МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРНІГВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ
КАФЕДРА ІНФОРМАЦІЙНИХ ТА КОМП'ЮТЕРНИХ СИСТЕМ

АНАЛІЗ ФУНКІОНУВАННЯ ЛОКАЛЬНИХ ОБЧИСЛЮВАЛЬНИХ МЕРЕЖ

Методичні вказівки
до виконання самостійних робіт
з дисципліни «Комп'ютерні мережі»
для студентів спеціальності
123 "Комп'ютерна інженерія"

Затверджено
на засіданні кафедри
інформаційних і комп'ютерних систем

Протокол № 1 від «27» серпня 2018 р.

Чернігів ЧНТУ 2018

Аналіз функціонування локальних обчислюваних мереж. Методичні вказівки до виконання самостійних робіт з дисципліни „Комп’ютерні мережі” для студентів спеціальності 123 "Комп'ютерна інженерія"/ Укл. Риндич Є. В., Зайцев С.В., Нікітенко Є.В. – Чернігів: ЧНТУ, 2018. – 47 с.

Укладачі: Риндич Євген Володимирович, кандидат технічних наук, доцент, доцент кафедри інформаційних та комп’ютерних систем;
Зайцев Сергій Васильович, доктор технічних наук, доцент, завідувач кафедри інформаційних та комп’ютерних систем;
Нікітенко Євгеній Васильович, кандидат фізико-математичних наук, доцент, доцент кафедри інформаційних та комп’ютерних систем

Відповідальний за випуск: С.В. Зайцев, зав. кафедрою інформаційних та комп’ютерних систем, д-р. техн. наук, доцент.

Рецензент: С. О. Нестеренко, канд. техн. наук, доцент, доцент кафедри інформаційних і комп'ютерних систем Чернігівського національного технологічного університету

ЗМІСТ

ВСТУП

Міжмережевий протокол IP на сьогоднішній день переважає як в локальних, так і в глобальних мережах. З розвитком мережі Інтернет сек протоколів TCP/IP «обріс» великою кількістю мережевих сервісів, що і обумовило домінацію даного сімейства протоколів в усіх різновидах мереж.

Дане керівництво призначене для початкового знайомства зі стеком протоколів TCP/IP, а також із взаємодією міжмережевого протоколу з несучими мережами, в тому числі, з мережами, побудованими за технологією Ethernet.

Цикл самостійних робіт, котрий запропоновано в даному посібнику допоможе студентам засвоїти базові навики з конфігурування мережевих інтерфейсів та пристроїв, діагностики мережі, роботі з мережевими аналізаторами.

1 Самостійна робота №1. Налаштування мережевих інтерфейсів та тестування роботи локальної мережі.

1.1 Мета роботи

Познайомитися з особливостями реалізації мережі Ethernet, ознайомитися з її характеристиками. Отримати практичні навички конфігурування мережевих інтерфейсів в різних ОС.

1.2 Короткі теоретичні відомості

Логічний пристрій, який здійснює передачу даних в мережі на логічному рівні називається мережевим інтерфейсом. Мережевий інтерфейс може бути зв'язаним з певним фізичним пристроєм, наприклад, адаптером Ethternet або послідовним портом, а може бути просто логічним інтерфейсом, наприклад, інтерфейс локального зворотного зв'язку localhost.

Мережеві інтерфейси поділяються на 2 основних типи: інтерфейс в широкомовну мережу, broadcast, наприклад, Ethternet та інтерфейс «точка-точка», point-to-pint. Перші асоціюються з одною мережевою адресою, другі – з двома, локальною та віддаленою адресою. Мережевий інтерфейс можк використовуватися для передачі даних по різноманітним протоколам: v.4, IP v.6, IPX, AppleTalk і т.д. Найбільше поширення на сьогодні здобув протокол IP v.4 (далі – IP), оскільки він використовується в Інтернеті. В ОС Windows можливе використання протоколу NetBIOS, однак, з точки зору «прозорості» мережі рекомендується використовувати як основний протокол IP, з котрим асоціюються служби протоколу NetBIOS. В ОС UNIX використовується додаткові програми для організації сервісів протоколу NetBIOS (пакет Samba) поверх IP. Далі будемо використовувати тільки проокол IP.

Мережеві інтерфейси можуть мати псевдоніми, тобто з інтерфейсом можуть асоціюватися декілька мережевих адрес. Псевдоніми дозволяють використовувати різноманітні схеми адресації на одному сегменті мережі. Наприклад, можливо одночасне використання приватних та публічних IP адрес.

Для конфігурування мережевих інтерфейсів в POSIX-сумісних ОС використовується команда ifconfig. Команда ifconfig без параметрів видає поточну конфігурацію активних мережевих інтерфейсів.

1.2.1 Packet Tracer

Cisco Packet Tracer розроблений компанією Cisco і рекомендований використовуватися при вивченні телекомунікаційних мереж і мережевого устаткування, а також для проведення уроків з лабораторних та самостійних робіт у вищих закладах.

Основні можливості Packet Tracer:

- Дружній графічний інтерфейс (GUI), що сприяє до кращого розуміння організації мережі, принципів роботи пристрою;
- Можливість змоделювати логічну топологію: робочий простір для того, щоб створити мережі будь-якого розміру на CCNA-рівні складності;

- моделювання в режимі real-time (реального часу);
- режим симуляції;
- Багатомовність інтерфейсу програми: що дозволяє вивчати програму на своїй рідній мові.
- вдосконалене зображення мережевого обладнання зі здатністю додавати / видаляти різні компоненти;
- наявність Activity Wizard дозволяє мережевим інженерам, студентам і викладачам створювати шаблони мереж і використовувати їх в подальшому.
- проектування фізичної топології: доступне взаємодія з фізичними пристроями, використовуючи такі поняття як місто, будівля, стійка і т.д.

Широке коло можливостей даного продукту дозволяє мережевим інженерам: конфігурувати, налагоджувати і будувати обчислювальну мережу. Також даний продукт незамінний в навчальному процесі, оскільки дає наочне відображення роботи мережі, що підвищує освоєння матеріалу учнями.

Емулятор мережі дозволяє мережевим інженерам проектувати мережі будь-якої складності, створюючи і відправляючи різні пакети даних, зберігати і коментувати свою роботу. Фахівці можуть вивчати і використовувати такі мережеві пристрої, як комутатори другого і третього рівнів, робочі станції, визначати типи зв'язків між ними і з'єднувати їх.

Вибір необхідного елементу мережі необхідно робити в нижній панелі робочого вікна (рисунок 1.1). На панелі є групи приладів і їхні модифікації.

Рисунок 1.1 – Вибір необхідного елементу мережі

Типи кабелів для з’єднання знаходяться на цій же панелі в закладці «Connections». Тут можна вибрати для з’єднання кабель, який необхідний (рисунок 1.2).

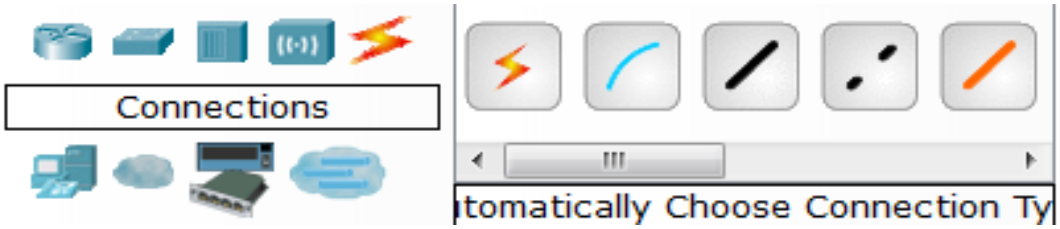


Рисунок 1.2 – Вибір необхідного типу кабеля

Для створення мережі використано 2 сервери: DNS-DHSP і mail, комутатор Cisco Catalyst 2960, 10 робочих станцій.

Елементи потрібно з’єднати, як показано на рисунку 1.3.

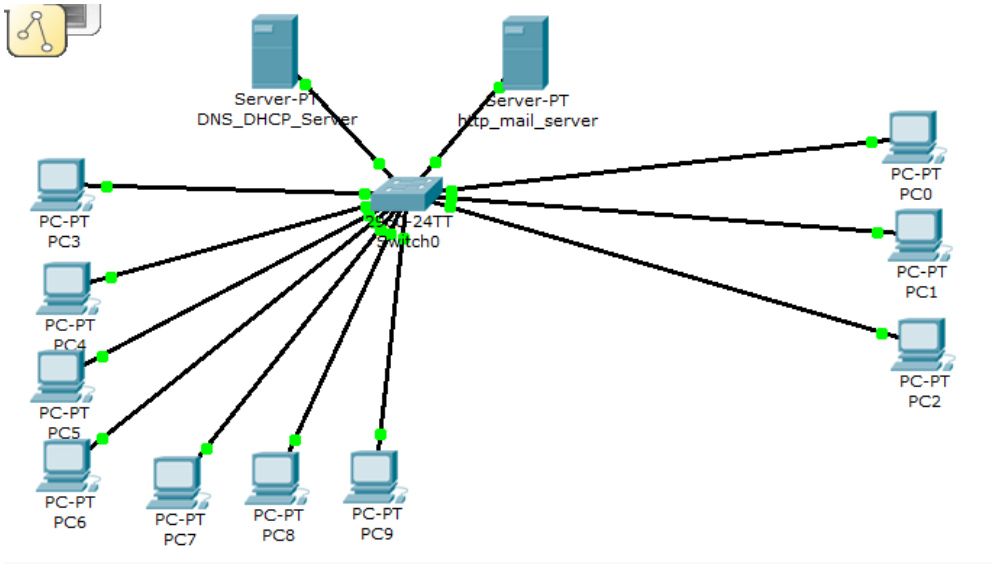


Рисунок 1.3 – Топологія створеної мережі

Під'єднані до мережі хости, сервери або інші елементи потрібно налаштувати. IP-адреси для компонентів можна роздавати автоматично або присвоювати кожному елементу окремо.

Для налагодження хоста необхідно клацнути два рази на ньому лівою клавішею мишки. В відкритому вікні перейти на закладку «Config» і вибрати тип отримання ip-адреси (рисунок 1.4).

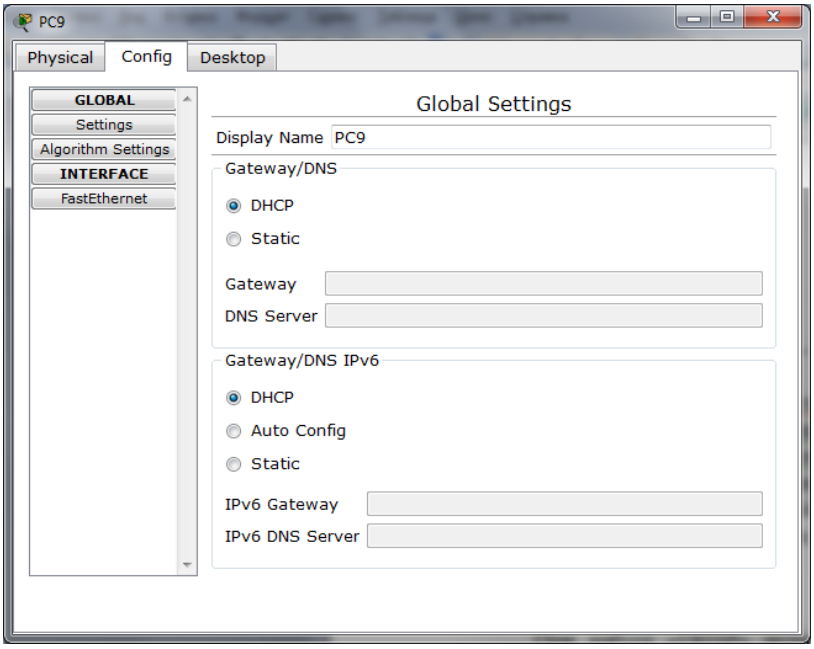


Рисунок 1.4 – Динамічне або статичне отримання адреси для хостів

При виборі «DHSP» ардесу хосту буде присвоєно автоматично в залежності від налаштування DHSP-серверу.

Для статичного отримання адреси необхідно відмітити поле «Static», написати gateway-адресу (необхідна для зв'язування мереж, зазвичай одна з вільних адрес даної підмережі) і DNS-сервер. Після цього перейти на закладку

«Desktop», вибрати «IP Cofiguration», заповнити всі поля в відкритому вікні (рисунок 1.5).

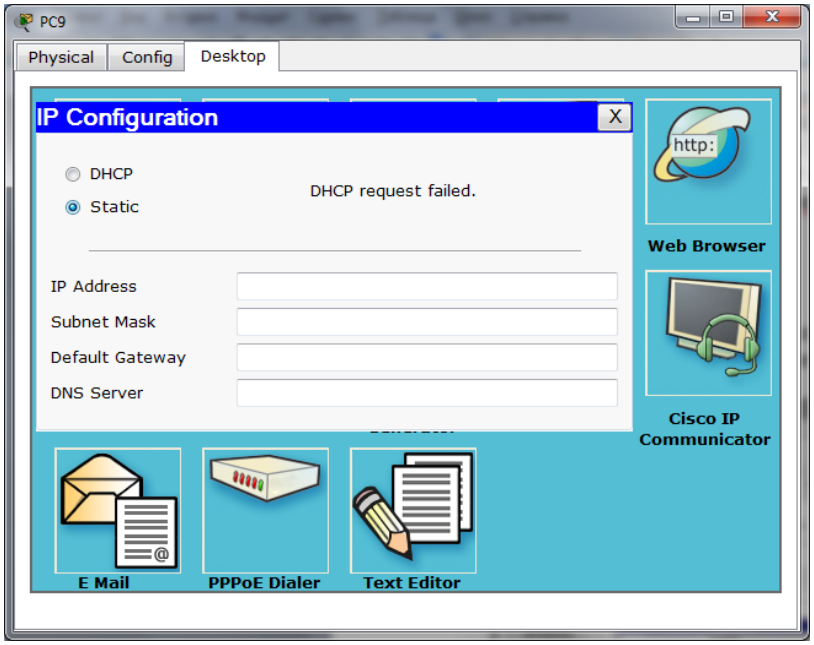


Рисунок 1.5 – Вікно для задання IP-адреси, маски, шлюзу, DNS-серверу для хоста

Також потрібно налаштувати DNS і DHCP сервери, дати їм назви і IP-адреси. Налаштування показане на рисунках 1.6 та 1.7.

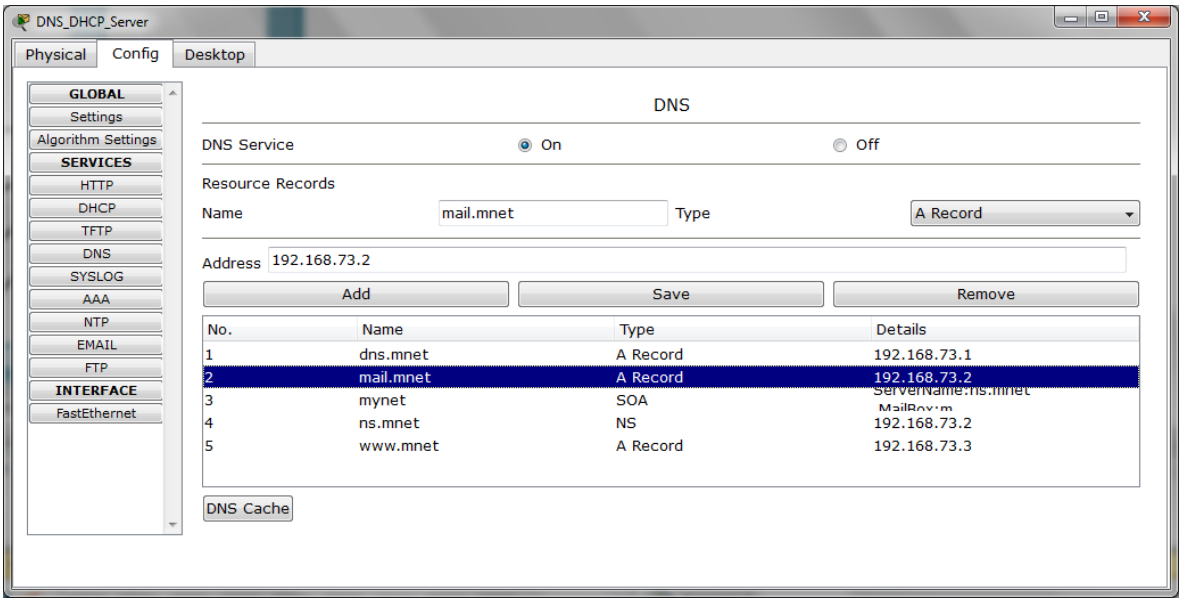


Рисунок 1.6 – Налаштування DNS-сервера мережі

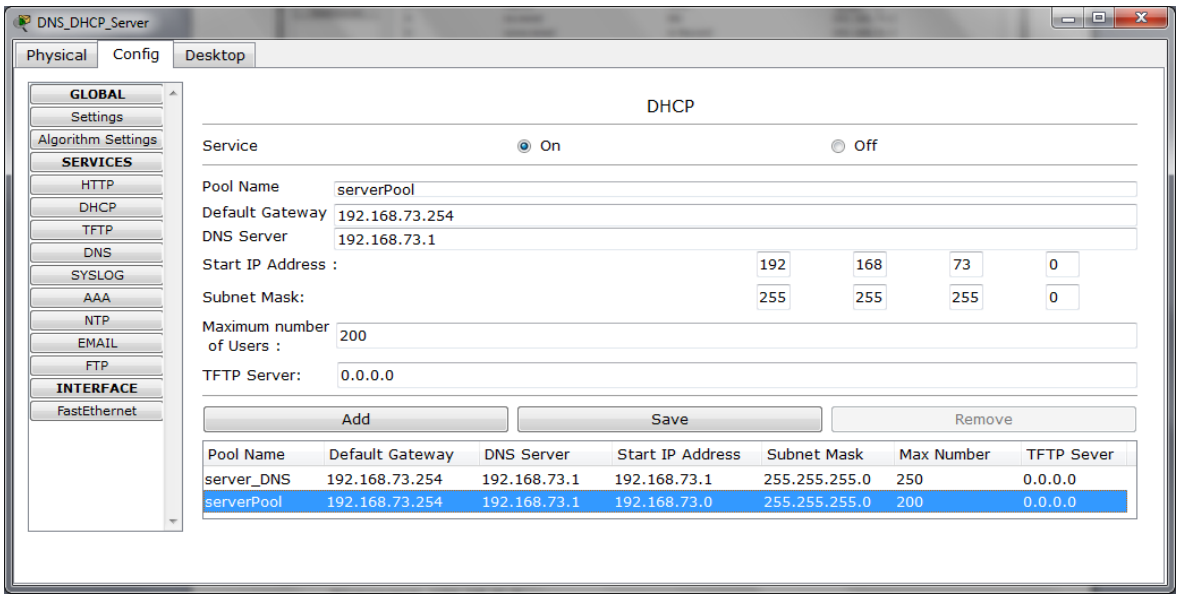


Рисунок 1.7 – Налаштування DHCP-сервера мережі

В результаті виконання команди ping на один із хостів, отриманий вивід:

```
Packet Tracer SERVER Command Line 1.0
SERVER>ping 192.168.73.7
Pinging 192.168.73.7 with 32 bytes of data:
Reply from 192.168.73.7: bytes=32 time=10ms TTL=128
Reply from 192.168.73.7: bytes=32 time=10ms TTL=128
Reply from 192.168.73.7: bytes=32 time=12ms TTL=128
Reply from 192.168.73.7: bytes=32 time=12ms TTL=128
Ping statistics for 192.168.73.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 10ms, Maximum = 12ms, Average = 11ms
```

Результати виконання команди tracert для обох серверів показані нижче:

```
PC>traert 192.168.73.1
Invalid Command.
PC>tracert 192.168.73.1
Tracing route to 192.168.73.1 over a maximum of 30 hops:
  1    8 ms      8 ms      6 ms      192.168.73.1
Trace complete.
PC>tracert 192.168.73.2
Tracing route to 192.168.73.2 over a maximum of 30 hops:
  1    7 ms      7 ms      6 ms      192.168.73.2
Trace complete.
```

Перевірку роботи мережі можна також за допомогою елементів (рисунок 1.8) на боковій панелі.

Рисунок 1.8 – Елементи для перевірки передачі пакетів

У панелі на рисунку 1.9 буде зображено виконання або невиконання передачі пакетів.

Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num	Edit	Delete
	Successful	PC1	192.168.73.3	ICMP		100.000	N	0	(edit)	(delete)
	Successful	PC0	DNS_DHCP_Server	ICMP		0.000	N	1	(edit)	(delete)
	Successful	DNS_DHCP_Server	http_mail_server	ICMP		0.000	N	2	(edit)	(delete)
	Successful	PC5	DNS_DHCP_Server	ICMP		0.000	N	3	(edit)	(delete)

Рисунок 1.9– Моделювання передачі пакетів серверів та хостів

1.3 Хід роботи

1. Дослідити налаштування мережевого адаптера Win/Linux (вікна ipconfig, ifconfig).
2. Визначити реальну пропускну спроможність мережі Ethernet з використанням комутатора: Провести тестування мережі (ping, traceroute/tracert, route, netstat).
3. Пошук відмінностей в роботі (ifconfig, netstat) та iproute (ip, nstat, ss).
4. Дослідження топології мережі: розглянути топологію локальної мережі поверха, де знаходиться лабораторія. Побудувати модель мережі в Packet Tracer (задати всі параметри та перевірити роботу моделі мережі).
5. Зробити висновки.

1.4 Зміст звіту

1. Опис мережевого адаптеру (технічні характеристики та стан перемикачів).
2. Способи об’єднання сегментів.
3. Аналіз отриманих графічних залежностей пропускної здатності мережі від числа робочих станцій.
4. Графіки та розрахунки пропускної здатності.
5. Схема та аналіз ЛВС ІКС ЧНТУ.
6. Результати та опис виконаних команд.

1.5 Контрольні запитання

1. Які параметри мережевого інтерфейсу свідчать про проблеми в фізичному середовищі передачі даних?
2. Які параметри інтерфейсу свідчать про надмірне завантаження мережі?
3. Яке середнє значення затримки в мережі 100 Мбіт / с?

4. Чим визначається час затримки в мережі Ethernet?
5. Чому в стандарті 100baseT довжина некомутованого сегмента не повинна перевищувати 100 метрів? Підкріпіть ваше твердження розрахунками часу поширення 1 біта інформації і одного кадру інформації.
6. Перерахуйте основні параметри команди ifconfig при конфігуруванні широкомовного мережевого інтерфейсу.
7. Перерахуйте основні параметри команди ifconfig при конфігуруванні мережевого інтерфейсу точка-точка.
8. Опишіть процес конфігурації мережевого інтерфейсу в ОС RedHat Linux при завантаженні.
9. Як додати мережевий інтерфейс? Як додати псевдонім для мережевого інтерфейсу?
10. Який час займає один пакет даних в мережі Ethernet 100BaseT?
11. Чому в некомутовані мережах Ethernet максимальна довжина кабелю не повинна перевищувати 100м?
12. При якому рівні завантаження мережі Ethernet відсоток втрат зростає до критичного?
13. Скільки часу потрібно спостерігати мережу, що б оцінити рівень втрат пакетів?
14. У якому випадку в каналі допустимі значні (> 250ms) затримки?
15. Як знизити завантаження мережі службовими даними?
16. Який середній відсоток "корисних" даних в мережевому трафіку ви спостерігали при перекачуванні файлу по протоколу FTP?
17. Поясніть, навіщо в ході експериментів використовувалися 3 хоста.
18. Є комутатор на 16 портів 100BaseT із загальною пропускною спроможністю коммутирующей матриці 1.2 Гбіт / с. Оцініть якість даного комутатора.
19. Поясніть, чому довжина кабелю в комутованому сегменті Ethernet 100BaseT може перевищувати встановлені стандартом 100 метрів при збереженні задовільної роботи мережі. На що впливає надмірна довжина кабелю?

2 Самостійна робота № 2. Моделювання локальної мережі. Робота з VLAN.

2.1 Мета роботи

Вивчити на практиці основи побудови віртуальних мереж (VLAN), використовуючи навики мережевої арифметики, навчитися розбивати мережу на підмережі довільних масок.

2.2 Короткі теоретичні відомості

2.2.1 VLAN

VLAN (аббр. від англ. Virtual Local Area Network) — віртуальна локальна комп'ютерна мережа, є групою хостів із загальним набором вимог, які взаємодіють так, як якби вони були підключені до широкомовного домена, незалежно від їх фізичного місцезнаходження. VLAN має ті ж властивості, що і фізична локальна мережа, але дозволяє кінцевим станціям групуватися разом, навіть якщо вони не знаходяться в одній фізичній мережі. Така реорганізація може бути зроблена на основі програмного забезпечення замість фізичного переміщення пристроїв.

Переваги:

1. Полегшується переміщення, додавання пристроїв і зміна їх з'єднань один з одним.
2. Досягається велика міра адміністративного контролю унаслідок наявності пристрою, що здійснює між мережами VLAN маршрутизацію на 3-м-коді рівні.
3. Зменшується вжиток смуги пропускання в порівнянні з ситуацією одного широкомовного домена.
4. Скорочується невиробниче використання CPU за рахунок скорочення пересилки широкомовних повідомлень.
5. Запобігання широкомовним штормам і запобігання петлям.

2.2.2 Структура IP-адреси

Схема маршрутизації повідомлень в TCP/IP базується на унікальних адресах, названих *адресами Internet* або *IP-адресами*, які утворюють пару:

<адреса локальної мережі, адреса вузла в локальній мережі> або *(<NetID, HostID>)*

IP-адреси представлені 32-бітовим кодом і діляться на класи: A, B, C, D, E (табл. 2.1). Найбільше використання на даний час мають перші 3 класи.

Таблиця 2.1 – IP-адреси

	1	2	3	4	5	6	7	8	9				…	16						….	24						…	32
Клас **A**	0 0	Адреса мережі NetID (7 біт)							Адреса вузла HostID (24 біти)																			
Клас **B**	0 1	0 0	Адреса мережі NetID (14 біт)												Адреса вузла HostID (16 біт)													
Клас **C**	1 1	1 1	0 0	Адреса мережі NetID (21 біт)																				Адреса вузла HostID(8 біт)				
Клас **D**	1 1	1 1	1 1	0 0	Багатоадресна MulticastGroupID (28 біт)																							
Клас **E**	1 1	1 1	1 1	1 1	1 0	Зарезервовано для майбутніх застосувань (27 біт)																						

- **Клас A** - адреса починається з **0**
- **Клас B** - адреса починається з **10**
- **Клас C** - адреса починається з **110**
- **Клас D** - адреса починається з **1110**
- **Клас E** - адреса починається з **1111**

Розподіл кількості адрес мереж та вузлів у різних класах показано в таблиці 2.2.

Таблиця 2.2 – Розподіл кількості адрес у класах

	Число мереж N_d (domains)	Число вузлів N_n (hosts)
Клас **A**	2^8 -2	2^{24}-2
Клас **B**	2^{14} -2	2^{16}-2
Клас **C**	2^{21}-2	2^8 -2

Коли комп’ютер має два або більше фізичних під’єднань він називається multi-home host. Такий комп’ютер або маршрутизатор потребує множину IP-адрес, при цьому кожна адреса відповідає одному з під’єднань машини до мережі.

Оскільки IP-адреси кодують як мережу, так і вузол в мережі, то ці адреси не визначають конкретний комп’ютер, а визначають під’єднання до мережі.

Тому роутер, який з’єднує n мереж, має n різних IP-адрес, по одній для кожного мережевого під’єднання.

2.2.3 Десятковий запис IP-адреси

IP-адреса має чотири поля (байти) у формі aaa.bbb.ccc.ddd, розділених крапками (таблиця 2.3). Кожне поле звичайно подається у формі десяткового числа. IP-адреси можна розрізняти за класами, використовуючи десяткове значення aaa першого байта:

Таблиця 2.3. – Розподіл IP адрес за класами

Клас	Найменша адреса	Найбільша адреса
A	0.1.0.0	126.0.0.0
B	128.0.0.0	191.255.0.0
C	192.0.1.0	223.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

2.2.4 Мережі та підмережі. Маски

В оригінальній схемі IP-адресації будь-якій фізичній мережі призначена унікальна мережева адреса; будь-який вузол в мережі використовує мережеву адресу як префікс до індивідуальної адреси вузла. Такий поділ обумовлений потребами процесу маршрутизації пакетів. Окремі територіальні мережі мають певну свободу в модифікації адрес і маршрутів, що дозволяє розширити кількість адрес.

В багатьох випадках, наприклад з метою зниження трафіка, чи для організації робочих груп, проявляється необхідність розбиття на підмережі або сегменти. Здійснюється таке розбиття за допомогою масок підмереж. Це призводить до зниження кількості вузлів в мережі, а також спрощує адресацію між ними за рахунок скорочення кількості біт, що залишаються для визначення адреси хоста.

Додатково комп'ютер має знати, скільки біт відведено для SubNetID та HostID. Саме за допомогою маски є можливість вказати розмір цих полів. Маска - 32-розрядне число, що має біти, які відповідають полям NetID та SubNetID, рівні 1, а біти для HostID рівні 0. Адреса підмережі визначається шляхом логічного множення:

<Адреса підмережі> = <IP-адреса & маска>

Розглянемо сегментацію мереж IP на прикладі мережі класу C. Організація пімереж в цьому випадку виконується за допомогою "позичення" для адресації мережі декількох біт з останнього октета. Кількість позичених біт залежить від потрібної кількості підмереж або від обмеження щодо кількості вузлів в підмережі.

У випадку розбиття на дві підмережі відповідно відповідно позичаються 2 біти, залишаючи 6 біт для адресації хостів. Ці два біти з останнього октета будуть додані до бітів, що використовуються для адресації мережі. Шість біт, що залишились, дозволяють кожній з двох підмереж підтримувати 62 унікальних адреси (64 адреси в піжмережі, але перша і остання виділені на адресу мережі і адресу бродкасту відповідно), а маска підмережі, що використовується, буде виглядати як 255. 255. 255.192(11111111.11111111.11111111.11000000).

Для організації шести підмереж потрібно використовувати три біти, що обмежує кількість вузлів в кожній мережі до тридцяти і з маскою 255.255.255.224 (11111111.11111111.11111111.11100000*)*

Легко зауважити, що розбиття на більшу кількість підмереж призведе до різкого зниження кількості доступних адрес в підмережі.

Досить часто використовують запис адресації виду XXXX.XXXX.XXXX.0/**Y**

Число, що стоїть за дробом, визначає кількість біт, позичених для визначення адреси мережі, що дозволяє відмовитись від стандартної форми запису. Таким чином, адреса класу С 204. 251. 122.0 з маскою 255.255.255.224 може бути записана як 204.251.122.0/27, що означає використання 27 з 32 адресних біт для визначення адреси мережі, залишаючи решту адресного простору для призначення адрес хостам.

Це число називають номером CIDR (Classless InterDomain Routing)

Маска також часто записується у шістнадцятковій формі, особливо тоді, коли приходиться маніпулювати SubNetID з розміром, не кратним 8 бітам. Вищеприведена маска у шістнадцятковій формі записується так: 0xFFFFFF00.

Маючи IP адресу і маску, комп'ютер може визначити чи IP адреса вказує:

- на комп'ютер який знаходиться на його ж підмережі;
- на комп'ютер який знаходиться на іншій підмережі;
- на комп'ютер який знаходиться на іншій мережі.

Наприклад адреса комп'ютера 184.12.44.45 (клас В), а його маска рівна 255.255.255.0 (тобто для SubNetID виділено 8 біт). Якщо комп'ютеру необхідно передати інформацію, призначену для іншого комп'ютера, IP-адреса якого рівна: 184.12.80.2, то комп'ютер може визначити що їх NetID однакові, а от SubNetID різні (44 ≠ 80).

- адреса 184.12.44.50, то комп'ютер може визачити, що вони належать як до одної мережі, так і до одної підмережі, оскільки їх NetID та SubNetID однакові;
- адреса 192.168.0.3 (адреса класу С) - оскільки NetID різні, то подальші уточнення не проводяться.

Ці порівняння необхідні наприклад для того щоб можна було визначитися, чи комп'ютер повинен посилати пакети до призначення прямо на мережу (підмережу), до якої він безпосередної під'єднаний, чи до маршрутизатора, який асоціюється з необхідним напрямком.

2.2.5 Організація підмереж

Сам Internet не бачить організації підмереж, так що організація підмереж відома і розпізнається тільки локально всередині загальної мережі. Однак будучи один раз утворена, кожна підмережа локально діє як окрема мережа, і комунікація між підмережами вимагає того ж, що й комунікація між мережами. Комп'ютери в різних підмережах не можуть бачити один одного, доки не передбачено спеціального способу для цього. Очевидно, що 16777214 адрес станцій мережі класу А незручні для використання, як і 65534 адрес класу В. Звичайно неможливо використати такий розмір мережі, однак існує простий спосіб організації підмереж в мережах класів А і В: підмереж класу А у вигляді еквівалентних мереж класу В і підмереж класу В у еквівалентні мережі класу С.

Зауважимо ще раз, що організація підмереж є тільки внутрішньою, а зовні мережа класу А завжди залишається такою.

Використаємо для прикладу мережу класу С і розглянемо, як можна утворити підмережу. Нехай мережева адреса є 192.168.255.0 і мережева маска для неї є 255.255.255.0. Маємо 8 бітів для адрес станцій, що дає можливих 254 адреси. Нагадаємо, що адреси станцій з усіма двійковими одиницями (тобто 255) або з усімома нулями (тобто 0) не можна застосовувати. Для маски можна призначити будь-які з бітів, зарезервованих в мережевій адресі класу С для станції, тобто від 1 до 6 бітів, однак не можна вживати 7 бітів, бо це означатиме 0 станцій. Зауважимо, що перші три 255.255.255 не змінюються. Доброю практикою при організації підмереж є виділення бітів у мережевій масці неперервно зліва направо. Це не вимога, однак впровадження інших варіантів не дає добрих результатів. Використання 2 та 3 адресних біт для утворення підмереж наведені в таблицях 2.4-2.5.

Таблиця 2.4 –Використання 2 адресних біт для утворення підмереж

2 біти Мережева маска 255.255.255.192	Адреси станцій
підмережа 0: $(00)_2$	від 192.168.255.1 до 192.168.255.62
підмережа 1: $(01)_2$	від 192.168.255.65 до 192.168.255.126
підмережа 2: $(10)_2$	від 192.168.255.129 до 192.168.255.190
підмережа 3: $(11)_2$	від 192.168.255.193 до 192.168.255.254

Оскільки у двійковій формі $192_{10}=(1100\ 0000)_2$, то це забезпечує 4 підмережі, кожна з 62 станціями.

Таблиця 2.5 – Використання 3 адресних біт для утворення підмереж

3 біти Мережева маска 255.255.255.224	Адреси станцій
підмережа 0: $(000)_2$	від 192.168.255.1 до 192.168.255.30
підмережа 1: $(001)_2$	від 192.168.255.33 до 192.168.255.62
підмережа 2: $(010)_2$	від 192.168.255.65 до 192.168.255.94
підмережа 3: $(011)_2$	від 192.168.255.97 до 192.168.255.126
підмережа 4: $(100)_2$	від 192.168.255.129 до 192.168.255.158
підмережа 5: $(101)_2$	від 192.168.255.161 до 192.168.255.190
підмережа 6: $(110)_2$	від 192.168.255.193 до 192.168.255.222
підмережа 7: $(111)_2$	від 192.168.255.225 до 192.168.255.254

Оскільки у двійковій формі $224_{10}=(1110\ 0000)_2$, то це забезпечує 8 підмереж, кожна з 30 станціями.

Подібним чином можна отримати діапазони адрес для мережевих масок:

- 4 біти Мережева маска 255.255.255.240 з 16 підмережами по 14 станцій у кожній;

- 5 бітів Мережева маска 255.255.255.248 з 32 підмережами по 6 станцій у кожній;

- 6 бітів Мережева маска 255.255.255.252 з 64 підмережами по 2 станції у кожній.

Перші 24 біти в IP-адресі можна ігнорувати (у нашому прикладі це десяткове 192.168.255), оскільки вони відносяться до базової мережі, в якій організуються підмережі. При використанні двобітової мережевої маски для підмережі 0 бачимо, що останні 8 бітів завжди починаються від 00_8, що означає їх десятковий еквівалент в інтервалі від 0 до 63. Оскільки всі нулі та всі одиниці в адресі станції не можна використовувати, то це виключає із вжитку 0 і 63, так що залишаються числа від 1 до 62. Для підмережі 2 перші два біти останнього октету адреси для всіх станцій підмережі завжди рівні 10_8, так що наявні комбінації решти шести бітів дають десяткові числа від 128 до 191. Виключаючи вживання всіх нулів та всіх одиниць в адресах станцій, отримуємо прийнятні числа від 129 до 190, що знову забезпечує 62 станції. Аналогічно можна пояснити отримані діапазони адрес для інших мережевих масок.

2.3 Матеріали для виконанння роботи

2.3.1 Проведення розрахунків для визначення адрес та маски мережі

Візьмемо базову адресу мережі 10.1.0.0, кількість під мереж: 7, кількість робочих станцій: 55.

Таблиця 2.6 – Розрахунок ip-адрес

Номер підмережі	Адреса мережі	Широковісна адреса	Початкова адреса хостів	Кінцева адреса хостів
0	10.1.0.0	10.1.0.63	10.1.0.2	10.1.0.57
1	10.1.0.64	10.1.0.127	10.1.0.66	10.1.0.121
2	10.1.0.128	10.1.0.191	10.1.0.130	10.1.0.185
3	10.1.0.192	10.1.0.255	10.1.0.194	10.1.0.249
4	10.1.1.0	10.1.1.63	10.1.1.2	10.1.1.57
5	10.1.1.64	10.1.1.127	10.1.1.66	10.1.1.121
6	10.1.1.128	10.1.1.191	10.1.1.130	10.1.1.185
Маска підмережі				
255.255.255.192				

Адреса 10.1.0.1 – адреса gateway в першій підмережі (10.1.0.0 - адреса підмережі). Аналогічно в кожній підмережі значення адреси gateway буде наступним після адресу підмережі. Створена модель мережі зображена на рисунку 2.1.

2.3.2 Модель мережі і налагодження свіча

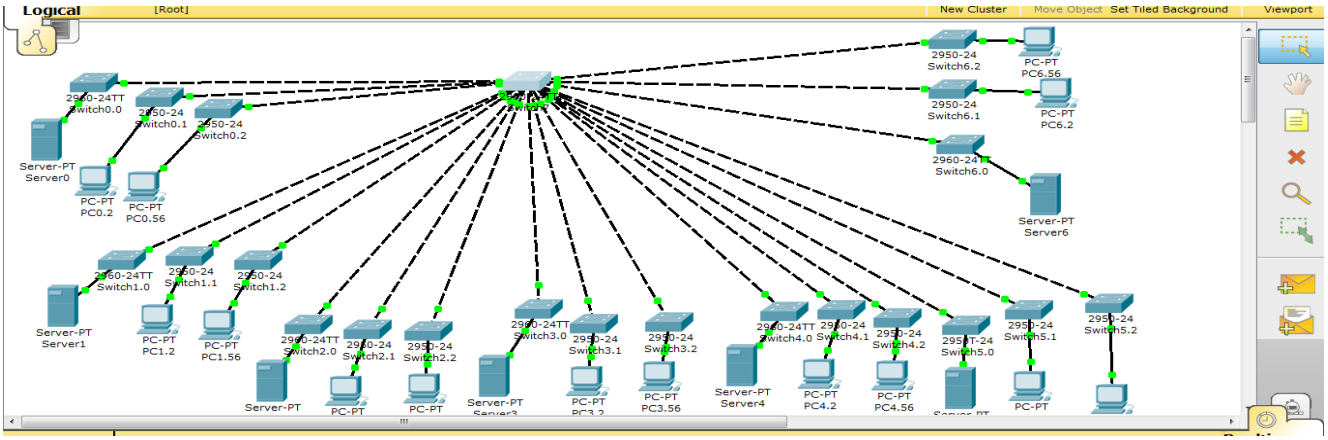


Рисунок 2.1 –Модель мережі

Для створення підмереж треба зайти у закладку «Config» свіча та створити додати необхідну кількість підмереж («VLAN Database»), вказавши в поле VLAN Number номер мережі, починаючи з 2 (1й – зарезервовано), VLAN Name – ім’я мережі (рисунок 2.2).

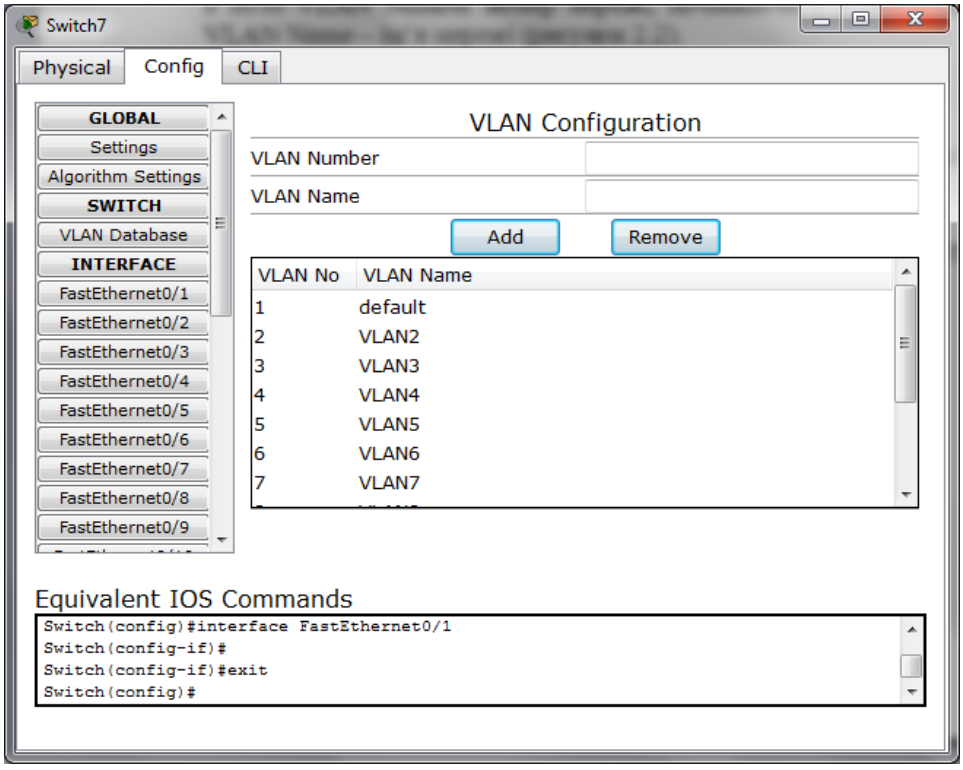


Рисунок 2.2 – Скриншот створення підмереж в базі

На кожний з портів центрального свіча встановити необхідну VLAN (рисунок 2.3).

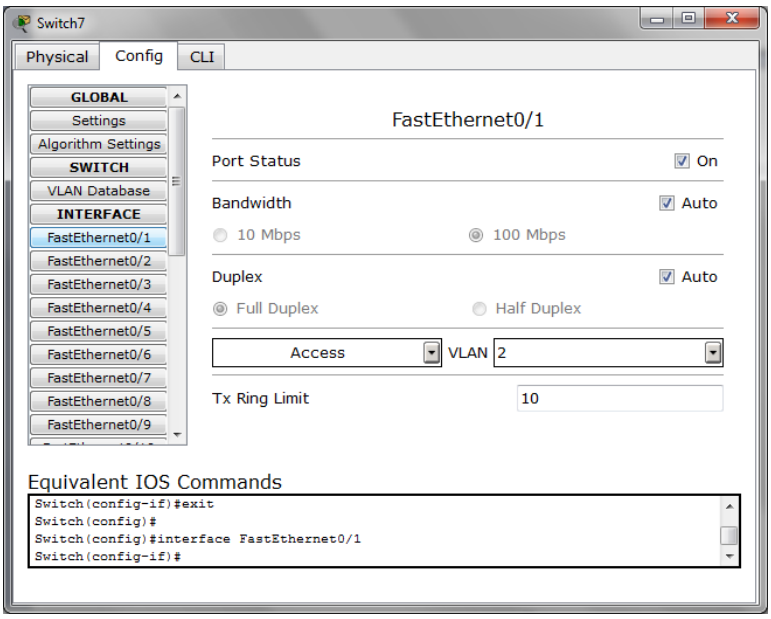


Рисунок 2.3 – Скриншот налаштування портів для підмереж

Свіч складається з 24 портів. Треба під'єднати 55 хостів однієї віртуальної мережі до нього. Отже, використовується 3 додаткових свіча, які під'єднані до центрального свіча, а на портах центрального свіча вказується відповідний однаковий номер підмережі.

Всі підключені порти свіча будуть мати значення «UP» при наведенні на свіч курсором (рисунок 2.4).

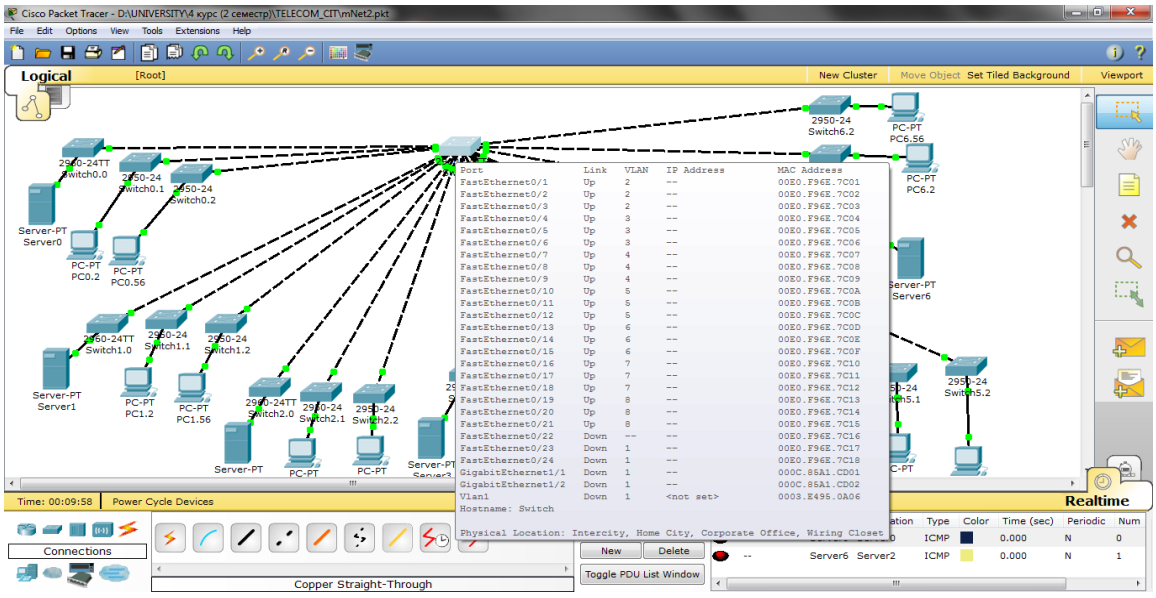
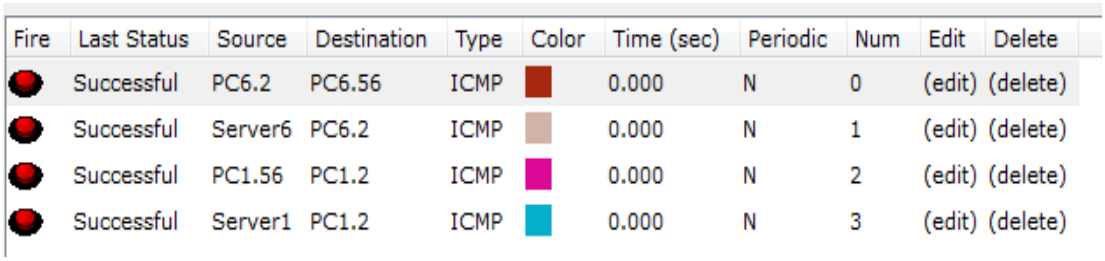


Рисунок 2.4 – Скриншот налагодженого свіча

2.3.3 Аналіз і тестування мережі

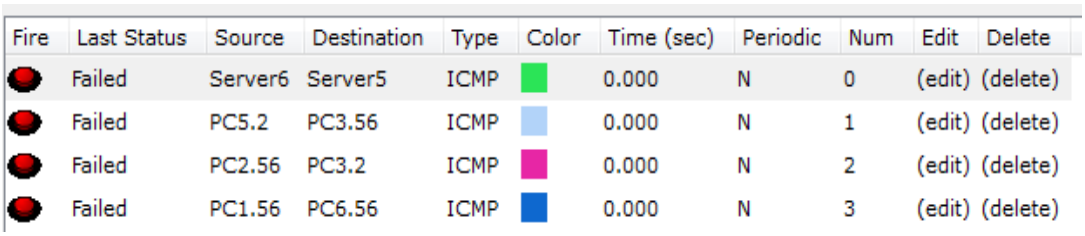
- Відправлення пакетів на робочі станції, які знаходяться в одній віртуальній мережі (рисунок 2.5).



Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num	Edit	Delete
	Successful	PC6.2	PC6.56	ICMP		0.000	N	0	(edit)	(delete)
	Successful	Server6	PC6.2	ICMP		0.000	N	1	(edit)	(delete)
	Successful	PC1.56	PC1.2	ICMP		0.000	N	2	(edit)	(delete)
	Successful	Server1	PC1.2	ICMP		0.000	N	3	(edit)	(delete)

Рисунок 2.5 – Скриншот перевірки передачі пакетів в одній віртуальній мережі

- Відправлення пакетів на робочі станції, які знаходяться в різних віртуальних мережах (рисунок 2.6).



Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num	Edit	Delete
	Failed	Server6	Server5	ICMP		0.000	N	0	(edit)	(delete)
	Failed	PC5.2	PC3.56	ICMP		0.000	N	1	(edit)	(delete)
	Failed	PC2.56	PC3.2	ICMP		0.000	N	2	(edit)	(delete)
	Failed	PC1.56	PC6.56	ICMP		0.000	N	3	(edit)	(delete)

Рисунок 2.6 – Скриншот перевірки передачі пакетів в різні віртуальні мережі

2.4 Хід роботи

1. Створити модель мережі що складається з X підмереж і Y робочих станцій. Використати адреса мережі, X і Y згідно з варіантом завдання (таблиця 2.7).

Таблиця 2.7 –Варіанти завдань

Номер варіанта	Базова адреса мережі	X кількість підмереж	Y кількість робочих станцій
1	192.168.0.0	7	60
2	172.16.0.0	9	40
3	172.17.0.0	8	35
4	172.18.0.0	11	25
5	172.19.0.0	10	30
6	10.1.0.0	7	55
7	10.2.0.0	8	12
8	10.3.0.0	9	20
9	10.4.0.0	10	15
10	10.5.0.0	11	85
11	10.6.0.0	7	67
12	10.7.0.0	8	16

2. Для кожної підмережі створити свій VLAN.

3. Розрахувати маску підмережі, для реалізації потрібної кількості підмреж.

4. Розрахувати адреси мережі, широкомовну адресу і адреси хостів згідно з варіантом завдання. Надати результати розрахунку у вигляді таблиці 2.8.

Таблиця 2.8 — Результати розрахунку IP-адрес.

Номер підмережі	Адрес мережі	Широковісна адреса	Початкова адреса хостів	Кінцева адреса хостів

5. У Packet Tracer 5.0 створити модель LAN, яка складається з потрібної кількості під мереж. Кількість робочих станцій в кожній підмережі моделі визначається початковою і кінцевою адресою. Таким чином в кожній підмережі буде 2 робочі станції, IP-адреси яких будуть початковий і кінцеві адреси хостів для цієї підмережі.

6. З'єднати необхідне число свічів Cisco Catalyst 2960 між собою і підключити до них робочі станції.

7. Об'єднати робочі станції і сервера в різні віртуальні мережі, номер віртуальної мережі встановити по порядку.

8. Скріншот моделі мережі і налаштувань свіча надати в звіті.

9. Аналіз і тестування мережі :

- відправити пакети на робочі станції, що знаходяться в одній віртуальній мережі. Відобразити результати в звіті;

- відправити пакети на робочі станції, що знаходяться в різних віртуальних мережах. Відобразити результати в звіті.

2.5 Контрольні запитання

1. Що таке VLAN? Призначення VLAN?
2. Як реалізований VLAN на рівні протоколу Ethernet?
3. Що таке маска підмережі, на що вона впливає?
4. Що таке адреса мережі? Як виглядає адреса мережі на бітовому рівні?
5. Що таке широкомовна адреса? Як вона виглядає на бітовому рівні? Для чого використовується?
6. Як залежить кількість робочих станцій в мережі від маски?
7. Якою командою призначається VLAN на порт комутатора?
8. Як проглянути список VLAN?
9. Як проглянути налаштування порту?
10. Як поглянути всі налаштування комутатора?

3 Самостійна робота № 3. Моделювання локальних мереж. Робота з VLAN і маршрутизатором.

3.1 Мета роботи.

Вивчити на практиці основи маршрутизації на прикладі маршрутизації віртуальних мереж (VLAN). Використовуючи навички, отримані в попередніх самостійних роботах, створити мережу, до складу якої входитимуть декілька віртуальних мереж і маршрутизатор, що забезпечує зв'язок між віртуальними мережами і, що надають вихід з віртуальних підмереж в зовнішню мережу.

3.2 Теоретичні відомості

Маршрутизатор – мережевий пристрій, що пересилає пакети даних між різними сегментами мережі і приймає рішення на підставі інформації про топологію мережі і певних правил, заданих адміністратором.

Зазвичай маршрутизатор використовує адресу одержувача, вказану в пакетах даних, і визначає по таблиці маршрутизації шлях, по якому слід передати дані. Якщо в таблиці маршрутизації для адреси немає описаного маршруту, пакет відкидається.

Існують і інші способи визначення маршруту пересилки пакетів, коли, наприклад, використовується адреса відправника, використовувані протоколи верхніх рівнів і інша інформація, що міститься в заголовках пакетів мережевого рівня. Часто маршрутизатори можуть здійснювати трансляцію адрес відправника і одержувача, фільтрацію транзитного потоку даних на основі певних правил з метою обмеження доступу, шифрування/дешифровка даних, що передаються.

Таблиця маршрутизації містить інформацію, на основі якої маршрутизатор приймає вирішення про подальшу пересилку пакетів. Таблиця складається з деякого числа записів — маршрутів, в кожній з яких міститься адреса мережі одержувача, адреса наступного вузла, якому слід передавати пакети і деяку вагу запису, — метрика.

Метрики записів в таблиці відіграють роль в обчисленні найкоротших маршрутів до різних одержувачів. Залежно від моделі маршрутизатора і використовуваних протоколів маршрутизації, в таблиці може міститися деяка додаткова службова інформація.

Таблиця маршрутизації може складатися двома способами:

- статична маршрутизація - коли записи в таблиці вводяться і змінюються вручну. Такий спосіб вимагає втручання адміністратора кожного разу, коли відбуваються зміни в топології мережі. З іншого боку, він є найбільш стабільним і вимагає мінімум апаратних ресурсів маршрутизатора для обслуговування таблиці.
- динамічна маршрутизація - коли записи в таблиці оновлюються автоматично за допомогою одного або декількох протоколів маршрутизації - RIP, OSPF, IGRP, EIGRP, IS-IS, BGP, і ін. Крім того, маршрутизатор будує

таблицю оптимальних шляхів до мереж призначення на основі різних критеріїв: кількості проміжних вузлів, пропускної спроможності каналів, затримки передачі даних.

Критерії обчислення оптимальних маршрутів найчастіше залежать від протоколу маршрутизації, а також задаються конфігурацією маршрутизатора. Такий спосіб побудови таблиці дозволяє автоматично тримати таблицю маршрутизації в актуальному стані і обчислювати оптимальні маршрути на основі поточної топології мережі. Проте динамічна маршрутизація надає додаткове навантаження на пристрої, а висока нестабільність мережі може приводити до ситуацій, коли маршрутизатори не встигають синхронізувати свої таблиці, що приводить до суперечливих відомостей про топологію мережі в різних її частинах і втраті даних, що передаються.

Найчастіше для побудови таблиць маршрутизації використовують теорію графів.

3.3 Матеріали для виконання роботи

3.3.1 Розділення мережі на підмережі

Візьмемо базову адресу першої мережі 10.1.0.0, кількість робочих станцій: 55, базову адресу другої мережі: 10.1.0.64 (таблиця 3.1).

Таблиця 3.1 – Розрахунок ip-адрес для мереж

Номер підмережі	Адреса мережі	Широковісна адреса	Початкова адреса хостів	Кінцева адреса хостів
1	10.1.0.0	10.1.0.63	10.1.0.2	10.1.0.57
2	10.1.0.64	10.1.0.127	10.1.0.66	10.1.0.121
Маска підмережі				
255.255.255.192				

3.3.2 Модель мережі і налагодження свіча та маршрутизатора

Створена модель мережі зображена на рисунку 3.1.

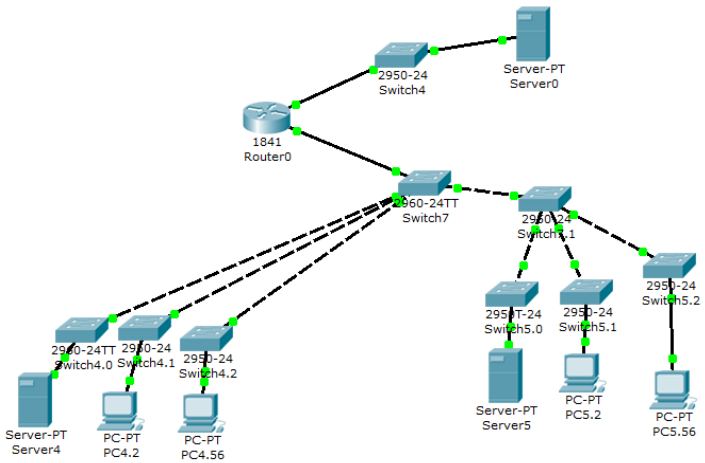


Рисунок 3.1 –Модель мережі

Для створення підмереж треба зайти у закладку «Config» свіча та створити додати необхідну кількість підмереж («VLAN Database»), вказавши в поле VLAN Number номер мережі, починаючи з 2 (1й – зарезервовано), VLAN Name – ім’я мережі (рисунок 3.2).

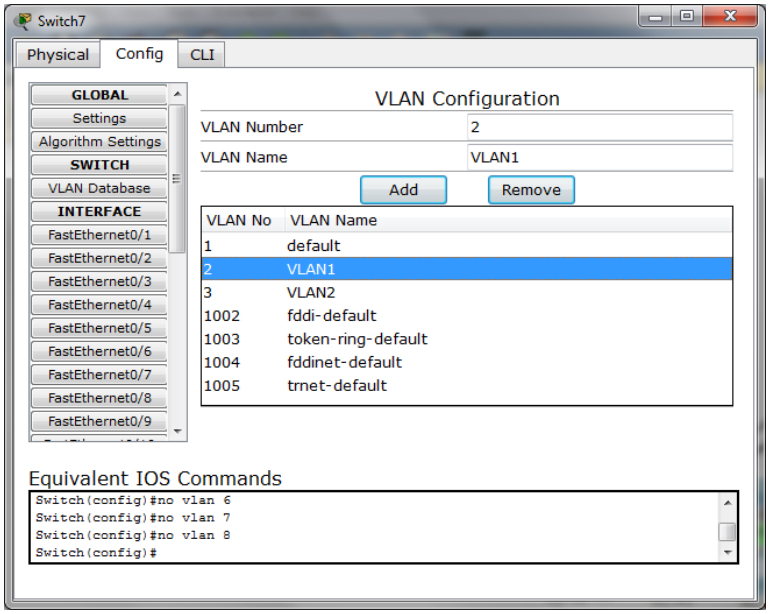


Рисунок 3.2 – Скриншот створення підмереж в базі

На кожний з портів центрального свіча встановити необхідну VLAN (рисунок 3.3).

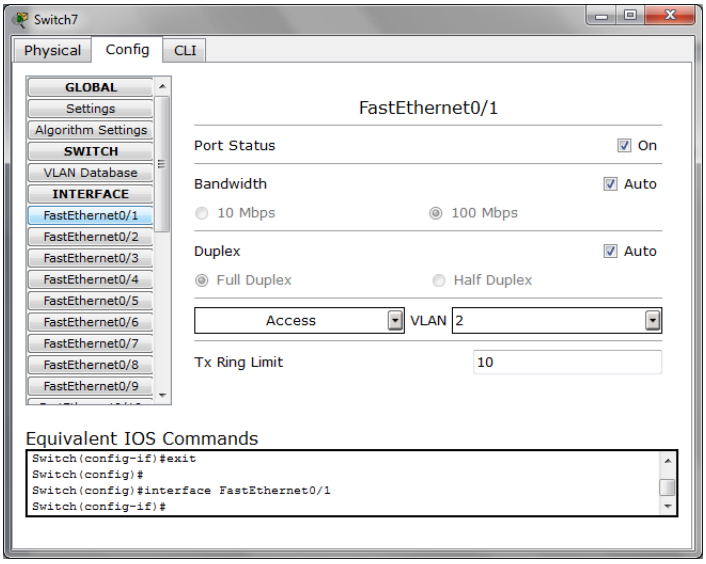


Рисунок 3.3 – Скриншот налаштування портів для підмереж

Підключення VLAN1 було здійснено до портів FastEthernet0/1, FastEthernet0/2, FastEthernet0/3; VLAN2 до порту FastEthernet0/4.

Всі підключені порти свіча будуть мати значення «UP» при наведенні на свіч курсором (рисунок 3.4).

Port	Link	VLAN	IP Address	MAC Address
FastEthernet0/1	Up	2	--	000D.BD2D.8401
FastEthernet0/2	Up	2	--	000D.BD2D.8402
FastEthernet0/3	Up	2	--	000D.BD2D.8403
FastEthernet0/4	Up	3	--	000D.BD2D.8404
FastEthernet0/5	Up	--	--	000D.BD2D.8405
FastEthernet0/6	Down	1	--	000D.BD2D.8406
FastEthernet0/7	Down	1	--	000D.BD2D.8407
FastEthernet0/8	Down	1	--	000D.BD2D.8408
FastEthernet0/9	Down	1	--	000D.BD2D.8409
FastEthernet0/10	Down	1	--	000D.BD2D.840A
FastEthernet0/11	Down	1	--	000D.BD2D.840B
FastEthernet0/12	Down	1	--	000D.BD2D.840C
FastEthernet0/13	Down	1	--	000D.BD2D.840D
FastEthernet0/14	Down	1	--	000D.BD2D.840E
FastEthernet0/15	Down	1	--	000D.BD2D.840F
FastEthernet0/16	Down	1	--	000D.BD2D.8410
FastEthernet0/17	Down	1	--	000D.BD2D.8411
FastEthernet0/18	Down	1	--	000D.BD2D.8412
FastEthernet0/19	Down	1	--	000D.BD2D.8413
FastEthernet0/20	Down	1	--	000D.BD2D.8414
FastEthernet0/21	Down	1	--	000D.BD2D.8415
FastEthernet0/22	Down	1	--	000D.BD2D.8416
FastEthernet0/23	Down	1	--	000D.BD2D.8417
FastEthernet0/24	Down	1	--	000D.BD2D.8418
Vlan1	Down	1	<not set>	00E0.B042.E726

Hostname: Switch

Рисунок 3.4 – Скриншот налагодженого свіча

До порту FastEthernet0/5 свіча необхідно підключити маршрутизатор до порту FastEthernet0/1. Для цього необхідно вибрати тип підключення «Trunk» на свічі (рисунок 3.5).

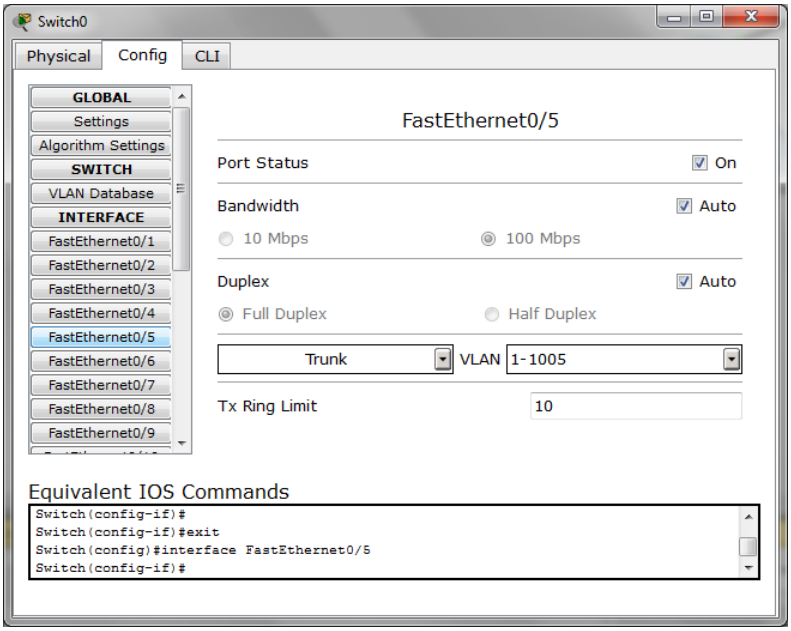


Рисунок 3.5 – Скриншот налаштування порту для маршрутизатора

Після цього слід налаштувати маршрутизатор. Для цього треба зайти у закладку «CLI» маршрутизатора і виконати такі команди :

- Включення роутера:

Router>enable

- Команда для налаштування:

Router#configure terminal

- Вибір порту, який буде налаштовуватись (0/1 – порт маршрутизатора; цифра після крапки – це номер порту свіча):

Router(config-subif)#interface fastEthernet 0/1.1

- Встановлення номеру VLAN (після Dot1Q вказується номер VLAN):

Router(config-subif)#encapsulation Dot1Q 2

- Встановлення ip адреси (ip адреса гейтвею і маска підмережі):

Router(config-subif)#ip address 10.1.0.62 255.255.255.192

Таким чином слід налаштувати маршрутизатор для всіх VLAN, в залежності від того до якого з портів свіча вони під'єднані.

Налаштунки роутера зображено на рисунку 3.6.

```
Port               Link   VLAN   IP Address         IPv6 Address                            MAC Address
FastEthernet0/0    Down   --     <not set>          <not set>                               0000.0CB1.D501
FastEthernet0/1    Up     --     <not set>          <not set>                               0000.0CB1.D502
FastEthernet0/1.1  Up     --     10.1.0.62/26       <not set>                               0000.0CB1.D502
FastEthernet0/1.2  Up     --     <not set>          <not set>                               0000.0CB1.D502
FastEthernet0/1.3  Up     --     <not set>          <not set>                               0000.0CB1.D502
FastEthernet0/1.4  Up     --     10.1.0.126/26      <not set>                               0000.0CB1.D502
Vlan1              Down   1      <not set>          <not set>                               0040.0B12.3CDE
Hostname: Router

Physical Location: Intercity, Home City, Corporate Office, Wiring Closet
```

Рисунок 3.6 – Скриншот налагодженого роутера

Для створення зовнішньої мережі необхідно до іншого порту маршрутизатора (FastEthernet0/0) підключити інший комутатор до порту FastEthernet0/1. Також до цього свіча під'єднати сервер з ip адресою гейтвею і маскою (наприклад, 192.168.10.1 255.255.255.0).

Налаштовувати роутер для зовнішньої мережі слід такими командами:

Router>enable

Router#configure terminal

Router(config-subif)#interface fastEthernet 0/0

Router(config-subif)#ip address 192.168.10.1 255.255.255.0

Налаштунки роутера з двома VlAN і зовнішньою мережею зображено на рисунку 3.6.

```
Port               Link   VLAN   IP Address         IPv6 Address                            MAC Address
FastEthernet0/0    Up     --     192.168.10.1/24    <not set>                               0000.0CB1.D501
FastEthernet0/1    Up     --     <not set>          <not set>                               0000.0CB1.D502
FastEthernet0/1.1  Up     --     10.1.0.62/26       <not set>                               0000.0CB1.D502
FastEthernet0/1.2  Up     --     <not set>          <not set>                               0000.0CB1.D502
FastEthernet0/1.3  Up     --     <not set>          <not set>                               0000.0CB1.D502
FastEthernet0/1.4  Up     --     10.1.0.126/26      <not set>                               0000.0CB1.D502
Vlan1              Down   1      <not set>          <not set>                               0040.0B12.3CDE
Hostname: Router

Physical Location: Intercity, Home City, Corporate Office, Wiring Closet
```

Рисунок 3.7 – Скриншот налагодженого роутера з двома VlAN і зовнішньою мережею

3.3.3 Аналіз і тестування мережі

Перевірка мережі до підключення маршрутизатора:

Відправлення пакетів на робочі станції, які знаходяться в одній віртуальній мережі (рисунок 3.8).

Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num	Edit	Delete
	Successful	Server5	PC5.2	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC5.56	Server5	ICMP		0.000	N	1	(edit)	(delete)
	Successful	PC4.56	PC4.2	ICMP		0.000	N	2	(edit)	(delete)
	Successful	Server4	PC4.56	ICMP		0.000	N	3	(edit)	(delete)

Рисунок 3.8– Скриншот перевірки передачі пакетів в одній віртуальній мережі

Відправлення пакетів на робочі станції, які знаходяться в різних віртуальних мережах (рисунок 3.9).

Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num	Edit	Delete
	Failed	Server5	Server4	ICMP		0.000	N	0	(edit)	(delete)
	Failed	PC5.56	PC4.2	ICMP		0.000	N	1	(edit)	(delete)
	Failed	PC4.56	Server5	ICMP		0.000	N	2	(edit)	(delete)
	Failed	PC5.2	Server4	ICMP		0.000	N	3	(edit)	(delete)

Рисунок 3.9 – Скриншот перевірки передачі пакетів в різні віртуальні мережі

Перевірка мережі після підключення маршрутизатора:

- Відправлення пакетів на робочі станції, які знаходяться в одній віртуальній мережі (рисунок 3.10).

Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num	Edit	Delete
	Successful	Server5	PC5.2	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC5.56	Server5	ICMP		0.000	N	1	(edit)	(delete)
	Successful	PC4.56	PC4.2	ICMP		0.000	N	2	(edit)	(delete)
	Successful	Server4	PC4.56	ICMP		0.000	N	3	(edit)	(delete)

Рисунок 3.10– Скриншот перевірки передачі пакетів в одній віртуальній мережі

- Відправлення пакетів на робочі станції, які знаходяться в різних віртуальних мережах (рисунок 3.11).

Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num	Edit	Delete
	Successful	Server5	Server4	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC5.56	PC4.2	ICMP		0.000	N	1	(edit)	(delete)
	Successful	PC4.56	Server5	ICMP		0.000	N	2	(edit)	(delete)
	Successful	PC5.2	Server4	ICMP		0.000	N	3	(edit)	(delete)

Рисунок 3.11 – Скриншот перевірки передачі пакетів в різні віртуальні мережі

- Виконання трасування пакетів до робочих станцій, які знаходяться в різних віртуальних мережах.

```
SERVER>tracert 10.1.0.121
Tracing route to 10.1.0.121 over a maximum of 30 hops:
  1    15 ms     9 ms      14 ms     10.1.0.62
  2    30 ms     30 ms     30 ms     10.1.0.121

Trace complete.

SERVER>tracert 10.1.0.66
Tracing route to 10.1.0.66 over a maximum of 30 hops:
  1    12 ms     11 ms     14 ms     10.1.0.62
  2    27 ms     13 ms     28 ms     10.1.0.66
Trace complete.
```

- Виконання трасування пакетів до зовнішньої мережі, використовувати IP адресу сервера.

```
SERVER>tracert 192.168.10.2
Tracing route to 192.168.10.2 over a maximum of 30 hops:
  1    11 ms     12 ms     13 ms     10.1.0.62
  2    20 ms     22 ms     22 ms     192.168.10.2
Trace complete.
```

3.4 Хід роботи

1 Створити модель мережі, що складається з 2 підмереж, що входять до складу двох VLAN з номерами Номер Варіанта+1 і X. Розбити базову підмережу по масці Y. Використовувати адресу мережі, X і Y згідно з варіантом завдання.

2 З'єднати мережі 2-ма комутаторами, налаштувати порти комутатора для роботи з відповідними VLAN і з'єднати комутатори між собою.

3 До одного з комутаторів підключити маршрутизатор з 2 мережевими інтерфейсами.

4 До одного з комутаторів підключити маршрутизатор з 2 мережевими інтерфейсами. Зовнішня мережа емулюється комутатором і сервером.

Для виконання робити необхідно зробити наступне:

1. Визначити адресу мережі, X і Y для свого варіанту (таблиця 3.2), варіант згідно з номером за списком.

Таблиця 3.2 – Варіанти завдань

Номер варіанту	Базова адреса мережі	X номер 2-го VLAN	Y маска підмережі
1	192.168.0.0	7	/ 24
2	172.16.0.0	9	/ 23
3	172.17.0.0	8	/ 25
4	172.18.0.0	11	/ 28
5	172.19.0.0	10	/ 27
6	10.1.0.0	6	/ 26
7	10.2.0.0	7	/ 23
8	10.3.0.0	8	/ 28
9	10.4.0.0	6	/ 24
10	10.5.0.0	10	/ 25
11	10.6.0.0	7	/ 26
12	10.7.0.0	8	/ 27

2. Розрахувати які IP адреси потрапляють в задану маску підмережі, для реалізації потрібної кількості підмереж.

3. У Packet Tracer 5.0 створити модель LAN такою, що складається з двох підмереж, кількість робочих станцій в кожній підмережі моделі визначається початковою і кінцевою адресою. Таким чином в кожній підмережі буде 2 робочі станції, IP адреси яких будуть початковими і кінцевми адресами хостів для цієї підмережі.

4. З'єднати комутатори Cisco Catalyst 2960 між собою і підключити до них робочі станції.

5. Об'єднати робочі станції в різні віртуальні мережі, згідно з варіантом завдання.

6. Провести аналіз і тестування мережі, згідно з алгоритмом наведеному нижче:

- до одного з комутаторів підключити один з інтерфейсів маршрутизатора;
- налаштувати порт комутатора для пропускання всіх Vlan-ів на маршрутизатор;
- на маршрутизаторі створити 2 підінтерфейси, з IP адресою, маскою і номером відповідного Vlan-а;
- на другому інтерфейсі маршрутизатора налаштувати IP адреси з зовнішньої мережі, для зовнішньої підмережі використовувати довільні IP адреси, що не потрапляють в діапазон локальних адрес;

- з'єднати другий інтерфейс маршрутизатора з комутатором зовнішньої мережі, до якого підключений сервер;
- на робочих станціях прописати основний шлюз, відповідний IP адресі підінтерфейсу маршрутизатора для даного VLAN.

7. Знімок екрану моделі мережі, налаштування комутатора і маршрутизатора надати в звіті.

8. Аналіз і тестування мережі:

1) До підключення маршрутизатора:

- відправити пакети на робочі станції, що знаходяться в одній віртуальній мережі. Відобразити результати в звіті;
- відправити пакети на робочі станції, що знаходяться в різних віртуальних мережах. Відобразити результати в звіті.

2) Після підключення маршрутизатора:

- відправити пакети на робочі станції, що знаходяться в одній віртуальній мережі. Відобразити результати в звіті;
- відправити пакети на робочі станції, що знаходяться в різних віртуальних мережах. Відобразити результати в звіті;
- виконати трасування пакетів до робочих станцій, що знаходяться в різних віртуальних мережах. Відобразити результати в звіті;
- виконати трасування пакетів до зовнішньої мережі, використовувати IP адреса сервера. Відобразити результати в звіті.

3.5 Контрольні запитання

1. Якою командою призначається VLAN на порт комутатора?
2. Як проглянути список VLAN?
3. Як проглянути налаштування порту?
4. Як поглянути всі налаштування комутатора?
5. Як створити підінтерфейс на маршрутизаторі?
6. Як додати підінтерфейс маршрутизатора в потрібний VLAN?
7. Як перевірити трасу по якій йдуть пакети до кінцевої крапки?
8. Як проглянути всі налаштування маршрутизатора?
9. Як включити маршрутизацію?
10. Якими властивостями повинен володіти порт комутатора, до якого підключений маршрутизатор?

4 Самостійна робота №4. Вивчення мережевих аналізаторів tcpdump і Wireshark

4.1 Мета роботи

Вивчити мережеві аналізатори tcpdump і Wireshark. Провести дослідження мережевого трафіку і виявити його особливості.

4.2 Короткі теоретичні відомості

Аналізатор трафіку, або сніфер (від англ. to sniff – нюхати) – мережевий аналізатор трафіку, програма або програмно-апаратний пристрій, призначений для перехоплення і подальшого аналізу, або тільки аналізу мережевого трафіку, призначеного для інших вузлів.

Під час роботи сніфера мережевий інтерфейс перемикається в т.з. «режим прослуховування» (Promiscuous mode), що і дозволяє йому одержувати пакети, адресовані іншим інтерфейсам в мережі.

Утиліта tcpdump призначена для аналізу мережевого трафіку і входить до складу всіх POSIX систем. Ця утиліта виводить заголовки пакетів, які відповідають заданим критеріям, на мережевому інтерфейсі, перекладеному попередньо в режим прийому всіх пакетів (promiscous mode). Критерії задаються в формі логічного виразу.

Логічні вирази для критеріїв необхідні для того, що б із загального мережевого трафіку виділити тільки ті, котрі цікавлять нас пакети. Синтаксис логічних виразів включає наступні ключові слова:

host - IP адреса або DNS ім’я хоста

net - адреса мережі, наприклад

net 192.168.7, net 192.168.7.0 mask 255.255.255.224

port – номер порту (має сенс для протоколів TCP та UDP)

proto – тип протокола. Можливі типи: ether, fddi, tr, ip, ip6, arp, rarp, decnet, lat, sca, moprc, mopdl, iso, esis, isis, icmp, icmp6, tcp and udp. Наприклад, tcpdump tcp port 80

dir – напрямок, можливі значення – src або dst . Наприклад,

```
tcpdump src host cs.stu.cn.ua.
```

Крім того, можна використовувати адреси несучої мережі Ethernet:

```
tcpdump ether dst 00:02:44:5b:ee:9b
```

або IP мережа:

```
tcpdump net src 192.168.7.0/27
```

Більш повну інформацію про можливості утиліти tcpdunp можна отримати зі стандартної сторінки допомоги, задавши команду man tcpdump.

Мережевий аналізатор Wireshark побудований на тій же бібліотеці (libpacap), що і утиліта tcpdump, але має зручний графічний користувальницький інтерфейс.

Головне вікно програми наведено на рис. 2.1

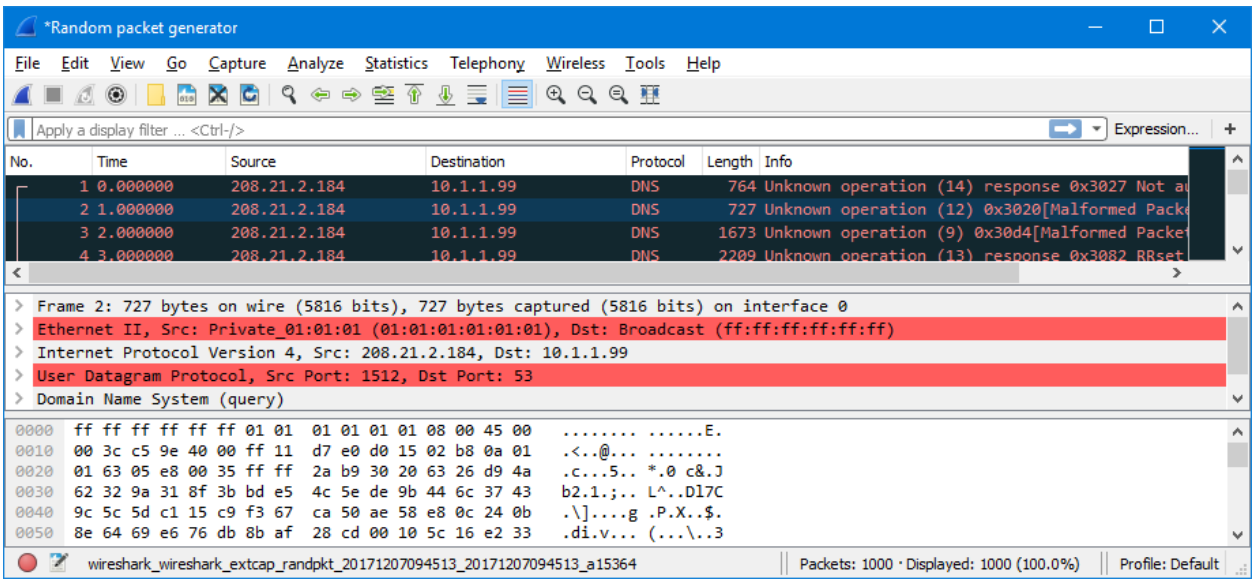


Рисунок 2.1. Головне вікно Wireshark підчас захоплення мережевих пакетів

У верхній частині вікна міститься перелік пакетів, захоплених з мережі. При натисканні правої кнопки миші на тому чи іншому пакеті з'являється контекстне меню. Список можна відсортувати за будь-яким полем.

Середнє вікно містить інформацію щодо протоколів, що інкапсулюються відповідно загальним принципам моделі OSI для обраного у верхньому вікні пакета.

Нижнє вікно містить шістнадцяткове подання обраного пакета. При обиранні того або іншого поля в середньому вікні автоматично буде підсвічуватися відповідна ділянка шістнадцяткового подання.

Для захоплення або відображення тільки тих пакетів, які цікавлять, використовуються два види фільтрації: під час захоплення і під час відображення.

4.3 Хід роботи

1. Проведіть захоплення пакетів утилітою Wireshark без фільтра протягом декількох хвилин і уважно перегляньте результат. Відмітьте, які протоколи використовуються в мережі. Визначте призначення знайдених протоколів.
2. Налаштуйте фільтр на захоплення тільки широкомовних пакетів. Проведіть захоплення протягом декількох хвилин, розгляньте результат.
3. Налаштуйте фільтр на захоплення пактів ICMP. Перевірте результат, включивши захоплення пакетів і запустивши зі свого хоста команду ping на сусідній хост.
4. Не змінюючи налаштувань фільтра, запустіть захоплення пакетів і запустіть ping на іншому хості, направлений на третій хост в вашому сегменті. Чи вдалося Вам захопити ці пакети? Поясніть результат.
5. Запустіть утиліту tcpdump з порожнім фільтром з перенаправленням пакетів в файл на кілька хвилин з ключами, що забезпечують розшифрування пакетів. Поясніть результат захоплення пакетів.

4.4 Зміст звіту

Звіт повинен містити послідовність скріншотів по ходу виконання роботи і відповідні коментарі.

4.5 Контрольні запитання

1. Чому Ви не бачите все пакети, що проходять в даному сегменті Ethernet?
2. У якому випадку хост може бачити усі пакети в даному сегменті Ethernet?
3. Які пакети буде видно спостерігачу хосту в комутованому сегменті Ethernet?
4. Як забезпечити захоплення всіх пакетів, що приходять в даний сегмент мережі і йдуть з нього?
5. Що таке "дзеркальний" порт комутатора, і яке його призначення?
6. Який параметр комутатора відповідає за загальну пропускну здатність?

5 Самостійна робота №5. Вивчення мережевого протоколу TCP і протоколу рівня додатків telnet

5.1 Мета роботи

Вивчити мережевий протокол TCP і протокол додатків telnet. Визначити основні етапи і особливості використання протоколів.

5.2 Короткі теоретичні відомості

Протокол TCP є транспортним протоколом з гарантованою доставкою даних, з встановленням з'єднання і повторною втрачених сегментів.

Встановлення з'єднання відбувається за допомогою механізму т.зв. триразового рукостискання (three way handshake).

Хост А, ініціатор з'єднання, посилає сегмент без даних з встановленим прапором SYN. Ініціює з'єднання програма - клієнт. Порт призначення визначається жорстко або як добре відомий сервіс (Well Known Service), наприклад web-сервіс зазвичай має порт 80, або задається користувачем. Порт джерела зазвичай виділяється системою з пулу вільних НЕ привілейованих портів (> 1023).

У відповідь хост В посилає сегмент без даних з встановленими прапорами SYN, ACK. При цьому, в залежності від режиму роботи серверного сокета, можлива заміна фіксованого порту на динамічний в якості вихідного.

Хост А, прийнявши описаний вище пакет, сигналізує про готовність до обміну даними, посилаючи сегмент без даних з встановленим прапором ACK. Після цього сокети готові до двонаправленого обміну даними.

Розрив з'єднання відбувається аналогічно з використанням прапора FIN.

Протокол TELNET є протоколом рівня додатків, призначеним для віддаленого доступу по мережі до текстового терміналу. При ініціалізації з'єднання надсилаються команди і клієнтом, і сервером, що забезпечують узгодження можливостей для користувача терміналу і установку необхідних змінних оточення на сервері.

5.3 Хід роботи

1. Встановіть сервіс telnet на сусідній робочої станції (назвемо її host2). Для цього в директорії /etc/xinetd.d в файлі telnet рядок "disabled = yes" замініть рядком "disabled = no" і перезапустіть сервіс xinetd командою **/etc/rc.d/init.d/xinetd restart.**
2. Перевірте працездатність сервісу: telnet host2. Ви повинні отримати термінальний доступ до машини host2.
3. Запустіть на своїй робочій станції (далі -host1) програму захоплення пакетів ethereal з фільтром, що обмежує захоплення трафіку між host 1 і host2 по протоколу tcp, що б в захоплені пакети не потрапляло "сміття". Запустіть захоплення пакетів.

4. В термінальному віконці запустіть сесію telnet на host2. Увійдіть в систему, ввівши логін і пароль. Вийдіть з системи командою logout. Помістіть протокол сесії в файл для звіту.
5. Зупиніть захоплення пактів і збережіть результат в форматі tcpdump. Збережений результат захоплення подайте на вхід утиліти tcpdump і результат розбору перенаправьте файл для звіту.
6. Вивчаючи паралельно протокол сесії telnet, результати захоплення пакетів у вікні ethereal і в файлі з розібраними пакетами, знайдіть і прокоментуйте: встановлення і розрив до втрати з'єднання по протоколу TCP з урахуванням прапорів; пакети, що містять логін і пароль користувача.
7. Покажіть механізм інкапсуляції даних на прикладі пакета з даними від host2.

5.4 Зміст звіту

Звіт повинен містити протокол telnet сесії і роздруківку дампа tcpdump. Пакети, що містять істотну інформацію для даної самостійної роботи, повинні бути ретельно прокоментовані. Крім того, звіт повинен містити висновки, що описують процес вивчення протоколів TCP і telnet.

5.5 Контрольні запитання

1. З яких рівнів складається стек TCP / IP і для чого вони призначені?
2. Формат заголовка протоколу TCP / IP?
3. Встановлення з'єднання, передачі даних і завершення з'єднання між клієнтом і сервером за допомогою протоколу TCP / IP?
4. У чому полягають особливості протоколу telnet?

6 Самостійна робота №6.Вивчення мережевого протоколу UDP і протоколу рівня додатків DNS

6.1 Мета роботи

Вивчити мережевий протокол UDP і протокол рівня додатків DNS. Визначити основні етапи і особливості використання протоколів.

6.2 Короткі теоретичні відомості

Протокол UDP є найпростішим транспортним протоколом без гарантії доставки даних і без встановлення з'єднання. Даний протокол забезпечує мультиплексування даних між додатками за допомогою поля port а так само контроль правильності даних за допомогою поля checksum. Даний протокол використовується для обміну короткими структурованими даними в режимі "запит-відповідь" а так само для посилки широкомовних повідомлень. У порівнянні з протоколом TCP цей протокол забезпечує більшу швидкодію, оскільки не має витрат на установку і призвести до втрати з'єднання. Зручно так само застосування протоколу UDP для випадків спеціального транспорту, коли транспорт TCP з яких-небудь міркувань не влаштовує розробника. Однак слід зазначити, що механізми підтвердження і збірки потоку в цьому випадку повинні забезпечуватися додатком.

Служба доменних імен DNS є основною системною службою в мережах TCP / IP, оскільки ця служба забезпечує дозвіл символьних імен в IP адреси і навпаки. Кожна програма, що використовує мережеві функції, звертається до базової системної бібліотеці libc, частиною якої є так званий резольвер (resolver). Резольвер має свій файл конфігурації /etc/resolv.conf, в якому описані найближчі сервера імен та порядок підстановки суфіксів.

```
order bind,hosts
search stu stu.cn.ua
nameserver 192.168.0.10
nameserver 192.168.0.14
```

Основою DNS є уявлення про ієрархічну структуру доменного ім’я і зонах. Кожен сервер, що відповідає за ім'я, може делегувати відповідальність за подальшу частину домена іншому серверу (з адміністративної точки зору — іншій організації або людині), що дозволяє покласти відповідальність за актуальність інформації на сервери різних організацій (людей), що відповідають лише за «свою» частину доменного імені.

У мережі повинно бути як мінімум два сервера імен для забезпечення безперебійного дозволу імен. Звернення резольвера відбувається спочатку до першого сервера, і якщо відповідь не отримана протягом короткого часу, до другого. Якщо сервер не може самостійно відпрацювати запит, він звертається до серверів домену кореневого домена "." і здійснює пошук сервера, здатного обробити запит. Отримана відповідь перенаправляється клієнту і кешується на сервері для прискорення наступних відповідей.

6.3 Хід роботи

1. Запустіть аналізатор ethereal с фільтром, налаштованим на відстеження трафіку від вашого хоста до сервера DNS (далі -dns_host) і до сервера www (далі - www_host).
2. Перевірте працездатність фільтру командами:

ping www_host
ping dns_host

Аналізатор повинен показати захоплені пакети.

3. Перезапустить захоплення пакетів.
4. Запустіть програму перегляду web на хост www_host.
5. Збережіть захоплені пакети в форматі libpcap.
6. Розпочніть аналізатор tcpdump для розбору збереженого файлу з наступними ключами:

tcpdump -vvv -X -r файл_пакетів >файл_розбору

Даний файл буде містити розібрані пакети звернення браузера до веб-сторінки і до сервера DNS.

7. Використовуючи обидва аналізатора, відстежити, як відбувалися звернення до серверів і прокоментуйте в файлі всі пакети.
8. Запустіть нову сесію захоплення пакетів.
9. Виконати команди **host www.yahoo.com** та **host 193.193.193.100** та повторіть описану вище процедуру для захоплених пакетів.
10. Детально розберіть всі запити до сервера DNS.

6.4 Зміст звіту

Звіт повинен містити прокоментовані файли з розібраними результатами захоплення пакетів. Пакети, які не становлять особливого інтересу в ключі даної роботи можна видалити з файлу дампа.

6.5 Контрольні запитання

1. Чому для служби DNS використовується протокол UDP?
2. Яке поле заголовка UDP забезпечує мультиплексування пакетів між додатками?
3. Як програми визначають, де знаходиться найближчий сервер імен?
4. Які додаткові параметри передаються у відповіді сервера імен?
5. Якого типу запити відпрацьовувалися під час перегляду веб-сторінки?
6. Який запит використовувався для визначення імені хоста на його адресу?

7 Самостійна робота №7. Конфігурація служби імен DNS в корпоративній мережі

7.1 Мета роботи

Навчитися конфігурувати і тестувати службу імен для корпоративної мережі на основі сервера DNS bind.

7.2 Короткі теоретичні відомості

DNS - розшифровується як Domain Name System, тобто є розподіленою системою для отримання інформації про доменні зонах. Найбільш часто використовується для отримання IP-адреси по імені хоста, адже зак кожним існуючим в Мережі сайтом закріплений певний IP-адреса.

DNS-сервера - це безліч серверів розташованих в Інтернет, які передають дані між собою в реальному часі про доменах. По суті, це система управління доменними іменами усього світу. Відрізняють кілька видів DNS-серверів, в залежності від функцій, які вони виконують: авторитативні, кешуючий, локальний, що перенаправляє, кореневої, реєструючий і DNSBL-сервер .

Наприклад, авторитативні відповідає за певну зону; кешуючий обслуговує запити клієнтів; локальний використовується для обслуговування DNS-клієнтів, які виконуються на локальній машині; в більшості випадків перенаправляє сервер використовується для зниження навантаження на кешуючий DNS-сервер; кореневої - авторитативні за кореневу зону, в світі всього 13 кореневих серверів; реєструючий приймає оновлення від користувачів; DNSBL використовує механізм і протокол роботи, що і DNS-сервери.

В мережі Інтернет коренем дерева є домен “.”. Повне - *абсолютне* або абсолютно визначене, *fully qualified domain name* – домене імя, що закінчується крапкою, що визначає корень доменого дерева, однак ця крапка майже завжди не додається.

На рис.7.1 наведено ієрархію імен в мережі Інтернет.

Основні параметри NS-записи домену:

1. Запис типу A - перекладає з мовного формату в Цифровий. У тому числі необхідний для зв'язку домену та хостингу, де розташовується сайт;

2. Запис AAAA (address record для IPv6) - пов'язує хост з адресою IPv6. Запис AAAA є повним еквівалентом записи типу A, яка описувалася вище, але має інший вигляд.

3. Запис CNAME (canonical name record) або канонічний запис імені перенаправляє на інше ім'я. CNAME це канонічне ім'я або синонім існуючого імені хоста, який повинен мати запис A.

4. TXT запис - додаткові записи, які використовують в основному для настройки пошти;

5. MX - це запис, яка ідентифікує поштовий сервер для домена, допомагає налаштувати обмін поштовими повідомленнями від домену;

6. IP - показує унікальну адресу сервера, де розташований Ваш домен або хостинг;

7. NS записи - призначені для зв'язку вашого домену і хостингу, а також для делегування домену.

За кожно зону відповіє мінімум два сервери: первинний (*primary*) та вторинний (*secondary*). Зміни в базу даних DNS додаються тільки на первинному сервері. Вторинний сервер в свою чергу використовує спеціальний запит ("zone transfer") для оновлення своїх даних.

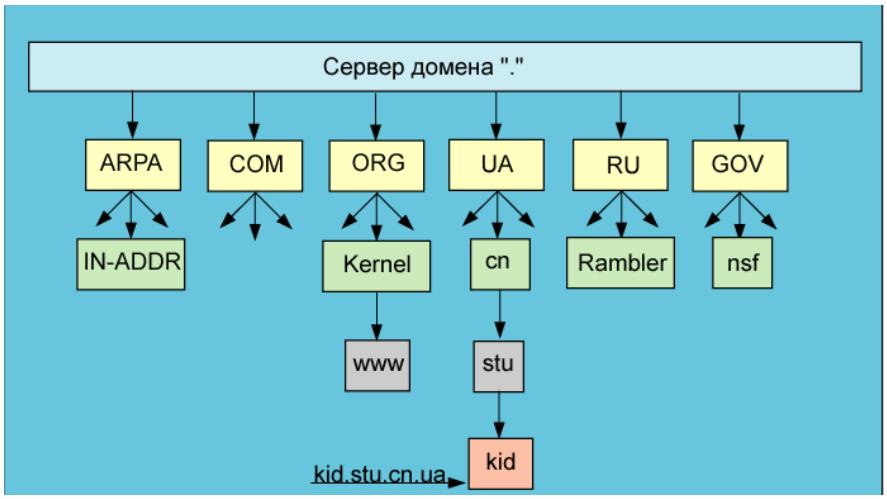


Рисунок 7.1 – Иерархия имен в Интернет-DNS

За каждую зону DNS отвечает не менее двух серверов. Один из них

До основної класифікації роботи DNS серверів відносять 3 режима роботи:

1 **master (primary)**. Даний режим використовуэться адмыныстратором зони, файли баз даних змінюються адміністратором власноруч. Даний режим є найавторітетнішим;
2 **slave (secondary)**. Даний режим використовується за вимогою адміністратора зони, яка автоматично копіюється з master сервера. Цей сервер також є авторитетним джерелом для даної зони;
3 **hint (caching)**. Режим кешування всіх запитів.

7.3 Хід роботи

Виконання даної самостійної роботи складається з наступних кроків:

1. Створіть файли зон для прямої і зворотної приватної зони. Візьміть блок адрес 172.16.X.0, де X - номер машини в класі. Налаштуйте первинний сервер DNS для цих зон на локальному комп'ютері і запустіть його. Не забудьте, що всі повідомлення виводяться не на консоль, а в системний журнал. Повідомлення найзручніше переглядати в окремому терміналі командою:

tail -f /var/log/messages

2. Проведіть тестування створених прямої і зворотної зони за допомогою команди ***dig***.
3. Проведіть тестування дозволу зовнішніх імен вашим сервером. Запитайте, наприклад інформацію про зону slashdot.org.
4. Налаштуйте ваш сервер як вторинний для зон, які створив ваш сусід по лабораторії. Проведіть коректування та перевірку записів зон на предмет записів типу NS. Проведіть перевірку командою dig, звертаючи особливу увагу на секцію "AUTHORITY SECTION".

7.4 Зміст звіту

Звіт повинен містити файли зон, файл конфігурації сервера і результати перевірок. Наявність коментарів і висновків необхідно.

7.5 Контрольні запитання

1. Чому для корпоративних зон зручніше використовувати 3-х буквені імена?
2. У якому випадку сервер імен вважається авторитетним?
3. Які параметри відповідають за час відновлення зони вторинним сервером?
4. Як забезпечити захист зони від скачування і від перегляду?
5. Який запис RR застосовується при створенні віртуальних серверів?
6. Який запис RR задає поштовий обмінник для всієї зони?
7. Який файл містить адреси кореневих серверів імен, необхідних для ініціалізації кеша і рекурсивних запитів?
8. Що таке рекурсивний запит?
9. Як проводиться установка додаткового сервера для конкретної зони?

8 Самостійна робота №8. Конфігурація служби DHCP в корпоративній мережі

8.1 Мета роботи

Ознайомитися з сервісом DHCP, а також отримати навички в його налаштуванні.

8.2 Короткі теоретичні відомості

DHCP (Dynamic Host Configuration Protocol - протокол динамічної конфігурації вузла) - це мережевий протокол, що дозволяє комп'ютерам автоматично отримувати IP-адресу та інші параметри, необхідні для роботи в мережі TCP / IP. Даний протокол працює по моделі «клієнт-сервер». Для автоматичної конфігурації комп'ютер-клієнт на етапі конфігурації мережевого пристрою звертається до так званого сервера DHCP, і отримує від нього потрібні параметри. Адміністратор може задати діапазон адрес, що розподіляються сервером серед комп'ютерів. Це дозволяє уникнути ручного налаштування комп'ютерів мережі й зменшує кількість помилок. Протокол DHCP використовується в більшості мереж TCP / IP.

DHCP є розширенням протоколу BOOTP, що використовувався раніше для забезпечення бездискових робочих станцій IP-адресами при їх завантаженні. DHCP зберігає зворотну сумісність з BOOTP.

8.3 Хід роботи

Виконання самостійної роботи складається з наступних кроків:

1. Перевірка наявності встановлених пакетів dhcp (команда rpm -qa | grep dhcp). Якщо пакети встановлені - приступаємо до роботи (крок два), якщо немає - встановлюємо необхідні пакети (dhcpd-XXX.rmp).
2. Створення конфігураційного файлу сервера /etc/dhcpd.conf за наведеним вище прикладом (подробиці man dhcpd.conf).
3. Налаштування однієї з машин в аудиторії на отримання IP адреси автоматично. Зафіксуємо результати в звіті (ifconfig / all на клієнті і /var/lib/dhcp/dhcpd.leases на сервері).
4. Створення статичного записи для кожного клієнта мережі на основі файлу /var/lib/dhcp/dhcpd.leases. Зафіксуємо результат в звіт.

8.4 Зміст звіту

Звіт повинен містити конфігураційний файл сервера /etc/dhcpd.conf, а також відображати результати всіх вище зазначених дій. Наявність відповідних коментарів і висновків необхідно.

8.5 Контрольні запитання

1. Що являє собою DHCP?
2. Як настроїти DHCP-сервер під Unix?

3. Як налаштувати DHCP-клієнт під Unix?
4. Які основні параметри вказуються у файлі конфігурації dhcpd.conf?
5. Опишіть механізм виділення IP-адрес за допомогою мережевого сервісу DHCP.
6. У якому випадку рекомендується виділяти фіксовані адреси хостів?
7. Які параметри отримує робоча станція від сервера DHCP?

9 Самостійна робота №9. Робота з сокетами

9.1 Мета роботи

Вивчити роботу з потоковими сокетами в режимі опиту. Створити сервер котрий буде відповідати на запити клієнтів.

9.2 Короткі теоретичні відомості

Socket API був вперше реалізований в операційній системі UNIX. Зараз цей програмний інтерфейс доступний практично в будь-якій операційній системі. Хоча всі реалізації чимось відрізняються один від одного, основний набір функцій у них збігається. Спочатку сокети використовувалися в про-грамах на C/C++, але в даний час вони є майже в всіх нових мовах програ-мування (Perl, C#, Java та ін.).

Сокети надають дуже потужний і гнучкий механізм взаємодії між процесами (IPC). Вони можуть використовуватися для організації взаємодії програм на одному комп'ютері, по локальній мережі або через Інтернет, що дозволяє вам створювати розподілені додатки різної складності. Крім того, з їх допомогою можна організувати взаємодію з програмами, що працюють під управлінням інших операційних систем.

Сокети підтримують багато стандартних мережевих протоколів (конкретний їх список залежить від реалізації) і надають уніфікований інтерфейс для роботи з ними. Найбільш часто сокети використовуються для роботи в IP-мережах.

Сокети, незалежно від виду, поділяються на три типи: потокові, сирі і дейтаграмні. Потокові сокети працюють з установкою з'єднання, забезпечу-ючи надійну ідентифікацію обох сторін і гарантують цілісність і успішність доставки даних, спираючись на протокол TCP. Дейтаграмні сокети працюють без встановлення з'єднання і не забезпечують ні ідентифікації відправника, ні контролю успішності доставки даних, зате вони швидше по-токових, спираючись на протокол UDP. Сирі сокети, вони надають мож-ливість ручного формування TCP \ IP-пакетів.

Також існує 2 види сокетів:

- синхронні – затримують управління на час виконання операції;
- асинхронні – повертають управління, але продовжують виконувати роботу в фоні та після закінчення повідомляють про це.

Розглянемо роботу з синхронними потоковими сокетами (TCP).

Реалізація сервера:

1) підготувати бібліотеку до використання;
2) створити об’єкт типу Socket;
3) зв’язати цей об’єкт з локальною адресою/портом;
4) перейти в режим очікування;
5) витягнути із черги запит на з’єднання;
6) прийняти/передати дані;
7) закрити сокети, звільнити ресурси.

Реалізація клієнта:

1) підготувати бібліотеку до використання;
2) створити об’єкт типу Socket;
3) з’єднатися з сервером;
4) прийняти/передати дані;
5) закрити сокети, звільнити ресурси.

Датаграмні сокети використовуються в програмах не так часто, оскіль-ки надійність дуже низька. Корисні вони тоді, коли потрібно постійно пере-давати звук і відео по мережі. Оскільки для датаграмних сокетів не потрібно встановлювати з’єднання, використовувати їх значно простіше.

Реалізація сервера:

1) підготувати бібліотеку до використання;
2) створити об’єкт типу Socket;
3) зв’язати цей об’єкт з локальною адресою/портом;
4) прийняти/передати дані;
5) закрити сокети, звільнити ресурси.

Реалізація клієнта:

1) підготувати бібліотеку до використання;
2) створити об’єкт типу Socket;
3) прийняти/передати дані;
4) закрити сокети, звільнити ресурси.

Як ви пам’ятаєте, синхронні сокети затримують управління на час ви-конання операцію, в свою чергу асинхронні повертають управління, але продовжують роботу в фоні та після закінчення повідомляють про це.

У випадку з синхронними сокетами сервер прийнявши нового клієнта працює з ним (обмінюється інформацією), але інші клієнти чекають в черзі, поки сервер не завершить роботу з цим. Асинхронні сокети працюють пара-лельно - витягує клієнта з черги, породжує потік/процес, передає йому де-скриптор клієнта (котрий повернула функція accept), цей потік/процес почи-нає працювати в фоні, в свою чергу сервер знову витягує нового клієнта з черги і так далі.

Асинхронні сокети слід використовувати там, де є велике навантаження при передачі даних.

Реалізація програмної частини залишається такою ж як і при синхрон-них сонетах. Винятком є те, що ми оброблюємо інформацію в потоці, а саме, витягнувши запит від клієнта з черги, ми передаємо його дескриптор в функцію, котра виконується в потоці.

9.3 Хід роботи

1 Написати сервер котрий одночасно буде відповідати на запити від клієнтів. Клієнт повинен надсилати щосекунди повідомлення с поточною датою. На стороні сервера реалізувати перевірку, якщо прийнята дата парна (секунди), у відповідь відсилати дату що прислав клієнт та повідомлення «Число парне», інакше «Число не парне».

2. Написати клієнт-серверний застосунок, котрий буде працювати з датаграмними сокетами. Клієнт повинен надіслати число від 1 до 10, а у відповідь від сервера отримати стільки пові-домлень, скільки було вказано клієнтом.

3. Написати клієнт-серверний застосунок, котрий буде працювати з потоковими або датаграмними сокетами в асинхронному режимі. Клієнт повинен надіслати якийсь файл, наприклад рисунок. Задача сервера, зберегти його в папці, з назвою яка відповідає даті прийняття файлу.

9.4 Зміст звіту

Звіт повинен містити прокоментовані файли з розібраними результатами захоплення пакетів. Звіт також повинен містити файли зон, файл конфігурації сервера і результати перевірок. Наявність коментарів і висновків необхідно.

9.5 Контрольні запитання

1. Що таке сокет? Які існують типи сокетів?
2. Яка функція використовується при створені сокету? Охарактеризуйте параметри котрі передаються у функцію?
3. Опишіть принцип роботи датаграмних сокетів?
4. В чому полягає різниця між датаграмними сокетами і потоковими?
5. Назвіть функції для отримання і відправлення повідомлень?
6. Які поля містить структура sockaddr?
7. В чому полягає різниця між Winsock та сокетами, котрі використовуються в UNIX подібних системах?
8. Які функції потрібно використовувати при роботі з потоками?
9. Коли бажано використовувати асинхронні сокети? Чому ви так вважаєте?

Рекомендована література

1. В.Г. Олифер, Н.А. Олифер. Компьютерные сети. Принципы, технологии, протоколы. СПб., Питер, 2001-672с.:ил, ШЫИТ 5-8046-0133-4
2. Э. Таненбаум. Компьютерные сети. / Пер. с англ. Под ред. д – К.: BHV, 2002 р.
3. Craig Hunt. TCP/IP network adminisration. O'Relly & Associates, Inc, 1994-1998. 472 pages.
4. http://www.freebsd.org/handbook– Проект документирования FreeBSD
5. http://www.isc.org Сайт проектов bind, dhcpd
6. http://www.kernel.org/LDP – Проект документирования Linux
7. http://www.rfc-editor.org RFC center
8. http://www.samba.org Сайт проекта Samba
9. UNIX. Пособие системного администратора. / Пер. с англ. Под ред. д – К.: BHV, 2002 р.
10. Комп'ютерні мережі : навчальний посібник / [Азаров О. Д., Захарченко С. М., Кадук О. В. та ін.] — Вінниця : ВНТУ, 2013. — 371 с. ISBN 978-966-641-543-4