Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут телекомунікаційних систем
(повна назва інституту/факультету)

Кафедра телекомунікацій
(повна назва кафедри)

«На правах рукопису»
УДК _____

До захисту допущено
В.о. завідувача кафедри

_____ Явіся В.С.___
(підпис) (ініціали, прізвище)

“___”_____2018_р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 172 Телекомунікації та радіотехніка,
(код і назва)

спеціалізація Апаратно-програмні засоби електронних комунікацій

на тему: Принципи побудови і методи управління трафіком в мережах MPLS

Виконав: студент 2 курсу, групи ТЗ-71мп
(шифр групи)

Денисюк Сергій Володимирович
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник професор, Романов Олександр Іванович
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018_ рік

ЗМІСТ

Список скорочень……..

Вступ…….

1. Основні відомості про технологію MPLS, її складові та принципи їх функціонування…….

1.1 Передісторія виникнення технології MPLS…….

1.2. Основні поняття технології MPLS……..

1.3 Класи еквівалентності пересилання FEC…….

1.4 Комутовані по міткам тракти LSP…….

1.5 Комутація по міткам……...

1.6 Структура мітки…….

1.7 Стек міток MPLS…….

1.8 Інкапсуляція міток…….

1.9 Таблиці пересилання……..

1.10 Прив'язка «мітка-FEC»……...

1.11 Режими операцій з мітками……..

1.12 Фіксовані значення мітки……

Висновки……...

2. Інжиніринг трафіку в мережах MPLS, огляд та порівняльна характеристика протоколів RSVP-TE та CR-LDP……...

2.1 Концепція інжинірингу трафіку в MPLS……...

2.2 Протоколи сигналізації……..

2.3 Атрибути потоків трафіку і мережевих ресурсів……..

2.4 Атрибути об'єднаних потоків трафіку…….

2.5 Атрибути мережевих ресурсів……..

2.6 Маршрутизація на основі обмежень……...

2.7 Механізми TE в MPLS……...

2.8 Порівняння протоколів CR-LDP і RSVP - TE……...

2.8.1 Порівняння функціональних можливостей……..
2.8.2 Порівняння технічних характеристик……..
2.8.3 Службовий трафік в RSVP-TE і CR-LDP…….
2.8.4 Порівняння перемаршрутизації в RSVP - TE і CR - LDP…….
2.8.5 Порівняння протоколів по їх впровадженню……
Висновки…….
3. Практична реалізація RSVP - TE за допомогою GNS3……
Висновки……..
Список використаної літератури……..

ПЕРЕЛІК СКОРОЧЕНЬ

MPLS	Multiprotocol Label Switching - багатопротокольна комутація за мітками
FEC	Forwarding Equivalence Class - клас еквівалентності пересилання
LSP	Label Switched Path - комутований по мітках тракт
ER - LSP	Explicitly routed LSP - LSP з явно заданим маршрутом
LER	MPLS edge router - граничний вузол мережі MPLS
IPLSR	Label Switching Router - маршрутизатор комутації по мітках
IP	Internet Protocol - інтернет протокол, міжмережевий протокол
TCP	Transmission Control Protocol - протокол керування передачею
UDP	User Datagram Protocol - протокол датаграм користувача
RSVP	Resource ReSerVation Protocol - протокол резервування мережевих ресурсів
LDP	Label Distribution Protocol - протокол розподілу міток
TE	Traffic Engineering - інжиніринг трафіку
QoS	Quality of service - якість обслуговування
LSR	Label switching router - комутуючий мітки маршрутизатор
LSP	Label switch path - шлях комутації міток

Вступ

З самого свого виникнення комп’ютерні мережі виконували дуже важливу функцію: з’єднували між собою окремі вузли в одну мережу для передачі даних. Саме передача даних являлася головною причиною виникнення таких мереж та їх розвитку. Великі корпорації шукали шляхи для вирішення головної задачі: передачі інформації, спочатку в межах одного офісу, а потім між віддаленими філіалами.

У сучасному світі мережі вже давно перестали бути прикладною частиною компанії, і стали одним із ключових пунктів у її структурі. Ні одна організація не може повноцінно функціонувати і здійснювати свою діяльність без правильно побудованої комп’ютерної мережі. Дане твердження відноситься не тільки до великого та середнього бізнесу, але й до малого.

Крім того, варто зауважити, що для побудови якісної корпоративної мережі вже недостатньо тільки підключити виділений канал Інтернету. Якщо раніше це й було можливим та підходящим рішенням, то зараз, із врахуваннями сучасних реалій, воно таким не являється. І на це є кілька підстав.

По-перше, з кожним роком зменшується кількість адрес IPv-4, а IPv-6 поки що не знайшов широкого застосування. Рішення цієї проблеми було знайдемо вже давно: за допомогою звичайних транспортних каналів офіси з’єднувалися з центральним або між собою, а взаємодія із зовнішнім світом відбувалася через одну точку. Це хороше рішення, у випадку, якщо Вам не потрібно додаткові етапи захисту або додаткова надійність.

По – друге, програмне забезпечення, яке зараз використовується в компаніях для виконання тих чи інших задач набагато вимогливіше до взаємодії клієнт – сервер. Раніше для роботи віддаленого філіалу було достатньо просто запустити переглядач електронної пошти, і співробітник вже міг виконувати свої функції на повну.

Існування, як мінімум, дев'яносто дев'яти претендентів на позицію головного механізму забезпечення якості обслуговування QoS в мережах зв'язку наступного по-

колінах NGN (Next Generation Network) обумовлено швидким зростанням числа користувачів IP мереж і відповідним збільшенням мультимедійного трафіку. Спочатку ж IP технології були орієнтовані на передачу даних найпростіших додатків, для чого було цілком достатньо програмних маршрутизаторів. У міру появи нових мультимедійних додатків типу IP телефонії, що вимагають більш високих швидкостей передачі і підтримки більш широкої смуги пропускання, виникла потреба в створенні технологій та пристроїв, що забезпечують можливість швидкої комутації на рівні 2 і 3. Відповідно, з'явився пристрій комутації на рівні 2, вирішальна проблема вузьких місць комутації в середовищі LAN, а також нові маршрутизатори, що поліпшують ситуація з маршрутизацією на рівні 3 шляхом переводу в швидкодіючої апаратної реалізації процедури перегляду маршрутних таблиць для пересилання пакетів. Досяг свій апогей найбільш перспективних технології 90-х років Frame Relay і ATM. З'явилися різноманітні засоби забезпечення якості обслуговування DiffServ і IntServ, протокол резервування RSVP і багато іншого.

І все ж, всі ці дев'яносто дев'ять претендентів на найбільш ефективне рішення проблеми мережевого QoS при передачі мультимедійної інформації в реальному часі з урахуванням таких показників, як затримка, тремтіння фази, перевантаження і т.п. фактично вже поступилися зайнятій лідируючу позицію багатопротокольній комутації по мітках MPLS (Multiprotocol Label Switching).

MPLS є вельми витонченими і універсальним рішенням проблем QoS, що стоїть перед сьогоднішньою пакетної мережами, рішення, яке забезпечує швидкість передачі, масштабованість, оптимізацію розподілу трафіку і ефективну маршрутизація (на основі показників QoS) в пакетних мережах IP, ATM і Frame Relay. Лідерство MPLS обумовлене вдало обраній позиції, що дозволяє оптимальним чином відображати наскрізний трафік третього рівня від вихідного мережевого вузла (маршрутизатора) до вхідного вузла в трафіку між сусіднім вузлами на другому рівні мережевий ієрархії. Тому можна сказати, що MPLS, будучи гібридом рівнів 2 і 3 OSI

семирівневої моделі, зібрала разом найкраще з двох світів: рівня 2 і рівня 3, світу ATM та світу IC.

Деяка упередженість у відповідь на настільки значну увагу до технології MPLS проявилася у виниклій плутанині щодо того, що таке MPLS, до якого рівня OSI вона відноситься, що ця технологія може і для чого вона призначена. Така плутанина зумовлена, зокрема, тим, що технологія MPLS не використовує який-небудь єдиний формат для транспорту пакетів, а базується на кількох протоколах (і відповідних їм стандартах), кожен з яких вирішує окрему проблему. Спроба розплутати цей клубок і якимось чином структурувати опис протоколів MPLS якраз і зроблено в даній роботі.

1. Основні відомості про технологію MPLS, її складові та принципи їх функціонування

1.1 Передісторія виникнення технології MPLS

Технологія багато протокольної комутації по мітках MPLS з'явилася як результат злиття кілька подібних технології, які були винайдені в середині 1990-ого року. Ці механізми мають ряд спільних рис. Всі вони використовують для пересилання пакетів простий метод заміни міток і розроблену для Інтернет структуру управління, тобто IP-адреси і стандартні протоколи маршрутизації, наприклад OSPF і BGP.

Зростаючий інтерес до комутації з використанням міток привів до створення спеціальної робочої групи IETF, метою якої було виробити на основі вищезазначених механізмів загальний стандарт. Аби не допустити давати групу назву, яке відповідало б продукту якоїсь із компаній, IETF зупинив свій вибір на нейтральній (правда, злегка громіздкій) назві: багато протокольна комутація по мітках. Багато протокольна комутація MPLS називається так тому, що її засоби застосовні до будь-якого протоколу мережевого рівня, тобто MPLS - це свого роду інкапсулююий протокол, здатний транспортувати інформацію множини протоколів нижчих рівнів моделі OSI.

Нагадаємо, що перший, фізичний рівень містить функції, що забезпечують використання фізичного середовища для двосторонньої передачі бітів (з такою достовірністю, який забезпечує це середовище) по прямому тракту, що зв’язує два вузли мережі. Другий рівень -рівень ланки даних (канальний) - містить функції, що забезпечує формування в цьому тракті надійної логічної ланки зв'язку, в якому проходить двосторонній обмін інформаційних блоків між названими вузлами; при цьому шляхом виявлення та виправлення помилок гарантується задана достовірність передачі. Третій, мережевий рівень містить функції, що забезпечують транспортування інформаційних блоків від відправника до одержувача через кілька вузлів мережі по невластивому маршруту, який складається з ланок другого рівня.

Загальна ідея протоколів всіх рівнів (крім фізичного) полягає в тому, що інформаційний блок кожного рівня містить заголовок і інформаційні поле, і в тому, що блок протоколу вищого рівня поміщається в інформаційне поле блоку протоколу, розташованого відразу під ним нижчого рівня.

Однак у всіх цих попередніх MPLS роботах не піддавався сумніву базовий принцип: маршрутизатори виконують функції маршрутизації, а комутатори виконують функції комутації, і влаштування цих двох типів завжди функціонують окремо. При цьому маються на увазі не тільки IP-маршрутизатори і ATM- комутатори, а й саме правило поділу функцій рівнів 3 і 2 між різними технологіями і пристроями. Подальша історія відповідає вислову, який приписують Альберту Ейнштейну: «Всі давно знають, що то і то абсолютно неможливе. Але ось знаходиться невіглас, який цього не знає, він і робить відкриття ».

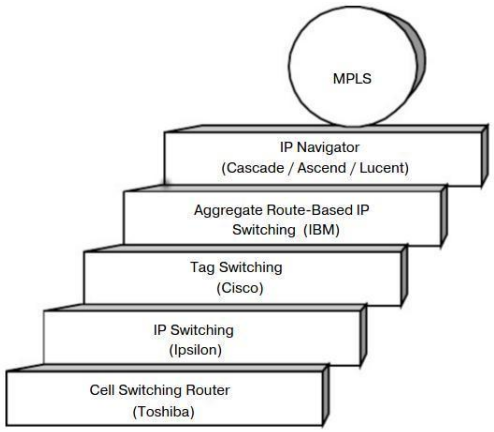


Рисунок 1.1 Етапи еволюції технологій до технології MPLS

У ролі такого «невігласа» виступила компанія Toshiba, практично вперше поставила під сумнів цей принцип і в 1994 році анонсувала маршрутизатор комутації осередків КСВ (стільниковий маршрутизатор комутації). В архітектурі CSR реалізована ідея управління комутаційного поля ATM-комутатором за допомогою протоколів IP, а не протоколами сигналізації мережі ATM типу Q.2931. Подібний підхід, будучи доведеним до логічного завершення, зміг би звести нанівець

необхідність використання практично всієї сигналізації ATM і всі функції маппінга між IP і ATM.

Компанії Ipsilon, завдяки більш повної технічної специфікації IP Switching наявність готового продукту IP switch, зазвичай приписують честь створення першої, по-справжньому формалізованої концепції MPLS, подобної по комутації міток в мережах IP, що отримала значно більше визнання, ніж технологія КСА. Сам IP комутатор складався з ATM-комутатора і контролера IP-комутатора, який виконував функцію управління. Контролер IP-коммутатора фактично був окремим пристроєм, що містить функціональні об'єкти маршрутизації і пересилання даних.

Незабаром компанія Cisco Systems анонсувала свій варіант технології комутації по мітках під назвою комутація по тегам (Тег Switching), яка істотно відійшла від двох розглянутих вище технологій IP Switching і CSR. Зокрема, для створення таблиць пересилання в комутаторі вона не спиралася на потік трафіку даних і, до того ж, була специфікована для ряду технологій рівня 2, відмінних від ATM. Таким чином, технологія Tag Switching виявилася ближче до остаточної концепції MPLS, ніж механізм IP Switching. Більш того, технологія MPLS в значній мірі вийшла з механізму Tag Switching. Сам так званий тег, тобто фіксована кількість біт, що використовується для адресації, багато в чому аналогічна мітка MPLS. Механізм Tag Switching призначався для спільної роботи з рядом протоколів нижніх рівнів і включав в себе протокол розподілу ключових слів (Протокол розподілу тегів, TDP), як і в MPLS, механізм Tag Switching підтримував освіту стека тегів. Крім більш швидкого пошуку адреси, нові маршрутизатори могли обслуговувати виклики по-різному залежності від необхідної якість обслуговування (при передачі мови, відео та зображення). До того ж, всі маршрутизатори виробництва Cisco, в яких був реалізований механізм Tag Switching, були пізніше модернізовані і змогли підтримувати MPLS.

Практично відразу ж після того, як Cisco опублікувала інформацію про технологію Tag Switching і оголосила про її стандартизації в IETF, від корпорації IBM надійшли проекти Інтернет стандартів, в яких пропонувалася інша технологія

комутації по мітках — Agregate Route-based IP Switching (APIC). Механізм ARIS призначався для використання з ATM- і FR-коммутаторамb, а також з пристроями комутації на рівні 2 в локальних мережах. Пристрій, в якому був реалізований механізм ARIS, отримав назву ARIS вбудований комутатор-маршрутизатор (ISR), технологія ARIS має більше спільних рис з технологією Tag Switching, ніж з іншими вже згадуваними технологіями, - в обох для створення таблиць пересилання використовується трафік керуючої інформації, а не трафік даних, - але при цьому технологія ARIS має деякі суттєві відмінності від Tag Switching. Основна відмінність полягає в тому, що ARIS заснований на маршрутах, а не на потоках. Маршрути в домені ARIS будуються на базі вихідного вузла. Конфігуруються вихідні вузлів домен ARIS, а потім від них поширюється маршрути в сторону вхідного вузла. Вихідний вузол може бути заданий поруч ідентифікаторів: префіксом одержувача протоколу IPv4, IP-адресою вихідного маршрутизатору, ідентифікатором маршрутизатору OSPF або ідентифікатором пари багатоадресної передачі. Маршрути встановлюють незалежно від потоків пакетів.

Ще однією технологією, що передувала MPLS, є технологія IP Navigator, запропонована компанією Cascade. Каскад була потім куплена компанією Ascend, яка, в свою чергу, стала частиною компанії Lucent Technologies. В технології IP Navigator були використано багато ідей комутації в IP-мережах, розроблених раніше компаніями Toshiba, Ipsilon, Cisco і IBM.

Після публікації першої серії проектів стандартів Tag Switching 9-13 грудня 1996 року в Сан-Дієго, Каліфорнії, відбувалася рекордна за всю історію IETF по відвідуваності сесія BOF, на якій Cisco Systems, IBM і Toshiba провели презентації своїх технологій. Такий інтерес, а також той факт, що настільки багато провідних компаній розробили багато в чому близькі технічні пропозиції для вирішення проблеми, дозволили зробити очевидний висновок про необхідність створити для стандартизації механізму комутації по мітках спеціальну групу. У квітні 1997 року в Мемфісі, Теннісі, відбулося перше засідання робочої групи MPLS WG. Сама назва

Multiprotocol Label Switching було прийнято, в першу чергу, з тієї, вже згаданої нами причини, що назви IP Switching і Tag Switching асоціювалися з продуктами, що випускаються конкретними компаніями, і був потрібний нейтральний термін.

1.2. Основні поняття технології MPLS

Отже, MPLS, може розглядатися як сукупність технологій які, працюючи спільно, забезпечують доставку пакетів від відправника до одержувача в контрольований, ефективний і передбачуваний спосіб. У MPLS для пересилання пакетів використовуються комутовані по мітках тракти LSP, які були організовані за допомогою протоколів маршрутизації і сигналізації рівня 3.

FEC - Forwarding Equivalence Class - клас еквівалентності пересилання - безліч пакетів, що пересилаються однаково, наприклад, з метою забезпечити заданий рівень QoS.

Label — мітка - короткий ідентифікатор фіксованої довжини, що визначає приналежність пакета того чи іншого FEC.

Label swapping - заміна міток - заміна мітки прийнятого вузлом мережі MPLS пакета новою міткою, пов'язаною з тим же FEC, при пересиланні пакета до нижчому вузлу.

LER - (MPLS edge router - граничний вузол мережі MPLS) - прикордонний вузол мережі MPLS, який з'єднує домен MPLS з вузлом, що знаходиться поза цим доменом.

Loop detection - виявлення закільцьованих маршрутів - метод, що дозволяє виявити, що пакет пройшов через вузол більше одного разу.

Loop prevention - попередження створення закільцьованих маршрутів - метод виявлення і усунення закільцьованих маршрутів.

LSP - (Label Switched Path) комутований по мітках тракт — приходящий через один або більше LSR тракт, по якому йдуть пакети одного і того ж FEC.

ER - LSP - (explicitly routed LSP) - LSP з явно заданим маршрутом - тракт LSP, який організований у спосіб, відмінний від традиційної маршрутизації пакетів.

IPLSR - (Label Switching Router) - маршрутизатор комутації по мітках - маршрутизатор, здатний пересилати пакети за технологією MPLS.

MPLS domain - домен MPLS - сукупність вузлів MPLS, між якими існують безперервні LSP.

MPLS egress node - вихідний вузол мережі MPLS - останній MPLS - вузол в LSP, що направляє вихідний пакет до адресату, який знаходиться поза MPLS-мережею.

MPLS ingress node - вхідний вузол мережі MPLS - перший MPLS-вузол в LSP, що приймає вихідний пакет і поміщає в нього мітку MPLS.

1.3 Класи еквівалентності пересилання FEC

Робочими групами, згаданими в попередньому параграфі, визначені три основні елементи технології MPLS: FEC - Forwarding Equivalency Class - клас еквівалентності пересилки, LSR - Label Switching Router - маршрутизатор комутації по мет-кам і LSP - Label Switched Path - комутований по мітках тракт. Почнемо з класів еквівалентності пересилки.

При традиційній транспортуванні пакета через мережу з використанням протоколу рівня 3, який не передбачає створення віртуальних з'єднань, кожен маршрутизатор на шляху слідування пакета самостійно приймає рішення про те, до якого маршрутизатора переслати цей пакет далі (спосіб транспортування hop-by-hop). Інакше кажучи, в кожному маршрутизаторі на шляху проходження пакету аналізується його заголовок і виконується алгоритм мережевого рівня. Тут і далі використовується англійське слово hop - стрибок , - а в термінах маршрутизації - одне пересилання. Під пересиланням пакета розуміється його передача до найближчого маршрутизатора з тих, що розташовані на можливому шляху прямування цього пакета, тобто слово «пересилання» використовується як еквівалент англійського слова forwarding.

У заголовку пакета міститься набагато більше інформації, ніж потрібно для того, щоб вибрати наступний маршрутизатор. Цей вибір можна організувати простіше

- шляхом виконання двох функцій. Одна з них полягає в поділі всієї множини пакетів, що прибувають, на класи, які називаються класами еквівалентності пересилання FEC (Forwarding Equivalence Classes). Друга ставить у відповідність кожному FEC певний «напрямок» пересилки (слово «напрямок» написано в лапках тому, що в мережі використовується режим hop-by-hop, і різні пакети одного і того ж FEC можуть пересилатися до різних маршрутизаторів, тобто фізичні напрямки пересилання можуть бути різними). З точки зору вибору наступного маршрутизатора все пакети, що належать одному FEC, невиразні.

Ідея класів еквівалентності більш універсальна, ніж MPLS. При традиційній IP-маршрутизації той чи інший маршрутизатор теж може вважати, що два пакети належать одному і тому ж умовному класу еквівалентності, якщо в його таблицях маршрутизації використовується якийсь адресний префікс, що ідентифікує напрям, в якому передбачувані маршрути транспортування цих двох пакетів збігаються найдовше. У міру просування пакету по мережі кожен наступний маршрутизатор аналізує його заголовок і приписує цей пакет до такого з власних, (певних тільки в даному маршрутизаторі) класів еквівалентності, який відповідає тому ж напрямку.

При використанні ж багато протокольної комутації по міткам MPLS пакет приписується до певного класу FEC тільки один раз, коли він потрапляє в мережу. Цьому FEC присвоюється мітка - ідентифікатор фіксованої довжини, який подається із пакетом, коли той пересилається до наступного маршрутизатора. Завдяки цьому в інших маршрутизаторах заголовок мережевого рівня не аналізується. Мітка, встановлена прикордонним маршрутизатором при вході пакета в MPLS-мережу, використовується як покажчик входу таблиці, яка визначає черговий маршрутизатор для пересилання до нього пакету, а також нову мітку для FEC, до якого належить пакет.

Таким чином, клас еквівалентності пересилки FEC є формою представлення групи пакетів з однаковими вимогами до напрямку їх передачі, тобто всі пакети в такій групі об’єднуються в маршрутизаторі однаково і однаково йдуть до пункту

призначення. Прикладом FEC можуть служити всі IP-пакети з адресами пунктів призначення, відповідними деякого префіксу, наприклад, 212.18.6. Можливі також FEC на основі префіксу адреси і ще якого-небудь поля IP-заголовку, наприклад, тип обслуговування (ToS). Кожен маршрутизатор мережі MPLS створює таблицю, за допомогою якої визначає, яким чином повинен пересилатися пакет. Ця таблиця, яка називається інформаційною базою міток LIB, містить безліч міток і для кожної з них - прив'язку «FEC-мітка». Мітки, що використовуються маршрутизатором LSR при прив'язці «FEC-мітка», поділяються на такі категорії:

- на платформовій основі, коли значення міток унікальні по всьому тракту LSP; мітки вибираються із загального пулу міток, і ніякі дві мітки, що розподіляються за різними інтерфейсами, не мають однакових значень;
- на інтерфейсній основі, коли значення міток пов'язані з інтерфейсами: для кожного інтерфейсу визначається окремий пул міток, з якого для цього інтерфейсу і вибираються мітки. При цьому мітки, які призначаються для різних інтерфейсів, можуть бути однаковими.

Метод пересилки пакетів на основі прив'язки «FEC-мітка», прийнятий в MPLS, має ряд переваг перед методами, заснованими на аналізі заголовка блоків мережевого рівня. Зокрема, пересилання по методу MPLS можуть виконувати маршрутизатори, які здатні читати і замінювати мітки, але при цьому або взагалі не здатні аналізувати заголовки блоків мережевого рівня, або не здатні робити це досить швидко.

Так чи інакше, дії маршрутизатора LSR залежать від значення мітки, яку він приймає від попереднього LSR. Фактично, дії, що виконуються LSR, специфіковані в Next Hop Level Forwarding Entry (NHLFE), який вказує наступну ділянку, операцію, яка повинна бути виконана зі стеком міток і кодування, яке слід використовувати для стека в вихідному тракті. Виконувана зі стеком операція може полягати в тому, що LSR повинен змінити мітку на вершині стека. Ця операція може зажадати, щоб LSR просто виштовхнув верхню мітку з стеку, або виштовхнув і замінив її новою, або просто помістив нову мітку над тією, яка до цього була верхньої, нічого не

виштовхуючи і не замінюючи). Наступним ділянкою для оброблюваного пакета з мітками може виявитися і той же самий LSR. В цьому випадку LSR виштовхує верхню мітку стека і пересилає пакет самому собі. У цей момент пакет може мати ще одну мітку, яку слід аналізувати, чи опинитися без міток, тобто вихідним пакетом IP.

В останньому випадку пакет пересилається відповідно до стандартної маршрутизації IP.

Якщо маршрутизатор виявляє, що він виявився передостаннім LSR в тракті, то він повинен видалити весь стек і передати пакет в останній LSR. Завдяки цьому мінімізується обсяг обробки, яку повинен виконати останній LSR. Те, яким чином LSR визначає, що він в даному тракті передостанній, є завданням розподілу міток і використовуваного для цього протоколу розподілу міток.

1.4 Комутовані по міткам тракти LSP

LSP організовуються або перед передачею даних (з керуванням від програми), або при виявленні певного потоку даних (керовані даними). Мітки в LSP призначаються за допомогою протоколу розподілу міток LDP (Label Distribution Protocol), причому існують різні способи такого розподілу на основі даних допоміжних протоколів.

Головне завдання розподілу міток - це організація і обслуговування трактів LSP, в тому числі, визначення кожної прив’язки «FEC-мітка» в кожному LSR тракту LSP. Маршрутизатор LSR помагає протоколу розподілу міток, щоб інформувати про прив'язку «FEC-мітка» вищестоящий LSR. Нижчий LSR може безпосередньо повідомляти про прив'язку «мітка-FEC» вищестоящому LSR, що називається прив'язкою з ініціативи нижчестоящого (unsolicited downstream). Крім того, можливо повідомлення про прив'язання, що передається нижчестоящим на вимогу (downstream on demand), Коли вищестоящий LSR запитують прив'язку у нижчестоящого LSR. Організований LSP завжди є однобічним. Трафік зворотного напрямку йде по іншому LSP. Технологія MPLS підтримує наступні два варіанти створення LSP:

- послідовна маршрутизація по ділянках маршруту (hop-by-hop routing) - кожен LSR самостійно вибирає наступну ділянку маршруту для даного FEC. Ця методологія подібна до тієї, що застосовується зараз в IP-мережах. LSR використовує наявні протоколи маршрутизації, такі, наприклад, як OSPF;
- явна маршрутизація (ER) - подібна до методом маршрутизації з боку відправника. Вхідний LSR (тобто LSR, від якого виходить потік даних в мережі MPLS) специфікує ланцюжок вузлів, через які проходить ER-LSP.

Специфікований тракт може виявитися не оптимальним. Уздовж тракту можуть резервуватися ресурси для забезпечення заданого QoS трафіку даних. Це полегшує оптимальний розподіл трафіку по всій мережі і дозволяє надавати диференційоване обслуговування потокам трафіку різних класів, сформованих на основі прийнятих правил і методів управління мережею.

Таким чином, в MPLS-мережі є маршрутизатори двох типів: прикордонні LSR і транзитні LSR. Прикордонні маршрутизатори LSR в ряді випадків включають в себе шлюзи інтерфейсів мереж різних видів (наприклад, Frame Relay, ATM або Ethernet) і пересилають їх трафік в MPLS-мережу після організації трактів LSP, а також розподіляють трафік зворотного напрямку при виході його з MPLS-мережі. До цього слід додати, що будь-який MPLS-сумісний маршрутизатор повинен бути здатний приймати в будь-якому своєму інтерфейсі пакет зі вставленою міткою, відшукувати її в таблиці комутації, вставляти нову мітку в відповідному форматі і потім відправляти пакет через інший інтерфейс. Іншими словами, прикордонний LSR може комутувати пакет з міткою від будь-якого інтерфейсу до будь-якого іншого інтерфейсу з заміною мітки. Такий підхід набагато гнучкіший, ніж в разі ATM, так як він не обмежений виключно каналами передачі осередків. Прикордонні маршрутизатори виконують основну роль в процесі призначення та видалення міток, коли трафік надходить в MPLS-мережу або виходить з неї.

При цьому корисно відзначити, що будь-який транзитний LSR здатний приймати пакети без міток, тобто зі звичайними IP-заголовками. Доволі часто

зустрічається в літературі твердження, що всередині домену MPLS пакети між транзитними LSR маршрутизуються тільки по мітках, не зовсім вірно. Для звичайного MPLS-трафіку це дійсно так, але службові повідомлення передаються із використанням користуванням IP-заголовків.

1.5 Комутація по міткам

З самого визначення MPLS видно, що мітки - основа основ цієї технології. Саме з мітками виконуються процедури їх розподілу по маршрутизаторів LSR і процедури створення трактів LSP, за якими будуть слідувати пакети MPLS. Після розподілу міток і створення трактів LSP може виконуватися головна функція MPLS - пересилання забезпечених мітками пакетів по мережі MPLS. Крім цієї функції повинні вирішуватися і допоміжні завдання, пов'язані з мітками, а саме, контроль часу збереження міток, впорядкування міток і обробка помилок.

У вказаному прикладі домен мережі MPLS визначений як сукупність вузлів LSR, між якими створені безперервні LSP. Розглянемо докладніше основні кроки алгоритму, який виконується в відношенні пакетів даних в ньому. На базі рис. 1.2, перечислимо основні кроки, які необхідні, щоб забезпечити проходження пакета даних через домен MLPS:

Крок 1. Створення і розподіл міток. До початку передачі через мережу MPLS пакетів трафіку будь-якого виду маршрутизатори LSR встановлюють відповідність між мітками і FEC в своїх таблицях. Далі буде показано, як нижчестоящі маршрутизатори за допомогою сигналізації LDP, що використовує транспортний протокол TCP, виробляють розподіл міток і прив'язку їх до класів FEC. Крім того, проводиться погодження характеристик трафіку і функціональних можливостей MPLS.

Нагадаємо, що значення міток можуть вибиратися і розсилатися або заздалегідь, до передачі даних (крива 2 на рис. 1.2), або генеруватися як пакети, що належать вступнику в мережу MPLS певного потоку даних або трафіку певного класу (крива 3).

Ці два підходи до призначення міток, як зазначалося в попередньому розділі, називаються, відповідно, призначенням з керуванням від програми і призначенням, керованим трафіком (даними). Але після того як домен комутації по мітках налаштований для обслуговування пакетного трафіку, що пересилається через MPLS-мережу за допомогою міток, всі пакети обробляються однаково.

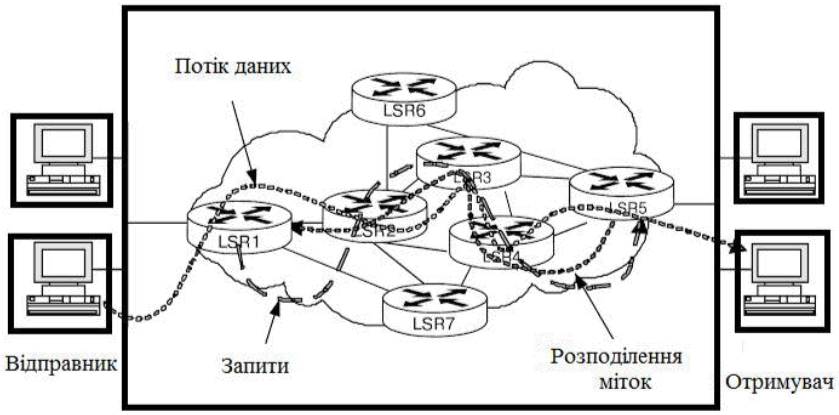


Рисунок 1.2 Проходження пакету через домен MPLS

Крок 2. Створення таблиці в кожному LSR. При отриманні даних про прив'язку міток до FEC кожен маршрутизатор LSR створює записи в таблиці LIB. Вміст таблиці відображає відповідність між мітками і FEC і ставить у відповідність кожній парі «вхідний інтерфейс, що входить мітка» пару «вихідний інтерфейс, що виходить мітка». При будь-якому новому узгодженні прив'язки міток до FEC записи в таблиці оновлюються. Структура таблиці буде розглянута далі в цій главі. Слід нагадати, що таблиці міток, відповідно яким кожен пакет направляється за відповідним тракту LSP, завжди повинні бути задані до того, як пакет почне свій шлях по мережі. Крім того, комутований по мітках тракт - завжди односторонній. Якщо необхідно, щоб пакетний трафік між двома прикордонними LSR проходив і в протилежному напрямку, слід створити два тракти.

Крок 3. Створення комутованого по мітках тракту LSP. Як показано лінією 3 на рис. 1.2, тракти LSP створюються в напрямку, протилежному створення

записів в таблицях LIB. Нагадаємо, що кожен LSR отримує мітку від нижчестоящого маршрутизатору. LSP створюється шляхом послідовної маршрутизації по ділянках, а якщо потрібна оптимізація розподілу трафіку, для визначення тракту використовується протокол CR-LDP, що гарантує виконання вимог до QoS / CoS, або протокол RSVP-TE.

Крок 4. Табличний пошук і інкапсуляція мітки в пакет. Вхідний маршрутизатор (LSR1 на рис. 1.2), визначивши, яким FEC належить прийнятий ним ззовні пакет, використовує таблицю LIB, щоб відшукати потрібну прив'язку «FEC-метка», і інкапсулює цю мітку способом, відповідним застосовуваної на рівні 2 технології, як це буде показано нижче.

Крок 5. Пересилання пакета. Розглянемо проходження пакета від вхідного маршрутизатора LSR1 до вихідного маршрутизатора LSR5. Відзначимо, що LSR1 може не мати мітки для цього пакета. В такому випадку він знаходить наступний маршрутизатор по IP-адресі. Нехай таким маршрутизатором для LSR1 є LSR2. Маршрутизатор LSR1 ініціює запит мітки від LSR2. Отриману мітку LSR1 вставляє в пакет і пересилає його до LSR2. Кожен наступний LSR (в даному випадку - LSR3 і LSR4) аналізує мітку, що міститься в прийнятому пакеті, замінює її вихідної міткою і пересилає пакет далі. Коли пакет досягає LSR5, відкидає мітку з пакета, оскільки пакет залишає домен MPLS, і доставляє пакет адресату. Тракт LSP, по якому проходить пакет, показаний переривчастими лініями 3.

Розглянемо докладніше використовуваний на кроці 5 алгоритм заміни міток при пересиланні пакетів через домен MPLS. Отримавши пакет, маршрутизатор LSR витягує з нього мітку і використовує її в якості індексу в своїй таблиці пересилання. Як тільки знайдено запис, в якій значення вхідної мітки дорівнює значенню мітки, витягнутої з пакета, маршрутизатор, згідно підзапису цього запису, замінює вхідну мітку в пакеті вихідною міткою і пересилає пакет через вихідний інтерфейс, вказаний в підзапису, до наступного LSR , також зазначеному в цьому підзаписі. Якщо підзапис вказує певну вихідну чергу, маршрутизатор ставить пакет саме в цю чергу. Простота

алгоритму пересилання пакетів, що використовується в MPLS, обумовлює просту і економічну його реалізацію в апаратному забезпеченні, що, в свою чергу, дозволяє підвищити продуктивність пересилання без використання дорогої апаратури. Якщо LSR підтримує не одну, а кілька таблиць (по одній для кожного зі своїх інтерфейсів), то єдина зміна алгоритму полягає в тому, що після отримання пакету LSR попередньо вибирає ту таблицю, яка буде використовуватися для обробки пакета. Вибір таблиці проводиться згідно ідентифікатору інтерфейсу, через який пакет був отриманий.

Таким чином, мітка, що переноситься в складі пакету, завжди передає семантику пересилання, тому що вона однозначно визначає потрібний запис в таблиці, яку веде LSR, і тому що цей запис містить інформацію про те, куди пересилати пакет. В якості опції мітка може також передавати семантику резервування ресурсів, оскільки обумовлений нею запис може містити інформацію про те, які ресурси буде використовувати пакет, наприклад, ту вихідну чергу, в яку він повинен ставитися. Коли мітка переноситься в заголовку ATM або Frame Relay, вона повинна передавати семантику як пересилання, так і резервування ресурсів. Коли мітка переноситься в спеціальному заголовку, інформація про те, які ресурси будуть доступні пакету, може кодуватися як частина цього заголовка, а не переноситися міткою, яка служить в цьому випадку тільки для пересилання. Ще один можливий варіант полягає в спільному використанні для кодування цієї інформації як мітки, так і «не міткової» частини спеціального заголовка. І, зрозуміло, навіть при використанні спеціального заголовка мітка може передавати і семантику пересилання, і семантику резервування ресурсів.

1.6 Структура мітки

Мітка являє собою короткий елемент фіксованої довжини, що використовується для локальної ідентифікації класу еквівалентності пересилання FEC. За значенням мітки пакета в кожному вузлі маршруту, по якому він передається, визначається його приналежність певного FEC.

Структура мітки MPLS представлена на рис. 1.3. Її довжина становить 32 біта (4 байта): 12 бітів - заголовок і 20 бітів - значення мітки. Тема мітки складається з трьох полів: 3-бітового поля Exp, яке може служити для позначення класу обслуговування, S-біта ознаки «дна» стека і 8-бітового поля «час життя» TTL (Time-to-Live).

20-бітовое поле мітки містить значення MPLS-мітки, яке може бути будь-яким числом в діапазоні від 0 до 1048575, за виключним резервних значень (0, 1, 2, 3 та ін.), визначенням використовування яких займається робоча група MPLS в складі комітету IETF.

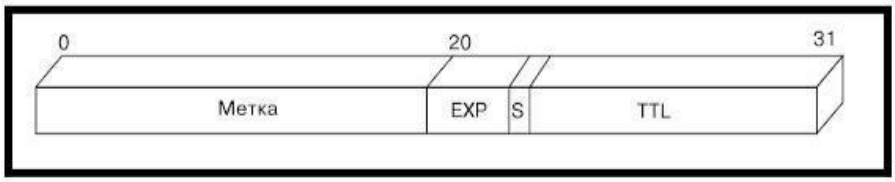


Рисунок 1.3 Структура мітки пакету MPLS

Тепер розглянемо вміст полів заголовка мітки. Поле експериментальних бітів (EXP) містить три біта, які зарезервовані для подальших досліджень та експериментів. В даний час проводиться робота, спрямована на створення узгодженого стандарту використання цих бітів для підтримки диференційованого обслуговування різнотипного трафіку і ідентифікації класу обслуговування. Спочатку поле так і називалося - «Клас обслуговування» (CoS), і ця назва до цих пір зустрічається в літературі. При наданні диференційованих послуг MPLS-мережі це поле може вказувати клас обслуговування, наприклад, аналогічний класам DiffServ. Щоб забезпечити наскрізну якість IP-послуг, на кордоні MPLS-мережі можна скопіювати в поле CoS поле IP-пріоритету з врахуванням того, що поле CoS в MPLS-заголовку містить всього 3 біта, і в ньому може передаватися тільки 3-бітовое поле IP-пріоритету, а 6-бітовое поле коду диференційованої послуги (Differentiated Service Code Point - DSCP) - немає. При необхідності інформація CoS може передаватися і в

вигляді однієї з міток MPLS-стеку, тому що поле мітки має розмір 20 бітів і здатне вмістити як поле IP-пріоритету, так поле DSCP.

Біт S є засобом підтримки ієрархічної структури стека міток MPLS. У заголовку останньої (тобто найглибшої або нижньої) мітки біт S = 1, а у всіх інших мітках в стеці біт S = 0. Детальніше стек міток розглядається в наступному параграфі. Поле часу життя (TTL) в заголовку MPLS працює аналогічних полю TTL в IP-дейтаграмі; це поле є механізмом, що запобігає можливість нескінченної циркуляції пакетів по мережі внаслідок утворення закільцьованих маршрутів. Байт TTL знаходиться в кінці заголовка мітки і, як представлено на рис. 1.3, займає біти 24 - 31. Діапазон значень цього поля становить від 0 до 255. Пояснимо його призначення докладніше.

Як уже згадувалося, протокол MPLS підтримує два способи визначення маршруту, яким прямуватиме пакет. Перший з них аналогічний способу hop-by-hop, використовуваному сьогодні в IP-мережах, і передбачає, що кожен маршрутизатор самостійно вибирає, куди переслати прийнятий ним пакет, так що маршрут, по якому транспортується пакет, виявляється, взагалі кажучи, випадковим. Другий спосіб заснований на тому, що маршрутизатори на шляху проходження пакету діють не самостійно, а відповідно до інструкцій, отриманих від одного з LSR тракту LSP (зазвичай - від верхнього LSR, що дозволяє поєднати процедуру «роздачі» цих інструкцій з процедурою розподілення міток). Таким чином, маршрут проходження пакетів однозначно визначається заздалегідь.

Перший спосіб має ряд переваг, пов'язаних зі порівняною простотою його реалізації, але не виключає феномен «закільцювання» маршрутів, для боротьби з яким і використовується поле TTL. Вхідний LSR поміщає в це поле максимальне число LSR, через які має право пройти пакет, а кожен маршрутизатор, через який пакет проходить, зменшує це число на одиницю. Якщо пакет не призначений вузлу, де він опинився в той момент, коли його поле TTL прийняло нульове значення, цей пакет просто відкидається.

Таким чином, значення TTL, встановлене на межі MPLS-мережі, зменшується в міру проходження пакетом кожного наступного LSR. Під час введення в пакет мітки поле TTL протоколу IP за замовчуванням копіюється в поле TTL MPLS. Це дозволяє утиліті traceroute показувати все проміжні вузли мережі MPLS в тому випадку, коли при досягненні точки призначення пакет проходить домен MPLS. Якщо потрібно, щоб при вході в MPLS-мережу поле TTL протоколу IP не копіювалося в поле TTL MPLS, використовується команда no mpls ip propagate-ttl. У цьому випадку значення поля TTL MPLS встановлюється рівним 255, і службова програма traceroute не вказує проміжні вузли в MPLS-мережі, а відображає весь домен MPLS як один IP-перехід.

1.7 Стек міток MPLS

Специфікація кодування стека міток MPLS визначена документом RFC 3032 «MPLS Label Stack Encoding». Цей документ містить детальну інформацію про мітки MPLS і про те, як вони використовуються з різними мережевими технологіями, а також визначає ключове для технології MPLS поняття - стек міток. Можливість мати в пакеті більше однієї мітки у вигляді стека дозволяє створювати ієрархію міток, що відкриває дорогу багатьом додаткам MPLS.

MPLS може виконувати зі стеком наступні операції: переміщувати мітку в стек, видаляти мітку зі стека і замінювати мітку. Ці операції можуть використовуватися для злиття і розгалуження інформаційних потоків. операція переміщення мітки в стек (push operation) додає нову мітку поверх стека, а операція видалення мітки зі стека (pop operation) видаляє верхню мітку стека.

Функціональні можливості стека MPLS дозволяють об'єднати кілька LSP в один. До стеку міток кожного з цих LSP зверху додається загальна мітка, в результаті чого утворюється агрегований тракт MPLS. У точці закінчення такого тракту він розгалужується на складові його індивідуальні LSP. Так можуть об'єднувати тракти, які мають загальну частину маршруту. Отже, MPLS здатна забезпечувати ієрархічне пересилання, що може стати важливою і затребуваною функціональної можливістю.

При її використанні не потрібно переносити глобальну маршрутну інформацію, що робить мережу MPLS більш стабільною і масштабованою, ніж мережу з традиційною маршрутизацією.

Згідно даним в наступному параграфі правилам інкапсуляції міток, за міткою MPLS в пакеті завжди повинен слідувати заголовок мережевого рівня. Так як MPLS починає роботу з верхнього рівня стека, цей стек використовується за принципом LIFO «останнім прийшов, першим пішов».Приклад чотирьох рівневого стека міток представлений на рис. 1.4. Тут заголовок MPLS №1 був першим заголовком MPLS, переміщеним в пакет, потім в нього були поміщені заголовки №2, №3 і, нарешті, - заголовок №4. Комутація по мітках завжди використовує верхню мітку стека, і мітки видаляються з пакета так, як це визначено вихідним вузлом для кожного LSP, за яким слід пакет. Розглянутий в попередньому параграфі біт S має значення 1 в нижній мітці стека і 0 - у всіх інших мітках. Це дозволяє прив'язувати префікс до кількох міток, іншими словами — до стеку міток (Label Stack). Кожна мітка стека має власні значення поля EXP, S-біта і поля TTL.

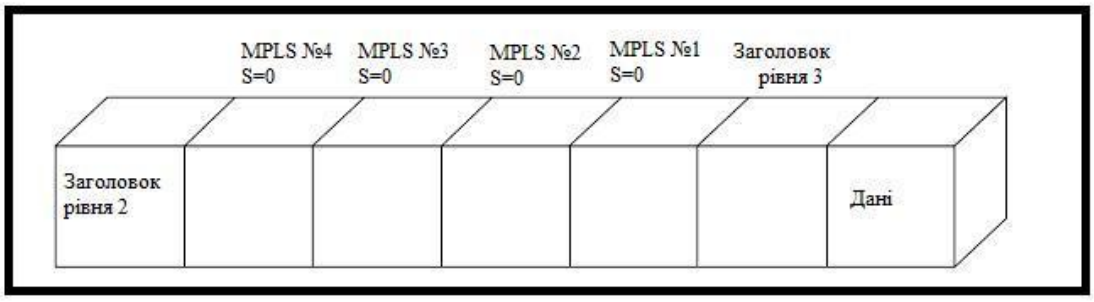


Рисунок 1.4 Стек міток MPLS

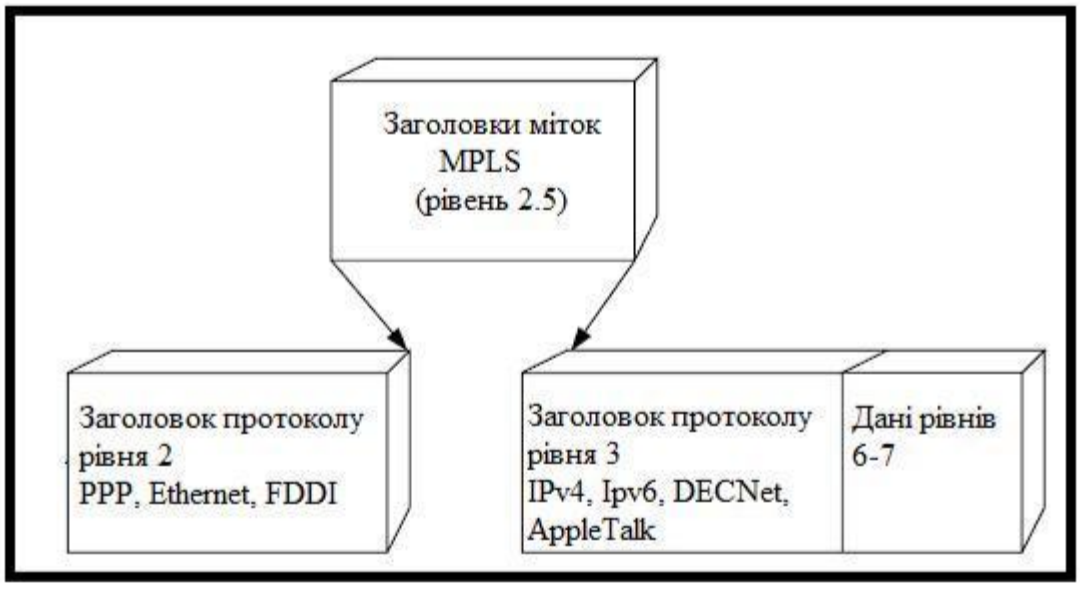
1.8 Інкапсуляція міток

При використанні протоколів комутації на рівні ланки даних, таких як ATM і Frame Relay, верхня MPLS-мітка вписується в поле ідентифікаторів цих протоколів. Далі буде показано, як при використанні ATM для розміщення MPLS-меткі

використовується поле VPI / VCI, а при використанні Frame Relay - поле DLCI. У тих випадках, коли MPLS забезпечує пересилання IP-пакетів мережевого рівня і коли технологія рівня ланки даних не підтримує власне поле міток, MPLS-заголовок повинен інкапсулюватися між заголовками рівня ланки даних і мережевого рівня.

Механізм інкапсуляції переносить один або більше протоколів верхніх рівнів всередині корисного навантаження дейтаграми інкапсульованого протоколу. По суті, вводиться новий заголовок, який робить з інкапсульованого заголовка і поля даних нове поле даних. Загальна модель інкапсуляції представлена на рис. 1.5, де мається на увазі, що інкапсуляція MPLS може бути використана з будь-якою технологією рівня 2. Позначка MPLS може бути поміщена в існуючий формат заголовка рівня 2, як у випадку ATM або FR, або вписана в спеціальний заголовок MPLS, як у випадку Ethernet або PPP. У всіх випадках будь-які додаткові мітки знаходяться між верхньою міткою стека і IP-заголовком рівня 3.

Показаний на рис. 1.5 заголовок MPLS часто називають shim header (заголовком-клином), підкреслюючи в метафоричній формі той факт, що цей заголовок «рівня 2.5» вклинюється в пакеті між заголовками рівня ланки даних і мережевого рівня.



Риунок 1.5 Інкапсуляція міток

Отже, однією з найсильніших сторін технології MPLS (і тому відображеної в її назві) є те, що вона може використовуватися спільно з різними протоколами рівня 2. Серед цих протоколів - ATM, Frame Relay, PPP і Ethernet, FDDI і інші, передбачені документами по MPLS. Покажемо, як мітка може вписуватися в заголовок рівня ланки даних (VCI / VPI для мережі ATM, DLCI для мережі Frame Relay і т.п.) або «вставлятися» між заголовками рівня ланки даних і мережевого рівня. З самого початку робоча група IETF MPLS вирішила, що у всіх випадках, коли це можливо, MPLS повинна використовувати наявні формати. З цієї причини інформація мітки MPLS може передаватися в пакеті декількома різними методами:

- як частина заголовка другого рівня ATM, коли інформація мітки передається в ідентифікаторах віртуального каналу VCI і віртуального тракту VPI, що показано на рис. й.6;

- як частина кадру AAL5 рівня адаптації ATM (ATM Adaptation Layer 5) перед сегментацією і складанням SAR (Segmentation and Reassembly), що виконується в ATM-оточенні в разі, коли ця інформація містить дані про стек міток (декілька полів MPLS-міток);

- як частина заголовка другого рівня Frame Relay, коли інформація мітки передається в ідентифікаторах DLCI (Data Link Connection Identifier); як нова 4-байтова мітка, звана клином або прокладкою (shim), яка вставляється між заголовками другого і третього рівнів;

Отже, мітка може бути поміщена в пакет різними способами - вписуватися в спеціальний заголовок, що поміщається або між заголовками рівня 2 і рівня 3, або у вільний і доступне поле заголовка одного з цих двох рівнів, якщо, звичайно, таке є. Очевидно, що питання про те, куди слід поміщати заголовок, що містить мітку, повинен узгоджуватися між об'єктами, що її використовують.

Незважаючи на те, що в документі по MPLS специфікована робота MPLS рядом протоколів мережевого рівня (а теоретично, оскільки застосовується спеціальний заголовок MPLS, - з будь-яким мережевим протоколом), спочатку вона стала

використовуватися з поточною версією протоколу IP, тобто з протоколом IPv4. Продовжується робота над протоколом IPv6, і цей протокол повільно, але вірно реалізується на практиці. Як і в разі IPv4, IPv6 поміщає заголовок MPLS перед заголовком мережевого рівня і, в залежності від використовуваного протоколу рівня 2, або у вигляді спеціального заголовка, або всередині заголовка рівня ланки даних. Значна частина поточних робіт в області MPLS сконцентрована на технологіях більш низьких рівнів і на використанні оптичних засобів передачі пакетів по мережі IP. Так що не буде помилкою сказати, що IP стане переважаючим протоколом мережевого рівня, що використовуються з MPLS.

1.9 Таблиці пересилання

Раніше зазначалося, що коли пакет MPLS потрапляє в маршрутизатор комутації по мітках LSR, цей маршрутизатор переглядає наявну у нього таблицю з так званою інформаційною базою міток LIB (Label Information Base) для того, щоб прийняти рішення по подальшій обробці пакету. Цю інформаційну базу іноді називають також Next Hop Label Forwarding Entry (NHLFE), і відповідно до RFC 3031 в неї зазвичай входить наступна інформація:

- операція, яку треба зробити зі стеком міток пакета (замінити верхню мітку стека, видалити верхню мітку, помістити поверх стека нову мітку);
- наступний маршрутизатор в LSP, причому «наступним» може бути той же самий LSR;

Вхідна мітка	Перший підзапис	Другий підзапис
Значення вхідної мітки	Вихідна мітка	Вихідна мітка
	Вихідний інтерфейс	Вихідний інтерфейс
	Адреса наступного LSR	Адреса наступного LSR

Рисунок 1.6 Приклад таблиці пересилання

використовувана при передачі пакета інкапсуляція на рівні ланки даних;

- спосіб кодування стека міток при передачі пакета- інша інформація, що відноситься до пересилання пакета.

Таблиця, що містить цю інформацію, яку веде кожен LSR, як, втім, і будь-які інші таблиці, представляє собою послідовність записів. Кожен запис таблиці пересилання LSR складається з вхідної мітки і одного або більше підзаписів, причому кожен підзапис містить значення для вихідної мітки, ідентифікатор вихідного інтерфейсу і адреса наступного маршрутизатора в LSP. Приклад простої таблиці пересилання LIB представлений на рис. 1.6.

Різні підзаписи всередині однієї записи можуть мати або однакові, або різні значення вихідних міток. Більше одного підзапису буває потрібно для підтримки під LGPL пакета, коли пакет, який надійшов до одного входить інтерфейс, повинен потім розсилатися через кілька вихідних інтерфейсів. Звернення до таблиці записів проводиться по значенню вхідної мітки, тобто по вхідній мітці k відбувається звернення до k-того запису в таблиці.

Як уже зазначалося, на додаток до інформації, яка показує, куди повинен пересилатися пакет - наступну ділянку маршруту або наступний маршрутизатор, - запис в таблиці може також містити інформацію, яка вказує, які ресурси має можливість використовувати пакет, наприклад, певну вихідну чергу.

LSR може підтримувати або одну загальну таблицю, або різні таблиці для кожного зі своїх інтерфейсів. У першому варіанті обробка пакета визначається виключно міткою, яку переносять в пакеті. У другому варіанті обробка пакету визначається не тільки міткою, а й інтерфейсом, до якого надійшов пакет. LSR може використовувати або перший варіант, або другий варіант, або їх поєднання.

1.10 Прив'язка «мітка-FEC»

На рис. 1.6 видно, що кожен запис в таблиці пересилання, яку веде LSR, містить одну що входить мітку і одну або більше вихідних міток. Відповідно до цими двома типами міток забезпечується два типи прив'язки міток до FEC. Перший тип - мітка для прив'язки вибирається і призначається в LSR локально. Таку прив'язку ми будемо називати локальною. Другий тип - LSR отримує від деякого іншого LSR інформацію про прив'язку мітки, яка відповідає прив'язці, створеної на цьому другому LSR. Таку прив'язку ми будемо називати віддаленою.

Засоби управління комутацією по мітках використовують для заповнення таблиць пересилання як локальну, так і віддалену прив'язку міток до FEC. Це може робитися в двох варіантах: upstream і downstream. Перший - це коли мітки локальної прив'язки використовуються як вхідні мітки, а мітки з віддаленої прив'язки використовуються як вихідні мітки. Другий варіант - прямо про-протилежний, тобто мітки з локальної прив'язки використовуються як вихідні мітки, а мітки з віддаленої прив'язки - як вхідні мітки. Розглянемо кожен з цих варіантів.

Перший варіант називається прив'язкою мітки до FEC «знизу» (downstream label binding), тому що в цьому випадку прив'язка перенесеної пакетом мітки до того FEC, якому належить цей пакет, створюється нижчестоящим LSR, тобто LSR, розташованим ближче до адресата пакета, ніж LSR, який поміщає мітку в пакет. Відзначимо, що при прив'язці «знизу» пакети, які переносять певну мітку, передаються в напрямку, протилежному напрямку передачі інформації про прив'язку цієї мітки до FEC.

Другий варіант називається прив'язкою мітки до FEC «зверху» (upstream label binding), тому що в цьому випадку прив'язка перенесеною пакетом мітки до того FEC, якому належить цей пакет, створюється тим же LSR, який поміщає мітку в пакет; тобто творець прив'язки розташований «вище» (ближче до відправника пакета), ніж LSR, до якого пересилається цей пакет. Відзначимо, що при прив'язці «зверху» пакети,

які переносять певну мітку, передаються в тому ж напрямку, що і інформація про прив'язку цієї мітки до FEC.

LSR обслуговує також пул «вільних» міток (тобто міток без прив'язки). При початковій установці LSR пул містить всі мітки, які може використовувати LSR для їх локальної прив'язки до FEC. Саме ємність цього пулу і визначає, в кінцевому рахунку, скільки пар «мітка-FEC» може одночасно підтримувати LSR. Коли маршрутизатор створює нову локальну прив'язку, він бере мітку з пулу; коли маршрутизатор знищує раніше створену прив'язку, він повертає мітку, пов'язану з цією прив'язкою, назад в пул.

Згадаємо, що LSR може вести або одну загальну таблицю пере-посиланням, або кілька таблиць - по одній на кожен інтерфейс. Коли маршрутизатор веде загальну таблицю пересилання, він має один пул міток.

Коли LSR веде кілька таблиць, він має окремий пул міток для кожного інтерфейсу.

1.11 Режими операцій з мітками

Розглянемо режими трьох базових операцій, які представляють, по суті, основні принципи механізму комутації по міткам: призначення міток, розподіл міток і збереження міток.

Призначення міток. Коли FEC створюється шляхом аналізу адресних префіксів, які розподіляються протоколом внутрішньої маршрутизації і використовуються для створення трактів LSP по ділянках, можливі два режими призначення міток:

- незалежне призначення (тобто незалежне створення трактів LSP);
- впорядковане призначення (тобто впорядковане створення трактів LSP).

Незалежне призначення. При незалежному призначення міток кожен LSR сам, незалежно від інших подій, приймає рішення про прив'язку мітки до виявленого FEC і про повідомлення вищого LSR про цю прив'язку. Така ситуація аналогічна

традиційної маршрутизації, що виконується в звичайних IP-мережах, коли виявляються нові маршрути.

Впорядковане призначення. Цей спосіб призначення міток має більше обмежень, ніж попередній, в тому сенсі, що прив'язка мітки до певного FEC відбувається тільки тоді, коли LSR або виступає в ролі вихідного вузла для цього FEC, або вже отримав інформацію про прив'язку «мітка-FEC» від нижчестоящого маршрутизатора.

Розподіл міток. В технології MPLS можуть використовуватися два режими розподілу міток: нижчестоящим LSR за запитом вищого або нижчестоящим LSR за власною ініціативою. Режим розподілу нижчестоящим за запитом вищого користується для створення трактів LSP по ділянках. Він дозволяє вищестоящому LSR в явному вигляді запитувати прив'язку мітки до визначеному FEC біля сусіднього з ним нижчестоящого LSR. Режим розподілення міток нижчим за власною ініціативою використовується тоді, коли нижчому LSR потрібно «роздати» мітки вищим LSR, хоча ті не замовляли у нього цього в явному вигляді.

Обидва режими розподілу міток обговорювалися в попередньому параграфі. Додамо лише, що архітектура MPLS дозволяє застосовувати в одній системі один або обидва режими в залежності від деталей реалізації, таких як характеристики інтерфейсів і наявні ресурси мережі MPLS. Однак у кожній сукупності суміжних вузлів, які беруть участь в розподілі міток, вищестоящий LSR і нижчий LSR повинні узгодити між собою і використовувати для створення трактів LSP один і той же режим розподілення. Це пов'язано з тим, що архітектура MPLS не припускає якогось єдиного протоколу розподілу міток.

В одній і тій же мережі MPLS можуть використовуватися: спеціальний протокол розподілу міток Label Distribution Protocol (LDP), протокол сигналізації RSVP, а також розширення можливостей протоколів маршрутизації, наприклад, протоколу міждоменної маршрутизації Border Gateway Protocol (BGP).

Режими збереження міток. Ще однією характеристикою використання міток є режим їх збереження. Коли вищестоящий LSR отримує мітку від нижчестоящого LSR, який в даний момент не є для нього суміжним з точки зору даного FEC, він може прийняти щодо цієї мітки рішення: або використовувати її (тобто «зберегти» у себе її прив'язку до FEC), або відкинути.

У MPLS використовуються два основні режими збереження назв: ліберальний режим, консервативний режим.

При ліберальному режимі вищестоящий LSR зберігає у себе будь-яку прив'язку до FEC міток, які він отримав від НЕ суміжних нищестоящих LSR (тобто мітки прийшли до нього транзитом). При консервативному режимі вищестоящий LSR відмовляється від таких міток, тобто відкидає їх. Перевагою ліберального режиму являється те, що якщо нижчий LSR стане з точки зору даного FEC суміжних маршрутизатором, змінювати прив'язку «мітка-FEC» не знадобиться. Це дозволяє набагато швидше реагувати на зміну маршрутів.

Звичайно, якщо прив'язка мітки до FEC вже була відкинута, то її доведеться призначити повторно до того, як можна буде використовувати LSP. При консервативному режимі потрібно менше ресурсів, але реакція на зміни в наступних ділянках маршруту відбувається набагато повільніше.

1.12 Фіксовані значення мітки

Наведемо кілька фіксованих (резервованих) значень міток:

0 - «IPv4 Explicit NULL Label». Ця мітка повинна знаходитися на дні стека міток. Вона вказує, що стек повинен бути виділений з пакета, і подальша маршрутизація цього пакета повинна ґрунтуватися на заголовку IPv4.

1 - «Router Alert Label». Ця мітка може знаходитися в будь-якому місці стека міток за винятком його дна. Коли приходить пакет, що містить таку мітку нагорі стека, він доставляється місцевим програмного модулю для обробки. Подальша маршрутизація буде проводитися або по нищележащій мітці, або на основі інформації,

що міститься в заголовку протоколу мережевого рівня або інкапсульованому в нього протоколі. При пересиланні пакета мітка Router Alert Label повинна бути знову поміщена наверх стека міток. Використання цієї мітки регламентовано в RFC 2113 і аналогічно використанням опції Router Alert в IP-пакетах.

2 - «IPv6 Explicit NULL Label». Ця мітка повинна знаходитися на дні стека міток. Вона вказує, що стек повинен бути виділений з пакета, і подальша маршрутизація пакету повинна засновуватися на заголовку Ipv6.

3 - «Implicit NULL Label». Ця мітку LSR може привласнювати і розповсюджувати, але пакети їй ніколи не позначаються. Коли LSR, відповідно LIB, повинен замінити (swap) мітку, що знаходиться вгорі стека, а нова мітка є Implicit NULL Label, то замість заміни, LSR видаляє (pop) стек міток.

4 - 15 - Значення зарезервовані для подальшого використання.

Резервування значень необхідно не тільки для службових повідомлень. У зв'язку з активним розвитком технології MPLS воно видається дуже важливим і з інших точок зору.

Висновки

В розділі було розглянуто основні положення та поняття побудови мереж на основі технології MPLS, її складові та елементи. З розглянутого матеріалу стає зрозуміло, що технологія являє собою доволі прогресивне рішення, яке дозволяє краще розподіляти ресурси мережі, гарантувати забезпечення високої якості роботи та управління трафіком.

2. Інжиніринг трафіку в мережах MPLS, огляд та порівняльна характеристика протоколів RSVP-TE та CR-LDP

3.1 Концепція інжинірингу трафіку в MPLS

Інжиніринг трафіку - це моніторинг та моделювання потоків трафіку, а також управління трафіком з тим, щоб забезпечити потрібну якість його обслуговування шляхом раціонального використання мережевих ресурсів за рахунок збалансованого їх завантаження. У вітчизняній літературі сукупність механізмів, що забезпечують виконання перерахованих функцій, називається також перерозподілом трафіку, конструюванням трафіку, оптимізацією трафіку, проектуванням трафіку, керуванням трафіком. Більш точним є назва управління різнотипним трафіком, тому що воно підкреслює зв'язок розглянутих тут механізмів із завданням забезпечити різну якість обслуговування (QoS) трафіку різних типів, але в роботі використовується найбільш розповсюджена пряма калька з англійського еквівалента - Traffic Engineering (TE).

Так чи інакше, можливості інжинірингу трафіку - головна або, принаймні, одна з головних причин, за якими технологія MPLS реалізується в сьогоднішніх мережах зв'язку. Вони дозволяють зняти обмеження, властиві протоколам маршрутизації внутрішнього шлюзу IGP, як дистанційно векторним (RIP), так і на основі стану каналів (OSPF і ISIS). Ці протоколи направляють трафік від відправника до адресата за найкоротшим маршрутом, заздалегідь обраному в певній метриці. Для протоколу RIP ця метрика являє собою просто число пересилань (hops) між маршрутизаторами. Протоколи маршрутизації на основі параметрів стану каналів - OSPF і ISIS - використовують більш «просунуті» метрики: граничну пропускну здатність каналів, що вноситься каналом затримку при передачі пакета і т.п. Однак і вони ігнорують поточну ситуацію з перевантаженнями в мережі і тип трафіку, який несуть маршрутизовані пакети. Це ж справедливо і по відношенню до протоколів зовнішнього шлюзу EGP, зокрема, протоколу BGP4. Нагадаємо, що протокол BGP4 використовується також і в середині автономних систем, але і в цій якості він, як і інші

традиційні протоколи маршрутізації IGP і EGP, непридатний для інжинірингу трафіку в силу залежності використовуваних їм метрик від топології мережі, а не від реальної обстановки, пов'язаної з проходженням і обробкою трафіку. При такому підході на обраному найкоротшому маршруті може виникнути перевантаження, що призведе до затримки або втрати пакетів, незважаючи на наявність в інших альтернативних LSP не використовуваної пропускної здатності. Наприклад, протокол OSPF може навіть збільшити перевантаження, оскільки він прагне направити трафік по перевантаженому найкоротшому маршруті і тоді, коли інший, нехай не самий короткий, але цілком прийнятний маршрут завантажений менше.

Класичний приклад, який ілюструє цю проблему маршрутизації за принципом SPF (Shortest Path First), демонструє елемент мережевої топології, що має через зовнішню схожість фольклорне найменування «риба». На рис. 2.1 зображений в такому вигляді все той же фрагмент MPLS - мережі з 7 маршрутизаторів LSR1 - LSR7. Весь трафік від маршрутизатора LSR2 до LSR5, згідно з алгоритмом OSPF йде через маршрутизатор LSR6. Таким чином, маршрут LSR2 - LSR6 - LSR5 перевантажений, а ресурс маршруту LSR2 - LSR3 - LSR4 - LSR5 використовується неефективно.

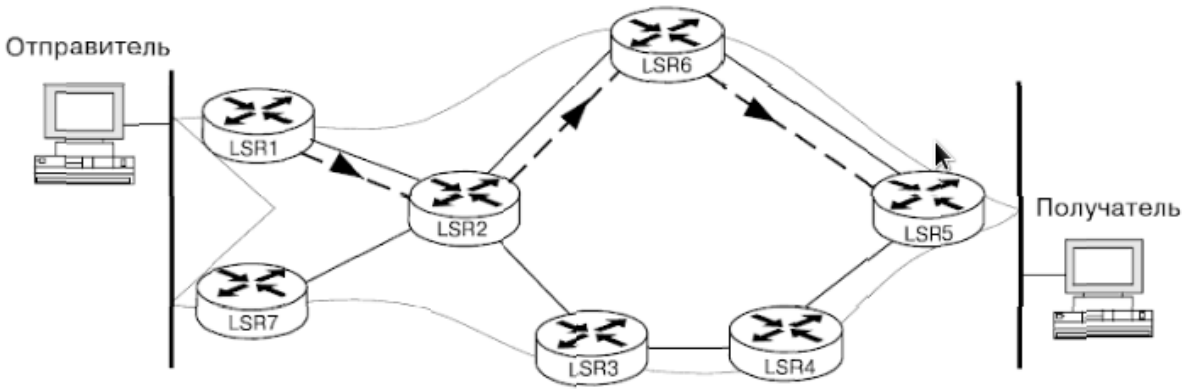


Рисунок 2.1 Традиційний розподіл трафіку

Застосування механізмів TE дозволяє вирішити цю проблему, вказавши два різних шляхи від LSR1 до LSR5, як це зображено на рис. 2.2.

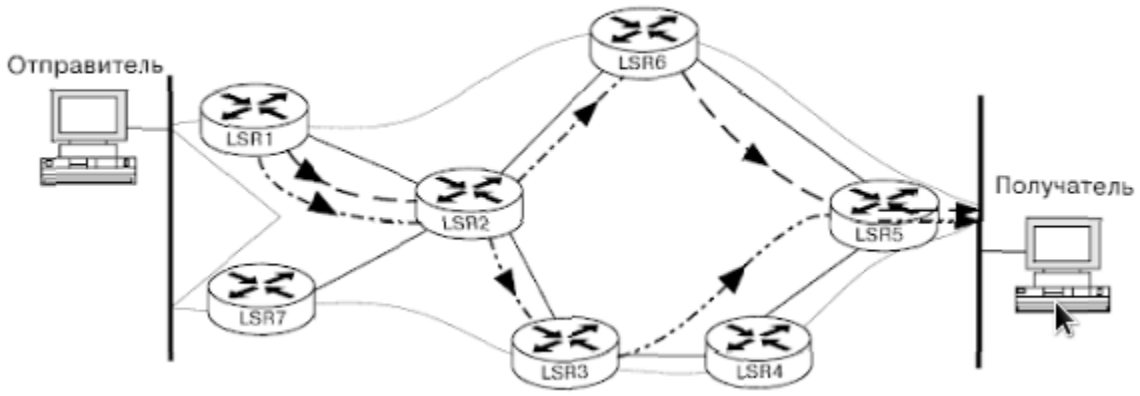


Рисунок 2.2 Рівномірний розподіл трафіку

Порівняння рис. 3.1 і 3.2 показує, що ТЕ дозволяє краще використовувати мережеві ресурси за рахунок переведення частини трафіку з більш завантаженої на менш завантажену ділянку мережі. При цьому не тільки краще використовується доступна смуга пропускання, але і досягається більш висока якість обслуговування трафіку, оскільки зменшується ймовірність перевантаження в мережі. Крім того, для послуг, які вимагають виконання заданих норм якості обслуговування QoS, наприклад, заданого коефіцієнта втрат пакетів і / або затримки / джиттера, інжиніринг трафіку дозволяє забезпечувати належне значення QoS шляхом призначення явно визначених маршрутів. Як експлуатаційний інструмент, інжиніринг трафіку регулярно оптимізує використання мережевих ресурсів при змінах розподілу навантаження в мережі.

Ключові характеристики, пов'язані з управлінням трафікоком, можуть бути орієнтовані або на трафік, або на ресурси. Завдання ТЕ включають в себе аспекти поліпшення показників QoS інформаційних потоків, а центральною функцією ТЕ є оптимальне управління пропускною здатністю.

Оптимізація робочих характеристик мережі є фундаментальною проблемою управління. У моделі процесу ТЕ трафік інженер (Traffic Engineer) діє як контролер в системі з адаптивним зворотним зв'язком. Ця система включає в себе набір взаємопов’язаних мережевих елементів, систему моніторингу стану мережі та набір

засобів управління конфігурацією. Трафік інженер формулює політику управління, контролює стан мережі, використовуючи систему моніторингу, визначає характеристики трафіку і робить керуючі дії, щоб перевести мережу в стан, що узгоджується з політикою управління. Це може бути виконано за допомогою операцій, вироблених в порядку відгуку на поточний стан мережі, або превентивно, на основі аналізу тенденції змін стану мережі та їх прогнозування з тим, щоб запобігти виникненню небажаних станів. В ідеалі керуючі дії повинні включати в себе:

- модифікацію параметрів управління трафіком;
- модифікацію параметрів, пов'язаних з маршрутизацією;
- модифікацію атрибутів і констант, пов'язаних з ресурсами;

Трафік інженер тут означає не посада співробітника (хоча можливо і таке), а деякий абстрактний керуючий об'єкт. Участь людини в процесі управління трафіком має бути мінімізована настільки, наскільки це можливо, що забезпечується автоматизацією перерахованих вище операцій.

Зауважимо, що інжиніринг трафіку - аж ніяк не пов'язаний з MPLS винахід. Завдання управління трафіком вирішувалися в зв'язку завжди, правда, підходи були принципово різноманітні. Розрахунки, що вироблялися при проектуванні АТС на основі теорії телетрафіка, як і вимір характеристик навантаження та обслуговування в процесі експлуатації з подальшою ре-конфігурацією пучків сполучних ліній, теж можна віднести до управління трафіком. З появою пакетних мереж і мультимедійного трафіку завдання і методи управління трафіком зазнали значних змін.

В цих нових умовах застосування теорії телетрафіка можна віднести до проектування і до експлуатаційному управлінню мережами зв'язку. Час, що витрачається на вирішення цих завдань (час реакції), може вимірюватися в тижнях, днях і годинах, відповідно, як це зображено на рис. 3.3.

Для розглянутих в цьому розділі методів інжинірингу трафіку MPLS час реакції вимірюється в секундах, хвилинах і годинах. В мережевому обладнанні можуть застосовуватися ще більш швидкодіючі методи боротьби з перевантаженнями, час

реакції яких вимірюється в секундах і навіть в мілісекундах. Перевантаження, пов'язані з неефективним розміщенням ресурсів, може бути зменшені вибором належної політики балансування навантаження в різних пристроях мережі. Завданням такої політики є також мінімізація перевантаження ресурсу. Коли перевантаження мінімізовані, втрати пакетів і затримка доставки зменшуються, а сукупна пропускна здатність зростає.

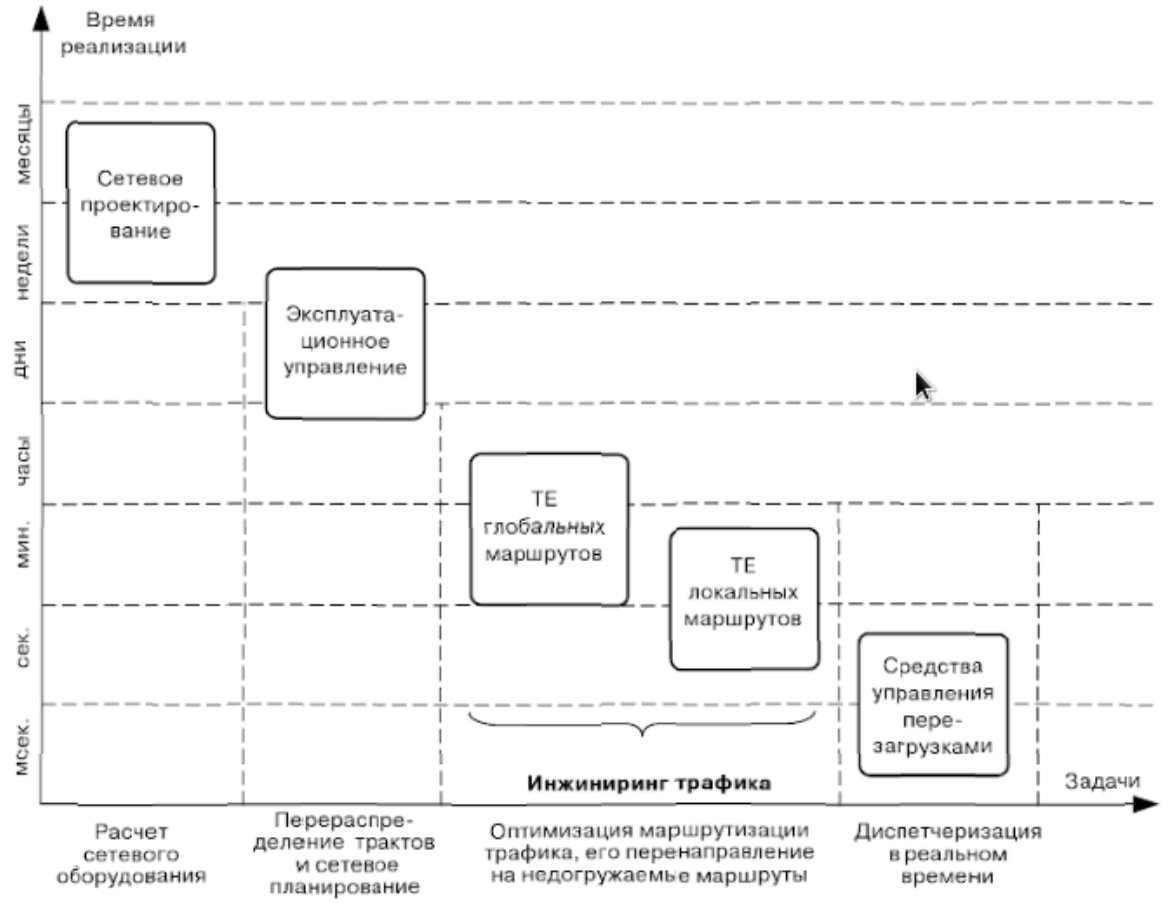


Рисунок 2.3 Інжиніринг трафіку в мережах зв’язку

Слід сказати про деякі умовності рис. 2.3, головним задумом якого є ілюстрація підходів до аналізу трафіку в мережах зв'язку і місця розглянутого тут інжинірингу трафіку MPLS. В якості компенсації цієї умовності наведемо на закінчення параграфа більш суворе визначення ТЕ згідно RFC 2702: “інжиніринг трафіку включає в себе

технологію та научні принципи вимірювання, моделювання та опису трафіку, управління трафіком і використання цих знань і техніки для отримання певних робочих характеристик мережі”.

2.2 Протоколи сигналізації

Хоча методи TE з'явилися раніше технології MPLS, саме ця технологія дуже добре підходить для управління трафіком і може надати більшу частину функцій TE по відносно низькій (в порівнянні з конкуруючими рішеннями) ціні. Не менш важливо, що в MPLS можна автоматизувати функції управління трафіком. Для всього цього потрібно тільки одне - протоколи сигналізації MPLS повинні вміти переносити інформацію, яка необхідна для роботи механізмів управління трафіком, що знаходяться на прикладному рівні, а також «прокладати» тракти LSP по явно заданих маршрутах. При цьому в MPLS є можливість досягти додаткової гнучкості, тому що маршрут можна задати не тільки строго, але і не строго, тобто група вузлів може бути задана як «абстрактний вузол», всередині якого існує відома свобода вибору маршрута.

Перед IETF, точніше перед її робочою групою MPLS, встала задача вибрати такий протокол. Кращими виявилися два варіанти, RSVP - TE і CR-LDP. У першому варіанті протоколу RSVP необхідно робити в мережі MPLS те, що він вже виконував в мережах IP, а саме - обробляти інформацію, пов'язану з QoS, і резервувати ресурси. Залишається лише додати можливість розподілу міток. У другому варіанті теж не представлялося складним додати до вже використовуваному для розподілу міток в MPLS протоколу LDP кілька нових об'єктів для перенесення інформації про QoS. У підсумку, робоча група так і не змогла прийти до єдиного рішення з цього питання, і обидва протоколи були розвинені до рівня proposed standard.

Як це часто трапляється останнім часом, практично відразу ж слідом за робочою групою IETF завдання вибору протоколу сигналізації для інжинірингу трафіку в MPLS встала також перед виробниками мережевого устаткування і програмного

забезпечення, а потім - і перед операторами. Для виробників це вилилося в необхідність створювати в своєму обладнанні засоби підтримки обох протоколів і випускати різні версії, що підтримують або той, або інший протокол. Операторам довелося ще складніше - перед ними виникла проблема вибору, який з протоколів використовувати в своїй мережі MPLS, а також споконвічна проблема взаємодії між мережевими областями, які використовують різні протоколи, наприклад, при злитті мереж або при покупці мережі альтернативного оператора.

Спочатку протоколи мали ряд помітних функціональних і технічних відмінностей, базуючись на яких, можна було робити вибір на користь того чи іншого конкурента. Згодом протоколи еволюціонували і розвивалися, згладжуючи свої недоліки і зберігаючи гідності. Таким чином, вибір між ними все ускладнювався. До порівняльного аналізу цих двох протоколів ми повернемося в кінці глави, після того як розглянемо принципи і моделі TE.

2.3 Атрибути потоків трафіку і мережевих ресурсів

Нагадаємо, що тракт LSP - це логічне з'єднання, яке створюється в MPLS мережі для перенесення через неї пакетів, що належать одному FEC. Елементами такого з'єднання є маршрутизатори LSR і зв'язуючі їх ланки. При цьому не можна забувати, що ланцюжки <LSR i - ланка - LSR i + 1 - ланка - ..... LSR n1 - ланка - lSR n >, за якими проходять різні пакети одного і того ж FEC (тобто маршрути, які використовуються в LSP), в загальному випадку, можуть бути разнимі. В області інжинірингу трафіку стосовно MPLS використовується термін traffic trunk, яким позначають об'єднання потоків трафіку, що належать одному класу і проходять по одному LSP. Іншими словами, traffic trunk - це об'єднаний потік трафіку, віднесеного до одного FEC. При інжинірингу трафіку в MPLS вводиться поняття наведений MPLS граф.

Цей граф утворюють набір LSR, що представляють його вузли, набір ланок, які представляють його ребра, і набір використовуваних в LSP маршрутів, які

представляють шляху наведеного графа. Керування трафіком в MPLS передбачає наявність таких функціональних засобів і можливостей:

- набір атрибутів, які пов'язані з об'єднаними потоками трафіку;
- набір атрибутів, які пов'язані з ресурсами; ці атрибути обмежують можливості вибору маршрутів для LSP і можуть розглядатися як топологічні обмеження;
- маршрутизація на основі обмежень, яка використовується для вибору маршруту відповідно до заданих наборами параметрів;

Атрибути, пов'язані з потоками трафіку і з ресурсами, а також параметри, пов'язані з маршрутизацією, в сукупності представляють собою набір керуючих змінних, які можуть бути модифіковані в результаті дій адміністраторів або автоматичних агентів (трафік інженерів) управління мережею. У робочій мережі завжди бажано, щоб ці атрибути можна було міняти динамічно в реальному часі. Розглянемо атрибути об'єднаних потоків трафіку і атрибути мережевих ресурсів.

2.4 Атрибути об'єднаних потоків трафіку

Атрибут об'єднаного потоку трафіку описує характеристики цього потоку. Значення атрибутів можуть бути явно привласнені потокам адміністратором або задані неявно базовими протоколами, коли пакети класифікуються і упорядковано відповідно до FEC при вході в домен MPLS. Наведемо основні з цих атрибутів, специ-ваних IETF:

Атрибути параметрів трафіку можуть використовуватися при зборі даних про рух трафіку (або, точніше, про FEC), які належить транспортувати через тракт LSP. Параметри трафіку визначають вимоги до ресурсів цього LSP.

Атрибути управління і вибору маршрутів визначають правила вибору маршрутів для LSP, а також правила роботи з маршрутами, які вже існують. Вони включають в себе:

- Адміністративно специфіковані маршрути, які конфігуруються оператором. Такий маршрут може бути визначений повністю або частково і бути або обов'язковим для використання, або лише кращим;
- Ієрархія переваги для набору маршрутів адміністративно специфікує ієрархію в наборі можливих маршрутів для даного LSP;
- Атрибути Resource Class Affinity використовуються для специфікації класу ресурсів, які слід явно включити в LSP або виключити з нього. Це атрибути політики, які можуть використовуватися з метою введення додаткових обмежень на маршрути для LSP;
- Атрибут адаптивності є двійковою змінною, яка визначить, як повинен реагувати тракт на зміну стану мережі: дозволити або заборонити адаптивну оптимізацію;
- Розподіл навантаження в паралельних ланках (трактах). Коли об'єднаний трафік між вузлами такий, що одна ланка (один тракт) не зможе його пропустити, і єдиним рішенням є розподіл трафіку на кілька потоків, то ці атрибути вказують частки трафіку, що проходить через кожне (кожен) з паралельних ланок (трактів);

Атрибут пріоритету визначає відносну важливість об'єднання потоку трафіку. Пріоритети використовуються в разі відмов для того, щоб визначити порядок, в якому вибираються з наявного списку маршрути для відповідних LSP, а також в реалізаціях, що допускають пріоритетне обслуговування.

Атрибут Preemption визначає, чи може потік трафіку замістити інший потік в даному тракті, і задає умови пріоритетного заміщення:

- preemptor enabled - може заміщати;
- nonpreemptor - не може заміщати;
- preemptable - допускає заміщення;
- onpreemptable - не допускає заміщення;

Потік трафіку, який допускає заміщення, може бути заданим іншим потоком, які мають більш високий пріоритет і атрибут Preemption зі значенням preemptor enabled.

Атрибут стійкості (Resilience) визначає поведінку ланки (тракту) в разі виникнення помилок. Базовий атрибут Resilience вказує процедуру відновлення, яка повинна бути запущена для даної ланки (тракту) при виникненні відмови. Розширений атрибут Resilience може використовуватися для детальної специфікації дій в разі відмови.

Атрибут Policing визначає дії, які слід прийняти, коли ланка / тракт стає не повноцінним, тобто коли якісь його параметри виходять за допустимі межі. Атрибут Policing може вказувати, чи треба лімітувати ланка / тракт по по-лося пропускання, позначити його або просто пересилати його трафік без якихось дій.

3.5 Атрибути мережевих ресурсів

Атрибути ресурсів мережі входять в параметри топології і слугують для того, щоб визначити для маршрутизації потоків трафіку обмеження, що враховують характеристики заданих ресурсів.

Maximum Allocation Multiplier (MAM) є адміністративно заданим атрибутом, який визначає частку ресурсу, доступному ланці (тракту). Цей атрибут використовується, в основному, для розподілу смуги пропускання. Однак він може бути застосований також для резервування ресурсів LSR.

Атрибути Resource Class також присвоюються адміністративно і вводять поняття *клас ресурсу*. Атрибути класу ресурсу можуть розглядатися як приписані ресурсів *кольори*, такі, що набір ресурсів з одним кольором належить одному класу. Ці атрибути можуть використовуватися для реалізації різних варіантів політики.

3.6 Маршрутизація на основі обмежень

Маршрутизація на основі обмежень (constraint - based routing) використовує в якості вхідних даних розглянуті вище атрибути, пов'язані з потоками трафіку, атрибути, пов'язані з ресурсами, а також іншу інформацію, що характеризує поточну топологію мережі, і дає можливість резервувати за запитом ресурси для управління трафіком. При цьому вона може співіснувати з наявними IGP протоколами маршрутизації, що працюють за принципом
hop - by - hop.

Базуючись на названій інформації, процес маршрутизації на основі обмежень автоматично обчислює маршрут для кожного потоку, що виходить від певного вузла. Результат обчислень являє собою специфікацію маршруту, який задовольняє вимогам, записаним в атрибутах потоку трафіку. Обмеження формулюються на основі відомостей про доступність ресурсів, адміністративної політики і топологічної інформації.

Маршрутизація на основі обмежень призначена для того, щоб максимально скоротити обсяг робіт, пов'язаних з ручною конфігурацією і ступінь участі адміністратора в реалізації політики управління трафіком.

Відомо, що для більшості реальних ситуацій проблему маршрутизації на основі обмежень важко вирішити. Однак на практиці для знаходження прийнятного маршруту може використовуватися наступний простий метод. Спочатку відкидаються ресурси, які не задовольняють вимогам атрибутів потоку трафіку. Потім для графу зв'язків, який залишився, запускається алгоритм пошуку найкоротшого шляху. Оптимізація зазвичай зводиться до мінімізації перевантаження. Але в разі, коли потрібно маршрутизувати пакети кількох потоків трафіку, тобто пакети кількох FEC, запропонований алгоритм не завжди дозволяє знайти рішення, навіть якщо таке існує. При реалізації маршрутизації на основі обмежень в мережевих пристроях необхідна наявність:

- механізмів обміну інформацією про топологічний статус (даними про доступність ресурсів, інформацією про стан зв'язку, інформацією про атрибути ресурсів);
- механізмів роботи з інформацією про топологічний статус;
- засобів взаємодії між процесами маршрутизації на основі обмежень і процесами традиційних IGP;
- механізмів, що забезпечують адаптивність ланок і трактів;
- механізмів, що забезпечують стійкість і живучість трактів;

3.7 Механізми TE в MPLS

Резюмуючи наведені в попередніх параграфах базові відомості про TE, можна сказати, що інжиніринг трафіку в MPLS заснований на управлінні наборами атрибутів, значення яких враховуються при виборі маршрутів для створюваних в MPLS мережі LSP і LSP тунелів. Тепер спробуємо описати загальну картину роботи програми TE в MPLS, не заглиблюючись, однак, в подробиці, розгляд яких міг би скласти окрему книгу. Основними компонентами підсистеми TE є:

- призначений для користувача інтерфейс, через який адміністратор мережі може управляти політикою TE;
- IGP компонент, що поширює інформацію про топології мережі і відомості про стан мережевих ресурсів;
- маршрутизація на основі обмежень - модуль, який проводить розрахунок маршруту в мережі MPLS на основі інформації, одержуваної від призначеного для користувача інтерфейсу і IGP компонента;
- компонент сигналізації для створення і підтримки LSP (або LSP тунелю), для управління LSP (LSP тунелем) і для резервування мережевих ресурсів;
- компонент пересилання даних, в якості якого виступає сама мережа MPLS. Нагадаємо, що TE - це механізм оптимізації мережі, і тому мережеві вузли, поряд з обчисленням маршрутів на основі обмежень, повинні вміти

розраховувати традиційні маршрути згідно алгоритму SPF. Функціональна модель такої LSR TE мережі MPLS представлена на рис. 2.4.

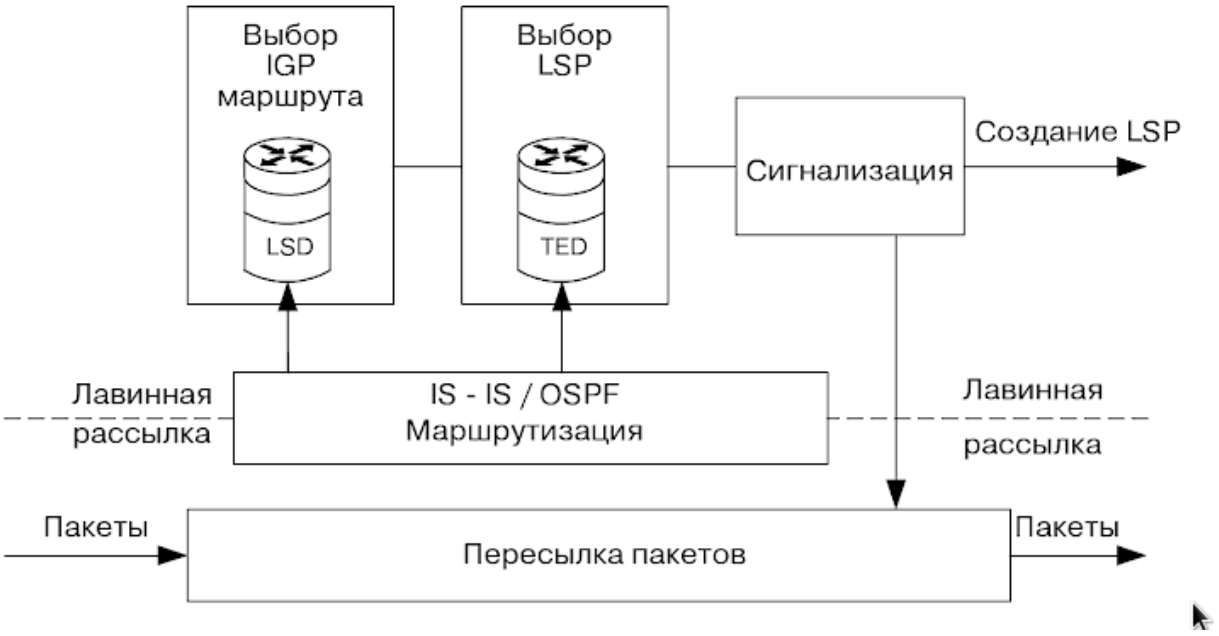


Рисунок 2.4 Модель LSR з функціями TE

LSR з функціями TE має дві окремі бази даних: одну - традиційну Link - State Database (LSD), а іншу - Traffic Engineering Database (TED), де зберігаються атрибути ланок і топологічна інформація. При використанні в мережі MPLS механізмів TE спочатку необхідно отримати статистичні дані про трафік. Зручніше збирати статистичні дані не по ланках, а по LSP, так як в цьому випадку видно обсяг трафіку між LSR для кожної пари вузлів мережі. Для пояснення наведемо приклад, представлений на рис. 2.5 та 2.6.

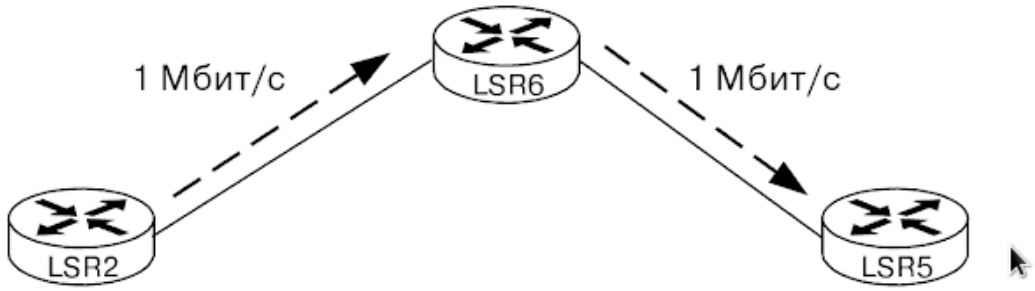


Рисунок 2.5 Трафік ланок

Якщо збирати статистичні дані по ланках (рис. 2.5), то незрозуміло, який обсяг трафіку адресований від маршрутизатору до LSR2 маршрутизатору LSR6, а який - маршрутизатору LSR5. Збираючи ж статистичні дані по LSP, як це показано на рис. 2.6, можна побачити, наприклад, що по LSP від LSR2 до LSR6 передається 300Кбіт / с, по LSP від LSR2 до LSR5 через LSR6 - 700 Кбіт / с і по LSP від LSR6 до LSR5 - 300 Кбіт / с.

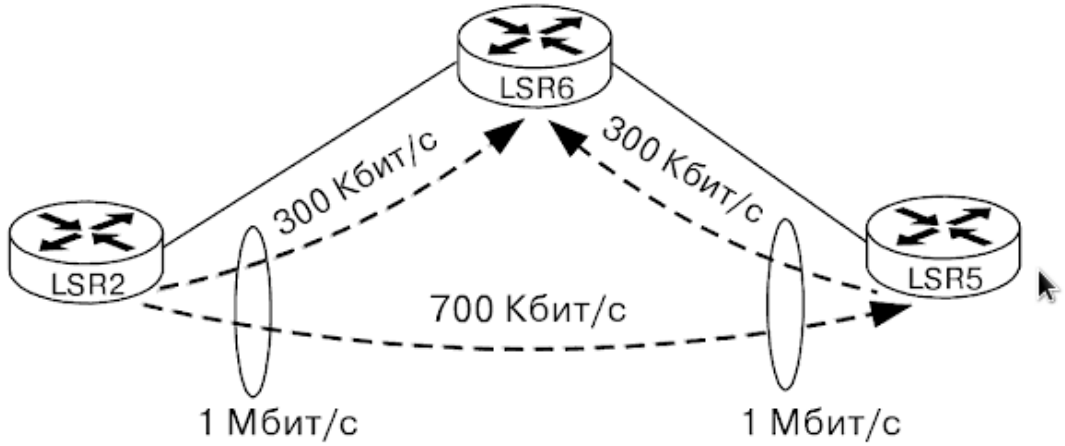


Рисунок 2.6 Трафік LSP

Можлива ситуація, коли отримувати статистичні дані не потрібно, наприклад, коли адміністратор заздалегідь знає, яка наскрізна пропускна здатність йому буде потрібно між двома вузлами мережі. Так чи інакше, тепер адміністратор може сформувати вимоги до пропускної здатності за напрямками, причому це може бути виконано як вручну, так і за допомогою засобів автоматизації. Ці вимоги адміністратор повідомляє системі через призначений для користувача інтерфейс. В процесі експлуатації мережі MPLS розширені маршрутні IGP протоколи, такі як OSPFTE і ISISTE, поширюють в мережі наступну інформацію про топології мережі та стан ресурсів:

- максимальна пропускна здатність ланки;
- максимальна пропускна здатність ланки, доступна для резервування;
- резервована на ланці пропускна здатність;
- поточне використання пропускної здатності;

- «колір» ресурсу;

Оскільки стан мережевих ресурсів змінюється набагато частіше, ніж топологія мережі, ці розширені версії протоколів OSPF і ISIS створюють в мережі більш інтенсивний службовий трафік, ніж базові протоколи, однак це виправдані затрати.

При маршрутизації, заснованої на обмеженнях (CSPF - маршрутизації), модуль маршрутизації обчислює маршрут в мережі, використовуючи інформацію, що зберігається в TED. Розрахунок може бути *тактичним* або *стратегічним*.

Тактичний розрахунок використовується при виникненні аварій або перевантажень на маршруті, і замість розрахованого маршруту, наприклад, IGP протоколом, буде автоматично створений обхідний TE маршрут.

Стратегічний розрахунок може бути розділений на online і offline маршрутизацію. При стратегічній online маршрутізації всі TE маршрути (і TE тунелі) обчислюються між прикордонними вузлами MPLS домену відповідно до заданих обмежень, і проводиться відповідне резервування ресурсів, при чому обчислення виконують самі вузли.

Стратегічна offline маршрутизація відрізняється від online маршрутизації тільки тим, що для обчислення всіх маршрутів використовується окремий сервер, який має спільне бачення мережі і її ресурсів. Таким чином, при offline маршрутизації можна домогтися найбільш ефективного використання мережевих ресурсів, так як в мережі не виникатиме суперечливих запитів резервування, а при обчисленні маршруту будуть враховуватися і інші маршрути (TE тунелі).

Але online маршрутизація швидше адаптується до змін, що відбуваються в мережі, тому на практиці ці два підходи часто використовуються спільно: LSP розраховує offline сервер, а online маршрутизація запускається лише тоді, коли робочі характеристики LSP перестають задовольняти пропонованим до них вимогам, або відбувається помітна зміна стану мережі. Крім того, процеси offline і online маршрутизації можуть запускатися з різним періодом, причому перший з них, що має більший період, обчислює маршрути, а другий виробляє їх корекцію.

Вирахувані таким чином маршрути для LSP дозволяють організувати в MPLS мережі самі тракти. Вони створюються засобами зображеного в правій частині рис. 3.4 компонента сигналізації. Модуль маршрутизації передає в сигнальний модуль дані про послідовність абстрактних вузлів. При створенні кожного LSP відбувається обмін протокольними повідомленнями, в процесі якого вздовж маршруту розподіляються мітки і резервуються мережеві ресурси. Може також враховуватися пріоритетність створюваних LSP, проводитися витіснення низько пріоритетного трафіку і оброблятися ситуації суперництва за ресурси.

Після того як всі LSP створені, підсистема TE продовжує ефективно їх підтримувати, використовуючи свої додаткові можливості. Тут не можна не згадати про таку функції, як швидка перемаршрутизація FRR (Fast ReRoute). Оскільки з появою потужних і продуктивних маршрутизаторів можливості MPLS, що спрощують процес маршрутизації, поступово відходять у тінь, основними перевагами цієї технології стають, по перше, гнучке управління проходженням трафіку (власне, основне завдання TE) і, по друге, сама FRR.

Важливість швидкої перемаршрутизації обумовлена тим, що для оператора вельми небезпечні втрати при виході з ладу ланки. Розуміється, про таку ситуацію буде проінформований крайній маршрутизатор, він зробить спробу створити новий LSP (LSP тунель) в обхід пошкодженої ділянки мережі, але через затримки, що виникають при передачі сигнальних повідомлень до кінцевого вузла в процесі розрахунку нового маршруту, можуть статися відчутні втрати даних в несправній ланці. FRR забезпечує захист від цих втрат, перемаршрутизовуючи трафік, що проходить по LSP, в обхід пошкодженої ланки. При цьому рішення про перемаршрутизацію приймаються вузлом, безпосередньо з'єднаним з несправною ланкою. Така локальна перемаршрутизація дозволяє запобігати подальшій втраті пакетів і виграти час для того, щоб інформувати крайній вузол і створити новий LSP.

Наведений на рис. 2.7 приклад показує, як FRR використовується для захисту трафіку, що переноситься між вузлами LSR1 і LSR4 при проході через ланку LSR2 -

LSR3. Для LSP від LSR1 до LSR4 через LSR2 і LSR3 використовуються мітки 25 і 9. Для захисту ланки LSR2 - LSR3 створюється резервний тунель від LSR2 до LSR3, що проходить через вузли LSR5 і LSR6. У цьому резервному тунелі будуть використовуватися мітки 38 і 15. Коли LSR2 виявить, що ланка між ним і LSR3 стало недоступною, він просто відправить трафік, адресований в сторону LSR3, в резервний тунель. Це виконується переміщенням мітки 38 наверх стеку після виконання звичайної процедури Label Swapping (Заміни мітки 25 міткою 9).

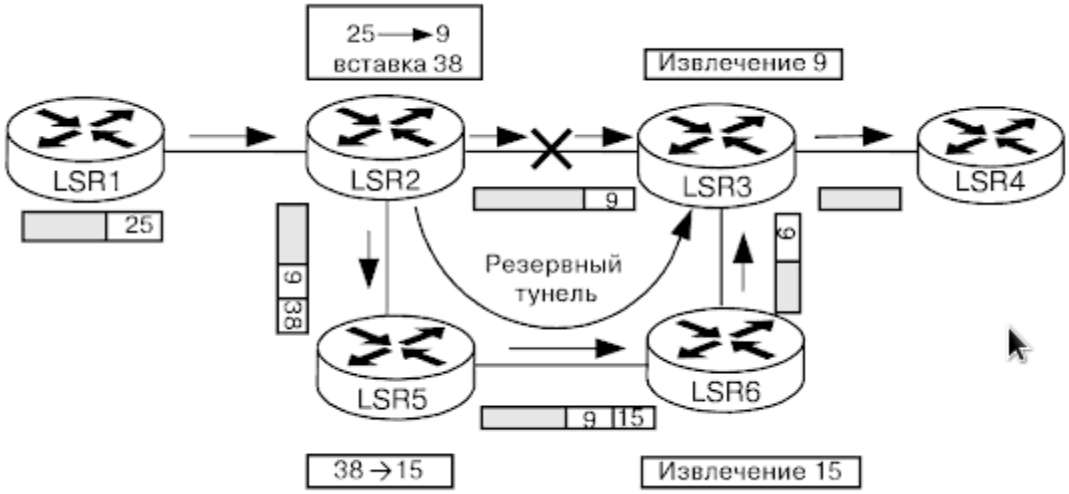


Рисунок 2.7 Приклад застосування FRR

Крім того, що FRR забезпечує додаткову надійність при передачі трафіку, вона є дуже добре маштабованим рішенням, так як всі LSP, що проходять через пошкоджену ланку, можуть бути переведені в єдиний резервний тунель, створений з урахуванням тих же обмежень, що і при розрахунку захищуваних LSP.

3.8 Порівняння протоколів CR-LDP і RSVP - TE

3.8.1 Порівняння функціональних можливостей

Слід відразу ж зазначити, що обидва розглянутих протоколи відповідають вимогам документа RFC 2702 до сигналізації MPLS. Однак для виконання цих вимог CR-LDP і RSVP - TE використовують різні механізми, хоча і схожого в реалізації ряду

функцій дуже багато. У цьому параграфі розглянуті основні функції досліджуваних протоколів і варіанти їх реалізації в кожному з них.

Можливість присвоєння і перенесення параметрів трафіку і QoS в RSVP - TE реалізується за допомогою передачі в повідомленнях непрозорих даних, призначених для підсистеми управління трафіком. CR-LDP може визначати правила для прикордонних вузлів (edge rules) і рекомендації для проміжних пересилань (per hop behaviors), що базуються на швидкості передачі даних, на смузі пропускання ланки і на вагах, присвоєних цим параметрам.

Про несправності протокол RSVP - TE сповіщає повідомленням про помилку, але момент його відправки залежить від встановлених значень таймерів для поновлення стану. CR-LDP ж користується для сповіщення про несправності можливостями транспортного протоколу TCP.

Відновлення після несправності протокол RSVP - TE забезпечує перемаршрутизацією тракту відповідно до принципу «Makebeforebrake», коли спочатку створюється новий LSP, потім в нього перекладається трафік, і лише після цього руйнується несправний тракт. CR-LDP дозволяє задати політику обробки несправності в кожному з вузлів, через які проходить LSP.

Виявлення закільцьованих маршрутів потрібно лише для не строго заданих маршрутів і в RSVP - TE проводиться за допомогою об'єкта RRO. Протокол CR-LDP використовує для цього традиційний для LDP об'єкт Path_Vector_TLV. Обидва цих об'єкта можуть також застосовуватися для визначення того, який маршрут використовує LSP.

Для управління трактами в обох протоколах застосовується ідентифікація кожного LSP ідентифікатором LSP ID. При цьому RSVP - TE може ідентифікувати і тунель (Tunnel ID), дозволяючи перевести його з одного LSP в інший.

Оба протоколу підтримують функцію витіснення високо пріоритетним трафіком низько пріоритетного. В обох протоколах це досягається присвоєнням пріоритетів утримання і захоплення ресурса.

Звісно, обидва протоколи можуть визначати для тракту явно заданий маршрут, причому обидва вони можуть працювати з абстрактними вузлами і зі строго і не строго заданими маршрутами.

3.8.2 Порівняння технічних характеристик

Ключовими відмінностями між протоколами CR-LDP і RSVP являються використовувані тим і іншим транспортні протоколи і те, в якому напрямку - прямому чи зворотному - проводиться резервування ресурсів. З відмінності за цими двома ознаками витікають і багато інших відмінності цих протоколів. У чому схожі й чим відрізняються протоколи CR-LDP і RSVPTE, показує табл. 2.1.

	CR - LDP	RSVP - TE
Використовуваний транспортний протокол	TCP	Вихідний IP
Надійність операторського класу	Ні	Так
Підтримка трафіку “багато точок - точка”	Так	Так
Підтримка мовного пересилання	Ні	Ні
Підтримка зливання LSP	Так	Так
Явна маршрутизація	З строгими та не строгими ділянками маршруту	З строгими та не строгими ділянками маршруту
Перемаршрутизація LSP	Так	Так
Закріплення маршруту	Так	Так, шляхом запису маршруту
Витіснення потоків в LSP	Так, на основі пріорітету	Так, на основі пріорітету
Засоби безпеки	Так	Так
Захист LSP	Так	Так
Стан LSP	Жосткий	Нежосткий
Регенерація станів LSP	Не потребується	Періодична, по ділянках

Резервування спільно використовуваних ресурсів	Ні	Так
Обмін параметрами трафіку	Так	Так
Керування трафіком	В прямому напрямку	В оберненому напрямку
Авторизація користувачів	Неявна	Явна
Індикація протокола рівня 3	Ні	Так
Обмеження в залежності від класу ресурсу	Так	Ні

Найбільш очевидним розходженням протоколів CR-LDP і RSVP, показаним в таблиці 2.1, є те, який транспортний протокол використовується для передачі запитів міток. RSVP використовує для цього протокол IP, що не орієнтований на з'єднання. CR-LDP використовує транспортний протокол UDP, але тільки для виявлення тимчасових маршрутизаторів мережі MPLS; для передачі протокольних повідомлень він використовує орієнтовані на з'єднання TCP сеанси. У RSVP потрібно, щоб всі прийняті пакети IP, які переносять повідомлення протоколу RSVP, доставлялися до модулю протоколу без посилання на фактичну IP адресу отримувача, що міститься в пакеті. Ця особливість може вимагати внесення незначних змін в реалізацію IP. Є два негативних аспекти використання протоколом CR-LDP транспорту TCP. По перше, реалізований в TCP механізм запобігання перевантаження може помітно гальмувати передачу інформації між маршрутизаторами. По друге, коли між двома LSR є тільки одне TCP з’єднання, TCP примусово вводить FIFO обслуговування черг повідомлень, при якій для критично важливого повідомлення немає можливості прийти до адресата раніше менш важливого повідомлення, яке було відправлено першим. Крім того, коли втрачається якийсь пакет, всі повідомлення, які йдуть за цим пакетом, затримуються до тих пір, поки не буде успішно виконана його повторна передача.

Протокол CR-LDP успадковує всі функції забезпечення безпеки, які є у протоколу TCP. На жаль, TCP вразливий з боку атак, що мають на меті порушити обслуговування, при яких робочі характеристики TCP сеансу можуть серйозно

постраждати в результаті несанкціонованого доступу до мережі, що негативно вплине на роботу протоколу CR-LDP. Адресатом повідомлень Path протоколу RSVP є вихідний LSR, а не проміжні LSR. Це означає що IPSec (Серія розроблених комітетом IETF проектів стандартів забезпечення аутентифікації та захисту передачі по протоколу IP пакетів шляхом шифрування) не може використовуватися, оскільки проміжні LSR будуть не в стані отримати доступ до інформації, що міститься в повідомленнях Path. Але RSVP має свої механізми аутентифікації і авторизації користувачів, що дозволяють перевіряти повноваження кожного відправника повідомлень і не допускати несанкціонованого або зловмисного резервування ресурсів. Аналогічні можливості могли б бути специфіковані і для протоколу CR-LDP, але орієнтований на з'єднання характер TCP сеансу робить цю вимогу менш актуальною, тому що TCP може використовувати якийсь стандартний криптографічний алгоритм.

IP трафік «точка - група точок» не підтримується протоколами CR-LDP і RSVP - TE. Але базовий протокол RSVP спочатку розроблявся з урахуванням можливості резервувати ресурси для дерев під LGPL по протоколу IP, так що його простіше розширити для підтримки багатоадресного трафіку.

Ще під час специфікації протоколу RSVP виникли сумніви щодо можливості його застосування в великих мережах через недостатню масштабованість. Ці сумніви були викликані тим фактом, що RSVP резервує ресурси для індивідуальних «мікропотоків», тобто для потоків таких даних, які відповідають, як правило, одному додатку, що функціонує на парі робочих станцій. Число «мікропотоків», що проходять через один маршрутизатор у великій IP мережі, вимірюється мільйонами, що може пред'явити серйозні вимоги до обсягу пам'яті і до продуктивності маршрутизаторів. Ця обставина іноді призводить до твердження про слабку масштабованість RSVP. Але в дійсності ми говоримо про відпустність хорошого масштабування RSVP тільки тоді, коли базовий RSVP використовується, щоб резервувати ресурси для «мікропотоків» індивідуально призначених для додатків, а коли протокол RSVP - TE

використовується для створення явно заданих трактів LSP, таке «мікрорезервування» не робиться. Домінуючим фактором, що визначає масштабованість протоколу створення LSP, є число створюваних трактів LSP, яке, зрозуміло, від протоколу не залежить.

Всі орієнтовані на з'єднання протоколи вимагають збереження даних про станах з'єднань, як на кінцевих LSR, так і на проміжних. При використанні протоколу RSVP - TE вимоги багато в чому схожі у всій мережі, тому що інформація про стан повинна зберігатися і підлягати періодичному оновленню в кожному LSR. У цю інформацію включаються параметри трафіку, дані про резервовані ресурси і про явно задані маршрути. Обсяг цієї інформації становить близько 500 байтів на один LSP.

Протокол CR-LDP вимагає, щоб вхідний і вихідний LSR зберігали подібні обсяги інформації про стан, включаючи параметри трафіку і дані про явно заданих маршрутах. Сумарний обсяг інформації про стан, необхідний для функціонування протоколу CR-LDP, становить на кінцевих вузлах теж близько 500 байтів. У проміжних LSR можливо знизити обсяг інформації, що зберігається приблизно до 200 байтів за рахунок відмови від функції модифікації LSP, тобто від можливості перемаршрутізаціі LSP або від обліку змін ресурсів. Слід зазначити, що буфери пересилання даних, необхідні для забезпечення гарантованого QoS при передачі по LSP, матимуть набагато більший розмір, ніж обсяг пам'яті, потрібний для зберігання даних про стан. Таким чином, відмінність між протоколами RSVP і CR-LDP в мережі MPLS, яка потребує підтримки модифікації LSP, робиться менш важливим.

Під надійністю операторського класу розуміється коефіцієнт готовності в «п'ять дев'яток», тобто 99,999%. Висока готовність досягається за рахунок своєчасного виявлення та усунення несправностей без якогось (або, в крайньому випадку, тільки мінімального) порушення обслуговування. Питання забезпечення живучості LSP при виникненні програмних або апаратних відмов відносяться до реалізації обладнання, і ці питання повинен розглядати і вирішувати кожен постачальник мережевого обладнання MPLS.

В зв'язку з тим, що протокол RSVP використовує транспортний механізм без встановлення з'єднання, він добре пристосований до системи, яка повинна бути стійкою до апаратних відмов або збоїв програмного забезпечення. Відомості про будь-які керівних діях, що втрачаються під час аварійного перемикання на резервну систему, можуть бути відновлені за допомогою вбудованого в протокол RSVP - TE механізму регенерації стану.

С іншого боку, протокол CR-LDP передбачає надійну доставку керуючих повідомлень і тому він більш чутливий до аварійних перемикань на резерв. Крім цього, протокол TCP дуже складно зробити відмовостійким, і тому аварійне перемикання на резервний стек протоколів TCP призводить до втраті TCP з'єднань. Така ситуація інтерпретується протоколом CR-LDP як відмова всіх відповідних LSP, які треба створювати заново від вхідного LSR.

Таким чином, спочатку протокол RSVP може забезпечувати кращі рішення для мереж MPLS з високим рівнем готовності. Ситуацію з CR-LDP виправляє документ RFC 3479, що специфікує механізми відмовостійкості для цього протоколу. Після впровадження цих розширень CR-LDP наблизиться по характеристикам забезпечення високої готовності до протоколу RSVP.

З надійністю безпосередньо пов'язано виявлення відмов в ланках і в маршрутизаторах. Якщо два LSR безпосередньо пов'язані двохточним каналом, наприклад, каналом ATM, несправність в LSP можна, як правило, виявити шляхом моніторингу стану інтерфейсів LSP. Наприклад, якщо в каналі ATM пропадає сигнал, то і протоколі CR-LDP, і протоколі RSVP можуть використовувати повідомлення про відмову в інтерфейсі для обнару-вання відмови LSP. Якщо два LSR з'єднані один з одним через спільно використовувану середу передачі, наприклад, Ethernet, або з'єднані один з одним не безпосередньо, а через хмару WAN, то вони не обов'язково отримують повідомлення про відмову ланки з канального устаткування. У таких випадках завдання виявлення відмов LSP перекладається на засоби, наявні в протоколах сигналізації. Протокол CR-LDP використовує обмін повідомленнями

Hello і Keepalive для того, щоб упевнитися, що суміжний LSR і ланка продовжують залишатися активними. Незважаючи на те, що протокол TCP має вбудовану систему підтримки з'єднань, вона, як правило, занадто повільно, з точки зору потреб LSP мережі MPLS, реагує на відмови в ланці і в маршрутизаторі. У протоколі RSVP періодично передаються для регенерації стану повідомлення Path і Resv, утворюють фоновий трафік, який вказує на те, що ланка продовжує залишатися працездатною. Однак, для зведення до мінімуму цього трафіку у відносно стабільній мережі, для таймера регенерації може бути встановлено досить велике значення. У протоколі RSVP - TE для підтвердження активності і працездатності ланки і суміжного LSR може використовуватися обмін повідомленнями Hello. Таким чином, методика і швидкість виявлення відмов в обох порівнюваних протоколах схожі.

Організація лямбда мереж пов'язана зі значною кількістю проблем, що виникають при реалізації технології MPLS в оптичній мережі. Всіма перевагами спектральної комутації можна скористатися тільки в тому випадку, якщо комутація LSP виконується апаратними засобами без допомоги ПО. Число хвиль, однак, занадто мале в порівнянні з імовірним числом LSP. Крім того, можливості однієї хвилі значно перевищують звичайні вимоги одного LSP, тому організація взаємооднозначної відповідності між ними була б недозволеною марнотратою мережевих ресурсів. Робочою групою IETF з питань MPLS випущені документи RFC, що містять необхідні доповнення як для протоколу RSVP - TE, так і для CR-LDP, що забезпечують їх працездатність в лямбда-мережах в рамках концепції GMPLS.

Важливо відзначити, що в контексті управління трафіком протоколи CR-LDP і RSVP - TE виконують резервування ресурсів на різних стадіях процесу створення LSP.

Протокол CR-LDP переносить повну інформацію про параметри трафіку в повідомленні запиту мітки Label Request. Це дозволяє кожному маршрутизатору мережі MPLS управляти трафіком при створенні LSP. Параметри трафіку можуть узгоджуватися в міру просування процесу створення LSP, а остаточні значення параметрів передаються в зворотному напрямку в повідомленні назначення мітки

Label Mapping, що забезпечує контроль доступу та резервування ресурсів в кожному LSR в реальному часі. Цей підхід дозволяє не створювати LSP по маршруту, який не має в даний момент часу достатніх ресурсів.

Протокол RSVP - TE переносить в повідомленні Path набір параметрів трафіку у вигляді специфікації потоку даних відправника Tspec. Ця специфікація описує дані, які будуть передаватися по LSP. Транзитні LSR можуть аналізувати цю інформацію і на її основі приймати рішення про маршрутизації. Однак тільки тоді, коли повідомлення Path дійде до вихідного LSR, специфікація Tspec буде перетворена в специфікацію Flowspec, яка переноситься в зворотному напрямку повідомленням Resv і в якій даються деталі резервування ресурсів, потрібних для LSP. Це означає, що резервування не проходить до тих пір, поки повідомлення Resv не пройде через мережу, а результатом може стати те, що на обраному маршруті створити LSP не вдасться через нестачу ресурсів.

Протокол RSVP - TE містить необов'язкову функцію Adspec, за допомогою якої можна повідомити про доступні ресурси в повідомленні Path. Це дозволяє вихідному LSR дізнатися, які ресурси доступні, і, відповідно до цього, модифікувати специфікацію Flowspec, що переноситься в повідомленні Resv. На жаль, ця функція не тільки вимагає, щоб її підтримували всі LSR на маршруті, але також має той очевидний недолік, що ресурси, відомості про яких несе повідомлення Path, на той час, коли буде отримано повідомлення Resv, вже можуть виявитися зайнятими іншим LSP.

Часткове рішення цієї проблеми для LSR, що використовують протокол RSVP, лежить в області реалізації: можна забезпечити попереднє резервування ресурсів при обробці повідомлення Path. Це резервування буде лише приблизними, оскільки воно спирається на специфікацію потоку даних відправника Tspec, а не на специфікацію Flowspec, але, тим не менше, воно може значно полегшити становище.

Протокол CR-LDP пропонує більш жорсткий підхід до управління трафіком, особливо в мережах, які відчувають високе навантаження.

Протокол RSVP - TE дозволяє переносити в повідомленнях Path і Resv об'єкт авторизації користувачів з непрозорим вмістом. Ця інформація використовується при обробці повідомлень для контролю доступу відповідно до встановлених правил. Це дозволяє тісно прив'язати протокол RSVP - TE, що підтримує розподіл міток, до протоколів авторизації і нагляду, наприклад, відповідно до RFC 2749, до COPS (Common Open Policy Service).

В відмінності від цього, протокол CR-LDP дозволяє переносити в неявному вигляді тільки інформацію про авторизацію у вигляді адреси одержувача та класу адміністративного ресурсу в параметрів трафіку.

Різниця між протоколами існує і в індикації протоколів рівня 3. Хоча LSP може переносити будь-які дані, бувають випадки, коли транзитному або вихідному маршрутизатору мережі MPLS може знадобитися інформація про протокол рівня 3. Якщо транзитний LSR не в змозі доставити пакет (наприклад, через відмову ресурсу), він може передати в зворотному напрямку не специфічне для протоколу рівня 3 повідомлення про помилку, що повідомляє відправника про виникнення проблеми. Щоб цей механізм працював, LSP, який виявляє помилку, повинен знати, який саме протокол рівня 3 використовується. Інформація про протокол рівня 3 також може допомогти вихідному маршрутизатору пересилати пакетні дані.

RSVP - TE ідентифікує один єдиний протокол передачі корисного навантаження на етапі створення LSP, а CR - LDP і це зробити не в стані. Але навіть RSVP не може допомогти, когда в один LSP направляется трафік більш ніж одного протоколу. Трактами LSP, створеними з вибором оптимального маршруту в мережі, можна керувати на їх вході і їх можна контролювати на їх виході за допомогою адміністративній базі даних MIB мережі MPLS.

3.8.3 Службовий трафік в RSVP-TE і CR-LDP

Обидва протоколи створюють потоки службових повідомлень для розповсюдження міток, передачі наскрізного запиту та скрізної відповіді на запит.

Протокол RSVP орієнтується на "нежорсткий стан" маршрутів. Це означає, що протокол повинен періодично оновлювати стан кожного LSP між суміжними вузлами, що дозволяє автоматично враховувати зміни в маршрутизації дерева. В якості транспортного засобу протокол RSVP використовує IP дейтаграми, так що керуючі повідомлення можуть бути втрачені, і суміжний вузол, не отримавши відповідного повідомлення, може припинити обслуговування. Регенерація стану LSP гарантує, що воно належним чином синхронізована між сусідніми вузлами. Навантаження на мережу від таких періодичних оновлень залежить від чутливості до відмов, яка регулюється виборчим значенням таймера регенерації (оновлення). Повідомлення Path протоколу RSVP буде мати довжину порядку близько 128 байт, збільшуючись на 16 байтів у кожному маршрутизаторі, якщо використовується фіксований маршрут. Повідомлення Resv буде мати довжину порядку 100 байтів. При 10 000 LSP в межузловом звене і періоді регенерації 30 секунд на регенерацію буде споживатися більше 600 Кбіт / с пропускної здатності ланки. Чимало це чи ні - залежить від характеристик зв'язку і від того, яку це становить частку від переданого по ній трафіку.

Протокол CR - LDP не вимагає від LSR періодичної регенерації кожного LSP після його створення. Це досягається завдяки тому, що в якості транспортного засобу для передачі сигнальних повідомлень протоколу CR - LDP використовує протокол TCP. Протокол CR - LDP може забезпечити надійну доставку повідомлень Label Request і Label Mapping. Використання транспортного протоколу TCP у тракті сигналізації ніякої службової інформації в цей тракт не вносить, а лише додає 20 байт до довжини кожного сигнального повідомлення. Для підтримання зв'язку з сусідніми вузлами протокол CR - LDP використовує повідомлення Hello, за допомогою якого він засвідчує, що суміжні вузли продовжують залишатися активними, і повідомлення KeepAlive для моніторингу TCP з’єднання. Періодичний обмін даними відносно коректурних повідомлень ведеться для всієї ланки, а не для кожного з численних LSP, що проходять по ним, і тому ці зв'язки практично не впливають на пропускну

здатність звена. Таким чином, протокол CR - LDP, в принципі, вносить менше навантаження в мережу, ніж протокол RSVP. Але IETF специфікував документ RFC 2961, в який входили ідеї зменшення числа реєстрацій, що вимагаються протоколом RSVP. Розглянемо детальніше рішення цього питання, запропонованого та деталізованого в рабочих групах MPLS и RSVP в складі комитету IETF.

З врахуванням взаємозв'язку між надійної доставкою інформації та непродуктивними витратами на процедури регенерації, в одному з кроків для зниження обсягу оброблюваної інформації регенерації міг би стати додаток до протоколу RSVP механізму надійної доставки повідомлень. Це було зроблено шляхом визначення розділів об'єктів протоколу RSVP, а саме об'єктів MES-SAGE_ID і MESSAGE_ID_ACK. Ці об'єкти забезпечують надійну доставку повідомлень наступним чином. Припустимо, що вузол LSR1 має протокол RSVP, представляючи новий стан, наприклад, нове повідомлення PATH, яке він повинен передавати до сусіднього вузла LSR2. Вузол LSR1 створює локально унікальний ідентифікатор нового стану, поміщає цей ідентифікатор у об'єкт MESSAGE_ID і додає його до повідомлення протоколу RSVP, встановивши в об'єкті прапорець «потребує підтвердження». Отримавши повідомлення, LSR2 підтверджує його отримання, передаючи до LSR1 повідомлення, яке містить об'єкт MESSAGE_ID_ACK з тим же ідентифікатором. Якщо LSR2 вже має якесь повідомлення, що надлежить передачі на LSR1, він просто включає MESSAGE_ID_ACK в це повідомлення; в противному випадку йому потрібно буде передавати спеціально створене повідомлення підтвердження ACK. Цей простий механізм може наступним чином забезпечити вирішення проблеми, пов'язаної з великими обсягами інформації регенерації. Вузол, що підтримує RSVP, може використати дуже короткий таймер регенерації для переданих їм повідомлень, що призводять до встановлення нового стану. При отриманні від сусіднього вузла підтвердження того, що новий стан встановлений, він може переключитися на дуже довгий таймер регенерації. Таким чином, об'єм інформаційної регенерації знижується без шкоди для своєчасності резервування.

Для всього цього такий механізм відрізняється від "жорсткого стану", оскільки повідомлення регенерації все одно повинні періодично передаватися. Відповідно, зберігаються присутні не жорсткому стану властивості «самовідновлення», включаючи інтерактивну обробку відмов. Крім того, цей механізм володіє зворотньою сумісністю, тобто не створює ніяких проблем, якщо в окремих вузлах він реалізується, а в інших - нет.

Подальше зниження обсягу інформації регенерації досягнуто за допомогою механізму, який називається скороченою або прискореною регенерацією. Суть його в тому, що після того як ідентифікатор повідомлень вже прив'язаний до повідомлення протоколу RSVP, для регенерації стану немає необхідності передавати знову і знову повне повідомлення, як це зазвичай робиться. Вузол може просто передавати сам ідентифікуюче повідомлення. Крім того, можлива упаковка досить великої кількості ідентифікаторів повідомлень в одне повідомлення, так що одне повідомлення протоколу RSVP може оновити цілий ряд станцій. Така можливість корисна, тому що часто потрібні значні обчислювальні витрати, тільки для отримання повідомлень в процесорі маршрутизатора, незалежно від того, наскільки велике це повідомлення чи від того, які витрати на обробку його вмісту. Для підтримки механізму скороченої регенерації було визначено нове повідомлення протоколу RSVP - повідомлення SRE-FRESH. При адресації до певного пристрою ця пошта містить просто список перерахованих ідентифікаторів повідомлення. Вузол, що отримує таке повідомлення, веде себе так, якби він отримав набір повідомлень про регенерацію, кожен з яких ідентичний повідомленню, в якому був первісно переданий об'єкт MESSAGE_ID, тобто він скидає таймер регенерації.

Одна з можливих проблем - отримання ідентифікатора MESSAGE_ID, який не розпізнається. Це може виникнути, якщо наступна ділянка маршруту змінилася, а передаючий вузол цю зміну не виявив. Для виходу з такої ситуації приймаючий вузол, не розпізнавши ідентифікатор повідомлення, може передати негативне підтвердження

MESSAGE_ID_NAK. Передаючий вузол, прийнявши таке підтвердження, повинен повторити повне повідомлення протоколу RSVP, до якого відносився MESSAGE_ID.

У результаті всіх цих удосконалень були вирішені проблеми, пов'язані з масштабуванням RSVP при його використанні для MPLS. З’явилася можливість багатократно здійснювати резервування ресурсів і трактів LSP при невеликому обсязі трафіку регенерації.

Проблема зниження обсягу передаваної службової інформації в RSVP була розглянута нами так детально тому, що зазвичай одним із аргументів противників RSVP був саме трафік поновлення стану. Тепер же видно, що після реалізації в протоколі RSVP механізму, що скорочує об'єм інформації регенерації, відмінність між RSVP і CR-LDP через цю ознаку стає неважливим.

3.8.4 Порівняння перемаршрутизації в RSVP - TE і CR - LDP

Розглянемо питання зміни маршруту для LSP при отриманні повідомлення про відмову або при зміні топології мережі. Попереднє програмування резервних (альтернативних) маршрутів для тракту ЛСП називається захистом LSP і розглядається в цьому підрозділі нижче. Явно заданий LSP може бути перемаршрутизований тільки вхідним LSR - відправником даних. Відповідно, про відмову в деякій точці LSP повинна бути інформована вхідною LSR, і при цьому поступово руйнується вся LSP. Однак не строго специфікована ділянка LSP з явним маршрутом і будь-яка частина LSP, маршрут для якої заданий по ділянках, можуть бути перемаршрутизовані, якщо виявлено відмову ланки або з'єднувального маршрутизатора (локальне відновлення), або якщо стає доступним кращий маршрут, або якщо ресурси LSP потрібні для створення нового LSP з більш високим приоритетом (пріоритетне витіснення). Вище вже розглядається той випадок, коли мережа підтримує функцію Fast ReRoute.

Перемаршрутизацію, що управляється вхідним вузлом, підтримує як протокол CR - LDP, так і протокол RSVP - TE, хоча є і незначні розбіжності в тому, як це робиться.

Маршрутизатор LSR, що використовує протокол RSVP - TE, може визначити новий маршрут, як тільки стане доступним і / або потребуется альтернативный маршрут, просто шляхом поновлення LSP, в результаті чого вибирається інший наступний маршрутизатор. Старий тракт при цьому не використовується і руйнується після спрацьовування таймера, оскільки повідомлення про регенерацію по ньому більше не передаються. Ясно, що при такому способі недоцільно тратяться ресурси старого LSP.

Це можна уникнути, передав по явно заданому маршруту повідомлення PathTear або ResvTear. При цьому активізується механізм включення нового тракту до розриву старого (make - before - break), тобто механізм, при якому старий тракт продовжує використовуватися (і обновляться) весь час, поки створюється новий тракт, а після його створення LSR, що виробляє перемаршрутизацію, виконує переключення на новий тракт і руйнує старий тракт. Ця методика дозволяє уникнути подвійного резервування ресурсів як у протоколі CR - LDP (використання значення modify флагу дії в повідомленні Label Request), так і в протоколі RSVP - TE (використанням фільтрів при стилі резервування Shared - Explicit).

У нестабільних мережах інтенсивний службовий трафік, що забезпечує перемаршрутизацію нестрого специфічних ділянок ключів LSP на проміжних LSR при появі кращих маршрутів, може призвести до виникнення перевантажень. Для того, щоб це запобігти, нестрого специфічний маршрут ділянки може бути закріплений таким чином.

В протоколі CR - LDP це робиться просто шляхом позначення нестрогої ділянки явно заданого маршруту як закріпленого. Це означає, що як тільки маршрут буде визначений, до нього будуть відноситися, як до строго конкретного маршруту, який змінюватися не може.

В протоколі RSVP закріплення вимагає деякої додаткової обробки. Нехай першочерговий маршрут специфікується з нестрогою ділянкою. Щоб повідомити вхідний LSR про ваш маршрут, в повідомленнях Path і Resv використовується об'єкт Route Record (RRO). Вхідний LSR може потім використовувати цю інформацію для повторної передачі повідомлення Path, який буде містити строго специфічний явний маршрут.

І RSVP - TE, і CR - LDP використовують гнучкий підхід до перемаршрутизації LSP і до застосування механізму з включенням нового LSP до розриву старого. Протокол CR - LDP спирається на внесену в неї специфікацію додавання, що дозволяє підтримувати включення нового до розриву старого, а протокол RSVP - TE вимагає додаткового обміну повідомленнями для закріплення маршруту. Слід зазначити один з недоліків протоколу CR - LDP, пов'язаний з використанням протоколу TCP для LDP-сесій. При роботі CR - LDP мають місце додаткові витрати часу на визначення відносини суміжності між двома LSP. Перед тим, як маршрутизатори зможуть ініціювати LDP-сесії, вони повинні пройти процедуру входу в зв'язок за протоколом TCP. Це дає перевагу протоколу RSVP, який не вимагає встановлення зв'язку перед початком процедури розподілу міток. Можна сказати, що протокол RSVP має «облегчені відносини суміжності», які дозволяють визначати нові взаємозв'язки між сусідніми маршрутизаторами швидко, за необхідності. І це важливо для виконання швидкої перемаршрутизації.

Модифікація LSP, наприклад, при зміні параметрів трафіку в LSP, це операція, еквівалентна перемаршрутизації, хоча при модифікації LSP зміна маршруту не обов’язкова. Тому ця функція завжди присутня в протоколі RSVP - TE і буде присутня в протоколі CR - LDP за умови, що реалізація протоколу підтримує значення модифікації флагу дії в повідомленнях Request Label (справа в тому, що ранні реалізації протоколу модифікації LSP не підтримують).

Захист LSP заключається в програмуванні резервних шляхів для тракту LSP з автоматичним переключенням на резервний тракт при відмові основного тракту.

Незважаючи на те, що з концептуальної точки зору ця функція також аналогічна перемаршрутизації, захист LSP звичайно розглядається як набагато більш важлива операція, метою якого є оперативне переключення на новий тракт з мінімально можливим перериванням передачі даних по LSP (як правило, розрахункова тривалість переривання повинна бути меншою 50 мс). В обох порівнюваних протоколах можуть підтримуватися декілька рівнів захисту LSP. Простішим видом захисту LSP є спроба вхідного або транзитного LSR виконати перемаршрутизацію LSP негайно після отримання повідомлення про відмову. Така можливість існує в обох протоколах, однак, через необхідності передавати різноманітні сигнальні повідомлення, аварійне переключення на новий маршрут відбувається відносно повільно (зазвичай це займає, як мінімум, кілька секунд). Така невисока швидкість перемикання неприйнятна для IP-телефона та деяких інших додатків реального часу.

Набагато більш швидкого захисту LSP можна досягти, якщо між носіями двох LSR захищена схема захисту на рівні 2. Такий захист прозорий для LSP і може застосовуватися з будь-яким з двох порівнюваних протоколів. Взагалі, реалізація захисту на рівні 2 може виявитися дорогою справою, і сам захист обмежений учасником LSP між двома сусідніми маршрутизаторами. До того ж, захист ланки не забезпечує захисту від відхилення окремих LSP.

3.8.5 Порівняння протоколів по їх впровадженню

Єдиним фактором, що впливає на практичне впровадження як RSVP - TE, так і CR - LDP, є перевага виробників відповідного мережевого обладнання. Після призупинення IETF роботи над протоколом CR - LDP, виробники були вимушені концентрувати свої зусилля на RSVP - TE, але так як багато мереж вже використовують CR - LDP, і оператори не поспішають від нього відмовлятися, виробники продовжують підтримувати і рішення на його основі.

Незважаючи на те, що стандартний протокол RSVP вже протягом ряду років надається багатьма постачальниками обладнання і є поширеним мережевим

протоколом, ті зміни, які потрібно внести в стандартний RSVP для підтримки розповсюдження міток, настільки нетривіальні, що протокол RSVP - TE стає в багатьох відносинах новим протоколом.

Протокол CR - LDP базується на ідеях, які були реалізовані в відомчих мережах вже більше десяти років тому, але в якості стандартизованого протоколу IETF він є відносно молодим. Як вже зазначалося вище, виробники обладнання на даний момент підтримують або обидва протоколи, або тільки RSVP, хоча спочатку такі компанії, як Nortel Networks та Nokia, активно просунули рішення на базі CR - LDP протоколу, а Cisco Systems та Juniper Networks займалися протоколом RSVP - TE. Великі компанії, що надають телекомунікаційну техніку, звичайно розміщують в Інтернеті поновлення та нові версії свого програмного забезпечення, які або доступні безкоштовно, або призначені тільки для авторизованих клієнтів компанії. В нових версіях враховуються зміни, внесені в той чи інший протокол міжнародними стандартизуючими організаціями та дослідницькими центрами самих компаній. Але звичайно це - вже готовое програмне забезпечення, предназначене для роботи з устаткуванням певного типу. Компанія Nortel Networks, продовжуючи підтримку протоколу CR - LDP, розмістила на своєму веб-сайті вільно доступний вихідний код своєї програмної реалізації CR - LDP. Таким чином, операторам, що потребують в реалізації функцій MPLS, у вже встановленому мережевому обладнанні, трохи простіше робити це з протоколом CR - LDP, ніж з RSVP - TE. При виборі нового обладнання, перевага, швидше за все, буде віддаватися RSVP, як протоколу який розвивається і вдосконалюється.

Другим питанням, який встає перед оператором, що впроваджує в своїй мережі новий протокол, є функціональна сумісність версій вже обраного протоколу. Всегда існують два питання на які потрібні відповіді про сумісність: чи підтримують дві реалізації протоколу ті самі опції і чи інтерпретують вони специфікації однаково. Протокол RSVPE - TE має більше варіантів реалізації, ніж CR - LDP протокол, і тому, на перший погляд, більш високий ризик того, що реалізація RSVP - TE буде

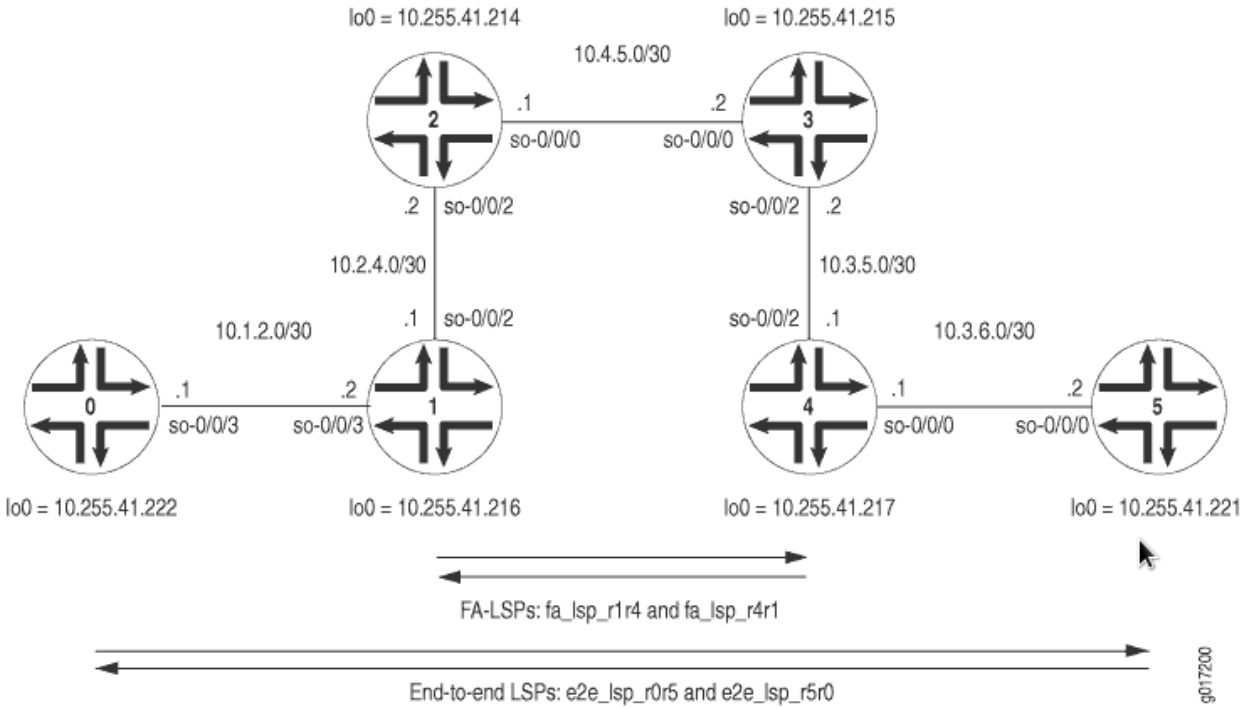
виявлятися функціонально несумісними. Єдиним способом перевірити, що дві реалізації протоколу коректно взаємодіють одна з одною, це, зрозуміло, тестування їх функціональної сумісності. Участь у спільному тестуванні функціональної сумісності є обов'язковим для всіх постачальників мережевого обладнання MPLS.

Оскільки в мережі MPLS TE звичайно функціонує і традиційний для MPLS механізм розподілу міток, тобто протокол LDP, ми також розглянемо питання про його підключення з аналізованими тут протоколами. В зв'язку з тим, що протокол CR - LDP побудований як розширення протоколу LDP, реалізаціям CR - LDP легше підтримувати і функції LDP. З іншого боку, протокол RSVPE - TE є повністю самостійним протоколом, і хоча RSVPE - TE також підтримує послідовне створення LSP по ділянках, для підтримки функцій протоколу LDP в ньому має бути реалізовано додатковий протокол стеку. Цей факт очевидний, але совсем не очевидно, що краще: мати більше протоколів або менше. Наприклад, IP, ICMP і ARP є трьома самостійними протоколами, які потрібні для передачі IP-пакетів по Інтернету, і мені здається малоймовірним, що IP-мережі працювали б краще, якби ці три протоколи раптом будь-яким способом злилися в один. Можна сказать, тільки те, що це відмінність протоколів RSVP - TE і CR - LDP не говорить про істотну перевагу останнього.

Висновки

Виведений аналіз двох протоколів сигналізації в мережах MPLS - TE дозволяє оцінити особливості застосування кожного з них. В цілому обидва протоколи володіють сьогодні практично однаковими можливостями і технічними характеристиками. Але це сьогодні. Після рішення робочої групи IETF по питанням MPLS припинити роботу по CR - LDP, цей протокол з часом розпочне вступ до позиції всіх удосконалюваних протоколів RSVP - TE. І в майбутньому, судячи по кількості до протоколу RSVP - TE документів RFC, що знаходяться на розгляді в IETF, цей розрив буде тільки збільшуватися.

4. Практична реалізація RSVP - TE за допомогою GNS3



На схемі показано кінцевий RSVP LSP, який називається e2e_lsp_r0r5, що походить з маршрутизатора 0 і закінчується на маршрутизаторі 5. При переході цей LSP перетинає FA_LSP fa_lsp_r1r4. Шлях до зворотного зв'язку представлений кінцевим інтервалом RSVP LSP e2e_lsp_r5r0, що переміщується через FA-LSP fa_lsp_r4r1.

Конфігурація на Router 0:

На маршрутизаторі 0 налаштували кінцевий RSVP LSP, що переміщується до маршрутизатора 5. Використали суворий шлях, який переміщує маршрутизатор 1 та інженерний зв'язок LMP, що переходить від маршрутизатора 1 до маршрутизатора 4.

```
[edit]
interfaces {
   so-0/0/3 {
      unit 0 {
         family inet {
            address 10.1.2.1/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.41.222/32;
         }
         family mpls;
      }
   }
}
routing-options {
   forwarding-table {
      export pplb;
   }
}
protocols {
   rsvp {
      interface all;
      interface fxp0.0 {
         disable;
      }
   }
   mpls {
      admin-groups {
         fa 1;
         backup 2;
         other 3;
      }
```

```
      label-switched-path e2e_lsp_r0r5 { # An end-to-end LSP traveling to Router 5.
         to 10.255.41.221;
         bandwidth 30k;
         primary path-fa; # Reference the requested path here.
      }
      path path-fa { # Configure the strict path here.
         10.1.2.2 strict;
         172.16.30.2 strict; # This traverses the TE link heading to Router 4.
      }
      interface all;
      interface fxp0.0 {
         disable;
      }
      interface so-3/2/1.0 {
         admin-group other;
      }
      interface so-0/0/3.0 {
         admin-group other;
      }
   }
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface fxp0.0 {
            disable;
         }
         interface all;
      }
   }
}
policy-options {
   policy-statement pplb {
      then {
         load-balance per-packet;
      }
   }
}
```

Конфігурація на Router 1

На маршрутизаторі 1 налаштували FA-LSP, щоб досягти маршрутизатора 4. Встановили зв'язок LMP-трафіку та зв'язок LMP-партнера з маршрутизатором 4. Посилання на FA-LSP в посиланнях на технологію трафіку та додали інтерфейс peer в OSPF та RSVP.

Коли маршрут повернення до кінцевої LSP надходить на маршрутизатор 1, платформа маршрутизації виконує пошук маршрутизації і може передати трафік маршрутизатору 0.

```
[edit]
interfaces {
   so-0/0/1 {
      unit 0 {
         family inet {
            address 10.2.3.1/30;
         }
         family mpls;
      }
   }
   so-0/0/2 {
      unit 0 {
         family inet {
            address 10.2.4.1/30;
         }
         family mpls;
      }
   }
   so-0/0/3 {
      unit 0 {
         family inet {
            address 10.1.2.2/30;
         }
         family mpls;
      }
   }
   fe-0/1/2 {
      unit 0 {
         family inet {
            address 10.2.5.1/30;
         }
         family mpls;
      }
   }
   at-1/0/0 {
      atm-options {
         vpi 1;
```

```
      }
      unit 0 {
         vci 1.100;
         family inet {
            address 10.2.3.5/30;
         }
         family mpls;
      }
   }
}
routing-options {
   forwarding-table {
      export [ pplb choose_lsp ];
   }
}
protocols {
   rsvp {
      interface all;
      interface fxp0.0 {
         disable;
      }
      peer-interface r4; # Apply the LMP peer interface here.
   }
   mpls {
      admin-groups {
         fa 1;
         backup 2;
         other 3;
      }
      label-switched-path fa_lsp_r1r4 { # Configure your FA-LSP to Router 4 here.
         to 10.255.41.217;
         bandwidth 400k;
         primary path_r1r4; # Apply the FA-LSP path here.
      }
      path path_r1r4 { # Configure the FA-LSP path here.
         10.2.4.2;
         10.4.5.2;
         10.3.5.1;
      }
      interface so-0/0/3.0 {
         admin-group other;
      }
```

```
      }
      interface so-0/0/1.0 {
         admin-group fa;
      }
      interface at-1/0/0.0 {
         admin-group backup;
      }
      interface fe-0/1/2.0 {
         admin-group backup;
      }
      interface so-0/0/2.0 {
         admin-group fa;
      }
   }
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface fxp0.0 {
            disable;
         }
         interface all;
         peer-interface r4; # Apply the LMP peer interface here.
      }
   }
   link-management { # Configure LMP statements here.
      te-link link_r1r4 { # Assign a name to the TE link here.
         local-address 172.16.30.1; # Configure a local address for the TE link.
         remote-address 172.16.30.2; # Configure a remote address for the TE link.
         te-metric 1; # Manually set a metric here if you are not relying on CSPF.
         label-switched-path fa_lsp_r1r4; # Reference the FA-LSP here.
      }
      peer r4 { # Configure LMP peers here.
         address 10.255.41.217; # Configure the loopback address of your peer here.
         te-link link_r1r4; # Apply the LMP TE link here.
      }
   }
}
policy-options {
   policy-statement choose_lsp {
      term A {
         from community choose_e2e_lsp;
         then {
```

```
            install-nexthop strict lsp e2e_lsp_r1r4;
            accept;
         }
      }
      term B {
         from community choose_fa_lsp;
         then {
            install-nexthop strict lsp fa_lsp_r1r4;
            accept;
         }
      }
   }
   policy-statement pplb {
      then {
         load-balance per-packet;
      }
   }
   community choose_e2e_lsp members 1000:1000;
   community choose_fa_lsp members 2000:2000;
   community set_e2e_lsp members 1000:1000;
   community set_fa_lsp members 2000:2000;
}
```

Конфігурація на Router 2

На маршрутизаторі 2 налаштувалие OSPF, MPLS та RSVP на всіх інтерфейсах, які транспортують FA-LSP через основну мережу.

```
[edit]
interfaces {
   so-0/0/0 {
      unit 0 {
         family inet {
            address 10.4.5.1/30;
         }
         family mpls;
      }
   }
   so-0/0/1 {
      unit 0 {
         family inet {
            address 10.1.4.2/30;
         }
         family mpls;
      }
   }
```

```
   so-0/0/2 {
      unit 0 {
         family inet {
            address 10.2.4.2/30;
         }
         family mpls;
      }
   }
   fe-0/1/2 {
      unit 0 {
         family inet {
            address 10.3.4.2/30;
         }
         family mpls;
      }
   }
}
routing-options {
   forwarding-table {
      export pplb;
   }
}
protocols { # OSPF, MPLS, and RSVP form the core backbone for the FA-LSPs.
   rsvp {
      interface all;
      interface fxp0.0 {
         disable;
      }
   }
   mpls {
      admin-groups {
         fa 1;
         backup 2;
         other 3;
      }
      path path_r1 {
         10.2.4.1;
      }
      path path_r3r4 {
         10.4.5.2;
         10.3.5.1;
      }
```

```
      }
      interface all;
      interface fxp0.0 {
         disable;
      }
      interface so-0/0/1.0 {
         admin-group other;
      }
      interface fe-0/1/2.0 {
         admin-group backup;
      }
      interface so-0/0/2.0 {
         admin-group fa;
      }
      interface so-0/0/0.0 {
         admin-group fa;
      }
   }
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface fxp0.0 {
            disable;
         }
         interface all;
      }
   }
}
policy-options {
   policy-statement pplb {
      then {
         load-balance per-packet;
      }
   }
}
```

Конфігурація на Router 3

На маршрутизаторі 3 налаштували OSPF, MPLS та RSVP на всіх інтерфейсах, які передають FA-LSP через основну мережу.

```
[edit]
interfaces {
   so-0/0/0 {
      unit 0 {
         family inet {
            address 10.4.5.2/30;
         }
         family mpls;
      }
   }
   so-0/0/1 {
      unit 0 {
         family inet {
            address 10.5.6.1/30;
         }
         family mpls;
      }
   }
   so-0/0/2 {
      unit 0 {
         family inet {
            address 10.3.5.2/30;
         }
         family mpls;
      }
   }
   fe-0/1/2 {
      unit 0 {
         family inet {
            address 10.2.5.2/30;
         }
         family mpls;
      }
   }
}
routing-options {
   forwarding-table {
      export pplb;
   }
}
```

```
protocols { # OSPF, MPLS, and RSVP form the core backbone for the FA-LSPs.
   rsvp {
      interface all;
      interface fxp0.0 {
         disable;
      }
   }
   mpls {
      admin-groups {
         fa 1;
         backup 2;
         other 3;
      }
      path path_r4 {
         10.3.5.1;
      }
      path path_r2r1 {
         10.4.5.1;
         10.2.4.1;
      }
      interface all;
      interface fxp0.0 {
         disable;
      }
      interface so-0/0/2.0 {
         admin-group fa;
      }
      interface fe-0/1/2.0 {
         admin-group backup;
      }
      interface so-0/0/1.0 {
         admin-group other;
      }
      interface so-0/0/0.0 {
         admin-group fa;
      }
   }
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface fxp0.0 {
```

```
            disable;
         }
         interface all;
      }
   }
}
policy-options {
   policy-statement pplb {
      then {
         load-balance per-packet;
      }
   }
}
```

Конфігурація на Router 4

На маршрутизаторі 4 налаштували шлях повернення FA-LSP для досягнення маршрутизатора 1. Встановили зв'язок LMP-трафіку та LMP-співрозмовників з маршрутизатором 1. Поставили посилання на FA-LSP у лінію зв'язку трафіку та додали одноранговий інтерфейс як в OSPF, так і в RSVP.

Коли початковий кінцевий LSP надходить на маршрутизатор 4, платформа маршрутизації виконує пошук маршрутизації і може передати трафік на маршрутизатор 5.

```
[edit]
interfaces {
   so-0/0/0 {
      unit 0 {
         family inet {
            address 10.3.6.1/30;
         }
         family mpls;
      }
   }
   so-0/0/1 {
      unit 0 {
         family inet {
            address 10.2.3.2/30;
```

```
      interface fxp0.0 {
         disable;
      }
      peer-interface r1; # Apply the LMP peer interface here.
   }
   mpls {
      admin-groups {
         fa 1;
         backup 2;
         other 3;
      }
      label-switched-path fa_lsp_r4r1 { # Configure your FA-LSP here.
         to 10.255.41.216;
         bandwidth 400k;
         primary path_r4r1; # Apply the FA-LSP path here.
      }
      path path_r4r1 { # Configure the FA-LSP path here.
         10.3.5.2;
         10.4.5.1;
         10.2.4.1;
      }
      interface all;
      interface fxp0.0 {
         disable;
      }
      interface at-1/0/0.0 {
         admin-group backup;
      }
      interface so-0/0/2.0 {
         admin-group fa;
      }
      interface fe-0/1/2.0 {
         admin-group backup;
      }
      interface so-0/0/0.0 {
         admin-group other;
      }
      interface so-0/0/1.0 {
         admin-group fa;
      }
   }
```

```
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface fxp0.0 {
            disable;
         }
         interface all;
         peer-interface r1; # Apply the LMP peer interface here.
      }
   }
   link-management { # Configure LMP statements here.
      te-link link_r4r1 { # Assign a name to the TE link here.
         local-address 172.16.30.2; # Configure a local address for the TE link.
         remote-address 172.16.30.1; # Configure a remote address for the TE link.
         te-metric 1; # Manually set a metric here if you are not relying on CSPF.
         label-switched-path fa_lsp_r4r1; # Reference the FA-LSP here.
      }
      peer r1 { # Configure LMP peers here.
         address 10.255.41.216; # Configure the loopback address of your peer here.
         te-link link_r4r1; # Apply the LMP TE link here.
      }
   }
}
policy-options {
   policy-statement choose_lsp {
      term A {
         from community choose_e2e_lsp;
         then {
            install-nexthop strict lsp e2e_lsp_r4r1;
            accept;
         }
      }
      term B {
         from community choose_fa_lsp;
         then {
            install-nexthop strict lsp fa_lsp_r4r1;
            accept;
         }
      }
   }
```

```
   policy-statement pplb {
      then {
         load-balance per-packet;
      }
   }
   community choose_e2e_lsp members 1000:1000;
   community choose_fa_lsp members 2000:2000;
   community set_e2e_lsp members 1000:1000;
   community set_fa_lsp members 2000:2000;
}
```

Конфігурація на Router 5

На маршрутизаторі 5 налаштували кінцевий RSVP LSP зворотного шляху, який переходить до маршрутизатора 0. Використали суворий шлях, який перетинає маршрутизатор 4 та інженерну лінію LMP, що переходить з маршрутизатора 4 в маршрутизатор 1.

```
[edit]
interfaces {
   so-0/0/2 {
      unit 0 {
         family inet {
            address 10.3.6.2/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.41.221/32;
         }
      }
   }
}
routing-options {
   forwarding-table {
      export pplb;
   }
}
```

```
protocols {
   rsvp {
      interface all;
      interface fxp0.0 {
         disable;
      }
   }
   mpls {
      admin-groups {
         fa 1;
         backup 2;
         other 3;
      }
      label-switched-path e2e_lsp_r5r0 { # An end-to-end LSP returning to Router 0.
         to 10.255.41.222;
         bandwidth 30k;
         primary path-fa; # Reference the requested path here.
      }
      path path-fa { # Configure the strict path here.
         10.3.6.1 strict;
         172.16.30.1 strict; # This traverses the TE link heading to Router 1.
      }
      interface all;
      interface fxp0.0 {
         disable;
      }
      interface so-0/0/2.0 {
         admin-group other;
      }
      interface so-0/0/1.0 {
         admin-group other;
      }
   }
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface fxp0.0 {
            disable;
         }
         interface all;
      }
```

```
   }
}
policy-options {
   policy-statement pplb {
      then {
         load-balance per-packet;
      }
   }
}
```

Перевірка:

Щоб перевірити, чи працює наш тунель RSVP LSP, вели наступні команди:

- `show ted database`
- `show rsvp session name`
- `show link-management`
- `show link-management te-link name`

Маршрутизатор 0

На маршрутизаторі 0 можна перевірити, що FA-LSP виглядають як допустимі шляхи в базі даних техніки руху. У цьому випадку знайшли шляхи з маршрутизатора 1 (10.255.41.216) та маршрутизатора 4 (10.255.41.217), що посилається на адреси лінійки трафіку LMP 172.16.30.1 та 172.16.30.2. Також можете ввели велику команду show rsvp session, щоб шукати шлях кінцевого LSP, коли він переходить до маршрутизатора 5 через FA-LSP.

```
user @ router0> show ted database
```

```
TED database: 0 ISIS nodes 8 INET nodes
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.214               Rtr     486     4      4 OSPF(0.0.0.0)
    To: 10.255.41.222, Local: 10.1.4.2, Remote: 10.1.4.1
    To: 10.255.41.216, Local: 10.2.4.2, Remote: 10.2.4.1
    To: 10.255.41.215, Local: 10.4.5.1, Remote: 10.4.5.2
    To: 10.3.4.1-1, Local: 10.3.4.2, Remote: 0.0.0.0
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.215               Rtr     187     4      4 OSPF(0.0.0.0)
    To: 10.255.41.214, Local: 10.4.5.2, Remote: 10.4.5.1
    To: 10.255.41.217, Local: 10.3.5.2, Remote: 10.3.5.1
    To: 10.255.41.221, Local: 10.5.6.1, Remote: 10.5.6.2
    To: 10.2.5.1-1, Local: 10.2.5.2, Remote: 0.0.0.0
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.216                Rtr     396     6      6 OSPF(0.0.0.0)
    To: 10.255.41.222, Local: 10.1.2.2, Remote: 10.1.2.1
    To: 10.255.41.214, Local: 10.2.4.1, Remote: 10.2.4.2
    To: 10.255.41.217, Local: 10.2.3.1, Remote: 10.2.3.2
     To: 10.255.41.217, Local: 172.16.30.1, Remote: 172.16.30.2
    To: 10.255.41.217, Local: 10.2.3.5, Remote: 10.2.3.6
    To: 10.2.5.1-1, Local: 10.2.5.1, Remote: 0.0.0.0
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.217                Rtr     404     6      6 OSPF(0.0.0.0)
    To: 10.255.41.216, Local: 10.2.3.2, Remote: 10.2.3.1
     To: 10.255.41.216, Local: 172.16.30.2, Remote: 172.16.30.1
    To: 10.255.41.216, Local: 10.2.3.6, Remote: 10.2.3.5
    To: 10.255.41.215, Local: 10.3.5.1, Remote: 10.3.5.2
    To: 10.255.41.221, Local: 10.3.6.1, Remote: 10.3.6.2
    To: 10.3.4.1-1, Local: 10.3.4.1, Remote: 0.0.0.0
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.221               Rtr     481     2      2 OSPF(0.0.0.0)
    To: 10.255.41.215, Local: 10.5.6.2, Remote: 10.5.6.1
    To: 10.255.41.217, Local: 10.3.6.2, Remote: 10.3.6.1
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.222               Rtr    2883     2      2 OSPF(0.0.0.0)
    To: 10.255.41.216, Local: 10.1.2.1, Remote: 10.1.2.2
    To: 10.255.41.214, Local: 10.1.4.1, Remote: 10.1.4.2
```

```
user@router0>  show ted database 10.255.41.216 extensive
TED database: 0 ISIS nodes 8 INET nodes
NodeID: 10.255.41.216
  Type: Rtr, Age: 421 secs, LinkIn: 6, LinkOut: 6
  Protocol: OSPF(0.0.0.0)
    To: 10.255.41.222, Local: 10.1.2.2, Remote: 10.1.2.1
      Color: 0x8 other
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
          [0] 155.4Mbps    [1] 155.4Mbps   [2] 155.4Mbps   [3] 155.4Mbps
          [4] 155.4Mbps    [5] 155.4Mbps   [6] 155.4Mbps   [7] 155.4Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 155.4Mbps    [1] 155.4Mbps   [2] 155.4Mbps   [3] 155.4Mbps
          [4] 155.4Mbps    [5] 155.4Mbps   [6] 155.4Mbps   [7] 155.4Mbps
    To: 10.255.41.214, Local: 10.2.4.1, Remote: 10.2.4.2
      Color: 0x2 fa
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
          [0] 155.12Mbps   [1] 155.12Mbps  [2] 155.12Mbps  [3] 155.12Mbps
          [4] 155.12Mbps   [5] 155.12Mbps  [6] 155.12Mbps  [7] 155.12Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 155.12Mbps   [1] 155.12Mbps  [2] 155.12Mbps  [3] 155.12Mbps
          [4] 155.12Mbps   [5] 155.12Mbps  [6] 155.12Mbps  [7] 155.12Mbps
    To: 10.255.41.217, Local: 10.2.3.1, Remote: 10.2.3.2
      Color: 0x2 fa
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
```

```
      [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
  Interface Switching Capability Descriptor(1):
    Switching type: Packet
    Encoding type: Packet
    Maximum LSP BW [priority] bps:
      [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
      [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
To: 10.255.41.217,  Local: 172.16.30.1, Remote: 172.16.30.2
  Metric: 1
  Static BW: 400kbps
  Reservable BW: 400kbps
  Available BW [priority] bps:
      [0] 370kbps      [1] 370kbps     [2] 370kbps     [3] 370kbps
      [4] 370kbps      [5] 370kbps     [6] 370kbps     [7] 370kbps
  Interface Switching Capability Descriptor(1):
    Switching type: Packet
    Encoding type: Packet
    Maximum LSP BW [priority] bps:
      [0] 370kbps      [1] 370kbps     [2] 370kbps     [3] 370kbps
      [4] 370kbps      [5] 370kbps     [6] 370kbps     [7] 370kbps
To: 10.255.41.217, Local: 10.2.3.5, Remote: 10.2.3.6
  Color: 0x4 backup
  Metric: 1
  Static BW: 155.52Mbps
  Reservable BW: 155.52Mbps
  Available BW [priority] bps:
      [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
      [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
  Interface Switching Capability Descriptor(1):
    Switching type: Packet
    Encoding type: Packet
    Maximum LSP BW [priority] bps:
      [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
      [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
To: 10.2.5.1-1, Local: 10.2.5.1, Remote: 0.0.0.0
  Color: 0x4 backup
  Metric: 1
  Static BW: 100Mbps
  Reservable BW: 100Mbps
  Available BW [priority] bps:
      [0] 100Mbps      [1] 100Mbps     [2] 100Mbps     [3] 100Mbps
```

```
          [4] 100Mbps      [5] 100Mbps     [6] 100Mbps     [7] 100Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 100Mbps      [1] 100Mbps     [2] 100Mbps     [3] 100Mbps
          [4] 100Mbps      [5] 100Mbps     [6] 100Mbps     [7] 100Mbps

user@router0>  show ted database 10.255.41.217 extensive
TED database: 0 ISIS nodes 8 INET nodes
NodeID: 10.255.41.217
  Type: Rtr, Age: 473 secs, LinkIn: 6, LinkOut: 6
  Protocol: OSPF(0.0.0.0)
    To: 10.255.41.216, Local: 10.2.3.2, Remote: 10.2.3.1
      Color: 0x2 fa
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    To: 10.255.41.216,  Local: 172.16.30.2, Remote: 172.16.30.1
      Metric: 1
      Static BW: 400kbps
      Reservable BW: 400kbps
      Available BW [priority] bps:
          [0] 370kbps      [1] 370kbps     [2] 370kbps     [3] 370kbps
          [4] 370kbps      [5] 370kbps     [6] 370kbps     [7] 370kbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 370kbps      [1] 370kbps     [2] 370kbps     [3] 370kbps
          [4] 370kbps      [5] 370kbps     [6] 370kbps     [7] 370kbps
    To: 10.255.41.216, Local: 10.2.3.6, Remote: 10.2.3.5
```

```
    Color: 0x4 backup
    Metric: 1
    Static BW: 155.52Mbps
    Reservable BW: 155.52Mbps
    Available BW [priority] bps:
        [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
        [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    Interface Switching Capability Descriptor(1):
      Switching type: Packet
      Encoding type: Packet
      Maximum LSP BW [priority] bps:
        [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
        [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
  To: 10.255.41.215, Local: 10.3.5.1, Remote: 10.3.5.2
    Color: 0x2 fa
    Metric: 1
    Static BW: 155.52Mbps
    Reservable BW: 155.52Mbps
    Available BW [priority] bps:
        [0] 155.12Mbps   [1] 155.12Mbps  [2] 155.12Mbps  [3] 155.12Mbps
        [4] 155.12Mbps   [5] 155.12Mbps  [6] 155.12Mbps  [7] 155.12Mbps
    Interface Switching Capability Descriptor(1):
      Switching type: Packet
      Encoding type: Packet
      Maximum LSP BW [priority] bps:
        [0] 155.12Mbps   [1] 155.12Mbps  [2] 155.12Mbps  [3] 155.12Mbps
        [4] 155.12Mbps   [5] 155.12Mbps  [6] 155.12Mbps  [7] 155.12Mbps
  To: 10.255.41.221, Local: 10.3.6.1, Remote: 10.3.6.2
    Color: 0x8 other
    Metric: 1
    Static BW: 155.52Mbps
    Reservable BW: 155.52Mbps
    Available BW [priority] bps:
        [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
        [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    Interface Switching Capability Descriptor(1):
      Switching type: Packet
      Encoding type: Packet
      Maximum LSP BW [priority] bps:
        [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
        [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
```

```
    To: 10.3.4.1-1, Local: 10.3.4.1, Remote: 0.0.0.0
      Color: 0x4 backup
      Metric: 1
      Static BW: 100Mbps
      Reservable BW: 100Mbps
      Available BW [priority] bps:
          [0] 100Mbps      [1] 100Mbps     [2] 100Mbps     [3] 100Mbps
          [4] 100Mbps      [5] 100Mbps     [6] 100Mbps     [7] 100Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 100Mbps      [1] 100Mbps     [2] 100Mbps     [3] 100Mbps
          [4] 100Mbps      [5] 100Mbps     [6] 100Mbps     [7] 100Mbps

user@router0>  show rsvp session name e2e_lsp_r0r5 extensive
Ingress RSVP: 1 sessions
10.255.41.221
  From: 10.255.41.222, LSPstate: Up, ActiveRoute: 2
  LSPname: e2e_lsp_r0r5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 101584
  Resv style: 1 FF, Label in: -, Label out: 101584
  Time left:    -, Since: Wed Sep  7 19:02:56 2005
  Tspec: rate 30kbps size 30kbps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 29458 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.1.2.2 (so-0/0/3.0) 15 pkts
  RESV rcvfrom: 10.1.2.2 (so-0/0/3.0) 16 pkts
  Explct route: 10.1.2.2 172.16.30.2 10.3.6.2
   Record route: <self> 10.1.2.2 172.16.30.2 10.3.6.2
Total 1 displayed, Up 1, Down 0

Egress RSVP: 1 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Router 1:

На маршрутизаторі 1 перевірили, чи працює наша конфігурація каналів зв'язку трафіку LMP, і що кінцевий LSP перетинає трафік, надавши команду керування каналами керування покажчиками. Також ми ввели велику команду show rsvp session, щоб підтвердити, що FA-LSP працює.

```
Peer name: r4 , System identifier: 10758
 State: Up, Control address: 10.255.41.217
   TE links:
   link_r1r4

 TE link name: link_r1r4, State: Up
  Local identifier: 16299, Remote identifier: 0, Local address: 172.16.30.1, Remote address: 172
  Encoding: Packet, Switching: Packet, Minimum bandwidth: 0bps, Maximum bandwidth: 400kbps,
  Total bandwidth: 400kbps, Available bandwidth: 370kbps
    Name        State Local ID  Remote ID      Bandwidth Used  LSP-name
    fa_lsp_r1r4 Up       22642          0        400kbps  Yes  e2e_lsp_r0r5

user@router1>  show link-management te-link name link_r1r4 detail
 TE link name: link_r1r4, State: Up
  Local identifier: 16299, Remote identifier: 0, Local address: 172.16.30.1,  Remote address: 17
  Encoding: Packet, Switching: Packet, Minimum bandwidth: 0bps, Maximum bandwidth: 400kbps,
  Total bandwidth: 400kbps, Available bandwidth: 370kbps
    Resource: fa_lsp_r1r4, Type: LSP, System identifier: 2147483683, State: Up, Local identifier
    Remote identifier: 0
    Total bandwidth: 400kbps, Unallocated bandwidth: 370kbps
    Traffic parameters: Encoding: Packet, Switching: Packet, Granularity: Unknown
     Number of allocations: 1, In use: Yes
      LSP name: e2e_lsp_r0r5,  Allocated bandwidth: 30kbps

user@router1>  show rsvp session name fa_lsp_r1r4 extensive
Ingress RSVP: 1 sessions
10.255.41.217
  From: 10.255.41.216, LSPstate: Up, ActiveRoute: 0
  LSPname: fa_lsp_r1r4, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100816
  Resv style: 1 FF, Label in: -, Label out: 100816
  Time left:    -, Since: Wed Sep  7 19:02:33 2005
  Tspec: rate 400kbps size 400kbps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 5933 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.2.4.2 (so-0/0/2.0) 28 pkts
  RESV rcvfrom: 10.2.4.2 (so-0/0/2.0) 26 pkts
  Explct route: 10.2.4.2 10.4.5.2 10.3.5.1
```

```
Total 1 displayed, Up 1, Down 0

Egress RSVP: 1 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 2 sessions
Total 0 displayed, Up 0, Down 0
```

Висновки

В розділі було продемонстровано приклад налаштування протоколу RSVP-TE для невеликої мережі з п’яти роутерів. Ми перевірили конфігурація каналів, проходження трафіку а також наявність створених та працюючих RSVP - сесій.