

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
Національний університет «Запорізька політехніка»

Г.Г. Киричек

**Конспект лекцій з дисципліни**  
**«Методи оптимізації комп'ютерних мереж»**  
для студентів спеціальності  
123 Комп'ютерна інженерія, усіх форм навчання

Конспект лекцій з дисципліни «Методи оптимізації комп'ютерних мереж» для студентів спеціальності 123 Комп'ютерна інженерія, усіх форм навчання / Укл. Г.Г Киричек. – Запоріжжя: Національний університет «Запорізька політехніка», 2025. – 38 с.

Укладачі:

Г.Г. Киричек, доцент, к.т.н.

Рецензент:

М.Ю. Тягунова, доцент, к.т.н.

Відповідальний за випуск:

Г.Г. Киричек, доцент, к.т.н.

Затверджено  
на засіданні кафедри КСМ  
Протокол № 8 від 24.03.2025

Рекомендовано до видання  
НМК КНТ  
Протокол № 7 від 24.03.2025

## ЗМІСТ

|     |                                                          |    |
|-----|----------------------------------------------------------|----|
| 1   | Методи досягнення оптимізації комп'ютерних мереж.....    | 4  |
| 1.1 | Оновлення обладнання .....                               | 4  |
| 1.2 | Покращення конфігурації мережі.....                      | 6  |
| 1.3 | Оптимізація за допомогою математичних методів.....       | 9  |
| 2   | Впровадження та конфігурація OSPF.....                   | 12 |
| 2.1 | Терміни та визначення. Адміністративна відстань.....     | 12 |
| 2.2 | Обмін LSDB між сусідами.....                             | 13 |
| 2.3 | Розрахунок найкращих маршрутів із SPF.....               | 15 |
| 3   | Оптимізація маршрутів між мережами VLAN .....            | 17 |
| 3.1 | Методи маршрутизації між VLAN .....                      | 17 |
| 3.2 | Маршрутизація за допомогою SVI комутаторів 3 рівня.....  | 19 |
| 3.3 | VLAN з маршрутизованими портами комутатора рівня 3 ..... | 21 |
| 3.4 | Впровадження 3-го рівня EtherChannels .....              | 22 |
| 4   | Enhanced interior gateway routing protocol .....         | 24 |
| 4.1 | Особливості та функції EIGRP.....                        | 24 |
| 4.2 | Налаштування EIGRP .....                                 | 29 |
| 4.3 | Балансування навантаження.....                           | 33 |
|     | Рекомендована література .....                           | 38 |

## 1 МЕТОДИ ДОСЯГНЕННЯ ОПТИМІЗАЦІЇ КОМП'ЮТЕРНИХ МЕРЕЖ

**Мета** – розглянути та проаналізувати методи досягнення оптимізації комп'ютерних мереж, які застосовуються при реалізації та впровадженні сучасних транспортних магістральних мереж, а також дослідити, як реалізується кожен з цих методів.

### 1.1 Оновлення обладнання

Ключові методи досягнення оптимізації комп'ютерних мереж передбачають підвищення продуктивності, ефективності та надійності мереж передачі даних. До цих методів відносимо оновлення обладнання: оновлення мережевих пристроїв; заміна (оновлення) кабелів та використання балансувальників навантаження.

Маємо зображення мережі, яке містить бездротові точки доступу, розумні комутатори, оптоволоконні кабелі високопродуктивні маршрутизатори та встановлений у стійку сервер (рис.1.1).



Рисунок 1.1 – Візуальне зображення оновленої мережі

Ці оновлення забезпечують покращену продуктивність, вищу швидкість і підвищену надійність мережевої інфраструктури. Оновлення застарілих однодіапазонних маршрутизаторів до дводіапазонних або тридіапазонних маршрутизаторів із підтримкою

Wi-Fi 6 або Wi-Fi 6e для забезпечення вищої швидкості, меншої затримки та кращого керування кількома пристроями. Оновлення базових некерованих комутаторів до розумних або керованих комутаторів з портами Gigabit або 10-Gigabit Ethernet. Оновлення базового WAP із застарілими стандартами Wi-Fi до Wi-Fi 6 WAP з можливостями сітчастої мережі. Оновлення CAT5 або кабелів нижчої категорії Ethernet до CAT6A, CAT7 або волоконно-оптичних кабелів. Оновлення базових брандмауерів до міжмережевих екранів нового покоління (NGFW) із системами запобігання вторгненням (IPS). Оновлення основних засобів діагностики до комплексних платформ моніторингу мережі, таких як SolarWinds або PRTG. Оновлення автономних серверів із обмеженою обчислювальною потужністю до серверів, що монтуються в стійку, із накопичувачем SSD і підтримкою віртуалізації. Оновлення базових налаштувань до використання резервних блоків (RPSU) і джерел безперебійного живлення (UPS).

Щоб візуалізувати оновлену інфраструктуру, доповнимо цю інформацію прикладом схеми розподіленої мережі (рис.1.2).

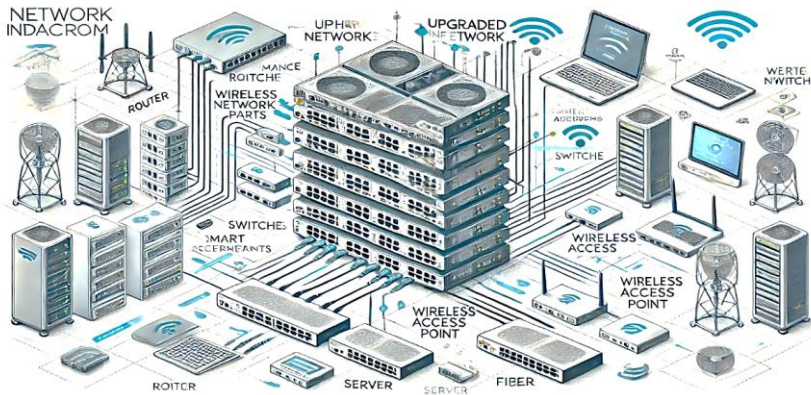


Рисунок 1.2 – Діаграма мережі

Отримали детальну діаграму мережі, яка ілюструє оновлену інфраструктуру та демонструє сучасні компоненти, такі як високопродуктивні маршрутизатори для покращеного керування пропускнуною здатністю та пристроями; інтелектуальні комутатори, що підтримують VLAN і гігабітні/10-гігабітні з'єднання; бездротові точки

доступу з підтримкою сітчастої мережі та технології Wi-Fi 6; оптоволоконні з'єднання для високошвидкісної передачі даних із запобіганням виникнення перешкод та встановлений у стійку сервер, з метою ефективної обробки даних і централізованого керування. Ця схема відображає з'єднання між пристроями, демонструючи потік даних і їх роль у підвищенні продуктивності та надійності мережі.

Маршрутизатор діє як центральний пристрій, під'єднуючи мережу до Інтернету. Комутатори виконують функції посередників, керуючи кількома кабельними з'єднаннями. Вони організують трафік віртуальних локальних мереж (VLAN), підвищуючи безпеку та зменшуючи перевантаження в мережі в цілому. Бездротові точки доступу (WAP) розширюють відстань, при підключенні пристроїв, за допомогою підтримки сітчастої мережі. Сервери керують хостингом програм, централізованим зберіганням даних і критично важливими службами. Оптоволоконні з'єднання забезпечують високу швидкість передачі даних. Потік даних позначається лініями, які вказують на спрямований або двонаправлений зв'язок.

Цей структурований макет зменшує вузькі місця, підвищує надійність і підтримує сучасні технології, такі як IoT і хмарні сервіси.

## 1.2 Покращення конфігурації мережі

Далі, щоб підкреслити роль та взаємодію кожного пристрою, розглянемо проєкт оновленої мережевої інфраструктури, який демонструє структуровану схему із з'єднаннями, організованими за допомогою оптоволоконних кабелів, кабелів Ethernet та визначеним потоком даних. Покращуючи конфігурацію мережі маємо (рис.1.3):

- реалізацію пріоритезації трафіку або якості обслуговування (QoS) для визначення пріоритету критичних програм або пристроїв;
- розділення мережі на менші, керовані підмережі, щоб зменшити перевантаження та підвищити безпеку;
- використання віртуальних локальних мереж (VLAN) для розділення трафіку, кращого ним керування і зменшення перешкод.

Маємо візуалізацію пріоритезації трафіку (якість обслуговування) у мережі, яка ілюструє схему організованих потоків даних для критичних програм, потокового відео та фонових процесів.

На рисунку 1.3 зображено мережеве середовище із пріоритетністю трафіку з точки зору якості обслуговування (QoS).



Рисунок 1.3 – Пріоритезація трафіку

Критичні програми мають (високий пріоритет), який представлено у вигляді широкої, швидкої смуги із яскравими кольорами. Виділено такі пристрої, як медичні монітори, VoIP телефони або фінансові системи. Пакети даних для цих програм швидко переміщуються мережею. Потокове відео має (середній пріоритет), який зображено в дещо вужчій смузі. Потоки даних відеоплеєрів і розважальних платформ, які передаються в помірному темпі. Фонові процеси мають (низький пріоритет), який показано у вузькій, повільнішій смузі. Ці процеси відбуваються повільніше, але зберігають постійний перебіг. Допоміжними елементами є маршрутизатори та комутатори - центральні вузли, які спрямовують трафік на основі призначених пріоритетів.

Процес оптимізації лежить в основі процесу передачі даних, оскільки потрібно проєктувати більш ефективні і продуктивні за пропускну здатністю та іншими параметрами розподілені комп'ютерні мережі. Ефективність методів оптимізації, які дозволяють здійснити вибір найкращого маршруту, пов'язана із протоколами, що використовують алгоритми маршрутизації, а також досягнення в

математичних обчисленнях, які спираються на обґрунтовані логічні схеми і застосовують програмні методи.

Схема (рис.1.4) візуалізує розділення мережі на менші підмережі, демонструючи взаємозв'язок між основною мережею та підмережами, такими як: офісні мережі; гостьові мережі; пристрої Інтернету речей; обладнання центрів обробки даних із брандмауерами та іншими пристроями, які керують потоком даних.

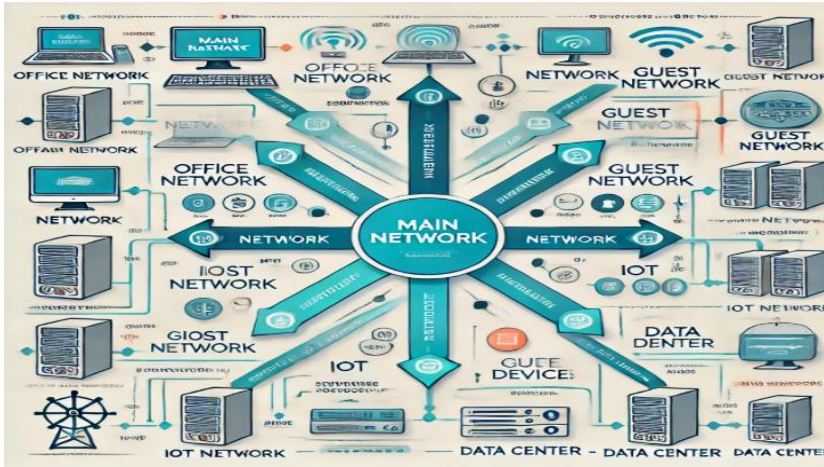


Рисунок 1.4 – Схема розділення мережі

Методи оптимізації ефективно застосовуються при проектуванні і керуванні потоками даних в мережі. При проектуванні мереж вибираються найкращі моделі і способи, включаючи вибір протоколів маршрутизації та схем їх налаштування. Вирішення задач оптимізації з використанням технічних та програмних засобів, залежить від самої задачі оптимізації. Першим є завдання реалізації алгоритмів прийняття оптимальних рішень при проектуванні і керуванні процесом передачі та об'ємом трафіку. Від правильного формулювання задачі в значній мірі залежить застосування алгоритмів та методів оптимізації.

### 1.3 Оптимізація за допомогою математичних методів

Оптимізація комп'ютерних мереж за допомогою математичних методів дозволяє покращити продуктивність, надійність і ефективність мережі, мінімізуючи витрати. Ось основні математичні методи, які застосовуються для цього:

- лінійне програмування (LP) - математичний метод, який у контексті комп'ютерних мереж дозволяє оптимізувати використання ресурсів (пропускна здатність, затримки, енергоспоживання тощо);
- цілочислове програмування (IP) - коли змінні повинні бути цілочисленними (кількість серверів, каналів зв'язку). Прикладом є оптимізація розташування точок доступу Wi-Fi у приміщенні;
- теорія черг (Queueing Theory) - аналіз і моделювання параметрів продуктивності мереж (пропускна здатність, втрата пакетів, середній час очікування), як систем обслуговування (M/M/1, M/G/1). Приклад - оцінка впливу додаткового сервера на час відповіді мережі;
- теорія графів (Graph Theory) - проектування топології мережі та оптимізація маршрутів (вузли - це пристрої, а ребра - лінії зв'язку). Використання алгоритмів Дейкстри) або Форда-Фалкерсона;
- евристичні та метаевристичні алгоритми - коли точна оптимізація занадто витратна для великих мереж. Використання генетичних алгоритмів, пошук наближених рішень. Прикладом є оптимізація розподілу навантаження між серверами;
- теорія ігор (Game Theory) - оптимізація поведінки мережі за участі багатьох агентів (користувачів або пристроїв). Моделювання взаємодії як гри зі стратегіями і винагородами. Прикладом є управління перевантаженнями, з метою справедливого розділення пропускної здатності;
- оптимізація за допомогою машинного навчання - прогнозування трафіку або динамічне налаштування мереж. Приклад - адаптивна маршрутизація залежно від прогнозованих перевантажень;
- стохастичне моделювання (Stochastic Modeling) - робота з невизначеністю в трафіку або відмовах обладнання. Представлення змінних у вигляді ймовірнісних розподілів (розподіл Пуассона при отриманні пакетів). Прикладом є проектування відмовостійких систем із використанням надлишковості;

– динамічне програмування (Dynamic Programming, DP) - послідовне прийняття рішень для складних завдань. Розбиття проблеми на підзадачі з подальшим їх розв'язанням. Прикладом є оптимізація багатокрокової маршрутизації в бездротових мережах;

– багатокритеріальна оптимізація (Multi-Objective Optimization) - вирішення конфліктів (продуктивність проти витрат). Пошук Парето-оптимального набору рішень. Прикладом є мінімізація енергоспоживання при максимізації пропускної здатності.

Для реалізації цих методів на практиці можуть застосовуватись або інструменти: MATLAB, Python (бібліотеки SciPy, NetworkX, PuLP), оптимізаційні пакети (Gurobi, CPLEX), або фреймворки: використання SDN (програмовані мережі) для гнучкої оптимізації.

Для прикладу розглянемо метод лінійного програмування (ЛП) для оптимізації комп'ютерних мереж. Ключовими елементами ЛП є:

– змінні, що описують параметри мережі, які потрібно оптимізувати (пропускна здатність каналу зв'язку; кількість трафіку, що проходить через певний вузол;

– цільова функція - функція, яку потрібно мінімізувати або максимізувати (мінімізація затримки трафіку; мінімізація витрат на передачу даних; максимізація загальної пропускної здатності мережі);

– обмеження, що задають правила або умови, які потрібно виконувати (пропускна здатність каналу не повинна перевищувати його фізичну межу; трафік повинен бути спрямований через задані вузли).

Наведемо загальний вигляд задачі лінійного програмування. Перед інженером стоїть завдання мінімізувати/максимізувати функцію:  $c_1x_1 + c_2x_2 + \dots + c_nx_n$ .

$$\text{За умов: } \begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2, \\ x_i \geq 0, \forall i. \end{cases}$$

–  $x_1, x_2, x_3, \dots, x_n$  – змінні;

–  $c_1, c_2, c_3, \dots, c_n$  – коефіцієнти цільової функції;

–  $a_{ij}$  – коефіцієнти обмежень;

–  $b_1, b_2, b_3, \dots, b_n$  – значення правих частин обмежень.

І далі маємо приклад застосування ЛП у мережах при маршрутизації трафіку. Уявімо, що потрібно мінімізувати загальну затримку передачі даних між вузлами в мережі. Змінними є:  $x_{ij}$  – об'єм трафіку, який передається через канал між вузлами  $i$  та  $j$ . Цільова функція для мінімізації сумарної затримки, використовується, як умова за якою потрібно мінімізувати  $\sum_{(i,j)} d_{ij} x_{ij}$ , де  $d_{ij}$  є затримкою на каналі між вузлами  $i$  та  $j$ . При цьому маємо обмеження:

- сумарний об'єм трафіку, що надходить на кінцевий вузол, дорівнює сумі трафіку, який його покидає (закон збереження потоку):  $\sum_{(i)} x_{ij} = \sum_{(k)} x_{jk}, \forall j$ ;

- пропускна здатність каналу не перевищує ту межу, яка для нього встановлена:  $x_{ij} \leq C_{ij}, \forall (i,j)$ , при цьому  $C_{ij}$  є максимальною пропускною здатністю каналу зв'язку;

- зворотні потоки заборонені:  $x_{ij} \geq 0, \forall (i,j)$ .

Для розв'язання задачі ЛП можна скористатися наведеними алгоритмами:

- геометрична інтерпретація цільової функції і обмежень. При цьому кінцева ціль може розглядатися як функція, визначена в  $n$ -мірному просторі змінних  $x_j$ , де  $j \in \{1, \dots, n\}$ ;

- симплекс-метод, який є класичним і ефективним методом для точного розв'язання задач ЛП. Має ітераційний характер, де обчислювальні кроки однакового (того ж самого) типу повторюються у певній послідовності поки не отримано оптимальний план вирішення завдання або з'ясовано, що його не існує. Транспортна модель використовується для мінімізації шляху передачі трафіку між кінцевими вузлами. Специфічна структура транспортної задачі надає можливість модифікувати симплекс-метод таким чином, що обчислювальні процедури стають більш ефективними;

- метод внутрішньої точки – використовується для великих задач ЛП із багатьма змінними;

- реалізація в програмному забезпеченні: Python: бібліотека PuLP, SciPy.optimize, CVXPY; MATLAB: функція linprog; Gurobi/CPLEX: інструменти для комерційного використання.

Результатами оптимізації, для наведеного прикладу вирішення завдання з маршрутизації, є: отримані оптимальні маршрути для

передачі даних; зменшення затримок у мережі; раціональне використання ресурсів, наприклад, пропускнуої здатності каналів.

## 2 ВПРОВАДЖЕННЯ ТА КОНФІГУРАЦІЯ OSPF

### 2.1 Терміни та визначення. Адміністративна відстань

Протоколи стану зв'язку створюють IP-маршрути за допомогою наступних етапів:

- маршрутизатори формують інформацію про мережу: маршрутизатори; посилання; IP-адреси; статус тощо;
- маршрутизатори передають інформацію один одному, тому всі вони мають однакову інформацію;
- кожен маршрутизатор розраховує маршрути до всіх підмереж, але з позиції кожного маршрутизатора.

Open Shortest Path First (OSPF), організовує пересилання інформації про топологію за допомогою LSA (link-state advertisements) та бази даних про стан зв'язків (link-state database (LSDB)). Кожен LSA є структурою даних з деякою конкретною інформацією про топологію мережі; а LSDB - це просто набір усіх LSA, відомих маршрутизатору.

На рисунку 2.1 наведено загальний процес флудингу (алгоритм маршрутизації, в якому кожен наступний вхідний пакет надсилається через кожне вихідне посилання, крім того, на яке він надійшов), коли маршрутизатор R8 створює та запускає процес флудингу LSA. LSA для маршрутизатора R8 формує сам маршрутизатор, маючи інформацію про наявність у нього підмережі 172.16.3.0/24.

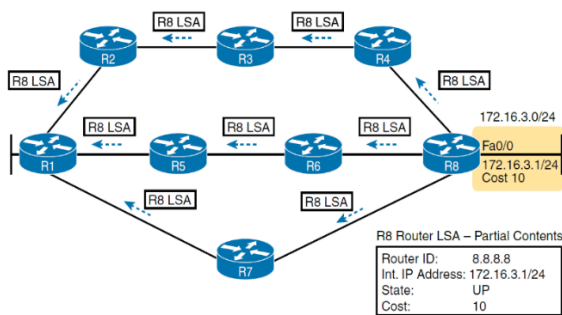


Рисунок 2.1 – Передача LSA

Щоб побудувати маршрути, маршрутизатори мають виконувати математичні обчислення. Всі протоколи стану зв'язку використовують для обробки LSDB тип математичних алгоритмів, які мають назву алгоритм «Dijkstra Shortest Path First» (SPF). Цей алгоритм математично аналізує LSDB і створює маршрути, які локальний маршрутизатор повинен додати до таблиці IP-маршрутизації. Усі ці маршрути мають номер і маску підмережі (мережі), вихідний інтерфейс і IP-адресу наступного маршрутизатора.

Маємо три основні етапи виконання роботи з обміну LSA та з розрахунку маршрутів маршрутизаторами:

- стати сусідами: відносини між двома маршрутизаторами, які підключаються до одного каналу передачі даних, створені так, щоб сусідні маршрутизатори мали можливість обмінюватися LSDB;
- обмін базами даних: процес розсилки LSA сусідам, щоб усі маршрутизатори вивчали однакові LSA;
- додавання найкращих маршрутів: процес, коли кожен маршрутизатор, незалежно, запускає алгоритм SPF на своїй LSDB, обчислює найкращі маршрути і додає їх до таблиці маршрутизації.

## 2.2 Обмін LSDB між сусідами

Коли два маршрутизатори вирішують обмінятися базами даних, вони не просто надсилають вміст усієї бази даних. По-перше, вони повідомляють один одному список LSA у відповідних базах даних. Зауважте, що не всі деталі LSA, а лише список. При цьому, кожен маршрутизатор може перевірити, які LSA він вже має, а потім зробити запит у іншого маршрутизатора лише тих LSA, які йому ще не відомі. Наприклад, R1 може надіслати R2 контрольний список із переліком 10 LSA. Потім R2 перевіряє свою LSDB і знаходить шість із цих 10 LSA. R2 просить R1 надіслати тільки чотири додаткових LSA.

Зокрема, повідомлення OSPF, які фактично надсилають LSA між сусідами, мають назву пакетів оновлення стану зв'язку (Link-State Update, LSU). Тобто пакет LSU містить структури даних, які є LSA. Тому LSA це не пакети, а скоріше структури даних, які знаходяться всередині LSDB і описують топологію зв'язків. Рисунок 2.2 демонструє (об'єднує) деякі з цих термінів і процесів. Схема наводить приклад процесу обміну базою даних між маршрутизаторами R1 і R2.

Для каналів Ethernet OSPF за замовчуванням використовує мережевий тип широкомовної передачі, тому OSPF вибирає один із маршрутизаторів у підмережі який є призначеним маршрутизатором (designated router, DR). DR відіграє ключову роль у процесі обміну базою даних за іншими правилами, ніж у з'єднаннях «точка-точка».

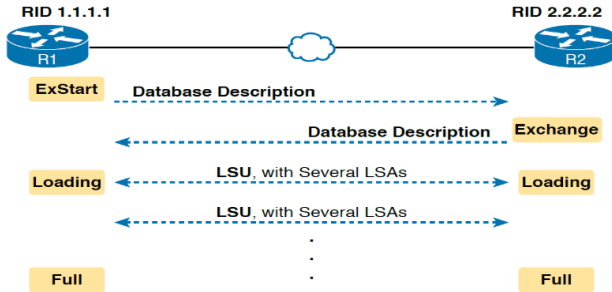


Рисунок 2.2 – Приклад обміну базою даних

Розглянемо приклад, який наведено на рисунку 2.3. Маємо п'ять маршрутизаторів на одній VLAN технології Ethernet, які підтримують роботу OSPFv2. П'ять OSPF маршрутизаторів обирають один маршрутизатор як призначений DR і один, як резервний DR (BDR).

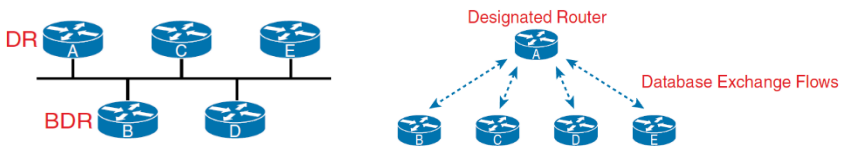


Рисунок 2.3 – Обмін базою даних через Ethernet

Процес обміну базою даних по каналу Ethernet відбувається між призначеним DR та кожним іншим маршрутизатором, при цьому DR гарантує, що всі інші маршрутизатори отримують копію кожного LSA. Іншими словами, обмін базою даних відбувається через потоки, які наведені на рисунку. Використання DR/BDR разом із використанням багатоадресних IP-адрес робить обмін OSPF LSDB більш ефективним у мережах, які дозволяють підключатися до одного каналу більше ніж двома маршрутизаторами.

### 2.3 Розрахунок найкращих маршрутів із SPF

Інженерам потрібно знати які маршрути, згідно SPF будуть обрані, як найкращі. Алгоритм SPF розраховує всі маршрути для підмережі, тобто всі можливі маршрути від маршрутизатора до мережі призначення. Якщо існує більше одного маршруту, маршрутизатор порівнює показники, обирає найкращий маршрут, за найнижчими показниками, і додає до таблиці маршрутизації. Інженер, маючи схему мережі і інформацію про стан, обчислює метрику кожного маршруту. SPF визначає маршрут, OSPF розраховує метрику, знаходить суму вартості всіх вихідних інтерфейсів. На рисунку 2.4 наведено приклад із трьома можливими маршрутами від R1 до підмережі X (172.16.3.0/24).

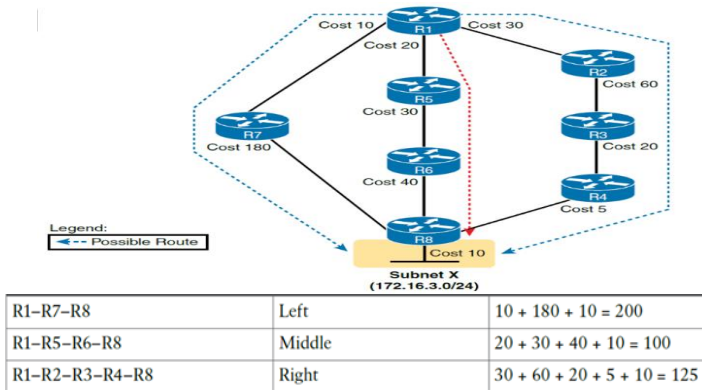


Рисунок 2.4 - Дерево SPF для пошуку маршруту

OSPF враховує тільки вартість вихідних інтерфейсів у кожному маршруті. Це не стосується вартості вхідних інтерфейсів у маршруті.

У результаті аналізу LSDB алгоритмом SPF, R1 додає маршрут до підмережі 172.16.3.0/24 в таблицю маршрутизації, вказавши маршрутизатор R5 як наступний.

Великі мережі OSPFv2 можуть бути не зовсім працездатними при налаштуванні їх в одній зоні (області, підмережі). Наприклад, є мережа підприємства, яка має не 11, а 900 маршрутизаторів і кілька тисяч підмереж. Тому маємо перелік вимог, які виникають при призначенні великої кількості маршрутизаторів в одну область (зону):

- велика топологічна база даних вимагає більше пам'яті на кожному маршрутизаторі;
- для алгоритму SPF потрібна обчислювальна потужність, яка експоненціально зростає із розміром топологічної бази даних;
- одна зміна параметрів інтерфейсу в будь-якій точці великої мережі вимагає знову запускати SPF на кожному маршрутизаторі.

Рішення в тому, щоб одну велику LSDB розбити на декілька менших LSDB за допомогою зон дії OSPF. При цьому кожне посилення передається тільки в одній області. SPF виконує складні обчислення для топології всередині конкретної області (зони). Наприклад, мережа з 1000 маршрутизаторами та з 2000 підмережами, розбитими на 100 областей, у середньому матиме 10 маршрутизаторів і 20 підмереж на одну область. Таким чином, розрахунок SPF на маршрутизаторі, оброблятиме лише топологію приблизно для 10 маршрутизаторів і 20 каналів зв'язків, а не для 1000 маршрутизаторів і 2000 каналів зв'язків.

Проектування зон OSPF здійснюється за правилами. Почніть з проектування об'єднаної мережі, розміщення маршрутизаторів і всіх їх інтерфейсів, оберіть область для кожного інтерфейсу. Рисунок 2.5 демонструє приклад. Інженер почав із діаграми мережі, яка має 11 маршрутизаторів з каналами зв'язків. Ліворуч, у зону 1, він підключив чотири канали глобальної мережі і локальні мережі, підключені до маршрутизаторів B1–B4. А в зону 2 - B11–B14 та їх локальні мережі. Обидві області потребують підключення до магістралі, це область 0, тому інтерфейси локальної мережі D1 і D2 підключені в область 0 разом із інтерфейсом D3, створюючи, таким чином магістральну область.

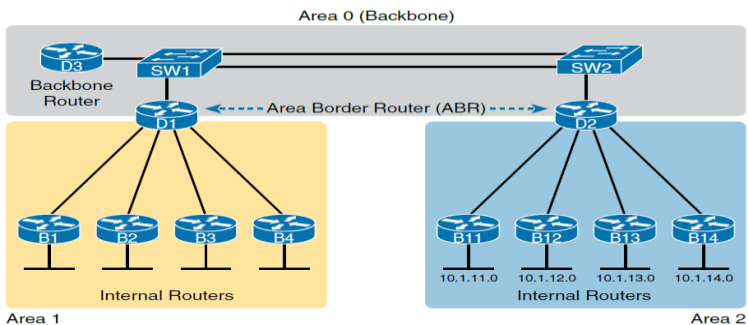


Рисунок 2.5 - Тризонний OSPF з D1 і D2 як ABR

Області зменшують робоче навантаження на SPF, оскільки для певної зони LSDB враховує лише маршрутизатори та канали зв'язку всередині цієї області. Маємо, що використання декількох областей покращує роботу OSPF для великих мереж.

## 3 ОПТИМІЗАЦІЯ МАРШРУТІВ МІЖ МЕРЕЖАМИ VLAN

### 3.1 Методи маршрутизації між VLAN

Етапи оптимізації, пов'язані з трьома методами маршрутизації між мережами VLAN:

- маршрутизація VLAN із магістральними лініями маршрутизатора 802.1Q;
- маршрутизація VLAN за допомогою комутаторів 3 рівня SVI (switched virtual interfaces);
- маршрутизація VLAN із маршрутизованими портами комутатора 3 рівня та концепція EtherChannel.

IP-адреси на маршрутизаторах або комутаторах 3 рівня, можуть служити шлюзами за замовчуванням у цих підмережах. Розіб'ємо параметри маршрутизації локальної мережі на чотири категорії:

- використання маршрутизатора з одним інтерфейсом локальної мережі та кабелем, підключеним до комутатора для VLAN;
- використання маршрутизатора із магістральною мережею VLAN, яка з'єднується з комутатором локальної мережі (ROAS);
- використання комутатору 3 рівня із SVI;
- використання комутатору з маршрутизованими інтерфейсами.

На рисунку 3.1 показані випадки, в яких можна використовувати наведені параметри.

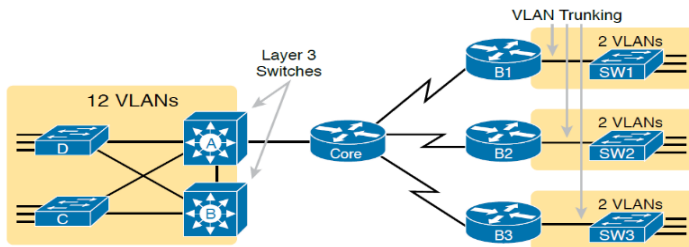


Рисунок 3.1 – Приклад використання

Два комутатори, позначені А і В, можуть працювати як комутатори 3 рівня. Обидва мають SVI і маршрутизовані інтерфейси. На рисунку ліворуч наведено центральну локальну мережу кампусу із 12 VLAN. Комутатори А і В діють як комутатори 3 рівня, поєднуючи функції маршрутизатора та комутатора, маршрутизуючи дані між усіма 12 підмережами/VLAN, а також маршрутизуючи дані до/та від основного маршрутизатора. На рисунку також наведений класичний випадок використання маршрутизатора з VLAN транкінгом. Віддалені мережі з правої сторони рисунка, можуть мати підключений до WAN маршрутизатор і LAN комутатор. Ці мережі, щоб скористатися можливістю маршрутизувати дані через магістраль із підтримкою стандарту 802.1Q, можуть використовувати керований ROAS.

Для підключення інтерфейсу до VLAN маршрутизатори використовують підінтерфейси. При цьому маршрутизатор повинен мати IP-адресу/та маску, пов'язану з кожною VLAN на магістралі. Cisco вирішує проблему одного фізичного інтерфейсу, створюючи, для з'єднання, декілька віртуальних інтерфейсів - по одному, пов'язаному з кожною VLAN на цій магістралі.

На рисунку 3.2 наведено цей приклад для маршрутизатора B1, як маршрутизатора ROAS однієї з гілок мережі з рисунку 3.1. Оскільки цей маршрутизатор потребує маршрутизації лише між двома мережами VLAN, то тут наведено два підінтерфейси G0/0.10 і G0/0.20, на яких можна застосовувати налаштування конфігурації для кожної VLAN. Маршрутизатор обробляє кадри з тегами VLAN 10 так, ніби вони надходять або відправлені з G0/0.10, а кадри з тегами VLAN 20 так, ніби вони надходять або відправлені з G0/0.20.

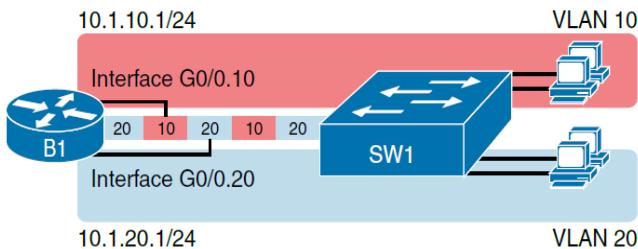


Рисунок 3.2 – Налаштування ROAS

Крім використання команди `show running-config`, конфігурацію ROAS на маршрутизаторі можна перевірити за допомогою двох команд: `show ip route [connected]` і `show vlans`. Як і у випадку з будь-яким інтерфейсом маршрутизатора, якщо інтерфейс знаходиться в активному стані та має налаштовану IPv4-адресу, IOS розмістить підключений маршрут в таблиці IP-маршрутизації. Отже, першою перевіркою повинна бути перевірка, чи існують усі з'єднані маршрути.

### 3.2 Маршрутизація за допомогою SVI комутаторів 3 рівня

Щоб виконати комутацію 3 рівня за допомогою SVI, у наступному прикладі наведемо офіс філії з двома мережами VLAN. Тепер проєкт використовує у комутаторі локальної мережі комутацію 3 рівня. На рисунку 3.3 бачимо зміни у схемі та конфігурацію для функції комутатора рівня 3 із піктограмою маршрутизатора всередині комутатора, щоб підкреслити, що комутатор маршрутизує пакети.

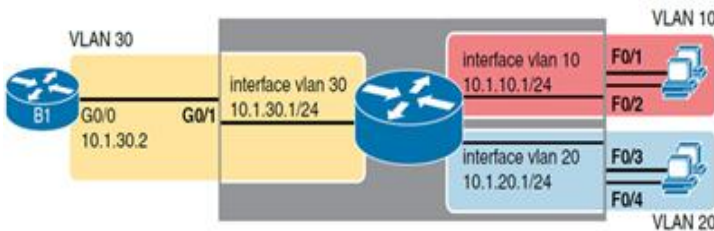


Рисунок 3.3 – Комутація 3 рівня за допомогою SVI

На рисунку показано внутрішні елементи комутатора рівня 3 у прямокутнику посередині рисунка. У філії все ще є дві VLAN (10 і 20) користувачів, тому комутатор рівня 3 потребує одного інтерфейсу VLAN на кожен VLAN. Піктограма маршрутизатора представляє функцію комутації рівня 3, з двома інтерфейсами VLAN праворуч від неї. Трафік все ще повинен дістатися до фізичного маршрутизатора B1, щоб отримати доступ до WAN, тому комутатор використовує третю VLAN (VLAN 30) для підключення до маршрутизатора B1. Фізичне з'єднання між комутатором рівня 3 і маршрутизатором B1 не є магістральним каналом, а є з'єднанням доступу (access link).

Щоб підтримувати маршрутизацію пакетів, комутатор додає підключені маршрути IP, як показано нижче (зверніть увагу, що кожен маршрут вказано як підключений до іншого інтерфейсу VLAN).

```
! legend omitted for brevity
10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C 10.1.10.0/24 is directly connected, Vlan10
L 10.1.10.1/32 is directly connected, Vlan10
C 10.1.20.0/24 is directly connected, Vlan20
L 10.1.20.1/32 is directly connected, Vlan20
C 10.1.30.0/24 is directly connected, Vlan30
L 10.1.30.1/32 is directly connected, Vlan30
```

Комутатор також потребує додаткових маршрутів до решти мережі. Комутатор рівня 3 може використовувати статичні маршрути або протокол маршрутизації, залежно від можливостей комутатора. Якщо ви увімкнули OSPF на комутаторі рівня 3 конфігурація та перевірка працюватимуть так само, як і на маршрутизаторі. Маршрути, які IOS додає до таблиці IP-маршрутизації комутатора рівня 3, перелічуватимуть інтерфейси VLAN як вихідні інтерфейси.

Коли усунення помилок SVI стосується їх стану, який пов'язаний зі станом VLAN, кожен інтерфейс VLAN має відповідний VLAN з таким самим номером і певний стан інтерфейсу VLAN. Зокрема, щоб інтерфейс VLAN перебував у включеному стані.

Крок 1. VLAN має бути визначено на локальному комутаторі (явно або за допомогою VTP).

Крок 2. Комутатор повинен мати принаймні один інтерфейс up/up за допомогою VLAN, один/або обидва:

- інтерфейс доступу (access interface) up/up, назначений VLAN;
- магістральний інтерфейс (trunk interface), для якого VLAN знаходиться в дозволеному списку, переадресовується STP і не скорочується VTP;

Крок 3. VLAN (а не інтерфейс VLAN) має бути адміністративно ввімкнено (тобто не командою shutdown).

Крок 4. Інтерфейс VLAN (а не VLAN) має бути адміністративно ввімкнено (тобто без команди shutdown).

Виконуючи ці кроки, майте на увазі, що VLAN та інтерфейс VLAN пов'язані, але елементи конфігурації задаємо в CLI. Якщо ви хочете маршрутизувати пакети для підмереж у VLAN 11, 12 і 13, відповідні інтерфейси VLAN мають бути пронумеровані 11, 12 і 13. І

VLAN і інтерфейси VLAN можна вимикати та вмикати за допомогою shutdown та без команди shutdown.

### 3.3 VLAN з маршрутизованими портами комутатора рівня 3

Коли комутатор рівня 3 потребує інтерфейсу мережевого рівня, підключеного до підмережі, і лише один фізичний інтерфейс підключається до цієї підмережі, мережевий інженер може вибрати використання маршрутизованого порту замість SVI. І навпаки, коли комутатор рівня 3 потребує багато фізичних інтерфейсів, то потрібно використовувати тільки SVI. Щоб зрозуміти чому, розглянемо схему на рисунку 3.4. Праворуч від комутатора принаймні два порти доступу належать як до VLAN 10, так і до VLAN 20. Маємо одне з'єднання від комутатора до маршрутизатора B1. Комутатор може налаштувати порт як порт доступу в окремій VLAN. Однак якщо потрібен лише один порт комутатора, комутатор може налаштувати це з'єднання як маршрутизований порт.

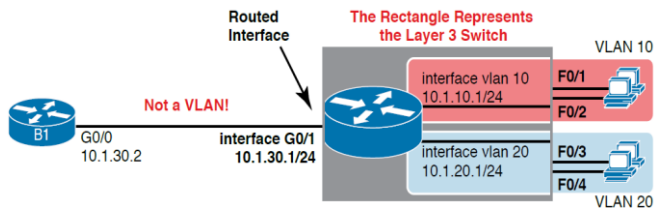


Рисунок 3.4 – Інтерфейс комутатора як маршрутизований інтерфейс

Увімкнути інтерфейс комутатора як маршрутизований інтерфейс замість комутованого інтерфейсу просто: потрібно скористатися підкомандою `no switchport` на фізичному інтерфейсі. Комутатори Cisco, які можуть бути комутаторами рівня 3, використовують команду `switchport` за замовчуванням для кожного фізичного інтерфейсу. За допомогою цієї команди Cisco повідомляє комутатору, що порт є портом рівня 2. Щоб порт перестав працювати як порт комутатора, а функціонував як порт маршрутизатора, потрібно скористатися командою `no switchport` на інтерфейсі.

Приклад показує завершену конфігурацію для інтерфейсів, налаштованих на комутаторі. Зверніть увагу, що тепер порт, який

підключений до підмережі 10.1.30.0, з метою оптимізації, налаштовано, як маршрутизований порт. І все, що потрібно зробити, так це додати команду по switchport до фізичного інтерфейсу та налаштувати IP-адресу на фізичному інтерфейсі.

Зокрема, для інтерфейсу, налаштованого як маршрутизований порт з IP-адресою, команди:

- `show interfaces`: подібно до команди на маршрутизаторі, на виході відображатиме IP-адресу інтерфейсу (для портів комутатора ця команда не відображає IP-адресу);

- `show interfaces status`: під заголовком «VLAN» замість переліку VLAN доступу (access) або слова trunk у виводі показано слово routed, що означає, що це маршрутизований порт;

- `show ip route`: у маршрутах визначає маршрутизований порт як вихідний інтерфейс;

- `show interfaces type number switchport`: якщо порт маршрутизований, то коротко підтверджує, що порт не є портом комутатора. (Якщо порт є портом рівня 2, ця команда надає багато деталей про конфігурацію та стан).

### 3.4 Впровадження 3-го рівня EtherChannels

Поки ми визначили, що маршрутизовані інтерфейси можна використовувати з одним з'єднанням «точка-точка» між парами комутаторів рівня 3 або між комутатором рівня 3 і маршрутизатором. Однак у більшості проєктів системні адміністратори використовують принаймні два канали зв'язку між кожною парою розподільних і основних комутаторів, як показано на рисунку 3.5.

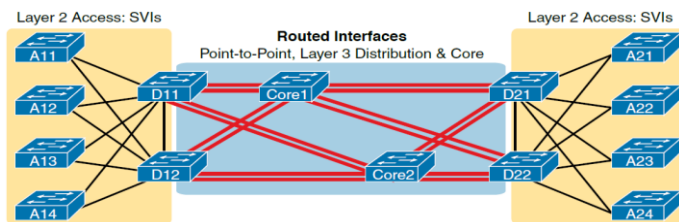


Рисунок 3.5 – З'єднання між парами комутаторів рівня 3

Хоча кожен окремий порт на рівнях розподілу та ядра можна розглядати як окремий маршрутизований порт, краще об'єднати кожен пару паралельних каналів у канал EtherChannel рівня 3. Без використання EtherChannel ви все одно можете зробити кожен порт на кожному комутаторі (в центрі рисунку) маршрутизованим портом. Однак, якщо ви ввімкнете протокол маршрутизації, але не використовуєте EtherChannels, кожен комутатор рівня 3 тепер вивчатиме два IP-маршрути, які пов'язані з тим самим сусіднім комутатором, що й наступний перехід - один маршрут для одного посилення, інший маршрут для іншого посилення.

Використання Layer 3 EtherChannel має сенс із декількома паралельними з'єднаннями між двома комутаторами. Таким чином, кожна пара посилень діє як одне посилення рівня 3. Окрім того, кожна пара комутаторів має один сусідній зв'язок за протоколом маршрутизації з сусідом, а не два. Кожен комутатор вивчає один маршрут до пункту призначення на пару посилень, а не два і це оптимізує їх роботу. Тепер IOS балансує трафік краще ніж із використанням декількох IP-маршрутів до однієї підмережі. Загалом, підхід рівня 3 EtherChannel працює набагато краще, ніж коли кожне посилення розглядається як окремий маршрутизований порт і використовує балансування рівня 3.

Але налаштування EtherChannel рівня 3 потребує трохи більше часу та етапів. Нам потрібно поєднати конфігурацію EtherChannel і маршрутизованого порту. Наступні кроки показують, як це зробити, припускаючи статичне налаштування.

Крок 1. Налаштуйте фізичні інтерфейси у режимі конфігурації інтерфейсу таким чином:

- додайте до каналу номер командою `channel-group number mode on`. Використовуйте однаковий номер для всіх фізичних інтерфейсів на одному комутаторі. При цьому номер (`channel-group number`), який використовується, може відрізнятися на двох сусідніх комутаторах;

- додайте команду `no switchport`, щоб зробити кожен фізичний порт портом маршрутизації.

Крок 2. Налаштуйте інтерфейс PortChannel:

- використовуйте команду `interface port-channel number`, щоб перейти в режим конфігурації `port-channel` для того самого номера каналу, який налаштовано на фізичних інтерфейсах;

- додайте команду `no switchport`, щоб переконатися, що інтерфейс `port-channel` діє як маршрутизований порт (IOS, може, вже її додала);

- використовуйте команду `ip address address mask`, щоб налаштувати адресу та маску.

Особливо важливо зауважити, що хоча всі фізичні інтерфейси та інтерфейс `PortChannel` є маршрутизованими портами, IP-адресу слід налаштовувати лише в інтерфейсі `PortChannel`.

## 4 ENHANCED INTERIOR GATEWAY ROUTING PROTOCOL

### 4.1 Особливості та функції EIGRP

Є багато функцій, завдяки яким протокол EIGRP справді виділяється серед інших протоколів. Далі наведені деякі з них:

- підтримка IP та IPv6 (інших протоколів маршрутизації) через модулі, які залежать від протоколу;

- вважається безкласовим (RIPv2 і OSPF);

- підтримка VLSM/CIDR;

- підтримка підсумовування маршрутів і розривних мереж;

- ефективне виявлення сусідів;

- зв'язок через надійний транспортний протокол (RTP);

- вибір найкращого шляху за допомогою алгоритму дифузійного оновлення (DUAL);

- зменшене використання пропускнуої здатності з обмеженими оновленнями;

- немає трансляцій.

В Cisco EIGRP має назви: протокол дистанційно-векторної маршрутизації; дистанційно-векторний протокол або гібридний протокол маршрутизації.

Перш ніж маршрутизатори, які працюють за протоколом EIGRP зможуть обмінюватися маршрутами один з одним, вони повинні стати сусідами, і є три умови, які повинні бути виконані (рис. 4.1).

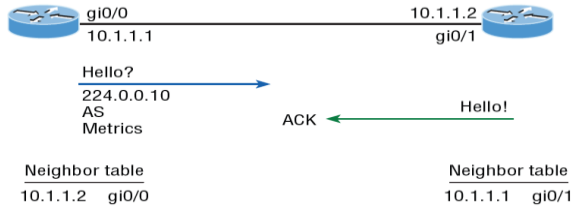


Рисунок 4.1 – Виявлення сусідів EIGRP

І цими умовами є обмін даними із безпосередньо підключеними сусідами: повідомлення Hello або, що підтвердження ACK отримано; номери AS збігаються та, при підключенні, показники ідентичні.

Маршрутизатори, які працюють за протоколом EIGRP та належать до різних AS, автоматично не обмінюються інформацією про маршрутизацію і, як ми розуміємо, не стають сусідами. І це є корисною функцією у великих мережах, оскільки зменшує кількість інформації про маршрут, що передається через певну AS. Але, оскільки показники відіграють важливу роль у виборі між значеннями, які потрібно оцінити під час вибору найкращого маршруту, є важливим, щоб усі сусіди за протоколом EIGRP погодилися із вибором конкретного маршруту. Це означає, що іноді перерозподіл між різними AS потрібно зробити вручну. Все це є важливим, оскільки обчислення на одному маршрутизаторі залежать від розрахунків його сусідів.

Таймер на повідомлення Hello (hello timer) між маршрутизаторами, які працюють за протоколом EIGRP, за замовченням, встановлено на 5 секунд. Є ще один таймер, що пов'язаний із таймером Hello – це таймер утримання (hold timer), який визначає кількість часу, що маршрутизатор готовий чекати, щоб отримати Hello від сусіда, перш ніж оголосити його не активним. Або це інтервал часу в секундах, після якого сусід вважається недоступним. Коли сусіда оголошено не активним, його видаляють із таблиці сусідів, і всі маршрути обчислюються повторно.

Конфігурація таймера утримання (hold timer) не визначає, скільки часу маршрутизатор чекає, перш ніж оголосити сусідів не активними; він визначає час, за яким маршрутизатори повинні чекати, перш ніж вони зможуть оголосити його не активним. Це означає, що таймери

утримання на сусідніх маршрутизаторах не повинні збігатися, оскільки вони повідомляють іншим лише про час очікування.

Єдиний раз, коли протокол EIGRP оголошує цю інформацію, це коли він виявляє нового сусіда та формує встановлення з ним зв'язку шляхом обміну пакетами Hello. Коли це відбувається, обидва сусіди передають повну інформацію один одному. І, як кожен дізнається маршрути сусіда, він поширює лише зміни в таблиці маршрутизації.

Під час кожного сеансу EIGRP на маршрутизаторі, створюється таблиця сусідів, у якій маршрутизатор зберігає інформацію про всі маршрутизатори, відомі як безпосередньо підключені сусіди. У таблиці також зберігаються: IP-адреса кожного сусіднього маршрутизатора; часовий інтервал утримання; таймер зворотного зв'язку (smooth round-trip timer, SRTT) і інформація про чергу. Це важлива довідка, яка використовується для встановлення того, що відбулися зміни топології, про які сусідні маршрутизатори повинні знати.

Маємо підсумок, що маршрутизатори EIGRP отримують оновлення своїх сусідів і зберігають їх у локальній таблиці топології. Таблиця містить усі відомі маршрути від усіх відомих сусідів і є вихідними даними, з яких вибираються найкращі маршрути, що оптимізує роботу мережі в цілому. Далі визначимо деякі терміни.

Повідомлена/оголошена відстань (Reported/advertised, RD/AD) – є показником віддаленої мережі, яку повідомляє сусід, або метрикою таблиці маршрутизації сусіднього маршрутизатора, яка збігається і відображається в таблиці топології, де першим числом є адміністративна відстань. На рисунку 4.2 маршрутизатори SF і NY оголошують шлях до маршрутизатора Corp, мережа 10.0.0.0, але вартість через SF до мережі 10.0.0.0 менша, ніж через NY. Маршрутизатор Corp має розрахувати вартість для кожного сусіда.

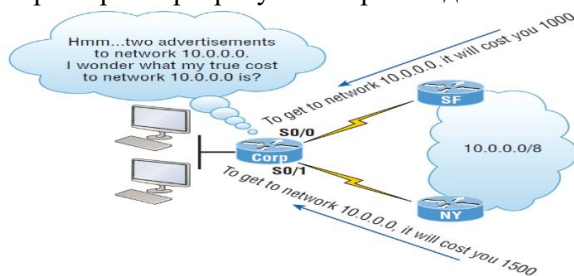


Рисунок 4.2 – Заявлена відстань

Feasible distance (FD) - допустима відстань є найкращою серед усіх шляхів до віддаленої мережі, включаючи метрику до сусіда, який оголошує віддалену мережу. Маршрут із найменшою FD фіксується в таблиці маршрутизації, оскільки вважається найкращим маршрутом.

Метрика можливої відстані обчислюється з використанням метрики, наданої сусідом, яка має назву оголошеною відстанню, плюс метрика сусіда, який повідомляє про маршрут. На рисунку 4.3 маршрутизатор Corp, у таблиці маршрутизації, має шлях через маршрутизатор SF до мережі 10.0.0.0, оскільки це найменша можлива відстань або найнижча вартість між кінцевими пристроями. Маємо маршрут EIGRP, який введено в таблицю маршрутизації, і допустиму відстань: D 10.0.0.0/8 [90/2195456] via 172.16.10.2, 00:27:06, Serial0/0.

Маємо, що сусідній маршрутизатор надсилає оголошену відстань (RD/AD) для мережі 10.0.0.0, а протокол EIGRP обчислює вартість доступу до сусіда, а потім сумує два числа разом, щоб отримати допустиму вартість FD, або загальну вартість.

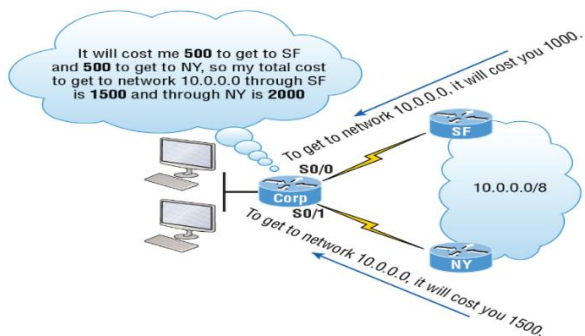


Рисунок 4.3 – Допустима відстань

Таблиця сусідів - це таблиця, в якій кожен маршрутизатор зберігає інформацію про стан сусідів. Коли знайдено нового сусіда, його адреса та інтерфейс фіксуються в таблиці сусідів, яка зберігається в оперативній пам'яті. Порядкові номери використовуються для порівняння підтвердження з пакетами оновлення. Останній порядковий номер, отриманий від сусіда, записується, для виявлення не упорядкованих пакетів. Це є корисним для усунення помилок у встановленні зв'язків між сусідніми маршрутизаторами.

Таблиця топології є таблицею, яка заповнюється таблицею сусідів, при цьому алгоритм розповсюдження оновлення (Diffusing Update Algorithm, DUAL) обчислює найкращий шлях без петель до кожної віддаленої мережі. Таблиця містить усі пункти призначення, оголошені сусідніми маршрутизаторами, з адресою кожного пункту призначення та списком сусідів, які оголосили пункт призначення. Для кожного сусіда записується оголошена метрика (відстань), яка надходить лише з таблиці маршрутизації сусіда, а також допустима вартість FD. Найкращий шлях до кожної віддаленої мережі копіюється та розміщується в таблиці маршрутизації, а потім IP використовуватиме цей маршрут для перенаправлення трафіку до віддаленої мережі. Маршрут, який скопійований до таблиці маршрутизації, це адреса наступного маршрутизатора і він є успішним маршрутом. Маршрут із хорошою, але менш бажаною вартістю додається в таблицю топології як резервний маршрут та має назву можливого наступного (Feasible successor) маршруту.

Можливий наступний (Feasible successor, FS) маршрут – це запис в таблиці топології, який ідентифікує маршрут, який є нижчим (за вартістю) за наступні маршрути. FS визначається як шлях, оголошена відстань якого менша за можливу відстань поточного маршруту і є резервним маршрутом. EIGRP зберігатиме до 32 можливих наступних маршрутів у таблиці топології в версії 15.0, і лише до 16 у попередніх версіях IOS. Команда `show ip eigrp topology` відобразить усі можливі наступні маршрути EIGRP, які відомі маршрутизатору.

Можливий наступний маршрут є резервним маршрутом і зберігається в таблиці топології. Наступний маршрут зберігається в таблиці топології, копіюється та розміщується в таблиці маршрутизації, щоб протокол IP міг використовувати його для доступу до віддаленої мережі. Таблиця маршрутизації містить лише наступні маршрути; таблиця топології містить і наступні і можливі наступні маршрути.

На рисунку 4.4 показано, що маршрутизатори SF і NY мають підмережі в мережі 10.0.0.0, а Corp має два шляхи доступу до неї.

Як показано на рисунку, є два шляхи до мережі 10.0.0.0, які можуть використовуватися маршрутизатором Corp. EIGRP вибирає найкращий шлях і розміщує його в таблиці маршрутизації, але якщо обидва канали зв'язку мають шляхи однакової вартості, то EIGRP балансує навантаження між ними, використовуючи до чотирьох каналів за замовчуванням.

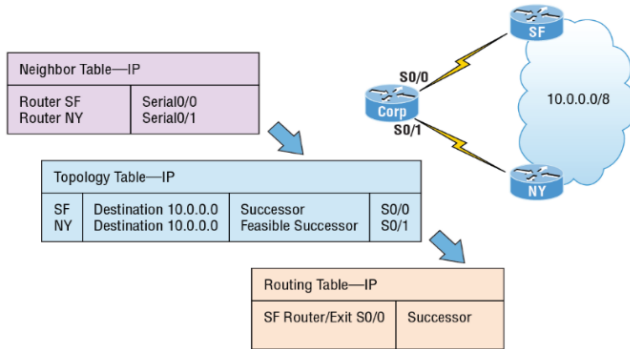


Рисунок 4.4 – Таблиці EIGRP

Маючи можливі наступні маршрути в таблиці топології у вигляді резервних каналів, мережа може миттєво об'єднуватися, а оновлення будь-якого сусіда складають єдиний трафік, який надсилається за протоколом EIGRP.

## 4.2 Налаштування EIGRP

EIGRP залежить від власного протоколу, який має назву надійного транспортного протоколу (Reliable Transport Protocol, RTP). Він використовується для керування передачею повідомлень між маршрутизаторами, які спілкуються за протоколом EIGRP.

Маємо п'ять різних типів пакетів, які використовує EIGRP:

- Update - пакет оновлення містить інформацію про маршрут;
- Query - є запитом на певні маршрути, у ньому завжди використовується надійний метод групової розсилки;
- Reply - пакет відповіді надсилається у відповідь на запит за допомогою методу одноадресної передачі;
- Hello - пакет привітання використовується для виявлення сусідів EIGRP і надсилається через ненадійну групову розсилку, тобто не потребує підтвердження;
- ACK - пакет підтвердження надсилається у відповідь на оновлення та завжди є одноадресним.

Всі пакети, є типами контейнерів, які маршрутизатори EIGRP використовують для зв'язку зі своїми сусідами.

Для вибору та підтримки найкращого шляху до кожної віддаленої мережі, протокол EIGRP використовує алгоритм розповсюдженого оновлення (Diffusing Update Algorithm, DUAL). Цей алгоритм дозволяє EIGRP виконувати наступні важливі завдання:

- визначає резервний маршрут, якщо він доступний;
- підтримує маски підмережі змінної довжини (VLSM);
- виконує динамічне відновлення маршруту;
- здійснює запит до сусідів, щодо невідомих альтернативних маршрутів;
- надсилає запити щодо альтернативного маршруту.

Але, насправді, цей алгоритм є таким чудовим тому, що він надає протоколу EIGRP можливість виконувати дуже швидку конвергенцію.

Ключом до швидкості є те, що:

- по-перше, маршрутизатори EIGRP зберігають копії всіх маршрутів своїх сусідів, і використовують їх для розрахунку власної вартості для кожної віддаленої мережі. Отже, якщо найкращий маршрут проходить, всі вказані точки, то, щоб знайти інший маршрут, здійснюється швидке сканування таблиці топології в пошуках можливого наступного маршрутизатора;

- по-друге, якщо таке швидке сканування таблиці топології не спрацює, маршрутизатори EIGRP негайно звертаються до своїх сусідів за допомогою, щоб знайти найкращий шлях. Саме такий підхід та використання інформації інших маршрутизаторів пояснює «розподілену» основу цього алгоритму.

На відміну від інших протоколів маршрутизації, які поширюють зміни по всій мережі, оновлення протоколу EIGRP поширюються лише настільки, наскільки це необхідно. Тому для чіткої роботи алгоритму розповсюдження оновлення (Diffusing Update Algorithm, DUAL) потрібно дотримуватись трьох критичних умов: сусіди визначаються або помічаються, як неактивні протягом обмеженого часу; всі передані повідомлення отримані правильно; всі зміни та повідомлення обробляються в тому порядку, у якому вони виявлені.

І так є два режими для введення команд EIGRP:

- режим конфігурації маршрутизатора;
- режим конфігурації інтерфейсу.

У режимі конфігурації маршрутизатора вмикаємо протокол, визначаємо, у яких мережах працюватиме EIGRP і встановлюємо

глобальні параметри. У режимі конфігурації інтерфейсу виконуємо загальні налаштування та задаємо пропускну здатність.

Розглянемо налаштування протоколу EIGRP для автономної системи з номером 20, на маршрутизаторі Corp, підключеному до чотирьох мереж. На рисунку 4.5 показано мережу, яку будемо далі налаштовувати.

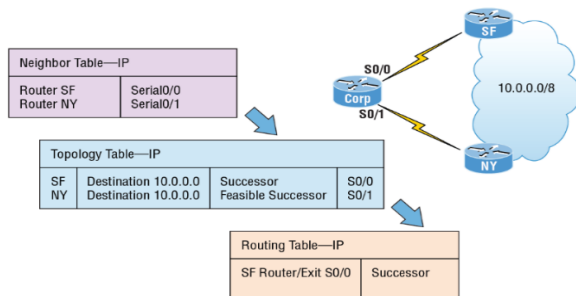


Рисунок 4.5 – Налаштування мережі

Маємо наступну конфігурацію маршрутизатора Corp:

```
Corp#config t
Corp(config)#router eigrp 20
Corp(config-router)#network 172.16.0.0
Corp(config-router)#network 10.0.0.0
```

Пам'ятаємо, що як і під час налаштування RIP, потрібно використовувати класову мережеву адресу, для якої всі біти відповідальні за адресацію підмереж та хостів є вимкненими.

EIGRP використовує поняття AS для ідентифікації групи маршрутизаторів, які обмінюватимуться інформацією про маршрут. Лише маршрутизатори, які мають однаковий номер AS, спільно використовують маршрути. Діапазон значень для створення AS з EIGRP становить від 1 до 65535.

```
Corp(config)#router eigrp ?
<1-65535> Autonomous System
WORD EIGRP Virtual-Instance Name
Corp(config)#router eigrp 20
```

Поки всі маршрутизатори використовують однаковий номер, вони створюватимуть суміжність. Тепер номер AS має сенс, але схоже, що ми можемо ввести слово замість номера AS.

```
Corp(config)#router eigrp Todd
Corp(config-router)#address-family ipv4 autonomous-system
20
Corp(config-router-af)#network 10.0.0.0
Corp(config-router-af)#network 172.16.0.0
```

Це не є обов'язковим для використання в будь-якій конфігурації IPv4-маршрутизації у мережі. Але цей останній приклад конфігурації вже є опцією в коді IOS.

Зверніть увагу, що протокол EIGRP автоматично підсумовує мережі. Функція є дуже корисною, але перевіривши таблицю на Corp, бачимо ускладнення від такого автоматичного підсумовування.

```
Corp#sh ip route
[output cut]
172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C 172.16.10.4/30 is directly connected, Serial0/1
C 172.16.10.0/30 is directly connected, Serial0/0
D 172.16.0.0/16 is a summary, 00:01:37, Null0
10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C 10.10.10.0/24 is directly connected, GigabitEthernet0/0
D 10.0.0.0/8 is a summary, 00:01:19, Null0
C 10.10.11.0/24 is directly connected, GigabitEthernet0/1
```

Тепер це виглядає не так добре, бо 172.16.0.0/16 і 10.0.0.0/8 оголошуються (розсилаються) як зведені маршрути за протоколом EIGRP, але, як ви бачите, ми маємо декілька підмереж у класовій мережевій адресі 10.0.0.0/8.

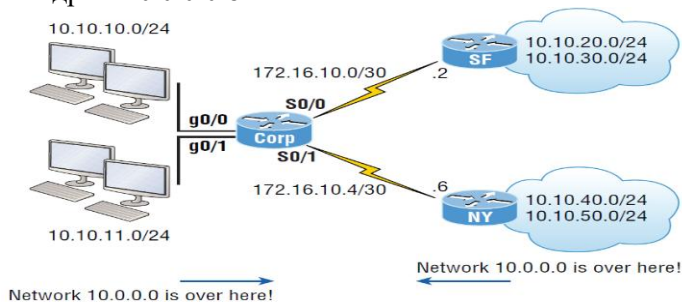


Рисунок 4.6 – Несуміжні мережі

Сукупність мереж, які ми використовуємо, є частинами розподіленої мережі, тому що у нас є мережа класу А 10.0.0.0/8, розділена на інший клас адрес, це адреси 1010.10(11).0/24. Далі мережа класу В розділена на підмережі 172.16.0.0/16 (172.16.10.0(4)/30), і ще група підмереж мережі класу А 10.0.0.0/8 (10.10.20(30).0/24) по обидва боки від двоточкових мереж (172.16.10.0(4)/30). Маршрутизатори SF і NY можуть створити автоматично агреговану адресу 10.0.0.0/8, а потім ввести її у свої таблиці маршрутизації. І це вже буде проблемою. Тому для цього типу топології кращим варіантом є вимкнення автоматичного підсумовування. І це є єдиним варіантом, щоб мережа працювала.

### 4.3 Балансування навантаження

Протокол EIGRP автоматично підтримує балансування навантаження за чотирма рівноцінними маршрутами та, починаючи з 15.0 версії IOS, може налаштовуватись на підтримку до 32 рівноцінних маршрутів. Розглянемо, як налаштувати балансування навантаження з нерівною вартістю за допомогою протоколу EIGRP.

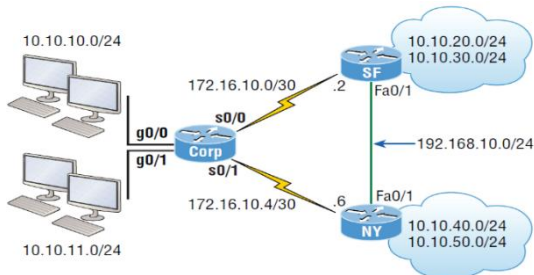


Рисунок 4.7 – Процес вибору маршруту EIGRP

Спочатку подивимося на налаштування маршрутизатора Corp, ввівши команду show ip protocols:

```
Corp#sh ip protocols
Routing Protocol is "eigrp 20"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Default networks flagged in outgoing updates
Default networks accepted from incoming updates
```

```

EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
EIGRP maximum hopcount 100
EIGRP maximum metric variance 1
Redistributing: eigrp 20
EIGRP NSF-aware route hold timer is 240s
Automatic network summarization is not in effect
Maximum path: 4
Routing for Networks:
10.0.0.0
172.16.0.0
Routing Information Sources:
Gateway                Distance                Last Update
(this router)          90                      19:15:10
172.16.10.6            90                      00:25:38
172.16.10.2            90                      00:25:38
Distance: internal 90 external 170

```

Відхилення (дисперсія) 1 означає балансування навантаження за рівнозначними маршрутами із максимальним їх числом, встановленим за замовчуванням, яке дорівнює 4. На відміну від більшості інших протоколів, як ми вже знаємо, EIGRP також підтримує балансування навантаження з нерівною вартістю за допомогою такого ж параметра дисперсії (відхилення).

Припустімо, що для параметра дисперсії встановлено значення 2. Це дозволяє ефективно розподілити трафік між найкращим маршрутом і будь-яким маршрутом із відстанню, яка, наприклад, є вдвічі більшою. Але балансування навантаження відбувається пропорційно, відносно вартості маршруту, тобто більша частина трафіку пройде по найкращому маршруту. Далі налаштуємо відхилення на Corp і подивимось, чи можна збалансувати навантаження між можливими наступними маршрутизаторами в цій мережі:

```

Corp# config t
Corp(config)#router eigrp 20
Corp(config-router)#variance 2
Corp(config-router)#
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-
Table:20):route installed for 10.10.20.0
.....
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-
Table:20):route installed for 10.10.50.0
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-
Table:20):route installed for 192.168.10.0

```

```

*Feb      26      22:24:24:IP-EIGRP(Default-IP-Routing-
Table:20):route installed for 192.168.10.0
Corp(config-router)#do show ip route
[output cut]
D 192.168.10.0/24    [90/2172416]    via    172.16.10.6,
00:00:18, Serial0/1
[90/2172416] via 172.16.10.2, 00:00:18, Serial0/0
172.16.0.0/30 is subnetted, 2 subnets
C 172.16.10.4 is directly connected, Serial0/1
C 172.16.10.0 is directly connected, Serial0/0
10.0.0.0/24 is subnetted, 6 subnets
C 10.10.10.0 is directly connected, GigabitEthernet0/0
C 10.10.11.0 is directly connected, GigabitEthernet0/1
D 10.10.20.0 [90/2300416] via 172.16.10.6, 00:00:18,
Serial0/1
[90/2297856] via 172.16.10.2, 00:00:19, Serial0/0
D 10.10.30.0 [90/2300416] via 172.16.10.6, 00:00:19,
Serial0/1
[90/2297856] via 172.16.10.2, 00:00:19, Serial0/0
D 10.10.40.0 [90/2297856] via 172.16.10.6, 00:00:19,
Serial0/1
[90/2300416] via 172.16.10.2, 00:00:19, Serial0/0
D 10.10.50.0 [90/2297856] via 172.16.10.6, 00:00:20,
Serial0/1
[90/2300416] via 172.16.10.2, 00:00:20, Serial0/0
Corp(config-router)#

```

Все працює. Тепер ми маємо два шляхи до кожної віддаленої мережі в таблиці маршрутизації, навіть якщо можливі відстані до кожного маршруту не однакові.

Нерівне балансування навантаження не ввімкнено за замовчуванням і можна виконувати балансування навантаження через маршрути, які мають у 128 разів гірші показники, ніж наступний (або кращий за вартістю) маршрут.

Розщеплений горизонт - метод запобігання петель маршрутизації, викликаних повільною збіжністю дистанційно-векторних протоколів. Його ввімкнено на інтерфейсах за замовчуванням, і це означає, що якщо оновлення маршруту отримано на інтерфейсі від сусіднього маршрутизатора, то цей інтерфейс не пересилатиме інформацію про ці мережі назад маршрутизатору, який її надіслав.

Далі подивимося на інтерфейс і розглянемо приклад:

```
Corp#sh ip int S0/0
```

```

Serial0/0 is up, line protocol is up
Internet address is 172.16.10.1/24
Broadcast address is 255.255.255.255
Address determined by setup command
MTU is 1500 bytes
Helper address is not set
Directed broadcast forwarding is disabled
Multicast reserved groups joined: 224.0.0.10
Outgoing access list is not set
Inbound access list is not set
Proxy ARP is enabled
Local Proxy ARP is disabled
Security level is default
Split horizon is enabled
[output cut]

```

Бачимо інформацію про налаштування серіал-порту Serial0/0: адресу і маску; значення MTU, яке дорівнює 1500 байт; Multicast адресу; інформацію про те, що Proxy ARP працює і розщеплений горизонт ввімкнено за замовчуванням.

У більшості випадків, ввімкнення розщепленого горизонту за замовчуванням більше допомагає, ніж шкодить. Розглянемо мережу на рисунку 4.8 і пояснимо, що робить команда `split horizon`. Зверніть увагу, що маршрутизатори SF і NY оголошують свої маршрути маршрутизатору Corp.

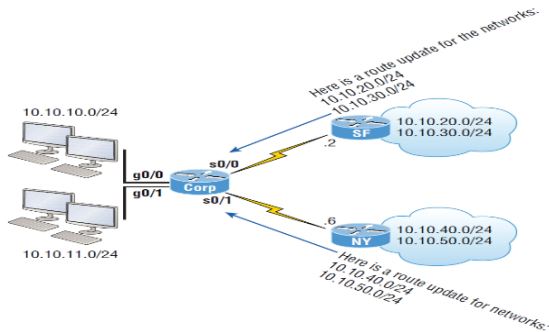


Рисунок 4.8 – Метод розщепленого горизонту

Тепер подивимося, що Corp надсилає кожному маршрутизатору (рис. 4.9). Маршрутизатор Corp не оголошує вже оголошені мережі,

інформацію про які він отримав на кожному інтерфейсі. Це рятує маршрутизатори SF і NY від отримання невірної інформації про маршрут, через який вони могли б дістатися до своєї мережі використовуючи Corp і невірний маршрут.

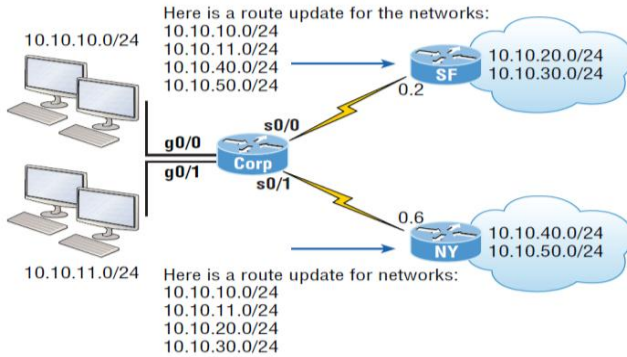


Рисунок 4.9 – Розщеплений горизонт (повідомлення)

Найчастіше це створює проблему на з'єднаннях «точка-багатоточка». Прикладом є Frame Relay в ситуації, коли декілька віддалених маршрутизаторів підключаються до одного інтерфейсу. У цьому випадку, для вирішення проблеми розщепленого горизонту в інтерфейсі «точка-багато точок», можна використовувати логічні інтерфейси, які є підінтерфейсами.

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Ладієва Л.Р. Методи оптимізації та пошуку оптимальних рішень. Навчальний посібник. Київ, 2023. 73с.
2. Глушик М.М., Копич І.М., Сороковський В.М. Математичне програмування: підручник. Львів: Новий Світ, 2014. 280 с.
3. Клименко М. І., Панасенко Є. В., Ткаченко І. Г. Оптимальне керування : конспект лекцій для здобувачів ступеня вищої освіти магістра спеціальності «Математика» освітньо-професійної програми «Математика». Запоріжжя: ЗНУ, 2023. 77 с.
4. Соколов С.В. Оптимальні та адаптивні системи. Суми: Сумський державний університет, 2018. 221 с.
5. Odom, W. CCNA 200-301 Official Cert Guide, Volume 1. Cisco Press. 2019.
6. Odom, W. CCNA 200-301 Official Cert Guide, Volume 2. Cisco Press. 2019.
7. Santitoro, R. Metro Ethernet Services–A Technical Overview. In Metro Ethernet Forum (Vol. 2006). 2003.
8. Hucaby D. CCNP Routing and Switching SWITCH 300-115 Official Cert Guide. 2nd Edition. USA: Cisco Press, 2015. 578 p.
9. Бортник Г. Г., Васильківський М. В., Кичак В. М. Транспортні телекомунікаційні технології. Вінниця: ВНТУ, 2017. 162 с.
10. Балашов В. О., Барба І.Б., Корнійчук В.І., Лашко А.Г., Ляховецький Л.М., Орешков В.І. Проектування, будівництво та експлуатація мереж широкосмугового доступу. Навчальний посібник. Одеса, 2012.