

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

КОМП'ЮТЕРНІ МЕРЕЖІ

НАВЧАЛЬНИЙ ПОСІБНИК

Житомир
2025

УДК 004.7
К63

*Рекомендовано до друку Вченою радою
Черкаського державного технологічного університету
(протокол № 7 від 17 березня 2025 року)*

*Рекомендовано до друку Вченою радою
Державного університету «Житомирська політехніка»
(протокол № 6 від 24 березня 2025 року)*

Автори:

Чепинога Анатолій Володимирович,
Єфіменко Андрій Анатолійович,
Рудаков Костянтин Сергійович,
Лавданський Артем Олександрович,
Ланських Євген Володимирович,
Фауре Еміль Віталійович

Рецензенти:

М. О. Євдокименко, д-р техн. наук, професор, професор кафедри інфокомунікаційної інженерії ім. В. В. Поповського, Харківський національний університет радіоелектроніки
С. В. Заболотній, д-р техн. наук, професор, професор кафедри комп'ютерної інженерії та інформаційних технологій, Черкаський державний бізнес-коледж
В. Д. Голь, канд. техн. наук, професор, завідувач спеціальної кафедри №3, Інститут спеціального зв'язку та захисту інформації НТУУ «КПІ ім. Ігоря Сікорського»
В. В. Воротніков, д-р техн. наук, професор, професор кафедри комп'ютерної інженерії та кібербезпеки, Державний університет «Житомирська політехніка»

Комп'ютерні мережі: навч. посіб. [Електронний ресурс] /
К63 **А. В. Чепинога, А. А. Єфіменко, К. С. Рудаков, А. О. Лавданський, Є. В. Ланських, Е. В. Фауре ;** М-во освіти і науки України, Черкас. держ. технол. ун-т, Державний університет «Житомирська політехніка». – Житомир : Державний університет «Житомирська політехніка», 2025. – 386 с.

ISBN 978-966-683-695-6

Навчальний посібник охоплює базові теоретичні засади основних технологій побудови комп'ютерних мереж, їх безпеки та налаштування на прикладі рішень та обладнання Cisco. Окремо подано принципи побудови комутаторів, маршрутизаторів та контролерів бездротових мереж. Приділено особливу увагу будові і функціонуванню мережної операційної системи Cisco IOS. Практикум містить матеріали до виконання лабораторних робіт з детальним описом технічних рішень та налаштувань. Видання спрямоване на формування у здобувачів вищої освіти комплексних навичок побудови, налаштування та захисту комп'ютерних мереж.

УДК 004.7

ISBN 978-966-683-695-6

© А. В. Чепинога, А. А. Єфіменко, К. С. Рудаков,
А. О. Лавданський, Є. В. Ланських, Е. В. Фауре, 2025

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 СУЧАСНІ ТЕХНОЛОГІЇ МЕРЕЖНОГО ЗВ'ЯЗКУ	7
1.1 ПРОТОКОЛИ ТА МОДЕЛІ	12
1.2 ФІЗИЧНИЙ РІВЕНЬ	16
1.3 КАНАЛЬНИЙ РІВЕНЬ. КОМУТАЦІЯ ETHERNET	26
1.4 МЕРЕЖНИЙ РІВЕНЬ	33
1.4.1 Протокол ARP	36
1.4.2 Адресація IPv4	39
1.4.3 Адресація IPv6	43
1.4.4 Протокол ICMP	47
1.5 ТРАНСПОРТНИЙ РІВЕНЬ.....	50
1.6 ПРИКЛАДНИЙ РІВЕНЬ	57
1.7 ОСНОВИ МЕРЕЖНОЇ БЕЗПЕКИ.....	61
РОЗДІЛ 2 ТЕХНОЛОГІЇ УПРАВЛІННЯ ТА СТРУКТУРИЗАЦІЯ МЕРЕЖ..	67
2.1 ТЕХНОЛОГІЯ VLAN.....	67
2.1.1 Вступ до VLAN.....	67
2.1.2 Типи VLAN	68
2.1.3 Архітектура VLAN	69
2.1.4 Протокол DTP.....	71
2.1.5 Пересилання даних між VLAN	72
2.2 ПРОТОКОЛ STP.....	75
2.2.1 Призначення STP.....	75
2.2.2 Принципи роботи STP	77
2.2.3 Протокол RSTP	81
2.3 АГРЕГАЦІЯ КАНАЛІВ	81
2.3.1 Принципи агрегації каналів	81
2.3.2 EtherChannel	82
2.1.3 Протокол агрегації портів	83
2.1.4 Протокол керування агрегацією з'єднань	84
2.4 АВТОМАТИЗАЦІЯ НАЛАШТУВАННЯ МЕРЕЖНИХ ПАРАМЕТРІВ	86
2.4.1 DHCPv4	86
2.4.2 Автоконфігурація IPv6	90
2.5 ПРОТОКОЛИ РЕЗЕРВУВАННЯ ОСНОВНОГО ШЛЮЗУ.....	94
2.5.1 Огляд технології резервування основного шлюзу (FHRP).....	94
2.5.2 Протокол гарячого резервування маршрутизації (HSRP).....	94
2.5.3 Протокол резервування віртуального маршрутизатора (VRRP).....	95
2.5.4 Протокол балансування навантаження шлюзу (GLBP)	96
2.6 ТЕХНОЛОГІЇ ПОБУДОВИ ЗАХИСТУ LAN.....	97
2.6.1 Запобігання атакам переповнення CAM-таблиць	97
2.6.2 Атаки VLAN-переходів	99
2.6.3 Атаки маніпуляції з STP	101
2.6.4 Атаки з підміною MAC-адрес	102
2.6.4 Атаки виснаження DHCP	104

2.7 БЕЗДРОТОВА МЕРЕЖЕВА ІНФРАСТРУКТУРА	105
2.7.1 Огляд технології бездротового зв'язку	105
2.7.2 Основні компоненти бездротових мереж	107
2.7.3 Принципи функціонування бездротових мереж	108
2.7.4 Протокол керування точками доступу <i>CAPWAP</i>	112
2.7.5 Частотний ресурс каналів	113
2.8 ПРИНЦИПИ МАРШРУТИЗАЦІЇ	115
2.8.1 Огляд основних функцій маршрутизації	115
2.8.2 Таблиця маршрутизації	116
2.8.3 Показники маршрутизації	118
2.8.4 Протоколи динамічної маршрутизації	118
2.8.5 Статична маршрутизація	119
РОЗДІЛ 3 ЛАБОРАТОРНЕ ОБЛАДНАННЯ	123
3.1 КОМУТАТОРИ <i>CISCO CATALYST</i>	123
3.1.1 Інтерфейси/порти комутаторів	126
3.1.2 Програмне забезпечення комутаторів	126
3.2 МАРШРУТИЗАТОРИ <i>CISCO ISR</i>	127
3.2.1 Інтерфейси маршрутизаторів <i>Cisco</i>	130
3.2.2 Процес завантаження маршрутизатора	133
3.2.3 Типи під'єднання до обладнання	135
3.3 КОНТРОЛЕРИ БЕЗДРОТОВОЇ МЕРЕЖІ <i>CISCO</i>	137
3.3.1 Призначення та структура контролера	137
3.3.2 Інтерфейси контролера	139
РОЗДІЛ 4 МЕРЕЖНА ОПЕРАЦІЙНА СИСТЕМА <i>CISCO IOS</i>	143
4.1 РЕЖИМИ РОБОТИ ПРИСТРОЇВ ПІД КЕРУВАННЯМ <i>CISCO IOS</i>	143
4.2 ЛОГІЧНІ І ФІЗИЧНІ ІНТЕРФЕЙСИ МАРШРУТИЗАТОРА <i>CISCO</i>	146
4.3 КОМАНДИ БАЗОВОГО НАЛАШТУВАННЯ КЕРОВАНОГО КОМУТАТОРА <i>CISCO</i>	148
4.4 ОСНОВНІ КОМАНДИ ДІАГНОСТИКИ ПАРАМЕТРІВ РОБОТИ КОМУТАТОРА <i>CISCO</i>	149
4.5 ПОРЯДОК НАЛАШТУВАННЯ ІНТЕРФЕЙСІВ МАРШРУТИЗАТОРА <i>CISCO</i>	149
4.6 ОСНОВНІ КОМАНДИ НАЛАГОДЖЕННЯ ПАРАМЕТРІВ ІНТЕРФЕЙСІВ МАРШРУТИЗАТОРА	150
4.7 ОСНОВНІ КОМАНДИ ДЛЯ БАЗОВОЇ ДІАГНОСТИКИ РОБОТИ МАРШРУТИЗАТОРА <i>CISCO</i>	154
РОЗДІЛ 5 ПРАКТИКУМ	155
5.1 ПРАКТИЧНИЙ МОДУЛЬ 1	155
5.1.1 Базова навігація в <i>IOS</i> . Базові налаштування	155
5.1.2 Налаштування <i>OC Windows</i> та <i>OC Linux</i>	179
5.1.3 Фізична та логічна адресація вузлів комп'ютерних мереж	199
5.1.4 IP адресація	205
5.1.5 Відповідність логічних <i>IPv4</i> адрес фізичним	212
5.1.6 Дослідження <i>PDU</i> пакетів	228

5.1.7 Засоби віддаленого доступу та адміністрування	245
5.2 ПРАКТИЧНИЙ МОДУЛЬ 2.....	273
5.2.1 Налаштування <i>VLAN</i> та транкових каналів	273
5.2.2 Налаштування маршрутизації між <i>VLAN</i>	285
5.2.3 Налаштування <i>Etherchannel</i>	295
5.2.4 Налаштування сервера <i>DHCPv4</i>	305
5.2.5 Налаштування резервних маршрутизаторів <i>HSRP</i>	321
5.2.6 Налаштування безпеки засобами комутатора	331
5.2.7 Налаштування <i>WLAN</i> із застосуванням <i>WLC</i>	341
5.2.8 Базові налаштування маршрутизатора та статичних маршрутів ..	364
СПИСОК ЛІТЕРАТУРИ.....	384

ВСТУП

Опанування навичок роботи з комп'ютерними мережами є критично важливим для здобувачів вищої освіти технічних напрямків, оскільки сучасні інформаційні технології та інженерні рішення значною мірою базуються на мережевій взаємодії. В зв'язку з цим вивчення принципів побудови, функціонування, налаштування та пошуку несправностей в комп'ютерних мережах відіграє важливу роль при формуванні цілісного пулу компетенцій фахівця.

Пропонований навчальний посібник розроблено з метою всебічного охоплення як теоретичної компоненти так і практичної підготовки здобувачів для роботи з основними технологіями і обладнанням комп'ютерних мереж, функціонуючими в них протоколами та базовими моделями і стандартами. Матеріал структуровано таким чином, щоб забезпечити поетапне засвоєння знань – від основних концепцій до практичної реалізації складних мережових рішень.

Основна увага приділена функціонуванню стеку протоколів TCP/IP, моделі взаємодії відкритих систем OSI, стандартам дротових мереж *Ethernet* та бездротових – типу WI-FI. В посібнику розкрито питання використання схем IP-адресації з використанням маски змінної довжини при реалізації підмереж, зокрема для віртуальних локальних мереж, застосуванню складових корпоративних мереж – таких як технології магістральних каналів та протоколи агрегації каналів, методи резервування основного шлюзу, застосування статичної маршрутизації та базові прийоми забезпечення безпеки мережі.

Практикум сформовано з використанням досвіду практичної роботи з обладнанням корпорації *Cisco* та рекомендацій Мережевої академії *Cisco* для підготовки здобувачів. Короткий опис апаратних складових комутаторів, маршрутизаторів, бездротових контролерів дає необхідні знання для розуміння процесів, які відбуваються в проміжних мережних пристроях для правильного налагодження роботи ОС *Cisco IOS* та ОС кінцевих пристроїв *Windows* та *Linux*. Власні назви, скорочення та аббревіатури є усталеними формами, що характерні для комп'ютерних мереж та галузей знань «Інформаційні технології» і «Автоматизація та приладобудування».

Посібник відрізняється від попередніх аналогічних за назвою видань наявністю детальних моделей, які містять схеми під'єднання, таблиці адресації, сценарії налаштування пристроїв, а також результати виконання команд, що демонструють налаштування та роботу пристроїв і мережі загалом.

Видання підготовлено у співпраці науковців Черкаського державного технологічного університету та Державного університету «Житомирська політехніка».

Автори висловлюють подяку фахівцям корпорації *Cisco* та стейкхолдерам за надані матеріали, ґрунтовні поради і зауваження, що стали в нагоді при підготовці навчального посібника до друку.

РОЗДІЛ 1

СУЧАСНІ ТЕХНОЛОГІЇ МЕРЕЖНОГО ЗВ'ЯЗКУ

У сучасному світі мережні технології відіграють критично важливу роль. Вони забезпечують швидке та надійне передавання даних між різними пристроями та користувачами, що є основою для функціонування інформаційних систем, бізнес-процесів, соціальних мереж, електронної комерції, наукових досліджень і навіть розваг. Мережі дозволяють організаціям та приватним особам обмінюватися інформацією, спільно використовувати ресурси та співпрацювати незалежно від географічного розташування.

Важливість використання мереж у сучасних системах важко переоцінити. Наприклад, компанії використовують мережі для зв'язку між філіями, співробітники отримують доступ до корпоративних ресурсів із будь-якої точки світу, завдяки хмарним сервісам зберігання даних, користувачі можуть зберігати, обробляти та обмінюватися файлами без необхідності в локальному зберіганні. Інтернет, як глобальна мережа, об'єднує мільярди пристроїв і користувачів, забезпечуючи доступ до безмежного обсягу інформації та сервісів, таких як електронна пошта, соціальні мережі, онлайн-банкінг і потокове відео.

Щоб мережа могла забезпечувати послуги ефективно, вона складається з кількох ключових компонентів, кожен з яких має свою унікальну роль. Ці компоненти працюють разом, щоб забезпечити надійний і зручний доступ до інформації.

Вузли або кінцеві пристрої є базовими елементами будь-якої мережі. Це пристрої, такі як комп'ютери, планшети або смартфони, які підключені до мережі і безпосередньо беруть участь у мережному спілкуванні. Кожен вузол має унікальну IP-адресу, що дозволяє ідентифікувати його в межах мережі.

На відміну від кінцевих пристроїв, сервери є спеціалізованими комп'ютерами, які надають послуги іншим пристроям у мережі. Сервери можуть надавати різні послуги, такі як електронна пошта або доступ до вебзастосунків, і використовують відповідне програмне забезпечення для цього. Клієнти, з іншого боку, є пристроями, які використовують програмне забезпечення для запиту та отримання інформації з серверів. Наприклад, веббраузери є клієнтським програмним забезпеченням, яке звертається до вебсерверів для отримання вебсторінок.

У малих мережах, таких як домашні або невеликі офісні мережі, один комп'ютер може виконувати функції і клієнта, і сервера одночасно. Це створює мережі типу *peer-to-peer*, які прості в налаштуванні і мають низькі накладні витрати, але можуть бути менш безпечними і масштабованими.

Кінцеві пристрої, такі як комп'ютери і принтери, є основними учасниками мережі: через них або надсилають, або отримують дані. Для забезпечення зв'язку між кінцевими пристроями та мережею використовуються проміжні пристрої (рисунок 1.1), які забезпечують підключення, регенерацію сигналів, управління

маршрутизацією даних та інші функції, що сприяють безперебійній передачі інформації.

Один з таких пристроїв – комутатор, який з'єднує різні кінцеві пристрої в межах однієї мережі. Комутатор отримує дані і надсилає їх лише на той порт, до якого підключений конкретний пристрій, що допомагає зменшити навантаження на мережу.

Маршрутизатор, з іншого боку, з'єднує різні мережі між собою і відповідає за визначення оптимального маршруту для передачі даних між ними. Він забезпечує взаємодію між мережами і підключення до Інтернету.

Шлюзи виконують роль перекладачів між мережами, які використовують різні протоколи або технології. Вони конвертують дані з одного формату в інший, що дозволяє взаємодіяти різним мережам.

Модеми перетворюють цифрові сигнали в аналогові і назад, що дозволяє передавати дані через телефонні лінії або інші аналогові канали зв'язку.

Точки доступу надають бездротовий доступ до мережі для різних пристроїв, як-то ноутбуки чи смартфони, розширюючи можливості мережі без використання дротових з'єднань.

Нарешті, дані в мережі передаються через різні типи середовищ: через металеві дроти, оптоволоконні кабелі або з використанням бездротових технологій. Кожен тип середовища має свої особливості, які визначають його відповідність для конкретних завдань і вимог мережі.

Загалом, ці компоненти разом забезпечують ефективну та надійну роботу мережі, що дозволяє користувачам здійснювати обмін інформацією та використовувати ресурси без значних перешкод.

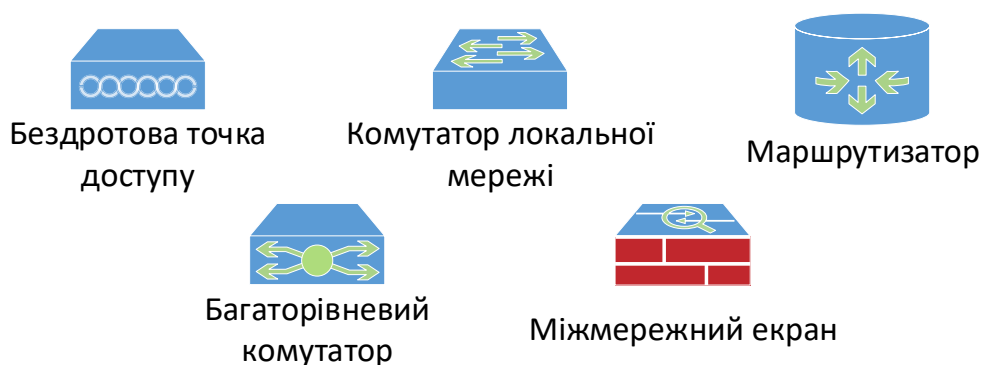


Рисунок 1.1 – Проміжні пристрої мережі

Інфраструктура мережі документується за допомогою спеціальних символів та різних типів діаграм, які відображають з'єднання між пристроями в мережі. Розуміння цих символів та діаграм є важливим аспектом для того, щоб повністю зрозуміти мережну комунікацію.

Використання топологій є критично важливим для ефективного проектування, реалізації та обслуговування мереж. Топології надають чітке уявлення про те, як компоненти мережі з'єднані між собою, що дає змогу:

- Планувати та проектувати мережі: Знання топології допомагає в розробці мереж, визначаючи оптимальні розташування пристроїв і підключення між ними. Це забезпечує ефективне використання ресурсів і уникнення можливих проблем.
- Виявляти і усувати проблеми: Топології спрощують ідентифікацію і локалізацію проблем у мережі. Коли виникають неполадки, топології допомагають швидко знайти місце збою і вжити заходів для його усунення.
- Оцінювати продуктивність і масштабованість: Розуміння топології дозволяє оцінити, як мережа справляється з навантаженням і як вона може бути розширена або змінена без порушення її функціонування.
- Забезпечувати безпеку: Топології допомагають виявити потенційні вразливості в мережі і спроектувати заходи безпеки, щоб захистити мережу від загроз.
- Документувати мережу: Топології забезпечують документацію, яка є важливою для майбутнього обслуговування мережі, навчання нових співробітників та планування оновлень.

Топології мереж можна поділити на фізичні та логічні, кожна з яких має свої особливості і призначення.

Фізична топологія (рисунок 1.2) показує реальне фізичне розташування пристроїв і кабелів в мережі. Вона включає інформацію про те, де саме розміщені проміжні пристрої, комп'ютери, сервери та інші елементи мережі, а також про те, як фізично прокладені кабелі між ними. Фізична топологія є важливою для розуміння того, як саме з'єднані пристрої, і для планування фізичної інфраструктури мережі. Вона допомагає при встановленні нового обладнання та проведенні обслуговування.

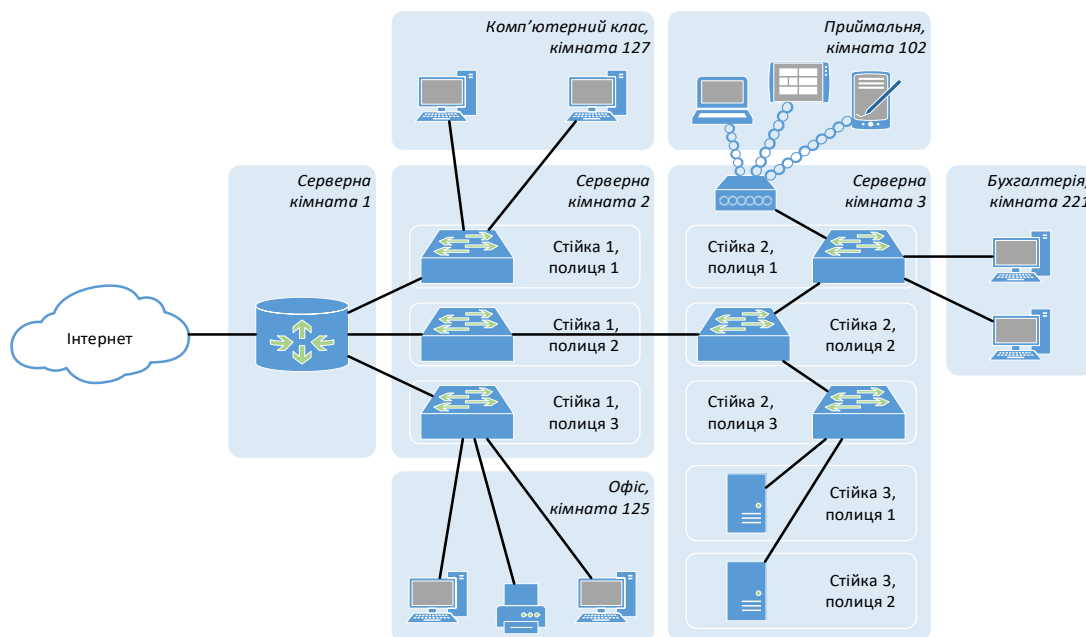


Рисунок 1.2 – Приклад фізичної топології

Логічна топологія описує, як саме дані переміщуються через мережу. Вона відображає логічні з'єднання між пристроями, порти, *VLAN* і схему адресації. Логічна топологія показує, як інформація проходить між пристроями, які використовують різні протоколи та адреси для комунікації. Цей тип топології допомагає зрозуміти, як мережний трафік обробляється і передається, і є корисним для налаштування мережі, управління трафіком та забезпечення безпеки. Приклад логічної топології зображено на рисунку 1.3.

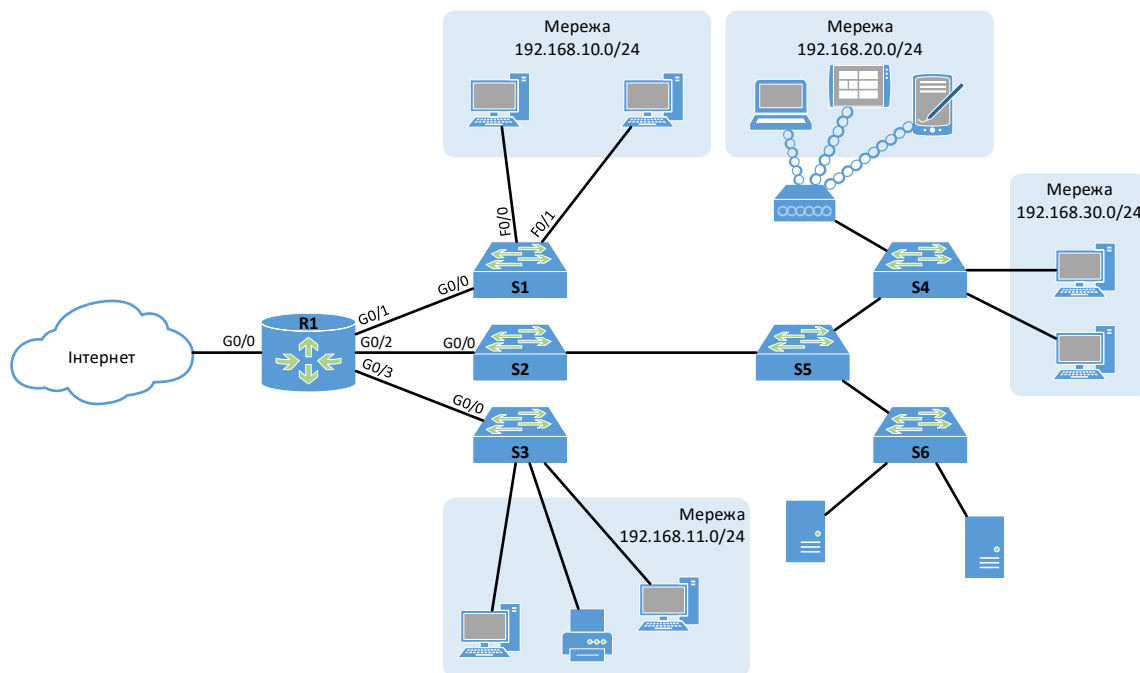


Рисунок 1.3 – Приклад логічної топології

Обидва типи топологій разом забезпечують повну картину мережної інфраструктури. Фізичні топології дають уявлення про реальну конфігурацію обладнання, тоді як логічні топології пояснюють, як саме дані переміщуються та обробляються в мережі. Розуміння обох аспектів є необхідним для ефективного управління і підтримки мережних систем.

Надійність мережі є критичним аспектом, що забезпечує стабільність і ефективність у передачі даних та наданні послуг. Вона включає в себе кілька ключових концепцій, які допомагають забезпечити безперебійність роботи мережі, її здатність до розширення, високу якість обслуговування та безпеку.

Відмовостійкість мережі забезпечує її стабільну роботу навіть у випадку збоїв або відмов компонентів. Це досягається шляхом створення резервних шляхів для передачі даних. Якщо один шлях виходить з ладу, дані автоматично перенаправляються іншим шляхом, завдяки чому забезпечується безперервність комунікації. Наприклад, у мережах з пакетною комутацією дані розбиваються на пакети, які можуть подорожувати різними маршрутами до одного й того ж пункту призначення. Це дозволяє уникнути переривань у зв'язку та забезпечити безперебійне передавання інформації.

Масштабованість мережі означає її здатність адаптуватися до зростаючих потреб у користувачах та обсягах даних без значного зниження продуктивності. Мережа має бути спроектована так, щоб нові пристрої та послуги могли бути легко інтегровані без порушення роботи існуючих сервісів. Це досягається через дотримання стандартів і протоколів, що забезпечують безшовну інтеграцію нових елементів в існуючу мережу.

Якість обслуговування (*Quality of Service – QoS*) є критично важливою для забезпечення високої якості послуг, таких як голосові та відеотрансляції. QoS дозволяє управляти трафіком у мережі, надаючи пріоритет критичним додаткам і сервісам, особливо у випадку перевантаження мережі. Це допомагає уникнути затримок і втрат пакетів, що може погіршити якість обслуговування.

Безпека мережі включає захист як фізичної інфраструктури, так і даних, що передаються мережею. Основні аспекти безпеки включають: конфіденційність (забезпечення того, що дані можуть бути доступні лише авторизованим користувачам); цілісність (гарантія, що дані не були змінені або пошкоджені під час передачі); доступність (забезпечення своєчасного та надійного доступу до даних для авторизованих користувачів).

У мережних комунікаціях існують кілька основних типів доставки повідомлень, які визначають, як саме дані передаються від відправника до одержувача (рисунок 1.4).

При **індивідуальній розсилці (*unicast*)** повідомлення надсилається від одного джерела до одного призначення. Це найпоширеніший тип доставки у мережах, коли одне повідомлення направляється конкретному отримувачу. Протоколи, як-то *TCP* і *UDP*, можуть використовувати індивідуальну розсилку для передавання даних між однією парою пристроїв.

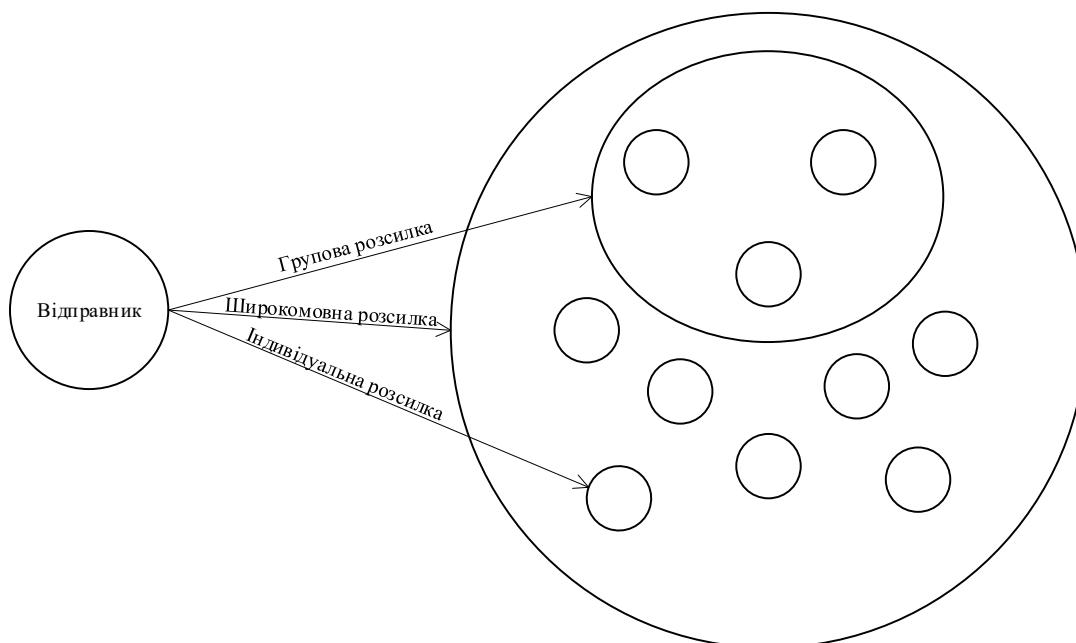


Рисунок 1.4 – Типи розсилок в мережі

При **груповій розсилці (*multicast*)** повідомлення надсилається до вибраної групи пристроїв у мережі, але не до всіх пристроїв. Це дозволяє зменшити навантаження на мережу, передаючи дані лише тим пристроям, які підписані на певну групу. Протокол *IGMP (Internet Group Management Protocol)* використовується для управління членством у групах групової розсилки.

При **широкомовній розсилці (*broadcast*)** повідомлення надсилається всім пристроям у мережі або підмережі. Це означає, що одне повідомлення надходить до всіх пристроїв без розрізнення, хто є одержувачем. Ця розсилка підходить для сценаріїв, коли потрібно розповсюдити інформацію серед усіх пристроїв, наприклад, для автоматичного виявлення пристроїв або оголошень у мережі.

1.1 ПРОТОКОЛИ ТА МОДЕЛІ

Протокол є набором правил і стандартів, які визначають, як дані передаються між різними пристроями в мережі. Це свого роду мова спілкування між комп'ютерами, подібно до того, як люди спілкуються мовами. Якщо два комп'ютери використовують однаковий протокол, вони можуть ефективно обмінюватися інформацією – як люди, що говорять однією мовою. Уявіть, що двоє людей намагаються спілкуватися, але один говорить англійською, а інший – китайською. Без перекладача або спільної мови їхня комунікація буде неефективною або навіть неможливою. Так само, якщо комп'ютери використовують різні протоколи без можливості взаємодії, вони не зможуть обмінюватися даними.

Протоколи виконують різні функції, які є критично важливими для забезпечення успішної комунікації між пристроями в мережі. Основні функції мережних протоколів включають: адресацію, забезпечення надійності, контроль потоку, сегментування, виявлення помилок, надання програмного інтерфейсу.

Протоколи використовують визначені схеми **адресації** для ідентифікації відправника та одержувача повідомлення. Це дозволяє пристроям у мережі точно визначити, куди саме слід доставити дані. Наприклад, протоколи *Ethernet*, *IPv4* та *IPv6* забезпечують адресацію, що дозволяє безпомилково направляти дані до відповідного пристрою.

Функція **надійності** забезпечує доставку повідомлень навіть у випадку їх втрати чи пошкодження під час передачі. Наприклад, протокол *TCP* гарантує доставку даних шляхом повторних спроб передачі пакетів, якщо вони не були отримані належним чином.

Керування потоком регулює швидкість передачі даних між двома пристроями, щоб уникнути перевантаження приймача. Це необхідно для того, щоб пристрій, який отримує дані, міг обробити їх вчасно. Протокол *TCP*, наприклад, реалізує керування потоком, регулюючи швидкість передачі даних залежно від можливостей приймача.

Сегментування забезпечує унікальне маркування кожного сегмента даних, що передається. Це дозволяє приймаючому пристрою правильно зібрати

дані в потрібному порядку, навіть якщо сегменти були доставлені з затримкою або в неправильному порядку. Протокол *TCP* виконує сегментування, що є необхідним для відновлення цілісності повідомлень.

Функція **виявлення помилок** перевіряє, чи дані не були пошкоджені під час передачі. Це дозволяє виявити помилки, які могли статися через завади чи інші проблеми з передаванням. Протоколи – такі як *Ethernet*, *IPv4*, *IPv6* і *TCP* – використовують різні методи для виявлення помилок і забезпечення цілісності переданих даних.

Програмний інтерфейс містить інформацію, яка використовується для комунікації між процесами різних мережних додатків. Наприклад, коли користувач звертається до вебсторінки, протоколи *HTTP* або *HTTPS* використовуються для взаємодії між веббраузером (клієнтом) і вебсервером. Це дозволяє додаткам ефективно обмінюватися інформацією через мережу.

Протоколи групуються в стеки або моделі, щоб упорядкувати й стандартизувати процес передавання даних. Це забезпечує сумісність і ефективність роботи мереж різних типів. Найвідоміші моделі протоколів – це модель *OSI* (*Open Systems Interconnection*) та *TCP/IP* (*Transmission Control Protocol/Internet Protocol*). Їх порівняння сумісно з поширеними протоколами зображено на рисунку 1.5.

Модель *OSI* була розроблена в 1978 році Міжнародним комітетом зі стандартизації (*ISO*). Ця модель складається з семи рівнів, кожен з яких виконує певну функцію. Рівні моделі *OSI* знизу вгору: фізичний, каналний, мережний, транспортний, сеансовий, рівень подання даних і прикладний. Кожен рівень моделі *OSI* спирається на функціонал нижнього рівня і надає сервіси верхньому рівню. Важливою перевагою моделі *OSI* є її універсальність і незалежність від конкретних технологій.

Модель OSI	Протоколи	Модель TCP/IP
Прикладний	DHCP, DNS, FTP, HTTP, IMAP, LDAP, POP3, SIP, SMB, SMTP, SSH, RDP	Прикладний
Рівень подання даних		
Сеансовий		
Транспортний	TCP, UDP	Транспортний
Мережний	IPv4, IPv6, ICMPv4, ICMPv6	Міжмережний
Канальний	Ethernet, WLAN	Рівень мережного доступу
Фізичний		

Рисунок 1.5 – Порівняння моделей *OSI* та *TCP/IP*

Модель *TCP/IP*, що виникла на початку 1970-х років у результаті досліджень Управління перспективних науково-дослідних розробок (*Advanced Research Project Agency, ARPA*) Міністерства оборони США, більш спрощена й практична. Вона складається з чотирьох рівнів: рівень мережного доступу, міжмережний рівень, транспортний і прикладний рівні. Ця модель була створена для забезпечення надійного передавання даних у мережах *ARPANET*, які стали основою Інтернету. Однією з переваг моделі *TCP/IP* є її практична орієнтація та широке використання в сучасних мережах.

Одне велике повідомлення, наприклад, велике зображення або відео, може бути передано мережею як один великий безперервний потік бітів. Однак це може викликати проблеми для інших пристроїв, що використовують ті ж канали зв'язку, оскільки великі обсяги даних можуть призвести до значних затримок, передавання великих обсягів даних монополізує канал. Якщо під час передавання повідомлення якась частина мережі зазнає збоїв, усе повідомлення може бути втрачено і його доведеться передавати заново.

Більш ефективним методом є розділення даних на менші частини. Сегментування – це процес розподілу великого потоку даних на менші, зручніші блоки для передачі через мережу. У мережах, побудованих на основі стека *TCP/IP*, дані передаються у вигляді окремих *IP*-пакетів. Пакети, що містять частини одного і того ж повідомлення, можуть бути відправлені різними маршрутами.

Оскільки великі обсяги даних розбиваються на пакети, передача через мережу відбувається без залежності від одного каналу зв'язку. Це також дозволяє ефективно управляти численними обмінами даних у мережі, забезпечуючи мультимплексування. У випадку проблем з мережею або перевантаженням, при втраті одного сегмента необхідно повторно передати лише цей сегмент, а не весь обсяг даних.

Сегментування дозволяє мережним протоколам адаптуватися до різних умов мережі, наприклад, до змінної пропускної здатності або затримок, що може бути корисним для підтримки високої якості обслуговування. Якщо під час передачі даних виникає помилка, вона зазвичай обмежується конкретним сегментом, що спрощує діагностику і виправлення помилок. Менші сегменти даних можна ефективніше маршрутизувати мережею, що дозволяє краще використовувати мережні ресурси та зменшити затримки.

Сегментування є ключовим елементом для забезпечення ефективної і надійної передачі даних у мережах, особливо в умовах високих навантажень або змінних умов зв'язку.

На етапі підготовки користувацьких даних до передачі, кожен рівень моделі додає свою інформацію до даних, що надходять з верхнього рівня. Цей процес (рисунок 1.6) можна порівняти з надсиланням поштового листа: лист вкладається в конверт, на який додається адреса, потім конверт може бути вкладений в інший конверт для відправлення через поштову службу. На каналному рівні

інкапсульовані дані разом із заголовками називаються кадрами, на мережному рівні – пакетами, а на транспортному – сегментами.

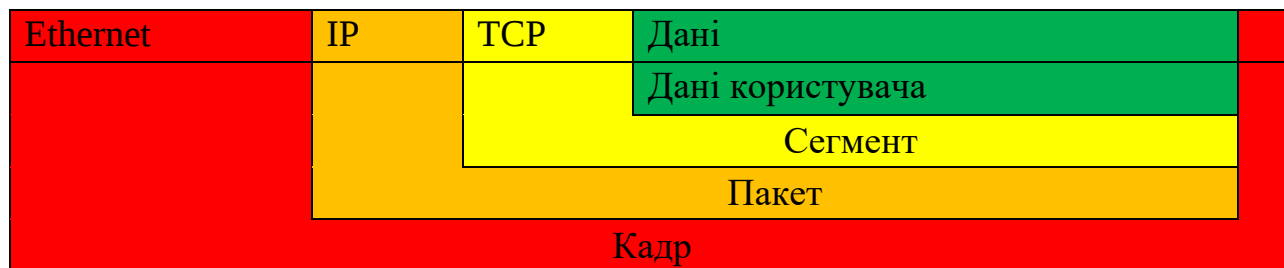


Рисунок 1.6 – Процес інкапсуляції повідомлень

Інкапсуляція – це процес обгортання даних у відповідні заголовки та трейлери на кожному рівні протокольного стека перед їх передачею мережею.

Спочатку дані створюються на рівні додатка, наприклад, текст повідомлення або вебсторінка. Дані розбиваються на сегменти (у випадку *TCP*) і отримують заголовок транспортного рівня. Цей заголовок містить важливу інформацію, як-то порти відправника та отримувача, номер сегмента та контроль помилок. Кожен сегмент інкапсулюється у пакет, до якого додається заголовок мережного рівня. Заголовок мережного рівня включає інформацію про *IP*-адреси відправника та отримувача. Пакет отримує заголовок і трейлер канального рівня, що включає інформацію про фізичну адресу (*MAC*-адресу) та контроль помилок на канальному рівні. Отриманий кадр перетворюється у біти і передається фізичним середовищем.

Деінкапсуляція – це процес, при якому заголовки та трейлери, додані під час інкапсуляції, видаляються на кожному рівні при отриманні даних.

Біти, отримані мережею, конвертуються назад у кадри. На канальному рівні відкидаються заголовок та трейлер і отриманий пакет передається на мережний рівень. Від пакета відкидається мережний заголовок і отриманий сегмент передається на транспортний рівень. Транспортний рівень відкидає заголовки сегментів та збирає сегментоване повідомлення в дані, які потім передаються додаткам у їх первісному вигляді.

Таким чином, інкапсуляція та деінкапсуляція забезпечують правильне форматування і обробку даних на різних етапах їх передачі мережею, що є критично важливим для забезпечення надійної і коректної комунікації.

Стандартизація протоколів є критично важливою для забезпечення ефективної роботи комп'ютерних мереж. Коли протоколи стандартизовані, це дозволяє різним пристроям і системам, виготовленим різними виробниками, працювати разом без проблем. Це створює основу для безперебійної комунікації між різними компонентами мережі, що знижує витрати на розробку нових продуктів та технологій, оскільки виробники можуть орієнтуватися на спільні специфікації.

Стандарти також сприяють підвищенню якості, оскільки всі пристрої, які відповідають стандартам, повинні пройти певні тестування і сертифікацію, що забезпечує стабільність і ефективність мережі. Крім того, відкриті стандарти стимулюють інновації, дозволяючи розробникам створювати нові технології без обмежень, що забезпечує швидше впровадження нових рішень.

Стандарти також відіграють важливу роль у забезпеченні безпеки, визначаючи кращі практики і методи захисту даних. Це допомагає запобігти загрозам і вразливостям у мережах. Нарешті, стандартизація спрощує інтеграцію нових елементів у мережу та обслуговування існуючих компонентів, оскільки не потрібно вносити великі зміни в інфраструктуру для підтримки нових технологій.

Організації, як-то *ISOC (Internet Society)*, *IAB (Internet Architecture Board)*, *IETF (Internet Engineering Task Force)*, *IRTF (Internet Research Task Force)*, *ICANN (Internet Corporation for Assigned Names and Numbers)*, *IANA (Internet Assigned Numbers Authority)* займаються стандартизацією протоколів та моделей.



Рисунок 1.7 – Організації зі стандартизації

Таким чином, протоколи є основою сучасних мережних технологій. Вони забезпечують стандартизовані методи передавання даних, що дозволяє різним пристроям і мережам ефективно взаємодіяти між собою, забезпечуючи надійну та ефективну комунікацію.

1.2 ФІЗИЧНИЙ РІВЕНЬ

Фізичний рівень є основою мережного зв'язку, оскільки саме він відповідає за безпосередню передачу даних між пристроями в мережі. Його головна мета полягає у забезпеченні фізичної передачі бітів через різноманітні середовища, як-то електричні кабелі, оптичні волокна або бездротові канали. Цей рівень займається перетворенням цифрових даних у сигнали, які можна передавати

через ці фізичні середовища, та зворотним перетворенням сигналів у бітові потоки на приймальній стороні.

Процес роботи фізичного рівня починається з того, що він приймає дані з вищого рівня (наприклад, кадри з канального рівня) і перетворює їх на сигнали, які можуть бути зрозумілі для конкретного середовища передачі. Це можуть бути електричні імпульси у випадку мідного кабелю, світлові імпульси для оптоволоконного зв'язку або радіохвилі для бездротових мереж. Потім ці сигнали передаються через середовище до приймального пристрою.

На приймальній стороні пристрій на фізичному рівні зчитує ці сигнали з середовища, перетворює їх назад у бітові потоки, і передає їх на канальний рівень для подальшої обробки. Важливо відзначити, що фізичний рівень не займається розпізнаванням або обробкою даних на рівні кадрів чи пакетів; його завдання – лише передати послідовність бітів з одного пристрою на інший з мінімальними втратами.

Фізичний рівень також відповідає за такі важливі аспекти, як визначення фізичних характеристик середовища передачі, управління швидкістю передачі даних, синхронізацію сигналів і контроль за цілісністю переданих бітів. Він забезпечує базову інфраструктуру, на якій будується вся інша мережа, гарантуючи, що дані можуть бути надійно передані від відправника до отримувача фізичним середовищем.



Рисунок 1.8 – Мережева інтерфейсна карта

Для передачі сигналу використовуються мережеві інтерфейсні карти (*NIC – Network Interface Controller*), які підключають пристрої до мережі та забезпечують передачу сигналів через різні середовища, такі як дротові або бездротові з'єднання. Середовища передачі можуть включати кабелі та інші фізичні

компоненти, що забезпечують фізичну передачу сигналів між пристроями. Інтерфейси та з'єднувачі також є важливими елементами, які з'єднують пристрої і забезпечують їх взаємодію в межах мережі.

Для передавання даних фізичними середовищами над ними відбувається процес кодування. Це процес, під час якого групи бітів представляються в певному шаблоні, що є зрозумілим як для відправника, так і для отримувача. Кодування необхідне для забезпечення надійної та ефективної передачі даних у мережах. Воно перетворює цифрову інформацію в спеціальні сигнали, які можуть бути розпізнані та правильно інтерпретовані як відправником, так і отримувачем. Це допомагає уникнути помилок під час передачі даних фізичним середовищем, забезпечує синхронізацію між пристроями та підвищує загальну ефективність передачі, особливо на високих швидкостях.

У мережах використовуються різні типи кодування, щоб забезпечити коректну передачу даних. Наприклад, Манчестерське кодування використовується в старих стандартах *Ethernet*, таких як *10BASE-T*, де перехід від високого до низького рівня напруги представляє **0**, а перехід від низького до високого – **1**. Для передачі даних на більших швидкостях використовуються більш складні методи кодування, такі як *4B/5B* в *Ethernet 100BASE-TX* або *8B/10B* в *1000BASE-T*.

Фізичне середовище характеризується пропускнуою здатністю – це здатність фізичного середовища передавати дані. Вона вимірюється в кількості бітів, які можуть бути передані з одного місця в інше за певний проміжок часу. Основними одиницями вимірювання пропускнуої здатності є біти на секунду (б/с, bps), кілобіти на секунду (Кб/с, Kbps), мегабіти на секунду (Мб/с, Mbps), гігабіти на секунду (Гб/с, Gbps) і терабіти на секунду (Тб/с, Tbps). Зверніть увагу, що літера «б» у одиницях вимірювання є маленькою, і це показує, що це біти. Для байтів літера «б» була б великою – «Б», як-от МБ/с.

Пропускна здатність використовується для оцінки того, скільки даних може бути передано мережею за певний час, що є ключовим показником ефективності мережних підключень і загальної продуктивності мережі.

Реальна мережа завжди характеризується такими особливостями, як затримка, пропускна здатність і корисна пропускна здатність.

Затримка відображає час, необхідний для передачі даних від джерела до отримувача. Вона включає в себе всі затримки, які виникають при передачі і обробці даних через різні пристрої і мережі. Затримка найчастіше вимірюється в мілісекундах (мс) і є важливою для оцінки швидкості реагування мережі.

Пропускна здатність визначає максимальну кількість даних, яка може бути передана через мережне з'єднання за одиницю часу. Це теоретичний максимум швидкості передачі. Пропускна здатність показує потенціал мережі щодо швидкості передачі даних.

Корисна пропускна здатність описує реальну кількість корисних даних, що можуть бути передані мережею за одиницю часу, з урахуванням різних

накладних витрат, таких як заголовки пакетів, підтвердження отримання і інші витрати на протокол. Корисна пропускна здатність завжди менша за пропускну здатність, оскільки враховує ці додаткові витрати.

Мідний кабель є одним із найстаріших і найпоширеніших засобів для передачі даних у мережах. Він має декілька ключових характеристик, а також переваги і недоліки, що впливають на його застосування.

Мідний кабель може бути представлений у формі незахищеної крученої пари (UTP), екранованої крученої пари (STP) або коаксіального кабелю. Основні характеристики мідного кабелю базуються на тому, що мідь є хорошим провідником електричного струму, що забезпечує високу ефективність передачі сигналів. При цьому мідні кабелі можуть мати різний рівень захисту від фізичних пошкоджень і електромагнітних завад.

Переваги мідного кабелю:

- **Вартість:** Мідний кабель зазвичай дешевший у порівнянні з оптичними волокнами, що робить його економічно вигідним вибором для багатьох мереж.
- **Простота встановлення:** Встановлення мідного кабелю є менш складним і потребує менше спеціалізованого обладнання, ніж оптичне волокно.
- **Сумісність:** Мідний кабель має добре розвинуту інфраструктуру і підтримується багатьма пристроями та стандартами, що полегшує інтеграцію в існуючі мережі.
- **Гнучкість:** Мідні кабелі є досить гнучкими, що полегшує їх прокладання в складних умовах.

Недоліки мідного кабелю:

- **Обмежена дальність:** Сигнали в мідних кабелях зазнають ослаблення на великій відстані, що обмежує їх використання на довгі дистанції.
- **Завади:** Мідний кабель підлягає впливу електромагнітних завад (EMI) і радіочастотних завад (RFI), що може погіршити якість сигналу.
- **Швидкість передачі:** Хоча мідні кабелі підтримують високі швидкості передачі даних, вони зазвичай не можуть досягати швидкостей, доступних для оптичних волокон.

Вибір мідного кабелю для конкретної мережі залежить від багатьох факторів, включаючи вартість, вимоги до швидкості передачі даних, відстані та рівня завад.

Типи кабелів на основі крученої пари, що використовуються в мережах: незахищена кручена пара і екранована кручена пара.

Неекранована кручена пара (Unshielded Twisted-Pair, UTP) – це найпоширеніший тип кабелів у сучасних мережах (рисунок 1.9). Він складається з чотирьох пар кольорових дротів, які скручені разом і захищені лише тонким пластиковим покриттям. Таке скручування допомагає зменшити вплив електромагнітних завад та взаємної завади (перешкоди між сусідніми парами проводів). UTP-

кабелі зазвичай використовуються для з'єднання комп'ютерів, маршрутизаторів і комутаторів в локальних мережах (*LAN*). Попри свою простоту та доступність, *UTP*-кабелі не мають додаткового екранування для захисту від сильних завад.

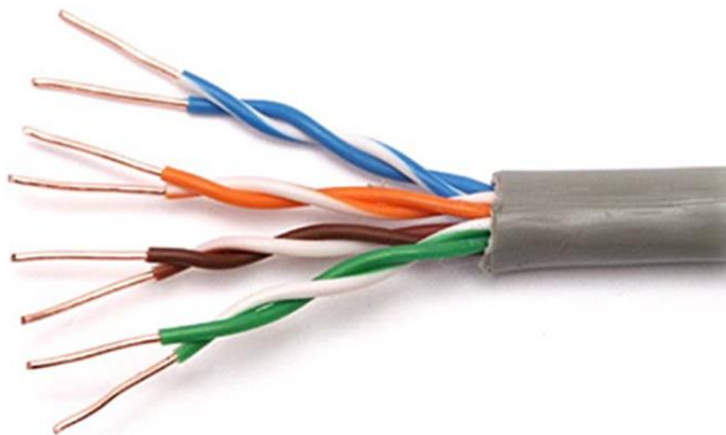


Рисунок 1.9 – Неекранована кручена пара

Екранована кручена пара (Shielded Twisted-Pair, *STP*) забезпечує кращий захист від електромагнітних перешкод і взаємних завад завдяки додатковому екрануванню (рисунок 1.10). У *STP*-кабелях кожна пара дротів може мати окремий шар фольги або екранування, а весь кабель може бути обгорнутий додатковим металевим плетенням. Це робить *STP*-кабелі ідеальними для середовищ з високим рівнем електромагнітних перешкод, як-то промислові зони або великі офісні будівлі, де звичайний *UTP* може не забезпечити достатнього захисту.



Рисунок 1.10 – Екранована кручена пара

UTP кабелі класифікують за категоріями (рисунок 1.11), які визначають їх характеристики і максимальні швидкості передачі даних:

- **Cat 5:** Підтримує швидкість до 100 Mbps на відстані до 100 метрів. Використовується для *Fast Ethernet*.
- **Cat 5e:** Покращена версія Cat 5, яка підтримує швидкість до 1 Gbps і знижує взаємні завади.

- **Cat 6:** Підтримує швидкість до 10 Gbps на коротких відстанях (до 55 метрів) і має покращене екранування для зменшення перешкод.
- **Cat 6a:** Додатково покращене екранування дозволяє досягти швидкості до 10 Gbps на відстані до 100 метрів.
- **Cat 7:** Забезпечує ще краще екранування і підтримує швидкість до 10 Gbps на довших відстанях.
- **Cat 8:** Призначений для швидкостей до 40 Gbps на відстані до 30 метрів.

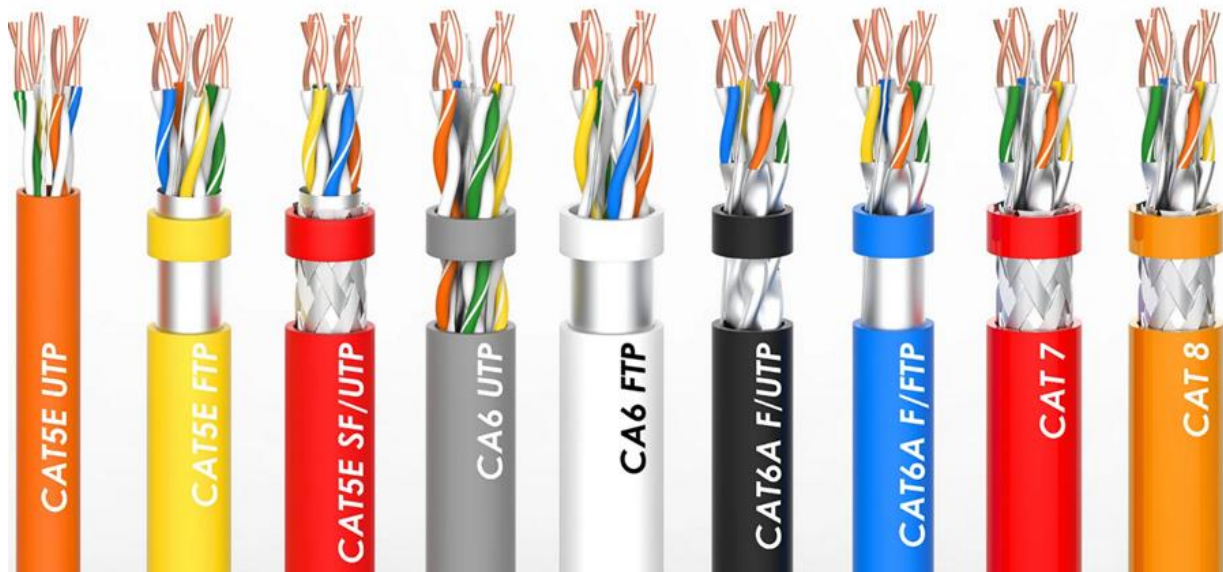


Рисунок 1.11 – Категорії кабелів

Для *UTP* кабелів переважно використовуються роз'єми *RJ-45*. Цей тип з'єднувачів найпоширеніший, особливо для *Ethernet*. Він має вісім контактів і використовується для підключення кабелів до мережних пристроїв, як-то комутатори, маршрутизатори і комп'ютери.

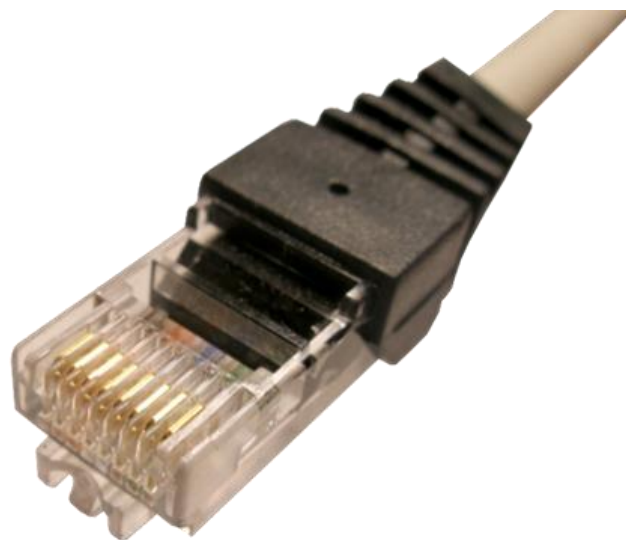


Рисунок 1.12 – Роз'єм *RJ-45*

Типи обтиснення *RJ-45* визначаються за стандартами *T568A* і *T568B*. Ці стандарти визначають, як саме проводи всередині кабелю повинні бути підключені до контактів роз'єму *RJ-45*.

Вибір між *T568A* і *T568B* залежить від того, який стандарт використовують ваші мережеві пристрої (рисунок 1.13). Для правильного функціонування мережі важливо, щоб обидва кінці кабелю мали однаковий стандарт обтиснення: або обидва *T568A*, або обидва *T568B*. Стандарт *T568B* використовується частіше.

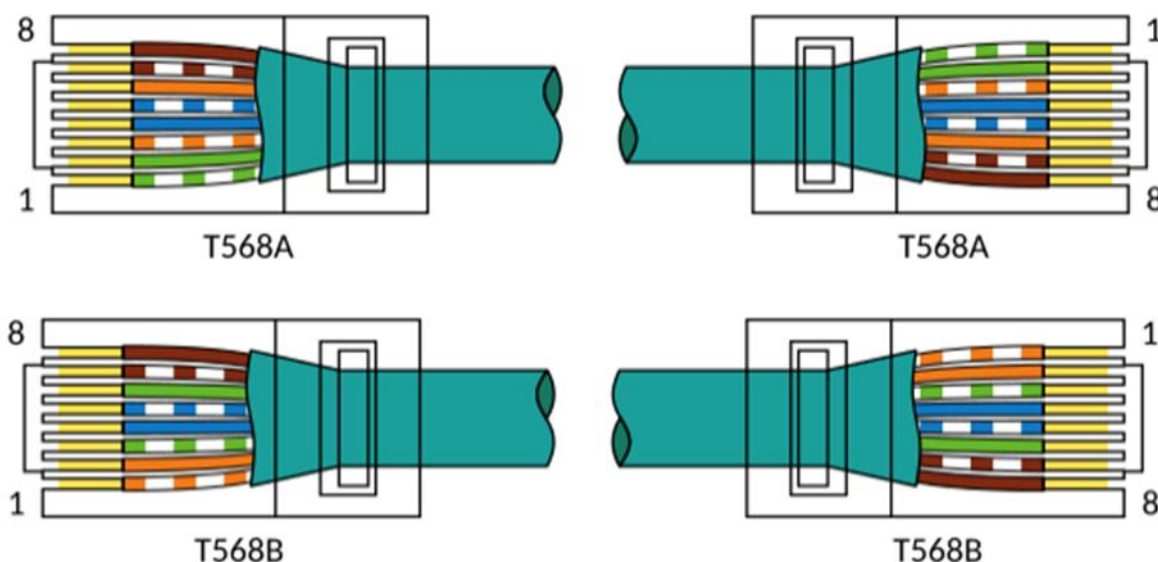


Рисунок 1.13 – Стандарти з'єднання провідників кабелю типу «кручена пара»

На відміну від мідної крученої пари, оптичні лінії зв'язку відзначаються високою пропускнуою здатністю та здатністю передавати дані на великі відстані з мінімальним загасанням сигналу. Вони є повністю нечутливими до електромагнітних та радіочастотних завад, що забезпечує високу надійність передачі даних. Оптичні кабелі складаються з тонких, дуже чистих скляних ниток, які діють як світлові хвилеводи, передаючи світлові імпульси з незначними втратами. Це робить їх незамінними для використання в магістральних мережах, міжбудинкових з'єднаннях та інших застосуваннях, де критично важливі якість і стабільність передавання даних.

Оптичне волокно поділяють на два основні типи: одномодове (*SMF*) і багатомодове (*MMF*).

Одномодове волокно (*single-mode fiber – SMF*) має дуже маленьке ядро і в ньому використовується лазерна технологія для передачі одного променя світла. Цей тип волокна призначений для передачі сигналів на великі відстані, що робить його ідеальним для використання в телекомунікаційних мережах, де потрібна передача даних на сотні кілометрів. Одномодове волокно мінімізує ефект дисперсії, що дозволяє передавати сигнали з меншими втратами і на більші відстані.



Рисунок 1.14 – Одномодове волокно

Багатомодове волокно (*multimode fiber – MMF*) має більше ядро, і в ньому використовуються світлодіодні (*LED*) випромінювачі для передавання світлових імпульсів. Світло в багатомодовому волокні проходить під різними кутами, що приводить до більшої дисперсії, ніж в одномодовому волокні. Це обмежує відстань передачі до 550 метрів, проте *MMF* є популярним у локальних мережах (*LAN*) завдяки меншій вартості *LED* випромінювачів. Багатомодове волокно підтримує швидкість передачі даних до 10 Гбіт/с.

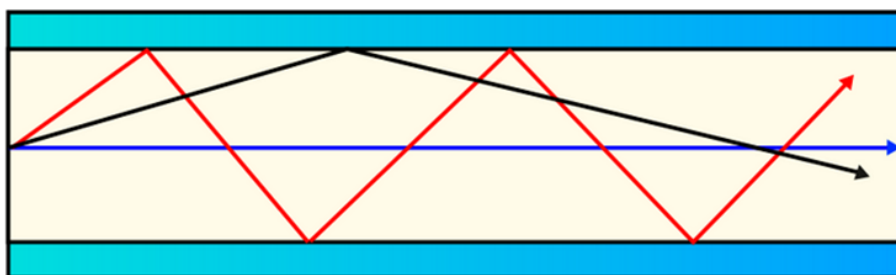


Рисунок 1.15 – Багатомодове волокно

Основною відмінністю між цими двома типами волокон є ступінь дисперсії, який вказує на розсіювання світлового імпульсу з часом. Через більшу дисперсію багатомодове волокно не може передавати сигнали на такі великі відстані, як одномодове волокно.

Оптоволокно використовується в кількох ключових сферах. У корпоративних мережах воно служить для прокладання магістральних ліній, що забезпечує високу пропускну здатність і надійність передачі даних. У технології «оптика до дому» (*FTTH*) оптоволокно застосовується для забезпечення постійного широкосмугового доступу до інтернету для домогосподарств і малого бізнесу. У міжміських та міжнародних мережах воно використовується провайдерами для з'єднання міст і країн. Також оптоволокно прокладається в підводних кабелях, що забезпечують надійні високошвидкісні з'єднання на великих відстанях під водою.

Оптичні конектори використовуються для забезпечення з'єднання між різними оптичними пристроями. Існує декілька типів конекторів, кожен із яких має свої характеристики і методи з'єднання.

ST-роз'єм (Straight-Tip) був одним із перших, який використовувався в оптоволоконних мережах. Він має байонетний механізм з фіксацією, що забезпечує надійне з'єднання. Цей тип роз'єму зазвичай застосовується в мережах локального та корпоративного рівня.



Рисунок 1.16 – ST-конектор

SC-роз'єм (Subscriber Connector), відомий також як квадратний або стандартний роз'єм, є одним із найпоширеніших роз'ємів для оптоволоконних з'єднань у локальних та глобальних мережах. SC-роз'єми можуть бути використані як із одномодовим, так і з багатомодовим оптоволоконном.



Рисунок 1.17 – SC-роз'єм

LC-роз'єм (Lucent Connector), відомий як компактний варіант SC-роз'єму, набуває дедалі більшої популярності завдяки своїм меншим розмірам, що дозволяє економити місце в розподільчих панелях. LC-роз'єми доступні як у простому, так і в дуплексному варіантах. Дуплексний LC-роз'єм дозволяє одночасно передавати і приймати сигнали через один з'єднувач, що зменшує кількість необхідних кабелів.

Також варто зазначити, що для повнодуплексної роботи оптоволоконний кабель раніше потребував використання двох окремих волокон для передавання і приймання сигналів. Але деякі сучасні роз'єми дозволяють передавати і приймати сигнали через один волоконний кабель за допомогою використання різних довжин хвиль світла.



Рисунок 1.18 – LC-роз'єм

Крім того, у роз'ємах оптоволоконних патч-кордів використовуються кольорові оболонки для розрізнення різних типів волокон. Наприклад, жовта оболонка вказує на одномодове волокно, тоді як помаранчева – на багатомодове.

Велика кількість пристроїв, наприклад, смартфони, планшети, ноутбуки, тощо, в наш час використовують бездротові методи передачі. Бездротове середовище забезпечує передачу даних за допомогою електромагнітних сигналів, що використовують радіохвилі для представлення двійкових даних. Це середовище надає користувачам високу мобільність, оскільки не потребує фізичних підключень для з'єднання з мережею. Такий підхід до підключення став основним у багатьох випадках, особливо з урахуванням зростаючої кількості бездротових пристроїв.

Однак бездротове середовище має певні обмеження. Наприклад, ефективність передачі сигналу може залежати від матеріалів будівель або рельєфу місцевості, що може обмежувати зону покриття. Бездротовий зв'язок також схильний до інтерференції, що може виникати через інші електронні пристрої, як-от бездротові телефони, мікрохвильові печі або флуоресцентні лампи.

Безпека є ще одним критичним аспектом. Оскільки бездротове середовище не вимагає фізичного кабелю для передачі даних, це створює ризики несанкціонованого доступу до мережі, що потребує ретельного адміністрування безпеки. Окрім того, оскільки бездротові локальні мережі (WLAN) працюють у напівдуплексному режимі, лише один пристрій може передавати або отримувати дані в певний момент часу. Це може призводити до зниження пропускну здатності мережі при збільшенні кількості користувачів, які одночасно використовують з'єднання.

Бездротове середовище представлено кількома основними типами технологій, що визначаються стандартами *IEEE* та використовуються для передачі даних на різних відстанях і в різних умовах.

WI-FI (IEEE 802.11) є найпоширенішою технологією бездротових локальних мереж (WLAN), що забезпечує підключення пристроїв до мережі. Ця технологія використовує протокол *CSMA/CA* для уникнення колізій під час передачі даних.

Bluetooth (IEEE 802.15) забезпечує зв'язок на коротких відстанях, створюючи бездротові персональні мережі (*WPAN*). Ця технологія використовується для підключення пристроїв на відстанях від одного до ста метрів.

WiMAX (IEEE 802.16) призначений для бездротового широкосмугового доступу і використовує топологію «точка-багатоточка». Ця технологія забезпечує передачу даних на великі відстані та використовується для організації бездротових мереж у масштабах міст або регіонів.

Zigbee (IEEE 802.15.4) створений для передачі даних на невеликі відстані з низькою швидкістю і низьким енергоспоживанням. Ця технологія широко використовується в промислових мережах та Інтернеті речей (*IoT*), наприклад, для керування освітленням або збору даних з різних пристроїв.

Таким чином, фізичний рівень мережі визначає способи передачі даних через різні типи середовищ. Мідні кабелі використовують електричні сигнали для передачі даних, забезпечуючи різну відстань та швидкість, але мають обмеження, такі як чутливість до електромагнітних завад і згасання сигналу на великих відстанях. Оптичні волокна передають дані за допомогою світлових імпульсів, що дозволяє досягти високих швидкостей і великих відстаней без значних втрат сигналу і зовнішніх перешкод. Бездротові технології забезпечують мобільність, використовуючи радіочастоти для передачі даних, але їх ефективність може бути знижена через перешкоди, обмежену зону покриття і питання безпеки. Кожен тип середовища має свої переваги і недоліки, що впливають на вибір у різних мережних сценаріях.

1.3 КАНАЛЬНИЙ РІВЕНЬ. КОМУТАЦІЯ *ETHERNET*

Канальний рівень мережної моделі *OSI* є важливим компонентом для забезпечення коректної і ефективної передачі даних між пристроями в мережі. Цей рівень відповідає за підготовку даних до передачі через фізичний рівень і забезпечує механізми для контролю і управління доступом до мережного середовища. Він відповідає за з'єднання між мережними картами пристроїв.

Згідно зі стандартами *IEEE 802*, канальний рівень мережної моделі *OSI* поділяється на два підрівні: підрівень *LLC (Logical Link Control)* і підрівень *MAC (Media Access Control)*. Кожен із цих підрівнів виконує специфічні функції, які є критичними для забезпечення коректної передачі даних мережею.

Підрівень *LLC* є верхнім підрівнем канального рівня. Його основне завдання – забезпечити зв'язок між мережним програмним забезпеченням верхніх рівнів і апаратним забезпеченням нижчих рівнів. *LLC* відповідає за обробку даних, які надходять з мережного рівня, і додає інформацію, що вказує, який протокол мережного рівня використовується для передачі кадру. Це дозволяє різним протоколам, таким як *IPv4* або *IPv6*, використовувати одне й те саме мережне обладнання і середовище.

Підрівень *MAC* є нижнім підрівнем канального рівня і безпосередньо взаємодіє з фізичним рівнем мережі. Основне завдання *MAC* полягає в управлінні доступом до середовища передачі даних, формуванні кадрів і забезпеченні їхньої доставки. Це включає в себе інкапсуляцію даних у кадри, де додаються заголовки з адресами джерела та призначення, а також трейлера, що використовується для виявлення помилок при передачі. *MAC* відповідає за контроль доступу до спільного середовища у провідних мережах, регулюючи які, пристроям надають право на передавання даних у певний момент часу, а також за управління бездротовою передачею в бездротових мережах, щоб уникнути перешкод і зменшити ймовірність колізій.

Таким чином, *LLC* і *MAC* працюють разом, щоб забезпечити ефективну і надійну передачу даних фізичним рівнем мережі, виконуючи функції, що забезпечують інтеграцію з мережними протоколами і управління доступом до мережного середовища.

Кожне середовище, через яке проходять пакети на шляху від локального до віддаленого вузла, може мати різні характеристики. Наприклад, у локальних мережах *Ethernet* часто багато вузлів конкурують за доступ до середовища передавання даних. Цю проблему вирішує підрівень *MAC*. У випадку послідовних з'єднань метод доступу передбачає прямий зв'язок між лише двома пристроями, зазвичай між двома маршрутизаторами.

На канальному рівні маршрутизатор виконує кілька критичних функцій для обробки і пересилання даних між різними мережами. Ось як це відбувається:

1. **Прийом кадрів:** Коли маршрутизатор отримує кадри з одного з мережних інтерфейсів, він приймає їх з фізичного рівня і декодує вміст на канальному рівні. Цей процес включає перевірку заголовків кадрів, які містять інформацію про джерело та призначення, а також можливі трейлери для виявлення помилок.
2. **Деінкапсуляція:** Маршрутизатор видаляє заголовки і трейлери, що додаються на канальному рівні, щоб отримати чисті дані, які були інкапсульовані у кадрі. Це дозволяє маршрутизатору дістати пакет даних (Layer 3 PDU), який потім буде оброблятися на мережному рівні.
3. **Повторна інкапсуляція:** Після визначення наступного переходу (тобто наступного пристрою на шляху до кінцевої точки), маршрутизатор інкапсулює пакет у новий кадр, що відповідає специфікаціям канального рівня для відповідного інтерфейсу. Це може включати додавання нових заголовків з *MAC*-адресами, які відповідають новому середовищу передавання.
4. **Пересилання кадрів:** Маршрутизатор передає нові кадри через відповідний фізичний інтерфейс до наступного пристрою або маршрутизатора в мережі. Це пересилання включає використання технологій доступу до середовища, які управляють тим, як дані передаються через мережне середовище, щоб уникнути колізій і забезпечити ефективний доступ.

Топології є необхідними для визначення структури та організації мережі, що включає фізичне розташування пристроїв і засоби з'єднання, а також логічний спосіб передавання даних між ними. Вони виконують кілька важливих функцій.

Топології забезпечують чітке уявлення про конфігурацію мережі, допомагаючи проектувати її відповідно до потреб організації. Завдяки правильній топології можна оптимізувати використання ресурсів мережі, мінімізувати затримки та підвищити ефективність передачі даних. Логічна топологія допомагає зрозуміти, як саме дані проходять мережею, що є ключовим для налаштування протоколів маршрутизації та управління доступом до середовища.

У локальних мережах (LAN) популярними є топології типу «зірка» та «розширена зірка». У цих топологіях кінцеві пристрої підключаються до центрального комутатора, що робить мережу масштабованою та простою для діагностики несправностей.

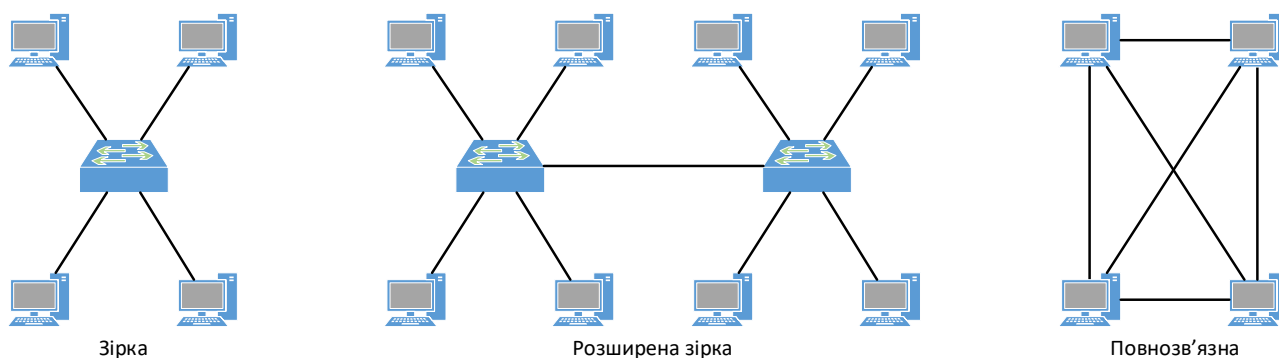


Рисунок 1.19 – Поширені типи топологій

Також топології впливають на надійність мережі. Наприклад, повнозв'язна топологія забезпечує кілька маршрутів для передачі даних, що підвищує стійкість до відмов. Вони також сприяють масштабованості мережі, дозволяючи додавати нові пристрої або змінювати структуру без значних перебоїв у роботі.

Таким чином, топології є фундаментом для побудови ефективної, надійної та легко масштабованої мережі, що відповідає вимогам конкретних завдань.

Залежно від напрямку передавання даних передача в мережах можлива трьома різними способами: повнодуплексним, напівдуплексним та симплексним режимами.

Симплексний режим передбачає передавання даних тільки в одному напрямку. Це означає, що один пристрій завжди відправляє інформацію, а інший тільки приймає. Прикладом може бути телевізійне мовлення, де сигнал передається від телевізійної станції до телевізора, але не навпаки.

Напівдуплексний режим дозволяє обом пристроям передавати та приймати дані, але не одночасно. Передавання даних можливе лише в один момент часу або в одному напрямку. Прикладом напівдуплексного режиму є рації, де

один користувач говорить, а інший слухає, і тільки після завершення передачі повідомлення інший користувач може відповісти.

Повнодуплексний режим забезпечує одночасну передачу даних в обидва напрямки. Це означає, що обидва пристрої можуть одночасно відправляти і приймати інформацію. Прикладом повнодуплексного режиму є телефонний зв'язок, де обидва співрозмовники можуть говорити і слухати одночасно.

Методи контролю доступу визначають, як пристрої в мережі спільно використовують середовище передачі даних та керують процесом доступу до нього. Існує два основних підходи до контролю доступу:

Методи, засновані на конкурентному доступі (*contention-based access*):

У таких мережах пристрої працюють у напівдуплексному режимі та конкурують за право використання середовища. Тільки один пристрій може передавати дані в один момент часу. Якщо два або більше пристроїв намагаються передавати дані одночасно, виникає конфлікт, або колізія. Для уникнення колізій використовуються кілька механізмів:

- **CSMA/CD (*Carrier Sense Multiple Access/Collision Detect*):** Використовується в старих мережах Ethernet із топологією «шина» або з концентраторами (*hubs*). Пристрій перевіряє, чи є вільним середовище, перш ніж почати передачу. Якщо відбувається колізія, пристрої припиняють передачу та намагаються передати дані знову через випадковий інтервал часу.
- **CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*):** Використовується в бездротових мережах (*WLAN*). Пристрій перевіряє середовище і, якщо воно вільне, планує передачу з урахуванням можливого уникнення колізій. Тут передбачено уникнення колізій за рахунок очікування перед передачею та підтвердження отримання кадру від приймаючого пристрою.

Методи контрольованого доступу (*controlled access*): У таких мережах кожен пристрій отримує своєчасне право доступу до середовища передавання даних, що усуває можливість колізій. Цей підхід є детермінованим і менш ефективним, коли йдеться про високу швидкість або велику кількість пристроїв. Приклади таких методів:

- **Token Ring:** пристрої передають спеціальний маркер (*token*) по кільцю. Лише пристрій, що володіє маркером, може передавати дані. Це гарантує, що тільки один пристрій в мережі передає дані в будь-який момент часу.
- **ARCNET (*Attached Resource Computer NETwork*):** Ще один приклад керованого доступу, що базується на подібних принципах.

Обидва методи мають свої переваги і недоліки, і їх вибір залежить від конкретних вимог до мережі, зокрема від швидкості передачі, надійності та типу використовуваного середовища.

На сьогоднішній день *Ethernet*-мережі працюють у повнодуплексному режимі, через що немає потреби у використанні методів доступу.

Канальний рівень, або другий рівень моделі *OSI*, відіграє критичну роль у процесі передачі даних між мережними пристроями. Цей рівень відповідає за забезпечення надійної та зрозумілої передачі даних від одного пристрою до іншого через фізичне середовище. Для цього канальний рівень використовує спеціальні структури даних, відомі як **кадри (frames)**, які включають важливу інформацію для адресації, управління доступом та перевірки цілісності даних.

Під час передачі даних між мережними пристроями (*NIC*) на канальному рівні відбувається процес інкапсуляції. Дані, сформовані на мережному рівні (Layer 3), наприклад, у вигляді IP-пакета, обгортаються спеціальним заголовком та трейлером, створюючи кадр.

Заголовок кадру включає:

- **Фізичну адресу джерела** – це унікальний ідентифікатор, що визначає пристрій, який відправляє дані.
- **Фізичну адресу призначення** – це ідентифікатор пристрою, якому призначений цей кадр.

Крім того, **трейлер кадру** містить інформацію для перевірки цілісності переданих даних (**CRC**). Цей контроль забезпечує виявлення помилок, які могли виникнути під час передавання кадру через фізичне середовище.

Канальний рівень підтримує різні протоколи, залежно від типу мережі та середовища передачі. Найпоширеніші з них: *Ethernet*, *802.11 (Wi-Fi)*, *Point-to-Point Protocol (PPP)*, *High-Level Data Link Control (HDLC)*, *Frame Relay* та *Asynchronous Transfer Mode (ATM)*.

Ці протоколи не тільки забезпечують інкапсуляцію даних, але й регулюють доступ до середовища передавання, що є критичним у багатокористувацьких мережах.

Канальний рівень використовує **фізичні адреси** для передачі кадрів через локальне середовище. На відміну від мережних адрес (*IP-адрес*), які є ієрархічними та використовуються для маршрутизації даних через різні мережі, фізичні адреси унікальні для кожного мережного інтерфейсу та діють тільки в межах однієї локальної мережі.

Кадри, передані на канальному рівні, залишаються дійсними тільки в межах локальної мережі. Якщо дані мають бути передані в іншу мережу або через маршрутизатор, кадр деінкапсулюється для аналізу *IP-адреси*. Маршрутизатор створює новий кадр з новими фізичними адресами для передачі даних через наступний сегмент мережі.

Вибір протоколу на канальному рівні залежить від таких факторів, як топологія мережі, середовище передачі, кількість користувачів та географічний масштаб. У локальних мережах (*LAN*) використовуються високошвидкісні протоколи, такі як *Ethernet*, які здатні підтримувати значну кількість пристроїв. У той же час, для мереж *WAN*, які охоплюють великі відстані, застосовуються інші протоколи, що враховують вартість і пропускну здатність каналів зв'язку.

Формат кадру *Ethernet* описує структуру даних, що передаються між пристроями в мережі. Він включає кілька важливих компонентів:

- **Преамбула** (7 байтів) та **Початковий обмежувач кадра (SFD)** (1 байт) використовуються для синхронізації між відправником і приймачем. Преамбула допомагає налаштувати приймач на правильну частоту сигналу, а SFD вказує на початок кадру.
- **MAC-адреса призначення** і **MAC-адреса джерела** по 6 байтів кожна. Перша адреса вказує на пристрій, для якого призначено кадр, а друга – на пристрій, що відправив кадр.
- **Тип/Довжина** (2 байти) визначає, який протокол верхнього рівня інкапсульовано в кадрі, або довжину даних.
- **Дані** (від 46 до 1500 байтів) містять інформацію, яка передається на верхніх рівнях. Якщо дані менші 46 байтів, додаються додаткові біти для досягнення мінімального розміру кадру.
- **Контрольна послідовність кадру (FCS)** (4 байти) використовується для перевірки цілісності кадру за допомогою *CRC*.

Розмір кадру *Ethernet* варіюється від 64 до 1518 байтів з урахуванням заголовків (рисунок 1.20). Кадри менші за 64 байт і більші за 1518 байт відкидаються.

MAC-адреса *Ethernet* – це 48-бітне значення, яке зазвичай виражається у 12 шістнадцяткових цифрах, що також можна розглядати як 6 байтів.

64-1518 байт					
8 байт	6 байт	6 байт	2 байти	46-1500 байт	4 байти
Преамбула та початковий обмежувач кадра	MAC-адреса призначення	MAC-адреса джерела	Тип/Довжина	Дані	Контрольна послідовність кадра

Рисунок 1.20 – Поля кадру *Ethernet*

Структура MAC-адреси складається з двох частин. *OUI (Organizationally Unique Identifier)* – перші 6 цифр (24 біти) визначають виробника обладнання. Виробник отримує цей ідентифікатор від *IEEE*. *Vendor-Specific Portion* – останні 6 цифр (24 біти) є унікальними для кожного пристрою, який виготовляється виробником (рисунок 1.21).

24 біти			24 біти		
00	60	2F	3A	07	BC
Organizationally Unique Identifier			Vendor-Specific Portion		

Рисунок 1.21 – MAC-адреса *Ethernet*

Наприклад, якщо *Cisco* отримала *OUI* 00-60-2F, і надає унікальний код 3A-07-BC своєму пристрою, то *MAC*-адреса буде виглядати як 00-60-2F-3A-07-BC.

За доменом розсилки виділяють три типи *MAC*-адрес, *MAC*-адреса індивідуальної розсилки (*unicast*), ширококомовна *MAC*-адреса (*broadcast*) та групова *MAC*-адреса (*multicast*).

Unicast: Ідентифікує лише один пристрій. Наприклад, кадр з адресою 00-60-2F-3A-07-BC відправляється конкретному пристрою.

Broadcast: Адреса FF-FF-FF-FF-FF-FF вказує на всі пристрої в мережі. Такі кадри обробляються кожним пристроєм в сегменті мережі. Не може бути в полі адреси відправника.

Multicast: Адреси, як-то 01-00-5E для *IPv4* або 33-33 для *IPv6*, використовуються для передачі даних до групи пристроїв.

При отриманні кадру *Ethernet*, пристрій перевіряє *MAC*-адресу призначення. Якщо вона збігається з адресою пристрою або є ширококомовною/груповою адресою, кадр оброблюється. В іншому випадку кадр відкидається.

Сучасні операційні системи та мережні карти дозволяють змінювати *MAC*-адресу програмним способом, що може вплинути на безпеку мережі, якщо вона використовує *MAC*-фільтрацію для контролю доступу.

Таблиця *MAC*-адрес комутатора є критично важливим елементом для ефективної роботи *Ethernet*-мережі. Вона дозволяє комутатору приймати рішення про те, як і куди пересилати *Ethernet*-кадри.

Пересилання кадрів через комутатор *Ethernet* включає кілька ключових етапів. Коли комутатор отримує кадр, він спочатку перевіряє *MAC*-адресу джерела, яка дозволяє йому дізнатися, на якому порту цей кадр з'явився. Якщо *MAC*-адреса джерела ще не є в таблиці *MAC*-адрес, комутатор додає новий запис з адресою та портом, через який був отриманий кадр. Якщо адреса вже є в таблиці, комутатор оновлює таймер для цього запису, підтверджуючи, що адреса все ще актуальна.

Після того як *MAC*-адреса джерела зафіксована, комутатор перевіряє *MAC*-адресу призначення кадру. Якщо призначена адреса є в таблиці, комутатор пересилає кадр тільки на порт, відповідний цій адресі. Це дозволяє уникнути перевантаження мережі, направляючи дані лише на необхідний порт.

Якщо адреса призначення відсутня в таблиці, це означає, що комутатор не знає, на який порт слід відправити кадр. У такому випадку комутатор розсилає кадр на всі порти, крім того, з якого він надійшов. Це гарантує, що кадр потрапить до призначеного пристрою, хоча це може тимчасово збільшити навантаження на мережу.

Для ширококомовних та групових кадрів комутатор завжди надсилає дані на всі порти. Широкомовні кадри надсилаються на всі пристрої в мережі, а групові кадри відправляються на пристрої, які є частиною відповідної групи.

Таким чином, комутатор використовує свою таблицю MAC-адрес для прийняття рішень про пересилання кадрів, що дозволяє ефективно управляти мережею, зменшувати колізії та покращувати загальну продуктивність мережі.

Таблиця MAC-адрес комутатора є динамічною структурою даних, що постійно оновлюється, щоб забезпечити швидке і ефективне пересилання кадрів у локальних мережах.

1.4 МЕРЕЖНИЙ РІВЕНЬ

Мережний рівень (Рівень 3) моделі OSI забезпечує комунікацію між пристроями в різних мережах. Основні протоколи цього рівня включають IPv4 та IPv6, а також протоколи маршрутизації, такі як OSPF, і протоколи повідомлень, такі як ICMP.

Основні операції, що виконуються на мережному рівні:

- **Адресація:** Кожен пристрій у мережі повинен мати унікальну IP-адресу для ідентифікації. Це забезпечує можливість передачі даних між різними пристроями.
- **Інкапсуляція:** Мережний рівень інкапсулює дані з транспортного рівня в пакет, додаючи до нього IP-заголовок. Заголовок містить IP-адреси відправника та отримувача, що необхідно для доставки пакета.
- **Маршрутизація:** Для передачі пакетів між мережами використовуються маршрутизатори. Вони обирають найкращий шлях для пакетів і передають їх до кінцевого пристрою. Пакет може проходити через кілька маршрутизаторів, кожен з яких називається «переходом».
- **Деінкапсуляція:** При досягненні пакета мережного рівня отримувач перевіряє IP-заголовок і, якщо адреса відповідає його власній, видаляє заголовок. Пакет потім передається на відповідний сервіс транспортного рівня.

IP-адресація залишається незмінною до досягнення кінцевого пристрою, за винятком випадків, коли застосовується *Network Address Translation (NAT)*.

Характеристики IP-протоколу включають:

- **Не орієнтований на з'єднання:** IP-протокол не створює постійного з'єднання перед відправкою даних. Це схоже на надсилання листа без попереднього повідомлення отримувача.
- **Негарантована доставка:** IP-протокол не гарантує доставку пакетів. Це означає, що пакети можуть бути втрачені або пошкоджені без відома відправника.
- **Незалежний від середовища:** IP-протокол працює незалежно від типу середовища передачі (мідь, оптичне волокно, бездротовий зв'язок). Однак IP-пакети можуть бути розподілені різними типами середовищ.

Коли пакет пересилається через мережі з різним розміром блоку переданих даних (*Maximum Transmission Unit – MTU*), може знадобитися розбиття пакета на частини. Це називається фрагментацією і може спричиняти затримки. Важливо, що пакети *IPv6* не можуть бути фрагментовані маршрутизаторами.

Протокол *IPv4* є основним протоколом мережного рівня, який забезпечує передачу даних від відправника до отримувача. Заголовок пакета *IPv4* містить важливу інформацію, яка дозволяє забезпечити доставку пакета до його призначення (рисунок 1.22).

Байт 1		Байт 2		Байт 3	Байт 4	20 байт	
Версія	IHL	DS		Загальна довжина			
		DSCP	ECN				
Ідентифікація			Прапори	Зсув фрагмента			
Час життя		Протокол		Контрольна сума заголовка			
IP-адреса джерела							
IP-адреса призначення							

Рисунок 1.22 – Заголовок *IPv4*

Поля заголовка *IPv4* містять:

- **Версія:** Це 4-бітне поле, яке встановлено в 0100, що вказує на те, що це пакет *IPv4*.
- **DiffServ (Differentiated Services):** Раніше відоме як поле *Type of Service (ToS)*. Це 8-бітне поле визначає пріоритет пакета. Воно містить 6 біт для точки коду диференційованих послуг (*Differentiated Services Code Point – DSCP*) та 2 біти для явного повідомлення про перевантаження (*Explicit Congestion Notification – ECN*).
- **Контрольна сума заголовка:** Це поле використовується для виявлення пошкоджень в заголовку *IPv4*.
- **Час життя (Time to Live – TTL):** Це 8-бітне поле обмежує час існування пакета в мережі. Початкове значення *TTL* задається відправником і зменшується на одиницю кожен раз, коли пакет обробляється маршрутизатором. Коли *TTL* досягає нуля, маршрутизатор відкидає пакет і надсилає повідомлення *ICMP Time Exceeded* до відправника.
- **Протокол:** Це 8-бітне поле ідентифікує наступний рівень протоколу, що вказує тип даних у корисному навантаженні пакета. Це дозволяє мережному рівню передати дані відповідному протоколу верхнього рівня. Типові значення включають *ICMP* (1), *TCP* (6) та *UDP* (17).
- **Адреса джерела IPv4:** Це 32-бітне поле представляє *IP*-адресу джерела пакета. Адреса джерела завжди є індивідуальною.
- **Адреса призначення IPv4:** Це 32-бітне поле представляє *IP*-адресу призначення пакета. Адреса призначення може бути індивідуальною, груповою або широкомовною.

Інші поля заголовка:

- **Довжина Інтернет-заголовка (Internet Header Length – IHL):** Визначає довжину заголовка пакета.
- **Загальна довжина:** Вказує загальну довжину пакета, включаючи заголовок і корисне навантаження.
- **Ідентифікація, Прапори та Зсув фрагмента:** Використовуються для відстеження фрагментів, якщо пакет потрібно фрагментувати через менший MTU.

Заголовок *IPv4* дозволяє маршрутизаторам і іншим мережним пристроям правильно обробляти пакети, перевіряючи ці поля для забезпечення коректності і доставлення даних до отримувача.

IPv6 розроблений як наступник *IPv4* для вирішення обмежень останнього. *IPv4*, хоча ще використовується, має три основні обмеження, які впливають на його функціональність:

1. **Вичерпання адрес *IPv4*:** *IPv4* має обмежену кількість унікальних публічних адрес, близько 4 мільярдів. Зростання числа пристроїв, які потребують *IP*-адрес, створює потребу в більших обсягах адрес.
2. **Відсутність прямого з'єднання:** Використання NAT (Network Address Translation) в *IPv4* дозволяє кільком пристроям ділитися однією публічною *IP*-адресою, що ускладнює прямий зв'язок між кінцевими пристроями і ускладнює роботу деяких технологій.
3. **Збільшена складність мережі:** NAT створює додаткову складність в мережі, що призводить до затримок і ускладнює усунення несправностей.

IPv6 був розроблений у 1990-х роках для вирішення цих обмежень. Основні переваги *IPv6* включають збільшений простір адрес (*IPv6* використовує 128-бітні адреси, що забезпечує набагато більшу кількість унікальних адрес (340 ундецильйонів), порівняно з 32-бітними адресами *IPv4*); поліпшене оброблення пакетів (заголовок *IPv6* спрощений і має менше полів, що підвищує ефективність обробки); відсутність необхідності в NAT (велика кількість публічних адрес *IPv6* усуває потребу в NAT, що спрощує мережні комунікації).

Заголовок *IPv6* пакета (рисунок 1.23) має фіксовану довжину 40 байт і містить такі поля:

1. **Версія:** 4-бітне поле, яке має значення 0110, що вказує на *IPv6*.
2. **Клас трафіку:** 8-бітне поле, еквівалентне полю *Differentiated Services (DS)* в *IPv4*. Використовується для визначення пріоритету пакета.
3. **Мітка потоку:** 20-бітне поле, яке пропонує однаковий тип обробки для всіх пакетів з однаковим тегом потоку.
4. **Довжина корисного навантаження:** 16-бітне поле, яке вказує довжину корисного навантаження пакета, без урахування заголовка *IPv6*.
5. **Наступний заголовок:** 8-бітне поле, еквівалентне полю Протокол в *IPv4*. Вказує тип даних, які передаються до відповідного протоколу верхнього рівня.

6. **Обмеження переходів:** 8-бітне поле, яке замінює поле *TTL* в *IPv4*. Значення зменшується на одиницю кожен раз, коли пакет проходить через маршрутизатор. При досягненні нуля пакет відкидається, і надсилається повідомлення *ICMPv6 Time Exceeded*.
7. **IPv6-адреса джерела:** 128-бітне поле, яке ідентифікує *IP*-адресу відправника.
8. **IPv6-адреса призначення:** 128-бітне поле, яке ідентифікує *IP*-адресу отримувача.

Байт 1		Байт 2		Байт 3		Байт 4		
Версія	Клас трафіку		Мітка потоку					40 байт
Довжина корисного навантаження			Наступний заголовок		Обмеження переходів			
IP-адреса джерела								
IP-адреса призначення								

Рисунок 1.23 – Заголовок *IPv6*

Заголовок *IPv6* спрощений і полегшує обробку пакетів, що підвищує ефективність мережі. Зокрема, *IPv6* не вимагає перерахунку контрольної суми заголовка при передачі через маршрутизатори, що зменшує навантаження на мережу.

1.4.1 ПРОТОКОЛ ARP

Протокол *ARP* (*Address Resolution Protocol*) є необхідним для забезпечення коректної комунікації між пристроями в мережах *IPv4*, які використовують *Ethernet*. На мережному рівні (Рівень 3) пристрої ідентифікуються за допомогою *IPv4*-адрес, які використовуються для маршрутизації пакетів між мережами. Однак, на канальному рівні (Рівень 2) передача даних відбувається за допомогою фізичних *MAC*-адрес.

ARP виконує критичну функцію відображення між цими двома типами адрес: він зіставляє відомі *IPv4*-адреси з відповідними *MAC*-адресами. Коли пристрій хоче надіслати дані іншому пристрою в тій самій локальній мережі, він повинен знати *MAC*-адресу призначення, щоб сформувати *Ethernet*-кадр для передачі на канальному рівні. Якщо *MAC*-адреса невідома, пристрій використовує протокол *ARP* для її визначення.

ARP-таблиця (іноді також називається *ARP*-кеш) – це тимчасовий список зіставлень між *IP*-адресами та *MAC*-адресами пристроїв у локальній мережі. Кожен запис у цій таблиці містить пару «*IP*-адреса - *MAC*-адреса», що дозволяє мережним пристроям швидко знайти необхідну *MAC*-адресу, якщо відома *IP*-адреса.

Використання *ARP*-таблиці дозволяє досягти:

- прискорення процесу передачі даних – *ARP*-таблиця дозволяє пристроям у мережі швидко визначити *MAC*-адресу, пов'язану з певною *IP*-адресою, без необхідності кожного разу надсилати широкомовний *ARP*-запит. Це значно прискорює процес передавання даних у мережі;
- зменшення навантаження на мережу – оскільки кожен *ARP*-запит є широкомовним, його обробляють усі пристрої в мережі. Збереження зіставлень у *ARP*-таблиці допомагає знизити кількість таких запитів, що зменшує навантаження на мережу.

Записи в *ARP*-таблиці зберігаються лише тимчасово. Вони мають свій часовий ліміт і видаляються з таблиці після закінчення певного періоду, якщо не використовуються. Це забезпечує актуальність і відповідність інформації у таблиці.

Для перегляду *ARP*-таблиці у таких операційних системах як Windows, Linux та macOS можна використати команду **arp -a** (рисунок 1.24).

```
C:\Windows\system32> arp -a -v
Interface: 127.6.6.1 --- 0x1
    Internet Address      Physical Address      Type
    224.0.0.22            static
    224.0.1.60            static
    239.255.255.250       static

Interface: 192.168.1.107 --- 0x6
    Internet Address      Physical Address      Type
    192.168.1.1           10-93-97-53-e4-20     dynamic
    192.168.1.255         ff.ff.ff.ff.ff.ff     static
    224.0.0.22            01-09-5e-00-00-16     static
    224.0.0.251           01-00-5e-00-00-fb     static
    224.0.0.252           01-00-5e-00-00-fc     static
    224.0.1.60            01-09-5e-00-01-3c     static
    239.255.255.250       01-00-5e-7f-ff-fa     static
    255.255.255.255       ff.ff.ff.ff.ff.ff     static
```

Рисунок 1.24 – Перегляд *ARP*-таблиці

Процес визначення *MAC*-адреси за *IPv4*-адресою відбувається наступним чином:

1. **Перевірка *ARP*-таблиці:** пристрій спочатку перевіряє свою локальну *ARP*-таблицю (кеш), щоб знайти *MAC*-адресу, яка відповідає потрібній *IPv4*-адресі;

2. **ARP-запит:** якщо відповідного запису немає, пристрій надсилає широкомовний ARP-запит мережею, запитуючи: «Хто має цю IPv4-адресу? Повідомте мені вашу MAC-адресу.»;
3. **ARP-відповідь:** пристрій, який має запитувану IPv4-адресу, відповідає ARP-відповіддю, надсилаючи свою MAC-адресу безпосередньо запитувачу;
4. **Оновлення ARP-таблиці:** запитуючий пристрій отримує MAC-адресу та оновлює свою ARP-таблицю для подальшого використання;
5. **Передача даних:** Тепер, знаючи MAC-адресу призначення, пристрій може сформулювати Ethernet-кадр і передати дані на канальному рівні.

Протокол ARP має декілька проблем, які можуть впливати на ефективність і безпеку мережі.

ARP-спуфінг та отруєння кешу ARP (ARP Spoofing/Poisoning) є однією з найбільш критичних атак на ARP. Зловмисник може відправити підроблені ARP-відповіді, у яких вказується, що певна IP-адреса відповідає іншій, підставній MAC-адресі. Це дозволяє зловмиснику перехоплювати, змінювати або перенаправляти мережний трафік. Така атака може призвести до значних проблем із безпекою, наприклад, до крадіжки даних або здійснення атак типу «людина посередині» (*Man-in-the-Middle*).

ARP-запити надсилаються у вигляді широкомовних повідомлень (*broadcast*), які приймаються всіма пристроями на локальній мережі. Це може призвести до перевантаження мережі, особливо у великих мережах із багатьма пристроями, що впливає на її продуктивність.

ARP не має механізмів автентифікації, що дозволяє легко підробляти ARP-відповіді та проводити атаки. Це означає, що ARP-відповіді приймаються без перевірки їх достовірності.

Також зловмисники можуть використовувати ARP-запити для флуду в мережі, що може призвести до погіршення її роботи або навіть до атаки відмови в обслуговуванні (*DoS-атака*).

В цілому, незважаючи на важливість ARP для функціонування IPv4-мереж, ці проблеми вимагають додаткових заходів захисту, таких як використання фільтрів ARP, динамічної перевірки ARP (*Dynamic ARP Inspection – DAI*) або переходу на IPv6, який використовує більш захищений протокол виявлення сусіда (*Neighbor Discovery – ND*).

Протокол ND є ключовим елементом у функціонуванні мереж на базі IPv6. Він забезпечує різноманітні функції, пов'язані з адресацією і маршрутизацією, аналогічні до тих, що виконує ARP в мережах IPv4, але значно розширює їх можливості.

Основна задача ND полягає у визначенні відповідності між MAC-адресами та IPv6-адресами. Для цього він використовує повідомлення ICMPv6, серед яких:

- *Neighbor Solicitation* (запит сусіда) – це повідомлення, яке використовується для визначення MAC-адреси пристрою, що має відому IPv6-адресу.
- *Neighbor Advertisement* (анонсування сусіда) – це відповідь на запит, що містить MAC-адресу пристрою.

Додатково у протоколі *ND* для IPv6 повідомлення *Router Solicitation* (запит маршрутизатора) та *Router Advertisement* (анонсування маршрутизатора) відіграють важливу роль в автоматичній конфігурації пристроїв і виявленні маршрутизаторів у мережі.

Коли пристрій з'являється в мережі або потребує оновлення своєї конфігурації, він надсилає повідомлення *Router Solicitation*. Це повідомлення є запитом до наявних маршрутизаторів у мережі, щоб вони відповіли своїми параметрами конфігурації. Запит маршрутизатора відправляється у вигляді групової розсилки, щоб досягнути всі маршрутизатори в локальній мережі.

Маршрутизатор, отримавши повідомлення *Router Solicitation*, або за власною ініціативою періодично надсилає повідомлення *Router Advertisement*. Це повідомлення містить важливу інформацію, яку пристрої можуть використовувати для налаштування своїх мережних інтерфейсів. Наприклад, префікси IPv6-адрес, які пристрої можуть використовувати для налаштування власних IPv6-адрес, адреси DNS-серверів, параметри конфігурації, такі як періоди оновлення, інформація про стан мережі, можливість використання *Stateless Address Autoconfiguration (SLAAC)* для автоматичної конфігурації.

На відміну від *ARP*, *ND* використовує групову розсилку замість широкомовної, що зменшує навантаження на мережу і підвищує її ефективність. Також *ND* має вбудовані засоби безпеки, як-то *Secure Neighbor Discovery (SEND)*, що допомагає захистити мережу від атак типу підробки повідомлень.

1.4.2 АДРЕСАЦІЯ IPv4

Протокол *IPv4*, попри активний перехід на *IPv6*, залишається важливим для багатьох мереж, тож мережні адміністратори повинні володіти знаннями про його адресацію та принципи роботи. Основні аспекти *IPv4* включають структуру адреси, розділення мережі на підмережі та використання змінної довжини маски підмережі (*VLSM*).

IPv4-адреса складається з 32 бітів. Ці біти діляться на дві частини: мережну і вузлову. Мережна частина визначає, до якої мережі належить пристрій, а вузлова частина вказує на конкретний пристрій (вузол) у цій мережі.

Щоб зрозуміти, де закінчується мережна частина і починається вузлова, використовується маска підмережі. Маска підмережі також складається з 32 бітів і містить одиниці для мережної частини та нулі для вузлової. Одиничні біти в масці показують, які біти *IPv4*-адреси відповідають за мережу, а нульові біти вказують на вузлову частину. Для маски підмережі обов'язковою є послідовність одиниць на початку та послідовність нулів у кінці.

Для зручності сприйняття *IPv4*-адреса представляється як послідовність з чотирьох чисел, розділених крапками, де кожне число може бути в діапазоні від 0 до 255. Це представлення – десяткове, а фактично кожна частина адреси є байтом (8 біт), тому загалом *IPv4*-адреса містить 32 біти.

Маска підмережі також записується у вигляді чотирьох чисел, розділених крапками, але її біти завжди розпочинаються з одиниць, що переходять у нулі. Наприклад, маска 255.255.255.0 вказує, що перші 24 біти використовуються для визначення адреси мережі, а останні 8 – для адрес вузлів у цій мережі.

Маска підмережі може бути представлена двома способами. Один – це класичне представлення через чотири числа, як у прикладі 255.255.255.0. Інший, більш сучасний формат, відомий як префіксна нотація або «слеш-нотація». У цьому форматі після *IPv4*-адреси через слеш («/») записується кількість одиничних біт у масці. Наприклад, 192.168.1.1/24 означає, що перші 24 біти маски це одиниці, а решта 8 біт – нулі, що еквівалентно масці 255.255.255.0.

Префіксна нотація є зручнішою для запису та швидшого сприйняття, оскільки не потребує перетворення маски у вигляді десяткових чисел. Однак обидва способи вказують на те, яка частина адреси використовується для ідентифікації мережі, а яка для адресації пристроїв у цій мережі.

У класичному прикладі з адресою 192.168.10.10 і маскою 255.255.255.0 (що еквівалентно /24), перші 24 біти адреси позначають мережну частину, а останні 8 біт вузлову. У цій мережі можуть існувати 256 можливих *IP*-адрес (від 192.168.10.0 до 192.168.10.255), де перша адреса використовується як адреса мережі (192.168.10.0), а остання як широкомовна адреса (192.168.10.255).

Пристрої визначають мережну і вузлову частини *IPv4*-адреси за допомогою маски підмережі, використовуючи логічну операцію І (AND). Кожен біт *IPv4*-адреси порівнюється з відповідним бітом маски. Якщо біт маски дорівнює 1, то цей біт адреси належить до мережної частини. Якщо біт маски дорівнює 0, він належить до вузлової частини. Завдяки цьому пристрій може відокремити частину адреси, що відповідає мережі, від частини, яка ідентифікує конкретний вузол у цій мережі.

Наведемо приклад для *IPv4*-адреси 192.168.1.10 з маскою підмережі 255.255.255.0. Щоб визначити, яка частина цієї адреси є мережною, а яка вузловою, необхідно виконати операцію AND між бітами адреси та маски.

$$\begin{array}{rcl} 192.168.1.10 & = & 11000000.10101000.00000001.00001010 \\ & \text{AND} & \\ 255.255.255.0 & = & 11111111.11111111.11111111.00000000 \\ & = & \\ 192.168.1.0 & = & 11000000.10101000.00000001.00000000 \end{array}$$

Таким чином, для адреси 192.168.1.10 з маскою підмережі 255.255.255.0 мережна частина буде рівною 192.168.1.0, а вузлова частина буде рівною 10. Мережна частина (192.168.1.0) є спільною для всіх пристроїв у цій мережі, а вузлова частина змінюється для кожного пристрою, для його ідентифікації.

У будь-якій підмережі існують дві спеціальні адреси, що розміщуються на першій та останній позиції, які не можуть бути призначені окремим пристроям, але виконують інші важливі функції.

Перша адреса підмережі називається мережною адресою або адресою мережі. Вона використовується для ідентифікації всієї мережі і містить нулі в вузловій частині адреси. Наприклад, у мережі 192.168.1.0/24 мережна адреса – це 192.168.1.0. Вона не може бути присвоєна окремому пристрою, оскільки вказує на всю мережу загалом.

Остання адреса в мережі називається широкомовною адресою (*broadcast address*). Вона використовується для того, щоб відправляти повідомлення всім пристроям у цій мережі одночасно. Ця адреса має всі одиниці у вузловій частині. У прикладі з мережею 192.168.1.0/24 широкомовна адреса рівна 192.168.1.255. Вона також не може бути призначена окремому пристрою, бо призначена для широкомовної передачі.

IPv4-адреси поділяються на декілька типів, кожен з яких має свої особливості та використовується в різних контекстах. Публічні IPv4-адреси призначені для глобального маршрутизації та використовуються для взаємодії з інтернетом. Вони повинні бути унікальними в масштабах всього інтернету. Приватні адреси, навпаки, використовуються всередині організацій або локальних мереж і не можуть бути безпосередньо доступними через інтернет. Вони були введені для збереження обмеженого простору IPv4 і включають певні діапазони, які дозволяють організаціям призначати адреси своїм внутрішнім пристроям. Приватні адреси потребують перетворення в публічні через процес NAT.

Приватні адреси визначені в RFC 1918 і включають три основні діапазони:

- 10.0.0.0/8 – включає адреси від 10.0.0.0 до 10.255.255.255 та забезпечує значний простір для адресації, що підходить для великих організацій та мереж;
- 172.16.0.0/12 – охоплює адреси від 172.16.0.0 до 172.31.255.255, забезпечує менший простір для адресації в порівнянні з діапазоном 10.0.0.0/8, але все ще підходить для середніх та великих мереж;
- 192.168.0.0/16 – діапазон від 192.168.0.0 до 192.168.255.255, найбільш часто використовувані приватні адреси для малих і середніх домашніх і офісних мереж.

Додатково виділяють діапазони адрес для спеціального використання, зокрема:

- *Loopback* адреси – використовуються для тестування мережних функцій на локальному комп'ютері. Адреси в мережі 127.0.0.0/8, найчастіше 127.0.0.1, відправляють пакети назад до самого пристрою, що їх надіслав. Це дозволяє перевіряти працездатність стеку *TCP/IP* без фактичної передачі пакетів мережею;
- *Link-Local* адреси – адреси в мережі 169.254.0.0/16, також відомі як *Automatic Private IP Addressing (APIPA)*, автоматично призначаються пристроєм, коли *DHCP*-сервер не доступний. Вони використовуються для комунікації між пристроями в одній локальній мережі без потреби в централізованому управлінні адресами;
- *Multicast* адреси – діапазон від 224.0.0.0 до 239.255.255.255 використовується для *multicast*-комунікації, де один пакет надсилається групі пристроїв замість одного або всіх пристроїв у мережі. Це дозволяє ефективно передавати дані, такі як відео або аудіо стріми, що необхідно доставити до кількох одержувачів одночасно;
- Експериментальні адреси – діапазон від 240.0.0.0 до 255.255.255 зарезервований для експериментальних або майбутніх призначень. Ці адреси не використовуються в стандартних мережах і можуть бути призначені для нових технологій або функцій у майбутньому.

Розподіл мережі на підмережі, або сегментація мережі, є важливою практикою в управлінні мережами, що має кілька ключових цілей.

Сегментація мережі допомагає зменшити обсяг широкомовного трафіку. У великих мережах з численними пристроями широкий діапазон широкомовних пакетів може стати проблемою, оскільки всі пристрої в мережі отримують ці пакети, навіть якщо вони не мають до них відношення. Це може призвести до перевантаження мережі та уповільнення її роботи. Розділення мережі на менші підмережі дозволяє обмежити область поширення таких широкомовних пакетів лише тією частиною мережі, де вони дійсно потрібні.

Сегментація підвищує загальну продуктивність мережі. Коли мережа розділена на менші підмережі, кількість пристроїв, які можуть обробляти широкий діапазон трафіку, зменшується. Це дозволяє зменшити навантаження на окремі частини мережі та знизити затримки в передачі даних, що в цілому підвищує швидкість та ефективність мережі.

Сегментація забезпечує більший контроль над безпекою мережі. В окремих підмережах можна застосовувати специфічні правила безпеки, що обмежують доступ між різними частинами мережі. Це означає, що якщо в одній підмережі виникає проблема, наприклад, зловмисна атака або вірус, вона не зможе легко поширитися на інші підмережі, що допомагає захистити всю мережу.

Підмережі полегшують управління та організацію мережі. Адміністратори можуть організовувати підмережі за різними критеріями, такими як географічне

розташування, функціональні групи або типи пристроїв. Це дає змогу краще розподілити ресурси мережі та спростити адміністративні завдання, такі як налаштування адрес та політик доступу.

Сегментація мережі сприяє легшому усуненню неполадок. Коли мережа розділена на менші підмережі, виявлення та вирішення проблем з'єднання або конфігурації стає менш складним, оскільки проблеми обмежуються однією частиною мережі і не впливають на всю мережу.

Variable-Length Subnet Masking (VLSM) дозволяє ефективніше використовувати IPv4 адреси шляхом створення підмереж з різними розмірами, що оптимізує використання адресного простору.

Традиційне розділення мереж за класами, де використовується однакова маска підмережі для всіх підмереж, може бути неефективним, особливо коли мережа має різні вимоги до кількості вузлів у підмережах. Це може призвести до підвищених витрат адрес, оскільки деякі підмережі можуть мати значно більше адрес, ніж їм потрібно.

VLSM вирішує цю проблему, дозволяючи використовувати маски підмережі різної довжини для різних частин мережі. Це означає, що можна створювати підмережі з різними розмірами в одній і тій же адресній області. Можна мати великі підмережі для мереж, що потребують багато вузлів, і менші підмережі для тих, що потребують лише кількох адрес.

Процес *VLSM* полягає в наступному: спочатку визначаються найбільші підмережі, які потребують найбільше адрес. Для цих підмереж використовують найменше бітів у масці підмережі, залишаючи більше адрес для вузлів. Потім, залишивши менше адрес для менших підмереж, виділяють менше бітів для підмережі. Цей підхід дозволяє зменшити загальну витрату адрес і оптимізувати їх використання.

Наприклад, у випадку з адресою 192.168.20.0/24 ви можете спочатку створити великі підмережі, використовуючи маску /27, що забезпечує 30 використовуваних адрес у кожній підмережі. Для з'єднань між маршрутизаторами, які потребують лише 2 адреси, можна створити менші підмережі з маскою /30. Це дозволяє економно використовувати адреси, зменшуючи кількість адрес у підмережах, що не використовуються.

1.4.3 АДРЕСАЦІЯ IPv6

IPv4 має значні обмеження, головне з яких – обмежений простір адрес. При теоретичному максимумі в 4,3 мільярда унікальних IP-адрес, реальний попит значно перевищує цю кількість через глобальне поширення інтернету. Зростання інтернет-користувачів, особливо в Африці та Азії, збільшення кількості мобільних пристроїв і розвиток Інтернету речей (IoT) створили дефіцит IPv4-адрес, що спричинило необхідність у рішенні, яке надасть більше адресного простору.

Хоча приватні адреси та технології, як-то NAT (Network Address Translation), допомогли уповільнити виснаження IPv4-адрес, NAT має низку обмежень і створює труднощі для багатьох додатків. Він знижує продуктивність (через затримки) і негативно впливає на роботу однорангових з'єднань, ускладнюючи безпосередню комунікацію між пристроями в мережі.

IPv6 було розроблено для вирішення цих проблем. Замість 32-бітних адрес IPv4, IPv6 використовує 128-бітні адреси, що забезпечує неймовірно великий простір – 340 ундецильйонів (340 з 36 нулями) унікальних адрес.

Протоколи IPv4 і IPv6 можуть співіснувати, забезпечуючи поступовий перехід до повністю IPv6-сумісної мережі. Оскільки повний перехід на IPv6 займе багато часу, розроблено кілька технологій для одночасної роботи обох протоколів.

Одним із ключових способів є використання *Dual Stack*, що дозволяє пристроям одночасно підтримувати обидва протоколи. У такій системі мережні пристрої працюють із двома стеками протоколів, обробляючи як IPv4-трафік, так і IPv6-трафік. Це особливо корисно, коли мережа потребує з'єднання з використанням обох протоколів, оскільки деякі частини інтернету ще не підтримують IPv6. Завдяки *Dual Stack* можна поступово збільшувати використання IPv6, не втрачаючи можливості обробляти IPv4-запити.

Іншою технологією є тунелювання, за допомогою якого IPv6-пакети передаються через існуючу інфраструктуру IPv4. У цьому випадку IPv6-пакет інкапсулюється всередині IPv4-пакета, проходячи IPv4-мережею. Це дозволяє створити IPv6-з'єднання, навіть якщо мережа між відправником і отримувачем не повністю підтримує IPv6, адже тунелювання забезпечує транспортування пакетів безпосередньо до IPv6-вузлів.

Третій підхід – це трансляція, яку реалізують за допомогою NAT64. Ця технологія дозволяє пристроям, що працюють на IPv6, спілкуватися з пристроями на IPv4, навіть якщо між ними немає прямої підтримки обох протоколів. NAT64 перетворює IPv6-пакети на IPv4-пакети, дозволяючи обмін даними між різними мережами. Так, IPv6-адреси перетворюються на IPv4-адреси й навпаки, що забезпечує безперебійний зв'язок між різними сегментами мережі.

IPv6-адреса – це 128-бітна адреса, яка використовується для ідентифікації пристроїв у мережах, що підтримують цей протокол. Її записують у шістнадцятковому форматі, де кожен 16-бітний сегмент адреси називається "текстет" і складається з чотирьох шістнадцяткових цифр. Повна IPv6-адреса виглядає так: **2001:0db8:0000:0045:0000:0000:0000:1234**.

Через велику кількість цифр IPv6-адреси мають два основні правила скорочення, які полегшують запис і зменшують його довжину. Ці правила зберігають точність і унікальність адреси, але дозволяють скоротити деякі елементи.

Перше правило: опущення провідних нулів у текстетах. Якщо у 16-бітному сегменті адреси (текстеті) є нулі на початку, їх можна не писати.

Наприклад, гекстет `0db8` скорочується до `db8`, `0045` скорочується до `45`, а `0001` скорочується до `1`. Однак це правило стосується лише провідних нулів, тобто нулів, які стоять на початку сегменту, а не в кінці, оскільки це могло б призвести до плутанини між різними значеннями. Для прикладу, гекстет `0450` скоротиться до `450`, але `4500` залишиться без змін, оскільки відкидання кінцевих нулів може змінити значення.

Друге правило: використання подвійних двокрапок (`::`) для заміни послідовностей нульових гекстетів. Це правило дозволяє замінювати одну або більше послідовних груп із нульових гекстетів подвійними двокрапками. Наприклад, адреса `2001:0db8:0000:0000:0000:0000:0000:1234` може бути записана як `2001:db8::1234`, де `::` замінює п'ять нульових гекстетів. Подвійні двокрапки дозволено використовувати лише один раз в адресі, оскільки їхнє повторне застосування може призвести до неоднозначності при розгортанні. Тобто, якщо в адресі є дві або більше послідовності нульових гекстетів, скорочення подвійними двокрапками використовують для найдовшої з них, а якщо довжина послідовностей однакова, скорочення роблять для першої.

Приклади скорочення IPv6-адрес:

- повна адреса `2001:0db8:0000:0000:0000:0000:0000:0001` може бути записана як `2001:db8::1`. Тут усі п'ять гекстетів із нулями замінені подвійними двокрапками, а останній сегмент записаний як `1`.
- адреса `fe80:0000:0000:0000:0202:b3ff:fe1e:8329` скорочується до `fe80::202:b3ff:fe1e:8329`, де перші три нульові гексети замінені на `::`.
- спеціальна адреса `0000:0000:0000:0000:0000:0000:0000:0001`, яка позначає локальну адресу *localhost*, скорочується до `::1`.
- адреса `2001:0db8:0000:0000:0000:ff00:0042:8329` після застосування обох правил скоротиться до `2001:db8::ff00:42:8329`.

Важливо зазначити, що такі скорочення не змінюють значення адреси, а лише спрощують її запис, що особливо корисно для ручного введення та зручності зчитування.

IPv6-адреси поділяють на кілька основних типів, кожен із яких призначений для певних завдань у мережах. Основними категоріями є: індивідуальні (*unicast*), групові (*multicast*) і альтернативні (*anycast*) адреси. Індивідуальна (*unicast*) адреса унікально ідентифікує конкретний інтерфейс на пристрої, що підтримує IPv6, забезпечуючи зв'язок один-до-одного, тобто пакети, надіслані на індивідуальну адресу, досягають лише одного пристрою. Групова (*multicast*) адреса, навпаки, дозволяє надсилати один пакет відразу на декілька цільових пристроїв, що корисно для поширення повідомлень або спільного використання ресурсів у межах групи. Альтернативні адреси (*anycast*) є особливими унікальними індивідуальними адресами, що можуть бути присвоєні декільком

пристроєм: у цьому випадку пакети надсилаються на найближчий пристрій, що має таку адресу. Варто зазначити, що, на відміну від *IPv4*, у *IPv6* немає широкомовних (*broadcast*) адрес, а їхню функцію частково виконує групова адреса для всіх вузлів (*all-nodes multicast*).

В *IPv6*, як і в *IPv4*, мережну частину адреси можна визначити за допомогою довжини префікса, що зазначається у вигляді числа після косої риски (наприклад, /64). Довжина префікса показує кількість бітів, що належать до мережної частини *IPv6*-адреси. Для більшості мереж рекомендується довжина префіксу в 64 біти, оскільки вона дозволяє використовувати статичну конфігурацію адрес (*SLAAC*), спрощуючи структуру мережі і забезпечуючи достатню кількість вузлів.

IPv6-пристрій, як правило, має дві індивідуальні *IPv6*-адреси: глобальну індивідуальну адресу (*GUA* – *Global unicast address*) та локальну адресу каналу (*LLA* – *Link-local address*). Глобальна індивідуальна адреса є аналогом публічної *IPv4*-адреси і призначена для ідентифікації пристрою в Інтернеті; вона унікальна та може бути статичною або динамічною. Локальна адреса каналу використовується для зв'язку між пристроями в межах однієї мережі (каналу), і ця адреса не виходить за межі цієї мережі, оскільки маршрутизатори не пересилають пакети, адресовані на *LLA*.

Глобальні індивідуальні адреси мають три основні складові: префікс глобальної маршрутизації, ідентифікатор підмережі та ідентифікатор інтерфейсу. **Префікс глобальної маршрутизації** – це частина адреси, яка призначається постачальником інтернет-послуг (*ISP*) або іншим органом для ідентифікації мережі клієнта. Зазвичай префікс для клієнтів має довжину /48, що дозволяє *ISP* ідентифікувати певну мережу.

Ідентифікатор підмережі (*Subnet ID*) дає змогу організації розподілити свою мережу на окремі підмережі.

Ідентифікатор інтерфейсу визначає конкретний пристрій в підмережі, і, як правило, займає 64 біти. Це дозволяє використовувати механізм *SLAAC*, за якого пристрої автоматично формують свій ідентифікатор інтерфейсу на основі *MAC*-адреси або інших унікальних значень.

Локальні адреси каналу обов'язкові для всіх *IPv6*-пристроїв і призначені для зв'язку в межах однієї локальної мережі (каналу). Адреси *LLA* починаються з префіксу *fe80::/10*, і можуть бути створені автоматично, що дозволяє пристроям на одному каналі обмінюватися даними без попередньої конфігурації мережі.

Розподіл мережі *IPv6* на підмережі значно простіший і більш гнучкий, ніж в *IPv4*, оскільки *IPv6* спочатку був спроектований з урахуванням необхідності розподілу мереж на підмережі. У структурі *IPv6*-адреси передбачено спеціальне поле для ідентифікації підмереж – *Subnet ID* (рисунк 1.25). Це поле знаходиться між префіксом глобальної маршрутизації і ідентифікатором інтерфейсу, що дозволяє безпосередньо створювати підмережі, не змінюючи інші частини адреси.

48 біт	16 біт	64 біти
Префікс глобальної маршрутизації	Ідентифікатор підмережі	Ідентифікатор інтерфейсу
Префікс маршрутизації /48 + 16-бітний ідентифікатор підмережі = /64 префікс		

Рисунок 1.25 – Структура IPv6-адреси

Наприклад, якщо організація отримала префікс глобальної маршрутизації 2001:db8:acad::/48, то маючи 48-бітний префікс мережі і стандартний 64-бітний ідентифікатор інтерфейсу, залишається 16 бітів для підмереж. Це означає, що організація може створити до 65536 підмереж із префіксом /64. Кожна така підмережа зможе мати 64-бітний ідентифікатор інтерфейсу, що дозволяє підтримувати до 18 квінтільйонів пристроїв у кожній підмережі. Наприклад, підмережа з ідентифікатором 0001 буде представлена як 2001:db8:acad:1::/64, підмережа з ідентифікатором 0002 – 2001:db8:acad:2::/64 і так далі. Таке розширення підмережі простим додаванням нового значення в 16-бітному полі не вимагає перетворень у двійкову форму, а лише використання шістнадцяткової системи, що значно спрощує адміністрування мереж.

Розподіл адрес для підмереж в IPv6 дозволяє гнучко планувати структуру мережі без обмежень, які характерні для IPv4, оскільки проблема економії адрес у IPv6 практично не є актуальною. Наприклад, у мережній топології з п'ятьма підмережами можна призначити такі префікси: 2001:db8:acad:1::/64, 2001:db8:acad:2::/64 і так далі до 2001:db8:acad:5::/64. Це дозволяє задовольнити потреби великої кількості пристроїв у кожній підмережі, залишаючи простір для подальшого масштабування.

Процес налаштування маршрутизаторів в IPv6 для кожної підмережі також є простішим, адже кожен інтерфейс може бути налаштований з відповідним префіксом підмережі без змін у загальній структурі мережі. Наприклад, інтерфейсу маршрутизатора можна присвоїти адресу 2001:db8:acad:1::1/64 для підмережі з ідентифікатором 0001, 2001:db8:acad:2::1/64 для підмережі з ідентифікатором 0002 і так далі.

1.4.4 ПРОТОКОЛ ICMP

ICMP (Internet Control Message Protocol) є важливим компонентом протоколу *IP (Internet Protocol)* і використовується для передавання контрольних і помилкових повідомлень між мережними пристроями. Метою *ICMP* є надання зворотного зв'язку щодо стану мережі, що дозволяє діагностувати проблеми в процесі передачі даних між пристроями. Протокол не призначений для передачі користувачьких даних, а лише для обміну контрольними повідомленнями, що допомагають визначати, чи виникають проблеми з мережею.

Однією з основних функцій *ICMP* є повідомлення про помилки, коли передача *IP*-пакета не може бути виконана, а також інформування про інші

проблеми, які можуть виникнути при маршрутизації. Однак *ICMP* не гарантує, що самі пакети будуть доставлені успішно; він лише сповіщає про невдачі та дає вказівки щодо того, що сталося.

Один з найвідоміших аспектів використання *ICMP* – це обмін повідомленнями *Echo Request* та *Echo Reply*, які є основою для утиліти **ping**. Ця утиліта дозволяє перевіряти доступність вузла в мережі. Коли комп'ютер відправляє *ICMP Echo Request* до іншого вузла, той, якщо доступний, відповідає *ICMP Echo Reply*, і користувач отримує інформацію про час затримки та наявність проблем у мережі. Якщо вузол не відповідає, це може вказувати на проблему з його доступністю, неполадки в мережі чи блокування *ICMP*-пакетів міжмережним екраном.

```
PS C:\Users\User> ping 8.8.8.8
```

```
Pinging 8.8.8.8 with 32 bytes of data:
```

```
Reply from 8.8.8.8: bytes=32 time=17ms TTL=118
```

```
Reply from 8.8.8.8: bytes=32 time=17ms TTL=118
```

```
Reply from 8.8.8.8: bytes=32 time=17ms TTL=118
```

```
Reply from 8.8.8.8: bytes=32 time=17ms TTL=118
```

```
Ping statistics for 8.8.8.8:
```

```
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
Minimum = 17ms. Maximum = 17ms, Average = 17ms
```

Рисунок 1.26 – Використання протоколу *ICMP*

ICMP також надає механізм для повідомлення про інші помилки в процесі маршрутизації. Наприклад, коли маршрутизатор не може доставити пакет до його цільової адреси, він може надіслати повідомлення *Destination Unreachable*, яке вказує на причину невдачі, наприклад, недоступність мережі, вузла або порту. Це дає змогу адміністратору з'ясувати, чому пакет не може бути доставлений, і спробувати вжити відповідних заходів для виправлення ситуації.

```
PS C:\Users\User> ping 192.168.1.123
```

```
Pinging 192.168.1.123 with 32 bytes of data:
```

```
Reply from 192.168.1.100: Destination host unreachable.
```

```
Reply from 192.168.1.100: Destination host unreachable.
```

```
Reply from 192.168.1.100: Destination host unreachable.
```

```
Reply from 192.168.1.100: Destination host unreachable.
```

```
Ping statistics for 192.168.1.123:
```

```
Packets: Sent = 4 Received = 4 Lost = 0 (0% loss),
```

Рисунок 1.27 – Приклад повідомлення *Destination Unreachable*

Іншим важливим повідомленням *ICMP* є *Time Exceeded*, яке використовується для повідомлення про те, що час життя (*TTL*) пакета вичерпався до того, як пакет досягнув своєї цілі. Це повідомлення часто використовується в застосунку *traceroute*, який дозволяє відстежити шлях пакета мережею.

```
PS C:\Users\User> ping 2.3.4.5

Pinging 2.3.4.5 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 2.3.4.5:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Рисунок 1.28 – Приклад повідомлення *Time Exceeded*

За допомогою *traceroute* можна визначити, які маршрутизатори обробляють пакет на його шляху до кінцевої точки, а також де виникають затримки або проблеми з доставкою.

ICMP є важливим інструментом для діагностики проблем у мережах, оскільки він дозволяє визначити, де і які саме проблеми виникають у процесі передачі даних. Утиліти, такі як **ping** та **traceroute**, використовують *ICMP* для надання адміністратору мережі інструментів для перевірки доступності вузлів та відстеження шляхів даних у мережі. Ці інструменти дозволяють не лише перевіряти доступність окремих пристроїв, але й діагностувати мережні проблеми, визначаючи, чи є які-небудь затримки або неполадки на проміжних маршрутизаторах.

ICMPv6 (*Internet Control Message Protocol for IPv6*) є важливою частиною протоколу *IPv6*, аналогічною *ICMP* для *IPv4*, але з рядом додаткових функцій і покращень, що відповідають потребам нового покоління протоколів. Як і *ICMP* для *IPv4*, *ICMPv6* використовується для передачі повідомлень про помилки та інформаційних повідомлень, що допомагають у діагностиці та управлінні мережами *IPv6*.

Основні функції *ICMPv6* включають оброблення повідомлень про недосяжність вузлів і мереж, сповіщення про помилки маршрутизації, а також важливі нові функції, пов'язані з підтримкою *IPv6*, такі як Автоматична адресація та Виявлення суміжних вузлів.

Основні типи повідомлень *ICMPv6* включають *Echo Request* і *Echo Reply*, *Destination Unreachable*, *Time Exceeded*, *Redirect*.

***Echo Request* і *Echo Reply*:** Як і в *ICMP* для *IPv4*, *ICMPv6* використовує типи повідомлень *Echo Request* (запит на відгук) і *Echo Reply* (відповідь на запит) для тестування досяжності вузлів у мережі. Це найпоширеніше використання

ICMPv6 і є основою для застосунків, як-то **ping**. Ці повідомлення використовуються для перевірки, чи доступний певний вузол в IPv6-мережі, що дозволяє діагностувати базові мережні проблеми.

Destination Unreachable (неможливо досягнути призначення): Це повідомлення відправляється, коли маршрутизатор або вузол не може доставити пакет до кінцевого пункту призначення.

Time Exceeded (час вичерпано): Цей тип повідомлення сигналізує, що пакет не може бути доставлений через те, що час життя (*TTL*) або ліміт стрибків (*Hop Limit*) пакета вичерпався. В IPv6 це є аналогом того, як *TTL* працює в IPv4. Зазвичай використовується для визначення маршрутів за допомогою інструментів, таких як **traceroute**. Якщо пакет подорожує через декілька маршрутизаторів і *TTL* або *Hop Limit* досягає нуля, маршрутизатор відправляє повідомлення «*Time Exceeded*».

Однією з найважливіших функцій ICMPv6 є підтримка *Neighbor Discovery Protocol (NDP)*, який виконує функції, схожі на *ARP* в IPv4. *NDP* є критичним для виявлення пристроїв в мережі, вирішення проблем з адресами та налаштування мережі. Він включає декілька типів повідомлень:

- **Router Solicitation (RS)**: вузли використовують ці повідомлення для запиту інформації від маршрутизаторів про мережі, зокрема для отримання інформації про адреси.
- **Router Advertisement (RA)**: маршрутизатори надсилають ці повідомлення, щоб повідомити вузли про свою наявність у мережі та надати інформацію для автоконфігурації адрес.
- **Neighbor Solicitation (NS)**: повідомлення використовуються для перевірки унікальності адрес у мережі, а також для визначення фізичних адрес (MAC) на основі IPv6-адрес.
- **Neighbor Advertisement (NA)**: Відповідь на Neighbor Solicitation, що містить MAC-адресу пристрою.

1.5 ТРАНСПОРТНИЙ РІВЕНЬ

Транспортний рівень є одним із ключових елементів мережної моделі, що забезпечує надійну передачу даних між пристроями. Роль транспортного рівня полягає у забезпеченні надійної та ефективної передачі даних між додатками, що працюють на різних пристроях у мережі. Він відповідає за логічне з'єднання між цими додатками, навіть якщо фізичне з'єднання проходить через численні мережні вузли, сегменти та інші рівні мережної архітектури. Завдяки транспортному рівню дані, які генерує додаток на одному пристрої, можуть бути передані на інший пристрій, де інший додаток зможе отримати, обробити та представити їх користувачеві або іншій системі.

Щоб забезпечити передавання даних належним чином, транспортний рівень виконує кілька важливих функцій. Він відстежує всі обміни між додатками, що одночасно можуть відбуватися на одному пристрої, таким чином підтримуючи унікальність кожного потоку даних і гарантуючи, що інформація надходить до відповідного додатку. Для цього транспортний рівень використовує номери портів як ідентифікатори для розмежування різних потоків даних. Це дозволяє декільком додаткам на одному пристрої обмінюватися даними без конфліктів, оскільки кожен додаток отримує свій власний порт.

Транспортний рівень також розподіляє дані на менші блоки, які можуть називатися сегментами (для *TCP*) або дейтаграмами (для *UDP*). У випадку з протоколом *TCP* транспортний рівень додає до кожного сегмента інформацію про послідовність, контрольну суму та інші параметри, що дозволяють перевіряти та відновлювати дані у разі помилок. Завдяки цьому *TCP* стає надійним протоколом з механізмами підтвердження отримання даних, повторного відправлення втрачених сегментів та контролю потоку даних, щоб уникати перевантаження мережі.

З іншого боку, *UDP*, як більш «легкий» протокол, не забезпечує жорсткої перевірки доставки даних та не встановлює з'єднання. Це робить його швидшим і менш вимогливим до ресурсів, проте й менш надійним, тому його використовують для застосунків, де важлива швидкість передачі даних без зволікань, таких як потокове аудіо або відео.

TCP (Transmission Control Protocol) є одним із основних протоколів транспортного рівня, який надає надійність і контроль потоку даних між вузлами мережі. Розробники можуть вибирати між *TCP* та *UDP* залежно від вимог до передачі даних, але *TCP* особливо корисний, коли необхідно забезпечити надійність, точність та послідовність доставки.

TCP має ряд ключових характеристик, що відрізняють його від інших протоколів. Перед початком передачі даних *TCP* встановлює з'єднання між джерелом і призначенням, що дозволяє вузлам домовитися про умови передачі й кількість даних, які можуть передаватися в одиницю часу. Це відрізняє *TCP* як протокол, орієнтований на з'єднання, завдяки чому можна уникнути втрат та затримок в обміні.

Ще однією важливою характеристикою *TCP* є надійність. Під час передачі мережею пакети даних можуть загубитися або бути пошкодженими, але *TCP* забезпечує, що кожен відправлений сегмент досягне призначення, використовуючи механізм підтвердження та повторної відправки втрачених сегментів. Крім того, *TCP* підтримує отримання даних у правильному порядку, що є критично важливим для застосунків. Завдяки нумерації *TCP* забезпечує коректне збирання сегментів на стороні отримувача.

TCP також реалізує механізм керування потоком, який регулює швидкість передачі даних. Якщо ресурсів приймаючого вузла недостатньо для обробки всіх

даних, *TCP* може надіслати сигнал для уповільнення передачі, запобігаючи втра-там і зменшуючи потребу в повторних запитах.

Формат *TCP*-сегмента складається з заголовка, який містить важливу службову інформацію для управління передачею даних, і самої частини даних. Заголовок *TCP*-сегмента включає ряд полів (рисунок 1.29), кожне з яких виконує певну функцію для забезпечення надійності, послідовності і контролю передачі.

1. **Порт джерела (*Source Port*):** 16-бітне поле, що містить номер порту відправника. Використовується для ідентифікації застосунку, який ініціював з'єднання.

2. **Порт призначення (*Destination Port*):** 16-бітне поле для ідентифікації порту отримувача. Це поле вказує, якому застосунку на приймаючій стороні призначені дані.

3. **Порядковий номер (*Sequence Number*):** 32-бітне поле, яке використовується для відновлення послідовності надходження даних. Якщо сегмент є частиною довшого потоку, це поле вказує на позицію цього сегмента у всьому потоці. Це поле дозволяє *TCP* забезпечити доставку даних у правильному порядку.

4. **Номер підтвердження (*Acknowledgment Number*):** 32-бітне поле, яке використовується для підтвердження отримання попередніх сегментів.

5. **Довжина заголовку (*Header Length*):** 4-бітне поле, відоме також як «зсув даних» (*Data Offset*), яке вказує довжину заголовка *TCP*. Це дозволяє отримувачу визначити, де починаються власне дані.

6. **Зарезервовано (*Reserved*):** 6-бітне поле, зарезервоване для майбутнього використання.

7. **Контрольні біти (*Control Flags*):** 6-бітне поле з прапорами, що визначають певні функції *TCP*-сегмента.

8. **Розмір вікна (*Window Size*):** 16-бітне поле, що визначає кількість байтів, яку відправник готовий прийняти без підтвердження. Це важливе поле для управління потоком даних, яке дозволяє збалансувати швидкість передачі даних.

9. **Контрольна сума (*Checksum*):** 16-бітне поле, яке використовується для перевірки цілісності заголовка і даних *TCP*-сегмента. Відправник обчислює контрольну суму і вказує її в цьому полі, а отримувач використовує її для перевірки, чи були дані пошкоджені під час передачі.

10. **Показчик терміновості (*Urgent Pointer*):** 16-бітне поле, яке використовується як ознака того, що сегмент містить термінові дані. Воно дозволяє сигналізувати про термінову інформацію, яку слід обробити відразу.

11. **Параметри (*Options*):** поле змінної довжини, яке може містити додаткову інформацію для налаштування параметрів *TCP*-з'єднання. Наприклад, тут можуть бути налаштування максимального розміру сегмента (*MSS*) або ж часові мітки для покращення продуктивності.

12. **Дані прикладного рівня (*Data*):** після заголовка йде секція з власне даними, які передаються від одного застосунку до іншого. Розмір даних може варіюватися залежно від специфічних параметрів з'єднання і вимог до мережі.

Порт джерела (16)			Порт призначення (16)		20 байт
Порядковий номер (32)					
Номер підтвердження (32)					
Довжина заголовку (4)	Зарезервовано (6)	Контрольні біти (6)	Розмір вікна (16)		
Контрольна сума (16)			Показчик терміновості (16)		
Параметри (0 або 32, якщо такі є)					
Дані прикладного рівня (змінного розміру)					

Рисунок 1.29 – Структура заголовку *TCP*

TCP активно використовується додатками, де надійність передачі є критичною. Наприклад, веббраузери, електронна пошта, бази даних та інші сервіси, які потребують повної та точної передачі даних, використовують *TCP*. Цей протокол знімає із застосунків завдання контролю послідовності та надійності, даючи можливість розробникам зосередитися на функціоналі без огляду на мережні аспекти роботи.

На відміну від *TCP*, *UDP (User Datagram Protocol)* – це простий і легкий протокол транспортного рівня, який працює за принципом «best-effort» (тобто не гарантує доставку даних і не забезпечує контроль потоку). *UDP* призначений для застосунків, де важлива швидкість передачі, а не надійність або порядок доставки, що робить його ідеальним для додатків, чутливих до затримок, як-от *VoIP* та потокове відео.

Основні характеристики *UDP* включають відсутність встановлення з'єднання, відсутність відновлення втрачених пакетів і непідтверджену доставку. Це означає, що *UDP* не створює з'єднання між клієнтом і сервером, не перевіряє наявність ресурсів перед відправкою даних і не відстежує статус передачі. Якщо певні дані втрачаються, вони не надсилаються повторно. Весь контроль надійності покладається на сам застосунок.

Формат заголовка *UDP* є дуже простим (рисунок 1.30): містить лише 4 поля загальною довжиною 8 байтів (64 біти):

1. **Порт джерела (*Source Port*):** 16-бітне поле для ідентифікації застосунку, який ініціював передачу даних.

2. **Порт призначення (*Destination Port*):** 16-бітне поле для ідентифікації застосунку, якому призначені дані на приймаючій стороні.

3. **Довжина (*Length*):** 16-бітне поле, яке вказує загальну довжину заголовка і даних *UDP*-дейтаграми.

4. **Контрольна сума (*Checksum*):** 16-бітне поле для перевірки цілісності заголовка і даних.

UDP не слід використовувати в додатках, які вимагають надійної доставки даних. Застосунки, що використовують *UDP*, або можуть допускати втрату даних, наприклад *VoIP* і потокове відео, або самостійно забезпечують необхідні функції надійності, наприклад, *DNS* або *TFTP*.

Порт джерела (16)	Порт призначення (16)	8 байт
Довжина (16)	Контрольна сума (16)	
Дані прикладного рівня (змінного розміру)		

Рисунок 1.30 – Структура заголовка *UDP*

Порти – це ключовий елемент транспортного рівня моделі *TCP/IP*, який використовується для ідентифікації програмних процесів на кінцевих пристроях. Порт дозволяє комп'ютеру одночасно обробляти декілька мережних з'єднань, забезпечуючи ідентифікацію кожного з них.

Кожне з'єднання має два порти: **порт джерела** (для програми, яка ініціює з'єднання) і **порт призначення** (для програми, яка приймає з'єднання). Наприклад, при запиті вебсторінки джерело створює динамічний порт для кожного запиту, тоді як порт призначення зазвичай статичний (наприклад, 80 для *HTTP* або 443 для *HTTPS*). Це дозволяє одночасно обробляти декілька запитів до одного й того ж сервера.

Сокетом називається комбінація *IP*-адреси та номера порту, що унікально ідентифікує кожне з'єднання. Наприклад, клієнтський сокет може виглядати як «192.168.1.5:51099», а серверний – як «192.168.1.7:80». Разом вони утворюють пару сокетів («192.168.1.5:51099», «192.168.1.7:80»), яка дозволяє відстежувати, яка саме програма веде з'єднання.

Порти поділяють на три групи:

- **Відомі порти (0-1023):** зарезервовані для стандартних протоколів та сервісів, наприклад таких як *HTTP* (80), *FTP* (21) і *DNS* (53).
- **Зареєстровані порти (1024-49151):** виділяються *IANA* для специфічних програм чи сервісів (наприклад, *RADIUS* використовує порт 1812).
- **Динамічні/приватні порти (49152-65535):** використовуються операційною системою для тимчасових сокетів.

Для моніторингу активних з'єднань на комп'ютері використовується утиліта *netstat*, яка відображає протоколи, локальні та віддалені адреси і порти, а також стан з'єднань. Наприклад, запит «netstat» надає список усіх поточних з'єднань, їх *IP*-адреси та порти, а опція «-n» дозволяє переглядати їх у числовому форматі.

Процес зв'язку в *TCP* є основою для забезпечення надійності та керованості передачі даних між клієнтом і сервером. *TCP* забезпечує контроль за встановленням, підтримкою та завершенням з'єднань через використання таких механізмів: «тристороннє рукошлякування», порти, сокети та контрольні прапорці.

TCP є протоколом із збереженням стану. Це означає, що до початку передавання даних *TCP* створює сесію, використовуючи механізм «тристороннього рукостискання». Цей процес синхронізує обидві сторони (клієнт і сервер) та перевіряє, чи готові вони до передачі даних.

«Тристороннє рукостискання» складається з наступних кроків:

SYN (synchronize): Клієнт ініціює з'єднання, надсилаючи серверу сегмент із прапорцем *SYN*. У цьому запиті клієнт передає своє **початкове значення номеру послідовності (Sequence Number, SEQ)**, щоб сервер знав, із якого моменту слід починати відлік отриманих даних.

SYN-ACK (synchronize + acknowledgment): Сервер відповідає, надсилаючи клієнту сегмент із прапорцями *SYN* та *ACK*. Прапорець *SYN* повідомляє, що сервер готовий до з'єднання. Прапорець *ACK* підтверджує отримання початкового запиту клієнта. Сервер також надсилає своє початкове значення *SEQ*.

ACK (acknowledgment): Клієнт відповідає серверу, надсилаючи сегмент із прапорцем *ACK*, підтверджуючи отримання відповіді сервера. На цьому етапі з'єднання встановлено, і обидві сторони можуть починати передавання даних.

Після встановлення з'єднання дані передаються сегментами, кожен із яких маркується унікальним номером послідовності (*SEQ*). Отримувач підтверджує кожен сегмент, надсилаючи *ACK* із номером, що вказує, які дані він успішно отримав.

Механізми *TCP* гарантують:

- **Повторну передачу загублених сегментів:** Якщо отримувач не підтверджує отримання певного сегмента, відправник повторно надсилає його.
- **Керування потоком:** Використовуючи механізм ковзного вікна, *TCP* регулює обсяг даних, які можна передати, перш ніж чекати підтвердження. Це дозволяє уникати перевантаження мережі.

Коли клієнт або сервер хоче завершити передачу даних, вони ініціюють процес закриття з'єднання, який потребує чотирьох кроків:

- **FIN (finish):** Одна сторона (наприклад, клієнт) надсилає сегмент із прапорцем *FIN*, щоб повідомити, що більше даних передаватися не буде.
- **ACK:** Друга сторона (сервер) відповідає сегментом із прапорцем *ACK*, підтверджуючи отримання запиту на завершення.
- **FIN:** Сервер надсилає клієнту свій сегмент із прапорцем *FIN*, щоб завершити власний потік передачі даних.
- **ACK:** Клієнт відповідає серверу, надсилаючи сегмент із прапорцем *ACK*, після чого з'єднання вважається закритим.

Після цього процесу сесія повністю закрыта, і всі ресурси, пов'язані із цим з'єднанням, звільнені.

Гарантована та впорядкована доставка в *TCP* забезпечується за допомогою механізмів нумерації байтів, контрольних підтверджень і буферизації на стороні отримувача.

Під час встановлення *TCP*-з'єднання відправник і отримувач узгоджують початковий номер послідовності (***Initial Sequence Number, ISN***), який обирається випадковим чином для кожної сесії. Цей номер визначає перший байт даних у потоці.

Кожен *TCP*-сегмент містить номер першого байту, який передається в цьому сегменті. У процесі передачі номери послідовності збільшуються відповідно до кількості байтів у кожному сегменті.

Наприклад:

- Якщо $ISN = 1$, то перший сегмент матиме номер 1.
- Якщо сегмент містить 100 байтів, наступний сегмент почнеться з номера 101.

Ця нумерація дозволяє отримувачу ідентифікувати порядок байтів у потоці.

Отримувач підтверджує отримання даних. Він надсилає номер наступного байту, який очікує отримати. Наприклад, якщо отримувач отримав дані з номерами від 1 до 100, він відповідає підтвердженням $ACK = 101$, що означає, що наступним він очікує байт 101. Це гарантує, що усі попередні байти успішно отримані та відправник може продовжувати передавати нові дані.

Отримувач використовує буфер для тимчасового зберігання даних, які надійшли невпорядковано. Якщо сегмент із номером послідовності не прибув, усі наступні сегменти, що надходять, зберігаються у буфері. Коли відсутній сегмент нарешті прибуває, дані впорядковуються відповідно до номерів послідовностей, після чого передаються на прикладний рівень.

Якщо відправник не отримує підтвердження від отримувача протягом заданого часу, він повторно надсилає втрачені сегменти. У сучасних реалізаціях *TCP* використовується механізм вибіркового підтвердження (***Selective Acknowledgment, SACK***), що дозволяє отримувачу повідомляти, які сегменти вже отримані, навіть якщо деякі з них пропущені.

Наприклад, якщо сегменти з номерами 1-2 і 5-10 отримані, але 3-4 відсутні, отримувач надсилає $ACK = 3$ (очікує сегмент 3) та $SACK = 5-10$, щоб вказати, що сегменти 5-10 вже отримані. Це дозволяє відправнику повторно надіслати лише сегменти 3-4.

Розмір вікна *TCP* визначає, скільки байтів даних відправник може передати без отримання підтвердження від отримувача. Під час встановлення з'єднання узгоджується початковий розмір вікна, який отримувач може змінювати в процесі передачі, залежно від доступності свого буфера. Цей механізм дозволяє динамічно регулювати обсяг переданих даних, уникати перевантаження отримувача та забезпечувати ефективність передачі. Вікно «ковзає» вперед, коли надходять підтвердження про отримані дані, що дозволяє відправнику продовжувати передавати нові байти без затримки.

Наприклад, отримувач встановив розмір вікна в 10000 байт. Відправник починає передачу з байту номером 1 і може відправити дані до байту номер

10000, чекаючи підтвердження. Після отримання підтвердження, наприклад, для байту номер 2921 (наступний очікуваний байт), вікно «ковзає» вперед, дозволяючи передавати дані до байту номер 12920. Це забезпечує безперервну передачу, поки отримувач обробляє та підтверджує прийом даних.

На відміну від *TCP*, *UDP* не використовує номерів сегментів, тому дейтаграми можуть надходити до отримувача у довільному порядку. *UDP* не намагається перевпорядкувати дейтаграми чи забезпечити їхню повторну передачу, якщо вони втрачені. Якщо порядок доставки критично важливий, його відстеження має виконувати сама прикладна програма, яка обробляє отримані дані.

Для роботи з портами *UDP*, як і *TCP*, використовує поняття порту джерела та призначення. Серверні програми мають визначені порти, які можуть бути «відомими» (наприклад, порт 53 для *DNS*) або «zareєстрованими» (порт 1812 для *RADIUS*). Клієнт при відправленні запиту до сервера динамічно обирає вихідний порт зі свого доступного діапазону. Сервер, отримуючи запит, відповідає, змінюючи місцями порт джерела та порт призначення.

Наприклад, клієнт 1 відправляє *DNS*-запит із вихідного порту 49152 до порту 53 сервера. Сервер обробляє запит і відправляє відповідь, використовуючи порт 53 як вихідний і порт 49152 клієнта як порт призначення. Аналогічно, клієнт 2 запитує автентифікацію через *RADIUS*, використовуючи вихідний порт 51152 і порт призначення 1812, а сервер відповідає, повертаючи дані на порт 51152. Цей простий механізм дозволяє здійснювати передавання даних без значних затримок і складності управління.

1.6 ПРИКЛАДНИЙ РІВЕНЬ

Основна мета прикладного рівня полягає в забезпеченні взаємодії між програмами, які використовують мережу, і мережними технологіями, щоб дані були передані в форматі, зрозумілому для пристрою-отримувача, та доступними для кінцевого користувача.

Прикладний рівень є найближчим до користувача та надає доступ до мережних послуг. Його протоколи визначають, як саме програми на пристроях-джерелах і отримувачах повинні взаємодіяти. Серед відомих протоколів – *HTTP*, *FTP*, *TFTP*, *DNS*, *SMTP*, *POP3*, *IMAP*. Вони забезпечують різноманітні функції: від доступу до вебресурсів до передачі файлів чи електронної пошти.

Рівень подання даних відповідає за форматування даних у формат, сумісний із пристроєм-отримувачем, стиснення та розпакування даних, шифрування для безпечної передачі та розшифрування на отримувачі тощо.

Сеансовий рівень встановлює, підтримує та завершує сеанси зв'язку між додатками. Його функції включають управління діалогами, відновлення роботи після перерв і підтримку активності сеансів.

Для успішної комунікації протоколи прикладного рівня повинні бути сумісними на пристроях-джерелах і отримувачах. Наприклад, *HTTP* визначає

правила передачі вебданих, а *FTP* – правила для передачі файлів. Деякі протоколи, наприклад *HTTPS*, забезпечують додаткову безпеку за рахунок шифрування. Основні з протоколів прикладного рівня показано в таблиці 1.1 разом із їх короткою характеристикою.

Розглянемо деякі з протоколів прикладного рівня.

HTTP є основним протоколом для передачі вебсторінок між клієнтом (веббраузером) і сервером. Коли користувач вводить *URL*-адресу в браузері, виконується декілька етапів: браузер аналізує адресу, звертається до *DNS*-сервера для отримання *IP*-адреси вебсервера, надсилає *HTTP*-запит (наприклад, *GET*-запит для завантаження сторінки), а сервер відповідає, надсилаючи *HTML*-код сторінки. Браузер інтерпретує отриманий *HTML* код і відображає вебсторінку. *HTTP* працює за принципом «запит-відповідь» із різними типами запитів, наприклад *GET* (отримання даних), *POST* (завантаження даних на сервер), *PUT* (оновлення ресурсів) тощо.

Таблиця 1.1 – Приклади протоколів прикладного рівня

Протокол	Порт(и)	Характеристика
DNS	TCP, UDP 53	Перетворює доменні імена (наприклад, example.com) у <i>IP</i> -адреси.
BOOTP	UDP 67, 68	Дозволяє бездисківій робочій станції отримувати <i>IP</i> -адресу та дані для завантаження.
DHCP	UDP 67, 68	Динамічно призначає <i>IP</i> -адреси з можливістю повторного використання.
SMTP	TCP 25	Забезпечує відправку електронної пошти клієнтами та між серверами.
POP3	TCP 110	Дає змогу клієнтам отримувати електронну пошту з сервера, завантажуючи її локально.
IMAP	TCP 143	Забезпечує доступ до електронної пошти, залишаючи її на сервері.
FTP	TCP 20, 21	Дозволяє надійно передавати файли між вузлами мережею.
TFTP	UDP 69	Проста передача файлів без встановлення з'єднання, менш надійна, але з меншими витратами.
HTTP	TCP 80, 8080	Правила для обміну текстом, зображеннями, відео та іншими мультимедійними файлами у вебсередовищі.
HTTPS	TCP, UDP 443	Захищений <i>HTTP</i> із шифруванням та автентифікацією вебсайтів.

HTTPS є безпечною версією **HTTP**. Він використовує шифрування **TLS/SSL** для захисту переданих даних, забезпечуючи конфіденційність і захист від перехоплення.

Електронна пошта використовує три основні протоколи: **SMTP**, **POP** та **IMAP**.

SMTP (*Simple Mail Transfer Protocol*) відповідає за надсилання листів. Клієнт **SMTP** встановлює з'єднання з сервером **SMTP** на порті 25 і передає повідомлення. Якщо одержувач недоступний, **SMTP** зберігає повідомлення в черзі для повторної спроби доставки.

POP (*Post Office Protocol*) дозволяє завантажувати листи з сервера на клієнт. Після завантаження листи можуть видалятися із сервера. **POP** працює на порті 110 і найчастіше використовується у версії **POP3**.

IMAP (*Internet Message Access Protocol*) забезпечує доступ до листів, залишаючи їх копії на сервері. Це дозволяє синхронізувати повідомлення між кількома пристроями та створювати ієрархії папок. **IMAP** працює на портах 143 (незашифрований) і 993 (зашифрований).

Протоколи **DNS** (*Domain Name System*) та **DHCP** (*Dynamic Host Configuration Protocol*) є ключовими компонентами мережної інфраструктури. Вони спрощують процеси ідентифікації пристроїв у мережі та автоматизацію їх налаштування.

DNS – це система, яка забезпечує перетворення зрозумілих користувачам доменних імен, як-то **www.example.com**, на числові **IP**-адреси, необхідні для передачі даних у мережах. Завдяки цьому користувачі можуть звертатися до ресурсів за іменами, а не запам'ятовувати складні числові адреси.

Процес роботи **DNS** складається з наступних кроків:

1. Користувач вводить **URL**, браузер формує **DNS**-запит, щоб знайти відповідну **IP**-адресу. Запит надсилається до локального **DNS**-сервера.
2. Локальний **DNS**-сервер шукає відповідність між доменним ім'ям та **IP**-адресою у своїй базі даних. Якщо інформація відсутня, сервер перенаправляє запит до інших **DNS**-серверів у системі.
3. Після отримання відповідної адреси сервер надсилає її клієнту, і клієнт використовує цю адресу для запиту до цільового сервера.

DNS використовує ієрархічну структуру, що складається з доменів і піддоменів. На вершині ієрархії знаходяться кореневі домени, за ними – домени верхнього рівня (**.com**, **.org**, **.ua**).

Для зберігання інформації про доменні імена використовуються ресурсні записи різного типу. До основних типів ресурсних записів у **DNS** належать:

- **A** – **IPv4**-адреса кінцевого пристрою.
- **AAAA** – **IPv6**-адреса кінцевого пристрою.
- **NS** – адреса авторитетного **DNS**-сервера.
- **MX** – записи поштового сервера для обробки електронної пошти.

DNS-сервери можуть кешувати відповіді, щоб зменшити затримки при повторних запитах. У випадку збоїв користувач може використовувати інструмент **nslookup** для діагностики DNS серверів та перевірки їх роботи.

DHCP забезпечує автоматичне призначення IP-адрес та інших мережних параметрів пристроям у мережі. Це значно спрощує адміністрування, особливо в середніх і великих мережах, де ручне налаштування кожного пристрою було б надто трудомістким.

Робота DHCP базується на обміні повідомленнями між клієнтом і сервером. Цей процес зазвичай включає чотири основні етапи, відомі як DORA (*Discover, Offer, Request, Acknowledgment*).

Коли пристрій (DHCP-клієнт) підключається до мережі, він розпочинає процес отримання IP-адреси, надсилаючи ширококомовний запит **DHCPDISCOVER**. Це повідомлення містить MAC-адресу клієнта для ідентифікації пристрою і ширококомовну адресу (255.255.255.255) як адресу призначення. Мета цього запиту – знайти всі доступні DHCP-сервери у мережі.

Коли DHCP-сервер отримує запит **DHCPDISCOVER**, він перевіряє свій пул доступних IP-адрес і пропонує одну з них клієнту. Сервер відповідає повідомленням **DHCPOFFER**, яке містить: пропонувану IP-адресу, маску підмережі, шлюз за замовчуванням, IP-адресу DNS-сервера, термін оренди (*lease time*), що визначає, на скільки часу виділяється IP-адреса. Це повідомлення також надсилається ширококомовно, якщо клієнт ще не має власної IP-адреси.

Клієнт отримує одну або декілька пропозицій від DHCP-серверів, обирає одну з них і надсилає ширококомовний запит **DHCPREQUEST**. У цьому повідомленні клієнт підтверджує обраний сервер та пропонувану адресу.

Обраний DHCP-сервер отримує запит **DHCPREQUEST**, перевіряє, чи обрана адреса досі доступна, і відповідає повідомленням **DHCPACK**. У цьому повідомленні сервер підтверджує оренду IP-адреси.

Якщо IP-адреса за запитом клієнта недоступна (наприклад, вже використовується іншим пристроєм), сервер надсилає негативну відповідь **DHCPNAK**. У такому разі клієнт повторює процес з етапу **DHCPDISCOVER**. Після цього клієнт налаштовує мережний інтерфейс і починає використовувати отримані параметри.

Клієнт-серверна модель та однорангові (*peer-to-peer, P2P*) мережі є двома підходами до організації взаємодії між пристроями в мережах, які мають суттєві відмінності у своїй структурі та функціонуванні.

У клієнт-серверній моделі пристрої поділяють на клієнти і сервери. Клієнт – це пристрій або програмне забезпечення, яке ініціює запит на ресурс чи послугу, наприклад, отримання файлу, пошти чи бази даних. Сервер – це пристрій або програмне забезпечення, яке відповідає на запити клієнтів, надаючи доступ до ресурсів чи обчислювальних потужностей. У такій моделі сервери є централізованими, і всі дані чи послуги зберігаються або управляються через ці вузли.

Наприклад, електронна пошта працює за цією схемою: клієнт, як-от програма пошти, запитує дані (нові повідомлення) у поштового сервера, який відповідає, надаючи необхідну інформацію. Основні переваги цієї моделі – централізоване управління, що забезпечує високу ефективність, безпеку та контроль доступу до даних.

Однорангові мережі мають іншу структуру. У них немає чітко визначеного сервера, натомість усі пристрої, які беруть участь у мережі, виконують одночасно функції клієнта і сервера. Кожен пристрій може надавати ресурси, наприклад, файли чи периферійні пристрої, іншим пристроям, а також запитувати доступ до ресурсів, якими володіють інші учасники мережі. У такій моделі обмін відбувається безпосередньо між пристроями, без централізованого управління. Наприклад, у *P2P*-файлообмінних мережах кожен пристрій може як передавати частини файлу, так і отримувати їх від інших учасників мережі.

Клієнт-серверна модель і однорангові мережі відрізняються за ключовими аспектами. У клієнт-серверній моделі є чітка ієрархія ролей і централізована структура, що забезпечує ефективний контроль, надійність і безпеку, але залежить від доступності серверів. Якщо сервер виходить з ладу, клієнти втрачають доступ до послуг. Водночас однорангові мережі мають децентралізовану структуру, яка дозволяє їм бути гнучкішими та стійкішими до відмов окремих вузлів, адже ресурси розподілені між багатьма учасниками. Проте це створює складнощі у забезпеченні безпеки та управлінні ресурсами, оскільки відсутній централізований контроль.

Обидва підходи мають свої переваги й недоліки та застосовуються залежно від завдань. Клієнт-серверна модель підходить для корпоративних систем, де важливі централізація і контроль, тоді як однорангові мережі ефективні для обміну ресурсами між рівноправними учасниками, наприклад, у файлообмінних чи розподілених обчислювальних системах.

1.7 ОСНОВИ МЕРЕЖЕВОЇ БЕЗПЕКИ

Безпека мережі є критично важливою, оскільки сучасні організації та окремі користувачі значною мірою покладаються на комп'ютери та мережні системи для повсякденної діяльності. Без належного рівня захисту мережа стає вразливою до зовнішніх атак, які можуть призводити до серйозних фінансових, інформаційних та репутаційних втрат.

Типи загроз у мережах можна класифікувати за їхнім впливом на дані, ресурси або послуги, а також за методами, які використовують зловмисники. Основні загрози, що можуть виникати:

1. **Крадіжка інформації.** Зловмисники можуть проникати до системи, щоб отримати конфіденційну інформацію, таку як фінансові дані, персональні записи

або корпоративні секрети. Наприклад, викрадення даних досліджень і розробок може завдати значної шкоди бізнесу, позбавляючи його конкурентних переваг.

2. Втрата та маніпуляція даними. У цьому випадку зловмисники можуть знищувати або змінювати важливі записи. Наприклад, вірус може відформатувати жорсткий диск комп'ютера, знищуючи дані. Також можливе внесення змін до інформаційних систем, наприклад, змінення цін на товари чи підроблення фінансових звітів.

3. Крадіжка особистості (ідентичності). Це форма крадіжки інформації, коли особисті дані використовуються для захоплення контролю над чиеюсь особистістю. За допомогою таких даних зловмисники можуть оформлювати кредити, здійснювати онлайн-покупки або отримувати інші юридичні та фінансові вигоди, які в кінцевому підсумку завдають шкоди жертві.

4. Переривання доступу до послуг. Зловмисники можуть здійснювати атаки на доступність послуг, наприклад, через атаки на відмову в обслуговуванні (*DoS*). Це заважає легітимним користувачам отримувати доступ до ресурсів, таких як вебсайти, сервери або мережні служби, і може паралізувати бізнес-процеси.

Ці загрози можуть використовувати різноманітні шляхи проникнення, зокрема програмні уразливості, слабкі паролі, незахищені облікові записи чи фізичний доступ до обладнання. Розуміння цих загроз є ключовим для створення ефективної системи мережної безпеки.

Типи вразливостей у мережах можна поділити на три основні категорії: технологічні, конфігураційні та адміністративні (уразливості політики). Кожна з них має свої особливості та шляхи впливу на безпеку мережі.

Технологічні вразливості – це слабкі місця, що виникають через особливості апаратного чи програмного забезпечення. Вони можуть містити:

1. Слабкості протоколів *TCP/IP*. Деякі протоколи, такі як *HTTP*, *FTP*, *ICMP*, *SNMP*, та *SMTP*, були спроектовані без врахування сучасних вимог до безпеки і їхня структура дозволяє перехоплювати дані чи виконувати атаки.

2. Уразливості операційних систем. Кожна операційна система має свої проблеми із безпекою, які з часом знаходять і документують. Приклади містять архіви *CERT*, що описують відомі слабкі місця ОС.

3. Проблеми мережного обладнання. Маршрутизатори, комутатори чи міжмережні екрани мають слабкі місця, такі як відсутність належного захисту паролем, недоліки автентифікації чи вразливості в протоколах маршрутизації.

Конфігураційні вразливості – це слабкі місця, які виникають через неправильні налаштування або необережне використання обладнання та програмного забезпечення. Наприклад:

1. Незахищені облікові записи. Паролі та облікові дані можуть передаватися у відкритому вигляді, що дозволяє зловмисникам перехопити їх.

2. Системні облікові записи з простими паролями. Слабкі або стандартні паролі роблять системи вразливими до атак перебором.

3. Помилки в налаштуванні інтернет-сервісів. Наприклад, увімкнені за замовчуванням функції, як-от *JavaScript* у браузері, або неправильно налаштовані вебсервери (*IIS, Apache*) відкривають додаткові ризики.

4. Неправильно налаштоване мережне обладнання. Наприклад, невірно складені списки доступу чи помилки у маршрутизації можуть створювати лазівки для атак.

Адміністративні (уразливості політики) – ці слабкі місця пов'язані з відсутністю чітких правил, процедур та контролю. Наприклад:

1. Відсутність письмової політики безпеки. Без чітко задокументованих правил їх важко застосовувати на практиці.

2. Недостатня автентифікація та контроль доступу. Слабкі паролі чи відсутність регулярного аудиту доступу до систем створюють ризики.

3. Неавторизовані зміни. Інсталяція програм або зміни в мережі без дотримання політики можуть відкривати вразливості.

4. Відсутність плану аварійного відновлення. Якщо компанія не має плану дій у разі інцидентів, це може призвести до паніки та значних втрат.

Окремо слід виділити фізичні загрози, які включають доступ до обладнання, пошкодження його через вплив навколишнього середовища або неправильне обслуговування. Приклади: несанкціонований доступ до серверних кімнат, перепади напруги, неправильне прокладання кабелів.

Зловмисне програмне забезпечення, або *malware*, є однією з найпоширеніших загроз у сучасному цифровому середовищі. Воно створюється з метою завдання шкоди даним, системам або користувачам. Існує декілька основних типів такого ПЗ, кожен з яких має свої унікальні характеристики та методи впливу на цільові системи.

Віруси є одним із найстаріших видів зловмисного ПЗ. Вони прикріплюються до виконуваних файлів або програм і активуються, коли користувач запускає заражений файл. Після активації вірус поширюється між системами, викликаючи різноманітні наслідки – від незначних незручностей до серйозного пошкодження даних і навіть створення умов для атак типу «відмова у обслуговуванні» (*DoS*). Віруси зазвичай потребують дій користувача для свого розповсюдження, наприклад відкриття зараженого електронного листа або запуску небезпечної програми.

Хробаки комп'ютерні схожі на віруси за своєю природою, але мають здатність до самостійного розповсюдження без необхідності прикріплення до інших програм. Вони використовують вразливості в операційних системах або мережах для проникнення в системи і поширення на інші пристрої. Оскільки хробаки не потребують взаємодії з користувачем, вони можуть завдати значної шкоди в

короткий час, перевантажуючи мережі або виводячи з ладу критично важливі ресурси.

Трояни – «Троянські коні» – отримали свою назву на честь міфологічних хитрощів, що дали змогу давнім грекам проникнути в Трою. Це програми, які маскуються під легітимне ПЗ, спокушаючи користувачів завантажити та встановити їх. Після активації троян може виконувати різні шкідливі дії, такі як видалення файлів, крадіжка даних або створення бекдору для доступу злоумисників. На відміну від вірусів і хробаків, трояни не здатні до самостійного поширення і потребують дій користувача, таких як завантаження підозрілого файлу чи відкриття шкідливого вкладення в електронному листі.

Кожен із цих типів злоумисного ПЗ становить серйозну загрозу, яка може призвести до втрати даних, фінансових збитків або навіть повної компрометації мережної інфраструктури. Знання їхніх характеристик та механізмів дії є ключовим для ефективного захисту систем.

Атаки на мережі – це навмисні дії, спрямовані на отримання несанкціонованого доступу до ресурсів, викрадення даних або виведення з ладу систем і служб. Вони реалізуються через вразливості у програмному забезпеченні, мережній архітектурі чи недостатньому захисті користувачів. Існують різні типи атак, кожен з яких має свої особливості та механізми реалізації.

Розвідувальні атаки спрямовані на збір інформації про цільову систему чи мережу. Злоумисники використовують такі інструменти, як пошук у мережі, утиліти типу **nslookup** або **whois**, а також сканування портів і активних IP-адрес. Ці атаки є підготовчим етапом перед більш агресивними діями, оскільки дозволяють визначити вразливі точки в системі.

Атаки доступу націлені на несанкціоноване отримання доступу до ресурсів, даних або привілеїв. Злоумисники можуть використовувати підбір паролів, експлуатацію довіри між системами або перехоплення даних для входу. Наприклад, атаки типу «людина посередині» дозволяють злоумиснику втручатися в обмін даними між двома сторонами, змінюючи або викрадаючи інформацію.

Атаки типу «відмова в обслуговуванні» (DoS) створюють перевантаження системи, що робить її недоступною для легітимних користувачів. Більш складний варіант таких атак – розподілена відмова в обслуговуванні (DDoS), коли атака виконується одночасно з багатьох джерел. Це може зупинити роботу цілих мереж або вебсервісів, спричинивши фінансові втрати і втрату репутації для організацій.

Для захисту мережі від атак важливо впроваджувати стратегії, які дозволяють запобігти несанкціонованому доступу, поширенню шкідливого програмного забезпечення та втраті даних. Необхідно дотримуватися комплексного підходу до безпеки, що включає багатошарову стратегію захисту, регулярне оновлення систем, впровадження механізмів автентифікації, використання міжмережних екранів та забезпечення безпеки кінцевих пристроїв.

Одним із ключових методів захисту є багатошарова модель безпеки, яка передбачає використання різних мережних пристроїв і послуг для взаємодії та захисту від атак на протокол *TCP/IP*. Ця модель забезпечує захист не лише кінцевих пристроїв, але й маршрутизаторів, комутаторів та серверів.

Регулярне створення резервних копій конфігурацій пристроїв та даних допомагає уникнути втрат інформації у разі збоїв або атак. Резервні копії повинні зберігатися в безпечних місцях і регулярно перевірятися на цілісність. Це дозволяє швидко відновити роботу систем у разі збою.

Оновлення та встановлення виправлень ПЗ є критично важливими для усунення вразливостей, які можуть використовувати зловмисники. Використання стандартних образів програмного забезпечення для нових пристроїв, автоматизація оновлень та регулярна перевірка систем на наявність нових оновлень значно знижують ризик атак.

Для забезпечення доступу лише авторизованим користувачам застосовуються сервіси автентифікації, авторизації та аудиту (AAA). Вони дозволяють чітко визначати, хто має доступ до ресурсів, які дії дозволені для кожного користувача, та фіксувати всі дії для подальшого аналізу.

Міжмережні екрани виступають ефективним інструментом для захисту внутрішніх мереж від зовнішніх загроз, обмежуючи небажаний трафік. Вони можуть блокувати трафік на основі *IP*-адрес, портів або специфічних додатків. Розміщення певних серверів у демілітаризованій зоні (*DMZ*) дозволяє отримати доступ до них, зберігаючи основну мережу в безпеці.

Захист кінцевих пристроїв є важливою частиною стратегії безпеки, адже саме вони часто стають об'єктом атак через людські помилки. Впровадження політик безпеки, використання антивірусів та систем виявлення вторгнень допомагають знижувати ризики. Мережний контроль доступу додає ще один рівень захисту, забезпечуючи моніторинг і контроль над пристроями, що під'єднуються до мережі.

Захист пристроїв, які підключені до мережі, є ще одним важливим аспектом забезпечення безпеки всієї інфраструктури. Ці пристрої можуть бути як кінцевими (наприклад, комп'ютери, смартфони або планшети), так і проміжними, як-то маршрутизатори і комутатори. Без належного захисту ці пристрої можуть стати мішенню для зловмисників, що призведе до витоку конфіденційної інформації, несанкціонованого доступу чи навіть атак на всю мережу.

Паролі відіграють критичну роль у захисті мережних пристроїв. Використання простих паролів, таких як «123456» або «admin», створює серйозні вразливості. Для створення надійних паролів слід використовувати комбінацію великих і малих літер, цифр, спеціальних символів і, якщо дозволяє система, пробілів. Наприклад, створення фраз, що складаються з кількох слів, може бути ефективним способом для створення пароля, який важче вгадати, але його легше

запам'ятати. Важливо також регулярно змінювати паролі, щоб зменшити ризик компрометації.

Що стосується віддаленого доступу до мережних пристроїв, то *Telnet* є дуже небезпечним методом, оскільки передає всі дані у відкритому вигляді. Замість *Telnet* рекомендується використовувати протокол *SSH*, який забезпечує шифрування всього трафіку, що передається.

Завершальним етапом є вимкнення непотрібних сервісів. Багато маршрутизаторів і комутаторів постачаються з набором активованих сервісів, які можуть бути не потрібні в конкретному випадку. Вимкнення цих сервісів не тільки зберігає ресурси системи, як-то процесорний час та пам'ять, але й зменшує можливість для зломисників використати ці сервіси для атак. Для цього можна вимкнути *HTTP*-сервер або вимкнути *Telnet* і дозволити лише *SSH* з'єднання.

РОЗДІЛ 2

ТЕХНОЛОГІЇ УПРАВЛІННЯ ТА СТРУКТУРИЗАЦІЯ МЕРЕЖ

2.1 ТЕХНОЛОГІЯ VLAN

2.1.1 ВСТУП ДО VLAN

Віртуальні локальні мережі (*VLAN, Virtual Local Area Network*) – це технологія, яка дозволяє розділити фізичну мережу на декілька логічних сегментів. Незважаючи на те, що пристрої можуть бути підключені до однієї фізичної мережі, *VLAN* дозволяє створювати окремі віртуальні мережі, які функціонують ізольовано. Це забезпечує гнучкість і безпеку мережного середовища, дозволяючи ефективніше керувати трафіком.

VLAN використовуються для вирішення різноманітних завдань, таких як ізоляція мережного трафіку для підвищення безпеки, розподіл ресурсів за групами, оптимізація використання мережних пристроїв та зниження широкомерного трафіку. Це особливо корисно в великих організаціях, де потрібно сегментувати мережу для різних відділів.

Основою *VLAN* є концепція створення логічних мереж, які незалежно від фізичного розташування пристроїв можуть об'єднувати їх в окремі сегменти. Це робить мережу більш керованою, масштабованою та безпечною, дозволяючи легко адаптувати її до змін потреб організації. Типова топологія *VLAN* показана на рисунку 2.1.

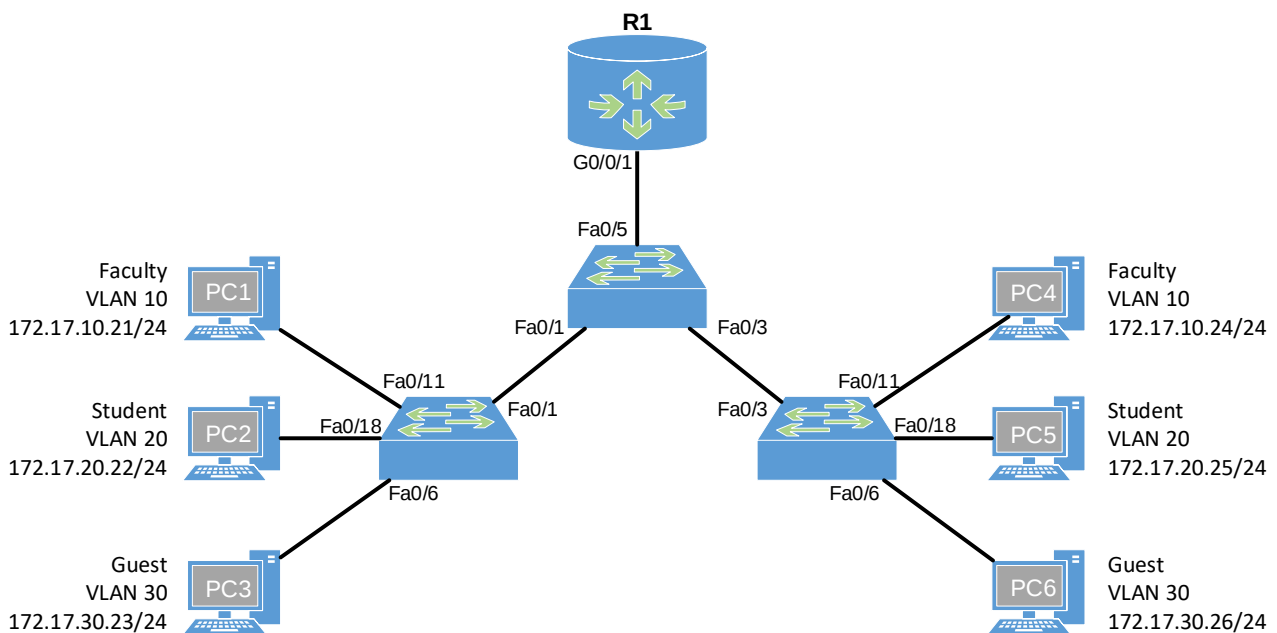


Рисунок 2.1 – Типова топологія мережі з *VLAN*

Можна виділити такі переваги використання *VLAN*:

1. Сегментація мережі:

– *VLAN* дозволяє розділити велику фізичну мережу на менші логічні сегменти. Це допомагає ізолювати трафік, зменшити кількість широкомовних повідомлень і знизити навантаження на мережу.

2. Підвищення безпеки:

– *VLAN* забезпечує ізоляцію трафіку між різними групами користувачів або підмережами. Це дозволяє зменшити ризики несанкціонованого доступу до конфіденційної інформації, обмежуючи трафік для потрібних сегментів мережі.

3. Гнучкість у налаштуванні:

– *VLAN* дає можливість логічно об'єднувати користувачів або пристрої незалежно від їх фізичного розташування. Це зручно для організації мережі в різних підрозділах або філіалах компанії.

4. Оптимізація використання ресурсів:

– *VLAN* дозволяє ефективніше розподіляти мережні ресурси, такі як ширина каналу або IP-адреси. Це допомагає знизити витрати і покращити продуктивність мережі.

5. Зменшення широкомовного трафіку:

– Широкомовні повідомлення обмежуються межами *VLAN*, що знижує кількість зайвого трафіку в мережі та покращує продуктивність. *VLAN* створює логічний широкомовний домен, який може охоплювати кілька фізичних сегментів мережі.

6. Покращення управління мережевим трафіком:

– Адміністратори можуть легко керувати трафіком між *VLAN*, застосовувати політики безпеки, QoS та інші налаштування, що забезпечують оптимальну роботу мережі.

7. Підтримка мережного управління:

– *VLAN* спрощує управління великою мережею, дозволяючи централізовано керувати доступом до ресурсів, впроваджувати політики безпеки та моніторинг мережного трафіку.

8. Масштабованість:

– *VLAN* дозволяє легко розширювати мережу, додаючи нові підрозділи або групи користувачів, без необхідності значних змін в існуючій фізичній інфраструктурі.

2.1.2 ТИПИ *VLAN*

Зазвичай класифікація *VLAN* визначає сферу використання того чи іншого типу віртуальної мережі для передачі відповідного трафіку. Основні типи *VLAN* можна виділити такі:

1. *VLAN* за замовчуванням. Це *VLAN*, яка налаштована за замовчуванням на всіх портах комутатора. Зазвичай, це *VLAN 1*, до якої відносяться всі порти і

пристрої, якщо їм не призначено інший *VLAN*. Використовується для початкової конфігурації та виявлення пристроїв у мережі. *VLAN 1* не можна перейменувати або видалити.

2. *VLAN* даних. Основний тип *VLAN*, який використовується для сегментації користувацького трафіку. Ця *VLAN* ізолює дані різних груп користувачів або підрозділів, забезпечуючи організацію логічних підмереж всередині фізичної мережі.

3. *Голосова VLAN*. Спеціалізована *VLAN*, призначена для передачі голосового трафіку, наприклад, для IP-телефонії. Використання *Voice VLAN* дозволяє гарантувати якість обслуговування (*QoS*) для голосових даних, зменшуючи затримки та втрати пакетів.

4. *VLAN* керування. Це *VLAN*, що використовується для керування мережними пристроями (комутаторами, маршрутизаторами, точками доступу тощо). Ця *VLAN* ізолює трафік керування від користувацького, підвищуючи безпеку мережі.

5. *Native VLAN*. *VLAN*, яка призначена для обробки трафіку, що не має приналежності до якої небудь *VLAN* на магістральних портах. За замовчуванням, *Native VLAN* – це *VLAN 1*, але її можна змінити для підвищення безпеки або відповідно до політики мережі.

6. *Гостьова VLAN*. Призначена для гостьового доступу до мережі. Використовується для забезпечення обмеженого доступу до ресурсів мережі, надаючи гостям окремий сегмент з обмеженими правами.

2.1.3 АРХІТЕКТУРА *VLAN*

Архітектура *VLAN* базується на концепції розподілу фізичної мережі на окремі логічні сегменти – віртуальні мережі, що дозволяє ефективніше керувати трафіком та забезпечувати безпеку мережі.

Кожен *VLAN* має унікальний ідентифікатор *VLAN ID* (від 1 до 4094), який використовується для розмежування трафіку. *VLAN ID* додається до *Ethernet* кадрів і дозволяє комутаторам і маршрутизаторам ідентифікувати, до якого *VLAN* належить конкретний пакет даних.

Використання *VLAN* було б обмеженим без магістральних (*trunking*) каналів. Такі канали дозволяють передавати весь трафік *VLAN* між комутаторами, забезпечуючи можливість пристроям, які підключено до різних комутаторів, але які належать до однієї *VLAN*, взаємодіяти без потреби в маршрутизаторі.

Магістральний канал – це зв'язок точка-точка між мережними пристроями, який дозволяє передавати трафік декількох *VLAN* одночасно, поширюючи *VLAN* по всій мережі (рисунок 2.2). Наприклад, *Cisco* підтримує стандарт *IEEE 802.1Q* для налаштування магістральних каналів на інтерфейсах *Fast Ethernet*, *Gigabit Ethernet* та *10-Gigabit Ethernet*.

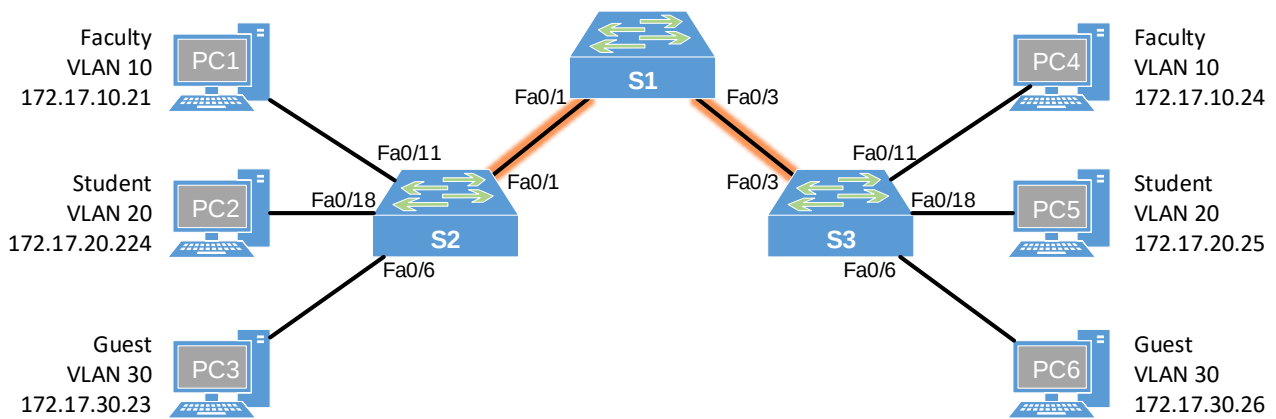


Рисунок 2.2 – Топологія мережі з магістральними каналами

Магістральний канал *VLAN* не прив'язаний до конкретної *VLAN*, а слугує для передачі трафіку між пристроями мережі з підтримкою стандарту *802.1Q*, який на сьогоднішній день є єдиним, що підтримується сучасними комутаторами. Коли комутатор отримує кадр на порту доступу, що належить певній *VLAN*, комутатор додає тег *VLAN* в заголовок кадру, заново розраховує контрольну суму кадру і надсилає тегований кадр через магістральний порт.

Стандартний заголовок *Ethernet* не містить інформації про те, до якого *VLAN* належить кадр. Тому для передачі *Ethernet*-кадрів через магістральний канал потрібно додати цю інформацію. Цей процес називається «тегування» і здійснюється за допомогою заголовка *IEEE 802.1Q* (рисунок 2.3).

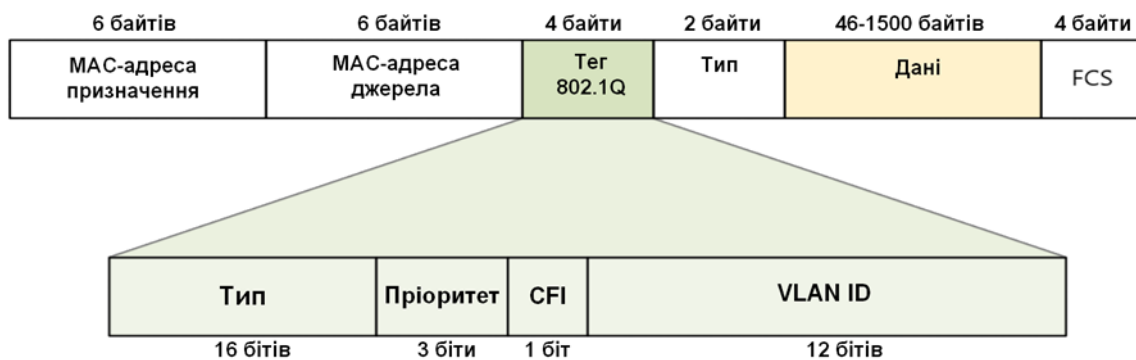


Рисунок 2.3 – Кадр *Ethernet*, тегований заголовком *802.1Q*

Потрібно зазначити, що тег додає 4 додаткових байти до *Ethernet*-заголовка кадру, який вказує на *VLAN*, до якої належить кадр. Найважливішим полем тегу є ідентифікатор *VLAN ID* довжиною 12 бітів. Він визначає *VLAN*, до якої належить кадр. Оскільки значення *0x000* і *0xFFFF* зарезервовані, існує 4094 можливих номерів *VLAN*. Поле «Тип» називається ідентифікатором протоколу тегу і для технологій *Ethernet* воно має шістнадцяткове значення *0x8100*. Поле пріоритету задає пріоритезацію трафіку, ідентифікатор канонічного формату (*CFI*) дозволяє передавати кадри технології *Token Ring* через канали мережі *Ethernet*.

За замовчуванням всі порти комутатора, якщо не було виконано налаштувань, належать до *VLAN 1*. Вона також є *Native VLAN* для магістральних

каналів і визначається стандартом *IEEE 802.1Q*. Якщо нетегований кадр отримується на магістральному порту, то він направляється у *Native VLAN*. Наприклад, якщо налаштувати *Native VLAN* на магістралі за номером 20, то дані без заголовка *IEEE 802.1Q* коли придуть на цей порт, вони будуть перенаправлені у *VLAN 20*. Існує ще один дуже важливий аспект цієї концепції. Комутатори не тільки поміщають отримані нетеговані дані у *Native VLAN*, але й надсилають дані у *Native VLAN* без тегів.

Конфігурація інтерфейсу магістралі є локально значущою. Це означає, що якщо налаштування магістралі на одному комутаторі точно не збігаються із налаштуваннями на іншій стороні лінії зв'язку, то це спричинить несправний стан, який називається невідповідністю *Native VLAN*.

2.1.4 ПРОТОКОЛ DTP

Кожен інтерфейс комутатора може працювати як порт доступу або магістральний порт. Оскільки в типовому розгортанні локальної мережі є сотні або навіть тисячі портів комутаторів, існує пропрієтарний протокол *Cisco* під назвою *Dynamic Trunking Protocol (DTP)*, який допомагає мережним адміністраторам автоматично встановлювати режим роботи інтерфейсів. За замовчуванням всі порти комутаторів *Cisco* знаходяться в робочому стані **dynamic auto**, що означає, що цей *Dynamic Trunking Protocol (DTP)* слухає і намагається зрозуміти, що налаштовано на іншому кінці кабелю, і на основі цього вирішити, стати йому портом доступу чи магістральним портом. Наприклад, якщо є з'єднання між двома комутаторами, і налаштовано інтерфейс на одному комутаторі як магістральний порт, то *DTP* повідомить про це іншій стороні, інтерфейс на другому комутаторі автоматично перейде в магістральний режим і між комутаторами буде створено магістральне з'єднання.

Таблиця 2.1 ілюструє результати застосування різних варіантів налаштування *DTP* на протилежних кінцях транкового каналу, сформованого між портами комутаторів *Cisco*.

Таблиця 2.1 – Застосування різних варіантів налаштування *DTP*

	Dynamic Auto	Dynamic Desirable	Trunk	Access
Dynamic Auto	Access	Trunk	Trunk	Access
Dynamic Desirable	Trunk	Trunk	Trunk	Access
Trunk	Trunk	Trunk	Trunk	Limited connectivity
Access	Access	Access	Limited connectivity	Access

Найкращою практикою, за можливості, є налаштування статичних магістральних каналів.

2.1.5 ПЕРЕСИЛАННЯ ДАНИХ МІЖ VLAN

Комутатори локальної мережі пересилають кадри на основі каналного Рівня 2. Це означає, що коли комутатор отримує кадр *Ethernet*, він аналізує *MAC-адресу* призначення і пересилає кадр через інший інтерфейс або кілька інтерфейсів, якщо це широкомовний кадр. Цей тип комутаторів часто називають комутаторами 2-го рівня. Логіка пересилання на Рівні 2 виконується для кожної *VLAN*, оскільки кожна з них є окремим широкомовним доменом.

Створити зв'язок між *VLAN* означає увімкнути зв'язок між *IP*-підмережами. Отже, потрібно мати пристрій, який виконує функції маршрутизатора. Існує два можливих рішення: використати для маршрутизації власне маршрутизатор, або ж комутатор, що може виконувати функції маршрутизації. Комутатори, які можуть виконувати функції маршрутизації на Рівні 3, називаються комутаторами Рівня 3 або багаторівневими комутаторами.

Класична схема містить маршрутизатор, що має один фізичний інтерфейс, під'єднаний до комутатора у *VLAN10*, і один фізичний інтерфейс, під'єднаний до комутатора у *VLAN20*. Таким чином, маршрутизатор має один інтерфейс у підмережі *172.25.10.0/24* і один інтерфейс у підмережі *172.25.20.0/24* і робить те, що роблять усі маршрутизатори – маршрутизує *IP*-пакети між підмережами (рисунок 2.4).

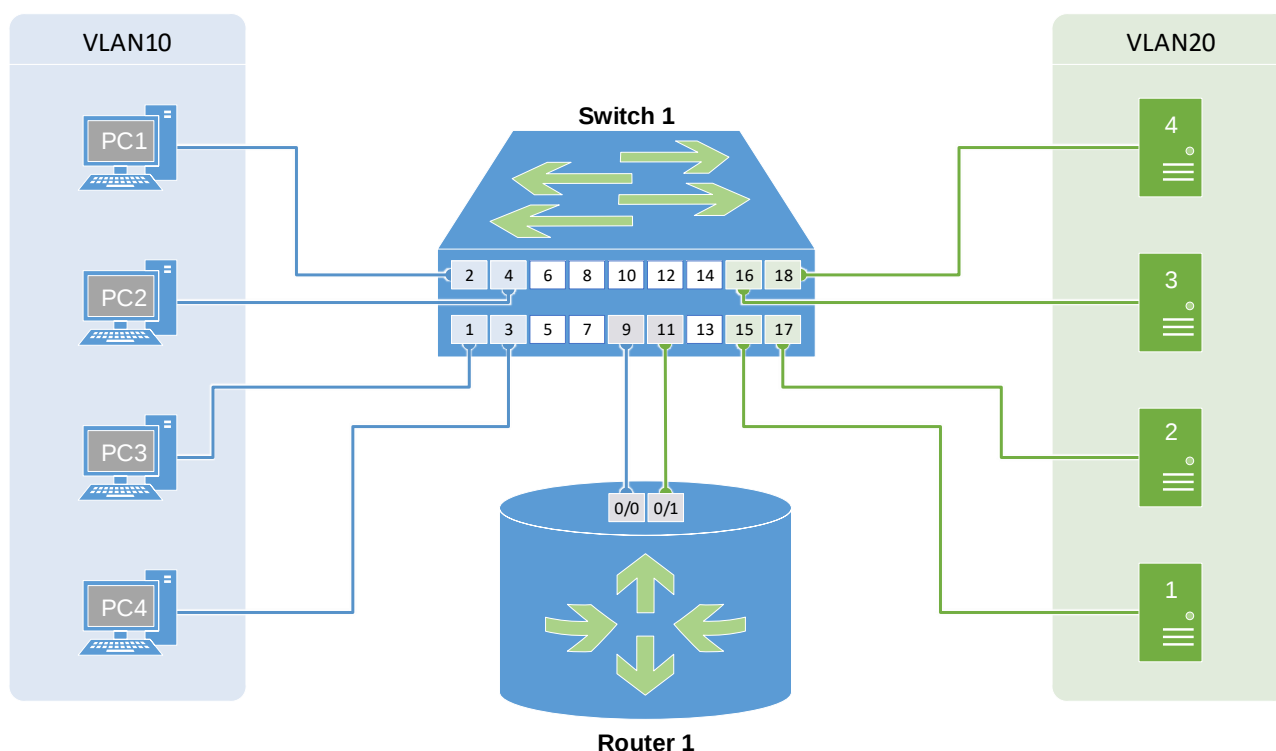


Рисунок 2.4 – Маршрутизація між *VLAN* за допомогою маршрутизатора

Такий метод маршрутизації між *VLAN* більше не використовується в комутуваних мережах, тому що має погану масштабованість та неефективне використання портів маршрутизатора.

На даний час існує два типових пристрої, які використовуються для виконання маршрутизації між VLAN:

1. Багаторівневий комутатор (*MLS*, комутатор Рівня 3). Такі комутатори працюють як на 2-му, так і на 3-му рівнях моделі *OSI*. Вони можуть комутувати кадри і виконувати *IP*-маршрутизацію між *VLAN*.

2. Маршрутизатор. Існує два способи використання маршрутизатора як пристрою, який виконує *IP*-маршрутизацію між *VLAN*. Але, на відміну від класичного застосування, підключення маршрутизатора відбувається за одним каналом зв'язку до магістрального порту комутатора із визначенням підінтерфейсів для кожного *VLAN*. Потім на кожному підінтерфейсі з відповідної *VLAN* налаштовується *IP*-адреса. Цю технологію називають *router-on-a-stick* (*ROAS*), оскільки між маршрутизатором і комутатором є лише одне фізичне з'єднання, як показано на рисунку 2.5.

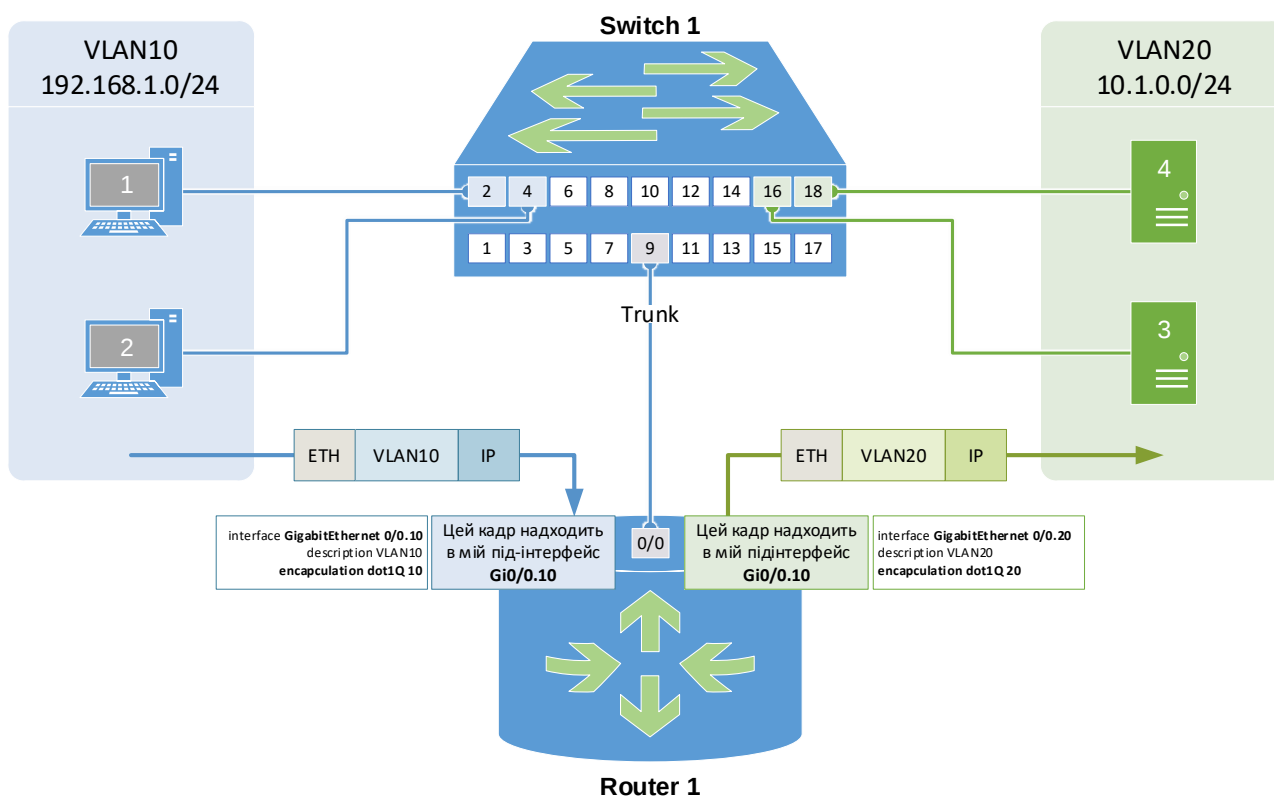


Рисунок 2.5 – Маршрутизація між *VLAN* за методом *router-on-a-stick*

Router-on-a-stick (*ROAS*) – це технологія під'єднання маршрутизатора за допомогою одного фізичного каналу до комутатора і виконання *IP*-маршрутизації між *VLAN*. З точки зору комутатора, цей фізичний канал налаштований як магістральний порт, що дозволяє під'єднати всі *VLAN*, які будуть маршрутизуватися. З точки зору маршрутизатора, цей фізичний інтерфейс представлений у вигляді декількох віртуальних підінтерфейсів, по одному для кожної *VLAN*. Потім на кожному підінтерфейсі налаштовується *IP*-адреса з кожної

VLAN, і маршрутизатор виконує IP-маршрутизацію між підключеними мережами.

Якщо уважніше розглянути наведену вище фізичну схему, то є один кабель, який з'єднує маршрутизатор **Router 1** з комутатором **Switch 1**. З точки зору комутатора, порт 9 є магістраллю 802.1Q, яка надсилає всі кадри до маршрутизатора з тегом VLAN.

З точки зору маршрутизатора, всі кадри надходять з тегами. Отже, на основі цієї мітки маршрутизатор може розрізняти, які вхідні кадри є частиною якої VLAN. Знаючи номер VLAN кожного кадру, маршрутизатор також знає, до якої підмережі він належить, оскільки **VLAN = широкомовний домен = підмережа**. Таким чином, на основі цієї логіки можна створити віртуальний інтерфейс на маршрутизаторі, який отримує всі кадри з тегом VLAN 10 як підключені до віртуального інтерфейсу **Gi0/0.10**, а всі кадри з тегом VLAN20 як підключені до підінтерфейсу **Gi0/0.20**.

Проте слід зазначити, що метод *ROAS* для маршрутизації між VLAN не підходить для масштабування на понад 50 VLAN, що може бути актуально для великих корпоративних мереж. Тому для розробки такого масштабного проекту може бути застосовано метод маршрутизації між VLAN за допомогою комутаторів Рівня 3.

Багаторівневі комутатори за рахунок використання апаратної комутації та роботи на 3 рівні моделі OSI можуть комутувати кадри як звичайний комутатор і виконувати IP-маршрутизацію як маршрутизатор. Таким чином, вони можуть виконувати функції як на Рівні 2 моделі OSI, так і на Рівні 3. Саме тому їх називають багаторівневими комутаторами або комутаторами 3-го рівня.

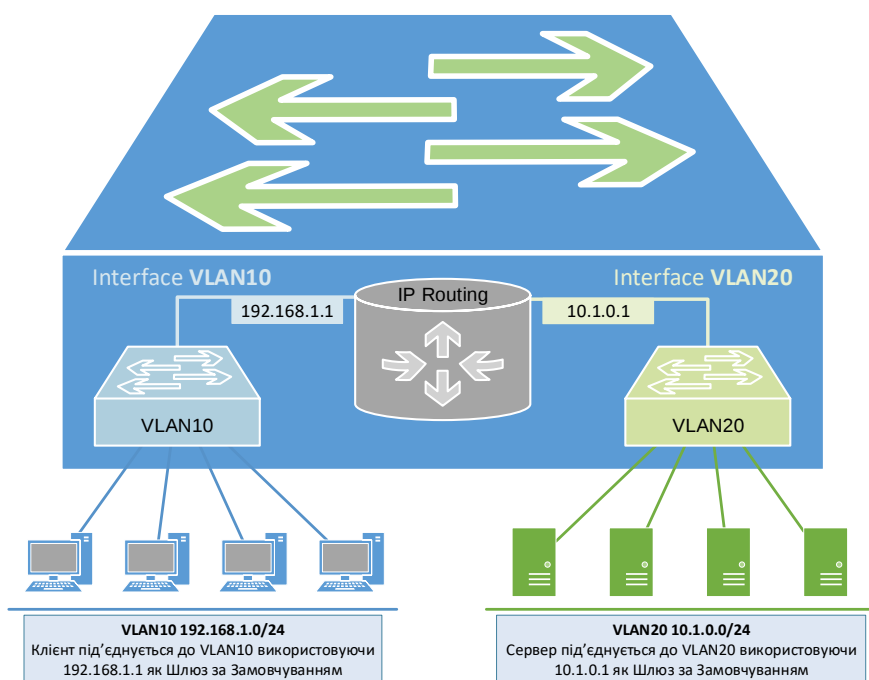


Рисунок 2.6 – Маршрутизація між VLAN комутатором Рівня 3

Відповідно до концепції міжмережної маршрутизації, маршрутизатор, який здійснює її, має інтерфейс маршрутизації в кожному широкомовному домені. Цей інтерфейс маршрутизації має *IP*-адресу з відповідної підмережі, і ця *IP*-адреса використовується вузлами *VLAN* як шлюз за замовчуванням. Але у випадку багаторівневого комутатора, який виконує маршрутизацію між *VLAN*, немає ні маршрутизатора, ні інтерфейсів маршрутизації. Саме тому комутатори 3-го рівня використовують концепцію віртуального інтерфейсу *SVI* (*switched virtual interface*), який з'єднує широкомовний домен *VLAN* з процесом маршрутизації.

Основні переваги використання комутаторів третього рівня для маршрутизації між *VLAN* такі:

- Вони значно швидші за метод *ROAS*, оскільки комутація та маршрутизація виконуються апаратно.
- Немає потреби у зовнішніх з'єднаннях між комутатором і маршрутизатором для здійснення маршрутизації.
- Комутатори третього рівня не обмежуються одним каналом, оскільки канали *EtherChannel* другого рівня можуть використовуватися як магістральні з'єднання для підвищення пропускної здатності.
- Значно нижча затримка, оскільки дані не залишають комутатор для переспрямування в іншу мережу.

2.2 ПРОТОКОЛ *STP*

2.2.1 ПРИЗНАЧЕННЯ *STP*

Протокол єднального дерева (*Spanning Tree Protocol, STP*) дозволяє локальним мережам *Ethernet* мати резервні канали, які можуть бути використані в разі відмови каналу або комутатора. *STP* дозволяє використовувати достатню кількість надлишкових каналів, щоб жодна точка відмови не призвела до виходу з ладу всієї локальної мережі. *IEEE 802.1D* є оригінальним стандартом *IEEE*, що регламентує роботу *STP*.

Ethernet не працює в надлишкових топологіях через петлі *Ethernet*, які також називають петлями Рівня 2 або широкомовними штормами. Без технології, яка розбиває локальну мережу з надлишковими каналами на топологію без петель, кадри штормів (широкомовні, невідомі неодноадресні та багатоадресні) циркулювали б нескінченно довго, поки канал (або пристрій) не вийшов би з ладу. Це відбувається через те, яким чином комутатори пересилають широкомовні кадри.

Коли комутатор отримує кадр, він перевіряє *MAC*-адресу одержувача у своїй таблиці *MAC*-адрес і, якщо не знаходить відповідного запису, пересилає кадр на всі інтерфейси, окрім вхідного. Цей процес називають штормом, а кадр, *MAC*-адреса призначення якого комутатору невідома – невідомим одно-

адресним кадром. Також є ймовірність, що одержувач відповість, тому комутатор дізнається *MAC*-адреси обох вузлів і продовжить подальший процес пересилання як відомий одноадресний кадр.

Комутатори також пересилають два інших типи кадрів:

- широкомовні – призначені для широкомовної адреси *Ethernet* FF-FF-FF-FF-FF-FF-FF;
- багатоадресні – призначені для *MAC*-адреси, яка починається з бітів «1110».

На рисунку 2.7 показано приклад, в якому **PC1** надсилає один широкомовний кадр.

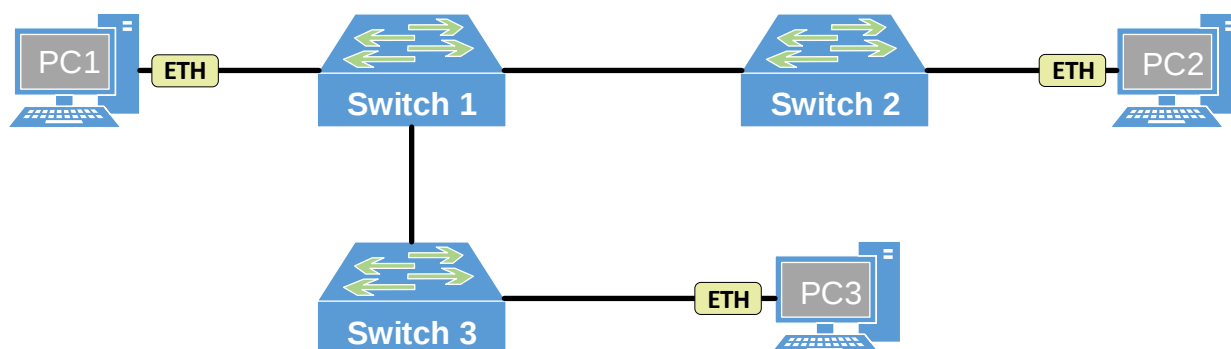


Рисунок 2.7 – LAN без надлишкових каналів

Виходячи з *MAC*-адреси призначення, **Switch 1** знає, що це широкомовний кадр, і тому розсилає його копію через усі свої порти, окрім вхідного. Таким чином, копія кадру потрапляє до комутаторів **Switch 2** і **Switch 3**, які виконують ту ж саму логіку. Зрештою, кожен вузол у широкомовному домені отримує копію цього пакету.

У наведеній топології локальної мережі все працює належним чином, проте відсутнє резервування у випадку збою. Резервування, як ключовий елемент ієрархічної мережної архітектури, забезпечує захист від єдиної точки відмови і зберігає доступність мережних послуг для користувачів. Воно містить не тільки створення додаткових фізичних шляхів, але й використання логічної надлишковості в мережі. Завдяки альтернативним фізичним шляхам, дані можуть передаватися, навіть якщо один з маршрутів вийде з ладу. Однак у комутованих мережах *Ethernet* наявність надлишкових шляхів може призвести до виникнення фізичних і логічних петель на другому рівні моделі *OSI*. Розглянемо, що відбудеться, якщо додати резервний зв'язок між **Switch 2** і **Switch 3**.

Перша проблема – це те, що навіть один зациклений кадр викликає широкомовний шторм. Це відбувається, коли кадри зациклюються навколо комутаторів до нескінченності. Широкомовний шторм триває доти, доки канал не перенасититься і не вийде з ладу, або поки комутатор не вийде з ладу через високе навантаження на процесор (рисунки 2.8).

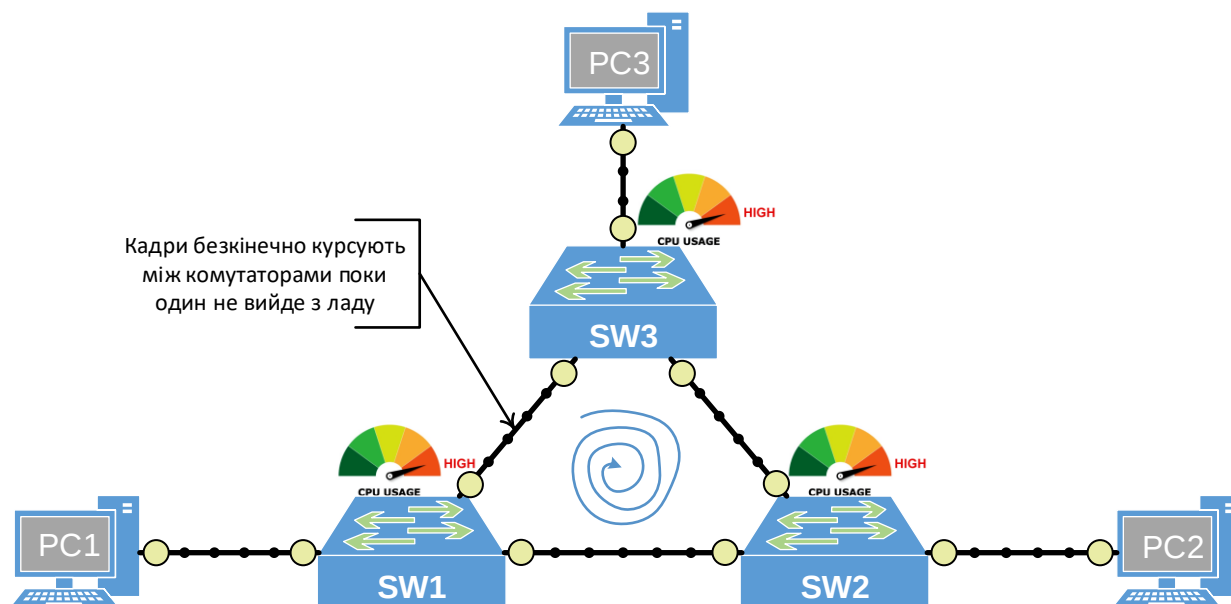


Рисунок 2.8 – LAN без надлишкових каналів

Шторм спричиняє ще одну проблему – хаотичність MAC-таблиці. Якщо згадати процес вивчення MAC-адрес, то коли комутатор отримує кадр, він створює запис у таблиці MAC-адрес для адреси джерела та вхідного порту. Але у випадку широкомовного шторму, кілька копій одного і того ж кадру циклічно повторюються, і комутатор отримує його на кілька інтерфейсів. Але одна MAC-адреса може бути прив'язана лише до одного інтерфейсу. Тому комутатор постійно переписує запис для вихідної MAC-адреси з іншим інтерфейсом, а звідси і хаотичність.

Крайня проблема, яка виникає у випадку петлі, полягає в тому, що кінцеві клієнти отримують кілька копій кадрів знову і знову, поки активний широкомовний шторм. В результаті кінцеві пристрої можуть перевантажити процесор, а критичні застосунки можуть відчувати нестачу ресурсів і вийти з ладу. Ось чому для усунення проблеми петель потрібен протокол, який зможе контролювати утворення петельних з'єднань та усувати в такій топології надлишкові зв'язки.

2.2.2 ПРИНЦИПИ РОБОТИ STP

STP базується на алгоритмі, винайденому Радією Перлман у 1985 році і опублікованому в статті під назвою «Алгоритм для розподіленого обчислення єдиного дерева в розширеній локальній мережі». Алгоритм створює топологію без петель, вибираючи один кореневий міст, а потім всі інші комутатори обчислюють єдиний мінімально витратний шлях до кореня.

Алгоритм виконує кілька кроків, щоб переконатися, що топологія не містить петель і *Ethernet* буде працювати правильно:

1. Вибір кореневого моста. Перше, що робить STP, це вибирає кореневий міст. Це головний комутатор в топології. Він буде коренем дерева без петель.

2. Пошук за кільцьованих топологій. Після того, як кореневий міст обрано, він починає надсилати повідомлення *Spanning-Tree*, які називаються *BPDU* (*Bridge Protocol Data Units*). На основі цих повідомлень комутатори знаходять за кільцьовані частини топології.

3. Налаштування ролей портів. Після виявлення за кільцьованої частини топології, кожен комутатор встановлює стільки портів, скільки потрібно, щоб забезпечити відсутність петель в топології.

4. Повторна збіжність у разі збоїв. Комутатори продовжують обмінюватися повідомленнями, щоб відстежувати доступність каналів і сусідніх комутаторів. Якщо канал або комутатор виходить з ладу, комутатори знову виконують кроки 2 і 3, щоб переконатися, що нова топологія не містить петель.

Вибір кореневого моста. Комутатори обирають кореневий міст на основі значення, яке називається ідентифікатором моста (*bridge ID, BID*). Комутатор, який має найнижче значення *BID*, обирається кореневим мостом топології. *BID* не є єдиним значенням, а складається з двох різних типів значень:

$$BID = (\text{Пріоритет} + \text{Номер VLAN}) : (\text{MAC-адреса системи})$$

Перша частина значення *BID* налаштовується і використовується мережними адміністраторами для налаштування певного комутатора як кореневого моста.

Друга частина значення *BID* використовується лише тоді, коли існує зв'язок, тобто коли є принаймні два комутатори, які мають однакове значення пріоритету. Зазвичай це трапляється, коли всі комутатори залишено зі значеннями за замовчуванням, тому всі комутатори мають пріоритет 32768. У цьому випадку процес виборів відбувається шляхом вибору комутатора з найнижчою системною MAC-адресою.

Коли комутатор завантажується, він не знає значень *BID* всіх інших комутаторів у топології. Тому він обирає себе кореневим мостом топології. Як тільки він отримує *BPDU* зі значенням *Root BID*, меншим за його власне, він негайно припиняє анонсувати себе як кореневий і починає пересилати значення головного кореневого моста (рисунок 2.9 а). У повідомленнях *BPDU* він вказує власне значення *BID* і *BID* кореневого моста, відомого їм на даний момент.

Приклад на рисунку 2.9 б) показує стан після того, як відбувся обмін першими *BPDU*-повідомленнями. Таким чином, комутатор **SW2** отримує два повідомлення *BPDU*, одне від **SW1** і одне від **SW3**. *BPDU* від **SW1** показує, що *Root Bridge* має значення 32769:0000.0000.0001. Повідомлення *BPDU*, яке має менше значення *Root BID*, ніж ваше власне, називається *Superior BPDU*. Як тільки **SW2** отримує це повідомлення, він припиняє рекламувати себе як кореневий і починає пересилати цей *Superior BPDU* до всіх інших комутаторів, крім кореневого.

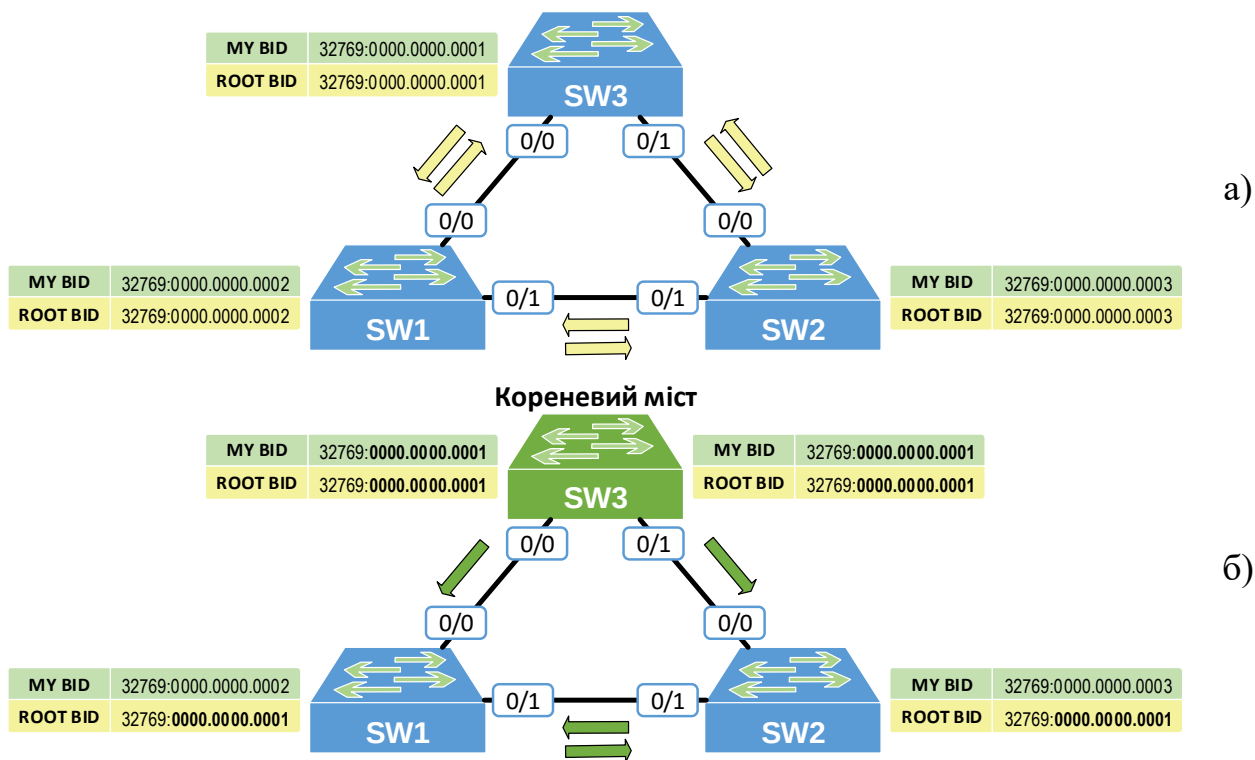


Рисунок 2.9 – Вибори кореневого моста

BPDU, отримане від **SW3**, має значення *Root-BID* 32769:0000.0000.0003. Повідомлення *BPDU*, яке має таке ж або більше значення *Root BID*, аніж власне, називають *Inferior BPDU*. Як тільки комутатор отримує таке повідомлення, він його відкидає.

В кінці цього процесу всі комутатори в топології повинні погодитися з тим, що існує тільки один кореневий міст і він однаковий з точки зору кожного комутатора.

Після завершення виборів кореневого моста комутатори починають ідентифікувати петлі. Комутатор розуміє, що є петля, коли він отримує *Superior BPDU* від кореневого моста на більш ніж одному порту, і цей порт має бути переведений у стан блокування.

Кожен некореневий комутатор обирає один кореневий порт. Кореневий порт – це порт на комутаторі, який має найкращий шлях (найменшу вартість) до кореневого моста. Кореневий комутатор не має кореневих портів, всі його порти стають призначеними (*designated ports*). Вартість шляху до кореневого моста обчислюється для кожного комутатора на основі швидкості та кількості підключених портів на шляху. Вартість каналу визначається за стандартом *IEEE*, де порти з вищою пропускною здатністю мають нижчу вартість. Порт з найменшою вартістю стає кореневим портом (*root port*).

Якщо кілька портів мають однакову вартість шляху, порівнюються *BID* сусідніх комутаторів. Порт, підключений до комутатора з нижчим *BID*, обирається кореневим. Якщо *BID* теж однакові, рішення приймається на основі пріоритету порту та ідентифікатора порту. На рисунку 2.10 показано обрані відповідні типи портів за згаданим алгоритмом.

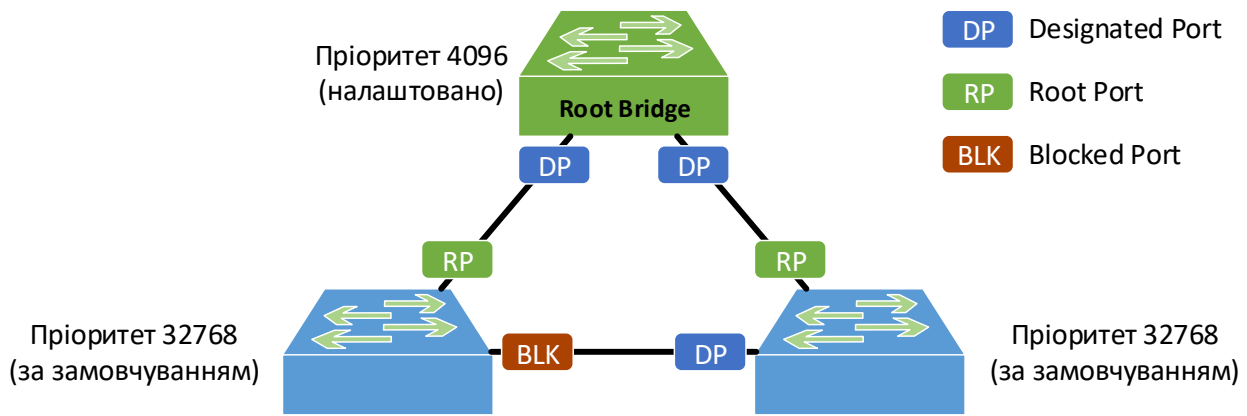


Рисунок 2.10 – Обрання портів для усунення петель

Після визначення кожним комутатором кореневого порту комутатори обирають призначені порти. Призначеним портом стає той порт, який має найнижчу вартість шляху до кореневого мосту для даного сегмента. Порти, які не обираються як кореневі або призначені, стають альтернативними або заблокованими портами. У кінцевому підсумку від кожного комутатора в системі пролягатиме лише один шлях до кореневого моста.

Таблиця 2.2 – Стани портів на комутаторі

Стан портів	Опис стану
Блокування	Порт не передає і не приймає трафік, крім пакетів <i>BPDU</i> . Використовується для запобігання петель.
Прослуховування	Порт готується до переходу в активний стан, прослуховує <i>BPDU</i> для перевірки, чи є безпечним вмикання порту, але ще не вивчає <i>MAC</i> -адреси і не передає трафік.
Навчання	Порт починає вивчати <i>MAC</i> -адреси, щоб оновити таблицю комутації, але все ще не передає дані.
Пересилання	Порт передає і приймає трафік. Це активний стан порту.
Вимкнено	Порт вимкнений адміністратором або через несправність. Він не бере участі в <i>STP</i> .

Процедури збіжності *STP* використовують такі три часових параметри:

- **Hello Timer** (Таймер привітання). Це інтервал між пакетами *BPDU*. Зазвичай складає 2 с. (діапазон від 1 до 10 с.).
- **Forward Delay Timer** (Затримка переадресації). Це час перебування у стані прослуховування і навчання. За замовчуванням складає 15 с. (діапазон від 4 до 30 с.).
- **Max Age Timer** (Максимальний вік). Це максимальна тривалість часу очікування комутатора перед спробою змінити топологію *STP*. Типове значення – 20 с. (діапазон від 6 до 40 с.).

2.2.3 ПРОТОКОЛ RSTP

Rapid Spanning Tree Protocol (RSTP), або швидкий протокол *STP*, є удосконаленою версією традиційного протоколу, визначений стандартом *IEEE 802.1w*. *RSTP* був розроблений для швидшого усунення комутаційних петель у мережах *Ethernet* і забезпечення швидшого відновлення з'єднань після змін у топології мережі.

В *RSTP* скорочено кількість станів портів до трьох: пересилання – порт передає трафік, прослуховування – порт вивчає *MAC*-адреси, але ще не передає трафік, блокування – порт не передає і не приймає трафік.

З іншого боку в *RSTP* додали ще одну роль портів: кореневий порт – найкращий шлях до кореневого мосту, призначений порт – порт, який обробляє трафік для сегмента мережі, альтернативний порт – неактивний порт, що забезпечує резервний шлях до кореневого мосту, резервний порт – неактивний резервний порт для призначеного порту.

Саме ця модифікація протоколу і створена для того, щоб у разі відмови основного порту і переходу на резервне з'єднання *RSTP* не потрібно було заново прораховувати топологію – він просто перейде на запасний, заздалегідь прорахований порт.

Окрім вже розглянутих версій протоколу єднального існують такі поширені версії: *MSTP (Multiple STP)* – дозволяє створення кількох незалежних дерев для різних *VLAN*, що забезпечує ефективніше використання ресурсів мережі, *PVST (Per-VLAN Spanning Tree)* – пропрієтарна версія від *Cisco*, яка створює окреме дерево для кожної *VLAN*. *RPVST+* – яка поєднує можливості *RSTP* і *PVST+*, забезпечуючи швидке зближення і підтримку окремих дерев для кожної *VLAN*.

2.3 АГРЕГАЦІЯ КАНАЛІВ

2.3.1 ПРИНЦИПИ АГРЕГАЦІЇ КАНАЛІВ

Технології агрегації каналів дозволяють об'єднувати декілька фізичних мережних каналів в один логічний канал для збільшення пропускної здатності, підвищення надійності та балансування навантаження. На рисунку 2.11 показано принцип об'єднання фізичних з'єднань мережі в один логічний канал.

В загальному випадку набір фізичних портів, які входять до логічного каналу називають групою агрегації каналів (*Link Aggregation Group, LAG*). Технологія затверджена в стандарті *IEEE 802.1AX*. І хоча іноді таку технологію іноді називають *EtherChannel*, це не зовсім правильно. *EtherChannel* – це пропрієтарна технологія *Cisco*, на якій і було побудовано стандарт, який регламентує *LAG*.

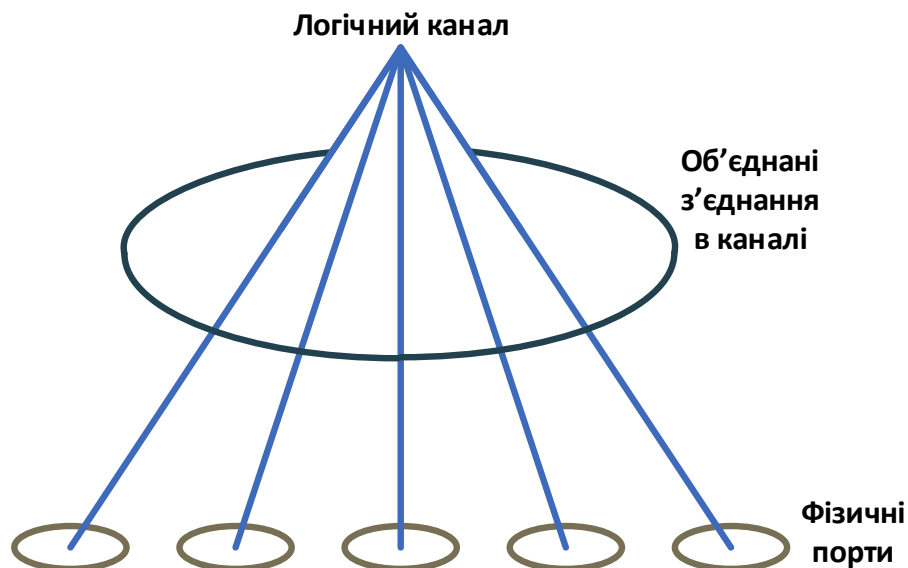


Рисунок 2.11 – Співставлення логічного каналу з фізичними портами

2.3.2 ETHERCHANNEL

Використання *EtherChannel* має численні переваги, і, мабуть, найбільш бажаним аспектом є пропускна здатність. При використанні максимум 8 фізичних портів загальна пропускна здатність може становити 800 Мбіт/с, 8 Гбіт/с або 80 Гбіт/с, залежно від швидкості порту. Формування каналу може бути з використанням стандарту *Ethernet* на основі витої пари, одномодового та багатомодового скловолокна.

Технологія *EtherChannel* пропонує кілька ключових переваг:

- Налаштування виконується на рівні *EtherChannel*, а не кожного окремого порту, забезпечуючи узгодженість конфігурації.
- Використання існуючих портів комутатора дозволяє збільшити пропускну здатність без заміни на дорожчі й швидші канали.
- Балансування навантаження здійснюється між з'єднаннями в *EtherChannel*, використовуючи методи на основі MAC- або IP-адрес джерела та призначення (таблиця 2.3).
- *EtherChannel* сприймається як один логічний канал, що зменшує кількість комутаційних петель та спрощує роботу протоколу *STP*.
- Забезпечується резервування: втрата одного з фізичних з'єднань не викликає зміни топології, і *EtherChannel* залишається функціональним, хоча й зі зменшеною пропускну здатністю.

У разі виходу з ладу одного з каналів, технологія *EtherChannel* автоматично перерозподіляє трафік між каналами, що залишилися. Це автоматичне відновлення займає менше однієї секунди і є прозорим для мережевих додатків і кінцевого користувача. Це робить її дуже відмовостійкою і бажаною для критично важливих додатків.

Таблиця 2.3 – Балансування навантаження в *EtherChannel*

Кількість портів в каналі	Розподіл балансування між портами
8	1:1:1:1:1:1:1:1
7	2:1:1:1:1:1:1
6	2:2:1:1:1:1
5	2:2:2:1:1
4	2:2:2:2
3	3:3:2
2	4:4

Критерії для формування *EtherChannel* для всіх портів існують такі:

- Однаковий дуплексний режим.
- Однакова швидкість портів.
- Однакова конфігурація *VLAN* (тобто, *Native VLAN* і дозволені *VLAN* повинні бути однаковими).
- Режими портів комутатора повинні бути однаковими (режим доступу або магістралі).

Крім статичного налаштування *EtherChannel* – *LAG*, для формування *EtherChannel* існує 2 протоколи автоузгодження: протокол агрегації портів (*PAgP*) і протокол управління агрегацією каналів (*LACP*).

2.1.3 ПРОТОКОЛ АГРЕГАЦІЇ ПОРТІВ

Протокол агрегації портів (*Port Aggregation Protocol, PAgP*) – це пропрієтарний протокол *Cisco*, що може працювати тільки на пристроях *Cisco* і на тих пристроях, які мають ліцензію від постачальників на підтримку *PAgP*. *PAgP* полегшує автоматичне створення *EtherChannel* шляхом обміну *PAgP*-пакетами між портами *Ethernet*. *PAgP* можна ввімкнути на *Cross-Stack EtherChannel*.

Використовуючи *PAgP*, комутатор або стек комутаторів встановлює ідентифікаційні дані пристроїв, здатних підтримувати *PAgP*, і можливості кожного порту. Потім він динамічно групує однаково налаштовані порти (на одному пристрої в стеку) в єдине логічне з'єднання (канал або агрегований порт). Однаково налаштовані порти групуються на основі апаратних, адміністративних вимог та обмежень параметрів портів. Наприклад, *PAgP* групує порти з однаковою швидкістю, дуплексним режимом, *Native VLAN*, діапазоном *VLAN*, а також статусом і типом магістралі. Після групування каналів в *EtherChannel*, *PAgP* додає групу до єдиного дерева як один порт пристрою.

Режими *PAgP* визначають, чи може порт надсилати *PAgP*-пакети, які починають *PAgP*-узгодження, або тільки відповідати на отримані *PAgP*-пакети.

Таблиця 2.4 – Режими *EtherChannel PAgP*

Режим	Опис
Автоматичний	Переводить порт у пасивний стан узгодження, у якому порт відповідає на отримані <i>PAgP</i> -пакети, але не починає узгодження <i>PAgP</i> -пакетів. Цей параметр мінімізує передачу <i>PAgP</i> -пакетів.
Рекомендований	Переводить порт в активний стан узгодження, у якому порт починає узгодження з іншими портами, надсилаючи <i>PAgP</i> -пакети.

Порти комутатора обмінюються *PAgP*-пакетами лише з портами-партнерами, які налаштовані в автоматичному або рекомендованому режимах.

І автоматичний, і рекомендований режими дозволяють портам узгоджувати з портами-партнерами формування *EtherChannel* на основі таких критеріїв, як швидкість порту, а для *EtherChannel* 2-го рівня – на основі стану магістралі та номерів *VLAN*.

Порти можуть формувати *EtherChannel*, перебуваючи в різних режимах *PAgP*, якщо ці режими сумісні. Наприклад:

- Порт у рекомендованому режимі може сформувати *EtherChannel* з іншим портом, який перебуває в рекомендованому або автоматичному режимі.
- Порт в автоматичному режимі може сформувати *EtherChannel* з іншим портом у рекомендованому режимі.

Порт в автоматичному режимі не може сформувати *EtherChannel* з іншим портом, який також перебуває в автоматичному режимі, оскільки жоден з портів не починає узгодження *PAgP*.

2.1.4 ПРОТОКОЛ КЕРУВАННЯ АГРЕГАЦІЄЮ З'ЄДНАНЬ

Протокол *LACP* визначений у стандарті *IEEE 802.3ad* і дозволяє пристроям керувати каналами *Ethernet* між пристроями, які відповідають протоколу *IEEE 802.3ad*. *LACP* полегшує автоматичне створення каналів *EtherChannel* шляхом обміну пакетами *LACP* між портами *Ethernet*.

Використовуючи *LACP*, комутатор або стек комутаторів встановлює ідентифікаційні дані пристроїв, здатних підтримувати *LACP*, і можливості кожного порту. Потім він динамічно групує схожі за конфігурацією порти в єдине логічне з'єднання (канал або агрегований порт). Порти з однаковою конфігурацією групуються на основі апаратних, адміністративних обмежень та обмежень параметрів порту. Наприклад, *LACP* групує порти з однаковою швидкістю, дуплексним режимом, *Native VLAN*, діапазоном *VLAN*, а також статусом і типом магістралі. Після групування каналів в *EtherChannel*, *LACP* додає групу до єднального дерева як один порт пристрою.

Режими *LACP* визначають, чи може порт надсилати пакети *LACP* або тільки приймати пакети *LACP*.

Таблиця 2.5 – Режими *EtherChannel LACP*

Режим	Опис
Активний	Переводить порт в активний стан узгодження, в якому порт починає узгодження з іншими портами, надсилаючи пакети <i>LACP</i> .
Пасивний	Переводить порт у пасивний стан узгодження, в якому порт відповідає на отримані <i>LACP</i> -пакети, але не починає узгодження. Цей параметр мінімізує передавання <i>LACP</i> -пакетів.

Як активний, так і пасивний режими *LACP* дозволяють портам вести узгодження з портами-партнерами по *EtherChannel* на основі таких критеріїв, як швидкість порту, а для *EtherChannel* 2-го рівня – на основі стану магістралі і номерів *VLAN*.

Порти можуть утворювати *EtherChannel*, коли вони перебувають у різних режимах *LACP*, якщо ці режими сумісні. Наприклад:

- Порт в активному режимі може формувати *EtherChannel* з іншим портом, який перебуває в активному або пасивному режимі.
- Порт у пасивному режимі не може сформувати *EtherChannel* з іншим портом, який також перебуває в пасивному режимі, оскільки жоден з портів не починає *LACP*-переговорів.

Робота *LACP* на рівні каналу, доступність пропускної здатності та резервування можуть бути покращені за допомогою функцій *LACP min-link* та *LACP max-bundle*.

Можливості *min-links* для каналу *LACP*:

- Конфігурує мінімальну кількість портів, які повинні бути з'єднані та об'єднані в канал *LACP*.
- Запобігає активації каналу *LACP* з низькою пропускною здатністю.
- Спричиняє вимкнення каналу *LACP*, якщо занадто мало активних портів для забезпечення необхідної мінімальної пропускної здатності.

Можливості *max-bundle LACP*:

- Визначає верхню межу кількості об'єднаних портів у каналі *LACP*.
- Дозволяє використовувати порти гарячого резерву з меншою кількістю зв'язаних портів. Наприклад, у каналі *LACP* з п'ятьма портами можна вказати максимальну кількість портів – три, а два інші порти призначити як порти гарячого резерву.

Протоколи *DTP* і *CDP* надсилають і приймають пакети через фізичні порти в *EtherChannel*. Магістральні порти надсилають і приймають *LACP PDU* у *VLAN* з найнижчим номером.

У *EtherChannel* Рівня 2 перший порт у каналі, який з'являється, надає свою *MAC*-адресу *EtherChannel*. Якщо цей порт вилучається з каналу, то один з портів, що залишився, надає свою *MAC*-адресу каналу *EtherChannel*. Для *EtherChannel* 3-го рівня *MAC*-адреса виділяється активним пристроєм одразу після створення інтерфейсу за допомогою команди глобальної конфігурації інтерфейсу **port-channel**.

LACP надсилає та отримує *LACP PDU* тільки з портів, які працюють і мають увімкнений *LACP* для активного або пасивного режиму.

2.4 АВТОМАТИЗАЦІЯ НАЛАШТУВАННЯ МЕРЕЖНИХ ПАРАМЕТРІВ

2.4.1 *DHCPv4*

Протокол динамічної конфігурації вузла (*Dynamic Host Configuration Protocol, DHCP*) – це мережний протокол, який використовується для автоматизації процесу призначення *IP*-адрес та інших параметрів мережної конфігурації пристроям (наприклад, комп'ютерам, смартфонам і принтерам) у мережі. Замість того, щоб вручну конфігурувати *IP*-адресу для кожного пристрою, *DHCP* дозволяє пристроям підключатися до мережі та отримувати всю необхідну мережну інформацію, таку як *IP*-адреса, маска підмережі, шлюз за замовчуванням та адреси *DNS*-серверів, автоматично від *DHCP*-сервера.

Це полегшує керування та обслуговування великих мереж, забезпечуючи ефективну взаємодію пристроїв без конфліктів у їхніх мережних налаштуваннях. *DHCP* відіграє вирішальну роль у сучасних мережах, спрощуючи процес підключення пристроїв та ефективно керуючи мережевими ресурсами.

Протокол *DHCP* є клієнт-серверним, тобто в його роботі беруть участь клієнти *DHCP*, які потребують сервісу надання *IP*-адреси, і сервер *DHCP*. Передавання даних здійснюється за протоколом *UDP*: сервер отримує повідомлення від клієнтів на порту 67 і відправляє їх клієнтам на порту 68.

Основні компоненти *DHCP* та термінологія:

- Сервер *DHCP*. Це сервер, який зберігає *IP*-адреси та іншу інформацію, пов'язану з конфігурацією.
- Клієнт *DHCP*. Це пристрій, який отримує інформацію про конфігурацію *IP* протоколу від сервера. Це може бути мобільний телефон, ноутбук, комп'ютер або будь-який інший електронний пристрій, який потребує під'єднання до мережі.
- *DHCP*-ретранслятор. *DHCP*-ретранслятори в основному працюють як канал зв'язку між клієнтом і сервером *DHCP*, коли пакетам цього протоколу потрібно перетинати границі широкомовлення.
- Пул *IP*-адрес. Це пул або контейнер *IP*-адрес, яким володіє сервер *DHCP*. Він має діапазон адрес, які можуть бути призначені пристроям.

- Оренда *IP*-адреси. Це час, протягом якого інформація, отримана від сервера, дійсна, у разі закінчення терміну оренди, потрібно повторити процедуру заново.
- Параметри. Сервери *DHCP* можуть надавати клієнтам додаткові параметри конфігурації, такі як маска підмережі, доменне ім'я та інформація про сервер часу.
- Поновлення. Клієнти *DHCP* можуть подавати запит на поновлення оренди до закінчення терміну дії, щоб гарантувати, що вони і надалі матимуть дійсну *IP*-адресу та інформацію про конфігурацію.
- Обхід відмови. Сервери *DHCP* можна налаштувати на обхід відмов, коли два сервери працюють разом, щоб забезпечити надмірність і гарантувати, що клієнти завжди зможуть отримати *IP*-адресу та інформацію про конфігурацію, навіть якщо один сервер вийде з ладу.
- Ведення журналів аудиту. Сервери *DHCP* можуть вести журнали аудиту всіх транзакцій *DHCP*, надаючи адміністраторам можливість бачити, які пристрої використовують які *IP*-адреси і коли призначаються або поновлюються договори оренди.

Структура пакету *DHCP* показана на рисунку 2.12.

Код операції	Тип фізичної адреси	Довжина фізичної адреси	Кількість переходів
Ідентифікатор транзакції			
Кількість секунд		Прапорець	
IP-адреса клієнта			
Ваша IP-адреса			
IP-адреса сервера			
IP-адреса шлюзу			
Фізична адреса клієнта			
Ім'я сервера			
Ім'я завантажувального файлу			
Параметри			

Рисунок 2.12 – Формат пакету *DHCP*

Самі ж поля пакету *DHCP* мають такі призначення.

Код операції. Вказує на загальний тип повідомлення. Це або *DHCP Discover* або *DHCP Offer*.

Тип фізичної адреси. Тут вказується фізична адреса якої технології використовується для адресації в мережі.

Довжина фізичної адреси. Це 8-бітне поле, що визначає довжину фізичної адреси в байтах, наприклад, для *Ethernet* це значення дорівнює 6.

Кількість переходів. Це 8-бітне поле, що визначає максимальну кількість переходів, які може пройти пакет.

Ідентифікатор транзакції. Це 4-байтне поле, що встановлюється клієнтом і використовується для зіставлення відповіді із запитом. Сервер повертає те саме значення у своїй відповіді.

Кількість секунд. Це 16-бітне поле, яке вказує на кількість секунд, що минули з моменту початку процесу отримання адреси клієнтом.

Прапорець. Це 16-бітне поле, в якому використовується лише крайній лівий біт, а решта бітів повинні бути встановлені в 0. Крайній лівий біт визначає примусову широкомовну відповідь від сервера.

IP-адреса клієнта. Це 4-байтне поле, яке містить IP-адресу клієнта. Якщо клієнт не має цієї інформації, це поле має значення 0.

Ваша IP-адреса. Це 4-байтне поле, яке містить IP-адресу клієнта. Заповнюється сервером за запитом клієнта.

IP-адреса сервера. Це 4-байтне поле, що містить IP-адресу сервера. Заповнюється сервером у повідомленні-відповіді.

IP-адреса шлюзу. Це 4-байтне поле, що містить IP-адресу маршрутизатора. Заповнюється сервером у повідомленні-відповіді.

Фізична адреса клієнта. Хоча сервер може отримати цю адресу з кадру, надісланого клієнтом, ефективніше, якщо адреса буде вказана клієнтом у повідомленні запиту в явному вигляді.

Ім'я сервера. Це 64-байтне поле, яке необов'язково заповнюється сервером у пакеті відповіді. Воно містить рядок з нульовим закінченням, що складається з доменного імені сервера.

Ім'я завантажувального файлу. Це 128-байтне поле, яке може бути додатково заповнене сервером у пакеті-відповіді. Воно містить рядок з нульовим завершенням, що складається з повного шляху до завантажувального файлу.

Параметри. Це 64-байтне поле з подвійним призначенням. ІТ-спеціалісти можуть вносити або додаткову інформацію, або певну інформацію про постачальника. Поле використовується тільки у повідомленні-відповіді. Сервер використовує число у форматі IP-адреси зі значенням 99.130.83.99.

Надання інформації про IP-адреси користувацьким пристроям – одне з найважливіших завдань, яке виконують *DHCP*-сервери в мережах. Ці завдання виконуються одним з трьох способів:

1. Ручне призначення IP-адрес. У цьому типі розподілу мережний адміністратор призначає користувачам IP-адреси з *DHCP*-сервера шляхом зіставлення

їх з фізичною *MAC*-адресою клієнта, а потім *DHCP*-сервер передає цю інформацію клієнтам.

2. Автоматичний розподіл *IP*-адрес. У цьому режимі *DHCP*-сервер призначає клієнтам *IP*-адреси з пулу для постійного використання. Ці адреси не змінюються, якщо адміністратор не налаштує по-іншому.

3. Динамічний розподіл *IP*-адрес. У цьому режимі адміністратор налаштовує пул адрес, які можуть бути призначені клієнтам. Потім клієнти запитують інформацію про *IP*-адреси у *DHCP*-сервера, і їм надається *IP*-адреса та інша інформація про адресацію на певний період часу, після закінчення якого *IP*-адреса повертається до пулу *DHCP*, і клієнт повинен запросити іншу *IP*-адресу.

На рисунку 2.13 показано процес отримання клієнтом *IP*-адреси від сервера *DHCP*.

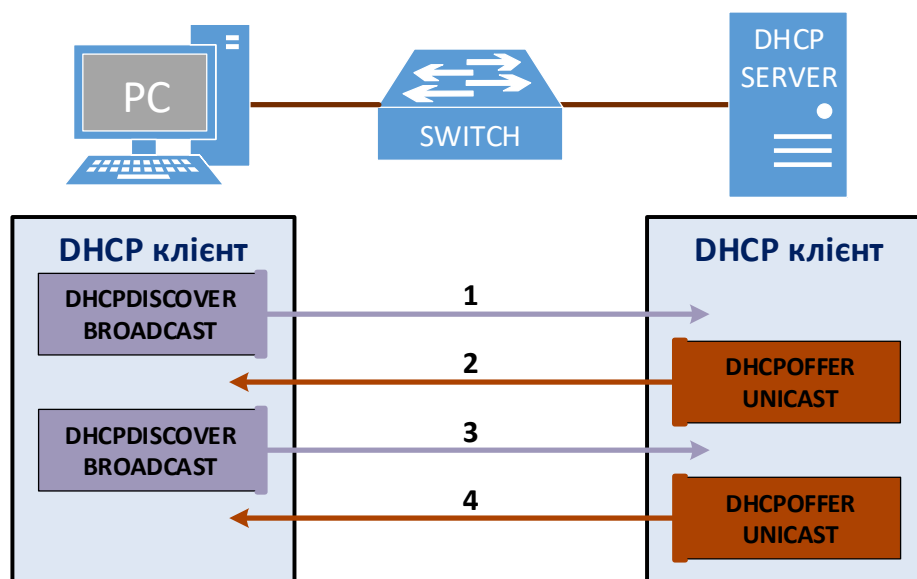


Рисунок 2.13 – Процес отримання клієнтом *IP*-адреси від сервера *DHCP*

Повідомлення, якими обмінюються пристрої для отримання автоматичної конфігурації *IP*-протоколу, мають таке призначення.

1. ***DHCPDISCOVER***. Коли клієнт завантажується, він спочатку надсилає широкомовне повідомлення, щоб спробувати виявити наявність серверів *DHCP*. Оскільки клієнт не має налаштованої *IP*-адреси, він використовує для зв'язку глобальну широкомовну адресу.

2. ***DHCPOFFER***. Коли *DHCP*-сервер отримує повідомлення від клієнта, він шукає у своєму пулі *IP*-адресу, яку може надати клієнту. Потім він додає інформацію про *MAC*-адресу клієнта та *IP*-адресу, яку він може надати в оренду, до таблиці *ARP*. Після цього сервер надсилає цю інформацію клієнту у вигляді повідомлення *DHCPOFFER*. Це повідомлення зазвичай є одно-адресним, оскільки сервер вже знає *MAC*-адресу клієнта.

3. ***DHCPREQUEST***. Коли клієнт отримує повідомлення *DHCPOFFER*, він надсилає повідомлення назад на сервер *DHCP* із запитом на отримання додаткової інформації про час оренди *IP*-адреси та перевірку. Повідомлення, яке надсилається, є *DHCPREQUEST*, це повідомлення повідомляє серверу, що він прийме

надіслану *IP*-адресу, а також перевіряє, чи все ще дійсна *IP*-адреса, надіслана сервером.

4. **DHCPACK**. Коли *DHCP*-сервер отримує *DHCP*-запит від клієнта, він підтверджує оренду і створює нове *ARP*-співставлення з *IP*-адресою, яку він призначив клієнту, і *MAC*-адресою клієнта. Потім він надсилає це повідомлення клієнту як одно-адресне повідомлення **DHCPACK**.

Коли клієнт отримує це повідомлення, він додає інформацію про адресацію і зіставляє *IP*-адресу з *MAC*-адресою під час *ARP*-пошуку.

2.4.2 АВТОКОНФІГУРАЦІЯ *IPv6*

Кожен вузол в мережі *IPv6* потребує глобальної унікальної адреси для зв'язку за межами свого локального сегмента. Для цього передбачено кілька варіантів:

1. Ручне призначення – кожен вузол може бути налаштований на *IPv6*-адресу вручну адміністратором. Цей підхід не є масштабованим і має схильність до людських помилок.

2. *DHCPv6* – найпоширеніший протокол для динамічного призначення адрес вузлам. Вимагає наявності *DHCP*-сервера в мережі та додаткового налаштування.

3. *SLAAC* (Автоконфігурація адрес без збереження стану) – був розроблений як простіший і зрозуміліший підхід до автоматичної адресації *IPv6*. У своїй поточній реалізації, яку визначено в *RFC 4862*, *SLAAC* не надає вузлам адреси *DNS*-серверів, і саме тому він не є широко прийнятим на даний момент. Це механізм, який дозволяє кожному вузлу в мережі автоматично конфігурувати унікальну *IPv6*-адресу без жодного пристрою, який би відстежував, яка адреса якому вузлу призначена.

Поняття «без збереження стану» і «зі збереженням стану» у контексті присвоєння адрес означають наступне:

- Призначення адрес зі збереженням стану передбачає наявність сервера або іншого пристрою, який відстежує стан кожного призначення. Він відстежує доступність пулу адрес і вирішує конфлікти дубльованих адрес, а також реєструє кожне призначення і відстежує час закінчення терміну дії.

- Призначення адрес без збереження стану означає, що жоден сервер не відстежує, які адреси були призначені і які адреси все ще доступні для призначення. Також при призначенні адрес без збереження стану вузли відповідають за вирішення будь-яких конфліктів дублювання адрес відповідно до такої логіки: генерується *IPv6*-адреса, запускається процедура *Duplicate Address Detection (DAD)*, якщо адреса вже використовується, то генерується інша і знову запускається *DAD*.

Щоб повністю зрозуміти, як працює автоматична адресація *IPv6*, потрібно простежити кроки, які виконує вузол *IPv6* з моменту підключення до мережі до моменту, коли він отримує унікальну глобальну адресу за процедурою *SLAAC*.

Коли вузол *IPv6* підключається до мережі з підтримкою *IPv6*, перше, що він зазвичай робить, це автоматично створює собі локальну адресу. Мета цього – дозволити вузлу взаємодіяти на Рівні 3 з іншими пристроями *IPv6* в локальному сегменті. Найпоширеніший спосіб автоматичного налаштування локальної адреси – це комбінування префіксу *FE80::/64* та ідентифікатора інтерфейсу *EUI-64*, згенерованого з *MAC*-адреси інтерфейсу.

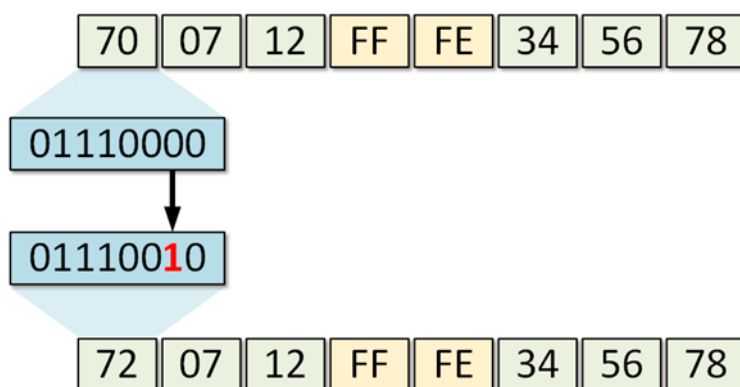
1. Нехай вузол має *MAC*-адресу мережної карти таку:

70	07	12	34	56	78
----	----	----	----	----	----

2. Потім він додає в середину адреси шістнадцяткову комбінацію *FFFE*.

70	07	12	FF	FE	34	56	78
----	----	----	----	----	----	----	----

3. Наступного кроку інвертується 7-ий біт *MAC*-адреси:



4. Поеднується префікс локальної адреси каналу *FE80::* зі створеним ідентифікатором *EUI-64* та отримується *IPv6*-адреса.

FE80::7207:12FF:FE34:5678/64

Після завершення наведених вище кроків вузол отримує повністю функціональну локальну адресу зв'язку у форматі *EUI-64*, як показано на рисунку 2.14:

```
C:\>ipconfig /all
```

```
Ethernet adapter Ethernet0:
```

```
Connection-specific DNS Suffix...: 7007.1234.5678
```

```
Physical Address.....: FE80::7207:12FF:FE34:5678
```

```
Link-local IPv6 Address.....: 0.0.0.0
```

```
IP Address.....: 0.0.0.0
```

```
Subnet Mask.....: 0.0.0.0
```

```
Default Gateway.....: 0.0.0.0
```

```
DNS Servers.....: 0.0.0.0
```

```
DHCP Servers.....: 0.0.0.0
```

```
DHCPv6 Client DUID.....: 00-01-00-01-C4-35-08-8E-70-07-12-34-56-78
```

Рисунок 2.14 – Згенерована локальна адреса з *MAC*-адреси інтерфейсу

Після автоматичної конфігурації локальної IPv6-адреси вузол має переконатися, що вона унікальна в локальному сегменті. Хоча ймовірність збігу адрес мінімальна, потрібно виконати процедуру виявлення дублікатів адрес (*Duplicate Address Detection, DAD*).

DAD використовує багатоадресну адресу типу пошук вузла. Після налаштування IPv6-адреси кожен вузол приєднується до групи багатоадресного розсилання, що визначена адресою у форматі `FF02::1:FFXX:XXXX`, де `XX:XXXX` – це останні 6 шістнадцяткових цифр IPv6-адреси. Це стосується всіх адрес одноадресного розсилання – як локальних, так і глобальних.

У прикладі з адресою, де останні 6 символів – `34:5678`, вузол приєднується до групи `FF02::1:FF34:5678`. На ПК з *Windows 10* це можна перевірити за допомогою відповідної команди, як показано на рисунку 2.15.

```
C:\>netsh interface ipv6 show joins
```

Interface 8: Ethernet0			
Scope	References	Last	Address
-----	-----	----	-----
0	0	Yes	ff01::1
0	0	Yes	ff02::1
0	1	Yes	ff02::c
0	2	Yes	ff02::fb
0	1	Yes	ff02::1:3
0	2	Yes	ff02::1:ff34:5678

Рисунок 2.15 – Приєднання до групи багатоадресного розсилання IPv6

З огляду на цю процедуру, якщо інший вузол має таку саму локальну адресу каналу, він також прослуховуватиме повідомлення групи багатоадресної розсилання пошуку сусідів, автоматично згенерованої з адреси – `FF02::1:FF34:5678`. Щоб перевірити це, вузол надсилає *ICMPv6*-повідомлення з адресою призначення цієї групи, і з невизначеною IPv6-адресою джерела. У *ICMPv6* частині пакета вузол записує повну адресу в поле *Target Address* (Адреса призначення). Тільки вузли, що слухають цю авто-згенеровану групу багатоадресного розсилання, відкриють пакет, інші його відхилять. Якщо ще у якогось вузла є IPv6-адреса з такими ж останніми шістьма шістнадцятковими цифрами, він перевірить у *ICMPv6*, чи збігається цільова адреса з його власною. У разі збігу вузол відповість, що ця адреса вже використовується. Якщо відповіді не буде, то адреса вважатиметься унікальною та призначається вузлу, який її для себе створив.

Після отримання локальної адреси вузол починає автоналаштування унікальної глобальної адреси через *SLAAC*. Першим кроком є надсилання *ICMPv6* повідомлення *Router Solicitation (RS)* з метою дізнатися у маршрутизатора префікс глобальної маршрутизації. Повідомлення надсилається на адресу багатоадресного розсилання для всіх маршрутизаторів `FF02::2`, а як джерело

використовується локальна адреса вузла. Тільки маршрутизатори, що входять в групу FF02::2, оброблять це повідомлення, а інші вузли його відхилять. Отримавши повідомлення *Router Solicitation*, маршрутизатор відповідає ICMPv6-повідомленням *Router Advertisement (RA)*. Воно містить глобальний префікс IPv6 та його довжину. Наприклад, префікс 2001:1234:A:B:: з довжиною /64. Отримавши *Router Advertisement*, вузол поєднує префікс 2001:1234:A:B::/64 з ідентифікатором інтерфейсу EUI-64 (7207:12FF:FE34:5678), отримуючи унікальну глобальну адресу 2001:1234:A:B:7207:12FF:FE34:5678/64. Оскільки RA надійшло від маршрутизатора, ПК встановлює його локальну адресу як шлюзу за замовчуванням.

Однак, SLAAC не надає інформацію про DNS, а без DNS багато сервісів, таких як перегляд сайтів в мережі «Інтернет», неможливі. Проте у заголовку RA є поле, призначене для вирішення цієї проблеми. На рисунку 2.16 зображено повідомлення RA з 8-бітним полем прапорців автоналаштування.

Якщо прапорець M встановлений на 1, це означає, що адреси та DNS доступні через DHCPv6. Якщо встановлено прапорець M, прапорець O можна ігнорувати, оскільки DHCPv6 надає всю доступну інформацію. Якщо тільки ж прапорець O встановлено на 1, це означає, що інформація DNS доступна через DHCPv6, а автоматичне налаштування адреси повинно бути здійснене через SLAAC. Якщо прапорці M та O не встановлено, це означає, що в сегменті немає жодного сервера DHCPv6, і процедура SLAAC має бути реалізована в повному обсязі.

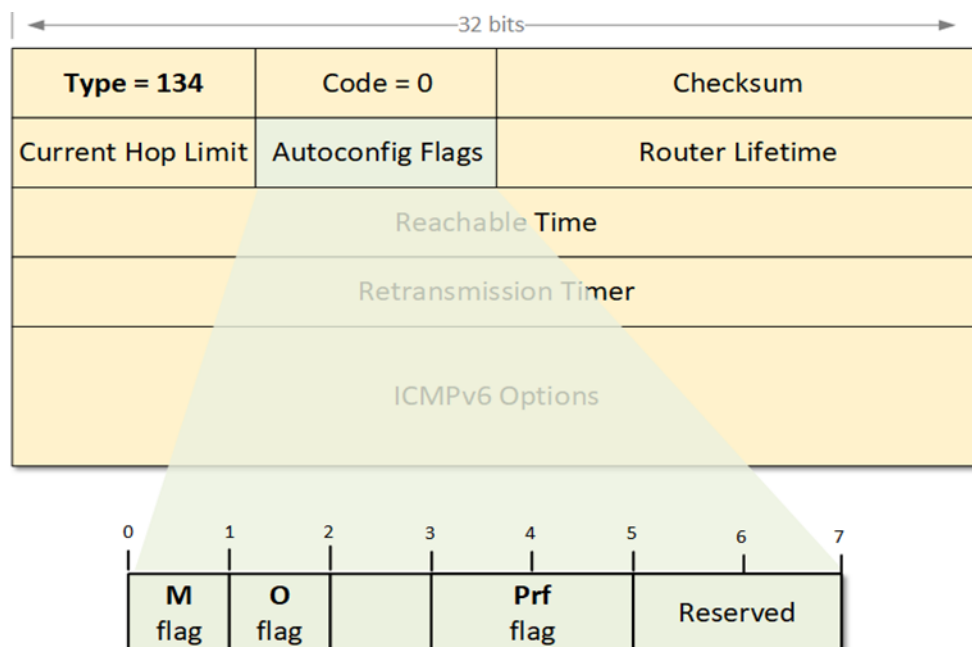


Рисунок 2.16 – Прапорці повідомлення RA

Прапорець Prf (Пріоритет маршрутизатора за замовчуванням) може бути встановлений у значення Low (1), Medium (0) або High (3). Коли вузол отримує повідомлення *Router Advertisement* від декількох маршрутизаторів, *Default Router*

Preference (DRP) використовується для визначення того, якому маршрутизатору надати перевагу як шлюзу за замовчуванням.

2.5 ПРОТОКОЛИ РЕЗЕРВУВАННЯ ОСНОВНОГО ШЛЮЗУ

Сучасні проекти мережі потребують урахування того, як мережа справляється з відмовами. З огляду на це, під час проектування необхідно передбачити максимально можливе резервування. З точки зору клієнта, за межами його локальної мережі наступною важливою частиною, з якою вони мають справу, є шлюз за замовчуванням. Щоб впоратися з резервуванням шлюзу, необхідно впровадити протокол резервування з основного шлюзу (*First Hop Redundancy Protocol, FHRP*). Існує ряд таких технологій: протокол *Cisco Hot Standby Router Protocol (HSRP)*, протокол резервування віртуального маршрутизатора (*VRRP*) і протокол балансування навантаження шлюзу (*GLBP*). Звичайно, за використання сучасного обладнання *Cisco* бажано використовувати пропрієтарні протоколи.

2.5.1 ОГЛЯД ТЕХНОЛОГІЇ РЕЗЕРВУВАННЯ ОСНОВНОГО ШЛЮЗУ (*FHRP*)

При під'єднанні вузла до мережі, він повинен отримати інформацію про параметри *IP*-адресації, налаштовану вручну чи отримати від *DHCP*-сервера. Там і буде інформація про шлюз за замовчуванням – маршрутизатор чи комутатор Рівня 3, який забезпечує з'єднання мережі із зовнішніми мережами та Інтернет. Якщо він виходить з ладу, то втрачається доступ до можливості відправляти трафік із локальної мережі. *FHRP* дозволяє використовувати резервний маршрутизатор, який підміняє основний у разі збою, при цьому клієнти продовжують використовувати ту саму *IP*-адресу шлюзу, яку отримали від *DHCP*.

2.5.2 ПРОТОКОЛ ГАРЯЧОГО РЕЗЕРВУВАННЯ МАРШРУТИЗАЦІЇ (*HSRP*)

HSRP – це пропрієтарний протокол *Cisco*, призначений для забезпечення резервування першого переходу. У випадку відмови маршрутизатора, який виконує роль шлюзу, *HSRP* дозволяє іншому маршрутизатору миттєво підмінити його, запобігаючи простою. *HSRP* використовує віртуальний шлюз (логічний маршрутизатор), який представляє реальні маршрутизатори в мережі. Якщо один маршрутизатор виходить з ладу, інший підхоплює його роль без зупинки роботи. Схему такої мережі показано на рисунку 2.17.

У цій схемі архітектури високої доступності є згадані два маршрутизатори в активному та резервному режимах з комутатором доступу та робочою станцією, налаштованими відповідно до вищезазначеної конфігурації. Нехай у мережі клієнт має *IP*-адресу 10.1.1.120 та шлюз за замовчуванням 10.1.1.1. Цей шлюз зазвичай є маршрутизатором, який дозволяє передавати трафік за межі підмережі, наприклад, в мережу «Інтернет».

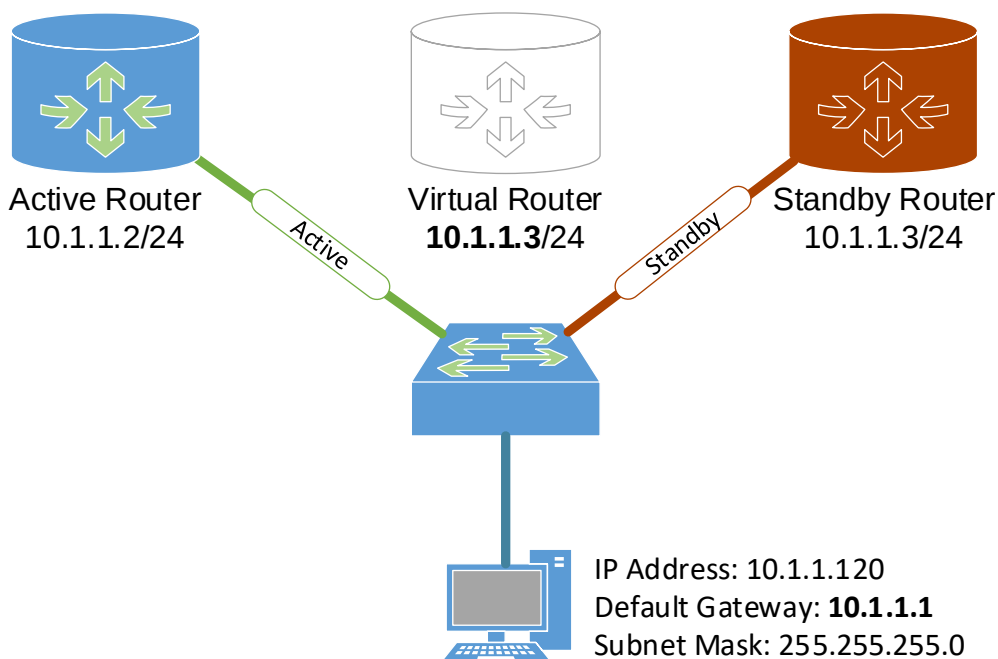


Рисунок 2.17 – Топологія мережі при застосуванні *HSRP*

HSRP надсилає повідомлення **HELLO** на групову адресу 224.0.0.2 («всі маршрутизатори») через порт *UDP* 1985.

Зафарбований сірим кольором маршрутизатор всередині схеми зазвичай називають віртуальним або логічним маршрутизатором. Це маршрутизатор, який логічно використовується комп'ютером.

У сучасних мережах більшість мережного обладнання складається з комутаторів третього рівня, які підтримують *FHRP*. Зазвичай пристрої використовують ці комутатори як шлюзи за замовчуванням. Це, як правило, багаторівневі комутатори, що виконують роль маршрутизаторів для реалізації *FHRP*. *IP*-адреса 10.1.1.1 також має відповідну *MAC*-адресу, яку ПК може отримати через *ARP*.

2.5.3 ПРОТОКОЛ РЕЗЕРВУВАННЯ ВІРТУАЛЬНОГО МАРШРУТИЗАТОРА (*VRRP*)

Цей протокол є відкритим стандартом, що описаний в *RFC* 3768 і відповідно підтримується багатьма виробниками, зокрема і *Cisco*.

Як і у випадку з *HSRP*, *VRRP* має налаштовану групу, яка містить кілька маршрутизаторів, також відомих як шлюзи. У процесі проектування мережі один зі шлюзів налаштовується як головний, а інший – як резервний або запасний.

У прикладі на рисунку 2.17 активний маршрутизатор має *IP*-адресу 10.1.1.2, а резервний – 10.1.1.3. *IP*-адреса віртуального маршрутизатора була 10.1.1.1, яка надавалася клієнтам як шлюз за замовчуванням. У *VRRP* фізична *IP*-адреса головного маршрутизатора інтерфейсу, що з'єднує підмережу,

використовується клієнтами як шлюз за замовчуванням, яка відповідно до попередньої схеми буде 10.1.1.2.

Для віртуального маршрутизатора не існує логічної IP-адреси, оскільки VRRP працює зовсім по-іншому. За лаштунками, резервний маршрутизатор групи VRRP зв'язується з головним маршрутизатором і бере на себе обов'язок пересилання трафіку, якщо головний маршрутизатор виходить з ладу з будь-яких відомих причин. IP-адреса, що використовується для VRRP, завжди належить головному маршрутизатору, який називається власником IP-адреси. Коли головний маршрутизатор відновиться, він знову візьме на себе обов'язки маршрутизації для цієї IP-адреси. VRRP використовує групову IP-адресу 224.0.0.18 для обміну повідомленнями між маршрутизаторами.

Можна мати кілька груп VRRP в одній підмережі, які можна використовувати для розподілу навантаження трафіку, що надходить із підмережі. Якщо використовується DHCP-сервер, то він має видавати половині комп'ютерів у мережі IP-адресу маршрутизатора за замовчуванням 10.0.1.2, а іншій половині – 10.0.1.3. Це можна реалізувати за допомогою опції 82 DHCP. Ідея полягає в тому, щоб призначати різні значення IP-адреси шлюзу за замовчуванням для комп'ютерів, підключених до різних комутаторів.

2.5.4 ПРОТОКОЛ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ШЛЮЗУ (GLBP)

GLBP – це ще один патентований протокол Cisco, який можна використовувати для резервування основного шлюзу. Ключова особливість GLBP, якої немає у перших двох, – це динамічне балансування навантаження.

У випадку з GLBP, на відміну від HSRP або VRRP, маршрутизатори, які існують у групі GLBP, активні та пересилають трафік. Коли налаштовано групу GLBP, один із маршрутизаторів у групі буде обрано як активний віртуальний шлюз (AVG); кожен з інших маршрутизаторів створить резервну копію AVG, якщо він вийде з ладу. AVG відповідає за присвоєння віртуальних MAC-адрес кожному з членів групи GLBP. Кожен з цих учасників називається активним віртуальним ретранслятором (AVF). AVG відповідає за відповідь на запит ARP пристроями мережі та вибір маршрутизатора групи, який буде обробляти трафік. IP-адреса шлюзу за замовчуванням однакова для всіх пристроїв підмережі; ця IP-адреса є віртуальною. Коли пристрій надсилає ARP-запит на отримання MAC-адреси, AVG відповідає однією з віртуальних MAC-адрес, оскільки саме він відповідає за створення або призначення MAC-адрес. Таким чином, AVG здатний контролювати, який маршрутизатор буде справлятися з навантаженням кожного окремого пристрою мережі. За замовчуванням маршрутизатори GLBP використовують групову адресу 224.0.0.102 для надсилання пакетів своїм сусідам кожні 3 секунди за протоколом UDP 3222. Цей протокол також було розроблено уже з підтримкою IPv6.

2.6 ТЕХНОЛОГІЇ ПОБУДОВИ ЗАХИСТУ LAN

Сучасні кіберзагрози стрімко зростають у масштабах та витонченості. А тому на сучасному етапі розвитку мережових технологій найкращою практикою вважається побудова глибинного захисту. Це комплексний підхід, який використовує комбінацію новітніх ефективних інструментів безпеки для захисту кінцевих точок, даних, додатків і мереж організації. Мета полягає в тому, щоб зупинити кіберзагрози до того, як вони відбудуться, але надійна стратегія глибинного захисту також стає на заваді атакам, які вже відбуваються, що попереджає додаткові збитки.

Одним із складників такого підходу є побудова захисту локальної мережі, зокрема, як від зовнішніх атак, так і внутрішніх, використовуючи можливості керованих комутаторів та вбудоване програмне забезпечення.

Найпоширенішими типами атак Рівня 2 є такі:

- Переповнення *CAM*-таблиці;
- Переходи між *VLAN*;
- Маніпуляції з протоколом *STP*;
- Підробка *MAC*-адрес;
- Виснаження *DHCP*.

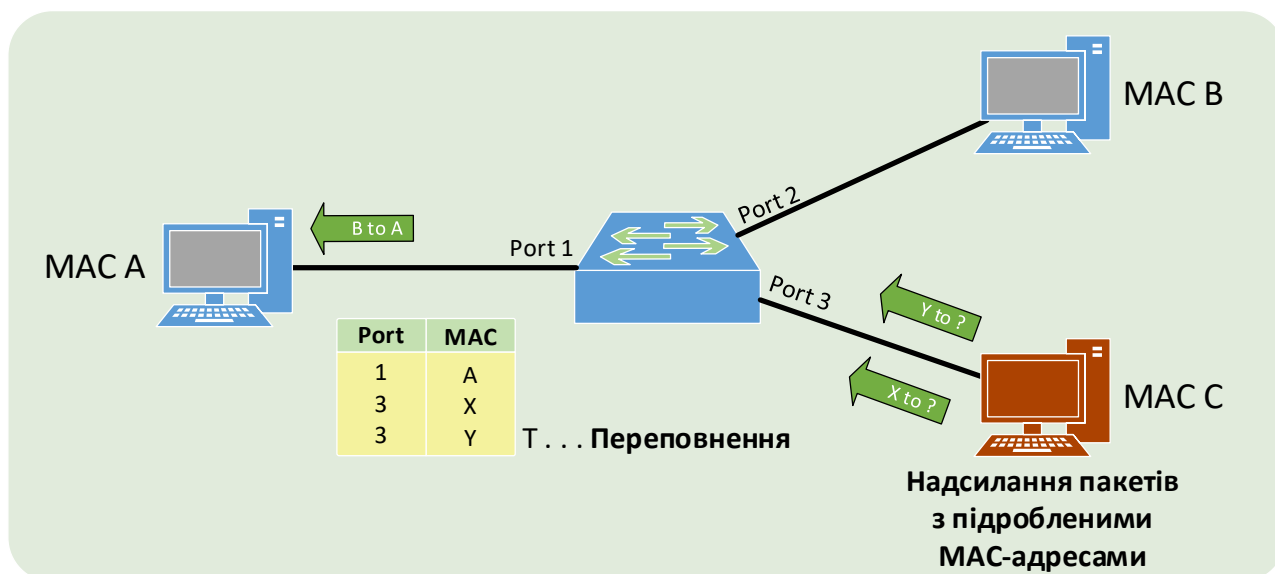
2.6.1 ЗАПОБІГАННЯ АТАКАМ ПЕРЕПОВНЕННЯ *CAM*-ТАБЛИЦЬ

Таблиця асоціативної пам'яті (*CAM*) у комутаторі зберігає інформацію про *MAC*-адреси та пов'язані з ними параметри *VLAN*. Таблиці *CAM* мають фіксований розмір. Коли комутатор отримує кадр, він шукає в таблиці *CAM* *MAC*-адресу призначення. Якщо для неї існує запис, комутатор пересилає кадр на порт, визначений у таблиці *CAM* для цієї *MAC*-адреси. Якщо ж *MAC*-адреса там відсутня, комутатор пересилає кадр на всі свої порти. Отримуючи відповідь на пересланий кадр, комутатор оновлює *CAM*-таблицю, додаючи порт, на якому було отримано відповідь.

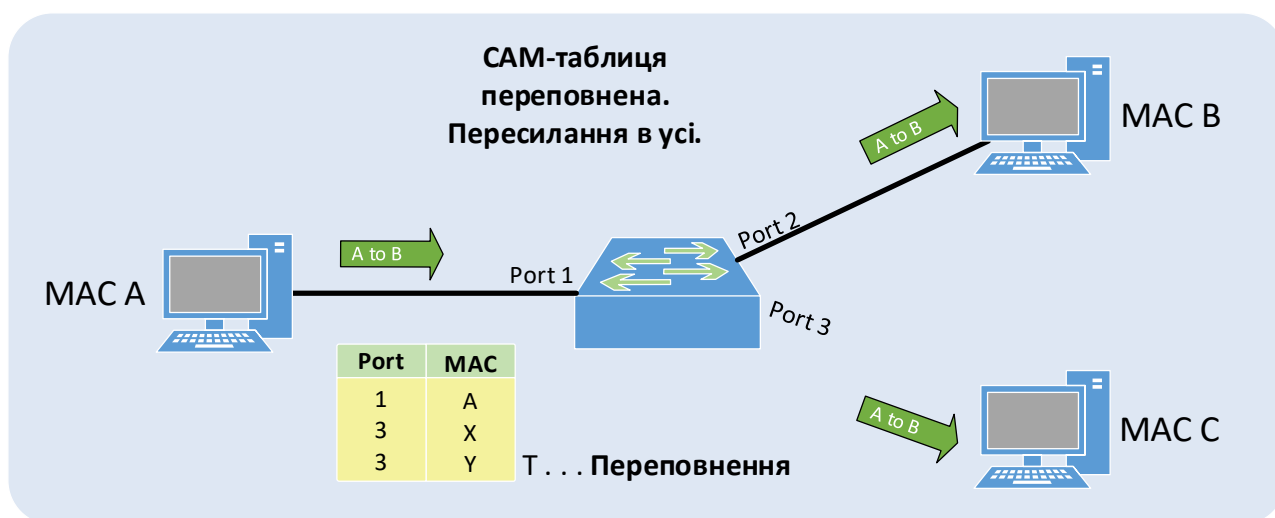
Як зазначалося раніше, *CAM*-таблиця має обмежений розмір. Комутатори *Cisco Catalyst* використовують 63 біти від джерела (*MAC*, *VLAN* тощо) і створюють 14-бітне хеш-значення. При співпадині значення є вісім комірок, в які можна помістити *CAM*-записи. Проте вони втрачають чинність після певного періоду бездіяльності. (За замовчуванням на комутаторі *Cisco Catalyst* це 5 хвилин.) За умови, що на комутатор надходить певна кількість *MAC*-адрес до закінчення терміну дії існуючих записів, таблиця *CAM* заповнюється, і нові записи не приймаються. Коли таблиця *CAM* заповнена, комутатор починає передавати пакети через усі порти. Цей сценарій називається переповненням таблиці *CAM*.

В атаці переповнення *CAM*-таблиці зломисник надсилає на комутатор тисячі фальшивих *MAC*-адрес з одного порту, що виглядає як зв'язок між дійсними вузлами (рисунки 2.18). Один з найпопулярніших інструментів для запуску цього типу атак називається *Macof*. *Macof* може генерувати 155 000 *MAC*-записів за

хвилину. Мета полягає в тому, щоб переповнити комутатор трафіком, заповнивши таблицю *SAM* фальшивими записами. Після переповнення комутатор транслює трафік без *SAM*-записів у свою власну *VLAN*, таким чином дозволяючи зловмиснику бачити трафік інших *VLAN*, який в іншому випадку не було видно.



а) заповнення таблиці *SAM* фальшивими записами



б) трансляція трафіка в усі порти

Рисунок 2.18 – Реалізація атаки переповнення *MAC*-адрес

Запобігти атакам переповнення таблиці *SAM* можна декількома способами. Одним з основних способів є налаштування безпеки портів на комутаторі. Захист портів можна реалізувати трьома способами:

- **Статичні захищені *MAC*-адреси** – порт комутатора можна вручну налаштувати на певну *MAC*-адресу пристрою, який до нього підключається.
- **Динамічні захищені *MAC*-адреси** – вказується максимальна кількість *MAC*-адрес, які будуть запам'ятовуватися на одному порту комутатора. Ці *MAC*-

адреси запам'ятовуються динамічно, зберігаються лише в адресній таблиці і видаляються після перезавантаження комутатора.

Клейкі захищені MAC-адреси – максимальна кількість MAC-адрес на певному порту може бути визначена динамічно або налаштована вручну. Ручне налаштування не рекомендується через високі адміністративні витрати. Приклеєні адреси будуть збережені в адресній таблиці і додані до поточної конфігурації.

У разі порушення безпеки можна виділити такі дії, що відбуваються з портом:

- **Protect (захист)** – якщо кількість захищених MAC-адрес досягає ліміту, дозволеного для порту, пакети з невідомими адресами джерела відкидаються до тих пір, поки не буде видалено певну кількість MAC-адрес або не буде збільшено кількість дозволених адрес. У цьому випадку не отримується сповіщення про порушення безпеки.

- **Restrict (обмеження)** – якщо кількість безпечних MAC-адрес досягає ліміту, дозволеного для порту, пакети з невідомими адресами джерела відкидаються, доки не буде видалено певну кількість безпечних MAC-адрес або не буде збільшено максимально дозволену кількість адрес. У цьому режимі на сервер *SNMP* (якщо його налаштовано) надсилається сповіщення про порушення безпеки та реєструється повідомлення *syslog*. Лічильник порушень також збільшується.

- **Shutdown (вимкнення)** – якщо відбувається порушення безпеки порту, інтерфейс переходить у стан вимкнення за помилкою, а світлодіод вимикається. Він надсилає пастку *SNMP*, записує повідомлення до *syslog* і збільшує лічильник порушень.

2.6.2 АТАКИ VLAN-ПЕРЕХОДІВ

Віртуальні локальні мережі (*VLAN*) – це простий спосіб сегментувати мережу всередині підприємства для підвищення продуктивності та спрощення обслуговування. Кожна *VLAN* складається з одного широкомовного домену. Віртуальні локальні мережі працюють за допомогою тегування пакетів ідентифікаційним заголовком. Порти обмежені для прийому тільки тих пакетів, які є частиною *VLAN*. Інформація про *VLAN* може передаватися між комутаторами в локальній мережі за допомогою магістральних портів. За замовчуванням магістральні порти мають доступ до всіх *VLAN*. Вони спрямовують трафік для декількох *VLAN* по одному фізичному каналу. На даний час стандартом є *IEEE 802.1Q*. Режим магістралі на порту комутатора можна визначити за допомогою протоколу *Dynamic Trunk Protocol (DTP)*, який автоматично визначає, чи може сусідній пристрій, підключений до порту, підтримувати магістраль. Якщо так, то він синхронізує режим магістралі на обох кінцях. Стан *DTP* на магістральному порту може бути встановлений як автоматичний, увімкнений, вимкнений, бажаний або неузгоджений. На більшості комутаторів стан *DTP* за замовчуванням – автоматичний.

Однією з проблемних областей безпеки 2-го рівня є різноманітні механізми, за допомогою яких пакети, що надсилаються з однієї *VLAN*, можуть бути перехоплені або перенаправлені в іншу *VLAN*, що і називається переходами *VLAN*. Атаки переходу *VLAN* призначені для того, щоб дозволити зловмисникам обійти пристрій 3-го рівня при передаванні даних з однієї *VLAN* в іншу. Атака працює, використовуючи переваги неправильно налаштованого магістрального порту.

Важливо відзначити, що цей тип атаки не працює на одному комутаторі, оскільки кадр ніколи не буде переадресований до місця призначення. Але в середовищі з декількома комутаторами для передачі пакета може бути використаний магістральний канал. Існує два різних типи атак з переходом через *VLAN*:

- **Підробка** комутатора – мережний зловмисник налаштовує систему так, щоб вона видавала себе за комутатор, імітуючи *802.1Q*, а також сигналізацію *DTP*. Це дозволяє зловмиснику видавати себе за комутатор з магістральним портом, а отже, за члена всіх *VLAN*.

- **Подвійне тегування** – ще одна варіація атаки переходу через *VLAN*, що передбачає тегування кадрів, що передаються, двома заголовками *802.1Q*. Більшість комутаторів сьогодні виконують тільки один рівень декапсуляції. Отже, коли перший комутатор бачить кадр з подвійним тегом, він видаляє перший тег з кадру, а потім пересилає його з внутрішнім тегом *802.1Q* на всі порти комутатора в *VLAN* зловмисника, а також на всі магістральні порти. Другий комутатор пересилає пакет на основі ідентифікатора *VLAN* у другому заголовку *802.1Q*.

На рисунку 2.19 показано перехід між *VLAN* зі сценарієм подвійного тегування.

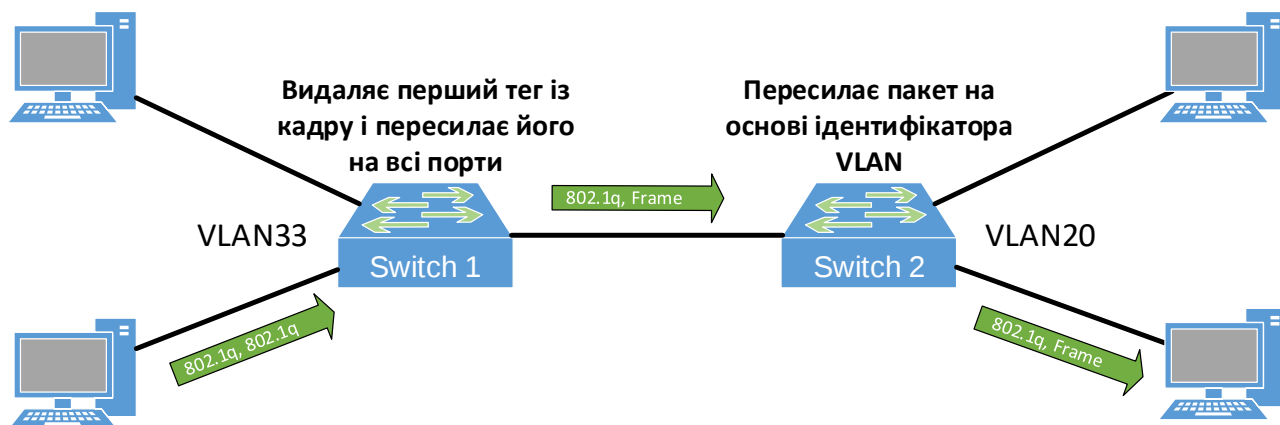


Рисунок 2.19 – Перехід між *VLAN* зі сценарієм подвійного тегування

Щоб запобігти атакам із переходом через *VLAN*, потрібно внести такі зміни до конфігурації:

- Завжди використовувати спеціальні ідентифікатори *VLAN* для всіх магістральних портів.
- Вимкнути усі невикористовувані порти та помістити їх у невикористовувану *VLAN*.

- Перевести усі користувацькі порти у нетранкінговий режим, вимкнувши *DTP* командою **switchport mode access** у режимі конфігурації інтерфейсу.
- Для з'єднань між комутаторами магістралі явно налаштувати транкінг.
- Не використовувати власну *VLAN* користувача як власну *VLAN* магістрального порту.
- Не використовувати *VLAN* 1 як *VLAN* керування комутатором.

2.6.3 АТАКИ МАНІПУЛЯЦІЇ З *STP*

STP запобігає утворенню петель у надлишковому комутованому мережному середовищі. Уникаючи петель, можна гарантувати, що широкомовний трафік не перетвориться на трафік повені.

STP – це ієрархічна деревоподібна топологія з «кореневим» комутатором на вершині. Він обирається кореневим на основі найнижчого налаштованого пріоритету (від 0 до 65 535). Коли комутатор завантажується, то починається процес ідентифікації інших комутаторів і визначення кореневого моста. Після того, як кореневий міст обрано, топологія встановлюється з точки зору з'єднань. Комутатори визначають шлях до кореневого моста, а всі надлишкові шляхи блокуються. *STP* надсилає повідомлення і підтвердження про зміну конфігурації і топології, використовуючи блоки даних мостового протоколу (*BPDU*).

Атака *STP* передбачає підміну зломисником кореневого мосту в топології. Зломисник розсилає *BPDU* про зміну конфігурації/топології, намагаючись змусити перерахувати *STP*. Розсилка *BPDU* повідомляє, що система зломисника має нижчий пріоритет мосту. Після цього зломисник може бачити різноманітні кадри, переадресовані на нього з інших комутаторів. Перерахунок *STP* також може спричинити відмову в обслуговуванні (*DoS*) в мережі, викликаючи переривання на 30-45 секунд при кожній зміні кореневого мосту. На рисунку 2.20 показано, як зломисник використовує зміни топології мережі *STP*, щоб змусити всіх обрати кореневим мостом свій вузол.

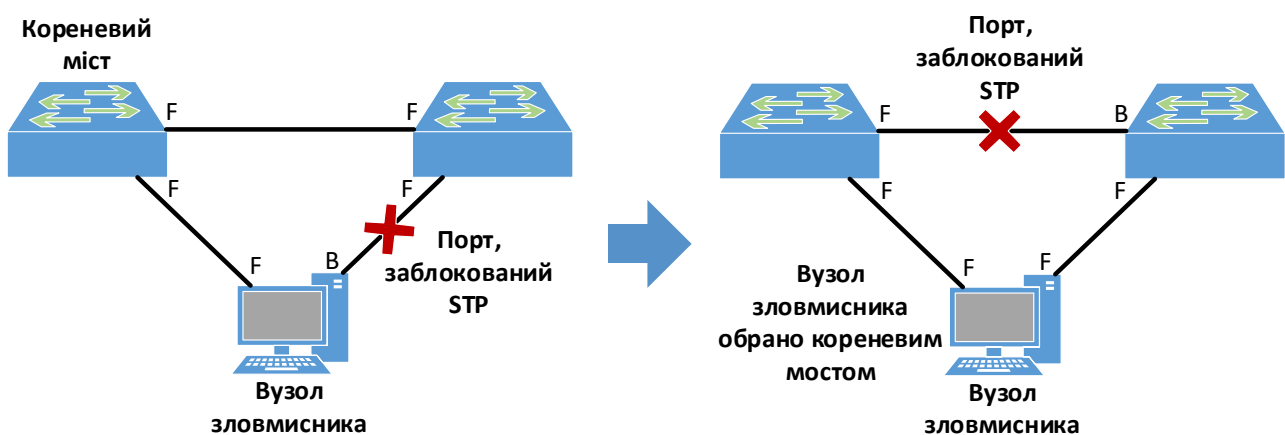


Рисунок 2.20 – Атака маніпуляції з *STP*

Щоб зменшити маніпуляції з *STP*, потрібно використати функції захисту кореневого мосту та захисту BPDU в програмному забезпеченні *Cisco IOS*. Ці команди забезпечують примусове розміщення кореневого мосту і кордонів домену *STP*. Функція захисту кореневого мосту *STP* призначена для розміщення кореневого мосту в мережі. Функція захисту *BPDU STP* використовується для забезпечення передбачуваності всієї активної топології мережі.

2.6.4 АТАКИ З ПІДМІНОЮ MAC-АДРЕС

Підміна *MAC*-адреси передбачає використання відомої *MAC*-адреси іншого вузла, що має дозвіл на доступ до мережі. Зловмисник намагається змусити цільовий комутатор переадресовувати кадри, призначені для справжнього вузла, на пристрій зловмисника. Це робиться шляхом надсилання кадру з вихідною *Ethernet*-адресою іншого вузла з метою перезаписати запис у таблиці *CAM*. Після того, як *CAM* буде перезаписано, всі пакети, призначені для справжнього вузла, будуть перенаправлені зловмиснику. Якщо справжній вузол відправляє трафік, таблиця *CAM* буде переписана знову, переміщуючи трафік назад на порт справжнього вузла. На рисунку 2.21 показано, як працює підміна *MAC*-адрес.



Рисунок 2.21 – Атака підміни *MAC*-адреси

Іншим методом підробки *MAC*-адрес є використання протоколу дозволу адрес (*Address Resolution Protocol, ARP*), який використовується для зіставлення *IP*-адрес з *MAC*-адресами, що знаходяться в одному сегменті локальної мережі. Коли вузол надсилає ширококомовний *ARP*-запит, щоб знайти *MAC*-адресу іншого вузла, *ARP*-відповідь надходить від того, адреса якого відповідає запиту. *ARP*-відповідь кешується вузлом, який надсилає запит. Протокол *ARP* також має інший метод визначення зв'язку між *IP* і *MAC*-адресами, який називається *Gratuitous ARP (GARP)*, що являє собою ширококомовний пакет, який

використовується вузлом для оголошення своєї *IP*-адреси в локальній мережі, щоб уникнути дублювання *IP*-адрес в мережі. *GARP* може бути використаний зловмисником для підміни ідентичності *IP*-адреси в сегменті локальної мережі. Зазвичай це використовується для підміни ідентичності між двома вузлами або всього трафіку до і від шлюзу за замовчуванням.

Для протидії підробці *MAC*-адрес потрібно використати команду **port-security**, щоб вказати *MAC*-адреси, підключені до певних портів; однак такий тип конфігурації має високі адміністративні витрати і схильний до помилок. Існують інші механізми, такі як таймери затримки, які можна використовувати для захисту від атак підміни *ARP*, встановлюючи час, протягом якого запис залишатиметься в *ARP*-кеші. Самих по собі таймерів затримки недостатньо для запобігання атакам. Можна поєднати їх зі зміною часу закінчення терміну дії *ARP*-кешу для всіх вузлів, але це також некеровано. Однією з рекомендованих альтернатив є використання приватних *VLAN* для пом'якшення цих типів мережних атак. Кілька інших функцій програмного забезпечення *Cisco IOS* забезпечують захист від цього типу атак: динамічна перевірка *ARP* (*DAI*) і *DHCP* snooping.

DHCP-snooping – це функція безпеки *DHCP*, яка забезпечує мережну безпеку, фільтруючи ненадійні *DHCP*-повідомлення за допомогою бази даних прив'язок *DHCP-snooping*, яку вона створює і підтримує. Ненадійне повідомлення – це повідомлення, отримане ззовні мережі або брандмауера. Коли комутатор отримує пакет через ненадійний інтерфейс і на цьому інтерфейсі або *VLAN* увімкнено *DHCP-snooping*, комутатор порівнює *MAC*-адресу джерела та апаратну адресу *DHCP*-клієнта-запитувача. Якщо адреси збігаються, комутатор пересилає пакет. Якщо адреси не збігаються, комутатор відкидає пакет. *DHCP-snooping* розглядає *DHCP*-повідомлення, що надходять з будь-якого порту користувача до *DHCP*-сервера, як ненадійні. Ненадійний порт не повинен надсилати відповіді типу *DHCP*-сервера, як-то *DHCPOFFER*, *DHCPACK* тощо.

Довірений інтерфейс – це інтерфейс, налаштований на отримання повідомлень лише з мережі.

Таблиця прив'язок *DHCP-snooping* містить таку інформацію, як *MAC*-адреса вузла (динамічна і статична), *IP*-адреса, час оренди, тип прив'язки і номер *VLAN*. База даних може містити до 8192 прив'язок.

DAI – це функція безпеки, яка перехоплює і перевіряє прив'язки *IP*-адрес до *MAC*-адрес і відкидає недійсні *ARP*-пакети. *DAI* використовує базу даних *DHCP* для перевірки прив'язок. Вона пов'язує стан довіри з кожним інтерфейсом на комутаторі. Пакети, що надходять на довірені інтерфейси, обходять всі перевірки *DAI*, а ті, що надходять на недовірені інтерфейси, проходять процес перевірки *DAI*. У типовій мережі всі порти комутатора, підключені до вузла, налаштовані як ненадійні, а порти комутатора вважаються довіреними. Для цього використовують команду **ip arp inspection trust interface** для

налаштування параметрів довіри. Коли комутатор налаштовано на *DAI*, він обмежує швидкість вхідних *ARP*-пакетів, щоб запобігти *DoS*-атакам. Швидкість за замовчуванням для ненадійного інтерфейсу становить 14 пакетів на секунду. На довірених інтерфейсах швидкість не обмежується. *DAI* використовує списки контролю доступу *ARP (ACL)* і базу даних *DHCP* для отримання списку допустимих прив'язок *IP*-адрес до *MAC*-адрес. Списки *ARP ACL* мають пріоритет над записами у базі даних прив'язок *DHCP*, але їх потрібно налаштувати вручну. Використовується команда глобальної конфігурації **ip arp inspection filter** для налаштування списків *ARP ACL*. Комутатор відкине пакет, який заборонено у списках *ARP ACL*, навіть якщо у базі даних *DHCP-snooping* для нього є дійсне прив'язування. Коли комутатор відкидає пакет, він записується у буфер і генерує системне повідомлення. Можна використати **ip arp inspection log-buffer**, щоб налаштувати кількість буферів і кількість записів, необхідних для генерації системних повідомлень через певний інтервал часу.

2.6.4 АТАКИ ВИСНАЖЕННЯ *DHCP*

Сервер *DHCP* динамічно призначає *IP*-адреси вузлам у мережі. Адміністратор створює пули адрес, доступних для призначення. З адресами пов'язаний час оренди. *DHCP* – це стандарт, визначений у *RFC 2131*.

Атака виснаження *DHCP* працює шляхом широкомовної розсилки *DHCP*-запитів з підробленими *MAC*-адресами. Цей сценарій досягається за допомогою таких інструментів атаки, як *gobbler*, який переглядає всю область *DHCP* і намагається орендувати всі доступні в ній *DHCP*-адреси. Це проста атака «виснаження ресурсів», схожа на атаку *SYN*-флуд. Зловмисник може створити неавторизований *DHCP*-сервер і відповідати на нові *DHCP*-запити від клієнтів у мережі. Це може призвести до атаки «людина посередині».

Методи, що використовуються для запобігання атакам підміни *MAC*-адрес, можуть також запобігти виснаженню *DHCP* за допомогою функції *DHCP snooping*. Впровадження *RFC 3118, Автентифікація для DHCP-повідомлень*, також допоможе запобігти цьому типу атак.

Також можна обмежити кількість *MAC*-адрес на порту комутатора, що є стратегією пом'якшення наслідків переповнення *SAM*-таблиці, щоб запобігти атакам *DHCP*-виснаження.

Інші функції комутатора *Cisco Catalyst*, такі як *IP source guard*, також можуть забезпечити додатковий захист від атак. Функція *IP source guard* спочатку блокує весь *IP*-трафік, окрім пакетів *DHCP*, перехоплених процесом *DHCP snooping*. Коли клієнт отримує дійсну *IP*-адресу від сервера *DHCP*, до порту застосовується *ACL*. Цей *ACL* обмежує трафік від клієнта тими *IP*-адресами, які вказані у прив'язці.

Одним із способів запобігти відповіді неавторизованого *DHCP*-сервера на *DHCP*-запити є використання списків *VLAN ACL (VACL)*. Можна використовувати *VACL*, щоб обмежити відповіді *DHCP* для легітимних *DHCP*-серверів і заборонити їх для всіх інших. Цей тип налаштувань слід використовувати, якщо мережа не підтримує *DHCP snooping*.

2.7 БЕЗДРОВОТА МЕРЕЖЕВА ІНФРАСТРУКТУРА

2.7.1 ОГЛЯД ТЕХНОЛОГІЇ БЕЗДРОВОГО ЗВ'ЯЗКУ

Бездротові з'єднання, які сьогодні широко використовуються для різноманітних задач, є невідчутними на дотик, а оскільки бездротові технології розвиваються так швидко, зрозуміло, що зростає і кількість запитань, пов'язаних із їхнім розвитком. Оскільки бездротові мережі стають невід'ємною частиною повсякденного життя, забезпечуючи роботу всього – від смартфонів і ноутбуків до «розумних» будинків і промислової автоматизації, корисно розглянути різні типи бездротових мереж.

Однією з беззаперечних переваг бездротових мереж є також підтримка абонентів, які знаходяться в русі, що дає можливість виконувати різноманітні завдання без дротових обмежень. Бездротова інфраструктура може адаптуватися до швидкозмінних потреб і технологій.

Проте бездротові мережі мають і свій ряд недоліків. Це:

- Вплив радіочастотних і електромагнітних перешкод.
- Коливання та нестабільність сигналу.
- Можливість перехоплення сигналів в ефірі.
- Вплив погодних умов тощо.

Бездротові мережі використовують радіохвилі для з'єднання та обміну даними з пристроями в мережі. Залежно від кількості під'єднаних пристроїв і розміру мережі, бездротові мережі можна розділити на чотири типи:

- Бездротова локальна мережа (*WLAN*). Локальна мережа з'єднує два або більше пристроїв за допомогою бездротової технології.
- Бездротові міські мережі (*WMAN*). *WMAN* з'єднує дві або більше бездротових локальних мереж масштабу міста.
- Бездротова глобальна мережа (*WWAN*). Мережа *WWAN* охоплює великі сусідні міста. *WWAN* підходить для організації національних і глобальних комунікацій.
- Бездротова персональна мережа (*WPAN*). *WPAN*, як випливає з назви, з'єднує пристрої на коротких відстанях, зазвичай у межах досяжності людини (її приватна мережа).
- Стільникова мережа.

Оскільки бездротові мережі передають дані через радіочастоти, вони регулюються тими самими законами, які використовуються для регулювання таких речей, як радіостанції *AM/FM*. Державні органи, які опікуються розподілом радіочастот, регулюють і використання пристроїв бездротових локальних мереж, а Інститут інженерів з електротехніки та електроніки (*IEEE*) створює стандарти на основі частот, які дозволені для загального користування.

Діапазони 900 МГц і 2,4 ГГц називають діапазонами промислового, наукового і медичного призначення, а діапазон 5 ГГц – діапазоном неліцензованої національної інформаційної інфраструктури. Звідси випливає, що якщо потрібно розгорнути бездротову мережу в діапазоні за межами трьох загальнодоступних діапазонів, то треба отримати спеціальну ліцензію для цього.

На даний час найпоширенішою технологією побудови бездротових мереж є *WI-FI* (торговельною маркою Wi-Fi володіє **Wi-Fi Alliance** – некомерційна організація), що затверджена в стандартах *IEEE 802.11*. Цей набір стандартів (таблиця 2.6) регулює використання частотних діапазонів (0,9; 2,4; 3,6; 5; 6 та 60 ГГц), методів передачі та обробки радіосигналів (модуляція, кодування, шифрування), типів обладнання, кількість антен тощо.

Таблиця 2.6 – Стандарти технології *WI-FI*

Стандарт <i>IEEE</i>	Частотний діапазон	Опис
802.11a	5 ГГц	Швидкість до 54 Мбіт/с. Мультиплексування з ортогональним частотним розділенням каналів (<i>OFDM</i>).
802.11b	2,4 ГГц	Швидкість до 11 Мбіт/с. Комплементарна кодова маніпуляція (<i>ССК</i>). 13 каналів.
802.11g	2,4 ГГц	Швидкість до 54 Мбіт/с. Зворотна сумісність з 802.11b. 13 каналів.
802.11n	2,4 ГГц, 5 ГГц	Швидкість від 150 до 600 Мбіт/с. Підтримка технології <i>MIMO</i> . Зворотна сумісність з 802.11a/b/g
802.11ac	5 ГГц	Швидкість від 450 Мбіт/с до 1,3 Гбіт/с. Підтримка технології <i>MIMO</i> до 8 антен. Зворотна сумісність з 802.11a/n
802.11ax	2,4 ГГц, 5 ГГц	Швидкість до 11 Гбіт/с. Підтримка технології <i>MU-MIMO</i> . Здатність використовувати частоти 1 ГГц і 7 ГГц, коли вони стають доступними

2.7.2 ОСНОВНІ КОМПОНЕНТИ БЕЗДРОТОВИХ МЕРЕЖ

Відповідно до структури будь-якої мережі, до її складу входять кінцеві пристрої (джерела і отримувачі інформації), проміжні пристрої та відповідно середовище передачі, яке для бездротових мереж визначене у вигляді радіохвиль. Для з'єднання з бездротовою мережею кінцеві пристрої повинні бути оснащені бездротовими мережними адаптерами – вбудованими або ж під'єднаними до пристрою за допомогою, наприклад, інтерфейсу *USB*.

Як проміжні пристрої у бездротових мережах можуть слугувати точки доступу (*AP*), бездротові маршрутизатори та ретранслятори радіосигналу.

Бездротовий маршрутизатор зазвичай використовується для побудови невеликих мереж бездротового доступу типу *SOHO*. В таких мережах зазвичай навантаження становить до 10 бездротових клієнтів, а розширити зону покриття буває доцільно за використання одного чи двох ретрансляторів сигналу. Налаштування маршрутизаторів типу *SOHO* зазвичай не становить труднощів.

Якщо ж необхідно організувати досить потужну, відмовостійку та безшовну корпоративну бездротову мережу, то в такому випадку на допомогу прийде використання точок доступу, які зазвичай під'єднуються до дротової мережі через комутатор з підтримкою живлення через *Ethernet (PoE)*.

Існує кілька типів точок доступу, які можна використати в бездротових мережах, що відрізняються за типом керування ними. І хоча деякі з них виробники дозволяють конвертувати з одного типу в інший, кожен має свої особливості роботи.

Автономні точки доступу (*AP*) – це бездротові точки доступу, що працюють незалежно. Більше підходять для невеликих мереж або середовищ з низькими вимогами до управління бездротовими мережами. Серед переваг можна виділити такі: просте розгортання, низька вартість, не потребують додаткового програмного забезпечення чи обладнання.

Полегшені точки доступу (*LAP, Lightweight AP*) керуються централізовано контролером бездротової мережі (*WLC*), що підходить для середніх і великих мереж, особливо тих, що вимагають великої кількості точок доступу. Переваги такого використання такі: централізована конфігурація, уніфіковані політики та безпека, оптимізована продуктивність і зручність роботи користувачів, балансування навантаження, безперебійний роумінг тощо.

Однією з новітніх технологій є централізоване керування точками доступу через хмару. Провідні виробники мережного обладнання надають вже такі послуги, надаючи хмарну платформу з усім необхідним програмним забезпеченням. Хмарне керування підходить для мереж будь-якого масштабу, особливо тих, що розподілені по різних місцях або потребують віддаленого керування. До інших переваг додаються також досить потужна масштабованість бездротової мережі, автоматизоване обслуговування, доступ до розширених інструментів аналітики.

Серед інших важливих компонент бездротової мережі, необхідно також загадати основні компоненти прийому та передавання радіосигналів – антени. Більшість пристроїв технології *WI-FI* використовують всеспрямовані антени, що забезпечують покриття на 360 градусів. Проте, якщо потрібно направити радіосигнал в певну область, то це можна зробити за допомогою спрямованих антен. Для збільшення пропускної здатності може використовуватися технологія множинного входу і множинного виходу (*MIMO*), що підтримує до восьми передавальних і приймальних антен.

2.7.3 ПРИНЦИПИ ФУНКЦІОНУВАННЯ БЕЗДРотовИХ МЕРЕЖ

Для поєднання пристроїв між собою в бездротових мережах стандарту *IEEE 802.11* можуть бути використані кілька режимів, які фактично і визначають топологію мережі.

Режим *Ad-hoc*. У цьому режимі бездротові пристрої зв'язуються безпосередньо один з одним без використання точки доступу або маршрутизатора. Такі мережі зазвичай використовуються для однорангового зв'язку або для створення тимчасових мереж, коли проміжні пристрої недоступні. Один із варіантів такої топології – це режим прив'язки, коли для доступу до мережі Інтернет використовується смартфон чи планшет з під'єднанням до стільникових даних. Стандарт *IEEE 802.11* визначає мережу *ad-hoc* як незалежний базовий набір послуг (*IBSS, Independent Basic Service Set*).

Інфраструктурний режим. У цьому режимі бездротові пристрої зв'язуються один з одним через точку доступу (*AP*) або маршрутизатор, що діє як центральний вузол для всіх бездротових пристроїв і забезпечує під'єднання до Інтернету або дротових мереж.

У стандарті бездротових локальних мереж *IEEE 802.11* для інфраструктурного режиму визначено поняття набору послуг – це група бездротових мережевих пристроїв, які мають спільний ідентифікатор набору послуг (*SSID, service set identifier*) – як правило, мітку природною мовою, яку користувачі сприймають як назву мережі. Набір послуг утворює логічну мережу вузлів, що працюють зі спільними мережними параметрами канального рівня; вони утворюють один логічний сегмент мережі.

Набір послуг може бути двох типів: базовий набір послуг (*BSS*), або розширений набір послуг (*ESS*).

Базовий набір послуг – це підгрупа в межах набору послуг, що складається з пристроїв, які мають спільні характеристики доступу до середовища на фізичному рівні (наприклад, радіочастоту, схему модуляції, налаштування безпеки), завдяки чому вони об'єднані в бездротову мережу. Базовий набір послуг визначається ідентифікатором базового набору послуг (*BSSID*), спільним для всіх пристроїв, що входять до нього. *BSSID* – це 48-бітова мітка, яка відповідає стандарту *MAC-48*. Хоча пристрій може мати кілька *BSSID*, зазвичай кожен *BSSID*

асоціюється щонайбільше з одним базовим набором послуг. Базовий набір послуг не слід плутати з зоною покриття точки доступу, відомою як зона базового обслуговування (BSA).

Розширений набір послуг (ESS) – це бездротова мережа, створена декількома точками доступу, яка виглядає для користувачів як єдина, безшовна мережа, наприклад, мережа, що охоплює будинок або офіс, що занадто великі для покриття однією точкою доступу. Кожен ESS ідентифікується за SSID. Це набір з декількох інфраструктурних BSS в загальному логічному сегменті мережі (тобто в одній IP-підмережі та VLAN). Ключовим моментом концепції є те, що на підрівні керування логічним зв'язком (LLC) BSS, які беруть участь, виглядають як єдина мережа. Таким чином, з точки зору підрівня LLC, AP в межах ESS можуть взаємодіяти одна з одною, а мобільні пристрої можуть прозоро переходити від одного BSS до іншого (в межах одного ESS). ESS дають змогу поширення таких служб, як централізована автентифікація. З точки зору каналного рівня, всі станції в межах ESS знаходяться на одному каналі, і перехід від одного BSS до іншого є прозорим для LLC.

WLAN використовують спільне середовище передавання інформації. Це означає, що бездротові клієнти можуть передавати і приймати дані по одному радіоканалу. Для цього використовується метод доступу CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), який дає змогу уникнення колізій (зіткнень) пакетів результаті використання спільного середовища передавання.

Для передавання даних бездротовий клієнт виконує таке:

- Перед тим, як передавати дані, пристрій прослуховує середовище передавання, щоб перевірити, чи не зайняте воно. Це означає, що пристрій визначає, чи є інша активна передача на тому ж каналі.

- Якщо середовище передавання вільне, пристрій не передає дані одразу, а чекає певний випадковий проміжок часу. Це допомагає уникнути колізій, якщо інший пристрій також вирішив передавати дані в той самий момент.

- Після завершення цього часу очікування пристрій знову перевіряє середовище. Якщо воно ще вільне, дані передаються. Якщо ні – процес повторюється.

- Після успішної передачі пакета пристрій-одержувач посилає спеціальний пакет підтвердження (ACK) назад до відправника. Якщо підтвердження не отримано, вважається, що відбулася колізія, і дані передаються знову.

- Для уникнення колізій в мережах з високим навантаженням або коли станції не можуть почути одна одну, використовується механізм RTS/CTS.

- Перед передаванням даних пристрій посилає сигнал RTS (*Request to Send*) до точки доступу або іншого пристрою. Якщо середовище вільне, той у відповідь відправляє сигнал CTS (*Clear to Send*), що дозволяє передати дані.

Рівень MAC забезпечує функціональність для кількох завдань, як-то управління доступом до середовища передавання даних, а також може пропонувати підтримку роумінгу, автентифікації та енергозбереження. Основними

послугами, що надаються *MAC*, є обов'язкова послуга асинхронної передачі даних і необов'язкова послуга обмеженої за часом передачі даних. Стандарт *IEEE 802.11* визначає два підрівні *MAC*:

1. Розподілена функція координації (*DCF*). *DCF* використовує *CSMA/CA* як метод доступу, оскільки бездротова мережа не може реалізувати *CSMA/CD*. Вона пропонує лише асинхронний сервіс.

2. Функція координації точок (*PCF*). *PCF* реалізується поверх *DCF* і в основному використовується для передачі часового сервісу. Вона використовує централізований, безконфліктний метод доступу з опитуванням. Пропонує як асинхронне, так і обмежене в часі обслуговування.

Рисунок 2.22 демонструє структуру кадру *IEEE 802.11*. Кожен прямокутник представляє окреме поле в кадрі зі своїм розміром і назвою.

Керування кадром	Тривалість /ID	Адреса 1	Адреса 2	Адреса 3	Керування послідовністю	Адреса 4	Корисне навантаження	Послідовність перевірки кадру
2 байти	2 байти	6 байтів	6 байтів	6 байтів	2 байти	6 байтів	0-2312 байтів	4 байти

Рисунок 2.22 – Структура кадру *IEEE 802.11*

Керування кадром. Це поле складається з кількох підполів, які визначають тип кадру, версію протоколу, керування фрагментацією, захистом.

Тривалість/ID. Поле містить інформацію про час, який необхідний для завершення поточної передачі або спеціальні ідентифікатори.

Адреса 1. Містить *MAC*-адресу отримувача кадру. **Адреса 2.** Містить *MAC*-адресу відправника кадру. **Адреса 3.** Містить *MAC*-адресу *AP* або кінцевого пункту призначення, залежно від типу кадру.

Керування послідовністю. Складається з двох підполів: номер фрагменту (4 біти) та порядковий номер (12 біт). Вони використовуються для відстеження послідовності та фрагментації кадрів.

Адреса 4. Використовується лише в кадрах типу **WDS** (Wireless Distribution System) і містить додаткову *MAC*-адресу.

Корисне навантаження. Змінна частина кадру, яка містить корисну інформацію, наприклад, *IP*-пакет в кадрах типу даних.

Послідовність перевірки кадру. Використовується для перевірки цілісності кадру за допомогою алгоритму *CRC-32*.

Щоб встановити зв'язок в мережі *WI-FI*, бездротовий пристрій використовує методи сканування для пошуку точки доступу чи маршрутизатора.

При активному скануванні бездротовий пристрій транслює зондувальний сигнал на кожен канал у своєму частотному діапазоні і чекає на відповідь точки доступу. Так клієнт ініціює процес пошуку, показаний на рисунку 2.23.

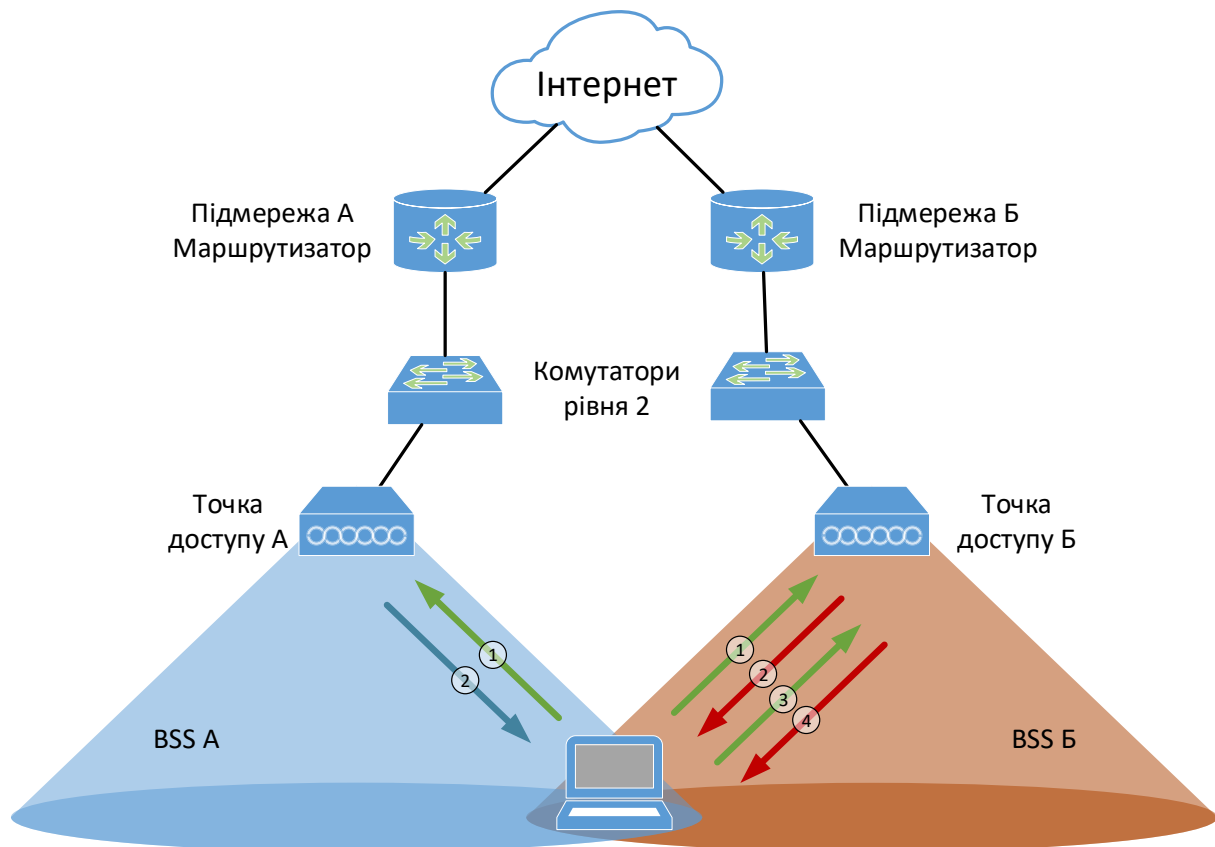


Рисунок 2.23 – Процес активного сканування

Послідовність встановлення з'єднання при активному скануванні:

Крок 1. Пристрій безперервно надсилає сигнал у пошуках точки доступу.

Крок 2. Кожна точка доступу відповідає, надсилаючи клієнту свої *SSID* та іншу інформацію.

Крок 3. Клієнт може надіслати запит на підключення до точки доступу Б, яка може мати кращий сигнал.

Крок 4. Точка доступу Б надсилає повідомлення з підтвердженням та згодою на асоціацію. Пристрій може отримати повний доступ до мережі через точку доступу Б.

При пасивному скануванні пристрій користувача простоює і не ініціює метод пошуку. Він лише прослуховує канали в діапазоні частот доступних точок доступу. Точка доступу постійно транслює кадр-маячок у межах своєї зони обслуговування. Саму ж послідовність утворення з'єднання можна представити так (рисунок 2.24):

Крок 1. Обидві точки доступу постійно транслюють кадр маячка в межах своєї зони обслуговування. Цей спеціальний кадр містить *SSID* точки доступу та іншу інформацію. Бездротовий пристрій прослуховує всі канали на предмет наявності маячків.

Крок 2. Пристрій може вибрати точку доступу, з якою він може встановити зв'язок. Припустимо, він обирає точку доступу А. Він надсилає запит на встановлення зв'язку з точкою доступу А.

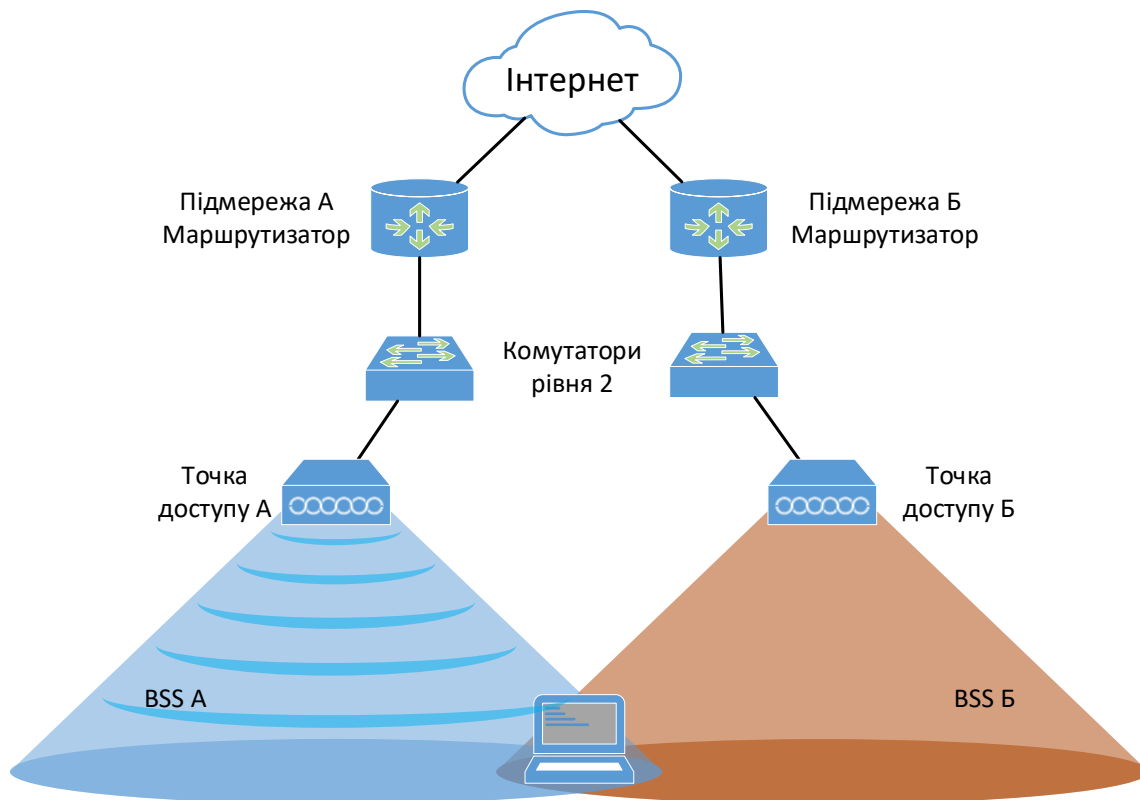


Рисунок 2.24 – Процес пасивного сканування

Крок 3. Точка доступу А відповідає згодою на встановлення зв'язку. Тепер пристрій має повний доступ до мережі.

2.7.4 ПРОТОКОЛ КЕРУВАННЯ ТОЧКАМИ ДОСТУПУ CAPWAP

CAPWAP (Control And Provisioning of Wireless Access Points) – це протокол, розроблений для керування AP з централізованого контролера. *CAPWAP* стандартизований *IETF* у *RFC 5415* і використовується в бездротових мережах для централізованого управління кількома AP.

Тунель *CAPWAP* – це логічне з'єднання між бездротовим контролером (*WLC*) і точкою доступу (*AP*), що використовується для передавання керуючого трафіку та даних користувачів у бездротових мережах. *CAPWAP* тунель забезпечує централізоване управління і безпеку передачі даних, дозволяючи контролеру і точкам доступу взаємодіяти незалежно від фізичного місця розташування.

До основних характеристик *CAPWAP* тунелю відносять:

1. Два типи трафіку:

- Керуючий трафік (*Control Plane*) – передає команди керування та налаштування між *WLC* і *AP*. Передається через порт *UDP 5246*. Він містить команди для налаштування *AP*, автентифікацію, моніторинг стану тощо.
- Користувацький трафік (*Data Plane*) – передає дані користувачів. Передається через порт *UDP 5247*. Це дані, які проходять через *AP* від кінцевих користувачів до їхніх цільових пунктів.

2. Інкапсуляція:

Дані, що проходять через тунель *CAPWAP*, інкапсулюються в *CAPWAP* пакети, що забезпечує їх захист та ізоляцію від інших мережних протоколів.

3. Шифрування:

CAPWAP підтримує шифрування трафіку, щоб забезпечити конфіденційність і цілісність передачі даних. Додаткова безпека забезпечується за допомогою протоколу дейтаграм безпеки транспортного рівня (*DTLS*, *Datagram Transport Layer Security*). Це важливо для захисту мережі від атак і забезпечення безпечного обміну даними.

4. Прозорість для користувачів:

Користувачі мережі не помічають роботи *CAPWAP* тунелю, оскільки він працює на рівні інфраструктури і прозорий для кінцевих користувачів. Вони лише отримують доступ до мережних ресурсів через точки доступу.

5. Гнучкість і масштабованість:

CAPWAP тунелі дозволяють легко масштабувати бездротові мережі, додаючи нові точки доступу, які автоматично підключаються до існуючого контролера. Це спрощує управління великими мережами і дозволяє централізовано керувати всіма точками доступу.

2.7.5 ЧАСТОТНИЙ РЕСУРС КАНАЛІВ

Існує багато завдань, пов'язаних із правильним проектуванням і розгортанням бездротової мережі, і одним з найважливіших є розробка плану каналів. Оскільки пристрої бездротової мережі використовують передавачі і приймачі, налаштовані на певні радіочастоти, частоти поділяються на діапазони, а ті, в свою чергу, на канали. Добре розроблена схема каналів допоможе ефективно використовувати кожен біт дорогоцінного ефірного часу, що є основою для створення високопродуктивних мереж *WI-FI*.

Стандарт *IEEE 802.11* визначає роботу бездротових мереж у діапазонах частот 2,4 ГГц, 5 ГГц, а тепер і 6 ГГц. В Україні діапазон 2,4 ГГц розбитий на 13 каналів (в деяких країнах тільки 11), кожен шириною 20 МГц (рисунк 2.25). У діапазоні 5 ГГц маємо канали від 36 до 165, а в діапазоні 6 ГГц – канали *WI-FI* від 1-233.

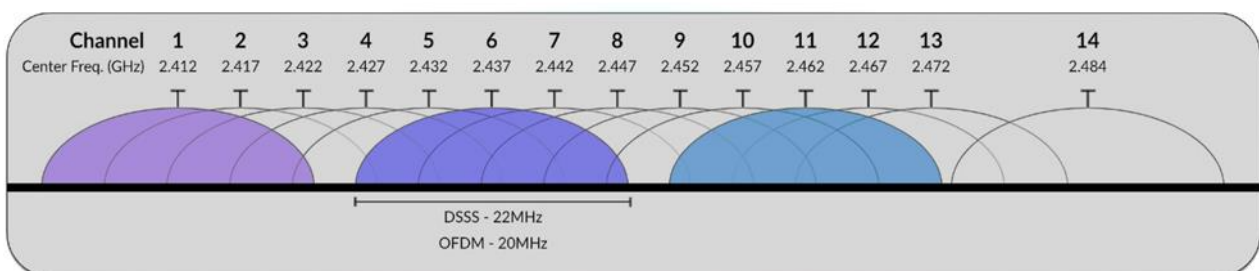


Рисунок 2.25 – Перекриття каналів для діапазону 2,4 ГГц

У діапазоні 2,4 ГГц канали 1, 6 і 11 – єдині, що не перекриваються (рисунок 2.26). Вибір одного або кількох із цих каналів є важливою частиною правильного налаштування мережі.

У діапазоні 5 ГГц доступно значно більше спектру частот, причому кожен канал займає свій власний відрізок у 20 МГц, що не перекриваються.

Стандартні канали 20 МГц можна об'єднувати для збільшення ширини каналу, що підвищує швидкість передачі. Високі показники пропускної здатності, зазначені в специфікаціях точок доступу, досягаються завдяки таким широким каналам. Деяке обладнання вже налаштоване на широкі канали за замовчуванням. Такі канали створюються шляхом об'єднання сусідніх 20 МГц каналів, використовуючи центральну частоту, наприклад, канали 36 і 40 об'єднуються в 40 МГц канал з частотою 38.

Frequency (GHz)	5.150				5.250				5.470				5.600				5.640				5.725				5.850			
													TDWR															
802.11 Allocations	UNII-1				UNII-2a				UNII-2c (Extended)												UNII-3							
Center Frequency	5180	5200	5220	5240	5260	5280	5300	5320	5500	5520	5540	5560	5580	5600	5620	5640	5660	5680	5700	5720	5745	5765	5785	5805	5825			
20 MHz	36	40	44	48	52	56	60	64	100	104	108	112	116	120	124	128	132	136	140	144	149	153	157	161	165			
40 MHz	38		46		54		62		102		110		118		126		134		142		151		159					
80 MHz	42				58				106				122		138				155									
160 MHz	50								114																			
FCC	1,000 mW Tx Power Indoor & Outdoor No DFS needed				250 mw w/6dBi Indoor & Outdoor DFS Required				250mw w/6dBi Indoor & Outdoor DFS Required 144 Now Allowed				120, 124, 128 Devices Now Allowed				1,000 mW EIRP Indoor & Outdoor No DFS needed 165 was ISM, now UNII-3											
DFS Channels													DFS Channels															

Рисунок 2.26 – Розподіл каналів для діапазону 5 ГГц

Точки доступу і клієнтські пристрої з підтримкою 6 ГГц мають різні категорії на основі рівнів їх потужності, а це означає, що вони будуть поводитися по-різному і матимуть доступ до різної кількості каналів 6 ГГц і максимальної еквівалентної ізотропної випромінюваної потужності, яку вони можуть використовувати для роботи.

На даний час Європейський Союз наразі надав додаткові 500 МГц спектру в діапазоні 6 ГГц. Це забезпечує доступ виключно до каналів радіочастотного діапазону UNII-5. Як результат, тут можна використовувати наступні додаткові канали 6 ГГц і конфігурації ширини каналів:

- канали шириною 24 x 20 МГц;
- канали шириною 12 x 40 МГц;
- 6 каналів шириною 80 МГц;
- 3 канали шириною 160 МГц.

В будь-якому випадку, при плануванні і розгортанні бездротових мереж, потрібно завжди враховувати очікувану зону покриття точки доступу, що змінюється залежно від використовуваного стандарту WLAN, налаштованої потужності передавання та умов об'єкта, на якому планується розгортання.

2.8 ПРИНЦИПИ МАРШРУТИЗАЦІЇ

2.8.1 ОГЛЯД ОСНОВНИХ ФУНКЦІЙ МАРШРУТИЗАЦІЇ

Маршрутизатори реалізують функції мережного Рівня 3. Їхнє основне завдання – пересилати пакети на основі таблиці маршрутизації, забезпечуючи сегментацію трафіку, розділення широкомовних доменів і визначають адресацію мереж. Ці мережі визначаються мережними інтерфейсами, яким присвоюються *IP*-адреси, що як правило, є шлюзами за замовчуванням для ПК і серверів або інших мережних пристроїв.

Маршрутизатори підключаються до провайдерів та діють як шлюзи до інших мереж, зазвичай розташовані на межі мережі. Такі підключення можуть відрізнятися від *Ethernet*, охоплюючи послідовні інтерфейси, *DSL* та інші типи *WAN*. Основні компоненти маршрутизатора подібні до будь-якого комп'ютера: процесор, материнська плата, ОЗП, ПЗП. У маршрутизаторів *Cisco* наявні кілька типів пам'яті, включно з флеш-пам'яттю для образу ОС. Деякі маршрутизатори також підтримують функції брандмауера й *IP*-телефонію.

Основну функцію маршрутизації можна розділити на дві частини: одна з них полягає в побудові карти мережі, і для цього маршрутизатори зазвичай використовують статичні маршрути чи динамічні протоколи маршрутизації. За допомогою останніх маршрутизатори повідомляють іншим мережним пристроям не тільки про топологію мережі, але й про зміни в ній. Статична маршрутизація залишається статичною і не адаптується до змін у мережі. Обидві моделі виконують завдання побудови карти мережі у вигляді таблиці маршрутизації.

У процесі визначення шляху маршрутизатори розглядають кілька альтернативних варіантів, щоб дістатися до одного й того ж місця. Ці альтернативи є результатом надмірності, вбудованої в більшість мережних конструкцій. Потрібно мати кілька шляхів, щоб у разі відмови одного з них інші були доступними. Визначаючи найкращий шлях, маршрутизатори враховують декілька факторів. Одним з них є джерело інформації, тому може бути декілька протоколів динамічної маршрутизації або навіть статичної маршрутизації, які заповнюють таблицю маршрутизації і повідомляють про наявні варіанти.

Друга частина інформації – це вартість проходження кожного шляху, знаючи, що він складається з декількох з'єднань або переходів, які визначаються іншими маршрутизаторами. Тоді додається поняття вартості в контексті загального шляху, але ця вартість є нічим іншим, як сумою всіх витрат на досягнення кожного переходу на шляху.

Дві ці процедури визначаються різними наборами інформації. Наприклад, для вирішення конфлікту між джерелами інформації маршрутизатори використовують адміністративну відстань. Якщо протоколи маршрутизації, такі як *OSPF* і *RIP*, надають маршрутизатору інформацію про одне і те ж місце призначення, то адміністративна відстань визначає, який із них буде пріоритетним. Коли

джерело обрано, має значення вартість, іншими словами, якщо *OSPF* надає інформацію про два шляхи, то вибір шляху визначається його вартістю.

2.8.2 ТАБЛИЦЯ МАРШРУТИЗАЦІЇ

Таблиця маршрутизації містить інформацію мережного рівня, що вказує маршрутизатору, як пересилати пакети до віддалених пунктів призначення. Спочатку вона складається з мереж, безпосередньо підключених до маршрутизатора. Для отримання даних про віддалені пункти таблиця маршрутизації може заповнюватися або статичними маршрутами, які налаштовує адміністратор, або маршрутами, отриманими з оголошень від інших маршрутизаторів.

В обох випадках – статичної та динамічної маршрутизації – зверніть увагу, як маршрутизатори використовують зарезервовані адреси підмереж або мережні адреси, що містять усі 0 в частині *IP*-адреси, відведений для вузлів (рисунк 2.27).

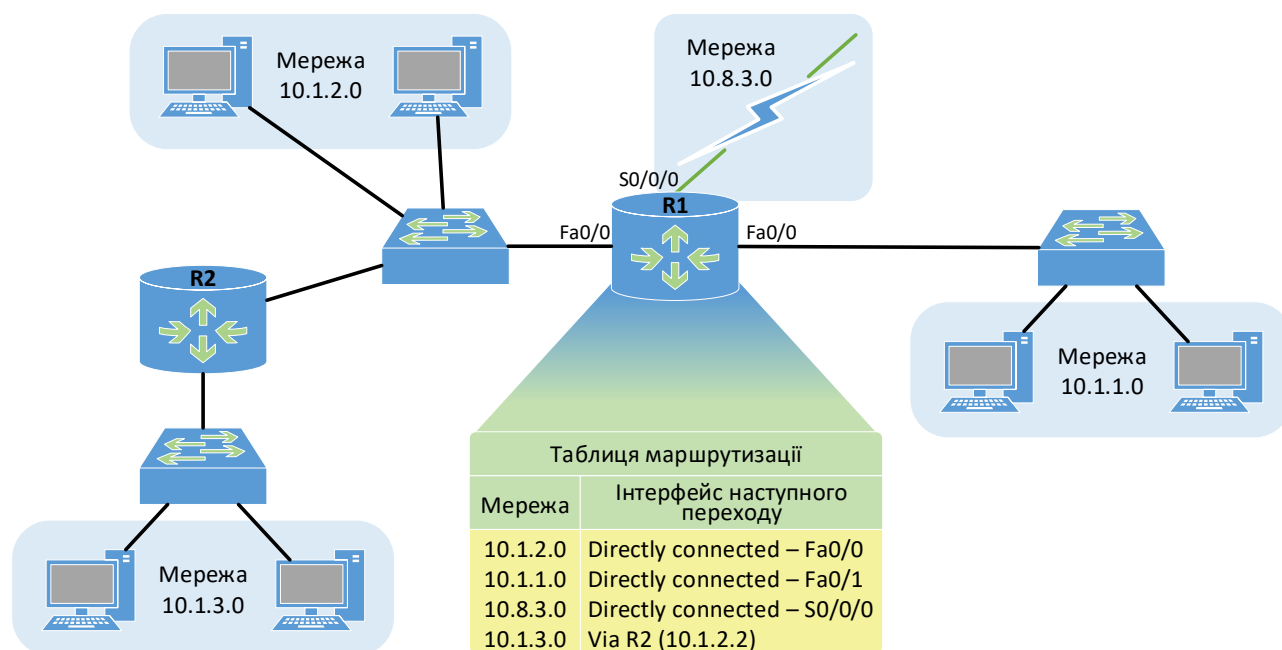


Рисунок 2.27 – Записи в таблиці маршрутизації

У цьому випадку йдеться про безкласову підмережну адресу класу А. Мережа 10 розбита на підмережі, подібно до класу С. Однак у всіх випадках тут вузлова частина адреси, четвертий байт, містить всі 0 і представляє цю підмережу або саму мережу. Іншими словами, це мережі або підмережі призначення. У випадку віддалених пунктів призначення, записи таблиці маршрутизації показують, який наступний крок потрібно зробити, щоб досягти цього пункту призначення. У цьому випадку, щоб досягти 10.1.3.0, наступний крок – це маршрутизатор R2 за адресою 10.1.2.2.

Для цього маршрутизатор перевіряє *IP*-префікс призначення, шукає відповідний запис у таблиці маршрутизації та пересилає пакет на наступний вузол згідно з цим записом. Оскільки використовується безкласова адресація, префікси

можуть перекриватися, і IP-префікс вхідного пакета може відповідати кільком записам у таблиці. Наприклад, розглянемо таблицю маршрутизації 2.7.

Таблиця 2.7 – Таблиця маршрутизації

Мережа	Наступний перехід
10.1.0.0/18	D
10.1.12.0/22	B

У наведеній таблиці адреси від 10.1.12.0 до 10.1.15.255 перекриваються, тобто відповідають одразу двом записам у таблиці. Для вирішення такої ситуації маршрутизатори використовують правило найдовшого збігу префіксу (*Longest Prefix Matching*). Це правило полягає в тому, щоб знайти запис у таблиці з найдовшим префіксом, який збігається з IP-адресою призначення вхідного пакета, та передати пакет на відповідний наступний вузол.

У наведеному прикладі всі пакети з діапазону, що перекривається (10.1.12.0 – 10.1.15.255), перенаправляються на вузол B, оскільки він має довший префікс (22 біти).

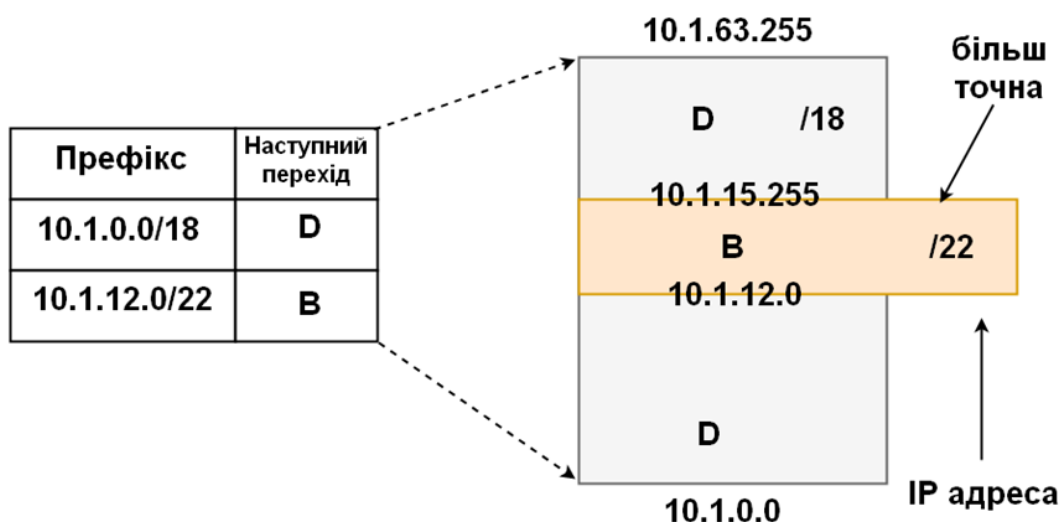


Рисунок 2.28 – Записи в таблиці маршрутизації, що перекриваються

Протоколи маршрутизації можуть виявляти зміни топології та повідомляти про них один одного. Записи в таблиці маршрутизації будуть додаватися та видалятися залежно від доступності. Адміністратор також може вручну додавати статичні маршрути, але це не завжди рекомендується, оскільки вони не адаптуються до змін у мережі. Якщо пункт призначення стане недоступним, запис залишиться, і маршрутизатор продовжуватиме пересилати пакети в недосяжне місце. Особливим випадком статичного маршруту є маршрут за замовчуванням. Він використовується, коли немає явного маршруту до призначення, і визначає шлях для всіх невідомих пунктів призначення.

2.8.3 ПОКАЗНИКИ МАРШРУТИЗАЦІЇ

Оптимальний вибір шляху залежить від вартості досягнення пункту призначення. Вартість, або метрика, складається з витрат на кожен перехід на шляху. Різні протоколи маршрутизації використовують різні критерії для обчислення метрики. Старіші протоколи враховують кількість переходів, але цей підхід не завжди ефективний, оскільки пропускна здатність з'єднань може відрізнятися на різних переходах (рисунок 2.29).

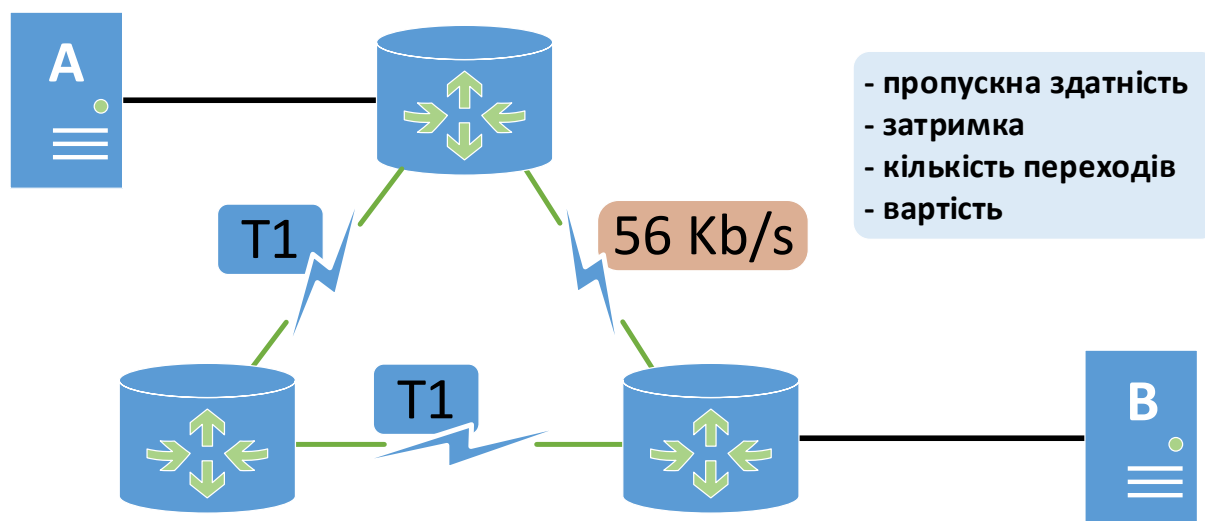


Рисунок 2.29 – Критерії обчислення метрики

У цьому прикладі шлях з двома переходами є кращим через більшу пропускну здатність. Деякі протоколи, такі як *EIGRP*, враховують додаткові параметри: затримку, надійність, навантаження та максимальну одиницю передачі (*MTU*). Може бути обраний навіть шлях з меншою пропускну здатністю, якщо він менше перевантажений і надійніший.

Вибір протоколу маршрутизації є ключовим для визначення метрики та ефективності вибору шляху. Також важливим є час збіжності – час, потрібний протоколу для виявлення змін топології й вибору альтернативного маршруту. Протоколи маршрутизації поділяються на різні категорії, що впливають на їхню метрику, поведінку та реакцію на зміни в мережі.

2.8.4 ПРОТОКОЛИ ДИНАМІЧНОЇ МАРШРУТИЗАЦІЇ

Вибір протоколу маршрутизації є ключовим у визначенні вартості або метрики і, отже, наскільки ефективним і оптимальним буде вибір шляху, а також часу збіжності, який визначається часом, необхідним протоколу маршрутизації для виявлення зміни топології і пристосування, вибравши альтернативний шлях, якщо основний шлях не працює. Існують різні категорії протоколів динамічної маршрутизації, які визначають їх вартість і метрику, а також їхню поведінку за цих обставин.

Використовуючи дистанційно-векторний підхід, який є однією з категорій, маршрутизаторам не обов'язково знати весь шлях до місця призначення. Потрібно знати лише напрямок або вектор, в якому потрібно відправити пакет. У цьому сенсі в таблицях маршрутизації зберігається лише інформація, пов'язана з тим, яким має бути наступний пункт призначення, щоб досягти певного місця призначення.

Дистанційно-векторні протоколи мають багато недоліків, один із яких полягає в тому, що вони періодично анонсують таблиці маршрутизації. Деякі з них використовують широкомовні повідомлення для анонсування всієї таблиці. Це створює занадто багато накладних витрат у мережі і може бути непотрібним, якщо мережа насправді не змінюється.

Протоколи стану каналу ефективніші за дистанційно-векторні у створенні топологій, спільному використанні даних і виборі найкращого шляху. Вони відрізняються тим, що використовують багатоадресну розсилку для обміну інформацією, а не передають її кожному маршрутизатору окремо. Після початкового обміну даними протоколи стану каналу повідомляють лише про зміни в топології. Наприклад, якщо пропадає з'єднання, про це сповіщається через багатоадресну розсилку. Також маршрутизатори отримують повну карту мережі, що дозволяє їм обчислювати найкращі шляхи за допомогою алгоритму найкоротшого шляху. Зміни, як-от вихід з ладу каналу, впливають лише на відповідну частину топології, і тільки ці зміни анонсуються в мережі.

2.8.5 СТАТИЧНА МАРШРУТИЗАЦІЯ

Статичні маршрути не створюють додаткового навантаження у вигляді оголошень протоколів маршрутизації чи складної логіки на маршрутизаторах. Вони досить прості у налаштуванні й залишаються гнучкими, якщо їх кількість не перевищує певний ліміт. Однак статична маршрутизація не адаптується до змін у мережі.

Динамічні протоколи маршрутизації створюють додаткове навантаження через оголошення та вивчення мереж. Проте вони дозволяють швидко реагувати на зміни топології. Швидкість адаптації та вибору нового маршруту залежить від використаного протоколу маршрутизації.

Саме для таких сценаріїв з однією лінією зв'язку з рештою мережі, як показано на рисунку, статична маршрутизація є правильним рішенням.

Для вихідного трафіку з крайньої мережі зазвичай використовується маршрут за замовчуванням, який направляє весь трафік до всіх отримувачів за межами цієї мережі. Інтернет є ідеальним прикладом – немає потреби знати кожне окреме призначення, оскільки весь трафік можна передавати через одне з'єднання.

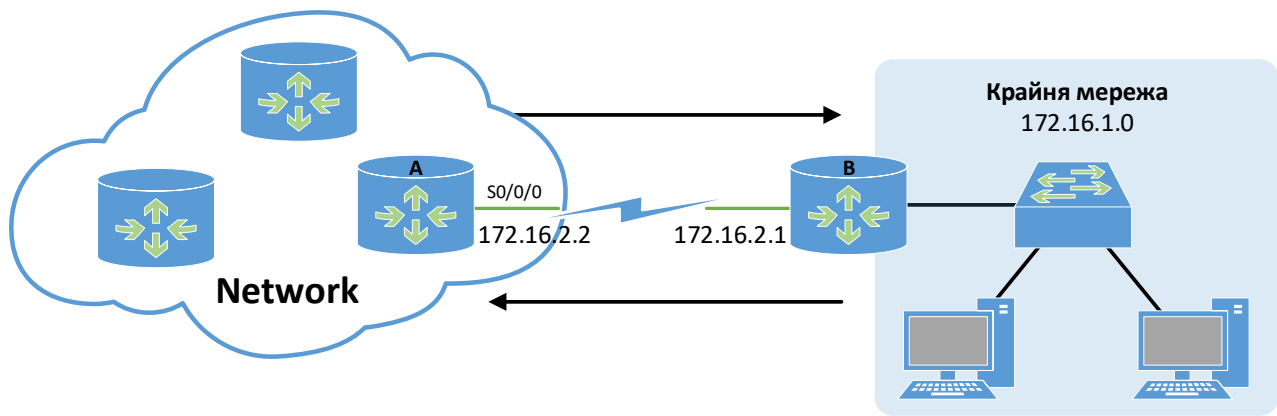


Рисунок 2.30 – Крайня мережа з однією лінією зв'язку

Маршрут за замовчуванням, який спрямовує весь невідомий трафік через маршрутизатор А, буде достатнім. Однак слід враховувати двосторонній характер трафіку – якщо пакети виходять, то повинні мати можливість повернутися. Тому необхідно налаштувати статичні маршрути на маршрутизаторі А, що вказують на мережі крайнього сегмента.

У деяких випадках можливе використання гібридної маршрутизації. Наприклад, на маршрутизаторі В можна налаштувати маршрут за замовчуванням для вихідного трафіку, а для вхідного – налаштувати В на оголошення маршрутів за допомогою динамічного протоколу, щоб маршрутизатор А знав про них і міг пересилати трафік.

Для налаштування статичної маршрутизації в маршрутизаторах *Cisco IOS* слід використовувати цю команду в режимі глобальної конфігурації:

```
ip route [network/host] [mask] [address/interface]
          [distance] [permanent]
```

Команда **ip route** містить мережу призначення та маску, дозволяючи додавати записи *CIDR* або безкласової маршрутизації для підмереж з різними масками. Кінцевий запис – це маршрут /32, де пунктом призначення є конкретний вузол. Наступний вузол може бути *IP*-адресою маршрутизатора або локальним інтерфейсом, який використовується для доступу до мережі призначення. Крайній варіант використовується для інтерфейсів типу «точка-точка», оскільки на інтерфейсах з множинним доступом, як-от *Ethernet*, це не працює. Статичні маршрути мають адміністративну відстань 1 за замовчуванням, але її можна змінити для створення плаваючих маршрутів для резервування.

Адміністративна відстань визначає пріоритет протоколу: маршрут із меншою відстанню витісняє інші. Наприклад, якщо для одного й того ж пункту призначення існує статичний і динамічний маршрут, статичний матиме пріоритет через меншу адміністративну відстань.

Приклад статичного маршруту, налаштованого на маршрутизаторі А наведено на рисунку 2.31. Він вказує на крайню мережу, і тут є два варіанти: вказати IP-адресу маршрутизатора В на послідовному інтерфейсі або просто вказати локальний послідовний інтерфейс маршрутизатора А.

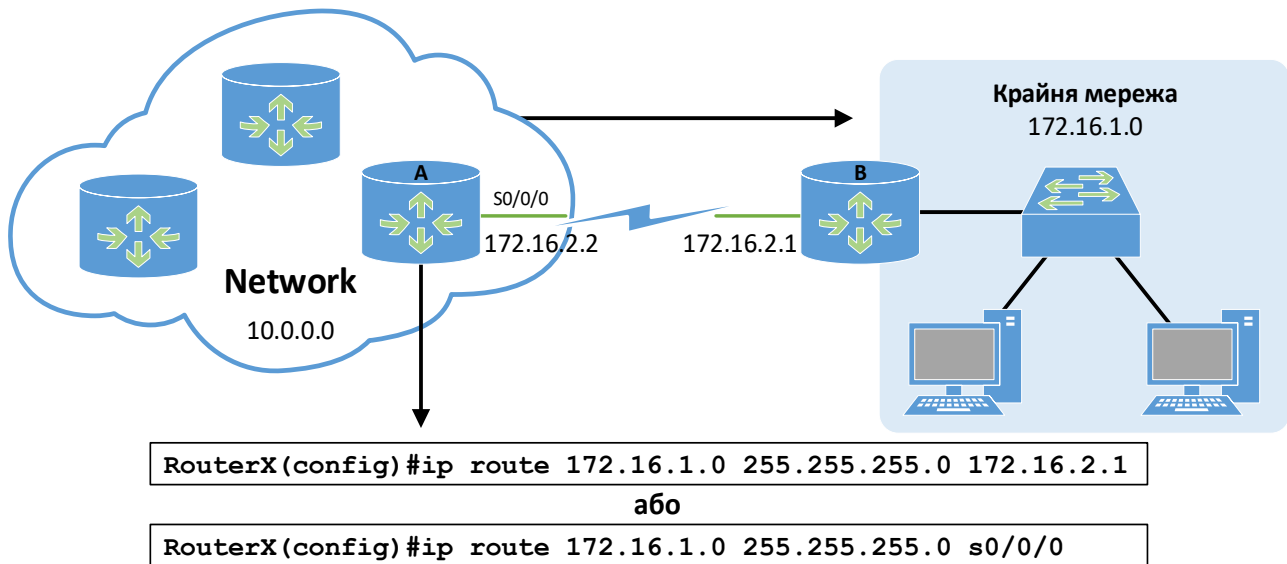


Рисунок 2.31 – Приклад статичного маршруту

Параметр інтерфейсу зазвичай використовується у плаваючих статичних маршрутах і резервних копіях, а IP-адреса зазвичай використовується у сценарії з'єднання «точка-точка». Потрібно зауважити, що це лише односпрямований маршрут; тому, щоб трафік виходив з крайньої мережі, потрібно визначити статичну або динамічну маршрутизацію також на маршрутизаторі В. Це робиться за допомогою маршруту за замовчуванням (рисунок 2.32). Цього разу статичний маршрут на маршрутизаторі В вказує на всі невідомі пункти призначення. На це вказують всі нулі в адресі призначення і масці.

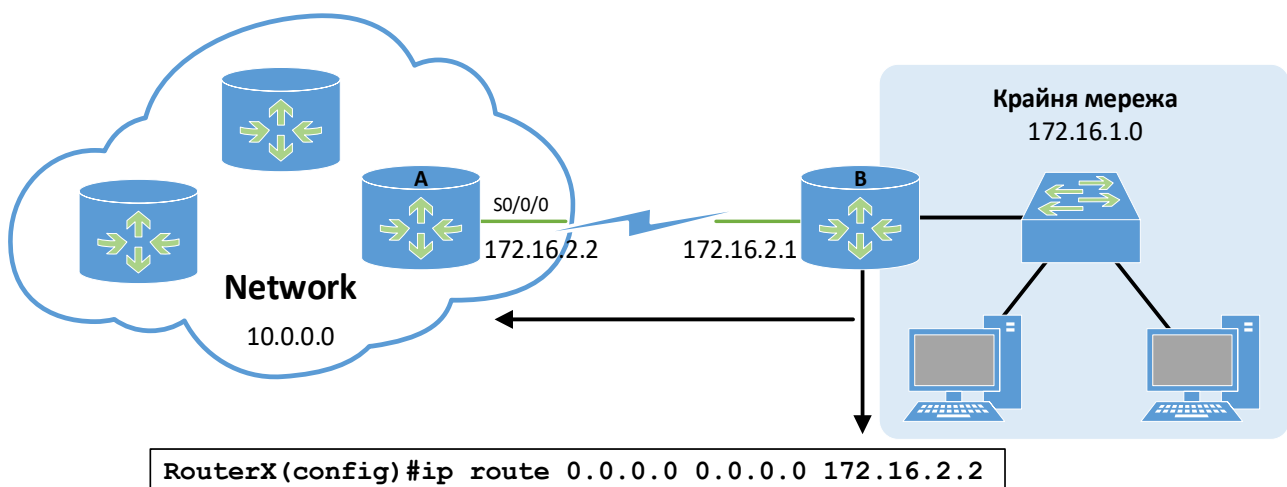


Рисунок 2.32 – Приклад маршруту за замовчуванням

Для перевірки маршрутів зазвичай використовуються команди відображення таблиці маршрутизації. Далі на рисунку 2.33 показано приклад роботи команди **show ip route** для виведення таблиці маршрутизації на маршрутизаторі B:

```
Router B#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is 172.16.2.2 to network 0.0.0.0
C    172.16.2.0/24 is directly connected, FastEthernet0/0
C    172.16.1.0/24 is directly connected, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 172.16.2.2
```

Рисунок 2.33 – Виведення команди **show ip route**

Зверніть увагу на три індикатори маршрутів за замовчуванням: перший – це всі нулі пункту призначення, він сигналізує про маршрут за замовчуванням; по-друге, зірочка (*) – це кандидат на маршрут за замовчуванням, їх може бути більше одного на певному маршрутизаторі; і, нарешті, вказівка вгорі, яка говорить, що шлюз останньої інстанції – 0.0.0.0. Цей запис є не просто маршрутом за замовчуванням, а статичним маршрутом за замовчуванням, на що вказує літера «S» у першому стовпчику.

РОЗДІЛ 3

ЛАБОРАТОРНЕ ОБЛАДНАННЯ

3.1 КОМУТАТОРИ CISCO CATALYST

Компанія *Cisco* є одним із лідерів на ринку мережевого обладнання для різних сегментів – від користувацького доступу для мереж малих офісів до потужних мереж провайдерів послуг та центрів обробки даних. В комутаторах *Cisco* крім базового функціоналу для підтримки технології *Ethernet* було реалізовано велику кількість нових функцій, підтримку багатьох мережевих протоколів, спеціалізованих технологій, архітектурних та технологічних рішень.

Узагальнену структурну схему комутатора *Ethernet* компанії *Cisco* показано на рисунку 3.1.

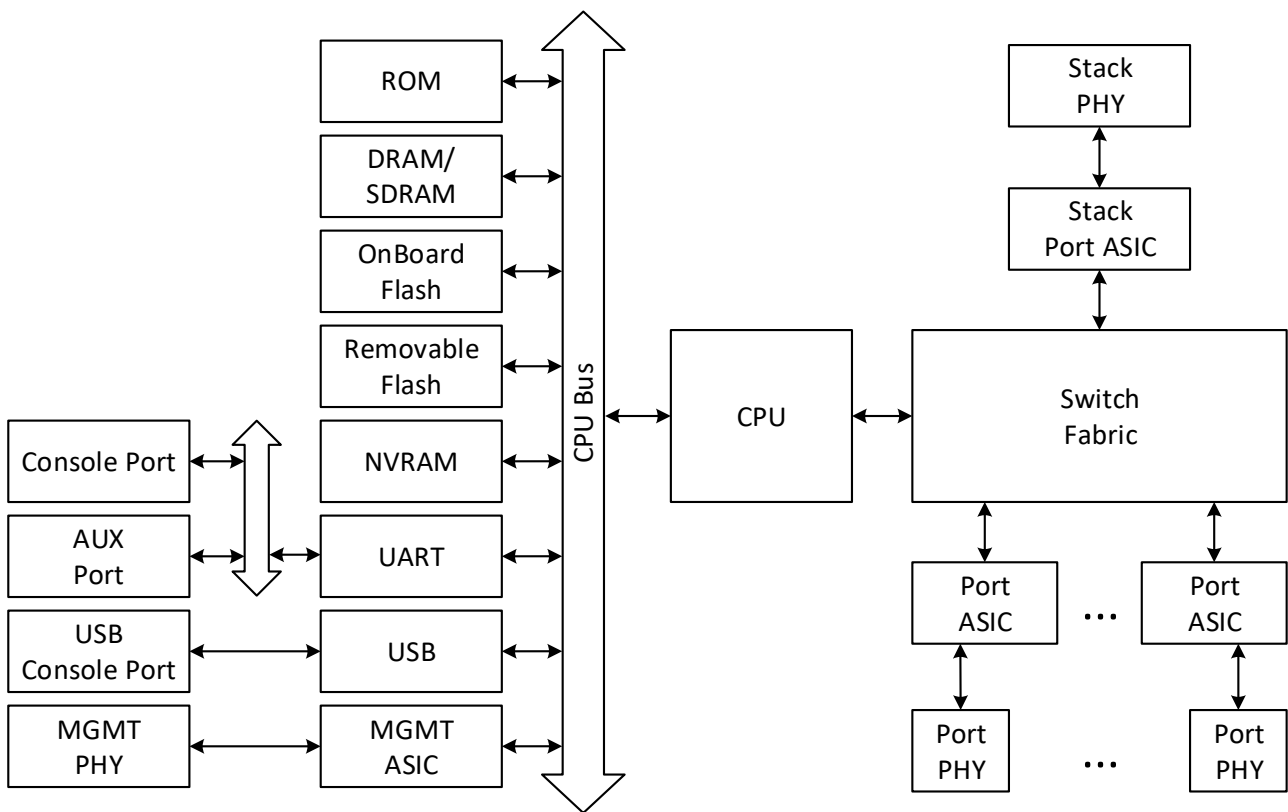


Рисунок 3.1 – Типова структурна схема комутатора фірми *Cisco*

Комутатор фірми *Cisco* містить в своєму складі такі основні блоки:

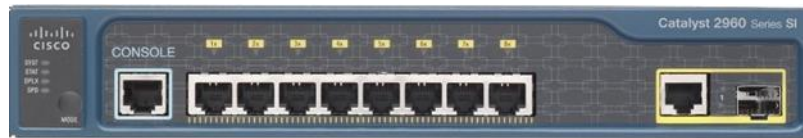
- **CPU** – центральний процесор, що забезпечує керування комутатором та координацію роботи його складових, а також забезпечує актуальність основної таблиці комутації.
- **Switch Fabric** – блок комутації, виконує функції по забезпеченню передачі трафіку між портами, наданні якості обслуговування (QoS),

відмовостійкості комутатора. Він відповідає за пропускну здатність комутатора, що вимірюється у Гбіт/с.

- **ROM** – блок постійної пам'яті зберігає процедуру власного тестування при ввімкненні живлення *POST (Power-On Self-Test)* та програму початкового завантаження *Boot Loader*.
- **RAM** – блок оперативної пам'яті виконує функції, що аналогічні такому ж блоку персонального комп'ютера. В оперативній пам'яті після завантаження розміщується *Cisco IOS*, ця пам'ять також використовується для забезпечення процесу передачі кадрів між портами.
- **OnBoard Flash** – блок постійної перезаписуваної пам'яті виконує функції накопичувача пристрою. Він містить файл образу *Cisco IOS* та деякі конфігураційні файли, які створюються у процесі налагодження та використовуються у процесі роботи комутатора. Файлів образів у перезаписуваній пам'яті може міститися кілька. Можливості постійної перезаписуваної пам'яті можуть бути розширені за рахунок застосування блока змінної перезаписуваної пам'яті (*Removable Flash*).
- **NVRAM** – блок енергонезалежної пам'яті застосовується для збереження конфігурації комутатора.
- **Port ASIC** – блоки керування інтерфейсами/портами *Ethernet*. Один такий блок може забезпечувати функціонування кількох портів *Ethernet* або кількох стекових портів.
- **Port PHY** – блоки фізичного рівня, трансивери *Ethernet*.
- **Stack Port PHY** – блоки фізичного рівня для формування стеку.
- **MGMT PHY** – блоки керування інтерфейсів/портів для підключення робочих станцій керування, що підтримують застосування різних фізичних інтерфейсів: послідовного інтерфейсу, інтерфейсу *USB*, інтерфейсу *Ethernet*. Для забезпечення функціонування послідовних інтерфейсів (основного – *Console Port*, допоміжного – *AUX Port*) застосовується блок, що базується на мікросхемі *UART*. Для забезпечення функціонування інтерфейсу *USB (USB Console Port)* – відповідний блок *USB*. Для підключення робочих станцій керування із застосуванням технологій *Ethernet* через інтерфейс керування *MGMT* – блок, який аналогічний за функціями блока керування портом *Ethernet*. У деяких моделях комутаторів блок інтерфейсу *USB* також підтримує можливість підключення зовнішніх змінних *USB*-носіїв або пристроїв.

Типовий комутатор *Cisco* має фіксований набір з 8, 16, 24, 48 інтерфейсів/портів *Ethernet* 10/100/1000 Мбіт/с для підключення кінцевих вузлів за допомогою витої пари. Іноді такі порти називають лінійними портами. Зовнішній вигляд передньої панелі комутатора *Catalyst 2960*, який має лише 8 лінійних портів наведений на рисунку 3.2 а), а на рисунку 3.2 б) – 24-портовий комутатор *Catalyst 2960*. У більшості моделей комутаторів *Cisco* наявні додаткові один, два або

чотири порти 1 Гбіт/с або 10 Гбіт/с чи вище. Ці високошвидкісні порти призначені для підключення серверів, з'єднання комутаторів між собою, підключення до маршрутизатора, агрегації каналів тощо. можуть бути фіксованими (вита пара) та змінними з роз'ємами для модулів *SFP*, що підтримують різноманітні інтерфейси (оптичні тощо). Детальніша інформація стосовно цих портів міститься у технічній документації.



а)



б)

Рисунок 3.2 – Зовнішній вигляд комутаторів *Cisco C2960* спереду

На передній панелі комутатора *Cisco* розміщуються (рисунок 3.3) мережеві інтерфейси/порти, кнопка переключення режимів світлодіодних індикаторів (*Mode*), світлодіодні індикатори (*LED*), що призначені для відображення стану комутатора (таблиці 3.1). Кожен порт *Ethernet* також має власний індикатор, який відображає його стан.

Загальні правила розуміння світіння індикаторів є такими. Якщо індикатор не світиться (*Off*) – це свідчить про відключення або непрацездатність пристрою в цілому, певного його блока, підсистеми або каналу зв'язку. Якщо індикатор світиться зеленим кольором (*Green*) або мерехтить зеленим кольором (*Blinking Green*) – це свідчить про нормальний режим роботи, якщо ж індикатор світиться жовтим кольором (*Amber*) або мерехтить жовтим кольором (*Blinking Amber*) – це свідчить про те, що виникла певна проблема.

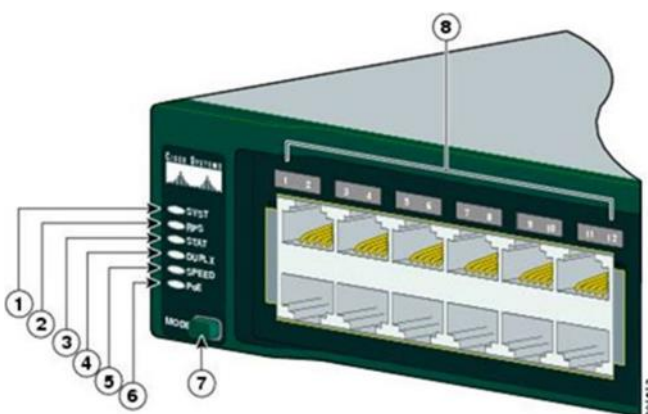


Рисунок 3.3 – Панель індикації комутаторів *Cisco*

Таблиця 3.1 – Світлодіодні індикатори комутатора *Cisco*

№	Позначення	Функціональне призначення
1.	SYST	Індикатор загального стану системи
2.	RPS	Індикатор стану системи резервного живлення
3.	STAT	Індикатор стану портів
4.	DUPLX	Індикатор напівдуплексної/дуплексної передачі портів
5.	SPEED	Індикатор швидкості передачі портів
6.	PoE	Індикатор живлення підключених вузлів з <i>PoE</i>
7.	Кнопка MODE	Кнопка для перемикання відображення режимів роботи

3.1.1 ІНТЕРФЕЙСИ/ПОРТИ КОМУТАТОРІВ

Комутатори *Cisco* забезпечують можливість під'єднання до мережі кінцевих та проміжних вузлів із використанням різних середовищ передачі даних та різних швидкостей. Найбільш поширеними на сьогодні є з'єднання на основі витної пари та скло-волоконного кабелю. Під'єднання на основі витної пари зазвичай забезпечують швидкості від 100 Мбіт/с до 1 Гбіт/с. Скловолокно має більшу пропускну здатність і використовується для зв'язку на швидкостях більше 1 Гбіт/с.

Сучасні комутатори від компанії *Cisco* фактично не мають фіксованих оптичних інтерфейсів. Замість них застосовуються інтерфейсні слоти для змінних мережних інтерфейсних модулів. Це забезпечує можливість гнучко змінювати конфігурацію комутатора з метою під'єднання сегментів мереж *Ethernet* різних середовищ і для різних швидкостей передачі.

3.1.2 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМУТАТОРІВ

Програмне забезпечення комутатора може бути реалізоване як у вигляді мікропрограми для керуючого контролера, так і у вигляді повноцінної мережної ОС. Мікропрограмний код (*Firmware*) – це системне ПЗ, яке є вбудованим у пристрій і зберігається в його енергонезалежній пам'яті. Основною відмінністю ОС від *Firmware* є наявність файлової системи, яка забезпечує можливість виконання різних операцій з файлами. Як правило, і мікропрограма, і ОС комутатора є монолітними, тобто поставляються у вигляді одного бінарного файлу, зміни в який вносити неможливо. Цей файл-образ зазвичай зберігається у флеш-пам'яті пристрою. Заміна *Firmware* або ОС здійснюється шляхом заміни файла-образу. Для комутаторів, які використовують *Firmware*, налаштування пристрою зберігаються у вигляді бітової послідовності у спеціально відведений області енергонезалежної пам'яті. У комутаторах, які використовують мережеву ОС, як правило, збереження конфігурації здійснюється у вигляді структурованого текстового файлу.

Оскільки комутатори є досить спеціалізованими мережними пристроями, то під час розробки їх програмної частини застосовується модельно-орієнтований підхід – *Firmware* та ОС розробляються з урахуванням апаратної складової та призначення пристроїв.

Для забезпечення функціонування мережних пристроїв фірмою *Cisco* розробляються як різні варіанти *Firmware*, так і спеціалізовані мережеві ОС. У сучасних комутаторах та маршрутизаторах *Cisco* найчастіше використовується спеціалізована мережева ОС *Cisco IOS*. *Cisco IOS* поставляється у вигляді монолітного образу, який орієнтований на конкретну модель пристрою. Образи можуть мати певні набори властивостей та версії.

3.2 МАРШРУТИЗАТОРИ CISCO ISR

Маршрутизатори з інтегрованими послугами (*ISR*) *Cisco* серії 1800 (модульні) – це модульні маршрутизатори з *LAN* і *WAN*-з'єднаннями, які можна конфігурувати за допомогою змінних інтерфейсних плат і розширених інтеграційних модулів (*AIM*). Модульна конструкція маршрутизаторів забезпечує гнучкість, дозволяючи конфігурувати або реконфігурувати маршрутизатор відповідно до ваших потреб.

Архітектура побудови маршрутизаторів *Cisco* залежить від призначення та продуктивності. Типову структурну схему маршрутизатора *Cisco* можна подати у вигляді, показаному на рисунку 3.4.

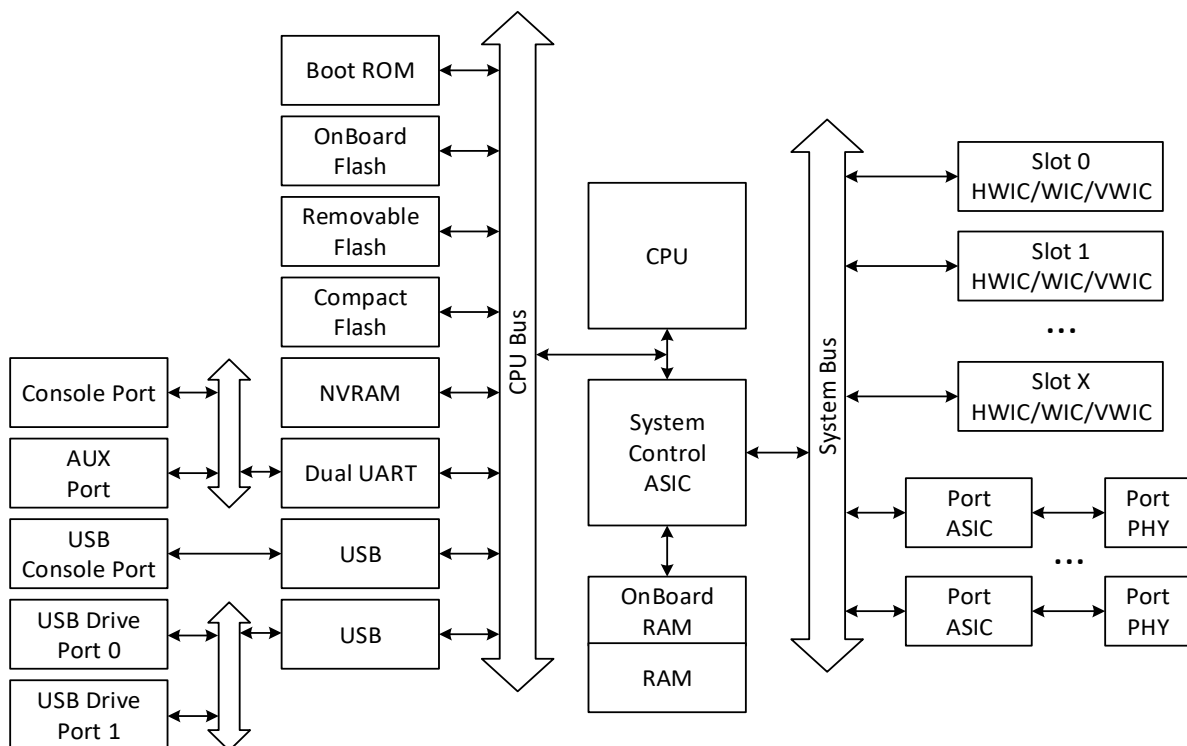


Рисунок 3.4 – Типова структурна схема маршрутизатора Cisco

Як видно з рисунка 3.4, маршрутизатор має типову структуру комп'ютерної системи. Призначення основних компонентів маршрутизатора таке:

- **CPU** – центральний процесор, який забезпечує пересилання пакетів між інтерфейсами, підтримку функціонування протоколів маршрутизації, керування та обслуговування маршрутизатора.
- **System Control ASIC** – блок керування системою координує роботу складових маршрутизатора та забезпечує між ними зв'язок.
- **ROM** – постійний запам'ятовуючий пристрій, використовується маршрутизатором для зберігання початкового завантаження *Bootstrap*, програмного забезпечення операційної системи, програми само-тестування після ввімкнення живлення (*POST*), утиліти *ROMMON* та *mini-IOS*.
- **OnBoard Flash** – блок постійної перезаписуваної пам'яті виконує функції накопичувача маршрутизатора. Він містить файл(и) образу *Cisco IOS* та деякі конфігураційні файли, які створюються у процесі налагодження та роботи пристрою. Ємність пам'яті може бути збільшена за рахунок застосування змінної перезаписуваної пам'яті (*Removable Flash*).
- **NVRAM** – енергонезалежна пам'ять, в якій зберігається файл початкової конфігурації (**startup configuration**).
- **RAM** – оперативна пам'ять виконує функції, що аналогічні оперативній пам'яті звичайного комп'ютера. Оперативна пам'ять маршрутизатора логічно поділяється на дві частини: основну процесорну пам'ять та пам'ять вводу-виводу. В основну пам'ять завантажується *Cisco IOS*, у ній також розміщуються поточна конфігурація пристрою та різні таблиці (таблиці маршрутизації, *ARP*-таблиці, *CDP*-таблиці тощо). У пам'яті вводу-виводу містяться вхідні та вихідні буфери інтерфейсів, у яких розміщуються пакети, що маршрутизуються.
- **UART** – блок керування послідовним інтерфейсом *UART*.
- **USB** – блок керування інтерфейсами *USB*.
- **Port ASIC** – блоки керування вбудованими інтерфейсами/портами *Ethernet/Fast Ethernet/Gigabit Ethernet, Wi-Fi* тощо.
- **PHY** – блоки фізичного рівня, трансивери *Ethernet/Fast Ethernet/Gigabit Ethernet, Wi-Fi* тощо.
- **Slot X** – блоки підключення змінних плат/модулів розширення для відповідних технологій зв'язку.

На рисунку 3.5, як приклад, показано зовнішній вигляд маршрутизатора з інтегрованими службами (*ISR – Integrated Services Router*) *Cisco 1841*. На передній панелі маршрутизатор має 2 світлодіодних індикатори:

1. Індикатор живлення системи (**SYS PWR**) – не світиться – означає, що живлення вимкнено, а блимання зеленим означає, що маршрутизатор завантажується.

- Індикатор системної активності (**SYS ACT**) – вимкнений означає, що через маршрутизатор не проходить трафік, а миготіння зеленим кольором означає, що маршрутизатор в даний момент передає трафік через один з портів.

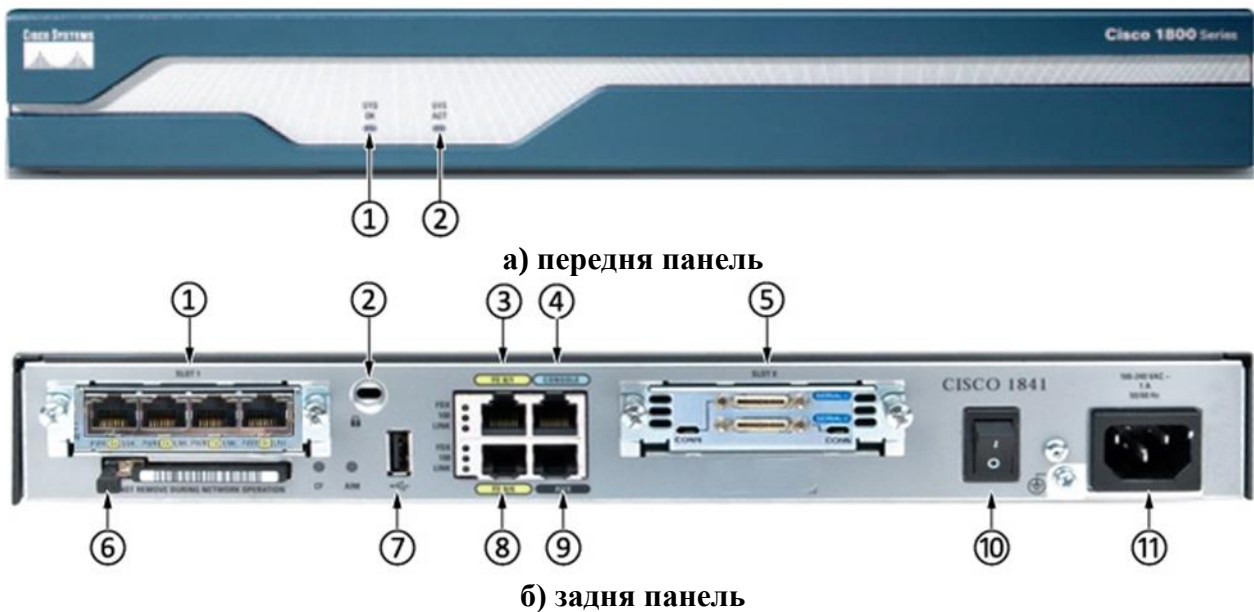


Рисунок 3.5 – Зовнішній вигляд маршрутизатора

На задній панелі розташовані:

- Модульний слот 1 (WIC, VWIC-дані або HWIC)** – слот розширення мережної карти. В даному випадку встановлено чотири-портовий *Ethernet*-комутатор.
- Слот Kensington Security Slot** – фізичний захист маршрутизатора від крадіжки за допомогою замка.
- Вбудовані порти Fast Ethernet** – здатні передавати дані зі швидкістю 100 Мбіт/с. Є три світлодіоди: *FDX* (коли він увімкнений, це означає повний дуплекс, а коли вимкнений - напівдуплекс), *100* (коли він увімкнений, він показує 100 Мбіт/с, а коли вимкнений - 10 Мбіт/с) і *Link* (коли він увімкнений, це означає, що маршрутизатор підключений до локальної мережі).
- Консольний порт** – (порт керування) використовують для налаштування з безпосередньо під'єднаного комп'ютера за допомогою консольного кабелю.
- Модульний слот 0 (WIC, VWIC-дані або HWIC)** – слот розширення мережної карти зі встановленою платою високошвидкісного інтерфейсу глобальної мережі (*HWIC – High-speed WAN Interface Card*), що містить два послідовних порти *Serial 1/0* та *Serial 0/0*.
- Гніздо для карти пам'яті CompactFlash** – для встановлення карти пам'яті. Є індикатор (*CF*), який блимає, якщо карта вставлена і використовується.

- Праворуч від нього знаходиться індикатор (AM), він світиться, якщо в маршрутизаторі встановлений модуль розширеної інтеграції.
7. **Порт USB** – для підключення USB-накопичувача.
 8. Вбудовані порти та світлодіоди *Fast Ethernet*.
 9. **Порт AUX** – (порт керування) використовується як резервний порт консолі.
 10. Перемикач увімкнення/вимкнення – увімкнення та вимкнення маршрутизатора.
 11. Під'єднання вхідного живлення – подача живлення змінного струму на маршрутизатор.

Маршрутизатори *Cisco* працюють під управлінням міжмережної операційної системи (*IOS – Internetworking Operating Software*). Операційна система надається і зберігається у вигляді образу *Cisco IOS* з розширенням *.bin*. До одних і тих же пристроїв може бути розроблено багато різних типів і версій образів *Cisco IOS*.

3.2.1 ІНТЕРФЕЙСИ МАРШРУТИЗАТОРІВ CISCO

Маршрутизатори *Cisco* мають можливість організації з'єднань з використанням різноманітних мережних стандартів, технологій та протоколів. Найпоширенішим є застосування з'єднань на базі технологій стандарту *Ethernet* (*ISO/IEC/IEEE 8802-3:2021*), послідовних (*Serial*) каналів зв'язку (*ITU-T V.35, EIA/TIA-530*), каналів оптоволоконних технологій *PDH/SDH/SONET* тощо.

Для під'єднання на основі витой пари забезпечуються швидкості 10 Мбіт/с, 100 Мбіт/с та 1 Гбіт/с, а на основі оптоволоконного кабелю забезпечуються швидкості 1 Гбіт/с, 10 Гбіт/с, 40 Гбіт/с та 100 Гбіт/с. Для під'єднання пристроїв за допомогою витой пари у кожному маршрутизаторі *Cisco* наявні вбудовані інтерфейси *RJ-45* технологій *100Base-TX/1000-Base-T*. У більшості сучасних маршрутизаторів *Cisco* застосовуються інтерфейсні слоти для змінних мережних інтерфейсних модулів (трансиверів), які дають змогу здійснювати підключення пристроїв різних технологій *Ethernet*. Як правило, у сучасних маршрутизаторах застосовуються модулі *SPF, SFP+, XFP, CFP, QSFP, QSFP+*.

Окрім інтерфейсів стандарту *Ethernet* у маршрутизаторах *Cisco* досить часто реалізуються послідовні (*Serial*) інтерфейси, які можуть підтримувати різні стандарти передачі даних та різні мережні технології. Послідовні інтерфейси забезпечують підключення маршрутизатора до мережевого пристрою провайдера послуг.

Перелік стандартів та роз'ємів, що застосовуються для побудови послідовних каналів зв'язку наведено у таблиці 3.2.

Таблиця 3.2 – Стандарти та роз'єми послідовних інтерфейсів

Назва роз'єму	Стандарт	Зображення роз'єму
34-pin Rectangular Connector	V.35 (ITU-T V.35)	
15-pin D-Connector	X.21bis	
25-pin D-connector	EIA/TIA-232 (RS-232, ITUT V.24)	
37-pin D-Connector	EIA/TIA-449 (RS-449)	
25-pin D-Connector	EIA/TIA-530, EIA/TIA-530A, (RS-422&RS-423)	
50-pin D-Connector	EIA/TIA-612/613 (HSSI, High-Speed Serial Interface)	
26-pin Cisco Smart serial	Cisco Smart serial	

Маршрутизатори *Cisco* можуть бути реалізовані монолітно або у модульному виконанні. Для розширення функціоналу останніх є можливість встановлення модулів та плат розширення, що які забезпечують певну кількість інтерфейсів тієї чи іншої мережної технології. Позначення модулів і плат розширення показано у таблиці 3.3.

Таблиця 3.3 – Позначення модулів і плат розширення для маршрутизаторів *Cisco*

Позначення	Розшифровка
<i>Інтерфейсні модулі</i>	
NM, NME	Network Module, single-wide Network Module
NME-X	eXtended single-wide Network Module
NMD	Double-wide Network Module
NME-XD	eXtended Double-wide Network Module
SM	Service Module
SPE	Services Performance Engine
PVDM	Packet Voice Data Module
ISM	Internal Services Module
AIM	Advanced Integration Module
<i>Інтерфейсні плати</i>	
WIC	WAN Interface Card
VIC	Voice Interface Card
HWIC	High-Speed WIC
VWIC	Voice WIC
EHWIC	Enhanced High-Speed WIC
DW-HWIC	Double-Wide HWIC
DW-EHWIC	Double-Wide EHWIC

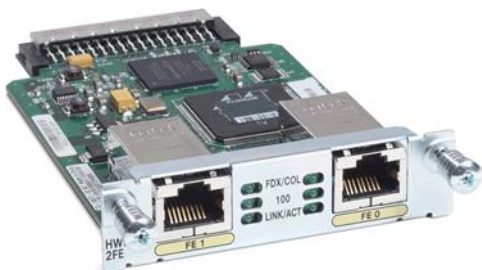
Нумерація інтерфейсів маршрутизатора проводиться починаючи з нуля. Наприклад **FastEthernet 0/0**, **FastEthernet 0/1**. У разі використання модулів та/або плат розширення вказуються номери модулів та номери плат. Наприклад, **Ethernet 0/0/0**, **Fast Ethernet 0/0/1**; **Serial 0/0/0**, **Serial 0/1/0**. Для зручності введення в інтерфейсі командного рядка використовують скорочення. Наприклад, **f0/0**, **s0/0/1**, **g0/0/1**. Зовнішній вигляд плат розширення показано на рисунку 3.6.



а) Cisco WIC-1T (последовний інтерфейс, роз'єм V.35)



б) Cisco HWIC-1T1/E1 (последовний інтерфейс, роз'єм RJ-48)



в) Cisco HWIC-2FE (2 інтерфейси Fast Ethernet, роз'єм RJ-45)



г) Cisco WIC-2A/S (2 последовних інтерфейси, роз'єм Smart Serial)



д) Cisco HWIC-4ESW (4 інтерфейси Fast Ethernet, роз'єм RJ-45)

Рисунок 3.6 – Зовнішній вигляд модулів і плат розширення

3.2.2 ПРОЦЕС ЗАВАНТАЖЕННЯ МАРШРУТИЗАТОРА

Необхідно відмітити, що сучасний маршрутизатор – це цілісна сукупність апаратної платформи та спеціалізованої мережної операційної системи. У маршрутизаторах *Cisco* застосовуються такі мережні ОС: *Cisco IOS*, *Cisco IOS XR*, *Cisco IOS XE*. З точки зору внутрішньої архітектури ці ОС відрізняються між собою. З точки зору налагодження та адміністрування вони мають багато спільних рис. Як правило *Cisco IOS XR* та *Cisco IOS XE* застосовують у високошвидкісних та високопродуктивних магістральних маршрутизаторах, *Cisco IOS* – у простіших маршрутизаторах мереж розподілу та доступу.

Порядок завантаження маршрутизатора *Cisco* збігається з порядком завантаження комутатора *Cisco*. Певні відмінності наявні лише на етапі вибору джерела завантаження образу *Cisco IOS* та джерела завантаження конфігураційного файлу. Узагальнений алгоритм завантаження маршрутизатора *Cisco* наведено на рисунку 3.7.

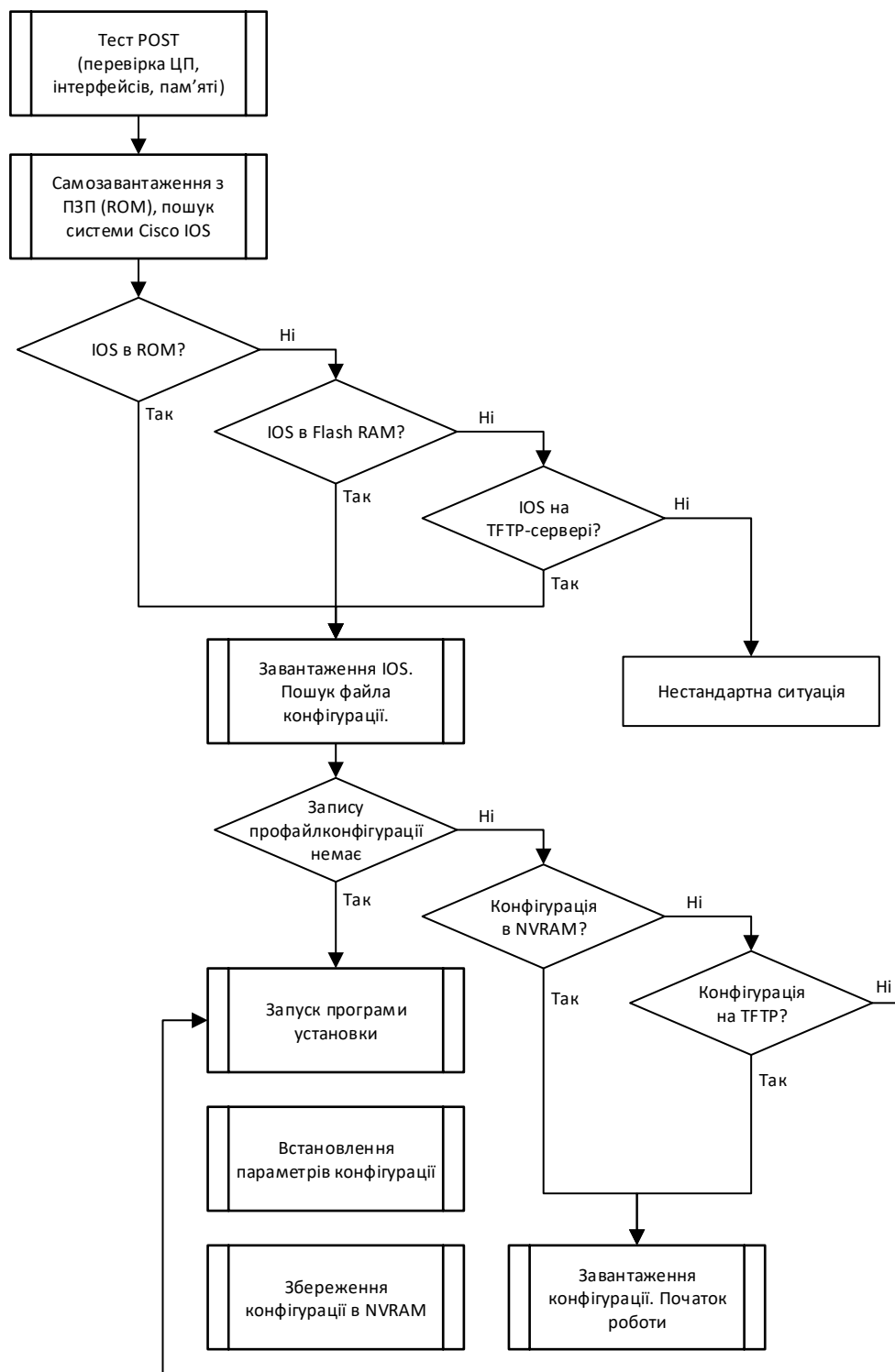


Рисунок 3.7 – Алгоритм завантаження маршрутизатора Cisco

КОНФІГУРАЦІЙНІ ФАЙЛИ

Є два системні файли, які зберігають конфігурацію пристрою:

- **startup-config** – це збережений файл початкової конфігурації, який зберігається в *NVRAM*. Він містить усі команди, які будуть використовуватися пристроєм під час запуску або перезавантаження. Флеш-накопичувач не втрачає свого вмісту, коли пристрій вимкнено.

- **running-config** – це файл поточної конфігурації, який зберігається в оперативній пам'яті (RAM). В ньому відображена поточна конфігурація. Зміна поточної конфігурації негайно впливає на роботу пристрою Cisco. Оперативна пам'ять (RAM) – енергозалежна пам'ять: вона втрачає весь свій вміст при вимкненні або перезавантаженні пристрою.

3.2.3 ТИПИ ПІД'ЄДНАННЯ ДО ОБЛАДНАННЯ

Налагодження та керування комутатором (маршрутизатором) може здійснюватися з використанням таких видів під'єднань:

1. Консольне під'єднання (*Console Connection*).
2. Допоміжне під'єднання (*Auxiliary Connection*).
3. Мережне керуюче під'єднання (*Network Management Connection*).
4. Мережне під'єднання (*Network Connection*).

Термін «під'єднання» (*Connection*) охоплює як фізичну (пряме кабельне з'єднання чи з'єднання через наявну мережну інфраструктуру), так і програмну складові (програму – термінальний клієнт). Для пристроїв Cisco поряд із терміном «під'єднання» застовується термін-синонім «лінія» (*Line*).

Консольне під'єднання (*Console Connection*) – це пряме з'єднання послідовного порту комп'ютера з консольним портом комутатора чи маршрутизатора за допомогою спеціального консольного кабелю (рисунок 3.8). Це під'єднання також позначається як *Console Out-of-Band Connection*. Воно є основним типом під'єднання для початкового налагодження пристроїв з коробки. На схемах консольне під'єднання позначається лінією з коротких штрихів.

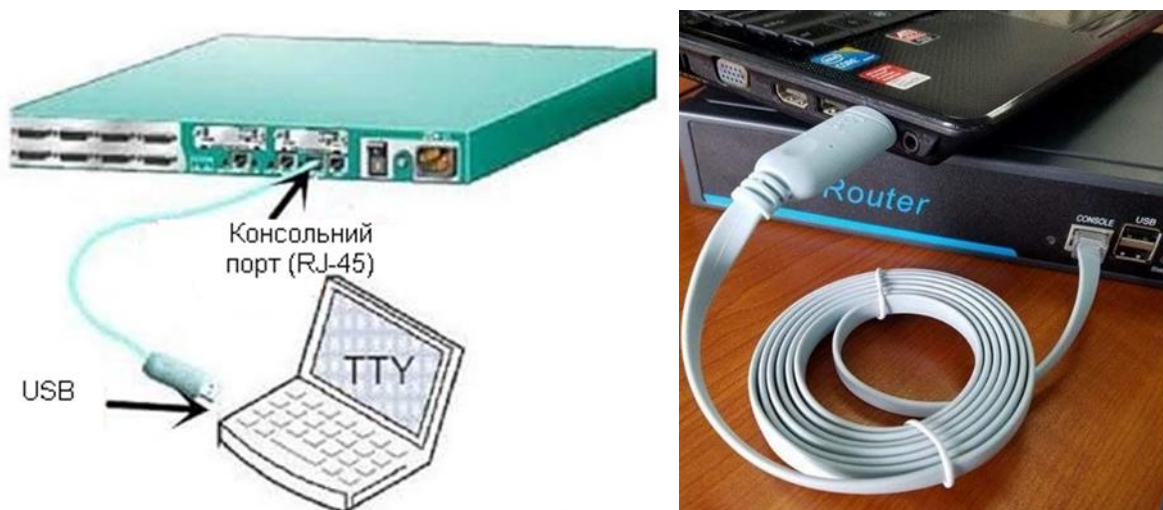


Рисунок 3.8 – Консольне під'єднання до пристроїв Cisco

Основним типом під'єднання для поточного налагодження, керування та діагностування процесів роботи є мережне під'єднання (*Network Connection*) –

під'єднання через наявну мережну інфраструктуру. Це під'єднання також позначається як *Network In-Band Connection*.

У всіх випадках під'єднання використовуються спеціальні термінальні програмні застосунки, що мають засоби забезпечення функціонування як прямого під'єднання, так і мережного, що формуються з використанням протоколів віддаленого доступу.

У більшості сучасних моделей пристроїв Cisco для консольного під'єднання наявні або одне гніздо роз'єму *RJ-45 (8P8C)*, або два гнізда – роз'єму *RJ-45* і роз'єму *USB 5 pin mini-Type B* одночасно. Слід зазначити, що у певний момент часу активним може бути лише один консольний порт. Більшість виробників мережного обладнання у своїх пристроях застосовують аналогічні роз'єми. Консольний порт може розміщуватися як на передній, так і на задній панелях пристроїв. Для моніторингу функціонування консольного порту, які і решти портів, застосовуються світлодіодні індикатори.

ПРОГРАМНІ ЗАСОБИ ЗАБЕЗПЕЧЕННЯ ПІД'ЄДНАННЯ

Налагодження та керування функціонуванням пристроїв Cisco може здійснюватися з використанням різних підходів та засобів. З метою налагодження з використанням консольного та допоміжного під'єднання на комп'ютері адміністратора необхідно застосовувати спеціальні програми – емулятори терміналу (термінальні додатки). Найбільш відомими термінальними додатками є *Hyper Terminal*, *PuTTY*, *Tera Term*, *SecureCRT*, *Minicom*, *ZTerm Pro*. Інтерфейс додатку *PuTTY* в режимі створення з'єднання наведено на рисунку 3.9.

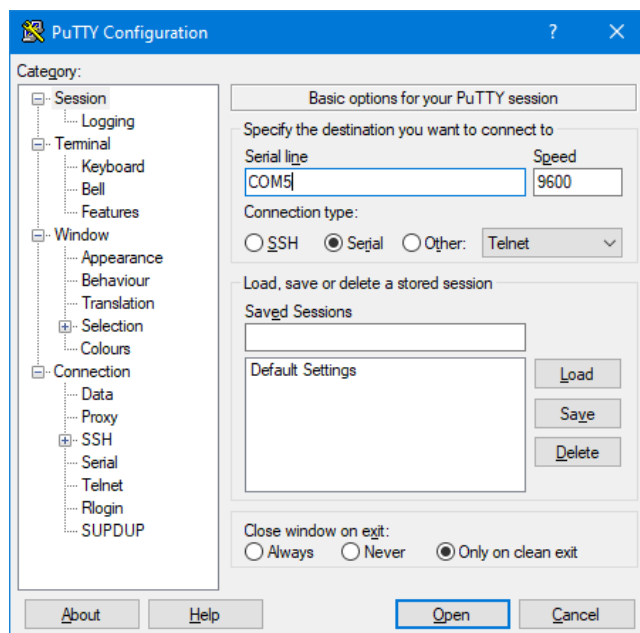


Рисунок 3.9 – Інтерфейс термінального застосунку *PuTTY*

Для налагодження комутатора за допомогою мережного керуючого підключення або мережного підключення необхідно застосовувати засоби, що

забезпечують можливість віддаленого доступу на базі мережних протоколів *Telnet*, *SSH*, *HTTP*, *SNMP* тощо. Вищезгадані термінальні додатки теж підтримують цю можливість.

Фірмою *Cisco* розроблено ряд спеціалізованих програмних додатків, які дають змогу проводити віддалене налагодження та керування комунікаційними пристроями. Серед них слід згадати такі: *Cisco Network Assistant*; *Cisco Device Manager*; *CiscoWorks LAN Management Solution (LMS)*; *CiscoView Application*; *Cisco Configuration Engine*; *SNMP Network Management Application*; *Cisco Security Manager*; *Catalyst Smart Operations*. Ці засоби, як правило, мають графічний інтерфейс та використовують для обміну даними між пристроями протоколи прикладного рівня (*HTTP*, *HTTPS*, *SNMP*). У багатьох випадках їх застосування передбачає початкове налагодження пристрою за допомогою консольного під'єднання.

3.3 КОНТРОЛЕРИ БЕЗДРОВОЇ МЕРЕЖІ CISCO

3.3.1 ПРИЗНАЧЕННЯ ТА СТРУКТУРА КОНТРОЛЕРА

Контролер бездротової локальної мережі, або WLAN-контролер (WLC), контролює та керує бездротовими точками доступу і дозволяє бездротовим пристроям підключатися до WLAN – архітектури бездротової мережі. Як централізований пристрій у мережі, контролер бездротової локальної мережі зазвичай знаходиться в центрі обробки даних, до якого прямо чи опосередковано під'єднані всі бездротові точки доступу в мережі.

Серед переваг використання WLC можна виокремити такі:

- Захищена дротова та бездротова мережа. Керування правами доступу бездротових користувачів за допомогою різноманітних визначених критеріїв (метод автентифікації, тип пристрою, запитувана програма тощо) для забезпечення диференційованого доступу з метою підтримання безпеки. Замість того, щоб залишати шифрування в межах точки доступу, воно допомагає повністю ізолювати трафік бездротової мережі, поки він не пройде через брандмауер на WLC. Виявлення та зупинка несанкціонованих (неавторизованих) точок доступу для запобігання несанкціонованому бездротовому під'єднанню в мережі.

- Централізоване та гнучке керування мережею. Централізований WLC забезпечує гнучкість розгортання, що дозволяє скоротити загальний бюджет, інструменти планування і час, витрачений на організацію бездротової мережі на підприємстві. Забезпечення централізованого моніторингу всієї бездротової інфраструктури, що знизить загальну вартість і спростить об'єднання дротового і бездротового доступу, що є перспективною інвестицією для модернізації в майбутньому.

– Спрощене обслуговування мережі. WLC усувають необхідність обстеження об'єктів завдяки інтелектуальному програмному забезпеченню для радіочастотного планування. Бездротова мережа, що самоконфігурується та самовідновлюється, краще підходить для управління та усунення несправностей. Вона може точно визначити місцезнаходження та ідентифікувати кожного користувача. Завдяки роботі з радіочастотним середовищем WLC може легко виявити перешкоди між сусідніми точками доступу та автоматично переналаштувати їхню потужність та вибір каналів. Якщо одна з точок доступу виходить з ладу, вона може дати вказівку сусіднім точкам збільшити рівень потужності, щоб заповнити прогалину в покритті.

WLC виробництва Cisco представлені лінійкою як самостійних пристроїв, так і модулів, що розширюють функції маршрутизаторів чи комутаторів. Сам принцип їх роботи майже не відрізняється, проте відмінності полягають у кількості точок доступу та стандартів бездротового зв'язку, які підтримуються, наявності інтересів під'єднання до мережі, типів керування, безпекових функцій тощо. На рисунку 3.10 показано передню та задню панель контролера бездротової мережі Cisco серії 2100 з поясненням функціонального призначення розташованих елементів.

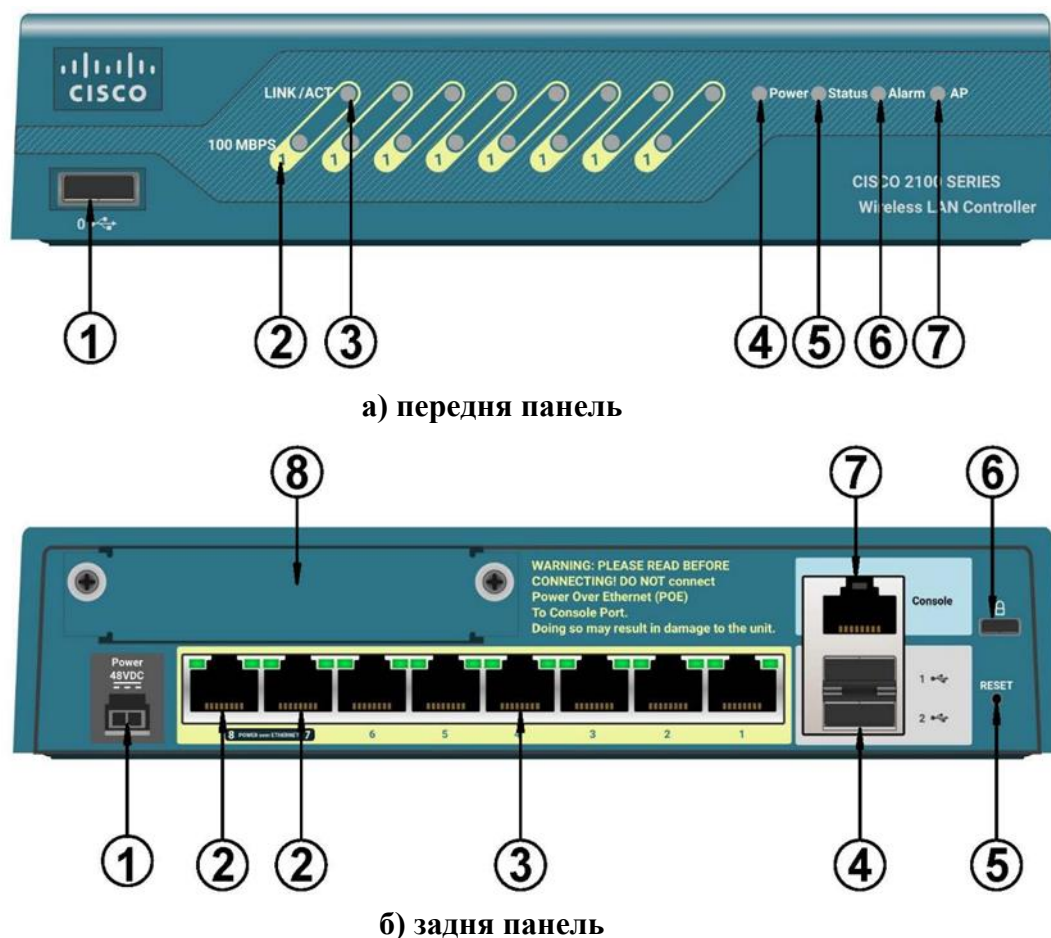


Рисунок 3.10 – Зовнішній вигляд контролера бездротової мережі

Позначення елементів передньої панелі:

1. **Порт USB** – в даній моделі не використовується.
2. **Індикатори швидкості портів.** Вимкнений – порт працює на швидкості 10 Мбіт/с, зелений – на швидкості 100 Мбіт/с.
3. **Індикатори активності з'єднань.** Зелений – фізичне з'єднання встановлено, блимає зеленим – присутня мережна активність.
4. **Індикатор живлення.** Зелений – контролер увімкнений, вимкнений – контролер вимкнено.
5. **Індикатор стану.** Блимає зеленим – діагностика увімкнення/завантаження, зелений – контролер працює, бурштиновий – під час завантаження виникла проблема.
6. **Індикатор тривоги.** Зелений – не використовується, бурштиновий – існує непогашена тривога.
7. **Індикатор точки доступу.** Зелений – принаймні одна точка доступу приєдналася, вимкнено – жодної точки доступу не приєднано.

Позначення елементів передньої панелі:

1. **Вилка живлення.** Для під'єднання джерела живлення 48 В.
2. **Порти з підтримкою живлення через *Ethernet*.** Порти з підтримкою живлення по витій парі (*PoE*) з номерами 7-8.
3. **Порти комутатора.** Порти без підтримки живлення по витій парі з номерами 1-6.
4. **Порти USB** – для під'єднання накопичувачів.
5. **Кнопка скидання.** Спеціалізована кнопка внутрішнього розташування для скидання налаштувань до заводських.
6. **Гніздо для замка.** Кріплення для забезпечення фіксації пристрою.
7. **Консольний порт.** Порт для під'єднання до контролера бездротової мережі для його налаштування.
8. **Гніздо для карток розширення.** В цій моделі не використовується.

3.3.2 ІНТЕРФЕЙСИ КОНТРОЛЕРА

Кожен WLC крім портів (фізичних інтерфейсів) мають ще і певні логічні інтерфейси, які є критично важливими для правильної роботи пристрою та його інтеграції з мережевою інфраструктурою. Типова структурна схема контролера показана на рисунку 3.11.

Логічні інтерфейси WLC використовуються для керування ідентифікаторами бездротових мереж (*SSID*), що транслюються точками доступу, керування контролером, точками доступу і даними користувачів, а також для інших цілей.

Структурна схема на рисунку 3.10 показує, як кожен бездротовий *SSID* (WLAN 1, WLAN 2) зіставляється з динамічним інтерфейсом. У свою чергу, кожен динамічний інтерфейс зіставляється з певною віртуальною локальною мережею. Кількість WLAN та динамічних інтерфейсів залежить від моделі WLC. Чим

потужніша модель *WLC*, тим більше *SSID* (бездротових мереж)/динамічних інтерфейсів вона підтримує.

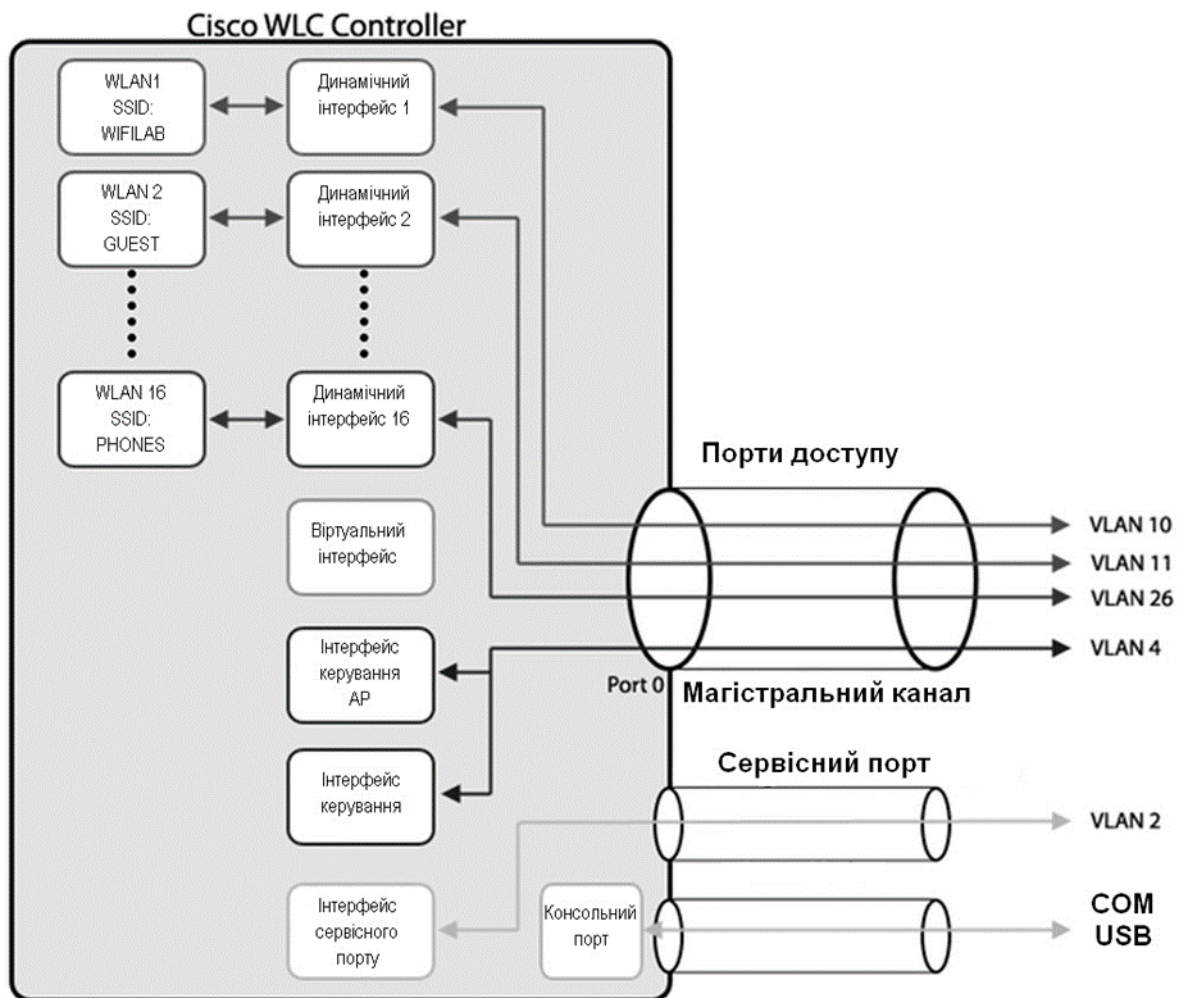


Рисунок 3.11 – Структурна схема контролера бездротової мережі

Всі динамічні інтерфейси та інтерфейси AP-Manager/Manager підключаються до мережевої інфраструктури через розподільчі порти, які, залежно від моделі *WLC*, є інтерфейсами *SFP* або *Ethernet* (10/100 або Gigabit).

Оскільки всі *WLC* мають кілька фізичних розподільчих портів, можна призначити всі динамічні інтерфейси та інтерфейси AP-Manager/Manager до одного фізичного розподільчого порту, як показано на схемі. У цьому випадку порт розподілу буде налаштовано як магістральний порт *802.1Q*. Крім того, динамічні інтерфейси також можуть бути призначені окремим фізичним розподільчим портам, щоб певний *WLAN*/динамічний інтерфейс міг створити тунель для свого трафіку через один розподільчий порт.

Виділений сервісний порт, показаний на наведеній вище схемі, можна знайти тільки на пристроях *WLC* серій 5500 і 7500/8500, який підключається безпосередньо до мережі.

Інтерфейс керування – це інтерфейс за замовчуванням, який використовується для доступу до *WLC* та керування ним. Інтерфейс керування також

використовується точками доступу для зв'язку з *WLC*. *IP*-адреса інтерфейсу керування – це єдина *IP*-адреса, на яку можна надсилати пінг, і яка використовується адміністраторами для керування *WLC*.

Адміністратори можуть увійти до графічного інтерфейсу конфігурації *WLC*, ввівши *IP*-адресу інтерфейсу керування у веббраузері та увійшовши до системи.

Інтерфейс керування точками доступу (AP-Manager). Контролер може мати один або декілька інтерфейсів AP-Manager, які використовуються для комунікацій Рівня 3-го між контролером і полегшеними точками доступу (*LAP*) після їх підключення до контролера. *IP*-адреса цього інтерфейсу використовується як джерело для пакетів тунелю *CAPWAP/LWAPP* від контролера до точок доступу і як *IP*-адреса призначення для пакетів *CAPWAP/LWAPP* від точок доступу до контролера.

Хоча конфігурація і використання інтерфейсів AP-Manager не є обов'язковими, такі моделі, як *WLC2504* і *WLC5508*, не мають спеціального інтерфейсу AP-Manager.

Віртуальний інтерфейс використовується для керування та підтримки бездротових клієнтів, надаючи функції ретрансляції *DHCP*, гостьової вебавтентифікації, завершення VPN та інші послуги. Віртуальний інтерфейс виконує дві основні ролі:

1. Виконує роль ретранслятора *DHCP*-сервера для бездротових клієнтів, які отримують свою *IP*-адресу від *DHCP*-сервера.
2. Слугує адресою перенаправлення на сторінку входу для вебавтентифікації (якщо налаштовано).

IP-адреса віртуального інтерфейсу використовується лише для зв'язку між контролером і бездротовими клієнтами. Вона ніколи не з'являється як адреса джерела або призначення пакетів, які виходять через розподільчі порти в локальну мережу. Тому, *IP*-адреса віртуального інтерфейсу повинна бути унікальною в мережі. З цієї причини для віртуального інтерфейсу використовується загальна *IP*-адреса 1.1.1.1. Всі контролери в групі мобільності повинні бути налаштовані з однаковою *IP*-адресою віртуального інтерфейсу, щоб забезпечити коректну роботу роумінгу між контролерами без втрати зв'язку.

Інтерфейс сервісного порту використовується для позасмугового керування контролером. Якщо робоча станція керування знаходиться у віддаленій підмережі, може знадобитися додати *IPv4*-маршрут на контролері, щоб керувати контролером з віддаленої робочої станції.

Важливо зазначити, що *IP*-адреса сервісного порту не повинна знаходитися в тій самій підмережі, що й інтерфейс керування контролером та точками доступу.

Динамічний інтерфейс. Найпростіший спосіб пояснити динамічні інтерфейси – уявити їх як інтерфейси *VLAN* для створених бездротових мереж (*SSID*).

Для кожної бездротової мережі/*SSID* створюється один динамічний інтерфейс. Бездротова мережа або *SSID* зіставляється з динамічним інтерфейсом, який потім зіставляється з певною мережею *VLAN*.

Як згадувалося раніше, динамічні інтерфейси можна призначити окремим фізичним портам доступу, щоб трафік з певних мереж *WLAN* проходив до дротової мережі через певні порти. У цьому сценарії кожен порт є єдиним каналом доступу, який обслуговує лише одну віртуальну локальну мережу (*VLAN*).

Крім того, всі динамічні інтерфейси можна зіставити з одним розподільчим портом, який у цьому випадку буде магістральним портом і зможе обслуговувати всі *WLAN/VLAN*. Це звичайний метод налаштування для невеликих мереж.

Нарешті, кожен динамічний інтерфейс повинен знаходитися в іншій *VLAN* або IP-підмережі, ніж всі інші інтерфейси.

Оскільки контролер *WLC2504* може обробляти до 16 *SSID*, він може мати максимум 16 динамічних інтерфейсів і підтримувати максимум 16 *VLAN*.

Всі *WLC* підтримують об'єднання декількох розподільчих портів в один порт за допомогою стандарту портів *802.3ad*. Це дозволяє адміністратору створити одне велике з'єднання між *WLC* і локальним комутатором. Для роботи агрегації каналів необхідно налаштувати *EtherChannel* на локальному комутаторі. *WLC* не підтримують Протокол керування агрегацією каналів (*Link Aggregation Control Protocol, LACP*) або власний протокол *Cisco Port Aggregation Protocol (PAgP)*, тому комутатор повинен бути безумовно налаштований на групову агрегацію каналів (*LAG*). Для кожного контролера підтримується лише одна група *LAG*.

РОЗДІЛ 4

МЕРЕЖНА ОПЕРАЦІЙНА СИСТЕМА CISCO IOS

Для забезпечення функціонування мережевих пристроїв фірмою *Cisco* розробляються як різні варіанти мікропрограмного забезпечення, так і спеціалізовані мережні ОС. Використання мікропрограмного забезпечення характерне для точок доступу та бездротових маршрутизаторів, деяких моделей комутаторів. Більшість моделей комутаторів та маршрутизаторів *Cisco* використовують спеціалізовані мережеві ОС.

Основними мережевими ОС *Cisco* є:

- *Cisco IOS (Cisco Internetwork Operating System)*;
- *Cisco NX-OS (NeXt-generation OS, Nexus OS)*;
- *Cisco IOS XR*;
- *Cisco IOS XE*.

У сучасних комутаторах та маршрутизаторах *Cisco* найчастіше використовується спеціалізована мережева ОС *Cisco IOS*.

Cisco IOS є багатозадачною ОС, яка виконує функції мережевої організації, комутації, маршрутизації та передачі даних. Ядро цієї ОС є монолітним, це означає, що всі елементи системи розміщені в одному образі і всі процеси запускаються в одному адресному просторі. У *Cisco IOS* немає міжпроцесного захисту пам'яті, це означає, що крах одного процесу може викликати крах або перезавантаження всієї системи.

Cisco IOS поставляється у вигляді монолітного образу, який орієнтований на конкретну модель пристрою. Образи можуть мати певні набори властивостей та версії. Конкретний образ *IOS* ідентифікується трьома параметрами:

- апаратна платформа (серія) пристрою, для якої він призначений;
- набір можливостей (*Feature Set, Packages*);
- версія ОС.

4.1 РЕЖИМИ РОБОТИ ПРИСТРОЇВ ПІД КЕРУВАННЯМ CISCO IOS

Більшість пристроїв *Cisco*, в тому числі комутатори і маршрутизатори, що працюють під керуванням *Cisco IOS*, мають основні та додаткові командні режими функціонування. Комутатори і маршрутизатори відрізняються кількістю додаткових режимів конфігурування. Проте для зручності порівняння і розуміння схеми режимів роботи та переходу між ними для комутатора показано на рисунку 4.1, а для маршрутизатора – на рисунку 4.2. Кожен із режимів надає певні функціональні можливості з діагностування та налагодження роботи пристроїв. У кожному режимі користувач може оперувати певним набором команд. Зміна режимів також здійснюються за допомогою відповідних команд.

Основними режимами є такі:

- режим користувача (*User Mode, User EXEC Mode*);
- привілейований режим (*Privilege Mode, Privilege EXEC Mode*);
- режим глобального конфігурування (*Global Configuration Mode*).

Режим користувача надає обмежений доступ до пристрою, дозволяє переглядати деякі параметри конфігурації без можливості внесення змін. У привілейованому режимі є змога детально аналізувати стан роботи та налаштування пристрою. Адміністраторові надається можливість виконати велику кількість команд. У цьому режимі можливе налагодження деяких параметрів і їх збереження. У режимі глобального конфігурування, що активується тільки з привілейованого режиму, надається повний доступ до набору команд налагодження та доступ до додаткових режимів.

Додатковими режимами комутатора є:

- режим конфігурування інтерфейсу (*Interface Configuration Mode*);
- режим конфігурування групи інтерфейсів (*Interface-range Configuration Mode*);
- режим конфігурування лінії (*Line Configuration Mode*);
- режим конфігурування віртуальної локальної мережі (*Config-VLAN Mode*);
- режим конфігурування параметрів бази даних віртуальної локальної мережі (*VLAN Configuration Mode*).

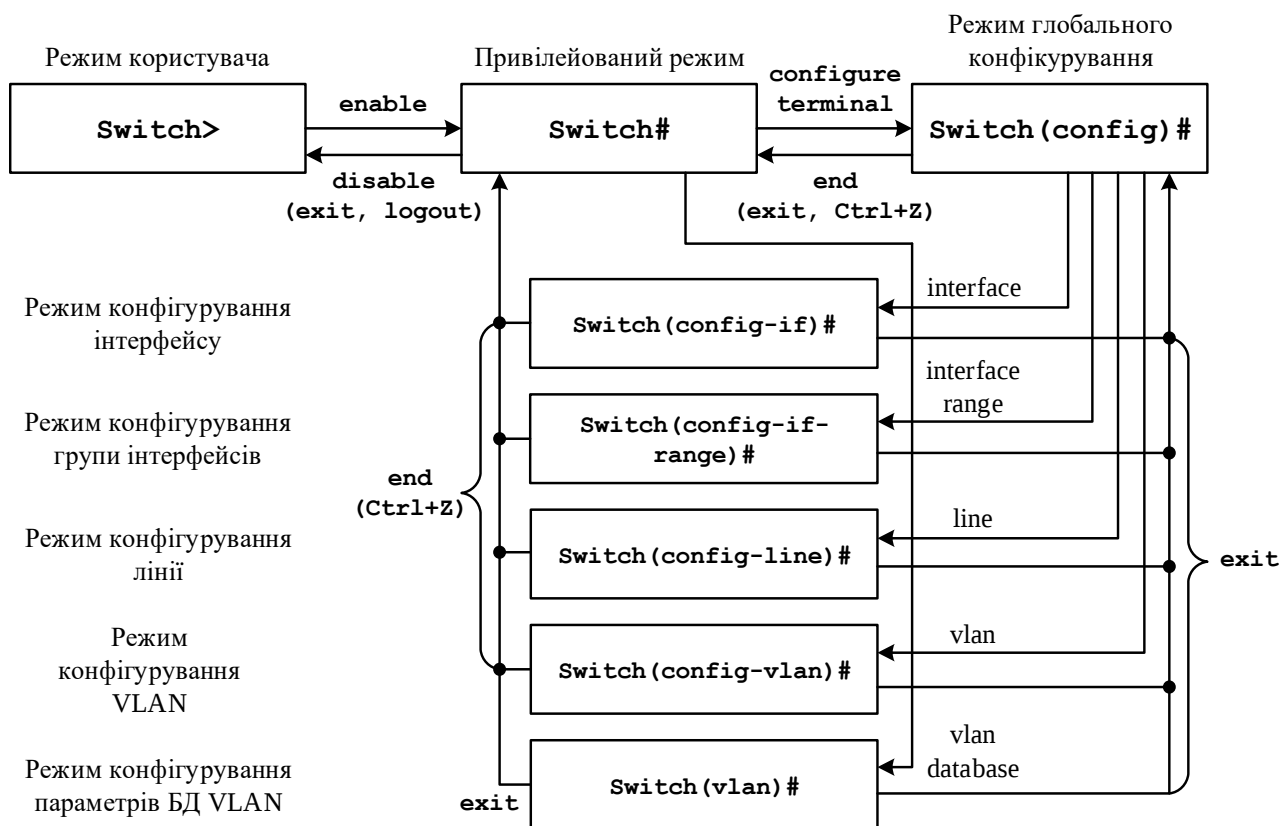


Рисунок 4.1 – Командні режими комутатора Cisco та команди переходів між режимами

Додатковими режимами маршрутизатора є:

- режим конфігурування інтерфейсу (*Interface Configuration Mode*);
- режим конфігурування підінтерфейсу (*Subinterface Configuration Mode*);
- режим конфігурування лінії (*Line Configuration Mode*);
- режим конфігурування протоколу маршрутизації (*Router Configuration Mode*);
- режим конфігурування *DHCP*-сервера (*DHCP Configuration Mode*);
- режим конфігурування стандартного списку доступу (*Standard ACL Configuration Mode*);
- режим конфігурування розширеного списку доступу (*Extended ACL Configuration Mode*).

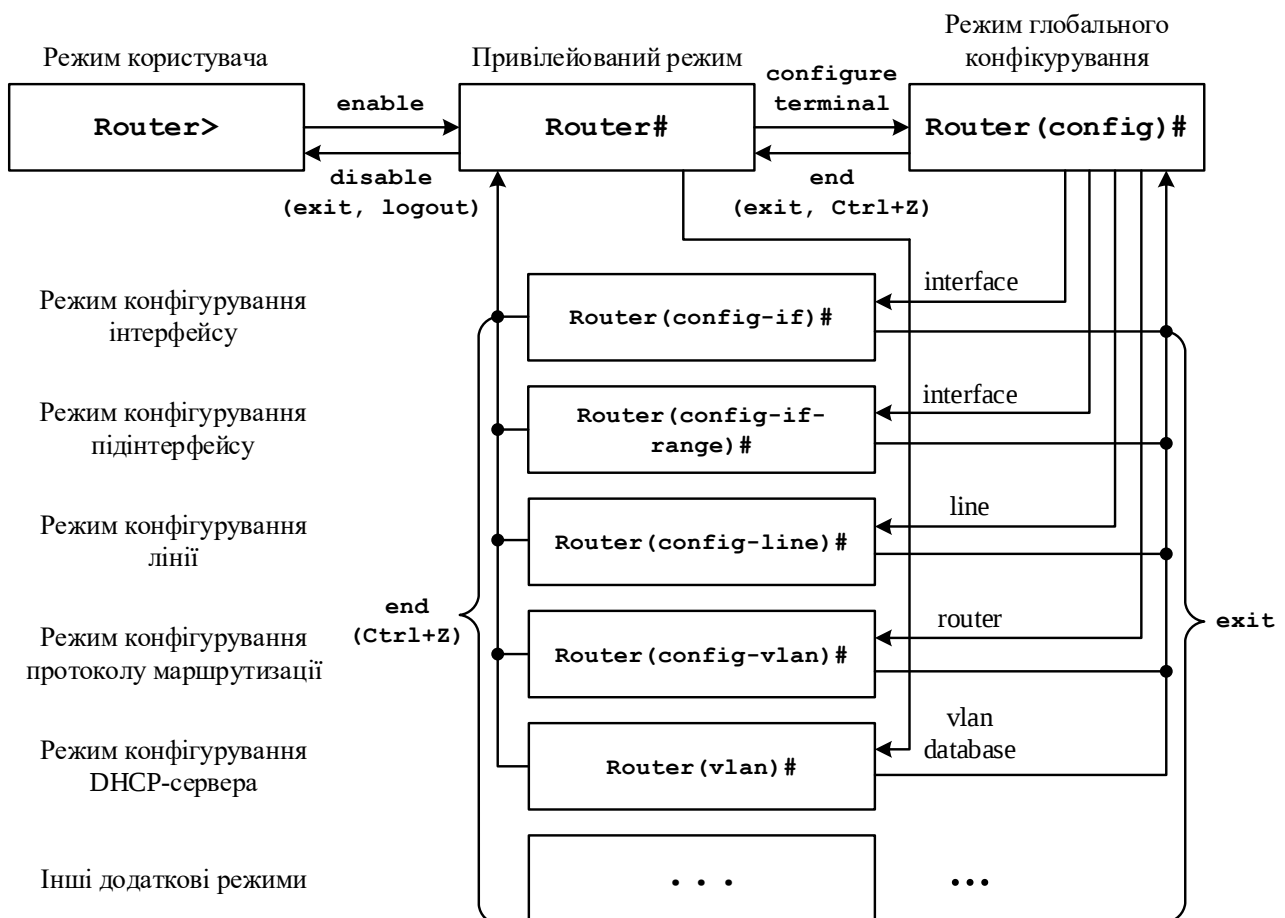


Рисунок 4.2 – Командні режими маршрутизатора Cisco та команди переходів між режимами

Після завантаження комутатор чи маршрутизатор Cisco перебуває в режимі користувача. Для переходу до привілейованого режиму використовується команда **enable**. Для повернення – або **disable**, або **exit**, або **logout**. Для переходу до режиму глобального конфігурування використовується команда **configure terminal**, для виходу – або команда **end**, або команда **exit**. Для переходів до інших режимів використовуються відповідні команди. Можливе

пряме повернення з будь-якого нижчого режиму до привілейованого режиму командою **end** або натисненням комбінації клавіш **<Ctrl>+<Z>**.

Перелік комбінацій клавіш, які можна використовувати для редагування командного рядка, наведено у таблиці 4.1.

Таблиця 4.1 – Клавіші редагування командного рядка *Cisco IOS*

Клавіша/ Комбінація клавіш	Виконувані дії
Ctrl+A	Переміщення курсора на початок рядка
Ctrl+B	Переміщення курсора на один символ назад (аналог клавіші ←)
Ctrl+D	Видалення символу ліворуч від курсора
Ctrl+E	Переміщення курсора на кінець рядка
Ctrl+F	Переміщення курсора вперед на один символ (аналог клавіші →)
Ctrl+K	Видалення всіх символів від поточної позиції курсора до кінця рядка
Ctrl+N	Перехід на наступну в «історії сеансу» команду (аналог клавіші ↓)
Ctrl+P	Перехід на попередню в «історії сеансу» команду (аналог клавіші ↑)
Ctrl+T	Міняє місцями поточний символ і символ ліворуч від курсора
Ctrl+R	Перерисовує або заново виводить поточний рядок
Ctrl+U	Очищення рядка
Ctrl+W	Видалення слова ліворуч від курсора
Ctrl+X	Видалення символів від поточної позиції курсора і до початку рядка
Ctrl+Y	Вставка символів, що видалені останніми, на місце, яке відповідає поточній позиції курсора
Ctrl+Z	Вихід із поточного режиму налагодження і перехід у привілейований режим
Tab	Доповнення поточної команди
↑	Перехід на попередній запис у списку «історії команд»
↓	Перехід на наступний запис у списку «історії команд»
←	Переміщення курсора ліворуч
→	Переміщення курсора праворуч
Ctrl + ^, X	Відміна послідовності. Переривання виконання будь-якої виконуваної команди

4.2 ЛОГІЧНІ І ФІЗИЧНІ ІНТЕРФЕЙСИ МАРШРУТИЗАТОРА CISCO

Інтерфейси маршрутизатора *Cisco* з точки зору адміністрування можна поділити на дві групи: фізичні інтерфейси та логічні інтерфейси. Фізичні інтерфейси – це інтерфейси відповідних мережних технологій. Логічні інтерфейси – це інтерфейси, які автоматично створені операційною системою *Cisco IOS* для виконання певних функцій, або інтерфейси, які створюються адміністратором із

певною метою. Позначення і налагодження фізичних інтерфейсів не залежить від того, є вони електричними чи оптичними. Слід зазначити, що фізичні інтерфейси маршрутизатора *Cisco* за замовчуванням є неактивними. Активність логічних інтерфейсів залежить від їх видів та зв'язків з фізичними інтерфейсами. Перелік та опис основних фізичних і логічних інтерфейсів маршрутизатора *Cisco* наведено у таблиці 4.2.

Таблиця 4.2 – Основні фізичні і логічні інтерфейси маршрутизатора *Cisco*

Назва інтерфейсу	Опис інтерфейсу
Фізичні інтерфейси	
Ethernet	Інтерфейс класичного Ethernet, 10 Мбіт/с
FastEthernet	Інтерфейс FastEthernet, 100 Мбіт/с
GigabitEthernet	Інтерфейс GigabitEthernet, 1 Гбіт/с
TenGigabitEthernet	Інтерфейс 10 GigabitEthernet, 10 Гбіт/с
Dot11radio	Інтерфейс Wi-Fi
Serial	Послідовний інтерфейс
POS	Інтерфейс Packet over SONET
ATM	Інтерфейс технології ATM (DSL-з'єднання також)
Async	Асинхронний інтерфейс
BRI	Інтерфейс BRI мереж технології ISDN (служба 2B+D)
Tokenring	Інтерфейс Token Ring
FDDI	Інтерфейс FDDI
Hub	Вбудований концентратор (вважається інтерфейсом)
HSSI	Високошвидкісний (до 52 Мбіт/с) послідовний інтерфейс
Логічні інтерфейси	
Null	Інтерфейс бітоприймача. Будь-які дані, відіслані на даний інтерфейс, видаляються. Використовується для простої фільтрації маршрутів.
PortChannel	Інтерфейс агрегованого каналу (L3 EtherChannel).
Tunnel	Інтерфейс віртуального GRE/IP Security тунелю.
Loopback	Віртуальний інтерфейс маршрутизатора (не плутати з Loop-back/Localhost/127.0.0.1 звичайного комп'ютера).
Dialer	Інтерфейс маршрутизації викликів за запитом.
Virtual-Template	Інтерфейс віртуального тунелю IP-Security.
Multilink	Інтерфейс агрегованого за протоколом MLPPP каналу (канал складається з кількох фізичних послідовних каналів).
BVI	Інтерфейс віртуального мосту.
Group-Async	Логічна група асинхронних інтерфейсів.
VLAN	Інтерфейс VLAN. На деяких маршрутизаторах автоматично створено vlan 1 за замовчуванням.

4.3 КОМАНДИ БАЗОВОГО НАЛАШТУВАННЯ КЕРОВАНОГО КОМУТАТОРА CISCO

Налаштування керованого комутатора *Cisco* передбачає такі кроки: надання імені пристрою та введення назви домену, встановлення системного годинника, налаштування параметрів консольного під'єднання, типу та способу відділеного під'єднання, часових інтервалів від'єднання сеансу при бездіяльності користувача, системних повідомлень, забезпечення безпеки пристрою, введення параметрів *IP*-адресації для віртуального інтерфейсу, основного шлюзу тощо.

Необхідність надання імен для пристроїв під керуванням *Cisco IOS* зумовлюється такими чинниками:

- потребою ідентифікації пристрою під час під'єднання як консольного, так і в разі внутрішньо-смугового керування за допомогою протоколів віддаленого доступу *Telnet* чи *SSH*;
- для обміну інформацією про пристрій із сусідніми пристроями (наприклад, за протоколами *LLDP* чи *CDP*);
- для генерування *RSA* ключів у разі використання криптографічних засобів захисту сеансу (наприклад, у *SSH*).

Для зміни імені пристрою призначена команда **hostname**. Повернення імені пристрою за замовчуванням – **no hostname**. За замовчуванням заводське ім'я комутатора **Switch**.

Операційна система *Cisco IOS* на пристроях *Cisco* забезпечує функціонування системного (програмного) годинника/календаря. В деяких моделях пристроїв також наявний і апаратний годинник/календар. Відповідно існують механізми обміну даними між ними. Параметри системних часу (та дати) пристрою можуть встановлюватися як за допомогою команд локального застосування, так і з використанням мережевого джерела часу. У першому випадку за умови відсутності апаратного годинника параметри системного часу не зберігаються у конфігураційному файлі і є актуальними лише на період роботи пристрою. Після перезавантаження їх необхідно встановлювати заново. У другому випадку системний час після завантаження пристрою синхронізується з часом сервера часу за протоколом *NTP*. Надалі операція синхронізації виконується періодично. Звичайно, що це потребує певних специфічних налагоджень.

Встановлення системного часу здійснюється за допомогою команди **clock set**, встановлення часового поясу – за допомогою команди **clock timezone**. Для активації переходу на літній час застосовується команда **clock summertime**. Для виведення параметрів часу та дати апаратного годинника пристрою у ручному режимі застосовується команда **clock read-calendar**. Для налагодження використання апаратного годинника пристрою як авторитетного джерела мережевого часу застосовується команда **clock calendar-**

valid. Для одноразової ручної синхронізації параметрів часу апаратного годинника з параметрами часу програмного годинника пристрою застосовується команда **clock update-calendar**. Синтаксис розглянутих команд наведено нижче.

4.4 ОСНОВНІ КОМАНДИ ДІАГНОСТИКИ ПАРАМЕТРІВ РОБОТИ КОМУТАТОРА CISCO

Для виведення діагностичної інформації про фізичні параметри комутатора, результати налагоджень, результати роботи комутатора, стан комутатора тощо використовується команда **show**. Вона є доступною як із режиму користувача, так і з привілейованого режиму. Залежно від режиму дана команда може мати різні параметри. Часто команда **show** із певним параметром вважається окремою командою. Перелік основних команд **show** та їх призначення наведено у таблиці 4.3.

Таблиця 4.3 – Основні команди **show**

Команда	Призначення
show version	Виведення технічної інформації про пристрій
show tech-support	Виведення розширеної технічної інформації про пристрій
show flash	Виведення вмісту <i>Flash</i> -пам'яті
show boot	Виведення параметрів завантаження
show processes	Виведення інформації про стан процесів, що запущені в системі
show terminal	Виведення параметрів роботи термінала
show clock	Виведення параметрів системного часу
show history	Виведення історії команд
show running-config	Виведення поточної конфігурації комутатора
show startup-config	Виведення стартової конфігурації комутатора

4.5 ПОРЯДОК НАЛАШТУВАННЯ ІНТЕРФЕЙСІВ МАРШРУТИЗАТОРА CISCO

Специфіка налагодження певного інтерфейсу маршрутизатора *Cisco* зумовлена специфікою використання відповідної мережевої технології чи з'єднання. Наприклад, для маршрутизатора інтерфейс *Ethernet* одночасно є і *DTE*, і *DCE*-пристроєм, тому не виникає потреби в деталізації функцій інтерфейсу. Послідовне з'єднання в одних випадках при використанні певного типу кабелю передбачає виділення ролі *DTE*-пристрою для одного маршрутизатора та *DCE* для іншого, в інших випадках – ні.

Порядок налаштування *Ethernet* інтерфейсу на маршрутизаторах *Cisco* згідно з рекомендаціями виробника є наступним:

1. Вибір інтерфейсу (обов'язково).
2. Присвоєння *IP*-адреси та маски підмережі (обов'язково).
3. Активація інтерфейсу(обов'язково).
4. Налаштування додаткових параметрів: пропускної здатності, затримки, опису тощо (необов'язково).

Порядок налаштування послідовного інтерфейсу при використанні прямого з'єднання нуль-модемним кабелем на маршрутизаторах *Cisco* згідно з рекомендаціями виробника є наступним:

1. Вибір інтерфейсу (обов'язково).
2. Налаштування на одному з двох інтерфейсів ролі інтерфейсу *DCE* (обов'язково).
3. Присвоєння *IP*-адреси та маски підмережі (обов'язково).
4. Активація інтерфейсу(обов'язково).
5. Налаштування додаткових параметрів: пропускної здатності, затримки, опису тощо (необов'язково).

4.6 ОСНОВНІ КОМАНДИ НАЛАГОДЖЕННЯ ПАРАМЕТРІВ ІНТЕРФЕЙСІВ МАРШРУТИЗАТОРА

Вибір інтерфейсу маршрутизатора для налагодження виконується командою **interface**. Можливе одночасне налагодження групи інтерфейсів. Для цього використовується команда **interface range**. Слід нагадати, що за замовчуванням фізичні інтерфейси маршрутизатора знаходяться у відключеному стані, а логічні інтерфейси залежно від типу можуть знаходитися як у відключеному, так і включеному станах. Відключення інтерфейсу виконується командою **shutdown**, увімкнення – командою **no shutdown**. Для налагодження параметрів інтерфейсів маршрутизатора, залежно від їх типу використовується достатньо великий набір команд. Більшість команд є загальними для всіх інтерфейсів, частина – характерними лише для інтерфейсів певних технологій.

Основними командами налаштування параметрів фізичного і каналного рівня для інтерфейсів маршрутизатора є такі: **arp**, **bandwidth**, **clock rate**, **delay**, **description**, **duplex**, **encapsulation**, **keepalive**, **ip**, **mac-address**, **mtu**, **speed**. Відміна дії команд – використання форми **no**, або команда **default**.

Команда **arp** та її модифікації служать для обробки *ARP*-запитів та їх параметрів на інтерфейсі. Команда **bandwidth** служить для встановлення значення пропускної здатності, що використовується при обчисленні метрик маршрутів у протоколах маршрутизації, не встановлює швидкість передачі даних інтерфейсу і не впливає на фактичну швидкість передачі даних по каналу зв'язку. Команда **clock rate** служить для налаштування частоти тактових імпульсів

на одному з пари інтерфейсів (типу *DCE*), що формують прямий двоточковий послідовний канал між двома маршрутизаторами (з'єднання типу нуль-модем).

При підключенні маршрутизатора через *DCE*-пристрій (наприклад, *CSU/DSU*) команда не задається, оскільки синхронізація здійснюється провайдером послуг. Команда **delay** служить для встановлення значення затримки на інтерфейсі, це значення використовується при обчисленні метрик у деяких протоколах маршрутизації, команда не визначає параметрів інтерфейсу.

Команда **description** служить для опису інтерфейсу, використовується з метою полегшення аналізу результатів виводу команд при адмініструванні. Команда **duplex** (та її модифікації **duplex-full**, **duplex-half**) служать для зазначення режиму передачі даних на інтерфейсі. Команда **encapsulation** служить для налаштування типу інкапсуляції на інтерфейсі. Часто використовується на послідовних інтерфейсах для зазначення протоколу або технології каналного рівня, на інтерфейсах *Ethernet* використовується для тегування *VLAN* (як *802.1Q*, так і *ISL*). Команда **keepalive** служить для зазначення інтервалу, протягом якого маршрутизатор буде очікувати перед тим, як відправити через інтерфейс повідомлення про перевірку зв'язку для визначення чи працює інтерфейс на іншому кінці послідовного каналу. На *Ethernet*-інтерфейсах маршрутизатор пересилає повідомлення самому собі. Команда **mtu** служить для зазначення *MTU* інтерфейса, це значення варто змінювати для оптимізації продуктивності мережі, наприклад, для каналів з великими втратами його варто зменшувати.

Синтаксис команди **interface** (режим глобального конфігурування):

```
interface interface-type interface-id.subinterface-id  
[{{point-to-point | multipoint}}]
```

де **interface-type** – тип інтерфейсу, може набувати значення *Ethernet*, *FastEthernet*, *Serial*, *ATM*, *Loopback*, *Tunnel*, *Vlan* та ін.;

interface-id – ідентифікатор інтерфейсу, може мати одночислове позначення *number* (номер інтерфейсу), двочислове позначення *module/number* (номер модуля (адаптера) / номер інтерфейсу), тричислове позначення **slot/module/number** (номер слота / номер модуля (адаптера) / номер інтерфейсу);

subinterface-id – ідентифікатор підінтерфейсу, може набувати значення від 0 до 4294967295, за замовчуванням інтерфейс не містить підінтерфейсів: вони створюються у процесі виконання команди **interface**; підінтерфейси використовуються для забезпечення роботи протоколу *802.1Q* та технологій *Frame Relay* і *ATM*;

point-to-point – службова конструкція, яка зазначає, що підінтерфейс логічно з'єднаний з одним віддаленим вузлом;

multipoint – службова конструкція, яка зазначає, що підінтерфейс логічно з'єднаний з кількома віддаленими вузлами.

Приклад:

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface GigabitEthernet 0/0/0
Router(config-if)#
```

Синтаксис команди **bandwidth** (режим конфігурування інтерфейсу):

bandwidth *value*

де **value** – значення пропускної здатності в Кбіт/с, за замовчуванням залежить від типу інтерфейсу.

Синтаксис команди **clock rate** (режим конфігурування інтерфейсу):

clock rate *bps*

де **bps** – значення частоти тактових імпульсів (біт/с), може приймати значення 1200, 2400, 4800, 9600, 19 200, 38400, 56000, 64000, 72000, 125000, 148000, 500000, 800000, 1000000, 1300000, 2000000, 4000000.

Синтаксис команди **delay** (режим конфігурування інтерфейсу):

delay *value*

де **value** – значення затримки на інтерфейсі в десятках мілісекунд, за замовчуванням залежить від типу інтерфейсу.

Синтаксис команди **description** (режим конфігурування інтерфейсу):

description *text-line*

де **text-line** – тестовий рядок опису інтерфейсу (до 240 символів).

Синтаксис команди **duplex** (режим конфігурування інтерфейсу):

duplex {auto | full | half}

де **auto** – автоматичний вибір режиму;

full – повнодуплексний режим;

half – напівдуплексний режим.

Синтаксис команди **ip address** (режим конфігурування інтерфейсу):

ip address {*address network_mask*} | dhcp

де **address** – IP-адреса в десятковому записі;

network_mask – маска мережі, записана у звичайній формі;

dhcp – службова конструкція, яка вказує, що IP-адресу необхідно отримати автоматично по протоколу DHCP.

Синтаксис команди **mac-address** (режим конфігурування інтерфейсу):

mac-address *hw-address*

де **hw-address** – MAC-адреса інтерфейсу у вигляді **НННН.НННН.НННН**, кожне число **НННН** має довжину 2 байти і записується в шістнадцятковій формі.

Синтаксис команди **mtu** (режим конфігурування інтерфейсу):

mtu *value*

де **value** – значення *MTU* в байтах, значення за замовчуванням залежить від технології або протоколу канального рівня.

Синтаксис команди **speed** (режим конфігурування інтерфейсу):

speed {10 | 100 | 1000 | auto [10 | 100 | 1000] |
nonegotiate}

де **10, 100, 1000** – значення швидкості в Мбіт/с;

auto – службова конструкція, яка вказує на автоматичний вибір швидкості; наприклад, якщо використовується форма **auto 10 (auto 100, auto 1000)**, то інтерфейс веде перемовини лише на цій швидкості;

nonegotiate – службова конструкція, яка відключає режим автоперемовин про швидкість.

Приклад налаштування інтерфейсу **GigabitEthernet**:

```
Router(config)#interface GigabitEthernet 0/0/0
Router(config-if)#description LINK_TO_R_2
Router(config-if)#speed 1000
Router(config-if)#duplex full
Router(config-if)#mac-address 00aa.00ad.0001
Router(config-if)#ip address 192.168.1.1 255.255-255.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

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

Приклад налаштування інтерфейсу **Serial**:

```
Router(config)#interface Serial 0/1/0
Router(config-if)#description LINK_TO_R_2
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 128
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to down
Router(config-if)#exit
```

4.7 ОСНОВНІ КОМАНДИ ДЛЯ БАЗОВОЇ ДІАГНОСТИКИ РОБОТИ МАРШРУТИЗАТОРА CISCO

Для виведення діагностичної інформації про фізичні параметри маршрутизатора чи його інтерфейсів, стан маршрутизатора, результати налагоджень або результати роботи маршрутизатора тощо використовується команда **show**. Вона є доступною як із режиму користувача, так і з привілейованого режиму. Залежно від режиму дана команда може мати різні параметри. Частина параметрів є однаковими і доступними в обох режимах. Часто команда **show** із певним параметром вважається окремою командою. Перелік основних команд **show** та їх призначення наведені у таблиці 4.4.

Таблиця 4.4 – Перелік основних параметрів команди **show**

Команда	Призначення
show version	Виведення поточної інформації про апаратне і програмне забезпечення
show flash	Перегляд вмісту флеш-пам'яті
show processes	Виведення інформації про процеси, запущені на пристрої
show interfaces	Виведення деталізованої інформації про інтерфейси маршрутизатора та їх стан
show ip interface	Виведення інформації про функціонування протоколу IP версії 4 та суміжних протоколів
show ip interface brief	Виведення інформації про функціонування протоколу IP версії 4 на інтерфейсі у скороченому вигляді
show ipv6 interface	Виведення інформації про функціонування протоколу IP версії 6 та суміжних протоколів
show ipv6 interface brief	Виведення інформації про функціонування протоколу IP версії 6 на інтерфейсі у скороченому вигляді
show protocols	Виведення глобальної та інтерфейсно-залежної інформації про протоколи 3-го рівня, що функціонують на маршрутизаторі
show startup-config	Перегляд стартової конфігурації пристрою
show running-config	Перегляд поточної конфігурації пристрою
show history	Перегляд списку останніх виконаних команд (за замовчуванням 10 рядків)
show clock	Виведення часу, встановленого на маршрутизаторі

В більшості випадків, наданих команд цілком досить для виявлення та виправлення переважного числа несправностей, які виникають на маршрутизаторах та їх інтерфейсах.

РОЗДІЛ 5

ПРАКТИКУМ

5.1 ПРАКТИЧНИЙ МОДУЛЬ 1

5.1.1 БАЗОВА НАВІГАЦІЯ В IOS. БАЗОВІ НАЛАШТУВАННЯ

КОМАНДИ БАЗОВОГО НАЛАГОДЖЕННЯ КЕРОВАНОГО КОМУТАТОРА CISCO

Конфігурування керованого комутатора *Cisco* передбачає налагодження: параметрів іменування, системного годинника, параметрів консольного підключення, параметрів термінального вікна, часових періодів (тайм-аутів) сеансу, системних повідомлень, безпечного доступу до пристрою, параметрів IP-адресації та багато ін.

Іменування пристроїв у *Cisco IOS* використовується:

- для ідентифікації пристрою під час підключення (як за консольного підключення, так і в разі мережних термінальних підключень за допомогою протоколів віддаленого доступу *Telnet* чи *SSH*);
- під час розсилки інформації про пристрій іншим пристроям (наприклад, за допомогою протоколів виявлення пристроїв *LLDP* чи *CDP*);
- для генерації ключів у разі використання криптографічних засобів (наприклад, у протоколі *SSH*).

Для зміни імені пристрою призначена команда **hostname**. Повернення імені пристрою за замовчуванням – **no hostname**. За замовчуванням заводське ім'я комутатора **Switch**.

Операційна система *Cisco IOS* на пристроях *Cisco* забезпечує функціонування системного (програмного) годинника/календаря. У деяких моделях пристроїв також наявний і апаратний годинник/календар. Відповідно існують механізми обміну даними між ними. Параметри системних часу (та дати) пристрою можуть встановлюватися як за допомогою команд локального застосування, так і з використанням мережного джерела часу. У першому випадку за умови відсутності апаратного годинника параметри системного часу не зберігаються у конфігураційному файлі і є актуальними лише на період роботи пристрою. Після перезавантаження їх необхідно встановлювати заново. У другому випадку системний час після завантаження пристрою синхронізується з часом сервера часу за протоколом *NTP*. Надалі операція синхронізації виконується періодично. Звичайно, це потребує певних специфічних налагоджень.

Для відображення конфігурації апаратного забезпечення системи, версії програмного забезпечення, назв і джерел конфігураційних файлів і завантажувальних образів можна скористатися командою **show version** в режимі *EXEC*.

Нижче наведено приклад результатів виконання команди **show version** на маршрутизаторі серії *Cisco 7000*:

```
Router> show version
GS Software (GS7), Version 10.0
Copyright (c) 1986-1993 by cisco Systems, Inc.
Compiled Mon 11-Jan-93 14:44
System Bootstrap, Version 4.6(1)
Current date and time is Fri 2-26-1993 2:18:52
Boot date and time is Fri 1-29-1993 11:42:38
Router uptime is 3 weeks, 6 days, 14 hours, 36 minutes
System restarted by power-on
Running default software
Network configuration file is "Router", booted via tftp from 172.16.2.333
RP1 (68040) processor with 16384K bytes of memory.
X.25 software.
Bridging software.
1 CIP controller (3 IBM Channels).
1 CIP2 controller (3 IBM Channels).
1 Switch Processor.
1 TRIP controller (4 Token Ring).
4 Token Ring/IEEE 802.5 interface.
1 AIP controller (1(ATM)
1 ATM network interface
4096K bytes of flash memory on embedded flash (in RP1).
Configuration register is 0x0
```

Встановлення системного часу здійснюється за допомогою команди **clock set**, встановлення часового поясу – за допомогою команди **clock timezone**. Для активації переходу на літній час застосовується команда **clock summertime**. Для виведення параметрів часу та дати апаратного годинника пристрою у ручному режимі застосовується команда **clock read-calendar**. Для налагодження використання апаратного годинника пристрою як авторитетного джерела мережного часу застосовується команда **clock calendar-valid**. Для одноразової ручної синхронізації параметрів часу апаратного годинника з параметрами часу програмного годинника пристрою застосовується команда **clock update-calendar**. Синтаксис розглянутих команд наведено нижче.

Синтаксис команди **hostname** (режим глобального конфігурування):

hostname *device-name*,

де **device-name** – текстове ім'я пристрою; теоретично може містити до 63 символів (літер, цифр, спец. символів), рекомендується задавати ім'я довжиною до 10 символів, оскільки в більшості систем існує обмеження на довжину службової частини командного рядка.

Синтаксис команди **clock set** (привілейований режим):

```
clock set hh:mm:ss dd month yyyy,
```

де **hh:mm:ss** – години (у 24-годинному форматі), хвилини, секунди;

dd – день, значення у діапазоні від 1 до 31;

month – місяць, назва місяця англійською мовою;

yyyy – рік, чотирицифрове значення в діапазоні від 1993 до 2035.

Синтаксис команди **clock timezone** (режим глобального конфігурування):

```
clock timezone time-zone hh[mm],
```

де **time-zone** – часовий пояс (текстове значення вигляду *WET* – *Western European Time*, *CET* – *Central European Time*, *EET* – *Eastern European Time*, *EEST* – *Eastern European Summer Time* і т.д.), за замовчуванням встановлено універсальний глобальний час (*UTC*, *Coordinated Universal Time*);

hh – години, зсув від *UTC*, ціле число в діапазоні від – 0 до 23;

mm – хвилини, зсув від *UTC*, ціле число в діапазоні від – 0 до 59.

Синтаксис команди **clock summertime** (режим глобального конфігурування):

```
clock summertime time-zone date [b_day, b_month, b_year,  
b_hh:mm e_day, e_month, e_year, e_hh:mm] [shift]
```

```
clock summertime time-zone recurring [b_week, b_day,  
b_month, b_hh:mm e_week, e_day, e_month, e_hh:mm]  
[shift],
```

де **time-zone** – часовий пояс;

date – службова конструкція, за допомогою якої зазначається початкова і кінцева дати літнього часу;

recurring – службова конструкція, яка зазначає, що перехід на літній час повинен здійснюватися щороку;

b_day, **e_day** – день початку і закінчення дії літнього часу, решта параметрів трактуються подібним чином;

shift – кількість хвилин, які необхідно додати у момент переходу на літній час, за замовчуванням – 60 хв.

Синтаксис команди **clock read-calendar** (режим глобального конфігурування).

Синтаксис команди **clock calendar-valid** (режим глобального конфігурування).

Синтаксис команди **clock update-calendar** (режим глобального конфігурування).

ОСНОВНІ КОМАНДИ НАЛАГОДЖЕННЯ КОНСОЛЬНОГО ПІДКЛЮЧЕННЯ ДО ПРИСТРОЇВ *CISCO*

Для консольного підключення (а також і для підключень по інших термінальних лініях) використовуються спеціальні програми-емулятори терміналу, які мають можливість працювати з послідовними портами комп'ютера. Це можуть бути як вбудовані в систему програмні продукти, так і розробки сторонніх виробників. Як приклади можна навести вбудовану в *ОС Windows* програму *HyperTerminal* та широкоживані відкриті кросплатформені розробки *PuTTY*, *SecureCRT*.

Для термінальної програми, за допомогою якої здійснюється консольне підключення до пристрою (комутатора чи маршрутизатора), можна налагодити такі параметри взаємодії, як:

- швидкість (приймання і передавання даних для лінії, біт/с);
- біти даних (кількість бітів даних на символ, яку розуміє і генерує апаратне забезпечення);
- парність (біт парності для асинхронної послідовної лінії зв'язку, фактично це сума бітів даних, яка показує, що дані містять або не містять парну чи непарну кількість одиничних бітів);
- стопові біти (стопові розряди, які передаються для кожного байта);
- керування потоком (керування потоком даних між пристроями, які підключені через послідовну лінію зв'язку).

Параметри за замовчуванням на прикладі програм *HyperTerminal* та *PuTTY* наведені на рисунку 5.1.

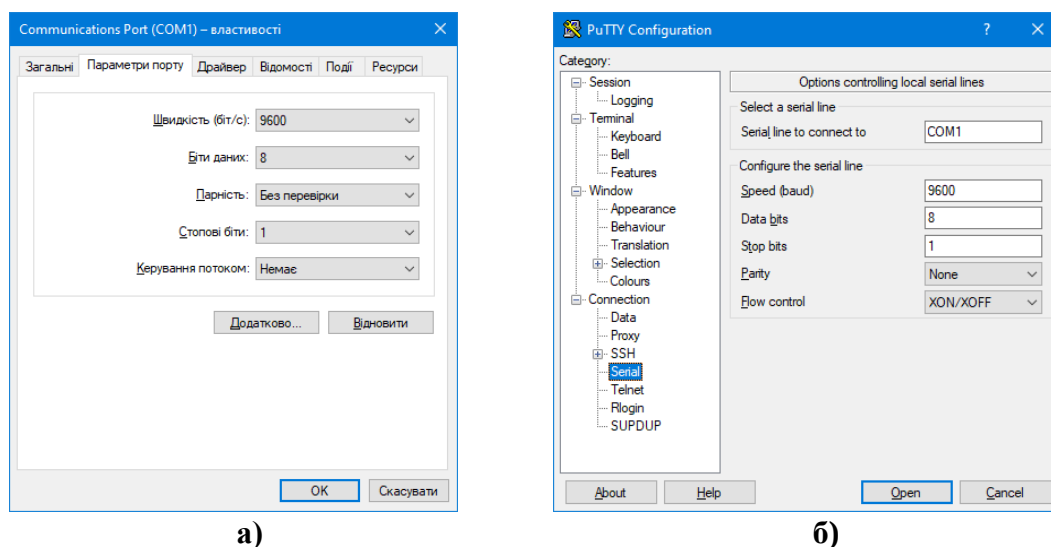


Рисунок 5.1 – Параметри за замовчуванням для консольного підключення за допомогою програм: а) *HyperTerminal*; б) *PuTTY*

Вибір лінії консольного підключення для налагодження здійснюється командою **line console 0** (режим глобального конфігурування). Для налагодження параметрів лінії на стороні комутатора (чи іншого пристрою Cisco)

використовуються команди **speed**, **databits**, **parity**, **stopbits**, **flowcontrol** відповідно. Повернення до стандартних значень параметрів здійснюється з використанням службового слова **no** з відповідною командою (наприклад, **no speed**). Також можна використати команду **default** (наприклад, **default speed**). Синтаксис указаних команд наведено нижче.

Синтаксис команди **speed** (режим конфігурування лінії):

speed *value*,

де **value** – значення швидкості у біт/с, число з діапазону 0...4294967295. Як правило, задається з набору стандартних значень 110, 300, 1200, 2400, 4800, 9600, 19200 і т.д. Верхня межа залежить від мікросхеми *UART*, на якій реалізовано послідовний порт консолі. За замовчуванням встановлюється швидкість 9600 біт/с.

Синтаксис команди **databits** (режим конфігурування лінії):

databits *value*,

де **value** – кількість бітів даних на символ, набуває значень 5, 6, 7, 8. За замовчуванням становить 8 бітів.

Синтаксис команди **parity** (режим конфігурування лінії):

parity *value*,

де **value** – параметр, який може набувати значень **even**, **mark**, **none**, **odd**, **space**; за замовчуванням значення не визначене;

none – біт парності відсутній і не передається;

even – біт парності дорівнює 0, якщо у переданому символі парна кількість одиничних бітів;

mark – біт парності завжди дорівнює 1;

odd – біт парності дорівнює 0, якщо у переданому символі непарна кількість одиничних бітів;

space – біт парності завжди дорівнює 0;

Синтаксис команди **stopbits** (режим конфігурування лінії):

stopbits *value*,

де **value** – параметр, який може набувати значень 1; 1.5; 2. За замовчуванням – 2.

Синтаксис команди **flowcontrol** (режим конфігурування лінії):

flowcontrol *value* [*lock*] [*in* | *out*],

де **value** – параметр, який може набувати значень **none**, **hardware**, **software**, за замовчуванням керування потоком даних відсутнє;

none – параметр вимикання режиму керування потоком даних;

hardware – параметр вмикання режиму апаратного керування потоком даних;

software – параметр вмикання режиму програмного керування потоком даних;

lock – службова конструкція, яка забороняє вимикання режиму керування потоком даних, застосовується лише для параметра **software**;

in – параметр, який вказує на встановлення контролю потоку на вхід лінії;

out – параметр, який вказує на встановлення контролю потоку на вихід лінії;

Якщо не вказаний жоден із параметрів **in** або **out**, то вважається, що контроль потоку здійснюється в обох напрямках.

Синтаксис команди **default** (режим конфігурування лінії):

default *value*,

де **value** – параметр, який може набувати значень **speed**, **databits**, **parity**, **stopbits**, **flowcontrol**, **history size**.

Для інших ліній можуть здійснюватися налагодження, подібні до тих, що здійснюються для консольної лінії.

Для зручності відображення інформації під час налагодження пристрою доцільно встановити параметри термінального вікна, у якому вводяться команди та виводяться їх результати. Правильний підбір параметрів допомагає розв'язати проблему занадто довгих рядків або їх великої кількості. Для налагодження ширини та висоти використовуються команди **width** та **length**. Повернення до стандартних розмірів здійснюється командами **no width** та **no length** відповідно.

Синтаксис команди **width** (режим конфігурування лінії):

width *columns*,

де **columns** – кількість стовпчиків вікна термінальної програми, за замовчуванням – 80.

Синтаксис команди **length** (режим конфігурування лінії):

length *lines*,

де **lines** – кількість рядків термінальної програми (може змінюватися в діапазоні від 0 до 512), за замовчуванням – 24.

У пристроях Cisco наявна можливість використовувати попередньо введені в сеансі роботи команди. Ця можливість називається „історія команд”. Її можна використовувати як для пристрою в цілому, так і для окремих ліній. Включення і відключення режиму історії команд у цілому для пристрою здійснюється командами **history**, **no history**. Для встановлення кількості команд, які вводилися останніми і зберігаються у пам'яті пристрою, використовується команда

history size. Налаштування даної команди зберігаються у конфігурації пристрою і застосовуються до всіх сеансів користувачів. Результати роботи команди переглядаються командою **show history**.

Синтаксис команди **history size** (режим конфігурування лінії):

history size *value*,

де **value** – кількість команд, про які повинен пам'ятати пристрій, може змінюватися в діапазоні від 0 до 255, за замовчуванням становить 10 команд.

Для поточного сеансу використовуються подібні команди: **terminal width**, **terminal length**, **terminal history**, **terminal history size**. Їх синтаксис аналогічний попередньо розглянутим командам.

Для поточного сеансу зв'язку по лінії існує можливість встановити певні часові періоди (тайм-аути) та режими його роботи, наприклад, інтервал часу, протягом якого сеанс може залишатися відкритим, інтервал часу, протягом якого сеанс зв'язку може бути неактивним, активацію виведення повідомлення у разі виходу із системи тощо. З цією метою використовуються команди **absolute-timeout**, **session-timeout**, **exec-timeout**, **logout-warning**, **logging synchronous** тощо.

Команда **absolute-timeout** встановлює чіткий інтервал часу до того моменту, коли сеанс буде закрито. На відміну від інших періодів, цей інтервал не залежить від періоду простою, тобто сеанс буде закрито через зазначений час, незалежно від того, активно використовується сеанс чи ні. Відміна дії команди **no absolute-timeout** або **absolute-timeout 0**. Команда **session-timeout** встановлює інтервал часу, протягом якого пристрій очікує передавання даних перед тим, як закрити сеанс, тобто інтервал простою для лінії. Відміна дії команди **no session-timeout** або **session-timeout 0**. Команда **exec-timeout** встановлює інтервал часу, протягом якого пристрій очікує введення даних в активному сеансі привілейованого режиму. Після закінчення даного інтервалу здійснюється перехід у попередній режим. Відміна дії команди **no exec-timeout** або **exec-timeout 0**.

Команда **logout-warning** активізує виведення попередження у разі виходу із системи. Це попередження інформує користувача, що найближчим часом відбудеться примусовий вихід із сеансу. Відміна дії команди **no logout-warning**. Команда **logging synchronous** керує виведенням журнальних повідомлень на термінал користувача. За замовчуванням повідомлення можуть виводитися у будь-який момент, часто перериваючи виконання поточної команди користувача. За допомогою команди **logging synchronous** можна примусити пристрій очікувати завершення поточної команди і виведення її результатів і лише після цього відображати журнальні повідомлення. Відміна дії команди **no logging synchronous**.

Синтаксис команди **absolute-timeout** (режим конфігурування лінії):

absolute-timeout *minutes*,

де **minutes** – тривалість періоду, протягом якого сеанс може залишатися відкритим, зазначається у хвилинах, може змінюватися у діапазоні від 0 до 10000; за замовчуванням значення дорівнює 0, тобто не визначене.

Синтаксис команди **session-timeout** (режим конфігурування лінії):

session-timeout *minutes* [*output*],

де **minutes** – тривалість періоду до моменту, як сеанс буде припинено за тайм-аутом; зазначається у хвилинах, може змінюватися у діапазоні від 0 до 35791; за замовчуванням дорівнює 0, тобто не визначено;

output – службова конструкція, яка примушує пристрій враховувати у разі обнулення лічильників як вхідний, так і вихідний трафік, якщо вона відсутня, то лише вхідний трафік викликає обнулення лічильника.

Синтаксис команди **exec-timeout** (режим конфігурування лінії):

exec-timeout *minutes*,

де **minutes** – тривалість періоду, протягом якого сеанс зв'язку може бути неактивним; зазначається у хвилинах, може змінюватися у діапазоні від 0 до 35791, за замовчуванням становить 10 хвилин; не рекомендується налаштовувати цей період занадто коротким, оскільки існує ймовірність втрати можливості контролю над пристроєм.

Синтаксис команди **logout-warning** (режим конфігурування лінії):

logout-warning *seconds*,

де **seconds** – тривалість періоду до завершення сеансу, може змінюватися в діапазоні від 0 до 4294967295; за замовчуванням значення не визначене.

Синтаксис команди **logging synchronous** (режим конфігурування лінії):

logging synchronous [*level importance* | *all*] [*limit number_of_messages*],

де **level** – службова конструкція, яка вказує на зміну рівня важливості команди;

importance – значення рівня важливості, змінюється від 0 до 7, за замовчуванням у разі активації команди без параметрів дорівнює 2; всі повідомлення даного і більш низького рівнів (із номерами більше даного) відправляються синхронно (тобто після того, як користувач завершить поточну команду, а пристрій виведе результат);

all – параметр, який вказує, що всі повідомлення відправляються синхронно;

limit – службова конструкція, яка вказує, що використовуються обмеження щодо кількості повідомлень;

number_of_messages – кількість повідомлень, які можуть бути розміщені в черзі доставки, максимальна – 20.

СИСТЕМНІ ПОВІДОМЛЕННЯ ПРИСТРОЇВ CISCO

У пристроях Cisco передбачено кілька стандартних системних повідомлень (банерів) для користувачів. Як правило, ці повідомлення асоціюються з процесом входу в систему або виходу з неї. Виведення того чи іншого повідомлення (чи кількох по чергові) залежить від налагоджень пристрою. У пристроях Cisco використовуються такі повідомлення:

1. Повідомлення дня (*Motd Banner, Message of the Day*).
2. Повідомлення входу в систему (*Login Banner*).
3. Повідомлення виконання (*Exec Banner*).
4. Повідомлення вхідного термінального з'єднання (*Incoming Banner*).
5. Повідомлення тайм-ауту входу в систему (*Prompt-Timeout Banner*).
6. Повідомлення для протоколів SLIP/PPP.

Приклад використання кількох повідомлень-банерів може бути таким. При вході користувача виводиться повідомлення дня, після нього йде повідомлення входу в систему, а далі саме запрошення входу. Після успішної реєстрації на екрані виводиться повідомлення виконання. Для встановлення повідомлень-банерів використовується команда **banner**. Відключення виведення повідомлення здійснюється командою **no banner**.

Синтаксис команди **banner** (режим глобального конфігурування):

banner btype # banner-text #,

де **btype** вказує тип банера і може набувати значень:

LINE – текстове повідомлення;

exec – повідомлення виконання;

incoming – повідомлення вхідного термінального з'єднання;

login – повідомлення входу в систему;

motd – повідомлення дня;

prompt-timeout – повідомлення тайм-ауту входу в систему;

slip-ppp – повідомлення для протоколів SLIP/PPP;

banner-text # – текст повідомлення, **#** – знак початку і кінця повідомлення, замість цього знака можуть використовуватися будь-які символи, які не зустрічаються у тексті повідомлення.

Як правило, якщо повідомлення-банери встановлені, то вони виводяться на екран. Відключити виведення більшості повідомлень-банерів неможливо, їх можна лише видалити. Повідомлення виконання та повідомлення дня можна відключити чи включити, сториставши команди **no exec-banner**, **no motd-banner** та **exec-banner**, **motd-banner**. Відключення повідомлення

виконання призводить до відключення повідомлення дня, а відключення повідомлення дня не впливає на повідомлення виконання.

ТИПИ ПАРОЛІВ *Cisco IOS*

На пристроях *Cisco* існує можливість налагодити безпечний доступ для всіх видів підключень та певних режимів *Cisco IOS* із використанням парольного захисту. Для забезпечення парольного захисту на пристроях *Cisco* передбачено такі паролі:

1. Пароль ліній (пароль для входу в режим користувача).
2. Пароль входу у привілейований режим.
3. Паролі користувачів.

Перші два паролі встановлюються на пристрій у цілому й обмежують вхід до відповідних режимів. Паролі для користувачів можуть мати різний рівень привілеїв щодо виконання команд. Інформація про паролі та користувачів зберігається у конфігураційному файлі пристрою.

Для пристроїв *Cisco* використовуються три типи паролів:

1. Звичайний пароль (*Plain-text password*).
2. Пароль типу 7 (*Type 7 password*).
3. Пароль типу 5 (*MD5 hash password*).

Звичайні паролі встановлюються за замовчуванням і зберігаються у конфігураційному файлі пристрою у відкритому вигляді, що є загрозою безпеці. Паролі типу 7 для підвищення рівня безпеки використовують шифрування за алгоритмом Віженера (*Vigenere*). Паролі даного типу доволі легко розшифровуються, тому рекомендується використовувати паролі типу 5, які мають найвищий рівень безпеки.

НАЛАГОДЖЕННЯ ПАРОЛЬНОГО ДОСТУПУ НА ПРИСТРОЯХ *Cisco*

Для налагодження доступу по лініях до пристрою *Cisco* використовуються команди **password** та **login**. Застосування цих команд передбачає те, що паролі є звичайними (відкритими). Для налагодження парольного доступу до привілейованого режиму у пристроях *Cisco* передбачено команду **enable password**. Якщо цю команду використати без параметрів, то пароль буде теж звичайним відкритим. Існує можливість використання цієї команди із встановленням шифрованого пароля типу 7. Для шифрування всіх паролів відразу (встановлення паролів типу 7) використовується команда **service password-encryption**. Оскільки пароль даного типу вважається слабким, використовувати дану команду не рекомендується. Замість неї рекомендується використовувати команду **enable secret**, яка активує використання шифрованих паролів типу 5. Існує можливість створення окремих користувачів із різними привілеями входу в різні режими на пристроях *Cisco*. Для цього використовується команда

username. Відміна дії всіх розглянутих команд здійснюється за допомогою службової конструкції **no**.

Синтаксис команди **password** (режим конфігурування лінії):

```
password password-string,
```

де ***password-string*** – текстовий рядок пароля довжиною до 80 символів, який повинен починатися з літери.

Синтаксис команди **login** (режим конфігурування лінії):

```
login {local},
```

де ***local*** – службова конструкція, яка вказує, що для входу необхідно використовувати імена створених користувачів та їх паролі.

Синтаксис команди **enable password** (режим глобального конфігурування):

```
enable password [level level-value] {password-string |  
[encryption-type] encrypted-password-string},
```

де ***level*** – службова конструкція, яка зазначає рівень привілеїв пароля.

level-value – значення рівня, число в межах від 0 до 15;

password-string – текстовий рядок пароля;

encryption-type – тип шифрування;

encrypted-password-string – зашифрований пароль, отриманий з іншого джерела шифрування.

Синтаксис команди **enable secret** (режим глобального конфігурування):

```
enable secret [level level-value] {password-string |  
[encryption-type] encrypted-password-string }.
```

Параметри команди аналогічні параметрам попередньої команди.

Синтаксис команди **username** (режим глобального конфігурування):

```
username name {nopassword | password password-string |  
password encryption-type encrypted-password-string},
```

де ***name*** – текстове ім'я користувача;

nopassword – службова конструкція, яка вказує на те, що не потрібно використовувати пароль;

password – службова конструкція, яка вказує на використання пароля;

password-string – текстовий рядок пароля;

encryption-type – тип шифрування.

encrypted-password-string – зашифрований пароль, отриманий з іншого джерела шифрування.

Основні команди діагностики параметрів роботи комутатора Cisco.

Для виведення діагностичної інформації про фізичні параметри комутатора, результати налагоджень, результати роботи комутатора, стан комутатора тощо використовується команда **show**. Вона є доступною як із режиму користувача, так і з привілейованого режиму. Залежно від режиму дана команда може мати різні параметри. Частина параметрів є однаковими і доступними в обох режимах. Часто команда **show** із певним параметром вважається окремою командою. Перелік основних команд **show** та їх призначення наведені у таблиці 5.1.

Таблиця 5.1 – Перелік основних команд **show**

Команда	Призначення
show version	Виведення технічної інформації про пристрій
show tech-support	Виведення розширеної технічної інформації про пристрій
show flash	Виведення вмісту Flash-пам'яті
show boot	Виведення параметрів завантаження
show processes	Виведення інформації про стан процесів, що запущені в системі
show terminal	Виведення параметрів роботи термінала
show clock	Виведення параметрів системного часу
show history	Виведення історії команд
show running-config	Виведення поточної конфігурації комутатора
show startup-config	Виведення стартової конфігурації комутатора

ОСНОВНІ КОМАНДИ РОБОТИ З КОНФІГУРАЦІЯМИ

Важливим питанням налагодження комутатора Cisco є перегляд та збереження його конфігурацій. Як уже зазначалося, для перегляду стартової конфігурації використовується команда **show startup-config**, а для перегляду поточної конфігурації – команда **show running-config**. Обов'язковим є збереження налагоджень поточної конфігурації у стартовій. Для цього використовується команда **copy running-config startup-config**. Існує можливість копіювання конфігурації на зовнішні сервери або із зовнішніх серверів. Це можуть бути традиційні *FTP/TFTP* або менш уживані *SCP* чи *RCP*-сервери.

Синтаксис команди **copy** (привілейований режим):

copy *source destination*,

де **source**, **destination** – джерело та приймач даних відповідно.

Основні комбінації параметрів команди **copy** наведені у таблиці 5.1.2.

Для видалення стартової конфігурації використовується команда **erase startup-config**. Для перезавантаження пристрою використовується команда **reload**. Окрім зазначених команд для роботи з образами ОС та конфігураціями застосовуються команди **dir**, **more** та інші.

Таблиця 5.2 – Можливі комбінації параметрів команди **copy**

Джерело	Отримувач				
	running-config	startup-config	flash	ftp:	tftp:
running-config	–	+	+	+	+
startup-config	+	–	+	+	+
flash	+	+	–	+	+
ftp:	+	+	+	–	–
tftp:	+	+	+	–	–

Також для роботи з поточною конфігурацією може використовуватися команда **write**. Вона вважається застарілою, і замість неї рекомендується використовувати команди **copy**, **show**, **erase**.

Синтаксис команди **write** (привілейований режим):

write { erase | memory | network | terminal },

де **erase** – параметр, який указує, що необхідно видалити стартову конфігурацію пристрою;

memory – параметр, який указує, що необхідно зберегти поточну конфігурацію пристрою;

network – параметр, який указує, що необхідно зберегти поточну конфігурацію на *TFTP*-сервері;

terminal – параметр, який указує, що необхідно вивести поточну конфігурацію на екран.

Відповідності застарілих команд **write** сучасним командам наведені у таблиці 5.3.

Таблиця 5.3 – Команди **write** та їх сучасні еквіваленти

Команда write	Сучасний еквівалент
write erase	erase startup-config
write memory	copy running-config startup-config
write network	copy running-config tftp
write terminal	show running-config

МОДЕЛЬНІ ПРИКЛАДИ БАЗОВОГО НАЛАГОДЖЕННЯ КЕРОВАНОВОГО КОМУТАТОРА *CISCO*

Розглянемо специфіку базового налагодження параметрів функціонування комутатора *Cisco* моделі *WS-C2960-24TT-L* за допомогою консольного підключення (рисунок 5.2, а) із використанням *Cisco Console Cable* (*Cisco USB Console Cable*) та термінального додатка *HyperTerminal*. Слід зазначити, що параметри консольного підключення у разі першого підключення до нового пристрою *Cisco* (пристрою без конфігурації) зазначаються за замовчуванням. Результат такого підключення до нового комутатора *Cisco* (комутатора без конфігурації) наведено на рисунку 5.2, б.

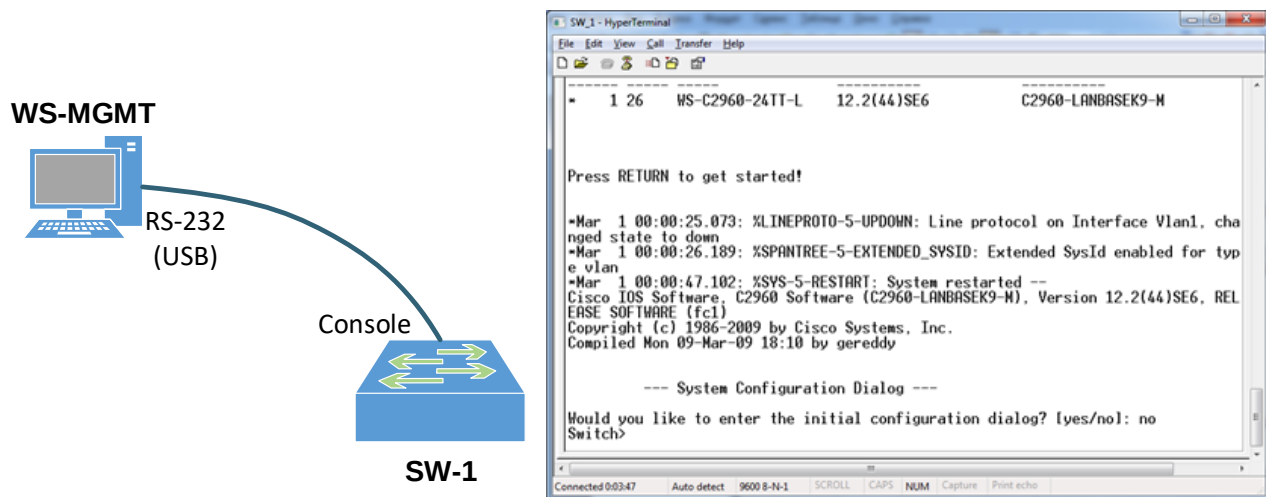


Рисунок 5.2 – Приклад консольного підключення з результатами

Для налагодження основних параметрів комутатора та параметрів консольного підключення до комутатора використано дані таблиці 5.4.

Таблиця 5.4 – Параметри налагодження комутатора для прикладу

Параметр	Значення
Основні параметри	
Назва комутатора	SW-1
Повідомлення дня	1 рядок, текст – Switch SW-1 / Location – 101v
Параметри часу	
Системний час	Поточний (включає поточні час та дату)
Часовий пояс	Східноєвропейський (EET, Eastern European Time)
Перехід на літній час	Україна
Параметри консольного підключення	
Швидкість, біт/с	19200
Біти даних	7
Парність	Парні
Стопові біти	2
Керування потоком	Апаратне
Параметри термінального сеансу	
Історія команд, шт.	100
Тайм-аут виходу, с	30
Виведення журнальних повідомлень	Так

Сценарій налагодження основних параметрів комутатора (імені, повідомлення дня) наведений нижче. У цьому сценарії виконано збереження конфігурації та перезавантаження.

```

Switch>enable
Switch#configure terminal
Switch(config)#hostname SW-1
SW-1(config)#banner motd #Switch SW-1 / Location - 101v#
SW-1(config)#exit
SW-1#copy running-config startup-config
SW-1#reload

```

Результат виконання даного сценарію у разі повторного підключення до комутатора наведений на рисунку 5.3.

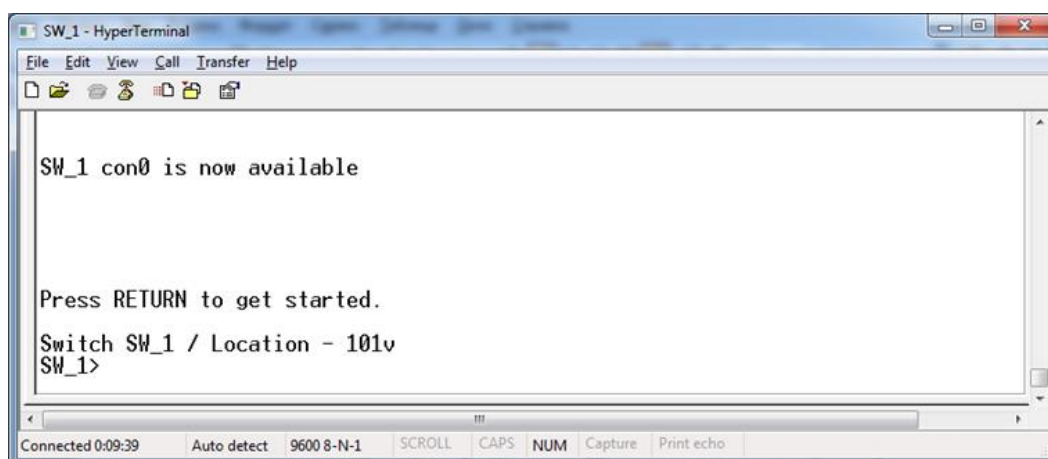


Рисунок 5.3 – Результат підключення до комутатора SW-1 після базового налагодження

Сценарій налагодження часових параметрів комутатора (часового поясу та поточного часу) наведений нижче. Слід зазначити, якщо має місце перезавантаження комутатора, то системний час потрібно встановити повторно, оскільки для моделі комутатора *WS-C2960-24TT-L* він не зберігається в апаратному годиннику.

```

SW-1>enable
SW-1#configure terminal
SW-1(config)#clock timezone EET 2
SW-1(config)#exit
SW-1#clock set 8:00:00 01 may 2016
SW-1#copy running-config startup-config
SW-1#exit

```

Сценарій налагодження параметрів консольного підключення (лінії) до комутатора та термінального сеансу наведений нижче.

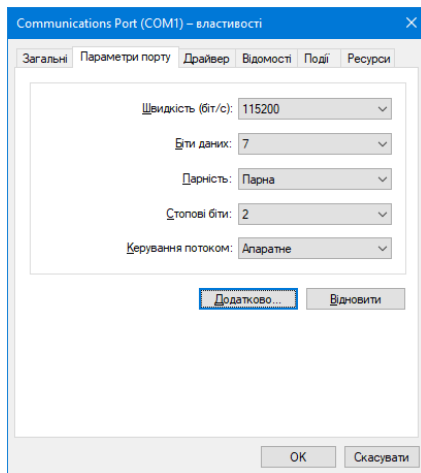
```

SW-1>enable
SW-1#configure terminal
SW-1(config)#line console 0
SW-1(config-line)#speed 19200
SW-1(config-line)#databits 7
SW-1(config-line)#parity even
SW-1(config-line)#stopbits 2
SW-1(config-line)#flowcontrol hardware
SW-1(config-line)#logging synchronous
SW-1(config-line)#history size 100

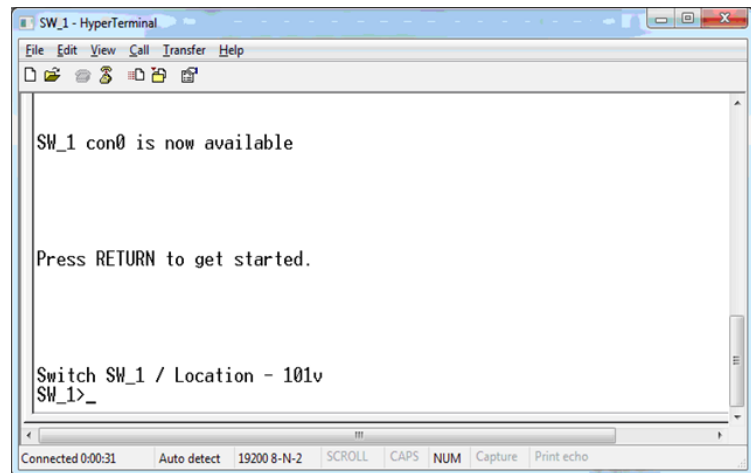
```

```
SW-1(config-line)#exec-timeout 30
SW-1(config-line)#end
SW-1#copy running-config startup-config
SW-1#exit
```

Для перевірки налагоджених за даним сценарієм параметрів консольного підключення необхідно перезапустити термінальний додаток, у якому встановити необхідні значення параметрів підключення. Результати налагодження параметрів для *Windows*-дodatка *HyperTerminal* та результат виконання підключення наведено відповідно на рисунку 5.4, а, б.



а)



б)

Рисунок 5.4 – Результат налагодження:

- а – параметрів підключення для *Windows*-дodatка *HyperTerminal*;
- б – консольного підключення до комутатора *SW-1* та термінального сеансу

Сценарій налагодження доступу до комутатора з використанням механізму паролів наведено нижче (встановлюється пароль на вхід для консольного підключення та пароль на перехід до привілейованого режиму). Паролі зберігаються у файлі конфігурації у відкритому вигляді.

```
SW-1>enable
SW-1#configure terminal
SW-1(config)#line console 0
SW-1(config-line)#password mypass1
SW-1(config-line)#login
SW-1(config-line)#exit
SW-1(config)#enable password mypass2
SW-1(config)#exit
SW-1#exit
```

Файл конфігурації, що створений як результат виконання попередніх трьох сценаріїв, і виведений за допомогою команди **show running-config** наведений далі (частину файла, яка містить несуттєву інформацію, вилучено; жирним шрифтом виділено додані за останнім сценарієм рядки конфігурації). Файл конфігурації комутатора *SW-1* для модельного прикладу з використанням відкритих паролів:

```

Building configuration...
Current configuration : 1288 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname SW-1
!
enable password mypass2
!
clock timezone EET 2
!
spanning-tree mode pvst
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
...
!
interface FastEthernet0/23
!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
no ip address
shutdown
!
banner motd ^C Switch SW-1
Location - 101v^C
!
!
!
line con 0
  password mypass1
  logging synchronous
  login
  history size 100
  exec-timeout 30 0
  speed 19200
  databits 7
  parity even
  stopbits 2
  flowcontrol hardware
!
line vty 0 4
  login
line vty 5 15
  login
!
!
end

```

Сценарій налагодження доступу до комутатора з використанням механізму користувачів та паролів типу 7 наведено нижче (; встановлюється пароль на вхід у привілейований режим; створюються користувач *User1* із рівнем привілеїв 1 (за замовчуванням) та користувач *Admin* із максимальним рівнем привілеїв 15; відключається пароль на вхід для консольного підключення; активується застосування механізму користувачів для консольного підключення; здійснюється операція шифрування звичайних відкритих паролів із метою отримання паролів типу 7).

```
SW-1#configure terminal
SW-1(config)#enable password mypass2
SW-1(config)#username User1 password mypass3
SW-1(config)#username Admin privilege 15 password mypass4
SW-1(config)#line console 0
SW-1(config-line)#no password
SW-1(config-line)#login local
SW-1(config-line)#exit
SW-1(config)#service password-encryption
SW-1(config)#exit
SW-1#
```

Файл конфігурації, що створений як результат виконання цього сценарію, наведено далі (частину файла, яка містить несуттєву інформацію, вилучено; жирним шрифтом виділено модифіковані та додані рядки конфігурації). Файл конфігурації комутатора *SW-1* для модельного прикладу з використанням механізму користувачів та шифрованих паролів типу 7:

```
Building configuration...
Current configuration : 1442 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname SW-1
!
enable password 082C555E080A1645
!
clock timezone EET 2
!
username Admin privilege 15 password 7 082C555E080A1643
username User1 privilege 1 password 7 082C555E080A1644
!
spanning-tree mode pvst
!
interface FastEthernet0/1
!
interface FastEthernet0/3
!
...
```

```

!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
no ip address
shutdown
!
banner motd ^CSwitch SW-1
Location - 101v^C
!
!
!
line con 0
  logging synchronous
  login local
  history size 100
  exec-timeout 30 0
  speed 19200
  databits 7
  parity even
  stopbits 2
  flowcontrol hardware
!
line vty 0 4
  login
line vty 5 15
  login
!
!
end

```

Для підвищення рівня безпеки комутатора рекомендується замість паролів типу 7 застосовувати паролі типу 5. У такому разі краще застосовувати наведений нижче модифікований сценарій налагодження доступу до комутатора.

```

SW-1#configure terminal
SW-1(config)#no enable password
SW-1(config)#enable secret mypass2
SW-1(config)#no username User1
SW-1(config)#no username Admin
SW-1(config)#username User1 secret mypass3
SW-1(config)#username Admin privilege 15 secret mypass4
SW-1(config)#line console 0
SW-1(config-line)#login local
SW-1(config-line)#exit
SW-1(config)#

```

РЕЗУЛЬТАТИ ВИКОНАННЯ КОМАНД МОНІТОРИНГУ ТА ДІАГНОСТИКИ РОБОТИ КОМУТАТОРА ДЛЯ РОЗГЛЯНУТОГО МОДЕЛЬНОГО ПРИКЛАДУ

З метою перегляду інформації про налагоджені параметри системного часу для розглянутого прикладу виконано команду **show clock**. Із метою визначення параметрів консольного підключення та термінального вікна виконано команду **show terminal**. Із метою визначення імені користувача, що здійснив підключення до пристрою, та рівня його привілеїв виконано команди **show users** та **show privilege**. Результати роботи зазначених команд наведено відповідно в наступних блоках.

Результати виконання команди **show clock** на комутаторі SW-1:

```
SW-1#show clock
11:17:37.780 EET Sun May 1 2016
SW-1#
```

Результати виконання команди **show terminal** на комутаторі SW-1:

```
SW-1#show terminal
Line 0, Location: , Type:
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 19200/19200, even parity, 2 stopbits, 7 databits
Status: PSI Enabled, Ready, Active, Automore On
Capabilities: none
Modem state: Ready
Modem hardware state: CTS* noDSR DTR RTS
Special Chars: Escape Hold Stop Start Disconnect Activation
                ^^x   none - - none
Timeouts:      Idle EXEC Idle Session Modem Answer Session Dispatch
                00:10:00 never none not set
                Idle Session Disconnect Warning
                never
                Login-sequence User Response
                00:00:30
                Autoselect Initial Wait
                not set

Modem type is unknown.
Session limit is not set.
Time since activation: 00:03:04
Editing is enabled.
History is enabled, history size is 100.
DNS resolution in show commands is enabled
Full user help is disabled
Allowed input transports are All.
Allowed output transports are pad telnet rlogin.
Preferred transport is telnet.
No output characters are padded
No special data dispatching characters
SW-1#
```

Результати виконання команди **show users** та **show privilege** на комутаторі SW-1:

```
SW-1#show users
      Line      User      Host(s)      Idle      Location
*  0 con 0      Admin      idle         00:00:00
      Interface  User      Mode         Idle      Peer Address
SW-1#show privilege
Current privilege level is 15
SW-1#
```

ПРАКТИЧНЕ ЗАВДАННЯ

Мета роботи: ознайомитися із загальною будовою керованого комутатора *Cisco*; ознайомитися з основними можливостями мережної операційної системи *Cisco IOS* та розглянути особливості її застосування на керованих комутаторах *Cisco*; дослідити можливості *Cisco IOS* з налагодження та діагностування основних параметрів функціонування керованих комутаторів *Cisco*.

Порядок виконання роботи

1. Розглянути та скласти повну і спрощену схеми нуль-модемного кабелю, побудованого з використанням двох рознімів *DB-9*. На схемах зазначити відповідні сигнали для відповідних контактів.

2. На основі схем з'єднань п. 1 та відповідних таблиць сигналів, наведених у теоретичних відомостях, скласти повну і спрощену схеми нуль-модемного кабелю, побудованого з використанням двох рознімів *DB-25*.

3. На основі схем з'єднань п. 1 та відповідних таблиць сигналів, наведених у теоретичних відомостях, скласти повну і спрощену схеми нуль-модемного кабелю, побудованого з використанням рознімів *DB-9* та *DB-25*.

4. На основі схем з'єднань п. 1 та відповідних таблиць сигналів, наведених у теоретичних відомостях, скласти повну і спрощену схеми кабелю *Cisco Rollover Cable*, побудованого з використанням двох рознімів *RJ-45*.

5. На основі схем з'єднань п. 1 та відповідних таблиць сигналів, наведених у теоретичних відомостях, скласти повну і спрощену схеми кабелю *Cisco Console Cable*, побудованого з використанням рознімів *DB-9* та *RJ-45*. На схемах зазначити відповідні сигнали для відповідних контактів.

6. На основі схем з'єднань п. 1 та відповідних таблиць сигналів, наведених у теоретичних відомостях, скласти повну і спрощену схеми кабелю *Cisco Console Cable*, побудованого з використанням рознімів *DB-25* та *RJ-45*.

7. Отримати необхідне обладнання для створення проекту мережі, у якому здійснити фізичне підключення робочої станції до комутатора за допомогою консольного кабелю (рисунок 5.5). Виконати підключення з робочої станції до комутатора за допомогою термінальної програми. Визначити основні параметри комутатора та занотувати їх у вигляді таблиці 5.5.

8. Провести налагодження параметрів консольного підключення та термінального сеансу (за даними таблиці 5.7). Зберегти налагодження. Перезавантажити комутатор та перевірити можливість підключення за допомогою

термінальної програми з налагодженими параметрами, вивести параметри налагоджень поточного термінального сеансу.

9. Провести налагодження параметрів іменування, системного часу (за даними таблиці 5.6), системних повідомлень-банерів. Зберегти налагодження. Перевірити відображення банера при вході в систему.

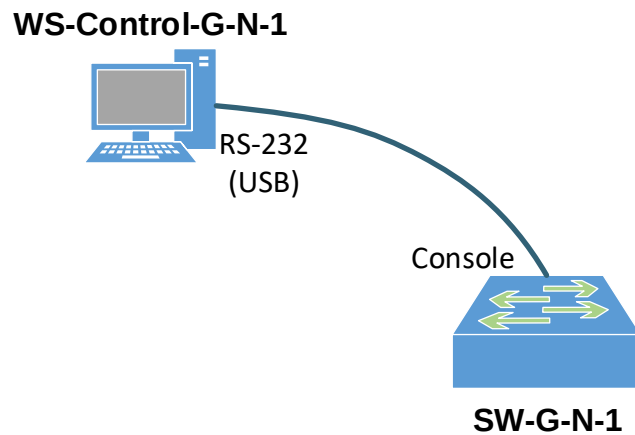


Рисунок 5.5 – Схема підключення

10. Провести налагодження парольного доступу до комутатора (його режимів користувача та привілейованого режиму) із використанням відкритих паролів. Зберегти налагодження. Перезавантажити комутатор та перевірити виконані налагодження. Зашифрувати паролі за типом 7 та перевірити результати шифрування.

Таблиця 5.5 – Параметри комутатора

Параметр	Значення
Модель комутатора	
Модель та номер процесора	
Об'єм пам'яті (RAM, Flash, NVRAM)	
Кількість інтерфейсів Ethernet/Fast Ethernet	
Кількість інтерфейсів Gigabit Ethernet	
Серійний номер системи	
Серійний номер материнської плати	
Серійний номер блока живлення	
Базова MAC адреса блока управління	
Конфігураційний регістр	
Версія IOS	
Образ IOS	
Розмір файлу образу IOS	

11. Провести налагодження доступу до комутатора з використанням механізму користувачів. Для цього створити трьох користувачів (два користувачі з мінімальним рівнем привілеїв 0 – *Technic-G-N-X*, один – із максимальним рівнем

привілеїв 15 – *Admin-G-N-1*). Зберегти налагодження. Перезавантажити комутатор та перевірити виконані налагодження. Дослідити відмітності у можливостях для користувачів із різними рівнями привілеїв.

12. Вивести, проаналізувати та скопіювати в звіт файл конфігурації комутатора.

Таблиця 5.6 – Параметри налагодження системного часу

№ варіанта	Часовий пояс	Години	Хвилини	Перехід на літній час
1	GMT	0	00	+
2	BST	+1	00	+
3	IST	+1	00	+
4	WET	0	00	+
5	WEST	+1	00	+
6	CET	+1	00	+
7	CEST	+2	00	+
8	EET	+2	00	+
9	EEST	+3	00	+
10	MSK	+4	00	+
11	GMT	0	30	+
12	BST	+1	30	+
13	IST	+1	30	+
14	WET	0	30	+
15	WEST	+1	30	+
16	CET	+1	30	+
17	CEST	+2	30	+
18	EET	+2	30	+
19	EEST	+3	30	+
20	MSK	+4	30	+
21	GMT	0	00	+
22	BST	+1	00	+
23	IST	+1	00	+
24	WET	0	00	+
25	WEST	+1	00	+
26	CET	+1	30	+
27	CEST	+2	30	+
28	EET	+2	30	+
29	EEST	+3	30	+
30	MSK	+4	30	+

Примітка: GMT, Greenwich Mean Time (UTC); BST, British Summer Time (UTC + 1 год.); IST, Irish Standard Time (UTC + 1 год.); WET, Western European Time (UTC); WEST, Western European Summer Time (UTC + 1 год.); CET, Central European Time (UTC + 1 год.); CEST, Central European Summer Time (UTC + 2 год.); EET, Eastern European Time (UTC + 2 год.); EEST, Eastern European Summer Time (UTC + 3 год.); MSK, Moscow Standard Time (UTC + 4 год.).

Таблиця 5.7 – Параметри налагодження консольного підключення та сеансу

№ варіанта	Speed, біт/с	Databits	Parity	Stopbits	Flow-control	History size	Width, стовпчиків	Length, рядків	Exec-timeout, XB	Logout- warning, с	Logging synchronous
1	1200	7	even	1	None	20	80	24	20	20	+
2	115200	8	odd	2	Software	50	75	24	60	30	+
3	2400	7	none	1	Hardware	25	70	24	30	40	+
4	115200	8	mark	2	None	45	65	24	40	40	+
5	4800	7	space	1	Hardware	30	60	24	50	30	+
6	57600	8	mark	2	Software	35	80	22	60	30	+
7	9600	7	none	1	Software	15	75	22	15	20	+
8	38400	8	odd	2	None	65	70	22	55	20	+
9	19200	7	even	1	Hardware	20	65	22	20	20	+
10	2400	8	odd	2	Software	60	60	22	50	30	+
11	57600	7	mark	1	Hardware	25	80	20	25	20	+
12	1200	8	even	2	None	55	75	20	45	30	+
13	115200	7	none	1	Hardware	30	70	20	30	30	+
14	9600	8	space	2	Software	50	65	20	40	30	+
15	19200	7	even	1	None	35	60	20	35	30	+
16	4800	8	none	2	Hardware	45	60	24	50	40	+
17	38400	7	space	1	None	40	65	24	10	20	+
18	4800	8	odd	2	Software	10	70	24	45	30	+
19	38400	7	mark	1	None	50	75	24	15	20	+
20	1200	8	even	2	Software	15	80	24	40	20	+
21	115200	7	odd	1	Hardware	45	60	22	20	20	+
22	9600	8	mark	2	None	20	65	22	35	30	+
23	19200	7	space	1	Hardware	40	70	22	25	20	+
24	2400	8	none	2	Software	25	75	22	30	30	+
25	57600	7	even	1	Software	35	80	22	60	60	+
26	19200	8	mark	2	None	30	60	20	35	30	+
27	57600	7	odd	1	Hardware	60	65	20	55	60	+
28	9600	8	none	2	Software	80	70	20	40	40	+
29	38400	7	space	1	Hardware	65	75	20	50	60	+
30	115200	8	none	2	None	75	80	20	45	40	+

Примітки: 1) Параметр *Software*, який налаштовується на пристроях *Cisco*, відповідає параметру *Xon/Xoff* термінальних програм; 2) Налаштовані значення, які збігаються із тими, що передбачені за замовчуванням, у конфігураційному файлі не відображаються.

КОНТРОЛЬНІ ПИТАННЯ

1. Типова структурна схема комутатора *Ethernet* фірми *Cisco*.
2. Кабельні підключення, що застосовуються з метою налагодження та керування комутатором *Cisco*.

3. Наведіть повну і спрощену схеми кабелю *Cisco Rollover Cable*.
4. Наведіть повну і спрощену схеми кабелю *Cisco Console Cable*.
5. Програмне забезпечення сучасних комутаторів *Ethernet*.
6. Загальна характеристика *Cisco IOS*. Платформи, набори можливостей та версії *Cisco IOS*, які використовуються для комутаторів *Cisco*.
7. Командні режими *Cisco IOS* для комутаторів *Cisco*. Команди переходів між командними режимами *Cisco IOS* для комутаторів.
8. Особливості отримання довідкової інформації у командному рядку *Cisco IOS*.
9. Наведіть перелік та поясніть призначення основних команд налагодження системного часу в комутаторах *Cisco*.
10. Наведіть перелік та поясніть призначення команд іменування пристроїв та створення системних повідомлень для пристроїв *Cisco*.
11. Наведіть перелік та поясніть призначення основних команд налагодження консольного підключення для пристроїв *Cisco*.
12. Наведіть перелік та поясніть призначення основних команд налагодження часових тайм-аутів та команд термінального виведення для пристроїв *Cisco*.
13. Наведіть перелік та поясніть призначення основних команд, що стосуються роботи з конфігураціями пристроїв *Cisco*.
14. Парольний доступ до пристроїв *Cisco*.
15. Наведіть перелік та поясніть призначення основних команд налагодження парольного доступу до пристроїв *Cisco*.

5.1.2 НАЛАШТУВАННЯ ОС *WINDOWS* ТА ОС *LINUX*

АДРЕСАЦІЯ РОБОЧИХ СТАНЦІЙ ОС *WINDOWS*

Розробниками ОС *Windows* для спрощення налагодження функціонування вузла в мережі введено узагальнене поняття „Мережне підключення”, яке об’єднує у собі всі компоненти, що необхідні для забезпечення передавання та приймання трафіка. Мережне підключення містить обов’язкові (базові) та додаткові (сервісні) компоненти. Базові компоненти забезпечують функціонування вузла в одній із ролей – клієнта, однорангового вузла чи сервера. Сервісні компоненти забезпечують обмін службовими повідомленнями.

Базовими компонентами мережного підключення вузла ОС *Windows* за замовчуванням є:

- драйвер мережевого адаптера (*Network Adapter Driver*);
- міжмережний протокол *IP* (*Internet Protocol*);
- служба доступу до файлів і принтерів мереж Microsoft (*File and Printer Sharing for Microsoft Networks*);
- клієнт для мереж Microsoft (*Client for Microsoft Networks*).

Додатковими компонентами мережного підключення вузла ОС *Windows* за замовчуванням є:

- планувальник пакетів QoS (*QoS Packet Sheduler*);
- драйвер в/в „картографа” топології каналного рівня (*Link-Layer Topology Discovery Mapper I/O Driver*);
- „відповідач” виявлення топології каналного рівня (*Link-Layer Topology Discovery Responder*).

За необхідності до мережного підключення можуть входити додаткові служби, клієнти, протоколи та інші сервісні компоненти.

Приклад створених мережних підключень для вузла ОС *Windows 7/8/10* наведені на рисунку 5.6.

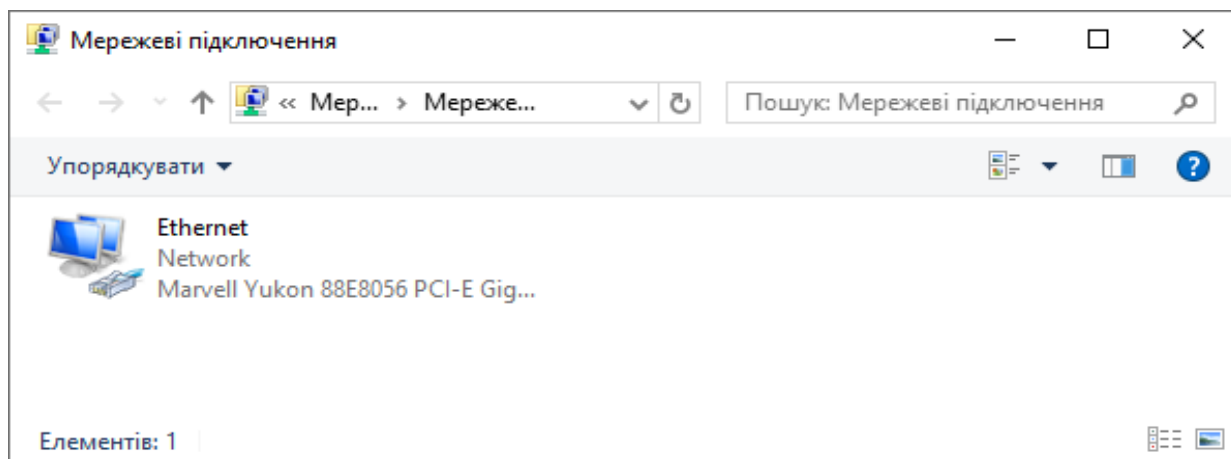


Рисунок 5.6 – Мережні підключення ОС *Windows*

Мережне підключення створюється автоматично після встановлення драйвера відповідного мережного адаптера/інтерфейсу. Можливе створення і видалення підключень у ручному режимі. Відмінності підключень різних мережних технологій є незначними. Перелік та стан створених підключень для вузлів ОС *Windows 7/8/10* можна побачити за допомогою додатків „Мережеві підключення” та „Центр мережевих підключень і спільного доступу”.

Після вибору мережного підключення можна отримати інформацію про його стан, зокрема, швидкість, період активності, кількість прийнятих і відправлених повідомлень, параметри адресації підключення. Приклад стану мережного підключення для вузла ОС *Windows 10* (на базі мережного адаптера *Ethernet*) наведено на рисунку 5.7.

Складові мережних підключень *Windows 10* (на базі мережних адаптерів *Ethernet*) за умови встановлення компонентів за замовчуванням наведені на рисунку 5.8.

Для налагодження або зміни параметрів *IP*-адресації мережного підключення необхідно обрати компонент – міжмережний протокол *IP*. Приклад статично встановлених основних параметрів *IP*-адресації версії 4 (*IP*-адреса вузла, маска, *IP*-адреса основного шлюзу, дві *IP*-адреси *DNS*-серверів) для мережного підключення вузла ОС *Windows 10* наведено на рисунку 5.8.

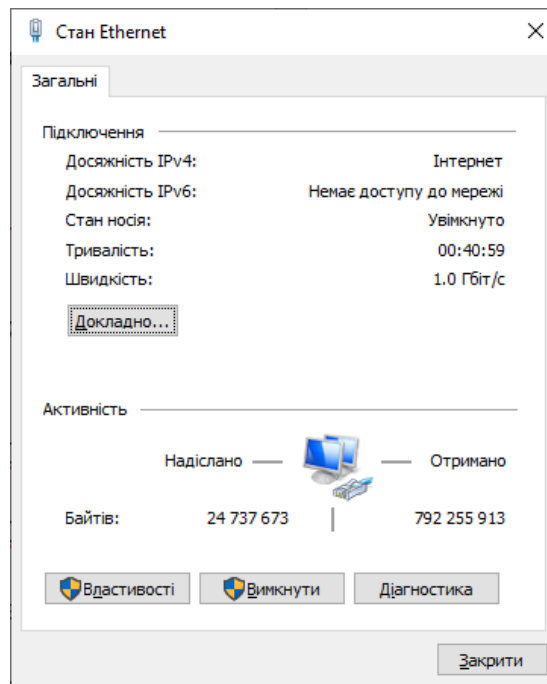


Рисунок 5.7 – Стан мережні підключення ОС Windows

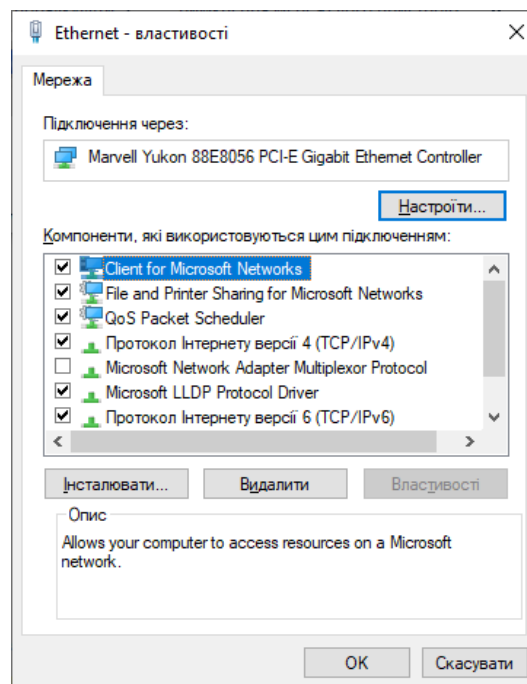


Рисунок 5.8 – Встановлені параметри IP-адресації вузла ОС Windows

Параметри іменування *Windows*-вузла (текстове ім'я та назву робочої групи/домену) можна вивести або змінити за допомогою вбудованого додатка „Система”. Приклад такого додатка для вузла ОС *Windows* 10 наведено на рисунку 5.9.

В ОС *Windows* існує можливість діагностики параметрів іменування вузла та параметрів адресації мережних адаптерів/інтерфейсів за допомогою відповідних мережних команд. Основними командами є команди **hostname**, **ipconfig**. Як додаткові команди можна застосовувати команди **getmac** та

systeminfo. Для перевірки зв'язку між вузлами *IP*-мережі застосовується команда **ping**.

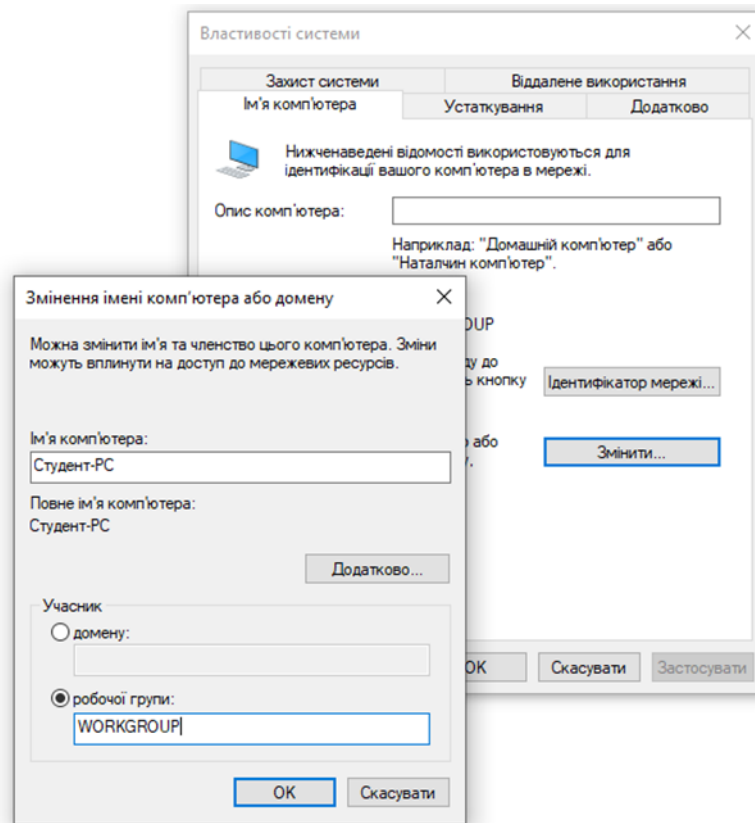


Рисунок 5.9 – Встановлені параметри іменування вузла ОС Windows

Команда **hostname** призначена лише для виведення текстового імені вузла. Основним призначенням команди **ipconfig** є діагностика стану мережних адаптерів та діагностика налагоджених параметрів *IP*-адресації мережних адаптерів/інтерфейсів вузла. За рахунок застосування різних ключів (параметрів) ця команда може застосовуватися і для керування процесом отримання, оновлення та вивільнення *IP*-адрес за умови динамічного їх призначення. Також команда **ipconfig** застосовується з метою перегляду та керування DNS-кешем пристрою. Команда **getmac** дає змогу визначити MAC-адреси та список мережних протоколів, пов'язаних із кожною адресою мережних адаптерів як локального вузла, так і інших вузлів локальної мережі. Команда **systeminfo** виводить повну інформацію про поточний вузол, зокрема і параметри *IP*-адресації мережних адаптерів/інтерфейсів. Детальну інформацію стосовно ключів зазначених вище команд можна отримати через довідку командного рядка або у загальній довідковій системі ОС Windows.

МОДЕЛЬНИЙ ПРИКЛАД НАЛАГОДЖЕННЯ ФУНКЦІОНУВАННЯ ВУЗЛІВ ОС Windows ЛОКАЛЬНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ

Розглянемо специфіку налагодження вузлів (робочих станцій) ОС Windows для локальної комп'ютерної мережі, схему якої наведено на рисунок 5.10.

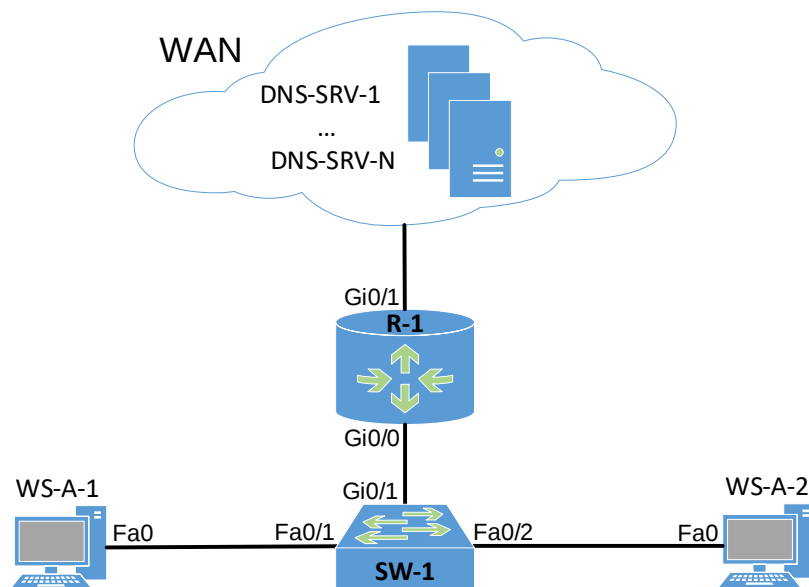


Рисунок 5.10 – Приклад мережі

Під час побудови даної мережі для з'єднання пристроїв використано дані з таблиці 5.8. Для налагодження параметрів адресації вузлів та комунікаційних пристроїв мережі використано дані з таблиці 5.9.

Таблиця 5.8 – Параметри інтерфейсів пристроїв для прикладу

Пристрій	Інтерфейс	Підключення до пристрою	Підключення до інтерфейсу
Маршрутизатор R-1	Gi0/1	WAN	WAN Interface
	Gi0/0	Комутатор SW-1	Gi0/1
Комутатор SW-1	Fa0/1	Робоча станція WS-A-1	Fa0
	Fa0/2	Робоча станція WS-A-2	Fa0
	Gi0/1	Маршрутизатор R-1	Gi0/0
WAN	WAN Interface	Маршрутизатор R-1	Gi0/1
Робоча станція WS-A-1	Fa0	Комутатор SW-1	Fa0/1
Робоча станція WS-A-2	Fa0		Fa0/2

Таблиця 5.9 – Параметри адресації мережі для прикладу

Мережа/Пристрій	Інтерфейс/Мережний адаптер/Шлюз	MAC-адреса	IP-адреса	Маска	Префікс
Мережа А	—	—	195.10.1.0	255.255.255.0	/24
Маршрутизатор R-1	GigabitEthernet 0/1	×	×	×	×
	GigabitEthernet 0/0	00-D0-B1-E1-14-11	195.10.1.254	255.255.255.0	/24
Комутатор SW-1	Інтерфейс Vlan 1	00-D0-BA-E4-0D-9B	195.10.1.252	255.255.255.0	/24
	Шлюз за замовчуванням	—	195.10.1.254	—	—

Продовження таблиці 5.9

Мережа/ Пристрій	Інтерфейс/Мереж- ний адаптер/Шлюз	MAC-адреса	IP-адреса	Маска	Пре- фікс
Робоча станція WS-A-1	Мережний адаптер	00-60-5C-16- 8B-30	195.10.1.1	255.255.255.0	/24
	Шлюз за замовчуванням	—	195.10.1.254	—	—
	Основний DNS-сер- вер	—	195.10.1.254	—	—
	Альтернат. DNS- сервер 1	—	8.8.8.8		—
	Альтернат. DNS- сервер 2	—	8.8.4.4	—	—
Робоча станція WS-A-2	Мережний адаптер	00-10-43-2C- BD-BB	195.10.1.2	255.255.255.0	/24
	Шлюз за замовчуванням	—	195.10.1.254	—	—
	Основний DNS-сер- вер	—	195.10.1.254	—	—
	Альтернат. DNS- сервер 1	—	8.8.8.8		—
	Альтернат. DNS- сервер 2	—	8.8.4.4	—	—

Результати виконання команд моніторингу та діагностування параметрів адресації та зв'язку для розглянутого прикладу

З метою перегляду інформації про налагоджені параметри іменування та параметри адресації мережних адаптерів/інтерфейсів вузлів мережі для розглянутого прикладу використано команди ОС *Windows* **hostname**, **ipconfig**. Для перевірки зв'язку між вузлами використано команду **ping**. Результати роботи цих команд для робочих станцій WS-A-1 – WS-A-2 наведено далі:

Результат виконання команди **hostname** на робочій станції WS-A-1

```
C:\>hostname
WS-A-1
C:\>
```

Результат виконання команди **ipconfig** на робочій станції WS-A-1

```
C:\>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : 
    IPv4 Address. . . . . : 195.10.1.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 195.10.1.254
```

Результат виконання команди ipconfig /all на робочій станції WS-A-1

C:\>ipconfig /all

Windows IP Configuration

Host Name : WS-A-1
Primary Dns Suffix :
Node Type : Hybrid
IP Routing Enabled. : No
WINS Proxy Enabled. : No

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . :
Description : AMD PCNET Family Ethernet Adapter (PCI)
Physical Address. : 00-60-5C-16-8B-30
DHCP Enabled. : Yes
Autoconfiguration Enabled : Yes
Link-local IPv6 Address : fe80::90ab:9744:8e9e:a0df%14(Preferred)
IPv4 Address. : 195.10.1.1
Subnet Mask : 255.255.255.0
Default Gateway : 192.168.1.254
DHCP Server : 192.168.1.254
DNS Servers : 195.10.1.254
8.8.8.8
8.8.4.4

Результат виконання команди hostname на робочій станції WS-A-2

C:\>hostname

WS-A-2

Результат виконання команди ipconfig /all на робочій станції WS-A-2

C:\>ipconfig /all

Windows IP Configuration

Host Name : WS-A-2
Primary Dns Suffix :
Node Type : Hybrid
IP Routing Enabled. : No
WINS Proxy Enabled. : No

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . :
Description : AMD PCNET Family Ethernet Adapter (PCI)
Physical Address. : 00-10-43-2C-BD-BB
DHCP Enabled. : No
Autoconfiguration Enabled : Yes
IPv4 Address. : 195.10.1.1(Preferred)
Subnet Mask : 255.255.255.0
Default Gateway : 192.168.1.254
DHCP Server : 192.168.1.254
DNS Servers : 195.10.1.254
8.8.8.8
8.8.4.4
NetBIOS over Tcpip. : Enabled

Результат успішної перевірки зв'язку між робочою станцією WS-A-1 та робочою станцією WS-A-2

```
C:\>ping 195.10.1.2
```

```
Pinging 195.10.1.2 with 32 bytes of data:
```

```
Reply from 195.10.1.2: bytes=32 time<1ms TTL=64
```

```
Reply from 195.10.1.2: bytes=32 time<1ms TTL=64
```

```
Reply from 195.10.1.2: bytes=32 time<1ms TTL=64
```

```
Reply from 195.10.1.2: bytes=32 time<1ms TTL=64
```

```
Ping statistics for 195.10.1.2:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Результат неуспішної перевірки зв'язку між робочою станцією WS-A-1 та шлюзом за замовчуванням

```
C:\>ping 195.10.1.254
```

```
Pinging 195.10.1.254 with 32 bytes of data:
```

```
Request timed out.
```

```
Request timed out.
```

```
Request timed out.
```

```
Request timed out.
```

```
Ping statistics for 195.10.1.254:
```

```
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

В ОС *Windows* після запуску команда **ping** здійснює чотири спроби перевірки зв'язку і завершує процес автоматично.

АДРЕСАЦІЯ РОБОЧИХ СТАНЦІЙ ОС *LINUX*

Налагоджені параметри адресації мережних інтерфейсів вузла ОС *Linux Debian/Ubuntu* зберігаються у конфігураційних файлах **/etc/network/interfaces** та **/etc/resolv.conf**. Перший файл призначений для збереження більшості параметрів *IP*-адресації мережних адаптерів/інтерфейсів (*IP*-адреса, маска, *IP*-адреса шлюзу тощо), другий – лише для збереження *IP*-адрес *DNS*-серверів. В останніх версіях ОС *Linux Debian/Ubuntu* *IP*-адреси *DNS*-серверів також зберігаються у файлі **/etc/network/interfaces**, відповідно не потрібно їх зберігати у файлі **/etc/resolv.conf**. Параметри іменування системи ОС *Linux Debian/Ubuntu* типово зберігаються у конфігураційному файлі **/etc/hostname**. Локальні відповідності між *IP*-адресами та доменними іменами вузлів зберігаються у файлі **/etc/hosts**.

Файл **/etc/network/interfaces** містить основні та додаткові параметри адресації мережних інтерфейсів вузла. Основними параметрами є параметри **auto**, **allow-auto**, **allow-hotplug**, **iface**. Параметр **auto** застосовується для автоматичної активації логічного або фізичного мережного інтерфейсу під час завантаження ОС. Параметр містить назву інтерфейсу. У деяких випадках

може застосовуватися аналогічний за функціоналом параметр **allow-auto**. Параметр **allow-hotplug** призначений для активації інтерфейсу після того, як ядро системи отримає повідомлення типу „HotPlug” від інтерфейсу. Параметр **iface** застосовується для зазначення протоколу (IPv4/IPv6), що активується на інтерфейсі, та способу призначення параметрів IP-адресації інтерфейсу. Активація IPv4 здійснюється за допомогою службової конструкції **inet**, IPv6 – конструкції **inet6**. Якщо призначення параметрів IP-адресації здійснюється динамічно, то додається допоміжна конструкція **dhcp**, якщо статично – конструкція **static**. Якщо застосовується статична IP-адресація інтерфейсу, то конфігураційний файл містить додаткові параметри **address**, **network**, **netmask**, **broadcast**, **gateway**, **dns-domain**, **dns-nameservers**, що містять відповідно IP-адресу вузла, IP-адресу мережі, мережну маску, широкомовну адресу мережі, IP-адресу шлюзу за замовчуванням, назву домену, IP-адреси основного та альтернативного DNS-серверів.

У конфігураційному файлі **/etc/resolv.conf** у ОС Linux Debian, як і в решти ОС Linux, містяться параметри **nameserver** та **search**. Перший із них містить IP-адресу DNS-сервера. Другий – назву домену, в якому здійснюється DNS-пошук. Параметр **nameserver** може повторюватися кілька разів для зазначення IP-адрес основного та альтернативних DNS-серверів поточного вузла. У конфігураційному файлі **/etc/hostname** міститься лише текстова назва вузла.

Приклади структур файлів **/etc/hostname**, **/etc/network/interfaces**, **/etc/resolv.conf** для ОС Linux Debian версії 8.3 наведені далі:

Структура файла **/etc/hostname** ОС Linux Debian

Debian8-3

Структура файла **/etc/network/interfaces** ОС Linux Debian

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
Source /etc/network/interfaces.d/*
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
allow-hotplug eth0
iface eth0 inet static
    address 195.10.1.1
    netmask 255.255.255.128
    broadcast 195.10.1.127
    gateway 195.10.1.126
    dns-domain example.com
    dns-nameservers 195.10.1.126 8.8.8.8
...
```

Структура файла `/etc/resolv.conf` ОС Linux Debian

```
nameserver 195.10.1.126
nameserver 8.8.8.8
search example.com
```

Для сучасних ОС *Linux Debian/Ubuntu* для налагодження текстового імені вузла та параметрів *IP*-адресації його мережних адаптерів/інтерфейсів рекомендується використовувати лише конфігураційні файли `/etc/hostname` та `/etc/network/interfaces`.

Активація збережених у конфігураційних файлах параметрів іменування вузла та *IP*-адресації мережних адаптерів/інтерфейсів ОС *Linux Debian* здійснюється або після повного перезавантаження системи, або після перезавантаження сервісу **networking**, що відповідає за функціонування мережі. Зупинку, запуск, перезавантаження згаданого сервісу можна виконати відповідно командами:

```
/etc/init.d/networking stop,
/etc/init.d/networking start,
/etc/init.d/networking restart
```

або

```
service networking stop,
service networking start,
service networking restart.
```

НАЛАГОДЖЕННЯ ПАРАМЕТРІВ *IP*-АДРЕСАЦІЇ В ОС *LINUX CENTOS/RED HAT*

Для ОС *Linux CentOS/Red Hat/Oracle/Fedora* налагоджені параметри іменування та загальні параметри адресації вузла зберігаються у конфігураційному файлі `/etc/sysconfig/network`. Параметри окремих мережних інтерфейсів (зокрема і параметри *IP*-адресації) зберігаються у файлах `/etc/sysconfig/network-scripts/ifcfg-if-name`, де **if-name** – назва мережного інтерфейсу (наприклад, **eth0** чи **enp0s3**). *IP*-адреси *DNS*-серверів зберігаються у файлі `/etc/resolv.conf`, а локальні відповідності між *IP*-адресами та доменними іменами – у файлі `/etc/hosts`.

Файл `/etc/sysconfig/network` може містити такі параметри, як: **NETWORKING**, **HOSTNAME**, **GATEWAY**, **DNS1**, **DNS2**, **SEARCH**, **GATEWAYDEV**, **NISDOMAIN**, **NOZEROCONF**. Параметри **NETWORKING** та **HOSTNAME** є обов'язковими параметрами. Решта параметрів є факультативними і зазначаються за потреби. Наприклад, із метою статичного призначення *IP*-адреси мережного інтерфейсу вузла зазначаються параметри **GATEWAY**, **DNS1**, **DNS2**, **SEARCH**.

Параметр **NETWORKING** застосовується для зазначення, чи буде налагоджено функціонування мережі, відповідно може набувати логічних значень **yes** або **no**. Параметр **HOSTNAME** повинен містити текстову назву вузла у форматі

повного доменного імені (*FQDN*, *Fully Qualified Domain Name*), за потреби може містити назву вузла у короткому записі. Параметр **GATEWAY** застосовується для зазначення *IP*-адреси шлюзу за замовчуванням. Параметр **GATEWAYDEV** типово містить назву мережного інтерфейсу, через який здійснюється надсилання повідомлень у мережу до шлюзу за замовчуванням. Цей параметр застосовується у випадках, коли кілька мережних інтерфейсів вузла підключені до однієї мережі. Параметри **DNS1**, **DNS2** містять *IP*-адреси основного та альтернативного *DNS*-серверів. Параметр **SEARCH** застосовується для зазначення текстового імені домену, в якому буде здійснюватися пошук службою *DNS*. Параметр **NISDOMAIN** зазначається у разі, коли необхідно налагодити доступ до інформаційної служби мережі NIS (Network Information Service). Містить доменну назву цієї служби. Параметр **NOZEROCONF** відповідає за деактивацію/активацію маршруту типу *zeroconf*, може набувати логічних значень **true** (маршрут не активовано) або **false** (маршрут активовано).

На відміну від конфігураційного файла **/etc/sysconfig/network**, файл **/etc/sysconfig/network-scripts/ifcfg-*<if-name>*** має набагато більше параметрів. Основними параметрами є: **TYPE**, **DEVICE**, **HWADDR**, **ONBOOT**, **BOOTPROTO**, **NETWORK**, **IPADDR**, **NETMASK**, **GATEWAY**, **DNS1**, **DNS2**, **DEFROUTE**, **PEERDNS**, **PEERROUTES**, **NAME**, **UUID**, **IPV4_FAILURE_FATAL**, **NM_CONTROLLED**.

Параметр **TYPE** застосовується для зазначення типу (технології) мережного інтерфейсу. Для технологій Ethernet/Fast Ethernet тощо застосовується загальне позначення **Ethernet**. Параметр **DEVICE** містить назву мережного інтерфейсу. Параметр **HWADDR** застосовується для зазначення апаратної адреси інтерфейсу. Для адаптерів Ethernet – це MAC-адреса, записана як AA:BB:CC:DD:EE:FF. Параметр **ONBOOT** застосовується для активації/деактивації пристрою у ході завантаження системи, може набувати значень **yes** або **no**. Параметр **BOOTPROTO** містить назву мережного протоколу, що застосовується під час завантаження системи. Може містити значення **DHCP**, **BOOTP**, **none**. Для статичного призначення параметрів *IP*-адресації в деяких системах слід зазначати **static**. Параметри **NETWORK**, **IPADDR**, **NETMASK**, **GATEWAY**, **DNS1**, **DNS2** містять відповідно *IP*-адресу мережі, *IP*-адресу вузла, мережну маску, *IP*-адресу шлюзу за замовчуванням, *IP*-адреси основного та альтернативного *DNS*-серверів.

Параметр **DEFROUTE** застосовується для зазначення поточного інтерфейсу як інтерфейсу за замовчуванням, може набувати значень **yes** або **no**. Параметр **PEERDNS** відповідає за активацію/деактивації можливості модифікації файла **/etc/resolv.conf** у процесі налагодження параметрів адресації вузла. Якщо цей параметр містить значення **no**, то зміни файлу заборонені, якщо значення **yes**, то зміни допускаються. За умови застосування протоколу DHCP параметр **PEERDNS** містить значення **yes**. Параметр **PEERROUTES** відповідає за

активацію/деактивацію внесення у таблицю маршрутизації вузла маршруту за замовчуванням, який отримано за допомогою протоколу DHCP. Може набувати значень **yes** або **no**. За умови застосування протоколу DHCP параметр **PEERROUTES** містить значення **yes**.

Параметр **NAME** застосовується для зазначення імені інтерфейсу, що буде використовуватися під час виведення інформації про мережні з'єднання. Параметр **UUID** містить унікальний ідентифікатор мережного інтерфейсу у системі. Ідентифікатор зазначається як послідовність, що містить шістнадцяткові цифри та знаки тире. Параметр **IPV4_FAILURE_FATAL** містить значення **yes** або **no**, за допомогою яких указується, які дії (вимикати чи не вимикати) виконувати з мережним адаптером у разі помилки. Параметр **NM_CONTROLLED** застосовується для активації можливості керування інтерфейсом за допомогою мережного демона Network Manager. Цей параметр може набувати значень **yes** або **no**.

Окрім вищезгаданих параметрів, можуть застосовувати й інші специфічні параметри. Інформацію стосовно їх використання можна отримати з документації системи.

У конфігураційному файлі **/etc/resolv.conf**, як правило, містяться параметри **nameserver** та **search**. Перший із них містить *IP*-адресу *DNS*-сервера. Другий – назву домену, в якому здійснюється *DNS*-пошук. Параметр **nameserver** може повторюватися кілька разів для зазначення *IP*-адрес основного та альтернативних *DNS*-серверів поточного вузла.

Приклади структур файлів **/etc/sysconfig/network**, **/etc/sysconfig/network-scripts/ifcfg-enp0s3**, **/etc/resolv.conf** для ОС *Linux CentOS* версії 7 наведені далі:

Структура файла **/etc/sysconfig/network** ОС *Linux CentOS*

```
NETWORKING=YES
HOSTNAME=WS-1
DNS1=195.10.1.126
DNS2=8.8.8.8
SEARCH=example.com
```

Структура файла **/etc/sysconfig/network-scripts/ifcfg-enp0s3** ОС *Linux CentOS*

```
DEVICE=enp0s3
HWADDR=00:21:70:10:7E:CD
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=static
IPADDR=195.10.1.1
NETMASK=255.255.255.128
#the GATEWAY is sometimes in: /etc/sysconfig/network
GATEWAY=195.10.1.126
```

Структура файла **/etc/resolv.conf** ОС *Linux CentOS*

```
nameserver 195.10.1.126
nameserver 8.8.8.8
search example.com
```

НАЛАГОДЖЕННЯ ПАРАМЕТРІВ IP-АДРЕСАЦІЇ В ОС *LINUX MICROCORE/TINYCORE*

Налагодження параметрів адресації та іменування в ОС *Linux TinyCore* можливе із застосуванням засобів графічного інтерфейсу та командного рядка. В ОС *Linux MicroCore* – лише із застосуванням засобів командного рядка. Налагодження із застосуванням засобів графічного інтерфейсу є досить простим й інтуїтивно зрозумілим процесом, налагодження із застосування засобів командного рядка є процесом більш складним, передбачає вміння застосовувати команди операційної системи та вміння користуватися одним із текстових редакторів (*nano* або *vi*) для редагування конфігураційних файлів.

Налагодження постійних параметрів IP-адресації в ОС *Linux MicroCore* має певну специфіку, пов'язану з тим, що, на відміну від інших ОС *Linux*, у цій ОС не застосовуються стандартні конфігураційні файли, які містять параметри мережної адресації. Тому для налагодження необхідно скористатися таким підходом. У файл початкового завантаження **/opt/bootlocal.sh** за допомогою текстового редактора необхідно додати один із наведених нижче сценаріїв. У першому з них для налагодження параметрів IP-адресації застосовуються команди **ifconfig** та **route add**, у другому – команди **ip addr** та **ip route add**.

Сценарій 1.

```
hostname WS-1
ifconfig eth0 10.10.10.2 netmask 255.255.255.0 up
route add default gw 10.10.10.1
echo "nameserver 8.8.8.8" > /etc/resolv.conf
echo "nameserver 8.8.4.4" > /etc/resolv.conf
```

Сценарій 2.

```
hostname WS_1
ip addr add 10.10.10.2/24 broadcast 10.10.10.255 dev eth0
ip route add default via 10.10.10.1
```

Для налагодження IP-адрес DNS-серверів у сценарії 2 замість команд **echo** необхідно за допомогою текстового редактора у конфігураційний файл **/etc/resolv.conf** системи додати такі рядки: **nameserver 8.8.8.8** та **nameserver 8.8.4.4**. Для збереження внесених змін необхідно виконати команду **/usr/bin/filetool.sh -b**. Після перезавантаження встановлені параметри адресації будуть активними.

МОДЕЛЬНИЙ ПРИКЛАД НАЛАГОДЖЕННЯ ФУНКЦІОНУВАННЯ ВУЗЛІВ ОС *LINUX* ЛОКАЛЬНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ

Розглянемо специфіку налагодження вузлів (робочих станцій) ОС *Linux* для локальної комп'ютерної мережі, схему якої наведено на рисунку 5.11.

Під час побудови даної мережі для з'єднання пристроїв використано дані з таблиці 5.8. Для налагодження параметрів адресації вузлів та комунікаційних пристроїв мережі використано дані з таблиці 5.9.

Два альтернативні сценарії налагодження параметрів адресації (MAC-адреси, IP-адреси, маски, IP-адреси шлюзу за замовчуванням) робочої станції WS-A-1 (ОС *Linux Debian*) за допомогою команд наведено нижче.

```
...
root@debian8-3:~# hostname WS-A-1
root@debian8-3:~# ifconfig eth0 down
root@debian8-3:~# ifconfig eth0 hw ether 00:10:11:49:ED:09
root@debian8-3:~# ifconfig eth0 195.10.1.3 netmask 255.255.255.0
root@debian8-3:~# ifconfig eth0 up
root@debian8-3:~# route add default gw 195.10.1.254
root@debian8-3:~#
...
```

```
...
root@debian8-3:~# hostname WS-A-1
root@debian8-3:~# ifdown eth0
root@debian8-3:~# ifconfig eth0 hw ether 00:10:11:49:ED:09
root@debian8-3:~# ifconfig eth0 195.10.1.3 netmask 255.255.255.0
root@debian8-3:~# ifup eth0
root@debian8-3:~# route add default gw 195.10.1.254
root@debian8-3:~#
...
```

Сценарій налагодження параметрів адресації (IP-адреси, маски, IP-адреси шлюзу за замовчуванням) робочої станції WS-A-2 (ОС *Linux CentOS*) за допомогою команд **ip** наведено нижче.

```
...
root@centos:~#hostname WS-A-2
root@WS_A_4:~#ip link set enp0s3 down
root@WS_A_4:~#ip addr add 195.10.1.4/24 dev enp0s3
root@WS_A_4:~#ip link set enp0s3 up
root@WS_A_4:~#ip route add default via 195.10.1.254
root@WS_A_4:~#
...
```

Сценарії налагодження параметрів IP-адресації комутатора та маршрутизатора мережі у модельному прикладі не розглядаються.

Результати виконання команд моніторингу та діагностування параметрів адресації та зв'язку для розглянутого прикладу

З метою перегляду інформації про налагоджені параметри іменування та параметри адресації мережевих адаптерів/інтерфейсів вузлів мережі для розглянутого прикладу використано команди ОС *Linux* **ifconfig**, **route**,

ip addr show, ip route show. Для перевірки зв'язку між вузлами використано команду **ping**. Результати роботи цих команд для робочих станцій WS-A-1 – WS-A-2 наведено відповідно далі:

Результат виконання команди ifconfig eth0 на робочій станції WS-A-1

```
root@debian8-3:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:10:11:49:ed:09
          inet addr:195.10.1.1  Bcast:195.10.1.255  Mask:255.255.255.0
          inet6 addr: fe80::210:11ff:fe49:ed09/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:148 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:26368 (25.7 KiB)  TX bytes:4014 (3.9KiB)

root@debian8-3:~#
```

Результат виконання команди route на робочій станції WS-A-1

```
root@debian8-3:~# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
Default          195.10.1.254    0.0.0.0          UG    0      0        0 eth0
195.10.1.0       *               255.255.255.0    U      0      0        0 eth0

root@debian8-3:~#
```

Результат виконання команди ip addr show enp0s3 на робочій станції WS-A-2

```
[root@localhost /]# ip addr show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP, LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    Link/ether 00:00:27:32:f9:e7 brd ff:ff:ff:ff:ff:ff
    inet 195.10.1.2/24 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::200:27ff:fe32:f9e7/64 scope link
        valid_lft forever preferred_lft forever
[root@localhost /]#
```

Результат виконання команди ip route show на робочій станції WS-A-2

```
[root@localhost /]# ip route show
Default via 195.10.1.254 dev enp0s3
195.10.1.0/24 dev enp0s3  proto kernel  scope link  src 195.10.1.2

[root@localhost /]#
```

Результат успішної перевірки зв'язку між робочою станцією WS-A-1 та робочою станцією WS-A-2

```
root@debian8-3:~# ping 195.10.1.2

PING (195.10.1.2) 56(84) bytes of data.
64 bytes from 195.10.1.2: icmp_seq=1 ttl=128 time=1.35 ms
64 bytes from 195.10.1.2: icmp_seq=2 ttl=128 time=0.987 ms
64 bytes from 195.10.1.2: icmp_seq=3 ttl=128 time=1.23 ms
64 bytes from 195.10.1.2: icmp_seq=4 ttl=128 time=0.936 ms
64 bytes from 195.10.1.2: icmp_seq=5 ttl=128 time=1.35 ms
64 bytes from 195.10.1.2: icmp_seq=6 ttl=128 time=0.837 ms
64 bytes from 195.10.1.2: icmp_seq=7 ttl=128 time=0.967 ms
```

```

64 bytes from 195.10.1.2: icmp_seq=8 ttl=128 time=1.40 ms
64 bytes from 195.10.1.2: icmp_seq=9 ttl=128 time=0.897 ms
64 bytes from 195.10.1.2: icmp_seq=10 ttl=128 time=1.27 ms
64 bytes from 195.10.1.2: icmp_seq=11 ttl=128 time=1.25 ms
64 bytes from 195.10.1.2: icmp_seq=12 ttl=128 time=1.25 ms
^C
--- 195.10.1.2 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 10018ms
Rtt min/avg/max/mdev = 0.837/1.136/1.404/0.203 ms

root@debian8-3:~#

```

Результат неуспішної перевірки зв'язку між робочою станцією WS-A-3 та шлюзом за замовчуванням (відімкненим інтерфейсом GigabitEthernet 0/0 маршрутизатора R-1

```

root@debian8-3:~#ping 195.10.254

PING 195.10.1.254 (195.10.1.254) 56(84) bytes of data.
from 195.10.1.1 icmp_seq=1 Destination Host Unreachable
from 195.10.1.1 icmp_seq=2 Destination Host Unreachable
from 195.10.1.1 icmp_seq=3 Destination Host Unreachable
from 195.10.1.1 icmp_seq=4 Destination Host Unreachable
from 195.10.1.1 icmp_seq=5 Destination Host Unreachable
from 195.10.1.1 icmp_seq=6 Destination Host Unreachable
from 195.10.1.1 icmp_seq=7 Destination Host Unreachable
from 195.10.1.1 icmp_seq=8 Destination Host Unreachable
from 195.10.1.1 icmp_seq=9 Destination Host Unreachable
from 195.10.1.1 icmp_seq=10 Destination Host Unreachable
from 195.10.1.1 icmp_seq=11 Destination Host Unreachable
from 195.10.1.1 icmp_seq=12 Destination Host Unreachable
^C
--- 195.10.1.254 ping statistics ---
14 packets transmitted, 0 received, +12 errors, 100% packet loss, time 13042ms
pipe3

root@debian8-3:~#

```

В ОС *Linux* після запуску команда **ping** здійснює перевірку зв'язку безперервно, тому для переривання процесу необхідно натиснути комбінацію клавіш **<Ctrl>+<C>**.

ПРАКТИЧНЕ ЗАВДАННЯ

Мета роботи: ознайомитися з основними відомостями стосовно адресації вузлів в *IP*-мережах; ознайомитися з основними засобами налагодження параметрів адресації мережних адаптерів/інтерфейсів робочих станцій ОС *Windows/Linux*; отримати практичні навички по-будови локальної мережі на базі комутатора *Ethernet* та навички налагодження, керування, моніторингу та діагностування роботи мережних адаптерів/інтерфейсів робочих станцій ОС *Windows/Linux*; дослідити процеси функціонування мережних адаптерів/інтерфейсів робочих станцій та процеси передачі даних у побудованій мережі.

Порядок виконання роботи

1. У середовищі програмного емулятора створити проект локальної комп'ютерної мережі (рисунок 5.11), яка складається не менше ніж із чотирьох вузлів (робочих станцій) ОС *Windows*. Для вибору ОС вузла скористатися даними таблиці 5.10. Для побудованої мережі заповнити описову таблицю, яка аналогічна таблиці 5.8.

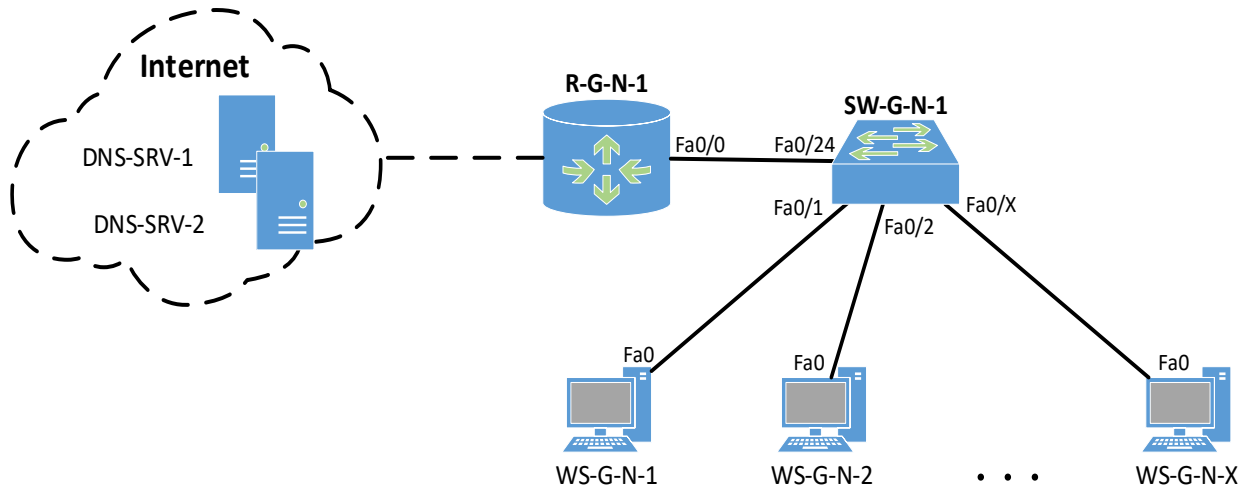


Рисунок 5.11 – Проект мережі

Таблиця 5.10 – Операційні системи вузлів (робочих станцій) локальної комп'ютерної мережі

№ варіанта	WS-G-N-01	WS-G-N-02	WS-G-N-03	WS-G-N-04	№ варіанта	WS-G-N-01	WS-G-N-02	WS-G-N-03	WS-G-N-04
1	W7	W10	LC/R	LM/T	16	W10	W7	LD/U	LC/R
2	W10	W7	LM/T	LC/R	17	W7	W10	LM/T	LD/U
3	W7	W10	LD/U	LM/T	18	W10	W7	LC/R	LD/U
4	W10	WXP	LD/U	LC/R	19	W10	W7	LC/R	LM/T
5	W10	W7	LM/T	LD/U	20	W7	W10	LM/T	LC/R
6	W10	W7	LC/R	LD/U	21	W7	W10	LD/U	LM/T
7	W10	W7	LC/R	LM/T	22	W10	W7	LD/U	LC/R
8	W7	W10	LM/T	LC/R	23	W10	W7	LM/T	LD/U
9	W7	W10	LD/U	LM/T	24	W7	W10	LC/R	LD/U
10	W10	W10	LD/U	LC/R	25	W7	W7	LC/R	LM/T
11	W7	W7	LM/T	LD/U	26	W10	W10	LM/T	LC/R
12	W7	W10	LC/R	LD/U	27	W7	W10	LD/U	LM/T
13	W10	W7	LC/R	LM/T	28	W10	W7	LD/U	LC/R
14	W7	W7	LM/T	LC/R	29	W7	W10	LM/T	LD/U
15	W7	W10	LD/U	LM/T	30	W10	W7	LC/R	LD/U

Примітка: W7 – ОС *Windows 7*; W10 – *Windows 10*; LD/U – ОС *Linux Debian/Ubuntu*; LC/R – ОС *Linux CentOS/Red Hat*; LM/T – ОС *Linux MicroCore/TinyCore*

2. Розробити схему адресації пристроїв (як кінцевих, так і проміжних вузлів) мережі. Для цього скористатися даними таблиць 5.11-12. Під час розрахунку враховувати, що комутатору та інтерфейсу маршрутизатора мережі також виділяється по одній *IP*-адресі. Маску/префікс мережі визначити з урахуванням необхідності економії адрес. Результати навести у вигляді таблиці, яка аналогічна таблиці 5.9.

3. Провести налагодження параметрів іменування та *IP*-адресації мережних адаптерів/інтерфейсів робочих станцій мережі згідно з даними з використанням засобів графічного інтерфейсу.

4. Перевірити можливість інформаційного обміну між робочими станціями мережі. У разі виявлення проблем зв'язку знайти та усунути їх причини.

Таблиця 5.11 – Параметри для розрахунку

№ варіанта	<i>IP</i> -адреса мережі	Кількість робочих станцій у мережі	№ варіанта	<i>IP</i> -адреса мережі	Кількість робочих станцій у мережі
1	191.G.N.0	9	16	206.G.N.0	10
2	192.G.N.0	8	17	207.G.N.0	35
3	193.G.N.0	12	18	208.G.N.0	11
4	194.G.N.0	16	19	209.G.N.0	15
5	195.G.N.0	20	20	210.G.N.0	19
6	196.G.N.0	24	21	211.G.N.0	23
7	197.G.N.0	28	22	212.G.N.0	27
8	198.G.N.0	32	23	213.G.N.0	31
9	199.G.N.0	36	24	214.G.N.0	39
10	200.G.N.0	40	25	215.G.N.0	47
11	201.G.N.0	48	26	216.G.N.0	55
12	202.G.N.0	56	27	217.G.N.0	63
13	203.G.N.0	64	28	218.G.N.0	71
14	204.G.N.0	72	29	219.G.N.0	78
15	205.G.N.0	80	30	220.G.N.0	87

5. Провести налагодження параметрів *IP*-адресації із застосуванням конфігураційних файлів.

6. Перевірити можливість інформаційного обміну між робочими станціями мережі після змін. У разі виявлення проблем зв'язку знайти та усунути їх причини.

Таблиця 5.12 – Дані для визначення параметрів адресації мережі

№ ва-ріанта	IP-адреса шлюзу за замовчуванням, IP-адреса основного DNS-сервера	IP-адреса альтернативного DNS-сервера 1	IP-адреса альтернативного DNS-сервера 2
1	Перша IP-адреса діапазону	Level3 Communications	Level3 Communications
2	Остання IP-адреса діапазону	Google	Google
3	Перша IP-адреса діапазону	OpenDNS Home	OpenDNS Home
4	Остання IP-адреса діапазону	Securly	Securly
5	Перша IP-адреса діапазону	Comodo Secure DNS	Comodo Secure DNS
6	Остання IP-адреса діапазону	DNS Advantage	DNS Advantage
7	Перша IP-адреса діапазону	Norton ConnectSafe	Norton ConnectSafe
8	Остання IP-адреса діапазону	SafeDNS	SafeDNS
9	Перша IP-адреса діапазону	OpenNIC	OpenNIC
10	Остання IP-адреса діапазону	Public-Root	Public-Root
11	Перша IP-адреса діапазону	Level3 Com-ns	Level3 Com-ns
12	Остання IP-адреса діапазону	Google	Google
13	Перша IP-адреса діапазону	OpenDNS Home	OpenDNS Home
14	Остання IP-адреса діапазону	Securly	Securly
15	Перша IP-адреса діапазону	Comodo Secure DNS	Comodo Secure DNS
16	Остання IP-адреса діапазону	DNS Advantage	DNS Advantage
17	Перша IP-адреса діапазону	Norton ConnectSafe	Norton ConnectSafe
18	Остання IP-адреса діапазону	SafeDNS	SafeDNS
19	Перша IP-адреса діапазону	OpenNIC	OpenNIC
20	Остання IP-адреса діапазону	Public-Root	Public-Root
21	Перша IP-адреса діапазону	Level3 Com-ns	Level3 Com-ns
22	Остання IP-адреса діапазону	Google	Google
23	Перша IP-адреса діапазону	OpenDNS Home	OpenDNS Home
24	Остання IP-адреса діапазону	Securly	Securly
25	Перша IP-адреса діапазону	Comodo Secure DNS	Comodo Secure DNS
26	Остання IP-адреса діапазону	DNS Advantage	DNS Advantage
27	Перша IP-адреса діапазону	Norton ConnectSafe	Norton ConnectSafe
28	Остання IP-адреса діапазону	SafeDNS	SafeDNS
29	Перша IP-адреса діапазону	OpenNIC	OpenNIC
30	Остання IP-адреса діапазону	Public-Root	Public-Root

КОНТРОЛЬНІ ПИТАННЯ

1. Які адреси достатньо застосовувати для здійснення інформаційного обміну між вузлами будь-якої IP-мережі?

2. Які адреси було введено для зручності роботи користувача?
3. Наведіть приклади текстових адрес.
4. Наведіть перелік основних публічних *DNS*-серверів.
5. Особливості призначення *IP*-адрес *DNS*-серверів.
6. Наведіть перелік параметрів *IP*-адресації вузла.
7. Поняття „мережне підключення” ОС *Windows*.
8. Яким чином створюється мережне підключення в ОС *Windows*?
9. Основні складові «мережного під’єднання» ОС *Windows* та їх призначення.
10. Додаткові компоненти мережного підключення вузла ОС *Windows* за замовчуванням.
11. Особливості налагодження *MAC*-адрес мережних адаптерів/інтерфейсів ОС *Windows*.
12. Особливості налагодження текстових імен вузлів ОС *Windows*.
13. Призначення команд **hostname** та **ipconfig** в ОС *Windows*?
14. Призначення команд **getmac** та **systeminfo** в ОС *Windows*?
15. Яким чином здійснюється налагодження текстового імені вузла та параметрів *IP*-адресації мережних адаптерів/інтерфейсів вузла ОС *Linux* на відміну від ОС *Windows*?
16. Позначення мережних адаптерів/інтерфейсів ОС *Linux Debian/Ubuntu*.
17. Позначення мережних адаптерів/інтерфейсів ОС *Linux CentOS/Redhat*.
18. Наведіть перелік та поясніть призначення конфігураційних файлів, що містять параметри адресації мережних адаптерів/інтерфейсів ОС *Debian/Ubuntu*.
19. Наведіть перелік та поясніть призначення параметрів конфігураційного файлу **/etc/network/interfaces** ОС *Debian/Ubuntu*.
20. Наведіть перелік та поясніть призначення параметрів конфігураційного файлу **/etc/resolv.conf** ОС *Debian/Ubuntu*.
21. Яким чином здійснюється активація збережених у конфігураційних файлах параметрів іменування вузла та *IP*-адресації мережних адаптерів/інтерфейсів ОС *Linux Debian*?
22. Наведіть перелік та поясніть призначення конфігураційних файлів, що містять параметри адресації мережних адаптерів/інтерфейсів ОС *CentOS/Redhat*.
23. Наведіть перелік та поясніть призначення параметрів конфігураційного файлу **/etc/sysconfig/network** ОС *CentOS/Redhat*.
24. Наведіть перелік та поясніть призначення параметрів конфігураційного файлу **/etc/sysconfig/network-scripts/ifcfg-<if-name>** ОС *CentOS/Redhat*.
25. Поясніть особливості налагодження *MAC*-адрес мережних адаптерів/інтерфейсів ОС *Linux*.
26. Наведіть перелік та поясніть призначення основних команд для налагодження параметрів *IP*-адресації мережних адаптерів/інтерфейсів ОС *Linux*.

5.1.3 ФІЗИЧНА ТА ЛОГІЧНА АДРЕСАЦІЯ ВУЗЛІВ КОМП'ЮТЕРНИХ МЕРЕЖ

АНАЛІЗ ТА ВИЗНАЧЕННЯ ПАРАМЕТРІВ АДРЕСАЦІЇ

Приклад визначення виробника за MAC-адресою

Визначити, якою (унікальною, груповою, широкомовною) та у яких випадках (адреса відправника, адреса отримувача) може застосовуватися задана MAC-адреса **0C-8B-FD-93-63-EB**. За можливості визначити виробника мережного адаптера/інтерфейсу чи мережний протокол, який застосовує дану адресу.

Розв'язання. Для розв'язання даного прикладу необхідно старший байт 0C заданої MAC-адреси записати у двійковій системі числення: **00001100**

Молодші два біти цього байта дають змогу визначити, якою є MAC-адреса. Оскільки молодший біт $G/L = 0$ та наступний за ним біт $I/G = 0$, можна зробити висновок, що задана MAC-адреса є унікальною глобальною адресою, тобто може бути призначеною мережевому адаптеру/інтерфейсу. Оскільки проаналізована адреса є унікальною, то вона може застосовуватися і як адреса відправника, і як адреса отримувача кадру.

Унікальний ідентифікатор виробника *OUI* заданої MAC-адреси має значення: **0C-8B-FD**

Для визначення виробника, якому виділений даний *OUI*, скористаємося пошуковою системою <http://www.macvendorlookup.com>. Результати пошуку наведені на рисунку 5.12.

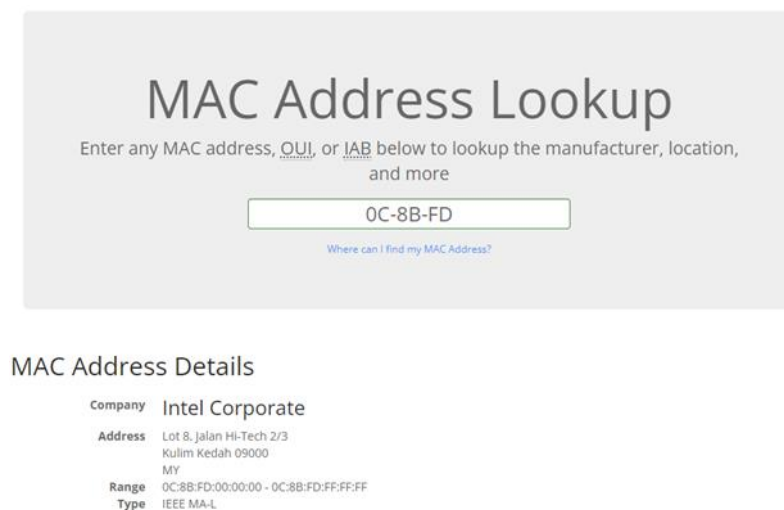


Рисунок 5.12 – Результат пошуку *OUI* виробника

Ідентифікатор виробника **0C-8B-FD** виділено для Intel Corporate. Діапазон можливих адрес мережних адаптерів/інтерфейсів для цього *OUI*:

0C-8B-FD-00-00-00 – 0C-8B-FD-FF-FF-FF.

Приклад визначення параметрів *IP*-адрес

Для заданої *IP*-адреси мережного адаптера/інтерфейсу вузла **172.205.14.1** із застосуванням класового підходу визначити такі параметри *IP*-адресації: клас *IP*-адреси; пряму класову маску мережі; інверсну класову маску мережі; класовий префікс мережі; *IP*-адресу (номер) мережі; *IP*-адресу (номер) вузла; мінімальну *IP*-адресу діапазону, що може використовуватися для адресації вузлів мережі; максимальну *IP*-адресу діапазону, що може використовуватися для адресації вузлів мережі; широкомовну *IP*-адресу мережі; кількість вузлів (*IP*-адрес вузлів), які можуть входити в мережу.

Розв’язання. Як відомо, *IP*-адреса містить у собі як *IP*-адресу (номер) мережі, так і *IP*-адресу (номер) вузла. Кількості байтів, які виділяються на *IP*-адресу мережі та *IP*-адресу вузла, визначаються на основі таблиці класів. Задана *IP*-адреса **172.205.14.1** за даними таблиці класів належить до класу **B**:

- Класовою маскою для мереж класу B є маска: **255.255.0.0**
- Інверсною класовою маскою для мереж класу B є маска: **0.0.255.255**
- Класовим префіксом для мереж класу B відповідно є префікс: **/16**
- Для класу B на номер мережі виділяється два перших байти *IP*-адреси. Відповідно *IP*-адреса мережі матиме вигляд: **172.205.0.0**
- Для класу B на номер вузла виділяється два останніх байти *IP*-адреси. Відповідно частина *IP*-адреси, що відповідатиме за номер вузла матиме вигляд: ***.*.14.1**

IP-адреса мережі і широкомовна *IP*-адреса (нульова й остання *IP*-адреси відповідно) не можуть призначатися вузлам. Тому мінімальною *IP*-адресою для діапазону, що може використовуватися для адресації вузлів, є *IP*-адреса, наступна за *IP*-адресою мережі, а максимальною *IP*-адресою – *IP*-адреса, яка передує широкомовній *IP*-адресі:

- У нашому випадку мінімальною *IP*-адресою вузла є адреса: **172.205.0.1**
- Максимальною *IP*-адресою вузла є адреса: **172.205.255.254**
- Широкомовною *IP*-адресою мережі є адреса: **172.205.255.255**

Кількість вузлів (*IP*-адрес вузлів), які можуть входити в мережу, розраховується за формулою:

$$K_{\text{вузлів}} = 2^{(32 - \text{Класовий префікс})} - 2,$$

або визначається за даними таблиці класів.

У нашому випадку кількість вузлів становить:

$$K_{\text{вузлів}} = 2^{(32 - 16)} - 2 = 2^{16} - 2 = 65536 - 2 = 65534 \text{ вузли.}$$

Приклад визначення оптимальних параметрів в залежності від кількості вузлів

Для мережі, у якій функціонує задана кількість вузлів – 1262, із застосуванням класового підходу: визначити оптимальні (щодо економії адрес) маску і префікс мережі; обрати відповідну *IP*-адресу мережі; визначити параметри *IP*-адресації обраної мережі.

Розв’язання. Під час розв’язання даного виду задач слід пам’ятати, що, окрім *IP*-адрес, що призначаються вузлам, у мережі наявні і розраховуються *IP*-адреса мережі та широкомовна *IP*-адреса. Тому до заданої кількості *IP*-адрес вузлів необхідно додати ще дві адреси. Оскільки адресація починається з нуля, то одну *IP*-адресу необхідно відняти. Тому загальна кількість *IP*-адрес мережі (включаючи *IP*-адресу мережі та широкомовну адресу) X формується як:

$$X = K_{\text{вузлів}} + 2 - 1.$$

Для умов задачі X дорівнює:

$$X = 1262 + 2 - 1 = 1263.$$

За даними таблиці класів одночасне використання такої кількості *IP*-адрес в одній мережі можливе у випадках, коли мережа належить або до класу **A** (максимальна кількість *IP*-адрес вузлів – 16777214), або до класу **B** (максимальна кількість *IP*-адрес вузлів – 65534). Задля економії адрес доцільно обрати мережу класу **B**.

Отже, оптимальною маскою для мережі з кількістю вузлів 1262 буде класова маска **255.255.0.0**. Цій масці відповідає класовий префікс **/16**.

Як *IP*-адресу мережі обираємо довільну *IP*-адресу класу **B**, наприклад адресу – **180.1.0.0**:

- Мінімальною *IP*-адресою вузла цієї мережі є адреса: **180.1.0.1**
- Максимальною *IP*-адресою вузла цієї мережі є адреса: **180.1.255.254**
- Широкомовною *IP*-адресою мережі є адреса: **180.1.255.255**

Кількість вузлів (*IP*-адрес вузлів), які можуть входити в мережу, становить:

$$K_{\text{вузлів}} = 2^{(32-16)} - 2 = 2^{16} - 2 = 65536 - 2 = 65534 \text{ вузли.}$$

З них 1262 *IP*-адреси використовуються, а 64272 *IP*-адреси – не використовуються.

ПРАКТИЧНЕ ЗАВДАННЯ

Мета роботи: ознайомитися із загальними принципами адресації вузлів комп'ютерних мереж; ознайомитися із структурою, видами та застосуванням MAC-адрес; ознайомитися із структурою, видами та застосуванням IP-адрес версій 4; отримати практичні навички аналізу та визначення параметрів MAC-адрес; отримати практичні навички аналізу, визначення та розрахунку параметрів IP-адрес версії 4 із застосуванням класового підходу.

Порядок виконання роботи

1. Визначити, якими (унікальними, груповими, широкомовними) є задані три MAC-адреси (таблиця 5.13). Також визначити, у яких випадках (як адреси відправників чи як адреси отримувачів) можуть застосовуватися ці MAC-адреси. За можливості для кожної із MAC-адрес визначити виробника мережевого адаптера/інтерфейсу чи мережний протокол, який застосовує цю адресу.

Таблиця 5.13 – Параметри для розрахунку

№ варіанта	MAC-адреса 1	MAC-адреса 2	MAC-адреса 3
1	000C418545AA	01000CCCCCD	FFFFFFFFFFFF
2	4485001278D2	FFFFFFFFFFFF	0180C2000001
3	FFFFFFFFFFFF	4C80937895AA	0180C2000003
4	0180C2000008	000C87D2347A	FFFFFFFFFFFF
5	00E0FC91A23F	FFFFFFFFFFFF	0180C2000000
6	FFFFFFFFFFFF	0180C200000E	28315200128D
7	0005851D54FF	180C2000002F	FFFFFFFFFFFF
8	0180C2000002	FFFFFFFFFFFF	F4A73939468A
9	FFFFFFFFFFFF	1CFA6886ABE1	01005E000002
10	0180C200000E	0080C881C2C1	FFFFFFFFFFFF
11	C8F4061145D1	FFFFFFFFFFFF	333300000005
12	FFFFFFFFFFFF	C40415DA13E1	01005E000002
13	080088A080A8	01005E000002	FFFFFFFFFFFF
14	333300000066	FFFFFFFFFFFF	CC5D4E0101FF
15	FFFFFFFFFFFF	001460105AD	01005E000005
16	000C41A145E2	01000CCCCC	FFFFFFFFFFFF
17	0180C2000000	FFFFFFFFFFFF	3413E8114585
18	FFFFFFFFFFFF	14ABC5B1D1A1	0180C2000002
19	000C87DD11A1	0180C2000007	FFFFFFFFFFFF
20	0180C200000D	FFFFFFFFFFFF	001E10FFD311
21	FFFFFFFFFFFF	18D11F0125DF	0180C2000003
22	00058512DDA1	180C20000020	FFFFFFFFFFFF
23	180C20000020	FFFFFFFFFFFF	88A2E5FF23A1
24	FFFFFFFFFFFF	000AEB74CB11	01005E000001
25	00A0C078D113	011B19000000	FFFFFFFFFFFF
26	333300000001	FFFFFFFFFFFF	00040DD0041A
27	FFFFFFFFFFFF	2CB05D7EE111	333300000001
28	000088000001	01005E000008	FFFFFFFFFFFF
29	333300000016	FFFFFFFFFFFF	F41563F22F22
30	FFFFFFFFFFFF	040A8383040A	01005E000016
31	C40415DA13EE	0180C2000007	FFFFFFFFFFFF
32	000C41A1F5E8	FFFFFFFFFFFF	01005E000002
33	FFFFFFFFFFFF	0180C2000008	0080C88DF2AA

2. Для кожної із заданих трьох *IP*-адрес мережних адаптерів/інтерфейсів вузлів (таблиця 5.14) із застосуванням класового підходу визначити такі параметри *IP*-адресації мереж: клас *IP*-адреси; пряму класову маску мережі; інверсну класову маску мережі; класовий префікс мережі; *IP*-адресу (номер) мережі; *IP*-адресу (номер) вузла; мінімальну *IP*-адресу діапазону, що може використовуватися для адресації вузлів мережі; максимальну *IP*-адресу діапазону, що може використовуватися для адресації вузлів мережі; широкомовну *IP*-адресу мережі; кількість вузлів (*IP*-адрес вузлів), які можуть входити в мережу.

Таблиця 5.14 – Параметри для розрахунку

№ варіанта	<i>IP</i> -адреса 1	<i>IP</i> -адреса 2	<i>IP</i> -адреса 3
1	45.12.17.199	206.157.15.1	134.143.14.13
2	136.88.226.25	55.17.18.19	207.80.218.33
3	208.74.183.175	138.68.177.181	65.65.55.66
4	75.164.52.13	209.86.224.27	140.76.185.173
5	142.98.241.2	85.73.182.176	210.12.201.102
6	211.71.208.43	144.73.210.41	95.69.178.180
7	105.84.222.29	212.78.216.35	146.1.0.189
8	150.96.237.6	115.75.212.39	213.67.176.182
9	214.90.55.7	160.255.1.1	125.25.52.12
10	5.97.239.4	215.76.214.37	170.72.181.177
11	180.77.98.174	10.255.255.254	216.70.179.179
12	217.222.25.187	190.255.255.15	15.71.180.178
13	20.66.202.49	218.0.255.254	185.85.15.155
14	175.1.1.255	25.94.235.12	219.19.21.254
15	220.92.231.10	165.93.233.8	30.130.13.31
16	10.174.1.55	130.12.5.134	192.255.1.1
17	135.1.255.147	20.255.255.1	195.145.13.240
18	193.85.197.7	140.0.0.51	30.255.255.1
19	40.7.7.143	194.255.1.254	145.1.25.14
20	150.136.18.177	50.1.1.254	195.0.0.1
21	196.88.99.11	155.0.5.7	60.60.20.215
22	70.85.19.1	197.1.0.143	160.169.240.232
23	170.2.15.218	80.10.0.155	198.162.0.1
24	199.66.75.201	175.19.0.7	90.255.255.1
25	100.99.18.55	200.82.220.31	180.255.1.254
26	185.69.206.45	110.255.255.0	201.54.254.4
27	202.76.17.11	190.0.0.252	120.25.4.254
28	15.10.11.33	203.75.184.174	128.1.1.255
29	130.65.200.51	25.0.255.254	204.254.0.254
30	205.55.87.94	130.67.204.47	35.10.19.147
31	12.47.243.200	168.15.86.79	199.185.145.12
32	176.172.92.15	205.61.17.48	8.188.8.9
33	220.38.0.10	56.17.198.200	155.55.15.51

3. Для мереж А та В, у яких функціонує задана кількість вузлів (таблиця 5.15), із застосуванням класового підходу: визначити оптимальні (щодо економії адрес) маску і префікс мережі; обрати відповідну *IP*-адресу мережі; визначити параметри *IP*-адресації обраної мережі. Розрахувати відсоток використання адресного простору для кожної із мереж.

Таблиця 5.15 – Параметри для розрахунку

№ варіанта	Кількість вузлів мережі А	Кількість вузлів мережі В	№ варіанта	Кількість вузлів мережі А	Кількість вузлів мережі В
1	35	511	18	16542	140
2	19	1023	19	7	1978
3	78	2047	20	12	9657
4	48	4095	21	143	1205
5	1999	63	22	1512	45
6	20365	3	23	872	69
7	299	10	24	652	82
8	220	986	25	7841	188
9	200200	125	26	15	255
10	58794	252	27	143	13018
11	9875	1011	28	126	1400
12	174	65535	29	2550	10
13	99	16382	30	738	78
14	130	131071	31	12	3348
15	37	32737	32	2058	120
16	31	987	33	28	140895
17	8191	15	34	14	9854

КОНТРОЛЬНІ ПИТАННЯ

1. Які є типи адрес, що застосовуються у сучасних мережах?
2. Дайте визначення фізичної адреси. Наведіть приклади фізичних адрес.
3. Дайте визначення логічної адреси. Наведіть приклади логічних адрес.
4. Дайте визначення текстової адреси. Наведіть приклади текстових адрес.
5. Надайте приклади видів та застосування *MAC*-адрес.
6. Поясніть структуру *MAC*-адреси.
7. *IP*-адреса версії 4. Види та застосування.
8. Структура *IP*-адреси версії 4.
9. *IP*-адреси вилучення версії 4.
10. Дайте перелік приватних *IP*-адрес версії 4.
11. Поясніть поняття маски та префікса мережі. Види масок.
12. I правило формування класів *IP*-адрес.
13. II правило формування класів *IP*-адрес.
14. Класи *IP*-адрес.
15. Класові маски та префікси.

5.1.4 IP АДРЕСАЦІЯ

БЕЗКЛАСОВА IP-АДРЕСАЦІЯ

Приклад визначення параметрів IP-адрес

Для заданої IP-адреси мережного адаптера/інтерфейсу вузла **175.12.187.92** та префікса **/21** мережі із застосуванням безкласового підходу визначити такі параметри IP-адресації мережі: маску (пряму маску) мережі; інверсну маску мережі; IP-адресу (номер) мережі; IP-адресу (номер) вузла; мінімальну IP-адресу діапазону, що може використовуватися для адресації вузлів мережі; максимальну IP-адресу діапазону, що може використовуватися для адресації вузлів мережі; широкомовну IP-адресу мережі; кількість вузлів (IP-адрес вузлів), які можуть входити в мережу.

Розв'язання. Для розв'язання даної задачі переводимо IP-адресу **175.12.187.92** з десяткової у двійкову систему числення:

10101111.00001100.10111011.01011100

Для визначення маски мережі скористаємося такими твердженнями: довжина маски мережі становить 32 біти; маска мережі у двійковій системі числення подається як дві взаємопродовжувані послідовності: перша послідовність (ліворуч) – неперервна послідовність одиниць та друга послідовність (праворуч) – неперервна послідовність нулів.

Записуємо маску мережі як послідовність одиниць (їх кількість – префікс показує кількість бітів, які використовуються для адресації (номера) мережі) та нулів (решта бітів, які використовуються для адресації (номера) вузла):

11111111.11111111.11111000.00000000

Результат у десятковій системі числення має вигляд:

255.255.248.0

Інверсна маска визначається шляхом виконання логічної операції інверсії (логічне NOT) над кожним із бітів прямої маски.

Результат виконання інверсії над попередньо визначеною прямою маскою у двійковій системі числення має вигляд:

00000000.00000000.00000111.11111111

Результат у десятковій системі числення має вигляд:

0.0.7.255

IP-адреса мережі визначається шляхом накладання прямої маски на вихідну *IP*-адресу, тобто виконання логічної операції кон'юнкції (логічне AND) між відповідними бітами вихідної *IP*-адреси та прямої маски:

$$\begin{array}{r} 10101111.00001100.10111011.01011100 \\ 11111111.11111111.11111000.00000000 \\ \hline 10101111.00001100.10111000.00000000 \end{array}$$

Результат виконання кон'юнкції між відповідними бітами вихідної *IP*-адреси та прямої маски у двійковій системі числення має вигляд:

$$10101111.00001100.10111000.00000000$$

Результат у десятковій системі числення має вигляд:

$$175.12.184.0$$

IP-адреса вузла визначається шляхом накладання інверсної маски на вихідну *IP*-адресу **175.12.184.0**, тобто виконання логічної операції кон'юнкції (логічне AND) між відповідними бітами вихідної *IP*-адреси та інверсної маски:

$$\begin{array}{r} 10101111.00001100.10111011.01011100 \\ 00000000.00000000.00000111.11111111 \\ \hline 00000000.00000000.00000111.01011100 \end{array}$$

Результат виконання кон'юнкції між відповідними бітами вихідної *IP*-адреси та інверсної маски у двійковій системі числення має вигляд:

$$00000000.00000000.00000111.01011100$$

Результат у десятковій системі числення має вигляд:

$$0.0.3.92$$

Як і в разі використання класового підходу, *IP*-адреса мережі і широкомовна *IP*-адреса (нульова й остання *IP*-адреси відповідно) не можуть призначатися вузлам. Тому мінімальною *IP*-адресою для діапазону, який може використовуватися для адресації вузлів мережі, є *IP*-адреса, наступна за *IP*-адресою мережі, а максимальною *IP*-адресою – *IP*-адреса, яка передусє широкомовній *IP*-адресі.

У нашому випадку мінімальна *IP*-адреса для нумерації вузлів у двійковій та десятковій системах числення має вигляд:

10101111.	00001100.	10111000.	00000001
175.	12.	184.	1

Максимальна *IP*-адреса для нумерації вузлів відповідно має вигляд:

10101111.	00001100.	10111111.	11111110
175.	12.	191.	254

Широкомовна *IP*-адреса відповідно має вигляд:

10101111.	00001100.	10111111.	11111111
175.	12.	191.	255

Кількість вузлів (*IP*-адрес вузлів) розраховується за формулою:

$$K_{\text{вузлів}} = 2^{(32-P)} - 2.$$

У нашому випадку з умови задачі префікс дорівнює 21, відповідно кількість вузлів (*IP*-адрес вузлів) дорівнює:

$$K_{\text{вузлів}} = 2^{(32-21)} - 2 = 2^{11} - 2 = 2048 - 2 = 2046.$$

Приклад визначення оптимальних параметрів в залежності від кількості вузлів

Для мережі, у якій функціонує задана кількість вузлів 62 із застосуванням безкласового підходу: визначити оптимальні (щодо економії адрес) маску і префікс мережі; обрати відповідну *IP*-адресу мережі; визначити параметри *IP*-адресації обраної мережі.

Розв’язання. Для розв’язання даного виду задач слід скористатися такими залежностями, що описують довжини *IP*-адреси та префікса у загальному вигляді:

$$\begin{aligned} N + H &= 32 \text{ біти,} \\ P &= N, \end{aligned}$$

де N – кількість бітів, які виділені для адресації мережі (номер мережі);
 H – кількість бітів, які виділені для адресації вузлів мережі;
 P – кількість бітів, які виділені для формування префікса мережі.

Кожному вузлу мережі відповідає одна *IP*-адреса. Слід пам’ятати, що, окрім *IP*-адрес вузлів, у мережі наявні і розраховуються *IP*-адреса мережі та широкомовна *IP*-адреса. Тому до заданої кількості *IP*-адрес вузлів необхідно додати ще дві адреси. Оскільки адресація починається з нуля, то необхідно одну *IP*-адресу відняти.

Для визначення значення H формується число X вигляду:

$$X = K_{\text{вузлів}} + 2 - 1.$$

Для умов задачі число X дорівнює:

$$X = 62 + 2 - 1 = 63.$$

Отримане число X переводиться із десяткової у двійкову систему числення:

$$X_{10} \rightarrow X_2.$$

Тобто

$$63_{10} = 111111_2$$

Кількість бітів у даному числі $H = 6$, і саме вони використовуються для нумерації вузлів.

Префікс мережі визначається як:

$$P = 32 - H.$$

Для нашого випадку $H = 6$ бітів. Отже,

$$P = 32 - 6 = 26 \text{ бітів.}$$

Префікс відповідно має вигляд – **/26**.

У двійковій системі числення маска мережі записується як послідовність бітів, що зазначають номер мережі (одиниці) та послідовність бітів, що зазначають номер вузла (нулі).

Для нашого випадку маска мережі у двійковій системі числення має вигляд:

$$\mathbf{11111111.11111111.11111111.11000000}$$

У десятковій формі маска мережі має вигляд:

$$\mathbf{255.255.255.192}$$

Як IP-адресу мережі обираємо довільну IP-адресу, наприклад адресу **195.10.1.0**.

Узагальнена *IP*-адреса мережі має вигляд:

195.10.1.0
255.255.255.192 або 195.10.1.0/26

Мінімальною *IP*-адресою вузла цієї мережі є адреса:

195.10.1.1

Максимальною *IP*-адресою вузла цієї мережі є адреса:

195.10.1.62

Широкомовною *IP*-адресою мережі є адреса:

195.10.1.63

Кількість вузлів (*IP*-адрес вузлів), які можуть входити в мережу, становить:

$$K_{\text{вузлів}} = 2^{(32-26)} - 2 = 2^6 - 2 = 64 - 2 = 62 \text{ вузли.}$$

Для цього прикладу всі *IP*-адреси, що наявні у мережі (окрім *IP*-адреси мережі та широкомовної *IP*-адреси), призначаються вузлам мережі.

ПРАКТИЧНЕ ЗАВДАННЯ

Мета роботи: ознайомитися із принципами безкласової адресації вузлів комп'ютерних мереж; отримати практичні навички аналізу, ви-значення та розрахунку параметрів *IP*-адрес версії 4 із застосуванням безкласового підходу; дослідити закономірності змін розмірності адресного простору мережі залежно від обраної маски/префіксу.

Порядок виконання роботи

1. Для заданих *IP*-адрес мережних адаптерів/інтерфейсів та префіксів мереж двох вузлів **А-1** та **В-1** (таблиця 5.16) із застосуванням безкласового підходу визначити такі параметри *IP*-адресації мереж: маску (пряму маску) мережі; інверсну маску мережі; *IP*-адресу (номер) мережі; *IP*-адресу (номер) вузла; мінімальну *IP*-адресу діапазону, що може використовуватися для адресації вузлів мережі; максимальну *IP*-адресу діапазону, що може використовуватися для адресації вузлів мережі; широкомовну *IP*-адресу мережі; кількість вузлів (*IP*-адрес вузлів), які можуть входити в мережу.

Таблиця 5.16 – Параметри для розрахунку

№ варіанта	IP-адреса мережного адаптера вузла А-1	Префікс мережного адаптера вузла А-1	IP-адреса мережного адаптера вузла В-1	Префікс мережного адаптера вузла В-1
1	192.255.1.1	/25	45.12.17.199	/12
2	195.145.13.240	/13	136.88.226.25	/26
3	30.255.255.1	/27	208.74.183.175	/14
4	145.1.25.14	/15	75.164.52.13	/28
5	195.0.0.1	/29	142.98.241.2	/16
6	60.60.20.215	/17	211.71.208.43	/30
7	160.169.240.232	/25	105.84.222.29	/18
8	198.162.0.1	/19	150.96.237.6	/26
9	90.255.255.1	/27	214.90.55.7	/20
10	180.255.1.254	/21	5.97.239.4	/28
11	201.54.254.4	/29	180.77.98.174	/22
12	120.25.4.254	/23	217.222.25.187	/30
13	128.1.1.255	/25	20.66.202.49	/13
14	204.254.0.254	/14	175.1.1.255	/26
15	35.10.19.147	/27	220.92.231.10	/15
16	206.157.15.1	/16	130.12.5.134	/28
17	55.17.18.19	/29	20.255.255.1	/17
18	138.68.177.181	/18	140.0.0.51	/30
19	209.86.224.27	/26	194.255.1.254	/19
20	85.73.182.176	/20	50.1.1.254	/28
21	144.73.210.41	/30	155.0.5.7	/21
22	212.78.216.35	/22	197.1.0.143	/27
23	115.75.212.39	/9	80.10.0.155	/23
24	160.255.1.1	/24	175.19.0.7	/30
25	215.76.214.37	/25	200.82.220.31	/18
26	10.255.255.254	/20	110.255.255.0	/29
27	190.255.255.15	/26	190.0.0.252	/21
28	218.0.255.254	/22	203.75.184.174	/28
29	25.94.235.12	/30	25.0.255.254	/23
30	165.93.233.8	/16	130.67.204.47	/30
31	205.13.160.155	/26	140.185.12.125	/18
32	176.12.18.120	/22	221.180.15.180	/28
33	195.180.14.180	/27	75.180.13.189	/20

2. Для мереж **А** та **В**, у яких функціонує задана кількість вузлів (таблиця 5.17), із застосуванням безкласового підходу: визначити оптимальні (щодо економії адрес) маску і префікс мережі; обрати відповідну IP-адресу мережі; визначити параметри IP-адресації обраної мережі; розрахувати відсоток використання адресного простору та відсоток вільних адрес для кожної із мереж.

Таблиця 5.17 – Параметри для розрахунку

№ варіанта	Кількість вузлів мережі А	Кількість вузлів мережі В	№ варіанта	Кількість вузлів мережі А	Кількість вузлів мережі В
1	15	51100	18	1011	31
2	143	10230	19	65535	7
3	126	20471	20	16382	12
4	255	40956	21	131071	143
5	738	63	22	32737	1512
6	51101	3	23	986	872
7	10230	10	24	125	652
8	20475	986	25	252	7841
9	4095	125	26	3	15
10	63	252	27	10	143
11	1011	1011	28	986	126
12	65535	65	29	125	2550
13	16382	182	30	252	738
14	131071	188	31	120	8792
15	32737	25	32	2750	50
16	125	8191	33	124	78945
17	252	16542	34	758	10

КОНТРОЛЬНІ ПИТАННЯ

1. Поняття безкласової адресації.
2. Що таке *CIDR*?
3. Відмінності класової та безкласової *IP*-адресації.
4. Якими залежностями користуються для аналізу та розрахунку параметрів *IP*-мережі за умови застосування безкласової *IP*-адресації?
5. За якою формулою розраховується кількість *IP*-адрес однієї *IP*-мережі, що можуть призначатися вузлам?
6. Які значення префіксу не застосовуються при розрахунку кількості *IP*-адрес однієї *IP*-мережі, що можуть призначатися вузлам?
7. Як впливає збільшення/зменшення значення префіксу на кількість *IP*-адрес вузлів мережі?
8. Яка пряма маска відповідає префіксу /6?
9. Яка пряма маска відповідає префіксу /13?
10. Яка пряма маска відповідає префіксу /27?
11. Яка інверсна маска відповідає прямій масці 224.0.0.0?
12. Яка інверсна маска відповідає прямій масці 255.252.0.0?
13. Яка інверсна маска відповідає прямій масці 255.255.128.0?
14. Яка інверсна маска відповідає прямій масці 255.255.255.248?
15. Яка кількість вузлів в мережі з префіксом /4?
16. Яка кількість вузлів в мережі з префіксом /12?
17. Яка кількість вузлів в мережі з префіксом /25?

5.1.5 ВІДПОВІДНІСТЬ ЛОГІЧНИХ *IPv4* АДРЕС ФІЗИЧНИМ

КОМАНДИ МОНІТОРИНГУ СТАНУ ТА ОПЕРАЦІЙ З *ARP*-ТАБЛИЦЯМИ ВУЗЛІВ ОС *WINDOWS* ТА ОС *LINUX*

Для моніторингу та операцій із записами *ARP*-таблиць у більшості сучасних ОС реалізовано однойменну команду (утиліту) **arp**. За допомогою цієї команди можна здійснювати як перегляд вмісту, так і видалення та додавання записів до *ARP*-таблиці. Набір параметрів команди **arp** є досить універсальним, близьким за записом і функціоналом для різних ОС. Це пов'язано з тим фактом, що спочатку команду, як і весь стек *TCP/IP*, було розроблено для ОС *Unix*, і пізніше перенесено в інші ОС. Перелік параметрів команди **arp**, що застосовуються в ОС *Windows 7/8/10*, ОС *Linux MicroCore*, ОС *Linux Debian*, наведено відповідно далі:

ОС *Windows*:

```
C:\>arp
```

Displays and modifies the IP-to-Physical address translation tables used by address resolution protocol (ARP).

```
ARP -s inet_addr eth_addr [if_addr]
```

```
ARP -d inet_addr [if_addr]
```

```
ARP -a [inet_addr] [-N if_addr] [-v]
```

- | | |
|------------|---|
| -a | Displays current ARP entries by interrogating the current protocol data. If inet_addr is specified, the IP and Physical addresses for only the specified computer are displayed. If more than one network interface uses ARP, entries for each ARP table are displayed. |
| -g | Same as -a. |
| -v | Displays current ARP entries in verbose mode. All invalid entries and entries on the loop-back interface will be shown. |
| inet_addr | Specifies an internet address. |
| -N if_addr | Displays the ARP entries for the network interface specified by if_addr. |
| -d | Deletes the host specified by inet_addr. inet_addr may be wildcarded with * to delete all hosts. |
| -s | Adds the host and associates the Internet address inet_addr with the Physical address eth_addr. The Physical address is given as 6 hexadecimal bytes separated by hyphens. The entry is permanent. |
| eth_addr | Specifies a physical address. |
| if_addr | If present, this specifies the Internet address of the interface whose address translation table should be modified. If not present, the first applicable interface will be used. |

Example:

- ```
> arp -s 157.55.85.212 00-aa-00-62-c6-09 Adds a static entry.
> arp -a Displays the arp table.
```

#### Наприклад:

|                      |                   |                                 |
|----------------------|-------------------|---------------------------------|
| arp -s 157.55.85.212 | 00-aa-00-62-c6-09 | - Додає статичний запис.        |
| arp -a               |                   | - Виводить <i>ARP</i> -таблицю. |

## OC Linux MicroCore:

```
root@box:~# arp
```

```
BusyBox v1.19.0 (2011-08-14 21:05:38 UTC) multi-call binary.
```

```
Usage: arp
```

```
[-vn] [-H HWTYPE] [-i IF] -a [HOSTNAME]
[-v] [-i IF] -d HOSTNAME [pub]
[-v] [-H HWTYPE] [-i IF] -s HOSTNAME HWADDR [temp]
[-v] [-H HWTYPE] [-i IF] -s HOSTNAME HWADDR [netmask MASK] pub
[-v] [-H HWTYPE] [-i IF] -Ds HOSTNAME IFACE [netmask MASK] pub
```

```
Manipulate ARP cache
```

```
-a Display (all) hosts
-s Set new ARP entry
-d Delete a specified entry
-v Verbose
-n Don't resolve names
-i IF Network interface
-D Read <hwaddr> from given device
-A,-p AF Protocol family
-H HWTYPE Hardware address type
```

## OC Linux Debian:

```
root@debian8-3~: # arp
```

```
Usage:
```

```
arp [-vn] [<HW>] [-i <if>] [-a] [<hostname>] <-Display ARP cache
arp [-v] [-i <if>] -d <host> [pub] <-Delete ARP entry
arp [-vnD] [<HW>] [-i <if>] -f [<filename>] <-Add entry from file
arp [-v] [<HW>] [-i <if>] -s <host> <hwaddr> [temp] <-Add entry
arp [-v] [<HW>] [-i <if>] -Ds <host> <if> [netmask <nm>] pub <-'''
```

```
-a display (all) hosts in alternative (BSD) style
-s, --set set a new ARP entry
-d, --delete delete a specified entry
-v, --verbose be verbose
-n, --numeric do n't resolve names
-i, --device specify network interface (e.g. eth0)
-D, --use-device read <hwaddr> from given device
-A, -p, --protocol specify protocol family
-f, --file read new entries from file or from /etc/ethers
```

```
<HW>=Use '-H <hw>' to specify hardware address type. Default: ether
```

```
List of possible hardware types (which support ARP) :
```

```
ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
netrom (AMPR NET/ROM) rose (AMPR ROSE) arcnet (ARCnet)
dlci (Frame Relay DLCI) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
irda (IrLAP) x25 (generic X.25) eui64 (Generic EUI-64)
```

## НАЛАГОДЖЕННЯ ПАРАМЕТРІВ ФУНКЦІОНУВАННЯ ПРОТОКОЛУ *ARP* НА ВУЗЛАХ ОС *WINDOWS* ТА ОС *LINUX*

Механізм керування часовими параметрами функціонування протоколу *ARP*, зокрема і *ARP*-таблицею, детально описаний у стандарті *RFC-826 „An Ethernet Address Resolution Protocol (or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware)”*. Кожен з виробників ОС має свою програмну реалізацію цього механізму. Типові часові параметри протоколу *ARP* зберігаються у конфігураційних файлах систем і пристроїв. Відповідно налагодження параметрів, насамперед тайм-ауту утримання записів у *ARP*-таблиці, передбачає внесення змін у зазначені файли.

В ОС *Windows XP* часові параметри протоколу *ARP* зберігаються у реєстрі системи. З цією метою у розділі реєстру **HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters** створені два параметри – **ArpCacheLife** та **ArpCacheMinReferenceLife**. Їх значення зазначаються в секундах. У параметрі **ArpCacheLife** міститься час існування *ARP*-записів, що не використовуються. За замовчуванням його значення дорівнює 2 хв. У параметрі **ArpCacheMinReferenceLife** – час існування *ARP*-записів, до яких звернення здійснюються часто. За замовчуванням його значення дорівнює 10 хв. В останніх версіях ОС *Windows* ці параметри усунуті із реєстру та застосовується інший, складніший механізм керування вмістом *ARP*-таблиці. Детальний опис механізму та його реалізації для ОС *Windows* можна отримати з документації, розміщеної на *Web*-сайті виробника.

В ОС *Linux* базові параметри керування *ARP*-таблицею, що необхідні для функціонування протоколу *ARP* згідно з *RFC-826*, зберігаються у кількох конфігураційних файлах, що містяться у шаблонному каталозі */proc/sys/net/ipv4/neigh/default/*. Зокрема, початковий тайм-аут існування записів у *ARP*-таблиці міститься у файлі */proc/sys/net/ipv4/neigh/default/gc\_stale\_time* і за замовчування його значення дорівнює 60 с. Для кожного з мережевих інтерфейсів на етапі встановлення ОС на базі шаблонного каталога створюється власний конфігураційний каталог */proc/sys/net/ipv4/neigh/if-name/* та власні конфігураційні файли, які надалі можна редагувати.

### КОМАНДИ МОНІТОРИНГУ СТАНУ ТА ОПЕРАЦІЙ З *ARP*-ТАБЛИЦЯМИ КОМУНІКАЦІЙНИХ ПРИСТРОЇВ *CISCO*

Моніторинг стану *ARP*-таблиць комунікаційних пристроїв *Cisco* здійснюється за допомогою команди **show arp** та похідних від неї команд **show arp detail**, **show arp dynamic**, **show arp static**, **show arp summary**, **show arp interface** тощо. Видалення записів з *ARP*-таблиць здійснюється за допомогою команди **clear arp** та похідних від неї команд. Додавання записів до *ARP*-таблиць здійснюється за допомогою команди **arp**.

Перелік основних команд **show arp**, **clear arp** та їх призначення наведено у таблиці 5.18. Синтаксис команди **arp** для випадку застосування протоколу *IP* версії 4 наведено нижче.

Таблиця 5.18 – Перелік основних команд **show arp** та **clear arp** комунікаційних пристроїв *Cisco*

| Команда                                                       | Призначення                                                                                           |
|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| <b>Команди show arp</b>                                       |                                                                                                       |
| <b>show arp</b>                                               | Виведення загальної <i>ARP</i> -таблиці пристрою                                                      |
| <b>show arp detail</b>                                        | Виведення <i>ARP</i> -таблиці пристрою у деталізованому вигляді                                       |
| <b>show arp dynamic</b>                                       | Виведення інформації про динамічні <i>ARP</i> -записи, що містяться у таблиці                         |
| <b>show arp static</b>                                        | Виведення інформації про статичні <i>ARP</i> -записи, що містяться у таблиці                          |
| <b>show arp summary</b>                                       | Виведення сумарної інформації про роботу протоколу <i>ARP</i> на пристрої                             |
| <b>show arp interface<br/>interface_type<br/>interface_id</b> | Виведення <i>ARP</i> -таблиці у розрізі окремого фізичного або логічного інтерфейсу                   |
| <b>show arp IP_address</b>                                    | Виведення <i>ARP</i> -таблиці для окремої <i>IP</i> -адреси або окремої <i>IP</i> -мережі             |
| <b>show arp ethernet<br/>interface-id</b>                     | Виведення <i>ARP</i> -таблиці у розрізі окремого інтерфейсу <i>Ethernet</i>                           |
| <b>Команди clear arp</b>                                      |                                                                                                       |
| <b>clear arp</b>                                              | Повне очищення (видалення всіх <i>ARP</i> -записів) <i>ARP</i> -таблиці пристрою                      |
| <b>clear arp IP_address</b>                                   | Видалення окремого (за <i>IP</i> -адресою) <i>ARP</i> -запису з <i>ARP</i> -таблиці пристрою          |
| <b>clear arp interface_type<br/>interface_id</b>              | Видалення <i>ARP</i> -записів з <i>ARP</i> -таблиці пристрою, які пов'язані із зазначеним інтерфейсом |

Синтаксис команди **arp** (режим глобального конфігурування):

```
arp IP_address hw_address arpa interface_type
interface_id,
```

де **IP\_address** – *IP*-адреса віддаленого вузла у десятковому записі;

**hw\_address** – *MAC*-адреса віддаленого вузла у вигляді **НННН.НННН.НННН**; кожне число **НННН** має довжину 2 байти і записується в шістнадцятковій формі.

**arpa** – службова конструкція, за допомогою якої на пристрої зазначається встановлення відповідностей між *IP*-адресами і *MAC*-адресами;

**interface\_type** – тип інтерфейсу (порту), може набувати значень **Ethernet**, **FastEthernet**, **GigabitEthernet**, **Port-channel**, **Vlan** тощо;

**interface\_id** – ідентифікатор інтерфейсу (порту), може мати однозначне позначення **number** (номер порту), або двочислове позначення **module/number** (номер модуля/номер порту).

## НАЛАГОДЖЕННЯ ПАРАМЕТРІВ ФУНКЦІОНУВАННЯ ПРОТОКОЛУ *ARP* НА КОМУНІКАЦІЙНИХ ПРИСТРОЯХ *CISCO*

Налагодження параметрів функціонування протоколу *ARP* на комунікаційних пристроях *Cisco* передбачає встановлення певних загальних параметрів (наприклад, тайм-ауту утримання записів в *ARP*-таблиці) та набору спеціалізованих параметрів (наприклад, параметрів розсилки повідомлень *ARP-Probe*). Основною командою, від якої походить решта команд для налагодження параметрів та засобів протоколу *ARP* у *Cisco IOS*, є команда **arp**. До переліку похідних від цієї команди належать такі команди, як: **arp arpa**, **arp authorized**, **arp frame-relay**, **arp log threshold entries**, **arp probe interval**, **arp snap**, **arp timeout**. Призначення та синтаксис згаданих команд наведено нижче.

Команди **arp arpa** та **arp snap** застосовуються для зазначення протоколу інкапсуляції у кадри канального рівня; **arp arpa** – звичайна інкапсуляція, **arp snap** – інкапсуляція із застосуванням заголовків кадрів *SNAP*. Команда **arp authorized** дозволяє застосовувати лише внутрішні авторизовані записи. Команда **arp frame-relay** активує застосування протоколу *ARP* для інтерфейсів технології *Frame Relay*. Команда **arp log threshold entries** призначена для налагодження параметрів підсистеми журналювання, які стосуються протоколу *ARP*. За допомогою команди **arp probe interval** здійснюється налагодження механізму *ARP-Probe*. Команда **arp timeout** застосовується для встановлення тайм-ауту утримання *ARP*-записів в *ARP*-таблиці пристрою.

Синтаксис команди **arp log threshold entries** (режим конфігурування інтерфейсу):

**arp log threshold entries threshold\_value,**

де **threshold\_value** – кількість записів, може змінюватися у діапазоні від 1 до 2147483647.

Синтаксис команди **arp probe interval** (режим конфігурування інтерфейсу):

**arp probe interval probe\_int\_value count count\_value,**

де **probe\_int\_value** – тривалість інтервалу для розсилки повідомлень *ARP-Probe*; зазначається у секундах, може змінюватися у діапазоні від 1 до 10 с;

**count** – службова конструкція, за допомогою якої зазначається кількість повідомлень, що посилаються у послідовності повідомлень *ARP-Probe*;

**count\_value** – кількість повідомлень у послідовності *ARP-Probe*; може змінюватися у діапазоні від 1 до 60;

Синтаксис команди **arp timeout** (режим конфігурування інтерфейсу):

**arp timeout arp\_timeout\_value,**

де **arp\_timeout\_value** – тривалість тайм-ауту протоколу *ARP*; зазначається у секундах, може змінюватися у діапазоні від 0 до 2147483; за замовчуванням дорівнює 14400 с (4 год).

## ПРОГРАМНІ РОЗРОБКИ НА БАЗІ ПРОТОКОЛУ *ARP*

На базі протоколу *ARP* розроблено утиліту **arping**, яка за функціоналом є подібною до утиліти **ping**. Проте її застосування обмежене лише каналним сегментом (широкомовним доменом), до якого належить вузол. Найуживанішим випадком застосування утиліти **arping** є перевірка доступності вузлів, на яких налагоджено блокування відповідей на ICMP-запити утиліти **ping**. Автором утиліти **arping** є Томас Хабетс (*Thomas Habets*). На основі його розробки реалізовано кілька відмітних за функціоналом варіантів утиліти. Проте найбільш функціональним варіантом є саме варіант автора. Зокрема, авторський варіант дає змогу перевіряти доступність вузла і за *MAC*-адресою.

Спочатку утиліту **arping** було розроблено лише для ОС *Unix/Linux*, нині наявні варіанти і для ОС *Windows*. У деяких ОС *Linux* (наприклад, ОС *Linux Microcore*) цю утиліту встановлено за замовчуванням, в інших (наприклад, ОС *Linux Debian*) – її можна встановити з репозиторію ОС. Актуальна інформація стосовно стану розробки утиліти **arping** знаходиться на Web-сторінці її автора за адресою <http://www.habets.pp.se/synscan/programs.php?prog=arping>.

Перелік параметрів для спрощеного варіанта утиліти **arping**, що застосовується в ОС *Linux*, наведено далі:

```
root@box:~# arping
BusyBox v1.19.0 (2011-08-14 21:05:38 UTC) multi-call binary.
Usage: arping [-fqbdUA] [-c CNT] [-w TIMEOUT] [-I IFACE] [-s SRC_IP] DST_IP
Send ARP requests/replies
 -f Quit on first ARP reply
 -q Quiet
 -b Keep broadcasting, don't go unicast
 -D Duplicated address detection mode
 -U Unsolicited ARP mode, update your neighbors
 -A ARP answer mode, update your neighbors
 -c N Stop after sending N ARP requests
 -w TIMEOUT Time to wait for ARP reply, seconds
 -I IFACE Interface to use (default eth0)
 -s SRC_IP Sender IP address
 DST_IP Target IP address
```

## МОДЕЛЬНИЙ ПРИКЛАД ДОСЛІДЖЕННЯ ФУНКЦІОНУВАННЯ ПРОТОКОЛУ *ARP* В ЛОКАЛЬНІЙ КОМП'ЮТЕРНІЙ МЕРЕЖІ

Розглянемо специфіку налагодження вузлів (робочих станцій) ОС *Windows* та ОС *Linux* для локальної комп'ютерної мережі, схему якої наведено на рисунку 5.13.

Таблиця 5.19 – Параметри інтерфейсів пристроїв для прикладу

| Пристрій              | Інтерфейс     | Підключення до пристрою | Підключення до інтерфейсу |
|-----------------------|---------------|-------------------------|---------------------------|
| Маршрутизатор R-1     | Gi0/1         | WAN                     | WAN Interface             |
|                       | Gi0/0         | Комутатор SW-1          | Gi0/1                     |
| Комутатор SW-1        | Fa0/1         | Робоча станція WS-A-1   | Fa0                       |
|                       | Fa0/2         | Робоча станція WS-A-2   | Fa0                       |
|                       | Fa0/3         | Робоча станція WS-A-3   | Fa0                       |
|                       | Fa0/4         | Робоча станція WS-A-4   | Fa0                       |
|                       | Gi0/1         | Маршрутизатор R-1       | Gi0/0                     |
| WAN                   | WAN Interface | Маршрутизатор R-1       | Gi0/1                     |
| Робоча станція WS-A-1 | Fa0           | Комутатор SW-1          | Fa0/1                     |
| Робоча станція WS-A-2 | Fa0           |                         | Fa0/2                     |
| Робоча станція WS-A-3 | Fa0           |                         | Fa0/3                     |
| Робоча станція WS-A-4 | Fa0           |                         | Fa0/4                     |

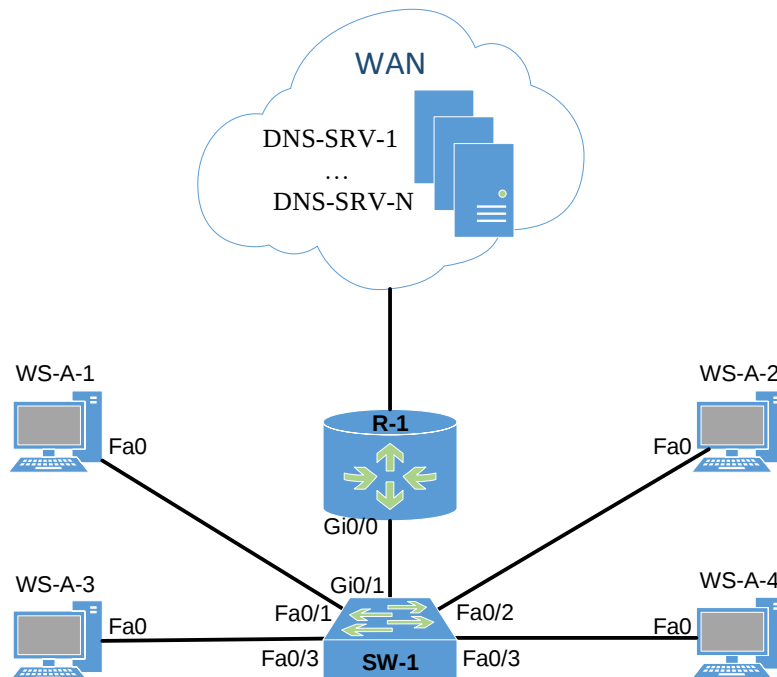


Рисунок 5.13 – Приклад мережі

Під час побудови цієї мережі для з'єднання пристроїв використано дані таблиці 5.19. Для налагодження параметрів адресації вузлів та комунікаційних пристроїв мережі використано дані таблиці 5.20.

Таблиця 5.20 – Параметри адресації мережі для прикладу

| Мережа/<br>Пристрій                        | Інтерфейс/Мережний<br>адаптер/Шлюз | MAC-адреса        | IP-адреса    | Маска         | Префікс |
|--------------------------------------------|------------------------------------|-------------------|--------------|---------------|---------|
| Мережа А                                   | –                                  | –                 | 195.10.1.0   | 255.255.255.0 | /24     |
| Маршрутизатор<br>R-1                       | Інтерфейс Gi0/0                    | CA-01-07-FE-00-08 | 195.10.1.254 | 255.255.255.0 | /24     |
|                                            | Інтерфейс Gi0/1                    | *                 | *            | *             | *       |
| Комутатор<br>SW-1                          | Інтерфейс Vlan 1                   | 00-D0-B1-E1-14-11 | 195.10.1.252 | 255.255.255.0 | /24     |
|                                            | Шлюз за замовчуванням              | –                 | 195.10.1.254 | –             | –       |
|                                            | Основний DNS-сервер                | –                 | 195.10.1.254 | –             | –       |
| Робоча станція<br>WS-A-1<br>(Windows 10)   | Мережний адаптер                   | 00-60-5C-16-8B-30 | 195.10.1.1   | 255.255.255.0 | /24     |
|                                            | Шлюз за замовчуванням              | –                 | 195.10.1.254 | –             | –       |
|                                            | Основний DNS-сервер                | –                 | 195.10.1.254 | –             | –       |
|                                            | Альтернат. DNS-сервер 1            | –                 | 8.8.8.8      | –             | –       |
|                                            | Альтернат. DNS-сервер 2            | –                 | 8.8.4.4      | –             | –       |
| Робоча станція<br>WS-A-2<br>(Windows 7)    | Мережний адаптер                   | 00-10-43-2C-BD-BB | 195.10.1.2   | 255.255.255.0 | /24     |
|                                            | Шлюз за замовчуванням              | –                 | 195.10.1.254 | –             | –       |
|                                            | Основний DNS-сервер                | –                 | 195.10.1.254 | –             | –       |
|                                            | Альтернат. DNS-сервер 1            | –                 | 8.8.8.8      | –             | –       |
|                                            | Альтернат. DNS-сервер 2            | –                 | 8.8.4.4      | –             | –       |
| Робоча станція<br>WS-A-3<br>(Linux Debian) | Мережний адаптер                   | 00-10-11-49-ED-09 | 195.10.1.3   | 255.255.255.0 | /24     |
|                                            | Шлюз за замовчуванням              | –                 | 195.10.1.254 | –             | –       |
|                                            | Основний DNS-сервер                | –                 | 195.10.1.254 | –             | –       |
|                                            | Альтернат. DNS-сервер 1            | –                 | 8.8.8.8      | –             | –       |
|                                            | Альтернат. DNS-сервер 2            | –                 | 8.8.4.4      | –             | –       |
| Робоча станція<br>WS-A-4<br>(Linux CentOS) | Мережний адаптер                   | 00-00-27-32-F9-E7 | 195.10.1.4   | 255.255.255.0 | /24     |
|                                            | Шлюз за замовчуванням              | –                 | 195.10.1.254 | –             | –       |
|                                            | Основний DNS-сервер                | –                 | 195.10.1.254 | –             | –       |
|                                            | Альтернат. DNS-сервер 1            | –                 | 8.8.8.8      | –             | –       |
|                                            | Альтернат. DNS-сервер 2            | –                 | 8.8.4.4      | –             | –       |

Примітка: \* – параметри адресації не зазначені.

Сценарій налагодження параметрів функціонування маршрутизатора мережі **R-1** наведено нижче:

```
Router>enable
Router#configure terminal
Router(config)#hostname R_1
R-1(config)#interface GigabitEthernet 0/0
R-1(config-if)#description LAN-A(LINK_TO_SW_1)
R-1(config-if)#ip address 195.10.1.254 255.255.255.0
R-1(config-if)#arp timeout 600
R-1(config-if)#no shutdown
R-1 (config-if)#exit
R-1 (config)#exit
R-1#
```

Сценарій налагодження параметрів функціонування комутатора мережі **SW-1** наведено нижче.

```
Switch>enable
Switch#configure terminal
Switch(config)#hostname SW_1
SW-1(config)#interface vlan 1
SW-1(config-if)#ip address 195.10.1.252 255.255.255.0
SW-1(config-if)#mac-address 00D0.B1E1.1411
SW-1(config-if)#arp timeout 600
SW-1(config-if)#no shutdown
SW-1(config-if)#exit
SW-1(config)#ip default-gateway 195.10.1.254
SW-1(config)#ip name-server 195.10.1.254
SW-1(config)#ip domain-name MY.NET
SW-1(config)#no ip domain-lookup
SW-1(config)#exit
SW-1#
```

Сценарій налагодження параметрів адресації робочої станції на базі ОС *Windows* та *Linux* були розглянуті в попередніх практичних завданнях.

### Результати виконання команд моніторингу встановлених відповідності між фізичними і логічними адресами для розглянутого прикладу

З метою перегляду інформації про налагоджені параметри фізичної і логічної адресації мережних адаптерів/інтерфейсів робочих станцій мережі для розглянутого прикладу використано команди ОС *Windows* **ipconfig** та ОС *Linux* **ifconfig**, **ip addr show**, для комутатора та маршрутизатора *Cisco* – команду *Cisco IOS* **show interfaces**. Для перевірки зв'язку між вузлами використано команди **ping** та **arping**. З метою перегляду встановлених відповідності між логічними і фізичними адресами на вузлах ОС *Windows* та ОС *Linux* використано команду **arp**, на комутаторі та маршрутизаторі – команду *Cisco IOS* **show arp**. Результати роботи цих команд для робочих станцій **WS-A-1 – WS-A-3**, комутатора **SW-1** та маршрутизатора **R-1** наведено далі:

```
SW-1#show interfaces vlan 1
Vlan1 is up, line protocol is up
 Hardware is Ethernet SVI, address is 00d0.b1e1.1411 (bia aabb.cc80.0100)
 Internet address is 195.10.1.252/24
 MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
 reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation ARPA, loopback not set
 Keepalive not supported
 ARP type: ARPA, ARP Timeout 00:10:00
 Last input never, output never, output hang never
 Last clearing of "show interface" counters never
```

```

Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 0 packets input, 0 bytes, 0 no buffer
 Received 0 broadcasts (0 IP multicasts)
R-1#show interfaces GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
 Hardware is i82543 (Livengood), address is ca01.07fe.0008 (bia ca01.07fe.0008)
 Description: LAN-A(LINK_TO_SW_1)
 Internet address is 195.10.1.254/24
 MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
 reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation ARPA, loopback not set
 Keepalive set (10 sec)
 Full-duplex, 1000Mb/s, link type is autonegotiation, media type is SX
 output flow-control is XON, input flow-control is XON
 ARP type: ARPA, ARP Timeout 00:10:00
 Last input 00:00:01, output 00:00:01, output hang never
 Last clearing of "show interface" counters never
 Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
 Queueing strategy: fifo
 Output queue: 0/40 (size/max)
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
 35 packets input, 4611 bytes, 0 no buffer
 Received 38 broadcasts, 0 runts, 0 giants, 0 throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored

```

```

SW-1#show arp

```

| Protocol | Address      | Age (min) | Hardware Addr  | Type | Interface |
|----------|--------------|-----------|----------------|------|-----------|
| Internet | 195.10.1.1   | 2         | 0060.5c16.8b30 | ARPA | Vlan1     |
| Internet | 195.10.1.2   | 0         | 0010.432c.bdbb | ARPA | Vlan1     |
| Internet | 195.10.1.3   | 2         | 0010.1149.ed09 | ARPA | Vlan1     |
| Internet | 195.10.1.4   | 2         | 0000.2732.f9e7 | ARPA | Vlan1     |
| Internet | 195.10.1.252 | -         | 00d0.b1e1.1411 | ARPA | Vlan1     |
| Internet | 195.10.1.244 | 2         | ca01.07fe.0008 | ARPA | Vlan1     |

```

C:\>arp -a

```

```

Интерфейс: 195.10.1.1 --- 0x2

```

| Адрес IP     | физический адрес  | Тип          |
|--------------|-------------------|--------------|
| 195.10.1.2   | 00-10-43-2c-bd-bb | динамический |
| 195.10.1.3   | 00-10-11-49-ed-09 | динамический |
| 195.10.1.4   | 00-00-27-32-f9-e7 | динамический |
| 195.10.1.252 | 00-d0-b1-e1-14-11 | динамический |
| 195.10.1.254 | ca-01-07-fe-00-08 | динамический |

```

C:\>arp -a

```

```

Интерфейс: 195.10.1.2 --- 0xb

```

| адрес в Интернете | физический адрес  | Тип          |
|-------------------|-------------------|--------------|
| 195.10.1.1        | 00-60-5c-16-8b-30 | динамический |
| 195.10.1.3        | 00-10-11-49-ed-09 | динамический |
| 195.10.1.4        | 00-00-27-32-f9-e7 | динамический |

|                 |                   |              |
|-----------------|-------------------|--------------|
| 195.10.1.252    | 00-d0-b1-e1-14-11 | динамический |
| 195.10.1.254    | ca-01-07-fe-00-08 | динамический |
| 195.10.1.255    | ff-ff-ff-ff-ff-ff | статический  |
| 224.0.0.22      | 01-00-5e-00-00-16 | статический  |
| 224.0.0.252     | 01-00-5e-00-00-fc | статический  |
| 255.255.255.255 | ff-ff-ff-ff-ff-ff | статический  |

```
root@debian8-3:~# arp -a
? (195.10.1.252) at 00:d0:b1:e1:14:11 [ether] on eth0
? (195.10.1.1) at 00:60:5c:16:8b:30 [ether] on eth0
? (195.10.1.2) at 00:10:43:2c:bd:bb [ether] on eth0
? (195.10.1.3) at 00:10:11:49:ed:09 [ether] on eth0
? (195.10.1.4) at 00:00:27:32:f9:e7 [ether] on eth0
? (195.10.1.252) at 00:d0:b1:e1:14:11 [ether] on eth0
? (195.10.1.254) at ca:01:07:fe:00:08 [ether] on eth0
```

```
root@debian8-3:~# arping -C 5 195.10.1.1
ARPING 195.10.1.1
60 bytes from 00:60:5c:16:8b:30 (195.10.1.1): index=0 time=1.001 sec
60 bytes from 00:60:5c:16:8b:30 (195.10.1.1): index=1 time=1.002 sec
60 bytes from 00:60:5c:16:8b:30 (195.10.1.1): index=2 time=1.001 sec
60 bytes from 00:60:5c:16:8b:30 (195.10.1.1): index=3 time=1.001 sec
60 bytes from 00:60:5c:16:8b:30 (195.10.1.1): index=4 time=1.002 sec
--- 195.10.1.1 statistics ---
5 packets transmitted, 5 packets received, 0% unanswered (0 extra)
rtt min/avg/max/std-dev = 1001.226/1001.487/1001.837/0.202 ms
```

## ПРАКТИЧНЕ ЗАВДАННЯ

**Мета роботи:** ознайомитися з основними правилами та протоколами встановлення відповідності між логічними і фізичними адресами в *IPv4*-мережах; ознайомитися з правилами встановлення відповідності для групових та широкомовних адрес; ознайомитися з деталями організації та функціонування протоколу *ARP*; отримати практичні навички побудови локальної мережі на базі комутатора *Ethernet* та навички моніторингу, діагностики та керування процесами встановлення відповідності між логічними і фізичними адресами в *IPv4*-мережах для вузлів ОС *Windows*, ОС *Linux* та комунікаційних пристроїв *Cisco*; дослідити процеси встановлення відповідності між логічними і фізичними адресами та процеси передачі повідомлень протоколу *ARP* у побудованій мережі.

### Порядок виконання роботи

1. Під'єднати комп'ютери та комунікаційне обладнання згідно проекту (рисунок 5.14) або створити проект середовищі програмного емулятора локальної комп'ютерної мережі, яка складається не менше ніж з п'яти вузлів (робочих станцій) ОС *Windows* та ОС *Linux*. Для вибору ОС вузла скористатися даними таблиці 5.21. Для побудованої мережі заповнити описову таблицю, аналогічну таблиці 5.19.

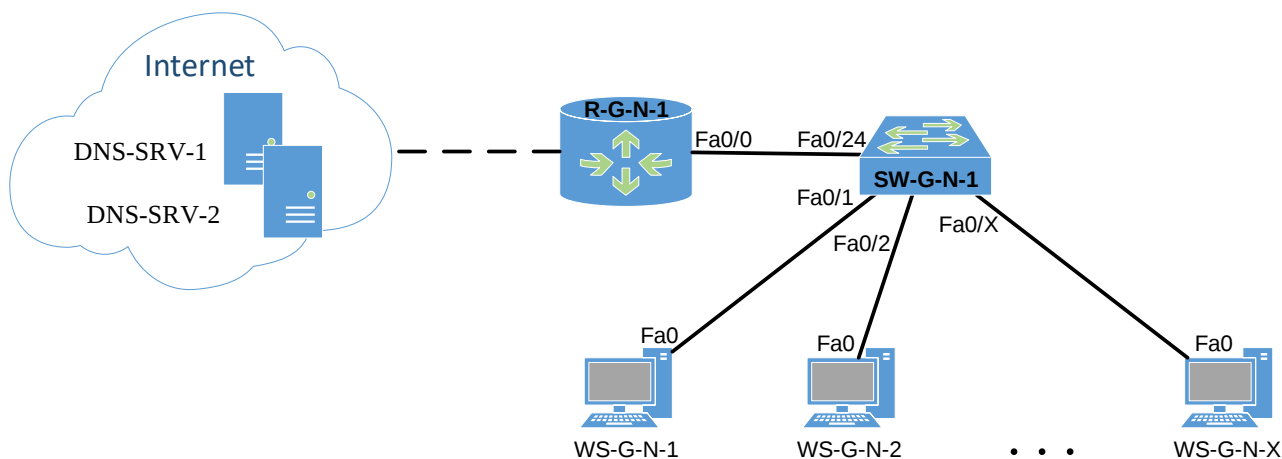


Рисунок 5.14 – Проект мережі

Таблиця 5.21 – Операційні системи вузлів (робочих станцій) ЛОМ

| №<br>варіанта | WS-G-N-01 | WS-G-N-02 | WS-G-N-03 | WS-G-N-04 | WS-G-N-05 | №<br>варіанта | WS-G-N-01 | WS-G-N-02 | WS-G-N-03 | WS-G-N-04 | WS-G-N-05 |
|---------------|-----------|-----------|-----------|-----------|-----------|---------------|-----------|-----------|-----------|-----------|-----------|
| 1             | W10       | W7        | LD/U      | LC/R      | LM/T      | 16            | LC/R      | W10       | W7        | LD/U      | LM/T      |
| 2             | W7        | W10       | LD/U      | LC/R      | LM/T      | 17            | W10       | LC/R      | W7        | LD/U      | LM/T      |
| 3             | W7        | LD/U      | W10       | LC/R      | LM/T      | 18            | W10       | W7        | LC/R      | LD/U      | LM/T      |
| 4             | W7        | LD/U      | LC/R      | W10       | LM/T      | 19            | W10       | W7        | LD/U      | LC/R      | LM/T      |
| 5             | W7        | LD/U      | LC/R      | LM/T      | W10       | 20            | W10       | W7        | LD/U      | LM/T      | LC/R      |
| 6             | W7        | W10       | LD/U      | LC/R      | LM/T      | 21            | LM/T      | W10       | W7        | LD/U      | LC/R      |
| 7             | W10       | W7        | LD/U      | LC/R      | LM/T      | 22            | W10       | LM/T      | W7        | LD/U      | LC/R      |
| 8             | W10       | LD/U      | W7        | LC/R      | LM/T      | 23            | W10       | W7        | LM/T      | LD/U      | LC/R      |
| 9             | W10       | LD/U      | LC/R      | W7        | LM/T      | 24            | W10       | W7        | LD/U      | LM/T      | LC/R      |
| 10            | W10       | LD/U      | LC/R      | LM/T      | W7        | 25            | W10       | W7        | LD/U      | LC/R      | LM/T      |
| 11            | LD/U      | W10       | W7        | LC/R      | LM/T      | 26            | W10       | W7        | LD/U      | LC/R      | LM/T      |
| 12            | W10       | LD/U      | W7        | LC/R      | LM/T      | 27            | W7        | W10       | LD/U      | LC/R      | LM/T      |
| 13            | W10       | W7        | LD/U      | LC/R      | LM/T      | 28            | LD/U      | W10       | W7        | LC/R      | LM/T      |
| 14            | W10       | W7        | LC/R      | LD/U      | LM/T      | 29            | LC/R      | W10       | W7        | LD/U      | LM/T      |
| 15            | W10       | W7        | LC/R      | LM/T      | LD/U      | 30            | LM/T      | W10       | W7        | LD/U      | LC/R      |

Примітка: W10 – ОС Windows 10; W7 – ОС Windows 7; LD/U – ОС Linux Debian/Ubuntu; LC/R – ОС Linux CentOS/Red Hat; LM/T – ОС Linux MicroCore/TinyCore.

2. Розробити схему адресації пристроїв (як кінцевих вузлів, так і комунікаційних пристроїв) мережі. Для цього скористатися даними таблиці 5.22. Під час розрахунку враховувати, що комутатору та інтерфейсу маршрутизатора мережі також виділяється по одній *IP*-адресі. Маску/префікс мережі визначити з урахуванням необхідності економії адрес. Результати навести у вигляді таблиці, аналогічної таблиці 5.20.

Таблиця 5.22 – Параметри для розрахунку

| № варіанта | IP-адреса мережі | Кількість робочих станцій у мережі | № варіанта | IP-адреса мережі | Кількість робочих станцій у мережі |
|------------|------------------|------------------------------------|------------|------------------|------------------------------------|
| 1          | 191.G.N.0        | 6                                  | 16         | 206.G.N.0        | 5                                  |
| 2          | 192.G.N.0        | 8                                  | 17         | 207.G.N.0        | 7                                  |
| 3          | 193.G.N.0        | 12                                 | 18         | 208.G.N.0        | 11                                 |
| 4          | 194.G.N.0        | 16                                 | 19         | 209.G.N.0        | 15                                 |
| 5          | 195.G.N.0        | 20                                 | 20         | 210.G.N.0        | 19                                 |
| 6          | 196.G.N.0        | 24                                 | 21         | 211.G.N.0        | 23                                 |
| 7          | 197.G.N.0        | 28                                 | 22         | 212.G.N.0        | 27                                 |
| 8          | 198.G.N.0        | 32                                 | 23         | 213.G.N.0        | 31                                 |
| 9          | 199.G.N.0        | 36                                 | 24         | 214.G.N.0        | 39                                 |
| 10         | 200.G.N.0        | 40                                 | 25         | 215.G.N.0        | 47                                 |
| 11         | 201.G.N.0        | 48                                 | 26         | 216.G.N.0        | 55                                 |
| 12         | 202.G.N.0        | 56                                 | 27         | 217.G.N.0        | 63                                 |
| 13         | 203.G.N.0        | 64                                 | 28         | 218.G.N.0        | 71                                 |
| 14         | 204.G.N.0        | 72                                 | 29         | 219.G.N.0        | 78                                 |
| 15         | 205.G.N.0        | 80                                 | 30         | 220.G.N.0        | 87                                 |

3. Сформувати повідомлення *ARP*-запити, що надсилаються з робочої станції (таблиця 5.23) до комутатора, маршрутизатора та решти станцій мережі для формування адресних відповідностей. Сформувати повідомлення *ARP*-відповіді, що надходять від вузлів-відповідачів. *ARP*-запити та *ARP*-відповіді показати як такі, що інкапсульовані у кадри Ethernet. Побудувати *ARP*-таблицю робочої станції після надходження *ARP*-відповідей.

Таблиця 5.23 – Параметри для виконання п. 3

| № варіанта | Робоча станція | № варіанта | Робоча станція | № варіанта | Робоча станція |
|------------|----------------|------------|----------------|------------|----------------|
| 1          | WS-G-N-1       | 11         | WS-G-N-1       | 21         | WS-G-N-1       |
| 2          | WS-G-N-2       | 12         | WS-G-N-2       | 22         | WS-G-N-2       |
| 3          | WS-G-N-3       | 13         | WS-G-N-3       | 23         | WS-G-N-3       |
| 4          | WS-G-N-4       | 14         | WS-G-N-4       | 24         | WS-G-N-4       |
| 5          | WS-G-N-5       | 15         | WS-G-N-5       | 25         | WS-G-N-5       |
| 6          | WS-G-N-1       | 16         | WS-G-N-1       | 26         | WS-G-N-1       |
| 7          | WS-G-N-2       | 17         | WS-G-N-2       | 27         | WS-G-N-2       |
| 8          | WS-G-N-3       | 18         | WS-G-N-3       | 28         | WS-G-N-3       |
| 9          | WS-G-N-4       | 19         | WS-G-N-4       | 29         | WS-G-N-4       |
| 10         | WS-G-N-5       | 20         | WS-G-N-5       | 30         | WS-G-N-5       |

4. Провести налагодження параметрів іменування та параметрів *IP*-адресації мережних адаптерів/інтерфейсів пристроїв мережі згідно з даними пункту 2, рисунка 5.14 та таблиці 5.24.

Таблиця 5.24 – Дані для визначення параметрів адресації мережі

| № варі-<br>анта | <i>IP-адреса шлюзу<br/>за замовчуванням, IP-адреса<br/>основного DNS-сервера</i> | <i>IP-адреса<br/>альтернативного<br/>DNS-сервера 1</i> | <i>IP-адреса<br/>альтернативного<br/>DNS-сервера 2</i> |
|-----------------|----------------------------------------------------------------------------------|--------------------------------------------------------|--------------------------------------------------------|
| 1               | Перша <i>IP-адреса</i> діапазону                                                 | Level3 Communications                                  | Level3 Communications                                  |
| 2               | Остання <i>IP-адреса</i> діапазону                                               | Google                                                 | Google                                                 |
| 3               | Перша <i>IP-адреса</i> діапазону                                                 | OpenDNS Home                                           | OpenDNS Home                                           |
| 4               | Остання <i>IP-адреса</i> діапазону                                               | Securly                                                | Securly                                                |
| 5               | Перша <i>IP-адреса</i> діапазону                                                 | Comodo Secure DNS                                      | Comodo Secure DNS                                      |
| 6               | Остання <i>IP-адреса</i> діапазону                                               | DNS Advantage                                          | DNS Advantage                                          |
| 7               | Перша <i>IP-адреса</i> діапазону                                                 | Norton ConnectSafe                                     | Norton ConnectSafe                                     |
| 8               | Остання <i>IP-адреса</i> діапазону                                               | SafeDNS                                                | SafeDNS                                                |
| 9               | Перша <i>IP-адреса</i> діапазону                                                 | OpenNIC                                                | OpenNIC                                                |
| 10              | Остання <i>IP-адреса</i> діапазону                                               | Public-Root                                            | Public-Root                                            |
| 11              | Перша <i>IP-адреса</i> діапазону                                                 | Level3 Com-ns                                          | Level3 Com-ns                                          |
| 12              | Остання <i>IP-адреса</i> діапазону                                               | Google                                                 | Google                                                 |
| 13              | Перша <i>IP-адреса</i> діапазону                                                 | OpenDNS Home                                           | OpenDNS Home                                           |
| 14              | Остання <i>IP-адреса</i> діапазону                                               | Securly                                                | Securly                                                |
| 15              | Перша <i>IP-адреса</i> діапазону                                                 | Comodo Secure DNS                                      | Comodo Secure DNS                                      |
| 16              | Остання <i>IP-адреса</i> діапазону                                               | DNS Advantage                                          | DNS Advantage                                          |
| 17              | Перша <i>IP-адреса</i> діапазону                                                 | Norton ConnectSafe                                     | Norton ConnectSafe                                     |
| 18              | Остання <i>IP-адреса</i> діапазону                                               | SafeDNS                                                | SafeDNS                                                |
| 19              | Перша <i>IP-адреса</i> діапазону                                                 | OpenNIC                                                | OpenNIC                                                |
| 20              | Остання <i>IP-адреса</i> діапазону                                               | Public-Root                                            | Public-Root                                            |
| 21              | Перша <i>IP-адреса</i> діапазону                                                 | Level3 Com-ns                                          | Level3 Com-ns                                          |
| 22              | Остання <i>IP-адреса</i> діапазону                                               | Google                                                 | Google                                                 |
| 23              | Перша <i>IP-адреса</i> діапазону                                                 | OpenDNS Home                                           | OpenDNS Home                                           |
| 24              | Остання <i>IP-адреса</i> діапазону                                               | Securly                                                | Securly                                                |
| 25              | Перша <i>IP-адреса</i> діапазону                                                 | Comodo Secure DNS                                      | Comodo Secure DNS                                      |
| 26              | Остання <i>IP-адреса</i> діапазону                                               | DNS Advantage                                          | DNS Advantage                                          |
| 27              | Перша <i>IP-адреса</i> діапазону                                                 | Norton ConnectSafe                                     | Norton ConnectSafe                                     |
| 28              | Остання <i>IP-адреса</i> діапазону                                               | SafeDNS                                                | SafeDNS                                                |
| 29              | Перша <i>IP-адреса</i> діапазону                                                 | OpenNIC                                                | OpenNIC                                                |
| 30              | Остання <i>IP-адреса</i> діапазону                                               | Public-Root                                            | Public-Root                                            |

5. Провести налагодження тайм-ауту утримання *ARP*-записів в *ARP*-таблицях пристроїв мережі. Для вибору значення тайм-ауту скористатися даними таблиці 5.25 (необов'язково).

Таблиця 5.25 – Параметри для виконання

| № варіанта | Тайм-аут, хв | № варіанта | Тайм-аут, хв | № варіанта | Тайм-аут, хв |
|------------|--------------|------------|--------------|------------|--------------|
| 1          | 5            | 11         | 3            | 21         | 12           |
| 2          | 10           | 12         | 5            | 22         | 16           |
| 3          | 15           | 13         | 8            | 23         | 20           |
| 4          | 20           | 14         | 10           | 24         | 3            |
| 5          | 6            | 15         | 12           | 25         | 5            |
| 6          | 11           | 16         | 15           | 26         | 7            |
| 7          | 16           | 17         | 20           | 27         | 9            |
| 8          | 21           | 18         | 25           | 28         | 11           |
| 9          | 7            | 19         | 4            | 29         | 13           |
| 10         | 12           | 20         | 8            | 30         | 15           |

6. Перевірити можливість інформаційного обміну між робочою станцією (таблиця 5.23 та рештою робочих станцій та комунікаційних пристроїв мережі за допомогою команд **ping** та **arping** (за можливості).

7. Вивести *ARP*-таблицю робочої станції (таблиця 5.23) та порівняти її з отриманою у п. 3. Вивести *ARP*-таблиці решти пристроїв мережі.

8. Очистити *ARP*-таблиці всіх робочих станцій та комунікаційних пристроїв мережі. Ввести статичні *ARP*-записи у *ARP*-таблиці всіх робочих станцій та комунікаційних пристроїв мережі.

9. Для заданої *IP*-адреси версії 4 (таблиця 5.26) визначити *MAC*-адресу групової розсилки. За можливості визначити, повідомлення якого протоколу передається за допомогою даної адреси.

Таблиця 5.26 – Параметри для розрахунку

| № варіанта | <i>IP</i> -адреса | № варіанта | <i>IP</i> -адреса |
|------------|-------------------|------------|-------------------|
| 1          | 224.0.0.1         | 16         | 224.0.0.10        |
| 2          | 224.0.0.2         | 17         | 224.15.0.65       |
| 3          | 224.20.75.25      | 18         | 224.0.0.11        |
| 4          | 224.0.0.4         | 19         | 224.35.0.85       |
| 5          | 224.40.115.45     | 20         | 224.0.0.12        |
| 6          | 224.0.0.5         | 21         | 224.0.0.13        |
| 7          | 224.60.155.85     | 22         | 224.0.0.18        |
| 8          | 224.0.0.6         | 23         | 224.75.20.135     |
| 9          | 224.80.195.125    | 24         | 224.0.0.22        |
| 10         | 224.0.0.7         | 25         | 224.95.40.155     |
| 11         | 224.100.0.5       | 26         | 224.0.0.102       |
| 12         | 224.0.0.8         | 27         | 224.215.60.165    |
| 13         | 224.0.0.107       | 28         | 224.0.0.251       |
| 14         | 224.0.0.9         | 29         | 224.235.80.185    |
| 15         | 224.140.0.45      | 30         | 224.0.1.1         |

10. Для заданої *MAC*-адреси групової розсилки (таблиця 5.27) визначити *IP*-адресу (можливі *IP*-адреси) групової розсилки. За можливості визначити,

повідомлення якого протоколу передається у кадрі з такою адресою (за можливості).

Таблиця 5.27 – Параметри для розрахунку

| № варіанта | MAC-адреса        | № варіанта | MAC-адреса        |
|------------|-------------------|------------|-------------------|
| 1          | 01-00-5E-00-AF-B1 | 16         | 01-00-5E-10-CC-B2 |
| 2          | 01-00-5E-01-AE-B3 | 17         | 01-00-5E-11-CB-B4 |
| 3          | 01-00-5E-02-AD-B7 | 18         | 01-00-5E-12-CA-B6 |
| 4          | 01-00-5E-03-AC-B9 | 19         | 01-00-5E-13-DF-B8 |
| 5          | 01-00-5E-04-AB-BB | 20         | 01-00-5E-14-DE-BA |
| 6          | 01-00-5E-05-AA-BE | 21         | 01-00-5E-15-DD-BC |
| 7          | 01-00-5E-06-BF-BF | 22         | 01-00-5E-16-DC-BE |
| 8          | 01-00-5E-07-BE-A1 | 23         | 01-00-5E-17-DB-A2 |
| 9          | 01-00-5E-08-BD-A3 | 24         | 01-00-5E-18-DA-A4 |
| 10         | 01-00-5E-09-BC-A5 | 25         | 01-00-5E-19-EF-A6 |
| 11         | 01-00-5E-0A-BB-A7 | 26         | 01-00-5E-1A-EE-A8 |
| 12         | 01-00-5E-0B-BA-A9 | 27         | 01-00-5E-1B-ED-AA |
| 13         | 01-00-5E-0C-CF-AB | 28         | 01-00-5E-1C-EC-AC |
| 14         | 01-00-5E-0D-CE-AE | 29         | 01-00-5E-1D-EB-AF |
| 15         | 01-00-5E-0E-CD-AF | 30         | 01-00-5E-1E-EA-FF |

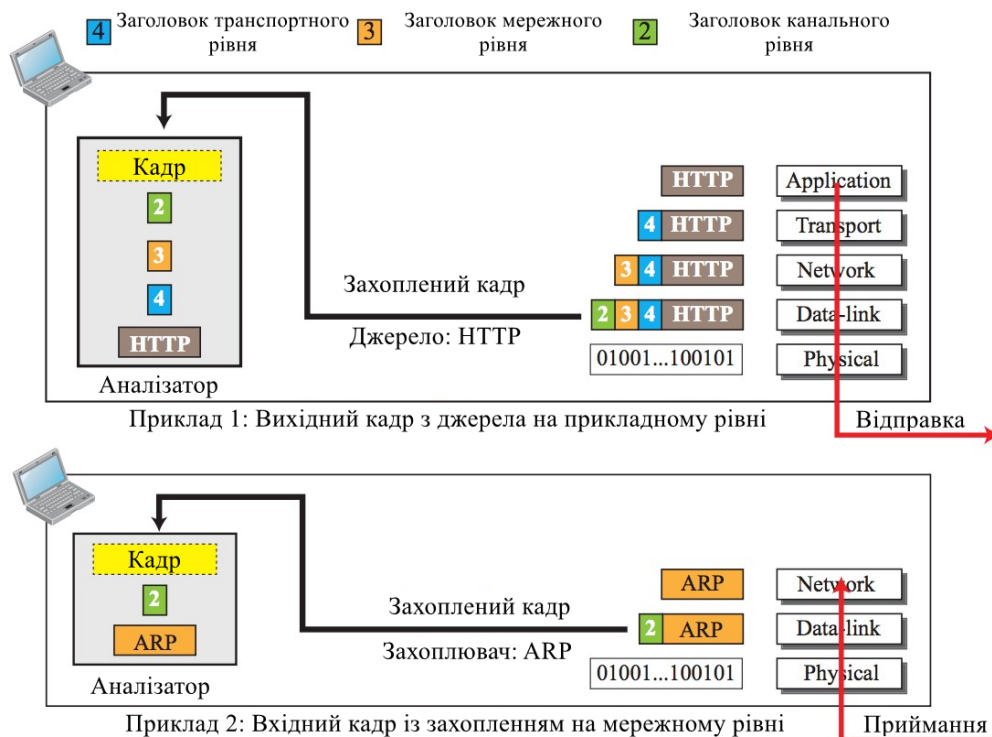
### КОНТРОЛЬНІ ПИТАННЯ

1. Протоколи та правила встановлення відповідностей між фізичними і логічними адресами в *IP*-мережах за умови застосування *IP*-адрес версії 4.
2. Правило формування групових *MAC*-адрес на основі групових *IP*-адрес версії 4.
3. Правило формування широкомовних *MAC*-адрес на основі широкомовних *IP*-адрес версії 4.
4. Загальна характеристика протоколу *ARP*.
5. Стандартизація протоколу *ARP*.
6. Характеристики протоколу *ARP* стосовно моделі *OSI* та стека *TCP/IP*.
7. Структура повідомлення протоколу *ARP*.
8. Види повідомлень протоколу *ARP*.
9. Структура *ARP*-таблиці.
10. Джерела заповнення *ARP*-таблиці.
11. Алгоритми роботи протоколу *ARP*.
12. Призначення та синтаксис команди **arp**.
13. Призначення та синтаксис команди **arping**.
14. Команди моніторингу стану *ARP*-таблиць комунікаційних пристроїв *Cisco*.
15. Команди операцій з *ARP*-таблицями комунікаційних пристроїв *Cisco*.

### 5.1.6 ДОСЛІДЖЕННЯ RDU ПАКЕТІВ

## БАЗОВІ ПОНЯТТЯ З ЗАХОПЛЕННЯ МЕРЕЖЕВИХ ПАКЕТІВ

Одним із способів отримати глибше розуміння мережевих концепцій це використання сніффера пакетів для перехоплення та аналізу пакетів. Сніффер пакетів – це частина програмного забезпечення, яка повинна працювати паралельно з програмою, чії пакети потрібно проаналізувати. Пакети можна аналізувати на чотирьох рівнях: прикладному, транспортному, мережевому і каналному. При цьому потрібно пам'ятати що пакети верхніх рівнів інкапсулюються в пакети нижніх рівнів і деінкапсулюються в зворотному напрямку (рисунком 5.15). Це означає, що нам потрібно перехоплювати лише вихідні або вхідні кадри; програмне забезпечення для аналізу пакетів може виділити з цих кадрів пакети будь-якого рівня, які потрібно проаналізувати. З цієї причини програмне забезпечення для перехоплення пакетів зазвичай складається з двох компонентів: перехоплювача пакетів і аналізатора пакетів. Захоплювач пакетів перехоплює копії всіх вихідних і вхідних кадрів (на каналному рівні) і передає їх аналізатору пакетів. Потім аналізатор пакетів може виділити різні заголовки і кінцеве повідомлення для аналізу.



**Рисунок 5.15 – Захоплення кадрів та аналіз пакетів у сніфері пакетів**

У прикладі I на рисунку 5.15 перехоплено вихідний кадр. Джерелом кадру є протокол *HTTP* на прикладному рівні. Копія кадру передається аналізатору. Аналізатор витягує загальну інформацію, що міститься у кадрі (поле з позначкою кадр), заголовки 2, 3 і 4, а також *HTTP*-повідомлення для аналізу. У прикладі II

перехоплюється вхідний кадр. Поток (кінцевим пунктом призначення) є протокол *ARP* на мережевому рівні. Копія кадру передається аналізатору. Аналізатор витягує загальну інформацію, що міститься в заголовку (поле з позначкою кадр), заголовку 2 і *ARP*-повідомленні для аналізу.

## WIRESHARK

Програма аналізу пакетів *Wireshark*. *Wireshark* (раніше відома як *ETHERREAL*) – це безкоштовний сніффер/аналізатор пакетів, доступний як для *UNIX*-подібних (*Unix*, *Linux*, *Mac OS X*, *BSD* і *Solaris*), так і для *Windows* операційних систем. Він перехоплює пакети з мережевого інтерфейсу і відображає їх з детальною інформацією про протокол *Wireshark*, однак, є пасивним аналізатором. Він лише перехоплює пакети, не маніпулюючи ними; він не надсилає пакети в мережу і не виконує інших активних операцій. *Wireshark* також не є інструментом виявлення вторгнень. Він не попереджає про вторгнення в мережу. Тим не менш, він може допомогти мережевим адміністраторам з'ясувати, що відбувається всередині мережі, і усунути мережеві проблеми. Крім того, що *Wireshark* є незамінним інструментом для мережеских адміністраторів, він є цінним інструментом для розробників протоколів, які можуть використовувати його для налагодження реалізацій протоколів. Це також чудовий навчальний інструмент для студентів, що вивчають комп'ютерні мережі, які можуть використовувати його, щоб бачити деталі роботи протоколу в реальному часі.

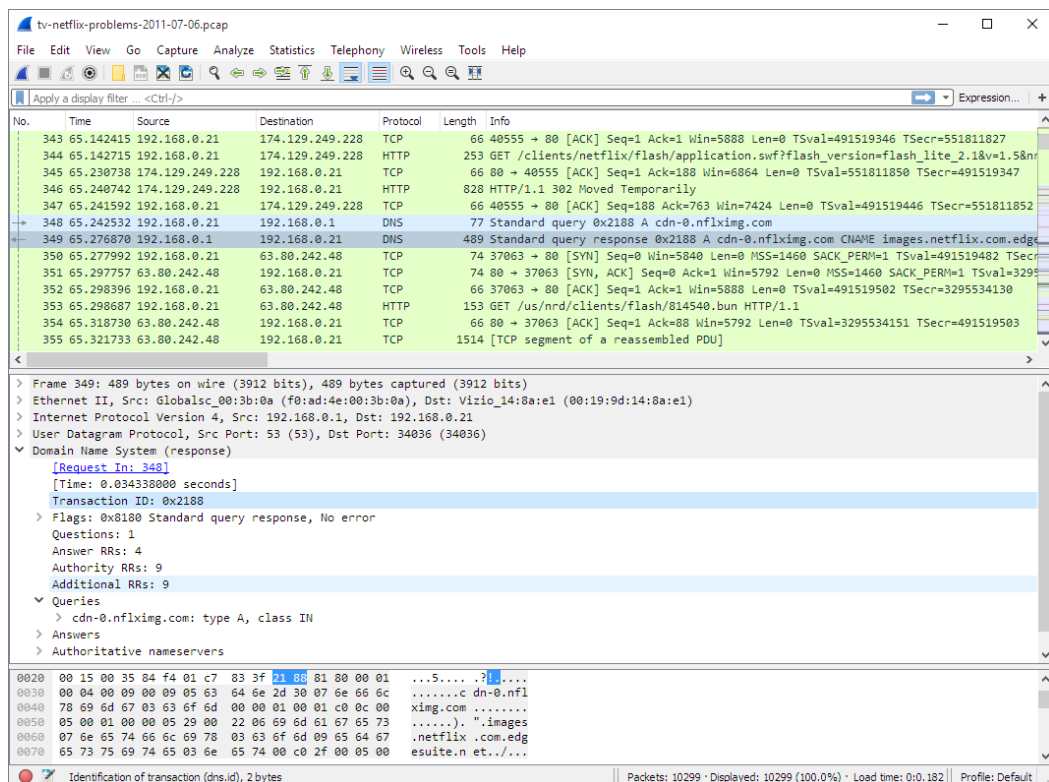


Рисунок 5.16 – *Wireshark* перехоплює пакети і дозволяє вивчати їх вміст

Головне вікно *Wireshark* складається з частин, які добре відомі з багатьох інших програм з графічним інтерфейсом:

1. **Меню** використовується для запуску дій.
2. **Головна панель інструментів** забезпечує швидкий доступ до часто використовуваних пунктів меню.
3. **Панель фільтрів** дозволяє користувачам встановлювати фільтри відображення, щоб відфільтрувати пакети, які відображаються.
4. **Панель списку пакетів** відображає короткий опис кожного перехопленого пакета. Натискаючи на пакети на цій панелі, ви керуєте тим, що відображається на двох інших панелях.
5. На **панелі деталі пакета** буде показано пакет, що вибраний на панелі списку пакетів, з докладнішими відомостями.
6. **Панель байтів пакета** відображає дані пакета, вибраного на панелі списку пакетів, і виділяє поле, вибране на панелі деталей пакета.
7. На **панелі діаграми пакета** показано пакет, вибраний у списку пакунків, у вигляді діаграми у стилі підручника.
8. **Рядок стану** показує деяку детальну інформацію про поточний стан програми і перехоплені дані.

Панель списку пакетів відображає усі пакети у поточному файлі перехоплення (рисунок 5.17).

| No. | Time     | Source       | Destination  | Protocol | Length | Info                                                                                    |
|-----|----------|--------------|--------------|----------|--------|-----------------------------------------------------------------------------------------|
| 1   | 0.000000 | 192.168.0.21 | 192.168.0.1  | DNS      | 84     | Standard query 0x403d A moviecontrol.netflix.com                                        |
| 2   | 0.055880 | 192.168.0.1  | 192.168.0.21 | DNS      | 479    | Standard query response 0x403d A moviecontrol.netflix.com CNAME nccp-moviecontrol-fro   |
| 3   | 0.057690 | 192.168.0.21 | 50.17.249.22 | TCP      | 74     | 37314->443 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=491454310 TSecr=0 WS=1 |
| 4   | 0.154716 | 50.17.249.22 | 192.168.0.21 | TCP      | 74     | 443->37314 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=2102931926  |
| 5   | 0.155962 | 192.168.0.21 | 50.17.249.22 | TCP      | 66     | 37314->443 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=491454408 TSecr=2102931926            |
| 6   | 0.163169 | 192.168.0.21 | 50.17.249.22 | TLSv1    | 187    | Client Hello                                                                            |
| 7   | 0.250734 | 50.17.249.22 | 192.168.0.21 | TCP      | 66     | 443->37314 [ACK] Seq=1 Ack=122 Win=5792 Len=0 TSval=2102931950 TSecr=491454416          |
| 8   | 0.252716 | 50.17.249.22 | 192.168.0.21 | TLSv1    | 1514   | Server Hello                                                                            |
| 9   | 0.253826 | 192.168.0.21 | 50.17.249.22 | TCP      | 66     | 37314->443 [ACK] Seq=122 Ack=1449 Win=8768 Len=0 TSval=491454507 TSecr=2102931950       |
| 10  | 0.254730 | 50.17.249.22 | 192.168.0.21 | TCP      | 1514   | [TCP segment of a reassembled PDU]                                                      |
| 11  | 0.254778 | 50.17.249.22 | 192.168.0.21 | TLSv1    | 349    | Certificate                                                                             |
| 12  | 0.255853 | 192.168.0.21 | 50.17.249.22 | TCP      | 66     | 37314->443 [ACK] Seq=122 Ack=2897 Win=11648 Len=0 TSval=491454509 TSecr=2102931950      |
| 13  | 0.256102 | 192.168.0.21 | 50.17.249.22 | TCP      | 66     | 37314->443 [ACK] Seq=122 Ack=3180 Win=14528 Len=0 TSval=491454509 TSecr=2102931950      |
| 14  | 0.319870 | 192.168.0.21 | 50.17.249.22 | TLSv1    | 264    | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message                    |
| 15  | 0.411795 | 50.17.249.22 | 192.168.0.21 | TLSv1    | 125    | Change Cipher Spec, Encrypted Handshake Message                                         |

Рисунок 5.17 – Панель «Список пакетів»

Кожен рядок у списку пакетів відповідає одному пакету у файлі перехоплення. Якщо ви виберете рядок на цій панелі, буде показано докладнішу інформацію на панелях **Деталі пакета** і **Байти пакета**.

Під час розбирання пакета *Wireshark* розмістить у стовпчиках інформацію, отриману від аналізаторів протоколів. Оскільки протоколи вищих рівнів можуть перезаписувати інформацію з нижчих рівнів, ви зазвичай бачитимете інформацію лише з найвищого можливого рівня.

Панель **Деталей пакета** показує поточний пакет (вибраний на панелі **Список пакетів**) у детальнішому вигляді (рисунок 5.18).

На цій панелі показано протоколи і поля протоколів пакета, вибраного на панелі **Список пакетів**. Рядки протоколу (мітки піддерева) і поля пакета показано у вигляді дерева, яке можна розгортати і згортати. Деякі поля протоколу мають спеціальне значення.

```
> Ethernet II, Src: Globalsc_00:3b:0a (f0:ad:4e:00:3b:0a), Dst: Vizio_14:8a:e1 (00:19:9d:14:8a:e1)
> Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.21
> User Datagram Protocol, Src Port: 53 (53), Dst Port: 34036 (34036)
▼ Domain Name System (response)
 [Request In: 1]
 [Time: 0.055880000 seconds]
 Transaction ID: 0x403d
 > Flags: 0x8180 Standard query response, No error
 Questions: 1
 Answer RRs: 2
 Authority RRs: 8
 Additional RRs: 8
 > Queries
 > Answers
 > Authoritative nameservers
 > Additional records
```

Рисунок 5.18 – Панель «Деталі пакета»

*Wireshark* сам генерує додаткову інформацію про протокол, яка відсутня у перехоплених даних. Цю інформацію взято у квадратні дужки («[» and «]»). Згенерована інформація включає час відгуку, аналіз TCP, інформацію про геолокацію IP-адреси і перевірку контрольної суми.

Якщо *Wireshark* виявляє зв'язок з іншим пакетом у файлі перехоплення, він створить посилання на цей пакет. Посилання підкреслюються і відображаються синім кольором. Якщо ви двічі клацнете по посиланню, *Wireshark* перейде до відповідного пакета.

Панель **Байти пакета** показує дані поточного пакета (вибраного на панелі **Список пакетів**) у форматі hexdump (рисунок 5.19).

```
0000 00 19 9d 14 8a e1 f0 ad 4e 00 3b 0a 08 00 45 00 N.;...E.
0010 01 d1 00 00 40 00 40 11 b7 b5 c0 a8 00 01 c0 a8 @.@.
0020 00 15 00 35 84 f4 01 bd 83 35 40 3d 81 80 00 01 ...5.... .5@=...
0030 00 02 00 08 00 08 0c 6d 6f 76 69 65 63 6f 6e 74 m oviecont
0040 72 6f 6c 07 6e 65 74 66 6c 69 78 03 63 6f 6d 00 rol.netf lix.com.
0050 00 01 00 01 c0 0c 00 05 00 01 00 00 00 2d 00 40 -.@
0060 25 6e 63 63 70 2d 6d 6f 76 69 65 63 6f 6e 74 72 %ncsp-mo viecontr
0070 6f 6c 2d 66 72 6f 6e 74 65 6e 64 2d 31 37 31 32 ol-front end-1712
0080 31 38 38 39 32 31 09 75 73 2d 65 61 73 74 2d 31 188921.u s-east-1
0090 03 65 6c 62 09 61 6d 61 7a 6f 6e 61 77 73 c0 21 .elb.ama zonaws.!
```

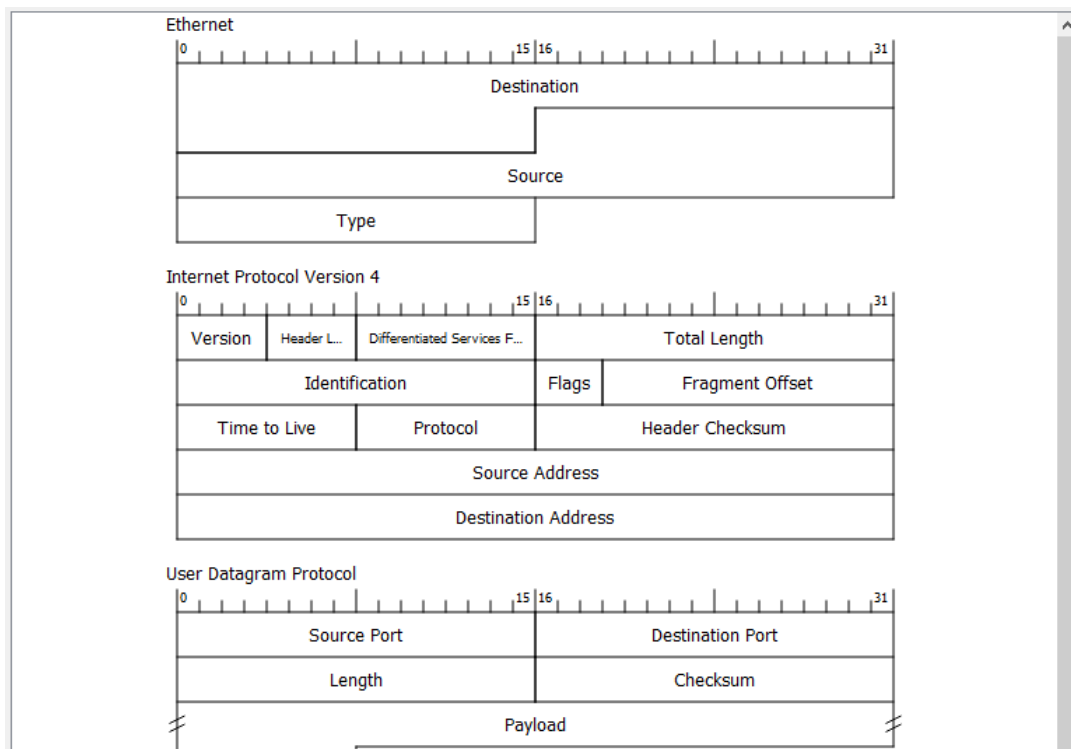
Рисунок 5.19 – Панель «Байти пакета»

На панелі **Байти пакета** показано канонічний шістнадцятковий дамп даних пакета. Кожен рядок містить зміщення даних, шістнадцять шістнадцяткових байт і шістнадцять байт ASCII. Байти, які не можна друкувати, замінюються крапкою («.»).

Залежно від даних пакета, іноді може бути доступно більше однієї сторінки, наприклад, коли *Wireshark* зібрав деякі пакети в один фрагмент даних. У цьому випадку можна побачити кожне джерело даних, натиснувши відповідну вкладку в нижній частині панелі.

Додаткові вкладки зазвичай містять дані, зібрані з декількох пакетів, або розшифровані дані.

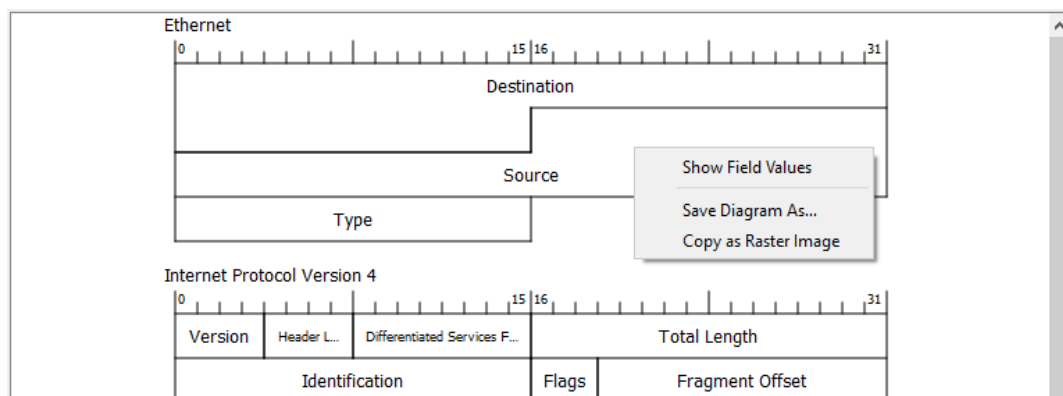
Панель **Діаграми пакетів** показує поточний пакет (вибраний на панелі «Список пакетів») у вигляді діаграми, подібної до тих, що використовуються у підручниках і RFC IETF (рисунок 5.20).



**Рисунок 5.20 – Панель «Діаграма пакета»**

На цій панелі показано протоколи і поля протоколів верхнього рівня пакета, вибраного на панелі «Список пакетів», у вигляді серії діаграм.

Також доступне контекстне меню (клацання правою кнопкою миші (рисунок 5.21)).



**Рисунок 5.21– Меню панелі «Діаграма пакетів»**

## РОБОТА ІЗ ЗАСТОСУНКОМ *WIRESHARK*

Захоплення мережових даних в реальному часі є однією з основних функцій *Wireshark*.

Механізм перехоплення *Wireshark* надає наступні можливості:

- Захоплення з різних типів мережевого обладнання, таких як *Ethernet* або 802.11.
- Одночасне перехоплення з декількох мережевих інтерфейсів.
- Зупинка перехоплення за різними тригерами, такими як обсяг перехоплених даних, час, що минув, або кількість пакетів.
- Одночасне відображення розшифрованих пакетів під час перехоплення.
- Фільтр пакетів, зменшуючи кількість перехоплених даних.
- Зберігання пакетів у декількох файлах під час тривалого перехоплення.

Налаштування *Wireshark* для перехоплення пакетів вперше може бути складним. Нижче наведено кілька моментів на які потрібно звернути увагу:

- можуть знадобитися спеціальні привілеї для запуску перехоплення в реальному часі;
- потрібно вибрати правильний мережевий інтерфейс для перехоплення пакетних даних;
- треба здійснювати перехоплення в потрібному місці мережі, щоб побачити трафік, який ви очікуєте побачити.

### Запуск перехоплення

Почати перехоплення пакетів за допомогою *Wireshark* можна наступними способами:

- Двічі клацнути на інтерфейсі на екрані привітання.
- Вибрати інтерфейс на екрані привітання, а потім вибрати **Capture** → **Start** або натиснути першу кнопку на панелі інструментів.

### Розділ Capture на екрані привітання

Коли відкриваєте *Wireshark* без запуску захоплення або відкриття файлу захоплення, програма відображає «Екран привітання», на якому перераховані всі нещодавно відкриті файли захоплення і доступні інтерфейси захоплення. Мережева активність для кожного інтерфейсу буде показана у вигляді рядка поруч з назвою інтерфейсу. Можна вибрати декілька інтерфейсів і виконувати перехоплення з них одночасно (рисунки 5.22).

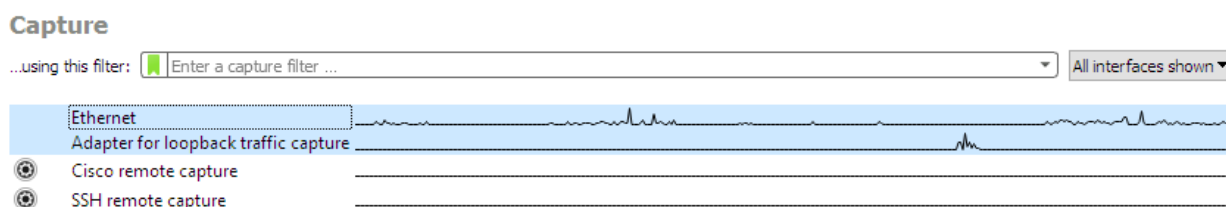


Рисунок 5.22 – Інтерфейси перехоплення у *Microsoft Windows*

Деякі інтерфейси дозволяють або вимагають конфігурації перед захопленням. На це вказуватиме піктограма конфігурації (⚙) ліворуч від назви

інтерфейсу. Натискання на піктограму відкриє діалогове вікно конфігурації для цього інтерфейсу.

При наведенні курсору на інтерфейс будуть показані всі пов'язані з ним IPv4 і IPv6 адреси і фільтр перехоплення.

Wireshark не обмежується лише мережевими інтерфейсами - на більшості систем ви також можете перехоплювати пакети *USB*, *Bluetooth* та інших типів. Зауважте також, що інтерфейс може бути прихованим.

### Діалогове вікно «Параметри перехоплення»

Коли ви виберете **Capture** → **Options...** (або скористаєтеся відповідним пунктом на головній панелі інструментів), *Wireshark* відкриє діалогове вікно **Параметри захоплення**, як показано на рисунку 5.23. Якщо ви не впевнені, які саме параметри слід вибрати у цьому діалоговому вікні, залиште значення за замовчуванням, і у багатьох випадках програма буде працювати належним чином.

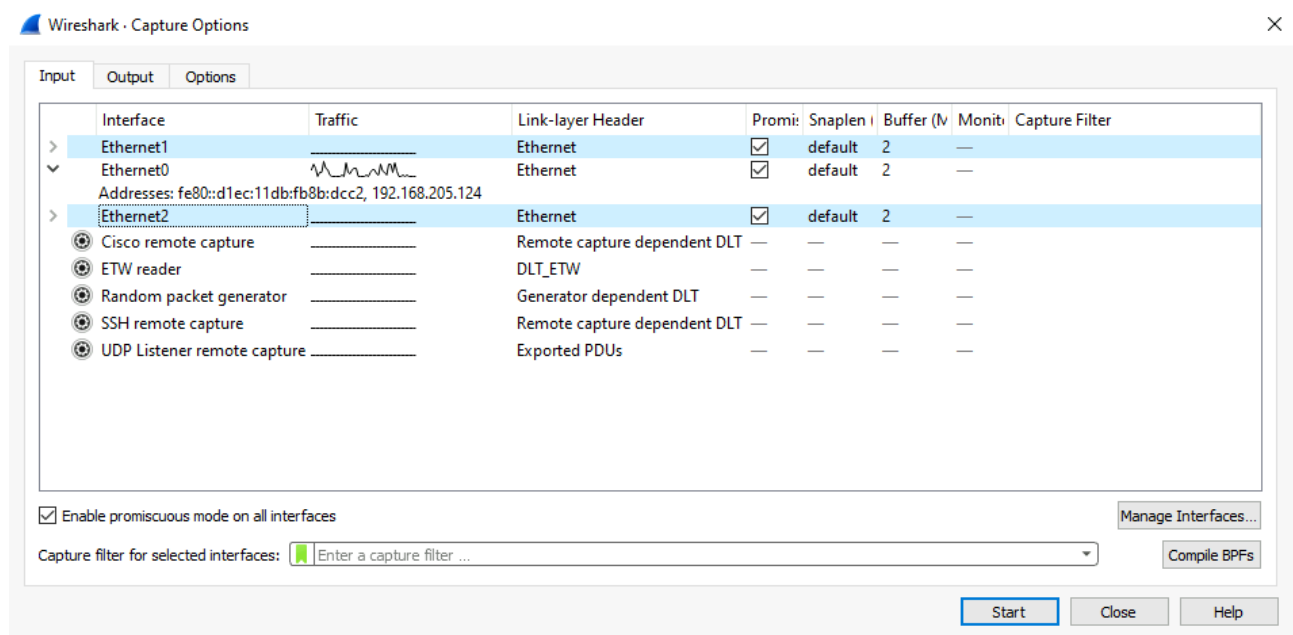


Рисунок 5.23 – Вкладка «Параметри захоплення»

Діалогове вікно **Керування інтерфейсами** спочатку показує вкладку **Локальні інтерфейси**, за допомогою якої ви можете керувати параметрами наведеними на рисунку 5.24.

У *Microsoft Windows* за допомогою вкладки **Віддалені інтерфейси** можна здійснювати перехоплення з інтерфейсу на іншому комп'ютері. На цільовій платформі повинна бути запущена служба *Remote Packet Capture Protocol*, перш ніж *Wireshark* зможе підключитися до неї.

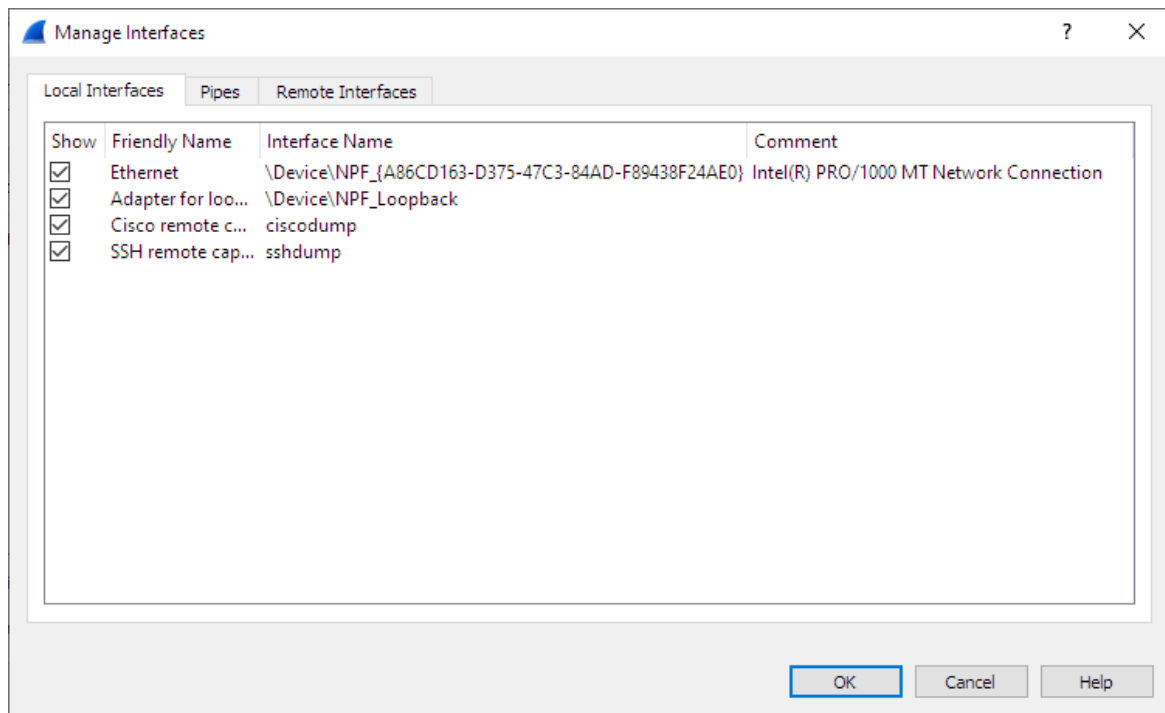


Рисунок 5.24 – Діалогове вікно «Керування інтерфейсами»

### Файли перехоплення та режими файлів

Під час перехоплення базовий механізм перехоплення *libpcap* перехоплює пакети з мережевої карти і зберігає дані пакетів у (відносно) невеликому буфері ядра. Ці дані зчитуються *Wireshark* і зберігаються у файл перехоплення.

За замовчуванням *Wireshark* зберігає пакети у тимчасовий файл. Ви також можете вказати *Wireshark* зберігати дані у певний («постійний») файл і перемикатися на інший файл після закінчення певного часу або перехоплення певної кількості пакетів. Цими параметрами можна керувати на вкладці **Output** діалогового вікна **Параметри перехоплення**.

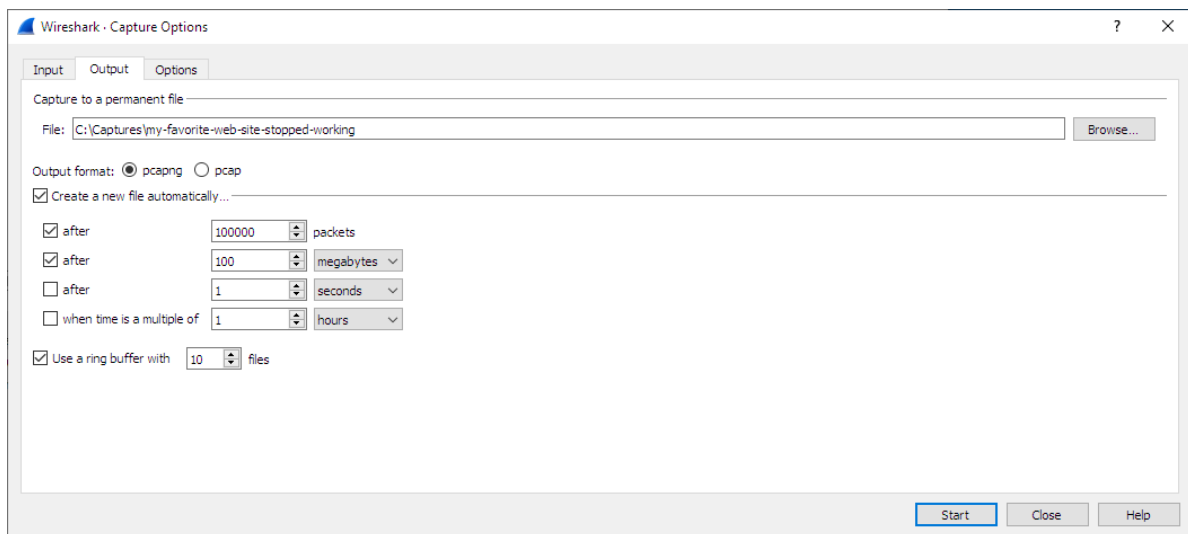


Рисунок 5.25 – Параметри виведення даних захоплення

Під час перехоплення можна побачити діалогове вікно як на рисунку 5.26. Це діалогове вікно показує список протоколів та їхню активність. Його можна увімкнути за допомогою параметра «*capture.show\_info*» у налаштуваннях **Додатково**.

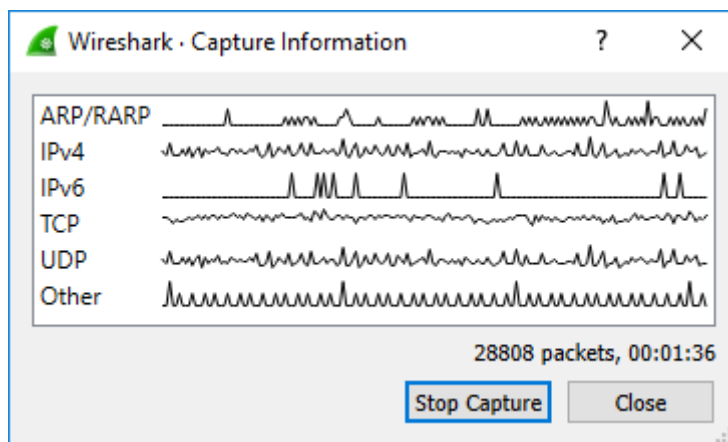


Рисунок 5.26 – Діалогове вікно «Інформація про захоплення»

Після того, як ви припинили перехоплення, ви можете зберегти перехоплений файл для подальшого використання.

Коли ми бачимо список перехоплених кадрів, треба з'ясувати, які з них є вхідними, а які вихідними. Це можна зробити, подивившись на кадр на панелі списку пакетів. Панель списку пакетів показує адреси джерела і призначення кадру (згенеровані і вставлені на мережевому рівні). Якщо адресою джерела є адреса вузла, з яким ви працюєте (показана у вікні перехоплення, коли ви починаєте перехоплення), то кадр є вихідним; якщо адресою призначення є адреса вашого вузла, то кадр є вхідним.

### АНАЛІЗ ТА СТАТИСТИКА

Після того, як ви перехопили деякі пакети або відкрили раніше збережений файл перехоплення, ви можете переглянути пакети, які відображаються на панелі списку пакетів, просто натиснувши на пакет на панелі списку пакетів, що приведе до відображення вибраного пакета у вигляді дерева і байтових панелей перегляду.

Ви можете розгорнути будь-яку частину дерева, щоб переглянути детальну інформацію про кожен протокол у кожному пакеті. Клацання на елементі у дереві виділить відповідні байти у побайтовому поданні. Приклад з вибраним *TCP*-пакетом показано на рисунку 5.27. У вибраному заголовку *TCP*-пакета також є номер підтвердження, який відображається у байтовому поданні як вибрані байти.

Крім того, ви можете переглядати окремі пакети в окремому вікні, як показано на рисунку 5.28. Це можна зробити, двічі клацнувши на елементі у списку пакетів або вибравши пакет, який вас цікавить, на панелі списку пакетів і

вибравши пункт меню **View** → **Show Packet in New Window**. Це дасть вам змогу легко порівняти два або більше пакунків, навіть якщо вони містяться у різних файлах.

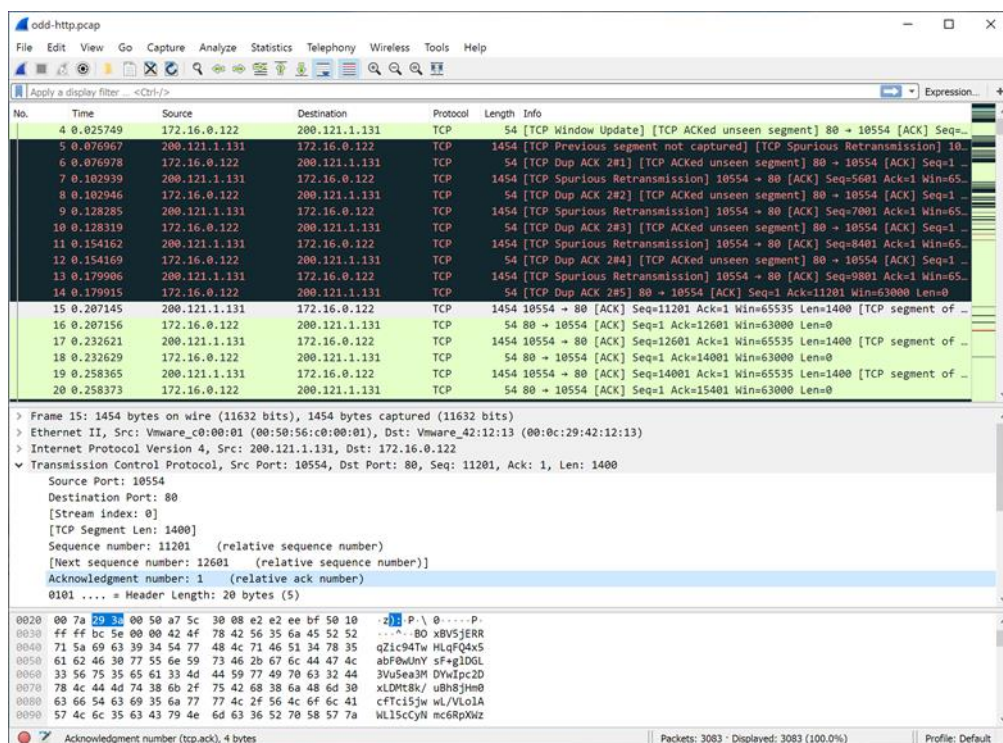


Рисунок 5.27 – Wireshark з вибраним для перегляду TCP-пакетом

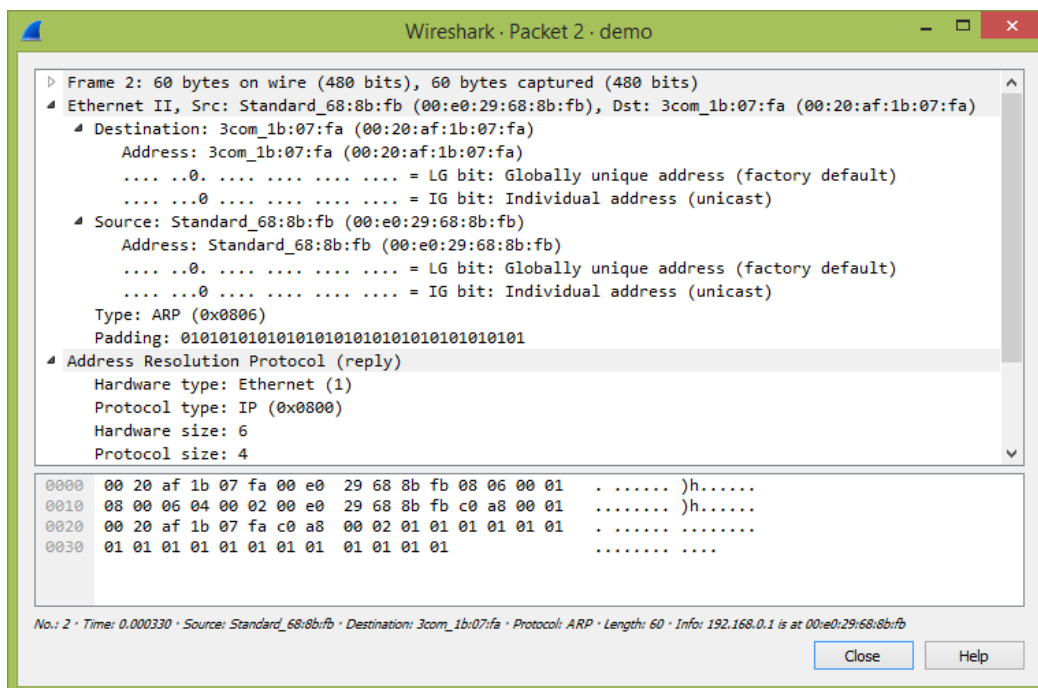


Рисунок 5.28 – Перегляд пакета в окремому вікні

Дуже корисним механізмом, доступним у **Wireshark**, є розфарбування пакетів. Ви можете налаштувати **Wireshark** так, щоб він розфарбовував пакети

відповідно до фільтра відображення. Це дозволить вам підкреслити пакети, які можуть вас зацікавити.

У **Wireshark** є два типи правил розфарбовування: тимчасові правила, які діють лише до завершення роботи програми, і постійні правила, які зберігаються у файлі налаштувань, щоб бути доступними під час наступного запуску **Wireshark**.

Тимчасові правила можна додати, вибравши пакет і натиснувши клавішу **Ctrl** разом з однією з цифрових клавіш. Це призведе до створення правила розфарбовування на основі поточної вибраної бесіди. Програма спробує створити фільтр розмови спочатку на основі **TCP**, потім **UDP**, потім **IP** і, нарешті, **Ethernet**.

Для постійного розфарбовування пакетів виберіть **Colorize with Filter** → **Color X** Wireshark відобразить діалогове вікно **Правила розфарбовування**, як показано на рисунку 5.29.

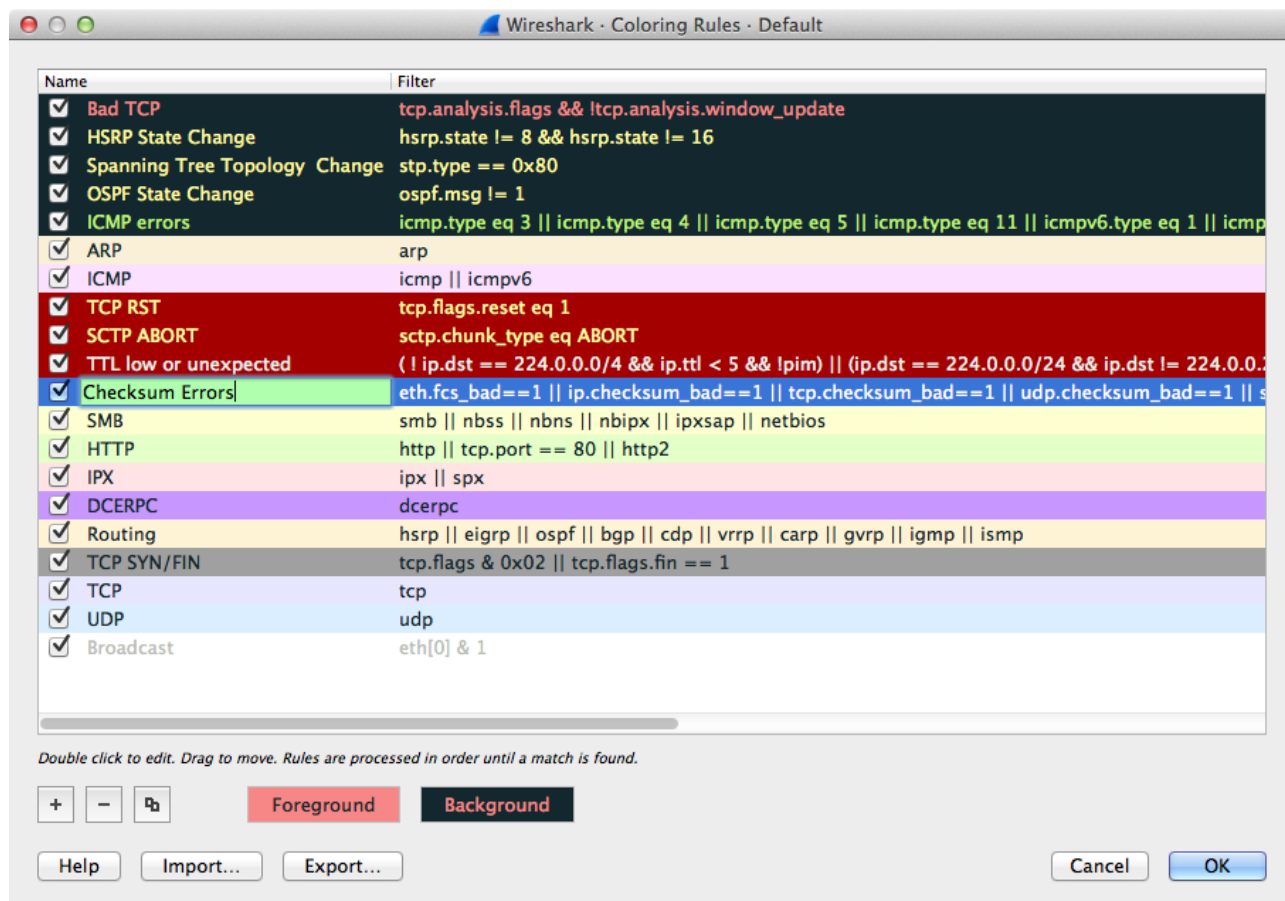


Рисунок 5.29 – Діалогове вікно «Правила розфарбовування»

**Analyze** → **Expert Information** (рисунок 5.30) покаже список основних подій, що відбулися під час перехоплення – відкриття нових сесій, не зовсім правильна поведінка протоколу (повторні запити в **TCP**, повторна передача сегментів тощо).

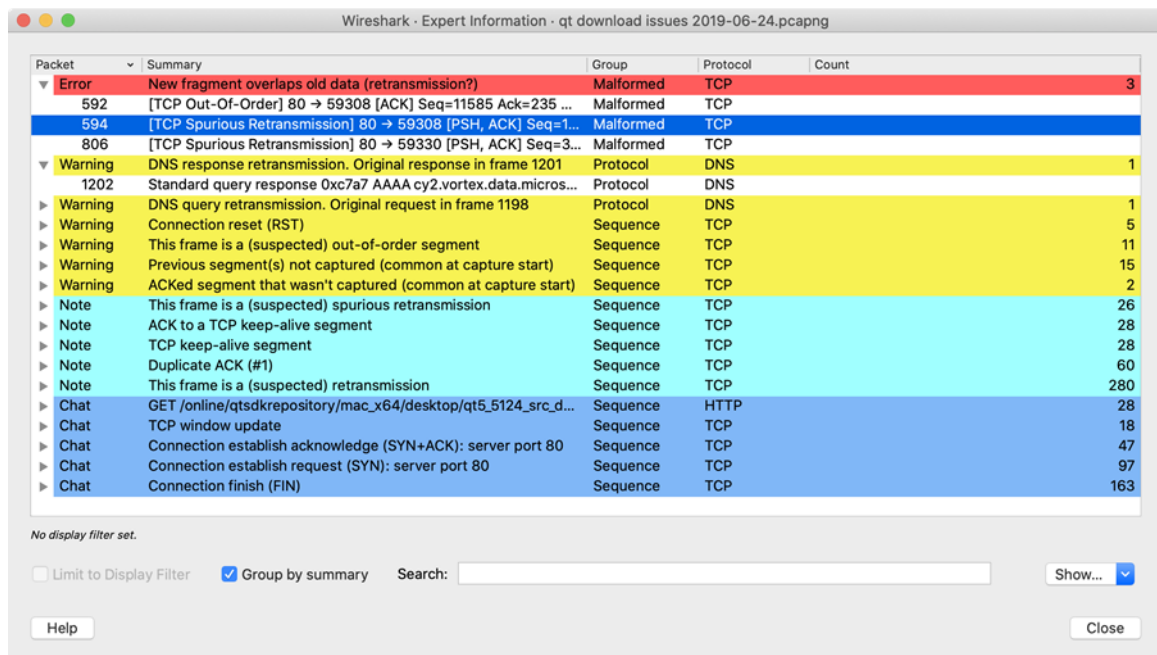


Рисунок 5.30 – Діалогове вікно «Експертна інформація»

**Statistics** → **Capture File Properties** перехоплення дозволяє переглянути деякі статистичні дані для сеансу перехоплення загалом – зокрема, середню кількість пакетів за секунду та обсяг переданих даних (рисунок 5.31).

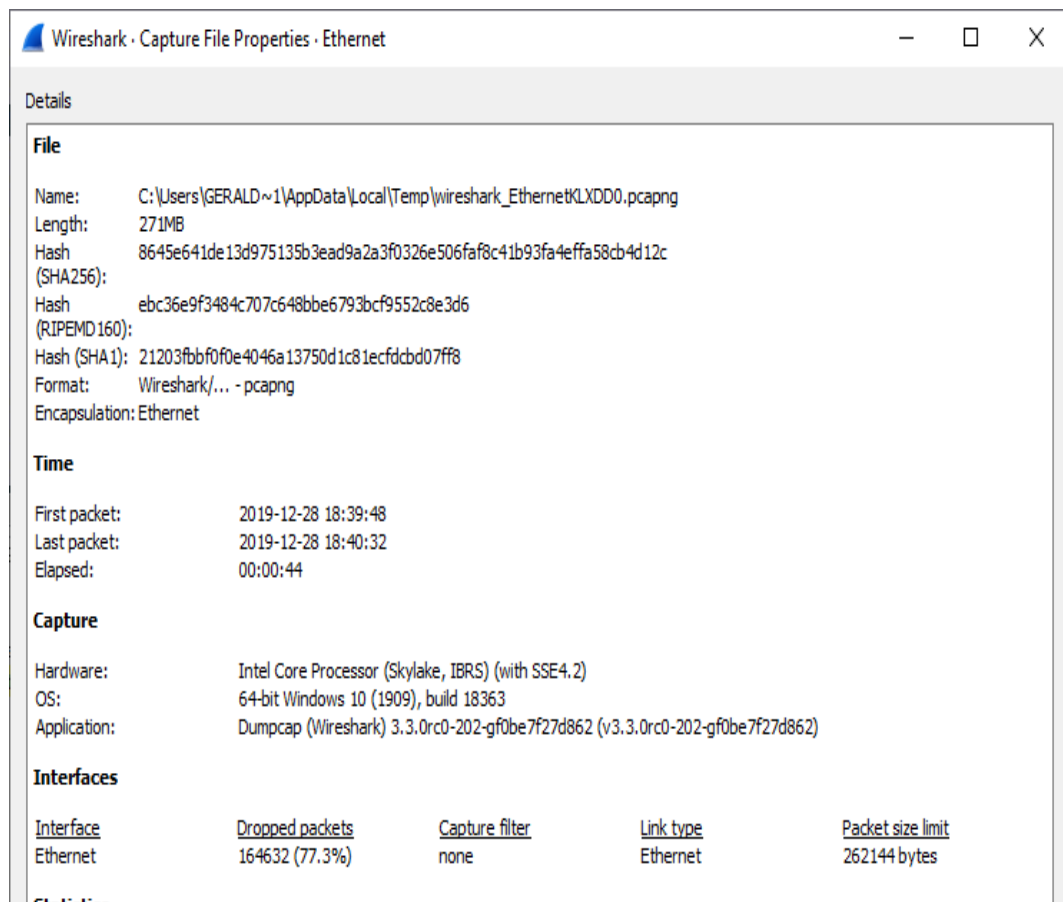


Рисунок 5.31 – Діалогове вікно «Властивості файлу захоплення»

Також є можливість переглянути дерево всіх протоколів у перехопленні. Кожен рядок містить статистичні значення одного протоколу. Два стовпчики (Відсоток пакетів і Відсоток байт) виконують подвійну функцію – слугують гістограмами. Якщо встановлено фільтр відображення, він буде показаний внизу.

| Protocol                           | Percent Packets | Packets | Percent Bytes | Bytes  | Bits/s | End Packets | End Bytes | End Bits/s | PDU's |
|------------------------------------|-----------------|---------|---------------|--------|--------|-------------|-----------|------------|-------|
| Frame                              | 100.0           | 1413    | 100.0         | 717001 | 39 k   | 0           | 0         | 0          | 1413  |
| Linux cooked-mode capture          | 100.0           | 1413    | 3.2           | 22608  | 1,242  | 0           | 0         | 0          | 1413  |
| Internet Protocol Version 4        | 100.0           | 1413    | 3.9           | 28260  | 1,553  | 0           | 0         | 0          | 1413  |
| User Datagram Protocol             | 6.4             | 91      | 0.1           | 728    | 40     | 0           | 0         | 0          | 91    |
| Domain Name System                 | 6.4             | 90      | 0.9           | 6378   | 350    | 90          | 6378      | 350        | 90    |
| Data                               | 0.1             | 1       | 0.0           | 31     | 1      | 1           | 31        | 1          | 1     |
| Transmission Control Protocol      | 93.3            | 1319    | 91.9          | 658589 | 36 k   | 960         | 338701    | 18 k       | 1319  |
| Transport Layer Security           | 9.0             | 127     | 15.4          | 110215 | 6,059  | 127         | 83785     | 4,606      | 134   |
| Hypertext Transfer Protocol        | 5.0             | 70      | 40.9          | 293086 | 16 k   | 39          | 15325     | 842        | 70    |
| Online Certificate Status Protocol | 0.6             | 8       | 1.0           | 7031   | 386    | 8           | 8629      | 474        | 8     |
| Media Type                         | 0.1             | 1       | 0.0           | 282    | 15     | 1           | 282       | 15         | 1     |
| Line-based text data               | 0.8             | 12      | 63.9          | 458331 | 25 k   | 12          | 226139    | 12 k       | 12    |
| JPEG File Interchange Format       | 0.4             | 6       | 9.1           | 65439  | 3,597  | 6           | 67006     | 3,683      | 6     |
| eXtensible Markup Language         | 0.2             | 3       | 49.7          | 356175 | 19 k   | 3           | 33811     | 1,858      | 3     |
| CompuServe GIF                     | 0.1             | 1       | 0.0           | 43     | 2      | 1           | 43        | 2          | 1     |
| Git Smart Protocol                 | 11.5            | 162     | 31.4          | 225057 | 12 k   | 162         | 33299     | 1,830      | 3142  |
| Internet Control Message Protocol  | 0.2             | 3       | 0.1           | 407    | 22     | 0           | 0         | 0          | 3     |
| Domain Name System                 | 0.2             | 3       | 0.0           | 299    | 16     | 3           | 299       | 16         | 3     |

Рисунок 5.32 – Вікно «Ієрархія протоколів»

Мережева розмова – це трафік між двома конкретними кінцевими точками. Наприклад, *IP*-розмова – це весь трафік між двома *IP*-адресами. Опис відомих типів кінцевих точок можна побачити на рисунку 5.33.

Вікно розмов подібне до вікна кінцевих точок. Разом з адресами, лічильниками пакетів і лічильниками байтів у вікні розмови додано чотири стовпчики: час початку розмови («*Rel Start*») або («*Abs Start*»), тривалість розмови у секундах і середня швидкість передачі бітів (не байтів) за секунду у кожному напрямку. Графік часової шкали також побудовано по стовпчиках «*Rel Start*» / «*Abs Start*» і «*Duration*».

| Address A       | Address B       | Packets | Bytes      | Total Packets | Percent Filtered | Packets A → B | Bytes A → B | Packets B → A | Bytes B → A | Rel Start | D   |
|-----------------|-----------------|---------|------------|---------------|------------------|---------------|-------------|---------------|-------------|-----------|-----|
| 0.0.0.0         | 255.255.255.255 | 1       | 383 bytes  | 1             | 100.00%          | 0             | 0 bytes     | 1             | 383 bytes   | 34.656060 | 0   |
| 142.250.186.67  | 192.168.1.85    | 4       | 301 bytes  | 30            | 13.33%           | 2             | 148 bytes   | 2             | 153 bytes   | 31.378530 | 1   |
| 142.250.186.165 | 192.168.1.85    | 28      | 12,467 Kib | 29            | 96.55%           | 16            | 8,460 Kib   | 12            | 4,007 Kib   | 0.000000  | 0   |
| 152.199.19.160  | 192.168.1.85    | 7       | 543 bytes  | 7             | 100.00%          | 4             | 345 bytes   | 3             | 198 bytes   | 7.470308  | 0   |
| 162.159.134.234 | 192.168.1.85    | 43      | 5,452 Kib  | 43            | 100.00%          | 22            | 4,239 Kib   | 21            | 1,213 Kib   | 8.030960  | 87  |
| 172.65.251.78   | 192.168.1.85    | 204     | 23,745 Kib | 204           | 100.00%          | 102           | 15,616 Kib  | 102           | 8,129 Kib   | 1.374770  | 102 |
| 172.104.245.130 | 192.168.1.85    | 66      | 5,221 Kib  | 66            | 100.00%          | 35            | 2,559 Kib   | 31            | 2,662 Kib   | 5.106435  | 90  |
| 172.217.18.106  | 192.168.1.85    | 29      | 3,502 Kib  | 29            | 100.00%          | 16            | 1,716 Kib   | 13            | 1,786 Kib   | 12.638637 | 81  |
| 192.168.1.1     | 192.168.1.255   | 10      | 600 bytes  | 10            | 100.00%          | 0             | 0 bytes     | 0             | 0 bytes     | 6.083682  | 90  |
| 192.168.1.1     | 224.0.0.251     | 329     | 56,480 Kib | 329           | 100.00%          | 329           | 56,480 Kib  | 0             | 0 bytes     | 0.555528  | 103 |
| 192.168.1.1     | 255.255.255.255 | 12      | 720 bytes  | 12            | 100.00%          | 12            | 720 bytes   | 0             | 0 bytes     | 0.862892  | 102 |
| 192.168.1.21    | 192.168.1.255   | 10      | 1,295 Kib  | 10            | 100.00%          | 10            | 1,295 Kib   | 0             | 0 bytes     | 1.476232  | 97  |
| 192.168.1.21    | 255.255.255.255 | 4       | 792 bytes  | 4             | 100.00%          | 4             | 792 bytes   | 0             | 0 bytes     | 8.048800  | 90  |
| 192.168.1.35    | 224.0.0.251     | 1       | 279 bytes  | 1             | 100.00%          | 1             | 279 bytes   | 0             | 0 bytes     | 43.258929 | 0   |
| 192.168.1.35    | 239.255.255.250 | 93      | 50,372 Kib | 93            | 100.00%          | 93            | 50,372 Kib  | 0             | 0 bytes     | 19.908648 | 82  |
| 192.168.1.36    | 224.0.0.251     | 2       | 1,633 Kib  | 2             | 100.00%          | 2             | 1,633 Kib   | 0             | 0 bytes     | 39.577346 | 25  |
| 192.168.1.54    | 192.168.1.85    | 2       | 120 bytes  | 2             | 100.00%          | 1             | 54 bytes    | 1             | 66 bytes    | 33.074264 | 0   |
| 192.168.1.54    | 224.0.0.251     | 1       | 326 bytes  | 1             | 100.00%          | 1             | 326 bytes   | 0             | 0 bytes     | 29.124620 | 0   |
| 192.168.1.85    | 172.248.145.106 | 55      | 28,001 Kib | 55            | 100.00%          | 26            | 11,065 Kib  | 29            | 16,936 Kib  | 53.787326 | 19  |
| 192.168.1.85    | 311.84.51       | 17      | 1,813 Kib  | 17            | 100.00%          | 10            | 1,011 Kib   | 7             | 822 bytes   | 3.847714  | 86  |
| 192.168.1.85    | 311.84.52       | 21      | 1,750 Kib  | 21            | 100.00%          | 14            | 1,107 Kib   | 7             | 658 bytes   | 2.200680  | 60  |
| 192.168.1.85    | 35.186.224.25   | 19      | 2,550 Kib  | 43            | 44.19%           | 8             | 782 bytes   | 11            | 1,766 Kib   | 13.462394 | 90  |
| 192.168.1.85    | 35.186.224.40   | 16      | 1,355 Kib  | 16            | 100.00%          | 8             | 700 bytes   | 8             | 688 bytes   | 10.725210 | 90  |
| 192.168.1.85    | 35.186.224.47   | 29      | 5,505 Kib  | 29            | 100.00%          | 14            | 1,143 Kib   | 15            | 4,362 Kib   | 17.476152 | 79  |
| 192.168.1.85    | 54.216.252.255  | 15      | 1,230 Kib  | 15            | 100.00%          | 10            | 805 bytes   | 5             | 455 bytes   | 2.423718  | 100 |
| 192.168.1.85    | 104.199.65.124  | 5       | 363 bytes  | 5             | 100.00%          | 3             | 209 bytes   | 2             | 154 bytes   | 40.114078 | 59  |
| 192.168.1.85    | 136.143.190.75  | 40      | 9,621 Kib  | 40            | 100.00%          | 23            | 2,975 Kib   | 17            | 6,646 Kib   | 0.855213  | 100 |
| 192.168.1.85    | 172.217.18.1    | 83      | 39,547 Kib | 83            | 100.00%          | 41            | 10,481 Kib  | 42            | 29,065 Kib  | 53.775691 | 1   |
| 192.168.1.85    | 172.217.18.1    | 86      | 49,088 Kib | 86            | 100.00%          | 42            | 10,907 Kib  | 44            | 38,181 Kib  | 54.327884 | 30  |
| 192.168.1.85    | 192.168.1.21    | 9       | 670 bytes  | 9             | 100.00%          | 5             | 390 bytes   | 4             | 280 bytes   | 1.477461  | 97  |
| 192.168.1.85    | 192.168.1.32    | 1       | 46 bytes   | 1             | 100.00%          | 1             | 46 bytes    | 0             | 0 bytes     | 20.537148 | 0   |
| 192.168.1.85    | 192.168.1.250   | 36      | 3,768 Kib  | 36            | 100.00%          | 18            | 1,433 Kib   | 18            | 2,335 Kib   | 14.918659 | 39  |

Рисунок 5.34 – Вікно «Мережеві розмови»

Також є можливість будувати графіки даних пакетів і протоколів різними способами. Як показано на рисунку 5.35, це вікно містить область для малювання графіків, а також список графіків, які можна налаштувати. Графіки зберігаються у вашому поточному профілі. Вони розділені на часові інтервали, які можна встановити. Наведення на графік показує останній пакет в кожному інтервалі, за винятком випадків, зазначених нижче. Натиснувши на графік, ви перейдете до відповідного пакета у списку пакетів.

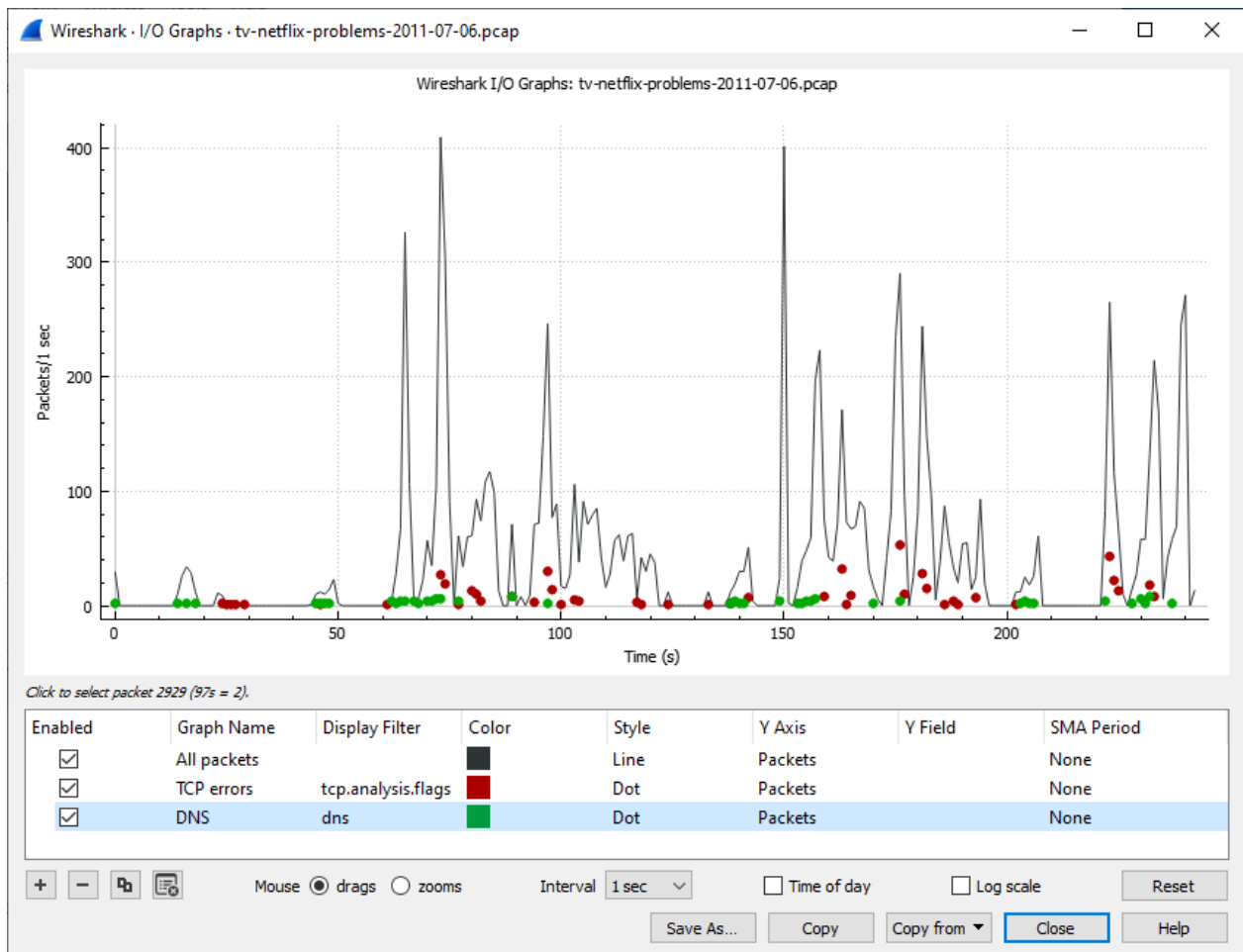


Рисунок 5.35 – Вікно «Графіки введення/виведення»

Вікно **Flow Graph** показує з'єднання між вузлами. Для кожного перехопленого з'єднання відображається час передачі пакетів, напрямок, порти і коментарі. Ви можете відфільтрувати всі з'єднання за *ICMP*-потокми, *ICMPv6*-потокми, *UIM*-потокми і *TCP*-потокми. Вікно *Flow Graph* використовується для відображення декількох різних тем. На основі цього воно пропонує різні елементи керування (рисунок 5.36). Кожна вертикальна лінія представляє певний вузол, який можна побачити у верхній частині вікна.

**Довжина пакетів** показує розподіл довжин пакетів та відповідну інформацію. Інформація розбита за діапазонами довжин пакетів, як показано на рисунку 5.37.

Буває дуже корисно бачити протокол так, як його бачить прикладний рівень. Можливо, ви шукаєте паролі в потоці Telnet або намагаєтеся розібратися в

потоці даних. Можливо, вам просто потрібен фільтр відображення, щоб показати тільки пакети в потоці *TLS* або *SSL*. У цьому випадку вам стане в нагоді можливість Wireshark відстежувати потоки протоколів.

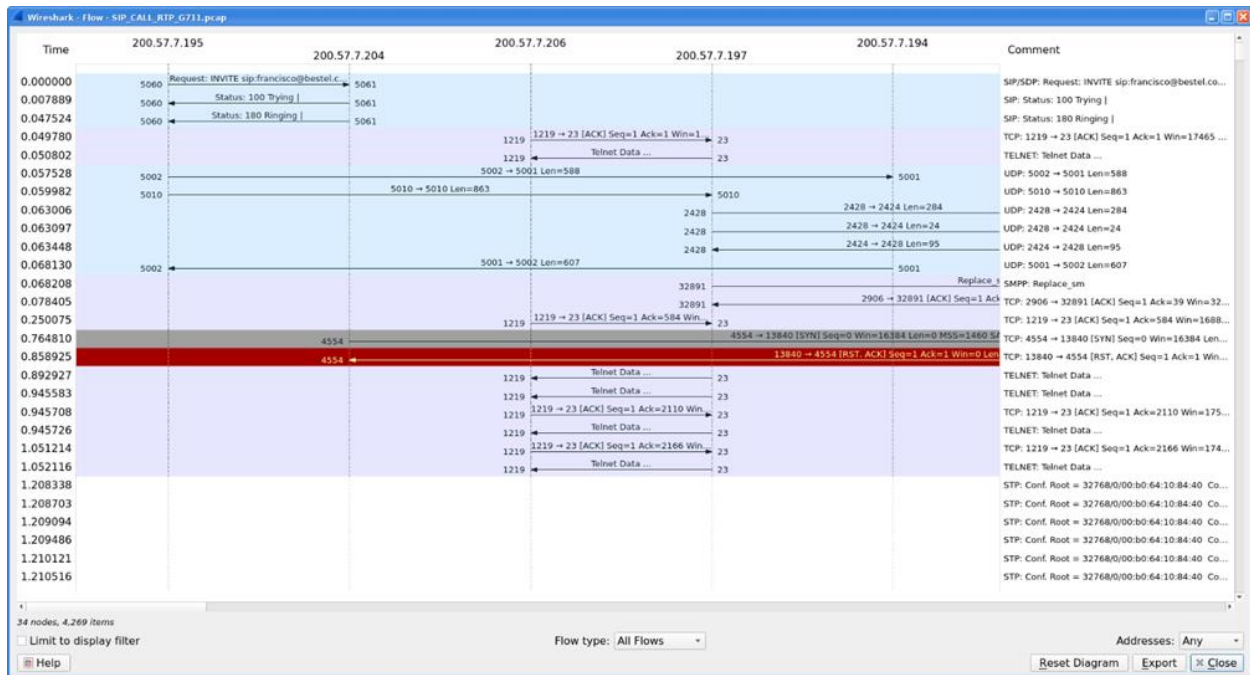


Рисунок 5.36 – Вікно *Flow Graph*

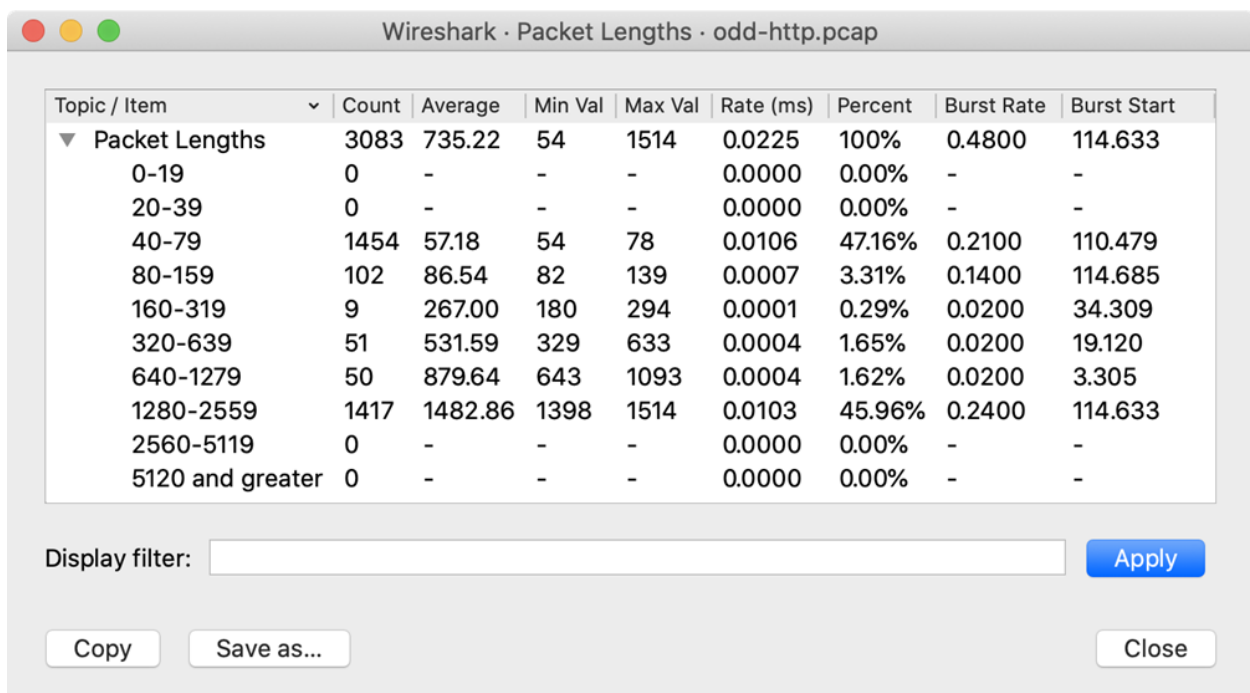


Рисунок 5.37 – Вікно «Довжина пакетів»

Щоб відфільтрувати певний потік, виберіть пакет *TCP*, *UDP*, *DHCP*, *TLS*, *HTTP*, *HTTP/2*, *QUIC* або *SIP* у списку пакетів потоку/з'єднання, який вас цікавить, а потім виберіть пункт меню **Analyze → Follow → TCP Stream** (або скористайтеся контекстним меню у списку пакетів). Wireshark встановить відповідний фільтр відображення і відобразить діалогове вікно з даними з потоку, як показано на рисунку 5.38.

Вміст потоку відображається у тій самій послідовності, у якій він з'явився у мережі. Недруковані символи замінюються крапками. Трафік від клієнта до сервера позначено червоним кольором, а трафік від сервера до клієнта – синім. Вміст потоку не оновлюватиметься під час захоплення в реальному часі. Щоб отримати найсвіжіший вміст, вам доведеться знову відкрити діалогове вікно.

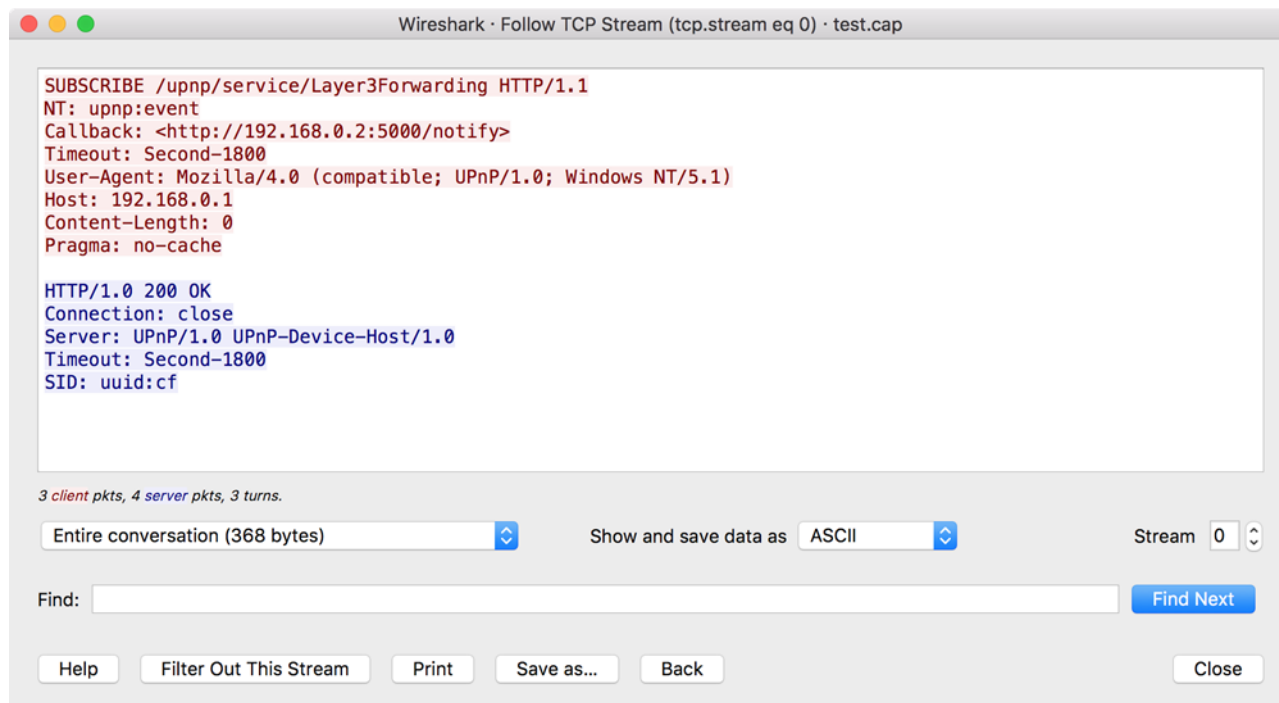


Рисунок 5.38 – Діалогове вікно «Відстежити TCP-потік»

## ПРАКТИЧНЕ ЗАВДАННЯ

**Мета роботи:** отримати поглиблене розуміння мережевих концепцій, перехоплюючи та аналізуючи пакети, що надсилаються та отримуються з вузла.

### Порядок виконання роботи

1. За потреби, завантажити та встановити *Wireshark*.
2. Запустити *Wireshark* в режимі перехоплення трафіку, що проходить через інтерфейс, підключений до локальної мережі (зазвичай `eth0`).
4. Зімітуйте мережеву активність протягом 10-15 хвилин з різних вузлів. Для цього треба виконати наступні дії:
  - Відкрити вебсайт `http:// ...` (таблиця 5.28).
  - Надіслати **ping** на віддалений вузол (таблиця 5.28).
  - Підключитися до одного з доступних мережевих дисків Windows (якщо такі ресурси доступні в мережі).
  - Виконати інші дії, що вимагають підключення до мережі.
5. Зупинити перехоплення, зберегти файл *pcap* та прикріпити його до звіту (якщо файл більший за 10 Мб, то мати його на флешці під час захисту роботи).
6. Відкрити фрейм *Ethernet*, що відправляється на шлюз, та проаналізувати його *PDU*.

7. Відкрити кадр *TCP* обміну з віддаленим вебсайтом та проаналізувати його *PDU*.

8. Провести аналіз мережевого трафіка:

- за інтенсивністю між вузлами;
- за довжинами пакетів;
- за активністю мережі.

Таблиця 5.28 – Дані для визначення параметрів адресації мережі

| № варіанта | http:// ...       | ping              |
|------------|-------------------|-------------------|
| 1          | www.google.com    | www.ebay.com      |
| 2          | www.youtube.com   | www.pinterest.com |
| 3          | www.facebook.com  | www.tumblr.com    |
| 4          | www.baidu.com     | www.snapchat.com  |
| 5          | www.wikipedia.org | www.airbnb.com    |
| 6          | www.twitter.com   | www.dropbox.com   |
| 7          | www.instagram.com | www.spotify.com   |
| 8          | www.linkedin.com  | www.alibaba.com   |
| 9          | www.reddit.com    | www.quora.com     |
| 10         | www.amazon.com    | www.github.com    |
| 11         | www.netflix.com   | www.cnn.com       |
| 12         | www.yahoo.com     | www.bbc.com       |
| 13         | www.microsoft.com | www.espn.com      |
| 14         | www.whatsapp.com  | www.adobe.com     |
| 15         | www.tiktok.com    | www.wordpress.org |

### КОНТРОЛЬНІ ПИТАННЯ

1. Як відфільтрувати пакети, які мають більшу певного значення мережеву відстань?
2. Як відфільтрувати пакети певного відправника?
3. Як експортувати захоплені пакети у файл заданого типу?
4. *Wireshark* не відображає поле преамбули заголовку кадру. Що містить преамбула?
5. Що позначається терміном *sniffer*?
6. До якого типу програмних засобів відноситься *Wireshark*?
7. Які функції виконує аналізатор *Wireshark*?
8. Яка довжина та структура *MAC*-адреси?
9. Яку максимальну і яку мінімальну довжину можуть мати *Ethernet* кадри?
10. Що таке преамбула та який вона має вміст?
11. Яке значення першого байту *IP* пакету є найбільш типовим? Дайте пояснення.
12. Як у програмі *Wireshark* задати захоплення тільки пакетів *ICMP*?
13. Які мінімальне та максимальне значення має поле *TTL* у заголовку *IP*-пакета?
14. Які значення буде мати поле «Протокол», якщо *IP*-пакет використовується для пересилання *ICMP* повідомлень, даних протоколів *TCP* та *UDP*?

## 5.1.7 ЗАСОБИ ВІДДАЛЕНОГО ДОСТУПУ ТА АДМІНІСТРУВАННЯ

### ВСТАНОВЛЕННЯ ТА ВИКОРИСТАННЯ СЕРВЕРНИХ ТА КЛІЄНТСЬКИХ ДОДАТКІВ ПРОТОКОЛІВ *TELNET* ТА *SSH* В ОС *WINDOWS*

В ОС *Windows XP* сервер та клієнт протоколу *Telnet* встановлюються автоматично під час початкового встановлення системи. В ОС *Windows 7/8/10* необхідно виконати додаткові дії щодо їх встановлення. У *Windows 7* інсталяція *Telnet*-сервера та *Telnet*-клієнта проводиться через "Панель керування" → "Програми та засоби" → "Увімкнення або вимкнення засобів *Windows*" → "*Telnet*-сервер" ("*Telnet*-клієнт"). В ОС *Windows* запуск (або зупинка) *Telnet*-сервера здійснюється через , , "Панель керування" → , , "Адміністрування" → , , "Служби" (запускається додаток **tlntsvr**). Альтернативним способом запуску/зупинки *Telnet*-сервера є використання мережних команд **net start tlntsvr** та **net stop tlntsvr** відповідно. Для адміністрування *Telnet*-сервера в ОС *Windows* передбачено використання специфічного текстового додатка **tlntadmn.exe**.

*Telnet*-клієнти ОС *Windows* реалізуються у вигляді текстових та графічних утиліт. Серед вбудованих в ОС графічних *Telnet*-клієнтів можна відмітити *Hyper Terminal*. У багатьох випадках адміністратори використовують багатофункціональні термінальні клієнти сторонніх виробників, які дають змогу здійснювати віддалені мережні підключення не лише за протоколом *Telnet*, а й за іншими протоколами (зокрема, *SSH* та *rlogin*). Часто такі клієнти забезпечують роботу і консольних підключень. Найбільш відомими термінальними клієнтами для ОС *Windows* є *Putty* ([www.putty.org](http://www.putty.org)) та *SecureCRT* ([www.vandyke.com](http://www.vandyke.com)).

Запуск вбудованого в ОС *Windows* *Telnet*-клієнта проводиться за допомогою командного рядка. Для безпосереднього зазначення параметрів підключення можуть використовуватися відповідні ключі:

```
C:\>telnet /?
telnet [-a][-e escape char][-f log file][-l user][-t term][host [port]]
-a Attempt automatic logon. Same as -l option except uses
 the currently logged on user's name.
-e Escape character to enter telnet client prompt.
-f File name for client side logging
-l Specifies the user name to log in with on the remote system.
 Requires that the remote system support the TELNET ENVIRON option.
-t Specifies terminal type.
 Supported term types are vt100, vt52, ansi and vtnt only.
host Specifies the hostname or IP address of the remote computer
 to connect to.
port Specifies a port number or service name.
```

Слід зазначити, що для різних версій ОС *Windows* можуть бути наявні відмінності у переліку та використанні ключів.

Якщо додаток запускається без ключів, то адміністратор потрапляє у командний рядок *Telnet*-клієнта. У цьому разі параметри мережного підключення налагоджуються за допомогою відповідних команд:

```
Welcome to Microsoft Telnet Client

Escape Character is ']'

Microsoft Telnet> ?

Commands may be abbreviated. Supported commands are:

c - close close current connection
d - display display operating parameters
o - open hostname [port] connect to hostname (default port 23).
q - quit exit telnet
set - set set options (type 'set ?' for a list)
sen - send send strings to server
st - status print status information
u - unset unset options (type 'unset ?' for a list)
?/h - help print help information
```

Слід звернути уваги на особливості налагодження параметрів *Telnet*-з'єднання за допомогою команди **set**:

```
Microsoft Telnet> set ?
bsasdel Backspace will be sent as delete
crlf New line mode - Causes return key to send CR & LF
delasbs Delete will be sent as backspace
escape x x is an escape charater to enter telnet client prompt
localecho Turn on localecho.
logfile x x is current client log file
logging Turn on logging
mode x x is console or stream
ntlm Turn on NTLM authentication.
term x x is ansi, vt100, vt52, or vtnt
```

## ВСТАНОВЛЕННЯ ТА ВИКОРИСТАННЯ СЕРВЕРНИХ ТА КЛІЄНТСЬКИХ ДОДАТКІВ ПРОТОКОЛІВ ВІДДАЛЕНОГО ДОСТУПУ *TELNET* ТА *SSH* В ОС *LINUX/UNIX*

Більшість *Linux/Unix*-подібних ОС у своєму складі мають засоби забезпечення роботи протоколу віддаленого доступу *Telnet*. У деяких з них *Telnet*-сервер та *Telnet*-клієнт встановлюються і активуються автоматично при початковому встановленні системи. Через проблеми безпеки протоколу у більшості сучасних

ОС не виконується встановлення та активація роботи *Telnet*-сервера, а виконується лише встановлення *Telnet*-клієнта.

Для запуску *Telnet*-сервера використовується системна служба (демон) **telnetd**. Запуск може здійснюватися як в автоматичному, так і у ручному режимі. Додаток *Telnet*-клієнта встановлюється автоматично у більшості ОС *Linux/Unix*. Його запуск проводиться за допомогою командного рядка. За допомогою відповідних ключів існує можливість налагодити відповідні параметри підключення. Перелік ключів для запуску *Telnet*-клієнта ОС *Linux Microcore* наведено далі:

```
tc@box:~$ telnet
BusyBox v1.19.0 (2011-08-14 21:05:38 UTC) multi-call binary.
Usage: telnet [-a] [-l USER] HOST [PORT]
Connect to telnet server
 -a Automatic login with $USER variable
 -l USER Automatic login as USER
```

Як безпечна альтернатива протоколу *Telnet* в *Linux/Unix*-системах широко використовується протокол віддаленого доступу *SSH*. На ринку програмних додатків існує досить багато відкритих розробок протоколу *SSH*. Найбільш поширеною і такою, що динамічно розвивається, є реалізація відома як *OpenBSD Secure Shell* або *OpenSSH* ([www.openssh.org](http://www.openssh.org)). У більшості сучасних *Linux/Unix*-подібних ОС *SSH*-сервер та *SSH*-клієнт встановлюються та активуються автоматично під час початкового встановлення системи. У багатьох ОС користувач має можливість керувати процесом встановлення *SSH*-сервера та *SSH*-клієнта у діалоговому режимі.

Сценарій налагодження та активації *OpenSSH* сервера в ОС *Linux Microcore* наведений нижче:

```
root@box:~#ls /mnt/hda1/tce/optional/openssh*
/mnt/sda1/tce/optional/openssh.tcz
/mnt/sda1/tce/optional/openssh.tcz.dep
/mnt/sda1/tce/optional/openssh.tcz.md5.txt
root@box:~#mv /usr/local/etc/ssh/sshd_config.example /usr/local/etc/ssh/sshd_config
root@box:~# /usr/local/etc/init.d/openssh start
root@box:~#
```

Для автоматичної активації *OpenSSH*-сервера при запуску ОС *Linux Microcore* необхідно відредагувати та зберегти відповідні конфігураційні файли системи. Внесення змін у зазначені файли виконується або за допомогою системної команди **echo**, або за допомогою текстового редактора. Оскільки *Linux Microcore* є специфічно побудованою системою, то для збереження змін конфігурації слід скористатися спеціальною командою **filetool.sh -b**.

Сценарій виконання дій щодо внесення змін у конфігураційні файли *OpenSSH*-сервера в ОС *Linux Microcore* наведено нижче:

```

root@box:~# echo ,,openssh.tcz,, >> /mnt/hdal/tce/onboot.lst
root@box:~# echo ,,/usr/local/etc/init.d/openssh start,, >> /opt/bootlocal.sh
root@box:~# echo ,,/usr/local/etc/ssh,, >> /opt/.filetool.lst
root@box:~# /usr/bin/filetool.sh -b

```

Перелік ключів для запуску SSH-клієнта ОС Linux Microcore наведено далі:

```

root@box:~# ssh
usage: ssh [-1246AaCfGkMNNqsTtVvXxYy] [-b bind_address] [-c cipher_spec]
 [-D [bind_address:]port] [-e escape_char] [-F configfile]
 [-I pkcs11] [-i identify_file]
 [-L [bind_address:]port:host:hostport]
 [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
 [-R [bind_address:]port:host:hostport] [-S ctl_path]
 [-W host:port] [-w local_tun[:remote_tun]
 [user@]hostname [command]

```

### ПОРЯДОК НАЛАГОДЖЕННЯ СЕРВЕРА ТА КЛІЄНТА ПРОТОКОЛУ *TELNET* НА ОБЛАДНАННІ *CISCO*

Налагодження функціонування *Telnet*-сервера на пристроях *Cisco* для забезпечення організації віддаленого доступу може здійснюватися з використанням трьох підходів:

- безпарольний вхід;
- вхід із використанням паролів на мережні підключення (та командні режими);
- вхід із використанням механізму користувачів.

Для безпарольного входу порядок виконання етапів налагодження є таким:

1. Обрати мережне (мережні) підключення для подальшої активації віддаленого доступу за протоколом *Telnet* (обов'язково).
2. Відключити використання аутентифікації для входу в систему для обраного мережного підключення/обраних мережних підключень (обов'язково).
3. Активувати можливість *Telnet*-підключення для відповідного мережного підключення/відповідних мережних підключень (обов'язково).
4. Налогодити додаткові параметри (тайм-аути, системні повідомлення тощо) для відповідного мережного підключення/відповідних мережних підключень (необов'язково).

Для входу з використанням паролів на мережні підключення порядок виконання етапів налагодження є таким:

1. Обрати мережне (мережні) підключення для подальшої активації віддаленого доступу за протоколом *Telnet* (обов'язково).
2. Створити пароль входу для відповідного мережного підключення/відповідних мережних підключень та паролі на командні режими (обов'язково).

3. Активувати використання парольної аутентифікації для відповідного мережного підключення/відповідних мережних підключень (обов'язково).
4. Активувати можливість *Telnet*-підключення для відповідного мережного підключення/відповідних мережних підключень (обов'язково).
5. Налаштувати додаткові параметри (тайм-аути, системні повідомлення тощо) для відповідного мережного підключення/відповідних мережних підключень (необов'язково).

Для входу з використанням механізму користувачів порядок виконання етапів налагодження є таким:

1. Створити локального користувача із зазначенням відповідного рівня привілеїв та пароля (обов'язково).
2. Обрати мережне (мережні) підключення для подальшої активації віддаленого доступу за протоколом *Telnet* (обов'язково).
3. Активувати використання парольної аутентифікації з використанням локальної бази користувачів для відповідного мережного підключення/відповідних мережних підключень (обов'язково).
4. Активувати можливість *Telnet*-підключення для відповідного мережного підключення/відповідних мережних підключень (обов'язково).
5. Налаштувати додаткові параметри (тайм-аути, системні повідомлення тощо) для відповідного мережного підключення/відповідних мережних підключень (необов'язково).

Для організації звичайного підключення для *Telnet*-клієнта не потрібно проводити налагодження параметрів. За потреби організації специфічного складного підключення існує можливість налагодження певних специфічних параметрів, наприклад, інтерфейсу виходу підключення на маршрутизаторі.

За звичайного використання для *Telnet*-клієнта немає необхідності виконувати налагодження параметрів підключення. За потреби можливе використання великої кількості специфічних параметрів підключення (наприклад, тип терміналу, перевірка достовірності, інтерфейс виходу для маршрутизатора). Перелік параметрів можна визначити з довідки системи. Налаштування параметрів здійснюється безпосередньо у командному рядку під час організації сеансу. Слід зазначити, що за допомогою *Telnet*-клієнта можна підключатися не лише до *Telnet*-сервера, а й до серверів та складових інших мережних протоколів стеку *TCP/IP* (зокрема, поштових протоколів *SMTP*, *POP3*, протоколу маршрутизації *BGP* тощо).

## ПОРЯДОК НАЛАГОДЖЕННЯ СЕРВЕРА ТА КЛІЄНТА ПРОТОКОЛУ *SSH* НА ОБЛАДНАННІ *CISCO*

Налагодження функціонування *SSH*-сервера на пристроях *Cisco* для забезпечення віддаленого доступу може здійснюватися з використанням двох підходів:

- з використанням імені пристрою та імені домену;
- з використанням ключових пар *RSA* (без використання імені пристрою та імені домену).

Слід зазначити, що одним із обов'язкових попередніх етапів налагодження *SSH*-сервера є створення локального користувача з зазначенням відповідного рівня привілеїв та пароля.

Для підходу з використанням імені пристрою та імені домену порядок виконання етапів налагодження є таким:

1. Виконати іменування пристрою (обов'язково).
2. Виконати іменування домену (обов'язково).
3. Згенерувати *SSH*-ключ (ключову пару *RSA*), який буде використовуватися у процесі роботи (обов'язково).
4. Налогодити додаткові параметри *SSH*-сервера: версію протоколу, час тайм-ауту, кількість спроб аутентифікації тощо. (необов'язково).
5. Обрати мережне (мережні) підключення для подальшої активації віддаленого доступу за протоколом *SSH* (обов'язково).
6. Активувати використання локальної бази даних користувачів для обраного мережного підключення/обраних мережних підключень (обов'язково).
7. Активувати можливість *SSH*-підключення для відповідного мережного підключення / відповідних мережних підключень (обов'язково).
8. Налогодити додаткові параметри (тайм-аути, системні повідомлення тощо) для відповідного мережного підключення/відповідних мережних підключень (необов'язково).

Для підходу з використанням ключових пар *RSA* (без використання імені пристрою та імені домену) порядок виконання етапів налагодження є таким:

1. Створити ключову пару *RSA*, яка буде використовуватися у процесі роботи (обов'язково).
2. Згенерувати ключову пару *RSA* із зазначенням довжини ключа (обов'язково).
3. Налогодити додаткові параметри *SSH*-сервера: версію протоколу, час тайм-ауту, кількість спроб аутентифікації та ін. (необов'язково).
4. Обрати мережне (мережні) підключення для подальшої активації віддаленого доступу за протоколом *SSH* (обов'язково).

5. Активувати використання локальної бази даних користувачів для обраного мережного підключення/обраних мережних підключень (обов'язково).
6. Активувати можливість *SSH*-підключення для відповідного мережного підключення/відповідних мережних підключень (обов'язково).
7. Налаштувати додаткові параметри (тайм-аут, системні повідомлення тощо) для відповідного мережного підключення/відповідних мережних підключень (необов'язково).

За звичайного використання для *SSH*-клієнта не потрібно виконувати налагодження параметрів підключення. Специфічні параметри підключення встановлюються за рахунок використання ключів у командному рядку клієнта.

### ЗАГАЛЬНІ КОМАНДИ НАЛАГОДЖЕННЯ ФУНКЦІОНУВАННЯ ПРОТОКОЛІВ ВІДДАЛЕНОГО ДОСТУПУ НА ПРИСТРОЯХ *CISCO*

Для налагодження функціонування протоколів віддаленого доступу (зокрема, *Telnet* та *SSH*) на пристроях *Cisco* використовуються як деякі загальні для всіх протоколів команди, так і характерні лише для певного протоколу команди. До загальних команд належать такі команди: **password**, **username**, **login**, **transport**, **rotary**, **autocommand**, **login**, **security authentication** та похідні від них команди.

Команди **login**, **password**, **username** призначені для налагодження параметрів аутентифікації для певного мережного підключення, команди групи **transport** призначені для дозволу/заборони віддалених підключень до/з пристрою з використанням різних мережних протоколів. Команда **rotary** відповідає за налагодження нестандартних портів для підключень. Команда **autocommand** дає можливість налагодити виконання певної команди режиму користувача після підключення. Для управління сеансами мережних протоколів можуть використовуватися як певні комбінації клавіш (для призупинення сесії **Ctrl+Shift+6, x**), так і певні команди (повернення до сеансу – команда **resume**, завершення сеансу – команда **disconnect**).

Важливими командами, що дають змогу здійснювати контроль та журналювання процесу віддалених підключень є команди, похідні від команд **login** та **security: login block-for**, **login delay**, **login on-failure log**, **login on-success log**, **login quiet-mode** та **security authentication failure rate** відповідно. Команда **login block-for** застосовується для блокування можливості підключення до пристрою на певний період часу, якщо перевищено кількість спроб підключення на вставлений інтервал часу. Команда **login delay** зазначає затримку між спробами підключення. Команди **login on-failure log** та **login on-success log** застосовуються для активації журналювання подій при підключенні та

аутентифікації користувача у системі. Вказані команди активують журналювання подій про неуспішні та успішні спроби відповідно. Команда **login quiet-mode access-class** застосовується для активації списків доступу для віддалених підключень. Команда **security authentication failure** встановлює кількість дозволених невдалих спроб входу в систему (за хвилину), перевищення якої викличе генерацію повідомлення для журналювання подій. Призначення та синтаксис усіх вищезгаданих команд наведено нижче.

Синтаксис команди **transport input** (режим конфігурування лінії):

```
transport input {value | values},
```

де **value** – параметр, який може набувати значень **all**, **lapb-ta**, **lat**, **mop**, **none**, **pad**, **rlogin**, **ssh**, **telnet**, **udptn**, **v120**:

**all** – всі протоколи;

**lapb-ta** – термінальний адаптер протоколу LAPB;

**lat** – протокол DEC LAT;

**mop** – протокол DEC MOP Remote Console Protocol;

**none** – жоден із протоколів;

**pad** – протокол X.3 PAD;

**rlogin** – протокол Rlogin;

**ssh** – протокол SSH;

**telnet** – протокол Telnet;

**udptn** – асинхронний UDPTN через UDP протокол;

**v120** – Асинхронне підключення через ISDN.

Синтаксис команди **transport output** (режим конфігурування лінії):

```
transport output {value | values}.
```

Параметри команди аналогічні параметрам попередньої команди.

Синтаксис команди **transport preffered** (режим конфігурування лінії):

```
transport preffered value.
```

Параметр команди аналогічний параметру **value** попередньої команди, за винятком значення **all**.

Синтаксис команди **rotary** (режим конфігурування лінії):

```
rotary value [queued [by-role [round-robin] | round-robin]] | round-robin [queued [by-role]] ,
```

де **value** – значення номера групи, що задається для номера порту; число з діапазону 0 ... 127.

**queued** – параметр, який вказує на необхідність використання черги, коли група заповнена;

**round-robin** – параметр, який вказує на необхідність кругового вибору;

**by-role** – параметр, який вказує на необхідність вибору за ролями.

Синтаксис команди **autocommand** (режим конфігурування лінії):

```
autocommand { LINE | no-suppress-linenumber LINE },
```

де **LINE** – текстовий рядок, який містить команду (режиму **EXEC**), що буде автоматично виконуватися;

**no-suppress-linenumber** – параметр, який призначений для активації виведення повідомлення.

Синтаксис команди **login block-for** (режим глобального конфігурування):

```
login block-for block_value attempts attempts_value
 within interval_value,
```

де **block\_value** – значення інтервалу часу (с), на який буде заблоковано можливість виконання підключення; може змінюватися у межах від 1 до 65535 с;

**attempts** – службова конструкція, за допомогою якої зазначається максимально можлива кількість спроб підключення на заданий інтервал часу;

**attempts\_value** – максимальне значення кількості спроб підключення на встановлений інтервал часу; може змінюватися у межах від 1 до 65535;

**within** – службова конструкція, за допомогою якої зазначається встановлений для можливої кількості спроб підключення інтервал часу;

**interval\_value** – значення інтервалу часу (с), на якому задається максимальне значення кількості спроб підключення; може змінюватися у межах від 1 до 65535 с.

Синтаксис команди **login delay** (режим глобального конфігурування):

```
login delay delay_value,
```

де **delay\_value** – значення інтервалу затримки (с) між спробами підключення; може змінюватися у межах від 1 до 10 с; за замовчуванням становить 1 с.

Синтаксис команди **login on-failure log** (режим глобального конфігурування):

```
login on-failure log [every login_value],
```

де **every** – службова конструкція, за допомогою якої зазначається необхідність виконання журналювання неуспішних спроб входу в систему;

**login\_value** – числове значення кількості спроб підключення, може змінюватися у межах від 1 до 65535.

Синтаксис команди **login on-success log** (режим глобального конфігурування):

```
login on-success log [every login_number],
```

де **every** – службова конструкція, за допомогою якої зазначається необхідність виконання журналювання успішних спроб входу в систему;

**login\_number** – числове значення кількості спроб підключення, може змінюватися у межах від 1 до 65535.

Синтаксис команди **login quiet-mode access-class** (режим глобального конфігурування):

```
login quiet-mode access-class { acl_name | acl_number },
```

де **acl\_name** – текстова назва списку доступу;

**acl\_number** – номер списку доступу.

Синтаксис команди **security authentication failure rate** (режим глобального конфігурування):

```
security authentication failure rate rate_value log,
```

де **rate\_value** – кількість дозволених невдалих спроб входу в систему (аутифікації) за хвилину, перевищення якої викличе генерацію повідомлення для журналювання подій;

**log** – службова конструкція, за допомогою якої активується можливість виконання журналювання подій, пов'язаних із невдалими спробами входу в систему.

### Команди налагодження функціонування протоколу *Telnet* на пристроях *Cisco*

Для налагодження параметрів функціонування протоколу *Telnet* на пристроях *Cisco* використовуються спеціалізовані команди **ip telnet:ip telnet comport,ip telnet hidden,ip telnet quiet,ip telnet source-interface,ip telnet timeout retransmit,ip telnet tos**. Команди групи **ip telnet comport** призначені для налагодження параметрів функціонування засобів протоколу *Telnet* згідно із *RFC-2217*. Команда **ip telnet hidden** призначена для відключення виведення *IP*-адрес або назв вузлів протоколу *Telnet*. Команда **ip telnet quiet** відповідає за відключення виведення непомилкових повідомлень. Команда **ip telnet source-interface** призначена для встановлення інтерфейсу виходу сеансів протоколу *Telnet*. Згадана команда застосовується, як правило, на маршрутизаторах та комутаторах третього рівня. Команда **ip telnet timeout retransmit** відповідає за встановлення значення тайм-ауту повторної передачі. Команда **ip telnet tos** застосовується для встановлення значення типу сервісу (*TOS, Type Of Service*). Призначення та синтаксис усіх вищезгаданих команд наведено нижче.

Синтаксис команди **ip telnet comport** (режим глобального конфігурування):

```
ip telnet comport { disconnect delay disconnect_value |
 enable | flow level flow_value | receive window
 window_value },
```

де ***disconnect\_value*** – значення інтервалу затримки перед закриттям TCP-з'єднання (с), число з діапазону 0 ... 360;

***flow\_value*** – кількість символів для буферизації на пристрої перед відправленням повідомлення *RFC-2217 SUSPEND*; число з діапазону 0 ... 1023;

***window\_value*** – максимальне значення вікна отримання TCP; число з діапазону 1 ... 4128.

Синтаксис команди **ip telnet hidden** (режим глобального конфігурування):

```
ip telnet hidden {addresses | hostnames},
```

де **addresses** – службова конструкція, за допомогою якої відключається введення адрес;

**hostnames** – службова конструкція, за допомогою якої відключається введення назв вузлів.

Синтаксис команди **ip telnet quiet** (режим глобального конфігурування):

```
ip telnet quiet.
```

Синтаксис команди **ip telnet source-interface** (режим глобального конфігурування):

```
ip telnet source-interface interface-type interface-id,
```

де ***interface-type*** – тип інтерфейсу, може набувати значень **Ethernet**, **FastEthernet**, **Gigabit Ethernet**, **Serial**, **Loopback**, **Tunnel**, **Vlan** та ін.;

***interface-id*** – ідентифікатор інтерфейсу, може мати одночислове позначення ***number*** (номер інтерфейсу), двочислове позначення ***module/number*** (номер модуля (адаптера)/номер інтерфейсу), тричислове позначення ***slot/module/number*** (номер слоту/номер модуля (адаптера)/номер інтерфейсу);

Синтаксис команди **ip telnet timeout retransmit** (режим глобального конфігурування):

```
ip telnet timeout retransmit retransmit-value,
```

де ***retransmit-value*** – значення інтервалу повторної передачі (с), число з діапазону 1 ... 2147483.

Синтаксис команди **ip telnet tos** (режим глобального конфігурування):

```
ip telnet tos tos-value,
```

де **tos-value** – значення параметра *TOS*, число з діапазону *0h ...FFh*.

### КОМАНДИ НАЛАГОДЖЕННЯ ФУНКЦІОНУВАННЯ ПРОТОКОЛУ *SSH* НА ПРИСТРОЯХ *CISCO*

Для налагодження параметрів функціонування протоколу *SSH* на пристроях *Cisco* використовуються команди **ip ssh authentication-retries**, **ip ssh break-string**, **ip ssh dh min size**, **ip ssh dscp**, **ip ssh logging events**, **ip ssh maxstartups**, **ip ssh port**, **ip ssh precedence**, **ip ssh rsa keypair-name**, **ip ssh source-interface**, **ip ssh time-out**, **ip ssh version**, **crypto key generate**, **crypto key generate rsa general-keys modulus**, **crypto key generate rsa usage-keys label** та деякі інші. Призначення та синтаксис усіх вищезгаданих команд наведено нижче.

Команда **ip ssh authentication-retries** призначена для встановлення кількості спроб аутентифікації, після якої *SSH*-клієнтові забороняється доступ. Команда **ip ssh break-string** відповідає за активацію обробки та встановлення зазначення текстового рядка, що передається *SSH*-клієнтом *SSH*-серверу, який надалі передає сигнал зупинки для асинхронного підключення. Команда **ip ssh logging events** призначена для активації журналювання подій протоколу *SSH*. Команда **ip ssh maxstartups** використовується для обмеження кількості сесій протоколу.

Команда **ip ssh port** відповідає за активацію безпечного доступу до пристрою через асинхронні підключення та встановлення початкового номера *TCP*-порту, що буде використовуватися для цих підключень. Команди **ip ssh dscp** та **ip ssh precedence** застосовуються для встановлення значень полів *DSCP* та поля *Precedence* *IP*-пакета, що переносить повідомлення протоколу *SSH*.

Команда **ip ssh source-interface** використовується для зазначення певного вихідного інтерфейсу для всіх *SSH*-сесій. Згадана команда застосовується, як правило, на маршрутизаторах та комутаторах третього рівня. Команда **ip ssh time-out** використовується для обмеження часу відповіді *SSH*-клієнта (*SSH*-сервер перериває з'єднання, якщо дані не передаються протягом часу очікування). Команда **ip ssh version** призначена для вказування версії протоколу *SSH*, що буде використовуватися у процесі роботи. За замовчуванням на пристроях *Cisco* активовано використання протоколу *SSH* версії 1. Відміна дії більшості команд **ip ssh** виконується формою **no**.

Команда **ip ssh rsa keypair-name** застосовується для зазначення назви ключа *RSA*, що буде використовуватися у процесі роботи протоколу *SSH*. Ключ може формуватися двома способами: або з використанням назви пристрою та назви домену або без їх використання. Також для робіт із ключами застосовується команда **ip ssh dh min size**, за допомогою якої задається номер групи

Діффі-Хеллмана для обміну ключами.

Для роботи з ключами використовуються команди групи **crypto key**. Для генерації ключів застосовуються команди **crypto key generate**, **crypto key generate rsa general-keys modulus**, **crypto key generate rsa usage-keys label**. Для видалення ключів призначені команди **crypto key zeroize**, **crypto key zeroize rsa**.

Слід звернути увагу, що однакового результату у процесі налагодження функціонування протоколу *SSH* на пристроях *Cisco* можна досягнути у разі використання різних команд. Детальний опис дії команд можна знайти в документації виробника.

Синтаксис команди **ip ssh authentication-retries** (режим глобального конфігурування):

```
ip ssh authentication-retries retries-value,
```

де **retries-value** – значення максимальної кількості спроб аутентифікації підряд, число з діапазону 0...5; за замовчуванням встановлюється 3 спроби.

Синтаксис команди **ip ssh break-string** (режим глобального конфігурування):

```
ip ssh break-string text-string,
```

де **text-string** – текстовий рядок сигналу зупинки, за замовчуванням не встановлено.

Синтаксис команди **ip ssh dh min size** (режим глобального конфігурування):

```
ip ssh dh min size dh_value,
```

де **dh\_value** – значення номер групи Діффі-Хеллмана, може набувати значень 1024 (група 1), 2048 (група 14), 4096 (група 16); за замовчуванням дорівнює 1024.

Синтаксис команди **ip ssh dscp** (режим глобального конфігурування):

```
ip ssh dscp dscp_value,
```

де **dscp\_value** – значення поля *DSCP* IP-пакета, може набувати значень від 0 до 63; за замовчуванням дорівнює 0.

Синтаксис команди **ip ssh logging events** (режим глобального конфігурування):

```
ip ssh logging events.
```

Синтаксис команди **ip ssh maxstartups** (режим глобального конфігурування):

```
ssh ip ssh maxstartups [max-value],
```

де **max-value** – значення кількості сесій протоколу, число з діапазону 0...128; за замовчуванням встановлюється 128 сесій.

Синтаксис команди **ip ssh port** (режим глобального конфігурування):

**ip ssh port** *port-number rotary group*,

де **port-number** – значення номера порту; число з діапазону 2000 ... 10000;  
**rotary** – службова конструкція, призначена для активації доступу з використанням значення **group**;

**group** – значення номера групи; число з діапазону 1 ... 127.

Синтаксис команди **ip ssh precedence** (режим глобального конфігурування):

**ip ssh precedence** *precedence\_value*,

де **precedence\_value** – значення поля *Precedence*, може набувати значень від 0 до 7; за замовчуванням дорівнює 0.

Синтаксис команди **ip ssh rsa keypair-name** (режим глобального конфігурування):

**ip ssh rsa keypair-name** *keypair-name-string*,

де **keypair-name-string** – текстовий рядок, який містить назву ключа RSA.

Синтаксис команди **ip ssh source-interface** (режим глобального конфігурування):

**ip ssh source-interface** *interface-type interface-id*,

де **interface-type** – тип інтерфейсу, може набувати значень **Serial**, **Ethernet**, **FastEthernet**, **Gigabit Ethernet**, **Serial**, **Port-channel**, **Tunnel** та ін.;

**interface-id** – ідентифікатор інтерфейсу, може мати одночислове позначення **number** (номер інтерфейсу), двочислове позначення **module/number** (номер модуля (адаптера)/номер інтерфейсу), тричислове позначення **slot/module/number** (номер слоту/номер модуля (адаптера)/номер інтерфейсу).

Синтаксис команди **ip ssh time-out** (режим глобального конфігурування):

**ip ssh time-out** *seconds*,

де **seconds** – значення інтервалу часу очікування відповіді клієнта (с), число з діапазону 1 ... 120; за замовчуванням встановлюється 120 с.

Синтаксис команди **ip ssh version** (режим глобального конфігурування):

**ip ssh version** *version-number*,

де **version-number** – номер версії протоколу, може набувати значень 1 або 2; якщо значення не встановлене, то функціонування протоколу здійснюється у змішаному режимі.

Синтаксис команди **crypto key generate** (режим глобального конфігурування):

**crypto key generate.**

Синтаксис команди **crypto key generate rsa general-keys modulus** (режим глобального конфігурування):

**crypto key generate rsa general-keys modulus** *modulus-value*,

де **modulus-value** – значення довжини ключа (бітів), число з діапазону 360 ... 2048; за замовчуванням генеруються ключі довжиною 512 бітів.

Синтаксис команди **crypto key generate rsa usage-keys label** (режим глобального конфігурування):

**crypto key generate rsa usage-keys label** *keypair-name-string* **modulus** *modulus-value*,

де **keypair-name-string** – текстовий рядок, який містить назву ключової пари RSA;

**modulus-value** – значення довжини ключа (бітів), число з діапазону 360 ... 2048; за замовчуванням генеруються ключі довжиною 512 бітів.

Синтаксис команди **crypto key zeroize** (режим глобального конфігурування):

**crypto key zeroize**

Синтаксис команди **crypto key zeroize rsa** (режим глобального конфігурування):

**crypto key zeroize rsa** *keypair-name-string*,

де **keypair-name-string** – текстовий рядок, який містить назву ключової пари RSA.

## ОСНОВНІ КОМАНДИ МОНІТОРИНГУ ТА ДІАГНОСТИКИ ФУНКЦІОНУВАННЯ ПРОТОКОЛІВ *TELNET* ТА *SSH* НА ПРИСТРОЯХ *CISCO*

Для перегляду параметрів налагоджень мережних підключень, параметрів роботи протоколів віддаленого доступу та інших параметрів використовуються різні варіанти команд **show**. Перелік основних команд та їх призначення наведені у таблиці 5.29. Для відстеження подій та повідомлень, які генеруються протоколом SSH використовуються команди **debug ip ssh** та **debug ip ssh client**.

Таблиця 5.29 – Перелік команд **show**, необхідних для діагностики параметрів мережних підключень та параметрів протоколів віддаленого доступу на пристроях *Cisco*

| Команда                                   | Призначення                                                                                                                                                                                              |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>show line</b>                          | Виведення інформації про наявні підключення та їх параметри                                                                                                                                              |
| <b>show sessions</b>                      | Виведення інформації про параметри вихідних сеансів                                                                                                                                                      |
| <b>show users</b>                         | Виведення інформації про параметри вхідних підключень (зокрема, мережних) та підключених користувачів                                                                                                    |
| <b>show ssh</b>                           | Виведення інформації про параметри сеансів протоколу <i>SSH</i>                                                                                                                                          |
| <b>show ip ssh</b>                        | Виведення інформації про налагоджені параметри протоколу <i>SSH</i> (версія, час тайм-ауту, кількість спроб аутентифікації тощо)                                                                         |
| <b>show login</b>                         | Виведення узагальненої інформації про налагодження входу в систему                                                                                                                                       |
| <b>show crypto key mypubkey rsa</b>       | Виведення публічного ключа <i>RSA</i>                                                                                                                                                                    |
| <b>show crypto key pubkey-chain rsa</b>   | Виведення публічного ключа <i>RSA</i> з'єднання, що збережений на комп'ютері.                                                                                                                            |
| <b>show crypto key storage</b>            | Виведення інформації про місце розміщення ключової пари                                                                                                                                                  |
| <b>show control-plane host open-ports</b> | Виведення інформації про відкриті <i>TCP</i> - та <i>UDP</i> -порти пристрою                                                                                                                             |
| <b>show tcp</b>                           | Виведення детальної інформації про встановлені з'єднання транспортного протоколу <i>TCP</i> (використовується для визначення <i>IP</i> -адрес з'єднань та номерів портів протоколів віддаленого доступу) |

### Модельний приклад налагодження віддаленого доступу до пристроїв *Cisco* з використанням протоколу *Telnet*

Розглянемо специфіку налагодження роботи протоколу віддаленого доступу *Telnet* для комунікаційних пристроїв мережі, схему якої наведено на рисунку 5.39. У цьому випадку підключення здійснюється з робочої станції **WS-Control** до маршрутизатора **R-1** та комутатора **SW-1**.

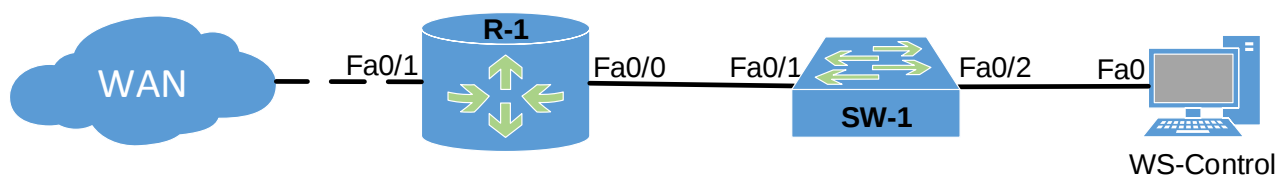


Рисунок 5.39– Приклад мережі

Під час побудови даної мережі для з'єднання пристроїв використано дані таблиці 5.30.

Таблиця 5.30 – Параметри інтерфейсів пристроїв для прикладу

| Пристрій          | Інтерфейс | Підключення до пристрою | Підключення до інтерфейсу |
|-------------------|-----------|-------------------------|---------------------------|
| Маршрутизатор R-1 | Fa0/0     | Комутатор SW-1          | Fa0/1                     |
|                   | Fa0/1     | WAN                     | WAN Interface             |
| Комутатор SW-1    | Fa0/1     | Маршрутизатор R-1       | Fa0/0                     |
|                   | Fa0/2     | WS-Control              | Fa0                       |
| WS-Control        | Fa0       | Комутатор SW-1          | Fa0/2                     |

Для налагодження параметрів адресації пристроїв використано дані таблиці 5.31.

Таблиця 5.31 – Параметри адресації мережі

| Підмережа/<br>Пристрій    | Інтерфейс/Мережний адаптер/Шлюз | IP-адреса    | Маска підмережі | Префікс |
|---------------------------|---------------------------------|--------------|-----------------|---------|
| Підмережа А               | –                               | 195.10.1.0   | 255.255.255.0   | /24     |
| WAN                       | –                               | 196.10.1.0   | 255.255.255.252 | /30     |
| Маршрутизатор R-1         | Інтерфейс Fa0/0                 | 195.10.1.254 | 255.255.255.0   | /24     |
|                           | Інтерфейс Fa0/1                 | 196.10.1.2   | 255.255.255.252 | /30     |
| Комутатор SW-1            | Інтерфейс Vlan 1                | 195.10.1.250 | 255.255.255.0   | /24     |
|                           | Шлюз за замовчуванням           | 195.10.1.254 | –               | –       |
| Робоча станція WS-Control | Мережний адаптер                | 195.10.1.1   | 255.255.255.0   | /24     |
|                           | Шлюз за замовчуванням           | 195.10.1.254 | –               | –       |

Для налагодження параметрів комунікаційних пристроїв із метою забезпечення підключення за протоколами *Telnet/SSH* використано дані таблиці 5.32.

Таблиця 5.32 – Параметри налагодження комунікаційних пристроїв

| Параметр                                                       | Значення                 |
|----------------------------------------------------------------|--------------------------|
| Системний час                                                  | Поточний                 |
| Часовий пояс                                                   | Східноєвропейський       |
| Перехід на літній час                                          | Україна                  |
| Банер                                                          | Connection to router R-1 |
| Кількість підключень VTU                                       | 5 (0 ... 4)              |
| Інтервал перед виведенням попередження про вихід із системи, с | 30                       |
| Загальна тривалість сеансу, хв                                 | 10                       |
| Синхронне виведення журнальних повідомлень на екран            | Активоване               |

Сценарій налагодження часових параметрів, повідомлення попередження та параметрів адресації інтерфейсів для маршрутизатора мережі **R-1** наведено нижче. Сценарій налагодження параметрів комутатора мережі **SW-1** подібний до сценарію для маршрутизатора **R-1**.

```
R-1>enable
R-1#clock set 10:04:00 11 dec 2022
R-1#configure terminal
R-1(config)#clock timezone EET 2
R-1(config)#clock summertime EET reccuring last monday october 3:00
last monday march 3:00
R-1(config)#banner motd # Connection to router R-1
 For legal users only
#
R-1(config)#interface FastEthernet 0/0
R-1(config-if)#description LINK-TO-LAN-A
R-1(config-if)#ip address 195.10.1.254 255.255.255.0
R-1(config-if)#no shutdown
R-1(config-if)#exit
R-1(config)#interface FastEthernet 0/1
R-1(config-if)#description LINK_TO_WAN
R-1(config-if)#ip address 196.10.1.2 255.255.255.252
R-1(config-if)#no shutdown
R-1(config-if)#exit
R-1(config)#exit
```

Сценарій налагодження віддаленого підключення за протоколом *Telnet* із входом без пароля (без аутентифікації) для маршрутизатора мережі **R-1** наведений нижче. Слід зазначити, що у даному сценарії передбачено прямий перехід у привілейований режим за рахунок встановлення найвищого рівня привілеїв. Сценарій налагодження параметрів комутатора мережі **SW-1** подібний до сценарію для маршрутизатора **R-1**. У практиці експлуатації мереж такий сценарій має обмежене застосування, його рекомендовано застосовувати лише у випадку, коли мережна інфраструктура є надійно захищеною. Якщо існує ймовірність перехоплення інформації під час підключення, то розглянутий сценарій застосовувати не рекомендується.

```
R-1>enable
R-1#configure terminal
R-1(config)#line vty 0 4
R-1(config-line)#no login
R-1(config-line)#transport input telnet
R-1(config-line)#privilege level 15
R-1(config-line)#logout-warning 30
R-1(config-line)#absolute-timeout 10
R-1(config-line)#logging synchronous
R-1(config-line)#exit
R-1(config)#exit
```

Сценарій підключення/відключення до маршрутизатора **R-1** з робочої станції ОС *Windows* за допомогою вбудованого термінального додатка *Telnet* наведено нижче.

```
C:>telnet
Welcome to Microsoft Telnet Client

Escape Character is '^]'

Microsoft Telnet>open 195.10.1.254
Connection to 195.10.1.254...
Connection to router R-1
For legal users only
R-1#...
...
R-1#exit
Connection to the node has been lost
Press any key ...
Microsoft Telnet>quit
```

Сценарій налагодження віддаленого підключення за протоколом *Telnet* до маршрутизатора *Cisco* з використанням засобів локальної автентифікації на базі механізму паролів на вхід до відповідних командних режимів пристрою наведений нижче. У даному сценарії застосовані паролі типу 7.

```
R-1>enable
R-1#configure terminal
R-1(config)#service password-encryption
R-1(config)#enable secret adminpass2
R-1(config)#line vty 0 4
R-1(config-line)#password adminpass1
R-1(config-line)#login
R-1(config-line)#transport input telnet
R-1(config-line)#logout-warning 30
R-1(config-line)#absolute-timeout 10
R-1(config-line)#logging synchronous
R-1(config-line)#exit
R-1(config)#exit
```

Сценарій підключення/відключення до маршрутизатора **R-1** із робочої станції ОС *Windows* за допомогою вбудованого термінального додатка *Telnet* наведено нижче. Слід зазначити, що пароль під час введення не відображається.

```
C:>telnet 195.10.1.254
User Access Verificaton
Password:
Connection to router R-1
For legal users only
R-1>enable
```

```
Password:
R-1#...
R-1#exit
Connection to the node has been lost.
```

Сценарій налагодження віддаленого підключення за протоколом *Telnet* до маршрутизатора *Cisco* з використанням засобів локальної автентифікації на базі механізму користувачів наведений нижче. У даному сценарії застосовані паролі типу 5.

```
R-1>enable
R-1#configure terminal
R-1(config)#username adminer privilege 15 secret adminerpass
R-1(config)#username technic privilege 1 secret technicpass
R-1(config)#enable secret adminerpass2
R-1(config)#line vty 0 4
R-1(config-line)#login local
R-1(config-line)#transport input telnet
R-1(config-line)#logout-warning 30
R-1(config-line)#absolute-timeout 10
R-1(config-line)#logging synchronous
R-1(config-line)#exit
```

Сценарій підключення/відключення до маршрутизатора **R-1** з робочої станції *Windows* за допомогою вбудованого термінального додатка *Telnet* наведено нижче. Слід зазначити, що пароль при введенні не відображається.

```
C:>telnet 195.10.1.254
User Access Verificaton
Username:adminer
Password:
Connection to router R-1
For legal users only
R-1#...
...
R-1#exit
Connection to the node has been lost.
```

### Результати виконання команд моніторингу та діагностики роботи протоколу віддаленого доступу *Telnet* для розглянутого прикладу

З метою перегляду інформації про роботу мережних підключень, параметрів роботи протоколів віддаленого доступу та інших параметрів використовуються як загальні, так і специфічні для певного протоколу команди. Для розглянутого прикладу використано загальні команди **show line**, **show users**, **show tcp**. Результати роботи цих команд для маршрутизатора R-1 наведено

відповідно далі. Для перевірки підключення з боку робочої станції *Windows* використано команду **netstat -n**.

Результати роботи цієї команди наведено далі:

R-1#show line

|   | Tty | Typ | Tx/Rx     | A | Modem | Roty | AccO | AccI | Uses | Noise | Overruns | Int |
|---|-----|-----|-----------|---|-------|------|------|------|------|-------|----------|-----|
| * | 0   | CTY |           | - | -     | -    | -    | -    | 0    | 0     | 0/0      | -   |
|   | 1   | AUX | 9600/9600 | - | -     | -    | -    | -    | 0    | 0     | 0/0      | -   |
| * | 2   | VTY |           | - | -     | -    | -    | -    | 2    | 0     | 0/0      | -   |
|   | 3   | VTY |           | - | -     | -    | -    | -    | 0    | 0     | 0/0      | -   |
|   | 4   | VTY |           | - | -     | -    | -    | -    | 0    | 0     | 0/0      | -   |
|   | 5   | VTY |           | - | -     | -    | -    | -    | 0    | 0     | 0/0      | -   |
|   | 6   | VTY |           | - | -     | -    | -    | -    | 0    | 0     | 0/0      | -   |

R-1#show users

|   | Line      | User    | Host(s) | Idle     | Location     |
|---|-----------|---------|---------|----------|--------------|
| * | 0 con 0   |         | idle    | 00:00:00 |              |
|   | 2 vty 0   | adminer | idle    | 00:00:03 | 195.10.1.1   |
|   | Interface | User    | Mode    | Idle     | Peer Address |

R-1#show tcp

```
tty2, virtual tty from host 195.10.1.1
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 195.10.1.254, Local port: 23
Foreign host: 195.10.1.1, Foreign port: 1030
```

C:>netstat -n

Active Connections

| Proto | Local Address   | Foreign Address | State       |
|-------|-----------------|-----------------|-------------|
| TCP   | 195.10.1.1:1030 | 195.10.1.254:23 | ESTABLISHED |

### Модельний приклад налагодження віддаленого доступу до пристроїв *Cisco* з використанням протоколу *SSH*

Розглянемо специфіку налагодження роботи протоколу віддаленого доступу *SSH* для комунікаційних пристроїв мережі, схема якої наведена на рисунку 5.39. У даному випадку підключення здійснюється з робочої станції **WS-Control** до маршрутизатора **R-1** та комутатора **SW-1**. Під час побудови даної мережі для з'єднання пристроїв використано дані таблиці 5.30. Для налагодження параметрів адресації пристроїв використано дані таблиці 5.31. Для налагодження параметрів комунікаційних пристроїв з метою забезпечення підключення за протоколом *SSH* використано дані таблиці 5.32.

Сценарій налагодження часових параметрів, повідомлення попередження та параметрів адресації інтерфейсів для маршрутизатора мережі **R-1** аналогічний сценарію попереднього модельного прикладу. Сценарій налагодження параметрів комутатора мережі **SW-1** подібний до сценарію для маршрутизатора **R-1**.

Сценарій налагодження віддаленого підключення за протоколом *SSH* до маршрутизатора *Cisco* з використанням імені пристрою та імені домену та з

використанням засобів локальної аутентифікації на базі механізму користувачів наведений нижче. У цьому сценарії застосовані паролі типу 5.

```
R-1>enable
R-1#configure terminal
R-1(config)#username adminer privilege 15 secret adminerpass
R-1(config)#username technic privilege 1 secret technicpass
R-1(config)#enable secret adminerpass2
R-1(config)#ip domain-name mynet.net
R-1(config)#crypto key generate rsa general-keys modulus 1024
R-1(config)#ip ssh version 2
R-1(config)#line vty 0 4
R-1(config-line)#login local
R-1(config-line)#transport input ssh
R-1(config-line)#transport output ssh
R-1(config-line)#logout-warning 30
R-1(config-line)#absolute-timeout 10
R-1(config-line)#logging synchronous
R-1(config-line)#exit
R-1(config)#exit
R-1#exit
R-1>
```

Сценарій налагодження елементів захисту від атак на пристрій та налагодження підсистеми журналювання подій, пов'язаних із вдалими та невдалими спробами SSH-підключень.

```
R-1>enable
R-1#configure terminal
R-1(config)#login block-for 300 attempts 3 within 3
R-1(config)#login delay 5
R-1(config)#login on-failure log
R-1(config)#login on-success log
R-1(config)#ip ssh time-out 60
R-1(config)#ip ssh authentication-retries 5
R-1(config)#ip ssh maxstartups 5
R-1(config)#ip ssh logging events
R-1(config)#exit
R-1#exit
```

Для підключення до маршрутизатора **R-1** з робочої станції ОС *Windows* використано термінальний додаток *Putty* (*SuperPutty*). Сценарій підключення/відключення за допомогою цього додатка наведений нижче:

```
login as: adminer
Using keyboard-interactive authentication.
Password:
Connection to router R-1
For legal users only
R-1#...
...
R-1#exit
```

Сценарій налагодження віддаленого підключення за протоколом *SSH* до маршрутизатора *Cisco* з використанням ключових пар *RSA* та з використанням засобів локальної аутентифікації на базі механізму користувачів наведений нижче.

```
R-1>enable
R-1#configure terminal
R-1(config)#username adminer privilege 15 secret adminerpass
R-1(config)#username technic privilege 1 secret technicpass
R-1(config)#enable secret adminerpass2
R-1(config)#ip ssh rsa keypair-name MySSHkeys
R-1(config)#crypto key generate rsa usage-keys label MySSHkeys modulus 1024
R-1(config)#ip ssh version 2
R-1(config)#line vty 0 4
R-1(config-line)#login local
R-1(config-line)#transport input ssh
R-1(config-line)#transport output ssh
R-1(config-line)#exit
R-1(config)#exit
R-1#
```

У *Cisco IOS* існує можливість виконання підключення з одного пристрою до іншого. Для цього застосовуються вбудовані *Cisco IOS Telnet* та *SSH*-клієнти. Приклад сценарію такого підключення з комутатора **SW-1** до маршрутизатора **R-1** наведений нижче:

```
SW-1#ssh -v 2 -l adminer 195.10.1.1
Password:
Connection to router R-1
For legal users only
R-1#
```

Слід зазначити, що для виконання подібного сценарію на пристрої повинна бути активована можливість виконання віддалених підключень до інших пристроїв. Ця можливість активується командою **transport output**, у якій зазначається відповідний мережний протокол.

### Результати виконання команд моніторингу та діагностики роботи протоколу віддаленого доступу *SSH* для розглянутого прикладу

З метою перегляду інформації про роботу мережних підключень, параметрів роботи протоколів віддаленого доступу та інших параметрів використовуються як загальні, так і специфічні для певного протоколу команди. Для розглянутого прикладу використано загальні команди **show line**, **show users**, **show tcp** та специфічні команди протоколу *SSH* **show ssh**, **show ip ssh**, **show crypto key mypubkey rsa**. Результати роботи цих команд для маршрутизатора **R-1** (за умови використання імені пристрою та імені домену для підключення) наведено відповідно далі:

R-1#show line

|   | Tty | Type | Tx/Rx     | A | Modem | Roty | AccO | AccI | Uses | Noise | Overruns | Int |
|---|-----|------|-----------|---|-------|------|------|------|------|-------|----------|-----|
| * | 0   | CTY  | -         | - | -     | -    | 0    | 0    | 0/0  | -     |          |     |
|   | 1   | AUX  | 9600/9600 | - | -     | -    | -    | -    | 0    | 0     | 0/0      | -   |
| * | 2   | VTY  |           | - | -     | -    | -    | -    | 2    | 0     | 0/0      | -   |
|   | 3   | VTY  |           | - | -     | -    | -    | -    | 0    | 0     | 0/0      | -   |
|   | 4   | VTY  |           | - | -     | -    | -    | -    | 0    | 0     | 0/0      | -   |
|   | 5   | VTY  |           | - | -     | -    | -    | -    | 0    | 0     | 0/0      | -   |
|   | 6   | VTY  |           | - | -     | -    | -    | -    | 0    | 0     | 0/0      | -   |

R-1#show users

| Line      | User    | Host(s) | Idle     | Location   |
|-----------|---------|---------|----------|------------|
| * 0 con 0 |         | idle    | 00:00:00 |            |
| 2 vty 0   | adminer | idle    | 00:02:57 | 195.10.1.1 |

| Interface | User | Mode | Idle | Peer Address |
|-----------|------|------|------|--------------|
|-----------|------|------|------|--------------|

R-1#show tcp

tty2, virtual tty from host 195.10.1.1  
Connection state is ESTAB, I/O status: 1, unread input bytes: 0  
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255  
Local host: 195.10.1.254, Local port: 22  
Foreign host: 195.10.1.1, Foreign port: 1030

R-1#show ssh

| Connection | Version | Mode | Encryption | Hmac      | State           | Username |
|------------|---------|------|------------|-----------|-----------------|----------|
| 0          | 2.0     | IN   | aes256-cbc | hmac-sha1 | Session started | adminer  |
| 0          | 2.0     | OUT  | aes256-cbc | hmac-sha1 | Session started | adminer  |

%No SSHv1 server connections running.

R-1#show ip ssh

SSH Enabled - version 2.0  
Authentication timeout: 60 secs; Authentication retries: 5  
Minimum expected Diffie Hellman key size : 1024 bits

R-1#show crypto key mypubkey rsa

% Key pair was generated at: 22:49:37 UTC Jan 18 2017  
Key name: R-1.mynet.net  
Storage Device: not specified  
Usage: General Purpose Key  
Key is not exportable.  
Key Data:  
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00B1BDCC  
97462A4E F581491B A8F0A289 8E2EBA0C 247844A8 B53BCF0A 94700F5D 496E9E71  
541451BD C2BFFD6F 10FEFC02 5A18712C 7755EF54 1816AD97 F11F4694 8EE277A0  
FFA1F692 3D140EC6 813433A7 22C9A27C 72F3911F 6904313B 5919AA43 F086EA34  
1F4625A6 C50CC5AF CA072709 E94E2DFF 884564C2 00D43780 6D43475C 45020301 0001  
% Key pair was generated at: 22:49:38 UTC Jan 18 2017  
Key name: R-1.mynet.net.server  
Temporary key  
Usage: Encryption Key  
Key is not exportable.  
Key Data:  
307C300D 06092A86 4886F70D 01010105 00036B00 30680261 0091A903 3A23444A  
0A4DAB64 6C1C3250 2661F503 328A9A3B 54B618E2 8FF0CF0B 6461FF1E CEDD0DB6  
4BC54A5B BD97A159 B6C82E7E E5A560C9 CCCFDB40 B7F8F898 EE3CC2E5 F1B2B716  
741B9063 DE7EB172 E2C59F81 F851A229 99450322 D2224140 91020301 0001

Для перевірки підключення з боку робочої станції Windows використано команду **netstat -n**. Результати роботи цієї команди наведено далі:

```
C:>netstat -n
```

Активные подключения

| Имя | Локальный адрес | Внешний адрес   | Состояние   |
|-----|-----------------|-----------------|-------------|
| TCP | 195.10.1.1:1030 | 195.10.1.254:22 | ESTABLISHED |

## ПРАКТИЧНЕ ЗАВДАННЯ

**Мета роботи:** ознайомитися з особливостями функціонування протоколів та засобів віддаленого доступу та адміністрування; отримати практичні навички налагодження, моніторингу та діагностування засобів віддаленого доступу та адміністрування сучасних ОС; дослідити можливості ОС Windows, Linux, Cisco IOS з організації, налагодження та функціонування незахищених та захищених віддалених мережних підключень на базі протоколів Telnet та SSH.

### Порядок виконання роботи

1. У середовищі програмного симулятора/емулятора створити проект мережі (рисунок 5.40). Під час побудови звернути увагу на вибір моделей комутаторів та маршрутизаторів, мережних модулів та адаптерів, а також мережних з'єднань. Для побудованої мережі заповнити описову таблицю, яка аналогічна таблиці 5.30.

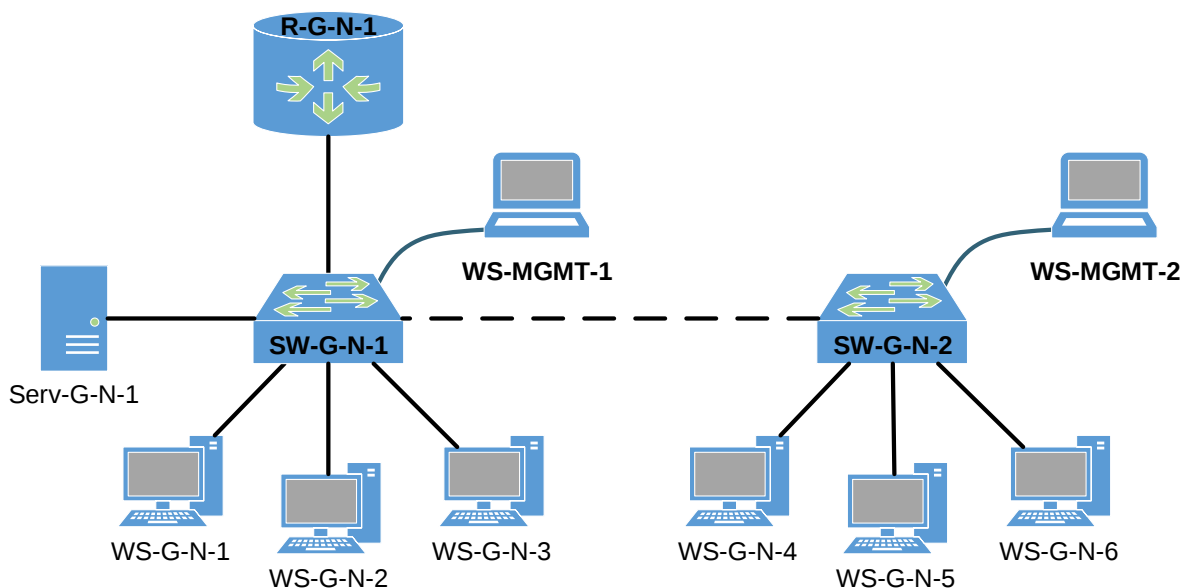


Рисунок 5.40 – Проект мережі

2. Розробити схему адресації пристроїв мережі. Для цього використовувати дані таблиці 5.33. Результати навести у вигляді таблиці, яка аналогічна таблиці 5.31.

3. Провести базове налагодження пристроїв, інтерфейсів та каналів зв'язку. Провести налагодження параметрів IP-адресації пристроїв мережі відповідно до даних, які отримані у п. 2.

4. Перевірити наявність зв'язку між всіма пристроями мережі.

Таблиця 5.33 – Параметри IP-адресації мережі

| № варіанта | IP-адреса мережі А | Префікс | IP-адреса шлюзу за замовчуванням/<br>IP-адреса DNS-сервера |
|------------|--------------------|---------|------------------------------------------------------------|
| 1          | 191.G.N.0          | /24     | Перша IP-адреса діапазону                                  |
| 2          | 192.G.N.0          | /25     | Остання IP-адреса діапазону                                |
| 3          | 193.G.N.0          | /26     | Перша IP-адреса діапазону                                  |
| 4          | 194.G.N.0          | /27     | Остання IP-адреса діапазону                                |
| 5          | 195.G.N.0          | /28     | Перша IP-адреса діапазону                                  |
| 6          | 196.G.N.0          | /24     | Остання IP-адреса діапазону                                |
| 7          | 197.G.N.0          | /25     | Перша IP-адреса діапазону                                  |
| 8          | 198.G.N.0          | /26     | Остання IP-адреса діапазону                                |
| 9          | 199.G.N.0          | /27     | Перша IP-адреса діапазону                                  |
| 10         | 200.G.N.0          | /28     | Остання IP-адреса діапазону                                |
| 11         | 201.G.N.0          | /24     | Перша IP-адреса діапазону                                  |
| 12         | 202.G.N.0          | /25     | Остання IP-адреса діапазону                                |
| 13         | 203.G.N.0          | /26     | Перша IP-адреса діапазону                                  |
| 14         | 204.G.N.0          | /27     | Остання IP-адреса діапазону                                |
| 15         | 205.G.N.0          | /28     | Перша IP-адреса діапазону                                  |
| 16         | 206.G.N.0          | /24     | Остання IP-адреса діапазону                                |
| 17         | 207.G.N.0          | /25     | Перша IP-адреса діапазону                                  |
| 18         | 208.G.N.0          | /26     | Остання IP-адреса діапазону                                |
| 19         | 209.G.N.0          | /27     | Перша IP-адреса діапазону                                  |
| 20         | 210.G.N.0          | /28     | Остання IP-адреса діапазону                                |
| 21         | 211.G.N.0          | /24     | Перша IP-адреса діапазону                                  |
| 22         | 212.G.N.0          | /25     | Остання IP-адреса діапазону                                |
| 23         | 213.G.N.0          | /26     | Перша IP-адреса діапазону                                  |
| 24         | 214.G.N.0          | /27     | Остання IP-адреса діапазону                                |
| 25         | 215.G.N.0          | /28     | Перша IP-адреса діапазону                                  |
| 26         | 216.G.N.0          | /24     | Остання IP-адреса діапазону                                |
| 27         | 217.G.N.0          | /25     | Перша IP-адреса діапазону                                  |
| 28         | 218.G.N.0          | /26     | Остання IP-адреса діапазону                                |
| 29         | 219.G.N.0          | /27     | Перша IP-адреса діапазону                                  |
| 30         | 220.G.N.0          | /28     | Остання IP-адреса діапазону                                |

5. Провести налагодження віддаленого доступу до пристроїв мережі згідно з даними таблиці 5.34 (за потреби створити користувачів на пристроях, рівень їх привілеїв встановити довільним чином).

Таблиця 5.34 – Дані для вибору протоколів віддаленого доступу

| №<br>варіанта | Протоколи віддаленого доступу |             |             |
|---------------|-------------------------------|-------------|-------------|
|               | R-G-N-1                       | SW-G-N-1    | SW-G-N-2    |
| 1             | Telnet&Pwd                    | Telnet&User | SSHv1       |
| 2             | Telnet&User                   | SSHv1       | Telnet&Pwd  |
| 3             | SSHv1                         | Telnet&Pwd  | Telnet&User |
| 4             | Telnet&Pwd                    | Telnet&User | SSHv2       |
| 5             | Telnet&User                   | SSHv2       | Telnet&Pwd  |
| 6             | SSHv2                         | Telnet&Pwd  | Telnet&User |
| 7             | Telnet&Pwd                    | Telnet&User | SSHv1       |
| 8             | Telnet&User                   | SSHv1       | Telnet&Pwd  |
| 9             | SSHv1                         | Telnet&Pwd  | Telnet&User |
| 10            | Telnet&Pwd                    | Telnet&User | SSHv2       |
| 11            | Telnet&User                   | SSHv2       | Telnet&Pwd  |
| 12            | SSHv2                         | Telnet&Pwd  | Telnet&User |
| 13            | Telnet&Pwd                    | Telnet&User | SSHv1       |
| 14            | Telnet&User                   | SSHv1       | Telnet&Pwd  |
| 15            | SSHv1                         | Telnet&Pwd  | Telnet&User |
| 16            | Telnet&Pwd                    | Telnet&User | SSHv2       |
| 17            | Telnet&User                   | SSHv2       | Telnet&Pwd  |
| 18            | SSHv2                         | Telnet&Pwd  | Telnet&User |
| 19            | Telnet&Pwd                    | Telnet&User | SSHv1       |
| 20            | Telnet&User                   | SSHv1       | Telnet&Pwd  |
| 21            | SSHv1                         | Telnet&Pwd  | Telnet&User |
| 22            | Telnet&Pwd                    | Telnet&User | SSHv2       |
| 23            | Telnet&User                   | SSHv2       | Telnet&Pwd  |
| 24            | SSHv2                         | Telnet&Pwd  | Telnet&User |
| 25            | Telnet&Pwd                    | Telnet&User | SSHv1       |
| 26            | Telnet&User                   | SSHv1       | Telnet&Pwd  |
| 27            | SSHv1                         | Telnet&Pwd  | Telnet&User |
| 28            | Telnet&Pwd                    | Telnet&User | SSHv2       |
| 29            | Telnet&User                   | SSHv2       | Telnet&Pwd  |
| 30            | SSHv2                         | Telnet&Pwd  | Telnet&User |

*Примітка:* *Telnet&Pwd* – підключення за протоколом *Telnet* із використанням засобів локальної аутентифікації на базі механізму паролів на вхід до відповідних командних режимів; *Telnet&User* – підключення за протоколом *Telnet* із використанням засобів локальної аутентифікації на базі механізму користувачів; *SSHv1*, *SSHv2* – підключення за протоколом *SSH* відповідних версій з використанням засобів локальної аутентифікації на базі механізму користувачів.

6. Дослідити процеси віддаленого доступу до налагоджених у п. 5 комунікаційних пристроїв. У разі відсутності доступу визначити проблеми та усунути їх.

7. Для маршрутизатора мережі, на якому налагоджено підключення з використанням засобів локальної аутентифікації на базі механізму користувачів, налагодити можливість підключення як за допомогою протоколу *Telnet*, так і за допомогою протоколу *SSH*. Дослідити можливості підключення до налагодженого пристрою за допомогою додатка *Putty* або подібного.

8. Дослідити та проаналізувати відмінності віддаленого доступу за протоколом *Telnet* і за протоколом *SSH* у розрізі передачі даних аутентифікації та передачі даних сеансу зв'язку. Для перехоплення повідомлень використати штатні засоби програмного симулятора/емулятора або програмного аналізатор трафіка *WireShark* (за можливості).

### КОНТРОЛЬНІ ПИТАННЯ

1. Поняття та призначення протоколу віддаленого доступу.
2. Основні протоколи віддаленого доступу.
3. Загальна характеристика протоколів *Telnet* та *SSH*.
4. Сфера застосування протоколів *Telnet* та *SSH*.
5. Стандартизація протоколів *Telnet* та *SSH*.
6. Характеристики протоколів *Telnet* та *SSH* стосовно моделі *OSI* та стеку *TCP/IP*.
7. Рівні протоколу *SSH*.
8. Характеристика рівня безпеки протоколу *Telnet*.
9. Характеристика рівня безпеки протоколу *SSH*.
10. Реалізація протоколів *Telnet* та *SSH* у сучасних ОС.
11. Реалізація протоколів *Telnet* та *SSH* провідними виробниками мережного обладнання.
12. Перелік та призначення основних команд для налагодження протоколу *Telnet* на пристроях *Cisco*.
13. Перелік та призначення основних команд моніторингу роботи протоколу *Telnet* на пристроях *Cisco*.
14. Перелік та призначення основних команд для налагодження протоколу *SSH* на пристроях *Cisco*.
15. Перелік та призначення основних команд моніторингу роботи протоколу *SSH* на пристроях *Cisco*.

## 5.2 ПРАКТИЧНИЙ МОДУЛЬ 2

### 5.2.1 НАЛАШТУВАННЯ VLAN ТА ТРАНКОВИХ КАНАЛІВ

#### ОСНОВНІ КОМАНДИ НАЛАШТУВАННЯ VLAN

Локальна мережа, яка створена за допомогою одних тільки комутаторів, представляє один широкомовний домен. Зменшити такий домен можна, фізично розділивши локальну мережу на незалежні підмережі (незалежні групи попарно пов'язаних комутаторів) і з'єднати їх в єдине ціле з використанням маршрутизаторів. Таке завдання можна вирішити за допомогою віртуальних локальних мереж *VLAN (virtual local area network)*.

VLAN поводяться так само, як і фізично розділені локальні мережі. Тобто після розбиття мережі на VLAN одержується декілька локальних мереж, які далі необхідно об'єднати в єдине ціле за допомогою маршрутизації на третьому мережному рівні.

VLAN за замовчуванням (*Default VLAN*) на комутаторі Cisco - це VLAN 1. Всі порти комутатора належать до VLAN 1 до того моменту, поки вони явно не будуть налаштовані для участі у інших VLAN. Увесь трафік управління каналного рівня пов'язаний з VLAN 1.

Слід пам'ятати про VLAN 1 такі важливі факти:

- За замовчуванням всі порти комутатора належать до VLAN 1.
- За замовчуванням VLAN з нетегованим трафіком (*Native VLAN*) - це VLAN 1.
- За замовчуванням управлінська VLAN (*Management VLAN*) - це VLAN 1.
- Не можна перейменувати або видалити VLAN 1.

Номери VLAN (*VLAN ID*) можуть бути в діапазоні від 1 до 4094:

- 1 - 1005 базовий діапазон (*normal-range*);
- 1002 - 1005 зарезервовані для *Token Ring* і *FDDI VLAN*;
- 1006 - 4094 розширений діапазон (*extended-range*).

Основними командами налаштування VLAN є команди: **vlan**, **switchport access vlan**, **show vlan**, **switchport mode trunk**, **show interfaces trunk**.

Щоб додати VLAN і увійти у підрежим **config-VLAN**, скористайтеся командою **vlan** у режимі глобального конфігурування.

```
vlan {vlan-id | vlan-range}
```

де **vlan-id** – номер VLAN; допустимі значення від 1 до 4094;

**vlan-range** – діапазон налаштованих VLAN

Коли ви вводите команду **vlan**, створюється нова VLAN з усіма параметрами за замовчуванням у тимчасовому буфері і змушує CLI перейти у підрежим **config-VLAN**. Якщо введений вами ідентифікатор **vlan-id** збігається з

існуючою VLAN, будь-які команди конфігурації, які ви введете у підрежимі `config-VLAN`, будуть застосовані до існуючої VLAN.

Щоб видалити VLAN, скористайтеся командою **no vlan**.

```
no vlan {vlan-id | vlan-range}
```

де **vlan-id** – номер VLAN; допустимі значення від 1 до 4094;

**vlan-range** – діапазон налаштованих VLAN

Команда **switchport access vlan** приєднує інтерфейс 2-го рівня на пристрої *Cisco IOS* до вказаної VLAN. Ця команда діє лише для інтерфейсів, які працюють у режимі доступу. Процес налаштування інтерфейсу 2-го рівня як порту доступу, призначеного до певної VLAN, складається з 2 кроків:

- налаштування інтерфейсу на роботу в режимі доступу за допомогою команди **switchport mode access**;
- приєднання інтерфейсу до потрібної VLAN за допомогою команди **switchport access vlan**.

Формат команди наступний:

```
switchport access vlan vlan-id
```

де **vlan-id** – бажана VLAN, до якої повинен належати інтерфейс.

Щоб повернути порт або діапазон портів до VLAN за замовчуванням, введіть наступне:

```
no switchport access vlan
```

Для відображення інформації про VLAN скористайтеся командою **show vlan** у привілейованому режимі **EXEC**.

```
show vlan [brief | id vlan-id | name name [ifindex]
| ifindex]
```

де **brief** – відображає лише один рядок для кожної VLAN з назвою VLAN, статусом і портами.

**id vlan-id** – показує інформацію про окрему VLAN, яка ідентифікується ідентифікаційним номером VLAN; допустимі значення від 1 до 4094.

**name name** – показує інформацію про окрему VLAN, яка ідентифікується назвою VLAN; допустимі значення - ASCII-рядок від 1 до 32 символів.

**ifindex** – відображає номер *ifIndex* VLAN.

Використання VLAN було б не дуже корисним без транкових (магістральних) каналів VLAN. Магістральні канали VLAN дають змогу передавати увесь трафік VLAN між комутаторами. Вони дають змогу пристроям, які підключені до різних комутаторів, але належать одній VLAN, взаємодіяти без передачі трафіку через маршрутизатор.

Магістральний канал - це двоточковий канал зв'язку між мережними пристроями, який дає змогу передавати трафік більше ніж однієї VLAN.

Магістральний канал *VLAN* поширює *VLAN* по всій мережі. *Cisco* підтримує стандарт *IEEE 802.1Q* для узгодження магістральних каналів на інтерфейсах технологій *Fast Ethernet*, *Gigabit Ethernet* та *10-Gigabit Ethernet*.

Магістральний канал *VLAN* не належить до певної *VLAN*. Він є каналом передавання трафіку декількох *VLAN* між комутаторами і маршрутизаторами. Магістральний канал також може використовуватися між мережним пристроєм і сервером або іншим пристроєм, який оснащений мережною платою з підтримкою *802.1Q*.

У налаштуваннях інтерфейсу скористайтеся командою **switchport mode**, щоб налаштувати інтерфейс в режим магістралі (транку).

```
switchport mode {dynamic {auto | desirable} | trunk}
```

де **dynamic auto** – переводить інтерфейс у режим магістралі, якщо сусідній інтерфейс налаштовано у магістральний або бажаний режим. Це значення за замовчуванням.

**dynamic desirable** – переводить інтерфейс у режим магістралі, якщо сусідній інтерфейс перебуває у режимі магістралі.

**trunk** – переводить інтерфейс у постійний режим магістралі і веде переговори про перетворення з'єднання на магістраль, навіть якщо сусідній інтерфейс не є магістральним.

Щоб повернути порт до стандартної *VLAN*, введіть команду:

```
no switchport mode trunk
```

Використовуйте команду **switchport trunk allowed vlan**, щоб вказати, до яких мереж *VLAN* належить порт, коли його режим налаштовано як магістральний.

```
switchport trunk allowed vlan {all | none | add vlan-list
| remove vlan-list | except vlan-list}
```

де **all** – вказує всі *VLAN* від 1 до 4094. Порт належить до всіх *VLAN*, існуючих на даний момент.

**none** - вказує порожній список *VLAN*. Порт не належить жодній *VLAN*.

**add vlan-list** – список ідентифікаторів *VLAN* для додавання до порту. Розділяйте непослідовні ідентифікатори *VLAN* комами без пробілів. Використовуйте дефіс, щоб позначити діапазон ідентифікаторів.

**remove vlan-list** – список ідентифікаторів *VLAN* для видалення з порту. Розділяйте непослідовні ідентифікатори *VLAN* комами без пробілів. Використовуйте дефіс, щоб позначити діапазон ідентифікаторів.

**except vlan-list** – список ідентифікаторів *VLAN*, що включає всі *VLAN* з діапазону 1-4094, за винятком *VLAN*, що належать до **vlan-list**.

Виконайте команду **show interfaces trunk** для перегляду режимів магістральних інтерфейсів.

### Модельний приклад налагодження VLAN

Схему мережі з особливостями налагодження параметрів VLAN на комутаторах Cisco наведено на рисунку 5.41.

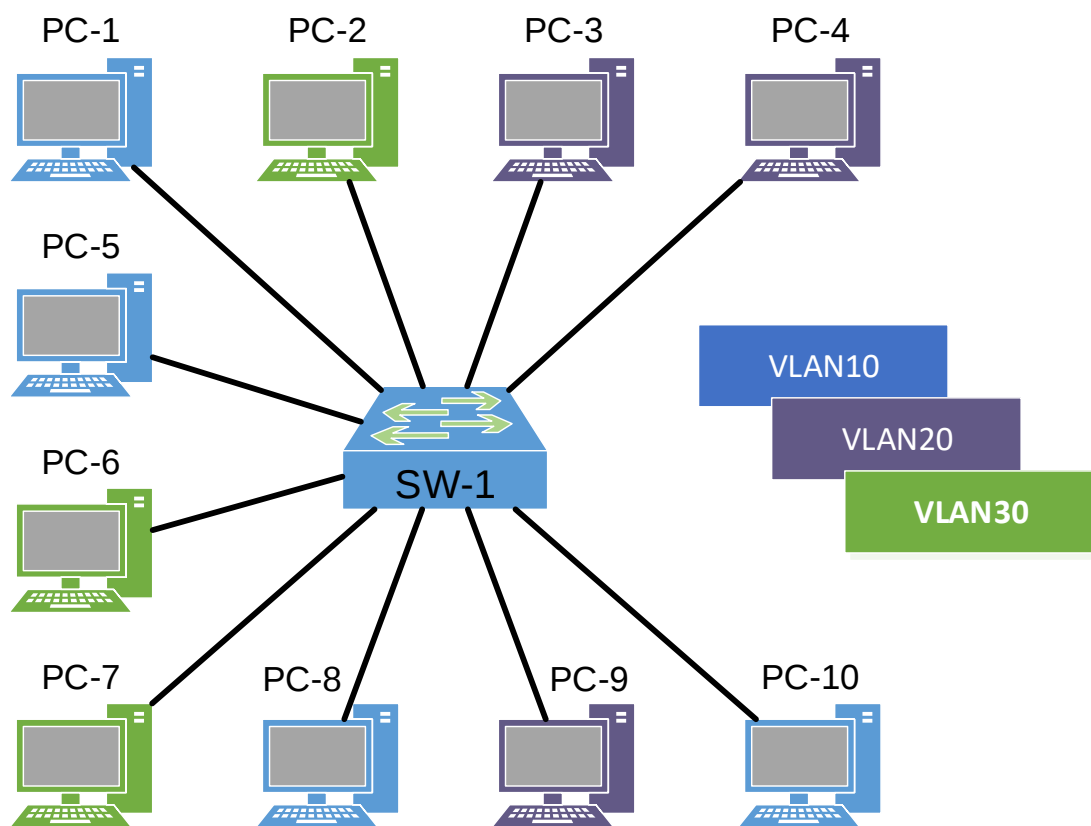


Рисунок 5.41 – Приклад мережі

Під час побудови каналу зв'язку для з'єднання пристроїв використано дані таблиці 5.35.

Таблиця 5.35 – Параметри інтерфейсів пристроїв для прикладу

| Інтерфейс комутатора SW-1 | Підключення до пристрою | vlan-id | Назва VLAN     |
|---------------------------|-------------------------|---------|----------------|
| Fa0/1                     | PC-1                    | 10      | Administrators |
| Fa0/2                     | PC-2                    | 30      | Instructors    |
| Fa0/3                     | PC-3                    | 20      | Students       |
| Fa0/4                     | PC-4                    | 20      | Students       |
| Fa0/5                     | PC-5                    | 10      | Administrators |
| Fa0/6                     | PC-6                    | 30      | Instructors    |
| Fa0/7                     | PC-7                    | 30      | Instructors    |
| Fa0/8                     | PC-8                    | 10      | Administrators |
| Fa0/9                     | PC-9                    | 20      | Students       |
| Fa0/10                    | PC-10                   | 10      | Administrators |

Для налагодження параметрів адресації інтерфейсів пристроїв використано дані таблиці 5.36.

Таблиця 5.36 – Параметри адресації мережі

| Підмережа/<br>Пристрій | Інтерфейс/Мережний<br>адаптер/Шлюз | IP-адреса/<br>Префікс | Маска<br>підмережі | Шлюз за замов-<br>чуванням |
|------------------------|------------------------------------|-----------------------|--------------------|----------------------------|
| <b>Підмережа А</b>     | —                                  | 195.1.1.0/24          | 255.255.255.0      |                            |
| PC-1                   | FastEthernet0                      | 195.1.1.10/24         | 255.255.255.0      | 195.1.1.1                  |
| PC-5                   | FastEthernet0                      | 195.1.1.11/24         | 255.255.255.0      | 195.1.1.1                  |
| PC-8                   | FastEthernet0                      | 195.1.1.12/24         | 255.255.255.0      | 195.1.1.1                  |
| PC-10                  | FastEthernet0                      | 195.1.1.13/24         | 255.255.255.0      | 195.1.1.1                  |
| <b>Підмережа В</b>     | —                                  | 195.1.2.0/24          | 255.255.255.0      |                            |
| PC-3                   | FastEthernet0                      | 195.1.2.10/24         | 255.255.255.0      | 195.1.2.1                  |
| PC-4                   | FastEthernet0                      | 195.1.2.11/24         | 255.255.255.0      | 195.1.2.1                  |
| PC-9                   | FastEthernet0                      | 195.1.2.12/24         | 255.255.255.0      | 195.1.2.1                  |
| <b>Підмережа С</b>     | —                                  | 195.1.3.0/24          | 255.255.255.0      |                            |
| PC-2                   | FastEthernet0                      | 195.1.3.10/24         | 255.255.255.0      | 195.1.3.1                  |
| PC-6                   | FastEthernet0                      | 195.1.3.11/24         | 255.255.255.0      | 195.1.3.1                  |
| PC-7                   | FastEthernet0                      | 195.1.3.12/24         | 255.255.255.0      | 195.1.3.1                  |

Сценарії налагодження параметрів *VLAN* на комутаторі **SW-1** наведені нижче.

Створення *VLAN*:

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname SW-1
SW-1(config)#vlan 10
SW-1(config-vlan)#name Administrators
SW-1(config-vlan)#vlan 20
SW-1(config-vlan)#name Students
SW-1(config-vlan)#vlan 30
SW-1(config-vlan)#name Instructors
```

Налаштувати належність портів комутатора до певної *VLAN*:

```
SW-1(config)#interface range f0/1, f0/5, f0/8, f0/10
SW-1(config-if-range)#switchport mode access
SW-1(config-if-range)#switchport access vlan 10
SW-1(config-if-range)#
SW-1(config-if-range)#exit
SW-1(config)#interface range f0/3, f0/4, f0/9
SW-1(config-if-range)#switchport mode access
SW-1(config-if-range)#switchport access vlan 20
SW-1(config-if-range)#exit
SW-1(config)#interface range f0/2, f0/6, f0/7
SW-1(config-if-range)#switchport mode access
SW-1(config-if-range)#switchport access vlan 30
```

## Результати виконання команд моніторингу та діагностики для розглянутого прикладу

З метою перегляду інформації про функціонування інтерфейсів комутатора для розглянутого прикладу використано команди **show ip interface brief**, **show vlan brief**. Перевірка зв'язку між комп'ютерами в межах однієї **VLAN** здійснена за допомогою команди **ping**. Результати роботи цих команд наведено відповідно далі:

Команда **show ip interface brief** для перевірки коректності встановлених інтерфейсів:

```
SW-1#show ip interface brief
```

| Interface          | IP-Address | OK? | Method | Status                | Protocol |
|--------------------|------------|-----|--------|-----------------------|----------|
| FastEthernet0/1    | unassigned | YES | manual | down                  | down     |
| -----              |            |     |        |                       |          |
| ...                |            |     |        |                       |          |
| -----              |            |     |        |                       |          |
| FastEthernet0/24   | unassigned | YES | manual | down                  | down     |
| GigabitEthernet0/1 | unassigned | YES | manual | up                    | up       |
| GigabitEthernet0/2 | unassigned | YES | manual | down                  | down     |
| Vlan1              | unassigned | YES | manual | administratively down | down     |
| Vlan10             | unassigned | YES | manual | up                    | down     |
| Vlan20             | unassigned | YES | manual | up                    | down     |
| Vlan30             | unassigned | YES | manual | up                    | down     |

Перевірка списку **VLAN** на комутаторі **SW-1**:

```
SW-1#show vlan brief
```

| VLAN  | Name               | Status | Ports                                                                                                                                |
|-------|--------------------|--------|--------------------------------------------------------------------------------------------------------------------------------------|
| ----- |                    |        |                                                                                                                                      |
| 1     | default            | active | Fa0/11, Fa0/12, Fa0/13, Fa0/14<br>Fa0/15, Fa0/16, Fa0/17, Fa0/18<br>Fa0/19, Fa0/20, Fa0/21, Fa0/22<br>Fa0/23, Fa0/24, Gig0/1, Gig0/2 |
| 10    | Administrators     | active | Fa0/1, Fa0/5, Fa0/8, Fa0/10                                                                                                          |
| 20    | Students           | active | Fa0/3, Fa0/4, Fa0/9                                                                                                                  |
| 30    | Instructors        | active | Fa0/2, Fa0/6, Fa0/7                                                                                                                  |
| 1002  | fddi-default       | active |                                                                                                                                      |
| 1003  | token-ring-default | active |                                                                                                                                      |
| 1004  | fddinet-default    | active |                                                                                                                                      |
| 1005  | trnet-default      | active |                                                                                                                                      |

Перевірка зв'язку між комп'ютером **PC-1** та **PC-5**, що належать до однієї віртуальної мережі **VLAN10**:

```
C:\>ping 195.1.1.11
Pinging 195.1.1.11 with 32 bytes of data:

Reply from 195.1.1.11: bytes=32 time<1ms TTL=128
Reply from 195.1.1.11: bytes=32 time=3ms TTL=128
Reply from 195.1.1.11: bytes=32 time<1ms TTL=128
Reply from 195.1.1.11: bytes=32 time<1ms TTL=128

Ping statistics for 195.1.1.11:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
 Minimum = 0ms, Maximum = 3ms, Average = 0ms
```

Перевірка зв'язку між комп'ютером PC-1 та PC-2, що належать до різних віртуальних мереж:

```
C:\>ping 195.1.3.10
```

```
Pinging 195.1.3.10 with 32 bytes of data:
```

```
Request timed out.
```

```
Request timed out.
```

```
Request timed out.
```

```
Request timed out.
```

```
Ping statistics for 195.1.3.10:
```

```
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

### Модельний приклад налагодження магістрального каналу зв'язку, побудованого між комутаторами Cisco

Розглянемо специфіку налагодження гігабітних інтерфейсів комутаторів Cisco у ході організації магістрального каналу зв'язку для забезпечення взаємодії VLAN, що зображений на рисунку 5.42.

Під час побудови даного каналу для з'єднання пристроїв використано дані таблиці 5.37.

Таблиця 5.37 – Параметри інтерфейсів пристроїв для прикладу

| Інтерфейс             | Підключення до пристрою | vlan-id | Назва VLAN     |
|-----------------------|-------------------------|---------|----------------|
| <b>Комутатор SW-1</b> |                         |         |                |
| Gig0/1                | SW-2                    | trunk   |                |
| Fa0/1                 | PC-1                    | 10      | Administrators |
| Fa0/2                 | PC-2                    | 30      | Instructors    |
| Fa0/3                 | PC-3                    | 20      | Students       |
| Fa0/4                 | PC-4                    | 20      | Students       |
| Fa0/5                 | PC-5                    | 10      | Administrators |
| Fa0/6                 | PC-6                    | 30      | Instructors    |
| Fa0/7                 | PC-7                    | 30      | Instructors    |
| Fa0/8                 | PC-8                    | 10      | Administrators |
| Fa0/9                 | PC-9                    | 20      | Students       |
| Fa0/10                | PC-10                   | 10      | Administrators |
| <b>Комутатор SW-2</b> |                         |         |                |
| Gig0/1                | SW-1                    | trunk   |                |
| Fa0/1                 | PC-11                   | 10      | Administrators |
| Fa0/2                 | PC-12                   | 30      | Instructors    |
| Fa0/3                 | PC-13                   | 20      | Students       |
| Fa0/4                 | PC-14                   | 30      | Students       |
| Fa0/5                 | PC-15                   | 10      | Administrators |
| Fa0/6                 | PC-16                   | 30      | Instructors    |
| Fa0/7                 | PC-17                   | 10      | Instructors    |
| Fa0/8                 | PC-18                   | 10      | Administrators |
| Fa0/9                 | PC-19                   | 30      | Students       |
| Fa0/10                | PC-20                   | 10      | Administrators |

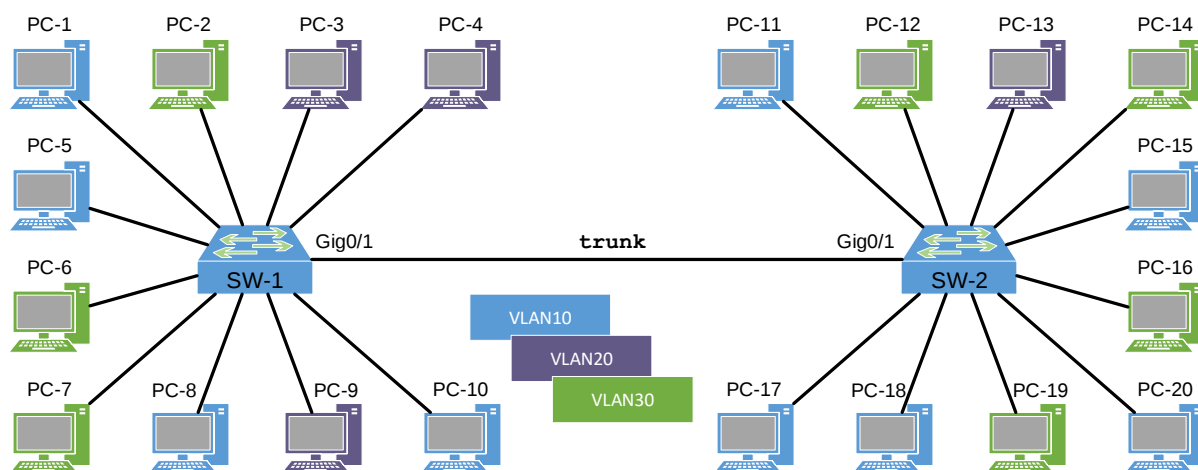


Рисунок 5.42 – Приклад мережі

Для налагодження параметрів адресації інтерфейсів пристроїв використано дані таблиці 5.37 та 5.38.

Таблиця 5.38 – Додаткові параметри адресації мережі

| Підмережа/<br>Пристрій | Інтерфейс/Мережний<br>адаптер/Шлюз | IP-адреса/<br>Префікс | Маска підмережі | Шлюз за замов-<br>чуванням |
|------------------------|------------------------------------|-----------------------|-----------------|----------------------------|
| <b>Підмережа А</b>     | —                                  | 195.1.1.0/24          | 255.255.255.0   |                            |
| <b>PC-11</b>           | FastEthernet0                      | 195.1.1.20/24         | 255.255.255.0   | 195.1.1.1                  |
| <b>PC-15</b>           | FastEthernet0                      | 195.1.1.21/24         | 255.255.255.0   | 195.1.1.1                  |
| <b>PC-17</b>           | FastEthernet0                      | 195.1.1.22/24         | 255.255.255.0   | 195.1.1.1                  |
| <b>PC-18</b>           | FastEthernet0                      | 195.1.1.23/24         | 255.255.255.0   | 195.1.1.1                  |
| <b>PC-20</b>           | FastEthernet0                      | 195.1.1.24/24         | 255.255.255.0   | 195.1.1.1                  |
| <b>Підмережа В</b>     | —                                  | 195.1.2.0/24          | 255.255.255.0   |                            |
| <b>PC-13</b>           | FastEthernet0                      | 195.1.2.20/24         | 255.255.255.0   | 195.1.2.1                  |
| <b>Підмережа С</b>     | —                                  | 195.1.3.0/24          | 255.255.255.0   |                            |
| <b>PC-12</b>           | FastEthernet0                      | 195.1.3.20/24         | 255.255.255.0   | 195.1.3.1                  |
| <b>PC-14</b>           | FastEthernet0                      | 195.1.3.21/24         | 255.255.255.0   | 195.1.3.1                  |
| <b>PC-16</b>           | FastEthernet0                      | 195.1.3.22/24         | 255.255.255.0   | 195.1.3.1                  |
| <b>PC-19</b>           | FastEthernet0                      | 195.1.3.23/24         | 255.255.255.0   | 195.1.3.1                  |

Сценарії налагодження параметрів *VLAN* на комутаторі **SW-2** аналогічні налаштуванням комутатора **SW-1** і наведені вище.

Ручне налаштування магістральних інтерфейсів **Gig0/1**, якими з'єднані комутатори наведено далі:

```

SW-1>enable
SW-1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW-1(config)#interface Gig0/1
SW-1(config-if)#switchport mode trunk

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up

SW-1(config-if)#switchport trunk allowed vlan all
SW-1(config-if)#exit
SW-2>enable
SW-2#configure terminal
SW-2(config)#interface Gig0/1
SW-2(config-if)#switchport mode trunk
SW-2(config-if)#switchport trunk allowed vlan all
SW-2(config-if)#exit

```

## Результати виконання команд моніторингу та діагностики для розглянутого прикладу

З метою перегляду інформації про функціонування транкових інтерфейсів для розглянутого прикладу використано команду **show interfaces trunk**. Перевірка зв'язку між комп'ютерами в межах однієї **VLAN** і таких що підключені до різних комутаторів здійснена за допомогою команди **ping**. Результати роботи цих команд наведено відповідно далі:

```

SW-1>show interfaces trunk

```

| Port   | Mode | Encapsulation | Status   | Native vlan |
|--------|------|---------------|----------|-------------|
| Gig0/1 | on   | 802.1q        | trunking | 1           |

```

Port Vlans allowed on trunk
Gig0/1 1-1005

Port Vlans allowed and active in management domain
Gig0/1 1,10,20,30

Port Vlans in spanning tree forwarding state and not pruned
Gig0/1 1,10,20,30

```

Перевірка зв'язку між комп'ютером **PC-1** та **PC-20** що належать до однієї віртуальної мережі **VLAN10** та під'єднані до різних комутаторів:

```

C:\>ping 195.1.1.24

Pinging 195.1.1.24 with 32 bytes of data:

Reply from 195.1.1.24: bytes=32 time<1ms TTL=128
Reply from 195.1.1.24: bytes=32 time=3ms TTL=128
Reply from 195.1.1.24: bytes=32 time<1ms TTL=128
Reply from 195.1.1.24: bytes=32 time<1ms TTL=128

Ping statistics for 195.1.1.24:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
 Minimum = 0ms, Maximum = 3ms, Average = 0ms

```

Перевірка зв'язку між комп'ютером PC-1 та PC-16 що належать до різних віртуальних мереж:

```
C:\>ping 195.1.3.22
```

```
Pinging 195.1.3.22 with 32 bytes of data:
```

```
Request timed out.
```

```
Request timed out.
```

```
Request timed out.
```

```
Request timed out.
```

```
Ping statistics for 195.1.3.22:
```

```
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

## ПРАКТИЧНЕ ЗАВДАННЯ

**Мета роботи:** ознайомитися з технологіями віртуальних мереж та магістральних каналів та особливостями їх налаштування на комутаторах Cisco.

### Порядок виконання роботи

1. Створити фізичний проєкт мережі або у середовищі програмного симулятора/емулятора створити мережу (рисунок 5.43). При побудові звернути увагу на вибір моделей комутаторів, мережних модулів та плат, а також мережних з'єднань. При побудові підмережі слід вибирати потрібний тип кабелю для відповідної технології.

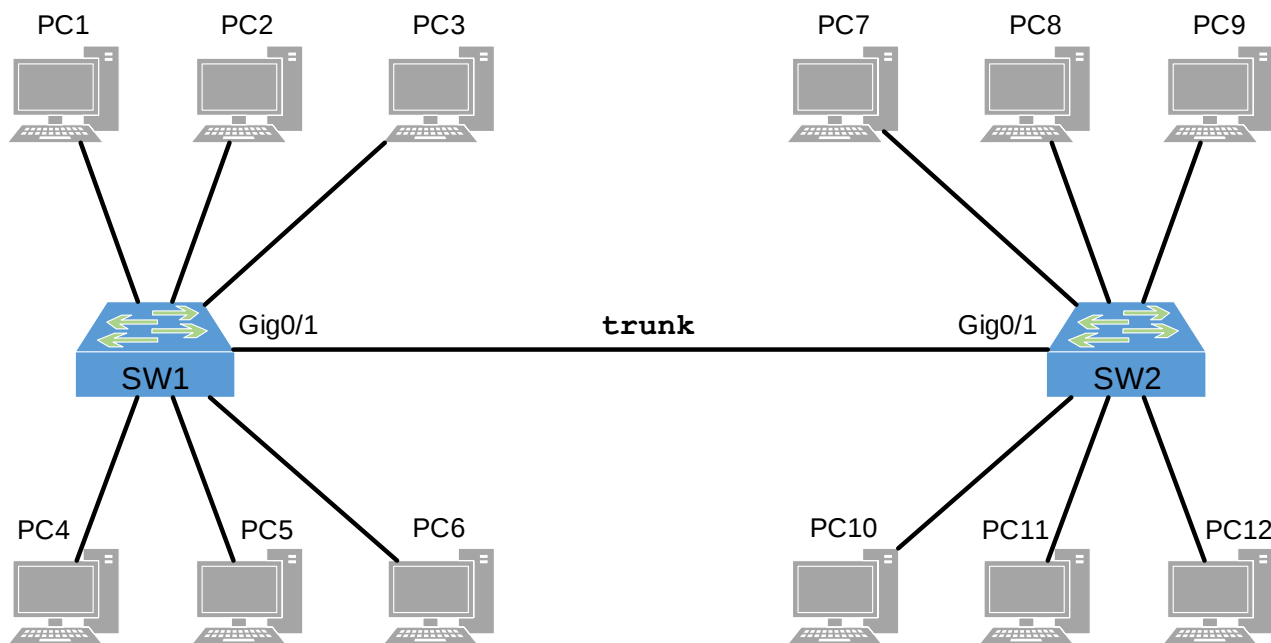


Рисунок 5.43 – Проєкт мережі

2. Провести базове налаштування комутаторів, мережних інтерфейсів, з'єднань, додати VLAN та створити магістральне з'єднання. Належність комп'ютерів до певної VLAN наведено в таблиці 5.40.

3. Розробити схему адресації пристроїв мережі на основі даних, які наведені у таблицях 5.39, 5.40. Результати навести у вигляді таблиці.

Таблиця 5.39 – Параметри IP-адресації підмереж

| Віртуальна мережа | Адреса мережі | Префікс |
|-------------------|---------------|---------|
| <b>VLAN10</b>     | 192.168.10.0  | /24     |
| <b>VLAN20</b>     | 192.168.20.0  | /24     |
| <b>VLAN30</b>     | 192.168.30.0  | /24     |

Таблиця 5.40 – Належність ПК до VLAN

| №<br>варіанта | VLAN10               | VLAN20               | VLAN30               |
|---------------|----------------------|----------------------|----------------------|
| 1             | PC1, PC2, PC3, PC9   | PC4, PC5, PC8, PC10  | PC6, PC7, PC11, PC12 |
| 2             | PC2, PC3, PC6, PC10  | PC1, PC4, PC5, PC12  | PC7, PC8, PC9, PC11  |
| 3             | PC3, PC5, PC11, PC12 | PC4, PC6, PC9, PC10  | PC1, PC2, PC7, PC8   |
| 4             | PC1, PC2, PC3, PC9   | PC6, PC7, PC11, PC12 | PC4, PC5, PC8, PC10  |
| 5             | PC2, PC3, PC6, PC10  | PC7, PC8, PC9, PC11  | PC1, PC4, PC5, PC12  |
| 6             | PC3, PC5, PC11, PC12 | PC1, PC2, PC7, PC8   | PC4, PC6, PC9, PC10  |
| 7             | PC6, PC7, PC11, PC12 | PC1, PC2, PC3, PC9   | PC4, PC5, PC8, PC10  |
| 8             | PC7, PC8, PC9, PC11  | PC2, PC3, PC6, PC10  | PC1, PC4, PC5, PC12  |
| 9             | PC1, PC2, PC7, PC8   | PC3, PC5, PC11, PC12 | PC4, PC6, PC9, PC10  |
| 10            | PC4, PC5, PC8, PC10  | PC1, PC2, PC3, PC9   | PC6, PC7, PC11, PC12 |
| 11            | PC1, PC4, PC5, PC12  | PC2, PC3, PC6, PC10  | PC7, PC8, PC9, PC11  |
| 12            | PC4, PC6, PC9, PC10  | PC3, PC5, PC11, PC12 | PC1, PC2, PC7, PC8   |
| 13            | PC4, PC5, PC8, PC10  | PC6, PC7, PC11, PC12 | PC1, PC2, PC3, PC9   |
| 14            | PC1, PC4, PC5, PC12  | PC7, PC8, PC9, PC11  | PC2, PC3, PC6, PC10  |
| 15            | PC4, PC6, PC9, PC10  | PC1, PC2, PC7, PC8   | PC3, PC5, PC11, PC12 |
| 16            | PC6, PC7, PC11, PC12 | PC4, PC5, PC8, PC10  | PC1, PC2, PC3, PC9   |
| 17            | PC7, PC8, PC9, PC11  | PC1, PC4, PC5, PC12  | PC2, PC3, PC6, PC10  |
| 18            | PC1, PC2, PC7, PC8   | PC4, PC6, PC9, PC10  | PC3, PC5, PC11, PC12 |
| 19            | PC1, PC2, PC3, PC9   | PC6, PC7, PC11, PC12 | PC4, PC5, PC8, PC10  |
| 20            | PC2, PC3, PC6, PC10  | PC7, PC8, PC9, PC11  | PC1, PC4, PC5, PC12  |
| 21            | PC3, PC5, PC11, PC12 | PC1, PC2, PC7, PC8   | PC4, PC6, PC9, PC10  |
| 22            | PC6, PC7, PC11, PC12 | PC1, PC2, PC3, PC9   | PC4, PC5, PC8, PC10  |
| 23            | PC7, PC8, PC9, PC11  | PC2, PC3, PC6, PC10  | PC1, PC4, PC5, PC12  |
| 24            | PC1, PC2, PC7, PC8   | PC3, PC5, PC11, PC12 | PC4, PC6, PC9, PC10  |
| 25            | PC4, PC5, PC8, PC10  | PC1, PC2, PC3, PC9   | PC6, PC7, PC11, PC12 |
| 26            | PC1, PC4, PC5, PC12  | PC2, PC3, PC6, PC10  | PC7, PC8, PC9, PC11  |
| 27            | PC4, PC6, PC9, PC10  | PC3, PC5, PC11, PC12 | PC1, PC2, PC7, PC8   |
| 28            | PC4, PC5, PC8, PC10  | PC6, PC7, PC11, PC12 | PC1, PC2, PC3, PC9   |
| 29            | PC1, PC4, PC5, PC12  | PC7, PC8, PC9, PC11  | PC2, PC3, PC6, PC10  |
| 30            | PC4, PC6, PC9, PC10  | PC1, PC2, PC7, PC8   | PC3, PC5, PC11, PC12 |

4. Провести налаштування параметрів *IP*-адресації пристроїв мережі у відповідності до даних п. 3. Перевірити наявність зв'язку між парами пристроїв мережі, що належать одній *VLAN*. Та перевірити відсутність зв'язку між парами, що належать різним *VLAN*.

### КОНТРОЛЬНІ ПИТАННЯ

1. Дайте визначення *VLAN* і яку роль вони відіграють в мережах?
2. Які переваги використання *VLAN* в мережі?
3. Як визначити *VLAN* на комутаторі *Cisco*?
4. Як налаштувати магістральний канал (транк) на комутаторі *Cisco*?
5. Які протоколи використовуються для передачі *VLAN*-інформації через транк?
6. Як визначити, чи працює транк між двома комутаторами *Cisco*?
7. Що таке *ISL* і *802.1Q* і в чому їхня роль в роботі з транками?
8. Як видалити *VLAN* на комутаторі *Cisco*?
9. Як встановити Native *VLAN* на транку?
10. Як налаштувати *IP*-адреси на інтерфейсах *VLAN* на маршрутизаторі *Cisco*?
11. Які команди дозволяють перевірити конфігурації *VLAN*?
12. Як визначити та виправити проблеми з транками на комутаторі *Cisco*?
13. Для чого можна використати команду **show interfaces trunk**?
14. Які є відмінності між статичним та динамічним маркуванням *VLAN* на комутаторі?
15. Чи можна зробити резервний транк для забезпечення надійності мережі?

## 5.2.2 НАЛАШТУВАННЯ МАРШРУТИЗАЦІЇ МІЖ VLAN

### ОСНОВНІ КОМАНДИ НАЛАШТУВАННЯ VLAN

Передача трафіку між VLAN може здійснюватися за допомогою маршрутизатора. Для того щоб маршрутизатор міг передавати трафік з однієї VLAN в іншу (з однієї мережі в іншу), необхідно, щоб у кожній мережі в нього був інтерфейс. Для того щоб не виділяти для кожної VLAN окремий фізичний інтерфейс, створюють логічні підінтерфейси на фізичному інтерфейсі для кожної VLAN.

На комутаторі порт, що веде до маршрутизатора, має бути налаштований як магістральний (*trunk*). Така схема, в якій маршрутизація між VLAN виконується на маршрутизаторі, називається *router-on-a-stick*.

Сучасні корпоративні мережі не використовують *router-on-a-stick*, оскільки його важко масштабувати відповідно до вимог. У цих дуже великих мережах мережеві адміністратори використовують комутатори 3-го рівня для налаштування міжмережної маршрутизації.

Можливості комутатора рівня 3 включають в себе наступні функції:

- Маршрутизація з однієї VLAN в іншу за допомогою декількох комутованих віртуальних інтерфейсів (SVI).
- Перетворення порту комутатора рівня 2 в інтерфейс рівня 3 (тобто в маршрутизований порт).

SVI налаштовуються за допомогою тієї ж команди **interface vlan vlan-id**, яка використовується для створення керуючого SVI на комутаторі 2-го рівня. Для кожної з маршрутизованих VLAN необхідно створити SVI рівня 3.

Основними командами налаштування маршрутизації між VLAN методом *router-on-a-stick* є команди створення логічного підінтерфейсу. У випадку з комутатором 3 рівня базовими командами є: **ip routing** та **ip route** для перенаправлення всього невідомого трафіку.

Щоб визначити формат інкапсуляції VLAN як *IEEE 802.1Q*, використовуйте наступні команди в режимі конфігурації інтерфейсу:

- Вказує підінтерфейс, на якому буде використовуватися *IEEE 802.1Q*, і переходить у режим конфігурації інтерфейсу:

```
interface interface-number slot/port.subinterface-number
```

- Визначає формат інкапсуляції як *IEEE 802.1Q (dot1q)* і вказує ідентифікатор VLAN:

```
encapsulation dot1q vlanid
```

де **vlanid** – визначає ідентифікатор VLAN.

IP-маршрутизація автоматично увімкнена в програмному забезпеченні Cisco IOS для маршрутизаторів. Щоб увімкнути IP-маршрутизацію для комутатора 3 рівня або, якщо її було вимкнено на маршрутизаторі, скористайтеся наступною командою у режимі глобальних налаштувань: **ip routing**.

Для прямих пакетів, адресованих у мережі, які не описані явним чином у таблиці маршрутизації може виникнути потреба вказати стандартний маршрут. Наприклад, окремий випадок, переадресація пакетів із локальної мережі в *Internet*. Створення статичного маршруту до мережі *0.0.0.0 0.0.0.0* є одним із способів визначення шлюзу останньої черги:

```
ip route 0.0.0.0 0.0.0.0 ipaddress
```

де **ipaddress** – IP-адреса шлюзу останньої черги.

### Модельний приклад налагодження маршрутизації між VLAN методом *router-on-a-stick*

Розглянемо специфіку налагодження маршрутизації між VLAN з допомогою метода *router-on-a-stick* на маршрутизаторі Cisco, схема якої наведена на рисунку 5.44.

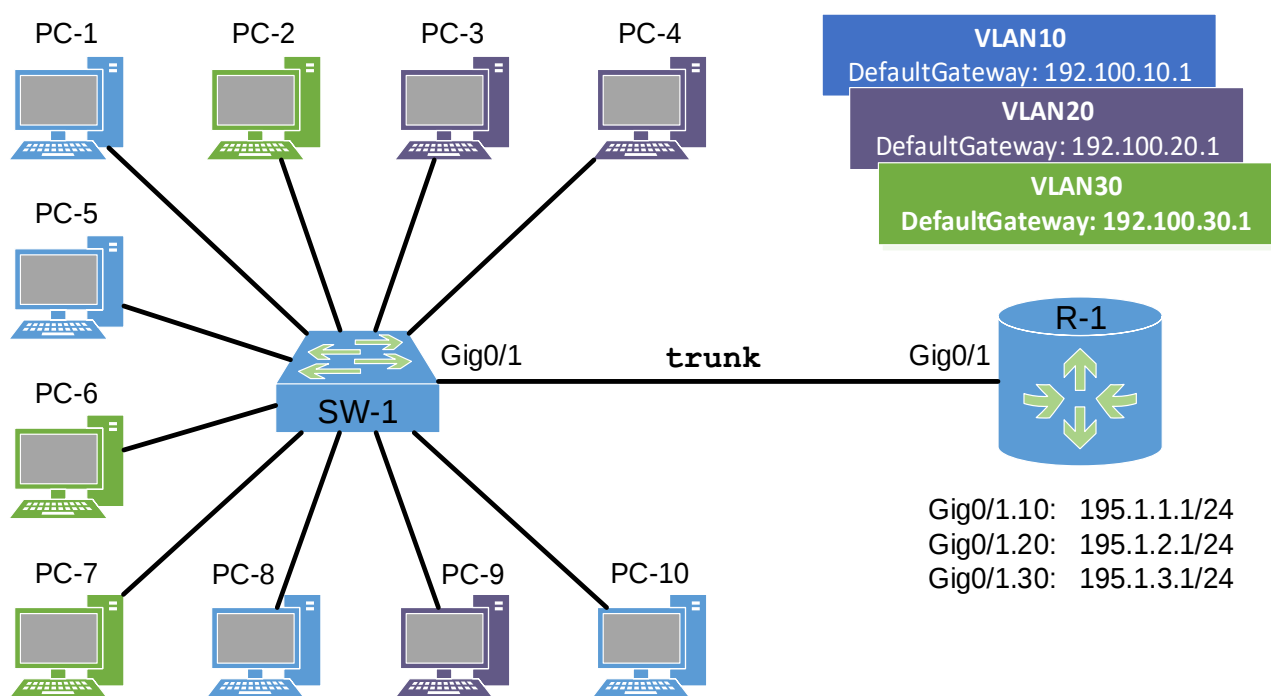


Рисунок 5.44 – Приклад мережі

Під час побудови каналів зв'язку для з'єднання пристроїв використано дані таблиці 5.35 (див. п.5.2.1). Для налагодження параметрів адресації інтерфейсів пристроїв використано дані таблиці 5.36 (див. п.5.2.1).

Для налагодження підінтерфейсів маршрутизатора використана таблиця 5.41.

Таблиця 5.41 – Параметри адресації підінтерфейсів маршрутизатора

| Пристрій | Інтерфейс | IP-адреса/Префікс | Маска підмережі |
|----------|-----------|-------------------|-----------------|
| R1       | G0/0.10   | 195.1.1.1/24      | 255.255.255.0   |
| R1       | G0/0.20   | 195.1.2.1/24      | 255.255.255.0   |
| R1       | G0/0.30   | 195.1.3.1/24      | 255.255.255.0   |

Сценарії налагодження параметрів *VLAN* на комутаторі **SW-1** аналогічні прикладам наведеним в пункті посібника 5.2.1.

Додаткові сценарії налагодження параметрів маршрутизації *VLAN* на маршрутизаторі **R-1** наведені далі:

Створення логічних підінтерфейсів із зазначенням номера *VLAN*, що відповідає цьому інтерфейсу, і зазначенням, що інтерфейс отримуватиме тегований трафік:

```
R-1>enable
R-1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R-1(config)#interface Gig0/1.10
R-1(config-subif)#encapsulation dot1q 10
R-1(config-subif)#ip address 195.1.1.1 255.255.255.0
R-1(config-subif)#exit
R-1(config)#interface Gig0/1.20
R-1(config-subif)#encapsulation dot1q 20
R-1(config-subif)#ip address 195.1.2.1 255.255.255.0
R-1(config-subif)#exit
R-1(config)#interface Gig0/1.30
R-1(config-subif)#encapsulation dot1q 30
R-1(config-subif)#ip address 195.1.3.1 255.255.255.0
R-1(config-subif)#exit
```

Щоб включити підінтерфейси, необхідно обов'язково активувати фізичний інтерфейс, з яким вони пов'язані:

```
R-1(config)#interface Gig0/1
R-1(config-if)#no shutdown
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up

%LINK-5-CHANGED: Interface GigabitEthernet0/1.10, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1.10, changed state to up

%LINK-5-CHANGED: Interface GigabitEthernet0/1.20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1.20, changed state to up

%LINK-5-CHANGED: Interface GigabitEthernet0/1.30, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1.30, changed state to up
```

### Результати виконання команд моніторингу та діагностики для розглянутого прикладу

З метою перегляду інформації про функціонування інтерфейсів маршрутизатора для розглянутого прикладу використано команди **show ip interface brief**, **show interface**. Перевірка маршрутизації між комп'ютерами в різних *VLAN* здійснена за допомогою команди **ping**. Результати роботи цих команд наведено відповідно далі:

Команда **show ip interface brief** для перевірки коректності встановлених інтерфейсів:

```
R-1#show ip interface brief
```

| Interface             | IP-Address | OK? | Method | Status                | Protocol |
|-----------------------|------------|-----|--------|-----------------------|----------|
| GigabitEthernet0/0    | unassigned | YES | unset  | administratively down | down     |
| GigabitEthernet0/1    | unassigned | YES | unset  | up                    | up       |
| GigabitEthernet0/1.10 | 195.1.1.1  | YES | manual | up                    | up       |
| GigabitEthernet0/1.20 | 195.1.2.1  | YES | manual | up                    | up       |
| GigabitEthernet0/1.30 | 195.1.3.1  | YES | manual | up                    | up       |
| Vlan1                 | unassigned | YES | unset  | administratively down | down     |

Перевірка стану підінтерфейсу маршрутизатора **Gig0/1.20**:

```
R-1#show interfaces Gig0/1.20
GigabitEthernet0/1.20 is up, line protocol is up (connected)
 Hardware is PQUICC_FEC, address is 00d0.ba52.dd02 (bia 00d0.ba52.dd02)
 Internet address is 195.1.2.1/24
 MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
 reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation 802.1Q Virtual LAN, Vlan ID 20
 ARP type: ARPA, ARP Timeout 04:00:00,
 Last clearing of "show interface" counters never
```

Перевірка зв'язку між комп'ютером **PC-2** та інтерфейсом маршрутизатора (шлюзом за замовчуванням):

```
C:\>ping 195.1.3.1
```

Pinging 195.1.3.1 with 32 bytes of data:

```
Reply from 195.1.3.1: bytes=32 time<1ms TTL=255
Reply from 195.1.3.1: bytes=32 time<1ms TTL=255
Reply from 195.1.3.1: bytes=32 time<1ms TTL=255
Reply from 195.1.3.1: bytes=32 time<1ms TTL=255
```

Ping statistics for 195.1.3.1:

```
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
 Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Перевірка зв'язку між комп'ютером **PC-2** та **PC-1** що належать до різних віртуальних мереж:

```
C:\>ipconfig
```

FastEthernet0 Connection: (default port)

```
 Connection-specific DNS Suffix...:
 Link-local IPv6 Address: FE80::203:E4FF:FE67:E301
 IPv6 Address: ::
 IPv4 Address: 195.1.3.10
 Subnet Mask: 255.255.255.0
 Default Gateway: ::
 195.1.3.1
```

```
C:\>ping 195.1.1.10
```

Pinging 195.1.1.10 with 32 bytes of data:

```
Reply from 195.1.1.10: bytes=32 time=10ms TTL=127
Reply from 195.1.1.10: bytes=32 time=11ms TTL=127
```

```

Reply from 195.1.1.10: bytes=32 time=11ms TTL=127
Reply from 195.1.1.10: bytes=32 time=19ms TTL=127

Ping statistics for 195.1.1.10:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
 Minimum = 10ms, Maximum = 19ms, Average = 12ms

```

### Модельний приклад налагодження магістрального каналу зв'язку, побудованого між комутаторами *Cisco*

Схему мережі з особливостями налагодження маршрутизації між *VLAN* за допомогою комутатора 3 рівня та використанням шлюзу останньої черги для доступу до *Internet*, наведено на рисунку 5.45.

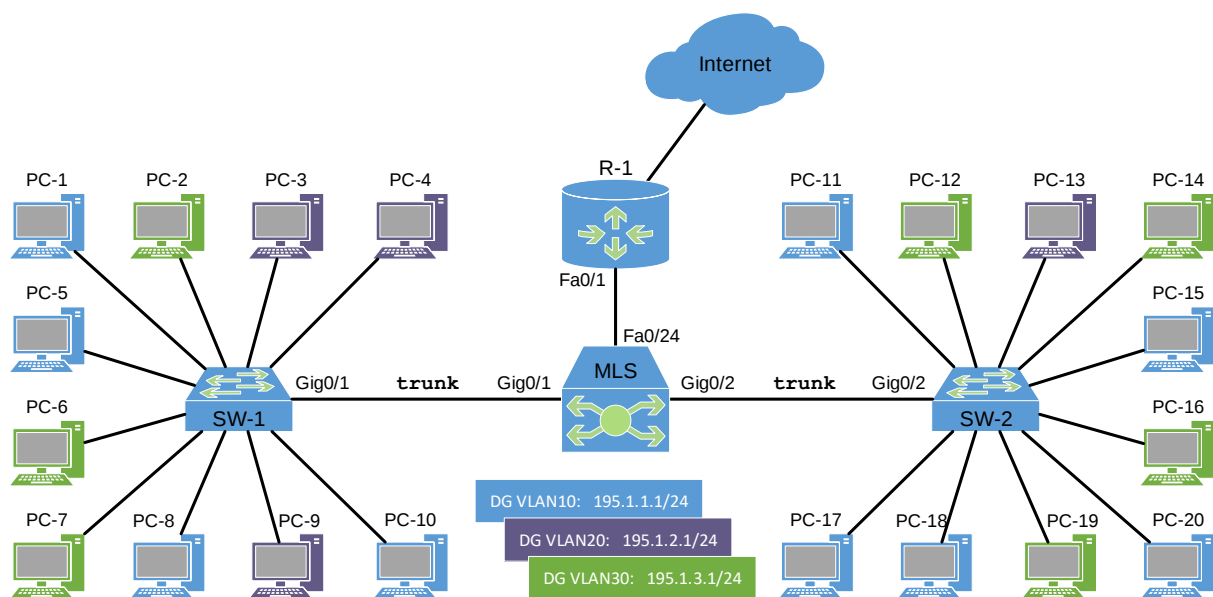


Рисунок 5.45 – Приклад мережі

Під час побудови каналів зв'язку для з'єднання пристроїв використано дані таблиці 5.37 (див. п.5.2.1). Для налагодження параметрів адресації інтерфейсів пристроїв використано дані таблиці 5.38 (див. п.5.2.1).

Для налагодження інтерфейсів *SVI* для *VLAN* комутатора 3 рівня використана таблиця 5.42.

Таблиця 5.42 – Параметри адресації інтерфейсів комутатора 3 рівня

| Пристрій | Інтерфейс | IP-адреса/Префікс | Маска підмережі |
|----------|-----------|-------------------|-----------------|
| MLS      | VLAN10    | 195.1.1.1/24      | 255.255.255.0   |
| MLS      | VLAN20    | 195.1.2.1/24      | 255.255.255.0   |
| MLS      | VLAN30    | 195.1.3.1/24      | 255.255.255.0   |
| MLS      | Fa0/24    | 172.100.100.2/30  | 255.255.255.252 |
| R-1      | Fa0/1     | 172.100.100.1/30  | 255.255.255.252 |

Сценарії налагодження параметрів *VLAN* на комутаторі **SW-1** та **SW-2** аналогічні прикладам наведеним в пункті 5.2.1.

Сценарії налагодження параметрів *VLAN* та магістральних інтерфейсів на комутаторі **SW-1** та **SW-2** аналогічні прикладам наведеним в пункті посібника 5.2.1.

Для налаштування маршрутизації між *VLAN* та маршруту за замовчуванням на комутаторі 3 рівня **MLS** потрібно виконати наступні сценарії:

Включити маршрутизацію:

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname MLS
MLS(config)# ip routing
```

Додати *VLAN* до комутатора 3 рівня:

```
MLS(config)#vlan 10
MLS(config-vlan)#name Administrators
MLS(config-vlan)#vlan 20
MLS(config-vlan)#name Students
MLS(config-vlan)#vlan 30
MLS(config-vlan)#name Instructors
```

Призначити *IP*-адреси до *VLAN* комутатора. Ці адреси будуть маршрутом за замовчуванням для комп'ютерів у відповідній *VLAN*:

```
MLS(config)#interface vlan10
MLS(config-if)#ip address 195.1.1.1 255.255.255.0
MLS(config-if)#exit
MLS(config)#interface vlan20
MLS(config-if)#ip address 195.1.2.1 255.255.255.0
MLS(config-if)#exit
MLS(config)#interface vlan30
MLS(config-if)#ip address 195.1.3.1 255.255.255.0
MLS(config-if)#exit
MLS(config)#
```

Інтерфейс **fa0/24** з'єднаний із маршрутизатором перевести в режим 3 рівня та призначити *IP*-адресу:

```
MLS(config)#interface Fa0/24
MLS(config-if)#no switchport
MLS(config-if)#ip address 172.100.100.2 255.255.255.252
MLS(config-if)#no shutdown
```

Налаштувати магістральні інтерфейси **Gig0/1** та **Gig0/2**, якими комутатор **MLS** з'єднаний з іншими комутаторами робочих груп:

```
MLS>enable
MLS#configure terminal
MLS(config)#interface Gig0/1
MLS(config-if)#switchport mode trunk
MLS(config-if)#switchport trunk allowed vlan all
MLS(config-if)#exit

MLS(config)#interface Gig0/2
MLS(config-if)#switchport mode trunk
MLS(config-if)#switchport trunk allowed vlan all
MLS(config-if)#exit
```

Налаштування маршруту за замовчуванням передбачає використання **R-1** як шлюзу за замовчуванням для розглянутої мережі. Трафік, не призначений мережам **VLAN**, буде передаватися на **R-1**:

```
MLS(config) ip route 0.0.0.0 0.0.0.0 172.100.100.1
```

### Результати виконання команд моніторингу та діагностики для розглянутого прикладу

З метою перегляду інформації про функціонування інтерфейсів комутаційних пристроїв для розглянутого прикладу можна використати команди **show interfaces trunk** та **show interface switchport**. Перевірка зв'язку між комп'ютерами в межах однієї **VLAN** і таких що підключені до різних комутаторів здійснюється за допомогою команди **ping**. Результати роботи цих команд наведено відповідно далі:

```
MLS#show interface Gig0/1 switchport
Name: Gig0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: All
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
Appliance trust: none
```

Перевірка зв'язку між комп'ютером **PC-1** та **PC-16** що належать до різних віртуальних мереж та під'єднані до різних комутаторів:

```
C:\>ping 195.1.3.22
```

```
Pinging 195.1.3.22 with 32 bytes of data:
```

```
Reply from 195.1.3.22: bytes=32 time<1ms TTL=128
Reply from 195.1.3.22: bytes=32 time=3ms TTL=128
Reply from 195.1.3.22: bytes=32 time<1ms TTL=128
Reply from 195.1.3.22: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 195.1.3.22:
```

```
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
 Minimum = 0ms, Maximum = 3ms, Average = 0ms
```

Перевірка зв'язку між комп'ютером **PC-1** та маршрутизатором:

```
C:\>ping 172.100.100.1
```

```
Pinging 172.100.100.1 with 32 bytes of data:
```

```
Reply from 172.100.100.1: bytes=32 time<1ms TTL=128
```

```
Reply from 172.100.100.1: bytes=32 time=3ms TTL=128
```

```
Reply from 172.100.100.1: bytes=32 time<1ms TTL=128
```

```
Reply from 172.100.100.1: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 172.100.100.1:
```

```
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
 Minimum = 0ms, Maximum = 3ms, Average = 0ms
```

## ПРАКТИЧНЕ ЗАВДАННЯ

**Мета роботи:** ознайомитися з маршрутизацією віртуальних мереж, поняттям шлюзу останньої черги та особливостями їх налаштування на комутаторах *Cisco*.

### Порядок виконання роботи

1. Створити фізичний проєкт мережі або у середовищі програмного симулятора/емулятора створити мережу (рисунок 5.46). При побудові звернути увагу на вибір моделей комутаторів, мережних модулів та плат, а також мережних з'єднань. При побудові підмережі слід вибирати потрібний тип кабелю для відповідної технології. В якості комутаторів робочих груп взяти комутатори 2 рівня, наприклад *Cisco 2950T-24* або аналогічні. В якості маршрутизуючого комутатора взяти комутатор 3 рівня, наприклад *Cisco 3560-24PS* або інші аналоги.

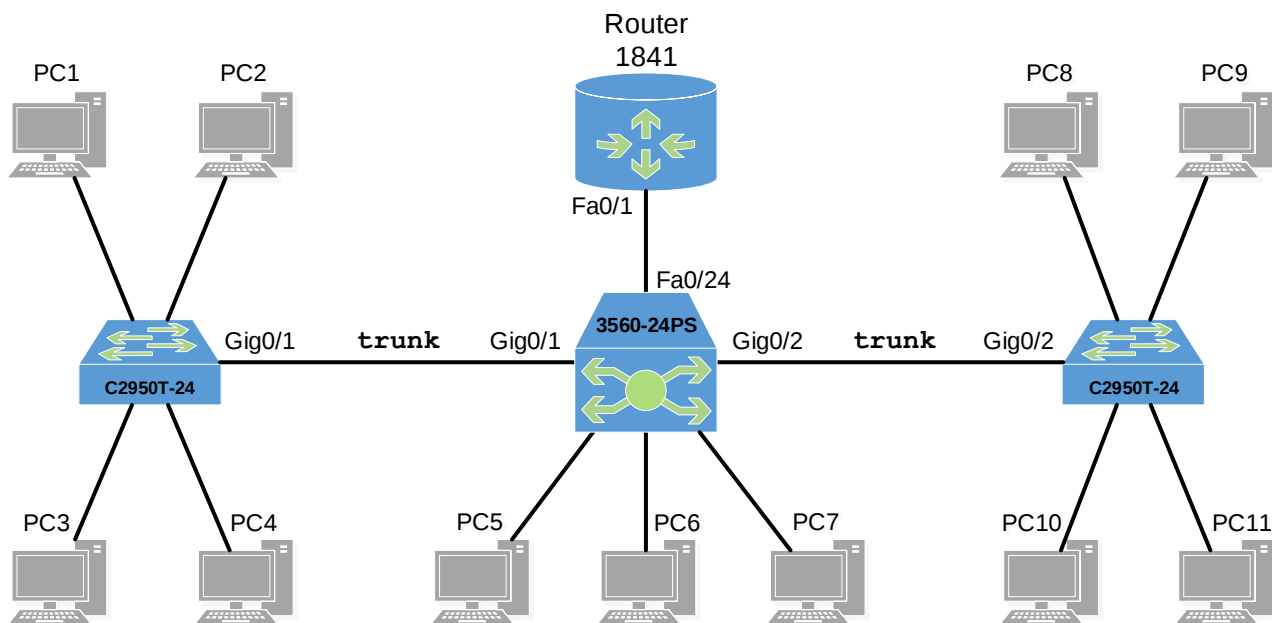


Рисунок 5.46 – Проєкт мережі

2. Провести базове налаштування маршрутизатора, комутаторів, мережних інтерфейсів, з'єднань, додати *VLAN*, створити магістральні з'єднання, налаштувати маршрутизацію та шлюз останньої черги.

3. Розробити схему адресації пристроїв мережі на основі даних, які

наведені у таблицях 5.43. Належність комп'ютерів до певної *VLAN* наведено в таблиці 5.44. Результати навести у вигляді таблиці.

Таблиця 5.43 – Параметри *IP*-адресації

| Віртуальна мережа або інтерфейс | Адреса мережі | Префікс |
|---------------------------------|---------------|---------|
| <b>VLAN10</b>                   | 192.168.10.0  | /24     |
| <b>VLAN20</b>                   | 192.168.20.0  | /24     |
| <b>VLAN30</b>                   | 192.168.30.0  | /24     |
| <b>Fa0/1 (1841)</b>             | 192.168.1.0   | /30     |
| <b>Fa0/24 (3560-24PS)</b>       | 192.168.1.0   | /30     |

Таблиця 5.44 – Належність *PC* до *VLAN*

| № варіанта | VLAN10              | VLAN20              | VLAN30              |
|------------|---------------------|---------------------|---------------------|
| 1          | PC6, PC7, PC11      | PC4, PC5, PC8, PC10 | PC1, PC2, PC3, PC9  |
| 2          | PC7, PC8, PC9, PC11 | PC1, PC4, PC5       | PC2, PC3, PC6, PC10 |
| 3          | PC1, PC2, PC7, PC8  | PC4, PC6, PC9, PC10 | PC3, PC5, PC11      |
| 4          | PC1, PC2, PC3, PC9  | PC6, PC7, PC11      | PC4, PC5, PC8, PC10 |
| 5          | PC2, PC3, PC6, PC10 | PC7, PC8, PC9, PC11 | PC1, PC4, PC5       |
| 6          | PC3, PC5, PC11      | PC1, PC2, PC7, PC8  | PC4, PC6, PC9, PC10 |
| 7          | PC6, PC7, PC11      | PC1, PC2, PC3, PC9  | PC4, PC5, PC8, PC10 |
| 8          | PC7, PC8, PC9, PC11 | PC2, PC3, PC6, PC10 | PC1, PC4, PC5       |
| 9          | PC1, PC2, PC7, PC8  | PC3, PC5, PC11      | PC4, PC6, PC9, PC10 |
| 10         | PC4, PC5, PC8, PC10 | PC1, PC2, PC3, PC9  | PC6, PC7, PC11      |
| 11         | PC1, PC4, PC5       | PC2, PC3, PC6, PC10 | PC7, PC8, PC9, PC11 |
| 12         | PC4, PC6, PC9, PC10 | PC3, PC5, PC11      | PC1, PC2, PC7, PC8  |
| 13         | PC4, PC5, PC8, PC10 | PC6, PC7, PC11      | PC1, PC2, PC3, PC9  |
| 14         | PC1, PC4, PC5       | PC7, PC8, PC9, PC11 | PC2, PC3, PC6, PC10 |
| 15         | PC4, PC6, PC9, PC10 | PC1, PC2, PC7, PC8  | PC3, PC5, PC11      |
| 16         | PC1, PC2, PC3, PC9  | PC4, PC5, PC8, PC10 | PC6, PC7, PC11      |
| 17         | PC2, PC3, PC6, PC10 | PC1, PC4, PC5       | PC7, PC8, PC9, PC11 |
| 18         | PC3, PC5, PC11      | PC4, PC6, PC9, PC10 | PC1, PC2, PC7, PC8  |
| 19         | PC1, PC2, PC3, PC9  | PC6, PC7, PC11      | PC4, PC5, PC8, PC10 |
| 20         | PC2, PC3, PC6, PC10 | PC7, PC8, PC9, PC11 | PC1, PC4, PC5       |
| 21         | PC3, PC5, PC11      | PC1, PC2, PC7, PC8  | PC4, PC6, PC9, PC10 |
| 22         | PC6, PC7, PC11      | PC1, PC2, PC3, PC9  | PC4, PC5, PC8, PC10 |
| 23         | PC7, PC8, PC9, PC11 | PC2, PC3, PC6, PC10 | PC1, PC4, PC5       |
| 24         | PC1, PC2, PC7, PC8  | PC3, PC5, PC11      | PC4, PC6, PC9, PC10 |
| 25         | PC4, PC5, PC8, PC10 | PC1, PC2, PC3, PC9  | PC6, PC7, PC11      |
| 26         | PC1, PC4, PC5       | PC2, PC3, PC6, PC10 | PC7, PC8, PC9, PC11 |
| 27         | PC4, PC6, PC9, PC10 | PC3, PC5, PC11      | PC1, PC2, PC7, PC8  |
| 28         | PC4, PC5, PC8, PC10 | PC6, PC7, PC11      | PC1, PC2, PC3, PC9  |
| 29         | PC1, PC4, PC5       | PC7, PC8, PC9, PC11 | PC2, PC3, PC6, PC10 |
| 30         | PC4, PC6, PC9, PC10 | PC1, PC2, PC7, PC8  | PC3, PC5, PC11      |

4. Провести налаштування параметрів *IP*-адресації пристроїв мережі у відповідності до даних п. 3. Перевірити наявність зв'язку між парами пристроїв мережі, що належать до різних *VLAN*. Перевірити наявність зв'язку комп'ютерів до шлюзу останньої черги.

### КОНТРОЛЬНІ ПИТАННЯ

1. Поясніть принципи маршрутизації *VLAN* і яку роль вона відіграє в мережевих оточеннях?
2. Які переваги має використання маршрутизації віртуальних мереж для сегментації мережі?
3. Як визначається шлюз останньої черги і для чого він потрібний в мережі?
4. Як налаштовується маршрутизація між *VLAN* на комутаторі *Cisco*?
5. Як визначаються таблиці маршрутизації для кожної віртуальної мережі на комутаторі?
6. Як забезпечується ізоляція трафіку між різними віртуальними мережами на комутаторі?
7. Як встановлюється *IP*-адреси на інтерфейсах для маршрутизації *VLAN* на комутаторі *Cisco*?
8. Які є відмінності між *IPv4* та *IPv6* маршрутизацією віртуальних мереж на комутаторах *Cisco*?
9. Які є рекомендації щодо безпеки при налаштуванні маршрутизації віртуальних мереж на комутаторах *Cisco*?
10. Як можна проводити моніторинг та аналіз трафіку для кожної *VLAN*?
11. Які обмеження використання методу *router-on-a-stick*?
12. Які переваги використання комутатора Рівня 3 для маршрутизації між *VLAN*?

## 5.2.3 НАЛАШТУВАННЯ *ETHERCHANNEL*

### ОСНОВНІ КОМАНДИ НАЛАШТУВАННЯ *ETHERCHANNEL*

Для агрегування каналів у *Cisco* може бути використаний один із трьох варіантів:

- *LACP* (*Link Aggregation Control Protocol*) – стандартний протокол.
- *PAgP* (*Port Aggregation Protocol*) – пропрієтарний протокол *Cisco*.
- Статичне агрегування без використання протоколів.

Оскільки *LACP* і *PAgP* вирішують одні й ті самі завдання (з невеликими відмінностями за можливостями), то краще використовувати стандартний протокол. Фактично залишається вибір між *LACP* і статичним агрегуванням.

Статичне агрегування не вносить додаткову затримку при піднятті агрегованого каналу або зміні його налаштувань, але також немає узгодження налаштувань з віддаленою стороною. Помилки в налаштуванні можуть призвести до утворення петель.

Агрегування за допомогою *LACP* виконує узгодження налаштувань із віддаленою стороною, що дає змогу уникнути помилок і петель у мережі. Підтримка *standby*-інтерфейсів дає змогу агрегувати до 16-ти портів, 8 з яких будуть активними, а решта в режимі *standby*. Недоліком можна вважати додаткову затримку при піднятті агрегованого каналу або зміні його налаштувань.

При налаштуванні *EtherChannel* треба дотримуватись певних правил:

- *LACP* і *PAgP* групують інтерфейси з однаковими:
  - швидкістю (*speed*),
  - режимом дуплексу (*duplex mode*),
  - *native VLAN*,
  - діапазон дозволених *VLAN*,
  - *trunking status*,
  - типом інтерфейсу.
- Налаштування *EtherChannel*:
  - Оскільки для об'єднання в *EtherChannel* на інтерфейсах мають збігатися багато налаштувань, простіше об'єднувати їх, коли вони налаштовані за замовчуванням. А потім налаштовувати логічний інтерфейс.
  - Перед об'єднанням інтерфейсів краще відключити їх. Це дасть змогу уникнути блокування інтерфейсів *STP* (або переведення їх у стан *err-disable*).
  - Для того щоб видалити налаштування *EtherChannel*, достатньо видалити логічний інтерфейс. Команди *channel-group* видаляться автоматично.
- Створення *EtherChannel* для портів рівня 2 і портів рівня 3 відрізняється:
  - Для інтерфейсів 3го рівня вручну створюється логічний інтерфейс командою **interface port-channel**.

- Для інтерфейсів 2го рівня логічний інтерфейс створюється динамічно.
- Для обох типів інтерфейсів необхідно вручну призначати інтерфейс в *EtherChannel*. Для цього використовується команда **channel-group** у режимі налаштування інтерфейсу. Ця команда пов'язує разом фізичні та логічні порти.

Основною командою налаштування агрегованих каналів є команда **channel-group**:

```
channel-group channel-number [force] [mode {on | active | passive}]
```

де **force** – Вказує на примусове додавання порту локальної мережі до групи каналів.

**mode** – Вказує режим каналу порту інтерфейсу.

**active** – Вказує, що коли ви вмикаєте *LACP*, ця команда вмикає *LACP* на вказаному інтерфейсі. Інтерфейс знаходиться в активному переговорному стані, в якому порт ініціює переговори з іншими портами, надсилаючи пакети *LACP*.

**on** – (режим за замовчуванням) Вказує, що всі канали порту, на яких не запущено *LACP*, залишаються у цьому режимі.

**passive** – Вмикає *LACP*, лише якщо виявлено пристрій *LACP*. Інтерфейс перебуває у стані пасивних переговорів, у якому порт відповідає на отримані пакети *LACP*, але не ініціює переговорів *LACP*. **vlanid** – визначає ідентифікатор *VLAN*.

Для повернення режиму порту увімкненим для вказаного інтерфейсу є команда:

```
no channel-group number mode
```

Вирішення поширених проблем з мережею *EtherChannel* проводиться за допомогою таких основних команд: **show interfaces port-channel**, **show etherchannel summary**, **show etherchannel port-channel**, **show interfaces etherchannel**.

### Модельний приклад налагодження статичного *EtherChannel* 2 рівня

Розглянемо специфіку налагодження статичного *EtherChannel* 2 рівня на комутаторах *Cisco*, схема якої наведена на рисунку 5.47.

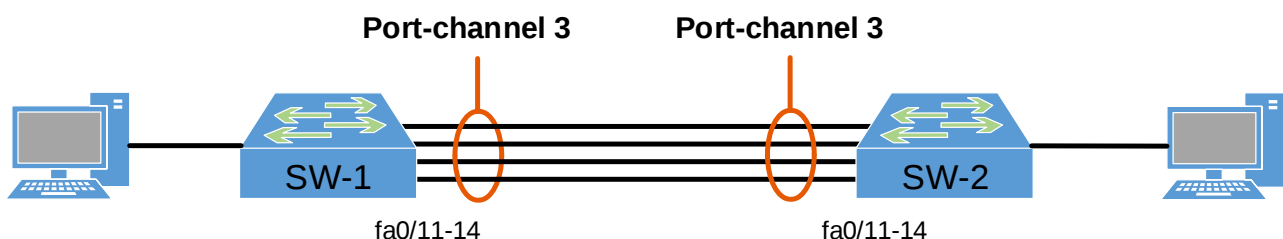


Рисунок 5.47 – Приклад мережі

Послідовність сценаріїв налагодження агрегованих каналів комутаторів **SW-1** та **SW-2** наведені далі:

Налаштування *EtherChannel* на **SW-1**:

```
SW-1(config)#interface range fa0/11-14
SW-1(config-if-range)#shutdown
SW-1(config-if-range)#channel-group 3 mode on
Creating a port-channel interface Port-channel 3
```

Налаштування *EtherChannel* на **SW-2**:

```
SW-2(config)#interface range fa0/11-14
SW-2(config-if-range)#channel-group 3 mode on
Creating a port-channel interface Port-channel 3
```

Вмикання фізичних інтерфейсів на **SW-1**:

```
SW-1(config)# interface range fa0/11-14
SW-1(config-if-range)#no shutdown
```

### Результати виконання команд моніторингу та діагностики для розглянутого прикладу

З метою перегляду інформації про функціонування агрегованих каналів для розглянутого прикладу використано команди **show etherchannel summary**, **show etherchannel port-channel**:

```
SW-1#show etherchannel summary
Flags: D - down P - in port-channel
 I - stand-alone s - suspended
 H - Hot-standby (LACP only)
 R - Layer3 S - Layer2
 U - in use f - failed to allocate aggregator
 u - unsuitable for bundling
 w - waiting to be aggregated
 d - default port
```

```
Number of channel-groups in use: 1
Number of aggregators: 1
```

| Group | Port-channel | Protocol | Ports                                       |
|-------|--------------|----------|---------------------------------------------|
| 3     | Po3 (SU)     | -        | Fa0/11 (P) Fa0/12 (P) Fa0/13 (P) Fa0/14 (P) |

```
SW-1#show etherchannel port-channel
Channel-group listing:

```

```
Group: 3

```

```
Port-channels in the group:

```

```
Port-channel: Po3

```

```

Age of the Port-channel = 00d:00h:02m:21s
Logical slot/port = 2/3 Number of ports = 4
GC = 0x00000000 HotStandBy port = null
Port state = Port-channel
Protocol = PAGP
Port Security = Disabled

```

Ports in the Port-channel:

| Index | Load | Port   | EC state | No of bits |
|-------|------|--------|----------|------------|
| 0     | 00   | Fa0/11 | On       | 0          |
| 0     | 00   | Fa0/12 | On       | 0          |
| 0     | 00   | Fa0/13 | On       | 0          |
| 0     | 00   | Fa0/14 | On       | 0          |

Time since last port bundled: 00d:00h:01m:13s Fa0/14

### Модельний приклад налагодження *EtherChannel* 2 рівня за допомогою *LACP*

Розглянемо специфіку налагодження *EtherChannel* 2 рівня за допомогою *LACP* на комутаторах *Cisco*, схема якої аналогічна минулому прикладу.

Послідовність сценаріїв налагодження агрегованих каналів комутаторів **SW-1** та **SW-2** наведені далі:

Налаштування *EtherChannel* на **SW1**:

```

SW-1(config)#interface range fa0/11-14
SW-1(config-if-range)#shutdown
SW-1(config-if-range)#channel-group 3 mode active
Creating a port-channel interface Port-channel 1

```

Налаштування *EtherChannel* на **SW-2**:

```

SW-2(config)#interface range fa0/11-14
SW-2(config-if-range)#channel-group 3 mode passive
Creating a port-channel interface Port-channel 1

```

Вмикання фізичних інтерфейсів на **SW-1**:

```

SW-1(config)# interface range fa0/11-14
SW-1(config-if-range)#no shutdown

```

### Результати виконання команд моніторингу та діагностики для розглянутого прикладу

З метою перегляду інформації про функціонування агрегованих каналів для розглянутого прикладу використано команди **show etherchannel summary**, **show etherchannel port-channel**, **show lacp 1 internal**, **show lacp 1 neighbor**, **show lacp 1 counters**, **show lacp sys-id**:

Сумарна інформація про стан *Etherchannel*:

SW-1#show etherchannel summary

Flags: D - down P - in port-channel  
I - stand-alone s - suspended  
H - Hot-standby (LACP only)  
R - Layer3 S - Layer2  
U - in use f - failed to allocate aggregator  
u - unsuitable for bundling  
w - waiting to be aggregated  
d - default port

Number of channel-groups in use: 1

Number of aggregators: 1

| Group | Port-channel | Protocol | Ports                                       |
|-------|--------------|----------|---------------------------------------------|
| 1     | Po1 (SU)     | LACP     | Fa0/11 (P) Fa0/12 (P) Fa0/13 (P) Fa0/14 (P) |

Інформація про *port-channel* на SW-1:

SW-1#show etherchannel port-channel

Channel-group listing:

-----

Group: 1

-----

Port-channels in the group:

-----

Port-channel: Po1 (Primary Aggregator)

-----

Age of the Port-channel = 0d:00h:14m:21s

Logical slot/port = 1/0 Number of ports = 4

HotStandBy port = null

Port state = Port-channel Ag-Inuse

Protocol = LACP

Port security = Disabled

Ports in the Port-channel:

| Index | Load | Port   | EC state | No of bits |
|-------|------|--------|----------|------------|
| 0     | 00   | Fa0/11 | Active   | 0          |
| 0     | 00   | Fa0/12 | Active   | 0          |
| 0     | 00   | Fa0/13 | Active   | 0          |
| 0     | 00   | Fa0/14 | Active   | 0          |

Time since last port bundled: 0d:00h:01m:49s Fa0/13

Time since last port Un-bundled: 0d:00h:04m:20s Fa0/14

Інформація про *port-channel* на SW-2:

```
SW-2#show etherchannel port-channel
Channel-group listing:

Group: 1

Port-channels in the group:

Port-channel: Po1 (Primary Aggregator)

Age of the Port-channel = 0d:00h:13m:49s
Logical slot/port = 2/1 Number of ports = 4
HotStandBy port = null
Port state = Port-channel Ag-Inuse
Protocol = LACP
Port security = Disabled

Ports in the Port-channel:
Index Load Port EC state No of bits
-----+-----+-----+-----+-----
 0 00 Fa0/11 Passive 0
 0 00 Fa0/12 Passive 0
 0 00 Fa0/13 Passive 0
 0 00 Fa0/14 Passive 0

Time since last port bundled: 0d:00h:03m:48s Fa0/13
Time since last port Un-bundled: 0d:00h:06m:18s Fa0/14
```

Інформація *LACP* по локальному комутатору:

```
SW-1#show lacp 1 internal
Flags: S - Device is requesting Slow LACPDUs
 F - Device is requesting Fast LACPDUs
 A - Device is in Active mode P - Device is in Passive mode

Channel group 1

Port Flags State LACP port Admin Oper Port Port
Fa0/11 SA bndl 32768 0x1 0x1 0xC 0x3D
Fa0/12 SA bndl 32768 0x1 0x1 0xD 0x3D
Fa0/13 SA bndl 32768 0x1 0x1 0x16 0x3D
Fa0/14 SA bndl 32768 0x1 0x1 0x17 0x3D
```

Інформація *LACP* по віддаленому комутатору:

```
SW-1#show lacp 1 neighbor
Flags: S - Device is requesting Slow LACPDUs
 F - Device is requesting Fast LACPDUs
 A - Device is in Active mode P - Device is in Passive mode

Channel group 1 neighbors
Partner's information:

Port Flags State LACP port Admin Oper Port Port
Fa0/11 SP 32768 000a.b8ab.eb80 5s 0x0 0x1 0x10E 0x3C
Fa0/12 SP 32768 000a.b8ab.eb80 13s 0x0 0x1 0x10F 0x3C
Fa0/13 SP 32768 000a.b8ab.eb80 5s 0x0 0x1 0x110 0x3C
Fa0/14 SP 32768 000a.b8ab.eb80 16s 0x0 0x1 0x111 0x3C
```

Лічильник *LACP*:

```
SW-1# show lacp 1 counters
```

| Port             | LACPDUs |      | Marker |      | Marker Response |      | LACPDUs |     |
|------------------|---------|------|--------|------|-----------------|------|---------|-----|
|                  | Sent    | Recv | Sent   | Recv | Sent            | Recv | Pkts    | Err |
| -----            |         |      |        |      |                 |      |         |     |
| Channel group: 1 |         |      |        |      |                 |      |         |     |
| Fa0/11           | 13      | 11   | 0      | 0    | 0               | 0    | 0       |     |
| Fa0/12           | 13      | 10   | 0      | 0    | 0               | 0    | 0       |     |
| Fa0/13           | 25      | 22   | 0      | 0    | 0               | 0    | 0       |     |
| Fa0/14           | 13      | 11   | 0      | 0    | 0               | 0    | 0       |     |

*LACP* system ID:

```
SW-1# show lacp sys-id
32768, 0012.0111.e580
```

### Моделльний приклад налагодження *EtherChannel* 3 рівня за допомогою *LACP*

Розглянемо специфіку налагодження статичного *EtherChannel* 3 рівня на комутаторах *Cisco*, схема якої наведена на рисунку 5.48.

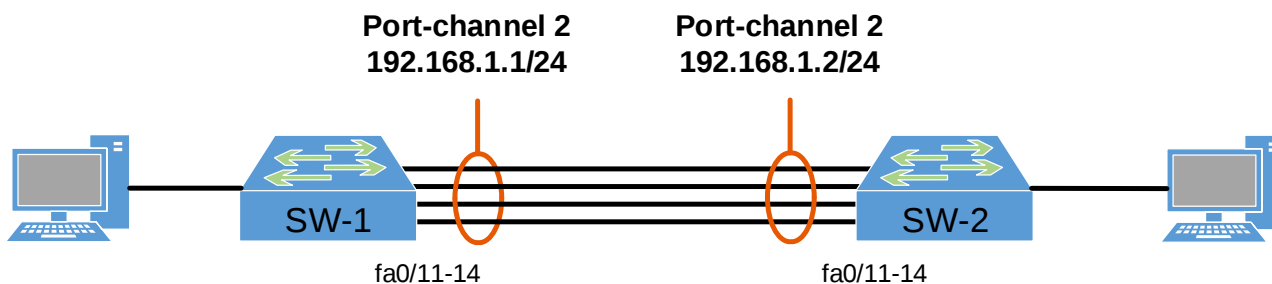


Рисунок 5.48 – Приклад мережі

Послідовність сценаріїв налагодження агрегованих каналів комутаторів **SW-1** та **SW-2** наведені далі:

Налаштування логічного інтерфейсу на **SW-1**:

```
SW-1(config)# int port-channel 2
SW-1(config-if)# no switchport
SW-1(config-if)# ip address 192.168.1.1 255.255.255.0
```

Налаштування фізичних інтерфейсів на **SW-1**:

```
SW-1(config)#interface range fa0/11-14
SW-1(config-if-range)#shutdown
SW-1(config-if-range)#no switchport
SW-1(config-if-range)#channel-group 2 mode active
```

Налаштування логічного інтерфейсу на **SW-2**:

```
SW-2(config)# int port-channel 2
SW-2(config-if)# no switchport
SW-2(config-if)# ip address 192.168.1.2 255.255.255.0
```

Налаштування фізичних інтерфейсів на **SW-2**:

```
SW-2(config)#interface range fa0/11-14
SW-2(config-if-range)#no switchport
SW-2(config-if-range)#channel-group 2 mode passive
```

Вмикання фізичних інтерфейсів на **SW-1**:

```
SW-1(config)# interface range fa0/11-14
SW-1(config-if-range)#no shutdown
```

### Результати виконання команд моніторингу та діагностики для розглянутого прикладу

```
SW-1# show etherchannel summary
Flags: D - down P - bundled in port-channel
 I - stand-alone s - suspended
 H - Hot-standby (LACP only)
 R - Layer3 S - Layer2
 U - in use f - failed to allocate aggregator
```

```
 M - not in use, minimum links not met
 u - unsuitable for bundling
 w - waiting to be aggregated
 d - default port
```

```
Number of channel-groups in use: 2
Number of aggregators: 2
```

| Group | Port-channel | Protocol | Ports                                       |
|-------|--------------|----------|---------------------------------------------|
| 2     | Po2 (RU)     | LACP     | Fa0/11 (P) Fa0/12 (P) Fa0/13 (P) Fa0/14 (P) |

### ПРАКТИЧНЕ ЗАВДАННЯ

**Мета роботи:** ознайомитися з технологіями, які дозволяють об'єднати кілька фізичних каналів в один логічний задля збільшення пропускної здатності і надійності каналу за допомогою комутаційного обладнання *Cisco*.

#### Порядок виконання роботи

1. Створити фізичний проєкт мережі або у середовищі програмного симулятора/емулятора створити мережу (рисунок 5.49). При побудові звернути увагу на вибір моделей комутаторів, мережних модулів та плат, а також мережних з'єднань. При побудові підмережі слід вибирати потрібний тип кабелю для відповідної технології. В якості комутаторів робочих груп взяти комутатори 2 рівня, наприклад *Cisco 2960-24TT*.

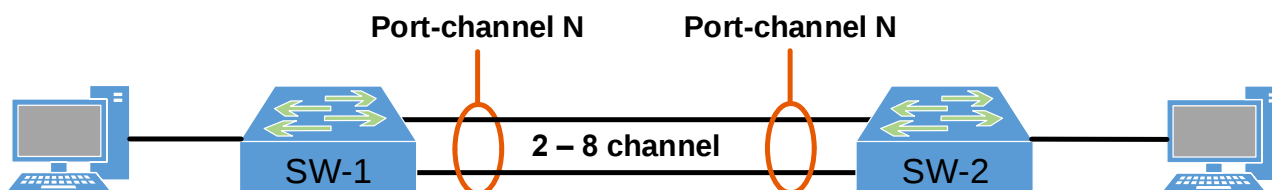


Рисунок 5.49 – Проєкт мережі

2. Провести базове налаштування комутаторів, мережних інтерфейсів, з'єднань, створити агрегований канал між комутаторами. Параметри агрегованого каналу відповідно до варіанту наведені в таблиці 5.45.

Таблиця 5.45 – Параметри агрегованого каналу

| № варіанта | Тип <i>EtherChannel</i> | Кількість каналів | Port-channel, N | IP address        |
|------------|-------------------------|-------------------|-----------------|-------------------|
| 1          | Статичний 2 рівень      | 2                 | 1               | n/a               |
| 2          | LACP 2 рівень           | 2                 | 3               | n/a               |
| 3          | Статичний 3 рівень      | 2                 | 1               | 10.10.10.1-2      |
| 4          | LACP 3 рівень           | 2                 | 3               | 192.168.1.1-2     |
| 5          | Статичний 2 рівень      | 4                 | 1               | n/a               |
| 6          | LACP 2 рівень           | 4                 | 3               | n/a               |
| 7          | Статичний 3 рівень      | 4                 | 1               | 192.168.1.11-12   |
| 8          | LACP 3 рівень           | 4                 | 3               | 10.10.10.11-12    |
| 9          | Статичний 2 рівень      | 6                 | 1               | n/a n/a           |
| 10         | LACP 2 рівень           | 6                 | 3               |                   |
| 11         | Статичний 3 рівень      | 6                 | 1               | 10.10.20.1-2      |
| 12         | LACP 3 рівень           | 6                 | 3               | 192.165.1.1-2     |
| 13         | Статичний 2 рівень      | 8                 | 1               | n/a               |
| 14         | LACP 2 рівень           | 8                 | 3               | n/a               |
| 15         | Статичний 3 рівень      | 8                 | 1               | 192.168.50.11-12  |
| 16         | LACP 3 рівень           | 8                 | 3               | 10.50.10.11-12    |
| 17         | Статичний 2 рівень      | 2                 | 2               | n/a               |
| 18         | LACP 2 рівень           | 2                 | 4               | n/a               |
| 19         | Статичний 3 рівень      | 2                 | 2               | 10.40.10.1-2      |
| 20         | LACP 3 рівень           | 2                 | 4               | 192.168.66.1-2    |
| 21         | Статичний 2 рівень      | 4                 | 2               | n/a               |
| 22         | LACP 2 рівень           | 4                 | 4               | n/a               |
| 23         | Статичний 3 рівень      | 4                 | 2               | 192.168.222.11-12 |
| 24         | LACP 3 рівень           | 4                 | 4               | 10.10.222.11-12   |
| 25         | Статичний 2 рівень      | 6                 | 2               | n/a               |
| 26         | LACP 2 рівень           | 6                 | 4               | n/a               |
| 27         | Статичний 3 рівень      | 6                 | 2               | 10.10.99.1-2      |
| 28         | LACP 3 рівень           | 6                 | 4               | 192.168.99.1-2    |
| 29         | Статичний 2 рівень      | 8                 | 2               | n/a               |
| 30         | LACP 2 рівень           | 8                 | 4               | n/a               |

## КОНТРОЛЬНІ ПИТАННЯ

1. Що таке агрегування каналів у *Cisco* мережах і як воно працює?
2. Які переваги надає агрегування каналів у мережах *Cisco*?
3. Які типи агрегування каналів підтримуються у *Cisco* мережах?
4. Як визначити, чи працює агрегація каналів між двома комутаторами *Cisco*?
5. Як вибрати правильний протокол агрегування каналів у *Cisco* мережах?
6. Як налаштувати динамічне агрегування каналів на комутаторах *Cisco*?
7. Як визначити, який канал в агрегованій групі є активним?
8. Що таке *LACP* і як воно використовується в агрегації каналів у *Cisco* мережах?
9. Що таке *PAgP* і як воно використовується в агрегації каналів у *Cisco* мережах?
10. Як визначити максимальну швидкість агрегованого каналу на комутаторі *Cisco*?
11. Як діагностувати проблеми з агрегацією каналів на комутаторах *Cisco*?
12. Як перевірити статус агрегації каналів на комутаторах *Cisco*?
13. Як видалити агреговану групу з комутатора *Cisco*?
14. Як вибрати оптимальну конфігурацію агрегації каналів для конкретної мережної ситуації?
15. Які є найкращі практики для резервного агрегування каналів для забезпечення надійності мережі?

## 5.2.4 НАЛАШТУВАННЯ СЕРВЕРА *DHCPv4*

### ПОРЯДОК НАЛАГОДЖЕННЯ *DHCP*-СЕРВЕРА НА БАЗІ МАРШРУТИЗАТОРА *CISCO*

У практиці побудови мереж маршрутизатор *Cisco* у більшості випадків налагоджується як *DHCP*-сервер, у деяких випадках – як зв'язний агент *DHCP* і досить рідко – як *DHCP*-клієнт. Налагодження функціонування *DHCP*-сервера на базі маршрутизатора *Cisco* згідно з рекомендаціями виробника складається із певних обов'язкових та необов'язкових етапів. Порядок виконання згаданих етапів є таким:

1. Включити функціонування *DHCP*-сервера на маршрутизаторі (залежно від ситуації, за замовчуванням запускається автоматично).
2. Налагодити *DHCP Database Agent* або відключити *DHCP Conflict Logging* (обов'язково).
3. Виключити *IP*-адреси, які не будуть призначатися *DHCP*-клієнтам (обов'язково).
4. Створити та налагодити набір (набори) *IP*-адрес, які будуть призначатися *DHCP*-клієнтам (обов'язково).
5. Налагодити ручне призначення *IP*-адрес (необов'язково).
6. Налагодити параметри завантажувального файлу для даного *DHCP*-сервера (необов'язково).
7. Зазначити кількість перевірочних запитів протоколу *ICMP* (необов'язково).
8. Встановити значення тайм-ауту для перевірочних запитів протоколу *ICMP* (необов'язково).
9. При налагодженні зв'язного агенту *DHCP* для даного порядку додаються наступні етапи :
10. Активувати *Cisco IOS DHCP*-клієнта на інтерфейсах *Ethernet/Fast Ethernet ...* (необов'язково).
11. Налагодити параметри імпорту опцій *DHCP*-сервера та автоконфігурування (необов'язково).
12. Налагодити параметри опцій зв'язного агенту *DHCP* в повідомленнях *BOOTREPLY* (необов'язково).
13. Налагодити параметри політики передачі для зв'язного агенту *DHCP* (необов'язково).
14. Активувати додаткові можливості зв'язного агенту *DHCP* (необов'язково).

На практиці можливе застосування іншого порядку.

## КОМАНДИ НАЛАГОДЖЕННЯ *DHCP*-СЕРВЕРА НА БАЗІ МАРШРУТИЗАТОРА *CISCO*

Включення функціонування *DHCP*-сервера на маршрутизаторі *Cisco* виконується командою **service dhcp**. Виключення — командою **no service dhcp**. За замовчуванням на маршрутизаторі *Cisco* *DHCP*-сервер є включеним. Якщо використання *DHCP*-сервера не планується, то з метою підвищення рівня захисту пристрою рекомендується даний функціонал відключати.

Основною командою, від якої походить більшість команд для налагодження засобів протоколу *DHCP* у *Cisco IOS* є команда **ip dhcp**. Перелік та призначення похідних команд можна отримати за допомогою інтерактивної довідки командного рядка. Слід зауважити, що для різних моделей маршрутизаторів, версій *IOS* та наборів властивостей, переліки можуть відрізнятися. У більшості випадків у режимі конфігурування протоколу маршрутизації доступними є такі команди **ip dhcp aaa**, **ip dhcp binding**, **ip dhcp bootp**, **ip dhcp class**, **ip dhcp compatibility**, **ip dhcp conflict**, **ip dhcp database**, **ip dhcp excluded-address**, **ip dhcp limit**, **ip dhcp limited-broadcast-address**, **ip dhcp ping**, **ip dhcp pool**, **ip dhcp relay**, **ip dhcp smart-relay**, **ip dhcp update**, **ip dhcp use**. У деяких випадках використовуються і інші команди. Призначення та синтаксис основних команд наведено нижче.

Створення або редагування набору (пулу) адрес, які будуть видаватися *DHCP*-клієнтами, виконується командою **ip dhcp pool**. Після виконання даної команди здійснюється перехід до режиму налагодження протоколу *DHCP*. У цьому режимі наявно більше ніж 25 команд, які стосуються різних аспектів налагодження пулу адрес протоколу *DHCP*.

Після переходу до режиму налагодження протоколу *DHCP* наступним кроком є зазначення *IP*-адреси та маски (префіксу) мережі, адреси якої будуть призначатися *DHCP*-клієнтам. Для цього використовується команда **network**. Ще одним важливим кроком є зазначення *IP*-адреси шлюзу за замовчуванням для даної мережі. Для цього використовується команда **default-router**. Можливе використання до 8 шлюзів. На практиці достатньо одного. Наступними (не обов'язковими) кроками є зазначення сервера (серверів) служб *DNS*, *WINS*, а також типу вузлів *NetBIOS* для *WINS*. Для цього виконуються відповідно команди **dns-server**, **netbios-name-server**, **netbios-node-type**. *IP*-адреси, що вказуються як параметри цих команд, можуть належати іншим мережам. *DHCP*-сервер може також видавати назву домену. Для цього використовується команда **domain-name**. Важливим параметром налагодження *DHCP*-маршрутизатора є час, на який виділяється *IP*-адреса, відомий також як час оренди адреси. Для його налагодження використовується команда **lease**.

Для вилучення з пулу *IP*-адрес, які не будуть призначатися, використовується команда **ip dhcp excluded-address**. Цією командою можна вилучити

як одну адресу, так і певний діапазон адрес. З метою усунення конфліктів вилучення IP-адрес рекомендується виконувати перед створенням пулу.

Перед виділенням IP-адреси з пулу алгоритмом роботи DHCP-сервера передбачена попередня перевірка, чи дійсно дана адреса є вільною. Для цього DHCP-сервер двічі посилає ICMP-запит за даною адресою. Якщо відповіді на запит немає, то DHCP-сервер вважає, що адреса є вільною і надає її клієнтові. Параметри даної перевірки (кількість запитів, інтервал між ними) можна змінити. Для цього використовуються команди **ip dhcp ping packets** та **ip dhcp ping timeout**. Для активації/деактивації журналювання конфліктів адрес використовується команда **ip dhcp conflict logging**.

Одним з режимів роботи DHCP є ручне призначення IP-адрес. У цьому випадку для кожного клієнта формується окремий пул і в цьому пулі за допомогою спеціалізованих команд проводиться налагодження призначення адреси. Для налагодження ручного призначення IP-адрес використовуються спеціальні команди **host**, **client-identifier**, **client-name**. Для зазначення IP-адреси шлюзу за замовчуванням, IP-адрес DNS-сервера та WINS-сервера використовуються вищезгадані команди **default-router**, **dns-server**, **netbios-name-server** та деякі інші.

Завантаження конфігурації для DHCP-сервера можливе з внутрішнього (Flash-пам'ять) або зовнішнього джерела (TFP, TFTP, RCP-сервери). Для цього використовуються команди **bootfile** та **ip dhcp database**.

Синтаксис команди **ip dhcp pool** (режим глобального конфігурування):

```
ip dhcp pool name,
```

де **name** – текстова назва (англійською мовою) набору IP-адрес, які будуть призначатися.

Синтаксис команди **network** (режим конфігурування протоколу DHCP):

```
network network_IP-address [network_mask | /prefix-length],
```

де **network\_IP-address** – IP-адреса мережі, з якої призначаються IP-адреси вузлам;

**network\_mask** – маска мережі для IP-адреси, що призначається, записана у звичайній формі (необов'язково, може вказуватися префікс);

**prefix-length** – довжина префікса для IP-адреси, що призначається (необов'язково, може вказуватися маска).

Синтаксис команди **domain-name** (режим конфігурування протоколу DHCP):

```
domain-name domain_name,
```

де **domain\_name** – текстова назва домену для клієнта.

Синтаксис команди **default-router** (режим конфігурування протоколу *DHCP*):

```
default-router IP-address [IP-address2 ... IP-address8],
```

де ***IP-address*** – IP-адреса першого шлюзу за замовчуванням;

***IP-address2* ...** – IP-адреса наступного шлюзу за замовчуванням.

Синтаксис команд **dns-server**, **netbios-name-server** аналогічний синтаксису команди **default-router**.

Синтаксис команди **netbios-node-type** (режим конфігурування протоколу *DHCP*):

```
netbios-node-type type,
```

де ***type*** – тип вузла NetBIOS (може набувати значень **b-node**, **p-node**, **m-node**, **h-node**); рекомендоване значення **h-node**.

Синтаксис команди **lease** (режим конфігурування протоколу *DHCP*):

```
lease {days [hours][minutes] | infinite},
```

де ***days*** – кількість днів оренди;

***hours*** – кількість годин оренди (необов'язково);

***minutes*** – кількість хвилин оренди (необов'язково);

***infinite*** – час оренди необмежений.

Синтаксис команди **host** (режим конфігурування протоколу *DHCP*):

```
host IP-address [network_mask | /prefix-length],
```

де ***IP-address*** – IP-адреса, яка виділяється вузлові;

***network\_mask*** – маска мережі для IP-адреси, що призначається, записана у звичайній формі (необов'язково, може вказуватися префікс);

***prefix-length*** – довжина префіксу для IP-адреси, що призначається (необов'язково, може вказуватися маска).

Синтаксис команди **client-identifier** (режим конфігурування протоколу *DHCP*):

```
client-identifier unique-identifier,
```

де ***unique-identifier*** – унікальний семибайтний ідентифікатор клієнта записаний у шістнадцятковій формі; формується як байт, що позначає середовище (для *Ethernet* – 01h) та фізична (*MAC*) адреса клієнта.

Синтаксис команди **client-name** (режим конфігурування протоколу *DHCP*):

```
client-name client_name,
```

де ***client\_name*** – текстовий ідентифікатор клієнта.

Синтаксис команди **ip dhcp excluded-address** (режим глобального конфігурування):

```
ip dhcp excluded-address low_IP_address
[high_IP_address],
```

де **low\_IP\_address** – початкова IP-адреса діапазону (вона вказується також у випадку, якщо вилучається одна адреса);

**high\_IP\_address** – кінцева IP-адреса діапазону.

Синтаксис команди **ip dhcp ping packets** (режим глобального конфігурування):

```
ip dhcp ping packets number,
```

де **number** – кількість перевірочних запитів, за замовчуванням – 2.

Синтаксис команди **ip dhcp ping timeout** (режим глобального конфігурування):

```
ip dhcp ping timeout interval,
```

де **interval** – значення інтервалу між перевірочними запитами; зазначається у мілісекундах; за замовчуванням – 500 мс.

Синтаксис команди **ip dhcp conflict logging** (режим глобального конфігурування):

```
ip dhcp ping conflict logging.
```

З синтаксисом та особливостями використання решти команд можна ознайомитися у технічній документації.

### КОМАНДИ МОНІТОРИНГУ, ДІАГНОСТИКИ ТА КЕРУВАННЯ ПРОЦЕСОМ РОБОТИ СКЛАДОВИХ ПРОТОКОЛУ *DHCP* НА МАРШРУТИЗАТОРІ *CISCO*

Для моніторингу та діагностики роботи протоколу *DHCP* на маршрутизаторах *Cisco* використовуються як команди загального призначення, так і спеціалізовані команди. Серед команд загального призначення можна виділити такі команди: **show running-config**, **show startup-config**, **show interface interface-type interface-id**. Остання команда дає змогу визначити яким чином призначена *IP*-адреса відповідному інтерфейсу. Перелік спеціалізованих команд є відносно невеликим і включає такі команди як: **show ip dhcp binding**, **show ip dhcp conflict**, **show ip dhcp database**, **show ip dhcp import**, **show ip dhcp pool**, **show ip dhcp relay**, **show ip dhcp server statistics**.

Важливими командами, які допомагають зрозуміти процеси передачі повідомлень протоколу є команди трасування такі як: **debug ip dhcp server class**, **debug ip dhcp server events**, **debug ip dhcp server linkage**, **debug ip dhcp server packet**. Також важливими є команди, які дають змогу впливати на процес роботи складових протоколу такі як: **clear**

**ip dhcp binding, clear ip dhcp conflict, clear ip dhcp pool, clear ip dhcp server statistics, clear ip dhcp subnet.**

Узагальнений перелік команд моніторингу, діагностики та керування процесом роботи складових протоколу *DHCP* на маршрутизаторі *Cisco* наведений у таблиці 5.46.

Таблиця 5.46 – Перелік команд моніторингу, діагностики та керування процесом роботи складових протоколу *DHCP* на маршрутизаторі *Cisco*

| Команда                                | Призначення                                                                                         |
|----------------------------------------|-----------------------------------------------------------------------------------------------------|
| Команди <b>show ip dhcp</b>            |                                                                                                     |
| <b>show ip dhcp binding</b>            | Виведення інформації про видані вузлам <i>IP</i> -адреси                                            |
| <b>show ip dhcp conflict</b>           | Виведення інформації про конфлікти при виділенні <i>IP</i> -адрес                                   |
| <b>show ip dhcp database</b>           | Виведення інформації про розміщення та стан бази даних <i>DHCP</i>                                  |
| <b>show ip dhcp import</b>             | Виведення інформації про імпортовані параметри протоколу                                            |
| <b>show ip dhcp pool</b>               | Виведення інформації про параметри та стан використання пулів <i>IP</i> -адрес                      |
| <b>show ip dhcp relay</b>              | Виведення інформації про роботу зв'язного агента <i>DHCP</i>                                        |
| <b>show ip dhcp server statistics</b>  | Виведення статистичної інформації про роботу <i>DHCP</i> -сервера                                   |
| Команди <b>debug ip dhcp</b>           |                                                                                                     |
| <b>debug ip dhcp server class</b>      | Активувати виведення інформації, пов'язаної з класовою адресацією                                   |
| <b>debug ip dhcp server events</b>     | Активувати виведення інформації про події, пов'язані з роботою <i>DHCP</i> -сервера                 |
| <b>debug ip dhcp server linkage</b>    | Активувати виведення інформації, пов'язаної з операціями з базою даних <i>DHCP</i> -сервера         |
| <b>debug ip dhcp server packet</b>     | Активувати декодування інформації з отриманих і відправлених повідомлень                            |
| Команди <b>clear ip dhcp</b>           |                                                                                                     |
| <b>clear ip dhcp binding</b>           | Очистити інформацію про автоматичне призначення <i>IP</i> -адрес у базі даних протоколу <i>DHCP</i> |
| <b>clear ip dhcp conflict</b>          | Очистити інформацію про конфлікти в базі даних <i>DHCP</i>                                          |
| <b>clear ip dhcp pool</b>              | Очистити пул <i>IP</i> -адрес                                                                       |
| <b>clear ip dhcp server statistics</b> | Очистити лічильники <i>DHCP</i> -сервера                                                            |
| <b>clear ip dhcp subnet</b>            | Очистити інформацію про підмережі                                                                   |

## НАЛАГОДЖЕННЯ, МОНІТОРИНГ ТА ДІАГНОСТИКА РОБОТИ DHCP-КЛІЄНТІВ СУЧАСНИХ МЕРЕЖНИХ ОС

Більшість сучасних мережних клієнтських ОС мають вбудовані засоби динамічного отримання параметрів IP-адресації. В ОС *Windows* наявний реалізований у вигляді служби DHCP-клієнт, який дає змогу вирішувати питання призначення адрес як за допомогою технології *APIPA*, так і за допомогою протоколу *DHCP*. Окрім цього, на нього покладаються завдання по реєстрації і оновленню *DNS*-записів. За замовчуванням служба DHCP-клієнта ОС *Windows* запускається автоматично. При потребі можливе її відключення стандартними засобами.

Визначити актуальні параметри IP-адресації певного мережного адаптера чи підключення в ОС *Windows* можна як за допомогою засобів графічного інтерфейсу, так і з використанням командного рядка. Основним засобом в останньому випадку є команда **ipconfig**. Найбільш інформативним варіантом команди є використання з параметром **/all** – команда **ipconfig /all**. Для керування процесом модифікації параметрів IP-адресації мережного адаптера/підключення також використовується команда **ipconfig** з відповідними параметрами: параметр **/release** призначений для вивільнення адреси, параметр **/renew** – для отримання нових параметрів IP-адресації або їх оновлення. Більш детальну інформацію про застосування команди можна отримати за допомогою довідки ОС.

Для ОС *Unix/Linux* реалізовано кілька програмних реалізацій DHCP-клієнтів. Для ОС, які орієнтовані на використання мінімуму обчислювальних ресурсів (наприклад, вбудованих систем), розроблено DHCP-клієнт *Udhcpc* (*μDHCPc*, *Micro DHCP Client*). Цей клієнт застосовується у таких відомих системах як *Busybox*, *Linux MicroCore*, *Linux TinyCore*. Програмно DHCP-клієнт *Udhcpc* реалізований у вигляді однойменної утиліти. Параметри роботи **udhcpc** зберігаються у відповідних конфігураційних файлах. Для багатьох популярних ОС *Linux*, наприклад, ОС *Debian*, розроблена стандартна утиліта **dhclient**. Запуск DHCP-клієнта у більшості випадків в ОС *Linux* виконується автоматично.

Визначити актуальні параметри IP-адресації певного мережного адаптера чи підключення в ОС *Linux*, як і в ОС *Windows* можна як за допомогою засобів графічного інтерфейсу (якщо він підтримується), так і з використанням командного рядка. Основним засобом в останньому випадку є команда **ifconfig**. На відміну від утиліти ОС *Windows* **ipconfig** ця команда не така інформативна, тому доводиться використовувати інші команди, наприклад, для виведення маршруту за замовчуванням – команду **route**. Функціональні можливості **ifconfig** теж відрізняються від **ipconfig**. За допомогою **ifconfig** неможливо керувати процесом вивільнення чи оновлення параметрів IP-адресації – для цього необхідно скористатися утилітою **dhclient** з відповідними параметрами: параметр **-r** призначений для вивільнення адреси, параметр **-v** – для отримання нових параметрів IP-адресації або їх оновлення. Для отримання повної інформації стосовно

параметрів *IP*-адресації для різновидів ОС *Unix/Linux* рекомендується використовувати технічну документацію.

Мережний пристрій *Cisco* (маршрутизатора, комутатор тощо) в певних випадках теж може бути *DHCP*-клієнтом. Найпростіший спосіб у такому разі налагодити динамічне отримання параметрів *IP*-адресації – на обраному інтерфейсі у режимі конфігурування інтерфейсу виконати команду **ip address dhcp**. Можливі і інші способи, особливості їх використання описані у технічній документації.

### Модельний приклад налагодження функціонування *DHCP*-сервера на базі маршрутизатора *Cisco*

Розглянемо специфіку налагодження роботи протоколу *DHCP* для мережі, схема якої наведена на рисунку 5.50. Параметри *IP*-адресації комутатора та маршрутизатора призначені статично. У даній мережі передбачається функціонування одного сервера та одного мережного принтера, параметри *IP*-адресації яких призначені статично та 7 робочих станцій, параметри *IP*-адресації яких призначаються динамічно.

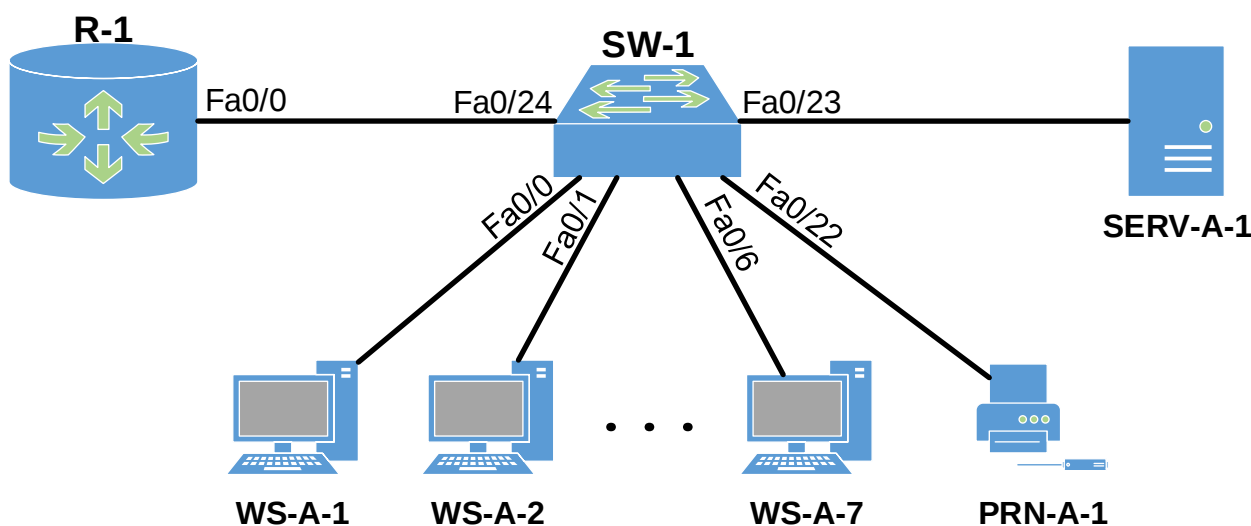


Рисунок 5.50 – Приклад мережі

При побудові даної мережі для з'єднання пристроїв використано дані таблиці 5.47.

Загальна кількість *IP*-адрес, які передбачається використовувати у мережі становить 11. Зокрема, це 4 статичні *IP*-адреси (*IP*-адреси маршрутизатора, комутатора, сервера та мережного принтера) та 7 динамічних *IP*-адрес (*IP*-адреси робочих станцій). Як *IP*-адресу мережі застосуємо адресу **195.1.1.0**. З метою економного використання адресного простору для даної мережі застосуємо маску **255.255.255.240**. За такої маски у мережі можливе використання лише 14 *IP*-адрес (діапазон **195.1.1.1–195.1.1.14**), що цілком задовольняє умовам прикладу.

Таблиця 5.47 – Параметри з'єднань пристроїв та каналів для прикладу

| Пристрій                     | Інтерфейс | Підключення до пристрою | Підключення до інтерфейсу |
|------------------------------|-----------|-------------------------|---------------------------|
| Маршрутизатор <b>R-1</b>     | Fa0/0     | Комутатор SW-1          | Fa0/24                    |
| Комутатор <b>SW-1</b>        | Fa0/0     | Робоча станція WS-A-1   | Fa0                       |
|                              | Fa0/1     | Робоча станція WS-A-2   | Fa0                       |
|                              | ...       | ...                     | ...                       |
|                              | Fa0/6     | Робоча станція WS-A-7   | Fa0                       |
|                              | ...       | ...                     | ...                       |
|                              | F0/22     | Принтер PRN-A-1         | Fa0                       |
|                              | Fa0/23    | Сервер SERV-A-1         | Fa0                       |
|                              | Fa0/24    | Маршрутизатор R-1       | Fa0/0                     |
| Робоча станція <b>WS-A-1</b> | Fa0       | Комутатор SW-1          | Fa0/0                     |
| Робоча станція <b>WS-A-2</b> | Fa0       | Комутатор SW-1          | Fa0/1                     |
| ...                          | ...       | ...                     | ...                       |
| Робоча станція <b>WS-A-7</b> | Fa0       | Комутатор SW-1          | Fa0/6                     |
| Принтер <b>PRN-A-1</b>       | Fa0       | Комутатор SW-1          | Fa0/22                    |
| Сервер <b>SERV-A-1</b>       | Fa0       | Комутатор SW-1          | Fa0/23                    |

Виконаємо узагальнений розподіл *IP*-адрес за використанням для *DHCP*-сервера. Результати розподілу наведені у таблиці 5.48.

Таблиця 5.48 – Узагальнений розподіл IP-адрес мережі за використанням

| Вид адрес                      | Кількість | Діапазони IP-адрес/<br>Окремі IP-адреси | Адреси DHCP-сервером |
|--------------------------------|-----------|-----------------------------------------|----------------------|
| Підмережа А                    |           |                                         |                      |
| Динамічні IP-адреси            | 7         | 195.1.1.2–195.1.1.8                     | призначаються        |
| Статичні IP-адреси             | 4         | 195.1.1.1,<br>195.1.1.12–195.1.1.14     | не призначаються     |
| Адреси, що не використовуються | 3         | 195.1.1.9–195.1.1.11                    |                      |

Для налагодження параметрів адресації пристроїв використано дані таблиці 5.49.

Таблиця 5.49. – Параметри IP-адресації мережі

| Мережа / Пристрій                      | Інтерфейс/Мережний адаптер/Шлюз | IP-адреса  | Маска           | Префікс |
|----------------------------------------|---------------------------------|------------|-----------------|---------|
| Підмережа А                            | —                               | 195.1.1.0  | 255.255.255.240 | /28     |
| Маршрутизатор <b>R-1</b>               | Інтерфейс Fa0/0                 | 195.1.1.1  | 255.255.255.240 | /28     |
| Комутатор <b>SW-1</b>                  | Інтерфейс VLAN 1                | 195.1.1.14 | 255.255.255.240 | /28     |
|                                        | Шлюз за замовчуванням           | 195.1.1.1  | —               | —       |
| Сервер <b>SERV-A-1</b>                 | Мережний адаптер                | 195.1.1.13 | 255.255.255.240 | /28     |
|                                        | Шлюз за замовчуванням           | 195.1.1.1  | —               | —       |
| Принтер <b>PRN-A-1</b>                 | Інтерфейс                       | 195.1.1.12 | 255.255.255.240 | /28     |
|                                        | Шлюз за замовчуванням           | 195.1.1.1  | —               | —       |
| Робоча станція <b>WS-A-1</b> (Linux)   | Мережний адаптер                | DHCP       |                 |         |
|                                        | Шлюз за замовчуванням           |            |                 |         |
| Робоча станція <b>WS-A-2</b> (Windows) | Мережний адаптер                | DHCP       |                 |         |
|                                        | Шлюз за замовчуванням           |            |                 |         |
| ...                                    | ...                             | ...        |                 |         |
| Робоча станція <b>WS-A-7</b> (...)     | Мережний адаптер                | DHCP       |                 |         |
|                                        | Шлюз за замовчуванням           |            |                 |         |

Сценарій налагодження параметрів статичної адресації комутатора **SW-1** та сценарій налагодження маршрутизатора **R-1** як *DHCP*-сервера наведені нижче.

```
SW-1>enable
SW-1(config)#interface Vlan 1
SW-1(config-if)#ip address 195.1.1.14 255.255.255.240
SW-1(config-if)#no shutdown
SW-1(config-if)#exit
SW-1(config)#ip default-gateway 195.1.1.1
SW-1(config)#ip name-server 196.1.1.10
SW-1(config)#ip domain-name my.net
SW-1(config)#no ip domain-lookup
SW-1(config)#exit
SW-1#
```

```
R-1>enable
R-1#configure terminal
R-1(config)#interface FastEthernet 0/0
R-1(config-if)#description LAN_A
R-1(config-if)#ip address 195.1.1.1 255.255.255.240
R-1(config-if)#no shutdown
R-1(config-if)#exit
R-1(config)#service dhcp
R-1(config)#ip dhcp excluded-address 195.1.1.1
R-1(config)#ip dhcp excluded-address 195.1.1.9 195.1.1.14
R-1(config)#ip dhcp pool LAN_A
R-1(dhcp-config)#network 195.1.1.0 255.255.255.240
R-1(dhcp-config)#default-router 195.1.1.1
R-1(dhcp-config)#domain-name my.net
R-1(dhcp-config)#dns-server 196.1.1.10
R-1(dhcp-config)#exit
R-1(config)#exit
R-1#
```

## Результати виконання команд моніторингу та діагностики роботи протоколу *DHCP* для розглянутого прикладу

З метою перегляду інформації про роботу протоколу *DHCP* на маршрутизаторі *Cisco* для розглянутого прикладу використано команди **show ip dhcp pool**, **show ip dhcp binding**, **show ip dhcp server statistics**. Результати роботи цих команд для маршрутизатора **R-1** наведено далі:

```
R-1#show ip dhcp pool
```

```
Pool LAN_A :
```

```
Utilization mark (high/low) : 100 / 0
```

```
Subnet size (first/next) : 0 / 0
```

```
Total addresses : 14
```

```
Leased addresses : 3
```

```
Pending event : none
```

```
1 subnet is currently in the pool :
```

| Current index | IP address range       | Leased addresses |
|---------------|------------------------|------------------|
| 195.1.1.5     | 195.1.1.1 - 195.1.1.14 | 3                |

```
R-1#show ip dhcp binding
```

```
Bindings from all pools not associated with VRF:
```

| IP address | Client-ID/<br>Hardware address/<br>User name | Lease expiration    | Type      |
|------------|----------------------------------------------|---------------------|-----------|
| 195.1.1.2  | 0108.0027.7f91.0a                            | Nov 14 2015 9:50 AM | Automatic |
| 195.1.1.3  | 0108.0027.c9c8.30                            | Nov 14 2015 9:51 AM | Automatic |
| 195.1.1.4  | 0100.5079.6668.01                            | Nov 14 2015 9:56 AM | Automatic |

```
R-1#show ip dhcp server statistics
```

```
Memory usage 24463
```

```
Address pools 1
```

```
Database agents 0
```

```
Automatic bindings 3
```

```
Manual bindings 0
```

```
Expired bindings 0
```

```
Malformed messages 0
```

```
Secure arp entries 0
```

```
Message Received
```

```
BOOTREQUEST 0
```

```
DHCPDISCOVER 9
```

```
DHCPREQUEST 10
```

```
DHCPDECLINE 0
```

```
DHCPRELEASE 1
```

```
DHCPINFORM 0
```

```
Message Sent
```

```
BOOTREPLY 0
```

```
DHCPOFFER 9
```

```
DHCPACK 3
```

```
DHCPNAK 0
```

З метою перегляду інформації про роботу протоколу *DHCP* на робочих станція **WS-A-1** (*Linux*) та **WS-A-2** (*Windows*) відповідно використано команди **ifconfig** та **ipconfig** з необхідними параметрами. Результати їх виконання наведено далі:

```
root@WS-A-1~$ifconfig eth0
eth0 Link encap:Ethernet HWaddr 08:00:27:7F:91:0A
 inet addr:195.1.1.2 Bcast:195.1.1.15 Mask:255.255.255.240
 inet6 addr: fe80::a00:27ff:fe7f:910a/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:148 errors:0 dropped:0 overruns:0 frame:0
 TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:26368 (25.7 KiB) TX bytes:4014 (3.9KiB)
 Interrupt:10 Base address:0xd020
```

```
C:\Documents and Settings\Admin>ipconfig /all

Windows IP Configuration

Host Name : WS-A-2
Primary DNS Suffix :
Node Type : неизвестный
IP Routing Enabled. : нет
WINS Proxy Enabled. : нет

Ethernet adapter Ethernet:

Description. : AMD PCNET Family Ethernet Adapter (PCI)
Physical Address. : 08-00-27-C9-C8-30
DHCP Enabled. : да
Autoconfiguration Enabled : да
IPv4 Address. : 195.1.1.3
Subnet Mask : 255.255.255.240
Default Gateway : 195.1.1.1
DHCP Server : 195.1.1.1
DNS Servers : 196.1.1.10
Lease Obtained. : 14 ноября 2015 г. 9:50:46
Lease Expires : 15 ноября 2015 г. 9:50:46
```

Для дослідження поведінки *DHCP*-сервера при виявленні конфліктів змодельована ситуація, коли одній із робочих станцій (**WS-C**) параметри *IP*-адресації призначено статично. Сервером виявлено конфлікт адрес та виведено системне повідомлення-попередження. Для виведення детальної інформації про конфлікт використано команду **show ip dhcp conflict**. Результат виконання даної команди наведено далі:

```
R-1#show ip dhcp conflict
IP address Detection method Detection time VRF
195.1.1.2 Ping Nov 14 2015 10:50 AM
```

## ПРАКТИЧНЕ ЗАВДАННЯ

**Мета роботи:** ознайомитися з особливостями функціонування та налагодження роботи протоколу динамічного конфігурування вузлів *DHCP* на обладнанні *Cisco*; отримати практичні навички налагодження, моніторингу та діагностування роботи *DHCP*-сервера на базі маршрутизатора *Cisco*; отримати практичні навички налагодження, моніторингу та діагностування роботи *DHCP*-клієнтів різних ОС; дослідити процес роботи протоколу *DHCP* та процеси передачі даних у побудованій мережі.

### Порядок виконання роботи

1. Створити фізичний проект мережі або у середовищі програмного симулятора/емулятора (рисунок 5.51). При побудові звернути увагу на вибір моделей мережних пристроїв, мережних модулів та адаптерів, а також мережних з'єднань. Різновиди технологій *Ethernet* для підмереж **A**, **B**, **C** обираються довільно. Кожну з підключених локальних мереж **A** та **B** показати за допомогою комутаторів та точок доступу. Для вибору кількості серверів, комутаторів, точок доступу слід скористатися даними таблиці 5.50. Кількість підключених робочих станцій та мережних принтерів для кожної мережі – довільна, але не менше 2-х пристроїв одного типу на один комутатор або одну точку доступу. Для побудованої мережі заповнити описову таблицю, яка аналогічна таблиці 5.47.

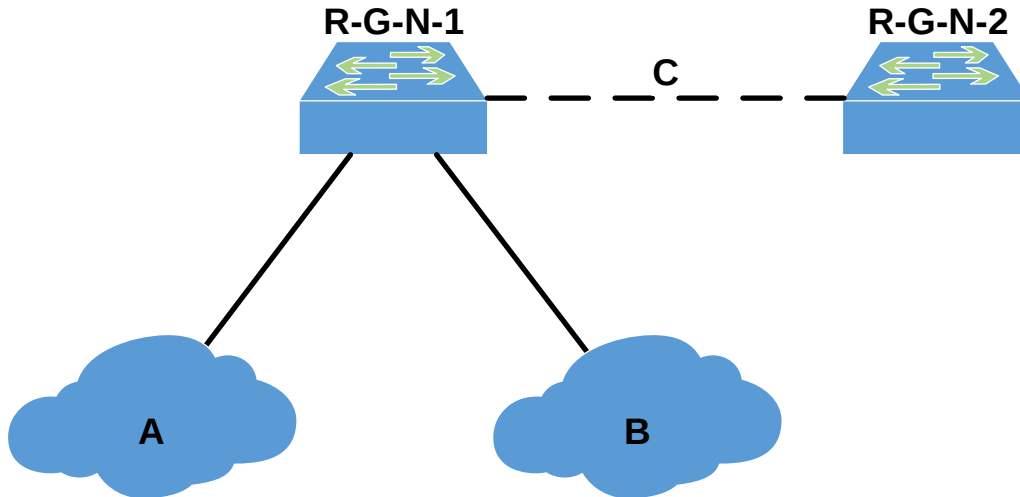


Рисунок 5.51 – Проект мережі

2. Розробити узагальнену схему адресації пристроїв мережі. Для цього скористатися даними таблиць 5.50, 5.51. При виконанні розрахунків звернути увагу на те, що динамічне призначення параметрів *IP*-адресації буде застосовуватися на робочих станціях мереж **A** та **B** та інтерфейсі маршрутизатора **R-G-N-2**, через яких здійснено підключення до маршрутизатора **R-G-N-1**. На всіх інтерфейсах маршрутизатора **R-G-N-1**, комутаторах, точках доступу, серверах та мережних

принтерах параметри *IP*-адресації зазначаються статично. Результати навести у вигляді таблиці, яка аналогічна таблиці 5.48.

3. З врахуванням даних п. 2. провести розподіл *IP*-адрес. Дані розподілу навести у вигляді таблиці, яка аналогічна таблиці 5.49.

Таблиця 5.50 – Кількість пристроїв для побудови мережі

| № варіанта | Мережа А – Кількість |          |                    |               |               | Мережа В – Кількість |          |                    |               |               |
|------------|----------------------|----------|--------------------|---------------|---------------|----------------------|----------|--------------------|---------------|---------------|
|            | Робочих станцій      | Серверів | Мережних принтерів | Комунікаторів | Точок доступу | Робочих станцій      | Серверів | Мережних принтерів | Комунікаторів | Точок доступу |
| 1          | 50                   | 3        | 5                  | 2             | 1             | 40                   | 2        | 4                  | 2             | 2             |
| 2          | 65                   | 3        | 6                  | 2             | 2             | 20                   | 1        | 3                  | 2             | 1             |
| 3          | 13                   | 2        | 2                  | 3             | 1             | 50                   | 2        | 5                  | 3             | 2             |
| 4          | 122                  | 2        | 8                  | 3             | 2             | 22                   | 1        | 2                  | 3             | 1             |
| 5          | 50                   | 2        | 5                  | 2             | 1             | 40                   | 2        | 4                  | 3             | 2             |
| 6          | 240                  | 5        | 12                 | 2             | 2             | 30                   | 5        | 3                  | 3             | 1             |
| 7          | 140                  | 3        | 10                 | 3             | 1             | 20                   | 2        | 2                  | 2             | 2             |
| 8          | 220                  | 4        | 15                 | 3             | 2             | 58                   | 3        | 5                  | 2             | 1             |
| 9          | 144                  | 3        | 13                 | 2             | 0             | 140                  | 4        | 14                 | 2             | 3             |
| 10         | 20                   | 2        | 2                  | 2             | 3             | 120                  | 3        | 12                 | 2             | 0             |
| 11         | 51                   | 3        | 3                  | 3             | 0             | 80                   | 3        | 8                  | 3             | 3             |
| 12         | 192                  | 4        | 18                 | 3             | 3             | 14                   | 1        | 2                  | 3             | 0             |
| 13         | 232                  | 5        | 16                 | 2             | 0             | 80                   | 3        | 10                 | 2             | 2             |
| 14         | 100                  | 3        | 10                 | 2             | 0             | 50                   | 2        | 10                 | 2             | 1             |
| 15         | 200                  | 4        | 20                 | 2             | 2             | 100                  | 4        | 15                 | 2             | 0             |
| 16         | 26                   | 1        | 4                  | 2             | 1             | 14                   | 1        | 2                  | 2             | 0             |
| 17         | 24                   | 3        | 6                  | 3             | 1             | 30                   | 2        | 5                  | 3             | 2             |
| 18         | 28                   | 2        | 8                  | 3             | 2             | 62                   | 3        | 6                  | 3             | 1             |
| 19         | 62                   | 2        | 8                  | 2             | 0             | 126                  | 3        | 13                 | 2             | 3             |
| 20         | 126                  | 3        | 15                 | 2             | 3             | 255                  | 5        | 25                 | 2             | 0             |
| 21         | 50                   | 2        | 5                  | 2             | 0             | 70                   | 2        | 7                  | 2             | 2             |
| 22         | 140                  | 5        | 12                 | 2             | 0             | 28                   | 1        | 3                  | 2             | 1             |
| 23         | 200                  | 5        | 28                 | 2             | 1             | 50                   | 2        | 5                  | 2             | 2             |
| 24         | 50                   | 1        | 5                  | 2             | 2             | 100                  | 2        | 12                 | 2             | 1             |
| 25         | 40                   | 2        | 4                  | 2             | 1             | 80                   | 3        | 10                 | 3             | 2             |
| 26         | 90                   | 3        | 9                  | 2             | 2             | 110                  | 3        | 11                 | 3             | 1             |
| 27         | 120                  | 5        | 12                 | 2             | 0             | 240                  | 4        | 28                 | 2             | 3             |
| 28         | 50                   | 2        | 5                  | 2             | 3             | 120                  | 2        | 10                 | 2             | 0             |
| 29         | 240                  | 3        | 24                 | 3             | 1             | 150                  | 3        | 15                 | 3             | 2             |
| 30         | 42                   | 2        | 4                  | 3             | 2             | 24                   | 2        | 3                  | 3             | 1             |
| 31         | 50                   | 3        | 5                  | 2             | 1             | 22                   | 1        | 2                  | 3             | 1             |
| 32         | 65                   | 3        | 6                  | 2             | 2             | 40                   | 2        | 4                  | 3             | 2             |
| 33         | 13                   | 2        | 2                  | 3             | 1             | 30                   | 5        | 3                  | 3             | 1             |

Таблиця 5.51 – IP-адреси підмереж та параметри налагодження DHCP-пулів

| № з/п | IP-адреси мереж |           |              | Перевірочний інтервал |             | Час оренди |        |         |
|-------|-----------------|-----------|--------------|-----------------------|-------------|------------|--------|---------|
|       | Мережа А        | Мережа В  | Мережа С     | Кількість спроб       | Таймаут, мс | Дні        | Години | Хвилини |
| 1     | 191.G.N.0       | 192.G.N.0 | 193.G.N.N/30 | 3                     | 350         | 0          | 6      | 30      |
| 2     | 192.G.N.0       | 193.G.N.0 | 194.G.N.N/30 | 4                     | 450         | 0          | 8      | 30      |
| 3     | 193.G.N.0       | 194.G.N.0 | 195.G.N.N/30 | 5                     | 550         | 0          | 10     | 30      |
| 4     | 194.G.N.0       | 195.G.N.0 | 196.G.N.N/30 | 5                     | 550         | 0          | 12     | 30      |
| 5     | 195.G.N.0       | 196.G.N.0 | 197.G.N.N/30 | 4                     | 450         | 0          | 14     | 30      |
| 6     | 196.G.N.0       | 197.G.N.0 | 198.G.N.N/30 | 3                     | 600         | 0          | 16     | 30      |
| 7     | 197.G.N.0       | 198.G.N.0 | 199.G.N.N/30 | 3                     | 600         | 0          | 6      | 30      |
| 8     | 198.G.N.0       | 199.G.N.0 | 200.G.N.N/30 | 5                     | 550         | 0          | 12     | 30      |
| 9     | 199.G.N.0       | 200.G.N.0 | 201.G.N.N/30 | 4                     | 450         | 0          | 16     | 30      |
| 10    | 200.G.N.0       | 201.G.N.0 | 202.G.N.N/30 | 4                     | 450         | 0          | 18     | 30      |
| 11    | 201.G.N.0       | 202.G.N.0 | 203.G.N.N/30 | 5                     | 550         | 1          | 4      | 0       |
| 12    | 202.G.N.0       | 203.G.N.0 | 204.G.N.N/30 | 3                     | 350         | 1          | 8      | 0       |
| 13    | 203.G.N.0       | 204.G.N.0 | 205.G.N.N/30 | 3                     | 350         | 1          | 12     | 0       |
| 14    | 204.G.N.0       | 205.G.N.0 | 206.G.N.N/30 | 3                     | 600         | 1          | 16     | 0       |
| 15    | 205.G.N.0       | 206.G.N.0 | 207.G.N.N/30 | 4                     | 600         | 1          | 20     | 0       |
| 16    | 206.G.N.0       | 207.G.N.0 | 208.G.N.N/30 | 4                     | 700         | 1          | 20     | 30      |
| 17    | 207.G.N.0       | 208.G.N.0 | 209.G.N.N/30 | 5                     | 700         | 1          | 16     | 30      |
| 18    | 208.G.N.0       | 209.G.N.0 | 210.G.N.N/30 | 5                     | 450         | 1          | 12     | 30      |
| 19    | 209.G.N.0       | 210.G.N.0 | 211.G.N.N/30 | 5                     | 350         | 1          | 8      | 30      |
| 20    | 210.G.N.0       | 211.G.N.0 | 212.G.N.N/30 | 5                     | 550         | 1          | 4      | 30      |
| 21    | 211.G.N.0       | 212.G.N.0 | 213.G.N.N/30 | 4                     | 450         | 0          | 4      | 0       |
| 22    | 212.G.N.0       | 213.G.N.0 | 214.G.N.N/30 | 4                     | 350         | 0          | 8      | 0       |
| 23    | 213.G.N.0       | 214.G.N.0 | 215.G.N.N/30 | 3                     | 600         | 0          | 12     | 0       |
| 24    | 214.G.N.0       | 215.G.N.0 | 216.G.N.N/30 | 3                     | 650         | 0          | 16     | 0       |
| 25    | 215.G.N.0       | 216.G.N.0 | 217.G.N.N/30 | 3                     | 250         | 0          | 20     | 0       |
| 26    | 216.G.N.0       | 217.G.N.0 | 218.G.N.N/30 | 4                     | 350         | 0          | 20     | 0       |
| 27    | 217.G.N.0       | 218.G.N.0 | 219.G.N.N/30 | 4                     | 450         | 0          | 16     | 0       |
| 28    | 218.G.N.0       | 219.G.N.0 | 220.G.N.N/30 | 3                     | 350         | 0          | 12     | 0       |
| 29    | 219.G.N.0       | 220.G.N.0 | 221.G.N.N/30 | 5                     | 550         | 0          | 8      | 0       |
| 30    | 220.G.N.0       | 221.G.N.0 | 222.G.N.N/30 | 5                     | 600         | 0          | 4      | 0       |
| 31    | 221.G.N.0       | 222.G.N.0 | 223.G.N.0/30 | 3                     | 350         | 0          | 6      | 30      |
| 32    | 222.G.N.0       | 223.G.N.0 | 191.G.N.0/30 | 4                     | 450         | 0          | 8      | 30      |
| 33    | 223.G.N.0       | 191.G.N.0 | 192.G.N.0/30 | 5                     | 550         | 0          | 10     | 30      |

4. Провести базове налагодження пристроїв, інтерфейсів та каналів зв'язку побудованої мережі. При налагодженні пристроїв безпроводних сегментів локальної мережі **А** використовувати унікальні ідентифікатори (*SSID*) вигляду **SSID-А-Г-Н-Х**, локальної мережі **В** – вигляду **SSID-В-Г-Н-Х**. Для пристроїв мережі, що використовують статичне призначення, виконати налагодження параметрів

*IP*-адресації відповідно до даних, які отримані у п. 2, 3. Перевірити наявність зв'язку між сусідніми парами пристроїв.

5. Провести налагодження функціонування *DHCP*-сервера на маршрутизаторі *R-G-N-1* з урахуванням даних розрахунку п. 2, 3 та даних таблиці 5.51. Налаштування перевірконого інтервалу та часу оренди виконати за можливості (якщо відповідні команди підтримуються симулятором/емулятором).

6. Провести налагодження функціонування маршрутизатора **R-G-N-2** та робочих станцій як *DHCP*-клієнтів.

7. Дослідити особливості отримання службової та діагностичної інформації протоколу *DHCP* за допомогою відповідних команд.

8. Дослідити процеси передачі даних між *DHCP*-клієнтами та *DHCP*-сервером. У разі появи конфліктів визначити та усунути їх джерела.

### КОНТРОЛЬНІ ПИТАННЯ

1. Наведіть перелік параметрів *IP*-адресації вузла.
2. Які ви знаєте технології та протоколи динамічного призначення параметрів *IP*-адресації.
3. Поясніть роботу технології *APIPA*.
4. Надайте загальну характеристику протоколу *DHCP*.
5. Опишіть ролі пристроїв у протоколі *DHCP*.
6. Наведіть характеристики протоколу *DHCP* стосовно моделі *OSI* та стеку *TCP/IP*.
7. Наведіть основні параметри адресації, які надаються мережному вузлу за протоколом *DHCP*.
8. Наведіть перелік та призначення способів призначення параметрів *IP*-адресації у протоколі *DHCP*.
9. Наведіть перелік та призначення основних повідомлень протоколу *DHCP*.
10. Які механізми (елементи) захисту мережі може надати протокол *DHCP*?
11. Наведіть перелік та призначення основних команд для налагодження *DHCP*-сервера на маршрутизаторі *Cisco* при умові динамічного призначення адрес.
12. Наведіть перелік та призначення основних команд для налагодження *DHCP*-сервера на маршрутизаторі *Cisco* при умові статичного призначення адрес.
13. Наведіть перелік та призначення основних команд моніторингу роботи протоколу *DHCP* на маршрутизаторі *Cisco*.
14. Визначте застосування протоколу *DHCP* у сучасних мережних ОС.
15. Які є методи діагностики та керування процесом роботи *DHCP*-клієнта в ОС *Windows* та ОС *Unix/Linux*.

## 5.2.5 НАЛАШТУВАННЯ РЕЗЕРВНИХ МАРШРУТИЗАТОРІВ *HSRP*

### ОСНОВНІ КОМАНДИ НАЛАШТУВАННЯ РЕЗЕРВНИХ МАРШРУТИЗАТОРІВ *HSRP*

*HSRP* налаштовується однаково на маршрутизаторах і комутаторах 3 рівня, налаштування виконуються на інтерфейсах 3 рівня. На маршрутизаторі це можуть бути фізичні інтерфейси або підінтерфейси, а на комутаторі 3 рівня інтерфейси, переведені в режим роботи на 3 рівень або *SVI*-інтерфейси (*interface vlan*).

Основні команди налаштування резервних маршрутизаторів є команди **standby** та **standby version**.

Команда **standby version** необов'язкова і якщо не ввести цю команду або не вказати ключове слово, інтерфейс запустить версію *HSRP* за замовчуванням, тобто версію 1

**standby version** {1 | 2}

Команда для створення (або включення) групи *HSRP*, використовуючи її номер та віртуальну *IP*-адресу:

**standby** [*group-number*] ip [*ip-address* [*secondary*]]

де **group-number** – (необов'язково) номер групи на інтерфейсі, для якої вмикається *HSRP*. Діапазон від 0 до 255; значення за замовчуванням – 0. Якщо є лише одна група *HSRP*, номер групи вводити не потрібно.

**ip-address** – (необов'язково для всіх інтерфейсів, крім одного) віртуальна *IP*-адреса інтерфейсу маршрутизатора гарячого резерву. Ви повинні ввести віртуальну *IP*-адресу принаймні для одного з інтерфейсів; її можна дізнатися на інших інтерфейсах.

**secondary** – (необов'язково) *IP*-адреса вторинного інтерфейсу маршрутизатора гарячого резерву. Якщо жоден з маршрутизаторів не визначено як вторинний або резервний і не встановлено жодних пріоритетів, основні *IP*-адреси порівнюються, і вища *IP*-адреса стає активним маршрутизатором, а наступна за значенням – резервним маршрутизатором.

Значення таймерів для всіх маршрутизаторів в одній групі має бути однакове. Значення таймерів маршрутизатор може отримати від **active** маршрутизатора. Команда зміни таймерів *HSRP* на інтерфейсі:

**standby** [*group-number*] timers <hellotime> <holdtime>

де **hellotime** – за замовчуванням 3 секунди, діапазон значень від 1 до 255;

**holdtime** – за замовчуванням 10 секунд, діапазон значень від 1 до 255.

Активний маршрутизатор для групи *HSRP*. Це маршрутизатор, який буде використовуватися в якості шлюзу, поки не вийде з ладу або шлях до нього стане неактивним чи непридатним для використання. Значення за замовчуванням пріоритету інтерфейсу маршрутизатора – 100. Більш високе значення визначатиме,

який маршрутизатор є активним. Якщо пріоритети маршрутизаторів у групі *HSRP* однакові, то активним стане маршрутизатор з найбільшою *IP*-адресою. Команда налаштування пріоритету маршрутизатора:

```
standby [group-number] priority <priority>
```

Режим *preempt* дає змогу маршрутизатору з вищим пріоритетом перехоплювати роль *active* маршрутизатора. За замовчуванням режим *preempt* вимкнений. Команда **preempt**:

```
standby [group-number] preempt [delay <minimum <delay> |
reload <delay> | sync <delay>>]
```

де **delay** – дає змогу вказати затримку перехоплення ролі маршрутизатора. Під час увімкнення у маршрутизатора ще не заповнена таблиця маршрутизації. Якщо він одразу перехопить роль *active* маршрутизатора, то, можливо, він не зможе передавати трафік деякий час. Затримка дає змогу маршрутизатору заповнити таблицю маршрутизації. За замовчуванням значення параметра 0, тобто, маршрутизатор перехоплює роль негайно.

Для того щоб вимкнути *HSRP* для вказаного інтерфейсу є команда:

```
no standby [group-number] ip [ip-address]
```

Для прямих пакетів, адресованих у мережі, які не описані явним чином у таблиці маршрутизації може виникнути потреба вказати стандартний маршрут. Наприклад, окремий випадок, переадресація пакетів із локальної мережі в *Internet*. Створення статичного маршруту до мережі *0.0.0.0 0.0 0.0.0.0* є одним із способів визначення шлюзу останньої черги:

```
ip route 0.0.0.0 0.0.0.0 ipaddress
```

де **ipaddress** – *IP*-адреса шлюзу останньої черги.

Перевірка станів резервних маршрутів проводиться за допомогою команд: **show standby**, **show standby brief**.

### Модельний приклад налагодження резервних маршрутизаторів *HSRP*

Розглянемо специфіку налагодження резервних маршрутизаторів *Cisco*, а саме створення груп і призначення віртуальних *IP*-адрес на маршрутизаторах, увімкнення *HSRP*, схема якої наведена на рисунку 5.52.

В *HSRP* існує велика кількість налаштувань, які можуть бути виконані після увімкнення *HSRP* на інтерфейсі. Однак, рекомендується спочатку виконати всі налаштування (наприклад, аутентифікація, таймери), а потім вмикати *HSRP* на інтерфейсі.

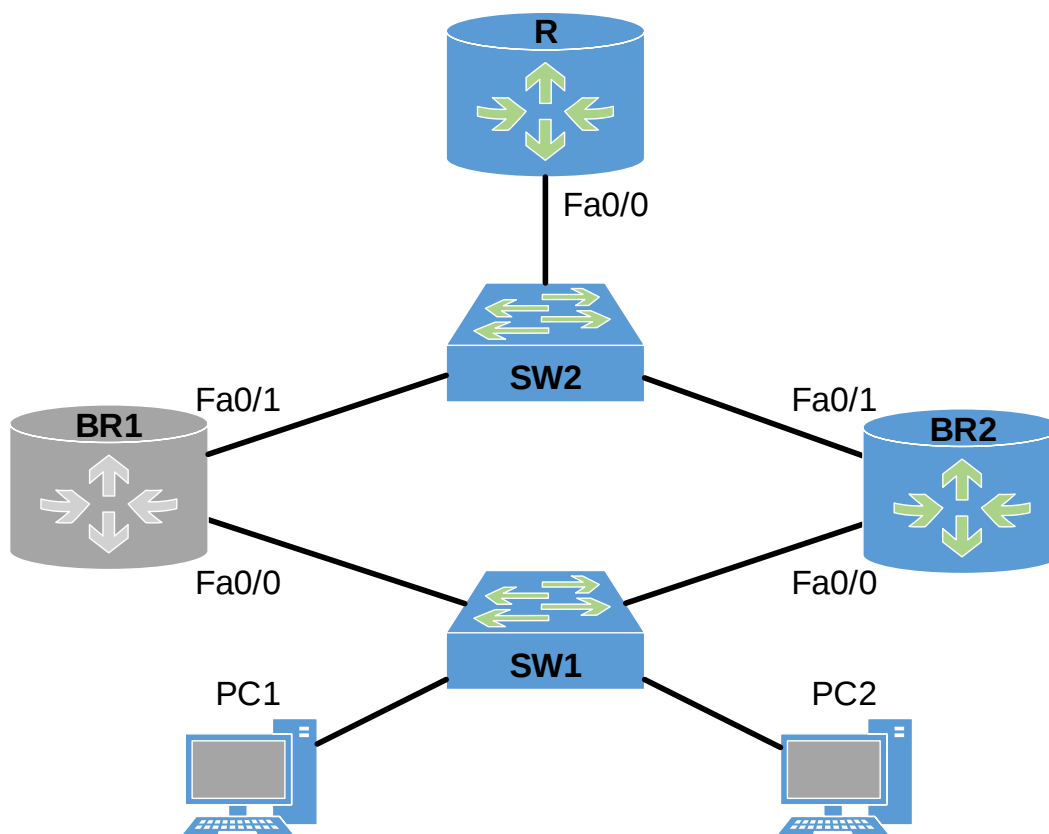


Рисунок 5.52 – Приклад мережі

Для налагодження параметрів адресації інтерфейсів пристроїв використано дані таблиці 5.52.

Таблиця 5.52 – Параметри адресації мережі

| Пристрій                    | Інтерфейс       | IP-адреса/<br>Префікс | Маска підме-<br>режі | Шлюз за замо-<br>вчуванням |
|-----------------------------|-----------------|-----------------------|----------------------|----------------------------|
| <b>BR1</b>                  | Loopback0       | 1.1.1.1/8             | 255.0.0.0            | n/a                        |
|                             | FastEthernet0/0 | 192.168.1.1/24        | 255.255.255.0        | n/a                        |
|                             | FastEthernet0/1 | 192.168.2.1/24        | 255.255.255.0        | n/a                        |
| <b>BR2</b>                  | Loopback0       | 2.2.2.2/8             | 255.0.0.0            | n/a                        |
|                             | FastEthernet0/0 | 192.168.1.2/24        | 255.255.255.0        | n/a                        |
|                             | FastEthernet0/1 | 192.168.2.2/24        | 255.255.255.0        | n/a                        |
| <b>R</b>                    | FastEthernet0/0 | 192.168.2.3/24        | 255.255.255.0        | n/a                        |
| <b>PC1</b>                  | NIC             | 192.168.1.101/24      | 255.255.255.0        | 192.168.1.11               |
| <b>PC2</b>                  | NIC             | 192.168.1.102/24      | 255.255.255.0        | 192.168.1.33               |
| <b>HSRP Virtual Gateway</b> | Virtual 1       | 192.168.1.11          | -                    | n/a                        |
| <b>HSRP Virtual Gateway</b> | Virtual 3       | 192.168.1.33          | -                    | n/a                        |

Сценарії налагодження параметрів резервних маршрутизаторів **BR1** та **BR2** наведені нижче.

### Налаштування HSRP на BR1:

```
BR1(config)#interface Loopback0
```

```
%LINK-5-CHANGED: Interface Loopback0, changed state to up
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
```

```
BR1(config-if)#ip address 1.1.1.1 255.0.0.0
```

```
BR1(config-if)#exit
```

```
BR1(config)#interface Fa0/0
```

```
BR1(config-if)#ip address 192.168.1.1 255.255.255.0
```

*Вибір номеру версії протоколу HSRP*

```
BR1(config-if)#standby version 2
```

#### Екземпляр HSRP - 1

*Налаштування IP-адреси віртуального шлюзу за замовчуванням*

```
BR1(config-if)#standby 1 ip 192.168.1.11
```

```
%HSRP-6-STATECHANGE: FastEthernet0/0 Grp 1 state Init -> Init
```

*Зміна таймерів HSRP на інтерфейсі*

```
BR1(config-if)#standby 1 timers 5 15
```

```
%HSRP-6-STATECHANGE: FastEthernet0/0 Grp 1 state Speak -> Standby
```

```
%HSRP-6-STATECHANGE: FastEthernet0/0 Grp 1 state Standby -> Active
```

*Налаштування пріоритету*

```
BR1(config-if)#standby 1 priority 101
```

*Включення режиму переривання і відновлення активного стану резервного маршрутизатора*

```
BR1(config-if)#standby 1 preempt
```

#### Екземпляр HSRP - 3

```
BR1(config-if)#standby 3 ip 192.168.1.33
```

```
BR1(config-if)#standby 3 timers 5 15
```

```
BR1(config-if)#standby 3 preempt
```

```
BR1(config-if)#no shutdown
```

```
BR1(config-if)#exit
```

```
BR1(config)#interface Fa0/1
```

```
BR1(config-if)#ip address 192.168.2.1 255.255.255.0
```

```
BR1(config-if)#no shutdown
```

```
BR1(config-if)#exit
```

*Додавання статичного маршруту для просування невідомих пакетів в іншу мережу*

```
BR1(config)#ip route 0.0.0.0 0.0.0.0 192.168.2.3
```

### Налаштування HSRP на BR2:

```
BR2(config)#interface Loopback0
```

```
BR2(config-if)#ip address 2.2.2.2 255.0.0.0
```

```
BR2(config-if)#exit
```

```
BR2(config)#interface Fa0/0
```

```
BR2(config-if)#ip address 192.168.1.2 255.255.255.0
```

```
BR2(config-if)#standby version 2
```

```
BR2(config-if)#standby 1 ip 192.168.1.11
```

```
BR2(config-if)#standby 1 timers 5 15
```

```
BR2(config-if)#standby 3 ip 192.168.1.33
```

```
BR2(config-if)#standby 3 timers 5 15
```

```
BR2(config-if)#standby 1 priority 101
```

```
BR2(config-if)#no shutdown
```

```
BR2(config-if)#exit
```

```
BR2(config)#interface Fa0/1
BR2(config-if)#ip address 192.168.2.2 255.255.255.0
BR2(config-if)#no shutdown
BR2(config-if)#exit
```

```
BR2(config)#ip route 0.0.0.0 0.0.0.0 192.168.2.3
```

### Налаштування IP на R:

```
R(config)#interface fa0/0
R(config-if)#ip address 192.168.2.3 255.255.255.0
R(config-if)#no shutdown
R(config-if)#exit

R(config)#ip route 192.168.1.0 255.255.255.0 192.168.2.1
R(config)#ip route 192.168.1.0 255.255.255.0 192.168.2.2
```

## Результати виконання команд моніторингу та діагностики для розглянутого прикладу

З метою перегляду інформації про функціонування резервних маршрутів для розглянутого прикладу використано команди **show standby**, **show standby brief**, **tracert**:

Перегляд короткої інформації про групи на **BR1**:

```
BR1#show standby brief
 P indicates configured to preempt.
 |
Interface Grp Pri P State Active Standby Virtual IP
Fa0/0 1 101 P Standby 192.168.1.2 local 192.168.1.11
Fa0/0 3 100 P Standby 192.168.1.2 local 192.168.1.33
```

Перегляд короткої інформації про групи на **BR2**:

```
BR2#show standby brief
 P indicates configured to preempt.
 |
Interface Grp Pri P State Active Standby Virtual IP
Fa0/0 1 101 Active local 192.168.1.1 192.168.1.11
Fa0/0 3 100 Active local 192.168.1.1 192.168.1.33
```

Перегляд інформації про всі ввімкнені групи **HSRP** на **BR1**:

```
BR1#show standby
FastEthernet0/0 - Group 1 (version 2)
 State is Standby
 10 state changes, last state change 00:05:48
 Virtual IP address is 192.168.1.11
 Active virtual MAC address is 0000.0C9F.F001
 Local virtual MAC address is 0000.0C9F.F001 (v2 default)
 Hello time 5 sec, hold time 15 sec
 Next hello sent in 1.12 secs
 Preemption enabled
 Active router is 192.168.1.2
 Standby router is local
 Priority 101 (configured 101)
 Group name is hsrp-Fa0/0-1 (default)
FastEthernet0/0 - Group 3 (version 2)
 State is Standby
 9 state changes, last state change 00:05:49
```

```
Virtual IP address is 192.168.1.33
Active virtual MAC address is 0000.0C9F.F003
 Local virtual MAC address is 0000.0C9F.F003 (v2 default)
Hello time 5 sec, hold time 15 sec
 Next hello sent in 1.415 secs
Preemption enabled
Active router is 192.168.1.2
Standby router is local
Priority 100 (default 100)
Group name is hsrp-Fa0/0-3 (default)
```

Перегляд інформації про всі ввімкнені групи *HSRP* на **BR2**:

```
BR2#show standby
FastEthernet0/0 - Group 1 (version 2)
 State is Active
 13 state changes, last state change 00:05:30
 Virtual IP address is 192.168.1.11
 Active virtual MAC address is 0000.0C9F.F001
 Local virtual MAC address is 0000.0C9F.F001 (v2 default)
 Hello time 5 sec, hold time 15 sec
 Next hello sent in 1.148 secs
 Preemption disabled
 Active router is local
 Standby router is 192.168.1.1, priority 101 (expires in 9 sec)
 Priority 101 (configured 101)
 Group name is hsrp-Fa0/0-1 (default)
FastEthernet0/0 - Group 3 (version 2)
 State is Active
 12 state changes, last state change 00:05:30
 Virtual IP address is 192.168.1.33
 Active virtual MAC address is 0000.0C9F.F003
 Local virtual MAC address is 0000.0C9F.F003 (v2 default)
 Hello time 5 sec, hold time 15 sec
 Next hello sent in 2.717 secs
 Preemption disabled
 Active router is local
 Standby router is 192.168.1.1
 Priority 100 (default 100)
 Group name is hsrp-Fa0/0-3 (default)
```

Перевірка маршруту від **PC1** до **R**:

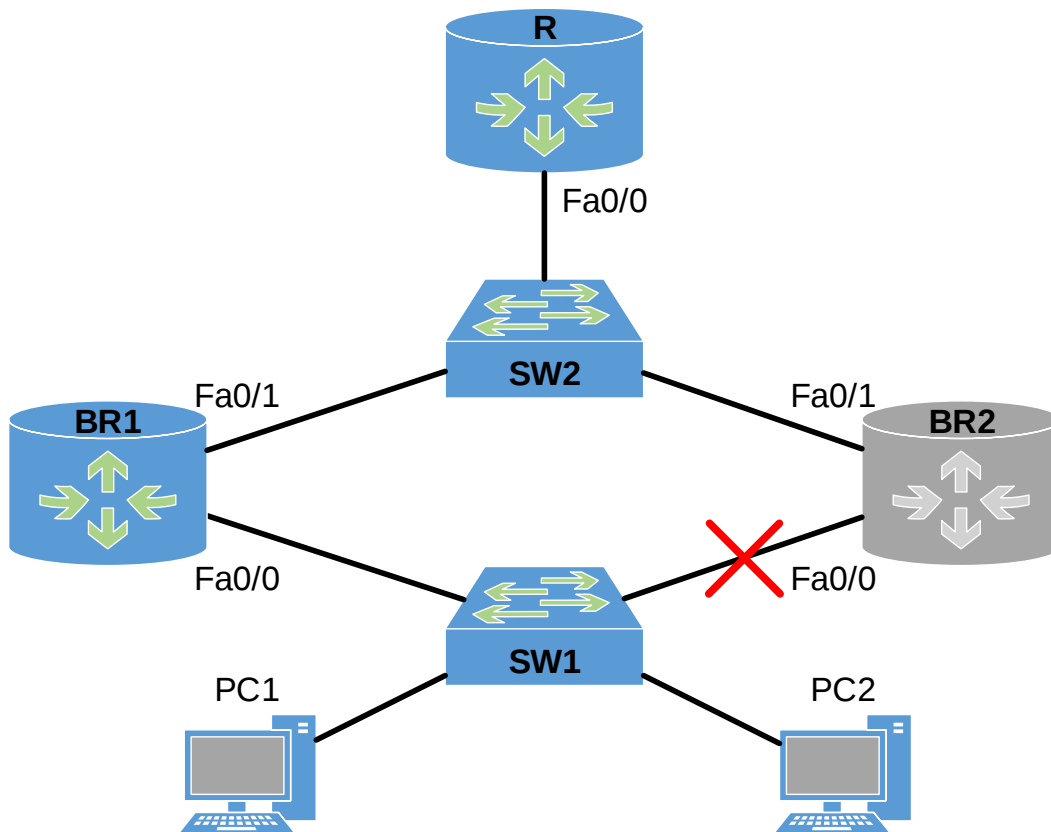
```
C:\>tracert 192.168.2.3
Tracing route to 192.168.2.3 over a maximum of 30 hops:
 1 0 ms 0 ms 0 ms 192.168.1.2
 2 0 ms 0 ms 0 ms 192.168.2.3
Trace complete.
```

Перевірка маршруту від **PC2** до **R**:

```
C:\>tracert 192.168.2.3
Tracing route to 192.168.2.3 over a maximum of 30 hops:
 1 0 ms 0 ms 0 ms 192.168.1.1
 2 0 ms 0 ms 0 ms 192.168.2.3
Trace complete.
```

## Результати виконання команд моніторингу та діагностики для модифікованого прикладу

З метою чистоти експерименту в прикладі наведеному на рисунку 5.53 виведено з ладу активний маршрут резервного зв'язку і повторно зроблено перегляд інформації про функціонування резервних маршрутів.



**Рисунок 5.53 – Приклад мережі модифікований**

Перегляд короткої інформації про групи на BR1:

```
BR1#show standby brief
```

| P indicates configured to preempt. |     |     |   |        |        |         |              |
|------------------------------------|-----|-----|---|--------|--------|---------|--------------|
|                                    |     |     |   |        |        |         |              |
| Interface                          | Grp | Pri | P | State  | Active | Standby | Virtual IP   |
| Fa0/0                              | 1   | 101 | P | Active | local  | unknown | 192.168.1.11 |
| Fa0/0                              | 3   | 100 | P | Active | local  | unknown | 192.168.1.33 |

Перегляд короткої інформації про групи на BR2:

```
BR2#show standby brief
```

| P indicates configured to preempt. |     |     |   |       |         |         |              |
|------------------------------------|-----|-----|---|-------|---------|---------|--------------|
|                                    |     |     |   |       |         |         |              |
| Interface                          | Grp | Pri | P | State | Active  | Standby | Virtual IP   |
| Fa0/0                              | 1   | 101 |   | Init  | unknown | unknown | 192.168.1.11 |
| Fa0/0                              | 3   | 100 |   | Init  | unknown | unknown | 192.168.1.33 |

Перегляд інформації про всі ввімкнені групи HSRP на BR1:

```
BR1#show standby
```

```
FastEthernet0/0 - Group 1 (version 2)
```

```
State is Active
```

```
11 state changes, last state change 00:33:35
```

```

Virtual IP address is 192.168.1.11
Active virtual MAC address is 0000.0C9F.F001
 Local virtual MAC address is 0000.0C9F.F001 (v2 default)
Hello time 5 sec, hold time 15 sec
 Next hello sent in 3.005 secs
Preemption enabled
Active router is local
Standby router is unknown
Priority 101 (configured 101)
Group name is hsrp-Fa0/0-1 (default)
FastEthernet0/0 - Group 3 (version 2)
 State is Active
 10 state changes, last state change 00:33:37
Virtual IP address is 192.168.1.33
Active virtual MAC address is 0000.0C9F.F003
 Local virtual MAC address is 0000.0C9F.F003 (v2 default)
Hello time 5 sec, hold time 15 sec
 Next hello sent in 2.19 secs
Preemption enabled
Active router is local
Standby router is unknown
Priority 100 (default 100)
Group name is hsrp-Fa0/0-3 (default)

```

Перегляд інформації про всі ввімкнені групи *HSRP* на **BR2**:

```

BR2#show standby
FastEthernet0/0 - Group 1 (version 2)
 State is Init (interface down)
Virtual IP address is 192.168.1.11
Active virtual MAC address is 0000.0C9F.F001
 Local virtual MAC address is 0000.0C9F.F001 (v2 default)
Hello time 5 sec, hold time 15 sec
 Next hello sent in 1.180 secs
Preemption disabled
Active router is unknown
Standby router is unknown
Priority 101 (configured 101)
Group name is hsrp-Fa0/0-1 (default)
FastEthernet0/0 - Group 3 (version 2)
 State is Init (interface down)
Virtual IP address is 192.168.1.33
Active virtual MAC address is 0000.0C9F.F003
 Local virtual MAC address is 0000.0C9F.F003 (v2 default)
Hello time 5 sec, hold time 15 sec
 Next hello sent in 1.180 secs
Preemption disabled
Active router is unknown
Standby router is unknown
Priority 100 (default 100)
Group name is hsrp-Fa0/0-3 (default)

```

Перевірка маршруту від **PC1** до **R**:

```

C:\>tracert 192.168.2.3
Tracing route to 192.168.2.3 over a maximum of 30 hops:
 1 0 ms 0 ms 0 ms 192.168.1.1
 2 0 ms 0 ms 0 ms 192.168.2.3
Trace complete.

```

Перевірка маршруту від **PC2** до **R**:

```
C:\>tracert 192.168.2.3
```

Tracing route to 192.168.2.3 over a maximum of 30 hops:

|   |      |      |      |             |
|---|------|------|------|-------------|
| 1 | 0 ms | 0 ms | 0 ms | 192.168.1.1 |
| 2 | 0 ms | 0 ms | 0 ms | 192.168.2.3 |

Trace complete.

## ПРАКТИЧНЕ ЗАВДАННЯ

**Мета роботи:** ознайомитися з технологіями, які дозволяють налаштувати протокол маршрутизатора гарячого резервування (*HSRP, Hot Standby Router Protocol*), щоб передбачити резервні (надлишкові) шляхи за замовчуванням для вузлів у локальних мережах за допомогою комутаційного обладнання *Cisco*.

### Порядок виконання роботи

1. Створити фізичний проєкт мережі або у середовищі програмного симулятора/емулятора створити мережу (рисунок 5.54). При побудові звернути увагу на вибір моделей комутаторів, мережних модулів та плат, а також мережних з'єднань. При побудові підмережі слід вибирати потрібний тип кабелю для відповідної технології.

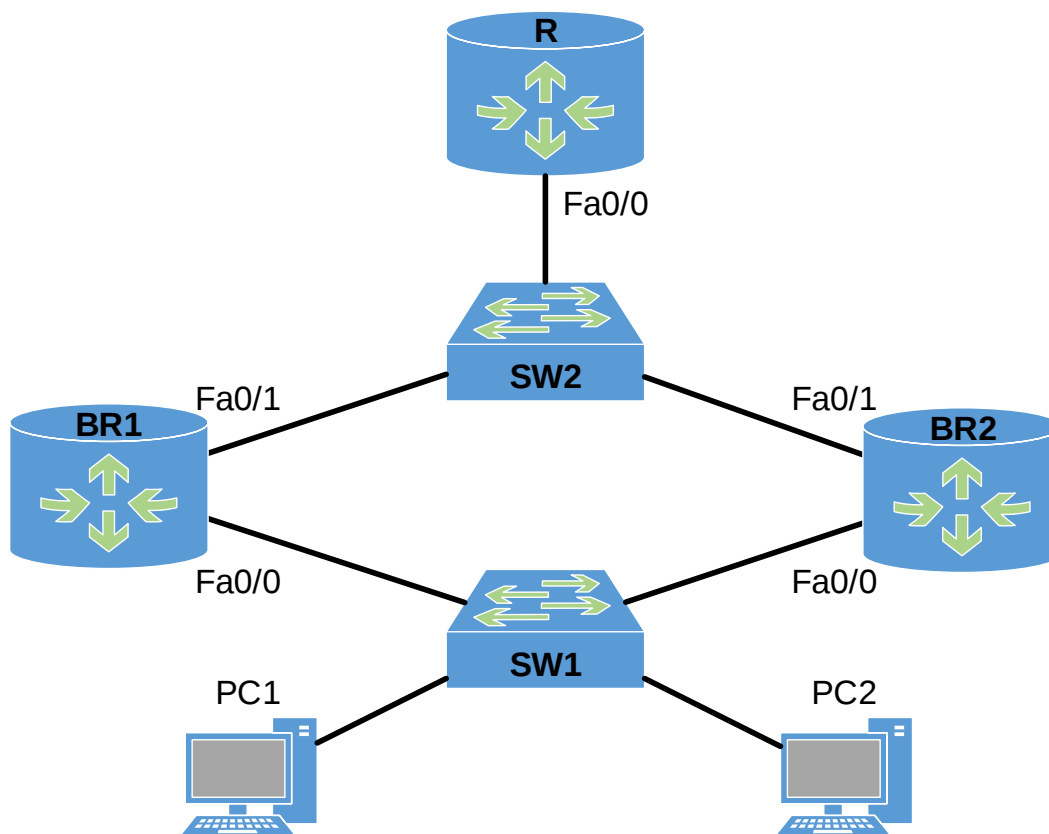


Рисунок 5.54 – Проєкт мережі

2. Провести налаштування маршрутизаторів, мережних інтерфейсів, з'єднань, створити налаштування резервних маршрутів на маршрутизаторах. Параметри резервних маршрутів та *IP*-адресації відповідно до варіанту наведені в таблиці 5.53.

Таблиця 5.53 – Параметри резервних маршрутів та IP-адресації

| № варіанта | Локальна мережа | Зовнішня мережа | Резервний маршрутизатор | Значення таймерів, <hellotime> / <holdtime> |
|------------|-----------------|-----------------|-------------------------|---------------------------------------------|
| 1          | 192.168.1.0/24  | 172.25.30.0/28  | BR1                     | 5 / 15                                      |
| 2          | 192.168.2.0/24  | 172.25.29.0/28  | BR2                     | 3 / 10                                      |
| 3          | 192.168.3.0/24  | 172.25.28.0/28  | BR1                     | 10 / 20                                     |
| 4          | 192.168.4.0/24  | 172.25.27.0/28  | BR2                     | 4 / 12                                      |
| 5          | 192.168.5.0/24  | 172.25.26.0/28  | BR1                     | 8 / 25                                      |
| 6          | 192.168.6.0/24  | 172.25.25.0/28  | BR2                     | 5 / 15                                      |
| 7          | 192.168.7.0/24  | 172.25.24.0/28  | BR1                     | 3 / 10                                      |
| 8          | 192.168.8.0/24  | 172.25.23.0/28  | BR2                     | 10 / 20                                     |
| 9          | 192.168.9.0/24  | 172.25.22.0/28  | BR1                     | 4 / 12                                      |
| 10         | 192.168.10.0/24 | 172.25.21.0/28  | BR2                     | 8 / 25                                      |
| 11         | 192.168.11.0/24 | 172.25.20.0/28  | BR1                     | 5 / 15                                      |
| 12         | 192.168.12.0/24 | 172.25.19.0/28  | BR2                     | 3 / 10                                      |
| 13         | 192.168.13.0/24 | 172.25.18.0/28  | BR1                     | 10 / 20                                     |
| 14         | 192.168.14.0/24 | 172.25.17.0/28  | BR2                     | 4 / 12                                      |
| 15         | 192.168.15.0/24 | 172.25.16.0/28  | BR1                     | 8 / 25                                      |
| 16         | 192.168.16.0/24 | 172.25.15.0/28  | BR2                     | 5 / 15                                      |
| 17         | 192.168.17.0/24 | 172.25.14.0/28  | BR1                     | 3 / 10                                      |
| 18         | 192.168.18.0/24 | 172.25.13.0/28  | BR2                     | 10 / 20                                     |
| 19         | 192.168.19.0/24 | 172.25.12.0/28  | BR1                     | 4 / 12                                      |
| 20         | 192.168.20.0/24 | 172.25.11.0/28  | BR2                     | 8 / 25                                      |
| 21         | 192.168.21.0/24 | 172.25.10.0/28  | BR1                     | 5 / 15                                      |
| 22         | 192.168.22.0/24 | 172.25.9.0/28   | BR2                     | 3 / 10                                      |
| 23         | 192.168.23.0/24 | 172.25.8.0/28   | BR1                     | 10 / 20                                     |
| 24         | 192.168.24.0/24 | 172.25.7.0/28   | BR2                     | 4 / 12                                      |
| 25         | 192.168.25.0/24 | 172.25.6.0/28   | BR1                     | 8 / 25                                      |
| 26         | 192.168.26.0/24 | 172.25.5.0/28   | BR2                     | 5 / 15                                      |
| 27         | 192.168.27.0/24 | 172.25.4.0/28   | BR1                     | 3 / 10                                      |
| 28         | 192.168.28.0/24 | 172.25.3.0/28   | BR2                     | 10 / 20                                     |
| 29         | 192.168.29.0/24 | 172.25.2.0/28   | BR1                     | 4 / 12                                      |
| 30         | 192.168.30.0/24 | 172.25.1.0/28   | BR2                     | 8 / 25                                      |

### КОНТРОЛЬНІ ПИТАННЯ

1. Що таке *HSRP* і як він використовується в мережах *Cisco*?
2. Як визначити основний та резервний маршрутизатори в *HSRP*?
3. Які критерії ви використовуєте для вибору *HSRP*-групи?
4. Як налаштувати *HSRP* на маршрутизаторі *Cisco*?
5. Як визначити пріоритет маршрутизатора в *HSRP*?
6. Які є можливі стани *HSRP*-інтерфейсів та їхні характеристики?
7. Як визначити час переходу (*Hello time*) та час утримання (*Hold time*) в *HSRP*?
8. Що таке *virtual IP (VIP)* адреса в *HSRP* і як вона працює?
9. Які протоколи використовуються для обміну інформацією між маршрутизаторами в *HSRP*?

## 5.2.6 НАЛАШТУВАННЯ БЕЗПЕКИ ЗАСОБАМИ КОМУТАТОРА

### ОСНОВНІ КОМАНДИ НАЛАШТУВАННЯ БЕЗПЕКИ ЗАСОБАМИ КОМУТАТОРА

При налаштуванні безпеки мережі, в залежності від її конфігурації та її параметрів, може знадобитись виконати певні налаштування з розширеного захисту на комутаторах. Кінцева реалізація повинна передбачати запровадження комплексних заходів безпеки відповідно до висунутих вимог. Наприклад це може бути: створення захищеної магістралі, вимкнення невикористаних портів комутатора, функція захисту портів *port security*, відстеження *DHCP*, налаштування *PortFast*, *BPDU Guard* і ще багато іншого.

Для запобігання атаці з виснаження ресурсів *DHCP* можна відслідковувати пакети *DHCP* (*DHCP snooping*) на надійних (*trusted*) портах. Увімкнути відстеження *DHCP* на певних портах або *VLAN* можна за допомогою команди **ip dhcp snooping trust**.

У випадку запобігання атакам переходів між *VLAN* треба вимкнути перемовини *DTP* на портах, заблокувати всі порти що не використовуються та *native VLAN* зробити відмінною від *VLAN 1*.

Для стримування несанкціонованих маніпуляцій з протоколом *STP* використовуйте засоби *PortFast* і *Bridge Protocol Data Unit (BPDU) Guard*.

Функція комутатора, що дає змогу вказати *MAC*-адреси вузлів, яким дозволено передавати дані через порт отримала назву *Port security*. Ця функція є базовою і використовується для запобігання:

- несанкціонованої зміни *MAC*-адреси мережевого пристрою або підключення до мережі,
- атак, спрямованих на переповнення таблиці комутації.

Але є певні обмеження на сумісність, так наприклад *Port security* несумісна з певними функціями комутатора:

- порт, на якому ввімкнений *DTP* (*switchport mode dynamic*),
- інтерфейс, переведений у режим третього рівня (*no switchport*),
- *SPAN destination port*,

Функція *Port security* на комутаторах *Cisco* підтримує такі типи безпеки *MAC*-адрес:

- Статичні *MAC*-адреси:
  - задаються статично командою **switchport port-security mac-address mac-address** у режимі налаштування інтерфейсу,
  - зберігаються в таблиці адрес,
  - додаються в поточну конфігурацію комутатора;
- Динамічні *MAC*-адреси:
  - динамічно вивчаються,
  - зберігаються тільки в таблиці адрес,
  - видаляються під час перезавантаження комутатора;

- *Sticky* MAC-адреси:
  - можуть бути статично налаштовані або динамічно вивчені,
  - зберігаються в таблиці адрес,
  - додаються в поточну конфігурацію комутатора. Якщо ці адреси збережені в конфігураційному файлі, після перезавантаження комутатора, їх не треба заново перелаштовувати.

Тригерами порушення безпеки для *Port security* будуть вважатися наступні ситуації:

- максимальну кількість безпечних MAC-адрес було додано в таблицю адрес і вузол, чия MAC-адреса не записана в таблиці адрес, намагається отримати доступ через інтерфейс,
- адреса, вивчена або налаштована як безпечна на одному інтерфейсі, з'явилася на іншому безпечному інтерфейсі в тій самій VLAN.

На інтерфейсі можуть бути налаштовані такі режими реагування на порушення безпеки:

*Port security* налаштовується в режимі налаштування інтерфейсу. На багатьох комутаторах Cisco за замовчуванням порт перебуває в режимі *dynamic auto*, однак цей режим не сумісний із функцією *port security*. Тому інтерфейс треба перевести в режим *trunk* або *access*:

```
switchport mode <access | trunk>
```

Увімкнення *Port security* на інтерфейсі:

```
switchport port-security
```

Після цього порт починає працювати в режимі захисту з налаштуваннями за замовчуванням:

- Запам'ятовування *sticky*-адрес – вимкнено.
- Максимальна кількість безпечних MAC-адрес на порту – **1**.
- Режим реагування на порушення – **shutdown**.
- Час зберігання адрес:
  - значення *aging time* – **0**,
  - для статичних адрес – вимкнено,
  - тип часу – абсолютний.

Оскільки, після команди **switchport port-security** одразу вмикається режим *Port security* з налаштуваннями за замовчуванням, то спочатку потрібно їх виправити (за потреби), а потім увімкнути функцію.

Задає максимальну кількість безпечних MAC-адрес для порту (за замовчуванням 1):

```
switchport port-security maximum number_of_addresses
```

Для повернення до конфігурації за замовчуванням:

```
no switchport port-security maximum
```

Увімкнення захисту порту за допомогою *sticky* (прив'язаних) MAC-адрес на порту:

```
switchport port-security mac-address sticky
```

Вимикає захист порту за допомогою *sticky* MAC-адрес на порту:

```
no switchport port-security mac-address sticky
```

Увімкнувши захист портів за допомогою *sticky* MAC-адрес, зверніть увагу на таку інформацію:

- Коли ви вводите команду **switchport port-security mac-address sticky**:
  - Усі динамічно отримані безпечні MAC-адреси на порту будуть перетворені на постійні безпечні MAC-адреси.
  - Статичні безпечні MAC-адреси не перетворюються на *sticky* MAC-адреси.
  - Безпечні MAC-адреси, що динамічно вивчаються в голосових VLAN, не перетворюються на незмінні MAC-адреси.
  - Нові безпечні MAC-адреси, отримані динамічно, є постійними.
- Коли ви вводите команду **no switchport port-security mac-address sticky**, усі захищені MAC-адреси на порту перетворюються на динамічні захищені MAC-адреси.
- Щоб зберегти динамічно запам'ятовувані *sticky* MAC-адреси і налаштувати їх на порту після завантаження або перезавантаження, після того, як динамічно запам'ятовувані *sticky* MAC-адреси буде запам'ятовано, необхідно ввести команду **write memory** або **copy running-config startup-config**, щоб зберегти їх у файлі **startup-config**.

Щоб налаштувати режим порушення безпеки порту, використайте наступну команду:

```
switchport port-security violation {protect | restrict | shutdown}
```

де **protect** – коли кількість безпечних MAC-адрес досягає максимального обмеження, пакети з невідомою MAC-адресою відправника відкидаються доти, доки не буде видалена достатня кількість безпечних MAC-адрес, щоб їхня кількість була меншою за максимальне значення, або збільшено максимальну кількість дозволених адрес.

**restrict** – Аналогічно режиму **protect**, але на відміну від попереднього у цьому режимі в разі порушення безпеки надсилається сповіщення в *syslog* і збільшується лічильник порушень (*violation counter*).

**shutdown** – порушення безпеки призводить до того, що інтерфейс переводиться в стан *error-disabled* і негайно вимикається, також вимикається *LED* порту. Надсилається повідомлення в *syslog* і збільшується лічильник порушень (*violation counter*). Коли порт у стані *error-disabled*, вивести з цього стану його можна тільки ввівши команду **errdisable recovery cause psecure-violation**, або вручну ввімкнути інтерфейс, ввівши в режимі налаштування інтерфейсу **shutdown** і **no shutdown**.

Для повернення до конфігурації за замовчуванням (вимкнення):

**no switchport port-security violation**

Якщо порт був налаштований (або залишений за замовчуванням) у режимі реагування *shutdown*, то в разі порушення порт перейде в стан *error-disabled*. Для перевірки стану порту можна скористатися командою **show interfaces <interface-number> status**.

Якщо порт заблокувався, згідно правил безпеки:

- Потрібно очистити таблицю MAC-адрес:

```
clear port-security [all|configured|dynamic|sticky]
[address <mac>|interface <int-id>]
```

- Підняти порт з *error-disabled*:

**no shutdown**

Для перегляду інформації з налаштувань *Port security* передбачені команди: **show port-security**, **show port-security interface <interface-number>**, **show port-security address**, **show interfaces <interface-number> status**.

### Модельний приклад налагодження безпеки портів на комутаторі Cisco

Розглянемо специфіку налагодження безпеки портів на комутаторі Cisco, схема наведена на рисунку 5.55.

Встановлюємо обмеження вивчення адрес – 1 адреса. Встановлюємо режим порушення *sticky*. В прикладі включаємо функцію захисту на інтерфейсах **Fa0/1** та **Fa0/2**. Всі інші порти вимикаємо. Після вивчення MAC-адреси блокуємо функцію вивчення нових адрес для інтерфейсу з метою відкидання пакетів з невивченими MAC-адресами джерела.

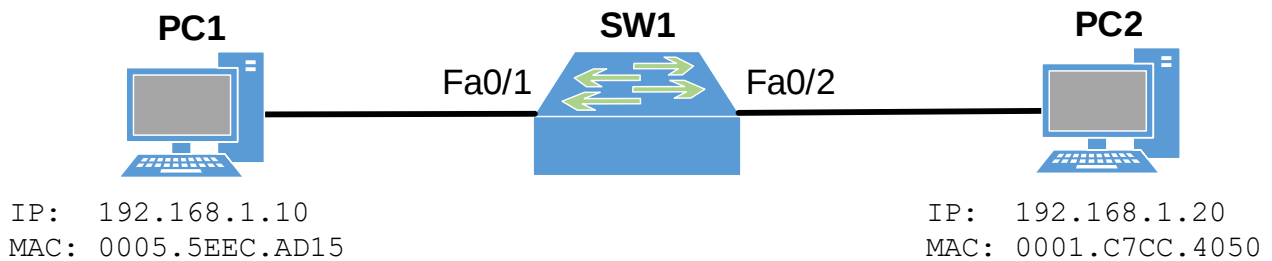


Рисунок 5.55 – Приклад мережі

#### Налаштування безпеки портів:

```
SW1(config)#interface range f0/1-2
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport port-security maximum 1
SW1(config-if-range)#switchport port-security mac-address sticky
SW1(config-if-range)#switchport port-security violation shutdown
SW1(config-if-range)#switchport port-security
SW1(config-if-range)#exit
```

#### Вимкнення решти портів:

```
SW1(config)#interface range f0/3-24, g0/1-2
SW1(config-if-range)#shutdown
```

#### За потреби можна очистити таблицю MAC-адрес для підключення нових пристроїв:

```
SW1# clear port-security all
```

### Результати виконання команд моніторингу та діагностики для розглянутого прикладу

З метою перегляду інформації про функціонування портів комутатора у захищеному режимі для розглянутого прикладу використано команди **show port-security**, **show port-security interface <interface-number>**, **show port-security address**:

Налаштовуємо IP-адреси комп'ютерів та перевіряємо зв'язок між ними:

```
C:\>ping 192.168.1.20
```

```
Pinging 192.168.1.20 with 32 bytes of data:
```

```
Reply from 192.168.1.20: bytes=32 time=1ms TTL=128
Reply from 192.168.1.20: bytes=32 time<1ms TTL=128
Reply from 192.168.1.20: bytes=32 time<1ms TTL=128
Reply from 192.168.1.20: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 192.168.1.20:
```

```
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
 Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Переглядаємо, чи захист портів включено на потрібних портах:

```
SW1#show run | begin interface
interface FastEthernet0/1
switchport mode access
```

```

switchport port-security
switchport port-security mac-address sticky
switchport port-security mac-address sticky 0005.5EEC.AD15
!
interface FastEthernet0/2
switchport mode access
switchport port-security
switchport port-security mac-address sticky
switchport port-security mac-address sticky 0001.C7CC.4050
!
interface FastEthernet0/3
shutdown
!
...

interface GigabitEthernet0/2
shutdown
!
...

```

Перегляд короткої інформації по станам *port-security*:

```

SW1#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
 (Count) (Count) (Count)

 Fa0/1 1 1 0 Shutdown
 Fa0/2 1 1 0 Shutdown

```

```

SW1#show port-security address
 Secure Mac Address Table

Vlan Mac Address Type Ports Remaining Age
 (mins)

 1 0005.5EEC.AD15 SecureSticky Fa0/1 -
 1 0001.C7CC.4050 SecureSticky Fa0/2 -

Total Addresses in System (excluding one mac per port) : 0
Max Addresses limit in System (excluding one mac per port) : 1024

```

Перегляд інформації по стану *port-security* на порту **f0/2**:

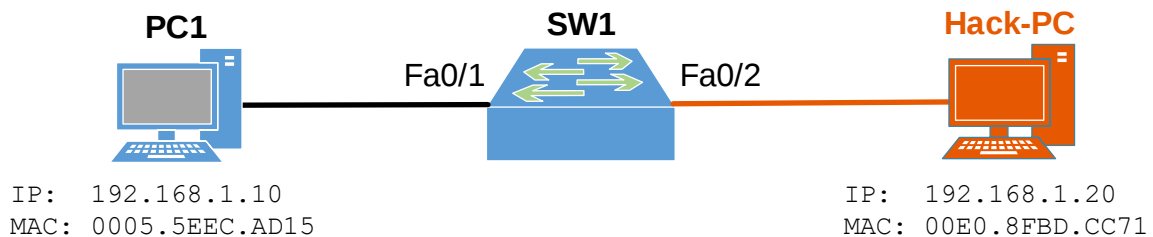
```

SW1#show port-security interface f0/2
Port Security : Enabled
Port Status : Secure-up
Violation Mode : Shutdown
Aging Time : 0 mins
Aging Type : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 1
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 1
Last Source Address:Vlan : 0001.C7CC.4050:1
Security Violation Count : 0

```

## Результати виконання команд моніторингу та діагностики для прикладу з спробою порушення безпеки

З метою чистоти експерименту в прикладі наведеному на рисунку 5.56 до порту комутатора **Fa0/2** під'єднано несанкціонований комп'ютер і повторно зроблено перегляд інформації про функціонування портів комутатора за допомогою команд: **show port-security**, **show port-security interface <interface-number>**, **show interfaces <interface-number> status**.



**Рисунок 5.56 – Приклад мережі модифікований**

На **Hack-PC** використана та сама IP-адреса, що була і на **PC-2**:

```
C:\>ping 192.168.1.20
```

```
Pinging 192.168.1.20 with 32 bytes of data:
```

```
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

```
Ping statistics for 192.168.1.20:
```

```
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Перегляд короткої інформації по станам *port-security*:

```
SW1#show port-security
```

| Secure Port | MaxSecureAddr<br>(Count) | CurrentAddr<br>(Count) | SecurityViolation<br>(Count) | Security Action |
|-------------|--------------------------|------------------------|------------------------------|-----------------|
| Fa0/1       | 1                        | 1                      | 0                            | Shutdown        |
| Fa0/2       | 1                        | 1                      | 1                            | Shutdown        |

Перегляд інформації по стану *port-security* на порту **f0/2**:

```
SW1#show port-security interface f0/2
```

```
Port Security : Enabled
Port Status : Secure-shutdown
Violation Mode : Shutdown
Aging Time : 0 mins
Aging Type : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 1
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 1
Last Source Address:Vlan : 00E0.8FBD.CC71:1
Security Violation Count : 1
```

Перевірки стану порту f0/2:

```
SW1#show interfaces f0/2 status
```

| Port  | Name | Status       | Vlan | Duplex | Speed | Type         |
|-------|------|--------------|------|--------|-------|--------------|
| Fa0/2 |      | err-disabled | 1    | auto   | auto  | 10/100BaseTX |

Для виведення порту з режиму *err-disabled* достатньо виключити і включити порт комутатора:

```
SW1(config-if)#shutdown
```

```
%LINK-5-CHANGED: Interface FastEthernet0/2, changed state to administratively down
```

```
SW1(config-if)#no shutdown
```

*\*Примітка:* перед перевантаженням порту треба впевнитись в відновленні безпеки (для розглянутого випадку – повернутись до схеми як на рисунку 5.2.6.1)

```
SW1#show interfaces f0/2 status
```

| Port  | Name | Status    | Vlan | Duplex | Speed | Type         |
|-------|------|-----------|------|--------|-------|--------------|
| Fa0/2 |      | connected | 1    | auto   | auto  | 10/100BaseTX |

## ПРАКТИЧНЕ ЗАВДАННЯ

**Мета роботи:** ознайомитися з технологіями захисту мережі засобами комутатора, а саме забезпечення контролю доступу до мережевого обладнання шляхом обмеження фізичного доступу до портів комутатора, запобігання несанкціонованому підключенню пристроїв із мережею, виявлення атак на рівні мережевого доступу та забезпечення цілісності та конфіденційності мережевого трафіку за допомогою комутаційного обладнання *Cisco*.

### Порядок виконання роботи

1. Створити фізичний проєкт мережі або у середовищі програмного симулятора/емулятора створити мережу (рисunek 5.57). При побудові звернути увагу на вибір комутатора, точки доступу, мережних модулів та плат, а також мережних з'єднань. При побудові підмережі слід вибирати потрібний тип кабелю для відповідної технології.

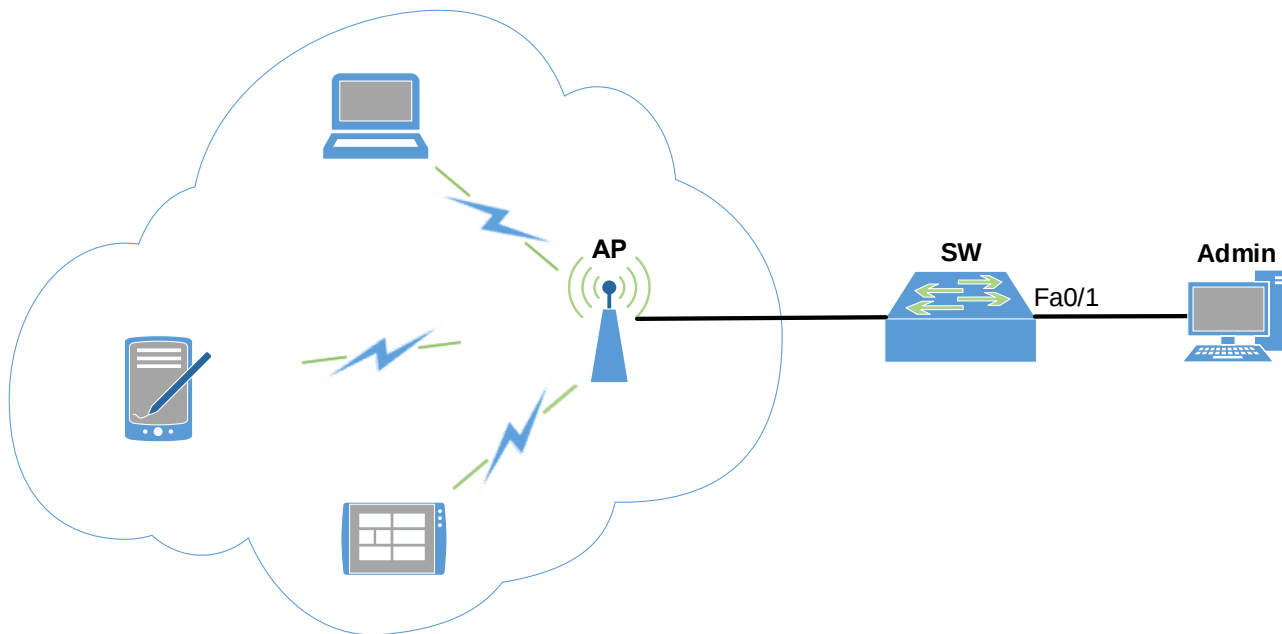


Рисунок 5.57 – Проєкт мережі

2. Під'єднати комп'ютер **Admin** до порту комутатора **Fa0/1**. Під'єднати бездротову точку доступу **AP** до порту комутатора згідно варіанту таблиці 5.54.

Таблиця 5.54 – Параметри захисту мережі засобами комутатора

| № варіанта | Локальна мережа | Порт комутатора для підключення AP | Кіл-ть дозволених бездротових пристроїв | Режим порушення безпеки порту |
|------------|-----------------|------------------------------------|-----------------------------------------|-------------------------------|
| 1          | 192.168.1.0/24  | Fa0/2                              | 2                                       | restrict                      |
| 2          | 192.168.2.0/24  | Fa0/3                              | 3                                       | shutdown                      |
| 3          | 192.168.3.0/24  | Fa0/4                              | 4                                       | restrict                      |
| 4          | 192.168.4.0/24  | Fa0/5                              | 2                                       | shutdown                      |
| 5          | 192.168.5.0/24  | Fa0/6                              | 3                                       | restrict                      |
| 6          | 192.168.6.0/24  | Fa0/7                              | 4                                       | shutdown                      |
| 7          | 192.168.7.0/24  | Fa0/8                              | 2                                       | restrict                      |
| 8          | 192.168.8.0/24  | Fa0/9                              | 3                                       | shutdown                      |
| 9          | 192.168.9.0/24  | Fa0/10                             | 4                                       | restrict                      |
| 10         | 192.168.10.0/24 | Fa0/11                             | 2                                       | shutdown                      |
| 11         | 192.168.11.0/24 | Fa0/12                             | 3                                       | restrict                      |
| 12         | 192.168.12.0/24 | Fa0/13                             | 4                                       | shutdown                      |
| 13         | 192.168.13.0/24 | Fa0/14                             | 2                                       | restrict                      |
| 14         | 192.168.14.0/24 | Fa0/15                             | 3                                       | shutdown                      |
| 15         | 192.168.15.0/24 | Fa0/16                             | 4                                       | restrict                      |
| 16         | 192.168.16.0/24 | Fa0/17                             | 2                                       | shutdown                      |
| 17         | 192.168.17.0/24 | Fa0/18                             | 3                                       | restrict                      |
| 18         | 192.168.18.0/24 | Fa0/19                             | 4                                       | shutdown                      |
| 19         | 192.168.19.0/24 | Fa0/20                             | 2                                       | restrict                      |
| 20         | 192.168.20.0/24 | Fa0/21                             | 3                                       | shutdown                      |
| 21         | 192.168.21.0/24 | Fa0/22                             | 4                                       | restrict                      |
| 22         | 192.168.22.0/24 | Fa0/23                             | 2                                       | shutdown                      |
| 23         | 192.168.23.0/24 | Fa0/24                             | 3                                       | restrict                      |
| 24         | 192.168.24.0/24 | Fa0/2                              | 4                                       | shutdown                      |
| 25         | 192.168.25.0/24 | Fa0/3                              | 2                                       | restrict                      |
| 26         | 192.168.26.0/24 | Fa0/4                              | 3                                       | shutdown                      |
| 27         | 192.168.27.0/24 | Fa0/5                              | 4                                       | restrict                      |
| 28         | 192.168.28.0/24 | Fa0/6                              | 2                                       | shutdown                      |
| 29         | 192.168.29.0/24 | Fa0/7                              | 3                                       | restrict                      |
| 30         | 192.168.30.0/24 | Fa0/8                              | 4                                       | shutdown                      |

3. Призначити *IP*-адреси згідно варіанту всім дротовим і бездротовим пристроям (телефони, ноутбуки, планшети, тощо), щоб вони належали до однієї локальної мережі. Кількість бездротових пристроїв, яким дозволяється мати доступ до локальної мережі визначаються варіантом з таблиці 5.54.

4. Перевірити з'єднання між пристроями за допомогою команди **ping**.

5. Налаштувати безпеку порту комутатора, до якого підключена точка доступу **AP**, з урахуванням максимальної можливої кількості бездротових пристроїв та режиму порушень безпеки порту (таблиця 5.54)

6. Спробувати під'єднати зайвий бездротовий пристрій. Перевірити інформації про функціонування портів комутатора у захищеному режимі за допомогою команд діагностування та зробити висновки про реагування захисту на спробу несанкційного підключення.

### КОНТРОЛЬНІ ПИТАННЯ

1. Які основні загрози для мережі можуть бути зменшені за допомогою заходів безпеки на комутаторах *Cisco*?
2. Як використовується *Port Security* для обмеження кількості підключених пристроїв до комутатора *Cisco*?
3. Для чого *DHCP Snooping* при захисті мережі на комутаторах *Cisco*?
4. Що таке *BPDUGuard* для захисту мережі на комутаторах *Cisco*?
5. Які переваги має використання *Port Security*?
6. Як визначити максимальну кількість *MAC*-адрес, яку можна вивчити для кожного порту з *Port Security*?
7. Як включити функцію *Port Security* на портах комутатора?
8. Що таке *sticky* режим у *Port Security* і в чому його особливості?
9. Які інші функції безпеки можна поєднати з *Port Security* для покращення захисту мережі?
10. Як встановити періодичність вивчення *MAC*-адрес для портів у режимі *Port Security*?

## 5.2.7 НАЛАШТУВАННЯ WLAN ІЗ ЗАСТОСУВАННЯМ WLC

### ПОЧАТКОВЕ НАЛАШТУВАННЯ WLC КОНТРОЛЕРА CISCO

Контролер бездротової мережі (*Wireless LAN controller, WLC*) – це спеціалізований компонент комп'ютерної мережі, що призначений для моніторингу, керування та забезпечення точок бездротового доступу (*AP*) в централізовано керованих бездротових мережах. Застосування контролерів дозволяє суттєво скоротити час, що витрачається на налаштування, моніторинг та усунення несправностей. Без використання таких контролерів кожен з *AP* в бездротовій мережі доводиться налаштовувати, контролювати й обслуговувати окремо. Контролер бездротової мережі централізує інфраструктуру бездротової мережі, керує налаштуваннями бездротових *AP*, може автоматично розподіляти їх канали, смуги пропускання тощо. Це називається архітектурою *Split-MAC*. Архітектура *Split MAC* розділяє реалізацію функцій *MAC* між точкою доступу та контролером. Функції *MAC* в реальному часі містять такі функції, як генерація маяка, передача і відповідь на опитування, обробка керуючих кадрів (наприклад, запит на відправку і дозвіл на відправку), ретрансляція і так далі. Функції не в режимі реального часу містять автентифікацію і деавтентифікацію, асоціацію і повторну асоціацію, з'єднання між *Ethernet* і бездротовою локальною мережею; фрагментацію тощо. Топологію під'єднання *AP* до контролера показано на рисунку 5.58.

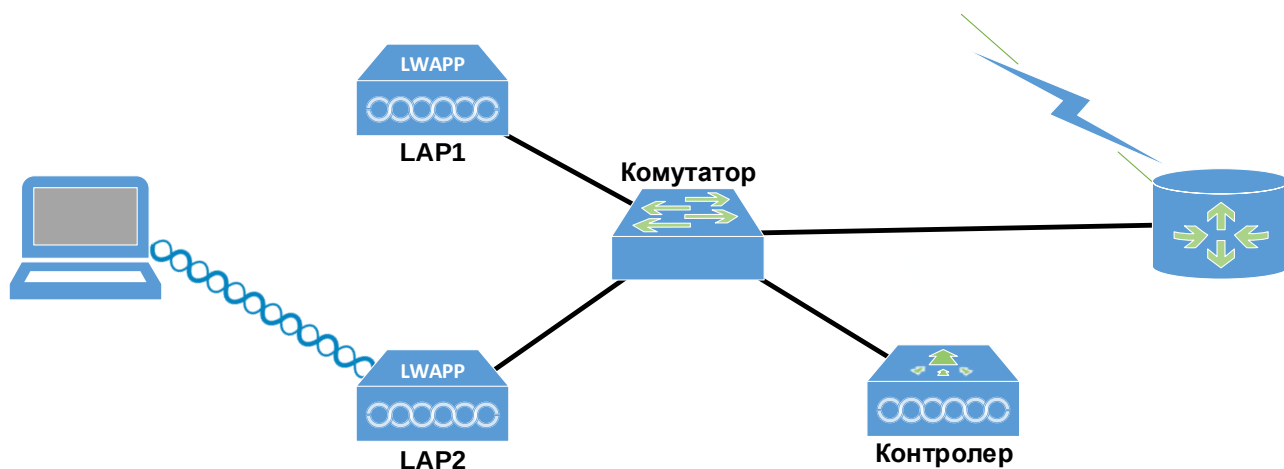


Рисунок 5.58 – Топологія під'єднання контролера до мережі

*AP* під'єднуються до контролера через локальну комутовану мережу (за протоколом *CAPWAP*). Один контролер може підтримувати до ста *AP*, в залежності від його продуктивності. Декілька контролерів можуть бути згруповані разом, забезпечуючи злагоджену роботу *AP*, відповідно до загальних налаштувань. Переваги такого підходу очевидні: централізоване управління, гнучкість налаштувань, відмовостійкість, балансування навантаження, інтелектуальні функції, такі як безшовний роумінг, управління частотним ресурсом, потужністю, якістю обслуговування, авторизацією.

На *AP*, під'єднаних до контролера, так званих «полегшених» (*AIR-LAP-XXXX*), образ *Cisco IOS* працює практично так само, як і на звичайних (автономних, *AIR-AP-XXXX*). Відмінність полягає в тому, що тут немає режиму конфігурації терміналу, так як точки налаштовуються централізовано контролером. Програмне забезпечення між автономними та полегшеними точками доступу можна змінювати вручну.

Після завантаження *AP* здійснює пошук найкращого доступного контролера, авторизується, завантажує образ *IOS* (якщо потрібно), налаштування та починає обслуговувати клієнтів. Між *AP* і контролером організовуються канал керування (порт *UDP 5246*) і канал передачі даних (порт *UDP 5247*). Дані користувача інкапсулюються в *UDP*-пакети незалежно від номера *WLAN/VLAN* клієнта, що дозволяє розміщувати точки доступу в будь-якому місці мережі (на портах доступу комутаторів) і централізовано управляти безпекою на рівні *VLAN* на стороні контролера.

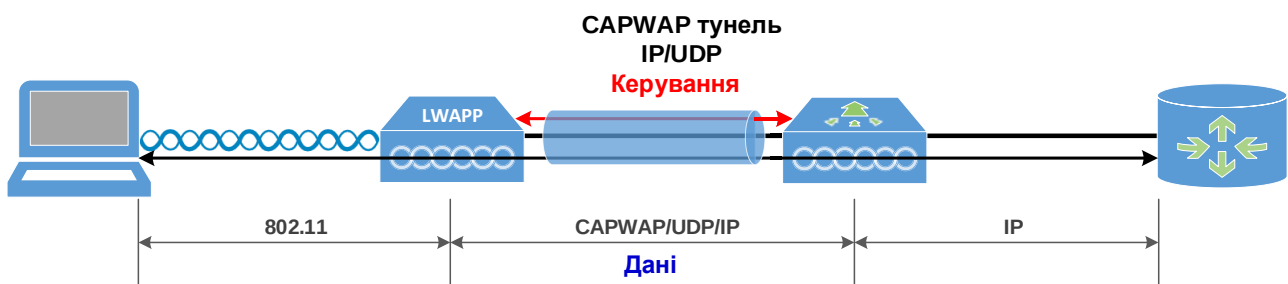


Рисунок 5.59 – Протоколи передачі даних в бездротовій та дротовій мережі

Після під'єднання контролера до живлення, консольного порту та локальної мережі потрібне початкове налаштування. Зазвичай таке налаштування потрібно зробити також після повного скидання контролера. Далі буде показана процедура первинного налаштування на прикладі контролера *WLC 2112*. Перед доступом до командного рядка процедура завантаження запускає служби:

```
Cryptographic library self-test...passed!
XML config selected
Validating XML configuration
Cisco is a trademark of Cisco Systems, Inc.
Software Copyright Cisco Systems, Inc. All rights reserved.
Cisco AireOS Version 7.0.220.0
Initializing OS Services: ok
Initializing Serial Services: ok
Initializing Network Services: ok
Starting ARP Services: ok
Starting Trap Manager: ok
Starting Network Interface Management Services: ok
Starting System Services: ok
Starting Fastpath Hardware Acceleration: ok
Starting Switching Services: ok
Starting QoS Services: ok
Starting Policy Manager: ok
Starting Data Transport Link Layer: ok
```

```
Starting Access Control List Services: ok
Starting System Interfaces: ok
Starting Client Troubleshooting Service: ok
Starting Management Frame Protection: ok
Starting Certificate Database: ok
Starting VPN Services: ok
Starting LWAPP: ok
Starting CAPWAP: ok
Starting LOCP: ok
Starting Security Services: ok
Starting Policy Manager: ok
Starting Authentication Engine: ok
Starting Mobility Management: ok
Starting Virtual AP Services: ok
Starting AireWave Director: ok
Starting Network Time Services: ok
Starting Cisco Discovery Protocol: ok
Starting Broadcast Services: ok
Starting Logging Services: ok
Starting DHCP Server: ok
Starting IDS Signature Manager: ok
Starting RFID Tag Tracking: ok
Starting RBCP: ok
Starting Mesh Services: ok
Starting TSM: ok
Starting CIDS Services: ok
Starting Ethernet-over-IP: ok
Starting DTLS server: enabled in CAPWAP
Starting CleanAir: ok
Starting WIPS: ok
Starting SSHPM LSC PROV LIST: ok
Starting RRC Services: ok
Starting FMC HS: ok
Starting Management Services:
Web Server: ok
CLI: ok
Secure Web: Web Authentication Certificate not found (error) . If you cannot access
management interface via HTTPS please reconfig
Secure Web: Web Authentication Certificate not found (error) . If you cannot access
management interface via HTTPS please reconfigure Virtual Interface.
(Cisco Controller)
Welcome to the Cisco Wizard Configuration Tool
Use the '-' character to backup
```

На цьому етапі потрібно призупинити функцію авто-налаштування. Це дозволяє контролеру автоматично завантажувати заздалегідь підготовлений конфігураційний файл через *TFTP*. Таким чином, можна розгорнути велику кількість контролерів у мережі без необхідності вручну налаштовувати кожен з них.

```
Would you like to terminate autoinstall? [yes]: yes
AUTO-INSTALL: process terminated -- no configuration loaded
```

Спочатку потрібно задати ім'я контролера (запропоноване генерується з *MAC*-адреси), логін і пароль адміністратора:

```
System Name [Cisco_bb:bb:40] (31 characters max): wlc.lab.local
Enter Administrative User Name (24 characters max): adminwlc
Enter Administrative Password (3 to 24 characters): c!sc0Wlc
Re-enter Administrative Password : c!sc0Wlc
```

Далі слідує налаштування адресації інтерфейсу управління. Це логічний інтерфейс, за допомогою якого контролер обмінюється даними із локальною мережею. Загальноприйнятою практикою є об'єднання всіх фізичних інтерфейсів контролера в групу *EtherChannel* (це можна зробити пізніше), і в такому загальному каналі налаштувати, скільки логічних інтерфейсів потрібно, кожен у певній VLAN.

Для порту доступу, або якщо існує *Native VLAN*, потрібно вказати число 0. Адреса сервера *DHCP* потрібна для пересилання клієнтських запитів *DHCP* на нього, якщо такий налаштовано. Контролер в такому випадку буде пересилати *DHCP*-запити (хоча він також може виступати в ролі *DHCP*-сервера).

```
Management Interface IP Address: 192.168.200.11
Management Interface Netmask: 255.255.255.0
Management Interface Default Router: 192.168.200.1
Management Interface VLAN Identifier (0 = untagged): 0
Management Interface Port Num [1]: 1
Management Interface DHCP Server IP Address: 192.168.200.1
```

Всі *WLC* контролери, крім моделі 5508, використовують окремий логічний інтерфейс *AP-manager* для зв'язку з точками доступу. Налаштовувати його адресу потрібно, як правило, з тієї ж мережі:

```
AP Manager Interface IP Address: 192.168.200.12
AP-Manager is on Management subnet, using same values
AP Manager Interface DHCP Server (192.168.200.1): 192.168.200.1
Virtual Gateway IP Address: 10.1.1.1
```

Останній логічний інтерфейс – віртуальний. Використовується для пересилання *DHCP*-запитів клієнтам, вебавторизації, роумінгу. Він призначається адресі, яка ніде в мережі не використовується і не маршрутизується. Досить часто використовують адресу *1.1.1.1*. Але зараз є така адреса в Інтернет, тому рекомендовано далі використати адресу *10.1.1.1*.

**Примітка:** Вона повинна бути однаковою для всіх контролерів, між якими можуть переміщатися клієнти.

Для цілей клієнтського роумінгу контролери об'єднуються в групи мобільності, назву яких пропонується встановити. Також пропонується вказати назву групи контролерів, які в сукупності забезпечують радіоуправління (частотний план і потужність, радіочастотна група). Хоча ці групи можуть відрізнятися за набором пристроїв та назвою, тут вказано загальну назву (можна змінити потім):

```
Mobility/RF Group Name: WIFILAB
```

Майстер пропонує створити одну бездротову мережу (*WLAN*). Потрібно обов'язково створити таку мережу, але потім її можна видалити або налаштувати через вебінтерфейс.

```
Network Name (SSID) : WIFILAB1
```

Режим мосту пакетів *DHCP* фактично не використовується на практиці.

```
Configure DHCP Bridging Mode [yes][NO] : no
```

Є можливість дозволити роботу бездротових клієнтів, у яких *IP*-адреса прописана статично. В дійсності ця опція вмикає кешування *ARP* на контролері, що напряду впливає на продуктивність.

```
Allow Static IP Addresses [YES][no] : yes
```

Для цілей авторизації клієнта (а також доступу до контролера, і деяких специфічних інших цілей) може використовуватися зовнішній *RADIUS*-сервер, який зручніше налаштувати пізніше через вебінтерфейс.

```
Configure a RADIUS Server now? [YES][no] : no
Warning! The default WLAN security policy requires a RADIUS server.
Please see documentation for more details.
```

Кожна точка доступу та контролер мають дозволені коди для країни, в якій вони працюють. Насправді код країни визначає допустимий набір частот (каналів) та межі потужності, що впливає на роботу алгоритмів автоматичного вибору частоти тощо. Рекомендується, щоб список кодів країн був налаштований однаково по всій мережі.

```
Enter Country Code list (enter 'help' for a list of countries) [US] : UA
```

Контролер пропонує увімкнути мережі 2,4 і 5 ГГц (11b/g/n і 11a/n відповідно). В реальності цей параметр передається на точки доступу, пов'язані з контролером, на яких, власне, і працює радіо-інтерфейс. Найкращою практикою є увімкнення радіо-інтерфейсів через вебінтерфейс після того, як будуть налаштовані всі інші параметри.

```
Enable 802.11b Network [YES][no] : no
Enable 802.11a Network [YES][no] : no
Enable 802.11g Network [YES][no] : no
```

Запит на увімкнення алгоритмів динамічного управління частотними ресурсами і потужністю.

```
Enable Auto-RF [YES][no]: yes
```

Налаштування часу (сервер часу і статичний час/часовий пояс) важливі не тільки для коректності міток у файлах журналу, але і для роботи авторизації сертифікатів точок доступу (в них використовується *CAPWAP* на основі *DTLS*, заснованих на сертифікатах з перевірками термінів дії).

```
Configure a NTP server now? [YES][no]: no
Configure the system time now? [YES][no]: no
Warning! No AP will come up unless the time is set.
Please see documentation for more details.
```

В кінцевому діалозі контролер запитає збереження конфігурації та перезавантажиться:

```
Configuration correct? If yes, system will save it and reset. [yes][NO]: yes
Configuration saved!
Resetting system with new configuration...
```

Після перезавантаження пристрій повинен бути доступний для налаштування через вебінтерфейс керування за адресою: **https://192.168.200.11**.

Якщо браузер видає інформацію про небезпечність під'єднання, необхідно погодитися на відкриття небезпечного з'єднання.

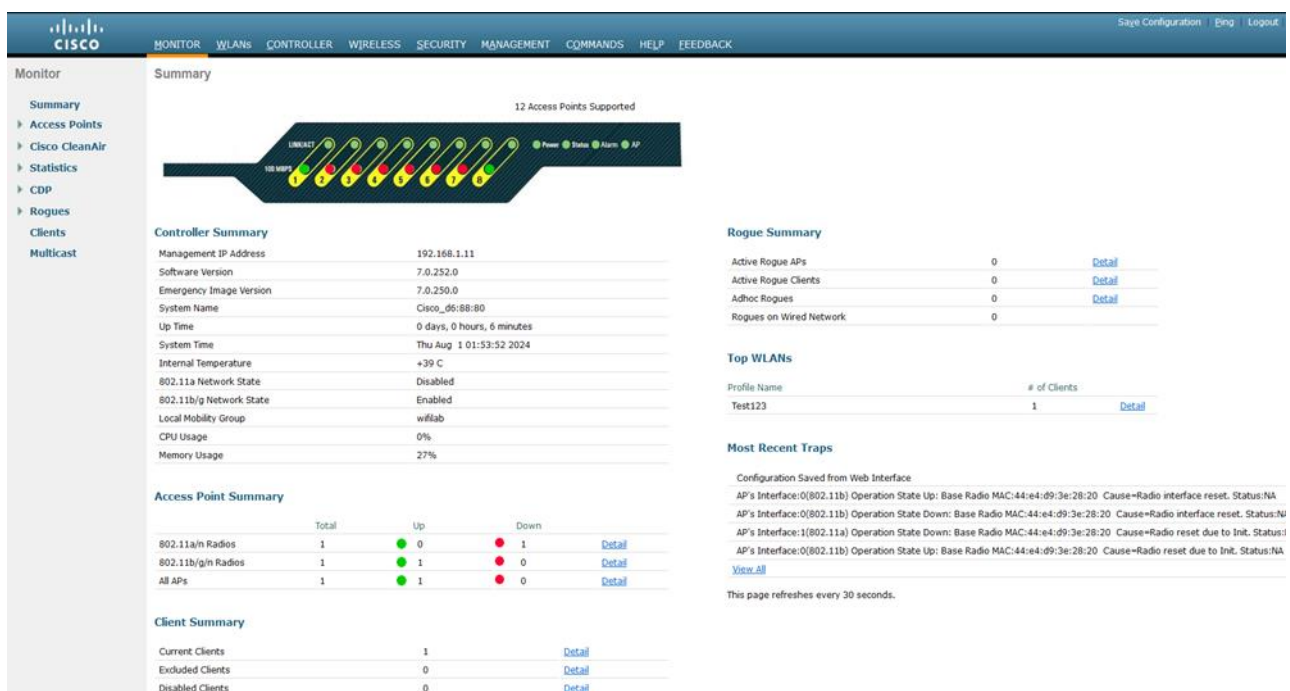


Рисунок 5.59 – Вебінтерфейс керування контролером бездротової мережі

Для підтвердження легальності сайту, контролеру можна згодом видати сертифікат для роботи *SSL*.

## Під'єднання точок доступу до контролера

Наступним кроком у побудові бездротової мережі буде під'єднання точок доступу до контролера, який буде обслуговувати клієнтів бездротової мережі.

Незважаючи на відкритість протоколу *CAPWAP*, контролер *Cisco* буде працювати тільки з точками доступу *Cisco*, про що прямо заявлено виробником в документації. На сьогоднішній день існує кілька класів *AP*:

- з одним радіоприймачем 2,4 ГГц (b/g/n) або двома 2,4 і 5 ГГц (a/n/ac). У зв'язку із забрудненням першого діапазону (каналів 20 МГц, що не перекриваються, всього три: 1, 6, 11) настійно рекомендується вибирати дводіапазонні точки доступу, так як контролер має засоби переміщення клієнтів в більш вільний діапазон 5 ГГц (23 канали);
- з підтримкою високих швидкостей передавання, що в основному досягається іншими модуляціями сигналу (*MCS*), збільшенням кількості вхідних/вихідних потоків (*MIMO*), об'єднанням каналів, зміною міжкадрового інтервалу тощо). Слід пам'ятати, що швидкість передачі даних між клієнтом і точкою доступу має зворотну залежність від відстані, а сумарна пропускна здатність сегмента мережі розподіляється між усіма користувачами;
- з підтримкою під'єднання зовнішніх антен, або з вбудованими;
- мають можливість працювати автономно (*AP*), або тільки через контролер (*LAP*).

Щоб з'ясувати, скільки точок доступу потрібно для покриття бездротовою мережею того чи іншого приміщення або будівлі, необхідно виміряти рівень сигналу після перешкод (стіни, меблі тощо), провести розрахунки врахувавши кращі практики.

Кількість підтримуваних точок доступу визначається ліцензією. Для деяких контролерів можна придбати додаткові ліцензії. Рекомендується, для підвищення відмово-стійкості, мати мінімум два контролера, і розподілити точки між ними.

Розглянемо процес під'єднання точки доступу до *WLC*. Для того, щоб зареєструватися на *WLC*, *AP* необхідно виконати таку послідовність дій:

1. *AP* надсилає запит на виявлення *DHCP* для отримання *IP*-адреси, якщо раніше не була налаштована статична *IP*-адреса.
2. Точка доступу надсилає повідомлення із запитом на виявлення за протоколом *LWAPP* до *WLC*.
3. Будь-який *WLC*, який отримує запит на виявлення з *LWAPP*, відповідає на нього повідомленням-відповіддю на виявлення *LWAPP*.
4. З отриманих *LWAPP*-відповідей *LAP* вибирає *WLC* для приєднання.
5. Потім *LAP* надсилає запит на приєднання за протоколом *LWAPP* до *WLC* і очікує на відповідь про приєднання.
6. *WLC* перевіряє *LAP*, а потім надсилає *LAP* відповідь про приєднання за протоколом *LWAPP*.

7. *LAP* перевіряє *WLC*, що завершує процес виявлення та приєднання. Процес приєднання за протоколом *LWAPP* включає взаємну автентифікацію та отримання ключа шифрування, який використовується для захисту процесу приєднання та майбутніх контрольних повідомлень.
8. *LAP* реєструється на контролері.

Перша проблема, з якою стикається *LAP*, полягає в тому, як визначити, куди надсилати запити на виявлення за протоколом *LWAPP* (крок 2). *LAP* використовує процедуру пошуку та певний алгоритм виявлення для того, щоб визначити список *WLC*, до яких *LAP* може надсилати повідомлення із запитом на виявлення.

Розглянемо детальніше процес під'єднання. Точка доступу може або отримати *IP*-адресу від *DHCP*-сервера (будь-якого виробника: Cisco IOS, Microsoft, Unix ISC-DHCP), або використовувати статично налаштовану *IP*-адресу. Залежно від конфігурації існуючої мережі можна використовувати будь-який спосіб.

Щоб статично налаштувати *IP*-адресу для *AP*, потрібно під'єднатися до консольного порту, маючи подане живлення на *LAP*. За замовчуванням логін і пароль – «**Cisco**» (з великої літери), пароль для привілейованого режиму такий самий. Якщо є ряд невдалих спроб отримати *IP*-адресу, точка перезавантажується і продовжує процес безкінечно. Щоб запобігти перезавантаженню, можна скасувати цей процес за допомогою такої команди:

```
debug capwap client no-reload
```

Далі є можливість налаштувати *IP*-адресу *LAP*:

```
capwap ap ip address 192.168.200.101 255.255.255.0
capwap ap ip default-gateway 192.168.200.1
```

**Увага!** *LAP* виконує спеціальний варіант *IOS*, в якому режим конфігурації недоступний.

Далі необхідно встановити адресу контролерів. Перший контролер (достатньо одного):

```
capwap ap controller ip address 192.168.200.11
```

Необхідно вказувати адресу інтерфейсу керування контролером (вебінтерфейс), а не адресу інтерфейсу *AP-manager*. Результат налаштування можна побачити за допомогою команди **show capwap client config**:

```
configMagicMark 0xF1E2D3C4
chkSumV2 47358
chkSumV1 27913
swVer 7.0.220.0
adminState ADMIN_ENABLED(1)
```

```

name ap2.wlab
location Lab510
group name WIFILAB
mwarName wlc2.xxx.xxx.nxx
mwarIPAddress 192.168.200.11
mwarName
mwarIPAddress 0.0.0.0
mwarName
mwarIPAddress 0.0.0.0
ssh status Enabled
Telnet status Disabled

```

Для автоматичного налаштування *IP*-адреси точки доступу через *DHCP* необхідно також вказати додаткову опцію, номер 43, яка повідомляє точці доступні адреси контролера. Значення параметра (в *HEX*-вигляді) знаходиться в форматі *TLV*, наприклад, *f108c0a8c80b*, де *0xf1* (241) — номер параметра опції, *0x08* (довжина даних, два рази по 4 байти/октети *IP*-адреси), *0xc0a8c80b* перетворюється на 192.168.200.11.

Точка доступу також може отримувати адреси контролерів через *DNS*-запит, або від сусідів по радіо-каналі.

Для роботи *LAP* в локальній мережі потрібно тільки це підключення по *IP* протоколу до контролера. Проте, у випадку з відео або голосовими застосунками в радіомережі потрібно буде налаштувати пріоритезацію трафіку. Навіть якщо одна точка доступу надає можливість підключення до декількох *SSID* (радіомереж) зі своїми політиками безпеки і *VLAN*-з'єднаннями, турбуватися потрібно тільки про налаштування дротового мережного інтерфейсу на рівні контролера.

Після встановлення *AP* в локальній мережі необхідно перевірити, як вона намагається підключитися до контролера в меню останнього **Monitor->AP Join**, що показано на рисунку 5.60:

The screenshot shows the Cisco WLC Monitor interface. The left sidebar contains a navigation menu with options like Summary, Access Points, Cisco CleanAir, Statistics, CDP, Rogues, Clients, and Multicast. The main content area is titled 'AP Join Stats Detail' and includes a 'General' section with fields for Base MAC Address, AP Name, Ethernet MAC Address, IP Address, and Status. Below this is a 'Last AP Join' table showing three entries with timestamps and messages. At the bottom, there is a 'Discovery Phase Statistics' table.

| General              |                   |
|----------------------|-------------------|
| Base MAC Address     | 44:e4:d9:3e:28:20 |
| AP Name              | AP503d.e5f0.ea1b  |
| Ethernet MAC Address | 50:3d:e5:f0:ea:1b |
| IP Address           | 192.168.1.20      |
| Status               | Joined            |

| Last AP Join        |                                              |
|---------------------|----------------------------------------------|
| Timestamp           | Message                                      |
| Aug 01 01:51:19.616 | Received Discovery request and sent response |
| Aug 01 01:51:31.311 | Received Join request and sent response      |
| Aug 01 01:51:31.720 | Received Config request and sent response    |

| Discovery Phase Statistics     |   |
|--------------------------------|---|
| Requests Received              | 3 |
| Responses Sent                 | 3 |
| Unsuccessful Request Processed | 0 |

Рисунок 5.60 – Під'єднання точки доступу в меню Monitor-AP Join

## УСУНЕННЯ ПРОБЛЕМ З ПІД'ЄДНАННЯМ ТОЧКИ ДОСТУПУ

Однак існує імовірність, що точка доступу начебто виявляє контролер, але не може підключитися до нього. Це майже напевно пов'язано з політиками безпеки, які встановлені в меню контролера «**Security→AAA→AP Policies**». Практично всі ці політики так чи інакше пов'язані з сертифікатами.

У зв'язку з деталями роботи *CAPWAP*, дані між точкою і контролером шифруються сертифікатами. Сертифікати є, звичайно ж, на контролері і на точці доступу. Контролер поставляється з уже встановленими сертифікатами *Cisco Systems* (*root* і його похідні), може нести додаткові сертифікати для вебавторизації і на локальний сервер *RADIUS*, які не поширюються на точки доступу, а також може бути доповнений сертифікатом власного Центру сертифікації (*PKI*). Існує чотири типи сертифікатів точки доступу, які потім авторизуються на контролері:

- *MIC* (*Manufacture Installed Certificate*).
- *SSC* (*Self-Signed Certificate*).
- *LSC* (*Locally Significant Certificate*).
- *LBS-SSC* (*Location-Based Services-SSC*).

Сертифікат першого типу присутній на всіх точках доступу, включаючи *LAP*, випущені після 2006 року. Цей сертифікат програмується при виготовленні, підписується постачальником і прив'язується до вбудованої *MAC*-адреси точки доступу.

The screenshot shows the Cisco WLC configuration interface. The left sidebar has a tree view with 'Security' expanded, showing 'AAA' and 'RADIUS' sub-items. The main content area is titled 'AP Policies' and includes a 'Policy Configuration' section with three checkboxes: 'Accept Self Signed Certificate (SSC)' (checked), 'Accept Manufactured Installed Certificate (MIC)' (checked), and 'Accept Local Significant Certificate (LSC)' (unchecked). Below this is the 'AP Authorization List' table with columns 'MAC Address', 'Certificate Type', and 'SHA1 Key Hash'. There are four entries in the table, each with a blue checkmark in the right margin.

| MAC Address       | Certificate Type | SHA1 Key Hash                            |
|-------------------|------------------|------------------------------------------|
| 00:0f:24:4e:d5:e8 | SSC              | b045f1c4eaf54b7ff3f91209fb0ccf8905b2ea45 |
| 00:0f:24:4e:d6:aa | SSC              | d886bcdf0d22398538ccdef3f53d6ec7893463b8 |
| 00:0f:f7:29:7b:e8 | SSC              | de9f8b7f24ce7f924a9cae8f71049aea12851ba7 |
| 00:50:56:8c:00:14 | LBS-SSC          | 44e61eddb7fc8f648881bc5bfe1d16bf9bfa0d7  |

Рисунок 5.61 – Налаштувати типів сертифікатів AP для авторизації

Сертифікати *SSC* генеруються самою точкою в момент її конвертації в «полегшену», якщо сертифікат *MIC* відсутній. Він підписується сам собою.

Є можливість самостійно видавати сертифікати *LSC* разом зі своїм *PKI*, якщо у організації є інструкції щодо використання лише локальних інструментів для систем цифрового підпису та шифрування.

Сертифікат *LBS-SSC* генерується та використовується пристроями *Mobility Services Engine* під час зв'язку з контролером.

В меню «**Security**→**AAA**→**AP Policies**» можна налаштувати, які типи сертифікатів точок доступу приймати під час спроби авторизації.

Крім того, можна використовувати сервер *RADIUS* для перевірки дійсності *MAC*-адреси точки доступу. Також можна явно вказати *MAC*-адресу точки, яка використовує сертифікат *MIC* (auth-list). Однак для *SSC*-сертифікатів необхідно перерахувати кожен з них в будь-якому випадку, вказавши не тільки *MAC*-адресу *Ethernet*-інтерфейсу точки доступу, але і хеш сертифіката. На контролері необхідно ввімкнути налагодження процесу валідації сертифіката:

```
(Cisco Controller) > debug pm pki enable
*spamReceiveTask: Jun 10 21:49:39.450: sshpmGetCID: called to evaluate
ciscoDefaultIdCert
*spamReceiveTask: Jun 10 21:49:39.450: sshpmGetCID: comparing to row 0, CA cert
bsnOldDefaultCaCert
*spamReceiveTask: Jun 10 21:49:39.450: sshpmGetCID: comparing to row 1, CA cert
bsnDefaultRootCaCert
*spamReceiveTask: Jun 10 21:49:39.450: sshpmGetCID: comparing to row 2, CA cert
bsnDefaultCaCert
*spamReceiveTask: Jun 10 21:49:39.731: sshpmGetIssuerHandles: subject L=San Jose,
ST=California, C=US, O=Cisco Systems, MAILTO=support@cisco.com, CN=C1100-
000f244ed6aa
*spamReceiveTask: Jun 10 21:49:39.731: sshpmGetIssuerHandles: issuer L=San Jose,
ST=California, C=US, O=Cisco Systems, MAILTO=support@cisco.com, CN=C1100-
000f244ed6aa
*spamReceiveTask: Jun 10 21:49:39.732: sshpmGetIssuerHandles: Mac Address in
subject is 00:0f:24:4e:d6:aa
*spamReceiveTask: Jun 10 21:49:39.732: sshpmGetIssuerHandles: Cert Name in subject
is C1100-000f244ed6aa
*spamReceiveTask: Jun 10 21:49:39.732: sshpmGetIssuerHandles: Cert is issued by
Cisco Systems.
*spamReceiveTask: Jun 10 21:49:39.741: ssphmSsUserCertVerify: self-signed user
cert verified.
*spamReceiveTask: Jun 10 21:49:39.752: sshpmGetIssuerHandles: SSC Key Hash is
d886bcacf0d22398538ccdef3f53d6ec7893463b8
*spamReceiveTask: Jun 10 21:49:39.755: sshpmFreePublicKeyHandle: called with
0xf2de114
```

Потрібно обрати **Add**, тип *SSC*, скопіювати і вставити *MAC*-адресу і хеш у відповідні поля, Застосувати, зачекати перезавантаження *AP*, відключити налагодження (**debug disable-all**) та отримати інформацію про під'єднання *AP* (рисунок 5.62).

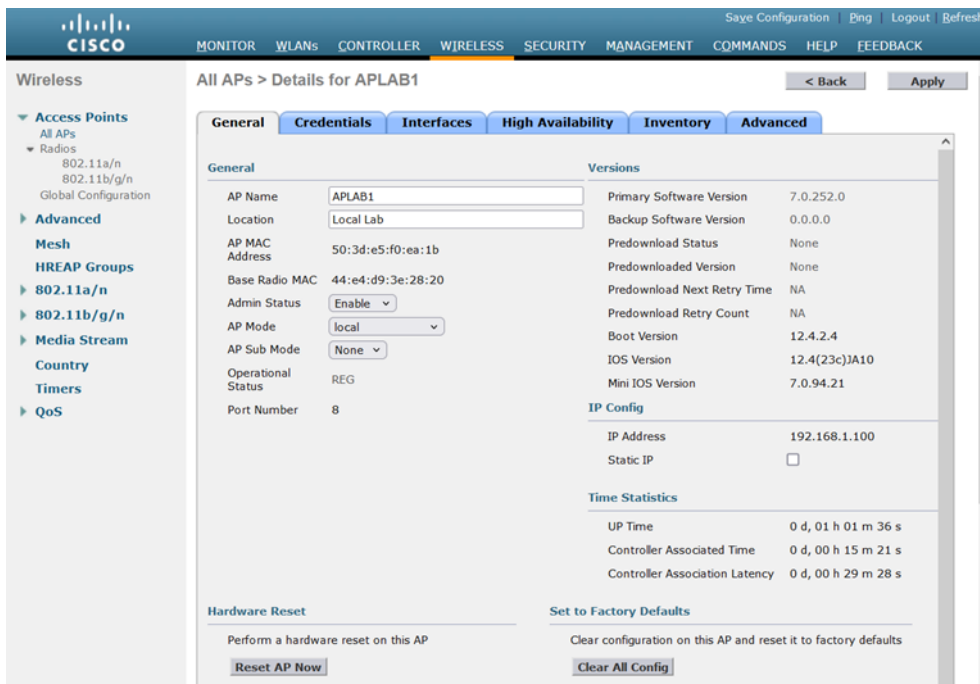


Рисунок 5.62 – Інформація про під'єднання AP

Також можна додати хеш сертифіката вручну:

```
config auth-list ap-policy ssc enable
config auth-list add ssc 00:0f:24:4e:d6:aa
d886bcaf0d22398538ccdef3f53d6ec7893463b8
```

Тепер, коли AP підключена, можна увімкнути радіоінтерфейси (*Wireless* – 802.11a/n, 802.11b/g/n), режимами -g, -n і перейти до налаштування бездротових мереж (SSID).

## НАЛАШТУВАННЯ БЕЗДРотовИХ МЕРЕЖ НА КОНТРОЛЕРІ CISCO

Коли виконано базові налаштування AP, які відповідно з'явилися у меню керування контролера, потрібно налаштувати самі мережі (WLAN, SSID) для надання послуг зв'язку користувачам.

Незважаючи на те, що керування контролером може здійснюватися за допомогою командного рядка, майже всі операції з налаштування проходять швидше і зручніше через вебінтерфейс. Доступ до контролера здійснюється через HTTP (рекомендовано перейти на HTTPS) з логіном та паролем, встановленими під час базового налаштування. Всі налаштування здійснюються в меню **WLANs** (рисунок 5.63).

Зазвичай кожна з бездротових мереж ідентифікується унікальним ім'ям або SSID. Кожна мережа може мати свій, незалежний набір параметрів для авторизації, шифрування, QoS, додаткових властивостей. Кожна точка доступу може обслуговувати (рекламувати) до 16 бездротових мереж. Залежно від моделі, контролер може обслуговувати до 512 мереж і до сотні точок доступу.

The screenshot shows the Cisco WLAN configuration interface. On the left, there's a sidebar with 'WLANs' and 'Advanced' options. The main area is titled 'WLANs' and shows a table of existing WLANs. The table has columns for 'WLAN ID', 'Type', 'Profile Name', 'WLAN SSID', 'Admin Status', and 'Security Policies'. There are two entries: ID 1 (Type: WLAN, Profile: LABTEST, SSID: LABTEST, Status: Enabled) and ID 2 (Type: WLAN, Profile: Guest, SSID: Guest, Status: Disabled). Above the table, there's a 'Current Filter' section and a 'Create New' button.

| WLAN ID | Type | Profile Name | WLAN SSID | Admin Status | Security Policies    |
|---------|------|--------------|-----------|--------------|----------------------|
| 1       | WLAN | LABTEST      | LABTEST   | Enabled      | [WPA2][Auth(802.1X)] |
| 2       | WLAN | Guest        | Guest     | Disabled     | [WPA2][Auth(802.1X)] |

Рисунок 5.63 – Інформація про створені WLAN

Щоб створити нову мережу, потрібно обрати пункт меню «Create new» (рисунок 5.64) і встановити основні параметри:

The screenshot shows the 'WLANs > New' configuration page. It has a sidebar with 'WLANs' and 'Advanced' options. The main area contains a form with fields for 'Type' (set to WLAN), 'Profile Name' (LABTEST), 'SSID' (LABTEST), and 'ID' (1). There are 'Back' and 'Apply' buttons at the top right.

Рисунок 5.64 – Створення бездротової мережі

Серед параметрів є можливість обрати такі:

- **Type** – WLAN (бездротова). Контролер також може виступати в якості порталу аутентифікації для дротової мережі (гостьової локальної мережі).
- **Profile name** – зазвичай назва та сама, що й назва мережі, використовується при використанні систем керування WCS/NCS.
- **SSID** – ім'я мережі, яке буде відображатися для клієнтських комп'ютерів.
- **ID** – за замовчуванням точки доступу анонсуватимуть мережі з номерами менше чи рівно 16.

Після створення нової мережі налаштування переходить до вікна (рисунок 5.65) із закладками, в якому вказані всі параметри її роботи:

- **Enable** – вмикає/вимикає обслуговування мережі точки доступу.
- **Security policy** – сповіщає про поточний набір мережних політик безпеки, які налаштовуються далі.
- **Radio policy** – дозволяє обрати, в якому діапазоні частот (2,4, 5 ГГц) і з якими швидкостями буде працювати мережа. Можливі комбінації варіантів. Відповідно під'єднані AP повинні підтримувати обрані діапазони.
- **Interface** – визначає, який дротовий мережний під-інтерфейс (VLAN) контролера за замовчуванням буде отримувати бездротові клієнтські підключення. Є можливість створити кілька так званих «динамічних інтерфейсів», кожен зі своїм ідентифікатором VLAN, і розподіляти між ними

користувачів залежно від того, до якої мережі вони підключені (що найчастіше використовується для надання гостьового доступу).

- **Multicast VLAN** – визначає, куди піде багатоадресний трафік у разі наявності кількох груп інтерфейсів.
- **Broadcast SSID** – визначає, чи буде ім'я мережі відображатися в маяках (анонсах) пакетів, що періодично відправляються точкою доступу (відкрита/закрита мережа).

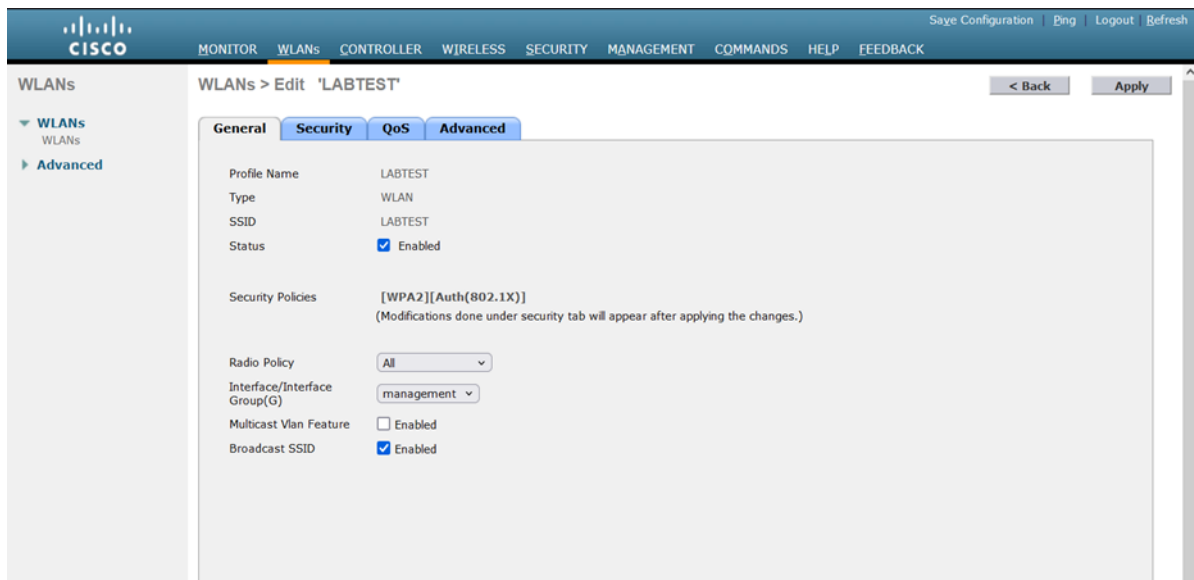


Рисунок 5.65 – Налаштування бездротової мережі

У вкладці «**Security**» є можливість налаштувати параметри безпеки бездротової мережі. Що і показано на рисунку 5.66.

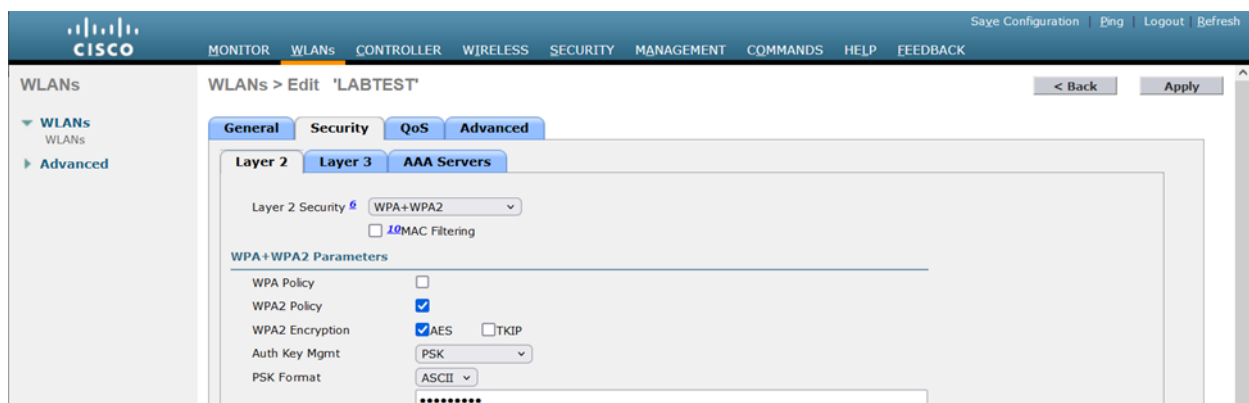


Рисунок 5.66 – Налаштування безпеки бездротової мережі

Безпека бездротової мережі складається з трьох компонентів:

- Авторизація.
- Шифрування.
- Вебполітика (необов'язково).

Перші дві політики працюють на рівні 2 моделі OSI. Авторизація відповідає за допуск користувачів у мережу. Шифрування визначає сам алгоритм шифрування пакетів в радіосередовищі. Вебполітика дозволяє розгорнути клієнтську *HTTP*-сесію на вбудованому вебсервері контролера (або зовнішньому) і запитати підтвердження/логін-пароль через форму.

Доступні опції безпеки 2-го рівня такі:

- **None** – авторизація або шифрування трафіку не застосовується (не захищена мережа). Використовується для гостьового доступу в *AP*.
- **WPA+WPA2** – дозволяє обрати політику *WPA* або *WPA2* (або обидві), тип шифрування *TKIP* або *AES* (або обидва). Ці параметри просто анонсуються клієнтам в пакетах маяків. Не всі клієнтські адаптери (особливо старі) здатні зрозуміти сучасний стандарт. Якщо всі клієнти нові, *WPA2/AES* є найкращим варіантом. Крім того, пропонується вказати, яким чином буде генеруватися ключ шифрування:

- **802.1X** – індивідуальний ключ для кожного клієнта буде згенерований сервером *RADIUS* під час авторизації. Найбезпечніший варіант, також відомий як *WPA2 Enterprise*;
- **CCMK** – використовує власний механізм генерації ключів *Cisco*, підходить лише для телефонів *Cisco Wi-Fi*;
- **PSK** – спільний (попередньо спільний) ключ, пароль для мережі, у цьому випадку називається *WPA2 Personal*;
- **802.1x+CCMK** – гібрид ключа *CCMK* і *RADIUS* (для телефонів *Cisco*).

– **802.1X** – індивідуальний ключ для кожного клієнта буде згенерований сервером *RADIUS* під час авторизації. Однак це ключ *WEP*, а протокол шифрування радіоканалу *WEP* більше не можна використовувати.

- *Static WEP* – статичний ключ *WEP*.
- *Static WEP+802.1X* є гібридом двох попередніх.
- *SKIP* – це пропрієтарна версія *WEP* для телефонів *Cisco*.

Підводячи підсумок про безпеку *L2*, необхідно відмітити, що в реальності потрібно зробити вибір між:

- Без шифрування/авторизації (гостьовий доступ).
- *WPA2 (AES) PSK (WPA2 Personal)* для доступу до мережі за допомогою спільного пароля.
- *WPA2 (AES) + 802.1X (WPA2 Enterprise)* для доступу до мережі через авторизацію на сервері *RADIUS* (*EAP*: за доменним обліковим записом, сертифікатом тощо), що є найкращою практикою в корпоративному середовищі.

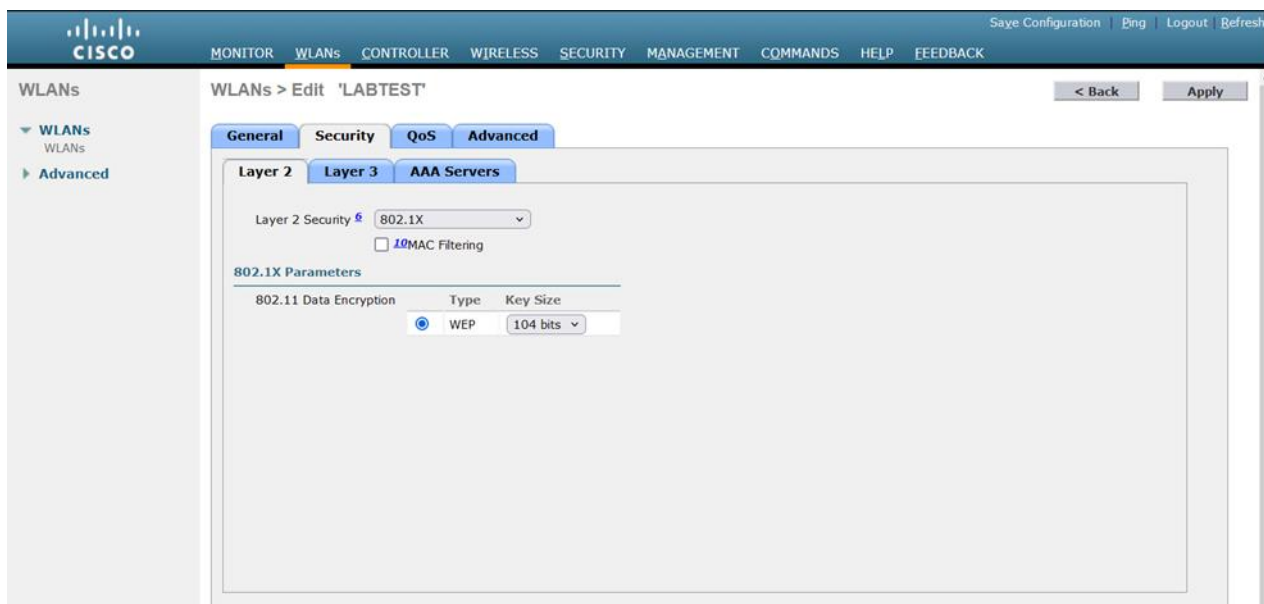


Рисунок 5.67 – Налаштування безпеки 802.1X бездротової мережі

При будь-якому варіанті конфігурації безпеки/авторизації можна увімкнути додаткову політику *L3*, яка полягає в перехопленні вебсесії клієнта:

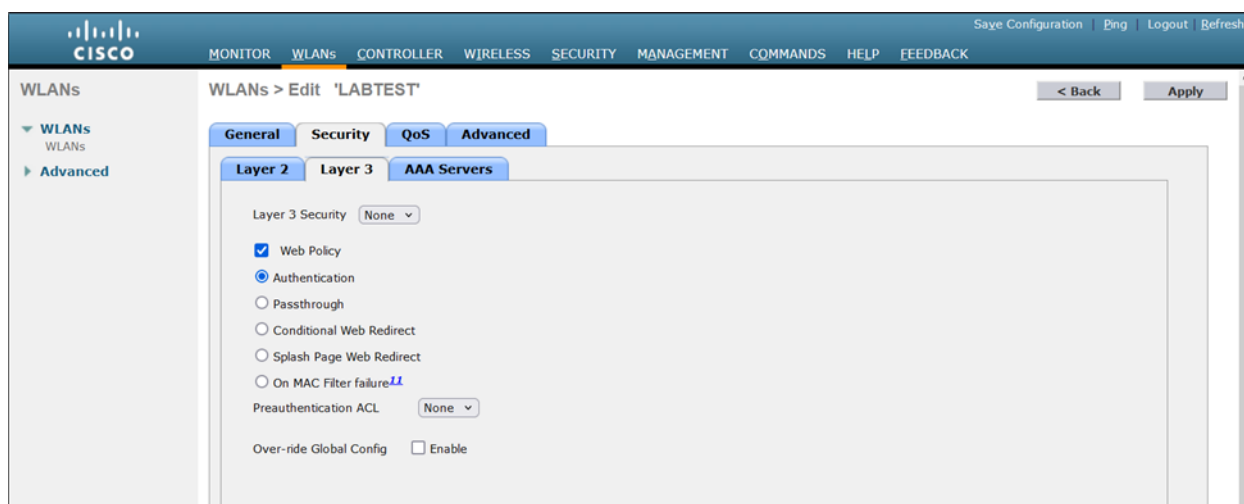


Рисунок 5.68 – Налаштування безпеки на рівні L3 бездротової мережі

Доступні такі варіанти:

- **Authentication** – користувач бачить вікно для введення логіна та пароля, які потім перевіряються на контролері (у його локальній базі даних) або на сервері *RADIUS*.
- **Passthrough** – користувач бачить вікно привітання, де у нього можуть запитати його адресу електронної пошти (вона ніде не використовується і не перевіряється).
- **Conditional Web Redirect** — дозволяє після авторизації перенаправити сесію користувача на сторінку, зазначену у відповіді *RADIUS*. Наприклад, на сторінку поповнення балансу. Після перенаправлення користувач повинен авторизуватися знову.

- **Splash Page Web Redirect** – аналогічно до попереднього, але з негайним доступом до Інтернету.
- **On MAC Filter failure** – перенаправлення відбувається, коли користувача блокує фільтр MAC-адрес.

За бажанням можна вказати *ACL* (список доступу) для користувачів, які не пройшли аутентифікацію (наприклад, для *DNS*-сервера або зовнішнього вебсервера з логотипом). Також існує можливість обрати, яку сторінку (форму) показувати користувачеві під час авторизації (стандартну, модифіковану стандартну або розташовану на зовнішньому вебсервері). Якщо в мережі використовується сервер *RADIUS*, то потрібно виконати додаткові налаштування. Перш за все, налаштувати сервер авторизації в меню **Security->RADIUS->AAA->Authentication**, що показано на рисунку 5.69:

Рисунок 5.69 – Налаштування автентифікації на сервері *RADIUS*

Необхідно вказати такі параметри:

- **Server Address** – IP-адреса сервера. Підтримуються *FreeRADIUS*, *Cisco ACS*, *Cisco ISE*, *Microsoft Server*.
- **Shared secret** – інформація про ключ сервера *RADIUS*.
- **Network user** – увімкнення підтримки авторизації користувачів мережі WI-FI.
- **Management** – увімкнення підтримки авторизації адміністраторів самого контролера.

Також необхідно вказувати один і той же сервер для аудиту в розділі **Accounting**.

У налаштуваннях безпеки бездротової мережі у вкладці **AAA Servers** всі опції, зазначені за замовчуванням, забезпечують роботу всіх серверів *RADIUS*, зареєстрованих на контролері:

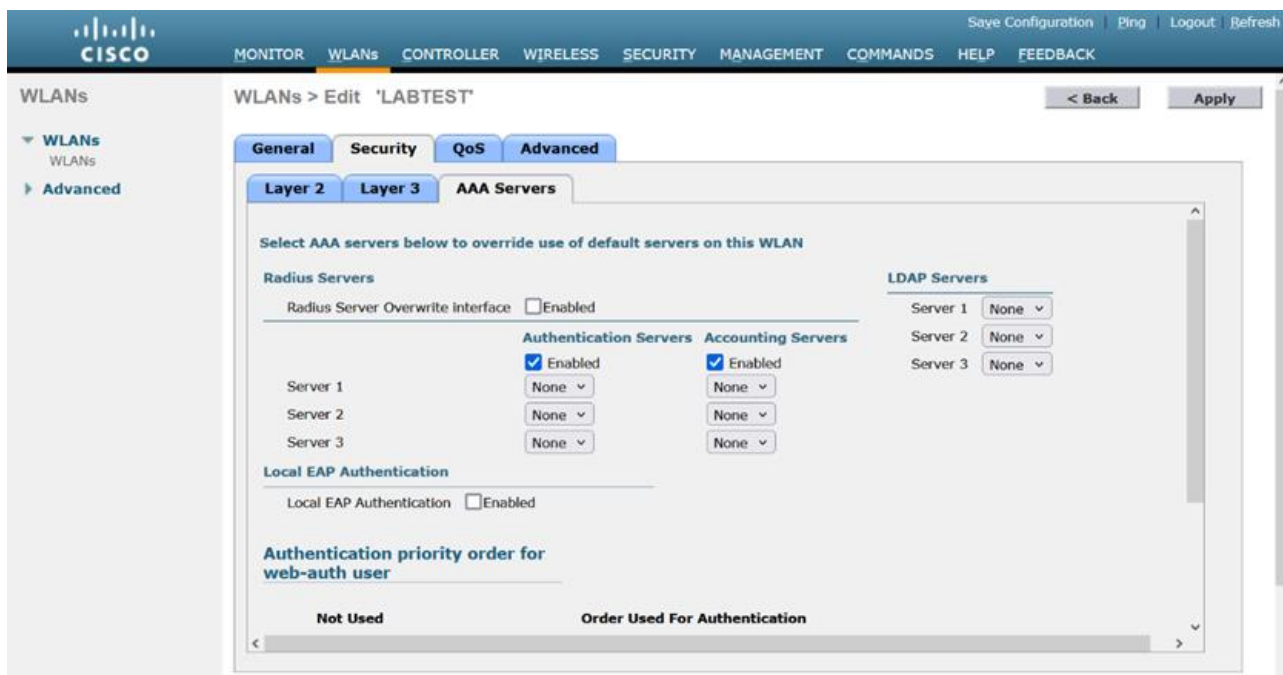


Рисунок 5.70 – Налаштування безпеки у вкладці AAA Servers

Також можна призначити окремий сервер для цієї бездротової мережі, виключити аудит, увімкнути вбудований міні-сервер *RADIUS* в контролері тощо.

Вкладка *QoS* відповідає за параметри *QoS* у мережі, пріоритезуючи різні типи трафіку та користувачів. Дані опції потребують налаштування, якщо у відповідній мережі наявне широке використання голосу, відео, багато гостей користувачів під великим навантаженням тощо.

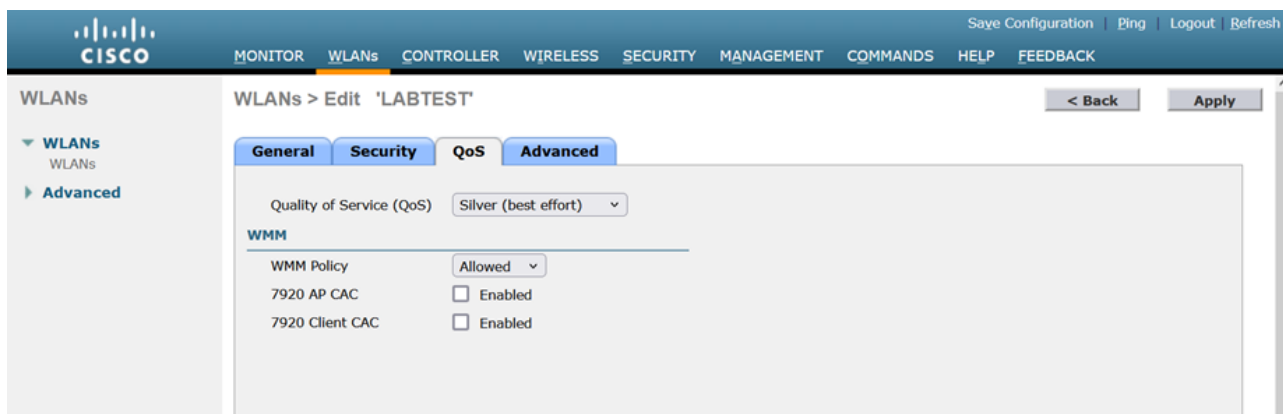


Рисунок 5.71 – Налаштування параметрів QoS

Остання вкладка, «**Advanced**», описує різні додаткові параметри бездротової мережі, яких існує досить багато.

**Allow AAA Override** дозволяє передавати додаткові параметри з сервера *RADIUS* в момент успішної авторизації клієнта, і застосовувати їх до цього клієнта індивідуально. Такими параметрами можуть бути номер *VLAN*, ім'я локального інтерфейсу, список доступу (*ACL*), *URL* для перенаправлення, політика *QoS* тощо.

**Coverage Hole Detection** керує механізмом виявлення та компенсації області недостатнього покриття для клієнтів у даній бездротовій мережі. Рекомендовано вимкнути для гостей бездротових локальних мереж.

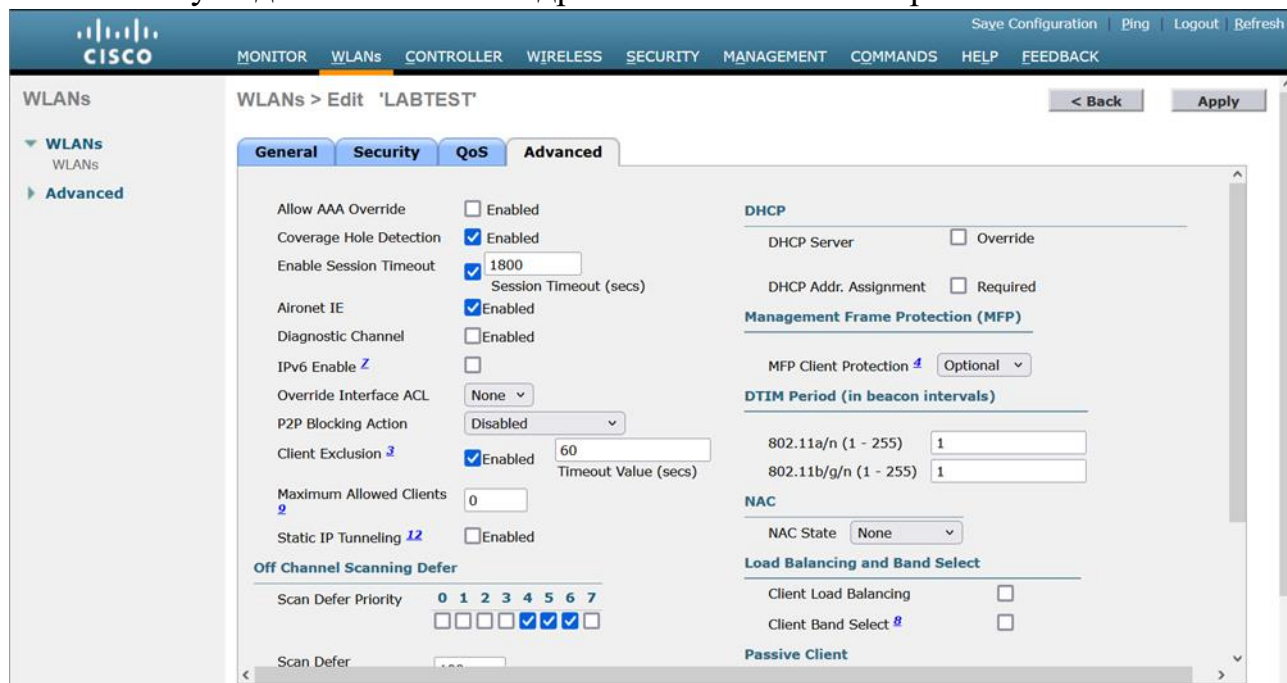


Рисунок 5.72 – Налаштування параметрів у вкладці Advanced

**Enable Session Timeout, Session Timeout (secs)** вмикає та визначає час очікування сеансу клієнта для вебавторизації.

**Aironet IE** включає специфічні для Cisco розширення параметрів рамок маяків. При цьому смарт-клієнтські адаптери працюють краще (роумінг, енергозбереження), а інші можуть взагалі не працювати в такій мережі.

**Diagnostic Chanel** активує додатковий канал діагностичної логіки для клієнтських адаптерів, сумісних із CCX5.

**IPv6** насправді опція дозволяє лише IPv6-трафік для вебавторизації.

**Override Interface ACL** дозволяє задати альтернативний список доступу (ACL) замість заданого на дротовому VLAN (керуючому, динамічному) інтерфейсі контролера.

**P2P Blocking Action** визначає політику передачі трафіку між бездротовими клієнтами (всередині контролера). Допустимі значення: Disabled, Drop, Forward-UpStream (відправити на маршрутизатор, нехай вирішить).

**Client Exclusion, Timeout Value (secs)** вмикає та встановлює час очікування виключення (часового блоку) клієнта, авторизація якого у бездротовій мережі не пройшла.

**Maximum Allowed Clients** встановлює максимальну кількість одночасних асоціацій з даною мережею.

**Static IP Tunneling** дозволяє роумінг між контролерами клієнтам зі статичною IP-адресою.

**Off Channel Scanning Defer, Scan Defer Priority** кожна точка іноді виходить зі свого робочого каналу (частоти) і прослуховує інші канали сусідніх мереж, завантаженість спектру, абонентів тощо. У той же час такий стрибок може негативно позначитися на голосовому трафіку, що передається до цього моменту. Якщо точка передає пакети зі значенням поля *802.1p*, позначеним (0 1 2 3 4 5 6 7), тоді стрибок з робочої частоти буде відкладений.

**Scan Defer Time (мс)** на скільки мілісекунд затримується стрибок на іншу частоту за попередньою опцією.

**H-REAP Local Switching** дозволяє точці доступу, яка знаходиться в режимі *H-REAP* (віддалений офіс), замикає трафік абонента локально при обслуговуванні даної бездротової мережі, а не перенаправляти його на контролер в тунелі *CAPWAP*.

**H-REAP Local Auth** дозволяє точці доступу, яка знаходиться в режимі *H-REAP* (віддалений офіс) при обслуговуванні даної бездротової мережі, авторизуватися локально, а не на контролері.

**Learn Client IP Address** у режимі *H-REAP* точка повідомить *IP*-адресу клієнта контролеру, якщо така доступна.

**DHCP Server Override, DHCP Server IP Addr** використовувати вказану *IP*-адресу *DHCP*-сервера замість зазначеної в налаштуваннях дротового інтерфейсу контролера, до якої направляється запити бездротових клієнтів.

**DHCP Addr. Assignment** вимагати від клієнтів у цій бездротовій мережі використовувати *DHCP*-сервер (статично налаштовані клієнти не працюватимуть).

**MFP Client Protection** увімкнення та обов'язковий захист кадру клієнта, **Disabled** –вимкнено, **Optional** – необов'язково та **Required** – обов'язково (повинна бути підтримка *CCX5* на стороні клієнта).

**DTIM Period (in beacon intervals)** частота передачі широкомовних/багатоадресних кадрів, впливає на енергоефективність клієнтів.

**NAC State** дозволяє обрати режим роботи спільно з пристроєм *NAC*.

**Client Load Balancing** дозволяє збалансувати клієнтів між точками доступу на основі їх навантаження.

**Client Band Select** дозволяє переводити клієнтів у діапазон 5 ГГц, що є кращим через низьку завантаженість.

**Passive Client** дозволяє клієнтські пристрої, які на навантажують канал (наприклад, ваги *Wi-Fi*).

**Media Session Snooping** дозволяє перехоплювати *SIP*-телефонні сесії.

**Re-anchor Roamed Voice Clients** дозволяє примусово передавати дані між контролерами голосових клієнтів, які знаходяться в роумінгу.

Після завершення всіх налаштувань потрібно натиснути кнопку **Apply** та зберегти конфігурацію контролера.

## ПРАКТИЧНЕ ЗАВДАННЯ

**Мета роботи:** ознайомитися з технологіями побудови та захисту корпоративних бездротових мережі на основі контролера, а саме здійснити налаштування контролера, під'єднання бездротових точок доступу та створення бездротових мереж з відповідним рівнем захисту доступу за допомогою бездротового обладнання *Cisco*.

### Порядок виконання роботи

1. Створити фізичний проєкт мережі або у середовищі програмного симулятора/емулятора створити мережу (рисунок 5.73). При побудові звернути увагу на вибір бездротового контролера, комутатора, точки доступу, мережних модулів та плат, а також мережних з'єднань. При побудові підмережі слід вибирати потрібний тип кабелю для відповідної технології.

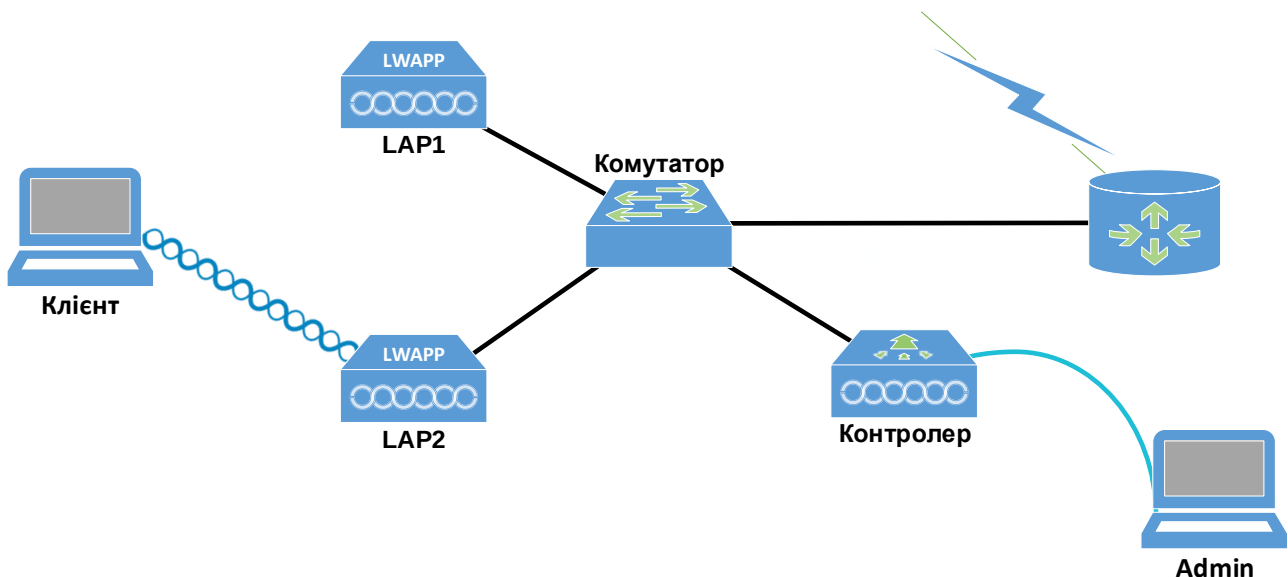


Рисунок 5.73 – Проєкт мережі

2. Після під'єднання контролера до живлення та до його консольного порту з комп'ютера **Admin** зробити повне скидання контролера та провести процедуру його початкового налаштування згідно варіанту таблиці 5.55.

3. Під'єднати точку доступу за допомогою кабелю до порту з подачею живлення через *Ethernet (PoE)*, перевірити світлодіодні індикатори контролера на наявність з'єднання та індикатор на точці доступу. Дочекатися приєднання точки доступу до контролера, в іншому випадку вирішити проблеми приєднання шляхом додавання сертифікатів точки доступу для приймання контролером.

4. Налаштувати на контролері дві бездротові мережі: внутрішню та гостю з вказанням відповідних параметрів при створенні. Налаштувати безпеку 2-го рівня бездротових мереж з використанням типу *WPA2 Personal* чи *WPA2 Enterprise* за наявності в мережі налаштованого *RADIUS*-сервера.

5. Під'єднати до внутрішньої мережі як мінімум два бездротових пристрої з використанням налаштованих параметрів безпеки.

Таблиця 5.55 – Параметри налаштування бездротової мережі

| <b>№ варіанта</b> | <b>Локальна мережа</b> | <b>IP-адреса інтерфейсу керування</b> | <b>IP-адреса основного шлюзу</b> | <b>IP-адреса інтерфейсу керування AP</b> | <b>Стандарт мережі</b> |
|-------------------|------------------------|---------------------------------------|----------------------------------|------------------------------------------|------------------------|
| 1                 | 192.168.1.0/24         | 192.168.1.11/24                       | 192.168.1.1/24                   | 192.168.1.12/24                          | a/n                    |
| 2                 | 192.168.2.0/24         | 192.168.2.11/24                       | 192.168.2.1/24                   | 192.168.2.12/24                          | a/n, b/g/n             |
| 3                 | 192.168.3.0/24         | 192.168.3.11/24                       | 192.168.3.1/24                   | 192.168.3.12/24                          | b/g/n                  |
| 4                 | 192.168.4.0/24         | 192.168.4.11/24                       | 192.168.4.1/24                   | 192.168.4.12/24                          | a/n                    |
| 5                 | 192.168.5.0/24         | 192.168.5.11/24                       | 192.168.5.1/24                   | 192.168.5.12/24                          | a/n, b/g/n             |
| 6                 | 192.168.6.0/24         | 192.168.6.11/24                       | 192.168.6.1/24                   | 192.168.6.12/24                          | b/g/n                  |
| 7                 | 192.168.7.0/24         | 192.168.7.11/24                       | 192.168.7.1/24                   | 192.168.7.12/24                          | a/n                    |
| 8                 | 192.168.8.0/24         | 192.168.8.11/24                       | 192.168.8.1/24                   | 192.168.8.12/24                          | a/n, b/g/n             |
| 9                 | 192.168.9.0/24         | 192.168.9.11/24                       | 192.168.9.1/24                   | 192.168.9.12/24                          | b/g/n                  |
| 10                | 192.168.10.0/24        | 192.168.10.11/24                      | 192.168.10.1/24                  | 192.168.10.1/24                          | a/n                    |
| 11                | 192.168.11.0/24        | 192.168.11.11/24                      | 192.168.11.1/24                  | 192.168.11.12/24                         | a/n, b/g/n             |
| 12                | 192.168.12.0/24        | 192.168.12.11/24                      | 192.168.12.1/24                  | 192.168.12.12/24                         | b/g/n                  |
| 13                | 192.168.13.0/24        | 192.168.13.11/24                      | 192.168.13.1/24                  | 192.168.13.12/24                         | a/n                    |
| 14                | 192.168.14.0/24        | 192.168.14.11/24                      | 192.168.14.1/24                  | 192.168.14.12/24                         | a/n, b/g/n             |
| 15                | 192.168.15.0/24        | 192.168.15.11/24                      | 192.168.15.1/24                  | 192.168.15.12/24                         | b/g/n                  |
| 16                | 192.168.16.0/24        | 192.168.16.11/24                      | 192.168.16.1/24                  | 192.168.16.12/24                         | a/n                    |
| 17                | 192.168.17.0/24        | 192.168.17.11/24                      | 192.168.17.1/24                  | 192.168.17.12/24                         | a/n, b/g/n             |
| 18                | 192.168.18.0/24        | 192.168.18.11/24                      | 192.168.18.1/24                  | 192.168.18.12/24                         | b/g/n                  |
| 19                | 192.168.19.0/24        | 192.168.19.11/24                      | 192.168.19.1/24                  | 192.168.19.12/24                         | a/n                    |
| 20                | 192.168.20.0/24        | 192.168.20.11/24                      | 192.168.20.1/24                  | 192.168.20.12/24                         | a/n, b/g/n             |
| 21                | 192.168.21.0/24        | 192.168.21.11/24                      | 192.168.21.1/24                  | 192.168.21.12/24                         | b/g/n                  |
| 22                | 192.168.22.0/24        | 192.168.22.11/24                      | 192.168.22.1/24                  | 192.168.22.12/24                         | a/n                    |
| 23                | 192.168.23.0/24        | 192.168.23.11/24                      | 192.168.23.1/24                  | 192.168.23.12/24                         | a/n, b/g/n             |
| 24                | 192.168.24.0/24        | 192.168.24.11/24                      | 192.168.24.1/24                  | 192.168.24.12/24                         | b/g/n                  |
| 25                | 192.168.25.0/24        | 192.168.25.11/24                      | 192.168.25.1/24                  | 192.168.25.12/24                         | a/n                    |
| 26                | 192.168.26.0/24        | 192.168.26.11/24                      | 192.168.26.1/24                  | 192.168.26.12/24                         | a/n, b/g/n             |
| 27                | 192.168.27.0/24        | 192.168.27.11/24                      | 192.168.27.1/24                  | 192.168.27.12/24                         | b/g/n                  |
| 28                | 192.168.28.0/24        | 192.168.28.11/24                      | 192.168.28.1/24                  | 192.168.28.12/24                         | a/n                    |
| 29                | 192.168.29.0/24        | 192.168.29.11/24                      | 192.168.29.1/24                  | 192.168.29.12/24                         | a/n, b/g/n             |
| 30                | 192.168.30.0/24        | 192.168.30.11/24                      | 192.168.30.1/24                  | 192.168.30.12/24                         | b/g/n                  |

6. Перевірити налаштування мережного з'єднання, отримання *IP*-адреси пристроями та наскрізне з'єднання між пристроями у внутрішній мережі за допомогою команди **ping**.

7. Під'єднати до гостьової мережі як мінімум два бездротових пристрої з використанням налаштованих параметрів безпеки.

8. Перевірити налаштування мережного з'єднання, отримання *IP*-адреси пристроями та наскрізне з'єднання між пристроями у гостьовій мережі за допомогою команди **ping**.

9. Перевірити наскрізне з'єднання між пристроями у гостьовій мережі та внутрішній за допомогою команди **ping**.

### КОНТРОЛЬНІ ПИТАННЯ

1. Які основні функції виконує контролер бездротової мережі?
2. Для чого призначені логічні інтерфейси *WLC Cisco*?
3. Перелічіть основні *IP*-адреси, які використовуються для початкового налаштування *WLC*?
4. Поясніть призначення протоколу *CAPWAP*?
5. Які відмінності в роботі існують між полегшеною точкою доступу *Cisco* і звичайною?
6. Опишіть процес під'єднання точки доступу до *WLC*?
7. За якими типами сертифікатів *AP* можуть приєднуватися до *WLC*?
8. Чим відрізняються типи безпеки на рівні 2 *WPA2 Enterprise* та *WPA2 Personal*?
9. Поясніть, чому ключ *WEP* не можна використовувати для захисту бездротової мережі?
10. Які основні переваги використання сервера *RADIUS* для авторизації клієнтів бездротової мережі?

## 5.2.8 БАЗОВІ НАЛАШТУВАННЯ МАРШРУТИЗАТОРА ТА СТАТИЧНИХ МАРШРУТІВ

### ОСНОВНІ КОМАНДИ НАЛАГОДЖЕННЯ ПАРАМЕТРІВ ІНТЕРФЕЙСІВ МАРШРУТИЗАТОРА CISCO

Вибір інтерфейсу маршрутизатора для налагодження виконується командою **interface**. Можливе одночасне налагодження групи інтерфейсів. Для цього використовується команда **interface range**. Слід нагадати, що за замовчуванням фізичні інтерфейси маршрутизатора знаходяться у відключеному стані, а логічні інтерфейси залежно від типу можуть знаходитися як у відключеному, так і в включеному станах. Відключення інтерфейсу виконується командою **shutdown**, включення – командою **no shutdown**. Для налагодження параметрів інтерфейсів маршрутизатора, залежно від їх типу, використовується досить великий набір команд. Більшість команд є загальними для всіх інтерфейсів, частина – характерними лише для інтерфейсів певних технологій.

Основними командами налаштування параметрів фізичного і каналного рівня для інтерфейсів маршрутизатора є такі команди: **arp**, **bandwidth**, **clock rate**, **delay**, **description**, **duplex**, **encapsulation**, **keepalive**, **ip**, **mac-address**, **mtu**, **speed**. Відміна дії команд – використання форми **no**, або команда **default**.

Команда **arp** та її модифікації служать для обробки ARP-запитів та їх параметрів на інтерфейсі. Команда **bandwidth** служить для встановлення значення пропускної здатності, що використовується при обчисленні метрик маршрутів у протоколах маршрутизації, не встановлює швидкість передачі даних інтерфейсу і не впливає на фактичну швидкість передачі даних по каналу зв'язку. Команда **clock rate** служить для налаштування частоти тактових імпульсів на одному з пари інтерфейсів (типу *DCE*), що формують прямий двоточковий послідовний канал між двома маршрутизаторами (з'єднання типу нуль-модем). При підключенні маршрутизатора через *DCE*-пристрій (наприклад, *CSU/DSU*) команда не задається, оскільки синхронізація здійснюється провайдером послуг. Команда **delay** служить для встановлення значення затримки на інтерфейсі, це значення використовується при обчисленні метрик у деяких протоколах маршрутизації, команда не визначає параметрів інтерфейсу. Команда **description** служить для опису інтерфейсу, використовується з метою полегшення аналізу результатів виводу команд при адмініструванні. Команда **duplex** (та її модифікації **duplex-full**, **duplex-half**) служать для зазначення режиму передачі даних на інтерфейсі. Команда **encapsulation** служить для налаштування типу інкапсуляції на інтерфейсі. Часто використовується на послідовних інтерфейсах для зазначення протоколу або технології каналного рівня, на інтерфейсах *Ethernet* використовується для тегування VLAN (як *802.1Q*, так і *ISL*).

Команда **keepalive** служить для зазначення інтервалу, протягом якого маршрутизатор буде очікувати перед тим, як відправити через інтерфейс повідомлення про перевірку зв'язку для визначення чи працює інтерфейс на іншому кінці послідовного каналу. На *Ethernet*-інтерфейсах маршрутизатор пересилає повідомлення самому собі. Команда **mtu** служить для зазначення *MTU* інтерфейсу, це значення варто змінювати для оптимізації продуктивності мережі, наприклад, для каналів з великими втратами його варто зменшувати.

Синтаксис команди **interface** (режим глобального конфігурування).

```
interface interface-type interface-id.subinterface-id
[{{point-to-point | multipoint}}]
```

де **interface-type** – тип інтерфейса, може приймати значення **Ethernet**, **FastEthernet**, **Serial**, **ATM**, **Loopback**, **Tunnel**, **Vlan** та ін.;

**interface-id** – ідентифікатор інтерфейса, може мати одночислове позначення **number** (номер інтерфейса), двочислове позначення **module/number** (номер модуля (адаптера)/номер інтерфейса), тричислове позначення **slot/module/number** (номер слота/номер модуля(адаптера)/ номер інтерфейса);

**subinterface-id** – ідентифікатор підінтерфейса, може приймати значення від 0 до 4294967295, за замовчуванням інтерфейс не містить підінтерфейсів, вони створюються у процесі виконання команди **interface**; підінтерфейси використовуються для забезпечення роботи протоколу *802.1Q* та технологій *Frame Relay* і *ATM*;

**point-to-point** – службова конструкція, яка зазначає, що підінтерфейс логічно з'єднаний з одним віддаленим вузлом;

**multipoint** – службова конструкція, яка зазначає, що підінтерфейс логічно з'єднаний з кількома віддаленими вузлами;

Параметри **point-to-point** та **multipoint**, як правило, зазначаються при роботі з інтерфейсами *Frame Relay* і *ATM*.

Синтаксис команди **arp** (режим конфігурування інтерфейсу).

```
arp {arpa | frame-relay | probe | snap }
```

де **arpa** – інкапсуляція для мереж *Ethernet*, встановлюється за замовчуванням;

**frame-relay** – інкапсуляція для мереж *Frame Relay*;

**probe** – інкапсуляція для протоколу *HP Probe*;

**snap** – інкапсуляція для *SNAP* (згідно *RFC 1042*).

Синтаксис команди **arp timeout** (режим конфігурування інтерфейсу).

```
arp timeout seconds
```

де **seconds** – час життя *ARP*-запису в *ARP*-таблиці (с), за замовчуванням дорівнює **14400**.

Синтаксис команди **bandwidth** (режим конфігурування інтерфейсу).

**bandwidth** *value*

де **value** – значення пропускної здатності в **Kbit/c**, за замовчуванням залежить від типу інтерфейсу.

Синтаксис команди **clock rate** (режим конфігурування інтерфейсу).

**clock rate** *bps*

де **bps** – значення частоти тактових імпульсів (**bit/c**), може приймати значення 1200, 2400, 4800, 9600, 19 200, 38400, 56000, 64000, 72000, 125000, 148000, 500000, 800000, 1000000, 1300000, 2000000, 4000000; за замовчуванням не зазначається.

Синтаксис команди **delay** (режим конфігурування інтерфейсу).

**delay** *value*

де **value** – значення затримки на інтерфейсі в десятках мілісекунд, за замовчуванням залежить від типу інтерфейсу.

Синтаксис команди **description** (режим конфігурування інтерфейсу).

**description** *text-line*

де **text-line** – тестовий рядок опису інтерфейсу (до 240 символів).

Синтаксис команди **duplex** (режим конфігурування інтерфейсу).

**duplex** {auto | full | half}

де **auto** – автоматичний вибір режиму;

**full** – повнодуплексний режим;

**half** – напівдуплексний режим.

Синтаксис команди **encapsulation** (режим конфігурування інтерфейсу/підінтерфейсу).

**encapsulation** { ppp | hdlc | frame-relay [cisco | ietf] |  
dot1q *vlan-id* ... }

де **ppp** – службова конструкція, яка вказує, що інкапсуляцію здійснювати згідно стандартів протоколу *PPP*;

**hdlc** – службова конструкція, яка вказує, що інкапсуляцію здійснювати згідно стандартів протоколу *HDLC*;

**frame-relay** – службова конструкція, яка вказує, що інкапсуляцію здійснювати згідно стандартів технології *Frame Relay*;

**cisco** – фірмовий спосіб інкапсуляції *Cisco* для *Frame Relay*;

**ietf** – стандартний спосіб інкапсуляції *IETF*;

**dot1q** – інкапсуляція по протоколу 802.1Q;

**vlan-id** – номер VLAN в діапазоні від 1 до 1005 при використанні стандартного образу IOS, при використанні образу з розширеними можливостями – в діапазоні від 1 до 4094.

Синтаксис команди **keepalive** (режим конфігурування інтерфейсу).

**keepalive** *seconds*

де **seconds** – значення інтервалу часу очікування, яке задається в секундах, за замовчуванням становить 10 с.

Синтаксис команди **ip address** (режим конфігурування інтерфейсу).

**ip address** {*address network\_mask*} | **dhcp**

де **address** – IP-адреса в десятковому записі;

**network\_mask** – маска мережі, записана у звичайній формі;

**dhcp** – службова конструкція, яка вказує, що IP-адресу необхідно отримати автоматично по протоколу DHCP.

Синтаксис команди **mac-address** (режим конфігурування інтерфейсу).

**mac-address** *hw-address*

де **hw-address** – MAC-адреса інтерфейсу у вигляді **HHHH.HHHH.HHHH**, кожне число **HHHH** має довжину 2 байти і записується в шістнадцятковій формі.

Синтаксис команди **mtu** (режим конфігурування інтерфейсу).

**mtu** *value*

де **value** – значення MTU в байтах, значення за замовчуванням залежить від технології або протоколу канального рівня.

Синтаксис команди **speed** (режим конфігурування інтерфейсу).

**speed** {10 | 100 | 1000 | auto [10 | 100 | 1000]  
| **nonegotiate**}

де 10, 100, 1000 – значення швидкості в Мбіт/с,

**auto** – службова конструкція, яка вказує автоматичний вибір швидкості; якщо використовується форма **auto 10** (**auto 100**, **auto 1000**) інтерфейс веде переговори лише на цій швидкості;

**nonegotiate** – службова конструкція, яка відключає режим автопереговорів про швидкість.

Синтаксис команди **config-register** (режим глобального конфігурування):

**config-register** *conf\_reg\_value*

де **conf\_reg\_value** – значення конфігураційного регістру, число з діапазону 0x0000 ... 0xFFFF; за замовчуванням становить 0x2102.

## ОСНОВНІ КОМАНДИ *CISCO IOS* ДЛЯ БАЗОВОЇ ДІАГНОСТИКИ РОБОТИ МАРШРУТИЗАТОРА *CISCO*

Для виведення діагностичної інформації про фізичні параметри маршрутизатора чи його інтерфейсів, стан маршрутизатора, результати налагоджень або результати роботи маршрутизатора тощо використовується команда **show**. Вона є доступною як із режиму користувача, так і з привілейованого режиму. Залежно від режиму дана команда може мати різні параметри. Частина параметрів є однаковими і доступними в обох режимах. Часто команда **show** із певним параметром вважається окремою командою. Перелік основних команд **show** та їх призначення наведені у таблиці 5.56.

Таблиця 5.56 – Перелік основних параметрів команди **show**

| Команда                          | Призначення                                                                                                        |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------|
| <b>show version</b>              | Виведення поточної інформації про апаратне і програмне забезпечення                                                |
| <b>show tech-support</b>         | Виведення системної інформації для технічної підтримки                                                             |
| <b>show flash</b>                | Перегляд вмісту флеш-пам'яті                                                                                       |
| <b>show file systems</b>         | Виведення про файлову систему                                                                                      |
| <b>show memory</b>               | Виведення інформації про використання пам'яті                                                                      |
| <b>show processes</b>            | Виведення інформації про процеси, запущені на пристрої                                                             |
| <b>show processes cpu</b>        | Виведення деталізованої інформації про завантаження процесора                                                      |
| <b>show processes memory</b>     | Виведення інформації про завантаження процесами оперативної пам'яті                                                |
| <b>show controllers</b>          | Виведення деталізованої інформації про роботу контролера інтерфейсу                                                |
| <b>show interfaces</b>           | Виведення деталізованої інформації про інтерфейси маршрутизатора та їх стан                                        |
| <b>show ip interface</b>         | Виведення інформації про функціонування протоколу IP версії 4 та суміжних протоколів                               |
| <b>show ip interface brief</b>   | Виведення інформації про функціонування протоколу IP версії 4 на інтерфейсі у скороченому вигляді                  |
| <b>show ipv6 interface</b>       | Виведення інформації про функціонування протоколу IP версії 6 та суміжних протоколів                               |
| <b>show ipv6 interface brief</b> | Виведення інформації про функціонування протоколу IP версії 6 на інтерфейсі у скороченому вигляді                  |
| <b>show route</b>                | Виведення таблиці маршрутизації протоколу IP версії 4                                                              |
| <b>Show ipv6 route</b>           | Виведення таблиці маршрутизації протоколу IP версії 6                                                              |
| <b>show protocols</b>            | Виведення глобальної та інтерфейснозалежної інформації про протоколи 3-го рівня, що функціонують на маршрутизаторі |
| <b>show startup-config</b>       | Перегляд стартової конфігурації пристрою                                                                           |
| <b>show running-config</b>       | Перегляд поточної конфігурації пристрою                                                                            |
| <b>show history</b>              | Перегляд списку останніх виконаних команд (за замовчуванням 10 рядків)                                             |
| <b>show clock</b>                | Виведення часу, встановленого на маршрутизаторі                                                                    |

## ОСНОВНІ КОМАНДИ НАЛАГОДЖЕННЯ СТАТИЧНИХ МАРШРУТІВ НА МАРШРУТИЗАТОРАХ CISCO

Статичні маршрути використовуються з різних причин і часто використовуються, коли немає динамічного маршруту до *IP*-адреси призначення або для заміни динамічно отриманого маршруту. За замовчуванням статичні маршрути мають адміністративну відстань, що дорівнює одиниці, що надає їм пріоритет над будь-яким протоколом динамічної маршрутизації. Коли адміністративна відстань збільшується до значення, що перевищує значення протоколу динамічної маршрутизації, статичний маршрут може бути запобіжним засобом у разі збою динамічної маршрутизації.

Щоб встановити статичні маршрути *IPv4*, використовуйте команду глобальної конфігурації **ip route**:

```
ip route network-address subnet-mask {ip-address | exit-
intf [ip-address]} [distance]
```

де **network-address** – визначає *IPv4*-адресу призначення віддаленої мережі, яку необхідно додати до таблиці маршрутизації;

**subnet-mask** – визначає маску підмережі віддаленої мережі.

**ip-address** – визначає адресу *IPv4* маршрутизатора наступного переходу. Зазвичай використовується у ширококомовних (*broadcast*) мережах (наприклад, *Ethernet*). Може створити рекурсивний статичний маршрут, де маршрутизатор виконує додатковий пошук для знаходження вихідного інтерфейсу.

**exit-intf** – визначає вихідний інтерфейс для пересилання пакетів. Створює безпосередньо під'єднаний статичний маршрут. Зазвичай використовується у конфігурації *point-to-point* («точка-точка»).

**exit-intf ip-address** – створює повністю заданий статичний маршрут, оскільки визначає вихідний інтерфейс і адресу наступного переходу *IPv4*.

**distance** – необов'язковий параметр, який можна використовувати для призначення адміністративної відстані (*administrative distance*) від 1 до 255.

Щоб видалити статичні маршрути, використовуйте форму **no** цієї команди.

Статичні маршрути *IPv6* налаштовують за допомогою наступної команди глобальної конфігурації:

```
ipv6 route ipv6-prefix/prefix-length {ipv6-address |
exit-intf [ipv6-address]} [distance]
```

В *IOS Cisco* передбачені такі основні команди для усунення неполадок зі статичними маршрутами: **ping**, **traceroute**, **show ip route**, **show ip interface brief**, **show cdp neighbors detail**.

## Модельний приклад налагодження параметрів каналу зв'язку технології *Fast Ethernet*, побудованого між маршрутизаторами *Cisco*

Розглянемо специфіку налагодження параметрів каналу зв'язку технології *Fast Ethernet* між маршрутизаторами *Cisco* для з'єднання, схема якої наведена на рисунку 5.74.

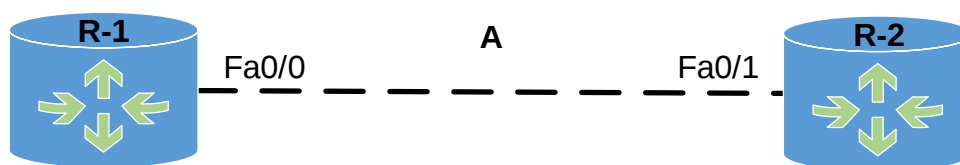


Рисунок 5.74 – Приклад мережі

Під час побудови каналу зв'язку для з'єднання пристроїв використано дані таблиці 5.57.

Таблиця 5.57 – Параметри інтерфейсів пристроїв для прикладу

| Пристрій          | Інтерфейс | Підключення до пристрою | Підключення до інтерфейсу |
|-------------------|-----------|-------------------------|---------------------------|
| Маршрутизатор R-1 | Fa0/0     | Маршрутизатор R-2       | Fa0/1                     |
| Маршрутизатор R-2 | Fa0/1     | Маршрутизатор R-1       | Fa0/0                     |

Для налагодження параметрів адресації інтерфейсів пристроїв використано дані таблиці 5.58.

Таблиця 5.58 – Параметри адресації мережі

| Підмережа/Пристрій | Інтерфейс/Мережний адаптер/Шлюз | IP-адреса | Маска підмережі | Префікс |
|--------------------|---------------------------------|-----------|-----------------|---------|
| Підмережа A        | —                               | 195.1.1.0 | 255.255.255.252 | /30     |
| Маршрутизатор R-1  | Інтерфейс Fa0/0                 | 195.1.1.1 | 255.255.255.252 | /30     |
| Маршрутизатор R-2  | Інтерфейс Fa0/1                 | 195.1.1.2 | 255.255.255.252 | /30     |

Сценарії налагодження параметрів інтерфейсів технології *Fast Ethernet* (швидкість, режим роботи, *MAC*-адреса) та параметрів *IP*-адресації для маршрутизаторів R-1, R-2 наведені нижче:

```

Router>enable
Router#configure terminal
Router(config)#hostname R-1
R-1(config)#interface FastEthernet 0/0
R-1(config-if)#description LINK-TO-R-2
R-1(config-if)#speed 100
R-1(config-if)#duplex full
R-1(config-if)#mac-address 00aa.00ad.0001

```

```

R-1(config-if)#ip address 195.1.1.1 255.255.255.252
R-1(config-if)#no shutdown
R-1(config-if)#exit
R-1(config)#exit
Router>enable
Router#configure terminal
Router(config)#hostname R-2
R-2(config)#interface FastEthernet 0/1
R-2(config-if)#description LINK-TO-R-1
R-2(config-if)#speed 100
R-2(config-if)#duplex full
R-2(config-if)#mac-address 00aa.00ad.0002
R-2(config-if)#ip address 195.1.1.2 255.255.255.252
R-2(config-if)#no shutdown
R-2(config-if)#exit
R-2(config)#exit

```

### Результати виконання команд моніторингу та діагностики роботи інтерфейсів маршрутизаторів для розглянутого прикладу

З метою перегляду інформації про функціонування інтерфейсів маршрутизаторів для розглянутого прикладу використано команди **show interfaces**, **show ip interface brief**. Перевірка зв'язку між маршрутизаторами здійснена за допомогою команди **ping**. Результати роботи цих команд для маршрутизаторів **R-1** та **R-2** наведено відповідно далі:

```

R-1#show interfaces FastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
 Hardware is Gt96k FE, address is 00aa.00ad.0001 (bia c403.0755.0000)
 Description: LINK-TO-R-2
 Internet address is 195.1.1.1/30
 MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
 reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation ARPA, loopback not set
 Keepalive set (10 sec)
 Full-duplex, 100Mb/s, 100BaseTX/FX
 ARP type: ARPA, ARP Timeout 04:00:00
 Last input 00:00:30, output 00:00:03, output hang never
 Last clearing of "show interface" counters never
 Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
 Queueing strategy: fifo
 Output queue: 0/40 (size/max)
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
 24 packets input, 4536 bytes
 Received 13 broadcasts, 0 runts, 0 giants, 0 throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
 0 watchdog
 0 input packets with dribble condition detected
 66 packets output, 7056 bytes, 0 underruns
 0 output errors, 0 collisions, 3 interface resets
 0 unknown protocol drops
 0 babbles, 0 late collision, 0 deferred
 0 lost carrier, 0 no carrier
 0 output buffer failures, 0 output buffers swapped out

```

```

R-2#show interfaces FastEthernet 0/1
FastEthernet0/1 is up, line protocol is up
 Hardware is Gt96k FE, address is 00aa.00ad.0002 (bia c404.0764.0001)
 Description: LINK-TO-R-1
 Internet address is 195.1.1.2/30
 MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
 reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation ARPA, loopback not set
 Keepalive set (10 sec)
 Full-duplex, 100Mb/s, 100BaseTX/FX
 ARP type: ARPA, ARP Timeout 04:00:00
 Last input 00:00:36, output 00:00:06, output hang never
 Last clearing of "show interface" counters never
 Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
 Queueing strategy: fifo
 Output queue: 0/40 (size/max)
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
 20 packets input, 3716 bytes
 Received 8 broadcasts, 0 runts, 0 giants, 0 throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
 0 watchdog
 0 input packets with dribble condition detected
 69 packets output, 7526 bytes, 0 underruns
 0 output errors, 0 collisions, 3 interface resets
 0 unknown protocol drops
 0 babbles, 0 late collision, 0 deferred
 0 lost carrier, 0 no carrier
 0 output buffer failures, 0 output buffers swapped out

```

```

R-1#show ip interface brief

```

| Interface       | IP-Address | OK? | Method | Status                | Protocol |
|-----------------|------------|-----|--------|-----------------------|----------|
| FastEthernet0/0 | 196.1.1.1  | YES | NVRAM  | up                    | up       |
| FastEthernet0/1 | unassigned | YES | NVRAM  | administratively down | down     |
| FastEthernet1/0 | unassigned | YES | NVRAM  | administratively down | down     |

```

R-2#show ip interface brief

```

| Interface       | IP-Address | OK? | Method | Status                | Protocol |
|-----------------|------------|-----|--------|-----------------------|----------|
| FastEthernet0/0 | unassigned | YES | NVRAM  | administratively down | down     |
| FastEthernet0/1 | 196.1.1.2  | YES | NVRAM  | up                    | up       |
| FastEthernet1/0 | unassigned | YES | NVRAM  | administratively down | down     |

```

R-1#ping 195.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 195.1.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/36/84 ms

```

## Модельний приклад налагодження параметрів послідовного каналу зв'язку, побудованого між маршрутизаторами Cisco

Розглянемо специфіку налагодження послідовних інтерфейсів маршрутизаторів Cisco у ході організації двоточкового послідовного каналу зв'язку, що зображений на рисунку 5.75.

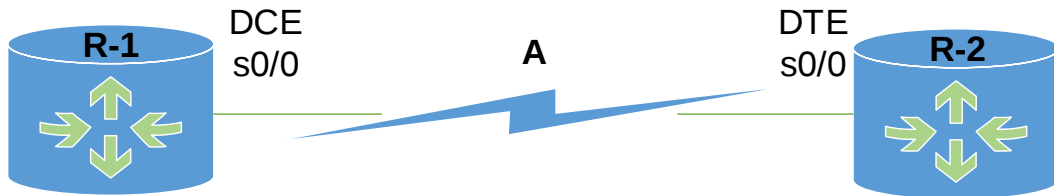


Рисунок 5.75 – Приклад мережі

Під час побудови даного каналу для з'єднання пристроїв використано дані таблиці 5.59.

Таблиця 5.59 – Параметри інтерфейсів пристроїв для прикладу

| Пристрій          | Інтерфейс   | Підключення до пристрою | Підключення до інтерфейсу |
|-------------------|-------------|-------------------------|---------------------------|
| Маршрутизатор R-1 | Se0/0 (DCE) | Маршрутизатор R-2       | Se0/0 (DTE)               |
| Маршрутизатор R-2 | Se0/0 (DTE) | Маршрутизатор R-1       | Se0/0 (DCE)               |

Для налаштування параметрів адресації пристроїв використано дані таблиці 5.60.

Таблиця 5.60 – Параметри адресації мережі

| Підмережа/Пристрій | Інтерфейс/Мережний адаптер/Шлюз | IP-адреса | Маска підмережі | Префікс |
|--------------------|---------------------------------|-----------|-----------------|---------|
| Підмережа А        | –                               | 196.1.1.0 | 255.255.255.252 | /30     |
| Маршрутизатор R-1  | Інтерфейс Se0/0                 | 196.1.1.1 | 255.255.255.252 | /30     |
| Маршрутизатор R-2  | Інтерфейс Se0/0                 | 196.1.1.2 | 255.255.255.252 | /30     |

Сценарії налагодження параметрів послідовних інтерфейсів та параметрів адресації для маршрутизаторів R-1 та R-2 за умови використання встановленого за замовчуванням каналного протоколу HDLC наведені нижче.

```
R-1>enable
R-1#configure terminal
R-1(config)#interface Serial 0/0
R-1(config-if)#description LINK-TO-R-2
R-1(config-if)#clock rate 64000
R-1(config-if)#ip address 196.1.1.1 255.255.255.252
R-1(config-if)#no shutdown
R-1(config-if)#exit
R-1(config)#exit
```

```

R-2>enable
R-2#configure terminal
R-2(config)#interface Serial 0/0
R-2(config-if)#description LINK-TO-R-1
R-2(config-if)#ip address 196.1.1.2 255.255.255.252
R-2(config-if)#no shutdown
R-2(config-if)#exit
R-2(config)#exit

```

## Результати виконання команд моніторингу та діагностики роботи інтерфейсів маршрутизаторів для розглянутого прикладу

З метою перегляду інформації про функціонування інтерфейсів маршрутизаторів для розглянутого прикладу використано команди **show controllers**, **show interfaces**. Перевірка зв'язку між маршрутизаторами здійснена за допомогою команди **ping**. Результати роботи цих команд для маршрутизаторів **R-1** та **R-2** наведено відповідно далі:

```

R-1#show controllers serial 0/0
Interface Serial0/0
Hardware is GT96K
DCE 530, clock rate 64000
idb at 0x6570905C, driver data structure at 0x65710780
wic_info 0x65710D84
Physical Port 1, SCC Num 1

```

```

R-2#show controllers serial 0/0
Interface Serial0/0
Hardware is GT96K
DTE 530 serial cable attached
idb at 0x6570905C, driver data structure at 0x65710780
wic_info 0x65710D84
Physical Port 1, SCC Num 1

```

```

R-1#show interfaces Serial 0/0
Serial0/0 is up, line protocol is up
 Hardware is GT96K Serial
 Description: LINK-TO-R-2
 Internet address is 196.1.1.1/30
 MTU 1492 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
 reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation HDLC, loopback not set
 Keepalive set (5 sec)
 Restart-Delay is 3 secs
 Last input 00:00:01, output 00:00:01, output hang never
 Last clearing of "show interface" counters never
 Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
 Queueing strategy: weighted fair
 Output queue: 0/1000/64/0 (size/max total/threshold/drops)
 Conversations 0/1/256 (active/max active/max total)
 Reserved Conversations 0/0 (allocated/max allocated)
 Available Bandwidth 1158 kilobits/sec
 5 minute input rate 0 bits/sec, 0 packets/sec

```

```
5 minute output rate 0 bits/sec, 0 packets/sec
 6 packets input, 1038 bytes, 0 no buffer
Received 6 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
56 packets output, 3123 bytes, 0 underruns
0 output errors, 0 collisions, 78 interface resets
0 unknown protocol drops
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

```
R-2#show interfaces Serial 0/0
Serial0/0 is up, line protocol is up
 Hardware is GT96K Serial
 Description: LINK-TO-R-1
 Internet address is 196.1.1.2/30
 MTU 1492 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
 reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation HDLC, loopback not set
 Keepalive set (5 sec)
 Restart-Delay is 3 secs
 Last input 00:00:02, output 00:00:02, output hang never
 Last clearing of "show interface" counters never
 Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
 Queueing strategy: weighted fair
 Output queue: 0/1000/64/0 (size/max total/threshold/drops)
 Conversations 0/1/256 (active/max active/max total)
 Reserved Conversations 0/0 (allocated/max allocated)
 Available Bandwidth 1158 kilobits/sec
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 100 packets input, 5073 bytes, 0 no buffer
Received 100 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
50 packets output, 2988 bytes, 0 underruns
0 output errors, 0 collisions, 6 interface resets
0 unknown protocol drops
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

```
R-1#ping 196.1.1.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 196.1.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/36/84 ms
```

## Модельний приклад налагодження статичних маршрутів та маршрутів за замовчуванням на маршрутизаторах Cisco

Розглянемо специфіку налагодження статичних маршрутів та маршрутів за замовчуванням на маршрутизаторах Cisco для з'єднання, схема якої наведена на рисунку 5.76. Параметрів адресації інтерфейсів пристроїв також наведені на рисунку 5.76.

Зауважимо, що попередні сценарії базового налаштування для всіх маршрутизаторів вже виконані і кожному інтерфейсу вже призначена *IPv4* адреса. Налаштування мережі передбачає, щоб комп'ютери бачили один одного, а всі невідомі пакети направлялись в зовнішню мережу *Network*. Сценарії налагодження статичних маршрутів наведені нижче.

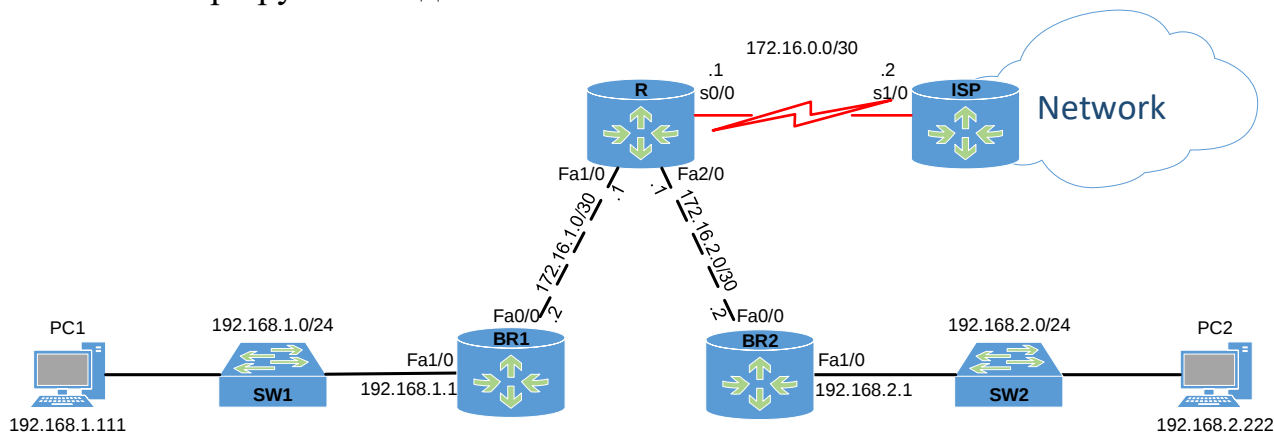


Рисунок 5.76 – Приклад мережі

Маршрутизатори **BR1** та **BR2** налаштовуються однаково. На цих маршрутизаторах достатньо пропи-сати маршрут за замовчуванням, оскільки вони є тупиковими маршрутизаторами (*stub router*) (тобто маршрутизатори, які під'єднані тільки до одного сусіднього маршрутизатора).

```
BR1(config)#ip route 0.0.0.0 0.0.0.0 172.16.1.1
```

Для створення переходів маршрутизатора **R** між локальними мережами використаємо метод статич-ного маршруту наступного переходу, для якого вказується тільки *IP*-адреса наступного переходу:

```
R(config)#ip route 192.168.1.0 255.255.255.0 172.16.1.2
```

```
R(config)#ip route 192.168.2.0 255.255.255.0 172.16.2.2
```

Та метод безпосередньо під'єданого статичного маршруту, для якого використовується тільки вихід-ний інтерфейс для маршруту за замовчуванням (в цьому випадку використовуються при під'єднанні граничного маршрутизатора **ISP** (*edge router*) до мережі постачальника послуг):

```
R(config)#ip route 0.0.0.0 0.0.0.0 s0/0
```

Також додаємо зворотні маршрути до всіх підмереж на граничний маршрутизатор **ISP**:

```
ISP(config)#ip route 192.168.1.0 255.255.255.0 s1/0
```

```
ISP(config)#ip route 192.168.2.0 255.255.255.0 s1/0
```

```
ISP(config)#ip route 172.16.1.0 255.255.255.252 s1/0
```

```
ISP(config)#ip route 172.16.2.0 255.255.255.252 s1/0
```

### Результати виконання команд моніторингу та діагностики роботи маршрутизації для розглянутого прикладу

Для перевірки статичних маршрутів поряд з командами **show ip route**, **show ipv6 route**, **ping** та **traceroute** також використовуються наступні команди: **show ip route static**, **show ip route network**, **show running-config | section ip route**. Результати роботи цих команд частково наведено далі:

Перевіримо на маршрутизаторах налаштування статичних маршрутів:

```
BR1#show ip route static
S* 0.0.0.0/0 [1/0] via 172.16.1.1
```

```
BR2#show ip route static
S* 0.0.0.0/0 [1/0] via 172.16.2.1
R#show ip route static
S 192.168.1.0/24 [1/0] via 172.16.1.2
S 192.168.2.0/24 [1/0] via 172.16.2.2
S* 0.0.0.0/0 is directly connected, Serial0/0
```

```
ISP#show ip route static
172.16.0.0/30 is subnetted, 3 subnets
S 172.16.1.0 is directly connected, Serial1/0
S 172.16.2.0 is directly connected, Serial1/0
S 192.168.1.0/24 is directly connected, Serial1/0
S 192.168.2.0/24 is directly connected, Serial1/0
```

Приклад виконання команди **show ip route** на маршрутизаторі **BR1**:

```
BR1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
 D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
 N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
 E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
 i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
 * - candidate default, U - per-user static route, o - ODR
 P - periodic downloaded static route

Gateway of last resort is 172.16.1.1 to network 0.0.0.0

172.16.0.0/30 is subnetted, 1 subnets
C 172.16.1.0 is directly connected, FastEthernet0/0
C 192.168.1.0/24 is directly connected, FastEthernet1/0
S* 0.0.0.0/0 [1/0] via 172.16.1.1
```

Приклад виконання команди **show ip route** на маршрутизаторі **ISP**:

```
ISP#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
 D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
 N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
 E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
 i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
 * - candidate default, U - per-user static route, o - ODR
 P - periodic downloaded static route

Gateway of last resort is not set

172.16.0.0/30 is subnetted, 3 subnets
C 172.16.0.0 is directly connected, Serial1/0
S 172.16.1.0 is directly connected, Serial1/0
S 172.16.2.0 is directly connected, Serial1/0
S 192.168.1.0/24 is directly connected, Serial1/0
S 192.168.2.0/24 is directly connected, Serial1/0
```

Перевірка зв'язку між **PC1** та **PC2**:

```
C:\>ping 192.168.2.222
```

```
Pinging 192.168.2.222 with 32 bytes of data:
```

```
Reply from 192.168.2.222: bytes=32 time<1ms TTL=125
Reply from 192.168.2.222: bytes=32 time<1ms TTL=125
Reply from 192.168.2.222: bytes=32 time<1ms TTL=125
Reply from 192.168.2.222: bytes=32 time<1ms TTL=125

Ping statistics for 192.168.2.222:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
 Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Перевірка маршруту між **PC1** та **PC2**:

```
C:\>tracert 192.168.2.222
```

```
Tracing route to 192.168.2.222 over a maximum of 30 hops:
 1 0 ms 0 ms 0 ms 192.168.1.1
 2 0 ms 0 ms 0 ms 172.16.1.1
 3 1 ms 1 ms 0 ms 172.16.2.2
 4 0 ms 0 ms 0 ms 192.168.2.222
Trace complete.
```

Перевірка зв'язку між **PC1** та граничним маршрутизатором провайдера **ISP**:

```
C:\>ping 172.16.0.2
```

Pinging 172.16.0.2 with 32 bytes of data:

```
Reply from 172.16.0.2: bytes=32 time=1ms TTL=253
Reply from 172.16.0.2: bytes=32 time=1ms TTL=253
Reply from 172.16.0.2: bytes=32 time=1ms TTL=253
Reply from 172.16.0.2: bytes=32 time=1ms TTL=253
```

```
Ping statistics for 172.16.0.2:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
 Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Перевірка маршруту між **PC1** та граничним маршрутизатором провайдера **ISP**:

```
C:\>tracert 172.16.0.2
```

```
Tracing route to 172.16.0.2 over a maximum of 30 hops:
 1 0 ms 0 ms 9 ms 192.168.1.1
 2 0 ms 0 ms 0 ms 172.16.1.1
 3 1 ms 0 ms 1 ms 172.16.0.2
Trace complete.
```

## ПРАКТИЧНЕ ЗАВДАННЯ

**Мета роботи:** ознайомитися з загальною будовою маршрутизатора *Cisco*; ознайомитися з основними можливостями мережної операційної системи *Cisco IOS* для маршрутизаторів та розглянути особливості її застосування на маршрутизаторах *Cisco*; дослідити можливості *Cisco IOS* з налагодження та діагностування основних параметрів функціонування маршрутизаторів *Cisco*; налаштувати статичні маршрути та маршрути за замовчуванням засобами маршрутизаторів *Cisco*.

## Порядок виконання роботи

1. Створити фізичний проєкт мережі або у середовищі програмного симулятора/емулятора створити мережу (рисунок 5.77). При побудові звернути увагу на вибір моделей маршрутизаторів, мережних модулів та плат, а також мережних з'єднань. На схемі канали зв'язку підмереж показані у загальному вигляді, при побудові підмережі слід вибирати потрібний тип кабелю для відповідної технології. Для побудованої мережі заповнити описову таблицю.

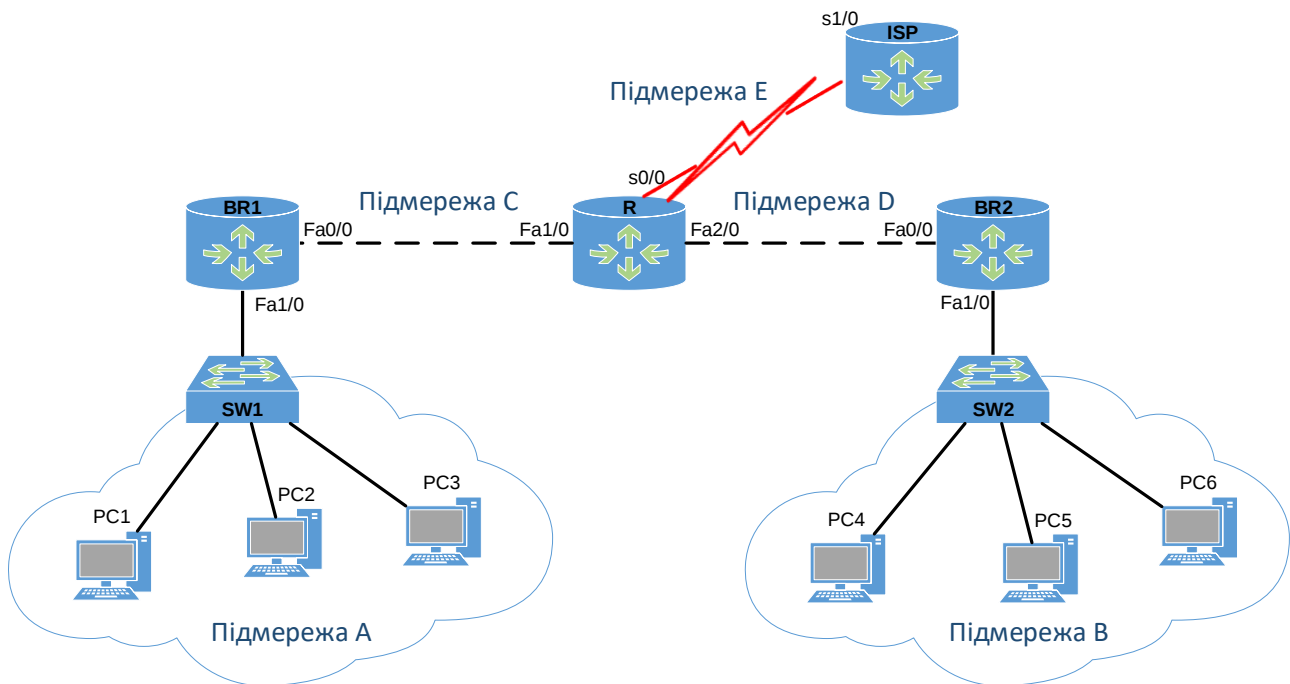


Рисунок 5.77 – Проєкт мережі

2. Провести базове налаштування маршрутизаторів, мережних інтерфейсів та з'єднань. Для цього використовувати дані таблиці 5.61. На маршрутизаторі **ISP** створити віртуальний інтерфейс *Loopback*.

3. Розробити схему адресації пристроїв мережі на основі даних, які наведені у таблицях 5.62-5.64, де **G** це номер варіанту. Результати навести у вигляді таблиці.

4. Провести налаштування параметрів IP-адресації пристроїв мережі у відповідності до даних п. 3. Перевірити наявність зв'язку між парами пристроїв мережі.

5. Визначити основні параметри апаратної частини маршрутизаторів та інформацію про встановлені на маршрутизаторах *Cisco IOS*. Результати навести у вигляді таблиці 5.65.

6. Налаштувати маршрути між маршрутизаторами для забезпечення зв'язку.

Таблиця 5.61 – Параметри каналів зв’язку підмереж

| № варіанта | Підмережа А | Підмережа В | Підмережа С       |                   |
|------------|-------------|-------------|-------------------|-------------------|
|            |             |             | Clock rate, біт/с | Bandwidth, Кбіт/с |
| 1          | 10BaseT     | 100BaseTX   | 9600              | 64                |
| 2          | 10BaseT     | 100BaseTX   | 1000000           | 128               |
| 3          | 10BaseT     | 100BaseFX   | 38400             | 192               |
| 4          | 10BaseT     | 100BaseFX   | 250000            | 256               |
| 5          | 100BaseTX   | 100BaseTX   | 64000             | 320               |
| 6          | 100BaseTX   | 100BaseTX   | 128000            | 384               |
| 7          | 100BaseTX   | 100BaseFX   | 125000            | 448               |
| 8          | 100BaseTX   | 100BaseFX   | 128000            | 512               |
| 9          | 100BaseFX   | 100BaseTX   | 148000            | 576               |
| 10         | 100BaseFX   | 100BaseTX   | 250000            | 640               |
| 11         | 100BaseFX   | 100BaseFX   | 500000            | 704               |
| 12         | 100BaseFX   | 100BaseFX   | 800000            | 768               |
| 13         | 1000BaseT   | 100BaseTX   | 1000000           | 832               |
| 14         | 1000BaseT   | 100BaseTX   | 1300000           | 896               |
| 15         | 1000BaseT   | 100BaseFX   | 2000000           | 960               |
| 16         | 1000BaseT   | 100BaseFX   | 1000000           | 1024              |
| 17         | 100BaseFX   | 100BaseTX   | 19200             | 1088              |
| 18         | 100BaseFX   | 100BaseTX   | 2000000           | 1152              |
| 19         | 100BaseFX   | 100BaseFX   | 56000             | 1216              |
| 20         | 100BaseFX   | 100BaseFX   | 19200             | 1280              |
| 21         | 1000BaseFX  | 100BaseTX   | 72000             | 32                |
| 22         | 1000BaseFX  | 100BaseTX   | 500000            | 64                |
| 23         | 1000BaseFX  | 100BaseFX   | 64000             | 96                |
| 24         | 1000BaseFX  | 100BaseFX   | 128000            | 128               |
| 25         | 10BaseT     | 100BaseFX   | 250000            | 160               |
| 26         | 100BaseFX   | 10BaseT     | 800000            | 192               |
| 27         | 1000BaseFX  | 1000BaseT   | 128000            | 128               |
| 28         | 1000BaseFX  | 1000BaseT   | 19200             | 256               |
| 29         | 100BaseFX   | 10BaseT     | 2000000           | 256               |
| 30         | 10BaseT     | 100BaseFX   | 1000000           | 512               |

Таблиця 5.62 – Параметри IP-адресації підмереж

| Мережа | Адреса мережі | Префікс |
|--------|---------------|---------|
| A      | 192.G.10.0    | /24     |
| B      | 192.G.20.0    | /24     |
| C      | 172.10.G.0    | /30     |
| D      | 172.20.G.0    | /30     |
| E      | 172.30.G.0    | /30     |

Таблиця 5.63 – Дані для визначення параметрів адресації мережі

| <b>№ варіанта</b> | <b><i>IP-адреса шлюзу за замовчуванням, IP-адреса основного DNS-сервера</i></b> | <b><i>IP-адреса альтернативного DNS-сервера 1</i></b> | <b><i>IP-адреса альтернативного DNS-сервера 2</i></b> |
|-------------------|---------------------------------------------------------------------------------|-------------------------------------------------------|-------------------------------------------------------|
| 1                 | Перша <i>IP-адреса</i> діапазону                                                | Level3 Communications                                 | Level3 Communications                                 |
| 2                 | Остання <i>IP-адреса</i> діапазону                                              | Google                                                | Google                                                |
| 3                 | Перша <i>IP-адреса</i> діапазону                                                | OpenDNS Home                                          | OpenDNS Home                                          |
| 4                 | Остання <i>IP-адреса</i> діапазону                                              | Securly                                               | Securly                                               |
| 5                 | Перша <i>IP-адреса</i> діапазону                                                | Comodo Secure DNS                                     | Comodo Secure DNS                                     |
| 6                 | Остання <i>IP-адреса</i> діапазону                                              | DNS Advantage                                         | DNS Advantage                                         |
| 7                 | Перша <i>IP-адреса</i> діапазону                                                | Norton ConnectSafe                                    | Norton ConnectSafe                                    |
| 8                 | Остання <i>IP-адреса</i> діапазону                                              | SafeDNS                                               | SafeDNS                                               |
| 9                 | Перша <i>IP-адреса</i> діапазону                                                | OpenNIC                                               | OpenNIC                                               |
| 10                | Остання <i>IP-адреса</i> діапазону                                              | Public-Root                                           | Public-Root                                           |
| 11                | Перша <i>IP-адреса</i> діапазону                                                | Level3 Com-ns                                         | Level3 Com-ns                                         |
| 12                | Остання <i>IP-адреса</i> діапазону                                              | Google                                                | Google                                                |
| 13                | Перша <i>IP-адреса</i> діапазону                                                | OpenDNS Home                                          | OpenDNS Home                                          |
| 14                | Остання <i>IP-адреса</i> діапазону                                              | Securly                                               | Securly                                               |
| 15                | Перша <i>IP-адреса</i> діапазону                                                | Comodo Secure DNS                                     | Comodo Secure DNS                                     |
| 16                | Остання <i>IP-адреса</i> діапазону                                              | DNS Advantage                                         | DNS Advantage                                         |
| 17                | Перша <i>IP-адреса</i> діапазону                                                | Norton ConnectSafe                                    | Norton ConnectSafe                                    |
| 18                | Остання <i>IP-адреса</i> діапазону                                              | SafeDNS                                               | SafeDNS                                               |
| 19                | Перша <i>IP-адреса</i> діапазону                                                | OpenNIC                                               | OpenNIC                                               |
| 20                | Остання <i>IP-адреса</i> діапазону                                              | Public-Root                                           | Public-Root                                           |
| 21                | Перша <i>IP-адреса</i> діапазону                                                | Level3 Com-ns                                         | Level3 Com-ns                                         |
| 22                | Остання <i>IP-адреса</i> діапазону                                              | Google                                                | Google                                                |
| 23                | Перша <i>IP-адреса</i> діапазону                                                | OpenDNS Home                                          | OpenDNS Home                                          |
| 24                | Остання <i>IP-адреса</i> діапазону                                              | Securly                                               | Securly                                               |
| 25                | Перша <i>IP-адреса</i> діапазону                                                | Comodo Secure DNS                                     | Comodo Secure DNS                                     |
| 26                | Остання <i>IP-адреса</i> діапазону                                              | DNS Advantage                                         | DNS Advantage                                         |
| 27                | Перша <i>IP-адреса</i> діапазону                                                | Norton ConnectSafe                                    | Norton ConnectSafe                                    |
| 28                | Остання <i>IP-адреса</i> діапазону                                              | SafeDNS                                               | SafeDNS                                               |
| 29                | Перша <i>IP-адреса</i> діапазону                                                | OpenNIC                                               | OpenNIC                                               |
| 30                | Остання <i>IP-адреса</i> діапазону                                              | Public-Root                                           | Public-Root                                           |

Таблиця 5.64 – Основні публічні DNS-сервери

| № з/п | Провайдер             | IP-адреса основного (первинного) DNS-сервера | IP-адреса альтернативного (вторинного) DNS-сервера |
|-------|-----------------------|----------------------------------------------|----------------------------------------------------|
| 1     | Level3 Communications | 209.244.0.3                                  | 209.244.0.4                                        |
| 2     | Verisign              | 64.6.64.6                                    | 64.6.65.6                                          |
| 3     | Google                | 8.8.8.8                                      | 8.8.4.4                                            |
| 4     | DNS.WATCH             | 84.200.69.80                                 | 84.200.70.40                                       |
| 5     | Comodo Secure DNS     | 8.26.56.26                                   | 8.20.247.20                                        |
| 6     | OpenDNS Home          | 208.67.222.222                               | 208.67.220.220                                     |
| 7     | DNS Advantage         | 156.154.70.1                                 | 156.154.71.1                                       |
| 8     | Norton ConnectSafe    | 198.153.192.40                               | 198.153.194.40                                     |
| 9     | SafeDNS               | 195.46.39.39                                 | 195.46.39.40                                       |
| 10    | OpenNIC               | 74.207.247.4                                 | 64.0.55.201                                        |
| 11    | Securly               | 184.169.143.224                              | 184.169.161.155                                    |
| 12    | SmartViper            | 208.76.50.50                                 | 208.76.51.51                                       |
| 13    | Dyn                   | 216.146.35.35                                | 216.146.36.36                                      |
| 14    | FreeDNS               | 37.235.1.174                                 | 37.235.1.177                                       |
| 15    | Public-Root           | 199.5.157.131                                | 208.71.35.137                                      |
| 16    | Alternate DNS         | 198.101.242.72                               | 23.253.163.53                                      |
| 17    | Yandex.DNS            | 77.88.8.8                                    | 77.88.8.1                                          |
| 18    | UncensoredDNS         | 91.239.100.100                               | 89.233.43.71                                       |
| 19    | censurfridns.dk       | 89.233.43.71                                 | 89.104.194.142                                     |
| 20    | ScrubIt               | 67.138.54.100                                | 207.225.209.66                                     |
| 21    | Quad9                 | 9.9.9.9                                      | 149.112.112.112                                    |
| 22    | Neustar               | 156.154.70.1                                 | 156.154.71.1                                       |
| 23    | Claudflare DNS        | 1.1.1.1                                      | 1.0.0.1                                            |

Таблиця 5.65 – Параметри маршрутизаторів мережі

| Параметр                  | R | BR1 | BR2 | ISP |
|---------------------------|---|-----|-----|-----|
| Модель маршрутизатора     |   |     |     |     |
| Модель та номер процесора |   |     |     |     |
| Об'єм пам'яті RAM, Мб     |   |     |     |     |
| Об'єм пам'яті NVRAM, Мб   |   |     |     |     |
| Об'єм Flash:              |   |     |     |     |
| – всього, Мб              |   |     |     |     |
| – зайнято, Мб             |   |     |     |     |
| – вільно, Мб              |   |     |     |     |
| Конфігураційний реєстр    |   |     |     |     |
| Кількість інтерфейсів:    |   |     |     |     |
| – Ethernet                |   |     |     |     |
| – Fast Ethernet           |   |     |     |     |
| – Gigabit Ethernet        |   |     |     |     |
| – Serial                  |   |     |     |     |
| – Loopback                |   |     |     |     |
| – Tunnel                  |   |     |     |     |
| – VLAN                    |   |     |     |     |
| Версія IOS                |   |     |     |     |
| Образ IOS                 |   |     |     |     |
| Розмір файла образу IOS   |   |     |     |     |
| Системний час             |   |     |     |     |

## КОНТРОЛЬНІ ПИТАННЯ

1. Типова структурна схема маршрутизатора фірми *Cisco*.
2. Блоки пам'яті маршрутизатора *Cisco* та їх призначення.
3. Інтерфейси маршрутизаторів *Cisco*.
4. Модулі та плати розширення маршрутизаторів *Cisco*.
5. Змінні інтерфейсні модулі маршрутизаторів *Ethernet*.
6. Кабельні з'єднання *Ethernet* та *Serial* інтерфейсів маршрутизаторів *Cisco*.
7. Кабельні підключення, що застосовуються з метою налагодження та керування маршрутизатором *Cisco*.
8. Фізичні і логічні інтерфейси маршрутизаторів *Cisco*.
9. Загальна характеристика *Cisco IOS*. Платформи, набори можливостей та версії *Cisco IOS*, які використовуються для маршрутизаторів *Cisco*.
10. Джерела завантаження образу *Cisco IOS* та конфігурації маршрутизатора *Cisco*.
11. Командні режими *Cisco IOS* для маршрутизаторів *Cisco*. Команди переходів між командними режимами *Cisco IOS* для маршрутизаторів.
12. Команди діагностики апаратних складових маршрутизатора *Cisco*.
13. Команди діагностики функціонування інтерфейсів маршрутизатора *Cisco*.
14. Порядок та основні команди налагодження *Ethernet*-інтерфейсу маршрутизатора *Cisco*.
15. Порядок та основні команди налагодження послідовного інтерфейсу маршрутизатора *Cisco*.
16. Що таке статичний маршрут і як він відрізняється від динамічного?
17. Які випадки використання статичних маршрутів є раціональними у мережах *Cisco*?
18. Які команди використовують для створення статичних маршрутів на маршрутизаторі *Cisco*?
19. Як встановити пріоритет маршруту при наявності кількох маршрутів до однієї мережі?
20. Як перевірити правильність налаштування статичного маршруту на маршрутизаторі *Cisco*?
21. Що таке маршрут за замовчуванням і як він використовується у мережах *Cisco*?

## СПИСОК ЛІТЕРАТУРИ

1. Буров Є. В., Митник М. М. Комп'ютерні мережі: підручник. Львів: Магнолія 2006, 2021. Т. 1. 334 с.
2. Буров Є. В., Митник М. М. Комп'ютерні мережі: підручник. Львів: Магнолія 2006, 2021. Т. 2. 204 с.
3. Єфіменко А. А. Основи побудови локальних комп'ютерних мереж Ethernet на базі керованих комутаторів компанії Cisco: навч. посіб. Житомир: Житомирська політехніка, 2021. 116 с.
4. Тарнавський Ю. А., Кузьменко І. М. Організація комп'ютерних мереж: підручник для студ. спец. 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки». Київ: НТУУ «КПІ ім. Ігоря Сікорського», 2018. 259 с.
5. Конспект лекцій з дисципліни «Інформаційно-комунікаційні системи», для студентів усіх форм навчання спеціальності 125 «Кібербезпека» за освітньою програмою «Безпека інформаційних комунікаційних систем». Ч. І. Електрон. вид. / упоряд. Г. З. Халімов. Харків: ХНУРЕ, 2019. 207 с.
6. Odom W. CCNA 200-301 Official Cert Guide. Hoboken, NJ: Cisco Press, 2020. Vol. 1. 1095 p.
7. Odom W. CCNA 200-301 Official Cert Guide. Hoboken, NJ: Cisco Press, 2020. Vol. 2. 1444 p.
8. Cisco Wireless LAN Controller Configuration Guide, Release 6.0. URL: <https://www.cisco.com/c/en/us/support/index.html?mode=prod> (дата звернення: 10.10.2024).
9. Технології забезпечення безпеки мережевої інфраструктури / В. Л. Бурячок, А. О. Аносов, В. В. Семко та ін. Київ: КУБГ, 2019. 218 с.
10. Комп'ютерні мережі: підручник / О. Д. Азаров, С. М. Захарченко, О. В. Кадук та ін. Вінниця: ВНТУ, 2020. 378 с.
11. Блозва А. І., Матус Ю. В., Касаткін Д. Ю. Комп'ютерні мережі. Київ: Компрінт, 2019. 483 с.
12. Graziani R. IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6. 2nd ed. Hoboken, NJ: Cisco Press, 2017. 688 p.
13. Cisco WLC Interfaces, Ports & Their Functionality. Understand How WLCs Work, Connect to the Network Infrastructure & Wi-Fi SSID/VLAN Mappings. URL: <https://www.firewall.cx/cisco/cisco-wireless/cisco-wireless-controllers-interfaces-ports-functionality.html> (дата звернення: 10.10.2024).
14. Peterson L. L., Davie B. S. Computer Networks: A Systems Approach. Cambridge, MS: Elsevier, 2022. 992 p.
15. Cisco Feature Navigator. Configuring EtherChannels. URL: [https://www.cisco.com/c/dam/en/us/td/docs/switches/lan/catalyst9600/software/release/16-12/configuration\\_guide/lyr2/configuring\\_etherchannels.html#topic\\_z1l\\_bn1\\_tkb](https://www.cisco.com/c/dam/en/us/td/docs/switches/lan/catalyst9600/software/release/16-12/configuration_guide/lyr2/configuring_etherchannels.html#topic_z1l_bn1_tkb) (дата звернення: 10.10.2024).

16. Tanenbaum A. S., Wetherall D. J. Computer Networks. 6th ed. Pearson, 2021. 945 p.
17. Configuration Fundamentals Configuration Guide, Cisco IOS Release 15.1S. URL: <https://www.cisco.com/en/US/docs/ios-xml/ios/fundamentals/configuration/15-1s/cf-15-1s-book.pdf> (дата звернення: 10.10.2024).
18. Catalyst 2960 and 2960-S Switches Software Configuration Guide. URL: [https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960/software/release/15-2\\_1\\_e/configuration/guide/2960\\_scg.html](https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960/software/release/15-2_1_e/configuration/guide/2960_scg.html) (дата звернення: 10.10.2024).
19. GLBP - Gateway Load Balancing Protocol. URL: [https://www.cisco.com/en/US/docs/ios/12\\_2t/12\\_2t15/feature/guide/ft\\_glb.html](https://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ft_glb.html) (дата звернення: 10.10.2024).
20. CCSP SNRS Exam Self-Study: Mitigating Layer 2 Attacks. URL: <https://www.ciscopress.com/articles/article.asp?p=474239&seqNum=2> (дата звернення: 10.10.2024).
21. Coleman D. D., Westcott D. A. Certified Wireless Network Administrator. Study Guide. Indiana: John Wiley & Sons, 2021. 1013 p.
22. Sharpe R., Warnicke E., Lamping U. Wireshark User's Guide. URL: [https://www.wireshark.org/docs/wsug\\_html\\_chunked/](https://www.wireshark.org/docs/wsug_html_chunked/) (дата звернення: 10.10.2024).
23. Cisco Interface Cards. URL: <https://www.cisco.com/c/en/us/products/interfaces-modules/interface-cards/index.html> (дата звернення: 10.10.2024).
24. Lammle T. CCNA: Routing and Switching: Complete Study Guide. Indianapolis, IN: John Wiley and Sons, 2016. 1069 p.
25. Velte A. T., Velte T. J. Cisco: A Beginner's Guide. New York: McGraw-Hill Education, 2014. 670 p.
26. White R. Cisco Certified Support Technician CCST Networking 100-150 Official Cert Guide. IN: Cisco Press, 2023. 608 p.
27. Meyers M., Jernigan S. CompTIA Network + Certification All-in-One Exam Guide. Eighth ed. (Exam N10-008). Cengage Learning, 2024. 880 p.

Навчальне електронне видання

# КОМП'ЮТЕРНІ МЕРЕЖІ

Навчальний посібник

**Авторський колектив:**

*Четинога Анатолій Володимирович,  
Єфіменко Андрій Анатолійович,  
Рудаков Костянтин Сергійович,  
Лавданський Артем Олександрович,  
Ланських Євген Володимирович,  
Фауре Еміль Віталійович*

*В авторській редакції*

Технічний редактор *Катерина Давиденко*