

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

_____ Ігор ШЕЛЕХОВ
(підпис)

_____ 18 грудня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
з теми: «Інформаційно-комунікаційна технологія автоматизованого
налаштування мереж на основі багатопроTOCOLьної комутації міток з
використанням різних протоколів маршрутизації»
здобувача групи ІН.м-25 Анашкіна Богдана Андрійовича

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

_____ Богдан АНАШКІН
(підпис)

Керівник,
старший викладач,
кандидат фізико-математичних наук, Дмитро ВЕЛИКОДНИЙ _____
(підпис)

Суми – 2023

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

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН.м-25 Анашкіна Богдана Андрійовича

1. Тема роботи: «Інформаційна технологія прогнозування курсу валют»
затверджую наказом по СумДУ від «06» грудня 2023 р. № 1412- VI
2. Термін здачі здобувачем кваліфікаційної роботи до 18 грудня 2023 року
3. Вхідні дані до кваліфікаційної роботи: Зростаюча потреба оптимізації та автоматизації операторських мереж MPLS є актуальною задачею з огляду на масштаби таких мереж та їх значущість для функціонування інфо-комунікаційних мереж масштабу країни.
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.
2) Огляд технологій, що використовуються для прогнозування курсу валют. 3) *Розробка інтелектуальної системи з прогнозування курсу валют.* 4) *Аналіз результатів.*
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

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

Завдання прийняв до виконання _____

(підпис)

Керівник _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд технологій, що використовуються для прогнозування курсу валют</i>		
3	<i>Розробка інтелектуальної системи з прогнозування курсу валют</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти _____

(підпис)

Керівник _____

(підпис)

АНОТАЦІЯ

Записка: 57 стор., 26 рис., 1 додаток, 24 джерел.

Обґрунтування актуальності теми роботи – Тема кваліфікаційної роботи на сьогодні є актуальною тому, що присвячена розв'язанню важливої практичної задачі налаштування мереж на основі технології багатопроTOCOLЬНОЇ комутації за мітками (MPLS).

Об'єкт дослідження — комп'ютерні мережі операторського рівня.

Мета роботи — розробка десктопного додатку для покращеного та пришвидшеного налаштування даної технології.

Методи дослідження — розбиття цілісного об'єкта дослідження на складові частини з метою більш глибокого вивчення і розуміння його сутності, структури, функцій та взаємозв'язків.

Результати — створено десктопний додаток, зчитуючи введені дані, формує код для налаштування роутерів за обраним протоколом маршрутизації. Та генерує код для налаштування технології багатопроTOCOLЬНОЇ комутації міток. Проведено тестування розробленого додатку на створеній мережі та перевірка її працездатності технології MPLS.

БАГАТОПРОТОКОЛЬНА КОМУТАЦІЯ МІТОК, ПРОТОКОЛ
МАРШРУТИЗАЦІЇ, МЕРЕЖА, C#, WINDOWS FORM

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Основи мережевої комутації	7
1.2 Технологія MPLS	10
1.3 Основні Характеристики MPLS	14
1.4 Важливість розробки десктопних додатків	20
1.5 Постановка задачі	22
2 ВИБІР МЕТОДУ РІШЕННЯ	24
2.1 Основні протоколи маршрутизації	24
2.2 Налаштування мережі за допомогою протоколу EIGRP	30
2.3 Налаштування MPLS в симуляторі GNS3	33
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	36
3.1 Розробка інтерфейсу додатку	36
3.2 Опис функціоналу додатка	40
3.3 Тестування додатку	43
ВИСНОВКИ	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	47
ДОДАТОК	50

ВСТУП

Актуальність. Питання дослідження розвитку телекомунікаційних мереж важливий та перспективний у наш час. Розуміння базових речей в телекомунікації може дозволити розвиватися, як спеціалісту, так і початківцям, які зацікавлені в цій тематиці. Допомогти саме таким початківцям є першочерговою задачею нашої держави. Для забезпечення пришвидшеного налаштування технології MPLS потрібно мати зрозумілу і просту програму, яка може запускатися на всіх поширених пристроях з ОС Windows. Тому розробка може мати попит не лише серед студентів технічних вузів, але і не байдужих до розвитку нашої телекомунікаційних систем.

Об'єкт дослідження. Комп'ютерні мережі операторського рівня.

Предмет дослідження. Додаток для напівавтоматичного налаштування багатопроTOCOLЬНОЇ комутації міток з використанням графічного інтерфейсу

Гіпотеза. За допомогою розробленого додатку зменшити час, необхідний для написання кода налаштування мережі MPLS, та зменшити ризик людської помилки під час написання коду.

Наукова новизна. За допомогою розробки десктопного додатку створити алгоритм, який генерує автоматично код для налаштування протоколів маршрутизації та технології MPLS.

Структура. Кваліфікаційна робота складається зі вступу, вибір методики та інструментів для рішення поставленої проблеми, постановки задачі дослідження, опису програмного забезпечення для інформаційної системи, висновків, списку використаної літератури та додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основи мережевої комутації

Мережева комутація є фундаментальною концепцією в області комп'ютерних мереж, відіграючи ключову роль у визначенні того, як дані передаються між різними пристроями в мережі. Вона включає в себе ряд технологій та методик, які дозволяють ефективно маршрутизувати трафік від однієї точки до іншої, забезпечуючи при цьому надійність, швидкість і безпеку передачі даних. Основна мета мережевої комутації - це забезпечення взаємодії між різними компонентами мережі, будь то комп'ютери, мобільні пристрої, сервери, або інші мережеві пристрої.

У світі, де обсяги даних та їхня важливість невідомо зростають, мережева комутація стає ще більш критичною. Вона використовується в широкому спектрі застосувань, від базових домашніх та офісних мереж до складних корпоративних структур і глобальних інтернет-провайдерів. Через це розуміння основ мережевої комутації та її архітектури є важливим для будь-кого, хто працює в сфері інформаційних технологій.

Архітектура мережевої комутації включає в себе різні аспекти, від фізичного підключення мережевих пристроїв до більш складних концепцій, таких як багатопрокольна комутація міток (MPLS) або інтелектуальні мережеві алгоритми. Залежно від потреб мережі, комутація може включати в себе різні протоколи та технології, такі як Ethernet для локальних мереж або IP для маршрутизації міжмережевого трафіку[1].

Ключовим аспектом мережевої комутації є її спроможність адаптуватися до змінних умов мережі та забезпечувати високу пропускну здатність при збереженні низької затримки та високої надійності. Це досягається завдяки складним алгоритмам та архітектурі комутаційного обладнання, яке може автоматично аналізувати та оптимізувати шляхи передачі даних відповідно до поточних умов мережі.

В сучасному світі, де швидкість та ефективність передачі даних є критичними для бізнесу та особистого спілкування, мережева комутація відіграє життєво важливу роль. Ця технологія не лише дозволяє безперебійно підтримувати зв'язок між мільйонами пристроїв по всьому світу, але й є основою для розвитку нових, інноваційних способів передачі та обробки інформації. Таким чином, мережева комутація є ключовою складовою в глобальній інформаційній інфраструктурі, що продовжує розвиватися та адаптуватися до нових викликів сучасного світу.



Рисунок 1.1 – Концептуальне зображення мережі

Комутація пакетів використовується в різноманітних мережевих технологіях, від традиційних Ethernet мереж до сучасних мобільних та бездротових мереж. Вона стала основою для багатьох інтернет-технологій, включаючи IP-маршрутизацію та транспортні протоколи, такі як TCP та UDP. Особливо важливою комутація пакетів є у створенні та підтримці глобального

Інтернету, де вона забезпечує швидку та надійну передачу даних між мільйонами пристроїв по всьому світу.

Однак, з розвитком мережевих технологій та підвищенням вимог до гнучкості та ефективності використання ресурсів, комутація каналів поступово відходить на задній план, поступаючись місцем комутації пакетів. Проте, її важливість залишається у певних застосуваннях, особливо там, де потрібна висока надійність та стабільність з'єднання, як, наприклад, в контрольних системах або в деяких видів спеціалізованих телекомунікаційних послуг.

В цілому, комутація каналів залишається важливою частиною історії та розвитку телекомунікацій, надаючи основу для розуміння більш сучасних технологій мережевої комутації. Її вплив та принципи продовжують бути актуальними в певних аспектах сучасного мережевого дизайну та архітектури.

Комутація повідомлень є ще одним важливим типом комутації в мережах, який відрізняється від комутації пакетів та комутації каналів. В комутації повідомлень, повні повідомлення передаються від одного вузла до іншого, а не будучи розділеними на менші пакети. Цей метод був особливо популярний у ранній епосі комп'ютерних мереж, коли обсяги передаваних даних були відносно малі, а мережеві ресурси – обмежені[2].

Далі описано особливості для комутації повідомлень:

- Цілісність повідомлень: В комутації повідомлень, повідомлення передаються в цілому від джерела до призначення, забезпечуючи їх цілісність без необхідності повторного збирання з пакетів.
- Простота обробки: Оскільки повідомлення передаються цілком, мережеві пристрої не потребують складних механізмів для обробки та відновлення порядку пакетів, що спрощує мережеву архітектуру.
- Затримка та пропускна здатність: Передача повних повідомлень може призводити до більших затримок, особливо при великому розмірі повідомлень, та може бути менш ефективною з точки зору використання пропускної спроможності мережі.

- Масштабування: Комутація повідомлень менш гнучка у порівнянні з комутацією пакетів, особливо у великих та складних мережах з високими обсягами трафіку.
- Надійність: Хоча комутація повідомлень забезпечує простоту, вона може бути менш надійною в умовах високого трафіку або збоїв у мережі, оскільки втрата одного повідомлення може означати втрату великого обсягу даних.

Хоча сьогодні комутація повідомлень використовується рідше, ніж інші методи комутації, вона все ще може бути корисною в певних застосуваннях, де важлива простота та цілісність даних. Наприклад, в системах, де необхідно забезпечити надійну передачу великих повідомлень без розбиття їх на частини, як це може бути у випадку з деякими видами контрольних або управлінських систем. Однак, у більшості сучасних застосувань, особливо в Інтернеті та інших великих мережах, комутація пакетів залишається переважаючим методом через її гнучкість, масштабованість та ефективність.

1.2 Технологія MPLS

MPLS (Multi-Protocol Label Switching) представляє собою мережеву техніку, яка впроваджує процес комутації на основі міток для направлення даних через мережеві вузли. Відрізняючись від традиційних методів маршрутизації, які залежать від адрес та маршрутів на кожному кроці, MPLS спрощує та прискорює процес пересилання даних, асоціюючи кожен пакет з короткою міткою — числовим ідентифікатором.

Ця мітка вказує на попередньо визначений шлях (так званий "Label Switched Path" або LSP) через мережу, який встановлюється на основі політики мережі та потреб користувачів. Після встановлення цього шляху, мережеві пристрої — комутатори MPLS — здійснюють пересилання пакетів, орієнтуючись виключно на мітку, без необхідності аналізувати IP-адреси або інші інформаційні поля пакета. Такий підхід дозволяє значно зменшити час

обробки на кожному вузлі, підвищуючи загальну ефективність та швидкість мережі.

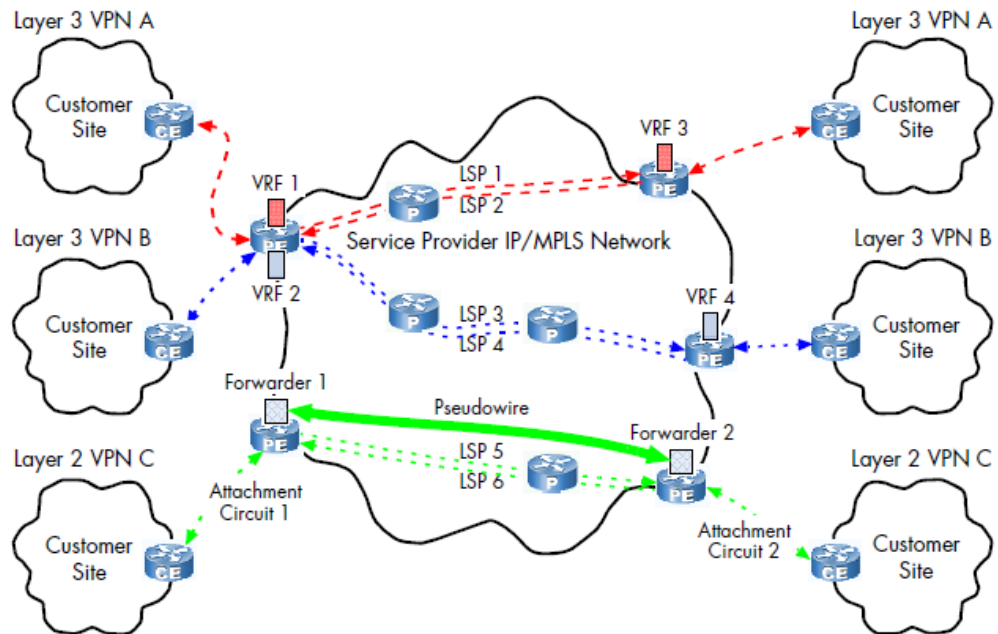


Рисунок 1.2 – Технологія MPLS [3]

На початку 1990-х років, з появою та стрімким розвитком Інтернету, стало очевидним, що існуючі методи маршрутизації потребують вдосконалення для підтримки зростаючого обсягу даних та динамічності мереж. Це спонукало компанії та дослідників до пошуку нових підходів у мережевій комутації [4].

Декілька провідних компаній у сфері мережевих технологій, таких як Cisco, Juniper та IBM, розпочали розробку своїх версій MPLS. Ці ранні версії вже містили основні принципи MPLS, такі як використання міток для спрощення та прискорення маршрутизації.

Перші реалізації MPLS стикалися з викликами, пов'язаними з необхідністю інтеграції з існуючими мережевими протоколами та обладнанням. Також, вони вимагали значних зусиль для оптимізації та налагодження, особливо у складних мережевих середовищах[5].

Варіативність ранніх реалізацій MPLS та їх нездатність ефективно взаємодіяти одна з одною призвели до потреби у стандартизації. Це стало

ключовим моментом для IETF (Internet Engineering Task Force), який взяв на себе роль координатора стандартизаційних зусиль.

Зародження MPLS відбулося на тлі стрімкого розвитку Інтернету та появи нових вимог до ефективності та гнучкості мережевих технологій у 1990-х роках. Традиційні методи маршрутизації на основі IP-адрес стикалися з обмеженнями у швидкості обробки даних та масштабованості, особливо у великих мережах. В цей період вчені та інженери почали досліджувати нові підходи до мережевої комутації, шукаючи способи оптимізації маршрутизації. Однією з ключових ідей було використання коротких міток для швидшої ідентифікації маршрутів, що дозволяло знизити затримку та підвищити продуктивність мережі. На початкових етапах розвитку MPLS, були зроблені спроби інтегрувати різні існуючі технології, такі як ATM (Asynchronous Transfer Mode) та Frame Relay. Ці технології вже використовувались для маршрутизації на основі міток, але не були достатньо гнучкими для задоволення всіх потреб сучасних мереж [6].

Важливою інновацією у розвитку MPLS стало поєднання гнучкості IP-маршрутизації з ефективністю маршрутизації на основі міток. Це дозволило створити систему, яка могла б швидко та ефективно перенаправляти трафік, використовуючи прості ідентифікатори, але при цьому залишатися сумісною з різноманітними протоколами та архітектурами мереж.

Розвиток MPLS відбувався через постійні інновації та вдосконалення. Ідеї та концепції MPLS були відкрито обговорені та розвивались в спільноті Інтернет-інженерів, що сприяло широкому прийняттю та впровадженню цієї технології[7].

MPLS революціонізувала підходи до мережевої архітектури, надаючи можливість для більш гнучкої та ефективної організації мережевого трафіку. Це відкрило двері для нових мережевих рішень, зокрема, у сферах віртуальних приватних мереж (VPN)[8], якості обслуговування (QoS) та інженерії мережевого трафіку.

Ранні реалізації MPLS зародилися у відповідь на потребу оптимізації

мережевої інфраструктури в середині 1990-х. В цей час мережі зазнавали стрімкого росту, що породжувало виклики для ефективної обробки та маршрутизації даних.

Серед перших розробників MPLS були такі компанії, як Cisco Systems, Juniper Networks та IBM. Ці організації інвестували значні ресурси у розробку нових методів мережевої комутації, які б усунули обмеження існуючих IP-мереж. Ранні реалізації MPLS були зосереджені на використанні міток для спрощення процесу маршрутизації. Вони впроваджували систему, де пакети даних отримували короткі ідентифікатори (мітки), що дозволяли комутаторам швидко перенаправляти трафік без необхідності аналізувати кожен пакет на високому рівні. Компанії, які працювали над MPLS, інтегрували цю технологію у свої мережеві продукти. Наприклад, Cisco запровадила MPLS у своїх маршрутизаторах, пропонуючи клієнтам нові можливості для створення більш ефективних та гнучких мережевих рішень[9].

Початкове впровадження MPLS у комерційні мережі було відносно повільним, оскільки це вимагало змін у стандартних мережевих практиках та адаптації до нової технології. Однак, з часом, завдяки своїм перевагам, MPLS почала набувати популярності, особливо серед провайдерів телекомунікаційних послуг та великих корпоративних мереж.

Ці ранні реалізації MPLS були критично важливими для її подальшого розвитку та прийняття. Вони демонстрували практичну цінність MPLS у поліпшенні ефективності мережевих операцій та надавали основу для подальших досліджень та розвитку цієї технології.

З появою стандартизованого MPLS, технологія швидко знайшла застосування в різних сферах. Вона стала ключовою для віртуальних приватних мереж (VPN), управління якістю обслуговування (QoS) та інженерії мережевого трафіку.

Далі MPLS продовжила свій розвиток з покращенням функціональності та впровадженням нових можливостей. Серед них - підтримка різних протоколів,

гнучкість у управлінні мережевими ресурсами та забезпечення високого рівня безпеки[10].

1.3 Основні Характеристики MPLS

Основою MPLS є використання міток для маршрутизації пакетів даних у мережі. Відмінною особливістю MPLS є те, що кожен пакет даних асоціюється з короткою міткою (label), яка використовується для визначення його шляху крізь мережу. Це суттєво відрізняється від традиційної маршрутизації, заснованої на аналізі IP-адрес пакетів.

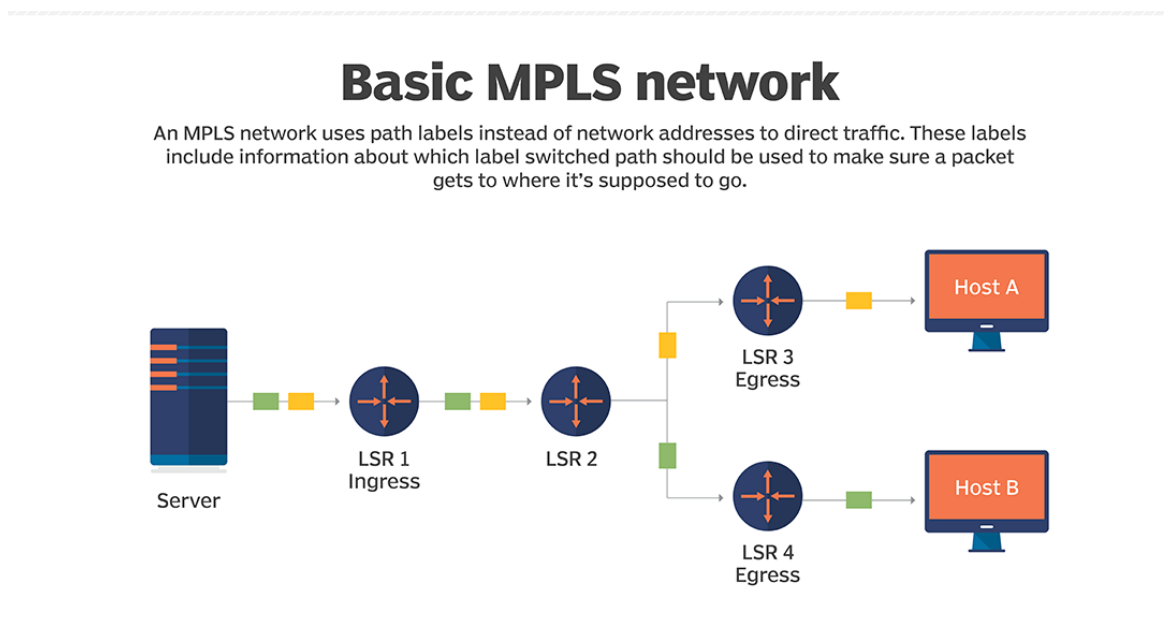


Рисунок 1.3 – Вигляд базової мережі MPLS [11]

Коли пакет входить у мережу MPLS, на нього накладається мітка, яка визначається на основі ряду критеріїв, таких як джерело, призначення або тип трафіку. Ця мітка далі використовується на кожному вузлі мережі (комутаторі MPLS) для швидкого визначення шляху передачі пакета, без необхідності перевіряти його IP-адресу або виконувати складний процес вибору маршруту.

Переваги Маршрутизації на Основі Міток:

- **Зниження Затримки:** Оскільки маршрутизація на основі міток зменшує кількість оброблюваних даних на кожному комутаторі, це призводить до зниження затримки в мережі.

- **Покращення Пропускної Спроможності:** Використання міток дозволяє комутаторам MPLS швидко перенаправляти трафік, що значно підвищує загальну пропускну спроможність мережі.
- **Гнучкість і Спрощення Управління Мережею:** Маршрутизація на основі міток надає більшу гнучкість у управлінні трафіком, оскільки мітки можуть бути змінені або адаптовані залежно від потреб мережі.
- **Підвищення Ефективності Маршрутизації:** Традиційна IP-маршрутизація вимагає часу та ресурсів для аналізу кожного пакета. Натомість MPLS забезпечує більш ефективне управління трафіком, використовуючи прості мітки для визначення шляхів передачі даних.

Маршрутизація на основі міток в MPLS ефективно використовується в різноманітних мережевих сценаріях, включаючи корпоративні мережі, постачальників інтернет-послуг та центри обробки даних. Вона забезпечує оптимальне управління трафіком у складних мережевих архітектурах, забезпечуючи високу продуктивність та надійність[12].

Масштабованість є однією з найважливіших характеристик MPLS, що дозволяє цій технології ефективно функціонувати в мережах різного розміру та складності - від невеликих корпоративних мереж до глобальних мереж провайдерів. Основою масштабованості MPLS є її здатність управляти трафіком за допомогою міток, що значно спрощує процес маршрутизації та зменшує навантаження на мережеве обладнання .

Гнучкість MPLS полягає в її здатності адаптуватися до різних мережевих вимог і політик. MPLS дозволяє створювати мережеві шляхи, звані Label Switched Paths (LSPs), які можуть бути оптимізовані для різних типів трафіку та змінюватися в залежності від умов мережі.

Великі мережі, особливо ті, що використовуються інтернет-провайдерами, виграють від впровадження MPLS, оскільки вона дозволяє оптимізувати та спрямовувати трафік більш ефективно. Завдяки міткам, MPLS зменшує кількість

необхідних маршрутизаційних рішень на кожному мережевому вузлі, що істотно покращує продуктивність мережі. MPLS надає мережевим адміністраторам гнучкі інструменти для управління трафіком, дозволяючи призначати пріоритети різним видам трафіку, наприклад, голосовим або відеоданим. Це забезпечує краще використання мережевих ресурсів та гарантує високу якість обслуговування для важливих додатків[13].

Однією з переваг MPLS є можливість легкого масштабування мережі без необхідності внесення радикальних змін у існуючу інфраструктуру. Нові LSPs можуть бути додані або змінені з мінімальними зусиллями, що робить MPLS ідеальним рішенням для швидкозростаючих або динамічно змінюваних мережевих середовищ.

MPLS може бути впроваджена в різноманітних мережевих середовищах, включаючи корпоративні мережі, дата-центри, мережі мобільних операторів та інтернет-провайдерів. Це робить її універсальним рішенням, що підходить для широкого спектру мережевих застосувань. Масштабованість та гнучкість MPLS не лише роблять цю технологію ефективною для сучасних мережевих потреб, але й забезпечують її тривалу актуальність у світі, де мережеві вимоги постійно змінюються. Завдяки цим характеристикам, MPLS продовжує бути ключовим інструментом для розробників та адміністраторів мереж по всьому світу.

Однією з головних переваг MPLS є її висока ступінь сумісності з широким спектром існуючих мережевих протоколів і технологій. Це включає, але не обмежується, IP, ATM (Asynchronous Transfer Mode), Frame Relay та Ethernet. Можливість інтеграції MPLS з цими різними технологіями робить її ідеальним вибором для різноманітних мережевих середовищ і застосунків.

MPLS розроблена таким чином, щоб легко інтегруватися з існуючими мережевими протоколами. Вона використовує механізм маркування пакетів, що дозволяє обробляти трафік без змін у первинній структурі пакетів. Це означає, що MPLS може бути впроваджена в мережу без необхідності заміни або радикального переконфігурування існуючого мережевого обладнання. MPLS

ефективно співпрацює з такими протоколами маршрутизації, як BGP (Border Gateway Protocol), OSPF (Open Shortest Path First) та іншими. Це дозволяє MPLS взаємодіяти з існуючими маршрутами та політиками маршрутизації, пропонуючи при цьому додаткові можливості оптимізації та управління трафіком.

Сумісність MPLS з існуючими протоколами робить її ідеальним рішенням для різноманітних застосунків, включаючи корпоративні мережі, мережі інтернет-провайдерів, мобільні мережі та центри обробки даних. Це дозволяє організаціям використовувати MPLS для покращення продуктивності, безпеки та ефективності їхніх мереж без необхідності повного переоснащення існуючої інфраструктури.

Сумісність MPLS з існуючими мережевими протоколами і технологіями є однією з її найважливіших характеристик, яка забезпечує широкі можливості для інтеграції та впровадження цієї технології в різноманітні мережеві середовища. Ця гнучкість і сумісність роблять MPLS важливим інструментом для будь-якої організації, що прагне оптимізувати свою мережеву інфраструктуру.

Ефективне управління трафіком є фундаментальним для забезпечення надійності, продуктивності та масштабованості мереж. У світі зростаючих обсягів даних та складності мереж, здатність точно керувати трафіком стає ключовою для виконання сучасних мережеских вимог.

Однією з переваг MPLS є її здатність гнучко розподіляти мережеві ресурси. Це досягається шляхом динамічного призначення ширини смуги та інших ресурсів різним LSPs в залежності від поточних умов мережі та політик управління. Ефективність MPLS в управлінні трафіком проявляється у різних мережеских сценаріях, від корпоративних мереж до мереж інтернет-провайдерів. MPLS забезпечує оптимальне використання ресурсів, мінімізацію затримок та забезпечення високої продуктивності для важливого трафіку[14].

MPLS ефективно управляє різними типами трафіку - від звичайних даних до критичних бізнес-додатків та сервісів реального часу. Це забезпечує, що

критичні застосунки, такі як VoIP або відеоконференції, отримують необхідні ресурси для високої продуктивності.

Хоча MPLS надає значні можливості для оптимізації трафіку, вона також ставить перед адміністраторами мереж виклики, пов'язані з налаштуванням та управлінням LSPs. Проте, завдяки гнучкості MPLS та її інтеграції з різними мережевими інструментами та аналітикою, ці виклики можуть бути ефективно вирішені. Ефективність управління трафіком у MPLS є ключовим фактором, який робить цю технологію вкрай цінною для сучасних мережевих інфраструктур. Завдяки гнучкості, масштабованості та здатності оптимізувати трафік, MPLS забезпечує високий рівень продуктивності та ефективності, що є важливим у управлінні складними мережами та забезпеченні високої якості сервісу.

У сучасних мережевих середовищах, де обсяги даних та вимоги до швидкості передачі постійно зростають, зниження затримки стає критично важливим. Затримка в мережі може негативно впливати на продуктивність застосунків, особливо тих, що вимагають високої пропускної спроможності та мінімальної затримки, таких як VoIP, відеоконференції та додатки реального часу.

MPLS покращує загальну ефективність мережі, знижуючи затримку завдяки своїм механізмам маршрутизації на основі міток. На відміну від традиційної IP-маршрутизації, де кожен маршрутизатор аналізує IP-адресу та визначає маршрут для кожного пакета, MPLS використовує простішу та швидшу модель маршрутизації. У MPLS пакетах даних присвоюються короткі мітки на межі мережі, і ці мітки використовуються для швидкого перенаправлення пакетів через мережу. Комутатори MPLS (Label Switch Routers) просто переглядають мітку та швидко пересилають пакет до наступного вузла, згідно з встановленою таблицею маршрутизації для цієї мітки. Це істотно скорочує час обробки на кожному мережевому вузлі. MPLS дозволяє адміністраторам мережі оптимізувати шляхи проходження даних, забезпечуючи менш завантажені

маршрути для критичного трафіку. Це може включати в себе визначення пріоритетних шляхів для пакетів з низькою затримкою, а також обхід точок заторів у мережі.

Зниження загальної затримки є однією з ключових переваг MPLS, яка робить цю технологію незамінною в мережевих інфраструктурах, де швидкість та надійність передачі даних мають критичне значення. Завдяки своїм механізмам маршрутизації на основі міток, MPLS забезпечує високу продуктивність, оптимізуючи мережевий трафік та знижуючи затримки в передачі даних.

Надійність мережі є ключовим фактором для підтримки безперебійної роботи критичних бізнес-процесів та послуг. У сучасних мережевих інфраструктурах, де перебої в роботі мережі можуть призводити до значних фінансових збитків та негативного впливу на репутацію, надійність стає пріоритетним аспектом.

MPLS підвищує надійність мережі за рахунок своєї архітектури та механізмів управління трафіком. Завдяки ефективній маршрутизації на основі міток та гнучкості у налаштуванні мережевих шляхів, MPLS забезпечує стабільність та високу доступність мережевих ресурсів[15].

Незважаючи на переваги MPLS у підвищенні надійності мережі, існують виклики, пов'язані з комплексністю конфігурації та управління. Ефективне впровадження MPLS вимагає детального планування та управління для забезпечення оптимальної роботи мережевої інфраструктури. Підвищення надійності мереж за допомогою MPLS є важливим аспектом для забезпечення стабільності та безперервності бізнес-процесів. Завдяки своїй гнучкості, ефективності та розширеним можливостям управління трафіком, MPLS відіграє ключову роль у створенні надійних і масштабованих мережевих рішень для сучасних організацій.

1.4 Важливість розробки десктопних додатків

Розробка додатків відіграє вирішальну роль у сучасному світі, де технології глибоко інтегровані в різні аспекти нашого життя. Додатки розробляються для автоматизації та оптимізації бізнес-процесів, підвищення ефективності та продуктивності. Користувацькі додатки, адаптовані до конкретних потреб бізнесу, можуть оптимізувати робочі процеси, зменшити кількість ручної роботи та пришвидшити прийняття рішень[16].

Добре розроблені додатки забезпечують безперебійний та інтуїтивно зрозумілий користувацький досвід, підвищуючи задоволеність клієнтів. Зручні, швидкі та багатофункціональні додатки можуть залучати та утримувати користувачів, що призводить до підвищення їхньої залученості та лояльності. Інновації та конкурентні переваги:

Розробка додатків стимулює інновації, дозволяючи організаціям створювати нові продукти, послуги або бізнес-моделі. Використовуючи нові технології та розробляючи унікальні додатки, компанії можуть отримати конкурентну перевагу на ринку та диференціювати себе від конкурентів.

Додатки часто інтегровані з можливостями збору та аналізу даних, що дає цінну інформацію для прийняття стратегічних рішень. Збираючи та аналізуючи дані користувачів, організації можуть зрозуміти вподобання клієнтів, ринкові тенденції та операційні закономірності, що полегшує прийняття рішень на основі даних[17].

Додатки забезпечують безперешкодний зв'язок і співпрацю між окремими особами та організаціями. Вони надають платформи для спілкування, обміну файлами, управління проектами та співпраці в режимі реального часу, покращуючи командну роботу та сприяючи інноваціям.

У цифрову епоху розробка додатків є ключовим фактором цифрової трансформації. Вона дозволяє компаніям використовувати цифрові технології, такі як хмарні обчислення, мобільні платформи, Інтернет речей (IoT) та штучний інтелект, щоб трансформувати свою діяльність, надавати нові послуги та

взаємодіяти з клієнтами в інноваційний спосіб.

Користувацькі додатки можуть бути адаптовані до конкретних потреб бізнесу і можуть легко пристосовуватися до мінливих вимог. Крім того, масштабовані архітектури додатків дозволяють компаніям задовольняти зростаючі потреби користувачів і масштабувати свої операції без значних перебоїв[18].

Коли справа доходить до вибору технології та мови програмування для розробки десктопного додатку, необхідно враховувати декілька факторів. В даному випадку, давайте обговоримо причини вибору Visual Studio як середовища розробки та мови програмування C# для десктопного додатку. Ось кілька причин, чому ця комбінація може бути сприятливим вибором:

1. Інтегроване середовище розробки (IDE): Visual Studio є багатофункціональним і широко використовуваним IDE для розробки додатків. Вона надає повний набір інструментів, можливостей налагодження та зручний інтерфейс, що робить розробку десктопних додатків простішою та ефективнішою. IDE пропонує безшовну інтеграцію з іншими технологіями Microsoft, такими як .NET Framework, що полегшує розробку та підвищує продуктивність.

2. Широкий вибір бібліотек та фреймворків: Visual Studio у поєднанні з мовою програмування C# надає доступ до величезної екосистеми бібліотек, фреймворків та ресурсів. Ці ресурси спрощують і прискорюють завдання розробки, дозволяючи розробникам використовувати готові компоненти для різних функціональних можливостей, таких як дизайн інтерфейсу користувача, маніпулювання даними, робота в мережі тощо. Це пришвидшує розробку та зменшує кількість коду, який потрібно писати з нуля[19].

3. Знайомство та спільнота розробників: C# - популярна мова програмування, відома своєю легкістю у вивченні та схожістю з іншими широко використовуваними мовами, такими як Java[20]. Багато розробників вже знайомі з C#, що скорочує час навчання та збільшує кількість талановитих людей,

доступних для розробки. Крім того, велика спільнота розробників означає надійну документацію, онлайн-ресурси, форуми та активну підтримку для подолання труднощів і пошуку рішень.

4. Платформна сумісність та крос-платформна розробка: C# та Visual Studio дозволяють розробляти десктопні додатки, які можуть працювати на платформі Windows з нативною підтримкою[22]. Однак, з появою .NET Core та Xamarin, розробники можуть також орієнтуватися на інші платформи, такі як macOS та Linux. Це дає можливість створювати крос-платформні десктопні додатки, забезпечуючи ширшу доступність та гнучкість.

5. Продуктивність та зручність супроводу: C# як статично типізована мова пропонує потужну перевірку під час компіляції, що допомагає виявляти помилки на ранніх стадіях та забезпечувати коректність коду. Мова також підтримує надійні об'єктно-орієнтовані парадигми програмування, що полегшує розробку модульного та підтримуваного коду[23]. Крім того, засоби розробки та налагодження Visual Studio, такі як IntelliSense та рефакторинг коду, сприяють підвищенню продуктивності розробників та допомагають забезпечити якість коду.

1.5 Постановка задачі

Задача полягає в тому, щоб розробити систему в котрій можна інтерактивно обирати схему для налаштування та тип протоколу, котрий буде використовуватись. Також розробити покрокове налаштування системи. Це дослідження буде робитися в рамках курсу кваліфікаційної роботи магістра за спеціальністю 122 «Комп'ютерні науки».

Основні вимоги, які ставляться перед програмою

- Наявність декількох протоколів маршрутизації;
- Наявність не перевантаженого інтерфейсу;
- Зрозумілий та легко підтримуємий код;
- Декілька схем для налаштування системи;
- Код для кожного окремого роутера;

- Здатність запускатися на будь-якому пристрої

Для виконання поставленої задачі потрібно вирішити наступ перелік задач:

- 1) виконання аналізу області дослідження;
- 2) визначення актуальності предметної області;
- 3) обрання стеку технологій, які будуть використовуватись під час розробки додатку;
- 4) реалізація моделі та системи для налаштування мережі;

Система також повинна мати зрозумілий інтерфейс для користувача, та мати підказки на випадок, допомогти користувачу мінімізувати помилки та практично зрозуміти технологію налаштування. Також зробити можливий розрахунок того, де найчастіше будуть робитися помилки та зробити обмеження на випадок невірно введених даних.

2 ВИБІР МЕТОДУ РІШЕННЯ

2.1 Основні протоколи маршрутизації

Протоколи маршрутизації є критично важливою частиною мережевого зв'язку і відіграють значну роль у тому, як пакети даних передаються мережею. Протокол маршрутизації - це набір правил і процедур, які дозволяють маршрутизаторам обмінюватися інформацією та визначати найкращий шлях для мережевого трафіку. Основна мета протоколів маршрутизації - забезпечити ефективну, надійну і безпечну доставку даних між мережевими пристроями. Існують різні типи протоколів маршрутизації, в тому числі вектор-відстань, стан каналу, вектор-шлях і гібридні протоколи, кожен з яких має свій набір переваг і недоліків. Вибір протоколу маршрутизації залежить від топології мережі, її розміру та вимог. Протоколи маршрутизації є важливими для мережевої інфраструктури і використовуються в різних сферах, включаючи локальні мережі (LAN), глобальні мережі (WAN) та Інтернет.

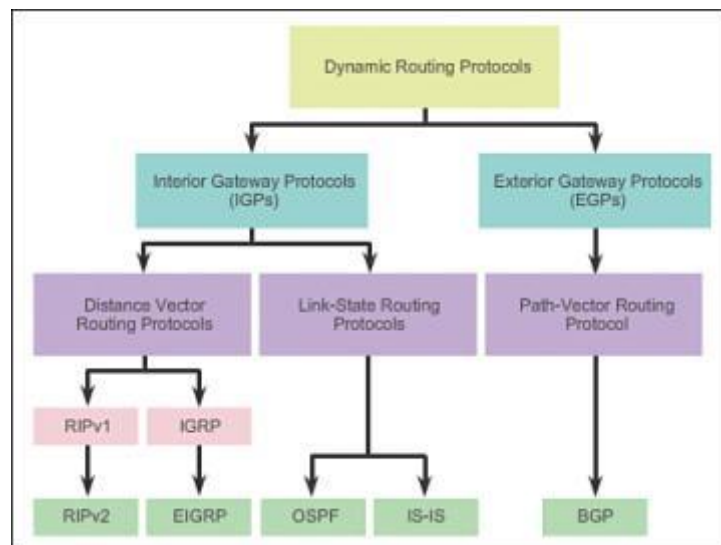


Рисунок 2.1 – Схематичне зображення протоколів [24]

EIGRP (Enhanced Interior Gateway Routing Protocol) - це розширений внутрішній маршрутизаційний протокол, розроблений компанією Cisco Systems. Як гібридний протокол, EIGRP об'єднує кращі характеристики протоколів відстаневого вектора та лінкового стану, пропонуючи швидке та ефективне маршрутизування в корпоративних мережах. Основні характеристики EIGRP

включають:

- Швидке З'єднання та Стабільність: EIGRP забезпечує швидке відновлення маршрутів при збоях, завдяки механізмам DUAL (Diffusing Update Algorithm).
- Масштабованість: Протокол легко масштабується та підходить для великих мереж.
- Ефективність: EIGRP зменшує навантаження на мережу, оптимізуючи обмін маршрутизаційною інформацією.

Розвиток EIGRP був розроблений в 1990-х роках як вдосконалена версія IGRP (Interior Gateway Routing Protocol), протоколу, також розробленого Cisco. IGRP, що був протоколом відстаневого вектора, мав обмеження з точки зору масштабованості та ефективності, особливо у великих мережеских середовищах.

Від IGRP до EIGRP: Перехід від IGRP до EIGRP ознаменував значний стрибок у функціональності. EIGRP приніс кілька ключових інновацій:

- Покращена Алгоритміка: EIGRP використовує DUAL для більш ефективного та стабільного обчислення маршрутів.
- Підтримка Різних Протоколів Мережі: EIGRP підтримує кілька протоколів мережі, таких як IP, IPX та AppleTalk.
- Гнучкість та Масштабованість: Вдосконалення в механізмах маршрутизації роблять EIGRP більш гнучким та масштабованим, ніж його попередник.

Еволюція Протоколу: З часом EIGRP еволюціонував, щоб задовольнити потреби зростаючих та різноманітних мережеских середовищ. Він пропонує підвищену надійність та ефективність у порівнянні з традиційними протоколами відстаневого вектора, водночас зберігаючи простоту налаштування та управління.

Адаптація до Сучасних Вимог: EIGRP постійно оновлюється, щоб

відповідати сучасним вимогам мережевих технологій. Його здатність адаптуватися до змінних мережевих умов та вимог робить EIGRP важливим інструментом для мережевих інженерів. EIGRP ефективно використовується в різних типах мереж, від невеликих корпоративних мереж до великих мереж інтернет-провайдерів. Його гнучкість і масштабованість роблять його придатним для широкого спектра мережевих сценаріїв.

EIGRP як протокол маршрутизації відіграє важливу роль у сучасних мережевих інфраструктурах, пропонуючи комбінацію швидкості, надійності, гнучкості та масштабованості. Його історичний розвиток з IGRP до EIGRP та постійні оновлення та вдосконалення засвідчують його значення як адаптивного та ефективного рішення для мережевої маршрутизації.



Рисунок 2.2 – Схематичне використання протоколу OSPF

OSPF (Open Shortest Path First) є одним з основних протоколів маршрутизації внутрішньої області (Interior Gateway Protocol) та використовується в багатьох IP-мережах. Він є протоколом динамічної

маршрутизації, який використовує алгоритм Дейкстри для визначення найкоротшого шляху між вузлами мережі. Основні характеристики OSPF включають:

- **Протокол Стану Лінків (Link-State Protocol):** На відміну від протоколів відстаневого вектора, OSPF використовує інформацію про стан лінків для побудови точної карти мережі.
- **Швидке Відновлення:** OSPF забезпечує швидке відновлення після збоїв у мережі, швидко перераховуючи маршрути.
- **Масштабованість:** OSPF підходить для великих та складних мереж завдяки розбиттю мережі на області.
- **Підтримка Різних Топологій:** OSPF може бути використаний у різних топологіях, включаючи нестабільні з'єднання, такі як супутникові або бездротові мережі.

Огляд Історичного Розвитку: OSPF був розроблений в 1980-х як вдосконалення протоколів маршрутизації, які були доступні на той час. Його створення було спрямоване на подолання обмежень та проблем, пов'язаних з протоколами маршрутизації відстаневого вектора, такими як RIP (Routing Information Protocol).

Еволюція та Версії OSPF:

- **OSPFv2 для IPv4:** Перша версія OSPF (OSPFv2), описана в RFC 2328, була призначена для мереж з IPv4. Ця версія включала підтримку різних типів мереж та алгоритмів, що забезпечували високу ефективність та масштабованість.
- **OSPFv3 для IPv6:** З розвитком IPv6 виникла необхідність адаптувати OSPF під новий протокол інтернету. OSPFv3, описаний в RFC 5340, включає підтримку IPv6, забезпечуючи розширену адресацію та покращену безпеку.

Border Gateway Protocol (BGP) — це протокол шлюзу, який забезпечує обмін інформацією про маршрутизацію між різними автономними системами (AS) в Інтернеті. BGP дозволяє мережам спілкуватися та вирішувати, як маршрутизувати трафік між ними, обмінюючись інформацією про доступність мережі та найкращі шляхи для пересилання пакетів.

BGP був створений наприкінці 1980-х років як заміна своєму попереднику, Exterior Gateway Protocol (EGP), який міг обробляти маршрутизацію лише в межах одного адміністративного домену. BGP було введено для задоволення потреби в масштабованому та надійному протоколі, який міг би впоратися зі зростаючою складністю з'єднання автономних систем в Інтернеті, що розвивається.

Векторний протокол шляху: BGP — це векторний протокол шляху, що означає, що він повідомляє інформацію про шлях, по якому має пересилатися пакет. Це дозволяє BGP приймати інтелектуальні рішення щодо маршрутизації на основі багатьох атрибутів, таких як політика мережі, довжина шляху та показники шляху.

Маршрутизація на основі AS: основною функцією BGP є обмін інформацією про маршрути між автономними системами (AS). Автономна система — це сукупність IP-мереж під керуванням одного оператора. BGP дозволяє AS спілкуватися та обмінюватися інформацією про мережі, до яких вони можуть дістатися, і як до них дістатися.

Маршрутизація на основі політики: BGP надає розширені можливості політики маршрутизації, що дозволяє операторам мережі визначати та застосовувати свої пріоритетні політики маршрутизації. Це включає в себе контроль шляхів трафіку, впровадження інженерії трафіку та фільтрацію маршрутів на основі різних критеріїв, таких як довжина префікса, шлях AS і теги спільноти.

Стабільний і надійний: BGP розроблено для забезпечення стабільності та надійності рішень щодо маршрутизації. Він використовує різні механізми, такі

як фільтрація маршрутів, вибір шляху та пом'якшення маршруту, щоб гарантувати, що рішення щодо маршрутизації є послідовними, ефективними та стійкими до збоїв або коливань мережі.

Масштабований та ієрархічний: BGP розроблено для масштабування для великої кількості автономних систем в Інтернеті. Це досягає масштабованості завдяки ієрархічній маршрутизації, де маршрутизатори агрегують і узагальнюють інформацію про маршрутизацію, щоб зменшити розмір таблиць маршрутизації та мінімізувати вплив оновлень маршрутизації на мережу.

Повільна конвергенція: через свою консервативну природу та зосередженість на стабільності час конвергенції BGP зазвичай нижчий порівняно з внутрішніми протоколами шлюзу. Це пояснюється тим, що BGP враховує різні міркування перед прийняттям рішення про маршрутизацію, включаючи вимоги політики та стабільність мережі.

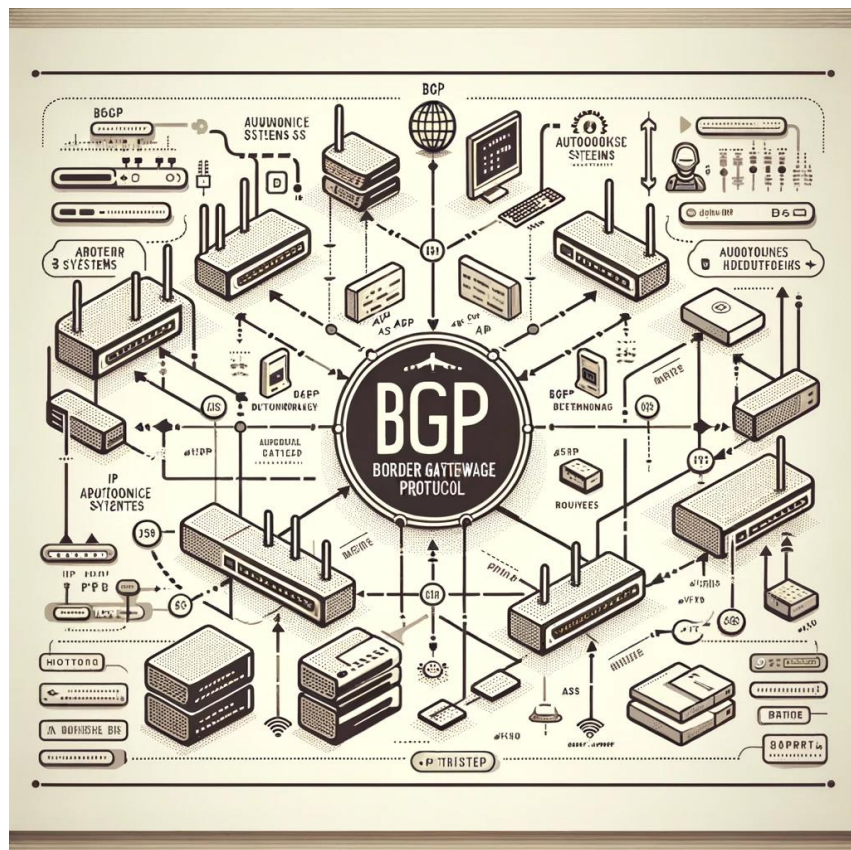


Рисунок 2.3 – Схематичне застосування BGP

BGP-4: впровадження BGP-4 у 1994 році принесло значні покращення

протоколу. BGP-4 запровадив підтримку безкласової міждоменної маршрутизації (CIDR), що дозволило більш ефективне розподілення та маршрутизацію IP-адрес.

Багатопротокольний BGP (MBGP): із розвитком Інтернету та необхідністю підтримки кількох протоколів було розроблено MBGP. MBGP дозволяє обмінюватися інформацією про маршрутизацію для різних протоколів мережевого рівня, таких як IPv4, IPv6 і багатоадресна передача, дозволяючи Інтернету обробляти різноманітний мережевий трафік.

З часом BGP адаптував і включив функції для вирішення сучасних вимог і викликів у маршрутизації Інтернету. Удосконалення безпеки: BGP включив удосконалення безпеки для зменшення ризиків, таких як атаки маршрутизації, викрадення маршруту та зловмисне поширення маршруту. Такі функції, як Route Origin Validation (ROV) та Resource Public Key Infrastructure (RPKI), забезпечують механізми для забезпечення автентичності та цілісності інформації про маршрутизацію.

2.2 Налаштування мережі за допомогою протоколу EIGRP

Для виконання налаштування схеми буде використана програма GNS3. Основні критерії чому саме була обрана програма GNS3:

GNS3 (Graphical Network Simulator-3) є потужним мережевим програмним забезпеченням, яке дозволяє симулювати комплексні мережеві структури. Програма може використовуватися інженерами та студентами для моделювання віртуальних мереж, включаючи різноманітне мережеве обладнання: маршрутизатори, перемикачі та файрволи, від різних виробників. Програмне забезпечення GNS3 може пропонувати гнучкі можливості для тестування мережевих конфігурацій, дозволяючи користувачам експериментувати та вивчати мережеві принципи маршрутизації та протоколи в контрольованому віртуальному середовищі. Програма дозволяє без необхідності використання фізичного мережевого обладнання, що робить його достатньо цінним інструментом для навчання та розробки мереж.

До програмному додатку для симуляції GNS3 пропонується така схема:

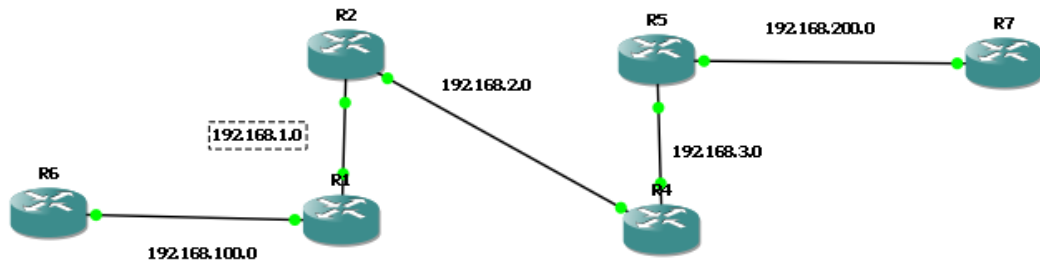


Рисунок 2.4 – Топологія тестової мережі

Для налаштування роутерів, обрано протокол EIGRP, тому що в протокола EIGRP висока ефективність у маршрутизації також швидкість для знаходження та перебудови маршрутів у мережі. Крім того, протокол також підтримує гнучкі механізми налаштування та масштабування, саме це робить його ідеальним для складних і динамічних мережевих середовищ. Використаємо приклад коду для налаштування роутера R1:

```
conf t – відкрити режим конфігурації для роутера
interface fa 0/0 - перехід до інтерфейсу fa 0/0
ip add 193.169.10.3 254.254.254.0 – код для зберігання адреси та маски
no sh – зробити вихід з інтерфейсу
exit
interface fa 0/1
ip add 193.169.10.2 254.254.254.0
no sh
exit
```

Для кожного порту, котрий наявний у роутері потрібно прописати його ір-адресу та маску. Під час налаштування потрібно не помилитися який порт налаштується в мережі, для того, щоб порти мали однакову адресу, тільки в

такому випадку налаштування буде вірне. Для полегшення сприйняття інформації, а саме який номер у кожної мережі, рекомендовано робити помітки на мережею.

Для налаштування протоколу EIGRP потрібно ввести наступні команди:

`router eigrp 200` – Код для входу в налаштування протоколу EIGRP

`network 193.169.1.0 1.1.1.255` – Сусідня мережа, до якої додається зворотня маска

`network 193.169.110.0 1.1.1.255` – Друга сусідня мережа, зі зворотною маскою

Даний набір команд потрібно вводити одразу після налаштування ір-адреси та маски для порта.

По завершенні всіх налаштувань, слід здійснити перевірку ефективності нашої мережі. Для цього необхідно ввести команду `ping 192.168.3.3` на роутері R1. Якщо конфігурація була виконана правильно, результат повинен відповідати тому, що представлено на рисунку:

```
R1#ping 192.168.3.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/96/172 ms
```

Рисунок 2.5 – Командна строка R1

На данному рисунку після маніпуляції з налаштуванням мережі, робимо контрольну спробу з'єднання роутера R1 до роутера R5. Зроблено це через надходження пакетів ICMP від роутера R1 до роутера R5. Важливо пам'ятати, що саме на цьому етапі потрібно правильно налаштувати нашу мережу. Після контрольної перевірки можна переходити до наступного етапу

2.3 Налаштування MPLS в симуляторі GNS3

Для налаштування технології MPLS необхідно для кожного роутера який знаходиться в налаштовуючій мережі, прописати наступні команди:

`ip cef` - запускаємо Cisco Express Forwarding – ця технологія для швидкої комутації за мітками

`mpls ip` - запуск глобального процесу комутації за мітками

`mpls label protocol ldp` – потрібно обрати протокол, за яким обмінюються мітками LSR (ELSR) між собою роутери

`mpls ldp router-id loopback 0` – потрібно визначити, який інтерфейс буде братися як ID роутера в процесі MPLS.

`interface fa 0/0`

`mpls ip` – запуск технології MPLS для інтерфейсу.

`mpls mtu 1512` – потрібно збільшити розмір mtu для, для запобігання фрагментації фреймів.

`exit`

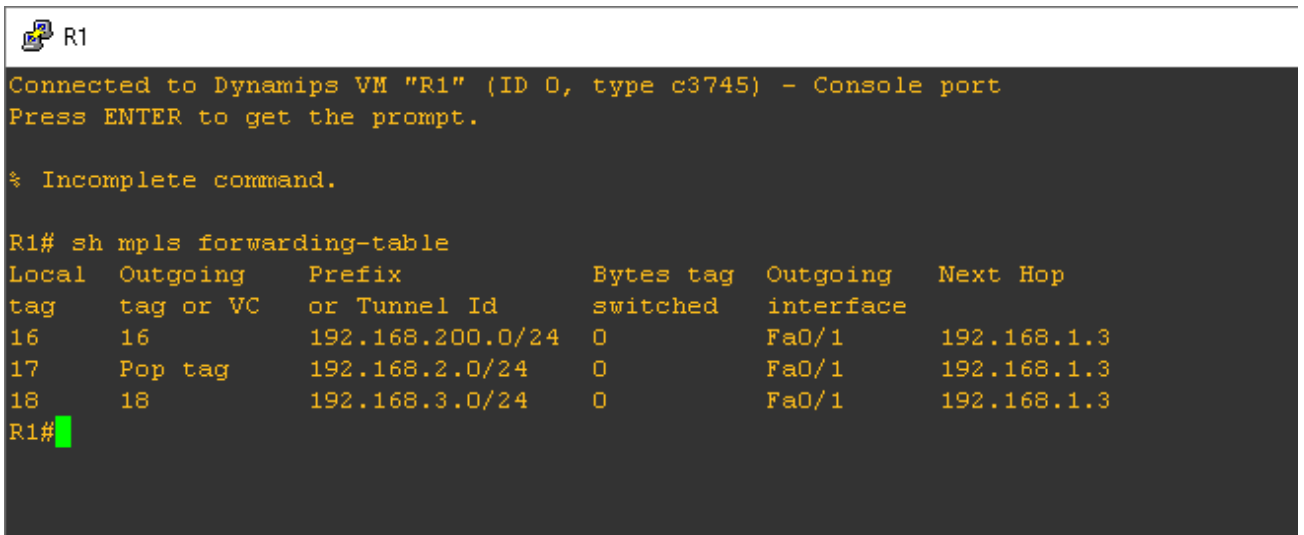
`interface fa 0/1`

`mpls ip`

`mpls mtu 1512`

`exit`

Після заповнення даними командами на кожному роутері, потрібно перевірити, що технологія MPLS вдало налаштована, для цього потрібно використати команду `sh mpls forwarding-table`, команду потрібно вводити на будь-якому з роутерів, на котрих налаштована технологія MPLS.



```

R1
Connected to Dynamips VM "R1" (ID 0, type c3745) - Console port
Press ENTER to get the prompt.

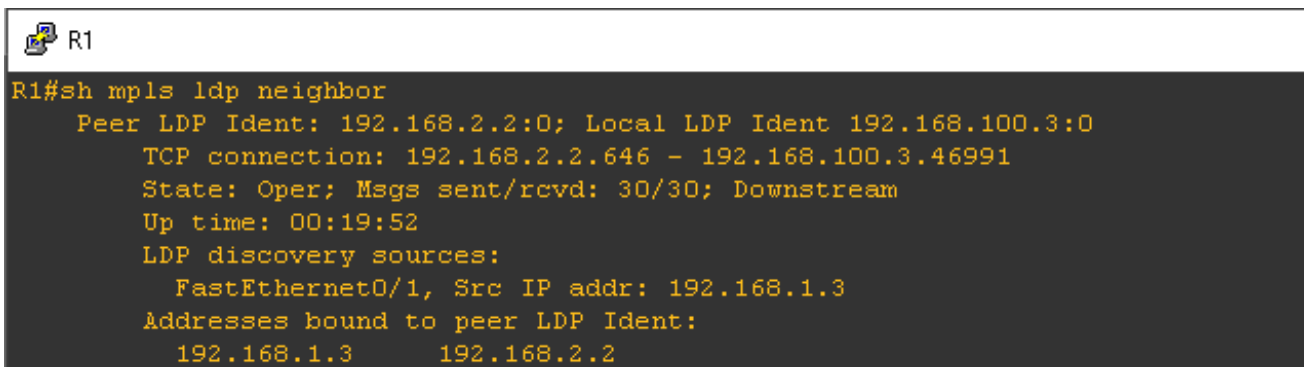
% Incomplete command.

R1# sh mpls forwarding-table
Local   Outgoing   Prefix          Bytes tag  Outgoing   Next Hop
tag     tag or VC  or Tunnel Id    switched  interface
16      16         192.168.200.0/24  0         Fa0/1      192.168.1.3
17      Pop tag    192.168.2.0/24  0         Fa0/1      192.168.1.3
18      18         192.168.3.0/24  0         Fa0/1      192.168.1.3
R1#

```

Рисунок 2.6 – Результат роботи команди sh mpls forwarding-table

Коли було перевірено, що мережа налаштована вдало, потрібно ввести наступну команду sh mpls ldp neighbor, для того, щоб наочно побачити, які сусідні мережі ідентифікує роутер, на якому було введено команду.



```

R1
R1#sh mpls ldp neighbor
  Peer LDP Ident: 192.168.2.2:0; Local LDP Ident 192.168.100.3:0
  TCP connection: 192.168.2.2.646 - 192.168.100.3.46991
  State: Oper; Msgs sent/rcvd: 30/30; Downstream
  Up time: 00:19:52
  LDP discovery sources:
    FastEthernet0/1, Src IP addr: 192.168.1.3
  Addresses bound to peer LDP Ident:
    192.168.1.3      192.168.2.2

```

Рисунок 2.7 – Результат роботи команди sh mpls ldp neighbor

В відповіді яку було отримано від роутера можна дійти висновку, що для роутера R1 відображається його сусідній роутер R2.

Для фінальної перевірки ефективності мережі MPLS ми використаємо Wireshark, відому і широко застосовувану утиліту для аналізу мережевих протоколів. За допомогою цієї програми можна детально оглянути процеси, що відбуваються у мережі на базовому рівні, що робить її незамінною у багатьох комерційних і некомерційних організаціях, урядових установах та освітніх інституціях. Відправляємо пакет від роутера R6 до роутера R7

No.	Time	Source	Destination	Protocol	Length	Info
26	22.0094970	192.168.1.2	224.0.0.2	LDP	76	Hello Message
27	22.7488940	192.168.1.3	224.0.0.2	LDP	76	Hello Message
28	25.8323860	192.168.1.3	224.0.0.10	EIGRP	74	Hello
29	25.8880000	192.168.1.2	224.0.0.2	LDP	76	Hello Message
30	26.0002070	192.168.1.2	224.0.0.10	EIGRP	74	Hello
31	27.0750530	c4:00:34:dc:00:01	c4:00:34:dc:00:01	LOOP	60	Reply
32	27.6181650	192.168.1.3	224.0.0.2	LDP	76	Hello Message
33	28.1224020	c4:01:34:dc:00:00	c4:01:34:dc:00:00	LOOP	60	Reply
34	30.2480780	192.168.100.2	192.168.200.3	ICMP	118	Echo (ping) request id=0x0001, seq=0/0, ttl=254 (reply in 35)
35	30.3680290	192.168.200.3	192.168.100.2	ICMP	114	Echo (ping) reply id=0x0001, seq=0/0, ttl=252 (request in 34)
36	30.4228930	192.168.100.2	192.168.200.3	ICMP	118	Echo (ping) request id=0x0001, seq=1/256, ttl=254 (reply in 37)
37	30.5307360	192.168.200.3	192.168.100.2	ICMP	114	Echo (ping) reply id=0x0001, seq=1/256, ttl=252 (request in 36)
38	30.5751300	192.168.100.2	192.168.200.3	ICMP	118	Echo (ping) request id=0x0001, seq=2/512, ttl=254 (reply in 40)
39	30.7022940	192.168.1.2	224.0.0.10	EIGRP	74	Hello
Frame 36: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0 Ethernet II, Src: c4:00:34:dc:00:01 (c4:00:34:dc:00:01), Dst: c4:01:34:dc:00:00 (c4:01:34:dc:00:00) MultiProtocol Label Switching Header, Label: 16, Exp: 0, S: 1, TTL: 254 Internet Protocol Version 4, Src: 192.168.100.2 (192.168.100.2), Dst: 192.168.200.3 (192.168.200.3) Internet Control Message Protocol						

Рисунок 2.8 – Результат перехоплення пакету

Отже, на рисунку 2.8 ми можемо спостерігати зображення перехопленого пакету даних, який містить заголовок MultiProtocol Label Switching (MPLS). Виявлення цієї конкретної мітки на пакеті дозволяє нам зробити висновок про коректну та ефективну роботу мережі.

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Розробка інтерфейсу додатку

Для створення додатку було обрано Microsoft Visual Studio, оскільки це інтегроване середовище розробки пропонує значні переваги. По-перше, воно доступне безкоштовно. По-друге, воно має інтуїтивно зрозумілий інтерфейс, що полегшує процес розробки. По-третє, у мережі існує велика кількість ресурсів та матеріалів, що допомагають у написанні різноманітних програм. В процесі налаштування були виявлені деякі аспекти, які можуть бути спрощені за допомогою цього додатку.

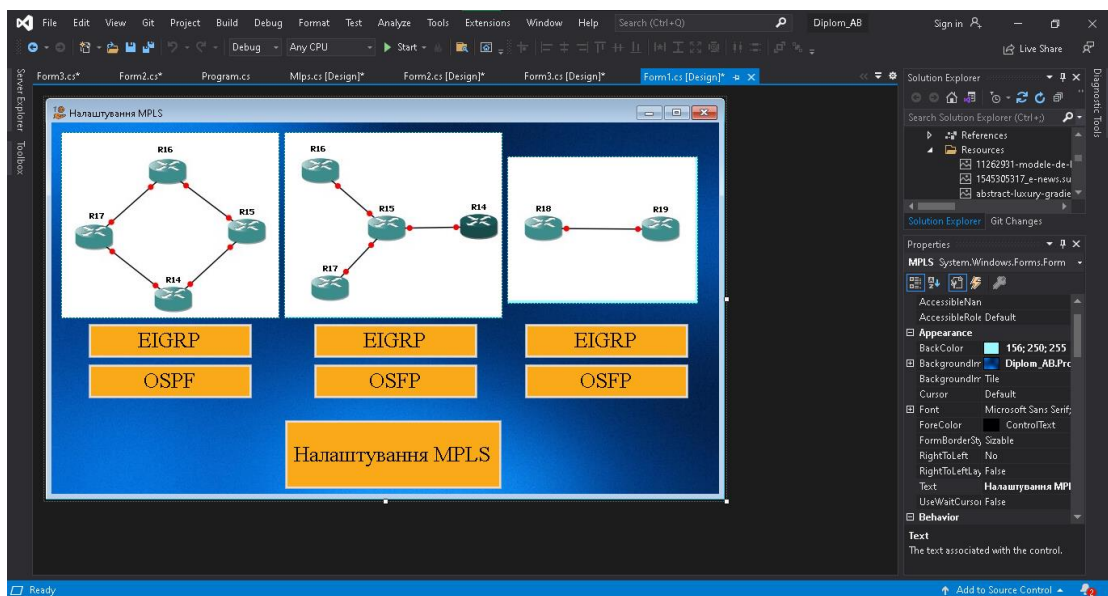


Рисунок 3.1 – Додатку в середовищі розробки Microsoft Visual Studio

Після відкриття додатку, користувач може побачити сім активних кнопок, які допоможуть налаштувати мережу MPLS. Користувача зустрічає доволі простий інтерфейс, це був одним із критеріїв при створенні додатку. Для налаштування доступні тільки два протоколи. Це протокол EIGRP та OSPF

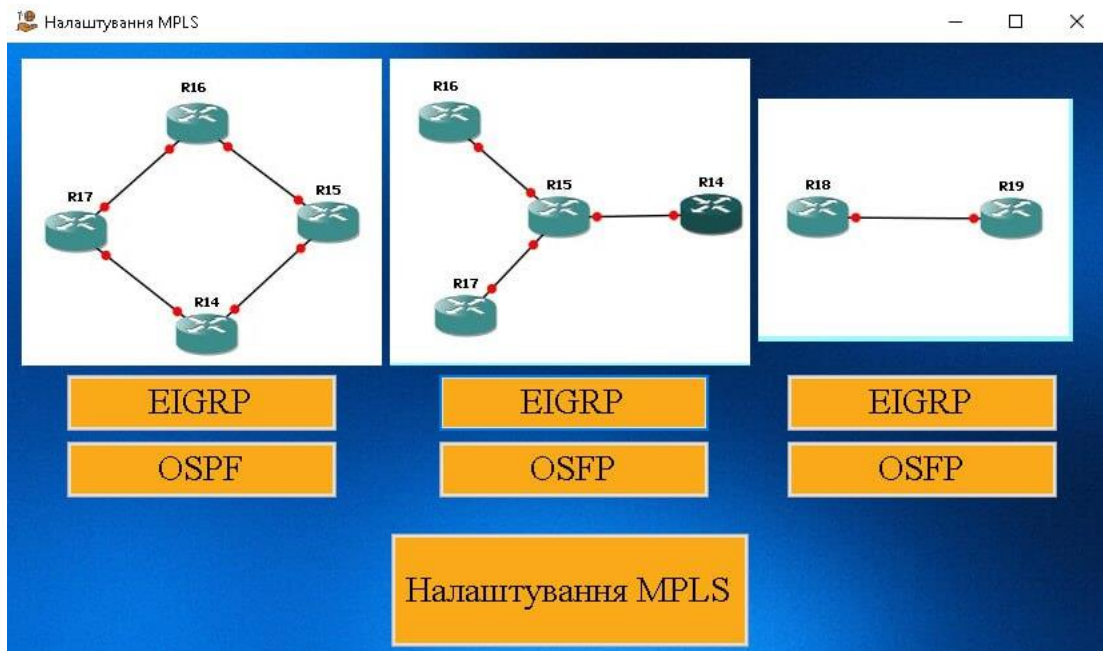


Рисунок 3.2 – Головне меню додатку

При натисканні на «Налаштування EIGRP» користувачу буде доступне поле з налаштуванням IP-адреси роутера та протоколу EIGRP. На данному етапі створення додатку, даних необхідних для налаштування цього протоколу достатньо.



Рисунок 3.3 – Меню налаштування EIGRP

На зображенні 3.3 відображені вхідні поля для введення IP-адрес та мережеских масок для різних інтерфейсів, а також поля, призначені для конфігурації EIGRP. Це включає поля для адрес сусідніх мереж, інверсних масок цих мереж та ідентифікаційних номерів мережі EIGRP. Щоб налаштувати мережу, необхідно просто ввести відповідні налаштування на кожному етапі, перш ніж перейти до останньої стадії конфігурації роутерів.

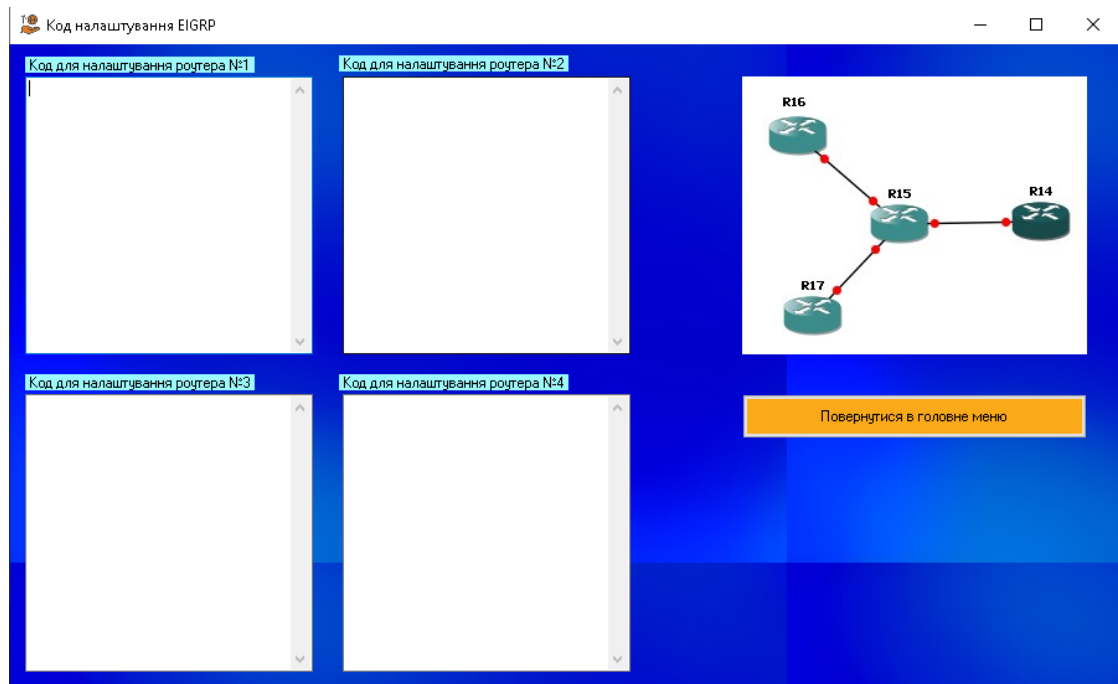
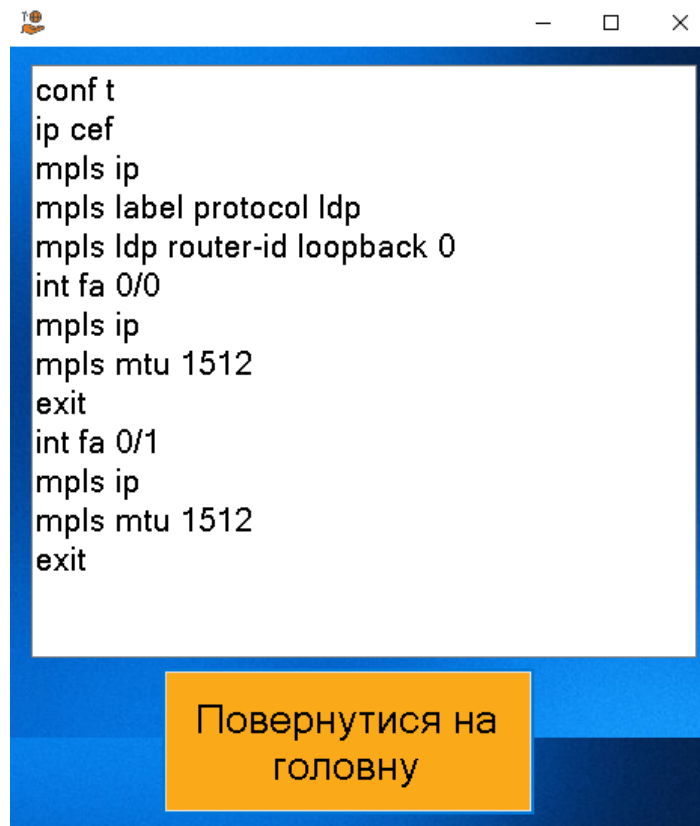


Рисунок 3.4 – Налаштування MPLS

В програмі в останньому етапі програма надає код для налаштування кожного роутера. В цьому програмі буде мати деякі переваги, так, як людина можливо неправильно визначила інтерфейс, чи зробила помилку в даних. Тому на даному етапі, важливо не помилитися.

Останнім етапом для налаштування користувачем мережі MPLS буде код для налаштування цієї мережі. Його потрібно буде вводити на кожному роутері. Тому достатньо його скопіювати та вставити для кожного роутера



```
conf t
ip cef
mpls ip
mpls label protocol ldp
mpls ldp router-id loopback 0
int fa 0/0
mpls ip
mpls mtu 1512
exit
int fa 0/1
mpls ip
mpls mtu 1512
exit
```

Повернутися на головну

Рисунок 3.5 – Налаштування MPLS

Цих кроків достатньо, щоб запустити мережу MPLS на роутерах та користуватися цією мережею.

3.2 Опис функціоналу додатка

Для того щоб отримати код для налаштування додатку, користувачу потрібно ввести в поля, котрі підписані з лівої частини від додатку, відповідні значення. Так як обрана схема буде нараховувати декілька мереж.

Наступним етапом користувач має ввести ір-адреси для роутера який зараз налаштовується.

Налаштування EIGRP

Налаштування кроку 1

IP інтерфейсу fa 0/0	192.168.1.1
Маска інтерфейсу fa 0/0	255.255.255.0
IP інтерфейсу fa 0/1	192.168.1.2
Маска інтерфейсу fa 0/0	255.255.255.0
Номер EIGRP мережі	100
Сусідня мережа №1	192.168.2.0
Зворотня маска мережі №1	0.0.0.255
Сусідня мережа №2	192.168.3.0
Зворотня маска мережі №2	0.0.0.255

Наступний етап

Рисунок 3.6 – Приклад заповнених полів

Налаштування EIGRP

Налаштування кроку 2

IP інтерфейсу fa 0/0	
Маска інтерфейсу fa 0/0	
IP інтерфейсу fa 0/1	
Маска інтерфейсу fa 0/0	
Номер EIGRP мережі	
Сусідня мережа №1	
Зворотня маска мережі №1	
Сусідня мережа №2	
Зворотня маска мережі №2	

Наступний етап

Рисунок 3.7 – Вигляд другого етапу

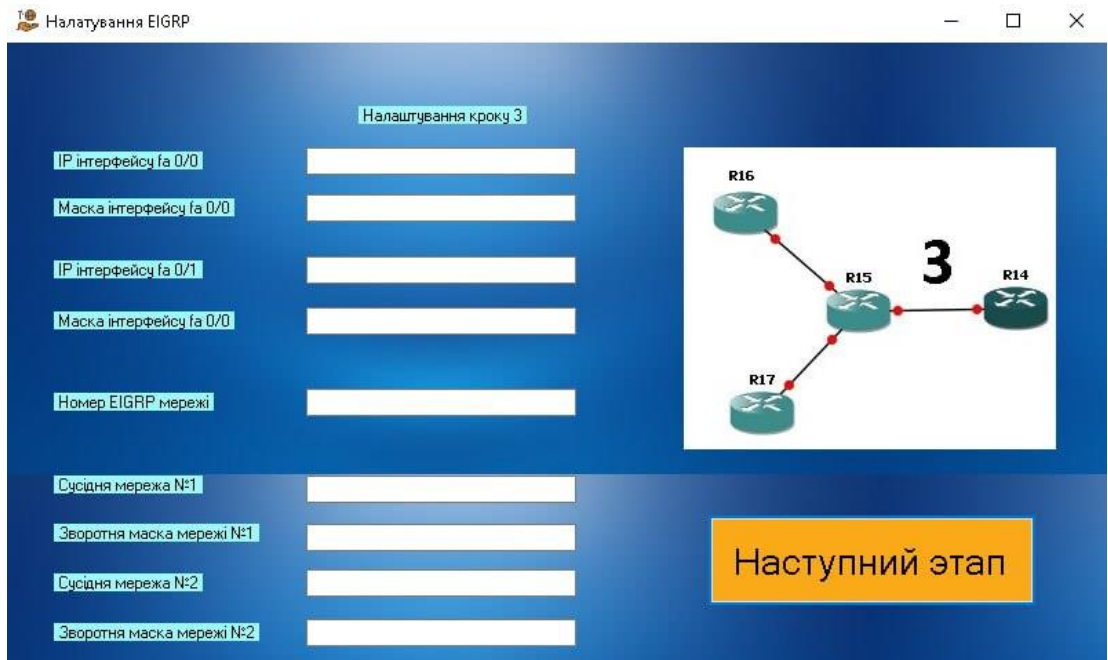


Рисунок 3.8 – Вигляд третього етапу

Після введення даних в поля, програма автоматично їх зберігає та очищує, готуючись до наступного кроку. На ілюстраціях 3.7 та 3.8 можна побачити, що кожен етап відрізнятиметься лише візуальним зображенням та написом, що інформує користувача про те, на якому етапі він знаходиться.

Коли код для кожного етапу буде написаний, користувач, натиснувши кнопку "Наступний етап", перейде до нового вікна. Там йому буде представлено код для налаштування роутерів. Цей код може бути відредагованим і скопійованим за потребою.

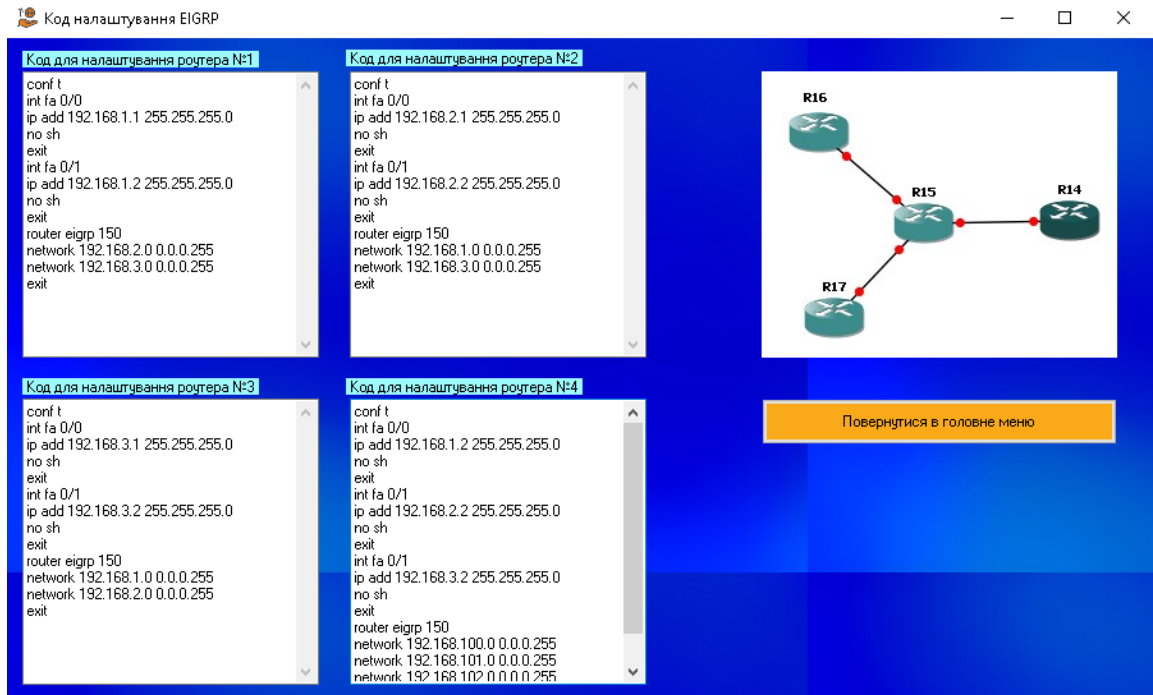


Рисунок 3.9 – Результат роботи додатку

Також на даному етапі, якщо роутери налаштовані користувач може повернутися до головного меню, для того, щоб продовжити налаштування мережі, точніше окремих частин мережі. Або користувач може мати можливість налаштувати MPLS для усіх роутерів. Важливо зазначити, що код для налаштування може підходити до будь-якого роутера, але якщо налаштуванні були інтерфейси так, як зазначено в коді. Якщо були використані інші інтерфейси, користувач має можливість змінити код, та скопіювати вірні налаштування.

3.3 Тестування додатку

Для того щоб перевірити мережу на налаштування, потрібно скористатися тестовою схемою. Для тесту була використана проста схема, для того, щоб продемонструвати правильну роботу додатку.

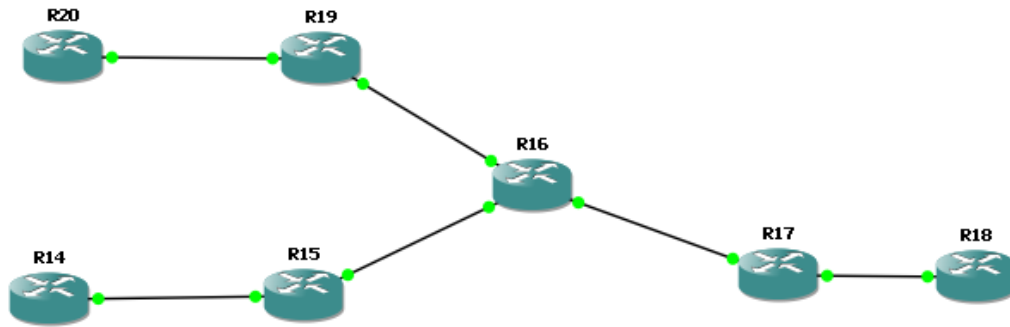


Рисунок 3.10 – Тестова мережа

Після вводу всіх підходящих для нашої мережі даних, переходимо до останнього етапу, де саме зв'являється можливість скопіювати код для налаштування роутерів

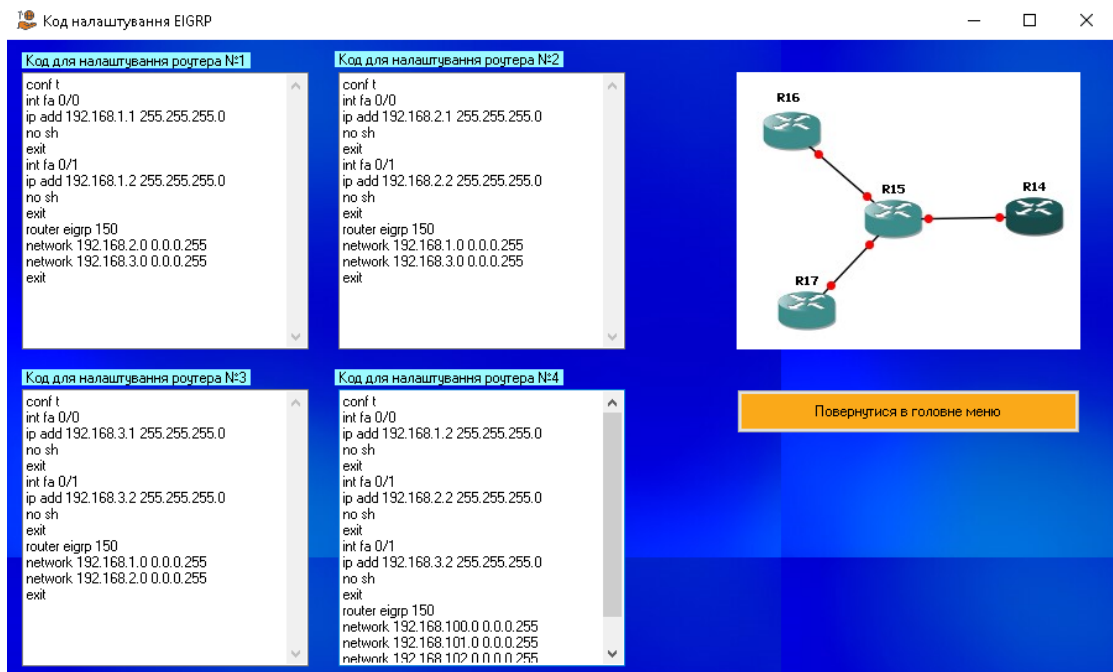


Рисунок 3.11 – Останній крок з кодом для роутерів

Налаштуємо кожен роутер та вводимо команди з додатку. Для прикладу на наступних скріншотах буде показаний тільки один роутер, так як принцип налаштування кожного роутеру буде однаковий.

```

R15
Connected to Dynamips VM "R15" (ID 14, type c3745) - Console port
Press ENTER to get the prompt.

R15#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R15(config)#int fa 0/0
R15(config-if)#ip add 192.168.10.2 255.255.255.0
R15(config-if)#no sh
R15(config-if)#exit
R15(config)#int fa 0/1
R15(config-if)#ip add 192.168.100.1 255.255.255.0
R15(config-if)#no sh
R15(config-if)#exit
R15(config)#router eigrp 200
R15(config-router)#network 192.168.10.0 0.0.0.255
R15(config-router)#network 192.168.100.0 0.0.0.255
R15(config-router)#exit
R15(config)#
*Mar 1 00:01:23.135: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:01:23.331: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar 1 00:01:24.135: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R15(config)#
*Mar 1 00:01:24.331: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R15(config)#
*Mar 1 00:02:32.119: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 200: Neighbor 192.168.100.2 (FastEthernet0/1) is up: new adjacency

```

Рисунок 3.12 – Налаштування IP та EIGRP

```

R15(config)#ip cef
R15(config)#mpls ip
R15(config)#mpls label protocol ldp
R15(config)#mpls ldp router-id loopback 0
R15(config)#int fa 0/0
R15(config-if)#mpls ip
R15(config-if)#mpls mtu 1512
R15(config-if)#exit
R15(config)#int fa 0/1
R15(config-if)#mpls ip
R15(config-if)#mpls mtu 1512
R15(config-if)#exit
R15(config)#
*Mar 1 00:06:49.343: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R15(config)#
*Mar 1 00:06:52.859: %LDP-5-NBRCHG: LDP Neighbor 192.168.101.1:0 (1) is UP
R15(config)#exit

```

Рисунок 3.13 – Налаштування технології MPLS

Далі вводимо команди для перевірки таблиці маршрутизації та наявності сусіда для роутера. Цей крок користувачу потрібен для того, щоб впевнитися, що налаштування проходить успішно. Якщо з'явиться помилка під час налаштування, то буде можливість одразу оперативно виправити налаштування роутерів.

```

R15#sh mpls forwarding-table
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched  interface
16     17         192.168.11.0/24  0         Fa0/1     192.168.100.2
17     18         192.168.102.0/24 0         Fa0/1     192.168.100.2
18     Pop tag    192.168.101.0/24 0         Fa0/1     192.168.100.2

```

Рисунок 3.14 – Результат команди sh mpls forwarding-table

```

R15#sh mpls ldp neighbor
Peer LDP Ident: 192.168.101.1:0; Local LDP Ident 192.168.100.1:0
TCP connection: 192.168.101.1.28422 - 192.168.100.1.646
State: Oper; Msgs sent/rcvd: 22/22; Downstream
Up time: 00:12:30
LDP discovery sources:
FastEthernet0/1, Src IP addr: 192.168.100.2
Addresses bound to peer LDP Ident:
192.168.100.2 192.168.101.1

```

Рисунок 3.15 – Результат команди sh mpls ldp neighbor

Аналіз таблиці маршрутизації, яку можна побачити на рисунку 3.14, дозволяє глибше зрозуміти процес передачі даних між роутерами. На прикладі, перший роутер із міткою 16 передає інформацію на другий роутер з міткою 17. У таблиці маршрутизації другого роутера, у другому рядку, ми бачимо локальну мітку 17, яку передав перший роутер, і вихідну мітку "Pop tag". Присутність мітки "Pop tag" свідчить про те, що роутер повинен зняти мітку під час пересилання пакета далі.

За допомогою такого детального аналізу таблиці маршрутизації можна виявити, що MPLS налаштована коректно. Це важливий аспект перевірки, оскільки правильна настройка MPLS забезпечує ефективне і надійне управління трафіком у мережі, гарантуючи оптимальний маршрут для кожного пакета даних.

ВИСНОВКИ

У результаті виконання роботи було досягнуто значних успіхів у розробці десктопного додатку для налаштування мереж з використанням технології багатопроTOCOLЬНОЇ комутації міток (MPLS) та різних протоколів маршрутизації. Робота сприяла поліпшенню процесів налаштування мережі, зниженню ризику помилок та підвищенню ефективності використання MPLS. Результати роботи додатку демонструють його практичну цінність і важливість для області мережевих технологій.

У ході виконання кваліфікаційної роботи магістра було виконано наступні завдання:

1. Здійснено аналіз проблематики, встановлено актуальність розроблення системи.
2. Проведено огляд існуючих методів маршрутизації, оцінено їхні переваги та недоліки.
3. Вибрано тестові схеми, які були застосовані у процесі розробки.
4. Визначено технології, використані для створення програмного додатку.
5. Розроблено додаток для автоматизації генерації коду налаштування маршрутизації.
6. Здійснено тестування системи та перевірено її функціональність.
7. Проведено аналіз отриманого коду і сформульовано висновки стосовно ефективності додатку.

За результатами роботи десктопного додатку, можна зробити висновок, що додаток забезпечує ефективне налаштування мереж за обраним протоколом маршрутизації та генерацію коду для налаштування технології MPLS, сприяючи оптимізації процесу налаштування та зменшенню ризику людських помилок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Коробейнікова Т.І., Загарченко С.М. Навчальний посібник «Комп'ютерні мережі». – Львів: Львівська політехніка, 2022. – 228 с.
2. Segment Routing with the MPLS Data Plane [Електронний ресурс] – Режим доступу <https://rap.mirror.cyberbits.eu/rfcs/rfc8660.pdf>
3. Recent trends in MPLS networks: technologies, applications and challenges [Електронний ресурс] – Режим доступу <https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-com.2018.6129>
4. Ю. В. Волосюк. Комп'ютерні мережі : курс лекцій – Миколаїв: МНАУ, 2019. – 203 с.
5. Building reliable MPLS networks using a path protection mechanism [Електронний ресурс] – Режим доступу <https://ieeexplore.ieee.org/abstract/document/989779>
6. Definitive MPLS Network Designs – No 3 (95). – 2020. – P. 59-66.
7. Mobile Network Operator A Complete Guide. Gerardus Blokdyk. Edition Paperback, 2021. - 306 p.
8. Multiprotocol Label Switching (MPLS) [Електронний ресурс] – Режим доступу [https://www.techtarget.com/searchnetworking/definition/Multiprotocol-Label-Switching-MPLS#:~:text=In%201997%2C%20the%20Internet%20Engineering,asynchronous%20transfer%20mode%20\(ATM\).](https://www.techtarget.com/searchnetworking/definition/Multiprotocol-Label-Switching-MPLS#:~:text=In%201997%2C%20the%20Internet%20Engineering,asynchronous%20transfer%20mode%20(ATM).)
9. Traffic Analysis of MPLS and Non MPLS Network including MPLS Signaling Protocols and Traffic Distribution in OSPF and MPLS [Електронний ресурс] – Режим доступу <https://ieeexplore.ieee.org/abstract/document/4579892>
10. A constrained multipath traffic engineering scheme for MPLS networks

- [Электронный ресурс] – Режим доступа
<https://ieeexplore.ieee.org/abstract/document/997280>
11. MPLS terms and concepts [Электронный ресурс] – Режим доступа
<https://docs.vmware.com/en/VMware-Smart-Assurance/10.1.0/mpls-manager-user-guide-101/GUID-8EB1D677-B262-475F-9C1B-8D2D9826CC0D.html>
12. Protection performance components in MPLS networks [Электронный ресурс] – Режим доступа
<https://www.sciencedirect.com/science/article/abs/pii/S014036640400115>
13. Multi-layer MPLS network design: The impact of statistical multiplexing [Электронный ресурс] – Режим доступа
<https://www.sciencedirect.com/science/article/abs/pii/S138912860800023>
14. Protection And Restoration In MPLS Networks [Электронный ресурс] – Режим доступа <http://www.olddog.co.uk/mplsprotwp2.pdf>
15. P-Rex: fast verification of MPLS networks with multiple link failures [Электронный ресурс] – Режим доступа
<https://dl.acm.org/doi/abs/10.1145/3281411.3281432>
16. MPLS and traffic engineering in IP networks [Электронный ресурс] – Режим доступа <https://ieeexplore.ieee.org/abstract/document/809383>
17. A Layered MPLS Network Architecture [Электронный ресурс] – Режим доступа <https://ieeexplore.ieee.org/abstract/document/5601465>
18. Encapsulation Methods for Transport of Ethernet over MPLS Networks [Электронный ресурс] – Режим доступа <https://www.rfc-editor.org/rfc/rfc4448>
19. A novel path protection scheme for MPLS networks using multi-path routing [Электронный ресурс] – Режим доступа
<https://www.sciencedirect.com/science/article/abs/pii/S138912860900036>
- X
20. Packet Loss and Delay Measurement for MPLS Networks [Электронный

- ресурс] – Режим доступа <https://www.rfc-editor.org/rfc/rfc6374>
21. Advanced MPLS Design and Implementation , pp. 629-635. Copernicus GmbH, doi:10.5194/isprs-archivesxlii-2-629-2019
22. Traffic engineering with MPLS in the Internet [Электронный ресурс] – Режим доступа <https://ieeexplore.ieee.org/abstract/document/826369>.
23. Topological design of MPLS networks [Электронный ресурс] – Режим доступа <https://ieeexplore.ieee.org/abstract/document/965071>
24. Types of Routing Protocols. Cisco Networking Academy's Introduction to Routing Dynamically | Cisco Press [Электронный ресурс] – Режим доступа <https://www.ciscopress.com/articles/article.asp?p=2180210&seqNum=7>

ДОДАТОК

Form2

```
namespace Diplom_AB
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }
        InfoText IT_Form;

        private void label6_Click(object sender, EventArgs e)
        {

        }
        InfoText InfoT;
        private void button1_Click(object sender, EventArgs e)
        {
            InfoT = new InfoText();
            InfoT.ShowDialog(this);
        }

        private int clickCount = 0;

        private void button2_Click(object sender, EventArgs e)
        {
            clickCount++;

            string ipfirstfa = Ip_first_fa.Text;
            string maskfistfa = Mask_first_fa.Text;
            string ipsecondfa = Ip_second_fa.Text;
            string masksecondfa = Mask_second_fa.Text;
            string firstfriendnet = First_friend_net.Text;
            string firstfriendmask = First_friend_mask.Text;
            string secondfriendnet = Second_friend_net.Text;
```

```

string secondfriendmask = Second_friend_mask.Text;
string eigrpnumb = Eigrpnumb.Text;

if (clickCount == 1)
{
    // Перше зображення
    pictureBox1.Image
Image.FromFile("/Program/Diplom_AB/Diplom_AB/Resources/Step_1.2.jpg");
    ipfirstfa = Ip_first_fa.Text;
    Ip_first_fa.Text = "";
    maskfistfa = Mask_first_fa.Text;
    Mask_first_fa.Text = "";
    ipsecondfa = Ip_second_fa.Text;
    Ip_second_fa.Text = "";
    masksecondfa = Mask_second_fa.Text;
    Mask_second_fa.Text = "";
    firstfriendnet = First_friend_net.Text;
    First_friend_net.Text = "";
    firstfriendmask = First_friend_mask.Text;
    First_friend_mask.Text = "";
    secondfriendnet = Second_friend_net.Text;
    Second_friend_net.Text = "";
    secondfriendmask = Second_friend_mask.Text;
    Second_friend_mask.Text = "";
    eigrpnumb = Eigrpnumb.Text;
    Eigrpnumb.Text = "";
    labelMainText.Text = "Налаштування кроку 2";
}
else if (clickCount == 2)
{
    // Друге зображення
    pictureBox1.Image
Image.FromFile("/Program/Diplom_AB/Diplom_AB/Resources/Step_1.3.jpg");
    ipfirstfa = Ip_first_fa.Text;
    Ip_first_fa.Text = "";
    maskfistfa = Mask_first_fa.Text;
    Mask_first_fa.Text = "";

```

```
ipsecondfa = Ip_second_fa.Text;
Ip_second_fa.Text = "";
masksecondfa = Mask_second_fa.Text;
Mask_second_fa.Text = "";
firstfriendnet = First_friend_net.Text;
First_friend_net.Text = "";
firstfriendmask = First_friend_mask.Text;
First_friend_mask.Text = "";
secondfriendnet = Second_friend_net.Text;
Second_friend_net.Text = "";
secondfriendmask = Second_friend_mask.Text;
Second_friend_mask.Text = "";
eigrpnumb = Eigrpnumb.Text;
Eigrpnumb.Text = "";
labelMainText.Text = "Налаштування кроку 3";

}
else if (clickCount == 3)
{
ipsecondfa = Ip_second_fa.Text;
Ip_second_fa.Text = "";
masksecondfa = Mask_second_fa.Text;
Mask_second_fa.Text = "";
ipsecondfa = Ip_second_fa.Text;
Ip_second_fa.Text = "";
masksecondfa = Mask_second_fa.Text;
Mask_second_fa.Text = "";
firstfriendnet = First_friend_net.Text;
First_friend_net.Text = "";
firstfriendmask = First_friend_mask.Text;
First_friend_mask.Text = "";
secondfriendnet = Second_friend_net.Text;
Second_friend_net.Text = "";
secondfriendmask = Second_friend_mask.Text;
Second_friend_mask.Text = "";
eigrpnumb = Eigrpnumb.Text;
Eigrpnumb.Text = "";
```

```

        clickCount = 0;
        IT_Form = new InfoText();
        IT_Form.Show();
    }
}
private void clearFullArea (object sender, EventArgs e)
{
    Ip_first_fa.Text = "";
    Mask_first_fa.Text = "";
    Ip_second_fa.Text = "";
    Mask_second_fa.Text = "";
    First_friend_net.Text = "";
    First_friend_mask.Text = "";
    Second_friend_net.Text = "";
    Second_friend_mask.Text = "";
    Eigrpnumb.Text = "";
}
}

```

Form3

namespace Diplom_AB

```

{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void label6_Click(object sender, EventArgs e)
        {

        }
        InfoText InfoT;
        private void button1_Click(object sender, EventArgs e)
        {

```

```

InfoT = new InfoText();
InfoT.ShowDialog(this);
}

private void button2_Click(object sender, EventArgs e)
{
    clickCount == 3
    string ipfirstfa = Ip_first_fa.Text;
    string maskfistfa = Mask_first_fa.Text;
    string ipsecondfa = Ip_second_fa.Text;
    string masksecondfa = Mask_second_fa.Text;
    string firstfriendnet = First_friend_net.Text;
    string firstfriendmask = First_friend_mask.Text;
    string secondfriendnet = Second_friend_net.Text;
    string secondfriendmask = Second_friend_mask.Text;
    string eigrpnumb = Eigrpnumb.Text;
    Random rnd = new Random();
    int rndip1 = rnd.Next(0,254);
    int rndip2 = rnd.Next(0,254);
    int eigrpnet1 = rnd.Next(0, 20) * 10;
    if (maskfistfa == "")
    {
        maskfistfa = "254.254.254.0";
    }
    if (masksecondfa == "")
    {
        masksecondfa = "254.254.254.0";
    }
    if (ipfirstfa == "" || ipsecondfa == "" || masksecondfa == "" || maskfistfa == "" || firstfriendmask == ""
|| firstfriendnet == "" || secondfriendnet == "" || secondfriendmask == "")
    {
        Text_net.Text = "conf t" + Environment.NewLine + "int fa 0/0" + Environment.NewLine +
        "ip add 193.167." + rndip1 + ".1 254.254.254.0" + Environment.NewLine + "no sh" +
Environment.NewLine + "exit"

```

```

+ Environment.NewLine + "int fa 0/1" + Environment.NewLine +
"ip add 193.167." + rndip2 + ".1 254.254.254.0" + Environment.NewLine + "no sh" +
Environment.NewLine + "exit"
+ Environment.NewLine +

"router eigrp " + eigrpnet1 + Environment.NewLine +
"network " + "193.167." + rndip1 + ".0" + " 0.0.0.254" + Environment.NewLine +
"network " + "193.167." + rndip2 + ".0" + " 0.0.0.254" + Environment.NewLine + "exit"
;
}
else if (clickCount == 1)
{
Text_net.Text = "conf t" + Environment.NewLine +
"int fa 0/0" + Environment.NewLine +
"ip add " + ipfirstfa + " " + maskfistfa + Environment.NewLine + "no sh" +
Environment.NewLine + "exit" + Environment.NewLine
+ "int fa 0/1" + Environment.NewLine +
"ip add " + ipsecondfa + " " + masksecondfa + Environment.NewLine + "no sh" +
Environment.NewLine + "exit" +
Environment.NewLine +

"router eigrp " + eigrpnumb + Environment.NewLine +
"network " + firstfriendnet + " " + firstfriendmask + Environment.NewLine +
"network " + secondfriendnet + " " + secondfriendmask + Environment.NewLine + "exit"
;
}
else if (clickCount == 2)
{
Text_net.Text = "conf t" + Environment.NewLine +
"int fa 0/0" + Environment.NewLine +
"ip add " + ipfirstfa + " " + maskfistfa + Environment.NewLine + "no sh" +
Environment.NewLine + "exit" + Environment.NewLine
+ "int fa 0/1" + Environment.NewLine +
"ip add " + ipsecondfa + " " + masksecondfa + Environment.NewLine + "no sh" +
Environment.NewLine + "exit" +
Environment.NewLine +

```

```

"router eigrp " + eigrpnumb + Environment.NewLine +
"network " + firstfriendnet + " " + firstfriendmask + Environment.NewLine +
"network " + secondfriendnet + " " + secondfriendmask + Environment.NewLine + "exit"
;
}
else if (clickCount == 3)
{
Text_net.Text = "conf t" + Environment.NewLine +
"int fa 0/0" + Environment.NewLine +
"ip add " + ipfirstfa + " " + maskfistfa + Environment.NewLine + "no sh" +
Environment.NewLine + "exit" + Environment.NewLine
+ "int fa 0/1" + Environment.NewLine +
"ip add " + ipsecondfa + " " + masksecondfa + Environment.NewLine + "no sh" +
Environment.NewLine + "exit" +
Environment.NewLine +

"router eigrp " + eigrpnumb + Environment.NewLine +
"network " + firstfriendnet + " " + firstfriendmask + Environment.NewLine +
"network " + secondfriendnet + " " + secondfriendmask + Environment.NewLine + "exit"
;
}
else if (clickCount == 4)
{
Text_net.Text = "conf t" + Environment.NewLine +
"int fa 0/0" + Environment.NewLine +
"ip add " + ipfirstfa + " " + maskfistfa + Environment.NewLine + "no sh" +
Environment.NewLine + "exit" + Environment.NewLine
+ "int fa 0/1" + Environment.NewLine +
"ip add " + ipsecondfa + " " + masksecondfa + Environment.NewLine + "no sh" +
Environment.NewLine + "exit" +
Environment.NewLine +

"int fa 1/1" + Environment.NewLine +
"ip add " + ipsecondfa + " " + masksecondfa + Environment.NewLine + "no sh" +
Environment.NewLine + "exit" +
Environment.NewLine +

"router eigrp " + eigrpnumb + Environment.NewLine +

```



```
"network " + firstfriendnet + " " + firstfriendmask + Environment.NewLine +  
"network " + secondfriendnet + " " + secondfriendmask + Environment.NewLine +  
    "network " + thirdfriendnet + " " + thirdfriendmask + Environment.NewLine + "exit"  
;  
}  
}  
}
```