МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Проєктування корпоративних мереж» для студентів спеціальності 123 – «Комп'ютерна інженерія»

Затверджено
редакційно-видавничою
радою університету НТУ «ХПІ»,
протокол № 1 від 15.02.2024

Харків – 2024

Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Проєктування корпоративних мереж» для студентів денної та заочної форм навчання спеціальності 123 – «Комп’ютерна інженерія» / уклад.: Мезенцев М.В. – Харків: НТУ «ХПІ», – 2024. – 110 с.

Укладач: М.В. Мезенцев

Рецензент проф. О.А. Серков

Кафедра комп’ютерної інженерії та програмування

ВСТУП

Корпоративна мережа – це досить складна структура, яка використовує різні типи зв’язку, комунікаційні протоколи та засоби підключення ресурсів. З точки зору зручності побудови та керованості мережі слід орієнтуватися на однотипне обладнання одного виробника. Однак практика показує, що постачальників, які пропонують максимально ефективні рішення для всіх завдань, не існує. Мережа, що працює, завжди є результатом компромісу – або це однорідна система, неоптимальна з точки зору ціни і можливостей, або більш складне в установці та управлінні поєднання продуктів різних виробників.

Найбільш поширеним підходом до проєктування інформаційних систем в даний час є використання експертних оцінок. Відповідно до цього підходу фахівці в області обчислювальних засобів, активного мережного обладнання та кабельних мереж на підставі наявного у них досвіду та експертних оцінок здійснюють проєктування корпоративної мережі, що забезпечує рішення конкретної задачі або класу задач. Цей підхід дозволяє мінімізувати витрати на етапі проєктування, швидко оцінити вартість реалізації інформаційної системи. Однак рішення, отримані з використанням експертних оцінок, носять суб’єктивний характер, вимоги до обладнання та програмного забезпечення також грішать суб’єктивністю, як і оцінка гарантій працездатності і развіваемо пропонованого проєкту системи.

В якості альтернативного може бути використаний підхід, що передбачає розробку моделі і моделювання поведінки корпоративної мережі.

Запропонований курс лабораторних робіт відповідає програмам Міністерства освіти і науки України та навчальним планам Національного технічного університету «ХПІ» з дисципліни «Проєктування корпоративних мереж». Лабораторний практикум допоможе розібратися у структурі операційних систем настільки, щоб можна було з успіхом працювати в мережі,

використовувати всі можливості, які вона надає, виконувати функції мережного адміністратора.

Основним середовищем для виконання лабораторних робіт є віртуальна машина Oracle VM VirtualBox, на якій установлені операційні системи Microsoft Windows. Дане середовище дозволяє виконувати складні експерименти з операційною системою й мережними настроюваннями, незалежно від конфігурацій реальних машин і мереж. Крім того у курсі розглядається пакет GNS3, який дозволяє будувати мережеві топології та досліджувати певні параметри та технології, які використовуються при проєктуванні та подальшому використанні корпоративних мереж.

ЛАБОРАТОРНА РОБОТА 1

ОСНОВИ РОБОТИ У ВІРТУАЛЬНІЙ МАШИНІ ORACLE VM VIRTUALBOX. ВСТАНОВЛЕННЯ ГОСТЬОВИХ ОПЕРАЦІЙНИХ СИСТЕМ

Мета роботи: навчитися працювати з віртуальними машинами Oracle VM VirtualBox; навчитися інсталювати гостьові операційні системи у віртуальній машині.

Хід роботи

Роботу віртуальної машини спрощено ілюструє рис. 1.1. На комп'ютері встановлюється базова операційна система, після чого встановлюється програмне забезпечення віртуальної машини (ВМ). Воно дозволяє встановити одну або кілька гостьових операційних систем і запускати в них програми, розроблені для даних ОС. Як ПО віртуалізації в наших лабораторних роботах буде використовуватися Oracle VM VirtualBox (версія 7.0).

Рисунок 1.1 – Схема роботи віртуальних машин

Створення віртуальної машини

З меню Пуск (Програми → Oracle VM VirtualBox) запустимо консоль керування віртуальними машинами (рис. 1.2). Натиснувши кнопку New, приступимо до створення нової машини.

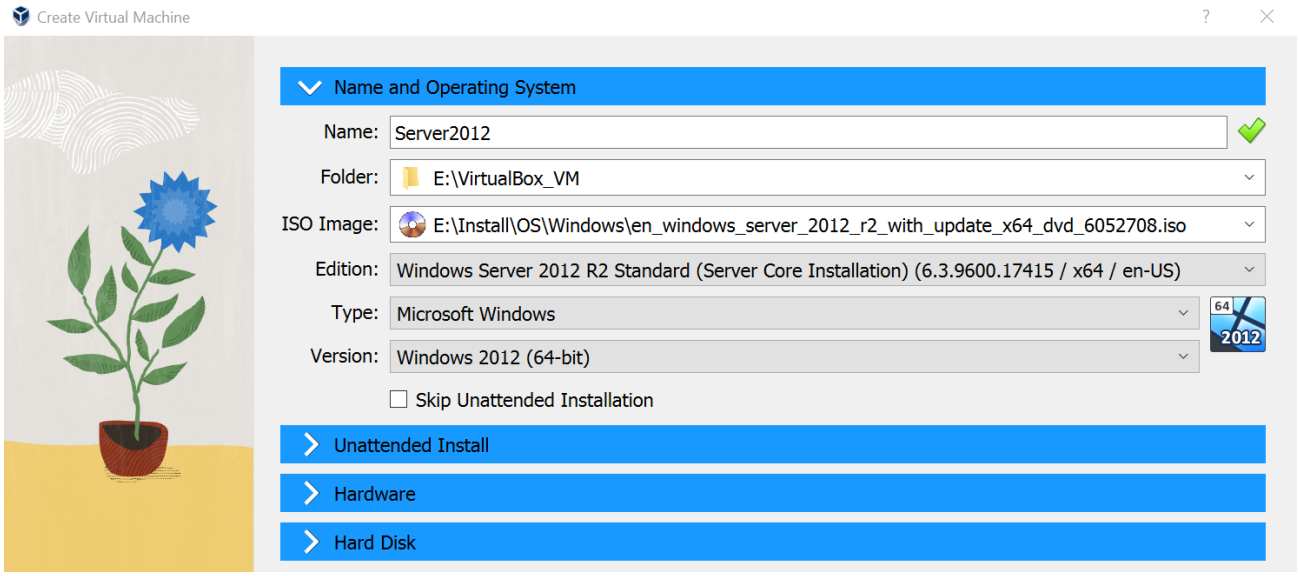

Рисунок 1.2 – Створення нової віртуальної машини

При створенні нової машини треба буде описати розташування її файлів (файла віртуальної машини з розширенням .vbox і файла віртуального жорсткого диска з розширенням .vdi). Після цих дій з натисканням кнопки запускається майстер, якому вказуємо:

- ім’я й розташування файла;
- ім’я образу з програмою встановлення ОС;
- тип установлюваної операційної системи. Якщо це ОС розробки Microsoft, будуть автоматично виставлені рекомендований обсяг оперативної пам’яті й віртуального жорсткого диска. У вікні вибору версії ОС оберіть Windows 2012 (64-bit);
- у наступних вікнах майстра встановимо обсяг виділюваній машині оперативної пам’яті (4Гб); укажемо, що треба створити новий віртуальний жорсткий диск (Create a virtual hard disk now) і залишимо пропонований розмір з 50 Гб. Розмір і ім’я можна залишити й за замовчуванням, зміна настроювань пропонується, «щоб потренуватися». Отут треба зазначити, що при використанні даного майстра файл віртуального диска створюється мінімального розміру й збільшується в міру потреби, тобто відразу 50 Гб не буде потрібно (рис. 1.3 – 1.4).

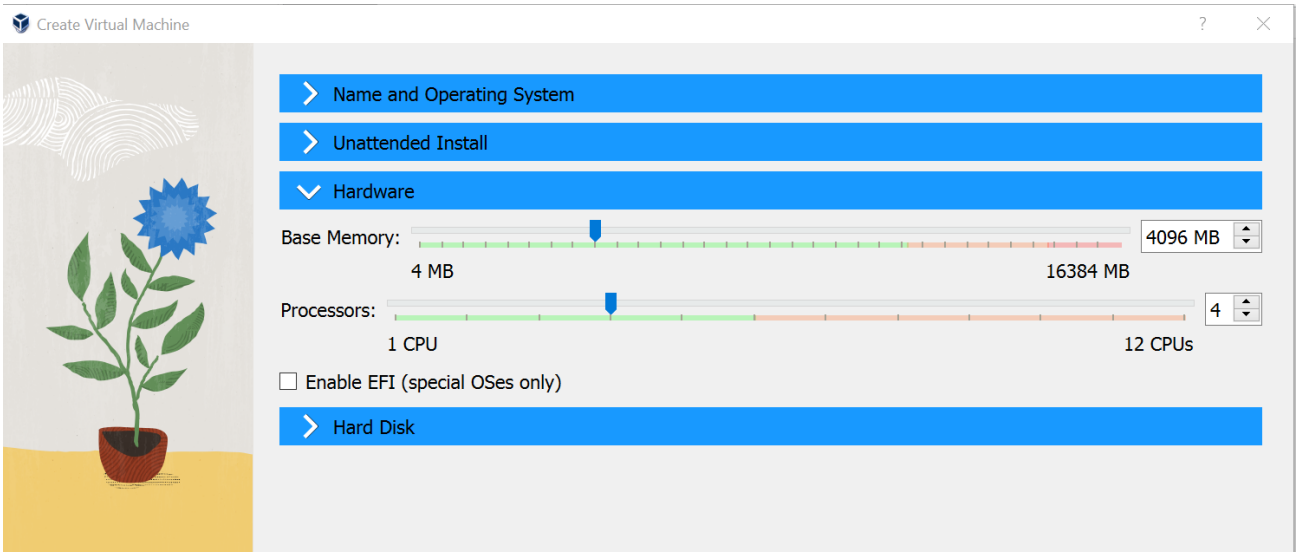

Рисунок 1.3 – Встановлення розміру пам’яті та кількості процесорів

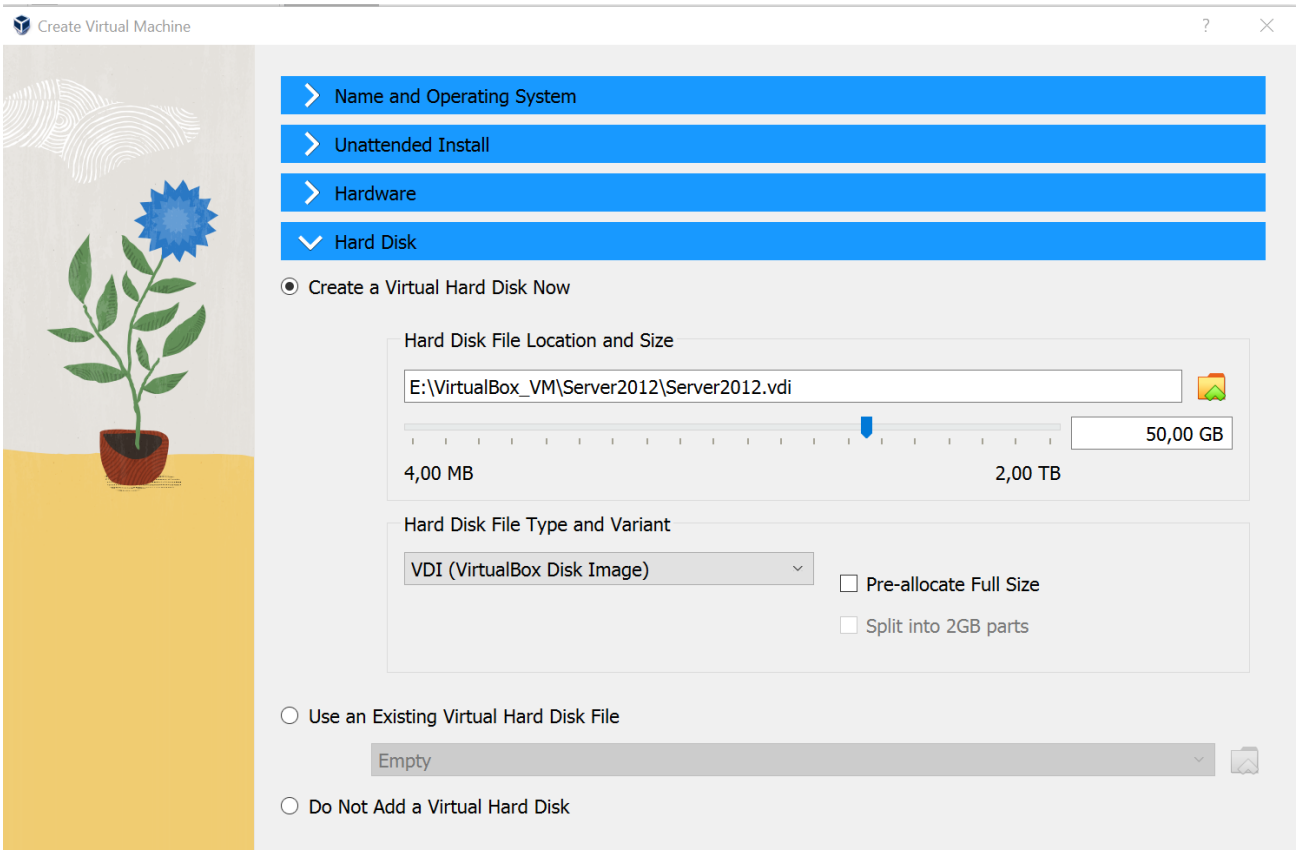

Рисунок 1.4 – Встановлення параметрів жорсткого диску

Після зроблених настроювань у вікні консолі (рис. 1.3) з’явиться нова віртуальна машина, яку можна запустити, виділивши її й натиснувши кнопку Start.

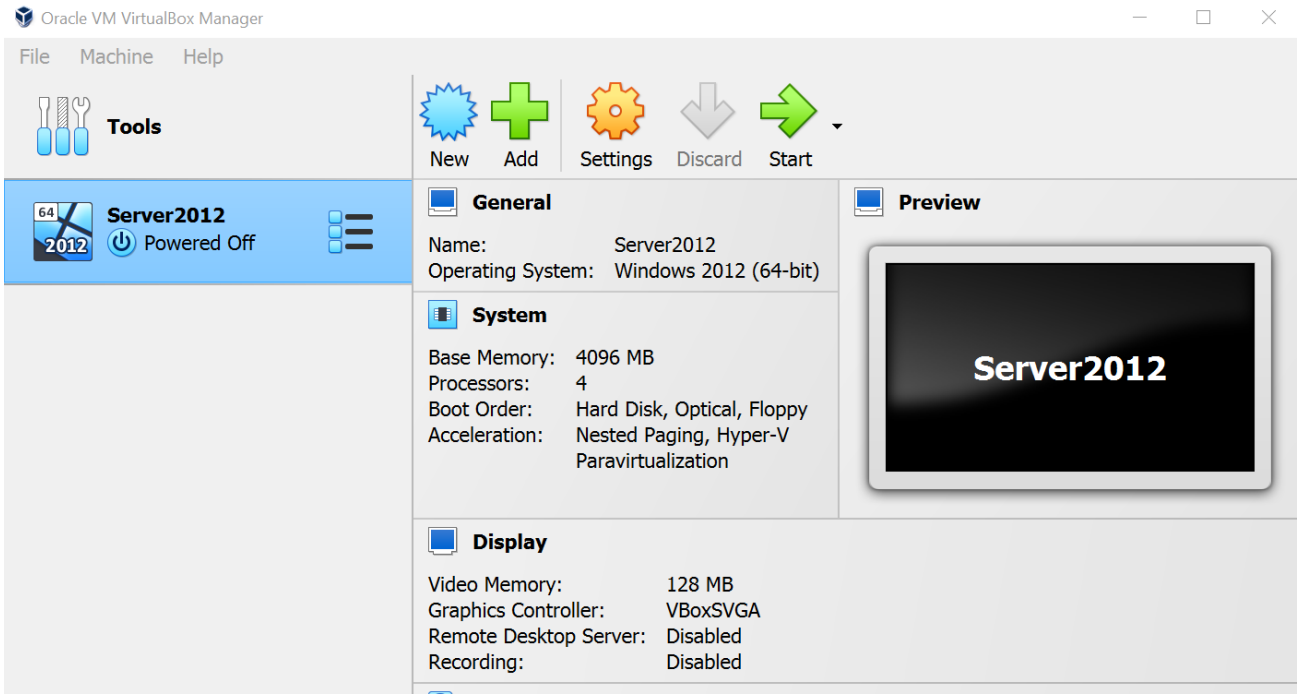

Рисунок 1.5 – Консоль керування віртуальними машинами

При цьому може з’явитися повідомлення про помилку «The virtual machine could not be started because there was not enough memory available on the host» – віртуальна машина не може бути запущена, тому що недостатньо пам’яті. Це відбудеться, наприклад, якщо на комп’ютері встановлено менше 1Гб оперативної пам’яті або в момент запуску віртуальної машини запущено багато інших програм. В останньому випадку проблема може бути розв’язана завершенням роботи тимчасово непотрібних запущених програм. Якщо ж мало фізичної пам’яті, то через настроювання віртуальної машини (кнопка Settings у консолі) можна спробувати трохи зменшити розмір пам’яті виділюваній віртуальній машині. Але треба пам’ятати, що мінімально рекомендований обсяг пам’яті для Windows Server – це 512 МБ, і сильно урізати його без втрати працездатності ОС не вийде.

Серед інших настроювань віртуальної машини хотілося б відзначити параметри мережі (рис. 1.6). Обравши розділ «Network», можна вказати кылькысть адаптерів віртуальної машини, а також параметри для кожного адаптера.

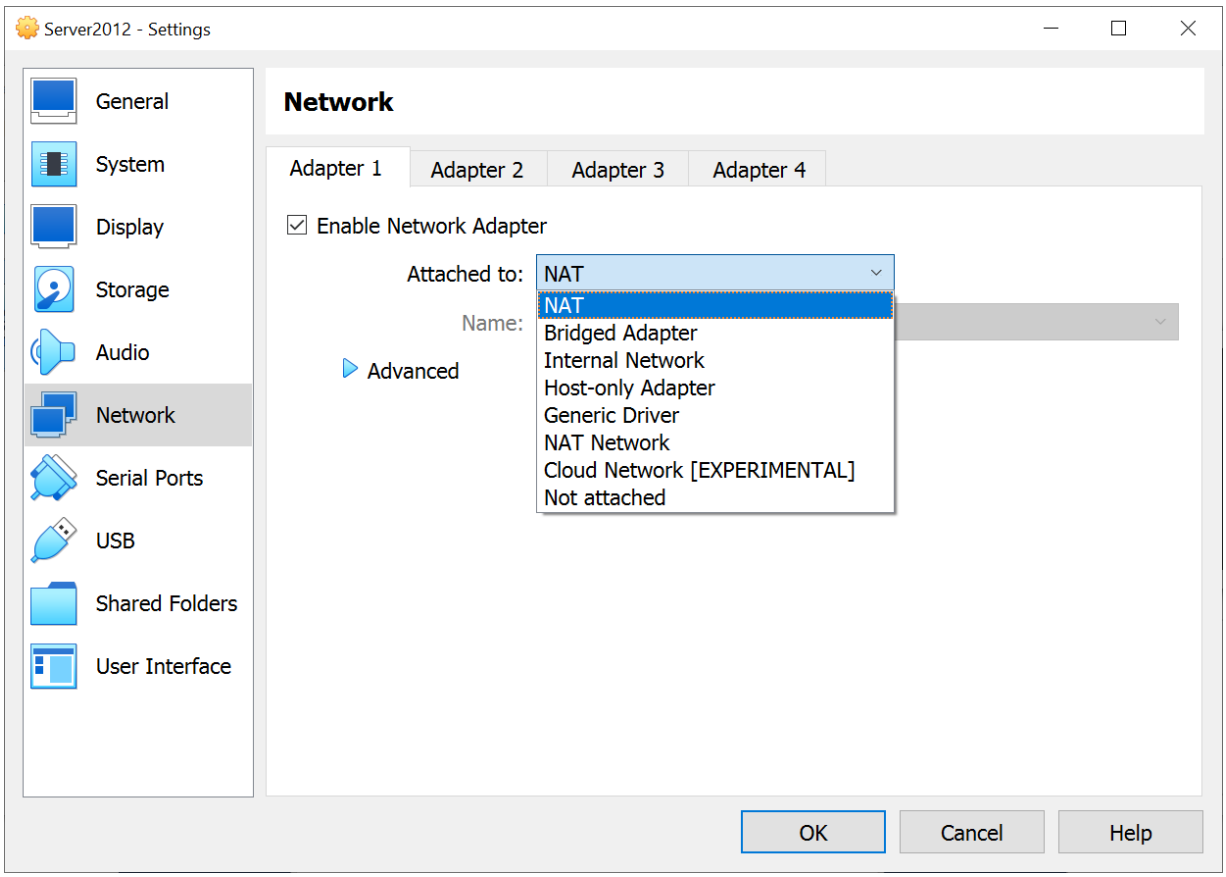

Рисунок 1.6 – Настроювання мережних адаптерів віртуальної машини

- **NAT** – дозволяє гостьовій операційній системі виходити в Інтернет, використовуючи при цьому приватний IP, який не доступний з боку зовнішньої мережі або для всіх машин локальної фізичної мережі. Таке мережеве налаштування дозволяє відвідувати web-сторінки, завантажувати файли, переглядати електронну пошту. І все це, використовуючи гостьову операційну систему. Однак, ззовні неможливо безпосередньо з'єднатися з такою системою, якщо вона використовує NAT.

Принцип трансляції мережевих адрес полягає в наступному. Коли гостьова ОС відправляє пакети на конкретну адресу віддаленої машини в мережі, сервіс NAT, що працює під VirtualBox, перехоплює ці пакети, витягує з них сегменти, що містять адресу пункту відправки (IP-адреса гостьової операційної системи) і здійснює їх заміну на IP-адресу машини-хоста. Потім знову упаковує їх і відправляє за вказаною адресою.

- **Bridged Adapter** – віртуальна машина працює так само, як і всі інші комп'ютери в мережі. У цьому випадку адаптер виступає в ролі моста між

віртуальною та фізичною мережами. З боку зовнішньої мережі є можливість безпосередньо з’єднуватися з гостьовою операційною системою.

Адаптер у режимі "Мережевий міст" підключається, минаючи хост, до роутера, який розподіляє IP-адреси всередині локальної мережі для всіх фізичних мережевих карток. VirtualBox з’єднується з однією із встановлених мережевих карток і передає пакети через неї безпосередньо; виходить робота моста, яким передаються дані. Як правило, адаптер у моделі "Мережевий міст" отримує стандартну адресу з діапазону 192.168.x.x від роутера. Тому віртуальна машина в мережі виглядає так, ніби це звичайний фізичний пристрій, який не відрізняється від інших.

- **Internal Network** – якщо потрібно налаштувати взаємозв’язок між декількома гостьовими операційними системами, що працюють на одному хості та можуть спілкуватися лише між собою, тоді можна скористатися режимом "Внутрішня мережа". Звичайно, для цієї мети можна використовувати режим "Мережевий міст", але режим "Внутрішня мережа" має більшу безпеку. У режимі "Мережевий міст" всі пакети відправляються та виходять через адаптер фізичної мережі, встановлений на машині-хості.
- **Host-only Adapter** – гостьові ОС можуть взаємодіяти між собою та з хостом. Але все це тільки всередині самої віртуальної машини VirtualBox. У цьому режимі адаптер хоста використовує свій власний, спеціально для цього призначений пристрій, який називається vboxnet0. Також їм створюється підмережа і призначаються IP-адреси мережевим картам гостьових операційних систем. Гостьові ОС неспроможні взаємодіяти з пристроями, що у зовнішньої мережі, оскільки вони підключені до неї через фізичний інтерфейс. Режим "Віртуальний адаптер хоста" надає обмежений набір служб, корисних для створення приватних мереж під VirtualBox для гостьових ОС.
- **Gereric Driver** – користувач сам вибирає драйвер мережного адаптера, який може бути входить до складу VirtualBox або завантажується з пакетом доповнень до VirtualBox. На даний момент існує 2 драйвера реалізують 2 режими роботи віртуального адаптера:

UDP Тунель. Режим зв'язку віртуальних машин, запущених різних хостах. Працює над існуючою мережевою інфраструктурою.

VDE (Віртуальний розподілений Ethernet). Цей режим може бути використаний для підключення розподілених віртуальних машин до віртуального комутатора Ethernet на Linux або FreeBSD хостах.

- **Мережа NAT** – кілька гостей за NAT не бачать один одного.
- **Not attached** – в даному режимі адаптер присутній у гостьовій системі, але поводиться так, ніби мережевий кабель до нього не ввімкнений.

Середовище виконання лабораторного практикуму може бути настроєне з використанням усіх перерахованих способів. Поки зупинимося на конфігурації з одним мережним адаптером і типом з'єднання NAT.

Після запуску віртуальної машини розпочинається стандартний процес встановлення ОС (рис. 1.7).

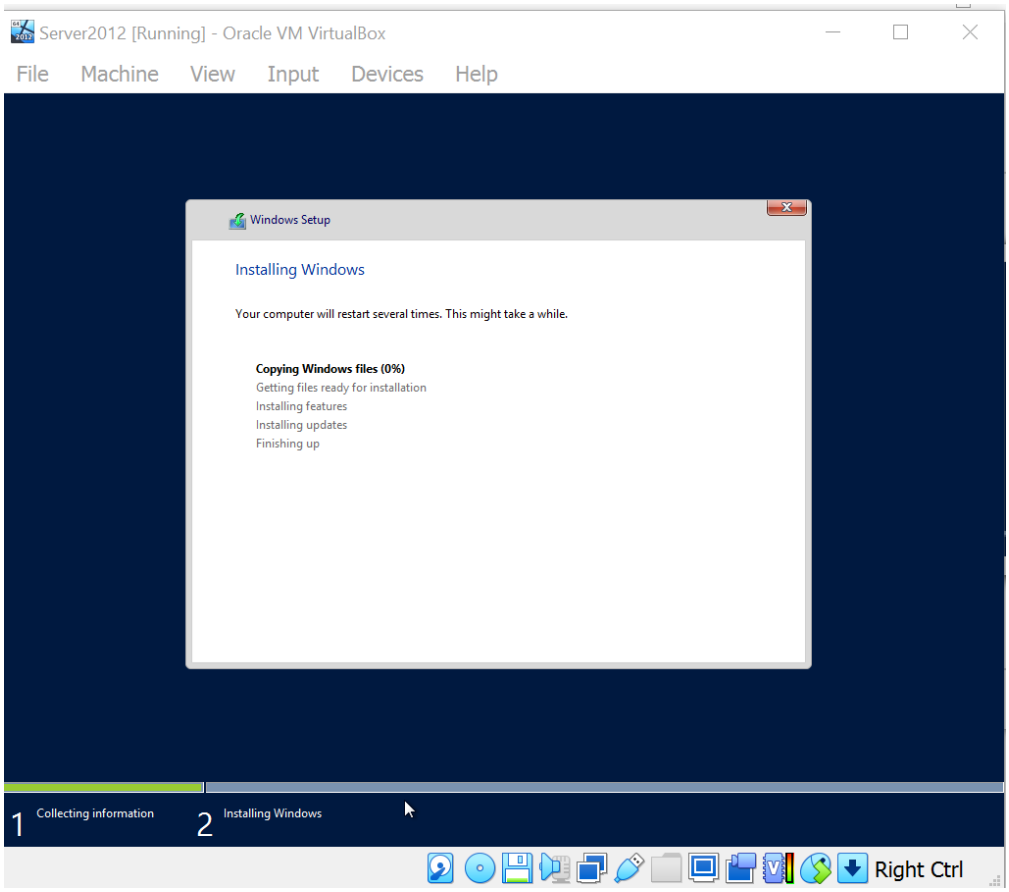

Рисунок 1.7 – Встановлення ОС Windows Server 2012

Індивідуальне завдання

1. Запустіть програму Oracle VM VirtualBox. Відкриється Oracle VM VirtualBox Console (Консоль віртуального комп'ютера). У цій консолі вам потрібно вибрати віртуальну машину із установленою операційною системою Microsoft Windows Server 2012 і відкрити її настроювання (**Settings**).

Перегляньте настроювання віртуальної машини:

- **General** – ім'я файлу віртуальної машини.
- **System** – обсяг використовуваної пам'яті. Рекомендується розподіляти наявну фізичну пам'ять нарівно між усіма запущеними віртуальними машинами, а також фізичним комп'ютером.
- **Storage** – місцезнаходження файлу жорсткого диска віртуальної машини.
- **Network** – мережні настроювання. Виберіть перший мережний адаптер. У списку типу адаптеру оберіть Host-only Adapter. Таким чином, створюється мережне з'єднання фізичного комп'ютера й віртуальної машини, що не впливає на реальну мережу.

Інші настроювання залишіть незмінними.

2. Вимкніть віртуальну машину. Виберіть у меню вікна віртуальної машини пункт **Machine**, потім **Stop** (**Закрити**). Існує три способи завершення роботи:

- **ACPI** – **Shutdown** (**Пуск** – **Вимикання**);
- **Power off** – повне вимикання, аналог «Вимикання» на фізичному комп'ютері;
- **Save state** – зберегти стан. На жорсткому диску в спеціальному файлі зберігається поточний стан віртуальної машини й при наступному старті робота починається з нього (аналог «Сплячого режиму»).

Виберіть **ACPI** – **Shut Down**.

3. Якщо у вашій редакції Windows Server відключено GUI інтерфейс, то необхідно його включити. Якщо доступ в Інтернет у сервера відсутній, нам доведеться вказати альтернативні джерело встановлення (за допомогою команди

powershell Install-WindowsFeature з параметром -Source). Для встановлення графічного інтерфейсу нам знадобиться дистрибутив Windows Server 2012. Припустимо, що iso-образ дистрибутиву Windows Server 2012 змонтували у пристрій, якому призначено букву D.

Далі потрібно визначити індекс встановленої версії Windows Server 2012 в інсталяційному wim образі. Для цього наберіть команду, яка відображає інформацію про вміст інсталяційного образу:

Dism /get-wiminfo /wimfile:D:\sources\install.wim

На сервері встановлено Windows Server 2012 Datacenter, нас цікавить дистрибутив SERVERDATACENTER, індекс якого 4.

Далі потрібно встановити відсутні компоненти (Server GUI) з wim файлу командою:

Install-WindowsFeature Server-Gui-Mgmt-Infra,Server-Gui-Shell -Restart -source:wim:d:\sources\install.wim:4

3. Для конфігурації встановленої гостьовий ОС нам будуть потрібні імена фізичного комп'ютера й назва робочої групи.

Існує два способи довідатися ім'я й робочу групу комп'ютера. Перший спосіб: відкрийте вікно системних властивостей (клацніть правою кнопкою миші по значкові **Мій комп'ютер** – **Властивості**). На вкладці **Ім'я комп'ютера** визначте ім'я комп'ютера й назву робочої групи.

Другий спосіб (за допомогою командного рядка): для визначення імені комп'ютера скористайтеся утилітою **hostname**.

Щоб довідатися назву робочої групи, застосуйте утиліту **nbtstat** (утиліта відображає інформацію про протокол NBT – Netbios через TCP/IP). У командному рядку введіть: nbtstat –a <ім'я комп'ютера>.

Випишіть ім'я фізичного комп'ютера й назву робочої групи.

Зміст звіту

1. Тема роботи.
2. Позначка виконання роботи.

3. Хід виконання індивідуального завдання. Для всіх завдань помістите у звіті скріншоти, що відбивають правильність виконання завдань.

4. Помістіть у звіт скріншоти, у яких відбиті:

– вікно **Ім’я комп’ютера** з назвою робочої групи віртуальної машини;

– результат виконання утиліти hostname;

– вікно **Мережне оточення**.

5. Висновки.

Контрольні питання

1. Які настроювання мережних адаптерів віртуальної машини існують у віртуальній машині Oracle VM VirtualBox?

2. Які типи операційних систем можуть бути використані як гостьові для віртуальних машин Oracle VM VirtualBox?

3. З яким розширенням зберігаються файли віртуальної машини та файли віртуального жорсткого диска?

4. Чи впливають параметри фізичного комп’ютера на швидкодію віртуальних машин?

ЛАБОРАТОРНА РОБОТА 2

НАСТРОЮВАННЯ ВЗАЄМОДІЙ ВІРТУАЛЬНИХ МАШИН ORACLE VM VIRTUALBOX

Мета роботи: навчитися працювати з віртуальними машинами Oracle VM VirtualBox; навчитися інсталювати гостьові операційні системи у віртуальній машині.

Індивідуальне завдання.

Завдання 1. Запустіть програму Oracle VM VirtualBox. Відкриється Oracle VM VirtualBox Console (Консоль віртуального комп'ютера).

Створіть нову віртуальну машину, у яку буде встановлена й гостьова несерверна ОС Windows (як приклад оберіть Windows 10 (рис. 2.1).

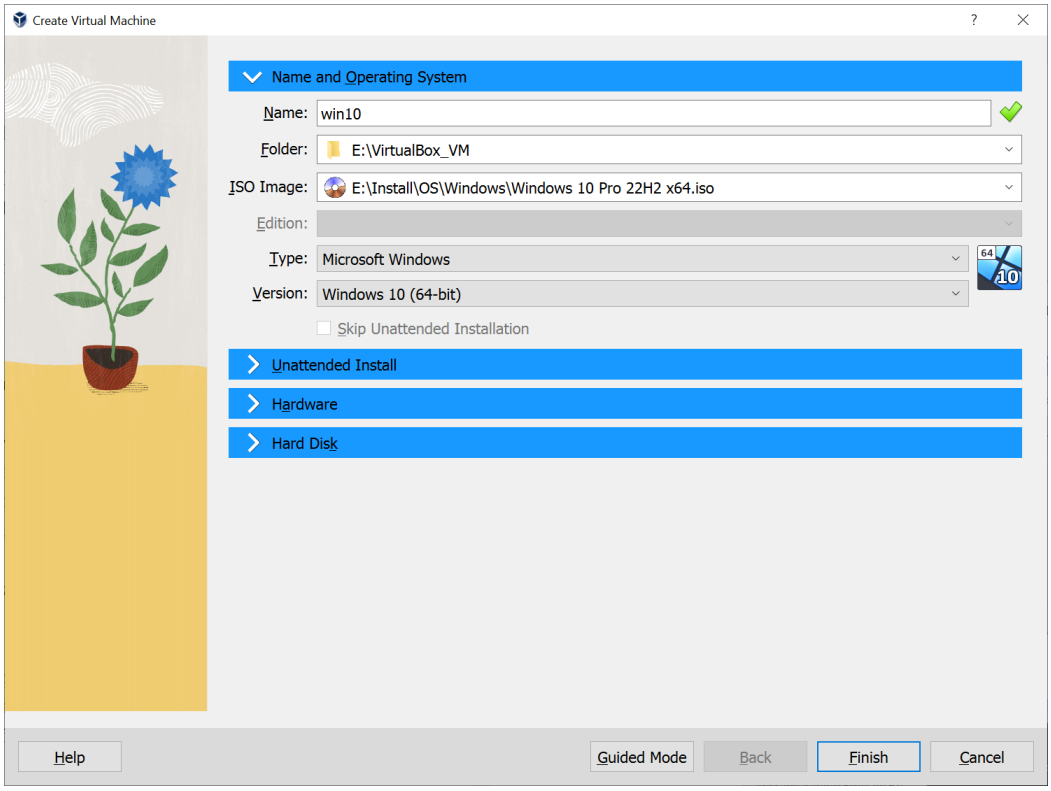

Рисунок 2.1 – Обрання ОС для робочої станції

Налаштуйте параметри ОЗП та процесору та створіть новий віртуальний диск з рекомендованими параметрами (рис. 2.2 і рис. 2.3).

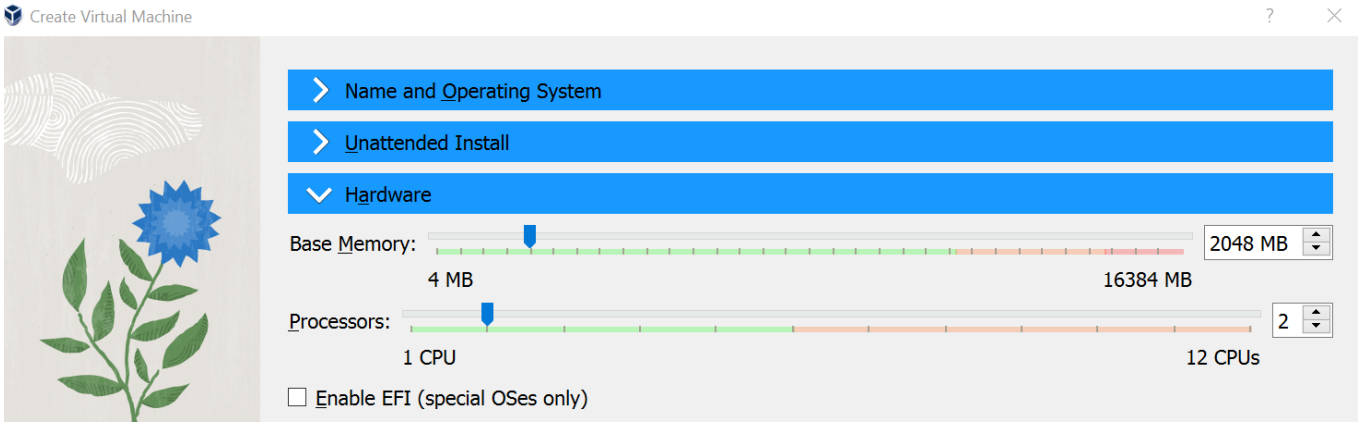

Рисунок 2.2 – Налаштування параметрів ОЗП та процесору

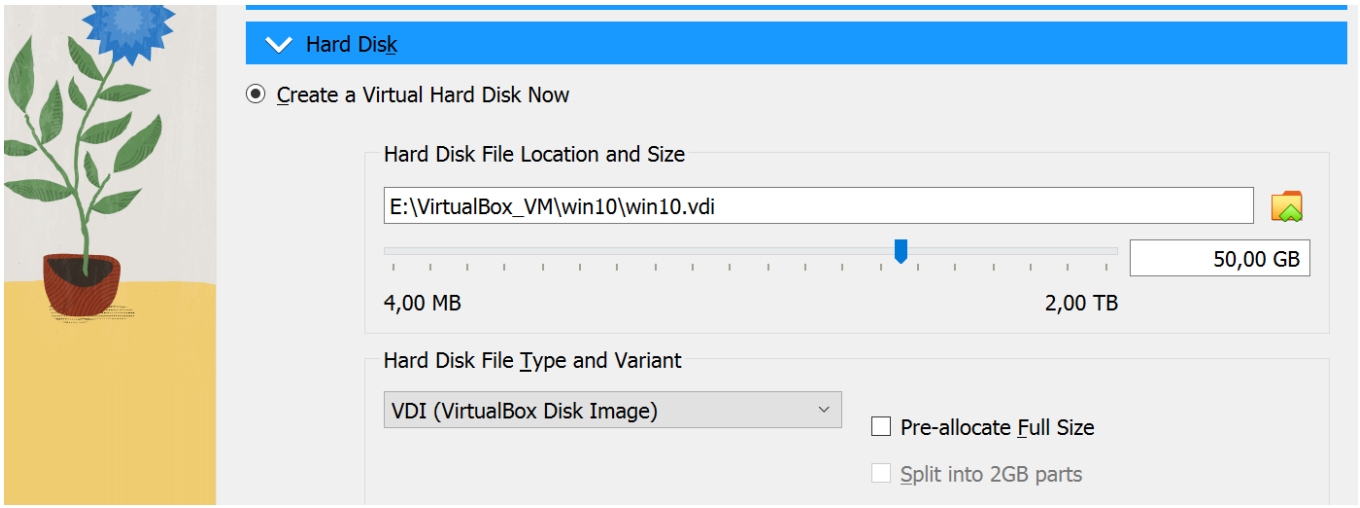

Рисунок 2.3 – Налаштування віртуального диску для робочої станції

Далі необхідно виділити нову віртуальну машину й натиснути "Start". З'явиться вікно завантаження у якому необхідно обирати певні параметри, які запитує інсталятор ОС. При установці гостьової операційної системи використовуйте ім'я робочої групи таке саме, як і на фізичному комп'ютері.

Завдання 2. У кожної гостьовий ОС у настроюваннях віртуальної машини (кнопка Settings у консолі віртуальної машини) вибрати параметр мережного адаптера **Host-only Adapter** (рис. 2.4).

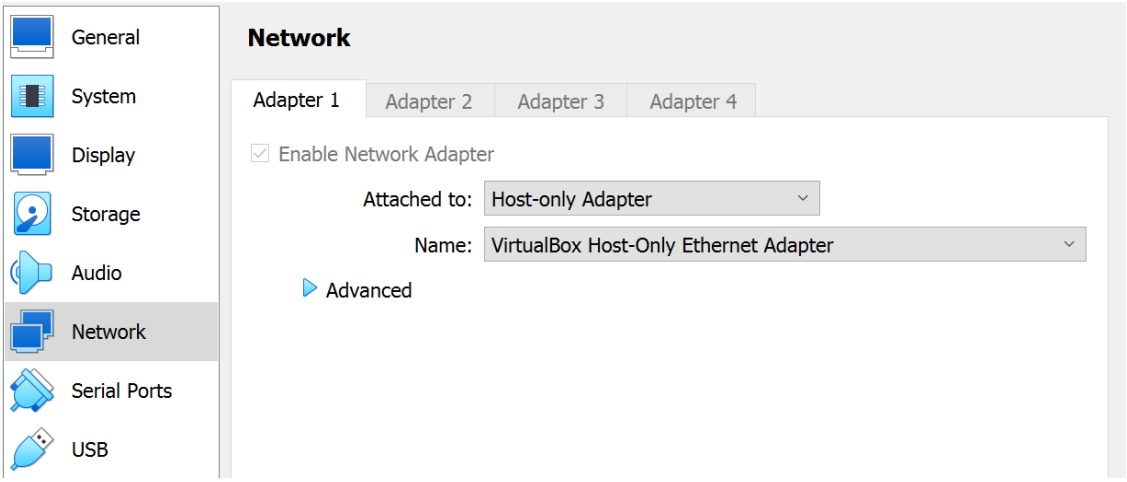

Рисунок 2.4 – Налаштування віртуального диску для робочої станції

Запустити обидві ОС. У кожній настроїти параметри підключення по локальній мережі. Наприклад, у Windows Server обрати:

Пуск → Панель керування → Мережні підключення → Відкрити (правою кнопкою миші).

Відкрити властивості «Підключення по локальній мережі», вибрати Протокол Інтернету(TCP/IP) → Властивості (рис. 2.5) і встановити там такі значення, наприклад:

– IP-адреса: 192.168.100.1;
– маска підмережі: 255.255.255.0;
– шлюз за замовчуванням: 192.168.100.1.

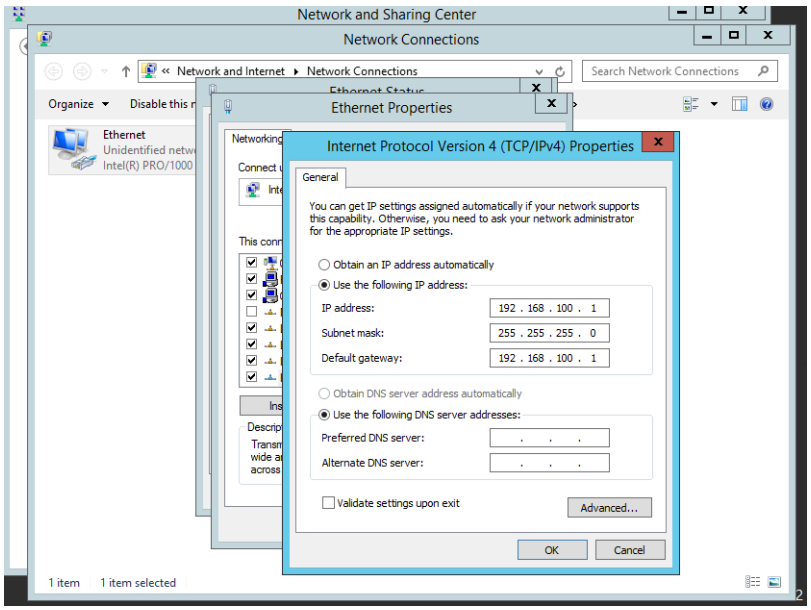

Рисунок 2.5 – Параметри мережного адаптера для гостьової Windows Server 2012

У правому нижньому куті з’явиться повідомлення "Підключення по локальній мережі встановлене".

Аналогічні дії з настроювання параметрів мережного підключення виконати у Windows, але з IP-адресою, наприклад: 192.168.100.2.

Виконайте команду ping на адресу іншої гостьової ОС (наприклад з гостьової ОС Windows Server на гостьову ОС Windows 10).

За допомогою команди **ifconfig /all** дізнатися IP-адресу свого фізичного комп’ютера. З кожної гостьовий ОС перевірити доступ до фізичного комп’ютера через **ping X.X.X.X** (підставити IP-адресу фізичного комп’ютера).

Відповіді доповнювати скріншотами з результатами настроювань мережних підключень у гостьових ОС, скріншотами з настроюваннями мережних адаптерів віртуальних машин і скріншотами виконання команд **ping** у кожній з ОС і **route print** у кожній гостьовій ОС.

Завдання 3. Додайте ще одну віртуальну машину з несерверною ОС. Це можна зробити виконавши клонування встановленої ОС (рис. 2.6).

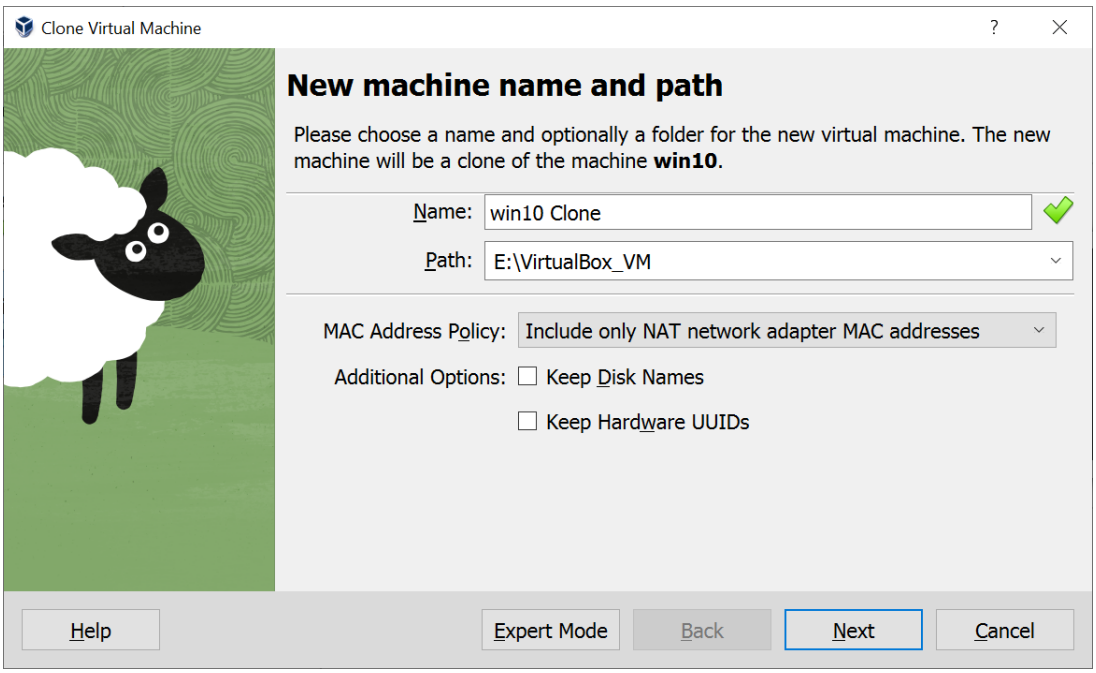

Рисунок 2.6 – Клонування ОС у консолі VirtualBox

Встановити у властивостях мережного підключення обидві гостьові ОС IP-адреси й маски підмережі так, щоб усі три комп'ютери знаходились в одній підмережі. Перевірити командою **ping** з'єднання між трьома ОС. Перебрати всі можливі варіанти в налаштуваннях адаптерів віртуальних машин з розділу "Network".

Відповіді доповнювати скріншотами з результатами настроювань мережних підключень у гостьових ОС, скріншотами з настроюваннями мережних адаптерів віртуальних машин і скріншотами виконання команд **ping** у кожній з ОС і **route print** у кожній гостьовій ОС.

Завдання 4. Встановити в гостьових ОС IP-адреси з різних підмереж, наприклад, 192.168.100.100 і 192.168.200.200, і об'єднати дві підмережі за допомогою шлюзу на основі віртуальної машини з Windows Server.

Відповіді доповнювати скріншотами з результатами настроювань мережних підключень у гостьових ОС, скріншотами з настроюваннями мережних адаптерів віртуальних машин і скріншотами виконання команд **ping** у кожній з ОС і **route print** у кожної гостьовий ОС.

Зміст звіту

1. Тема роботи.

2. Мета виконання роботи.

3. Хід виконання індивідуального завдання. Для всіх завдань помістіть у звіті скріншоти, що відбивають правильність виконання завдань.

4. Висновки.

Контрольні запитання

1. Визначте, при яких установках мережних адаптерів та IP-адрес існує зв’язок тільки між гостьовими ОС?

2. Визначте, при яких установках мережних адаптерів та IP-адрес існує зв’язок між гостьовими ОС і фізичним комп’ютером?

3. Визначте, при яких установках мережних адаптерів та IP-адрес існує зв’язок між окремо взятою гостьовою ОС і фізичним комп’ютером, але не існує зв’язку між гостьовими ОС?

ЛАБОРАТОРНА РОБОТА 3

ДОСЛІДЖЕННЯ ПРОТОКОЛІВ TCP/IP

Мета роботи: Ознайомлення з принципами побудови стеку протоколів TCP/IP, а також системних утиліт, що працюють на цих протоколах та здобуття практичних навичок їх використання.

Теоретична частина

Протоколами в світі комунікацій називають розподілені алгоритми, що визначають, яким чином здійснюється обмін даними між фізичними пристроямиабо логічними об'єктами (процесами). Під сімейством протоколів TCP/IP в широкому сенсі розуміють зазвичай весь набір реалізацій стандартів RFC. Проте, загальним і основоположним елементом для всіх цих протоколів є Internet Protocol (IP). Цей протокол, власне, і реалізує поширення інформації по IP-мережі. Протокол IP здійснює передачу інформації від вузла до вузла мережі у вигляді дискретних блоків – пакетів. При цьому IP не несе відповідальності за надійність доставки інформації, цілісність або збереження порядку потоку пакетів і, таким чином, не вирішує з необхідною для додатків якістю задачу передачі інформації. Цю задачу вирішує два інших протоколу – TCP (Transmission Control Protocol, протокол управління передачею даних) і UDP (User Datagram Protocol, дейтаграмний протокол передачі даних), які, як то кажуть, "знаходяться" над IP (тобто використовують процедури протоколу IP для передачі інформації, додаючи до них свою додаткову функціональність). TCP і UDP реалізують різні режими доставки даних. TCP, як то кажуть, – протокол зі встановленням з'єднання. UDP (як, власне, і IP) є дейтаграмним протоколом, тобто таким, що кожен блок інформації (пакет) обробляє і поширює від вузла до вузла не як частина деякого потоку, а як незалежна одиниця інформації – дейтаграма (datagram).

Необхідність в UDP обумовлена тим, що IP працює як мережевий протокол, що вирішує завдання доставки інформації від вузла до вузла, тоді як UDP "вміє" розрізняти застосунки і передає інформацію від застосунка до застосунка (яких, відмітимо, на одному мережевому вузлі, на приклад на сервері, може бути декілька). Вище, над транспортними протоколами TCP

або UDP, занходяться протоколи, що реалізовують ті або інші прикладні служби, такі як обмін файлами (File Transfer Protocol, FTP) і повідомленнями електронної пошти (Simple Mail Transfer Protocol, SMTP), термінальний доступ до видалених серверів (Telnet).

Сімейство протоколів TCP/IP має високу складність, яка може викликати збої в мережі, коли щось працює неправильно. У даній роботі пропонується огляд методології пошуку несправностей і деяких спеціальних діагностичних утиліт TCP/IP. Розглянуті різні завдання при пошуку і усуненні проблем в TCP/IP і всіляких інструментах, які допоможуть налаштувати мережу.

Утиліти для пошуку проблем в TCP/IP

ARP – дозволяє переглядати і змінювати таблиці трансляції апаратних адрес, які використовуються протоколом ARP. Може бути використана на локальному комп'ютері для пошуку записів з невірними адресами.

HOSTNAME – виводить на екран ім'я локального вузла.

IPCONFIG – виводить всі значення параметрів настройки мережі. Особливо корисна на комп'ютерах, що використовують DHCP.

NBTSTAT – виводить статистику протоколу і список поточних TCP/IP-з'єднань, використовуваних NetBT. Дуже корисна утиліта NETSTAT. Аналогічна NBTSTAT. Виводить тільки TCP/IP-статистику і список TCP/IP-

з'єднань.

NSLOOKUP – виводить інформацію про сервери DNS. Доступна, тільки якщо була проведено встановлення TCP/IP.

PING – найбільш корисна утиліта. Перевіряє можливість найпростішої мережевої взаємодії з одним або кількома віддаленими комп'ютерами.

ROUTE – керує таблицями маршрутизації.

TRACERT – визначає шлях до зазначеного вузла, відправляючи ICMP exo-запити і збільшуючи значення параметра TTL (час життя).

При роботі з протоколами використовуються також утиліти: Event Viewer – відстежує помилки і події.

Performance Monitor – аналізує продуктивність сервера.

Network Monitor – аналізує мережеві протоколи на низькому рівні. Registry Editor – дозволяє переглядати і змінювати значення параметрів реєстру.

Методи пошуку несправностей в TCP/IP

Запам’ятайте: процес пошуку несправностей зазвичай доводиться повторювати – можна не досягти успіху з першого разу. У разі TCP/IP загальним методом пошуку несправностей може бути такою:

1. Визначте проблему або симптоми в яких вона себе проявляє. Це буває найважчим кроком. Хоча проблема може на перший погляд здаватися викликаної якимось одним елементом, процес виключення і детективна робота можуть вказати на що-небудь зовсім інше.

2. Виключіть те, що працює правильно, щоб звузити "коло підозрюваних".

3. Дослідіть спочатку фізичний рівень, а потім кожен рівень над ним; 90 відсотків збоїв в мережах викликані поганим кріпленням кабелю.

4. Висуньте гіпотезу.

5. Перевірте вашу гіпотезу.

6. Проаналізуйте дані.

7. Докладайте дії, потрібні для усунення проблеми.

Складіть список, в якому вкажіть, що працює і що – ні, потім вивчіть список, щоб ізолювати збої і відмови. Не забудьте використовувати Event Viewer для визначення результатів будь-яких змін. Взагалі кажучи, краще з самого початку перевірити, чи правильно налаштовано TCP/IP на комп’ютері. Потім перевірте, що між комп’ютером і мережним вузлом існує з’єднання і шлях, почавши з перевірки локального устаткування. Спробуйте декілька разів використовувати команду PING з різною кількістю пакетів в випадкові моменти часу і намалюйте графік відсотка успіху, щоб перевірити стабільність каналу зв’язку. Потім визначте, чи існують проблеми з IP-адресацією, потім – із визначенням імен та, нарешті, розгляньте всі проблеми з NetBIOS, які можуть існувати.

Використання IPCONFIG для перевірки конфігурації

Використання IPCONFIG для отримання інформації про конфігурацію комп'ютера, який має проблеми, – розумний початок для пошуку TCP/IP-проблем. Така корисна інформація, як IP-адреса, маска підмережі і шлюз за замовчуванням, може бути отримана за допомогою цієї утиліти командного рядка. При використанні з ключем /all утиліта IPCONFIG виводить дуже докладний звіт про всі використовувані інтерфейси, включаючи всі налаштовані послідовні порти.

Ключі команди IPCONFIG Ключ/Опис:

All – виведення повної інформації; за замовчуванням виводяться тільки IP-адреса, маска підмережі і шлюз за замовчуванням для кожного мережевого інтерфейсу.

Renew – цей ключ доступний тільки на клієнтах DHCP та викликає оновлення параметрів конфігурації. Введіть ім'я адаптера (яке виводиться при виконанні команди IPCONFIG без параметрів) для оновлення налаштувань конкретного адаптера.

Release – цей ключ доступний тільки на клієнтах DHCP та звільняє поточні значення параметрів конфігурації. Введіть ім'я адаптера (яке виводиться при виконанні команди IPCONFIG без параметрів) для звільнення значень параметрів для конкретного адаптера. Цей ключ корисний для користувачів переносних комп'ютерів.

Використання PING для перевірки зв'язку

PING – ваша перша лінія захисту при пошуку несправностей. PING є утилітою, яка перевіряє наявність зв'язку на IP-рівні, відправляючи на вказану IP-адресу або комп'ютер із заданим ім'ям NetBIOS ICMP exo-запити. Ви можете використовувати PING для визначення апаратних проблем, несумісних налаштувань і збоїв в каналах зв'язку. Спробуйте спочатку в якості параметра передати IP-адресу віддаленого вузла, це найпростіший випадок. Синтаксис команди такий:

ping IP-адреса

Для того щоб побачити список доступних ключів, введіть ping -?. Ви можете вибрати розмір пакета, кількість пакетів, записати використаний шлях, задати, яке значення TTL використовувати, якому вузлу відправляти запити, вибрати тип використовуваної служби і т.п. Як можна побачити, це дуже корисна діагностична утиліта командного рядка; вона також використовується в Unix-подібних системах.

Щоб ефективно використовувати PING, виконайте наступні кроки:

1. Надішліть запит на адресу локального адаптера для перевірки локальної установки і настройки: PING 127.0.0.1.2.

Надішліть запит на IP-адресу локального комп'ютера, щоб переконатися, що мережа розпізнає його: PING <IP-адреса локального вузла>.

3. Потім перевірте зв'язок зі шлюзом за замовчуванням. Це дозволить перевірити, чи працює шлюз, а також чи доступний він локальному вузлу: PING <IP-адреса шлюзу>.

4. Надішліть ехо-запит на IP-адресу віддаленого вузла для перевірки роботи маршрутизатора: PING <IP-адреса віддаленого вузла>.

Шлюз повинен знаходитися в тій же логічної підмережі, що і IP-адреса локального вузла. Якщо шлюз за замовчуванням не знаходиться в тій самій підмережі, вузол зможе взаємодіяти тільки з вузлами, що перебувають в одній із ним підмережі.

Ключі команди PING можуть допомогти підлаштуватися під практично будь-яку ситуацію. Наприклад, за замовчуванням PING очікує кожного відповіді тільки 750 мілісекунд, після чого відповідь вважається неодержаною. Ключ -w може використовуватися для встановлення більш тривалого тайм-ауту. Це може виявитися корисним в тих ситуаціях, коли віддалена система пов'язана з локальним каналом зв'язку з великою затримкою, наприклад супутниковим, в якому відповідь може прийти через більш тривалий час. Нижче наведено список ключів PING і їх короткий опис.

Ключі команди PING:

-t – відправка запитів до переривання роботи. -a – перетворення адрес в імена вузлів.

-n (число) – відправлення зазначеного числа запитів; за замовчуванням відправляється 4 запити.

-l (довжина) – відправлення exo-запитів зазначеної довжини (значення за замовчуванням – 64 байта; максимально допустиме значення – 8192).

-t – час встановлення значення TTL для пакетів, що відправляються. -V – встановлення вказаного значення типу обслуговування.

-r (число) – запис шляху у відповідне поле. Може бути вказано від одного до дев'яти вузлів.

-k (список) – маршрутизація пакетів через зазначені вузли. Максимальне допустиме IP число вузлів – 9. Послідовні вузли не можуть бути розділені шлюзами (strict source routed).

-w (тайм-аут) – встановення тайм-ауту в мілісекундах.

Для того щоб визначити проблеми з IP-адресацією, можна використовувати утиліти ARP, ROUTE і TRACERT.

Використання ARP

ARP – це ще одна дуже корисна утиліта, що дозволяє переглядати і змінювати таблицю ARP на локальному вузлі, а також переглядати кеш ARP і знаходити будь-які проблеми.

Протокол визначення адрес (ARP, Address Resolution Protocol) дозволяє знайти апаратну адресу потрібного вузла. Для підвищення ефективності на кожному вузлі або маршрутизаторі на деякий час кешуються вже певні відповідності між IP-адресою і апаратною адресою, утиліта ARP опитує цей кеш. Наявність такого кеша дозволяє знизити кількість повторних широкомовних запитів. Однак, за замовчуванням кеш оновлюється з 10-хвилинним інтервалом для забезпечення правильності визначення адрес.

Ключі утиліти ARP досить корисні. Наприклад, щоб зробити пошук запису, що відноситься до певного вузла, використовують ключ -a. Синтаксис команди такий:

arp -a IP-адреса -N IP-адрес_інтерфейса

Крім того, можна використовувати такі форми цієї команди:

arp -d IP-адреса [IP-адреса_інтерфейса] – ця команда видаляє запис, що відповідає заданій IP-адресі.

arp -s IP-адреса Ethernet-address [IP-адреса_інтерфейса] – ця команда створює запис в кеші ARP, встановлюючи відповідність між заданими Ethernet- і IP-адресами. Апаратна Ethernet-адреса задається в вигляді шести байтів в шістнадцятирічному форматі, розділеному дефісами. IP-адреса задається в стандартному десятковому форматі. Створювана запис стає статичною і не видаляється з кешу згодом, однак вона буде втрачена після перезавантаження комп'ютера.

Використання NSLOOKUP

NSLOOKUP – дуже потужна утиліта, що дозволяє виводити інформацію, отриману від серверів DNS. Природно, NSLOOKUP доступна, тільки якщо була проведена установка TCP/IP і доступний сервер DNS. Ця команда має наступний синтаксис:

nslookup [-параметр] [імя_вузла 1 - [сервер]]

NSLOOKUP має два режими: інтерактивний і неінтерактивний, які використовуються в залежності від того, як багато даних вам потрібно отримати. Для того щоб отримати певний запис з сервера DNS, використовуйте неінтерактивний режим, вказавши IP-адресу або ім'я вузла, яке повинно бути визначено, як перший аргумент. В якості другого аргументу вкажіть IP-адресу або ім'я сервера DNS. Якщо другий аргумент опущений, буде використовуватися сервер DNS за умовчанням.

В інтерактивному режимі можна виробляти послідовний пошук інформації. У цьому випадку використовують в якості першого аргументу дефіс (-), а другий аргумент або не вказують (щоб використовувати сервер DNS за умовчанням), або вводять ім'я або IP-адресу, щоб використовувати певний сервер. Для того щоб завершити роботу в інтерактивному режимі, треба натиснути Ctrl + C. Запам'ятайте, що в інтерактивному режимі невідомі команди будуть трактуватися як імена вузлів. Для того, щоб NSLOOKUP сприйняла вбудовану команду як ім'я вузла, введіть перед нею символ "\". З

утилітою NSLOOKUP може бути використано більше 25 параметрів, але загальна довжина командного рядка не повинна перевищувати 256 символів.

Усунення проблем з маршрутизацією

У Windows є MPR – мультипротокольний маршрутизатор (Multiprotocol Router). Він може використовуватися для підтримки маршрутизації на комп'ютерах з одним або декількома мережевими інтерфейсами. MPR використовує протокол управління маршрутизацією (RIP, Routing Information Protocol) для TCP/IP і IPX. Розглянемо дві утиліти, що дозволяють проводити пошук проблем з маршрутизацією – ROUTE та TRACERT.

ROUTE

ROUTE – діагностична утиліта, яка дозволяє маніпулювати мережевими таблицями маршрутизації. Вона використовує файл Networks для перетворення імен вузлів призначення в адреси. Для того щоб утиліта ROUTE працювала вірно, необхідно, щоб мережеві номера були вказані в цьому файлі коректно; тобто всі чотири октети були б записані в десятковому форматі. Наприклад, мережевий номер 10.10.1 повинен бути зазначений у файлі Networks як 10.10.1.0; додаткові нулі приєднуються, щоб утворити потрібну кількість октетів.

Щоб визначити, чи викликана проблема з помилками в IP-адресації, перевірте шлях, який обирається для відправки пакетів. Проблема може бути пов'язана з невірною таблицею маршрутизації або відмови маршрутизатора. Якщо отримується відповідь на команду PING від локального вузла, але не отримується відповідь від маршрутизатора, це свідчить про проблеми з маршрутизатором. Якщо не отримується відповідь на PING від вузлів за маршрутизатором, проблема може полягати в таблицях маршрутизації.

Команда ROUTE print дозволяє виводити таблиці маршрутизації на екран. Інші ключі команди ROUTE: add, delete і change – дозволяють відповідно додавати, видаляти і змінювати записи в таблиці маршрутизації.

TRACERT

Щоб перевірити маршрутизатори на шляху до вузла призначення, використовуйте утиліту TRACERT. Якщо вузол призначення не може бути досягнутий, то можна побачити, який маршрутизатор не працює; якщо мережа працює повільно, TRACERT покаже скільки часу витрачається на передачу пакетів від одного маршрутизатора іншому. У наступному прикладі шлюз визначив, що до вузла 10.10.0.1 немає шляху. Це означає, що або маршрутизатор налаштований невірно, або мережа 10.10.0.0 не існує (або задана невірна IP-адреса).

C:> \ tracert 10.10.0.1

Tracing route to 10.10.0.1 over a maximum of 30 hops 192.54.48.1 reports: Destination net unreachable. Trace complete.

Пошук проблем за допомогою спостереження

WireShark дозволяє захоплювати вхідний і вихідний трафіки локального комп'ютера. Щоб виділити необхідну для подальшого аналізу інформацію, можна визначати фільтри. Фільтри можуть ґрунтуватися на апаратній адресі відправника або одержувача пакета, на адресі, що використовується протоколом, а також на збігу зі зразком. Фільтри для виведення дозволяють ізолювати потенційні проблеми і зменшити обсяг інформації, яка повинна бути проаналізована. Виведений на екран звіт складається з вікна, що містить коротку інформацію, вікна з докладним описом інформації та шістнадцятирічного виведення інформації.

Самостійна робота

1. Повторити принципи роботи протоколів по матеріалах лекцій або по електронних посібниках з TCP/IP.

2. Провести експерименти з утилітами відповідно до порядку їх викладу в даній роботі, якщо вони вимагають введення параметрів, можна

обмежитися трьома варіантами параметрів для кожної утиліти. Експерименти проводяться в межах локальної мережі з лабораторної роботи №2.

3. Ознайомитися з налаштуваннями TCP/IP в операційній системі Windows в віртуальній мережі з ЛР №2.

4. Оформити звіт.

Вміст звіту

1. Мета роботи.

2. Короткий виклад контрольних питань.

3.Результати використання утиліт.

4.Висновки

Контрольні запитання

1. Перерахувати протоколи, що входять в сімейство TCP/IP.
2. Привести функції протоколу TCP.
3. Привести функції протоколу IP.
4. Привести функції протоколу UDP.
5. Існуючі різновиди протоколу IP і в чому їх відмінність.
6. Перерахуєте класи ip-мереж і дайте їм характеристику.
7. Що таке маска мережі і для чого вона використовується?
8. Поясніть принципи маршрутизації в IP-мережах.
9. Як відбувається перетворення IP-адреси в фізичний?
10. Поясніть призначення протоколу ICMP.
11. Поясніть процес з’єднання в протоколі TCP.
12. Як забезпечується достовірність передачі в протоколі TCP?
13. Поясніть принципи управління потоком в протоколі TCP. 14.

Коли застосовується протокол UDP?

15. Поясніть принципи безкласової маршрутизації.
16. Чому використовується фрагментація дейтаграм в протоколі IP?
17. Приведіть класифікацію утиліт для відлагодження протоколів TCP/IP.
18. Приведіть методику пошуку несправностей в IP-мережах.

ЛАБОРАТОРНА РОБОТА 4

ОСНОВИ АДМІНІСТРУВАННЯ ДОМЕНУ WINDOWS. ВСТАНОВЛЕННЯ СЛУЖБ ACTIVE DIRECTORY ТА DNS

Мета роботи: навчитися встановлювати контролер домену; навчитися додавати комп'ютер до домену; навчитися встановлювати й конфігурувати службу каталогів.

Хід роботи

Перед початком роботи до запуску віртуальних машин зробіть настроювання, які вказують, що віртуальна машина буде використовувати фізичний мережний адаптер. У консолі керування Virtual PC у властивостях віртуальної машини Networking виберіть для використання один з мережних адаптерів фізичного комп'ютера.

Для того щоб використовувати встановлену операційну систему Windows Server для виконання деякої серверної функції, треба встановити роль (role). Роль включає одну або кілька служб (role services), необхідних для виконання певної функції. Наприклад, File Services (файлові служби) або Web-server (IIS). Коли роль поєднує кілька служб, то можуть установлюватися або всі відразу, або окремі служби. Додаткова функціональність може бути отримана шляхом установки програмних модулів, називаних компонентами (feature). Приклад компоненти – це SMTP Server. Ролі й компоненти можуть бути як незалежними, так і взаємозалежними.

Додати або вилучити ролі й компонент можна за допомогою оснащення Пуск → Адміністрування → Керування даним сервером (рис. 4.1).

Одне із завдань поточної лабораторної роботи – зробити наш сервер контролером домену Windows. Для цього знадобиться встановити роль Контролер домену (Active Directiry) і виконати настроювання параметрів.

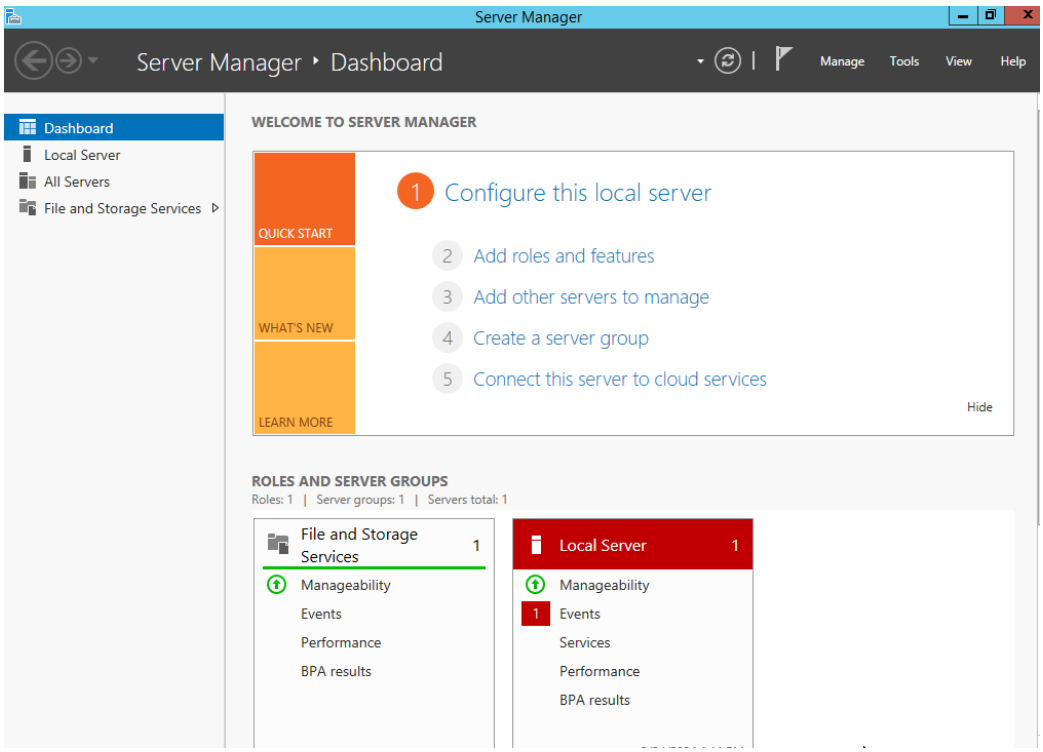

Рисунок 4.1 – Оснащення "Керування даним сервером"

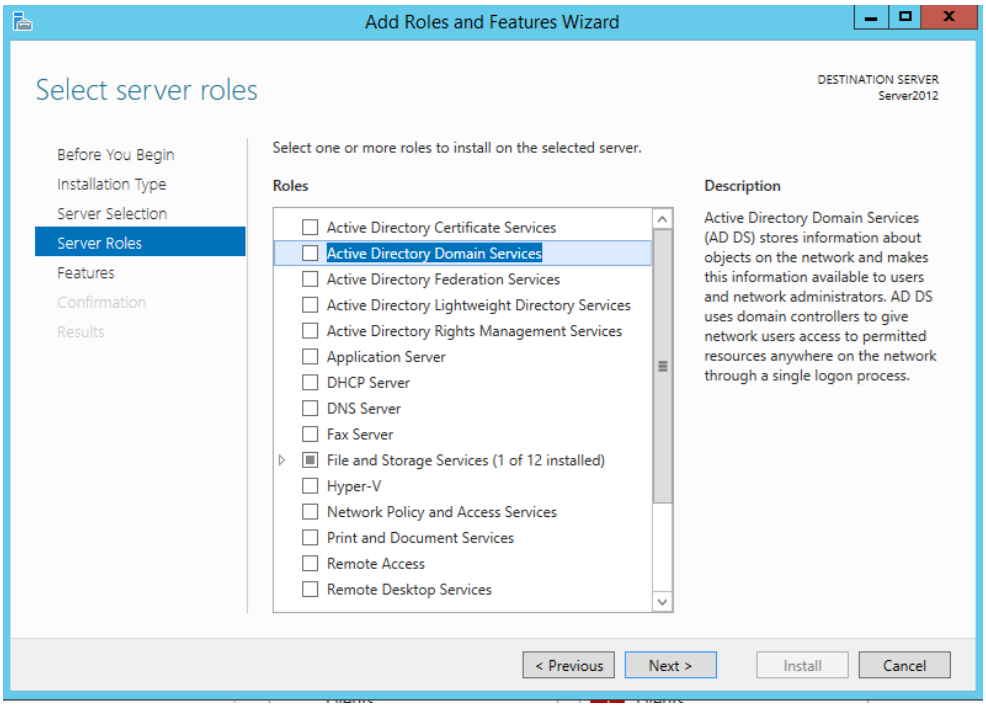

Рисунок 4.2 – Вибір майстра

Домен Windows логічно поєднує кілька комп’ютерів для того, щоб можна було їх централізовано адмініструвати. Прикладом адміністративного завдання може бути створення такого облікового запису, щоб користувач міг входити під

нею на будь-який комп’ютер свого підрозділу організації. У цьому випадку, щоб такий запис увести тільки один раз (а не на кожному комп’ютері), потрібно вести єдину базу даних з інформацією про користувачів і комп’ютери. Подібна база називається каталогом, а розроблена Microsoft служба каталогу – Active Directory. Сервери, на яких працює служба і які, зокрема, виконують перевірку користувачів з доменними обліковими записами, називаються контролерами домену.

Після встановлення служби необхідно налаштувати контроллер домену. Для цього перейдіть у налаштування (рис. 4.3).

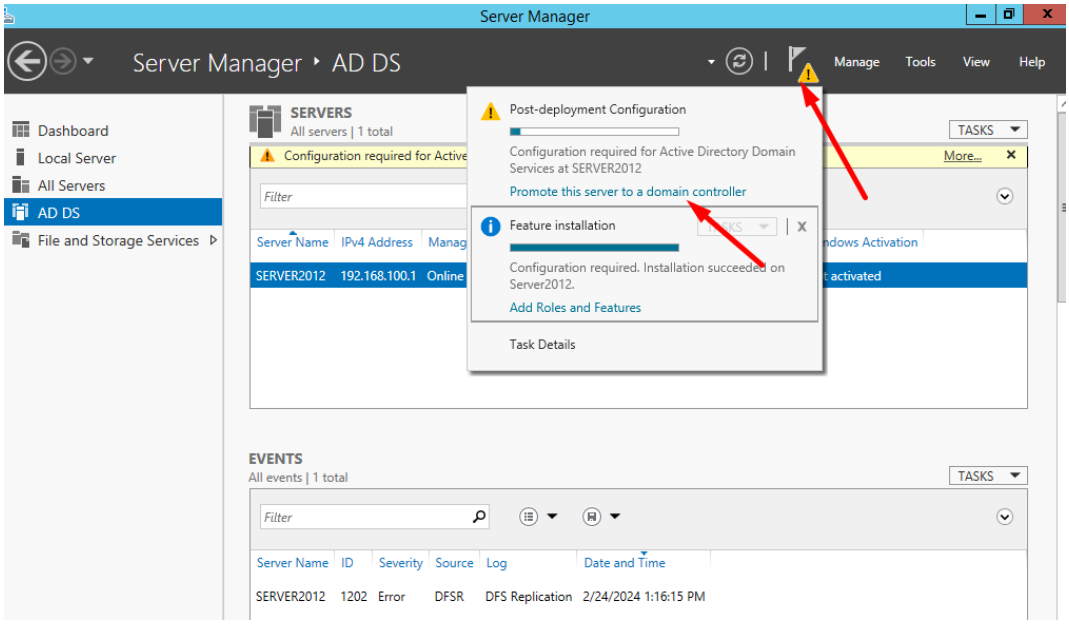

Рисунок 4.3 – Налаштування контролеру домену

Інформація про об’єкти мережі зберігається в каталозі. Для цього спочатку створюються визначення об’єктів, які містяться в службову структуру, називану схема каталогу. При створенні об’єкта нового типу потрібно спочатку помістити в схему його визначення. Сукупність доменів, що використовують єдину схему каталогу й загальну конфігурацію, називають лісом доменів (forest).

У вікнах майстра, поданих на рис. 4.4 і рис. 4.5, показано, що створюється новий домен, тобто конфігурується перший контролер першого (кореневого) домену в локальній мережі.

У наступному вікні майстра запитується ім’я домену (рис. 4.5). Для наших лабораторних робіт будемо використовувати ім’я **PKMXXXNN.test**, де XXX – номер групи, NN – номер студента в журналі групи.

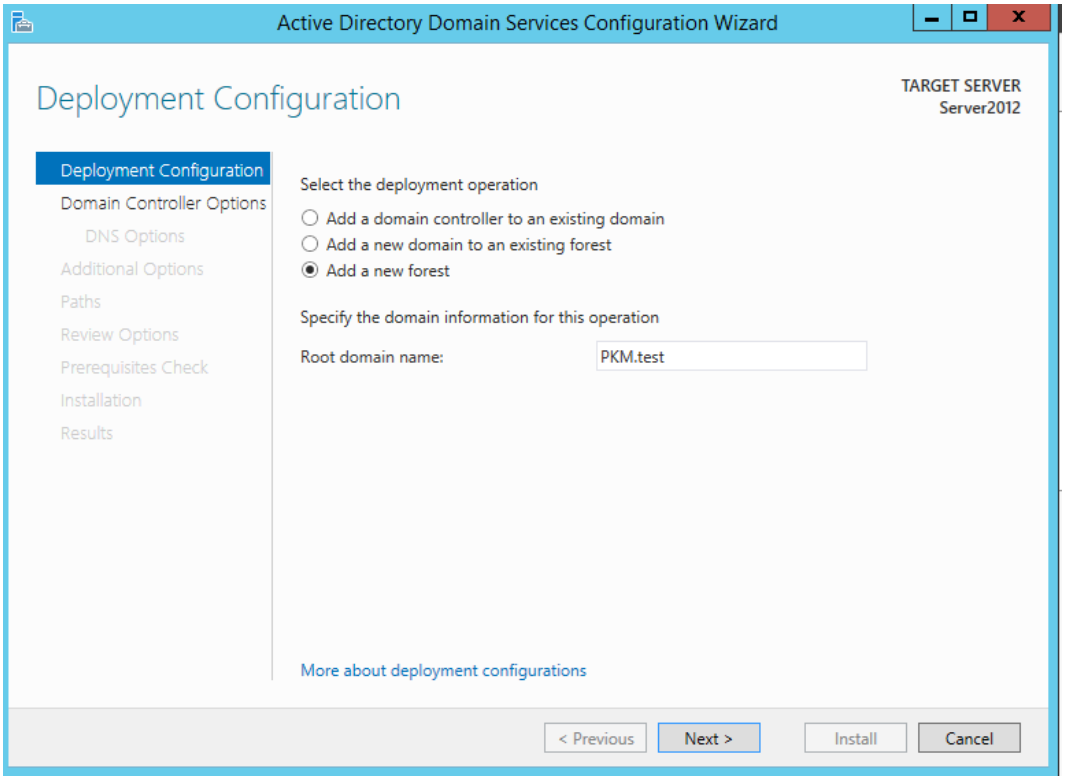

Рисунок 4.4 – Вибір типу домену

Встановіть пароля для доступу до налаштувань контролеру домена (пароль повинен відповідати певних параметрам) (рис. 4.5).

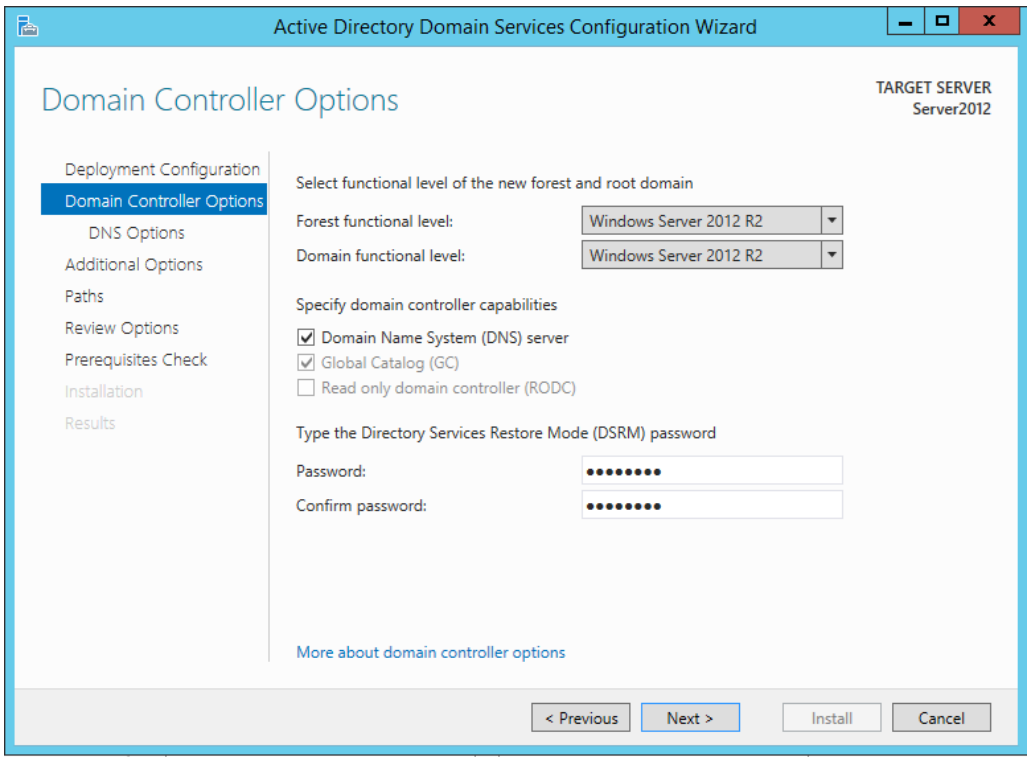

Рисунок 4.5 – Налаштування функціонального рівня та паролю

Укажіть Netbios-Ім’я домена (рис. 4.6).

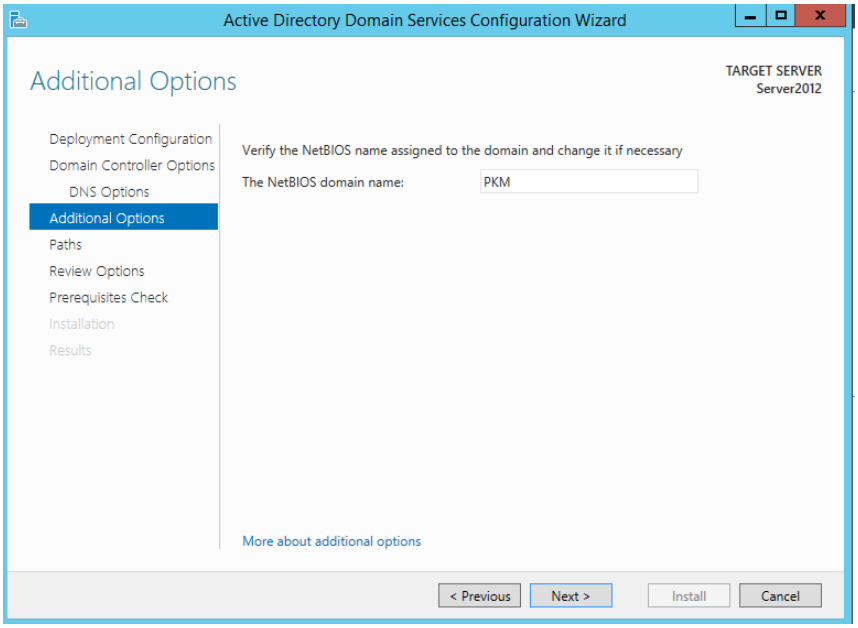

Рисунок 4.6 – Вибір Netbios-імені домену

Переконайтеся, що для розміщення бази даних і протоколу обраний шлях **C:\WINDOWS\NTDS**, а для розміщення каталогу **SYSVOL** зазначений шлях **C:\WINDOWS\SYSVOL** (рис 4.7).

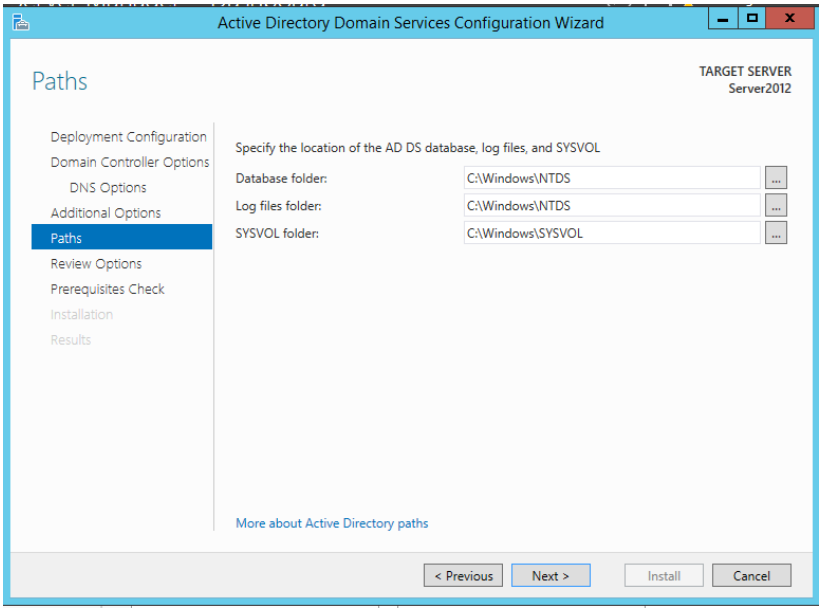

Рисунок 4.7 – Встанолення шляхів

Далі, тому що в нашій віртуальній мережі поки немає DNS сервера, майстер автоматично встановить DNS сервер. Служба DNS використовується для дозволу доменних імен комп'ютерів в IP-адреси. У домені Windows клієнтські комп'ютери за допомогою DNS одержують інформацію про контролери домену, тому хоча б один DNS сервер необхідний.

Служба доменних імен — це головний метод дозволу імен в Windows Server. При використанні DNS необхідне розгортання ролі сервера доменних служб Active Directory.

Перевірте правильність обраних установок і дочекайтеся закінчення установки Active Directory.

Як окремий випадок DNS може зберігати й обробляти й зворотні запити, визначення імені хоста за його IP-адресою — IP-адреса за таблицею відповідності перетвориться в доменне ім'я і посилати запит на інформацію типу PTR.

Для цього використовуються вже наявні засоби DNS. Справа в тому, що із записом DNS можуть бути зіставлені різні дані, у тому числі і яке-небудь

символьне ім’я. Існує спеціальний домен in-addr.arpa, записи в якому використовуються для перетворення IP-адреси у символьні імена. Наприклад, для одержання DNS-Імені для адреси 11.22.33.44 можна запросити в DNS-сервера запис 44.33.22.11.in-addr.arpa, і той поверне відповідне символьне ім’я. Зворотний порядок запису частин IP-адреси пояснюється тим, що в IP-адресах старші октети розташовані на початку, а в символьних DNS-Іменах старші (що знаходяться ближче до кореня) частини розташовані наприкінці.

При запиті здійснюється зчитування запису PTR, що містить шукане доменне ім’я. Якщо запис відсутній, то вважається, що не має IP-адреси зворотного DNS. DNS -запис in-addr.arpa виглядає так:

78.56.34.12.in-addr.arpa. IN PTR domain.ltd.

Це буде означати, що імені хосту domain.ltd відповідає IP-адреса 12.34.56.78.

Об’єкти в ієрархії DNS ідентифікуються за допомогою записів ресурсів *(Resource Record)*. Вони використовуються для виконання основних операцій пошуку користувачів і ресурсів усередині зазначеного домену й унікальні для утримуючого їх домену, що їх утримує. Розрізнять такі типи записів.

1. **Початковий запис зони SOA (Start of Authority).** Указує, на якому сервері зберігається еталонна інформація про даний домен, містить контактну інформацію особи, відповідального за дану зону, таймінги кешування зонної інформації й взаємодії DNS-Серверів.

2. **Записи хостів (записи A, Address Record).** Тип, що найбільше часто зустрічається, для записів ресурсів. Містить ім’я хоста й відповідну йому IP-адресу.

3. **Записи сервера імен (Name Server – NS).** Указують, які комп’ютери в базі даних DNS є серверами імен – тобто DNS -Серверами для конкретної зони. Для кожної зони може існувати тільки один запис типу SOA, але може бути трохи NS – записів.

4. **Запис AAAA (IPv6 address record)** зв’язує ім’я хоста з адресою протоколу IPv6.

5. **Запис CNAME (canonical name record)** або **канонічний запис імені** дозволяє привласнювати хосту мнемонічні імена. Мнемонічні імена, або псевдоніми, широко застосовуються для зв'язування з хостом якої-небудь функції або просто для скорочення імені.

6. **Запис MX (mail exchange)** визначає поштовий сервер-машину, яка обробляє пошту для даного домену.

7. **Запис SRV (server selection)** використовується для пошуку серверів, що забезпечують роботу тих або інших служб у даному домені.

8. **Запис PTR (pointer)** або запис указівника зв'язує IP хоста з його канонічним іменем.

Зона – це логічний вузол у дереві імен. Розрізняють такі типи зон у DNS.

Зона прямого перегляду (forward lookup zone) створюється в DNS за замовчуванням під час установки. Вона необхідна для перетворення доменного імені в IP-адресу й інформацію про ресурси. Більшість записів у прямій зоні типу A.

Зона зворотного перегляду (reverse lookup zone) реалізує зворотний DNS-Запит. Розгортання зони зворотного перегляду зазвичай поліпшує продуктивність DNS і суттєво підвищує успішність DNS-Запитів. Зона зворотного перегляду складається майже цілком із записів типу PTR (Pointer).

Первинна зона (Primary zone). В DNS (без інтегрованої Active Directory) один сервер служить первинним DNS – сервером зони, і всі зміни, виконувані в даній зоні, здійснюються на цьому конкретному сервері. Один DNS-Сервер може містити кілька зон, будучи первинним для однієї зони й вторинним для іншої. Однак, якщо зона є первинною, усі запитані зміни для даної зони повинні виконуватися на сервері, що містить основну копію зони.

Вторинна зона (Secondary zone) створюється для забезпечення резервування й розвантаження первинної зони. Однак кожна копія бази даних DNS доступна тільки для читання, тому що всі модифікації записів виконуються в первинній зоні. Один DNS-Сервер може містити кілька первинних і вторинних зон.

Зона-заглушка (Stub zone) являє собою зону, яка не містить ніякої інформації про членів домену, а служить тільки для переадресації запитів до списку призначених серверів імен для різних доменів. Тому вона містить тільки

записи NS, SOA і зв’язані записи (glue records – записи A, які використовуються в комбінації з конкретним записом NS для перетворення IP-адреси конкретного сервера імен). Сервер, що містить зону-заглушку якого-небудь простору імен, не керує зоною. Вона використовується для прискорення роботи.

Індивідуальне завдання

Завдання 1. Створіть зону прямого перегляду **PKMXXXNN.test.**

1. Відкрийте оснащення DNS.

2. Розгорніть вузол DNS, далі розгорніть вузол **<Ім’я комп’ютера>**.

3. Для створення нового домену клацніть правою кнопкою по **Зоні прямого перегляду** й виберіть пункт **Нова зона**.

4. У вікні **Тип зони** вкажіть **Основна зона** й натисніть **Next**.

5. У вікні **Ділянка реплікації зони** виберіть перемикач **На всі DNS-Сервери в лісі PKMXXXNN.test Active Directory.**

6. У вікні **Ім’я зони** вкажіть ім’я зони – **zona1.test** і натисніть **Next**.

7. Для додавання нового вузла (хоста) у створену зону клацніть правою кнопкою по вузлу **zona1.test** і виберіть **Новий хост**. У поле **Ім’я** введіть ім’я вузла – **Aserver**. Поле **IP Address** установіть рівним IP-адресі вашого комп’ютера. Натисніть **Додати вузол**.

8. Переконайтеся, що в **Зоні прямого перегляду** з’явився новий вузол **zona1.test** і сгенеровані записи **Початковий запис зони** (**SOA**), **Сервер імен** (**NS**) і **Вузол** (**A**) (рис. 4.8).

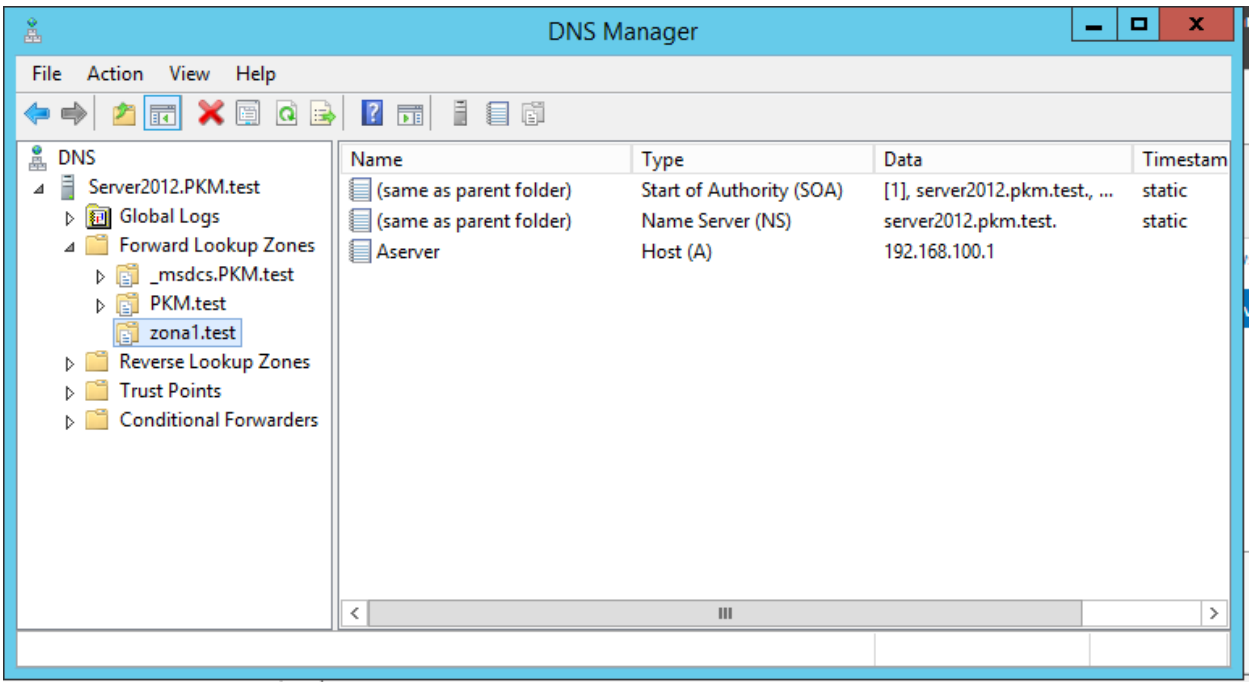

Рисунок 3.8 – Встановлення зони прямого перегляду

Завдання 2. Протестуйте роботу служби DNS.

1. Запустіть віртуальну машину з Windows. Виконайте в ній команду:

ping Aserver.zona1.test

2. Переконайтеся, що такий вузол був знайдений і відображається його IP-адреса. Якщо ping не проходить, потрібно виправити настроювання.

3. Для перетворення IP-адреси в доменне ім’я виконайте утиліту *nslookup* з параметром, рівним IP-адресі віртуальної машини. Поясніть, чому з’явилася помилка.

Завдання 3. Створіть зону зворотного перегляду (для перетворення IP-адреси в доменне ім’я).

1. На віртуальній машині з Windows Server у вузлі **Зони зворотного перегляду** клацніть правою кнопкою миші й виберіть **Майстер створення нової зони**.

2. У вікні **Тип зони** вкажіть **Основна зона** й натисніть **Next**.

3. Переконайтеся, що обраний перемикач **Номер мережі**. У поле під ним уведіть адресу вашої мережі (наприклад, 192.168.0). Поле **Ім’я зони зворотного перегляду** внизу вікна повинне виглядати так: **0.168.192.in-addr.arpa**.

4. Завершіть роботу майстра, залишивши всі настроювання за замовчуванням.

5. Клацніть правою кнопкою миші по новому вузлу в **Зоні зворотного перегляду** (наприклад, 192.168.0.x Subnet) і виберіть **Новий указівник**. Останнє число встановіть рівним останньому числу в IP-адресі. У поле **Ім'я вузла** запишіть повне ім'я вузла, наприклад **Aserver.zona1.test**.

Завдання 4. Створіть псевдонім для вузла **Aserver.zona1.test**.

Клацніть правою кнопкою миші по вузлу **zona1.test** і виберіть **Новий псевдонім**. У полі **Псевдонім** укажіть псевдонім вузла (наприклад, **Myserver1**). У полі **Повне доменне ім'я** побачите повне ім'я **Myserver1.zona1.test**

Завдання 5. Протестуйте роботу служби DNS.

Використовуйте утиліти **ping**, **nslookup**.

У дереві консолі відкрийте властивості DNS-вузла через команду контекстного меню **Властивості**. Перейдіть на вкладку **Спостереження**.

У групі **Виберіть тип тесту** позначте прапорці **Простий запит до цього DNS-Серверу** й **Рекурсивний запит до інших DNS-Серверам**. Клацніть кнопку **Тест**.

У списку **Результати тесту** проти обох записів ви побачите **ПРОЙДЕНИЙ**. Якщо ви працюєте на автономному сервері, напроти **Рекурсивний запит** ви побачите **FAIL (помилка)** (рис. 4.9).

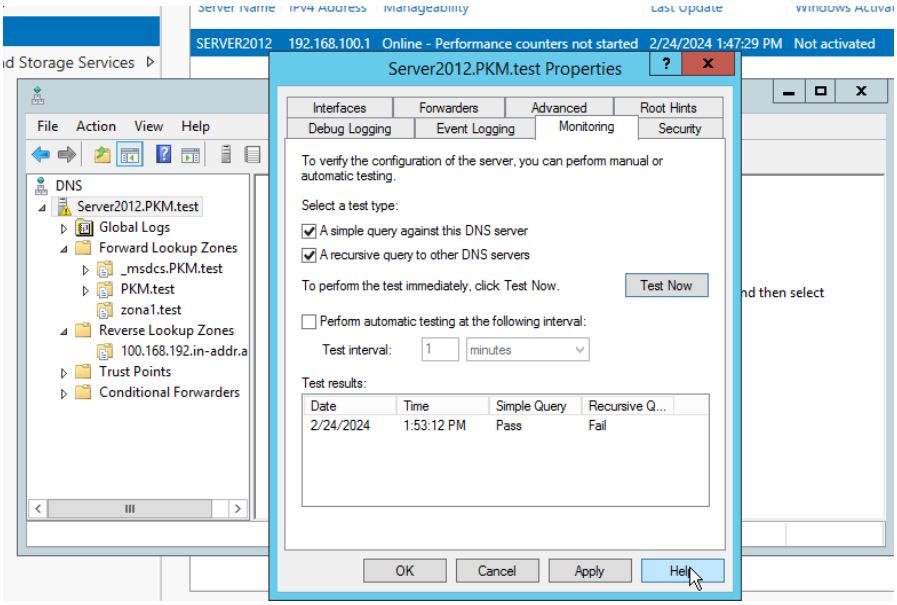

Рисунок 4.9 – Тест конфігурації DNS-серверу

Завдання 6. Включіть робочу станцію в домен і сконфігуруйте клієнта для використання служби DNS.

1. На віртуальній машині з Windows відкрийте в діалоговому вікні **Мережні підключення** властивості TCP/IP. Настройте систему для автоматичного одержання адреси DNS і вручну вкажіть IP-адресу віртуальної машини з Windows Server.

2. Виконайте команду:

ping Aserver.zona1.test

Завдання 7. Включіть робочу станцію у домен **PKMXXXNN.test**

1. Для приєднання комп'ютера до домену на робочій станції слід відкрити вікно Пуск → Мій комп'ютер → Властивості системи. Перейдіть на вкладку **Ім'я комп'ютера**. Виберіть **Ідентифікація**. Відкриється майстер мережної ідентифікації. Натисніть **Next**.

2. На вкладці **Підключення до мережі** виберіть **Комп'ютер входить у корпоративну мережу, і під час роботи я використовую його для з'єднання з іншими комп'ютерами** – **Моя організація використовує мережу з доменами**.

3. У вікні **Мережна інформація** вивчте, які мережні параметри знадобляться. У вікні **Відомості про обліковий запис і домен** залишіть усе без зміни. Натисніть **Next**.

4. У вікні **Домен комп'ютера** запишіть ім'я домена й вузла – **Ім'я комп'ютера** – win10, а **Домен комп'ютера** – PKMXXXNN. Натисніть **Next**.

5. З'явиться вікно, у якому потрібно ввести ім'я й пароль обліковому запису, який має дозвіл на додавання користувачів у домен. Наприклад, у нашому випадку це будуть:

6. **Ім'я користувача** – vboxuser (ім'я користувача у Windows Server).

7. **Пароль** – порожній (або поточний пароль для користувача).

8. **Домен** – PKMXXXNN.test

9. У вікні **Облікові записи користувачів** буде запропоновано додати нових користувачів. Виберіть перемикач **Не додавати користувачів зараз**. Натисніть **Готово** й перезавантажте комп'ютер.

10. Перевірте на віртуальній машині з Windows Server в оснащенні Active Directory → Користувачі й комп'ютери належність робочої станції домену (рис. 4.10).

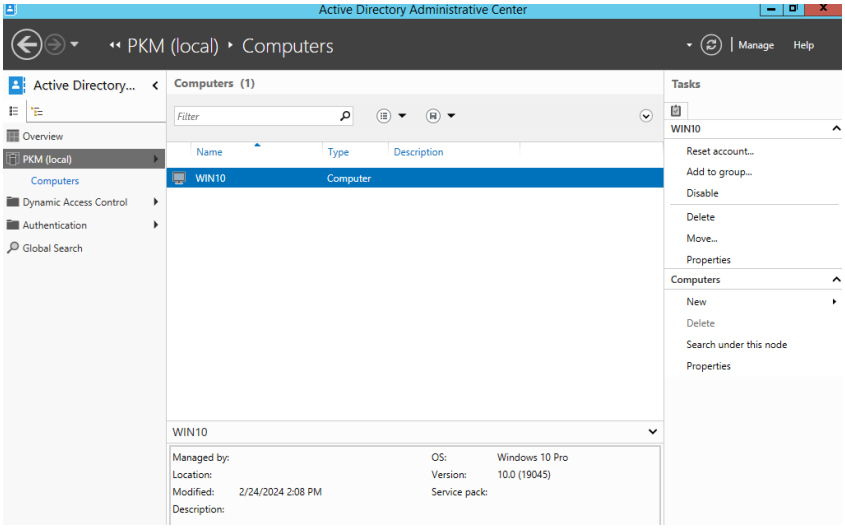

Рисунок 4.10 – Перевірка приналежності робочої станції до домену

Завдання 8. Вилучте робочу станцію з домену.

1. На робочій станції ввійдіть під обліковим записом адміністратора. Викличте Пуск → Мій комп'ютер → Властивості системи → Ім'я комп'ютера →Ідентифікація

2. На вкладці **Підключення до мережі** виберіть **Комп'ютер призначений для домашнього використання й не входить у корпоративну мережу**. Запропонуйте інший спосіб виключення робочої станції з домену й реалізуйте.

Зміст звіту

1. Тема роботи.

2. Мета виконання роботи.

3. Хід виконання Індивідуальне завдання. Для всіх завдань помістіть у звіті скріншоти, що відбивають правильність виконання завдань.

4. Висновки.

Контрольні запитання

1. Яку інформацію зберігає служба каталогів Active Directory?
2. Роз’ясніть призначення і переваги використання утиліти nslookup?
3. Навіщо об’єкти в ієрархії DNS ідентифікуються?
4. Для чого використовуються зони прямого та зворотного перегляду?
5. Обґрунтуйте необхідність створення псевдонімів для вузлів мережі.

ЛАБОРАТОРНА РОБОТА 5

АДМІНІСТРУВАННЯ ФАЙЛОВОГО СЕРВЕРА. АДМІНІСТРУВАННЯ КОРИСТУВАЧІВ І ГРУП

Мета роботи: навчитися створювати й адмініструвати облікові записи користувачів і груп; навчитися встановлювати й конфігурувати дискові квоти файлового сервера.

Індивідуальне завдання

Перед початком роботи переконайтеся, що робоча станція з Windows 10 є членом домену PKMXXXNN.

Завдання 1. Створити облікові записи користувачів.

1. Завантажте віртуальну машину з Windows Server і увійдіть під обліковим записом адміністратора.

2. Відкрийте вікно Пуск → Адміністрування → Active Directory→ Користувачі і комп'ютери. Перейдіть на вкладку **Users** (рис. 5.1).

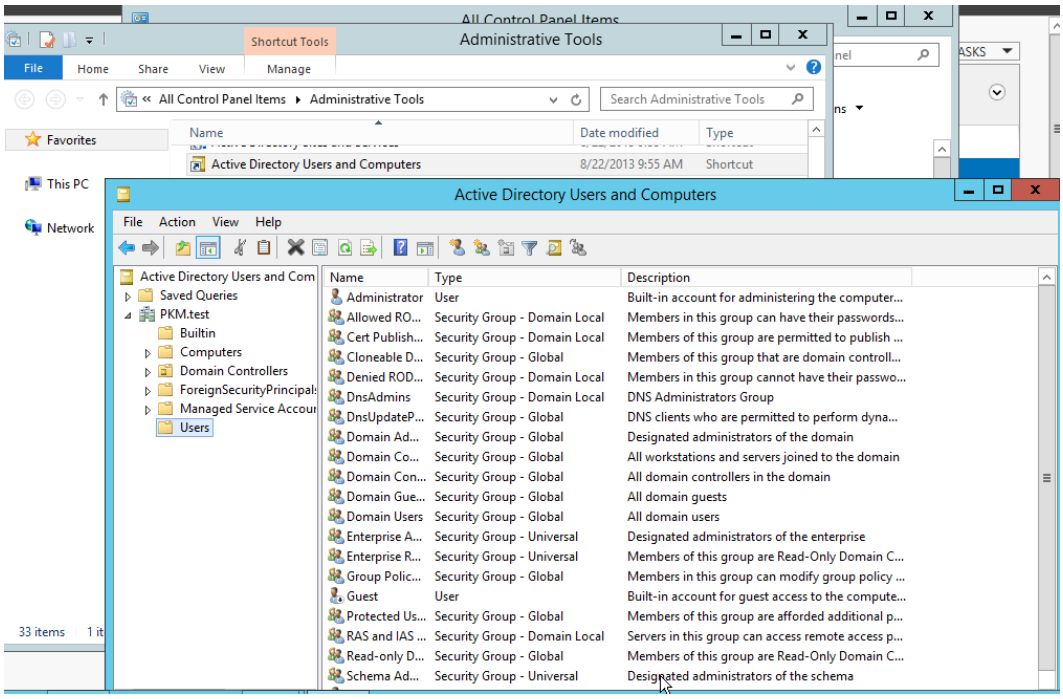

Рисунок 5.1 – Вікно створення нового облікового запису

3. У контекстному меню виберіть створення нового користувача (New→User), нехай ім’я облікового запису буде **proba**. Укажіть, що користувачеві не треба міняти пароль при першому вході й термін дії його пароля не обмежений. Які вимоги до складності пароля ставляться на вашому сервері? Для цього можна використовувати вікно Пуск → Адміністрування → Політика безпеки домену**.**

Відкрийте вікно Пуск → Адміністрування → Active Directory → користувачі й комп’ютери. Перейдіть на вкладку **Builtin.** У ній правою кнопкою миші виберіть Створити → Група. Створіть нову групу безпеки з іменем **labs**, ділянка дії якої – локальна в домені (рис. 5.2).

4. Коли обліковий запис **proba** буде створено, спробуйте ввійти під ним на сервер. Повинне з’явитися повідомлення, що діюча політика не дозволяє це зробити. Справа в тому, що настроювання за замовчуванням не дозволяють звичайним користувачам локально входити на контролер домену.

5. Завантажте віртуальну машину з Windows і ввійдіть під обліковим записом **proba**. Зрівняйте список зареєстрованих облікових записів на робочій станції з іменем облікового запису, під яким Ви ввійшли на робочу станцію. Визначити ім’я, під яким зроблений вхід на комп’ютер, можна, якщо відкрити вкладку Пуск → Виконати і ввести:

cmd /k echo %username%

або в командному рядку набрати:

set user

Зробіть скріншот отриманих вікон.

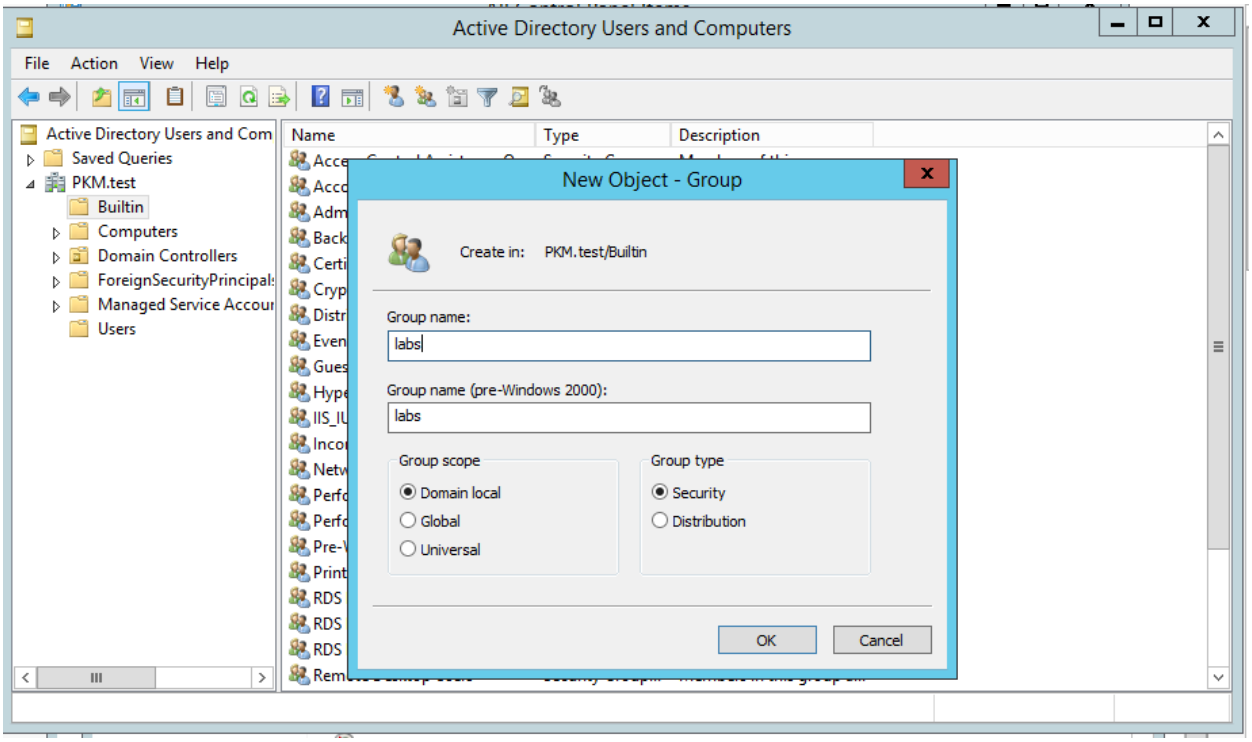

Рисунок 5.2 – Створення нової групи

6. Перевантажте віртуальну машину з Windows і ввійдіть під обліковим записом адміністратора. Відкрийте вікно Пуск → Адміністрування→ Керування комп'ютером. Перейдіть на вкладку **Users**. У контекстному меню виберіть створення нового користувача (New→User) нехай ім'я облікового запису буде **rabota**. Укажіть, що користувачеві дане право змінювати пароль і термін дії його пароля не обмежений. Користувач **rabota** буде входити у вбудовану групу **Користувач.**

Завдання 2. Встановити роль "Файл-Сервер".

1. Завантажте віртуальну машину з Windows Server і ввійдіть під обліковим записом адміністратора.

2. Відкрийте вікно Пуск → Адміністрування → Керування даним сервером і виберіть установку ролі "Файл-сервер" (рис. 5.3).

Найчастіше при адмініструванні файлового сервера виникає необхідність відстежувати, скільки дискового простору використовується кожним з користувачів і, за необхідності, обмежувати це значення. Зробити це дозволяє механізм дискових квот.

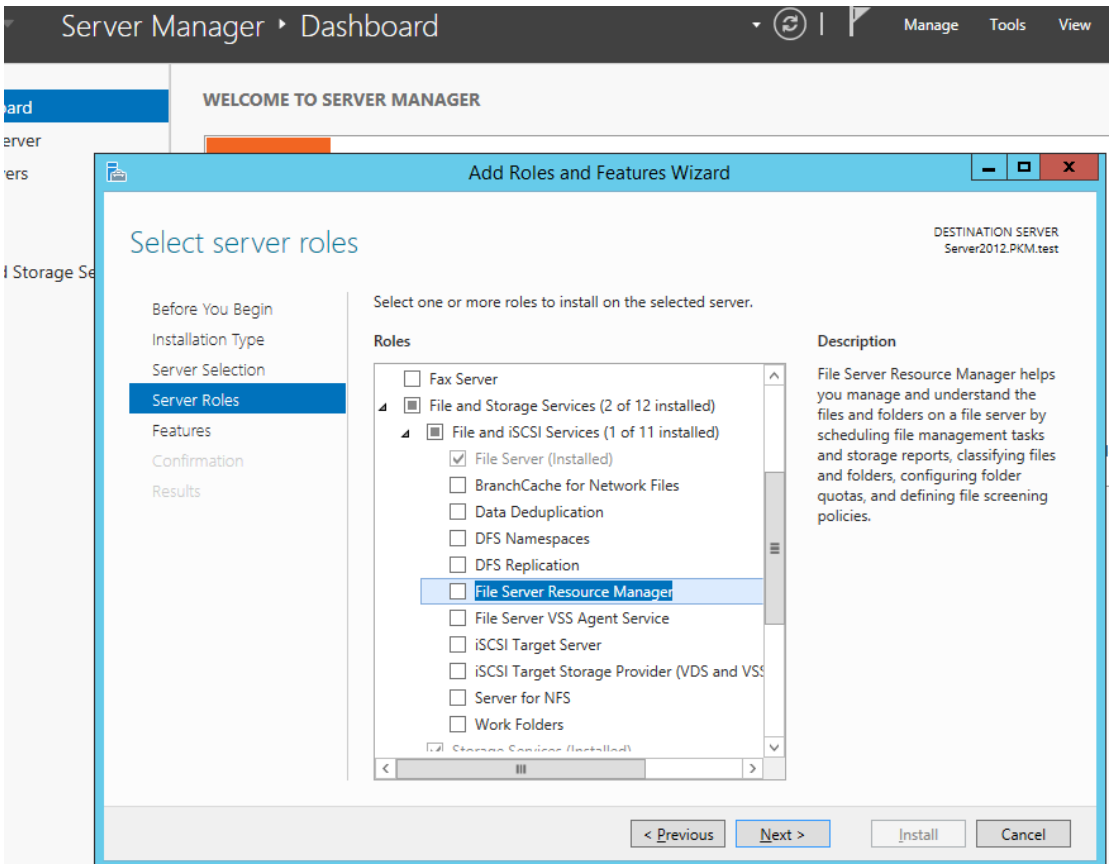

Рисунок 5.3 – Майстер встановлення нових ролей сервера

Надайте в загальний доступ папку, наприклад C:\Shared. Для цього натисніть на папці правою кнопкою миші та оберіть пункт Prorerties (рис. 5.4).

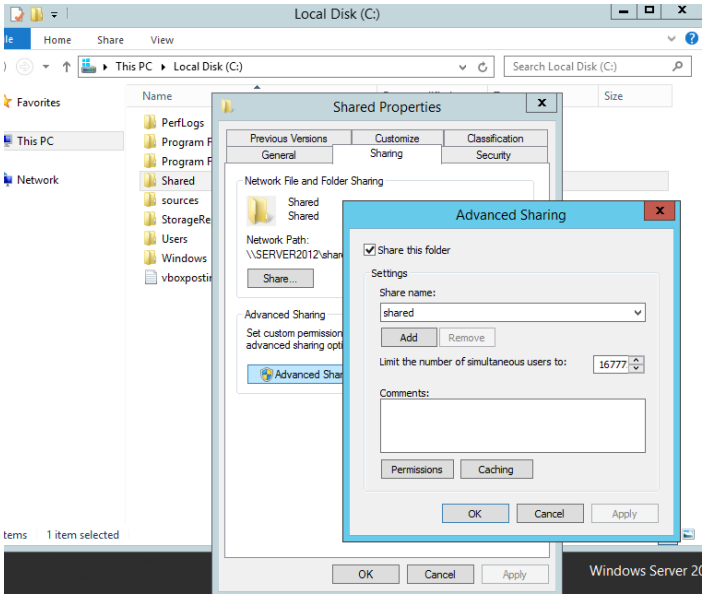

Рисунок 5.4 – Надання загального доступу на папку

Для встановлення квот перейдіть у налаштування фалових ресурсів (Server Manager-> Tools -> File Server Resource Manager рис. 5.5).

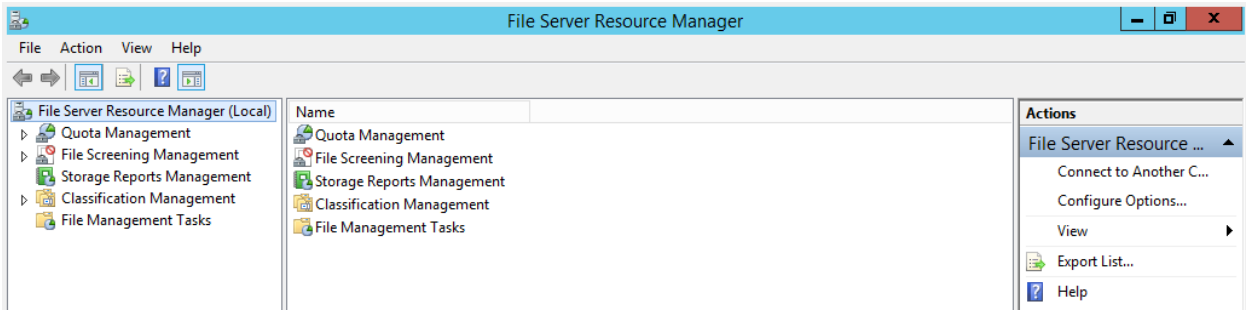

Рисунок 5.5 – Налаштування ресурсыв файлового серверу

Для встановлення дискових квот необхідно скористатися закладкою Quota Management. Встановіть яку-небудь невелику квоту (наприклад, 1 Мб) і граничне значення для попередження адміністратора: воно повинне бути не більше розміру квоти й частіше береться трохи менше (рис. 5.6).

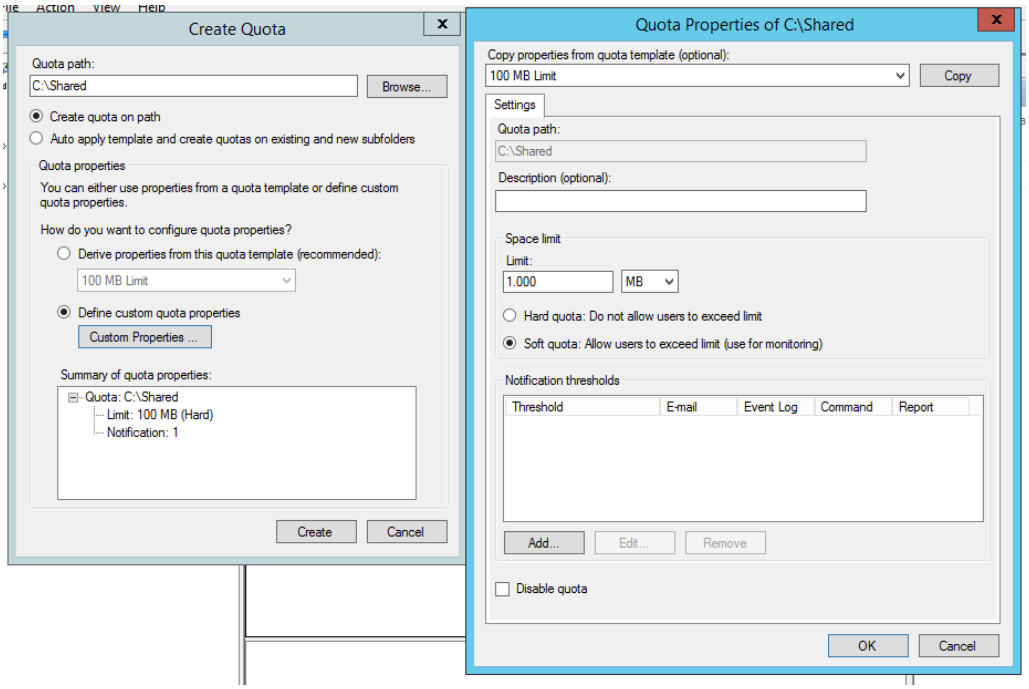

Рисунок 5.6 – Настроювання квот

Квоти бувають двох типів: м’які0 та жорсткі. М’які квоти спрацювують у разі використання користувачем певного обсягу та дозволяють інформувати

адміністратора про цю подію. Жорсткі квоти забороняють використовувати дисковий простір більшого обсягу, ніж зазначено у квоті.

3. Перевантажте віртуальну машину з Windows і ввійдіть під обліковим записом **rabota**. Відкрийте Мій комп'ютер → Мережне оточення → Уся мережа → Microsoft Windows Network**.** Спробуйте знайти сервер і загальну папку. Поясніть отримані результати.

Увійдіть у віртуальну машину з Windows 10 під обліковим записом **proba**. Підключить мережевий ресурс як окремий диск. Запишіть на цей диск кілька файлів, щоб їх сумарний розмір був більше 1 Мбайта.

4. На сервері відкрийте властивості диска **C:** і у вкладці **Квоти** натисніть кнопку **Запису квот…** Відкриється вікно, у якому відображається, що користувач **proba** перевищив ліміт, але поки йому записувати файли на диск не забороняється (рис. 5.7). Спрацьовує так звана м'яка (soft) квота, яка дозволяє проінформувати адміністратора.

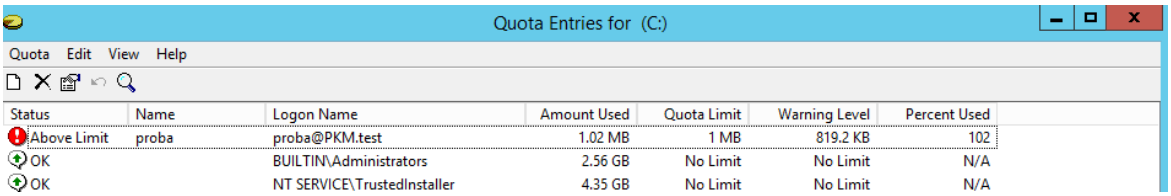

Рисунок 5.7 – Реєстрація перевищення квоти на загальний ресурс

5. Змініть й перевірте роботу настроювань квот так, щоб спрацювала "жорстка" квота й користувач не зміг перевищувати квоту.

Завдання 3. Одержання інформації про папки із загальним доступом.

1. Увійдіть у віртуальну машину з Windows Server під обліковим записом адміністратора. З командного рядка одержіть інформацію про папки в загальному доступі за допомогою команди net share. Її запуск без додаткових параметрів приведе до одержання списку всіх наявних папок (рис. 5.8.).

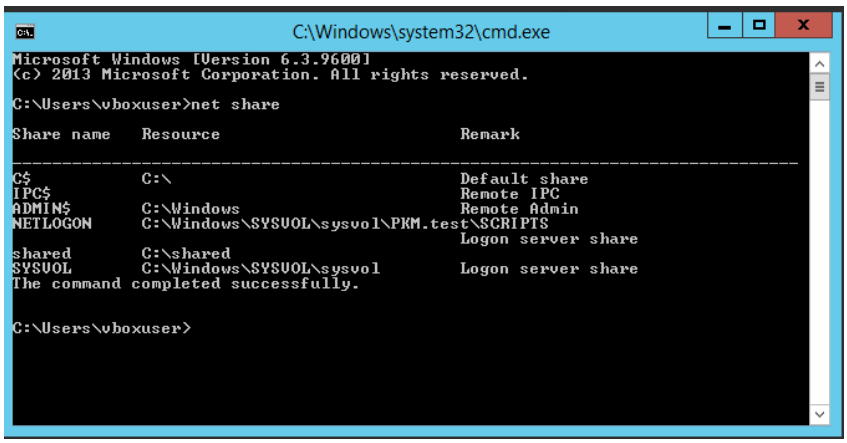

Рисунок 5.8 – Загальна інформація про папки в загальному доступі

Докладну інформацію про папку можна одержати, запустивши команду:

net share ім'я_папки

За допомогою **net share** також можна надавати загальний доступ до папки, скасовувати його, міняти параметри, тобто робити всі ті операції, які можуть виконуватися засобами із графічним інтерфейсом. Перелік параметрів запуску цієї команди можна одержати за допомогою

net share /?

Переглянути список папок, надаваних у загальний доступ вилученим комп'ютером, можна за допомогою команди

net view

2. Засобами графічного інтерфейсу надайте загальний доступ на створену Вами папку.

3. Перейдіть у віртуальну машину з Windows 10 за допомогою команди **net view**, перегляньте з неї список папок, надаваних у загальний доступ сервером **Server**. Зрівняйте отриману інформацію зі списком, який повертає команда **net share**, виконана на сервері. За довідкою знайдіть ключ команди **net view**, який дозволить одержати повний список папок.

Підключіть як мережний диск на віртуальній машині з Windows 10 папку, надавану сервером у загальний доступ. Відключіть мережний диск із командного рядка за допомогою команди

net use

Наприклад,

net use F: \\server1\data

підключить як мережний диск F: папку data з комп’ютера server1. А команда

net use F: /delete

відключіть мережний диск F:.

Зміст звіту

1. Тема роботи.
2. Мета виконання роботи.
3. Хід виконання індивідуального завдання. Для всіх завдань помістить у звіті скріншоти, що відбивають правильність виконання завдань
4. Висновки.

Контрольні запитання

1. Для чого встановлюється роль файлового сервера?
2. Які типи квот існують?
3. Для яких видів ресурсів домену можливо створити та відстежити квоту?
4. За допомогою яких утиліт командного рядка можливе адміністрування загальних ресурсів сервера?

ЛАБОРАТОРНА РОБОТА 6

ТЕХНОЛОГІЯ FRAME RELAY

Мета роботи: навчитися створювати й налаштовувати мережеве обладнання, яке працює на основі технології Frame Relay.

Індивідуальне завдання

Для виконання ЛР знадобиться пакет моделювання GNS3 (надає викладач).

Створіть топологію, що зображена на рис. 6.1.

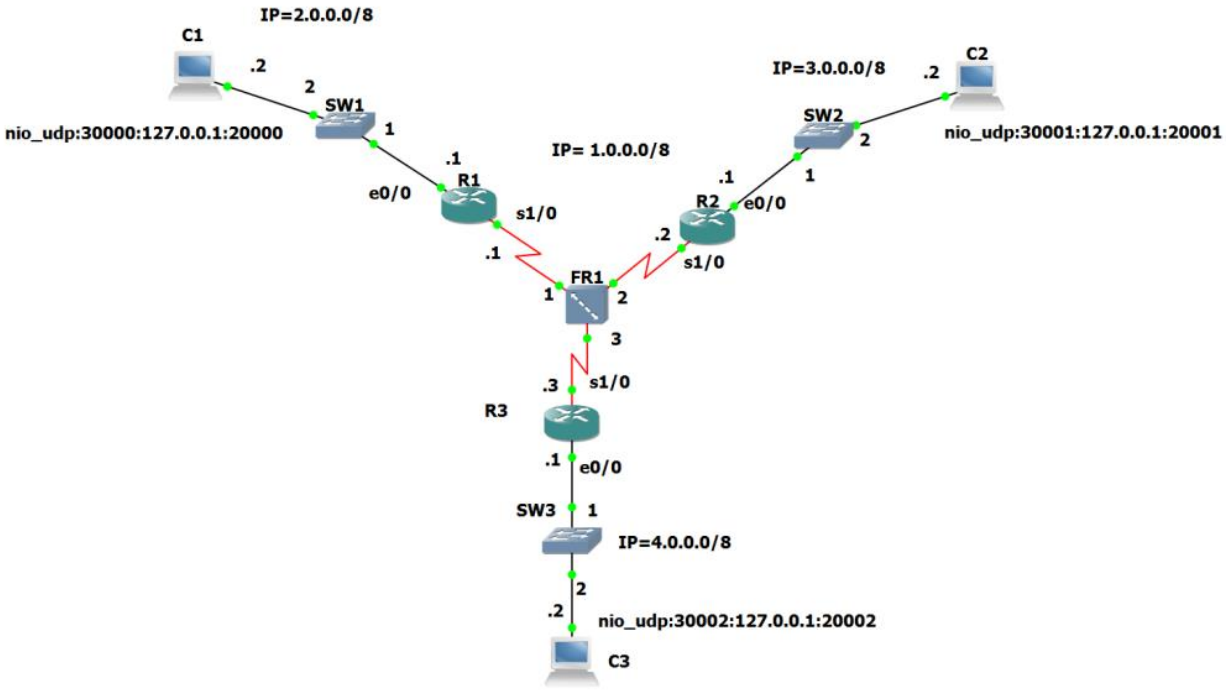

Рисунок 6.1 – Топологія для виконання практичної частини

Таблиця 1 – IP адреси інтерфейсів

R1	R2	R3	PC1	PC2	PC3
F0/0 – 2.0.0.1/8 S1/0 – 1.0.0.1/8	F0/0 – 3.0.0.1/8 S1/0 – 1.0.0.2/8	F0/0 – 4.0.0.1/8 S1/0 – 1.0.0.3/8	2.0.0.2/8	3.0.0.2/8	4.0.0.2/8

У цьому прикладі як маршрутизатори використовуються Cisco 3640 з мережевими модулями NM-4E і NM-4T, конфігурація всіх маршрутизаторів ідентична. Для налаштування мережевих модулів клацніть правою кнопкою миші по маршрутизатору і виберіть пункт «налаштувати», після чого відкриється вікно «Конфігуратор вузла» рис. 6.2.

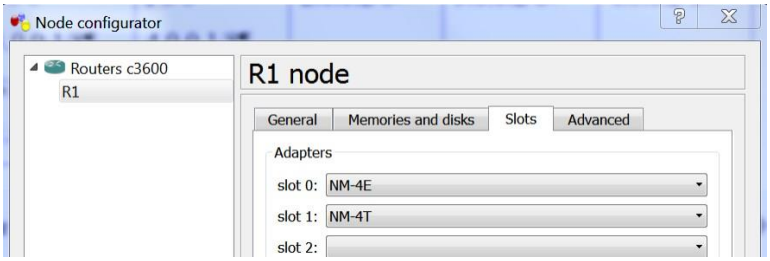

Рисунок 6.2 – Конфігуратор маршрутизатору

Для налаштування комутатора Frame Relay клацніть правою кнопкою миші по ньому, після чого виберете пункт "налаштувати", відкриється вікно «Конфігуратор вузла» (рис. 6.3).

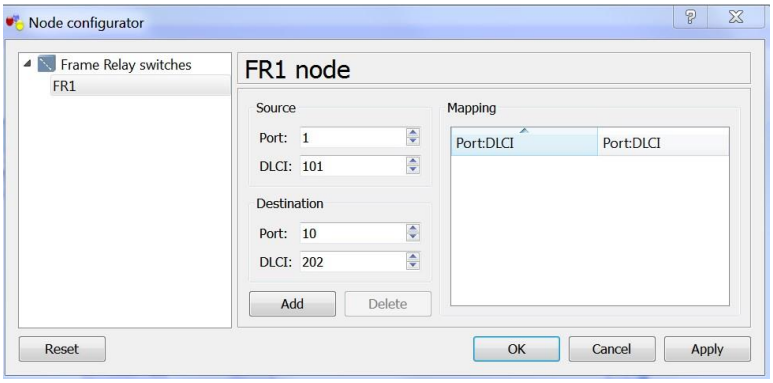

Рисунок 6.3 – Конфигуратор вузла Frame Relay

У цьому вікні потрібно скласти таблицю комутації комутатора Frame Relay. У лабораторній роботі необхідно з’єднати три маршрутизатори, перший маршрутизатор буде з’єднаний з першим портом комутатора, другий маршрутизатор з другим портом та третій маршрутизатор з третім портом. Порти на комутаторі Frame Relay з’являться, коли буде складено таблицю комутації. У цій лабораторній роботі таблиця комутації буде виглядати так:

Таблица 6.2 – Таблиця комутації

	Порт	DLCI	Порт	DLCI
1	1	102	2	201
2	1	103	3	301
3	2	203	3	302

Згідно з рядком 1 таблиці комутації пакети, що приходять на порт 1 комутатора FR с DLCI=102, будуть перенаправлятися на порт 2 із заміною ідентифікатора DLCI на 201, те саме відбувається і коли на порт 2 приходить пакет з DLCI=201 він буде перенаправлений на порт 1 із заміною DLCI на 102. Оскільки в таблиці комутації фігурують лише три порти, то ці порти будуть створені на комутаторі. Для створення першого рядка такої таблиці комутації необхідно виконати дії, показані на рис. 6.4.

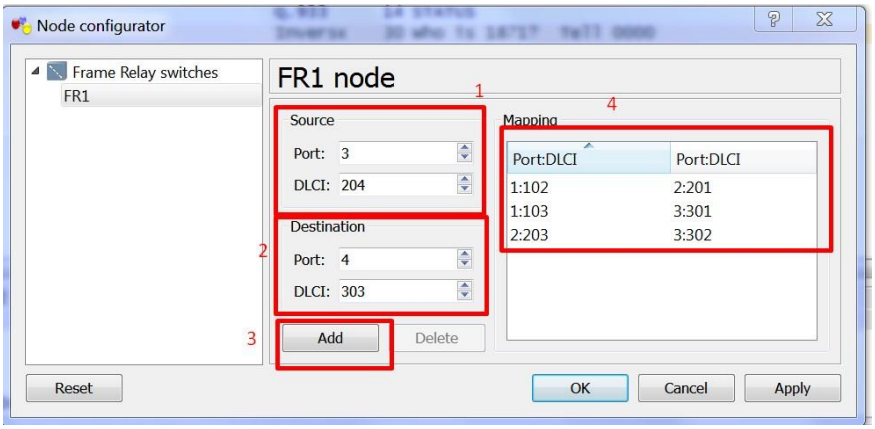

Рисунок 6.4 – Створення таблиці комутації

1. У розділі «Джерело» у полі «Порт» введіть 1, у полі «DLCI» введіть 102;

2. У розділі "Призначення" в полі "Порт" вводимо 2, у поле "DLCI" введіть 201;

3. Натискаємо кнопку «Додати».

Аналогічно додаємо наступні рядки таблиці комутації.

Для з'єднання комутаторів з маршрутизаторами необхідно натиснути кнопку «Додати Лінк»  на панелі інструментів, для зручності скористайтеся пунктом Manual.

Після чого лівою кнопкою миші натискаєте по комутатору FR, обираючи порт 1, далі лівою кнопкою миші по маршрутизатору і вибираєте один з

послідовних портів, в даній лабораторній роботі на всіх маршрутизаторах буде обраний порт s1/0. Після з'єднання всіх маршрутизаторів з комутаторам, з'єднайте комутатори Ethernet з маршрутизаторами, аналогічно, тільки як порт для з'єднання використовуєте порт e0/0 після чого з'єднайте PC з комутатором Ethernet.

Для того щоб бачити як порти пристроїв пов'язані між собою, необхідно натиснути наступну кнопку на панелі інструментів  .

Для виконання лабораторної роботи запустіть аналізатор протоколів, в даній роботі використовуватиметься WireShark, клацнувши правою кнопкою на з'єднанні між R1 та FR1 вибрати пункт «Захоплення», після чого запуститься програма WireShark (рис. 6.5).

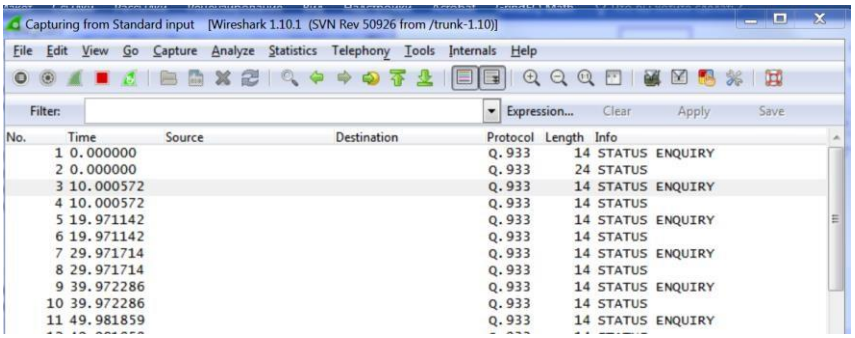

Рисунок 6.5 – Вікно WireShark

Налаштування інтерфейсів Frame Relay

Запустіть маршрутизатор, виділивши його лівою кнопкою миші та натиснувши кнопку start  , запустіть консольне вікно для введення команд, натиснувши кнопку  . Після завантаження маршрутизатора у вікні наберіть команду no і натисніть Enter, щоб відмовитися від первинного налаштування маршрутизатора. Для налаштування інтерфейсу s1/0 маршрутизатора R1 на роботу з протоколом Frame Relay введіть наступні команди:

```
R1>enable
R1#configure terminal
R1(config)#int s1/0
R1(config-if)#encapsulation frame-relay ietf
```

```
R1(config-if)#ip address 1.0.0.1 255.0.0.0
R1(config-if)#no shut
```

Як можна бачити, єдиною командою, яка змушує інтерфейс працювати в режимі Frame Relay є encapsulation frame-relay ietf, параметр ietf вказує на те, що після заголовка кадру Frame Relay повинен йти покажчик на протокол верхнього рівня згідно з рекомендацією комітету IETF RFC 2427.

Після введення останньої команди, яка включає інтерфейс (no shutd), у програмі WireShark натисніть кнопку оновити  , після чого програма виведе список повідомлень, якими обмінювалися між собою маршрутизатор і комутатор за інтерфейсом LMI (Local Management Interface – інтерфейс локального керування), як показано на рис. 6.6.

Как видно единственной командой, которая заставляет интерфейс работать в режиме Frame Relay является encapsulation frame-relay ietf, параметр ietf указывает на то, что после заголовка кадра Frame Relay должен идти указатель на протокол более верхнего уровня согласно рекомендации комитета IETF RFC 2427.

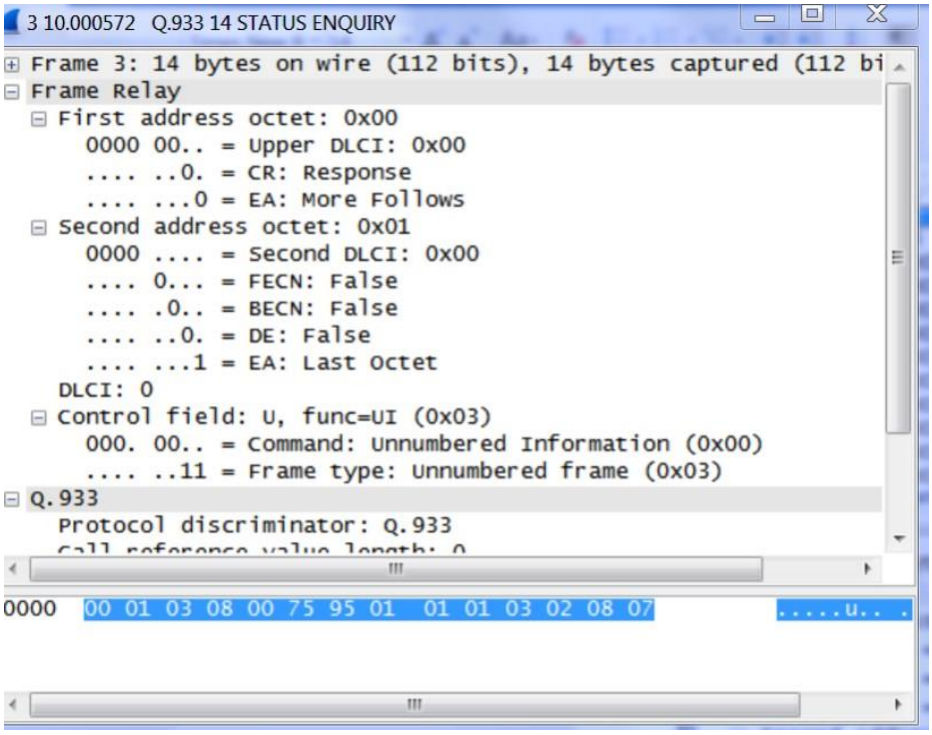

Рисунок 6.6 – Захоплення повідомлень інтерфейса LMI

Зверніть увагу на повідомлення, що періодично повторюються, Status (стан) і Status Enquiry (запит стану). При цьому маршрутизатор надсилає повідомлення Status Enquiry, а комутатор відповідає на них повідомленням Status.

Формат повідомлень Status Enquiry (Запит стану) та Status наведені на рис. 6.7 та рис. 6.8.

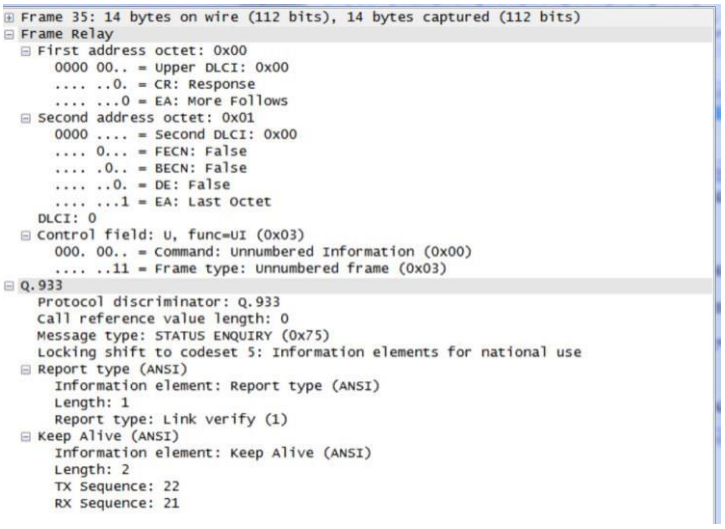

Рисунок 6.7 – Формат повідомлення Status Enquiry

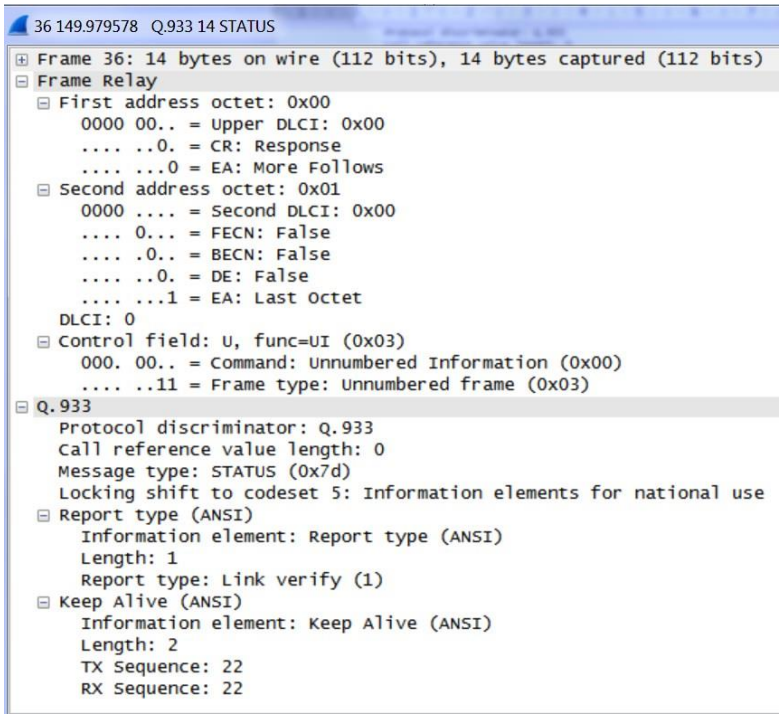

Рисунок 6.8 – Формат повідомлення Status

Ці два типи повідомлення необхідні перевірки цілісності фізичного каналу зв'язку між маршрутизатором і комутатором Frame Relay. Для відстеження цілісності з'єднання в цих повідомленнях передбачені поля переданого і приймаємо повідомлення TX і RX відповідно. Номер переданого повідомлення Status Enquiry повинен збігатися з номером прийнятого повідомлення Status, у разі маршрутизатор у повідомленні Status Enquiry вказує те що, що він передає TX=22 повідомлення Status Enquiry і до цього прийняв RX=21 повідомлення Status, потім комутатор відповідає переданим повідомленням TX=22 і те, що повідомлення під Status Enquiry з номером RX=22 було прийнято.

Після п'яти переданих повідомлень Status Enquiry маршрутизатор передає повідомлення Status Enquiry із запитом повного стану.

Повідомлення Status Enquiry із запитом повного стану наведено на рис 6.9.

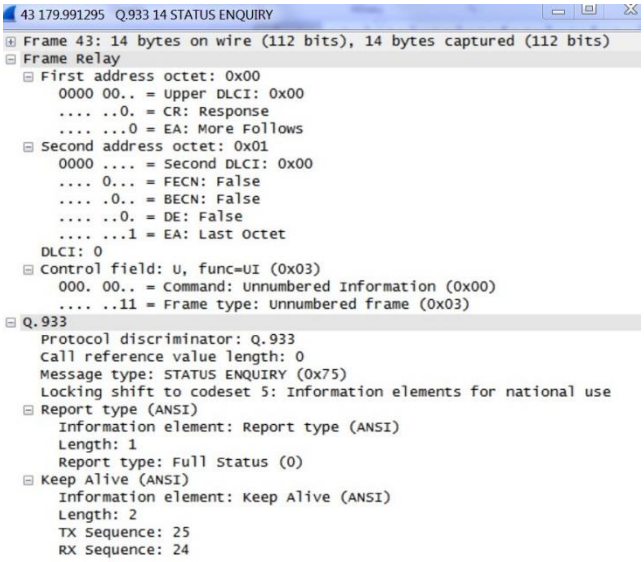

Рисунок 6.9 – Формат повідомлення Status Enquiry

У відповідь запит повного стану комутатор передає повідомлення Full Status (повний стан), у якому вказує існуючі віртуальні канали (рис. 6.10.).

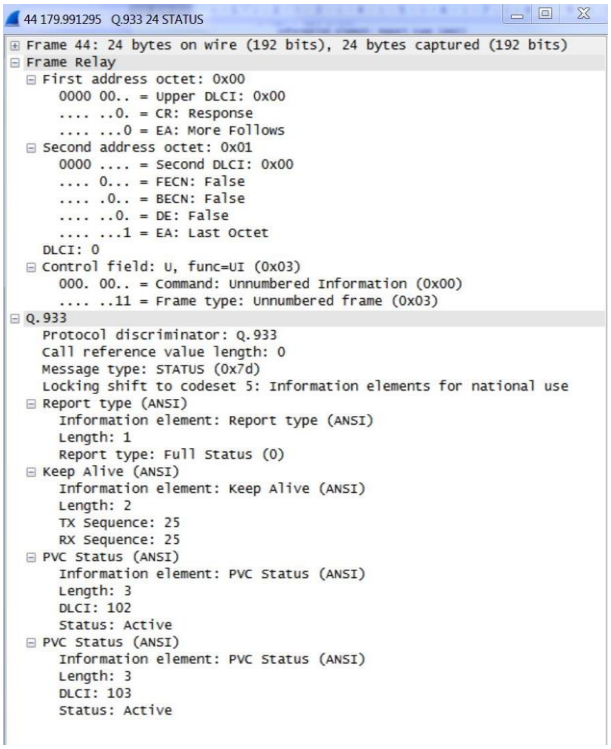

Рисунок 6.10 – Повідомлення Full Status

Конфигурування маршрутизаторів для роботи в повнозв'язній мережі Frame Relay

Крім того, що необхідно задати тип інкапсуляції на послідовних інтерфейсах маршрутизаторів, необхідно також створити маршрутні таблиці для зв'язку всіх мереж між собою, для цих цілей будемо використовувати протокол динамічної маршрутизації RIP, таким чином повна конфігурація всіх маршрутизаторів вказана нижче:

Налаштування R1

R1>enable R1#configure terminal R1(config)#int s1/0
R1(config-if)#encapsulation frame-relay ietf
R1(config-if)#ip address 1.0.0.1 255.0.0.0 R1(config-if)#no shutd
R1(config-if)#int f0/0
R1(config-if)#ip address 2.0.0.1 255.0.0.0
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#router rip
R1(config-router)#network 1.0.0.0 R1(config-router)#network 2.0.0.0

Налаштування R2

```
R2>enable
R2#configure terminal
R2(config)#int s1/0
R2(config-if)#encapsulation frame-relay ietf
 R2(config-if)#ip address 1.0.0.2 255.0.0.0
R2(config-if)#no shutd
R2(config-if)#int f0/0
R2(config-if)#ip address 3.0.0.1 255.0.0.0
R2(config-if)#no shutd
R2(config-if)#exit
R2(config)#router rip
R2(config-router)#network 1.0.0.0
R2(config-router)#network 3.0.0.0
```

Налаштування R3

```
R3>enable
R3#configure terminal
R3(config)#int s1/0
R3(config-if)#encapsulation frame-relay ietf
R3(config-if)#ip address 1.0.0.3 255.0.0.0
R3(config-if)#no shutd
R3(config-if)#int f0/0
R3(config-if)#ip address 4.0.0.1 255.0.0.0
R3(config-if)#no shutd
R3(config-if)#router rip
R3(config-router)#network 1.0.0.0
R3(config-router)#network 4.0.0.0
```

Після того, як були налаштовані всі інтерфейси на маршрутизаторах і за інтерфейсом LMI у повідомленні "повний стан" були отримані ідентифікатори DLCI активних віртуальних каналів, маршрутизатори повинні були обмінятися повідомленнями InARP для того, щоб скласти таблицю відповідності між IP-адресами маршрутизаторів і віртуальних ідентифікаторами. Використовуючи програму WireShark, перехопіть повідомлення InARP і відповідь на нього в повідомленні InARP reply.

InARP request	InARP reply
Frame Relay First address octet: 0x18 Second address octet: 0x61 DLCI: 102 Control field: U, func=UI (0x03) Padding NLPID: SNAP (0x80) Organization Code: Encapsulated Ethernet (0x000000) Type: ARP (0x0806) Address Resolution Protocol (inverse request) Hardware type: Frame Relay DLCI (15) Protocol type: IP (0x0800) Hardware size: 2 Protocol size: 4 Opcode: inverse request (8) [Is gratuitous: False] Sender hardware address: 0000 Sender IP address: 1.0.0.2 (1.0.0.2) Target hardware address: 3091 Target IP address: 0.0.0.0 (0.0.0.0)	Frame Relay First address octet: 0x18 Second address octet: 0x61 DLCI: 102 Control field: U, func=UI (0x03) Padding NLPID: SNAP (0x80) Organization Code: Encapsulated Ethernet (0x000000) Type: ARP (0x0806) Address Resolution Protocol (inverse reply) Hardware type: Frame Relay DLCI (15) Protocol type: IP (0x0800) Hardware size: 2 Protocol size: 4 Opcode: inverse reply (9) [Is gratuitous: False] Sender hardware address: 0000 Sender IP address: 1.0.0.1 (1.0.0.1) Target hardware address: 3091 Target IP address: 1.0.0.2 (1.0.0.2)

Згідно з ARP запитом, маршрутизатор R2 з'ясував, що на протилежній стороні каналу 201 знаходиться маршрутизатор з IP адресою 1.0.0.1. ідентифікатор каналу 201 зашифрований у полі Target hardware address: 3091h повідомлення InARP reply, річ у тому, що полі Target hardware address вказується не конкретний номер DLCI, а весь заголовок кадру Frame Relay. Таким чином, щоб зрозуміти ідентифікатор DLCI необхідно 3091h представити в двійковій формі 3091h=0011 0000 1001 0001. Ідентифікатор DLCI розташовується в перших шести бітах першого октету і в перших чотирьох бітах другого октету DLCI=0011001001=201.

Щоб дізнатися, яку таблицю дозволу адрес склав маршрутизатор необхідно у привілейованому режимі набрати команду show frame-relay map.

```
R1#show frame-relay map
Serial1/0 (up): ip 1.0.0.2 dlci 102(0x66,0x1860), dynamic,
        broadcast,
        IETF, status defined, active
Serial1/0 (up): ip 1.0.0.3 dlci 103(0x67,0x1870), dynamic,
        broadcast,
        IETF, status defined, active
```

Як можна бачити з виведення цієї команди маршрутизатор R1 встановив відповідності: IP 1.0.0.2 – DLCI 102; IP 1.0.0.3 – DLCI 103. Також зверніть на ключове слово broadcast, це означає, що через дані віртуальні

канали дозволена передача багатоадресатних та широкомовних повідомлень, тому передача анонсів протоколу RIPv1 можлива цими віртуальними каналами. Для перевірки правильності конфігурації використовуйте утиліту ping із PC1 (IP = 2.0.0.2) для перевірки доступності PC3 (IP = 4.0.0.2). За допомогою програми WireShark перехопіть повідомлення ICMP (рис. 6.11).

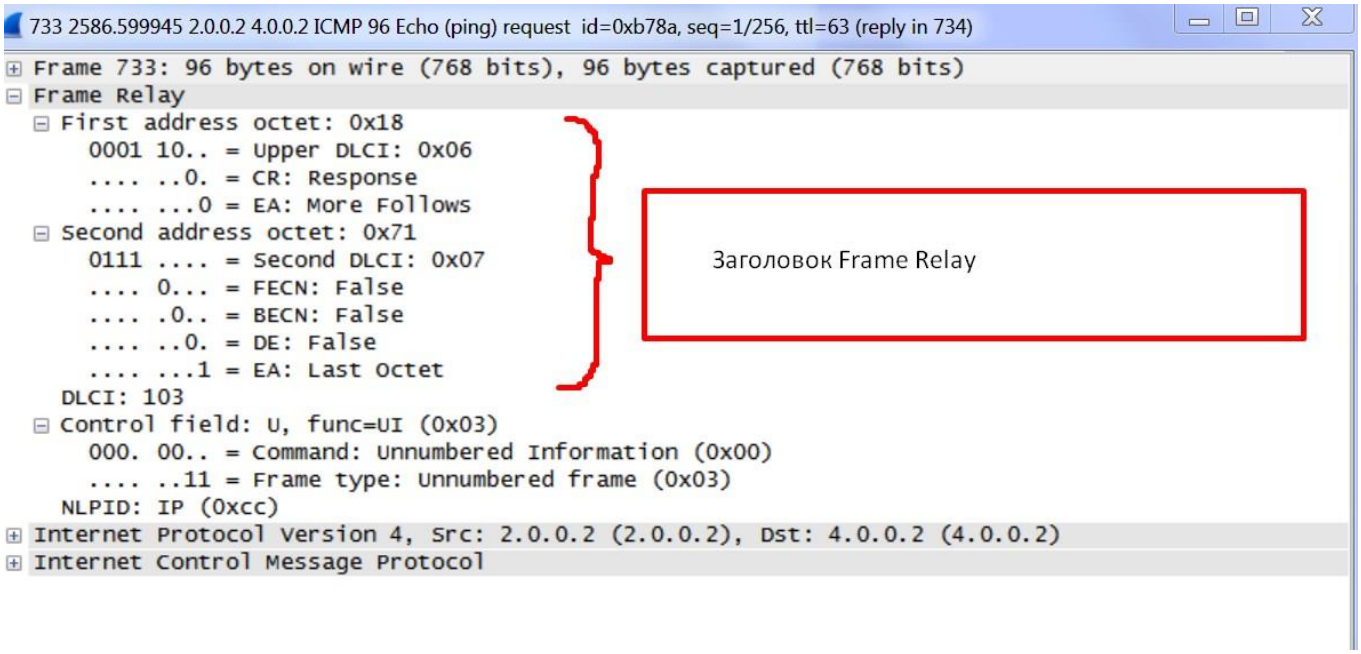

Рисунок 6.11 – повідомлення ICMP echo request

Аналіз роботи протокола RIP

Як відомо для запобігання маршрутним циклам, які можуть виникнути під час роботи протоколів динамічної маршрутизації, у маршрутизаторах передбачено механізм поділу горизонту, згідно з яким маршрутизатор не повинен анонсувати через інтерфейс ті мережі, які були отримані на цей інтерфейс. Таким чином, якщо метод поділу горизонту включений, маршрутизатор R1 повинен анонсувати через свій послідовний інтерфейс тільки мережу 2.0.0.0, але у перехопленому повідомленні протоколу RIP видно зворотне, тобто метод поділу горизонту вимкнений.

Анонс протоколу RIP:

Frame Relay
First address octet: 0x18
Second address octet: 0x61

DLCI: 102
Control field: U, func=UI (0x03)
NLPID: IP (0xcc)
Internet Protocol Version 4, Src: 1.0.0.1 (1.0.0.1), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: router (520), Dst Port: router (520)
Routing Information Protocol
Command: Response (2)
Version: RIPv1 (1)
IP Address: 1.0.0.0, Metric: 1
IP Address: 2.0.0.0, Metric: 1
IP Address: 3.0.0.0, Metric: 1
IP Address: 4.0.0.0, Metric: 1

Для того щоб включити метод поділу горизонту необхідно в режимі конфігурації кожного інтерфейса ввести команду ip split-horizon.

```
R1(config)#int s1/0
R1(config-if)#ip split-horizon

R2(config)#int s1/0
R2(config-if)#ip split-horizon

R3(config)#int s1/0
R3(config-if)#ip split-horizon
```

В результаті захват анонса RIP того самого маршрутизатора R1 буде зовсім інший:

Frame Relay
First address octet:
0x18 Second
address octet: 0x61
DLCI: 102
Control field: U, func=UI
(0x03) NLPID: IP (0xcc)
Internet Protocol Version 4, Src: 1.0.0.1 (1.0.0.1), Dst: 255.255.255.255
(255.255.255.255) User Datagram Protocol, Src Port: router (520), Dst Port: router
(520)
Routing Information
Protocol Command:
Response (2)
Version: RIPv1 (1)
IP Address: 2.0.0.0, Metric: 1

Дослідження роботи протокола RIP в топології точка - багато точек та однією підмережею.

У досліджуваної топології на рис. 6.1 на канальному рівні була налаштована повнозв'язна топологія, тобто це означає, що кожен маршрутизатор має віртуальний канал до кожного іншого маршрутизатора. Але такі топології не завжди є виправданими. Часто зустрічаються і такі, коли тільки один маршрутизатор пов'язаний з усіма рештою маршрутизаторів віртуальними каналами (Hub and Spoke топологія). Нехай тепер немає віртуального каналу від маршрутизатора 2 до маршрутизатора 3, таким чином, таблиця комутації комутатора Frame Relay буде виглядати наступним чином:

Таблиця 6.3 – Таблица комутації

	Порт	DLCI	Порт	DLCI
1	1	102	2	201
2	1	103	3	301

Для видалення запису з таблиці, видалить комутатор Frame Relay та створіть таблицю комутації згідно таблиці 3.

Командою show frame-relay map переконайтеся, що маршрутизатори R2 та R3 мають тілько по одному активному виртуальному каналу, а маршрутизатор R1 два виртуальних канала.

```
R1#show frame-relay map
Serial1/0 (up): ip 1.0.0.2 dlci 102(0x66,0x1860),
        dynamic, broadcast,
        IETF, status defined, active
Serial1/0 (up): ip 1.0.0.3 dlci 103(0x67,0x1870),
        dynamic, broadcast,
        IETF, status defined, active
R2#show frame-relay map
Serial1/0 (up): ip 1.0.0.1 dlci 201(0xC9,0x3090),
        dynamic, broadcast,
        IETF, status defined, active
R3#show frame-relay map
Serial1/0 (up): ip 1.0.0.1 dlci 301(0x12D,0x48D0),
        dynamic, broadcast,
        IETF, status defined, active
```

Оскільки метод поділу горизонту включений, то тепер маршрутизатор R2 не отримає анонс про мережу 4.0.0.0, а маршрутизатор R3 не отримає анонс про мережу 3.0.0.0, оскільки згідно з правилом поділу горизонту маршрутизатор R1 не анонсуватиме їх через свій інтерфейс, а віртуального каналу між R2 та R3 не існує. За допомогою команди show ip route перевірте таблиці маршрутизації на кожному маршрутизаторі. Для появи маршрутних записів у таблицях R3 та R2 про мережі 3.0.0.0 та 4.0.0.0 відповідно, необхідно відключити метод поділу горизонту на маршрутизаторі R1, тоді отримані ним анонси про мережі 3.0.0.0 та 4.0.0.0 на інтерфейс s1/0 будуть проанонсовані через нього маршрутизаторам R3 та R2. Щоб вимкнути розділення діапазону, введіть у режимі конфігурації інтерфейсу команду no ip split-horizon.

```
R1(config)#int s1/0
R1(config-if)#no ip split-horizon
```

Після чого знову перевірте таблиці маршрутизації маршрутизаторов R2 та R3.

Зміна методів інкапсуляції та створення підінтерфейсів

Для правильного функціонування мережі кожен віртуальний канал має бути налаштований на свій метод інкапсуляції, під інкапсуляцією розуміється механізм вказівки на протокол більш верхнього рівня, який знаходиться в кадрі LAPF. За замовчуванням маршрутизатор Cisc використовуют фірмови мето інкапсуляції, але також існують стандартизовані методи інкапсуляції, наприклад RFC 2427. Для завдання того чи іншого методу інкапсуляції існує кілька способів, що залежать від режиму роботи інтерфейсу.

Інтерфейс може працювати у двох режимах point-to-point (точка-точка) та multipoint (точка-множина точок). Режим point-to-point передбачає, що з інтерфейсом буде пов’язаний лише один віртуальний канал, у режимі multipoint з інтерфейсом може бути пов’язано більше одного віртуального каналу. За замовчуванням усі інтерфейси працюють у режимі multipoint.

Для завдання методу інкапсуляції у режимі point-to-multipoint існує два варіанти:

1. Вказати один вид інкапсуляції для всіх віртуальни каналів використовуючи команду encapsulation frame-relay [cisco | ietf]. Ця команда вводиться у режимі конфігурації інтерфейсу.

2. Вказати для кожного віртуального каналу свій метод інкапсуляції, використовуючи команду frame-relay map ip [ip адресу] [DLCI] [cisco | ietf], ця команда одночасно використовується також для статичного перетворення ідентифікатора DLCI у IP адресу.

Для завдання методу інкапсуляції в режимі point-to-point існує два варіанти:

1. Вказати для кожного віртуального каналу свій метод інкапсуляції, використовуючи команду frame-relay interface-dlci [DLCI] [cisco | ietf]

2. Вказати для кожного віртуального каналу свій метод інкапсуляції, використовуючи команду frame-relay map ip [ip адресу] [DLCI] [cisco | ietf].

Конфігурування маршрутизаторів для роботи в мережі Frame Relay з неповнозв'язною топологією та повнозв'язними ділянками

Зберіть наступну топологію в GNS3 (рис 6.7).

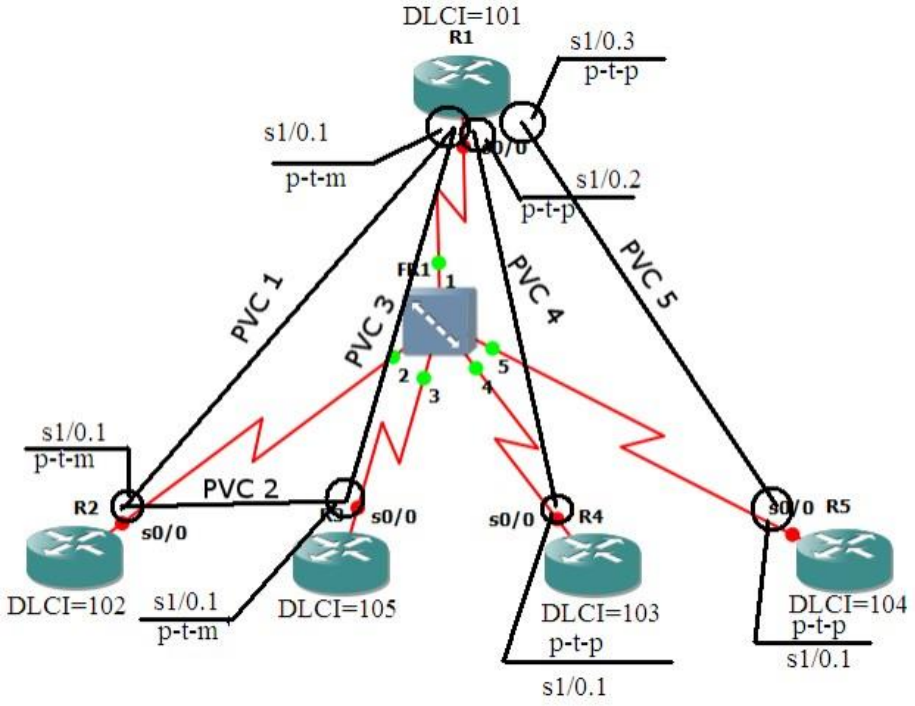

Рисунок 6.7 –Неповнозв'язна топологія з повнозв'язними ділянками

Таблиця 6.4 – IP-адреси інтерфейсів

R1	R2	R4	R7	R9
s0/1.1 – 1.0.0.1 s0/1.2 – 2.0.0.1 s0/1.3 – 3.0.0.1	s0/1 – 1.0.0.2	s0/1 – 1.0.0.3	s0/1.1 – 2.0.0.2	s0/1.1 – 3.0.0.2

Конфигурація маршрутизатора R1

```
R1(config)#int s0/0
R1(config-if)#encapsulation frame-relay ietf
R1(config-if)#no shutd
R1(config)#int s0/0.1 multipoint
R1(config-subif)#ip address 1.0.0.1 255.0.0.0
R1(config-subif)#frame-relay interface-dlci 102
R1(config-fr-dlci)#exit
R1(config-subif)#frame-relay interface-dlci 103
R1(config-fr-dlci)#exit
R1(config)#int s0/0.2 point-to-point
R1(config-subif)#ip address 2.0.0.1 255.0.0.0
R1(config-subif)#frame-relay interface-dlci 104 ietf
R1(config-fr-dlci)#exit
R1(config-subif)#exit
R1(config)#int s0/0.3 point-to-point
R1(config-subif)#ip address 3.0.0.1 255.0.0.0
R1(config-subif)#frame-relay interface-dlci 105 ietf
R1(config-fr-dlci)#exit
```

Конфигурація маршрутизатора R2

```
R2#config t
R2(config)#int s0/0
R2(config-if)#encap
R2(config-if)#encapsulation fram
R2(config-if)#encapsulation frame-relay ietf
R2(config-if)#no shutd
R2(config-if)#int s0/0.1 multipoint
R2(config-subif)#ip address 1.0.0.2 255.0.0.0
R2(config-subif)#frame-relay interface-dlci 101
R2(config-fr-dlci)#exit
R2(config-subif)#frame-relay interface-dlci 103
```

```
R2(config-fr-dlci)#exit
R2(config-subif)#exit
```

Конфигурація маршрутизатора R3

```
R3#config t
R3(config)#int s0/0
R3(config-if)#encapsulation frame-relay ietf
R3(config-if)#no shutd
R3(config-if)#int s0/0
R3(config)#int s0/0.1 multipoint
R3(config-subif)#ip address 1.0.0.3 255.0.0.0
R3(config-subif)#frame-relay interface-dlci 101
R3(config-fr-dlci)#exit
R3(config-subif)#frame-relay interface-dlci 102
R3(config-fr-dlci)#exit
R3(config-subif)#exit
R3(config)#exit
```

Конфигурація маршрутизатора R4

```
R4#config t
R4(config)#int s0/0
R4(config-if)#encapsulation frame-relay
R4(config-if)#no shutd
R4(config)#int s0/0.1 point-to-point
R4(config-subif)#ip address 2.0.0.2 255.0.0.0
R4(config-subif)#frame-relay interface-dlci 101 ietf
R4(config-fr-dlci)#exit
R4(config-subif)#exit
R4(config)#exit
```

Конфигурація маршрутизатора R5

```
R5#config t
R5(config)#int s0/0
R5(config-if)#encapsulation frame-relay
R5(config-if)#no shutd
R5(config)#int s0/0.1 point-to-point
R5(config-subif)#frame-relay interface-dlci 101 ietf
R5(config-fr-dlci)#exit
R5(config-subif)#ip address 3.0.0.2 255.0.0.0
R5(config-subif)#exit
R5(config)#exit
```

Після конфігурування усех маршрутизаторів на маршрутизаторі R1 введіть команду show frame-relay map.

```
R1#show frame-relay map
        Serial0/0.1 (up): ip 1.0.0.2 dlci
102(0x66,0x1860), dynamic, broadcast,
        IETF, status defined, active
        Serial0/0.1 (up): ip 1.0.0.3 dlci
103(0x67,0x1870), dynamic, broadcast,
        IETF, status defined, active
      Serial0/0.3 (up): point-to-point dlci, dlci
105(0x69,0x1890), broadcast, IETF status defined, active
      Serial0/0.2 (up): point-to-point dlci, dlci
104(0x68,0x1880), broadcast, IETF status defined, active
```

Можна побачити, що маршрутизатору відомо про всі ідентифікатори DLCI та їх зв'язок з кожним підінтерфейсом та IP-адресою. Зверніть увагу на ключове слово broadcast для кожного віртуального каналу, це означає, що через кожен віртуальний канал дозволено передавати багатоадресатний і широкомовний трафік.

Також з виведення даної команди видно, що зо допомогою протоколу InARP було отримано дозвіл DLCI в IP-адреси для мережі 1.0.0.0/8. Але трапляються випадки, коли протокол InARP може бути відключений. В цьому випадку необхідно створювати статичні записи дозволу адрес.

Конфігурування маршрутизаторів без підтримки протокола inARP

Вимкніть протокол InARP на маршрутизаторах R1, R2, R3, використовуючи команду no frame-relay inverse-arp в режимі конфигурації інтерфейса.

```
R1(config-if)#no frame-relay inverse-arp
R2(config-if)#no frame-relay inverse-arp
R3(config-if)#no frame-relay inverse-arp
```

А для того, щоб динамічні записи про дозволи адрес швидше зникли з таблиці, вимкніть і знову вмікніть інтерфейс. Після того, як протокол InARP буде

вимкнений зв’язок з маршрутизаторами, що знаходяться в мережі 1.0.0.0/8, буде втрачено. Для відновлення зв’язку можна налаштувати статичні записи дозволу адрес, використовуючи наступну команду в режимі конфігурації відповідного підінтерфейсу/інтерфейсу:

```
frame-relay map ip [ip адрес] [DLCI] ietf broadcast
```

Параметр ietf та broadcast є опціями до основної команди frame-relay map ip [ip адрес] [DLCI]. Використовуючи цю команду, створіть статичні дозволи адрес на маршрутизаторах R1, R2, R3.

Конфигурація маршрутизатора R1

```
R1(config)#int s0/0.1 multipoint
R1(config-subif)#frame-relay map ip 1.0.0.2 102 ietf broadcast
R1(config-subif)#frame-relay map ip 1.0.0.3 103 ietf broadcast
```

Конфигурація маршрутизатора R2

```
R2(config-if)#int s0/0.1 multipoint
R2(config-subif)#frame-relay map ip 1.0.0.1 101 ietf broadcast
R2(config-subif)#frame-relay map ip 1.0.0.3 103 ietf broadcast
```

Конфигурація маршрутизатора R3

```
R3(config)#int s0/0.1 multipoint
R3(config-subif)#frame-relay map ip 1.0.0.1 101 ietf broadcast
R3(config-subif)#frame-relay map ip 1.0.0.2 102 ietf broadcast
```

Теперь, після введення команди show frame-relay map, записи будуть мати статус static.

```
R1#show frame-relay map
Serial0/0.1 (up): ip 1.0.0.2 dlci 102(0x66,0x1860), static,
 broadcast,
 IETF, status defined, active
Serial0/0.1 (up): ip 1.0.0.3 dlci 103(0x67,0x1870), static,
 broadcast,
 IETF, status defined, active
Serial0/0.3 (up): point-to-point dlci, dlci 105(0x69,0x1890), broadcast, IETF
 status defined, active
Serial0/0.2 (up): point-to-point dlci, dlci 104(0x68,0x1880), broadcast, IETF
```

status defined, active

Зміст звіту

1. Тема роботи.

2. Мета виконання роботи.

2. Обидві топології рис. 6.1 та 6.7;

3. Лістинг налаштування маршрутизаторів для кожної топології, для топології на рис. 6.7 має використовуватися статичне перетворення адрес;

4. Захоплення повідомлень протоколу LMI: запит стану, стан, запит повного стану, повний стан – з описом призначення кожного повідомлення;

5. Захоплення кадру Frame Relay із розшифровкою полів заголовка;

6. Захоплення повідомлень протоколу inARP;

7. Пояснення процедури передачі пакета від хоста PC1 до хоста PC2.

8. Висновки.

Контрольні запитання

1. Структура кадру Frame Relay; 2. Призначення інтерфейсу LMI;

3. Алгоритм роботи протоколу inARP;

4. Поясніть, у яких випадках необхідно відключати, а в яких включати метод поділу горизонту в мережі Frame Relay;

5. Поясніть різницю в алгоритмах роботи інтерфейсів point-to-point та multipoint; 6. Структура таблиці комутації комутатора Frame Relay;

7. Викладіть зміст RFC 2427.

8. Поясніть призначення ключового слова broadcast у команді frame-relay map ip [ip адресу] [DLCI] ietf broadcast.

Лабораторна робота 7

ТЕХНОЛОГІЯ MPLS L2VPN

Мета роботи: Ознайомлення з принципами побудови VPN рівня 2 на основі MPLS-мереж. Набуття практичних навичок налаштування параметрів VPN рівня 2 для MPLS-мереж на мережному обладнанні.

Теоретична частина

Протокол MPLS дозволяє мережному провайдеру робити своїм замовникам послуги віртуальних приватних мереж VPN. Розглянемо (рис. 7.1) провайдера мережних послуг, що з'єднує 3 выддалених сайти A1, A2 і A3 замовника A и 3 віддалених сайти B1, B2 і B3 замовника B, використовуючи свою мережу MPLS. Прийнято виділяти машрутизаторы провайдера, прикордонні із сайтами замовників і позначати їх з використанням символів PE (provider edge – границя провайдеру). На рис. 7.1 бачимо три граничних машрутизатора PE1, PE2 і PE3.

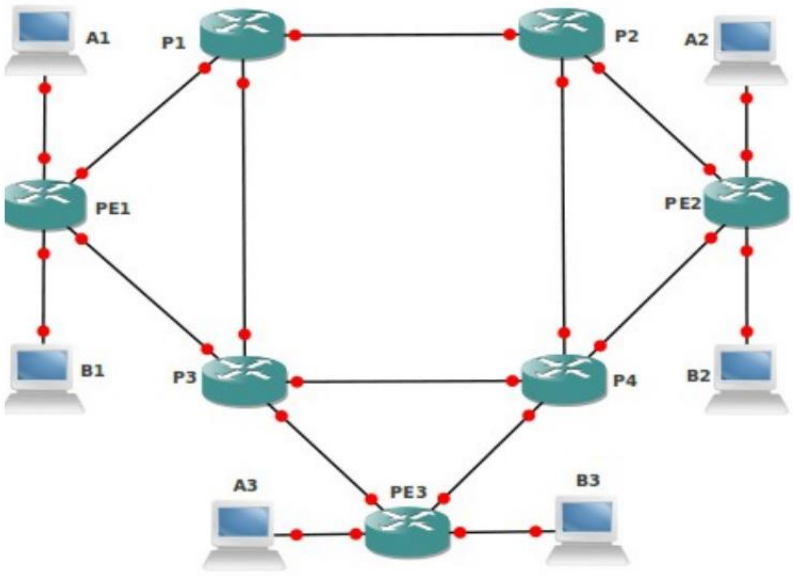

Рисунок 7.1 – Мережа провайдера мережних послуг

VPN організуються на другому та третьму рівні моделі OSI. Розглянемо оргагнізацію VPN рівня 2 за допомогою протоколів VPLS.

Організація MPLS VPN рівня 2 за допомогою VPLS

VPLS – це служба віртуальних приватних локальних мереж (Virtual Private LAN Service). Служить для забезпечення широкомовної функціональності Ethernet-мереж поверх мереж MPLS. Дозволяє об'єднати територіально віддалені локальні мережі в один домен широкомовлення через віртуальні псевдокабелі Ethernet. Таке об'єднання може бути зроблено й за допомогою EoIP-тунелів. VPLS, на відміну від EoIP:

• не вимагає інкапсуляції кадрів Ethernet в IP-пакети й уникає накладних витрат, пов'язаних з обробкою IP-заголовків;

• має більшу гнучкість і функціональність та підтримується більшим числом виробників мережного обладнання;

• є більш ефективним рішенням для створення VPN рівня 2.

У якості віртуального псевдокабелю Ethernet виступає VPLS-тунель. Для організації тунелю використовуються дві мітки: тунельна і транспортна.

Переговори про організацію VPLS-тунелю здійснюються або з використанням протоколу LDP або протоколу BGP.

Практична частина

Розглянемо настроювання VPLS VPN на прикладі топології на рис. 7.2. Замовники A і B вимагають прозоре Ethernet-з'єднання між сайтами й хочуть самі призначати IP-адреси. З рис. 7.2 бачимо, що замовники вибрали однакові адреси 172.16.1.1 – 172.16.1.3 для своїх філій A1, A2, A3 і B1, B2, B3.

Зберіть топологію в GNS3. Призначте адреси згідно з рис. 7.2. На кожному маршрутизаторі провайдеру створіть міст із ім'ям lobridge і призначте на нього адресу 9.9.9.*/32, згідно рис. 7.2.

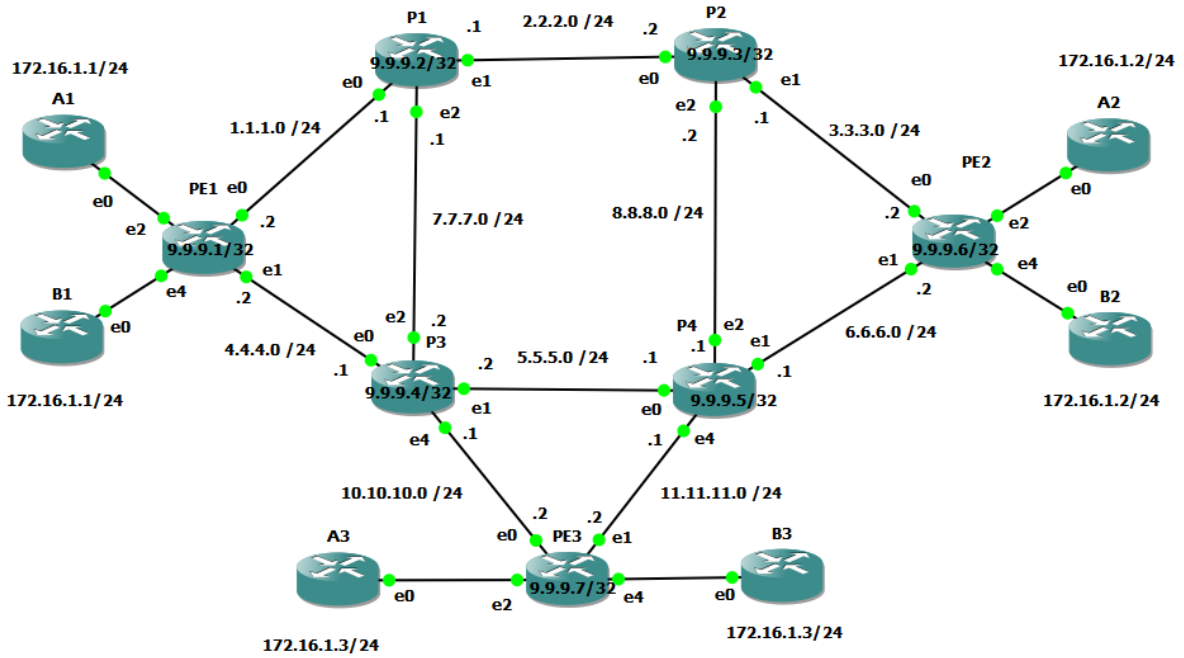

Рисунок 7.2 – Топологія для виконання практичної частини (MPLS-мережа)

На маршрутизаторах PE1, PE2 и PE3, створіть по два моста з ім’ям A та B. Помістіть в ці мости Ethernet-інтерфейс, що йде в сторону замовника. Наприклад, на PE1

```
[admin@PE1]>interface bridge add name=A
[admin@PE1]>interface bridge port add bridge=A interface=ether3
[admin@PE1]>interface bridge add name=B
[admin@PE1]>interface bridge port add bridge=B interface=ether5
```

Усередині хмари MPLS шляхи комутації пакетів по мітці (LSP-label switching path) організовуються або за допомогою протоколу LDP або за допомогою протоколу RSVP. Маємо чотири способи організації VPLS VPN:

№ з/п	Організація VPLS-тунелю	Організація LSP
1	LDP	LDP
2	LDP	RSVP
3	BGP	LDP
4	BGP	RSVP

1. Налаштування LDP VPLS

Розглянемо організацію VPLS-тунелю за допомогою протоколу LDP. LSP організовуються або за допомогою протоколу LDP або за допомогою протоколу RSVP.

1.1. LDP VPLS з організацією LSP за допомогою LDP

Для розподілу міток і організації шляхів комутації пакетів по мітці активуємо на кожному LSR-маршрутизаторі протокол LDP. Транспортну адресу встановимо як адресу інтерфейсу Loopback з ім'ям lobridge. Це змушує маршрутизатор рекламувати сусідам по LDP цю адреса як транспортну адресу.

Оголосимо інтерфейси, що дивляться усередину MPLS-мережі, як інтерфейси, що беруть участь в обміні міток. Наприклад для PE1 це виконується командами:

```
[admin@PE1]>mpls ldp set enabled=yes transport-address=9.9.9.1 lsr-id=9.9.9.1
[admin@PE1]>mpls ldp interface add interface=ether1
[admin@PE1]>mpls ldp interface add interface=ether2
```

Для P4 – командами:

```
[admin@P4]>mpls ldp set enabled=yes transport-address=9.9.9.5 lsr-id=9.9.9.5
[admin@P4]>mpls ldp interface add interface=ether1
[admin@P4]>mpls ldp interface add interface=ether2
[admin@P4]>mpls ldp interface add interface=ether3
[admin@P4]>mpls ldp interface add interface=ether5
```

Інші маршрутизатори настроюються подібним чином. LDP активується на інтерфейсах, що йдуть у сторону MPLS-хмари й не активується на інтерфейсах маршрутизаторів, що йдуть убік замовників.

Для досягнення прозорого Ethernet-з'єднання між сайтами замовника варто організувати наступні VPLS-тунелі, що утворять повнозв'язну топологію кожний з кожним: **PE1 – PE2**, **PE1 – PE3** і **PE2 – PE3**.

Кожен тунель вимагає створення VPLS-інтерфейсів на обох своїх кінцях.

VPLS-тунелі настроюються в меню **/interface vpls**. Параметр **vpls-id** ідентифікує тунель і повинен бути унікальним для кожного тунелю між двома пірами. Рекомендується призначити MAC-адресу. Необхідні налаштування для замовника A (vpls-id=0:10).

```
[admin@PE1]>interface vpls add name=A1toA2 remote-peer=9.9.9.6 mac-
address=00:00:00:00:A1:A2 vpls-id=0:10 disabled=no
```

```
[admin@PE1]>interface vpls add name=A1toA3 remote-peer=9.9.9.7 mac-
address=00:00:00:00:A1:A3 vpls-id=0:10 disabled=no
[admin@PE2]>interface vpls add name=A2toA1 remote-peer=9.9.9.1 mac-
address=00:00:00:00:A2:A1 vpls-id=0:10 disabled=no
[admin@PE2]>interface vpls add name=A2toA3 remote-peer=9.9.9.7 mac-
address=00:00:00:00:A2:A3 vpls-id=0:10 disabled=no
[admin@PE3]>interface vpls add name=A3toA1 remote-peer=9.9.9.1 mac-
address=00:00:00:00:A3:A1 vpls-id=0:10 disabled=no
[admin@PE3]>interface vpls add name=A3toA2 remote-peer=9.9.9.6 mac-
address=00:00:00:00:A3:A2 vpls-id=0:10 disabled=no
```

Необхідні налаштування для замовника B (vpls-id=0:20) .

```
[admin@PE1]>interface vpls add name=B1toB2 remote-peer=9.9.9.6 mac-
address=00:00:00:00:B1:B2 vpls-id=0:20 disabled=no
[admin@PE1]>interface vpls add name=B1toB3 remote-peer=9.9.9.7 mac-
address=00:00:00:00:B1:B3 vpls-id=0:20 disabled=no
[admin@PE2]>interface vpls add name=B2toB1 remote-peer=9.9.9.1 mac-
address=00:00:00:00:B2:B1 vpls-id=0:20 disabled=no
[admin@PE2]>interface vpls add name=B2toB3 remote-peer=9.9.9.7 mac-
address=00:00:00:00:B2:B3 vpls-id=0:20 disabled=no
[admin@PE3]>interface vpls add name=B3toB1 remote-peer=9.9.9.1 mac-
address=00:00:00:00:B3:B1 vpls-id=0:20 disabled=no
[admin@PE3]>interface vpls add name=B3toB2 remote-peer=9.9.9.6 mac-
address=00:00:00:00:B3:B2 vpls-id=0:20 disabled=no
```

Налаштування VPLS-тунеля призводить до створення динамічного LDP-сусіда та встановлення цільової LDP-сесії. Цільова LDP-сесія – це сесія, що встановлюється між маршрутизаторами, які не є прямими сусідами. Наприклад

```
[admin@PE1] > mpls ldp neighbor pr
       Flags: X - disabled, D - dynamic, O - operational, T - sending-targeted-hello, V - vpls
        #       TRANSPORT   LOCAL-TRANSPORT    PEER   SEND-TARGETED     ADDRESSES
        0 DO   9.9.9.2    9.9.9.1      9.9.9.2:0    no             1.1.1.1
                                                                     2.2.2.1
                                                                   7.7.7.1
                                                                   9.9.9.2
        1 DO   9.9.9.4    9.9.9.1      9.9.9.4:0           no      4.4.4.1
                                                                   5.5.5.2
                                                                   7.7.7.2
                                                                   9.9.9.4
                                                                 10.10.10.1
        2 DOTV 9.9.9.6    9.9.9.1      9.9.9.6:0           yes     3.3.3.2
                                                                   6.6.6.2
                                                                   9.9.9.6
        3 DOTV 9.9.9.7    9.9.9.1      9.9.9.7:0           yes     9.9.9.7
                                                                 10.10.10.2
                                                                 11.11.11.2
```

VPLS-тунель надає віртуальний Ethernet-зв’язок між маршрутизаторами. Для прозорого з’єднання фізичних Ethernet-сегментів вони повинні бути об’єднані в міст із VPLS-тунелем.

У нашому прикладі Ethernet-сегменти мають повнозв’язну топологію. Якщо використовувати мости без протоколу (R)STP, то можуть виникнути петлі

трафіка. Наприклад, якщо від PE1 виходить широкомовний кадр, то він досягне через VPLS-тунелі й PE2 і PE3. PE3, одержавши такий кадр, відішле його PE2. PE2 одержить дві копії одного кадру, що є петлею. Уникнути петель можна так: дозволити (R)STP, використовувати файервол, використовувати властивість horizon. Останнє рішення найпростіше й ефективне.

Базова ідея розщепленого обрію (split horizon) – не пересилати трафік, що виникає на порту, на деяку безліч портів. Для VPLS це значить не пересилати кадри, що з'явилися на одному тунелі в інший тунель, тому що для цього є прямий зв'язок.

Для організації віртуальної мережі замовника А мости на PE1 організувати можна так:

```
[admin@PE1]>interface bridge port add bridge=A interface=A1toA2 horizon=1
[admin@PE1]>interface bridge port add bridge=A interface=A1toA3 horizon=1
```

Для організації віртуальної мережі замовника А мости на PE1 організувати можна так:

```
[admin@PE1]>interface bridge port add bridge=B interface=B1toB2 horizon=1
[admin@PE1]>interface bridge port add bridge=B interface=B1toB3 horizon=1
```

Аналогічним чином виконайте конфігурацію на PE2 та PE3.

Переконайтеся, що отримали VPN рівня 2. Для замовника А:

```
[admin@A1] > ip neighbor pr
      # INTERFACE ADDRESS         MAC-ADDRESS       IDENTITY   VERSION    BOARD
       0 ether1                   00:AB:F6:D2:C3:02 PE2        5.26       x86
       1 ether1                   00:AB:98:83:24:02 PE3        5.26       x86
       2 ether1                   00:AB:F8:16:AE:02 PE1        5.26       x86
       3 ether1    172.16.1.2     00:AB:BD:F6:F6:00 A2         5.26       x86
       4 ether1    172.16.1.3     00:AB:93:D7:B1:00 A3         5.26       x86
```

А1 побачив А2 та А3 на другому рівні. В цьому можна переконатися, використовуючи утиліту ping на другому рівні.

```
[admin@A1] >ping 00:AB:BD:F6:F6:00
[admin@A1] >ping 00:AB:93:D7:B1:00
```

Тут використовуються MAC-адреси інтерфейсів ether1 маршрутизаторів А2 та А3.

```
[admin@A2] > ip neighbor pr

      # INTERFACE ADDRESS         MAC-ADDRESS       IDENTITY   VERSION    BOARD
       0 ether1    172.16.1.1     00:AB:F2:37:E6:00 A1         5.26       x86
       1 ether1                   00:AB:F6:D2:C3:02 PE2        5.26       x86
       2 ether1                   00:AB:98:83:24:02 PE3        5.26       x86
       3 ether1                   00:AB:F8:16:AE:02 PE1        5.26       x86
       4 ether1    172.16.1.3     00:AB:93:D7:B1:00 A3         5.26       x86
```

А2 побачив А1 та А3 на другому рівні. В цьому можна переконатися, використовуючи утиліту ping на другому рівні.

```
[admin@A3] > ip neighbor print
      # INTERFACE ADDRESS         MAC-ADDRESS       IDENTITY   VERSION    BOARD
      0 ether1                    00:AB:F6:D2:C3:02 PE2        5.26       x86
      1 ether1                    00:AB:F8:16:AE:02 PE1        5.26       x86
      2 ether1                    00:AB:98:83:24:02 PE3        5.26       x86
      3 ether1    172.16.1.2      00:AB:BD:F6:F6:00 A2         5.26       x86
      4 ether1    172.16.1.1      00:AB:F2:37:E6:00 A1         5.26       x86
```

А3 побачив А1 та А2 на другому рівні.

Аналогічні результати отримаємо і для сайтів змовника В.

Вивчіть, як MPLS-мітки допомагають органвзувати VPN. Виконайте звичайні пінги з сайта А1 в бік сайта А2.

```
[admin@ A1]>  ping 172.16.1.2
```

Спочатку подивіться інформацію про стан VPLS-інтерфейсів:

```
[admin@PE1] > int vpls monitor numbers=A1toA2
          remote-label: 42
              local-label: 32
            remote-status:
                transport: 9.9.9.6/32
       transport-nexthop: 4.4.4.1
          imposed-labels: 16,42

[admin@PE2] > int vpls monitor numbers=A2toA1
             remote-label: 32
              local-label: 42
            remote-status:
                transport: 9.9.9.1/32
       transport-nexthop: 6.6.6.1
          imposed-labels: 28,32
```

Як видно зі стану інтерфейсу A1toA2 передача здійснюється через Р3. Тобто у цьому випадку для аналізу трафіка необхідно запускати аналізатор пакетів Wireshark на інтерфейсах e1 маршрутизатора PE1, e1 маршрутизатора Р3 і e1 маршрутизатори Р4 (рис. 7.3 – 7.5).

```
⊞ Frame 462: 96 bytes on wire (768 bits), 96 bytes captured (768 bits)
⊞ Ethernet II, Src: 00:ab:f8:16:ae:01 (00:ab:f8:16:ae:01), Dst: 00:ab:11:53:7f:00 (00:ab:11:53:7f:00)
⊞ MultiProtocol Label Switching Header, Label: 16, Exp: 0, S: 0, TTL: 64
⊞ MultiProtocol Label Switching Header, Label: 42, Exp: 0, S: 1, TTL: 64
⊞ PW Ethernet Control Word
⊞ Ethernet II, Src: 00:ab:f2:37:e6:00 (00:ab:f2:37:e6:00), Dst: 00:ab:bd:f6:f6:00 (00:ab:bd:f6:f6:00)
⊞ Internet Protocol Version 4, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.1.2 (172.16.1.2)
⊞ Internet Control Message Protocol
```

Рисунок 7.3 – Структура ICMP-пакету на інтерфейсі e1 маршрутизатора PE1

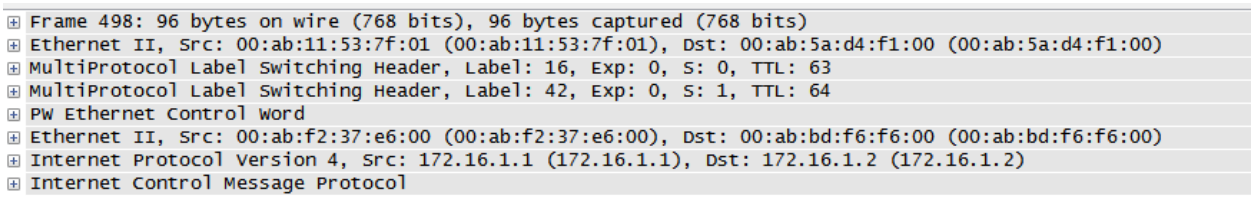

Рисунок 7.4 – Структура ICMP-пакету на інтерфесі e1 маршрутизатора P3

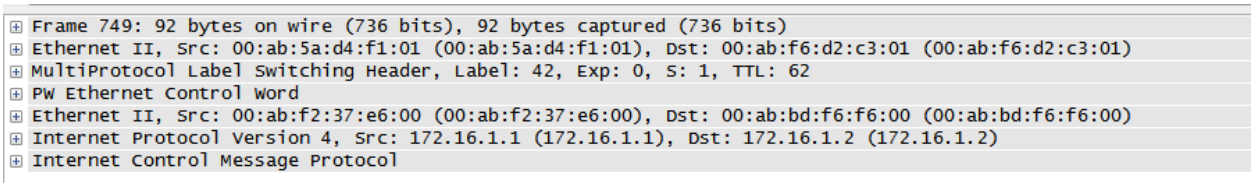

Рисунок 7.5 – Структура ICMP-пакету на інтерфесе e1 маршрутизатора P4

Пояснимо отримане. На всіх 3-х рисунках у частині Ethernet II, бачимо інкапсульований Ethernet-кадр від MAC-адреси 00:AB:F2:37:E6:00 інтерфейсу ether1 сайту A1 до MAC-адреси 00:AB:BD:F6:F6:00 інтерфейсу ether1 сайту A2.

```
[admin@A1] > interface ethernet print where name=ether1
       Flags: X - disabled, R - running, S - slave
        #    NAME     MTU       MAC-ADDRESS       ARP
        0 R  ether1   1500    00:AB:F2:37:E6:00 enabled

[admin@A2] > int ethernet print brief where name=ether1
       Flags: X - disabled, R - running, S - slave
        #    NAME      MTU     MAC-ADDRESS       ARP
        0 R  ether1   1500   00:AB:BD:F6:F6:00   enabled
```

Бачимо, що VPLS на PE1 призначив мітку 32 для тунелю між A1 і A2. Віддаленій стороні призначена мітка 42. Цю мітку можна побачити на всіх 3-х рисунках. Для транспорту Ehernet-кадрів від A1 до A2 використовується мітка 16. Цю мітку можна побачити на рис. 7.3 і 7.4. Таблиця пробросів MPLS на PE1 показує, що пакети з міткою 16 будуть спрямовані на адресу 4.4.4.1 маршрутизатора P3.

```
[admin@PE1] > mpls forwarding-table print
       Flags: L - ldp, V - vpls, T - traffic-eng
        #   IN-LABEL    OUT-LABELS  DESTINATION     INTERFACE      NEXTHOP
       14 L 29             16       9.9.9.6/32      ether2         4.4.4.1
```

Таблиця пробросів MPLS на P3 показує, що пакети з вхідною міткой 16 отримають таку ж вихідну мітку та будуть спрамовані на адресу 5.5.5.1 маршрутизатора P4. Цю мітку можна побачити на рис. 7.4.

```
[admin@P3] > mpls forwarding-table pr
       Flags: L - ldp, V - vpls, T - traffic-eng
        #   IN-LABEL    OUT-LABELS  DESTINATION     INTERFACE      NEXTHOP
       1 L    16           16       9.9.9.6/32      ether2         5.5.5.1
```

1.2. LDP VPLS з організацією LSP за допомогою RSVP

Розглянемо мережу на рис. 7.2. Видаліть в ній підтримку протоколу LDP на маршрутизаторах P1, P2, P3 и P4, які не є граничними:

```
mpls ldp set enabled=no transport-address=0.0.0.0 lsr-id=0.0.0.0
mpls ldp interface remove [find]
```

Бачимо, що VPLS-Інтерфейси перейшли в неробочий стан.

Підтримка протоколу LDP на граничних маршрутизаторах необхідна для організації VPLS-тунелю.

Для всіх LSR-маршрутизаторахз включыть в настроюваннях протоколу OSPF підтримку CSPF

```
routing ospf instance set 0 mpls-te-area=backbone mpls-te-router-id=lobridge
```

Вкажіть, що всі інтерфейси граничних маршрутизаторів, що йдуть убік мережі провайдеру, будуть брати участь у роботі RSVP TE і заявіть пропускні здатності. Наприклад, інтерфейс ether1 граничного маршрутизатора PE1 іде убік MPLS-хмари

```
[admin@PE1]>mpls traffic-eng inter add interface=ether1 bandwidth=100000
```

Аналогічним чином налаштуйте усі інтерфейси маршрутизаторів, що знаходяться в MPLS-мережі.

На граничних маршрутизаторах для майбутніх RSVP TE-туннелей визначте динамично шлях dyn за допомогою CSPF

```
mpls traffic-eng tunnel-path add use-cspf=yes name=dyn
```

Створіть RSVP TE-тунелі, утворюючи повнозв’язну топологію для граничних маршрутизаторів

```
[admin@PE1]>interface traffic-eng add from-address=9.9.9.1 to-address=9.9.9.6
bandwidth=100000 primary-path=dyn disabled=no record-route=yes name = 1to2
[admin@PE1]>interface traffic-eng add from-address=9.9.9.1 to-address=9.9.9.7
bandwidth=100000 primary-path=dyn disabled=no record-route=yes name = 1to3
```

```
[admin@PE2]>interface traffic-eng add from-address=9.9.9.6 to-address=9.9.9.1
bandwidth=100000 primary-path=dyn disabled=no record-route=yes name = 2to1
[admin@PE2]>interface traffic-eng add from-address=9.9.9.6 to-address=9.9.9.7
bandwidth=100000 primary-path=dyn disabled=no record-route=yes name = 2to3

[admin@PE3]>interface traffic-eng add from-address=9.9.9.7 to-address=9.9.9.1
bandwidth=100000 primary-path=dyn disabled=no record-route=yes name = 3to1
[admin@PE3]>interface traffic-eng add from-address=9.9.9.7 to-address=9.9.9.6
bandwidth=100000 primary-path=dyn disabled=no record-route=yes name = 3to2
```

Почекайте, пока підіймуться RSVP TE-інтерфейси. Побачте, що VPLS-інтерфейси також перешли в робочий стан. Переконайтеся, що VPN уровня 2 як і раніше функціонує:

```
[admin@A1] > ip neighbor print
      # INTERFACE ADDRESS   MAC-ADDRESS       IDENTITY   VERSION    BOARD
      0 ether1              00:AB:F8:16:AE:02   PE1        5.26       x86
      1 ether1  172.16.1.2  00:AB:BD:F6:F6:00   A2         5.26       x86
      2 ether1              00:AB:F6:D2:C3:02   PE2        5.26       x86
      3 ether1  172.16.1.3  00:AB:93:D7:B1:00   A3         5.26       x86
      4 ether1              00:AB:98:83:24:02   PE3        5.26       x86
```

Вивчіть, як MPLS-мітки допомагають організувати VPN. Виконайте пінг з сайту A1 в сторону сайта A2:

```
[admin@A1]>  ping 172.16.1.2
```

Подивіться на інформацію про стан VPLS-интерфейса:

```
[admin@PE1] > interface vpls monitor numbers=A1toA2
       remote-label: 78
        local-label: 29
      remote-status:
          transport: 1to2
  transport-nexthop: 1.1.1.1
     imposed-labels: 17,78
```

Як бачемо в цьому разі тунель організовано через маршрутизатор P1 (transport-nexthop: 1.1.1.1). Тому аналізатор трафіка необхідно запускати на інтерфейсах e0 маршрутизатора PE1, e1 маршрутизатора P1 и e1 маршрутизатора P2 . На рис 7.6 – 7.8 наведено відповідні кадри.

```
⊞ Frame 3141: 96 bytes on wire (768 bits), 96 bytes captured (768 bits)
⊞ Ethernet II, Src: 00:ab:f8:16:ae:00 (00:ab:f8:16:ae:00), Dst: 00:ab:2b:93:23:00 (00:ab:2b:93:23:00)
⊞ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 0, TTL: 64
⊞ MultiProtocol Label Switching Header, Label: 78, Exp: 0, S: 1, TTL: 64
⊞ PW Ethernet Control Word
⊞ Ethernet II, Src: 00:ab:f2:37:e6:00 (00:ab:f2:37:e6:00), Dst: 00:ab:bd:f6:f6:00 (00:ab:bd:f6:f6:00)
⊞ Internet Protocol Version 4, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.1.2 (172.16.1.2)
⊞ Internet Control Message Protocol
```

Рисунок 7.6 – Структура ICMP-пакету на інтерфейсі e0 маршрутизатора PE1

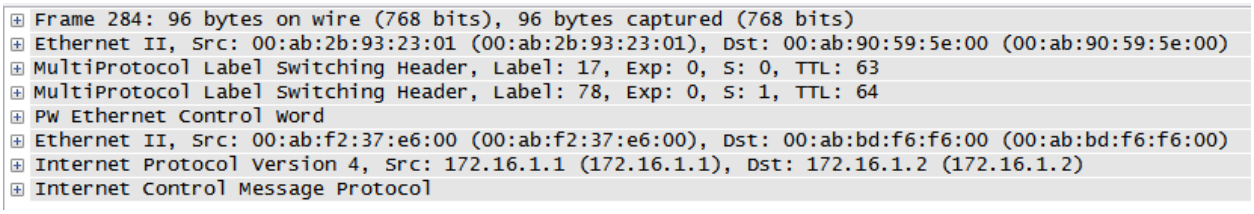

Рисунок 7.7 – Структура ICMP-пакету на інтерфейсі e1 маршрутизатора P1

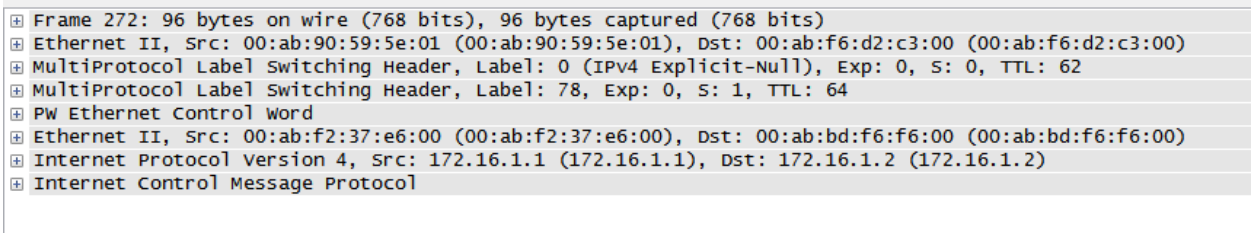

Рисунок 7.8 – Структура ICMP-пакету на інтерфейсі e1 маршрутизатора P2

Пояснимо отримане. На всіх 3-х рисунках у частині Ethernet II, бачимо ынкапсульований Ethernet-кадр від MAC-адреси 00:AB:F2:37:E6:00 інтерфейсу ether1 маршрутизатора A1 до MAC-адреси 00:AB:BD:F6:F6:00 інтерфейсу ether1 маршрутизатора A2.

Бачимо, що VPLS на PE1 призначив мітку 29 для тунеля між A1 і A2. Выддаленій стороні призначена мітка 78. Цю мітку можна побачити на всіх 3-х рисунках. Для транспорту Ethernet-кадрів від A1 до A2 використовується RSVP TE-Інтерфейс 1to2 і мітка 17. Цю мітку можна побачити на рис. 7.6.

Можна побачити інформацію про стан RSVP TE-інтерфейсу:

```
[admin@PE1] > int traffic-eng monitor numbers="1to2"
      tunnel-id: 1
      tunnel-id: 1
      primary-path-state: established
      primary-path: dyn
      secondary-path-state: not-necessary
      active-path: dyn
      active-lspid: 1
      active-label: 17
      explicit-route: S:1.1.1.1/32,S:2.2.2.1/32,S:2.2.2.2/32,S:3.3.3.1/32,
                      S:3.3.3.2/32
      recorded-route: 2.2.2.1[17],3.3.3.1[17],3.3.3.2[0]
      reserved-bandwidth: 10.0kbps
```

Таблиця пробросів MPLS на P1 показує, що пакети з вхідною міткою 17 отримують таку ж вихідну мітку та будуть передані на адресу 2.2.2.2 маршрутизатора P2. Цю мітку можна побачити на рис. 7.7.

```
[admin@P1] > mpls forwarding-table pr
```

```
Flags: L - ldp, V - vpls, T - traffic-eng
 #  IN-LABEL   OUT-LABELS   DESTINATION               INT NEXTHOP
 0   expl-null
 1 T 16        expl-null  9.9.9.6:1->9.9.9.1:3       eth 1.1.1.2
 2 T 17         17        9.9.9.1:1->9.9.9.6:1       eth 2.2.2.2
 3 T 18         18        9.9.9.1:1->9.9.9.7:2       eth 2.2.2.2
```

2. Настройка BGP VPLS

Розглянемо організацію VPLS-тунелю за допомогою протоколу BGP.

Завдяки своїй статичній природі, VPLS-тунелі, засновані на LDP мають проблеми масштабованості, які зростають при збільшенні числа сайтів, підключених по VPLS. Однієї із проблем є вимога збереження повнозв'язної топології LDP-тунелів між сайтами, що формують VPLS. У випадку якщо число вузлів в VPLS високе, додавання нового сайту до існуючої VPLS-мережі може стати обтяжним для адміністратора мережі. Додавання нового вузла потребує створення необхідної кількості VPLS-тунелів на маршрутизаторі, до якого приєднаний новий вузол, що потребує настроювання маршрутизаторів на інших кінцях тунелів.

Загалом,VPLS, засновані на BGP, служать двом цілям:

• автоматичного виявлення: немає необхідності налаштовувати VPLS-зв'язок кожного нового граничного маршрутизатора з усіма віддаленими кінцевими точками тунелів VPLS;

• сигналізації: мітки, що використовуються для тунелів VPLS на віддалених кінцевих точках, поширюються в тому самому відновленні BGP.

Це значить, що немає необхідності для цільових сесій LDP між кінцевими точками тунелю, як у випадку VPLS на основі LDP.

Правильне настроювання VPLS, заснованих на BGP, дозволяє уникнути конфігурації маршрутизаторів, які не підключені до нового вузла.

Для організації VPLS BGP маршрутизатори обмінюються повідомленнями NLRI (Network Layer Reachability Information), що містять якусь інформацію про VPLS.

BGP VPLS це лише метод обміну мітками для тунелів VPLS, а не метод обміну трафиком між кінцевими крапками тунелів VPLS. Тому варто забезпечити поширення кадрів MPLS між кінцевими точками тунелів VPLS.

Кадри MPLS поширюються по шляхах LSP. LSP організовуються або за допомогою протоколу LDP або за допомогою протоколу RSVP.

2.1. Конфігурація сесій BGP. Відбивач маршрутів

Для забезпечення поширення повідомлень VPLS NLRI по BGP повинна бути використана многопротокольная можливість BGP. Це здійснюється установкою в BGP-пірі для властивості **address-families** значення **l2vpn**.

Для організації BGP VPLS необхідно забезпечити доставку многопротокольної NLRI між VPLS-маршрутизаторами. Це можна зробити або шляхом встановлення BGP-сесій між всіма парами VPLS-маршрутизаторів, або використанням відбивача маршрутів. У першому випадку перевага BGP VPLS над LDP VPLS сумнівна: при додаванні нового сайту в VPLS необхідно настроїти BGP-піри на всіх маршрутизаторах, що формують VPLS.

При використанні відбивача маршрутів додавання нового сайту до VPLS стає більше простим – маршрутизатор, до якого приєднаний новий сайт, повинен тільки встановити BGP сесію з відбивачем маршрутів. На інших маршрутизаторах (крім відбивача маршрутів) настроювань не потрібно. Сам маршрутизатор – відбивач маршрутів може також брати участь у формуванні VPLS.

У найпростішому випадку відбивач маршрутів передає отриманий BGP-маршрут без зміни атрибута NextHop для маршруту. Ця властивість дозволяє уникнути BGP-з'єднань типу кожен з кожним. Для того щоб маршрутизатор був відбивачем маршрутів для проходження повідомлень VPLS NLRI він не зобов'язаний брати участь в VPLS і навіть може не підтримувати MPLS.

Є кілька зауважень про конфігурацію пірів BGP:

1. Для обміну повідомленнями VPLS NLRI немає потреби поширювати IP або IPv6 маршрути, досить визначити address-families=l2vpn ;

2. Для адресації пірів використовуються адреси мостів, тобто інтерфейсу lobridge (локальна адреса визначається установкою update-source). BGP пір, починаючи передавати повідомлення VPLS NLRI, призначає свою локальну адреса як BGP NextHop. Приймаючий VPLS-маршрутизатор використовує отриману адресу BGP NextHop, як адресу кінцевої крапки тунелю.

Зробимо маршрутизатор P1 відбивачем маршрутів. Екземпляр BGP для відбивача маршрутів повинен мати властивість **client-to-client-reflection=yes**, що є установкою за замовчуванням.

Для забезпечення належного поширення повідомлень VPLS NLRI BGP-піри убік маршрутизаторів PE1, PE2 і PE3 повинні бути обов'язково налаштовані з установкою **route-reflect=yes**:

```
[admin@P1]>routing bgp peer add remote-address=9.9.9.1 remote-as=65530 route-reflect=yes address-families=l2vpn update-source=lobridge
[admin@P1]>routing bgp peer add remote-address=9.9.9.6 remote-as=65530 route-reflect=yes address-families=l2vpn update-source=lobridge
[admin@P1]>routing bgp peer add remote-address=9.9.9.7 remote-as=65530 route-reflect=yes  address-families=l2vpn update-source=lobridge
```

В налаштуваннях BGP-пірів в PE1, PE2 та PE3 в бік відбивача P1 (9.9.9.2) залишаємо для **route-reflect** значення за замовченням **no**.

```
routing bgp peer add remote-address=9.9.9.2 remote-as=65530 address-families=l2vpn update-source=lobridge
```

Повинно встановитися три BGP-з'єднання. Про це можна переконатися, переглянувши час uptime існування з'єднання командою

```
routing bgp peer print status.
```

Якщо треба додати новий сайт в VPLS-мережу, приєднуючи його до маршрутизатора P2, необхідно лише настроїти BGP-пір на відбивачі P1 убік P2 і BGP-пір на P2 убік відбивача P1. Причому на P1 з установкою **route-reflect=yes**.

2.2. BGP VPLS з організацією LSP за допомогою RSVP

Підготуйте топологію для подальшої роботи. Спочатку видаліть VPLS-інтерфейси з мостів, а потім видаліть і самі VPLS-інтерфейси. Пам'ятаємо, що на даний момент LSP організовані за допомогою протоколу RSVP.

VPLS-тунелі створюються динамічно при одержанні за допомогою BGP-сигналізації правильного повідомлення BGP NLRI. Отже, не треба створювати VPLS-інтерфейси.

Для прозорої доставки Ethernet-сегментів крізь VPLS-мережу повинні бути настроєні мости. Мости вже створені (для LDP VPLS). Залиште в них тільки фізичні Ethernet-інтерфейси.

Активація VPLS на основі BGP-сигналізації змушує маршрутизатор розсилати інформацію VPLS BGP NLRI, що вказує, що він належить деякої

VPLS. Одержавши таку інформацію, інші члени тієї ж VPLS будуть знати, як установити VPLS-тунель із цим маршрутизатором.

Для налаштування двох VPLS для замовників A і B на граничних маршрутизаторах треба виконати команди для активації VPLS на основі BGP-сигналізації:

```
[admin@PE1]>interface vpls bgp-vpls add bridge=A bridge-horizon=1 route-distinguisher=9.9.9.1:1 site-id=1 export-route-targets=1:1 import-route-targets=6:1,7:1
[admin@PE1]>interface vpls bgp-vpls add bridge=B bridge-horizon=1 route-distinguisher=9.9.9.1:2 site-id=1 export-route-targets=1:2 import-route-targets=6:2,7:2

[admin@PE2]>interface vpls bgp-vpls add bridge=A bridge-horizon=1 route-distinguisher=9.9.9.6:1 site-id=6 export-route-targets=6:1 import-route-targets=1:1,7:1
[admin@PE2]>interface vpls bgp-vpls add bridge=B bridge-horizon=1 route-distinguisher=9.9.9.6:2 site-id=6 export-route-targets=6:2 import-route-targets=1:2,7:2

[admin@PE3]>interface vpls bgp-vpls add bridge=A bridge-horizon=1 route-distinguisher=9.9.9.7:1 site-id=7 export-route-targets=7:1 import-route-targets=1:1,6:1
[admin@PE3]>interface vpls bgp-vpls add bridge=B bridge-horizon=1 route-distinguisher=9.9.9.7:2 site-id=7 export-route-targets=7:2 import-route-targets=1:2,6:2
```

Тут

site-id – повинне бути унікально для кожного маршрутизатора в межах конкретної VPLS. Для підвищення ефективності варто брати ці значення з вузького діапазону цілих чисел.

Bridge – визначає міст, до якого додадуться динамічно створювані VPLS-тунелі.

Bridge-horizon – служить для запобігання петель.

route-distinguisher визначає значення, що прикріплюється до VPLS NLRI; маршрутизатори використовують це значення для утворення тунелю. Щоб приймаючі маршрутизатори розрізняли інформацію від різних VPLS, це значення повинне бути різним для різних VPLS на одному пристрої. **Route-distinguisher** не використовується маршрутизатором, що одержує, для визначення приналежності передавального маршрутизатора конкретної VPLS. Для цього використовується **route-targets**.

export-route-targets – це свого роду мітка (синонім) для **route-distinguisher**.

import-route-targets – це список із зовнішніх **export-route-targets**, який використовується для визначення сукупності маршрутизаторів, що утворять конкретну VPLS.

route-distinguisher, **export-route-targets** і **import-route-targets** мають вигляд **A:B**, де **A** – або IP-адреса (будь-яка), або ціле число (іноді номер автономної системи). **B** – ціле число. У призначенні цих параметрів має місце певне свавілля. У принципі **route-distinguisher** і **export-route-targets** можуть бути будь-якою (унікальної для роутера) парою цілих чисел або парою адреса-число.

Після настроювання роутеры обміняються BGP-пакетами (рис. 7.9).

У підсумку створяться динамічні VPLS-Інтерфейси на PE1, PE2 і PE3.

Наприклад, на PE1

```
[admin@PE1]>interface vpls print
Flags: X - disabled, R - running, D - dynamic,
B - bgp-signaled, C - cisco-bgp-signaled
 0 RDB name="vpls1" mtu=1500 l2mtu=1500 mac-address=02:BE:67:2A:E3:EA arp=enabled
       disable-running-check=no remote-peer=9.9.9.6 cisco-style=no
       cisco-style-id=0 advertised-l2mtu=1500 pw-type=raw-ethernet
       use-control-word=yes vpls=bgp-vpls1

 1 RDB name="vpls2" mtu=1500 l2mtu=1500 mac-address=02:70:36:CD:4D:C2 arp=enabled
       disable-running-check=no remote-peer=9.9.9.6 cisco-style=no
       cisco-style-id=0 advertised-l2mtu=1500 pw-type=raw-ethernet
       use-control-word=yes vpls=bgp-vpls2

 2 RDB name="vpls3" mtu=1500 l2mtu=1500 mac-address=02:05:39:BD:E2:1D arp=enabled
       disable-running-check=no remote-peer=9.9.9.7 cisco-style=no
       cisco-style-id=0 advertised-l2mtu=1500 pw-type=raw-ethernet
       use-control-word=yes vpls=bgp-vpls1

 3 RDB name="vpls4" mtu=1500 l2mtu=1500 mac-address=02:9D:B8:4D:7A:D3 arp=enabled
       disable-running-check=no remote-peer=9.9.9.7 cisco-style=no
       cisco-style-id=0 advertised-l2mtu=1500 pw-type=raw-ethernet
       use-control-word=yes vpls=bgp-vpls2
```

```
123 77.402337  9.9.9.2          9.9.9.1          BGP      369 UPDATE Message, UPDATE Message, UPDATE Message
Border Gateway Protocol - UPDATE Message
  Marker: ffffffffffffffffffffffffffffffff
  Length: 101
  Type: UPDATE Message (2)
  Unfeasible routes length: 0 bytes
  Total path attribute length: 78 bytes
- Path attributes
  + ORIGIN: INCOMPLETE (4 bytes)
  + AS_PATH: empty (3 bytes)
  + LOCAL_PREF: 100 (7 bytes)
  + ORIGINATOR_ID: 9.9.9.2 (7 bytes)
  + CLUSTER_LIST: 9.9.9.2 (7 bytes)
  - EXTENDED_COMMUNITIES: (19 bytes)
    + Flags: 0xc0 (Optional, Transitive, Complete)
      Type code: EXTENDED_COMMUNITIES (16)
      Length: 16 bytes
    - Carried Extended communities
        two-octet AS specific Route Target: 6:1
      + Layer 2 Information: Unknown, Control Flags: FS, MTU: 1500 bytes
  - MP_REACH_NLRI (31 bytes)
    + Flags: 0x80 (Optional, Non-transitive, Complete)
      Type code: MP_REACH_NLRI (14)
      Length: 28 bytes
      Address family: Layer-2 VPN (25)
      Subsequent address family identifier: VPLS (65)
    - Next hop network address (4 bytes)
        Next hop: IPv4=9.9.9.6 (4)
      Subnetwork points of attachment: 0
    - Network layer reachability information (19 bytes)
        RD: 6.9.9.9:1, CE-ID: 6, Label-Block Offset: 0, Label-Block Size: 16, Label Base 16 (bottom)
```

Рисунок 7.9 – BGP-повідомлення Update від PE2 до PE1.

На рис. 7.9 можна побачити в розділі NLRI route-targets 6:1 и route-distinguisher 9.9.9.6:1 у вигляді 6.9.9.9:1.

Динамічні VPLS-інтерфейси автоматчно додаються у міст, що можна переглянути на PE1, PE2 и PE3 командою **`interface bridge port print`**, наприклад

```
[admin@PE1] > interface bridge port print
      Flags: X - disabled, I - inactive, D - dynamic
       #    INTERFACE      BRIDGE        PRIORITY  PATH-COST    HORIZON
       0    ether3           A             0x80         10       none
       1    ether5           B             0x80         10       none
       2  D vpls1            B             0x80         50          1
       3  D vpls2            A             0x80         50          1
       4  D vpls3            A             0x80         50          1
       5  D vpls4            B             0x80         50          1
```

Тут ми також отримали підтвердження, що працює відбиття маршруту на P1, тому що немає BGP-пірів між PE1 і PE2, PE1 і PE3, PE2 і PE3.

Встановлена повнозв'язна топологія VPLS-тунелів. Подивіться сусідів на сайтах замовника A и B. Переконайтеся, що наша VPN рівня 2 як і раніше функціонує.

Вивчить, як MPLS-мітки допомагають організувати VPN. Виконайте пінг з сайту A1 убік сайту A2:

```
[admin@A1]> ping 172.16.1.2
```

```
⊞ Frame 999: 96 bytes on wire (768 bits), 96 bytes captured (768 bits)
⊞ Ethernet II, Src: 00:ab:f8:16:ae:00 (00:ab:f8:16:ae:00), Dst: 00:ab:2b:93:23:00 (00:ab:2b:93:23:00)
⊞ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 0, TTL: 64
⊞ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 1, TTL: 64
⊞ PW Ethernet Control Word
⊞ Ethernet II, Src: 00:ab:f2:37:e6:00 (00:ab:f2:37:e6:00), Dst: 00:ab:bd:f6:f6:00 (00:ab:bd:f6:f6:00)
⊞ Internet Protocol Version 4, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.1.2 (172.16.1.2)
⊞ Internet Control Message Protocol
```

Рисунок 7.10 – Структура ICMP-пакету на інтерфейсі e0 маршрутизатора PE1

```
⊞ Frame 1536: 96 bytes on wire (768 bits), 96 bytes captured (768 bits)
⊞ Ethernet II, Src: 00:ab:2b:93:23:01 (00:ab:2b:93:23:01), Dst: 00:ab:90:59:5e:00 (00:ab:90:59:5e:00)
⊞ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 0, TTL: 63
⊞ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 1, TTL: 64
⊞ PW Ethernet Control Word
⊞ Ethernet II, Src: 00:ab:f2:37:e6:00 (00:ab:f2:37:e6:00), Dst: 00:ab:bd:f6:f6:00 (00:ab:bd:f6:f6:00)
⊞ Internet Protocol Version 4, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.1.2 (172.16.1.2)
⊞ Internet Control Message Protocol
```

Рисунок 7.11 – Структура ICMP-пакету на інтерфейсі e1 маршрутизатора P1

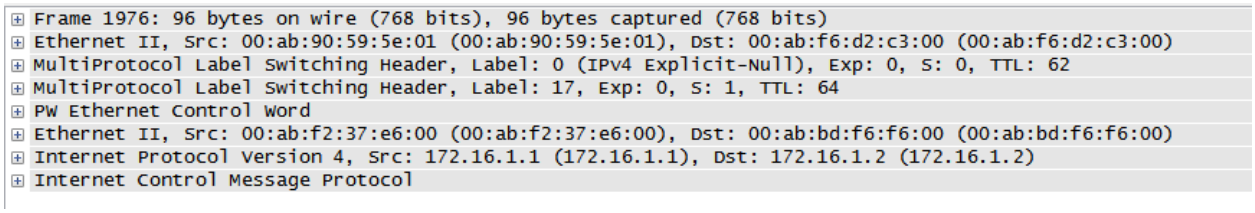

Рисунок 7.12 – Структура ICMP-пакету на інтерфейсі e1 маршрутизатора P1

На рис. 7.10 – 7.12 наведені структури кадрів, захоплені на інтерфейсі e0 маршрутизатора PE1, інтерфейсі e1 маршрутизатора P1 та інтерфейсі e1 маршрутизатора P1 при виконанні пінгу з сайту A1 убік сайту A2.

Пояснимо отримане. На всіх 3-х рисунках у частині Ethernet II, бачимо інкапсульований Ethernet-кадр від інтерфейсу ether1 сайту A1 до інтерфейсу ether1 сайту A2.

Можна побачити інформацію про стан VPLS-інтерфейсу

```
[admin@PE1] > interface vpls monitor numbers=2
             remote-label: 17
              local-label: 22
            remote-status:
                transport: 1to2
        transport-nexthop: 1.1.1.1
           imposed-labels: 17,17
```

Бачимо, що VPLS на PE1 призначив мітку 22 для тунелю між A1 і A2. Выддаленій стороні призначена мітка 17. Цю мітку можна побачити на всіх 3-х рисунках. Для транспорту Ehernet-кадрів від A1 до A2 використовується RSVP TE-Інтерфейс 1to2 і мітка 17. Цю мітку можна побачити на рис. 7.10.

Можна побачити інформацію про стан RSVP TE-інтерфейсу

```
[admin@PE1] > interface traffic-eng monitor numbers="1to2"
                   tunnel-id: 1
          primary-path-state: established
                primary-path: dyn
        secondary-path-state: not-necessary
                 active-path: dyn
                active-lspid: 1
                active-label: 17
               explicit-route:
S:1.1.1.1/32,S:2.2.2.1/32,S:2.2.2.2/32,S:3.3.3.1/32,S:3.3.3.2/32
              recorded-route: 2.2.2.1[17],3.3.3.1[17],3.3.3.2[0]
          reserved-bandwidth: 10.0kbps
```

Таблиця пробросыв MPLS на P1 показує, що пакети із вхідною міткою 17 отримують таку ж вихідну мітку й будуть спрямовані на адресу 2.2.2.2 маршрутизатора P2. Цю мітку можна побачити на рис. 7.11.

```
[admin@P1]> mpls forwarding-table print
      Flags: L - ldp, V - vpls, T - traffic-eng
       #   IN-LABEL   OUT-LABELS        DESTINATION    INTERFACE    NEXTHOP
      0  expl-null
      1 T   16        expl-null   9.9.9.6:1->9.9.9.1:1   ether1     1.1.1.2
      2 T   17            17      9.9.9.1:1->9.9.9.6:1   ether2     2.2.2.2
```

2.3. BGP VPLS з організацією LSP за допомогою LDP

Пам’ятаємо, що на даний момент LSP організовані за допомогою протоколу RSVP. Замінимо RSVP на LDP. Для цього на всіх граничних маршрутизаторах видалимо RSVP TE інтерфейси

```
interface traffic-eng rem [find]
```

Для всіх LSR-маршрутизаторів відключіть в налаштуваннях протоколу OSPF підтримку CSPF

```
routing ospf instance set 0 mpls-te-area=
routing ospf instance set 0  mpls-te-router-id=
```

Видаліть інтерфейси mpls traffic-eng в маршрутизаторах PE1, PE2 та PE3, які беруть участь в роботі RSVP TE

```
mpls traffic-eng interface rem [find]
```

Для організації шляхів LSP комутації пакетів по мітці активуйте на кожному LSR-маршрутизаторі протокол LDP відповідно до розділу 1.1. Транспортну адресу встановіть як адресу інтерфейсу Loopback з ім’ям lobridge. Оголосіть інтерфейси, що дивляться усередину MPLS-мережі, як інтерфейси, що беруть участь в обміні міток.

Автоматично створяться динамічні VPLS-інтерфейси на PE1, PE2 і PE3. Наприклад, на PE1

```
[admin@PE1] > interface vpls print
```

```
Flags: X - disabled, R - running, D - dynamic, B - bgp-signaled, C - cisco-bgp-signaled
 0 RDB name="vpls1" mtu=1500 l2mtu=1500 mac-address=02:A7:B0:F6:31:6C arp=enabled disable-running-check=no
       remote-peer=9.9.9.7 cisco-style=no cisco-style-id=0 advertised-l2mtu=1500 pw-type=raw-ethernet
       use-control-word=yes vpls=bgp-vpls2

 1 RDB name="vpls3" mtu=1500 l2mtu=1500 mac-address=02:C8:28:96:B6:15 arp=enabled disable-running-check=no
       remote-peer=9.9.9.7 cisco-style=no cisco-style-id=0 advertised-l2mtu=1500 pw-type=raw-ethernet
       use-control-word=yes vpls=bgp-vpls1

 2 RDB name="vpls2" mtu=1500 l2mtu=1500 mac-address=02:38:3B:5F:93:DD arp=enabled disable-running-check=no
       remote-peer=9.9.9.6 cisco-style=no cisco-style-id=0 advertised-l2mtu=1500 pw-type=raw-ethernet
       use-control-word=yes vpls=bgp-vpls1

 3 RDB name="vpls4" mtu=1500 l2mtu=1500 mac-address=02:6F:0C:E1:29:78 arp=enabled disable-running-check=no
       remote-peer=9.9.9.6 cisco-style=no cisco-style-id=0 advertised-l2mtu=1500 pw-type=raw-ethernet
       use-control-word=yes vpls=bgp-vpls2
```

Динамічні VPLS-інтерфейси автоматично додадуться у міст, що можна побачити на PE1, PE2 и PE3 командою interface bridge port print, наприклад:

```
[admin@PE1] > interface bridge port print
       Flags: X - disabled, I - inactive, D - dynamic
        #    INTERFACE       BRIDGE      PRIORITY  PATH-COST    HORIZON
        0     ether3           A              0x80         10       none
        1     ether4           B              0x80         10       none
        2  D  vpls1           A              0x80         50          1
        3  D  vpls2           A              0x80         50          1
        4  D  vpls3           B              0x80         50          1
        5  D  vpls4           B              0x80         50          1
```

Встановлена повнозв’язна топологія VPLS-тунелів. Подивиться сусідів на сайтах замовника A и B. Переконайтеся, що VPN рівня 2 як і раніше функціонує.

Вивчіть, як MPLS-мітки допомагають організувати VPN. Виконайте пінг з сайту A1 убік сайту A2.

```
[admin@A1]>  ping 172.16.1.2
```

За допомогою аналізатора пакетів Wireshark отримайте структуру ICMP-пакетів на інтерфесах e0 маршрутизатора PE1, e1 маршрутизатора P1 і e1 маршрутизатора P2, яка повністю аналогічна структурі для випадку п. 1.1 (рис. 7.13).

```
⊞ Frame 6240: 96 bytes on wire (768 bits), 96 bytes captured (768 bits)
⊞ Ethernet II, Src: 00:ab:f8:16:ae:00 (00:ab:f8:16:ae:00), Dst: 00:ab:2b:93:23:00 (00:ab:2b:93:23:00)
⊞ MultiProtocol Label Switching Header, Label: 33, Exp: 0, S: 0, TTL: 64
⊞ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 1, TTL: 64
⊞ PW Ethernet Control Word
⊞ Ethernet II, Src: 00:ab:f2:37:e6:00 (00:ab:f2:37:e6:00), Dst: 00:ab:bd:f6:f6:00 (00:ab:bd:f6:f6:00)
⊞ Internet Protocol Version 4, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.1.2 (172.16.1.2)
⊞ Internet Control Message Protocol
```

```
⊞ Frame 6468: 96 bytes on wire (768 bits), 96 bytes captured (768 bits)
⊞ Ethernet II, Src: 00:ab:2b:93:23:01 (00:ab:2b:93:23:01), Dst: 00:ab:90:59:5e:00 (00:ab:90:59:5e:00)
⊞ MultiProtocol Label Switching Header, Label: 16, Exp: 0, S: 0, TTL: 63
⊞ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 1, TTL: 64
⊞ PW Ethernet Control Word
⊞ Ethernet II, Src: 00:ab:f2:37:e6:00 (00:ab:f2:37:e6:00), Dst: 00:ab:bd:f6:f6:00 (00:ab:bd:f6:f6:00)
⊞ Internet Protocol Version 4, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.1.2 (172.16.1.2)
⊞ Internet Control Message Protocol
```

```
⊞ Frame 269: 92 bytes on wire (736 bits), 92 bytes captured (736 bits)
⊞ Ethernet II, Src: 00:ab:90:59:5e:01 (00:ab:90:59:5e:01), Dst: 00:ab:f6:d2:c3:00 (00:ab:f6:d2:c3:00)
⊞ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 1, TTL: 62
⊞ PW Ethernet Control Word
⊞ Ethernet II, Src: 00:ab:f2:37:e6:00 (00:ab:f2:37:e6:00), Dst: 00:ab:bd:f6:f6:00 (00:ab:bd:f6:f6:00)
⊞ Internet Protocol Version 4, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.1.2 (172.16.1.2)
⊞ Internet Control Message Protocol
```

Рисунок 7.13 – Структура ICMP-пакетів при виконанні пінгу від A1 до A2

Можна побачити інформацію про стан VPLS-інтерфейсів

```
[admin@PE1] > interface vpls monitor numbers=2
             remote-label: 17
              local-label: 22
            remote-status:
                transport: 9.9.9.6/32
          transport-nexthop: 1.1.1.1
             imposed-labels: 33,17
```

Бачимо, що VPLS на PE1 призначив мітку 22 для тунеля між A1 і A2.

Віддаленій стороні призначена мітка 17. Для транспорту Ehernet-кадрів від A1 до A2 використовується мітка 33. Таблиця пробросів MPLS на PE1 показує, що пакети з міткою 33 будуть напрівлені на адресу 1.1.1.1 маршрутизатора P1.

```
[admin@PE1] > mpls forwarding-table print
      Flags: L - ldp, V - vpls, T - traffic-eng
       #   IN-LABEL   OUT-LABELS     DESTINATION   INTERFACE      NEXTHOP
       0  expl-null
      1      V 22                       vpls2
      15     L 90        33         9.9.9.6/32      ether1        1.1.1.1
```

Таблиця пробросів MPLS на P1 показує, що пакети з вхідною міткою 33 отримують вихідну мітку 16 та будуть направлені на адресу 2.2.2.2 маршрутизатора P2.

```
[admin@P1] > mpls forwarding-table print
      Flags: L - ldp, V - vpls, T - traffic-eng
      #   IN-LABEL   OUT-LABELS     DESTINATION   INTERFACE      NEXTHOP
      0  expl-null
      10     L 33        16         9.9.9.6/32      ether2        2.2.2.2
```

Таблиця пробросів MPLS на P2 показует, що пакети з вхідною міткою 16 не отримують вихідну мітку (тобто P2 виконує операцію **pop label**) та будуть направлені на адресу 3.3.3.2 маршрутизатора P3.

```
[admin@P2] > mpls forwarding-table print
      Flags: L - ldp, V - vpls, T - traffic-eng
      #   IN-LABEL   OUT-LABELS     DESTINATION   INTERFACE      NEXTHOP
      0  expl-null
      1     L 16                   9.9.9.6/32      ether2        3.3.3.2
```

Таблиця пробросів MPLS на PE2 показує, что пакети з вхідною міткою 17 не отримують вихідну мітку (тобто PE2 виконує операцію **pop label**) та будуть направлені на інтерфейс **vpls4**.

```
[admin@PE2] > mpls forwarding-table print
      Flags: L - ldp, V - vpls, T - traffic-eng
      #    IN-LABEL    OUT-LABELS      DESTINATION    INTERFACE       NEXTHOP
      0  expl-null
      1      L 17                         vpls4
```

Інформацію про vpls інтерфейси можні переглянути командою

```
[admin@PE2] > interface vpls print
```

```
Flags: X - disabled, R - running, D - dynamic, B - bgp-signaled, C - cisco-bgp-signaled
 0 RDB name="vpls1" mtu=1500 l2mtu=1500 mac-address=02:7C:CC:81:14:67 arp=enabled disable-running-check=no remote-peer=9.9.9.7
       cisco-style=no cisco-style-id=0 advertised-l2mtu=1500 pw-type=raw-ethernet use-control-word=yes vpls=bgp-vpls1

 1 RDB name="vpls2" mtu=1500 l2mtu=1500 mac-address=02:4F:29:FC:3D:75 arp=enabled disable-running-check=no remote-peer=9.9.9.7
       cisco-style=no cisco-style-id=0 advertised-l2mtu=1500 pw-type=raw-ethernet use-control-word=yes vpls=bgp-vpls2

 2 RDB name="vpls3" mtu=1500 l2mtu=1500 mac-address=02:B3:FB:D9:D2:E2 arp=enabled disable-running-check=no remote-peer=9.9.9.1
       cisco-style=no cisco-style-id=0 advertised-l2mtu=1500 pw-type=raw-ethernet use-control-word=yes vpls=bgp-vpls2

 3 RDB name="vpls4" mtu=1500 l2mtu=1500 mac-address=02:A6:08:73:C9:22 arp=enabled disable-running-check=no remote-peer=9.9.9.1
       cisco-style=no cisco-style-id=0 advertised-l2mtu=1500 pw-type=raw-ethernet use-control-word=yes vpls=bgp-vpls1
```

А які порти знаходяться в яких мостах командою:

```
[admin@PE2] > interface bridge port print
      Flags: X - disabled, I - inactive, D - dynamic
       #    INTERFACE      BRIDGE     PRIORITY  PATH-COST    HORIZON
       0    ether3           A          0x80       10          none
       1    ether5           B          0x80       10          none
       2  D vpls1            A          0x80       50            1
       3  D vpls2            B          0x80       50            1
       4  D vpls3            B          0x80       50            1
       5  D vpls4            A          0x80       50            1
```

Прослідкуйте призначення міток при передачі пакетів у зворотньому напрямку (від A2 до A1). Збережіть топологію, вона буде використовуватися в ЛР №21.

Зміст звіту

Звіт готується в електронному вигляді і роздруковується. Звіт містить усі виконані пункти практичної роботи зі скріншотами.

Порядок здачі роботи

1. Вивчити теоретичну і практичну частину.
2. Здати викладачеві теорію роботи, відповівши на контрольні питання.
3. Виконати практичну частину.
4. Оформити та захистити звіт.

Контрольні питання

1. Для чого використовується технологія MPLS L2VPN2?
2. Що таке VPLS та для чого використовується?
3. Які переваги дає VPLS у порівнянні з іншими L2VPN?
4. Як організується VPLS-тунель за допомогою LDP з LSP на основі LDP?
5. Як організується VPLS-тунель за допомогою LDP з LSP на основі RSVP?
6. Як організується VPLS-тунель за допомогою BGP з LSP на основі LDP?
7. Як організується VPLS-тунель за допомогою BGP з LSP на основі RSVP?
7. Для чого використовується split horizon при організації мосту?
8. Для чого використовується стек міток і як він обробляється?

Лабораторна робота 8

ТЕХНОЛОГІЯ MPLS L3VPN

Мета роботи: Ознайомлення з принципами побудови VPN рівня 3 на основі MPLS-мереж. Набуття практичних навичок налаштування параметрів VPN рівня 3 для MPLS-мереж на мережному обладнанні.

Теоретична частина

Цей тип VPN ґрунтується на технології VRF (Virtual Routing and Forwarding – віртуальна маршрутизація й пересилання), що дозволяє одночасне існування на пристрої декількох таблиць маршрутизації. RouterOS підтримує VRF, що дозволяє створювати багато екземплярів таблиць для віртуальної маршрутизації й пересилання. Це корисно для MPLS VPN, заснованих на BGP. На відміну від BGP VPLS, що працює на 2-м рівні OSI, VRF VPN працюють на 3-м рівні й обмінюється IP-префіксами між маршрутизаторами. VRF вирішує проблем перетинання однакових IP-префіксів і забезпечує конфіденційність за допомогою роздільної маршрутизації для різних VPN.

VRF инициализируется в меню **/ip route vrf**. Для визначення таблиці маршрутизації VRF варто задати атрибут routing-mark, інтерфейс, route distinguisher і списки експорту й імпорту. Приєднаний маршрут, що відповідає цьому інтерфейсу, автоматично додається в цю VRF-таблицю маршрутизації. Список експорту повинен містити хоча б один елемент для цього VRF. Зазвичай використовують співвідношення один-до-одного між route distinguisher і окремою таблицею маршрутизації VRF, але це не обов’язково.

Додавати маршрут у таблицю маршрутизації VRF можна, визначаючи атрибут routing-mark.

Для поширення між маршрутизаторами маршрутів з таблиці маршрутів VRF використовується многопротокольный BGP з адресним сімейством VPNv4. Для кожного екземпляра BGP, що беруть участь в VRF-маршрутизації варто задати список VRF. VRF визначається атрибутом routing-mark.

Додавання маршруту в таблицю VRF управляється атрибутами BGP. Як тільки VRF для BGP настроєні, створюються активні маршрути сімейства адрес

VPNv4. Ці маршрути встановлюються в окремі таблицю маршрутів і їх можна побачити з меню **/routing bgp vpnv4-route**. Через BGP можна поширювати однакові префікси IPv4 для різних мереж. Як правило, маршрути VPNv4 з однаковими префіксами будуть поширюватися тільки після належного налаштування MPLS.

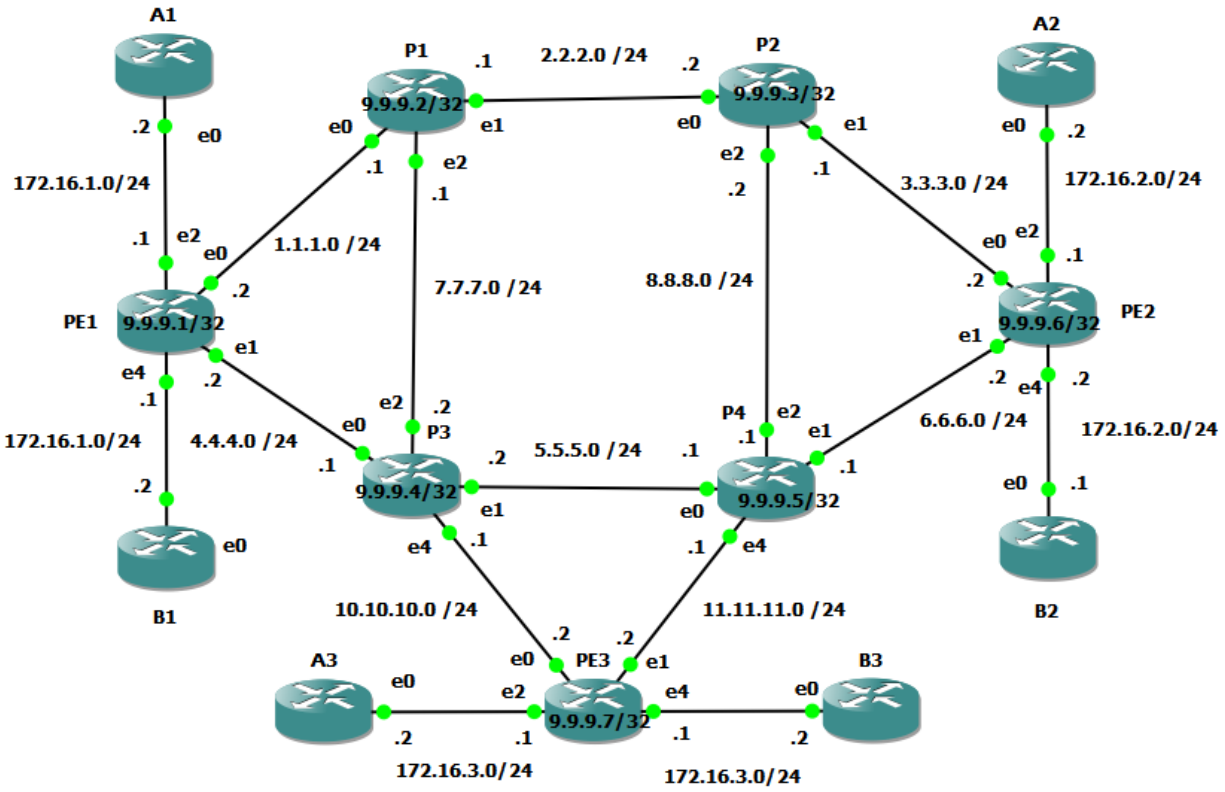

Рисунок 8.1 – Топологія для MPLS VPN 3-го рівня

VRF-маршрутизація використовує так звані VPNv4-маршрути, що працює із префіксами, які складаються з route-distinguisher і префікса IPv4. Тим самим на одному маршрутизаторі можна по-різному маршрутизувати пакети з однаковими префіксами IPv4, що приходять від різних джерел. Як правило VRF-маршрутизація функціонує тільки після належного настроювання MPLS.

Практична частина

1. MPLS VPN 3-го рівня з організацією LSP за допомогою LDP

Розглянемо налаштування MPLS VPN 3-го рівня на прикладі топології на рис. 8.1, що є копією топології рис. 8.2 і відрізняється від неї лише

налаштуванням адрес пристроїв A1, A2, A3, B1, B2, B3 замовників (скористайтеся топологією ЛР №7).

Замовники A і B вимагають зв'язати в єдиний адресний простір свої три IP-мережі. Причому ці мережі в них виявилися однаковими: 172.16.1.0/24, 172.16.2.0/24, 172.16.3.0/24.

Усередині хмари MPLS шляхи комутації пакетів по мітці LSP організовуються або за допомогою протоколу LDP або за допомогою протоколу RSVP. На теперішній момент топологія використовує LDP (останнє налаштування в ЛР№7).

Видаліть з топології існуючу там підтримку BGP VPLS і інших настроювань, що стосуються VPN рівня 2.

На всіх граничних маршрутизаторах провайдеру видаліть підтримку BGP VPLS:

```
interface vpls bgp-vpls remove [find]
```

Видаліть інтерфейси з мостів:

```
interface bridge port remove [find]
```

Видаліть мости A и B:

```
interface bridge remove A,B
```

Призначте відповідно до рис. 8.1, адреси для мереж 172.16.1.0/24, 172.16.2.0/24, 172.16.3.0/24 замовників A та B. На комп'ютерах (маршрутизаторах) замовників визначте маршрути за замовченням в бік мережі провайдера, наприклад для A1:

```
[admin@A1] > ip route add gateway=172.16.1.1
```

На граничних маршрутизаторах провайдера для BGP-пірів встановіть підтримку сімейства адрес VPNv4. Для PE1, PE2 и PE3:

```
routing bgp peer set 0 address-families=vpn4
```

Для відбивача маршрутів P1 є три BGP-піра:

```
routing bgp peer set 0,1,2 address-families=vpn4
```

Про встановлення BGP-з'єднань можна переконатися, переглянувши час існування з'єднання uptime в виводі команди `routing bgp peer print status.`

Інтерфейси граничних маршрутизвторів, які йдуть в сторону маршрутизаторів замовників, помістіть в дві VRF. Для VRF замовника A (B) призначте маркер маршрутів A (B).

```
[admin@PE1]>ip route vrf add routing-mark=A interfaces=ether3 route-
distinguisher=9.9.9.1:1 export-route-targets=1:1 import-route-argets=6:1,7:1
[admin@PE1]>ip route vrf add routing-mark=B interfaces=ether4 route-
distinguisher=9.9.9.1:2 export-route-targets=1:2 import-route-targets=6:2,7:2

[admin@PE2]>ip route vrf add routing-mark=A interfaces=ether3 route-
distinguisher=9.9.9.6:1 export-route-targets=6:1 import-route-targets=1:1,7:1
[admin@PE2]>ip route vrf add routing-mark=B interfaces=ether4 route-
distinguisher=9.9.9.6:2 export-route-targets=6:2 import-route-targets=1:2,7:2

[admin@PE3]>ip route vrf add routing-mark=A interfaces=ether3 route-
distinguisher=9.9.9.7:1 export-route-targets=7:1 import-route-targets=1:1,6:1
[admin@PE3]>ip route vrf add routing-mark=B interfaces=ether4 route-
distinguisher=9.9.9.7:2 export-route-targets=7:2 import-route-targets=1:2,6:2
```

Тут

route-distinguisher визначає значення, що прикріплюється до BGP-пакету; маршрутизатори, використовують це значення для заповнення таблиць VPNv4-маршрутизації й організації VPN. Щоб приймаючі маршрутизатори розрізняли інформацію від різних VPN, це значення повинне бути різне для різних VPN на одному пристрої. **Route-distinguisher** не використовується маршрутизатором для визначення приналежності передавального роутера конкретної VPN. Для цього використовується **route-targets.**

export-route-targets – це свого роду мітка (синонім) для route-distinguisher.

import-route-targets – це список із зовнішніх export-route-targets, який використовується для визначення сукупності маршрутизаторів, що утворять конкретну VPN.

Вкажіть BGP, що VRF, обумовлені маркерами маршрутів A і B, будуть брати участь у маршрутизації для сімейства адрес vpnv4.

```
[admin@PE1] > routing bgp instance vrf add routing-mark=A redistribute-
connected=yes
[admin@PE1] > routing bgp instance vrf add routing-mark=B redistribute-
connected=yes

[admin@PE2] > routing bgp instance vrf add routing-mark=A redistribute-
connected=yes
[admin@PE2] > routing bgp instance vrf add routing-mark=B redistribute-
connected=yes

[admin@PE3] > routing bgp instance vrf add routing-mark=A redistribute-
connected=yes
```

```
[admin@PE3] > routing bgp instance vrf add routing-mark=B redistribute-
connected=yes
```

Додайте IP-адреси на інтерфейси маршрутизаторів PE1, PE2, PE3, які направлені в сторону замовників.

Маршрутизатор PE1 отримає от маршрутизатора PE2 BGP-повідомлення Update (рис. 8.2), в якому для організації маршрутизації пакетів замовника A в бік мережі 172.16.2.0/24 призначається мітка 17. Зазначте, що route-distinguisher 9.9.9.6:1 для VPN замовника A в маршрутизаторі PE2 надано в повідомленні у вигляді 6.9.9.9:1. В повідомленні можна також побачити route-targets 6:1 VPN замовника A в маршрутизаторі PE2.

```
⊞ Frame 990: 163 bytes on wire (1304 bits), 163 bytes captured (1304 bits)
⊞ Ethernet II, Src: 00:ab:2b:93:23:00 (00:ab:2b:93:23:00), Dst: 00:ab:f8:16:ae:00 (00:ab:f8:16:ae:00)
⊞ Internet Protocol Version 4, Src: 9.9.9.2 (9.9.9.2), Dst: 9.9.9.1 (9.9.9.1)
⊞ Transmission Control Protocol, Src Port: 46437 (46437), Dst Port: bgp (179), Seq: 356, Ack: 231, Len: 97
⊟ Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 97
    Type: UPDATE Message (2)
    Unfeasible routes length: 0 bytes
    Total path attribute length: 74 bytes
  ⊟ Path attributes
    ⊞ ORIGIN: INCOMPLETE (4 bytes)
    ⊞ AS_PATH: empty (3 bytes)
    ⊞ LOCAL_PREF: 100 (7 bytes)
    ⊞ ORIGINATOR_ID: 7.7.7.1 (7 bytes)
    ⊞ CLUSTER_LIST: 7.7.7.1 (7 bytes)
    ⊟ EXTENDED_COMMUNITIES: (11 bytes)
      ⊞ Flags: 0xc0 (Optional, Transitive, Complete)
        Type code: EXTENDED_COMMUNITIES (16)
        Length: 8 bytes
      ⊟ Carried Extended communities
          two-octet AS specific Route Target: 6:1
    ⊟ MP_REACH_NLRI (35 bytes)
      ⊞ Flags: 0x80 (Optional, Non-transitive, Complete)
        Type code: MP_REACH_NLRI (14)
        Length: 32 bytes
        Address family: IPv4 (1)
        Subsequent address family identifier: Labeled VPN Unicast (128)
      ⊟ Next hop network address (12 bytes)
          Next hop: Empty Label Stack RD=0:0 IPv4=9.9.9.6 (12)
        Subnetwork points of attachment: 0
      ⊟ Network layer reachability information (15 bytes)
        ⊟ Label Stack=17 (bottom) RD=6.9.9.9:1, IPv4=172.16.2.0/24
            MP Reach NLRI Prefix length: 112
            MP Reach NLRI Label Stack: 17 (bottom)
            MP Reach NLRI Route Distinguisher: 6.9.9.9:1
            MP Reach NLRI IPv4 prefix: 172.16.2.0 (172.16.2.0)
```

Рисунок 8.2 – BGP-повідомлення Update

Аналогічні повідомлення отримають інші PE-маршрутизатори.

Подивіться на граничному маршрутизаторі PE1 маршрути з маркером A:

```
[admin@PE1] >  ip route print detail where routing-mark=A
```

```
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 0 ADC  dst-address=172.16.1.0/24 pref-src=172.16.1.1 gateway=ether3 gateway-status=ether3 reachable distance=0 scope=10
        routing-mark=A

 1 ADb  dst-address=172.16.2.0/24 gateway=9.9.9.6 gateway-status=9.9.9.6 recursive via 4.4.4.1,1.1.1.1 ether2,ether1
        distance=200 scope=40 target-scope=30 routing-mark=A bgp-local-pref=100 bgp-origin=incomplete
        bgp-ext-communities="RT:6:1"

 2 ADb  dst-address=172.16.3.0/24 gateway=9.9.9.7 gateway-status=9.9.9.7 recursive via 4.4.4.1 ether2 distance=200
        scope=40 target-scope=30 routing-mark=A bgp-local-pref=100 bgp-origin=incomplete bgp-ext-communities="RT:7:1"
```

Побачте наявність маршрутів в сторону всіх мереж 172.16.1.0/24, 172.16.2.0/24 та 172.16.3.0/24 комп'ютерів A1, A2 и A3 замовника A.

Подивіться на граничному маршрутизаторі PE1 маршрути з маркером B

```
[admin@PE1] > ip route print detail where routing-mark=B
```

```
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 3 ADC  dst-address=172.16.1.0/24 pref-src=172.16.1.1 gateway=ether5 gateway-status=ether5 reachable distance=0 scope=10
        routing-mark=B

 4 ADb  dst-address=172.16.2.0/24 gateway=9.9.9.6 gateway-status=9.9.9.6 recursive via 4.4.4.1,1.1.1.1 ether2,ether1
        distance=200 scope=40 target-scope=30 routing-mark=B bgp-local-pref=100 bgp-origin=incomplete
        bgp-ext-communities="RT:6:2"

 5 ADb  dst-address=172.16.3.0/24 gateway=9.9.9.7 gateway-status=9.9.9.7 recursive via 4.4.4.1 ether2 distance=200
        scope=40 target-scope=30 routing-mark=B bgp-local-pref=100 bgp-origin=incomplete bgp-ext-communities="RT:7:2"
```

Побачте наявність маршрутів в сторону всіх мереж 172.16.1.0/24, 172.16.2.0/24 и 172.16.3.0/24 комп'ютерів B1, B2 та B3 заказчика B.

Маршрут убік безпосередньо приєднаних мереж з однаковим префіксом 172.16.1.0/24 різний для різних замовників. Для замовника A він іде через інтерфейс ether3, а для замовника B він іде через інтерфейс ether5. Перегляд на граничних маршрутизаторах PE2 і PE3 маршрутів з маркерами A або B дасть аналогічний результат.

Одержали два незалежні VPN третього рівня для замовників A та B, у яких однакові адресні простори. Переконаєтеся в цьому за допомогою команди **/tool telnet**. Наприклад, зайдіть із A1 в A2, з A1 в A3, з A2 в A3, з B1 в B2, з B1 в A3 і з B2 в B3. Вихід із сесії telnet здійснюється за допомогою комбінації клавіш «Ctrl+D».

Вивчить, як MPLS-мітки допомагають організувати VPN. Виконайте пінги з сайту A1 убік сайту A2:

```
[admin@A1]> ping 172.16.2.2
```

За допомогою аналізатора пакетів Wireshark вивчимо структуру ICMP-пакетів на ынтерфесах e0 маршрутизатора PE1, e1 маршрутизатора P1 і e1 маршрутизатора P2 (рис. 8.3 – 8.5).

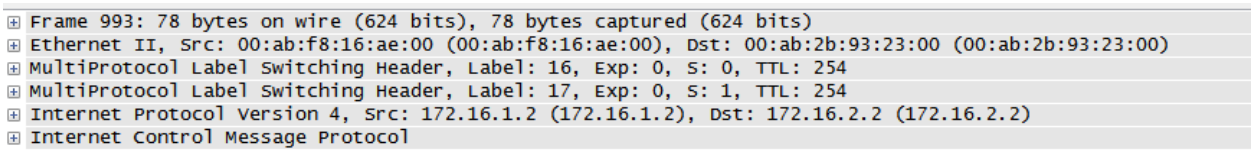

Рисунок 8.3 – Структура ICMP-пакету на ынтерфейсі e0 маршрутизатора PE1

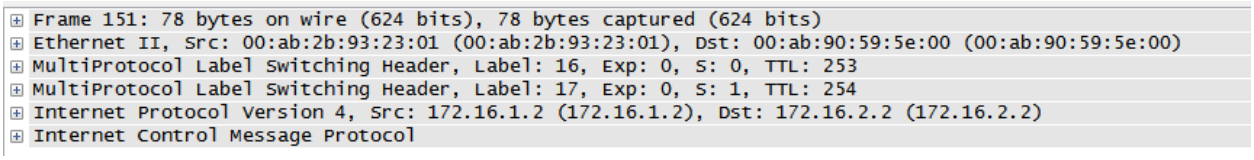

Рисунок 8.4 – Структура ICMP-пакету на інтерфейсі e1 маршрутизатора P1

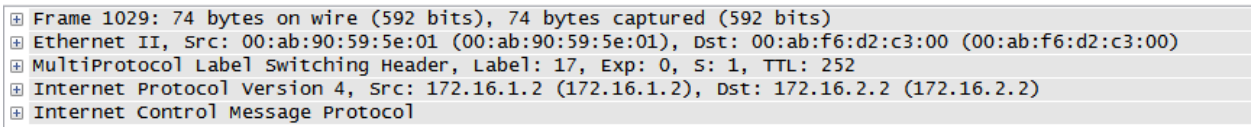

Рисунок 8.5 – Структура ICMP-пакету на інтерфейсі e1 маршрутизатора P2

Роз’яснемо отримане. Таблиця VPNv4-маршрутів на граничному маршрутизаторі PE1

```
[admin@PE1] > routing bgp vpnv4-route print
      Flags: L - label-present
# ROUTE-DISTINGUISHER    DST-ADDRESS  GATEWAY  INTERFACE  IN-LABEL  OUT-LABEL
 0 L 9.9.9.6:1     172.16.2.0/24      9.9.9.6   ether2      17         17
 1 L 9.9.9.7:1     172.16.3.0/24      9.9.9.7   ether2      17         17
 2 L 9.9.9.6:2     172.16.2.0/24      9.9.9.6   ether2      16         16
 3 L 9.9.9.7:2     172.16.3.0/24      9.9.9.7   ether2      16         16
 4 L 9.9.9.1:1     172.16.1.0/24                ether3      17
 5 L 9.9.9.1:2     172.16.1.0/24                ether5      16
```

показує, що для організації правильної маршрутизації пакетів замовника A (route-distinguisher = 9.9.9.6:1) убік мережі 172.16.2.0/24 їм призначена мітка 17. Цю мітку можна побачити на всіх 3-х рисунках. Для транспорту пакетів від A1 до A2 використовується мітка 16. Цю мітку можна побачити на рис. 8.3. і 8.4. Таблиця пробросів MPLS на PE1 показує, що пакети з міткою 17 будуть спрямовані на адресу 1.1.1.1 маршрутизатора P1.

```
[admin@PE1] > mpls forwarding-table print
Flags: L - ldp, V - vpls, T - traffic-eng
#   IN-LABEL      OUT-LABELS     DESTINATION    INTERFACE     NEXTHOP
4 L 19               16          9.9.9.6/32      ether1       1.1.1.1
```

Таблиця пробросів MPLS на P1 показує, що пакеты з вхідною міткою 16 отримують вихідну мітку 16 та будуть направлені на адресу 2.2.2.2 маршрутизатора P2. Цю мітку можно побачити на рис. 8.4.

```
[admin@P1] > mpls forwarding-table pr
Flags: L - ldp, V - vpls, T - traffic-eng
#   IN-LABEL      OUT-LABELS     DESTINATION    INTERFACE      NEXTHOP
1 L 16               16           9.9.9.6/32     ether2       2.2.2.2
```

Таблица VPNv4 маршрутів на граничному маршрутизаторі PE2

```
[admin@PE1] > routing bgp vpnv4-route print detail
Flags: L - label-present
4 L route-distinguisher=9.9.9.6:1 dst-address=172.16.2.0/24 interface=ether3 in-
label=17 bgp-ext-communities="RT:6:1"
```

та таблиця пробросів MPLS

```
[admin@PE2] > mpls forwarding-table print
Flags: L - ldp, V - vpls, T - traffic-eng
 #   IN-LABEL    OUT-LABELS DESTINATION
 2   17                   172.16.2.0/24@A
```

показує, що пакети з міткою 17 попадут в мережу 172.16.2.0/24 через інтерфейс ether3. Це означає, що вони попадуть на комп'ютер A2.

2. MPLS VPN 3-го рівня з організацією LSP за допомогою RSVP

Видаліть підтримку протокола LDP для всіх маршрутизаторів провайдера

```
mpls ldp set enabled=no transport-address=0.0.0.0 lsr-id=0.0.0.0
mpls ldp interface remove [find]
```

Для всіх LSR-маршрутизаторів застосуйте в налаштуванні протокола OSPF підтримку CSPF

```
routing ospf instance set 0 mpls-te-area=backbone mpls-te-router-id=lobridge
```

Вкажіть, що всі інтерфейси граничних маршрутизаторів, які направлені в бік мережі провайдера, будуть брати участь в роботе RSVP TE та заявіть пропускні здатності. Наприклад, інтерфейс ether1 граничного маршрутизатора PE1 йде в бік MPLS-облака

```
[admin@PE1]>mpls traffic-eng inter add interface=ether1 bandwidth=100000
```

Аналогічним чином налаштуйте усі інтерфейси маршрутизаторів, що знаходяться в MPLS-мережі.

На граничних маршрутизаторах для майбутніх RSVP TE-тунелей визначте динамічно шлях dyn за допомогою протокола CSPF:

```
mpls traffic-eng tunnel-path add use-cspf=yes name=dyn
```

Створіть RSVP TE-тунелі, створюючи повнозв'язну топологію для граничних маршрутизаторів:

```
[admin@PE1]>interface traffic-eng add from-address=9.9.9.1 to-address=9.9.9.6
bandwidth=100000 primary-path=dyn disabled=no record-route=yes name = 1to2
[admin@PE1]>interface traffic-eng add from-address=9.9.9.1 to-address=9.9.9.7
bandwidth=100000 primary-path=dyn disabled=no record-route=yes name = 1to3
[admin@PE2]>interface traffic-eng add from-address=9.9.9.6 to-address=9.9.9.1
bandwidth=100000 primary-path=dyn disabled=no record-route=yes name = 2to1
[admin@PE2]>interface traffic-eng add from-address=9.9.9.6 to-address=9.9.9.7
bandwidth=100000 primary-path=dyn disabled=no record-route=yes name = 2to3
[admin@PE3]>interface traffic-eng add from-address=9.9.9.7 to-address=9.9.9.1
bandwidth=100000 primary-path=dyn disabled=no record-route=yes name = 3to1
[admin@PE3]>interface traffic-eng add from-address=9.9.9.7 to-address=9.9.9.6
bandwidth=100000 primary-path=dyn disabled=no record-route=yes name = 3to2
```

Вивчить, як MPLS-мітки допомагають організувати VPN. Виконайте пінг з сайту A1 в бік сайта A2:

```
[admin@ A1]> ping 172.16.2.2
```

За допомогою аналізатора пакетів Wireshark проаналізуйте структуру ICMP-пакетів на інтерфесах e0 маршрутизатора PE1, e1 маршрутизатора P1 та e1 маршрутизатора P2 (рис 8.6 – 8.8).

Пояснемо отримане. Таблиця VPNv4-маршрутів на граничному маршрутизаторі PE1:

```
[admin@PE1] > routing bgp vpnv4-route print
# ROUTE-DISTINGUISHER  DST-ADDRESS   GATEWAY  INTERFACE  IN-LABEL  OUT-LABEL
0 L 9.9.9.6:1       172.16.2.0/24    9.9.9.6   ether2       16         16
2 L 9.9.9.6:2       172.16.2.0/24    9.9.9.6   ether2       17         17
```

показує, що для організації правильної маршрутизації пакетів замовника А (**route-distinguisher = 9.9.9.6:1**) в бік мережі 172.16.2.0/24 їм назначена мітка 16. Эту мітку можна побачити на всіх 3-х рисунках. Для транспорта пакетів від А1 до А2 використовується мітка 16. Цю мітку можна побачити на рис. 8.6. и на рис. 8.7.

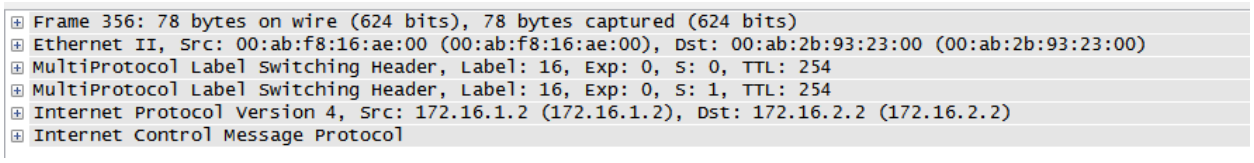

Рисунок 8.6 – Структура ICMP-пакету на інтерфейсі e0 маршрутизатора PE1

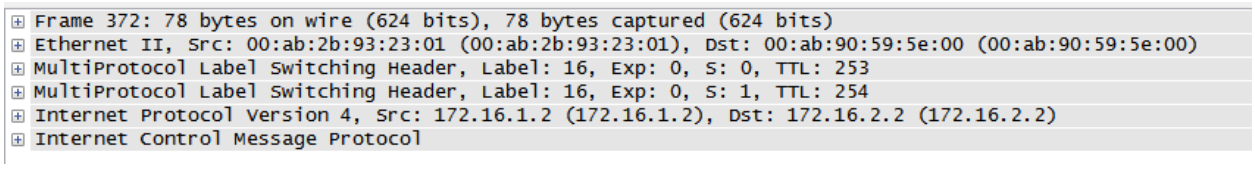

Рисунок 8.7 – Структура ICMP-пакету на інтерфейсі e1 маршрутизатора P1

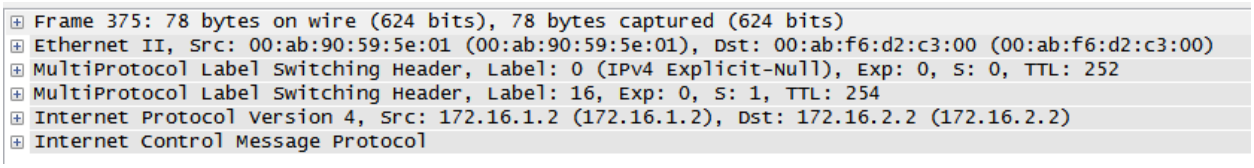

Рисунок 8.8 – Структура ICMP-пакету на інтерфейсі e1 маршрутизатора P2

Таблиця пробросів MPLS на P1 показує, що пакети з вхідною міткою 16 отримують таку ж вихідну мітку та будуть направлені на адресу 2.2.2.2 маршрутизатора P2. Цю мітку можна побачити на рис. 8.7.

```
[admin@P1] > mpls forwarding-table print
Flags: L - ldp, V - vpls, T - traffic-eng
 #    IN-LABEL    OUT-LABELS  DESTINATION          INTERFACE        NEXTHOP
 0    expl-null
 1 T 16             16       9.9.9.1:1->9.9.9.6:1     ether2        2.2.2.2
 2 T 17          expl-null   9.9.9.6:1->9.9.9.1:1     ether1        1.1.1.2
```

Таблиця пробросів MPLS на P2 показує, щоо пакет з вхідною міткой 16 отримують вихідну мітку 0 та будуть направлені на адресу 3.3.3.2 маршрутизатора P2. Цю мітку можна побачити на рис. 8.8.

```
[admin@P2] > mpls forwarding-table print
Flags: L - ldp, V - vpls, T - traffic-eng
 #    IN-LABEL    OUT-LABELS  DESTINATION          INTERFACE        NEXTHOP
 0    expl-null
```

```
 1 T 16          expl-null    9.9.9.1:1->9.9.9.6:1     ether2         3.3.3.2
 2 T 17            17         9.9.9.6:1->9.9.9.1:1     ether1         2.2.2.1
```

Таблиця VPNv4 маршрутов на граничному маршрутизаторі PE2

```
[admin@PE2] > routing bgp vpnv4-route print detail
```

```
4 L route-distinguisher=9.9.9.6:1 dst-address=172.16.2.0/24 interface=ether3 in-label=16 bgp-ext-communities="RT:6:1"

5 L route-distinguisher=9.9.9.6:2 dst-address=172.16.2.0/24 interface=ether5 in-label=17 bgp-ext-communities="RT:6:2"
```

та таблиця пробросів MPLS

```
[admin@PE2] > mpls forwarding-table print
Flags: L - ldp, V - vpls, T - traffic-eng
 #    IN-LABEL     OUT-LABELS       DESTINATION       INTERFACE          NEXTHOP
 0    expl-null
 1    16                          172.16.2.0/24@A
 2    17                          172.16.2.0/24@B
```

показують, що пакети з міткою 16 птрапляють в мережу 172.16.2.0/24 через інтерфейс ether3. Це означає, що вони попадуть на комп'ютер A2, а пакети з міткою 17 попадут в мережу 172.16.2.0/24 через інтерфейс ether5, тобто на комп'ютер B2.

На рис. 8.9 – 8.11 наведені структури ICMP-пакетів на інтерфесах e0 маршрутизатора PE1, e1 маршрутизатора P1 та e1 маршрутизатора P2 при виконанні команди

```
[admin@B1]> ping 172.16.2.2
```

```
⊞ Frame 368: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
⊞ Ethernet II, Src: 00:ab:f8:16:ae:00 (00:ab:f8:16:ae:00), Dst: 00:ab:2b:93:23:00 (00:ab:2b:93:23:00)
⊞ MultiProtocol Label Switching Header, Label: 16, Exp: 0, S: 0, TTL: 254
⊞ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 1, TTL: 254
⊞ Internet Protocol Version 4, Src: 172.16.1.2 (172.16.1.2), Dst: 172.16.2.2 (172.16.2.2)
⊞ Internet Control Message Protocol
```

Рисунок 8.9 – Структура ICMP-пакету на інтерфейсі e0 маршрутизатора PE1.

```
⊞ Frame 388: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
⊞ Ethernet II, Src: 00:ab:2b:93:23:01 (00:ab:2b:93:23:01), Dst: 00:ab:90:59:5e:00 (00:ab:90:59:5e:00)
⊞ MultiProtocol Label Switching Header, Label: 16, Exp: 0, S: 0, TTL: 253
⊞ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 1, TTL: 254
⊞ Internet Protocol Version 4, Src: 172.16.1.2 (172.16.1.2), Dst: 172.16.2.2 (172.16.2.2)
⊞ Internet Control Message Protocol
```

Рисунок 8.10 – Структура ICMP-пакету на інтерфейсі e1 маршрутизатора P1

```
⊞ Frame 391: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
⊞ Ethernet II, Src: 00:ab:90:59:5e:01 (00:ab:90:59:5e:01), Dst: 00:ab:f6:d2:c3:00 (00:ab:f6:d2:c3:00)
⊞ MultiProtocol Label Switching Header, Label: 0 (IPv4 Explicit-Null), Exp: 0, S: 0, TTL: 252
⊞ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 1, TTL: 254
⊞ Internet Protocol Version 4, Src: 172.16.1.2 (172.16.1.2), Dst: 172.16.2.2 (172.16.2.2)
⊞ Internet Control Message Protocol
```

Рисунок 8.11 – Структура ICMP-пакету на інтерфейсі e1 маршрутизатора P2

Як можна бачити із рис. 8.9 – 8.11 структура пакетів відрізняється тільки внутрішньою міткой (=17) для правильної маршрутизації трафіка між вузлами B1 та B2.

Зміст звіту

Звіт готується в електронному вигляді і роздруковується. Звіт містить усі виконані пункти практичної роботи зі скріншотами.

Порядок виконання і здачі роботи

1. Вивчити теоретичну і практичну частину.
2. Здати викладачеві теорію роботи, відповівши на контрольні питання.
3. Виконати практичну частину.
4. Захистити звіт.

Контрольні питання

1. Для чого використовується технологія MPLS L3VPN?
2. Що таке VRF та для чого використовується?
3. Яку опцію необхідно застосувати в налаштуванні BPG для підтримки VRF?
4. Для чого використовується route-distinguisher?
5. Як організується MPLS L3VPN з LSP на основі LDP?
6. Як організується MPLS L3VPN з LSP на основі RSVP?

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Dordal Peter Lars Peter. An Introduction to Computer Networks / Peter L. Dordal, Loyola University Chicago, 2022. – 951 p.

2. Peterson Larry, Davie Bruce. Read more about Computer Networks: A Systems Approach – Peterson and Davie, 2019. – 485 p.

3. Жураковський Б. Ю. Комп'ютерні мережі. Навчальний посібник [Електронний ресурс]/ Б.Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 8,6 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 336 с.

4. Windows Server Documentation / Microsoft Docs. Developer tools, technical documentation and coding examples | Microsoft Docs.

5. Beej's Guide to Network Programming Using Internet Sockets. Brian "Beej Jorgensen" Hall. Copyright © April 8, 2023

6. https://learn.microsoft.com/en-us/windows/win32/apiindex/api-index-portal.

7. Кеньо Г.В. Моделювання розумного будинку в середовищі Cisco Packet Tracer / Г.В. Кеньо, В.В. Хома. – Львів: Львівська політехніка, 2022. – 104 с.

ЗМІСТ

Навчальне видання

Методичні вказівки до виконання лабораторних робіт
з навчальної дисципліни «Проєктування корпоративних мереж»
для студентів денної та заочної форм навчання
спеціальності 123 – «Комп'ютерна інженерія»

Укладач: МЕЗЕНЦЕВ Микола Вікторович

Відповідальний за випуск проф. Заковоротний О.Ю.
Роботу до видання рекомендував проф. Дмитрієнко В.Д.

В авторській редакції

План 2024 р., поз. 131

Підп. до друку 15.02.2024.

Видавничий центр НТУ «ХПІ»
61002, Харків, вул. Кирпичова, 2.
Свідоцтво суб'єкта видавничої справи ДК № 5478 від 21.08.2017 р.

Електрона версія