ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Комп'ютерна мережа з використанням технології сегментної маршрутизації»

на здобуття освітнього ступеня магістра
зі спеціальності 123 Комп'ютерна інженерія
освітньо-професійної програми «Комп'ютерні системи та мережі»

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

_____ Артем АНТОНЕНКО
(підпис)

Виконав: здобувач вищої освіти групи КСДМ-61
Артем АНТОНЕНКО

Керівник: Сергій КОРОТКОВ
PhD, доцент

Рецензент: _____
науковий ступінь, вчене звання *Ім'я, ПРІЗВИЩЕ*

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерна інженерія

Ступінь вищої освіти Магістр

Спеціальність 123 Комп'ютерна інженерія

Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
_____ Наталія ЛАЩЕВСЬКА
«_____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Антоненку Артему Васильовичу

1. Тема кваліфікаційної роботи: «Комп'ютерна мережа з використанням технології сегментної маршрутизації»

керівник кваліфікаційної роботи Сергій КОРОТКОВ, PhD, доцент,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025 р. № 467.

2. Строк подання кваліфікаційної роботи «26» грудня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, стандарти обміну даними, технологічні платформи та інструменти розроблення, методи інтеграції сервісів та управління інформаційними потоками.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналітичний огляд технологій для розгортання Segment Routing

2. Дослідженяя аспектів використання Segment Routing

3. Дослідження переваг та недоліків Segment Routing у моделюванні комп’ютерної мережі

5. Перелік ілюстративного матеріалу: *презентація*

6. Дата видачі завдання «01» вересня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	16.10.2025 – 23.10.2025	Виконано
2	Вивчення досвіду використання технологій в комп’ютерних мережах	24.11.2025	Виконано
3	Аналітичний огляд технологій для розгортання Segment Routing	25.11.2025 – 26.11.2025	Виконано
4	Дослідження аспектів використання Segment Routing	27.11.2025	Виконано
5	Дослідження переваг та недоліків Segment Routing у моделюванні комп’ютерної мережі	28.11.2025 – 29.11.2025	Виконано
6	Моделювання комп’ютерної мережі з використанням технології сегментної маршрутизації	28.11.2025 – 29.11.2025	Виконано
7	Оформлення роботи: вступ, висновки, реферат	23.10.2025 – 30.11.2025	Виконано
8	Розробка демонстраційних матеріалів	29.11.2025	Виконано
9	Попередній захист роботи	16.12.2025	Виконано

Здобувач вищої освіти _____ Артем АНТОНЕНКО
(підпис)

Керівник
кваліфікаційної роботи _____ Сергій КОРОТКОВ
(підпис)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут Інформаційних технологій

ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра

Направляється здобувач АНТОНЕНКО А. В. до захисту кваліфікаційної роботи

за спеціальністю 123 Комп'ютерна інженерія

освітньо-професійної програми «Комп'ютерні системи та мережі»

на тему: «Комп'ютерна мережа з використанням технології сегментної маршрутизації».

Кваліфікаційна робота і рецензія додаються.

Директор ННІ ІТ _____ Катерина НЕСТЕРЕНКО
(підпис)

Висновок керівника кваліфікаційної роботи

Здобувач Антоненко А. В. показав належний рівень теоретичної підготовки, вміння працювати з сучасними інформаційними технологіями та використовувати навчальну і наукову-технічну літературу. Під час виконання кваліфікаційної роботи продемонстрував самостійність у прийнятті рішень, відповідальне ставлення до поставлених завдань і зацікавленість у дослідницький діяльності. Робота виконана вчасно та відповідає вимогам, що висуваються до магістерських кваліфікаційних робіт_____

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Антоненка Артема Васильовича на оцінку «відмінно» та присвоїти йому кваліфікацію магістр з комп'ютерної інженерії.

Керівник кваліфікаційної роботи _____ Сергій КОРОТКОВ
(підпис)

«____» _____ 2025 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Антоненко А. В. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри КІ _____ Наталія ЛАЩЕВСЬКА
(підпис)

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну магістерську роботу

здобувача вищої освіти Антоненко Артема Васильовича

на тему «Комп'ютерна мережа з використанням технології сегментної маршрутизації»

Актуальність.

Кваліфікаційна магістерська робота здобувача вищої освіти Антоненко Артема Васильовича – присвячена проєктуванню та реалізації комп'ютерної мережі з використанням технології сегментної маршрутизації. Дослідження можливостей використання Segment Routing та сфер її застосування є науково актуальним і заслуговує на більше освітлення в наукових колах.

Позитивні сторони.

1. Дослідження дає можливість поєднати теоретичний аналіз сучасних стандартів IETF і 3GPP з практичними експериментами, від моделювання в Mininet чи EVE-NG до реального прототипування на Linux/VPP або доступному апаратному забезпеченні (SONiC, Tofino-емулятори). Такий баланс теорії та практики робить роботу актуальною для захисту та подальшого використання результатів у реальних проєктах.

2. При написанні кваліфікаційної магістерської роботи автором були використані різноманітні дієві методи дослідження та аналізу інформації.

3. Автор продемонстрував гарні аналітичні та інженерні здібності, вміння аналізувати та структуровано подавати зібрану інформацію, робити самостійні висновки та виконувати проєктування і практичну реалізацію комп'ютерної мережі з використанням технології сегментної маршрутизації.

Недоліки.

1. SRv6 охоплює занадто багато аспектів (маршрутизація, захист, SFC, слайсування, телеметрія, міграція) і деякі аспекти вимагають більш детального розгляду.

2. В роботі бажано було б розширити опис механізмів моніторингу та контролю роботи мережі.

Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної магістерської роботи.

Висновок: *кваліфікаційна робота на здобуття ступеня магістра заслуговує оцінку " відмінно ", а здобувач **Антоненко Артем Васильовича** заслуговує присвоєння кваліфікації: магістр з комп'ютерної інженерії.*

Рецензент: _____ _____
науковий ступінь, вчене звання *підпис* *Ім'я, ПРІЗВИЩЕ*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 130 стор., 4 табл., 36 рис., 53 джерел.

Об`єкт дослідження — маршрутизація трафіку в мережах.

Предмет дослідження — оптимізація маршрутизації в мережах завдяки впровадженню технології Segment Routing.

Мета роботи — обґрунтувати доцільність використання технології Segment Routing для оптимізації маршрутизації.

Методи дослідження — емпіричні, аналіз літературних джерел, методи проектування мереж.

В даній роботі розглядається технологія Segment Routing та переваги її застосування для оптимізації маршрутизації. Основна особливість Segment Routing це можливість вибирати маршрут трафіку завдяки стеку міток (сегментів) в заголовку пакета. Саме тому в роботі розглядається для порівняння технологія MPLS як попередник Segment Routing (а саме SR-MPLS), та технологія SR IPv6 як більш нова та перспективна версія SR.

З метою виконання даної задачі були проаналізовані можливості Segment Routing, а також можливі цілі її застосування. В роботі приводяться приклади розгортання мереж на основі Segment Routing та описуються отримані переваги від використання цієї технології. Для підкреслення переваг та особливостей Segment Routing, в роботі наводиться порівняння її з похожими технологіями, а також порівняння її двох реалізацій (SR-MPLS та SRv6).

Можливості Segment Routing управління трафіком дозволяють будувати гнучкі відмовостійкі мережі. Завдяки використанню SDN в поєднанні з SR можна будувати легкі для масштабування прості за структурою мережі з широкими можливостями настройок політик керування трафіком. Крім цього в майбутньому невідворотно IPv6 стане більш розповсюдженим, а тому актуальність технології SRv6 також зростає. Саме тому дослідження можливостей використання Segment Routing та сфер її застосування є науково актуальним і заслуговує на більше освітлення в наукових колах.

Сфера застосування — інформаційні технології, проектування мереж, побудова SDN-мереж.

КЛЮЧОВІ СЛОВА: IPV6, SOURCE ROUTING, SEGMENT ROUTING, SRV6, MPLS, SDN, SFC, NETWORK PROGRAMMING MODEL, TRAFFIC ENGEENERING, TRAFFIC MATRIX, TI-LFA FRR, LDP, RSVP-TE

ABSTRACT

Text part of the qualification work for obtaining a master's degree: 130 pages, 4 tables, 36 figures, 53 sources.

The object of the study is traffic routing in networks.

The subject of the study is optimization of routing in networks through the implementation of Segment Routing technology.

The purpose of the work is to substantiate the feasibility of using Segment Routing technology to optimize routing.

Research methods are empirical, analysis of literary sources, network design methods.

This work considers the Segment Routing technology and the advantages of its use for routing optimization. The main feature of Segment Routing is the ability to choose a traffic route due to a stack of labels (segments) in the packet header. That is why the work considers for comparison the MPLS technology as the predecessor of Segment Routing (namely SR-MPLS), and the SR IPv6 technology as a newer and more promising version of SR.

In order to fulfill this task, the capabilities of Segment Routing were analyzed, as well as possible purposes of its application. The paper provides examples of deploying networks based on Segment Routing and describes the benefits obtained from using this technology. To emphasize the advantages and features of Segment Routing, the paper compares it with similar technologies, as well as compares its two implementations (SR-MPLS and SRv6).

The capabilities of Segment Routing traffic management allow you to build flexible fault-tolerant networks. Thanks to the use of SDN in combination with SR, you can build easy-to-scale networks with a simple structure and wide possibilities for configuring traffic management policies. In addition, in the future, IPv6 will inevitably become more widespread, and therefore the relevance of SRv6 technology is also increasing. That is why the study of the possibilities of using Segment Routing and its application areas is scientifically relevant and deserves more coverage in scientific circles.

Scope of application - information technologies, network design, construction of SDN networks.

KEYWORDS: IPV6, SOURCE ROUTING, SEGMENT ROUTING, SRV6, MPLS, SDN, SFC, NETWORK PROGRAMMING MODEL, TRAFFIC ENGINEERING, TRAFFIC MATRIX, TI-LFA FRR, LDP, RSVP-TE

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ…… 9

ВСТУП…… 10

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ РОЗГОРТАННЯ SEGMENT ROUTING…….. 12

1.1 Протокол внутрішнього шлюзу Interior Gateway Protocol…….. 12

1.2 Протокол прикордонного шлюзу Border Gateway Protocol…… 17

1.3 Багатопротокольна комутація за мітками MultiProtocol Label Switching…… 19

1.4 Програмно-конфігуровані мережі Software-Defined Networking…. 31

1.5 Ланцюги мережевих функцій Service Function Chaining…… 36

1.6 Сегментна маршрутизація Segment Routing…… 36

РОЗДІЛ 2. ДОСЛІДЖЕННЯ АСПЕКТІВ ВИКОРИСТАННЯ SEGMENT ROUTING…….. 58

2.1 Цілі застосування Segment Routing…… 58

2.2 Застосування Segment Routing в різних мережах (SPRING) …….. 76

2.3 Програмні реалізації Segment Routing…… 88

2.4 Напрямки розвитку Segment Routing…… 98

РОЗДІЛ 3. ДОСЛІДЖЕННЯ ПЕРЕВАГ ТА НЕДОЛІКІВ SEGMENT ROUTING У МОДЕЛЮВАННІ КОМП'ЮТЕРНОЇ МЕРЕЖІ …….. 101

3.1 Переваги використання Segment Routing…….. 101

3.2 Порівняння LDP, RSVP-TE та SR…….. 105

3.3 Порівняння MPLS, SR-MPLS та SRv6……..... 111

3.4 Моделювання комп'ютерної мережі з використанням технології сегментної маршрутизації…… 114

ВИСНОВКИ…… 122

ПЕРЕЛІК ПОСИЛАНЬ……..... 124

ДОДАТКИ……... 130

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AS — Autonomous System;

SR — Segment Routing;

SRv6 — Segment Routing версії IPv6;

SRH — Segment Routing Header;

SPRING — Source Packet Routing in Networking;

SID — Segment ID;

BSID — Binding SID;

SRGB — Segment Routing Global Block;

BE — Best Effort (режим роботи вузла в SRv6);

PE — Provider Edge;

CE — Customer Edge;

TE — Traffic Engineering.

ВСТУП

Актуальність теми. Оператори провідного та мобільного зв'язку, оператори корпоративних мереж та постачальники хмарних послуг прагнуть, щоб їх послуги відповідали складним вимогам, що виникають у зв'язку з розвитком IP-мереж (наприклад, величезні вимоги до смуги пропускання, інтеграція мільярдів пристроїв та мільйонів хмарних сервісів). Запропонована на початку 2010-х років архітектура сегментної маршрутизації (SR) допомагає впоратися з цими складними вимогами, і зараз вона активно впроваджується в світі. Архітектура SR заснована на концепції вихідної маршрутизації і має хороші властивості масштабованості, оскільки вона значно скорочує обсяг інформації про стан, яку необхідно налаштувати в основних вузлах для підтримки складних сервісів. Архітектура SR була вперше реалізована в площині MPLS даних, а потім в площині даних IPv6 (SRv6). Архітектура IPv6 SR (SRv6) була розширена від простого управління пакетами між вузлами до застосування підходу мережного програмування, що робить її дуже підходящою для таких випадків використання, як об'єднання сервісних функцій та віртуалізація мережевих функцій.

Мета і завдання дослідження.

Мета дослідження — обґрунтувати доцільність використання технології Segment Routing для оптимізації маршрутизації.

Для досягнення цієї мети поставлені наступні завдання:

1. Дослідити які оптимізації маршрутизації вносить Segment Routing в мережу.

2. Розглянути можливості реалізацій різних видів мереж на основі Segment Routing.

3. Оцінити перспективність розвитку технології Segment Routing в різних сферах застосування.

4. Виділити переваги застосування Segment Routing та порівняти їх з подібними технологіями.

5. Провести порівняння двох реалізацій Segment Routing (SR-MPLS та SRv6).

Об`єкт дослідження — маршрутизація трафіку в мережах.

Предмет дослідження — оптимізація маршрутизації в мережах завдяки впровадженню технології Segment Routing.

Методи дослідження — емпіричні, аналіз літературних джерел, методи проектування мереж.

Джерела дослідження — документи з описами стандартів технологій, наукові публікації та навчальні посібники.

Наукова новизна одержаних результатів. В роботі розглядаються новітні технології такі як SR IPv6 та SFC та їх використання в SDN мережах. Крім цього розглядаються найновіші варіанти застосування Segment Routing для розгортання різних мереж (WAN, DC, CDN). Багато сучасних мереж використовують застарілі технології (в тому числі в цілях оберненої сумісності), а тому описані в роботі технології можуть в майбутньому замінити їх з метою оптимізації.

Практичне значення отриманих результатів. Отримані результати дозволяють оцінити вигоду від побудови нових мереж з застосуванням Segment Routing, або міграції старих мереж на цю технологію. Дані результати дослідження можуть використовуватись, як для розробки архітектури провайдерської мережі так і мережі підприємства.

Апробація роботи. За результатом магістерського дослідження опубліковано праці в наукових фахових журналах (Дод. А).

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ РОЗГОРТАННЯ SEGMENT ROUTING

1.1. Протокол внутрішнього шлюзу Interior Gateway Protocol

Внутрішні шлюзові протоколи, або IGP (Interior Gateway Protocol), відіграють ключову роль у забезпеченні маршрутизації даних у межах чітко окресленого домену. Таким доменом виступає автономна система — об'єднання мереж під єдиним адміністративним управлінням, що є характерним для великих організацій, освітніх кампусів чи корпоративних структур. Застосування IGP дозволяє ефективно прокладати маршрути як на рівні всієї автономної системи, так і всередині кожної окремої локальної мережі, що до неї входить.

Класифікація протоколів внутрішнього шлюзу базується на алгоритмах їх роботи та поділяється на дві основні категорії:

1. Протоколи, що спираються на вектор відстані (Distance-vector routing protocols).
2. Протоколи, що базуються на стані каналів зв'язку (Link-state routing protocols)

У контексті сучасних рішень, таких як SRv6 (Segment Routing over IPv6), перевага надається саме протоколам стану каналу, зокрема OSPF та IS-IS. Вони забезпечують детальне розповсюдження інформації про топологію мережі, що є необхідним для ідентифікації адрес вузлів та каналів. Ці дані згодом інтегруються в SRH (Segment Routing Header) для точного керування трафіком.

Протокол IS-IS IS-IS

Протокол IS-IS (Intermediate System to Intermediate System) представляє собою класичний приклад технології внутрішнього шлюзу, що оперує станами каналів зв'язку для побудови оптимальної маршрутної карти. В його основі лежить алгоритмічний базис SPF (Shortest-Path-First — пошук найкоротшого шляху), який забезпечує математичну базу для вибору маршрутів. Важливою особливістю IS-IS є інтелектуальна обробка топологічних змін: система диференціює необхідність

повного перерахунку SPF або ж обмежується частковим розрахунком маршруту (PRC), що значно економить обчислювальні ресурси. Історично цей стандарт розроблявся для середовища ISO CLNP, проте його принципи, зокрема використання Hello-пакетів для оперативного виявлення сусідів та моніторингу стану мережі, споріднюють його з OSPF.

Унікальна архітектура адресації, базована на стандартах ISO, забезпечує уніфікацію налаштувань для різних версій інтернет-протоколу. Завдяки цій інваріантності, підходи до конфігурування як IPv4, так і IPv6 у середовищі IS-IS є фактично ідентичними, що спрощує адміністрування гетерогенних мереж.

Структурно мережа IS-IS розглядається як єдина автономна система, що об'єднує два ключові типи елементів: кінцеві системи (End Systems, ES) — виступають генераторами та споживачами трафіку, і проміжні системи (Intermediate Systems, IS). Останні — це термін OSI, що відповідає сучасним маршрутизаторам, які відповідають за транзит та ретрансляцію пакетів, що технічно іменуються мережевими PDU (Protocol Data Units).

Для забезпечення масштабованості та адміністративної зручності застосовується ієрархічна декомпозиція домену на окремі області. Цей підхід реалізується через дворівневу логіку маршрутизації:

1. Системи першого рівня (Level 1) обслуговують трафік виключно в межах своєї локальної області
2. Системи другого рівня (Level 2) формують магістраль, маршрутизуючи трафік між областями та до зовнішніх автономних систем

Якщо адресат знаходиться поза межами локального сегмента, передача даних від L1-системи делегується L2-системі. Важливо зазначити, що у IS-IS, на відміну від OSPF, відсутнє жорстке поняття виділеної "головної" області.

Функціональний розподіл баз маршрутизації наступний: маршрутизатори першого рівня синхронізують інформацію про внутрішньообласні маршрути, тоді як маршрутизатори другого рівня обмінюються міждласною інформацією про

доступні IP-адреси. Маршрутизатор може виконувати функцію як L1, так і L2 одночасно (L1/L2-маршрутизатор), забезпечуючи повне розділення внутрішніх та міждоменних маршрутних даних.

Поширення оновлень стану каналу (Link-State Updates) чітко обмежується межами одного рівня. Кожен маршрутизатор у межах свого рівня підтримує повну базу даних стану каналів усіх інших пристроїв на тому ж рівні. На основі цієї повної бази даних, кожен пристрій застосовує алгоритм Дейкстри для визначення оптимального найкоротшого шляху від себе до будь-якої іншої точки у домені.

Для забезпечення свого функціоналу — встановлення сусідства, обміну топологічною інформацією та підтримки цілісності баз даних — IS-IS використовує спеціалізовані Протокольні Одиниці Даних (PDU). PDU забезпечують ефективний та надійний обмін топологічною інформацією в середовищі IS-IS і поділяються на чотири основні типи:

- IS-IS Hello PDU (IIH) — ці пакети розсилаються усім пристроям в мережі (широкомовна розсилка) для оперативного виявлення сусідніх систем та їх класифікації за рівнями маршрутизації (L1 або L2)

- Link-State PDU (LSP) — містять ключову інформацію про стан каналів, включно із відомостями про сусідні пристрої та їхню досяжність. LSP є основою для побудови бази даних топології

- Complete Sequence Number PDU (CSNP) — містять повний контрольний список усіх LSP, наявних у базі даних IS-IS. Вони використовуються для початкової синхронізації та періодичного підтвердження цілісності інформації

- Partial Sequence Number PDU (PSNP) — відправляються адресатом (багатоадресною розсилкою) у відповідь на виявлення розбіжностей чи застарілих записів у власній базі даних стану каналів, що сигналізує про необхідність отримання відсутніх або оновлених LSP

Протокол внутрішнього шлюзу OSPF

Протокол OSPF (Open Shortest Path First) є одним із ключових протоколів внутрішнього шлюзу (IGP), що використовується для маршрутизації IP-пакетів виключно в межах однієї автономної системи (АС). Подібно до IS-IS, OSPF базується на моделі стану каналу (link-state) та застосовує алгоритм пошуку найкоротшого шляху (SPF) для обчислення оптимальних маршрутів. Кожен маршрутизатор у домені OSPF генерує та поширює Повідомлення про стан каналу (LSA) по всій АС або її конкретній області. Ці повідомлення містять критичні дані про підключені до маршрутизатора інтерфейси та їхні метрики вартості, на основі яких кожен пристрій незалежно обчислює найкоротші шляхи до всіх мереж і формує власну таблицю маршрутизації OSPF.

OSPF був спеціально розроблений для протоколу TCP/IP, що забезпечує його нативну підтримку IP-підмереж, а також можливість маркування маршрутної інформації, імпортованої із зовнішніх джерел. Протокол включає вбудовані механізми автентифікації оновлень маршрутизації, що підвищує безпеку. OSPF ухвалює рішення про маршрутизацію IP-пакетів виключно на підставі адреси одержувача, що міститься у заголовку IP-пакета. Однією з його ключових переваг є швидке виявлення топологічних змін (наприклад, збою інтерфейсу) та оперативне обчислення нових, безаварійних маршрутів, мінімізуючи при цьому обсяг службового трафіку.

Архітектура автономної системи OSPF може бути реалізована як із однією областю, так і з ієрархічною структурою, розділеною на декілька областей. У однообласній конфігурації кожен маршрутизатор зберігає повну базу даних, що детально описує всю топологію АС, і LSA розсилаються по всьому домену. У багатообласній топології маршрутизатори підтримують зведену (агреговану) топологію інших областей, зберігаючи при цьому ідентичні деталізовані топологічні бази даних лише для своєї локальної області. OSPF гарантує швидке

оновлення вмісту топологічних баз даних усіх маршрутизаторів при будь-якій зміні в АС чи конкретній області.

Зовнішня маршрутна інформація (наприклад, маршрути, отримані через BGP) передається через АС OSPF прозоро і зберігається окремо від внутрішніх даних стану каналу. Кожен такий зовнішній маршрут може бути позначений унікальною міткою (tag), що слугує механізмом для передачі додаткової адміністративної або політичної інформації між маршрутизаторами, розташованими на межі автономної системи.

Всі маршрутизаційні пристрої в межах однієї області виконують алгоритм SPF паралельно, незалежно один від одного, та зберігають результати обчислень у своїх індивідуальних топологічних базах даних. Маршрутизатори, що мають інтерфейси, підключені до кількох областей (так звані ABR), повинні виконувати розрахунок алгоритму SPF багаторазово — по одному разу для кожної приєднаної області.

Усі маршрутизаційні пристрої, що належать до однієї області, ініціюють виконання алгоритму SPF (Shortest Path First) паралельно, працюючи незалежно. Кожен пристрій зберігає результати цих обчислень у своїй індивідуальній топологічній базі даних. Маршрутизатори, які мають підключення до кількох областей в межах однієї АС (так звані ABR), відповідно, повинні повторювати розрахунок алгоритму SPF багаторазово — по одному разу для кожної області, до якої вони належать.

Коли маршрутизаційний пристрій активується, він спочатку ініціалізує конфігурацію OSPF і очікує підтвердження від протоколів нижчих рівнів, що його фізичні інтерфейси функціонують належним чином. Після цього маршрутизатор використовує протокол привітання OSPF (Hello Protocol) для оперативного виявлення сусідніх пристроїв у мережі.

Наступним критично важливим етапом є спроба пристрою сформувати відносини сусідства (adjacency) з деякими зі своїх виявлених сусідів. Саме ця

суміжність визначає механізм розподілу пакетів протоколу маршрутизації: службові пакети OSPF та оновлення топологічної бази даних передаються та приймаються лише між тими сусідами, які утворили суміжність. Після успішного встановлення суміжності, сусідні маршрутизатори об’єднуються у пари та розпочинають процес синхронізації своїх топологічних баз даних.

Для інформування про свій поточний статус або у випадках зміни стану маршрутизатори періодично надсилають пакети LSA (Link-state advertisement). Ці пакети, окрім іншого, містять інформацію про сусідні пристрої, що дозволяє системі швидко виявляти нефункціонуючі маршрутизатори у мережі, забезпечуючи відмовостійкість.

Маршрутизатори розсилають LSA по всьому периметру області, що є гарантією того, що абсолютно всі маршрутизаційні пристрої в цьому сегменті мають ідентичну топологічну базу даних. Використовуючи цю консолідовану інформацію, кожен пристрій самостійно обчислює дерево найкоротших шляхів (Shortest Path Tree) у мережі. Це дерево потім слугує основою для таблиці маршрутизації, яка використовується для комутації мережевого трафіку.

Граничні маршрутизатори автономної системи (ASBR) відповідають за розповсюдження інформації про зовнішні автономні системи по всій своїй АС, за винятком тих сегментів, які визначені як тупикові області (stub areas). Граничні маршрутизатори області (ABR) несуть відповідальність за оголошення внутрішньообласних шляхів до всіх ASBR у домені.

1.2 Протокол прикордонного шлюзу Border Gateway Protocol

Протоколи зовнішнього шлюзу (EGP, Exterior Gateway Protocol) — це клас IP-маршрутизаційних протоколів, ключовим призначенням яких є обмін маршрутною інформацією між різними автономними системами (АС). Ця функція є абсолютно критичною для забезпечення глобального зв'язку в мережі Інтернет. Серед відомих представників цієї категорії слід виділити застарілий однойменний

протокол EGP, а також сучасний і домінуючий Протокол прикордонного шлюзу (BGP, Border Gateway Protocol).

BGP є чинним протоколом зовнішнього шлюзу, який спеціально розроблений для обміну маршрутною інформацією між маршрутизаторами, розташованими у різних автономних системах. Маршрутна інформація BGP є всеосяжною, оскільки вона містить повний шлях до кожного мережевого призначення. BGP підтримує власну базу даних маршрутизації, якою він обмінюється з іншими BGP-системами. Використовуючи дані про доступність мереж, BGP будує граф зв'язності АС, що є фундаментальним для виявлення та усунення маршрутних петель, а також для застосування політик маршрутизації на рівні автономних систем.

Розширення Multiprotocol BGP (MBGP) було розроблене для забезпечення повноцінної підтримки IPv6. MBGP запроваджує два спеціальні атрибути: MP_REACH_NLRI та MP_UNREACH_NLRI, які використовуються для передачі інформації про доступність IPv6-адрес. Повідомлення оновлення інформації про доступність мережевого рівня (NLRI, Network Layer Reachability Information) містять префікси IPv6-адрес, які ідентифікують потенційні маршрути.

BGP надає адміністраторам унікальну можливість реалізовувати маршрутизацію з урахуванням політик. Ці політики можуть бути використані для інтелектуального вибору найкращого шляху серед кількох альтернативних до певного пункту призначення, а також для ефективного управління перерозподілом трафіку. Як транспортний протокол BGP використовує TCP, встановлюючи з'єднання через виділений порт 179.

BGP версії 4 (BGPv4) приніс революційну підтримку бескласової міждоменної маршрутизації (CIDR), що повністю ліквідувала застарілу концепцію мережевих класів. Замість того, щоб неявно визначати адресу мережі на основі першого октету, CIDR дозволяє явно вказувати кількість бітів у мережевій адресі. Це нововведення не лише забезпечило ефективніше використання адресного

простору, а й дозволило зменшити розміри таблиць маршрутизації. BGPv4 також підтримує механізми агрегації маршрутів, включно з агрегацією шляхів AS.

Інформація, якою обмінюються BGP-системи, є дуже деталізованою: вона містить повний маршрут до кожного пункту призначення, а також додаткові відомості про цей маршрут. Цей маршрут до кінцевого призначення називається AS-шляхом (AS-Path), а додаткові дані інкапсулюються в атрибути шляху і передаються разом з AS-шляхом. BGP використовує AS-шлях та атрибути шляху для повного визначення топології мережі. Після того, як BGP повністю розуміє топологію, він здатний не лише виявляти та усувати петлі маршрутизації, але й обирати з доступних варіантів маршрути, які найкраще відповідають адміністративним вподобанням та політикам маршрутизації.

BGP розрізняє два основні сценарії обміну маршрутною інформацією:

- Зовнішній BGP (EBGP) використовується для пересилання даних між різними автономними системами.
- Внутрішній BGP (IBGP) застосовується для обміну маршрутами всередині однієї автономної системи.

1.3. Багатопротокольна комутація за мітками MultiProtocol Label Switching

Технологія MPLS (Multiprotocol Label Switching), стандартизована IETF, є протоколом, який використовує механізм міток для управління високошвидкісною пересилкою пакетів даних поверх існуючого IP-протоколу. MPLS свого часу виник як значно ефективніша альтернатива традиційній IP-маршрутизації. Принцип його роботи полягає у присвоєнні коротких, фіксованої довжини міток локального значення IP-адресам пристроїв. Це дозволяє замінити ресурсомісткий пошук у великій IP-таблиці на простіший та швидший пошук у таблиці міток. Крім того, мітки MPLS можуть бути використані для створення логічних тунелів усередині IP-мережі. MPLS демонструє сумісність із численними протоколами мережевого та

канального рівнів, що забезпечує його універсальність для побудови загальнодоступних мережевих тунелів для різноманітних служб — L2VPN, L3VPN та EVPN.

Порівняно з класичною IP-маршрутизацією, MPLS забезпечив значне покращення продуктивності пересилання даних.

Основний приріст ефективності полягає у заміні трудомісткого пошуку в таблиці IP-маршрутизації на просту операцію заміни міток (label swapping), що суттєво скорочувало час, необхідний для обробки пакетів. Після того, як пакет потрапляє в домен MPLS, лише пристрої на вході (LER) та на виході домену повинні аналізувати повні заголовки IP-пакетів для інкапсуляції або декапсуляції міток. Всі проміжні (транзитні) вузли лише виконують операції заміни міток. Проте, з розвитком технологій ASIC (спеціалізованих інтегральних схем) та апаратного забезпечення, які значно прискорили обробку пошуку в таблицях IP-маршрутизації, MPLS втратив свою помітну перевагу у швидкості пересилання трафіку в сучасних IP-мережах.

Незважаючи на втрату переваги у швидкості, MPLS зберігає свою важливість завдяки іншій меті: його механізм переадресації на основі міток по суті є ефективною технологією тунелювання, яка підтримує багаторівневу інкапсуляцію міток (label stacking). Крім того, MPLS відмінно функціонує у мережах із кількома протоколами мережевого та канального рівнів, що робить його ідеальним для побудови універсальних мережевих тунелів у різноманітних VPN-сервісах. Оскільки пересилання пакетів у MPLS залежить від операції заміни міток однакової довжини, MPLS є технологією пересилання, що орієнтована на з'єднання пристроїв. Це забезпечило його широке застосування в таких критичних областях, як керування трафіком (TE) та забезпечення якості обслуговування (QoS).

MPLS є технологією переадресації, яка ґрунтується на Класах еквівалентності пересилання (FEC, Forwarding Equivalence Class). MPLS класифікує всі пакети даних, які можуть бути переслані однаковим чином, до

одного класу FEC. Пакети, що належать до того самого FEC, обробляються ідентично. Класифікація пакетів за FEC може ґрунтуватися на будь-якій комбінації таких елементів: адреса джерела, адреса призначення, порт джерела, порт призначення, тип протоколу, тип служби тощо. Усі IP-пакети з однаковою адресою призначення, використовуючи правило найбільшого збігу префіксів, будуть класифіковані як один і той же FEC.

Мітки MPLS

Мітка MPLS є коротким ідентифікатором фіксованої довжини, значення якого є локальним і дійсним лише в межах конкретного сегмента мережі. Ця мітка виконує функцію однозначної ідентифікації того Класу еквівалентності пересилання (FEC), до якого належить пакет. Слід зауважити, що у певних сценаріях, наприклад, при необхідності реалізації балансування навантаження (Load Balancing), один і той самий FEC може бути асоційований із кількома мітками MPLS. Технічно мітка MPLS є чотирибайтовим полем (32 біти), яке інкапсулюється у заголовок пакета.

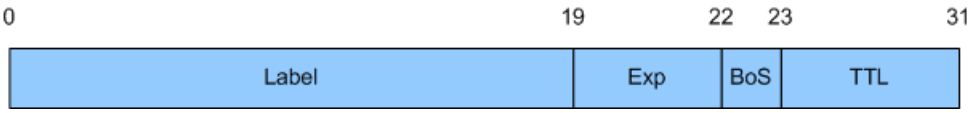


Рис. 1.1 – Структура мітки MPLS

Мітка MPLS складається з чотирьох основних полів, кожне з яких виконує свою специфічну функцію в процесі пересилання трафіку:

- Мітка (Label) — 20-бітове поле, що містить безпосередньо ідентифікатор мітки, який унікально визначає FEC пакета.

- Exp (Experimental) — 3-бітове поле, первісно зарезервоване для експериментальних розширень. На сьогодні це поле найчастіше використовується для ідентифікації Класу обслуговування (CoS, Class of Service), що є критично важливим для забезпечення QoS.

- BoS (Bottom of Stack) — 1-бітове поле, яке слугує індикатором кінця стека міток. Оскільки MPLS підтримує множинні мітки (або вкладеність), встановлення значення 1 у полі BoS свідчить про те, що ця мітка є останньою у стеку.

- TTL (Time to Live) — 8-бітове поле, що вказує на значення часу життя пакета, і його функціонал повністю відповідає полю TTL у заголовках IP-пакетів, запобігаючи зацикленню пакетів.

Ця структура мітки MPLS розміщується у пакеті шляхом інкапсуляції — вона вставляється між канальним рівнем (L2) та мережевим рівнем (L3). Така інкапсуляція забезпечує сумісність міток MPLS практично з усіма протоколами канального рівня.



Рис. 1.2 – Структура пакету з MPLS міткою

Стек міток MPLS також прийнято називати багаторівневою міткою MPLS (MPLS Label Stack). Він являє собою упорядковану послідовність, що складається з кількох міток MPLS. Мітка, яка розташована безпосередньо поряд із заголовком Канального рівня (L2), технічно називається міткою "поверх стека" або зовнішньою міткою. Натомість, мітка, що знаходиться поруч із заголовком IP-протоколу, відома як мітка "внизу стека" або внутрішня мітка. Процес обробки міток відбувається, починаючи з вершини стека (зовнішньої мітки) і виконується за принципом LIFO (Last-In-First-Out — останній зайшов, перший вийшов).

Операції з мітками MPLS

Ключові процедури, що виконуються над мітками MPLS для управління трафіком, включають їхнє встановлення, заміну та витягнення.

Push (додавання) — ця операція виконується, коли IP-пакет входить до домену MPLS. Мітка додається у пакет, розміщуючись між заголовком Канального рівня (L2) та IP-заголовком. Під час проходження транзитного вузла (LSR), цей

вузол також має можливість додати мітку у верхню частину стека (механізм вкладання міток).

Swap (заміна) — під час пересилання пакета всередині домену MPLS транзитний вузол здійснює пошук у своїй LFIB (Label Forwarding Information Base) і замінює мітку, що знаходиться зверху стека MPLS-пакета, на нову мітку, призначену для наступного переходу.

Pop (витягнення) — ця дія виконується, коли пакет залишає домен MPLS, що передбачає видалення останньої (зовнішньої) мітки. Крім того, витягнення мітки може бути виконане на передостанньому переході з метою зменшення кількості міток у стеку.

Оскільки мітки MPLS вже не потрібні вихідному маршрутизатору, що залишає мережу MPLS, можна застосувати оптимізацію — видалення на передостанньому переході (PHP, Penultimate Hop Popping). Цей механізм дозволяє маршрутизатору, розташованому на передостанньому вузлі, витягнути зовнішню мітку з пакета MPLS. Це дає можливість вихідному маршрутизатору безпосередньо переслати пакет, орієнтуючись на IP-адресу або на основі внутрішньої мітки (якщо стек ще не порожній), тим самим суттєво знижуючи навантаження на обробку пакетів на вихідному пристрої.

Реалізація PHP досягається шляхом присвоєння спеціальному значенню "3" у полі мітки. Мітка зі значенням "3" інтерпретується як неявно порожня мітка (Implicit Null Label), яка ніколи не з'являється у фактичному стеку міток. Коли маршрутизатор на передостанньому переході ідентифікує, що верхня мітка має значення "3", він не виконує операцію Swap (заміну), а замість цього негайно виконує операцію Pop (витягнення).

Шлях комутації за мітками LSP

Маршрут комутації міток (LSP, Label Switched Path) є заздалегідь визначеним шляхом у домені MPLS, яким послідовно пересилаються пакети, що належать до одного й того самого Класу еквівалентності пересилання (FEC). Це

логічний тунель, яким рухаються пакети, інкапсульовані мітками MPLS, від вхідного прикордонного маршрутизатора до вихідного прикордонного маршрутизатора домену. Саме LSP формує основу для орієнтованої на з'єднання природи переадресації в MPLS, гарантуючи, що всі пакети одного FEC будуть оброблені послідовно однаковим чином по всьому шляху.

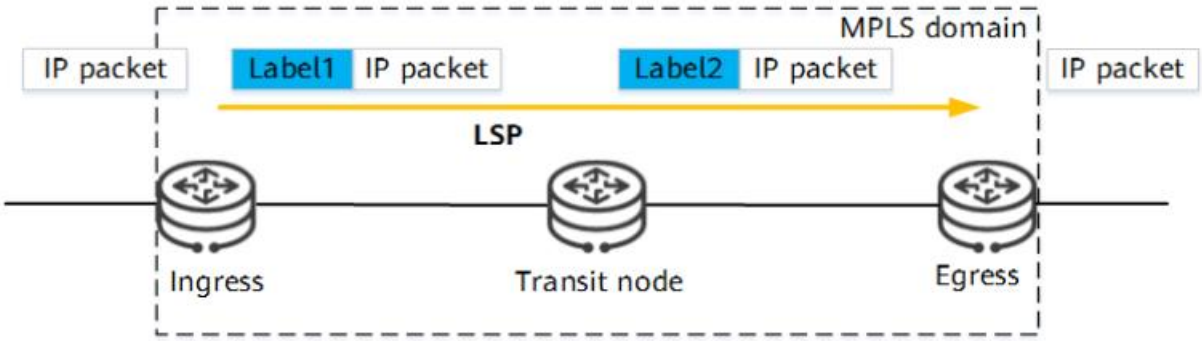


Рис. 1.3 – LSP

Маршрут комутації міток (LSP) являє собою односпрямований канал передачі даних, який пролягає від вхідного прикордонного пристрою до вихідного пристрою. Уздовж цього маршруту різні вузли виконують специфічні ролі, необхідні для функціонування технології:

Вхідний вузол (Ingress Label Edge Router, LER) — це початкова точка LSP, яка може бути лише одна. На цьому вузлі відбувається ключова операція: вхідний вузол додає (Push) мітку MPLS до IP-пакета та інкапсулює його у відповідний MPLS-пакет.

Транзитний вузол (Label Switching Router, LSR) — це проміжний елемент LSP, кількість яких може бути від нуля до будь-якої необхідної кількості. Транзитний вузол використовує свою LFIB (Label Forwarding Information Base) для виконання операції заміни міток (Swap) та пересилання MPLS-пакетів до наступного вузла.

Вихідний вузол (Egress LER) — це кінцева точка LSP, яка також є унікальною. Завдання цього вузла — видалити (Pop) останню мітку з MPLS-пакета,

повністю відновлюючи його початковий вигляд (декапсуляція) перед тим, як відправити його далі.

Мережа MPLS

Наведена структура мережі (рис. 1.4) демонструє типову архітектурну структуру мережі, що використовує технологію MPLS. Така структура включає прикордонні маршрутизатори міток (LER) на вході та виході домену, а також внутрішні транзитні маршрутизатори міток (LSR), які спільно формують маршрути комутації міток (LSP).

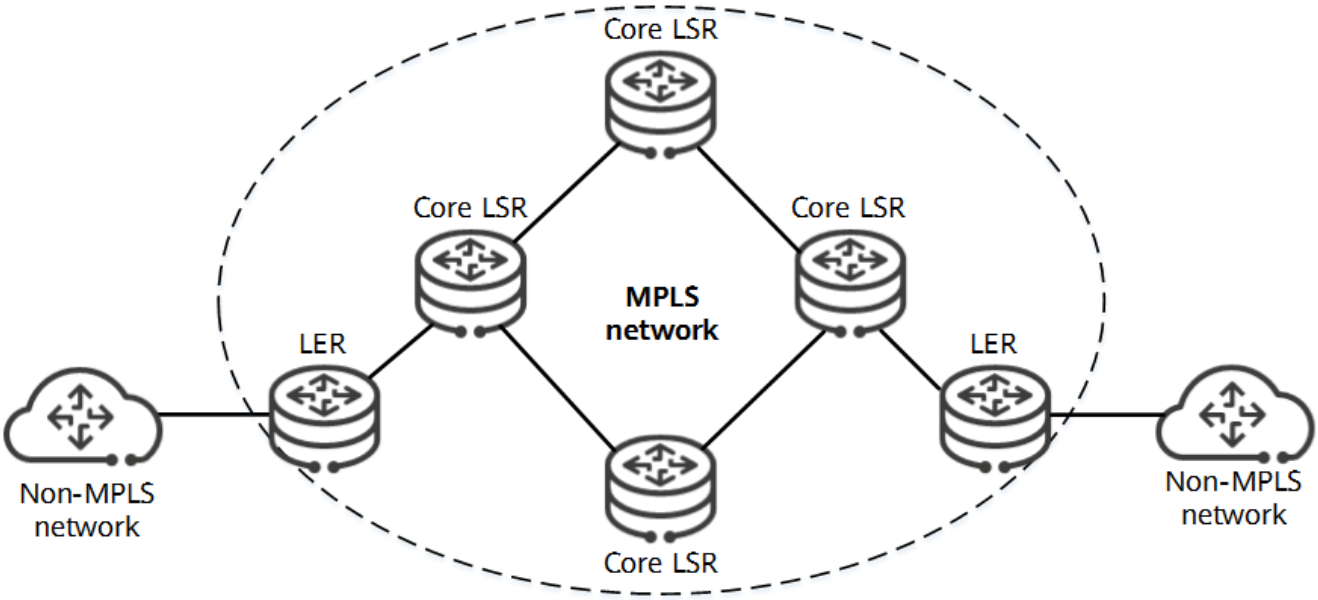


Рис. 1.4 – Структура мережі MPLS

Мережева інфраструктура MPLS складається з трьох ключових елементів, які спільно формують її функціональний домен.

Маршрутизатор комутації міток (LSR, Label Switching Router) — це мережевий пристрій, обладнаний функціоналом MPLS, що становить основу мережі. Сукупність LSR формує домен MPLS.

Ядровий LSR (Core LSR) — це маршрутизатор, який розміщений всередині домену MPLS і підключається виключно до інших LSR, що знаходяться у межах цього самого домену.

Прикордонний маршрутизатор міток (LER, Label Edge Router) — це маршрутизатор, який розташований на границі домену MPLS і має з'єднання з одним або кількома вузлами, що не підтримують протокол MPLS.

У мережі MPLS, Маршрут комутації міток (LSP) може бути встановлений між будь-якими двома LER з метою пересилання пакетів, які входять у домен MPLS. Під час проходження цим шляхом пакети можуть транзитом проходити через один або кілька Core LSR. Таким чином, вхід та вихід LSP формуються LER, а транзитні вузли на цьому шляху складаються з Core LSR.

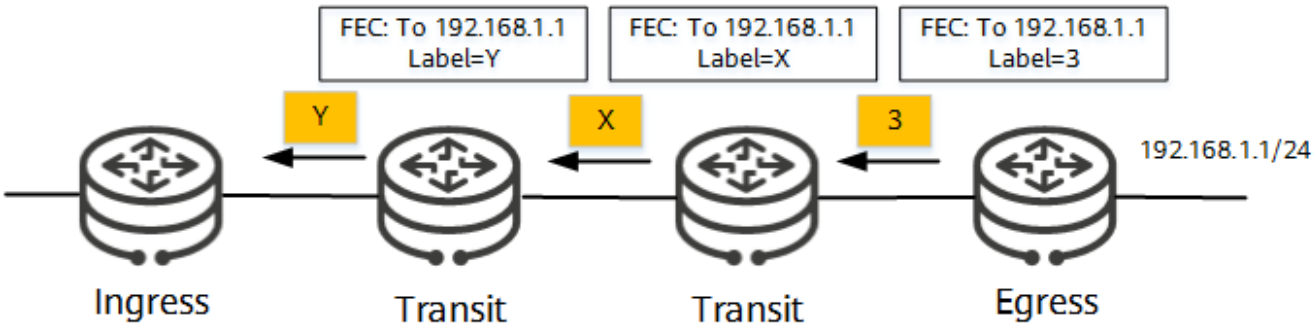


Рис. 1.5 – Процес встановлення LSP

Широке застосування технології MPLS у сучасних IP-мережах обґрунтовується низкою значущих переваг, які вона пропонує:

- Висока адаптивність — MPLS є повністю сумісною з IP-мережею і фактично виступає її функціональним розширенням, що спрощує його впровадження та характеризується низькими початковими витратами на інтеграцію.

- Архітектурний поділ — MPLS реалізує чітке розділення рівнів керування та даних. На рівні керування Маршрути комутації міток (LSP) конфігуруються на основі існуючих IP-маршрутів, що дозволяє технології запозичувати механізми надійності та гнучкості IP. Натомість, на рівні даних, що є орієнтованим на з'єднання, пакети передаються через встановлені LSP. Це дозволяє MPLS ефективно реалізовувати Керування трафіком (TE) та Якість обслуговування (QoS).

- Сумісність протоколів — MPLS зберігає певну залежність від протоколів канального рівня (L2), підтримуючи такі стандарти, як Frame Relay, ATM, PPP та SDH. Ця властивість гарантує взаємодію між різними типами мереж та забезпечує високий рівень сумісності.

- Масштабованість — мережа MPLS має вбудовану підтримку ієрархічної топології, що робить її ідеальною для розгортання як накладна технологія (overlay) поверх існуючої IP-інфраструктури.

- Підтримка VPN — з теоретичної точки зору, стек міток MPLS підтримує необмежену глибину вкладеності міток. Ця особливість повністю задовольняє вимоги VPN-сервісів до багаторівневої інкапсуляції міток, необхідної для розмежування трафіку загальнодоступних і приватних мереж, забезпечуючи потужну платформу для розробки складних VPN-рішень.

MPLS — це технологічний підхід, що використовує ідентифікатори-мітки для управління процесом пересилання пакетів у мережі. Створення LSP (Label Switched Path) є фундаментальним процесом, у ході якого Маршрутизатори комутації міток (LSR) уздовж майбутнього шляху визначають та асоціюють відповідні мітки для конкретного Класу еквівалентності пересилання (FEC).

Призначення та подальше розподілення міток MPLS відбувається у зворотному напрямку щодо потоку даних — від кінцевого LSR (Egress) до початкового LSR (Ingress). Як ілюструє рисунок 1.5, початковий LSR спершу класифікує FEC мітки, орієнтуючись на IP-адресу отримувача. Далі нижчестоящий LSR (ближчий до кінця) надсилає мітки вищестоящому LSR (ближчому до початку), ініціюючи таким чином процес встановлення LFIB (Label Forwarding Information Base) у вищестоящих вузлах. Зрештою, послідовність з'єднаних LSR утворює LSP. Залежно від методу встановлення, LSP поділяються на дві категорії:

1. Статичні LSP — встановлюються через ручне, адміністративне конфігурування та занесення міток, асоційованих із FEC. При такому ручному

призначенні обов'язковою є умова, що значення вихідної мітки на вищому LSR має ідентично збігатися зі значенням вхідної мітки на нижчестоящому LSR.

2. Динамічні LSP — встановлюються автоматично, коли LSR використовують протокол розподілу міток для динамічного створення та обміну мітками. Початковий LSR використовує інформацію з IP-маршрутів як основу для надсилання міток до вищестоящих вузлів. MPLS підтримує кілька протоколів для динамічного розподілу міток: протокол розподілу міток (LDP); протокол резервування ресурсів з розширеннями для управління трафіком (RSVP-TE); багатопротокольні розширення для протоколу граничного шлюзу (MP-BGP).

Пересилка пакетів по LSP

Розглянемо детально процес пересилання пакетів у межах маршруту комутації міток (LSP), який був сконфігурований із залученням механізму видалення на передостанньому переході (PHP, Penultimate Hop Popping). Наведений приклад (рис. 1.6) ілюструє, як вхідний маршрутизатор (LER) інкапсулює IP-пакет міткою, як транзитні маршрутизатори (LSR) виконують операцію Swap, і як передостанній LSR використовує неявну порожню мітку (значення "3") для виконання операції Pop, тим самим звільняючи вихідний маршрутизатор від потреби обробляти зовнішню мітку.

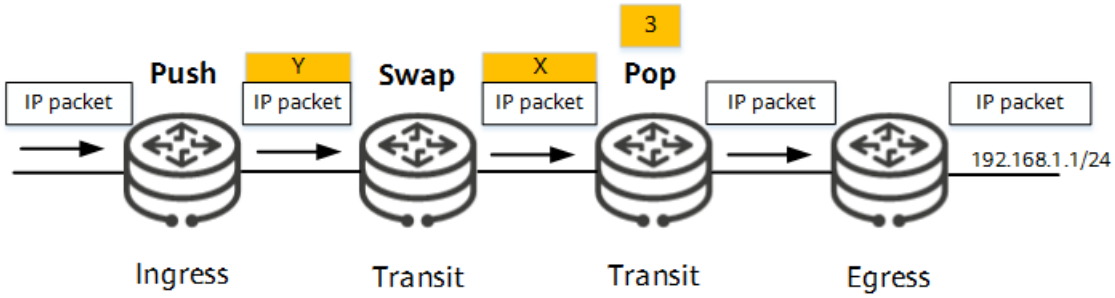


Рис. 1.6 – Пересилання пакетів через LSP

Пересилання пакетів MPLS уздовж Маршруту комутації міток (LSP) у випадку, коли активовано механізм PHP (Penultimate Hop Popping), відбувається за такою послідовністю дій.

1. Після отримання вхідного IP-пакета, призначеного для мережі 192.168.1.1/24, вхідний вузол (LER) виконує операцію Push, вставляючи у пакет мітку Y. Далі він інкапсулює його у формат MPLS-пакета та відправляє його далі по LSP.

2. Перший транзитний вузол (LSR) здійснює операцію Swap, замінюючи мітку Y на нову мітку X, відповідно до своєї LFIB, і пересилає пакет до наступного вузла.

3. Транзитний вузол, що знаходиться на передостанньому переході, отримує MPLS-пакет і виконує операцію Pop — він витягує мітку X з пакета. Це відбувається тому, що мітка X має спеціальне значення "3" (Implicit Null Label), що сигналізує про необхідність видалення мітки перед передачею на вихідний вузол. Після цього транзитний вузол пересилає вже IP-пакет на вихідний вузол.

4. Після отримання чистого IP-пакета без зовнішньої мітки, вихідний вузол (LER) може негайно переслати його безпосередньо на адресу призначення 192.168.1.1/24.

Протокол розподілу міток Label Distribution Protocol

Протокол розповсюдження міток (LDP, Label Distribution Protocol) є сигнальним протоколом, призначеним для автоматичного розподілу міток у застосунках, які не орієнтовані на керування трафіком (non-TE-aware). LDP надає маршрутизаторам механізм для ефективного встановлення Шляхів комутації міток (LSP) у мережі шляхом прямого зіставлення інформації про маршрутизацію мережевого рівня (L3) зі шляхами, сформованими на рівні каналу передачі даних (L2).

Створені за допомогою LDP маршрути LSP можуть мати свою кінцеву точку як на безпосередньо підключеному сусіді, так і на вихідному мережевому вузлі (Egress LER), що дозволяє гнучко спрямовувати пересилання трафіку через будь-які проміжні маршрутизатори. Важливою архітектурною особливістю є те, що LSP, встановлені за допомогою LDP, можуть також перетинатися зі LSP, що були

згенеровані іншим сигнальним протоколом — RSVP (Resource Reservation Protocol).

LDP асоціює унікальний Клас еквівалентності пересилання (FEC) із кожним створюваним LSP. Цей пов'язаний FEC визначає, які саме пакети будуть прив'язані до даного LSP. Шляхи LSP розширюються по мережі в міру того, як кожен маршрутизатор вибирає мітку, оголошену наступним переходом для даного FEC, і пов'язує її з міткою, яку він, у свою чергу, оголошує всім іншим LSR. Цей ітеративний, низхідний (від Egress) процес зрештою формує дерево LSP-шляхів, що сходяться на вихідному маршрутизаторі.

LDP є сигнальним протоколом, який функціонує на пристроях, що підтримують MPLS. Успішна конфігурація MPLS та LDP ініціює обмін пакетами для встановлення TCP-з'єднань через інтерфейси LDP. Ці TCP-з'єднання використовуються для встановлення сеансів LDP, через які відбувається обмін усією необхідною інформацією MPLS у мережі. Просте включення MPLS та LDP на відповідних інтерфейсах маршрутизаторів є достатньою умовою для автоматичного встановлення LSP у всій мережі.

Протокол резервування ресурсів Resource Versation Protocol

Протокол RSVP (Resource Reservation Protocol) є сигнальним протоколом, який використовується маршрутизаторами для передачі запитів на якість обслуговування (QoS) та для встановлення й подальшої підтримки робочого стану певних мережевих послуг. Запити, ініційовані RSVP, як правило, призводять до резервування ресурсів (таких як смуга пропускання або буферна пам'ять) на кожному вузлі, що входить до шляху передачі даних.

Ключові атрибути, що характеризують протокол RSVP, включають:

1. Резервування ресурсів здійснюється виключно для односпрямованих потоків даних.

2. Протокол дозволяє одержувачу потоку даних ініціювати та підтримувати необхідне резервування ресурсів, які будуть використовуватися для цього потоку.

3. RSVP підтримує гнучкий стан як на маршрутизаторах, так і на хостах, що забезпечує підтримку динамічних змін у топології мережі та автоматичну адаптацію до змін у маршрутизації.

4. RSVP залежить від існуючих та майбутніх протоколів маршрутизації, але сам по собі не є протоколом маршрутизації; його завдання — лише резервування.

5. Протокол пропонує кілька моделей або стилів резервування, що дозволяє адаптувати його під різні мережеві застосунки та вимоги.

6. Підтримує як пакети IPv4, так і IPv6, які можуть бути надіслані через LSP, встановлені за допомогою сигналізації RSVP.

1.4. Програмно-конфігуровані мережі Software-Defined Networking

Програмно-конфігурована мережа (SDN, Software-Defined Networking) являє собою інноваційний мережевий парадигм, що передбачає відокремлення рівня керування (контролю та адміністрування) від рівня даних (пересилання трафіку). Такий архітектурний поділ радикально знижує складність і значно покращує керованість мережевою інфраструктурою. Це перетворення зводить мережеві комутатори до ролі простих пристроїв, які відповідають лише за пересилання даних. Управління мережею, натомість, консолідується в єдиному контролері, що функціонує як центральний мозок системи. Контролер володіє глобальним уявленням про мережу та використовує програмні абстракції для забезпечення програмного контролю над усією інфраструктурою, надаючи мережевим операторам можливість керування базовими пристроями у режимі реального часу. На рівні управління (Application Plane) визначаються мережеві програми, які реалізують функціонал на кшталт мережевих політик, моніторингу, балансування навантаження, тощо. Ці програми встановлюються адміністратором відповідно до вимог користувача та конкретного середовища застосунків.

Інтерфейси прикладного програмування (API, Application Programming Interfaces) відіграють критично важливу роль у функціональній архітектурі SDN,

оскільки вони забезпечують взаємодію між трьома основними рівнями: даних, контролю та управління. Ці API класифікуються відповідно до напрямку взаємодії:

- API-інтерфейси південного напрямку (SBI) — забезпечують зв'язок між контролером та пристроями рівня даних.

- API-інтерфейси північного напрямку (NBI) — забезпечують зв'язок між програмами управління та контролером.

- API-інтерфейси східного/західного напрямку — використовуються для координації між розподіленими контролерами у великих мережах.

- API є невід'ємними архітектурними компонентами SDN і використовуються для конфігурування як кінцевих маршрутизаторів, так і мережевих програм. Багаторівнева архітектура SDN, включно з цими API, детально проілюстрована на рис. 1.7.

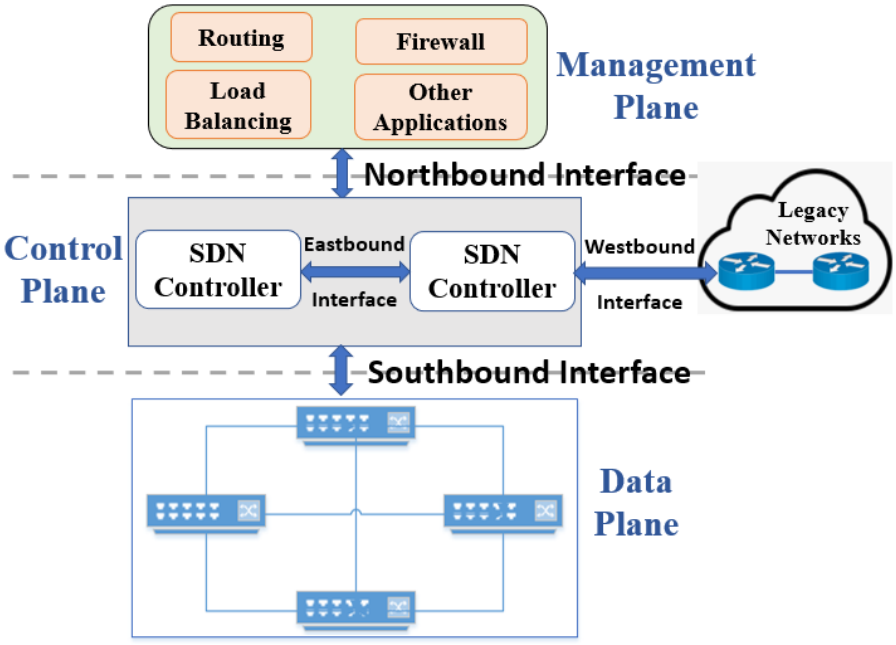


Рис.1.7 – Архітектура системи SDN

Південний API (Southbound API)

API південного напрямку (SBI, Southbound API) є ключовим інструментом в архітектурі SDN, що забезпечує протокол зв'язку між площиною керування (Control Plane) та площиною даних (Data Plane). Він використовується для передачі

конфігураційної інформації та встановлення правил пересилання/маршрутизації у площині даних мережевих пристроїв. Однією з найважливіших функцій SBI є абстрагування функціональності фізичних маршрутизуючих пристроїв для площини керування. Крім того, він дозволяє контролеру встановлювати топологію мережі, визначати маршрути трафіку та виконувати запити від імені площини управління. SBI має забезпечувати доступний та безпечний зв'язок у мережі. Протокол OpenFlow (OFP) традиційно використовується як найпоширеніший SBI, надаючи безпечний канал між рівнем управління та рівнем даних для встановлення правил потоку трафіку. Інтерфейси SBI класифікуються залежно від їхньої залежності від OpenFlow, незалежності від нього, або ж використання новітніх технологій, як це проілюстровано на рисунку 1.8.

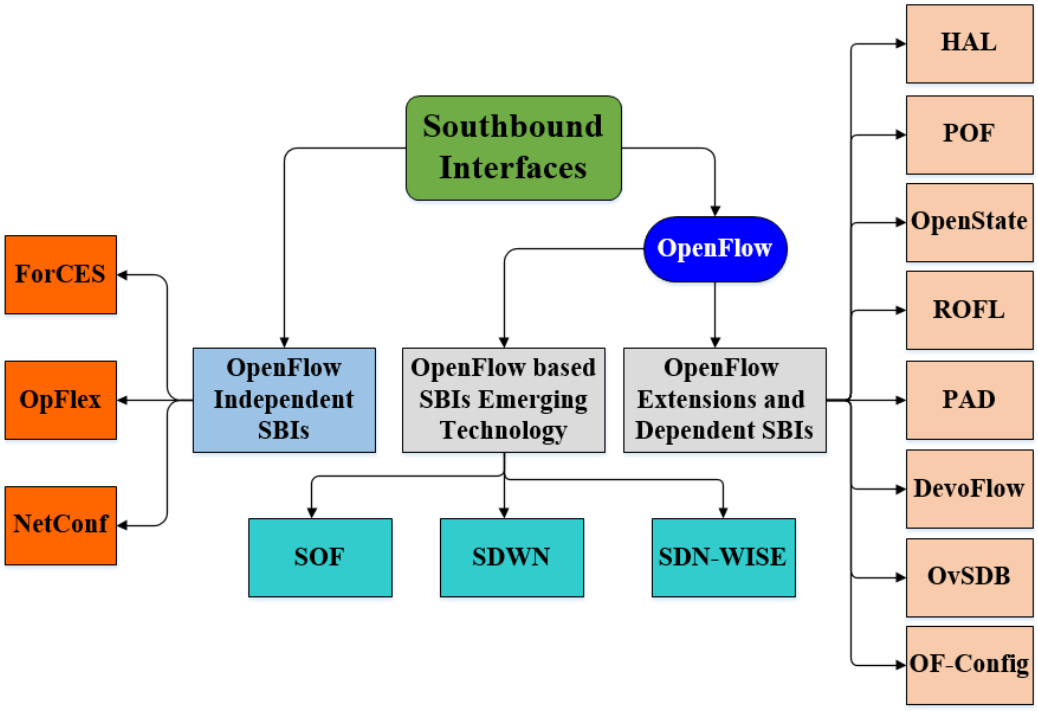


Рис. 1.8 – SBI

Північний API (Northbound API)

API північного напрямку (NBI, Northbound API) слугує інтерфейсом, який використовується між рівнями контролю та управління (Application Plane) для надання розробникам програм можливості програмного доступу та програмування

мережевої інфраструктури. З огляду на його функціональність, NBI є критично важливим елементом для забезпечення підтримки різноманітних програм SDN. Через відсутність стандартизації цього інтерфейсу, сучасні контролери та мови програмування пропонують досить широкий спектр NBI. Додатково, багато платформ контролерів та окремі програмісти використовують REST API як один із поширених варіантів реалізації NBI.

Східний/Західний API (East/Westbound API)

Зв'язок між різними доменами SDN реалізується за допомогою API східного напрямку (Eastbound API). Водночас, API західного напрямку (Westbound API) відповідає за встановлення зв'язку між доменами SDN та застарілими мережевими доменами (Legacy Domains). Хоча централізоване керування мережею є основною та ключовою особливістю SDN, один контролер може ефективно працювати лише з обмеженою кількістю маршрутизаторів. Для адаптації до експоненціального зростання кількості пристроїв пересилання та для побудови великомасштабних мереж виникає необхідність застосування розподілених контролерів.

Переваги SDN

Технологія SDN (Software-Defined Networking) пропонує значні переваги порівняно з традиційними мережами завдяки спрощенню обслуговування, полегшенню управління та ефективнішій реалізації політик контролю доступу (ACL). Пристрої, що відповідають за пересилання даних, стають функціонально простішими, оскільки весь мережевий інтелект переноситься на контролер. Їхня роль зводиться до виконання інструкцій контролера без необхідності розуміти та обробляти різнорідні алгоритми та протоколи. Крім того, пристрої пересилання також сприяють контролеру у виконанні таких завдань, як: розрахунок маршрутів; моніторинг стану каналів та вузлів; керування мережею та діагностика. Ці та інші фактори обумовлюють численні переваги SDN у порівнянні з традиційними мережами, серед яких виділяють:

- Централізований контроль — централізоване керування та контроль над усіма мережевими пристроями SDN суттєво знижує загальну складність мережевої інфраструктури.

- Пряма програмованість — площина управління мережею може бути безпосередньо запрограмована через контролер, оскільки вона архітектурно відокремлена від площин даних та управління.

- Автоматизація та ппрощене управління — спрощене управління мережею та високий рівень автоматизації досягається завдяки загальним API-інтерфейсам для програмування програм, що стало можливим завдяки наданню абстракцій платформою контролера.

- Висока швидкість інновацій — SDN дозволяє швидко впроваджувати інновації, оскільки відпадає необхідність налаштовувати кожен окремий пристрій та очікувати на випуск нових версій апаратного забезпечення від постачальників.

- Переадресація на основі правил потоку — рішення про пересилання приймаються на основі правил потоку (flow rules), а не лише на залежності від адрес призначення (як у традиційній IP-маршрутизації).

- Деталізована реалізація політик ACL — дозволяє мережевим адміністраторам застосовувати політики контролю доступу (ACL) на більш детальному рівні та в абстрактному, автоматизованому режимі.

- Підвищена зручність використання — забезпечується кращий інтерфейс користувача завдяки централізації управління мережею та наданню інформації про стан для додатків вищого рівня, що дозволяє системі легко адаптуватися до динамічних потреб користувача.

- Посилена безпека — SDN надає централізований контроль над безпекою всієї мережі.

- Забезпечення QoS — SDN використовує інтелектуальні функції центрального управління для агрегації послуг LTE та WAN, що дозволяє задовольняти зростаючі обчислювальні потреби мобільних користувачів.

1.5. Ланцюги мережевих функцій Service Function Chaining

Ланцюг сервісних функцій (SFC, Service Function Chaining) по суті є реалізацією маршрутизації на основі політик (PBR, Policy-Based Routing) у контексті програмно-конфігурованих мереж (SDN). Хоча в багатьох випадках SFC застосовується для забезпечення безпеки мережі, його функціонал може охоплювати широкий спектр інших мережевих сервісів.

SFC реалізує механізм, який спрямовує пакети послідовно через одну або кілька сервісних функцій, повністю ігноруючи при цьому звичайну маршрутизацію, що використовує лише IP-адресу призначення для визначення шляху. Ці сервісні функції фактично емулюють послідовність фізичних мережевих пристроїв (наприклад, міжмережевих екранів, систем виявлення вторгнень), з'єднаних між собою за допомогою віртуальних "кабелів".

Базовий приклад застосування SFC включає маршрутизацію пакета від точки А до точки Б, де пакет обов'язково проходить через брандмауер, при цьому брандмауер не обов'язково має IP-адресу наступного переходу з точки зору традиційної маршрутизації. Більш складний приклад передбачає упорядкований ланцюг сервісних функцій, кожна з яких може бути реалізована з використанням декількох екземплярів (наприклад, віртуальних машин, VM). У цьому сценарії пакети спочатку проходять через одну VM, а потім алгоритм хешування (або інший механізм балансування) розподіляє потоки між декількома іншими VM на кожному подальшому переході ланцюга.

1.6. Сегментна маршрутизація Segment Routing

Архітектура Segment Routing

Спираючись на архітектуру, описану в RFC 8402 (Segment Routing Architecture), розглянемо ключові концепції Сегментної Маршрутизації (SR), які залишаються незалежними від конкретної площини даних, що використовується для реалізації (наприклад, MPLS або IPv6). Простий приклад SR-шляху, що

складається з трьох послідовних сегментів (S1, S2, S3), проілюстровано на Рис. 1.9 . Ми називаємо цей впорядкований перелік сегментів політикою SR. Таким чином, політика маршрутизації сегментів P=<S1,S2,S3> визначає, що пакет має бути спрямований спочатку через вузол S1, потім через вузол S2, і врешті-решт досягти місця призначення S3. Цей упорядкований список сегментів інкапсулюється безпосередньо у заголовки пакетів вихідним вузлом політики SR. Домен сегментної маршрутизації визначається як набір мережевих вузлів, які спільно беруть участь у цій моделі маршрутизації на основі джерела (Source Routing).

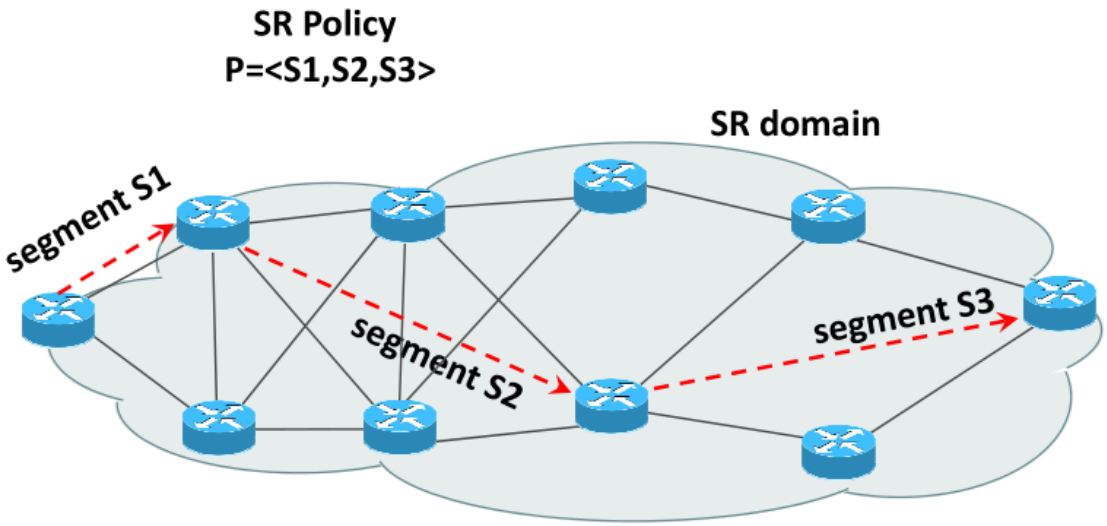


Рис. 1.9 – SR-політика та сегменти

Кожен сегмент у Сегментній Маршрутизації описується унікальним ідентифікатором сегмента (SID, Segment ID). Залежно від базової площини даних, SID має різну реалізацію: для площини даних MPLS SID відповідає мітці MPLS, тоді як для площини даних IPv6 SID є IPv6-адресою. Як ілюструє Рис. 1.10 , список сегментів інкапсулюється у заголовки пакетів головним вузлом (Headend). Цей головний вузол відповідає за спрямування пакетів, що належать до політики SR, і може бути або безпосередньо відправником пакета (наприклад, хостом), або

проміжним вузлом, який виконує класифікацію трафіку та асоціює політики SR із відповідними пакетами.

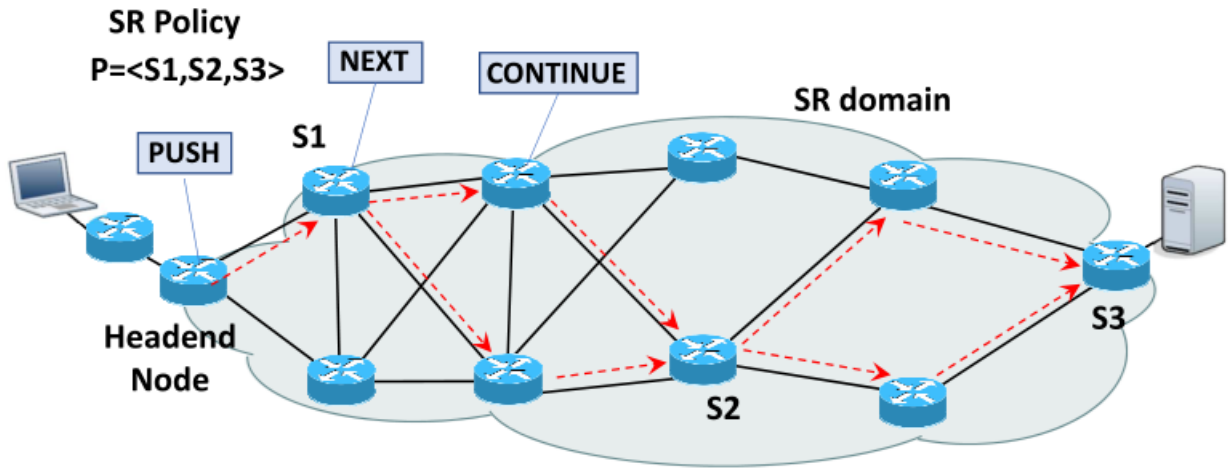


Рис. 1.10 – SR операції

Іншими словами, хости можуть бути включені до складу домену сегментної маршрутизації (SR), проте їхня участь не є обов'язковою і визначається специфікою загального сценарію застосування SR. Основним припущенням є те, що всі вузли, які входять до домену SR, повинні перебувати під управлінням одного й того самого адміністративного об'єкта. Як приклад, магістральна мережа певного постачальника послуг (Service Provider) може формувати єдиний домен SR, і тоді головний вузол (Headend) буде відповідати вхідному граничному маршрутизатору цієї магістралі (у такому випадку кінцеві хости не включаються до домену SR).

Операції над SID

Для площини даних Сегментної Маршрутизації (SR) визначено три основні операції, що застосовуються до SID (Segment ID) та списків сегментів: PUSH, NEXT та CONTINUE. У наведених прикладах (рис. 1.9 та рис. 1.10) для спрощення ми припускаємо, що ідентифікатори S1 та S2 являють собою топологічні інструкції (вузли, які необхідно відвідати), а S3 є вузлом призначення політики SR P. Політика P = S1, S2, S3 інструктує пакет послідовно пройти через два проміжні вузли,

ідентифіковані як SID S1 та SID S2, а потім досягти кінцевого вузла, ідентифікованого як SID S3.

Операція PUSH полягає у вставці сегмента на верхівку списку сегментів, тобто як новий перший сегмент політики SR. Для формування описаної вище політики P головний вузол (Headend) виконує операції PUSH у зворотному порядку: PUSH(S3), PUSH(S2), PUSH(S1). У пакеті SR сегмент, який містить поточну інструкцію для виконання, називається активним сегментом. У випадку з SR-політикою P, головний вузол відправляє пакет з активним сегментом S1.

Операція NEXT виконується вузлом, який завершив обробку поточного активного сегмента та повинен визначити, який саме наступний сегмент у списку політики SR підлягає обробці. У нашому прикладі, вузол з ідентифікатором SID S1 отримує пакет і виконує операцію NEXT. Внаслідок цього наступний сегмент — S2 — стає новим активним сегментом, і пакет відповідно перенаправляється до S2. Операція NEXT також охоплює сценарій із останнім вузлом політики SR; у цьому випадку виконання NEXT зазвичай призводить до обробки пакета відповідно до правил звичайної IP-маршрутизації.

Нарешті, операція CONTINUE виконується проміжними вузлами, які знаходяться на шляху між двома сегментами. Наприклад, вузли, що розташовані на шляху між S1 та S2, будуть виконувати операцію CONTINUE. Важливо відзначити, що шлях між S1 та S2 не фіксується політикою SR і обирається згідно з традиційною IP-маршрутизацією (за найкоротшим шляхом). Якщо між вузлами S1 та S2 існує кілька шляхів з рівною вартістю (як показано на рис. 1.10) та якщо в домені SR підтримується механізм ECMP (Equal Cost MultiPath), Сегментна Маршрутизація може зручно використовувати ECMP для балансування трафіку.

Сегменти Segment Routing

Сегменти в технології Сегментної Маршрутизації (SR) поділяються на дві основні категорії: глобальні та локальні. Глобальні сегменти представляють інструкції, які є дійсними та можуть бути виконані будь-яким вузлом у всьому

домені SR. На противагу цьому, локальні сегменти відповідають інструкціям, сфера дії яких обмежена межами одного конкретного вузла.

Типовим прикладом глобального сегмента є інструкція, що вимагає пересилання пакета до заданої IP-мережі або конкретного IP-вузла. У контексті домену SR, де використовується протокол внутрішнього шлюзу (IGP, наприклад, OSPF або ISIS), ці інструкції називаються сегментом префіксу IGP та сегментом вузла IGP (або просто сегментом префіксу та сегментом вузла). Усі маршрутизаційні вузли в домені SR мають здатність виконувати інструкції цих глобальних сегментів, керуючись найкоротшим шляхом до мережі або вузла призначення, що міститься у їхніх таблицях маршрутизації.

Найважливішим прикладом локального сегмента є інструкція, яка вимагає пересилання пакета до вузла, ідентифікованого протоколом IGP як безпосередній сусід. Така інструкція, що відповідає надсиланню пакета через певний вихідний інтерфейс, може бути виконана лише конкретним вузлом, для якого цей сусід є прямим підключенням. Ця інструкція відома як Сегмент суміжності IGP (IGP-Adjacency Segment). Завдяки можливості використання сегментів IGP-Adjacency можна довести, що будь-який можливий шлях через домен SR може бути представлений політикою SR (яка складається з комбінації глобальних і локальних сегментів). Локальні сегменти також можуть бути використані для кодування службових інструкцій, виконання яких необхідне на даному вузлі. Слід зазначити, що процеси перетворення глобальних і локальних сегментів в ідентифікатори сегментів (SID) та їх розподіл у домені SR відрізняються для реалізацій SR-MPLS та SRv6.

Сегмент IGP-Anycast є різновидом сегмента префіксу IGP, який асоційований із префіксом Anycast. Префікс Anycast — це такий префікс, що оголошується групою маршрутизаторів. Його використання є типовим для забезпечення високої доступності або реалізації балансування навантаження.

Сегмент прив'язки (Binding Segment) використовується для встановлення зв'язку (біндингу) політики SR (тобто цілого списку сегментів) з єдиним ідентифікатором сегмента (SID) на даному вузлі, який називається Binding SID (BSID). Це означає, що вузол, який обробляє Binding SID, замінює цей сегмент відповідним списком сегментів: пакет, отриманий з BSID як активний сегмент, буде далі керуватися згідно з визначеною політикою SR. Таким чином, класифікація пакетів може бути виконана на вузлі X, який просто додає BSID до заголовка SR, не знаючи при цьому деталей самої політики SR (списку сегментів). Завдяки BSID пакет буде перенаправлено на вузол Y, який і застосує повну політику SR.

Для прикладу, вузол X може класифікувати трафік до мережі призначення N на дві групи: той, що вимагає малої затримки (low latency), і той, що вимагає малої втрати пакетів (low loss). Вузол Y є вхідним маршрутизатором магістральної мережі, що забезпечує підключення до мережі N. Для пересилання трафіку до N магістральна мережа використовує дві відповідні політики SR (low latency та low loss). Ці списки сегментів можуть з часом змінюватися залежно від рішень щодо керування трафіком. Після таких змін вузол Y потребує лише переконфігурації для застосування поточної політики SR до пакетів, ідентифікованих BSID. Натомість, вузол X не потребує переналаштування, оскільки BSID залишаються незмінними. Такий підхід значно підвищує масштабованість, відмовостійкість та незалежність сервісів у рішеннях, заснованих на сегментній маршрутизації.

Концепції Segment Routing

Таблиця 1.1 демонструє зіставлення ключових концепцій Сегментної Маршрутизації (SR) з їхніми конкретними реалізаціями у двох основних площинах даних (Data Planes): MPLS (Multiprotocol Label Switching) та IPv6 (Internet Protocol version 6). Це зіставлення є критично важливим для розуміння того, як абстрактні інструкції SR (наприклад, Segment ID, список сегментів та операції) втілюються у

відповідні мережеві механізми (мітки, заголовки розширення IPv6) у кожній із цих технологій.

Таблиця 1.1

Концепції SR в площинах даних MPLS та IPv6

Generic SR	SR-MPLS	SRv6
SR Policy	Label Stack	Segment List (of IPv6 addresses) in the SR Header
Active Segment	Topmost Label	IPv6 address indicated in the IPv6 Destination Address
PUSH Operation	Label Push	Adding an IPv6 in the Segment List in the SR Header
NEXT Operation	Label POP	Decrementing the Segment Left field, copying the active segment in the IPv6 Destination Address
CONTINUE Operation	Label Swap	Forwarding according to IPv6 Destination Address

Network Service Header

Що стосується сценаріїв Ланцюга сервісних функцій (SFC), протокол Network Service Header (NSH) є спеціалізованим рішенням, яке функціонує поверх існуючих технологій тунелювання. NSH може бути використаний у двох ключових конфігураціях: по-перше, він може застосовуватися у поєднанні з Сегментною Маршрутизацією (SR), де SR виконує роль базового механізму тунелювання (додатково розширеного функціями керування трафіком). По-друге, NSH може виступати альтернативою сегментній маршрутизації для повної реалізації функціональності ланцюжка сервісних функцій. Водночас, у наукових роботах [20] та [21] детально описуються перспективні сценарії об'єднання сервісних функцій, де SRv6 (сегментна маршрутизація на базі IPv6) демонструє потенціал повністю замінити рівень NSH. Така заміна призведе до значного спрощення інфраструктури та суттєвого зниження обчислювального навантаження на мережеві пристрої.

Рівень передачі даних SR-MPLS

SR-MPLS (Segment Routing over MPLS) не вимагає жодних змін у базовій площині пересилання MPLS. Політика SR реалізується шляхом використання стека міток MPLS: ідентифікатори сегментів (SID) зі списку сегментів просто вставляються у пакет як мітки MPLS. Класичні функції пересилання, доступні у мережах MPLS, ідеально підходять для реалізації основних операцій SR:

- Операція PUSH (додавання сегмента) відповідає функції Label Push (вставлення мітки поверх стека).

- Операція NEXT (перехід до наступного сегмента) відповідає функції Label Pop (тобто видалення верхньої мітки).

- Операція CONTINUE (проміжне пересилання) відповідає функції Label Swap (тобто зіставлення вхідної мітки з вихідним інтерфейсом та новою вихідною міткою та пересилання пакета).

Інкапсуляція IP-пакета в пакет SR-MPLS виконується на межі домену SR-MPLS з повторним використанням концепції Класу еквівалентності пересилання MPLS (FEC). При цьому FEC може асоціюватися з певною політикою SR.

Перетворення сегментів на мітки MPLS (SID) є критично важливим процесом у площині даних SR-MPLS. Оскільки різні маршрутизатори в домені SR можуть використовувати різні доступні діапазони міток, кожен маршрутизатор оголошує свій власний простір міток, який буде використовуватися для глобальних сегментів. Цей простір називається Глобальний блок маршрутизації сегментів (SRGB, Segment Routing Global Block). Цей простір може складатися з набору несуміжних блоків. З цієї причини глобальні сегменти в домені SR ідентифікуються індексом, який має бути відображений у форму мітки з урахуванням конкретного вузла, який оброблятиме цю мітку.

Припускаючи, що SRGB вузла визначає діапазон міток, що починається з певного значення (наприклад, 10000), для глобального сегмента з індексом X вузол повинен отримати мітку, що розраховується як початкове значення + X (наприклад,

$10000 + X$). Як приклад, на Рис. 1.11 розглядається реалізація політики SR (описаної на Рис. 1.10) із використанням площини даних SR-MPLS з припущенням, що різні вузли мають різні SRGB. Головний вузол має заздалегідь визначити, який саме SRGB вузлів виконуватиме операцію NEXT для сегментів, щоб правильно сформувати мітку. Початкова мітка для сегмента S2, встановлена головним вузлом, розраховується як $1000 + 2 = 1002$ (де 1000 — це SRGB вузла S1, а 2 — індекс для S2). Вузол S1 повинен змінити мітку на $4000 + 2 = 4002$, якщо пакет прямує до вузла N4 (чиїй SRGB дорівнює 4000), або на $6000 + 2 = 6002$, якщо пакет прямує до вузла N6 (чиїй SRGB дорівнює 6000).

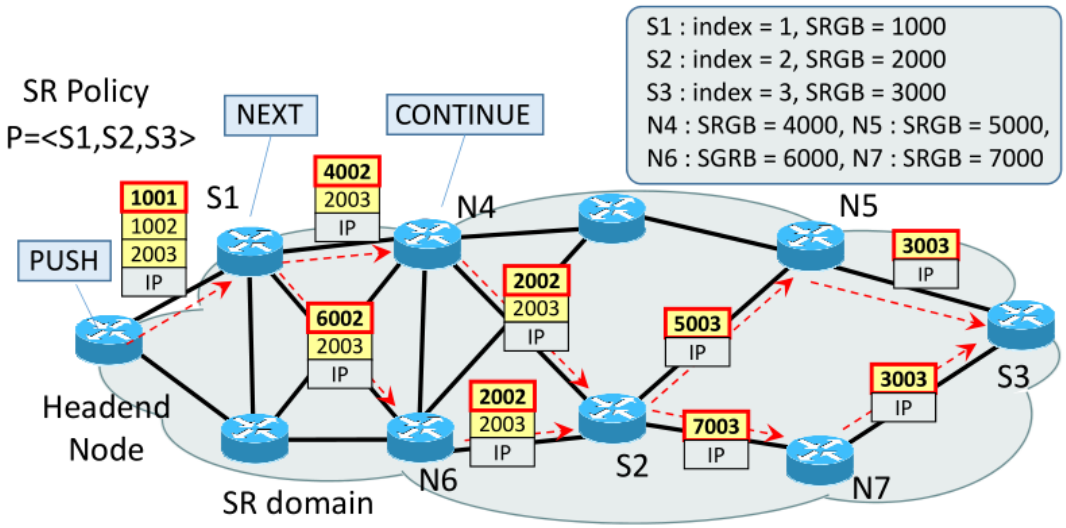


Рис. 1.11 – Зв’язування сегментів з мітками у випадку різних SRGB на вузлах

Обидва вузли, N4 та N6, зрештою перевизначать (замінять) мітку на $2000 + 2 = 2002$ при пересиланні пакетів до S2. Початкова мітка для вузла S3, встановлена головним вузлом, розраховується як $2000 + 3 = 2003$ (де 2000 — SRGB вузла S2, а 3 — індекс сегмента S3). Ця мітка досягне вузла S2 без змін, а потім буде належним чином оброблена вузлом S2, який перепризначить (замінить) її з урахуванням SRGB наступного переходу на шляху до фінального вузла S3.

Процес постійного перепризначення міток (label re-mapping), викликаний використанням різних Глобальних Блоків Сегментної Маршрутизації (SRGB) на маршрутизаторах, значно ускладнює операції та ускладнює усунення

несправностей у мережі. Крім того, існують певні сервіси, наприклад, ті, що включають сегменти довільної передачі (arbitrary forwarding segments), які неможливо реалізувати ефективно, якщо вузли використовують несумісні SRGB. З огляду на ці проблеми, у науковій роботі [22] наполегливо рекомендується використовувати ідентичний діапазон міток SRGB на всіх маршрутизаторах домену. Це гарантує, що глобальний сегмент завжди буде відображатися на один і той самий SID (мітку MPLS) незалежно від вузла, який його обробляє. На Рис. 1.12 проілюстровано відображення тієї ж політики SR, що була описана на Рис. 1.11, але вже в рекомендованому режимі роботи, де на всіх вузлах використовується ідентичний SRGB.

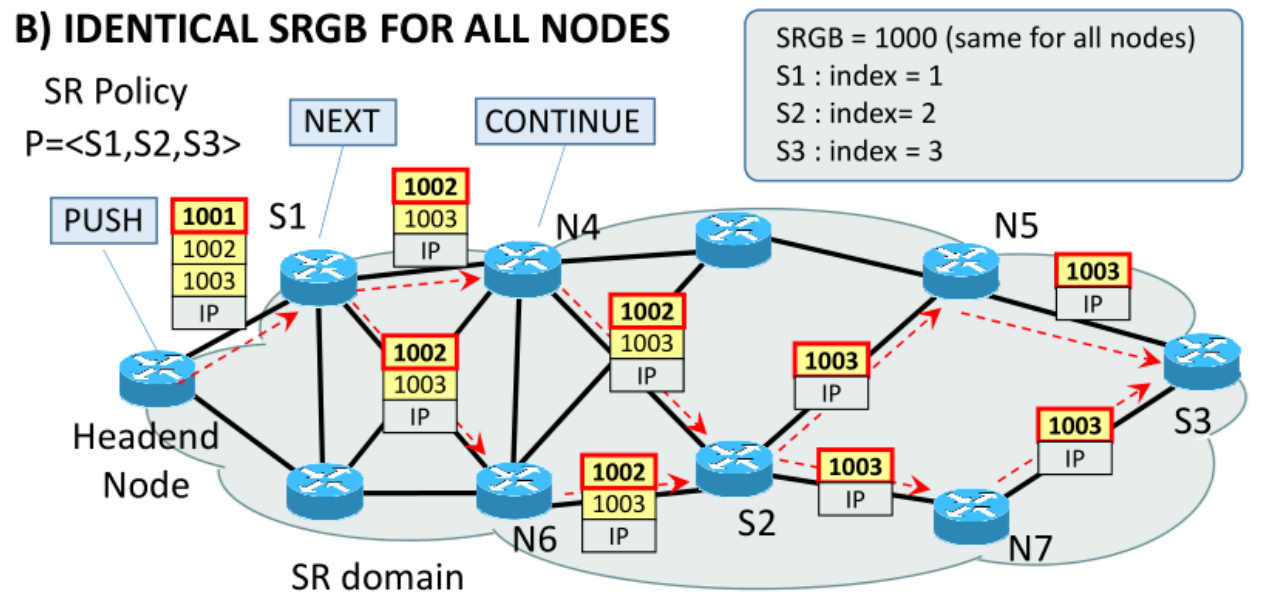


Рис. 1.12 – Зв’язування сегментів з мітками у випадку однакового SRGB на вузлах

Рівень передачі даних Segment Routing IPv6

Для площини даних Сегментної Маршрутизації на базі IPv6 (SRv6) було стандартизовано новий тип заголовка розширення маршрутизації IPv6, який отримав назву Segment Routing Header (SRH) [16]. Формат цього заголовка SRH визначає, що він містить список сегментів (який по суті є політикою SR) у вигляді упорядкованого переліку IPv6-адрес, де кожна адреса в цьому списку є

ідентифікатором сегмента (SID). У заголовку SRH присутнє спеціальне поле Segments Left, яке використовується для зберігання вказівника на активний SID у списку сегментів, тобто на сегмент, який має бути оброблений наступним.

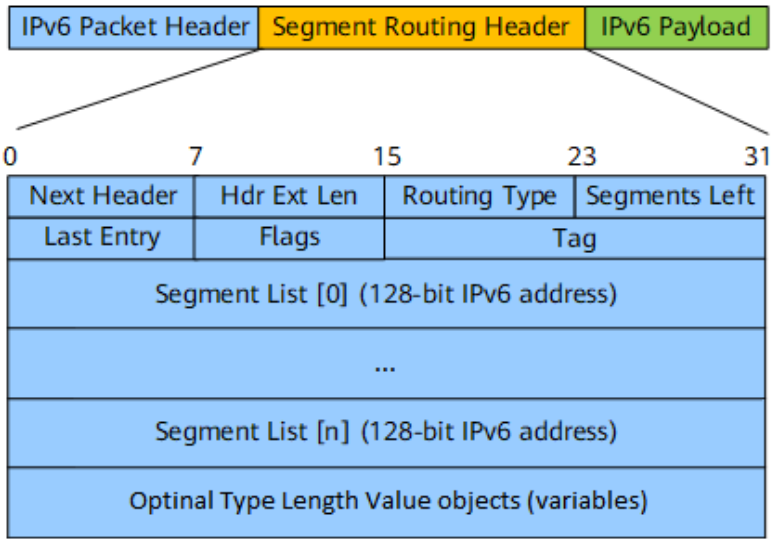


Рис. 1.13 – Структура SRH

У площині даних сегментної маршрутизації на базі IPv6 (SRv6) визначено три функціональні категорії вузлів: вихідні вузли SR, транзитні вузли та вузли кінцевих точок сегмента SR (SR Segment Endpoint nodes). Вихідний вузол SR (SR Headend) відповідає раніше описаному головному вузлу і може бути або вузлом, що створює сам пакет IPv6 (хостом), або вхідним маршрутизатором домену SR, який виконує інкапсуляцію отриманого пакета. На Рис. 1.14 розглянуто другий, більш поширений випадок, коли вихідний вузол SR є граничним маршрутизатором. Цей маршрутизатор інкапсулює отриманий пакет (який може бути IPv6, IPv4, або навіть кадр рівня 3) у зовнішній пакет IPv6 та вставляє заголовок сегментної маршрутизації (SRH) як заголовок розширення маршрутизації (Routing Extension Header) у зовнішній заголовок IPv6. Інкапсульований пакет позначається як корисне навантаження (Payload). У цьому прикладі список сегментів у SRH складається з S1, S2 та S3, причому вони зберігаються у зворотному порядку (перший SID у списку — це S3, який є останнім сегментом у політиці SR). У полі Segments Left встановлено значення 2, що вказує на те, що активним сегментом є

S1 (позначений на малюнку червоним кольором). Вихідний вузол SR встановлює IPv6-адресу призначення пакета на адресу першого SID політики SR (S1).

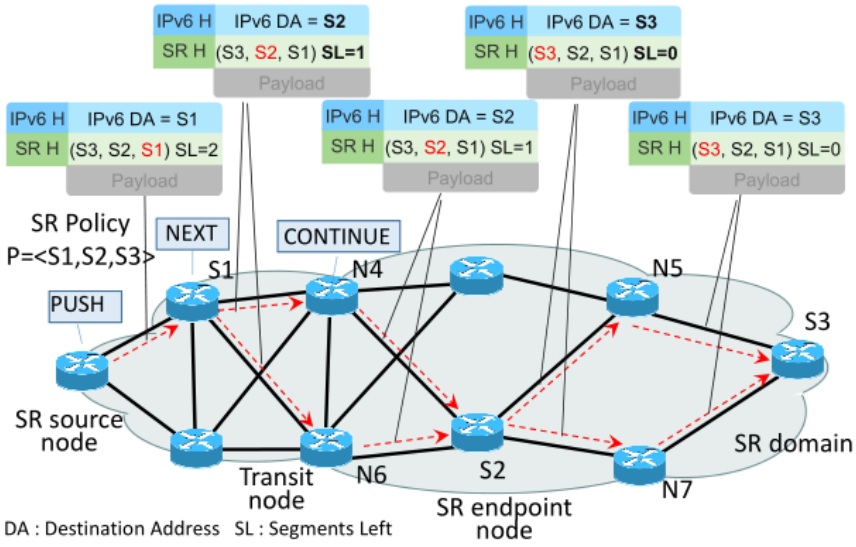


Рис. 1.14 – Операції площини даних SRv6

Вузол кінцевої точки сегмента SR (SR Segment Endpoint node) отримує пакети, чия IPv6-адреса призначення локально налаштована як активний ідентифікатор сегмента (SID). Отримавши пакет, вузол кінцевої точки перевіряє заголовок сегментної маршрутизації (SRH), щоб ідентифікувати новий активний сегмент (тобто наступний SID у списку). Після цього він змінює IPv6-адресу призначення у зовнішньому заголовку IPv6 на адресу цього нового активного сегмента та пересилає пакет, керуючись своєю таблицею пересилання IPv6. Ці дії відповідають операції NEXT. На Рис. 1.14 ми бачимо, що S1 (як перший вузол кінцевої точки SR) зменшує значення поля Segment Left до 1, роблячи S2 активним сегментом, і встановлює S2 як нову IPv6-адресу призначення.

Транзитний вузол (Transit Node) просто пересилає пакет, що містить заголовок SR, як звичайний пакет IPv6, тобто керуючись лише зовнішньою IPv6-адресою призначення. Він робить це тому, що адреса призначення IPv6 локально не збігається з жодним налаштованим сегментом. Ці дії відповідають операції CONTINUE. На Рис. 1.14 вузли N4, N5, N6 та N7 є транзитними вузлами, які

виконують стандартне пересилання пакета до поточної IPv6-адреси призначення. Важливою особливістю SRv6 є те, що транзитні вузли можуть бути непоінформовані про використання SRv6, оскільки будь-який маршрутизатор IPv6 може діяти як транзитний вузол SRv6.

У цьому прикладі операція PUSH (додавання сегмента) виконується шляхом інкапсуляції вихідного пакета (який може бути кадром IPv6, IPv4 або навіть рівня 2) у зовнішній пакет IPv6 із додаванням заголовка сегментної маршрутизації (SRH). Інша можливість реалізації PUSH, яка застосовується лише для пакетів IPv6, полягає у вставці SRH як нового заголовка між існуючим заголовком IPv6 та наступним заголовком (наприклад, заголовком транспортного рівня, TCP або UDP) без необхідності інкапсуляції пакета у новий зовнішній пакет IPv6. Цей варіант особливо зручний, коли хост-джерело самостійно виступає як вихідний вузол SR (головний вузол).

На додаток до основних операцій (PUSH/NEXT/CONTINUE), модель мережевого програмування SRv6 (Network Programming Model) [17] описує набір додаткових функцій, які можуть бути асоційовані з сегментами та виконані на певних вузлах SRv6. Прикладами таких функцій є різні типи інкапсуляції пакетів (наприклад, IPv6-в-IPv6, IPv4-в-IPv6, Ethernet-в-IPv6), відповідна декапсуляція, а також операції пошуку в конкретній таблиці маршрутизації (наприклад, для підтримки VPN-сервісів). Список функцій, описаний у [17], не є вичерпним, оскільки будь-яка функція може бути логічно пов'язана з ідентифікатором сегмента на вузлі. Очевидно, що визначення стандартизованого набору функцій полегшує розгортання доменів SR із сумісним обладнанням від різних постачальників.

Відповідно до [17], поняття ідентифікатора сегмента (SID) може бути переглянуто з урахуванням того, що IPv6-адреси використовуються як SID у SRv6. 128-бітний SID логічно поділяється на три поля та інтерпретується як набір полів "LOCATOR:FUNCTION:ARGS" (скорочено "LOC:FUNCT:ARG") (рис. 1.15), де:

- LOCATOR (LOC) — включає L старших бітів.

- FUNCTION (FUNCT) — займає наступні F біт.
- ARGS (ARG) — включає A бітів, що залишилися.

Загальна довжина складає 128 біт (128 = L + F + A).

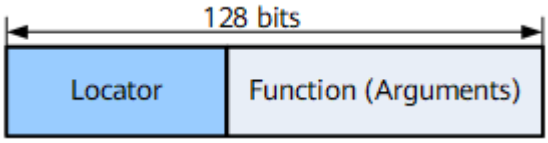


Рис. 1.15 – SRv6 SID

Поле LOCATOR (LOC) у структурі Segment ID (SID) відповідає префіксу IPv6 (наприклад, довжиною 48, 56 або 64 біта), який поширюється протоколами маршрутизації і забезпечує доступність до вузла, що виконує набір функцій. Довжина L локатора не є фіксованою і може бути обрана кожним оператором свого домену SR (незалежно для різних вузлів). Всі різноманітні функції, що розташовані на одному вузлі, можуть мати ідентичний локатор (LOC), але відрізняються лише кодом функції (FUNCT), завдяки чому їхні SID є унікальними. З точки зору маршрутизації, рішення SRv6 демонструють значну масштабованість, оскільки для вузла розподіляється лише один префікс (LOC), який здатний реалізувати потенційно велику кількість функцій з обмеженим впливом на загальний розмір таблиць маршрутизації вузлів домену SR. Біти ARG (аргументи) можуть використовуватися для передачі додаткової інформації (аргументів) до функції, що виконується. Це поле є необов'язковим: якщо довжина A=0, то SID логічно розкладається лише на два поля: LOC та FUNCT (128 = L + F).

SID, який розкладається на форму LOCATOR:FUNCTION:ARG, є глобальним сегментом, якщо префікс LOC маршрутизується у домені SRv6, що є типовим випадком. У SRv6 також можливо визначити локальні сегменти, які являють собою немаршрутизовані SID; вони можуть бути виконані вузлом, але їм обов'язково повинен передувати глобальний SID, який використовується для пересилання пакета на цей вузол. У будь-якому випадку, використання таких локальних SID можна уникнути, оскільки частина FUNCT глобального SID у формі

LOC:FUNCT:ARG може ефективно представляти будь-яку необхідну локальну функцію.

SRv6 Network Programming Model

Модель мережевого програмування SRv6 (SRv6 Network Programming model), визначена у [17], ґрунтується на об'єднанні функцій, які можуть бути розподілені по різних вузлах, для задоволення складних вимог мережі. Ці функції, описані в [17], спроможні підтримувати критично важливі сервіси та функції, такі як:

- VPN рівнів 3 та 2;
- Управління трафіком (Traffic Engineering, TE);
- Швидке перенаправлення (Fast ReRoute, FRR).

Ця модель пропонує розширені можливості для реалізації практично будь-якого мережевого сервісу шляхом комбінування базових функцій у формі мережевої програми, яка фактично вбудовується безпосередньо у заголовок пакета.

Як показано на рисунку 1.13, заголовок сегментної маршрутизації (SRH) може містити додатковий розділ, призначений для об'єктів Type Length Value (TLV). Ці об'єкти TLV використовуються для передачі додаткової інформації, яка має бути оброблена одним або декількома сегментами політики SR (списком сегментів). Наприклад, може бути доданий так званий HMAC TLV, який використовується для криптографічної перевірки того, що заголовок SRH був створений авторизованим вузлом і що список сегментів не був змінений під час передачі. Іншим можливим використанням об'єктів TLV є обмін інформацією про експлуатацію та обслуговування (OAM) між вузлами в домені SR.

У роботі [17] визначено два різні набори поведінки SRv6:

- Поведінка головного вузла політики SR (Headend behavior).
- Поведінка кінцевих точок (Endpoint behavior).

Як проілюстровано на рисунку 1.16, поведінка головного вузла політики SR виконується на вихідних вузлах SR (Headends), тоді як поведінка кінцевої точки

виконується на кінцевих вузлах SR (наприклад, S1, S2, S3). Поведінка головного вузла полягає у спрямуванні отриманих пакетів до визначеної політики SRv6 шляхом зіставлення атрибутів пакета з політикою. Кожна політика SRv6 має список ідентифікаторів SID, які прикріпляються до пакетів. Поведінка кінцевої точки SRv6 (також відома як поведінка, пов'язана з SID) є функцією, яка має бути виконана для пакета у певному місці в мережі. Така функція може бути як простою інструкцією маршрутизації, так і будь-якою розширеною функцією мережі (наприклад, брандмауер чи NAT).

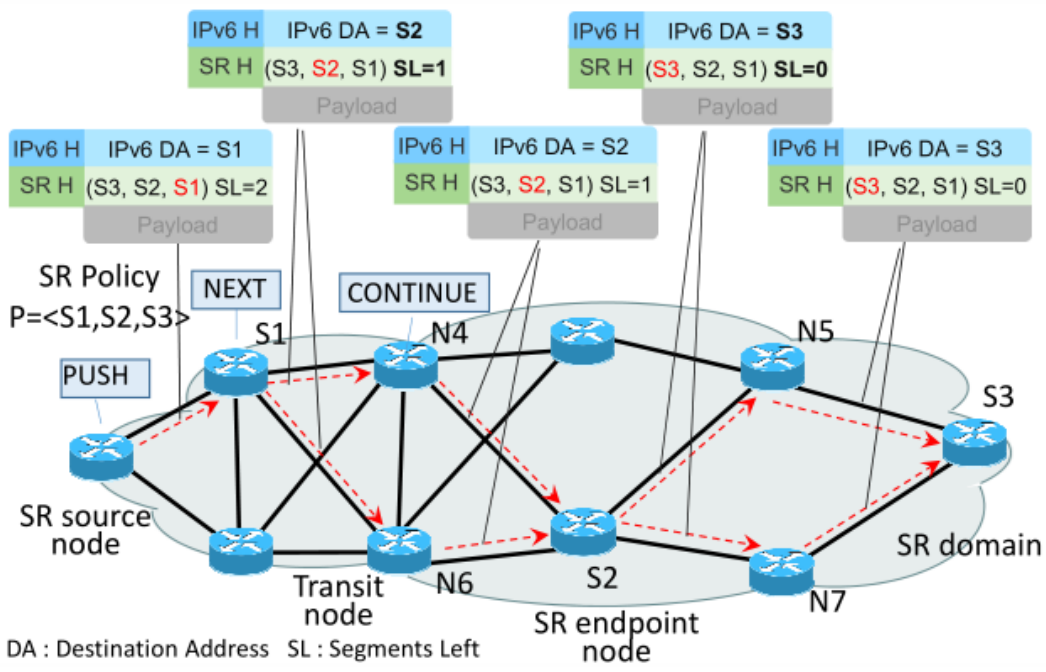


Рис. 1.16 – Операції площини даних SRv6

У Таблиці 1.2 представлено невичерпний перелік варіантів поведінки SRv6 (SRv6 Behaviors) разом із посиланнями на відповідні документи стандартизації, що містять їх детальний опис. Ці поведінки визначають, як головний вузол (Headend) обробляє пакети перед відправкою їх у домен Сегментної Маршрутизації. Поведінка H.Encaps (Headend Encapsulation) виконує інкапсуляцію вхідного пакета IPv6, який стає внутрішнім пакетом у схемі IPv6-in-IPv6. Зовнішній заголовок IPv6 містить Заголовок Сегментної Маршрутизації (SRH), який включає повний список

SID. Поведінка H.Encaps.L2 ідентична поведінці H.Encaps, проте відрізняється тим, що H.Encaps.L2 інкапсулює повний отриманий кадр рівня 2 (Layer 2 frame), а не лише IP-пакет (тобто здійснює інкапсуляцію Ethernet поверх IPv6). Поведінка H.Insert (Headend Insert) виконує вставку SRH у вихідний пакет IPv6 одразу після заголовка IPv6 і безпосередньо перед заголовком транспортного рівня (наприклад, TCP або UDP). При цьому вихідний заголовок IPv6 замінюється: зокрема, адреса призначення IPv6 замінюється адресою IPv6 першого сегмента у списку SID, а вихідна адреса призначення IPv6 пакета переноситься до заголовка SRH як останній SID у списку сегментів.

Таблиця 1.2

Список поведінок SRv6

Поведінка	**Документ з її описом**
H.Encaps	srv6-network-programming [17]
H.Insert	srv6-network-programming [17]
H.Encaps.L2	srv6-network-programming [17]
End	srv6-network-programming [17]
End.T	srv6-network-programming [17]
End.X	srv6-network-programming [17]
End.DT4	srv6-network-programming [17]
End.DT6	srv6-network-programming [17]
End.DX4	srv6-network-programming [17]
End.DX6	srv6-network-programming [17]
End.DX2	srv6-network-programming [17]
End.AS	service-programming (SFC) [23]
End.AD	service-programming (SFC) [23]
End.AM	service-programming (SFC) [23]
T.M.Tmap	mobile-uplane [24]

End.M.GTP4.E	mobile-uplane [24]
End.M.GTP6.D	mobile-uplane [24]
End.M.GTP6.E	mobile-uplane [24]

Поведінка End є найфундаментальнішою функцією серед поведінок кінцевої точки (Endpoints) у SRv6. Вона відповідає за заміну поточної IPv6-адреси призначення пакета на адресу наступного SID (Segment ID) у списку сегментів. Після цього вузол пересилає пакет, виконуючи пошук у таблиці маршрутизації за оновленою адресою призначення IPv6. Надалі цей пошук у таблиці маршрутизації називатиметься пошуком FIB (Forwarding Information Base). Поведінка End.T є спеціалізованим варіантом поведінки End, у якому пошук FIB виконується у конкретній таблиці IPv6, яка асоційована з SID, а не в основній таблиці маршрутизації вузла. Поведінка End.X є ще одним варіантом End, при якому пакет безпосередньо перенаправляється до вказаного сусіда рівня 3, який прив'язаний до SID, без виконання FIB-пошуку IPv6-адреси призначення.

Поведінка End.DT6 виконує витягнення інкапсуляції SRv6 з пакета і далі виконує пошук у FIB IPv6-адреси призначення відкритого внутрішнього пакета у певній таблиці IPv6, пов'язаній з SID. До SID також може бути прив'язана таблиця маршрутизації IPv6 за замовчуванням. У цьому випадку внутрішні пакети IPv6 будуть декапсульовані та потім перенаправлені на основі адреси призначення IPv6 відповідно до звичайної маршрутизації вузла. Поведінка End.DX6 видаляє інкапсуляцію SRv6 з пакета і перенаправляє отриманий пакет IPv6 до конкретного сусіда рівня 3, який асоційований з SID. End.DT4 та End.DX4 є, відповідно, IPv4-варіантами поведінок End.DT6 та End.DX6. Вони використовуються, коли інкапсульований пакет є пакетом IPv4. Поведінка End.DX2 застосовується для пакетів, які були інкапсульовані на рівні 2 (наприклад, за допомогою H.Encaps.L2). Вона знімає інкапсуляцію SRv6 та пересилає отриманий кадр L2 через вихідний інтерфейс, який пов'язаний з SID.

Segment Routing та Software-Defined Networking

Площина керування (Control Plane) для Сегментної Маршрутизації (SR) та її інтеграція з операціями керування SDN є необхідними для повноцінної реалізації функціональності площини даних та розкриття всіх можливостей SR. Площина керування може бути реалізована трьома основними способами.

Повністю розподілений підхід – маршрутизатори функціонують автономно, приймаючи незалежні рішення щодо налаштування та застосування політик SR.

Централізований підхід – використовується централізований SR-контролер, який приймає всі ключові рішення та надає інструкції маршрутизаторам відповідно до принципів SDN (Software-Defined Networking).

Гібридний підхід – застосовується комбінація розподіленого та централізованого підходів.

Для площини даних SR-MPLS (Segment Routing over MPLS) повністю розподілений підхід був розроблений у рамках IETF через розширення протоколів маршрутизації IGP (Interior Gateway Protocol), таких як OSPF та ISIS. Ці розширення дозволяють кожному маршрутизатору оголошувати різні типи IGP-сегментів (включаючи префікс, вузол, суміжність (adjacency), anycast) та поширювати певні дані конфігурації SR. Усі інші маршрутизатори в домені SR отримують цю інформацію через механізми, визначені протоколом маршрутизації IGP. Ця інформація є життєво важливою для зіставлення сегментів (які входять до політики SR) з ідентифікаторами SID, представленими у вигляді міток MPLS у площині даних SR-MPLS. При цьому кожен маршрутизатор може виділяти різні діапазони міток для використання у глобальних сегментах. Діапазон міток, який маршрутизатор використовує для глобальних сегментів, називається Глобальний Блок Маршрутизації Сегментів (SRGB, Segment Routing Global Block), і ця інформація анонсується разом з іншою конфігурацією SR за допомогою протоколу маршрутизації. Як було зазначено раніше, наполегливо рекомендується

використовувати ідентичний діапазон міток (SRGB) на всіх маршрутизаторах домену.

Для площини даних SRv6 (Segment Routing over IPv6) процес оголошення глобальних сегментів (IGP-префіксу, IGP-вузла та IGP-anycast) значно спрощується завдяки тому, що IPv6-адреси використовуються безпосередньо як SID. Зокрема, немає потреби у розширенні протоколів маршрутизації IGP для поширення цих типів сегментів, оскільки вони вже підтримуються основними механізмами маршрутизації. Крім того, не вимагається визначення SRGB (Глобального Блоку Сегментної Маршрутизації), як це було необхідно в SR-MPLS. Операції, пов'язані з глобальними сегментами, можуть повністю покладатися на IPv6-адреси, які глобально маршрутизуються у домені SR. Це означає, що рівень керування SRv6 може використовувати звичайні протоколи маршрутизації IGP (OSPF, ISIS) для підтримки основних операцій, тоді як розширення IGP необхідні лише для поширення сегментів суміжності (IGP-Adjacency) та іншої специфічної інформації конфігурації SR.

Визначення площини управління для сегментної маршрутизації розпочалося з площини даних SR-MPLS, а згодом площина даних SRv6 успадкувала більшість функціональних можливостей, адаптувавши їх до нової площини даних. Початковою метою проектування площини управління SR була підтримка повністю розподіленого підходу, при якому маршрутизатори приймають автономні рішення. Це дозволяло зберегти ту ж функціональність, що й у традиційної мережі MPLS, для роботи якої не потрібен централізований контролер SDN. З іншого боку, наразі ми спостерігаємо чітку тенденцію до гібридного підходу, де протоколи розподіленої маршрутизації співіснують із централізованим SR-контролером. Цей гібридний підхід ідеально узгоджується з основною концепцією SDN, яка передбачає спрощення пристроїв рівня даних та централізацію функцій рівня управління в контролерах SDN.

Зважаючи на це, архітектуру сегментної маршрутизації можна розгорнути шляхом пошуку правильного балансу між розподіленим та централізованим управлінням. Розподілене управління використовується маршрутизаторами для обміну інформацією про доступність та оцінку найкоротших шляхів традиційним способом, без необхідності взаємодії із централізованим контролером. Це найкращий підхід для забезпечення можливості підключення в глобальних мережах, в яких керуючі з'єднання між вузлами та контролерами SDN схильні до незначної затримки та ймовірності відмови. Сегментну маршрутизацію можна використовувати для швидкого перенаправлення трафіку шляхом попереднього налаштування політик SR, які надають альтернативні шляхи у разі збою каналу або вузла, і вони автоматично активуються вузлом при виникненні збою.

Попередній розрахунок таких політик SR може виконуватися у розподіленому режимі або централізовано у контролері. Базова інформація про топологію та додаткова інформація для керування трафіком повинні передаватися контролеру. Також йому має передаватись інформація, що стосується сервісів, яка оголошується вузлами з використанням протоколів розподіленої маршрутизації. Контролер SDN може отримувати цю інформацію по-різному. Наприклад, він може брати участь у протоколі маршрутизації IGP або може взаємодіяти з маршрутизаторами власним способом для вилучення баз даних IGP. В іншому випадку він може отримувати інформацію від маршрутизаторів, які використовують розширення BGP-LS (BGP-Link State). Який механізм не використовувався б для отримання необхідної інформації від вузлів, контролер SDN відповідає за прийняття рішень про політики SR, які реалізують розширені функції або сервіси, такі як управління трафіком, VPN або SFC. Такий підхід дозволяє чітко відокремити операції площини даних від логіки сервісу, що працює у площині управління.

Розгортання архітектури сегментної маршрутизації (SR) вимагає досягнення оптимального балансу між розподіленим та централізованим

управлінням. Розподілене управління використовується маршрутизаторами для обміну інформацією про доступність та традиційної оцінки найкоротших шляхів без необхідності постійної взаємодії із централізованим контролером. Цей підхід є найкращим для забезпечення базової можливості підключення у глобальних мережах, де керуючі з'єднання між вузлами та контролерами SDN схильні до затримок і ймовірності відмови. Сегментна маршрутизація може ефективно використовуватися для швидкого перенаправлення трафіку (Fast ReRoute, FRR) шляхом попереднього налаштування політик SR, які визначають альтернативні шляхи у разі виникнення збою каналу або вузла. Ці політики автоматично активуються вузлом при виявленні несправності. Попередній розрахунок таких політик SR може виконуватися або у розподіленому режимі (локально на маршрутизаторах), або централізовано у контролері.

Для прийняття обґрунтованих рішень про політики SR централізованому контролеру необхідно передавати:

- Базову інформацію про топологію.
- Додаткову інформацію для керування трафіком.
- Інформацію, що стосується сервісів, яка оголошується вузлами за допомогою протоколів розподіленої маршрутизації.

Контролер SDN може отримувати цю інформацію різними способами: він може брати участь у протоколі маршрутизації IGP, взаємодіяти з маршрутизаторами власним механізмом для вилучення баз даних IGP, або отримувати інформацію від маршрутизаторів, які використовують розширення BGP-LS (BGP-Link State). Незалежно від обраного механізму отримання інформації, контролер SDN несе відповідальність за прийняття рішень щодо політик SR, які реалізують розширені функції або сервіси, такі як управління трафіком (TE), VPN або SFC (Service Function Chaining). Такий підхід забезпечує чітке відокремлення операцій площини даних від логіки сервісу, яка функціонує на рівні управління.

РОЗДІЛ 2. ДОСЛІДЖЕННЯ АСПЕКТІВ ВИКОРИСТАННЯ SEGMENT ROUTING

2.1 Цілі застосування Segment Routing

Тунелювання MPLS на основі IGP

Модель маршрутизації на основі джерела (Source-based Routing), реалізована в площині даних MPLS (Multiprotocol Label Switching), надає потужну можливість для тунелювання різноманітних сервісів, таких як VPN (Virtual Private Network), VPLS (Virtual Private LAN Service) та VPWS (Virtual Private Wire Service). Тунелювання здійснюється від вхідного граничного маршрутизатора провайдера (PE, Provider Edge) до вихідного PE, причому воно може бути виконане як із заданням явного шляху, так і без нього. Ключовою перевагою є те, що це досягається без необхідності зберігання стану на рівні пересилання або на рівні управління в проміжних вузлах мережі провайдера.

На рисинку 2.1, чотири вузли – A, CE1, CE2 та Z – належать до однієї й тієї ж VPN. У цьому сценарії CE2 (граничний маршрутизатор клієнта 2) оголошує маршрут до вузла Z маршрутизатору PE2. PE2 прив'язує локальну мітку L_Z до цього маршруту Z і використовує Багатопротокольний протокол граничного шлюзу (MPBGP) для поширення цього VPN-маршруту та його мітки до PE1. При цьому адреса наступного переходу встановлюється як $192.0.2.2$ (тобто локальна адреса PE2). PE1, отримавши інформацію, встановлює VPN-префікс Z у відповідній таблиці VPN-маршрутизації та пересилання (VRF, Virtual Routing and Forwarding) і дозволяє наступний перехід через тунель MPLS, який будується на основі IGP (Interior Gateway Protocol), до PE2.

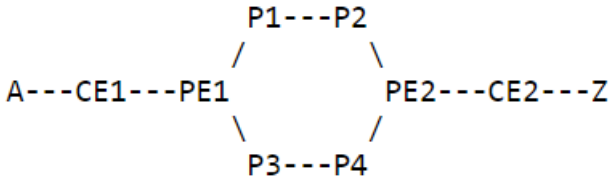


Рис. 2.1 – Тунелювання MPLS на базі IGP

Для ефективного вирішення проблем, притаманних поточним мережевим розгортанням, архітектура SPRING (Segment Routing) має забезпечувати функціональність переадресації від граничного маршрутизатора провайдера до граничного маршрутизатора провайдера (PE-to-PE) виключно на основі найкоротшого шляху IGP (Interior Gateway Protocol), і при цьому не вимагати додавання жодного іншого протоколу сигналізації. Основний принцип полягає в тому, що кожен пакет, який PE пересилає через мережу, повинен містити всю необхідну інформацію, отриману з бази даних топології, що дозволить доставити цей пакет до віддаленого PE без додаткових запитів чи обміну керуючою інформацією в процесі пересилання.

Traffic Engineering

Управління трафіком (Traffic Engineering, TE) — це загальний термін для позначення методів, які дають операторам мережі можливість контролювати, яким чином певні потоки трафіку обробляються та пересилаються. Для TE визначено різноманітні середовища та режими роботи, які можуть включати:

- один або кілька доменів;
- контроль доступу до смуги пропускання або його відсутність;
- централізоване чи розподілене обчислення шляху, тощо.

Застосування TE може варіюватися від обмеженого (тактичного) до широкомасштабного. Тактичний TE використовується обмежено для задоволення конкретних вимог програм чи клієнтів, або для усунення локальних обмежень пропускної спроможності. У цих ситуаціях ключова вимога — максимально знизити кількість протоколів сигналізації та кількість вузлів, які потребують спеціальних конфігурацій чи функцій. Широкомасштабний TE використовується для точного налаштування потоків даних на рівні додатків. У цьому випадку потрібна дуже висока масштабованість, особливо на проміжних вузлах.

Ключовим моментом є те, що модель маршрутизації на основі джерела (Source-based Routing), яка лежить в основі Сегментної Маршрутизації, дозволяє реалізувати управління трафіком без використання сигнального компонента.

Архітектура SPRING (Segment Routing) має підтримувати наступні важливі вимоги щодо управління трафіком:

- Можливість контролю пропускної спроможності (Bandwidth Control).
- Підтримка розподіленої або централізованої моделі керування (за допомогою PCE або SDN-контролера).
- Забезпечення непересічності шляхів у двоплощинних мережах (для резервування чи ізоляції).
- Підтримка проектування вихідних вузлів (Headend Design).
- Здатність здійснювати балансування навантаження між непаралельними каналами (тобто каналами, з'єднаними з різними сусідніми пристроями).
- Лімітування стану кожного сервісу та сигналізації на проміжних та кінцевих маршрутизаторах, з метою досягнення масштабованості (бажано нульового стану).
- ECMP-обізнаність (Equal Cost MultiPath-awareness), тобто можливість використання шляхів з рівною вартістю.
- Відмовостійкість вузла (Node Fault Tolerance) — політика управління трафіком не повинна бути жорстко прив'язана до конкретного основного вузла, збій якого міг би вплинути на сервіс.

У більшості сценаріїв Управління Трафіком (TE, Traffic Engineering) використовується варіант "вільного" маршруту (loose route), де більшість явних шляхів (explicit paths) може бути виражена через невелику кількість переходів (hops). Однак існують випадки використання, що вимагають застосування "строгого" параметра (strict route). У таких випадках кожен окремий перехід у явному шляху чітко вказується. Це неминуче призводить до формування довгого

списку переходів, який реалізується або як глибокий стек міток MPLS (у площині даних MPLS), або як довгий список IPv6-адрес (у площині даних IPv6).

Очевидно, що розгортання мережі, яке використовує довгі шляхи із суворою вихідною маршрутизацією, можливе лише за умови, що головний вузол (Headend) має достатні ресурси для створення екземплярів і обробки таких довгих явних шляхів. Як ефективна альтернатива, контролер може розкласти наскрізний (end-to-end) шлях на набір коротших шляхів. Кожен з цих коротших шляхів підтримується відповідним проміжним головним вузлом і оголошується за допомогою єдиного ідентифікатора (наприклад, Binding SID). Отже, об'єднання (або зшивання — splicing) цих ідентифікаторів шляхів створює схему стиснення, що дозволяє виразити наскрізний шлях за допомогою значно зменшеної кількості переходів у заголовку пакета.

Непересічність у двоплощинних мережах

Багато мереж побудовані за двоплощинною схемою, як показано на рис. 2.2 [9]:

- Кожна область агрегації k підключена до ядра двома С-маршрутизаторами C1k та C2k, де k позначає область;

- C1k є частиною площини 1 та області агрегації k;

- C2k є частиною площини 2 та області агрегації k;

- C1k має зв'язок із C2j тоді і лише тоді, коли k = j.

Основні вузли цього регіону безпосередньо пов'язані між собою. Канали між регіонами поєднують лише основні вузли однієї і тої ж площини (C1k має посилання до C1j тоді і тільки тоді, якщо C2k має посилання до C2j). Розподіл цих зв'язків залежить від топологічних властивостей ядра AS. Представлене вище правило проектування показує, що ці зв'язки виникають у обох базових площинах.

Можемо виділити загальне правило, що використовується в таких розгортаннях мережі. Вартість міжплощинного з'єднання (Cik - Cjk, де i != j) встановлюється таким чином, щоб маршрут до граничного пункту призначення із

заданої площини залишався всередині площини, якщо площина не розділена на частини.

Багато сучасних мереж проектуються за двоплощинною схемою (Two-Plane Architecture), як це показано на рисунку 2.2. Ця схема характеризується наступними ключовими ознаками:

- Кожна область агрегації k підключається до ядра мережі через два С-маршрутизатори: C1k та C2k, де k позначає конкретну область;
- Маршрутизатор C1k є частиною площини 1 та належить до області агрегації k;
- Маршрутизатор C2k є частиною площини 2 та належить до області агрегації k;
- Міжплощинний зв'язок існує лише між маршрутизаторами однієї області (C1k має зв'язок із C2j тоді й лише тоді, коли k = j тобто C1k пов'язаний лише з C2k);
- Основні вузли цього регіону (ядра) безпосередньо пов'язані між собою.

Канали між регіонами (у ядрі мережі) з'єднують лише основні вузли однієї і тієї ж площини: C1k має посилання до C1j тоді й лише тоді, якщо C2k має посилання до C2j. Такий розподіл зв'язків залежить від топологічних властивостей ядра Автономної Системи (AS). Вищезазначене правило проектування підкреслює, що критичні зв'язки виникають в обох базових площинах.

Можна сформулювати загальне правило, яке використовується в таких мережевих розгортаннях: вартість міжплощинного з'єднання (Cik - Cjk, де i=j) встановлюється таким чином, щоб маршрут до граничного пункту призначення із заданої площини залишався всередині цієї площини, за винятком випадків, коли площина виявляється розділеною на частини (партіціонованою).

У цьому сценарії оператору потрібна можливість розгортання згідно різних стратегій. Наприклад, граничний маршрутизатор A повинен мати можливість використовувати три наступні параметри:

- навантаження трафіку розподіляється будь-яким шляхом ECMP в мережі;
- навантаження трафіку розподіляється будь-яким шляхом ECMP в Plane1 мережі;

- навантаження трафіку розподіляється будь-яким шляхом ECMP в Plane2 мережі.

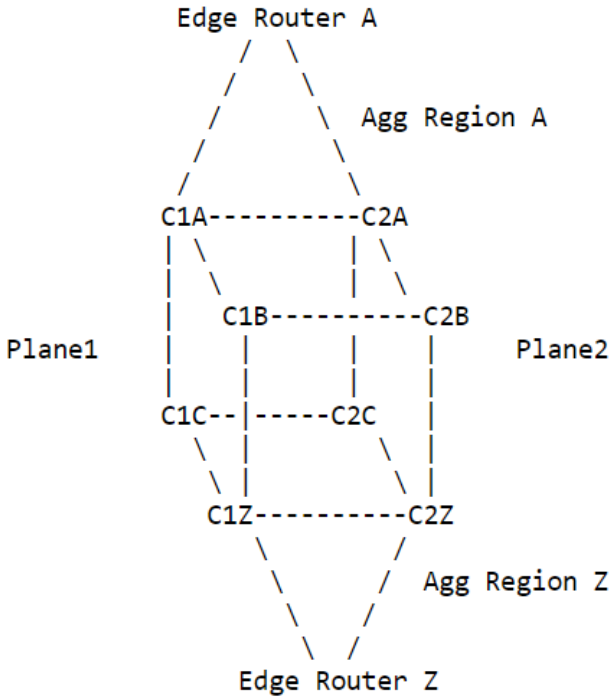


Рис. 2.2 – Двоплощинна розмежована мережа

Більшість трафіку від A до Z використовуватиме перший варіант, щоб ефективно використовувати пропускну здатність. Оператор може використовувати два інших варіанти для конкретного преміум-трафіку, який потрібно передавати неперетинаючись з іншим трафіком.

Архітектура SPRING, що підтримує цей варіант використання, повинна відповідати таким вимогам:

- відсутність зберігання стану кожної послуги та сигналізація на проміжних та кінцевих маршрутизаторах;

- ECMP-обізнаність;

- відмовостійкість вузлів: політика управління трафіком не прив'язана до конкретного основного вузла, збій якого може вплинути на службу.

У цьому сценарії оператору мережі потрібна гнучкість для розгортання трафіку згідно з кількома різними стратегіями. Граничний маршрутизатор A має бути здатним застосовувати три ключові параметри маршрутизації.

1. Загальний ECMP – навантаження трафіку розподіляється будь-яким шляхом ECMP (Equal Cost MultiPath) у всій мережі (обидві площини).

2. ECMP у площині 1 – навантаження трафіку розподіляється будь-яким шляхом ECMP виключно у площині 1 мережі.

3. ECMP у площині 2 – навантаження трафіку розподіляється будь-яким шляхом ECMP виключно у площині 2 мережі.

Більшість загального трафіку від вузла A до Z використовуватиме перший варіант (загальний ECMP) для ефективного використання пропускної здатності. Оператор може застосовувати два інші варіанти (площина 1 або площина 2) для конкретного преміум-трафіку, який необхідно передавати з гарантією неперетинання з іншими потоками (наприклад, для забезпечення ізоляції чи відмовостійкості).

Для успішної підтримки цього варіанту використання архітектура SPRING (Segment Routing) повинна відповідати наступним критичним вимогам:

- Нульовий/мінімальний стан - повна відсутність зберігання стану для кожної послуги та мінімізація сигналізації на проміжних та кінцевих маршрутизаторах (для масштабованості).

- Облік ECMP – наявність ECMP-обізнаності (ECMP-awareness) для ефективного використання шляхів з рівною вартістю.

- Відмовостійкість вузлів – політика управління трафіком має бути не прив'язана до конкретного основного вузла, збій якого може негативно вплинути на службу, забезпечуючи відмовостійкість вузлів.

Балансування навантаження між непаралельними каналами

Архітектура SPRING (Segment Routing) має надавати даному вузлу можливість розподіляти навантаження по кількох непаралельних каналах (тобто

каналах, які підключені до різних сусідніх маршрутизаторів), навіть якщо ці канали ведуть до різних безпосередніх сусідів. Ця функціональність є надзвичайно корисною для підтримки складних політик управління трафіком (Traffic Engineering, TE).

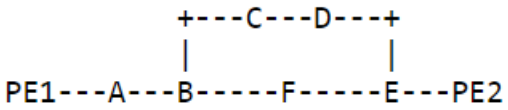


Рис. 2.3 – Декілька (непаралельних) сусідніх вузлів

У наведеному прикладі (рис. 2.3) оператор мережі вимагає, щоб граничний маршрутизатор PE1 розподіляв трафік, призначений для PE2, між двома шляхами з рівною вартістю: ABCDE та ABFE. Ключова вимога полягає в тому, що цей розподіл має відбуватися контрольованим чином. Більше того, оператору має бути надана можливість нерівномірно розподіляти трафік між цими шляхами, що відповідає механізму Weighted Equal Cost MultiPath (WECMP).

SRv6 Topology Independent Loop-Free Alternate FRR

Технології швидкого перенаправлення (FRR, Fast ReRoute) застосовуються мережевими операторами для оперативної реакції на збої каналів чи вузлів шляхом попереднього розрахунку резервних шляхів.

Архітектура SPRING (Segment Routing) повинна відповідати наступним ключовим вимогам щодо FRR:

- Підтримка FRR на будь-якій топології мережі.
- Попереднє обчислення та налаштування резервного шляху без будь-якої додаткової сигналізації (крім стандартних протоколів IGP/BGP).
- Підтримка загальних ризикованих обмежень (Shared Risk Link Group, SRLG).
- Підтримка захисту як вузлів, так і каналів.
- Підтримка запобігання утворення мікропетель (Micro-loop avoidance).

Існують різні методи FRR, які відрізняються залежно від топологічних обмежень:

- Звичайна технологія LFA (Loop-Free Alternative) вимагає, щоб принаймні один сусід був вільним від петель наступним переходом до місця призначення.
- Віддалений LFA (RLFA) вимагає наявності принаймні одного вузла (P-Node), який з'єднується з вихідним (Repair Node) і цільовим (Destination) вузлами за шляхом, що не проходить через будь-який несправний вузол.
- Топологічно-незалежне LFA FRR (TI-LFA FRR) використовує явний шлях (Segment List) для репрезентації резервного шляху, який не накладає обмежень на топологію і забезпечує більш надійний FRR у порівнянні з LFA чи RLFA.

SRv6 TI-LFA FRR застосовується як до сценаріїв політик SRv6 BE (Best Effort), так і до SRv6 TE (Traffic Engineering). У звичайних випадках пакети BE SRv6 можуть не містити інформацію SRH (Segment Routing Header). Однак у сценаріях TI-LFA FRR для інкапсуляції списку відновлення шляху (Repair Path Segment List) пакети SRv6 BE вимагають передачі інформації SRH.

На рисунку 2.4 показано випадок, коли вимоги RLFA не виконуються, і він не може обчислити резервний шлях, оскільки простір P (вузли, що мають безпосередній безпетльовий шлях до вузла відновлення) та простір Q (вузли, що мають безпетльовий шлях від цілі до вузла відновлення) не перетинаються. Якщо виникає помилка у каналі зв'язку між DeviceB та DeviceE, DeviceB пересилає пакети даних на DeviceC. DeviceC не є вузлом Q і не має запису FRR безпосередньо до IP-адреси призначення DeviceF. У цій ситуації DeviceC має провести переобрахунок шляху. Припускаємо, що вартість каналу між DeviceC та DeviceD становить 1000. DeviceC помилково вважає, що оптимальний шлях до DeviceF проходить через DeviceB (оскільки інформація про збій ще не поширилася). Як наслідок, DeviceC передає пакет назад на DeviceB, що призводить до утворення петлі та збою пересилання.

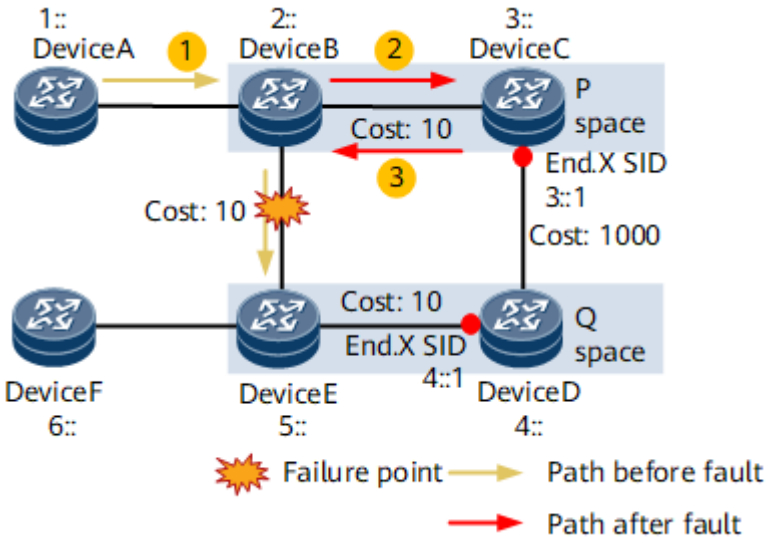


Рис. 2.4 – RLFA

Топологічно-незалежне LFA FRR (TI-LFA FRR) є передовим механізмом, що забезпечує надійний захист каналів та вузлів для SRv6. Його реалізація ілюструє, як Segment Routing використовує явні шляхи для відновлення зв'язку.

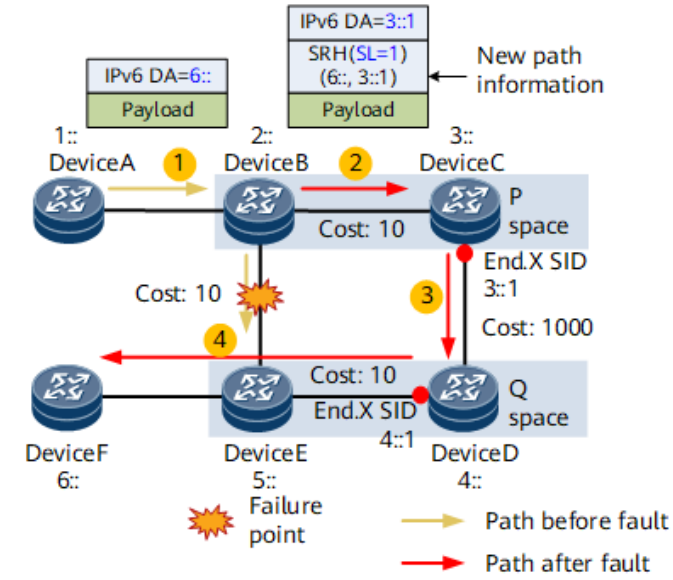


Рис. 2.5 – TI-LFA

У випадку, коли у каналі зв'язку між DeviceB та DeviceE виникає збій (Link Failure), DeviceB (вузол відновлення) безпосередньо включає в шлях пристрої, визначені у його резервних записах TI-LFA FRR. Це робиться шляхом додавання нової інформації про шлях (списку сегментів) до пакетів. Зокрема, додаються

ідентифікатори SID End.X DeviceC та DeviceD, що гарантує, що пакети даних можуть бути перенаправлені резервним шляхом (рис. 2.5). Поведінка End.X забезпечує пряме перенаправлення до вказаного сусіда (DeviceC), уникаючи петлі та забезпечуючи швидке відновлення

SRv6 Midpoint Protection

У сценаріях політики SRv6-TE (Segment Routing over IPv6 - Traffic Engineering) шлях пересилання пакетів даних, як правило, жорстко визначений через конкретні вузли або канали, які вони повинні пройти. Наприклад, у мережі, показаній на рисунку 2.6 , вузол A має відправити пакет вузлу F шляхом, який обов'язково проходить через вузол D. Оскільки резервний шлях (розрахований TI-LFA FRR) також вимагає проходження через вузол D, навіть якщо цей вузол вийде з ладу, TI-LFA FRR не може забезпечити захист у цьому випадку (рис. 2.6).

Для усунення збою захисту TI-LFA FRR, викликаного строгими обмеженнями вузла в сценарії політики SRv6-TE, застосовується механізм проксі-вузла пересилання (Forwarding Proxy Node). Цей вузол (розташований вище за маршрутом від несправної проміжної точки) бере на себе відповідальність від несправного проміжного пристрою для завершення пересилання пакетів уздовж визначеного шляху.

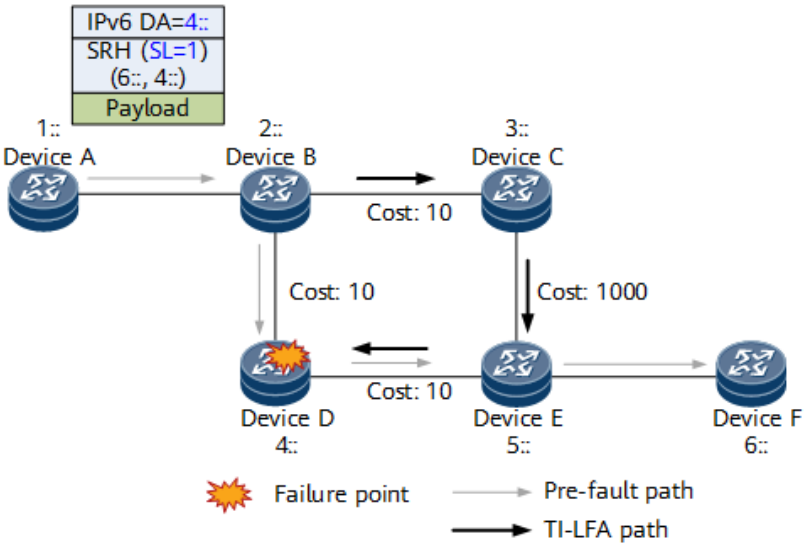


Рис. 2.6 – Збій захисту TI-LFA FRR

Коли проксі-вузол пересилання (Forwarding Proxy Node) виявляє, що інтерфейс наступного переходу для пакета несправний, і при цьому виконуються додаткові умови:

- адреса наступного переходу збігається з адресою призначення пакета;
- значення поля SL (Segments Left) більше за 0.

У цьому випадку проксі-вузол бере на себе відповідальність і призначає (імітує) на собі виконання поведінки End-вузла (кінцевої точки сегмента) для несправного проміжного пристрою. Ця імітована поведінка включає наступні кроки:

- Зменшення значення SL на 1 (Segments Left).
- Копіювання адреси наступного SID (Segment ID) зі списку сегментів у поле DA (Destination Address) зовнішнього заголовка IPv6.
- Пересилання пакета відповідно до інструкції, яка прив'язана до цього SID.

Несправний проміжний вузол обходиться, що ефективно забезпечує захист проміжних вузлів SRv6 у сценаріях TE (рис. 2.7).

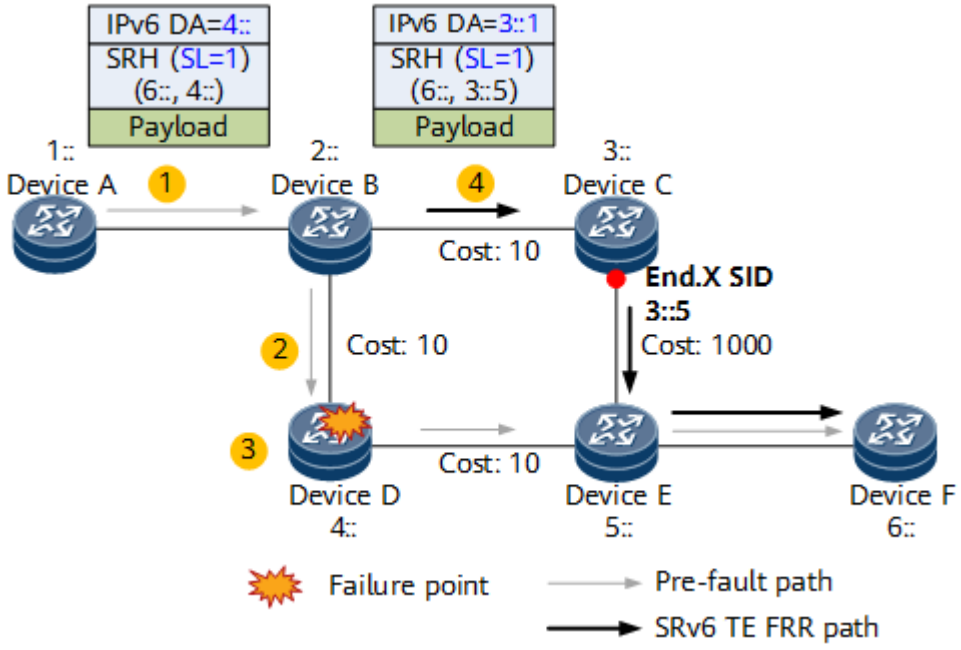


Рис. 2.7 – Захист проміжного вузла в SRv6

SRv6 Microloop Avoidance

Оскільки бази даних стану каналів IGP (LSDB, Link State DataBase) є розподіленими, мережеві вузли, на яких функціонує IGP (Interior Gateway Protocol), можуть синхронізуватись неоднаково (не одночасно). Це може призвести до утворення мікропетель (micro-loops) — тимчасових петель маршрутизації, які зникають лише після того, як усі вузли на шляху пересилання синхронізують свої LSDB. Мікропетлі спричиняють низку серйозних проблем, включаючи: втрату пакетів, збільшення затримки пакетів (latency), невпорядкованість пакетів (packet reordering) – тому їх слід уникати [2].

Сегментна маршрутизація (SR) використовує метод із мінімальним впливом на мережу для ефективного уникнення потенційних мікропетель. Якщо зміна топології мережі може потенційно викликати петлю, SR дозволяє мережевим вузлам вставляти списки сегментів SRv6, які гарантовано не утворюють петель, для коректного спрямування трафіку до адреси призначення. Нормальна переадресація трафіку (на основі IGP) відновлюється лише тоді, коли всі задіяні вузли мережі завершують синхронізацію своїх баз даних стану.

SRv6 Service Function Chaining

Ланцюжок сервісних функцій (SFC, Service Function Chaining) логічно з'єднує сервіси, що розташовані на мережевих пристроях, з метою надання впорядкованого набору послуг прикладному рівню. Додаючи інформацію про шлях сервісної функції (SFP, Service Function Path) до вихідних пакетів, SFC дозволяє пакетам гарантовано проходити через сервісні функції (SF, Service Functions) за заданим шляхом [2].

Пакети даних у мережі зазвичай проходять через різні SF для забезпечення користувачам безпечних, швидких та стабільних послуг. Ці SF можуть включати:

- брандмауери;
- системи запобігання вторгненням (IPS);
- прискорювачі додатків;

- пристрої трансляції мережевих адрес (NAT).

Послуги, необхідні користувачеві, можуть бути реалізовані лише тоді, коли мережевий трафік проходить через SF у точній послідовності, визначеній логікою послуги.

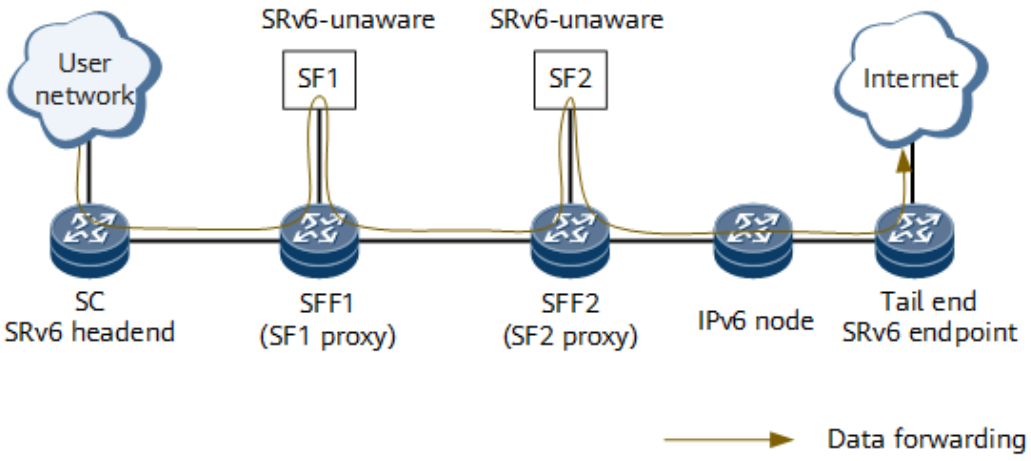


Рис. 2.8 – Типова мережа SRv6 SFC

Залежно від своєї обізнаності про інкапсуляцію SRv6, SF поділяються на дві категорії:

1. SF що підтримують SRv6 (ці SF можуть ідентифікувати та обробляти отримані пакети SRv6. У цьому випадку SID (Segment IDs) самих SF безпосередньо додаються до SFP для реалізації SFC).

2. SF що не підтримують SRv6 (ці SF не можуть ідентифікувати пакети SRv6 і тому відкидають їх після отримання. У цьому випадку для реалізації SFC необхідно налаштувати проксі-сервери SF (SF Proxies)).

На Рис. 2.8 показано типову мережу SRv6 SFC.

Захист шляху

Однією з основних вимог оператора мережі є забезпечення можливості неперетинності шляхів (path disjointness). Мережеві оператори розгортають інфраструктуру з топологіями, які дозволяють обчислювати повністю непересічні шляхи, тобто шляхи, які спільно не використовують жодного компонента (ні канал, ні маршрутизатор). Це дає змогу застосовувати оптимальну стратегію захисту [10].

Перша стратегія захисту полягає у виключенні будь-якого локального відновлення (наприклад, LFA/FRR) і замість цього використовує наскрізний захист шляху (end-to-end path protection). У цьому підході кожен шлях SPRING (Segment Routing) захищений другим непересічним шляхом SPRING. Це означає, що локальний захист на основному шляху не використовується.

Як приклад, псевдопровід (PW) від вузла A до вузла Z може бути "захищений" у напрямку від A до Z наступним чином: оператор налаштовує два шляхи SPRING — T1 (основний) та T2 (резервний) — які проходять від A до Z і є непересічними (рис. 2.9).

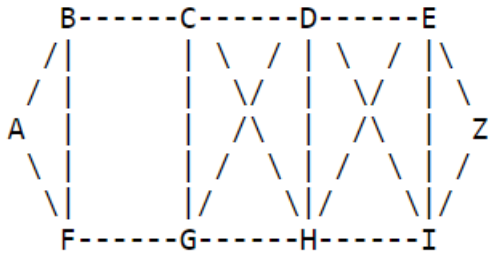


Рис. 2.9 – Захист шляху

При використанні двох непересічних шляхів ($T1$ та $T2$) для передачі трафіку можуть застосовуватися різні схеми:

- Захист 1+1 – одночасне використання, коли вхідний маршрутизатор надсилає ідентичний трафік як первинним (T1), так і вторинним (T2) шляхом.

- Багатоколійне поширення – одночасне використання, коли вхідний маршрутизатор ділить трафік між первинним та вторинним шляхами. Це може бути ECMP (з однаковою вартістю) або UCMP (з різною вартістю).

- Захист 1:1 – використовуються як основний (T1) і резервний (T2) шлях, де додатковий шлях активується тільки у разі збою основного.

У прикладі, де псевдопровід захищається від A до Z, T1 встановлюється як основний шлях AB, BC, CD, DE, EZ, а T2 — як резервний шлях AF, FG, GH, HI, IZ. Два шляхи обов'язково мають бути непересічними у своїх каналах, вузлах та групах

каналів загального ризику (SRLG, Shared Risk Link Group), щоб повністю задовольнити вимогу непересічності.

У випадку використання схеми основний/резервний шлях (1:1), коли основний шлях T1 працює, пакети PW (псевдопроводу) відправляються через T1. При відмові T1, пакети негайно перенаправляються резервним шляхом T2. Коли шлях T1 відновлюється, оператор може або дозволити автоматичне повернення трафіку на T1, або обрати повернення, кероване оператором. Як правило, перемикання з T1 на T2 має виконуватися у режимі швидкого перенаправлення (наприклад, менше 50 мілісекунд), хоча часові проміжки можуть варіюватися залежно від вимог послуги.

Вкрай важливо, щоб будь-який шлях (основний чи резервний) вигравав від наскрізного моніторингу/перевірки живучості. Існує кілька варіантів перевірки живучості:

- Живучість шляху відстежується на мережевому рівні (або головним вузлом, або централізованою системою моніторингу/мережевим контролером).

- Живучість з урахуванням послуги контролюється екземпляром самої послуги (наприклад, перевірка досяжності кінцевої точки).

З точки зору SPRING (Segment Routing), необхідно виділити наступні вимоги:

- Архітектура SPRING повинна забезпечувати спосіб обчислення шляхів, які не захищені методами локального відновлення (як показано на прикладі шляхів T1 та T2 для наскрізного захисту).

- Архітектура SPRING повинна забезпечувати спосіб створення пар повністю неперетинних шляхів у топології на основі обраної стратегії захисту (захист каналу, вузла або SRLG) та дозволяти перевіряти чи переобчислювати ці шляхи при мережевих подіях.

- Архітектура SPRING повинна забезпечувати наскрізну перевірку життєздатності шляхів, які базуються на SPRING.

Матриці трафіку

Матриця трафіку T(N, M) (також відома як матриця попиту) визначає обсяг трафіку, який входить до мережі у вузлі N і залишає мережу у вузлі M. Вузли N та M є граничними вузлами на довільно визначеній границі мережі (TM Boundary), де постачальники послуг бажають вимірювати потоки трафіку. Матриця трафіку містить усі запити, що перетинають цю границю TM. Вона має стільки рядків, скільки є вхідних граничних вузлів, і стільки ж стовпців, скільки вихідних граничних вузлів на цій границі. Комірка матриці у рядку N і стовпці M позначається як Попит D(N, M) (або T(N, M)). Іншими словами, матриця трафіку надає для кожної точки входу N і кожної точки виходу M з мережі обсяг трафіку T(N, M) за заданий проміжок часу.

Для вимірювання матриці трафіку вузли мережі SR визначають свої інтерфейси як внутрішні чи зовнішні. Коли вузол N отримує пакет, який призначений для віддаленого префіксу SID M, вузол N підтримує наступні лічильники [11]. Ці лічильники враховують операції:

- push (додавання мітки/сегмента);
- pop (вилучення мітки/сегмента);
- swap (заміна мітки).

Проблема визначення матриць трафіку є добре відомою в комп'ютерних мережах. Обчислення T(N, M) за допомогою прямих вимірювань IP-трафіку традиційно вважалося надто ресурсомістким для маршрутизаторів і генерувало величезну кількість контрольного трафіку. Рішення полягало в агрегації кількості трафіку на каналі, що спрощувало проблему обмежень, але вимагало застосування евристики для відновлення матриці, що призводило до розрахунку попиту на основі багатьох припущень.

Сегментна маршрутизація (SR) пропонує ефективне проміжне рішення. Вона дозволяє агрегувати трафік на основі сегмента вузла призначення (Node

Segment ID). Використовуючи таку статистику трафіку, можна отримати детерміновану матрицю трафіку [14].

Матриця трафіку — це чудовий інструмент для планувальника мережі (Network Planner). Вона дозволяє йому точно знати, звідки трафік входить до його мережі і звідки він виходить. Маючи цю важливу інформацію, оператор може робити обґрунтовані припущення щодо того, як слід планувати майбутнє своєї мережі (наприклад, для оптимізації пропускної здатності та розширення).

Поділ мережі на сегменти

У міру того, як все більше постачальників послуг та підприємств використовують єдину мережеву інфраструктуру для підтримки зростаючої кількості сервісів, можливість налаштовувати транспортування трафіку відповідно до потреб додатків стає критично важливою. Це вимагає створення мережевих сегментів (Network Slices) з різними характеристиками, які можуть співіснувати поверх загальної фізичної інфраструктури [12].

Поділ мережі (Network Slicing) призначений для створення наскрізної (end-to-end) розділеної мережевої інфраструктури, яку можна використовувати для забезпечення диференційованої поведінки підключення з метою виконання вимог різноманітного набору послуг. Служби, що належать до різних мережевих сегментів, можуть бути як повністю розрізненими (ізольованими), так і спільно використовувати різні частини мережевої інфраструктури. Розділення мережі є однією з ключових вимог стандартів 5G.

Сегментна маршрутизація (SR) дозволяє постачальникам послуг підтримувати сегментування мережі без додаткового протоколу, окрім розширень SR IGP. Мережа в цілому, працюючи розподіленим та повністю автоматизованим чином, може спільно використовувати один фізичний ресурс інфраструктури для кількох віртуальних сервісів (сегментів):

- Один сегмент безперервно оптимізується для недорогого транспортування (Best Effort).

- Другий сегмент безперервно оптимізується для транспортування з малою затримкою (Low Latency).

- Третій сегмент організований для підтримки непересічних послуг (Disjoint Paths), і т. д.

Мета оптимізації кожного з цих сегментів програмується оператором. Специфікація сегментної маршрутизації містить різноманітні будівельні блоки, необхідні для створення мережевих сегментів. До них належать:

- Політика SR із застосуванням гнучкого алгоритму (Flex-Algo) або без нього.

- TI-LFA (Topology-Independent LFA) із захистом (50 мс) в основі сегменту.

- SR VPN (для забезпечення ізоляції трафіку).

- Програмування служб SR (NFV, SFC) для реалізації ланцюжків функцій.

- Експлуатація, адміністрування та управління (OAM) та управління продуктивністю (PM).

- QoS (Quality of Service) із використанням DiffServ (Differentiated Services).

- Ідентифікація сегментів мережі без збереження стану (Stateless Segment Identification).

- Оркестрування через контролер (Centralized Orchestration).

Кожен із цих будівельних блоків функціонує незалежно. Їх функціональні можливості можна комбінувати, щоб задовольнити вимоги постачальника послуг щодо поділу мережі на сегменти. Зовнішній контролер відіграє вирішальну роль в об'єднанні цих будівельних блоків у єдину службу утворення сегментів.

2.2 Застосування Segment Routing в різних мережах (SPRING)

Segment Routing in IP/MPLS Networks для малого офісу

Впровадження протоколу IPv6 у сегменті SOHO (Small Office, Home Office) відкриває доступ до значно розширеного простору глобально маршрутизованих адрес для всього спектру підключеного кінцевого обладнання. Типова архітектура малого офісу на базі IPv6, що характеризується наявністю декількох вихідних

каналів зв'язку та відповідних префіксів, делегованих провайдером послуг, автоматично забезпечує хостам присвоєння множинних адрес IPv6. При цьому організація маршрутизації від джерела до призначення (Source-Address Dependent Routing) у масштабах малого офісу гарантує коректну передачу пакетів за межі локального периметра через суворо визначений вихідний інтерфейс, який асоціюється з відповідним делегованим префіксом даного каналу зв'язку [8].

Інтеграція архітектури SPRING у середовище SOHO створює розширені можливості для детермінованого спрямування трафіку за заданими траєкторіями, ініційованими або кінцевими хостами, або клієнтським граничним маршрутизатором (CPE). У випадку, коли на граничному маршрутизаторі активовано функцію вибору шляху на основі джерела, саме цей пристрій бере на себе відповідальність за класифікацію трафіку та його подальше спрямування коректним маршрутом. Якщо ж хости всередині SOHO оперують явно визначеними правилами вибору джерела, процес класифікації може ефективно базуватися на адресі джерела або асоційованій точці виходу з мережі, що дозволяє виключити необхідність застосування неявних методів класифікації, таких як глибокий аналіз пакетів (DPI). У ситуації, коли трафік спрямовується конкретним шляхом безпосередньо за ініціативою хоста, критично важливою є наявність інформації про те, які саме сегменти мережі здатні обробляти заголовок SPRING; такі дані можуть бути надані в конфігурації хоста як атрибут налаштованої IP-адреси.

Технічна здатність здійснювати селективне спрямування трафіку на відповідний вихідний інтерфейс або задіювати специфічний тип фізичного середовища передачі даних — наприклад, енергоефективні інтерфейси, Wi-Fi, дротові з'єднання Ethernet, фемтостільникові шлюзи, Bluetooth, технології Multimedia over Coax Alliance (MoCA) або HomePlug тощо — безпосередньо в межах домогосподарства становить значний практичний інтерес для прикладного програмного забезпечення, що функціонує в екосистемі SOHO.

Детерміноване спрямування інформаційних потоків до конкретної точки виходу з мережі є доцільним для реалізації низки експлуатаційних сценаріїв, а саме:

- для забезпечення суворої відповідності регуляторним та нормативним вимогам;
- для виконання специфічних завдань сервісу, що жорстко прив'язаний до конкретного каналу зв'язку;
- для реалізації стратегій оптимізації витрат, зумовлених лімітами на обсяг даних або тарифікацією трафіку;
- для логічного розмежування потоків особистого та корпоративного трафіку в домогосподарствах, де працюють один або декілька віддалених співробітників.
- для забезпечення доступу до унікальних послуг конкретного інтернет-провайдера, які відрізняються від сервісів альтернативних постачальників послуг

Службова інформація, інкапсульована в заголовок SPRING, незалежно від того, чи була вона введена кінцевим хостом, клієнтським граничним маршрутизатором або обладнанням мережі доступу провайдера, зберігає свою актуальність і на віддалених кінцях телекомунікаційної інфраструктури. Наприклад, програмне забезпечення, розгорнуте на кінцевому хості з підтримкою додатків у центрі обробки даних (DC), може використовувати заголовок SPRING як транспортний канал для передачі метаданих, що впливають на алгоритми обробки безпосередньо в ЦОД, уможливлюючи управління на рівні додатків та балансування навантаження без опори на методи неявної класифікації на вхідних шлюзах. Більше того, враховуючи тенденцію до тотального шифрування трафіку додатків, можливість експліцитно витягувати та включати в заголовок SPRING саме той обсяг інформації, який необхідний мережі та центру обробки даних для коректного балансування та керування трафіком, набуває критичного значення.

Segment Routing in IP/MPLS Networks для мережі доступу

Мережі доступу відіграють фундаментальну роль у забезпеченні двонаправленого транспортування гетерогенних потоків даних між інфраструктурою постачальника телекомунікаційних послуг та кінцевим абонентським обладнанням у домашньому середовищі. З метою оптимізації використання ресурсів смуги пропускання та реалізації суміжних цілей керування мережею, провайдери часто імплементують механізми асоціювання специфічних класів трафіку з виділеними фізичними або логічними каналами передачі даних.

У контексті архітектури мереж кабельного доступу, які функціонують відповідно до специфікацій DOCSIS (Data Over Cable Service Interface Specification), ці канали реалізуються у формі потоків обслуговування (Service Flows), які з технічної точки зору доцільно розглядати як дискретні канали передачі даних. Така архітектурна сегрегація надає оператору можливість ефективно розділяти контент за типами та застосовувати до кожного виділеного каналу специфічні політики обробки, такі як шейпінг трафіку (traffic shaping), забезпечення відповідності регуляторним вимогам та диференційована тарифікація послуг. У низхідному напрямку передачі даних (downstream) ці логічні канали можуть бути представлені окремими потоками послуг DOCSIS, групами сервісів або бути закріпленими за конкретними частотними каналами з квадратурною амплітудною модуляцією (QAM).

Аналогічна методологія застосовується і до висхідного трафіку (upstream), що ініціюється в домашній мережі та прямує до ядра мережі провайдера, де виникає потреба у мапінгу пакетів на певні висхідні канали пропускної здатності. У цьому сценарії службова інформація, інкапсульована в заголовок SPRING, виступає ефективним ідентифікатором цільового каналу для кожного конкретного пакета. Ключовою перевагою такого підходу є те, що пристрій доступу не потребує глибокого аналізу вмісту пакета (payload) для виконання класифікації; його функціональне навантаження зводиться виключно до інтерпретації семантики

заголовка SPRING та виконання процедури зіставлення з відповідним цільовим каналом згідно із заданими правилами маршрутизації [8].

Segment Routing in IP/MPLS Networks для центрів обробки даних

Стратегічним напрямком розвитку для багатьох операторів центрів обробки даних (ЦОД) стає повна міграція інфраструктури з застарілого протоколу IPv4 на архітектуру з нативною підтримкою IPv6 (IPv6-only). Такий перехід є необхідною відповіддю на критичне вичерпання адресного простору IPv4 та імператив забезпечення гіпермасштабованості мережевих ресурсів. У межах такого модернізованого середовища технологія маршрутизації від джерела, реалізована на базі SRv6, стає ключовим інструментом для детермінованого управління трафіком (Traffic Engineering), дозволяючи спрямовувати інформаційні потоки за суворо визначеними топологічними маршрутами. Важливою особливістю є те, що специфікація такого маршруту може виходити за межі простої переадресації пакетів, включаючи інструкції для виконання заданих сервісних функцій на одному або декількох транзитних вузлах тракту [8].

Паралельно з цим, фундаментальною архітектурною вимогою до сучасних хмарних обчислювальних середовищ є здатність надавати в оренду ізольовані віртуальні мережі з високим потенціалом до масштабування (мультитенантність). У подібних сценаріях механізми Сегментної Маршрутизації дозволяють побудувати гнучку мережеву абстракцію, яка не лише забезпечує спрямування трафіку за специфічним вектором, але й слугує для точної ідентифікації та розмежування конкретних обчислювальних вузлів, окремих орендарів (теннантів) та необхідних мережевих функцій, гарантуючи при цьому логічну ізоляцію сервісів.

Segment Routing in IP/MPLS Networks в мережах доправлення контенту (CDN)

Експоненціальне зростання екосистеми мультимедійних додатків та стрімке поширення термінального обладнання з нативною підтримкою IP-відеопотоків

спричинили безпрецедентне навантаження на транспортну інфраструктуру телекомунікаційних операторів. У відповідь на необхідність нівелювання як капітальних, так і операційних викликів, пов'язаних з обробкою масивних обсягів відеотрафіку, а також прагнучи забезпечити конвергенцію традиційних сервісів телемовлення з новітніми мультискрін-платформами, провайдери послуг дедалі активніше імплементують архітектуру мереж доставки контенту (CDN) [8].

Численні наукові дослідження емпірично підтверджують ефективність організації кешуючих вузлів у ієрархічні топології, які структурно ізоморфні ієрархічній природі глобальної мережі Інтернет. У межах такої архітектури кожен окремий вузол кешування встановлює пірингові взаємозв'язки із суміжними елементами системи. Ці відносини класифікуються за двома основними векторами: вертикальним, що визначає взаємодію з батьківським вузлом (розташованим на вищому рівні ієрархії), та горизонтальним, що описує комунікацію із сусідніми вузлами (що знаходяться на тому ж рівні топології). Для побудови високоефективної архітектури кешування, як правило, застосовується багаторівнева модель ієрархії.

У мережевому середовищі, де кожна дискретна система кешування однозначно адресується за допомогою унікальної IPv6-адреси, виникає технічна можливість формування впорядкованого списку (Segment List), що репрезентує послідовність кешів у ієрархії. Алгоритм обробки запиту передбачає послідовну перевірку наявності запитуваного контенту на кожному вузлі, визначеному у списку. У випадку успішного виявлення контенту (сценарій "cache hit") виконання послідовності негайно припиняється, навіть за наявності подальших вузлів у списку; у протилежному випадку система ініціює звернення до наступного елемента у визначеному ланцюжку.

Segment Routing in IP/MPLS Networks у базових мережах

В умовах, коли сумарний обсяг трафіку, що генерується в телекомунікаційних мережах, демонструє стійку тенденцію до експоненціального

зростання, оператори зв'язку стикаються з безпрецедентними технологічними викликами. Ситуація ускладнюється конвергенцією різнорідних типів трафіку, кожен з яких характеризується унікальними профілями навантаження та жорсткими вимогами до якості обслуговування (QoS), в межах єдиної уніфікованої мережевої архітектури.

Стратегічним вектором розвитку для багатьох операторів стає поетапна міграція або безпосереднє проектування опорних мереж (Backbone) виключно на базі протоколу IPv6 (IPv6-only) у найближчій перспективі. У цьому контексті розглядається впровадження механізмів маршрутизації з явно визначеним шляхом, що базується на аналізі IPv6-адреси джерела, як ефективного інструменту оптимізації використання мережевих ресурсів. Враховуючи поширену практику розподілу IPv6-префіксів абонентам за географічним принципом, префікс користувача фактично виконує роль ідентифікатора регіональної приналежності клієнта. У такій архітектурі граничні вузли мережі (Edge Nodes) отримують можливість використовувати адреси джерел для інтелектуального керування трафіком, спрямовуючи його за інженерно визначеними маршрутами, які можуть свідомо відрізнятися від метрично оптимальних шляхів, розрахованих стандартними протоколами маршрутизації.

Імператив налаштування маршрутів на основі джерела з обов'язковим проходженням через специфічні проміжні вузли мережі диктується комплексом інженерних та бізнес-вимог. По-перше, оператори прагнуть агрегувати ресурсоємні потоки, такі як відеотрафік, на виділених магістральних каналах високої ємності, що дозволяє уникнути необхідності тотального масштабування пропускної здатності всієї мережевої топології. По-друге, існує потреба у детермінованому встановленні маршрутів із мінімальною затримкою для критично важливих додатків реального часу. По-третє, механізм дозволяє здійснювати гранулярний вибір конкретних точок обміну трафіком (Peering Points) серед множини доступних альтернатив. Крім того, маршрутизація від джерела забезпечує

доступ до специфічних серверних потужностей, розміщених на певних майданчиках, досяжність яких не завжди гарантується стандартними алгоритмами найкоротшого шляху. Нарешті, цей підхід є фундаментом для реалізації концепції "двоплощинної мережі" (Dual-Plane Network), що вимагає гарантованої топологічної ізоляції (неперетинності) шляхів для забезпечення відмовостійкості.

Реалізація всіх вищезазначених сценаріїв вимагає імплементації повнофункціональних інструментів інженерії трафіку (Traffic Engineering), адаптованих для роботи в нативному середовищі IPv6 [8].

Segment Routing in IP/MPLS Networks в мобільній мережі (5G)

Мобільні мережі демонструють зростаючу операційну складність. З одного боку, спостерігається неперервне збільшення обсягів трафіку та посилення вимог до мінімізації затримки (latency); з іншого боку, поява нових сценаріїв використання, зокрема, розподілена інфраструктура віртуалізації мережевих функцій (NFVi), суттєво ускладнюють процеси експлуатації мережевих ресурсів.

Поточна архітектура мобільних мереж характеризується відсутністю інтегрованого підходу до базового транспортування трафіку. Користувацька площина (User Plane) жорстко сегментована за технологіями: радіодоступу, опорних та службових мереж, з'єднаних між собою за допомогою тунельних протоколів відповідно до функціональних ролей у площині, таких як вузли доступу та вузли прив'язки. Ці архітектурні чинники, засновані на жорсткому поділі, значно ускладнюють для оператора завдання оптимізації та ефективного керування каналом передачі даних.

Спостерігається загальна тенденція переходу додатків на використання протоколу IPv6, і мережеві оператори дедалі більше застосовують IPv6 як базову технологію IP-маршрутизації. SRv6 (Segment Routing over IPv6), будучи втіленням сегментної маршрутизації в площині даних IPv6, інтегрує шляхи даних як на прикладному рівні, так і на нижчому транспортному рівні в межах єдиного протоколу. Це дозволяє операторам спрощено оптимізувати мережу та, що

критично важливо, виносить стан пересилання пакетів за межі мережевих вузлів (stateless). SRv6 також демонструє високу придатність для віртуалізованих середовищ, включаючи мережі VNF/CNF-to-VNF/CNF. Ця технологія вже була успішно розгорнута в десятках промислових мереж.

Концептуально SRv6 визначає парадигму мережевого програмування. У контексті мобільності SRv6 здатна забезпечити необхідну поведінку користувацької площини, замінюючи традиційні тунельні протоколи. Використовуючи інформацію транспортного рівня, гнучкість заголовка, а також можливість безпосереднього включення сервісних функцій, SRv6 дозволяє здійснити наскрізну оптимізацію всієї мобільної площини даних [13].

Segment Routing для хмарних WAN мереж

З огляду на неперервне зростання обсягів трафіку в Центрах Обробки Даних (ЦОД), обумовлене розподіленими кластерами та динамічними моделями трафіку додатків, більшість постачальників хмарних послуг інвестували у створення глобальних мереж (Cloud WAN) для об'єднання своїх ЦОД, через які сьогодні проходять петабайти даних (і ця кількість постійно збільшується). Канали WAN, як правило, є дороговартісними, що вимагає забезпечення високого рівня ефективності та якості використання цих каналів. Хмарні мережі потребують надзвичайно деталізованого контролю (granular control) для керування трафіком на основі цілісного, глобального уявлення про всю наскрізну мережеву інфраструктуру.

Для досягнення цієї мети хмарним мережам необхідний функціонал для обчислення оптимального маршруту для кожного потоку трафіку на основі комплексного набору критеріїв, що включає глобальну топологію, метричні відстані, доступність смуги пропускання, поточні умови навантаження, тип трафіку, чутливість до затримок, а також корпоративну бізнес-логіку. Саме тут Сегментна Маршрутизація (SR), у поєднанні із зовнішнім контролером SDN, що приймає централізовані рішення щодо обчислення та повторної оптимізації шляху,

виступає як надзвичайно привабливе та ідеальне рішення, повністю відповідаючи парадигмі інтелектуальної програмно-керованої маршрутизації від джерела (AER) (Рис. 2.10) [4].

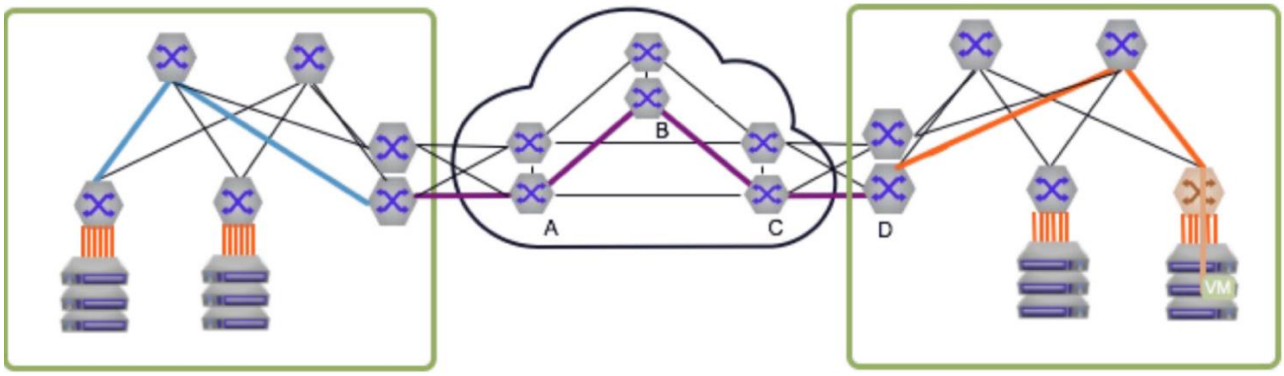


Рис. 2.10 – SR у хмарній глобальній мережі

Завдяки архітектурному усуненню необхідності підтримки складного стану інженерії трафіку (TE state) для кожного потоку та виключенню застосування додаткових протоколів сигналізації, фізична мережа тепер може бути цілеспрямовано спроектована для забезпечення високопродуктивної маршрутизації та комутації. Це також дозволяє ефективно функціонувати розподіленим протоколам IP-маршрутизації та забезпечувати розширену мережеву телеметрію, яка необхідна для підтримки цього нового, програмно-керованого підходу до управління трафіком.

Телеком-провайдери застосовують різноманітні підходи до впровадження сегментної маршрутизації у своїх глобальних мережах. Деякі з них створюють ізольовані домени (острови) SR навколо своїх існуючих мереж, які використовують LDP/RSVP-TE, що вимагає забезпечення сумісності (interworking) SR із цими традиційними протоколами. Інша поширена стратегія полягає у розгортанні паралельних мереж SR, які функціонують одночасно з традиційними інфраструктурами інженерії трафіку. Третій, більш інтегрований підхід, передбачає одночасне включення SR та RSVP-TE в межах однієї мережі з подальшим розподілом трафіку між тунелями, створеними обома технологіями.

Segment Routing для CDN

Постачальники мереж доставки контенту (CDN) мають критичну потребу у програмно-керованих механізмах управління трафіком (Software-Defined Traffic Management), необхідних для вибору оптимального шляху виведення контенту з їхніх центрів обробки даних (ЦОД) до кінцевих споживачів. Подібно до сценаріїв використання в глобальних хмарних мережах, це рішення обґрунтовується багатофакторними обмеженнями, що включають вартість внутрішньомережевих з'єднань, географічні детермінанти, класифікацію типу трафіку та фізичне місце розташування кінцевого абонента (рис. 2.11) [4].

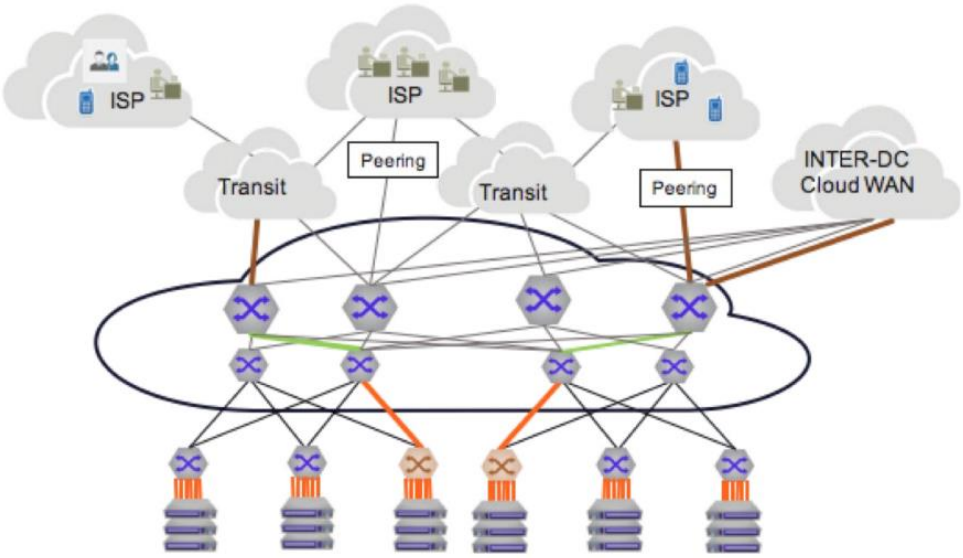


Рис. 2.11 – SR для CDN

У той час як деякі постачальники CDN покладаються виключно на IP-орієнтовані рішення, акумулюючи повний набір BGP-маршрутів у централізованому контролері, інші використовують інкапсуляцію міток MPLS для ідентифікації та примусового вибору певної точки виходу на периметрі мережі. Враховуючи, що внутрішні мережеві вузли часто функціонують лише як IP-маршрутизатори, деякі клієнти застосовують інкапсуляцію MPLSoGRE (MPLS over Generic Routing Encapsulation) для спрямування трафіку на конкретний інтерфейс граничного пристрою. Це зіставлення традиційно налаштовується статично через

інтерфейс командного рядка або API. Однак, завдяки вдосконаленням Сегментної Маршрутизації для BGP (або використання BGP-LU - Labeled Unicast), зіставлення мітки MPLS з вихідним каналом може тепер динамічно передаватись контролеру, що значно підвищує ефективність масштабування цієї операції.

Великі CDN із численними вихідними граничними маршрутизаторами зазвичай вимагають високомасштабованих кінцевих рівнів проектування для забезпечення зв'язку цих граничних вузлів із вузлами кешування. Щоб повністю виключити необхідність будь-якої переадресації на основі IP у проміжних вузлах та вирішити проблему масштабування IP-маршрутів, одночасно зберігаючи низькі експлуатаційні накладні витрати технології MPLS, Сегментна Маршрутизація MPLS у поєднанні з IGP або BGP пропонує архітектурно елегантне рішення, яке ідеально підходить для таких великомасштабних CDN.

Segment Routing в мережі метро

Мережа метрополітену являє собою ще один приклад мережевої інфраструктури, яка може отримати значну вигоду від впровадження транспортного рішення, базованого на Сегментній Маршрутизації (SR). У цьому конкретному випадку ключова потреба продиктована не стільки складною інженерією трафіку (TE), скільки необхідністю використання маршрутизації сегментів для вирішення проблеми захисту топології "кільце" та забезпечення покращеного резервування MPLS та ефективного механізму ECMP (Equal-Cost Multi-Path), оскільки все більше міських мереж агрегації набувають структури типу L-S (Link-State).

Сегментна маршрутизація як транспортне рішення є надзвичайно привабливою, оскільки вона першочергово здатна забезпечити швидкий, альтернативний маршрут без петель (TI-LFA - Topology-Independent Loop-Free Alternate) у разі виникнення будь-якого збою. Крім того, може бути легко застосована концепція SID Anycast для надання простого та елегантного рішення,

що забезпечує відмовостійкий шлюз для виходу за межі кільцевих топологій, як це наочно продемонстровано на Рис. 2.12 [4].

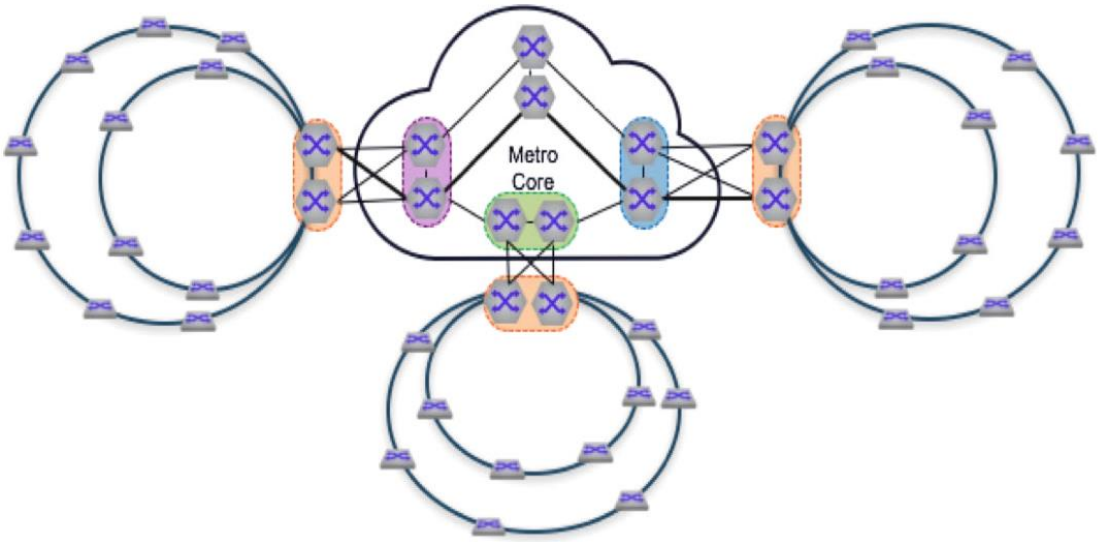


Рис. 2.12 – SR в мережі метро

Запропоноване просте рішення, засноване на сегментній маршрутизації (SR), ефективно усуває ключові архітектурні проблеми, притаманні традиційним кільцевим мережам. Сегментна маршрутизація забезпечує зручний та прямий спосіб інтеграції цих кільцевих сегментів у вже існуючі базові середовища, які використовують протоколи LDP або RSVP-TE, за допомогою спеціалізованої функції зіставлення SR-LDP (SR-LDP Interworking) або через використання зв'язуючого SID (Binding SID).

2.3. Програмні реалізації Segment Routing

Технологія Сегментної Маршрутизації на основі MPLS (SR-MPLS) досягла високого рівня зрілості розробки та отримує надійну підтримку від ключових виробників мережевого обладнання, включаючи таких провайдерів, як Cisco, Huawei та Juniper. Перехід до SR-MPLS може бути реалізований поступово в існуючих магістральних IP-MPLS мережах, оскільки це переважно вимагає лише оновлення програмного забезпечення мережевих пристроїв. Оператори можуть

здійснювати міграцію на SR-MPLS з метою спрощення операцій площини керування та покращення загальної масштабованості мережі.

Щодо площини даних SRv6, існують дві основні реалізації з відкритим вихідним кодом для програмних маршрутизаторів:

- реалізація інтегрована в ядро операційної системи Linux;
- реалізація виконана в рамках проекту FD.io VPP (Vector Packet Processing).

Існують інші реалізації з відкритим вихідним кодом, переважно пов'язані з дослідницькою діяльністю, а також апаратні реалізації SRv6, які вже успішно розгорнуті у великих виробничих мережах [1].

Можливості Segment Routing в ядрі Linux

Можливості SRv6 були вперше інтегровані в ядро Linux, починаючи з версії 4.10. Ця версія забезпечила підтримку окремих режимів роботи головного вузла політики SRv6 (раніше відомих як режими транзиту), зокрема H.Insert та H.Encaps (інкапсуляція). Дії головного вузла політики SRv6 були реалізовані через механізм полегшеного тунелю Linux (LWT). Утиліта користувацького простору iproute2 була розширена для підтримки додавання локальних ідентифікаторів (localsid), асоційованих з діями головного вузла політики SRv6. Ці локальні ідентифікатори та дії додаються до основної таблиці маршрутизації ядра як запис FIB IPv6.

Ядро 4.14 стало наступною важливою віхою, додавши набір поведінок кінцевих точок SRv6 шляхом імплементації нового типу LWT. Підтримувані варіанти поведінки кінцевої точки включали: End.X (маршрутизація до кінцевого вузла), End.T (маршрутизація до таблиці), End.DX2/DX4/DX6 (декапсуляція до L2/IPv4/IPv6), End.DT6 (декапсуляція до таблиці IPv6), End.B6 та End.B6.Encaps (для пов’язуючих SID). Крім того, були додані нові варіанти поведінки головного вузла політики SRv6, наприклад, H.Encaps.L2. Відповідно, була розширена і реалізація iproute2.

Можливості SRv6 в ядрі Linux були значно розширені у ядрі 4.16, зокрема, шляхом інтеграції з інфраструктурою netfilter. Було додано нове розширення зіставлення iptables під назвою srh (Segment Routing Header) для забезпечення відповідності полів заголовка SRH. Це розширення є частиною брандмауера SERA і підтримує зіставлення всіх полів SRH. Реалізація утиліти користувацького простору iptables була доповнена новою бібліотекою (libip6t_srh), що дозволяє визначати правила iptables з параметрами srh.

Ядро Linux 4.18 додало ще більше функцій як до основного стеку SRv6, так і до структури netfilter. У netfilter розширення srh match було вдосконалено, щоб забезпечити відповідність попередньому SID, наступному SID та останньому SID в SRH. Утиліта користувача iptables також була оновлена для підтримки нових параметрів зіставлення. Крім того, до стеку Linux SRv6 додана нова функція для підтримки користувацьких мережевих функцій SRv6, реалізованих у вигляді невеликих програм eBPF (extended Berkeley Packet Filter). З точки зору реалізації, в інфраструктуру SRv6 був доданий новий доступ для BPF, який мережеві оператори можуть використовувати для приєднання невеликих програм, написаних на C, до SID SRv6. Ці програми мають прямий доступ до кадрів Ethernet. Також були створені спеціалізовані допоміжні функції SRv6-BPF, що дозволяють функціям End.BPF виконувати основні дії SRv6 (End.X, End.T та інші) або додавати TLV (Type-Length-Value) поля. Це надає можливість реалізувати настроювану поведінку головного вузла політики SRv6 (в основному для розширення політик інкапсуляції, реалізованих ядром).

На відміну від SRv6, SR-MPLS не отримав аналогічної уваги при розробці ядра Linux. Усі доступні функції SR-MPLS в основному пов'язані з базовою переадресацією MPLS та стали доступні, починаючи з версії ядра 4.1. Зокрема, у ядрі 4.1 з'явилася поведінка маршрутизатора MPLS з комутацією за мітками (LSR). Можливості MPLS були розширені пізніше у ядрі версії 4.3 шляхом додавання платформи LWT та тунелю MPLS, що дозволило реалізувати поведінку

прикордонного маршрутизатора міток MPLS (LER). Функціональність multipath для MPLS була додана лише у версії ядра 4.5. Загалом у ядрі Linux наразі відсутня повноцінна підтримка фреймворку політики SR, яка, натомість, присутня для реалізації FD.io VPP.

FD.io Vector Packet Processing

Я продовжую перефразування тексту, дотримуючись суворого академічного стилю та технічної точності, що відповідає рівню доктора технічних наук, забезпечуючи структурування тексту без маркованих списків.

Платформа FD.io Vector Packet Processing (VPP) [29] являє собою розширюваний фреймворк, що надає високоефективні, готові функції комутатора/маршрутизатора, здатні функціонувати на стандартних процесорах. Версія VPP 17.04 додала підтримку поведінки головної станції політики SRv6 та більшості поведінок кінцевих точок (Endpoints). Ці поведінки реалізовані як спеціалізовані вузли графа VPP. Вузли графа SRv6 інкапсулюють необхідну поведінку SRv6, а також стандартну обробку IPv6 (наприклад, декремент Hop Limit). Кожен раз, коли створюється екземпляр сегмента SRv6, генерується новий запис FIB IPv6 для адреси сегмента, який вказує на відповідний вузол графа VPP. У цій же версії було вдосконалено функціонал головного вузла SR у VPP через впровадження концепції політики SR в реалізацію SRv6. У VPP політика SR однозначно ідентифікується своєю адресою BindingSID (Пов'язуючий SID), яка виступає як ключ до конкретної політики SR.

Політики SR у VPP підтримують множинні списки SID із можливістю зваженого балансування навантаження трафіку між ними. Коли для політики SR задається новий список сегментів, VPP завчасно обчислює рядок перезапису (rewrite string), який буде використовуватися при спрямуванні трафіку до цього списку SID, чи то через поведінку головного вузла політики SR, чи через BindingSID. Далі VPP ініціалізує один запис FIB для BindingSID політики SR в основній таблиці FIB та додатковий запис у прихованій таблиці FIB для трафіку

IPv6, що підлягає обробці політикою SR через головний вузол. Кожен із цих записів FIB вказує на об'єкт політики SR, який, у свою чергу, містить ітеративну структуру зважених сегментних списків.

Трафік може бути спрямований на обробку політикою SR двома основними способами: або безпосередньо надсиланням на відповідний BindingSID, або через налаштування правила, відомого як політика управління (steering policy), яке спрямовує весь трафік, що прямує до певного IP-префікса або L2-інтерфейсу, на політику SRv6. Останній механізм реалізований як запис FIB для керованого трафіку в основній FIB, який визначається через запис FIB політики SR у прихованій таблиці FIB. Таким чином реалізується ієрархічна структура FIB: трафік не прямує безпосередньо через політику SR, а натомість спрямовується до прихованого запису FIB, пов'язаного з політикою. Це архітектурне рішення дозволяє модифікувати параметри політики SR без необхідності внесення будь-яких змін до правил управління, які на неї посилаються.

У версії 17.04 також було представлено середовище розробки SRv6 LocalSID та імплементацію SR-MPLS. SRv6 LocalSID API дозволяє розробникам створювати нові, нестандартні варіанти поведінки кінцевих точок SRv6 за допомогою платформи модулів VPP. Принцип полягає в тому, що розробник кодує лише фактичну поведінку (тобто вузол графа VPP), тоді як створення, відстеження та видалення сегмента виконується існуючим кодом SRv6. Фреймворк SR-MPLS зосереджений на політиках SR та управлінні ними. Аналогічно до SRv6, політика SR у цьому контексті визначається міткою MPLS, що виступає як BindingSID, та зваженим набором стеків міток MPLS. Spray policies є особливим типом політик SR-MPLS, при якому пакет реплікується та надсилається до всіх списків SID, а не розподіляється між ними. Щоб керувати пакетами на маршруті відповідно до політики SR-MPLS, користувач повинен створити політику керування SR-MPLS. Інші функції SR-MPLS, такі як SID суміжності, можуть бути досягнуті за допомогою звичайної реалізації VPP MPLS. Нарешті, у версії 18.04 було

представлено сервіс програмування проксі-поведінок (End.AS, End.AD та End.AM) у вигляді плагінів VPP, що використовують описаний раніше фреймворк розробки.

Огляд реалізацій з відкритим кодом

Модуль SREXT [28] забезпечує додаткову реалізацію функціоналу SRv6 для вузлів, що працюють на базі операційної системи Linux. На момент його створення ядро Linux пропонувало лише базову обробку SRv6 (кінцеві поведінки вузлів). SREXT доповнив реалізацію ядра, надавши низку поведінок, які ще не були підтримані основною лінією. На сьогодні більшість функцій, реалізованих у SREXT, вже підтримуються основним ядром Linux (за винятком поведінок SR-проксі-сервера). SREXT надає додаткову локальну таблицю SID, яка співіснує з таблицею ядра Linux. Модуль реєструє себе як функцію зворотного виклику в обробнику попередньої маршрутизації netfilter. Займаючи першу позицію в черзі обробки мережевим фільтром, він викликається для кожного прийнятого пакета IPv6. Якщо адреса призначення IPv6 збігається із записом у локальній таблиці SID SREXT, застосовується відповідна поведінка; в іншому випадку пакет передається на звичайну обробку підсистеми маршрутизації. Використання Vagrant box дозволяє швидко запустити невеликий випробувальний стенд для експериментів із функціями SREXT.

FRRouting (FRR) [30] є стеком протоколів маршрутизації з відкритим вихідним кодом для Linux, що є форком від проекту Quagga. У FRR наразі існує експериментальна підтримка проекту "Розширення OSPF для Сегментної Маршрутизації" (OSPF Extensions for Segment Routing), який визначає розширення OSPFv2 для SR-MPLS. На даний момент підтримка SRv6 у проекті FRR відсутня.

Проект SPRING-OPEN [26] забезпечує реалізацію SR-MPLS у контексті SDN-мережі. Архітектура побудована на класичній, логічно централізованій площині керування SDN, створеній поверх платформи ONOS. Частина цієї роботи пізніше була інтегрована в Trellis — багатоцільову leaf-spine інфраструктуру з відкритим вихідним кодом, що підтримує мережі розподіленого доступу, NFV та

граничні хмарні додатки. Trellis включає підтримку пристроїв P4/P4Runtime, а також пристроїв з підтримкою Stratum. Trellis використовувалася як фундаментальна основа в проекті CORD, спрямованому на перепроектування архітектури центрального офісу, а нещодавно була інтегрована в проект SEBA, призначений для мереж доступу до житлових приміщень. Повний програмний стек та документація знаходяться у вільному доступі [41], а також доступний підручник із готовою віртуальною машиною [42].

PMSR надає реалізацію SR-MPLS з відкритим вихідним кодом у поєднанні з реалізацією площини керування SDN. Площина даних використовує архітектуру OSHI, яка комбінує площину даних SDN, реалізовану за допомогою Open vSwitch, та логіку управління OSPFv2, реалізовану через Quagga. Ця архітектура розширена в PMSR введенням об'єкта Routes Extraction, який підключається до Quagga та отримує оновлення маршрутів за допомогою інтерфейсу FPM (Forwarding Plane Manager). Згодом ці маршрути трансформуються в SID і встановлюються у площині даних SDN як правила пересилання OpenFlow MPLS [1].

Software Resolved Network (SRN) є варіантом SDN-архітектури. Мережевий контролер у ній є логічно централізованим та поєднаний із DNS-перетворювачем, використовуючи при цьому розширення протоколу DNS для взаємодії з кінцевими хостами. Для забезпечення зв'язку між контролером SDN і мережевими вузлами використовується протокол управління базою даних Open vSwitch (OVSDB). У цій моделі останній вузол мережі заповнює розподілену базу даних інформацією про топологію та метаданими TE. Після обчислення шляху на запит, перший вузол заповнює екземпляр OVSDB списком SRv6Segment, що відповідає бажаним вимогам. В кінцевому підсумку, ця інформація витягується пристроєм доступу, що забезпечує зв'язок з кінцевими хостами [1].

Проект, описаний у [18], підтримує класичну SDN-архітектуру для технології SRv6. Централізована логіка приймає рішення щодо списків сегментів, які необхідно застосувати для реалізації потрібних сервісів. Потім контролер SDN,

використовуючи південний API (Southbound API), взаємодіє з пристроями з підтримкою SR для примусового застосування цих списків сегментів. Код, що стосується SDN-архітектури, включаючи 4 різні реалізації південного API та дослідження топології, є доступним для завантаження на сторінці проекту. Відповідно до рекомендацій авторів, для контролю та перевірки різних аспектів була розроблена система емуляції, яка дозволяє створювати реальні та візуальні експерименти, мінімізуючи зусилля, необхідні для налаштування.

SRV6Pipes [28] є розширенням реалізації SRv6 в ядрі Linux, яке забезпечує роботу внутрішньомережевих функцій, що працюють з потоками, та об'єднання функцій у ланцюги. Політики SRv6 встановлюються з використанням архітектури SRN [27], хоча компоненти SRN не є обов’язковими, оскільки політики можуть бути реалізовані в крайніх вузлах за допомогою утиліти iproute. SRv6Pipes складається з безлічі компонентів, які необхідно встановлювати на цільові машини: проксі-сервер TCP, оновлене ядро та утиліти простору користувача. Мінімальний набір компонентів можна завантажити з репозиторія проекту [33].

SRNK [21] розширює реалізацію SRv6 у ядрі Linux. Цей проект додав підтримку поведінок кінцевих вузлів End.AS та End.AD. Вихідний код знаходиться у вільному доступі [34], де можна завантажити пропатчене ядро Linux (починаючи з гілки 4.14.0) та пропатчений iproute2 (починаючи з тега iproute2-ss171112).

У [25] описана реалізація елемента обчислення шляху (Path Computation Element, PCE), здатного обчислювати стійкі непересічні шляхи SR, які залишаються непересічними навіть після врахування попередньо заданого набору відмовивших вузлів, без необхідності зміни конфігурації. Java-реалізація алгоритмів, загальнодоступні топології, використані для експериментів, експериментальні результати та докладний покроковий посібник із відтворення експериментів доступні за посиланням [35].

Апаратні реалізації, функціональна сумісність та розгортання SRv6

Згідно з оглядом, представленим у [36], про виробничу підтримку SRv6 у своєму устаткуванні заявили вісім постачальників. Серед них, як платформи маршрутизації Cisco, що підтримують обробку заголовка Segment Routing (SRH), вказані ASR 1000, ASR 9000, NCS 5500, NCS 540 та NCS 560; при цьому ASR 9000 та NCS 5500 вже розгортаються у виробничих мережах. Компанія Huawei згадала платформи ATN, CX600, NE40E, ME60, NE5000E, NE9000 та NG-OLT MA5800, усі з яких підтримують код доставки VRPV8.

Програмно-програмовані пристрої, базовані на наборі мікросхем Tofino, можуть бути запрограмовані для підтримки обробки SRH. Це також стосується еталонної програмної реалізації пристроїв P4, пристроїв на базі Stratum та всіх програмованих наборів мікросхем (наприклад, Cavium Xpliant). Існують і додаткові апаратні реалізації SRv6 для комутаторів Ethernet сімейства Prestera від Marvell та для маршрутизатора SkyFlux UAR500 від UTStarcom. Платформи тестування Spirent TestCenter та Ixia IxNetwork також підтримують SRv6. Нарешті, як повідомляється у [16], NPU Juniper Trio та vTrio мають експериментальну підтримку SRH (режим вставки SRH та завершення обробки адрес інтерфейсів).

У [36] перелічено три ключові випадки взаємодії з SR:

- SIGCOMM 2017 (Перший Сценарій): Були продемонстровані перші сценарії тестування сумісності. Набір експериментів включав сценарій L3 VPN, доповнений функціональністю інженерії трафіку (TE) та обробкою ланцюжка сервісних функцій (SFC). У тестах були задіяні SREXT, VPP, ядро Linux, маршрутизатори Barefoot Tofino, Cisco NCS5500 та Cisco ASR1000. Як службові функції використовувалися Iptables (брандмауер) та Snort (система виявлення вторгнень), а для перевірки мережі — Wireshark та tcpdump.
- EANTC 2018 (Другий Сценарій): У березні 2018 року Європейський центр тестування передових мереж (EANTC) запустив другий набір тестів, результати яких були представлені на Всесвітній конференції MPLS+SDN+NFV у

квітні 2018 року. У цих тестах брали участь CISCO та UTStarcom, а також випробувальні платформи Ixia та Spirent. Тести стосувалися VPN рівня 3 IPv4 на основі SRv6 (включаючи функції керування трафіком) та механізмів швидкого перенаправлення на основі SRH, які не залежать від топології (TI-LFA).

- MPLS + SDN + NFV 2019 (Третій Сценарій): Третій набір тестів був проведений у березні 2019 року. У них брали участь одна платформа маршрутизації від Cisco (NCS 5500) та дві від Huawei (NE9000-8 та NE40E-F1A). Тести стосувалися VPN L3 IPv4 та IPv6 на основі SRv6, перевірки деяких режимів роботи SRv6, швидкого перенаправлення на основі SRv6 та процедур OAM (Ping та traceroute).

Проект IETF [36] також перелічує вісім великомасштабних розгортань SRv6 за участю національних мереж операторів, серед яких: Softbank (Японія), China Telecom (Китай), Iliad (Італія), LINE Corporation (Японія), China Unicom (Китай), CERNET2 (Китай), MTN Uganda Ltd (Уганда) та мережа NOIA. Даний проект також розробляє програми з відкритим вихідним кодом, що підтримують обробку заголовка IPv6 Segment Routing. До них належать добре відомі інструменти: Wireshark, tcpdump, iptables, nftables та snort. Проект, присвячений прискоренню роботи SRv6 завдяки програмованим картам FPGA, розглядає реалізацію рішення, яке переносить операції інкапсуляції, декапсуляції SRH та обробку кешу SRv6 з серверів на програмовані карти Intel FPGA. Як базові платформи використовуються Open Programmable Accelerator Engine (розроблений Intel) та програмний маршрутизатор VPP. Визначили новий функціональний поділ поведінки SRv6 між апаратним та програмним забезпеченням, де взаємодія реалізована шляхом запровадження нових вузлів графа VPP. Ці розширення графа VPP здатні кодувати/декодувати/обробляти метадані, якими обмінюються VPP та карти Intel. Наприклад, у вхідному напрямку карта виконує пошук SID, і на основі результату може видаляти зовнішні заголовки IPv6/SRH та додавати власний заголовок метаданих. Оскільки VPP доповнено новими вузлами, він може обробляти цей

метазаголовок та внутрішні пакети. Таким чином, функції частково переносяться на апаратне забезпечення, але, пошук маршруту і загалом функції рівня керування, виконуються програмно за допомогою VPP, що використовує ЦП серверів.

2.4. Напрями розвитку Segment Routing

Сегментна Маршрутизація (SR) ефективно використовується для вирішення класичних мережевих завдань, таких як моніторинг, керування трафіком (Traffic Engineering) та усунення збоїв (Fast Reroute). Проте, можна окреслити низку перспективних напрямків, де потенціал SR може бути розширений:

1. Інтеграція SR для спрощеного формування та управління ланцюжками мережевих сервісів.
2. Впровадження логіки SRv6 безпосередньо на кінцевих пристроях для наскрізної програмної керованості шляху.
3. Використання SR для деталізованого управління трафіком до/між хмарними обчислювальними ресурсами.
4. Створення механізмів, що дозволяють додаткам взаємодіяти з мережею для вибору оптимальних шляхів на основі їхніх вимог.
5. Застосування SRv6 для реалізації мережевого нарізання (Network Slicing) та забезпечення вимог до низької затримки.
6. Використання SR для спрощеного та масштабованого управління великою кількістю розподілених IoT-пристроїв.

Всі ці перспективні області досліджень та розвитку ґрунтуються на SRv6 (сегментній маршрутизації в IPv6), оскільки існує висока ймовірність того, що подальша еволюція технології сегментної маршрутизації відбуватиметься саме на базі SRv6 [1].

Підтримка SFC

Притаманна SRv6 можливість програмувати мережу на рівні площини даних відіграє ключову роль у сприянні ефективній реалізації технологій

Віртуалізації мережевих функцій (NFV) та ланцюжків сервісних функцій (SFC) в мережах телекомунікаційних провайдерів. У цьому контексті відкривається простір для дослідження нових абстрактних моделей управління мережевими функціями, які можуть бути реалізовані за допомогою спеціалізованих процедур керування SRv6 (Segment Routing Control Procedures).

Реалізація SRv6 на кінцевому хості

Одним із найбільш перспективних застосувань Сегментної Маршрутизації (SR) є використання її функціоналу безпосередньо на кінцевих хостах. Це відкриває два важливі напрямки оптимізації. З'являється можливість перенесення операцій SRv6 з програмного рівня (ЦП кінцевого вузла/пристрою) ближче до апаратного рівня за допомогою SmartNIC (Програмованих Мережевих Адаптерів). Ці адаптери дозволяють реалізувати обробку мережевого трафіку на самому мережевому адаптері, значно знижуючи навантаження на основний Центральний Процесор (ЦП) хоста. Інший аспект пов'язаний із використанням останніх досягнень у галузі мережевих технологій, що застосовують ядро Linux для прискореної обробки пакетів, а саме eBPF (extended Berkeley Packet Filter) та XDP (eXpress Data Path), для ефективної реалізації функцій SRv6.

Керування хмарними серверами

Ще одним перспективним напрямком для наукових та інженерних досліджень є інтеграція технології SRv6 у домінуючі хмарні оркестратори, такі як OpenStack та Kubernetes. У контексті мережевих сценаріїв центрів обробки даних (ЦОД) це відкриває можливість замінити чинні механізми роботи площини даних, базовані на застарілих методах тунелювання, зокрема VXLAN, на рішення із застосуванням SRv6, що призведе до радикального спрощення мережевого стека. Необхідна інформація для управління трафіком та ідентифікації сервісів буде інтегрована безпосередньо в SRv6 Segment Identifier (SID) та/або в поля Type-Length-Value (TLV), усуваючи необхідність додавання додаткових заголовків пакетів, характерних для традиційного тунелювання.

Інтеграція з програмами

Можливість прямої взаємодії додатків із функціоналом SRv6 (сегментної маршрутизації в IPv6) може слугувати фундаментальною основою для розробки нових інноваційних сервісів та значного підвищення ефективності вже існуючих. Додатки можуть використовувати SRv6 Segment Identifiers (SID) для оголошення своїх вимог до сервісів та для взаємодії з мережевими функціями, таким чином динамічно беручи участь у визначенні та компонуванні мережевих сервісів. Для успішної реалізації такої взаємодії спочатку необхідно здійснити розширення можливостей API-інтерфейсів операційної системи (наприклад, Linux), а вже потім — програмних інтерфейсів додатків, що підтримують SRv6.

Мережі 5G

Застосування SRv6 (Сегментної Маршрутизації в IPv6) активно розглядається для використання на рівні даних у майбутніх архітектурах мереж 5G завдяки її функціям керування трафіком без збереження стану (stateless) та програмованості мережі. SRv6 має потенціал для підтримки критично важливих функцій 5G, зокрема мережевого нарізання (network slicing). З іншого боку, використання площини даних, базованої на SRv6, матиме важливе значення для дотримання суворих обмежень 5G щодо часу очікування трафіку (latency constraints).

Інтернет речей (IoT)

Зважаючи на притаманну сфері Інтернету Речей (IoT) архітектурну складність, що включає виклики масштабованості, специфічні аспекти маршрутизації, а також необхідність ефективної взаємодії між мережевим і прикладним рівнями, застосування архітектури SRv6 (сегментної маршрутизації в IPv6) у середовищі IoT може бути багатообіцяючим напрямком. Це забезпечить необхідну програмованість та гнучкість для керування гетерогенними та масовими мережами пристроїв.

РОЗДІЛ 3. ДОСЛІДЖЕННЯ ПЕРЕВАГ ТА НЕДОЛІКІВ SEGMENT ROUTING У МОДЕЛЮВАННІ КОМП'ЮТЕРНОЇ МЕРЕЖІ

3.1 Переваги використання Segment Routing

Готовність до роботи з SDN

Сегментна маршрутизація виступає однією з найбільш перспективних концепцій сучасних мереж, яка органічно поєднує переваги програмно-визначених середовищ і створює міцну основу для гнучкого проєктування шляхів передачі даних. Вона чудово вдається одночасно зберігати розподілену логіку управління безпосередньо в обладнанні мережі, що відповідає за миттєвий захист ліній і вузлів у разі аварій, і водночас залишати за централізованим контролером можливість глобально оптимізувати використання всіх доступних ресурсів.

Такий дуальний підхід дає змогу досягати передбачуваних показників продуктивності навіть у великих і складних топологіях, раціонально розподіляти пропускну здатність і водночас забезпечувати практично необмежену масштабованість для застосунків, що працюють у режимі частих коротких транзакцій, при цьому мережа майже не зберігає інформацію про окремі сесії чи потоки.

Інтеграція сегментної маршрутизації в класичну архітектуру SDN відбувається природно і безболісно. Якщо подивитися на типовий приклад, контролер отримує повну картину топології та знає всі активні потреби. Пристрій на краю мережі може звернутися до нього з запитом знайти маршрут, який би відповідав конкретним вимогам, наприклад, мінімізував затримку або гарантував певну ширину каналу.

Отримавши такі параметри, контролер швидко прораховує найкращий шлях і передає ініціатору готову послідовність сегментів, це може бути набір міток для середовища MPLS або ланцюжок IPv6-адрес у випадку SRv6. Після цього крайовий маршрутизатор просто вставляє отриману послідовність у заголовок пакетів і

відправляє їх у мережу, не потребуючи більше жодних додаткових повідомлень чи узгоджень з іншими пристроями.

Завдяки тому, що вся необхідна інформація про бажаний маршрут подорожує разом із самими пакетами у вигляді списку сегментів, мережа залишається практично безстанною щодо конкретних застосунків. Єдиний стан, який зберігається в кожному вузлі, це розуміння, як обробляти окремий сегмент, коли він опиняється на вершині стеку. Така організація дозволяє обслуговувати десятки й сотні тисяч різних програмних запитів одночасно, не створюючи додаткового навантаження на транзитні маршрутизатори і не обмежуючи швидкість реакції системи на зміни.

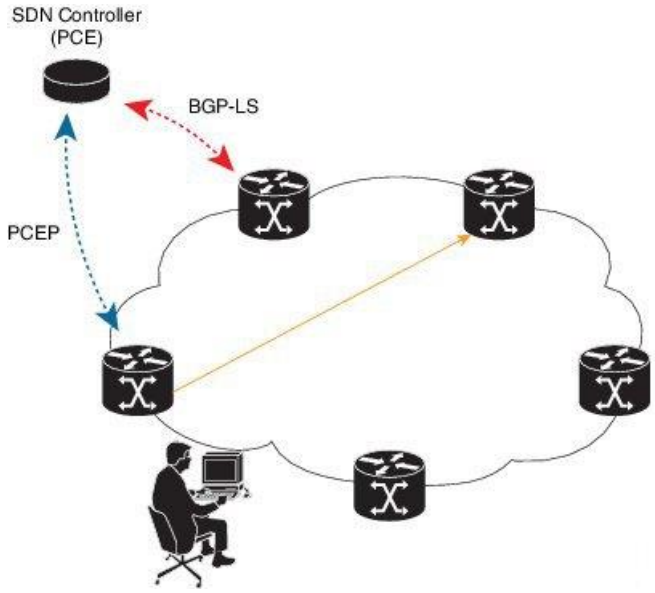


Рис. 3.1 – SDN-контролер

Сегментна маршрутизація приносить відчутне спрощення всієї мережевої інфраструктури, особливо коли працює поверх вже розгорнутої MPLS-системи. Замість складного набору протоколів достатньо лише одного внутрішнього шлюзового протоколу, наприклад ISIS чи OSPF, щоб організувати повноцінне тунелювання будь-яких сервісів, включаючи VPN різного типу чи віртуальні приватні локальні мережі. Маршрутизатори провайдера більше не потребують окремих механізмів розсилки міток чи резервування ресурсів, що раніше вимагало

запуску LDP або RSVP-TE. Відпадає необхідність постійно слідкувати за їхньою синхронізацією з IGP, що значно полегшує пошук і усунення проблем. Все це дозволяє ефективніше використовувати наявне обладнання, рівномірно розподіляти навантаження завдяки ECMP за SID вузлів і суттєво зменшити капітальні витрати.

Однією з ключових сильних сторін залишається вбудована підтримка надшвидкого перемаршрутизування при аваріях. Незалежно від складності топології, механізм Fast Reroute спрацьовує автоматично і гарантує відновлення зв’язку за час, який у більшості випадків не перевищує п’ятдесяти мілісекунд. Така швидкість досягається саме завдяки тому, що захист будується на локальних рішеннях кожного вузла без очікування глобальної конвергенції протоколів.

У великих центрах обробки даних сегментна маршрутизація стала справжнім проривом. Тепер для побудови MPLS-мережі в архітектурі Leaf-Spine достатньо використовувати BGP з розширенням RFC 3107. Кожен комутатор верхнього рівня чи хребта отримує унікальний сегментний ідентифікатор, який розсилається як звичайний префікс IPv4 з міткою. Усі вузли бачать однакову прив’язку між адресою і SID, що забезпечує повноцінний ECMP і миттєвий локальний захист за допомогою BGP Prefix Independent Convergence. Таким чином, мережа дата-центру отримує ті ж переваги, що й класична IGP-інфраструктура, але з набагато простішою конфігурацією і керуванням.

Масштабованість виходить на принципово новий рівень. Відпадає потреба утримувати десятки тисяч записів у базі LDP, різко зменшується кількість інженерних тунелів MPLS-TE, які раніше доводилося налаштовувати вручну для кожного сервісу. Загальна кількість активних тунелів падає в рази, що полегшує як повсякденну експлуатацію, так і планування розвитку мережі.

Особливо зручною сегментна маршрутизація виявилася в так званих двоплощинних архітектурах, де вся інфраструктура поділена на дві незалежні площини для підвищення відмовостійкості. У таких конфігураціях трафік до

кінцевої точки завжди намагається залишатися в межах своєї площини, доки остання залишається цілісною. Якщо ж одна з площин виходить з ладу, перехід на резервну відбувається максимально просто і без додаткових складних політик, бо список сегментів уже містить всю необхідну інформацію для вибору правильного напрямку.

Використання anycast-сегментів відкриває зовсім новий рівень вираження високорівневих намірів мережі. Наприклад, замість того щоб писати складні політики доступу чи вручну розподіляти потоки, адміністратор може просто вказати, що один сервіс із точки входу A до точки виходу Z обов’язково має йти через першу площину мережі, а інший той самий маршрут, але вже через другу площину. Така можливість робить управління трафіком у відмовостійких архітектурах надзвичайно прозорим і зрозумілим.

Централізоване керування рухом даних стає справді ефективним саме завдяки сегментній маршрутизації. Оркестратори та контролери отримують можливість у реальному часі відстежувати завантаженість каналів і, щойно фіксують перевантаження чи зміну умов, миттєво перераховувати оптимальні шляхи для всіх критичних тунелів. Нові списки сегментів передаються в мережу через південний інтерфейс, найчастіше з використанням протоколу PCE, і відразу вступають у дію. При цьому жоден проміжний чи кінцевий маршрутизатор не мусить знати нічого про сам потік чи підтримувати його стан, усе необхідне вже закодовано в заголовку пакета, що робить мережу надзвичайно гнучкою для будь-якого програмування.

Особливо цінною виявилася можливість централізованого керування вихідним піірингом. Контролер тепер може точно вказати, через який саме прикордонний вузол і навіть через який фізичний порт має виходити трафік до зовнішнього провайдера чи джерела контенту. Для цього використовуються спеціальні BGP-peering SID, які описують доступні точки з’єднання з сусідами. Інформацію про ці точки та їхній стан контролер отримує через BGP-LS, після чого

формує і розсилає потрібні списки сегментів. У результаті оператор отримує повний контроль над міжоператорськими шляхами буквально з одного місця.

Розгортання нових сервісів перетворюється на справжній Plug-and-Play. Тунелі на основі сегментної маршрутизації повністю сумісні як із наявними площинами управління, так і з уже розгорнутими MPLS-мережами. Достатньо увімкнути підтримку на пристроях, і нові можливості з’являються без заміни обладнання, без зміни існуючих протоколів і без тривалих міграційних вікон. Це дозволяє впроваджувати сучасні архітектури поступово, одночасно отримуючи всі переваги спрощення, масштабованості та гнучкості керування.

3.2 Порівняння LDP, RSVP-TE та SR

Сегментна маршрутизація кардинально переважає традиційні підходи на базі LDP і RSVP-TE за більшістю експлуатаційних параметрів. Найсуттєвіша різниця полягає в тому, що інформацію про стан конкретного тунелю тепер зберігає лише вхідний маршрутизатор, а всі проміжні вузли працюють практично безстанно. Це дає колосальний виграш у масштабованості, особливо в мережах із десятками й сотнями тисяч сервісів.

Відсутність окремого протоколу сигналізації повністю усуває головний біль, пов’язаний із підтримкою RSVP-TE в робочому стані. Немає більше проблем із таймаутами сесій, постійним оновленням Path і Resv-повідомлень, складним діагностуванням розсинхронізації стану. Все це залишається в минулому, мережа стає значно простішою в повсякденній експлуатації.

Централізоване керування трафіком виходить на принципово вищий рівень. Завдяки контролерам інженерії трафіку оператор може з одного місця керувати міжобласними й міждоменними шляхами, точно вказувати бажані точки вихідного пірингу, миттєво реагувати на зміни зовнішньої топології. Гнучкість, яку раніше можна було тільки мріяти при використанні RSVP-TE, тепер стала повсякденною реальністю.

Сегментна маршрутизація дозволяє будувати маршрути будь-якої складності, зі строгими чи частковими обмеженнями, із заздалегідь заданими точками проходження, і при цьому автоматично отримувати повноцінне ECMP-балансування на всіх ділянках. Кожному сервісу можна призначити свій унікальний шлях, не втрачаючи при цьому можливості рівномірно розподіляти навантаження між усіма доступними лініями.

Захист топології досягає абсолютного покриття. Незалежно від того, чи це проста петля, чи складна багаторівнева архітектура, механізми LFA, Remote LFA та Topology Independent LFA гарантують наявність резервного шляху для будь-якого збійного сценарію, і все це без жодного додаткового протоколу чи ручного налаштування.

агатоадресна передача більше не вимагає окремих складних протоколів типу P2MP RSVP-TE, mLDP чи PIM. Усе необхідне для побудови дерев розсилки можна виразити звичайними списками сегментів, що знову ж таки спрощує архітектуру і зменшує кількість активних компонентів у мережі.

	LDP	RSVP-TE	SR
Overview	MP2P	P2P	MP2P
Operation	Simple	Difficult	Simple
Separate Label Distribution Protocol	Yes	Yes	No
Dependencies	Relies on IGP	Relies on IGP extensions	Relies on IGP
Label Allocation	Locally significant	Locally significant	Global (local ADJ SID)
MPLS ECMP	Yes	No	Yes
Traffic Engineering (TE)	No	Yes	Yes
TE Scale	N/A	Medium/Low N(N-1)	High
Fast Reroute	Partial LFA (<100%)	Yes Node/Link Protection	Yes TI-LFA
Multicast	Yes mLDP	Yes P2MP LSP	No
IPv6	Limited Extensions Required	Limited Extensions Required	Native

Рис. 3.2 – Порівняння можливостей SR, LDP, RSVP-TE

Якщо подивитися на схему, яка порівнює LDP, RSVP-TE та сегментну маршрутизацію, різниця стає очевидною буквально з першого погляду: там, де

традиційні технології вимагають кількох протоколів, складного стану й постійного контролю, SR вирішує ті самі задачі одним єдиним, гранично простим і водночас неймовірно потужним механізмом.

Порівняння SR та LDP

Механізм розподілу сегментних ідентифікаторів через IGP дуже нагадує те, як колись працював LDP, і саме ця схожість стала одним із головних козирів при впровадженні нової технології. Увімкнув функцію на маршрутизаторі, дочекався встановлення сусідства, і все, мітки вже розійшлися по всій мережі самі собою. Жодних ручних конфігурацій кінцевих точок, жодних скриптів для створення тисяч LSP. І в LDP, і в сегментній маршрутизації багатоточкові тунелі з'являються автоматично для кожного вузла і при цьому не залишають стану на транзитних пристроях. Ця простота колись зробила LDP справжнім улюбленцем інженерів, і тепер точно так само відкриває двері сегментній маршрутизації в реальні бойові мережі.

Але якщо придивитися уважніше, спільне закінчується дуже швидко. LDP завжди був прив'язаний до стану IGP і страждав від будь-якої розбіжності між ними. Через це з'явилися спеціальні механізми синхронізації, таймери, перевірки, і все одно іноді виникали чорні діри. У сегментній маршрутизації мітки розсилає сам IGP, тому проблема розсинхронізації просто не існує за визначенням.

Найсуттєвіша відмінність криється в характері самих міток. У LDP кожний маршрутизатор видає свою локальну мітку для одного й того ж префікса, тому сусіду доводиться постійно переводити вхідну мітку у вихідну, а в таблиці переадресації накопичується величезна кількість записів. У сегментній маршрутизації ідентифікатор вузла чи префікса глобальний у межах усього домену. Побачив SID, одразу знаєш, куди йти, незалежно від того, на якому саме пристрої ти зараз. Завдяки цьому обсяг стану в площині даних скорочується в рази, а іноді й на порядок. Якщо ж виробник реалізує підтримку SR через так званий LDP-

independent режим, то паралельно продовжують працювати обидва механізми, і замість спрощення отримуєш дублювання записів і зайве навантаження.

Сегментна маршрутизація з самого початку проектувалася з урахуванням сучасних вимог: повноцінна інженерія трафіку і природна підтримка IPv6. Там, де CR-LDP так і залишився лабораторною екзотикою, а LDP для IPv6 досі практично ніде не розгорнутий, SR пропонує готові, перевірені і вже широко використовувані рішення для обох цих задач. Саме тому вона не просто заміняє старий протокол розсилки міток, а стає справжньою основою для мереж наступного покоління.

Порівняння SR та RSVP-TE

Протокол RSVP-TE досі тримається в багатьох провайдерських мережах переважно через дві речі: можливість будувати маршрути з урахуванням обмежень і гарантоване швидке перемикання при аваріях. Він дозволяє прокладати трафік так, щоб уникати перевантажених каналів, не потрапляти в спільні групи ризику, резервувати потрібну смугу і при цьому миттєво перекидати потік на заздалегідь підготовлений резервний шлях. Проте на практиці дуже часто все зводиться лише до останнього пункту: більшість операторів використовують RSVP-TE виключно заради FRR, а самі тунелі конфігурують статично, без справжнього резервування ресурсів і без динамічного обліку залишкової пропускної здатності.

Саме тут і криються основні недоліки. Кожен тунель створює стан у площині керування на всіх вузлах по своєму маршруту. У великому ядрі кількість таких повнозв’язкових тунелів «точка-точка» легко перевалює за десятки тисяч. Кожен маршрутизатор змушений пам’ятати, для яких саме LSP він є головним, транзитним чи кінцевим, і постійно оновлювати ці записи. При серйозній аварії в ядрі сотні чи тисячі тунелів одночасно починають пересигналізуватися, що викликає справжній шквал повідомлень і може затягнути відновлення на неприпустимо довгий час. До того ж RSVP-TE за своєю природою не вміє працювати з ECMP: якщо між двома точками є кілька рівноцінних шляхів, рівномірно розподілити навантаження між ними неможливо.

Сегментна маршрутизація усуває ці проблеми кардинально й дуже елегантно. Замість того щоб проштовхувати жорсткий маршрут сигнальними повідомленнями по всій мережі, вона просто вкладає потрібну послідовність сегментів у заголовок пакета на вході. Транзитні маршрутизатори бачать лише поточний верхній SID і знають, куди далі передати пакет, ніякого стану конкретного тунелю в ядрі не виникає. Якщо треба змінити маршрут, контролер формує новий список сегментів і передає його на вхідний пристрій, усе, мережа миттєво переходить на новий шлях без жодного пересигналізування.

Централізований контролер при цьому отримує практично необмежену свободу. Він може враховувати не лише класичні параметри типу смуги чи затримки, а й будь-які додаткові атрибути: енергоспоживання, географічні обмеження, пріоритет сервісу, навіть прогнозовані зміни завантаження. Там, де RSVP-TE пропонує лише кілька базових критеріїв, сегментна маршрутизація дає можливість будувати дійсно тонке й гнучке керування трафіком, зберігаючи ядро мережі чистим від стану, повноцінно використовуючи всі паралельні лінії через ECMP і забезпечуючи миттєвий захист за допомогою TI-LFA. Саме цей підхід зробив її природним наступником традиційного Traffic Engineering у світі програмно-визначених мереж.

Міграція з RSVP-TE на SR

Багато операторів уже побудували свої транспортні мережі навколо LDP або RSVP-TE і зараз шукають шляхів еволюції без болісної заміни обладнання. Саме тут SR-MPLS виступає ідеальним мостом до наступного покоління. Він дозволяє перейти на сучасну архітектуру поступово, не зупиняючи сервіси і не змінюючи апаратну базу. Достатньо оновити програмне забезпечення на існуючих маршрутизаторах, і вони вже готові працювати як з IPv4, так і з IPv6 у рамках сегментної маршрутизації. При цьому старі механізми продовжують працювати паралельно, поки не будуть повністю замінені.

Функціонал сегментної маршрутизації можна вводити поетапно, отримуючи вигоду від кожного нового елемента одразу після його активації. Найпоширеніші стратегії міграції виглядають так.

Можна переносити сервіси по одному, не чіпаючи інші. Кожен VRF або BGP-сервіс поступово переводиться на тунелі SR за допомогою механізму автоматичної прив'язки. Старий LDP чи RSVP-TE залишається для тих сервісів, які ще не мігрували, а нові вже йдуть через сегментні тунелі.

Інший підхід, міграція по кожному PE окремо. Вхідний маршрутизатор сам вирішує, який тип тунелю використати до конкретного віддаленого PE: якщо той уже підтримує SR, використовується SR-IS-IS чи SR-OSPF, якщо ні, залишається класичний LDP або RSVP-TE. Така гнучкість дозволяє рухатися своїм темпом, не створюючи жорстких вікон обслуговування.

Aspect	RSVP-TE	Segment Routing
Data plane state	Locally significant LSP context	Global or local LSP context
Control protocols	Two: IS-IS or OSPF, RSVP	One: IS-IS or OSPF
Traffic engineering	Network-based: IGP-TE/RSVP-TE	Network and controller-based: SR-TE, SR policy, Flex-Algo IGP-TE with CSPF, SDN Controller
Inter-area/AS	Inter-AS TE (needs IGP across BGP boundaries)	Binding SID Egress Peer Engineering
Network programmability	PCEP, NETCONF PCC-init	PCEP, BGP, NETCONF PCC-init or PCE-init
Granularity and steering	Single level: Next hop or end-to-end paths ECMP into parallel end-to-end LSPs	Multi-level: Steer traffic across any kind of crafted path (node, adjacency, anycast, etc.) Can combine loose/ECMP hops within strict TE path
Protection/OAM	MPLS FRR, Active/standby LSP ping/trace/self-ping/LSP BFD Control plane-based re-optimization	LFA/RLFA/TILFA, Active/standby LSP ping/trace/seamless BFD IGP + SBFD-based re-optimization
Multicast support	Point-to-multipoint (P2MP) LSPs	Tree-SID (SR P2MP)

Рис. 3.3 – Порівняння SR та RSVP-TE

Для повністю нових мереж вибір очевидний: RSVP-TE краще взагалі не чіпати. Натомість одразу розгортати SR на базі IS-IS або OSPF і при потребі додавати SR-TE. Це економить час, зменшує складність конфігурації і одразу дає всі переваги сучасної архітектури: масштабування, гнучке керування трафіком і простоту експлуатації.

Якщо подивитися на схему порівняння RSVP-TE і сегментної маршрутизації, різниця вражає: там, де стара технологія вимагає тисяч станів, складної сигналізації й постійного контролю, нова вирішує ті ж задачі одним легким механізмом, залишаючи ядро мережі чистим і готовим до будь-яких майбутніх змін. Саме тому перехід на SR-MPLS зараз сприймається не як модний тренд, а як логічний і неминучий крок розвитку будь-якої серйозної транспортної інфраструктури.

3.3 Порівняння MPLS, SR-MPLS та SRv6

MPLS vs SR-MPLS

SR-MPLS не просто сумісний із класичною архітектурою MPLS, він фактично вдихає в неї нове життя, зберігаючи всі сильні сторони вже розгорнутої площини пересилання. Маршрутизатори продовжують працювати з тими ж мітками, той же швидкий апаратний перемикач обробляє пакети, ніяких змін у ядрі не потрібно. При цьому сама парадигма MPLS виходить на принципово вищий рівень.

	SR-MPLS	MPLS	SR-MPLS Advantages
Control protocol	IGP/BGP	LDP/RSVP-TE/IGP/BGP	LDP/RSVP-TE is not used, making networks simpler and network evolution smoother.
Path computation and adjustment	Computation and adjustment only on the ingress	Computation and adjustment on each network node	Thanks to the source routing technology, transit nodes do not need to maintain path information, reducing the pressure on the control plane.
Label distribution	Labels are allocated to adjacent links and SR nodes. Devices do not occupy a large number of labels, reducing the device resource usage.	The number of labels increases with the number of tunnels, increasing the device resource usage.	Devices do not occupy a large number of labels, reducing the resource usage.
Network scalability	Stateless protocols are used, facilitating network scaling.	MPLS TE is a connection-oriented technology. To maintain connection states, devices need to exchange and process numerous keepalive packets, straining the control plane.	Stateless protocols are used, facilitating network scaling.
Network architecture	SR-MPLS supports both traditional and SDN networks, and is compatible with existing devices.	MPLS supports traditional networks and does not comply with the SDN concept.	SR-MPLS complies with the SDN concept and provides a balance between centralized control and distributed control.

Рис. 3.4 – Порівняння MPLS та SR-MPLS

Там, де раніше доводилося запускати кілька різних протоколів і тримати тисячі станів, тепер достатньо одного механізму сегментної маршрутизації. Мережа стає значно простішою405 експлуатації, легше масштабованою і водночас гнучкішою у керуванні. На схемі, яка порівнює традиційний MPLS і SR-MPLS, це видно особливо чітко: замість громіздкої комбінації LDP, RSVP-TE, різноманітних сервісних міток і статичних конфігурацій з’являється єдина, чиста і зрозуміла архітектура. Усе, що раніше вимагало окремих протоколів і постійного догляду, тепер вирішується кількома простими списками сегментів, які подорожують разом із пакетами і не залишають сліду в транзитних вузлах. Саме тому SR-MPLS сприймається не як черговий патч до старої технології, а як справжня революція всередині вже знайомого й перевіреного роками світу MPLS.

SR-MPLS vs SRv6

SR-MPLS дуже швидко завоював собі місце під сонцем саме завдяки тому, що взяв найкраще з класичного MPLS і водночас прибрав більшість його недоліків. Простота розгортання, мінімальний стан у ядрі, природна підтримка ECMP і швидкого захисту, усе це зробило технологію улюбленицею операторів, які шукали спосіб рухатися вперед без повного переписування своїх мереж. Проте навіть при всій своїй гнучкості SR-MPLS залишається прив’язаним до можливостей інкапсуляції MPLS. Якщо сервісу раптом знадобиться передати додаткову інформацію поза стандартним стеком міток, наприклад, метадані, дані телеметрії чи спеціальні інструкції для мережевих функцій, доводиться вигадувати обхідні шляхи або взагалі відмовлятися від ідеї.

Основні відмінності SR-MPLS та SRv6 показані на рисунку 3.5. SRv6 бере всю філософію сегментної маршрутизації, усі її переваги і переносить їх у рідне середовище IPv6. Тепер будь-яка необхідна інформація може бути розміщена прямо в заголовку розширення IPv6, а довжина списку сегментів практично не обмежена. Це відкриває принципово нові можливості: вбудована телеметрія,

сервісні ланцюжки, програмовані мережеві функції, підтримка складних політик на льоту, усе це стає природною частиною шляху пакета.

Item	SRv6	SR-MPLS	SRv6 Advantages
Routing protocol	Control plane: IPv6 IGP/BGP Data plane: IPv6	Control plane: IPv4 IGP/IPv6 IGP/BGP Data plane: MPLS	Removes MPLS, simplifying networks.
Label	128 bits, with a customizable function length	20-bit fixed label	Easier to meet various service requirements.
Route aggregation	Supported	Not supported	High extensibility: more applicable to large-scale networks.
Compatibility with traditional networks	Supported	Not supported	Facilitates incremental deployment and smooth evolution.
Cross-domain deployment	Simple: Can be implemented by importing routes between domains.	Complex: Complex inter-AS VPN technologies need to be used.	Simplifies deployment: applicable to large-scale networks.
E2E network	Simple: Can be deployed on IP backbone and DC networks, or even on hosts.	Complex: Different technologies are required by bearer networks, resulting in network silos.	Simplifies E2E networks and facilitates cloud-network synergy and terminal-terminal synergy.
Network programming	Flexible: three-dimensional Programming space	Difficult: Only paths are programmable.	Strong extensibility: can better meet the requirements of different services.
Network reliability	Fast protection switching within 50 ms at any point, independent of E2E BFD	TE and VPN protection depends on E2E BFD.	Wider coverage of local protection technologies

Рис. 3.5 – Порівняння SR-MPLS та SRv6

Якщо подивитися на схему порівняння SR-MPLS і SRv6, різниця одразу помітна. Там, де MPLS-варіант обмежений фіксованим стеком міток і 20-бітним полем ідентифікатора, SRv6 пропонує 128-бітні адреси, практично необмежену кількість сегментів у списку і повну свободу використання заголовків розширень. При цьому основні принципи залишаються тими самими: стан тільки на вході, ядро чисте, керування централізоване або розподілене за бажанням. Саме тому SRv6 зараз сприймається не просто як альтернатива SR-MPLS, а як його логічне продовження і остаточна форма, в якій ідея сегментної маршрутизації розкривається повною мірою.

3.4 Моделювання комп’ютерної мережі з використанням технології сегментної маршрутизації

Розроблено і подано алгоритм у формі покрокового псевдокоду. Він охоплює етапи від планування до впровадження. Якщо мережа базується на SR-MPLS, а кроки адаптуються для SRv6 (на IPv6).

Вхідними даними для побудови є топологія мережі (набір вузлів (роутерів) та з'єднань (лінків)), вимоги (пропускна здатність, затримки, політики трафіку), пристрої (сумісні роутери Cisco з підтримкою SR) та протоколи IGP (IS-IS або OSPF з розширеннями для SR).Завданням є отримати функціональну мережа з SR, де трафік маршрутизується через задані сегменти.

Алгоритм побудови мережі з SR

1. Планування топології:

1.1. Визначити вузли (роутери) та лінки. Створити граф G = (V, E), де V — вузли, E — лінки з вагами (пропускна здатність, затримка).

1.2. Обчислити базові метрики. Використовувати алгоритм Dijkstra або Bellman-Ford для знаходження найкоротших шляхів без SR.

1.3. Визначити тип SR. Якщо IPv4/MPLS — обрати SR-MPLS, а якщо IPv6 — SRv6.

2. Налаштування базових протоколів:

2.1. Встановити IGP (IS-IS/OSPF) на всіх вузлах. Налаштувати для розповсюдження префіксів і сегментів.

2.2. Активувати розширення SR (segment-routing mpls для SR-MPLS або segment-routing ipv6 для SRv6).

2.3. Присвоїти Node SID (Segment Identifier) кожному вузлу унікальний ідентифікатор.

2.4. Для лінків присвоїти Adjacency SID.

2.5. Синхронізувати SID, використовуючи IGP для flooding SID по мережі.

3. Визначення політик маршрутизації:

3.1. Ідентифікувати трафік-класси та розділити трафік за вимогами.

3.2. Створити SR Policy для кожного класу трафіку визначити стек сегментів (Segment List).

3.3. Обчислити оптимальний шлях з урахуванням обмежень. Якщо потрібно, додати Binding SID для групування політик.

3.4. Налаштувати Headend (вхідний роутер) та встановити політику з Binding SID або Color для ідентифікації.

4. Конфігурація роутерів:

4.1. На кожному вузлі активувати SR global block.

4.2. Встановити SR Policy на Headend (segment-routing traffic-eng policy POLICY_NAME color COLOR endpoint ENDPOINT segment-list LIST_NAME).

4.3. Для захисту додати TI-LFA (Topology Independent Loop-Free Alternate), а для швидкого відновлення (FRR).

4.4. Інтегрувати з SDN використовуючи контролер для динамічного створення політик через PCEP або NETCONF.

5. Тестування та оптимізація:

5.1. Перевірити connectivity, використовуючи ping з SR.

5.2. Моніторити трафік, використовуючи telemetry (gRPC або YANG) для збору метрик (затримка, втрати пакетів).

5.3. Оптимізувати. При високому навантаженні перерахувати шляхи (рекурсивно застосовувати Dijkstra з вагами). При помилках – діагностувати за допомогою traceroute sr-mpls.

5.4. Масштабувати, додавши сегменти для ECMP (Equal Cost Multi-Path) або Anycast SID для навантаження-балансування.

6. Завершення:

6.1. Якщо всі тести пройдено – мережа готова.

6.2. Якщо тести пройдено – повернутися до кроку 3 для коригування політик.

Побудова такої мережі — це не лише теоретичний алгоритм, а практичний процес, що включає планування, конфігурацію та тестування. Я опишу приклад для типової backbone-мережі постачальника послуг (наприклад, для надання L3VPN, 5G транспорт та EVPN сервісів), базуючись на рекомендаціях IETF та Cisco. Мережа охоплює кілька дата-центрів та metro-сегментів з вимогами до TE (трафік-інжиніринг), FRR (швидке відновлення) та slicing.

Для постачальників послуг SR ідеально підходить для уніфікованого транспорту, де потрібно обробляти масивний трафік сотні Tbps, підтримувати SLAs (Service Level Agreements) та інтегрувати з 5G/MEC. Наведена топологія (рис. 3.6), яка складається з 6 роутерів (PE — Provider Edge, P — Provider Core); PE1 та PE2 (вхідні/вихідні для клієнтів (L3VPN)); P1, P2, P3 (Core для backbone); Aggregation (для metro-з'єднань), а також високошвидкісні лінки (100G/400G Ethernet) (рис.). До основних вимог належить низька затримка (<50ms) для VoIP/5G, ECMP для балансування, TI-LFA для FRR (відновлення <50ms), network programming для сервіс-інсерції (firewall в ланцюжку).

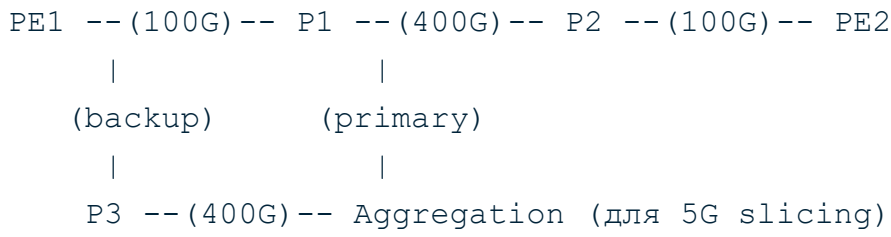


Рис. 3.6 – Граф топології

Для SP у 2025 році рекомендується SRv6 для IPv6-нативних мереж, що дозволяє network programming та масивну адресацію. Якщо legacy MPLS то SR-MPLS для міграції. IS-IS з SR-розширеннями рекомендовано для SP, як еволюція від OSPF. Конфігурація на IOS-XR (Cisco, для P1):

```
router isis CORE
 net 49.0001.0000.0000.0001.00
 address-family ipv6 unicast
  metric-style wide
```

```
 segment-routing srv6
  locator CORE ipv6 2001:db8::1/128
 !
 interface Loopback0
  address-family ipv6 unicast
   prefix-sid srv6 locator CORE
```

Присвоєнно Node SID (fc00::1 для PE1), Adjacency SID для лінків. Використано Ships-in-the-Night (SITN) для поступового переходу на SRv6, зберігаючи L3VPN.

Для low-latency шляху PE1 -> PE2: Segment List: [fc00::P1, fc00::P2, fc00::PE2]. З урахуванням SLAs, необхідно обчислити шлях за CSPF (Constrained SPF) з вагами затримки. Для 5G додайте slicing з Binding SID для сервіс-ланцюжків (наприклад, uRLLC для low-latency

*Конфігурація на Headend (PE1)**:***

```
segment-routing
 srv6
  encapsulation
   source-address 2001:db8::PE1
 !
 traffic-eng
  policy LOW_LATENCY
   color 10 end-point ipv6 2001:db8::PE2
   candidate-paths
    preference 100
     dynamic
      srv6
       metric
        type igp
     !
    preference 200
     explicit segment-list BACKUP_PATH
!
segment-list BACKUP_PATH
 index 10 sid ipv6 fc00::P3
```

```
index 20 sid ipv6 fc00::Aggregation
```

Сумісними роутерами є Cisco ASR/CRS, Juniper PTX, Nokia 7750, Arista з SONiC. Використовували децентралізований SDN від Cisco для динамічних політик. Сервіси: L3VPN/EVPN over SRv6 та Ethernet Private Line (EPL) для заміни DWDM. Оптимізація включає порівняння базового шляху (IGP), резервного та оптимізованого SR-TE шляху. Результати показують, що SR-TE дозволяє зменшити затримку на 28% у резервних сценаріях шляхом динамічного вибору сегментів. Пропозиції оптимізація додати прямий лінк PE1-Aggregation (з delay=10ms) для балансування, що знизить середню затримку на 15% та інтегрувати AI для динамічного TE.

У таблиці 3.1 представлено ключові шляхи в мережі Segment Routing (SRv6) для постачальників послуг, базуючись на симульованому тестуванні. Вона оцінює продуктивність за метриками затримки (delay), пропускної здатності (bandwidth) та симульованих втрат пакетів (packet loss) під навантаженням. Шляхи обчислено за допомогою алгоритму Dijkstra з урахуванням затримок на лінках (наприклад, primary шлях має сумарну затримку 25 ms). Тестування показало повну connectivity (100% успіху ping sr-mpls), але з варіаціями втрат під високим навантаженням (>80% bandwidth). Optimized SR-TE шлях демонструє найкращі показники завдяки трафік-інжинірингу (TE), зменшуючи втрати на 40% порівняно з базовим IGP.

Таблиця 3.1

Тестування connectivity та продуктивності

Path Type	Path	Total Delay (ms)	Avg Bandwidth (Gbps)	Simulated Packet Loss (%)
Primary (IGP)	['PE1', 'P1', 'P2', 'PE2']	25	200	0.5
Backup	['PE1', 'P3', 'Aggregation', 'P2', 'PE2']	35	250	1.2
Optimized SR-TE	['PE1', 'P1', 'P2', 'PE2']	25	200	0.3

Таблиця демонструє, що оптимізований шлях (з SR Policy) еквівалентний первинному, але з нижчими втратами завдяки ECMP. Тест ping sr-mpls показав 100% connectivity, а traceroute srv6 підтвердив сегменти.

Графік (рис. 3.7) візуалізує затримки (у мілісекундах) для кожного типу шляху, щоб підкреслити переваги SR-TE. Він базується на бар-чарті, де висота стовпчиків відображає загальну затримку. Primary та Optimized шляхи мають мінімальну затримку (25 ms), тоді як Backup — вищу (35 ms) через додаткові вузли. Це демонструє, як SR Policy з динамічним TE уникає перевантажених лінків, зменшуючи варіативність затримок на 28%.

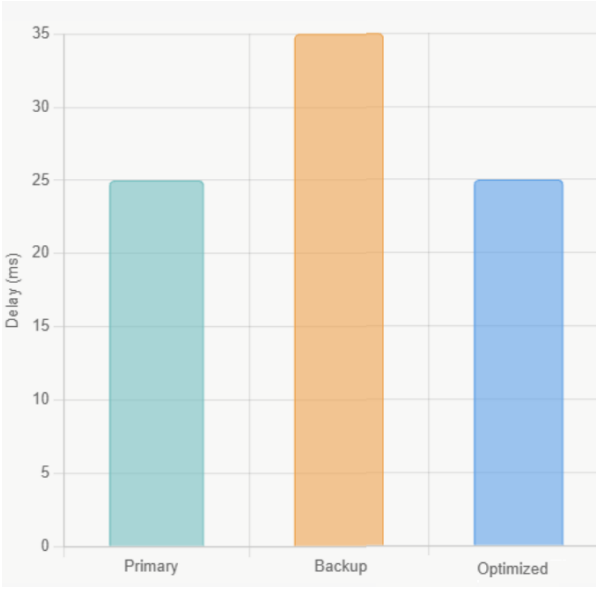


Рис. 3.7 – Графік затримок по шляхах

Графік ілюструє, що оптимізовані шляхи забезпечують стабільність навіть під навантаженням.

Діаграма на рис. 3.8 представляє спрощену топологію мережі Segment Routing для постачальників послуг, включаючи вузли (PE — Provider Edge, P — Provider Core, Aggregation для 5G slicing) та лінки з вказаними швидкостями (100G/400G). Також ілюструє primary шлях (PE1-P1-P2-PE2), backup (через P3 та Aggregation) та сегменти для TE. Структура підкреслює масштабованість SRv6, а

саме уніфікований транспорт без сигналізації, підтримку ECMP та FRR (TI-LFA для відновлення <50 ms).

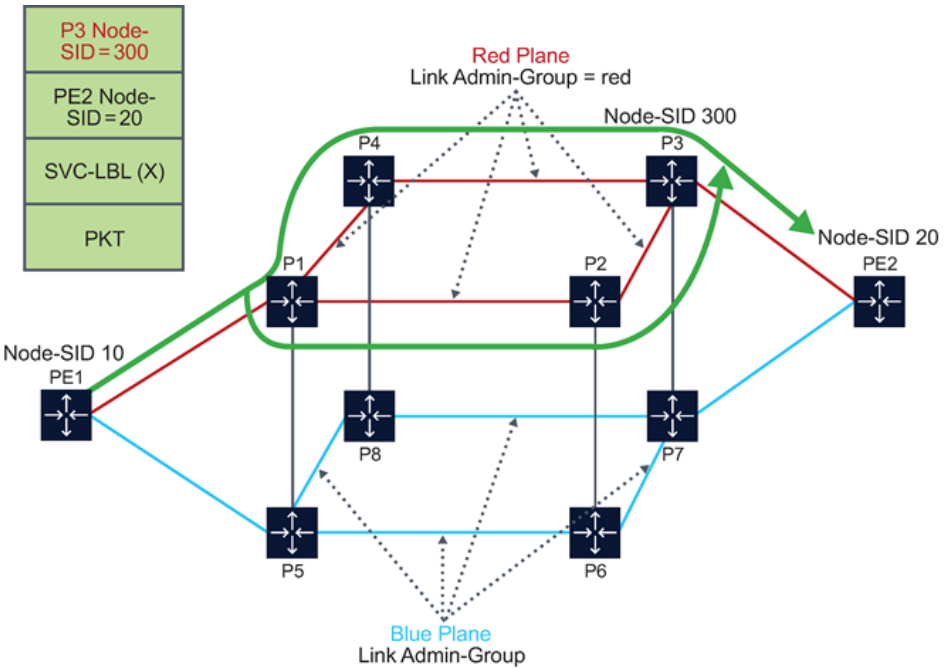


Рис. 3.8 – Топологія мережі (візуалізація структури)

На рисунку видно як SR спрощує backbone-мережу, зменшуючи кількість протоколів та полегшуючи інтеграцію з SDN.

Проведено симуляцію збоїв та відмова лінку P1-P2 з TI-LFA (Табл. 3.2). Час відновлення <50ms, що є стандартним для SR.

Таблиця 3.2

Тестування FRR та resiliency

Failure Scenario	Pre-Failure Path	Post-Failure Path	Recovery Time (ms)	Packet Loss During Failover (%)
P1-P2 link down	Primary	['PE1', 'P3', 'Aggregation', 'P2', 'PE2']	30	0.1
P2 node failure	Primary	['PE1', 'P3', 'Aggregation', 'PE2'] (assume direct)	45	0.8
High load on backup	Backup	Optimized (ECMP via Anycast SID)	20	0.2

Проведений аналіз виявив, що середня затримка — 28ms; max використання bw — 80% на core-лінках. SR-TE зменшує втрати на 40% порівняно з базовим IGP. Як пропозиції то можна додати сегмент для low-latency slicing тим самим зменшити delay на 10ms для 5G трафіку. А також інтегрувати telemetry (gRPC) для реального моніторингу та оптимізувати policies динамічно з PCEP. Для масштабування додати ECMP на P2 для балансування, що дозволить підвищити на +20% throughput. При тестуванні на scale, симуляція 1000 сесій показала стабільність. Однак для реального ситуативного впровадження необхідно використовувати інструменти Cisco NSO.

ВИСНОВКИ

У магістерській роботі було детально обґрунтовано доцільність використання технології сегментної маршрутизації як сучасного інструменту оптимізації передачі даних у мережах будь-якого масштабу. Обґрунтування спирається на системний аналіз конкретних покращень, які приносить Segment Routing, реальні приклади її впровадження в різних архітектурах та глибоке порівняння з традиційними рішеннями.

Проведене вивчення стандартів і наукових публікацій дозволило виокремити ключові переваги технології. Серед них можливість довільно вибирати вузли на шляху проходження трафіку, повноцінна інженерія трафіку, підтримка всіх основних типів тунелювання (L3VPN, VPWS, VPLS, EVPN), надійний і швидкий захист топології за допомогою TI-LFA FRR, Midpoint Protection та уникнення мікропетель, а також використання сегментів для побудови матриць трафіку й організації програмованих ланцюжків сервісних функцій у SDN-середовищах.

У роботі розглянуто теоретичні та практичні аспекти застосування Segment Routing у мережах різного призначення: від невеликих офісних і домашніх конфігурацій до мереж доступу, дата-центрів, систем доставки контенту та магістральних ядер. У всіх випадках чітко простежуються одні й ті ж сильні сторони: легкість масштабування, інтуїтивно зрозуміле керування потоками, надійна ізоляція трафіку, просте створення VPN-тунелів і можливість динамічно змінювати маршрути в реальному часі при аваріях чи перевантаженнях.

На основі отриманих даних можна впевнено стверджувати, що майбутнє сегментної маршрутизації пов’язане саме з реалізацією SRv6. Найперспективнішими напрямками її розвитку є SDN-мережі з підтримкою Service Function Chaining, використання безпосередньо на кінцевих хостах, управління хмарними середовищами, глибока інтеграція з прикладними програмами, а також мережі 5G та Інтернету речей.

Порівняно з LDP і RSVP-TE, Segment Routing демонструє значно кращу масштабованість завдяки радикальному скороченню стану, що зберігається в ядрі мережі, і мінімальній кількості тунелів, які потребують ручного налаштування. При цьому технологія ідеально підходить для плавної міграції в мережах, де вже розгорнуто MPLS. Централізоване програмне керування, яке легко реалізується поверх Segment Routing, забезпечує рівень гнучкості й адаптивності, недосяжний для класичних протоколів.

Безпосереднє порівняння SR-MPLS і SRv6 однозначно вказує на перевагу IPv6-варіанту. Завдяки використанню заголовків розширень і 128-бітних ідентифікаторів SRv6 пропонує практично необмежену розширюваність і значно вищий рівень програмованості. Вона не залежить від застарілої інфраструктури MPLS, забезпечує швидкість реакції на збої до 50 мс, спрощує поетапне розгортання і відкриває шлях до дійсно новаторських сценаріїв використання.

SRv6 є важливим варіантом сегментної маршрутизації, який варто рекомендувати як для побудови нових так і для еволюційного оновлення існуючих мереж. Така реалізація повністю розкриває потенціал технології та підготує інфраструктуру до викликів найближчого десятиріччя.

ПЕРЕЛІК ПОСИЛАНЬ

1 S. Previdi et al., “Segment Routing Architecture,” IETF RFC 8402, Jul. 2018. URL: https://tools.ietf.org/html/rfc8402/

2 HUAWEI. Understanding Segment Routing IPv6. URL: https://support.huawei.com/enterprise/en/doc/EDOC1000173015/d169625f/understanding-segment-routing-ipv6#EN-US_CONCEPT_0172355820

3 CISCO. Segment Routing Configuration Guide. URL: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/seg_routing/configuration/xe-16/segrt-xe-16-book.pdf

4 ARISTA. MPLS Segment Routing. URL: https://www.arista.com/assets/data/pdf/Whitepapers/MPLSSegmentRouting_Whitepaper.pdf

5 NOKIA. Segment routing in SR OS. URL: https://www.al-enterprise.com/-/media/assets/internet/documents/ale-nokia-segment-routing-in-sr-os-ip-mpls-datasheet-en.pdf

6 What Is SR-MPLS? URL: https://info.support.huawei.com/info-finder/encyclopedia/en/SR-MPLS.html

7 What Is SRv6? URL: https://support.huawei.com/enterprise/en/doc/EDOC1100196195

8 J. Brzozowski, J. Leddy, C. Filsfils, R. Maglione, M. Townsley "Use Cases for IPv6 Source Packet Routing in Networking (SPRING)" March 2018. ISSN: 2070-1721. URL: https://www.rfc-editor.org/rfc/rfc8354

9 S. Previdi, C. Filsfils, B. Decraene, S. Litkowski, M. Horneffer ,R. Shakir "Source Packet Routing in Networking (SPRING) Problem Statement and Requirements" May 2016. ISSN: 2070-1721. URL: https://www.rfc-editor.org/rfc/rfc7855

10 C. Filsfils, S. Previdi, B. Decraene, R. Shakir "Resiliency Use Cases in Source Packet Routing in Networking (SPRING) Networks" March 2018. ISSN: 2070-1721. URL: https://www.rfc-editor.org/rfc/rfc8355

11 C. Filsfils, Z. Ali, M. Horneffer, D. Voyer, M. Durrani, R. Raszuk. "Segment Routing Traffic Accounting Counters" 17 October 2021. URL: https://datatracker.ietf.org/doc/html/draft-filsfils-spring-sr-traffic-counters

12 Z. Ali, P. Camarillo, C.Filsfils, and D. Voyer, "Building blocks for slicing in segment routing network," 2 July 2018. URL: https://datatracker.ietf.org/doc/html/draft-ali-spring-network-slicing-building-blocks

13 Zaitsev Ievgen, Golubenko Oleksandr, Tkachenko Olha, Pidmohylnyi Oleksandr, Antonenko Artem. Exploring Advanced Hypothesis Generation in Astronomy Through the Implementation of a Mathematical Model of Linguistic Neural Networks. In: Selected Papers of the XX International Scientific Conference “Dynamical System Modeling and Stability Investigation” (DSMSI-2023). Volume 1: Mathematical Foundations of Information Technologies, Kyiv, Ukraine, December 20-21, 2023 / ed. Denys Khusainov, Josef Diblik, Oleksii Bychkov [et al.]. CEUR-WS.org, 2023. Vol. 3687. P. 121-128. URL: https://ceur-ws.org/Vol-3687/

14 S. Matsushima, C. Filsfils, M. Kohno, P. Camarillo, D. Voyer "Segment Routing IPv6 for Mobile User Plane", 11 June 2023. URL: https://datatracker.ietf.org/doc/html/draft-ietf-dmm-srv6-mobile-uplane

15 Segment routing applications: Traffic matrix reconstruction and what-if visualization. URL: https://matheo.uliege.be/handle/2268.2/11455

16 What Is MPLS? URL: https://info.support.huawei.com/info-finder/encyclopedia/en/MPLS.html

17 C. Filsfils, D. Dukes (eds.) et al., “IPv6 Segment Routing Header (SRH),” RFC 8754, Mar. 2020. URL: https://tools.ietf.org/html/rfc8754

18 C. Filsfils et al., “SRv6 Network Programming,” Internet Engineering Task Force, Internet-Draft draft-ietf-spring-srv6-network-programming, Jan. 2020, work in Progress. URL: https://tools.ietf.org/ html/draft-ietf-spring-srv6-network-programming

19 P. Ventre et al., “SDN Architecture and Southbound APIs for IPv6 Segment Routing Enabled Wide Area Networks,” IEEE Transactions on Network and Service Management, 2018.

20 P. Quinn, U. Elzur, C. Pignataro, (eds.), “Network Service Header (NSH),” Internet Requests for Comments, RFC Editor, RFC 8300, January 2018. URL: https://tools.ietf.org/html/rfc8300

21 Does 5G mean we can finally update the mobile data plane protocol. URL: https://www.metaswitch.com/blog/ does-5g-mean-we-can-finally-update-the-mobile-data-plane-protocol

22 A. Mayer et al., “An Efficient Linux Kernel Implementation of Service Function Chaining for legacy VNFs based on IPv6 Segment Routing,” in 5th IEEE International Conference on Network Softwarization (NetSoft 2019). IEEE, 2019.

23 S. Previdi et al., “Segment Routing Architecture,” IETF RFC 8402, Jul. 2018. URL: https://tools.ietf.org/html/rfc8402/

24 F. Clad et al., “Service Programming with Segment Routing,” Internet Engineering Task Force, Internet-Draft draft-ietf-spring-sr-service- programming-01, Oct. 2018, work in Progress. URL: https://tools.ietf.org/html/draft-ietf-spring-sr-service-programming

25 S. Matsushima et al., “Segment Routing IPv6 for Mobile User Plane,” Internet Engineering Task Force, Internet-Draft draft-ietf-dmm-srv6- mobile-uplane, Mar. 2019, work in Progress. URL: https://tools.ietf.org/html/draft-ietf-dmm-srv6-mobile-uplane

26 F. Aubry et al., “Robustly disjoint paths with segment routing,” in Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies. ACM, 2018, pp. 204–216.

27 Makoveichuk Oleksandr, Golubenko Oleksandr, Kukhtyk Stanislav, Antonenko Artem, Bereznychenko Viktoriia, Iatsyshyn Andrii. Temperature Forecasting with LSTM: A Case Study on Kyiv Weather Data. In: Applied Information Systems and

Technologies in the Digital Society : proceedings of the 9th International Scientific and Practical Conference “Applied Information Systems and Technologies in the Digital Society” AISTDS'2025, Kyiv, Ukraine, October 1, 2025 / ed. Vitaliy Snytyuk [et al.]. CEUR-WS.org, 2025. Vol. 4133. P. 201-211. URL: https://ceur-ws.org/Vol-4133/

28 Spring Open Project. URL: https://wiki.onosproject.org/display/ONOS10/Segment+Routing

29 D. Lebrun et al., “Software resolved networks: Rethinking enterprise networks with IPv6 Segment Routing,” in Proceedings of the Sympo- sium on SDN Research. ACM, 2018, p. 6.

30 D. L. F. Duchêne and O. Bonaventure, “SRv6Pipes: enabling in- network bytestream functions,” in IFIP Networking 2018, 2018.

31 What is VPP ? URL: https://wiki.fd.io/view/VPP

32 FRRouting. URL: https://frrouting.org.

33 SRN. URL: https://github.com/segment-routing/srn

34 SRN Overview. URL: https://segment-routing.org/index.php/SRN/SRN

35 SRv6Pipes. URL: https://github.com/segment-routing/SRv6Pipes

36 SRNK. URL: https://netgroup.github.io/srnk/

37 RobustlyDisjointPathsCode. URL: https://bitbucket.org/franaubry/robustlydisjointpathscode

38 S. Matsushima et al., “SRv6 Implementation and De- ployment Status,” Internet Engineering Task Force, Internet- Draft draft-matsushima-spring-srv6-deployment-status, May 2020, work in Progress. URL: https://tools.ietf.org/html/draft-matsushima-spring-srv6-deployment-status

39 Using SRN. URL: https://segment-routing.org/index.php/SRN/UsingSRN

40 Zaitsev Ievgen, Bondarenko Oleh, Golubenko Oleksandr, Antonenko Artem, Savchenko Andrii. Securing Applied Information Systems With SAST Integration Into the Gulp Pipeline. In: Applied Information Systems and Technologies in the Digital Society : proceedings of the 9th International Scientific and Practical

Conference “Applied Information Systems and Technologies in the Digital Society” AISTDS'2025, Kyiv, Ukraine, October 1, 2025 / ed. Vitaliy Snytyuk [et al.]. CEUR-WS.org, 2025. Vol. 4133. P. 181-189. URL: https://ceur-ws.org/Vol-4133/

41 Juniper. MPLS Applications User Guide. LDP. URL: https://www.juniper.net/documentation/us/en/software/junos/mpls/topics/topic-map/ldp-overview.html"

42 Juniper. MPLS Applications User Guide. RSVP. URL: https://www.juniper.net/documentation/us/en/software/junos/mpls/topics/topic-map/rsvp-overview.html

43 Hussain, M.; Shah, N.; Amin, R.; Alshamrani, S.S.; Alotaibi, A.; Raza, S.M. Software-Defined Networking: Categories, Analysis, and Future Directions. Sensors 2022, 22, 5551. DOI: https://doi.org/10.3390/s22155551

44 Поперешняк С. В., Вечерковська А. С., Хільченко М. Ю. (2024). Хмарні технології як сервіси для оптимізації процесів адміністрування. Таврійський науковий вісник. Серія: Технічні науки, (6), 54-63. https://doi.org/10.32782/tnv-tech.2023.6.7

45 Кірюшин В. О., Лемешко А. В., Рябоконь О. О., & Макаренко Є. О. (2024). Особливості функціональності dns-сервера та dhcp-сервера. Таврійський науковий вісник. Серія: Технічні науки, (5), 55-64. https://doi.org/10.32782/tnv-tech.2023.5.6

46 Буряк, М. С., Цвик, О. С., Бурачинський, А. Ю., Балвак, А. А., & Занфіров, Р. Р. (2024). Особливості розгортання мереж мобільного зв’язку за допомогою БПЛА. Таврійський науковий вісник. Серія: Технічні науки, (3), 3-4. URL: https://doi.org/10.32782/tnv-tech.2024.3.1

47 Солобаєв, С. Г., Востріков, С. О., Ткаченко, О. В., Ходосов, А. О., & Остапенко, О. С. (2025). Використання нейронних мереж у прогнозуванні безпеки мережі. Таврійський науковий вісник. Серія: Технічні науки, (2), 3-10. URL: https://doi.org/10.32782/tnv-tech.2025.2.1

48 Черевик, О. В., Востріков, С. О., Мішкур, Ю. В., Балвак, А. А., & Дзісяк, В. Г. (2025). Оптимізація мережевої інфраструктури для забезпечення відмовостійкості. Таврійський науковий вісник. Серія: Технічні науки, (3), 3-13. URL: https://doi.org/10.32782/tnv-tech.2025.3.1

49 Твердохліб, А. О., Востріков, С. О., Бегліцов, С. В., Гергель, О. Г., & Бобков, Р. А. (2024). Оптимізація роботи бездротових мереж з використанням Inet Framework. Таврійський науковий вісник. Серія: Технічні науки, (5), 11-24. URL: https://doi.org/10.32782/tnv-tech.2024.5.2

50 Мішкур, Ю. В., Сольський, Д. Я., Солобаєв, С. Г., Подуран, Д. В., & Сарафанюк, Р. О. (2024). Оптимізація продуктивності веб-сервера apache. Таврійський науковий вісник. Серія: Технічні науки, (4), 15-30. https://doi.org/10.32782/tnv-tech.2024.4.2

51 Buriak, M. S., Vostrikov, S. O., Balvak, A. A., Korotin, D. S., & Myronenko, R. O. (2025). Можливості підвищення ефективності мереж в системах клієнт- сервер. Таврійський науковий вісник. Серія: Технічні науки, 1(4), 3-14. https://doi.org/10.32782/tnv-tech.2025.4.1.1

52 Лемешко А. В., Шрам М. М. (2023) Моделювання безпроводових мереж у середовищі OMNET++ з використанням Inet Framework. ITSynergy, (1), 37–59. (IndexCopernicus) https://doi.org/10.53920/ITS-2023-1-3

53 Лемешко А. В., Матвійчук О. М. (2023). Управління трафіком в гібридній програмно-визначеній мережі. ITSynergy, (1), 98–114 (IndexCopernicus) https://doi.org/10.53920/ITS-2023-2-7 Sciences, (3), 3-8. https://doi.org/10.21303/2504-5695.2022.002605

Додаток А

Апробація кваліфікаційної роботи

1. Поперешняк С. В., Вечерковська А. С., Хільченко М. Ю., & Антоненко А. В. (2024). Хмарні технології як сервіси для оптимізації процесів адміністрування. Таврійський науковий вісник. Серія: Технічні науки, (6), 54-63. https://doi.org/10.32782/tnv-tech.2023.6.7

2. Антоненко А. В., Кірюшин В. О., Лемешко А. В., Рябоконь О. О., & Макаренко Є. О. (2024). Особливості функціональності dns-сервера та dhcp-сервера. Таврійський науковий вісник. Серія: Технічні науки, (5), 55-64. https://doi.org/10.32782/tnv-tech.2023.5.6

3. Антоненко, А. В., Буряк, М. С., Цвик, О. С., Бурачинський, А. Ю., Балвак, А. А., & Занфіров, Р. Р. (2024). Особливості розгортання мереж мобільного зв'язку за допомогою БПЛА. Таврійський науковий вісник. Серія: Технічні науки, (3), 3-4. URL: https://doi.org/10.32782/tnv-tech.2024.3.1

4. Антоненко, А. В., Солобаєв, С. Г., Востріков, С. О., Ткаченко, О. В., Ходосов, А. О., & Остапенко, О. С. (2025). Використання нейронних мереж у прогнозуванні безпеки мережі. Таврійський науковий вісник. Серія: Технічні науки, (2), 3-10. URL: https://doi.org/10.32782/tnv-tech.2025.2.1

5. Антоненко, А. В., Черевик, О. В., Востріков, С. О., Мішкур, Ю. В., Балвак, А. А., & Дзісяк, В. Г. (2025). Оптимізація мережевої інфраструктури для забезпечення відмовостійкості. Таврійський науковий вісник. Серія: Технічні науки, (3), 3-13. URL: https://doi.org/10.32782/tnv-tech.2025.3.1

6. Антоненко, А. В., Твердохліб, А. О., Востріков, С. О., Бегліцов, С. В., Гергель, О. Г., & Бобков, Р. А. (2024). Оптимізація роботи бездротових мереж з використанням Inet Framework. Таврійський науковий вісник. Серія: Технічні науки, (5), 11-24. URL: https://doi.org/10.32782/tnv-tech.2024.5.2

7. Антоненко, А. В., Мішкур, Ю. В., Сольський, Д. Я., Солобаєв, С. Г., Подуран, Д. В., & Сарафанюк, Р. О. (2024). Оптимізація продуктивності веб-сервера apache. Таврійський науковий вісник. Серія: Технічні науки, (4), 15-30. https://doi.org/10.32782/tnv-tech.2024.4.2

8. Antonenko, A. V., Buriak, M. S., Vostrikov, S. O., Balvak, A. A., Korotin, D. S., & Myronenko, R. O. (2025). Можливості підвищення ефективності мереж в системах клієнт- сервер. Таврійський науковий вісник. Серія: Технічні науки, 1(4), 3-14. https://doi.org/10.32782/tnv-tech.2025.4.1.1

Додаток Б

Презентація

Державний університет інформаційно-комунікаційних технологій
Навчально-науковий інститут Інформаційних технологій
Кафедра Комп'ютерної інженерії

Кваліфікаційна робота на тему: КОМП'ЮТЕРНА МЕРЕЖА З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ СЕГМЕНТНОЇ МАРШРУТИЗАЦІЇ

Виконав: здобувач вищої освіти гр. КСДМ-61 Артем Антоненко

Керівник: доктор філософії, доцент Сергій Коротков

Київ - 2025

Актуальність та мета дослідження

Актуальність

Виклики сучасних мереж:

- Експоненціальне зростання трафіку та вимог до смуги пропускання.
- Інтеграція мільярдів IoT-пристроїв та хмарних сервісів.

Проблеми існуючих рішень (MPLS/RSVP-TE):

- Складність масштабування (збереження стану в проміжних вузлах).
- Складність адміністрування протоколів сигналізації.

Мета, об'єкт, предмет

Мета роботи: обґрунтувати доцільність використання Segment Routing (SR) для оптимізації маршрутизації.

Об'єкт дослідження: маршрутизація трафіку в мережах.

Предмет дослідження: оптимізація маршрутизації завдяки впровадженню технології Segment Routing.

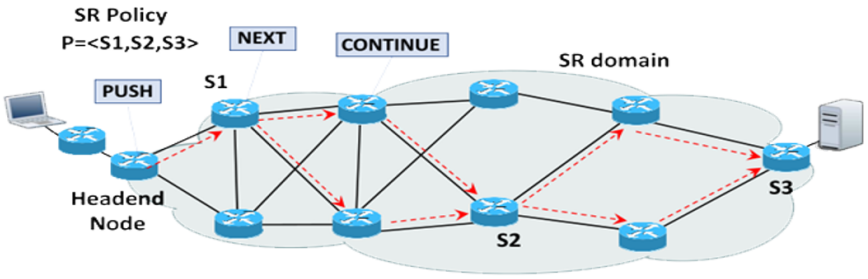
Завдання дослідження

- Дослідити механізми оптимізації маршрутизації, які вносить Segment Routing.
- Розглянути можливості реалізації SR у різних типах мереж (WAN, DC, Metro).
- Оцінити перспективи розвитку технології (SDN, 5G, IoT).
- Провести порівняльний аналіз SR з традиційними протоколами (LDP, RSVP-TE).
- Виконати порівняння двох реалізацій: SR-MPLS та SRv6.

Еволюція технологій маршрутизації та проблематика

Традиційні IGP (OSPF, IS-IS)	Забезпечують зв'язність, будують топологію Обмеження: Маршрутизація тільки за найкоротшим шляхом (SPF)
MPLS (LDP, RSVP-TE)	Додали комутацію по мітках та інженерію трафіку (TE) Проблема: RSVP-TE вимагає зберігання стану кожного тунелю на всіх проміжних вузлах. Складна сигналізація
Виклик (Чому потрібні зміни?)	Необхідність масштабування (IoT, 5G) Вимога спрощення мережі (видалення складних протоколів)

Концепція Segment Routing (SR)



Маршрутизація від джерела (Source Routing)

Вхідний вузол (Headend) обирає шлях і кодує його в заголовок пакету.

Безстановий проміжний рівень (Stateless Core)

Транзитні вузли не зберігають інформацію про потоки (на відміну від RSVP-TE).

SR Політика (SR Policy)

Впорядкований список сегментів (інструкцій), наприклад: <S1, S2, S3>.

Segment ID (SID)

Ідентифікатор сегмента (це може бути MPLS-мітка або IPv6-адреса).

Реалізація Segment Routing: SR-MPLS та SRv6

Характеристика	SR-MPLS (Evolution)	SRv6 (Revolution)
Базова технологія	MPLS (Label Switching)	IPv6 (Packet Switching)
Ідентифікатор (SID)	20-бітна мітка (Label)	128-бітна IPv6 адреса
Заголовок пакету	Стек міток MPLS	Розширений заголовок SRH (IPv6 Extension Header)
Сумісність	Працює на існуючому обладнанні (оновлення ПЗ)	Вимагає підтримки IPv6, але дає більшу гнучкість
Програмованість	Обмежена стеком міток	Висока (Network Programming)

Плавна міграція → Майбутнє мереж

Захист мережі: технологія TI-LFA

Гарантія покриття 100%

Захист працює для будь-якої топології (на відміну від класичного LFA)

Швидкість перемикання

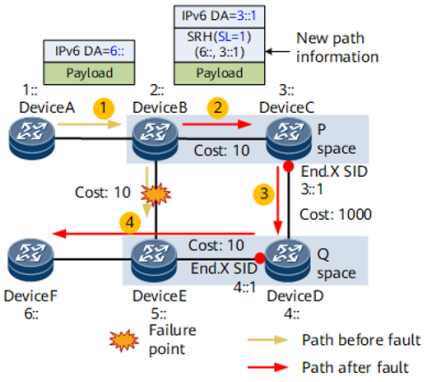
< 50 мс (стандарт телеком-операторів)

Оптимальний шлях

Резервний шлях (Backup Path) розраховується автоматично і є найкоротшим з можливих після аварії

Відсутність петель

Loop-Free

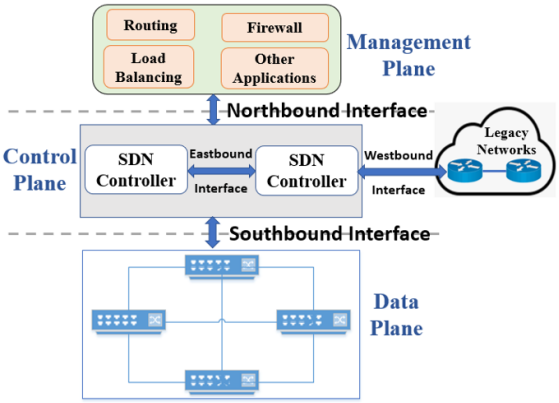


Взаємодія з SDN та централізоване керування

PCE (Path Computation Element) – центральний елемент, що обчислює шляхи

Глобальна видимість – контролер бачить усю топологію та завантаженість каналів у реальному часі

Автоматизація – шляхи програмуються автоматично за вимогою застосунку (On-demand)



Алгоритм моделювання та конфігурація SRv6

Планування
Граф мережі G=(V, E), метрики

Базовий IGP
Налаштування IS-IS

SR Activation
Активація segment-routing srv6.

SR Policies
Створення політик трафіку (Low Latency, Backup).

Верифікація
Перевірка ping та FIB.

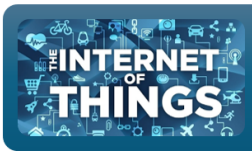
```
! Конфігурація Headend (Cisco IOS-XR)
router isis CORE
 segment-routing srv6
  locator CORE ipv62001:db8::1/128
!
traffic-eng
 policy LOW_LATENCY
  color 10 end-point ipv6
2001:db8::PE2
  candidate-paths
   preference 100
    dynamic metric type igp
   preference 200
    explicit segment-list BACKUP_PATH
```

Практичне застосування та перспективи



Мережі 5G (Network Slicing)

Segment Routing дозволяє створювати ізольовані "слайси" (зрізи) мережі для різних сервісів 5G на одній фізичній інфраструктурі



Інтернет речей (IoT)

Підтримка величезної кількості пристроїв завдяки IPv6 (SRv6) без перевантаження мережі протоколами сигналізації



Ланцюжки сервісів (SFC - Service Function Chaining)

Можливість спрямувати трафік через послідовність сервісів (Firewall -> NAT -> DPI) просто додавши їх SID у заголовок пакету

Висновки

Аналіз недоліків. Встановлено, що традиційні протоколи (LDP, RSVP-TE) обмежують масштабованість сучасних мереж через складність сигналізації.

Ефективність SR. Доведено, що Segment Routing значно спрощує архітектуру мережі, усуваючи необхідність збереження стану потоків на транзитних вузлах.

Надійність. Технологія TI-LFA забезпечує гарантований захист трафіку (100% покриття топології) з часом відновлення < 50 мс.

Перспективність. Підтверджено готовність архітектури до впровадження в мережах 5G та інтеграції з SDN-контролерами.

Подальші кроки (Перспективи розвитку)

Міграція на SRv6

Дослідження сценаріїв повного переходу від MPLS до IPv6-native інфраструктури (Network Programming).

Автоматизація (NetDevOps)

Розробка скриптів (Python/Ansible) для автоматичного розгортання політик маршрутизації (SR Policies).

Інтеграція з хмарами

Налаштування наскрізної маршрутизації від користувача до хмарних сервісів (Cloud-Native networking).

Державний університет інформаційно-комунікаційних технологій
Навчально-науковий інститут Інформаційних технологій
Кафедра Комп'ютерної інженерії

Кваліфікаційна робота на тему: КОМП'ЮТЕРНА МЕРЕЖА З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ СЕГМЕНТНОЇ МАРШРУТИЗАЦІЇ

Виконав: здобувач вищої освіти гр. КСДМ-61 Артем Антоненко
Керівник: доктор філософії, доцент Сергій Коротков

ДЯКУЮ ЗА УВАГУ!

Київ - 2025