

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему:

**«Інженеринг трафіку в мережах MPLS з
використанням протоколу RSVP-TE»**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Великодний Д.В.

Студента групи ІН–64–8

Лимар В.В.

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2020 р.

**ЗАВДАННЯ
до випускної роботи**

Студента четвертого курсу, групи ІН-64-8 спеціальності “Інформатика” денної форми навчання Лимаря Владислава Володимировича.

Тема: «Інженеринг трафіку в мережах MPLS з використанням протоколу RSVP-TE»

Затверджена наказом по СумДУ

№ _____ от _____ 2020 р.

Зміст пояснювальної записки: 1) аналітичний огляд існуючих рішень; 2) вибір інструментарію для рішення задачі; 3) виконання проектного завдання.

Дата видачі завдання « _____ » _____ 2020 р.

Керівник випускної роботи _____ Великодний Д.В.

Завдання прийняв до виконання _____ Лимар В.В.

РЕФЕРАТ

Записка: 47 сторінок, 19 рисунків, 8 джерел.

Мета роботи — розробка веб-інтерфейсу, через який можна буде налаштувати конфігурацію роутерів і протокол RSVP-TE

Об'єкт дослідження — інженеринг трафіку в мережах MPLS з використанням протоколу RSVP-TE

Методи дослідження — побудова схеми мережі за допомогою емулятора GNS3.

Результати — розроблений веб-інтерфейс, який дозволяє автоматично створювати конфігурації для інтерфейсів роутера, налаштувати протокол RSVP-TE. Система дозволяє зручно перенести створений код налаштувань роутеру. Система представлена у вигляді веб-інтерфейсу, для створення використовувався JavaScript.

MPLS, ПРОТОКОЛ RSVP-TE, ВЕБ-ІНТЕРФЕЙС,
ЕМУЛЯТОР GNS3, МОВА JAVASCRIPT, МЕРЕЖА

ЗМІСТ

ВСТУП.....	5
1 АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	6
1.1 Основи технології MPLS	6
1.2 Комутація по міткам	10
1.3 Протокол RSVP-TE	14
1.4 Постановка задачі.....	15
2 МОДЕЛЮВАННЯ ТЕХНОЛОГІЇ MPLS ЗА ДОПОМОГОЮ ПРОТОКОЛУ RSVP-TE В СИМУЛЯТОРІ GNS3.....	16
2.1 Симулятор GNS3. Його особливості.....	16
2.2 Реалізація проекту в GNS3	17
3 РОЗРОБКА ГРАФІЧНОГО ВЕБ-ІНТЕРФЕЙСУ ДЛЯ НАЛАШТУВАННЯ MPLS ЗА ДОПОМОГОЮ ПРОТОКОЛУ RSVP-TE.....	22
3.1 Засоби розробки веб-інтерфейсу	22
3.2 Розробка графічного інтерфейсу для налаштування MPLS за допомогою протоколу RSVP-TE.....	23
3.1 Тестування готового веб-інтерфейсу в емуляторі GNS3	25
ВИСНОВКИ	29
СПИСОК ЛІТЕРАТУРИ	30
ДОДАТКИ.....	31

ВСТУП

Розвиток галузі телекомунікацій з кожним днем все більше і більше набирає обертів. У зв'язку з тим, що постійно запроваджуються нові технології та підвищуються послуги в напрямку IP-технологій, спостерігається збільшення користувачів та використаного трафіку. Тож операторами зв'язку було вирішено впроваджувати новітні послуги, через те що, йде перехід до мультисервісності. Стикаючись з цим, з'являються обмеження з боку функціонування мереж та виникає необхідність виконання потреб обслуговування – QoS (Quality of Service). Для того щоб гарантувати ефективне розподілення ресурсів мережі, з метою надання всіх вимог QoS у систему взаємозв'язку можливо ввести систему управління трафіками, які в свою чергу повинні слідкувати за характерними рисами різних класів трафіку. Після появи додатків QoS в мережі Інтернет, було запроваджено створення нових мережевих технологій. Однією з яких є технологія MPLS.

MPLS є дуже популярною та сучасною технологією серед всіх телекомунікаційних мереж. Вона забезпечує взаємозв'язок в багатьох галузях життєдіяльності людини, з нею можна зіткнутися як в маленьких організаціях так і в великих підприємствах. Технологія MPLS застосовується в магістральних мережах майже всіх операторів, все більше проникаючи в зонові мережі. Перед усім, її головною метою є збільшення пропускної здатності глобальних і корпоративних мереж. Ця мережа являється однією з провідних технологій, з якою можливий вихід на новий рівень з метою створення нових мультисервісних мереж наступного покоління за допомогою NGN. Вони забезпечать нові передачі трафіку через єдину телекомунікаційну інфраструктуру.

1 АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Основи технології MPLS

MPLS (Multiprotocol Label Switching) — технологія для передавання даних, за допомогою якої вона буде емулювати різні властивості мереж з комутацією каналів через мережі з комутацією пакетів [5].

MPLS називаний протоколом другого рівня, через те що він працює на рівні, які знаходяться між канальним (2 рівень) і мережевим (3 рівень) моделями OSI. Він створює таблиці маршрутизації та за мітками розсилає їх по мультикасту, повідомляючи своїх сусідів-пристроїв [5].

Завдяки MPLS-технології пристрої швидко сканують мітку та передають дані вже не за IP-маршрутом, а за маршрутом по міткам.

Взагалі є три основні напрямки застосування MPLS технології :

- трафік-менеджмент;
- класове забезпечення обслуговування (Class Of Service);
- організація віртуальних мереж приватного типу (Virtual Private Network).

Корпоративні мережі часто будують на основі MPLS технологій. З цього виходить, що MPLS-мережа відноситься до мереж зв'язку в якій є комутація пакетів, основна побудова її еталонної моделі, яка має сім рівнів взаємодії відкритих систем.

(EMBOC / ISO).

Розташування технології MPLS в 7-рівневій моделі ISO показано на рисунку 1.1.

Площина управління MPLS		
Прикладний рівень	OSPF, RIP, BGP, IS-IS	Маршрутизація
Транспортний рівень	LDP, CR-LDP, RSVP, RSVP-TE	Сигналізація
Мережевий рівень	IP-пакели	
Площина пересилання даних MPLS		
Рівень ланки даних	Комірки, кадри, конверти, фрейми, ATM, FR Ethernet	
Фізичний рівень	Біти	

Рисунок 1.1 – Площини MPLS

На третьому рівні мережевої моделі OSI здійснюється маршрутизація. Принцип роботи полягає: коли, наприклад, два користувача розташовуються у різних підмережах, то постає питання в способі отримання оптимального маршруту.

Безперешкодна передача даних здійснюється по другому рівню, застосовуючи фізичний канал. Принцип його роботи полягає в тому, щоб забезпечити вірну фізичну адресацію, сформувати топологічну таблицю мережі, управляти потоками мережі, які попереджають створення помилок, здійснюється управління всієї доставки трафіку.

На першому рівні мережевої моделі OSI, визначаються усі функціональні характеристики мережі, контролюються усі каналні зв'язки в кінцевих пристроях.

MPLS протокол може здійснити інкапсуляцію заголовків протоколів та виконувати транспортування всіх даних вищих рівнів моделі.

Міжнародною організацією IETF було виділено декілька головних елементів MPLS-технології:

- для визначення до якого відносяться класу пересилки FEC (Forward Error Correction) – буде використовуватися мітка;

- мітка значення якої мають лише два сусідні маршрутизатори, що мають одну ділянку та пересилають данні один-до-одного. Коли вони переходять за допомогою міток, то пересилання здійснюються між сусідами, а згодом коли виконається вхід йде заміна на інші мітки (вихідні), яка буде дійсна на черговій ділянці. Мітки здійснюють передачу від любого типу пакета, однак їх місце знаходження залежний від протокольного типу каналного рівня.

- пакети, які передають мережею MPLS, за часту, вони вміщують в себе не одну, а більше міток. Стеком міток називають один загальний пакет, який має здібність пересилати мітки в мережі. Він, тобто стек, реалізує одну з важливих ролей у роботі технології MPLS. Він утворює дерево комутованого по міткам тракту та зберігає його, щоб скористуватися кожною міткою за призначення (LSP-тунелю).

Forward Error Correction таблиця, використовується для демонстрації групи пакетів з ідентичними параметрами до передачі. Взагалі в заголовку IP-пакету знаходиться багато інформації, ніж необхідної для визначення наступного LSR.

Щоб здійснити вибір слідуючого LSR необхідно зробити на маршрутизаторах деякі процеси:

- створити класифікацію пакета за FEC;
- зробити асоціювання для кожного наступного маршрутизаційного кроку.

Пріоритетна підтримка пакетів та IP-адресів визначають неабияку роль при систематизації Forward Error Correction. Якісна підтримка для всіх з FEC повинена постійно залишатися на найвищому рівні, через те що кожен обов'язково здійснює обробку особисто.

Передаючись по LSP-тунелю пакети належать до однієї FEC, та всі LSR-маршрутизатори в цьому тунелі присуджують особисті мітки. Створення LSP – тунель містить в собі LSP-тракт. Вважається, що тунель являється коротшим за тривалістю від LSP-тракту. У центрі деякого LSP-тракту буде можливість для створення кількох LSP-тунелів в яких прийом та передача може відрізнятись,

проте в кожному з тунелів може знаходитися LSP-тунелі іншого рівня. З цього виходить що так і буде демонструвати MPLS свою створену ієрархію.

Визначають наступні варіанти створення LSP-тунелів:

- hop-by-hop, при якому сам маршрутизатор вирішує загальні шляхи проходження пакету;
- при допомозі дійсної маршрутизації, коли маршрутизатор (LSR) вказує маршрут на один із вищих рівней у тракті.

Отже беручи до уваги клас деяких маршрутизаторів, то для нього наявний деякий LSP-тунель з різними місцями для входження і виходу. Зустрічається випадок коли для всіх links LSP є вихідним, то LER створює дерево, в якому корінь - вхідний маршрутизатор.

Маршрутом з'єднання усіх шляхів та на всіх ділянках, які долучаються до передачі пакету, будуть змінювати мітку на всіх послідуючих вузлах з вхідної на вихідну, то може бути представлений повний тракт LSP. LSP можливо з'єднувати в двох випадках, або якщо будуть розкриті потоки даних, або ж перед тим як ці дані будуть передаватися. Принцип пакетної інкапсуляції базується на створенні трактів в центрі інших трактів.

Для роботи MPLS використовують протоколи маршрутизації розповсюдження (LDP), простий необмежений протокол (без підтримки трафіку), протокол резервування ресурсів з проектуванням трафіку (RSVP-TE). На практиці ж зазвичай використовують протокол розповсюдження влучний (LDP), однак протокол RSVP-TE необхідний для функцій організації трафіку і в складних мережах практично не обійтися без цих двох протоколів з налаштуванням LDP для тунелювання всередині протоколу RSVP.

Передача і управління трафіків відбувається за рахунок технології Traffic Engineering, яка здійснює передачу трафіку по каналах по найбільш оптимальним маршрутом, але з деякими обмеженнями завдяки технології CSPF (Constrained Shortest Path First), яка вибирає шляхи не тільки користуючись критерієм, заснованому на його оптимальній довжині маршруту, але ще і враховує

завантаження маршрутів. Використовувані протоколи RSVP-TE дозволяють резервувати смуги пропускання в мережі.

Технологія MPLS також має захист від збоїв ґрунтуючись попередньому розрахунку шляхів резервного копіювання для потенційних збоїв каналу або вузла. При наявності збою в мережі автоматично відбувається розрахунок найкращого шляху, але при наявності одного збою розрахунок необхідної шляху починає відбуватися ще до виявлення збою. Шляхи резервного копіювання попередньо запрограмовані в FIB маршрутизатора в очікуванні активації, яка може статися в мілісекундах після виявлення збою.

Переваги MPLS-мережі в порівнянні з транспортної IP-мережею:

- технологія MPLS підтримує показники якості обслуговування QoS, надаючи різні класи обслуговування. IP-мережі не надають такої можливості;
- технологія MPLS дозволяє збалансувати навантаження в мережі, здійснюючи перерозподіл потоків (інжиніринг трафіку). Це підвищує показники QoS за рахунок оптимізації використання смуги пропускання на недостатньо завантажених маршрутах. Протоколи IP-мережі такої можливості не передбачають;
- при використанні технології MPLS провайдери служб можуть створювати так звані віртуальні приватні мережі VPN (Virtual Private Network). VPN-мережі містять географічно віддалені один від одного вузли, які можуть безпечно пов'язувати їх з спільно використовуваної магістралі. На відміну від IP-мереж технологія MPLS дозволяє створювати VPN-мережі без необхідності використовувати дороге шифрування.

1.2 Комутація по міткам

У мережах MPLS пакетам даних присвоюються так звані «мітки» (Label). Вони використовуються в якості своєрідного адреси вузла, якій призначений

конкретний пакет даних. При цьому зміст самого пакета не має значення, і дані передаються відповідно до міткою.

Основна перевага міток полягає в тому, що вони комутуються швидше, ніж маршрутизуються пакети в стандартних IP-мережах. Під терміном «стандартні IP-мережі» в даній статті маються на увазі мережі, в яких для ідентифікації пристрою, підключеного до локальної мережі або Інтернету, використовується унікальний ідентифікатор - IP-адреса (Internet Protocol Address). У стандартному варіанті Internet IETF (далі Інтернет) кінцевий користувач визначається по IP-адресою пункту призначення (Destination Internet Protocol address). Кожен пристрій в IP-мережах має можливість вибирати самостійно напрямок доставки пакета відповідно до IP-адресою пункту призначення і таблицею маршрутизації.

Технологія MPLS дає можливість створення наскрізного віртуального каналу з будь-яким протоколом передачі даних, незалежного від середовища передачі. Застосовуючи різні мітки, можна створювати кілька різних віртуальних мереж на базі одних і тих же вузлів. Крім того, мережі MPLS можна масштабувати.

Слід зазначити, що прототипи сучасних міток MPLS використовувалися і в більш ранніх технологіях, таких, наприклад, як FR (Frame Relay) і ATM (Asynchronous Transfer Mode). В технології FR використовуються мітки із змінним розміром, а в ATM мітки мають фіксований обсяг. Однак сама мітка трансформується в процесі передачі [3]. Аналогічний механізм задіяний в MPLS, де мітка змінюється після кожного шлюзу. В приклад розглянемо включення мітки в IP-пакет заголовка Ethernet (рис. 1.2).

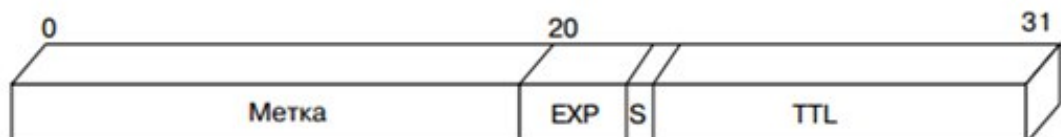


Рисунок 1.2 – Формат міток в MPLS

Чотири поля мітки займають загальний обсяг 32 біта. Перше поле («Метка», Label Value), розміром 20 біт, визначає шлях комутації по мітках. Друге поле (Experimental, Exp), що займає 3 біти, спочатку було зарезервовано для розвитку технології. Також це поле можна використовувати для вказівки класу трафіку, необхідного для забезпечення рівня QoS. Третє поле (Set field, S), розміром 1 біт, визначає ієрархію стека міток MPLS. У заголовку останньої мітки біт $S = 1$, а у всіх інших біт $S = 0$. Останнє поле (Time to Live, TTL), що займає 8 біт, використовується для визначення кількості діючих транзитних маршрутизаторів. Інформація цього поля дозволяє вибракувати з пакета закільцьованих або пошкоджені посилки.

У загальному випадку MPLS підтримує формати кадрів технологій PPP, Ethernet, Frame Relay і ATM. У цих кадрах можна розміщувати пакети мережевого рівня. Найпоширенішим на мережевому рівні є протокол IP. В цьому випадку мітка MPLS вбудовується в заголовок IP. Основні характеристики і параметри технології MPLS описані в Internet Engineering Task Force (IETF). Детальний опис структури мітки і методів її організації можна знайти, наприклад, в публікації .

Базовим пристроєм мережі MPLS є маршрутизатор комутації міток (Label Switch Router, LSR), який в загальному випадку визначається, як пристрій будь-якого типу, здатне створювати, змінювати і видаляти мітки MPLS в IP-пакетах [8, 9]. Маршрутизатор LSR також може виступати в ролі IP-роутера, комутатора Frame Relay, а також комутатора ATM.

Правила обробки TTL в мітці:

1. коли IP-пакет прибуває на вхідний прикордонний маршрутизатор MPLS-домену (Домен - це MPLS-мережа), в стек пакета поміщається одна мітка. Значення поля часу життя цієї мітки встановлюється рівним значенню поля часу життя IP заголовка;
2. коли MPLS-пакет прибуває на черговий транзитний маршрутизатор MPLS-домена, значення поля часу життя в мітці, що знаходиться на вершині стека, зменшується на одиницю;

- якщо вийшло значення часу життя нульове, MPLS-пакет далі не передається. Залежно від значення мітки в стеці, пакет або просто відкидається, або передається відповідному «звичайному» мережному рівню для обробки помилок (Наприклад, для формування повідомлення про помилку з використанням протоколу міжмережєвих керуючих повідомлень - Internet Control Message Protocol, ICMP);

- якщо значення часу життя вийшло позитивне, то воно буде поміщене в поле часу життя в верхній записи стека для вихідного MPLS-пакета, після цього сам MPLS-пакет перенаправляється далі. Вихідні значення поля часу життя є функцією тільки входить значення поля часу життя і не залежить від того, чи були поміщені в стек або вилучені з стека будь-які мітки до того, як переправити пакет далі. Значення полів часу життя в записах, які не перебувають на вершині стека, на хід обробки не впливають.

3. коли MPLS-пакет прибуває на вихідний прикордонний маршрутизатор MPLS домена, значення поля часу життя, єдиною знаходиться в стеку записи, зменшується на одиницю, після чого мітка вилучають із стека і стек міток стає порожнім. У цьому випадку пакет видається користувачеві або на мережєвий рівень для обробки помилок.

- якщо значення буде позитивним, то воно поміститься в поле часу життя IP-заголовка та після чого IP-пакет буде перенаправлений далі шляхом звичайної маршрутизації. Перед тим як пакет буде переправлений далі, повинна бути перерахована спочатку контрольна сума IP-заголовка. Ця процедура необхідна, щоб переконатися, що пошкодження заголовка не відбулося при пересиланні повідомлення.

Використання міток значно спрощує процедуру пересилки пакетів, так як маршрутизатор обробляє не весь заголовок IP- пакета, а тільки мітку, що займає значно менше часу.

Переваги комутації міток за методом MPLS полягають в подальшому:

— система міток значно знижує час, необхідний на пошук IP-маршрутизації;

- дозволяє здійснювати точний пошук збігів з найдовшим префіксом, що знижує ресурс звернення до пам'яті для маршрутизації одного пакета;
- точні збігу на основі міток набагато простіше реалізувати в обладнанні при меншому навантаженні на нього;
- дає можливість контролювати, де і як трафік розподілений в мережі, щоб управляти пропускнуою спроможністю, розставляти пріоритети для різних сервісів і запобігати перевантаженню устаткування.

1.3 Протокол RSVP-TE

Перша мета введення в мережу MPLS функцій підтримки протоколу RSVP полягає в тому, щоб LSR, які класифікують пакети, аналізуючи їх мітки, а не IP-заголовки, могли розпізнавати пакети, що належать тим потокам, для яких було зроблено резервування ресурсів. Іншими словами, нам потрібно створювати прив'язку міток до FEC для потоків, які забезпечені резервованими ресурсами за допомогою протоколу RSVP. Можна розглядати сукупність пакетів, для яких було виконано резервування по протоколу RSVP, як сукупність пакетів, що належать деякому новому класу FEC. У протоколі RSVP можливо доволі легко зробити прив'язку міток до водних потоків з резервуванням, принаймні, в тому випадку якщо використовувати унікальний адрес. У більш розширеній версії цього протоколу, описаного в RFC 3209 «Extensions to RSVP for LSP Tunnels» якому дісталась назва RSVP-TE. RSVP для Traffic Engineering, або якщо скорочено то RSVP-TE. Він був створений ексклюзивно для MPLS-TE. Якщо розглядати його, то його завдання буде схоже на LDP, тобто розподіляти мітки поміж LSR і обробити його в результаті LSP від одержувача до відправника. Проте з'являються деякі неприємні нюанси, це те що LSP повинен відповідати деяким умовам.

RSVP-TE дає можливість створювати головний і запасний LSP, ресурси на всіх вузлах, створювати наперед запасні шляхи, знаходити помилки на мережі,

робити швидке переправлення трафіку, уникати каналів, які фізично проходять по одному шляху.

1.4 Постановка задачі

Проаналізувавши інформації, яку було отримано в ході написання роботи, то мета наукової роботи є розробка веб-інтерфейсу з графічним інтерфейсом. Він дозволяє автоматично налаштовувати інтерфейси роутерів та протокол RSVP-TE. Він має облегшити налаштування роутерів шляхом копіювання вже згенерованого коду налаштувань в симулятор GNS3.

При розробці інтерфейсу повинно бути враховане, то що інтерфейс має бути зрозумілий як для досвідчених користувачів, так і для тих хто вперше має справу з налаштуванням даної технології. Він дозволяти інтерфейси роутерів та протокол RSVP-TE, не вимагаючи від недосвідченого користувача про знання команд конфігурацій роутерів Cisco.

При створенні веб-сторінки потрібно розробити такий графічний інтерфейс, на якому буде можливість вводити дані, а в результаті отримувати код з налаштуванням, який можна буде копіювати і вставляти в симулятор GNS3. Результатом буде створений зручний інтерфейс на якому можна буде налаштовувати маршрутизатори, а потім і всю мережу.

Постановка задачі заключається в:

1. створення схеми мережі на основі симулятора GNS3;
2. реалізація графічного інтерфейсу для налаштування маршрутизаторів за протоколом RSVP-TE;
3. перевірка розробленого веб-інтерфейсу в симуляторі GNS3.

2 МОДЕЛЮВАННЯ ТЕХНОЛОГІЇ MPLS ЗА ДОПОМОГОЮ ПРОТОКОЛУ RSVP-TE В СИМУЛЯТОРІ GNS3

2.1 Симулятор GNS3. Його особливості

GNS3 – симулятор графічної мережі. З його допомогою можливо будувати досить складні та новітні мережі. Це один із видатних ресурсів для користувачів, які хочуть вивчати і виконувати на практиці їх. GNS3 – це кросплатформене середовище з відкритим вихідним кодом утилітою, яке засноване на вже досить відомих Dynamips (емулятор для CISCO IOS), Pemu (емулятор для Cisco PIX) та Dynagen (текстовий інтерфейс для Dynamips). Симулятор надає простий та зрозумілий інтерфейс, до того ж має безліч різних інструментів. За допомогою GNS3 можливо моделювати нові конфігурації, різноманітні образи ISO, або, і зовсім розробити реконструкцію яких небуť частин досить складної мережі, що в декілька разів легше за допомогою цієї програми, ніж її використання в реальному світі і більш економічно ефективний процес. В пакет GNS3 вбудовано майже всі інсталяційні пакети. При його інсталяції на Windows ОС, обов'язково встановити драйвер WinPcap.

Для програми не важливо чи встановлений досить потужний процесор, оскільки цей симулятор, здійснює емуляції на Cisco Internetwork Operating System (IOS). IOS являється операційною системою, працююча на пристроях CISCO. Для комфортної роботи рекомендується мати потужний процесор, також рекомендується принаймні 4 Гб RAM пам'яті. На практиці він не має ніякого значення з точки навантаження. Команди будуть виконуватися по чергово при процесі всього моделювання.

Через ліцензійні обмеження, для повноцінного користування необхідно надати будь який образ якого-небудь роутера. Це потрібно, для того щоб зрозуміти чи є в наявності у користувача продукти, на яких є IOS образи. Якщо запустити програму без необхідних образів, то користувач має можливість зв'язувати наскільки легкою є програма. Для повноцінної роботи програми

необхідній хоча б один із образів IOS, при якому маємо можливість завантажувати та наслідувати операційні системи CISCO IOS, після чого є можливість створювати проекти і моделі мережі. Отже, для коректної роботи симулятора необхідний принаймні один образ IOS.

До переваг GNS3 можна віднести:

- симулятор містить цілий арсенал емульованих пристроїв;
- можливість побудови різних мереж;
- безкоштовність.

З недоліків GNS3 можна виділити наступні:

- відсутнє емулювання комутаторів;
- завищені вимоги до ресурсів систем;
- баги або глюки.

2.2 Реалізація проекту в GNS3

Початок роботи в GNS3 починається безпосередньо з головного вікна яке показано на рисунку 2.1.

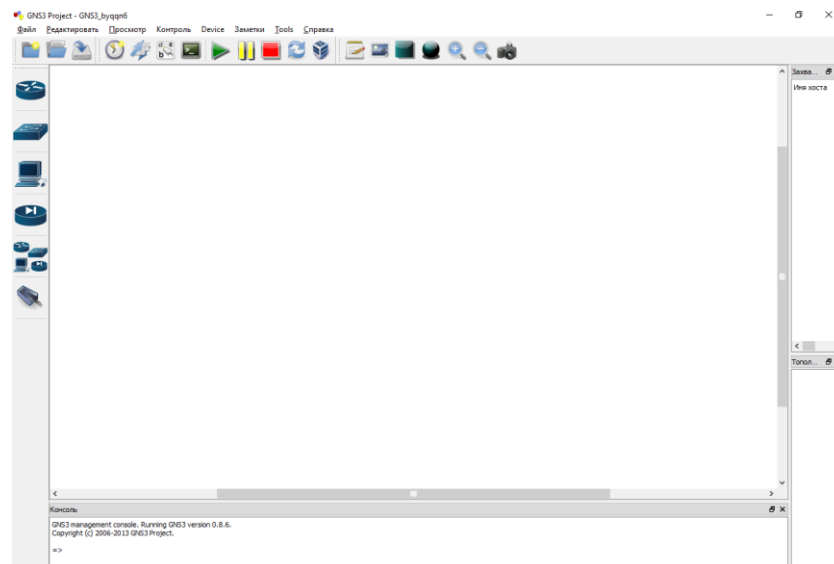


Рисунок 2.1 – Головне вікно середовища GNS3

Потім використовуючи панель пристроїв додаємо пристрої на робочий простір та з'єднуємо їх між собою.

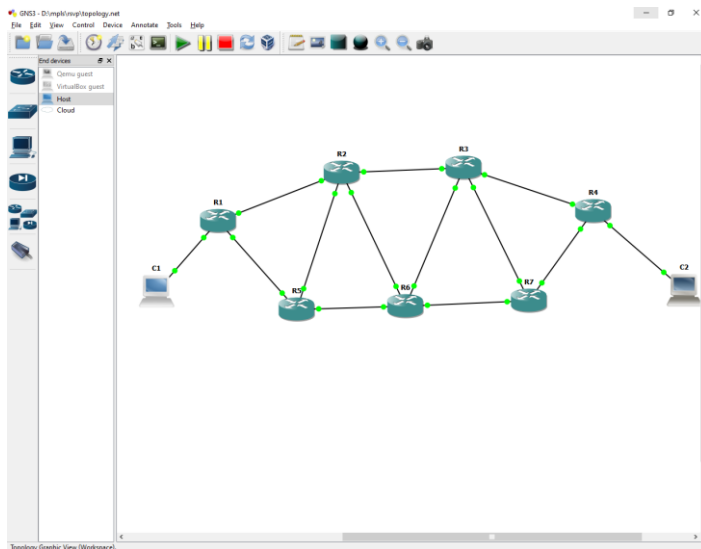


Рисунок 2.2 – Прототип схеми MPLS

Після цього додаємо IP до комп'ютерів та налаштовуємо роутери. Для того щоб налаштувати роутер необхідно вказати IP на кожному з портів, який використовується. На рисунку 2.3 наведений код з налаштування роутера R1 (Роутери 2-7 були налаштовані відповідно).

```

R1
Connected to Dynamips VM "R1" (ID 8, type c3745) - Console port
Press ENTER to get the prompt.

R1#Conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#hostname Router_1
Router_1(config)#int Loopback0
Router_1(config-if)#ip add 1.1.1.1 255.255.255.255
Router_1(config-if)#ip router isis 1
Router_1(config-if)#int fa 0/0
Router_1(config-if)#ip address 10.0.12.1 255.255.255.0
Router_1(config-if)#no sh
Router_1(config-if)#ip router isis 1
Router_1(config-if)#exit
Router_1(config)#int fa 1/0
Router_1(config-if)#ip address 10.0.15.1 255.255.255.0
Router_1(config-if)#no sh
Router_1(config-if)#ip router isis 1
Router_1(config-if)#exit
Router_1(config)#int fa 3/0
Router_1(config-if)#ip address 172.16.0.1 255.255.255.0
Router_1(config-if)#no sh
Router_1(config-if)#exit
Router_1(config)#router isis 1
Router_1(config-router)#net 10.0000.0000.0001.00
Router_1(config-router)#exit
Router_1(config)#exit

```

Рисунок 2.3 – Налаштування першого роутеру

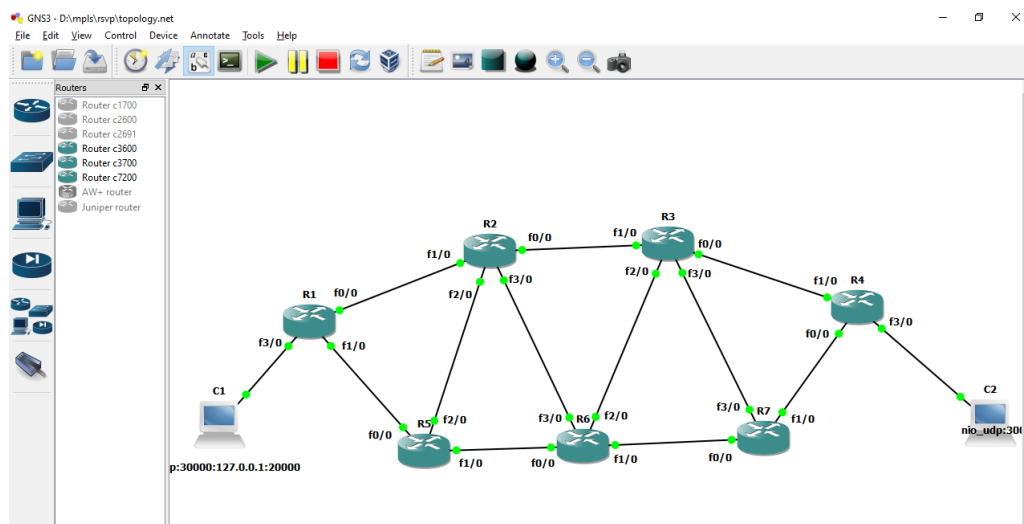


Рисунок 2.4 – Налаштування роутерів

Після додавання IP роутерах необхідно налаштувати технологію MPLS з використанням протоколу RSVP-TE. Для наочності нижче наведений код з налаштуванням MPLS роутеру R1(Router_1):

```

R1
Router_1(config)#mpls traffic-eng tunnels
Router_1(config)#int fa 0/0
Router_1(config-if)#mpls traffic-eng tunnels
Router_1(config-if)#ip rsvp bandwidth 5000
Router_1(config-if)#no sh
Router_1(config-if)#exit
Router_1(config)#int fa 1/0
Router_1(config-if)#mpls traffic-eng tunnels
Router_1(config-if)#ip rsvp bandwidth 10000
Router_1(config-if)#no sh
Router_1(config-if)#ex
Router_1(config)#exit
Router_1#w
*Mar  1 01:07:32.135: %SYS-5-CONFIG_I: Configured from console by console
Router_1#wr
Building configuration...
[OK]
Router_1#

```

Рисунок 2.5 – Налаштування MPLS

Налаштування IS-IS для можливості збору та передачі TE даних зображено на рисунку 2.6

```

R1
Router_1#
*Mar 1 01:30:03.367: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/0 (not full duplex),
  with Router_2 FastEthernet1/0 (full duplex).
Router_1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router_1(config)#router isis
Router_1(config-router)#metric-style wide
Router_1(config-router)#mpls traffic-eng router-id Loopback0
Router_1(config-router)#mpls traffic-eng level-1
Router_1(config-router)#exit
Router_1(config)#exit
Router_1#
*Mar 1 01:30:45.667: %SYS-5-CONFIG_I: Configured from console by console
Router_1#wr
Building configuration...

```

Рисунок 2.6 – Налаштування IS-IS для збору та передачі TE даних

Команда `metric-style wide` - обов'язкова тому, що TE використовує нові TLV з розширеними мітками, а за замовчуванням ISIS генерує тільки короткі.

```

R1
Connected to Dynamips VM "R1" (ID 1, type c3745) - Console port
Press ENTER to get the prompt.

Router_1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router_1(config)#interface Tunnel4
Router_1(config-if)#ip unnumbered Loopback0
Router_1(config-if)#tunnel mode mpls traffic-eng
Router_1(config-if)#tunnel destination 4.4.4.4
Router_1(config-if)#tunnel mpls traffic-eng bandwidth 8000
Router_1(config-if)#tunnel mpls traffic-eng path-option 1 dynamic
Router_1(config-if)#

```

Рисунок 2.7 – Налаштування TE-тунеля для Router_1

```

R4
Connected to Dynamips VM "R4" (ID 0, type c3745) - Console port
Press ENTER to get the prompt.

R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#interface Tunnel1
R4(config-if)#ip unnumbered Loopback0
R4(config-if)#tunnel mode mpls traffic-eng
R4(config-if)#tunnel destination 1.1.1.1
R4(config-if)#tunnel mpls traffic-eng bandwidth 8000
R4(config-if)#tunnel mpls traffic-eng path-option 1 dynamic
R4(config-if)#

```

Рисунок 2.8 – Налаштування TE-тунеля для Router_4

Після всіх вище зроблених процедур перевіряємо чи пінгуються роутери між собою (рис. 2.9) і після цього вже отримуємо робочу схему MPLS (рис. 2.10).

```

R1
Connected to Dynamips VM "R1" (ID 11, type c7200) - Console port
Press ENTER to get the prompt.

Router_1#ping 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!.
Success rate is 80 percent (4/5), round-trip min/avg/max = 324/415/516 ms
Router_1#
*Jun  2 22:24:52.483: %CLNS-4-BADIPMSK: ISIS: Inconsistent ip prefix and mask(10.0.47.0 /0) in ISIS LSP: pdb=1
, level=1, lspid=Router_7.00-00. The route wasn't added into RIB.
Router_1#ping 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!.
Success rate is 100 percent (5/5), round-trip min/avg/max = 288/349/396 ms
Router_1#

```

Рисунок 2.9 – Робота команди ping

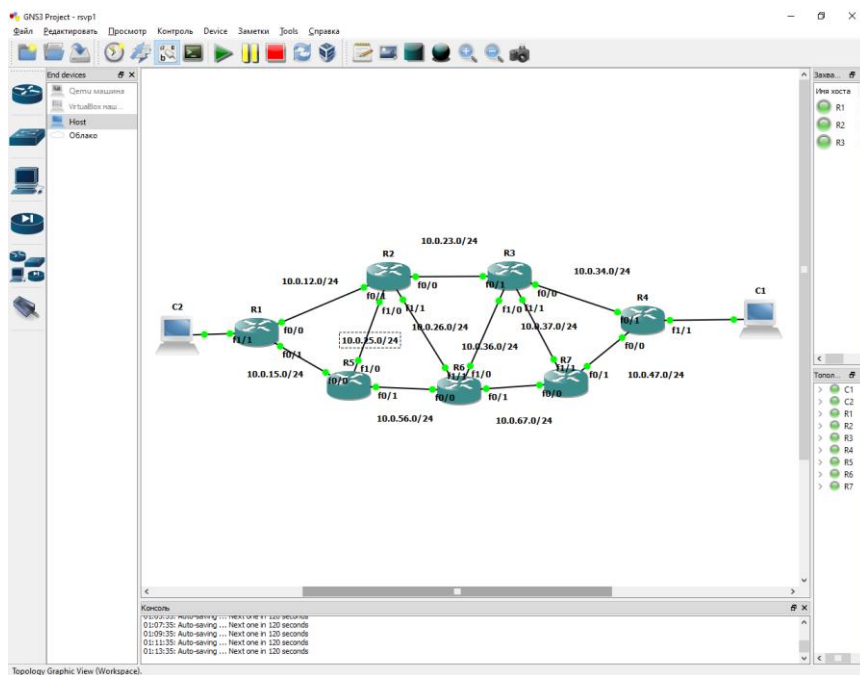


Рисунок 2.10 – Прототип готової схеми

3 РОЗРОБКА ГРАФІЧНОГО ВЕБ-ІНТЕРФЕЙСУ ДЛЯ НАЛАШТУВАННЯ MPLS ЗА ДОПОМОГОЮ ПРОТОКОЛУ RSVP-TE

3.1 Засоби розробки веб-інтерфейсу

Основою будь-якого сайту є HTML і CSS. Від них напряду залежить якість відображення сайту в різних браузерях і на різних пристроях. Валідність і семантичного HTML коду також визначає якість сайту з точки зору пошукових систем.

Кажучи про HTML і CSS не можливо не згадати фреймворки, які спрощують верстку. Вони містять в собі сітку для макетів, хорошу типографіку, готові контролери (кнопки, елементи форми) та багато іншого.

Для CSS існує кілька препроцесорів, які розширюють можливості мови. Завдяки їм спрощується розробка великих проектів, так як можливості використання її функцій, змінних і домішок істотно знижують повторення в коді і дозволяють повторно використовувати більшість ділянок коду

Мова програмування JavaScript використовується для розробки інтерфейсів сайтів, роблячи їх більш динамічними. На JavaScript реалізується велика частина гарних ефектів, які ми бачимо на сучасних сайтах (фотогалереї, слайдшоу). Ще одним з популярних типів використання є технологія AJAX, яка дозволяє без перезавантаження сторінки відправляти на сервер команди та отримувати звітні дані і вбудовувати їх в сторінку.

Говорячи про JavaScript можна не згадати популярну бібліотеку написану на ньому - це jQuery. Її використання в проектах дозволяє створювати інтерактивні сценарії поведінки інтерфейсу досить швидко і просто. Також на базі jQuery написано багато готових компонентів, що дозволяють вбудувати потрібний функціонал шляхом копіювання і вставки декількох рядків коду. Але при цьому складні інтерфейси на jQuery зазвичай дуже важко супроводжувати.

3.2 Розробка графічного інтерфейсу для налаштування MPLS за допомогою протоколу RSVP-TE

Після того як в емуляторі GNS3, де була налаштована мережа MPLS з використанням протоколу RSVP-TE, було виявлено що в GNS3 не міститься графічного інтерфейсу, щоб зручніше було налаштовувати роутерів. В такому випадку було розглянуто розробку веб-інтерфейсу для спрощення налаштування роутерів шляхом копіювання вже готового коду з цього інтерфейсу. При розробці було використано такі засоби розробки як HTML, CSS та JavaScript. При розробці повинно бути враховане, те що інтерфейс має бути зрозумілий як для досвідчених користувачів, так і для тих хто вперше має справу з налаштуванням данної технології. Він дозволяє налаштовувати інтерфейси роутерів та протокол RSVP-TE, не вимагаючи від недосвідченого користувача про знання команд конфігурацій роутерів Cisco.

Графічний інтерфейс виявився досить зручним у використанні та зрозумілим (рис. 3.1).

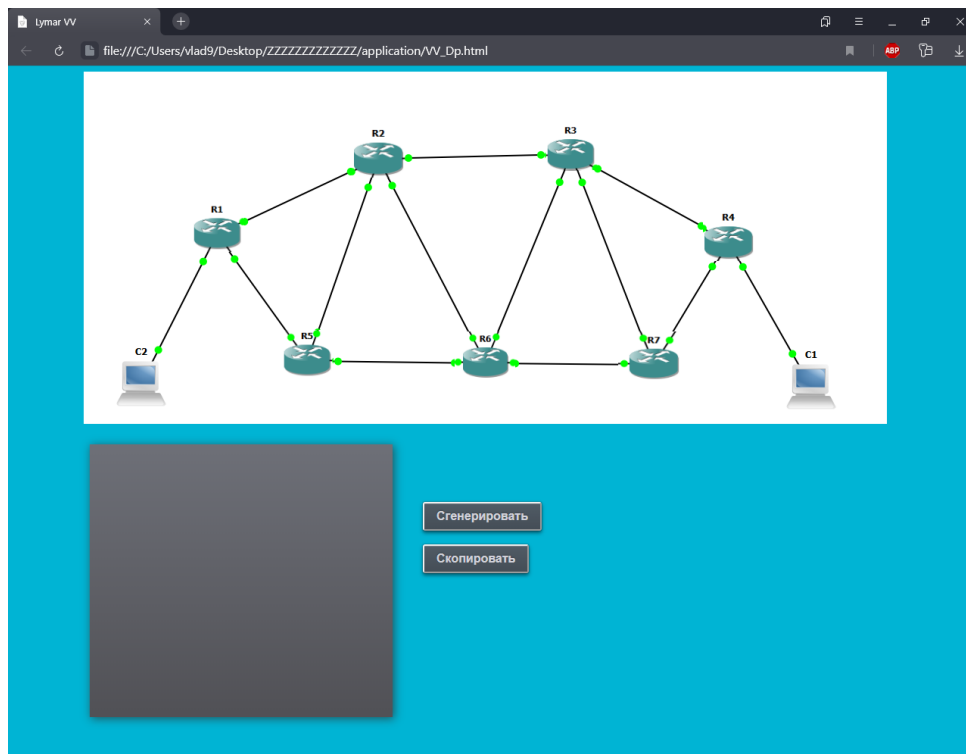


Рисунок 3.1 – Вигляд веб-інтерфейсу

Після його відкриття можна углядіти поле в якому буде генерувати код, саму схему с роутерів, кнопки «Сгенерировать» та «Скопировать».

При натисканні на будь який роутер з'являються поля для введення IP та MASK адреса, а також в полі для виведення коду з'являється шаблон коду але з пустими IP та MASK адресами(рис. 3.2). Після вводу необхідних даних та натисканні кнопки «Сгенерировать» з'являється вже готовий для копіювання код (рис. 3.3). Для облегшення копіювання була створена кнопка «Скопировать» при натисканні якої код автоматично копіюється в буфер обміну.

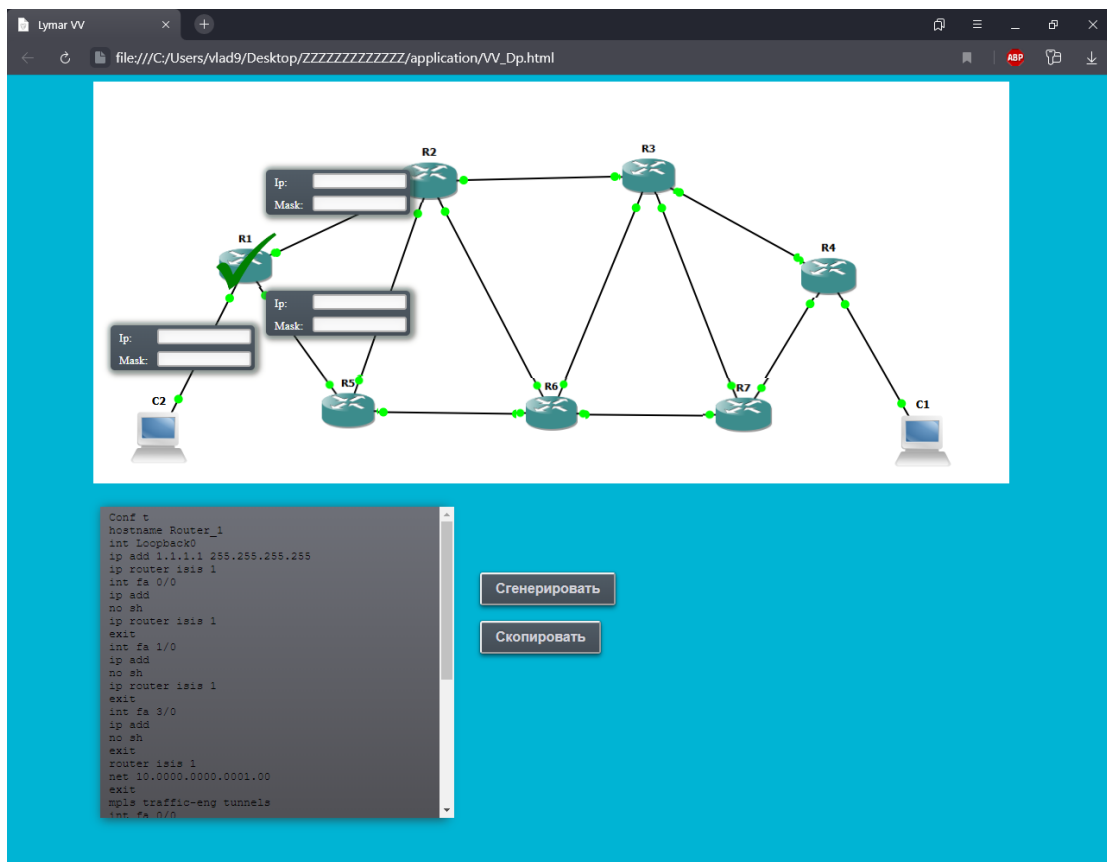


Рисунок 3.2 – Інтерфейс при натисканні на роутер

The screenshot displays a web browser window with a URL pointing to a local application. The main content is a network diagram with the following details:

- R1:** Ip: 192.168.1.1, Mask: 255.255.255.0
- R2:** Ip: 192.168.12.1, Mask: 255.255.255.0
- R3:** (No IP/Mask shown)
- R4:** (No IP/Mask shown)
- R5:** (No IP/Mask shown)
- R6:** (No IP/Mask shown)
- R7:** (No IP/Mask shown)
- C1:** (PC icon)
- C2:** (PC icon)

The configuration code in the text area is as follows:

```

Conf t
hostname Router_1
int Loopback0
ip add 1.1.1.1 255.255.255.255
ip router isis 1
int fa 0/0
ip add 192.168.12.1 255.255.255.0
no sh
ip router isis 1
exit
int fa 1/0
ip add 192.168.15.1 255.255.255.0
no sh
ip router isis 1
exit
int fa 3/0
ip add 192.168.1.1 255.255.255.0
no sh
exit
router isis 1
net 10.0000.0000.0001.00
exit
mpls traffic-eng tunnels
int fa 0/0
  
```

Рисунок 3.3 – Веб-інтерфейс з готовим кодом для першого роутеру

3.3 Тестування готового веб-інтерфейсу в емуляторі GNS3

Після створення самого веб-інтерфейсу необхідно перевірити його. Для його перевірки на працездатність отриманий код необхідно перевірити в емуляторі GNS3. Для перевірки було обрано шостий роутер (інші роуттери на схемі були зазделегідь налаштовані). Необхідно обрати роутер натисканням по ньому, далі ввести IP та MAC адреси та натиснути кнопку «Сгенерировать».

The screenshot shows a web browser window with the address bar displaying a file path. The main content area features a network diagram with routers R1 through R7 and computers C1 and C2. Below the diagram is a configuration window for Router_6 with the following text:

```

Conf t
hostname Router_6
int loopback0
ip add 6.6.6.6 255.255.255.255
ip router isis 1
int fa 0/0
ip add
no sh
ip router isis 1
exit
int fa 1/0
ip add
no sh
ip router isis 1
exit
int fa 2/0
ip add
no sh
exit
int fa 3/0
ip add __ip_4__
no sh
exit
router isis 1

```

To the right of the configuration window are two buttons: "Сгенерировать" and "Скопировать".

Рисунок 3.3 – Интерфейс при натисканні на шостий роутер

Після натискання на кнопку з'являється готовий код (рис 3.4), який копіюємо за допомогою кнопки «Скопировать» та вставляємо в консольне вікно роутеру (рис. 3.5).

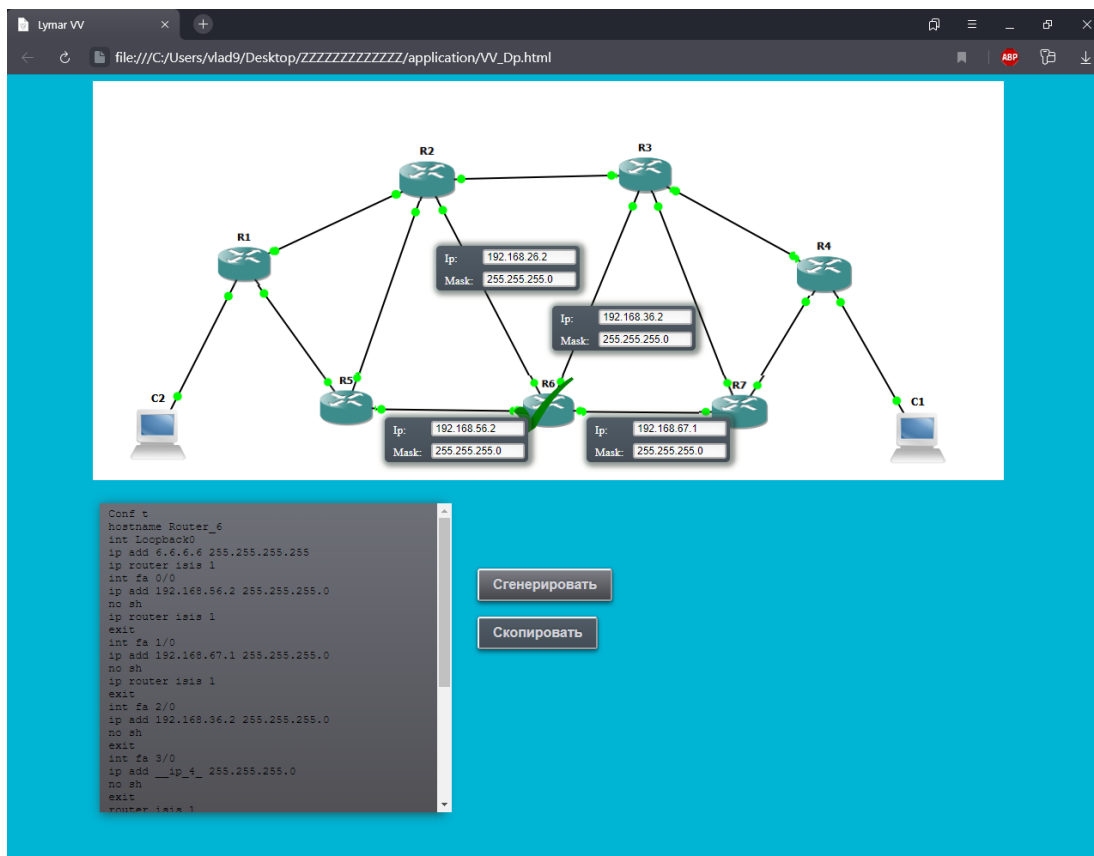


Рисунок 3.4 – Веб-інтерфейс з готовим кодом

```

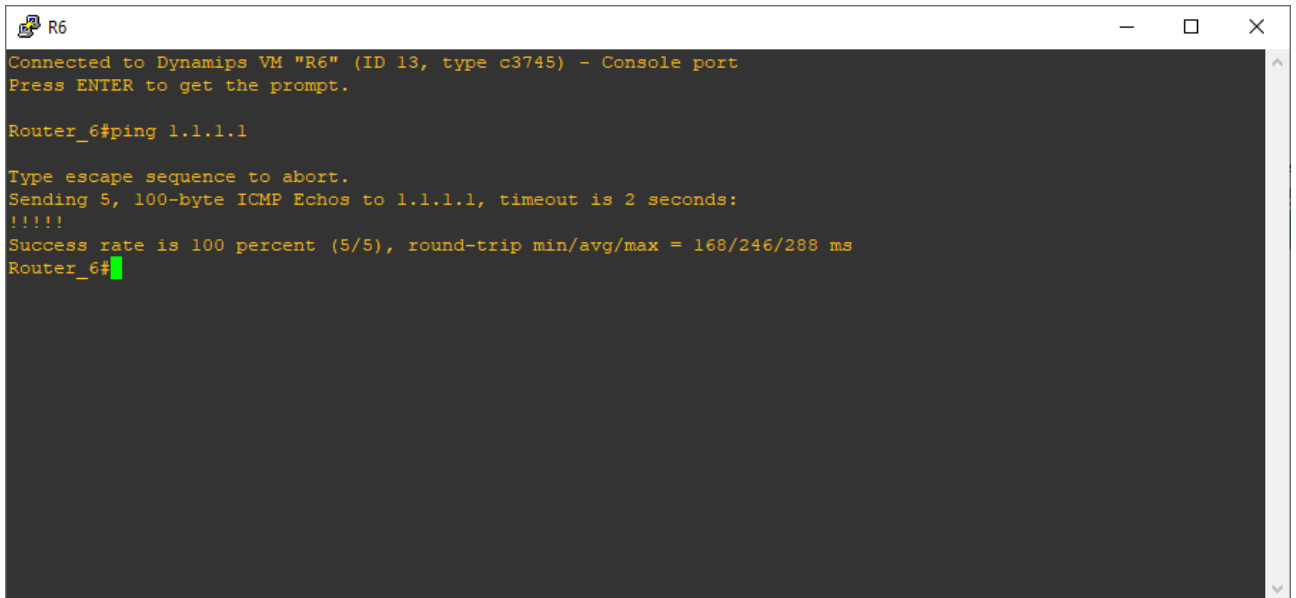
R6
Connected to Dynamips VM "R6" (ID 13, type c3745) - Console port
Press ENTER to get the prompt.

Router_6#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router_6(config)#hostname Router_6
Router_6(config)#int Loopback0
Router_6(config-if)#ip add 6.6.6.6 255.255.255.255
Router_6(config-if)#ip router isis 1
Router_6(config-if)#int fa 0/0
Router_6(config-if)#ip add 192.168.56.2 255.255.255.0
Router_6(config-if)#no sh
Router_6(config-if)#ip router isis 1
Router_6(config-if)#exit
Router_6(config)#int fa 1/0
Router_6(config-if)#ip add 192.168.67.1 255.255.255.0
Router_6(config-if)#no sh
Router_6(config-if)#ip router isis 1
Router_6(config-if)#exit
Router_6(config)#int fa 2/0
Router_6(config-if)#ip add 192.168.36.2 255.255.255.0
Router_6(config-if)#no sh
Router_6(config-if)#exit
Router_6(config)#int fa 3/0
Router_6(config-if)#ip add 192.168.26.2 255.255.255.0
Router_6(config-if)#no sh
Router_6(config-if)#exit
Router_6(config)#router isis 1
Router_6(config-router)#net 10.0000.0000.0006.00
Router_6(config-router)#exit
Router_6(config)#mpls traffic-eng tunnels
Router_6(config)#int fa 0/0
Router_6(config-if)#mpls traffic-eng tunnels
Router_6(config-if)#ip rsvp bandwidth 5000
Router_6(config-if)#int fa 1/0
Router_6(config-if)#mpls traffic-eng tunnels
Router_6(config-if)#ip rsvp bandwidth 10000
Router_6(config-if)#no sh
Router_6(config-if)#router isis
Router_6(config-router)#metric-style wide
Router_6(config-router)#mpls traffic-eng router-id Loopback0
Router_6(config-router)#mpls traffic-eng level-1
Router_6(config-router)#exit
Router_6(config)#

```

Рисунок 3.5 – Конфігурація Router_6 у консолі

Після налаштування потрібно перевірити чи працює топологія. Перевірка була виконана між шостим та першим роутером за допомогою команди «Ping 1.1.1.1». Як показано на рисунку 3.6 команда була виконана успішно та роутери пінгують один одного.



```
R6
Connected to Dynamips VM "R6" (ID 13, type c3745) - Console port
Press ENTER to get the prompt.

Router_6#ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 168/246/288 ms
Router_6#
```

Рисунок 3.5 – Робота команди ping

Отже протестувавши веб-інтерфейс стало зрозуміло що він працює як і планувалося, та значно спрощує налаштування протоколу в GNS3

ВИСНОВКИ

Дана кваліфікаційна робота бакалавра присвячена розробці програми налаштування технології MPLS з використанням протоколу RSVP-TE. Початок роботи розпочинався з вибору графічного симулятора, тому що в світі існує безліч безкоштовних, так і платних програм з вивчення теорії комп'ютерних мереж за допомогою проектування та емуляції їх роботи. Вибір пав на симулятор GNS3 тому, що він дозволяє відтворювати реальні мережі за допомогою технології віртуалізації активного обладнання. GNS3 — це відмінний додатковий інструмент для створення проектів. Він використовується при реалізації експериментують над Cisco IOS або задля перевірки деяких налаштувань, які в подальшому будуть реалізовані на реальних маршрутизаторах.

MPLS вважається багатофункціональним рішенням проблем якістю обслуговування, з якими стикаються пакетні мережі сьогодення, рішенням, яке забезпечує швидкість передачі, масштабованість, оптимізацію розподілення трафіку та ефективну маршрутизацію в пакетних мережах. Технологія MPLS підтримує протокол RSVP-TE. Цей протокол дозволяє будувати основний і запасний шлях с комутацією по міткам, резервувати ресурси на всіх вузлах, виявляти аварії на мережі, будувати заздалегідь обхідні шляхи, робити швидке перенаправлення трафіку, уникати каналів, які фізично проходять по одному шляху.

У ході виконання дипломної роботи було розроблено веб-орієнтовану інформаційну систему. Інтерфейс її дуже зрозумілий, налаштування роутерів відбувається за допомогою графічного інтерфейсу, де треба вказати ір-адреси та маски підмереж. Тому може саме цим інтерфейсом будуть в майбутньому користувалися люди для швидкого налаштування протоколу RSVP-TE.

СПИСОК ЛІТЕРАТУРИ

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы (5-е издание) – СПб.: Питер, 2016. — 992с.
2. Сети для самых матерых. Часть тринадцатая. MPLS Traffic Engineering [Электронный ресурс] - <https://habr.com/ru/post/338578/>
3. Графічний симулятор мережі gns3 [Електронний ресурс] - <https://pcuawiki.ru/rizne/9170-grafichnij-simuljator-merezhi-gns3.html>
4. Мережі VPN на базі технології MPLS. Налаштовуємо EoMPLS на маршрутизаторах Cisco [Електронний ресурс] - <https://beasthackerz.ru/uk/audio/seti-vpnna-baze-tehnologii-mpls-nastraiivaem-eompls-na-marshrutizatorah.html>
5. MPLS [Електронний ресурс] - <https://uk.wikipedia.org/wiki/MPLS>
6. MPLS: лиха беда начало [Электронный ресурс] - <https://www.osp.ru/lan/2000/01/130876/>
7. Эффективный механизм передачи данных в опорных IP-сетях с использованием технологии MPLS [Электронный ресурс] - <https://wireless-e.ru/gsm/mpls/>
8. Илья Кантор Современный учебник JavaScript в 3 книгах, 2019 . — 731с .+309с. +478с.

ДОДАТКИ

Додаток А

```

<html><head><title>Lymar VV</title>
<script
src="https://cdn.rawgit.com/zenorocha/clipboard.js/master/dist/clipboar
d.min.js"></script>

<script src="js/VV.js" type="text/javascript"></script>

<link rel="stylesheet" href="css/styleDIProb.css">
</head>

<body style="background-color: rgb(0,180,212)" >

    <br>
<div class="routers">
    <div class="stok">
        
        
        
        
        
        
        

    </div>
    <div class="blow">
        
        
        
        
        
        
        

    </div>
</div>

<input value="Сгенерировать" class="generete button"
type="button"><br><br>
<button value="Copy" class="copy-btn button" data-clipboard-
target="#code">Скопировать</button>
<div class="radio-buttons">

    <input type="radio" name="merega" id="radio" value="rsvp"
checked/>
    <label for="radio">rsvp</label>

```

```
</div>
```

```
<div class="block router1">
```

```

    <div class="polya polya-1">
        <div><label class="vvod">ip: </label><input
class="adresa" ></div>
        <div><label class="vvod">mask: </label><input
class="maska" ></div>
    </div>
    <div class="polya polya-2">
        <div><label class="vvod">ip: </label><input
class="adresa" ></div>
        <div><label class="vvod">mask: </label><input
class="maska" ></div>
    </div>
    <div class="polya polya-3">
        <div><label class="vvod">ip: </label><input
class="adresa" ></div>
        <div><label class="vvod">mask: </label><input
class="maska" ></div>
    </div>

```

```
</div>
```

```
<div class="block router2">
```

```

    <div class="polya polya-1">
        <div><label class="vvod">ip: </label><input
class="adresa" ></div>
        <div><label class="vvod">mask:
</label><input class="maska" ></div>
    </div>
    <div class="polya polya-2">
        <div><label class="vvod">ip: </label><input
class="adresa" ></div>
        <div><label class="vvod">mask:
</label><input class="maska" ></div>
    </div>
    <div class="polya polya-3">
        <div><label class="vvod">ip: </label><input
class="adresa" ></div>
        <div><label class="vvod">mask: </label><input
class="maska" ></div>
    </div>
    <div class="polya polya-4">
        <div><label class="vvod">ip: </label><input
class="adresa" ></div>

```



```

        <div><label class="vvod">mask: </label><input
class="maska" ></div>
        </div>

```

```

</div>

```

```

<div class="block router3">

```

```

        <div class="polya polya-1">
            <div><label class="vvod">ip: </label><input
class="adresa" ></div>
            <div><label class="vvod">mask: </label><input
class="maska" ></div>
            </div>
            <div class="polya polya-2">
                <div><label class="vvod">ip: </label><input
class="adresa" ></div>
                <div><label class="vvod">mask: </label><input
class="maska" ></div>
            </div>
            <div class="polya polya-3">
                <div><label class="vvod">ip: </label><input
class="adresa" ></div>
                <div><label class="vvod">mask: </label><input
class="maska" ></div>
            </div>
            <div class="polya polya-4">
                <div><label class="vvod">ip: </label><input
class="adresa" ></div>
                <div><label class="vvod">mask: </label><input
class="maska" ></div>
            </div>

```

```

</div>

```

```

<div class="block router4">

```

```

        <div class="polya polya-1">
            <div><label class="vvod">ip: </label><input
class="adresa" ></div>
            <div><label class="vvod">mask: </label><input
class="maska" ></div>
            </div>
            <div class="polya polya-2">
                <div><label class="vvod">ip: </label><input
class="adresa" ></div>
                <div><label class="vvod">mask: </label><input
class="maska" ></div>
            </div>
            <div class="polya polya-3">

```

```

        <div><label class="vvod">ip: </label><input
class="adres" ></div>
        <div><label class="vvod">mask: </label><input
class="maska" ></div>
        </div>
</div>
<div class="block router5">
        <div class="polya polya-1">
                <div><label class="vvod">ip: </label><input
class="adres" ></div>
                <div><label class="vvod">mask:
</label><input class="maska" ></div>
                </div>
                <div class="polya polya-2">
                        <div><label class="vvod">ip: </label><input
class="adres" ></div>
                        <div><label class="vvod">mask:
</label><input class="maska" ></div>
                        </div>
                <div class="polya polya-3">
                        <div><label class="vvod">ip: </label><input
class="adres" ></div>
                        <div><label class="vvod">mask:
</label><input class="maska" ></div>
                        </div>
                </div>
</div>
<div class="block router6">
        <div class="polya polya-1">
                <div><label class="vvod">ip: </label><input
class="adres" ></div>
                <div><label class="vvod">mask: </label><input
class="maska" ></div>
                </div>
                <div class="polya polya-2">
                        <div><label class="vvod">ip: </label><input
class="adres" ></div>
                        <div><label class="vvod">mask: </label><input
class="maska" ></div>
                        </div>
                <div class="polya polya-3">
                        <div><label class="vvod">ip: </label><input
class="adres" ></div>
                        <div><label class="vvod">mask: </label><input
class="maska" ></div>
                        </div>
                <div class="polya polya-4">
                        <div><label class="vvod">ip: </label><input
class="adres" ></div>

```

```

        <div><label class="vvod">mask: </label><input
class="maska" ></div>
        </div>

    </div>
    <div class="block router7">

        <div class="polya polya-1">
            <div><label class="vvod">ip: </label><input
class="adresa" ></div>
            <div><label class="vvod">mask: </label><input
class="maska" ></div>
            </div>
            <div class="polya polya-2">
                <div><label class="vvod">ip: </label><input
class="adresa" ></div>
                <div><label class="vvod">mask: </label><input
class="maska" ></div>
                </div>
                <div class="polya polya-3">
                    <div><label class="vvod">ip: </label><input
class="adresa" ></div>
                    <div><label class="vvod">mask: </label><input
class="maska" ></div>
                    </div>

            </div>

<div class="code-form" >
    <div class="code" id="code">

    </div>

</div>

<script>

new ClipboardJS('.copy-btn');

$('.blow .rout').click(function(e) {
var classes = $(e.currentTarget)[0].classList
var tabClass = '.'+classes[1]

hideAllBlocks();
hideAllRouters();

});

$('.stok .rout').click(function(e) {

```

```

$('.right-content').show();

var classes = $(e.currentTarget)[0].classList
var tabClass = '.'+classes[1]

hideAllBlocks();
hideAllRouters();

showBlock(tabClass);
showRouter(tabClass);
});

function showBlock(block) {

$(block).css('display', 'block')
$(block).addClass('active')
}

function showRouter(router) {
$('.blow '+router).css('display', 'block')

showCodeBlockIfNeeded();
}

function hideAllRouters() {
$('.blow .rout').css('display', 'none')
}

function hideAllBlocks() {
$('.block').css('display', 'none')
$('.block').removeClass('active')
}

$('.generete').click(showCodeBlockIfNeeded)

function showCodeBlockIfNeeded() {
var isValid = inputData();

if (isValid) {
    showCodeBlock();
} else {
    showCodeBlock();
}
}

function inputData() {
var ipMasksBlocks = $('.block.active .polya');
var wroteIPs = []

ipMasksBlocks.each(function(i, element) {
    var ipMasksBlock = ipMasksBlocks[i];

    var ip = $(element).find('.adress').val()
    var mask = $(element).find('.maska').val()

    wroteIPs.push({'adress': ip, 'maska':mask});
})
return wroteIPs
}

```

```

}

function showCodeBlock() {
$('.code-form').show();
generateCode();
}

var codes = {
"router1": {
  'rsvp': [
"Conf t",
      "hostname Router_1",
"int Loopback0",
"ip add 1.1.1.1 255.255.255.255",
"ip router isis 1",
"int fa 0/0",
"ip add __address_1_ __maska_1_",
"no sh",
"ip router isis 1",
"exit",
"int fa 1/0",
"ip add __address_2_ __maska_2_",
"no sh",
"ip router isis 1",
"exit",
"int fa 3/0",
"ip add __address_3_ __maska_3_",
"no sh",
"exit",
"router isis 1",
"net 10.0000.0000.0001.00",
"exit",
"mpls traffic-eng tunnels",
"int fa 0/0",
"mpls traffic-eng tunnels ",
"ip rsvp bandwidth 5000",
"int fa 1/0",
"mpls traffic-eng tunnels ",
"ip rsvp bandwidth 10000",
"no sh",
"router isis ",
"metric-style wide",
"mpls traffic-eng router-id Loopback0",
"mpls traffic-eng level-1",
"exit" ,
"interface Tunnel4",
"description To Router_4",
"ip unnumbered Loopback0",
"tunnel mode mpls traffic-eng",
"tunnel destination 4.4.4.4",
"tunnel mpls traffic-eng bandwidth 8000",
"tunnel mpls traffic-eng path-option 1 dynamic"

    ]
  },
},

```

```

"router2": {
  'rsvp': [
    "Conf t",
    "hostname Router_2",
    "int Loopback0",
    "ip add 2.2.2.2 255.255.255.255",
    "ip router isis 1",
    "int fa 0/0",
    "ip add __address_1_ __maska_1_",
    "no sh",
    "ip router isis 1",
    "exit",
    "int fa 1/0",
    "ip add __address_2_ __maska_2_",
    "no sh",
    "ip router isis 1",
    "exit",
    "int fa 2/0",
    "ip add __address_3_ __maska_3_",
    "no sh",
    "exit",
    "int fa 3/0",
    "ip add __address_4_ __maska_4_",
    "no sh",
    "exit",
    "router isis 1",
    "net 10.0000.0000.0002.00",
    "exit",
    "mpls traffic-eng tunnels",
    "int fa 0/0",
    "mpls traffic-eng tunnels ",
    "ip rsvp bandwidth 5000",
    "int fa 1/0",
    "mpls traffic-eng tunnels ",
    "ip rsvp bandwidth 10000",
    "no sh",
    "router isis ",
    "metric-style wide",
    "mpls traffic-eng router-id Loopback0",
    "mpls traffic-eng level-1",
    "exit"

  ]
},
"router3": {
  'rsvp': [
    "Conf t",
    "hostname Router_3",
    "int Loopback0",
    "ip add 3.3.3.3 255.255.255.255",
    "ip router isis 1",
    "int fa 0/0",
    "ip add __address_1_ __maska_1_",
    "no sh",

```

```

"ip router isis 1",
"exit",
"int fa 1/0",
"ip add __address_2_ __maska_2_",
"no sh",
"ip router isis 1",
"exit",
"int fa 2/0",
"ip add __address_3_ __maska_3_",
"no sh",
"exit",
"int fa 3/0",
"ip add __address_4_ __maska_4_",
"no sh",
"exit",
"router isis 1",
"net 10.0000.0000.0003.00",
"exit",
"mpls traffic-eng tunnels",
"int fa 0/0",
"mpls traffic-eng tunnels ",
"ip rsvp bandwidth 5000",
"int fa 1/0",
"mpls traffic-eng tunnels ",
"ip rsvp bandwidth 10000",
"no sh",
"router isis ",
"metric-style wide",
"mpls traffic-eng router-id Loopback0",
"mpls traffic-eng level-1",
"exit"
    ]

    },
"router4": {
    'rsvp': [
"Conf t",
        "hostname Router_4",
"int Loopback0",
"ip add 4.4.4.4 255.255.255.255",
"ip router isis 1",
"int fa 0/0",
"ip add __address_1_ __maska_1_",
"no sh",
"ip router isis 1",
"exit",
"int fa 1/0",
"ip add __address_2_ __maska_2_",
"no sh",
"ip router isis 1",
"exit",
"int fa 3/0",
"ip add __address_3_ __maska_3_",
"no sh",
"exit",
"router isis 1",
"net 10.0000.0000.0004.00",
"exit",

```

```

"mpls traffic-eng tunnels",
"int fa 0/0",
"mpls traffic-eng tunnels ",
"ip rsvp bandwidth 5000",
"int fa 1/0",
"mpls traffic-eng tunnels ",
"ip rsvp bandwidth 10000",
"no sh",
"router isis ",
"metric-style wide",
"mpls traffic-eng router-id Loopback0",
"mpls traffic-eng level-1",
"exit" ,
"interface Tunnell1",
"description To Router_1",
"ip unnumbered Loopback0",
"tunnel mode mpls traffic-eng",
"tunnel destination 1.1.1.1",
"tunnel mpls traffic-eng bandwidth 8000",
"tunnel mpls traffic-eng path-option 1 dynamic",
"exit"
    ],
    },
    "router5": {
        'rsvp': [
"Conf t",
        "hostname Router_5",
"int Loopback0",
"ip add 5.5.5.5 255.255.255.255",
"ip router isis 1",
"int fa 0/0",
"ip add __address_1_ __maska_1_",
"no sh",
"ip router isis 1",
"exit",
"int fa 1/0",
"ip add __address_2_ __maska_2_",
"no sh",
"ip router isis 1",
"exit",
"int fa 3/0",
"ip add __address_3_ __maska_3_",
"no sh",
"exit",
"router isis 1",
"net 10.0000.0000.0005.00",
"exit",
"mpls traffic-eng tunnels",
"int fa 0/0",
"mpls traffic-eng tunnels ",
"ip rsvp bandwidth 5000",
"int fa 1/0",
"mpls traffic-eng tunnels ",
"ip rsvp bandwidth 10000",
"no sh",
"router isis ",
"metric-style wide",
"mpls traffic-eng router-id Loopback0",

```



```

"mpls traffic-eng level-1",
"exit"
    ]
    },
    "router6": {
        'rsvp': [
"Conf t",
            "hostname Router_6",
"int Loopback0",
"ip add 6.6.6.6 255.255.255.255",
"ip router isis 1",
"int fa 0/0",
"ip add __address_1_ __maska_1_",
"no sh",
"ip router isis 1",
"exit",
"int fa 1/0",
"ip add __address_2_ __maska_2_",
"no sh",
"ip router isis 1",
"exit",
"int fa 2/0",
"ip add __address_3_ __maska_3_",
"no sh",
"exit",
"int fa 3/0",
"ip add __address_4_ __maska_4_",
"no sh",
"exit",
"router isis 1",
"net 10.0000.0000.0006.00",
"exit",
"mpls traffic-eng tunnels",
"int fa 0/0",
"mpls traffic-eng tunnels ",
"ip rsvp bandwidth 5000",
"int fa 1/0",
"mpls traffic-eng tunnels ",
"ip rsvp bandwidth 10000",
"no sh",
"router isis ",
"metric-style wide",
"mpls traffic-eng router-id Loopback0",
"mpls traffic-eng level-1",
"exit"
        ]
    },
    "router7": {
        'rsvp': [
"Conf t",
            "hostname Router_7",
"int Loopback0",
"ip add 7.7.7.7 255.255.255.255",
"ip router isis 1",
"int fa 0/0",

```

```

"ip add __address_1__ __maska_1__",
"no sh",
"ip router isis 1",
"exit",
"int fa 1/0",
"ip add __address_2__ __maska_2__",
"no sh",
"ip router isis 1",
"exit",
"int fa 3/0",
"ip add __address_3__ __maska_3__",
"no sh",
"exit",
"router isis 1",
"net 10.0000.0000.0007.00",
"exit",
"mpls traffic-eng tunnels",
"int fa 0/0",
"mpls traffic-eng tunnels ",
"ip rsvp bandwidth 5000",
"int fa 1/0",
"mpls traffic-eng tunnels ",
"ip rsvp bandwidth 10000",
"no sh",
"router isis ",
"metric-style wide",
"mpls traffic-eng router-id Loopback0",
"mpls traffic-eng level-1",
"exit"
    ]

    }
}

function generateCode(){
var data = inputData()
var dataIndex = -1
var netType = $('[name="meregga"]:checked').val()
var tabCodes = codes[$('.block.active')[0].className.split('
')[1]][netType]

var codesWithData = ""
for (var i in tabCodes){
    var val = tabCodes[i]

        val = val.replace('__address_1__', data[0].adress)
        val = val.replace('__address_2__', data[1].adress)

        val = val.replace('__maska_1__', data[0].maska)
        val = val.replace('__maska_2__', data[1].maska)

```

```

        val = val.replace('__adress_3_', data[2].adress)
        val = val.replace('__maska_3_', data[2].maska)

        if (data[3]){

            val = val.replace('__adress_4_', data[3].adress)
            val = val.replace('__maska_4_', data[3].maska)

        }

        codesWithData +=val + '<br>'
    }

$('.code')[0].innerHTML = codesWithData
$('#code')[0].value = codesWithData.replace(/<br>/g, '\n')
}

</script>
</body>
</html>

```

Додаток Б

```

.block {
display: none;
}

.schema {
    position: absolute;
left: 100px;
}

.blow .rout{
display: none;
}

.rout{
    width: 70px;
    position: absolute;
    cursor: pointer;
}

.vvod {
float: left;
    font-size: 10pt;
    width: 40px;
    text-transform: capitalize;
}

```

```

padding-top: 7px;
margin-left: 10px;
}

.polya{
position: absolute;
color: white;
background: rgb(81,92,102) linear-gradient(rgb(81,92,102),
rgb(69,78,87));
box-shadow: 0 0 0 4px rgba(173, 220, 221, 0.17), 2px 1px 6px 4px
rgba(10, 10, 0, 0.5);
border-radius: 5px;
}

.polya input {
margin: 4px;
width: 109px;
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;
}

.code-form {
margin-top: 380px;
margin-left: 100px;
text-align: left;
box-shadow: 2px 2px 12px 0px rgba(50, 50, 50, 0.75);
width: 400px;
height: 360px;
background: linear-gradient(rgb(110,112,120), rgb(81,81,86))
rgb(110,112,120);
}

.code {
width: 400px;
height: 350px;
top: 450px;
overflow: auto;
padding: 5px 0 5px 10px;
font-family: "Courier New";
font-size: 13;
}

.router1 .polya-1{
top: 110px;
left: 300px;
}

.router1 .polya-2{
top: 250px;
left: 300px;
}

.router1 .polya-3{
top: 290px;
}

```

```
    left: 120px;
}

.router2 .polya-1{
    top: 55px;
    left: 540px;
}
.router2 .polya-2{
    top: 100px;
    left: 300px;
}
.router2 .polya-3{
    top: 205px;
    left: 345px;
}
.router2 .polya-4{
    top: 165px;
    left: 505px;
}

.router3 .polya-1{
    top: 60px;
    left: 800px;
}
.router3 .polya-2{
    top: 55px;
    left: 570px;
}
.router3 .polya-3{
    top: 210px;
    left: 600px;
}
.router3 .polya-4{
    top: 170px;
    left: 780px;
}

.router4 .polya-1{
    top: 130px;
    left: 800px;
}
.router4 .polya-2{
    top: 280px;
    left: 800px;
}
.router4 .polya-3{
    top: 250px;
    left: 1000px;}

.router5 .polya-1{
    top: 300px;
    left: 250px;
}
.router5 .polya-2{
    top: 320px;
    left: 451px;
}
.router5 .polya-3{
```

```
    top: 210px;
    left: 350px;
}

.router6 .polya-1{
    top: 400px;
    left: 440px;
}
.router6 .polya-2{
    top: 400px;
    left: 675px;
}
.router6 .polya-3{
    top: 270px;
    left: 635px;
}
.router6 .polya-4{
    top: 200px;
    left: 500px;
}
.router7 .polya-1{
    top: 400px;
    left: 680px;
}
.router7 .polya-2{
    top: 290px;
    left: 860px;
}
.router7 .polya-3{
    top: 220px;
    left: 730px;
}

img.rout.router1 {
top: 181px;
left: 242px;

}

img.rout.router2 {
    top: 80px;
left: 455px;

}

img.rout.router3 {
    top: 72px;
left: 709px;

}

img.rout.router4 {
    top: 187px;
left: 918px;

}

img.rout.router5 {
```

```

        left: 360px;
        top: 349px;
    }
    img.rout.router6 {
        top: 351px;
        left: 590px;
    }

    img.rout.router7 {
        top: 351px;
        left: 806px;
    }

    .button {
        position: relative;
        display: inline-block;
        font-size: 100%;
        font-weight: 700;
        color: rgb(209,209,217);
        text-decoration: none;
        text-shadow: 0 -1px 2px rgba(0,0,0,.2);
        padding: .5em 1em;
        outline: none;
        border-radius: 3px;
        background: rgb(81,92,102) linear-gradient(rgb(81,92,102),
        rgb(69,78,87));
        box-shadow:
            0 1px rgba(255,255,255,.2) inset,
            0 3px 5px rgba(0,1,6,.5),
            0 0 1px 1px rgba(0,1,6,.2);
        transition: .2s ease-in-out;
        top: 550px;
        left: 540px;
    }
    .button:hover:not(:active) {
        background: linear-gradient(rgb(126,126,134), rgb(70,71,76))
        rgb(126,126,134);
    }
    .button:active {
        top: 1px;
        background: linear-gradient(rgb(76,77,82), rgb(56,57,62))
        rgb(76,77,82);
        box-shadow:
            0 0 1px rgba(0,0,0,.5) inset,
            0 2px 3px rgba(0,0,0,.5) inset,
            0 1px 1px rgba(255,255,255,.1);
        top: 550px;
        left: 540px;
    }
    .radio-buttons{
        display: none;
    }

```