**Міністерство освіти і науки України**
**Національний технічний університет «Дніпровська політехніка»**

_____Інститут електроенергетики_____
(інститут)
_____факультет інформаційних технологій_____
Кафедра інформаційних технологій та комп'ютерної інженерії___

# ПОЯСНЮВАЛЬНА ЗАПИСКА

**кваліфікаційної роботи ступеня** ____бакалавра___
(бакалавра, спеціаліста, магістра)

**Студента**_____Глєбова Ростислава Павловича_____
(ПІБ)

**академічної групи**_____123-19-1_____
(шифр)

**спеціальності** _____123 Комп'ютерна інженерія____
(код і назва спеціальності)

**за освітньо-професійною програмою**
_____«Комп'ютерна інженерія»_____
(офіційна назва)

**на тему** Моделювання та оптимізація процесів маршрутизації в комп'ютерних мережах

| **Керівники** | **Прізвище, ініціали** | **Оцінка за шкалою** | | **Підпис** |
|---|---|---|---|---|
| | | **рейтинговою** | **інституційною** | |
| кваліфікаційної роботи | проф. Гнатушенко В.В. | | | |
| розділів: | | | | |
| Аналітична частина | проф. Гнатушенко В.В. | | | |
| Практична частина | проф. Гнатушенко В.В. | | | |
| **Рецензент** | проф. Мороз Б.І. | | | |
| **Нормоконтролер** | проф. Цвіркун Л.І. | | | |

Дніпро
2023

**ЗАТВЕРДЖЕНО:**
завідувач кафедри
____інформаційних технологій____
___та комп'ютерної інженерії____
(повна назва)
_____ __Гнатушенко В.В.__
(підпис) (прізвище, ініціали)
«_____»_____ 2023 року

**ЗАВДАННЯ**
**на кваліфікаційну роботу**
**ступеня _____бакалавра _____**
(бакалавра, спеціаліста, магістра)
**студенту** _____Глєбову Р.П. **академічної групи** _____123-19-1____
(прізвище та ініціали) (шифр)

**спеціальності** ___123 Комп'ютерна інженерія_____

**за освітньою-професійною програмою**
_____«Комп'ютерна інженерія»_____

**на тему** Моделювання та оптимізація процесів маршрутизації в комп'ютерних мережах

затверджену наказом ректора НТУ «Дніпровська політехніка» від 16.05.2023р. № 350-с

| Розділ | Зміст | Термін виконання |
|---|---|---|
| 1 Аналітична частина | На основі матеріалів з відповідних джерел та даних проаналізувати існуючі рішення оптимізації. Розглянути інструменти оптимізації маршрутизації. | 10.05.2023 р. |
| 2 Практична частина | Реалізація алгоритмів, що спрямовані на вирішення виниклих проблем в високонавантаженій мережі. Розробити модулі, що працюють на маршрутизаторах Cisco | 10.06.2023 р. |

**Завдання видано** _____ Гнатушенко В.В.
(підпис керівника) (прізвище, ініціали)

**Дата видачі** _____

**Дата подання до екзаменаційної комісії** 12.06.2023 р.

**Прийнято до виконання** _____ Глєбов Р.П.
(підпис студента) (прізвище, ініціали)

# АНОТАЦІЯ

Текстова частина випускної роботи: 57 сторінок, 33 рисунка, 4 таблиці, 11 джерел.

*Об'єкт дослідження:* процес маршрутизації в високонавантажених мережах.

*Предмет дослідження:* способи переналаштування маршрутної інформації в високонавантажених мережах.

*Мета роботи* – розробити методику і провести тестування модуля переналаштування маршрутної інформації при виникненні позаштатних ситуацій в високонавантажених мережах.

Проведено дослідження предметної області, виявлені недоліки існуючих рішень оптимізації. Розглянуто інструменти оптимізації маршрутизації. Виконано дослідження високонавантажених мереж. Були виявлені проблеми, які можуть виникати в мережах передачі даних з високим навантаженням. Запропоновано два алгоритми, спрямовані на вирішення виниклих проблем в високонавантаженій мережі. На основі алгоритмів розроблені модулі, що працюють на маршрутизаторах Cisco. Отримані модулі були протестовані на лабораторному стенді, працездатність модулів була підтверджена.

Ключові слова: МАРШРУТИЗАЦІЯ, IP SLA, EEM, SDN, ВИСОКОНАВАНТАЖЕНА МЕРЕЖА, ДЖИТЕР.

# ABSTRACT

Graduate work: 57 pp., 33 figures, 4 tables, 11 sources.

The object of research is the process of routing in highly loaded networks.

The subject of the study is ways of reconfiguring route information in highly loaded networks.

The purpose of the work is to develop a methodology and conduct testing of the route information reconfiguration module in the event of out-of-hours situations in highly loaded networks.

The research of the subject area is carried out, the shortcomings of the existing optimization solutions are revealed. Routing optimization tools are considered. A study of high-load networks was performed. Problems that may occur in high-load data networks have been identified. Two algorithms aimed at solving problems in a highly loaded network are proposed. Modules running on Cisco routers have been developed based on algorithms. The advantages of using the module can be obtained in networks that have a communication channel with low bandwidth and a fairly large queue size. Inthis situation, the jitter will be high enough to exceed the threshold of the module, as a result of which the module will work according to the proposed algorithm. The obtained modules were tested on a laboratory stand, the efficiency of the modules was confirmed.

Keywords: ROUTING, IP SLA, EEM, SDN, HIGH LOAD NETWORK, JITTER.

ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

CLI (Command-line interface) – інтерфейс командного рядка. EEM – Embedded Event Manager.

EIGRP (Enhanced Interior Gateway Routing Protocol) – дистанційно-векторний протокол маршрутизації.

IP SLA – Internet Protocol Service Level Agreements.

IS-IS (Intermediate System to Intermediate System) — протокол маршрутизації проміжних систем.

OSPF (Open Shortest Path First) – протокол динамічної маршрутизації.

RIP (Routing Information Protocol) — протокол маршрутизації в невеликих комп'ютерних мережах/

SDN (Software Defined Network) – програмно – конфігурована мережа.

АС – автономна система (англ. Autonomous system), набір маршрутизаторів з підтримкою EIGRP, які повинні стати сусідами EIGRP.

Джитер (англ. jitter – тремтіння) – небажані фазові та/або частотні випадкові спотворення під час передачі сигналу.

МП – мережевий пристрій.

ПК – персональний комп’ютер.

# ВСТУП

Комп'ютерні мережі – це складна технологія, яка дозволяє кінцевим пристроям зв'язуватися один з одним. Типова мережева інфраструктура включає в себе маршрутизатори, комутатори, сервери, веб- сервери, міжмережеві екрани, балансувальник навантаження, системи запобігання вторгнень і інші пристрої. Ефективність, надійність, гнучкість і міцність  це вимоги для обробки і управління величезним обсягом інформації, котрий передається по мережі.

Все це змусило компанії-виробники реалізовувати складні і ресурсомісткі протоколи, які дозволяють маршрутизаторам і комутаторам взаємодіяти один з одним при допомозі комутації пакетів і створення топології мережі для цілей маршрутизації. На часі зараз питання щодо організації високонавантажених мереж, про використання того чи іншого протоколу динамічної маршрутизації. Обсяги трафіку, котрий пересилається, зростають з кожним днем. Зростають потужності підприємства, зростає і навантаження на мережу. Так само, досить часто, виникає потреба модернізації мереж. До того ж самі протоколи динамічної маршрутизації створюють навантаження на мережу службовим трафіком. Тому з'явилася необхідність в поліпшенні протоколів динамічної маршрутизації, а також поліпшенні алгоритмів пошуку оптимального маршруту.

**Мета дослідження:** розробити методику і провести тестування модуля переналаштування маршрутної інформації при виникненні позаштатних ситуаційв високонавантажених мережах.

В роботі поставлено та розв’язано **наступні задачі**:

– здійснити аналіз предметної області, виявити переваги і недоліки існуючих методів оптимізації;
– провести дослідження високонавантаженої мережі;
– розробити механізм переналаштування маршрутної інформації;
– провести тестування отриманих результатів на лабораторному стенді.

**Наукова новизна отриманих результатів.**

– Проведено ґрунтовний огляд протоколів внутрішньо доменної маршрутизації RIP, EIGRP, OSPF, IS-IS та інструментів IP SLA, EEM, SDN;

– експериментально доведено, що протокол EIGRP є найкращим кандидатом для використання високонавантажених мережах, оскільки він володіє кращими показниками за такими критеріями як: час збіжності, обсяг службового трафіку і адміністративна дистанція;

– розроблена методика переналаштування маршрутної інформації при виникненні позаштатних ситуацій в високонавантажених мережах;

– запропонована конструкція лабораторного стенду для тестування модулів переналаштування маршрутної інформації.

**Практичне значення одержаних результатів**. Впровадження результатів проведеного дослідження дозволить в мережах з маршрутизаторами Cisco переналаштувати таблицю маршрутизації з умовою збільшеного навантаження для забезпечення мінімальної втрати пакетів.

# 1 АНАЛІТИЧНА ЧАСТИНА

## 1.1 Огляд літературних джерел

Роботи по оптимізації маршрутизації ведуться не перший рік. Через зростання мереж і трафіку, котрий проходить по них, з'являється потреба в способах оптимізації доставки пакетів даних з однієї точки в іншу [1-11]. Автори, наприклад, проводять дослідження, в яких розглядають підвищення продуктивності традиційного протоколу внутрішньо доменної маршрутизації EIGRP, за рахунок додавання деякого функціоналу SDN у вигляді модифікованого підходу для поліпшення ряду показників продуктивності мережі, таких як завантаження каналів зв'язку, показник втрати пакетів. Покращення цих показників веде за собою покращення пропускної здатності. Такий підхід реалізується часто за допомогою мережі, в якій присутній інтелектуальний динамічний, диспетчерський контролер-коректор, який здатний виявити місце виникнення перевантаження, причому до виникнення перевантаження або на межі його виникнення. Після виявлення перевантаження, контролер-коректор втручається для вирішення проблеми перевантаження. Він застосовує запропоновані алгоритми, які складаються з трьох підалгоримтів (процесів); по-перше, підалгоритму розподілу потоку можливих наступників, потім підалгоритму тимчасового наступника і, нарешті, підалгоритму очищення шляху. Ці підалгоритми застосовуються послідовно для роботи з конкретним потоком і тільки в умовах високої інтенсивності трафіку, де також висока ймовірність перевантаження. Якщо поточний підалгоритм вирішує проблему перевантаження, тоді немає необхідності застосовувати наступний підалгоритм.

Якщо перші два процеси алгоритму, застосованого до певного потоку, не можуть вирішити проблему перевантаження, третій процес повторить перші два процеси на іншому потоці і так далі. За рахунок застосування цього алгоритму скорочується генерація керуючих повідомлень.

Іноді проблема балансування зростаючого навантаження вирішується через швидке збільшення інтернет-трафіку і вхідних додатків реального часу в IP-мережах.

Використовуючи рішення оптимальної спільної задачі маршрутизації, яка визначається за допомогою мультимодальної потокової регуляції, як основи для вимірювання продуктивності запропонованого алгоритму, автори показали, що останнє розширення евристичного методу виконується в межах декількох відсотків від оптимального. Результати показують, що знайдене налаштування ваги хороше для нових гарячих точок і збоїв зв'язку, тому, як правило, ми не очікуємо зміни налаштування ваги. Крім того, надійне налаштування ваги виявляється гарною відправною точкою для нового пошуку останнього реалізованого методу Рамакрішнана, що наближається до оптимального, особливов випадку відмови каналу.

Результати показують, що оптимізація коефіцієнтів поділу трафіку покращує продуктивність мережі в порівнянні з рівним розподілом. Коли набір найкоротших шляхів малий, також має сенс зміна ваг OSPF.

## 1.2 Протоколи динамічної маршрутизації

RIP. Зазвичай він використовується в невеликих мережах, оскільки його дуже просто налаштовувати і обслуговувати, але в ньому відсутні деякі розширеніфункції протоколів маршрутизації, таких як OSPF або EIGRP.

Існують дві версії протоколу: RIPv1 і RIPv2. Обидві використовують кількість переходів як метрики і мають адміністративну відстань 120. Щоб зробити RIP більш відповідним для сучасних мереж, в RIP v2 були поліпшені багато функції RIP v1. RIPv2, в свою чергу, використовує мультикастову адресу (англ. Multicast address) 224.0.0.9 для відправки оновлень маршрутизації і вміє працювати з мережевими масками. Інформація про маршрутизації в мережі RIP передається за допомогою пакетів RIP-запитів і RIP-відповідей. Маршрутизатор, який тільки що завантажився, може

транслювати запит RIP на всі інтерфейси з підтримкою RIP. Всі маршрутизатори, що використовують RIP на цих каналах, отримують запит і відповідають, відправляючи у відповідь пакет маршрутизатора. Пакет відповіді містить інформацію таблиці маршрутизації, необхідну для побудови локальної копії карти топології мережі. На додаток до звичайної передачі пакетів RIP кожні 30 секунд, якщо маршрутизатор виявляє нового сусіда або виявляє, що інтерфейс недоступний, він генерує ініційоване оновлення. Нова інформація про маршрутизації негайно передається в усі інтерфейси з підтримкою RIP, і ця зміна має відображатися у всіх наступних відповідних пакетах RIP. На рис. 1.1 проілюстровані таймери протоколу RIP.

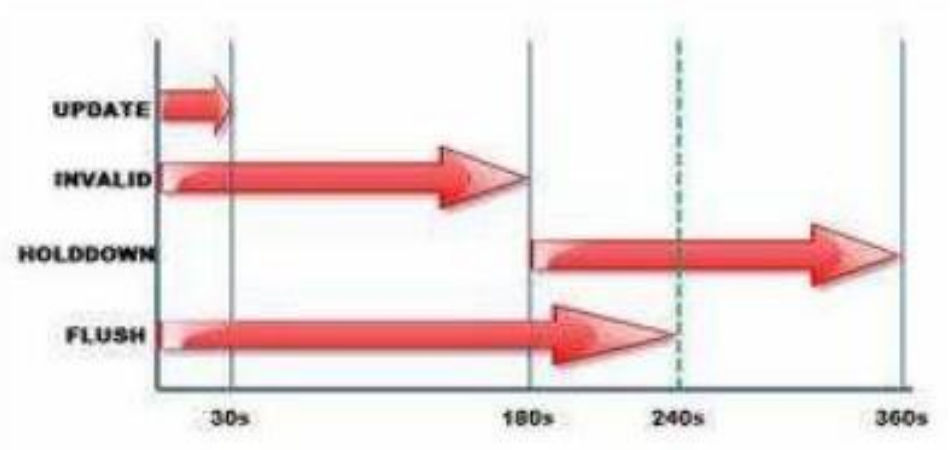


Рисунок 1.1 – Таймери протоколу RIP

EIGRP. Є вдосконаленим протоколом маршрутизації векторної відстані. Він підтримує безкласову маршрутизацію і VLSM, додавання маршрутів, інкрементніпоновлення, балансування навантаження і багато інших корисних речей. Є власним протоколом Cisco, тому всі МП, де працює EIGRP, повинні бути маршрутизаторами Cisco.

Маршрутизатор з EIGRP мають стати сусідами перед обміном інформацією про маршрутизацію. Протокол розраховує свою метрику, використовуючи пропускну здатність, затримку, надійність і навантаження. За замовчуванням при обчисленні метрики застосовуються тільки пропускна здатність і затримка, а два інші параметри встановлюються на нуль. EIGPR

застосовує концепцію АС. Кожен маршрутизатор всередині АС повинен мати один і той же номер АС, інакше маршрутизатори не стануть сусідами. Щоб встановити зв'язок, EIGRP застосовує 5 варіантів пакетів, наочно це можна побачити на рис. 1.2.

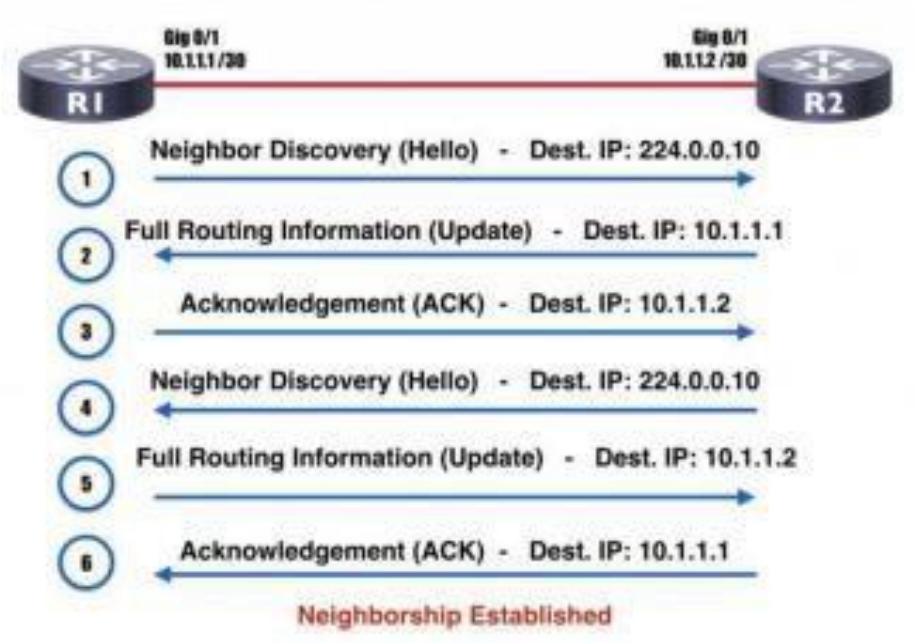


Рисунок 1.2 – Встановлення сусідства по протоколу EIGRP

OSPF. Це відкритий стандарт, він реалізується різними мережевими постачальниками. OSPF буде працювати на більшості маршрутизаторів. OSPF є безкласовим протоколом маршрутизації, підтримує VLSM, CIDR, ручне підсумовування маршрутів, а також рівномірне балансування навантаження. Як метрики застосовується тільки один параметр – вартість інтерфейсу. Застосовує багатоадресні адреси 224.0.0.5 і 224.0.0.6 для відновлення маршрутизації.

Алгоритм встановлення сусідства і встановлення маршрутизації між двома маршрутизаторами показаний на рис. 1.3.

IS-IS. Так само, як і OSPF є протоколом внутрішнього шлюзу, який використовує інформацію щодо стану каналу для прийняття рішень про маршрутизацію. Він оцінює зміни топології і визначає, чи виконувати повний перерахунок SPF або обчислення часткового маршруту. У великих мережах

(більше 500 маршрутизаторів), IS-IS забезпечує швидку збіжність.

Мережа IS-IS – це окрема АС, також звана доменом маршрутизації, яка складається з кінцевих і проміжних систем. Варто пояснити, що кінцеві системи – це мережеві об'єкти, які відправляють і отримують пакети; проміжні системи – крім того і ретранслюють (пересилають) їх.

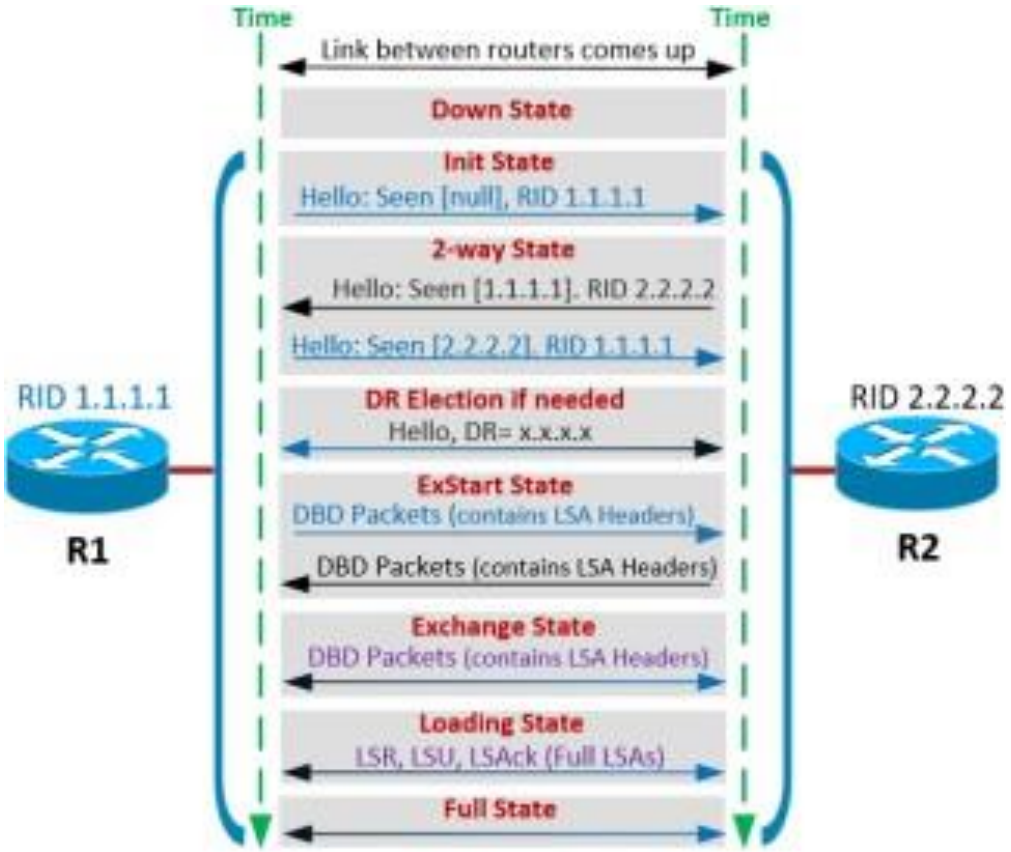


Рис 1.3 – Процес встановлення сусідства за OSPF

### 1.3 IP SLA

Ця технологія від Cisco активно моніторить трафік (генерує його безперервно, передбачувано і надійно) щоб визначити продуктивність мережі. IP SLA шле інформацію між декількома мережевими точками або кількома мережевими шляхами. Технологія моделює мережеві дані та IP-сервіси і збирає інформацію щодо продуктивності в реалі. В ній міститься час відгуку, односторонньої затримки, джитера, втрату пакетів, оцінку якості мови, доступність мережевих ресурсів, а також продуктивність додатків. Ця

технологія генерує і аналізує трафік для визначення продуктивності або між пристроями Cisco, або з пристрою Cisco на віддалений IP-пристрій, такий як сервер мережевих додатків. Статистичні дані по вимірюваннях, що надаються різними операціями IP SLA, може використовуватися для усунення неполадок, аналізування різного роду проблем та проектування мережевої топології.

Використовуючи IP SLA, клієнти постачальника послуг можуть вимірювати і надавати угоди власне щодо рівня обслуговування, для корпоративних клієнтів доступна функція перевірки цих рівнів і розуміння продуктивності мережі для нових або існуючих послуг і додатків IP. Пакети мають налаштовувані параметри IP і прикладного рівня, зокрема IP-адреса джерела і отримувача, номера портів протоколу, котрі визначені для користувача, байт типу обслуговування, примірник маршрутизації / пересилання віртуальної VPN і URL. Будучи незалежними від транспортного рівня 2, IP SLA може бути налаштований наскрізним чином по різнорідним мережам, щоб найкращим чином відображати метрики, які може випробувати кінцевий користувач.

Використовуючи IP SLA, мережевий інженер може відстежувати продуктивність між будь-областю мережі: ядром, розподілом і периферією. Моніторинг можна проводити в будь-який час і в будь-якому місці, без розгортання фізичного зонда. IP SLA використовує згенерований трафік для визначення продуктивності мережі між двома мережевими пристроями. На рис. 1.4 показано, як IP SLA запускається, коли пристрій посилає згенерований пакет за призначенням. Після того, як цільовий пристрій отримає пакет і, в залежності від типу операції IP SLA, пристрій відповість інформацією мітки часу для джерела, щоб виконати розрахунок по метриках продуктивності.

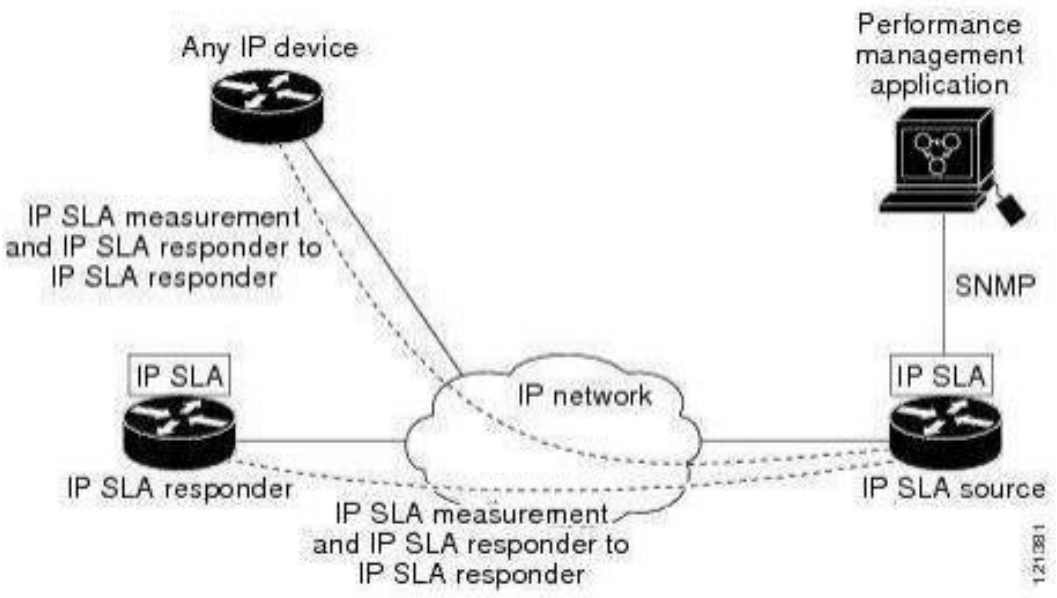


Рисунок 1.4 – Операції IP SLA

## 1.4 EEM

Це унікальна підсистема в ПЗ Cisco IOS. Власне EEM – це потужний і гнучкий інструмент для автоматизації завдань і налаштування поведінки ПЗ Cisco і роботи пристрою. Клієнти можуть використовувати EEM для створення і запуску програм або сценаріїв безпосередньо на маршрутизаторі або комутаторі. Сценарії називаються політиками EEM і можуть програмуватися з використанням простого інтерфейсу на основі CLI або з використанням мови сценаріїв Tcl. EEM дозволяє клієнтам використовувати значний інтелект в ПЗ Cisco IOS для реагування на події в реальному часі, автоматизації завдань, створення команд, які налаштовуються, і виконання локальних автоматичних дій на основі умов, виявлених самим ПЗ CiscoIOS.

EEM – це, в основному, незалежна від продукту функція ПЗ, котра складається з серії детекторів подій, сервера EEM і інтерфейсів, які дозволяють викликати підпрограми дій, звані політиками. Існують також внутрішні інтерфейси прикладного програмування для інших підсистем ПЗ Cisco IOS, що дозволяють використовувати переваги підсистеми EEM. На рис. 1.5 зображено компоненти EEM.

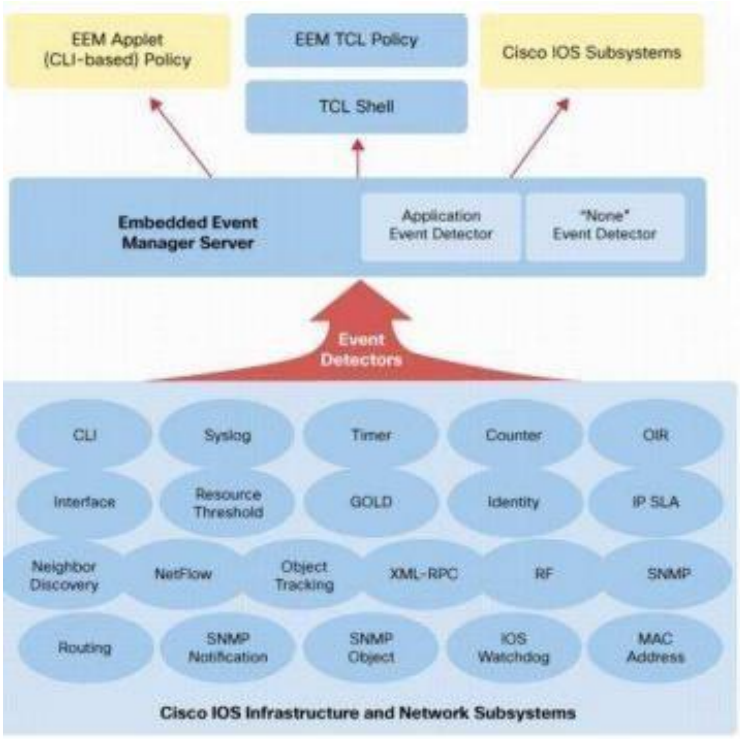


Рисунок 1.5 – Архітектура EMM

Є два типи політик EEM:

– політика аплету: простий у використанні інтерфейс; визначається задопомогою CLI;

– політики Tcl: більш гнучкі і широкі можливості; визначається з використанням Tcl.

Як тільки одна або декілька політик визначені, ПЗ детектора подій буде відслідковувати умови, які відповідають тим, котрі визначені політикою. При виникненні умови, подія передається на сервер диспетчера подій. Потім сервер викликає якусь політику, котра зареєстрована для даного конкретного випадку. Потім виконуються дії, визначені в рамках політики. Кожен тип події має певні параметри, параметри і детальну інформацію, яка доступна політиці при її виклику.

### 1.5 SDN

Є новим поколінням архітектури мереж, де керування нею здійснюється абсолютно незалежно від переадресації і є безпосередньо

програмованим. Ця міграція керування, раніше тісно пов’язана з окремими МП, дає змогу відволікти основні функції для додатків і мережевих служб, які можуть розглядати мережу як логічний або віртуальний об'єкт. Рис. 1.6 показує логіку архітектури SDN.

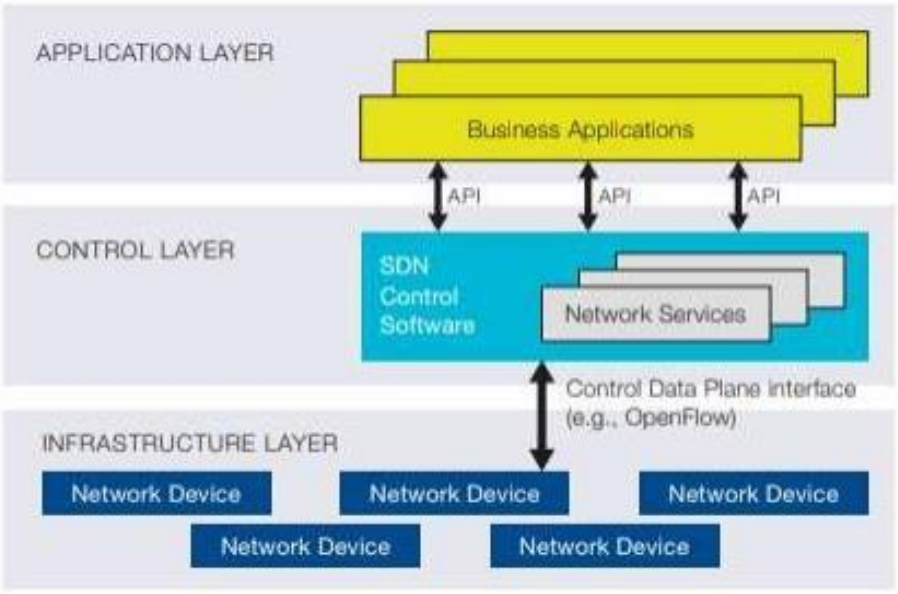


Рисунок 1.6 – Архітектура SDN

Варто відмітити, що мережевий «розум» зібраний в програмних контролерах SDN, котрі володіють глобальним поглядом на мережу. Як наслідок мережа сприймається додатками і механізмами політики як спільний логічнрий

пристрій перемикання. Завдяки SDN підприємства і оператори отримують окремий дозвіл контролю з єдиною логічною точкою над всією мережею. Цей факт робить простішим конструювання та експлуатацію мережі. Власне SDN на додачу робить простішими і МП, тому що для них пропадає потреба в розумінні і опрацюванні великої кількості стандартів протоколів. Оскільки потрібно всього лиш отримувати чіткі вказівки від власне контролерів SDN.

Ймовірно саме головне – мережеві оператори і адміністратори мають змогу оперувати програмно з цим спрощеним представленням мережі, а не

вручну кодувати величезну кількість стрічок конфігурації, котрі ще й розкидані по тисячам МП.

Окрім того, застосовуючи інтелектуальні централізовані можливості контролера SDN, IT-відділ може міняти поведінку мережі в реалі і розвертати свіжі додатки та сервіси упродовж кількох днів чи навіть годин, а не так як зараз це вимагає місяці чи тижні. SDN дає змогу менеджерам гнучко настроювати мережу, керувати, здійснювати захист ресурсів при допомозі динамічних програм SDN, котрі є автоматизованими. Більш того, вони можуть самі створювати ці програми і не чекати, поки функції будуть вбудовані в пропрієтарні і закриті середовища в центрі мережі.

Також архітектури SDN дозволяють використовувати масив API-інтерфейсів, котрі надають змогу реалізовувати спільні мережеві сервіси, в т.ч. маршрутизацію, багатоадресну передачу, безпеку, керування доступом, керування смугою пропускання, керування трафіком, енергоспоживання і всі види керування політиками, оптимізацію процесора і сховища, спеціально розроблені для досягнення бізнес-цілей. Як варіант, структура SDN дає змогу просто визначати і використовувати узгоджені політики для різного роду з'єднань в кампусі. Аналогічно, SDN дає змогу керувати всією мережею з використанням інтелектуальних систем оркестрації і забезпечення. Open Networking Foundation здійснює вивчення відкритих API-інтерфейсів для просування керування декількома постачальниками, що відкриває дозволяє розподіл ресурсів за вимогою, ва також самообслуговування, дійсно віртуалізованих мереж і безпечних хмарних сервісів. Як наслідок з open API-інтерфейсами між рівнями керування SDN та додатків властиво бізнес-додатки мають змогу робити в абстракції мережі, застосовуючи її можливості чи / або сервіси. Відсутня потреба бути прив’язаними до якихось деталей їх втілення. SDN дає змогу мережі не стільки «знати про додатки», скільки «бути налаштованою для них», а програми не стільки« обізнані про мережі», скільки «обізнані про їх можливості». Як наслідок всі ресурси мережі (обчислювальні, зберігання і т.п.) можуть оптимізуватися.

OpenFlow – це перший інтерфейс (стандарт) зв'язку, котрий врегульований на рівнях керування та пересилання в архітектурі SDN. Він забезпечує безпосередній доступ і керування площиною пересилки МП фізичних та віртуальних (з використанням гіпервізора). Саме неприсутність open- інтерфейсу з площиною пересилання дозволило сучасним МП бути монолітними, закритими і уподібнюватися мейнфреймам. Ніякий більше стандартний протокол не може виконувати те, що виконує OpenFlow, і такий протокол є необхідним для переміщення керування мережею з мережевих комутаторів в логічно централізоване кероване ПЗ. Протокол можна порівняти з набором команд процесора.

Як наведено на рис. 1.7, OpenFlow показує головні елементи, які можуть використовуватися зовнішнім програмним додатком для програмування площини пересилання МП, точно так, як набір команд ЦП буде здійснювати програмування комп'ютерної системи.

OpenFlow оперує концепцією потоків для визначення трафіку мережі на базі заданих параметрів відповідності, які можуть статично або динамічно програмуватися керуючим ПЗ SDN. Це також дозволяє IT-відділу визначати, як трафік повинен проходити через МП, на базі таких параметрів, як моделі використання, додатки і хмарні ресурси.

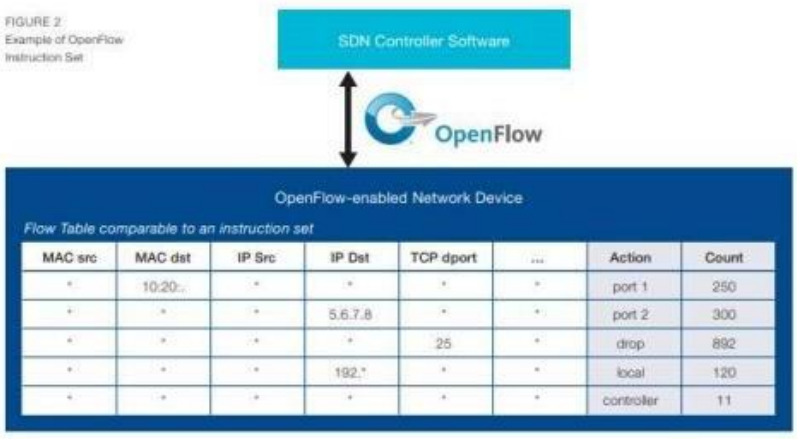


Рисунок 1.7 – Приклад набору вказівок

Так як OpenFlow дозволяє робити програмування мережі для кожного

потоку, архітектурне рішення SDN з використанням OpenFlow фіксує надзвичайно детальний контроль, дозволяючи мережі реагувати на зміни в реальному часу на рівні додатків, користувачів і сеансів. Поточна маршрутизація на основі IP не може надати такий рівень контролю, тому що всі абсолютно потоки, які проходять між двома кінцевими точками мають йти по одному і тому ж шляху через мережу, незалежно від їх різних вимог.

Варто згадати і те, що протокол OpenFlow є основним засобом підтримки SDN і в даний час є єдиним стандартним протоколом, який дозволяє здійснюватипряму маніпуляцію площиною пересилки МП.

Початково застосовувана для мереж на базі Ethernet, комутація OpenFlow може поширюватися на набагато більш широкий набір варіантів використання. SDN на основі OpenFlow можна розгорнути в наявних мережах (фізичних чи віртуальних). МП підтримують переадресацію на основі OpenFlow, а також традиційну переадресацію, що дозволяє підприємствам і операторам дуже легко впроваджувати технології SDN на основі OpenFlow навіть в мережевих середовищах різних постачальників.

Open Networking Foundation уповноважений стандартизувати OpenFlow і робить це через технічні робочі групи, відповідальні за протокол, конфігурацію, тестування сумісності та інші дії, допомагаючи забезпечити взаємодію між МП і керуючим ПЗ від різних постачальників. OpenFlow широко застосовується постачальниками інфраструктури, які зазвичай реалізують його за допомогою простого оновлення прошивки або ПЗ. Архітектура SDN на основі OpenFlow може безперешкодно інтегруватися з існуючою інфраструктурою підприємства або оператора і забезпечити простий шлях міграції для тих сегментів мережі, які найбільше потребують функціональності SDN.

Як для підприємств, так і для операторів зв'язку SDN дає мережі бути конкурентно відмінною, а не просто неминучим центром витрат. Технології SDN на основі OpenFlow дозволяють IT-фахівцям вирішувати проблему високої пропускної здатності, динамічного характеру сучасного ПЗ,

адаптувати мережу до постійно змінних потреб бізнесу і дуже знизити складність операцій і керування.

Переваги, які можуть отримати підприємства і оператори завдяки архітектурі SDN на основі OpenFlow, включають:

– централізоване управління середовищами різних постачальників: ПЗ управління SDN може керувати будь-яким МП з підтримкою OpenFlow від будь- якого постачальника, включаючи комутатори, маршрутизатори та віртуальні комутатори. Замість того, щоб управляти групами пристроїв від окремих постачальників, IT-фахівці можуть використовувати інструменти оркестрації і управління на основі SDN для швидкого розгортання, налаштування та оновленнявсіх МП;

– зниження складності за рахунок автоматизації: SDN на основі OpenFlow пропонує гнучку інфраструктуру для автоматизації та керування мережею, яка може створювати програми, котрі автоматизують багато задач керування, котрі виконуються зараз вручну;

– більш високий рівень інновацій: застосування SDN прискорює інноваціїв бізнесі, дозволяючи операторам IT-мереж буквально програмувати і перепрограмувати мережу в режимі реального часу для задоволення конкретних потреб бізнесу і користувачів по ходу їх виникнення;

– підвищення надійності і безпеки мережі: SDN дозволяє IT фахівцям визначати високорівневі конфігурації і заяви про політику, які потім транслюються в інфраструктуру через OpenFlow. Архітектура SDN на основі OpenFlow усуває необхідність індивідуального налаштування МП при кожному додаванні або переміщенні кінцевої точки, служби або програми, або зміні політики, що знижує ймовірність збоїв мережі через невідповідності конфігураціїабо політики;

– більш детальне управління мережею. модель управління OpenFlow на основі потоків дозволяє IT-фахівцям застосовувати політики на дуже детальному рівні, включаючи рівні сеансу, користувача, пристрою і додатки, в вигляді абстрактного, автоматизованого способу;

– поліпшення взаємодії з користувачем. Завдяки централізації управління мережею і надання інформації про стан для додатків вищого рівня, інфраструктура SDN може краще адаптуватися до динамічних потреб користувачів.

## 1.6 Висновки до розділу

У цьому розділі здійснено аналіз літературних джерел за темою роботи. Також зроблено огляд протоколів внутрішньо доменної маршрутизації, зокрема: RIP, EIGRP, OSPF, IS-IS. Достатньо докладно наведено процес роботи кожного з них. Були розглянуті такі інструменти як: IP SLA, EEM, SDN. За підсумком огляду інструментів було прийнято рішення використовувати зв'язку IP SLA і EEM, так як IP SLA дозволяє виявляти ряд подій в мережі, в тому числі виникнення затримки і втрати пакетів, а EEM вміє реагувати на події IP SLA. Дана зв'язка інструментів дозволяє в повній мірі досягти поставленої мети – оптимізувати маршрутизацію для високонавантажених мереж. Функціонал SDN також дозволяє досягти мети, однак програмування контролерів – це процес трудомісткий і, як правило, вимагає роботи відразу групи програмістів з великим досвідом роботи.

# 2 РОЗРОБКА МЕТОДИКИ ПЕРЕНАЛАШТУВАННЯ МАРШРУТНОЇ ІНФОРМАЦІЇ

## 2.1 Порівняльний аналіз протоколів динамічної маршрутизації

У першому розділі було дано огляд чотирьох протоколів внутрішньо доменної динамічної маршрутизації. Виділимо три критерії для порівняння протоколів: час конвергенції, кількість службового трафіку, адміністративна дистанція.

Далі описується лабораторний стенд, зібраний з шести маршрутизаторів Cisco і трьох комп'ютерів, і описує початкове налаштування МП. Потім по черзі здійснюємо конфігурування кожного протоколу динамічної маршрутизації і знімаємо показники за обраними критеріями. Результат роботи представлений в таблиці 2.1.

Таблиця 2.1 – Результати вимірювання критеріїв

| Критерій | RIP v2 | EIGRP | OSPF | IS-IS |
|---|---|---|---|---|
| Час конвергенції, с | 8,590100 | 0,609551 | 9,867057 | 9,467620 |
| Об’єм службового трафіка, байт | 356 | 596 | 706 | 2517 |
| Адміністративна дистанція | 120 | 90 | 110 | 115 |

За підсумками аналізу отриманих результатів робиться висновок, що протокол EIGRP є кращим для використання в високонавантажених мережах. Додатково варто порівняти алгоритми кожного протоколу для розрахунку метрик.

Метрика по протоколу RIP складається з кількості проміжних маршрутизаторів від початкового хоста до кінцевого. Після розсилання мінімальної таблиці маршрутизації сусідам, маршрутизатор нарощує кожне отримане поле метрики на одиницю і заносить в свою таблицю поля з найменшим значенням метрики. Після чого відбувається розсилання нових

таблиць маршрутизації і їх оновлення, поки не буде встановлено коректний режим маршрутизації. Такий алгоритм має серйозний недолік. При використанні протоколу RIP в великих мережах (до 15 вузлів) повідомлення з повними таблицями маршрутизації будуть істотно завантажувати канали зв'язку, а в мережах з кількістю вузлів більше 15-ти, RIP просто не працюватиме, тому використовувати даний протокол в великих, високонавантажених мережах неприпустимо.

Протокол EIGRP, розроблений компанією Cisco, при підрахунку метрики, використовує формулу 2.1.

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

де $k_1, \ldots, k_6$ – коефіцієнти для включення компонентів формули;

*bw* - найменша пропускна здатність на всьому маршруті;

*load* – найгірший показник завантаження каналу зв'язку на всьому шляху;

*delay* – сума затримок, налаштованих на інтерфейсах;

*extattributes* – показник джитера і енергії, витраченої на маршрут;

*reliability* – найгірший показник надійності на всьому маршруті.

Змінюючи значення коефіцієнтів $k_1, \ldots, k_6$ , можна гнучко налаштовуватизначення метрики. Це дає перевагу над іншими протоколами динамічної маршрутизації, так як дозволяє враховувати більшу кількість факторів, що впливають на маршрут [17]. За замовчуванням формула 2.2. розрахунку метрики маршруту виглядає так:

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

де bw і delay – те ж, що і у (2.1).

Протокол маршрутизації OSPF використовує метрику cost, розраховану за формулою (2.3):

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

У формулі (2.3) $10^8$ – це еталонна пропускна здатність, яка дорівнює 100 Мбіт/с. Наприклад, з'єднання Fast Ethernet має значення метрики рівним 1 одиниці, а Ethernet – рівним 10 одиницям. Для маршруту, що проходить через два і більше маршрутизатора, значення метрик сумуються.

Метрика в OSPF (формула (2.3)) враховує смугу пропускання каналу зв'язку, так само як і метрика EIGRP (формула (2.2)), але не враховує інші фактори, що впливають на канали зв'язку [18].

Протокол IS-IS підтримує чотири різних значення метрики:

– Default metric: кожен інтерфейс має метрику 10 за замовчуванням;
– Delay: аналогічно тому, як EIGRP використовує затримку;
– Expense: фактична грошова вартість каналу зв'язку;
– Error: аналогічно тому, як EIGRP використовує надійність (Reliability).

За замовчуванням маршрутизатори підтримують тільки default metric, це означає, що значення метрики маршруту буде рівною кількості проміжних вузлів. Даний підхід до розрахунку метрики, схожий з RIP, що не підходить для використання в високонавантажених мережах.

## 2.2 Дослідження високонавантажених мереж

Високонавантажена мережа – це мережа передачі даних, через яку проходить величезна кількість трафіку. Гігабіти інформації йдуть від одного вузла до іншого через що виникають ситуації із затримкою передачі інформації, а іноді і з втратою інформації. Така ситуація може виникнути в магістральних мережах передачі даних. Наприклад, відеоконференція віддаленого офісу, в іншому місті, з головним офісом. При підвищеного навантаження на мережу магістрального провайдера, відео дані будуть приходити з затримкою, через що відео на екранах учасників конференції

буде відображатися з зависанням. Те ж саме і з голосом – він буде постійно перериватися. Ще одним прикладом є епідеміологічна ситуація, коли більшість компаній переводять своїх співробітників на віддалену роботу, а люди, що знаходяться на карантині, частіше користуються різними відео онлайн сервісами, це все створює високе навантаження на мережу передачі даних. Тому працювати віддалено стає складніше, а відео онлайн сервіси змушені знизити якість своїх трансляцій, тому що мережа не витримує такого навантаження.

Як було сказано вище, в високонавантажених мережах виникають ситуації, коли проходить трафік забиває канали зв'язку, як наслідок з'являється затримка і втрати пакетів. З цього можна виділити метрики якості мережі передачі даних:

– втрати пакетів;
– затримки;
– джитер.

Метрика втрати пакетів говорить про те, скільки відправлених пакетів дійшло до адресата. На рис. 2.1 показана ситуація втрати пакетів. Відправник передає три пакети даних, на маршрутизаторі відбувається втрата пакетів, внаслідок чого одержувачу приходять не всі дані.

У мережі передачі даних постійно відбуваються втрати пакетів, але в більшості випадках це не є критичним. Однак втрати пакетів негативно впливаютьна передачу даних в режимі реального часу. Наприклад, телефонна розмова.

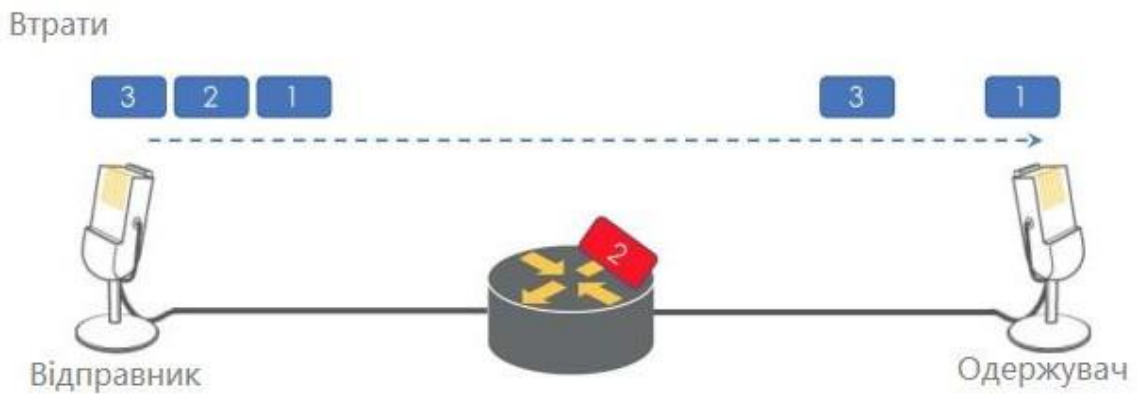


Рисунок 2.1 – Ілюстрація втрати пакетів

Причинами втрати пакетів можуть бути наступні обставини. Переповнення мережі – часта причина в високонавантажених мережах. Так як процес передачі пакетів слід певним крокам, збої в каналах зв'язку можуть привести до втрати пакетів, для того щоб мережа змогла впоратися із збільшеним навантаженням.

Помилки ПЗ також можуть бути причиною виникнення втрати пакетів. Погано написані і не протестовані додатки, що працюють з мережею передачі даних, швидше за все, приведуть до проблеми з мережею. І в свою чергу вплинутьна передачу пакетів.

Також втрати пакетів можуть бути пов'язані із застарілим або несправним обладнанням: маршрутизатори, комутатори, брандмауери. Таке обладнання може значно сповільнювати проходження пакетів даних. Через зростання трафіку, що проходить, підвищене навантаження на застаріле обладнання може викликати затримку і втрату пакетів.

Ще одна можлива причина втрати пакетів – кібернетична атака. Зловмисник може провести атаку падіння пакетів. Через що втрата пакетів неминуча.

Всі перераховані вище причини так чи інакше можуть викликати втрату пакетів в мережі передачі даних. Залежно від типу переданої інформації втрата пакетів може привести до різних наслідків. Зображення можуть бути отримані нечіткими. Аудіо файли можуть викликати шум або незрозумілу мову. Наступна метрика – затримки. Сукупна затримка – це час, необхідний пакетам даних дістатися від відправника до одержувача. На рис. 2.2 проілюстрована ситуація виникнення затримки, а на рис. 2.3 показані джерела затримки пакетів в мережі.

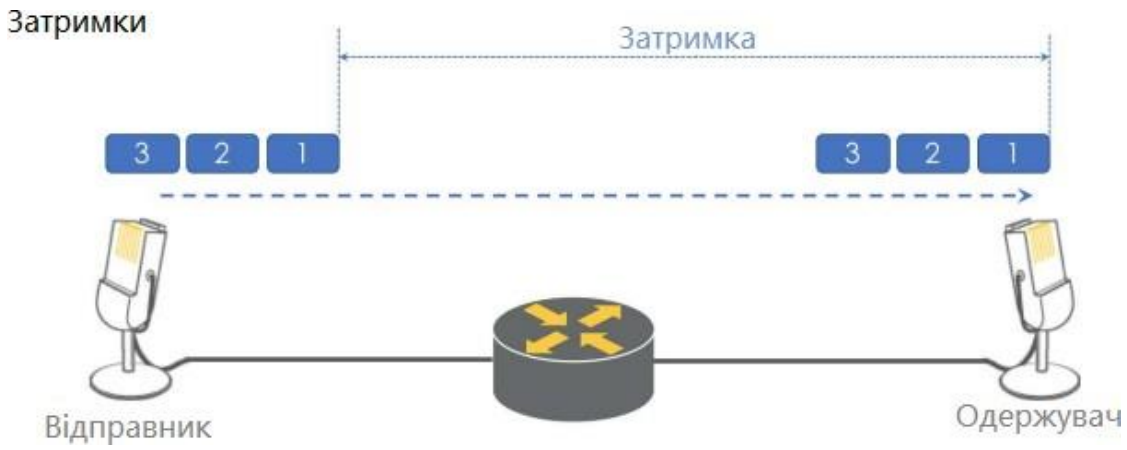


Рисунок 2.2 – Виникнення затримки в мережі

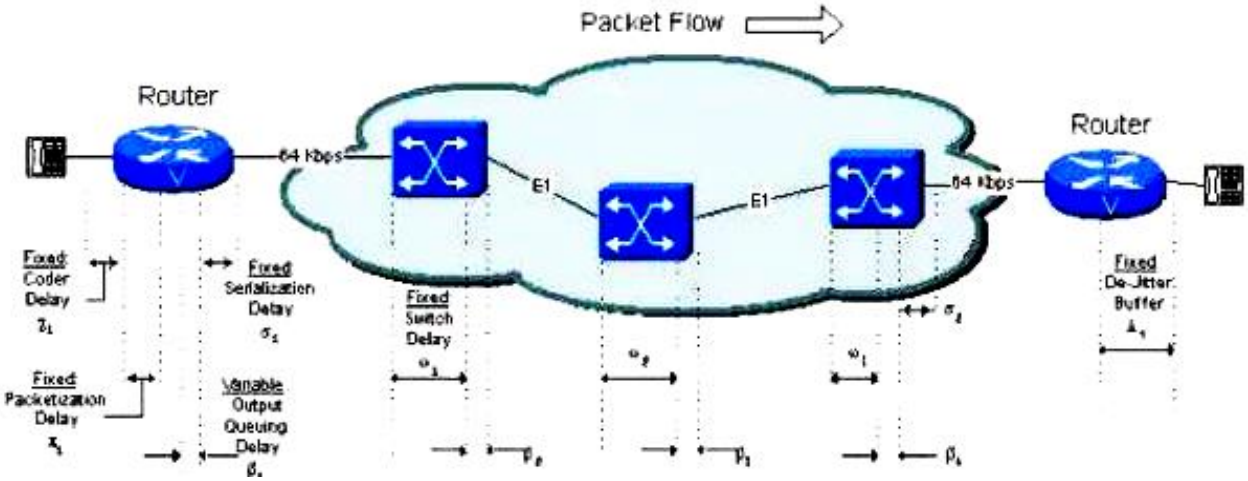


Рисунок 2.3 – Джерела затримки

Сукупна затримка складається з ряду компонентів:

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

Затримка серіалізації – це фіксована затримка, необхідна для синхронізації голосу або кадру даних на мережевому інтерфейсі. Це напряму зв’язано з тактовою частотою на магістралі. При низьких тактових частотах і невеликих розмірах кадрів додатковий прапорець, необхідний для поділу кадрів, має велике значення. У таблиці 2.2 показана затримка серіалізації,

необхідна для різнихрозмірів кадру на різних швидкостях лінії. У цій таблиці для розрахунку використовується загальний розмір кадру, а не розмір корисного навантаження.

Таблиця 2.2 – Затримка серіалізації для різних розмірів пакетів (ms)

| Розмір кадра (bytes) | Швидкість передачі даних по каналу (kb/сек) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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 |
| 128 | 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, необхідний для передачі сигналу від джерела до одержувача. Емпіричне правило полягає в тому, щосигнали можуть проходити один фут проводу за одну наносекунду. У таблиці 2.3 показана затримка передачі сигналу в середовищі для кабелю довжиною один метр.

Таблиця 2.3 – Затримка передачі сигналу в середовищі

| Delay for 1m | Optical Fiber | Twinax CX-1 | Copper RJ-45 |
|---|---|---|---|
| Pd (ns) | 5 | 4,3 | 5 |

Затримка в черзі – це очікування, поки маршрутизатор готує і передає пакети. Маршрутизатори з декількома пакетами для обробки налаштовують чергудля обробки, оскільки вони можуть обробляти тільки один пакет за раз.

Це створює затримку, поки маршрутизатор НЕ зможе очистити дані і почати передачу в режимі реального часу. Тривалість затримки залежить від декількох факторів. При усунення неполадок в мережі технічні фахівці можуть перевіряти затримку в черзі і подібні проблеми, щоб пояснити втрачені пакети, повільні з'єднання та інші скарги.

Коли користувач відправляє одиночний пакет, маршрутизатору котрий очікує, пристрій може негайно обробити і відправити інформацію. При пакетній передачі користувач відправляє кілька пакетів одночасно. Маршрутизатори також можуть отримувати інформацію від декількох користувачів, які намагаються відправляти пакети одночасно. Це змушує маршрутизатор призначати пріоритети і створювати чергу, оскільки він не може обробляти їх одночасно. Пакети чекають, поки маршрутизатор встигне обробити і відправити їх, і вони зазвичай обробляються в порядку надходження.

Затримка в черзі може бути дуже короткою, коли маршрутизатор має обмежену кількість пакетів, які мають бути надіслані. В цьому випадку користувачі можуть спочатку не помітити відставання. Однак, коли пакети починають складатися, затримка може збільшуватися. Маршрутизатор також може почати відкидати пакети. Йому не вистачає місця для зберігання інформації, яка чекає обробки, і він повинен відкинути деякі пакети, щоб залишатися працездатним, що пов'язане з втратою даних. Це може створити такі проблеми, як пакетні помилки, при яких відбувається збій при передачі великих даних, оскількиу маршрутизатора недостатньо буфера.

Невеликі буфери допускають тільки обмежену кількість пакетів в черзі, перш ніж маршрутизатор почне відкидати дані. Великі буфери створюють більше місця для зберігання, але також вимагають більше ресурсів. Проектувальники повинні думати про потреби мережі і вимоги, які можуть пред'являтися до маршрутизатора при розробці обладнання. Такі міркування також важливі для конфігурації маршрутизатора та мережі. Зміни в налаштуваннях можуть іноді усувати затримки і проблеми, такі як затримка в

черзі, якщо маршрутизатор має можливість реалізувати зміни.

Затримка обробки пакетів – це той час, який витрачає маршрутизатор для обробки пакета. Обробка пакета дозволяє виявити помилки на рівні бітів, які можуть виникнути під час передачі пакета до одержувача. У високошвидкісних маршрутизаторах затримки обробки пакетів зазвичай становлять близько мікросекунд або менше. Додатково необхідно збалансувати затримку обробки пакетів із завантаженням центрального процесора. Менша затримка призводить до більшої швидкості передачі кадрів і до більшого завантаження центрального процесора.

Ще одна метрика – це джитер. На рис. 2.4 показана ситуація з виникненням джитера.


Рисунок 2.4 – Виникнення джитера

Джитер в мережі означає невеликі періодичні затримки при передачі даних. Це може бути викликано рядом факторів, включаючи перевантаження мережі, колізії і перешкоди сигналу. Технічно, джитер – це затримки між тим, коли сигнал передається і коли він приймається. Всі мережі відчувають деяку затримку, особливо глобальні мережі, які охоплюють Інтернет. Така затримка, зазвичай вимірюється в мілісекундах, може бути проблематичною для додатків реального часу, таких як онлайн-ігри, потокова передача і цифровий голосовий зв'язок. Джитер посилює це, викликаючи додаткові затримки.

Тремтіння мережі викликає відправку пакетів з нерегулярними інтервалами. Наприклад, може бути затримка після того, як деякі пакети відправлені, і тоді кілька пакетів можуть бути відправлені всі відразу. то може привести до втрати пакету, якщо приймаюча система не може обробити всі вхідні пакети. Якщо це станеться під час завантаження файлу, втрачені пакети будуть відправлені повторно, що сповільнить передачу файлу. У разі служби в реальному часі, такої як потокова передача звуку, дані можуть бути просто втрачені, що призведе до втрати або зниження якості аудіосигналу.

Як показано на рис. 2.5, відправник відправляє пакети з постійною швидкістю (скажімо, один пакет в секунду; чорна лінія), але пакети досягають одержувача зі змінною швидкістю (синя лінія) через тремтіння мережі. Як показано, пакет 4 займає набагато більше часу для переміщення по мережі, ніж пакет 1. Тим часом, давайте припустимо, що додаток споживає прийняті пакети зпостійною швидкістю (знову ж таки, один пакет в секунду). Додатку вдалося отримати пакет 1 і 2 своєчасно, але не вдалося, коли він спробував отримати пакет3, оскільки пакет 3 ще не був доставлений! В залежності від програми, відсутній пакет буде викликати зміни в сприйнятті користувача. Наприклад, докучливі зависання при перегляді онлайн відео потоків.

Стандартний спосіб компенсації тремтіння в мережі – це використання буфера, в якому зберігаються дані до його використання, наприклад, кілька секунд аудіо чи відео. то згладить відтворенням мультимедіа, оскільки дає приймає комп'ютера кілька секунд для прийому будь-яких пакетів, втрачених через тремтіння. хоча буфери є ефективним рішенням, вони повинні бути дуже маленькими при використанні в додатках реального часу, таких як онлайн-ігри і відео-конференції. Якщо буфер занадто великий (більше 10 мс), це викличе помітну затримку.

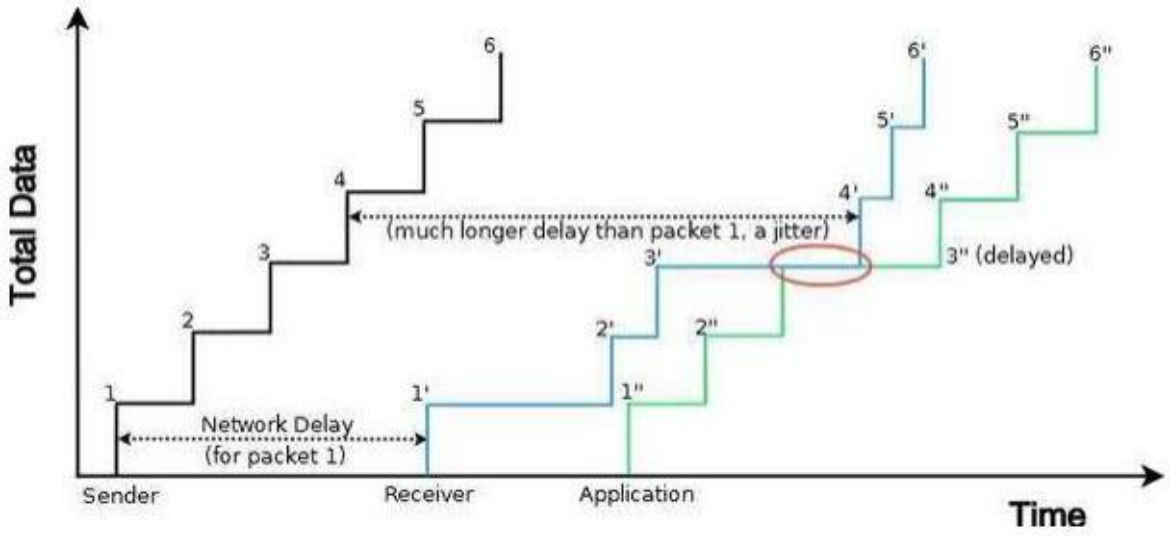


Рисунок 2.5 – Ілюстрація джитера при відправці пакетів

Додатково варто сказати про невпорядковану доставку, показану на рис. 2.6.

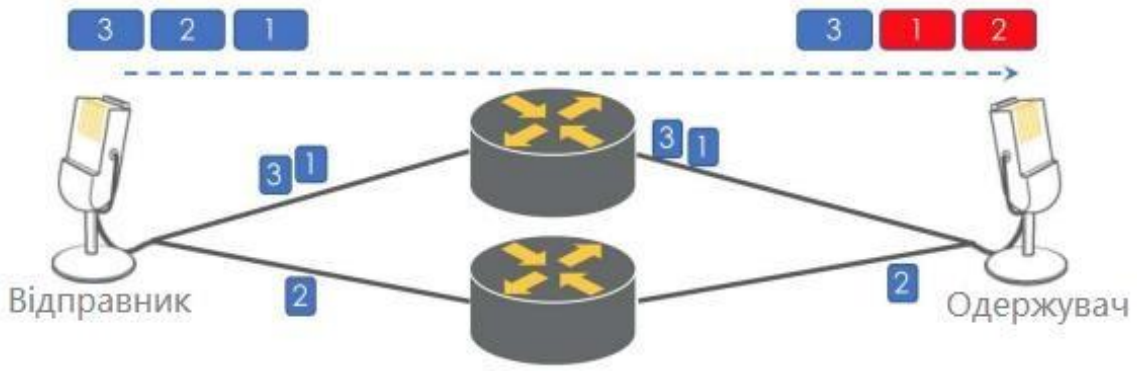


Рисунок 2.6 – Невпорядкована доставка пакетів

Як показано вище відправник посилає три пакети даних, по порядку, зномерами 1, 2 і 3. В силу певних обставин пакет під номером 2 пройшов через другий маршрутизатор і "прилетів" першим до одержувача. Пакети з номерами 1 і 3 "прилетіли" пізніше другого. Дана помилка може призводити до втрати зв'язності і пошкодження файлової системи. І навіть протокол TCP, який толерантний до цього виду проблем, може викликати дубльовані ACK і ретрансміти.

## 2.3 Опис модулів

Використовувати будемо зв'язку інструментів IP SLA і EEM. Обидва інструменти реалізуються в Cisco IOS шляхом написання сценаріїв. У дослідженні високонавантажених мереж були виявлені проблеми, які можуть виникнути в мережі передачі даних: втрата пакетів, висока затримка. Для досягнення поставленої мети модулі повинні запобігати появі цих проблем.

Перший модуль використовує тільки можливості інструменту Embedded Event Manager. Даний модуль повинен запобігти виникненню втрати пакетів, шляхом переналаштування таблиці маршрутизації. Блок – схема алгоритму роботи модуля показано на рис. 2.7.

Згідно даного алгоритму, спочатку перевіряється виникнення втрати пакетів, потім, якщо сталася втрата пакетів, то "скрипт", який працює за заданим алгоритмом генерує повідомлення про втрату пакетів і виводить його в командний рядок маршрутизатора, при цьому додатково можна налаштувати інформування про виникнення проблеми на пошту адміністратору. Далі йде перехід в режим конфігурації маршрутизатора, після чого "скрипт" запускає процес м'якого перезавантаження сусідів. Після перезавантаження сусідів, маршрути на маршрутизаторі будуть перераховані і буде обраний резервний маршрут, по якомупіде потік даних.

Після включення "скрипта", він почне відстежувати стан події по виникненню втрати пакетів. Дана подія використовує наступні параметри:

– interface name – задає інтерфейс для відстеження;

– parameter – визначає ім'я лічильника для моніторингу;

– output_packets_droped – кількість пакетів, відкинутих через повну чергу виводу;

– entry-op – порівнює поточне значення лічильника інтерфейсу зі значенням запису, використовуючи зазначений оператор. Якщо є збіг, подія запускається і моніторинг подій відключається до тих пір, поки не будуть виконані критерії виходу;


Рисунок 2.7 – Блок-схема алгоритму запобігання втрати пакетів

– entry-val – задає значення, з яким порівнюється поточне значення лічильника інтерфейсу, щоб вирішити, чи слід викликати подію інтерфейсу. Діапазон від -2147483648 до 2147483647;

– entry-type – визначає тип операції, яка буде застосована до

ідентифікатора об'єкта, зазначеного в аргументі entry-value,

– increment – інкремент використовує поле вводу / вводу або значення – виходу як інкрементну різницю. Параметр entry-value або exit-value порівнюється з різницею між поточним значенням лічильника і значенням, коли подія була запущена в останній раз (або першим опитаним зразком, якщо це нове подія). Від'ємне значення перевіряє зростаючу різницю для лічильника, котрий зменшується;

– pool-interval – визначає інтервал часу між послідовними опитуваннями. За замовчуванням це 1 секунда.

Використовуючи дані параметри, подія запускається на зазначеному інтерфейсі і починає відстежувати параметр втрати пакетів, при цьому порівнюючи поточне значення зі значенням попереднього спрацювання. Якщо поточне значення буде більше за попереднє, то "скрипт" виконає дії по встановленню нової маршрутизації. На рис. 2.8 показано опис інтерфейсу s2/0 на маршрутизаторі R2 до виникнення втрати пакетів і після.

```
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
```

Рисунок 2.8 – Опис інтерфейсу

Як видно з рисунку, до виникнення втрати пакетів черга виводу дорівнює нулю, так само, як і втрата пакетів. Після зростання навантаження, черга виходу заповнилася, і з'явилася втрата пакетів.

Додатково варто відзначити, що порівняння значень відбувається один раз в секунду (pool-interval 1). Зменшити це значення не має змоги. І хоча з одного боку одна секунда – це багато для мереж передачі даних, з іншого боку, при величезному потоці даних, пакети не будуть просто відкидатися, а підуть по резервному маршруту всього лиш через одну секунду.

Другий модуль працює за алгоритмом, представленим на рис. 2.9.

Другий модуль реалізується за допомогою обох інструментів: IP SLA і EEM. Алгоритм працює наступним чином. За допомогою IP SLA, маршрутизатор відправляє тестові пакети даних, маршрутизатор – відповідач (англ. responder) відповідає на них. Якщо джитер відповідних пакетів більше, ніж встановлене значення, то EEM реагує на це і генерує повідомлення про появу джитера. Далі здійснюється перехід в режим конфігурації пристрою, після чого запускається процес перерахунку метрики. В даному модулі IP SLA використовує такі параметри:

– icmp-jitter – використовуваний тест;

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

– num-packets – кількість тестових пакетів;

– interval – інтервал відправки тестових пакетів;

– threshold – встановлює верхнє порогове значення для розрахунку статистики моніторингу мережі, створеної операцією IP SLA;

– timeout – встановлює час, протягом якого операція IPSLA очікує відповідівід свого пакета запиту;

– frequency – встановлює швидкість, з якою повторюється зазначенаоперація IP SLA.


Рисунок 2.9 – Блок схема алгоритму, що визначає затримку

Додатково для роботи IP SLA необхідно задати цикл роботи і час

початку роботи тесту. Робиться це командою "ipsla schedule" Так само потрібно враховувати, що пристрій, адреса якого вказана в параметрі source-ip, має відповідати на тестові пакети. Для цього на цьому пристрої дається команда "ipslaresponder", після цього пристрій готовий відповідати.

Після того як тест готовий до роботи, потрібно переконатися в правильності роботи. Для цього можна задати команду "show ipsla summary". Приклад виводу цієї команди показаний на рис. 2.10. З нього видно, що тест використовує тип icmp-jitter, відправляє тестові пакети за адресою 192.168.3.1, при цьому затримка складає 8 мілісекунд.

```
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
```

Рисунок 2.10 – Вивід стану тестів IP SLA

Додатково можна подивитися статистику роботи IP SLA, виконавши команду "show ipsla statistics" (рис. 2.11). Тут видно, що ipsla10 відправляє 50 пакетів із середньою затримкою відповіді 8 мілісекунд. Також показана затримка в одну сторону, час джитера, втрата пакетів і інші показники.

Тепер необхідно написати сценарій EEM, який буде реагувати на показники P SLA. Засобами EEM, як і в першому модулі, створюється подія, в якій, використовуючи параметри: ipsla operation-id, reaction-type. Спочатку вказується з яким саме IP SLA буде працювати сценарій і на яку подію буде реагувати. Після цього сценарій відстежує стан зазначеного тесту IP SLA і, отримавши негативний статус (див. рис. 2.10, п'ятий стовпчик), виконує дії відповідно до алгоритму, на рис. 2.9.

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

IPSLA operation id: 10
Type of operation: icmp-jitter
        Latest RTT: 8 milliseconds
Latest operation start time: 16:24:44 KRAT Tue Jun 9 2020
Latest operation return code: OK
RTT Values:
        Number Of RTT: 50               RTT 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
```

Рисунок 2.11 – Статистика IP SLA

## 2.4 Висновки до розділу

У цьому розділі зроблений порівняльний аналіз протоколів динамічної маршрутизації. Зокрема, порівняння алгоритмів побудови метрик маршрутів. З отриманих результатів можна зробити наступний висновок. протокол динамічної маршрутизації EIGRP є найкращим кандидатом для використання у високонавантажених мережах. Даний протокол володіє кращими показниками за такими критеріями як: час збіжності, обсяг службового трафіку і адміністративна дистанція. Далі проведено дослідження високонавантажених мереж, виявлені можливі проблеми, які знижують якість мережі. Також були описані два модуля, спрямовані на запобігання можливих втрат,описаних вище.

Перший модуль використовує інструмент EEM і вирішує проблему з виникненням втрати пакетів. Другий модуль використовує інструменти IP SLA і EEM. В даному модулі IP SLA визначає поточний джитер і, якщо він

стає вище допустимого, то повідомляє про це. Сценарій EEM дивиться на це повідомлення і діє за заданим алгоритмом, за яким виконує дії по запуску процесу перерахунку метрик маршрутів. При спрацьовуванні обох модулів оптимізації, маршрути будуть перераховано з умовою зростаючого навантаження. Через що метрика завантаженого каналу зв'язку стане більшою, ніж метрика резервного каналу зв'язку. Це призведе до того, що використовувати резервний маршрут стане краще.

# 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

## 3.1 Опис лабораторного стенду

Обидва модулі будуть тестуватися на лабораторному стенді, показаному нарис. 3.1.

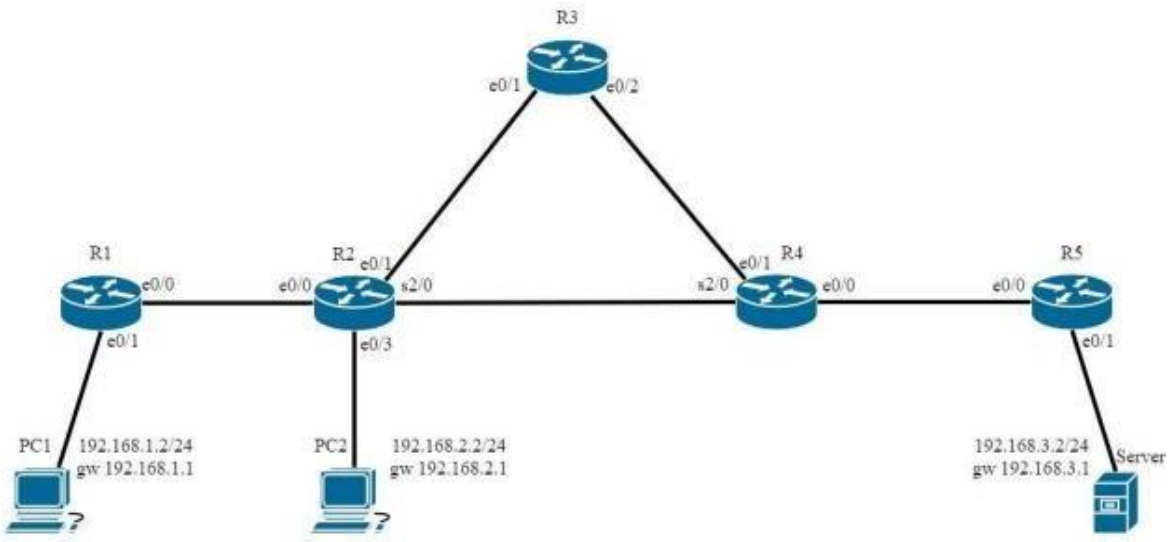


Рисунок 3.1 – Лабораторний стенд

Установка містить п'ять маршрутизаторів Cisco (R1-R5), з версією Cisco IOS 15.5, двох ПК, з ОС Linux (PC1-PC2) і одного сервера (Server). ПК і серверу присвоєні статичні адреси виду 192.168.A.2, де A – номер комп'ютера (сервера) (PC1, PC2,Server3), і шлюз за замовчуванням 192.168.X.1.

Принцип призначення ip адрес наступний. Інтерфейси, "дивляться" в сторону комп'ютерів і сервера, виступають в ролі шлюзу і мають адреси 192.168.A.1, де A – це номер комп'ютера (сервера) (PC1,PC2, Server3). Далі ip адреси на інтерфейсах маршрутизатора призначаються за правилом 10.0.XY.X (Y), где X, Y – номера маршрутизаторів (R1-R5).Наприклад, інтерфейс e0/0 на маршрутизаторі R1 матиме адресу 5510.0.12.1/24, а інтерфейс e0/0 на маршрутизаторі R2 – 10.0.12.2/24. Тут третій октет має

значення 12, так як порядковий номер маршрутизаторів на кінцях лінка – 1 і 2 відповідно. Четвертий октет – це номер маршрутизатора. Мережеві налаштування всіх маршрутизаторів можна побачити в табл. 3.1.

Таблиця 3.1 – Мережеві налаштування маршрутизаторів

| Маршрутизатор | Інтерфейс | IP адреса | Маска |
|---|---|---|---|
| R1 | E0/0 | 10.0.12.1 | 255.255.255.0 |
| | E0/1 | 192.168.1.1 | 255.255.255.0 |
| R2 | E0/0 | 10.0.12.2 | 255.255.255.0 |
| | E0/1 | 10.0.23.1 | 255.255.255.0 |
| | E0/3 | 192.168.2.1 | 255.255.255.0 |
| | S2/0 | 10.0.24.2 | 255.255.255.0 |
| R3 | E0/1 | 10.0.23.3 | 255.255.255.0 |
| | E0/2 | 10.0.34.3 | 255.255.255.0 |
| R4 | E0/0 | 10.0.45.4 | 255.255.255.0 |
| | E0/1 | 10.0.34.4 | 255.255.255.0 |
| | S2/0 | 10.0.24.4 | 255.255.255.0 |
| R5 | E0/0 | 10.0.45.5 | 255.255.255.0 |
| | E0/1 | 192.168.3.1 | 255.255.255.0 |

Тепер необхідно мережеві налаштування з табл. 3.1 прописати на маршрутизаторах. Для прикладу налаштуємо маршрутизатор R2. Для цього необхідно:

– під’єднатися до маршрутизатора, використовуючи консольний кабель;

– командою "enable" перейти в привілейований режим;

– командою "Configuration terminal" перейти в режим конфігурації;

– командою "interface e0/0" перейти в режим конфігурації інтерфейсу e0/ 0;

– командою "no shutdown" ввімкнути інтерфейс;

– командою "ipaddress 10.0.12.2 255.255.255.0" призначити ip адреси на інтерфейс e0 / 0.

На рис. 3.2 можна побачити мережеві налаштування маршрутизатора R2. Інші інтерфейси налаштовуються за аналогією.

```
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
!
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
!
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
```

Рисунок 3.2 – Мережеві налаштування на інтерфейсах маршрутизатора R2

Після налаштування ip адрес кожен маршрутизатор почне бачити своїх "сусідів". Також побудуються таблиці маршрутизації, в яких з'являться маршрути до сусідніх маршрутизаторів. Для забезпечення мережевої доступності комп'ютерів і маршрутизаторів необхідно налаштувати маршрутизацію. У висновку другого розділу було прийнято рішення використовувати EIGRP. Процес його конфігурації виглядає наступним чином:

– під’єднатися до маршрутизатора, використовуючи консольний кабель або SSH-клієнт;

– командою "enable" перейти в привілейований режим;

– командою "Configuration terminal" перейти в режим конфігурації;

– командою "router eigrp 34" перейти в режим конфігурації EIGRP. Тут 34 - це номер автономної системи, узятий в якості прикладу. Цей номер повинен бути однаковим для всіх маршрутизаторів;

– командою "eigrp router-id 2.2.2.2" задати ідентифікатор маршрутизатора. Тут ідентифікатор 2.2.2.2 використовується для прикладу, рекомендовано задавати ідентифікатор рівний loopback адресі маршрутизатора;

– командою "network 192.168.2.0 0.0.0.255" оголошуємо підмережу, для кожної підмережі.

Після конфігурації EIGRP, на всіх маршрутизаторах будуть побудовані таблиці маршрутизації. Наприклад, виконавши команду "Show ip route eigrp" в привілейованому режимі на маршрутизаторі R2, можна побачити побудовану таблицю маршрутизації за допомогою EIGRP (рис. 3.3).

```
D        10.0.34.0/24 [90/2192813] via 10.0.23.3, 00:00:39, Ethernet0/1
D        10.0.45.0/24 [90/2201957] via 10.0.2.4, 00:00:39, Serial2/0
D     192.168.1.0/24 [90/308203] via 10.0.12.1, 00:01:14, Ethernet0/0
D     192.168.3.0/24 [90/2227557] via 10.0.2.4, 00:00:34, Serial2/0
D     192.168.4.0/24 [90/2201957] via 10.0.2.4, 00:00:39, Serial2/0
```

Рисунок 3.3 – Таблиця маршрутизації R2

Перед початком тестування треба зробити ще кілька дій. По-перше, необхідно включити коефіцієнт "k2", що відповідає за навантаження в формулі

2.1. Робиться це командою "metric weights 0 1 1 1 0 0" в режимі конфігурації EIGRP. Тут перший елемент – це tos (type of service), tos =0. Після виконання цієї команди відбудеться перерахунок значень метрик маршрутів, побудованих за

допомогою EIGRP, з урахуванням значення завантаження. По-друге, необхідно скоригувати параметр bandwidth на наступних інтерфейсах: R2-e0 /

1, R3-e0 / 1, R3-e0 / 2, R4-e0 / 1. Командою "bandwidth1200 ", в режимі конфігурації інтерфейсу, зменшуємо теоретичну ширину смуги пропускання. На рис. 3.4 можна побачити значення метрик маршрутів до підмережі 192.168.3.0 на маршрутизаторі R2 до зміни параметра bandwidth і після.

```
P 192.168.3.0/24, 1 successors, FD is 359403, serno 17
        via 10.0.23.3 (359403/333803), Ethernet0/1
        via 10.0.2.4 (2227557/308203), Serial2/0

P 192.168.3.0/24, 1 successors, FD is 359403, serno 21
        via 10.0.2.4 (2227557/308203), Serial2/0
        via 10.0.23.3 (2244013/2218413), Ethernet0/1
```

Рисунок 3.4 – Значення метрик маршрутів

Тепер, коли ір адреси присвоєні інтерфейсам на маршрутизаторах, налаштовані відповідні коефіцієнти для розрахунку метрики, скориговані значення ширини смуг пропускання і побудовані таблиці маршрутизації за допомогою EIGRP, треба дописати в конфігурацію "скрипти", що реалізують модулі по оптимізації маршрутизації. Після внесення "скриптів" в конфігурацію маршрутизаторів, можна переходити до тестування модулів.

### 3.2 Тестування модулів

Тестування буде проводитися на стенді, показаному на рис. 3.1. Перед початком тестування необхідно провести початкові налаштування, зазначені в попередньому розділі. Потім необхідно написати сценарій, який реалізує перший модуль. Для працездатності модуля необхідно, щоб у маршрутизатора було більше одного маршруту, в іншому випадку модуль працювати не буде. З рис. 3.1 видно, що таких маршрутизаторів два: R2, R4. Для тестування будемо використовувати маршрутизатор R2.

1. Підключитися до R2, використовуючи SSH-клієнт, кабель і т.д .
2. Активувати режим конфігурації командами: "enable", "Configure terminal".

3. Командою "event manager session cli username" вказати ім'я користувача,від якого буде виконуватися сценарій.

4. Командою "event manager applet" задати ім'я сценарію.

5. Конфігурувати подію, при виникненні якої буде виконуватися сценарій. Командою "event interface name Serial2/0" вказуємо інтерфейс, за яким буде спостерігати модуль. Далі треба сконфігурувати реакцію на виникнення втрати пакетів на інтерфейсі.

6. Використовуючи команди "action" Label "", де "Label" порядковий номер команди, налаштувати дії, які будуть виконуватися при виникненні втрати пакетів. Дії налаштовуються згідно з алгоритмом на рис. 2.7.

Після конфігурації модуля можна приступати до тестування. Тестування буде складатися з трьох етапів. Перший етап покаже, як поведеться модуль при невисокому навантаження на мережу. Для цього на комп'ютері PC2 запустимо утиліту "ping" з параметром розміру пакета 1400 до Server. На рис. 3.5 видно, що всі пакети були доставлені до адресата і отримана відповідь на них. Середня затримка становить 17,985 ms.

```
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 ttl=61 time=18.0 ms
1408 bytes from 192.168.3.2: icmp_req=89 ttl=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
```

Рисунок 3.5 – Статистика утиліти ping

На маршрутизаторі R2 подивимося статистику інтерфейсу S2 / 0 (рис. 3.6).

```
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
```

Рисунок 3.6 – Статистика інтерфейсу S2 / 0

На рисунку видно, що на даний момент черга порожня, і передається один пакет за секунду. На другому етапі запустимо заздалегідь підготовлений скрипт на PC1, який створює потік даних заданого обсягу. встановимо параметр"Bandwidth" рівний 1400 kb / sec. З огляду на пропускну здатність інтерфейсу S2 / 0 1 544 kb / sec, створюваний потік даних навантажить канал, але не заб'є його повністю. Знову подивимося статистику інтерфейсу S2 / 0 (рис. 3.7). Тут видно, що черга все ще порожня, а кількість пакетів, які виходять, зросла до 123 пакета за секунду. Дане значення є середнім за п'ять хвилин роботи скрипта.

```
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
```

Рисунок 3.7 – Статистика інтерфейсу S2 / 0

На третьому етапі скорегуємо параметр "bandwidth" таким чином, щоб потік даних, котрий відправляється, був більшим, ніж пропускні здатність інтерфейсу. Таким чином пакети почнуть поміщатися в чергу, а після переповнення черги почнуть відкидатися. встановлюємо параметр "Bandwidth" рівним 2000 kb/sec і запустимо скрипт.

Через кілька секунд роботи, на маршрутизаторі R2 можна спостерігати таку ситуацію. Далі, слідуючи алгоритму, відбувається м'який перезапуск сусідів, за рахунок чого йде перерахунок метрик маршрутів. З огляду на створюване навантаження на другому етапі, вартість маршруту через завантажений канал стане більшою, ніж вартість маршруту резервного каналу. Тому потік даних піде в обхід завантаженого інтерфейсу. На рис. 3.8

проілюстрована дана ситуація. Додатково на рисунку видно, що після переналаштування маршрутів затримка утиліти "ping" знизилася до 9,641 ms.

```
*Jun 10 03:11:53.040: %HA_EM-6-LOG: INTERFACE: Serial2/0 packet drop detected
R2#
*Jun 10 03:11:53.246: %DUAL-5-NBRCHANGE: EIGRP-IPv4 34: Neighbor 10.0.2.4 (Seria
l2/0) is resync: manually cleared
*Jun 10 03:11:53.246: %DUAL-5-NBRCHANGE: EIGRP-IPv4 34: Neighbor 10.0.12.1 (Ethe
rnet0/0) is resync: manually cleared
*Jun 10 03:11:53.246: %DUAL-5-NBRCHANGE: EIGRP-IPv4 34: Neighbor 10.0.23.3 (Ethe
rnet0/1) is resync: manually cleared
*Jun 10 03:11:54.041: %HA_EM-6-LOG: INTERFACE: Serial2/0 packet drop detected
*Jun 10 03:11:54.244: %DUAL-5-NBRCHANGE: EIGRP-IPv4 34: Neighbor 10.0.12.1 (Ethe
rnet0/0) is resync: manually cleared
*Jun 10 03:11:54.244: %DUAL-5-NBRCHANGE: EIGRP-IPv4 34: Neighbor 10.0.23.3 (Ethe
rnet0/1) is resync: manually cleared
*Jun 10 03:11:54.245: %DUAL-5-NBRCHANGE: EIGRP-IPv4 34: Neighbor 10.0.2.4 (Seria
l2/0) is down: manually cleared
```

Рисунок 3.8 – Повідомлення про переналаштування маршрутів

```
1408 bytes from 192.168.3.2: icmp_req=54 ttl=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 ttl=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
```

Рисунок 3.9 – Статистика утиліти ping

Додатково упевнитися, що резервний маршрут став основним можна за допомогою команди "show ip eigrp topology all-links". На рис. 3.4 (друга частина) видно, що основний маршрут до підмережі192.168.3.0, до завантаження, йде через інтерфейс s2 / 0 і має вартість 2227557, а резервний 2244013. А на рис. 3.10 видно, що після перерахунку метрик, маршрут йде через інтерфейс E0 / 1 і має вартість нижчу, ніж маршрут через S2 / 0.

```
P 192.168.3.0/24, 1 successors, FD is 2227557, serno 17
        via 10.0.23.3 (2244013/2218413), Ethernet0/1
        via 10.0.2.4 (2257096/308203), Serial2/0
```

Рисунок 3.10 – Основний і резервний маршрути

Таким чином можна стверджувати, що модуль щодо запобігання втрат

пакетів на завантаженому каналі зв'язку працює повністю.

Для тестування другого модуля, так само необхідно провести його конфігурацію. Для цього потрібно:

1. Підключитися до маршрутизатора R2, використовуючи SSH-клієнт,консольний кабель і т.д .

2. Перейти в режим конфігурації командами: "enable", "Configure terminal".

3. Провести конфігурування IPSLA, вказавши тип тесту – "icmp-jitter ", адреса призначення – 192.168.3.1, адреса джерела – 10.0.12.2, кількість пакетів – 50,інтервал – 10.

4. Задати параметри "threshold", "timeout", "frequency" рівними 30,40, 1 відповідно.

5. Встановити правило часу життя і часу старту тесту, командою "Ipsla schedule 10 life forever start-time now".

6. Встановити реакцію на повідомлення про таймаут і кількості повідомлень, після яких реагувати. Команда "ip sla reaction-configuration 10 react timeout threshold-type consecutive 2 ".

7. Задати команду для роботи з EEM "ipsla enable reaction-alerts".

8. Задати команду для виведення повідомлень "ipsla logging traps".

Після конфігурації IP SLA, тест буде запущений і почне збір статистики. Дляреагування на повідомлення IP SLA необхідно налаштувати сценарій EEM. Робиться це наступними командами:

1. Командою "event manager session cli username" вказати ім'я користувача, відякого буде виконуватися сценарій.

2. Командою "event manager applet" задати ім'я сценарію.

3. Командою "event ipsla operation-id 10 reaction-type timeout" вказати, з якимтестом буде працювати сценарій і на що реагувати.

4. Використовуючи команди "action" Label "", де "Label" порядковий номер команди, налаштувати дії, які будуть виконуватися при виникненні втрати пакетів. Дії налаштовуються згідно з алгоритмом на рис. 2.9.

Виглядають дії так:

– "action 001 if $ _ipsla_condition eq" Occurred "". Значення внутрішньої змінної $ _ipsla_condition порівнюється з рядком "Occurred". При виникненні події timeout, генерується повідомлення "Threshold Occurred for timeout", тому якщо значення стане рівним "Occurred", то виконаються 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"-кінець блоку if;

– "action 006 cli command" end "";

– "action 007 cli command" exit "".

Після конфігурації модуля, можна приступати до тестування. Тестування даного модуля буде проходити в два етапи. Так як в модулі здійснюється реакція на виникнення джитера. А джитер буде виникати не завжди. Він може виникнутив разі, якщо ширина смуги пропускання низька, а довжина черги висока. Тому на першому етапі буде показано поведінку модуля при невеликій черзі, а на другомуетапі роботи модуля при збільшенні черги.

Для тестування модуля на першому етапі запустимо заздалегідь підготовлений скрипт. В параметрах зазначимо значення "bandwidth" рівним1544 kb/sec. Дане значення відповідає значенню ширини смуги пропускання інтерфейсу S2/0. Тому канал буде завантажений повністю. Вихідна довжина черги дорівнює 40 пакетам. Додатково на комп'ютері PC2 запустимо утиліту "ping" до Server.

Далі можна спостерігати таку ситуацію. Загальний потік даних, що проходить через інтерфейс S2/0 маршрутизатора R2 – більший, ніж пропускна здатність, тому пакети починають поміщатися в чергу, через що зростає показник утиліти "ping".Середнє значення затримки рівне 407,429 ms.

Далі, командою "show interfaces 2/0" можна побачити значення черги (рис. 3.11).

```
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
```

Рисунок 3.11 – Статистика інтерфейсу s2 / 0 зі стандартним розміром черги

Слідом за статистикою інтерфейсу подивимося статистику тесту IPSLA, командою "show ip sla statistics" (рис. 3.12). Зі статистики видно, що хоча середня затримка дорівнює 446 ms, середнє значення джитера дорівнює 6 ms. А так як в тесті IP SLA порогове значення для спрацьовування дорівнює 30 ms, то алгоритмне спрацьовує.

```
RTT Values:
        Number Of RTT: 0                RTT Min/Avg/Max: 432/450/455 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
```

Рисунок 3.12 – Висновок статистики IP SLA

На другому етапі будемо збільшувати максимально можливу кількість пакетів в черзі. Для зміни довжини черги перейдемо в режим конфігурації інтерфейсу. Потім скористаємося командою "hold-queue 100 out ". Дана команда збільшить виходить довжину черги до 100пакетів.

Як тільки була збільшена довжина черги, модуль відразу ж спрацював, за рахунок чого запустилася перезавантаження сусідів EIGRP, і встановилася нова маршрутизація. Це сталося з наступної причини: збільшення довжини

черги впливає на підвищення джитера, так як більша кількість пакетів поміщається в чергу, а не відкидається, в наслідок чого різниця між доставкою пакетів зростає.

На рис. 3.13 показана статистика інтерфейсу із збільшеною чергою. А на рис. 3.14 видно, що за допомогою модуля було виявлено виникнення джитера більше заданого значення, і відбулося переналаштування маршрутизації.

```
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
```

Рисунок 3.13 – Статистика інтерфейсу S2 / 0 зі збільшеною чергою

```
 09:20:32.723: %RTT-3-IPSLATHRESHOLD: IP SLAs(10): Threshold Occurred for timeout
 09:20:32.945: %DUAL-5-NBRCHANGE: EIGRP-IPv4 34: Neighbor 10.0.2.4 (Serial2/0) is resync
lly cleared
 09:20:32.945: %DUAL-5-NBRCHANGE: EIGRP-IPv4 34: Neighbor 10.0.12.1 (Ethernet0/0) is res
nually cleared
 09:20:32.945: %DUAL-5-NBRCHANGE: EIGRP-IPv4 34: Neighbor 10.0.23.3 (Ethernet0/1) is res
nually cleared
--
 09:20:32.957: %HA_EM-6-LOG: IPSLA10_timeout: Jitter detected
--
 09:20:34.463: %RTT-3-IPSLATHRESHOLD: IP SLAs(10): Threshold Cleared for timeout
carrier transitions     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Рисунок 3.14 – Повідомлення про виникнення джитера і переналаштування маршрутизації

На підтвердження цього можна вивести значення метрик командою "Show ip eigrp topology all-links". На рис. 3.15 видно, що метрика маршруту, йде через інтерфейс E0 / 1 – менша, ніж метрика маршруту через S2 / 0.

```
P 192.168.3.0/24, 1 successors, FD is 2227557, serno 55
        via 10.0.23.3 (2244013/2218413), Ethernet0/1
        via 10.0.2.4 (2247371/308203), Serial2/0
```

Рисунок 3.15 – Основний і резервний маршрути

## 3.3 Висновки до розділу

У цьому розділі було представлено опис лабораторного стенду, на якому розроблялися модулі переналаштування маршрутної інформації при виникненні нештатних ситуації в високонавантажених мережах і, в подальшому, проводилося тестування. Було стисло описані мережеві налаштування, необхідні для мережевої доступності всіх вузлів мережі; налаштований протокол EIGRP, після чого маршрутизатори встановили «сусідські» відносини, обмінялися даними і побудували таблиці маршрутизації.

Після виконання всіх налаштувань, було зроблено тестування обох модулів в кілька етапів. Перший модуль, спрямований на запобігання втрати пакетів, показав свою повну працездатність. У випадку виникнення втрати пакетів, модуль відпрацював згідно з алгоритмом і переналаштував таблицю маршрутизації з умовою збільшеного навантаження. За підсумками тестування другого модуля, був зроблений висновок, що даний модуль буде працювати тільки в мережах, в яких присутні канали з невеликою пропускною спроможністю, при цьому з достатньо довгою чергою. В таких умовах буде виникати досить великий джитер, для того щоб спрацював сценарій щодо його запобігання. На швидких каналах зв'язку черга спустошується набагато швидше, тому значення джитера вкрай мале, порядку 1-7 ms.

# ВИСНОВКИ

З кожним роком навантаження на мережу зростає. З'являються все більше стрімінгових онлайн сервісів, що випускають свій контент у високій якості. Величезна кількість людей працюють в інтернеті, підприємства все більше стали переводити своїх співробітників на віддалену роботу. Все це створює високе навантаження на мережу. Для запобігання цьому стала з'являтися необхідність в модулях переналаштування маршрутної інформації при виникненні нештатних ситуацій в високонавантажених мережах.

В роботі за підсумком вирішення поставлених завдань і досягнення поставленої мети було виконано наступне:

– за результатами проведеного аналізу предметної області виявлено переваги і недоліки існуючих методів оптимізації;

– проведено дослідження високонавантаженої мережі;

– розроблено методику переналаштування маршрутної інформації;

– отримано і протестовано два модуля по оптимізації маршрутизації, їх працездатність була підтверджена в ході тестування.

Однак слід врахувати наступне:

– в мережі повинні використовуватися мінімум 1 маршрутизатор Cisco дляпершого модуля і 2 маршрутизатора для другого;

– у мережі повинна бути надмірність з'єднань (використання модулів в мережі без надлишкових з'єднань не призведе до яких-небудь результатів).

Також був зроблений висновок про те, що переваги від використання другого модуля можна отримати в мережах, в яких є канал зв'язку з низькою пропускною спроможністю і досить великим розміром черги. У такій ситуації джитер буде досить високий, щоб перевищити поріг спрацьовування модуля, внаслідок чого модуль відпрацює згідно алгоритму.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. Improvement of Performance of EIGRP Network by Using a SupervisoryController with Smart Congestion Avoidance Algorithm . Research Gate GmbH.

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

3. Adaptive load balancing with OSPF. Research Gate GmbH. URL: https://www.researchgate.net/publication/228787806_Adaptive_load_balancing_with_ OSPF.

4. 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.

5. RFC 2328, OSPF Version 2. The Internet Engineering Task Force (IETF). URL: https://datatracker.ietf.org/doc/rfc2328/ (дата звертання: 28.10.2021).

6. RFC 1142, OSI IS-IS Intra-domain Routing Protocol. URL: https://tools.ietf.org/html/rfc1142 (дата звертання: 14.11.2021).

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

8. IPSLAs Configuration Guide . Официальная инструкция настройки IPSLA на сетевых устройствах Cisco. URL: https:/-www.cisco.com/c/en/us/td/docs/ios- xml/ios/ipsla/configuration/xe-16/sla-xe-16-book.html

9. Embedded Event Manager Configuration Guide. Официальная инструкция настройки IPSLA на сетевых устройствах Cisco. URL: https://www.cisco.com /c/en/us/td/docs/ios-xml/ios/eem/configuration/12-4t/eem-12-4t-book.html

10. Nadeau Thomas D. SDN: Software Defined Networks. Sebastopol: O`ReillyMedia Inc., 2013. 384 c.

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