МАГІСТЕРСЬКА РОБОТА

МР.КІ-37.00.00.000 ПЗ

Група КІм-24-2

Сербенюк Андрій

2025

Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра комп'ютерних систем і мереж

Сербенюк Андрій Сергійович

(прізвище, ім'я, по батькові)

УДК 004.7

(індекс)

МАГІСТЕРСЬКА РОБОТА

Тема: *Модифікація алгоритму переналаштування маршрутної інформації для підвищення стійкості високонавантажених мереж*

(назва роботи)

Комп'ютерна інженерія

(назва освітньої програми)

123 – комп'ютерна інженерія

(шифр і назва спеціальності)

_____

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник *Заячук Ярослав Іванович, к.т.н., доцент*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту
Завідувач кафедри

проф. *С. І. Мельничук*

(посада) (підпис) (дата) (ініціали та прізвище)

Рецензент

_____

(посада) (підпис) (дата) (ініціали та прізвище)

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

Івано-Франківськ – 2025

Івано-Франківський національний технічний університет нафти і газу

Інститут *Інформаційних технологій*

Кафедра *Комп'ютерних систем та мереж*

Спеціальність *123 – Комп'ютерна інженерія*

Освітній рівень *магістр*

ЗАТВЕРДЖУЮ**:**

Зав. кафедрою КСМ

проф. С. І. Мельничук

“ ____ ” _____2025 р.

ЗАВДАННЯ

НА ВИКОНАННЯ МАГІСТЕРСЬКОЇ РОБОТИ СТУДЕНТОВІ

Сербенюку Андрію Сергійовичу

(прізвище, ім’я, по-батькові)

1. Тема магістерської роботи ***“ Модифікація алгоритму переналаштування маршрутної інформації для підвищення стійкості високонавантажених мереж”***

Керівник проекту *Заячук Ярослав Іванович*

затверджена наказом по університету від _____

2. Термін здачі студентом закінченої роботи *10.12.25*

3. Вихідні дані до проекту (роботи) *Матеріали та результати, отримані під час проходження переддипломної практики, методичні вказівки, технічна література*

4. Зміст розрахунково - пояснювальної записки (перелік питань, що їх належить розробити)

Аналіз протоколів маршрутизації та методів підвищення їх продуктивності

Модифікація алгоритму динамічного переналаштування маршрутної інформації

Реалізація програмних модулів

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

6. Консультанти по дипломній роботі, із зазначенням розділів роботи, що стосуються їх

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
нормоконтроль	***О. В. Мойсеєнко***		

7. Дата видачі завдання *12.03.2025*

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів бакалаврської роботи	Термін виконання етапів роботи	Примітка
1	*Аналіз протоколів маршрутизації та методів підвищення їх продуктивності*	*12.03.2025-31.05.2025*	*Виконано*
2	*Порівняння протоколів динамічної маршрутизації. Дослідження високонавантажених мереж*	*01.06.2025-31.07.2025*	*Виконано*
3	*Побудова алгоритмів переналаштування маршрутної інформації*	*01.08.2025-30.09.2025*	*Виконано*
4	*Реалізація програмних модулів*	*01.10.2025-15.11.2025*	*Виконано*
5	*Оформлення пояснювальної записки*	*16.11.2025-10.12.2025*	*Виконано*

Студент-магістр _____ ***Сербенюк А. С.***
(підпис)

Керівник роботи _____ ***Заячук Я. І.***
(підпис)

АНОТАЦІЯ

Кваліфікаційна робота присвячена аналізу та розробці програмно-апаратних підходів до підвищення ефективності маршрутизації у мережах із великим навантаженням.

У межах дослідження було проведено детальний огляд предметної області та визначено основні недоліки існуючих методів оптимізації мережевого трафіку. Для усунення виявлених недоліків запропоновано алгоритми, які спрямовані на підвищення продуктивності та стабільності роботи мереж із великим навантаженням.

На основі розроблених алгоритмів створено модулі, призначені для використання на маршрутизаторах Cisco.

RIP, EIGRP, OSPF, IS-IS, МАРШРУТИЗАЦІЯ, IP SLA, EEM, SDN, ВИСОКОНАВАНТАЖЕНА МЕРЕЖА, ДЖИТЕР

SUMMARY

The qualification work is dedicated to the analysis and development of software and hardware approaches aimed at improving routing efficiency in high-load networks.

Within the framework of the study, a detailed review of the subject area was conducted, and the main shortcomings of existing network traffic optimization methods were identified. To address these shortcomings, algorithms were proposed to enhance the performance and stability of high-load networks.

Based on the developed algorithms, modules were created for use on Cisco routers.

RIP, EIGRP, OSPF, IS-IS, ROUTING, IP SLA, EEM, SDN, HIGH- LOADED NETWORK, JITER

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CIDR – Classless Inter-Domain Routing

CLI – Command-line interface.

EEM – Embedded Event Manager.

EIGRP – Enhanced Interior Gateway Routing Protocol.

IP SLA – Internet Protocol Service Level Agreements.

IS-IS – Intermediate System to Intermediate System.

OSPF – Open Shortest Path First.

RIP – Routing Information Protocol.

SPF – Shortest Path First.

SDN – Software-Defined Network.

VLSM – Variable Length Subnet Masking.

VPN – Virtual Private Network.

АС – автономна система.

Jitter – фазові та/або частотні випадкові спотворення під час передачі сигналу.

МП – мережевий пристрій.

ПК – персональний комп’ютер.

ВСТУП

Комп’ютерні мережі становлять собою складну та багаторівневу систему, що забезпечує обмін даними між різними пристроями. Типова мережева архітектура включає численні компоненти, серед яких маршрутизатори, комутатори, сервери різних типів (зокрема веб-сервери), міжмережеві екрани, балансувальники навантаження, системи виявлення та запобігання вторгненням, а також інші апаратно-програмні засоби.

Для ефективної роботи мережі необхідні такі якості, як стабільність, продуктивність, гнучкість і надійність, адже саме вони визначають можливість коректного передавання великих обсягів інформації в реальному часі.

Такі вимоги спонукали виробників мережевого обладнання до впровадження все складніших і ресурсомістких протоколів, які забезпечують узгоджену взаємодію маршрутизаторів та комутаторів шляхом пакетної комутації й побудови оптимальної топології для маршрутизації даних.

Метою роботи є модифікація алгоритму та тестування модуля переналаштування маршрутної інформації при виникненні позаштатних ситуацій у високонавантажених мережах.

Актуальність теми дослідження. На сьогодні особливої уваги набувають питання, пов’язані з організацією високонавантажених мереж і вибором ефективного протоколу динамічної маршрутизації. Обсяги переданого трафіку безперервно зростають, що обумовлює підвищення вимог до пропускної здатності мережевих рішень. Зі зростанням масштабів підприємств і кількості користувачів відповідно підвищується й навантаження на мережеву інфраструктуру. Часто виникає потреба у модернізації обладнання або вдосконаленні його конфігурації.

Водночас протоколи динамічної маршрутизації самі по собі створюють додаткове навантаження через службовий трафік. У зв’язку з цим виникла потреба вдосконалення таких протоколів, а також покращення алгоритмів вибору оптимальнішого маршруту передавання даних.

Об’єктом дослідження є процес маршрутизації даних у високонавантажених комп’ютерних мережах.

Предметом дослідження є методи та способи налаштування маршрутної інформації в умовах підвищеного навантаження.

Методи дослідження. У роботі застосовані загальнонаукові емпіричні й теоретичні методи дослідження.

Наукова новизна: отримала подальший розвиток методика динамічного переналаштування маршрутної інформації під час виникнення непередбачених ситуацій**.**

Практичне значення: запропоновані рішення можуть бути впроваджені у мережевих інфраструктурах, що базуються на обладнанні Cisco. Це дозволить автоматично оптимізувати таблиці маршрутизації при збільшенні навантаження, забезпечуючи мінімальні втрати пакетів і підвищуючи загальну стійкість мережі до позаштатних ситуацій.

Апробації результатів роботи. Результати досліджень роботи оприлюднені на Всеукраїнській науково-практичній конференції молодих вчених та студентів «Інформаційні технології в освіті, техніці та промисловості» - 2025.

Структура і обсяг роботи. Магістерська робота складається зі вступу, 3 розділів, висновків і списку літератури, що включає 45 найменувань. Основна частина роботи викладена на 59 сторінках машинописного тексту. Робота містить 17 рисунки, 5 таблиць та 1 додаток на 3 сторінках.

1 АНАЛІЗ ПРОТОКОЛІВ МАРШРУТИЗАЦІЇ ТА МЕТОДІВ ПІДВИЩЕННЯ ЇХ ПРОДУКТИВНОСТІ

1.1 Протоколи маршрутизації

Протокол маршрутизації RIP найчастіше застосовується в невеликих локальних мережах, оскільки він відзначається простотою налаштування та обслуговування. Проте через відсутність ряду сучасних можливостей, які мають інші протоколи маршрутизації, такі як OSPF чи EIGRP [3], його використання зазвичай обмежене мережами невеликого масштабу [1].

Розрізняють дві основні версії даного протоколу – RIPv1 і RIPv2. Обидва варіанти використовують кількість переходів (hop count) як основну метрику маршруту та мають однакову адміністративну відстань, що становить 120 одиниць. З метою пристосування протоколу до сучасних потреб мережевої інфраструктури, у версії RIPv2 було вдосконалено багато аспектів першої редакції.

Зокрема, RIPv2 передає оновлення маршрутизації за допомогою мультикастової адреси 224.0.0.9 (Multicast address), а також підтримує роботу з мережевими масками, чого не було у RIPv1 [2]. Передача даних про маршрути в межах RIP здійснюється через пакети запитів (RIP Request) і відповідей (RIP Response).

Коли маршрутизатор тільки запускається, він може надіслати широкомовний запит RIP на всі інтерфейси, де цей протокол активовано. Інші маршрутизатори, що використовують RIP, отримують цей запит і повертають відповідні дані, надсилаючи пакет-відповідь із таблицею маршрутів. Такі дані допомагають створити локальну копію топології мережі, необхідну для коректного функціонування системи маршрутизації.

Окрім регулярної відправки оновлень кожні 30 секунд, маршрутизатор може генерувати ініційоване оновлення у разі виявлення нового сусіднього вузла

або збоїв у роботі інтерфейсу. У такій ситуації актуальна інформація про маршрути негайно поширюється на всі інтерфейси, які підтримують RIP, і ці зміни надалі враховуються в кожному оновленому пакеті RIP [2].

На рисунку 1.1 подано схематичне зображення таймерів протоколу RIP, які регулюють періодичність обміну даними між маршрутизаторами.

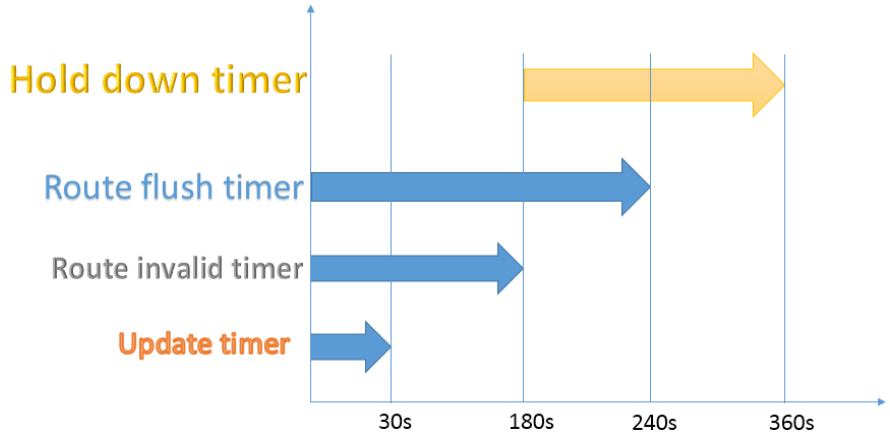

Рисунок 1.1 – RIP timers

Протокол EIGRP є сучасним і вдосконаленим рішенням для маршрутизації на основі вектору відстані [3]. Він забезпечує підтримку безкласової маршрутизації та VLSM, дозволяє додавати нові маршрути, виконувати інкрементні оновлення, балансувати навантаження, а також пропонує багато інших корисних функцій [1]. Оскільки це власний протокол компанії Cisco, всі маршрутизатори, де використовується EIGRP, мають бути обладнанням Cisco.

Перед тим як маршрутизатори почнуть обмінюватися інформацією про маршрути, вони повинні спочатку стати сусідами. Протокол обчислює метрику, враховуючи такі параметри, як пропускна здатність, затримка, надійність і навантаження. За замовчуванням при розрахунку метрики враховуються лише пропускна здатність та затримка, тоді як надійність і навантаження встановлюються як нульові значення.

EIGRP використовує концепцію автономних систем (АС). Всі маршрутизатори всередині однієї АС повинні мати однаковий номер, інакше вони не зможуть стати сусідами. Для встановлення зв’язку протокол застосовує п’ять різних типів пакетів, що продемонстровано на рисунку 1.2.

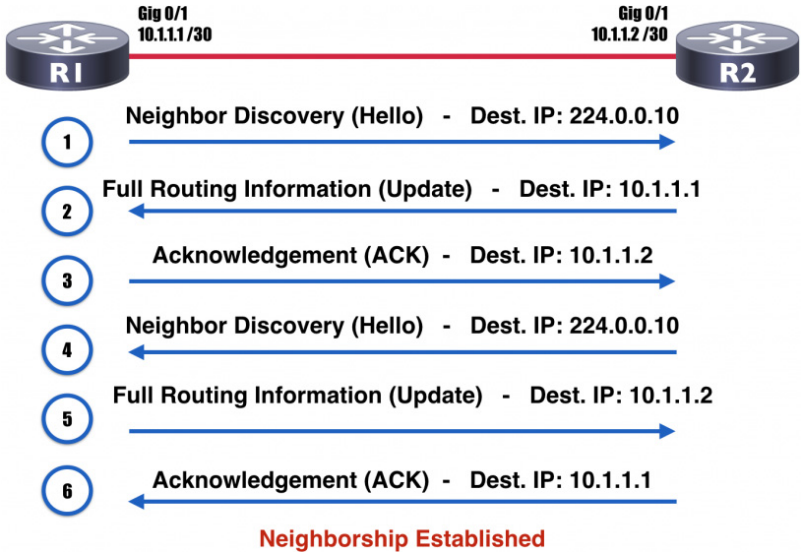

Рисунок 1.2 – EIGRP Neighbor Discovery

OSPF є відкритим стандартом і може реалізовуватися різними виробниками мережевого обладнання. Він сумісний із більшістю сучасних маршрутизаторів і гарантовано працюватиме на них. OSPF належить до безкласових протоколів маршрутизації, підтримує використання VLSM і CIDR, дозволяє вручну підсумовувати маршрути, а також забезпечує рівномірне балансування навантаження між лінками. Для оцінки маршрутів застосовується лише один параметр — вартість конкретного інтерфейсу. Для відновлення та обміну маршрутною інформацією використовуються спеціальні multicast адреси 224.0.0.5 і 224.0.0.6 [4, 5].

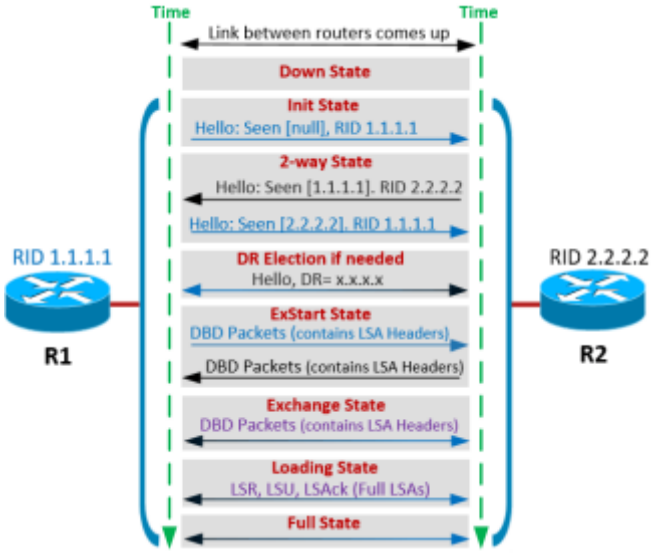

Рисунок 1.3 – OSPF Neighbor Discovery

Процес формування сусідства та встановлення маршрутів між двома маршрутизаторами представлено на рисунку 1.3.

IS-IS, подібно до OSPF, є протоколом внутрішнього шлюзу, що застосовує дані про стан каналів для прийняття рішень щодо маршрутизації. Протокол оцінює зміни у структурі мережі та визначає, чи потрібно виконувати повний SPF-перерахунок, чи обмежитися обчисленням лише часткових маршрутів (рис. 1.4). У великих мережах із кількістю маршрутизаторів понад 500 IS-IS демонструє високу швидкість збіжності [6].

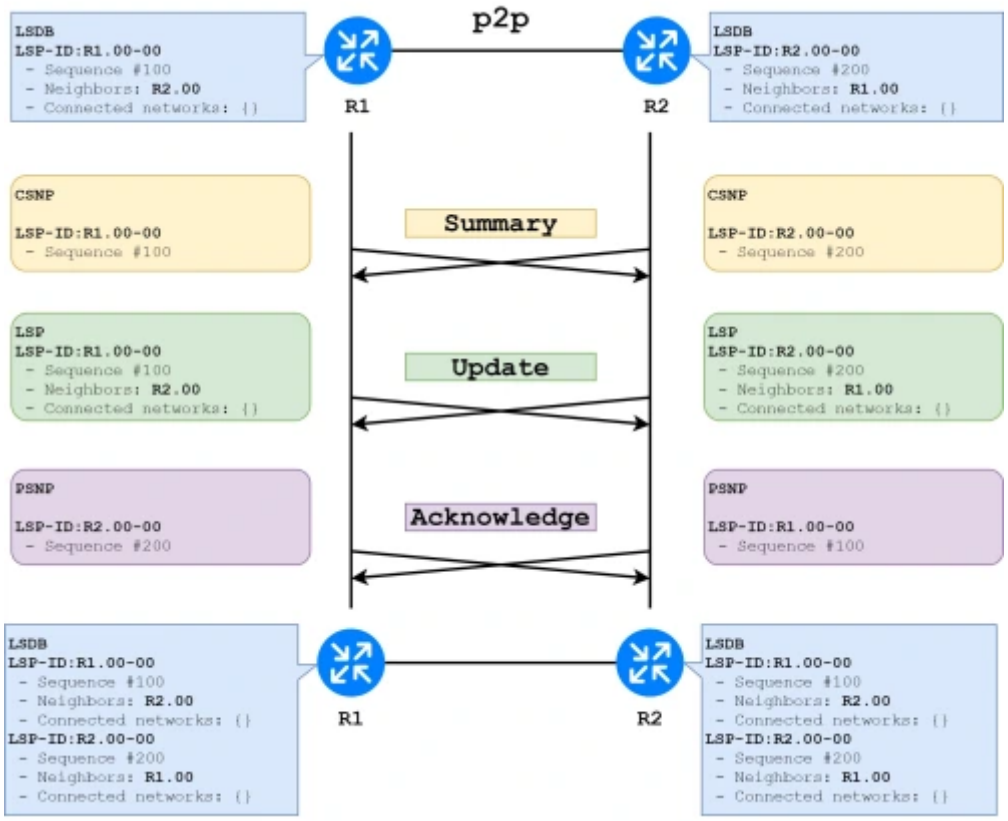

Рисунок 1.4 – IS-IS configuration

Мережа, яка працює за протоколом IS-IS, формує окрему автономну систему, або домен маршрутизації, що складається як із кінцевих, так і проміжних систем. Варто зазначити, що кінцеві системи відповідають за відправку та отримання пакетів, тоді як проміжні системи, крім цього, виконують функцію ретрансляції, пересилаючи пакети далі по мережі [7].

Протоколи типу DVA функціонують найбільш ефективно у випадках, коли:

- мережева структура є простою та не потребує складної ієрархії;

- фахівці-адміністратори не володіють достатнім досвідом для вибору конфігурацій і підтримання протоколів LSA;

- використовуються певні специфічні топології, наприклад, типу hub-and-spoke;

- коли тривалість конвергенції в найгіршій ситуації не має суттєвого значення для роботи системи.

У протоколах LSA для визначення найоптимальнішого маршруту застосовується алгоритм Дейкстри. Кожен маршрутизатор отримує дані від усіх інших пристроїв у мережі, завдяки чому формується повна топологічна карта комунікаційної мережі. Усі маршрутизатори мають ідентичне уявлення про структуру КМ. Ці протоколи не передбачають регулярних періодичних оновлень – після досягнення стабільного стану інформація передається лише у випадках змін топології.

LSA-протоколи найкраще проявляють себе в умовах, коли мережа має значний розмір і чітку ієрархічну будову; адміністратор володіє високим рівнем компетенції; а також коли для мережі критично важлива швидка конвергенція.

Порівняльний аналіз динамічних протоколів маршрутизації наведений у таблиці 1.1.

Таблиця 1.1 – Порівняння протоколів динамічної маршрутизації

Критерій	**DVA**				**LSA**	
	RIPv1	**RIPv2**	**IGRP**	**EIGRP**	**OSPF**	**IS-IS**
Швидкість конвергенції	Повільна	Повільна	Повільна	Висока	Висока	Висока
Масштабованість мережі	Мала	Мала	Мала	Велика	Велика	Велика
Підтримка VLSM	Ні	Так	Ні	Так	Так	Так
Ступінь використання ресурсів	Низька	Низька	Низька	Середня	Висока	Висока
Впровадження та підтримка	Проста	Проста	Проста	Складна	Складна	Складна

Варто зазначити, що пряме порівняння OSPF і RIP не є цілком коректним, оскільки вони призначені для кардинально різних типів мережевих середовищ. OSPF орієнтований на великі, складно організовані мережі, що створюються з

урахуванням детального проєктування. Натомість RIP – для невеликих систем, де простота реалізації дозволяє скоротити час на налаштування та зменшити складність конфігурації. Якщо мережа достатньо компактна для використання RIP, доцільно зупинитися саме на ньому, а в майбутньому за потреби перейти на EIGRP.

Серед переваг OSPF порівняно з RIP можна виділити: вищу масштабованість; підтримку VLSM і CIDR (що відсутня у RIPv1); менше споживання ресурсів у стабільних мережах; ефективніший вибір маршрутів; надійне уникнення маршрутних петель; більш точну та корисну метрику; підтримку створення ієрархічних топологій і значно швидшу стабілізацію мережевого стану.

Серед недоліків OSPF відносно RIP слід зазначити: несумісність ієрархічного підходу з неякісно спроєктованими IP-структурами; більшу складність налаштування; підвищене споживання обчислювальних ресурсів і пам’яті; а також необхідність значніших витрат часу на етапах проєктування й реалізації.

Протоколи OSPF та EIGRP мають чимало спільних характеристик. Обидва формують таблиці топологій і визначають маршрути на їх основі. У стандартних ситуаціях обидва протоколи запобігають виникненню петель маршрутизації. Проте залежно від умов роботи іноді доцільніше використовувати OSPF, а в інших випадках – EIGRP.

До переваг OSPF у порівнянні з EIGRP належать: сприяння впровадженню ієрархічних рішень у мережевому проєктуванні; простіша метрика у порівнянні з комбінованою метрикою EIGRP; відсутність проблем з «активним» станом маршруту; а також незалежність від конкретних виробників обладнання.

Водночас OSPF має і певні обмеження відносно EIGRP: менш гнучку систему метрик; відсутність можливості балансування навантаження між маршрутами з різною вартістю; несумісність ієрархічного підходу з неструктурованими IP-схемами; підвищені вимоги до апаратних ресурсів; і більші часові витрати на розробку, тестування та впровадження.

1.2 Internet Protocol Service Level Agreements

Технологія від Cisco активно відстежує мережевий трафік, генеруючи його постійно, передбачувано та надійно, для того, щоб оцінити продуктивність мережі [1]. IP SLA передає дані між кількома точками мережі або різними маршрутами для забезпечення детальної перевірки. Ця технологія моделює мережеві процеси та IP-сервіси, збираючи інформацію про реальні показники продуктивності. Вона фіксує час відгуку, односторонню затримку, джитер, втрату пакетів, якість голосового зв'язку, доступність мережевих ресурсів, а також ефективність роботи додатків. IP SLA створює та аналізує трафік для оцінки продуктивності між пристроями Cisco або між пристроєм Cisco і віддаленим IP-пристроєм, наприклад сервером мережевих додатків. Статистичні дані, що формуються різними операціями IP SLA, використовуються для усунення проблем, аналізу неполадок і планування оптимальної мережевої топології [8].

Завдяки IP SLA, клієнти постачальника послуг можуть перевіряти рівень обслуговування та надавати дані для SLA. Корпоративні користувачі мають можливість оцінювати ці показники та аналізувати продуктивність мережі для нових або існуючих IP-додатків і послуг. Пакети конфігуруються на IP- та прикладному рівні: можна визначати IP-адреси відправника й одержувача, номери портів, обрані користувачем, тип обслуговування, параметри маршрутизації чи пересилання через VPN, а також URL. Незалежно від другого рівня транспортного протоколу, IP SLA можна налаштовувати наскрізно через різнорідні мережі, щоб максимально точно відображати показники, які відчуває кінцевий користувач.

Використовуючи IP SLA, мережевий інженер має змогу контролювати продуктивність між будь-якими сегментами мережі: ядром, розподілом та периферією. Моніторинг можливий у будь-який час і з будь-якої точки без потреби розгортати фізичні зонди. IP SLA застосовує згенерований трафік для оцінки продуктивності між двома мережевими пристроями.

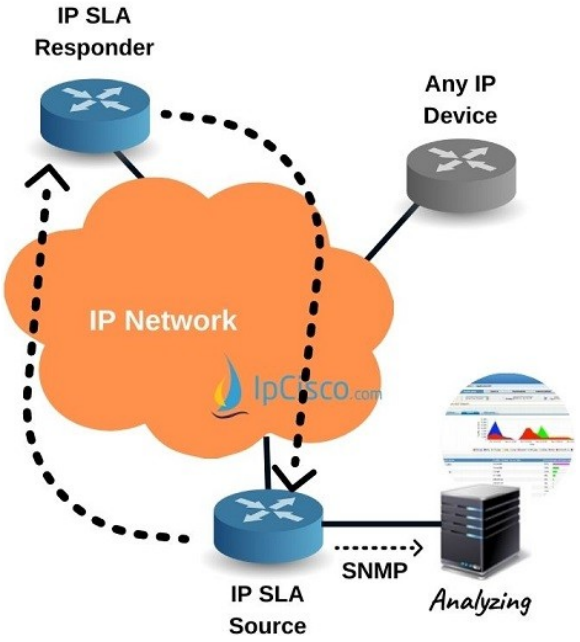

Рисунок 1.5 – Архітектура IP SLA

На рисунку 1.5 показано запуск IP SLA: пристрій відправляє згенерований пакет до цільового вузла. Після отримання пакета цільовий пристрій, залежно від обраної операції IP SLA, відправляє відповідь із міткою часу назад джерелу, що дозволяє розрахувати всі необхідні показники продуктивності.

1.3 Embedded Event Manager

Це особлива підсистема програмного забезпечення Cisco IOS. Насправді EEM є потужним і дуже гнучким інструментом, призначеним для автоматизації різних завдань та налаштування поведінки ПЗ Cisco і роботи пристроїв. Користувачі можуть застосовувати EEM для розробки та запуску програм або сценаріїв безпосередньо на маршрутизаторі чи комутаторі. Такі сценарії називають політиками EEM, і вони можуть бути запрограмовані через простий CLI-інтерфейс або з використанням мови сценаріїв Tcl [1]. EEM надає можливість користувачам застосовувати інтелект ПЗ Cisco IOS для миттєвої реакції на події, автоматизації операцій, створення настроюваних команд і виконання локальних автоматичних дій на основі умов, які визначає саме ПЗ Cisco IOS [9].

EEM, по суті, є функцією програмного забезпечення, незалежною від конкретного продукту. Вона складається з набору детекторів подій, сервера EEM і

спеціальних інтерфейсів, що дозволяють запускати підпрограми дій, відомі як політики. Крім того, існують внутрішні API для інших підсистем Cisco IOS, які дозволяють інтегрувати і використовувати можливості EEM. На рисунку 1.6 показано основні компоненти цієї підсистеми.

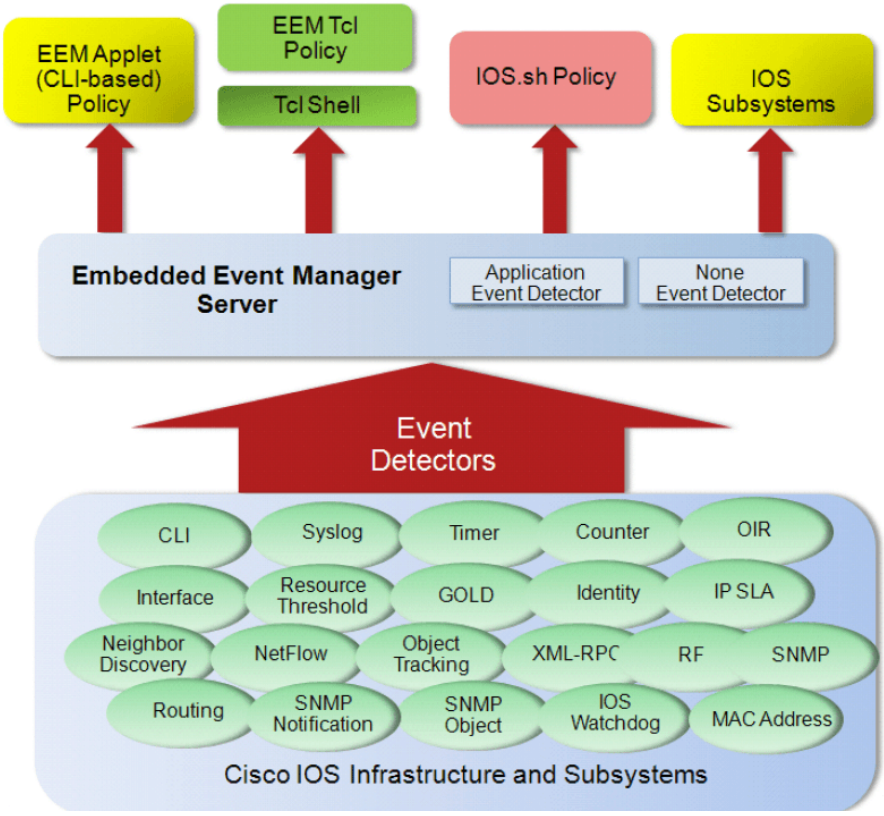

Рисунок 1.6 – Архітектура EMM

Існують два основні типи політик EEM:

- політика аплету: пропонує простий інтерфейс для користувача та визначається через CLI;

- політики Tcl: надають більш широкі можливості та гнучкість і програмуються з допомогою Tcl.

Якщо одна або кілька політик створені, детектор подій програмного забезпечення починає відслідковувати умови, що відповідають критеріям політики. Коли така умова спрацьовує, подія передається серверу диспетчера подій. Сервер, у свою чергу, активує конкретну політику, зареєстровану для цієї ситуації. Потім виконуються дії, визначені всередині політики. Кожен тип події має свої параметри та детальну інформацію, яка доступна політиці під час її активації.

1.4 Software-Defined Network

SDN представляє собою новітнє покоління мережевих архітектур, де управління здійснюється повністю незалежно від процесу пересилання пакетів і реалізується програмно. Перехід до такого підходу, раніше тісно інтегрованого з окремими МП, дозволяє звільнити базові функції для роботи додатків і сервісів, які можуть сприймати мережу як логічну або віртуальну структуру. На рисунку 1.7 наведено логіку архітектури SDN.

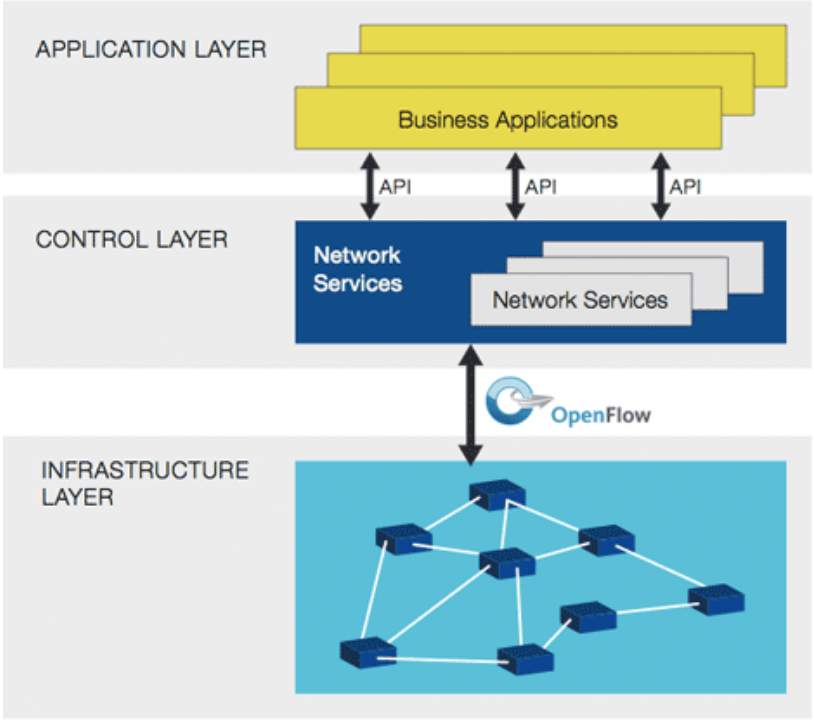

Рисунок 1.7 – Архітектура Software-Defined Network

Варто підкреслити, що “інтелект” мережі сконцентрований у програмних контролерах SDN, які володіють глобальним поглядом на всю мережу. Це дозволяє додаткам і політичним механізмам бачити мережу як єдиний логічний пристрій для пересилання даних. Завдяки SDN організації та оператори отримують централізовану можливість контролю через єдину логічну точку над всією інфраструктурою. Така модель значно спрощує проєктування і експлуатацію мережі. Крім того, SDN полегшує роботу МП, оскільки їм не потрібно самостійно розбиратися у великій кількості протоколів – достатньо отримувати чіткі вказівки від контролерів SDN [10].

Найголовніше, що оператори та адміністратори мереж можуть програмно управляти спрощеним поданням інфраструктури, замість ручного налаштування численних конфігурацій, розкиданих по сотнях чи тисячах МП.

Завдяки централізованому інтелекту контролера SDN, IT-відділ здатен змінювати поведінку мережі у реальному часі та швидко розгортати нові додатки і сервіси протягом кількох днів або навіть годин, а не місяців, як у традиційних підходах. SDN дає змогу менеджерам гнучко налаштовувати мережу, керувати нею і захищати ресурси за допомогою автоматизованих програм, розроблених для SDN. Більше того, організації можуть самостійно створювати такі програми, не очікуючи, поки функції будуть реалізовані у закритих пропрієтарних системах.

Архітектура SDN також дозволяє застосовувати широкий набір API-інтерфейсів, які забезпечують спільні мережеві сервіси: маршрутизацію, багатомасштабну передачу, безпеку, контроль доступу, управління пропускною здатністю та трафіком, енергоспоживання і різноманітні політики, оптимізацію процесорів і сховищ, орієнтовані на досягнення бізнес-цілей. SDN надає простий спосіб визначати узгоджені політики для різних типів з'єднань у кампусі. Аналогічно, мережу можна керувати через інтелектуальні системи оркестрації і забезпечення. Open Networking Foundation досліджує відкриті API для підтримки багатопостачальницьких мереж, що забезпечує гнучкий розподіл ресурсів, самообслуговування та безпечні віртуальні хмари.

Завдяки open API додатки можуть працювати на абстрактному рівні, використовуючи сервіси мережі без прив’язки до деталей реалізації. Мережа за допомогою SDN налаштовується під потреби програм, а програми використовують мережеві можливості, не знаючи її внутрішньої структури. Це дозволяє оптимізувати ресурси, включно з обчислювальними потужностями і сховищами.

OpenFlow – перший стандарт для взаємодії між контролером і площиною пересилання в SDN. Він забезпечує прямий доступ і управління як фізичними, так і віртуальними МП, зокрема через гіпервізор. Відсутність такого інтерфейсу раніше змушувала МП залишатися монолітними та закритими, подібними до

мейнфреймів. Ніякий інший протокол не замінює OpenFlow, який переносить контроль із комутаторів у централізоване програмне забезпечення. Його можна порівняти з набором команд процесора. На рисунку 1.8 показано, як OpenFlow надає зовнішньому додатку можливість програмувати пересилання пакетів МП, так само як набір команд управляє комп'ютером.

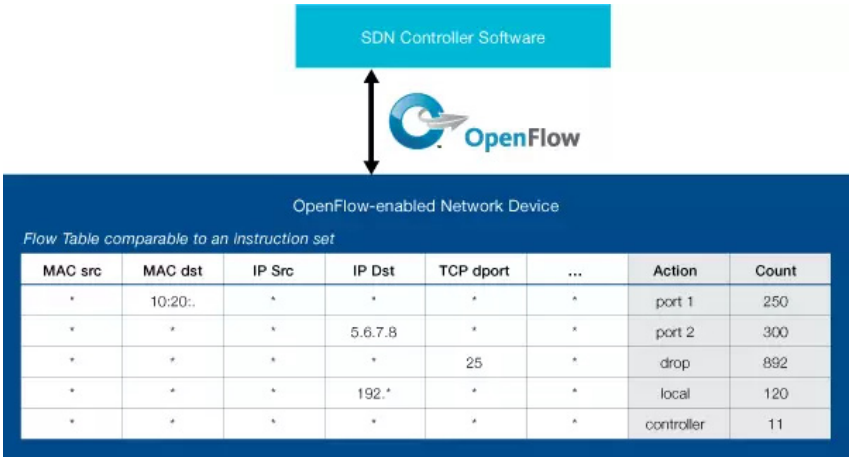

Рисунок 1.8 – Приклад OpenFlow Instruction Set

OpenFlow працює з потоками для визначення мережевого трафіку на основі параметрів відповідності, які можна задавати статично або динамічно через контролер SDN. Це дозволяє IT-відділу визначати маршрути потоків через МП з урахуванням моделей використання, додатків і хмарних ресурсів.

Оскільки OpenFlow підтримує програмування потоків окремо, SDN отримує дуже детальний контроль і здатність адаптувати мережу в реальному часі під додатки, користувачів та сеанси. Традиційна IP-маршрутизація не забезпечує такого рівня, бо всі потоки між двома кінцевими точками слідують одним шляхом, незалежно від різних потреб.

OpenFlow є ключовим протоколом підтримки SDN і на сьогодні є єдиним стандартним способом прямого управління площиною пересилання. Спочатку створений для Ethernet, OpenFlow можна застосовувати у ширшому спектрі мережевих сценаріїв. SDN на основі OpenFlow легко інтегрується у фізичні або віртуальні мережі, підтримуючи одночасно традиційну пересилку. Це полегшує впровадження SDN у мережах різних постачальників.

Open Networking Foundation стандартизує OpenFlow через робочі групи, що займаються протоколом, конфігурацією та тестуванням сумісності, забезпечуючи

взаємодію між різними МП та контролерами. Постачальники часто впроваджують OpenFlow оновленням ПЗ або прошивки. SDN на базі OpenFlow інтегрується у наявну інфраструктуру і створює простий шлях міграції для сегментів мережі, які найбільше потребують SDN-функцій.

Для підприємств та операторів SDN робить мережу конкурентоспроможною, а не просто джерелом витрат. Технології SDN дозволяють вирішувати питання високої пропускної здатності, адаптувати мережу під змінні потреби бізнесу та знижувати складність операцій і управління.

Переваги SDN на основі OpenFlow включають [1]:

- централізоване управління пристроями різних постачальників: контролер SDN може керувати будь-яким OpenFlow-підтримуваним МП, включно з комутаторами, маршрутизаторами та віртуальними комутаторами. Замість роботи з групами пристроїв окремих виробників, IT-фахівці можуть використовувати оркестрацію для швидкого розгортання та оновлення всіх МП;

- зниження складності завдяки автоматизації: SDN забезпечує гнучку інфраструктуру для автоматичного керування мережею та програмування завдань, які раніше виконували вручну;

- прискорення інновацій: застосування SDN дозволяє операторам і програмістам динамічно програмувати мережу у реальному часі відповідно до потреб бізнесу та користувачів;

- підвищення надійності і безпеки: IT-фахівці задають високорівневі політики, що транслюються в інфраструктуру через OpenFlow, усуваючи потребу ручного налаштування МП при зміні кінцевих точок, сервісів чи програм;

- детальне управління мережею: модель потоків OpenFlow дозволяє застосовувати політики на рівнях сеансу, користувача, пристрою та додатків у вигляді автоматизованих абстрактних правил;

– покращення взаємодії з користувачем: завдяки централізованому контролю та наданню інформації про стан мережі додаткам вищого рівня, інфраструктура SDN краще відповідає динамічним потребам користувачів.

1.5 Методи підвищення продуктивності протоколів маршрутизації

Зі збільшенням масштабів мереж та ростом трафіку, який по них проходить, виникає потреба у нових методах ефективної доставки пакетів даних між вузлами [1]. Наприклад, у роботі [11] автор ставить завдання підвищення продуктивності мережі під час пікових навантажень. Для цього пропонується модифікувати протокол маршрутизації EIGRP і провести моделювання результатів у фреймворку ANSAINET для середовища OMNeT++. В дослідженні аналізуються найбільш поширені протоколи динамічної маршрутизації, зокрема OSPF та EIGRP, а також проводиться їх моделювання в умовах пікових навантажень. Крім того, автор пропонує метод підвищення продуктивності, який базується на вимірюванні поточного навантаження на інтерфейсах маршрутизаторів і перерахунку маршрутів при зниженні ефективності роботи мережі. За результатами проведеного дослідження метод було реалізовано у бібліотеці ANSAINET для OMNeT++, підтвердивши його працездатність у процесі моделювання.

У роботі [12] автори досліджують шляхи підвищення продуктивності протоколу внутрішньодоменної маршрутизації EIGRP. Вони пропонують інтегрувати функціонал SDN, реалізуючи модифікований підхід для поліпшення ряду ключових показників: завантаження каналів, втрати пакетів та пропускної здатності. Суть підходу полягає у застосуванні інтелектуального динамічного контролера, який виявляє можливі перевантаження до їх виникнення або на самому початку проблеми. Після виявлення контролер застосовує три підалгоритми: розподіл потоку можливих наступників, визначення тимчасового наступника та очищення маршруту. Вони виконуються послідовно для конкретного потоку трафіку лише за умов високої інтенсивності навантаження. Якщо перший або другий процес вирішує проблему, наступні етапи не застосовуються. У випадку, коли перші два алгоритми не справляються, третій повторює їх для інших потоків, що зменшує генерацію керуючих повідомлень у мережі.

У роботі [13] досліджується балансування зростаючого навантаження, пов’язаного зі швидким збільшенням інтернет-трафіку та появою додатків реального часу в IP-мережах. Автор використав комбінаторний алгоритм, запропонований К.Г. Рамакрішнаном та М.А. Родрігесом, для наближеного вирішення задачі оптимальної маршрутизації по найкоротшому шляху, і запропонував його вдосконалення. Першим покращенням стало розширення багатопроменевого поширення з рівними витратами; другим — пошук локальних сусідів для балансування потоку даних; третім — можливість одночасного застосування декількох змін ваги, що дозволяє ефективніше розподіляти навантаження. Таким чином, автор забезпечив кращі результати порівняно з оригінальним методом.

Використовуючи рішення оптимальної спільної задачі маршрутизації на основі мультимодальної потокової регуляції для оцінки продуктивності алгоритму, автори показали, що останнє розширення евристичного методу дає результати в межах декількох відсотків від оптимального. Виявлене налаштування ваги є ефективним для нових «гарячих точок» та при збоях у каналах зв’язку, тому значних змін параметрів не очікується. Надійне початкове налаштування ваги також слугує хорошою відправною точкою для подальшого застосування методики Рамакрішнана, що наближається до оптимального рішення, особливо у випадках відмови каналів.

У статті [14] автори досліджують можливості адаптивного балансування навантаження в OSPF-мережах за допомогою розподіленого підходу. Вони розглядають оптимізацію ваг OSPF із застосуванням примітивно-подвійних методів і поєднання цього з адаптивною евристикою. Результати показують, що оптимізація коефіцієнтів поділу трафіку покращує продуктивність мережі порівняно з рівним розподілом навантаження. У випадках, коли кількість найкоротших шляхів невелика, зміна ваг OSPF також є доцільною для покращення ефективності маршрутизації.

1.6 Постановка завдання

Здійснено розгляд основних протоколів внутрішньодоменної маршрутизації, серед яких RIP, EIGRP, OSPF та IS-IS. Для кожного з них докладно описано принципи функціонування та особливості реалізації. Окрему увагу приділено інструментам IP SLA, EEM і SDN, які були проаналізовані з точки зору можливостей їх застосування у поставленому завданні.

За результатами розгляду зазначених технологій було прийнято рішення використовувати комбінацію IP SLA та EEM, оскільки перший інструмент забезпечує виявлення різних подій у мережевому середовищі, таких як затримки або втрата пакетів, тоді як другий здатен оперативно реагувати на ці події. Спільне використання IP SLA і EEM дає змогу повною мірою реалізувати мету дослідження – підвищити ефективність маршрутизації у високонавантажених мережах.

Водночас технологія SDN також потенційно може забезпечити досягнення аналогічних результатів, проте її впровадження потребує складного процесу програмування контролерів. Зазвичай це завдання виконується командою розробників, які мають значний практичний досвід і глибокі технічні знання, що робить цей підхід менш доцільним у межах даної роботи.

Таким чином метою роботи є модифікація алгоритму та тестування модуля переналаштування маршрутної інформації при виникненні позаштатних ситуацій у високонавантажених мережах.

Для досягнення мети поставлено наступні завдання:

- провести дослідження функціонування високонавантаженої мережі;
- спроєктувати механізм динамічного переналаштування маршрутної інформації;
- виконати тестування запропонованого рішення.

2 МОДИФІКАЦІЯ АЛГОРИТМУ ДИНАМІЧНОГО ПЕРЕНАЛАШТУВАННЯ МАРШРУТНОЇ ІНФОРМАЦІЇ

2.1 Порівняння протоколів динамічної маршрутизації

У роботі [15] наведено порівняльний аналіз протоколів маршрутизації: RIPv2, EIGRP, OSPF та IS-IS. Автор виділяє три основні критерії для порівняння протоколів: час конвергенції, обсяг службового трафіку та адміністративну відстань. Крім того, обґрунтовується значущість кожного критерію та пояснюється, як вони впливають на роботу мережі й створюване навантаження.

Також описується лабораторний стенд, який складається з шести маршрутизаторів Cisco та трьох комп'ютерів, а також початкове налаштування мережевого обладнання. Після цього покроково здійснюється конфігурування кожного протоколу динамічної маршрутизації, з фіксацією показників за обраними критеріями. Підсумкові дані роботи зібрані в таблиці 2.1.

Таблиця 2.1 – Результати вимірювання критеріїв

Критерій	RIPv2	EIGRP	OSPF	IS-IS
Час конвергенції, с	8,590100	0,609551	9,867057	9,467620
Обсяг службового трафіку, байт	356	596	706	2517
Адміністративна відстань	120	90	110	115

За результатами аналізу автор робить висновок, що протокол EIGRP є найефективнішим для застосування у мережах з високим навантаженням. Додатково доцільно провести порівняння алгоритмів розрахунку метрики кожного протоколу, оскільки вони не були розглянуті в статті.

Метрика протоколу RIP визначається кількістю проміжних маршрутизаторів між початковим хостом та кінцевим вузлом. Після відправлення мінімальної таблиці маршрутизації сусіднім вузлам, маршрутизатор збільшує кожне отримане значення метрики на одиницю та заносить у власну таблицю лише ті маршрути, що мають найменшу метрику. Після цього відбувається

повторне обмінювання таблицями маршрутизації та їхнє оновлення, поки не встановиться правильний режим маршрутизації [1].

Проте цей алгоритм має серйозний недолік. У мережах, наприклад, до 15 вузлів, повідомлення з повними таблицями маршрутизації значно навантажують канали зв'язку. Якщо кількість вузлів перевищує 15, протокол RIP стає непридатним для роботи, тому його застосування в великих та високонавантажених мережах є недоцільним.

Протокол EIGRP, розроблений компанією Cisco, використовує для обчислення метрики формулу:

$$metric = \left[k_1 \cdot bw + \frac{k_2 \cdot bw}{256 - load} + k_3 \cdot delay + k_6 \cdot extattributes \right] \cdot \frac{k_5}{k_4 + reliability} , \qquad (2.1)$$

де $k_1,...k_6$ – коефіцієнти для врахування різних компонентів формули;

bw – найменша пропускна здатність на маршруті;

$load$ – максимальне завантаження каналу;

$delay$ – сумарна затримка на інтерфейсах;

$extattributes$ – показник джиттера та витрати енергії на маршруті;

$reliability$ – найнижча надійність на всьому маршруті.

Змінюючи значення коефіцієнтів $k_1,...k_6$ можна гнучко регулювати метрику, що надає EIGRP перевагу перед іншими протоколами динамічної маршрутизації, оскільки дозволяє враховувати більше факторів, що впливають на маршрут [1]. За замовчуванням формула для розрахунку метрики маршруту:

$$metric = bw + delay , \qquad (2.2)$$

де bw – найменша пропускна здатність на маршруті;

$delay$ – сумарна затримка на інтерфейсах.

Протокол OSPF застосовує метрику *cost*, обчислювану за формулою:

$$metric(cost) = \frac{10^8}{bw} , \qquad (2.3)$$

де bw – найменша пропускна здатність на маршруті.

У ній 10^8 – еталонна пропускна здатність, що дорівнює 100 Мбіт/с. Наприклад, з'єднання Fast Ethernet має метрику 1, а Ethernet – 10. Для маршруту, що проходить через декілька маршрутизаторів, метрики сумуються. Таким чином, OSPF враховує лише пропускну здатність каналів, подібно до EIGRP, але не включає інші фактори, які можуть впливати на роботу мережі [1].

Протокол IS-IS підтримує чотири типи метрик:

- Default metric – стандартне значення 10 для кожного інтерфейсу;
- Delay – аналогічно затримці в EIGRP;
- Expense – фактична вартість використання каналу;
- Error – аналогічно показнику reliability у EIGRP [1].

За замовчуванням маршрутизатори IS-IS використовують тільки default metric, що робить значення метрики рівним кількості проміжних вузлів. Такий підхід подібний до RIP і не підходить для використання у мережах із великим навантаженням.

2.2 Дослідження високонавантажених мереж

Високонавантажена мережа – це система передачі даних, через яку проходить величезний обсяг трафіку. Гігабіти інформації пересуваються від одного вузла до іншого, що інколи призводить до затримок у передачі або навіть втрати даних. Подібні явища часто виникають у магістральних мережах. Наприклад, під час відеоконференції між віддаленим офісом у іншому місті та головним офісом, при перевантаженні магістральної мережі провайдера відеопотік може надходити із затримкою. Через це зображення на екранах учасників конференції відображається із зависаннями. Голосові сигнали також перериваються і стають нерозбірливими.

Ще один приклад – епідеміологічна ситуація, коли більшість компаній переводять співробітників на дистанційну роботу. Люди, які перебувають на карантині, активніше користуються відеосервісами онлайн, що значно підвищує навантаження на мережу. У таких умовах віддалена робота стає більш складною,

а платформи для відеотрансляцій змушені знижувати якість сигналу, оскільки мережа не витримує підвищеного трафіку.

Як зазначалося раніше, у високонавантажених мережах трафік може заповнювати канали зв’язку, що спричиняє затримки і втрати пакетів. Для оцінки якості таких мереж застосовують метрики:

- втрати пакетів;
- затримки;
- джитер.

Метрика втрати пакетів відображає частку даних, що дійшла до одержувача. На рисунку 2.1 продемонстровано випадок, коли відправник передає три пакети, але через втрату на маршрутизаторі одержувачу надходять не всі дані.

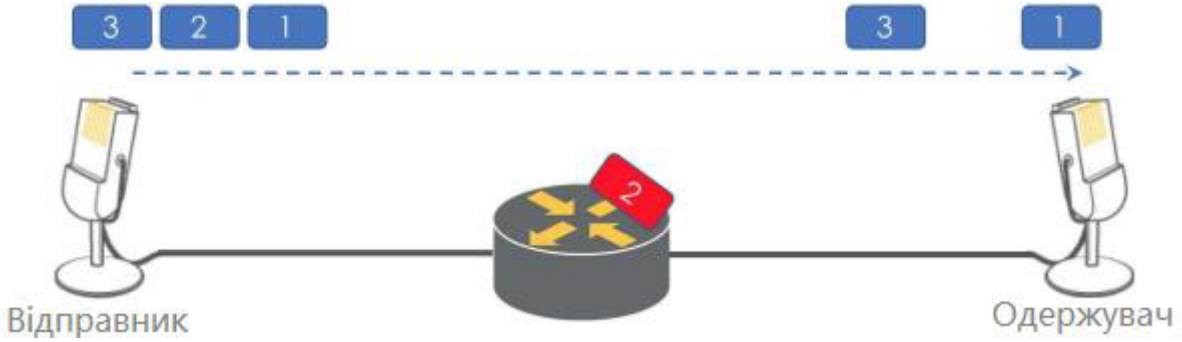

Рисунок 2.1 – Втрата пакетів

У мережах передачі даних пакети постійно можуть губитися, але зазвичай це не критично. Втрати стають проблемними при передачі даних у режимі реального часу, наприклад, під час телефонних розмов.

Причини втрат пакетів бувають різні. Однією з основних є переповнення каналів у мережах. Процес передачі пакетів відбувається послідовно, і збої у каналах можуть спричиняти втрати для компенсації перевантаження.

Помилки програмного забезпечення теж можуть бути джерелом проблем. Неправильно написані або неперевірені додатки, що працюють із мережею, здатні створювати збої та впливати на доставку пакетів.

Втрати іноді виникають через старе або несправне обладнання – маршрутизатори, комутатори, брандмауери. Таке устаткування уповільнює передачу даних, а підвищений трафік ще збільшує затримки і втрачені пакети.

Ще одна можлива причина – кібернетичні атаки. Зловмисники можуть спеціально викликати падіння пакетів, що неминуче призводить до втрат.

Усі перелічені фактори так чи інакше здатні спричинити втрату пакетів. Наслідки залежать від типу переданої інформації: зображення можуть стати нечіткими, аудіо – з шумами або нерозбірливим голосом.

Наступна метрика – затримки. Сукупна затримка визначає час, необхідний для доставки пакетів від відправника до одержувача. На рисунку 2.2 показано приклад затримки, а на рисунку 2.3 – джерела виникнення затримок у мережі.

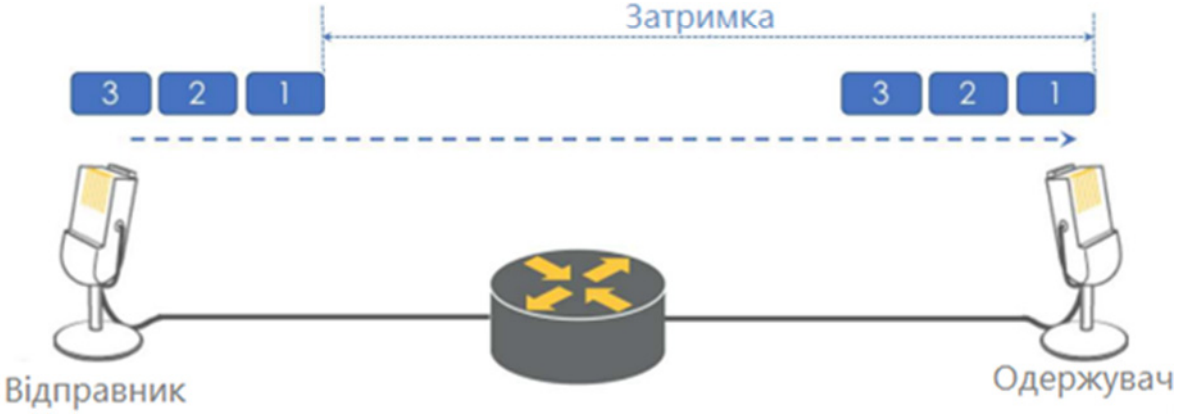

Рисунок 2.2 – Виникнення затримки в мережі

Сукупна затримка складається з кількох компонентів:

- затримка серіалізації (Serialization Delay);
- затримка передачі сигналу в середовищі (Propagation Delay);
- затримка в черзі (Queuing Delay);
- затримка обробки пакетів (Processing Delay).

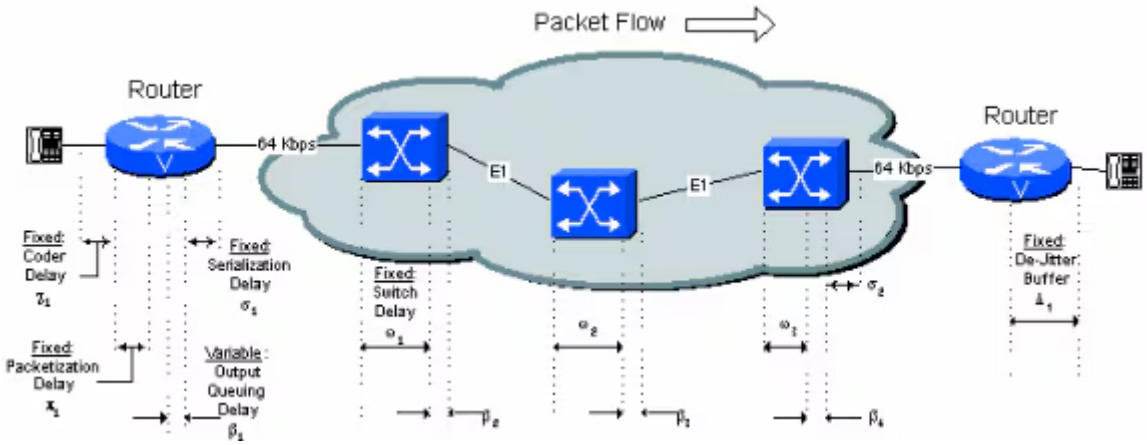

Рисунок 2.3 – Джерела затримки

Затримка серіалізації – це фіксований проміжок часу, необхідний для синхронізації голосу або кадру даних на мережевому інтерфейсі. Вона безпосередньо залежить від тактової частоти магістралі. При низьких частотах і невеликих кадрах додатковий прапорець, що ділить кадри, має велике значення. У таблиці 2.2 показано затримку серіалізації для різних розмірів кадру і швидкостей лінії. Для розрахунків використовується загальний розмір кадру, а не лише корисне навантаження.

Таблиця 2.2 – Затримка серіалізації для різних розмірів пакетів, мс

Розмір фрейма, байт	**Швидкість передачі даних у каналі, kb/c**										
	19,2	**56**	**64**	**128**	**256**	**384**	**512**	**768**	**1024**	**1544**	**2048**
38	15,83	5,43	4,75	2,38	1,19	0,79	0,59	0,40	0,30	0,20	0,15
48	20	6,86	6	3	1,50	1	0,75	0,50	0,38	0,25	0,19
64	26,67	9,14	8	4	2	1,33	1	0,67	0,50	0,33	0,25
125	53,33	18,29	16	8	4	2,67	2	1,33	1	0,66	0,50
256	106,67	36,57	32	16	8	5,33	4	2,67	2	1,33	1
512	213,33	73,14	64	32	16	10,67	8	5,33	4	2,65	2
1024	426,67	149,29	128	64	32	21,33	16	10,67	8	5,31	4
1500	625	214,29	187,50	93,75	46,88	31,25	23,44	15,63	11,72	7,77	5,86
2048	853,33	292,57	256	128	64	42,67	32	21,33	16	10,61	8

Затримка передачі сигналу в середовищі – це час, потрібний для проходження сигналу від джерела до приймача. За емпіричним правилом сигнал проходить один фут дроту приблизно за одну наносекунду. У таблиці 2.3 наведено затримки для кабелю довжиною один метр.

Таблиця 2.3 – Затримка передачі сигналу в середовищі

Delay	**Optical Fiber**	**Twinax CX-1**	**Copper RJ-45**
Pd, ns	5	4,3	5

Затримка в черзі виникає тоді, коли доводиться очікувати, поки маршрутизатор підготує і передасть пакети. Коли маршрутизатор отримує одночасно кілька пакетів для обробки, він створює внутрішню чергу, оскільки здатен обробляти лише один пакет у певний момент часу. Через це виникає затримка, поки маршрутизатор звільнить ресурси та почне передачу в режимі реального часу. Тривалість цього очікування визначається різними факторами.

Під час усунення несправностей у мережі інженери зазвичай вимірюють затримку в черзі та інші подібні показники, щоб пояснити втрату пакетів, уповільнене з’єднання або інші проблеми, про які повідомляють користувачі.

Якщо користувач надсилає один пакет, маршрутизатор, який не перевантажений, може негайно обробити його і передати далі. У випадку пакетної передачі користувач надсилає декілька пакетів одночасно. Крім того, маршрутизатори можуть одночасно отримувати інформацію від багатьох користувачів, які прагнуть передати дані. Це змушує маршрутизатор визначати пріоритети і формувати чергу, оскільки він не може обробляти всі пакети паралельно. Пакети чекають своєї черги, і зазвичай вони обробляються в порядку надходження.

Затримка в черзі може бути мінімальною, коли маршрутизатор отримує обмежену кількість пакетів для обробки. У таких випадках користувачі спочатку можуть не відчувати жодного уповільнення. Проте, якщо пакети накопичуються, час очікування збільшується. Маршрутизатор також може почати відкидати пакети через брак пам’яті для зберігання, що призводить до втрати даних. Це часто спричиняє помилки передачі, особливо при великих обсягах даних, коли буфер маршрутизатора недостатній.

Малі буфери дозволяють утримувати лише обмежену кількість пакетів, перш ніж маршрутизатор почне їх відкидати. Великі буфери забезпечують більше простору для зберігання, але потребують додаткових ресурсів. Проєктувальники мережевого обладнання повинні враховувати потреби мережі та вимоги до маршрутизатора під час створення пристрою. Такі міркування також важливі для правильної конфігурації мережі і маршрутизатора. Зміни у налаштуваннях можуть зменшити затримку та усунути проблеми черги, якщо пристрій здатний їх реалізувати.

Затримка обробки пакетів – це час, який маршрутизатор витрачає на перевірку і обробку кожного пакета. Під час цієї обробки виявляються помилки на рівні бітів, що могли виникнути під час передачі. У високошвидкісних маршрутизаторах затримка обробки зазвичай вимірюється мікросекундами або

менше. При цьому важливо збалансувати затримку обробки та завантаження центрального процесора: менша затримка прискорює передачу кадрів, але збільшує навантаження на CPU.

Ще одним важливим показником є джитер (рис. 2.5).

Рисунок 2.4 – Виникнення джитера

Джитер у мережі – це невеликі періодичні коливання затримки при передачі даних. Вони виникають через перевантаження, колізії або перешкоди сигналу. Технічно джитер відображає різницю між моментом передачі сигналу і його отриманням. Всі мережі мають певну затримку, особливо глобальні, включно з Інтернетом. Вона зазвичай вимірюється мілісекундами і може негативно впливати на додатки реального часу, наприклад онлайн-ігри, потокове відео чи голосовий зв’язок. Джитер лише посилює ці затримки.

Тремтіння мережі виникає, коли пакети надсилаються нерівномірно. Наприклад, після відправки кількох пакетів можуть відразу йти інші, що створює піки навантаження. Це може призвести до втрати пакетів, якщо приймаючий пристрій не здатний обробити всі дані одразу. У разі передачі файлів відсутні пакети доводиться повторно надсилати, що сповільнює передачу. У потокових сервісах реального часу, наприклад відео або аудіо, втрата пакетів може погіршити якість відтворення.

Як показано на рис. 2.5, відправник надсилає пакети з рівномірною швидкістю (наприклад, один пакет на секунду), але через тремтіння мережі пакети доходять до одержувача з різною швидкістю.

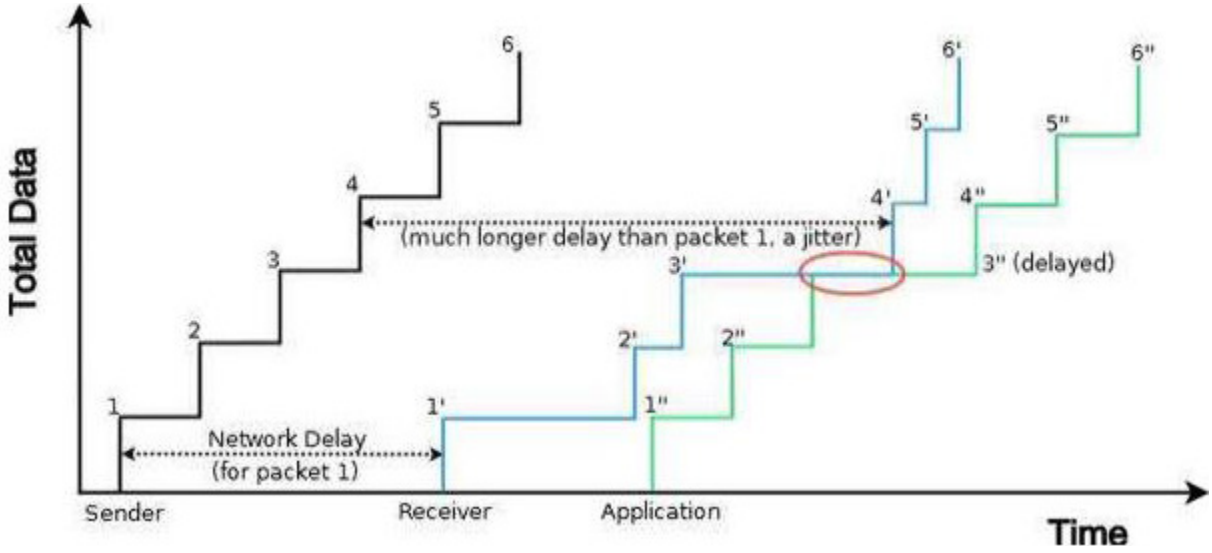

Рисунок 2.5 – Джитер при відправці пакетів

Наприклад, пакет №4 може дійти значно повільніше, ніж пакет №1. Якщо додаток обробляє пакети з постійною швидкістю, він може отримати пакет №1 та №2, але пакет №3 ще не прибув. Це призводить до непередбачуваних ефектів для користувача, наприклад затримок при перегляді онлайн-відео.

Стандартне рішення для компенсації тремтіння – буферизація, коли дані зберігаються на кілька секунд перед відтворенням. Це дозволяє плавніше відтворювати мультимедіа, даючи комп'ютеру час прийняти пакети, що затрималися. Однак у додатках реального часу, таких як ігри або відеоконференції, буфер повинен бути мінімальним. Надто великий буфер (понад 10 мс) викликає помітну затримку.

Ще один аспект – невпорядкована доставка, як показано на рисунку 2.6.

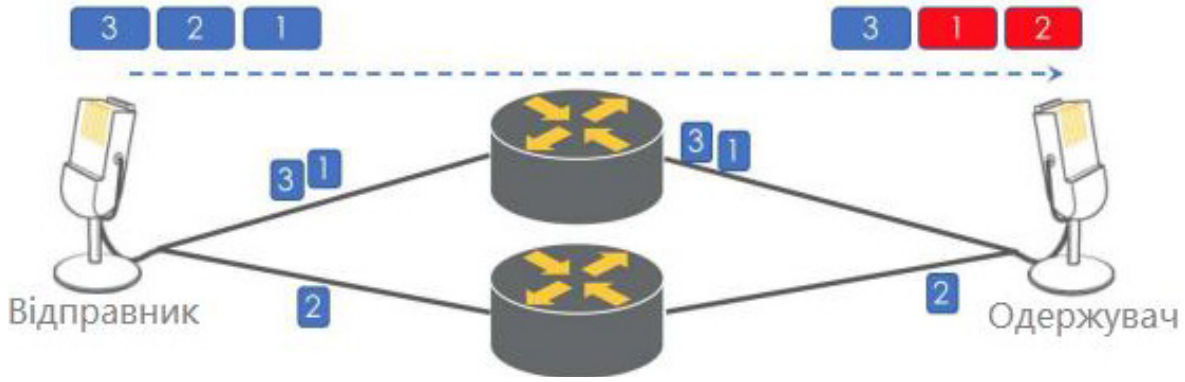

Рисунок 2.6 – Невпорядкована доставка пакетів

Відправник надсилає три пакети з номерами 1, 2 і 3. Через специфіку маршрутизації пакет №2 може дістатися одержувача першим, а пакети №1 та №3 прибудуть пізніше. Це може порушити цілісність даних і файлових систем. Навіть протокол TCP, що стійкий до таких проблем, може реагувати дубльованими підтвердженнями та повторними відправками.

2.3 Опис модулів переналаштування маршрутної інформації

У даній роботі пропонується застосовувати зв’язку двох інструментів – IP SLA та EEM. Обидві ці технології реалізуються в операційній системі Cisco IOS шляхом написання спеціальних сценаріїв. Під час аналізу функціонування високо навантажених мереж було виявлено низку типових проблем, які здатні виникати в середовищі передачі даних, зокрема – втрату пакетів і значну затримку. Щоб досягти поставленої мети, створювані модулі повинні вчасно виявляти та запобігати появі таких небажаних явищ.

Перший модуль побудований із використанням виключно можливостей інструмента Embedded Event Manager (EEM). Завдання цього модуля – попередити виникнення втрати пакетів, виконуючи переналаштування таблиці маршрутизації. Алгоритм його роботи подано у вигляді блок-схеми на рисунку 2.7.

Відповідно до зазначеного алгоритму, на початку перевіряється факт наявності втрати пакетів. Якщо така подія відбулася, сценарій, який функціонує за наперед визначеним алгоритмом, формує повідомлення про виявлену проблему та виводить його у командний рядок маршрутизатора. За потреби можна також налаштувати автоматичне надсилання сповіщення адміністратору електронною поштою. Після цього виконується перехід у режим конфігурації маршрутизатора, де сценарій ініціює процес м’якого перезавантаження сусідніх пристроїв. Після перезавантаження маршрутизатор виконує повторний розрахунок маршрутів і автоматично обирає резервний шлях для подальшої передачі даних.

Рисунок 2.7 – Блок-схема алгоритму запобігання втрати пакетів

Після активації сценарію він безперервно моніторить стан події, пов’язаної з появою втрати пакетів. Для цього подія налаштовується із використанням таких параметрів:

- interface name – визначає інтерфейс, на якому здійснюється контроль;

- parameter – задає назву лічильника, що використовується для спостереження;

- output_packets_dropped – кількість пакетів, відкинутих через заповнення черги виводу;

- entry-op – виконує порівняння поточного значення лічильника з контрольним, використовуючи зазначений оператор. Якщо умова виконується, подія спрацьовує, а моніторинг призупиняється до моменту виконання критеріїв виходу;

- entry-val – вказує порівняльне значення, із яким звіряється поточний показник лічильника, щоб визначити необхідність запуску події. Діапазон можливих значень становить від -2147483648 до 2147483647;

- entry-type – задає тип операції, що застосовується до об’єкта, визначеного аргументом entry-value;

- increment – використовується для обчислення інкрементної різниці між поточним і попереднім значенням лічильника. Якщо отримане від’ємне число, воно вказує на зростання різниці для зменшуваного лічильника;

- pool-interval – задає часовий інтервал між послідовними опитуваннями, стандартне значення становить 1 секунду.

Використовуючи наведені параметри, подія активується на вибраному інтерфейсі й починає відстежувати показник втрати пакетів, порівнюючи його з попереднім значенням. Якщо нове значення виявиться більшим, сценарій автоматично переналаштовує маршрутизацію. Приклад опису інтерфейсу s2/0 на маршрутизаторі R2 до моменту виникнення втрати пакетів і після неї:

```
   Serial2/0 is up, line protocol is up
     Hardware is M4T
     Internet address is 10.0.2.2/24
MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
        reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, crc 16, loopback not set
     Keepalive set (10 sec)
     Restart-Delay is 0 secs
     Last input 00:00:00, output 00:00:00, output hang never
     Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes): Total output drops:
0
     Queueing strategy: fifo
     Output queue: 0/40 (size/max)
     5 minute input rate 0 bits/sec, 0 packets/sec
     5 minute output rate 0 bits/sec, 0 packets/sec
```

```
   Serial2/0 is up, line protocol is up
     Hardware is M4T
     Internet address is 10.0.2.2/24
MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
        reliability 255/255, txload 32/255, rxload 32/255
Encapsulation HDLC, crc 16, loopback not set
     Keepalive set (10 sec)
     Restart-Delay is 0 secs
Last input 00:00:00, output 00:00:00, output hang never
     Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops:
675
     Queueing strategy: fifo
     Output queue: 39/40 (size/max)
     5 minute input rate 198000 bits/sec, 22 packets/sec
     5 minute output rate 199000 bits/sec, 22 packets/sec
```

У нормальному стані черга виводу дорівнює нулю, відповідно, пакети не втрачаються. Після збільшення навантаження черга заповнюється, що призводить до появи втрат. Важливо зазначити, що порівняння значень виконується раз на секунду (pool-interval 1). Зменшити цей інтервал неможливо. Хоча одна секунда може здатися занадто великою затримкою для високошвидкісних мереж, на практиці це не є критичним: навіть при великому потоці даних відновлення передавання відбувається вже через секунду завдяки вибору резервного маршруту.

Другий модуль функціонує згідно з алгоритмом, поданим на рисунку 2.8, і поєднує можливості IP SLA та EEM.

Принцип його роботи полягає в тому, що за допомогою IP SLA маршрутизатор надсилає тестові пакети даних до пристрою-відповідача (responder), який повертає відповіді. Якщо джитер отриманих пакетів перевищує допустиме порогове значення, то EEM реагує на цю ситуацію, генеруючи сповіщення про підвищений джитер. Після цього відбувається перехід у режим конфігурації, де запускається процедура перерахунку метрики маршрутів.

Рисунок 2.8 – Блок схема алгоритму, що визначає затримку

У рамках цього модуля IP SLA налаштовується з такими параметрами:

- icmp-jitter – тип виконуваного тесту;

- source-ip – IP-адреса інтерфейсу, з якого надсилаються тестові пакети;

- num-packets – кількість тестових пакетів, що беруть участь в операції;

- interval – інтервал між відправленням пакетів;

- threshold – верхня межа статистичного значення, при перевищенні якої фіксується подія;

- timeout – час очікування відповіді на запит IP SLA;

- frequency – частота повторення зазначеної операції IP SLA.

Для коректної роботи IP SLA додатково потрібно визначити періодичність виконання тесту та момент його старту. Це реалізується командою ipsla schedule. Також необхідно, щоб пристрій, зазначений у параметрі source-ip, міг відповідати на запити. Для цього на ньому активується команда ipsla responder, після чого пристрій готовий приймати тестові пакети.

Коли тест запущено, важливо перевірити правильність його роботи. Для цього використовується команда show ipsla summary. Приклад її виконання:

```
R2#sh ip sla summary
IPSLAS Latest Operation Summary
Codes: * active, ^ inactive,  ~ pending

ID           Type        Destination    Stats     Return      Last
                                        (ms)      Code        Run
-----------------------------------------------------------------
  *10          icmp-jitter 192.168.3.1    RTT=8     OK    1 second
ago
```

Видно, що тест типу icmp-jitter надсилає пакети на адресу 192.168.3.1, а середня затримка становить 8 мс.

Більш детальну статистику можна отримати командою show ipsla statistics:

```
R2#sh ip sla statistics
IPSLAS Latest Operation Statistics

IPSLA operation id: 10
Type of operation: icmp-jitter
        Latest RIT: 8 milliseconds
Latest operation start time: 16:24:44 KRAT Tue Jun 9 2025
Latest operation return code: OK
RTT Values:
        Number of RTT: 50      RIT Min/Avg/Max: 5/8/10 milliseconds
Latency one-way time:
        Number of Latency one-way Samples: 47
Source to Destination Latency one way Min/Avg/Max: 0/4/5 milliseconds
```

```
        Destination to Source Latency one way Min/Avg/Max: 4/4/5
   milliseconds
Jitter Time:
        Number of SD Jitter Samples: 49
        Number of DS Jitter Samples: 49
        Source to Destination Jitter Min/Avg/Max: 0/1/5 milliseconds
        Destination to Source Jitter Min/Avg/Max: 0/1/1 milliseconds
Over Threshold:
        Number Of RTT Over Threshold: 0 (0%)
Packet Late Arrival: 0
Out of Sequence: 0
        Source to Destination: 0 Destination to Source 0
        In both Directions: 0
Packet Skipped: 0    Packet Unprocessed: 0
Packet Loss: 0
        Loss Periods Number: 0
        Loss Period Length Min/Max: 0/0
        Inter Loss Period Length Min/Max: 0/0
Number of successes: 130
Number of failures: 12
   Operation time to live: Forever
```

Зі звіту видно, що операція ip sla 10 передає 50 пакетів із середньою затримкою відповіді 8 мілісекунд. У цьому ж виведенні відображаються показники односторонньої затримки, джитера, втрат пакетів та інші параметри продуктивності.

Наступним кроком є створення сценарію EEM, який реагуватиме на результати роботи IP SLA. Як і в першому модулі, в EEM задається подія з використанням параметрів ipsla operation-id і reaction-type. Спочатку визначається, із якою саме операцією IP SLA працюватиме сценарій, а потім – на яку подію він має реагувати. Після активації сценарій відстежує стан обраного тесту IP SLA, і якщо отримується негативний статус (що видно у п’ятому стовпчику Return Code), сценарій виконує дії відповідно до алгоритму, зображеного на рисунку 2.8.

2.4 Висновок до розділу

У цьому розділі основна увага була приділена аналізу алгоритмів формування маршрутних метрик і визначенню їх ефективності. На підставі отриманих результатів можна зробити узагальнення, що протокол EIGRP виявився найбільш придатним варіантом для функціонування в умовах високого навантаження на мережу.

Цей протокол демонструє переваги за кількома ключовими параметрами, зокрема такими як час збіжності, обсяг службового трафіку та показник адміністративної відстані. Після порівняльного аналізу було проведено поглиблене дослідження поведінки високонавантажених мережевих систем, у ході якого виявлено низку проблем, що негативно впливають на якість роботи мережі.

Крім того, були описані два функціональні модулі, спрямовані на запобігання можливим втратам і зниження ефективності передачі даних. Перший із них застосовує інструмент Event Manager (EEM) та вирішує питання, пов’язані з втратою пакетів у процесі передавання. Другий модуль поєднує механізми IP SLA та EEM. У цьому випадку система IP SLA контролює поточне значення джитера і, якщо воно перевищує встановлений поріг, формує повідомлення про відхилення.

Сценарій EEM реагує на таке повідомлення та діє відповідно до попередньо визначеного алгоритму, який ініціює перерахунок маршрутних метрик. Після активації обох модулів оптимізації маршрути оновлюються з урахуванням збільшення навантаження на мережу. У результаті метрика перевантаженого каналу зв’язку стає вищою, ніж у резервного каналу, що спричиняє автоматичний вибір альтернативного маршруту.

Таким чином, запропоноване рішення забезпечує підвищення стабільності та ефективності маршрутизації у складних мережевих середовищах із динамічним навантаженням.

3 РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ

3.1 Опис моделі мережі

Запропоновані програмні модулі було протестовано за допомогою моделі мережі у симуляторі Cisco Packet Tracer. Логічна топологія мережі показана на рисунку 3.1.

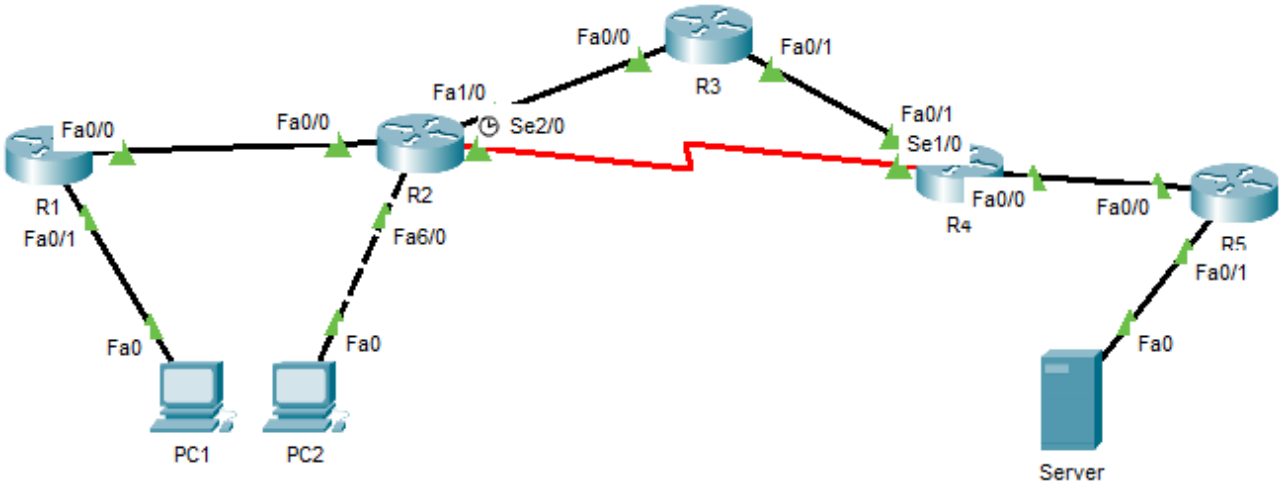

Рисунок 3.1 – Модель мережі у Cisco Packet Tracer

У мережевій установці використовується п’ять маршрутизаторів Cisco (R1–R5) із версією Cisco IOS 15.5, два персональні комп’ютери під керуванням операційної системи Linux (PC1 і PC2), а також один сервер (Server). Для ПК і сервера задаються статичні IP-адреси формату 192.168.*.2, де * відповідає номеру пристрою – наприклад, для PC1 – 192.168.1.2, для PC2 – 192.168.2.2, для Server – 192.168.3.2. Як шлюз за замовчуванням використовується адреса 192.168.*.1, тобто IP-адреса інтерфейсу маршрутизатора, підключеного до відповідного вузла.

Принцип присвоєння IP-адрес такий: усі інтерфейси маршрутизаторів, що з’єднані з ПК або сервером, виконують роль шлюзу для відповідної локальної підмережі й отримують IP виду 192.168.*.1, де * – номер підключеного пристрою (комп’ютера або сервера). Для внутрішніх з’єднань між маршрутизаторами використовується інший принцип: IP-адреси задаються за схемою 10.0.AX.A(X), де A і X – номери маршрутизаторів, між якими встановлено з’єднання, а A(X) –

номер маршрутизатора, на якому знаходиться цей інтерфейс. Наприклад, порт Fa0/0 на R1 отримає адресу 10.0.12.1/24, тоді як відповідний порт Fa0/0 на R2 матиме адресу 10.0.12.2/24. У цьому випадку третій октет дорівнює 12, оскільки лінк з'єднує маршрутизатори з номерами 1 і 2, а четвертий октет – номер конкретного пристрою (в даному випадку – R2).

У таблиці 3.1 наведені всі мережеві параметри для кожного маршрутизатора.

Таблиця 3.1 – Мережеві налаштування маршрутизаторів

Маршрутизатор	Інтерфейс	IP адреса	Маска
R1	Fa0/0	10.0.12.1	255.255.255.0
	Fa0/1	192.168.1.1	255.255.255.0
R2	Fa0/0	10.0.12.2	255.255.255.0
	Fa1/0	10.0.23.2	255.255.255.0
	Fa6/0	192.168.2.1	255.255.255.0
	Se2/0	10.0.24.2	255.255.255.0
R3	Fa0/0	10.0.23.3	255.255.255.0
	Fa0/1	10.0.34.3	255.255.255.0
R4	Fa0/0	10.0.45.4	255.255.255.0
	Fa0/1	10.0.34.4	255.255.255.0
	Se1/0	10.0.24.4	255.255.255.0
R5	Fa0/0	10.0.45.5	255.255.255.0
	Fa0/1	192.168.3.1	255.255.255.0

Щоб застосувати ці налаштування, їх потрібно послідовно прописати на обладнанні. Розглянемо приклад конфігурації для маршрутизатора R2.

Підключаємося до R2 через консольний кабель.

Виконуємо команду *enable*, щоб перейти у привілейований режим.

Далі за допомогою *configure terminal* відкриваємо режим глобальної конфігурації.

Вибираємо інтерфейс Fa0/0 командою *interface FastEthernet 0/0*.

Увімкнути порт можна через *no shutdown*.

Призначаємо IP-адресу командою *ip address 10.0.12.2 255.255.255.0*.

Приклад повного налаштування інтерфейсів маршрутизатора R2:

```
interface FastEthernet0/0
 ip address 10.0.12.2 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet1/0
 ip address 10.0.23.2 255.255.255.0
 duplex auto
 speed auto
!
interface Serial2/0
 ip address 10.0.24.2 255.255.255.0
 clock rate 2000000
!
interface Serial3/0
 no ip address
 clock rate 2000000
 shutdown
!
interface FastEthernet4/0
 no ip address
 shutdown
!
interface FastEthernet5/0
 no ip address
 shutdown
!
interface FastEthernet6/0
 ip address 192.168.2.1 255.255.255.0
 duplex auto
 speed auto
```

Інші порти на різних маршрутизаторах конфігуруються за тією ж логікою.

Після присвоєння IP-адрес усі маршрутизатори зможуть «бачити» своїх безпосередніх сусідів і автоматично формують таблиці маршрутизації, у яких з’являються записи про підключені мережі. Щоб забезпечити повну мережеву доступність між усіма вузлами, потрібно налаштувати протокол маршрутизації. У другому розділі було вирішено використати EIGRP (Enhanced Interior Gateway Routing Protocol).

Процедура налаштування EIGRP виглядає так [1].

Підключитися до маршрутизатора через консоль або SSH.

Увійти до привілейованого режиму командою *enable*.

В режим конфігурації перейти командою *configure terminal*.

Активувати EIGRP за допомогою команди *router eigrp 35*, де 35 – номер автономної системи (AS). Цей номер повинен бути однаковим для всіх маршрутизаторів у топології.

Встановити ідентифікатор маршрутизатора через *eigrp router-id 2.2.2.2*. Як правило, цей ID відповідає адресі loopback-інтерфейсу. Ми використаємо такий принцип призначення ID маршрутизатора – N.N.N.N, де N – номер маршрутизатора. Для R2 відповідно N=2.

Додати оголошення підмереж за допомогою команди *network 192.168.2.0 0.0.0.255* або відповідних значень для кожної підмережі.

Після впровадження конфігурації EIGRP усі маршрутизатори автоматично обмінюються маршрутами, створюючи оновлені таблиці маршрутизації. Для перевірки результату на R2 можна виконати команду *show ip route eigrp*, щоб побачити таблицю з маршрутами, отриманими через EIGRP. Приклад виконання цієї команди:

```
10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
D   10.0.0.0/8 is a summary, 00:01:14, Null0
D   10.0.34.0/24 [90/30720] via 10.0.23.3, 00:01:13, FastEthernet1/0
D   10.0.45.0/24 [90/33280] via 10.0.23.3, 00:01:13, FastEthernet1/0
D 192.168.1.0/24 [90/30720] via 10.0.12.1, 00:01:13, FastEthernet0/0
D 192.168.3.0/24 [90/35840] via 10.0.23.3, 00:01:13, FastEthernet1/0
```

Перед тестуванням потрібно виконати ще кілька кроків. Спочатку слід активувати коефіцієнт k_2, який враховує навантаження під час розрахунку метрики (2.1). Це робиться у режимі конфігурації EIGRP командою

metric weights 0 1 1 1 0 0,

де перший параметр (tos) дорівнює нулю. Після цього метрики маршрутів будуть перераховані з урахуванням завантаження каналів.

Далі потрібно скорегувати параметр *bandwidth* на кількох інтерфейсах: R2 – Fa1/0, R3 – Fa0/0, R3 – Fa0/1 та R4 – Fa0/1. Для цього в режимі конфігурації інтерфейсу задаємо команду *bandwidth 1200*, що зменшує теоретичну пропускну здатність лінку. Приклад зміни значень метрик до підмережі 192.168.3.0 на R2 до:

```
P 192.168.3.0/24, 1 successors, FD is 35840
         via 10.0.23.3 (35840/33280), FastEthernet1/0
         via 10.0.24.4 (20517120/30720), Serial2/0
```

і після корекції параметра bandwidth:

```
P 192.168.3.0/24, 1 successors, FD is 2143488
         via 10.0.23.3 (2143488/2140928), FastEthernet1/0
         via 10.0.24.4 (20517120/30720), Serial2/0
```

Після того як усім інтерфейсам призначено IP-адреси, скориговано параметри ширини смуги пропускання, активовано коефіцієнти для обчислення метрик і сформовано таблиці маршрутизації за допомогою EIGRP, можна переходити до завершального етапу. На цьому етапі до конфігураційних файлів маршрутизаторів додаються спеціальні скрипти, які реалізують модулі переналаштування маршрутизації. Після внесення цих скриптів мережа готова до тестування роботи модулів.

3.2 Тестування процедури переналаштування метрик

Тестування здійснюватиметься також з допомогою моделі, зображеної на рисунку 3.1. Перед стартом експерименту потрібно виконати попередні налаштування, описані у попередньому пункті. Після цього слід створити сценарій, який реалізує роботу першого модуля. Для його коректного функціонування необхідно, щоб маршрутизатор мав не менше двох маршрутів, інакше програма не запуститься. Із рисунка 3.1 видно, що таку умову виконують маршрутизатори R2 та R4. Для проведення тесту використаємо пристрій R2 згідно такої послідовності дій.

Під’єднатися до R2 за допомогою SSH-клієнта, консольного кабелю або іншого засобу доступу.

Увімкнути режим конфігурації командами: *enable*, *configure terminal*.

Командою *event manager session cli username* визначити користувача, від імені якого виконуватиметься сценарій.

Вказати ім’я сценарію командою *event manager applet*.

Задати подію, що слугуватиме тригером для запуску сценарію. Наприклад, командою *event interface name Serial2/0* визначити інтерфейс, за яким спостерігатиме модуль. Потім потрібно налаштувати реакцію на втрату пакетів у межах цього інтерфейсу.

Використовуючи послідовність команд action "Label", де Label – порядковий номер, визначити дії, які повинні виконуватись у разі фіксації втрат пакетів. Конфігурація проводиться відповідно до алгоритму, наведеного на рисунку 2.7.

Після того як модуль буде налаштований, можна переходити до етапів тестування. Випробування складається з трьох частин.

На першому етапі перевіряється реакція модуля при незначному навантаженні на мережу. Для цього на комп’ютері PC2 запускаємо утиліту *ping* із розміром пакета 1400 байт до сервера:

```
1408 bytes from 192.168.3.2: icmp_req=85 ttl=61 time=17.5 ms
1408 bytes from 192.168.3.2: icmp_req-86 ttl-61 time-17.9 ms
1408 bytes from 192.168.3.2: icmp_req-87 ttl-61 time-17.9 ms
1408 bytes from 192.168.3.2: icmp_req-88 tt1-61 time-18.0 ms
1408 bytes from 192.168.3.2: icmp_req-89 tt1-61 time-18.2 ms
1408 bytes from 192.168.3.2: icmp_req=90 ttl=61 time=18.1 ms
1408 bytes from 192.168.3.2: icmp_req=91 ttl=61 time=17.7 ms
1408 bytes from 192.168.3.2: icmp_req=92 ttl=61 time=17.8 ms
1408 bytes from 192.168.3.2: icmp_req-93 ttl=61 time-18.6 ms
1408 bytes from 192.168.3.2: icmp_req-94 ttl-61 time-18.0 ms
1408 bytes from 192.168.3.2: icmp_req-95 ttl-61 time-17.8 ms
^C
---192.168.3.2 ping statistics---
95 packets transmitted, 95 received, 0% packet loss, time 94110ms
     rtt min/avg/max/mdev = 17.209/17.985/29.464/1.214 ms
```

Як видно, усі пакети успішно доходять до адресата, а відповіді повертаються без затримок. Середній час відгуку становить приблизно 17,985 мс.

Далі, на маршрутизаторі R2 переглянемо статистику інтерфейсу Se2/0:

```
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 11000 bits/sec, 1 packets/sec
5 minute output rate 16000 bits/sec, 1 packets/sec
```

Видно, що черга порожня, передача відбувається зі швидкістю одного пакета за секунду.

На другому етапі запускаємо попередньо створений скрипт на PC1, який формує потік даних заданого обсягу. У параметрі Bandwidth встановлюємо значення 1400 кбіт/с. Оскільки пропускна спроможність інтерфейсу Se2/0 становить 1544 кбіт/с, такий потік завантажить канал, але не перевищить його меж. Знову переглянемо статистику інтерфейсу:

```
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 16000 bits/sec, 2 packets/sec
     5 minute output rate 1230000 bits/sec, 123 packets/sec
```

Черга залишається порожньою, а кількість переданих пакетів зростає до 123 пакетів/с, що є середнім показником за 5 хвилин роботи скрипта.

Під час третього етапу збільшуємо навантаження, виставивши параметр Bandwidth на 2000 кбіт/с, тобто більше за доступну пропускну здатність. Це призведе до заповнення черги, а згодом – до відкидання надлишкових пакетів. Після кількох секунд роботи на R2 можна спостерігати відповідну зміну показників. Згідно з алгоритмом, запускається м'який перезапуск сусідніх пристроїв, що викликає оновлення метрик маршрутів. Через збільшене навантаження вартість основного каналу стає вищою, ніж резервного, і тому трафік автоматично перенаправляється альтернативним шляхом:

```
*Nov 10 23:11:53.040: HA EM-6-LOG: INTERFACE: Serial2/0 packet drop detected
R2#
*Nov 10 23:11:53.246: DUAL-5-NBRCHANGE: EIGRP-IPv4 35: Neighbor 10.0.24.4
(Serial2/0) is resync: manually cleared
*Nov 10 23:11:53.246: DUAL-5-NBRCHANGE: EIGRP-IPv4 35: Neighbor 10.0.12.1
(FastEthernet0/0) is resync: manually cleared
*Nov 10 23:11:53.246: DUAL-5-NBRCHANGE: EIGRP-IPv4 35: Neighbor 10.0.23.3
(FastEthernet 1/0) is resync: manually cleared
*Nov 10 23:11:54.041: HA EM-6-LOG: INTERFACE: Serial2/0 packet drop detected
*Nov 10 23:11:54.244: DUAL-5-NBRCHANGE: EIGRP-IPV4 34: Neighbor 10.0.12.1
(FastEthernet 0/0) is resync: manually cleared
*Nov 10 23:11:54.244: DUAL-5-NBRCHANGE: EIGRP-IPv4 34: Neighbor 10.0.23.3
(FastEthernet 1/0) is resync: manually cleared
     *Nov  10  23:11:54.245:  DUAL-5-NBRCHANGE:  EIGRP-IPv4  34:  Neighbor
10.0.24.4 (Serial2/0) is down: manually cleared
```

Затримка при *ping* знизилася до 9,641 мс:

```
1408 bytes from 192.168.3.2: icmp_req-54 tt1-61 time-9.91 ms
1408 bytes from 192.168.3.2: icmp_req-55 ttl=61 time-9.59 ms
1408 bytes from 192.168.3.2: icmp_req-56 ttl=61 time=9.72 ms
1408 bytes from 192.168.3.2: icmp_req-57 ttl-61 time-9.49 ms
1408 bytes from 192.168.3.2: icmp_req=58 ttl=61 time=9.58 ms
1408 bytes from 192.168.3.2: icmp_req-59 ttl=61 time=9.41 ms
1408 bytes from 192.168.3.2: icmp_req-60 tt1-61 time-10.9 ms
^C
--- 192.168.3.2 ping statistics ---
60 packets transmitted, 60 received, 0% packet loss, time 59048ms
     rtt min/avg/max/mdev = 9.254/9.641/10.919/0.295 ms
```

Переконатися, що резервний маршрут став головним, можна за допомогою команди *show ip eigrp topology all-links*. В п. 3.2 було наведено приклад: до навантаження основний маршрут до підмережі 192.168.3.0 проходив через інтерфейс Se2/0 із вартістю 20517120, а резервний мав 2143488. Після

перерахунку метрик основний шлях проходить через Fa1/0, оскільки його вартість стала меншою:

```
P 192.168.3.0/24, 1 successors, FD is 2227557,
       via 10.0.23.3 (2244013/2218413), FastEthernet1/0
            via 10.0.24.4 (2257096/308203), Serial2/0
```

Таким чином, можна зробити висновок, що модуль запобігання втраті пакетів у перевантаженому каналі функціонує правильно.

Для тестування другого модуля необхідно провести його попереднє налаштування у такій послідовності [1].

Підключитися до маршрутизатора R2 за допомогою SSH або консольного кабелю.

Увійти до режиму конфігурації через команди *enable* і *configure terminal*.

Налаштувати IPSLA, вказавши параметри: тип тесту – icmp-jitter, адреса призначення – 192.168.3.1, джерело – 10.0.12.2, кількість пакетів – 50, інтервал – 10.

Встановити параметри *threshold*, *timeout*, *frequency* рівними *30*, *40*, *1* відповідно.

Задати правило запуску тесту командою *ipsla schedule 10 life forever start-time now*.

Визначити реакцію на таймаут, а також кількість повідомлень, після яких виконуватиметься дія: *ip sla reaction-configuration 10 react timeout threshold-type consecutive 2*.

Активувати взаємодію з EEM командою *ipsla enable reaction-alerts*.

Увімкнути виведення повідомлень через *ipsla logging traps*.

Після завершення конфігурації IP SLA тестування розпочнеться автоматично, а система почне збір статистичних даних. Щоб забезпечити реакцію на сповіщення IP SLA, потрібно додатково налаштувати відповідний сценарій EEM.

Налаштування виконується за допомогою таких послідовних команд [1].

Спочатку за допомогою інструкції *event manager session cli username* задається ім’я користувача, від імені якого працюватиме сценарій.

Потім через команду *event manager applet* визначається назва створюваного сценарію.

Наступним кроком, використовуючи *event ipsla operation-id 10 reaction-type timeout*, необхідно вказати, з яким саме тестом буде взаємодіяти сценарій і яку подію він має відстежувати.

Далі, застосовуючи послідовність команд формату *action <Label>*, де Label – порядковий номер дії, налаштовують конкретні кроки, що виконуються при виявленні втрати пакетів. Ці дії формуються відповідно до алгоритму, поданого на рисунку 2.8.

Приклад налаштування може мати такий вигляд [1]:

- "action 001 if $_ipsla_condition eq "Occurred"" – система порівнює значення внутрішньої змінної $_ipsla_condition із рядком “Occurred”. Якщо під час спрацювання події timeout генерується повідомлення “Threshold Occurred for timeout”, і змінна набуває цього значення, то виконуються наступні дії;

- "action 002 cli command "enable"";

- "action 003 syslog msg "Jitter was detected"";

- "action 004 cli command "clear ip eigrp neighbor soft"";

- "action 005 end" — завершує умову if;

- "action 006 cli command "end"";

- "action 007 cli command "exit"".

Після завершення конфігурації модуля можна переходити до етапу перевірки його роботи. Процес тестування відбувається у два кроки, оскільки механізм реагує на появу джитера (коливань затримки), який не завжди присутній. Джитер з’являється, якщо пропускна здатність каналу є низькою, а черга – занадто довгою.

На першому етапі демонструється поведінка модуля при короткій черзі. Для цього запускається попередньо підготовлений сценарій, де у параметрах вказується bandwidth 1544 kb/sec. Це значення відповідає швидкості інтерфейсу S2/0, тому канал працює на повному навантаженні. Початкова довжина черги – 40 пакетів. Додатково на комп’ютері PC2 запускається утиліта ping до сервера.

У результаті можна побачити таку ситуацію: загальний трафік, що проходить через інтерфейс Se2/0 маршрутизатора R2, перевищує його пропускну здатність, через що пакети накопичуються у черзі. Це спричиняє збільшення часу відгуку в утиліті *ping*. Середня затримка становить близько 407,429 мс. Виконавши команду *show interfaces Serial2/0*, можна перевірити параметри черги:

```
Queueing strategy: fifo
Output queue: 32/40 (size/max)
5 minute input rate 33000 bits/sec, 51 packets/sec
     5 minute output rate 1257000 bits/sec, 184 packets/sec
```

Далі переглянемо статистику IPSLA за допомогою *show ip sla statistics*:

```
RIT Values:
        Number of RTT: 0             RTT Min/Avg/Max: 432/450/458 milliseconds
Latency one-way time:
        Number of Latency one-way Samples: 32
        Source to Destination Latency one way Min/Avg/Max: 428/446/453 milliseconds
        Destination to Source Latency one way Min/Avg/Max: 4/4/5 milliseconds
Jitter Time:
        Number of SD Jitter Samples: 26
        Number of DS Jitter Samples: 26
        Source to Destination Jitter Min/Avg/Max: 0/6/20 milliseconds
        Destination to Source Jitter Min/Avg/Max: 0/1/1 milliseconds
Over Threshold:
        Number Of RTT Over Threshold: 32 (100%)
Packet Late Arrival: 32
Out of Sequence: 0
       Source to Destination: 0       Destination to Source 0
       In both Directions: 0
```

Із результатів видно, що хоча середній показник затримки сягає 446 мс, джитер у середньому дорівнює 6 мс. Оскільки у налаштуваннях IP SLA граничне значення для спрацювання становить 30 мс, алгоритм не активується.

На другому етапі експерименту збільшується довжина черги. Для цього потрібно перейти в режим конфігурації інтерфейсу та ввести команду *hold-queue 100 out*, яка підвищує максимальну кількість пакетів у черзі до 100.

Одразу після зміни цього параметра модуль спрацьовує автоматично – виконується перезавантаження сусідів EIGRP, а маршрутизація перебудовується. Причина полягає в тому, що збільшення черги сприяє зростанню джитера: більша кількість пакетів очікує в черзі замість відкидання, через що різниця в часі доставки між ними стає значнішою. Статистика інтерфейсу з подовженою чергою:

```
Queueing strategy: fifo
Output queue: 96/100 (size/max)
5 minute input rate 33000 bits/sec, 51 packets/sec
     5 minute output rate 1051000 bits/sec, 156 packets/sec
```

Модуль успішно зафіксував перевищення порогу джитера та ініціював повторне налаштування маршрутів:

```
*Nov 11 19:20:32.723: %RTT-3-IPSLATHRESHOLD: IP SLAS(10): Threshold
Occurred for timeout
*Oct 11 19:20:32.945: %DUAL-5-NBRCHANGE: EIGRP-IPv4 35: Neighbor
10.0.24.4 (Serial2/0) is resync : manually cleared
*Nov 11 19:20:32.945: %DUAL-5-NBRCHANGE: EIGRP-IPv4 35: Neighbor
10.0.12.1 (FastEthernet0/0) is resync: manually cleared
*Nov 11 09:20:32.945: %DUAL-5-NBRCHANGE: EIGRP-IPV4 35: Neighbor
10.0.23.3 (FastEthernet1/0) is resync: manually cleared
--More--
*Nov 11 09:20:32.957: %HA_EM-6-LOG: IPSLA10_timeout: Jitter detected
--More--
*Nov 11 19:20:34.463: %RTT-3-IPSLATHRESHOLD: IP SLAs(10): Threshold
cleared for timeout
2 carrier transitions        DCD=up DSR=up DTR=up RTS=up CTS=up
```

Для підтвердження результатів можна переглянути метрики командою *show ip eigrp topology all-links*:

```
P 192.168.3.0/24, 1 successors, FD is 2227557, serno 55
       via 10.0.23.3 (2244013/2218413), FastEthernet1/0
            via 10.0.2.4 (2247371/308203), Serial2/0
```

Як видно, метрика маршруту через інтерфейс Fa1/0 виявилась меншою, ніж через Se2/0, що свідчить про коректне переналаштування маршрутизації.

3.3 Висновок до розділу

У цьому розділі подано детальний опис моделі, на якій здійснювалася розробка модулів для динамічного переналаштування маршрутної інформації під час виникнення аварійних або нестандартних ситуацій у мережах із підвищеним навантаженням. На базі цієї моделі також проводилося експериментальне тестування запропонованих рішень.

У межах роботи наведено короткий огляд мережевих параметрів, що забезпечують доступність усіх вузлів системи, а також виконано налаштування

протоколу маршрутизації EIGRP. Після його активації маршрутизатори встановили сусідські з’єднання, обмінялися службовою інформацією та сформували власні таблиці маршрутів.

Після завершення конфігураційного етапу було проведено випробування обох модулів у кілька послідовних стадій. Перший із них, який має на меті мінімізацію втрат пакетів, продемонстрував стабільну роботу та повну функціональність. У разі фіксації втрати пакетів система діяла відповідно до заданого алгоритму, автоматично оновлюючи таблицю маршрутизації з урахуванням зростання навантаження на мережу.

Результати перевірки другого модуля дали змогу зробити висновок, що його застосування доцільне лише у мережах, де наявні канали зв’язку з обмеженою пропускною здатністю, але з доволі великою чергою пакетів. За таких обставин формується значний рівень джитера, який перевищує порогове значення спрацювання алгоритму, завдяки чому модуль коректно реагує на ситуацію. У високошвидкісних каналах черга пакетів обробляється значно швидше, тож коливання затримки не перевищують кілька мілісекунд (приблизно 1-7 ms), і модуль практично не активується.

ВИСНОВКИ

Щороку спостерігається поступове збільшення навантаження на мережеву інфраструктуру. З’являється дедалі більше онлайн-платформ для потокового перегляду, які створюють та поширюють власний контент у форматах високої роздільної здатності. Все більше людей здійснюють професійну діяльність через інтернет, а компанії масово переводять співробітників на дистанційний формат роботи. Усі ці чинники формують значне навантаження на телекомунікаційні мережі. Якщо пропускна здатність мережі не буде відповідати такому рівню використання, це призведе до погіршення якості з’єднання, що, своєю чергою, може знизити ефективність і прибутковість підприємств.

Для запобігання таким проблемам виникла потреба у створенні спеціальних модулів, які дозволяють переналаштовувати маршрутизаційну інформацію при виникненні аварійних або нестандартних ситуацій у мережах із підвищеним навантаженням.

У ході виконання роботи, спрямованої на реалізацію поставлених цілей і завдань, було досягнуто наступних результатів:

- проведено дослідження особливостей функціонування високонавантажених мережевих структур;

- розроблено методику динамічного переналаштування маршрутної інформації;

- створено й перевірено два модулі для оптимізації процесів маршрутизації, працездатність яких підтверджено під час тестових випробувань.

Разом із тим варто врахувати кілька технічних умов:

- для роботи першого модуля необхідно щонайменше один маршрутизатор Cisco, а для другого – два;

- у мережевій структурі має бути реалізована надлишковість з’єднань, адже без неї використання модулів не забезпечить помітного ефекту;

- мінімальна рекомендована версія Cisco IOS – 12.4(22)T або новіша.

Крім того, зроблено висновок, що другий модуль демонструє найбільшу ефективність у тих мережах, де наявний канал зв’язку з невеликою пропускною здатністю та значним розміром черги. У таких умовах рівень джитера стає достатнім, щоб перевищити порогове значення активації модуля, після чого він виконує необхідні дії згідно із закладеним алгоритмом.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1 Маційовський А.І. Методи оптимізації маршрутизації в високо-навантажених мережах : кваліфікаційна робота за освітнім рівнем «магістр» : 123 Комп'ютерна інженерія / Маційовський А. І. – Тернопіль, 2021. – 67 с.

2 RFC 2453 RIP Version 2. URL: https://tools.ietf.org/html/rfc2453, (дата звернення: 30.03.2025)

3 Enhanced Interior Gateway Routing Protocol. Cisco Systems, Inc. URL: https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp / 16406-eigrp-toc.html (дата звертання: 14.04.2025).

4 Adaptive load balancing with OSPF. Research Gate GmbH. URL: https://www.researchgate.net/publication/228787806_Adaptive_load_balancing_with_OSPF (дата звертання: 15.05.2025).

5 RFC 2328, OSPF Version 2. The Internet Engineering Task Force (IETF). URL: https://datatracker.ietf.org/doc/rfc2328/ (дата звертання: 15.05.2025).

6 RFC 1142, OSI IS-IS Intra-domain Routing Protocol. URL: https://tools.ietf.org/html/rfc1142 (дата звертання: 16.05.2025).

7 Gredler, H. The complete IS-IS routing protocol. United States of America: Springer, 2005. 540 с.

8 IPSLAs Configuration Guide . URL: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipsla/configuration/xe-16/sla-xe-16-book.html (дата звертання: 17.05.2025).

9 Embedded Event Manager Configuration Guide. URL: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/eem/configuration/12-4t/eem-12-4t-book.html (дата звертання: 18.05.2025).

10 Nadeau Thomas D. SDN: Software Defined Networks. Sebastopol: O`Reilly Media Inc., 2013. – 384 с.

11 OMNeT++ : OMNeT++ Discrete Event Simulator. URL: https://omnetpp.org (дата звертання: 20.05.2025).

12 Improvement of Performance of EIGRP Network by Using a Supervisory Controller with Smart Congestion Avoidance Algorithm. Research Gate GmbH. URL: https: //www.re-searchgate.net/publication/306925828_ Improvement_of_performance_of_EIGRP_network_by_using_a_supervisory _ controller_with_smart_congestionJavoidanceJalgorithm (дата звертання: 21.05.2025).

13 Johansson M. OSPF Weight Tuning for Efficient Routing in IP Networks: дис. студента магистра: / Johansson Mikael. -Stockholm, 2004. – 67 с.

14 Adaptive load balancing with OSPF. Research Gate GmbH. URL: https://www.researchgate.net/publication/228787806_Adaptive_load_balancing_with_OSPF (дата звертання: 21.05.2025).

15 Васильєв А. С. Порівняння протоколів динамічної маршрутизації // Молодий вчений. 2020. №8. – С. 10-14.

16 Онлайн курси з мережних технологій Cisco IOS. URL : https://networklessons.com/cisco/ccie-routing-switching-written/is-is-metric-on-cisco-ios (дата звертання: 11.10.2025).

17 Tanenbaum, A.S. and Wetheral, D.J., Computer Networks, Englewood Cliffs, NJ: Prentice-Hall, 2010, 5th ed.

18 Cisco Systems. Introduction to EIGRP. Cisco Press, 2020. URL : https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/13669-1.html (дата звертання: 10.11.2025)

19 RFC 4090 – Fast Reroute Extensions to RSVP-TE for LSP Tunnels. IETF, 2005.

20 Feamster N., Rexford J. Network-Wide Predictive Routing. ACM Queue, 2018.

21 Блозва А. І. Комп’ютерні мережі : підручник / А. І. Блозва, Ю. В. Матус, Д. Ю. Касаткін; Нац. ун-т біоресурсів і природокористування України. – Київ : Компрінт, 2019. – 483 с.

22 Микитишин А. Г. Комп’ютерні мережі. Книга 1 : навч. посібник / А. Г. Микитишин та ін. – Львів : Магнолія, 2013. – 256 с.

23 Medhi D. Network Routing: Algorithms, Protocols, and Architectures / Deepankar Medhi, Karthikeyan Ramasamy. – Morgan Kaufmann, 2017. – 824 с.

24 Doyle J. Routing TCP/IP, Volume 1 / Jeff Doyle, Jennifer DeHaven Carroll. – 2nd ed. – Cisco Press, 2005. – 848 с.

25 Huitema C. Routing in the Internet / Christian Huitema. – Prentice Hall, 1995. – 608 с.

26 Aweya J. IP Routing Protocols: Fundamentals and Distance-Vector Routing Protocols / James Aweya. – CRC Press, 2021. – 342 с.

27 Halabi S. Internet Routing Architectures / Sam Halabi. – 2nd ed. – Longman/Pearson Education, 2000. – 472 с.

28 Tadimety P. R. OSPF: A Network Routing Protocol / Phani Raj Tadimety. – Apress, 2015. – 312 с.

29 Graziani R. Routing Protocols and Concepts: CCNA Exploration Companion Guide / Rick Graziani, Allan Johnson. – Cisco Press, 2007. – 976 с.

30 Вишняков В. М. Маршрутизація та комутація в комп’ютерних мережах: методичні вказівки до виконання курсової роботи / В. М. Вишняков. – Київ : КНУБА, 2023. – 24 с.

31 Лисак Н. Маршрутизація в ситуативних мобільних комп’ютерних мережах в умовах високої рухливості вузлів мережі / Н. Лисак, В. І. Месюра, В. Ференець. – Вінниця : ВНТУ, 2008. – 108 с.

32 Волошин П. О. Дослідження параметрів маршрутизації в комп’ютерних мережах : пояснювальна записка до кваліфікаційної роботи / П. О. Волошин. – Харків : ХНУРЕ, 2021. – 90 с.

33 Чепурна І. С. Алгоритми маршрутизації в анонімних мережах : пояснювальна записка / І. С. Чепурна. – Харків : ХНУРЕ, 2025. – 113 с.

34 Воротніков В. В. Багатошляхова маршрутизація у мережах великої розмірності з регулярною фрактальною топологією / В. В. Воротніков. – Київ : ВЕК+, 2015. – 10 с.

35 Шевченко М. Принципи роботи комп’ютерних мереж / Максим Шевченко. – Київ : КНУБА, 2024. – 430 с.

36 Bezruk V. The Analysis of the Characteristics of Routing Protocols in IP Networks / Valery Bezruk, Vyacheslav Varich. – Львів : Видавництво Львівської політехніки, 2010. – 185 с.

37 Kurose J. Computer Networking: A Top-Down Approach / James F. Kurose, Keith W. Ross. – 8th ed. – Pearson, 2021. – 864 с.

38 Varghese G. Network Algorithmics: An Interdisciplinary Approach to Designing Fast Networked Devices / George Varghese. – Morgan Kaufmann, 2004. – 500 с.

39 Kurose J. Computer Networking: A Top-Down Approach / James F. Kurose, Keith W. Ross. – 7th ed. – Pearson, 2020. – 864 с.

40 Rangwala A. Comparative Study of Routing Protocols / Ammar Rangwala, Bhavesh Patil, Paresh Patel // International Journal of Engineering Research & Technology (IJERT). – 2024. – Vol. 13, Issue 10 URL : https://www.ijert.org/comparative-study-of-routing-protocols (дата звертання: 12.11.2025)

41 Golovin Y., Pastukh B. Efficiency analysis of routing protocols in wireless mesh networks / Yuriy Golovin, Bohdan Pastukh // Collection “Information Technology and Security”. – 2022. – Vol. 10 No. 1 URL : https://its.iszzi.kpi.ua/article/view/261121 (дата звертання: 13.11.2025)

42 Hryschuk I. Analysis of Routing Protocols Characteristics in Ad-Hoc Network / Iryna Hryschuk et al. // Information and Telecommunication Sciences. – 2024 URL : https://infotelesc.kpi.ua/article/view/306637 (дата звертання: 15.11.2025)

43 Internet Engineering Task Force. RFC 6551–2012: Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks / – 2017 (upd. 2019). URL : https://interoperable-europe.ec.europa.eu/collection/ict-standards-

procurement/solution/ietf-rfc-6551-2012-routing-metrics-used-path-calculation-low-power-and-lossy-networks/distribution/ietf-rfc-6551-2012-routing-metrics-used-path-calculation-low-power-and-lossy-networks (дата звертання: 21.11.2025)

44 Internet Engineering Task Force. RFC 6115: Recommendation for a Routing Architecture / Tony Li, Ed. – February 2011. URL : https://datatracker.ietf.org/doc/rfc6115/ (дата звертання: 21.11.2025)

45 Заячук Я. І., Сербенюк А. С. Аналіз підходів до маршрутизації у високонавантажених мережах та проблеми їх адаптації. Інформаційні технології в освіті, техніці та промисловості : матеріали Всеукр. наук.-практ. конф., 9 жовтня 2025. Івано-Франківськ : ІФНТУНГ, 2025. С. 265-266. URL : https://drive.google.com/file/d/1sim7SKa62RzaS5XZrTGt248h5VAFf1WQ/view?usp=drive_link (дата звертання: 21.11.2025)

ДОДАТКИ

Додаток А

Налаштування маршрутизатора R2

```
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
aaa new-model
!
aaa session-id common
!
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2
clock timezone KRAT 8 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180!
!
no ip domain lookup
ip domain name router.local
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
cts logging verbose
!
username admin secret 5 $ 1 $ / 6jh $ sk1xtFoZxSx73Rs8.NUVL1
!
redundancy
!
ip ssh version 2
!
interface Ethernet0 / 0
ip address 10.0.12.2 255.255.255.0
!
interface Ethernet0 / 1
bandwidth 1200
ip address 10.0.23.2 255.255.255.0
!
interface Ethernet0 / 2
ip address 10.0.24.2 255.255.255.0
shutdown
```

продовження додатку А

```
!
interface Ethernet0 / 3
ip address 192.168.2.1 255.255.255.0
!
interface Ethernet1 / 0
ip address 10.0.26.2 255.255.255.0
shutdown
!
interface Ethernet1 / 1
no ip address
shutdown
!
interface Ethernet1 / 2
no ip address
shutdown
!
interface Ethernet1 / 3
no ip address
shutdown
!
interface Serial2 / 0
ip address 10.0.2.2 255.255.255.0
serial restart-delay 0
!
interface Serial2 / 1
no ip address
shutdown
serial restart-delay 0
!
interface Serial2 / 2
no ip address
shutdown
serial restart-delay 0
!
interface Serial2 / 3
no ip address
shutdown
serial restart-delay 0
!
router eigrp 34
metric weights 0 1 1 1 0 0
network 10.0.2.0 0.0.0.255
network 10.0.12.0 0.0.0.255
network 10.0.23.0 0.0.0.255
network 10.0.24.0 0.0.0.255
network 10.0.26.0 0.0.0.255
network 192.168.2.0
eigrp router-id 2.2.2.2
!
ip forward-protocol nd
!
```

продовження додатку А

```
no ip http server
no ip http secure-server
!
ip sla 10
icmp-jitter 192.168.3.1 source-ip 10.0.12.2 num-packets 50 interval
10 threshold 30 timeout 40 frequency 1
ip sla schedule 10 life forever start-time now
ip sla reaction-configuration 10 react timeout threshold-type
consecutive 2
ip sla logging traps
ip sla enable reaction-alerts
!
access-list 13 permit 10.0.0.0 0.255.255.255
access-list 23 permit 192.168.0.0 0.0.255.255
!
control-plane
!
line con 0
logging synchronous
line aux 0
line vty 0 4
privilege level 15
logging synchronous
transport input ssh
!
event manager session cli username "admin"
event manager applet IPSLA10_timeout
event ipsla operation-id 10 reaction-type timeout
action 001 if $ _ipsla_condition eq "Occurred"
action 002 cli command "enable"
action 003 cli command "clear ip eigrp neighbors soft"
action 004 syslog msg "Jitter was detected"
action 005 end
action 006 cli command "end"
action 007 cli command "exit"
event manager applet INTERFACE
event interface name Serial2 / 0 parameter output_packets_dropped
entry-opgt entry-val 1 entry-type increment poll-interval 1
action 001 syslog msg "Serial2 / 0 packet drop detected"
action 002 cli command "enable"
action 003 cli command "clear ip eigrp neighbors soft"
action 004 cli command "end"
action 005 cli command "exit"
!
end
```

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи: "Модифікація алгоритму переналаштування маршрутної інформації для підвищення стійкості високонавантажених мереж"

Обсяг пояснювальної записки 59 аркушів.

17 рисунки;

5 таблиці;

1 додатків.

Дата завершення роботи: *10 грудня 2025 р.*

Підпис студента-дипломника _____ *Сербенюк А. С.*