І. Р. Арсенюк, А. А. Яровий

КОМП’ЮТЕРНІ МЕРЕЖІ

Частина 2

Міністерство освіти і науки, молоді та спорту України
Вінницький національний технічний університет

І. Р. Арсенюк, А. А. Яровий

КОМП'ЮТЕРНІ МЕРЕЖІ

Частина 2

Навчальний посібник

Вінниця
ВНТУ
2011

УДК 681.3
ББК 32.973.202
А85

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки, молоді та спорту України (протокол №5 від 23.12.2010 р.)

Рецензенти:
В. П. Кожем'яко, доктор технічних наук, професор
А. М. Пєтух, доктор технічних наук, професор
Л. І. Тимченко, доктор технічних наук, професор

Арсенюк, І. Р.

А85 Комп'ютерні мережі. Ч. 2 : навчальний посібник / І. Р. Арсенюк, А. А. Яровий. – Вінниця : ВНТУ, 2010. – 145 с.

В другій частині посібника розглянуто питання адресації в IP-мережах, протоколи міжмережевого і транспортного рівнів TCP/IP, основні пристрої комп'ютерних мереж, основи роботи CISCO IOS, а також детально розглянуто протоколи маршрутизації RIP, EIGRP та OSPF.

Посібник розроблений відповідно до плану кафедри та програми дисципліни „Комп'ютерні мережі", а також може бути використаний під час вивчення дисциплін „Корпоративні і глобальні комп'ютерні мережі" та „Корпоративні мережі".

УДК 681.3
ББК 32.973.202

ЗМІСТ

1 АДРЕСАЦІЯ В IP-МЕРЕЖАХ

1.1 Типи адрес стека TCP/IP

Розрізняють три типи адрес стека TCP/IP: локальні, IP-адреси та доменні імена [1 – 5].

Під *локальною (апаратною, фізичною) адресою* розуміють такий тип адреси, який використовується засобами базової технології для доставки даних в межах підмережі, що є елементом складеної інтермережі. В різних підмережах є допустимими різні мережеві технології, різні стеки протоколів, тому при створенні стека TCP/IP передбачалась наявність різних типів локальних адрес.

Для того, щоб в мережі Ethernet стала можливою локальна доставка фреймів, необхідна певна система адресації, тобто присвоєння імен комп'ютерам та інтерфейсум. Кожен вузол має унікальний спосіб самоідентифікації. Ніякі дві фізичні адреси в мережі не повинні збігатись. Фізичні адреси, які в Ethernet також називають *адресами керування доступом до середовища передавання* (Media Access Control, MAC), записані в мережевому адаптері ПК або мережевих інтерфейсах пристроїв (маршрутизаторів, комутаторів тощо).

MAC-адреса має довжину 48 бітів і записується у вигляді дванадцяти шістнадцяткових цифр (наприклад, 00-60-2F-3A-07-BC). Перші шість цифр, що задаються IEEE, ідентифікують виробника або продавця пристрою і, містять унікальний ідентифікатор організації (Organizationally Unique Identifier – OUI). Другі шість цифр містять серійний номер інтерфейсу або інше значення, що задається конкретним виробником. MAC-адресу іноді називають прошитою (Burned-In Address – BIA), оскільки вона записана в постійній пам'яті (Read-Only Memory – ROM) інтерфейсу або пристрою [1, 4 – 7]. На рис. 1.1 показано формат MAC-адреси.

Без MAC-адрес локальна мережа являла б собою лише групу ізольованих комп'ютерів, і доставка Ethernet-фреймів була б неможливою. Внаслідок цього на канальному рівні до даних верхніх рівнів додається заголовок, що містить MAC-адресу пристрою та кінцевик. Заголовок та кінцевик містять службову інформацію, призначену для канального рівня пристрою, якому направляється фрейм. Дані верхніх рівнів інкапсулюються в заголовок та кінцевик канального рівня.

IP-адреси є основним типом адрес, на основі яких мережевий рівень передає пакети між мережами. Ці адреси складаються з 4 байтів і записуються у десятково-точковій нотації (наприклад, 195.1.7.26). Кожна з частин адреси, розділених точками, називається октетом (оскільки складається з 8 біт). IP-адреса призначається адміністратором під час конфігурування комп'ютерів та маршрутизаторів [1, 7 – 9].

IP-адреси утворюють номер мережі та номер вузла. Номер мережі може бути вибраний адміністратором довільно або призначений за рекомен-

дацією спеціального підрозділу Internet (Internet Network Information Center, InterNIC), якщо мережа повинна працювати як складова частина Internet. Зазвичай постачальники послуг Internet отримують діапазони адрес у підрозділів InterNIC, а потім розподіляють їх між своїми абонентами. Номер вузла в протоколі IP призначається незалежно від локальної адреси вузла. Маршрутизатор за визначенням входить одразу до декількох мереж. Тому кожен порт маршрутизатора має власну IP-адресу. Кінцевий вузол також може входити до декількох IP-мереж В цьому випадку комп'ютер повинен мати декілька IP-адрес, за числом мережевих зв'язків. Таким чином, IP-адреса характеризує не окремий комп'ютер або маршрутизатор, а одне мережеве з'єднання.

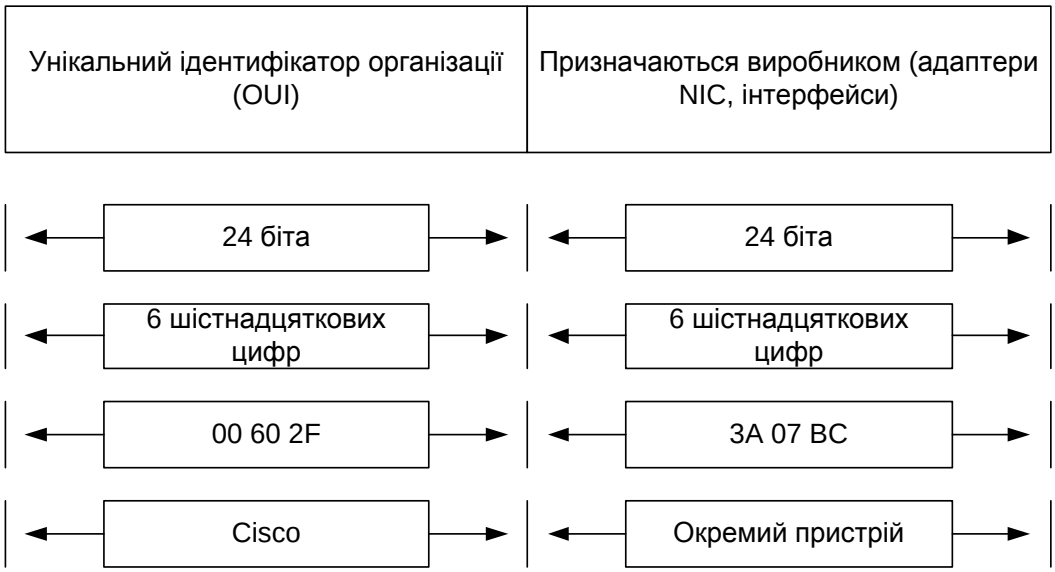


Рисунок 1.1 – Формат MAC-адреси

Символьні доменні імена. Вся мережа Internet побудована на ієрархічній системі адресації. Такий підхід дозволяє здійснювати маршрутизацію, засновану на класах адрес, а не на індивідуальних адресах. Однак використання IP-адрес не дуже зручно для користувачів. Так, різниця між адресами 194.6.197.26 та 194.6.197.62 практично непомітна, хоча обидві адреси належать до різних ресурсів мережі. Ймовірність того, що користувач може помилитьсь і ввести неправильну IP-адресу, досить висока, оскільки числова IP-адреса ніяк не пов'язана з тематикою ресурсу.

Для прив'язування вмісту Web-сторінки та її адреси була розроблена спеціальна система доменних імен (DNS). Служба DNS призначена для трансляції IP-адрес в імена і навпаки. Домен – це група вузлів, що розташовані в одній географічній області, або ж вузлів, що використовуються зі спільною метою. Доменним іменем називають рядок символів і/або цифр, і зазвичай таке ім'я відповідає цифровій IP-адресі Web-вузла в мережі Internet. На сьогодні в мережі Internet існує більш як 200 доменів верхнього

(або першого) рівня. Домени першого рівня можуть бути створені за географічною ознакою [1, 4, 10 – 12]: .ua – Україна; .ru – Росія; .us – США; .de – Німеччина; .uk – Об'єднане Королівство. Крім того, існує багато загальних доменних імен: .edu – Web-сторінки, присвячені освітнім закладам; .com – комерційні Web-вузли; .gov – урядові вузли; .org – некомерційні вузли; .net – мережеві служби.

Сервер доменних імен – мережевий пристрій, який за запитом користувача перетворює доменні імена у відповідні IP-адреси і повертає результат клієнту. Система доменних є строго ієрархічною, тому існує декілька рівнів імен і відповідно серверів DNS [1, 4, 11].

Якщо ім'я не може бути трансльовано в IP-адресу на місці, то запит передається вищому за рівнем DNS-серверу, який, в свою чергу, теж намагається визначити IP-адресу вузла. Якщо на цьому рівні DNS-сервер може здійснити перетворення імені в IP-адресу, то результат запиту повертається клієнту. Якщо ж і цей сервер не може виявити необхідний запис, то запит передається вище. Процес повторюється доти, доки не буде визначено IP-адресу запитаного вузла або доки не буде досягнуто DNS-сервера верхнього рівня. Якщо для доменного імені не знайдено відповідної адреси і на цьому рівні, то клієнту відправляється повідомлення про помилку. Всі додатки, які використовують доменні імена для надання IP-адрес, звертаються до DNS-серверів, які виконують відповідну трансляцію.

1.2 Класи IP-адрес

З метою отримання можливості опису мереж різного розміру та полегшити їх класифікацію, IP-адреси було розділено на групи, які називають класами. Така схема адресації називається *класовою*. Кожна повна 32-бітна IP-адреса поділяється на дві частини, що описують мережу та вузол. Біт або послідовність бітів на початку кожної адреси задають її клас (рис. 1.2). Є п'ять *класів IP-адрес* [1, 4].

Адреса класу А призначена для дуже великих мереж. В ній використовується тільки перший октет як ідентифікатор мережі. Три октети, що залишились, ідентифікують адресу вузлів. Перший біт в адресі класу А завжди нульовий. Враховуючи це, найменше допустиме число буде рівне 00000000 (десятковий 0), а найбільше – 01111111 (десяткове число 127). Варто відзначити, що обидва номери 0 та 127 є зарезервованими і не можуть бути використані як мережеві адреси. Будь-які адреси, що починаються з числа в діапазоні від 1 до 126 в першому октеті є адресами класу А. Мереж класу А небагато, але кількість вузлів у них може досягати $2^{24} - 2 =$ = 16 777 214 вузлів (два номери ідентифікують номера мережі та широкомовну адресу).

Мережа з номером 127.0.0.0 не може бути присвоєна мережі, оскільки зарезервована для зворотного петлевого (loopback) тестування (маршрутизатори або локальні вузли можуть використовувати його для передавання

пакетів самим собі).

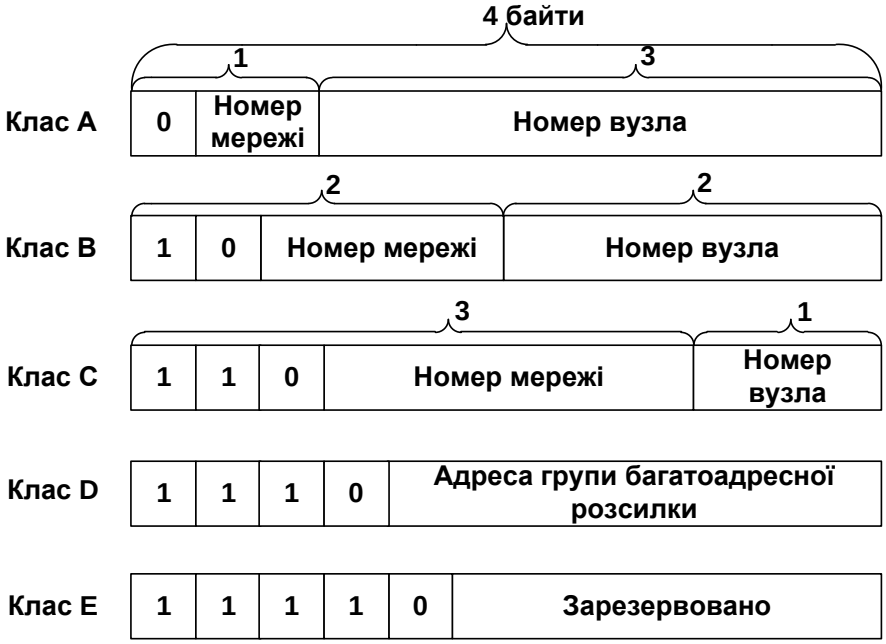


Рисунок 1.2 – Структура IP-адрес різних класів

Адреса класу B використовується для мереж середнього та великого розмірів. В IP-адресі класу B два перших октети використовується для мережевої адреси, а два других являють собою адресу вузла.

Перші два біти першого октета завжди приймають значення „1” і „0”, шість бітів, що залишились, можуть містити будь-які комбінації нулів та одиниць. Таким чином, найменше число, яке може бути використане для адрес цього класу рівне 10000000 (десяткове 128), і найбільше – 10111111 (десяткове значення рівне 191). Будь-які адреси, що містять в першому октеті числа від 128 до 191, є адресами класу B. Мережа класу B може містити максимум $2^{16} – 2 = 65\,534$ вузлів.

Адреси класу C – це найчастіше використовувані адреси, призначені для використання в малих мережах. Адреса даного класу починається з двійкової комбінації 110. Отже, найменше доступне число – 11000000 (десяткове 192), а найбільше – 1101111 (десяткове значення 223). Якщо адреса в першому октеті містить числа від 192 до 223, значить він належить до класу C. Максимальна кількість вузлів у мережі – $2^8 – 2 = 254$.

Адреси класу D були створені для реалізації в IP-адресах механізму багатоадресної розсилки. *Багатоадресною* або *груповою адресою (multicast address)* називається унікальна мережева адреса, що використовується для відправлення пакетів певним групам мережевих пристроїв. Таким чином, одна мережева станція може передавати один потік даних декільком отримувачам.

Діапазон адрес класу D, які називають багатоадресними IP-адресами також певним чином обмежений. Перші чотири біти такої адреси є 1110,

тому перший октет адрес цього класу може приймати значення від 11100000 до 11101111 або в десятковому записі від 224 до 239.

Адреси класу E також були описані в стандартах та виділені в окремий блок. Однак вони були зарезервовані проблемною групою проектування Internet (Internet Engineering Task Force – IETF) для власних дослідницьких потреб і не використовувались в мережі Internet. Перші чотири біти адрес класу E завжди одиничні. Значення першого октета знаходиться в діапазоні від 11110000 до 11111111 або від 240 до 255 – в десятковому вигляді.

Діапазони значень першого октета в IP-адресах для кожного з класів наведено в таблиці 1.1.

Таблиця 1.1 – Класи IP-адрес: діапазон значень першого октета

Клас	Перші біти	Мінімальний номер мережі	Максимальний номер мережі	Максимальна кількість мереж	Максимальна кіль-кількість вузлів у мережі
A	0	1.0.0.0	126.0.0.0	$2^7 - 2 = 126$	$2^{24} - 2$
B	10	128.0.0.0	191.255.0.0	$2^{14} = 16384$	$2^{16} - 2$
C	110	192.0.0.0	223.255.255.0	$2^{21} = 2097152$	$2^8 - 2$
D	1110	224.0.0.0	239.255.255.255	–	Багатоадресний
E	1111	240.0.0.0	255.255.255.255	–	Зарезервований

1.3 Особливі IP-адреси

Деякі адреси є особливими і не можуть належати мережевим пристроям. До них відносяться такі [1, 2]:

- IP-адреси, що складаються лише з двійкових нулів позначають адресу того вузла, котрий згенерував цей пакет. Цей режим використовується лише в деяких повідомленнях ICMP.
- IP-адреси, в полі номера мережі яких розташовані двійкові нулі. За замовчуванням вважається, що вузол призначення належить до тієї самої мережі, що й вузол, який відправив пакет.
- IP-адреси, у яких всі двійкові розряди одиничні. Пакет з такою адресою призначення повинен розсилатися усім адресам, які знаходяться в цій самій мережі, що й джерело пакета. Така розсилка називається *обмеженим широкомовним повідомленням (limited broadcast).*
- IP-адреси, у яких в полі номера вузла призначення стоять лише нулі. Такі адреси позначають номери мереж. Наприклад, 198.150.11.0.
- IP-адреси, в яких у полі номера вузла призначення стоять лише одиниці. Пакет, який має таку адресу, розсилається всім вузлам мережі із заданим номером мережі. Наприклад, пакет з адресою 198.150.11.255 доставляється всім вузлам мережі 198.150.11.0. Така розсилка називається *широкомовним повідомленням (broadcast)*.

Таким чином, реальна кількість адрес, які можна призначити пристроям мережі, на 2 менше, оскільки не можна присвоїти пристрою адресу мережі або широкомовної розсилки.

- IP-адреси, перший октет яких рівний 127. Адреса 127.0.0.1 (loopback) використовується для тестування програм та взаємодії процесів в межах однієї машини. Дані, відправлені за цією адресою, утворюють “петлю”. Дані не передаються по мережі, а повертаються до модулів верхнього рівня як такі, що тільки-но прийняті.
- IP-адреси для групової розсилки пакетів (*multicast)* (клас D). Наприклад, щоб обмінюватися повідомленнями маршрутизатори, які використовують у своїй роботі протокол маршрутизації OSPF, розсилають повідомлення за адресою 224.0.0.5. Будь-яке повідомлення, відправлене за цією адресою, буде отримане маршрутизаторами даної групи.

1.4 Застосування масок під час IP-адресації

Традиційна схема поділу IP-адреси на номер мережі та номер вузла базується на понятті класу, який визначається значеннями декількох перших бітів адреси. Саме тому, що перший байт адреси 129.54.65.3 потрапляє в діапазон 128 –191, ми можемо сказати, що ця адреса відноситься до класу B, а значить, номером мережі є перші два октети, доповнені двома нульовими байтами – 129.54.0.0, а номером вузла – 0.0.65.3.

Альтернативою цієї традиційної схеми є використання іншої ознаки, за допомогою якої можна більш гнучко встановлювати межу між номером мережі та номером вузла. Такою ознакою є *маска* – 32-розрядне число, яке використовується в парі з IP-адресою. Двійковий запис маски містить одиниці у тих розрядах, які повинні в IP-адресі інтерпретуватись як номер мережі [1, 3, 4, 14]. Оскільки номер мережі є цільною частиною адреси, одиниці у масці повинні являти собою неперервну послідовність.

Для стандартних класів IP-адрес маски мають такі значення:

Клас A – 11111111. 00000000. 00000000. 00000000 (255.0.0.0);

Клас B – 11111111. 11111111. 00000000. 00000000 (255.255.0.0);

Клас C – 11111111.11111111.11111111.00000000 (255.255.255.0).

Доволі часто зустрічається позначення маски у вигляді числа записаного після слешу, наприклад, 129.54.65.3/16. Даний запис означає, що маска для адреси 129.54.65.3 містить 16 одиниць, тобто під номер мережі відведено 16 двійкових розрядів (2 перших байти).

В основу механізму масок покладено принцип отримання номера мережі шляхом порозрядного перемноження адреси вузла і маски. Наприклад, для IP-адреси 180.34.23.134 з маскою 255.255.0.0 маємо

10110100.00100010.00010111.10000110
<u>11111111.11111111.00000000.00000000</u>
00001010.00100010.00000000.00000000 = 180.34.0.0

Супроводжуючи кожну IP-адресу маскою, можна відмовитися від понять класів адрес і зробити більш гнучкою систему адресації. Наприклад, якщо адресу 129.54.170.164 асоціювати з маскою 255.255.255.0 – номером

мережі (а точніше підмережі) буде 129.54.170.0 (а не 129.54.0.0, як це визначено системою класів), якщо з маскою 255.255.248.0 – то 129.54.168.0, а якщо з маскою 255.255.255.224 – то 129.54.170.160.

Взагалі, для виділення підмережі, частина бітів, що відповідає за нумерацію вузла повинна бути визначена як мережева. Такий механізм часто називають *позиченням* (*орендою*) *бітів*. Процес ділення завжди починається з крайнього лівого біта вузла, положення якого залежить від класу IP-адреси [1, 3, 4, 11].

Зазначимо також, що створення підмереж крім підвищення керованості, дозволяє мережевим адміністраторам обмежувати широкомовні розсилки та реалізувати механізм безпеки низького рівня в локальній мережі.

Як відомо, широкомовні пакети розсилаються усім вузлам мережі або підмережі. Коли широкомовний трафік починає витрачати значну частину доступної смуги пропускання, мережевий адміністратор може прийняти рішення щодо зменшення розмірів широкомовного домену [1, 4].

Безпека при використанні підмереж в ЛКМ реалізується завдяки тому, що доступ в інші підмережі організовується через маршрутизатори, які можуть бути налаштовані так, щоб дозволити або заборонити доступ до підмереж на основі різних критеріїв. Крім того, зовнішній світ "бачить" локальну мережу як єдину мережу, нічого не знаючи про її внутрішню будову. Крім підвищення рівня безпеки такий підхід також дозволяє зменшити ТМ і ефективно їх використовувати. Отримавши локальну адресу вузла 192.168.10.14, зовнішній світ за межами ЛКМ використовує лише об'явлену основну мережеву адресу 192.168.10.0, оскільки локальна адреса 192.168.10.14 дійсна лише в її межах.

Деякі організації виявили також, що використання механізму виділення підмереж може принести додаткові прибутки за рахунок продажу чи передачі в оренду адреси, що раніше не використовувались [1, 4].

Вибір необхідної кількості бітів для створення підмережі залежить від потрібної максимальної кількості її вузлів. Для визначення маски підмережі на основі кількості доступних підмереж і вузлів можна використати такі вирази: 2^n – кількість використовуваних підмереж, де n – кількість позичених з вузлової частини бітів; $2^m - 2$ – кількість доступних вузлів, де m – кількість бітів вузлової частини, що лишились (кількість бітів вузлової частини $v = n + m$).

Наприклад, при запозиченні трьох бітів з вузлової частини мережі класу С для адресації вузлів будуть використовуватися 5 бітів, тому кількість вузлів у кожній підмережі рівне $2^5 - 2 = 30$, а максимальна кількість підмереж складе $2^{(v-5)} = 3$ (тут $v = 8$).

Механізм масок широко використовується в IP-маршрутизації, причому маски можуть використовуватись з різною метою. За їх допомогою адміністратор може структурувати свою мережу, не вимагаючи від постачальника послуг додаткових номерів мереж. На основі цього ж механізму по-

стачальники послуг можуть об'єднати адресні простори декількох мереж шляхом введення так званих "префіксів" з метою зменшення об'єму таблиць маршрутизації та підвищення за рахунок цього продуктивності роботи маршрутизаторів.

1.4.1 Застосування масок постійної довжини

Розглянемо на конкретному прикладі процес сегментації мережі із застосування масок постійної довжини (технологія FLSM – Fixed-Length Subnet Masking) для структуризації мережі [1].

Наприклад, адміністратору мережі треба організувати чотири мережі, а постачальником послуг виділено лише один номер мережі класу B 129.144.0.0. Дана задача може бути розв'язана за допомогою застосування масок постійної довжини. Обрана маска в даному випадку буде 255.255.192.0 (оскільки для адресації чотирьох наших мереж треба запозичити з бітів номера вузла два старших розряди третього октету). Після накладанні такої маски на видану адресу, кількість розрядів, що відповідали номеру мережі збільшились з 16 до 18. Два додаткові біти (№ 17 і 18) у номері мережі часто інтерпретують як номери підмереж.

В результаті застосування масок схема розподілу адресного простору прийняла вигляд, як показано у таблиці 1.2 [3].

Таблиця 1.2 – Поділ адресного простору мережі класу B з використанням технології FLSM

1 октет	2 октет	3 октет			4 октет	Опис мережі
Поле номера мережі класу B		Номер підмережі		Поле адреса вузла		
129	44					
10000001	00101100	0	0	000000	00000000	Мережа 129.44.0.0
…	…	…	…	…	…	Маска 255.255.192.0
10000001	00101100	0	0	111111	11111111	Вузлів $2^{14} – 2$
10000001	00101100	0	1	000000	00000000	Мережа 129.44.64.0
…	…	…	…	…	…	Маска 255.255.192.0
10000001	00101100	0	1	111111	11111111	Вузлів $2^{14} – 2$
10000001	00101100	1	0	000000	00000000	Мережа 129.44.128.0
…	…	…	…	…	…	Маска 255.255.192.0
10000001	00101100	1	0	111111	11111111	Вузлів $2^{14} – 2$
10000001	00101100	1	1	000000	00000000	Мережа 129.44.192.0
10000001	00101100	1	1	000000	00000001	Маска 255.255.192.0
10000001	00101100	1	1	000000	00000010	Вузлів $2^{14} – 2$
…	…	…	…	…	…	
Невикористані адреси ($2^{14} – 4$)						
10000001	00101100	1	1	111111	11111111	

Мережа, отримана в результаті такої структуризації наведена на рис. 1.3. Весь трафік із зовнішнього світу потрапляє у наші внутріш-

ні мережі через маршрутизатор М1. Для структуризації інформаційних потоків у внутрішній мережі встановлено додатковий маршрутизатор М2.

В кожній з чотирьох підмереж може бути до $2^{14} - 2$ вузлів. Якщо такої кількості вузлів немає – адресний простір буде невикористаним.

Ззовні дана мережа виглядає як звичайна мережа класу В, а на локальному рівні – це складена мережа, що має підмережі. Це також має ще одну перевагу, оскільки дозволяє приховати від зовнішнього спостереження структуру такої мережі.

Даний підхід дозволяє у порівнянні з класовою адресацією ефективніше розподіляти адресний простір. Проте, як ми побачили в даному випадку, залишилось незадіяними $2^{14} - 4$ адрес.

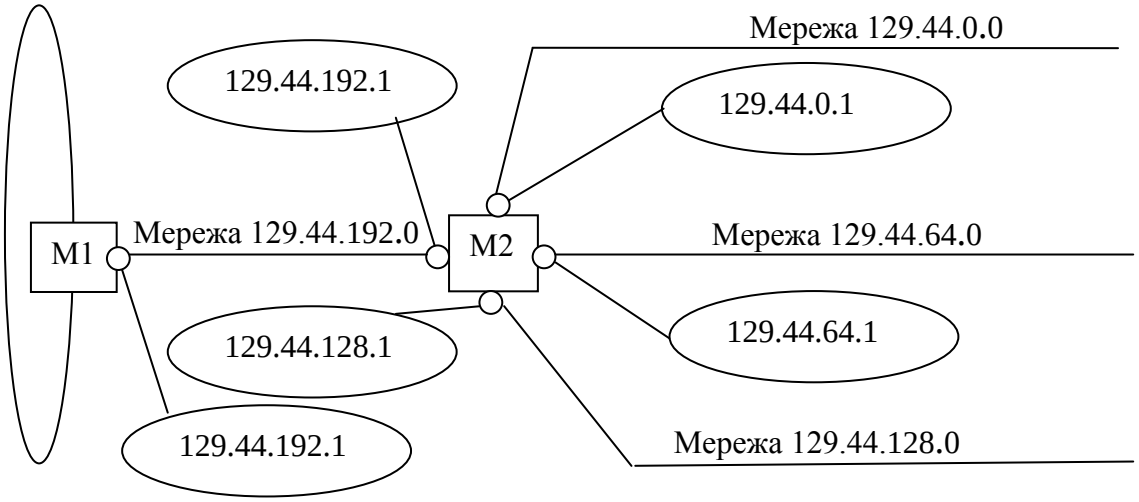


Рисунок 1.3 – Мережа, структурована із застосуванням FLSM

Цілком зрозуміло, що у випадку дефіциту IP-адрес таке неефективне використання IP-адрес також є неприпустимим. Для вирішення даної проблеми може бути використана технологія застосування масок змінної довжини (VLSM – Variable Length Subnet Masking).

1.4.2 Застосування масок змінної довжини

Застосування масок змінної довжини забезпечує заощадливіше використання адресного простору. Приклад розподілу адресного простору з масками змінної довжини для попереднього прикладу наведено у таблиці 1.3 [1]. Основна ідея застосування VLSM полягає в тому, що на відміну від технології FLSM, для кожної підмережі маска обчислюється окремо.

Зауважимо, що застосування технології VLSM дозволяє не лише заощадити простір IP-адрес, а й в деяких випадках дозволяє розв’язати задачу сегментації мережі, яка була б неможливою у випадку застосування технології FLSM.

Розглянемо такий приклад. Нехай провайдер виділив номер мережі класу С: 210.100.45.0 і адміністратору треба організувати чотири підмережі з кількістю вузлів 120, 58, 28 та 4. Зазначимо, що дану задачу принципово неможливо розв'язати з використанням масок постійної довжини. Це пов'язано з тим, що маску підмережі слід вибирати для найбільшої кількості вузлів (для того, щоб кожен вузол в кожній підмережі мав унікальну адресу). Отже в нашому випадку, маска підмережі буде 255.255.255.128. Це говорить про те, що з восьми бітів IP-адреси, які відповідають за нумерацію вузлів один біт виділено для нумерації підмереж. Отже ми можемо отримати лише дві підмережі замість бажаних чотирьох.

Обчислення маски окремо для кожної підмережі дозволяє розв'язати поставлену задачу (табл. 1.4).

Таблиця 1.3 – Поділ адресного простору мережі класу В з використанням технології VLSM

1 октет	2 октет	3 октет	4 октет	Опис мережі
Поле номера мережі класу В		Номер підмережі	Поле адреса вузла	
129	44			
10000001 ... 10000001	00101100 ... 00101100	0 0000000 ... ... 0 1111111	00000000 ... 11111111	Мережа 129.44.0.0 Маска 255.255.128.0 Вузлів $2^{15}-2$
10000001 ... 10000001	00101100 ... 00101100	1 0 000000 ... ... 1 0 111111	00000000 ... 11111111	Мережа 129.44.128.0 Маска 255.255.192.0 Вузлів $2^{14}-2$
10000001 ... 10000001	00101100 ... 00101100	1 1 0 0 0 0 0 0 ... 1 1 0 0 0 0 0 0	0 0 0 0 0 0 00 ... ... 0 0 0 0 0 0 11	Мережа 129.44.192.0 Маска 255.255.255.248 Вузлів $2^{4}-2$
Діапазон адрес ($2^{13}-4$) вільний для утворення нових мереж				
10000001 ... 10000001	00101100 ... 00101100	1 1 1 00000 ... ... 1 1 1 11111	00000000 ... 11111111	Мережа 129.44.224.0 Маска 255.255.224.0 Вузлів $2^{13}-2$

Таблиця 1.4 – Поділ адресного простору мережі класу С з використанням технології VLSM

1 октет	2 октет	3 октет	4 октет		Опис мережі
Поле номера мережі класу С			Поле адреси вузла		
210	100	45			
210	100	45	0 .. 0	0000000 ... 1111111	Мережа 210.100.45.0 Маска 255.255.255.128 Вузлів $2^7 - 2 = 126$
210	100	45	1 0 ... 1 0	000000 ... 111111	Мережа 210.100.45.128 Маска 255.255.255.192 Вузлів $2^6 - 2 = 62$
210	100	45	1 1 0 ... 1 1 0	00000 ... 11111	Мережа 210.100.45.192 Маска 255.255.255.224 Вузлів $2^5 - 2 = 30$
210	100	45	1 1 1 0 0 ... 1 1 1 0 0	000 ... 111	Мережа 210.100.45.224 Маска 255.255.255.224 Вузлів $2^3 - 2 = 6$

1.5 Протоколи дозволу IP-адрес

Для взаємодії пристроїв необхідно, щоб у пристрою-передавача була IP- та MAC-адреса отримувача. Коли один з пристроїв намагається встановити зв'язок з іншим, що має відому IP-адресу, йому слід визначити MAC-адресу отримувача (якщо отримувач знаходиться не в локальному мережевому сегменті, існує потреба у визначенні фізичних адрес проміжних пристроїв до пункту призначення). Це потрібно для того, щоб інкапсульовані у фрейми пакети могли досягти свого адресата.

Набір протоколів TCP/IP має в своєму складі спеціальний протокол, який називається *ARP (Address Resolution Protocol – протокол перетворення адрес)*, який дозволяє автоматично отримати MAC-адресу [1, 4]. На рис. 1.4 проілюстровано процес, що дозволяє визначити MAC-адресу, пов'язану з відомою IP-адресою.

Деякі пристрої зберігають спеціальні ARP-таблиці, в яких міститься інформація про MAC- та IP-адреси інших пристроїв, під'єднаних до тієї ж локальної мережі. ARP-таблиці дозволяють встановити однозначну відповідність між IP- та MAC-адресами. Такі таблиці зберігаються у певних областях оперативної пам'яті і обслуговуються автоматично на кожному із мережевих пристроїв (табл. 1.5 та 1.6).

Таблиця 1.5 – Запис в ARP-таблиці

IP-адреса	Фізична адреса	Тип
68.2.168.1	00-50-57-00-76-84	Динамічний

В окремих випадках доводиться створювати ARP-таблиці вручну. Зверніть увагу, що кожен комп'ютер в мережі підтримує свою власну ARP-таблицю.

Куди б не передавались дані мережевим пристроєм, для їх пересилання завжди використовується інформація, що зберігається в ARP-таблиці (рис. 1.5). У випадку, коли відправник не може отримати шукану фізичну адресу зі своєї власної ARP-таблиці, ініціюється процес, який називається ARP-запитом [1, 4, 11].

Таблиця 1.6 – ARP-таблиця для адреси 198.150.11.36

MAC-адреса	IP-адреса
FE:ED:F9:44:45:66	198.150.11.34
DD:EC:BC:AB:04:AC	198.150.11.33
DD:EC:BC:00:94:D4	198.150.11.35
FE:ED:F9:23:44:EF	198.150.11.36

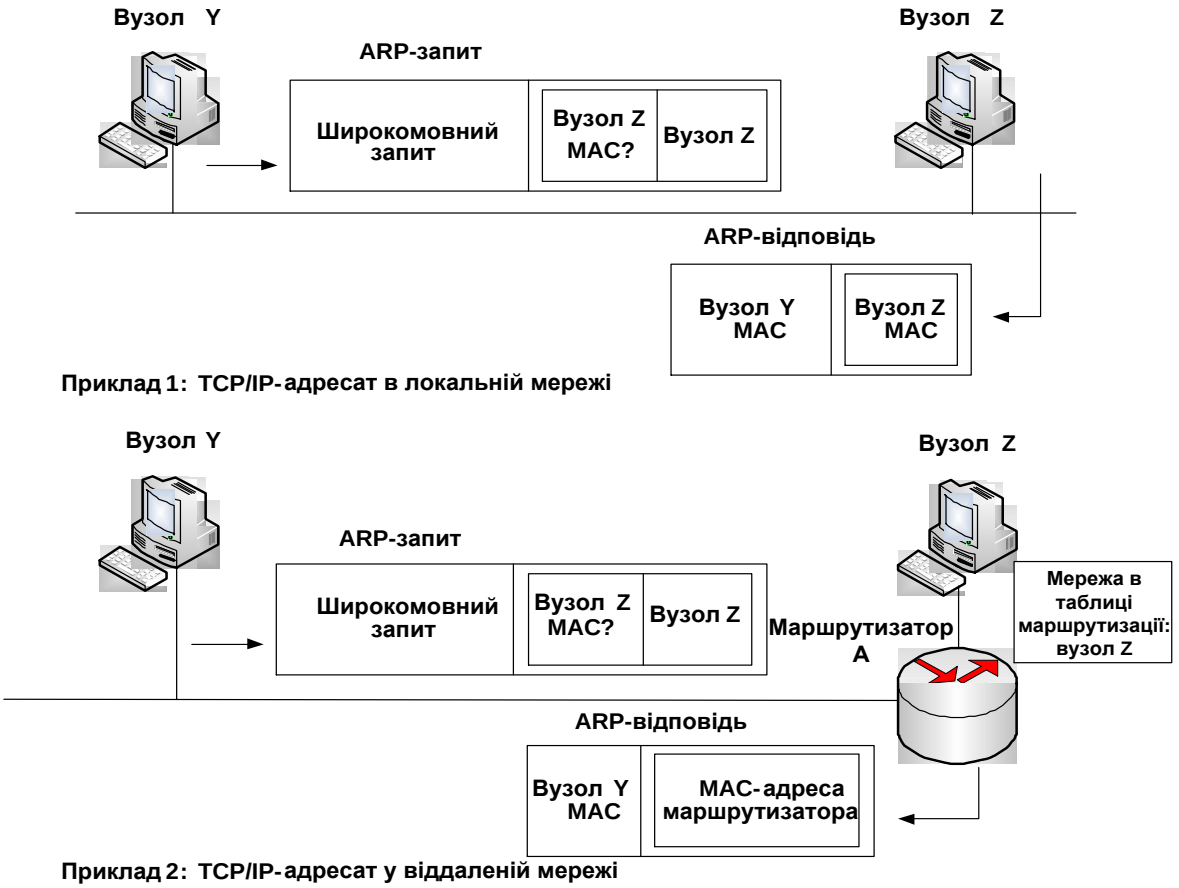


Рисунок 1.4 – Отримання MAC-адреси на основі IP-адреси

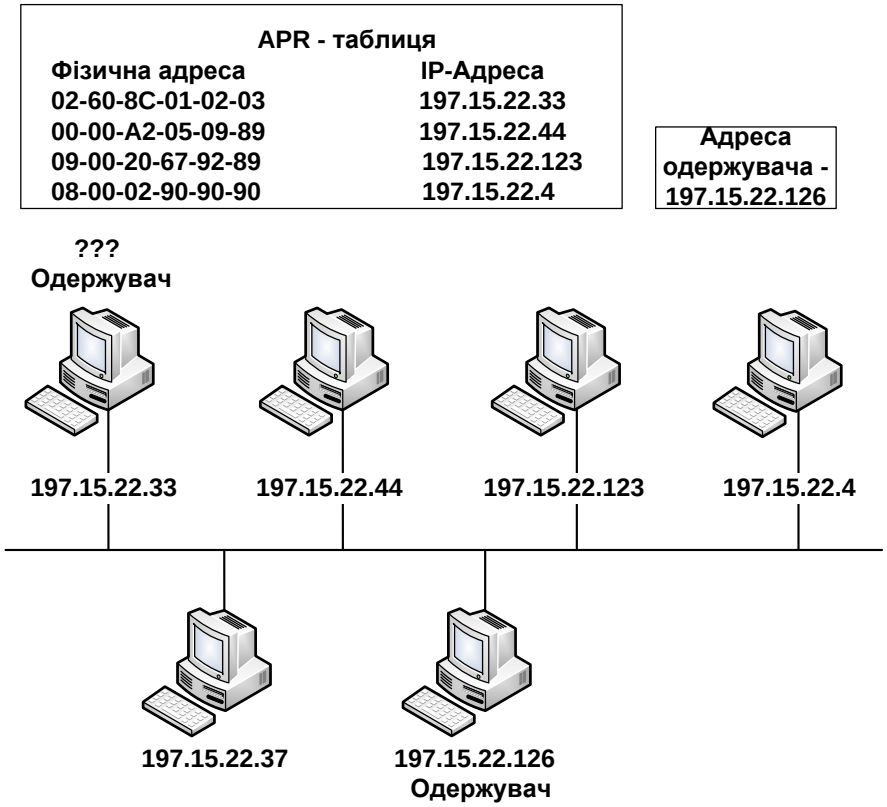


Рисунок 1.5 – ARP-таблиця для невеликої мережі

ARP-запит дозволяє вузлу визначити MAC-адресу отримувача. Вузол створює фрейм ARP-запиту і розсилає його всім мережевим пристроям. Фрейм ARP-запиту складається з двох частин:

- заголовку фрейма;
- повідомлення ARP-запиту.

Для того, щоб усі пристрої могли отримати ARP-запит, використовується широкомовна MAC-адреса (така адреса містить у всіх бітах одиничні значення: FF-FF-FF-FF-FF-FF). Оскільки фрейми ARP-запиту передаються в широкомовному режимі, всі мережеві пристрої, під'єднані до ЛКМ можуть отримувати такі фрейми і передавати інкапсульовану в них інформацію протоколам вищих рівнів для наступного оброблення. Якщо IP-адреса пристрою збігається з IP-адресою отримувача, в широкомовному ARP-запиті, то цей пристрій відповідає відправнику, повідомляючи свою MAC-адресу. Таке повідомлення називається ARP-відповіддю.

Після отримання ARP-відповіді пристрій-відправник широкомовного ARP-запиту вилучає MAC-адресу з поля апаратної адреси відправника та оновлює свою ARP-таблицю. Тепер цей пристрій може належним чином адресувати пакети, використовуючи як MAC-, так і IP-адресу. Отримана інформація використовується для інкапсуляції даних на другому та третьому рівнях перед їх відправленням мережею. Коли дані досягають пункту призначення, на канальному рівні проводиться перевірка на відповідність адреси, відкидається канальний заголовок, який містить MAC-адреси

і дані передаються на мережевий рівень. На мережевому рівні перевіряється відповідність власної IP-адреси та IP-адреси отримувача, що міститься в заголовку третього рівня. На мережевому рівні відкидається IP-заголовок, і інкапсульовані дані передаються на наступний рівень моделі OSI – транспортний (рівень 4). Подібний процес повторюється доти, доки дані, що залишилися частково розпаковані, не досягнуть додатку (рівень 7), в якому буде прочитана частина даних користувача.

Варто знати, що існує також протокол, який вирішує обернену задачу – знаходження IP-адреси за відомою локальною адресою. Він називається реверсивним RARP (Reverse Address Resolution Protocol, RARP) і використовується, наприклад, при стартуванні бездискових станцій, які не знають в початковий момент своєї IP-адреси, але знають адресу свого мережевого адаптера [1, 4, 7].

1.6 Контрольні запитання

1. Наведіть типи адрес стека TCP/IP з відповідними прикладами.
2. Поясніть призначення та застосування MAC-адрес. Яка структура MAC-адрес? Наведіть кілька прикладів MAC-адрес.
3. Поясніть призначення та застосування IP-адрес. Наведіть кілька прикладів IP-адрес.
4. Які адреси називають символьними? Наведіть кілька прикладів таких адрес та поясніть їх призначення.
5. Поясніть, яким чином символьні адреси перетворюються на IP-адреси. Який сенс такого перетворення?
6. Поясніть з якою метою використовується класова схема IP-адресації.
7. Перерахуйте та охарактеризуйте класи IP-адрес.
8. Поясніть, як визначити діапазон номерів мереж класів A, B та C.
9. Поясніть, як визначити максимальну кількість IP-адрес у мережах класів A, B та C.
10. Поясніть, з якою метою використовуються адреси класів D та E. Наведіть діапазони адрес цих класів.
11. Охарактеризуйте особливі IP-адреси та поясніть їх призначення. Наведіть кілька відповідних прикладів таких адрес.
12. Поясніть призначення групових IP-адрес. Чим групові адреси відрізнаються від широкомовних?
13. Поясніть з якою метою використовується широкомовне повідомлення і обмежене широкомовне повідомлення. В чому відмінність між ними?
14. Поясніть призначення IP-адреси 127.0.0.1.
15. Поясніть призначення масок під час IP-адресації з відповідними прикладами. Наведіть стандартні маски мереж класів A, B, C у десятково-точковій нотації та у вигляді числа, записаного через слеш.

16. Наведіть приклади українських доменів першого та другого рівнів.

17. Поясніть, до яких класів належать IP-адреси 179.54.65.3/16, 15.2.3.4/8, 199.78.65.124/24 та 224.0.0.10.

18. Поясніть сутність технології FLSM. Які переваги та недоліки її застосування?

19. Наведіть приклад сегментації мережі на п’ять підмереж із застосуванням масок постійної довжини. Визначте загальні втрати IP-адрес для цього прикладу.

20. Поясніть сутність технології VLSM. Які переваги та недоліки її застосування у порівнянні з технологією FLSM?

21. Наведіть приклад сегментації мережі на п’ять підмереж із застосуванням масок змінної довжини та визначте загальні втрати IP-адрес. Які втрати були б у випадку застосування масок постійної довжини?

22. Наведіть приклад сегментації мережі, який принципово не може бути реалізований із застосуванням масок постійної довжини, а лише змінної довжини.

23. Поясніть призначення протоколу ARP.

24. Продемонструйте на прикладі функціонування протоколу ARP.

25. Наведіть складові фрейму ARP-запиту.

26. Поясніть призначення протоколу RARP.

1.7 Завдання

1. Визначте номер мережі та номер вузла для таких IP-адрес:

а) 10.145.23.57/16;
б) 213.4.188.72/24;
в) 130.39.137.43/18;
г) 200.15.130.15/25;
д) 111.45.123.59/19;
е) 212.40.184.72/24;
ж) 120.77.23.105/27;
и) 195.15.73.100/28;
к) 143.93.125.67/23;
л) 152.14.100.224/26;
м) 67.12.193.35/30;
н) 194.15.100.61/25.

2. Вкажіть які з нижченаведених адрес є широкомовними. Відповідь мотивуйте.

а) 120.34.34.10/30;
б) 132.42.157.255/23;
в) 199.105.1.56/25;
г) 6.145.23.131/30;
д) 101.37.123.63/25;
е) 134.39.155.167/23;
ж) 197.15.73.99/28;
и) 141.93.124.255/23;
к) 210.44.188.72/24;
л) 158.4.1.123/29;
м) 76.12.193.83/30;
н) 143.104.10.24/26;

3. Вкажіть які з нижченаведених адрес є номерами мереж. Відповідь мотивуйте.

а) 217.4.188.0/24;
б) 197.34.73.100/28;
в) 163.9.125.67/23;
г) 221.10.130.15/25;
д) 132.3.192.0/18;
е) 199.4.83.2/24;
ж) 105.65.123.59/19;
и) 119.187.23.105/27;
к) 63.23.193.252/30;
л) 17.234.255.176/28;
м) 149.14.100.224/26;
н) 202.73.12.28/25.

4. Вкажіть які з нижченаведених адрес можна присвоювати хостам. Відповідь мотивуйте.

а) 213.4.134.160/26;
б) 120.77.23.133/30;
в) 130.39.163.255/22;
г) 41.145.255.254/16;
д) 195.15.73.100/28;
е) 207.45.73.222/27;
ж) 200.15.130.15/25;
и) 152.14.100.224/27;
к) 143.93.125.67/23;
л) 111.45.123.59/17;
м) 94.12.254.255/16;
н) 140.34.5.160/27.

5. Використовуючи маски постійної довжини виконайте сегментацію мережі для даних, наведених у таблиці 1.7. Обґрунтуйте вибір маски підмережі. Враховуючи, що кількість сегментів п’ять, для кожного сегмента наведіть:

- номер підмережі;
- IP-адресу широкомовного повідомлення;
- діапазон IP-адрес;
- кількість невикористаних IP-адрес.

Визначте максимально можливу кількість підмереж для Вашого випадку.

Таблиця 1.7 – Варіанти завдань для задачі № 1

Номер варіанта	Адреса мережі	Максимальна кількість вузлів у сегменті	Номер варіанта	Адреса мережі	Максимальна кількість вузлів у сегменті
1	2	3	4	5	6
1	125.0.0.0	260	16	20.0.0.0	240
2	142.56.0.0	210	17	144.60.0.0	50
3	195.24.15.0	14	18	195.3.20.0	9
4	143.28.0.0	121	19	9.0.0.0	300
5	152.110.0.0	86	20	130.13.0.0	200
6	18.0.0.0	256	21	18.0.0.0	512
7	64.0.0.0	230	22	129.66.0.0	110
8	138.31.0.0	130	23	212.4.1.0	10

Продовження таблиці 1.7

1	2	3	4	5	6
9	173.141.0.0	60	24	150.8.0.0	263
10	197.26.123.0	6	25	170.100.0.0	64
11	47.0.0.0	180	26	164.18.0.0	70
12	172.60.0.0	38	6	100.0.0.0	128
13	193.23.32.0	12	7	155.10.0.0	170
14	39.0.0.0	280	8	13.31.0.0	140
15	137.38.0.0	150	9	133.141.0.0	80

6. Використовуючи маски змінної довжини виконайте сегментацію мережі, наведеної на рис. 1.6. Обґрунтуйте вибір маски для кожної підмережі. Варіанти завдань наведено у таблиці 1.8. Для кожного сегмента мережі наведіть:

- номер підмережі;
- IP-адресу широкомовного повідомлення;
- діапазон IP-адрес;
- кількість невикористаних IP-адрес.

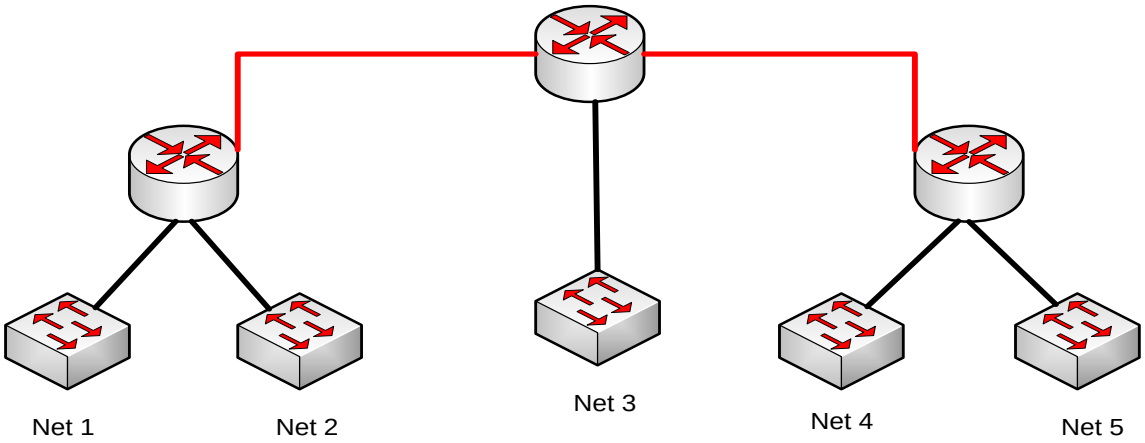


Рисунок 1.6 – Комп’ютерна мережа для завдання № 2

Таблиця 1.8 – Варіанти завдань для задачі № 2

Номер варіанта	Адреса мережі	Максимальна кількість вузлів у сегменті				
1	2	3				
		Net1	Net2	Net3	Net4	Net5
1	196.87.105.0	15	10	10	5	5
2	155.56.0.0	120	80	60	40	10
3	195.24.15.0	100	40	13	10	4
4	203.28.115.0	120	60	14	6	6

Продовження таблиці 1.8

1	2	3				
		Net1	Net2	Net3	Net4	Net5
5	21.0.0.0	130	130	80	62	40
6	218.101.24.0	85	12	12	50	5
7	194.200.15.0	45	18	100	60	3
8	138.31.0.0	120	28	30	150	100
9	205.141.67.0	70	56	27	14	4
10	197.26.123.0	12	10	58	20	7
11	193.14.95.0	50	54	30	10	10
12	21.0.0.0	160	125	100	60	25
13	193.40.50.0	40	30	14	14	3
14	220.80.15.0	20	40	13	10	6
15	145.110.0.0	28	70	20	14	4
16	197.47.83.0	16	32	8	5	5
17	206.61.44.0	100	27	54	4	8
18	198.40.10.0	60	30	2	16	12
19	176.76.0.0	120	160	142	100	90
20	202.62.23.0	12	10	58	32	15
21	213.14.5.0	57	33	55	2	2
22	116.0.0.0	256	112	170	63	18
23	166.24.0.0	140	125	55	110	16
24	203.100.150.0	50	2	14	23	40
25	215.14.18.0	13	15	50	40	5
26	176.76.0.0	220	100	12	80	32
27	200.200.203.0	9	18	25	32	6
28	211.104.50.0	7	40	25	10	2
29	33.0.0.0	256	60	300	63	180
30	138.49.0.0	30	50	100	128	160

2 МЕРЕЖЕВЕ АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ

Устаткування, безпосередньо приєднане до мережі, є *мережевим пристроєм.* Всі мережеві пристрої можна віднести до таких груп [2, 4, 16]:

- пристрої кінцевого користувача (*кінцеві вузли, станції*) – це пристрої, що зв'язують користувачів з мережею, фізично під'єднуючись до неї за допомогою *мережевого адаптеру* або *плати мережевого інтерфейсу (Network Interface Card, NIC).* До цієї групи входять комп'ютери, принтери, сканери та інші пристрої, які виконують функції, безпосередньо призначені для користувача мережі;
- мережеві пристрої – це пристрої, що приєднані до пристроїв кінцевих користувачів і дозволяють їм здійснювати зв'язок. Вони забезпечують транспортування даних між пристроями кінцевих користувачів, продовжують і об'єднують кабельні з'єднання, перетворюють дані з одного формату в інший та керують передаванням даних. Прикладами таких пристроїв є повторювачі, концентратори, мости, комутатори і маршрутизатори.

2.1 Плати мережевих адаптерів

Мережевий адаптер Ethernet, використовується для під'єднання персонального комп'ютера до мережі [1, 4, 16]. Він взаємодіє з мережею через кабель (або радіохвилі в безпровідних технологіях зв'язку), а з комп'ютером – через гніздо розширення. При виборі мережевого адаптера слід врахувати такі чинники [4]:

- тип мережі. Різні типи мереж вимагають різних мережевих адаптерів (наприклад, адаптери Gigabit-Ethernet розроблені для використання в локальних мережах Ethernet);
- тип середовища передавання даних – тип порту або мережевого роз'єму, використовуваного для під'єднання до різних середовищ передавання даних (наприклад: скручена пара, ВОК або безпровідна мережа);
- тип системної шини (є різні типи системної шини, наприклад, PCI).

2.2 Повторювачі

ЛКМ об'єднують між собою багато пристроїв різних типів. Як згадувалося вище, існує багато різних середовищ передавання даних, кожне з яких має свої переваги і недоліки. Наприклад, одним з недоліків кабелю категорії 5 UTP (який на сьогоднішній день є найчастіше використовуваним) є обмеженням на його довжину. Так, максимальна довжина кабелю UTP для одного сегмента мережі складає 100 м. Якщо потрібна більша відстань, то слід використовувати повторювачі. У більшості сучасних мереж Ethernet замість повторювачів використовуються комутатори, іноді ще можна зустріти і концентратори (багатопортові повторювачі).

Призначення повторювачів полягає в регенерації і ресинхронізації мережевих сигналів на бітовому рівні для того, щоб вони могли пройти біль-

шу відстань по середовищу, в якому виконується передавання.

Повторювачі зазвичай використовуються в тих випадках, коли в мережі є дуже багато вузлів або довжини наявного кабелю недостатньо для досягнення віддалених робочих станцій.

2.3 Функції, призначення та класифікація концентраторів

Концентратори (hub) – є багатопортовими повторювачами. Використання концентратора перетворює мережеву топологію з шинної в зіркоподібну. Концентратори належать до одного з таких типів [3]:

- активний концентратор повинен бути під'єднаний до джерела зовнішнього живлення, оскільки йому потрібна енергія для посилення вхідного сигналу перед передаванням його на зовнішні порти;
- інтелектуальний концентратор (*smart hubs*) – функціонує як звичайний концентратор, проте, має вбудований мікропроцесор і має можливості діагностики. Він дорожчий за звичайний концентратор, проте, корисний в аварійних ситуаціях;
- пасивний концентратор виступає виключно як точка фізичного з'єднання пристроїв. Такий концентратор не перевіряє трафік, що проходить через нього, і не виконує ніяких дій з потоками даних; він не підсилює і не очищає сигнал, а лише надає доступ до загальної шини.

2.4 Мости та комутатори

2.4.1 Основи функціонування мостів

Як вказувалось в розділі 2.3 велику ЛКМ часто потрібно поділяти на менші, легкокеровані сегменти. При цьому пристроями використовуваними для з'єднання мережевих сегментів, можуть бути мости, комутатори, маршрутизатори і шлюзи. Мости і комутатори функціонують на канальному рівні моделі OSI. Функція моста полягає у визначенні того, чи потрібно відправляти сигнали, що надійшли на один з його портів в інший сегмент мережі. Мости можуть також бути використані для з'єднання мереж, що використовують різні протоколи або різні середовища передавання [4, 14, 16].

Під час роботи міст використовує метод прозорого перенаправлення. Цей метод описаний у специфікації IEEE 802.1d, яка визначає п'ять процесів оброблення фрейму, при проходженні через коммутатор (рис. 2.1).

1. Перенаправлення фреймів (forwarding).
2. Лавинне передавання фреймів (flooding).
3. Фільтрування фреймів (filtering).
4. Комутація з вивченням топології або з самонавчанням (learning).
5. Застарівання таблиці MAC-адрес (aging).

Рисунок 2.1 – Кроки методу прозорого мостового перенаправлення

Коли міст отримує фрейм, він порівнює MAC-адресу відправника з адресною таблицею, що є в нього, для визначення того, чи слід відфільтрувати цей фрейм (відкинути), надіслати його лавинним способом або у визначений сегмент мережі.

Прийняття цього рішення відбувається таким чином [4, 7, 16]:

- якщо пристрій-одержувач знаходиться в тому ж сегменті, з якого цей фрейм був отриманий, то міст запобігає його передаванню в інші сегменти. Цей процес називається *фільтруванням (filtering);*
- якщо пристрій-одержувач знаходиться в іншому сегменті і його адреса присутня в адресній таблиці, то міст пересилає фрейм у відповідний сегмент;
- якщо пристрій-одержувач відсутній в таблиці адрес (тобто "невідомий" мосту) або фрейм є широкомовний чи багатоадресний – то міст розсилає фрейм у всі сегменти за винятком того, звідки був отриманий фрейм. Таку поведінку називають *лавинним розсиланням.*

Стратегічно правильно встановлений міст може значно збільшити продуктивність мережі.

2.4.2 Режими комутації

Під час роботи мостів передбачається кілька режимів комутації. Вони відрізняються моментом, коли слід комутувати фрейм [4, 5, 14]. Найчастіше використовуються три режими комутації:

- *з проміжним зберіганням* (Store-and-Forward);
- *наскрізий* (Cut-through);
- *з контролем фрагментів* (Fragment free).

Моменти часу, в які відбувається комутація для цих режимів наведені на рис. 2.2. Кожен з цих режимів має свої переваги та недоліки. Наприклад, наскрізна комутація характеризується максимальною швидкодією, але неможливістю аналізу фактів викривлень фреймів. Комутація з проміжним зберіганням навпаки, характеризується меншою швидкістю, проте можливістю аналізу фремів. При цьому, якщо фрейм викривлений – він відкидається комутатором, а отже, заощадливіше використовується пропускна спроможність каналів. Комутація з контролем фрагментів займає проміжне місце за рівнем контролю і швидкістю і дозволяє відкидати лише ті фрейми, розмір яких менший за 64 байти і які виникають внаслідок колізій. Ряд мостів підтримують усі ці режими і дозволяють автоматично вибирати їх в залежності від особливостей роботи мережі [14].

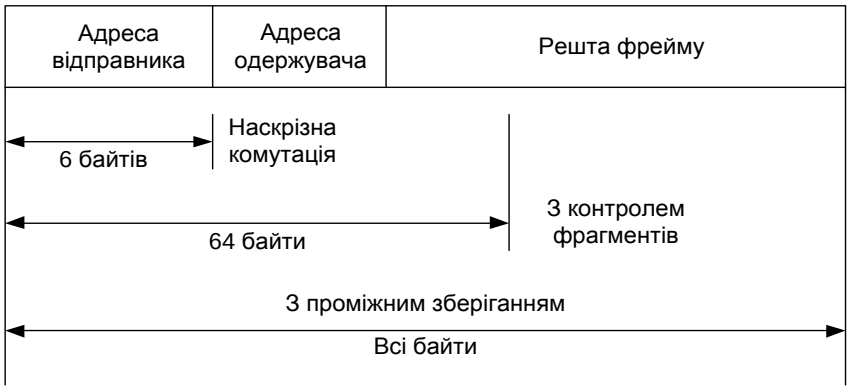


Рисунок 2.2 – Режими комутації

Ще комутація може бути *симетричною* та *асиметричною*. Перша забезпечує комутовані з'єднання між портуми з однаковою шириною смуги пропускання (наприклад, всі порти 100 Мбіт/с). Асиметрична комутація забезпечує комутовані з'єднання між портуми з різними значеннями смуги пропускання (наприклад, 100 Мбіт/с і 1000 Мбіт/с).

Асиметрична комутація використовується у випадку наявності великих мережевих потоків типу "клієнт-сервер", коли багато користувачів одночасно обмінюються інформацією з сервером. Це, очевидно, потребує більшої смуги пропускання для того порту коммутатора, до якого під'єднано сервер для перенаправлення потоку даних з порту 1000 Мбіт/с на порт 100 Мбіт/с без переповнення на останньому, асиметричний комутатор повинен мати буфер пам'яті. Асиметричний комутатор також потрібен для забезпечення більшої смуги пропускання каналів між комутаторами, що здійснюють вертикальні крос-з'єднання або каналів між сегментами магістралі [4, 14].

2.4.3 Проблеми у роботі мережі на основі мостів

У мережах на основі мостів з надлишковими елементами, які використовуються для підвищення надійності мережі (без застосування протоколу STP) можливе виникнення ряду проблем, до яких відносять:

- широкомовні шторми;
- викривлення інформації у таблицях MAC-адрес комутаторів.

Розглянемо детальніше ці проблеми.

Широкомовний шторм – це процес нескінченного циркулювання широкомовних повідомлень у петлях мережі на основі комутаторів. Такі шторми обумовлені тим, що мости надсилають широкомовні повідомлення в усі порти. За виключенням того, від якого це повідомлення було отримано. Так, коли станція А, надсилає широкомовний фрейм у мережу, він потрапляє до портів 1/1 обох мостів (рис. 2.3). Далі ці мости надсилають один одному через порти 1/2. І так далі. Отже, у мережі будуть нескінченно циркулювати широкомовні фрейми в обох напрямках (за та проти годинникової стрілки). Широкомовні фрейми значно знижують пропускну спроможність мережі, а в ряді випадків роблять її взагалі нероботоздатною [14].

Викривлення інформації у таблицях MAC-адрес мостів – це процес нескінченного циркулювання одноадресних повідомлень у петлях мережі на основі мостів [14].

Припустимо, наприклад, що станція А має запис про станцію Б в таблиці ARP і надсилає одноадресний пакет ping до станції Б. Станція Б тимчасово від'єднана від мережі, і відповідний запис в таблиці комутатора був вилучений. Припустимо, що обидва комутатори не використовують протокол STP. Тоді, фрейм надходить на порти 1/1 обох комутаторів (етап 2). Розглянемо ситуацію щодо моста Cat1. Оскільки станція Б знаходиться в несправному стані, то в таблиці моста Cat1 немає записів про MAC-адресу BB-BB-BB-BB-BB-BB, і тому Cat1 пропускає одержаний фрейм далі в мережу (етап 3). На етапі 4 Cat2 через порт 1/2 одержує фрейм відправлений Cat-1. Це призводить до виникнення двох таких ситуацій.

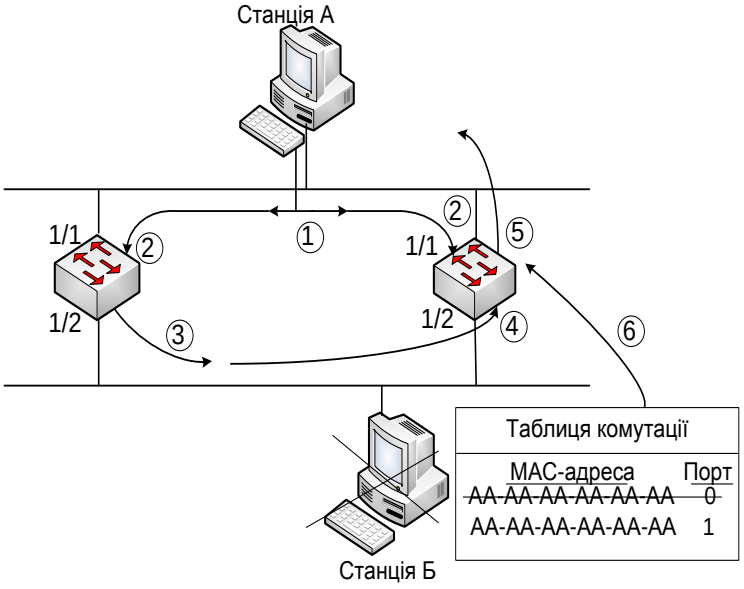


Рисунок 2.3 – Найпростіша мережа на основі мостів з елементами надлишковості

1. У таблиці комутатора Cat-2 немає запису з MAC-адресою BB-BB-BB-BB-BB-BB (етап 5), і фрейм надсилається далі на порт 1/1, що створює зворотну петлю і призводить до нероботоздатності мережі.

2. Комутатор Cat2 одержує через порт 1/2 фрейми з MAC-адресою відправника AA-AA-AA-AA-AA-AA, а потім змінює запис в своїй таблиці про MAC-адресу станції А з порту 1/1 на порт 1/2.

Оскільки фрейми циркулюють у зворотному напрямі (як було показано вище, петлі циркуляції фреймів існують в обох напрямах), то відбувається циклічне змінення даних про MAC-адресу станції А з порту 1/1 комутатора Cat2 на порт 1/2.

Отже, одноадресні повідомлення не лише насичують мережу, а й викривляють інформацію в MAC-таблицях комутаторів, що призводить до порушення роботоздатності такої мережі. Для уникнення вищевказаних проблем у мережах на основі мостів використовується протокол зв'язуючого дерева [14].

2.4.4 Протокол зв'язуючого дерева STP та його модифікації

Протокол зв'язуючого дерева (Spanning Tree Protocol – STP) – це протокол канального рівня, який використовується для підтримки такого стану мережі, у якому в ній немає петель. STP був розроблений корпорацією Digital Equipment у 1983 р. Потім комітет IEEE 802 модернізував його та опублікував у вигляді специфікації IEEE 802.1d (в цій специфікації описується і сам алгоритм роботи прозорого моста) [1, 5, 14].

Для того, щоб мережа була вільна від петель – міст при виявленні петель автоматично здійснює логічне блокування одного або кількох надлишкових портів. При цьому фізично у мережі петлі є, а логічно – немає.

Основні терміни протоколу STP

Ідентифікатор моста (BID – Bridge ID) – це восьмибайтове число, шість молодших байтів якого – це MAC-адреса блоку керування моста, а два старших байти – пріоритет моста.

Ідентифікатор порту (Port ID) моста – це двобайтове число, молодший байт якого містить порядковий номер даного порту у комутаторі, а старший задається вручну.

Кореневий міст (Root Bridge) – міст, що виконує функцію кореня дерева.

Кореневий порт (Root Port) моста – порт, що має мінімальну відстань до кореневого моста.

Призначений порт (*Designated Port*) *моста* – порт, який серед усіх портів усіх мостів даного сегмента має мінімальну відстань до кореневого моста.

Призначений міст (Designated Bridge) – це міст, якому належить призначений порт даного сегмента.

Протокольні одиниці даних моста (*BPDU – Bridge Protocol Data Unit*)

– спеціальні пакети, якими періодично обмінюються мости для автоматичного визначення конфігурації зв'язуючого дерева. Такі пакети несуть інформацію, наприклад, про ідентифікатори мостів та портів, відстань до кореневого моста тощо.

Функціонування STP

Зазначимо, що після конвергенції мережі, тобто після закінчення роботи STP кожна мережа має одне зв'язуюче дерево, тобто виконуються такі умови [4, 14]:

- у кожній мережі існує один кореневий міст;
- у кожного некореневого моста є один кореневий порт;
- в кожному сегменті є один призначений порт;
- усі інші порти (непризначені і некореневі) – не використовуються.

Для пересилання даних використовуються лише кореневі та призначені порти.

Алгоритм роботи протоколу STP має 3 етапи.

1. ***Вибір кореневого комутатора***

Зразу після завантаження кожен міст вважає себе кореневим. Всі мости починають обмінюватись BPDU (за замовчуванням кожні 2 секунди). Під час такого обміну міст з найменшим значенням ідентифікатора комутатора призначається кореневим. Зазначимо, що усі мости за замовчуванням мають ідентифікатори 32768.MAC, а отже найменший ідентифікатор матиме міст з мінімальною MAC-адресою. При цьому як кореневий може бути вибрано будь-який міст, який може не бути „центром" мережі. Для раціонального вибору кореневого моста варто змінити (зменшити) пріоритет (значення старших двох байтів BID) в того моста, який за бажанням адміністратора повинен стати кореневим.

2. ***Вибір кореневих портів***

Кожен некореневий міст повинен мати кореневий порт. Як кореневий порт вибирається той порт, що має найменшу кореневу вартість. Коренева вартість – це загальна вартість маршруту від даного порту до кореневого комутатора і обчислюється як сума умовних часів тих сегментів через які проходить шлях від даного порту до кореневого комутатора.

Вартість каналу сегмента – це величина обернена до пропускної здатності каналу. Значення вартостей каналів, залежно від їх пропускної здатності наведені у таблиці 2.1.

Таблиця 2.1 – Значення вартості шляху для деяких пропускних спроможностей каналів

Пропускна спроможність каналу (Мбіт/с)	Вартість шляху	Пропускна спроможність каналу (Мбіт/с)	Вартість шляху
10	100	155	14
16	62	622	6
45	39	1000	4
100	19	10000	2

Якщо кілька портів мають однакову кореневу вартість – то вибирається порт з найменшим значенням ідентифікатора.

3. *Вибір призначених портів*

Для кожного сегмента протокол STP вибирає один призначений порт. Призначеним портом стає той, який має найменшу оцінку маршруту до кореневого комутатора. Комутатор, у якого вибрано призначений порт для даного сегмента називається призначеним комутатором.

У кореневого комутатора усі порти є призначеними (виключенням є лише ситуація, коли деякі порти кореневого комутатора утворюють фізичні петлі).

Порти, що не стали кореневими та призначеними – блокуються і здійснюють логічне розірвання петель у мережі.

Математично доведено, що в результаті функціонування даного алгоритму для мережі отримуємо покриваюче дерево.

Приклад роботи протоколу STP

Розглянемо покроково роботу протоколу STP на прикладі простої мережі, наведеної на рисунку 2.4.

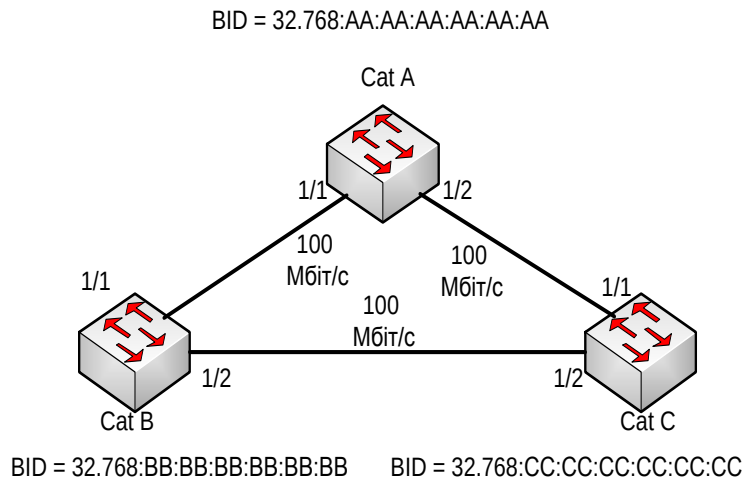


Рисунок 2.4 – Приклад роботи протоколу STP

1. *Вибір кореневого комутатора*. Оскільки пріоритети трьох мостів однакові (32768), то кореневим стає міст з найменшою MAC-адресою, тобто міст Cat A.

2. *Вибір кореневих портів.* Оскільки кожен некореневий міст повинен вибрати хоча б один кореневий порт, що має найменшу кореневу вартість, то такими кореневими портуми стануть порти 1/1 мостів Cat B та Cat C, оскільки коренева вартість кожного з них дорівнює 19 (кореневі вартості портів 1/2 мостів Cat B та Cat C дорівнюють 19 + 19 = 38).

3. *Вибір призначених портів.* Оскільки кожен сегмент у мережі повинен мати один призначений порт, то такими портуми для лівого і правого сегментів мережі стають відповідно порти 1/1 і 1/2 моста Cat A (оскільки мають найменшу кореневу вартість). Для нижнього сегмента призначеним було обрано порт 1/2 моста Cat B. Це пояснюється тим, що кореневі вартість портів 1/2 мостів Cat B та Cat C мають однакове значення 19. В такому випадку вирішальним фактором стає значення ідентифікатора відправника, а ідентифікатор моста Cat B менший, ніж моста Cat C. Порт, що за-

лишився (порт 1/2 моста Cat C) стає непризначеним і переходить у стан блокування. Отже, тепер у мережі логічно розірвано петлю.

Послідовність станів портів для STP

Є п’ять основних станів портів [1, 5, 14].

1. У стані *блокування* користувацькі фрейми не пересилаються, прослуховуються модулі BPDU.

2. У стані *прослуховування* фрейми користувачів не пересилаються, але прослуховуються. У цьому стані відбувається вибір кореневого комутатора, кореневих та призначених портів.

3. У стані *вивчення топології* фрейми користувачів не пересилаються, але вивчаються адреси інших пристроїв та заповнюється таблиця MAC-адрес.

4. У стані *пересилання* фрейми користувачів пересилаються, а також вивчаються адреси інших пристроїв та заповнюється таблиця MAC-адрес.

5. У стані *від’єднання* фрейми користувачів та BPDU не пересилаються.

На рис. 2.5 наведено послідовність портів на яких працює протокол STP [2].

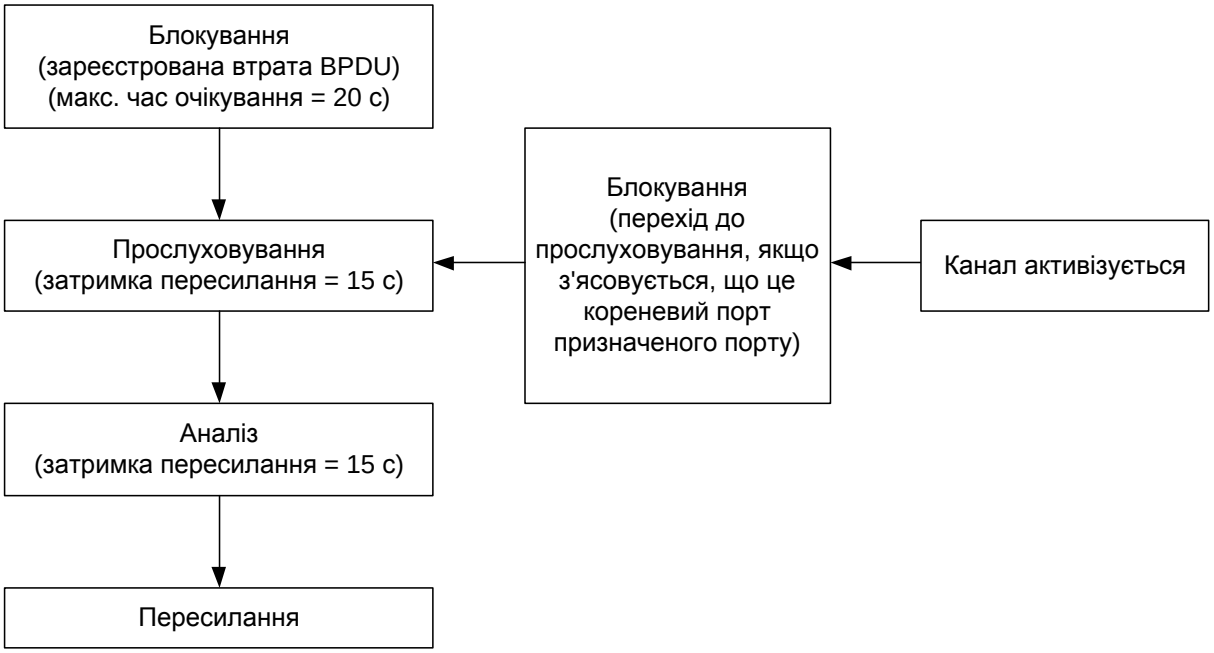


Рисунок 2.5 – Послідовність станів портів для протоколу STP

Спочатку усі порти комутатора знаходяться у стані блокування. Для переходу у стан пересилання потрібен час від 30 до 50 с.

Якщо порт під’єднано до кінцевих вузлів (не зв’язаний з іншими комутаторами), то для прискорення часу його переведення у стан пересилання на порту слід включити функцію швидкого порту (portfast). Тоді, при активізації порту він автоматично переходить зі стану блокування у стан пересилання. Це стає можливим завдяки тому, що такі порти не можуть

спричинити виникнення петель.

Розширення протоколу STP

Протокол STP має ряд обмежень та недоліків, наприклад, повільний час конвергенції мережі, необхідність перерахування дерева при кожній зміні топології мережі тощо. З метою усунення цих недоліків було розроблено ряд інших протоколів. В даному посібнику ми не будемо розглядати основи роботи цих протоколів, а лише перерахуємо основні з них [4, 14].

Rapid Spanning Tree Protocol (RSTP)

Rapid STP (RSTP) – це суттєво вдосконалений STP. Описаний у стандарті IEEE 802.1w (вподальшому включений у 802.1D-2004). Серед його переваг слід відзначити зменшення часу збіжності та більшу стійкість.

Per-VLAN Spanning Tree (PVST)

Per-VLAN STP (PVSTP) розширює функціональність STP у мережах з VLAN. Тут у кожному VLAN працює окремий екземпляр STP. Спершу цей протокол працював лише через ISL-транки, потім було розроблено розширення PVST+, яке дозволяло працювати через 802.1Q-транки, котрі використвуються набагато частіше, ніж ISL.

Є реалізації *rapid-pvst*. Вони об’єднують властивості PVST+ і RSTP.

Multiple Spanning Tree Protocol (MSTP)

Multiple STP (MSTP) є найсучаснішою стандартною реалізацією STP, що враховує усі переваги та недоліки попередніх рішень. MSTP описаний у стандарті IEEE 802.1s (в подальшому включений у стандарт IEEE 802.1Q-2003).

На відміну від PVST+ (в якому число екземплярів зв’язуючого дерева дорівнює числу VLAN), MSTP передбачає конфігурування необхідної кількості екземплярів незалежно від числа VLAN) на комутаторі. В один екземпляр MST можуть входити декілька VLAN. Проте усі комутатори, що беруть участь в MST, повинні мати однаково сконфігуровані групи VLAN, що обмежують гнучкість при зміненні конфігурації мережі.

2.4.5 Застосування комутаторів

Комутатор іноді називають багатопортовим мостом. Тоді як типовий міст має тільки два порти, комутатор має декілька десятків – сотень портів в залежності від моделі. Як і мости, комутатори отримують певну інформацію з пакетів даних від різних комп'ютерів мережі. Надалі ця інформація використовується для побудови таблиць комутації даних, які потім використовуються для визначення напряму потоків даних, що відправляються одним з комп’ютерів мережі іншому [4, 14].

Хоча в роботі мостів і комутаторів є багато спільного, комутатор складніший, ніж міст. Міст визначає, чи прямує фрейм в інший мережевий сегмент на основі MAC-адреси одержувача. Комутатор має декілька портів, до яких приєднані сегменти мережі. Комутатор вибирає порт, до якого

приєднаний пристрій-одержувач або робоча станція [14].

Комутація є технологією, що знижує вірогідність виникнення в мережах Ethernet LAN заторів за рахунок зменшення об'ємів переданих по мережі даних і збільшення смуги пропускання. Комутатори часто використовуються для заміни концентраторів, оскільки не вимагають зміни існуючої кабельної інфраструктури, що дозволяє підвищити продуктивність мережі з мінімальною кількістю змін в тій мережі, що вже існує. В наш час у сфері передавання даних все комутуюче устаткування виконує дві основні операції [4, 14, 16]:

- комутацію фреймів даних. Під цим терміном розуміється процес передавання фрейму, отриманого з одного мережевого середовища в інше (вихідне) середовище;
- підтримку комутації. Для виконання цієї функції комутатори будують і підтримують таблиці комутації і стежать за можливим утворенням маршрутних петель.

Комутатори працюють з більшими швидкостями, ніж мости, а також можуть підтримувати додаткові і достатньо важливі функції, такі, як віртуальні локальні мережі VLAN (Virtual LAN).

Комутатор Ethernet має багато переваг, зокрема, дозволяє багатьом користувачам здійснювати зв'язок паралельно за рахунок використання віртуальних каналів і створювати виділені мережеві сегменти, вільні від колізій, як показано на рис. 2.6. Такий підхід дозволяє максимізувати доступну смугу пропускання загального середовища. Другою перевагою є можливість повторно використовувати вже існуюче апаратне забезпечення і кабельну інфраструктуру, що робить перехід до використання комутаторів фінансово ефективним [1, 4].

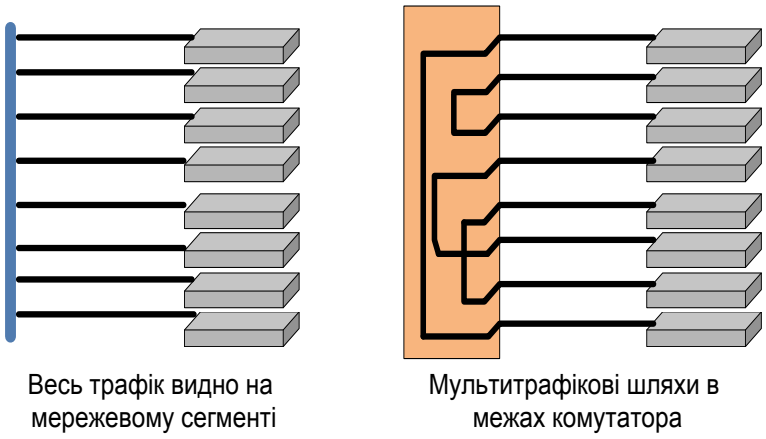


Рисунок 2.6 – Мікросегментація мережі за допомогою комутаторів

2.5 Маршрутизатори

Маршрутизатори призначені для забезпечення зв'язку між великою кількістю мереж. Такий зв'язок дає можливість комп'ютерам з різних мереж обмінюватися між собою інформацією. Зв'язані мережі можуть належати одній компанії або ж бути географічно розосереджені і належати кому завгодно. Зазвичай мережі, розділені великими відстанями, зв'язуються за допомогою розподільних мереж. Розподільні мережі засновані на великій

кількості різних технологій, включаючи маршрутизатори, засоби передавання і різних типів ліній. Маршрутизатори створювалися лише для об’єднання розподільних мереж в єдину глобальну мережу [1, 5, 12, 16].

Маршрутизатор є інтелектуальним пристроєм, який працює переважно на перших трьох рівнях еталонної моделі OSI. Проте подібно до будь-якого іншого вузла мережі, маршрутизатор здатний до взаємодії на будь-якому з семи рівнів моделі OSI. Необхідність використання перших трьох рівнів існує практично завжди. Для зв'язку з локальною мережею маршрутизатор використовує перші два рівні еталонної моделі (конструкції канального рівня). Найбільш важливою функцією є здатність маршрутизаторів ідентифікувати мережеві маршрути на основі адрес третього рівня. Цей механізм дозволяє маршрутизаторам взаємодіяти з численними мережами, використовуючи адресацію мережевого рівня незалежно від місця розташування і технології роботи мереж.

Для того, щоб зрозуміти принципи маршрутизації і розібратися в роботі маршрутизаторів, необхідно розуміти два аспекти їх роботи: фізичний і логічний. З фізичної точки зору маршрутизатор складається з величезної кількості компонентів, кожен з яких виконує строго задану функцію. З логічної точки зору маршрутизатор виконує певні дії, включаючи виявлення інших маршрутизаторів, здобуття інформації про потенційно досяжні мережі і вузли, визначення і відстежування потенційних маршрутів і передавання дейтаграм одержувачам. Це дозволяє формувати і використовувати міжнародні мережі, включаючи розподільні.

2.5.1 Основні функції та класифікація маршрутизаторів

Логічні функції маршрутизатора так само важливі, як і забезпечення фізичного взаємозв'язку множини мереж. Наприклад, для об'єднаної мережі потрібно, щоб між відправником і одержувачем був хоча б один фізичний канал передавання даних. Проте існування і використання фізичного каналу – це дві різні речі. Природно, що для нормальної роботи відправник і одержувач повинні "розмовляти" однією мовою (використовувати єдиний протокол маршрутизації). Крім того, така мова (протокол маршрутизації) дає можливість шляхом спілкування з проміжними маршрутизаторами знаходити найкоротший маршрут для передавання даних.

Таким чином, маршрутизатор повинен забезпечувати такі функції [1, 5, 12, 16]:

- фізична взаємодія;
- логічна взаємодія;
- безпека;
- визначення маршруту передавання даних.

Маршрутизатор має як мінімум два (зазвичай набагато більше) фізичні порти введення-виведення. Порти введення-виведення або, як їх часто називають інтерфейси, використовуються для фізичного приєднання сере-

довища в якому відбувається передавання до маршрутизатора. Кожен порт приєднаний до плати розширення, яка у свою чергу під'єднується до системної плати маршрутизатора. Таким чином, системна плата маршрутизатора забезпечує взаємодію декількох мереж.

Системний адміністратор повинен налаштовувати кожен інтерфейс маршрутизатора за допомогою відповідного інтерфейсу (консолі). Конфігурація включає визначення номерів портів в маршрутизаторі вказанням технології передавання даних і доступної смуги пропускання для мереж, під'єднаних до інтерфейсу, вказанням типів протоколів, які використовуватимуться з цим інтерфейсом. Параметри конкретного порту повинні залежати від типу мережевого інтерфейсу.

Слід зазначити, що в платформах верхнього рівня інтерфейси (VIP2 або лінійна плата) здатні передавати пакети без переривання роботи основного процесора.

За сферами застосування маршрутизатори поділяються на декілька класів.

Магістральні маршрутизатори (backbone routers) призначені для побудови центральної мережі корпорації. Центральна мережа може складатися з великої кількості локальних мереж, розкиданих по різних будівлях і використовуючих найрізноманітніші мережеві технології типів комп'ютерів і операційних систем. Магістральні маршрутизатори це найбільш потужні пристрої, що здатні обробляти декілька сотень тисяч або навіть декілька мільйонів пакетів в секунду, мають велику кількість інтерфейсів локальних і глобальних мереж. Підтримуються не лише середньошвидкісні інтерфейси глобальних мереж, такі як T1/E1, але і високошвидкісні, наприклад, ATM або SDH з швидкостями 155 Мбіт/с або 622 Мбіт/с і більше. Найчастіше магістральний маршрутизатор конструктивно виконаний за модульною схемою на основі шасі з великою кількістю слотів – до 12 – 14. Велика увага приділяється в магістральних моделях надійності і відмовостійкої маршрутизатора, яка досягається за рахунок системи терморегуляції, надлишкових джерел живлення, замінюваних „на ходу" (hot swap) модулів, а також симетричного мультипроцесорування.

Маршрутизатори регіональних відділень з'єднують регіональні відділення між собою і з центральною мережею. Мережа регіонального відділення, так само як і центральна мережа, може складатися з декількох локальних мереж. Такий маршрутизатор зазвичай є деякою спрощеною версією магістрального маршрутизатора. Якщо він виконаний на основі шасі, то кількість слотів його шасі менша: 4 – 5. Можливий також конструктив з фіксованою кількістю портів. Підтримувані інтерфейси локальних і глобальних мереж менш швидкісні.

Маршрутизатори віддалених офісів з'єднують, як правило, єдину локальну мережу видаленого офісу з центральною мережею або мережею регіонального відділення з глобального зв'язку. У максимальному варіанті

такі маршрутизатори можуть підтримувати і два інтерфейси локальних мереж. Маршрутизатор видаленого офісу може підтримувати роботу по комутованій телефонній лінії як резервний зв'язок для виділеного каналу. Існує дуже велика кількість типів маршрутизаторів видалених офісів. Це пояснюється як масовістю потенційних споживачів, так і спеціалізацією такого типу пристроїв, що виявляється в підтримці одного конкретного типу глобального зв’язку.

Маршрутизатори локальних мереж (комутатори 3-го рівня) призначені для розділення великих локальних мереж на підмережі. Основна вимога, що висувається до них – висока швидкість маршрутизації, оскільки в такій конфігурації відсутні низькошвидкісні порти. Всі порти мають швидкість принаймні 10 Мбіт/с, а багато хто працює на швидкості 100 Мбіт/с та більше.

2.5.2 Основні компоненти маршрутизаторів

Маршрутизатори – це надзвичайно складні пристрої. Складність їх структури полягає у певній логіці механізму маршрутизації, який дає можливість фізичному пристрою виконувати функції маршрутизації.

У загальному випадку маршрутизатор є звичайним спеціалізованим комп’ютером і відповідно складається зі схожих компонентів [5]:

- центрального процесора (Central Processing Unit – CPU);
- оперативної пам’яті (Random-Access Memory – RAM);
- базової системи введення-виведення (Basic Input/Output System, BIOS);
- операційної системи;
- системної плати;
- портів введення-виведення;
- джерела живлення, каркаса, металевого кожуха.

Функції деяких внутрішніх компонентів маршрутизатора наведено у таблиці 7.2 [1, 5, 12, 16].

Велика частина компонентів маршрутизатора закрита кожухом і недоступна для системних адміністраторів. Ці компоненти надзвичайно надійні і в нормальних умовах не повинні вийти з ладу. Виключенням з правила є встановлення додаткових модулів у маршрутизатор. У будь-який час можна додати додаткові ресурси маршрутизатору, проте при цьому доведеться знімати зовнішній кожух. Найчастіше фахівцеві доводиться встановлювати додаткові порти введення-виведення або додаткову пам'ять.

При роботі з маршрутизаторами системний адміністратор найчастіше матиме справу з його операційною системою – програмним забезпеченням, яке забезпечує спільну роботу апаратних компонентів (в разі використання маршутизаторів корпорації Cisco, поза сумнівом, це буде операційна система Internetwork Operation System, скорочено IOS), і портуми введення-виведення. Для зміни і створення конфігурації маршрутизатора системні

адміністратори зазвичай використовують інтерфейс командного рядка. Конфігурація системи визначає число, місцерозташування типів портів введення-виведення, параметри адресації і формацію про пропускну спроможність інтерфейсів і пристрою. Крім того, конфігурація маршрутизатора може включати інформацію про права і типи доступу користувачів до окремих портів введення-виведення.

Порти введення-виведення маршрутизатора – це єдиний фізичний компонент, який може побачити адміністратор. Порти надають унікальну можливість створення, мабуть, нескінченної кількості комбінацій локальних і розподільних мереж, реалізованих на основі різних технологій передавання даних. Кожен з портів в локальній або розподіленій мережі повинен мати власний порт введення-виведення на маршрутизаторі. Ці порти виконують функції, подібні до функцій мережевих інтерфейсних плат (NIC) в комп'ютері, під'єднаному до мережі; вони пов'язані з механізмами фреймування і забезпечують підтримку відповідних інтерфейсів. Багато фізичних інтерфейсів зовні здаються однаковими. Проте на більш високому рівні вони абсолютно різні. Тому перед використанням тих чи інших інтерфейсів корисно вивчити відповідні технології передавання.

Таблиця 2.2 – Функції деяких компонентів маршрутизатора

Компонент	Функції
1	2
Оперативна пам'ять (RAM/DRAM)	Використовується для зберігання таблиць маршрутизації Зберігає кеш протоколу ARP Містить швидкодіючий кеш Відповідає за буферизацію пакетів (оперативна пам'ять, що розподіляється) Забезпечує зберігання пакетів Забезпечує тимчасову і робочу пам'ять для файлів конфігурації маршрутизатора при увімкненому живленні Вміст RAM-пам'яті втрачається після вимкнення живлення або перезавантаження пристрою
Незалежна пам'ять (NVRAM)	Містить резервну або стартову копію файлу конфігурації При перезавантаженні або після вимкнення дані в цій пам'яті не стираються
Flash-пам'ять (перепрограмована пам'ять, яка зазвичай працює лише в режимі читання (EPROM). Містить дані, які при перезавантаженні або завершенні роботи маршрутизатора не знищуються	Містить образ операційної системи і мікрокод (у Flash-пам'яті може бути збережено декілька версій операційної системи Cisco IOS) Дозволяє оновлювати програмне забезпечення без заміни чіпа

Продовження таблиці 2.2

1	2
Постійний запам'ятовувальний пристрій (ROM)	Містить код команд самотестування при увімкненні живлення (Power-On Self Test – POST) Містить програми початкового завантаження і основне програмне забезпечення операційної системи Для оновлення програмного забезпечення в ПЗП потрібна заміна чіпа на системній платі пристрою
Інтерфейс (розміщується на системній платі або в окремому модулі інтерфейсу)	Утворюють мережеве з'єднання, через яке пакети даних передаються з маршрутизатора і надходять у пристрій

2.6 Порівняння комутації та маршрутизації

Маршрутизацію часто плутають з комутацією другого рівня. Принципова відмінність між ними полягає в тому, що комутація реалізована на другому рівні моделі OSI, а маршрутизація – на третьому, а отже вони використовують різну інформацію для організації передавання даних [1, 5, 12, 14, 16].

Маршрутизація призначена для передавання даних між широкомовними доменами і потребує ієрархічної схеми адресації, що і реалізовано в протоколах третього рівня (наприклад, в IP). Комутатор нічого не знає про IP-адреси і працює лише з MAC-адресами вузлів. Коли вузол відправляє інформацію нелокальному одержувачу, він адресує фрейм своєму стандартному шлюзу, використовуючи його MAC-адресу.

Комутатор об'єднує сегменти, що належать одній логічній мережі або *підмережі (subnetwork)*. Маршрутизатор, крім цього, підтримує ще і таблицю маршрутизації, яка дає можливість вибирати маршрут для доставлення даних за межі широкомовного домена. Кожна ARP-таблиця містить пари IP- і MAC-адреси. Таблиця маршрутизації містить інформацію про маршрути. MAC-адреси не організовані за певним принципом, але цей недолік не викликає проблем з управлінням мереж, оскільки окремі мережеві сегменти не містять великої кількості вузлів. Якби IP-адреса відповідала тим же правилам, мережа Internet просто не змогла б функціонувати, тому що не існувало б способу визначення маршруту для досягнення конкретних адресатів. Ієрархічна організація IP-адрес дозволяє розглядати групи адрес як єдине ціле доти, поки не потрібно буде визначити адресу індивідуального вузла.

Ще одна відмінність полягає в тому, що комутовані мережі другого рівня не блокують широкомовні розсилання третього рівня. Внаслідок цього вони можуть бути схильні до широкомовних штормів. Маршрутизатори зазвичай блокують широкомовні пакети, обмежуючи таким чином зону дії широкомовних штормів локальним широкомовним доменом і надають вищий, ніж комутатори, рівень захисту та контроль смуги пропускання.

Деякі функції маршрутизації і комутації порівнюються у табл. 2.3 [5].

Таблиця 2.3 – Порівняння функцій маршрутизатора і комутатора

Функція	Маршрутизатор	Комутатор
Швидкість	Повільніше	Швидше
Рівень OSI	Рівень 3	Рівень 2
Використовувана адресація	IP	MAC
Широкомовні розсилки	Блокуються	Пропускаються
Безпека	Вище	Нижче
Сегментація мереж	Сегментує мережу на широкомовні та колізійні домени	Сегментує мережу на домени колізій

2.7 Контрольні запитання

1. Які групи мережевих пристроїв Ви знаєте? Наведіть кілька прикладів пристроїв, що входять до таких груп.
2. Наведіть призначення мережевого адаптера.
3. Які основні чинники потрібно враховувати при виборі мережевого адаптера?
4. Поясніть призначення повторювачів.
5. Поясніть відмінність повторювача від концентратора.
6. На які типи поділяють концентратори за їх функціональним призначенням?
7. Яке функціональне призначення мостів?
8. За яким принципом мости сегментують мережу?
9. Наведіть метод прозорого мостового перенаправлення.
10. Наведіть основні методи комутації.
11. Поясніть різницю між симетричною та асиметричною комутаціями. В яких випадках доцільно використовувати асиметричну комутацію?
12. Поясніть, з якою метою у мережах на основі мостів (комутаторів) використовують надлишкові елементи.
13. Поясніть, які можуть виникати проблеми у мережах на основі мостів, без використання протоколу STP (або його похідних).
14. Наведіть термінологію протоколу STP.
15. Наведіть та охарактеризуйте послідовність станів портів протоколу STP.
16. Наведіть алгоритм роботи протоколу STP.
17. Поясніть на конкретному прикладі функціонування протоколу STP.
18. Наведіть протоколи, які покращують роботу протоколу STP.
19. Поясніть основну відмінність комутаторів від мостів.

20. Поясніть призначення таблиця комутації.

21. Поясніть призначення та основні функції маршрутизаторів.

22. За якими класами поділяють маршрутизатори відносно сфер застосування?

23. Наведіть основні компоненти маршрутизатора та їх призначення.

24. Наведіть порівняльний аналіз комутації та маршрутизації.

25. Поясніть призначення ARP-таблиці.

26. Поясніть призначення таблиця маршрутизації.

2.8 Завдання

1. На прикладі заданої комп'ютерної мережі (рис. 2.7), враховуючи умови, наведені у таблиці 2.4. поясніть:

а) на які порти комутатора буде надіслано фрейм, який надсилає вузол А до вузла В;

б) як зміниться після цього таблиця MAC-адрес комутатора;

в) на які порти комутатора буде надіслано широкомовний фрейм, якщо його надсилає вузол А;

г) на які порти комутатора буде надіслано багатоадресний фрейм, якщо його надсилає вузол В?

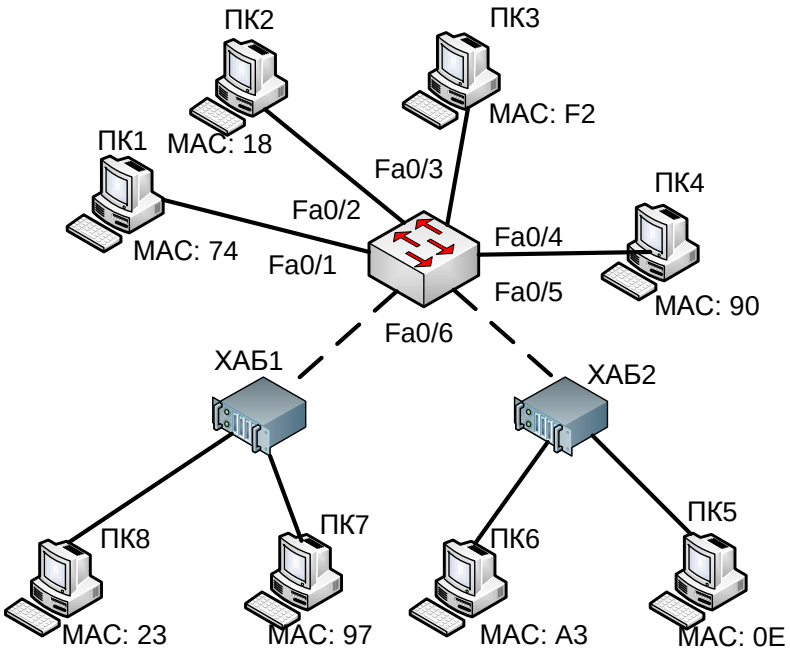


Рисунок 2.7 – Схема мережі до завдання 1

Таблиця 2.4 – Варіанти до завдання 1

<table>
<tr><th>Варіант</th><th>Вузол А</th><th>Вузол В</th><th>Таблиця MAC-адрес комутатора</th></tr>
<tr><td>1</td><td>PC1</td><td>PC5</td><td rowspan="7"><table><tr><th>Порт</th><th>Умовна MAC-адреса</th></tr><tr><td>Fa /1</td><td>74</td></tr><tr><td>Fa0/3</td><td>F2</td></tr><tr><td>Fa0/5</td><td>3</td></tr><tr><td>Fa0/6</td><td>23</td></tr></table></td></tr>
<tr><td>2</td><td>PC3</td><td>PC1</td></tr>
<tr><td>3</td><td>PC8</td><td>PC6</td></tr>
<tr><td>4</td><td>PC7</td><td>PC2</td></tr>
<tr><td>5</td><td>PC4</td><td>PC8</td></tr>
<tr><td>6</td><td>PC8</td><td>PC7</td></tr>
<tr><td>7</td><td>PC1</td><td>PC4</td></tr>
<tr><td>8</td><td>PC6</td><td>PC3</td><td rowspan="7"><table><tr><th>Порт</th><th>Умовна MAC-адреса</th></tr><tr><td>Fa0/2</td><td>18</td></tr><tr><td>Fa0/4</td><td>90</td></tr><tr><td>Fa0/5</td><td>A3
0E</td></tr></table></td></tr>
<tr><td>9</td><td>PC2</td><td>PC4</td></tr>
<tr><td>10</td><td>PC4</td><td>PC2</td></tr>
<tr><td>11</td><td>PC2</td><td>PC4</td></tr>
<tr><td>12</td><td>PC3</td><td>PC1</td></tr>
<tr><td>13</td><td>PC8</td><td>PC6</td></tr>
<tr><td>14</td><td>PC6</td><td>PC3</td></tr>
<tr><td>15</td><td>PC4</td><td>PC8</td><td rowspan="8"><table><tr><th>Порт</th><th>Умовн AC-адреса</th></tr><tr><td>Fa0/2</td><td>18</td></tr><tr><td>Fa0/3</td><td>F2</td></tr><tr><td>Fa0/5</td><td>A3
0E</td></tr><tr><td>Fa0/6</td><td>97</td></tr></table></td></tr>
<tr><td>16</td><td>PC8</td><td>PC7</td></tr>
<tr><td>17</td><td>PC1</td><td>PC4</td></tr>
<tr><td>18</td><td>PC6</td><td>PC3</td></tr>
<tr><td>19</td><td>PC2</td><td>PC4</td></tr>
<tr><td>20</td><td>PC5</td><td>PC1</td></tr>
<tr><td>21</td><td>PC1</td><td>PC5</td></tr>
<tr><td>22</td><td>PC3</td><td>PC1</td></tr>
<tr><td>23</td><td>PC8</td><td>PC6</td><td rowspan="8"><table><tr><th>Порт</th><th>Умовна MAC-адреса</th></tr><tr><td>Fa0/5</td><td>A3
0E</td></tr><tr><td>Fa0/6</td><td>23
97</td></tr></table></td></tr>
<tr><td>24</td><td>PC5</td><td>PC3</td></tr>
<tr><td>25</td><td>PC4</td><td>PC8</td></tr>
<tr><td>26</td><td>PC8</td><td>PC7</td></tr>
<tr><td>27</td><td>PC1</td><td>PC4</td></tr>
<tr><td>28</td><td>PC6</td><td>PC3</td></tr>
<tr><td>29</td><td>PC2</td><td>PC4</td></tr>
<tr><td>30</td><td>PC6</td><td>PC2</td></tr>
</table>

2. Продемонструте функціонування протоколу STP на прикладі мережі, наведеної на рис. 2.8. Вихідні дані до задачі – у табл. 2.5.

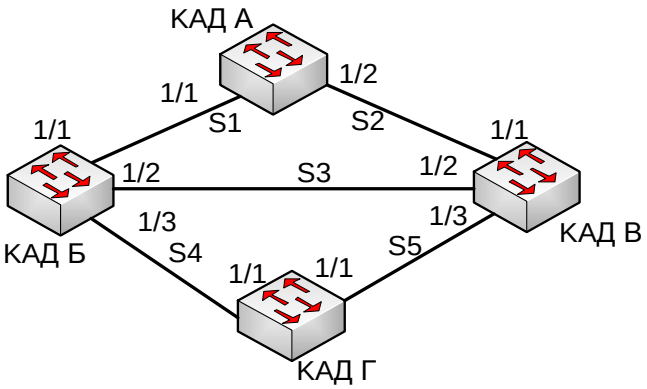


Рисунок 2.8 – Схема мережі до завдання 2

Таблиця 2.5 – Варіанти до завдання 2

Варіант	Пропускна спроможність сегментів мережі (Мбіт/с)					Умовні значення BID комутаторів			
	S1	S2	S3	S4	S5	Cat A	Cat B	Cat C	Cat D
1	100	100	100	100	100	1000:0D	1000:34	1000:E2	1000:10
2	10	100	10	100	10				
3	100	10	100	10	10				
4	1000	1000	100	100	100				
5	10	100	10	100	100				
6	100	1000	1000	100	100				
7	100	10	10	100	1000	200:F3	300:E2	400:34	500:03
8	1000	10	100	1000	1000				
9	10	100	1000	10	100				
10	1000	1000	1000	100	100				
11	1000	1000	100	100	100				
12	10	100	10	100	100				
13	100	1000	1000	100	100	1000:E2	120:FE	350:40	350:30
14	100	10	10	100	1000				
15	100	100	10	1000	100				
16	10	100	10	100	10				
17	100	10	100	10	10				
18	1000	10	100	1000	1000				
19	10	100	1000	10	100	1000:12	260:3E	100:23	800:06
20	1000	1000	1000	100	100				
21	1000	1000	100	100	100				
22	100	1000	1000	100	100				
23	1000	10	100	1000	1000				
24	10	100	1000	10	100				
25	100	10	10	100	1000	100:63	120:15	130:23	180:13
26	100	100	10	1000	100				
27	10	100	10	100	100				
28	1000	100	1000	100	100				
29	100	1000	100	100	1000				
30	100	100	10	100	100				

3 ВСТУП ДО CISCO IOS

На сьогоднішній день лідером з виробництва мережевого обладнання є компанія Cisco. Таке обладнання здебільшого працює на базі операційної системи Cisco IOS (Internetworking Operating System). Отже розглянемо детальніше деякі функції та особливості роботи в цій операційній системі як для маршрутизатора, так і для комутатора [4, 5, 9, 15, 16]. Зазначимо, що ряд команд IOS є спільними і для маршрутизатора і для комутатора, інші команди мають сенс тілька для одного з цих пристроїв.

Cisco IOS забезпечує [4]: роботу основних служб маршрутизації та комутації; надійний та безпечний доступ до мережевих ресурсів; надання засобів масштабування мережі.

3.1 Режими функціонування Cisco IOS

В Cisco IOS є три режими роботи.

1. ROM-монітор. Даний режим використовується здебільшого при несправностях у маршрутизаторі або у випадку відновлення пароля. Запрошення в цьому режимі має вигляд > або `ROMMON>`.

2. Завантаження з ROM. Даний режим використовується для заміни образу ОС. ПЗ ROM-монітора виконує процес початкового завантаження і забезпечує роботу та діагностику апаратного забезпечення на нижньому рівні. Доступ до цього режиму можливий тільки через консольний сеанс. Коли маршрутизатор (або комутатор) завантажується з ROM – доступними є обмежені функції: можна записати образ IOS у Flash з метою заміни ОС. Запрошення в цьому режимі має вигляд `Router(boot)>`.

3. Повнофункціональний режим Cisco IOS. В процесі запуску в нормальному режимі маршрутизатор (або комутатор) завантажує IOS в RAM. Для визначення або встановлення параметрів завантаження використовується конфігураційний регістр. Запрошення в цьому режимі має вигляд `Router>` (або `Switch>`).

3.2 Інтерфейс користувача

Як традиційне інтерактивне середовище в Cisco IOS використовується інтерфейс командного рядка (Command Line Interface – CLI). Доступ до CLI можна отримати кількома способами [4].

1. За допомогою сеансу через консольний порт маршрутизатора (комутатора). В такому випадку використовується низькошвидкісне послідовне з'єднання з комп'ютером, що емулює термінал. Для цього з'єднання використовується кабель типу „rollover". При цьому комп'ютер повинен підтримувати емуляцію режима VT100. Для встановлення сесії, наприклад, з операційної системи Windows можна використати HyperTerminal, вибравши такі настройки: вибрати потрібний com порт; встановити швидкість 9600 бод (baud); встановити 8 бітів даних; вказати відсутність перевірки парності (No parity); вибрати застосування одного стоп-біта (1 stop bit);

вказати відсутність механізму керування потоком (No flow control).

2. За допомогою комутованого з'єднання через модем або нуль-модемного з'єднання з портом AUX маршрутизатора (або комутатора).

3. За допомогою SSH- (Secure Shell) або Telnet-сеансу.

Перші два способи не потребують виконання будь-яких настроювань маршрутизатора (або комутатора), а третій – потребує.

Найбільше застосування отримали консольний доступ та доступ через SSH-сеанс (SSH набагато безпечніший спосіб зв'язку ніж Telnet, оскільки передбачає передавання інформації не у відкритому тексті (як Telnet), а зашифрованому вигляді за алгоритмом MD5 (Message Digest 5). Консольний доступ є безпечнішим, ніж доступ через віртуальний термінал.

Сучасне мережеве обладнання також підтримує конфігурування за допомогою графічних інтерфейсів (Graphical User Interface, GUI), Web-інтерфейсів.

Режими командного рядка

Інтерфейс командного рядка Cisco має ієрархічну структуру. Для виконання різних завдань ця структура потребує переходу в різні режими. В кожному з них командний рядок має різні мітки запрошення, що дозволяє адміністратору не плутати режими і використовувати тільки ті команди, що підтримуються даним режимом [4].

Cisco IOS забезпечує роботу інтерпретатора команд (EXEC), який перевіряє і виконує усі команди, що вводяться. В цілях безпеки EXEC-сеанси розділені на два рівні доступу: *користувацький* EXEC-режим (User Executive Mode) та *привілейований* (Privileged Executive Mode).

В користувацькому режимі доступний лише обмежений набір основних команд, які дозволяють відстежити режими роботи маршрутизатора. Часто він згадується як режим перегляду, оскільки не допускає зміни файлу конфігурації. Слово „користувацький" не означає, що доступ до пристрою може одержати будь-який звичайний користувач мережі. Цей режим призначений для співробітників, яким потрібен доступ до пристрою тільки для проведення моніторингу, але не потрібні права на зміну конфігурації пристрою. У командному рядку цей режим ідентифікується символом „>".

Привілейований режим доступу дає можливість використовувати всі команди ОС. Доступ до нього авторизований і може бути обмежений паролем та ідентифікатором користувача. Для виконання команд настроювання і керування системному адміністратору необхідно увійти до привілейованого режиму, оскільки доступ до режиму глобального конфігурування та інших спеціальних режимів доступні тільки з нього. В командному рядку цей режим ідентифікується символом „#".

Для переходу з користувацького у привілейований режим необхідно ввести команду enable, а для зворотного переходу – disable (рис. 3.1). Якщо пароль на привілейований режим був встановлений, то для продовження роботи маршрутизатор його запросить. Якщо введений пароль правильний,

то запрошення командного рядка змінюється на „#”, і IOS переходить до привілейованого EXEC-режиму.

З привілейованого режиму можна отримати доступ до режиму *глобального конфігурування*. В командному рядку він ідентифікується як „(config)#”. В ньому можна настроювати глобальні параметри та перейти в певний *специфічний режим конфігурування*, який в командному рядку ідентифікується як „(config-mode)#”, де mode означає який саме цей режим. Наприклад, з глобального режиму конфігурування можна перейти, зокрема, у режим конфігурування: інтерфейсу (командне запрошення в цьому режимі має вигляд: (config-if)#); підінтерфейсу: (config-subif)#; лінії: (config-line)#; маршрутизації:(config-router)#.

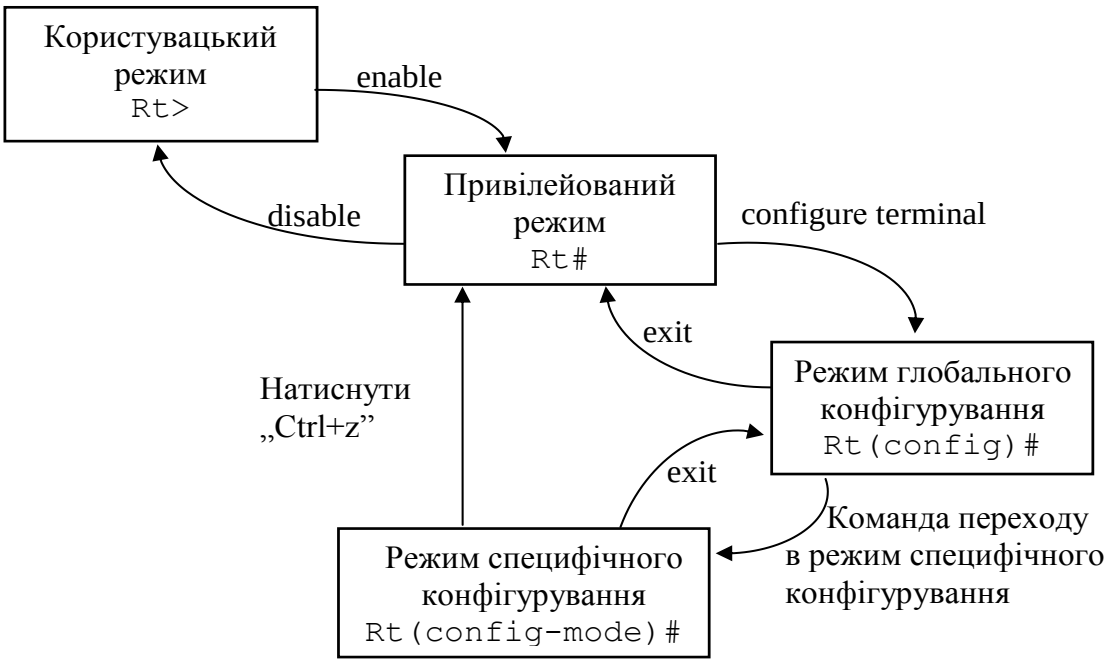


Рисунок 3.1 – Режими роботи користувацького інтерфейсу маршрутизатора

3.3 Допомога з команд Cisco IOS

Під час роботи в CLI, часто виникає потреба в отриманні допомоги щодо команд, аргументів, форматів команд. Для отримання такої допомоги варто пам’ятати таке.

Для виведення списку команд слід набрати знак питання „?”. Якщо лістинг не вміщується на екрані – внизу екрану буде виведено рядок „--More--”. Проглянути один нижчерозташований рядок можна натиснувши клавішу Enter, а нижчерозташований екран – пропуск. Натиснення будякої іншої клавіші приведе до повернення у режим командного рядка.

За допомогою знаку „?” можна уточнити будь-яку команду та її формат. Якщо знак питання ставиться безпосередньо у слові команди (без пропуску) – то виводиться список команд, що починаються на відповідні початкові літери, якщо після команди – то виводиться формат відповідного

аргументу. Приклад, що ілюструє вищесказане наведено нижче.

Зауважимо, що під час роботи в CLI можна отримати три різних види повідомлення про помилки.

- Ambiguous command – означає, що введена команда (а точніше її частина) є неоднозначною і не дозволяє визначити, яку саме команду хотів ввести адміністратор, адже існує кілька команд, які починаються на ці літери (див. другий рядок прикладу 3.1).
- Incomplete command – означає що команду введено не повністю, наприклад, не введені всі її аргументи тощо (див. восьмий рядок прикладу 3.1).
- Incorrect command – команда введена некоректно, тобто неправильно набрано цю команду або її аргументи тощо (наприклад, у п'ятому рядку прикладу 3.1 набрано неправильну команду „clok" замість „clock" а у рядку 20, замість слова "July" було неправильно введено порядковий номер цього місяця – 7 і на цю помилку вказує символ „^" 21-го рядка).

```
Router#cl
% Ambiguous command: "cl"
Router#cl?
clear clock
Router#clok
%Unknown command or computer name or unable ro finde computer
addres
Router#clock
%Incomplete command.
Router#clock ?
  set  Set the time and date
Router#clock set
%Incomplete command.
Router#clock set ?
  hh:mm:ss Current Time
Router#clock set 15:37:00
%Incomplete command.
Router#clock set 15:37:00 ?
  <1-31>  Day of the month
  MONTH  Month of the year
Router#clock set 15:37:00 14 7
                             ^
% Invalid input detected  at "^" marker.
Router#clock set 15:37:00 14 July
% Incomplete command.
Router#clock set 15:37:00 14 July ?
  <1993-2035>  Year
Router#clock set 15:37:00 14 July 2009
Router#
```

Приклад 3.1 – Отримання допомоги за командою clock set

3.4 Послідовність початкового завантаження маршрутизатора та комутатора

Розглянемо особливості завантаження маршрутизатора. При увімкненні маршрутизатора виконується процедура самотестування POST (Power-On Self Test), яка зберігається в ROM. Після цього запускається процес ініціалізації програмного забезпечення, який складається з двох етапів.

1. Системні стартові програми ініціалізують ПЗ маршрутизатора.

2. Резервні програми для відновлення ПЗ за необхідністю виконують альтернативний запуск ПЗ.

Стартові програми повинні переконатись у нормальному функціонування апаратного забезпечення; знайти та завантажити IOS; знайти і застосувати стартовий файл конфігурації (якщо його немає – увійти в режим початкового настроювання).

Отже, після виконання тесту POST виконуються такі події.

1. Виконується програма початкового завантаження (Bootstrap).

2. Завантажується образ IOS (рис. 3.2). Стандартно, при завантаженні маршрутизатора (комутатора), місцезнаходження IOS визначається апаратною платформою. Проте найчастіше маршрутизатор спочатку шукає збережені в NVRAM команди `boot system`:

```
Router(config)# boot system flash ios_filename,
Router(config)# boot system tftp ios_filename tftp_addr,
Router(config)# boot system ROM.
```

Взагалі ПЗ IOS надає користувачу кілька можливих альтернатив, наприклад, користувач може задати інші джерела завантаження IOS. Крім того за необхідністю можна використовувати власну резервну завантажувальну послідовність (fallback). Встановлення відповідних значень поля завантаження конфігураційного регістра (Configuration Register) дозволяє використовувати ряд альтернатив. Також

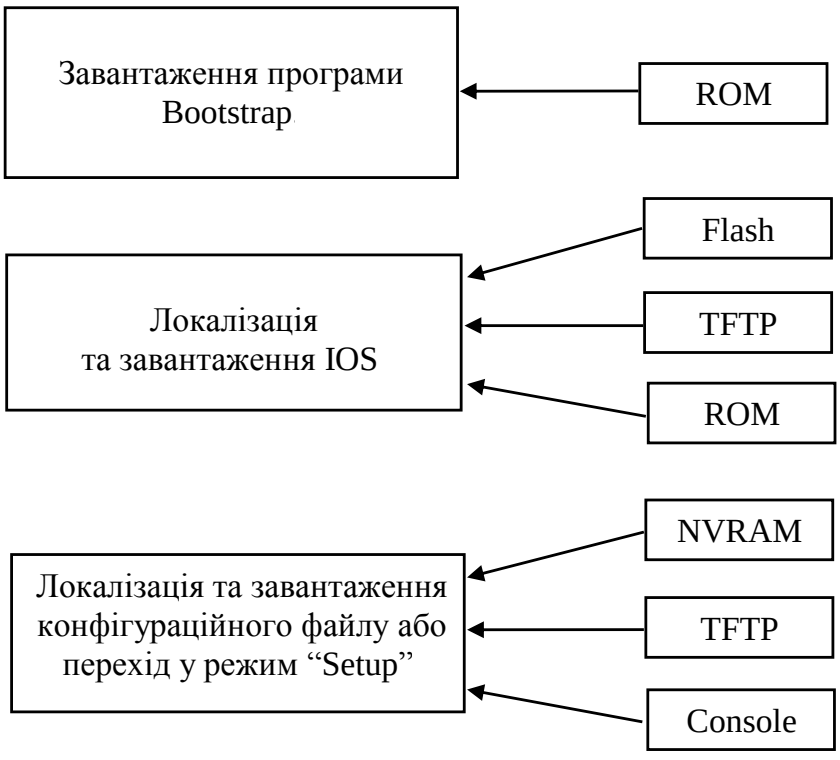


Рисунок 3.2 – Завантажувальна послідовність маршрутизатора

користувач може ввести кілька команд `boot system` (в режимі глобального конфігурування), що будуть визначати резервні джерела, які послідовно буде використовувати маршрутизатор. Якщо в NVRAM відсутні команди `boot system`, то стандартно маршрутизатор використовує IOS з Flash-пам'яті, якщо там IOS відсутня – маршрутизатор намагається використовувати TFTP. Якщо ж і він недоступний – буде завантажено скорочену версію IOS з ROM.

3. З пам'яті NVRAM у пам'ять RAM завантажується та виконується файл стартової конфігурації. Якщо в NVRAM файл конфігурації відсутній або неправильний – IOS шукає його на TFTP сервері, інакше викликає програму початкової конфігурації (Setup), яка в режимі діалогу задає системному адміністратору ряд запитань. Її мета – створення мінімальної конфігурації. В будь-який момент процес конфігурування можна перервати шляхом натиснення комбінації клавіш Ctrl-C. Зауважимо, що даний режим не рекомендується використовувати як основний засіб конфігурування.

Тепер коротко охарактеризуємо процес завантаження комутатора, (який має багато спільного з процесом завантаження маршрутизатора). Після увімкнення (або перезавантаження) комутатор завантажує програму-завантажувач (boot loader), яка зберігається в NVRAM. Ця програма:

- виконує ініціалізацію процесора на низькому рівні. Ініціалізуються регістри процесора, які контролюють відображення фізичної пам'яті, об'єм пам'яті та її швидкість;
- виконує самотестування POST для підсистем процесора. Відбувається перевірка оперативної пам'яті та частини пристрою flash, яка містить файлову систему;
- ініціалізує файлову систему flash на системній платі;
- завантажує заданий за замовчуванням образ операційної системи в пам'ять і виконує завантаження комутатора. Завантажувач знаходить образ Cisco IOS на комутаторі переглядаючи каталог, що має таке ж ім'я, як файл образ Cisco IOS (виключивши розширення .bin). Якщо там образ не знайдено, завантажувач проглядає кожен підкаталог перед продовженням пошуку в поточному каталозі.

Потім ОС ініціалізує інтерфейси, використовуючи команди файлу конфігурації config.text, що зберігається у flash-пам'яті комутатора.

3.5 Файли конфігурації маршрутизатора та комутатора

Нагадаємо, що файл конфігурації містить команди настройки маршрутизатора (комутатора) та його служб. Розмір цього файлу в більшості випадків становить від кількох сотень до кількох десятків тисяч байтів. Взагалі є два файли конфігурації: *робочий* (Running Config) – зберігається в RAM (його ще називають поточним або діючим). Після перезавантаження робочий файл конфігурації втрачається, оскільки пам'ять RAM залежна від

живлення (рис. 3.3); *стартовий* (Startup Config) – зберігається в NVRAM, не залежить від живлення і копіюється в RAM під час запуску системи.

Після будь-яких внесень змін у конфігурацію маршрутизатора (комутатора) ці зміни можна перевірити за допомогою команди `show running-config`, яка відображає поточну конфігурацію. Якщо при цьому значення змінних неправильні – можна виконати одну з таких дій:

- використати команди конфігурування з префіксом `no`;
- перезавантажити систему та перезапустити оригінальний файл конфігурації з NVRAM;
- скопіювати резервну копію файлу конфігурації з TFTP- сервера;
- вилучити файл початкової конфігурації за допомогою команди `erase startup-config`, перезавантажити маршрутизатор і увійти в режим початкової установки (Setup).

Для збереження конфігураційних змін в NVRAM слід ввести команду

```
Router# copy running-config startup-config
```

Доцільно зазначити, що дуже важливо в межах кожної організації (особливо, якщо вона велика) розробити єдиний стандарт для файлів конфігурації. Це дозволить уникнути хаосу, зайвої складності настроювання мережі, зменшити незапланований час простою тощо.

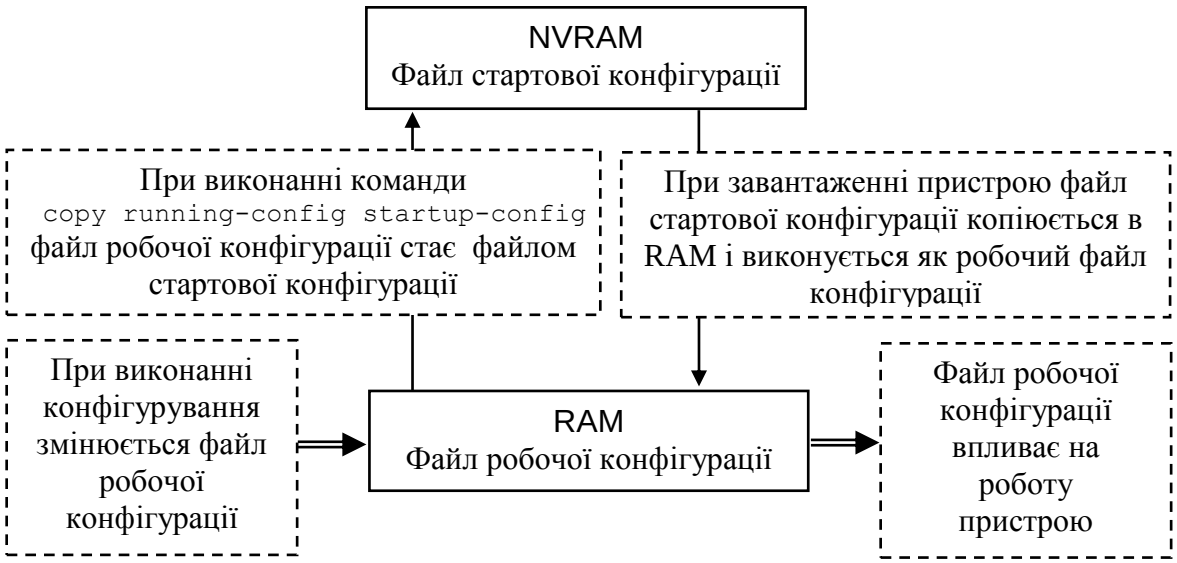


Рисунок 3.3 – Файли робочої та стартової конфігурацій

3.6 Початкова конфігурація комутатора

Варто зазначити, що новий комутатор на відміну від маршрутизатора, має стандартну початкову конфігурацію, встановлену виробником. Змінювати конфігурацію комутатора можна через інтерфейс командного рядка, а також Web-інтерфейс з використанням браузера [4].

При першому ввімкненні живлення комутатора в його файлі робочої конфігурації містяться стандартні установки. Комутатор має стандартне ім’я Switch. На консолі та лініях віртуального термінала (VTY) паролі не

встановлені.

За замовчуванням порти комутатора встановлені в автоматичний режим, це означає, що вони автоматично визначають режим роботи (дуплексний або півдуплексний) та швидкість порту.

За замовчуванням всі порти комутатора знаходяться в першій віртуальній локальній мережі (virtual LAN, VLAN). Вона вважається стандартною мережею керування (management VLAN). Для відображення інформації про мережі VLAN, визначених на комутаторі, використовується команда `show vlan`.

Оскільки новий комутатор ще не конфігурувався, в його флеш-каталозі відсутні файл бази даних мереж VLAN (vlan.dat) і файл збереженої конфігурації (config.text). Файл vlan.dat використовується для зберігання інформації про локальні VLAN даного комутатора і комутатор застосовує його для сумісного використання інформації про мережі VLAN з іншими комутаторами. За замовчуванням у флэш-каталозі міститься образ IOS (файл з розширенням .bin), файл змінних середовища з ім'ям env_vars і підкаталог з ім'ям html. Для відображення вмісту флэш-каталога використовується команда `dir flash:`.

У стандартній конфігурації комутатор має один широкомовний домен і може керуватись і конфігуруватись через консольний порт з використанням CLI. У цій конфігурації також встановлено протокол зв'язуючого дерева. При цьому в стандартній конфігурації гарантується найвищий рівень безпеки, оскільки комутатору ще не було призначено IP-адреси.

Насамкінець зазначимо, що для невеликих мереж стандартна конфігурація може виявитись цілком достатньою і користувач відразу може скористатися перевагами мікросегментації та високої продуктивності мережі, які забезпечуються комутатором. Проте за необхідності користувач може повністю змінити існуючу конфігурацію. Для цього слід: вилучити існуючу VLAN-інформацію шляхом видалення файлу vlan.dat бази даних VLAN з флеш-каталогу вилучити резервний файл конфігурації та перезавантажити комутатор (приклад 3.2).

```
Switch# delete flash:vlan.dat
Delete filename [vlan.dat]?
Delete  flash:vlan.dat? [confirm]
Switch# erase startup-config
<частина виведення опущено>
Switch# reload
```

Приклад 3.2 – команди вилучення поточної конфігурації комутатора

На прикладі 3.2 показано, як вилучити поточну конфігурацію у комутаторах серії Catalyst 2950.

3.7 Деякі команди початкового конфігурування та моніторингу роботи маршрутизатора та комутатора

Перш за все варто зазначити, що під час виконання деяких команд IOS видає певні повідомлення. Такі повідомлення можуть вставлятись усередину команд, що набирає мережевий адміністратор. Хоча це не приводить до необхідності нового набирання цих відповідних команд і операційна система їх правильно розуміє – це може спричиняти деякий дискомфорт адміністратору. Для виправлення такої ситуації доцільно скористатись командою `logging synchronous`:

```
Router(config)#line console 0
Router(config-line)#logging synchronous
Router(config-line)#line vty 0 4
Router(config-line)#logging synchronous
```

Тепер, під час консольної- або VTY-сесії команди адміністратора не будуть розриватися системними повідомленнями.

Настроювання імені – це одна з перших команд, яку слід виконати на маршрутизаторі або комутаторі. Ім'я повинно говорити адміністратору про місцезнаходження та функції даного пристрою. Адже адміністратор, як правило, має справу з багатьма маршрутизаторами, комутаторами тощо і для їх настроювання (тим більше за допомогою SSH-сесій), адміністратору необхідно орієнтуватись на якому пристрої він знаходиться. Задання імені виконується в глобальному режимі:

```
Router(config)#hostname Rt1_VNTU
Rt1_VNTU(config)#
```

Настроювання паролів. Паролі використовуються для захисту від несанкціонованого доступу. Паролем можна захистити доступ до:

- *консолі.* Встановлення пароля для консолі виконується за допомогою команд

```
Router(config)#line console 0
Router(config-line)#password <password>
Router(config-line)#login
```

- *віртуальної лінії терміналу* (virtual terminal – VTY). Telnet-доступ. Одночасно може бути встановлено кілька Telnet-сеансів. Для кожної лінії пароль можна встановити індивідуально, а можна один для всіх ліній. Встановлення пароля виконується за допомогою команд

```
Router(config)# line vty 0 4
Router(config-line)# password <password>
Router(config-line)# login
```

- *привілейованого режиму роботи*. Для обмеження доступу до привілейованого режиму слід ввести команду

```
Router(config)# enable secret <password>.
```

Для збереження пароля тут використовується одностороннє шифрування за алгоритмом MD5, що унеможливлює відновлення пароля. Якщо дана команда не підтримується можна скористатись командою

```
Router(config)# enable password <password>,
```
але в такому випадку пароль буде зберігатись в конфігураційних файлах у незашифрованому вигляді. Для заборони відображення пароля у відкритому вигляді можна скористатись командою
```
Router(config)# service password-encryption.
```
В такому випадку будуть шифруватися усі паролі (крім пароля заданого за командою `enable secret`), але рівень захисту інформації тут невисокий.

Для відмінення шифрування усіх паролів використовується команда
```
Router(config)#no service password-encryption.
```
Настроювання послідовних інтерфейсів (виконується для маршрутизатора) передбачає виконання таких кроків:

- *увійти в режим глобального конфігурування*;
- *увійти в режим настроювання потрібного інтерфейсу.*

Формат команди може бути
```
Router(config)# interface type port,
Router(config)# interface type slot/port,
Router(config)# interface type slot/subslot/port,
```
де `interface` – це тип інтерфейсу (наприклад Serial, FastEthernet, GigabitEthernet, Loopback тощо); `port`, `subslot`, `slot` – номери порту, слота та підлота, відповідно;

- *задати IP-адресу інтерфейсу та маску під мережі*;
- *вказати смугу пропускання каналу* (необов'язково);
- *встановити частоту синхроімпульсів для DCE* (за замовчуванням маршрутизатори функціонують як DTE, але можуть бути налаштовані і як DCE);
- *навести опис інтерфейсу* – для того, щоб адміністратор міг згадати якусь важливу інформацію про цей інтерфейс. Опис доцільно створювати відповідно спеціальному формату (наприклад, призначення та розміщення інтерфейсу, опис пристроїв, що під'єднані до нього, тощо);
- *увімкнути інтерфейс*.

Команди, що відповідають даним крокам наведено нижче
```
Router# configure terminal
Router(config)# interface serial 0/0/1
Router(config-if)# ip address <ip address> <netmask>
Router(config-if)# bandwidth 56
Router(config-if)# clockrate 56000
Router(config-if)# description interface ftom InITKI
Router(config-if)# no shutdown
```
Стандартно усі інтерфейси маршрутизатора вимкнені. Для увімкнення інтерфейсу маршрутизатора або комутатора використовують команду `no shutdown`, а для їх вимкнення – `shutdown`.

Настроювання Ethernet-інтерфейсів

Для маршрутизатора таке настроювання виконується аналогічно настроюванню його послідовних інтерфейсів, за виключенням того, що чет-

вертий і п’ятий кроки виконувати не потрібно.

Для комутатора настроювання Ethernet-інтерфейсів дещо інше. Порти FastEthernet комутатора за замовчуванням встановлені в режим автоматичного визначення швидкості передавання і дуплексний режим. Це дозволяє інтерфейсум пристроїв, що беруть участь у сеансі зв’язку, змінювати ці установки. Якщо адміністратору треба бути впевненим в тому, що даний інтерфейс має конкретну швидкість передавання та дуплексний або півдуплексний режим, то ці значення слід встановити вручну, як показано на прикладі 3.3.

```
Switch(config)# interface FastEthernet0/2
Switch(config-if)# duplex full
Switch(config-if)# speed 100
```

Приклад 3.3 – Встановлення швидкості та режиму передавання портів комутатора

Іншою корисною функцією, яка може бути встановлена на порту, є опція `portfast`. Якщо порт комутатора приєднаний тільки до станцій кінцевого користувача (тобто не приєднаний до іншого комутатора), то на ньому слід встановити функцію `portfast` за допомогою команди

```
Switch# set spantree portfast 4/1 enable.
```

В такому випадку при першому використанні порту він автоматично переходить із заблокованого стану в стан пересилання.

Настроювання Loopback-інтерфейсів

Іноді трапляються випадки, коли виникає необхідність проемулювати з’єднання котрого в даний момент фізично не існує (наприклад, в майбутньому планується під’єднатись до провайдера, до іншої організації тощо). Для цього на маршрутизаторі можна настроїти Loopback-інтерфейс, задавши йому потрібну IP-адресу та маску:

```
Router(config)# interface loopback number
Router(config-if)# ip address <ip address> <netmask>
Router(config-if)# description My virtul interface
Router(config-if)# no shutdown
```

Настроювання банерів

Банером називається повідомлення, що відображається під час входу до системи. Таке повідомлення (його ще називають повідомленням дня – Message Of The Day – MOTD) можна використовувати для передавання деякої інформації до всіх користувачів мережі. Наприклад, часто таке повідомлення попереджає користувачів про те, що вхід в систему заборонено для неавторизованих

```
Router(config)# banner motd #
Enter TEXT message. End with the character '#'.
**************************************
WARNING!! Unauthorized Access Prohibited!!
**************************************#
```

Приклад 3.4 – Настроювання повідомлення дня

користувачів. Для задання повідомлення дня використовують команду `banner motd`, після якої наводять потрібний текст, виділений символами „#”, як показано на прикладі 3.4.

Настроювання Telnet- та SSH-доступу

Насамперед зазначимо, що вказати потрібний метод доступу: або telnet, або SSH, або telnet і SSH можна так:

```
Router(config)# line vty 0 4
Router(config-line)# transport input <mode>,
```

а для вибору однієї з вищевказаних трьох альтернатив значення аргументу `mode` буде `telnet`, `ssh` та `all`, відповідно.

Telnet-доступ

Для маршрутизатора, як вказувалось вище, для отримання Telnet-доступу достатньо встановити пароль на VTY-доступ і задати IP-адресу і маску підмережі відповідному інтерфейсу [4].

Для комутатора організація Telnet-доступу передбачає призначення йому IP-адреси і задання шлюзу за замовчуванням. На прикладі 3.5 показано як це зробити для комутаторів моделей Catalyst 2950. За замовчуванням мережа VLAN 1 є віртуальною мережею керування. У мережі, побудованій на комутаторах, усі пристрої об’єднаної мережі повинні знаходитись в мережі керування VLAN. Це дозволяє з однієї робочої станції отримувати доступ до всіх пристроїв об’єднаної мережі, конфігурувати та керувати ними.

```
Switch(config)# interface VLAN1
Switch(config-if)# ip address 192.168.1.2 255.255.255.0
Switch(config)# ip default gateway 192.168.1.1
```

Приклад 3.5 – Призначення комутатору IP-адреси і шлюзу за замовчуванням

SSH-доступ

SSH-доступ також передбачає наявність SSH-клієнта та SSH-сервера. Для реалізації такого доступу слід згенерувати ключі RSA (Rivest, Shamir, Adleman encryption). RSA передбачає використання загальнодоступного ключа (public key), який зберігається на загальнодоступному RSA-сервері та приватного ключа (private key), який знають лише відправник та отримувач. Загальний ключ може бути відомий кожному і використовується для шифрування повідомлень. Такі зашифровані повідомлення можуть бути дешифровані тільки при використанні приватного ключа.

Вам потрібно згенерувати RSA-ключі, використовуючи команду

```
crypto key generate rsa
```

Ця процедура потрібна, якщо Ви конфігуруєте пристрій (наприклад, маршрутизатор) як SSH-сервер. Слід увійти в привілейований режим, призначити пристрою ім’я, вибрати доменне ім’я та згенерувати пару ключів RSA:

```
Router# configure terminal
Router(config)# hostname Rt
Rt(config)# ip domain-name my_domain
Rt(config)# crypto key generate rsa
```

У четвертій команді (`crypto key generate rsa`) Ви дозволяєте SSH-серверу виконувати локальну та віддалену аутентифікацію і генеруєте пару RSA-ключів. Під час генерації цих ключів Ви повинні вибрати їх довжину (Cisco, наприклад, рекомендує вибирати довжину у 1024 біти). Більша довжина хоч і була б безпечніша, але й потребує більше часу на генерування та використання.

Подивитись стан SSH-сервера можна за командою `show ip ssh` або `show ssh`. Для вилучення пари ключів RSA, слід ввести команду

```
Rt(config)# crypto key zeroize rsa,
```

після якої сервер SSH автоматично вимикається.

Під час конфігурування SSH-доступу можна також вказати:

- використовувану версію SSH (першу або другу) за допомогою команди `Rt(config)# ip ssh version <v_num>`, де `v_num` – номер версії.

Якщо цю команду не ввести – то SSH-сервер вибере найстаршу версію, підтримувану клієнтом (наприклад, якщо клієнт підтримує SSHv1 та SSHv2, сервер вибере SSHv2);

- керувальні параметри SSH: час тайм-ауту (time-out) в секундах з діапазону від 0 до 120 (стандартно 120 секунд) можна вказати за допомогою команди `Rt(config)# ip ssh timeout <seconds>`. Цей час відводиться для встановлення з'єднання; максимальну кількість спроб аутентифікації клієнта з діапазону від 0 до 5 (стандартно 3) можна вказати за допомогою команди `Rt(config)#ip ssh authentication-retries <number>`.

Робота з таблицею MAC-адрес (для комутаторів)

Комутатори дізнаються про MAC-адреси комп'ютерів або інших кінцевих пристроїв, приєднаних до їх портів шляхом аналізу адрес джерел у фреймах, що надходять на даний порт. Далі ці адреси заносяться в таблицю MAC-адрес комутатора. Для перегляду відомих комутатору адрес необхідно увійти до привілейованого режиму, як показано на прикладі 3.6.

```
Switch#show mac-address-table
Dynamic Address Count: 2
Secure Address Count: 0
Static Address (User-defined) Count: 0
System Self Address Count: 13
Total MAC addresses: 15
Maximum MAC addresses: 8192
Non-static Address Table:
Destination Address Address Type VLAN Destination Port
0010.7a60.ad7e         Dynamic       1    FastEthernet0/2
00e0.2917.1884         Dynamic       1    FastEthernet0/5
```

Приклад 3.6 – Результат виконання команди show mac-address-table

Адреси вивчаються динамічно і комутатор може підтримувати в цілому тисячі MAC-адрес, на кожному порту можуть бути декілька десятків адрес. Для економії пам'яті і оптимального функціонування комутатора іноді виникає необхідність вилучити деякі позиції з таблиці MAC-адрес. Для цього всі позиції таблиці мають часові мітки, які відображають час надходження на порт пакета з даною адресою. Робочі станції можуть бути від'єднані від порту, вимкнуті або перемкнуті на інший порт цього ж або іншого комутатора, можлива заміна карти мережевого інтерфейсу. Це все може призвести до плутанини під час пересилання фреймів. Для уникнення цього комутатор настроєно так, що за відсутності фреймів з раніше записаною адресою протягом певного часу (зазвичай 300 секунд) відповідна MAC-адреса автоматично вилучається з таблиці.

Замість очікування природного застарівання динамічної позиції адреси адміністратор може скористатись для її вилучення командою `clear mac-address-table` (приклад 3.7).

Позиції MAC-адрес, сконфігуровані адміністратором, можуть бути вилучені аналогічно, що забезпечує миттєве вилучення позицій таблиці з адресами, які стали недійсними.

```
Switch#clear mac-address-table
Switch#show mac-address-table
Dynamic Address Count: 0
Secure Address Count: 0
Static Address (User-defined) Count: 0
System Self Address  Count: 13
Total MAC addresses: 14
Maximum MAC addresses: 8192
Non-static Address Table:
Destination Address Address Type VLAN Destination Port
 ------------------  -----------  ----  ---------------
```

Приклад 3.7 – Результат виконання команди clear mac-address-table

Конфігування статичних MAC-адрес

Можлива ситуація, в якій доцільно постійно прив'язати деяку MAC-адресу до конкретного інтерфейсу комутатора. В цьому випадку автоматичного вилучення MAC-адреси після закінчення звичайного терміну збереження адреси не відбудеться.

Постійна адреса може бути прив'язана до інтерфейсу у разі потреби під'єднати сервер або робочу станцію користувача до даного порту за умови що MAC-адреса відома. При цьому також може бути підвищено рівень безпеки. Для встановлення статичної MAC-адреси на інтерфейсі комутатора використовується такий синтаксис команди:

```
Switch(config)# mac-address-table static mac-address-of-
```

```
host
interface FastEthernet ethernet-number vlan vlan-name
```

Наприклад,

```
Switch(config)# mac-address-table static 0a40.4в00.2341
interface FastEthernet0/5 vlan VLAN1
```

Для вилучення позиції адреси з таблиці використовується ця ж команди з ключовим словом `no`.

Конфігурування безпеки портів комутатора

Забезпечення безпеки в об'єднаній мережі є важливим завданням адміністратора. Порти комутатора, що відносяться до рівня доступу, внаслідок структурної схеми прокладки кабелів доступні в стінних роз'ємах офісів та інших приміщень і до будь-якого з них можна під'єднатись за допомогою ПК. Вони також є потенційними точками входу в мережу для несанкціонованих користувачів. Комутатори мають функцію надання безпеки портум. Зокрема, можна обмежити кількість адрес, про які можна дізнатися на конкретному інтерфейсі. Під час конфігурування можуть бути задані певні дії, якщо ця кількість перевищена, наприклад, команди

```
Switch(config)# interface FastEthernet0/1
Switch(config)# port security action shutdown,
```

призводять до вимкнення відповідного порту, якщо кількість MAC-адрес перевищено). Безпечні MAC-адреси можуть бути встановлені статично, але такий спосіб може виявитись досить складним, крім того велика вірогідність помилок.

Альтернативним підходом є реалізація заходів безпеки для порту на інтерфейсі комутатора. Першою адресою, про яку динамічно дізнається комутатор, стає безпечною. Для зміни типу безпеки на інтерфейсі використовується форма цієї команди з ключовим словом `no`. Для тестування статусу безпеки на порту використовується команда `show port security`.

Деякі команди групи show

В операційній системі ряд команд `show`, які надають статичну інформацію стосовно роботи пристрою [1,2,4]. Наприклад, вони дозволяють отримати інформацію щодо конфігурації, функціонування та статусу частин маршрутизатора або комутатора.

Переглянути список усіх параметрів команди `show` можна набравши `show ?`

Деякі параметри команди вказані нижче:

- `show arp` – відображає ARP таблицю пристрою;
- `show startup-config` – відображає конфігурацію, що міститься у NVRAM (стартовий файл конфігурації);
- `show running-config` – відображає конфігурацію, що є у RAM

(робочий файл конфігурації);

- `show interfaces` – відображає статистику про усі інтерфейси пристрою. Якщо треба проглянути статистику щодо конкретного інтерфейсу слід ввести команду `show interfaces fastethernet 0/1`;
- `show ip interface brief` – надає стислу інформацію про усі інтерфейси пристрою та їх статус;
- `show controllers serial` – відображує інформацію про апаратні засоби маршрутизатора;
- `show clock` – відображає встановлений час;
- `show hosts` – відображає список керованих імен вузлів та адрес;
- `show users` – відображає список користувачів, під'єднаних до пристрою;
- `show history` – відображає список введених команд;
- `show flash` – відображає інформацію стосовно flash-пам'яті та файлах, що в ній знаходяться;
- Досить корисною командою, що надає багато інформації є команда `show version`. Вона виводить інформацію, наприклад, для маршрутизатора про: версію IOS та її стислий опис; версію завантажувальної програми (Bootstrap ROM); версію скороченої IOS в ПЗУ (Boot ROM); час безперервної роботи пристрою (Router uptime); останній метод перезапуску пристрою (restart method); ім'я образу системного файлу та його місця розташування; номер апаратної платформи пристрою; настройки конфігураційного регістра;
- `show protocol` – відображає глобальний стан і стан інтерфейсів будь-якого протоколу третього рівня маршрутизатора;
- `show mac-address-table` – відображає таблицю MAC-адрес комутатора.

Зазначимо, що команд `show` є дуже багато і їх недоцільно наводити в даному посібнику. Подивитись детальніше ці команди та їх функції можна, наприклад у [4, 5, 16].

Загальні відомості стосовно команд групи debug

Команди налагоджування (або команди `debug`) дозволяють локалізовувати проблеми з протоколами та неправильними настройками [1, 2, 4]. Тоді як команди `show` надають лише статичну картину роботи пристрою, дані, отримані за допомогою команд `debug`, є динамічними і забезпечують глибше розуміння поточних подій у процесі роботи пристрою. Динамічний стиль роботи команд `debug` здійснюється за рахунок системних ресурсів, що може призводити до перевантаження процесора і порушувати нормальну роботу пристрою [4, 5]. Тому використовувати їх варто лише за необхідності. Крім того, рекомендується звужувати поле пошуку проблеми до кількох варіантів. Іншими словами команди групи `debug` слід викорис-

товувати для локалізації конкретних проблем, а не для моніторингу нормальної роботи КМ. Особливо слід пам'ятати, що команду `debug all` варто застосовувати якомога рідше, оскільки вона може призвести до порушення нормальної роботи мережі.

Додатковою корисною службою Cisco IOS, яка підвищує цінність роботи команд `debug` є команда `timestamp`, що помічає повідомлення команди `debug` часовими мітками, за якими можна дізнатись час, коли відбулась та чи інша подія та інтервал часу між ними.

Команди `no debug all` та `undebug all` вимикають видавання усіх діагностичних повідомлень. Для відключення конкретної команди `debug` використовують ту ж команду з додаванням ключового слова `no`. Проглянути все, що в даний час досліджується за допомогою команди `debug` можна за допомогою команди `show debugging`.

Детальніше команди `debug` та їх функції можна проглянути, наприклад, у літературі [4, 5, 16].

3.8 Контрольні запитання

1. Поясніть роль, призначення та функції ОС Cisco IOS.
2. Поясніть, яким чином можна отримати доступ до конфігурування мережевого обладнання на базі Cisco IOS.
3. Наведіть та стисло охарактеризуйте режими роботи Cisco IOS.
4. Наведіть способи отримання доступу до режиму командного рядка (CLI).
5. Поясніть відмінність між сеансами Telnet та SSH.
6. Поясніть, як виконати консольне з'єднання з маршрутизатором (комутатором) з операційної системи Windows.
7. Поясніть відмінність між консольним та віддаленим доступом.
8. Наведіть режими командного рядка. Які задачі виконують в кожному з цих режимів? Які ознаки цих режимів?
9. Наведіть діаграму переходів між різними режимами командного рядка.
10. Поясніть яким чином в Cisco IOS реалізовано допомогу за командами. Які повідомлення про помилки Ви можете отримувати при некоректному або неповному набиранні команд?
11. Наведіть послідовність початкового завантаження маршрутизатора з відповідними поясненнями.
12. Наведіть призначення та типи файлів конфігурації маршрутизатора (комутатора). Чим вони відрізняються?
13. Охарактеризуйте початкову конфігурацію комутатора.
14. Поясніть важливість настроювання імені маршрутизатора (комутатора) та наведіть відповідні команди.
15. Поясніть, що можна захистити паролями при роботі з маршутизатором (комутатором). Наведіть відповідні команди.

16. Наведіть команди настроювання Ethernet- та Serial-інтерфейсів. Чим відрізнаються такі настроювання?

17. Поясніть призначення та команди настроювання Loopback-інтерфейсу.

18. Поясніть призначення банерів та наведіть команди їх настроювання.

19. Поясніть, як виконати доступ по Telnet до маршрутизотора та комутатора.

20. Поясніть, як виконати SSH-доступ до маршрутизотора та комутатора.

21. Наведіть команди перегляду та очищення таблиці MAC-адрес комутатора.

22. Наведіть команди конфігурування статичних MAC-адрес.

23. Наведіть команди базового конфігурування безпеки портів комутатора.

24. Поясніть, з якою метою використовуються команди групи Show. Наведіть кілька таких команд та поясніть їх призначення.

25. Поясніть, яку інформацію можна отримати в результаті виконання команди show version.

26. Поясніть призначення команд групи debug.

27. Поясніть різницю між командами груп show та debug.

4 ПРОТОКОЛИ МАРШРУТИЗАЦІЇ

4.1 Призначення та класифікація протоколів маршрутизації

Протоколи маршрутизації призначені для автоматичної побудови таблиць маршрутизації (ТМ), на основі яких виконується переміщення пакетів. Такі таблиці містять дані яких достатньо для прийняття рішення для пересилання будь-якого пакета, що надійшов до маршрутизатора. Вміст таблиці залежить від технології складеної мережі. Як правило обирається "найкоротший" маршрут (під довжиною маршруту розуміють його метрику – числове значення, яке впливає на вибір маршруту: чим менша метрика – тим краще. Метрика може визначатись, наприклад, кількістю проміжних вузлів, пропускною здатністю, часом затримки, надійністю каналів між маршрутизаторами).

Усі способи маршрутизації можна поділити на 2 великі групи: без таблиць та з ТМ [1].

Маршрутизація без таблиць поділяється на лавинну; керовану подіями; від джерела.

Лавинна маршрутизація – це найпростіший спосіб передавання, який передбачає, що кожен маршрутизатор відправляє пакет усім своїм сусідам, крім того, від кого він отримав свій пакет. Пропускна здатність мережі в такому випадку використовується дуже неефективно.

Маршрутизація, керована подіями передбачає, що пакет до певної мережі призначення надсилається за маршрутом, який вже приводив до успіху. В такому випадку необхідно, щоб маршрутизатор-відправник міг фіксувати факт успіху доставки пакета.

Маршрутизація від джерела передбачає, що відправник розміщує у пакет інформацію про те, які проміжні маршрутизатори повинні брати участь у передаванні пакетів. Таку інформацію або надає адміністратор вручну, або вузол-відправник формує автоматично.

Маршрутизація на основі таблиць в свою чергу поділяється на статичну і динамічну (адаптивну). Статична маршрутизація передбачає ручне прописування маршрутів адміністратором. Така маршрутизація при зміні структури мережі потребує ручного змінення маршрутів.

У випадку динамічної маршрутизації мережі можуть оновлювати свої ТМ та швидко адаптуватися до змін топології та стану з'єднань. Успішне функціонування цього виду маршрутизації залежить від виконання маршрутизатором двох його основних функцій: підтримки ТМ в актуальному стані та своєчасного розповсюдження інформації у вигляді анонсів та оновлень маршрутів серед інших маршрутизаторів [5].

При розповсюдженні інформації про мережу, механізм динамічної маршрутизації використовує один із протоколів маршрутизації. Такий протокол визначає набір правил, що використовуються маршрутизатором при здійсненні зв'язку із сусідніми маршрутизаторами. Протокол маршру-

тизації визначає [1, 5, 10, 12, 16]: яким чином розсилаються оновлення маршрутів; яка інформація міститься в оновленнях; як часто розсилаються оновлення; яким чином виконується пошук отримувачів оновлень.

Кожен із алгоритмів маршрутизації використовує свій власний спосіб вибору найкращого шляху. Для цього він генерує певне значення, що називається *метрикою* для кожного маршруту у мережі. Зазвичай чим менше значення метрики, тим кращим вважається маршрут [1].

Метрики обчислюються на основі одного або більше параметрів:

- **смуга пропускання** – описує пропускну здатність каналу;
- **затримка** – час, який потрібен пакета для проходження по каналу від відправника до отримувача;
- **навантаження** – ступінь використання мережевих ресурсів на маршрутизаторі чи каналі;
- **надійність** характеризує рівень помилок у мереженому каналі;
- **кількість переходів** – число маршрутизаторів, через які повинен пройти пакет перед надходженням до пункту призначення;
- **вартість –** довільне значення, розраховується на основі ширини смуги пропускання, фінансових затрат або інших характеристик, які обирає мережевий адміністратор.

Отже, протокол маршрутизації – засіб комунікації між маршрутизаторами, яке дозволяє пристроям сумісно використовувати інформацію про мережі та визначати відстань до різних вузлів та мереж. Інформація, яку один маршрутизатор отримує від другого (шляхом протоколу маршрутизації), використовується для побудови та підтримки в актуальному стані ТМ.

Більшість алгоритмів маршрутизації може бути віднесено до однієї із двох категорій [1, 5, 16]:

- дистанційно-векторний протокол (ДВП);
- протокол з врахуванням стану каналу (ПСК).

Дистанційно-векторний протокол визначає напрям або вектор та відстань до потрібного вузла об’єднаної мережі. Прикладами таких протоколів є RIP, IGRP, EIGRP, BGP. Деякий час протокол EIGRP вважався гібридним протоколом, оскільки поєднує у собі особливості обох алгоритмів: дистанційно-векторного та з врахуванням стану каналу, але на сьогоднішній день фірма Cisco відносить його до ДВП. Хоча варто зазначити, що він має набагато кращі характеристики, ніж класичні ДВП [4, 15].

Протокол з врахуванням стану каналу, який також ще називають алгоритмом вибору найкоротшого шляху (shortest path first – SPF), відтворює топологію усієї мережі. Приклади: OSPF, IS-IS, NLSP.

При використанні дистанційно-векторних алгоритмів між маршрутизаторами періодично пересилаються копії таблиць маршрутизації. В таких регулярних оновленнях маршрутизатори повідомляють один одного про зміни у топології мережі. Дистанційно-векторні алгоритми маршрутизації

також називаються ще алгоритмами Беллмана-Форда. На рис. 4.1 кожен маршрутизатор отримує ТМ від сусідніх маршрутизаторів.

Маршрутизатор Б отримує таблицю від маршрутизатора А. Маршрутизатор додає значення вектора відстані, кількість переходів, що збільшує результуючий вектор відстані. Після цього маршрутизатор Б передає свою нову таблицю маршрутизації своєму сусіду маршрутизатору В. Такий покроковий процес відбувається на всіх сусідніх маршрутизаторах.

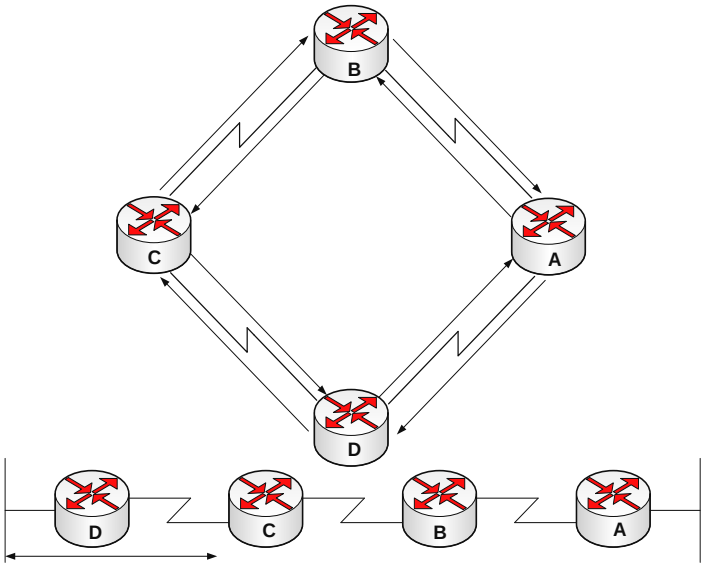


Рисунок 4.1 – Концепція дистанційно-векторної маршрутизації

В дистанційно-векторному алгоритмі накопичуються відстані в мережі, що дозволяє підтримувати базу даних (БД), яка містить інформацію про топологію мережі. Однак дистанційно-векторні алгоритми не надають маршрутизаторам точну топологію всієї мережі, оскільки кожному маршрутизатору відомі лише сусідні (прилеглі) маршрутизатори.

Кожен маршрутизатор, що використовує дистанційно-векторну маршрутизацію, починає свою роботу з визначення сусідніх маршрутизаторів. На рис. 4.2 проілюстровано формування вектора відстані. Для кожного інтерфейсу безпосередньо під’єднаної мережі, вектор відстані встановлюється нульовим. В процесі розрахунку вектора відстані, маршрутизатори знаходять найкращий маршрут до сусідів-отримувачів на основі інформації, отриманої від сусідів. Наприклад, маршрутизатор А взнає про інші мережі на основі інформації, яку він отримує від маршрутизатора Б. В кожній із позицій ТМ є сумарний вектор відстані, який показує, на якій відстані знаходиться відповідна віддалена мережа [1, 5, 15, 16].

Оновлення ТМ відбувається при зміні топології мережі. В міру формування векторів відстані зміни топології заносяться в ТМ наступних маршрутизаторів. Дистанційно-векторні алгоритми потребують, щоб кожен маршрутизатор пересилав всю ТМ кожному із своїх сусідів.

Вектор відстані можна порівняти з дорожними знаками на шосе. Ці знаки вказують напрям до пункту призначення та відстань до нього. Далі по цьому ж шосе можуть зустрічатися знаки, що вказують той самий напрям, однак, відстань вказувати вони будуть меншу. Зменшення цієї відстані при русі свідчить про правильний напрям руху.

Другим базовим алгоритмом маршрутизації є алгоритм вибору маршруту за станом каналу. Такі алгоритми відомі, як алгоритми Дейкстри або алгоритми вибору найкоротшого шляху (Shortest Path First). Вони підтримують складну базу топологічної інформації. Тоді як дистанційно-векторні алгоритми не містять певної інформації про віддалені мережі та маршрутизатори, алгоритми з використанням стану каналу підтримують повну інформацію про віддалені маршрутизатори та їх з'єднання. Під час маршрутизації за станом каналу використовуються [1, 5, 16]:

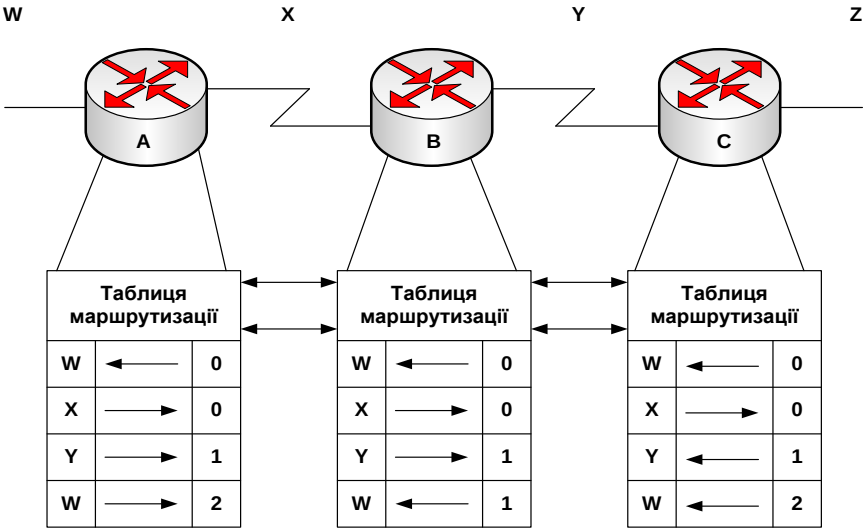


Рисунок 4.2 – Процес побудови структури мережі в дистанційно-векторному протоколі маршрутизації

- **анонси стану каналу** (Link-State Advertisement – LSA). Це невеликі пакети, що містять інформацію про маршрути, що розсилаються між маршрутизаторами;
- **топологічна база даних** (Topological Database). Ця база містить інформацію, отриману в повідомленнях LSA;
- **алгоритм вибору найкоротшого шляху** (Shortest Path First). Відповідний алгоритм здійснює обчислення над базою даних, результатом якого є побудова зв'язного дерева протоколу SPF;
- **таблиця маршрутизації** (Routing Table). Ця таблиця містить відомі маршрути та відповідні їм інтерфейси.

Маршрутизатори обмінюються повідомленнями LSA, починаючи з безпосередньо під'єднаних мереж. Кожен маршрутизатор паралельно з іншими створює топологічну БД, яка складається з інформації, що отримана з цих повідомлень (рис. 4.3). Якщо маршрутизатор взнає про зміну стану каналу, він розсилає цю інформацію всім іншим маршрутизаторам об'єднаної мережі, щоб вони могли її використовувати для маршрутизації. Для того, щоб закінчилась конвергенція, кожен маршрутизатор підтримує інформацію про сусідні маршрутизатори, їх імена, стани інтерфейсів та вартості каналів до сусідніх пристроїв. Маршрутизатор створює пакет LSA, в якому міститься перерахована інформація з інформацією про нових сусідів, зміни у вартостях каналів і про канали, що перестали функціонувати. Потім цей пакет LSA відправляється всім іншим маршрутизаторам.

4.2 Застосування кількох протоколів маршрутизації

В одній мережі можуть одночасно бути кілька різних протоколів маршрутизації. Оскільки інформація про мережу може надійти від кількох протоколів і містити різні раціональні маршрути – встановлюють пріоритети протоколів маршрутизації. Зазвичай перевагу надають LSA протоколам, оскільки вони мають повнішу інформацію [1].

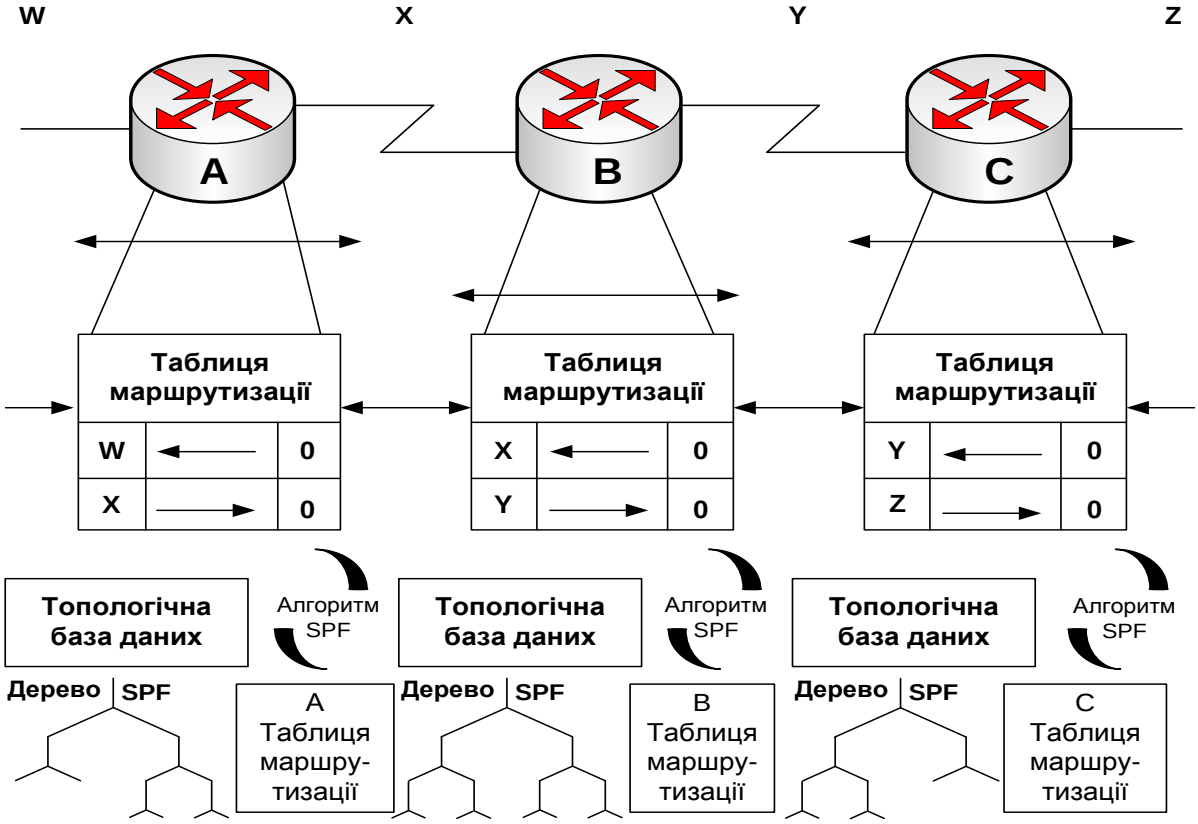


Рисунок 4.3 – Обчислення найкоротшого шляху

За замовчуванням кожен протокол маршрутизації на певному маршрутизаторі розповсюджує лише ту інформацію, котра була отримана маршрутизатором цим же протоколом. Оскільки не завжди кожен маршрутизатор підтримує усі протоколи для даної мережі, то застосовують внутрішній особливий режим роботи маршрутизотора, який називають перерозподілом (Redistribute).

Зазначимо, що зі зростанням мереж проблема взаємодії маршрутизаторів дуже зростає і для її розв’язання було знайдено інший підхід – поділ мережі на автономні системи.

4.3 Внутрішні та зовнішні протоколи Інтернету

Internet це всесвітня система добровільно об'єднаних КМ, побудована на використанні протоколу IP та маршрутизації пакетів [1, 14]. З самого початку Internet будувалась як мережа, що об'єднувала велику кількість існуючих систем. В її структурі визначають *магістральну мережу (core backbone network)*, а мережі, під'єднані до магістралі, розглядаються як *автономні системи (autonomous systems, AS)*.

Автономна система (АС) – це сукупність мереж, які знаходяться під єдиним адміністративним керуванням і в яких використовується єдина стратегія і правила маршрутизації. АС для зовнішніх мереж є єдиним об'єктом [1, 5, 15, 16].

Кожна АС повинна мати свій власний унікальний номер (Autonomous System Number – ASN). Номери виділяються організацією Internet Assigned Numbers Authority (IANA), яка також виділяє IP-адреси регіональним інтернет-реєстраторам (Regional Internet Registry, RIR) блоками. Локальні RIR після цього присвоюють організаціям номер АС з блоку, отриманого від IANA. Організації, що бажають отримати ASN, повинні пройти процес реєстрації в своєму локальному RIR та отримати схвалення.

Раніше використовувались 16-бітні номери АС, що дозволяло виконати максимум 65536 присвоювань. Сьогодні вже використовуються 32-бітні ASN, що дозволяє адресувати максимум 2^{32} автономних систем.

АС поділяють об'єднану КМ на декілька менших та легше керованих мереж. Кожна АС має свій набір правил та політик, а її номер є глобально унікальним, тобто відрізняє її від усіх інших автономних систем світу.

Шлюзи, що використовуються для утворення мереж та підмереж всередині автономної системи, називають *внутрішніми шлюзами (interior gateways)*, а шлюзи за допомогою яких автономні системи під'єднуються до магістралі мережі, відповідно називають *зовнішніми шлюзами (exterior gateways)*. Сама магістраль також є АС. Згідно з цими визначеннями протоколи маршрутизації також поділяють на два види: протокол внутрішнього та протокол зовнішнього шлюзу (рис. 4.4) [5].

Протокол внутрішнього шлюзу (interior gateway protocol, IGP) призначений для використання у мережі, що керується або адмініструється окремою організацією. Такий протокол служить для знаходження найкращого маршруту в одній мережі. Іншими словами, метрика та характер її використання є найбільш важливими елементами протоколу IGP.

Протокол зовнішнього шлюзу (exterior gateway protocol – EGP) призначений для здійснення маршрутизації між мережами, що знаходяться під управлінням різних організацій. Як правило, ці протоколи використовуються при маршрутизації між провайдерами служб Internet (Internet Service Providers, ISP) або між окремою компанією та Internet-провайдером. Протокол EGP повинен ізолювати АС. Оскільки в кожній АС використовують-

ся свої правила, в об'єднаній мережі повинен функціонувати загальний протокол, який дозволить здійснювати зв'язок між ними.

Рисунок 4.4 – Види протоколів маршрутизації

4.4 Порівняння статичної та динамічної маршрутизації

Статична маршрутизація має такі особливості [1 – 3, 15]:

- забезпечує підтримку ТМ для невеликих мереж, які не передбачено суттєво розширювати;
- забезпечує маршрутизацію для кінцевої (тупикової) мережі;
- задає єдиний маршрут за замовчуванням до будь-якої мережі, якщо ТМ не містить більш специфічного шляху.

Отже, аналізуючи вищесказане, можно навести переваги та недоліки статичної і динамічної маршрутизації.

Переваги статичної маршрутизації: мінімальне використання процесора; легша для розуміння адміністратора; легша для конфігурування.

Недоліки статичної М: конфігурування та обслуговування потребує багато часу; під час конфігурування можливі помилки (особливо у великих мережах); для підтримки заміни маршрутної інформації потрібне втручання адміністратора; зі зростанням мережі погано масштабується; для належного виконання потребує повного знання усієї мережі.

Переваги динамічної маршрутизації: потребує меншого втручання адміністратора, при доданні або вилученні мереж; протоколи автоматично реагують на зміни топології; конфігурація менш схильна до помилок; більш масштабована, нарощування мережі зазвичай не породжує проблем.

Недоліки динамічної маршрутизації: використовуються ресурси маршрутизатора; потребує більших знань адміністратора для конфігурування, перевірки та усунення несправностей.

Таблиця 4.1 – Порівняння статичної та динамічної маршрутизації

Критерій порівняння	Статична маршрутизація	Динамічна маршрутизація
Складність конфігурування	Ускладнюється зі зростанням складності мережі	В загальному плані не залежить від складності мережі
Вимоги до знань адміністратора	Потрібен невисокий рівень знань	Потрібен більший рівень знань
Зміни Топології	Потрібне адміністративне втручання	Автоматично адаптується під зміни
Масштабування	Підходить лише для простих топологій	Підходить і для складних і для простих топологій
Ступінь безпеки	Більш безпечна, ніж динамічна маршрутизація	Не гарантує безпеки
Ступінь застосовування ресурсів	Не потребує додаткових ресурсів	Застосовує процесор, оперативну пам'ять, смугу пропускання
Передбачуваність	Маршут завжди постійний	Маршрут залежить від поточної топології

4.5 Порівняння деяких протоколів динамічної маршрутизації

Як зазначалось вище, алгоритми динамічної маршрутизації поділяються на дистанційно-векторні (DVA) та стану зв'язків (LSA). В дистанційно-векторних протоколах для знаходження найкрашого шляху використовується алгоритм Беллмана-Форда. Деякі DVA періодично розсилають сусідам повні ТМ, що може породжувати значний трафік. DVA не мають уявлення про топологію усієї мережі. Вони знають про віддалені мережі лише відстань до них і вихідний порт (або адресу наступного хопа).

Протоколи DVA працюють найкраще в ситуаціях, коли [1, 16]: мережа проста і не потребує спеціальної ієрархічної структури; адміністратори не мають достатньо знань щодо вибору конфігурації та підтримки LSA; використовуються специфічні типи мереж, наприклад, мережі типу hub-and-spoke; час конвергенції у найгіршому випадку для мережі непринципові.

В LSA для знаходження найкрашого шляху використовується алгоритм Дейкстри. Маршрутизатори знають про всю КМ шляхом збирання інформації від усіх маршрутизаторів. Кожен маршрутизатор має повну топологічну карту КМ. Всі маршрутизатори КМ використовують одну й ту ж топологічну карту. LSA не здійснюють періодичних оновлень. Після конвергенції КМ оновлення надсилаються лише у випадку змін її топології.

Протоколи LSA працюють найкраще в ситуаціях, коли: КМ велика та ієрархічна; адміністратор має достатньо знань; швидка конвергенція у КМ дуже актуальна.

Порівняння традиційних дистанційно-векторних протоколів з протоколом EIGRP наведено у табл. 4.2, а деяких протоколів динамічної маршрутизації – у табл. 4.3.

Таблиця 4.2 – Порівняння традиційних DVA з EIGRP

Традиційні DVA	EIGRP
Використовують алгоритм Беллмана-Форда або Форда-Фалкерсона	Використовує алгоритм дифузії поновлень маршрутизації (Difussion Update Algorithm)
Використовуються періодичні оновлення і час життя записів ТМ	Не використовуються ні періодичні оновлення, ні час життя записів ТМ
Зберігає лише кращий маршрут до пунктів призначення	Підтримує топологічну таблицю (відмінну від ТМ), яка містить як найкращі, так і резервні шляхи до пунктів призначення
Коли маршрут стає недійсним маршрутизотор повинен чекати до наступного оновлення	Коли маршрут стає недійсним DUAL використовує резервний шлях з топологічної таблиці
Повільна конвергенція завдяки використанню таймера holddown	Швидка конвергенція завдяки відсутності таймера holddown і системи кординування обчислення маршрутів

Таблиця 4.3 – Порівняння деяких протоколів динамічної маршрутизації

Критерій	DVA				LSA	
	RIPv1	RIPv2	IGRP	EIGRP	OSPF	IS-IS
Швидкість конвергенції	Повільна	Повільна	Повільна	Висока	Висока	Висока
Масштабованість (розмір мережі)	Мала	Мала	Мала	Велика	Велика	Велика
Підтримка VLSM	–	+	–	+	+	+
Ступінь використання ресурсів	Низька	Низька	Низька	Середня	Висока	Висока
Впровадження та підтримка	Проста	Проста	Проста	Складна	Складна	Складна

Порівняння OSPF з RIP

Порівняння протоколів OSPF і RIP не зовсім правомірно, оскільки ці два протоколи призначені для мережевого середовища абсолютно різних типів. Протокол OSPF призначений для використання у великих, складних мережах, спроектованих на основі продуманого підходу. Протокол RIP призначений для невеликих мереж, в яких застосування простого протоколу дозволяє спростити проектування і скоротити тривалість настройки конфігурації. По суті, якщо мережа є достатньо невеликою для того, щоб в ній можна було застосовувати протокол RIP, то краще зупинитися на протоколі RIP, а потім перейти на EIGRP [15].

Переваги OSPF у порівнянні з RIP: набагато більш масштабований; підтримує VLSM та CIDR (на відміну від RIPv1); в цілому у достатньо стійких мережах споживає менше мережевих ресурсів; забезпечує вибір кращих маршрутів; дозволяє коректно запобігти маршрутним циклам; характеризується кориснішою метрикою; сприяє створенню ієрархічних проектів мереж; забезпечує швидкий перехід мережі у сталий стан.

Недоліки OSPF у порівнянні з RIP: не допускає використання ієрархічних проектів у поєднанні з погано спроектованими структурами IP; є набагато більш складним порівняно з RIP; потребує більших ресурсів процесора і оперативної пам'яті; потребує більших витрат часу на проектування і реалізацію.

Порівняння OSPF з EIGRP

Протоколи OSPF і EIGRP фактично багато в чому аналогічні. У протоколі EIGRP, як і в OSPF передбачено формування таблиці топології і пошук на її основі маршрутів до одержувачів. Крім того, при звичайних обставинах протокол EIGRP, як і OSPF виключає можливість створення маршрутних циклів. Проте в деяких умовах протокол OSPF є доцільнішим, ніж EIGRP, а в інших – навпаки.

Переваги OSPF у порівнянні з EIGRP: сприяє створенню ієрархічних проектів мереж; має менш складну метрику порівняно із складеною метрикою EIGRP; не схильний до проблем, пов'язаних з постійним перебуванням маршруту в активному стані; не залежить від виробника конкретного продукту.

Недоліки OSPF у порівнянні з EIGRP: метрика не така гнучка, як складена метрика EIGRP; не забезпечує розподілу навантаження по маршрутах з нерівною вартістю; не допускає використання ієрархічних проектів в поєднанні з погано спроектованими структурами IP; потребує більших ресурсів процесора і оперативної пам'яті; потребує більших витрат часу на проектування і реалізацію.

4.6 Основи статичної маршрутизації

Після задання адміністратором статичного маршруту маршрутизатор запам'ятовує його у своїй ТМ і використовує для пересилання пакетів. Команда задання статичного маршруту має такий синтаксис [4]:

`Rt(config)#ip route prefix mask {ip | int-type int-num}[dist],`

де `prefix`, `mask` – IP-адреса та маска пункту призначення, відповідно; `ip`, `int-type`, `int-num` – IP-адреса порту наступного транзитного переходу (хопа), тип та номер локального інтерфейсу на які слід надіслати пакет, котрий повинен дістатися вищевказаного пункту призначення; `dist` – адміністративна відстань. *Адміністративна відстань* (АВ) – це необов'язковий параметр, який характеризує надійність маршруту. Чим менша АВ, тим надійнішим є маршрут. Маршрут з меншою адміністративною відстанню бу-

де занесений у ТМ.

Конфігурування статичних маршрутів

Для конфігурування статичних маршрутів слід виконати такі кроки [1].

1. Визначити усі мережі-отримувачі, їх маски та шлюзи (як адресу шлюзу можна вказати або локальний інтерфейс маршрутизатора або адресу наступного хопа, на шляху до потрібного пункту призначення).

2. Увійти в режим глобального конфігурування.

3. Ввести команду `ip route` з відповідними параметрами як показано вище.

4. Повторити третій крок для усіх мереж-отримувачів, до яких слід задати статичний маршрут.

5. Вийти з режиму глобального конфігурування.

6. Виконати команду `copy running-config startup-config`.

Наприклад, для мережі наведеної на рис. 4.5 команди задання статичних маршрутів на прикладі маршрутизатора R2 будуть

```
R2(config)#ip route 192.168.1.0 255.255.255.0 192.168.4.1,
R2(config)#ip route 192.168.3.0 255.255.255.0 192.168.4.6,
```

(статичні маршрути до усіх інших мереж 192.168.2.0/24, 192.168.4.0/30, 192.168.4.4/30 маршрутизатор R2 знає, оскільки вони безпосередньо під'єднані до нього). У вищенаведених командах вказано IP-адреси наступних хопів на шляху до отримувачів. Якщо вказувати вихідні інтерфейси, команди задання статичних шляхів набудуть вигляду

```
R2(config)#ip route 192.168.1.0 255.255.255.0 s0/1,
R2(config)#ip route 192.168.3.0 255.255.255.0 s0/0.
```

Зауважимо, що дані та дві попередні команди для даного випадку еквівалентні. Єдина відмінність між ними полягає в тому, що будуть різні значення AB. Стандартно, при застосуванні адреси наступного переходу AB = 1, а вихідного інтерфейсу – AB = 0.

Взагалі значення AB – цілі числа в діапазоні від 0 до 255. Якщо треба ввести нестандартну адміністративну відстань (наприклад, яка дорівнює 140) то слід задати команду

```
R3(config)#ip route 192.168.1.0 255.255.255.0
192.168.4.5 140
```

Якщо маршрутизатор з деяких причин не може використовувати вихідний інтерфейс, заданий у маршруті – то цей маршрут не буде використовуватись, тобто не буде занесено до ТМ.

Іноді статичні маршрути використовують як резервні, котрі будуть використовуватись лише у випадку, якщо не вдається надіслати дані за динамічним маршрутом. В такому випадку AB повинна бути більшою, ніж у маршруту, отриманого протоколом динамічної маршрутизації.

Зазначимо, що на маршрутизаторі R1 аналогічно слід прописати шляхи до мереж 192.168.2.0/24, 192.168.3.0/24 та 192.168.4.4/30, а на R3 – до мереж 192.168.1.0/24, 192.168.2.0/24 та 192.168.4.0/30. Проте замість того, щоб прописувати ці три шляхи, на маршрутизаторах R1 та R3 можна вказати лише по одному маршруту за замовчуванням.

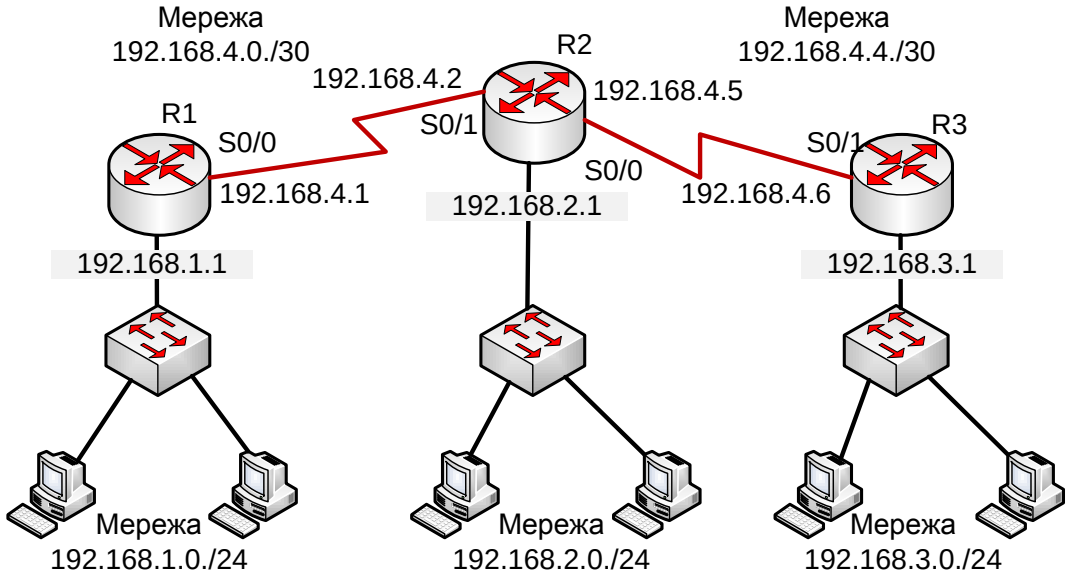


Рисунок 4.5 – Проста комп'ютерна мережа

Задання маршруту за замовчуванням

Маршрути за замовчуванням (або стандартні маршрути) використовуються маршрутизаторами у випадку, коли адреса мережі-отримувача пакета не збігається з жодним маршрутом ТМ. Стандартні маршрути, як правило, конфігуруються для передавання потоків даних через мережу Internet, оскільки нераціонально і немає необхідності підтримувати усі маршрути до всіх мереж Internet. Таким чином, стандартні маршрути дозволяють скоротити число записів у ТМ і зменшити час їх оброблення [4].

Задати стандартний маршрут можна за допомогою команди

```
ip route 0.0.0.0 0.0.0.0 {ip | int-type int-num}.
```

Операція логічного „І" над маскою 0.0.0.0 та IP-адресою пакета завжди дає результатом мережу 0.0.0.0. Якщо для пакета в ТМ не знаходиться відповідності мережі-отримувача – він надсилається у мережу 0.0.0.0.

Так, повертаючись до попереднього прикладу (див. рис. 9.5), команда задання маршруту за замовчуванням для маршрутизатора R1 буде

```
R1(config)# ip route 0.0.0.0 0.0.0.0 s0/0,
```

а для R3: `R3(config)# ip route 0.0.0.0 0.0.0.0 s0/1.`

Перевірка і усунення помилок у конфігурації статичних маршрутів

Після того, як статичні маршрути сконфігуровані – слід впевнитись, що вони є у ТМ і пересилка пакетів за ними виконується правильно. Для цього можна використати команди `show running-config` та `show ip route`. Перша команда дозволяє проглянути статичні маршрути у файлі робочого конфігурування маршрутизатора, а друга – у його ТМ. При цьому, якщо деякий маршрут введено неправильно – його слід вилучити, а замість ввести правильний.

Для пошуку та усунення помилок в конфігуруванні статичних маршрутів пропонується виконати такі кроки [4].

1. Впевнитись, що канал, який буде використовуватись як шлюз є доступним.

2. Виконати команду `show interfaces` і впевнитись у активності інтерфейсу і канального протоколу.

3. Перевірити правильність IP-адреси на інтерфейсі.

4. Виконати команду `ping` для IP-адреси віддаленого маршрутизатора, безпосередньо під'єднаного до шлюзу маршруту. Якщо результат цієї команди буде негативний – то проблема не пов'язана з маршрутизацією.

5. Якщо команда не спрацьовує на дальньому маршрутизаторі – слід виконати команду `traceroute` – для визначення вузла, де губиться пакет.

6. Під'єднатись до маршрутизатора, на якому не спрацювало трасування і виконати дії, описані у першому кроці.

7. Якщо команда `ping` спрацювала на дальньому кінці маршруту – тест можна вважати успішним і завершеним.

4.7 Дистанційно-векторний протокол RIP

4.7.1 Побудова таблиці маршрутизації

Протокол маршрутної інформації (Routing Information Protocol – RIP) початково був визначений в документі RFC 1058 в 1988 році. Найбільш суттєвими є такі його характеристики [1, 10]:

- RIP є дистанційно-векторним протоколом маршрутизації;
- як метрики при виборі маршруту використовується кількість переходів (або хопів);
- якщо кількість переходів більше ніж 15, пакет відкидається;
- стандартно оновлення маршрутизації розсилаються широкомовним способом кожних 30 секунд.

Протокол RIP значно еволюціонував: від основаного на класах протоколу першої версії (RIPv1) до безкласового протоколу другої версії (RIPv2). Вдосконалення останнього такі [1, 9, 10]:

- можливість переносити додаткову інформацію про маршрутизацію

пакетів;

- механізм аутентифікації для забезпечення безпечного оновлення ТМ;
- підтримка масок змінної довжини.

Протокол RIP запобігає появі петель маршрутизації, по яких пакети могли б циркулювати невизначено довго, встановлюючи максимально допустиму кількість переходів на маршруті між відправником та отримувачем. Стандартне максимальне значення кількості переходів становить 15. При отриманні маршрутизатором оновлення маршрутів, що містить новий або змінений запис, він збільшує значення метрики на одиницю. Якщо при цьому значення метрики перевищує 15, то мережа-отримувач вважається недосяжною. У протоколу RIP є ряд функцій спільних для нього та інших протоколів маршрутизації. Наприклад, він дозволяє використовувати механізми розщеплення горизонту та таймери утримання інформації для запобігання розповсюдження некоректних знань про маршрути.

Розглянемо процес побудови таблиці маршрутизації за допомогою протоколу RIPv1 на прикладі мережі, зображеної на рис. 4.6 [1].

Етап 1 – створення мінімальних таблиць маршрутизації

В даній мережі містяться вісім IP-мереж, зв'язаних чотирма маршрутизаторами з ідентифікаторами: M1, M2, M3 та M4. Маршрутизатори, що працюють за протоколом RIP, можуть мати ідентифікатори, однак, для роботи протоколу вони не є необхідними. В RIP-повідомленнях ці ідентифікатори не передаються.

У вихідному стані в кожному маршрутизаторі програмним забезпеченням стека TCP/IP автоматично створюється мінімальна таблиця маршрутизації, в якій враховуються лише безпосередньо під'єднані мережі.

Табл. 4.4 дозволяє оцінити приблизний вигляд мінімальної ТМ маршрутизатора M1.

Таблиця 4.4 – Мінімальна таблиця маршрутизації маршрутизатора M1

Номер мережі	Адреса наступного маршрутизатора	Порт	Відстань
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1

Мінімальні ТМ інших маршрутизаторів будуть виглядати відповідно.

Етап 2 – розсилання мінімальних таблиць сусідам

Після ініціалізації кожного маршрутизатора він починає відсилати своїм сусідам повідомлення протоколу RIP, в яких міститься його мінімальна ТМ. RIP-повідомлення надсилаються в пакетах протоколу UDP і містять для кожної мережі: її IP-адресу та відстань до неї від маршрутизатора, що надсилає повідомлення. Сусідами є маршрутизатори, яким даний маршрутизатор може надіслати IP-пакет не користуючись послугами

проміжних маршрутизаторів. Маршрутизатор М1 надсилає до М2 і М3 повідомлення: мережа 201.36.14.0, відстань 1; мережа 132.11.0.0, відстань 1; мережа 194.27.18.0, відстань 1.

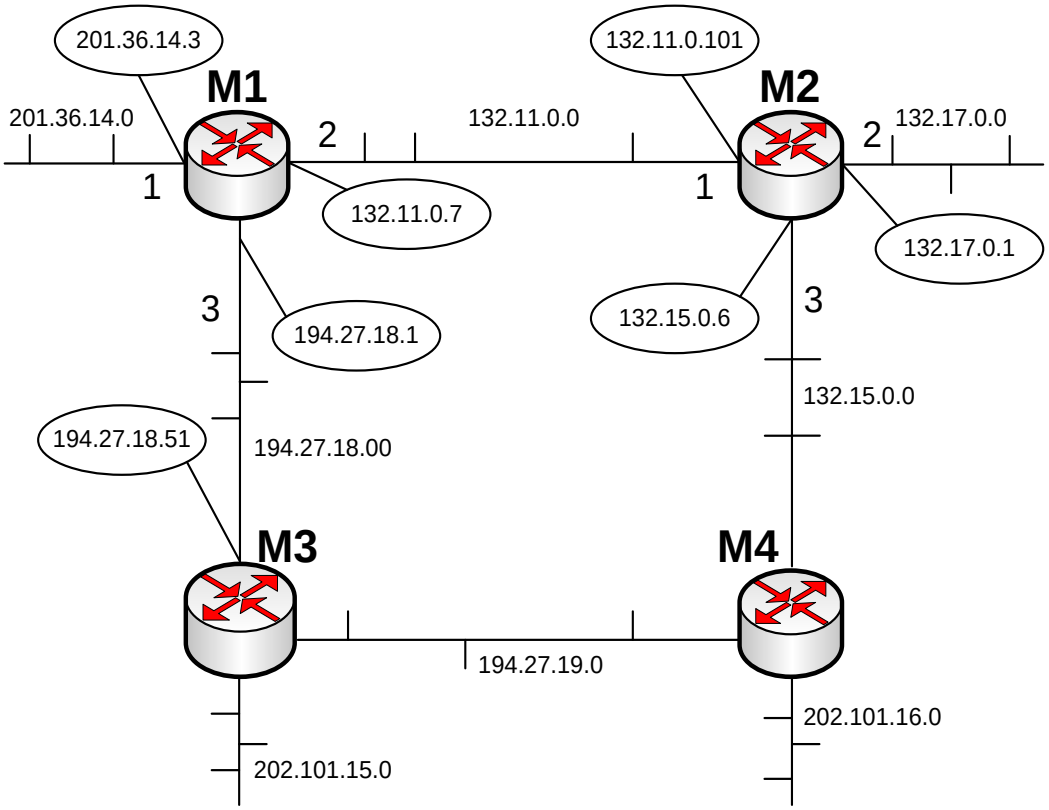


Рисунок 4.6 – Мережа, об’єднана RIP-маршрутизаторами

Етап 3 – отримання RIP-повідомлень від сусідів і оброблення отриманої інформації

Після отримання аналогічних повідомлень від маршрутизаторів М2 та М3 маршрутизатор М1 нарощує кожне отримане поле метрики на одиницю і запам’ятовує, через який порт і від якого маршрутизатора отримана ця інформація (адреса цього маршрутизатора буде адресою наступного хопа, якщо цей запис буде внесений до ТМ). Потім маршрутизатор починає порівнювати нову інформацію з тією, що зберігається в його ТМ (табл. 4.5).

Таблиця 4.5 – Таблиця маршрутизації маршрутизатора М1

Номер мережі	Адреса наступного маршрутизатора	Порт	Відстань
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1
132.17.0.0	132.11.0.101	2	2
132.15.0.0	132.11.0.101	2	2
194.27.19.0	194.27.18.51	3	2
202.101.15.0	194.27.18.51	3	2
~~132.11.0.0~~	~~132.11.0.101~~	~~2~~	~~2~~
~~194.27.10.0~~	~~194.27.10.51~~	~~3~~	~~2~~

Записи 4 – 9 отримані від сусідніх маршрутизаторів і претендують на занесення до ТМ. Однак тільки записи з 4 –7 потрапляють до неї, оскільки два останні – містять дані про вже наявні у таблиці М1 мережі, а відстань до них більша, ніж в існуючих записах.

Протокол RIP заміщує запис про будь-яку мережу лише тоді, якщо нова інформація має кращу метрику, ніж наявна. В результаті в ТМ про кожну мережу лишається лише один запис. Якщо є кілька рівнозначних за метрикою шляхів до однієї і тієї ж мережі, то в ТМ лишається один запис, що надійшов першим. Для цього правила є виняток – якщо гірша інформація про будь-яку мережу прийшла від того ж маршрутизатора, на основі повідомлення якого була створено даний запис, то вона заміщує кращу [1].

Аналогічні операції з новою інформацією виконують й інші маршрутизатори мережі.

Етап 4 – розсилання нової таблиці сусідам

Кожен маршрутизатор відсилає нове RIP-повідомлення всім своїм сусідам. В цьому повідомленні він розміщує дані про всі відомі йому мережі – як безпосередньо під'єднаних, так і віддалених.

Етап 5 – отримання RIP-повідомлень від сусідів та оброблення отриманої інформації

Етап 5 фактично повторює етап 3. Розглянемо, як це робить маршрутизатор М1 (табл. 4.6).

Таблиця 4.6 – Таблиця маршрутизації маршрутизатора М1

Номер мережі	Адреса наступного маршрутизатора	Порт	Відстань
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1
132.17.0.0	132.11.0.101	2	2
132.15.0.0	132.11.0.101	2	2
~~132.15.0.0~~	~~194.27.10.51~~	~~3~~	~~3~~
194.27.19.0	194.27.18.51	3	2
~~194.27.19.0~~	~~132.11.0.101~~	~~2~~	~~3~~
~~202.101.15.0~~	~~194.27.18.51~~	~~3~~	~~2~~
202.101.16.0	132.11.0.101	2	3
~~202.101.16.0~~	~~194.27.10.51~~	~~3~~	~~3~~

На цьому етапі маршрутизатор М1 отримав від М3 інформацію про мережу 132.15.0.0, яку той в свою чергу на попередньому циклі роботи отримав від М4. Маршрутизатор вже знає про мережу 132.15.0.0, причому стара інформація має кращу метрику, ніж нова, тому ця нова інформація відкидається.

Про мережу 202.101.16.0 маршрутизатор М1 дізнається на цьому етапі вперше, причому дані про неї приходять від двох сусідів – від М3 та М4. Оскільки метрики в цих повідомленнях однакові, то в ТМ потрапляють дані, які прийшли першими. В нашому прикладі вважається, що маршрутиза-

тор М2 випередив М3 і першим надіслав RIP-повідомлення до М1.

Якщо маршрутизатори періодично повторюють етапи розсилки та оброблення RIP-повідомлень, то за певний проміжок часу в мережі встановлюється коректний режим маршрутизації, коли коли всі мережі будуть досяжні з будь-якої мережі за допомогою деякого раціонального маршруту. Пакети будуть доходити до адресатів і не зациклюватися в петлях.

4.7.2 Методи боротьби з фальшивими маршрутуми в протоколі RIP

Основними методами боротьби за фальшивими маршрутуми в протоколі RIP є розщеплення горизонту, вилучення маршруту в зворотному напрямі, миттєві оновлення, таймери утримання інформації [1, 4].

Маршрутні петлі можуть виникати у тому випадку, якщо для протоколу маршрутизації характерна повільна конвергенція після змін в мережі або для топології мережі в маршрутизаторах виникла невідповідність між записами ТМ. На рис. 4.7 показано петлі маршрутизації. Їх виникнення відбувається так [1, 4, 9, 10].

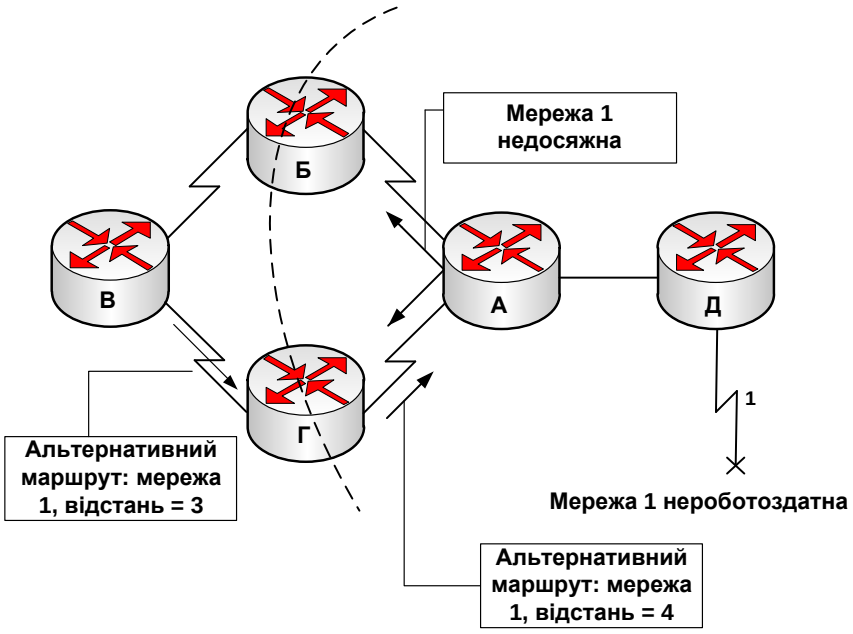


Рисунок 4.7 – Петлі маршрутизації

1. Перед виходом з ладу мережі 1 всі маршрутизатори мають узгоджені та коректні ТМ. В цьому випадку говорять, що в мережі відбулася конвергенція. До кінця цього прикладу вважається, що для маршрутизатора В найкращий маршрут до мережі 1 проходить через маршрутизатор Б, а відстань (метрика) від маршрутизатора В до мережі 1 дорівнює 3.

2. Якщо мережа 1 виходить з ладу, то маршрутизатор Д надсилає повідомлення про оновлення маршрутів маршрутизатору А. Після його отримання маршрутизатор А припиняє надсилати пакети у мережу 1, однак, маршрутизатори Б, В та Г продовжують, оскільки вони ще не інформовані про збій в мережі 1. Коли маршрутизатор А надсилає повідомлення про оновлення, маршрутизатори Б та Г припиняють надсилання пакетів у мережу 1. Однак в цей момент маршрутизатор В ще не отримав повідомлення про оновлення. Для нього мережа, як і раніш, вважається досяжною через маршрутизатор Б.

3. Припустимо, що маршрутизатор В відправляє періодичне оновлення маршрутів маршрутизатору Г, вказуючи маршрут до мережі 1 через маршрутизатор Б. Маршрутизатор Г змінює свою ТМ для того, щоб врахувати таку некоректну інформацію, і надсилає цю інформацію маршрутизатору А, який надсилає її маршрутизаторам Б та Д і т. д. Тепер будь-який пакет, призначений для мережі 1, рухається по кільцевому маршруту (петлі) від маршрутизатора В до маршрутизатора Б, далі до А і Г та знову до маршрутизатора В.

Некоректні відомості про мережу 1 продовжують циркулювати по кільцевому маршруту доти, поки будь-який інший процес це не припинить. При такому стані мережі, яке називають зациклюванням (count to infinity), пакети продовжують неперервно рухатись мережею, незважаючи на вихід з ладу мережі-отримувача. І поки маршрутизатор збільшує кількість переходів (потенційно до нескінченності), неправильна інформація допускає існування петлі (рис. 4.8) [4].

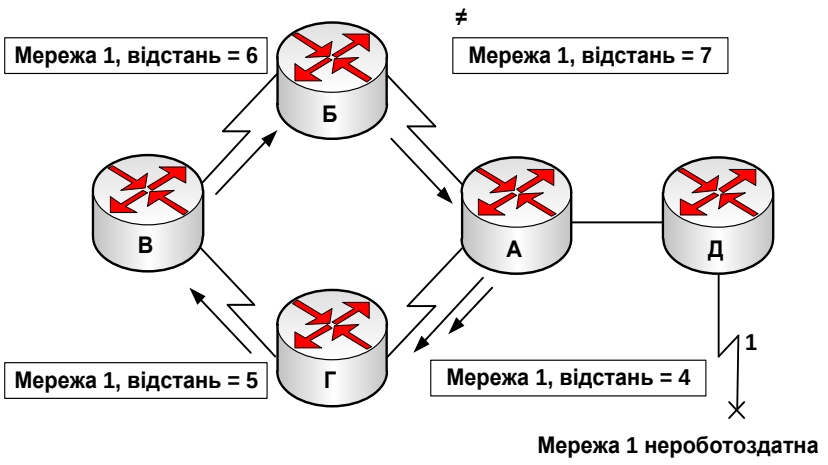


Рисунок 4.8 – Зациклювання

Якщо не будуть прийняті певні заходи для зупинення цього процесу, вектор відстані або метрика, що відображає кількість переходів, будуть зростати при кожному проходженні пакета через черговий маршрутизатор. Таким чином, пакети рухаються по колу внаслідок того, що в ТМ міститься помилкова інформація.

Дистанційно-векторні алгоритми маршрутизації мають здатність до самокорекції, однак, для усунення петлі в маршрутизації та проблеми зациклювання потрібні спеціальні заходи. Для того, щоб уникнути проблеми зациклю-

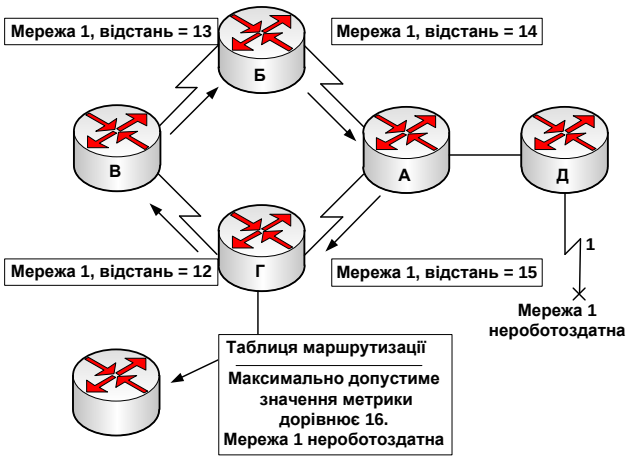


Рисунок 4.9 – Задання максимального значення метрики

вання, нескінченність визначається як деяке скінченне число. Для протоколу RIP таким числом є 16. Тепер протокол маршрутизації дозволяє петлі існувати лише до того моменту, доки метрика не перевищить 16. На рис. 4.9 показано, що значення метрики досягло шістнадцяти; оскільки вектор відстані перевищив стандартний максимум в 15 транзитних переходів, пакет маршрутизатором відкидається. В будь-якому випадку, коли значення метрики перевищує максимально допустиме, мережа 1 є вважається недосяжною.

Тепер подивимось, що відбувається з іншими IP-пакетами, які не є повідомленнями протоколів маршрутизації, коли виникає петля. Зрозуміло, що пакети будуть передаватись від одного маршрутизатора другому по колу. В протоколі IP є свій власний механізм попередження нескінченної циркуляції пакетів по колу – поле TTL (Time-to-Live – час існування пакета). Перед тим як IP-пакет буде переданий вузлом в це поле згідно з стандартом може бути записане значення між 1 та 255. Коли такий пакет надходить до маршрутизатора, пристрій зменшує значення в полі TTL на одиницю. Коли значення TTL досягає нуля, маршрутизатори зобов'язані відкинути такий IP-пакет та переслати відправнику відповідне інформаційне ICMP-повідомлення. Такий механізм усуває можливість нескінченної циркуляції IP-пакетів в мережі та допомагає вирішити проблему кільцевих маршрутів.

Друге можливе джерело петлі в маршрутизації виникає у випадку, коли маршрутизатору надіслана інформація, що суперечить правильній, яку він спочатку розповсюдив. Як показано на рис. 4.10, при цьому відбувається описаний нижче процес, який і створює проблему петлі маршрутизації [1, 4, 16].

1. Маршрутизатор А надсилає маршрутизаторам Б та Г оновлення, в якому вказується, що мережа 1 не працює.

2. Однак маршрутизатор В передає маршрутизатору Б інше повідомлення, в якому вказується, що мережа 1 доступна через маршрутизатор Г з відстанню, що дорівнює чотирьом переходам. Така дія не порушує правил розщеплення горизонту.

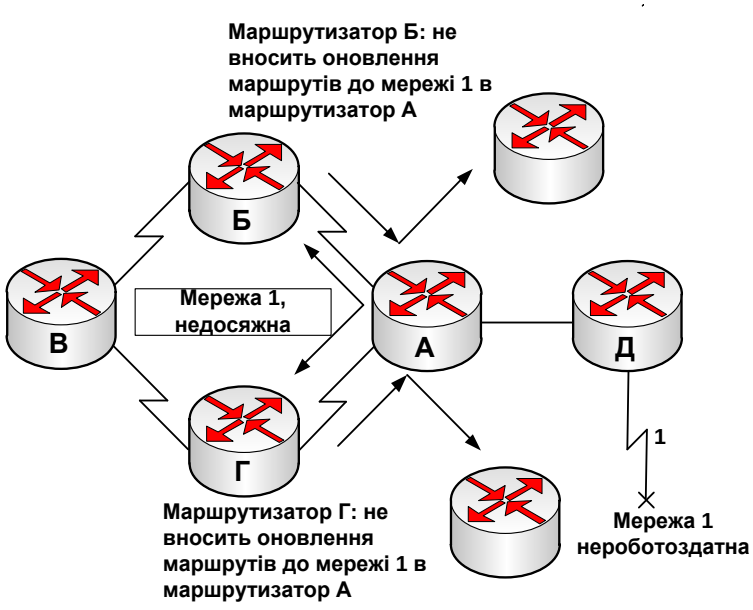


Рисунок 4.10 – Розщеплення горизонту

3. Після отримання останнього повідомлення маршрутизатор Б неправильно робить висновок, що у маршрутизатора В як і раніше є дійсний маршрут до мережі 1. Маршрутизатор Б відсилає повідомлення про оновлення маршрутизатору А, повідомляючи його про новий маршрут до мережі 1.

4. Отримавши його, пристрій А робить висновок, що він може надсилати інформацію у мережу 1 через маршрутизатор Б. Маршрутизатор Б вирішує, що він може надсилати інформацію у мережу 1 через маршрутизатор Г. В такій ситуації будь-який пакет буде рухатись по кільцевому маршруту між цими маршрутизаторами.

Розщеплення горизонту *(split horizon)* намагається запобігти виникненню такої ситуації. Згідно з цим методом, при надходженні повідомлення про оновлення маршрутів для мережі 1 від маршрутизатора А маршрутизатори Б та Г не можуть посилати інформацію про мережу 1 в зворотному напрямі, тобто маршрутизатору А, як показано на рис. 4.10. Таким чином, розщеплення горизонту не дозволяє розповсюджувати неправильну інформацію маршрутизації та зменшує об'єм службових повідомлень, що передаються.

Вилучення маршруту в зворотному напрямку *(route poisoning)* використовується різними ДВП для запобігання виникненню великих петель маршрутизації і наданні явної інформації про маршрути в тих випадках, коли мережа недосяжна. Таке вилученя маршруту зазвичай здійснюється шляхом встановлення кількості переходів на одиницю більшою, ніж максимальне значення. Цей механізм є альтернативним способом попередження петель маршрутизації. Даний підхід може бути сформульовано так: після отримання інформації про маршрут через будь-який інтерфейс необхідно оголосити його недосяжним через цей самий інтерфейс. Краще явно повідомити маршрутизатор про те, що маршрут потрібно ігнорувати, ніж лишити все неконтрольованим.

Припустимо, що на всіх маршрутизаторах на рис. 4.11 увімкнено механізм зворотного вилучення маршрутів. Після отримання інформації маршрутизатором 1 про мережу А від маршрутизатора 2 пристрій 1 оголошує мережу А недосяжною через свої канали до маршрутизаторів 2 та 3. Якщо маршрутизатор 3 має будь-який маршрут до мережі А через маршрутизатор 1, він видаляє цей маршрут, оскільки отримав повідомлення про недосяжність цієї мережі [2].

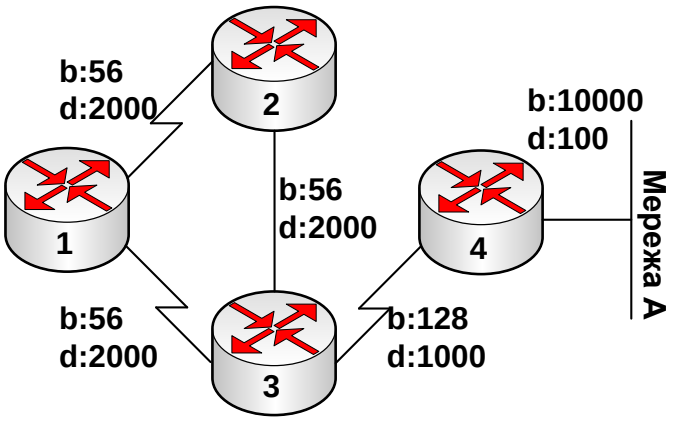


Рисунок 4.11 – Вилучення маршрута в зворотному напрямку

Нові копії ТМ зазвичай

регулярно розсилаються сусіднім маршрутизаторам. Протокол розсилає оновлення кожні 30 секунд. Однак, ***миттєві оновлення*** *(triggered updates)* розсилаються негайно у відповідь на будь-яку зміну у ТМ. Маршрутизатор, який виявив зміну в топології, негайно розсилає оновлення суміжним маршрутизаторам. Ті маршрутизатори, в свою чергу, також генерують миттєві оновлення, оповіщаючи про зміни своїх сусідів. При виході будь-якого маршруту з ладу повідомлення надсилається, не очікуючи закінчення часу таймера оновлення. Використання миттєвих оновлень в комбінації з механізмами вилучення маршрутів гарантує, що всі маршрутизатори будуть повідомлені про відмову маршрутів до закінчення часу будь-якого таймера зберігання інформації.

Миттєве оновлення, таким чином, являє собою анонс, який розсилається до закінчення часу таймера оновлення. Маршрутизатор також негайно надсилає повідомлення оновлення на всі свої інші інтерфейси, не чекаючи закінчення часу таймера. Такий принцип роботи приводить до розсилання оновленої інформації про стан маршруту та скидає таймери на сусідніх маршрутизаторах. Ця хвиля оновлень передається по всій мережі. Принцип описаної розсилки проілюстровано на рис. 4.12 [4].

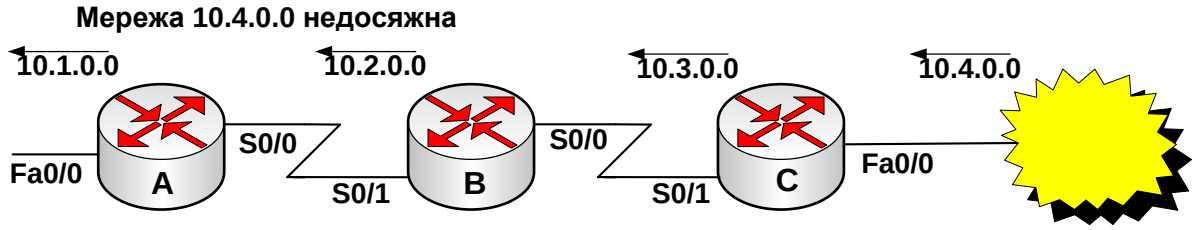


Рисунок 4.12 – Миттєві оновлення

Маршрутизатор B генерує миттєве оновлення, повідомляючи про те, що мережа 10.4.0.0 недосяжна. Після отримання цієї інформації маршрутизатор B повідомляє інші маршрутизатори про вихід з ладу мережі 10.4.0.0 через інтерфейс S0/1. В свою чергу, маршрутизатор A надсилає це повідомлення про оновлення через інтерфейс Fa0/0.

Зациклювання можна уникнути шляхом використання ***таймерів утримання інформації*** *(holddown timer)*. Послідовність дій при цьому така [1, 5, 9, 10, 16].

1. Коли маршрутизатор отримує від сусіднього пристрою оновлення маршрутів, яке вказує, що раніш доступна мережа не працює, він помічає цей маршрут як недоступний та запускає таймер.

2. Якщо до закінчення часу таймера від того ж сусіднього пристрою надходить нове повідомлення, що мережа, яка вийшла з ладу, знову доступна, то маршрутизатор помічає мережу як доступну та вимикає таймер утримання інформації.

3. Якщо нове оновлення надходить від іншого сусіднього маршрутизатора, і вказана в ньому метрика краще раніше зареєстрованої для даної

мережі, то маршрутизатор помічає мережу як доступну та вимикає таймер.

Якщо до закінчення часу таймера утримання інформації від іншого сусіднього маршрутизатора надходить нове оновлення і вказана у ньому метрика для даної мережі гірша раніш зареєстрованої – повідомлення оновлення ігнорується. В такій ситуації ігнорування повідомлень про оновлення надає більше часу для розповсюдження по всій мережі інформації про зміни топології мережі, як показано на рис. 4.13 [4].

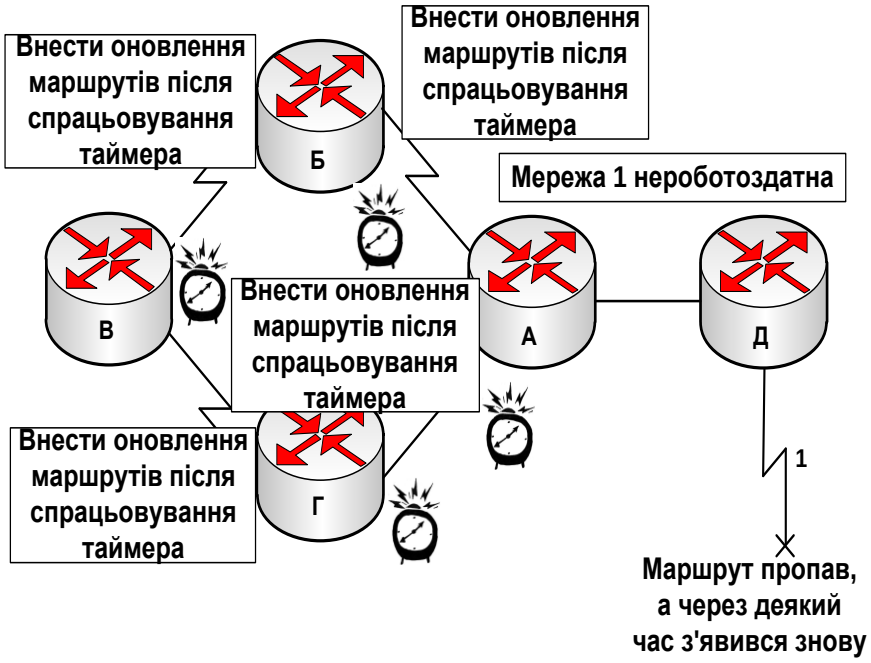


Рисунок 4.13 – Таймери утримання інформації

4.7.3 Конфігурування протоколу RIP

Спочатку слід увійти в режим конфігурування протоколу RIP за допомогою команди `router rip` (зупинити роботу протоколу RIP зі стиранням його конфігурування можна за допомогою команди `no router rip`). Далі за допомогою команди `Rt(config-router)#network ip-directly-connected-classful-net` слід вказати класові мережі, які безпосередньо під'єднані до даного маршрутизатора і повинні ним анонсуватись. Вона дозволяє надсилати та отримувати RIP-оновлення для інтерфейсів, що належать до цих мереж, а також дані мережі у RIP-повідомленнях.

Зауважимо, що у випадку, коли до маршрутизатора безпосередньо під'єднано кілька підмереж одного класу, то достатньо вказати лише одну цю класову мережу. Якщо ж вказати підмережу, то IOS автоматично конвертує її у повнокласову адресу (наприклад, якщо задати команду `network 192.168.1.64` – маршрутизатор сприйме її як `network 192.168.1.0`)

Приклади таких настроювань для маршрутизаторів R1 – R3 мережі, зображеної на рис. 4.5 наведені нижче.

```
R1(config)# router rip
R1(config-router)# network 192.168.1.0
R1(config-router)# network 192.168.4.0

R2(config)# router rip
R2(config-router)# network 192.168.2.0
```

```
R2(config-router)# network 192.168.4.0
R3(config)# router rip
R3(config-router)# network 192.168.3.0
R3(config-router)# network 192.168.4.0
```

Зауважимо, що команди `router rip` та `network` є обов'язковими для настроювання протоколу.

За замовчуванням програмне забезпечення отримує пакети RIPv1 та RIPv2, а надсилає лише пакети RIPv1. Як відомо, протокол RIPv1 не підтримує технологію VLSM, і якщо у мережі використовуються маски змінної довжини цей протокол буде працювати некоректно, отже в такому випадку слід використовувати RIPv2.

Для того, щоб сконфігурувати маршрутизатор на відправлення та отримання пакетів лише однієї версії протоколу RIP слід використовувати такі команди: `Router(config-router)# version {1 | 2}` – вказує IOS на необхідність відправлення лише пакетів версії RIPv1 або RIPv2.

`Router(config-router)# ip rip send version XX` – конфігурує інтерфейс для відправлення пакетів протоколу RIP певної версії (XX може приймати значення "1" або "2" або "1 2", в останньому випадку приймаються пакети версії 1 або 2). Конфігурування інтерфейсу для отримання пакетів протоколу RIP певної версії виконується аналогічно. Відповідна команда конфігурування має синтаксис `R1(config-router)# ip rip receive version XX`. В нашому випадку (див. рис. 4.5) слід вказати використання протоколу RIPv2.

Після виконання цих команд можна проглянути ТМ на маршрутизаторах за допомогою команди `show ip route`. Для маршрутизатора R1 ТМ має вигляд:

```
C 192.168.1.0/24 is directly connected, FastEthernet0/0
R 192.168.2.0/24 [120/1] via 192.168.4.2, 00:00:21, Serial0/0
R 192.168.3.0/24 [120/2] via 192.168.4.2, 00:00:21, Serial0/0
  192.168.4.0/30 is subnetted, 2 subnets
C   192.168.4.0 is directly connected, Serial0/0
R   192.168.4.4 [120/1] via 192.168.4.2, 00:00:21, Serial0/0
```

Для маршрутизатора R2 ТМ така:

```
R 192.168.1.0/24 [120/1] via 192.168.4.1, 00:00:05, Serial0/1
C 192.168.2.0/24 is directly connected, FastEthernet0/0
R 192.168.3.0/24 [120/1] via 192.168.4.6, 00:00:00, Serial0/0
  192.168.4.0/30 is subnetted, 2 subnets
C   192.168.4.0 is directly connected, Serial0/1
C   192.168.4.4 is directly connected, Serial0/0
```

Для маршрутизатора R3 ТМ така:

```
R 192.168.1.0/24 [120/2] via 192.168.4.5, 00:00:27, Serial0/1
R 192.168.2.0/24 [120/1] via 192.168.4.5, 00:00:27, Serial0/1
C 192.168.3.0/24 is directly connected, FastEthernet0/0
  192.168.4.0/30 is subnetted, 2 subnets
R   192.168.4.0 [120/1] via 192.168.4.5, 00:00:27, Serial0/1
C   192.168.4.4 is directly connected, Serial0/1
```

В першому стовпці ТМ є символи, які вказують на джерело отримання даного маршруту. `C` – вказує на те, що це безпосередньо під'єднана до даного маршрутизатора мережа (даний запис з'являється в ТМ в результаті настроювання певного порту маршрутизатора), а `R` – що даний маршрут отриманий від протоколу RIP.

Розглянемо складові маршрутів у ТМ, наприклад, третього рядка таблиці маршрутизації маршрутизатора R1

```
R 192.168.3.0/24 [120/2] via 192.168.4.2, 00:00:21, Serial0/0
```

Тут `192.168.3.0/24` – адреса мережі призначення з її маскою; `[120/2]`– адміністративна відстань і після слешу метрика маршруту; `192.168.4.2` – IP-адреса порту сусіднього пристрою через який отримано даний рядок; `00:00:21` – час, що минув з моменту отримання даного маршруту (пройшло 21 секунда, наступне поновлення повинно відбутись через 9 секунд); `Serial0/0` – тип та номер локального порту маршрутизатора, на який слід надіслати пакет, щоб він дістався вищевказаного пункту призначення.

Автосумаризація маршрутів

Розглянемо випадок, коли вищенаведені команди конфігурування протоколу RIP не приведуть до коректної роботи мережі. Так, якщо у складеній мережі існують підмережі, що належать одній класовій мережі, але під'єднані до різних маршрутизаторів, таблиці маршрутизації до цих підмереж будуть некоректними. Наприклад, якщо у мережі, наведеній на рис. 1 мережу 192.168.1.0/24 розбити на дві підмережі 192.168.1.0/25 і 192.168.1.128/25, перша з яких буде під'єднана до порту Fa0/0 маршрутизатора R1, а друга – Fa0/0 маршрутизатора R3 виникне така проблема. Маршрутизатори R1 і R3 будуть виконувати автоматичну сумаризацію підмереж у класову мережу, до якої входять дані підмережі (сумаризація виконується на граничному для цих підмереж маршрутизаторі, тобто маршрутизаторі, який має одну або більше підмереж, що входять до мережі певного класу і з'єднується з іншою частиною мережі через мережу, що не належить вищевказаній класовій мережі) і повідомляти маршрутизатор R2 про те, що мають безпосередній зв'язок з мережею 192.168.1.0/24. При цьому маршрутизатор R2 вважатиме, що існує два оптимальних шляхи до мережі 192.168.1.0/24. ТМ маршрутизатор R2 буде мати вигляд:

```
R 192.168.1.0/24 [120/1] via 192.168.4.1, 00:00:11, Serial0/1
                 [120/1] via 192.168.4.6, 00:00:23, Serial0/0
C 192.168.2.0/24 is directly connected, FastEthernet0/0
  192.168.4.0/30 is subnetted, 2 subnets
C   192.168.4.0 is directly connected, Serial0/1
C   192.168.4.4 is directly connected, Serial0/0
```

ТМ для R1 буде такою

```
  192.168.1.0/25 is subnetted, 1 subnets
C 192.168.1.0 is directly connected, FastEthernet0/0
R 192.168.2.0/24 [120/1] via 192.168.4.2, 00:00:25, Serial0/0
  192.168.4.0/30 is subnetted, 2 subnets
```

```
C   192.168.4.0 is directly connected, Serial0/0
R   192.168.4.4 [120/1] via 192.168.4.2, 00:00:25, Serial0/0
```

ТМ для R3 буде такою

```
    192.168.1.0/25 is subnetted, 1 subnets
C 192.168.1.128 is directly connected, FastEthernet0/0
R 192.168.2.0/24 [120/1] via 192.168.4.5, 00:00:24, Serial0/1
  192.168.4.0/30 is subnetted, 2 subnets
R   192.168.4.0 [120/1] via 192.168.4.5, 00:00:24, Serial0/1
C   192.168.4.4 is directly connected, Serial0/1
```

В результаті маршрутизація в такій мережі буде неправильною. Наприклад, якщо з маршрутизатора R2 пропінгувати будь-який вузол мережі `192.168.1.0/25` або `192.168.1.128/25` пакети, внаслідок балансування навантаження будуть циклічно (почергово) передаватись різними шляхами через інтерфейси S0/0 та S0/1. Крім того маршрутизатор R1 не містить маршруту до мережі 192.168.1.128/25, а R3 – мережі 192.168.1.0/25. Очевидно, що така ситуація недопустима.

Для вирішення даної ситуації на кожному маршрутизаторі слід відмінити автоматичну сумаризацію маршрутів за допомогою команди `no auto-summary` в режимі конфігурування протоколу RIP. Після цього ТМ маршрутизаторів набувають вигляду

для R1:

```
  192.168.1.0/25 is subnetted, 2 subnets
C 192.168.1.0 is directly connected, FastEthernet0/0
R 192.168.1.128 [120/2] via 192.168.4.2, 00:00:02, Serial0/0
R 192.168.2.0/24 [120/1] via 192.168.4.2, 00:00:02, Serial0/0
  192.168.4.0/30 is subnetted, 2 subnets
C   192.168.4.0 is directly connected, Serial0/0
R   192.168.4.4 [120/1] via 192.168.4.2, 00:00:02, Serial0/0;
```

для R2:

```
   192.168.1.0/25 is subnetted, 2 subnets
R  192.168.1.0 [120/1] via 192.168.4.1, 00:00:06, Serial0/1
R  192.168.1.128 [120/1] via 192.168.4.6, 00:00:01, Serial0/0
C  192.168.2.0/24 is directly connected, FastEthernet0/0
   192.168.4.0/30 is subnetted, 2 subnets
C    192.168.4.0 is directly connected, Serial0/1
C    192.168.4.4 is directly connected, Serial0/0;
```

для R3:

```
   192.168.1.0/25 is subnetted, 2 subnets
R  192.168.1.0 [120/1] via 192.168.4.1, 00:00:06, Serial0/1
R  192.168.1.128 [120/1] via 192.168.4.6, 00:00:01, Serial0/0
C  192.168.2.0/24 is directly connected, FastEthernet0/0
   192.168.4.0/30 is subnetted, 2 subnets
C    192.168.4.0 is directly connected, Serial0/1
C    192.168.4.4 is directly connected, Serial0/0.
```

Вимкнення маршрутних оновлень

Ще одна проблема – небажане надсилання оновлень з деяких інтерфейсів. Справа в тому, що під час застосування команди `network` протокол RIP надсилає інформацію про маршрут до вказаної в цій команді мережі зі всіх інтерфейсів у діапазоні адрес цієї мережі. Для вимкнення відправлень (але не отримувань) таких оновлень з окремих інтерфейсів можна скористатися командою `passive-interface`. Так, повертаючись до нашої мережі (див рис. 1) бачимо, що надсилати RIP-оновлення недоцільно у порти `fa0/0` маршрутизаторів R1 – R3, оскільки до відповідних мереж не під'єднані інші маршрутизатори, а лише робочі станції. Крім недоцільності такі оновлення ще будуть породжувати зайвий службовий трафік, який знижує пропускну спроможність мережі і надає можливість зловмисникам аналізувати ці оновлення. Отже, на маршрутизаторах R1 – R3 слід набрати команду `Router(config-router)# passive-interface fa0/0`.

Застосування команди ip classless

Іноді маршрутизатор отримує пакети, призначені для невідомої підмережі деякої мережі, яка входить у безпосередньо під'єднані мережі пристрою. Для пересилання цих пакетів за найкращим шляхом використовується команда глобального конфігурування `ip classless` [4]. Коли цю функцію вимкнено і пакет надсилається у підмережу мережі, до якої немає стандартного маршруту – пакет відкидається маршрутизатором.

Команда `ip classless` не впливає на ТМ, а лише на операцію пересилання пакета. Якщо маршрутизатор отримує пакет з невідомою адресою отримувача, що знаходиться в невідомій підмережі під'єднаної мережі, то передбачається, що такої підмережі неіснує. Тому маршрутизатор відкидає пакет навіть якщо існує стандартний маршрут. Виконання команди `ip classless` – вирішує цю проблему за рахунок вказання маршрутизатору заснованої на класах межі мереж в його ТМ і просто вибирати стандартний маршрут [1].

Нагадаємо, що під час отримання інформації про мережу RIP-маршрутизатори покладаються на сусідні маршрутизатори. Протоколам RIP, як будь-якому ДВП властиві проблеми, що спричинюють повільну конвергенцію. Для уникнення петель маршрутизації і зациклювання пакетів протокол RIP використовує: розщеплення горизонту (Split horizon); вилучення маршрутів у зворотному напрямку (Poison reverse); таймери утримання інформації (Holddown counters); миттєві (тригерні) оновлення (Triggered updates). Деякі з цих методів потребують додаткового конфігурування, деякі – ні. В деяких випадках потрібно вимкнути механізм розщеплення горизонту [4]. Таке вимкнення виконується за допомогою комади `Router(config-if)#no ip split-horizon`.

Встановлення значень таймерів

Ще один механізм, який може потребувати змін – це застосування таймера утримання інформації. Такий таймер дозволяє попередити зациклювання пакетів, проте збільшує час конвергенції мережі. Стандартно Holddown counters складає 180 секунд. Протягом цього часу не дозволяється оновлення внутрішніх маршрутів, однак і дійсні альтернативні маршрути також не будуть встановлюватись. Для прискорення конвергенції час Holddown counters може бути зменшено. В ідеальному випадку – це встановлення цього періоду утримання трошки більшим за максимальний час оновлення маршрутів у даній об'єднаній мережі [4].

Для змінення періоду таймера утримання інформації використовується команда `Rt(config-router)#timers basic update invalid holddown flush [sleeptime]`, де `update` – таймер оновлення (стандартно 30 секунд) – задає період розсилання оновлень про маршрути; `invalid` – таймер дійсності маршруту (стандартно 180 секунд) – задає час, протягом якого маршрутизатор при відсутності анонсів про оновлення деякого маршруту чекає, перед тим, як об'явити цей маршрут недійсним. Маршрут зберігатиметься у ТМ поки не спливе час таймера скидання маршрутів `flush`; `holddown` – таймер утримання (стандартно 180 секунд) – задає час, протягом якого нові повідомлення про оновлення маршрутизації ігноруються; `flush` – таймер скидання маршрутів – задає час, який проходить до того, як маршрут буде вилучений з ТМ (стандартно 240 секунд).

Додатковим параметром, що впливає на час конвергенції і підлягає конфігуруванню є інтервал розсилання повідомлень оновлень маршрутів (за замовчуванням кожні 30 секунд). Цей час можна збільшити (для економії смуги пропускання) або зменшити (для скорочення часу конвергенції) за допомогою команди

`Router(config-router)#update-timer seconds`.

Встановлення кількості паралельних маршрутів

Під час конфігурування протоколу RIP можна встановити кількість паралельних маршрутів. Стандартно для більшості протоколів динамічної маршрутизації в ТМ встановлюється до чотирьох таких маршрутів (для статичних маршрутів їх може бути шість). Для змінення стандартної кількості паралельних маршрутів можна використати команду

`Router(config-router)#maximum-paths [`*number*`]`.

Анонсування статичних та стандартних маршрутів

Статичні маршрути, вказані на деякому інтерфейсі за замовчуванням не анонсуються протоколом RIP. Якщо статичний маршрут призначений інтерфейсу, який не вказаний у команді `network` – то жоден протокол маршрутизації не анонсує такий маршрут. Дозволити анонсування можна за допомогою команди `redistribute static` [4].

Стандартні маршрути (маршрути за замовчуванням), протоколом RIP

за замовчуванням також не анонсуються. Якщо треба, щоб такі маршрути включалися до RIP-анонсів, на маршрутизаторі, де вказано маршрут за замовчуванням, слід виконати команду `default-information originate` [4].

4.7.4 Тестування та усунення помилок у роботі протоколу RIP

Є багато команд тестування протоколу RIP [1, 4, 11]. Розглянемо деякі з них. Команда `show ip protocols` – відображує інформацію про усі протоколи IP-маршрутизації, які сконфігуровані на маршрутизаторі. Перш за все слід перевірити: чи увімкнено на потрібних інтерфейсах протокол RIP; чи приймають і пересилають оновлення маршрутизації протоколу RIP відповідні інтерфейси; чи правильна версія оновлень маршрутизації протоколу RIP використовується; чи анонсує маршрутизатор потрібні мережі.

Команда `show ip route` – показує ТМ і дозволяє побачити чи заносяться туди маршрути отримані протоколом RIP (позначаються символом „R"). Такі записи у ТМ з'являються із затримкою внаслідок конвергенції.

Команда `show interface interface` – відображає інформацію про конкретний інтерфейс (зокрема його активність і тип протоколу). Команда `show ip interface interface` – відображає інформацію про протокол IP, що стосується конкретного інтерфейсу. Команда `show running-config` – дозволяє перевірити налаштування протоколу RIP, аналізуючи робочий файл конфігурації. Команда `show ip rip database` – відображає приватну БД протоколу RIP.

Зауважимо, що більшість помилок в роботі протоколу RIP пов'язані з виконанням некоректних команд `network`, з розривами у мережах або з розщепленням горизонту. Важливим інструментом для знаходження помилок, пов'язаних з поновленнями протоколу RIP є команда `debug ip rip`, під час її виконання відображаються поновлення маршрутизації RIP [4].

4.8 Удосконалений протокол маршрутизації EIGRP

4.8.1 Огляд протоколу EIGRP

Дистанційно-векторний протокол маршрутизації EIGRP (Enhanced Interior Gateway Routing Protocol) був розроблений та реалізований фірмою Cisco у 1992 р. Він є суттєвим вдосконаленням свого попередника – протоколу маршрутизації IGRP [5], який сьогодні фактично не використовується. Тому в даному посібнику протокол IGRP не розглядається, і основна увага приділяється протоколу EIGRP.

Переваги використання протоколу EIGRP

Перевагами протоколу EIGRP відносно простих дистанційно-векторних протоколів є [5, 15]:

- *швидка конвергенція*. На маршрутизаторах протоколу EIGRP конвергенція відбувається значно швидше, оскільки вона базується на сучасному алгоритмі дифузії поновлень маршрутизації DUAL (Diffusing Update

Algorithm) [4, 15, 16]. Цей алгоритм гарантує відсутність петель у кожний момент часу на всьому маршруті та дозволяє усім маршрутизаторам, що належать до даної топології, виконати одночасну синхронізацію. Крім того, якщо у традиційних дистанційно-векторних протоколів певний маршрут став недоступним – маршрутизатори повинні чекати чергового періодичного поновлення, а протокол EIGRP буде при цьому використовувати резервний шлях (якщо такий існує);

- *ефективне використання смуги пропускання*. По-перше, протокол EIGRP використовує розсилання часткових, обмежених за обсягом поновлень (Partial, bounded updates) маршрутизації, і як наслідок цього забезпечується мінімальне використання такими поновленнями смуги пропускання в умовах стабільної роботи мережі. Маршрутизатори EIGRP, як правило, розсилають часткові, поетапні поновлення маршрутизації, а не повні таблиці маршрутизації. Цей процес аналогічний роботі протоколу OSPF, однак на відміну від нього, маршрутизатори протоколу EIGRP розсилають ці часткові поновлення не всім маршрутизаторам даної області, лише тим, яким вони дійсно потрібні. Саме тому такі поновлення називаються обмеженими. По-друге, у протоколі EIGRP замість регулярного розсилання поновлень маршрутизації маршрутизатори підтримують постійний контакт один з одним шляхом розсилання невеликих пакетів вітання. Хоча пакети вітання розсилаються регулярно, внаслідок невеликого розміру вони досить незначно використовують смугу пропускання (на відміну від протоколів RIP та IGRP, які розсилають сусіднім пристроям свою повну таблицю маршрутизації кожні 30 або 90 секунд, відповідно);
- *підтримка масок підмереж змінної довжини VLSM* (Variable-Length Subnet Mask) *і безкласової міждоменної маршрутизації CIDR* (Classless Interdomain Routing). На відміну від протоколу IGRP, EIGRP забезпечує повну підтримку безкласового IP шляхом обміну масками підмереж у повідомленнях поновлення маршрутів. Це дозволяє мережевим проектувальникам максимально використовувати адресний простір;
- *підтримка декількох протоколів мережевого рівня*. Протокол EIGRP підтримує протоколи IP, IPX та AppleTalk шляхом використання залежних від протоколу модулів (protocol-dependent module, PDM);
- *використання складної та гнучкої метрики маршрутів*. Метрика протоколу EIGRP, на відміну від багатьох інших протоколів маршрутизації (крім протоколу IGRP), може враховувати одразу чотири показники (пропускна спроможність, час затримки, завантаженість та надійність каналу). При цьому адміністратор може задавати значимість кожного з цих показників.

Доцільно зауважити, що у деяких джерелах EIGRP називають гібридним протоколом маршрутизації, який поєднує кращі риси дистанційно-векторних алгоритмів і алгоритмів маршрутизації за станом каналу. Так, наприклад, протокол EIGRP використовує такі функції протоколу OSPF, як

часткові поновлення маршрутів та виявлення сусідніх пристроїв. Але слід пам'ятати, що у технічному аспекті протокол EIGRP є суто ДВП.

4.8.2 Обчислення метрики протоколу EIGRP

Протокол EIGRP використовує метрику довжиною 32 біта, яка обчислюється за формулою [5, 15]:

$$M_{EIGRP} = \left(K_1 \cdot \left\lfloor \frac{10^7}{B_w} \right\rfloor \cdot 256 + \frac{K_2 \cdot \left\lfloor \frac{10^7}{B_w} \right\rfloor \cdot 256}{256 - L_d} + K_3 \cdot \frac{D_l}{10} \cdot 256 \right) \cdot \frac{K_5}{R_l + K_4}, \quad (4.1)$$

де B_w – найменша смуга пропускання каналу на шляху між відправником та отримувачем у КБіт/с; D_l – сумарна затримка каналів передавання даних між відправником та отримувачем в мкс. Затримка визначається типом ЛЗ з'єднання (значення затримок для різних типів ліній зв'язку наведено у таблиці 4.7); L_d – максимальна завантаженість каналу між відправником та отримувачем; R_l – найнижча надійність каналу маршруту між відправником та отримувачем (характеризує як часто у каналі виникають помилки передавання даних); K_1 – K_5 – вагові коефіцієнти. Позначення $\lfloor X \rfloor$ в даному випадку означає – ціла частина від числа X.

Таблиця 4.7 – Значення затримки залежно від середовища передавання

Середовище передавання	Значення затримки (мкс)	Середовище передавання	Значення затримки (мкс)
Fast Ethernet	100	1544 K	20000
FDDI	100	1024 K	20000
100M ATM	100	512 K	20000
Ethernet	10000	64 K	20000

Значення B_w та D_l – це статичні величини, а L_d та R_l – вимірюються динамічно протягом 5 хвилин (для визначення відповідних середніх значень і уникнення впливу, наприклад, миттєвих заторів та помилок каналу).

Значення надійності може бути у діапазоні від 1 до 255, де 1 – відповідає мінімальній надійності, а 255 – максимальній. Надійність виражається у вигляді дробу $R_l/255$. Так, 255/255 – означає надійність 100 %, а 250/255 – 98 %.

Значення завантаженості також може бути у діапазоні від 1 до 255, де 1 – відповідає мінімальній завантаженості, а 255 – максимальній. Як і надійність, завантаженість також виражається у вигляді дробі $L_d/255$. Так, наприклад, 51/255 – означає 20 % завантаженість, а 255/255 – що дана лінія повністю завантажена.

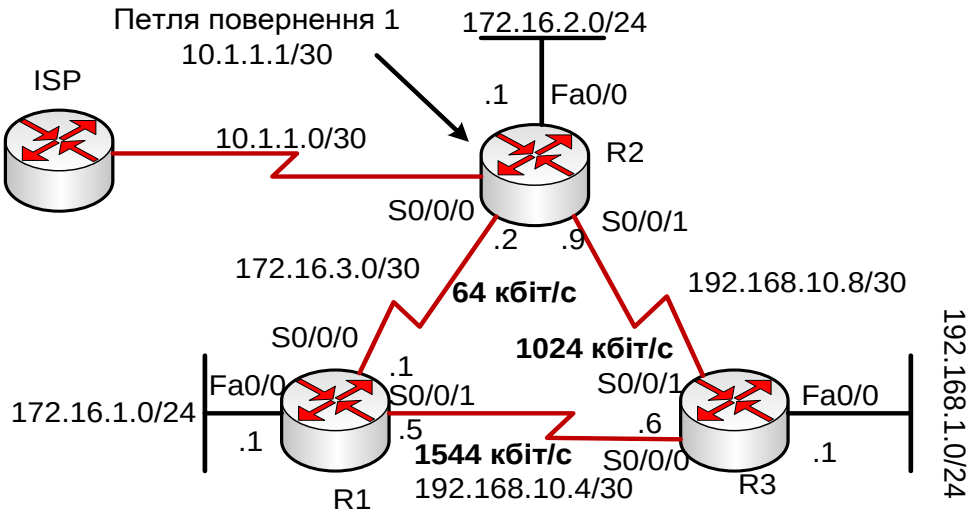
Зауважимо, що за замовчуванням значення коефіцієнтів такі: $K_1 = K_3 = 1$, $K_2 = K_4 = K_5 = 0$ і формула для обчислення метрики має вигляд:

$$M_{EIGRP}^{def} = \left(K_1 \cdot \left\lfloor \frac{10^7}{B_w} \right\rfloor + K_3 \cdot \frac{D_l}{10} \right) \cdot 256. \qquad (4.2)$$

Оскільки до складу метрики в даному випадку входять лише статичні величини, не буде виконуватись частих перерахунків даних топологічної таблиці.

Зазначимо, що максимальна кількість переходів для протоколу EIGRP дорівнює 224 (наприклад, для протоколу RIP кількість переходів становить всього 16), чого цілком достатньо для підтримки навіть найбільших сучасних мереж [5].

Розглянемо приклад обчислення метрики. Подивимось на приклад невеликої КМ (рис. 4.14.а) та зміст ТМ маршрутизатора R2 (рис. 4.14.б). ТМ містить маршрути до всіх відомих для R2 пунктів призначення. Літери „C” і „D” в лівих позиціях рядків таблиці означає джерело отримання даного рядка. Так, літера „C” означає безпосередньо під’єднані мережі, а літера



а)

```
R2# show ip route
<частина виведення пропущена>
     192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
D 192.168.10.0/24 is a summary, 00:00:9, Null0
D 192.168.10.4/30 [90/21024000] via 192.168.10.10, 00:00:9, Serial0/0/1
C 192.168.10.8/30 is directly connected, Serial0/0/1
     172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
D 172.16.0.0/16 is a summary, 00:00:9, Null0
D 172.16.1.0/24 [90/40514560] via 172.16.3.1, 00:00:9, Serial0/0/0
C 172.16.2.0/24 is directly connected, Fastethernet0/0
C 172.16.3.0/30 is directly connected, Serial0/0/0
     10.0.0.0/30 is subnetted, 1 subnets
C 10.1.1.0/30 is directly connected, loopback1
D 192.168.1.0/24 [90/3014400] via 192.168.10.10, 00:00:9, Serial0/0/1
```

б)

Рисунок 4.14 – Приклад обчислення метрики
а) невелика комп’ютерна мережа; б) ТМ маршрутизатора R2

„D” – що даний рядок отриманий за допомогою протоколу EIGRP. Розглянемо останній рядок ТМ:

```
D  192.168.1.0/24  [90/3014400]  via  192.168.10.10  00:00:09,
Serial0/0/1.
```

Тут після літери „D” йдуть: IP-адреса пункту призначення; у квадратних дужках адміністративна відстань (90) і через слеш метрика маршруту (3014400); IP-адреса інтерфейсу наступного вузла на шляху до пункту призначення (192.168.10.10); скільки часу існує цей рядок (9 сек.) та локальний вихідний інтерфейс, через який можна досягнути пункт призначення (Serial0/0/1). Обчислимо значення метрики.

Для визначення метрики згідно з (4.2) слід визначити найменшу смугу пропускання каналу уздовж маршруту від джерела до пункту призначення та знайти сумарну затримку. Найменша смуга пропускання 1024 Кбіт/с (оскільки оптимальний маршрут від R2 до мережі 192.168.1.0 проходить через R3, а смуга пропускання каналу від R2 до R1 складає всього 64 Кбіт/с). Сумарна затримка шляху складає 20000 + 100 = 20100 мкс (див. табл. 4.7). Отже, враховуючи сказане, остаточне шукане значення складе:

$$M_{EIGRP}^{def} = \left(1 \cdot \left\lfloor \frac{10^7}{1024} \right\rfloor + 1 \cdot \frac{20100}{10}\right) \cdot 256 = \left(9765 + 2010\right) \cdot 256 = 3014400.$$

Далі зупинемось детальніше на сутності роботи протоколу EIGRP, але для цього перш за все слід познайомитись з його основною термінологією.

4.8.3 Термінологія протоколу EIGRP

Протокол EIGRP у своїй роботі використовує дані трьох таблиць: *маршрутизації*, *сусідніх пристроїв* та *топології*. Ці таблиці ще називають базами даних протоколу. Призначення ТМ нам вже відоме. Тому розглянемо призначення двох інших таблиць [5, 15].

Таблиця сусідніх пристроїв

Кожний маршрутизатор EIGRP підтримує таблицю сусідніх пристроїв (neighbor table), в якій перераховані суміжні маршрутизатори. Для кожного протоколу (наприклад, IP, IPX), що підтримується протоколом EIGRP, є своя таблиця сусідніх пристроїв (ТСП). При виявленні нових сусідніх пристроїв їх адреси та інтерфейси заносяться у ці таблиці. Проглянути зміст ТСП можна за командою show ip eigrp neighbors).

При відправленні пакета привітання сусідній пристрій повідомляє час утримання, що вказує, як довго маршрутизатор розглядає свій сусідній пристрій як досяжний та роботоздатний. Якщо за період утримання від маршрутизатора не надійшов пакет привітання, то вважається, що час утримання вичерпано. В такому випадку алгоритм DUAL (цей алгоритм ми розглянемо пізніше) інформується про зміну топології і повинен знову обчислити параметри нової топології.

ТСП має, зокрема, такі поля.

Порядковий номер (H) запису в міру навчання даного пристрою стосовно сусідніх пристроїв.

Адреса сусіднього пристрою (Neighbor Address) – адреса мережевого рівня сусіднього пристрою.

Інтерфейс (Interface) – локальний інтерфейс, через який було отимано пакет Hello від сусіднього пристрою.

Час утримання (Hold Time) – часовий інтервал, після закінчення якого, у випадку відсутності будь-яких повідомлень від сусіднього пристрою, канал розглядається як нероботоздатний. При отриманні ж будь-якого пакета протоколу EIGRP, таймер приймає початкове значення.

Доступний час (Uptime) – час, що минув з моменту додання даного сусіднього пристрою у ТСП.

Таймер циклу обміну повідомленнями (Smooth Round-Trip Timer – SRTT) – середній час, потрібний для того, щоб надіслати пакет сусідньому пристрою та одержати від нього відповідний пакет. Цей таймер визначає інтервал повторного передавання (Retransmit Interval – RTI).

Час ретрансляції (Retransmission Timeout – RTO) – час в мілісекундах, протягом якого програмне забезпечення очікує моменту повторного пересилання пакета з черги повторного розсилання.

Лічильник черги (Queue Count – Q Cnt) – число пакетів, які перебувають у черзі очікуючи передавання. Якщо це значення постійно більше нуля – ймовірно маршрутизатор зазнає переповнення. Нульове значення свідчить, що пакетів протоколу EIGRP у черзі немає.

Послідовний номер (Sequence Number – Seq No) – номер останнього пакета, отриманого від даного сусіднього пристрою. Протокол EIGRP використовує це поле для підтвердження отримання пакета, переданого сусіднім пристроєм, та для ідентифікації пакетів, що передані з порушенням порядку. ТСП забезпечує надійне та впорядковане доставлення пакетів.

Наприклад, ТСП для маршрутизатора R2 (рис. 4.14.а) має вигляд

```
R2#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H Address         Interface  Hold  Uptime   SRTT  RTO   Q   Seq
                             (sec)          (ms)       Cnt  Num
1 192.168.10.10 Ser0/0/1     10  00:01:44  20   200   0    7
0 172.16.3.1    Ser0/0/0     10  00:03:27  25   200   0    12
```

Топологічна таблиця

Топологічна таблиця (topology table) містить всі ТМ протоколу EIGRP, наявні на пристроях даної автономної системи (проглянути зміст топологічної таблиці можна за командою show ip eigrp topology). Алгоритм DUAL отримує інформацію з ТСП і топологічної таблиці (ТТ) та обчислює маршрути з найменшою оцінкою до кожного пункту призначення. Завдяки цьому маршрутизатори протоколу EIGRP можуть швидко визначити альтерна-

тивні маршрути та використати їх у разі потреби. Первинний маршрут (successor) записується у ТМ, а його копія – у ТТ. Усі маршрутизатори EIGRP підтримують ТТ для кожного сконфігурованого мережевого протоколу. У цій таблиці містяться маршрути до усіх пунктів призначення, які стали відомі маршрутизатору.

ТТ має такі поля [5].

Передбачувана відстань (Feasible Distance – FD) – це найменша обчислена метрика до кожного пункту призначення. У прикладі 4.1 показано ТТ для маршрутизатора R2, з прикладу, наведеного на рис. 4.14. Тут передбачувана відстань, наприклад, до мережі 192.168.1.0 становить 3014400, на що вказує запис „FD is 3014400”.

```
R2# show  ip  eigrp  topology
IP-EIGRP Topology Table  AS(1)/ID 10.1.1.1
Codes: P - Passive, A - Active, U - Update,
Q - Query, R - Reply, r - Reply Status s – sia Status
P 192.168.10.4/30, 1 successors, FD is 3523840
   via 192.168.10.10 (3523840/2169856), Serial0/0/1
   via 172.16.3.1 (410240000/2169856), Serial0/0/0
P 192.168.1.0/24, 1 successors, FD is 3014400
   via 192.168.10.10 (3014400/28160), Serial0/0/1
   via 172.16.3.1 (410240000/2172416), Serial0/0/0
P 192.168.10.8/30, 1 successors, FD is 3011840
   via connected Serial0/0/1
P 172.16.1.0/24, 1 successors, FD is 3526400
   via 192.168.10.10 (3526400/2172416), Serial0/0/1
   via 172.16.3.1 (40514560/28160), Serial0/0/0
P 172.16.2.0/24, 1 successors, FD is 28160
   via connected FastEthernet 0/0
P 172.16.3.0/30, 1 successors, FD is 40512000
```

Приклад 4.1 – Топологічна таблиця протоколу EIGRP

Джерело маршруту (Route Source) – це ідентифікаційний номер маршрутизатора, який анонсував цей маршрут. Дане поле заповнюється лише тільки для маршрутів, які стали відомі ззовні від інших мереж протоколу EIGRP. У прикладі 4.1, джерелами маршруту до мережі 192.168.1.0 є 192.168.10.10 та 172.16.3.1, про що свідчать записи „via 192.168.10.10” та „via 172.16.3.1”, відповідно.

Повідомлена відстань (Reported Distance – RD) або об’явлена відстань (Advertised Distance – AD) – це значення відстані, яке сусідній маршрутизатор повідомляє конкретному одержувачу. У прикладі 4.1 повідомлена відстань до мережі 192.168.1.0 дорівнює 28160, на що вказує значення поля RD (3014400/28160).

Інформація про інтерфейс (Interface Information) – це номер інтерфейсу, через який можна досягти пункту призначення. З прикладу 4.1 видно, що мережу 192.168.10.10 можна досягнути через інтерфейс Serial0/0/1 (via

192.168.10.10 (3014400/28160), Serial0/0/1), а можна резервним шляхом через Serial0/0/0 (via 172.16.3.1 (410240000/2172416), Serial0/0/0)

Статус маршруту (Route Status) – може бути пасивний або активний. Пасивні (Passive – P) – це стійкі та готові до використання маршрути, активні (Active – A) це ті, для яких алгоритм DUAL продовжує процес перерахування маршруту. Протокол EIGRP сортує записи ТТ так, щоб первинні маршрути знаходились у її верхній частині, а за ними йшли резервні. У нижній частині цієї таблиці розташовуються маршрути, які алгоритм DUAL розглядає як можливі петлі маршрутизації.

Первинні маршрути

Первинним називається маршрут обраний як основний для досягнення певного пункту призначення. Цей маршрут визначається алгоритмом DUAL на основі інформації з ТСП і ТТ, і вноситься до ТМ. Для кожного конкретного маршруту може бути до чотирьох первинних маршрутів. Вони можуть мати як однакові, так і неоднакові оцінки й розглядаються як найкращі вільні від петель маршрути до даного пункту призначення.

Резервні маршрути

Потенційно первинний (Feasible Successor – FS) – це резервний маршрут. Такі маршрути встановлюються одночасно з первинними, однак, зберігаються тільки у ТТ. Одночасно можуть зберігатися кілька резервних маршрутів. Наявність резервного маршруту для досягнення одержувача не є обов’язковою.

Маршрутизатор розглядає пристрої на резервному маршруті як сусідні в спадному напрямку (він вважає, що вони перебувають ближче до пункту призначення, ніж він сам). Вони виражають анонсовану сусіднім маршрутизатором оцінку маршруту до пункту призначення. Якщо первинний маршрут стає недійсним, то маршрутизатор шукає резервний і підвищує його статус до первинного. Резервний маршрут до пункту призначення повинен мати менше значення FD, ніж значення RD даного первинного маршруту.

Якщо резервний маршрут не був установлений на основі наявної інформації, то маршрутизатор надає йому статус активного (Active) і надсилає пакети запитів усім сусіднім пристроям для перерахування топології. Після одержання відповідей на ці запити маршрутизатор може на їх основі установити нові первинні або резервні маршрути. Після цього маршрутизатор надає маршруту статус пасивного (Passive).

Вибір первинного та резервних маршрутів

Розглянемо питання визначення маршрутизатором первинних та резервних маршрутів. Нехай в ТМ маршрутизатора RTA є маршрут до мережі Network Z через маршрутизатор RTB (рис. 4.15). З точки зору маршрутизатора RTA маршрутизатор RTB перебуває на поточному первинному марш-

руті до мережі Network Z, тому RTA надсилає пакети, призначені для мережі Network Z у напрямку RTB. Маршрутизатор RTA повинен мати принаймні один первинний маршрут до Network Z для того, щоб алгоритм DUAL міг внести його в ТМ [5, 15].

Якщо деякий маршрутизатор RTC, який з'єднаний з RTA аналогічно маршрутизатору RTB і повідомляє RTA про наявність у нього маршруту до мережі Network Z з такою ж метрикою, як і у маршрутизатора RTB, то RTA також розглядає RTC як первинний маршрут і алгоритм DUAL установлює другий маршрут до мережі Network Z через RTC (рис. 4.16).

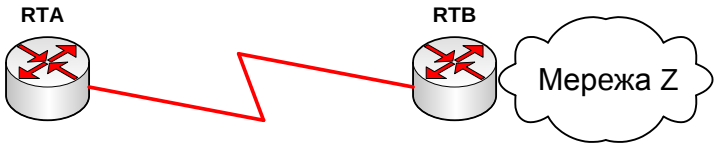


Рисунок 4.15 – Первинний маршрут протоколу EIGRP

Кожен сусідній пристрій маршрутизатора RTA, що анонсує вільний від петель маршрут до мережі Network Z (однак з FD більшою, ніж метрика найкращого маршруту й меншою, ніж його RD), ідентифікується у ТТ, як той, що знаходиться на резервному маршруті. Маршрутизатор розглядає свої пристрої на резервних маршрутах як сусідні пристрої, що перебувають у низхідному напрямку, тобто розташовані ближче до одержувача, ніж він сам. Якщо з будь-яких причин первинний маршрут не може виконувати свої функції, то алгоритм DUAL може швидко знайти резервний на основі даних ТТ й встановити новий маршрут до пункту призначення. Якщо ж такий резервний маршрут відсутній, то алгоритм DUAL переводить маршрут в активний стан і запитує допомоги у сусідів у знаходженні нового, вільного від петель маршруту. Сусідні маршрутизатори зобов'язані відповісти на цей запит. Якщо в сусіднього маршрутизатора є такий маршрут (маршрути), то надсилається інформація про нього (них). Інакше сусідній маршрутизатор повідомляє про відсутність маршруту до цього пункту призначення.

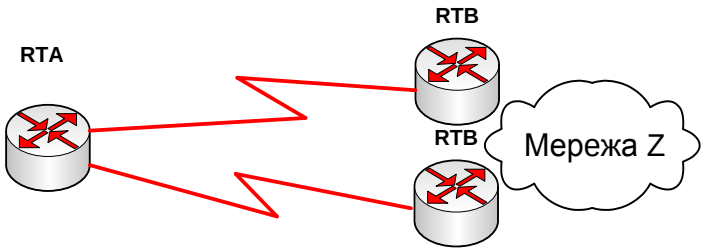


Рисунок 4.16 – Первинні маршрути протоколу EIGRP

Надлишкова кількість перерахувань маршрутів свідчить про нестабільну роботу мережі й знижує її продуктивність. Для запобігання проблем, пов'язаних з конвергенцією, алгоритм DUAL перед виконанням перерахування завжди намагається знайти резервний маршрут. Якщо такий є, то алгоритм DUAL може встановити новий маршрут без перерахування.

Застрявання активних маршрутів

Якщо один або більше маршрутизаторів, яким був розісланий запит, не відповідає протягом активного часу (180 секунд), то маршрут (або кілька маршрутів) переводиться у стан „застрявання" (stuck in active). В цьому випадку протокол EIGRP виключає зі своєї ТСП маршрутизатори, що не відповіли на запит і реєструє в системному журналі повідомлення про помилку „stuck in active" для маршрутів, які були активними.

Створення тегів для маршрутів

У ТТ може бути записана додаткова інформація про кожний маршрут. Протокол EIGRP класифікує маршрути на внутрішні і зовнішні. Внутрішніми називаються маршрути усередині даної автономної системи протоколу EIGRP, а зовнішніми – ті, що беруть свій початок поза даною автономною системою. Маршрути, отримані або перерозподілені від інших протоколів маршрутизації вважаються зовнішніми [5].

Як тег, маршруту може бути присвоєне значення від 0 до 255. Всі зовнішні маршрути заносяться в ТТ і їм призначається тег, що містить таку інформацію: ідентифікаційний номер маршрутизатора EIGRP, що поширив маршрут у мережу EIGRP; Номер АС одержувача; протокол, використовуваний у зовнішній мережі; оцінка або метрика, отримана від зовнішнього протоколу; конфігурований тег адміністратора [5].

Для задання строгої і точної політики маршрутизації рекомендується скористатися функцією задання маршрутум тегів і, особливо, тегів адміністратора. Останнім може бути будь-яке число від 0 до 255. По суті це звичайний тег, що можна використовувати для реалізації спеціальної стратегії маршрутизації. Зовнішні маршрути можуть прийматися, відкидатися або поширюватися на основі кожного з тегів маршруту, в тому числі й тегу адміністратора. Оскільки користувач може задати тег адміністратора будь-яким зручним для нього способом, функція задання тегів маршрутум надає більшу гнучкість під час керуванні мережею. Це виявляється особливо корисним у тих випадках, коли мережа протоколу EIGRP взаємодіє з мережею протоколу граничного шлюзу, що базується на використанні політик.

4.8.4 Функції і технології протоколу EIGRP

Протокол EIGRP використовує багато нових технологій, кожен з яких поліпшує операційну ефективність, підвищує швидкість конвергенції та розширює набір функцій протоколу IGRP та інших протоколів маршрутизації. Ці технології можна поділити на такі чотири категорії [5].

Виявлення сусідніх пристроїв і відновлення загубленого з ними зв'язку.

Надійний транспортний протокол (Reliable Transport Protocol).

Алгоритм DUAL кінцевих станів машини.

Модулі конкретних протоколів.

Розглянемо детальніше кожну з цих технологій.

Виявлення сусідніх пристроїв і відновлення втраченого з ними зв'язку

Звичайні прості дистанційно-векторні маршрутизатори не встановлюють зв'язків зі своїми сусідами. На відміну від них маршрутизатори протоколу EIGRP встановлюють зв'язки зі своїми сусідніми пристроями. На рис. 4.17 проілюстровано процес встановлення зв'язків між суміжними пристроями протоколу EIGRP [5].

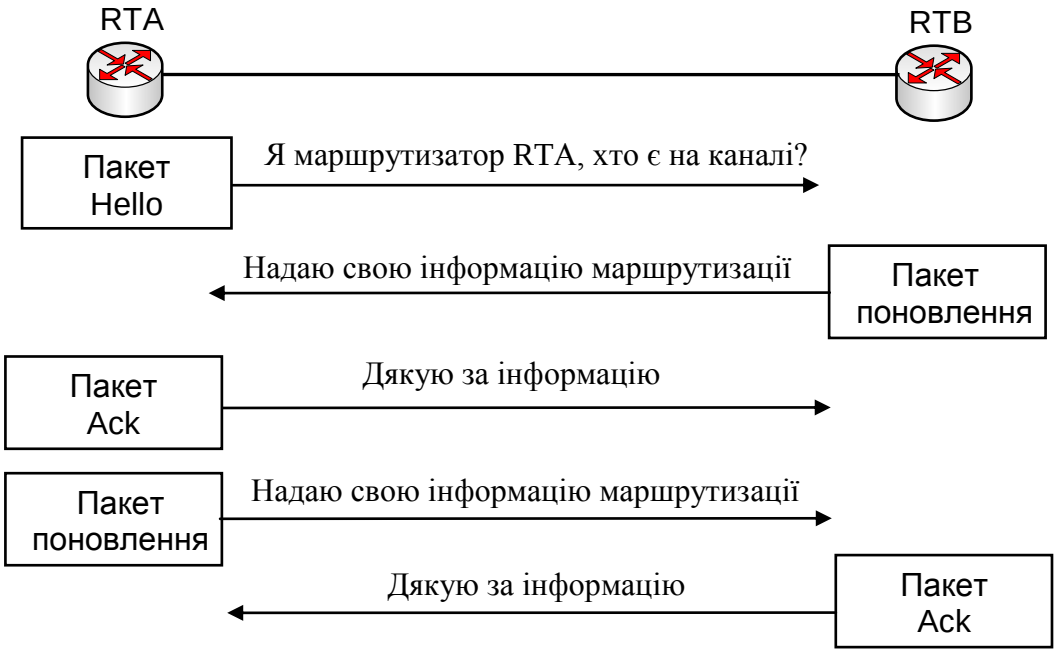


Рисунок 4.17 – Обмін інформацією сусідніх маршрутизаторів EIGRP

Маршрутизатори EIGRP встановлюють відношення суміжності із сусідніми маршрутизаторами шляхом розсилання невеликих пакетів-вітань. Ці пакети за замовчуванням розсилаються кожні 5 секунд на каналах з великою смугою пропускання й кожні 60 секунд на низькошвидкісних багаточочкових каналах. Маршрутизатор EIGRP припускає, що поки від відомих йому сусідніх пристроїв надходять пакети вітання, ці пристрої та відповідні маршрути залишаються діючими.

Формуючи відношення суміжності маршрутизатори EIGRP одержують можливості: динамічно дізнаватися про нові маршрути, що з'являються у мережі; ідентифікувати маршрутизатори, які стали недосяжними або нероботоздатними; виявляти маршрутизатори, які раніше були недосяжні.

Надійний транспортний протокол

Надійний транспортний протокол (Reliable Transport Protocol, RTP) – це протокол транспортного рівня, який може гарантувати впорядковане доставлення пакетів EIGRP всім сусідам. У мережах IP-протоколу для впорядкування і своєчасного доставлення пакетів використовується протокол

TCP. Однак протокол EIGRP незалежний від використовуваного мережевого протоколу і не використовує протокол TCP/IP для обміну інформацією маршрутизації (як це роблять протоколи RIP, IGRP, OSPF). Для реалізації такої незалежності від IP, протокол EIGRP використовує свій фірмовий транспортний протокол для гарантованого доставлення інформації.

EIGRP може активізувати протокол RTP для забезпечення служби надійної або негарантованої доставки залежно від конкретної ситуації. Наприклад, пакети вітання не потребують додаткового навантаження на мережу за рахунок гарантованого доставлення, оскільки вони розсилаються часто і їх розмір повинен бути невеликим. Проте гарантоване доставлення інформації про маршрути може прискорити конвергенцію, оскільки маршрутизатори EIGRP не очікують завершення часу таймера до повторного передавання. Використання надійного транспортного протоколу дозволяє протоколу EIGRP одночасно здійснювати багатоадресне та одноадресне розсилання, що забезпечує максимальну ефективність.

Машина кінцевих станів алгоритму DUAL

Головним компонентом протоколу EIGRP є алгоритм обчислення маршрутів. Повна назва цієї технології – абстрактна машина кінцевих станів (finite-state machine, FSM) алгоритму DUAL. Вона визначає набір можливих станів, через які можна пройти, які події викликають ці стани, а які є результатом цих станів. FSM містить всі логічні операції, необхідні для обчислення й порівняння маршрутів у мережі протоколу EIGRP [5, 15].

Алгоритм DUAL стежить за всіма маршрутуми, анонсованими сусідніми пристроями й використовує складену (композитну) метрику для кожного маршруту. Він гарантує, що кожний маршрут не містить петель. Після відповідних обчислень алгоритм DUAL заносить маршрути з найменшими оцінками в ТМ (тобто первинні маршрути), а їх копії – у ТТ.

Протокол зберігає важливу маршрутну й топологічну інформацію в ТСП і ТТ, які надають алгоритму DUAL маршрутну інформацію у випадку порушень у роботі мережі. Використовуючи інформацію цих таблиць DUAL може при необхідності швидко знаходити альтернативні маршрути: якщо будь-який канал стає непрацездатним, то він шукає у ТТ альтернативний (потенційно первинний або резервний) маршрут.

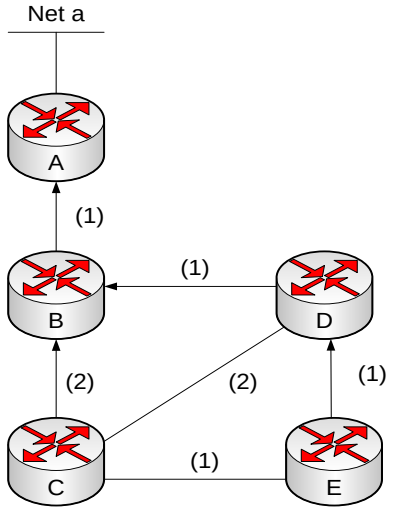
Пакети, надіслані у мережу-одержувач, негайно надсилаються за резервним маршрутом, що у цей момент одержує статус первинного, як показано на рис. 4.18. Тут маршрутизатор D втрачає прямий зв’язок з маршрутизатором B і не має ідентифікованого первинного маршруту. Ймовірна відстань FD (обчислена оцінка) для маршруту від маршрутизатора D до маршрутизатора A дорівнює 2, а анонсована відстань через маршрутизатор C дорівнює 3. Оскільки значення RD менше, ніж метрика найкращого маршруту, але більше, ніж відстань FD, жоден резервний маршрут не заноситься у ТТ. Маршрутизатор C має ідентифікований резервний маршрут,

так само як і маршрутизатор E, оскільки маршрут вільний від петель, а відстань RD до маршрутизатора наступного переходу менша, ніж відстань FD для первинного маршруту. Кінцевий результат роботи алгоритму DUAL наведено на рис. 4.18,б. Детально процес конвергенції наведений у [15].

Модулі PDM

Однією з привабливих якостей EIGRP є його модульна структура, що забезпечує максимальний рівень масштабованості та адаптованості. Підтримка різних мережевих протоколів (IP, IPX, AppleTalk), реалізована в протоколі EIGRP за допомогою модулів PDM. Фактично EIGRP може бути легко адаптований до нових або модифікованих мережевих протоколів (наприклад, IPv6) шляхом додання нового модуля PDM. На рис. 4.19 показана загальна схема роботи модуля PDM.

а)

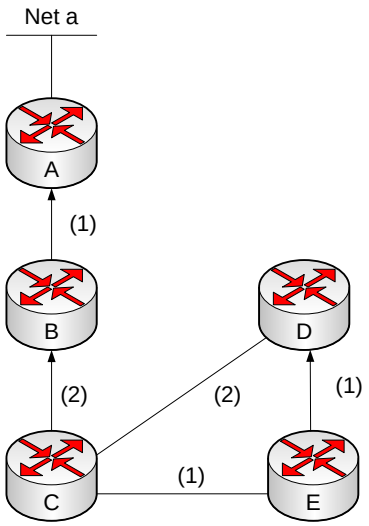


C	EIGRP	FD	RD	Топологія
Net a		3		(FD)
	Через B Через D Через E	3 4 4	1 2 3	(Наступник) (FS)

D	EIGRP	FD	RD	Топологія
Net a		2		(FD)
	Через C Через E	2 5	1 3	(Наступник) (Наступник)

E	EIGRP	FD	RD	Топологія
Net a		3		(FD)
	Через D Через C	3 4	2 3	(Наступник)

б)



C	EIGRP	FD	RD	Топологія
Net a		3		(FD)
	Через B Через D Через E	3	1	(Наступник)

D	EIGRP	FD	RD	Топологія
Net a		5		(FD)
	Через C Через E	5 5	3 4	(Наступник) (Наступник)

E	EIGRP	FD	RD	Топологія
Net a		4		(FD)
	Через C Через D	4	3	(Наступник)

Рисунок 4.18 – Приклад результату роботи алгоритму DUAL: а) мережа до порушення прямого зв’язку між марщрутизаторами D і B; б) мережа після такого порушення

Кожний модуль PDM відповідає за виконання всіх функцій, пов’язаних з відповідним мережним протоколом. Зокрема, модуль IP-EIGRP відповідає за виконання таких функцій:

- відправлення та одержання інформації протоколу EIGRP, що містить дані протоколу IP;
- повідомлення алгоритму DUAL про одержання нової інформації, що стосується IP-маршругизації;
- підтримка результатів прийнятих алгоритмом DUAL рішень про маршрутизацію в таблиці IP-маршрутизації;
- подальше поширення інформації про маршрути, яка стала відома іншим протоколам маршрутизації, що підтримують IP.

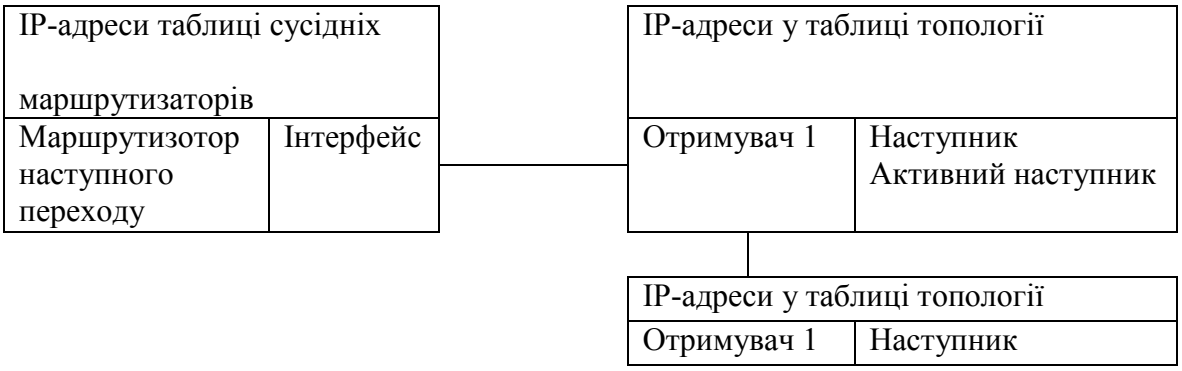


Рисунок 4.19 – Модулі PDM протоколу EIGRP

4.8.5 Типи пакетів протоколу EIGRP

Протокол EIGRP використовує кілька різних типів пакетів для підтримки різних своїх таблиць і встановлення складних (комплексних) зв’язків із сусідніми маршрутизаторами. Нижче наведено п’ять типів пакетів протоколу EIGRP [5, 15]:

- пакети вітання (Hello);
- пакети підтвердження (Acknowledgment);
- пакети відновлення маршрутів (Update);
- пакети запитів (Query);
- пакети відповідей на запити (Reply).

Розглянемо кожен з цих типів пакетів.

Пакети вітання

Протокол EIGRP використовує пакети вітання для виявлення сусідніх маршрутизаторів, їх тестування та повторного виявлення після збоїв. Повторне виявлення відбувається в тому випадку, якщо маршрутизатори не одержують один від одного пакетів вітання протягом часу утримання, але пізніше поновлюють зв’язок.

Маршрутизатори EIGRP розсилають пакети вітання з фіксованим інтервалом (задається у файлі конфігурації), який називається *інтервалом розсилання вітання* (hello interval). Прийнятий за замовчуванням інтервал

вітання залежить від ширини полоси пропускання інтерфейсу, як показано в табл. 4.8 [2]. Для відправлення пакетів вітання протокол EIGRP використовує багатоадресне розсилання.

Таблиця 4.8 – Інтервали розсилання пакетів вітання

Ширина смуги пропускання	Тип каналу	Інтервал вітання за замовчуванням	Час утримання за замовчуванням
Менше або дорівнює 1,544 Мбіт/с	Протокол Multipoint Frame Relay	60 секунд	180 секунд
Більше 1,544 Мбіт/с	Лінія T1, з'єднання "точка – точка"	5 секунд	15 секунд

Нагадаємо, що маршрутизатор протоколу EIGRP зберігає інформацію про сусідні пристрої у ТСП. В ній для кожного сусіднього пристрою є поле послідовного номера, у якому записується номер останнього отриманого від цього пристрою пакета протоколу EIGRP. Іншим полем ТСП є поле часу утримання, в якому записується час одержання останнього пакета. Для того, щоб у сусіднього маршрутизатора зберігався статус пасивного (досяжного і роботоздатного) необхідно, щоб за час утримання від нього надійшов хоча б один пакет. В протилежному випадку протокол EIGRP розглядає цей сусідній маршрутизатор як нероботоздатний і алгоритм DUAL починає перераховувати ТМ. Стандартно час утримання втричі більший інтервалу вітань, однак, адміністратор може сконфігурувати обидва таймери.

У протоколі EIGRP (на відміну від OSPF) для здійснення зв'язку відсутня умова рівності значень інтервалів вітання й блокування на сусідніх маршрутизаторах. При цьому останні дізнаються про інтервали таймерів з пакетами вітання і використовують дану інформацію для встановлення стійкого зв'язку незважаючи на різні інтервали таймерів.

Пакети вітання завжди розсилаються методом негарантованого доставлення і не вимагають підтвердження.

Пакети підтвердження

Маршрутизатор EIGRP використовує пакети підтвердження для того, щоб повідомити інші маршрутизатори про одержання ним пакета EIGRP протягом сеансу „надійного" обміну даними. Надійний транспортний протокол може забезпечити надійний зв'язок між вузлами EIGRP. Для забезпечення гарантованого доставлення, вузол що приймає повинен підтвердити отримання повідомлення від джерела. Для цього використовуються пакети підтвердження (які можна назвати пакетами вітання „без даних"), На відміну від багатоадресних пакетів вітання, ці пакети є одноадресними. Підтвердження також може бути здійснене шляхом суміщення передавання прямих і зворотних пакетів інших типів пакетів EIGRP, таких як пакети відповідей на запити.

Пакети відновлень маршрутів

Пакети відновлень маршрутів використовуються в тих випадках, коли маршрутизатор виявляє новий сусідній пристрій. Тоді маршрутизатор EIGRP надсилає одноадресні пакети відновлення маршрутів цьому новому сусідньому пристрою для того, щоб він міг додати цю інформацію у свою ТТ. Зауважимо, що для передавання новому сусідньому пристрою всієї топологічної інформації може знадобитись більше одного пакета.

Пакети відновлення використовуються також коли маршрутизатор виявляє зміну топології мережі, тоді він надсилає багатоадресні пакети відновлення усім своїм сусідам, попереджаючи їх про таку зміну. Всі пакети відновлення розсилаються методом гарантованого доставлення.

Пакети запитів і відповідей на запити

Маршрутизатор протоколу EIGRP використовує пакети запитів щоразу, коли йому потрібна конкретна інформація від будь-якого зі своїх сусідніх пристроїв. Пакет відповіді використовується для відповіді на запит.

Якщо у маршрутизатора EIGRP зникає первинний маршрут і він не може знайти резервного, то алгоритм DUAL переводить маршрут в активний стан. Після цього маршрутизатор виконує багатоадресне розсилання запиту всім своїм сусідам для знаходження первинного маршруту. Сусідні пристрої повинні надіслати відповіді на запити, в яких або надається інформація про первинний маршрут, або повідомляється про відсутність у них такої інформації.

Запити можуть бути як багато-, так і одноадресними, у той час як відповіді на запити завжди є одноадресними. Обидва типи пакетів розсилаються методом гарантованого доставлення.

4.8.6 Конвергенція протоколу EIGRP

Алгоритм DUAL забезпечує дуже швидку конвергенцію протоколу EIGRP. Для кращого розуміння процесу конвергенції з використанням DUAL, розглянемо схему, наведену на рис. 4.20 [5]. Маршрутизатор RTA може одержати доступ до мережі Network w через три різних маршрутизатори: RTX, RTY або RTZ. Для спрощення обчислень композитна метрика протоколу EIGRP замінена оцінкою для каналу. ТТ маршрутизатора RTA містить список всіх маршрутів, анонсованих сусідніми з ним пристроями.

Як показано у табл. 4.9, маршрутизатор RTA зберігає для кожної мережі реальну (обчислену) оцінку доступу до цієї мережі, а також анонсовану оцінку (повідомлену відстань) від свого сусіднього пристрою.

Спочатку RTY є первинним маршрутом до Network w, оскільки має найменшу обчислену оцінку. Найменша обчислена метрика RTA до Network w (передбачувана відстанню FD до Network w) дорівнює 31.

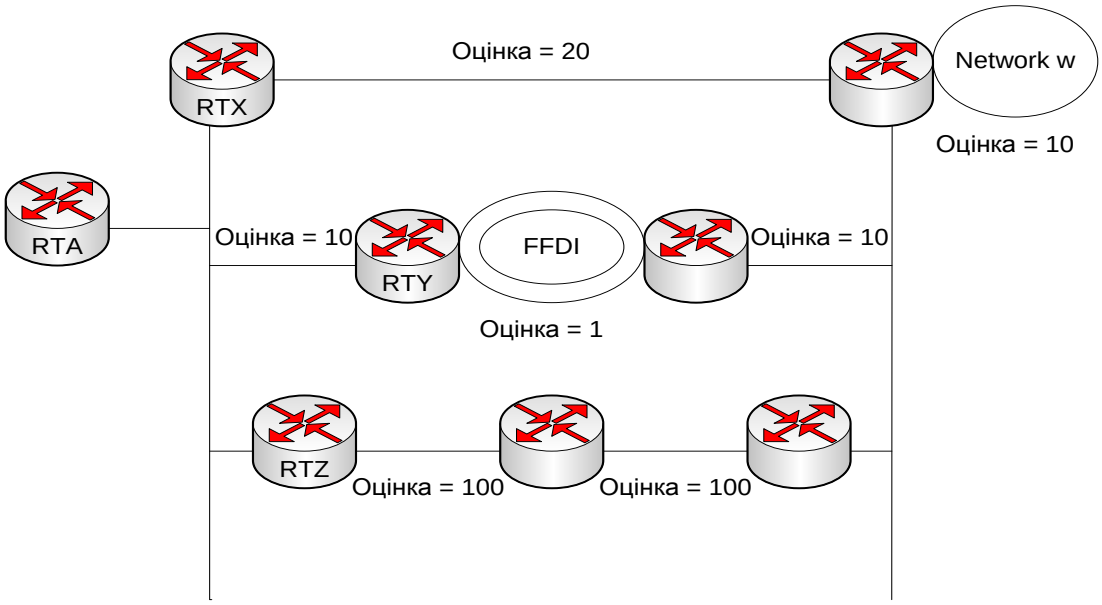


Рисунок 4.20 – Конвергенція протоколу EIGRP

Таблиця 4.9 – Композитна метрика

Сусідній пристрій	Обчислена оцінка маршруту (FD) до мережі Network w	Повідомлена відстань (RD) до мережі Network w
RTY	31	21
RTZ	230	220
RTX	40	30

Для вибору резервного маршруту, який би став первинним до Network w, маршрутизатор RTA виконує процес, що складається з трьох етапів.

Етап 1. Визначається, які сусідні пристрої мають відстань RD до Network w, меншу відстані FD для RTA до Network w. Це відстань FD дорівнює 31; RD для RTX дорівнює 30, а RD для RTZ дорівнює 220. Таким чином, RD для RTX менше поточного FD, у той час як RD для RTZ більше поточного FD.

Етап 2. Визначається мінімальна обчислена оцінка до Network w з доступних маршрутів, що залишились. Обчислена оцінка маршруту через RTX дорівнює 40, а через RTZ дорівнює 230. Таким чином, RTX забезпечує найменшу обчислену оцінку.

Етап 3. Визначається чи задовольняють маршрутизатори критерії першого та другого етапів. Маршрутизатор RTX задовольняє й ті та інші, тому він є резервним маршрутом.

Якщо маршрутизатор RTY стає нероботоздатним, то маршрутизатор RTA негайно переходить до використання маршрутизатора RTX (резервного маршруту) для пересилання пакетів у Network w. Здатність здійснювати негайне перемикання на резервний маршрут є основною передумовою дуже швидкої конвергенції протоколу EIGRP.

Чи може RTZ бути резервним маршрутом? Використовуючи вищена-

ведений триетапний процес, RTA з'ясовує, що RTZ анонсує оцінку 220, яка не менша, ніж відстань FD для RTA (дорівнює 31). Отже, RTZ не може бути резервним маршрутом (поки ще). Відстань FD може змінитись тільки під час переходу з активного у пасивний стан, а цей перехід поки ще не відбувся, тому ця відстань залишається рівною 31. До цього моменту, оскільки для Network w ще не відбувся перехід в активний стан, алгоритм DUAL здійснює процес, який називається локальним обчисленням.

Маршрутизатор RTA не може знайти резервних маршрутів, тому він в кінцевому підсумку переходить від пасивного до активного стану для мережі Network w і запитує свої сусідні пристрої про цю мережу. Даний процес називається обчисленням дифузії (diffusing computation). При переході мережі Network w у активний стан ця відстань FD перевстановлюється, що дозволяє RTA в остаточному підсумку прийняти RTZ як первинний маршрут до мережі Network w.

Тепер знову повернемось до прикладу КМ, наведеної на рис. 4.14а, та ТМ маршрутизатора R2 (рис. 4.14б). Якщо вийде з ладу безпосередній зв'язок між R2 та R3, чи є резервний шлях в ТТ маршрутизатора R2 до мережі 192.168.1.0? Так, є, оскільки метрика маршруту від R1 до мережі 192.168.1.0 складає 2172416, що менше ніж метрика маршруту від R2 до мережі 192.168.1.0, яка дорівнює 3014400 (наявність резервного шляху можна проглянути за допомогою введення на R2 команди `show ip eigrp topology` – приклад 4.2.). При невиконанні даної умови резервного шляху від мережі R2 до мережі 192.168.1.0 не було б і тоді, був би потрібен перерахунок. В даному ж випадку перерахунку не потрібно і при виході з ладу основного шляху зразу ж відбудеться використання резервного.

```
R2# show ip eigrp topology
 <частина результатів виведення опущена>
                . . .
P 192.168.1.0/24, 1 successors, FD is 3014400
   via 192.168.10.10 (3014400/28160) Serial0/0/1
   via 172.16.3.1 (41026560/2172416) Serial0/0/1
<частина результатів виведення опущена>
```

Приклад 4.2 – Основний та резервний шляхи до мережі 192.168.1.0

4.8.7 Конфігування протоколу EIGRP для IP

Незважаючи на складність алгоритму DUAL, конфігурування протоколу EIGRP виявляється відносно простим. Розглянемо процес конфігурування на невеликому прикладі (рис. 4.21) [5].

Для того, щоб сконфігурувати EIGRP для протоколу IP слід виконати такі етапи.

Етап 1. Для увімкнення протоколу EIGRP і визначення автономної системи треба виконати команду

```
Router(config)# router eigrp autonomous-system-number,
```

де параметр autonomous-system-number – це ідентифікатор АС, який

вказує на всі маршрутизатори, що належать даній об'єднаній мережі. Це значення повинно відповідати усім маршрутизаторам в цій об'єднаній мережі. Наприклад, для маршрутизатора A дана команда може бути

```
Router_A(config)# router eigrp 13.
```

Етап 2. Вказати, які мережі належать до даної АС EIGRP на локальному маршрутизаторі за допомогою команди

```
Router(config-router)#  network  network-number,
```

де параметр network-number – це номер мережі. Дана команда задає які інтерфейси даного маршрутизатора беруть участь у роботі протоколу EIGRP і які мережі ним анонсуються. Номер мережі вказується з врахуванням класу IP-адреси. Наприклад, мережі 2.2.0.0 і 2.7.0.0 вводяться за допомогою команди `Router_A(config-router)# network 2.0.0.0`, оскільки вони є підмережами мережі 2.0.0.0.

Команда network конфігурує тільки приєднані мережі. Наприклад, мережа 3.1.0.0 не приєднана безпосередньо до маршрутизатора A. Отже вона не є частиною конфігурації маршрутизатора A.

Якщо треба вказати для протоколу EIGRP лише окремі підмережі, варто скористатись командою

```
Router(config-router)# network network-number wildcard mask,
```

де wildcard mask – інвертована маска, тобто 32-бітне число, яке можна отримати шляхом інвертування маски підмережі. Біти інвертованої маски вказують будуть чи не будуть перевірятись відповідні біти IP-адреси. Там, де біти інвертованої маски нульові – відповідний біт IP-адреси повинен бути перевірений, а де одиничні – ні. Наприклад, якщо для протоколу EIGRP треба вказати тільки підмережу 2.2.0.0 слід ввести команду `Router_A(config-router)# network 2.2.0.0 0.0.255.255`.

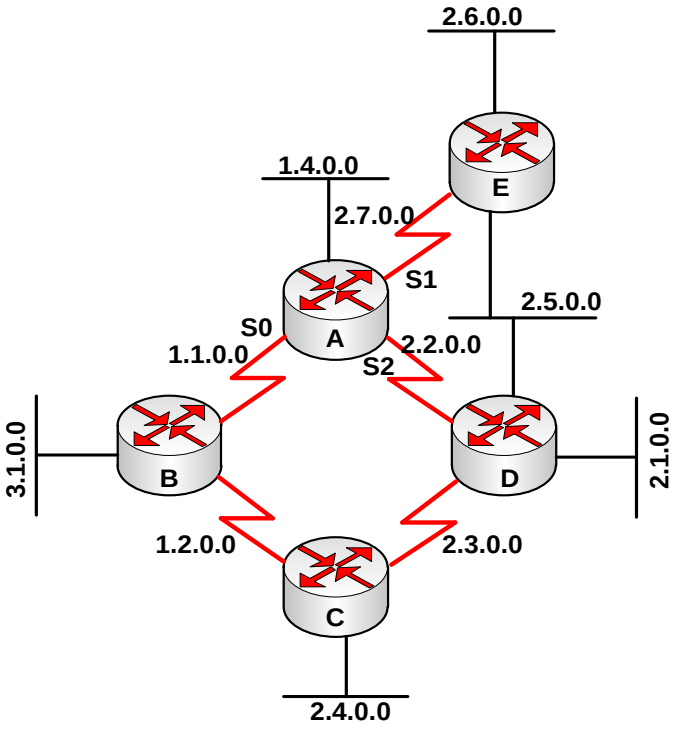


Рисунок 4.21 – Конфігурування EIGRP для протоколу IP

Доцільно зауважити, що ряд ОС дозволяють замість інвертованої маски вводити звичайну маску підмережі. Проте перш ніж використовувати таку можливість, слід дізнатись чи підтримує це дана версія ОС.

Етап 3. Під час конфігурування послідовних каналів, що використовують протокол EIGRP важливо задати смугу пропускання на даному інтерфейсі. Якщо вона для таких інтерфейсів не змінена, то протокол EIGRP приймає для смуги пропускання значення за замовчуванням (замість справжньої ширини смуги пропускання). Якщо канал має меншу швидкість, то маршрутизатор може бути не в змозі виконати конвергенцію або може відбутись втрата змін маршрутизації, або обрано неоптимальний маршрут. Значення смуги пропускання конфігурується за командою

```
Router(config-if)# bandwidth kilobits.
```

Команда для завдання смуги пропускання є єдиною, яка використовується в процесі маршрутизації і повинна бути встановлена відповідно до швидкості каналу для даного інтерфейсу.

Етап 4. Рекомендується також додавати в конфігурацію кожного маршрутизатора E1GRP команду

```
Router(config-if)# eigrp log-neighbor-changes.
```

Ця команда дозволяє записати в системний журнал зміни в станах суміжності (сусідніх пристроїв) для аналізу стійкості системи маршрутизації й допомагає виявляти проблеми, що виникають.

Конфігування смуги пропускання у мережах NBMA

При проектуванні протоколу EIGRP у середовищі нешироковомовної мережі множинного доступу (nonbroadcasl multiaccess – NBMA), такий як мережа Frame Relay, необхідно дотримуватися таких правил [5]:

- швидкість передавання даних протоколу EIGRP не повинна перевищувати узгодженої швидкості передавання інформації (committed information rate – CIR віртуального каналу (virtual circuit –VC);
- агрегований (сукупний) обсяг даних протоколу EIGRP no усім віртуальним каналам не повинен перевищувати швидкість каналу на інтерфейсі;
- смуга пропускання, виділена протоколу EIGRP на кожному каналі VC повинна бути однаковою в обох напрямках.

При правильному розумінні цих правил і виконання їх протокол E1GRP ефективно працює в середовищі розподільної мережі WAN. Якщо при конфігурування протоколу EIGRP у мережі WAN не вжито відповідних заходів, то потоки даних EIGRP можуть викликати переповнення.

Конфігурування смуги пропускання в багатоточковій мережі

Завдання під час конфігування команди bandwidth у середовищі NBMA залежить від того, як спроектовані віртуальні канати VC.

Якщо у багатоточковій конфігурації послідовний канал має багато каналів VC і усі ці канали рівномірно спільно використовують смугу пропускання, то в команді bandwidth повинна бути задана смуга пропускання, яка дорівнює сумі всіх швидкостей CIR. Наприклад, у мережі на рис. 4.22 швидкість CIR кожного каналу VC дорівнює 56 Кбіт/с. Оскільки є чотири канали VC, смуга пропускання повинна бути встановлена рівною 224 (4·56).

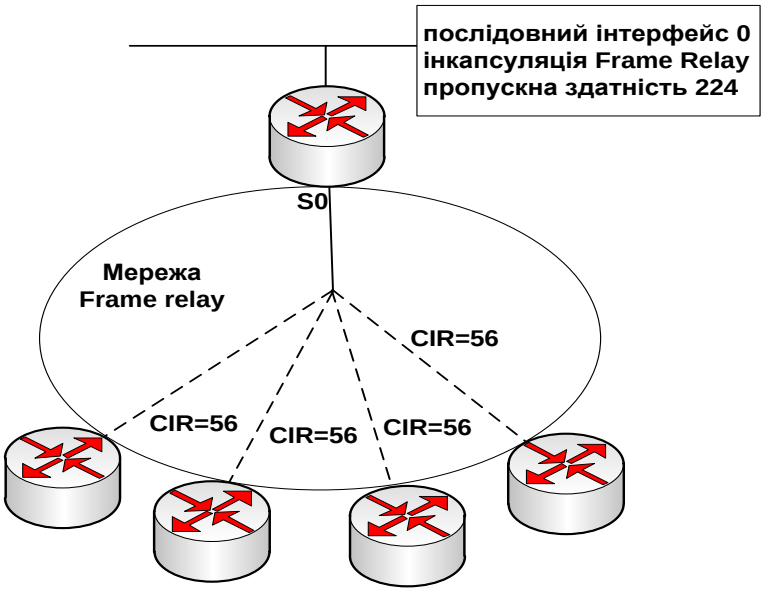


Рисунок 4.22 – Конфігурування EIGRP у багатоточковій мережі WAN

Конфігурування смуги пропускання в гібридній багатоточковій мережі

Якщо у багатоточковій мережі канали VC мають різні швидкості передавання, то потрібне дещо складніше конфігурування. При цьому можуть бути застосовані два нижченаведені основні підходи.

1. *Вибрати найменшу для всіх каналів швидкість CIR і помножити її на кількість віртуальних каналів* (рис. 4.23). Такий підхід застосований до фізичного інтерфейсу. Його недолік полягає в тому, що канали з великою смугою пропускання можуть виявитися недозавантаженими.

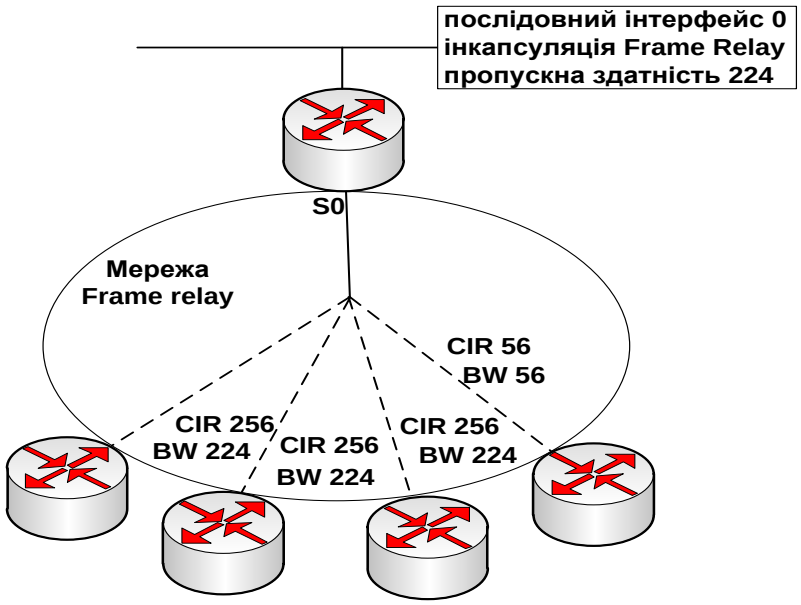


Рисунок 4.23 – Конфігурування EIGRP у багатоточковій гібридній мережі WAN

2. *Використання підінтерфейсів*. Команда bandwidth може бути сконфігурована на кожному підінтерфейсі, що дозволяє використовувати різні швидкості на кожному каналі VC. В цьому випадку підінтерфейси конфігуруються для каналів з різними швидкостями CIR. Канали, що мають одну й ту ж сконфігуровану швидкість CIR являються єдиним підінтерфейсом зі смугою пропускання, що відповідає сукупній швидкості CIR всіх каналів. На рис. 4.24 три віртуальних канали VC мають однакову CIR, що дорівнює 256 Кбіт/с. Вони групуються разом як один багатоточковий послідовний інтерфейс serial 0.1. Єдиний канал VC, що залишається, має меншу CIR (яка дорівнює 56), може бути визначений як послідовний підінтерфейс типу „точка-точка" – serial 0.2.

Використання команди ip bandwidth-percent

Команда `ip bandwidth-percent` задає у відсотковому відношенні частину смуги пропускання, яку протокол EIGRP може використовувати на деякому інтерфейсі. За замовчуванням протокол EIGRP може використовувати до 50 % смуги пропускання інтерфейсу для обміну інформацією маршрутизації. При обчисленні цієї процентної частини команда `ip bandwidth-percent` використовує значення, встановлене командою `bandwidth`. Команду `ip bandwidth-percent` слід використовувати в тих випадках, коли встановлена для каналу смуга пропускання не відповідає його справжній швидкості.

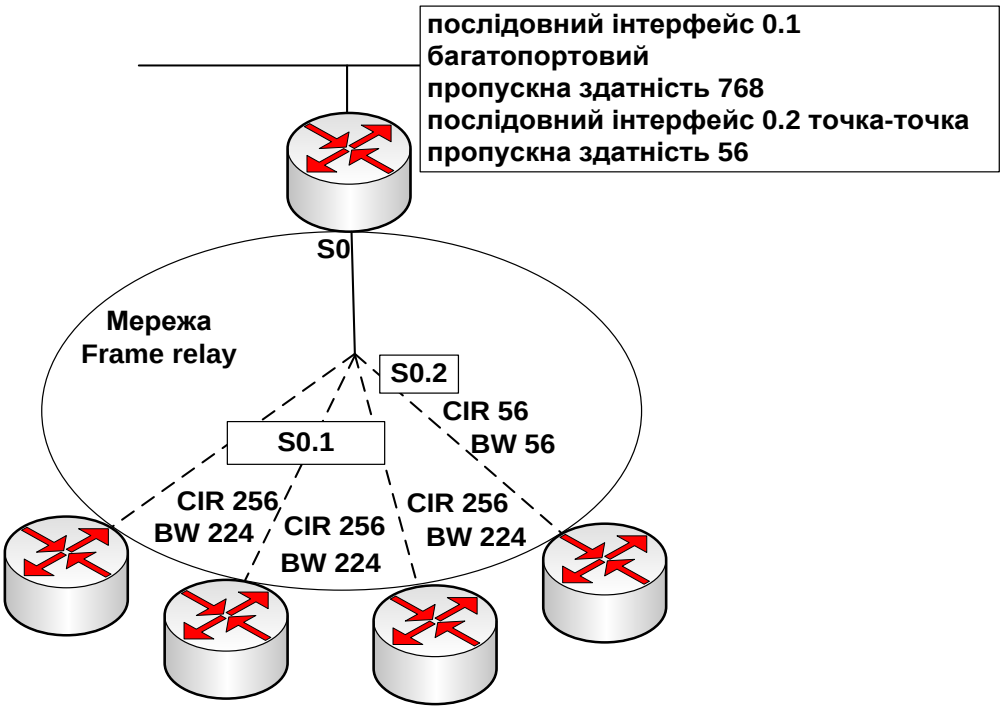


Рисунок 4.24 – Конфігурування EIGRP у багатоточковій гібридній мережі WAN (кращий варіант)

Значення смуги пропускання може бути штучно занижено з різних причин, зокрема, для керування метрикою маршрутизації або для того, щоб відрегулювати надлишкове навантаження у багатоточковій конфігурації протоколу Frame Relay. Незалежно від причини заниження, треба сконфігурувати EIGRP так, щоб замінити штучно занижену смугу пропускання на більш високе значення за допомогою команди ip bandwidth-percent. У деяких випадках значення, що задається цією командою, може навіть перевищувати 100 % [5].

Наприклад, припустимо, що реальна смуга пропускання послідовного каналу маршрутизатора дорівнює 64 Кбіт/с, однак, її значення штучно занижене до 32 Кбіт/с. На рис. 4.25 показано як слід змінити функціонування протоколу EIGRP так, щоб він обмежував обсяг потоків даних протоколу маршрутизації реальною смугою пропускання послідовного інтерфейсу. У наведеному прикладі конфігурації для процеса EIGRP, який функціонує для автономної системи 24, смуга пропускання у відсотках для послідовного інтерфейсу serial 0 встановлюється рівною 100 %. Оскільки 100 % від 32 Кбіт/с дорівнює 32 Кбіт/с, протоколу EIGRP надається можливість використовувати половину реальної смуги пропускання, яка дорівнює 64 Кбіт/с.

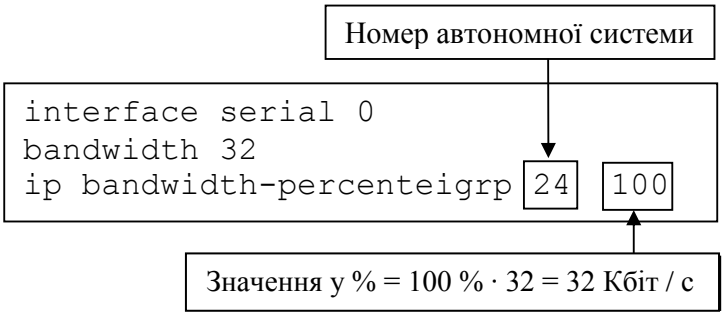


Рисунок 4.25 – Застосування команди ip bandwidth-percent для EIGRP.

Конфігурування узагальнення маршрутів протоколу EIGRP

Протокол EIGRP автоматично узагальнює маршрути на межі мережі, що використовує IP-адреси з класами (тобто на межі мережі, у якій мережева адреса містить у собі клас адреси). Це означає, що незважаючи на те, що маршрутизатор RTC під'єднаний тільки до підмережі 2.1.1.0, він об'являє, що під'єднаний до всієї мережі 2.0.0.0 класу А. У більшості випадків автоматичне узагальнення корисно, оскільки дозволяє зробити ТМ максимально компактними (рис. 4.26).

Однак за деяких обставин автоматичне узагальнення може виявитись небажаним. Якщо в мережі є підмережі, які не є безперервними (як наприклад, на рис. 4.27), то для правильної роботи механізму маршрутизації автоматичне узагальнення необхідно від'єднати (інакше маршрутизатор RTD не прийматиме маршуту до мережі 2.0.0.0/8, що під'єднана до RTC, оскільки він сам безпосередньо під'єднаний до мережі 2.0.0.0/8). Для такого від'єднання використовується команда `Router(config-router)#no auto-summary`.

Рисунок 4.26 – Автоматичне узагальнення маршрутів в EIGRP

При використанні протоколу EIGRP можна вручну сконфігурувати префікс, який буде використовуватись як узагальнена адреса. Ручне конфігурування узагальнення маршрутів здійснюється окремо для кожного інтерфейсу, тому першим повинен бути вибраний інтерфейс, що розповсюджує узагальнення маршрутів. Після цього узагальнена адреса може бути визначена за допомогою команди

```
Router(config-if)#ip summary-address eigrp autonomous-
system-number ip-address mask administrative-distance.
```

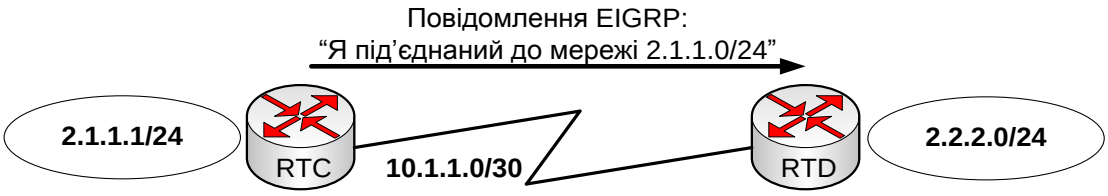


Рисунок 4.27 – Автоматичне узагальнення маршрутів протоколу EIGRP у мережі з розривами

Узагальнені маршрути протоколу EIGRP за замовчуванням мають адміністративну відстань, що дорівнює 5. Однак це значення може бути змінене під час конфігурування на будь-яке значення від 1 до 255.

Маршрутизатор RTC, наведений на рис. 4.27, може бути сконфігурований з використанням команд, наведених у прикладі 4.3.

```
RTC(config)# router eigrp 9
RTC(config-router)#no auto-summary
RTC(config-router)#exit
RTC(config)#interface serial0
RTC(config-if)#ip summary-address eigrp 9 2.1.0.0 255.255.0.0
```

Приклад 4.3 – Ручне узагальнення маршрутів

В результаті виконання команд цього прикладу, RTC додасть до таблиці маршрут `D 2.1.0.0/16 is a summary, 00:00:22, Null0`. Узагальнений маршрут має як джерело не реальний інтерфейс, а Null0, оскільки цей маршрут використовується тільки для цілей анонсування і не являє собою маршруту, який маршрутизатор RTC може обрати для досягнення цієї мережі. В RTC цей маршрут має адміністративну відстань, яка дорівнює 5.

Для маршрутизатора RTD на рис. 4.26. узагальнення маршрутів не має значення, однак, він приймає цей маршрут і призначає йому адміністративну відстань „нормального” маршруту EIGRP (стандартно 90). У конфігу-

рації для маршрутизатора RTC автоматичне узагальнення маршрутів вимкнено командою `no auto-summary`. Якби воно вимкнено не було, то маршрутизатор RTD отримав би два маршрути: сконфігуровану вручну узагальнену адресу (2.1.0.0/16) і призначений автоматично, що використовує класи адреси (2.0.0.0/8). У більшості випадків під час ручного узагальнення слід використовувати команду `no auto-summary`.

Конфігурування аутентифікації у протоколі EIGRP

Для підвищення рівня безпеки протоколу EIGRP на маршрутизаторах слід настроїти аутентифікацію. Така настройка складається з двох кроків.

1. Створити ключову послідовність (key chain), яку будуть використовувати усі маршрутизатори Вашої мережі за допомогою команд, наведених у перших трьох рядках прикладу 4.4. Ці команди створюють ключ, що має ім'я MY_KEY, задають номер ключа (1) та значення рядка ключа (CISCO).

2. Дозволити антентифікацію за алгоритмом MD5 на відповідному інтерфейсі (інтерфейсах) маршрутизатора (останні три рядки прикладу 4.4).

```
Router(config)# key chain MY_KEY
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string CISCO
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# interface serial0/0/0
Router(config-if)# ip autentification mode eigrp 1 md5
Router(config-if)# ip autentification key-chain eigrp 1 MY_KEY
```

Приклад 4.4 – Настроювання аутентифікації протоколу EIGRP

4.8.8 Тестування базової конфігурації протоколу EIGRP

Функція IOS Cisco debug надає корисні команди моніторингу протоколу EIGRP (табл. 4.10).

Прикладами таких команд можуть бути команди show. Основні варіанти команд show, що можуть бути використані для тестування роботи протоколу EIGRP зі стислими описами їх функцій наведені у таблиці 4.11.

Таблиця 4.10 – Основні команди налагоджування протоколу EIGRP

Команда	Опис
`debug eigrp fsm`	Дозволяє спостерігати роботу резервного маршруту протоколу EIGRP і перевірити, що процес маршрутизації встановлює і вилучає поновлення маршрутів
`debug eigrp packet`	Відображає передавання і отримання пакетів протоколу EIGRP. Цими пакетами можуть бути пакети вітання, поновлення маршрутів, запиту або відповіді на запит. У виведенні відображаються послідовні номери і номери підтверджень, використовувані алгоритмом надійного транспортування протоколу EIGRP

4.9 Протокол стану зв'язків OSPF

4.9.1 Загальні відомості та термінологія протоколу OSPF

Протокол OSPF (Open Shortest Path First) є протоколом маршрутизації за станом каналів, що базується на відкритих стандартах. Він описаний в декількох стандартах інженерної групи Internet (Internet Engineering Task Force – IETF), останнім з яких є стандарт RFC 2328. Термін „відкритий" в протоколі OSPF означає його доступність всім користувачам [1, 5, 15, 16].

Протокол OSPF це надійний, масштабованим та ефективний протокол, який може бути використаний в окремій зоні у невеликих КМ і в кількох зонах для великих КМ. Маршрутизація OSPF може бути розширена на великі мережі за умови, що під час проектування КМ використовувались ієрархічні принципи її побудови, які полягають у під'єднанні кількох зон до зони розподілення (нульової зони), яку також називають магістраллю. Таке проектування дозволяє здійснювати повний контроль над повідомленнями про оновлення маршрутів. Задання зон зменшує об'єм службового навантаження маршрутизації, прискорює збіжність, обмежує можливу нестабільність мережі однією зоною та підвищує продуктивність мережі [5, 15].

Таблиця 4.11 – Основні команди Show протоколу EIGRP

Команда	Опис
`show ip eigrp neighbors [type number] [details]`	Відображає таблицю сусідніх пристроїв протоколу EIGRP. Опції type і number використовуються для вказання інтерфейсу. Ключове слово details розширює виведення
`show ip eigrp interfaces [type number] [as-number] [details]`	Відображає інформацію протоколу EIGRP для кожного інтерфейсу. Необов'язкові ключові слова обмежують виведення конкретним інтерфейсом або автономною системою. Ключове слово details розширює виведення
`show ip eigrp topology [as-number]|[[ip-address] mask]`	Відображає всі допустимі резервні маршрути ТТ протоколу EIGRP. Необов'язкові ключові слова можуть використовуватись для фільтрації виведення на основі номера автономної системи або конкретної мережевої адреси
`show ip eigrp topology [active|pending|zero-successors]`	Залежно від використаного ключового слова відображає всі маршрути у ТТ, які є активними, готуються до перерахування, або не мають первинних маршрутів
`show ip eigrp topology all-links`	Відображає не тільки резервні маршрути, а й усі маршрути топології мережі протоколу EIGRP
`Show ip eigrp traffic [as-number]`	Відображає число відправлених і отриманих пакетів протоколу EIGRP. Виведення команди може бути відфільтровано шляхом задання необов'язкового номера автономної системи

Протокол OSPF функціонує не так як дистанційно-векторні протоколи. Маршрутизатори ідентифікують сусідні маршрутизатори та обмінюються з ними інформацією. У протоколу OSPF є свій набір термінів, які наведені на рис. 4.28 [1, 5, 15].

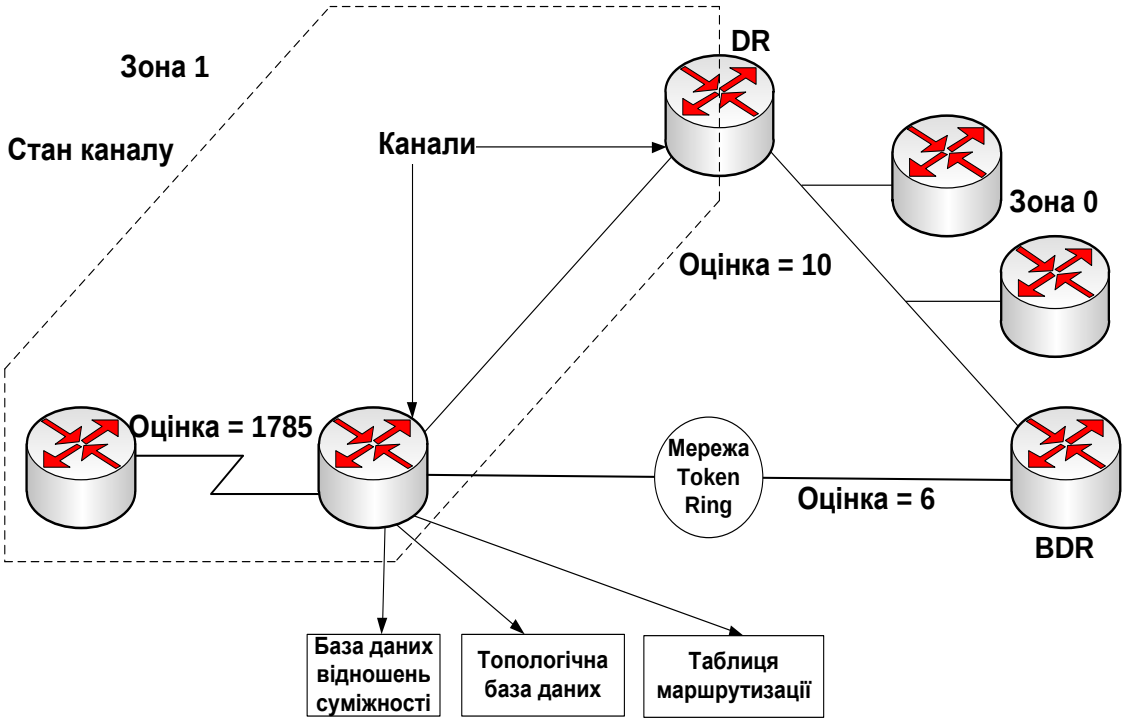


Рисунок 4.28 – Термінологія протоколу OSPF

Інформація, зібрана від сусідніх маршрутизаторів OSPF не є повною ТМ. Кожен OSPF-маршрутизатор повідомляє своїм сусідам про стан своїх зв'язків або каналів. Ця інформація розповсюджується методом лавинного розсилання. Під цим поняттям розуміється відправлення однієї і тієї ж інформації з усіх портів, за виключенням того, на який вона надійшла. Маршрутизатор OSPF оголошує стан своїх каналів та передає далі отриману ним інформацію про стани каналів інших маршрутизаторів.

Маршрутизатори в зоні 1 обробляють цю інформацію та будують свою топологічну БД, яку називають також БД стану каналів. Всі маршрутизатори в одній OSPF-зоні мають одну й ту ж БД стану каналів. Автономна система може бути розділена на ряд зон, що являють собою групи зв'язних (неперервних) мереж і під'єднаних до них пристроїв. Маршрутизатори з кількома інтерфейсуми можуть бути учасниками кількох зон – їх називають граничними маршрутизаторами зон (Area Border Routers). Вони підтримують окремі топологічні БД для кожної зони.

Після цього кожен маршрутизатор застосовує алгоритм вибору найкоротшого шляху SPF, який також називають алгоритмом Дейкстри, до своєї бази даних. Ці обчислення визначають найкращий шлях до пункту призначення. Алгоритм SPF додає вартості (оцінки) для окремих переходів, які зазвичай базуються на ширині смуги пропускання. Мінімальна оцін-

ка маршруту додається до ТМ, що також називається таблицею пересилання.

OSPF-маршрутизатори записують інформацію про своїх сусідів в ТСП. Для зменшення об'єму інформації, якою обмінюються сусідні пристрої в одній мережі, маршрутизатори OSPF обирають *призначений маршрутизатор (Designated Router, DR)* та *резервний призначений маршрутизатор (Backup Designated Router, BDR)*, які служать точками централізації при обміні інформацією маршрутизації [5, 15].

OSPF-маршрутизатори встановлюють зв'язки або стани (states) зі своїми сусідами для ефективного сумісного використання інформації канального рівня.

4.9.2 Стани протоколу OSPF

Маршрутизатори OSPF використовують п'ять різних типів пакетів для ідентифікації своїх сусідів та оновлення інформації маршрутизації канального рівня. В таблиці 4.12 описані типи пакетів протоколу OSPF [5]. Ці п'ять типів пакетів дозволяють протоколу OSPF здійснювати різні та складні типи зв'язків.

Таблиця 4.12 – Типи пакетів протоколу OSPF

Тип пакета протоколу OSPF	Опис
Тип 1 – Hello	Використовується для створення та підтримки таблиці сусідніх пристроїв
Тип 2 – Пакет опису бази даних (Database description packet, DBD)	Описує вміст бази даних стану каналів OSPF-маршрутизатора
Тип 3 – Запит інформації про стан каналів (link-state requests – LSR)	Здійснює запит окремих фрагментів бази даних стану каналів маршрутизатора
Тип 4 – Оновлення стану каналів (Link-state update, LSU)	Передає повідомлення про стан каналів (link-state advertisements, LSA) сусіднім маршрутизаторам
Тип 5 – Підтвердження отримання повідомлення про стан каналів (Link-state acknowledgement, LSAck)	Підтверджує отримання від сусіднього пристрою повідомлення LSA

Ключовим фактором при проектуванні OSPF-мереж та при усуненні помилок в них є розуміння зв'язків або станів, які виникають між OSPF-маршрутизаторами. Інтерфейси OSPF-маршрутизаторів можуть знаходитися в одному з наведених нижче семи станів. Зв'язки між сусідніми маршрутизаторами послідовно проходять ці стани зверху вниз [5, 15]:

- вимкнений стан (Down State);
- ініціалізація (Init State);
- двостороннє з'єднання (Two-way);
- ExStart;
- обмін (Exchange);
- завантаження (Loading);
- стан встановлення повного зв'язку між сусідніми (суміжними) при-

строями (Full adjacency).

Вимкнений стан

Вимкнений стан має місце, коли обмін інформацією між сусідніми пристроями не відбувався. Маршрутизатори очікують переходу в наступний стан – стан ініціалізації.

Стан ініціалізації

В стані ініціалізації OSPF-маршрутизатори регулярно (зазвичай 10 секунд) відсилають пакети першого типу (Hello) для встановлення зв'язку з сусідніми маршрутизаторами. Коли деякий інтерфейс отримує перший Hello-пакет, відповідний маршрутизатор переходить в стан ініціалізації. Це означає, що маршрутизатору відомо про наявність у нього сусіднього пристрою і він чекає переходу зв'язку з ним в наступний стан.

Існує два типи зв'язку між маршрутизаторами: двосторонній зв'язок та стан повного зв'язку сусідніх пристроїв, хоча між цими двома станами і знаходяться декілька проміжних станів. Перед тим, як стане можливим встановлення будь-якого типу зв'язку, маршрутизатор повинен отримати від свого сусіда повідомлення Hello.

Стан двостороннього зв'язку

Кожен OSPF-маршрутизатор намагається встановити з усіма своїми сусідами по мережі OSPF стан двостороннього зв'язку або двонаправленого зв'язку, використовуючи для цього пакети Hello, які зокрема містять список відомих відправнику сусідніх OSPF-маршрутизаторів.

Маршрутизатор переходить в стан двостороннього зв'язку в момент, коли бачить себе в пакеті Hello, отриманому від сусіднього пристрою. Тобто, коли перший маршрутизатор взнає, що другий маршрутизатор знає про нього, він оголошує наявність стану двостороннього зв'язку між ними.

Стан двостороннього зв'язку є базовим станом двох сусідніх пристроїв протоколу OSPF, однак, в ньому ще не відбувається сумісне використання інформації маршрутизації. Для того, щоб дізнатись про стан каналів інших маршрутизаторів і врешті решт створити ТМ, кожен OSPF-маршрутизатор повинен утворити хоча б одне з'єднання (стан суміжності) з сусіднім пристроєм. Стан суміжності це більш тісний зв'язок між OSPF-маршрутизаторами, що включає в себе ряд послідовних станів, які базуються не лише на Hello-повідомленнях, а й інших чотирьох типах OSPF-пакетів. Маршрутизатори, які намагаються стати суміжними обмінюються інформацією ще до того, як буде повністю встановлено стан суміжності. Першим етапом встановлення стану повної суміжності є стан ExStart.

Стан ExStart

В технічному аспекті в момент, коли маршрутизатор та його сусідній пристрій входять у стан ExStart, їх зв'язок характеризується як стан суміжності, однак, в дійсності ці пристрої ще не є повністю суміжними. Стан ExStart встановлюється за допомогою пакетів опису бази даних (DBD). Для обговорення того, який маршрутизатор в даному з'єднанні буде головним

(master), а який підлеглим (slave), маршрутизатори використовують пакети Hello, а для обміну вмістом БД використовуються пакети DBD (рис. 4.29).

Маршрутизатор з максимальним значенням OSPF-ідентифікатора (ID) стає головним. Коли два сусідніх маршрутизатора визначають свої ролі як головного та підлеглого, вони входять у стан обміну (Exchange) та починають надсилати один одному інформацію маршрутизації.

Стан обміну

В стані обміну сусідні маршрутизатори використовують пакети DBD для відправлення один одному своєї інформації про стан каналів, як показано на рис. 4.29. Іншими словами маршрутизатори описують один одному свої БД стану каналів. При цьому маршрутизатори порівнюють отриману інформацію з тією, що міститься в їх власних БД стану каналів. Якщо будь-який з маршрутизаторів отримує інформацію про канал, яка відсутня в його БД – він запитує у сусіднього маршрутизатора повне оновлення. Повний обмін інформації відбувається в стані завантаження (Loading).

Стан завантаження

Після того, як обидва маршрутизатора описали один одному свої БД, вони можуть запитати більш повну інформацію, використовуючи пакети запиту стану каналів (LSR). Коли маршрутизатор отримує запит LSR, він відповідає відправкою оновлення маршрутизації, використовуючи пакет оновлення стану каналів (LSU). Ці LSU-пакети містять оголошення актуального стану каналів (LSA), які складають сутність протоколів маршрутизації стану каналів. Як показано на рис. 4.29, підтвердження отримання LSU-пакетів здійснюється за допомогою пакетів підтвердження стану каналів (LSAck).

Стан повної суміжності

Після того, як повністю реалізований стан завантаження, маршрутизатори повністю суміжні. Кожен маршрутизатор підтримує свій список суміжних сусідніх маршрутизаторів (БД суміжних пристроїв).

В табл. 4.13 перераховано важливі бази даних протоколу OSPF [5, 15].

Таблиця 4.13 – Бази даних протоколу OSPF

База даних	Опис
База даних суміжних пристроїв	Список усіх сусідніх пристроїв, з якими даний маршрутизатор встановив двосторонні зв’язки
База даних стану каналів або топологічна база даних (Link State Data Base)	Інформація про всі маршрутизатори мережі. Ця база даних відображає поточну мережеву топологію. Всі маршрутизатори однієї і тієї ж області мають ідентичні бази даних канального рівня
База даних пересилання або таблиця маршрутизації (Routing Table)	Список маршрутів, що генерується при виконанні алгоритму над топологічною базою даних. Таблиця маршрутизації кожного маршрутизатора унікальна та містить інформацію про те, яким чином і за якими маршрутами слід направляти пакети, призначені іншим маршрутизаторам

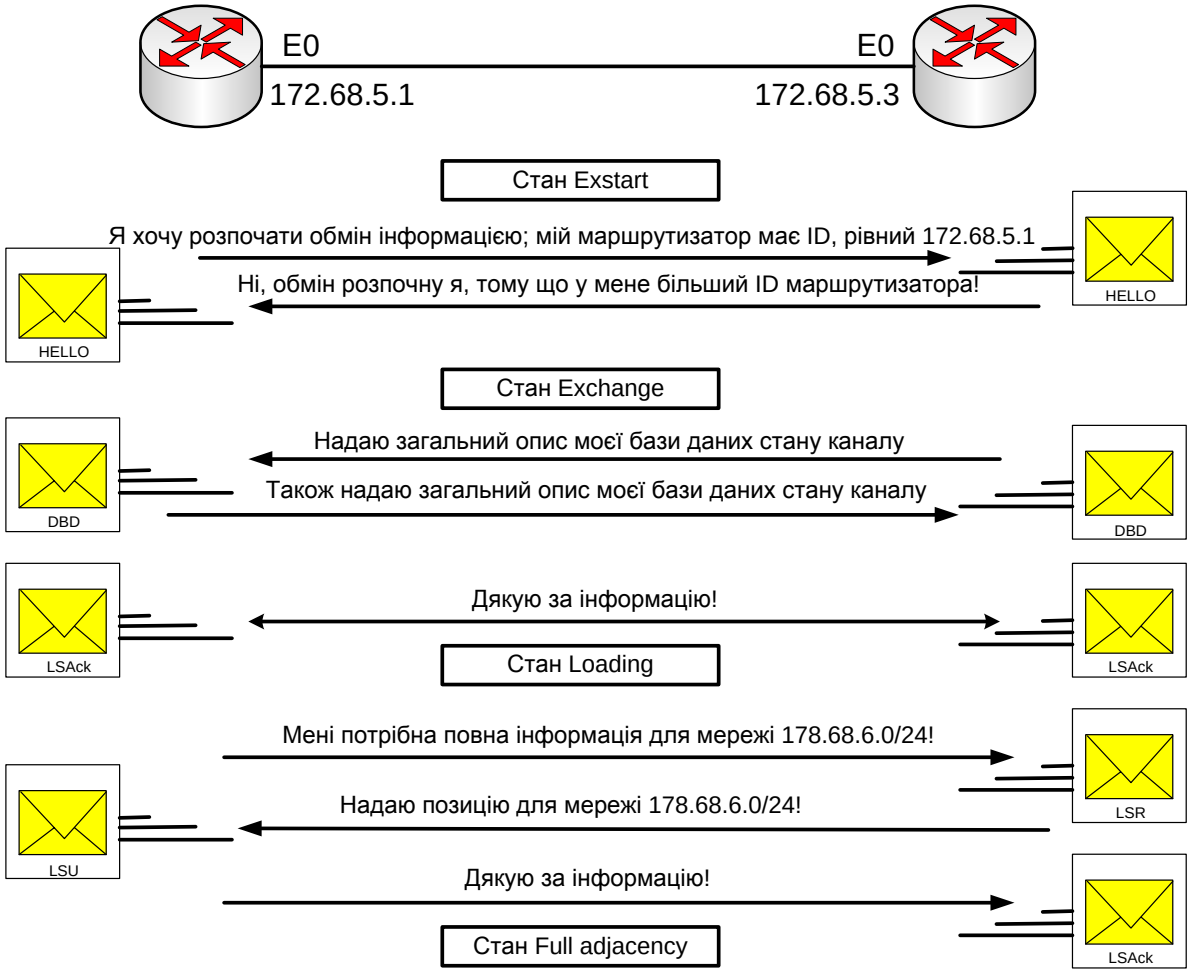


Рисунок 4.29 – Виявлення маршрутизатора за протоколом OSPF

4.9.3 Основи функціонування протоколу OSPF

Для визначення найкращого шляху до пункту призначення протокол OSPF використовує алгоритм вибору найкоротшого маршруту (тобто маршруту з найменшою оцінкою). Цей алгоритм було розроблено голандським комп'ютерним спеціалістом Дейкстра (Dijkstra) та опубліковано у 1959 році. В цьому алгоритмі КМ розглядається як множина вузлів, що з'єднані між собою каналами типу „точка-точка". Кожному каналу присвоюється деяке значення оцінки, а кожному вузлу – деяке ім'я. Кожен вузол має повну БД всіх каналів, тому всім вузлам відома вся інформація про фізичну топологію мережі. Після цього алгоритм вибору найкоротшого шляху обчислює вільну від петель топологію, використовуючи даний вузол як початкову точки та послідовно аналізуючи його інформацію про суміжні вузли.

Для того, щоб сумісно використовувати інформацію про маршрутизацію, OSPF-маршрутизатори повинні встановити зв'язок з сусіднам. Кожен

маршрутизатор намагається встановити відношення суміжності або сусідства хоча б з одним маршрутизатором кожної IP-мережі, до якої під'єднані усі його порти. Деякі маршрутизатори можуть намагатися встановити відношення суміжності з усіма сусідніми маршрутизаторами, в той час як інші – тільки з одним або двома. OSPF-маршрутизатори визначають, з якими іншими маршрутизаторами їм слід встановити відношення суміжності, на основі типу мережі, яка їх поєднує.

Після того, як між сусідніми пристроями встановлені відношення суміжності, між ними відбувається обмін інформацією про стан каналу. Як показано на рис. 4.30, і перераховано в наведеному нижче списку, інтерфейси OSPF-маршрутизаторів розпізнають три типи мереж [5, 15].

1. Широкомовні мережі множинного доступу.
2. Нешироковмовні мережі множинного доступу (nonbroadcast multiaccess – NBMA).
3. Мережі з каналами типу „точка-точка" .

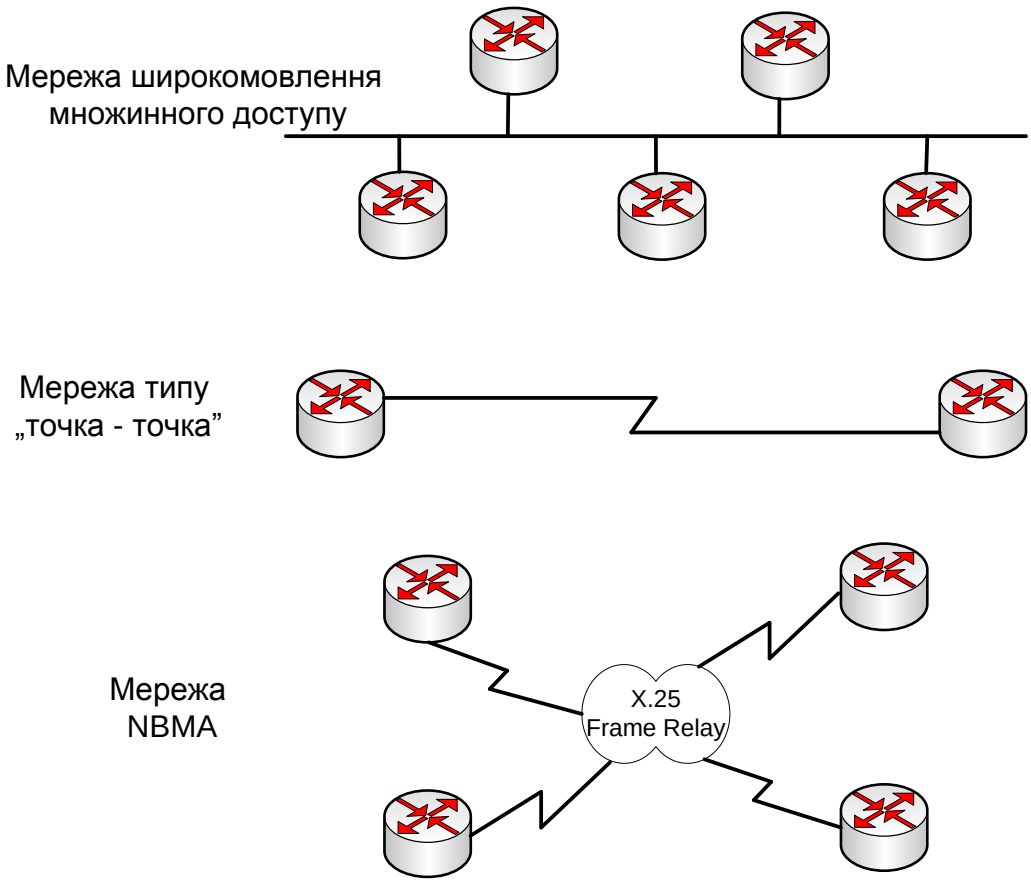


Рисунок 4.30 – Типи OSPF-мереж

Мережевий адміністратор може сконфігурувати на будь-якому типі інтерфейсу і четвертий тип мереж – мережа типу „точка – декілька точок". В табл. 4.13 наведені типи OSPF-мереж. В мережі *множинного доступу* (*multiaccess network*) неможливо заздалегідь знати, скільки маршрутизато-

рів буде з'єднано. В мережах типу „*точка-точка*" (*point-to-point*) можуть бути з'єднані тільки два маршрутизатори. Якщо всі маршрутизатори встановлять відношення суміжності з усіма іншими і будуть обмінюватися інформацією про стан каналів, то об'єм службових повідомлень стане занадто великим. Як згадувалось вище, проблема великого об'єму службових повідомлень, може бути вирішена вибором призначеного маршрутизатора.

Таблиця 4.13 – Типи мереж OSPF

Тип мережі	Характеристики, що визначаються	Чи є вибір DR-маршрутизатора?
Широкомовний множинний доступ	Ethernet, Token Ring, FDDI	Так
Нешироковомовний множинний доступ	Frame Relay, X.25, SMDS	Так
„Точка-точка"	PPP, HDLC	Ні
„Точка-декілька точок"	Конфігурується мережевим адміністратором	Ні

Цей призначений маршрутизатор (DR) стає суміжним пристроєм для всіх маршрутизаторів широкомовного сегмента. Всі інші маршрутизатори цього сегмента надсилають інформацію про стан каналу до DR, який стає джерелом інформації для даного сегмента і розсилає інформацію про стан каналів всім іншим маршрутизаторам сегмента, використовуючи адресу багатоадресного розсилання 224.0.0.5 для всіх OSPF-маршрутизаторів. Незважаючи на підвищення ефективності роботи КМ, яке забезпечується використанням DR, в даному підході є й недолік – призначений маршрутизатор являє собою точку, від якої залежить робота всього сегмента і у випадку виходу його з ладу весь сегмент припиняє працювати. Тому вибирається також резервний призначений маршрутизатор (BDR), який приймає на себе виконання функцій призначеного маршрутизатора у випадку відмови останнього. На рис. 4.31 наведено маршрутизатори DR та BDR, що отримують повідомлення LSA. Для того, щоб обоє маршрутизатори DR та BDR отримували всі повідомлення про стан каналу, які надсилаються в сегмент, використовується адреса багатоадресого розсилання 224.0.0.6.

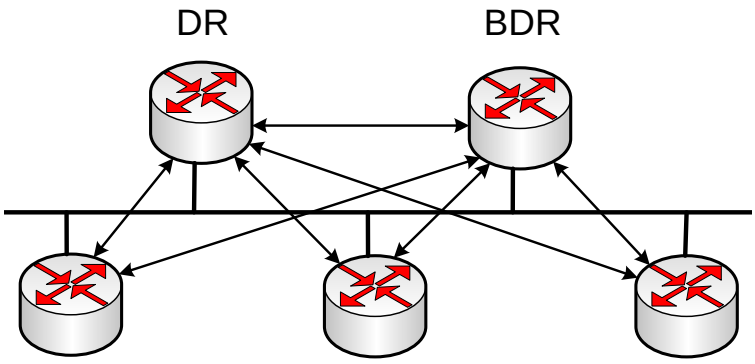


Рисунок 4.31 – Маршрутизатори DR та BDR отримують повідомлення LSA

Зазначимо, що маршрутизатор стає DR, якщо він має найвищий (найбільший) пріоритет інтерфейсу (OSPF interface priority), маршрутизатор з другим за величиною пріоритетом стає BDR. Якщо значення цих пріорите-

тів однакові (а за замовчуванням вони однакові і дорівнюють одиниці) – то до уваги береться ідентифікатора маршрутизатора (Router ID). Маршрутизатор з найбільшим значенням ID стає DR, а з другим за величиною пріоритетом – BDR. Ідентифікатором маршрутизатора стає найбільша IP-адреса Loopback-інтерфейсу або, якщо Loopback-інтерфейс не налаштований – найбільша IP-адреса маршрутизатора.

В мережах типу „точка-точка" існує лише два вузла і тому маршрутизатори DR та BDR не обираються. Обидва маршрутизатори з'єднання „точка-точка" є повністю суміжними пристроями.

Для визначення кращого маршруту протокол OSPF використовує оцінку як метрику, яка обчислюється за виразом [5, 15]:

$$10^8/(\text{ширина смуги пропускання інтерфейсу}).$$

Для того, щоб протокол OSPF правильно обчислював характеристики маршрутів, необхідно, щоб усі інтерфейси, під'єднані до будь-якого каналу, домовились про його оцінку. Ця оцінка може бути змінена для того, щоб здійснити вплив на результат обчислення протоколом OSPF його оцінки. Найбільш типовою ситуацією, в якій потрібно змінювати оцінку, є використання маршрутизаторів від різних виробників. Це пов'язано з тим, що оцінки каналу, зроблені різними пристроями, можуть бути різними.

В таблиці 4.14 наведені стандартні оцінки каналів [5, 15]. Зазначимо, що у випадку, коли маршрут до пункту призначення проходить через кілька сегментів – оцінка маршруту дорівнюватиме сумі оцінок цих сегментів.

Таблиця 4.14 – Деякі стандартні оцінки протоколу OSPF

Середовище передавання	Оцінка
Послідовний канал 56 Кбіт/с	1785
Послідовний канал 64 Кбіт/с	1562
T1 (Послідовний канал 1,544 Мбіт/с)	64
E1 (Послідовний канал 2,048 Мбіт/с)	48
Мережа Ethernet 10 Мбіт/с	10
Мережа Token Ring 16 Мбіт/с	6
100 Mbps Fast Ethernet, FDDI	1

4.9.4 Конфігурування протоколу OSPF

Для того, щоб увійти в режим настроювання протоколу OSPF слід ввести команду [5]:

```
Router(Config)# router ospf process-id,
```

де `process-id` – номер у діапазоні 1 – 65535, який має локальне значення і на відміну від EIGRP, не повинен збігатися на всіх маршрутизаторах мережі, в якій настроюється протокол OSPF. При цьому маршрутизатори мережі будуть бачити один одного і обмінюватись OSPF-анонсами.

Далі на кожному маршрутизаторі слід виконати команду

```
Router(config-router)#network net-addr wildcard-mask area a-id,
```

де `net-addr` – IP-адреса мережі або підмережі, яка безпосередньо

під'єднана до даного маршрутизатора (в кожній команді `network` вказується по одній такій мережі); `wildcard-mask` – шаблонна маска (на відміну від протоколу EIGRP є обов'язковою), яка в даному випадку вказує на ті розряди IP-адреси, які слід порівняти з шаблоном (шаблонна маска може бути отримана шляхом інвертування двійкових розрядів звичайної маски підмережі); `a-id` – ідентифікатор зони OSPF, в цій зоні всі OSPF-маршрутизатори обмінюються інформацією між собою і мають однакову базу даних link-state databases. Всі маршрутизатори в одній зоні повинні мати однакове значення `a-id`. Хоча це значення може бути довільне, прийнято використовувати 0, якщо є лише одна зона OSPF. В подальшому, при доданні нових зон нульова зона є магістральною (backbone area).

Дана команда виконує такі функції:

- всі інтерфейси маршрутизатора, які мають адреси, що належать мережам вказаним командами `network` беруть участь у надсиланні та отриманні OSPF-анонсів;
- ці мережі (підмережі) будуть включені в OSPF-анонси.

Так, наприклад, для мережі, наведеної на рис. 4.32, для маршрутизатора R1 команди настроювання протоколу OSPF будуть

```
R1(config)#router ospf 1
R1(config-router)# network 172.16.1.16 0.0.0.15 area 0
R1(config-router)# network 192.168.10.0 0.0.0.3 area 0
R1(config-router)# network 192.168.10.4 0.0.0.3 area 0;
```

для маршрутизатора R2

```
R2(config)#router ospf 1
R2(config-router)# network 10.10.10.0 0.0.0.255 area 0
R2(config-router)# network 192.168.10.0 0.0.0.3 area 0
R2(config-router)# network 192.168.10.8 0.0.0.3 area 0;
```

для маршрутизатора R3

```
R3(config)#router ospf 1
R3(config-router)# network 172.16.1.32 0.0.0.7 area 0
R3(config-router)# network 192.168.10.4 0.0.0.3 area 0
R3(config-router)# network 192.168.10.8 0.0.0.3 area 0.
```

Після виконання цих команд ТМ будуть:

для маршрутизатора R1

```
   10.0.0.0/24 is subnetted, 1 subnets
O 10.10.10.0 [110/65] via 192.168.10.2, 00:00:37, Serial0/0/0
  172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.16.1.16/28 is directly connected, FastEthernet0/0
O 172.16.1.32/29 [110/65] via 192.168.10.6, 00:01:58, Serial0/0/1
   192.168.10.0/30 is subnetted, 3 subnets
C 192.168.10.0 is directly connected, Serial0/0/0
C 192.168.10.4 is directly connected, Serial0/0/1
O 192.168.10.8 [110/128] via 192.168.10.6, 00:02:52, Serial0/0/1
               [110/128] via 192.168.10.2, 00:00:13, Serial0/0/0
```

для маршрутизатора R2

```
   10.0.0.0/24 is subnetted, 1 subnets
```

```
C 10.10.10.0 is directly connected, FastEthernet0/0
  172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
O 172.16.1.16/28 [110/65] via 192.168.10.1, 00:01:28, Serial0/0/0
O 172.16.1.32/29 [110/65] via 192.168.10.10, 00:00:54, Serial0/0/1
   192.168.10.0/30 is subnetted, 3 subnets
C 192.168.10.0 is directly connected, Serial0/0/0
O 192.168.10.4 [110/128] via 192.168.10.1, 00:01:28, Serial0/0/0
               [110/128] via 192.168.10.10, 00:00:54, Serial0/0/1
C 192.168.10.8 is directly connected, Serial0/0/1
```

для маршрутизатора R3

```
   10.0.0.0/24 is subnetted, 1 subnets
O 10.10.10.0 [110/65] via 192.168.10.9, 00:01:20, Serial0/0/1
  172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
O 172.16.1.16/28 [110/65] via 192.168.10.5, 00:04:07, Serial0/0/0
C 172.16.1.32/29 is directly connected, FastEthernet0/0
   192.168.10.0/30 is subnetted, 3 subnets
O 192.168.10.0 [110/128] via 192.168.10.5, 00:04:07, Serial0/0/0
               [110/128] via 192.168.10.9, 00:01:20, Serial0/0/1
C 192.168.10.4 is directly connected, Serial0/0/0
C 192.168.10.8 is directly connected, Serial0/0/1
```

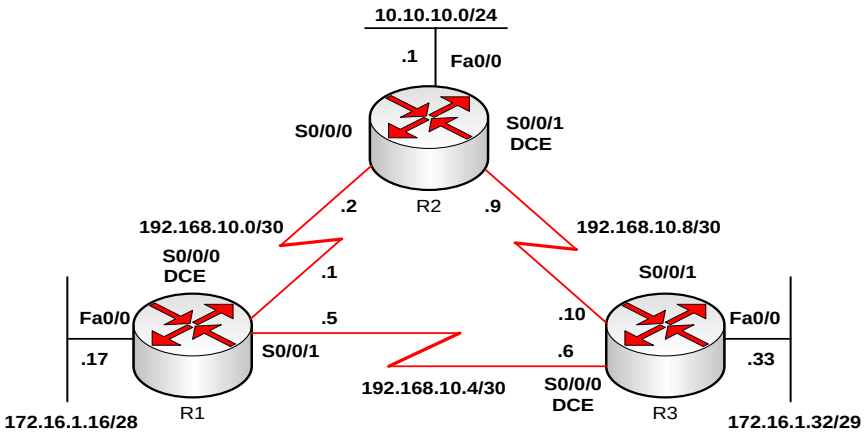


Рисунок 4.32 – Структура мережі для настроювання протоколу OSPF

Літерами "O" в ТМ починаються маршрути отримані за протоколом OSPF. Розглянемо, наприклад, перший рядок ТМ маршрутизатора R1

```
O 10.10.10.0 [110/65] via 192.168.10.2, 00:00:37, Serial0/0/0.
```

Отже, тут 10.10.10.0 – адреса мережі призначення; [110/65] – адміністративна відстань протоколу OSPF і через слеш – метрика маршуту; адреса порту наступного транзитного вузла на шляху до мережі призначення, 00:00:37 – час існування даного маршруту; Serial0/0/0 – локальний інтерфейс, через який слід надсилати пакети до мережі призначення.

Останній запис маршруту ТМ маршрутизатора R1 має вигляд

```
O 192.168.10.8 [110/128] via 192.168.10.6, 00:02:52, Serial0/0/1
               [110/128] via 192.168.10.2, 00:00:13, Serial0/0/0
```

Це означає, що до мережі `192.168.10.8` є два еквівалентні маршрути з метрикою 128, через потри маршрутизаторів R3 та R2, відповідно.

Обчислення метрики

Як зазначалось вище, метрика маршруту у протоколі OSPF обчислюється як сумарна оцінка каналів цього маршруту від джерела до пункту призначення. Так, наприклад, метрика маршруту від маршрутизатора R1 до мережі 10.10.10.0 буде 64 + 1 = 65, де 64 – оцінка каналу між R1 та R2 з пропускною спроможністю 1,544 Мбіт/с (див. табл. 4.14), а 1 – оцінка каналу від R2 до мережі 10.10.10.0/24. Аналогічно, метрика маршруту від маршрутизатора R1 до мережі 192.168.10.8 буде 64 + 64=128.

Подивитись пропускну спроможність інтерфейсу маршрутизатора можна за допомогою команди `show interfaces int num` (де – ім'я та номер інтерфейсу, відповідно).

Нижче наведено результати виконання цієї команди для деяких інтерфейсів маршрутизаторів R1 та R2 (на прикладах виведено лише один рядок, який містить інформацію стосовно пропускної здатності – Bw).

```
R1#show interfaces serial 0/0/0
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
R2#show interfaces serial 0/0/0
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
R2#show interfaces fastEthernet 0/0
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
R2#show interfaces serial 0/0/1
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec.
```

Слід звернути увагу на те, що під час конфігурування OSPF не слід покладатись на значення смуги пропускання послідовних каналів за замовчуванням, оскільки ці значення можуть не відповідати тим, що потрібні і бути не тими, що ви передбачаєте. Для задання потрібного значення смуги пропускання слід скористатись командою

`Router(config-if)#bandwidth bandwidth-kbps`.

Можна також скористатись командою `ip ospf cost cost_value` (де cost_value – оцінка маршруту) і задати вже саму оцінку каналу маршруту. Наприклад, команда `R1(config-if)#ip ospf cost 1562` еквівалетна команді `R1(config-if)#bandwidth 64`, за виключенням того, що в першому випадку обчислення вартості виконувати не слід.

Анонсування маршруту за замовчуванням та статичного маршруту

Аналогічно протоколу RIP, для анонсування маршруту за замовчуванням OSPF потребує задання команди `default-information originate` в режимі конфігурування протоколу OSPF. Такий анонсований маршрут в ТМ інших OSPF-маршрутизаторів буде починатись з символів O*E2 (або O*E1).

Налаштуємо на маршрутизаторі R1 Loopback 1 з IP-адресою 172.30.1.1, задамо маршрут за замовчуванням до цього інтерфейсу та вкажемо маршрутизатору на анонсування цього маршруту:

```
R1(config)#interface loopback 1
R1(config-if)#ip add 172.30.1.1 255.255.255.252
R1(config-if)#exit
R1(config)#ip route 0.0.0.0 0.0.0.0 loopback 1
R1(config)#router ospf 1
R1(config-router)#default-information originate
```

Тепер маршрут за замовчуванням анонсуватиметься до маршрутизаторів R2, R3. В ТМ маршрутизатора R1 з'явиться запис

```
S* 0.0.0.0/0 is directly connected, Loopback1,
```

а в ТМ маршрутизатора R3 –

```
O*E2 0.0.0.0/0 [110/1] via 192.168.10.5, 00:00:20, Serial0/0/0
```

Анонсування статичних маршрутів можна задати за допомогою команди `redistribute static` в режимі конфігурування протоколу OSPF.

Конфігурування аутентифікації в протоколі OSPF

Рівень безпеки мережі підвищується, якщо відомо, що маршрутна інформація поступила з конкретного джерела. Протокол OSPF дозволяє маршрутизаторам виконувати взаємну аутентифікацію. За замовчуванням маршрутизатор покладається на те, що:

- інформація про маршрути надходить від того маршрутизатора, який повинен її надсилати;
- в процесі передавання ця інформація не була спотворена.

Для гарантування цього, на маршрутизаторах однієї зони може бути сконфігурована взаємна аутентифікація [2].

Аутентифікація є іншим типом конфігурування окремих інтерфейсів. Кожному OSPF-інтерфейсу маршрутизатора може бути заданий відмінний від інших ключ аутентифікації, який виконує функції пароля для маршрутизаторів OSPF однієї і тієї ж зони. Для конфігурування OSPF-аутентифікації використовується команда

```
Router(config-if)#ip ospf authentication-key password.
```

Після конфігурування пароля, в зоні можна увімкнути функцію аутентифікації за допомогою команди

```
Router(config-router)#area num authentication [message-digest],
```

яка повинна бути виконана на всіх маршрутизаторах, що беруть участь в аутентифікації. Хоча ключове слово message-digest необов'язкове, рекомендується завжди використовувати його в даній команді, оскільки за замовчуванням паролі аутентифікації пересилаються відкритим текстом. При використанні ключового слова message-digest замість пароля пересилається дайджест повідомлення (хеш пароля). Якщо у одержувача сконфігуровано відповідний ключ аутентифікації, то потенційний зловмисник не зможе зрозуміти сенс цього дайджеста.

Якщо вибрана аутентифікація з використанням дайджеста повідомлення, то ключ аутентифікації не використовується. Натомість на інтерфейсі OSPF-маршрутизатора повинен бути сконфігурований ключ дайджеста повідомлення за допомогою команди

```
Router(config-if)#ip ospf message-digest-key key-id
                  md5 [encryption-type] password
```

Аутентифікація MD5 створює дайджест повідомлення, який є кодованими даними, створеними на базі пароля і вмісту пакета. Маршрутизатор-одержувач використовує для відновлення дайджеста спільно використовуваний пароль і цей пакет. Якщо дайджести збігаються – маршрутизатор вважає, що джерелу пакета можна довіряти і вміст пакета не був спотворений (підроблений) в процесі передавання.

Конфігурування таймерів протоколу OSPF

В деяких випадках необхідно прискорення сповіщення маршрутизаторів мережі про збої в роботі каналів. У протоколі OSPF з цією метою використовуються таймери.

Нагадаємо, що для того, щоб OSPF-маршрутизатори могли обмінюватися інформацією, вони повинні мати однакові інтервали розсилання повідомлень Hello і критичні інтервали. За замовчуванням критичний інтервал має значення в чотири рази більше, ніж інтервал розсилання повідомлень Hello. Це означає, що маршрутизатор має можливість чотири рази надіслати повідомлення Hello до того, як він буде оголошений нероботоздатним. У широкомовних мережах OSPF за замовчуванням інтервал повідомлень Hello дорівнює 10 секунд, а інтервал критичних повідомлень – 40 секунд. У нешироковомовних мережах OSPF ці інтервали дорівнюють 30 і 120 секунд, відповідно. Ці стандартні значення забезпечують ефективне функціонування протоколу OSPF, тому їх не рекомендується змінювати. Мережевий адміністратор може змінити ці значення, проте, для цього слід мати достатні підстави вважати, що така зміна підвищить ефективність роботи мережі. При конфігурування таймерів необхідно стежити, щоб у всіх маршрутизаторів ці значення збігались. При конфігурації на інтерфейсі інтервалів Hello і критичного використовуються команди:

```
Router(config-if)# ip ospf hello-interval seconds,
Router(config-if)# ip ospf dead-interval seconds.
```

Конфігурування пріоритета інтерфейсу для протоколу OSPF

Таке конфігурування дозволяє мереженому адміністратору впливати на процес вибору маршрутизаторів DR та BDR і виконується за допомогою команди `Router(config-if)#ip ospf priority num`,

де num – число з діапазону 0 – 255, яке і визначає цей пріоритет.

4.9.5 Тестування протоколу OSPF

Команда `show ip route` дозволяє проглянути ТМ маршрутизтора.

Команда `show ip protocols` показує протоколи мереженого рівня, які працюють на маршрутизаторі. Для протоколу OSPF вона дозволяє побачити ідентифікатор маршрутизотора (router ID). В деяких версіях IOS номер показано не буде, тоді його можна проглянути за командами `show ip ospf` або `show ip ospf interface`. Остання команда дозволяє також

подивитись значення таймерів протоколу OSPF та значення оцінки відповідного інтерфейсу [5, 15].

Команда `show ip ospf neighbor` показує сусідні OSPF-маршрутизатори, які знаходяться у сусідських відношеннях з даним маршрутизатором. З цієї таблиці можна дізнатись, зокрема, про ідентифікатори сусідніх маршрутизаторів (поле Neighbor ID); значення пріоритету інтерфейсу маршрутизатора (Pri) – використовується в процесі вибору DR та BDR; стан інтерфейсу для протоколу OSPF (State). Наприклад, стан Full означає, що ці маршрутизатори мають ідентичні топологічні бази даних; час, що буде чекати маршрутизатор на отримання пакета Hello, перш ніж об'явити, що даний сусідній пристрій зник (Dead Time); IP-адресу інтерфейсу сусіднього маршрутизатора, до якого безпосередньо під'єднано даний маршрутизатор (Address); локальний інтерфейс, через який маршрутизатор встановив дані сусідські відношення (Interface). Є також модифікація цієї команди яка дозволяє отримати детальнішу інформацію про сусудів – `show ip ospf neighbor detail`.

Наприклад, для нашого випадку (рис. 4.32) таблиці сусідів будуть:
для маршрутизатора R1

```
Neighbor ID     Pri State     Dead Time Address         Interface
192.168.10.9    0   FULL/ -   00:00:35  192.168.10.2    Serial0/0/0
192.168.10.10   0   FULL/ -   00:00:35  192.168.10.6    Serial0/0/1;
```

для маршрутизатора R2

```
Neighbor ID     Pri State     Dead Time Address         Interface
192.168.10.5    0   FULL/ -   00:00:32  192.168.10.1    Serial0/0/0
192.168.10.10   0   FULL/ -   00:00:35  192.168.10.10   Serial0/0/1;
```

для маршрутизатора R3

```
Neighbor ID     Pri State     Dead Time Address         Interface
192.168.10.5    0   FULL/ -   00:00:36  192.168.10.5    Serial0/0/0
192.168.10.9    0   FULL/ -   00:00:35  192.168.10.9    Serial0/0/1,
```

що свідчить про те, що кожен маршрутизотор знаходиться в сусідських відношеннях з двома іншими маршрутизаторами мережі.

Зауважимо, що два маршрутизатора не можуть сформувати сусідских відношень, якщо [5, 15]: на інтерфейсах маршрутизаторів, що з'єднані; маски підмереж різні (це говорить про те, що ці інтерфейси знаходяться в окремих мережах); OSPF Hello інтервал і/або час життя (Dead Timers) на цих маршрутизаторах різні; типи мереж OSPF різні; під час настроювання протоколу OSPF виконано неправильні і/або некоректні команди.

Команда `show ip ospf database` – відображає вміст топологічної бази даних. Команда `clear ip route *` – очищує всю ТМ. Команда `clear ip route a.b.c.d` – вилучає з ТМ лише маршрут, задний адресою a.b.c.d. Команди групи `debug ip ospf` – виконують налагоджування операцій протоколу OSPF. Детальніше про команди тестування та пошуку помилок для протоколу OSPF можна ознайомитись, наприклад у [5, 15].

4.10 Контрольні запитання

1. Наведіть основні функції маршрутизаторів.
2. Поясніть, що називають маршрутизацією.
3. Наведіть класифікацію протоколів маршрутизації.
4. Поясніть, з якою метою використовується метрика маршруту і на основі яких параметрів вона обчислюється.
5. Наведіть та стисло охарактеризуйте види маршрутизації без таблиць.
6. Поясніть, що таке статична та динамічна маршрутизація.
7. Назвіть дві основні категорії алгоритмів динамічної маршрутизації. Наведіть приклади протоколів, що використовують дані алгоритми.
8. Поясніть, що таке автономна система.
9. Які функції покладаються на внутрішні та зовнішні шлюзи?
10. Поясніть різницю між внутрішніми і зовнішніми протоколами маршрутизації та наведіть приклади таких протоколів. Наведіть порівняльний аналіз цих видів маршрутизацій.
11. Наведіть порівняння протоколів динамічної маршрутизації.
12. Поясніть основні відмінності між алгоритмами маршрутизації DVA та LSA.
13. Наведіть команди настроювання статичного маршруту та маршруту за замовчуванням.
14. Поясніть, яку роль відіграє маршрут за замовчуванням. В яких випадках він використовується?
15. Покажіть на власному прикладі застосування команд статичної маршрутизації.
16. Наведіть команди пошуку та усунення помилок у настроюванні статичних маршрутів.
17. Охарактеризуйте протокол маршрутизації RIP. Наведіть його обмеження, переваги та недоліки.
18. Наведіть загальний алгоритм функціонування протоколу RIP.
19. Поясніть, як відбувається адаптація RIP-маршрутизаторів до змін станів мережі.
20. Наведіть причини виникнення петель маршрутизації.
21. Наведіть основні методи боротьби з фальшивими маршрутуми в протоколі RIP. Поясніть сутність цих методів.
22. Наведіть основні команди конфігурування протоколу RIP.
23. Поясніть що таке автосумаризація маршрутів. Які переваги та недоліки використання автосумаризації Ви знаєте?
24. Поясніть, з якою метою в протоколі RIP вимикають маршрутні оновлення через певні інтерфейси. Наведіть відповідну команду.
25. Поясніть застосування команди ip classless для протоколу RIP.

26. Які таймери використовуються у протоколі RIP та з якою метою? Наведіть команду змінення значень цих таймерів.

27. За допомогою якої команди у протоколі RIP можна встановити кількість паралельних маршрутів.

28. Поясніть, як налаштувати протокол RIP, щоб він анонсував статичні маршрути та маршрут за замовчуванням. Наведіть відповідні команди.

29. Наведіть кілька основних команд тестування та усунення помилок у роботі протоколу RIP та поясніть їх функції.

30. Наведіть загальну характеристику протоколу EIGRP, а також його переваги та недоліки.

31. Наведіть загальну формулу обчислення метрики протоколу EIGRP. Порівняйте її з метриками протоколів RIP та OSPF.

32. Наведіть основну термінологію протоколу EIGRP.

33. Які три таблиці використовуються у протоколі EIGRP? Наведіть призначення цих таблиць.

34. Поясніть, які маршрути є первинними, а які резервними. Чи до кожного пункту призначення існують такі маршрути? Відповідь обгрунтуйте.

35. Наведіть основні технології протоколу EIGRP.

36. Стисло охарактеризуйте та наведіть приклад роботи алгоритму DUAL.

37. Поясніть, як у протоколі EIGRP відбувається адаптування під зміни структури мережі.

38. Які типи пакетів протоколу EIGRP Ви знаєте? Поясніть призначення цих пакетів.

39. За рахунок чого час конвергенції для протоколу EIGRP достатньо малий?

40. Наведіть основні етапи конфігурування протоколу EIGRP з відповідними командами.

41. Поясніть, як виконується конфігурування смуги пропускання в гібридній багатоточковій мережі для протоколу EIGRP.

42. Поясніть, як виконується автоматичне узагальнення маршрутів у протоколі EIGRP. Наведіть команду, яка дозволяє вимкнути та увімкнути таке узагальнення.

43. Наведіть команди конфігурування аутентифікації у протоколі EIGRP.

44. Наведіть кілька команд тестування базової конфігурації протоколу EIGRP з відповідними поясненнями.

45. Наведіть загальну характеристику протоколу OSPF. Які переваги та недоліки протоколу OSPF Ви знаєте?

46. Наведіть термінологію протоколу OSPF.

47. Поясніть призначення топологічної бази даних та бази даних відношень суміжності протоколу OSPF.

48. Поясніть для яких типів мереж відбувається вибір маршрутизаторів DR та BDR у протоколі OSPF і яку роль вони виконуть.

49. На власному прикладі покажіть процес вибору маршрутизаторів DR та BDR.

50. Охарактеризуйте основні типи пакетів OSPF.

51. Дайте стислу характеристику станів інтерфейсів OSPF-маршрутизато-рів.

52. Наведіть основні команди тестування конфігурування протоколу OSPF та поясніть їх призначення.

4.11 Завдання

1. Враховуючи дані таблиці 4.15, а також те, що Net 1, Net 2, Net 3, Net 4 – є підмережами мережі Network 1, а Net 5, Net 6, Net 7 – Network 2 (рис. 4.33):

- визначте оптимальні маски (тобто маски з мінімально можливою кількістю двійкових нульових розрядів) для підмереж Net 1 – Net 7;
- для кожної підмережі наведіть її номер, діапазон IP-адрес та широкомовну IP-адресу.

2. Настройте протокол RIPv2 для мережі, наведеної на рис. 4.33 (схема IP-адресації відповідає завданню №1). Протокол RIPv2 повинен анонсувати задані нижче маршрут за замовчуванням та статичний маршрути. При цьому слід врахувати, що:

- маршрутизатор R2 повинен анонсувати лише мережі Net 1, Net 6;
- на маршрутизаторі R2 повинен бути прописаний маршрут за замовчуванням до маршрутизатора R1;
- на маршрутизаторі R3 повинен бути прописаний статичний маршрут до Loopback 1;
- на маршрутизаторі R1 протокол RIP запускати не треба.

3. Перевірте роботоздатність мережі, виконавши команду `auto-summary` на маршрутизаторах R2 – R4, а потім виконавши на них команду `no auto-summary`. Поясніть отримані результати.

4. Перевірте роботоздатність мережі, виконавши команду `no ip classless` на маршрутизаторах R2 – R4, а потім виконавши на них команду `no auto-summary`. Поясніть отримані результати.

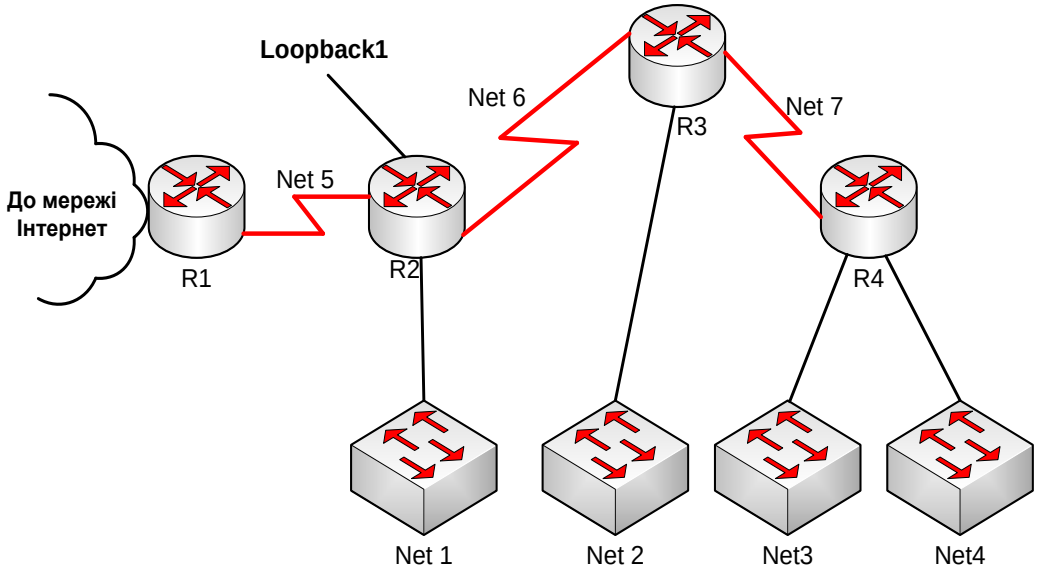


Рисунок 4.33 – Комп’ютерна мережа до завдань № 1, 2

Таблиця 4.15 – Варіанти до завдання №1

Номер вар.	Кількість вузлів у мережах				Адреси		
	Net 1	Net 2	Net 3	Net 4	Network 1	Network 2	Loopback1
1	2	3	4	5	6	7	8
1	120	60	16	10	220.80.15.0	10.0.0.0	15.0.0.1/24
2	15	85	23	18	195.110.10.0	172.16.0.0	201.12.3.1/24
3	10	100	4	8	197.47.83.0	192.168.1.0	45.0.0.1/24
4	45	60	15	30	206.61.44.0	172.17.0.0	10.0.0.1/24
5	12	10	58	20	198.40.10.0	10.0.0.0	192.168.3.1/24
6	50	54	30	10	196.76.32.0	192.168.4.0	45.0.0.1/24
7	100	40	13	10	202.62.23.0	192.168.10.0	172.12.34.1/24
8	120	60	14	6	213.14.5.0	172.30.0.0	192.168.3.1/24
9	40	30	14	14	200.165.13.0	10.0.0.0	10.57.32.1/24
10	20	40	13	10	199.24.23.0	172.23.0.0	192.168.7.1/24
11	28	70	20	14	203.100.15.0	10.0.0.0	10.3.1.1/24
12	16	32	8	5	212.10.100.0	192.168.32.0	202.12.34.1/24
13	100	27	54	4	194.200.15.0	172.14.0.0	59.10.3.1/24
14	60	30	2	16	198.31.13.0	10.0.0.0	14.0.0.1/24
15	12	10	58	32	205.141.67.0	172.31.0.0	192.168.9.1/24
16	57	33	55	2	197.26.123.0	192.168.25.0	23.56.17.1/24
17	115	45	4	10	193.14.95.0	172.27.0.0	192.168.8.1/24
18	50	2	14	23	210.3.2.0	10.0.0.0	192.168.1.1/24
19	10	50	37	17	193.4.50.0	192.168.4.0	10.0.0.1/24
20	125	37	24	12	196.87.105.0	192.168.51.0	172.12.34.1/24
21	10	60	50	17	215.56.18.0	172.24.0.0	38.34.23.1/24
22	8	4	28	70	214.24.15.0	10.0.0.0	43.15.2.1/24
23	6	64	16	32	203.28.115.0	172.20.0.0	212.24.13.1/24
24	16	4	100	27	202.16.13.0	10.0.0.0	93.17.48.1/24
25	14	20	70	30	208.3.14.0	192.168.35.0	197.26.12.0/24
26	36	33	5	9	194.6.13.0	192.168.22.0	39.56.107.1/24

Продовження таблиці 4.15

1	2	3	4	5	6	7	8
27	90	30	8	20	194.14.95.0	172.27.0.0	192.168.7.1/24
28	35	12	14	23	200.65.1.0	10.0.0.0	192.168.6.1/24
29	78	15	37	15	193.28.50.0	192.168.4.0	12.0.0.1/24
30	70	30	40	10	196.74.15.0	192.168.51.0	172.12.78.1/24

5. Настройте протокол OSPF на маршрутизаторах R2 – R6 мережі, наведеної на рис. 4.34, попередньо виконавши визначення IP-адрес для кожного її сегмента згідно з даними, наведеними у табл. 4.16 (маски підмереж повинні бути оптимальними). При цьому для мереж Net 1 – Net 9 використайте IP-адреси мережі 10.0.0.0/8, для мереж Net 10 – Net 12 – 172.16.0.0/16, а для Net 13, Net 14 – 193.18.24.0/24. Також врахуйте, що:

- маршрутизатор R2 повинен анонсувати лише мережі Net 10 -Net 12;
- на маршрутизаторі R2 повинен бути прописаний шлях за замовчуванням до маршрутизатора R1;
- на маршрутизаторі R2 повинен бути прописаний статичний маршрут до Loopback 1, що піднятий на маршрутизаторі R7. Адреса для Loopback 1: 192.168.3.15/24.

6. Настройте аутентифікацію на кожному маршрутизаторі мережі.

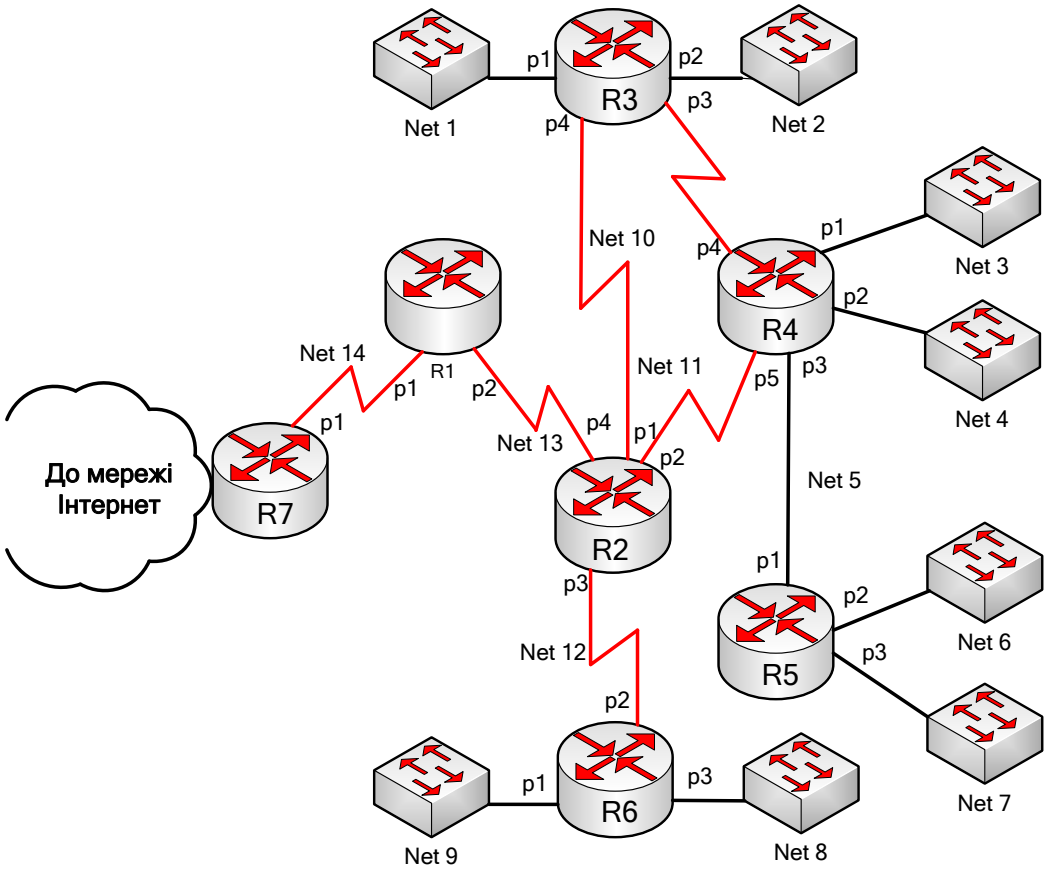


Рисунок 4.34 – Структура комп’ютерної мережі завдання № 3

Таблиця 4.16 – Варіанти до завдання № 3

Номер вар.	Кількість вузлів у мережах								
	Net 1	Net 2	Net 3	Net 4	Net 5	Net 6	Net 7	Net 8	Net 9
1	120	60	16	10	130	60	64	80	70
2	15	85	23	18	128	10	58	20	28
3	125	37	24	12	50	54	30	10	16
4	10	60	50	17	256	40	13	10	100
5	8	4	28	70	120	64	14	6	70
6	6	64	16	32	40	30	14	14	5
7	16	4	100	27	20	40	13	10	8
8	14	20	70	30	28	70	20	14	14
9	36	33	5	9	16	32	8	5	37
10	90	30	8	20	100	27	54	4	40
11	35	12	14	23	60	30	2	16	24
12	78	256	37	15	128	10	58	32	50
13	70	30	40	10	57	33	55	2	28
14	12	125	24	37	115	45	4	10	16
15	17	10	50	60	128	2	16	23	100
16	70	8	28	4	10	50	37	17	70
17	32	6	16	64	125	37	24	12	37
18	27	16	100	4	10	60	50	17	40
19	30	14	70	20	8	4	28	70	24
20	9	36	5	128	6	64	16	32	50
21	20	90	8	30	16	4	100	27	28
22	23	35	14	12	14	20	70	30	16
23	15	78	37	15	36	33	5	9	100
24	10	70	40	30	264	30	8	20	70
25	115	45	4	10	35	12	14	23	5
26	50	2	14	23	78	15	37	15	100
27	10	50	37	17	70	30	40	10	70
28	125	37	24	12	100	28	150	200	5
29	10	60	50	17	60	16	256	128	8
30	8	4	28	70	12	100	200	264	14

7. Вкажіть оптимальний маршрут для протоколу OSPF з точки зору маршрутизатора RS до мережі DN (рис. 4.35). Обчисліть значення метрики цього маршруту. Варіанти завдань наведено у табл. 4.17.

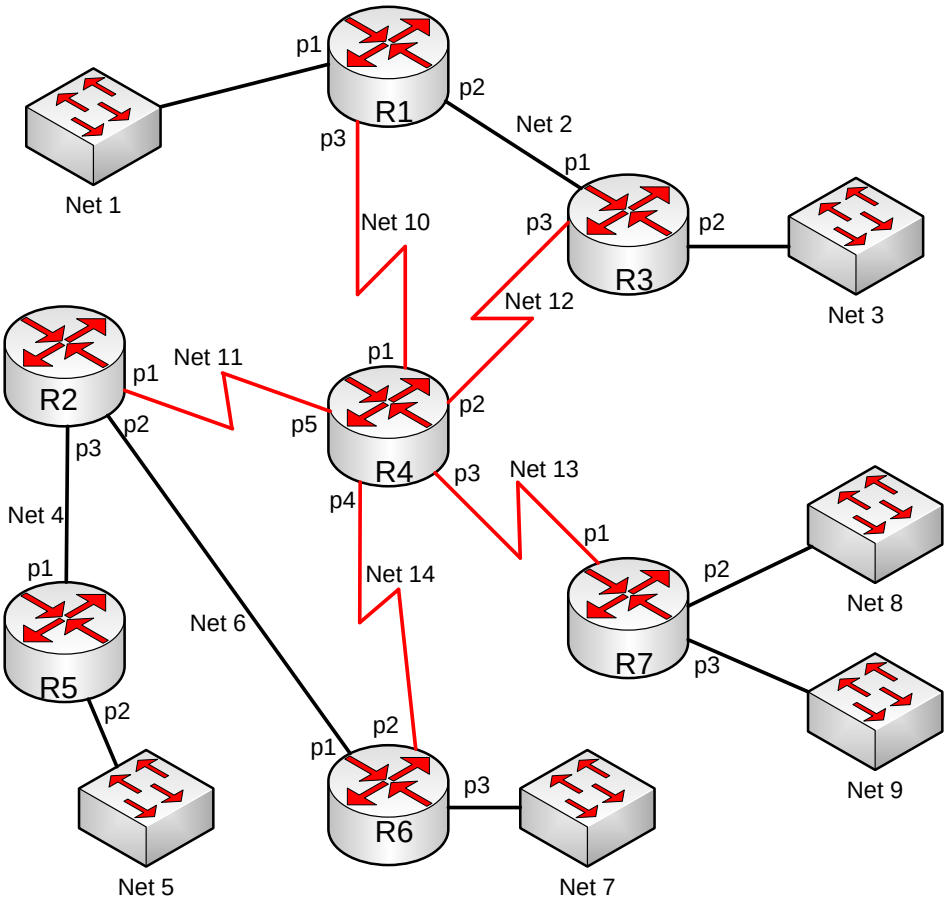


Рисунок 4.35 – Комп’ютерна мережа для завдань № 4 та № 6

Таблиця 4.17 – Варіанти до завдання № 4

Номер вар.	RS	DN	Номер варіанта з		Номер вар.	RS	DN	Номер варіанта з	
			табл. 4.18	табл. 4.19				табл. 4.18	табл. 4.19
1	R1	Net 7	1	1	16	R1	Net 7	4	3
2	R1	Net 5	2	1	17	R1	Net 5	5	3
3	R1	Net 4	3	1	18	R1	Net 4	6	3
4	R1	Net 8	4	1	19	R1	Net 8	1	4
5	R2	Net 1	5	1	20	R2	Net 1	2	4
6	R2	Net 3	6	1	21	R2	Net 3	3	4
7	R2	Net 7	1	2	22	R2	Net 7	4	4
8	R2	Net 8	2	2	23	R2	Net 8	5	4
9	R2	Net 4	3	2	24	R2	Net 4	6	4
10	R2	Net 8	4	2	25	R2	Net 8	1	5
11	R3	Net 7	5	2	26	R3	Net 7	2	5
12	R3	Net 5	6	2	27	R3	Net 5	3	5
13	R7	Net 1	1	3	28	R7	Net 1	4	5
14	R7	Net 3	2	3	29	R7	Net 3	5	5
15	R7	Net 5	3	3	30	R7	Net 5	6	5

Таблиця 4.18 – Значення пропускної здатності послідовних каналів

Номер варіанта	Значення пропускної здатності мережі (Кбіт/с)				
	Net 10	Net 11	Net 12	Net 13	Net 14
1	1024	128	1024	1024	512
2	256	512	1544	64	128
3	512	1024	1544	512	256
4	64	128	512	128	1024
5	128	64	256	256	64
6	1024	512	512	1544	1024

Таблиця 4.19 – Значення пропускної здатності каналів LAN

Номер вар.	Значення пропускної здатності мережі (Кбіт/с)								
	Net 1	Net 2	Net 3	Net 4	Net 5	Net 6	Net 7	Net 8	Net 9
1	100	100	1000	10	10	100	10	100	100
2	1000	10	100	100	10	100	10	100	100
3	10	1000	10	1000	100	100	100	100	100
4	1000	100	100	100	1000	10	1000	100	100
5	1000	100	100	100	100	100	1000	10	100

8. Настройте на маршрутизаторах R2 – R6 мережі, наведеної на рис. 4.36 протокол EIGRP, попередньо виконавши визначення IP-адрес для кожного її сегмента згідно з даними, наведеними у табл. 4.20 (маски підмереж повинні бути оптимальними). При цьому для мереж Net 1 – Net 7 використайте IP-адреси мережі 172.16.0.0/16, для мереж Net 8 – Net 11 – 10.0.0.0/8, а для Net 12, Net 13 – 202.180.4.0/24. Також врахуйте, що:

- маршрутизатор R2 повинен анонсувати лише мережі Net 8 – Net 11;
- на маршрутизаторі R2 повинен бути прописаний шлях за замовчуванням до маршрутизатора R1;
- на маршрутизаторі R2 повинен бути прописаний статичний маршрут до Loopback 1, що піднятий на маршрутизаторі R7. Адреса для Loopback 1: 192.168.3.15/24.

Поясніть, як буде працювати алгоритм EIGRP у нашій мережі, якщо на кожному EIGRP-маршрутизаторі виконати команду:

- auto-summary;
- no auto-summary.

9. Настройте аутентифікацію на всіх EIGRP-маршрутизаторах.

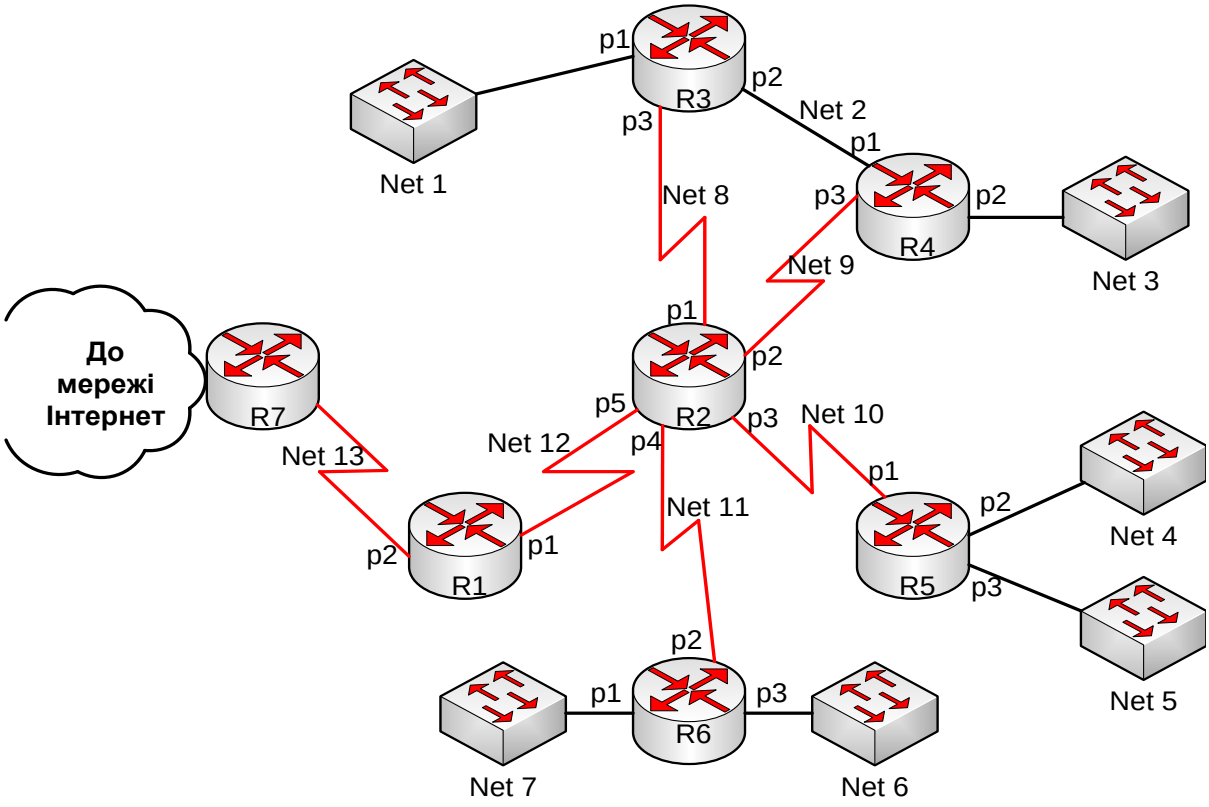


Рисунок 4.36 – Комп’ютерна мережа до завдання № 5

Таблиця 4.20 – Варіанти до завдання №5

Номер варіанта	Кількість вузлів у мережах						
	Net 1	Net 2	Net 3	Net 4	Net 5	Net 6	Net 7
1	2	3	4	5	6	7	8
1	17	70	28	37	50	4	10
2	12	37	16	24	37	64	125
3	17	40	100	50	60	4	10
4	70	24	70	28	4	20	8
5	32	50	5	16	64	128	6
6	27	28	8	100	4	30	16
7	30	16	14	70	20	12	14
8	9	100	37	5	33	15	36
9	20	70	40	8	30	30	264
10	23	5	4	14	12	10	35
11	15	100	14	37	15	23	78
12	10	70	37	40	30	17	70
13	200	5	24	150	28	12	100
14	128	8	50	256	16	17	60
15	264	14	28	200	100	70	12
16	80	70	16	64	60	10	130
17	20	28	23	58	10	18	128
18	10	16	24	30	54	12	50
19	10	100	50	13	40	17	256
20	6	70	28	14	64	70	120
21	14	5	16	14	30	32	40

Продовження таблиці 4.20

1	2	3	4	5	6	7	8
22	10	8	100	13	40	27	20
23	14	14	70	20	70	30	28
24	5	37	5	8	32	9	16
25	4	40	8	54	27	20	100
26	16	24	14	2	30	23	60
27	32	50	37	58	10	15	128
28	2	28	40	55	33	10	57
29	10	16	24	4	45	37	115
30	23	100	50	16	2	60	128

10. Вкажіть оптимальний маршрут для протоколу EIGRP з точки зору маршрутизатора RS до мережі DN (рис. 4.35). Обчисліть значення його метрики. Варіанти завдань наведено у табл. 4.21. Час затримки каналів зв'язку в залежності від їх пропускної здатності наведено у табл. 4.24.

Таблиця 4.21 – Варіанти до завдання № 6

Номер вар.	RS	DN	Номер варіанта з		Номер вар.	RS	DN	Номер варіанта з	
			табл. 4.22	табл. 4.23				табл. 4.22	табл. 4.23
1	R1	Net 7	1	1	16	R1	Net 7	4	3
2	R1	Net 5	2	1	17	R1	Net 5	5	3
3	R1	Net 4	3	1	18	R1	Net 4	6	3
4	R1	Net 8	4	1	19	R1	Net 8	1	4
5	R2	Net 1	5	1	20	R2	Net 1	2	4
6	R2	Net 3	6	1	21	R2	Net 3	3	4
7	R2	Net 7	1	2	22	R2	Net 7	4	4
8	R2	Net 8	2	2	23	R2	Net 8	5	4
9	R2	Net 4	3	2	24	R2	Net 4	6	4
10	R2	Net 8	4	2	25	R2	Net 8	1	5
11	R3	Net 7	5	2	26	R3	Net 7	2	5
12	R3	Net 5	6	2	27	R3	Net 5	3	5
13	R7	Net 1	1	3	28	R7	Net 1	4	5
14	R7	Net 3	2	3	29	R7	Net 3	5	5
15	R7	Net 5	3	3	30	R7	Net 5	6	5

Таблиця 4.22 – Значення пропускної здатності послідовних каналів

Номер варіанта	Значення пропускної здатності мережі (Кбіт/с)				
1	2				
	Net 10	Net 11	Net 12	Net 13	Net 14
1	64	1024	1024	64	256

Продовження таблиці 4.22

1	2				
	Net 10	Net 11	Net 12	Net 13	Net 14
2	128	256	1544	512	512
3	256	512	512	256	256
4	128	1024	64	1544	64
5	1024	1544	1024	1024	512
6	64	256	64	64	128

Таблиця 4.23 – Значення пропускної здатності каналів LAN

Номер варіанта	Значення пропускної здатності мережі (Кбіт/с)								
	Net 1	Net 2	Net 3	Net 4	Net 5	Net 6	Net 7	Net 8	Net 9
1	1000	10	100	10	10	100	100	100	1000
2	100	10	100	100	10	100	100	100	1000
3	100	100	100	1000	100	100	1000	100	1000
4	100	1000	10	100	1000	10	100	100	1000
5	100	100	100	100	100	100	100	10	1000

Таблиця 4.24 – Значення затримки каналів зв’язку

Значення пропускної здатності (Кбіт/с)	Значення затримки (мкс)
64	20000
128	20000
256	20000
512	20000
1024	20000
1544	20000
10000	1000
100000	100
1000000	10

11. На прикладі комп’ютерної мережі, наведеної на рис. 4.37 дослідіть процес вибору DR- та BDR- маршрутизаторів. Для цього виконайте такі кроки:

- на всіх маршрутизаторах настройте IP-адреси портів, що під’єднані до центрального комутатора (адреса мережі, якій належать порти 192.168.10.0/26). На кожному маршрутизаторі настройте протокол OSPF. Поясніть, який з маршрутизаторів став DR, а який BDR. Перезавантажте мережу та поясніть результати вибору DR та BDR. Вимкніть DR-маршрутизатор. Поясніть, які тепер маршрутизатори стали DR та BDR? Увімкніть вимкнений маршрутизатор і знову визначте DR- та BDR- маршрутизатори.

- на кожному маршрутизаторі настройте Loopback з довільними IP-адресами і № 7перезавантежте мережу. Поясніть результати вибору DR- та BDR- маршрутизаторів.
- виконайте конфігурування маршрутизаторів так, щоб маршрутизатор R2 став DR, а R6 – BDR. Для цього скористайтесь командою `ip ospf priority`.

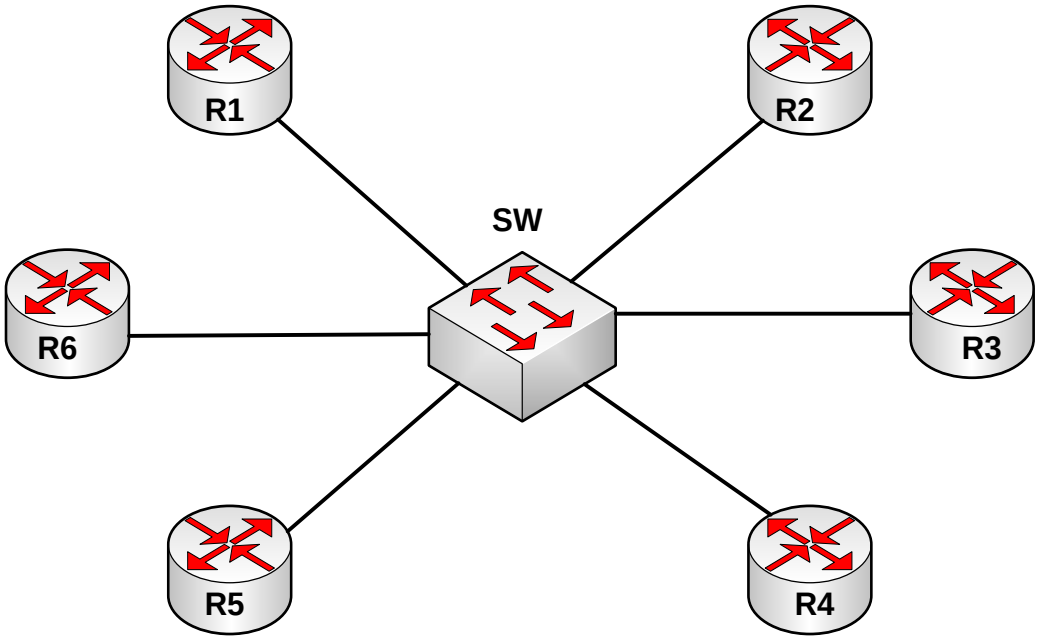


Рисунок 4.37 – Структура мережі до завдання №7

Поясніть, з якою метою мережевий адміністратор впливає на процес вибору DR- та BDR-маршрутизаторів.

Словник часто вживаних термінів

Access Network – мережа доступу
Address Resolution Protocol (ARP) – протокол дозволу адрес
Application layer – рівень застосувань
Attenuation – згасання
Autonomous systems (AS) – автономна система
Availability – готовність (коефіцієнт готовності)
Backbone – магістральна мережа
Back-end – додаток-сервер
Backup Designated Router (BDR) – резервний призначений маршрутизатор
Bandwidth – смуга пропускання
Basic Input/Output System (BIOS) – базова системи введення-виведення
Bipolar Alternate Mark Inversion (AMI) – біполярне кодування з альтернативною інверсією
Bit Error Rate (BER) – інтенсивність бітових помилок
Border Gateway Protocol (BGP) – протокол автоматичної побудови маршрутів
Bridge – міст
Bridge Protocol Data Unit (BPDU) – протокольні одиниці даних моста
Broadcast address – широкомовна адреса
Broadcast domain – широкомовний домен
Broadcast storm – широкомовний шторм
Central Processing Unit (CPU) – центральний процесор
Circuit switching – комутація каналів
Collision domain – колізійний домен
Command Line Interface (CLI) – інтерфейс командного рядка
Committed information rate (CIR) – гарантована смуга пропускання віртуального каналу
Configuration Register – конфігураційний регістр
Connectionless protocol – протокол без попереднього встановлення з’єднання
Connection-oriented protocol – протокол із встановленням з'єднання
Cyclic Redundancy Check (CRC) – циклічний надлишковий контроль
Cut-through switching – наскрізна комутація
Data Circuit terminating Equipment (DCE) – апаратура передавання даних
Data Link layer – канальний рівень
Data Terminal Equipment (DTE) – кінцеве інформаційне обладнання

Dedicated Server Network – мережа на основі виділеного сервера
Dense Wave Division Multiplexing (DWDM) – технологія спектрального мультиплексування
Designated port – призначений порт
Designated Router (DR) – призначений маршрутизатор
Diffusing Update Algorithm (DUAL) – алгоритм дифузійного оновлення
Distance Vector Algorithm (DVA) – дистанційно-векторний алгоритм
Domain Name Service (DNS) – доменна система імен
Internetwork Operation System (IOS) – міжмережева операційна система
International Organization for Standardization (ISO) – міжнародна організація зі стандартизації
Internet Engineering Task Force (IETF**)** – служба розробки стандартів Інтернету
Internet protocol (IP) – протокол Інтернету
Internet Service Provider (ISP) – постачальник послуг мережі Інтернет
Enhanced Interior Gateway Routing Protocol (EIGRP) – розширений протокол внутрішнього шлюзу
Extensibility – розширюваність
Far End Cross Talk (FEXT) – перехресні наведення на дальньому кінці
Fault tolerance – відмовостійкість
Filtering – фільтрування
Fixed Length Subnet Masking (FLSM) – сегментація мережі із застосуванням масок постійної довжини
Flash – енергонезалежна пам’ять
Flooding – лавинне передавання
Forwarding – перенаправлення
Fragment free switching – комутація з контролем фрагментів
Frequency Division Multiplexing (FDM) – частотне мультиплексування
Front-end – додаток-клієнт
Graphical User Interface (GUI) – користувацький графічний інтерфейс
Hub – концентратор (хаб)
Laser Diode – лазерний діод
Learning – навчання
Light Emmited Diode – світлодіод
Limited broadcast – обмежене широкомовне повідомлення
Link State Algorithm (LSA) – алгоритм на основі стану каналів
Local Area Network (LAN) – локальна комп’ютерна мережа
Media Access Control (MAC) – керування доступом до середовища передавання

Metropolitan Area Network (MAN) – міська комп’ютерна мережа
Multi Mode Fiber (MMF) – багатомодовий кабель
Multicast address – групова адреса
Near End Cross Talk (NEXT) – перехресні наведення на ближньому кінці
Neighbor table – таблиця сусідніх пристроїв
Network Interface Card (NIC) – плати мережевого інтерфейсу
Network layer – мережевий рівень
Non Return to Zero (NRZ) – кодування без повернення до нуля
Non Return to Zero with ones Inverted (NRZI) – потенціальний код з інверсією при одиниці
Non-volatile random access memory (NVRAM) – енергонезалежна пам’ять
Open Shortest Path First (OSPF) – відкритий протокол першочергового відкриття найкоротших маршрутів
Optical fiber – волоконно-оптичний кабель
Packet switching – комутація пакетів
Peer-to-Peer Network – однорангова мережа
Physical layer – фізичний рівень
Plesiochronous Digital Hierarchy (PDH) – технологія плезіохронної цифрової ієрархії
Presentation layer – представницький рівень
Privileged Executive Mode – привілейований EXEC-режим
Protocol Data Unit (PDU) – протокольний блок даних
Pulse Amplitude Modulation (PAM) – імпульсно-кодова модуляція
Random-Access Memory (RAM) – оперативна пам’ять
Repeater – повторювач
Read Only Memory (ROM) – пам’ять лише для читання
Reliable Transport Protocol (RTP) – надійний транспортний протокол
Reverse Address Resolution Protocol (RARP) – реверсивний протокол перетворення IP-адрес
Root Bridge – кореневий міст
Root Port – кореневий порт
Routed protocol – мережевий протокол
Router – маршрутизатор
Routing Information Protocol (RIP) – протокол маршрутної інформації
Routing table – таблиця маршрутизації
Routing protocol – протокол маршрутизації
Running Config – робочий файл конфігурації
Scabbility – масштабованість

Secure Shell (SSH) – безпечна оболонка
Security – безпека
Session layer – сеансовий рівень
Shielded Foiled twisted pair (SFTP) – фольговано-екранована скручена пара проводів
Shielded Twisted Pair (STP) – екранована скручена пара проводів
Single Mode Fiber (SMF) – одномодовий кабель
Sliding window technology – технологія ковзного вікна
Spanning Tree Protocol – протокол зв'язуючого дерева
Startup Config – стартовий файл конфігурації
Store-and-Forward switching – комутація з проміжним зберіганням
Subnet – підмережа
Switch – комутатор (свіч)
Throughput – пропускна здатність
Trivial File Transfer Protocol (TFTP) – простий протокол передавання файлів
Topology table – топологічна таблиця
Transparency – прозорість
Transport layer – транспортний рівень
Twisted pair – скручена пара проводів
Unshielded Twistedpair (UTP) – неекранована скручена пара проводів
Unicast address – адреса конкретного пристрою
User Executive Mode – користувацький EXEC-режим
Variable Length Subnet Masking (VLSM) – сегментація мережі із застосуванням масок змінної довжини
Virtual circuit (VC) – віртуальний канал
Wide Area Network (WAN) – глобальна комп'ютерна мережа

ЛІТЕРАТУРА

1. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов. [4-е изд.] / В. Г. Олифер, Н. А. Олифер – СПб. : Питер, 2010. – 944 с.
2. Кулаков Ю. О. Комп'ютерні мережі : підручник. / Ю. О. Кулаков, Г. М. Луцький. / за ред. Ю. С. Ковтанюка. – К. : Видавництво „Юніор", 2005. – 400 с.
3. Буров Є. Комп'ютерні мережі. [2-е вид., оновл. і допов.] / Буров Є. – Львів : БаК, 2003. – 584 с.
4. Рональд Бодчер. Программа сетевой академии Cisco CCNA [3-е изд.] : [пер. с англ.] / Рональд Бодчер, К. Р. Киркендаль. – М. : изд. Дом "Вильямс", 2005. – 1186 с.
5. Рональд Бодчер. Программа сетевой академии Cisco CCNA 3 и 4. [3-е изд.] : [пер. с англ.] / Рональд Бодчер, К. Р. Киркендаль. – М. : изд. Дом "Вильямс", 2007. – 944 с.
6. Таненбаум Э. Компьютерные сети. [4-е изд.] : [пер. с англ.] / Таненбаум Э. – СПб. : Питер, 2003. – 992 с.
7. Столлингс В. Компьютерные системы передачи данных. [7-е изд.] : [пер. с англ.] / Столлингс В. – М. : изд. дом "Вильямс", 2003. – 720 с.
8. Спортук М. Компьютерные сети и сетевые технологии. / М. Спортук, Ф. Папас – ООО „ТИД ДС", 2002. – 736 с.
9. Шиндер Д. Основы компьютерных сетей : [пер. с англ.] / Шиндер Д. – М. : изд. дом "Вильямс", 2002. – 656 с.
10. Кульгин М. В. Компьютерные сети. Практика построения. Для профессионалов. [2-е изд.] / Кульгин М. В. – СПб. : Питер, 2003. – 462 с.
11. Столлингс В. Современные компьютерные сети. [2-е изд.] / Столлингс В. – СПб. : Питер, 2003. – 783 с.
12. Закер К. Компьютерные сети. Модернизация и поиск неисправностей / Закер К. : [пер. с англ.] – СПб. : Питер, 2003. – 1008 с.
13. Камер Д. Компьютерные сети и Internet. Разработка приложений для Internet / Камер Д. : пер. с англ. – М. : изд. дом "Вильямс", 2002. – 640 с.
14. Кларк К. Принципы коммутации в локальних сетях CISCO / К. Кларк, К. Гамильтон : [пер. с англ.] – М. : изд. дом "Вильямс", 2003. – 976 с.
15. Пакет К. Создание масштабируемых сетей CISCO / К. Пакет, Д. Тир : [пер. с англ.] – М.: Изд. дом "Вильямс", 2002. – 792 с.
16. Хилл Б. Полный справочник по CISCO / Хилл Б. : [пер. с англ.] – М. : изд. дом "Вильямс", 2004. – 1088 с.

Навчальне видання

Арсенюк Ігор Ростиславович
Яровий Андрій Анатолійович

КОМП'ЮТЕРНІ МЕРЕЖІ

Частина 2

Навчальний посібник

Редактор О. Скалоцька

Оригінал-макет підготовлено І. Арсенюком

Підписано до друку
Формат 29,7×42¼. Папір офсетний.
Гарнітура Times New Roman.
Друк різографічний. Ум. друк. арк.
Наклад прим. Зам. №

Вінницький національний технічний університет,
навчально-методичний відділ ВНТУ.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, к. 2201.
Тел. (0432) 59-87-36.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.

Віддруковано у Вінницькому національному технічному університеті
в комп'ютерному інформаційно-видавничому центрі.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, ГНК, к. 114.
Тел. (0432) 59-87-38.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.