Міністерство освіти і науки України
Чернівецький національний університет
імені Юрія Федьковича

Комп'ютерні мережі

Методичні рекомендації та завдання для лабораторних робіт

Чернівці
Чернівецький національний університет
2023

УДК 004.7 : 075.8

Рекомендовано до друку кафедрою математичного моделювання факультету математики та інформатики
Чернівецького національного університету імені Юрія Федьковича
(протокол № 17 від « 27 » червня 2023 року)

Укладачі:
Олександр Матвій, кандидат фізико-математичних наук, доцент кафедри математичного моделювання;
Ігор Черевко, доктор фізико-математичних наук, професор кафедри математичного моделювання.

Комп'ютерні мережі: методичні рекомендації та завдання для лабораторних робіт. Укл.: Олександр Матвій, Ігор Черевко – Чернівці: Чернівецький національний університет, 2023. – 72 с.

Методичні рекомендації містять завдання для лабораторних робіт та теоретичні відомості, необхідні для їх виконання, з дисципліни «Комп'ютерні мережі».

Для студентів, що здобувають освіту в галузі знань „Інформаційні технології” для спеціальностей “Комп’ютерні науки”, “Системний аналіз” та інших спеціальностей з подібною програмою вивчення дисципліни.

УДК 004.7 : 075.8

Вступ

Дисципліна «Комп'ютерні мережі» призначена для вивчення основ проектування та організації комп'ютерних мереж. Основними завданнями дисципліни є вивчення загальних принципів і стандартів побудови та функціонування комп’ютерних мереж.

Методичні вказівки містять завдання до лабораторних робіт, а також теоретичні відомості, необхідні для їх виконання, які дають змогу студентам самостійно засвоїти здобуті знання.

Матеріали методичних вказівок можуть бути корисними студентам різних спеціальностей всіх форм навчання, як для самостійного вивчення теоретичного матеріалу, так і в якості допоміжного засобу при організації дистанційного навчання.

Мета та завдання навчальної дисципліни
“Комп'ютерні мережі**”**

Мета навчальної дисципліни:

формування у студентів знань з теорії та технологій проектування, побудови й супроводження комп’ютерних мереж ЕОМ, навичок їх використання для створення та експлуатації програмно-апаратних систем для використання в локальних та глобальних обчислювальних мережах.

Після вивчення даної дисципліни студент повинен

знати:

- основні поняття мережевих технологій;
- принципи опису функціонування мережевих технологій у межах моделі OSI;
- принципи архітектурної побудови (апаратне та програмне забезпечення) сучасних локальних та глобальних мереж;
- базові технології мереж та їх можливості;
- сучасні стеки протоколів, принципи побудови та функціонування стеку протоколів TCP/IP.

вміти:

- планувати мережеву інфраструктуру;
- виконувати ділення мережі на сегменти;
- розробляти логічну і фізичну структуру локальної комп’ютерної мережі, топологію структурованих кабельних систем;
- використовувати програмне забезпечення Cisco Packet Tracer (програмний симулятор роботи мережі) при проектуванні малої та середньої за розміром локальної мереж та налагодження різних мережевих пристроїв для організації та побудови мережі;
- використовувати мережеві можливості сучасних ОС;
- використовувати різні програмні засоби діагностики роботи локальних мереж;
- здійснювати пошук інформації в різних джерелах;
- ефективно працювати як індивідуально, так і у складі команди;
- поєднувати теорію і практику.

Теоретичний зміст програми навчальної дисципліни

Змістовий модуль 1.

Тема 1. Базові поняття і особливості локальних мереж

Тема 2. Топологія мереж

Тема 3. Рівні мережевої архітектури. Еталонна модель OSI

Тема 4. Фізичний та канальний рівень комп'ютерної мережі

Тема 5. Модель стеку протоколів TCP/IP

Тема 6. Проміжні пристрої. Об'єднання мереж за допомогою пристроїв 2 рівня моделі OSI.

Змістовий модуль 2.

Тема 7. Протоколи мережевого рівня

Тема 8. Класова та безкласова адресація

Тема 9. Статична та динамічна маршрутизація. Віртуальні локальні мережі.

Тема 10. Протоколи транспортного та прикладного рівня

Лабораторна робота №1. Середовище програмного емулятора Cisco Packet Tracer

Мета роботи: Засвоїти середовище програмного емулятора Cisco Packet Tracer для проектування мережевих топологій з використанням маршрутизаторів, комутаторів та робочих станцій компанії Cisco.

Методичні вказівки

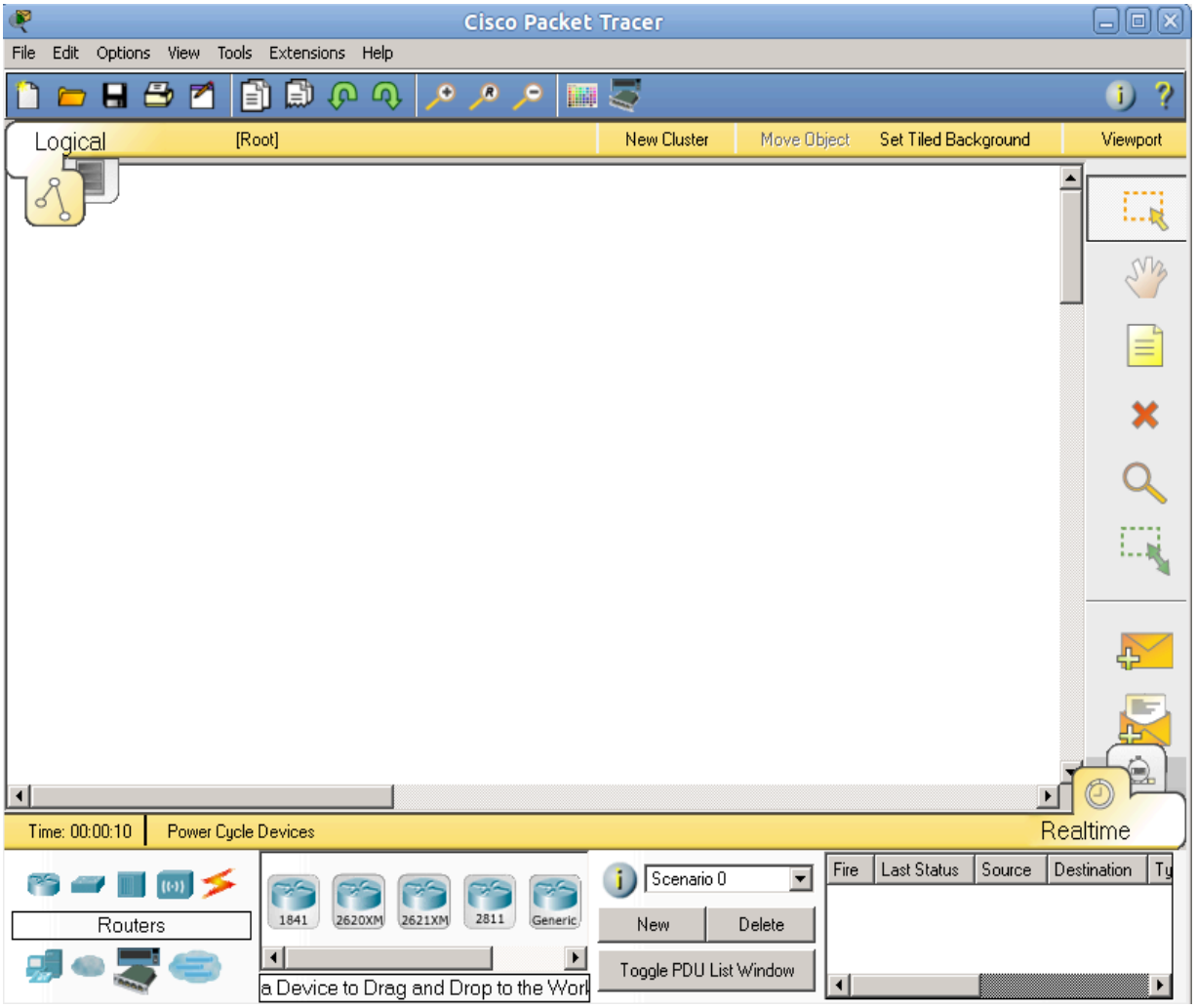
Початок роботи з програмою Cisco Packet Tracer (CPT)

Cisco *Packet Tracer* – це симулятор мережі, який створений компанією Cisco. *Програм*а (мал.1.1) дозволяє проектувати та аналізувати мережі на різноманітному обладнанні у довільних топологіях з підтримкою різних протоколів. У ній є можливість вивчати роботу різних мережевих пристроїв: маршрутизаторів, комутаторів, точок бездротового доступу, персональних комп'ютерів, мережевих принтерів і т.д. Даний програмний додаток є найбільш простим та ефективним серед своїх конкурентів. Розглянемо використання версії програми Cisco *Packet Tracer* 6.1.1.

Мал. 1.1. Логотип програми CPT

1. **Загальний вигляд програми Packet Tracer**

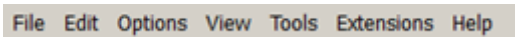
Робота із емулятором Cisco Packet Tracer розпочинається після запуску програми, загальний вигляд робочого вікна якої можна побачити на мал.1.2.



Мал.1.2. Загальний вигляд робочого вікна програми Packet Tracer

Основні елементи робочої області:

1) **Menu Bar** (*Головне меню)* − панель, яка містить меню *File, Edit, View Options, Tools, Extensions, Help (*мал.1.3*)*.



Мал.1.3. Головне меню

- **File** (*Файл)* - містить *операції* відкриття / збереження документів .
- **Edit** (Правка) – містить стандартні *операції* "копіювати/вирізати", "скасувати/ повторити".
- **Options** (Налаштування) – містить інструменти налаштування програми. Зокрема, тут розташована кнопка 

, що дозволяє змінити локалізацію програми на інші мови.
- **View** (Вигляд) - містить інструменти зміни масштабу робочої області та панелі інструментів.
- **Tools** (Інструменти) - містить палітру кольорів та вікно користувацьких пристроїв.
- **Extensions** (Розширення) - містить майстер проектів та ряд інших інструментів.

- **Help** (Допомога) – містить довідку за програмою.

2) **Main Tool Bar (***Панель інструментів***)** − панель, яка містить графічні зображення ярликів для доступу до основних команд меню *File, Edit, View і Tools* (мал.1.4).

Мал.1.4. Панель інструментів

3) **Common Tools Bar** (*Панель інструментів*) − панель, яка забезпечує доступ до таких інструментів головної програми: *Select, Move Layout, Place Note, Delete, Inspect, Add Simple PDU* та *Add Complex PDU* (мал.1.5).

Мал.1.5 Графічне меню

- Інструмент **Select** (Вибрати) можна активувати клавішею Esc. Він використовується для виділення одного або більше об'єктів для подальшого їх переміщення, копіювання або видалення.
- Інструмент **Move Layout** (Перемістити макет) використовується для прокручування великих проектів мереж.
- Інструмент **Place Note** (Зробити позначку, клавіша N) додає текст до робочої області проекту.
- Інструмент **Delete** (Видалити, клавіша Del) видаляє виділений *об'єкт* або групу об'єктів.
- Інструмент **Inspect** (Перевірка, клавіша I) дозволяє, залежно від типу пристрою, переглядати вміст таблиць (*ARP* , *NAT* , таблиці маршрутизації ін .).
- Інструмент **Drawapolygon** (Намалювати багатокутник) дозволяє малювати прямокутники, еліпси, лінії та зафарбовувати їх кольором.
- Інструмент **Resize Shape** (Змінити розмір форми, комбінація клавіш Alt+R) призначений для зміни розмірів об'єктів (чотирикутників та кіл).

4) **Logical/Physical Workspace and Navigation Bar** − панель, яка дає можливість перемикати робочу область: фізичну або логічну, а також дозволяє переміщатися між рівнями кластера.

5) **Workspace** (*Робоча область*) − область, в якій відбувається створення мережі, проводяться спостереження за симуляцією і проглядається різна інформація та статистика.

6) **Realtime/Simulation Bar (***Панель для перемикання між режимами роботи в реальному часі та симуляції***)** − за допомогою цієї панелі закладок можна перемикатися між режимами *Realtime* і *Simulation* (мал.1.6).

Мал.1.6

Інструменти **Add Simple PDU** (Додати простий *PDU*, клавіша P) та **Add Complex PDU** (Додати комплексний *PDU*, клавіша C) призначені для емулювання відправлення пакета з подальшим відстеженням його маршруту та даних всередині пакету.

Created User Packet Window − вікно, яке управляє пакетами, що були створені в мережі під час симуляції сценарію.

7) **Network Component Box** (*Область мережевих компонентів*) − область, в якій вибираються пристрої і зв'язки для розміщення їх на робочому просторі. Вона містить області *Device-Type Selection* і *Device-Specific Selection*.

Device-Type Selection Box − область містить доступні типи пристроїв і зв'язків у Packet Tracer. ***Device-Specific Selection Box*** − область, що використовується для вибору конкретних пристроїв і з'єднань, які необхідні в робочому просторі мережі (мал.1.7). Після вибору пристроїв та типу з'єднання індикатори на кожному кінці покажуть статус з'єднання.

Мал.1.7 Типи кабелів, що підтримуються в Packet Tracer

Панель обладнання містить у своїй лівій частині типи (класи) пристроїв, а у правій частини – їх найменування (моделі). При наведенні на пристрій у прямокутнику, що знаходиться в центрі між ними відображається його тип:

- **Маршрутизатори (роутери)** використовуються для пошуку оптимального маршруту передачі даних на базі алгоритмів маршрутизації.
- **Комутатори** - пристрої, які призначені для об'єднання кількох вузлів у межах одного або декількох сегментів мережі. *Комутатор* (світч) передає пакети інформації згідно таблиці комутації - тому трафік походить тільки на ту MAC- *адресу*, для якої він призначається, а не повторюється на всіх портах, як на концентраторі (хабі).
- **Бездротові пристрої** у програмі представлені бездротовим маршрутизатором та трьома точками доступу. Серед **кінцевих пристроїв** можна побачити ПК, ноутбук, *сервер*, принтер, телефони і так далі. *Інтернет* у програмі представлений у вигляді хмар та модемів *DSL*.

Packet Tracer підтримує широкий діапазон мережевих з'єднань. За допомогою ліній зв'язку створюються з'єднання вузлів мережі в єдину топологію і при цьому кожен тип кабелю може бути з'єднаний лише з певними типами інтерфейсів пристроїв:

- **Automatically choose connection type (***Автоматичний тип***)** – при цьому способі з'єднання *Packet Tracer* автоматично обирає найбільш оптимальний тип з'єднання для вибраних пристроїв.
- **Console (***Консоль***)** – консольне з'єднання. Консольне з'єднання може бути виконано між ПК та маршрутизаторами або комутаторами. При цьому мають бути виконані деякі вимоги для роботи консольного сеансу з ПК.
- **Copper Straight-Through (***Мідний прямий***)** - з'єднання мідним кабелем типу *кручена(вита) пара*, обидва кінця кабелю "обтиснуті" з використанням однакової розкладки. Цей тип кабелю є стандартним середовищем передачі даних Ethernet для підключення пристроїв, які функціонують на різних рівнях моделі OSI і з'єднується з такими типами портів: мідний 10 Мбіт/с (Ethernet), мідний 100 Мбіт/с (Fast Ethernet) і мідний 1000 Мбіт/с (Gigabit Ethernet).
- **Copper Cross-Over (***Мідний кросовер***)** - з'єднання мідним кабелем типу *кручена пара*, кінці кабелю "обтиснуті" як кросовер. Цей тип кабелю є середовищем передачі Ethernet для з'єднання пристроїв, які функціонують на однакових рівнях OSI і з'єднується з такими типами портів: мідний 10 Мбіт/с (Ethernet), мідний 100 Мбіт/с (Fast Ethernet) і мідний 1000 Мбіт/с (Gigabit Ethernet).
- **Fiber (***Оптика***)** – з'єднання за допомогою оптичного кабелю, це необхідно для з'єднання пристроїв, які мають оптичні інтерфейси.
- **Phone (***Телефонний кабель***)** – *кабель* для підключення телефонних апаратів. З'єднання через телефонну лінію може бути здійснено між пристроями, що мають модемні порти. Приклад - ПК, що додзвонюється до мережевої хмари.
- **Coaxial (***Коаксіальний кабель***)** – з'єднання пристроїв за допомогою коаксіального кабелю. Використовується для з'єднання між кабельним модемом та хмарою.
- **Serial DCE (***Серійний DCE***) та Serial DTE (***Серійний DTE***)** – з'єднання пристроїв через послідовні порти часто використовуються для зв'язків WAN. Для налаштування таких з'єднань необхідно встановити синхронізацію на стороні DCE-пристрою. Синхронізація DTE виконується за вибором. Сторону DCE можна визначити за маленькою іконкою "годинника" поруч з портом . При виборі типу з'єднання Serial DCE, перший пристрій, до якого застосовується з'єднання, стає DCE-пристроєм, а другий − автоматично стає стороною DTE. Можливе і зворотне розташування сторін, якщо вибрано тип з'єднання Serial DTE.

2. Проектування мережі

Створення топології мережі починається з вибору пристрою на панелі *Network Component* та його типу на панелі *Device-Type Selection* (якщо при цьому буде утримуватись клавіша *Ctrl,* то можна створити декілька екземплярів одного і того ж

пристрою). Користувач може додавати, налаштовувати та вилучати необхідні елементи в робочій області програми.

Діалогове вікно властивостей кожного елемента має дві вкладки:

− *physical* - містить графічний інтерфейс пристрою і дозволяє симулювати роботу з ним на фізичному рівні, зокрема можливо фізичне *подання* обладнання у вигляді його фізичної заміни (мал.1.8).

Мал.1.8. Фізична конфігурація ПК

Для зміни комплектації обладнання необхідно відключити його живлення, клацнувши мишею на кнопці живлення, та перетягнути мишкою потрібний *модуль* у вільний *слот*, потім увімкнути живлення. Як приклад, додавання до фізичної конфігурації ПК мікрофону (PT-MICROPHONE), внаслідок чого ПК змінює свій значок у програмі (мал. 1.9). Інші модулі додаються до пристроїв аналогічно.

Мал.1.9. Зміна піктограми ПК після підключення до нього мікрофона

− *config* - містить всі необхідні параметри для налаштування пристрою і має зручний для цього інтерфейс.

Також залежно від пристрою його властивості можуть мати додаткову вкладку для керування роботою вибраного елементу: *Desktop* (якщо вибрано кінцевий пристрій) або *CLI* (якщо вибрано маршрутизатор).

Packet Tracer дозволяє симулювати роботу з інтерфейсом командного рядка (ІКР) операційної системи IOS, встановленої на всіх комутаторах і маршрутизаторах компанії Cisco, перейшовши у вікно властивостей на вкладку CLI. Симулятор забезпечує підтримку практично всіх команд, доступних на реальних пристроях. Для симуляції роботи командного рядка на кінцевому пристрої необхідно у властивостях вибрати вкладку *Desktop*, а потім натиснути на ярлик *Command Prompt*.

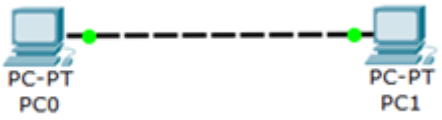
3. Збереження мережі та конфігурації пристроїв. Створену мережу можна зберегти. Файл збереженої топології має розширення *.pkt. Також Packet Tracer дає можливість користувачеві зберігати конфігурацію деяких пристроїв, наприклад маршрутизаторів або комутаторів, в текстових файлах. Для цього необхідно перейти до властивостей пристрою, у вкладці Config натиснути на кнопку “Export...” для експорту конфігурації Startup Config або Running Config. Текст файлу конфігурації пристрою running-config.txt є аналогічним до тексту інформації отриманої при використанні команди **show running-config** у IOS пристрої.

Необхідно зазначити, що конфігурація кожного пристрою зберігається в окремому файлі. Користувач має можливість змінювати конфігурацію в збереженому файлі вручну за допомогою будь-якого текстового редактора. Для завантаження необхідної конфігурації Stàrtup Config потрібно у вкладці Config натиснути кнопку “Load...” або кнопку “Merge...” для завантаження конфігурації Running Config.

Хід роботи

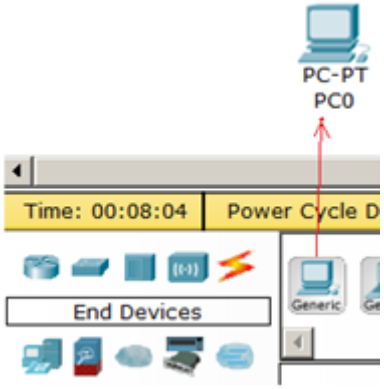
Практична робота 1. Створення мережі з двох ПК у програмі Cisco Parket Tracer

Як приклад для початкового знайомства з програмою побудуємо найпростішу *мережу* з двох ПК (мал. 1.10)



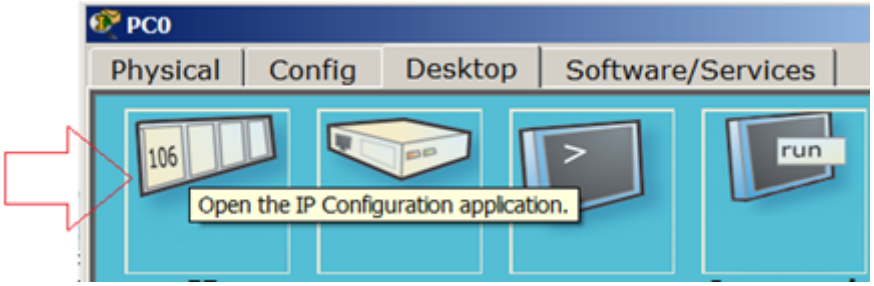
Мал.1.10 Мережа з двох ПК

Для вирішення завдання на вкладці **Кінцеві пристрої** вибираємо тип комп'ютера і переносимо його мишею в робочу область програми (мал. 1.11).



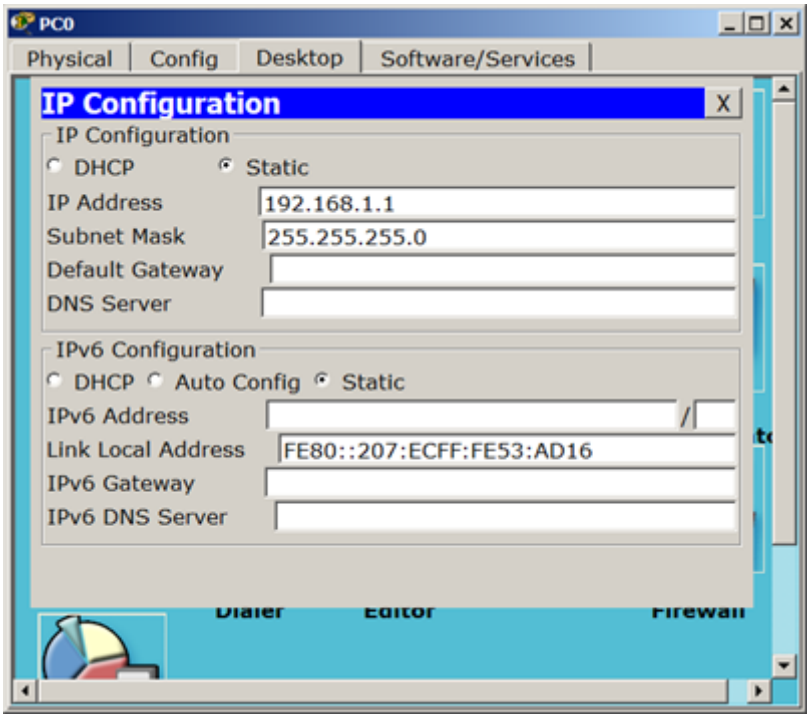
Мал.1.11 Встановлюємо в робочу область програми перший ПК

Комп'ютери з'єднуємо за допомогою мідного кросовера (*перехресний кабель*). Якщо при виборі кросовера зелені лампочки не загоряються, рекомендовано вибрати тип з'єднання **Автоматично**. Для налаштування лівого ПК: клацаємо на ньому мишею, переходимо на вкладку **Ip Configuration** (Налаштування *IP)* як на мал.1.12.



Мал.1.12

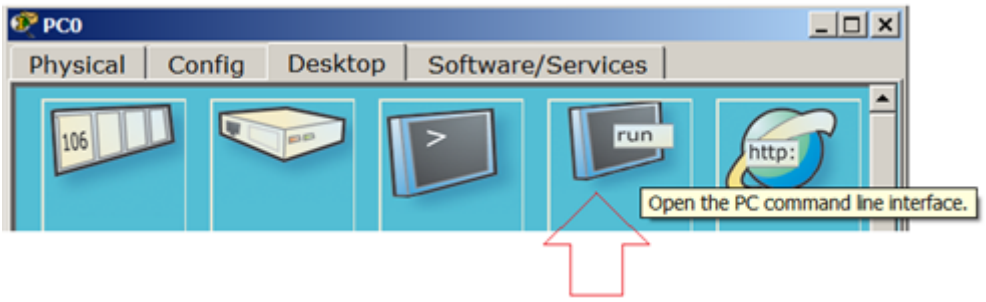
Для першого ПК вводимо IP адресу 192.168.1.1 та маску підмережі 255.255.255.0, вікно закриваємо (мал.13). Аналогічно налаштовуємо другий ПК на адресу 192.168.1.2 і ту саму маску. Також для конфігурування комп'ютера можна використовувати команду *ipconfig* з з вказанням ip-адреси та маски підмережі з командного рядка.



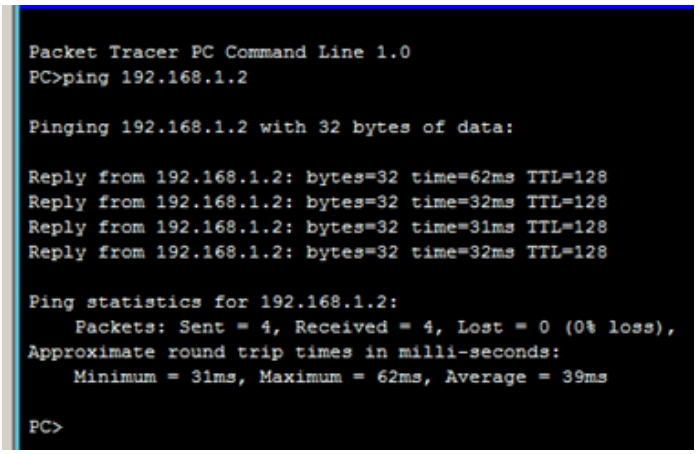
Мал.1.13 Вікно налаштування PC0

Далі перевіримо наявність зв'язку ПК і переконаємося, що ПК0 та ПК1 бачать один одного. Для цього у вкладці Desktop (Робочий стіл) перейдемо в поле run (Командна рядок, мал.1.14) та перевіримо з'єднання (“пропінгуємо”) сусідній ПК (мал.1.15). Ping – утиліта для перевірки з'єднань у мережах на основі TCP/IP, яка відправляє запити (ICMP Echo-Request) протоколу ICMP зазначеному вузлу мережі та очікує відповідь (ICMP Echo-Reply). Час між відправкою запиту та отриманням

відповіді (RTT) дозволяє визначати двосторонні затримки (RTT) за маршрутом та частоту втрати пакетів, тобто опосередковано визначати завантаженість на каналах передачі даних та проміжних пристроях. Повна відсутність ICMP-відповідей може також означати, що віддалений вузол (або якийсь із проміжних маршрутизаторів) блокує ICMP Echo-Reply або ігнорує ICMP Echo-Request .



Мал.1.14 Командний рядок



Мал.1.15 Успішне виконання команди ping

TTL – час життя відправленого пакета (визначає максимальну кількість маршрутизаторів, які пакет може пройти при його просуванні мережею);

time - час, який витрачений на відправку запиту та отримання відповіді;

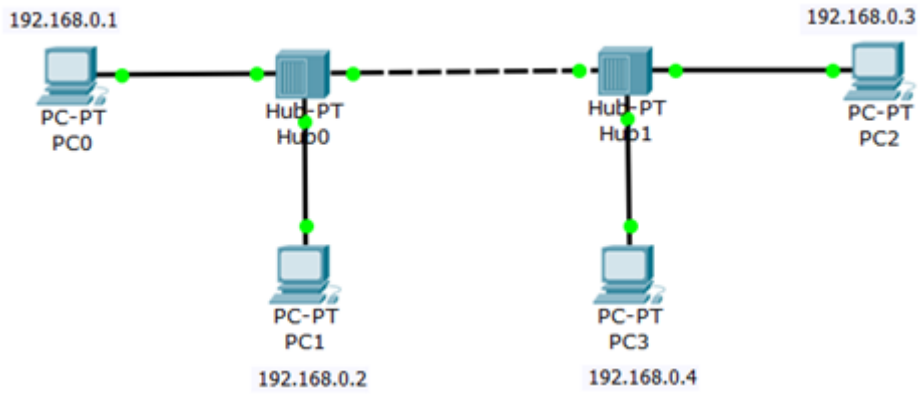
min - мінімальний час відповіді;

max - максимальний час відповіді;

avg - середній час відповіді.

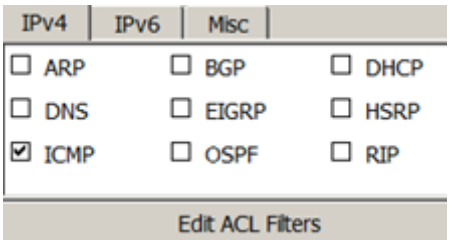
Практична робота 2. Організація режиму симуляції роботи мережі

Спроектуємо у робочій області програми мережу з 4-х ПК та 2-х хабів, налаштувавши IP-адреси для кінцевих пристроїв, як це показано на мал.16 та задавши а маску мережі 255.255.255.0.



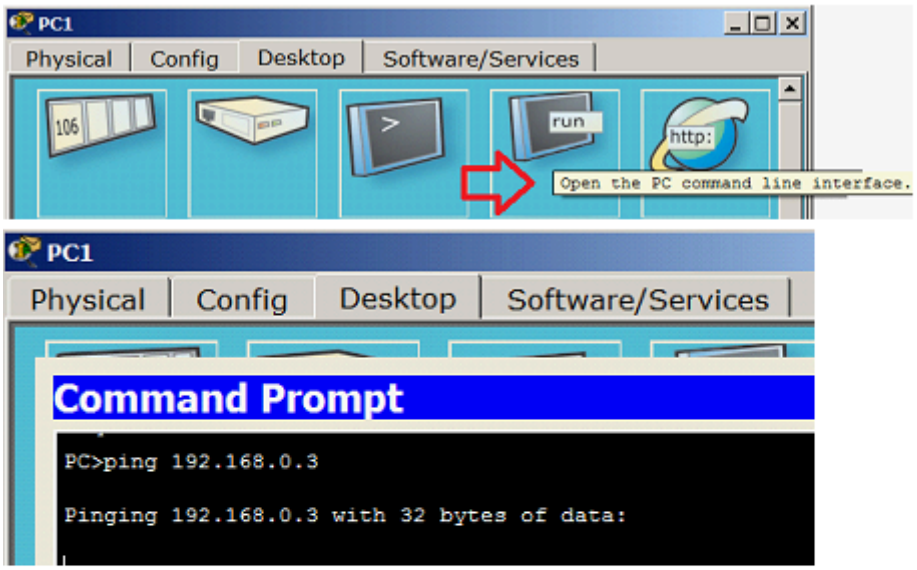
Мал.1.16

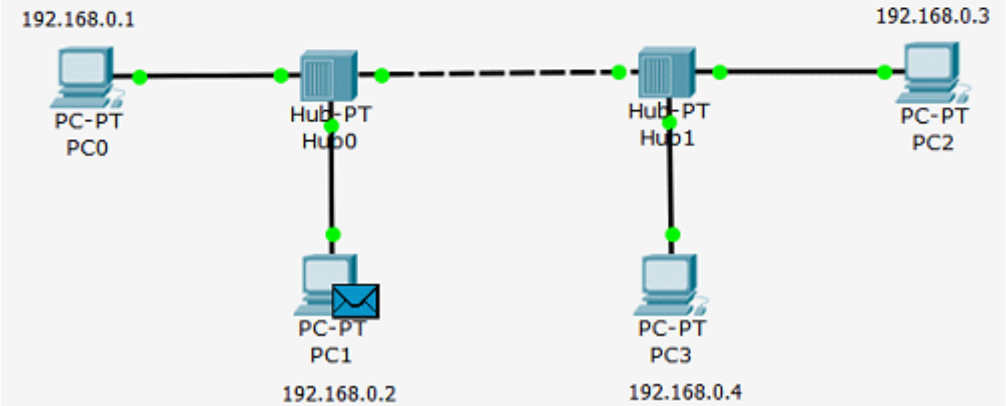
Перейдемо в режим симуляції комбінацією клавіш Shift+S або натиснувши мишкою на значок симуляції у правому нижньому кутку робочого простору. Натисніть кнопку Edit Filters (Змінити фільтри) та виключіть усі мережеві протоколи крім ICMP (мал.1.17).



Мал.1.17 Ознака ICMP активна

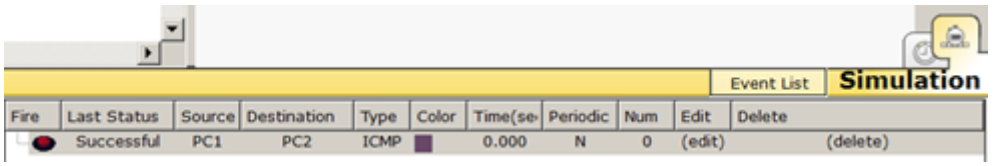
Із одного з хостів спробуємо пропінгувати інший вузол. Наприклад, з PC1 пінгуємо PC2, щоб наочніше побачити, як проходитимуть пакети мережею в режимі симуляції (мал.1.18).





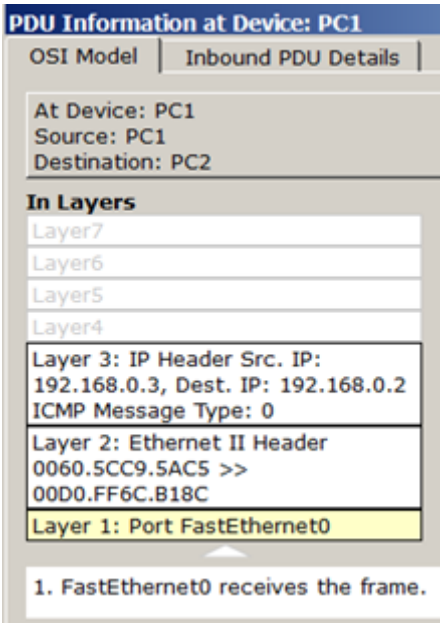
Мал.1.18 Початок процесу пінгування в режимі симуляції

На PC1 утворився пакет (конверт), який чекає на початок його руху мережею. Запустити просування пакета в мережу покроково можна, натиснувши кнопку Capture / Forward (Вперед) у вікні симуляції. Якщо натиснути на кнопку Auto Capture / Play (відтворення) то побачимо весь цикл проходження пакета мережею. В (Список подій) можна побачити успішний результат виконання команди ping (мал.1.19).



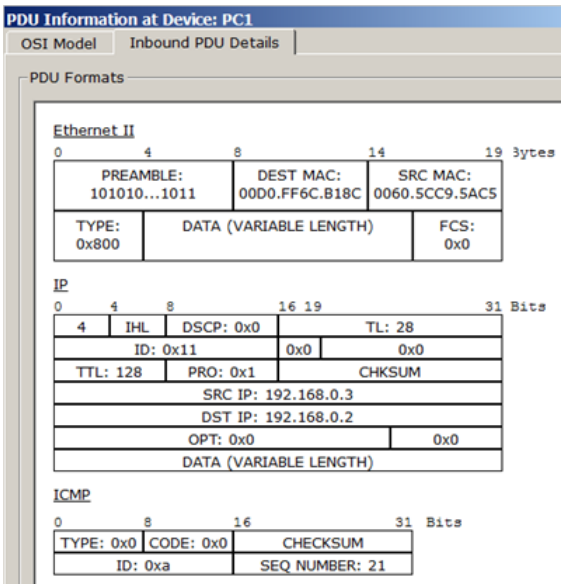
Мал.1.19 Наявність зв'язку між PC1 та PC2

Клацнувши мишкою на конверті, можна побачити додаткову інформацію про рух пакету через мережу. При цьому на першій вкладці відображається модель OSI (мал.1.20), де представлена інформація про рівні OSI, на яких працює даний мережевий пристрій.



Мал.1.20 Моніторинг руху пакета на моделі OSI

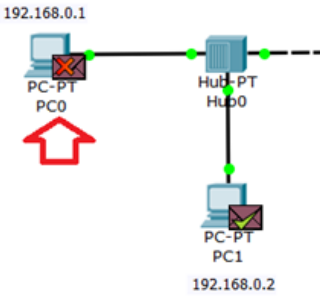
На іншій вкладці можна, можливо подивитися структуру пакета (мал.1.21).



Мал.1.21 Структура пакету

Отже, у Packet Tracer передбачений режим моделювання (*Симуляції*), в якому показується, як працює утиліта *ping*. Щоб перейти в цей режим, необхідно натиснути на значок Simulation Mode (*Симуляція*) у нижньому правому кутку робочої області або комбінацію клавіш Shift+S. Відкриється Simulation Panel (Панель симуляції), в якій будуть відображатися всі події, пов'язані з виконання ping-процесу. Моделювання припиняється або при завершенні ping-процесу або при закритті вікна симуляції. У режимі симуляції можна не тільки відстежувати використовувані протоколи, а й бачити, на якому із семи рівнів моделі OSI цей протокол задіяний. У процесі перегляду анімації було показано принцип роботи хаба. Концентратор (хаб)

повторює пакет на всіх портах, сподіваючись, що на одному з них є одержувач інформації. Якщо пакети деяким вузлам не призначені, ці вузли ігнорують пакети. Якщо пакет повертається відправнику, то можна побачити галочку "прийняття пакета". (мал.1.22).



Мал.1.22 Значки ігнорування пакетів та підтвердження з'єднання

Контрольні запитання

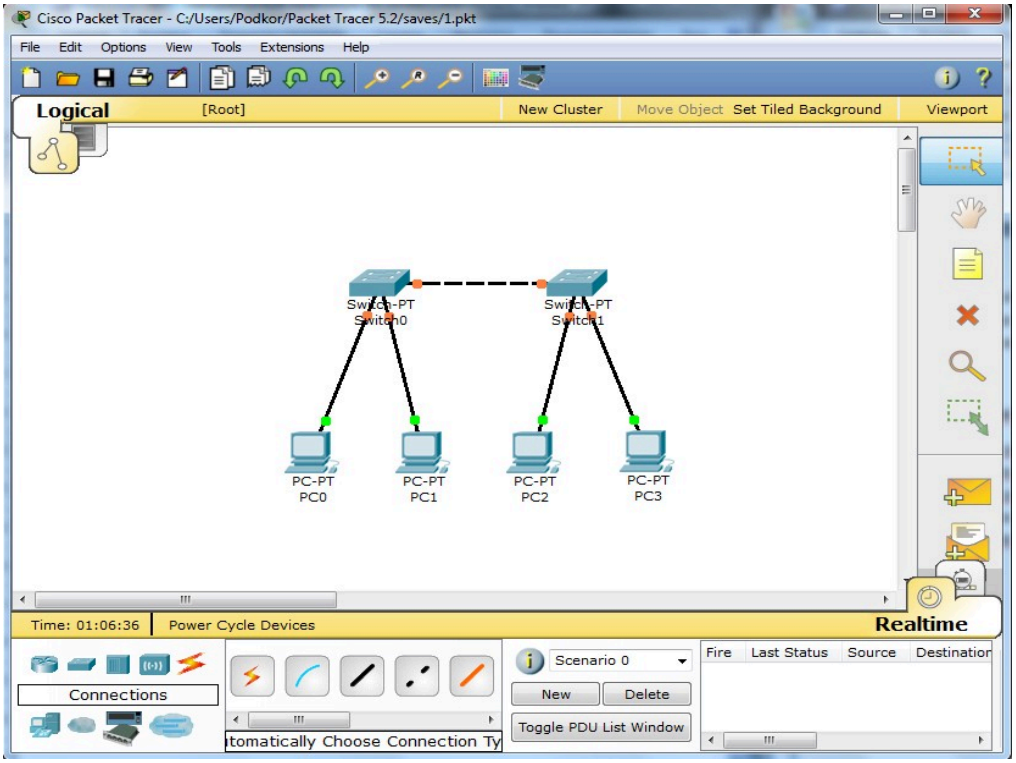
1. Які типи мережевих пристроїв і з'єднань можна використовувати в Packet Tracer ?
2. Яким способом можна перейти до інтерфейсу командного рядка пристрою ?
3. Як додати в топологію і налаштувати новий пристрій ?
4. Як зберегти конфігурацію пристрою в .txt файл ?

Порядок виконання і здачі роботи

1. Одержати індивідуальне завдання.
2. Вивчити теоретичну частину та методику виконання роботи.
3. В середовищі Packet Tracer виконати індивідуальне завдання.
4. Оформити звіт.
5. Відповісти на контрольні запитання.
6. Продемонструвати виконання індивідуального завдання.

Завдання для самостійної роботи

1) Створити топологію, що зображена на мал.1.23



Мал.1.23

2) Призначити комп'ютерам адреси відповідно до варіанту (*v=1-16*), який надається викладачем за допомогою командного рядка

Таблиця 1.1

Пристрій	IP ADDRESS	SUBNET MASK
PC0	10.1.v.1	255.255.255.0
PC1	10.1.v.2	255.255.255.0
PC2	10.1.v.3	255.255.255.0
PC3	10.1.v.4	255.255.255.0

3) Призначити комп'ютерам імена PC*i* (*i=0-3*).

4) Перевірити з'єднання між комп'ютерами.

5) Продемонструвати роботу перевірки з'єднання між комп'ютерами в режимі симуляції.

6) Зберегти топологію мережі.

Зміст звіту

Звіт готується в електронному вигляді та вивантажується в систему електронного навчання moodle. Звіт має містити виконання теоретичного та самостійного завдання з коментарями та скріншотами до кожного етапу.

Лабораторна робота № 2. Основи операційної системи IOS компанії Cisco

Мета роботи: Опанувати базові команди операційної системи IOS та режими роботи пристроїв компанії Cisco.

Методичні вказівки

1. Режими роботи пристроїв

При першому вході у мережевий пристрій користувач бачить командний рядок користувацького режиму:

Switch>

Команди, які доступні на рівні користувача, є підмножиною команд, що доступні в привілейованому режимі. Ці команди дозволяють виводити на екран інформацію без зміни налаштувань мережевого пристрою.

Активізація привілейованого режиму здійснюється командою **enable**. Про перехід в цей режим буде свідчити поява в командному рядку запрошення у вигляді знака #.

Press ENTER to start.

Switch>

Switch> **enable**

Switch#

Switch# **disable**

Switch>

В привілейованому режимі можна отримувати інформацію про налаштування системи та мати доступ до режиму глобального конфігурування та інших спеціальних режимів, включаючи режими конфігурування інтерфейсу, мережевого пристрою, таблиці маршрутизації і т.д. Для виходу із системи IOS необхідно використати команду **exit**

Switch> **exit**

Мережевий пристрій можна перевести в один із можливих режимів, використовуючи консоль термінальної програми або в сеансі протоколу Telnet.

- Користувацький режим – це режим, в якому користувач може лише переглядати певну інформацію про мережний пристрій, але не може нічого змінювати. В цьому режимі запрошення має вигляд *Switch>*.
- Привілейований режим – це режим, який підтримує команди налаштування та тестування, детальну перевірку мережевого пристрою, маніпуляцію з файлами і доступ в режим конфігурування. В цьому режимі запрошення має вигляд типу *Switch#.*

- Режим глобального конфігурування – це режим, в якому використовуються команди для конфігурування специфічних параметрів пристрою. В цьому режимі запрошення має вигляд *Switch (config)#.*

Для входу в режим глобального конфігурування використовується команда привілейованого режиму **configure,** після якої вказується джерело команд конфігурування: *terminal* (термінал), *memory* (енергонезалежна пам'ять або файл), *network* (сервер tftp в мережі). За замовчуванням, команди вводяться з терміналу консолі. Наприклад:

Switch# **configure terminal**

Switch(config)# **(commands)**

Switch(config)# **exit**

2. Основні команди

Команди в будь-якому режимі IOS розпізнає за першими унікальними символами, доповнення яких до повних імен відбувається при натисканні клавіші табуляції. Для одержання списку всіх команд певного режиму необхідно перейти в цей режим і використати команду **?**. Також можна використати символ *'?'* після імені команди для отримання детальнішої інформації про команду та її використання.

Команди режиму глобального конфігурування визначають поведінку системи в цілому, а також включають команди переходу в інші режими конфігурування, які використовуються для створення конфігурацій, що вимагають багаторядкових команд.

Варто зазначити, що команди глобального конфігурування застосовуються раніше команд активізації конфігурації вузького направлення. Наприклад, для конфігурації інтерфейсу, на можливість якої вказує запрошення *Switch(config-if)#*, спочатку вводиться глобальна команда для визначення типу інтерфейсу та номеру його порта:

Switch# **conf t**

Switch(config)# **interface type port**

Switch(config-if)# (**commands**)

Switch(config-if)# **exit**

Для обмеження доступу до системи використовують паролі. Команда **line console** встановлює пароль на вхід терміналу консолі:

Switch(config)# **line console 0**

Switch(config-line)# **login**

Switch(config-line)# **password Cisco**

Команда **line vty 0 4** встановлює парольний захист на вхід за протоколом Telnet:

Switch(config)# **line vty 0 4**

Switch(config-line)# **login**

Switch(config-line)# **password cisco**

Команда **enable password** обмежує доступ до привілейованого режиму:

Switch# **conf t**

Switch(config)# **enable password passw**

Далі Ctrl-Z

Switch# **exit**

...

Switch> **en**

Password: passw

Switch #

Тут пароль *passw* – послідовність латинських символів.

Для встановлення на мережевому інтерфейсі IP адреси використовується команда:

Router(config-if)# **ip address** [ip_address] [subnet_mask],

Router(config-if)# **no shut**

Команда **no shut** (або **no shutdown**) використовується для того, щоб інтерфейс був активним, без цієї команди можливе тимчасове відключення інтерфейсу. Команда **shut** використовується для вимкнення інтерфейсу.

Є певний набір команд, який використовується для контролю правильності функціонування мережевого пристрою та перевірки його стану в будь-який момент часу:

- **show version** – виводить на екран дані про конфігурацію апаратної частини системи, версії програмного забезпечення, імена і джерела конфігураційних файлів і завантажених образів;
- **show running-config –** відображення вмісту активної конфігурації;
- **show interfaces –** відображення даних про всі інтерфейси на пристрої;
- **show protocols –** вивід даних про протоколи третього мережевого рівня;
- **show version** – використовується для отримання типу платформи мережевого пристрою, версії операційної системи, імені файлу образу операційної

системи, часу роботи системи, обсягу пам'яті, кількості інтерфейсів і конфігураційного регістру;

- **show clock** – відображення годинника;
- **show flash** – у флеш-пам'яті мережевого пристрою зберігається файл-образ операційної системи Cisco IOS;
- **show history** - мережевий пристрій за замовчуванням зберігає 10 останніх введених команд: використовуючи команду <ctrl> P або стрілку вгору повертаємося до команд, які були введені раніше, а команди <ctrl> N або стрілка вниз переводять до наступної команди, яка збережена в буфері;
- **show hosts –** відображення списку хостів та IP-адреси всіх їх інтерфейсів;
- **show interfaces –** виведення детальної інформації про кожен інтерфейс;
- **show sessions –** виведення інформації про кожну telnet сесію;
- **show terminal –** відображення конфігураційних параметрів терміналу;
- **show users –** відображення списку всіх користувачів, які приєднані до пристрою термінальними лініями;
- **show controllers –** відображення стану контролерів інтерфейсів.

Протокол віртуального терміналу telnet, що входить до складу протоколів TCP/IP, дозволяє встановити з'єднання між мережевим пристроєм *telnet* клієнта і мережевим пристроєм *telnet* сервера, що забезпечує можливість роботи в режимі віртуального терміналу. *Telnet* використовується для віддаленого управління мережевим пристроєм або для перевірки зв'язку на рівні додатків. Мережеві пристрої Cisco здатні підтримувати одночасно до п'яти вхідних сеансів протоколу Telnet.

3. Команди діагностики мережевого з'єднання

Для діагностики можливості встановлення зв'язку в мережах використовуються протоколи ехо-пакетів, результати роботи яких можуть допомогти в оцінці надійності шляху до іншого пристрою, величин затримок як до кінцевого пристрою так і до проміжних пристроїв.

Команда **ping** надсилає ICMP (Internet Control Message Protocol) ехо-пакети для перевірки з'єднання. Якщо час проходження ехо-пакету перевищує заданий час, то висвічується не (!*)*, а крапка (.).

Switch> **ping 10.1.1.3**

Sending 5100-byte ICMP echoes to 10.1.1.3 timeout is 2 seconds:

!.!!

Success rate is 80 percent, round-trip min / avg / max = 6/6/6 ms

Таблиця 2.1. Результати роботи команди ping

Символ	Значення
!	Успішний прийом ехо-відповіді
.	Перевищено час очікування
U	Пункт призначення недосяжний
C	Перевантаження мережі

У привілейованому режимі підтримується розширена версія команди **ping**. Важливо зазначити, що можна, перебуваючи на одному пристрої, перевіряти зв'язок між мережевими інтерфейсами на інших пристроях.

Команда **traceroute** показує адреси проміжних інтерфейсів на шляху пакетів до пункту призначення та їх досяжність.

Switch> **traceroute 172.16.101.1**

4. Протокол CDP (Cisco Discovery Protocol)

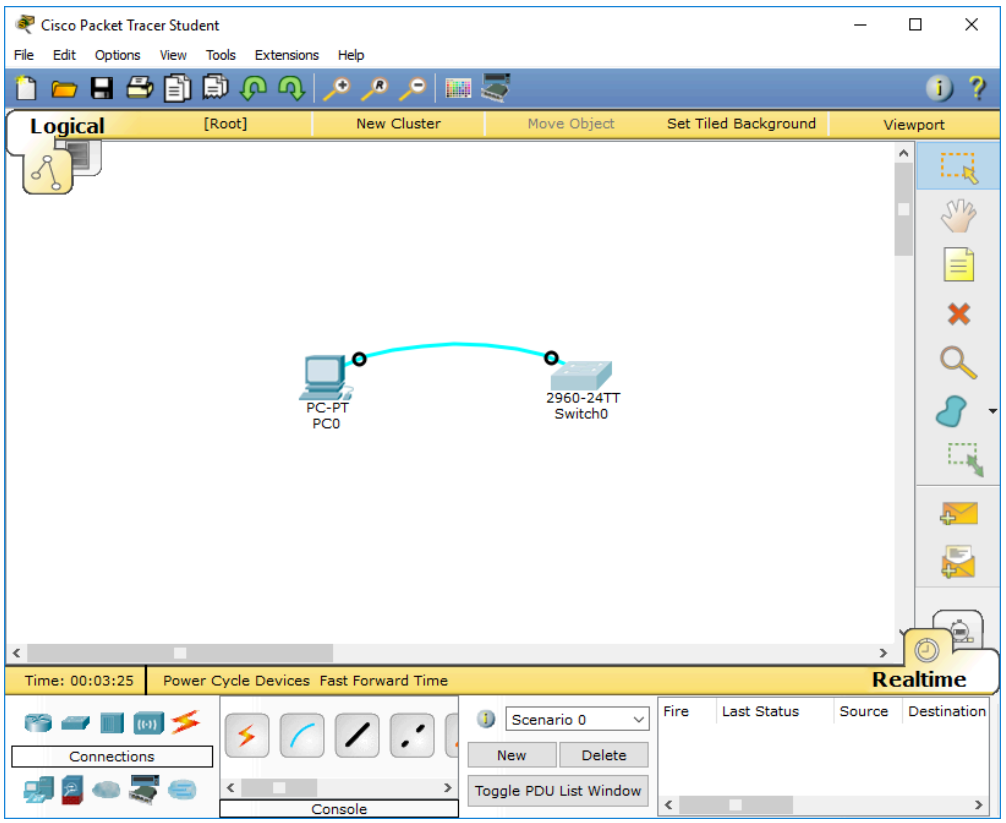
CDP дозволяє пристроям обмінюватися основною конфігураційною інформацією, працюючи на канальному рівні моделі OSI без налаштування будь-якого іншого протоколу. За замовчуванням, CDP включений на всіх інтерфейсах і працює тільки із безпосередньо підключеними пристроями. При запуску пристрою протокол CDP запускається автоматично і може автоматично визначити сусідні пристрої, на яких також виконується даний протокол. Варто зазначити, що серед знайдених пристроїв можуть бути пристрої, які працюють не тільки з протоколом IP.

CDP дозволяє адміністраторам мати доступ до даних про інший мережевий пристрій, до якого є безпосереднє з'єднання. Для виведення інформації про сусідні пристрої, які працюють з протоколом CDP, використовується сімейство команд **show cdp,** які виводять дані про кожен порт та пристрій, який підключений до нього (ідентифікатори пристрою, список адрес, ідентифікатор порту та ін.).

Хід роботи

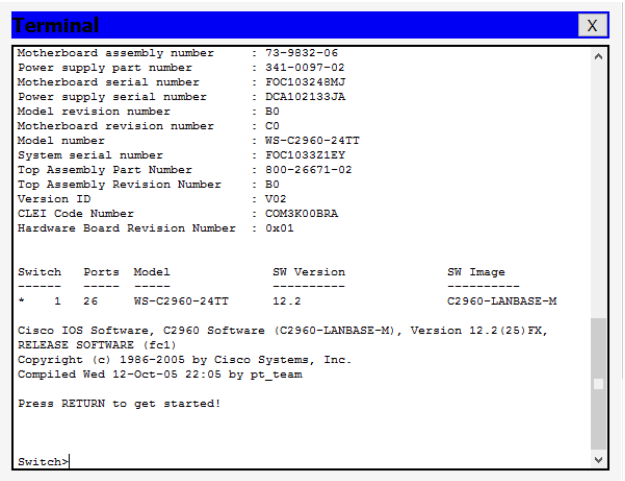
Практична робота 1. З'єднання кінцевого пристрою з комутатором Cisco та його налаштування

1. Створимо у Packet Tracer топологію, яка зображена на мал.2.1.



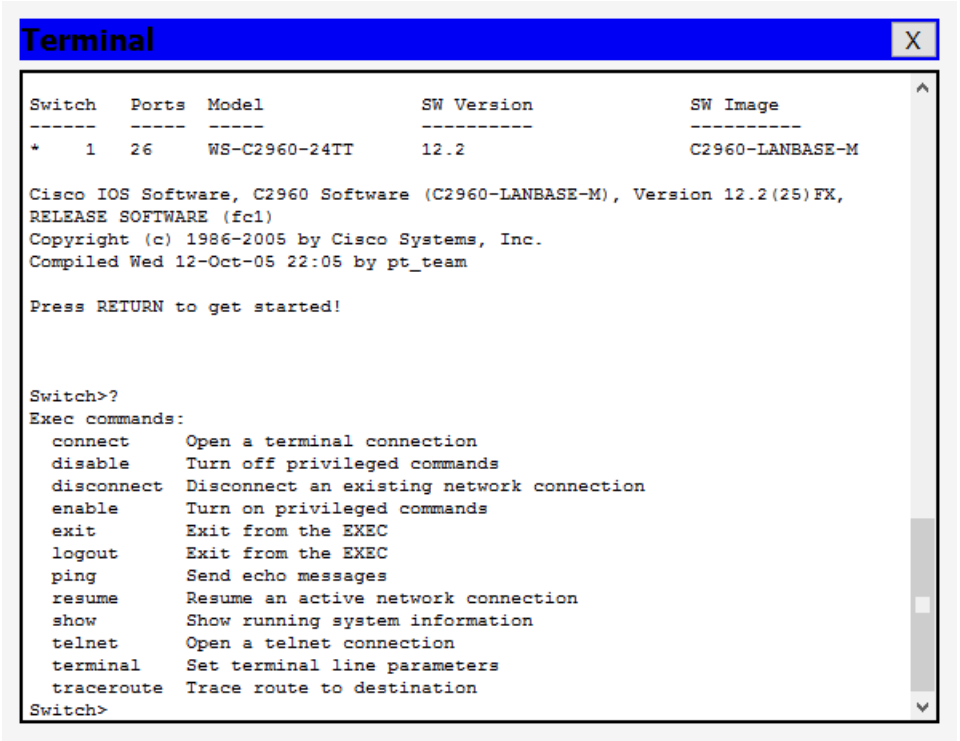
Мал.2.1

2. Відкриємо за допомогою терміналу вікно налаштувань мережевого пристрою як на мал.2.2



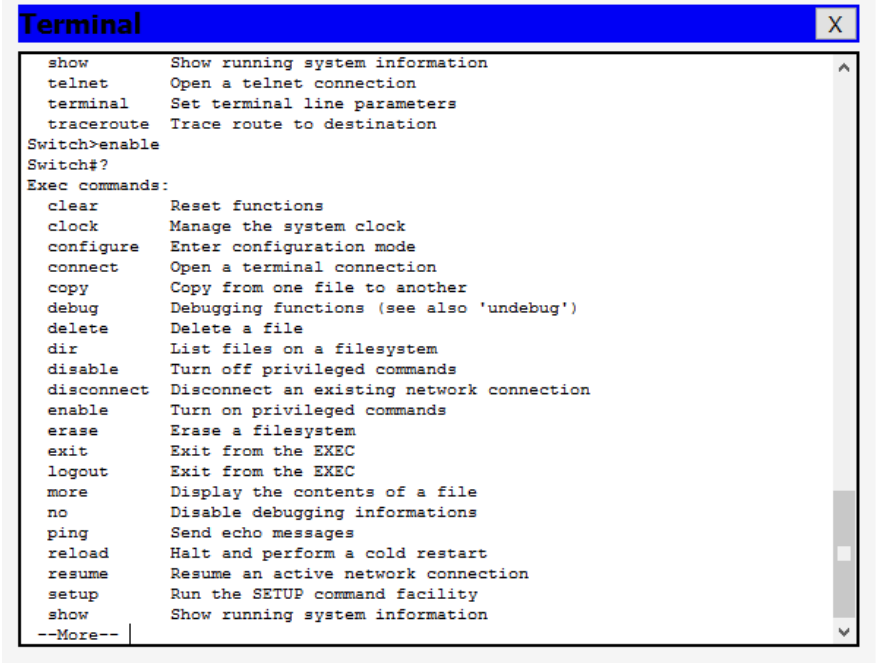
Мал.2.2

3. Переглянемо список доступних команд на мал. 2.3.



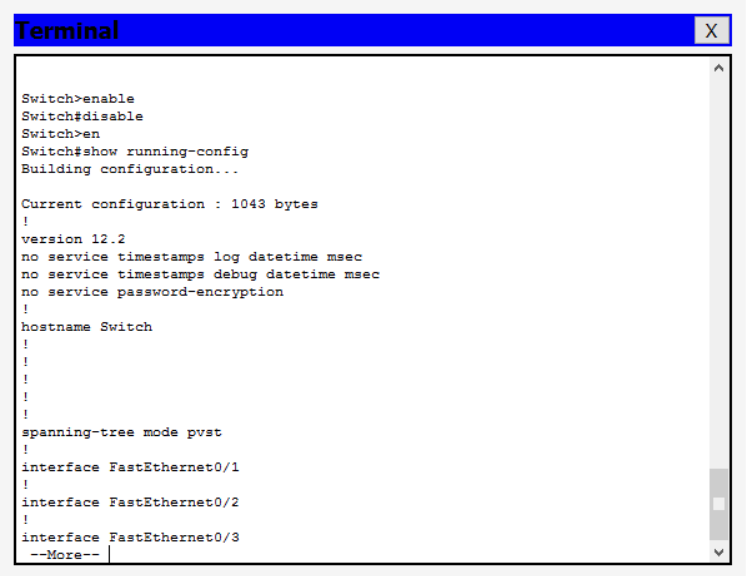
Мал.2.3

4. Перейдемо до привілейованого режиму та переглянемо список доступних команд (мал. 2.4).



Мал.2.4

5. Повернемося до режиму користувача, ввівши команду **disable**. Після цього знову перейдемо в привілейований режим та переглянемо активну конфігурацію в пам'яті мережевого пристрою за допомогою команди **show running-config** або **show run (**мал. 2.5**).** Активна конфігурація автоматично не зберігається і буде втрачена при збої електроживлення.



Мал.2.5

6. На мережевому пристрої увійдемо в режим глобальної конфігурації (мал. 2.6).

```
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#
```

Мал.2.6

7. У режимі глобального конфігурування встановлюємо пароль на вхід до привілейованого режиму, використовуючи командою **enable password cisco** (мал. 2.7).

```
Switch(config)#enable password cisco
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#exit

Switch>enable
Password:
Switch#
```

Мал.2.7

8. Переглянемо конфігурацію та переконаємося, що пароль зберігається у відкритому вигляді (мал. 2.8).

```
Switch#show run
Building configuration...

Current configuration : 1067 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Switch
!
enable password cisco
!
!
```

Мал.2.8

9. Використовуємо команду **service password-encryption** для шифрування паролю та переконуємося, що пароль зберігається у зашифрованому вигляді (мал. 2.9).

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#service pa
Switch(config)#service password-encryption
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#show run
Building configuration...

Current configuration : 1073 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname Switch
!
enable password 7 0822455D0A16
```

Мал.2.9

10. Для безпеки використаємо команду **enable secret** + *кодове_слово* та переконаємося, що даний пароль зберігається у зашифрованому вигляді та буде мати більший пріоритет ніж пароль, який встановлений командою **enable password**. Для цього використати команду **do show run** без виходу до користувацького режиму (мал. 2.10).

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#enable secret cisco2
Switch(config)#
```

```
Switch(config)#do show run
Building configuration...

Current configuration : 1120 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
```

Мал.2.10

11. Використовуючи команду **user privilege password (**з рівнем привілей від 0 до 15**)** створюємо користувача та зберігаємо дані про нього в локальній базі даних (мал. 2.11).

```
Switch(config)#
Switch(config)#user admin privilege 15 password cisco
Switch(config)#
```

Мал.2.11

12. Встановлюємо авторизацію на підключення до консолі:
    - переходимо в режим конфігурування термінальних ліній, використовуючи команду **line console 0** (мал. 2.12)**;**
    - для того щоб використовувати локальну базу для перевірки, виконуємо команду **login local**.

```
Switch(config)#line console 0
Switch(config-line)#?
Line configuration commands:
  access-class  Filter connections based on an IP access list
  databits      Set number of data bits per character
  default       Set a command to its defaults
  exec-timeout  Set the EXEC timeout
  exit          Exit from line configuration mode
  flowcontrol   Set the flow control
  history       Enable and control the command history function
  ipv6          IPv6 options
  logging       Modify message logging facilities
  login         Enable password checking
  motd-banner   Enable the display of the MOTD banner
  no            Negate a command or set its defaults
  parity        Set terminal parity
  password      Set a password
  privilege     Change privilege level for line
  speed         Set the transmit and receive speeds
  stopbits      Set async line stop bits
  transport     Define transport protocols for line
Switch(config-line)#
```

Мал.2.12

13. Повертаємося до користувацького режиму та входимо в привілейований режим.

```
User Access Verification

Username: admin
Password:

Switch#
```

14. Встановлюємо *IP-адресу* для *Vlan1* на комутаторі *Switch* та виконуємо команду **no shutdown**, щоб переконатися в тому, що інтерфейс піднятий (мал. 2.13).

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface Vlan1
Switch(config-if)#?
Interface configuration commands:
  arp          Set arp type (arpa, probe, snap) or timeout
  description  Interface specific description
  exit         Exit from interface configuration mode
  ip           Interface Internet Protocol config commands
  ipv6         IPv6 interface subcommands
  no           Negate a command or set its defaults
  shutdown     Shutdown the selected interface
  standby      HSRP interface configuration commands
Switch(config-if)#ip addres 198.168.0.1 255.255.255.0
Switch(config-if)#no shutdown

Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan1, changed state to up
```

Мал.2.13

15. Налаштовуємо режим віртуальних термінальних ліній та встановлюємо авторизацію на вхід через Telnet сервіс, використовуючи локальну базу.

```
Switch(config)#line vty 0 4
Switch(config-line)#transport input telnet
Switch(config-line)#login local
```

16. Зберігаємо конфігурацію, використовуючи команду **write memory.**
17. З'єднуємо комп'ютер та *Switch* з використанням *Ethernet***.**
18. Перевіряємо зв'язок, використовуючи команду **ping** (мал. 2.14)**.**

```
Pinging 198.168.0.1 with 32 bytes of data:

Reply from 198.168.0.1: bytes=32 time=1ms TTL=255
Reply from 198.168.0.1: bytes=32 time=0ms TTL=255
Reply from 198.168.0.1: bytes=32 time=0ms TTL=255
Reply from 198.168.0.1: bytes=32 time=0ms TTL=255

Ping statistics for 198.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Мал.2.14

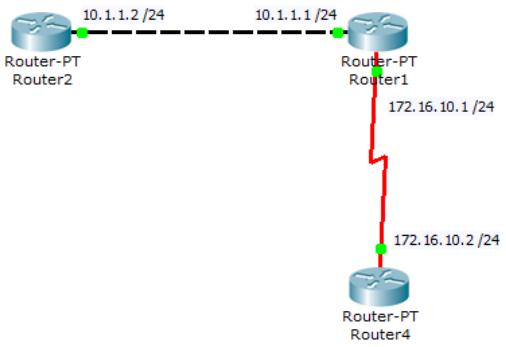
19. Встановити *telnet*-з'єднання (мал. 2.15).

```
PC>telnet 198.168.0.1
Trying 198.168.0.1 ...Open


User Access Verification

Username: admin
Password:
Switch#
```

Мал.2.15

Практична робота 2. З'єднання кінцевого пристрою з маршрутизатором Cisco та його налаштування.

1. Створимо у Packet Tracer топологію, яка зображена на мал.2.16, з використанням моделі маршрутизатора за замовчуванням *Generic*.



Мал.2.16

2. Дамо імена пристроям так, як на рис.16: *Router1, Router2 і Router4*. Чорна лінія означає Ethernet з'єднання, а червона лінія - послідовне з'єднання. Для створення послідовного з'єднання вибираємо послідовне з'єднання точка-точка (serial cable). Визначаємо, який маршрутизатор буде виконувати функції DCE пристрою так як даний пристрій буде задавати синхронізацію. У симуляторі для нього потрібно визначити частоту синхронізації. Збережемо топологію.

3. Відкриємо вікно налаштувань мережевого пристрою *Router1* і виберемо вкладку CLI для управління пристроєм.

4. У середині екрану пристрою *Router1* після одержання повідомлення "*Continue with configuration dialog? [Yes / no]:"* потрібно ввести "no" для переходу в командний рядок режиму користувача.

5. Перейдемо до привілейованого режиму: *Router>* **enable.**

6. Повернемося до режиму користувача, ввівши команду **disable**. Перебуваючи в режимі користувача, введемо **logout** або **exit**, щоб залишити мережевий пристрій.

7. Увійдемо до мережевого пристрою *Router1 (Router>)* і введемо команду **?** для перегляду списку всіх доступних команд: *Router>* **?** .

8. Перейдемо до привілейованого режиму: *Router>* **enable.**

9. Переглянемо список доступних команд: *Router #* **? .**

10. Перейдемо в режим конфігурації: *Router#* **config terminal .**

11. Ім'я хоста мережевого пристрою використовується для локальної ідентифікації. Коли користувач входить в мережевий пристрій, з'являється

повідомлення *Ім'я хоста* перед символом режиму (">" або "#"). Назначимо ім'я "*Router1*" як ім'я мережевого пристрою: *Router(config)#* **hostname Router1**.

12. Пароль доступу дозволяє контролювати доступ у привілейований режим. Встановимо пароль доступу "*parol*": *Router1(config)#* **enable password parol**.

13. Перевіримо роботу цього паролю, вийшовши з мережевого пристрою і спробувавши зайти в привілейований режим.

Router1> ***en***

Password: *****

14. Перейдемо до користувацького режиму командою **disable** і введемо команду для перегляду всіх доступних show команд: *Router1>* **show ?**.

15. Перейдемо в привілейований режим: *Router1>* **en**.

16. Введемо команду для перегляду всіх доступних show-команд: *Router1#* **show ?**.

Привілейований режим включає в себе всі show-команди режиму користувача і деякі нові.

17. Переглянемо активну конфігурацію в пам'яті мережевого пристрою: *Router1#* **show running-config.** Активна конфігурація автоматично не зберігається і буде втрачена при збою електроживлення.

18. Введемо команду, яка дозволить побачити поточний стан протоколів третього рівня: *Router1#* **show protocols**.

19. На мережевому пристрої *Router1* увійдемо в режим конфігурації:

Router1# **conf t.**

20. Налаштуємо Ethernet-інтерфейс, перейшовши в режим конфігурації інтерфейсу: *Router1(config)#* **interface FastEthernet0/0 (**або **int fa0/0).**

21. Переглянемо всі доступні в цьому режимі команди: *Router1(config-if)#* **?.**

Для виходу в режим глобальної конфігурації використовується команда **exit**.

22. Для кожної команди можна виконати протилежну команду, поставивши перед нею слово **no**. Наприклад, команда *Router1(config-if)#* **no shutdown** включає інтерфейс.

23. Додамо до інтерфейсу опис: *Router1(config-if)#* **description Ethernet interface on Router 1.**

Щоб побачити опис цього інтерфейсу, потрібно перейти в привілейований режим і виконати команду **show interface**.

24. Аналогічним чином приєднаємось до мережевого пристрою *Router 2*, замінимо ім'я його хоста на *Router2* і увімкнемо інтерфейс *FastEthernet 0*.

25. Розглянемо конфігурації послідовних інтерфейсів. Перейдемо до *Router1*. Перевіримо, яким пристроєм виступає наш маршрутизатор для послідовної лінії зв'язку: кінцевим пристроєм DTE (data terminal equipment) чи пристроєм зв'язку DCE (data circuit): *Router1#* **show controllers S2/0.**

Якщо бачимо - ..... DCE cable ..... -, то маршрутизатор є пристроєм зв'язку і він повинен задавати частоту синхронізації тактових імпульсів, що використовуються при передачі даних. Частота береться з певного ряду частот. Виберемо частоту 64000 і піднімемо інтерфейс.

Router1# **conf t**

Router1(config)# **int s2/0**

Router1(config-if)# **clock rate ?**

Router1(config-if)# **clock rate 64000**

Router1(config-if)# **no shut**

26. Переходимо до маршрутизатора *Router4* і задамо однойменне ім'я. Піднімаємо на ньому інтерфейс *serial2/0*. Тепер, коли інтерфейси на двох кінцях нашого послідовного з'єднання включені, на екрані з'явиться повідомлення про зміну стану інтерфейсу на активне.

27. Перевіримо інтерфейси *s2/0(Router1#), e0/0 (Router1#), e0/0 (Router2#), s2/0 (Router4#)* на кожному пристрої за допомогою команди **sh int.**

28. На маршрутизаторі *Router1* введемо команду для виведення стану всіх інтерфейсів, на яких працює CDP: *Router1#* **show cdp interface.** Ми повинні побачити, що обидва інтерфейсу підняті і посилають CDP пакети.

29. Переконавшись, що мережевий пристрій посилає і приймає CDP-пакети, можна використовувати CDP для отримання інформації про безпосередньо підключені пристрої, використовуючи команду *Router1#* **show cdp neighbors.** Для більш детальної інформації про сусідів використовується команда *Router1#* **show cdp neighbors detail.**

30. Щоб дізнатися інформацію про пристрій "*Router4*" введемо на *Router1* команду: *Router1#* **show cdp entry Router4.**

31. На пристрої *Router1* введемо команду для того, щоб побачити, як часто *Router1* посилає сусідам CDP- оновлення і як довго у сусідів вони повинні зберігатися: *Router1#* **show cdp.**

32. Відключення та включення підтримки CDP протоколу відбувається за допомогою команд *Router1(config)#* **no cdp run** та *Router1(config)#* **cdp run.** Можна відключити підтримку протоколу CDP для певного інтерфейсу.

Router1(config)# **interface fa0/0**
Router1(config-if)# **no cdp enable**
Router1(config)# **Ctrl-Z**
Router1# **show cdp interface**

В отриманому повідомленні не відображатимуться відомості про *FastEthernet 0/0*.

33. Підключимося до пристрою *Router1* (*Router1(config)#* **interface fa0/0**) і встановимо IP адресу Ethernet інтерфейсу, використовуючи команду *Router1(config-if)#* **ip address 10.1.1.1 255.255.255.0**.

34. Призначимо інтерфейсу S0/0 IP адресу 172.16.10.1 255.255.255.0, не виходячи з конфігурації інтерфейсу

Router1(config-if)# **in s0**

Router1(config-if)# **ip ad 172.16.10.1 255.255.255.0**

На послідовне з'єднання точка-точка завжди виділяється ціла підмережа.

35. Аналогічним чином, переключимося до пристрою *Router2* і призначимо інтерфейсу FastEthernet 0/0 IP адресу 10.1.1.2 255.255.255.0.

36. Підключимося до пристрою *Router4* і встановимо IP адресу 172.16.10.2 255.255.255.0 для інтерфейсу *Serial 2/0*.

37. На кожному пристрої переглянемо активну конфігурацію і переконаємося, що там з'явилися призначені IP адреси, наприклад: *Router1#* **show running-config**.

38. Переглянемо детальну IP інформацію про кожен інтерфейс і переконаємося, що сконфігуровані інтерфейси перейшли в стан UP, використовуючи команду **show ip interface.**

39. Стислу інформацію можна отримати командою **show ip interface brief**, наприклад: *Router1#* **show ip in bri.**

40. Переконуємося, що інтерфейси на всіх пристроях налаштовані коректно та існує з'єднання між ними, використовуючи команду **ping,** наприклад: *Router1#* **ping 10.1.1.2.**

41. Повернемося на *Router2*. Спробуємо пропінгувати адресу 172.16.10.1 інтерфейсу *Serial 2/0* на пристрої *Router1*. Невдача. Спробуємо пропінгувати адресу 172.16.10.2 інтерфейса *Serial 2/0* на пристрої Router4. Аналогічно. Невдачі нас спіткали тому, що не налаштована на маршрутизаторах маршрутизація.

42. Перейдемо до пристрою *Router1*. Мережеві пристрої підтримують до 5 ліній для прийому telnet-сесії, які призначені на віртуальні термінали vty. Вкажемо, що будемо використовувати всі 5 ліній, використавши команду *Router1(config)#* **line vty 0 4.**

43. Повідомимо мережевому пристрою, що нам знадобиться пароль входу у систему:

Router1(config-line)# **login**

Router1(config-line)# **password parol**

44. Перейдемо до пристрою *Router2* і встановимо telnet-з'єднання з пристроєм *Router1*. Для цього потрібно використати IP адресу його інтерфейсу FastEthernet 0/0: *Router2#* **telnet 10.1.1.1.**

45. Введемо пароль *parol,* після чого встановиться telnet-з'єднання з *Router1*. Команда *Router1>* **show user** покаже, що з'єднання здійснено з адреси 10.1.1.2 пристрою *Router2*.

Для повернення до пристрою *Router2* потрібно натиснути одночасно клавіші control-shift-6, і, відпустивши, відразу натиснути клавішу x.

46. Для відображення всіх активних telnet-сесії введемо команду **show sessions**. Для відновлення telnet-сесії потрібно використати команду **resume** *номер_сесії*, номер якої можна отримати, використовуючи команду *Router2#* **show sessions.** Завершення сесії здійснюється командою *disconnect номер_сесії.*

47. Збережемо проект в цілому і конфігурацію кожного роутера окремо.

Контрольні запитання

1. Які існують режими команд у командному рядку (CLI), як між ними можна перемикатися ?

2. Як увійти в привілейований режим та режим глобальної конфігурації, та як вийти з цих режимів ?

3. Яке призначення протоколу CDP, які можливості він надає для адміністратора пристрою ?

4. Яку інформацію повертають команди **ping** та **traceroute** ?

5. Чи можна, перебуваючи на одному пристрої, попарно пропінгувати всі пристрої в мережі ?

6. Яке призначення протоколу *telnet* ?

7. Як задати ім'я хоста ?

8. Яку інформацію можна подивитися за допомогою команди show в привілейованому режимі, але не можна подивитися в режимі користувача ?

9. На якому пристрої при послідовному з'єднанні можна встановлювати частоту синхронізації ?

10. Як перевести інтерфейс до стану UP та переконатися в цьому ?

11. Як призначити IP адресу на інтерфейс і переконатися, що вона призначена ?

12. Чому при виконанні команди **ping** може виникати невдача ?

13. Як організувати, призупинити, відновити та закрити telnet-сесію ?

Порядок виконання і здачі роботи

1. Вивчити теоретичну частину.

2. Виконати практичну роботу 1 та практичну роботу 2. Результати кожного кроку мають бути підкріплені скріншотом. При заданні ip-адреси використовуєте адреси 172.16.v.1 та 192.168.v.0 з маскою 255.255.255.0, де v - номер вашого варіанту, який співпадає з порядковим номером в групі.

3. Відповісти на контрольні запитання.

4. Оформити звіт.

5. Продемонструвати набуті вміння та навички при захисті виконаної роботи.

Зміст звіту

Звіт готується в електронному вигляді та вивантажується в систему електронного навчання moodle. Звіт має містити результати виконання теоретичного та самостійного завдання з коментарями та скріншотами до кожного етапу.

Лабораторна робота № 3. Статична та динамічна маршрутизація. Безкласова адресація та маски змінної довжини.

Мета роботи: Засвоїти налаштування статичної та динамічної маршрутизації при проектуванні комп'ютерних мереж різної складності з використанням пристроїв компанії Cisco, включаючи використання масок змінної довжини.

Методичні вказівки

1. Схеми адресації протоколу IP

Комп’ютерна мережа — сукупність комп’ютерів й інших пристроїв, з’єднаних між собою для обміну даними та спільного використання пристроїв. Потреба у схемах адресації виникає внаслідок можливої зміни структури комп'ютерних мереж.

На даний час проблема розподілу адрес гостро стоїть в зв'язку з експоненціальним ростом кількості пристроїв в мережі Інтернету.

Варто зазначити, що стек протоколів TCP/IP при початковому впровадженні передбачав наявність якісного зв’язку та базувався на дворівневій адресній схемі для унікальної ідентифікації комп’ютерів в мережі.

На сьогоднішній час розглядають дві версії протокола IP:

- IPv4 (далі просто IP): адреса складається з 4 байт - октетів (мал.3.1), максимум може бути 4 294 967 296 IP-адрес даного типу. Для зручності користувацького сприйняття IP-адреса записується у вигляді чотирьох десяткових чисел (кожному октету ставиться у відповідність десяткове число), які розділені крапками;
- IPv6 - це сучасна версія інтернет-протоколу, яка на відміну від 32-бітних адрес IPv4 використовує 128-бітні адреси і може мати значно більше адрес аніж IPv4. Максимальна кількість адрес даного типу складає 340 секстильйонів (3,4x1038). Також IPv6 допомагає запобігти підміні IP-адрес (IP-спуфінгу).

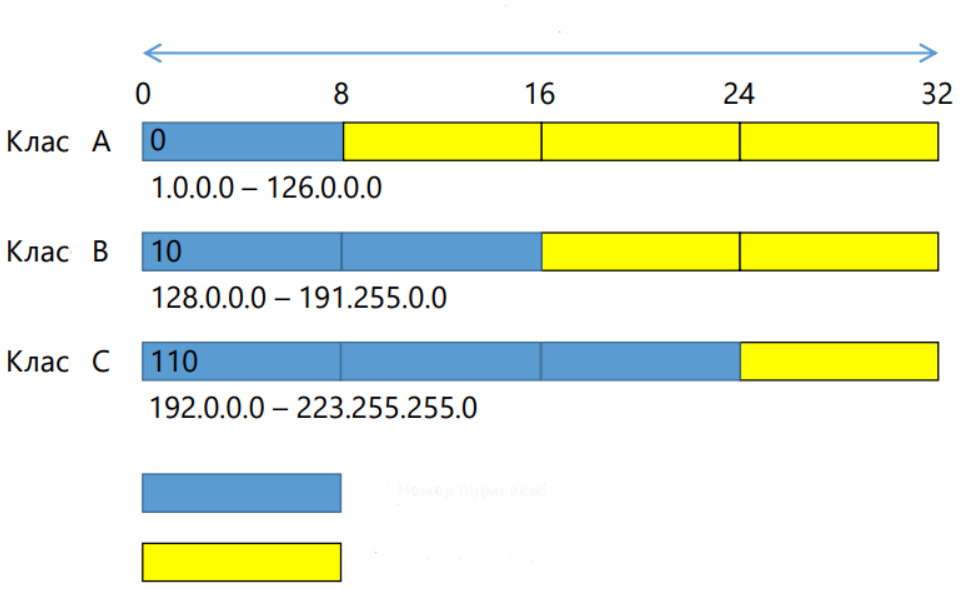
З точки зору протоколу IPv4 IP-мережа (subnet) - множина комп’ютерів, у якої старша частина IP-адреси однакова, так як старша частина 32-бітового IP-адресу визначає номер (адресу) мережі, молодша – номер хоста. Маршрутизатори, як правило, використовують мережеву частину адреси для організації зв’язку між хостами з різних мереж.

00100000.10001100.00011011.00000001 ~ 32.140.27.1

Мал.3.1

Для того, щоб реалізувати маршрутизацію (перенесення пакету з однієї IP-мережі до іншої) необхідно реалізувати механізм виділення у заданій IP-адресі адресу мережі та адресу хоста. На початку використання TCP/IP для вирішення цього питання використовувалась класова система адресації, яка полягала в тому, що IP-адреса відносилась до одного із п’яти класів, що не перетинаються. Розбиття IP-простору здійснювалось відповідно до значення декількох перших біт в першому октеті.

Якщо перший біт в першому октеті дорівнював нулю, то дана адреса відноситься до класу А та адреса мережі розміщується в першому октеті. Адреси класу В починалися з 10, а для адресації мережі використовуються перший та другий октети. Адреси класу С починалися з бінарних 110 та для адресації мережі використовуються перший, другий і третій октети. Використання класів D (групові адреси 224.0.0.0 – 239.255.255.255) та Е (зарезервовано 240.0.0.0 – 255.255.255.255) мало свої особливості. (мал.3.2).



Мал.3.2

Адреси класів А і В складають приблизно 75 відсотків адресного простору. Кількість мереж класів А і В приблизно рівне 17000. Адреси класу С складають близько 12,5 відсотка адресного простору, а кількість мереж даного класу приблизно дорівнює 2,1 мільйона. Однією із незручностей щодо використання мереж класу С є обмеження кількості хостів (254 адреси),що не відповідає потребам великих організацій.

На даний час при роботі з мережами класи ігноруються та використовується безкласова IP-адресація, яка базується на масках підмереж, які є необхідним

доповненням до IP-адреси.

З точки зору двійкової системи числення маска підмережі - це двійкове число, яке містить послідовність бінарних одиниць, які переходять у послідовність бінарних нулів. Загальна довжина маски підмережі 32 біти. Маски прийнято записувати в десятковій формі подібно IP-адресам. Наприклад,

11111111111111110000000000000000->11111111.1111111.0000000.0000000->255.255.0.0

Маска підмережі "розділяє" IP-адресу на номер мережі та номер хоста. Якщо біт в IP-адресі відповідає одиничному біту у масці, то цей біт в IP-адресі відноситься до номеру мережі, а якщо біт в IP-адресі відповідає нульовому біту в масці, то цей біт в IP-адресі відноситься до номера хоста. Наприклад:

	десяткове представлення	двійкове представлення
Адреса IP:	214.181.195.1	11010110.10110101.11000011.00000001
		and
Маска підмережі 255.255.255.0:		11111111.11111111.11111111.00000000
Адреса підмережі 214.181.195.0		11010110.10110101.11000011.00000000

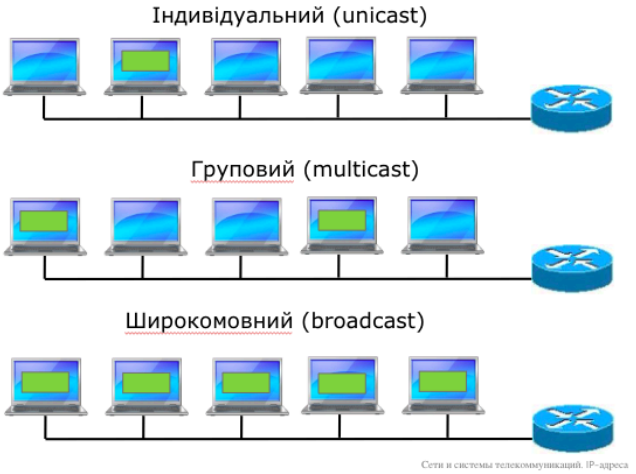
Інша форма запису маски - /N, де N – число одиниць в масці. Ця форма використовується тільки разом з IP-адресою, як наприклад 215.181.195.1/24. Варто зазначити, що обидва представлення еквіваленті.

Всі адреси класу A мають маску 255.0.0.0, адреси класу B мають маску 255.255.0.0, а адреси класу C мають маску 255.255.255.0. Обернене твердження не завжди вірне, оскільки при визначенні класу використовуються перші біти в першому октеті адреси.

Наприклад, якщо адреса 135.124.10.5 належить організації, то номером мережі класу B буде 135.124.0.0, а номер внутрішньо-корпоративної підмережі - 135.124.10.0. (дана підмережа не є мережею класу C).

Проте класова IP-адресація, навіть з використанням підмереж, не може задовольнити потреби за масштабованістю для всієї Інтернет – спільноти і вже на початку 90-х років майже всі мережі класу B були розподілені, а додавання в Інтернет нових мереж класу C приводило до значного росту таблиць маршрутизації та перевантаження маршрутизаторів.

Якщо число нулів у масці дорівнює M, то число доступних адрес хостів в підмережі дорівнює 2^M-2, тобто дві адреси в підмережі використовувати не рекомендується. Одна з цих адрес, останні M біт яких дорівнюють нулю, називається адресою підмережі, а інша з цих адрес, у якої останні M біт дорівнюють одиниці, називається широкомовною адресою (мал. 3.3). Наприклад, для адреси 170.124.105.5/24 адреса підмережі дорівнює 170.124.105.0, а широкомовна адреса дорівнює 170.124.105.255. Число адрес в підмережі дорівнює 2^8-2=254.



Мал. 3.3

Виділяють спеціальні адреси:

- 0.0.0.0 - поточний хост (підмережа);
- 255.255.255.255 - всі хости в поточній підмережі (обмежена широкомовна адреса, яка дозволяє охопити всі пристрої в мережі);
- 127.0.0.0/8 - зворотна петля (loopback), мережа для тестування - дані не передаються в мережу, а приходять назад;
- 127.0.0.1 - localhost (поточний комп'ютер);
- 169.254.0.0/16 - Link-local адреси. Дані адреси призначаються ОС хоста автоматично, якщо недоступна інша конфігурація IP, вони можуть використовуватися в межах підмережі.

Загальну координацію та управлінням IP адрес, які повинні бути унікальні в усьому світі, здійснює некомерційна американська організація IANA - Internet Assigned Numbers Authority - «Адміністрація адресного простору Інтернет». Кореневими доменними серверами управляє ICANN – Internet Corporation for Assigned Names and Numbers. Ця некомерційна американська організація підтримує безперебійну роботу 13-ти кореневих серверів, а також надає права відповідальним за доменні зони

верхнього рівня - ua, com та інші. У кожній країні є своя організація, відповідальна за зону верхнього рівня, яка визначає порядок роздачі доменних імен у своїй країні.

Також варто відмітити **зарезервовані діапазони адрес (RFC 1918):**

- 10.0.0.0/8;
- 172.16.0.0/12;
- 192.168.0.0/16.

Дані адреси не маршрутизируются в мережу Інтернет - вони можуть використовуватися всередині організації без звернення до організації IANA, а для підключення до мережі Інтернет частіше за все використовується технологія NAT (Network Address Translation).

2. Безкласова адресація

Безкласова міждоменна маршрутизація CIDR

Використання безкласової адресації дозволило в значній мірі розв'язати проблеми, які виникли при використанні класової адресації. Сучасні маршрутизатори використовують форму IP-адресації, яка ігнорує класи. Даний маршрутизатор використовує біти маски для визначення в адресі мережевої частини та номера хоста. і на відміну від класової адресації межа поділу адреси може проходити посеред октету.

Даний підхід значно покращує масштабованість і ефективність використання IP - адрес, він забезпечує гнучкість і економічне використання адрес у виділеному діапазоні, дозволяє маршрутизаторам агрегувати або сумувати інформацію про маршрути, що зменшує розміри таблиць маршрутів, оскільки використовується лише одна адреса і маска для представлення маршрутів до багатьох підмереж (Supernetting, "надмережа"). Наприклад, при класовій адресації для мережі класу А 10.0.0.0/8, в якій розглядається 10 підмереж, і маршрутизатор повинен створити рядок в таблиці маршрутів для кожної з цих підмереж, беручи до уваги, що 16 біт адреси кожної з цих підмереж унікальні (Таблиця 3.1).

Таблиця 3.1

Мережевий номер	Перший октет	Другий октет	Третій октет	Четвертий октет
10.1.0.0/16	00001010	00000001	00000000	00000000
10.2.0.0/16	00001010	00000010	00000000	00000000
10.3.0.0/16	00001010	00000011	00000000	00000000
10.4.0.0/16	00001010	00000100	00000000	00000000

10.5.0.0/16	00001010	00000101	00000000	00000000
10.6.0.0/16	00001010	00000110	00000000	00000000
10.7.0.0/16	00001010	00000111	00000000	00000000
10.8.0.0/16	00001010	00001000	00000000	00000000
10.9.0.0/16	00001010	00001001	00000000	00000000
10.10.0.0/16	00001010	00001010	00000000	00000000

Проте всі адреси підмереж мають спільну частину (перші 14 біт однакові). При безкласовому підході маршрутизатор може сумувати маршрути до цих підмереж, використовуючи спільний 14 бітовий префікс в адресах 000010100000. Доповнивши даний префікс нулями справа, можемо представити його префікса в десятковій формі 00001010 00000000 00000000 00000000 ~ 10.0.0.0, а 14-бітова маска підмережі має вигляд: 11111111 11110000 00000000 00000000 ~ 255.240.0.0 - одна адреса і одна маска визначає безкласовий префікс, який сумує маршрути до всіх десяти підмереж: 10.0.0.0/14.

Відмічають, що supernetting і агрегування маршрутів є різними назвами одного процесу. Якщо порівнювати поняття *supernetting ma subnetting, mo supernetting* бере біти із мережевої частини маски, а *subnetting* - із частини маски, що відноситься до хоста.

Проблему (яка згадувалася вище) вичерпання адрес мережі класів А і В можна вирішити, якщо організація отримує блок неперервних адрес в мережах класу С, де можна використати *supernetting*.

Маска змінної довжини VLSM

Маска змінної довжини (*Variable-Length Subnet Mask (VLSM)*) дозволяє організації використовувати більше однієї маски підмережі всередині одного і того ж мережевого адресного простору. Реалізацію *VLSM* часто називають «підмережі на підмережі».

Розглянемо підмережі, які створені за допомогою запозичення трьох перших біт в хостовій частині адреси класу С 207.21.24.0 (таблиця 3.2)

Таблиця 3.2

Підмережа	Адреса підмережі
0	1.2.3.0/27

1	1.2.3.32/27
2	1.2.3.64/27
3	1.2.3.96/27
4	1.2.3.128/27
5	1.2.3.160/27
6	1.2.3.192/27
7	1.2.3.224/27

В кожній підмережі може міститися не більше 30 хостів. Використання для з'єднання точка-точка довільної із підмереж /27 призведе до втрати адрес - для цієї задачі найкраще краще всього підходить 30-и бітова маска. Розіб'ємо одну з підмереж 1.2.3.192/27 на вісім підмереж, використовуючи 30-и бітову маску (таблиця 3.3), а кожну із семи підмереж /27, що залишилися, можна використовувати для адресації хостів в семи локальних мережах.

Таблиця 3.3

Підмережа	Адреса підмережі
0	1.2.3.192/30
1	1.2.3.196/30
2	1.2.3.200/30
3	1.2.3.204/30
4	1.2.3.208/30
5	1.2.3.212/30
6	1.2.3.220/30
7	1.2.3.224/30

3. Основи маршрутизації

Маршрутиза́ція — процес визначення маршруту прямування пакету від однієї мережі до іншої. При наявності маршрутів до віддалених мереж маршрутизатор приймає рішення, що базується на IP-адресі отримувача пакету. Дана адреса використовується і всіма подальшими пристроями для просування пакету до отримувача.

Розрізняють два типи маршрутизації: статична маршрутизація (маршрути задаються вручну адміністратором) та динамічна маршрутизація (маршрути обчислюються автоматично за допомогою протоколів динамічної маршрутизації).

Варто зазначити, що при статичній маршрутизації будь-які зміни мережевої топології, що супроводжується додаванням і видаленням маршрутів, здійснюється вручну та вимагають участі адміністратора. Тому у великих мережах підтримка таблиць маршрутизації вручну може вимагати величезних витрат часу адміністратора. Статична маршрутизація не має можливості масштабування, яку має динамічна маршрутизація через додаткові вимоги до налаштування і постійне втручання адміністратора. Статичні маршрути не змінюються маршрутизатором. Динамічні маршрути змінюються маршрутизатором автоматично при одержанні інформації про зміну маршрутів від сусідніх маршрутизаторів. Статична маршрутизація споживає мало обчислювальних ресурсів і корисна в мережах, які не мають декількох шляхів до адресата призначення.

Одним з головних завдань маршрутизатора є знаходження “найкращого шляху” до заданого адресата. При динамічній маршрутизації маршрутизатори обмінюються маршрутною інформацією за допомогою протоколів маршрутизації.

Використання статичної маршрутизації використовується, як приклад, при реалізації налаштування постійного віддаленого доступу до ПК або сервера, а також:

- камер відеоспостереження;
- файлових серверів;
- поштових серверів;
- серверів для реалізації архівування даних.

Маршрутизатор зберігає таблиці маршрутів в оперативній пам'яті. Для перегляду таблиці маршрутів використовується команда show ip route. Навіть, якщо на деякому маршрутизаторі не задавалися жодні команди маршрутизації, він все одно будує таблицю маршрутів для безпосередньо приєднаних до нього мереж, наприклад:

. . .

C 10.31.0.0 is directly connected, Serial0

C10.42.0.0 is directly connected, Serial1

Також маршрут на безпосередньо приєднані мережі відображається на інтерфейсі маршрутизатора, до якого вони приєднані.

Для конфігурації статичної маршрутизації в маршрутизаторах Cisco використовують дві версії команди **ip route**.

Перша версія

ip route Адреса_Мережі_Призначення Маска_Мережі_Призначення Інтерфейс

Команда вказує маршрутизатору, що всі пакети, які призначені для *Адреса_Мережі_Призначення-Маска_Мережі_Призначення* потрібно направляти на свій інтерфейс *Інтерфейс*. Якщо інтерфейс *Інтерфейс* – типу *Ethernet*, то фізичні (MAC) адреси вихідних пакетів будуть широкомовними, як наприклад: **ip route 10.6.0.0 255.255.0.0 Serial1**

Друга версія

ip route Адреса_Мережі_Призначення Маска_Мережі_Призначення Адреса

Команда вказує маршрутизатору, що всі пакети, які призначені для *Адреса_Мережі_Призначення-Маска_Мережі_Призначення*, треба направити на той інтерфейс з якого буде досяжною IP-адреса *Адреса* – адреса наступного кроку на шляху до *Адреса_Мережі_Призначення*. Вихідний інтерфейс та фізична адреса вихідних пакетів визначаються маршрутизатором згідно своїх ARP таблиць на підставі IP-адреси *Адреса*, як наприклад: **ip route 10.7.0.0 255.255.0.0 10.4.0.2.**

Обидві команди додадуть статичні маршрути в таблицю маршрутизації (*мітка S*):

S10.6.0.0 via Serial1

S 10.7.0.0 [1/0] via 10.4.0.2

Якщо інтерфейс "падає", всі статичні маршрути, що відображаються на цей інтерфейс, вилучаються з таблиці маршрутів.

Варто зауважити, що для мереж типу Ethernet рекомендується завжди використовувати другу форму команди **ip route**, так як дана команда дозволить маршрутизатору правильно сформувати фізичну адресу вихідного пакету за своїми ARP таблицями.

ARP (Address Resolution Protocol)

Після визначення IP адреси одержувача відправник здійснює пошук у своїй ARP таблиці його MAC адресу. Якщо MAC і IP адреси одержувача присутні в ARP таблиці відправника, між ними встановлюється відповідність, яка використовується пізніше

при формуванні фрейму на канальному рівні. Після того як MAC адреса береться з ARP таблиці, фрейм через фізичний канал відправляється від відправника до адресата.

Якщо MAC адреса одержувача з IP-адресою АДР відсутня в ARP таблиці, тоді відправник здійснює в мережу широкомовний ARP запит, який приймають всі мережеві пристрої в сегменті мережі, і лише пристрій, що має шукану IP-адресу *АДР*, реагує на нього, посилаючи відправникові інформацію про свою MAC-адресу. Відправник записує пару *<MAC адреса, IP-адреса АДР >* у свою ARP таблицю.

Маршрутизація за замовчуванням

Зовсім не обов'язково, щоб кожен маршрутизатор обслуговував маршрути до всіх можливих мереж призначення. Дуже часто маршрутизатор зберігає маршрут за замовчуванням, який використовуються, коли маршрутизатор не може поставити у відповідність мережі призначення рядок в таблиці маршрутів.

Маршрут за замовчуванням може бути статично введений адміністратором або динамічно отриманий з протоколу маршрутизації. Оскільки всі IP адреси належать мережі 0.0.0.0 з маскою 0.0.0.0 (0.0.0.0/0), то в цьому випадку потрібно використовувати команду

ip route 0.0.0.0 0.0.0.0 [адреса наступного кроку | вихідний інтерфейс]

Ручне задання маршруту за замовчуванням на кожному маршрутизаторі зручне для простих мереж, а у складних мережах рекомендовано організовувати динамічний обмін маршрутами за замовчуванням.

Інтерфейс петля

При поетапному проектуванні мереж на мережевих пристроях можна створювати мережеві інтерфейси, які не пов'язані з реальними каналами для передачі даних і призначати на них IP адреси з масками - інтерфейси-петлі (*loopback*). Якщо до якогось реального мережевого інтерфейсу маршрутизатора надалі буде приєднана підмережа, то на початку на маршрутизаторі створюється *loopback*, яка налаштовується з точки зору взаємодії з останніми ділянками мережі і лише потім замінюється на реальний інтерфейс. Для створення інтерфейсу петлі використовується команда **interface loopbackN** або скорочено **int lN**, де *N* ціле невід'ємне число - номер петлі. як наприклад: **int l0 1.1.1.1 255.0.0.0.**

Команда trace

Подібно до команди **ping,** команда **trace** ту ж технологію протоколу *ICMP* та є інструментом для з'ясування маршруту даних в мережі. Дана команда замість перевірки наскрізного зв'язку між відправником і одержувачем перевіряє кожен крок на шляху та використовує здатність маршрутизаторів генерувати повідомлення про помилку при перевищенні пакетом свого встановленого часу життя (*Time To Live, TTL*). Ця команда надсилає декілька пакетів і виводить на екран дані про час проходження туди і назад для кожного з них. Команда **trace** надає розгорнуту інформацію, показуючи черговий досягнутий маршрутизатор на шляху до пункту призначення. Це дуже потужний засіб для локалізації відмов на шляху від відправника до одержувача. Варіанти відповідей утиліти **trace (**таблиця 3.4):

Таблиця 3.4

Символ	**Значення**
!H	Зондуючий пакет був прийнятий маршрутизатором, але не переадресований
P	Протокол недосяжний
N	Мережа недосяжна
U	Порт недосяжний
*****	Перевищення границі очікування

4. Динамічна маршрутизація

Статична маршрутизація не підходить для великих, складних мереж, тому що зазвичай мережі включають надлишкові зв'язки, багато протоколів і змішані топології. Маршрутизатори в складних мережах повинні швидко адаптуватися до змін топології і вибирати кращий маршрут з багатьох кандидатів - в такому випадку використовується динамічна маршрутизація і маршрути обчислюються автоматично за допомогою протоколів динамічної маршрутизації. Протоколи маршрутизації – це правила, за якими здійснюється обмін інформації про шляхи передачі пакетів між маршрутизаторами, вони характеризуються часом збіжності, втратами і масштабованістю.

Для динамічної маршрутизації характерно:

- мінімальний ризик виникнення помилки;
- система чітко розуміє, що кому відправляти;
- масштабування мережі відбувається швидше;
- видалення людського фактора з ланцюжка.

Крім автоматизації та легкого масштабування, принцип динамічної маршрутизації дозволяє роутерам та іншому обладнанню будувати шлях самостійно, особливо це корисно, коли маршрут через певні причини стає неможливим для використання - тому зростає стійкість до відмов мережі, оскільки не потрібно думати над додатковими каналами. Також при динамічній маршрутизації підтримується балансування трафіку, оскільки всі сучасні роутери підтримують подібну функцію.

Динамічна маршрутизація здійснюється за допомогою спеціальних протоколів. IP мережі мають ієрархічну структуру і з точки зору маршрутизації мережа розглядається як сукупність автономних систем, для роботи всередині яких використовуються внутрішні протоколи шлюзів (interior gateway protocols (IGP)), а для роботи між автономними системами – зовнішні протоколи шлюзів (exterior gateway protocols (EGP)). До протоколів IGP відносяться RIP, RIP v2, IGRP, EIGRP, OSPF, а протоколи EGP3 і BGP4 відносяться до EGP. Ці протоколи можуть бути розділені на класи: дистанційно-векторні протоколи та протоколи стану зв'язку.

Для оцінки маршрутів маршрутизатори використовують метрики. Коли від маршрутизатора до мережі призначення існує багато маршрутів, що використовують один протокол маршрутизації, то маршрут з найменшою метрикою розглядається як найкращий. Наприклад, протокол RIP використовує в якості метрики кількість переходів (хопів), а EIGRP – складну комбінацію чинників, що включає смугу пропускання каналу і його надійність. Таблиця маршрутів, змінюючись при зміні ситуації в мережі, оновлюється результатами роботи маршрутизуючих протоколів.

Вигляд рядка в таблиці маршрутів, що відноситься до динамічної маршрутизації:

R 10.16.5.0/24 [151/2] via 10.1.2.1 12:22:16 Serial1 .

Перша літера вказує на протокол маршрутизації (R - RIP, а O – OSPF), запис [122/3] означає, що маршрут має адміністративну відстань 151 і метрику 2. Ці числа маршрутизатор використовує для вибору маршруту. Елемент 12:22:16 визначає час, коли оновився даний рядок, Serial1 – це локальний інтерфейс, через який маршрутизатор буде направляти пакети до мережі 10.16.5.0/24 через адресу 10.1.2.1.

Дистанційно-векторна маршрутизація базується на основі алгоритму Беллмана-Форда, яка полягає в тому, що через певні моменти часу маршрутизатор передає сусіднім маршрутизаторам всю свою таблицю маршрутизації. Прикладами таких протоколів є RIP і IGRP, які поширюють інформацію про таблиці маршрутів через всі інтерфейси маршрутизатора в широкомовному режимі, а сусідні маршрутизатори, отримуючи повідомлення, порівнює інформацію зі своєю поточною таблицею маршрутів і доповнює її маршрутами до нових мереж або маршрутами до відомих мереж з кращою метрикою, а також вилучає неіснуючі маршрути. Також маршрутизатор додає свої власні значення до метрик отриманих маршрутів і нову таблицю маршрутизації знову поширює по сусідніх маршрутизаторах.

Протоколи стану зв'язку забезпечують кращу масштабованість і збіжність в порівнянні з дистанційно-векторними протоколами, так як вони базуються на алгоритмі Дейкстри (shortest path first (SPF)), найбільш типовим представником яких є протокол OSPF (Open Shortest Path First). Дані протоколи стану мають більш швидку збіжність і краще використання смуги пропускання в порівнянні з дистанційно-векторними протоколами, проте мають і недоліки: підвищені вимоги до обчислювальної потужності маршрутизаторів та складне адміністрування.

Під час роботи динамічної маршрутизації важливо, щоб індивідуальні таблиці маршрутизації були точними та всі маршрутизатори мали однакову інформацію про топологію мережі. Кажуть, що у випадку домовленості маршрутизаторами щодо топології мережі, має місце їх збіжність. Швидка збіжність означає швидке відновлення після обриву зв'язків та інших змін у мережі.

Також варто зазначити, що коли маршрутизатори перебувають у процесі збіжності, мережа сприйнятлива до проблем маршрутизації, які виражаються у відкиданні пакетів та появі петлі маршрутизації. В залежності від багатьох чинників та від протоколу може пройти багато часу поки всі процеси маршрутизації в мережі зійдуться.

Конфігурування динамічної маршрутизації

Для конфігурування динамічної маршрутизації використовуються дві основні команди: **router** та **network**.

Команда **router** запускає процес маршрутизації і має форму:

Router(config)# **router protocol [keyword]**

де protocol - будь-який з протоколів маршрутизації: RIP, IGRP, OSPF тощо, keyword - додаткові параметри.

Потім необхідні команди **network**:

Router(config-router)# **network network-number [keyword]**

де *network-number* – ідентифікує безпосередньо підключену мережу, яка додається в процес маршрутизації, *keyword* – додаткові параметри; *network-number* дозволяє процесу маршрутизації визначити інтерфейси, які будуть брати участь у відсилання та прийомі пакетів, актуалізації маршрутної інформації.

Для перегляду інформації про протоколи маршрутизації використовується команда **show ip protocol**, яка виводить значення таймерів процесів маршрутизації і мережеву інформацію, що має відношення до маршрутизації. Вміст таблиці IP маршрутизації виводиться командою **show ip route**. Вона містить записи про всі відомі маршрутизатору мережі і підмережі та вказує на спосіб отримання цієї інформації.

Протокол RIP

Протокол RIP відноситься до протоколів дистанційно-векторної маршрутизації, для якого характерно:

- метрика при виборі шляху у вигляді кількості переходів ;
- максимальна допустима кількість переходів - 15;
- за замовчуванням пакети актуалізації маршрутної інформації надсилаються у режимі широкомовлення кожні 30 секунд.

Вибір протоколу маршрутизації RIP здійснюється командою **router rip,** а команда network призначає IP адресу мережі з якою маршрутизатор має безпосереднє з'єднання **network network-number**. Процес маршрутизації зв'язує інтерфейси з відповідними адресами і починає обробку пакетів у заданих мережах.

Для виведення змісту пакетів актуалізації маршрутної інформації використовується команда **debug ip rip**.

Протокол IGRP

Компанія Cisco розробила протокол маршрутизації за вектором відстані IGRP, який посилає пакети актуалізації маршрутної інформації з 90-секундним інтервалом, в яких містяться відомості про мережі для конкретної автономної системи. Цей протокол характеризує універсальність, що дозволяє автоматично справлятися зі складними топологіями. Метрика не має властивих протоколу RIP обмежень за кількістю переходів та включає наступні складові:

- *Ширина смуги пропускання;*
- *Величина затримки;*
- *Рівень завантаження;*
- *Надійність каналу;*
- *Розмір максимального блоку передачі в каналі.*

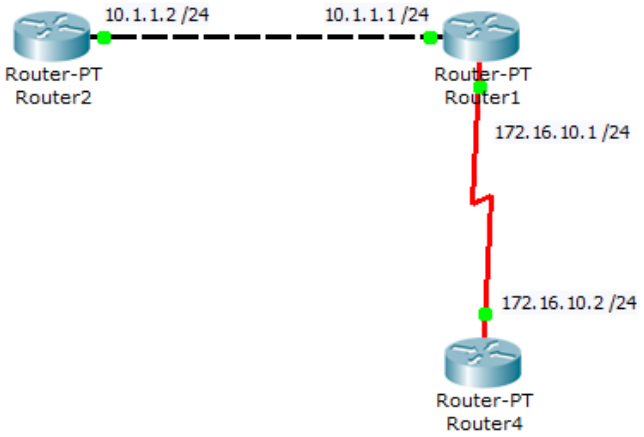
Вибір протоколу маршрутизації IGRP здійснюється за допомогою команди **router igrp autonomous-system,** де параметр *autonomous-system* – це номер автономної системи, який ідентифікує обчислювальний процес IGRP-маршрутизації. Процеси в маршрутизаторах мережі з однаковим номером *autonomous-system* будуть колективно використовувати маршрутну інформацію. Команда **network** задає безпосередньо приєднані мережі, що підлягають включенню в даний процес маршрутизації, як наприклад **network network-number.**

Вивід змісту пакетів актуалізації маршрутної інформації протоколу IGRP реалізується командами **debug ip igrp transactions** та **debug ip igrp events**.

Хід роботи

Практична робота 1. Налаштування маршрутизації за допомогою маршрутизатора Cisco

1. Створимо у Packet Tracer топологію, яка зображена на мал.2.16 з лабораторної роботи № 2.



2. Перейдемо до маршрутизатора *Router1* і переглянемо його ARP-таблицю

Router1# **show arp**

Дана таблиця містить тільки рядок про MAC-адресу свого Ethernet інтерфейсу з IP адресою 10.1.1.1, як наприклад:

Protokol	**Address**	**Age (min)**	**Hardware Addr**	**Type**	**Interface**
Internet	*10.1.1.1*	*-*	*00D0.58B7.92B3*	*ARPA*	*FastEthernet0/0*

3. Приєднаймося до маршрутизатора *Router2* і переглянемо його ARP-таблицю. Вона містить тільки один рядок про MAC-адресу свого Ethernet інтерфейсу з IP адресою 10.1.1.2.

Protokol	**Address**	**Age (min)**	**Hardware Addr**	**Type**	**Interface**
Internet	*10.1.1.2*	*-*	*00D0.7896.64B1*	*ARPA*	*FastEthernet0/0*

4. “Пропінгуємо” Ethernet інтерфейс маршрутизатора *Router1*

Router2# **ping 10.1.1.1**

5. Знову переглянемо ARP-таблицю. Вона містить вже два рядки. З'явився запис про MAC адресу Ethernet інтерфейсу *Router1* з IP адресою 10.1.1.1

Protokol	**Address**	**Age (min)**	**Hardware Addr**	**Type**	**Interface**
Internet	*10.1.1.1*	*0*	*00D0.58B7.92B3*	*ARPA*	*FastEthernet0/0*
Internet	*10.1.1.2*	*-*	*00D0.7896.64B1*	*ARPA*	*FastEthernet0/0*

6. Приєднаймося до маршрутизатора *Router1* і переглянемо його ARP-таблицю.

Protokol	**Address**	**Age (min)**	**Hardware Addr**	**Type**	**Interface**
Internet	*10.1.1.1*	*0*	*00D0.58B7.92B3*	*ARPA*	*FastEthernet0/0*
Internet	*10.1.1.2*	*2*	*00D0.7896.64B1*	*ARPA*	*FastEthernet0/0*

З'явився запис про MAC-адресу Ethernet інтерфейсу маршрутизатора *Router2* з IP адресою 10.1.1.2.

7. У лабораторній роботі №2 не було можливості з маршрутизаторів *Router2* і *Router4* пропінгувати деякі інтерфейси через відсутність маршрутизації (не могли пінгувати адреси 172.16.10.1 і 172.16.10.2). Для виправлення даної ситуації приєднаємося до маршрутизатора *Router2* та переглянемо таблицю маршрутизації

Router2# **show ip route**

За результатами роботи даної команди можна побачити тільки безпосередньо приєднані мережі, а до мережі 172.16.10.0/24 маршрут відсутній - тому додамо його через адресу 10.1.1.1, що є найближчим кроком на шляху до мереж призначення:

Router2(config)# **ip route 172.16.10.0 255.255.255.0 10.1.1.1**

8. Переглянемо таблицю маршрутизації та перевіримо зв'язок з послідовним інтерфейсом *Router1,* виконавши наступні команди

Router2# **show ip route**

Router2# **ping 172.16.10.1**

9. Перевіримо зв'язок з послідовним інтерфейсом *Router4*

Router2# **ping 172.16.10.2**

Перевірити з'єднання між *Router2* та *Router4* не вдається, так як на *Router4* не прописані необхідні маршрути для повернення пакетів назад.

10. Приєднаймося до маршрутизатора *Router4* та переглянемо таблицю маршрутизації

Router4# **show ip route**

```
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial2/0
```

Відсутній маршрут до мережі 10.1.1.0/24. Додамо цей маршрут через адресу 172.16.10.1 (найближчого кроку на шляху до мережі 10.1.1.0/24):

Router4(config)# **ip route 10.1.1.0 255.255.255.0 172.16.10.1**

Знову переглянемо таблицю маршрутизації

```
     10.0.0.0/24 is subnetted, 1 subnets
S       10.1.1.0 [1/0] via 172.16.10.1
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial2/0
```

11. Перевіримо, що тепер всі мережеві інтерфейси в мережі “пінгуються” з кожного мережевого пристрою.

12. В зв'язку з тим, що мережеві пристрої *Router2* і *Router4* мають тільки по одному виходу у зовнішній світ через інтерфейси з адресами 10.1.1.1 і 172.16.10.1 відповідно, можна не визначати на які підмережі маршрутизуються пакети і використовувати маршрутизацію за замовчуванням.

13. Для цього спочатку видалимо старі маршрути.

Router2(config)# **no ip route 172.16.10.0 255.255.255.0 10.1.1.1**

Router4(config)# **no ip route 10.1.1.0 255.255.255.0 172.16.10.1**

14. Та призначимо маршрути за замовчуванням.

Router2(config)# **ip route** 0.0.0.0 0.0.0.0 10.1.1.1

Router4(config)# **ip route** 0.0.0.0 0.0.0.0 172.16.10.1

15. Переглянемо таблицю маршрутизації на всіх пристроях.

Router2(config)# **sh ip route**
Router4(config)# **sh ip route**

16. Перевіримо, що всі мережеві інтерфейси в мережі "пінгуються" з кожного мережевого пристрою.

17. Визначимо інтерфейс петля (loopback interface) на пристрої *Router4*

Router4(config)# **int loopback 0 1.1.1.1 255.255.255.0**

18. Пропишемо на пристрої *Router1* маршрут на мережу петлі

Router1(config)# **ip route 1.1.1.0 255.255.255.0 172.16.10.2**

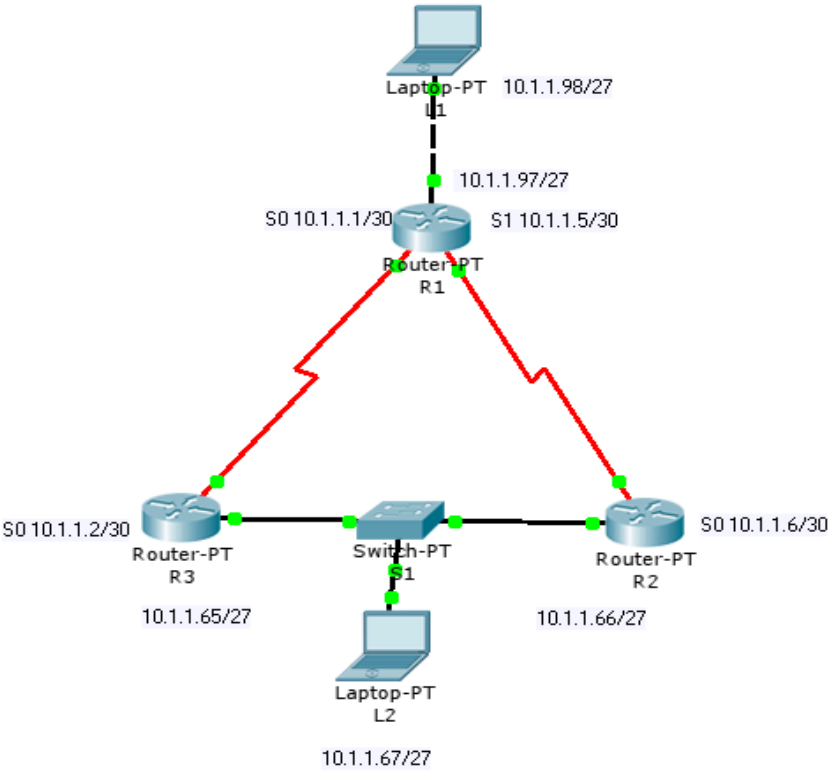
19. Приєднаємося до пристрою *Router*2 і пропінгуємо створену петлю

Router2# **ping 1.1.1.1**

Збережіть проект у цілому й конфігурацію кожного маршрутизатора в окремий файл.

Практична робота 2. Проектування та організація мережі з використанням масок змінної довжини

1. Створимо у Packet Tracer топологію, яка зображена на мал.3.4



Мал. 3.4

2. Задаємо IP-адреси інтерфейсам на абонентах та маршрутизаторах відповідно до малюнку 9.
3. У звіті вказати як виглядає маска підмережі /27 та маска підмережі /30 у десятковому вигляді.
4. При захисті роботи аргументувати чому вибрані саме такі маски.
5. Забезпечуємо маршрутизацію на кожному маршрутизаторі для організації доступу до будь-якої мережі.
6. Забезпечуємо маршрутизацію на кожному абоненті з використанням шляху за замовченням.
7. Перевіряємо з’єднання попарно між усіма абонентами.

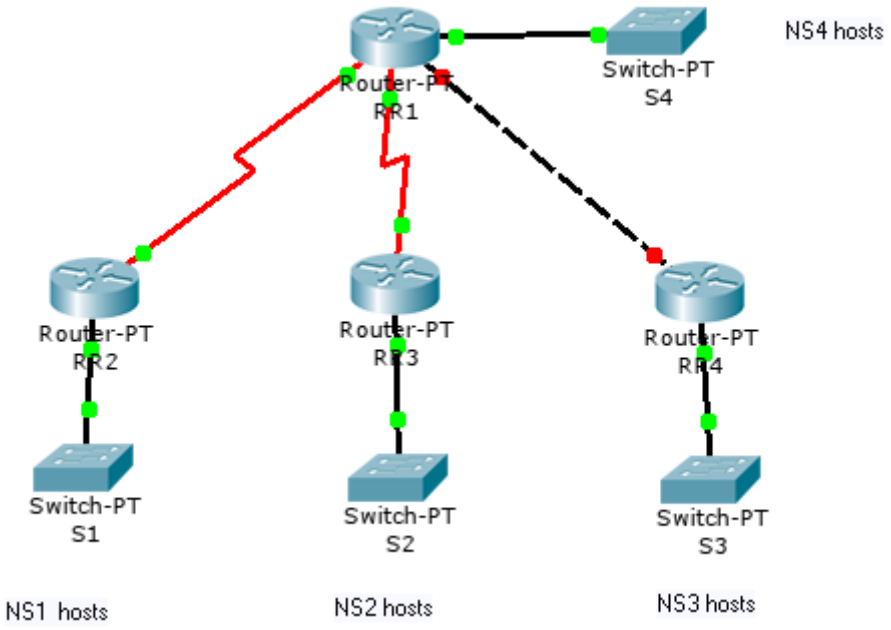
Завдання для самостійної роботи

1. Відповідно до номеру в підгрупі вибрати варіант.

№ варіанту / адреса мережі	NS1	NS2	NS3	NS4
1 - 10.1.0.0	20	40	70	90

2 - 10.2.0.0	15	25	50	60
3 - 10.3.0.0	10	35	55	70
4 - 10.4.0.0	5	12	21	100
5 - 10.5.0.0	18	25	38	45
6 - 10.6.0.0	42	50	72	120
7 - 10.7.0.0	29	20	8	6
8 - 10.8.0.0	33	43	63	73
9 - 10.9.0.0	110	60	25	17
10 - 10.10.0.0	80	70	60	50
11 - 10.11.0.0	70	50	30	10
12 - 10.12.0.0	120	90	60	28
13 - 10.13.0.0	85	55	25	5
14 - 10.14.0.0	42	62	78	4
15 - 10.15.0.0	10	25	40	80
16 - 10.16.0.0	8	31	54	108
17 - 10.17.0.0	3	7	15	31
18 - 10.18.0.0	20	25	55	63
19 - 10.19.0.0	15	18	7	5
20 - 10.20.0.0	100	80	60	50

2. Спроектуйте наступні чотири мережі, згідно отриманого варіанту та мал. 3.5.



Мал. 3.5

3. У дизайнері правильно підберіть маршрутизатори з потрібним числом і типом інтерфейсів. Використайте при потребі багатослотові пристрої, що мають нарощене число інтерфейсів. В симуляторі вони мають адреси виду Ethernet 2/0, що означає інтерфейс 0 в слоті 2.

4. Оптимально виберіть адресу та маску змінної довжини для кожної мережі з мінімальною кількістю "дірок". Аргументацію вибору заданих адрес зазначити в звіті.

5. Нанесіть в графічному редакторі на вашу топологію назначені адреси і маски.

7. В симуляторі назначьте адреси на мережеві інтерфейси маршрутизаторів і комп'ютерів.

8. Перевірте правильність назначення адрес командами **sh ip int br** (маршрутизатор) або **ipconfig** (комп'ютер).

9. Налаштуйте на кожному маршрутизаторі динамічну маршрутизацію за протоколом RIP другої версії.

10. На кожному маршрутизаторі виведіть таблицю маршрутизації і зробіть скріншот.

11. Перевірте наявність зв'язку між довільними комп'ютерами.

12. По за побажанням можна в маршрутизаторах використовувати інтерфейси петля loopback для моделювання локальних мереж з наступною їх заміною на Ethernet інтерфейс.

Контрольні запитання

1. Що таке маршрутизатор ?
2. Що таке маршрутизація ?
3. На якому рівні моделі OSI реалізовано маршрутизацію?
4. Що таке ip-адреса? Які види ip-адрес бувають ?
5. Що таке маска підмережі ?
6. Що таке маршрут за замовчуванням ?
7. Яким чином створюється інтерфейс-петля ?
8. Навести алгоритм визначення маршруту до пункту призначення.
9. Чим статична маршрутизація відрізняється від динамічної маршрутизації?

Порядок виконання і здачі роботи

1. Вивчити теоретичну частину.
2. Виконати практичну роботу 1, практичну роботу 2 та завдання для самостійної роботи. Результати кожного кроку мають бути підкріплені скріншотом.
3. Відповісти на контрольні запитання.
4. Оформити звіт.
5. Продемонструвати набуті вміння та навички при захисті виконаної роботи.

Зміст звіту

Звіт готується в електронному вигляді та вивантажується в систему електронного навчання moodle. Звіт має містити результати виконання теоретичного та самостійного завдання з коментарями та скріншотами до кожного етапу.

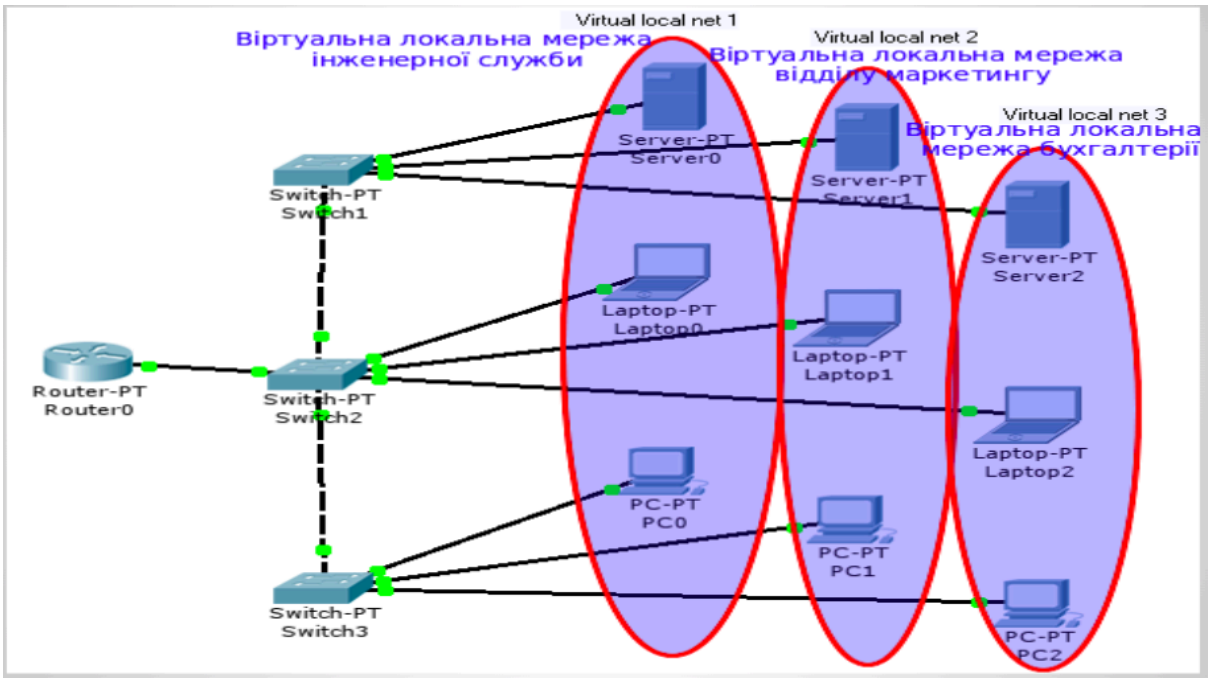
Лабораторна робота № 4. Віртуальні локальні мережі VLAN

Мета роботи: Засвоїти розбиття локальної мережі на декілька віртуальних локальних мереж з використанням пристроїв компанії Cisco.

Методичні вказівки

Головною перешкодою для створення великих локальних мереж за допомогою тільки комутаторів є нелінійний ріст обсягу широкомовного трафіку, який виникає з ростом кількості пристроїв у мережі. Локальна мережа, яка створена за допомогою тільки комутаторів, представляє один широкомовний домен. Зменшити такий домен можна фізично розділивши локальну мережу на незалежні підмережі (незалежні групи попарно пов'язаних комутаторів), з'єднавши їх в єдине ціле з використанням маршрутизаторів. Таке завдання можна вирішити тільки на етапі побудови мережі, а не при її експлуатації.

Virtual Local Area Network — віртуальна локальна комп'ютерна мережа, що складається з групи хостів трафік від яких на канальному рівні повністю ізольований від трафіку інших груп. VLAN дозволяють хостам групуватися або дистанціюватися між собою. Пристрої в межах однієї VLAN можуть “спілкуватися”, а вузли, що знаходяться в різних VLAN, невидимі один для одного.



Приклад організації віртуальної локальної мережі

Переваги використання VLAN випливають із можливості ізоляції мереж:

• Підвищення безпеки та захисту мережі.

• Розподіл навантаження.

• Обмеження широкомовного трафіка та збільшення швидкості мережі.

Варто зазначити, що VLAN поводяться так само, як і фізично розділені локальні мережі: після розбиття мережі на VLAN одержується декілька локальних мереж, які далі необхідно об'єднати в єдине ціле за допомогою маршрутизації на третьому мережному рівні.

Серед додаткових переваг концепції VLAN виділяють наступні:

- формування локальних мереж не за місцем розташування найближчого комутатора, а за належністю комп'ютерів до вирішення тієї чи іншої виробничої задачі;
- створення мережі за типом використовуваного обчислювального ресурсу та необхідної серверної послуги (файл-сервер, сервер баз даних);
- *VLAN* дозволяють вести різну політику безпеки для різних віртуальних мереж;
- переміщати комп'ютер з однієї мережі в іншу без здійснення фізичного переміщення або перепідключення.

Для налаштування *VLAN,* варто пам'ятати, що порт комутатора працює або в режимі доступу або в магістральному режимі.

Для обміну інформацією про VLAN комутатори використовують магістральний (транковий) протокол - тому між комутаторами потрібно створити магістральні порти.

Магістральний порт - це порт, який використовується для передачі інформації про VLAN до інших мережевих пристроїв, які приєднані до цього порту:

- магістральні порти не належать певній *VLAN*;
- магістральні порти використовуються для приєднання до інших комутаторів, маршрутизаторів або серверів, що мають мережеві адаптери з можливістю для підключення до багатьох *VLAN*;
- магістралі можуть розширити *VLAN* по всій мережі, для магістральних цілей призначають високошвидкісні порти комутаторів: *Gigabit Ethernet* та *10Gigabit*;
- для мультиплексування трафіку *VLAN* існують спеціальні протоколи, які дозволяють портам визначити, якому *VLAN* належить пакет: для зв'язку між пристроями *Cisco* використовується протокол *Inter-Switch Link* (*ISL*), а при наявності в мережі обладнання декількох виробників застосовується протокол *IEEE 802.1Q*;
- без магістральних зв'язків для підтримки *VLAN* необхідно організувати зв'язок доступу для кожної *VLAN* - тому магістральні зв'язки абсолютно необхідні при проектуванні локальних мереж.

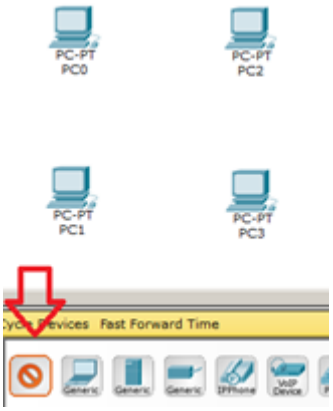
Також потрібно активізувати магістральний протокол на потрібних портах, так як він вимкнений за замовчуванням.

У режимі доступу порт належить тільки одній VLAN, а порт доступу приєднується до кінцевого пристрою: ПК, робочої станції, сервера, хабу, тощо. Фрейми, що проходять через порт доступу, є звичайними Ethernet-фреймами.

Хід роботи

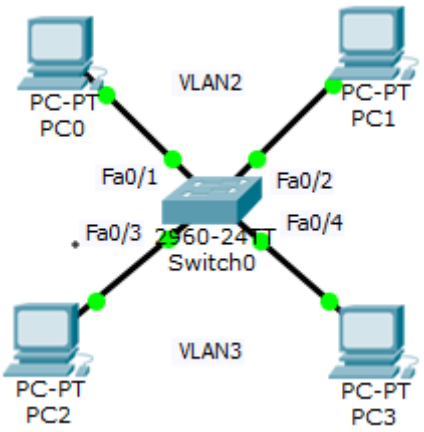
Практична робота 1. Налаштування VLAN з одним комутатором

Для додавання ПК оберіть в кінцевих пристроях настільний комп'ютер і, утримуючи **Ctrl** (так швидше), 1 раз натисніть на ПК а потім додайте потрібну кількість ПК, клацаючи мишкою (мал. 4.1). Цим прийомом ви зможете за один раз додати відразу 4 ПК.



Мал. 4.1. Додавання однотипних пристроїв

Встановлюємо комутатор і, утримуючи **Ctrl**, створюємо підключення прямим кабелем, обираючи порти комутатора. Після ініціалізації портів усі індикатори з'єднання стануть зеленими. На схемі буде дві підмережі (мал. 4.2).



Мал. 4.2. Дві підмережі: VLAN2 та VLAN3

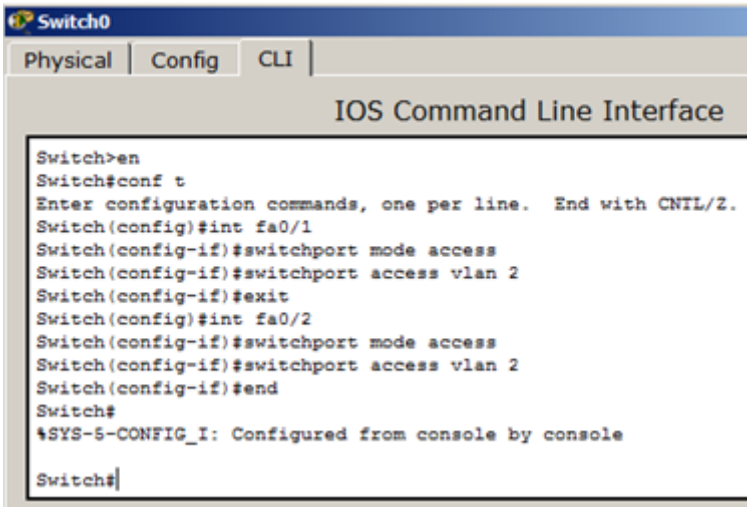
Примітка: *Ім'я VLAN1 використовується за замовчуванням, його не можна використовувати в нашому прикладі.*

На комутаторі набираємо команду **en** і входимо в привілейований режим. Потім набираємо команду **conf t** для входу в режим глобального конфігурування. Якщо підвести курсор миші до портів комутатора, ви побачите які порти в якому сегменті задіяні. Для VLAN3 – це Fa0/3 та Fa0/4 і для VLAN2 – це Fa0/1 та Fa0/2. Спочатку конфігуруємо другий сегмент мережі VLAN2 – мал. 4.3.

```
Switch#
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 2
Switch(config-vlan)#name sklad
```

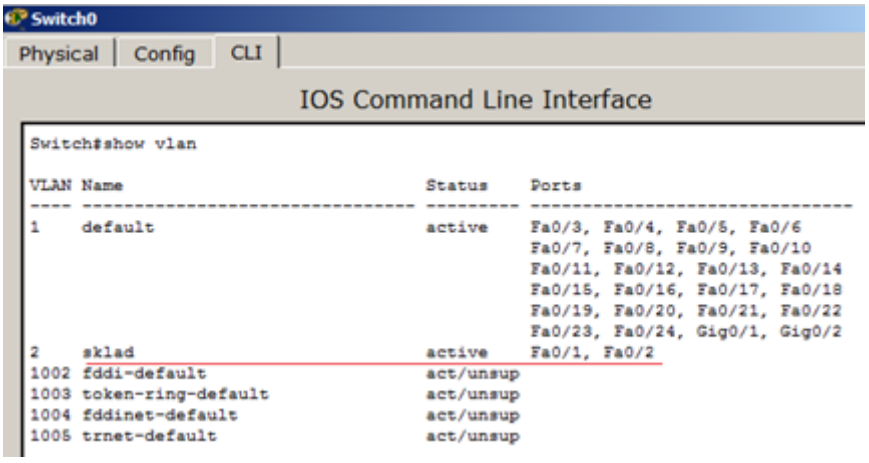
Мал. 4.3.

У віртуальній мережі VLAN2 налаштовуємо порти комутатора Fa0/1 і Fa0/2 як **access** порти, тобто порти для підключення користувачів (мал. 4.4).



Мал. 4.4

Тепер командою **show vlan** можна перевірити результат (мал. 4.5).



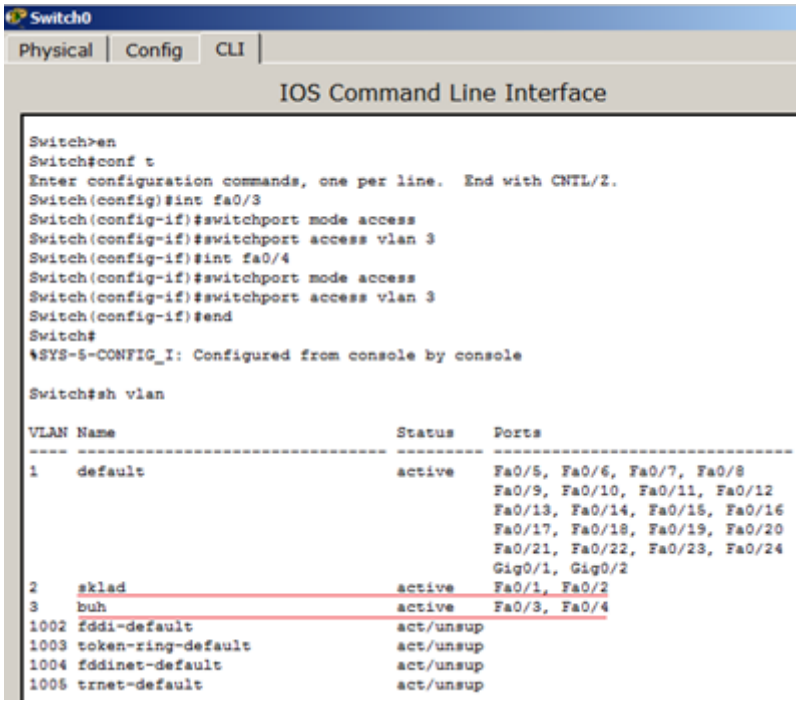
Мал. 4.5.

Далі працюємо з VLAN3 (мал. 4.6).

```
Switch#
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config-vlan)#vlan 3
Switch(config-vlan)#name buh
Switch(config-vlan)#exit
Switch(config)#
```

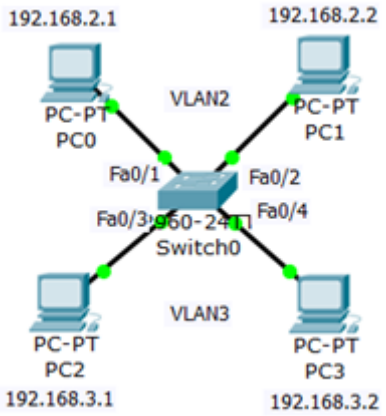
Мал. 4.6.

У віртуальній мережі VLAN3 налаштовуємо порти комутатора Fa0/3 і Fa0/4 як **access** порти, тобто порти для підключення користувачів, після цього командою **show vlan** можна перевірити та переконатися, що ми створили в мережі 2 сегменти на різні порти комутатора (мал. 4.7).



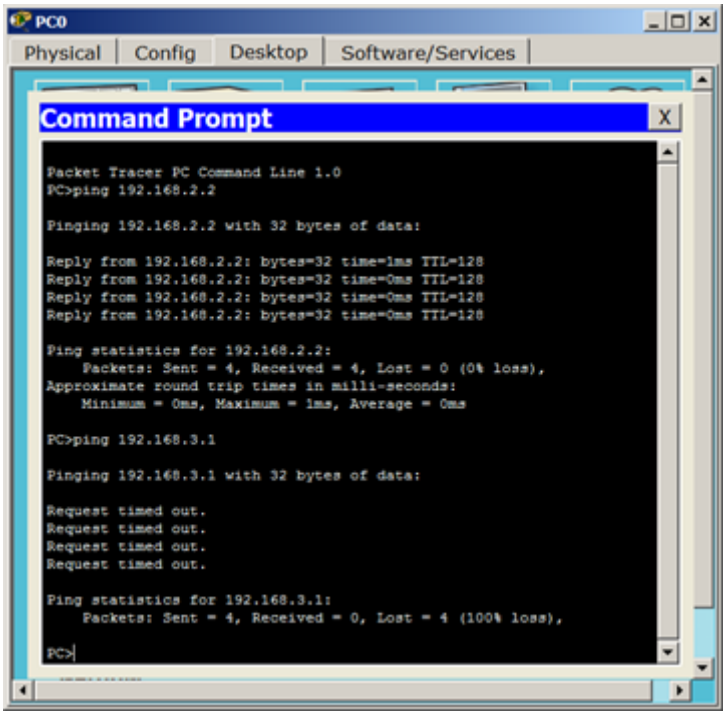
Мал. 4.7. Налаштування VLAN2 та VLAN3

Налаштовуємо IP адреси комп'ютерів – для VLAN2 з мережі 192.168.2.0, а для VLAN3 з мережі 192.168.3.0 (мал. 4.8).



Мал. 4.8. Налаштування IP адреси комп'ютерів

Перевіряємо зв'язок ПК у межах VLAN та відсутність зв'язку між VLAN2 та VLAN3 (мал. 4.9).

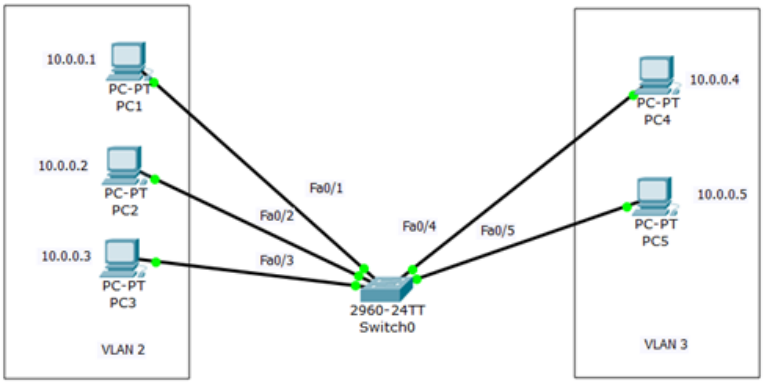


Мал. 4.9. Перевірка з’єднання

Отже, на прикладі комп'ютера PC0 можна переконатися, що комп'ютер бачить ПК тільки у своєму сегменті.

Практична робота 2. Налаштування віртуальної мережі на комутаторі

У даній роботі розглядається налаштування VLAN на комутаторі фірми Cisco у програмі **Cisco Packet Tracer**. Створюємо мережу, топологія якої представлена на мал. 4.10.



Мал. 4.10. Схема мережі з одним комутатором

Завданням даної роботи є створення 2-х незалежних груп комп'ютерів: PC1-PC3, які мають бути доступні лише один одному, та групи з двох комп'ютерів PC4 і PC5.

Налаштування комутатора

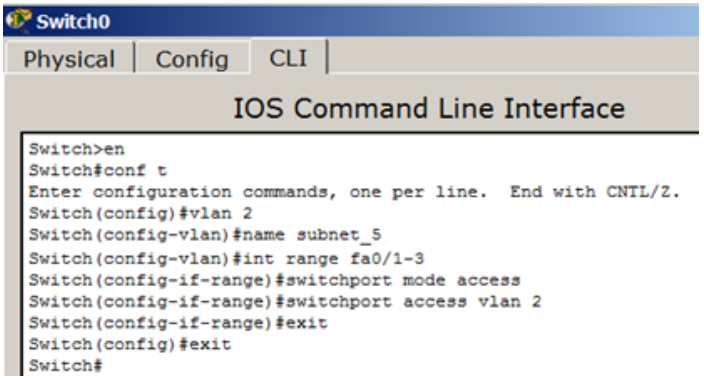
Спочатку сформуємо VLAN2. Двічі клацнути лівою кнопкою миші по комутатору. У вікні, перейшовши на вкладку **CLI**, побачимо вікно консолі. Натиснути клавішу **Enter** для того, щоб розпочати введення команд. Перейти у привілейований режим, виконавши команду **enable**:

Switch>en.

За замовчуванням, усі ПК об'єднані у VLAN1. Для реалізації мережі, яку ми запланували, створимо на комутаторі ще VLAN2 та VLAN3. Для цього в привілейованому режимі потрібно виконати наступну команду для переходу в режим конфігурації:

Switch#conf t.

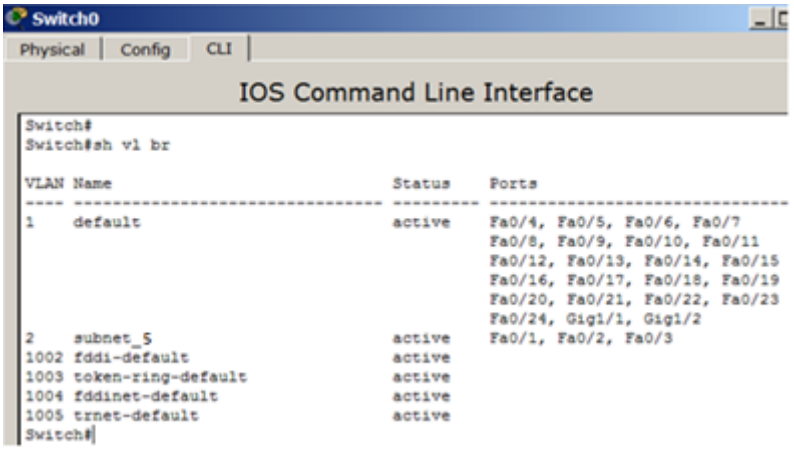
Після цього вводимо команду **VLAN 2**. Даною командою створюється на комутаторі VLAN з номером 2. Покажчик введення Switch(config)# зміниться на Switch(config-vlan)#, що свідчить про те, що зконфігуровано ще не весь комутатор в цілому, а тільки окремий VLAN, в даному випадку VLAN номер 2 (мал. 4.11).



Мал**. 4.11.** Лістинг команд для формування VLAN2

Командою VLAN2 створюємо на комутаторі новий VLAN з номером 2. Команда **name subnet_5** надає ім'я subnet_5 віртуальної мережі номер 2. Виконуючи команду **interface range fast Ethernet 0/1-3** ми переходимо до конфігурування інтерфейсів FastEthernet 0/1, FastEthernet 0/2 та FastEthernet 0/3 комутатора. Слово range у цій команді, вказує на те, що конфігуруватимемо не один порт, а діапазон портів. Команда **switch port mode access** конфігурує вибраний порт комутатора, як порт доступу (access порт). Команда **switch port access vlan 2** вказує, що цей порт є портом доступу для VLAN номер 2.

Вийти з режиму конфігурування, двічі набравши команду **exit** і переглянути результат конфігурування (мал. 4.12), виконавши команду **sh vl br**. Як бачимо, на комутаторі з'явився VLAN з номером 2 та ім'ям subnet_5, портами доступу якого є fastEthernet 0/1, fastEthernet 0/2 та fastEthernet 0/3.



Мал. 4.12. Перегляд інформації про VLAN на комутаторі

Команда **shvlbr** виводить інформацію про існуючі на комутаторі VLAN. В результаті виконання команди на екрані з'явиться: **номери VLAN** (перший стовпець), **назва VLAN** (другий стовпець), **стан VLAN** (працює він чи ні) – третій стовпець, **порти**, які належать до цього VLAN (четвертий стовпець).

Далі аналогічно створюється **VLAN 3** з ім'ям **subnet_6** та його порти доступу інтерфейси fastEthernet 0/4 і fastEthernet 0/5. Результат показано на мал. 4.13.

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 3
Switch(config-vlan)#name subnet_6
Switch(config-vlan)#int range fa0/4-5
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 3
Switch(config-if-range)#exit
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#sh vl br
VLAN Name                              Status    Ports
---- --------------------------------- --------
------------------------------
1    default                           active    Fa0/6, Fa0/7, Fa0/8,
Fa0/9
                                                 Fa0/10, Fa0/11, Fa0/12,
Fa0/13
                                                 Fa0/14, Fa0/15, Fa0/16,
Fa0/17
                                                 Fa0/18, Fa0/19, Fa0/20,
Fa0/21
                                                 Fa0/22, Fa0/23, Fa0/24,
Gig0/1
                                                 Gig0/2
2    subnet_5                          active    Fa0/1, Fa0/2, Fa0/3
3    subnet_6                          active    Fa0/4, Fa0/5
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
Switch#
```

Мал. 4.13.

Перевірка результатів роботи

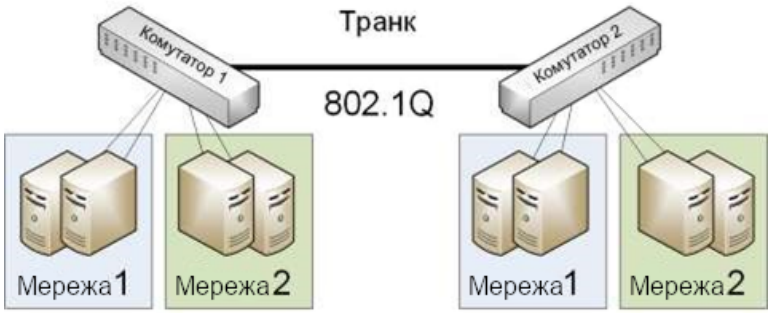
Мережа налаштована і потрібно її протестувати. Результат буде позитивним, якщо в межах своєї VLAN комп'ютери доступні, а комп'ютери з різних VLAN не доступні (мал. 4.14). Усі п'ять комп'ютерів знаходять в одній мережі 10.0.0.0/8, але вони знаходяться у різних віртуальних локальних мережах.

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Мал. 4.14. Пінг з PC1 на PC3 та PC4

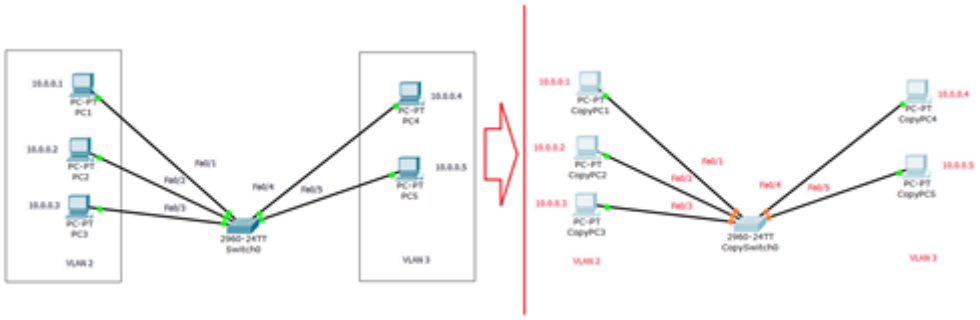
Практична робота 3. Налаштування VLAN із двома комутаторами.

На практиці часто виникає завдання поділу пристроїв, підключених до одного або кількох комутаторів на кілька локальних мереж, що не перетинаються. Якщо використовується тільки один комутатор, то це завдання вирішується шляхом конфігурування портів комутатора, вказавши кожному порту до якої локальної мережі він відноситься. Якщо ж використовується кілька комутаторів (мал. 4.15), необхідно між комутаторами крім даних додатково передавати інформацію до якої локальної мережі належить кадр. Для цього було розроблено стандарт 802.1Q.



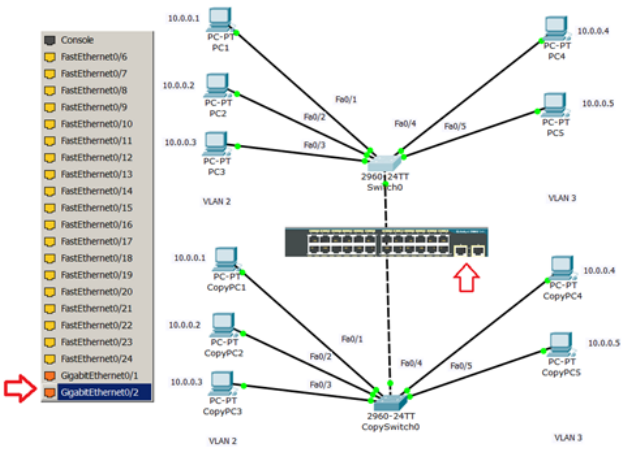
Мал. 4.15. Віртуальні локальні мережі (VLAN) з використанням двох комутаторів

Зробимо дублювання нашої мережі (яка була показана раніше на мал. 4.10). Для цього виділимо всю мережу інструментом **Select** (Виділити), і, утримуючи клавішу **Ctrl**, перетягнемо на нове місце робочої області програми. Так ми зробимо копіювання (мал. 4.16).



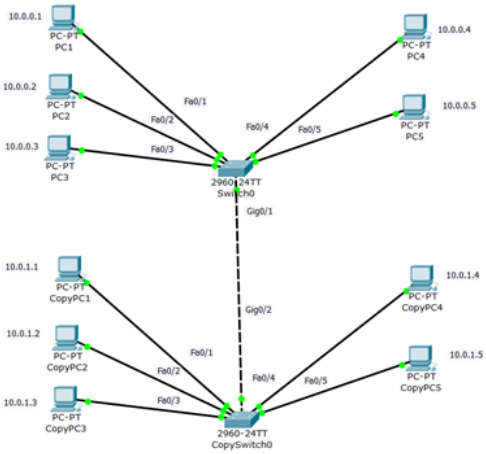
Мал. 4.16. Дублюємо мережу з одним комутатором

З'єднаємо комутатори перехресним кабелем (кросом) через найпродуктивніші порти – Gigabit Ethernet (мал. 4.17).



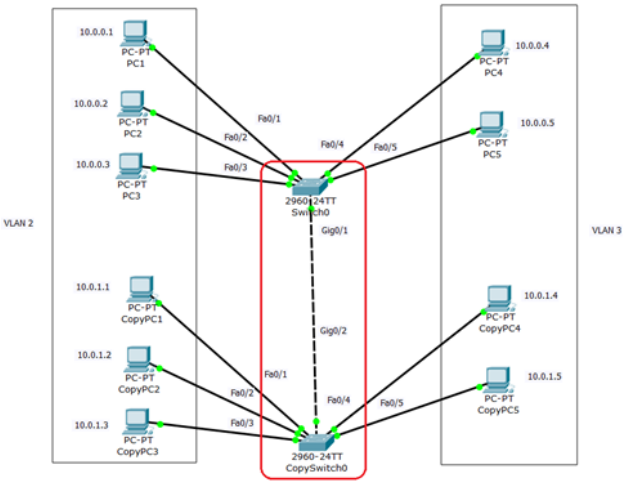
Мал. 4.17. З'єднуємо комутатори через Gigabit Ethernet порти

Тепер поправимо налаштування на дублікаті вихідної мережі (мал. 4.18).



Мал. 4.18. Налаштовуємо мережу-дублікат

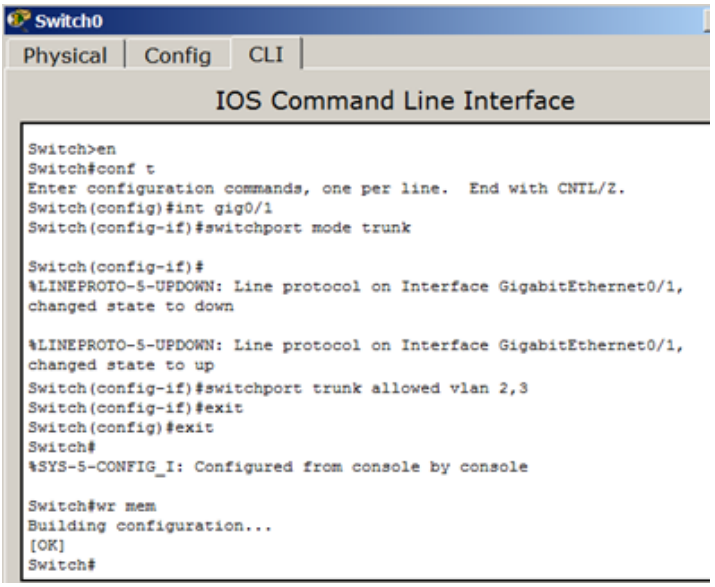
Вкажемо новий варіант підмереж VLAN2 і VLAN3, а також виділимо **trunk (транк)** зв'язок комутаторів (мал. 4.19).



Мал. 4.19. У мережі позначаємо підмережі VLAN2 та VLAN3

Налаштовуємо транк порт Gig0/1

При налаштуванні Gig0/1 на комутаторі Switch0 ми змінюємо стан порта та вказуємо vlan 2 та vlan 3 для роботи з ним (мал. 4.20).



Мал. 4.20. Налаштовуємо транк порт Gig0/1 на комутаторі Switch0

Налаштовуємо транк порт Gig0/2

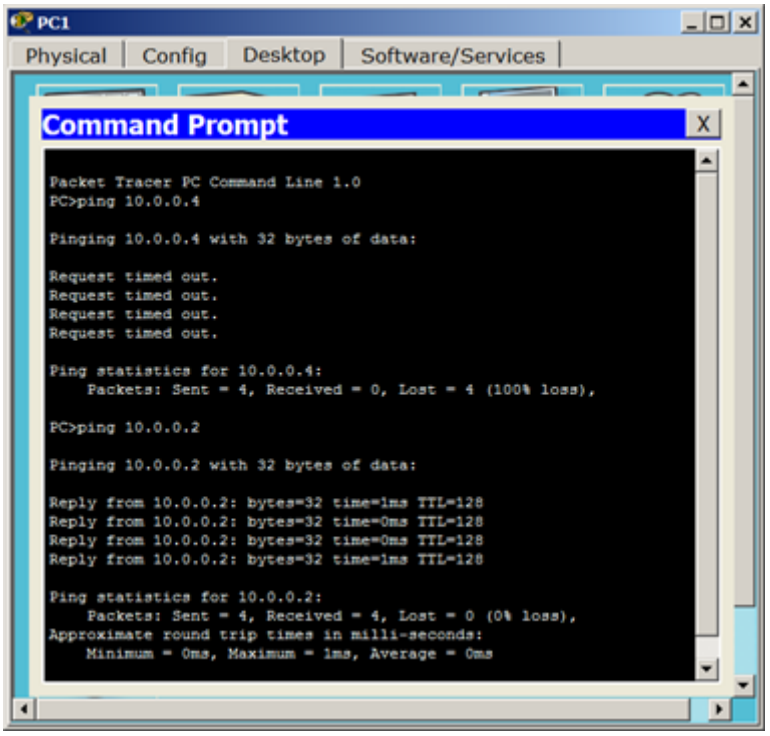
Транк порт Gig0/2 на комутаторі CopySwitch0 налаштовуємо аналогічно (мал. 4.21).

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int gi0/2
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allow vlan 2,3
Switch(config-if)#end
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#end
```

Мал. 4.21. Налаштовуємо trunk порт Gig0/2 на комутаторі CopySwitch0

Діагностика результатів роботи

Перевіряємо пінг з PC1 в різні vlan (мал. 4.22) - в межах своєї vlan ПК доступні, а між ПК різних vlan зв'язку немає.



Мал. 4.22. Пінг з PC1 в різні vlan

Контрольні запитання

1. Що таке VLAN ?
2. Які причини та переваги використання VLAN?
3. Що таке магістральний порт?
4. Що таке режим доступу порта?

Порядок виконання і здачі роботи

1. Вивчити теоретичну частину та методику виконання роботи.
2. В середовищі Packet Tracer виконати практичні завдання.
3. Оформити звіт.
4. Відповісти на контрольні запитання.
5. Продемонструвати виконання практичних завдань.

Зміст звіту

Звіт готується в електронному вигляді та вивантажується в систему електронного навчання moodle. Звіт має містити виконання теоретичного та практичного завдань з коментарями та скріншотами до кожного етапу.

Рекомендована література – основна

1. Буров Є.В. Комп'ютерні мережі. Підручник / Є.В. Буров // Вища освіта в Україні. - Л.: "Магнолія-плюс", 2015. – 262 с.
2. Блозва А.І., Матус Ю.В., Смолій В.В., Гусєв Б.С., Касаткін Д.Ю., Осипова Т.Ю., Савицька Я.А., Комп’ютерні мережі [навчальний посібник] - К.: Компрінт, 2017.-
   821 с.
3. Тарнавський Ю.А., Кузьменко І.М.. – Організація комп’ютерних мереж підручник: для студ. спеціальності 121 «Інженерія програмного забезпечення»та 122 «Комп’ютерні науки» –Київ : КПІ ім. Ігоря Сікорського, 2018. – 259 с.
4. Микитишин А.Г., Митник М.М., Стухляк П.Д., Пасічник В.В. Комп'ютерні мережі. Книга 2 [навчальний посібник] - Львів, "Магнолія 2006", 2017. - 328 с.

Додаткова література

1. Tanenbaum A., Wetherall D. Computer Networks, 6th Edition. – 2021.
2. Kurose J., Ross K. Computer Networking: A Top-Down Approach, 7th Edition. – 2017.

Зміст

Комп'ютерні мережі
Методичні рекомендації та завдання
для лабораторних робіт

Укладачі:

Олександр Матвій
Ігор Черевко

Комп’ютерний набір
Олександр Матвій, Ігор Черевко