори, налаштовані на NHRP в одній логічній мережі NBMA, повинні мати один і той же рядок автентифікації.
BOSS(config-if)#ip nhrp map multicast dynamic-Дозволяє NHRP автоматично додавати маршрутизатори зі споками до багатоадресних відображень NHRP.

```
<b>BOSS(config-if)#ip nhrp map group QoS service-policy output spoke-qos</b>-Піключення QoS для нашої мережі.<br>
<b>BOSS(config-if)#ip nhrp network-id 1</b>-Ця команда дозволяє NHRP на інтерфейсі, призначаючи унікальний ідентифікатор мережі.<br>
<b>BOSS(config-if)#ip nhrp holdtime 300</b>-Ця команда встановлює кількість секунд для реклами на інших маршрутизаторах про те, що вони повинні зберігати інформацію NHRP.<br>
<b>BOSS(config-if)#ip nhrp redirect</b>- Вмикає індикацію перенаправлення трафіку, якщо трафік переадресовується з мережею NHRP. Використовуйте аргумент кожне ключове слово та секунди, щоб вказати, коли закінчується термін дії перенаправлення, створений, щоб уникнути надсилання повторюваних перенаправлення.<br>
<b>BOSS(config-if)#ip tcp adjust-mss 1360</b><br>
<b>BOSS(config-if)#tunnel source fa 0/0</b><br>
<b>BOSS(config-if)#tunnel mode gre multipoint</b>-Включення mGRE-туннеля.<br>
<b>BOSS(config-if)#tunnel key 1234</b><br>
<b>BOSS(config-if)#tunnel protection ipsec profile DMVPN_IPSEC</b>-Використання вже створенного профілю шифрування.<br>
<b>BOSS(config-if)#end</b><br>

<p>Тепер розглянемо команди,які зміняться на споках(SP1 та  SP2).За приклад вызьмемо SP1.</p>
<b>R1(config-if)#ip nhrp group QoS</b>- Пыдключення до вже створенноъ політики якості на головному маршрутизації.<br>
<b>R1(config-if)#ip nhrp map multicast 10.10.1.2</b>- Дозволяє NHRP додати маршрутизатор зі нашого головного маршрутизатору.<br>
<b>R1(config-if)#ip nhrp map 192.168.0.1 10.10.1.2</b>- Статистична відповідність між адресами mGRE-тунелю та фізичним адресом хаба-маршрутизатора (перша адреса - адреса тунельного інтерфейсу, друга - адреса зовнішнього фізичного інтерфейсу).<br>
<b>R1(config-if)#ip nhrp nhs 192.168.0.1</b>- Зазначення next-hop-сервера, в нашому випадку -BOSS.<br>
<b>R1(config-if)#ip nhrp shortcut</b>-Обов'язково лише для третьої фази DMVPN,аби офіси могли створити динамічний тунель та спілкуватися найкоротшим шляхом замість тунелю через головний офіс.<br>
<p>Налаштування закінчено. Тепер трафік між двома офісами має завертатися в шифрований тунель. Можна перевірити спробувавши пінг з комп'ютера <b>PC1 (192.168.1.2 на рис.1) </b>комп'ютер <b>PC2 (192.168.2.2 на рис.1)</b></p>
<br>
<center><div class="block_button_conf"><input class="button_fill" type="button" onClick='location.href="DMVPN (Multipoint GRE+NHRP) over IPsec.html"' value="Повернутися до налаштувань" ></div></center></div>
</div>
<hr>
</body>
</html>
```

Script.js

```
$(document).ready(function(){
new Clipboard('.button_copy_BOSS');
new Clipboard('.button_copy_SP1');
new Clipboard('.button_copy_SP2');
$(".button_conf").click(function(){
var IP_BOSS_0_1 = document.getElementById("IP_BOSS_0_1").value;
var MASK_BOSS_0_1 = document.getElementById("MASK_BOSS_0_1").value;
var IP_BOSS_0_0 = document.getElementById("IP_BOSS_0_0").value;
var MASK_BOSS_0_0 = document.getElementById("MASK_BOSS_0_0").value;
var IP_SP1_0_0 = document.getElementById("IP_SP1_0_0").value;
var MASK_SP1_0_0 = document.getElementById("MASK_SP1_0_0").value;
var IP_SP1_0_1 = document.getElementById("IP_SP1_0_1").value;
var MASK_SP1_0_1 = document.getElementById("MASK_SP1_0_1").value;
```

```
var IP_SP2_0_0 = document.getElementById("IP_SP2_0_0").value;
var MASK_SP2_0_0 = document.getElementById("MASK_SP2_0_0").value;
var IP_SP2_0_1 = document.getElementById("IP_SP2_0_1").value;
var MASK_SP2_0_1 = document.getElementById("MASK_SP2_0_1").value;
var tr = document.getElementById('message').className.value;
var tunnel1 = document.getElementById("tunnel1").value;
var mask_tunnel1 = document.getElementById("mask_tunnel1").value;
var tunnel2 = document.getElementById("tunnel2").value;
var mask_tunnel2 = document.getElementById("mask_tunnel2").value;
var tunnel3 = document.getElementById("tunnel3").value;
var mask_tunnel3 = document.getElementById("mask_tunnel3").value;
if(IP_BOSS_0_1 == "" ||
      MASK_BOSS_0_1 == "" ||
      IP_BOSS_0_0 == "" ||
      MASK_BOSS_0_0 == "" ||
      IP_SP1_0_0 == "" ||
      MASK_SP1_0_0 == "" ||
      IP_SP1_0_1 == "" ||
      MASK_SP1_0_1 == "" ||
    IP_SP2_0_0 == "" ||
    MASK_SP2_0_0 == "" ||
    IP_SP2_0_1 == "" ||
    MASK_SP2_0_1 == "" ||
      mask_tunnel1 == "" ||
      mask_tunnel2 == "" ||
    mask_tunnel3 == "" ||
      tunnel1 == "" ||
    tunnel2 == "" ||
      tunnel3 == ""){
                  var                    Past_in_block_BOSS                    =
document.getElementById('BOSS_RESULT1');
                  Past_in_block_BOSS.innerHTML = " ";
                  var                     Past_in_block_SP1                    =
document.getElementById('block_result_SP1');
              Past_in_block_SP1.innerHTML = " ";
              var                       Past_in_block_SP2                      =
document.getElementById('block_result_SP2');
              Past_in_block_SP2.innerHTML = " ";
alert("Ви заповнили не всі поля");
}

else{
/*Узнаем сеть по маске например 10.10.3.1 10.10.3.0 reverse для ospf/eigrp*/
if(MASK_BOSS_0_1 == "255.0.0.0"){
      var a = IP_BOSS_0_1.split('.');
      var network1 = a[0] + ".0.0.0";
}
if(MASK_BOSS_0_1 == "255.255.0.0"){
      var a = IP_BOSS_0_1.split('.');
      var network1 = a[0] + "." + a[1] + ".0.0";
}
if(MASK_BOSS_0_1 == "255.255.255.0"){
      var a = IP_BOSS_0_1.split('.');
      var network1 = a[0] + "." + a[1] + "." + a[2] +".0";
}
if(MASK_BOSS_0_0 == "255.0.0.0"){
      var a = IP_BOSS_0_0.split('.');
      var network10 = a[0] + ".0.0.0";
}
if(MASK_BOSS_0_0 == "255.255.0.0"){
      var a = IP_BOSS_0_0.split('.');
      var network10 = a[0] + "." + a[1] + ".0.0";
}
```

```
if(MASK_BOSS_0_0 == "255.255.255.0"){
      var a = IP_BOSS_0_0.split('.');
      var network10 = a[0] + "." + a[1] + "." + a[2] +".0";
}

if(MASK_SP1_0_1 == "255.0.0.0"){
      var a = IP_SP1_0_1.split('.');
      var network2 = a[0] + ".0.0.0";
}
if(MASK_SP1_0_1 == "255.255.0.0"){
      var a = IP_SP1_0_1.split('.');
      var network2 = a[0] + "." + a[1] + ".0.0";
}
if(MASK_SP1_0_1 == "255.255.255.0"){
      var a = IP_SP1_0_1.split('.');
      var network2 = a[0] + "." + a[1] + "." + a[2] +".0";
}
if(MASK_SP1_0_0 == "255.0.0.0"){
      var a = IP_SP1_0_0.split('.');
      var network20 = a[0] + ".0.0.0";
}
if(MASK_SP1_0_0 == "255.255.0.0"){
      var a = IP_SP1_0_0.split('.');
      var network20 = a[0] + "." + a[1] + ".0.0";
}
if(MASK_SP1_0_0 == "255.255.255.0"){
      var a = IP_SP1_0_0.split('.');
      var network20 = a[0] + "." + a[1] + "." + a[2] +".0";
}
if(MASK_SP2_0_0 == "255.0.0.0"){
      var a = IP_SP2_0_0.split('.');
      var network_sp2 = a[0] + ".0.0.0";
}
if(MASK_SP2_0_0 == "255.255.0.0"){
      var a = IP_SP2_0_0.split('.');
      var network_sp2 = a[0] + "." + a[1] + ".0.0";
}
if(MASK_SP2_0_0  == "255.255.255.0"){
      var a = IP_SP2_0_0.split('.');
      var network_sp2 = a[0] + "." + a[1] + "." + a[2] +".0";
}
if(MASK_SP2_0_1  == "255.0.0.0"){
      var a = IP_SP2_0_1.split('.');
      var network_sp20 = a[0] + ".0.0.0";
}
if(MASK_SP2_0_1  == "255.255.0.0"){
      var a = IP_SP2_0_1.split('.');
      var network_sp20 = a[0] + "." + a[1] + ".0.0";
}
if(MASK_SP2_0_1  == "255.255.255.0"){
      var a = IP_SP2_0_1.split('.');
      var network_sp20 = a[0] + "." + a[1] + "." + a[2] +".0";
}
if(mask_tunnel1 == "255.0.0.0"){
      var a = tunnel1.split('.');
      var network3 = a[0] + ".0.0.0";
}
if(mask_tunnel1 == "255.255.0.0"){
      var a = tunnel1.split('.');
      var network3 = a[0] + "." + a[1] + ".0.0";
}
if(mask_tunnel1 == "255.255.255.0"){
      var a = tunnel1.split('.');
```

```
      var network3 = a[0] + "." + a[1] + "." + a[2] +".0";
}
if(mask_tunnel2 == "255.0.0.0"){
      var a = tunnel2.split('.');
      var network4 = a[0] + ".0.0.0";
}
if(mask_tunnel2 == "255.255.0.0"){
      var a = tunnel2.split('.');
      var network4 = a[0] + "." + a[1] + ".0.0";
}
if(mask_tunnel2 == "255.255.255.0"){
      var a = tunnel2.split('.');
      var network4 = a[0] + "." + a[1] + "." + a[2] +".0";
}
if(mask_tunnel1 == "255.0.0.0"){
      var a = mask_tunnel1.split('.');
      var reverse1 = "0"+a[2] + "." + a[1] + "." + a[0] ;
}
if(mask_tunnel1 == "255.255.0.0"){
      var a = mask_tunnel1.split('.');
      var reverse1 =  + "0.0."+a[1] + "." + a[0];
}
if(mask_tunnel1 == "255.255.255.0"){
      var a = mask_tunnel1.split('.');
      var reverse1 = "0.0.0." +a[0];
}

if(mask_tunnel2 == "255.0.0.0"){
      var a = mask_tunnel2.split('.');
      var reverse2 = "0"+a[2] + "." + a[1] + "." + a[0] ;
}
if(mask_tunnel2 == "255.255.0.0"){
      var a = mask_tunnel2.split('.');
      var reverse2 =  + "0.0."+a[1] + "." + a[0];
}
if(mask_tunnel2 == "255.255.255.0"){
      var a = mask_tunnel2.split('.');
      var reverse2 = "0.0.0." +a[0];
}
if(mask_tunnel3 == "255.0.0.0"){
      var a = tunnel3.split('.');
      var network_tunnel_sp2 = a[0] + ".0.0.0";
}
if(mask_tunnel3 == "255.255.0.0"){
      var a = tunnel3.split('.');
      var network_tunnel_sp2 = a[0] + "." + a[1] + ".0.0";
}
if(mask_tunnel3 == "255.255.255.0"){
      var a = tunnel3.split('.');
      var network_tunnel_sp2 = a[0] + "." + a[1] + "." + a[2] +".0";
}
      if(mask_tunnel3 == "255.0.0.0"){
      var a = mask_tunnel3.split('.');
      var reverse_tunnel_sp2 = "0"+a[2] + "." + a[1] + "." + a[0] ;
}
if(mask_tunnel3 == "255.255.0.0"){
      var a = mask_tunnel3.split('.');
      var reverse_tunnel_sp2 =  + "0.0."+a[1] + "." + a[0];
}
if(mask_tunnel3 == "255.255.255.0"){
      var a = mask_tunnel3.split('.');
      var reverse_tunnel_sp2 = "0.0.0." +a[0];
}
```

```
if(tr == "mes2")
{alert("ПЕРЕВІРИТИ IP ЧИ МАСКУ!");

var Past_in_block_BOSS = document.getElementById('BOSS_RESULT1');
Past_in_block_BOSS.innerHTML = " ";
var Past_in_block_SP1 = document.getElementById('block_result_SP1');
Past_in_block_SP1.innerHTML = " ";
var Past_in_block_SP2 = document.getElementById('block_result_SP2');
Past_in_block_SP2.innerHTML = " ";
}
var Past_in_block_BOSS = document.getElementById('BOSS_RESULT1');
Past_in_block_BOSS.innerHTML = "Enable" +
"<br>Conf term" +
"<br>Interface fa 0/1" +
"<br>Ip add " +IP_BOSS_0_1+" "+MASK_BOSS_0_0 +
"<br>no sh "+
"<br>ex" +
"<br>Interface fa 0/0" +
"<br>Ip add " +IP_BOSS_0_0+" " +MASK_BOSS_0_0 +
"<br>no sh" +
"<br>ex";
if (document.getElementById('happy').checked)
  {$('#BOSS_RESULT1').append(
"<br>class-map match-all voip" +
"<br>match protocol rtp" +
"<br>ex" +
"<br>policy-map child" +
"<br>class voip" +
"<br>priority percent 10" +
"<br>set dscp ef" +
"<br>ex" +
"<br>class class-default" +
"<br>set dscp default" +
"<br>queue-limit 1000 packets" +
"<br>bandwidth remaining percent 90" +
"<br>ex" +
"<br>ex" +
"<br>policy-map spoke-qos" +
"<br>class class-default" +
"<br>shape average 100000000" +
"<br>service-policy child" +
"<br>ex" +
"<br>ex");
}
$('#BOSS_RESULT1').append("<br>router eigrp 100" +
"<br>network " + network1  +
"<br>network " + network3  +
"<br>ex" +
"<br>ip route 0.0.0.0 0.0.0.0 fastEthernet0/0"+
"<br>crypto isakmp policy 1" +
"<br>authentication pre-share" +
"<br>ex" +
"<br>crypto isakmp key PASSWORD address 0.0.0.0 0.0.0.0" +
"<br>crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac" +
"<br> mode transport" +
"<br>ex" +
"<br>crypto ipsec profile DMVPN_IPSEC" +
"<br>set transform-set AES256-SHA" +
"<br>ex" +
"<br>interface tunnel 0" +
"<br>ip address " +tunnel1+" "+ mask_tunnel1 +
"<br>no ip redirects" +
"<br>ip mtu 1400" +
```

```
"<br>no ip next-hop-self eigrp 100" +
"<br>no ip split-horizon eigrp 100" +
"<br>ip nhrp authentication DMVPN" +
"<br>ip nhrp map multicast dynamic");
if (document.getElementById('happy').checked)
      {$('#BOSS_RESULT1').append("<br>ip  nhrp  map  group  QoS  service-policy
output spoke-qos");
}
$('#BOSS_RESULT1').append("<br>ip nhrp network-id 1" +
"<br>ip nhrp holdtime 300" +
"<br>ip nhrp redirect" +
"<br>ip tcp adjust-mss 1360" +
"<br>tunnel source fa 0/0" +
"<br>tunnel mode gre multipoint" +
"<br>tunnel key 1234" +
"<br>tunnel protection ipsec profile DMVPN_IPSEC" +
"<br>end");
var Past_in_block_SP1 = document.getElementById('block_result_SP1');
Past_in_block_SP1.innerHTML = "Enable" +
"<br>Conf term" +
"<br>Interface fa 0/1" +
"<br>Ip add " +IP_SP1_0_1+" "+MASK_SP1_0_0 +
"<br>no sh "+
"<br>ex" +
"<br>Interface fa 0/0" +
"<br>Ip add " +IP_SP1_0_0+" " +MASK_SP1_0_0 +
"<br>no sh"+
"<br>ex" +
"<br>router eigrp 100" +
"<br>network " + network2  +
"<br>network " + network4  +
"<br>ex" +
"<br>ip route 0.0.0.0 0.0.0.0 fastEthernet0/0"+
"<br>crypto isakmp policy 1" +
"<br>authentication pre-share" +
"<br>ex" +
"<br>crypto isakmp key PASSWORD address 0.0.0.0 0.0.0.0" +
"<br>crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac" +
"<br> mode transport" +
"<br>ex" +
"<br>crypto ipsec profile DMVPN_IPSEC" +
"<br>set transform-set AES256-SHA" +
"<br>ex" +
"<br>interface tunnel 1" +
"<br>ip add " +tunnel2+" "+ mask_tunnel2 +
"<br>no ip redirects" +
"<br>ip mtu 1400" +
"<br>ip nhrp authentication DMVPN";
if (document.getElementById('happy').checked){
      $('#block_result_SP1').append("<br>ip nhrp group QoS");
}
$('#block_result_SP1').append("<br>ip nhrp map multicast dynamic" +
"<br> ip nhrp map multicast " + IP_BOSS_0_0 +
"<br>ip nhrp map " + tunnel1 +" "+ IP_BOSS_0_0 +
"<br>ip nhrp network-id 1" +
"<br>ip nhrp nhs " + tunnel1 +
"<br>ip nhrp shortcut" +
"<br>ip nhrp redirect" +
"<br>ip tcp adjust-mss 1360" +
"<br>tunnel source fa 0/0" +
"<br>tunnel mode gre multipoint" +
"<br>tunnel key 1234"+
"<br>tunnel protection ipsec profile DMVPN_IPSEC"+
```

```
"<br>end");
var Past_in_block_SP2 = document.getElementById('block_result_SP2');
Past_in_block_SP2.innerHTML = "Enable" +
"<br>Conf term" +
"<br>Interface fa 0/1" +
"<br>Ip add " +IP_SP2_0_1+" "+MASK_SP2_0_1 +
"<br>no sh "+
"<br>ex" +
"<br>Interface fa 0/0" +
"<br>Ip add " +IP_SP2_0_0+" " +MASK_SP2_0_0 +
"<br>no sh"+
"<br>ex" +
"<br>router eigrp 100" +
"<br>network " + network_sp20  +
"<br>network " + network_tunnel_sp2  +
"<br>ex" +
"<br>ip route 0.0.0.0 0.0.0.0 fastEthernet0/0"+
"<br>crypto isakmp policy 1" +
"<br>authentication pre-share" +
"<br>ex" +
"<br>crypto isakmp key PASSWORD address 0.0.0.0 0.0.0.0" +
"<br>crypto ipsec transform-set AES256-SHA esp-aes 256 esp-sha-hmac" +
"<br> mode transport" +
"<br>ex" +
"<br>crypto ipsec profile DMVPN_IPSEC" +
"<br>set transform-set AES256-SHA" +
"<br>ex" +
"<br>interface tunnel 1" +
"<br>ip add " +tunnel3+" "+ mask_tunnel3 +
"<br>no ip redirects" +
"<br>ip mtu 1400" +
"<br>ip nhrp authentication DMVPN";
if (document.getElementById('happy').checked){
      $('#block_result_SP2').append("<br>ip nhrp group QoS");
}
$('#block_result_SP2').append("<br>ip nhrp map multicast dynamic" +
"<br> ip nhrp map multicast " + IP_BOSS_0_0 +
"<br>ip nhrp map " + tunnel1 +" "+ IP_BOSS_0_0 +
"<br>ip nhrp network-id 1" +
"<br>ip nhrp nhs " + tunnel1 +
"<br>ip nhrp shortcut" +
"<br>ip nhrp redirect" +
"<br>ip tcp adjust-mss 1360" +
"<br>tunnel source fa 0/0" +
"<br>tunnel mode gre multipoint" +
"<br>tunnel key 1234"+
"<br>tunnel protection ipsec profile DMVPN_IPSEC"+
"<br>end");
}});
/*Ввод в поля дефолтных адресов*/
$(".button_fill").click(function(){
      document.getElementById("IP_BOSS_0_1").value = "192.168.1.1";
      document.getElementById("MASK_BOSS_0_1").value = "255.255.255.0";
      document.getElementById("IP_BOSS_0_0").value = "10.10.1.2";
      document.getElementById("MASK_BOSS_0_0").value = "255.255.255.0";

      document.getElementById("IP_SP1_0_0").value = "10.10.2.2";
      document.getElementById("MASK_SP1_0_0").value = "255.255.255.0";
      document.getElementById("IP_SP1_0_1").value = "192.168.2.1";
      document.getElementById("MASK_SP1_0_1").value = "255.255.255.0";

      document.getElementById("IP_SP2_0_0").value = "10.10.3.2";
      document.getElementById("MASK_SP2_0_0").value = "255.255.255.0";
```

```
        document.getElementById("IP_SP2_0_1").value = "192.168.3.1";
        document.getElementById("MASK_SP2_0_1").value = "255.255.255.0";

        document.getElementById("tunnel1").value = "192.168.0.1";
        document.getElementById("mask_tunnel1").value = "255.255.255.0";
        document.getElementById("tunnel2").value = "192.168.0.2";
        document.getElementById("mask_tunnel2").value = "255.255.255.0";
        document.getElementById("tunnel3").value = "192.168.0.3";
        document.getElementById("mask_tunnel3").value = "255.255.255.0";
});

/*Очистить все заданые настройки*/
$(".button_delete_setting").click(function(){

        document.getElementById("IP_BOSS_0_1").value = "";
        document.getElementById("MASK_BOSS_0_1").value = "";
        document.getElementById("IP_BOSS_0_0").value = "";
        document.getElementById("MASK_BOSS_0_0").value = "";

        document.getElementById("IP_SP1_0_0").value = "";
        document.getElementById("MASK_SP1_0_0").value = "";
        document.getElementById("IP_SP1_0_1").value = "";
        document.getElementById("MASK_SP1_0_1").value = "";

        document.getElementById("IP_SP2_0_0").value = "";
        document.getElementById("MASK_SP2_0_0").value = "";
        document.getElementById("IP_SP2_0_1").value = "";
        document.getElementById("MASK_SP2_0_1").value = "";

        document.getElementById("tunnel1").value = "";
        document.getElementById("mask_tunnel1").value = "";
        document.getElementById("tunnel2").value = "";
        document.getElementById("mask_tunnel2").value = "";
        document.getElementById("tunnel3").value = "";
        document.getElementById("mask_tunnel3").value = "";

        var Past_in_block_BOSS = document.getElementById('BOSS_RESULT1');
        Past_in_block_BOSS.innerHTML = "";
        var Past_in_block_SP1 = document.getElementById('block_result_SP1');
        Past_in_block_SP1.innerHTML = "";
        var Past_in_block_SP2  = document.getElementById('block_result_SP2');
        Past_in_block_SP2.innerHTML = "";
});
});
```

Style.css

```
.mes{
    text-align: center;
    font-size: 20px;
    color: blue;

}
.mes1{
    text-align: center;
    font-size: 20px;
    color: green;

}
```

```
.mes2{
    text-align: center;
    font-size: 20px;
    color: red;

}
.title_prog {
    text-align: center;
    font-size: 25px;
    color: #2A3541;
}
.message{
      text-align: center;
      font-size: 25px;
      color:blue;
}
.BOSS_RESULT1 {
    width: 500px;
    height: 250px;
    /* border: 1px solid red; */
    overflow: auto;
    padding: 5px 0 5px 10px;
    box-shadow: 2px 2px 12px 0px rgba(50, 50, 50, 0.75);
}
.block_result_SP1 {
    width: 500px;
    height: 250px;
    /* border: 1px solid red; */
    overflow: auto;
    padding: 5px 0 5px 10px;
    box-shadow: 2px 2px 12px 0px rgba(50, 50, 50, 0.75);
}
.block_result_SP2 {
    width: 500px;
    height: 250px;
    /* border: 1px solid red; */
    overflow: auto;
    padding: 5px 0 5px 10px;
    box-shadow: 2px 2px 12px 0px rgba(50, 50, 50, 0.75);
}
.label{
      width: 35px;
      margin: 5px 0;
      display: inline-block;
    font-size: 13px;
}
.block_router input,
.block_tunnel_SP1 input,
.block_tunnel_SP2 input,
.block_user input,
.block_SP2 input,
.GRE_OVER_SP1 input,
.block_SP1 input{
      width: 110px;
    padding: 0 3px;
    background-color: #fcfcfc;
    border: 2px solid #b3b2b2;
    color: #000;
    font-size: 12px;
    border-radius: 3px;
    box-shadow: inset 1px 3px 10px 0 #eeeeef;
}
```

```
.GRE_OVER_SP1 input{
      margin: 3px 0;
}
.map{
    background: url("../image/1.jpg") no-repeat center;
      width: 1160px;
      height: 450px;
      position: relative;
      margin: 0 auto;
}
.map1{
    background: url("../image/example.jpg") no-repeat center;
      width: 1500px;
      height: 400px;
      position: relative;
      margin: 0 auto;
}
.GRE_OVER_SP1 {
    position: absolute;
    top: 28%;
    left: 1%;
      z-index: 10;
}
.block_SP1 {
    position: absolute;
    top: 16%;
    left: 66%;
      z-index: 10;
}
.block_SP2 {
    position: absolute;
    top: 75%;
    left: 66%;
      z-index: 10;
}
.block_button_conf{
      margin: 0 auto;
      position: relative;
      display: inline-block;
}
.right_block_asa {
    display: inline-block;
      float: center;
    margin: 0 0 0 25px ;
}
.fin_block_SP2 {
    display: inline-block;
    float: right;
    margin: 0 0 0 0 ;
}
.left_block_asa {
    display: inline-block;
      float: left;
    margin: 0 0 0 0;
}
.GRE_OVER_SP1 .int_et_0_0 {
    display: inline-block;
    margin: 0 0 0 41px;
}
.GRE_OVER_SP1 .int_et_0_1 {
    display: inline-block;
}
```

```
.block_SP1 .int_SP1_0_0 {
    display: inline-block;
}
.block_SP1 .int_et_0_1 {
    display: inline-block;
    margin: 0 0 0 40px;
}
.block_SP2 .int_SP2_0_0 {
    display: inline-block;
}
.block_SP2 .int_SP2_0_1 {
    display: inline-block;
    margin: 0 0 0 40px;
}
.bg_block_kontr {
    background: #E9EAED;
    border-top: 5px solid #C0C0C0;
    padding: 0 0 50px 0;
}
input.button_copy_SP1 {
    margin: 10px 0 0 0;
}
input.button_copy_BOSS {
    margin: 10px 0 0 0;
}
input.button_copy_SP2 {
    margin: 10px 0 0 0;
}

input.button_delete_setting,
input.button_fill,
input.button_conf,
input.button_enter_info,
input.button_enter_default {
    padding: 6px 15px;
    background-color: #1D599F;
    border: 0;
    border-radius: 3px;
    color: #fff;
    font-size: 15.4px;
    text-transform: uppercase;
    font-weight: 600;
    margin: 8px 0 0 0;
    cursor: pointer;
}
input.button_delete_setting:hover,
input.button_fill:hover,
input.button_conf:hover,
input.button_enter_info:hover,
input.button_enter_default:hover {
    background: #105DB7;
}
.title_BOSS_RESULT1,
.title_block_result_SP1,
.title_block_result_SP2 {
    text-align: center;
    font-size: 20px;
    margin: 0 0 6px 0;
    color: #2A3541;
}
input#button_copy_SP2,
input#button_copy_BOSS,
input#button_copy_SP1 {
```

```
background: #0B5A6F;
    padding: 6px 15px;
    border: 0;
    border-radius: 3px;
    color: #fff;
    font-size: 15.4px;
    font-weight: 600;
    margin: 8px 0 0 0;
    cursor: pointer;
}
input#button_copy_SP2:hover,
input#button_copy_BOSS:hover,
input#button_copy_SP1:hover {
    background: #047390;
}
.block_user input:focus,
.block_router input:focus,
.GRE_OVER_SP1 input:focus,
.block_tunnel_SP1 input:focus,
.block_tunnel_SP2 input:focus,
.block_SP2 input:focus,
.block_SP1 input:focus {
    border-color: #74d36b;
}
.block_user {
    position: absolute;
    top: 13px;
    left: 622px;
}
.block_user .label{
    width: 67px;
}
.block_router {
    position: absolute;
    top: 195px;
    left: 254px;
      z-index: 10;
}

.block_router .label{
    width: 150px;

}
.block_tunnel_SP1 {
    position: absolute;
    top: 123px;
    left: 520px;
      z-index: 10;
}

.block_tunnel_SP1 .label{
    width: 150px;

}
.block_tunnel_SP2 {
    position: absolute;
    top: 270px;
    left: 520px;
      z-index: 10;
}

.block_tunnel_SP2 .label{

width: 150px;
```

```
}
.block_kontr {
    width: 1580px;
    margin: 0 auto;
    background: #E9EAED;
}
.qos{
      margin: -20px 0 0 -155px;
      text-align: center;
}
.custom-checkbox {
  position: absolute;
  z-index: -1;
  opacity: 0;
}
.custom-checkbox+label {
  display: inline-flex;
  align-items: center;
  user-select: none;
}
.custom-checkbox+label::before {
  content: '';
  display: inline-block;
  width: 1em;
  height: 1em;
  flex-shrink: 0;
  flex-grow: 0;
  border: 1px solid #adb5bd;
  border-radius: 0.25em;
  margin-right: 0.5em;
  background-repeat: no-repeat;
  background-position: center center;
  background-size: 50% 50%;
}
.custom-checkbox:checked+label::before {
  border-color: #0b76ef;
  background-color: #0b76ef;
  background-image:                                  url("data:image/svg+xml,%3csvg
xmlns='http://www.w3.org/2000/svg'  viewBox='0  0  8  8'%3e%3cpath  fill='%23fff'
d='M6.564.75l-3.59      3.612-1.538-1.55L0      4.26      2.974      7.25      8
2.193z'/%3e%3c/svg%3e");
}
p
{
      text-indent: 1.5em;
    text-align: justify;
}
.block_button_ger {
    text-align: center;
    height: 65px;
}
```