

Міністерство освіти і науки України
Державний вищий навчальний заклад
«Національний гірничий університет»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
дипломної роботи

магістра

(назва освітньо-кваліфікаційного рівня)

галузь знань	<u>0509 Радіотехніка, радіоелектронні апарати та зв'язок</u> (шифр і назва галузі знань)
напрямок підготовки	<u>050903 Телекомунікації</u> (код і назва напрямку підготовки)
спеціальність	<u>8.05090302 Телекомунікаційні системи та мережі</u> (код і назва спеціальності)
освітній рівень	<u>магістр</u> (назва освітнього рівня)
кваліфікація	<u>2144 Професіонал в галузі електроніки та телекомунікацій,</u> <u>дослідник</u> (код і назва кваліфікації)

на тему: «Дослідження аналітичної моделі активного обладнання інформаційної телекомунікаційної мережі»

Виконавець: студент 5 курсу, групи 172м-16-1

Шапошников Максим Ігорович

(підпис)

(прізвище ім'я по-батькові)

Керівники	Прізвище, ініціали	Оцінка	Підпис
роботи	к.ф.-м.н., доц. Гусев А.Ю.		
розділів:			
спеціальний	к.ф.-м.н., доц. Гусев А.Ю.		
економічний	к.е.н., доц. Романюк Н.М.		

Рецензент			
-----------	--	--	--

Нормоконтроль	доц. Гусев А.Ю.		
---------------	-----------------	--	--

Дніпропетровськ
2018

Міністерство освіти і науки України
Державний вищий навчальний заклад
«Національний гірничий університет»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

« » _____ 20__ року

ЗАВДАННЯ
на виконання кваліфікаційної роботи магістра
спеціальності _____ *8.05090302 Телекомунікаційні системи та мережі*
(код і назва спеціальності)

студенту _____ **172М-16-1**
(група)

_____ **Шапошников Максим Ігорович**
(прізвище ім'я по-батькові)

Тема дипломної роботи _____ *«Дослідження аналітичної моделі активного обладнання інформаційної телекомунікаційної мережі».*

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора Державного ВНЗ «НГУ» від 27.04.2018 № 673-л.

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Об'єкт досліджень _____ *технологія багатопроTOCOLьної комутації по міткам.*

Предмет досліджень _____ *технологія організації VPN для віддалених мереж.*

Мета НДР _____ *аналіз активного обладнання інформаційних мереж на основі технології MPLS*

Вихідні дані для проведення роботи _____ *результати та матеріали з виробничої, переддипломної практики та курсового проекту з телекомунікаційних систем та інформаційних мереж.*

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна полягає в розробці оптимальної моделі VPN з використанням багатопроTOCOLьної комутації по міткам, для збільшення швидкості просування.

Практична цінність Підвищення ефективності роботи активного обладнання VPN.

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати досліджень мають бути подані у вигляді, що дозволяє безпосереднє використання для підвищення роботи активного обладнання інформаційної телекомунікаційної мережі.

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Проаналізувати технології моделювання активного обладнання інформаційних мереж.	19.10.15-07.11.17
Побудувати аналітичну модель активного обладнання на основі технології MPLS, для віддалених мереж.	09.11.15-28.12.17
Виконати порівняльну оцінку IP-маршрутизації та технології маршрутизації по міткам MPLS.	04.04.15-20.05.17
Вибрати методи і засоби реалізації проектного рішення, розрахувати витрати на створення програмного продукту.	23.05.16-6.06.17

6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект очікується позитивним завдяки зменшенню часу на доставку фреймів у інформаційно-телекомунікаційних мережах.

7 ДОДАТКОВІ ВИМОГИ

Відповідність оформлення ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення.

Завдання видав _____
(підпис)

к.ф.-м.н., доц. Гусєв А.Ю.
(прізвище, ініціали)

Завдання прийняв
до виконання _____
(підпис)

Шапошников М.І.
(прізвище, ініціали)

Дата видачі завдання: _____

Термін подання дипломного проекту до ДЕК 10.06.2017

РЕФЕРАТ

Пояснювальна записка: с., рис., табл., джерел.

Об'єкт дослідження: технологія багатопротокольної комутації по міткам.

Предмет дослідження: технологія організації VPN для віддалених мереж.

Мета дипломної роботи: аналіз активного обладнання інформаційних мереж на основі технології MPLS.

Наукова новизна результатів, що очікуються, полягає в розробці оптимальної моделі VPN з використанням, багатопротокольної комутації по мітках, яка дозволяє збільшити швидкість проходження пакетів.

В першому розділі проведено аналіз технологій моделювання активного обладнання та порівняно методи дослідження моделей активного обладнання на віддалених мережах.

В другому розділі розглянуті та обрані методи створення аналітичних моделей активного обладнання.

В третьому розділі розроблена аналітична модель VPN для віддалених мереж, використано технологію MPLS. Проведений порівняльний аналіз обробки пакетів в IP і MPLS мережах, проведено аналіз пакетів за допомогою програми Wireshark.

В економічному розділі проведено вибір методів і засобів реалізації проектного рішення, визначено трудомісткість розробки та опрацювання програмного продукту, розраховано витрати на створення програмного продукту.

MULTIPROTOCOL LABEL SWITCHING, МОДЕЛЮВАННЯ, NETWORK SIMULATOR 3, МАРШРУТИЗАЦІЯ, МОНІТОРИНГ ТРАФІКУ.

ABSTRACT

Explaining message: items, figs, tabl., sour.

An object of development multiprotocol label switching technology.

Subject of research: organization technology VPN for remote networks.

Purpose of diploma project: active equipment analysis for information networks based on MPLS technology.

Scientific novelty of the results that are expected, is to develop a suitable VPN model using multiprotocol label switching, which allows to increase packet transmission rate.

In the first section active equipment technologies were analyzed and research methods for active equipment modeling were compared.

In the second section: methods for constructing active equipment models were considered and choosen.

In the third section: VPN analytic model for remote networks was developed, MPLS technology was used. Packet processing comparative analysis in IP and MPLS network was performed, and traffic was analyzed using Wireshark software.

In the economic section: methods and technics were chosen for the project design implementation, software processing and development complexity was found, and software cost was calculated.

MULTIPROTOCOL LABEL SWITCHING, SIMULATION, NETWORK SIMULATOR 3, ROUTING, TRAFFIC MONITORING.

РЕФЕРАТ

Пояснительная записка с., рис., табл., источников.

Объект исследования: технология многопротокольной коммутации по меткам.

Предмет исследования: технология организации VPN для удаленных сетей.

Цель дипломной работы: анализ активного оборудования информационных сетей на основе технологии MPLS.

Научная новизна результатов, что ожидаются, заключается в разработке оптимальной модели VPN с использованием многопротокольной коммутации по меткам, которая позволяет увеличить скорость прохождения пакетов.

В первом разделе проведен анализ технологий моделирования активного оборудования и проведено сравнение методов исследования моделей активного оборудования для удаленных сетей.

Во втором разделе рассмотрены и выбраны методы создания аналитических моделей активного оборудования.

В третьем разделе разработана аналитическая модель VPN для удаленных сетей, использовано технологию MPLS. Проведен сравнительный анализ обработки пакетов в IP и MPLS сетях, проведен анализ пакетов с помощью программы Wireshark.

В экономическом разделе проведен выбор методов и средств реализации проектного решения, определены трудоемкость разработки и обработки программного продукта, рассчитаны затраты на создание программного продукта.

MULTIPROTOCOL LABEL SWICHING, МОДЕЛИРОВАНИЕ, GLOBAL NETWORK SIMULATOR 3, МАРШРУТИЗАЦИЯ, МОНИТОРИНГ ТРАФИКА.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

ATM – асинхронний спосіб передачі даних

BGP – протокол граничного шлюзу

CoS – клас обслуговування

EIGRP – покращений протокол маршрутизації внутрішнього шлюзу

FEC – клас еквівалентності при передачі

FTP – протокол передачі файлів

GNS3 – графічний симулятор мереж

IGRP – протокол маршрутизації в автономних системах

IPDC – Інтернет протокол відправки даних

ISDN – цифрова мережа з інтеграцією служб

IS-IS – протокол маршрутизації проміжних систем

LDP – протокол распределения информации о привязке меток к FEC

LER - прикордонний маршрутизатор з комутацією по мітках

LGPL – стандартна громадська ліцензія обмеженого застосування

LSP – шлях с комутацією по міткам

MGCP – протокол контролю медіашлюзів

MPLS – багатопрокольна комутація по мітка

OSI– взаємодія відкритих систем

OSPF – протокол динамічної маршрутизації

RIP – протокол Інтернет маршрутизації

RTP – протокол передачі мультимедійних файлів в реальному часі

SDH – синхронна цифрова ієрархія

TCP/IP – набір мережевих протоколів передачі даних, що використовуються в мережах, включаючи мережу Інтернет

UDP – протокол користувальницьких датаграм

ЗМІСТ

ВСТУП.....	
РОЗДІЛ 1. ТЕХНОЛОГІЇ ДОСЛІДЖЕННЯ АКТИВНОГО ОБЛАДНАННЯ ІНФОРМАЦІЙНОЇ ТЕЛЕУОСЦІНКАЦІЙНОЇ МЕРЕЖІ.	
1.1 Багатопротокольна комутація по міткам	
1.2 Класи еквівалентного обслуговування (FEC)	
1.3 Мережеві вузли MPLS	
1.4 Шлях комутації по мітках (LSP).....	
1.5 Мітки та способи маркування	
1.6 Стек міток.....	
1.7 Таблиці пересилання.....	
1.8 Прив'язка мітка – FEC.....	
1.9 Вибір маршруту	
1.10 Моделювання комп'ютерних мереж.....	
Висновки до першого розділу	
РОЗДІЛ 2. МЕТОДИ ОПТИМІЗАЦІЇ АКТИВНОГО ОБЛАДНАННЯ НА ОСНОВІ ТЕХНОЛОГІЇ MPLS	
2.1.Методи оптимізації	
2.2.Основні принципи імітаційного моделювання	
2.3.Структура імітаційних моделей.....	
2.4.Алгоритм імітаційного моделювання	
2.5 Етапи імітаційного моделювання активного обладнання.....	
2.6 Порівняння алгоритмів пошуку оптимального дизайну	
Висновки до другого розділу	

РОЗДІЛ 3. ЕКСПЕРЕМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МОДЕЛІ ВІДДАЛЕНИХ МЕРЕЖ НА ОСНОВІ КОМУТАЦІЇ ПО МІТКАМ	
3.1 Дослідження активного обладнання	
3.2 Побудова мережі	
3.3 Порівняльна оцінка IP-маршрутизації та технології маршрутизації по міткам MPLS.....	
Висновки до третього розділу.....	
4 ЕКОНОМІЧНА ЧАСТИНА.....	
4.1 Вибір методів і засобів реалізації проектного рішення	
4.2 Складові вартості ПЗ.....	
4.3 Визначення трудомісткості розробки та опрацювання програмного продукту	
4.4 Розрахунок витрат на створення програмного продукту	
Висновки до четвертого розділу.....	
ВИСНОВКИ.....	
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	
ДОДАТОК А.....	
ДОДАТОК Б.....	
ДОДАТОК В.....	
ДОДАТОК Г.....	

ВСТУП

Актуальність. Зростання інформатизації суспільства посилює вимоги до надійності та продуктивності передачі даних в інформаційних телекомунікаційних мережах. Збільшення потоків даних призводить до виникнення пікових навантажень, при яких спостерігаються значні втрати пакетів та істотні часові затримки.

MPLS являє собою технологію, що визначає напрямок розвитку майбутніх IP-мереж, враховуючи мережу Internet. Технологія комутації по міткам є універсальним рішенням проблем QoS, що виникають перед сьогоdnішніми пакетними мережами, рішенням, яке забезпечує швидкість передачі, масштабованість, оптимізацію розподілення трафіку та ефективну маршрутизацію в пакетних мережах [10].

Тепер постачальнику послуг недостатньо просто надавати доступ до своєї IP-магістралі. Змінені потреби користувачів включають в себе і доступ до інтегрованих сервісів мережі, і організацію віртуальних приватних мереж (VPN), і ряд інших інтелектуальних послуг. Зростаючий попит на додаткові послуги, що реалізуються поверх простого IP-доступу, обіцяє принести Internet-провайдерам величезні доходи.

Для вирішення виникаючих завдань і була розроблена архітектура MPLS, яка забезпечує побудову мереж, що мають практично необмежені можливості масштабування, підвищену швидкість обробки трафіку і безпрецедентну гнучкість з точки зору організації додаткових сервісів. Крім того, технологія MPLS дозволяє інтегрувати мережі IP і ATM, за рахунок чого постачальники послуг зможуть не тільки зберегти кошти, інвестовані в обладнання асинхронної передачі, а й отримати додаткову вигоду зі спільного використання цих протоколів.

Метою дипломної роботи є аналіз активного обладнання інформаційних мереж на основі технології MPLS.

Для досягнення цієї мети необхідно вирішити наступні задачі:

- 1) Проаналізувати технології моделювання активного обладнання.

2) Побудувати аналітичну модель активного обладнання на основі технології MPLS, для віддалених мереж.

3) Виконати порівняльну оцінку IP-маршрутизації та технології маршрутизації по міткам MPLS.

Об'єктом дослідження є технологія багатопротокольної комутації по міткам.

Предметом дослідження є технологія організації VPN для віддалених мереж.

Методи досліджень. Для дослідження моделей активного обладнання були використані засоби імітаційного моделювання, а саме графічний симулятор мереж GNS3, який дозволяє відтворювати реальні мережі за допомогою технології віртуалізації активного обладнання. Для перевірки роботи моделей були використані програмні засоби для моніторингу трафіку (Wireshark).

Наукова новизна результатів, що очікуються, полягає в розробці оптимальної моделі VPN з використанням, багатопротокольної комутації по мітках, яка дозволяє збільшити швидкість проходження пакетів.

У ході вирішення поставлених задач отримано такі результати:

1) Розроблений алгоритм налаштування мережевого обладнання з використанням технології L2 VPN over MPLS, для досягнення необхідного рівня безпеки трафіка при проходженні через мережу Інтернет. Працездатність алгоритму представлена за допомогою програми Wireshark, яка візуалізує рух пакетів через мережу MPLS.

2) На створеній моделі було проведено порівняння IP та MPLS маршрутизаторів. Використання аналітичної моделі дозволяє кількісно оцінити величину ефекту від впровадження технології MPLS над IP-маршрутизацією. Це пояснюється тим, що за рахунок зменшення операцій при обробці міток в разі перегляду таблиць просування (MPLS) в порівнянні обробкою пакетів і переглядом таблиць маршрутизації (IP-маршрутизація) збільшується швидкість просування пакетів.

Практичне значення отриманих результатів полягає в підвищенні ефективності роботи активного обладнання VPN.

РОЗДІЛ 1

ТЕХНОЛОГІЇ ДОСЛІДЖЕННЯ АКТИВНОГО ОБЛАДНАННЯ ІНФОРМАЦІЙНОЇ ТЕЛЕУОСЦІКАЦІЙНОЇ МЕРЕЖІ

1.1 Багатопротокольна комутація по міткам

Одним з перспективних напрямків побудови сучасної мережевої інфраструктури є використання оптичних технологій для організації високошвидкісної магістральної мережі і єдиної системи сигналізації, що дозволяє об'єднувати різні типи середовищ і систем передачі інформації. Як такої об'єднуючої технології зараз розглядається технологія багатопротокольної комутації по мітках (MultiProtocol Label Switching, MPLS).

Дана технологія є спробою прискорити просування IP-пакетів і зберегти гнучкість, характерну для IP мереж, за допомогою механізмів управління трафіком і підтримки якості обслуговування, що застосовуються в мережах АТМ. Впровадження технології MPLS дозволяє зберегти все краще, що властиво архітектурі IP-over-АТМ (ефективне мультиплексування і гнучкість трафіку, висока продуктивність), і при цьому вона ще більше підвищує масштабованість мереж, спрощує їх побудова та експлуатацію. Важливо і те, що MPLS може використовуватися не тільки з АТМ, але і з будь-якою іншою технологією канального рівня. MPLS використовує і розвиває концепцію віртуальних каналів, що застосовуються в мережах X.25, Frame Relay, об'єднуючи її з технікою вибору шляхів на основі інформації про топології і поточне завантаження мережі, отриманої за допомогою протоколів маршрутизації мереж IP. Такий підхід спрощує перехід до наступного покоління волоконно-оптичних магістралей Інтернет на основі технологій SDH / WDM або IP / WDM [1–5].

MPLS – це технологія швидкої комутації пакетів в багато протокольних мережах, заснована на використанні міток. Концепція MPLS поєднує в собі управління трафіком, характерне для технологій канального рівня, масштабованість і гнучкість протоколів мережевого рівня. «Багатопротокольних» в назві технології

означає, що MPLS є інкапсулюють протоколом і може транспортувати безліч інших протоколів, як показано на рис. 1.1.

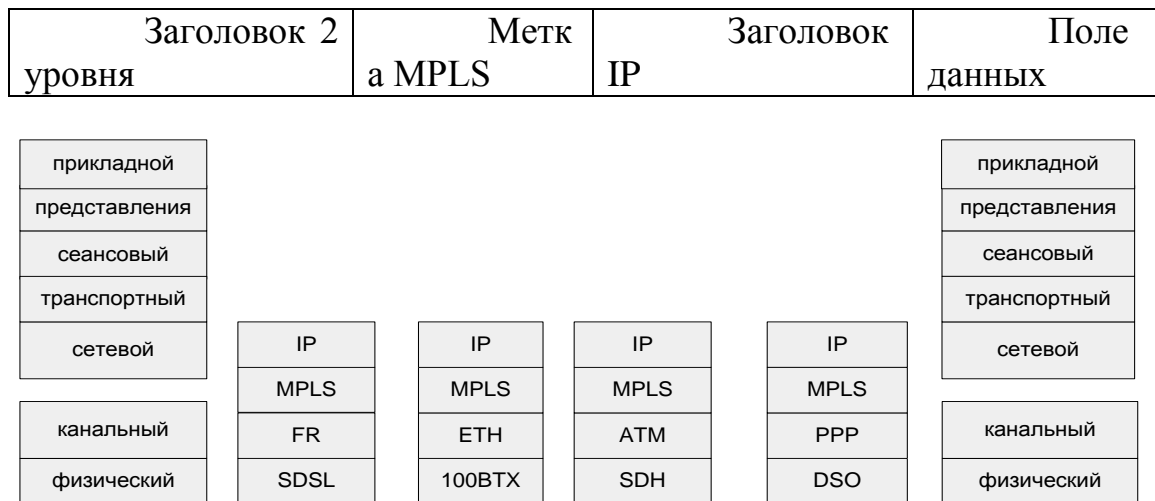


Рисунок 1.1 – Технологія MPLS в IP мережах і модель OSI \ ISO

Фізичний рівень містить функції, що забезпечують використання фізичної середовища для двосторонньої передачі бітів (з такою достовірністю, яку забезпечує це середовище) по прямому тракту, що зв'язує два вузла мережі. Другий рівень (канальний рівень) містить функції, що забезпечують формування в цьому тракті надійного логічного ланки зв'язку, за яким відбувається двосторонній обмін інформаційними блоками між вузлами. На даному рівні виявляються, і виправляються помилки, і гарантується достовірність передачі. Третій рівень (мережевий рівень) містить функції, які забезпечують транспортування інформаційних блоків від відправника до одержувача через кілька вузлів мережі по невласивому маршруту транспортування, який складається з ланок другого рівня.

Представлена модель на рис. 1.1 показує, що протокол MPLS не утворює повноцінного рівня, а «вклинюється» в мережі IP, ATM або Frame Relay між канальним і мережевим рівнями моделі OSI, залишаючись незалежним від цих рівнів. Можна сказати, що одночасне функціонування MPLS на мережевому рівні і на канальному рівні призводить до утворення так званого рівня 2.5, де і виконується комутація по мітках.

Мережі ряду Інтернет-провайдерів побудовані сьогодні на основі багаторівневої моделі, припускає, що логічна маршрутизована IP-мережа

функціонує поверх комутованої топології другого рівня (ATM, або Frame Relay) і незалежно від неї. Комутатори другого рівня забезпечують високошвидкісні з'єднання, в той час як IP-маршрутизатори на периферії мережі, пов'язані один з одним мережею віртуальних каналів другого рівня, здійснюють інтелектуальну пересилання IP-пакетів.

Таким чином, MPLS – це один з кроків на шляху еволюційного розвитку мережі Інтернет в бік спрощення її інфраструктури, шляхом інтеграції функцій другого (комутація) та третього (маршрутизація) рівнів.

Як видно з рис. 1.1 MPLS – універсальна технологія. З її допомогою можна вирішувати такі завдання:

- інтеграція ATM і Frame Relay з IP;
- прискорене просування пакетів всередині мережі оператора уздовж традиційних найкоротших маршрутів;
- створення віртуальних приватних мереж (VPN);
- вибір і встановлення шляхів з керуванням трафіку (Traffic Engineering, TE).

В даний час досить активно розвивається стандарт GMPLS (Generalized MultiProtocol Label Switching), призначений для вибору і встановлення оптичних шляхів в мережах SDH і WDM.

1.2 Класи еквівалентного обслуговування (FEC)

Класом еквівалентного обслуговування (Forwarding Equivalency Class, FEC) – називається група пакетів 3-го рівня, наприклад IP-пакетів, які однаково обслуговуються і пересилаються.

Термін FEC застосовують для операцій комутації по мітках. При використанні технології MPLS відповідність між пакетом і «класом еквівалентного обслуговування» FEC встановлюється один раз, на вході в мережу MPLS. Цьому FEC присвоюється мітка – ідентифікатор фіксованої довжини, який подається із пакетом до наступного маршрутизатора. Завдяки цьому в інших маршрутизаторах заголовок мережевого рівня не аналізується. Мітка, встановлена прикордонним маршрутизатором при вході пакета в MPLS-мережа, використовується як покажчик

входу таблиці, яка визначає черговий маршрутизатор для пересилання до нього пакету, а також нову мітку для FEC, до якого належить пакет. До одного FEC можуть ставитися пакети всіх потоків, якщо шляху їх слідування через мережу (або частина мережі) збігаються. З точки зору вибору найближчого маршрутизатора, до якого їх необхідно переслати, всі пакети одного FEC невиразні.

Клас еквівалентного обслуговування (FEC) дозволяє об'єднувати велику кількість потоків трафіку, що вимагають однакової обробки. Об'єднані в клас еквівалентності FEC потоки трафіку ідентифікуються однією і тією ж MPLS-міткою. Можливість об'єднання потоків трафіку незалежно від адреси мереж призначення значно збільшує можливість MPLS до масштабування за рахунок зменшення обсягу інформації про маршрути, зберігається та обробляється маршрутизаторами з комутацією по мітках (LSR-маршрутизаторами).

Прикладом FEC можуть служити всі IP-пакети з адресою кінцевого одержувача трафіку, наприклад хост-машини.

Використання FEC дозволяє:

1) Об'єднувати пакети в класи. При такому об'єднанні, значення FEC в пакеті може використовуватися для установки пріоритетів. При обробці пакетів, надається більш високий пріоритет одним пакетам по відношенню до інших.

2) Забезпечити підтримку ефективних операцій QoS. Наприклад, FEC можуть бути пов'язані з високопріоритетним голосовим трафіком в реальному часі, низькопріоритетним трафіком Інтернет-конференцій, і так далі.

FEC пакета може визначатися за одним або за кількома параметрами, зазначеними мережевим адміністратором. Серед можливих параметрів можна назвати:

- IP-адреса відправника та / або одержувача або IP-адреси мереж;
- номери портів відправника і / або одержувача;
- ідентифікатор IP-протоколу;
- код диференційованої служби;
- мітка потоку IPv6.

Для різних класів обслуговування використовуються різні FEC і пов'язані з ними мітки. У мережах MPLS можливі два підходи до пересилання пакетів з урахуванням класу обслуговування.

Перший підхід передбачає обробку пакетів в вихідних чергах маршрутизаторів з урахуванням значень пріоритету, зазначеного в заголовку MPLS.

Другий підхід базується на тому, що для кожної пари, що складається з вхідного і вихідного маршрутизаторів, визначається кілька шляхів комутації по мітках (Label Switched Path, LSP) с різними характеристиками продуктивності, смуги пропускання, часу затримки та інших параметрів. Після цього вхідний маршрутизатор направляє один тип трафіку по одному шляху, інший – по іншому, третій – по третьому і т.д.

Кожен маршрутизатор мережі MPLS створює таблицю, яка називається інформаційною базою міток (Incoming Label Mapping, ILM), за допомогою якої визначає, яким чином повинен пересилатися пакет. Така таблиця містить безліч міток що використовуються і для кожної з них містить прив'язку «FEC-мітка». Для її створення використовується таблиця переадресації помічених пакетів (Next Hop Label Forwarding Entry, NHLFE). Проводиться обмін старої мітки на нову, після чого пакет далі пересилається з новою міткою. Така процедура отримала назву "обмін міток". Одна вхідна мітка може змінюватися на кілька вихідних міток. Метод FEC є одним з основних компонентів MPLS, який визначає багато в чому роботу всієї мережі.

Таким чином, клас еквівалентного обслуговування FEC є формою подання групи пакетів з однаковими вимогами до напрямку їх передачі, тобто всі пакети в такій групі обробляються в маршрутизаторі однаково і однаково йдуть до пункту призначення.

При створенні мережі потрібно звернути увагу, щоб число класів обслуговування було оптимальним для реалізації всіх важливих додатків і необхідних параметрів якості.

1.3 Мережеві вузли MPLS

У мережах MPLS використовуються два види мережевих вузлів. Розташовані на кордоні мережі MPLS маршрутизатори повинні розпізнавати і аналізувати IP-потоки що надходять і направляти їх за відповідними маршрутами. Такі пристрої називаються прикордонними маршрутизаторами з комутацією по мітках (Label Edge Router, LER). Прикордонні маршрутизатори 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. Розрізняють вхідний і вихідний LER.

Вхідний LER аналізує IP-заголовок як і звичайний маршрутизатор і встановлює до якого FEC при виборі адреси наступної передачі пакета він належить. Такому FEC присвоюється мітка, яка передається разом з пакетом до транзитного маршрутизатора.

IP-дейтаграмма полягає в модуль даних протоколу (Protocol Data Unit, PDU) технології MPLS, а заголовок MPLS прикріплюється до дейтаграми. Якщо заголовок об'єднаний з операцією QoS (наприклад, DiffServ), то вхідний LER розглядатиме трафік відповідно до правил DiffServ. Далі LER приймає рішення про вибір шляху для даного пакета, посилаючи його до відповідного транзитному маршрутизатора з комутацією по мітках (Label Switch Routers, LSR) [16].

Транзитний LSR отримує PDU і використовує заголовок MPLS для прийняття рішень пересилання. Він також робить заміну міток. Даний LSR не займається обробкою заголовка третього рівня (IP-заголовка), а приймає рішення про пересилку на основі мітки пакета, а не на основі таблиці маршрутизації і пересилає пакет далі.

Далі, проходячи зазвичай через кілька транзитних LSR, пакет потрапляє до вихідного LER, який виробляє операцію розбирання PDU, видаляє з пакету мітку, аналізує заголовок пакету і направляє його до адресата, що знаходиться зовні MPLS-мережі (рис. 1.2).

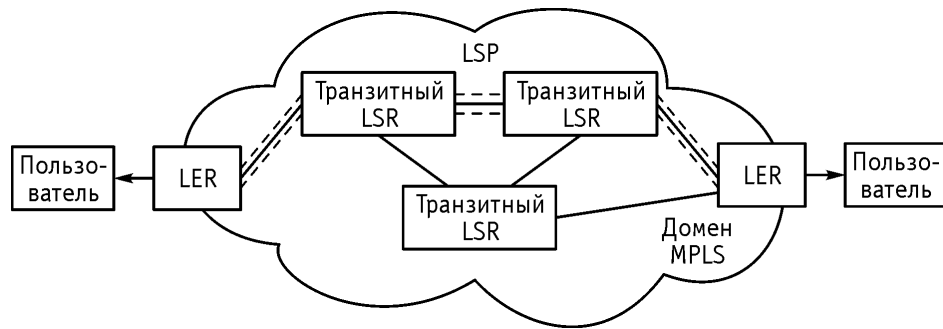


Рисунок 1.2 – Елементи мережі MPLS

1.4 Шлях комутації по мітках (LSP)

Трафік одного FEC перетинає домен MPLS по LSP. Для визначення топології і поточного стану домену потрібно протокол маршрутизації, що дозволяє кожному FEC призначати конкретний LSP. Протокол маршрутизації повинен бути здатний збирати і використовувати інформацію для підтримки вимог до якості обслуговування даного FEC. Окремі маршрутизатори повинні знати про LSP даного FEC, повинні призначати LSP для вхідної мітки. А також повинні обмінюватися цією міткою з усіма іншими маршрутизаторами, які можуть послати їм пакети даного FEC. Таким чином маршрут пакета в мережі MPLS визначається тим FEC, який встановлений для цього потоку в усіх LER. Такий шлях отримав назву Шляхи комутації по мітках (Label Switched Path, LSP) і ідентифікується послідовністю LSR, розташованих на шляху прямування пакета від одержувача до відправника. Мітки в LSP призначаються за допомогою протоколів розподілу міток (LDP, RSVP). LSP можна класифікувати наступним чином:

- Між двома граничними LER домену MPLS проходить один маршрут.
- Один вихідний LER, кілька вхідних маршрутизаторів. Призначений одному FEC трафік може надходити від різних джерел через різні вхідні LER. Прикладом такої ситуації є корпоративна Інтернет-мережа, розташована в одному регіоні, але з доступом до домену MPLS за коштами декількох вхідних LER. У такій ситуації через домен MPLS проходить кілька маршрутів, можливо, з загальними кінцевими ретрансляційних ділянками.
- Кілька вихідних маршрутизаторів для трафіку цільової розсилки. В рекомендації RFC 3031 стверджується, що найчастіше пакету присвоюється FEC на

основі (частково або повністю) адреси одержувача мережевого рівня. В іншому випадку для FEC потрібні маршрути до декількох різних вихідних маршрутизаторів. Однак, швидше за все, існує кілька мереж, в які трафік може бути доставлений через один вихідний LSR.

В рекомендації RFC 3031 групова розсилка згадується як предмет подальших досліджень.

LSP організується або перед передачею даних (з управлінням від програми), або при виявленні певного потоку даних (керовані даними).

Технологія MPLS підтримує два варіанти створення LSP:

1) Послідовна маршрутизація по транзитним ділянках маршруту (hop-by-hop routing). Кожен LSR самостійно вибирає наступну ділянку маршруту для даного FEC. Пакет, що належить одному FEC, проходить шлях від вхідного до вихідного LER через безліч транзитних LSR, утворюючи віртуальний тракт або шлях комутації по мітках, і ідентифікується послідовністю міток в LSR. Схожий метод застосовується в даний час в IP-мережах. LSR використовує протоколи маршрутизації OSPF, IS-IS.

2) Явна маршрутизація (ER) – подібна до методу маршрутизації з боку відправника. Вхідний LER (тобто LER, від якого виходить потік даних в мережі MPLS) визначає послідовність вузлів, через які проходить ER-LSP. Такий тракт може виявитися не оптимальним. Для забезпечення заданої якості обслуговування трафіку даних уздовж тракту можуть резервуватися ресурси. Такий підхід полегшує оптимальний розподіл трафіку уздовж мережі і дозволяє надати диференційоване обслуговування потокам трафіку різних класів, сформованих на основі прийнятих правил і методів управління мережею.

Встановлене з'єднання є симплексним. Для організації напівдуплексного з'єднання повинні бути встановлені два LSP. LSP завжди починається на кордоні мережі і закінчується на протилежному кінці, проходячи через кілька транзитних маршрутизаторів.

Розглянемо MPLS-мережу, яка складається з декількох граничних і транзитних маршрутизаторів (рис. 1.3).

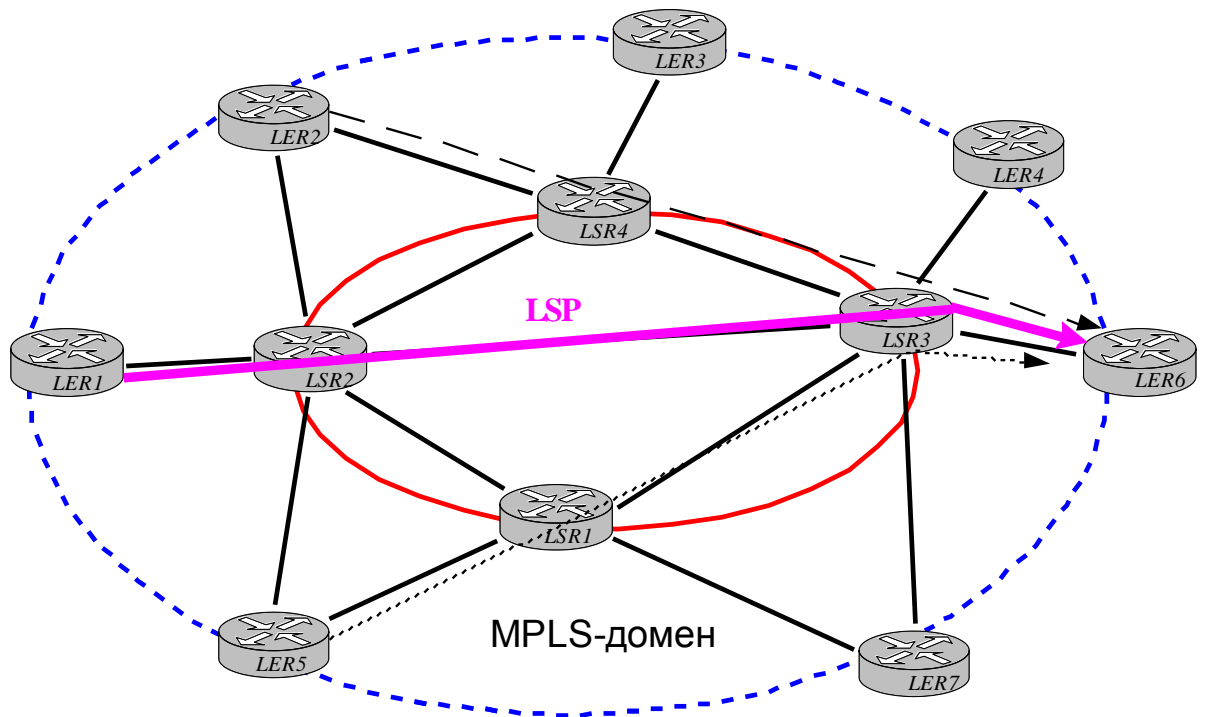


Рисунок 1.3 – Приклад домену MPLS-мережі

На вихідний вузол LER 6 надходять потоки пакетів від декількох вхідних вузлів LER1, LER5 і LER2. У транзитних маршрутизаторах деякі з цих потоків можуть об'єднуватися в один загальний потік пакетів. Такому потоку присвоюється загальний FEC. Таким чином, безліч трактів LSP, що йдуть до одного вихідного вузла LER6, утворює розгалужені дерево, коріння якого знаходиться у вихідному вузлі. Кожен з семи прикордонних вузлів виконує функції вхідного і вихідного вузла. Таким чином, на рис. 1.3 в мережі MPLS існує сім дерев такого роду, які разом містять $7 * (7-1) = 42$ LSP. Через один транзитний маршрутизатор може проходити кілька LSP, що належать різним деревам.

1.5 Мітки та способи маркування

Мітка є короткий ідентифікатор фіксованої довжини, що використовується на локальній ділянці мережі. Мітка призначена для визначення класу еквівалентного обслуговування пакета при його пересиланні по мережі. На сьогоднішній день стандартом визначено формат 32-бітної мітки, що розташовується між заголовками

канального і мережевого рівня. Розглянемо включення мітки в IP-пакет заголовка Ethernet (рис. 1.4).

Поле «Мітка» складається з 20 біт і містить власне значення мітки, що використовується для визначення наступного маршрутизатора, тобто для просування пакетів.

Поле cos (Class of Service) необхідно для надання диференційованих послуг в MPLS-мережі. Для наскрізного забезпечення QoS на кордоні MPLS-мережі можна скопіювати поле IP-пріоритету в поле CoS. Дане поле складається з 3 біт. Таким чином, в ньому може передаватися тільки 3-бітове поле IP-пріоритету, а 6-бітового поля диференційованої послуги (Differentiated Services Code Point, DSCP) – немає. При необхідності CoS може передаватися у вигляді однієї з міток MPLS-стека. Поле мітки здатне вмістити як поле IP-пріоритету, так і поле DSCP.

Поле S призначений для підтримки ієрархічного стека міток. Біт S встановлюється в одиницю для останньої мітки в стеці і в нуль для всіх інших міток стека. Це дозволяє прив'язати префікс до кількох міток, іншими словами – до стека міток.

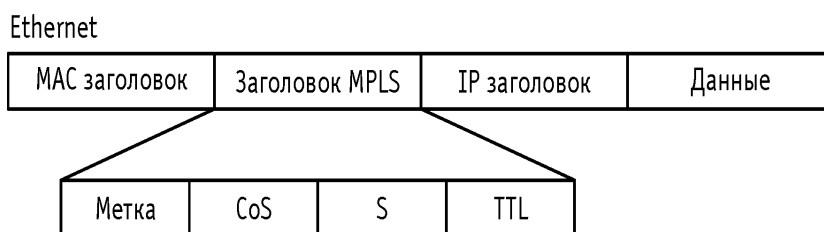


Рис 1.4 – Формат мітки MPLS

Кожна мітка стека має свої власні значення поля CoS, S-біта і поле TTL.

Поле TTL (Час життя) містить 8 біт, що використовуються для кодування кількості транзитних ділянок.

Поле «Час життя» є ключовим полем в заголовку IP-пакета. Зазвичай в об'єднаній IP-мережі це поле зменшується на одиницю на кожній ділянці маршруту, і пакет відкидається при досягненні нуля значення лічильника. Такий підхід необхідний для того, щоб уникнути зациклення пакета або занадто довгого

перебування пакета в об'єднаній мережі через невірну маршрутизації. Оскільки LSR не досліджує IP-заголовок, то поле TTL включається в мітку, що дозволяє зберегти функціональність цього поля.

Правила обробки поля TTL в мітці:

1) Коли IP-пакет прибуває на вхідний прикордонний маршрутизатор домену MPLS, в стек пакета поміщається одна мітка. Значення поля TTL цієї мітки встановлюється рівним значенню поля TTL IP-заголовка.

2) Коли пакет MPLS прибуває на черговий транзитний маршрутизатор домену MPLS, тоді значення поля TTL в мітці, що знаходиться на вершині стека, зменшується на одиницю.

Якщо вийшло значення TTL нульове, тоді MPLS-пакет далі не передається. Залежно від значення мітки в стеці, пакет або просто відкидається, або передається відповідному «звичайному» мережному рівню для обробки помилок (наприклад, для формування повідомлення про помилку з використанням протоколу ICMP – Internet Control Message Protocol).

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

Коли MPLS-пакет прибуває на вихідний прикордонний маршрутизатор домену MPLS, значення поля часу, єдиної в стеку записи, зменшується на одиницю. Далі мітка вилучається із стека і стек міток стає порожнім. В цьому випадку пакет видається користувачеві, або на мережевий рівень для обробки помилок.

Якщо вийшло значення позитивне, то воно поміщається в поле TTL IP-заголовка, після чого IP-пакет перенаправляється далі шляхом звичайної маршрутизації. До того як переправити пакет далі необхідно перерахувати заново контрольну суму IP-заголовка. Така процедура необхідна для перевірки цілісності заголовка при пересиланні повідомлення.

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

1.6 Стек міток

В рамках архітектури MPLS разом з пакетом дозволено передавати не одну мітку, а кілька. При цьому розрізняють верхні і нижні мітки:

- нижня мітка буде оброблятися самою останньою по шляху проходження пакету;
- верхня мітка обробляється найпершою по шляху проходження пакету.

Операції додавання/вилучення мітки визначені як операції на стеку. Результат комутації задає лише верхня мітка стека, а нижні ж передаються прозоро до операції вилучення верхньої. Такий підхід дозволяє створювати ієрархію потоків в мережі MPLS і організувати тунельні передачі. Стек складається з довільного числа заголовків. Якщо стек міток має глибину m , то вважається, що сама нижня мітка розміщена на рівні 1, мітка над нею має рівень 2 і т.д., а мітка нагорі стека має рівень m . Верхня мітка в стеку знаходиться ближче до заголовку мережевого рівня, а нижня мітка розташовується ближче до заголовку каналного рівня. CoS в стеках не використовуються. Мітка може приймати будь-яке значення, крім декількох зарезервованих.

Пакет мережевого рівня рухається відразу за записом стека з встановленим в одиницю бітом S. Записи стека міток розташовуються після заголовка рівня передачі даних (каналного рівня), але до заголовків мережевого рівня. У кадрі протоколу передачі даних (рис. 1.5 а), наприклад протоколу PPP (Point-to-Point Protocol – протокол точка-точка), стек міток розташовується між IP-заголовком і заголовком рівня передачі даних.

У кадрі мережі стандарту IEEE 802 (рис. 1.5б) стек міток розташовується між IP-заголовком і заголовком рівня LLC (Logical Link Control – управління логічним з'єднанням).

Якщо архітектура MPLS використовується поверх орієнтованої на з'єднання мережевої служби, то можливе застосування іншого підходу, який ілюструються рис. 1.5, в, г [9].

В осередках ATM верхня мітка поміщається в поле VPI/VCI в заголовку осередки ATM. Верхня мітка залишається на вершині стека, що вставляється між заголовком комірки і IP-заголовком. Приміщення значення мітки в заголовок ATM-осередки спрощує роботу ATM-комутатора, до якого як і раніше досить переглянути тільки заголовок осередку.

Подібним же чином значення самої верхньої мітки може бути вміщено в поле DLCI або в заголовку кадру FR (рис. 1.5, г). Але необхідно звернути увагу на те, що в обох випадках поле TTL залишається невидимим для комутатора і зменшується на одиницю в міру проходження через транзитні вузли до точки призначення.

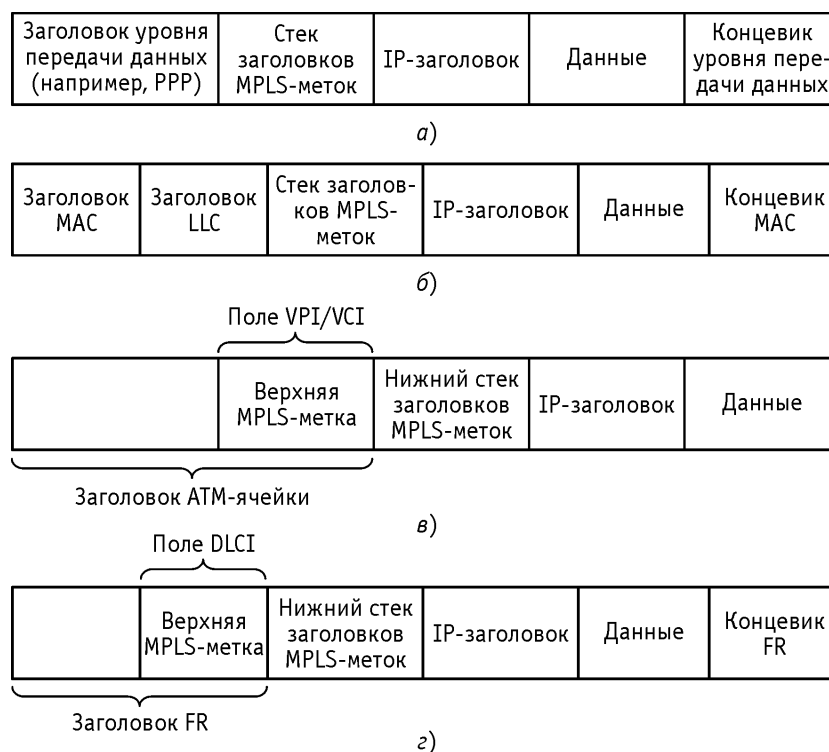


Рисунок 1.5 – а) Стек заголовків міток MPLS в протоколі PPP; б) стек заголовків міток в кадрі мережі стандарту 802.x; в) стек заголовків міток MPLS в поле VPI / VCI; г) стек заголовків міток MPLS в поле DLCI

1.7 Таблиці пересилання

Для зв'язку отриманих міток з вихідними мітками використовуються різні таблиці і карти. Вони призначені для подальшого управління стеками міток.

Коли пакет MPLS потрапляє в маршрутизатор з комутацією по мітках (LSR), тоді даний маршрутизатор переглядає наявну у нього таблицю з так званої інформаційної базою міток LIB (Label Information Base) для прийняття рішення про подальшій обробці пакетів (таблиця 1.1).

Таблиця 1.1 – Інформаційна база міток

Вхідна мітка	вхідний інтерфейс	Адреса наступного LSR	Вихідна мітка	Вихідний інтерфейс
--------------	-------------------	-----------------------	---------------	--------------------

Інформаційну базу іноді називають також таблицею NHLFE (Next Hop Label Forwarding Entry). Таблиця NHLFE використовується при пересиланні пакету з міткою (таблиця 1.2). Записи в таблиці NHLFE містять наступну інформацію:

- обмін зовнішньої мітки з стека;
- витяг зовнішньої мітки з стека;
- обмін мітки (тобто кожен LSR після отримання пакету змінює значення мітки перш, ніж послати пакет наступного LSR);
- вставка нової мітки в стек;
- наступний маршрутизатор в LSP, причому «наступним» може бути той же самий LSR.

Таблиця 1.2 – Приклад таблиці NHLFE

NHLFE	Наступний LSR	Операція з міткою	Вхідна мітка	вихідний порт
126	LSR C	Вставка	888	3821
546	LSR D	обмін	777	7653
338	LSR F	витяг	–	–

Таблиця так само може містити інформацію про послідовність збірки пакета на каналному рівні і кодуванні MPLS-заголовка стека.

Можливо, що прийняв пакет маршрутизатор є останнім на маршруті LSP. В даному випадку мітка з пакета витягується і пакет посилається далі на підставі рішення третього рівня еталонної моделі OSI.

Таблиця зв'язку (FEC-to-NHLFE, FTN) використовується, якщо був прийнятий пакет, який не мав до цього мітки, але до якого була додана мітка перед відправкою. У таблиці 1.3 записано відповідність між кожним FEC і набором NHLFE. Тобто карта FEC-to-NHLFE співвідносить кожен клас FEC до безлічі NHLFE, що містять більше ніж одну позначку, але перш ніж пакет буде посланий, повинна бути обрана рівно одна мітка безлічі.

Таблиця 1.3 – FEC-to-NHLFE

FEC	NHLFE
A	126
C	546
K	338

Існує також карта вхідний мітки (Incoming Label Map, ILM) – таблиця 1.4. У ній знаходиться зв'язка до таблиці NHLFE для пакетів, вже містять мітку, тобто протокол розподілу міток (Label Distribution Protocol, LDP) записує кожну вхідну мітку в NHLFE. Якщо ILM вносить конкретну мітку в безліч NHLFE, то перш ніж пакет буде відправлений, з стека повинна бути обрана рівно одна мітка з безлічі. Мітка на початку стека використовується як індекс в ILM. Якщо ILM вносить мітку в безліч, що містить більше ніж один NHLFE, то це може бути корисно, якщо, наприклад, вона застосовується для розподілу навантаження між різними каналами.

Таблиця 1.4 – Приклад таблиці Incoming Label Map (ILM)

Label	NHLFE
888	126
777	546
555	757

1.8 Прив'язка мітка – FEC

Кожен запис в таблиці пересилання, яку веде LSR, містить одну або більше вихідних міток. Відповідно до цього правила забезпечуються два типу прив'язки міток до FEC: локальний і віддалений.

Перший тип: мітка для прив'язки вибирається і призначається в LSR локально, тобто локальний маршрутизатор LSR встановлює відповідність мітки з FEC.

Другий тип: LSR отримує від сусіднього LSR інформацію про прив'язку мітки локального вузла.

Засоби управління комутацією по мітках використовують для заповнення таблиць пересилання як локальну, так і віддалену прив'язку міток до FEC. Це може відбуватися в двох варіантах: upstream і downstream. Перший – це коли мітки з локальної прив'язки використовуються як вхідні мітки, а мітки з віддаленої прив'язки використовуються як вихідні мітки. Другий варіант - прямо 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) або спадне зв'язування, тому що в цьому випадку прив'язка яку переносять 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. Перевагою даного методу є те, що спадний LSR може вибрати мітку так, що вона буде одночасно бути індексом в його NHLFE таблиці. Таким чином, вся необхідна для пересилання інформація може бути отримана за один крок.

Робочою групою MPLS передбачені наступні можливості:

- спадний LSR відповідає за створення міток і розподіляє їх з урахуванням його входів і класу обслуговування. Кожному FEC приєднується власна мітка L. Потім спадний LSR повідомляє висхідному LSR цей розподіл. Таким чином, той дізнається, яку мітку треба приєднати до даного пакету;

- спадний LSR може використовувати одну і ту ж мітку багаторазово. Для вибору правильного маркованого маршруту LSP, при багаторазовому використанні мітки, LSR повинен стежити за тим, щоб призначення було однозначним. Цей метод прив'язки мітки до FEC застосовується найбільш часто в MPLS.

Другий варіант називається прив'язкою мітки до FEC «зверху» (upstream label binding) або висхідний зв'язування, тому що в цьому випадку прив'язка пакетом мітки яку переносять 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.

Як зображено на рис. 1.7 висхідний розподіл мітки відноситься до методу, в якому розподіл мітки зроблено висхідним LSR A. Термін висхідний відноситься до напрямку, в якому посилається пакет користувача. Верхній LSR A посилає пакет нижньому LSR B. Пакет був визначений заздалегідь як пакет класу FEC і мітка (скажімо, мітка L) пов'язана з FEC. Таким чином, L – вихідна мітка для A, а для B L – вхідна мітка.

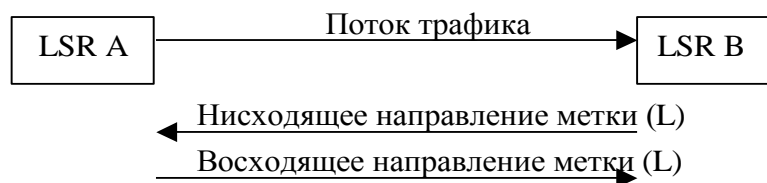


Рисунок 1.7 – Висхідне і спадне зв'язування

Важливим моментом при розподілі міток між LSR A і LSR B є, який з маршрутизаторів буде ініціатором прив'язки міток 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, то він бере мітку з пулу; коли маршрутизатор знищує раніше створену прив'язку, він повертає мітку, пов'язану з цією прив'язкою, назад в пул.

LSR може вести або одну загальну таблицю пересилання, або кілька таблиць – по одній на кожен інтерфейс. Коли маршрутизатор веде загальну таблицю пересилання, тоді він має один пул міток. Коли LSR веде кілька таблиць, тоді він має окремий пул міток для кожного інтерфейсу.

LSR створює або знищує прив'язку мітки до FEC при певному події. Така подія може ініціюватися або пакетами даних, які повинні пересилатися маршрутизатором LSR, або керуючої (маршрутної) інформацією (наприклад, маршрутною інформацією протоколу OSPF, повідомленнями, повідомленнями Path / Resv протоколу RSVP), яка повинна оброблятися LSR. Коли створення або знищення прив'язки ініціюється пакетами даних, це називається прив'язкою під впливом даних (data-driven). Коли створення або знищення прив'язки ініціюється керуючої інформацією, то це називається прив'язкою під впливом керуючої інформації (control-driven). Прив'язка під впливом даних передбачає, що LSR підтримує як функції пересилання при комутації по мітках, так і функції пересилання при традиційній маршрутизації. Підтримка функцій пересилання при традиційній маршрутизації необхідна тому, що прив'язка мітки є ефект, супутній традиційної маршрутизації пакета.

Кожен з цих двох механізмів має безліч варіантів. Наприклад, механізм прив'язки під впливом даних може створювати прив'язку для потоку додатки, як тільки він виявить перший пакет потоку, або чекати до тих пір, поки не виявить кілька пакетів, припускаючи при цьому, що потік існує досить довго для того, щоб встигнути створити прив'язку .

Вибір того чи іншого методу створення прив'язки буде, безумовно, певним чином впливати на продуктивність і масштабованість механізму, тобто на те, наскільки добре він буде працювати в міру зростання мережі. Можна також

очікувати деякий вплив методу прив'язки на живучість механізму, тобто наскільки добре він буде працювати в різних умовах експлуатації.

Важливою проблемою якості функціонування, що виникає при використанні схем прив'язки під впливом даних (і, в меншій мірі, – схем прив'язки під впливом керуючої інформації) є продуктивність. Кожен раз, коли LSR вирішує, що потік повинен комутуватися по мітках, йому необхідно обмінюватися інформацією про прив'язку міток з суміжними LSR і йому може також знадобитися внести деякі зміни в прив'язці міток до FEC. Всі ці процедури вимагають передачі трафіку, керуючого роздачею інформації про прив'язку, і, отже, споживають ресурси засобів управління комутацією по мітках. Більш того, ці процедури споживають тим більше ресурсів засобів управління, чим більше частка потоків, обраних для комутації по мітках. Якщо LSR не може призначати і розподіляти мітки зі швидкістю, необхідної алгоритмом виявлення потоків, то комутуватися по мітках буде менший відсоток потоків, а від цього буде зменшуватися загальна продуктивність.

У меншій мірі така ситуація відноситься до схем, які працюють під впливом керуючої інформації. Поки топологія залишається стабільною, весь трафік, що надходить в транзитний маршрутизатор LSR, може комутуватися по мітках без обробки пакетів керуючим процесором. Схеми, що працюють під впливом керуючої інформації, могли отримати інформацію про прив'язку маршрутів від сусідніх LSR, які не були наступними ділянками таких маршрутів. Коли топологія зміниться так, що ці «сусіди» стануть наступними ділянками маршрутів, комутація по мітках триватиме без переривання. Але на продуктивність схем, що працюють під впливом даних, впливає зміна топології. Якщо змінюється маршрут проходження потоку, нові LSR на цьому маршруті сприймають це так, як якби був створений новий потік [13]. Будь-який такий новий потік повинен спочатку пересилатися традиційно. Таким чином, зміна топології збільшує навантаження на LSR, який щойно став новим наступним маршрутизатором, для деякого іншого LSR. По-перше, він раптово отримує велике число потоків, які раніше йшли по іншому маршруту. Крім того, не

припиняється надходження нових потоків від знову запущених додатків. Всі ці потоки повинні пересилатися і аналізуватися алгоритмами виявлення потоків. Дана ситуація в свою чергу створює додаткове навантаження як на кошти пересилки при традиційній маршрутизації, так і на кошти управління комутацією по мітках. Під час таких перехідних процесів продуктивність засобів пересилання LSR може наблизитися до продуктивності його коштів пересилки при традиційній маршрутизації, тобто стати приблизно на порядок нижче, ніж максимальна продуктивність LSR.

Проблема з продуктивністю для схем, що працюють під впливом керуючої інформації, виникає у випадках об'єднання маршрутів. Це є конфліктом між масштабністю і продуктивністю.

1.9 Вибір маршруту

MPLS підтримує два методи вибору маршруту для FEC:

При послідовній маршрутизації (Hop-by-Hop) кожен маршрутизатор вибирає шлях для пакета незалежно від інших маршрутизаторів (рис. 1.8). Даний метод в даний час застосовується в багатьох мережах, наприклад, при використанні протоколу OSPF. Перевагою такого методу є гнучкість при виборі маршруту. Однак якщо один із шляхів постійно перевантажений, то на це не можна вплинути, тому що автоматично вибирається найкоротший шлях.

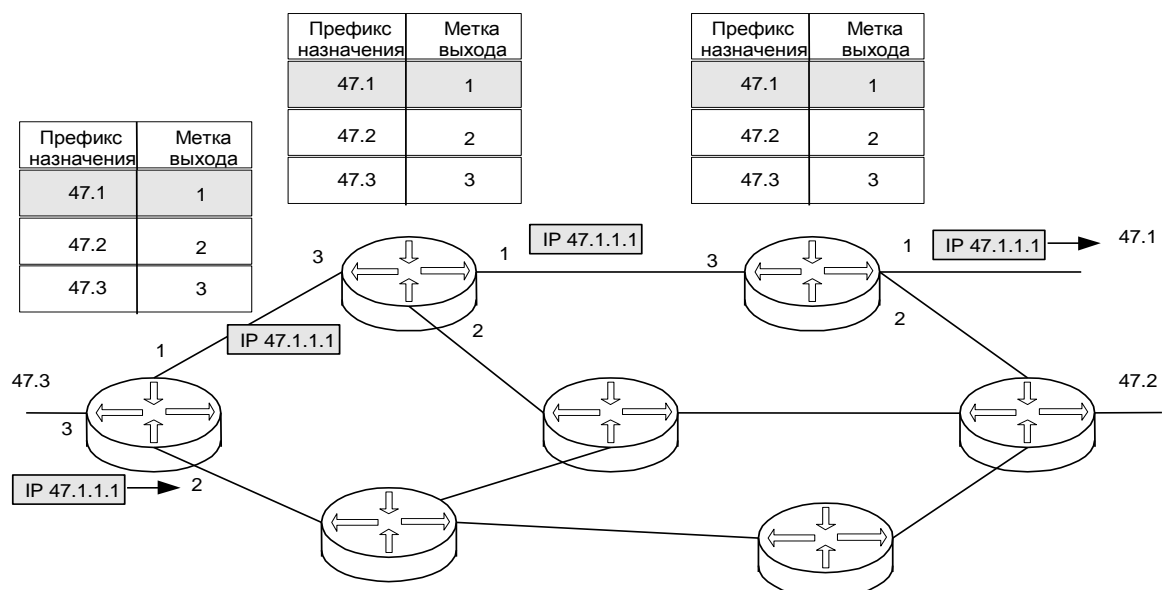


Рисунок 1.8 – Послідовна маршрутизація

При явній маршрутизації (Explicit Routing) не кожен маршрутизатор може приймати рішення (рис. 1.9). Вхідний LER вибирає маршрутизатори, які складуть маркований маршрут. Запит міток (Label Request) відбувається направлено і управляється джерелом. Такий метод використовується для передачі даних з високими вимогами до якості. Провайдер мережі може виробляти маршрутизацію в залежності від різних параметрів (якість, вартість, завантаженість мережі, і так далі). Можуть бути обрані шляхи, які не будуть обрані звичайними протоколами маршрутизації. Таким чином, такі шляхи не завжди є найкоротшими. Маршрутизація відбувається аналогічно технології ATM.

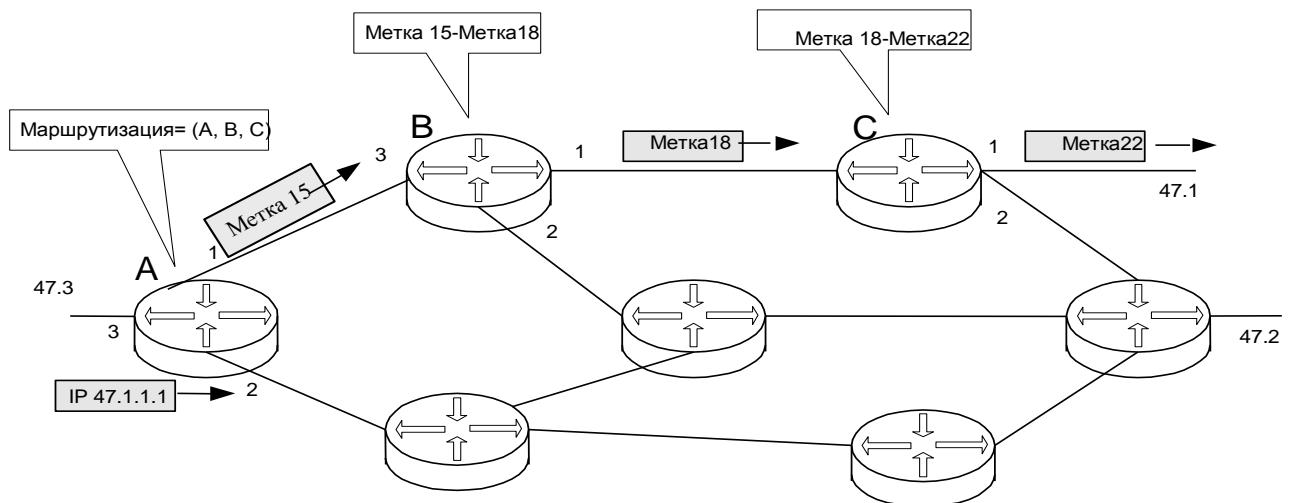


Рисунок 1.9 – Явна маршрутизація

1.10 Моделювання комп'ютерних мереж

Комп'ютерні мережі надають широкий вибір послуг і механізмів для забезпечення роботи необхідних додатків за допомогою реалізації служби каталогів, різноманітних мережевих сервісів та постійно ускладнюються. У зв'язку з цим, процес проектування комп'ютерних мереж також суттєво ускладнюється.

В теперішній час найбільш розповсюдженим підходом до проектування комп'ютерних мереж є експертні оцінки. Спеціалісти в галузі інформаційних систем проектують комп'ютерну мережу у відповідності до поставлених їй задач. Але рішення цих спеціалістів має суб'єктивний характер і не завжди є оптимальним.

Тому у поєднанні з даним методом проектування доцільно застосовувати засоби моделювання комп'ютерних мереж.

Класифікувати програмні продукти для моделювання можна за такими критеріями як вартість(платне програмне забезпечення, вільне програмне забезпечення), можливості програм (високофункціональні, середньофункціональні, низькофункціональні), типи мереж, що моделюються(глобальні, локальні, локальні і глобальні), принцип роботи(програми, що моделюють, програми, що оцінюють) [9,11].

Розглянемо розповсюджені програмні продукти для моделювання комп'ютерних мереж.

Такі програмні продукти, як COMNET III, NetMaker XA, MIND, AutoNet/Designer, AutoNet/MeshNET, BONES та Opnet є високофункціональними засобами моделювання комп'ютерних мереж, що мають можливість моделювання як локальних, так і глобальних мереж. Але вони мають дуже велику вартість.

Ряд програм, таких як Prophecy, StressMagic, Performance-1, Performance-3, Netcracker мають меншу вартість, тобто відносяться до дешевих засобів моделювання, але вони мають середню функціональність, хоча також можуть моделювати і локальні, і глобальні мережі.

Крім комерційних програм існують засоби моделювання які оцінюють комп'ютерні мережі, це: D-ITG, Friendly Pinger, Netperf, RUDE, MGEN, Iperf та

UDP Packet Generator. Вони мають низьку функціональність та не підтримують глобальні мережі. Детально програмні продукти представлені в таблиці 1.5.

Таблиця 1.5 – Програмне забезпечення для моделювання комп'ютерних мереж

Компанія та продукт	Вартість (доларів)	Тип мережі	Необхідні ресурси	Примітка
American NYTech, Prophecy	1495	ЛС	8Мб ОП, 6 Мбдиск, DOS, Windows, OS/2	Оцінювання продуктивності при роботі з текстовими і графічними даними по окремих сегментах і мережі в цілому.
Make System, NetMaker XA	6995-14995	ЛС, ГС	128 МбОП, 2000 Мбдиск, AIX, Sun OS, Sun Solaris	Перевірка даних про топологію мережі; імпорт інформації про трафік, що отримується в реальному часі.
Network Analysis Center, MIND	9400-70000	ГС	8 МбОП, 65 Мбдиск, DOS, Windows	Засіб проектування, оптимізації мережі, містить дані про вартість типових конфігурацій з можливістю точного оцінювання продуктивності
UDP Packet Generator	Вільне програмне забезпечення	ЛС	4 МбОП, 2 Мбдиск, Windows, Linux	Пакет інструментів, який дозволяє створювати і відправляти один або декілька UDP пакетів
GNS3	Вільне програмне забезпечення	ЛС, ГС	Windows XP/Vista/7, процесор з частотой 1.5 ГГц, 1 Гбайт ОП	Програмний симулятор маршрутизаторів Cisco, працює на більшості Linux-систем, Mac OS X і Windows, при цьому дозволяє емулювати апаратну частину маршрутизаторів, безпосередньо завантажуючи і взаємодіючи з реальними образами Cisco IOS

Але серед вільного програмного забезпечення є такі засоби моделювання комп'ютерних мереж, які за функціональністю не поступаються своїм високофункціональним комерційним аналогам, таким як COMNET III, Opnet та інші. Таким засобом моделювання є GNS3.

GNS3 – це графічний симулятор мережі, який дозволяє змоделювати віртуальну мережу з маршрутизаторів і віртуальних машин. Незамінний інструмент для навчання та тестів.

Графічний мережевий симулятор GNS3 призначений для віртуалізації роботи операційних систем різного призначення, в основному для операційних систем маршрутизаторів/комутаторів Cisco IOS (internet network operation system – між мережева операційна система) і JunOS (juniper operation system - операційна система фірми juniper) з метою моделювання роботи мережевих протоколів і технологій. Так само графічний симулятор дозволять підключати і інші операційні системи, використовуючи віртуальні машини Qemu і VirtualBox.

За допомогою GNS3 можливо створювати складні по топології мережі і об'єднувати реальну мережу з віртуальним обладнанням, яке повністю як програмно, так і апаратно віртуалізується. GNS3 буде працювати в Windows, Linux, MacOS X. Вона є вільним програмним забезпеченням, тому завантажити GNS3 можна абсолютно безкоштовно. GNS3 легко інтегрується з аналізаторами типу Wireshark. Також можна з'єднати емульованого мережу і мережеві інтерфейси на реальній машині.

Підтримувані типи образів операційних систем емульованого обладнання - IOS / IPS / PIX / ASA / JunOS.

Метою роботи є аналіз активного обладнання інформаційних телекомунікаційних мереж на основі технології MPLS. Для реалізації поставленої мети необхідно розв'язати наступні задачі:

- 1) Проаналізувати технології моделювання активного обладнання.
- 2) Побудувати аналітичну модель активного обладнання на основі технології MPLS, для віддалених мереж.
- 3) Виконати порівняльну оцінку IP-маршрутизації та технології маршрутизації по міткам MPLS.

1.11 Висновки до першого розділу

Була розглянута технологія багатопроTOCOLЬНОЇ комутації по міткам, приведені засоби ефективної організації QoS, обрано програмний продукт для моделювання аналітичної моделі. Виходячи с усього вище оглянутого були сформовані задачі дослідження.

РОЗДІЛ 2

МЕТОДИ ОПТИМІЗАЦІЇ АКТИВНОГО ОБЛАДНАННЯ НА ОСНОВІ ТЕХНОЛОГІЇ MPLS

2.1 Методи оптимізації

Вибір методу розв'язання задачі оптимізації – один з найважливіших етапів оптимізації. Можна виділити наступні групи методів [9–14]:

- аналітичні методи;
- методи математичного програмування.

Розглянемо групи цих методів і більш детально деякі з них. Група аналітичних методів оптимізації об'єднує аналітичний пошук екстремуму функції, метод множників Лагранжа, варіаційні методи і принцип максимуму. Метод максимального потоку (мінімального розрізу) визначає безліч ребер, при видаленні яких мережа ділиться на дві незв'язані частини. Пропускна здатність цих ребер обмежує обсяг трафіку між двома частинами мережі. Аналітичний пошук екстремуму функцій, заданих без обмежень на незалежні змінні, застосовується до завдань, у яких оптимізується функція має аналітичний вираз, що диференціюється в усьому діапазоні дослідження, а число змінних невелика. Це один з найбільш простих методів.

Група методів математичного програмування включає динамічне програмування, лінійне програмування та нелінійне програмування.

Динамічне програмування – ефективний метод вирішення завдань оптимізації багатостадійних процесів. Метод передбачає розбивку аналізованого процесу на стадії (в часі і просторі). Розгляд задачі починається з останньої стадії процесу, і оптимальний режим визначається по стадіям.

Методи нелінійного програмування – об'єднують різні способи вирішення оптимальних задач: градієнтні, безградієнтні і випадкового пошуку. Загальним для методів нелінійного програмування є те, що їх використовують при вирішенні завдань з нелінійними критеріями оптимальності.

Всі методи нелінійного програмування – це чисельні методи пошукового типу. Суть їх – у визначенні набору незалежних змінних, що дають найбільшу приріст оптимізується функції.

Лінійне програмування – метод для вирішення завдань оптимізації з лінійними виразами для критерію оптимальності і лінійними обмеженнями на область зміни змінних. Подібні завдання вирішуються ітераційним способами. Одним із способів реалізації лінійного програмування є симплекс-алгоритм, який майже завжди, за винятком деяких випадків, може знайти оптимальне рішення. В основі цього алгоритму лежить повний перебір можливих варіантів вирішення завдання.

Евристичний метод – спрямований на скорочення перебору. Рішення, одержувані даним методом, не є найкращими, а відносяться лише до безлічі допустимих.

Імітаційне моделювання – це найпоширеніший різновид аналогового моделювання, реалізованого за допомогою набору математичних інструментальних засобів, спеціальних імітуючих комп'ютерних програм і технологій програмування, що дозволяють за допомогою процесів-аналогів провести цілеспрямоване дослідження структури і функцій реального складного процесу в пам'яті комп'ютера в режимі «імітації», виконати оптимізацію деяких його параметрів.

Кожен метод має свої особливості, які визначаються їх принципом роботи і реалізацією, і відрізняються один від одного як складністю і граничними умовами, так і відхиленням від оптимального значення.

Використовуючи методи лінійного програмування і евристичний, отримуємо дизайн LSP, в той час як метод максимального потоку (мінімального розрізу), визначає максимальне навантаження на ребра мінімального розрізу, а також сам розріз (ребра, з яких він складається).

Точне рішення проблеми оптимізації можна отримати за допомогою лінійного програмування, проте складність обчислень при лінійному програмуванні швидко зростає зі збільшенням числа вузлів в мережі і для великих мереж є критичною, що призводить до необхідності використання евристичних методів.

2.2 Основні принципи імітаційного моделювання

Накопичений при розробці та використанні моделей досвід можна висловити кількома основними принципами.

Принцип інформаційної достатності. При повній відсутності інформації про досліджувану систему побудова її моделі неможливо, а за наявності повної інформації – недоцільно. Існує певний критичний рівень апріорних відомостей про систему (рівень інформаційної достатності), при досягненні якого може бути побудована адекватна модель.

Принцип доцільності. Модель створюється для досягнення деяких цілей, визначених на початковому етапі формулювання проблеми моделювання.

Принцип здійсненності. Модель повинна забезпечувати досягнення поставленої мети дослідження з ймовірністю, істотно відрізняється від нуля, і за кінцевий час. Зазвичай задають деяке порогове значення P_0 ймовірності досягнення мети моделювання $P(t)$, а також граничне час досягнення цієї мети t_0 . Модель вважається здійсненою, якщо $P(t) \geq P_0$ и $t \leq t_0$.

Принцип множинності моделей. Модель повинна в першу чергу відображати ті властивості реальної системи (явища), які впливають на обраний показник ефективності. При використанні будь-якої конкретної моделі пізнаються лише певні сторони реальності. Для більш повного її дослідження необхідний ряд моделей, що дозволяють відображати досліджуваний процес з різних сторін і з різним ступенем деталізації.

Принцип агрегування. Складну систему в більшості випадків можна що з агрегатів (підсистем), для адекватного формального опису яких виявляються придатними деякі стандартні математичні схеми. Цей принцип дозволяє досить гнучко перебудувати модель залежно від завдань дослідження [12].

Принцип параметризації. У ряді випадків модульована система має в своєму складі деякі відносно ізольовані підсистеми, що характеризуються певним параметром, в тому числі векторних. Такі підсистеми можна замінювати в моделі відповідними числовими величинами, а не описувати процес їх функціонування. При необхідності залежність значень цих величин від ситуації може здаватися в

вигляді таблиць, графіків або аналітичних виразів. Принцип параметризації дозволяє скоротити обсяг і тривалість моделювання, але при цьому параметризація знижує адекватність моделі.

2.3 Структура імітаційних моделей

У найзагальнішому вигляді структура імітаційної моделі виглядає наступним чином:

$$E = f(x_i, y_i), \quad (2.1)$$

де E – результат дії системи, x_i – змінні і параметри, якими можна управляти, y_j – змінні і параметри, управління якими недоступно, f – функціональна залежність між x_i і y_j визначає величину E .

Моделі являють собою деяку комбінацію таких складових: компоненти, змінні, параметри, функціональні залежності, обмеження, цільові функції. Компоненти – складові частини, які при відповідному об'єднанні утворюють систему (елементи системи або її підсистеми). Параметри – величини, які можна вибирати довільно на відміну від змінних. Наприклад, в рівнянні $y = 7x$, число 7 – параметр, а x і y – змінні. Змінні поділяються на екзогенні та ендогенні. Екзогенні (незалежні) змінні, називаються також вхідними – породжуються поза системою або є результатом впливу зовнішніх причин. Ендогенні (залежні) виникають в системі або в результаті впливу внутрішніх причин, часто є вихідними.

Функціональні залежності описують поведінку змінних і параметрів в межах компонента або висловлюють співвідношення між компонентами системи. Ці співвідношення, або операційні характеристики, є або детермінованими, або стохастичними. Детерміновані співвідношення встановлюють залежність між певними змінними або параметрами в випадках, коли процес на виході системи однозначно визначається заданою інформацією на вході. Стохастичні залежності при заданій вхідній інформації дають на виході невизначений результат. Можна вказати лише безліч можливих результатів і, в деяких випадках, ймовірності їх отримання.

Обмеження є встановлювані межі зміни значень змінних або обмежують умови розподілу і витрачання тих чи інших засобів (енергії, запасів, часу і т.п.). Вони можуть вводитися або розробником (штучні обмеження), або самою системою внаслідок притаманних їй властивостей (природні обмеження).

Цільова функція (функція критерію) – точне відображення цілей або завдань системи і необхідних правил оцінки їх виконання [13–19]. Побудована модель повинна бути гарною, якісною і відповідно задовольняти таким умовам, тобто бути:

- простою і зрозумілою користувачеві;
- зручною в управлінні і обігу;
- цілеспрямованою;
- надійною в сенсі гарантії від абсурдних відповідей;
- змістовної з точки зору можливості відображати основні властивості реальної системи або процесу;
- адаптивної, що дозволяє легко переходити до інших модифікацій або оновлювати дані;
- допускає поступові зміни в напрямку ускладнення своєї структури.

2.4 Алгоритм імітаційного моделювання

Імітаційне моделювання дозволяє досліджувати процеси в часі. Природно, що дійсна швидкість протікання більшості досліджуваних процесів значно відрізняється від швидкості моделювання на ЕОМ. Тому при використанні імітаційного моделювання необхідно співвідносити між собою три вистави часу: реальне, модельне (системне) і машинне.

Під реальним увазі час, в якому досліджувана система працює в дійсності. Тут можуть бути мс, хвилини, години, зміни, місяці, роки і т.д. У масштабі модельного часу організовується робота самої імітаційної моделі. Це умовні одиниці часу, виражені як цілими, так і дійсними числами.

Машинне час відображає реальні витрати часу ЕОМ на проведення моделювання. У зв'язку із зростанням обчислювальної потужності ЕОМ, значимо тільки при використанні досить складних моделей.

Поняття модельного часу служить для вирішення наступних завдань:

- відображення переходу модельованої системи з одного стану в інший;
- синхронізації роботи компонент моделі;
- зміни масштабу часу функціонування досліджуваної системи;
- управління ходом модельного часу;
- моделювання квазіпаралельної обробки подій в моделі.

Під квазіпаралельною розуміють послідовну обробку подій, які в досліджуваній системі відбуваються одночасно. Модельне час може бути реалізовано двома методами – з постійним кроком і з особливих обставин.

Якщо використовується принцип постійного кроку, відлік модельного часу ведеться через фіксовані інтервали Δt , на які розбитий весь модельований період часу. Події в моделі вважаються настали в момент закінчення такого інтервалу. Стан системи визначається для моменту часу t , потім $t+\Delta t$, $t+2\Delta t$ і т.д. Похибки тимчасових характеристик досліджуваної системи залежать від величини Δt .

Метод постійного кроку доцільно застосовувати, якщо:

- події з'являються регулярно, їх розподіл у часі досить рівномірно;
- число подій велике і моменти появи їх близькі;
- неможливо заздалегідь визначити моменти появи подій.

Цей метод досить легко реалізувати навіть з використанням універсальних мов програмування. Але потрібно врахувати, що доведеться визначати стан системи в кожен з моментів часу, навіть якщо в ньому не відбувається ніяких подій. А такі «порожні» відрізки часу можуть бути надто великими. У разі складних моделей навіть сучасна ЕОМ може «зав'язнути» надовго.

Алгоритм моделювання за принципом постійного кроку наведено на рис. 2.1. Тут T_M – поточне значення модельного часу, $z(t_m)$ – стан системи в цей момент часу, ΔT_M – інтервал моделювання.

При моделюванні з особливих станів модельне час змінюється від події до події. Події обробляються в порядку їх настання, а одночасно настали вважаються тільки ті, які є одночасними в дійсності. Алгоритм методу наведено на рис. 2.2.

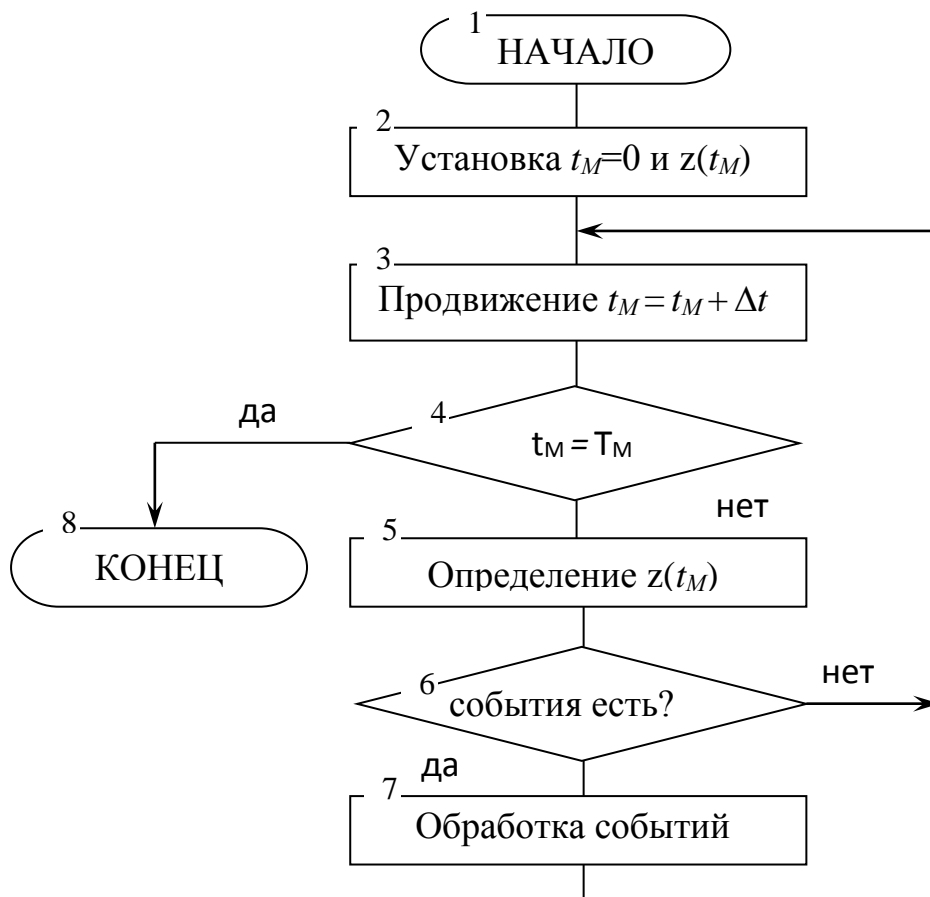


Рисунок 2.1 – Алгоритм моделювання з постійним шагом

Моделювання з особливих станів доцільно, якщо:

- події з'являються нерегулярно або інтервали між ними великі;
- пред'являються підвищені вимоги до точності взаємного положення подій в часі;
- необхідно реалізувати квазіпаралельної обробку одночасних подій.

Недоліком методу можна вважати відносну складність в реалізації, тому що для нього потрібна розробка спеціальної процедури планування подій (календаря подій). Головна перевага – «пропуск» ділянок на тимчасовій осі, де не відбувається ніяких подій.

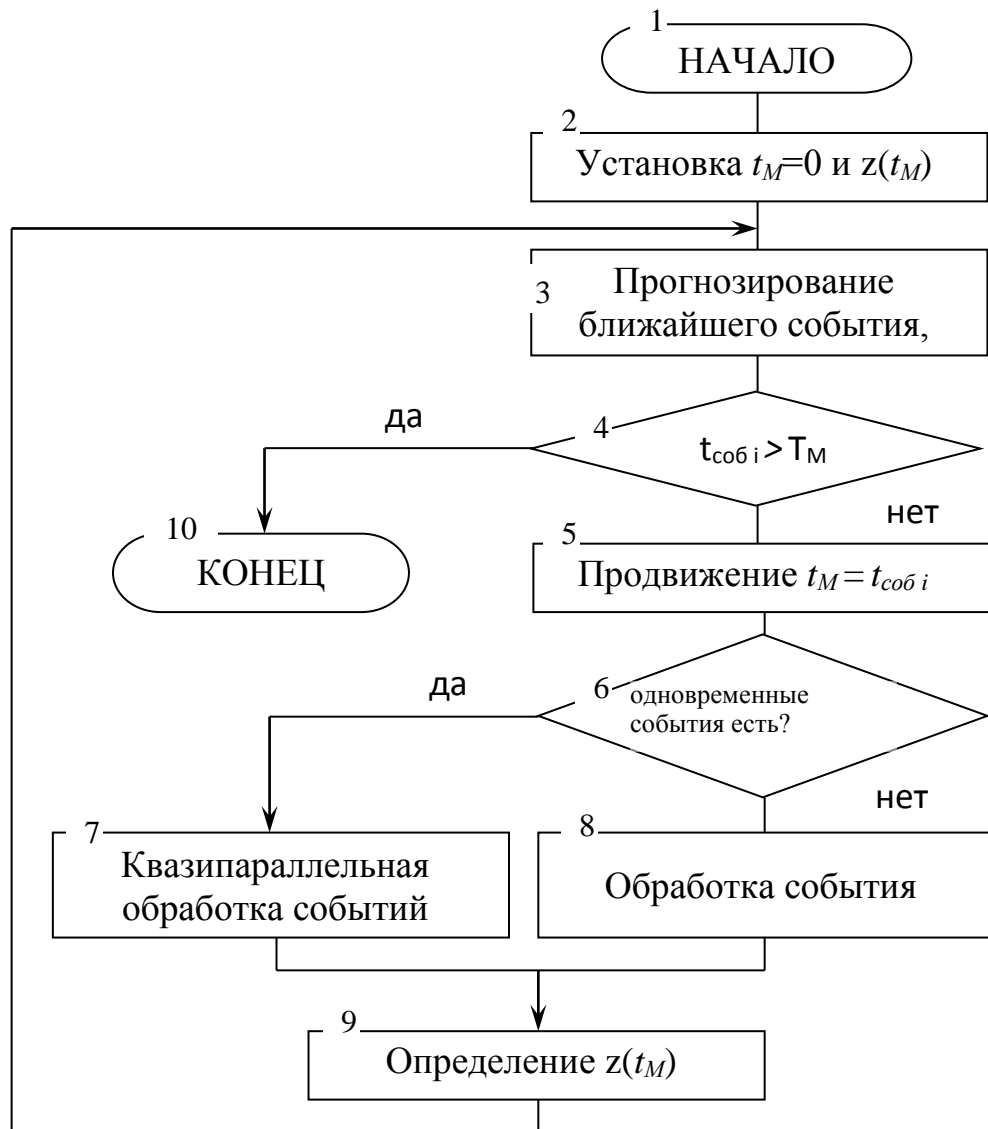


Рисунок 2.2 – Алгоритм моделювання по особливим обставинам

На відміну від інших видів і способів математичного моделювання із застосуванням ЕОМ імітаційне моделювання має свою специфіку: запуск в комп'ютері взаємодіючих обчислювальних процесів, які є за своїми часовими параметрами – з точністю до масштабів часу і простору – аналогами досліджуваних процесів.

2.5 Етапи імітаційного моделювання активного обладнання

Імітаційне моделювання як особлива інформаційна технологія складається з наступних основних етапів.

Структурний аналіз процесів. Проводиться формалізація структури складного реального процесу шляхом розкладання його на підпроцеси, що виконують певні функції і мають взаємні функціональні зв'язку згідно з легендою, розробленою робочою експертною групою. Виявлені підпроцеси, в свою чергу, можуть поділятися на інші функціональні підпроцеси.

Структура загального модульованого процесу може бути представлена у вигляді графа, що має ієрархічну багат шарову структуру, в результаті з'являється формалізоване зображення імітаційної моделі в графічному вигляді. Структурний аналіз особливо ефективний при моделюванні економічних процесів, де (на відміну від технічних) багато складових підпроцеси не мають фізичної основи і протікають віртуально, оскільки оперують з інформацією, грошима і логікою (законами) їх обробки.

Формалізований опис моделі. Графічне зображення імітаційної моделі, функції, виконувані кожним підпроцесом, умови взаємодії всіх підпроцесів і особливості поведінки модельованого процесу (тимчасова, просторова і фінансова динаміка) повинні бути описані на спеціальній мові для подальшої трансляції [15].

Побудова моделі. Зазвичай це трансляція і редагування зв'язків (складання моделі), верифікація (калібрування) параметрів.

Трансляція здійснюється в різних режимах: в режимі інтерпретації, або в режимі компіляції. Кожен режим має свої особливості. Режим інтерпретації простіше в реалізації. Спеціальна універсальна програма-інтерпретатор на підставі формалізованого опису моделі запускає все імітують підпрограми.

Даний режим не призводить до отримання окремої моделює програми, яку можна було б передати або продати замовнику (продавати довелося б і модель, і систему моделювання, що не завжди можливо).

Режим компіляції складніше реалізується при створенні моделюючої системи. Однак це не ускладнює процес розробки моделі. В результаті можна отримати

окрему моделює програму, яка працює незалежно від системи моделювання у вигляді окремого програмного продукту.

Верифікація (калібрування) параметрів моделі виконується відповідно до легендою, на підставі якої побудована модель, за допомогою спеціально обраних тестових прикладів.

Проведення екстремального експерименту для оптимізації певних параметрів реального процесу [1–7].

Можливий інший підхід до визначення основних етапів моделювання:

- 1) розробка імітаційної моделі;
- 2) розробка методики моделювання (планування імітаційного експерименту);
- 3) програмна реалізація моделі (вибір засобів – універсальних мов програмування або спеціалізованих мов моделювання);
- 4) виконання імітаційного моделювання, аналіз і узагальнення результатів, прийняття рішень.

Розглянемо докладніше кожен з цих етапів. Розробка імітаційної моделі є найбільш відповідальним етапом, від ретельності опрацювання якого залежить весь подальший успіх імітаційного моделювання. Послідовність робіт на цьому етапі наступна:

- визначення завдання і її аналіз.
- визначення вимог до інформації.
- збір необхідної інформації.
- висування гіпотез і прийняття припущень.
- визначення основного змісту моделі.
- визначення параметрів, змінних і критеріїв ефективності моделі.
- опис концептуальної моделі і перевірка її достовірності.
- побудова логічної структурної схеми (блок-схеми).

Визначення завдання і її аналіз є першими кроками при розробці імітаційної моделі. Для того щоб знайти прийнятне або оптимальне рішення задачі, необхідно знати, в чому вона полягає.

В першу чергу необхідно переконатися в самому існуванні завдання. Початковою формулюванні властива невизначеність, тому необхідно добре вивчити проблему, уточнюючи постановку задачі.

Формулювання завдання повинна давати чітке уявлення про її масштабі і діапазоні практичного застосування результатів. Повна формулювання завдання повинна містити визначальну формулювання і методологію її рішення.

У визначальну формулювання входять:

- твердження щодо існування і обґрунтування завдання;
- перелік проблемних питань, пов'язаних з вирішенням задачі;
- аналіз масштабності завдання і можливих меж її застосування;
- розбивка вихідної задачі на окремі підзадачі.

Методологія (порядок) виконання завдання включає:

- установку пріоритетності і черговості виконання завдання;
- визначення можливих методів вирішення підзадач;
- обґрунтування вимог необхідних витрат праці (розробка і налагодження програм, допоміжні роботи) і машинного часу;
- складання календарного (мережевого) графіка виконання робіт.

Визначення вимог до інформації, необхідної для кількісного і якісного опису вхідних даних, вимагає відповіді на наступні питання:

- яка інформація може вважатися необхідною;
- які джерела цієї інформації;
- в якому вигляді необхідно подати;
- якими методами доцільно обробляти інформацію?

У разі неможливості отримати певну інформацію, необхідно знайти шляхи її заміни, або розробити інший варіант вирішення завдання.

Під збором інформації розуміють її отримання та оцінювання. Отримати інформацію можна переглядом публікацій, аналізом виробничих джерел, обробкою документів і звітів, підготовкою апріорних і обробкою експериментальних даних, експертним шляхом. Отримана інформація повинна бути оцінена з точки зору її

відповідності розв'язуваної задачі і зручності використання, а також відфільтрована від непотрібних і випадкових даних.

У разі нестачі інформації висувають гіпотези і приймають допущення. Гіпотези замінюють невідомі закономірності розвитку системи та доповнюють постановку задачі. Доводячи гіпотези, отримують більш точне уявлення про рішення задачі. Допущення, тобто затвердження, які тимчасово (до встановлення істини) вважаються вірними, приймають у разі відсутності або неможливості отримання певних даних. Допущення дозволяють перетворити ускладнені величини в зручні для використання.

Основний зміст моделі розробляється з урахуванням висунутих гіпотез і зроблених припущень. При цьому необхідно врахувати специфічні особливості реальної обстановки, самого завдання і засобів її вирішення. Розглядаючи реальну обстановку, як елемент при створенні моделі, необхідно визначити:

- функції системи і способи їх реалізації;
- детерміновані і недетерміновані функції;
- апроксимацію цих функцій в моделі;
- вплив факторів середовища на роботу системи;
- способи взаємодії людини і системи, людини і середовища, системи і середовища;
- апроксимацію цих взаємодій в моделі.

Початковий етап опису імітаційної моделі – визначення параметрів і факторів системи, безпосередньо пов'язаних з модельованою ситуацією, а також виявлення змінних величин, які в процесі функціонування імітаційної моделі можуть набувати різних значень. Змінні величини розв'язуваної задачі складаються з випадкових величин (наприклад, кількість бракованих деталей в партії випуску), регульованих, які називаються змінними управління або керуючими параметрами (момент часу видачі і обсяг замовлення на поставку ресурсу в системах управління запасами), і нерегульованих (наприклад, кількість деталей, які необхідно виготовити в цеху протягом планового періоду).

Вибір критеріїв ефективності є найбільш відповідальний етап в задачах оптимізації систем. Якщо ці критерії встановлені, то за допомогою імітаційної моделі визначаються оптимальні значення змінних управління.

Концептуальна модель – це абстрактна модель, що відображає структуру модельованої системи, властивості її елементів, а також причинно-наслідкові зв'язки системи в рамках цілей дослідження. Вона є формальним описом модельованої системи, що відображає концепцію (сприйняття, систему поглядів на певний процес).

Концептуальна модель ґрунтується на всіх описаних раніше етапах роботи з побудови інформаційної моделі і може бути реалізована математичними і програмними засобами. Найчастіше концептуальну модель записують у вигляді безлічі вихідних передумов.

Рівень деталізації моделі залежить від таких факторів: мети проекту, критеріїв оцінки ефективності, доступності даних, достовірності результатів, думок експертів з даної проблеми, технічних, фінансових та часових обмежень.

Достовірність концептуальної моделі може бути перевірена в наступному порядку:

- з'ясування задуму моделі і доцільності її створення;
- виявлення зв'язку задуму моделі і доцільності її побудови з детермінованими, імовірнісними і середніми значеннями характеристик моделі;
- дослідження прийнятих апроксимацій реальних процесів;
- розгляд критеріїв ефективності;
- дослідження прийнятих припущень і гіпотез;
- встановлення зв'язку з реальними процесами;
- вивчення системи і обурюють факторів зовнішнього середовища;
- встановлення достовірності інформації та її джерел, використовуваних при побудові моделі;
- розгляд процедури в цілому в зв'язку з формулюванням завдання;
- розгляд постановки завдання.

Етап перевірки правильності створення концептуальної моделі часто називають валідацією, а етап перевірки правильності її реалізації (наприклад, у вигляді комп'ютерної програми) – верифікацією.

Створення логічної структурної схеми є заключним етапом побудови імітаційної моделі. Логічна структурна схема імітаційної моделі є впорядкованою і наочне зображення процесу, в якому визначені не тільки дії, а й порядок їх виконання. Логічна схема імітаційної моделі зазвичай створюється за модульним (блокового) принципом, тобто в вигляді сукупності стандартних блоків-модулів. Модульна побудова схем імітаційних моделей забезпечує їх гнучкість.

2.6 Порівняння алгоритмів пошуку оптимального дизайну

Існує кілька параметрів, за якими можна проводити порівняння алгоритмів оптимізації мереж MPLS. В даному випадку порівняння ми будемо проводити по складності алгоритмів, за отриманим вазі дизайну LSP і за інтегральним параметром, що проводить комплексну оцінку алгоритмів (час-якість).

Покажемо залежність складності лінійного програмування і евристичного алгоритму від розміру мережі (рис. 2.3). Функція складності лінійного програмування має більш високою швидкістю росту, ніж функція складності евристичного алгоритму.

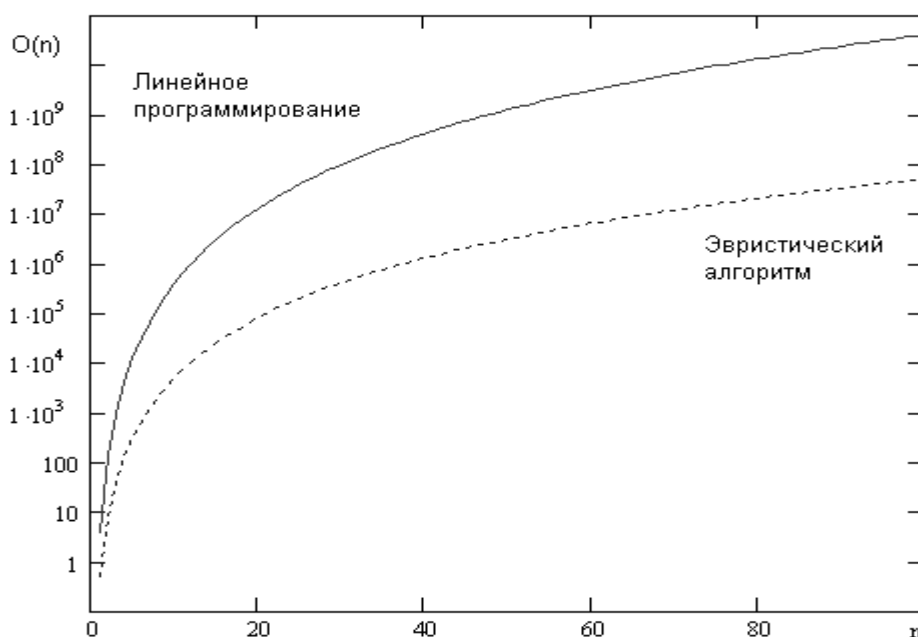


Рисунок 2.3 – Порівняння складності алгоритмів

Як видно з даного графіка, евристичний алгоритм є кращим з точки зору складності, так як для визначення оптимального дизайну LSP він виконує менше операцій в $8n$ раз.

Далі порівняємо ваги отриманого дизайну LSP. Для цього виберемо деякі графи з розмірністю n від 3 до 10. І застосуємо до них метод лінійного програмування і евристичний алгоритм. Далі порівняємо ваги дизайнів LSP, отриманих двома цими методами (рис. 2.4).

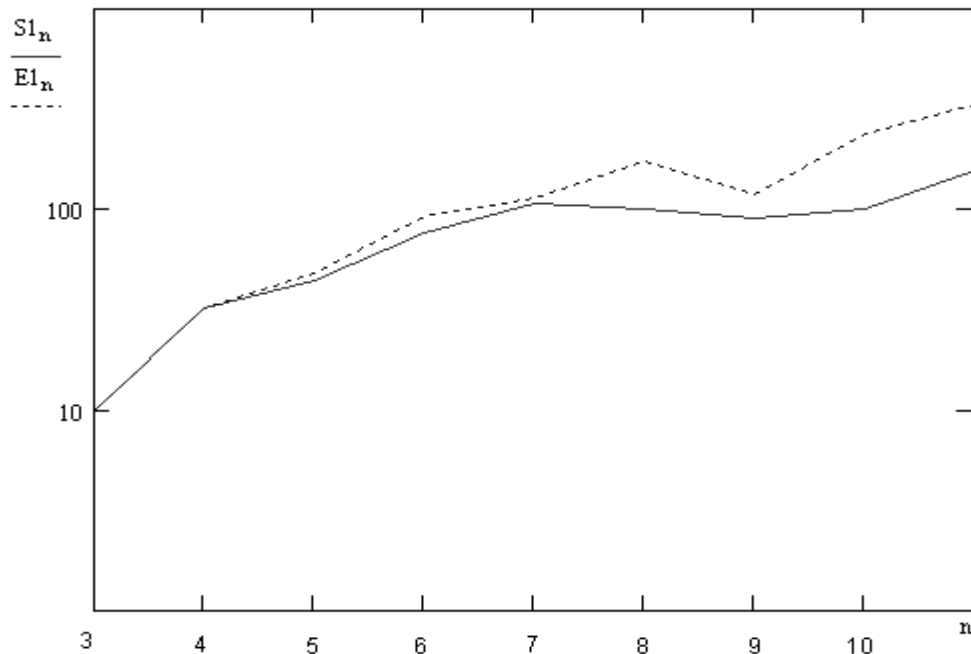


Рисунок 2.4 – Порівняння ваги дизайну LSP, отриманого евристичним алгоритмом (E1) і методом лінійного програмування (S1)

Для проведення комплексної оцінки алгоритмів введемо інтегральний параметр U , вважаємо, що він прямо пропорційний складності алгоритму і отримується цим алгоритмом вазі дизайну LSP. В даному випадку використовуємо значення ваг дизайну LSP, отриманих в попередньому кроці. і відповідно для евристичного алгоритму і методу лінійного програмування (рис. 2.5).

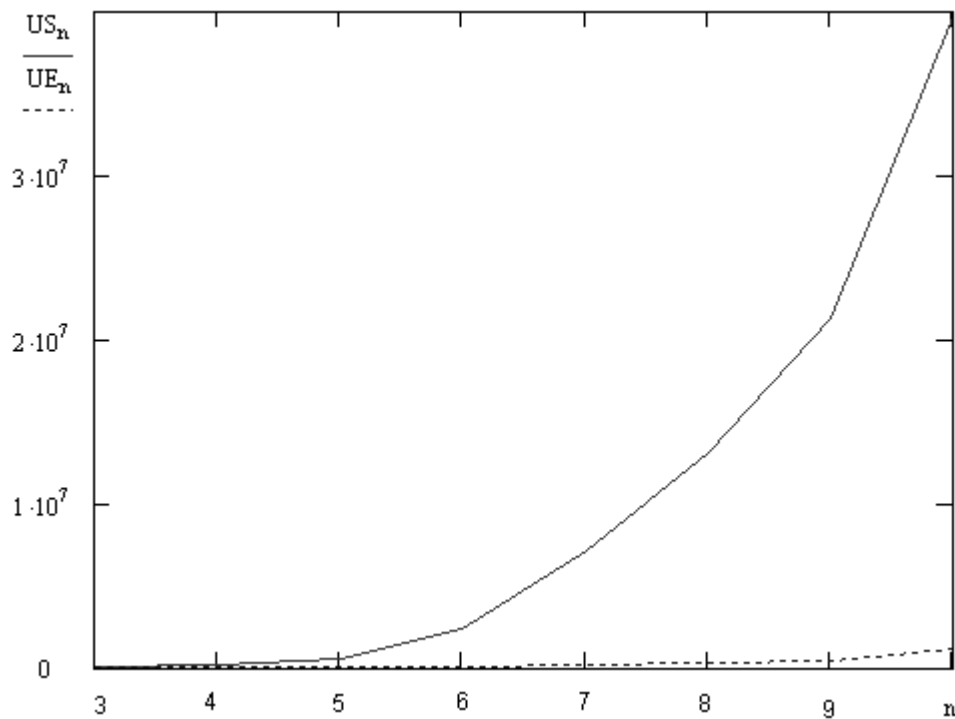


Рисунок 2.5 Комплексна оцінка

Для кожного алгоритму оптимізації необхідно отримати оптимальний дизайн LSP (дизайн з найменшою вагою) при мінімальних витратах операцій на його обчислення. Відповідно, кращим є такий метод, функція комплексної оцінки якого володіє меншим зростанням.

2.7 Висновки до другого розділу

В даному розділі були розглянуті методи оптимізації. Проведено порівняння запропонованого та існуючих методів оптимізації, яке виявило незаперечні переваги імітаційного моделювання. За результатами аналізу цей метод було обрано як інструмент для подальшого моделювання аналітичної моделі активного обладнання.

РОЗДІЛ 3

ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МОДЕЛІ ВІДДАЛЕНИХ МЕРЕЖ НА ОСНОВІ КОМУТАЦІЇ ПО МІТКАМ

3.1 Дослідження активного обладнання

Пристрої, які забезпечують функціонування і працездатність комп'ютерних мереж (глобальних, локальних) називають мережевим обладнанням. При цьому виділяють активне і пасивне мережеве обладнання. Розглянемо докладніше ці види оснащення:

Пасивна техніка, як правило, не отримує живлення від електричної мережі або інших джерел, і виконує функції розподілу або зниження рівня сигналів.

Активне обладнання – це оснащення, що містить електронні схеми, що отримує живлення від електричної мережі або інших джерел і виконує функції посилення, перетворення сигналів і інші. Це означає здатність такого обладнання обробляти сигнал за спеціальними алгоритмами. У мережах відбувається пакетна передача даних, кожен пакет даних містить також технічну інформацію: відомості про його джерелі, цілі, цілісності інформації та інші, що дозволяють доставити пакет за призначенням. Активне мережеве обладнання не тільки вловлює і передає сигнал, але і обробляє цю технічну інформацію, перенаправляючи і розподіляючи потоки які надходять відповідно з вбудованими в пам'ять пристрою алгоритмами. Ця «інтелектуальна» особливість, поряд з живленням від мережі, є ознакою активного обладнання. Під активним обладнанням будемо розуміти такі прилади: мережевий адаптер, ретрансляція, комутатор, маршрутизатор [17].

Мережевий адаптер це додатковий пристрій, що дозволяє комп'ютеру взаємодіяти з іншими пристроями мережі.

Репітер це прилад, як правило, з двома портами, призначений для повторення сигналу з метою збільшення довжини мережевого сегмента; комутатор – прилад з декількома портами, що часто використовується для об'єднання кількох робочих груп локальних мереж.

Основне завдання маршрутизатора – вибір найкращого маршруту в мережі. Крім цієї основної функції в коло відповідальності маршрутизатора входять і інші завдання, такі як буферизація, фільтрація і фрагментація переміщуються пакетів.

Розглянемо режими роботи даних пристроїв. З метою детального розгляду та аналізу активного обладнання мережі створимо і налаштуємо найпростішу модель мережі з використанням технології мультипротокольної маршрутизації по мітках (MPLS).

Технологія MPLS дозволяє наочно моделювати і модернізувати як існуючі так і майбутні IP-мережі. Дана технологія надає новий принцип передачі пакетів, який впливає на перерозподіл потоків даних і на реалізацію віртуальних приватних мереж.

3.2 Побудова мережі

Розглянемо загальну інформацію про технології MPLS. MPLS – (англ. Multiprotocol Label Switching) – мультипротокольна комутація по мітках. Технологія MPLS повинна «підніматися» до рівня вже існуючої і нормально функціонуючої IP мережі, так як дана технологія використовує таблицю маршрутизації (FIB - Forwarding Information Base).

Розглянемо докладніше технологію MPLS-комутації. Комутація MPLS є вдосконаленим методом передачі трафіку по мережі з використанням інформації, що міститься в мітках, які приєднуються до IP-пакетів. У разі використання технологій 2-го рівня, заснованих на передачі фреймів, мітки впроваджуються між заголовками 3-го і 2-го рівнів, а в разі застосування таких технологій як ATM, заснованих на передачі осередків, мітки містяться в полях ідентифікатора віртуального маршруту (Virtual Path Identifier - VPI) і ідентифікатора віртуального каналу (Virtual Channel Identifier - VCI). Метод MPLS об'єднує технології комутації 2-го рівня з технологією маршрутизації 3-го рівня.

Первинною метою використання методу MPLS є створення гнучкої мережевої структури, яка забезпечує підвищену продуктивність і стабільність роботи мережі. Такий технічний підхід передбачає перерозподіл потоків даних і використання

можливостей VPN-мереж, які забезпечують відповідну якість обслуговування і підтримку різноманітних класів обслуговування (Class of Service - CoS). У мережі MPLS мітка яка надходить пакетам призначається граничним вхідним маршрутизатором, який виконує комутацію по мітці (Label-Switched Router - LSR). Далі пакети проходять по маршруту з комутацією по мітках (Label-Switched Path - LSP). Кожен маршрутизатор LSR приймає рішення про відправку, яке базується тільки на змісті мітки. На кожному переході LSR-пристрій видаляє існуючу мітку і вставляє нову, яка задає напрям наступного переходу для відправки пакета. На вихідному граничному LSR-пристрої (egress Edge LSR) мітка видаляється, і пакет наплавляється до пункту призначення. Використання комутації по мітці дозволяє маршрутизаторам і АТМ-комутаторам, на яких встановлена служба MPLS, приймати рішення про відправку пакетів на підставі вмісту міток, тобто не вимагається виконання складного алгоритму пошуку маршрутів, заснованих на IP-адреси пунктів призначення. Застосування цього методу в IP-мережах має ряд наступних переваг [6–14].

- Мережі VPN. При використанні технології MPLS провайдери служб можуть створювати мережі VPN в магістральних мережах 3-го рівня для більшої кількості користувачів, використовуючи при цьому загальну інфраструктуру, без необхідності шифрування або використання додатків кінцевого користувача.

- Перерозподіл потоків. Ця функція надає можливість задати явним образів один або кілька маршрутів, за якими будуть проходити по мережі потоки даних. Вона також дозволяє задати різні параметри обробки трафіку для різних класів. Перерозподіл потоків дозволяє оптимізувати використані смуги пропускання на недостатньо завантажених маршрутах.

- Якість обслуговування. Використовуючи можливості якості обслуговування MPLS, провайдери служб можуть запропонувати користувачам VPN-мереж різні класи обслуговування з надійними гарантіями QoS.

- Інтеграція IP і АТМ. У більшості мереж операторів зв'язку використовується модель накладення, в якій на 2-му рівні застосовується метод АТМ, а на 3-му – механізм IP. Однак в цьому випадку при необхідності розширення мережі

виникають серйозні проблеми. При використанні служби MPLS оператори зв'язку можуть передати багато з функцій площині управління ATM третього рівня, тобто спрощується установка мережі, управління нею і зменшується її складність. Цей метод надає найширші можливості розширення мережі та усуває так званий «податок на осередок» (супутні передачі даних службові повідомлення) властивий середовищі ATM при передачі потоків даних протоколами IP.

Метод MPLS об'єднує високу продуктивність і можливості комутації 2-го (канального) рівня з перевіреними на практиці можливостями розширення мережі при використанні маршрутизації 3-го (мережевого) рівня. Це дозволяє провайдерам служб задовольнити вимоги, що виникають при «вибуховий» зростання кількості користувачів, забезпечуючи диференціацію служб без втрати існуючої інфраструктури мережі. Структура MPLS є гнучкою і може бути використана в будь-яких комбінаціях з технологіями другого рівня. Підтримка MPLS-комутації можлива для всіх протоколів 3-го рівня, а можливості розширення мережі значно перевершують ті, які зазвичай пропонуються в сучасних мережах. Технологія MPLS забезпечує ефективну передачу служб IP-з'єднанням мережі ATM і створення маршрутів між відправником і отримувачем в звичайній Internet-магістралі, заснованій на використанні тільки маршрутизаторів.

Первинною метою використання методу MPLS є створення гнучкої мережевої структури, яка забезпечує підвищену продуктивність і стабільність роботи мережі. Комутація по мітках дозволяє маршрутизаторів і комутаторів ATM з функціями MPLS приймати рішення про відправку пакетів шляхом аналізу вмісту простий мітки, замість використання досить складного алгоритму пошуку маршрутів, заснованого на IP-адресу одержувача.

Принцип роботи технології MPLS полягає в тому, що після організації IP зв'язності, на роутерах включається проектування по мітках (функція MPLS). Кожному маршруту присвоюється мітка. Відбувається формування таблиці міток (LIB - MPLS Label Information Base), яка містить інформацію про всі існуючі мітках (навіть про мітки, які можуть і не використовуватися). Після цього, кожен роутер формує для себе іншу таблицю (LFIB - MPLS Label Forwarding Information Base),

згідно з якою він вже і буде здійснювати передачу трафіку. Іншими словами, роутера не треба більше переглядати свою таблицю маршрутизації (FIB), вибирати маршрут і так далі (якщо звичайно цей маршрут є в LFIB). Практично виходить, що роутер здійснює комутацію, а не маршрутизацію, що в свою чергу дуже прискорює роботу з доставки трафіку в місце призначення. Поверх MPLS можуть працювати різні протоколи 2-го і 3-го рівнів еталонної моделі OSI.

Перевірка роботи аналітичної моделі активного обладнання виконана на основі типової мережі з використанням технології багатопротокольної комутації по мітках MPLS. Зазвичай більшість комерційних організацій, в тому числі банки, мають головний офіс та мережу філіалів. Ці філіали можуть знаходитися як в межах одного населеного пункту, так і в різних містах [9]. Вочевидь існує необхідність в швидкій передачі інформації між головним офісом та філіалами. Тому є актуальним питання підвищення ефективності роботи активного обладнання (маршрутизаторів), для збільшення швидкості передачі інформації будемо розглядати мінімально можливу конфігурацію, яка складається з одного головного офісу та одного філіалу. Для цього побудована модель Ethernet мережі і створена схема з Layer 2 VPN з інкапсуляцією MPLS яка представлена на рис. 3.1.

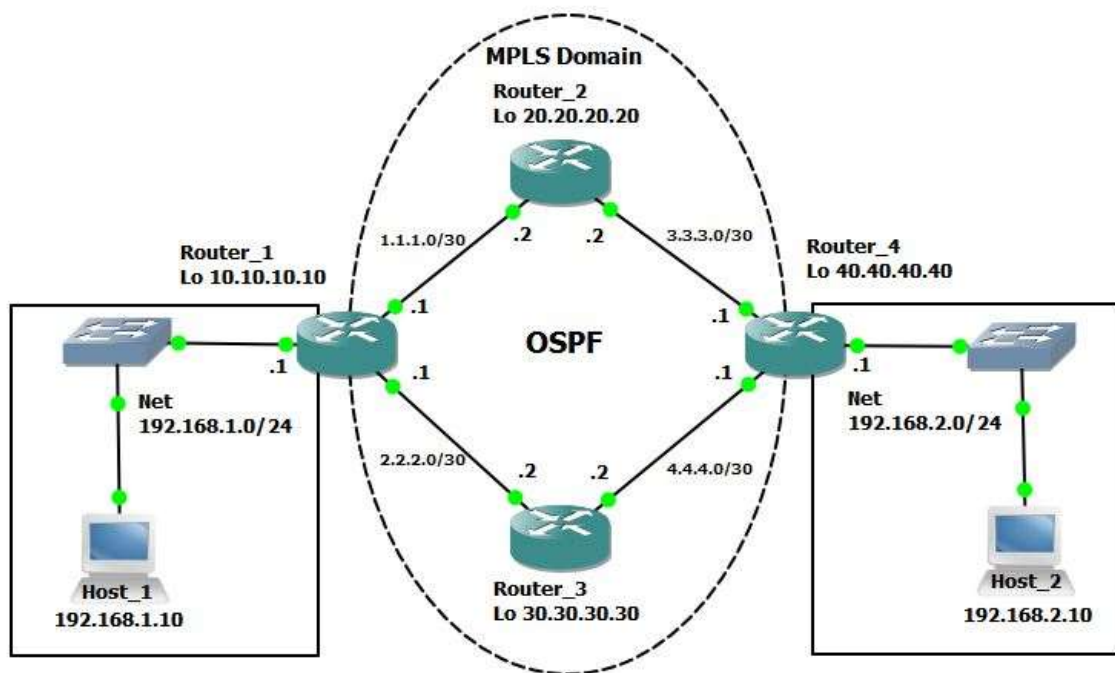


Рисунок 3.1 – Топологія мережі

На схемі представлені 4 роутера. Router_1 і Router_4 є прикордонними ELSR - Edge Label Switch Router. Вони використовуються для «навішування» (pushing) міток при вході пакета в домен MPLS і для «зняття» (popping) міток при виході з домена MPLS. Роутери Router_2 і Router_3 є внутрішніми роутерами домену MPLS і називаються LSR – Label Switch Router. Вони займаються тільки передачею пакетів від одного MPLS роутера іншому.

Для організації мережевої доступності (IP) був використаний протокол динамічної маршрутизації OSPF. Хости для перевірки розташовані в різних мережах. Першою дією в процесі аналізу було налаштування на стандартну мережевий доступності. Для цього на роутері Router_1 були виконані наступні конфігурації:

```
R1>en
```

```
R1#conf t
```

```
R1(config)#hostname Router_1
```

Наступним порядком налаштуємо loopback інтерфейс:

```
R1(config)#int loopback 0
```

```
Router_1(config-if)#ip address 10.10.10.10 255.255.255.255
```

```
Router_1(config-if)#exit
```

```
Router_1(config)#int fa 0/0
```

```
Router_1(config-if)#ip address 192.168.1.1 255.255.255.0
```

```
Router_1(config-if)#no sh
```

```
Router_1(config-if)#exit
```

```
Router_1(config)#int fa 1/0
```

```
Router_1(config-if)#ip address 1.1.1.1 255.255.255.252
```

```
Router_1(config-if)#no sh
```

```
Router_1(config-if)#exit
```

```
Router_1(config)#int fa 2/0
```

```
Router_1(config-if)#ip address 2.2.2.1 255.255.255.252
```

```
Router_1(config-if)#no sh
```

```
Router_1(config-if)#exit
```

Далі безпосередньо налаштовувався протокол динамічної маршрутизації OSPF:

```
Router_1(config)#router ospf
```

Після цього були обрані інтерфейси на які будуть надходити чергові пакети OSPF:

```
Router_1(config-router)#passive-interface default
```

```
Router_1(config-router)#no passive-interface fastEthernet 1/0
```

```
Router_1(config-router)#no passive-interface fastEthernet 2/0
```

Потім визначені мережі для оголошення:

```
Router_1(config-router)#network 192.168.1.0 0.0.0.255 area
```

```
Router_1(config-router)#network 1.1.1.0 0.0.0.3 area 0
```

```
Router_1(config-router)#network 2.2.2.0 0.0.0.3 area 0
```

```
Router_1(config-router)#network 10.10.10.10 0.0.0.0 area 0
```

```
Router_1(config-router)#exit
```

```
Router_1(config)#exit
```

```
Router_1#wr
```

```
Router_1#
```

На роутер Router 2 були налаштовані наступні конфігурації:

```
R2>en
```

```
R2#conf t
```

```
R2(config)#hostname Router_2
```

```
Router_2(config)#int loopback 0
```

```
Router_2(config-if)#ip address 20.20.20.20 255.255.255.255
```

```
Router_2(config-if)#exit
```

```
Router_2(config)#int fa 0/0
```

```
Router_2(config-if)#ip address 1.1.1.2 255.255.255.252
```

```
Router_2(config-if)#no sh
```

```
Router_2(config-if)#exit
```

```
Router_2(config)#int fa 1/0
```

```
Router_2(config-if)#ip address 3.3.3.2 255.255.255.252
```

```
Router_2(config-if)#no sh
```

```
Router_2(config-if)#exit
```

```
Router_2(config)#router ospf 1
Router_2(config-router)#network 1.1.1.0 0.0.0.3 area 0
Router_2(config-router)#network 3.3.3.0 0.0.0.3 area 0
Router_2(config-router)#network 20.20.20.20 0.0.0.0 area 0
Router_2(config-router)#exit
Router_2(config)#exit
Router_2#wr
Router_2#
```

Для Роутера Router 3 настройка виглядає наступним чином:

```
R3>en
R3#conf t
R3(config)#hostname Router_3
Router_3(config)#int loopback 0
Router_3(config-if)#ip address 30.30.30.30 255.255.255.255
Router_3(config-if)#exit
Router_3(config)#int fa 0/0
Router_3(config-if)#ip address 2.2.2.2 255.255.255.252
Router_3(config-if)#no sh
Router_3(config-if)#exit
Router_3(config)#int fa 1/0
Router_3(config-if)#ip address 4.4.4.2 255.255.255.252
Router_3(config-if)#no sh
Router_3(config-if)#exit
Router_3(config)#router ospf 1
Router_3(config-router)#network 2.2.2.0 0.0.0.3 area 0
Router_3(config-router)#network 4.4.4.0 0.0.0.3 area 0
Router_3(config-router)#network 30.30.30.30 0.0.0.0 area 0
Router_3(config-router)#exit
Router_3(config)#exit
Router_3#wr
```

Router_3#

І роутер Router 4 був налаштований відповідно:

R4>en

R4#conf t

R4(config)#hostname Router_4

Router_4(config)#int loopback 0

Router_4(config-if)#ip address 40.40.40.40 255.255.255.255

Router_4(config-if)#exit

Router_4(config)#int fa 0/0

Router_4(config-if)#ip address 3.3.3.1 255.255.255.252

Router_4(config-if)#no sh

Router_4(config-if)#exit

Router_4(config)#int fa 1/0

Router_4(config-if)#ip address 4.4.4.1 255.255.255.252

Router_4(config-if)#no sh

Router_4(config-if)#exit

Router_4(config)#int fa 2/0

Router_4(config-if)#ip address 192.168.2.1 255.255.255.0

Router_4(config-if)#no sh

Router_4(config-if)#exit

Router_4(config)#router ospf 1

Router_4(config-router)#passive-interface default

Router_4(config-router)#no passive-interface fastEthernet 0/0

Router_4(config-router)#no passive-interface fastEthernet 1/0

Router_4(config-router)#network 192.168.2.0 0.0.0.255 area 2

Router_4(config-router)#network 3.3.3.0 0.0.0.3 area 0

Router_4(config-router)#network 4.4.4.0 0.0.0.3 area 0

Router_4(config-router)#network 40.40.40.40 0.0.0.0 area 0

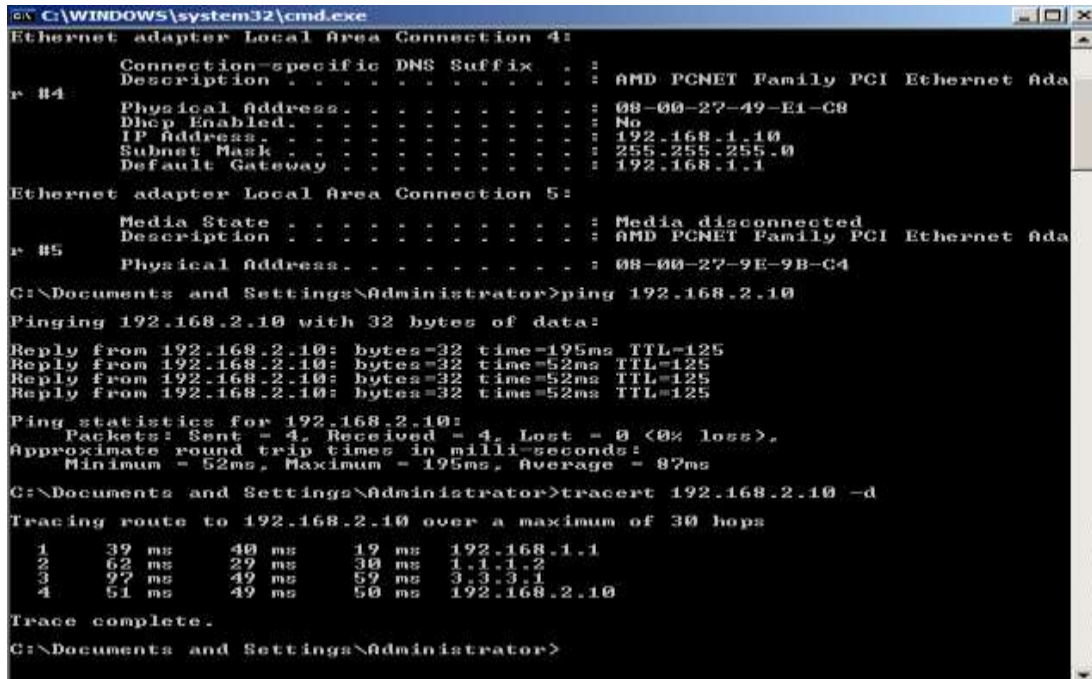
Router_4(config-router)#exit

Router_4(config)#exit

Router_4#wr

Router_4#

Після налаштування необхідних конфігурацій була перевірена мережева доступність за допомогою команд ping і tracert. Результат виконання цих команд наведено на рис. 3.2:



```
C:\WINDOWS\system32\cmd.exe
Ethernet adapter Local Area Connection 4:
   Connection-specific DNS Suffix  . : 
   Description . . . . .           : AMD PCNET Family PCI Ethernet Ada
   r #4
   Physical Address. . . . .       : 08-00-27-49-E1-C8
   Dhcp Enabled. . . . .           : No
   IP Address. . . . .             : 192.168.1.10
   Subnet Mask . . . . .           : 255.255.255.0
   Default Gateway . . . . .       : 192.168.1.1

Ethernet adapter Local Area Connection 5:
   Media State . . . . .           : Media disconnected
   Description . . . . .           : AMD PCNET Family PCI Ethernet Ada
   r #5
   Physical Address. . . . .       : 08-00-27-9E-9B-C4

C:\Documents and Settings\Administrator>ping 192.168.2.10
Pinging 192.168.2.10 with 32 bytes of data:
Reply from 192.168.2.10: bytes=32 time=195ms TTL=125
Reply from 192.168.2.10: bytes=32 time=52ms TTL=125
Reply from 192.168.2.10: bytes=32 time=52ms TTL=125
Reply from 192.168.2.10: bytes=32 time=52ms TTL=125

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 52ms, Maximum = 195ms, Average = 87ms

C:\Documents and Settings\Administrator>tracert 192.168.2.10 -d
Tracing route to 192.168.2.10 over a maximum of 30 hops:
  0  39 ms  40 ms  19 ms  192.168.1.1
  1  62 ms  29 ms  30 ms  1.1.1.2
  2  97 ms  49 ms  59 ms  3.3.3.1
  3  51 ms  49 ms  50 ms  192.168.2.10

Trace complete.

C:\Documents and Settings\Administrator>
```

Рисунок 3.2 – Налаштування мережевої доступності

Як видно, на рис. 3.2 мережевий доступ присутній.

Наступним дією була налаштована технологія MPLS. Для цього на роутер Router_1 були додані наступні команди:

Router_1#conf t

Була активована технологія швидкої комутації пакетів на 3-му рівні Cisco Express Forwarding:

Router_1(config)#ip cef

Підключення процесу комутації по мітках MPLS:

Router_1(config)#mpls ip

Обрано протокол за яким будуть обмінюватися нотатками таблиці LSR:

Router_1(config)#mpls label protocol ldp

Тут вказуємо який інтерфейс (IP-адреса) береться в якості ID роутера в процесі MPLS:

```
Router_1(config)#mpls ldp router-id loopback 0
```

```
Router_1(config)#int fa 1/0
```

Безпосередньо запуск MPLS на інтерфейсі:

```
Router_1(config-if)#mpls
```

Заданий новий, більш об'ємний розмір mtu, для уникнення фрагментації фреймів:

```
Router_1(config-if)#mpls mtu 1512
```

```
Router_1(config-if)#exit
```

```
Router_1(config)#int fa 2/0
```

```
Router_1(config-if)#mpls ip
```

```
Router_1(config-if)#mpls mtu 1512
```

```
Router_1(config-if)#exit
```

```
Router_1(config)#exit
```

```
Router_1#wr
```

```
Router_1#
```

Аналогічно був налаштований Router_2:

```
Router_2#conf t
```

```
Router_2(config)#ip cef
```

```
Router_2(config)#mpls ip
```

```
Router_2(config)#mpls label protocol ldp
```

```
Router_2(config)#mpls ldp router-id loopback 0
```

```
Router_2(config)#int fa 0/0
```

```
Router_2(config-if)#mpls ip
```

```
Router_2(config-if)#mpls mtu 1512
```

```
Router_2(config-if)#exit
```

```
Router_2(config)#int fa 1/0
```

```
Router_2(config-if)#mpls ip
```

```
Router_2(config-if)#mpls mtu 1512
```

```
Router_2(config-if)#exit
```

```
Router_2(config)#exit
Router_2#wr
Router_2#
Потім Router_3:
Router_3#conf t
Router_3(config)#ip cef
Router_3(config)#mpls ip
Router_3(config)#mpls label protocol ldp
Router_3(config)#mpls ldp router-id loopback 0
Router_3(config)#int fa 0/0
Router_3(config-if)#mpls ip
Router_3(config-if)#mpls mtu 1512
Router_3(config-if)#exit
Router_3(config)#int fa 1/0
Router_3(config-if)#mpls ip
Router_3(config-if)#mpls mtu 1512
Router_3(config-if)#exit
Router_3(config)#exit
Router_3#wr
І нарешті Router_4:
Router_4#conf t
Router_4(config)#ip cef
Router_4(config)#mpls ip
Router_4(config)#mpls label protocol ldp
Router_4(config)#mpls ldp router-id loopback 0
Router_4(config)#int fa 0/0
Router_4(config-if)#mpls ip
Router_4(config-if)#mpls mtu 1512
Router_4(config-if)#exit
Router_4(config)#int fa 1/0
```

```

Router_4(config-if)#mpls ip
Router_4(config-if)#mpls mtu 1512
Router_4(config-if)#exit
Router_4(config)#exit
Router_4#wr
Router_4#

```

Далі була виконана перевірка виконаних дій командами групи show. Для прикладу було взято Router_1. Вилонені пакети (при проходженні ping) на інтерфейсі FastEthernet 1/0 роутера Router_2 (дивиться в сторону Router_4) видно на рис. 3.3:

Загальний результат роботи над завданням створення топології показаний на рис. 3.3 (настройка MPLS):

```

R1
Router_1#
Router_1#sh mpls forwarding-table
Local   Outgoing  Prefix          Bytes Label  Outgoing  Next Hop
Label   Label     or Tunnel Id    Switched     interface
16      Pop Label  3.3.3.0/30      0            Fai/0     1.1.1.2
17      Pop Label  4.4.4.0/30      0            Fai/1     2.2.2.2
18      Pop Label  20.20.20.20/32  0            Fai/0     1.1.1.2
19      Pop Label  30.30.30.30/32  0            Fai/1     2.2.2.2
20      20        40.40.40.40/32  0            Fai/0     1.1.1.2
20      20        40.40.40.40/32  0            Fai/1     2.2.2.2
21      22        192.168.2.0/24  0            Fai/0     1.1.1.2
21      22        192.168.2.0/24  0            Fai/1     2.2.2.2
Router_1#sh mpls ldp neighbor
Peer LDP Ident: 20.20.20.20:0; Local LDP Ident 10.10.10.10:0
TCP connection: 20.20.20.20.48434 - 10.10.10.10.646
State: Oper; Msgs sent/rcvd: 21/21; Downstream
Up time: 00:07:55
LDP discovery sources:
FastEthernet1/0, Src IP addr: 1.1.1.2
Addresses bound to peer LDP Ident:
1.1.1.2      3.3.3.2      20.20.20.20
Peer LDP Ident: 30.30.30.30:0; Local LDP Ident 10.10.10.10:0
TCP connection: 30.30.30.30.55010 - 10.10.10.10.646
State: Oper; Msgs sent/rcvd: 19/19; Downstream
Up time: 00:05:37
LDP discovery sources:
FastEthernet1/1, Src IP addr: 2.2.2.2
Addresses bound to peer LDP Ident:
2.2.2.2      4.4.4.2      30.30.30.30
Router_1#

```

Рисунок 3.3 – Результат налаштування MPLS на роутерах 1–4

З рис. 3.3 видно що всі маршрути позначені мітками і видно всі «сусідні» роутери.

Нижче, на рис. 3.4, представлена таблиця всіх міток і маршрутів:


```

R1
Router_1#sh mpls ip binding
1.1.1.0/30
  in label:      imp-null
  out label:     imp-null  lsr: 20.20.20.20:0
  out label:     16        lsr: 30.30.30.30:0
2.2.2.0/30
  in label:      imp-null
  out label:     16        lsr: 20.20.20.20:0
  out label:     imp-null  lsr: 30.30.30.30:0
3.3.3.0/30
  in label:      16
  out label:     imp-null  lsr: 20.20.20.20:0
  out label:     17        lsr: 30.30.30.30:0
4.4.4.0/30
  in label:      17
  out label:     17        lsr: 20.20.20.20:0
  out label:     imp-null  lsr: 30.30.30.30:0
10.10.10.10/32
  in label:      imp-null
  out label:     18        lsr: 20.20.20.20:0
  out label:     18        lsr: 30.30.30.30:0
20.20.20.20/32
  in label:      18
  out label:     imp-null  lsr: 20.20.20.20:0
  out label:     19        lsr: 30.30.30.30:0
30.30.30.30/32
  in label:      19
  out label:     19        lsr: 20.20.20.20:0
  out label:     imp-null  lsr: 30.30.30.30:0
40.40.40.40/32
  in label:      20
  out label:     20        lsr: 20.20.20.20:0
  out label:     20        lsr: 30.30.30.30:0
192.168.1.0/24
  in label:      imp-null
  out label:     21        lsr: 20.20.20.20:0
  out label:     21        lsr: 30.30.30.30:0
192.168.2.0/24
  in label:      21
  out label:     22        lsr: 20.20.20.20:0
  out label:     22        lsr: 30.30.30.30:0
Router_1#

```

Рисунок 3.4 – Таблиця міток і маршрутів

За допомогою програми моніторингу трафіку Wireshark було розглянуто вміст відловлених пакетів. Результат роботи програми представлений на рис. 3.5:

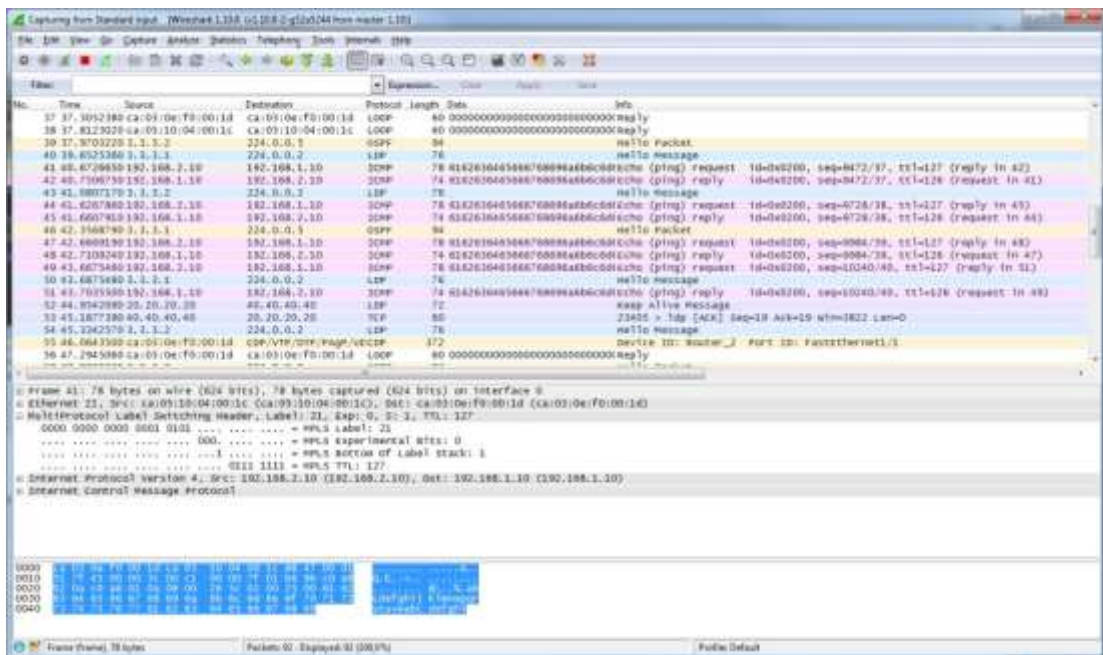


Рисунок 3.5 – Вміст відстежених пакетів

З відстежених пакетів візьмемо ICMP. І, як видно з рис. 3.5, що у нас з'явилася закладка MultiProtocol Label Switching Header. У ній показані різні параметри, які вставляються у фрейм LDP пакети (hello messages) та розсилаються роутерами на мультикаст адреси 224.0.0.2 [11].

Вивчення рис. 3.3–3.5 показало що базова конфігурація MPLS працює. Після цього була налаштована L2 VPN over MPLS. Для налаштування запропонованої технології була перетворена топологія мережі. Вона прийняла такий вигляд (рис. 3.6):

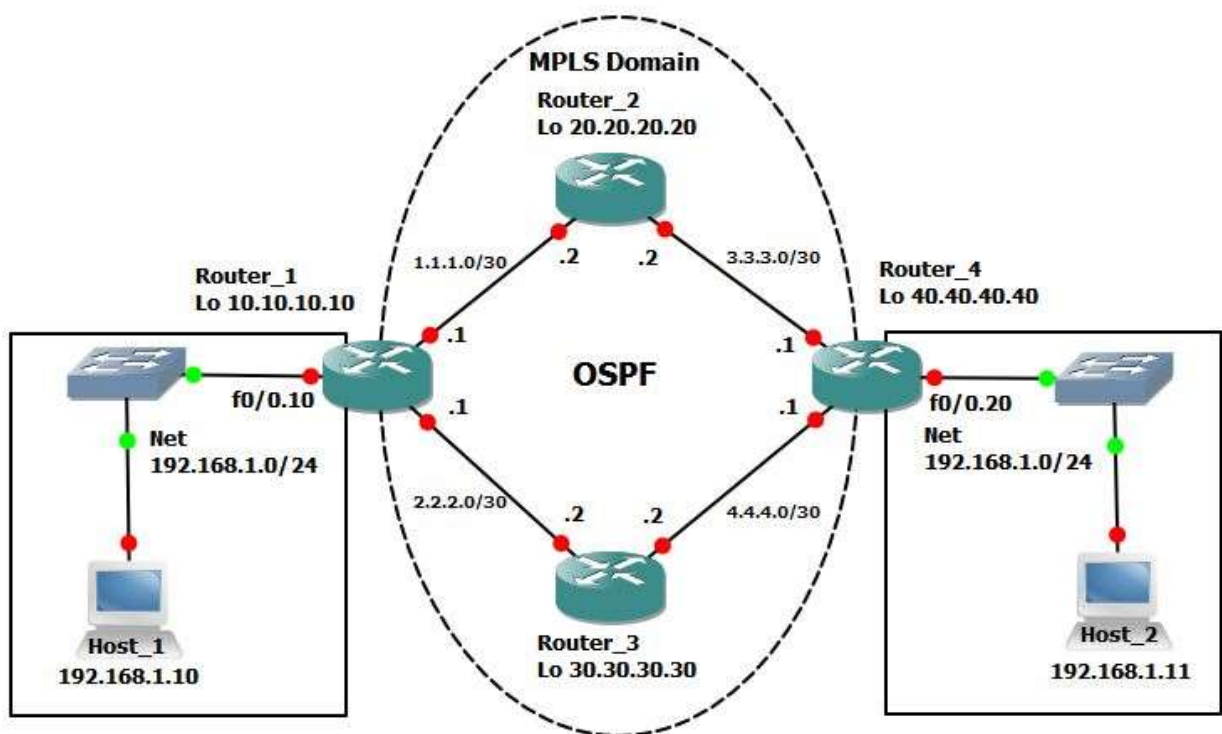


Рисунок 3.6 – Модернізована топологія мережі

Використовувалися ті ж 4 роутери, але на прикордонних маршрутизаторах, з'явилися sub-інтерфейси, які відповідають деяким vlan-ам (10 і 20 відповідно). І тепер, станції зайняли розташування в одній мережі 192.168.1.0/24. Завдання ставилося так, щоб Host_1 і Host_2 бачили один одного як ніби вони знаходяться в одному vlan-е. Для виконання поставленої задачі використана L2 VPN з інкапсуляція MPLS (l2mpls).

Змінивши топологію мережі на роутер Router_1 були зроблені наступні настройки:

```
R1>en
R1#conf t
R1(config)#hostname Router_1
Router_1(config)#ip cef
Router_1(config)#mpls ip
Router_1(config)#mpls label protocol ldp
Router_1(config)#mpls ldp router-id loopback 0
Router_1(config)#int loopback 0
Router_1(config-if)#ip address 10.10.10.10 255.255.255.255
Router_1(config-if)#exit
Router_1(config)#int fa 1/0
Router_1(config-if)#ip address 1.1.1.1 255.255.255.252
Router_1(config-if)#no sh
Router_1(config-if)#mtu 1512
Router_1(config-if)#mpls ip
Router_1(config-if)#mpls mtu 1512
Router_1(config-if)#exit
Router_1(config)#int fa 1/1
Router_1(config-if)#ip address 2.2.2.1 255.255.255.252
Router_1(config-if)#no sh
Router_1(config-if)#mtu 1512
Router_1(config-if)#mpls ip
Router_1(config-if)#mpls mtu 1512
Router_1(config-if)#exit
Router_1(config)#router ospf 1
Router_1(config-router)#passive-interface default
Router_1(config-router)#no passive-interface fastEthernet 1/0
Router_1(config-router)#no passive-interface fastEthernet 1/1
Router_1(config-router)#network 10.10.10.10 0.0.0.0 area 0
Router_1(config-router)#network 1.1.1.0 0.0.0.3 area 0
```

```
Router_1(config-router)#network 2.2.2.0 0.0.0.3 area 0
```

```
Router_1(config-router)#exit
```

```
Router_1(config)#int fa 0/0
```

```
Router_1(config-if)#no sh
```

```
Router_1(config-if)#exit
```

Создан sub-интерфейс для vlan 10:

```
Router_1(config)#int fa 0/0.10
```

Настроена инкапсуляция dot1q с тегом 10 (№ vlan):

```
Router_1(config-subif)#encapsulation dot1Q 10
```

Створено спеціальне підключення до другого роутера. Вказано метод інкапсуляції mpls. 9999 - це номер віртуального каналу VC (Virtual Circuit). Він повинен бути однаковим з двох сторін і ідентифікувати сам канал:

```
Router_1(config-subif)#xconnect 40.40.40.40 9999 encapsulation mpls
```

```
Router_1(config-subif-xconn)#exit
```

```
Router_1(config-subif)#nosh
```

```
Router_1(config-subif)#exit
```

```
Router_1(config)#exit
```

```
Router_1#wr
```

```
Router_1#
```

Після, відповідні налаштування були застосовані до Router_2:

```
R2>en
```

```
R2#conf t
```

```
R2(config)#hostname Router_2
```

```
Router_2(config)#ip cef
```

```
Router_2(config)#mpls ip
```

```
Router_2(config)#mpls label protocol ldp
```

```
Router_2(config)#mpls ldp router-id loopback 0
```

```
Router_2(config)#int loopback 0
```

```
Router_2(config-if)#ip address 20.20.20.20 255.255.255.255
```

```
Router_2(config-if)#exit
```

```
Router_2(config)#int fa 1/0
Router_2(config-if)#ip address 1.1.1.2 255.255.255.252
Router_2(config-if)#no sh
Router_2(config-if)#mtu 1512
Router_2(config-if)#mpls ip
Router_2(config-if)#mpls mtu 1512
Router_2(config-if)#exit
Router_2(config)#int fa 1/1
Router_2(config-if)#ip address 3.3.3.2 255.255.255.252
Router_2(config-if)#no sh
Router_2(config-if)#mpls ip
Router_2(config-if)#mtu 1512
Router_2(config-if)#mpls ip
Router_2(config-if)#mpls mtu 1512
Router_2(config-if)#exit
Router_2(config)#router ospf 1
Router_2(config-router)#network 20.20.20.20 0.0.0.0 area 0
Router_2(config-router)#network 1.1.1.0 0.0.0.3 area 0
Router_2(config-router)#network 1.1.1.0 0.0.0.3 area 0
Router_2(config-router)#network 3.3.3.0 0.0.0.3 area 0
Router_2(config-router)#exit
Router_2(config)#exit
Router_2#wr
Router_2#
```

Далі був налаштований Router_3:

```
R3>en
R3#conf t
R3(config)#hostname Router_3
Router_3(config)#ip cef
```

```
Router_3(config)#mpls ip
Router_3(config)#mpls label protocol ldp
Router_3(config)#mpls ldp router-id loopback 0
Router_3(config)#int loopback 0
Router_3(config-if)#ip address 30.30.30.30 255.255.255.255
Router_3(config-if)#exit
Router_3(config)#int fa 1/0
Router_3(config-if)#ip address 2.2.2.2 255.255.255.252
Router_3(config-if)#no sh
Router_3(config-if)#mtu 1512
Router_3(config-if)#mpls ip
Router_3(config-if)#mpls mtu 1512
Router_3(config-if)#exit
Router_3(config)#int fa 1/1
Router_3(config-if)#ip address 4.4.4.2 255.255.255.252
Router_3(config-if)#no sh
Router_3(config-if)#mtu 1512
Router_3(config-if)#mpls ip
Router_3(config-if)#mpls mtu 1512
Router_3(config-if)#exit
Router_3(config)#router ospf 1
Router_3(config-router)#network 30.30.30.30 0.0.0.0 area 0
Router_3(config-router)#network 2.2.2.0 0.0.0.3 area 0
Router_3(config-router)#network 4.4.4.0 0.0.0.3 area 0
Router_3(config-router)#exit
Router_3(config)#exit
Router_3#wr
Router_3#
Та останні зміни було проведено на Router_4:
R4>en
```

```
R4#conf t
R4(config)#hostname Router_4
Router_4(config)#ip cef
Router_4(config)#mpls ip
Router_4(config)#mpls label protocol ldp
Router_4(config)#mpls ldp router-id loopback 0
Router_4(config)#int loopback 0
Router_4(config-if)#ip address 40.40.40.40 255.255.255.255
Router_4(config-if)#exit
Router_4(config)#int fa 1/0
Router_4(config-if)#ip address 4.4.4.1 255.255.255.252
Router_4(config-if)#no sh
Router_4(config-if)#mtu 1512
Router_4(config-if)#mpls ip
Router_4(config-if)#mpls mtu 1512
Router_4(config-if)#exit
Router_4(config)#int fa 1/1
Router_4(config-if)#ip address 3.3.3.1 255.255.255.252
Router_4(config-if)#no sh
Router_4(config-if)#mtu 1512
Router_4(config-if)#mpls ip
Router_4(config-if)#mpls mtu 1512
Router_4(config-if)#exit
Router_4(config)#router ospf 1
Router_4(config-router)#passive-interface default
Router_4(config-router)#no passive-interface fastEthernet 1/0
Router_4(config-router)#no passive-interface fastEthernet 1/1
Router_4(config-router)#network 40.40.40.40 0.0.0.0 area 0
Router_4(config-router)#network 3.3.3.0 0.0.0.3 area 0
Router_4(config-router)#network 4.4.4.0 0.0.0.3 area 0
```

```

Router_4(config-router)#exit
Router_4(config)#int fa 0/0
Router_4(config-if)#no sh
Router_4(config-if)#exit
Router_4(config)#int fa 0/0.20
Router_4(config-subif)#encapsulation dot1Q 20
Router_4(config-subif)#xconnect 10.10.10.10 9999 encapsulation mpls
Router_4(config-subif-xconn)#exit

```

Після налаштування задіяного обладнання було отримано наступне повідомлення:

```
*Mar 22 15:05:48.439: %LDP-5-NBRCHG: LDP Neighbor 10.10.10.10:0 (3) is UP
```

```
Router_4(config-subif)#exit
```

```
Router_4(config)#exit
```

```
Router_4#wr
```

```
Router_4#
```

Використовуючи команду ping був проведений аналіз даних на Router_1 і Router_4.

Результат виконання представлений на рис. 3.7 і 3.8:

```

R1
Router_1#
Router_1#sh mpls l2transport vc
-----
Local intf   Local circuit   Dest address   VC ID   Status
-----
Fa0/0.10    Eth VLAN 10    40.40.40.40    9999    UP

Router_1#sh mpls l2transport bind
Destination Address: 40.40.40.40, VC ID: 9999
Local Label: 22
  Cbit: 1,   VC Type: Eth VLAN,   GroupID: 0
  MTU: 1500,   Interface Desc: n/a
  VCCV: CC Type: CW [1], RA [2], TTL [3]
  CV Type: LSPV [2], BFD/Baw [5]
Remote Label: 22
  Cbit: 1,   VC Type: Eth VLAN,   GroupID: 0
  MTU: 1500,   Interface Desc: n/a
  VCCV: CC Type: CW [1], RA [2], TTL [3]
  CV Type: LSPV [2], BFD/Baw [5]

Router_1#sh mpls forwarding-table
-----
Local   Outgoing   Prefix          Bytes Label   Outgoing   Next Hop
Label   Label      or Tunnel Id   Switched      interface
-----
16      Pop Label  3.3.3.0/30     0             Fa1/0      1.1.1.2
17      Pop Label  4.4.4.0/30     0             Fa1/1      2.2.2.2
18      Pop Label  20.20.20.20/32 0             Fa1/0      1.1.1.2
19      Pop Label  30.30.30.30/32 0             Fa1/1      2.2.2.2
20      20        40.40.40.40/32 0             Fa1/0      1.1.1.2
30      30        40.40.40.40/32 0             Fa1/1      2.2.2.2
21      22        192.168.2.0/24 0             Fa1/0      1.1.1.2
22      22        192.168.2.0/24 0             Fa1/1      2.2.2.2
22      No Label  12cct(1)       0             Fa0/0.10   point2point
Router_1#

```

Рисунок 3.7 – Конфігурація Router_1


```

Router_4#sh mpls l2transport vc
-----
Local intf      Local circuit      Dest address      VC ID      Status
-----
Fa0/0.20       Eth VLAN 20       10.10.10.10      9999       UP

Router_4#sh mpls l2transport binding
Destination Address: 10.10.10.10, VC ID: 9999
Local Label: 22
  Cbit: 1, VC Type: Eth VLAN, GroupID: 0
  MTU: 1500, Interface Desc: n/a
  VCCV: CC Type: CW [1], RA [2], TTL [3]
  CV Type: LSPV [2], BFD/Raw [5]
Remote Label: 22
  Cbit: 1, VC Type: Eth VLAN, GroupID: 0
  MTU: 1500, Interface Desc: n/a
  VCCV: CC Type: CW [1], RA [2], TTL [3]
  CV Type: LSPV [2], BFD/Raw [5]

Router_4#sh mpls forwarding-table
Local   Outgoing Prefix      Bytes Label  Outgoing Next Hop
Label   Label   or Tunnel Id Switched     interface
-----
16      Pop Label 1.1.1.0/30  0           Fa1/0      3.3.3.2
17      Pop Label 2.2.2.0/30  0           Fa1/1      4.4.4.2
18      18        10.10.10.10/32 0           Fa1/0      3.3.3.2
18      18        10.10.10.10/32 0           Fa1/1      4.4.4.2
19      Pop Label 20.20.20.20/32 0           Fa1/0      3.3.3.2
20      Pop Label 30.30.30.30/32 0           Fa1/1      4.4.4.2
21      21        192.168.1.0/24 0           Fa1/0      3.3.3.2
21      21        192.168.1.0/24 0           Fa1/1      4.4.4.2
22      No Label  L2ckt(1)     0           Fa0/0.20   point2point
Router_4#

```

Рисунок 3.8 – Конфігурація Router_4

З наведених вище рисунків видно, що l2 канал встановлений, тип цього каналу Ethernet, зіставлений він з потрібним інтерфейсом, показані IP-адреси «сусідів», ID віртуального каналу (vc) а його статус UP (активний). Так само, можна побачити, які локальні мітки навесив кожен роутер. Далі, виконана команда ping Host_2 на Host_1, результат якої видно на рисунку 3.9:

```

C:\WINDOWS\system32\cmd.exe
Ethernet adapter Local Area Connection 4:

    Connection-specific DNS Suffix  . : 
    Description . . . . .           : AMD PCNET Family PCI Ethernet Adapte
r #4
    Physical Address. . . . .       : 08-00-27-49-E1-C8
    Dhcp Enabled. . . . .           : No
    IP Address. . . . .             : 192.168.1.10
    Subnet Mask . . . . .           : 255.255.255.0
    Default Gateway . . . . .       : 192.168.1.1

Ethernet adapter Local Area Connection 5:

    Media State . . . . .           : Media disconnected
    Description . . . . .           : AMD PCNET Family PCI Ethernet Adapte
r #5
    Physical Address. . . . .       : 08-00-27-9E-9B-C4

C:\Documents and Settings\Administrator>arp -a

Interface: 192.168.1.10 --- 0x2
    Internet Address      Physical Address      Type
    192.168.1.11         08-00-27-5d-20-30    dynamic

C:\Documents and Settings\Administrator>_

```

Рисунок 3.9 – Результат команди ping для Host_2 на Host_1

Для повноти картини, були виловлені кілька пакетів і розглянуто їх вміст. Перша точка - Int fa 1/1 на Router_4 представлена на рис. 3.10:

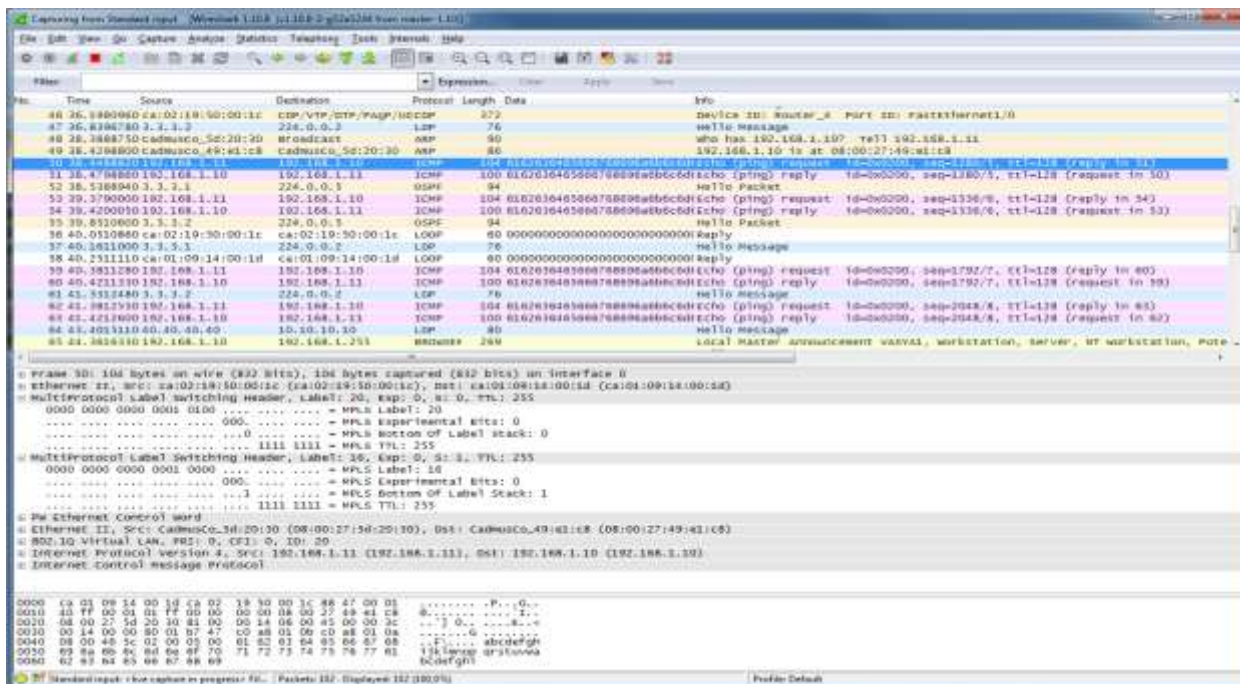


Рисунок 3.10 – Вміст відстежених пакетів на Router_4

Друга точка - Int fa 1 / 0 на Router_2 видно на рис. 3.11:

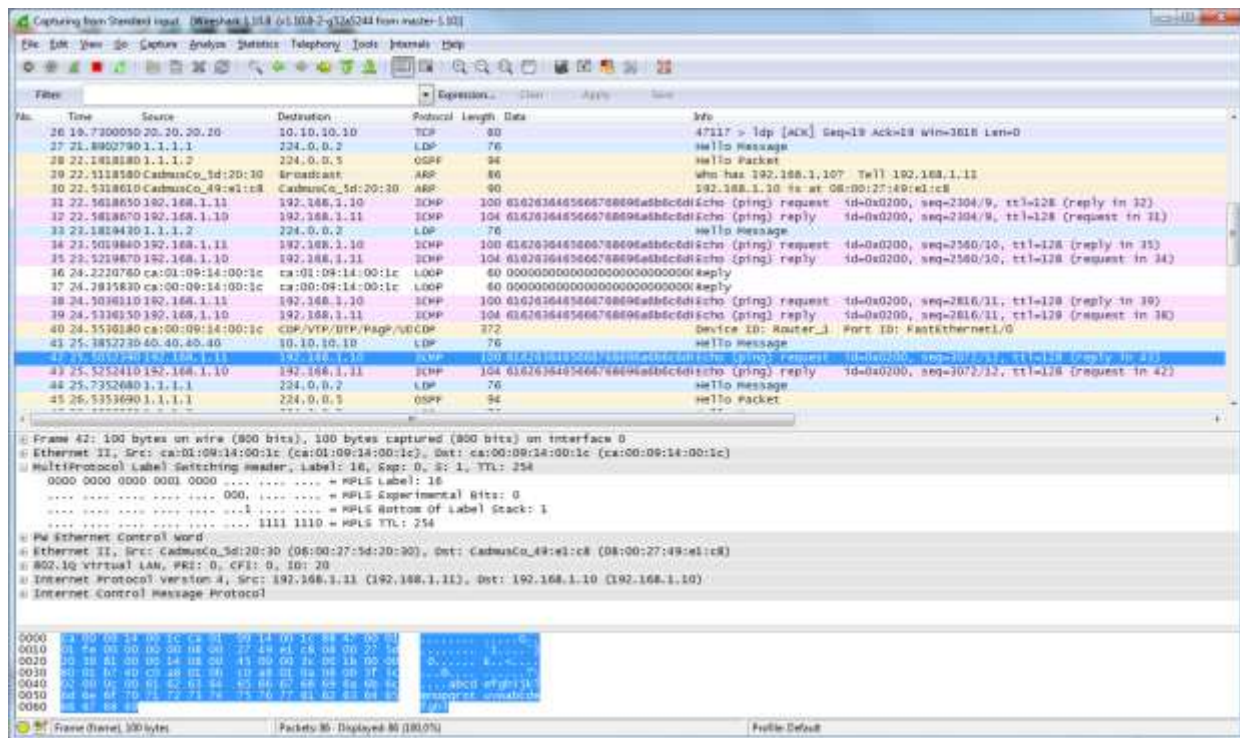


Рисунок 3.11 – Вміст відловлених пакетів на Router_2

Відповідно третя точка - Int fa0 / 0 на Router1 вказана на рис. 3.12:

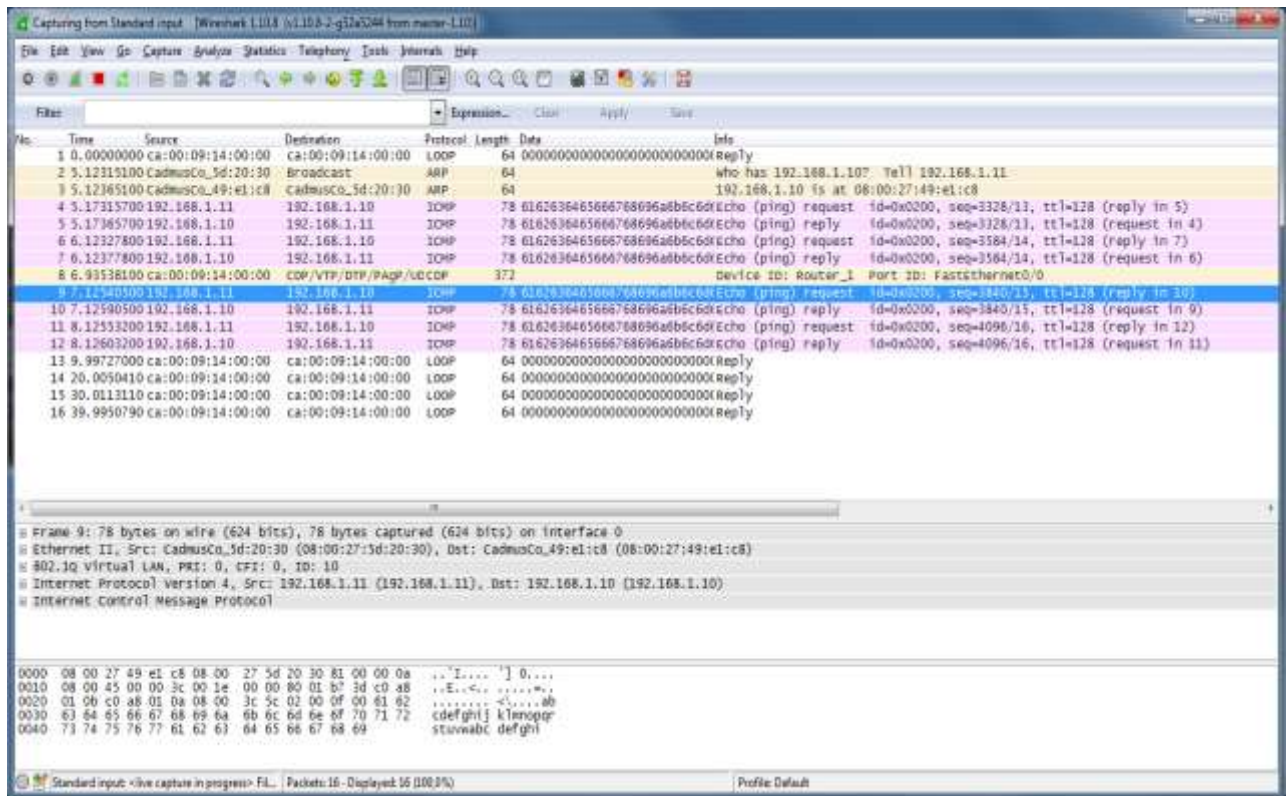


Рисунок 3.12 – Вміст відстежених пакетів на Router_1

Router_4, прийнявши пакет ICMP (Request) від Host_2 на інтерфейс Int fa 0/0.20, розуміє, що він відноситься до I2vpn mpls каналу і повинен бути переданий через мережу MPLS. Для цього він навішує на нього 2 мітки. Перша мітка використовується для комутації пакета в мережі MPLS, а друга мітка використовується для подальшої ідентифікації каналу vs. Обидві мітки мають ttl, рівний 255. Пакет відправляється далі. Router_2 приймає пакет і починає обробку. Він бачить, що пакет адресований мережі, яка знаходиться відразу за Router_1 і, щоб спростити завдання «сусідові» (технологія від'єднання на передостанньому LSR (PHP – Penultimate Hop Popping), знімає одну з міток, яка використовується для комутації. У другій мітки всього лише зменшується ttl на одиницю. пакет рухається далі по призначенню. Кінцевий роутер Router_1, приймаючи пакет, бачить, що транспортна мітка відсутня, але є мітка, яка відповідає наявному у нього віртуальному каналу vs. Він знімає цю мітку і, згідно з вже своєї внутрішньої таблиці, відправляє його на інтерфейс Int fa 0/0.10.

3.3 Порівняльна оцінка IP-маршрутизації та технології маршрутизації по міткам MPLS

Таким чином, було створено робочу топологію. Далі на даній моделі було проведено порівняння активного обладнання, а саме IP і MPLS маршрутизаторів.

Нижче розглянуто принцип функціонування і порядок обробки пакетів IP-маршрутизатором з точки зору обробки пакетів на 2-му і 3-му рівнях OSI, тобто частково рівень інтерфейсів і рівень мережевого протоколу. Нижче, на рис. 3.13 представлена функціональна модель IP-маршрутизатора.



Рисунок 3.13 – Функціональна модель IP-маршрутизатора

На рівні інтерфейсів перед IP-маршрутизатором ставляться наступні завдання:

- підрахунок контрольної суми (frame);
- призначення з кадру поля IP-пакету і його подальша передача для подальшої обробки на рівень мережевого протоколу, в разі коректної контрольної суми

упаковка IP-пакету в кадр (frame) відповідної технології канального рівня для подальшої передачі.

На рівні інтерфейсів IP-роутер має наступні функції:

- виділення з IP-пакета заголовка мережевого рівня;
- перевірка контрольної суми IP-пакету, довжина якої становить 16 біт;
- перевірка часу життя пакета (TTL) IP-пакету, довжина якого 8 біт;
- коригування часу життя пакета;
- перерахунок контрольної суми після модифікації відповідних полів;
- перевірка деяких полів за правилами фільтрації, якщо вони задані;
- визначення мережевої адреси наступного маршрутизатора і номера вхідного

інтерфейсу, до якого він безпосередньо підключений згідно з таблицею маршрутизації [18].

Розглянемо більш докладно порядок визначення адреси наступного маршрутизатора (визначення маршруту) за допомогою таблиці маршрутизації при використанні масок підмереж:

Видалення з вхідного IP-пакету адреси призначення IP-одержувача, який становить 32біт; Пошук специфічного маршруту. Шляхом порівняння IP одержувача, яка була видалена з вхідного пакета тільки з тими рядками адрес призначення таблиці маршрутизації P_N (32 біт) навпроти яких є запис 255.255.255.255 (маска підмережі), де i – номер рядка (запису) в таблиці маршрутизації ($i \in 1, K$). Для цього можна застосувати елементарну логічну операцію AND (логічне множення). Тобто IP-одержувач AND P_Ni : якщо є збіг то вибирається адреса наступного маршрутизатора, якщо немає виконується пошук неспецифічного маршруту.

Пошук неспецифічного маршруту, виконується шляхом накладення черзі масок адрес призначення M_i ; де i – номер рядка (запису) в таблиці маршрутизації ($i \in 1, K$), починаючи з першого рядка, на адресу призначення IP-одержувача вхідного IP-пакету. При цьому застосовується наступне правило IP одержувача AND M_i : всі збіги фіксуються, і здійснюється перегляд усієї таблиці маршрутизації.

Видача значення наступного маршрутизатора і номера вхідного інтерфейсу:

- якщо немає збігів, то використовується шлях за замовчуванням (Default router);

- якщо єдиний збіг (в результаті накладення масок на IP-одержувача) то видається значення адреси призначення і вихідного інтерфейсу, відповідне цьому рядку;

- якщо кілька збігів, тоді серед них обирається ті, які мали найбільше збіг двійкових розрядів.

Згідно вище наведеного кожному IP-маршрутизатора необхідно виконання наступних обов'язкових дій при обробці полів IP-паketу, в разі виконання прийнятих припущень і обмежень:

- перевірка контрольної суми IP-паketу KS здійснюється шляхом порівняння значення паketу що надійшов з реальною довжиною заголовка IP пакета. Причому перед початком розрахунку значення контрольної суми встановлюється в нульове значення. Якщо в результаті порівняння воно змінюється, то паket відкидається;

- перевірка часу життя пакета TTL, якщо TTL - 0 паket відкидається назад, ця операція не враховується;

- уоригування часу життя пакета, за рахунок віднімання одиниці від молодшого розряду;

- пошук специфічного маршруту;

- пошук більш ніж специфічного маршруту.

Для проведення кількісної оцінки, а саме визначення асимптотичної часової складності, можна застосувати бітну модель, яка базується на рівномірній ваговій функції. Дану модель також називають бітними розрахунками, причому всі змінні приймають значення "0" або "1", використовуються логічні операції (AND; OR; NOT AND).

Тобто, з огляду на порядок обробки полів IP-паketів маршрутизаторами можна застосувати наступні формули, засновані на використанні елементарних функцій:

1) перевірка контрольної суми KS:

$$KS = KS_{\text{вхідна}} \text{NOTAND} KS_{\text{розрахункова}} = 0 \quad (3.1)$$

де NOTAND - операція "І-НІ" (зворотна до логічного множення), $KS_{\text{вхідна}}$ – значення контрольної суми заголовки, яку було вилучено з відповідного поля вхідного макета; $KS_{\text{розрахункова}}$ – значення контрольної суми заголовка, яке було визначено Корегування $TUL_{IP} = TUL_{IP} \text{OR}$ одиниця (одиниця додається до молодшого розряду). де OR – логічне додавання для всіх розрядів даного поля в зв'язку з тим, що операція віднімання апаратно замінюється операцією додавання, а також може виникати операція перенесення в старший розряд.

2) пошук специфічного маршруту:

$$SR = IP_{\text{отримувача}} \text{AND} PN_i \quad (3.2)$$

де i – номер рядка (запису) в таблиці маршрутизації ($i \in 1, K$) навпроти якого є запис 255.255.255.255. Тобто кількість таких рядків (i) залежить від кількості специфічних маршрутів.

3) пошук неспецифічного маршруту:

$$NSR = IP_{\text{одержувача}} \text{AND} M_i \quad (3.3)$$

де i – номер рядка (запису) в таблиці маршрутизації ($i \in 1, K$), причому K – загальна кількість рядків в таблиці маршрутизації.

Неважко помітити в разі, коли відсутні специфічні маршрути, а їх використання не є обов'язковим, найбільші розрахункові витрати припадають на визначення не специфічного маршруту (NSR), які в свою чергу прямо пропорційні у порівнянні з таблицею маршрутизації.

Тобто загальна кількість логічних операцій, які витрачаються при обробці пакетів одним IP-маршрутизатором, при прийнятих припущеннях, становить:

$$QBIP = KS + TTL_{IP} + NSP \quad (3.4)$$

Розглянемо принцип функціонування і порядок обробки пакетів прикордонним LER-маршрутизатором, архітектура якого представлена на рис. 3.14,

причому необхідно відзначити, що LSR використовують тільки таблицю просування, а таблицю маршрутизації не використовують. Це дозволяє зменшити кількість розрахунків.



Рисунок 3.14 – Архітектура LER-маршрутизатора (комутатора) і LSR-комутатора

Для проведення оцінки, в разі використання в якості критерію визначення кількості елементарних операцій можна навести такий аналітичний апарат (в разі застосування вхідного або вихідного LER і за умови, що згідно з протоколом LPD реалізовано оптимально):

Згідно таблиці маршрутизації при отриманні IP-паketу, LER виконуються всі ті ж самі дії, що і при IP-маршрутизації: Перевірка контрольної суми KS: $KS = KS_{\text{вхідні}} \text{ NOTAND } KS_{\text{розрахована}} = 0$; Де: NOTAND – операція "І-НІ" (зворотня до логічного множення) $KS_{\text{вхідні}}$ – значення контрольної суми заголовка, яку було вилучено з відповідного поля вхідного пакета; $KS_{\text{розрахована}}$ – значення контрольної суми заголовка, яке було визначено.

Коригування $TTL_{IP} = TTL_{IP} \text{ OR } \text{одиниця}$ (одиниця додається до молодшого розряду). Де OR – логічне додавання для всіх розрядів даного поля в зв'язку з тим,

що операція віднімання апаратно замінюється операцією додавання, а також може виникати операція перенесення в старший розряд. Пошук Специфічного маршруту $SR = IP\text{-одержувача AND } PN_i$, де i – номер рядка (запису) в таблиці маршрутизації ($i \in 1, K$) навпроти якого є запис 255.255.255.255. Тобто кількість таких рядків (i) залежить від кількості специфічних маршрутів. Пошук неспецифічного маршруту $NSR = IP\text{-піддержувача AND } M_i$, де i – номер рядка (запису) в таблиці маршрутизації ($i \in 1, K$), причому K - загальна кількість рядків в таблиці маршрутизації.

А також виконується додаткова операція, при призначенні перед заголовком IP-паketу MPLS-заголовок або стеків MPLS-заголовків. Необхідно відзначити, що дана операція відсутня на вихідному прикордонному LER-маршрутизатор. Надалі на кожному LSR відбувається MPLS-комутація, а саме з кадру відповідної каналної технології (PPP, Ethernet, Frame Relay, ATM) вилучається MPLS-заголовок, який знаходиться на вершині стека і проводиться його аналіз.

Перевірка ознакою стека міток Bottom of Stack = S AND одиниця. То є, якщо $S = 1$, то на останньому LSR в LSP-маршруті, безпосередньо підключений до вихідного LER, відбувається відкидання MPLS-заголовку [12].

Загальна сума елементарних операцій при обробці мітки одним LSR становить:

$$Q_{BL} = TTL_{LABAL} + LV_{ВИХ.} + \text{БОТТОМ OF STACK} \quad (3.5)$$

Використовуємо аналітичні моделі, яка була запропонована вище для проведення порівняльної характеристики. При IP-маршрутизації кожного маршрутизатора: проходить перевірка контрольної суми $KS = 16$ біт NOTAND 16 біт = 16 NOTAND операцій коригування часу життя пакета $TTL_{IP} = 8$ біт OR одиниця = 8 OR операцій пошук неспецифічного маршруту:

$$NSR = IP_{отримувача} \text{ AND } M_i = (32\text{біт AND } 32\text{біт}) * 20(\text{полів}) = 640 \text{ AND операцій} \quad (3.6)$$

Загальна сума елементарних операцій на одному маршрутизаторі в середньому становить:

$$Q_{BL} = KS + TTL_{IP} + NSP = 664 \text{ елементарних операцій} \quad (3.7)$$

Загальна сума елементарних операцій на даному маршруті становить:

$(KS + TTL_{IP} + NSR)$ помножене на кількість IP-маршрутизаторів в шляху = 2656 елементарних операцій. При використанні MPLS технології:

а) На прикордонних LER-маршрутизаторах були здійснені ті ж самі дії, тобто $(KS + TTL_{IP} + NSR)$ помножене на 664 елементарних операцій. Операцією призначення MPLS-заголовка зневаги.

б) На LSR-комутаторах: коригування часу життя мітки $TTL_{Label} = 8$ біт OR одиниця = 8 OR операцій; знаходження значення вихідного мітки в таблиці просування (20 біт AND 20 біт) помножене на 15 записів (полів) = 300 AND операцій; перевірка ознаки стека міток Bottom of Stack = 1 біт AND одиниця – 1 AND операція.

Загальна сума елементарних операцій на одному LSR становить:

$$Q_{BL} = TTL_{Label} + LV_{вих.} + Bottom\ of\ Stack = 309 \text{ елементарних операцій} \quad (3.8)$$

Загальна сума елементарних операцій на даному маршруті становить $(KS + TTL_{IP} + NSR)$ помножене на кількість LER в дорозі + $(TTL_{Label} + LV_{вих.} + Bottom\ of\ Stack)$ помножене на кількість LSR в шляху = $664 * 2 + 309 * 2 = 1328 + 618 = 1946$ елементарних операцій.

Також необхідно відзначити, що при використанні розроблених бітних моделей можна провести узагальнену порівняльну оцінку ефективності роботи активного обладнання по технології IP і MPLS, при допущенні приблизно однакового часу t виконання логічних функцій ($t_{AND} = t_{NOTAND} = t_{OR}$) мікропроцесором. Для цього застосуємо наступні розрахункові формули:

$$Q_{IP} = (24 + 32L_{IP}) * D \quad (3.9)$$

Де 24 – $(KS + TTL_{IP} = 8OR + 16NOTAND)$ елементарних логічних операцій; 32 – $(NSP = 32AND)$ логічних операцій; L_{IP} – кількість ліній зв'язку (ребер графа) IP-мережі в цілому; D – діаметр мережі (графа).

$$Q_{MPLS} = (24 + 32L_{IP}) * 2 + (9 + 20L_{MPLS}) * (D - 2) \quad (3.10)$$

Де $(24 + 32L) * 2$ – кількість елементарних логічних операцій з обробки пакетів на двох граничних LER; $(9 + 20L) * (D - 2)$ – кількість елементарних логічних операцій в MPLS-домені; L_{MPLS} – кількість ліній зв'язку (ребер графа) в MPLS-домені.

При виконанні умови $D > 2$, тобто існує хоча б один транзит. Згідно проведеного аналізу можна зробити висновок, що кількість елементарних операцій в разі застосування MPLS-технології зменшилася приблизно на 50%. У свою чергу використання аналітичної моделі, яка розроблена з урахуванням обов'язкових дій по обробці повідомлень в процесі передачі дозволяє кількісно оцінити величину ефекту від впровадження технології MPLS над IP-маршрутизацією [7]. Це пояснюється тим, що за рахунок зменшення операцій при обробці міток в разі перегляду таблиць просування (MPLS) в порівнянні обробкою пакетів і переглядом таблиць маршрутизації (IP-маршрутизація) збільшується швидкість просування пакетів, що видно на рисунку 3.17.

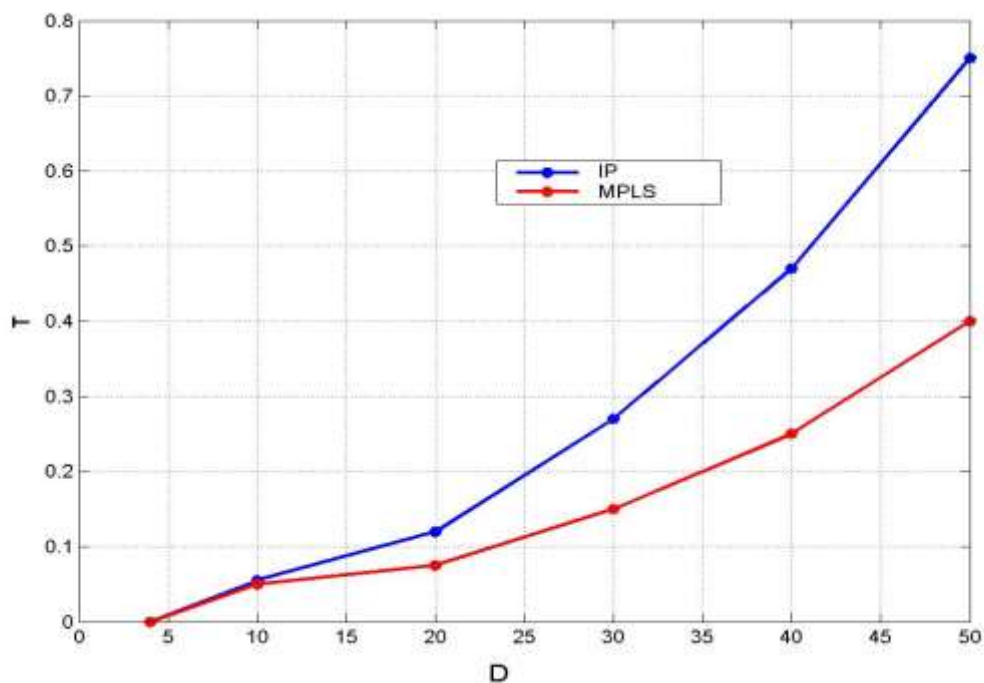


Рисунок 3.17 – Залежність часу на обробку фреймів (T) від діаметра мережі (D)

3.4 Висновки до третього розділу

З метою аналізу активного обладнання інформаційної мережі у середовищі GNS3 була побудована працездатна типова топологія, на основі технології MPLS.

Результатом роботи є розроблений алгоритм налаштування мережевого обладнання з використанням технології L2 VPN over MPLS, для досягнення необхідного рівня безпеки трафіка при проходженні через мережу Інтернет. Працездатність алгоритму представлена за допомогою програми Wireshark, яка візуалізує рух пакетів через мережу MPLS.

На створеній моделі було проведено порівняння IP та MPLS маршрутизаторів. Використання аналітичної моделі дозволяє кількісно оцінити величину ефекту від впровадження технології MPLS над IP-маршрутизацією. Це пояснюється тим, що за рахунок зменшення операцій при обробці міток в разі перегляду таблиць просування (MPLS) в порівнянні обробкою пакетів і переглядом таблиць маршрутизації (IP-маршрутизація) збільшується швидкість просування пакетів.

РОЗДІЛ 4

ЕКОНОМІЧНИЙ РОЗДІЛ

Під час розробки програмного забезпечення здійснюється: оцінка альтернативних проектних рішень; вибір методів і засобів реалізації проектних рішень; контроль відповідності досягнутого рівня якості програми вимогам до неї.

Оцінка альтернативних проектних рішень передбачає пошук серед багатьох варіантів реалізації поставленої задачі такого, який би забезпечив компроміс між витратами на виробництво й експлуатацію програми та рівнем її якості. Це пояснюється тим, що розробка програмного забезпечення проводиться, як правило, в умовах жорстких фінансових і часових обмежень, і питання забезпечення достатнього рівня якості, а саме модифікування та мобільності, залишаються поза увагою. У результаті цього з'являється програма, витрати на супроводження якої значно перевищують витрати на розробку. От чому виникає проблема пошуку компромісного рішення.

4.1 Вибір методів і засобів реалізації проектного рішення

Кожний проект може бути реалізований у будь-який спосіб, а саме: без застосування стандартів на програмування; з використанням принципів модульного програмування; з дотриманням правил структурного програмування; шляхом спадного проектування; з застосуванням мозаїкового (складального) програмування і т. ін.

Тобто, існує велика кількість методів та засобів (систем програмування, засобів редагування, засобів налагодження програм) розробки програм, які дозволяють створювати працездатні програми, що мають певний рівень якості. Серед цієї множини потрібно вибирати методи, які за мінімальної трудомісткості розробки гарантують достатній рівень якості.

4.2 Складові вартості моделювання

Вартість моделі складається з вартості її розробки, використання та супроводження.

Вартість розробки включає витрати на безпосередньо проектування програмного виробу, витрати на апаратні засоби, витрати на технологічні засоби розробки.

Вартість використання моделі включає витрати на експлуатацію програмного виробу, втрати внаслідок обмежених ресурсів ЕОМ, втрати внаслідок відказів у роботі програми.

Вартість супроводження моделі включає витрати на усунення помилок у програмі, витрати на її розвиток і модифікацію.

Основні чинники, що впливають на витрати та втрати, поділяються на такі групи: чинники, що впливають на витрати розробки; чинники, що впливають на витрати експлуатації; чинники, що впливають на витрати супроводження.

Чинники, що впливають на витрати розробки: призначення програми (загальна, прикладна або програма окремого користувача); обсяг і складність програми (загальна кількість операторів, кількість і складність функцій, обсяг вхідної та вихідної інформації, характер обробки: інформаційний пошук, арифметичні перетворення, складні математичні методи та ін.); режим використання (командний чи діалоговий, автономне використання або в межах обчислювальної мережі); кваліфікація розроблювачів (тематична – знання предметної сфери, технологічна – знання технологічних засобів і методів програмування, програмістська – знання мови програмування); технологічні методи та засоби, що використовуються під час розробки; мова програмування (застосування мов високого рівня при середній продуктивності праці зменшує трудомісткість розробки вдвічі); ступінь застосування раніш розробленого програмного забезпечення; організація управління розробкою; стабільність вимог користувача; розміри та структура баз даних; обмеження технічних засобів.

Чинники, що впливають на витрати експлуатації: вартість машинних ресурсів і носіїв; оплата праці персоналу; рівень інтерфейсу користувача з програмним

виробом (зручність управління); простота вводу даних і організація контролю за вводом; місткість пам'яті, швидкодія технічних засобів ЕОМ, на якій реалізується програма; організація обчислювального процесу; рівень налагодженості програми (обсяг тестового набору: основний, аварійний, граничних значень, вироджений і т. ін.); захист від перекручення початкових даних; наявність засобів захисту програми від відказів у процесі експлуатації.

Чинники, що впливають на витрати супроводження: якість документації; доступність для огляду структури програми; зрозумілість (читабельність) текстів програми; зміна вимог користувачів до методів обробки, структури даних, форм вихідних документів; повторне програмування з метою підвищення ефективності програми; зміна структури даних; зміна апаратних засобів і загального програмного забезпечення; вдосконалення документації.

4.3 Визначення трудомісткості розробки та опрацювання програмного продукту

Нормування праці в процесі створення моделі істотно ускладнено через творчий характер праці програмістів. Проте трудомісткість розробки і опрацювання моделі може бути розрахована на основі системи моделей з певною точністю оцінки.

Трудомісткість створення моделі визначається тривалістю кожної робочої операції, починаючи з складання технічного завдання і закінчуючи оформленням документації (за умови роботи одного програміста), що відображено у формулі (4.1) :

$$t = tmz + te + ta + tnp + tonp + t\partial, \text{ годин,} \quad (4.1)$$

де: tmz – тривалість складання технічного завдання на розробку моделі складає приблизно 16 годин;

te – тривалість вивчення ТЗ, літературних джерел за темою тощо;

ta – тривалість розробки блок-схеми алгоритму;

t_{np} – тривалість програмування (моделювання) за готовою блок-схемою;

t_{onp} – тривалість опрацювання програми на ПК;

t_{∂} – тривалість підготовки технічної документації на моделювання.

Складові трудомісткості визначаються на підставі кількості операторів у програмному продукті Q (з урахуванням можливих уточнень у процесі роботи над алгоритмом і програмою).

Кількість операторів у програмі розраховується по формулі (4.2):

$$Q = q \cdot c (1 + p), \text{ штук,} \quad (4.2)$$

$de : q$ – кількість операторів дорівнює;

c – коефіцієнт складності програми;

p – коефіцієнт корекції програми в процесі її опрацювання.

Коефіцієнт складності програми c визначає відносну складність програми щодо типового завдання, складність якого дорівнює одиниці. Діапазон його зміни – 1,25...2,0.

Коефіцієнт корекції програми p визначає збільшення обсягу робіт за рахунок внесення змін в алгоритм або програму внаслідок уточнення технічного завдання. Його величина знаходиться в межах 0,05...0,1, що відповідає внесенню 3...5 корекцій і переробці 5-10% готової програми.

Умовна кількість операторів у програмі складала:

$$Q = 300 \cdot 1.5 \cdot (1 + 0.1) = 495$$

Оцінка тривалості складання технічного завдання на розробку моделі t_{T3} залежить від конкретних умов і визначається дипломником на підставі експертних оцінок за узгодженням із керівником проекту. Тривалість вивчення технічного завдання, опрацювання довідкової літератури з урахуванням уточнення ТЗ і кваліфікації програміста можливо оцінити за формулою (4.3):

$$t_{\theta} = \frac{Q \cdot B}{(75 \dots 85) \cdot k}, \text{ годин,} \quad (4.3)$$

де: B – коефіцієнт збільшення тривалості етапу внаслідок недостатнього опису завдання, B = 1,2...1,5;

k – коефіцієнт, що враховує кваліфікацію програміста і визначається стажем роботи за фахом:

- до 2 років – 0,8;
- від 2 до 3 років – 1,0;
- від 3 до 5 років – 1,1...1,2;
- від 5 до 7 років – 1,3...1,4;
- понад 7 років – 1,5...1,6.

Тривалість вивчення технічного завдання склала:

$$t_{\theta} = \frac{495 \cdot 1.5}{80} = 9.28 \text{ годин}$$

Тривалість розробки блок-схеми алгоритму розраховується по формулі (4.4):

$$t_a = \frac{Q}{(20 \dots 25) \cdot k} = \frac{495}{23} = 21,52, \text{ годин.} \quad (4.4)$$

Тривалість складання моделі за готовою блок-схемою розраховується по формулі (4.5):

$$t_{np} = \frac{Q}{(20 \dots 25) \cdot k} = 21,52, \text{ годин.} \quad (4.5)$$

Тривалість опрацювання моделі на ПК розраховується по формулі (4.6):

$$t_{onp} = \frac{1,5Q}{(4..5) \cdot k} = 165, \text{ годин.} \quad (4.6)$$

Тривалість підготовки технічної документації на модель розраховується по формулі (3.7):

$$t_{д} = \frac{Q}{(15..20) \cdot k} + \frac{Q}{(15..20)} \cdot 0,75 = 49,5 \quad (4.7)$$

Трудомісткість створення моделі склала:

$$t = t_{mз} + t_{в} + t_{а} + t_{np} + t_{onp} + t_{д} = 16 + 9,28 + 21,52 + 21,52 + 165 + 49,5 = 283 \text{ годин.}$$

4.4 Розрахунок витрат на створення програмного продукту

Витрати на створення програмного продукту $K_{пз}$ складаються з витрат на заробітну плату виконавця програмного забезпечення $Z_{зп}$ та вартості машинного часу для налагодження програм:

$$K_{пз} = Z_{зп} + Z_{мч}, \text{ грн,}$$

Заробітна плата виконавця враховує основну і додаткову заробітну плату, а також відрахування на соціальне потреби (пенсійне страхування, страхування на випадок безробіття, соціальне страхування тощо) и визначається за формулою (4.8):

$$Z_{зп} = t \cdot Z_{пр}, \text{ грн,} \quad (4.8)$$

де: t – загальна тривалість створення моделі, годин;

Z_{np} – середньогодинна заробітна плата програміста з нарахуваннями, грн/годину.

Середня заробітна плата програміста складає 50 грн/год.

Заробітна плата виконавця склала:

$$Z_{zn} = t \cdot Z_{np} = 283 \cdot 50 = 14150 \text{ грн,}$$

Вартість машинного часу для налагодження програми склала:

$$Z_{мч} = t_{опр} \cdot C_{ч}, \text{ грн.}$$

де $C_{ч}$ – вартість машино-годин, 5 грн/год;

$$Z_{мч} = t_{опр} \cdot C_{ч} = 165 \cdot 5 = 825 \text{ грн}$$

Витрати на створення моделі склали:

$$K_{пз} = Z_{zn} + Z_{мч} = 14150 + 825 = 14975 \text{ грн.}$$

Таким чином було розраховано всі необхідні витрати для створення аналітичної моделі.

4.5 Висновки до четвертого розділу

Згідно розрахункам трудомісткість створення аналітичної моделі склала приблизно 283 години. Враховуючи середню заробітну плату програміста, витрати на створення моделі склали близько 14975 грн. В процесі створення моделі досить часто виникають ситуації, які не можна передбачити в розрахунках трудомісткості і

втрат на створення моделі, тому розважливо можна припустити, що обчислені показники трудомісткості і втрат на створення моделі необхідно буде збільшити на 10%.

ВИСНОВКИ

1. За допомогою моделювання MPLS вирішена актуальна задача: створення віддаленої мережі між двома офісами через мережу Інтернет. Для побудови аналітичної моделі використано симулятор GNS3, який дозволив дослідити властивості створеної моделі віддалених мереж. Разом з GNS3 був використаний модуль WireShark для аналізу трафіку на відповідність поставленій задачі. Побудована модель демонструє адекватну топологію та можливість швидкої реалізації даної моделі на практиці.

2. Для створення VPN через мережу Інтернет, використаний MPLS протокол, для інкапсуляції трафіку, який обраний після порівняння IP-маршрутизації та технології маршрутизації по міткам. Це дозволило досягнути оптимального співвідношення якості передачі і швидкості налаштування існуючої мережевої конфігурації. Порівняльна оцінка IP-маршрутизації та технології маршрутизації по міткам MPLS показала, що кількість елементарних операцій у випадку використання MPLS-технології зменшилась приблизно на 50%.

3. Для перевірки якості передачі мережевого трафіку між віддаленими мережами використані як емпіричні так і програмні інструменти. Результатом є зменшення затримки при обробці міток у випадку перегляду таблиці просування (MPLS) в порівнянні з обробкою пакетів та переглядом таблиці маршрутизації (IP-маршрутизація).

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1 An Introduction to IGRP. Document ID:26825. Charles L. Hedric, Rutgers, The State University of New Jersey, 1991
- 2 Callon, R. "Use of OSI IS-IS for routing in TCP/IP and dual environments" RFC-1195, Digital Equipment Corporation December 1990.
- 3 Hedrick, C "Routing Information Protocol" RFC-1058, Rutgers University June 1988.
- 4 Moy, J. "The OSPF Specification" RFC-1131, Proteon, Inc. October 1989.
- 5 Postel, J., "User Datagram Protocol", RFC 768, USC/Information Sciences Institute, November 1980.
- 6 Rekhter, Y., Li, T. "A Border Gateway Protocol 4 (BGP-4)" RFC-1771, T.J. Watson Research Center, IBM Corp., Cisco Systems, Inc., March 1995.
- 7 Васин, Н. Н. Сети передачи данных информационных систем железнодорожного транспорта на базе коммутаторов и маршрутизаторов CISCO: Учебное пособие для вузов ж.-д. транспорта / Н. Н. Васин, М.: Маршрут, 2005. - 232с.
- 8 Вито, Амато Основы организации сетей Cisco, том 2. Авторизованное учебное пособие. / А. Вито; пер. с англ. А. Н. Крикуна, М.: Издательский дом "Вильяме", 2002. - 464 с.
- 9 Будылдина Н.В., Телекоммуникационные сети с многопротокольной коммутацией по меткам. Построение и оптимизация/ Будылдина Н.В., Шувалов В.П. УрТИСИ ГОУ ВПО "СибГУТИ", 2006. — 274 с.
- 10 Гольдштейн А. Б., MPLS технология и протоколы/ Гольдштейн А.Б., Гольдштейн Б.С., — СПб.: БХВ 2005. — 304 с.
- 11 Кульгин, М. В. Компьютерные сети. Практика построения. / М. В.Кульгин, СПб.: Питер, 2003 - 462с.
- 12 Кэллон, Р. Межсетевой протокол / Р. Кэллон // ТИИЭР - 1983 -Том 71 № 12.-С. 73-80

13 Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер, СПб.: Питер, 2001. - 672с.

14 Олвейн В., Структура и реализация современной технологии MPLS.: Пер. с англ. / Олвейн В. — М.: Издательский дом "Вильямс", 2004. — 480 с.

15 Остерлох, Х. Маршрутизация в IP-сетях. Принципы, протоколы, настройка / Хизер Остерлох; пер. с англ. - СПб.: 000"ДиаСофтЮП", 2002. - 512с.

16 Столлингс, В. Современные компьютерные сети. 2-е изд. / В.Столлингс; пер. с англ. А. Леонтьев - СПб.: Питер, 2003. - 783с.

17 Таненбаум, Э. Компьютерные сети. / Э. Таненбаум; пер. с англ. В. Шрага, СПб.: Питер, 2003. - 992 с.

18 Уэйр, К. Сетевой уровень ВОС: Стандарты для реалий действительности / К. Уэйр // ТИИЭР - 1983 - Том 71, X» 12. - С. 68-72

19 Хелеби, Сэм Принципы маршрутизации в Internet / С. Хелеби, Д. Мак-Ферсон; Перевод с английского и редакция В.В. Ткаченко, М.: Издательский дом "Вильямс", 2001. - 448 с.

ДОДАТОК А

Перелік матеріалів дипломної роботи

- 1 Пояснювальна записка на листах.
- 2 Матеріали на оптичному носії:
 - 01 Титульна сторінка.doc
 - 02 Завдання.doc
 - 03 Реферат.doc
 - 04 Список умовних скорочень.doc
 - 05 Зміст.doc
 - 06 Вступ.doc
 - 07 Розділ 1.doc
 - 08 Розділ 2.doc
 - 09 Розділ 3.doc
 - 10 Розділ 4.doc
 - 11 Висновки.doc
 - 12 Список використаної літератури.doc
 - 13 Додаток А.doc
 - 14 Додаток Б.doc
 - 15 Додаток В.doc
 - 16 Додаток Д.doc
 - 17 Додаток Е.doc
- Презентація.ppt

ДОДАТОК В

В І Д Г У К

на дипломну роботу (проект)

студента Шапошников Максим гр. 172м-16-1

на тему: «Дослідження аналітичної моделі активного обладнання інформаційної телекомунікаційної мережі»

Актуальність теми: Зростання інформатизації суспільства посилює вимоги до надійності та продуктивності передачі даних в інформаційних телекомунікаційних мережах. Збільшення потоків даних призводить до виникнення пікових навантажень, при яких спостерігаються значні втрати пакетів та істотні часові затримки.

Повнота розкриття теми: Тема дипломного проекту розкрита повністю.

Теоретичний рівень: дипломного проекту достатньо високий.

Практична значущість: Технічний результат який отриманий в дипломній роботі полягає в практичне значення отриманих результатів полягає в підвищення ефективності роботи активного обладнання VPN.

Самостійність виконання роботи: робота виконана самостійно.

Якість оформлення, загальна та спеціальна грамотність: оформлення роботи відповідає вимогам, які до неї пред'являються.

Переваги та недоліки роботи: робота має практичне застосування, недоліками роботи є критичне значення розміру міток при масштабуванні мережі.

Загальна оцінка роботи та висновок, щодо рекомендацій до захисту в ДЕК: В цілому дипломний проект справляє позитивне враження та задовольняє вимогам, що пред'являються. Вважаю, що при відповідному захисті дипломний проект заслуговує оцінки «добре», а її автор, Шапошников М.І., присвоєння кваліфікації «Професіонала в галузі електроніки та телекомунікацій, дослідника» за напрям 6.050901 — телекомунікації.

Науковий керівник

доцент кафедри БІТ

_____ (доц. Гусев А.Ю.)

“08” червня 2017 р.

РЕЦЕНЗІЯ

на дипломну роботу (проект)

студента Шапошников Максим гр. 172м-16-1

на тему: «Дослідження аналітичної моделі активного обладнання інформаційної телекомунікаційної мережі»

Актуальність теми: Дипломний проект присвячено вирішенню актуальної проблеми: збільшенню швидкості пересилки фреймів за допомогою технології MPLS.

Наукова новизна: тема є новою та відповідає сучасним вимогам.

Якість проведеного аналізу: аналіз рівня розвитку техніки повністю відображено в дипломній роботі.

Уміння користуватися літературними джерелами: літературні джерела представлені в достатньому об'ємі.

Практична цінність висновків і рекомендацій: результати, що отримані в дипломній роботі мають практичну цінність і можуть бути застосовані на практиці.

Переваги та недоліки: в цілому дипломна робота справляє позитивне враження. Недоліком даної роботи є критичне значення розміру міток при масштабуванні мережі. Проте, це питання залишає простір для подальших досліджень даної теми.

Загальний висновок та оцінка роботи: в цілому дипломний проект справляє позитивне враження та задовольняє вимогам, що пред'являються. Вважаю, що при відповідному захисті дипломна робота заслуговує оцінки «добре», а її автор, Шапошников М.І., присвоєння кваліфікації «Професіонала в галузі електроніки та телекомунікацій, дослідника» за напрямом 6.050901 — телекомунікації.

Рецензент,

доцент кафедри ПЗКС, к.т.н.

_____ (доц. Гусев А.Ю.)

“ 12 ” червня 2017 р.