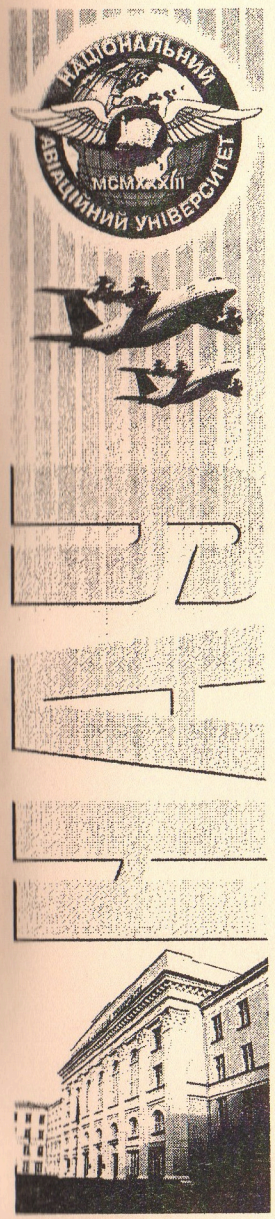


VIVERE!
VINCERE!
CREARE!

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний авіаційний університет

МЕРЕЖНЕ ПРОГРАМУВАННЯ ТА ІНТЕРФЕЙСИ ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

ЛАБОРАТОРНИЙ ПРАКТИКУМ
для здобувачів вищої освіти
ОС «Бакалавр» спеціальності 172
«Телекомунікації та радіотехніка»

Київ 2024

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
Національний авіаційний університет

МЕРЕЖНЕ ПРОГРАМУВАННЯ ТА ІНТЕРФЕЙСИ ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

ЛАБОРАТОРНИЙ ПРАКТИКУМ
для здобувачів вищої освіти
ОС «Бакалавр» спеціальності 172
«Телекомунікації та радіотехніка»

Київ 2024

УДК 621.39:004.451.84(076.5)
М 52

Укладачі: *О. Ю. Лавриненко* — канд. техн. наук;
В. В. Антонов — канд. техн. наук;
В. Є. Курушкін — канд. техн. наук

Рецензент *Ю. В. Мельник* — д-р техн. наук, проф.,
завідувач кафедри аерокосмічних систем управління
(Національний авіаційний університет)

*Затверджено Науково-методично-редакційною радою
Національного авіаційного університету
(протокол № 2/24 від 16.02.2024 р.).*

М 52 **Мережне програмування та інтерфейси телекомунікаційних систем:** лабораторний практикум / уклад. : О. Ю. Лавриненко, В. В. Антонов, В. Є. Курушкін. — К. : НАУ, 2024. — 64 с.

Наведено практичні вказівки до виконання лабораторних робіт за допомогою комп'ютерного моделювання в графічному симуляторі мереж Graphical Network Simulator 3 (GNS3) та мережевого обладнання Cisco, використовуючи допоміжні команди з інтерфейсу командного рядка (CLI) IOS Cisco з основними теоретичними відомостями та запитаннями для самоконтролю. Лабораторні роботи спрямовано на здобування знань та набуття навичок, необхідних для мережного програмування та конфігурації інтерфейсів мережевого обладнання, які є невід'ємною частиною у побудові та адмініструванні комп'ютерних та телекомунікаційних мереж.

Розроблений відповідно до програми дисципліни «Мережне програмування та інтерфейси телекомунікаційних систем», яка викладається для здобувачів вищої освіти бакалаврату на четвертому курсі спеціальності 172 «Телекомунікації та радіотехніка» під час навчального процесу, а також даний практикум може бути корисним для викладачів та фахівців з радіотехніки й телекомунікацій.

ВСТУП

Мета лабораторних робіт – закріплення здобувачами вищої освіти теоретичних знань та набуття ними практичних навичок у мережному програмуванні та конфігурації інтерфейсів мережевого обладнання, які є невід'ємною частиною у побудові та адмініструванні комп'ютерних та телекомунікаційних мереж, що являє собою базу в підготовці фахівців спеціальності 172 «Телекомунікації та радіотехніка».

У процесі виконання робіт здобувачі вищої освіти повинні дослідити особливості практичного застосування:

– мережевого протоколу динамічного налаштування хоста DHCP;

– протоколу дозволу адреси ARP;

– списків контролю доступу ACL до інтерфейсу, в якому виконується перетворення мережевих адрес NAT;

– протоколу динамічної маршрутизації RIP;

– протоколу маршрутизації кадрів у мережі Frame Relay;

– технології віртуальної маршрутизації та переадресації VRF;

– побудови тунелів на маршрутизаторах Cisco з використанням інтерфейсів Loopback;

– технології розподілу навантаження серверів SLB;

за допомогою комп'ютерного моделювання в графічному симуляторі мереж Graphical Network Simulator 3 (GNS3) та мережевого

СПИСОК ЛІТЕРАТУРИ

1. Буров Є.В. Комп'ютерні мережі: підручник. Львів: «Магнолія 2006», 2021. 262 с.

2. Микитишин А.Г., Митник М.М., Стухляк П.Д. Телекомунікаційні системи та мережі: навч. посіб. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017. 384 с.

3. Єфіменко А.А. Основи побудови локальних комп'ютерних мереж Ethernet на базі керованих комутаторів компанії Cisco: навч. посіб. Житомир: Житомирська політехніка, 2021. 116 с.

4. Коробейнікова Т.І., Захарченко С.М. Технології захисту локальних мереж на основі обладнання Cisco: навч. посіб. Львів: Видавництво Львівської політехніки, 2021. 232 с.

5. Жураковський Б.Ю., Зенів І.О. Комп'ютерні мережі: навч. посіб. К.: КПІ ім. Ігоря Сікорського, 2020. 366 с.

6. Тарнавський Ю.А., Кузьменко І.М. Організація комп'ютерних мереж: підручник. К.: КПІ ім. Ігоря Сікорського, 2018. 259 с.

7. Борисенко В.Д., Устенко С.А., Устенко І.В. Основи комп'ютерного моделювання в інженерній діяльності: навч. посіб. Миколаїв: МНУ, 2016. 276 с.

ЗМІСТ

Навчальне видання

МЕРЕЖНЕ ПРОГРАМУВАННЯ ТА ІНТЕРФЕЙСИ ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

ЛАБОРАТОРНИЙ ПРАКТИКУМ
для здобувачів вищої освіти
ОС «Бакалавр» спеціальності 172
«Телекомунікації та радіотехніка»

Укладачі:
ЛАВРИНЕНКО Олександр Юрійович
АНТОНОВ Веніамін Валерійович
КУРУШКІН Віталій Євгенович

В авторській редакції

Технічний редактор *А. І. Лавринович*
Комп'ютерна верстка *Н. В. Чорної*

Підп. до друку 28.02.2024. Формат 60x84/16. Папір офс.
Опер. друк. Ум. друк. арк. 3,72. Обл.-вид. арк. 4,0.
Тираж 25 прим. Замовлення № 36-1.

Видавець і виготівник
Національний авіаційний університет
03058. Київ-58, проспект Любомира Гузара, 1.

Свідоцтво про внесення до Державного реєстру ДК № 7604 від 15.02.2022.

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
Національний авіаційний університет

МЕРЕЖНЕ ПРОГРАМУВАННЯ ТА ІНТЕРФЕЙСИ ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

ЛАБОРАТОРНИЙ ПРАКТИКУМ
для здобувачів вищої освіти
ОС «Бакалавр» спеціальності 172
«Телекомунікації та радіотехніка»

Київ 2024

УДК 621.396.2 (076.5)
С 408

Укладачі: *О. Ю. Лавриненко* — канд. техн. наук, доцент кафедри ТКРС;
В. В. Антонов — канд. техн. наук, доцент кафедри ТКРС;
В. Є. Курушкін — канд. техн. наук, доцент кафедри ТКРС;

Рецензент *Ю. В. Мельник* — д-р техн. наук, проф. завідувач кафедри аерокосмічних систем управління (Національний авіаційний університет)

Затверджено Науково-методично-редакційною радою Національного авіаційного університету (протокол № ____ від _____ р.).

Мережне програмування та інтерфейси телекомунікаційних систем: лабораторний практикум / уклад. : О. Ю. Лавриненко, В. В. Антонов, В. Є. Курушкін. — К. : НАУ, 2024. — 64 с.

Наведено практичні вказівки до виконання лабораторних робіт за допомогою комп'ютерного моделювання в графічному симуляторі мереж Graphical Network Simulator 3 (GNS3) та мережевого обладнання Cisco, використовуючи допоміжні команди з інтерфейсу командного рядка (CLI) IOS Cisco з основними теоретичними відомостями та запитаннями для самоконтролю. Лабораторні роботи спрямовано на здобування знань та набуття навичок, необхідних для мережного програмування та конфігурації інтерфейсів мережевого обладнання, які є невід'ємною частиною у побудові та адмініструванні комп'ютерних та телекомунікаційних мереж.

Практикум розроблений відповідно до програми дисципліни «Мережне програмування та інтерфейси телекомунікаційних систем», яка викладається для здобувачів вищої освіти бакалаврату на четвертому курсі спеціальності 172 «Телекомунікації та радіотехніка» під час навчального процесу, а також даний практикум може бути корисним для викладачів та фахівців з радіотехніки й телекомунікацій.

ВСТУП

Мета лабораторних робіт – закріплення здобувачами вищої освіти теоретичних знань та набуття ними практичних навичок у мережному програмуванні та конфігурації інтерфейсів мережевого обладнання, які є невід'ємною частиною у побудові та адмініструванні комп'ютерних та телекомунікаційних мереж, що являє собою базу в підготовці фахівців спеціальності 172 «Телекомунікації та радіотехніка».

У процесі виконання робіт здобувачі вищої освіти повинні дослідити особливості практичного застосування:

- мережевого протоколу динамічного налаштування хоста DHCP;
- протоколу дозволу адреси ARP;
- списків контролю доступу ACL до інтерфейсу, в якому виконується перетворення мережевих адрес NAT;
- протоколу динамічної маршрутизації RIP;
- протоколу маршрутизації кадрів у мережі Frame Relay;
- технології віртуальної маршрутизації та переадресації VRF;
- побудови тунелів на маршрутизаторах Cisco з використанням інтерфейсів Loopback;
- технології розподілу навантаження серверів SLB;

за допомогою комп'ютерного моделювання в графічному симуляторі мереж Graphical Network Simulator 3 (GNS3) та мережевого обладнання Cisco, використовуючи допоміжні команди з інтерфейсу командного рядка (CLI) IOS Cisco.

Опис кожної лабораторної роботи супроводжується викладенням основних теоретичних відомостей досліджуваних процесів та послідовності виконання практичної частини робіт у програмному середовищі GNS3 та на мережевому обладнанні Cisco, після виконання яких потрібно оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу. Для більш поглибленого розуміння теоретичних та практичних аспектів досліджуваних процесів можна використовувати джерела, які вказані у списку літератури.

Лабораторна робота 1

ПРОТОКОЛ ДИНАМІЧНОГО НАЛАШТУВАННЯ ХОСТА DHCP

Мета роботи: дослідити використання мережевого протоколу динамічного налаштування хоста DHCP, що дає змогу мережевим пристроям автоматично отримувати IP-адресу та інші параметри, необхідні для роботи в мережі TCP/IP.

Основні теоретичні відомості

Протокол динамічного налаштування хоста (Dynamic Host Configuration Protocol, DHCP) належить до числа основних служб, що формують інфраструктуру мереж. Він застосовується для автоматичного виконання конфігурації мережевих параметрів. Будь-який фахівець, який займається побудовою та обслуговуванням комп'ютерних мереж, а також роботою в них, повинен мати принаймні загальне уявлення про цей протокол і розуміти, на якому рівні працює DHCP. Структурна схема використання протоколу DHCP зображена на рис. 1.1.

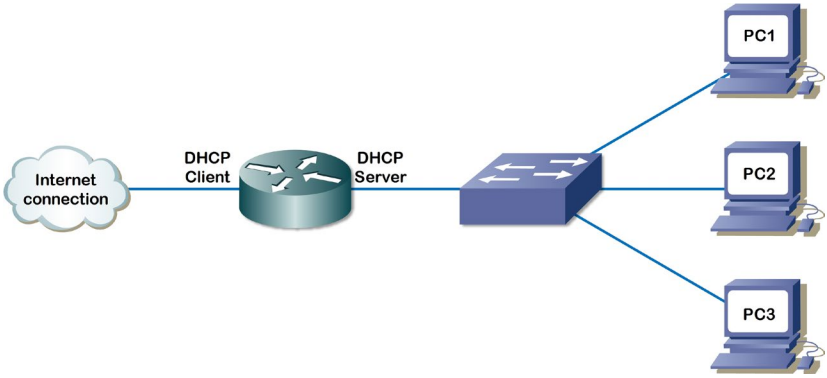


Рис. 1.1. Структурна схема використання протоколу DHCP

За наявності хоча б середніх навичок налаштування протоколу DHCP на комп'ютері не представляє складності і займає близько хвилини. Однак, якщо потрібно налаштувати велику кількість пристроїв, які до того ж можуть бути територіально віддалені один від одного, вручну з цим завданням не впоратися. Тому управління налаштуваннями в корпоративних мережах забезпечують DHCP-сервери. З їхньою допомогою досягається автоматизація налашту-

вань. Знаючи, як звернутися до протоколу DHCP, можна один раз налаштувати такий сервер, після чого подальше налаштування і встановлення параметрів на пристроях здійснюється автоматично. Крім того, сервер забезпечує централізоване управління наданими IP-адресами, що надаються, унеможливлює їхнє дублювання та оперативно звільняє адреси, які не використовуються.

Розберемо загальний принцип роботи протоколу DHCP за ключовими моментами.

Робота протоколу DHCP здійснюється за принципом клієнт-сервер. Для отримання налаштувань використовується схема DORA (Discover-Offer-Request-Acknowledge). Сам процес складається з таких етапів (рис. 1.2):

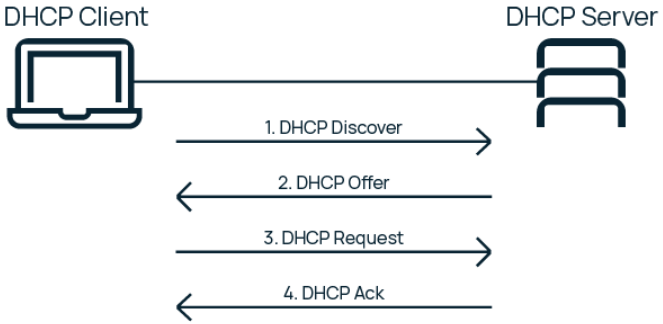


Рис. 1.2. Структурна схема функціонування протоколу DHCP

1. Виявлення (Discover). Після підключення клієнта починається процес його ініціалізації в мережі. Він знаходить відповідний DHCP-сервер шляхом надсилання спеціального запиту DHCPDISCOVER на адресу 255.255.255.255. З огляду на відсутність власного IP, у такому запиті вказується 0.0.0.0 і MAC. Запит надходить на всі ПК у відповідному сегменті мережі. При цьому відповідь на нього автоматично надсилається тільки DHCP-серверами.

2. Пропозиція (Offer). Отримавши від клієнта запит, DHCP-сервер здійснює його обробку і виконує підбір мережевої конфігурації. Ця конфігурація надсилається клієнту у зворотному повідомленні DHCPOFFER, яке, як правило, передається на вказаний MAC. Однак у деяких випадках застосовується широкомовлення. У разі знаходження декількох серверів у межах мережі клієнту надходить

відповідна кількість DHCPOFFER, з яких він вибирає один (зазвичай перший за часом отримання).

3. Запит (Request). Після отримання DHCPOFFER клієнт передає серверу спеціальне повідомлення DHCPREQUEST, яке містить запит налаштувань. У цьому запиті дублюється інформація з DHCPDISCOVER, а також вказується IP-адреса обраного на попередньому етапі DHCP-сервера.

4. Підтвердження (Acknowledge). Після отримання DHCPREQUEST обраний DHCP-сервер виконує фіксацію відповідного прив'язування для клієнта і направляє йому у відповідь повідомлення DHCPACK. У ньому підтверджуються надані автоматично налаштування. Це повідомлення передається на адресу MAC клієнта, яка була вказана на попередньому етапі. Отримавши DHCPACK, клієнт проводить автоматичну перевірку наданих налаштувань і застосовує конфігурацію мережі, отриману від сервера.

Отримана адреса може бути перевірена клієнтом шляхом надсилання широкомовного запиту ARP. У разі виявлення використання наданого IP іншим пристроєм, серверу передається повідомлення DHCPDECLINE. Після цього починається повторна ініціалізація.

Однією з важливих функцій, для чого призначений протокол DHCP, можна назвати функцію оновлення мережевих адрес. IP надається клієнту на встановлений період, який називається часом оренди. Його тривалість залежно від встановлених налаштувань сервера варіюється в межах від хвилин до місяців. Після завершення половини цього періоду клієнт робить спробу оновити оренду. У разі невдачі робляться повторні спроби поновлення, які будуть слідувати до завершення терміну. Якщо жодна спроба не завершиться успішним оновленням, клієнт розпочне пошук нового сервера.

У процесі поновлення оренди клієнт проходить два стани:

1. RENEWING – оновлення адреси.
2. REBINDING – оновлення конфігурації.

Настання стану RENEWING передбачено на половині часу оренди (T1), а стан REBINDING – після проходження 87,5 цього періоду (T2). Щоб унеможливити синхронізацію різних клієнтів,

використовується випадкова величина відхилення під час визначення T1 і T2.

Розглянемо, як працює протокол DHCP за обох станів.

У цьому стані клієнт запускає процес оновлення оренди. Для цього він направляє запит DHCPREQUEST на власний DHCP-сервер. У разі згоди сервера на продовження клієнту повертається відповідь DHCPACK, у якій прописано новий час оренди та оновлені параметри. Клієнт відзначає отримані значення, скидає відлік часу T1 і T2, після чого переходить у нормальний робочий стан.

У разі відмови сервер повертає повідомлення DHCPNACK. У результаті клієнт знову починає ініціалізацію і повторно запускає процедуру оновлення часу оренди.

У разі неотримання відповіді на запит, спрямований для оновлення оренди, клієнт очікує її протягом певного часу. Після цього серверу надсилається повторний запит. Клієнт продовжує підтримувати стан RENEWING і періодично відправляє запити DHCPREQUEST до отримання відповіді з боку сервера. Весь цей період він нормально працює на своїй поточній IP-адресі.

Якщо до настання T2 не надходить серверна відповідь, мережевий протокол DHCP передбачає переведення клієнта в стан REBINDING, після чого за допомогою широкомовлення надсилається запит DHCPREQUEST із зазначенням поточної адреси. Такі запити надсилаються через певний час.

Послідовність виконання роботи

1. За допомогою відрізка кросоверного кабелю під'єднайте комп'ютер до одного з Ethernet-портів маршрутизатора Cisco 1605 або іншого, наявного.

2. З конфігураційного режиму створіть новий пул протоколу DHCP за допомогою команди **ip dhcp pool *name***, де ***name*** – ім'я новостворюваного пулу.

3. У режимі конфігурації новоствореного пулу DHCP вкажіть за допомогою команди **network**

192.168.1.0 /24 адреси, які будуть роздаватися клієнтам.

4. Сконфігуруйте адреси серверів DNS, що віддаються по DHCP, для чого використовуйте команду **dns server *192.168.1.2 192.168.1.3***.

5. Аналогічно до попереднього пункту налаштуйте адреси серверів WINS командою **netbios-name- server *192.168.1.4 192.168.1.5***.

6. Задайте IP-адресу шлюзу за замовчуванням для клієнтів викликом **default-router *192.168.1.1***.

7. Встановіть час оренди адреси клієнтом, для чого виконайте команду **lease *0 0 15***.

8. Вийдіть із режиму конфігурування пулу DHCP командою **exit**.

9. Налаштуйте мережеву карту комп'ютера на автоматичне отримання параметрів IP.

10. Звільніть отриману раніше адресу викликом **ipconfig /release**.

11. Отримайте нові параметри за допомогою **ipconfig /renew**.

12. Переконайтеся в тому, що ви отримали конфігурацію протоколу IP.

13. Перегляньте на маршрутизаторі таблицю виданих IP-адрес командою привілейованого режиму командою **show ip dhcp binding**.

14. Виключіть адреси *192.168.1.2* і *192.168.1.3* з пулу адрес, що видаються, для чого використовуйте команду режиму конфігурації **ip dhcp excluded-address 192.168.*1.2 192.168.1.3***. Якщо комп'ютеру видано будь-яку іншу адресу, виключіть і її з пулу.

15. Відмовтеся від використовуваної раніше адреси на лабораторному комп'ютері та запросіть її знову.

16. Переконайтеся в тому, що тепер мережевій карті комп'ютера присвоєно іншу адресу.

17. Сконфігуруйте статичну прив'язку MAC-адреси комп'ютера до IP-адреси, що видається сервером DHCP. Для цього створіть новий пул DHCP, усередині цього пулу дайте команду **host *address mask***, де ***address*** і ***mask*** – IP-адреса і мережева маска конфігурованого хоста. Виконайте команду **client-identifier 01*aa.bbcc.ddee.ff***, де ***aabbccddeeff*** – MAC-адреса лабораторного комп'ютера, а **01** - вказує на тип середовища Ethernet. Командою **client-name *test*** вкажіть ім'я клієнту, що налаштовується.

18. Відмовтеся на комп'ютері від поточної адреси й отримайте її знову. Переконайтеся в тому, що отримана адреса - саме та, яку щойно було жорстко сконфігуровано.

Запитання для самоконтролю

1. Дайте визначення DHCP.
2. Що собою являє DHCP-сервер, у чому його функції?
3. Дайте визначення DHCP-клієнта.
4. У чому переваги використання DHCP?
5. Для чого необхідне налаштування DHCP у мережі?
6. Дайте визначення динамічному розподілу адрес у мережі.
7. Дайте визначення автоматичному розподілу адрес у мережі.
8. Дайте визначення ручному розподілу адрес у мережі.
9. На чому заснована робота DHCP-сервера?
10. Який порядок роботи DHCP-сервера?
11. Перелічіть види запитів сервера.

Лабораторна робота 2

ПРОТОКОЛ ДОЗВОЛУ АДРЕСИ ARP

Мета роботи: дослідити принципи роботи протоколу дозволу адреси ARP, отримати практичні навички в аналізі кадрів ARP та роботі з генератором пакетів Packit.

Основні теоретичні відомості

Для визначення локальної адреси за IP-адресою використовується протокол дозволу адреси (Address Resolution Protocol, ARP). Протокол ARP працює по-різному залежно від того, який протокол канального рівня працює в даній мережі - протокол локальної мережі (Ethernet, Token Ring, FDDI) з можливістю широкомовного доступу одночасно до всіх вузлів мережі, або ж протокол глобальної мережі (X.25, frame relay), який, зазвичай, не підтримує широкомовний доступ. Існує також протокол, що розв'язує зворотне завдання – знаходження IP-адреси за відомою локальною адресою. Він називається реверсивний ARP – RARP (Reverse Address Resolution Protocol) і використовується під час старту бездискових станцій, які не знають у початковий момент своєї IP-адреси, але знають адресу свого мережевого адаптера.

У локальних мережах протокол ARP використовує широкомовні кадри протоколу канального рівня для пошуку в мережі вузла із заданою IP-адресою.

Вузол, якому потрібно виконати відображення IP-адреси на локальну адресу, формує ARP запит, вкладає його в кадр протоколу канального рівня, вказуючи в ньому відому IP-адресу, і розсилає запит широкомовно. Усі вузли локальної мережі отримують ARP запит і порівнюють зазначену там IP-адресу з власною. У разі їхнього збігу вузол формує ARP-відповідь, у якій зазначає свою IP-адресу і свою локальну адресу, і надсилає її вже надіслано, оскільки в ARP-запиті відправник вказує свою локальну адресу. ARP-запити і відповіді використовують один і той самий формат пакета. Оскільки локальні адреси можуть у різних типах мереж мати різну довжину, то формат пакета протоколу ARP залежить від типу мережі.

У полі типу мережі для мереж Ethernet вказується значення 1. Поле типу протоколу дає змогу використовувати пакети ARP не тільки для протоколу IP, а й для інших мережевих протоколів. Для IP значення цього поля дорівнює 080016.

Довжина локальної адреси для протоколу Ethernet дорівнює 6 байтам, а довжина IP-адреси – 4 байтам. У полі операції для ARP запитів вказується значення 1 для протоколу ARP і 2 для протоколу RARP.

Вузол, що відправляє ARP-запит, заповнює в пакеті всі поля, крім поля шуканої локальної адреси (для RARP-запиту не вказується шукана IP-адреса). Значення цього поля заповнюється вузлом, який упізнав свою IP-адресу.

У глобальних мережах адміністратору мережі найчастіше доводиться вручну формувати ARP-таблиці, в яких він задає, наприклад, відповідність IP-адреси адресі вузла мережі X.25, яка має сенс локальної адреси. Останнім часом намітилася тенденція автоматизації роботи протоколу ARP і в глобальних мережах. Для цієї мети серед усіх маршрутизаторів, під'єднаних до будь-якої глобальної мережі, виділяється спеціальний маршрутизатор, який веде ARP-таблицю для всіх інших вузлів і маршрутизаторів цієї мережі. За такого централізованого підходу для всіх вузлів і маршрутизаторів вручну потрібно задати тільки IP-адресу і локальну адресу виділеного маршрутизатора. Потім кожен вузол і маршрутизатор реєструє свої адреси у виділеному маршрутизаторі, а за необхідності встановлення відповідності між IP-адресою і локальною адре-

сою вузол звертається до виділеного маршрутизатора із запитом і автоматично отримує відповідь без участі адміністратора.

Розглянемо приклад роботи протоколу.

Уявіть комп'ютери A і B, розташовані в офісі, об'єднані в локальну мережу один з одним за допомогою комутатора. У цій мережі не використовуються проміжні шлюзи або роутери. Комп'ютер A хоче надіслати дані комп'ютеру B. За допомогою DNS він визначає, що комп'ютеру B відповідає IP-адреса 192.168.0.55. Для передачі даних йому також необхідна MAC-адреса комп'ютера B. Спочатку комп'ютер A використовує ARP-таблицю з кеша для пошуку MAC-адреси для 192.168.0.55. Якщо запис із MAC-адресою було знайдено, то на MAC-адресу 00:eb:24:b2:05:ac передається IP-пакет, інкапсульований у фрейм канального рівня моделі OSI. Якщо ж запис у кеші не було знайдено, то комп'ютер A відправляє широкомовне ARP-повідомлення із запитом інформації про IP-адресу 192.168.0.55. У відповідь комп'ютер B відправляє пакет зі своєю IP і MAC-адресою. Після виконання цієї послідовності починається передача даних.

Послідовність виконання роботи

1. За допомогою відрізка витої пари п'ятої категорії, конекторів і обтискного інструменту зробіть два кросоверні кабелі.
2. Підключіться до маршрутизатора за допомогою консольного порту.
3. На його Ethernet-інтерфейсах вкажіть адреси 192.168.1.1/24 і 192.168.2.1/24.
4. На обох інтерфейсах Ethernet дайте команду **no ip proxy-arp**.
5. Увімкніть ці інтерфейси за допомогою команди **no shutdown** у режимі конфігурування інтерфейсу.
6. Сконфігуруйте комп'ютери таким чином

Таблиця 2.1

Таблиця маршрутизації

Параметр	Комп'ютер №1	Комп'ютер №2
IP-адреса	192.168.1.2	192.168.2.2
Маска підмережі	255.255.0.0	255.255.0.0
IP-адреса шлюзу	192.168.1.1	192.168.2.1

7. Підключіть комп'ютери за допомогою приготованих раніше кабелів до портів маршрутизатора.

8. Переконайтеся в тому, що комп'ютери отримують відповіді на ехо-запити протоколу ICMP від своїх шлюзів.

9. Переконайтеся в тому, що з комп'ютера №1 не можна отримати доступ до комп'ютера №2 (також за допомогою протоколу ICMP).

10. Скористайтеся сніфером, а також командою **arp -a** операційної системи комп'ютера, щоб установити точну картину подій, що відбуваються в мережі. Розкажіть про спостереження викладачеві.

11. Для обох Ethernet-інтерфейсів маршрутизатора дайте команду **ip proxy-arp**.

12. Подивіться, що зміниться в цьому випадку в доступності одного з комп'ютерів з іншого.

13. Скористайтеся сніфером, а також командами **arp -a** і **arp -d** операційної системи комп'ютера, щоб встановити точну картину подій, що відбуваються в мережі. Розкажіть про спостереження викладачеві.

Запитання для самоконтролю

1. Що таке протокол ARP? Для чого він використовується?

2. Скільки ARP-таблиць має комп'ютер? Маршрутизатор? Комутатор?

3. Протокол ARP функціонально можна розділити на клієнтську і серверну частини. Опишіть, які функції ви віднесли б до клієнтської частини, а які – до серверної?

4. Які адреси і з якою метою заносить адміністратор в ARP-таблицю?

5. У яких випадках корисно використовувати протокол Proxy-ARP?

6. Як вивести на екран ARP-таблицю? Як додати запис до ARP-таблиці?

7. За допомогою якої команди можна видалити запис з ARP-таблиці? Як видалити всю ARP-таблицю?

8. Хто отримує запит ARP і хто на нього відповідає?

9. Яка інформація міститься в ARP запиті?

10. Як надіслати ARP запит?

Лабораторна робота 3

СПИСКИ КОНТРОЛЮ ДОСТУПУ ACL ТА ПЕРЕТВОРЕННЯ МЕРЕЖЕВИХ АДРЕС NAT

Мета роботи: дослідити процес налаштування і застосування ACL-списків до інтерфейсу, в якому виконується перетворення мережевих адрес NAT та спостереження ефекту розміщення ACL-списку під час використання NAT.

Основні теоретичні відомості

Списки контролю доступу (Access Control List, ACL) – це набір текстових виразів, які щось дозволяють, або щось забороняють. Зазвичай ACL дозволяє або забороняє IP-пакети, але крім усього іншого він може заглядати всередину IP-пакета, переглядати тип пакета, TCP і UDP порти. Також ACL існує для різних мережевих протоколів (IP, IPX, AppleTalk тощо). Здебільшого застосування списків доступу розглядають з погляду пакетної фільтрації, тобто пакетна фільтрація необхідна в тих ситуаціях, коли у вас стоїть обладнання на кордоні Інтернету і вашої приватної мережі і треба відфільтрувати непотрібний трафік. Ви розміщуєте ACL на вхідному напрямку і блокуєте надлишкові види трафіку.

Основна мета ACL – підвищити безпеку шляхом обмеження або дозволу доступу до різних компонентів системи: файли, каталоги, пристрої, мережеві порти і служби. ACL забезпечують детальний контроль над мережевим трафіком, даючи змогу мережевим адміністраторам ефективно керувати і захищати свою мережу.

ACL дають змогу адміністраторам регулювати доступ користувачів і груп до конфіденційної інформації, запобігати несанкціонованим змінам, знижувати ризик несанкціонованого доступу неавторизованих користувачів або зловмисників.

ACL працюють послідовно, пакети перевіряються на відповідність кожному правилу в списку, поки не буде знайдено збіг. Якщо збіг не знайдено, застосовується дія за замовчуванням.

Правила ACL можуть бути налаштовані на дозвіл або заборону певного трафіку, або на пріоритет певних типів трафіку над іншими. ACL визначає, які операції з файлами, програмами або процесами дозволено або заборонено виконувати користувачеві або групі.

Існує чотири основні типи ACL.

1. Стандартні ACL засновані виключно на IP-адресі джерела трафіку. Вони простіші та менш деталізовані, але можуть бути корисними в певних сценаріях, де потрібна фільтрація тільки IP-адреси джерела. До того ж, вони використовують менше обчислювальних потужностей.

2. Розширені списки забезпечують більшу гнучкість, даючи змогу створювати правила на основі IP-адрес джерела і призначення, протоколів, номерів портів та інших параметрів. Вони забезпечують детальний контроль над фільтрацією трафіку, що робить їх придатними для складних мережевих середовищ. Розширені списки керування доступом дають змогу фільтрувати трафік, підтримуваний протоколами IP, TCP, ICMP, UDP.

3. Рефлексивні ACL проводять фільтрацію трафіку за допомогою даних сеансу верхнього рівня. Ці списки працюють за допомогою TCP-запитів і відповідей. Після отримання відповіді формується ACL, що розпізнає згенеровані з локальної мережі параметри сесії користувача. За допомогою цих параметрів приймається рішення про доступ.

4. Динамічні ACL дають системним адміністраторам можливості надати користувачеві тимчасовий доступ або обмежити доступ до маршрутизатора з інтернету. Динамічні списки використовують для розширених ACL, Telnet і аутентифікації.

Впровадження ACL включає в себе такі кроки:

1. Визначте завдання і цілі впровадження ACL. Позначте конкретні мережеві ресурси, які потребують захисту, і визначте бажані політики контролю доступу.

2. Проаналізуйте схеми мережевого трафіку для виявлення потенційних вразливостей безпеки та областей, які потребують контролю доступу. Цей аналіз допоможе у визначенні відповідних правил ACL.

3. Створіть правила ACL на основі виявлених вимог. Розгляньте тип ACL і визначте необхідні параметри.

4. Протестуйте правила ACL у контрольованому середовищі, перш ніж розгортати їх у виробничій мережі. Переконайтеся, що правила працюють так, як задумано, і не спричиняють жодних непередбачених наслідків або збоїв.

Щоб було трохи зрозуміліше, наведемо простий приклад. Спираючись на списки доступу, працює Policy-Based Routing (PBR).

Можна зробити тут так, щоб пакети, які надходять із мережі 192.168.1.0/24, відправлялися на next-hop 10.0.1.1, а з мережі 192.168.2.0/24 на 10.0.2.1 (зауважимо, що звичайна маршрутизація спирається на адресу призначення пакета й автоматично всі пакети відправляються на один next-hop).

NAT (Network Address Translation) – технологія перетворення приватних IP-адрес у зовнішні в IPv4. Завдяки цьому процесу ваша віртуальна машина отримує доступ до Інтернету.

У приватній мережі використовуються приватні (сірі) IP-адреси, які не використовуються в Інтернеті. Групою проектування Інтернету 1994 року було обрано такі підмережі (що залишається актуальним і донині):

У результаті для внутрішнього застосування було зарезервовано три блоки IP-адрес:

10.0.0.0 - 10.255.255.255.255/8 (16777216 хостів)

172.16.0.0 - 172.31.255.255/12 (1048576 хостів)

192.168.0.0 - 192.168.255.255.255/16 (65536 хостів)

Приватні IP-адреси, також звані внутрішніми, внутрішньомережевими, локальними або сірими, будь-яка організація має право використовувати на власний розсуд без будь-якої реєстрації в будь-якої організації.

Щоб виходити в Інтернет потрібен білий IP, який буде "маскувати" 1 або кілька приватних IP-адрес.

Механізм NAT якраз здійснює підміну (або "маскування") сірих адрес на білі і навпаки.

Таким чином, уся приватна мережа може підключатися до Інтернету через одну публічну IP-адресу (або пул адрес), надану провайдером. У результаті, ресурс глобальних адрес витрачається набагато економніше.

Перетворення NAT має важливу особливість з погляду забезпечення безпеки: трансляція приватних IP-адрес у публічні з пулу маршрутизатора дає змогу приховати топологію внутрішньої мережі від зовнішніх користувачів, що ускладнює несанкціонований доступ до ресурсів мережі..

Послідовність виконання роботи

1. За допомогою консольного порту налаштуйте маршрутизатор так, щоб на нього можна було зайти віддалено за протоколом telnet.

На маршрутизатор можна потрапити через консольний інтерфейс за протоколом RS-232. Для цього можна використовувати програму PuTTy або HyperTerminal. Налаштування RS-232 протоколу: bits per second 9600, data bits 8, parity none, stop bits 1. Для того щоб можна було зайти на маршрутизатор cisco за протоколом telnet, необхідно виконати наведені нижче пункти.

а) Налаштувати IP-адресу на інтерфейсі Ethernet 0 (у режимі конфігурації інтерфейсу).

ip address *ip_address mask*

б) Увімкнути інтерфейс (у режимі конфігурації інтерфейсу).

no shu

exit

в) Налаштувати пароль для входу через telnet (у режимі конфігурації).

line vty 0 4

pass *password*

exit

г) Налаштувати пароль для входу в привілейований режим (у режимі конфігурації).

enable secret *password*

2. Встановіть telnet-сесію.

Після встановлення telnet-сесії, якщо були правильно виконані попередні налаштування, маршуртизатор вимагатиме введення пароля. Треба ввести пароль, який був сконфігурований на line vty.

а) Далі для входу в привілейований режим введіть команду.

enable

б) Введіть пароль, який був сконфігурований раніше.

3. Налаштуйте логічний інтерфейс із довільною ip адресою і "хостовою" маскою (у підмережі може бути тільки одна IP-адреса).

Для цього в режимі конфігурації виконуються такі команди.

interface loopback 0

ip address *ip_address mask*

exit

4. Вийдіть із telnet-сесії – послідовно команди.

exit

exit

5. Спробуйте "потрапити телнетом" на логічний інтерфейс маршрутизатора.

Чому отримали такий результат? Зробіть так, щоб команда виконувалася успішно.

6. Створіть список доступу, що пропускає тільки пакети з source ip адреса вашого комп'ютера.

Для цього можна використовувати так званий "стандартний" список доступу. Для створення такого списку в режимі конфігурації виконуються такі команди.

access-list *number* permit | deny {any} | {host ip_adrress_host} | {ip_address_host} | {ip_address_network invert_mask}

number – номер списку доступу (1-99);

permit – дозвіл;

deny – заборона;

ip_adrress_host – IP адреса хоста;

ip_address_network – ідентифікатор мережі;

invert_mask – інвертована маска;

| – або;

{} – групує команди.

Ці команди вводяться послідовно, і обробляються процесором послідовно зверху вниз. Якщо в кінці немає явного permit any, то всі пакети, для яких не знайшлося відповідності в списку, знищуються.

Приклад (дозволяє пакети з хостів **1.1.1.1**, **2.2.2.2.2**, забороняє з сітки **3.0.0.0 255.0.0.0.0**, дозволяє всі інші, **log** – увімкнення "журналювання" для цього рядка).

access-list 1 permit 1.1.1.1

access-list 1 permit host 2.2.2.2

access-list 1 deny 3.0.0.0 0.0.0.255

access-list 1 permit any log

7. Поставити список доступу на інтерфейс.

Список доступу може бути поставлений на **in** або на **out**. Для маршрутизатора **in** – це те, що входить у маршрутизатор, **out** – те, що виходить із нього.

Команда в режимі конфігурування інтерфейсу:

ip access-group number in | out

number – номер списку доступу.

Якщо все зроблено правильно – сесія telnet не повинна переривати. Якщо перервалася – студент має використати консоль, щоб

знайти помилку і виправити. Для цього використовуються, наприклад, такі діагностичні команди в привілейованому режимі.

sh run
sh access-list number
sh inter e 0
sh ip inter e 0
term mon
term nomon
deb ip packet
no deb all

Якщо не вийшло зрозуміти, у чому помилка або незрозуміла діагностика – зверніться до викладача. Після виконання – покажіть викладачеві.

8. Студент створює список доступу, що дає змогу робити тільки telnet, тільки на interface loopback 0 (сконфігурований раніше) і тільки з його комп'ютера. При цьому передбачається, що telnet сесія встановлена на інтерфейс loopback 0.

Для цього використовується список доступу, званий "розширеним". Такий список доступу дає змогу під час фільтрації використовувати не тільки IP-адресу джерела, а й IP-адресу одержувача та інформацію четвертого рівня – номери TCP/UDP портів, прапори протоколу TCP. Для розширеного списку доступу використовуються номери від 100 до 199 або можна створювати іменовані списки доступу. Приклад (у режимі конфігурації) наведено нижче.

ip access-list extend 100
permit ip 1.1.1.1 0.0.0.0 192.168.5.0 0.0.0.255
permit tcp host 2.2.2.2 host 192.168.5.1 0.0.0.0 eq 80
deny ip any host 192.168.5.1 eq 139
deny udp any any eq rip
permit ip any any log

Під час написання команди користуйтеся командою "?".

9. Поставте цей список доступу на interface e 0 на in. Якщо все зроблено правильно, telnet сесія не повинна бути перервана. Якщо перервалася – користуйтеся консоллю для виявлення та виправлення помилки.

а) Перевірте, що з іншою ip адресою telnet "не проходить".
б) Перевірте, що ping "не проходить".
в) Перевірте, що telnet на Ethernet 0 не проходить.

г) Покажіть викладачеві.

10. Усі списки доступу видаляються.

Усі сконфігуровані рядки видаляються введенням тієї ж самої команди з префіксом "no".

11. Студент дізнається у викладача адресу комутатора і паролі. Заходить на нього за допомогою telnet.

Входить у привілейований режим (пароль – у викладача).

12. Налаштуйте інший Ethernet порт (Ethernet1) маршрутизатора.

Він має бути в тій самій мережі, що й виділений (див. "попередні налаштування") комп'ютер (дізнайтеся у викладача).

13. Підключіть цей порт маршрутизатора до комутатора. Досягніть того, щоб команда ping з маршрутизатора на IP-адресу виділеного комп'ютера виконалася успішно.

а) На комутаторі потрібно правильно налаштувати порт.

б) Запропонуйте алгоритм дій. Якщо алгоритм правильний, викладач підкаже команди, які необхідно ввести.

в) Після виконання цього пункту ми повинні отримати таку топологію мережі. Два інтерфейси маршрутизатора дивляться в різні локальні сегменти. В одному сегменті розташований комп'ютер студента (і вся локальна мережа), в іншому - комп'ютер викладача.

14. Налаштуйте статичний NAT таким чином, щоб ви могли виконати telnet-з'єднання з відкритим TCP-портом на "виділеному" комп'ютері (комп'ютері викладача), таким чином, щоб на "виділений" комп'ютер приходили пакети з IP-адресою джерела – IP-адресою логічного інтерфейсу маршрутизатора.

а) Для цього треба визначити, який локальний сегмент вважати внутрішнім, який зовнішнім. На тому інтерфейсі, який "дивиться" у внутрішній сегмент, треба прописати команду.

ip nat inside

б) На тому інтерфейсі, який "дивиться" у зовнішній сегмент, треба прописати команду.

ip nat outside

в) Для налаштування статичного NAT треба виконати команду.

ip nat inside source static inside_local_address inside_global_address

Користуйтеся командою "?".

15. Досягніть результату. На комп'ютері викладача продемонструйте, що пакети справді приходять з IP-адреси loopback.

Під час виконання цього пункту вам, можливо, доведеться змінювати деякі налаштування і на комп'ютері викладача.

16. Налаштуйте PAT таким чином, щоб уся локальна мережа класу мала доступ через telnet до відкритого TCP-порту "виділеного" комп'ютера.

а) Спочатку видаліть команду статичного NAT.

б) Створіть список доступу, в якому вкажіть, які IP-адреси дозволено "патити".

в) У режимі конфігурації введіть наведену нижче команду.

ip nat inside source list number interface loopback 0 overload

17. Продемонструйте викладачеві, що все працює правильно.

Запитання для самоконтролю

1. Що означає ACL?
2. Для чого потрібен ACL?
3. На якому рівні OSI працює ACL?
4. Як працює Access-list?
5. Як редагувати ACL?
6. Що визначають профілі доступу ACL?
7. Що фільтрують розширені ACL?
8. Які файлові системи підтримують ACL?
9. Як працюють дзеркальні ACL?
10. У якому порядку обробляються правила ACL?
11. Яка з команд access list дозволяє передавання пакетів будь-якому вебклієнту від усіх вебсерверів мережі, IP-адреси яких починаються з октетів 172.16.5?
12. Для яких полів може бути проведено порівняння з використанням іменованого розширеного списку управління доступом IP, але не нумерованого розширеного списку управління доступом IP?

Лабораторна робота 4

ПРОТОКОЛ ДИНАМІЧНОЇ МАРШРУТИЗАЦІЇ RIP

Мета роботи: дослідити процес конфігурування маршрутизаторів для роботи з протоколом динамічної маршрутизації RIP.

Основні теоретичні відомості

Протокол динамической маршрутизации (Routing information Protocol, RIP) – протокол дистанційно-векторної маршрутизації, що

використовує для знаходження оптимального шляху алгоритм Беллмана-Форда. Алгоритм маршрутизації RIP - один із найпростіших протоколів маршрутизації. Кожні 30 секунд він передає в мережу свою таблицю маршрутизації. Основна відмінність протоколів у тому, що RIPv2 (на відміну від RIPv1) може працювати за мультикастом, тобто, розсилаючись на мультикаст-адресу. Максимальна кількість "хопів" (кроків до місця призначення), дозволена в RIP1, дорівнює 15 (метрика 15). Обмеження в 15 хопів не дає застосовувати RIP у великих мережах, тому протокол найпоширеніший у невеликих комп'ютерних мережах. Друга версія протоколу - протокол RIP2 - була розроблена 1994 року і є поліпшеною версією першого. У цьому протоколі підвищено безпеку за рахунок введення додаткової маршрутної інформації. Принцип дистанційно-векторного протоколу: кожен маршрутизатор, що використовує протокол RIP, періодично широкомовно розсилає своїм сусідам спеціальний пакет-вектор, який містить відстані (вимірюються в метриці) від цього маршрутизатора до всіх відомих йому мереж. Маршрутизатор, який отримав такий вектор, нарощує компоненти вектора на величину відстані від себе до цього сусіда і доповнює вектор інформацією про відомі безпосередньо йому самому мережі або мережі, про які йому повідомили інші маршрутизатори. Доповнений вектор маршрутизатор розсилає всім своїм сусідам. Маршрутизатор обирає з кількох альтернативних маршрутів маршрут із найменшим значенням метрики, а маршрутизатор, який передав інформацію про такий маршрут, позначається як наступний (next hop). Протокол непридатний для роботи у великих мережах, оскільки засмічує мережу інтенсивним трафіком, а вузли мережі оперують тільки векторами-відстаней, не маючи точної інформації про стан каналів і топології мережі.

Основні принципи роботи протоколу RIP:

1. Маршрути зберігаються в таблиці маршрутизації кожного мережевого пристрою, на якому працює протокол RIP.
2. RIP протокол обмінюється інформацією про маршрути з сусідніми пристроями за допомогою RIP-повідомлень.
3. Інформація про маршрути містить перелік мереж, їхні метрики (вартість досягнення мереж) та адреси мережевих пристроїв, через які досягається ця мережа.

4. Метрика в RIP протоколі вимірюється в кількості переходів (хопів) до мережі. Максимальне значення метрики - 15. Якщо метрика перевищує це значення, вважається, що маршрут недоступний (немає зв'язку).

5. Мережевий пристрій на основі інформації про маршрути оновлює свою таблицю маршрутизації та ухвалює рішення про передачу пакетів даних.

Основні переваги RIP у мережах:

1. Простота налаштування і використання: Протокол RIP має простий і зрозумілий механізм налаштування, що робить його зручним для використання навіть недосвідченими адміністраторами. Усе, що потрібно зробити, це ввімкнути RIP на маршрутизаторах і налаштувати їх на обмін інформацією про маршрути.

2. Автоматичне виявлення мереж: RIP автоматично виявляє під'єднані мережі та вносить їх у таблицю маршрутизації. Це спрощує процес налаштування маршрутизаторів і дає змогу швидко додавати або видаляти мережі з мережевої інфраструктури.

3. Автоматична адаптація до змін: RIP автоматично оновлює інформацію про маршрути в разі зміни топології мережі. Якщо відбувається збій у роботі маршрутизатора або вимкнення мережевого інтерфейсу, RIP буде автоматично оновлювати таблицю маршрутизації, щоб забезпечити ефективне маршрутизування трафіку.

4. Метрика маршрутів: RIP використовує метрику для визначення вартості маршрутів. Метрика є числовим значенням, яке представляє вартість проходження через певний маршрут. Це дає змогу RIP обирати найефективніші маршрути для передавання даних.

5. Підтримка динамічного оновлення маршрутів: RIP підтримує динамічне оновлення маршрутів, що означає, що в разі зміни мережевої топології та автоматичної передачі інформації між маршрутизаторами таблиця маршрутизації буде автоматично оновлена. Це підвищує стійкість мережі та дає змогу уникнути ручного оновлення маршрутів у разі зміни мережевої інфраструктури.

6. Підтримка класових мереж: RIP підтримує класові мережі, що дає змогу групувати мережі в підмережі. Такий підхід полегшує управління і маршрутизацію трафіку в мережі.

7. Зниження витрат на обслуговування: Використання RIP у мережі дає змогу знизити витрати на обслуговування мережевої

інфраструктури. Завдяки автоматичному налаштуванню та оновленню маршрутів, адміністратори мережі можуть зосередитися на інших завданнях, пов'язаних із розвитком і масштабуванням мережі.

8. Широка підтримка: RIP є одним із найпоширеніших і широко підтримуваних протоколів маршрутизації в мережах. Це означає, що він може бути використаний з різними пристроями та операційними системами, що спрощує інтеграцію в уже наявну мережеву інфраструктуру.

Послідовність виконання роботи

1. Побудуйте мережу, представлену на рис. 4.1.

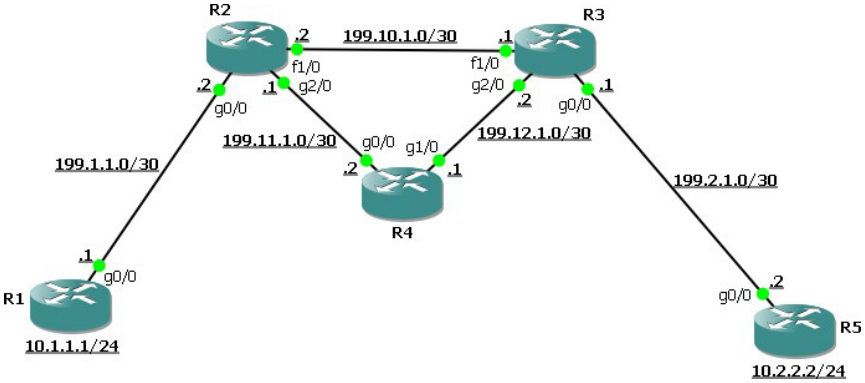


Рис. 4.1. Структурна схема RIP мережі

2. Для початку налаштуємо всі необхідні інтерфейси. На R1 це loopback, що моделює мережу клієнта, і інтерфейс у бік провайдера.

R1(config)# int lo1
R1(config-if)#ip addr 10.1.1.1 255.255.255.0
R1(config-if)# int g0/0
R1(config-if)#ip addr 199.1.1.1 255.255.255.252
R1(config-if)# no shutdown

3. Аналогічно на R5.

R2(config)# int lo1
R2(config-if)#ip addr 10.2.2.2 255.255.255.0
R2(config-if)# int g0/0
R2(config-if)#ip addr 199.2.1.2 255.255.255.252
R2(config-if)# no shutdown

4. На R2 – інтерфейси GigabitEthernet і один інтерфейс FastEthernet.

R2(config)# int g0/0
R2(config-if)#ip addr 199.1.1.2 255.255.255.252
R2(config-if)# no shutdown
R2(config-if)# int g2/0
R2(config-if)#ip addr 199.11.1.1 255.255.255.252
R2(config-if)# no shutdown
R2(config-if)# int f1/0
R2(config-if)#ip addr 199.10.1.2 255.255.255.252
R2(config-if)# no shutdown

5. Аналогічно на R3 і на R4.

R3(config)# int g0/0
R3(config-if)#ip addr 199.2.1.1 255.255.255.252
R3(config-if)# no shutdown
R3(config-if)# int g2/0
R3(config-if)#ip addr 199.12.1.2 255.255.255.252
R3(config-if)# no shutdown
R3(config-if)# int f1/0
R3(config-if)#ip addr 199.10.1.1 255.255.255.252
R3(config-if)# no shutdown
R4(config)# int g0/0
R4(config-if)#ip addr 199.11.1.2 255.255.255.252
R4(config-if)# no shutdown
R4(config-if)# int g1/0
R4(config-if)#ip addr 199.12.1.1 255.255.255.252
R4(config-if)# no shutdown

6. Наразі локальні мережі клієнта 10.1.1.1 і 10.2.2.2 не бачать одна одну (відсутня відповідна маршрутна інформація). Перейдемо до налаштування RIP. На кожному маршрутизаторі необхідно ввести такі команди: у режимі глобальної конфігурації ***router rip***, щоб перейти в режим конфігурації протоколу; ***network*** *адреса_мережі*, щоб увімкнути протокол на потрібних інтерфейсах. Необхідно пам'ятати, що в якості адреси мережі команди ***network*** потрібно вказувати тільки адресу класової мережі. Крім того, введемо команду ***no auto-summary*** для відключення підсумовування мереж на кордоні маршрутизації.

7. Приклад налаштування маршрутизатора R1.

R1(config)# router rip

R1(config-router)# network 199.1.1.0
R1(config-router)# redistribute connected
R1(config-router)# no auto-summary

Якби в другому рядку ми вказали як мережу адресу *10.1.1.0*, система не видала б жодних помилок і попереджень, проте, протокол RIP увімкнувся б на всіх інтерфейсах, підмережі яких входили б у мережу 10.0.0.0/8. За допомогою команди ***redistribute connected*** ми додали мережі всіх під'єднаних інтерфейсів (зокрема й мережу інтерфейсу loopback) до бази даних протоколу RIP – RIP DataBase (RDB), але не включили сам протокол на цих інтерфейсах. Таким чином, інформація про цю мережу розсилається протоколом через усі інтерфейси, на яких функціонує RIP (у нашому випадку через gi0/0), але водночас решта інтерфейсів не розсилає RIP-пакети і не слухає мережу на предмет вхідних RIP-повідомлень.

8. Приклад налаштування маршрутизатора R2.

R2(config)# router rip
R2(config-router)# network 199.10.1.0
R2(config-router)# network 199.1.1.0
R2(config-router)# network 199.11.1.0
R2(config-router)# no auto-summary

9. Налаштування маршрутизатора R3.

R3(config)# router rip
R3(config-router)# network 199.10.1.0
R3(config-router)# network 199.2.1.0
R3(config-router)# network 199.12.1.0
R3(config-router)# no auto-summary

10. Налаштування маршрутизатора R4.

R4(config)# router rip
R4(config-router)# network 199.12.1.0
R4(config-router)# network 199.11.1.0
R4(config-router)# no auto-summary

11. Налаштування маршрутизатора R5.

R5(config)# router rip
R5(config-router)# network 199.2.1.0
R5(config-router)# redistribute connected
R5(config-router)# no auto-summary

Якщо на даному етапі налаштування спробувати виконати команду ***ping 10.2.2.2 source 10.1.1.1*** с R1, то маршрутизатор повідомить про

недоступність вузла з адресою 10.2.2.2. Річ у тім, що за замовчуванням запускається RIPv1, який підтримує тільки з класові мережі. Тобто в нашому випадку в базу даних протоколу буде додано тільки одну мережу замість двох: 10.0.0.0/8 замість 10.1.1.0/24 і 10.2.2.0/24, тому що RIPv1 не враховує маски цих мереж. Переконатися в цьому можна шляхом перегляду таблиці маршрутизації та RDB на R2 і R3, яким маршрутизатори R1 і R5 повідомляють тільки про мережу 10.0.0.0/8. Виправте проблему, що виникла, прописавши команду ***version 2*** в режимі конфігурування протоколу маршрутизації на всіх пристроях.

На цьому налаштування пристроїв завершено, перейдемо безпосередньо до тестування.

12. За допомогою команд ***ping 10.2.2.2 source 10.1.1.1*** і ***trace 10.2.2.2 source 10.1.1.1***, виконаних із маршрутизатора R1, переконайтеся, що локальні мережі клієнта мають доступ одна до одної.

13. Проаналізуйте маршрут, яким слідують пакети між двома мережами, зазначеними в попередньому пункті.

14. Вимкніть низькошвидкісний канал між маршрутизаторами R2 і R3. Як зміниться маршрут проходження пакетів між мережами?

15. Використовуючи команду ***show ip protocols***, перевірте налаштування RIP на кожному маршрутизаторі.

16. Введіть команду ***show ip route rip*** і проаналізуйте її вивід.

17. Виконайте перехоплення трафіку між маршрутизаторами і проаналізуйте повідомлення RIP.

18. За допомогою перехоплення з попереднього пункту продемонструйте роботу методу розщеплення горизонту в RIP.

У налаштованій вище схемі трафік передається через канал FastEthernet, що може бути неефективно через меншу порівняно з GigabitEthernet пропускну здатність. У цьому пункті ми налаштуємо так званий "плаваючий" маршрут, який допоможе вирішити цю проблему.

19. Для початку необхідно відключити RIP на інтерфейсах fa1/0 на R2 і R3.

R2(config)# router rip
R2(config-router)# no network 199.10.1.0
R3(config)# router rip
R3(config-router)# no network 199.10.1.0

20. Тепер налаштуємо статичні маршрути в бік мереж на інтерфейсах loopback, але з адміністративною дистанцією, що дорівнює 130.

R2(config)# ip route 10.2.2.0 255.255.255.0 199.10.1.1 130

R3(config)# ip route 10.1.1.0 255.255.255.0 199.10.1.2 130

Додамо ці маршрути в RIP.

R2(config)# router rip

R2(config-router)# redistribute static

R3(config)# router rip

R3(config-router)# redistribute static

21. За допомогою команди ***trace 10.2.2.2 source 10.1.1.1***, виконаної з маршрутизатора R1, переконаємося, що пакети йдуть через R4. Крім того, подивимося таблицю маршрутизації на R2 за допомогою команди ***show ip route*** і переконаємося, що статичного маршруту в ній немає.

22. Тепер вимкнемо інтерфейси в бік R4.

R2(config)# int g2/0

R2(config-if)# shutdown

R3(config)# int g2/0

R3(config-if)# shutdown

23. Знову подивимося таблицю маршрутизації і переконаємося, що статичний маршрут з'явився в таблиці маршрутизації. Повторимо за допомогою команди ***trace 10.2.2.2 source 10.1.1.1*** з R1, що зв'язність мережі не порушено.

Запитання для самоконтролю

1. Який математичний алгоритм використовується у протоколі RIP?

2. До якого класу належить протокол маршрутизації RIP?

3. До якого виду шлюзових протоколів належить протокол маршрутизації RIP?

4. Яка метрика використовується у протоколі RIP і які її обмеження?

5. Наведіть перелік основних часових параметрів протоколу RIP та їх значення за замовчуванням.

6. Який транспортний протокол використовується для пересилки оновлень у протоколі маршрутизації RIP та яка (які) адреса для цього використовується?

7. Зазначте значення адміністративної відстані для протоколу маршрутизації RIP.

8. Наведіть перелік основних версій протоколу RIP та зазначте їх відмітності.

9. Наведіть перелік методів боротьби з хибними маршрутами у протоколі RIP.

10. Які основні параметри повинні бути зазначені у таблиці маршрутизації маршрутизатора, що працює за протоколом RIP?

11. Основні етапи налагодження роботи протоколу маршрутизації RIP на маршрутизаторах Cisco.

12. Основні команди налагодження протоколу маршрутизації RIP на маршрутизаторах Cisco.

13. Додаткові команди налагодження протоколу маршрутизації RIP на маршрутизаторах Cisco.

14. Основні команди діагностики роботи протоколу маршрутизації RIP на маршрутизаторах Cisco.

15. Способи зменшення об'єму службового трафіка у протоколі RIP.

Лабораторна робота 5

ПРОТОКОЛ МАРШРУТИЗАЦІЇ КАДРІВ У МЕРЕЖІ FRAME RELAY

Мета роботи: дослідити процес конфігурування обладнання для роботи з канальним протоколом Frame Relay, виконати налаштування мережі з реальними FR-комутаторами, а також використати один з маршрутизаторів як комутатор Frame Relay.

Основні теоретичні відомості

Frame relay – це протокол, що застосовує IP-адресу кадру (ідентифікатор каналу передавання даних) для маршрутизації кадрів у мережі та управління маршрутом віртуального з'єднання.

Мережі Frame Relay набагато краще підходять для передачі пульсуючого трафіку комп'ютерних мереж порівняно з мережами X.25. Щоправда, ця перевага проявляється тільки тоді, коли лінії зв'язку наближаються за якістю до ліній зв'язку локальних мереж, а для глобальних ліній таку якість зазвичай можна досягти тільки за використання волоконно-оптичних кабелів.

Технологія Frame Relay була спочатку стандартизована комітетом CCITT (ITU-T) як одна зі служб мереж ISDN. Технологія ISDN є першим широкомасштабним проектом зі створення всесвітньої універсальної мережі, що надає всі основні види послуг телефонних мереж і мереж передачі даних. На жаль, цей амбітний проєкт не досяг поставленої мети, і сьогодні мережі нового покоління будуються вже на основі інших технологій, зокрема IP. Водночас під час реалізації проєкту було досягнуто кілька хоча і не таких глобальних, але, тим не менш, дуже важливих цілей. До них можна зарахувати і створення технології Frame Relay, яка сьогодні є вже незалежною від ISDN технологією.

У рекомендаціях I.122, що вийшли друком 1988 року, послуги з передачі даних входили до переліку додаткових послуг пакетного режиму ISDN. Під час перегляду цих рекомендацій у 1992-93 рр. з'явилися стандарти на дві нові послуги: Frame Relay і Frame Switching. Різниця між ними полягає в тому, що Frame Switching забезпечує гарантовану доставку кадрів, а Frame Relay - доставку за можливістю.

Проста і водночас ефективна для волоконно-оптичних ліній зв'язку технологія Frame Relay відразу привернула увагу провідних телекомунікаційних компаній та організацій зі стандартизації. У її становленні та стандартизації, крім CCITT (ITU-T), активну участь брали форум із ретрансляції кадрів (Frame Relay Forum, FRF) і комітет T1S1 інституту ANSI. Технологія ж Frame Switching так і залишилася всього лише стандартом, який ніколи не мав широкого поширення.

Стандарти Frame Relay, підготовлені і ITU-T/ANSI, і FRF, визначають два типи віртуальних каналів - постійні (PVC) і комутовані (SVC). Це відповідає потребам користувачів, оскільки для з'єднань, якими трафік передають майже завжди, більше підходять постійні канали, а для з'єднань, які потрібні тільки кілька годин на місяць – комутовані. Однак виробники обладнання Frame Relay і постачальники послуг мереж Frame Relay почали з підтримки тільки постійних віртуальних каналів. Це, природно, значно збіднило технологію. Обладнання, що підтримує комутовані віртуальні канали, з'явилося на ринку з великою затримкою. Саме тому технологія Frame Relay часто асоціюється тільки з постійними віртуальними каналами.

Стек протоколів Frame Relay влаштований значно простіше, ніж стек технології X.25. Розробники технології Frame Relay, з огляду на високу якість каналів зв'язку на оптичному волокні, що з'явилися наприкінці 80-х років, вирішили, що можна не включати в протоколи стека функції забезпечення надійності. Якщо ж, незважаючи на малу ймовірність такої події, помилка все ж таки трапляється, то технологія Frame Relay ігнорує цю ситуацію, залишаючи роботу з відновлення загублених або спотворених кадрів протоколам верхніх рівнів, таким як TCP.

Саме завдяки низькій протокольній надмірності, технологія Frame Relay забезпечує високу пропускну здатність і невеликі часи затримки кадрів.

Паралельно було розроблено технологію Frame Switching, яка, як і X.25, забезпечує надійне передавання кадрів на канальному рівні та може застосовуватися в тих випадках, коли канали володіють недостатньо високим рівнем якості або ж до канального рівня з якихось причин висуваються вимоги надійного передавання кадрів. На практиці технологія Frame Switching не знайшла свого застосування, але оскільки стек протоколів Frame Relay створювався з урахуванням існування технології Frame Switching, ми далі все ж коротко зупинимося на ній

На канальному рівні мереж Frame Relay працює протокол LAP-F (Link Access Procedure for Frame mode bearer services), званий у рекомендаціях ITU-T абревіатурою Q.922. Існує дві версії цього протоколу:

1. Протокол LAP-F core є тією "робочою конячкою", яка працює у всіх мережах Frame Relay. Цей протокол забезпечує мінімум засобів, що дозволяють побудувати мережу Frame Relay. Щоправда, в цьому випадку мережа надаватиме тільки послуги постійних віртуальних каналів.
2. Протокол LAP-F control, що забезпечує відновлення кадрів за алгоритмом ковзного вікна, необхідний для того, щоб мережа надавала послуги Frame Switching (комутації кадрів).

Обидва протоколи (LAP-F core і LAP-F control) належать до протоколів канального рівня, забезпечуючи передачу кадрів між двома сусідніми комутаторами.

На фізичному рівні мережа Frame Relay може використовувати лінії зв'язку технології PDH/SDH або ISDN.

Тепер розглянемо шар управління, що виконує функції встановлення динамічно комутованих каналів SVC. Комутатори мережі повинні підтримувати два протоколи шару управління - на канальному рівні LAP-D (який називається також Q.921) і Q.933 на мережевому. Протокол LAP-D у мережах Frame Relay забезпечує надійну передачу сигнальних кадрів між сусідніми комутаторами.

Протокол Q.933 використовує адреси кінцевих вузлів, між якими встановлюється віртуальний канал. Ці адреси зазвичай задаються у форматі телефонних адрес, що відповідають стандарту E.164. Адреса складається з 15 десяткових цифр, які діляться, як і звичайні телефонні номери, на поля коду країни (від 1 до 3 цифр), коду міста і номера абонента. До адреси додається до 40 цифр підадреси, які потрібні для нумерації термінальних пристроїв, якщо в одного абонента їх кілька.

Протокол автоматичного складання таблиць маршрутизації для технології Frame Relay не визначено, тому може використовуватися фірмовий протокол виробника обладнання, або ж таблиці можуть складатися вручну.

Основна перевага Frame Relay порівняно з X.25 полягає в такому. У той час як у мережах X.25 після встановлення з'єднання користувацькі дані передаються протоколами канального і мережевого рівнів, у мережах Frame Relay після встановлення віртуального з'єднання дані передаються тільки за допомогою протоколу канального рівня, що значно знижує накладні витрати.

Технологію Frame Relay найчастіше відносять до технологій канального рівня, ставлячи на перше місце процедури передавання користувацьких даних і опускаючи процедури встановлення віртуального каналу, які виконуються із залученням протоколу мережевого рівня. Віртуальними каналами Frame Relay можуть передаватися дані різних протоколів. Специфікація RFC 1490 визначає методи інкапсуляції в кадри Frame Relay пакетів мережевих протоколів, як-от IP і IPX, протоколів локальних мереж, наприклад, Ethernet, а також протоколу SNA.

Послідовність виконання роботи

1. Побудуйте мережу, представлену на рис. 5.1. Зазначені червоним адреси мереж мають бути налаштовані на інтерфейсах Loopback 1, вони призначені для емуляції локальних мереж, підключених до роутерів.

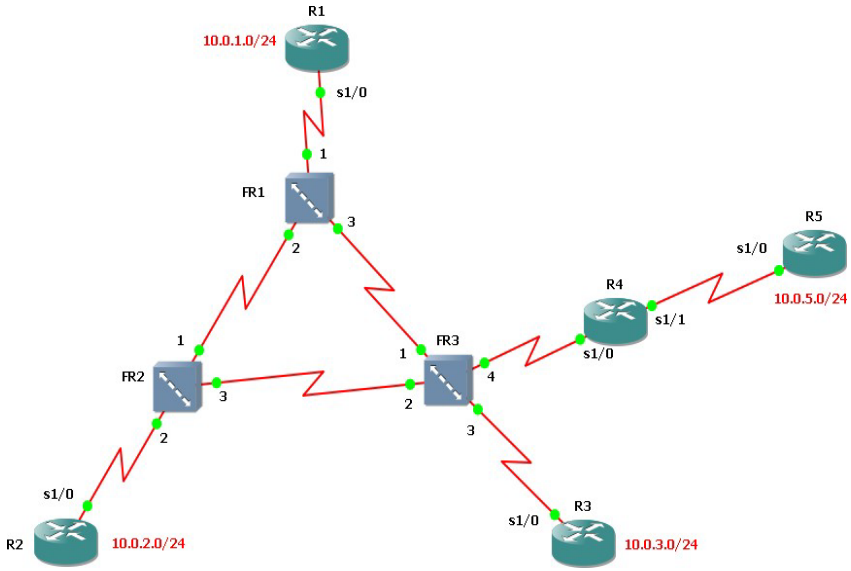


Рис. 5.1. Структурна схема Frame Relay мережі

2. Використовуйте глобальну адресацію Frame Relay: маршрутизатору R1 відповідає DLCI №101, R2 – 102 і так далі. Комутатори налаштовуються так, щоб забезпечити повну зв'язність між маршрутизаторами R1, R2 і R3. Також буде потрібен додатковий зв'язок між маршрутизаторами R3 і R5. Маршрутизатор R4 буде виконувати функції FR-комутатора (поки ніяк не налаштовується).

3. Використовуйте маршрутизатори серії 7200. У всі маршрутизатори додайте такі модулі: C7200- IO-FE і PA-4T+. Останній із зазначених модулів несе на собі чотири серіальних порти, які і будуть використовуватися для підключення до комутаторів Frame Relay.

4. Для всіх маршрутизаторів увімкніть усі серіальні інтерфейси, до яких під'єднані канали зв'язку.

5. На підключених серіальних інтерфейсах роутерів R1, R2 і R3 налаштуйте інкапсуляцію Frame Relay за допомогою команди *encapsulation frame-relay*.

6. З'ясуйте, про які номери DLCI повідомляють комутатори кожному маршрутизатору.

R1#sho frame-relay pvc

7. Для мережі між роутерами R1-R3 використовуйте адреси з мережі 192.168.0.0/24, так, наприклад, для R1 використовуйте адресу 192.168.0.1. Призначте відповідні адреси на інтерфейси Serial1/0 маршрутизаторів R2 і R3.

8. Використовуючи команду *sho int se 1/0*, з'ясуйте, який тип LMI використовується.

9. За допомогою команди *show frame-relay map* переконайтеся в коректності роботи протоколу Inverse ARP.

R1#sho frame-relay map

Serial1/0 (up): ip 192.168.0.2 dlci 102(0x66,0x1860), dynamic, broadcast, status defined, active

10. Перевірте наявність зв'язності між роутерами R1 і R2 за допомогою команди *ping*.

R1#ping 192.168.0.2

11. На маршрутизаторах R1 і R2 вимкніть інтерфейси Serial 1/0. За допомогою інтерфейсної команди *no frame-relay inverse-arp* вимкніть зазначений протокол. Увімкніть інтерфейси. Переконайтеся у відсутності правильних пар IP-DLCI.

12. Використовуйте інтерфейсну команду *frame-relay map ip* з відповідними параметрами для налаштування статичної відповідності між IP-адресою сусіднього пристрою та його DLCI.

R1(config-if)#frame-relay map ip 192.168.0.2 102

R1(config-if)#^Z

R1#sho frame-relay map

Serial1/0 (up): ip 192.168.0.2 dlci 102(0x66,0x1860), static, CISCO, status defined, active

13. Переконайтеся в працездатності статичного зв'язування за допомогою команди *ping*. Вимкніть настроювані інтерфейси, видаліть статичні записи, увімкніть підтримку протоколу Inverse ARP, увімкніть інтерфейси. Переконайтеся, що знову відбулося динамічне зіставлення IP-DLCI.

14. На роутері R3 створіть підінтерфейс Serial1/0.1 типу multipoint. Призначте на нього відповідну IP-адресу. За допомогою команди *frame-relay interface-dlci* з необхідним параметром підключіть потрібні для зв'язку з R1 і R2 номери віртуальних каналів. Приклад отриманої конфігурації представлено нижче.

interface Serial1/0

no ip address

encapsulation frame-relay
serial restart-delay 0
interface Serial1/0.1 multipoint
ip address 192.168.0.3 255.255.255.0
frame-relay interface-dlci 101
frame-relay interface-dlci 102

15. За допомогою команди *ping* переконайтеся в наявності зв'язку між роутерами R1, R2 і R3.

16. На роутерах R1-R3 налаштуйте динамічну маршрутизацію EIGRP (відключити автоматичне підсумовування маршрутів потрібно за допомогою команди *no auto-summary*) на серіальних інтерфейсах (підінтерфейсах). Налаштуйте передачу інформації про безпосередньо підключені мережі в EIGRP за допомогою команди *redistribute connected*.

17. Переконайтеся в появі коректної маршрутної інформації в таблицях маршрутизації роутерів R1-R3. Переконайтеся в доступності інтерфейсів Loopback 1 сусідніх маршрутизаторів.

18. Створіть новий підінтерфейс типу point-to-point на маршрутизаторі R3. Прив'яжіть до нього DLCI, яка буде використовуватися для обміну даними з роутером R5. Виділіть IP-мережу для зв'язку між R3 і R5. На новий підінтерфейс призначте IP-адресу з виділеної мережі.

19. На маршрутизаторі R5 також створіть субінтерфейс типу point-to-point, підключіть до нього DLCI для зв'язку з R3 і налаштуйте вільну адресу зі щойно виділеної підмережі.

20. Вивчіть, які DLCI відомі маршрутизаторам R3 і R5, особливу увагу приділіть їхнім статусам. Спробуйте пояснити, що бачите.

21. На роутері R4 увімкніть обидва серійні інтерфейси, що використовуються. У режимі глобальної конфігурації виконайте команду *frame-relay switching*, а для інтерфейсу Serial 1/1 (підключення до R5) – *frame-relay intf-type dce*. Зазначена команда переводить інтерфейс у режим DCE, у всіх інших випадках ця команда не потрібна, оскільки функції пристрою Frame Relay DCE виконують FR-комутатори.

22. Тепер необхідно налаштувати R4 на комутацію фреймів, що приходять через певний PVC одного інтерфейсу, у певний PVC іншого інтерфейсу. Домогтися поставленого завдання можна за допомогою інтерфейсної команди *frame-relay route* з відповідними

аргументами: спочатку потрібно вибрати номер DLCI поточного інтерфейсу, після чого вказати новий інтерфейс і новий номер DLCI. У лістингу нижче наведено приклад комутації DLCI №107 з інтерфейсу Serial 1/0 в DLCI №103 інтерфейсу Serial 1/1.

```
interface Serial1/0
no ip address
encapsulation frame-relay
serial restart-delay 0
frame-relay route 107 interface Serial1/1 103
interface Serial1/1
no ip address
encapsulation frame-relay
serial restart-delay 0
frame-relay intf-type dce
frame-relay route 103 interface Serial1/0 107
```

23. З'ясуйте, як змінився статус DLCI №103 на маршрутизаторі R5. Поясніть, у чому причина таких змін.

24. Переконайтеся в наявності зв'язності між роутерами R3 і R5.

25. Налаштуйте протокол динамічної маршрутизації EIGRP для роботи на каналі між роутерами R3 і R5. На маршрутизаторі R5 передайте в EIGRP інформацію про безпосередньо підключені мережі. Переконайтеся в можливості обміну даними між інтерфейсами Loopback 1 роутерів R1 і R5. З'ясуйте маршрут, яким передаються дані. Поясніть чому дані передаються саме таким шляхом.

26. Запропонуйте зміни в існуючій мережі так, щоб дані між R1 і R5 передавалися "напряму", тобто минаючи маршрутизатор R3.

27. Подивіться, які типи інтерфейсів доступні в команді *frame-relay route*. Придумайте ситуацію, в якій міг би використовуватися інтерфейс типу Tunnel. Намалюйте відповідну мережу, в якій би знадобилося використання комутації в тунель. Проведіть моделювання в емуляторі розробленої мережі.

28. Придумайте та реалізуйте складнішу топологію L2 сегмента мережі Frame Relay. Реалізуйте передачу даних між підключеними до мережі маршрутизаторами різними шляхами через FR-хмару.

Запитання для самоконтролю

1. Де використовується Frame Relay?
2. Як працює Frame Relay?
3. На якому рівні Frame Relay забезпечує виявлення помилок?

4. Які два типи віртуальних каналів визначають стандарти Frame Relay?

5. Який тип мультиплексування застосовується у Frame Relay?

6. На якому рівні моделі OSI працює технологія Frame Relay?

7. У чому відмінності комутації пакетів і каналів?

8. Де використовується комутація пакетів?

9. Можете сформулювати що таке frame relay комутатор і для чого він потрібен?

10. Які особливості функціонування Frame Relay.

Лабораторна робота 6

ТЕХНОЛОГІЯ ВІРТУАЛЬНОЇ МАРШРУТИЗАЦІЇ ТА ПЕРЕАДРЕСАЦІЇ VRF

Мета роботи: дослідити процес створення віртуальних маршрутизаторів на базі одного фізичного пристрою за допомогою технології віртуальної маршрутизації та переадресації VRF.

Основні теоретичні відомості

Віртуальна маршрутизація та переадресація (Virtual Routing and Forwarding, VRF) – це технологія, яка дозволяє кільком екземплярам таблиці маршрутизації співіснувати в одному і тому ж маршрутизаторі одночасно. Один або кілька логічних або фізичних інтерфейсів можуть мати VRF, і ці VRF не розділяють маршрути, тому пакети пересилаються тільки між інтерфейсами в одному і тому ж VRF. VRF є еквівалентом TCP / IP рівня 3 для VLAN . Оскільки екземпляри маршрутизації незалежні, однакові IP-адреси або ті, що перекриваються, можуть використовуватися без конфлікту одна з одною. Функціональність мережі покращено, оскільки мережеві шляхи можуть бути сегментовані без використання декількох маршрутизаторів.

Найпростішою формою реалізації VRF є VRF Lite. У цій реалізації кожен маршрутизатор у мережі бере участь у середовищі віртуальної маршрутизації на рівноправній основі. Незважаючи на те, що VRF Lite простий у розгортанні та підходить для малих і середніх підприємств і загальних центрів обробки даних, він не масштабується до розміру, необхідного для глобальних підприємств або великих операторів зв'язку, оскільки є потреба впроваджувати кожен примірник VRF на кожному маршрутизаторі, включно з проміжними мар-

шрутизаторами. Спочатку VRF були представлені в поєднанні з багатопротокольною комутацією міток (MPLS), але VRF виявився настільки корисним, що в кінцевому підсумку перетворився на незалежний від MPLS. Це історичне пояснення терміна VRF Lite: використання VRF без MPLS.

Обмеження масштабування VRF Lite усуваються реалізацією IPVPN. У цій реалізації базова магістральна мережа відповідає за передачу даних по широкій області між екземплярами VRF у кожному прикордонному місці розташування. IP VPN традиційно використовувалися операторами зв'язку для забезпечення спільної глобальної магістральної мережі для кількох клієнтів. Вони також підходять для великих підприємств, багатокористувацьких і загальних центрів обробки даних.

У типовому розгортанні граничні маршрутизатори (CE) обробляють локальну маршрутизацію традиційним чином і поширюють інформацію про маршрутизацію на кордон постачальника (PE), де виконується маршрутизація, таблиці віртуалізовані. Потім маршрутизатор PE інкапсулює трафік, маркує його для ідентифікації екземпляра VRF і передає його через магістральну мережу провайдера до маршрутизатора PE призначення. Потім маршрутизатор PE призначення декапсулює трафік і пересилає його маршрутизатору CE в пункті призначення. Магістральна мережа повністю прозора для обладнання замовника, що дає змогу кільком клієнтам або співтовариствам користувачів використовувати загальну магістральну мережу, зберігаючи водночас наскрізний поділ трафіку.

Маршрути через магістральну мережу провайдера підтримуються з використанням протоколу внутрішнього шлюзу - зазвичай iBGP. IBGP використовує розширені атрибути спільноти в загальній таблиці маршрутизації, щоб розрізняти маршрути клієнтів, у яких перекриваються IP-адреси.

IP VPN найчастіше розгортається в магістралі MPLS, оскільки притаманне маркування пакетів у MPLS дає змогу ідентифікувати VRF клієнта. Деякі реалізації IP VPN (зокрема, Nortel IP-VPN Lite) використовують простішу інкапсуляцію IP-in-IP поверх чистої IP-магістралі, що усуває необхідність в обслуговуванні та підтримці середовища MPLS.

Використання Cisco VRF (Virtual Routing and Forwarding) надає низку значних переваг для мережевої інфраструктури:

1. Віртуальний поділ мережевого трафіку. За допомогою VRF можна створювати віртуальні екземпляри маршрутизації, що розділяють мережевий трафік між різними сегментами мережі. Це дає змогу ізолювати й обробляти різні види трафіку незалежно один від одного, збільшуючи безпеку й ефективність мережі.

2. Логічний поділ маршрутизації. Кожен VRF має власну таблицю маршрутизації, що дає змогу розділяти маршрути та керувати ними незалежно для кожного логічного сегмента мережі. Це допомагає запобігти конфліктам і дає змогу гнучкіше налаштовувати маршрутизацію в різних частинах мережі.

3. Масштабованість і гнучкість. Використання VRF дає змогу легко додавати або змінювати віртуальні екземпляри маршрутизації без необхідності змін в основній мережевій інфраструктурі. Це робить мережу гнучкішою, що особливо важливо при зміні або розширенні мережевих вимог.

4. Керування та безпека. Можна налаштовувати окремі політики безпеки та керування для кожного VRF, що забезпечує більш гранульований контроль над трафіком у мережі. Це дає змогу оптимізувати передавання даних, ізолювати вразливі сегменти та запобігати поширенню мережевих атак на інші частини мережі.

Використання Cisco VRF є ефективним способом поліпшення мережевої архітектури та забезпечення безпечнішого та гнучкішого управління трафіком у мережі.

Послідовність виконання роботи

1. Побудуйте мережу, представлену на рис. 6.1:

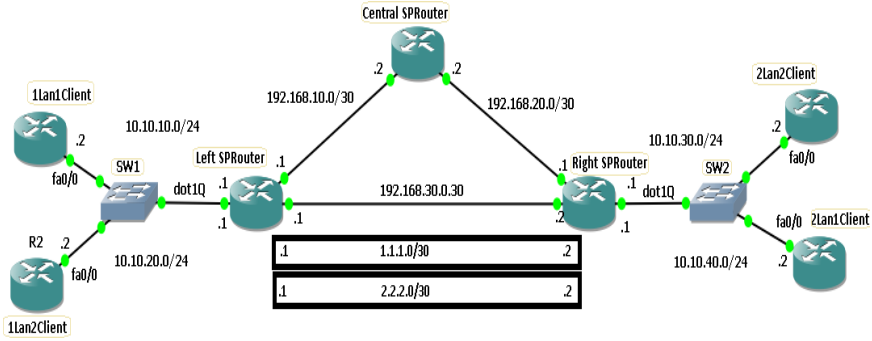


Рис. 6.1. Структурна схема мережі з використанням технології VRF

Тут маршрутизатори *LeftSPRouter, CentralSPRouter, RightSPRouter* моделюють мережу провайдера, *1Lan1Client, 2Lan1Client* – віддалені мережі першого клієнта, *1Lan2Client, 2Lan2Client* – другого клієнта. Для мережі провайдера використовуємо маршрутизатори Cisco серії 7200, офісні мережі реалізуються маршрутизаторами Cisco серії 3600.

2. Додамо необхідні маршрутизатори та комутатори. Налаштуємо комутатори: на SW1 інтерфейс 1 перебуває в режимі trunk (на комутаторі в GNS3 такий режим називається dot1q), інтерфейс 2 у VLAN 2 у режимі access, інтерфейс 3 у VLAN 3 у режимі access; на SW2 аналогічні налаштування. З'єднаємо пристрої як на схемі, причому *1Lan1Client* підключимо до другого інтерфейсу SW1, *1Lan2Client* – до третього інтерфейсу SW1; *2Lan2Client* - до третього інтерфейсу SW2, *2Lan1Client* – до другого інтерфейсу SW2. На цьому моделювання закінчено, переходимо до налаштування.

3. Налаштуємо запропоновану схему для першого клієнта. Для цього необхідно створити віртуальні пристрої на маршрутизаторах, потім налаштувати всі необхідні інтерфейси на пристроях, після чого підняти GRE-тунель на loopback-ах і, нарешті, налаштувати динамічну маршрутизацію.

Спочатку потрібно налаштувати VRF-маршрутизатори. На *LeftSPRouter* у режимі глобального конфігурування додамо новий віртуальний маршрутизатор.

LeftSPRouter(config)# ip vrf Client1vrf

Присвоїмо йому унікальний ідентифікатор.

LeftSPRouter(config-vrf)# rd 1:1

Проробимо те саме на *RightSPRouter*.

RightSPRouter(config)# ip vrf Client1vrf

RightSPRouter(config-vrf)#rd 1:2

4. Далі, налаштуємо *1Lan1Client.* Перейдемо в режим конфігурації інтерфейсу в бік комутатора (у нашому випадку він називається fa0/0) і присвоїмо йому IP-адресу з маскою командою **ip address** *10.10.10.2 255.255.255.0*. Увімкнемо інтерфейс командою **no shutdown**. Крім того, створимо loopback-інтерфейс, що моделює мережу клієнта.

1Lan1Client(config)# int loopback 1

1Lan1Client(config-if)# ip address 10.10.11.1 255.255.255.255.0

5. Переходимо до налаштування *LeftSPRouter*. У режимі конфігурації інтерфейсу в бік *CentralSPRouter* присвоїмо йому IP-адресу з маскою командою **ip address** *192.168.10.1 255.255.255.252.* Піднімемо інтерфейс командою **no shutdown**.

Те саме зробимо для інтерфейсу в бік *RightSPRouter*.

LeftSPRouter(config-if)# ip address 192.168.30.1 255.255.255.252

LeftSPRouter(config-if)# no shutdown

6. Тепер налаштуємо інтерфейс у бік комутатора SW1. Піднімемо підінтерфейс, додамо його в наш віртуальний маршрутизатор, налаштуємо інкапсуляцію, присвоїмо IP-адресу і маску.

LeftSPRouter(config)# int fa0/0.2

LeftSPRouter(config-if)# ip vrf forwarding Client1vrf

LeftSPRouter(config-if)# encapsulation dot1Q 2

LeftSPRouter(config)# ip address 10.10.10.1 255.255.255.0

7. Переходимо до налаштування *CentralSPRouter*.

CentralSPRouter(config)# int fa0/0

CentralSPRouter(config-if)# ip address 192.168.10.2 255.255.255.252

CentralSPRouter(config-if)# no shutdown

CentralSPRouter(config)# int fa0/1

CentralSPRouter(config-if)# ip address 192.168.20.2 255.255.255.252

CentralSPRouter(config-if)# no shutdown

8. Тепер *RightSPRouter*. Аналогічні налаштування, що і для *LeftSPRouter*. Налаштуємо і піднімемо інтерфейси в бік *LeftSPRouter* і *CentralSPRouter* командами **ip address** *192.168.20.1 255.255.255.252* і **ip address** *192.168.30.2 255.255.255.252* відповідно. Тепер налаштуємо інтерфейс у бік комутатора SW2. Піднімемо підінтерфейс, додамо його в наш віртуальний маршрутизатор, налаштуємо інкапсуляцію, присвоїмо IP-адресу і маску.

RightSPRouter(config)# int fa0/0.2

RightSPRouter(config-if)# ip vrf forwarding Client1vrf

RightSPRouter(config-if)# encapsulation dot1Q 2

RightSPRouter(config)# ip address 10.10.40.1 255.255.255.0

9. Залишилося налаштувати *2Lan1Client*. Перейдемо в режим конфігурування інтерфейсу в бік комутатора і присвоїмо йому IP-адресу з маскою командою **ip address** *10.10.40.2 255.255.255.0.*

Піднімемо інтерфейс командою **no shutdown**. Крім того, створимо loopback-інтерфейс, що моделює мережу клієнта.

2Lan1Client(config)# int loopback 1

2Lan1Client(config-if)# ip address 10.10.41.1 255.255.255.255.0

10. На цьому перший етап налаштування завершено. Тепер необхідно налаштувати GRE-тунель на інтерфейсах loopback між *RightSPRouter* і *LeftSPRouter*. У режимі глобальної конфігурації *LeftSPRouter* додаємо інтерфейс loopback1.

LeftSPRouter(config)# int loopback1

LeftSPRouter(config-if)# ip address 1.1.3.1 255.255.255.252

11. Налаштування тунелю: піднімаємо тунельний інтерфейс, додаємо його в наш віртуальний маршрутизатор, налаштовуємо тунель.

LeftSPRouter(config)# int tunnel1

LeftSPRouter(config-if)# ip vrf forwarding Client1vrf

LeftSPRouter(config-if)# ip address 1.1.1.1 255.255.255.252

LeftSPRouter(config-if)# tunnel source loopback1

LeftSPRouter(config-if)# tunnel destination 1.1.4.1

LeftSPRouter(config-if)# tunnel key 1

Передостання команда вказує адресу іншого кінця тунелю, який ми налаштуємо далі, а остання команда необхідна для ідентифікації тунелю.

Пояснимо призначення команди ідентифікації тунелю *tunnel key*. Часто тунелі клієнтів піднімаються на одних і тих самих інтерфейсах (у нашому випадку це інтерфейси loopback), через що виникає неоднозначність визначення приналежності вхідного пакета тому чи іншому тунелю. У цьому легко переконатися самостійно: без вказівки ключа функціонуватиме тільки один тунель (той, що був налаштований останнім), тобто в нашому випадку після налаштування тунелю другого клієнта на тих самих loopback-інтерфейсах тунель першого клієнта відключиться. Вирішити проблему неоднозначності можна, якщо налаштувати тунелі різних клієнтів на різних інтерфейсах, що ресурсоємно. Простіше вказати ключ ідентифікації тунелю, що і було зроблено.

12. Тепер налаштуємо *RightSPRouter*.

RightSPRouter(config)# int loopback1

RightSPRouter(config-if)# ip address 1.1.4.1 255.255.255.255.252

13. Налаштування тунелю: піднімаємо тунельний інтерфейс, додаємо його в наш віртуальний маршрутизатор, налаштовуємо тунель.

RightSPRouter(config)# int tunnel1
RightSPRouter(config-if)# ip vrf forwarding Client1vrf
RightSPRouter(config-if)# ip address 1.1.1.2 255.255.255.252
RightSPRouter(config-if)# tunnel source loopback1
RightSPRouter(config-if)# tunnel destination 1.1.3.1
RightSPRouter(config-if)# tunnel key 1

Тунель налаштований, але використовуватися він не буде, поки не буде налаштована динамічна маршрутизація.

14. Перейдемо до останнього етапу налаштування. В якості протоколу динамічної маршрутизації виберемо OSPF.

У мережі провайдера на *LeftSPRouter*.

LeftSPRouter(config)# router ospf 1
LeftSPRouter(config-router)# network 192.168.10.0 0.0.0.3 area 0
LeftSPRouter(config-router)# network 192.168.30.0 0.0.0.3 area 0
LeftSPRouter(config-router)# network 1.1.3.0 0 0.0.0.3 area 0

15. Тепер для маршрутизації в VRF *Client1vrf* на *LeftSPRouter* виконаємо нижченаведені команди.

LeftSPRouter(config)# router ospf 2 vrf Client1vrf
LeftSPRouter(config-router)# network 10.10.10.0 0.0.0.255 area 0
LeftSPRouter(config-router)# network 1.1.1.0 0.0.0.3 area 0

16. У мережі провайдера на *RightSPRouter*.

RightSPRouter(config)# router ospf 1
RightSPRouter(config-router)# network 192.168.20.0 0.0.0.3 area 0
RightSPRouter(config-router)# network 192.168.30.0 0.0.0.3 area 0
RightSPRouter(config-router)# network 1.1.4.0 0 0.0.0.3 area 0

17. Тепер для маршрутизації в VRF *Client1vrf* на *RightSPRouter*.

RightSPRouter(config)# router ospf 2 vrf Client1vrf
RightSPRouter(config-router)# network 10.10.40.0 0.0.0.255 area 0
RightSPRouter(config-router)# network 1.1.1.0 0.0.0.3 area 0

18. У мережі провайдера на *CentralSPRouter*.

CentralSPRouter(config)# router ospf 1
CentralSPRouter(config-router)# network 192.168.20.0 0.0.0.3 area 0
CentralSPRouter(config-router)# network 192.168.10.0 0.0.0.3 area 0

19. На *1Lan1Client* (номер OSPF-процесу – номер VLAN).

1Lan1Client(config)# router ospf 2

1Lan1Client(config-router)# network 10.10.10.0 0.0.0.255 area 0
1Lan1Client(config-router)# network 10.10.11.0 0.0.0.255 area 0

20. На *2Lan1Client* (номер OSPF-процесу - номер VLAN).

2Lan1Client(config)# router ospf 2
2Lan1Client(config-router)# network 10.10.40.0 0.0.0.255 area 0
2Lan1Client(config-router)# network 10.10.41.0 0.0.0.255 area 0

На цьому налаштування для першого клієнта завершено.

21. Налаштування для другого клієнта майже нічим не відрізняється. Нижче наводяться налаштування на кожному пристрої.

1Lan2Client(config)# int fa0/0
1Lan2Client(config-if)# ip address 10.10.20.2 255.255.255.0
1Lan2Client(config)# no shutdown
1Lan2Client(config)# int loopback 1
1Lan2Client(config-if)# ip address 10.10.21.1 255.255.255.0
1Lan2Client(config)# router ospf 3
1Lan2Client(config-router)# network 10.10.20.0 0.0.0.255 area 0
1Lan2Client(config-router)# network 10.10.21.0 0.0.0.255 area 0
LeftSPRouter(config)# ip vrf Client2vrf
LeftSPRouter(config-vrf)# rd 2:1
LeftSPRouter(config)# int fa0/0.3
LeftSPRouter(config-if)# ip vrf forwarding Client2vrf
LeftSPRouter(config-if)# encapsulation dot1Q 3
LeftSPRouter(config)# ip address 10.10.20.1 255.255.255.0
LeftSPRouter(config)# int tunnel2
LeftSPRouter(config-if)# ip vrf forwarding Client2vrf
LeftSPRouter(config-if)# ip address 1.1.2.1 255.255.255.252
LeftSPRouter(config-if)# tunnel source loopback1
LeftSPRouter(config-if)# tunnel destination 1.1.4.1
LeftSPRouter(config-if)# tunnel key 2
LeftSPRouter(config)# router ospf 3 vrf Client2vrf
LeftSPRouter(config-router)# network 10.10.20.0 0.0.0.255 area 0
LeftSPRouter(config-router)# network 1.1.2.0 0.0.0.3 area 0
RightSPRouter(config)# ip vrf Client2vrf
RightSPRouter(config-vrf)# rd 2:2
RightSPRouter(config)# int fa0/0.3
RightSPRouter(config-if)# ip vrf forwarding Client2vrf
RightSPRouter(config-if)# encapsulation dot1Q 3

RightSPRouter(config)# ip address 10.10.30.1 255.255.255.0
RightSPRouter(config)# int tunnel2
RightSPRouter(config-if)# ip vrf forwarding Client2vrf
RightSPRouter(config-if)# ip address 1.1.2.2 255.255.255.252
RightSPRouter(config-if)# tunnel source loopback1
RightSPRouter(config-if)# tunnel destination 1.1.3.1
RightSPRouter(config-if)# tunnel key 2
RightSPRouter(config)# router ospf 3 vrf Client2vrf
RightSPRouter(config-router)# network 10.10.30.0 0.0.0.255 area 0
RightSPRouter(config-router)# network 1.1.2.0 0.0.0.3 area 0
2Lan2Client(config)# int fa0/0
2Lan2Client(config-if)# ip address 10.10.30.2 255.255.255.0
2Lan2Client(config)# no shutdown
2Lan2Client(config)# int loopback 1
2Lan2Client(config-if)# ip address 10.10.31.1 255.255.255.0
2Lan2Client(config)# router ospf 3
2Lan2Client(config-router)# network 10.10.30.0 0.0.0.255 area 0
2Lan2Client(config-router)# network 10.10.31.0 0.0.0.255 area 0

CentralSPRouter налаштування непотребує. Нацьому конфігурація закінчується, перейдемо до тестування мережі.

22. Для початку за допомогою команд *ping 10.10.40.2 source 10.10.11.1* з *1Lan1Client* і *ping 10.10.30.2 source 10.10.41.1* з *1Lan2Client* переконаємося, що пакети успішно ходять мережею. Крім того, використовуючи *traceroute* з тих самих пристроїв за тими самими адресами, вивчимо шляхи проходження цих пакетів.

23. Далі переконаємося, що протокол OSPF функціонує, прописавши *show ip protocols* і *show ip route* на всіх пристроях мережі. Подумайте, як зазначені команди дають змогу переконатися в правильній роботі протоколу OSPF?

24. На *LeftSPRouter* і *RightSPRouter* переглянемо результат роботи команди *show ip protocols vrf Client1vrf (show ip protocols vrf Client2vrf)*. Проаналізуйте отримані дані.

25. На *LeftSPRouter* і *RightSPRouter* вивчимо результат роботи команди *show ip route vrf Client1vrf (show ip route vrf Client2vrf).*

26. Використовуючи Wireshark, перехопимо пакети на каналах між *LeftSPRouter* і *RightSPRouter*. Проаналізуйте результати перехоплення.

27. Нарешті, перевіримо відмовостійкість мережі: відключимо канал між *LeftSPRouter* і *RightSPRouter* і переконаємося, що мережа, як і раніше, функціонує. Вкажіть, які з пристроїв здатні виявити зміни в мережі провайдера.

28. Відновити працездатність відключеного в попередньому пункті каналу. Переконайтеся в нормалізації маршрутизації в мережі оператора.

29. Запропонуйте рішення, що дає змогу передавати користувацький IPv6 трафік між мережами клієнтів так, щоб не знадобилося переналаштування операторської мережі.

30. Реалізуйте запропоноване в попередньому пункті рішення.

Запитання та завдання для самоконтролю

1. Для чого потрібен VRF?
2. Яка технологія застосовується для створення віртуальних маршрутизаторів на базі одного фізичного?
3. У чому відмінність VRV від VRF?
4. Які сервіси з перерахованих можна використовувати тільки у віртуальному маршрутизаторі за замовчуванням?
5. Що таке система VRV?
6. У чому різниця між VLAN і VRF?
7. Що таке VRF в Cisco?
8. У чому різниця між VRF і VRRP?
9. У чому різниця між VRF і віртуальним маршрутизатором?
10. Що означає VRF в MPLS?
11. В чому різниця між Cisco ip VRF і VRF?

Лабораторна робота 7

ПОБУДОВА ТУНЕЛІВ НА МАРШРУТИЗАТОРАХ

Мета роботи: дослідити процес побудови тунелів на маршрутизаторах Cisco з використанням інтерфейсів Loopback.

Основні теоретичні відомості

Тунелювання забезпечує метод інкапсулювання довільних пакетів усередині транспортного протоколу. Ця можливість реалізована як віртуальний інтерфейс для того, щоб спростити її налаштування. Tunnel Interface не прив'язаний до "passenger" або "transport" протоколів, точніше це архітектура, розроблена для реалізації будь-якої

стандартної схеми інкапсуляції точка-точка. Оскільки тунелювання являє собою з'єднання точка-точка, потрібно конфігурувати тунель для кожного з'єднання.

Тунелювання складається з 3-х основних компонентів:

1. Passenger protocol, протокол, який інкапсулюється і передається в тунелі (AppleTalk, Banyan VINES, CLNS, DECnet, IP, or IPX);
2. Carrier protocol, один із таких протоколів інкапсуляції: Generic route encapsulation (GRE); мультипротокольний перевізниковий протокол Cisco; Cayman, власний протокол для AppleTalk over IP; EON, стандарт для перенесення CLNP через IP-мережі; NOS, IP over IP, сумісний із популярною програмою KA9Q; Distance Vector Multicast Routing Protocol (DVMRP) (IP в IP-тунелях, визначений RFC 20036);
3. Transport protocol, який використовується для перенесення інкапсулюючого протоколу (тільки IP).

Тунелювання зручно застосовувати в таких випадках:

1. для забезпечення роботи багатьох несумісних протоколів локальної мережі через backbone-мережу з одним протоколом;
2. для забезпечення робочого оточення для мереж, що містять протоколи з обмеженою кількістю вузлів – наприклад, AppleTalk;
3. для з'єднання переривчастих підмереж;
4. для роботи VPN через WAN.

Під час організації тунелів потрібно брати до уваги таке:

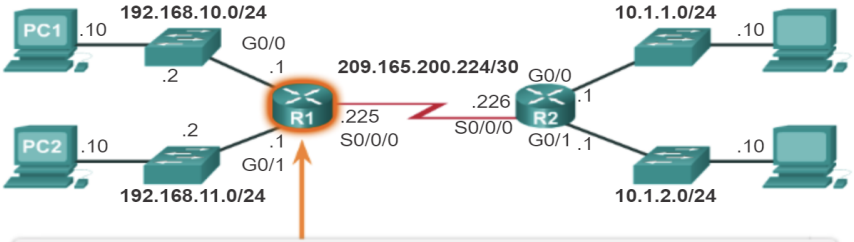
1. інкапсуляція і деінкапсуляція трафіку на кінцях тунелю є повільними операціями і сильно завантажують процесор пристрою Cisco (підтримується тільки процесорна комутація пакетів);
2. тунелювання може створювати проблеми з протоколами, що мають обмежувальні таймери (наприклад, DECnet), оскільки тунелінг збільшує затримку пакетів (latency);
3. найбільша проблема виходить, коли інформація роутингу тунельованої мережі змішується з інформацією роутингу транспортної мережі. У цьому випадку можуть утворюватися зациклення маршрутів. Для вирішення проблеми потрібно розділити роутинг тунельованої і транспортної мереж:
   1. використовувати для них різні AS номери;
   2. використовувати різні протоколи роутингу;
   3. використовувати статичні маршрути для подолання першого вузла маршруту.

Інтерфейс loopback – це логічний інтерфейс усередині маршрутизатора. Він не призначається фізичному порту, тому його не можна під'єднати до іншого пристрою. Він вважається програмним інтерфейсом, який автоматично переводиться в стан UP під час роботи маршрутизатора.

Застосування інтерфейсу loopback може бути доцільним під час тестування та управління пристроєм Cisco IOS, оскільки він забезпечує доступність хоча б одного інтерфейсу. Його можна використовувати з метою тестування – наприклад, для тестування внутрішніх процесів маршрутизації, шляхом імітації мереж за межами маршрутизатора.

Крім того, IPv4-адреса, призначена loopback-інтерфейсу, може бути необхідною для процесів маршрутизатора, в яких використовується IPv4-адреса інтерфейсу з метою ідентифікації. Один із таких процесів – алгоритм найкоротшого шляху (OSPF). Під час увімкнення інтерфейсу loopback для ідентифікації маршрутизатор використовуватиме завжди доступну адресу інтерфейсу loopback, ніж IP-адресу, призначену фізичному порту, робота якого може бути порушена.

Увімкнення інтерфейсу і призначення loopback-адрес виконується за допомогою простого набору команд (рис. 7.1).




```
R1(config)# interface loopback 0
R1(config-if)# ip address 10.0.0.1 255.255.255.0
R1(config-if)# exit
R1(config)#
*Jan 30 22:04:50.899: %LINK-3-UPDOWN: Interface loopback0,
changed state to up
*Jan 30 22:04:51.899: %LINEPROTO-5-UPDOWN: Line protocol on
Interface loopback0, changed state to up
```

Рис. 7.1. Структурна схема мережі та увімкнення інтерфейсу loopback

На маршрутизаторі можна активувати кілька інтерфейсів loopback. IPv4-адреса для кожного інтерфейсу loopback має бути унікальною і не повинна бути задіяна іншим інтерфейсом.

Усі TCP/IP реалізації підтримують loopback механізми, які реалізують віртуальний мережевий інтерфейс винятково програмно і не пов'язані з будь-яким обладнанням, але при цьому повністю інтегровані у внутрішню мережеву інфраструктуру комп'ютерної системи. Будь-який трафік, який надсилається комп'ютерною програмою на інтерфейс loopback, одразу ж отримується тим самим інтерфейсом.

Відповідно, Internet Protocol специфікує мережу loopback. В IPv4 це мережа з префіксом 127.0.0.0/8 (RFC 5735). Найбільш широко використовувана IP адреса в механізмах loopback – 127.0.0.1. В IPv4, у неї також відображається будь-яка адреса в межах від 127.0.0.0 до 127.255.255.255. IPv6 визначає єдину адресу для цієї функції – 0:0:0:0:0:0:0:0:1/128 (також записується як ::1/128) (RFC 4291). Стандартне, офіційно зарезервоване, доменне ім'я для цих адрес – localhost (RFC 2606).

На системах Unix інтерфейс loopback зазвичай має ім'я lo або lo0.

Інтерфейс loopback має кілька шляхів застосування. Він може бути використаний мережевим клієнтським програмним забезпеченням, щоб спілкуватися з серверним додатком, розташованим на тому ж комп'ютері. Тобто якщо на комп'ютері, на якому запущено веб-сервер, вказати у веб-браузері URL http://127.0.0.1/ або http://localhost/, то він потрапляє на веб-сайт цього комп'ютера. Цей механізм працює без будь-якого активного підключення, тому він корисний для тестування служб, не піддаючи їхню безпеку ризику, як під час віддаленого мережевого доступу. Подібним чином, пінгування адреси loopback – це основний тест функціонування IP стека в операційній системі.

Пакети, надіслані в IP-мережу з початковою адресою, що належить до loopback-інтерфейсу, можуть спричинити низку проблем для застарілого мережевого ПЗ або ПЗ з помилками. Такі пакети відомі як Марсіанські пакети. Специфікація Internet Protocol говорить, що такі пакети не повинні передаватися далі за хост, і їх має бути знищено, якщо їх буде отримано мережевим інтерфейсом (RFC 4213).

Один примітний виняток для використання мережевих адрес loopback (127.0.0.0.0/8) – це їх використання в MPLS (мультипротокольна комутація за мітками). Технологія пошуку помилок, що визначає маршрут проходження даних, в якій властивість loopback – відсутність маршруту, дає можливість уникнути доставки несправного пакета кінцевим користувачам.

Послідовність виконання роботи

1. Побудуйте мережу, представлену на рис. 7.2.

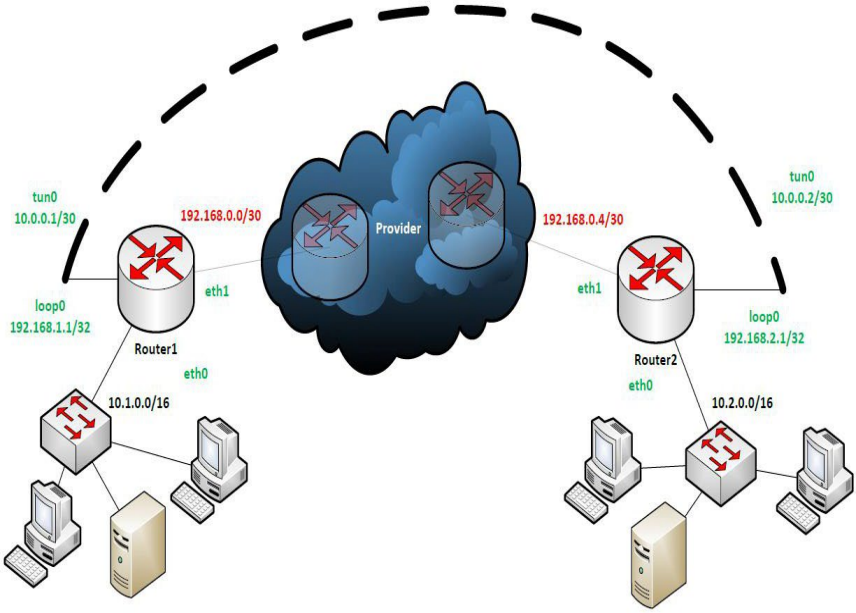


Рис. 7.2. Структурна схема мережі з тунелюванням та увімкненим інтерфейсом loopback

2. Підключіть два маршрутизатори Cisco серії 1600 до маршрутизатора Cisco серії 3600. До кожного маршрутизатора Cisco серії 1600 також підключіть комп'ютер.

3. Підключіться до консольного порту кожного з маршрутизаторів і з конфігураційного режиму задайте ім'я за допомогою команди **hostname** *name*. Наприклад, Provider, Router1, Router2.

4. Для кожного маршрутизатора в режимі конфігурації інтерфейсу налаштуйте IP-адреси інтерфейсів Ethernet0, Ethernet1 за допомогою команди **ip address** *address mask*. Для адрес інтерфейсів

Ethernet0 маршрутизаторів Cisco 1600 (до яких підключені офісні сітки) використовуємо маску /24 (mask 255.255.255.0).

5. Для кожного комп'ютера офісних клієнтських мереж налаштуйте IP-адресу, використовуючи маску /24, як default gateway вкажіть IP-адресу інтерфейсу Ethernet0 відповідного маршрутизатора Cisco серії 1600.

6. На маршрутизаторі Cisco серії 3600 (Provider) налаштуйте статичну маршрутизацію на виділені мережі, використовуючи конфігураційну команду **ip route** *ip-address mask next_hop.*

7. На кожному маршрутизаторі Cisco 1600 також пропишіть статичну маршрутизацію на віддалену офісну мережу через маршрутизатор провайдера.

8. Для відстеження стану таблиці маршрутизації використовуйте команду **show ip route**.

9. За допомогою ICMP echo-запитів (**ping**) переконайтеся в доступності кожного з маршрутизаторів із комп'ютера(-ів) і в тому, що між офісами клієнта встановлено зв'язок через маршрутизатор провайдера.

10. Тепер припустимо, що ситуація змінилася. Клієнту необхідно використовувати більшу кількість адрес, ніж надається виділеними провайдером мережами /24. Для цього при адресації офісних мереж використовується маска /16. Але провайдер, як і раніше, маршрутизує і надає клієнту мережі /24. Вирішенням проблеми зв'язку офісів клієнта може слугувати побудова тунелю на loopback-інтерфейсах двох граничних маршрутизаторів клієнта.

11. За допомогою команди **ip address** *address mask* для кожного маршрутизатора Cisco серії 1600 у режимі конфігурації інтерфейсу налаштуйте нові IP-адреси інтерфейсів Ethernet0, до яких під'єднані офісні мережі клієнта, необхідно використовувати маску /16 (mask 255.255.0.0).

12. Для кожного комп'ютера офісних клієнтських мереж налаштуйте нову IP-адресу, використовуючи маску /16.

13. Для кожного з маршрутизаторів Cisco 1600 у режимі конфігурації віртуального інтерфейсу loopback0 налаштуйте IP-адресу за допомогою команди **ip address** *address mask*, в якості mask слід використовувати /32 (mask 255.255.255.255), IP-адресу обирають із мережі, виділеної провайдером.

14. Для кожного маршрутизатора Cisco 1600 сконфігуруйте тунельний інтерфейс за допомогою конфігураційної команди **interface tunnel** *0.*

15. У режимі конфігурування інтерфейсу-тунелю налаштуйте IP-адресу, використовуючи команду **ip address** *address mask*, як маску слід вказати значення /30 (mask 255.255.255.252).

16. Використовуючи інтерфейсні команди **tunnel source** *ip-address* і **tunnel destination** *ip-address* налаштуйте IP-адреси кінцевих точок тунелю. Як *ip-address* слід використовувати налаштовані раніше адреси loopback-інтерфейсів.

17. Налаштуйте режим тунелювання за допомогою інтерфейсної команди **tunnel mode**

mode, де як mode вкажіть, наприклад, ipip.

18. Беручи до уваги сконфігурований вище тунель, створіть на кожному маршрутизаторі Cisco серії 1600 статичний маршрут на другу офісну мережу, використовуючи конфігураційну команду **ip route** *network ip-address mask next-hop.*

19. Переконайтеся в доступності кожного з маршрутизаторів з комп'ютера(-ів) за допомогою ICMP exo-запитів (**ping**).

20. Подивіться, як маршрутизуються офісні мережі, використовуючи команду **traceroute**

ip-address (**tracert** для Windows).

21. Подивіться IP-адреси відправника й одержувача пакетів, що маршрутизуються між офісними мережами клієнта. Для цього підключіть маршрутизатор Cisco серії 1600 і маршрутизатор Cisco серії 3600 (Provider) до комутатора Cisco серії Catalyst 2960, до якого також підключіть ще один комп'ютер.

22. Налаштуйте на комутаторі SPAN-сесію для копіювання даних, що проходять через комутатор, на порт, до якого під'єднано додатковий комп'ютер. Для цього використовуйте конфігураційні команди **monitor session 1 source interface** *interface* **both**, **monitor session 1 destination interface** *interface interface*, де як source interface і destination interface вкажіть порти, до яких під'єднані офісний маршрутизатор і додатковий комп'ютер, відповідно.

23. Для перегляду налаштованої SPAN-сесії скористайтеся командою **show monitor session 1**.

24. Запустіть на додатковому комп'ютері, підключеному до комутатора, програму Wireshark, "перехопіть" пакет, надісланий з

однієї офісної мережі в іншу. Подивіться MAC та IP-адреси відправника й одержувача.

25. Здійсніть таке "перехоплення" пакета в різних частинах побудованої мережі (між офісним ПК першого офісу і Router1, між Router1 і Provider, між Provider і Router2, між Router2 і офісним ПК у другому офісі), використовуючи комутатор Cisco серії 2960.

26. Запустіть протокол динамічної маршрутизації EIGRP, за допомогою якого встановіть сусідство між двома офісними маршрутизаторами через тунельні інтерфейси. Передайте за допомогою цього протоколу маршрутну інформацію про всі підключені мережі.

27. На кожному маршрутизаторі Cisco серії 1600 видаліть прописаний раніше статичний маршрут на відповідну офісну мережу з маскою /16. Це потрібно зробити, оскільки далі передбачається передавати інформацію про цю мережу по EIGRP. Зверніть увагу на стан тунелю.

28. На кожному клієнтському маршрутизаторі запустіть протокол EIGRP. Для цього слід використовувати конфігураційну команду **router eigrp** *process_number,* де як номер процесу вкажіть, наприклад, 1.

29. За допомогою команди *network network_ip-address* у режимі конфігурації роутера пропишіть сітку, у якій працюватиме EIGRP (протокол увімкнеться на всіх інтерфейсах цього маршрутизатора, адреси яких потрапляють у зазначений діапазон).

30. Вимкніть автоматичне підсумовування маршрутів за допомогою команди **no auto-summary** в режимі конфігурації маршрутизатора.

31. Командою **redistribute connected** вкажіть маршрутизатору, що слід надсилати маршрутну інформацію про всі підключені мережі.

32. Для відстеження змін використовуйте команди **show ip route eigrp**, **show ip eigrp neighbors**, **show ip eigrp interfaces**, **show ip eigrp topology**.

33. Зверніть увагу на періодичну зміну стану тунелю. Такий ефект – наслідок рекурсивної маршрутизації. Вивчіть і поясніть явище.

34. За погодженням із викладачем виправте проблему, що виникла. Для її розв'язання слід змінити конфігурацію на клієнтських

маршрутизаторах так, щоб по EIGRP надсилалася інформація про всі під'єднані мережі, але не пересилалася інформація про loopback-інтерфейс. Для цього будемо використовувати механізм route-map.

35. Створіть route-map на кожному маршрутизаторі Cisco серії 1600, використовуючи конфігураційну команду **route-map** *rm-name* **deny** *seq1*, як *seq1* вкажіть, наприклад, 10. Постфікс deny вказує на створення заборонного правила.

36. Пропишіть саму заборону за допомогою команди **match interface** *interface,* як interface вкажіть loopback-інтерфейс цього маршрутизатора. Поверніться в режим глобальної конфігурації.

37. Для створеного route-map створіть правило з дозволом, використовуючи команду **route-map** *rm-name* **permit** *seq2,* де, згідно з логікою роботи route-map, як *seq2* слід вказати значення, більше за *seq1*, наприклад, 20. За замовчуванням дозвільне правило дозволяє все, що не вказано в попередньому заборонному правилі.

38. Вкажіть, що за EIGRP слід відправляти маршрутну інформацію про всі підключені мережі, із застосуванням правил, зазначених у створеному route-map. Для цього скасуйте команду **redistribute connected** у режимі конфігурації роутера для eigrp 1, і застосуйте **redistribute connected route-map** *rm-name*.

39. Зверніть увагу на стан тунелю. Проаналізуйте зміни.

40. Запропонуйте та реалізуйте інші способи розв'язання проблеми.

Запитання для самоконтролю

1. Для чого потрібен Loopback інтерфейс?
2. Як пінгувати Loopback?
3. Як налаштувати інтерфейс Loopback на Cisco?
4. Як подивитися налаштування інтерфейсу Loopback Cisco?
5. Що таке Loopback Detection?
6. Що таке NAT Loopback?
7. Що таке тунель і в чому полягає принцип тунелювання?
8. Що таке GRE тунель навіщо він потрібен та як працює?
9. Як увімкнути Loopback Detection у VLAN?
10. Як налаштувати Loopback Detection в комутаторі Cisco?
11. Чому в OSPF використовується Loopback?
12. Навіщо використовувати Loopback для BGP?
13. Що таке тунель і в чому полягає принцип тунелювання?

Лабораторна робота 8

ТЕХНОЛОГІЯ РОЗПОДІЛУ НАВАНТАЖЕННЯ СЕРВЕРІВ SLB

Мета роботи: дослідити процес налаштування розподілу навантаження на маршрутизаторах Cisco між декількома серверами з використанням технології SLB.

Основні теоретичні відомості

Технологія розподілу навантаження серверів (Server Load Balancing, SLB) призначена для розподілу навантаження між декількома серверами, а також для забезпечення відмовостійкості сервісу, що надається, адже забезпечує продовження функціонування останнього навіть у разі відмови одного або декількох серверів. SLB – не єдина технологія, за допомогою якої можливий розподіл навантаження, часто в реальних мережах також застосовують NLB або балансування за допомогою DNS. Можливе навіть поєднання кількох варіантів розподілу навантаження і використання спеціалізованих балансерів.

Cisco Server Load Balancing може працювати у двох режимах: L2 і L3. Найпростішим для розуміння і реалізації є L3 режим, у якому маршрутизатор виконує не тільки розподіл TCP або UDP сесій між серверами, а й зміну IP-адреси одержувача (NAT) іноді разом із TCP/UDP портом одержувача (PAT). В якості адреси одержувача вибирається реальна адреса сервера. Перевагою цього методу балансування є можливість знаходження балансувального маршрутизатора і серверів у різних L3 мережах. До недоліків можна віднести підвищене навантаження на процесор, обумовлене виконанням NAT/PAT.

Під час роботи в L2 режимі трансляція NAT/PAT не виконується, тобто адреса одержувача і порт залишаються незмінними. "Вибір" сервера здійснюється лише на основі зміни MAC-адреси одержувача. Таким чином, до серверів застосовується додаткова вимога: операційна система повинна мати постійно працюючий інтерфейс, на якому призначено IP-адресу сервісу. Призначення такої IP-адреси безпосередньо на Ethernet-інтерфейс (вторинна (secondary) адреса) не є гарною ідеєю, тому що під час переходу його в робочий стан буде зроблено широкомовний ARP-запит про цю адресу, що призведе до появи повідомлення про помилку

дублювання IP-адрес у мережі (дадуть відповідь інші аналогічні сервери). На практиці для зазначеної мети використовують спеціальні інтерфейси замикання на себе (Loopback).

Коли людина користується онлайн-сервісом, незалежно від використовуваного пристрою в неї є тільки один спосіб доступу до нього – через IP-адресу або доменне ім'я. Те, що відбувається далі, від спостерігачів приховано. Прозорим залишається лише трафік від сервісу.

Це пов'язано з тим, що традиційна форма балансування навантаження використовує механізм перетворення мережевих адрес: NAT (Network Address Translation). Ця технологія маскує пул цільових серверів, на які поділяються потоки трафіку.

Історично балансувальник навантаження являв собою фізичне рішення між пулом серверів і клієнтським пристроєм. Балансувальник використовує віртуальну IP-адресу (VIP) для приймання трафіку, і запити користувачів "розмазуються" по всьому пулу серверів за допомогою алгоритму балансування. Цих алгоритмів кілька, ми розповімо про найвідоміші.

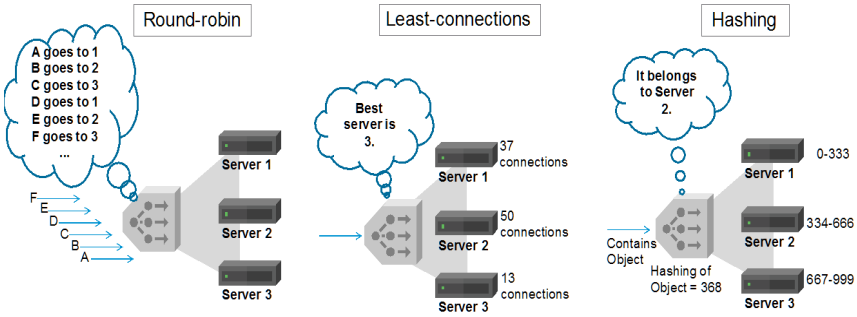


Рис. 8.1. Розподіл вхідного мережевого трафіку від клієнта між кількома бекенд-серверами

1. Round-robin. Також відомий як алгоритм кругового обслуговування. Він досить простий - усі сервери в пулі отримують запити по черзі. Наприклад, сервер №1 обробляє перший запит, сервер №2 – другий і так далі, по колу. Цей метод хороший тим, що незалежний від протоколу високого рівня, недорогий в експлуатації і не вимагає зв'язку між серверами в кластері. Але при цьому важливо, щоб усі сервери мали однаковий обсяг ресурсів. Завантаженість того чи іншого вузла в кластері не враховується, а значить

допускається перевантаження одного вузла при недовантаженості іншого.

2. Weighted. Модернізована версія попереднього алгоритму. У цій системі у кожного сервера з'являється свій коефіцієнт продуктивності залежно від його потужності та можливостей. Що більший цей ваговий коефіцієнт, то більше навантаження здатний відпрацювати сервер.

3. Least Connections. Цей алгоритм враховує кількість активних підключень, підтримуваних серверами в конкретний період часу. Іншими словами, він бачить, який сервер активно працює, а який простоює. І наступний запит направляє саме туди, де обробляється найменший обсяг трафіку.

Зростання кількості користувачів і обсягу трафіку тягне за собою збільшення навантаження на інфраструктуру сервісу. Балансувальник гарантує, що сервер не буде перевантажений трафіком і дані будуть ефективно переміщатися між компонентами кластера.

Відмовостійкість – головна мета. Складність додатків зростає, кількість точок відмови збільшується, вони можуть розташовуватися на різних рівнях інфраструктури, чи то сервери, чи то мережі. Балансувальник дає змогу уникати єдиної точки відмови - частини системи, у разі виходу з ладу якої вся робота буде зупинена. Якщо один сервер відмовить, балансувальник розподілить трафік між іншими елементами інфраструктури.

Балансувальник дає змогу більш оптимально використовувати ресурси і швидше обслуговувати запити. Наприклад, якщо у вас два сервери під бази даних, балансувальник зробить так, щоб обидва були однаково навантажені.

Також балансувальник забезпечить більш плавне масштабування інфраструктури: під час горизонтального зростання - додавання нового сервера в кластер – він швидко й акуратно завантажить нову "ланку" інфраструктури.

Інша важлива функція балансувальника – захист від DDoS-атак. Його забезпечує затримка відповіді, коли фонові сервери не бачать клієнта до підтвердження по TCP. Балансувальник навантаження Selectel проводить вихідний трафік через спеціальні алгоритми, які фільтрують TCP ACK/FIN/RST-атаки до 99,9%.

У разі, коли за одну IP-адресу відповідатимуть різні сервери, говорять про виконання мережевого балансування. Досягти такого результату можна такими способами:

1. Використання DNS-сервера. У цьому разі на ім'я одного домену виділяється кілька адрес. Користувацький запит, що надходить, буде розподілятися між ними за допомогою алгоритму Round Robin.
2. Застосування додаткового маршрутизатора. Виконується балансування за IP-адресою.
3. Створення NLB-кластера. Тут сервери поділяються на ті, які приймають вхідні запити, і ті, які займатимуться обчисленнями. Подібне рішення на практиці застосовує корпорація Майкрософт.
4. Розміщення однотипних сервісів з ідентичними адресами в різних інтернет-регіонах. Актуально під час виконання балансування за територіальною ознакою. На практиці такий варіант застосовують багато CDN. Також на ній працює і Ancast DNS-технологія.

Послідовність виконання роботи

Приклад мережі, в якій працює SLB, наведено нижче (рис. 8.2). Варто зазначити, що інтерфейси Loopback на серверах у цій схемі будуть використовуватися тільки в разі використання L2 режиму SLB. Маски для клієнта і зовнішнього інтерфейсу маршрутизатора не вказані, оскільки в реальній мережі ці адреси, звісно ж, будуть із різних підмереж і можуть бути довільними.

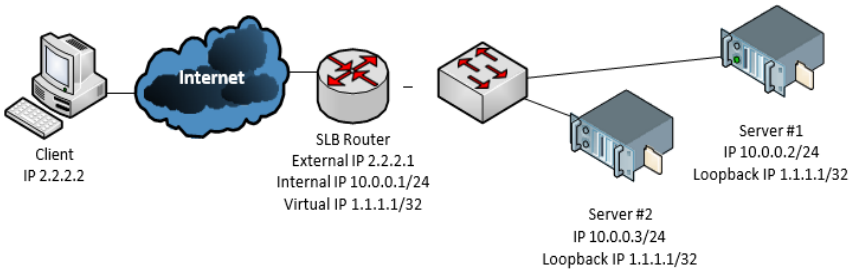


Рис. 8.2. Структурна схема мережі з розподілом вхідного мережевого трафіку між серверами

Конфігурування пристроїв у даній лабораторній роботі ділиться на три частини: спочатку збирається схема для режиму L3, після чого проводиться переналаштування для переходу на L2, далі

(тільки під час роботи в емуляторі) проводиться побудова відмовостійкої схеми балансування. Окремо варто зазначити, що цю лабораторну роботу варто виконувати лише після того, як добре освоєно попередні (простіші) роботи. Тут немає чіткої послідовності дій для всіх пунктів налаштування, що залишає простір для творчості у студента, даючи змогу якнайповніше ознайомитися з усіма допоміжними опціями. Частина №3 виконується тільки на емуляторі.

Під час виконання цієї лабораторної роботи на реальному обладнанні як маршрутизатор із SLB використовуйте Cisco 3640, під час роботи на емуляторі вибирайте маршрутизатор Cisco серії 7200. Також під час роботи в емуляторі в якості клієнта і серверів можна використовувати маршрутизатори.

1. Виконайте всі необхідні з'єднання і зробіть всі необхідні підготовчі налаштування (призначення IP-адрес, налаштування маршрутизації, конфігурація віртуальних мереж на комутаторі). Переконайтеся в працездатності отриманої схеми. Слід розуміти, що в реальній мережі доступу до серверів ззовні не буде з двох причин: по-перше, приватні IP-адреси не маршрутизуються глобально, а по-друге, всі "зайві" доступи закриваються за допомогою списків доступу.
2. На SLB маршрутизаторі з режиму глобальної конфігурації перейдіть у режим налаштування віртуальної ферми серверів test за допомогою команди **ip slb serverfarm *test***.
3. Дайте команду **nat server** для переведення ферми в режим роботи L3.
4. Визначтеся, який сервіс буде запущено для тестування зібраної схеми. Під час роботи в емуляторі як сервіс можна використовувати telnet до "маршрутизаторів-серверів". За допомогою команди **real 10.0.0.2 *port*** перейдіть у режим налаштування сервера 10.0.0.2 всередині цієї ферми, де *port* – номер порту, на якому працює тестований сервіс. Командою **inservice** переведіть цей сервер усередині ферми в робочий стан. Повторіть зазначену процедуру для другого сервера. У загальному випадку номер порту конкретного сервера може відрізнятися від номерів портів інших серверів і зовнішнього порту сервісу загалом.
5. Вивчіть такі допоміжні опції: **faildetect**, **maxconns**, **predictor**, **retry** і **weight**.

6. За допомогою команд привілейованого режиму **sho ip slb serverfarms**, **sho ip slb serverfarms detail**, **sho ip slb reals** вивчіть статуси серверних ферм і включених до їхнього складу серверів.

7. Використовуйте команду **ip slb vserver *test*** для переходу в режим налаштування віртуального сервера *test*.

8. За допомогою команди **serverfarm *test*** прив'яжіть раніше створену серверну ферму *test* до цього віртуального сервера.

9. Вказати адресу (у цій роботі це 1.1.1.1.1), протокол і порт зовнішнього сервісу можна за допомогою команди **virtual 1.1.1.1.1 *protocol port***.

10. Переведіть віртуальний сервер у робочий режим за допомогою команди **inservice**. Додатково вивчіть можливості таких опцій: **advertise**, **client**, **sticky** і **synguard**.

11. Вивчіть можливості переглядових команд **sho ip slb conns**, **sho ip slb reals**, **sho ip slb stats**, **sho ip slb sticky**, **sho ip slb vservers** і **sho ip slb wildcard**.

12. Запустіть тестові сервіси на тестових серверах і зробіть підключення клієнтом. На реальному обладнанні за допомогою утиліт netcps, iperf або аналогічних виміряйте продуктивність маршрутизатора з SLB у режимі L3.

13. Переконайтеся в тому, що вимкнення будь-якого з серверів не призводить до падіння сервісу.

14. На тестових серверах (або маршрутизаторах під час роботи в емуляторі) створіть інтерфейс Loopback, на який призначте адресу 1.1.1.1.

15. Створіть нову серверну ферму, проте в ній не давайте команду **nat server**.

16. Додайте сервери в нову ферму. Під час додавання сервера в L2 ферму не потрібно вказувати порт.

17. Переведіть віртуальний сервер test у неробочий режим.

18. Замініть ферму у віртуальному сервері test на нову.

19. Переведіть віртуальний сервер test у робочий режим.

20. Переконайтеся в працездатності нового режиму роботи SLB.

21. Перевірте доступність сервісу в разі вимкнення будь-якого із серверів.

22. Зберіть нову схему так, щоб паралельно з маршрутизатором SLB був встановлений новий маршрутизатор тієї ж моделі з тією ж версією IOS.

23. На старому і новому маршрутизаторах налаштуйте HSRP з іменованою групою.

24. Переведіть віртуальний сервер test у неробочий режим.

25. Скопіюйте налаштування зі старого маршрутизатора SLB на новий.

26. На обох маршрутизаторах переведіть віртуальний сервер test у робочий режим за допомогою команди **inservice standby *name***, де *name* – ім'я сконфігурованої раніше групи HSRP.

27. Переконайтеся, що статус віртуального сервера залежить від статусу вузла в налаштованій HSRP-групі.

28. Перевірте відмовостійкість нової схеми шляхом поперемінного відключення маршрутизаторів SLB.

Запитання та завдання для самоконтролю

1. Як працює load balancer?
2. Які бувають load balancer?
3. Який вебсервер використовують як балансувальник навантаження?
4. Як називається метод розподілу даних на кілька серверів?
5. Що таке глобальний балансувальник навантаження сервера?
6. Який алгоритм балансування навантаження використовується найчастіше?
7. Який алгоритм використовується балансувальником навантаження на прикладному рівні?
8. Який алгоритм найкраще підходить для балансування навантаження в хмарних обчисленнях?
9. Чи використовується Kubernetes для балансування навантаження?
10. Чи виконує Docker балансування навантаження?
11. Які існують три режими LACP?
12. Як працює Cisco LACP?

ПІСЛЯМОВА

Виконання лабораторних робіт в системі вивчення дисципліни «Мережне програмування та інтерфейси телекомунікаційних систем» сприяє набуттю базових практичних навичок у мережному програмуванні та конфігурації інтерфейсів мережевого обладнання Cisco за допомогою комп'ютерного моделювання в графічному симуляторі мереж Graphical Network Simulator 3 (GNS3) та використовуючи допоміжні команди з інтерфейсу командного рядка (CLI) IOS Cisco, що дозволяє набагато ефективніше засвоїти теоретичні викладення лабораторного практикуму та лекційного матеріалу.

Слід зауважити, що успішне виконання даного курсу лабораторних робіт неможливе без знань та практичних умінь використовувати сучасний програмний комплекс GNS3 та IOS Cisco, які є лідерами у вирішенні технічних та інженерних телекомунікаційних завдань.

Навчання за допомогою даного лабораторного практикуму сприяє формуванню у здобувачів вищої освіти професійних знань та навичок, необхідних для комп'ютерного моделювання процесів та роботи з обладнанням Cisco, які є невід'ємною частиною у побудові та адмініструванні комп'ютерних та телекомунікаційних мереж, а саме – мережевий протокол динамічного налаштування хоста DHCP; протокол дозволу адреси ARP; списки контролю доступу ACL до інтерфейсу, в якому виконується перетворення мережевих адрес NAT; протокол динамічної маршрутизації RIP; протокол маршрутизації кадрів у мережі Frame Relay; технологія віртуальної маршрутизації та переадресації VRF; побудова тунелів на маршрутизаторах Cisco з використанням інтерфейсів Loopback; технологія розподілу навантаження серверів SLB.

Опанування теоретичними та практичними основами, які викладені у даному лабораторному практикумі дозволить здобувачу вищої освіти сформувати базове розуміння процесу мережного програмування та конфігурації інтерфейсів мережевого обладнання, що є основоположним моментом в підготовці фахівців спеціальності 172 «Телекомунікації та радіотехніка».

СПИСОК ЛІТЕРАТУРИ

1. Буров Є.В. Комп'ютерні мережі: підручник. — Львів: «Магнолія 2006», 2021. — 262 с.

2. Микитишин А.Г., Митник М.М., Стухляк П.Д. Телекомунікаційні системи та мережі: навч. посіб. — Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017. — 384 с.

3. Єфіменко А.А. Основи побудови локальних комп'ютерних мереж Ethernet на базі керованих комутаторів компанії Cisco: навч. посіб. — Житомир: Житомирська політехніка, 2021. — 116 с.

4. Коробейнікова Т.І., Захарченко С.М. Технології захисту локальних мереж на основі обладнання Cisco: навч. посіб. — Львів: Видавництво Львівської політехніки, 2021. — 232 с.

5. Жураковський Б.Ю., Зенів І.О. Комп'ютерні мережі: навч. посіб. — К.: КПІ ім. Ігоря Сікорського, 2020. — 366 с.

6. Тарнавський Ю.А., Кузьменко І.М. Організація комп'ютерних мереж: підручник. — К.: КПІ ім. Ігоря Сікорського, 2018. — 259 с.

7. Борисенко В.Д., Устенко С.А., Устенко І.В. Основи комп'ютерного моделювання в інженерній діяльності: навч. посіб. — Миколаїв: МНУ, 2016. — 276 с.

ЗМІСТ

Навчальне видання

МЕРЕЖНЕ ПРОГРАМУВАННЯ ТА ІНТЕРФЕЙСИ ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

ЛАБОРАТОРНИЙ ПРАКТИКУМ
для здобувачів вищої освіти
ОС «Бакалавр» спеціальності 172
«Телекомунікації та радіотехніка»

Укладачі:

ЛАВРИНЕНКО Олександр Юрійович
АНТОНОВ Веніамін Валерійович
КУРУШКІН Віталій Євгенович

В авторській редакції

Технічний редактор *А. І. Лавринович*