МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

«Графічний інтерфейс налаштування протоколу DHCP в мережах з підтримкою IPv6»

Завідувач випускаючої кафедри — **Довбиш А.С.**

Керівник роботи — **Великодний Д.В.**

Студента групи ІН.м-81н — **Прокоф'єв П.С.**

Нормоконтроль — **Проценко О.Б.**

СУМИ 2020

Сумський державний університет
(назва вузу)

Факультет *ЕлІТ* Кафедра *Комп'ютерних наук*
Спеціальність *«Інформатика»*

Затверджую:
зав.кафедрою _____
_____
"_____" _____ 20___р.

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Прокоф`єву Платону Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) *Графічний інтерфейс налаштування протоколу DHCP в мережах з підтримкою IPv6*

затверджую наказом по інституту від "____" _____ 20 ___ р. № _____

2. Термін здачі студентом закінченого проекту (роботи) _____

3. Вхідні данні до проекту (роботи) _____
_____
_____
_____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Постановка задачі. Аналіз проблеми. 2) Докладне вивчення протоколу DHCP в мережах з підтримкою IPv6, всіх його можливостей і способів роботи. 3) Моделювання мережі в графічних симуляторах. 4) Написання графічного інтерфейсу і тестування його роботи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
_____
_____
_____
_____

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	*Постановка задачі. Аналіз проблеми.*		
2.	*Вивчення протоколу DHCP в мережах з підтримкою IPv6, всіх його можливостей і способів роботи.*		
3.	*Моделювання мережі.*		
4.	*Написання графічного інтерфейсу і тестування його роботи.*		
5.	*Оформлення звіту до магістерської роботи.*		

Студент – дипломник _____
(підпис)

Керівник проекту _____
(підпис)

РЕФЕРАТ

Записка: 59 стор., 19 рис., 4 таблиці, 1 додаток, 15 джерел.

Об’єкт дослідження — протокол DHCP в мережах з підтримкою IPv6.

Мета роботи — розробка графічного інтерфейсу, який дозволить користувачам початківцям налаштовувати з легкістю протокол DHCP в мережах з підтримкою IPv6.

Методи дослідження — моделювання схеми в симуляторі Cisco Packet Tracer. Застосування інструментарію JavaScript, HTML і CSS для розробки графічного інтерфейсу.

Результати — створено графічний інтерфейс в якому можна ввести вхідні дані, а саме IP адреси на інтерфейсах, і в результаті виконання обрахунків отримати налаштування протоколу DHCP в мережах з підтримкою IPv6. Через буфер обміну їх можливо скопіювати, як на віртуальні маршрутизатори в симуляторі Cisco Packet Tracer, так і на реальне мережеве обладнання.

DHCP, IPv6, JAVASCRIPT, CISCO PACKET TRACER, DHCPv6.

ЗМІСТ

ВСТУП

Інтернет та локальна мережа є повсякденними супутниками людства. Для підтримки локального та інтернет-сполучення використовують мережеві протоколи IP. Популярний в останній час мережевий протокол IPv4 потребує деяких змін, зокрема від того, що адреси цього протоколу закінчуються, та людство починає відчувати потребу у його модернізації.

Логічним послідовником цього протоколу став протокол IPv6. Новий протокол потребує налаштування та надання підтримки. Для забезпечення розширення IPv6 покриття потрібні кошти та час. Ці фактори показують на те, що є деяка проблема переходу з протоколу IPv4 до IPv6.

Можливим розв`язанням проблеми складності налаштування протоколу IPv6 може бути графічний інтерфейс налаштування протоколу DHCP (протокол, який дозволяє комп'ютерам автоматично отримувати IP-адресу) в мережах з підтримкою IPv6. Він буде перевірений в процесі наукової роботи. Його можна буде використовувати не тільки в емуляторі, але й на реальному обладнанні Cisco. Розроблений додаток стане в пригоді тим користувачам, які тільки починає свій шлях в побудові мереж із застосуванням IPv6, і досвідченним спеціалістам.

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Протокол DHCP

Протокол динамічної конфігурації хоста (Dynamic Host Configuration Protocol, DHCP) - це служба, яка приймає ряд IP-адрес з підмережі і автоматично призначає їх пристроїв у віртуальній мережі. Вона також може автоматично призначати інші мережеві параметри, такі як шлюзи за замовчуванням і сервери доменних імен (Domain Name System, DNS)[1]. Протокол спрощує адміністрування і використовується для мереж будь-якого розміру (від невеликих до дуже великих). Як сервер DHCP можуть виступати різні пристрої, в тому числі і маршрутизатори Cisco. Сервер DHCP надає різноманітну інформацію хосту, який реєструє на сервері свій IP-адресу. Вся ця інформація надається сервером DHCP:

- IP-адреса
- Маска підмережі
- Доменне ім'я
- Шлюз за замовчуванням (маршрутизатор)
- DNS
- Інформація WINS

Сервер DHCP здатний поставляти додаткові відомості, але перераховані в списку дані є найбільш типовими[7].

Протокол динамічної конфігурації хоста (DHCP) часто використовується в мережах, щоб дозволити пристроям автоматично отримувати їх мережеву конфігурацію під час першого підключення до мережі. В основному він розширюється на попередньому протоколі завантаження (BOOTP) і використовує ті ж порти UDP, числа 67 і 68. Сам протокол визначено в RFC 2131, а параметри конфігурації - в RFC 2132.

DHCP дозволяє надати мінімально загальну конфігурацію для всіх робочих станцій користувача. Тоді будь-хто може просто підключити пристрій до мережі

в будь-який момент, і DHCP подбає про отримання IP-адреси, яка буде працювати в цьому місці. Це мінімізує помилки через ручну конфігурацію, централізує контроль над конфігурацією інформаціїї та значно скорочує витрати техніків, тому що будь-хто може підключити пристрій до мережі.

У мережі DHCP є три різних типи елементів. Має бути клієнт і сервер, і якщо ці два елементи не в одній мережі 2 рівня, то там також повинен бути проксі, який зазвичай працює на маршрутизаторі. Проксі-сервер потрібен тому, що клієнтський пристрій спочатку не знає власної IP-адреси, тому він повинен надсилати трансляцію 2 рівня, щоб знайти сервер, який має цю інформацію. Маршрутизатор повинен ретранслювати ці трансляції на сервер DHCP і пересилати відповідь назад на правильну адресу 2 рівня, щоб правильний пристрій отримував правильну інформацію про конфігурацію.

Історично, єдиною роллю проксі-сервера була роль роутера в BOOTP або DHCP. Однак маршрутизатори Cisco нещодавно додали функцію клієнта DHCP та сервера. Це Розділ покаже приклади конфігурації для всіх трьох цих функцій, хоча конфігурація сервера найскладніша, тому більшість рецептів буде зосереджена на цьому.

Обмін DHCP починається з клієнтського пристрою, наприклад робочої станції кінцевого користувача. Зазвичай цей пристрій завантажуватиметься та підключатиметься до мережі без попередньо налаштованої мережевої інформації. Він не знає його IP-адресу, адресу свого маршрутизатора чи його підмережі або маски мережі, і він навіть не знає адреси сервера, який надаватиме ці фрагменти інформації. Тож це робить єдине, що він може зробити, і посилає пакет широкомовної передачі UDP, який шукає сервер[3].

Більшість мереж DHCP будь-якого розміру включають два або більше DHCP-сервери для надмірності. Кінцеві пристрої зазвичай просто потребують цього сервера під час запуску, але вони будуть зовсім не працювати без цього. Тому надмірність важлива. Це також означає, що це не так незвично для кінцевого пристрою, щоб побачити кілька відповідей на запит DHCP. Зазвичай він

просто використовуватиме першу відповідь. Однак це також підкреслює важливість гарантуючи, що всі сервери DHCP поширюють однакову інформацію. Їх бази даних параметрів конфігурації кінцевих пристроїв повинні бути синхронізовані.

Потім кінцевий пристрій запитує інформацію про конфігурацію з одного з серверів. Це повинен точно вказати, які опції він вимагає. Сервер не потребує відповіді всі запитувані параметри, однак він не може запропонувати додаткову неперевірену інформацію клієнт, навіть якщо він має додаткову інформацію в своїй базі даних. Це важлива невелика деталь, яку слід пам’ятати, оскільки вона може бути дуже заплутаною, коли кінцевий пристрій має кілька налаштовані вручну параметри, які не замінюються інформацією на сервері.

Оскільки подвійні IP-адреси можуть спричинити серйозні проблеми в мережі, більшість DHCP сервери відстежують конфлікти адрес. Вони роблять це, намагаючись провести PING кожної IP-адреси перш ніж сказати кінцевому пристрою, що безпечно ним користуватися. І багато клієнтів DHCP також будуть двічі перевіряти, чи адреса вже не використовується, попередньо надсилаючи запит ARP використовуючи його. Однак жодна з цих перевірок не є обов'язковою, а деякі клієнти DHCP і сервери не перевіряють перед використанням адреси.

Однією з важливих особливостей DHCP є можливість виділення IP-адрес лише для налаштованого періоду часу, який називається періодом оренди. Якщо клієнтський пристрій хоче зберегти його IP-адресу довше цього періоду, вона повинна поновити оренду до закінчення терміну її дії.

Клієнти можуть подовжувати оренду так часто, як їм заманеться.

Сервер може виділяти IP-адреси з пулу за принципом «перший в черзі, перший сервіс» або він може асоціювати IP-адреси з MAC-адресами кінцевого пристрою, щоб гарантувати, що певний клієнт завжди отримує однакову адресу[4].

Основні загрози DHCP

DoS-атака на сервер DHCP - ця атака має намір запобігти клієнтам отримати IP-адресу та інші параметри, що надаються DHCP, наприклад шлюз за замовчуванням, підмережа IP, адреси сервера DNS тощо . Ця атака може вплинути на хости, коли вони спочатку підключаються до мережі, або хости, що поновлюють DHCP-оренду по мірі її закінчення. У цьому випадку постраждалі хости втрачають мережевий зв'язок. DHCP-сервери також можуть бути піддані DoS-атакам з ресурсами, коли цільовий DHCP-сервер оснащений багатьма хибними DHCP-запитами, кожен з яких має унікальну MAC-адресу. У разі успіху ця атака може вичерпати пул адрес сервера DHCP, не даючи дійсним хостам отримати IP-адресу та мережеве підключення.

Spoofing атака:

Якщо зловмисник може маскуватися під DHCP, DNS або NTP-сервер, існує загроза посилання підробленої інформації про IP-шлюз клієнтам DHCP, яка дозволяє зловмиснику перехоплювати трафік, полегшуючи атаки MiTM ((англ. Man in the middle) термін з криптографії , що визначає ситуацію, коли криптоаналітик (атакуючий) здатний читати та видозмінювати по своєму бажанню повідомлення, якими обмінюються користувачі, причому жоден з них не може здогадатися про його присутність в каналі.), підслуховування та введення помилкових даних. Інформація, зібрана з перехоплених пакетів (наприклад, паролі), також може спричинити подальші атаки. Посилання неправдивої інформації DHCP також може заважати клієнтам спілкуватися в мережі, викликаючи стан DoS[5].

На рисунку 1.1 показана атака " man-in-the-middle", яка використовує DHCP. Законний сервер DHCP сидить на головному сайті, тоді як зловмисник сидить у локальній локальній мережі, виконуючи функції сервера DHCP.

Наступні кроки пояснюють, як ПК нападника може стати man-in-the-middle на рисунку 1.1:

Крок 1. PC-B запитує IP-адресу за допомогою DHCP.

Крок 2. ПК-зловмисник відповідає і призначає хороший IP / маску, але використовуючи власну IP-адресу як шлюз за замовчуванням.

Крок 3. PC-B надсилає кадри даних зловмиснику, думаючи, що зловмисник є шлюзом за замовчуванням.

Крок 4. Зловмисник пересилає копії пакетів, стаючи man-in-the-middle.

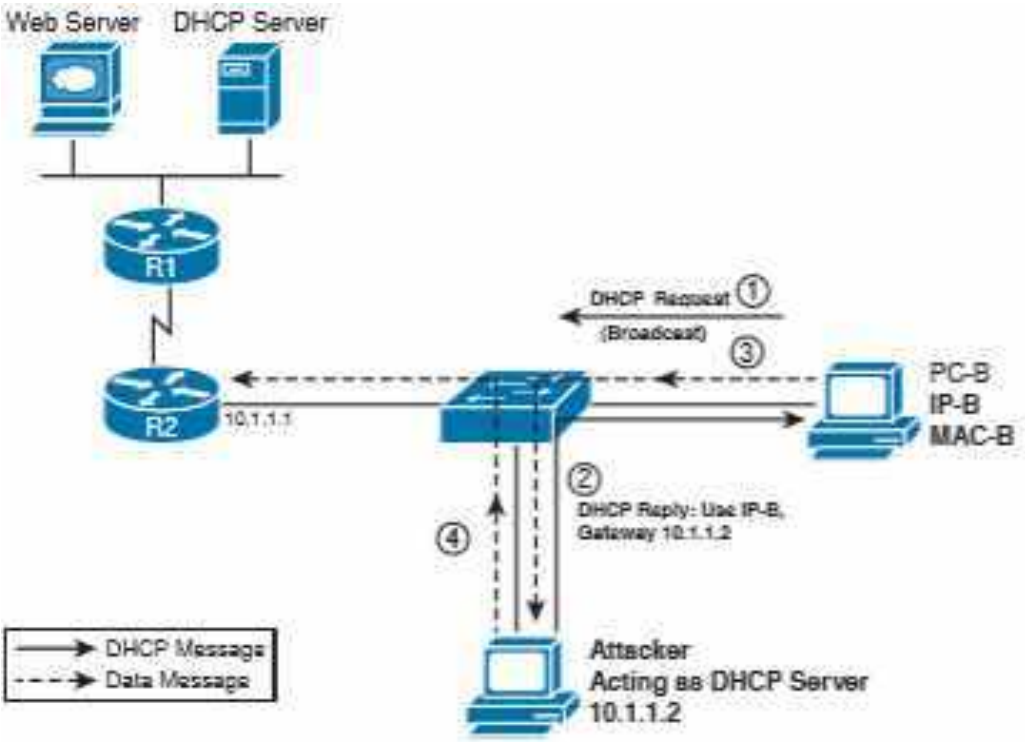

Рисунок 1.1 – атака man-in-the-middle[6].

Перевірка DHCP

Перевірка DHCP - це функція безпеки, яка діє як брандмауер між ненадійними хостами та надійними серверами DHCP. Функція перевірки DHCP виконує такі дії:

■ Перевіряє безладдя DHCP, отримані з ненадійних джерел, і фільтрує недійсні повідомлення. ■ Обмежуйте швидкість трафіку DHCP з надійних та ненадійних джерел.

■ Створює та підтримує базу даних прив'язки DHCP, який містить інформацію про ненадійних хостів з орендованими IP-адресами.

■ Використовує базу даних прив'язки DHCP для відстеження, щоб перевірити наступні запити від недовірених хостів

Перевірка DHCP включається на основі VLAN. За замовчуванням функція неактивна для всіх VLAN. Ви можете ввімкнути цю функцію в одній VLAN або в діапазоні VLAN.

Атаки підробки DHCP мають місце, коли пристрої навмисно намагаються генерувати достатню кількість запитів DHCP, щоб вичерпати кількість IP-адрес, призначених для пулу DHCP.

Функція перевірки DHCP визначає, яким джерелам трафіку довіряти чи ні. Ненадійне джерело може ініціювати дорожні атаки або інші ворожі дії. Щоб запобігти подібним атакам, функція перевірки DHCP фільтрує повідомлення та обмежує швидкість руху трафіку з ненадійних джерел.

Для впровадження перевірки DHCP у мережі потрібно виконати наступні кроки:

Крок 1. Визначте та налаштуйте DHCP-сервер.

Крок 2. Увімкнути прослуховування DHCP хоча б однієї VLAN. За замовчуванням прослуховування DHCP неактивне для всіх VLAN.

Крок 3. Переконатись, що DHCP-сервер підключений через надійний інтерфейс. За замовчуванням стан довіри всіх інтерфейсів не є довірчим.

Крок 4. Налаштувати агент баз даних DHCP, що відслідковується. Цей крок гарантує, що записи в базу даних будуть відновлені після перезавантаження або переключення.

Крок 5. Увімкнути глобальну перевірку DHCP. Функція відстеження DHCP не активна, поки не виконати цей крок.

Приклад налаштування DHCP перевірки:

```
! Enable DHCP Snooping Globally
sw2(config)# ip dhcp snooping
! Enable DHCP Snooping on VLAN 10
sw2(config)# ip dhcp snooping vlan 10
! Configure Interface Fa1/0/24 as a Trusted interface
sw2(config)# interface fa1/0/24
sw2(config-if)# ip dhcp snooping trust
! Configure the DHCP snooping database agent to store the bindings at a given location
sw2(config)# ip dhcp snooping database tftp://10.1.1.1/directory/file
sw2(config)# exit
sw2#
! Verify DHCP Snooping Configuration
sw2# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
10
DHCP snooping is operational on following VLANs:
none
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is enabled
   circuit-id default format: vlan-mod-port
   remote-id: 000f.90df.3400 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface                  Trusted    Allow option    Rate limit (pps)
-----------------------    -------    ------------    ----------------
FastEthernet1/0/24         yes        yes             unlimited
  Custom circuit-ids:
```

Рисунок 1.2 – налаштування DHCP перевірки. [2]

1.2 Протокол IPv6

Протокол IP версії 6 (IP version 6, IPv6) є вдосконаленим варіантом версії 4 протоколу IP. Для ідентифікації потоків в заголовок пакета поміщається спеціальний ідентифікатор. Спочатку називався IPng (IP next generation)[8].

IP - це місце сортування та доставки пакетів у стеку TCP / IP. На цьому шарі кожен вхідний або вихідний пакет називається дейтаграмою. Кожна IP-дейтаграма несе вихідну IP-адресу відправника та цільову IP-адресу призначеного одержувача. На відміну від MAC-адрес, IP-адреси в дейтаграмі залишаються однаковими протягом усієї поїздки пакету через Інтернет-роботу.

Функціонування IP є центральним для стека TCP / IP - усі інші протоколи TCP / IP використовують IP - і всі дані проходять через нього. IP є протоколом без підключення і має деякі обмеження. Якщо IP намагається здійснити доставку пакету, і в процесі пакет втрачається, доставляється з послідовності, дублюється або затримується, про це не повідомляє ні відправник, ні одержувач. Підтвердження пакетів обробляється транспортним протоколом вищого рівня, таким як TCP.

IP відповідає за адресу та маршрутизацію пакетів між хостами та визначає, чи потрібна фрагментація. Фрагментація передбачає розбиття дейтаграми на більш дрібні фрагменти для оптимізованої маршрутизації. IP-протокол буде фрагментувати пакети перед відправленням, а також збиратиме їх, коли вони досягнуть місця призначення.

Формат адреси IPv6

IPv6 використовує 16-байтові шістнадцяткові числові поля, розділені двокрапками (:), щоб представити 128-бітний формат адресації, що робить подання адреси менш громіздким та схильним до помилок. Ось приклад дійсної адреси IPv6:

2001: db8: 130F: 0000: 0000: 09C0: 876A: 130B

Крім того, щоб скоротити адресу IPv6 і полегшити уявлення адреси, IPv6 використовує такі умови:

■ Провідні 0 в адресному полі необов'язкові і можуть бути стиснені.

Наприклад: Наступні шістнадцяткові числа можуть бути представлені у вигляді стислого формату:

■ Приклад 1: 0000 = 0 (стисла форма)

■ Приклад 2: 2001: db8: 130F: 0000: 0000: 09C0: 876A: 130B = 2001: db8: 130F: 0: 0: 9C0: 876A: 130B (стисла форма)

■ Пара колонок (: :) являє собою послідовні поля 0. Однак пара колонок дозволена лише один раз у дійсній IPv6-адресі.

■ Приклад 1: 2001: db8: 130F: 0: 0: 9C0: 876A: 130B = 2001: db8: 130F :: 9C0: 876A: 130B (стисла форма)

■ Приклад 2: FF01: 0: 0: 0: 0: 0: 1 = FF01 :: 1 (стисла форма)

Аналізатор адреси може легко визначити кількість пропущених 0 в IPv6-адресі, розділивши дві частини адреси та заповнивши 0, поки не закінчиться 128-бітна адреса. Однак якщо дві пари колонок розміщені за однією адресою, немає можливості визначити розмір кожного блоку з нолями. Використання :: робить велику адресу IPv6 дуже малою.

Мережевий префікс

У IPv6 є посилання на префікси, які, з точки зору IPv4, слабо прирівнюються до підмереж. Префікс IPv6 складається з самих лівих бітів і виступає як мережевий ідентифікатор. Префікс IPv6 представлений з використанням формату префікса IPv6 або префікса так само, як адреса IPv4 представлена в безкласовій нотації маршрутизації між доменами (CIDR).

Змінна довжина префікса - це десяткове значення, яке вказує на кількість суміжних бітів високого порядку адреси, що утворюють префікс, який є мережевою частиною адреси. Наприклад, 2001: db8: 8086: 6502 :: / 64 є прийнятним префіксом IPv6. Якщо адреса закінчується подвійною двокрапкою, кінцеву подвійну двокрапку можна опустити. Тож та сама адреса може бути записана як 2001: db8:

8086: 6502/64. У будь-якому випадку довжина префікса записується у вигляді десяткового числа 64 і являє крайні ліві біти адреси IPv6. Аналогічна адреса в IPv4 буде xxx.xxx.xxx.xxx/16[9].

Включення протокола IPv6 на маршрутизаторі

Для включення протокола IPv6 на маршрутизаторі використовується команда ipv6 unicast-routing, наприклад:

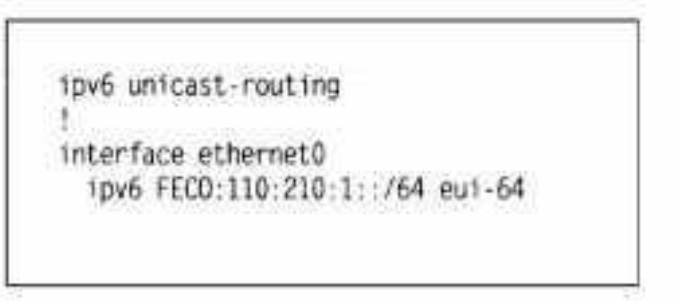

Рисунок 1.3 – включення IPv6 на маршрутизаторі[10].

Типи адрес IPv6

Існує значна різниця в вимогах до IP-адреси між хостом IPv4 та хостом IPv6. Хост IPv4 зазвичай використовує одну IP-адресу, але хост IPv6 може мати більше однієї IP-адреси.

Існує три основні типи адрес IPv6:

■ Unicast: адреса для одного інтерфейсу. Пакет, який надсилається на одноадресну адресу, доставляється в інтерфейс, визначений цією адресою.

■ Anycast: адреса для набору інтерфейсів, які зазвичай належать до різних вузлів. Пакет, надісланий на адресу anycast, доставляється до найближчого інтерфейсу, як визначено протоколами маршрутизації, які використовуються та ідентифіковані адресою anycast.

■ Багатоадресна передача: адреса для набору інтерфейсів (у заданій області), які зазвичай належать до різних вузлів. Пакет, надісланий на адресу багатоадресної пошти, доставляється до всіх інтерфейсів, ідентифікованих адресою багатоадресної пошти (у заданій області).

Type of Address	Purpose	Prefix	Easily Seen Hex Prefix(es)
Global unicast	Unicast packets sent through the public Internet	2000::/3	2 or 3
Unique local	Unicast packets inside one organization	FD00::/8	FD
Link Local	Packets sent in the local subnet	FE80::/10	FE80
Multicast (link local scope)	Multicasts that stay on the local subnet	FF02::/16	FF02

Таблиця 1.1 – типи адрес IPv6[9].

Управління адресами та їх призначення

Існує чотири способи налаштування адреси хоста в IPv6:

■ Статична конфігурація: Подібно до IPv4, адресу хоста, маску та адресу шлюзу визначають вручну.

■ Статична автоконфігурація (SLAAC): У цьому випадку хост автономно налаштовує власну адресу. Повідомлення про запити маршрутизаторів надсилаються вузлами завантаження для запиту реклами маршрутизаторів (RA) для налаштування інтерфейсів (RFC 2462).

■ Статична конфігурація DHCPv6: Хост використовує протокол конфігурації динамічного хоста (DHCP), щоб отримати свою IPv6 адресу. Це управління адресами подібне до поведінки IPv4 (RFC 3315).

■ DHCP автоконфігурація: Хост використовує SLAAC, а також DHCP для отримання додаткових параметрів, таких як TFTP Server, WINS тощо.

Статична конфігурація

Як і в IPv4, адресу хоста можна визначити статично. У цьому випадку IPv6-адреса, маска та адреса шлюзу визначаються вручну на хості.

Конфігурація статичної адреси зазвичай використовується для конфігурації інтерфейсу маршрутизатора, але, швидше за все, не використовується для хостів у IPv6. Майте на увазі, що використання статичної конфігурації означає, що всі функції автоконфігурації, надані IPv6, будуть відключені.

Статична автоконфігурація

Вузли можуть використовувати статичну автоконфігурацію IPv6 для генерування адрес без необхідності сервера DHCP. IPv6 адреси формуються шляхом поєднання мережних попередніх виправлень з ідентифікатором інтерфейсу. Для інтерфейсів із вбудованими ідентифікаторами Інституту інженерів електротехніки та електроніки (IEEE) ідентифікатор інтерфейсу, як правило, походить від ідентифікатора IEEE.

Функція автоконфігурації адреси вбудована в протокол IPv6 для полегшення внутрішньосезонного управління адресами, що дозволяє великій кількості IP-хостів легко відкривати мережу та отримувати нові та глобально унікальні адреси IPv6, пов'язані з їх місцезнаходженням. Функція автоконфігурації дозволяє підключати до Інтернету розгортання нових споживчих пристроїв, таких як стільникові телефони, бездротові пристрої, побутова техніка тощо. Як результат, мережеві пристрої можуть підключатися до мережі без ручної конфігурації та без будь-яких серверів, таких як DHCP-сервери. Нам потрібно трохи детальніше розглянути принципи, які стоять за цією ознакою.

Маршрутизатор на локальному посиланні надсилає інформацію про тип мережі через повідомлення RA, такі як префікс локальної посилання та маршрут за замовчуванням у своїх рекламних оголошеннях маршрутизатора. Маршрутизатор надає цю інформацію всім вузлам локального посилання.

Потім хост може побудувати свою адресу, додавши ідентифікатор хоста до префікса / 64, отриманого від маршрутизатора. Як результат, хости Ethernet можуть автоматично налаштувати себе, додавши свою 48-бітну адресу рівня посилання (MAC-адресу) у розширеному універсальному ідентифікаторному форматі EUI-64-біт до 64 біт префікса локальної посилання, рекламований маршрутизатором.

Ще одним надзвичайно сприятливим аспектом цього підходу є легкість, з якою можна реалізувати переосмислення адреси. У мережах IPv6 функція авто-

конфігурації робить перенумерування існуючої мережі простою та відносно простою порівняно з IPv4. Маршрутизатор надсилає новий префікс від нового постачальника вище за потоком у своїх оголошеннях про маршрутизатор. Хости в мережі автоматично вибирають нову префікс з реклами маршрутизатора, а потім використовують її для створення своїх нових адрес. Як результат, перехід від постачальника A до B стає керованим для мережевих операторів.

Статична конфігурація DHCPv6

Наразі багато підприємств використовують DHCP для поширення адрес своїм хостам. IPv6 можна розгорнути за допомогою того ж механізму DHCP.

Процес отримання даних конфігурації для клієнта в IPv6 схожий з процесом в IPv4. Однак DHCPv6 використовує багатоадресну передачу для багатьох своїх повідомлень. Спочатку клієнт повинен спочатку виявити наявність маршрутизаторів по посиланню за допомогою повідомлень про виявлення сусідів. Якщо маршрутизатор знайдений, клієнт вивчає рекламу маршрутизатора, щоб визначити, чи слід використовувати DHCP. Якщо рекламні маршрутизатори дозволяють використовувати DHCP на цьому посиланні (відключення прапора автоконфігурації та включення прапора керованого в повідомленнях RA дозволяє хосту використовувати DHCPv6 для отримання адреси IPv6), клієнт запускає фазу виклику DHCP, щоб знайти DHCP-сервер.

Використання DHCPv6 надає наступні переваги:

■ Більше контролю, ніж статична автоконфігурацію.

■ Його можна використовувати одночасно з статичною автоконфігурацію.

■ Його можна використовувати для перенумерування.

■ Його можна використовувати для автоматичної реєстрації доменних імен хостів за допомогою динамічного DNS.

■ За допомогою нього можна делегувати префікс IPv6 для маршрутизаторів маршрутизаторів обладнання клієнтів (CPE).

Автоконфігурація DHCP

Автоконфігурація DHCPv6 зазвичай поєднує статичну автоконфігурацію для призначення адреси з обміном DHCPv6 для всіх інших параметрів конфігурації. У цьому випадку DHCPv6 використовується тільки для хоста для отримання додаткових параметрів, таких як TFTP-сервер, DNS-сервер тощо.

Хост будує свою адресу, додаючи ідентифікатор хоста до префіксу / 64, отриманого від маршрутизатора, а потім видає DHCP-повідомлення з проханням на сервер DHCP[9].

Пошук Сусіда IPv6

Ввиявлення сусідів IPv6 (ND) - це набір повідомлень і процесів, які визначають відносини між двома сусідніми вузлами IPv6. У IPv6-й побудований на вершині протоколу ICMPv6, який визначений в RFC 2463. IPv6 ND замінює такі протоколи, як ARP, ICMP redirect і ICMP router discovery messages, використовувані в IPv4. Як IPv6 ND, так і ICMPv6 мають вирішальне значення для роботи IPv6.

IPv6 ND визначає п'ять пакетів ICMPv6 для надання вузлам інформації, яку вони повинні знати і повинні знати до встановлення зв'язку:

- Запит маршрутизатора (ICMPv6 тип 133, код 0)
- Реклама маршрутизатора (Тип ICMPv6 134, код 0)
- Залучення сусідів (ICMPv6 тип 135, код 0)
- Сусідська реклама (ICMPv6 тип 136, код 0)
- Перенаправлення повідомлення (ICMPv6 тип 137, код 0)

Коли інтерфейс включений, хости можуть відправляти запит маршрутизатора (RS), який змушує маршрутизатори генерувати оголошення маршрутизатора негайно, а не в їх наступний запланований час. Коли надсилається повідомлення RS, поле адреси джерела встановлюється на MAC-адресу картки мережевого інтерфейсу відправки (NIC). Поле адреси призначення має значення

33:33:00:00:00:02 у заголовку Ethernet. У заголовку адреси призначення встановлено 33: 33: 00: 00: 00: 02 у заголовку Ethernet. У заголовку IPv6 поле вихідної адреси встановлюється або локальною IPv6 адресою посилання, призначеної інтерфейсу відправки, або не вказаною адресою IPv6 (: :). Цільовою адресою встановлено адресу All Router multicast з локальним діапазоном зв'язку (FF02: 2), а межа стрибка встановлена на 255.

Маршрутизатори оголошують свою присутність разом з різними параметрами зв'язку та Інтернету або періодично, або у відповідь на запит маршрутизатора. Оголошення маршрутизатора (RAs) містять префікси, які використовуються для визначення на каналі та/або конфігурації адреси, запропонованого граничного значення стрибка, максимального блоку передачі (MTU) і так далі. У заголовку Ethernet повідомлення RA поле адреси джерела вставляється в NIC відправлення; поле адреси призначення встановлюється як 33:33:00:00:00:01 або як одноадресний MAC-адреса хоста, який відправив повідомлення RS з одноадресної адреси. Як і у випадку з повідомленням RS, поле адреси джерела задається локальною адресою каналу, призначеною інтерфейсу відправки; адреса D estination задається або багатоадресною адресою всіх вузлів з локальною областю дії каналу (FF02: 1), або одноадресною IPv6-адресою хоста, який відправив повідомлення RS. Поле обмеження стрибка має значення 255.

Запит сусіда (NS) надсилається вузлом для визначення адреси сусіднього рівня зв'язку або для перевірки того, що сусід все ще доступний через кешовану адресу рівня зв'язку. Сусідні запити також використовуються для виявлення дублікатів адрес (DAD). У заголовку Ethernet повідомлення NS кінцева MAC-адреса відповідає запитаній адресі вузла призначення. В одноадресному NS-повідомленні поле Адреса призначення задається як одноадресний MAC-адреса сусіда. У заголовку IPv6 вихідна адреса встановлюється на IPv6-адресу відправного інтерфейсу або, під час DAD, на невизначену адресу (::). Для багатоадресного NS адреса призначення задається як цільова адреса запитуваного вузла. Для

одноадресної НС, пункт призначення знаходиться на одноадресної передачі IPv6-адреса мети.

Оголошення сусіда (NA) - це відповідь на повідомлення про запрошення сусіда. Вузол також може надсилати небажані оголошення сусіда, щоб оголосити про зміну адреси рівня зв'язку. У заголовку Ethernet для домагався НС, MAC-адреса призначення знаходиться в одноадресний MAC-адресу початкового відправника НС. Для незапрошеного NA кінцевий MAC встановлюється в значення 33:33:00:00:00:01, який є адресою багатоадресної розсилки всіх вузлів локальної області зв'язку. У заголовку IPv6 вихідна адреса встановлюється в одноадресну адресу IPv6, призначену на відправному інтерфейсі. Кінцева IPv6-адреса для запитуваного NA встановлюється на IPv6-адресу одноадресної розсилки відправника вихідного повідомлення NS. Для незапрошеного na поле Призначення встановлюється на адресу багатоадресної розсилки всіх вузлів локальної області зв'язку (FF02:: 1).

Повідомлення про перенаправлення (PM) використовується маршрутизаторами для інформування господарів краще першого стрибка до місця призначення. У заголовку Ethernet, MAC-адреса призначення розташований на одноадресний Mac вихідного відправника. У заголовку IPv6 поле вихідної адреси встановлюється в одноадресний IPv6-адреса відправляє інтерфейсу, а адреса призначення - в одноадресний адресу вихідного хоста.

Щоб увімкнути виявлення сусідів, першим кроком є включення IPv6 або налаштування адреси IPv6 на інтерфейсі. Адреса IPv6 налаштований за допомогою команди IPv6-адреса IPv6-адреса [eui64] або команди IPv6 використовуватилокальні - тільки адреса. Параметр команди eui64 налаштовує адресу IPv6 у форматі EUI64. Параметр команди uselink-local-only вручну налаштовує локальну адресу в інтерфейсі замість використання автоматично призначеної локальної адреси посилання. Коли IPv6-адреса налаштована на обох сторонах каналу і одна зі сторін ініціює ping, починається процес ND і встановлюється сусідство IPv6.

Сусід IPv6 проглядається за допомогою команди show ipv6 neighbor [detail]. Рисунок 1.4 демонструє сусідство IPv6 між двома комутаторами[11].

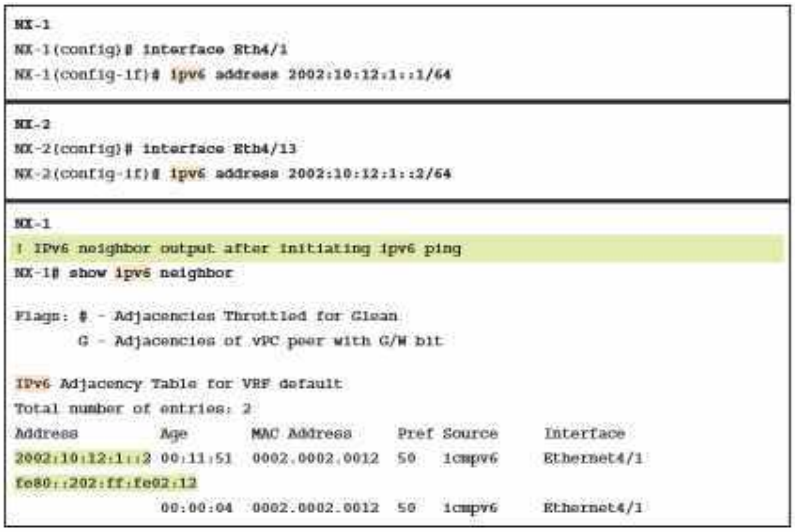

Рисунок 1.4 - демонстрація сусідства IPv6 між двома комутаторами[11].

Історичні причини IPv6

В останні 40+ років Інтернет перейшов від зародження до величезного впливу у світі. Вперше вона виросла за допомогою досліджень у університетах, починаючи з ARPANET за часів Інтернету в кінці 1960-х до 1970-х. Інтернет продовжував швидко розвиватися у 1980-х роках, при цьому швидкий ріст Інтернету все ще в першу чергу був зумовлений дослідженнями та університетами, які приєдналися до цього дослідження. На початку 1990-х Інтернет почав трансформуватися, щоб дозволити комерцію, дозволяючи людям продавати послуги та товари через Інтернет, що призвело до ще одного крутого спайка вгору в зростанні Інтернету. Врешті-решт, фіксований доступ до Інтернету (насамперед через телефонний набір, цифрову абонентську лінію [DSL] та кабель) став загальним явищем, після чого розповсюджене використання Інтернету з мобільних пристроїв, таких як смартфони.

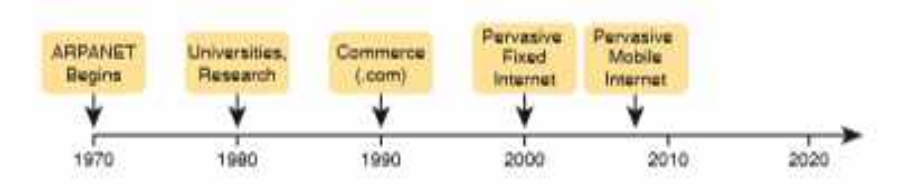

Рисунок 1.5 - деякі основні події в Інтернеті[12].

Компанії продовжували дотримуватися правил, запитуючи публічні мережі IP, і було зрозуміло, що поточна схема призначення адрес не може продовжуватися без деяких змін. Простіше кажучи, кількість мереж класів A, B і C, підтримуваних 32-бітовою адресою у версії 4 версії IP (IPv4), була недостатньою для підтримки однієї загальнодоступної мережі в одній організації, а також надання достатньої кількості IP-адрес у кожній компанії. Протягом 90-х років Інтернет-спільнота наполегливо працювала над вирішенням цієї проблеми, придумуючи зокремв нову версію IP (IP версія 6 [IPv6]) із значно більшими адресами - 128 біт[13].

Це означає, що IPv4 підтримує максимум 232 IP-адреси, що означає приблизно 4,29 мільярда загальних адрес. IPv6, оскільки він використовує 128 біт, підтримує максимум 2128 доступних адрес:

340,282,366,920,938,463,463,374,607,431,768,211,456[9].

Ця версія протоколу IP повинна забезпечити необхідну кількість адрес як на поточний момент, так і в майбутньому.

Для представлення 128-бітового адреси в протоколі IPv6 використовується запис з восьми шістнадцятибітових чисел, що подаються у вигляді чотирьох шістнадцяткових цифр, як це показано на Рисунку 1.7. Групи з чотирьох шістнадцяткових цифр розділені двокрапками, нулі в старших позиціях можуть бути опущені.

Internet-протокол версии 4 (IPv4) 4 октета
11010001.11011100.11001001.01110001
209.156.201.113
4,294,467,295 IP-адресов

Internet-протокол версии 6 (IPv6) 16 октетов
11010001.11011100.11001001.01110001.11010001.11011100. 110011001.01110001.11010001.11011100.11001001. 01110001.11010001.11011100.11001001.01110001
A524:72D3:2C80:DD02:0029:EC7A:002B:EA73
3.4 x 10^{38} IP-адресов

Рисунок 1.6 - Порівняння стандартів IPv4 і IPv6. [13]

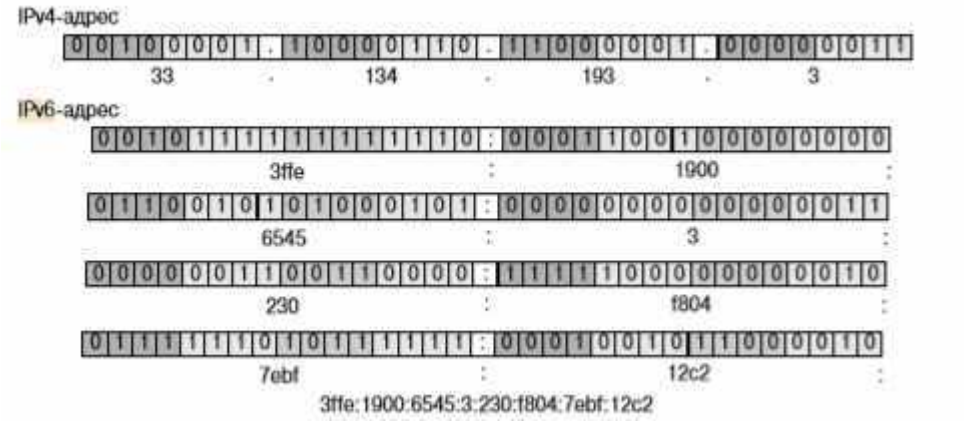

Рисунок 1.7 - Формати адрес IPv4 і IPv6. [13]

Розробка та планування технології зайняли роки, перш ніж протокол IPv6 поступово почав використовуватися в окремих мережах. У перспективі стандарт IPv6 може замінити IPv4 в якості домінуючого протоколу в мережі Internet[14].

Технології переходу IPv6

Спочатку, як вважалося, успіх IPv6 залежить від нових програм, що над ним працюють. Однак стає дуже зрозуміло, що вичерпання IPv4 в кінцевому підсумку стане рушієм для прийняття IPv6. Ключовою частиною будь-якого хорошого дизайну IPv6 є його здатність інтегруватись у та співіснувати з існуючими мережами IPv4. Веб-хости IPv4 та IPv6 потребують співіснування протягом значного періоду часу під час постійної міграції з IPv4 на IPv6, а розробка стратегій,

інструментів та механізмів переходу була частиною базового дизайну IPv6 з самого початку - існує три технології переходу IPv6: dual stack, tunneling, and translation.

Dual Stack

Dual stack - це основна стратегія, яка використовується для великих агентств, які впроваджують IPv6. Він передбачає налаштування пристроїв, щоб мати змогу одночасно запускати IPv4 та IPv6. Для зв'язку IPv4 використовується стек протоколів IPv4, а для зв'язку IPv6 використовується стек протоколів IPv6.

Програми вибирають між використанням IPv4 або IPv6 на основі відповіді на запити DNS. Додаток вибирає правильну адресу залежно від типу трафіку IP. Оскільки подвійний стек дозволяє хостам одночасно досягати наявного вмісту IPv4 та контенту IPv6, коли він стає доступним, подвійний стек пропонує дуже гнучку стратегію прийняття. Однак, оскільки адреси IPv4 все ще потрібні, подвійний стек не є довгостроковим рішенням для вирішення проблем з виснаженням.

Подвійний стек також уникає необхідності перекладати між стеками протоколів. Переклад є дійсним механізмом прийняття, але він вносить операційні складності та нижчу ефективність. Оскільки хост автоматично вибирає правильний транспорт, який буде використаний для досягнення пункту призначення на основі інформації DNS, не повинно виникати необхідності перекладати між хостом IPv6 та сервером IPv4[9].

Tunneling

Ще одним інструментом для підтримки переходу від IPv4 до IPv6 є тунелювання. Існує багато типів тунелювання, але в цьому випадку функція тунелювання зазвичай приймає пакет IPv6, відправлений хостом, і інкапсулює його в пакет IPv4. Потім пакет IPv4 може бути переданий через існуючу IPv4-Мережу, а інший пристрій видалить заголовок IPv4, розкриваючи вихідний пакет IPv6. Ця концепція дуже схожа на VPN-тунель, як описано в главі 17 " віртуальні приватні мережі.”

На Рисунку 1.8 показаний типовий приклад з типом тунелю, який зазвичай називають тунелем IPv6-toIPv4, що означає IPv6 всередині IPv4. На малюнку показаний приклад корпоративної мережі, в якій хости деяких локальних мереж перейшли на IPv6, але ядро мережі все ще працює по протоколу IPv4. Це може мати місце на початковому етапі тестування всередині підприємства, або це може бути зазвичай зроблено за допомогою інтернет-провайдера на основі IPv4, у якого є клієнти, які бажають перейти на IPv6.

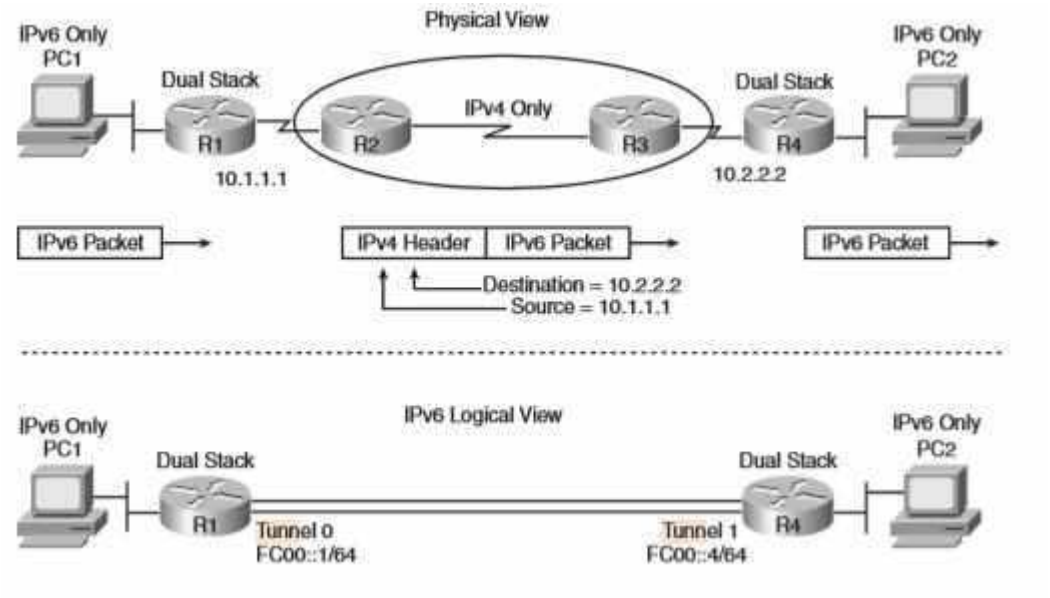

Рисунок 1.8 - Приклад тунелю IPv6-to-IPv4, фізичне та логічне представлення[15].

На Рисунку 1.8 показано, що PC1 на основі IPv6 відправляє пакет IPv6. Потім маршрутизатор R1 інкапсулює або тунелює пакет IPv6 в новий заголовок IPv4 з цільовою IPv4-адресою адреси на маршрутизаторі R4. Маршрутизатори R2 і R3 з радістю пересилають пакет, тому що він має звичайний заголовок IPv4, в той час як R4 деінкапсулює вихідний пакет IPv6, перенаправляючи його на ПК2, заснований на IPv6. Це називається тунелем частково тому, що пакети IPv6 всередині тунелю не можуть бути помічені під час проходження тунелю; маршрутизатори в середині мережі, R2 і R3 в цьому випадку, сприймають пакети як пакети IPv4.

Існує кілька типів тунелів IPv6-IPv4. Для виконання тунелювання, показаного маршрутизаторами на Рисунку 1.8, можна використовувати перші три з наступних типів тунелів, причому четвертий тип (тунелі Teredo) використовується хостами:

■ Manually configured tunnels (MCT): проста конфігурація, в якій створюються тунельні інтерфейси, Тип інтерфейсу віртуального маршрутизатора, з конфігурацією, що посилається на IPv4-адреси, використовувані в заголовку IPv4, який інкапсулює пакет IPv6.

■ Dynamic 6to4 tunnels: цей термін відноситься до певного типу динамічно створюваного тунелю, зазвичай виконуваного в Інтернеті IPv4, в якому IPv4-адреси кінцевих точок тунелю можуть бути динамічно знайдені на основі адреси призначення IPv6.

■ Intra-site Automatic Tunnel Addressing Protocol (ISATAP): ще один метод динамічного тунелювання, який зазвичай використовується всередині підприємства. На відміну від тунелів 6to4, тунелі ISATAP не працюють, якщо IPv4 NAT використовується між кінцевими точками тунелю.

■ Teredo tunneling: цей метод дозволяє хостам з двома стеками створювати тунель до іншого хосту, причому сам хост одночасно створює пакет IPv6 і інкапсулює пакет всередині заголовка IPv4.

На Рисунку 1.9 показана основна ідея Teredo tunnel[15].

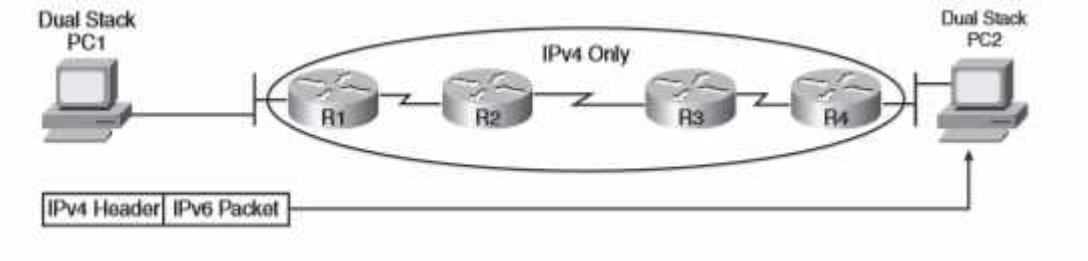

Рисунок 1.9 - приклад інкапсуляції для Teredo Host-Host Tunnel[15].

Translation

Address Family Translation (AFT) - це процес перекладу адрес з однієї родини адрес в іншу. Під час фази прийняття AFT в основному використовується

для перекладу між хостами IPv6 та вмістом IPv4. AFT може бути без стану, коли зарезервовані частини адресного простору IPv6 автоматично відображаються в IPv4, або вони можуть бути стаціонарними, з адресами з налаштованого діапазону, що використовується для відображення пакетів між сімействами адрес.

Майже всі розгортання підприємств IPv6 використовують внутрішній стек. Подвійний стек пропонує безперешкодний спосіб дізнатися про та отримати досвід роботи з новою "адресною сім'єю", яка є важливою частиною успішного управління переходом[9].

Translating Between IPv4 and IPv6 with NAT-PT

Обидва класи функцій переходу IPv6, подвійний стек і тунелі, покладаються на кінцеві хости, щоб підтримувати IPv6, якщо не обидва протоколи IPv4 і IPv6. Однак у деяких випадках хост, який працює лише на IPv4, повинен взаємодіяти з хостом, який працює лише на IPv6. У цьому випадку необхідно використовувати третій клас перехідних функцій: інструмент, який переводить заголовки пакета IPv6 так, щоб вони виглядали як пакет IPv4, і навпаки.

У маршрутизаторах Cisco для виконання трансляції можна використовувати перетворення мережевих адрес-протокол Translation (NAT-PT), визначене в RFC 2766. Для виконання своєї роботи маршрутизатор, налаштований за допомогою NAT-PT, повинен знати, який IPv6-адреса переводити на який IPv4-адресу і навпаки, таку ж інформацію зберігати в традиційній таблиці перекладу NAT. І як традиційний NAT, NAT-PT дозволяє статичне визначення, динамічне NAT і динамічне PAT, які можуть бути використані для збереження IPv4-адрес.

Таблиця 1.2 узагальнює параметри переходу для IPv6 для більш легкого використання і вивчення.

Назва	Тип	Опис
Dual stack	…	Підтримує обидва протоколи та посилає IPv4 хостам IPv4 та IPv6 хостам IPv6
Tunnel	MCT	Тунель налаштовано вручну; посилає IPv6 через мережу IPv4, як правило, між маршрутизаторами
Tunnel	6to4	Кінцеві точки тунелю динамічно виявляються; посилає IPv6 через мережу IPv4, як правило, між маршрутизаторами
Tunnel	ISATAP	Кінцеві точки тунелю динамічно виявляються; посилає IPv6 через мережу IPv4 між маршрутизаторами; не підтримує IPv4 NAT
Tunnel	Teredo	Зазвичай використовується хостами хост створює пакет IPv6 і інкапсулює в IPv4
NAT-PT	…	Маршрутизатор перекладає між IPv4 та IPv6; дозволяє хостам IPv4 спілкуватися з хостами IPv6

Таблиця 1.2 - Короткий опис варіантів переходу на IPv6. [12]

1.3 Постановка задачі

Для досягнення мети наукової роботи необхідно розробити графічний інтерфейс налаштування протоколу DHCP в мережах з підтримкою IPv6, який би мав такі характеристики:

- По перше, зручність, яка буде реалізована у вигляді графічного інтерфейсу, який буде нагадувати інтерфейс із Cisco Packet Tracer.
- По друге, достатній функціонал графічного інтерфейсу, до цього відноситься спосіб виведення сгенерованого коду у якості кнопки для пришвидшення і автоматизації процесу.

Такий інструмент став би у пригоді не тільки досвідченим користувачам, а і початківцям, не вимагаючи від них знання команд для налаштування протоколу DHCP в мережах з підтримкою IPv6.

Постановка задачі:

1. Конфігурація мережі в Cisco Packet Tracer.
2. Розробка графічного інтерфейсу для налаштування протоколу DHCP в мережах з підтримкою IPv6.
3. Тестування графічного інтерфейсу в Cisco Packet Tracer.

2 МОДЕЛЮВАННЯ ПРОТОКОЛУ DHCP В МЕРЕЖАХ З ПЫДТРИМКОЮ IPV6 ЗА ДОПОМОГОЮ ІНСТРУМЕНТА CISCO

2.1 Конфігурація мережі за допомогою інструмента CISCO

Одним із інструментів для розв'язку задач моделювання телекомунікаційних систем на ринку програмного забезпечення є CISCO Packet Tracer.

CISCO Packet Tracer - багатофункціональний емулятор, який дозволяє користувачам створювати віртуальні мережі, проектувати мережі, експериментувати з мережами та усувати несправності у мережах. Студенти та інструктори використовують Packet Tracer для вивчення складних технічних принципів і проектів мережних систем у безпечному віртуальному середовищі.

В Cisco Packet Tracer налаштування протоколу DHCP та DHCP server відбувається таким чином:

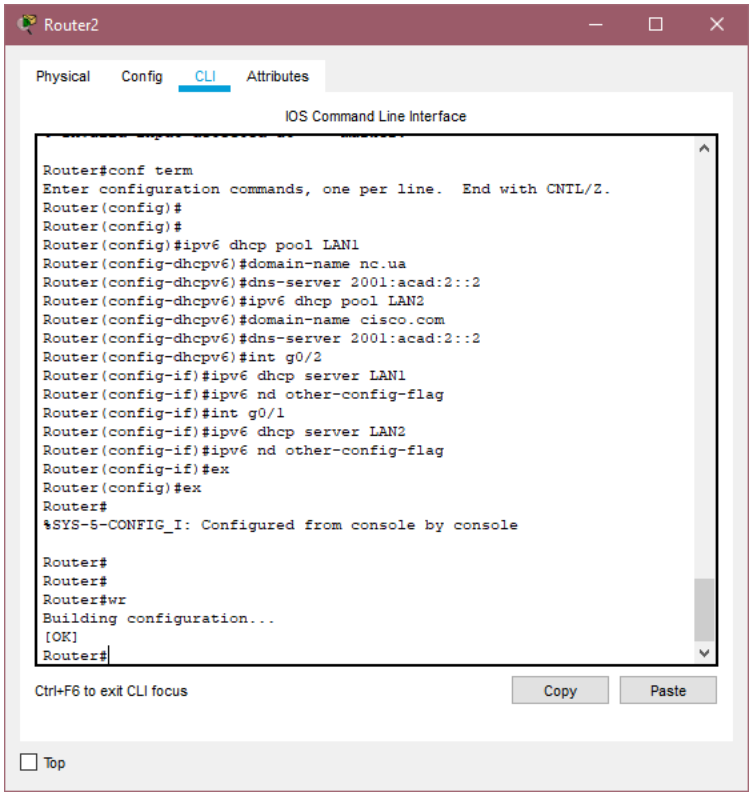

Рисунок 2.1 – налаштування протоколу DHCP та DHCP server

Комманды на рисунку 2.1 для налаштування протоколу DHCP та DHCP server:

	Команда	Призначення
Крок 1	**enable**	Вмикає привілейований режим EXEC.
Крок 2	**configure terminal**	Входить у режим глобальної конфігурації.
Крок 3	**ipv6 dhcp pool** *poolname*	Конфігурує пул інформації конфігурації DHCP для IPv6 (DHCPv6) та переходить у режим конфігурації пулу DHCPv6.
Крок 4	**dns-server** *ipv6-address*	Вказує сервери DNS IPv6, доступні клієнту DHCPv6.
Крок 5	**domain-name** *domain*	Налаштування доменного імені для клієнта DHCPv6.
Крок 6	**exit**	Виходить із режиму конфігурації пулу DHCPv6 та повертає пристрій у режим глобальної конфігурації.
Крок 7	**interface** *type number*	Вказує тип та номер інтерфейсу та переводить пристрій у режим конфігурації інтерфейсу.
Крок 8	**ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**]	Вмикає DHCPv6 в інтерфейсі.
Крок 9	**ipv6 nd other-config flag**	Встановлює прапор "інша конфігурація стану" в рекламних роутерах IPv6 (RA).
Крок 10	**wr**	Команда wr зберігає налаштування для роутера.

Таблиця 2.1 - налаштування протоколу DHCP та DHCP server

А налаштування протоколу IPv6 на роутерах має такий вигляд:

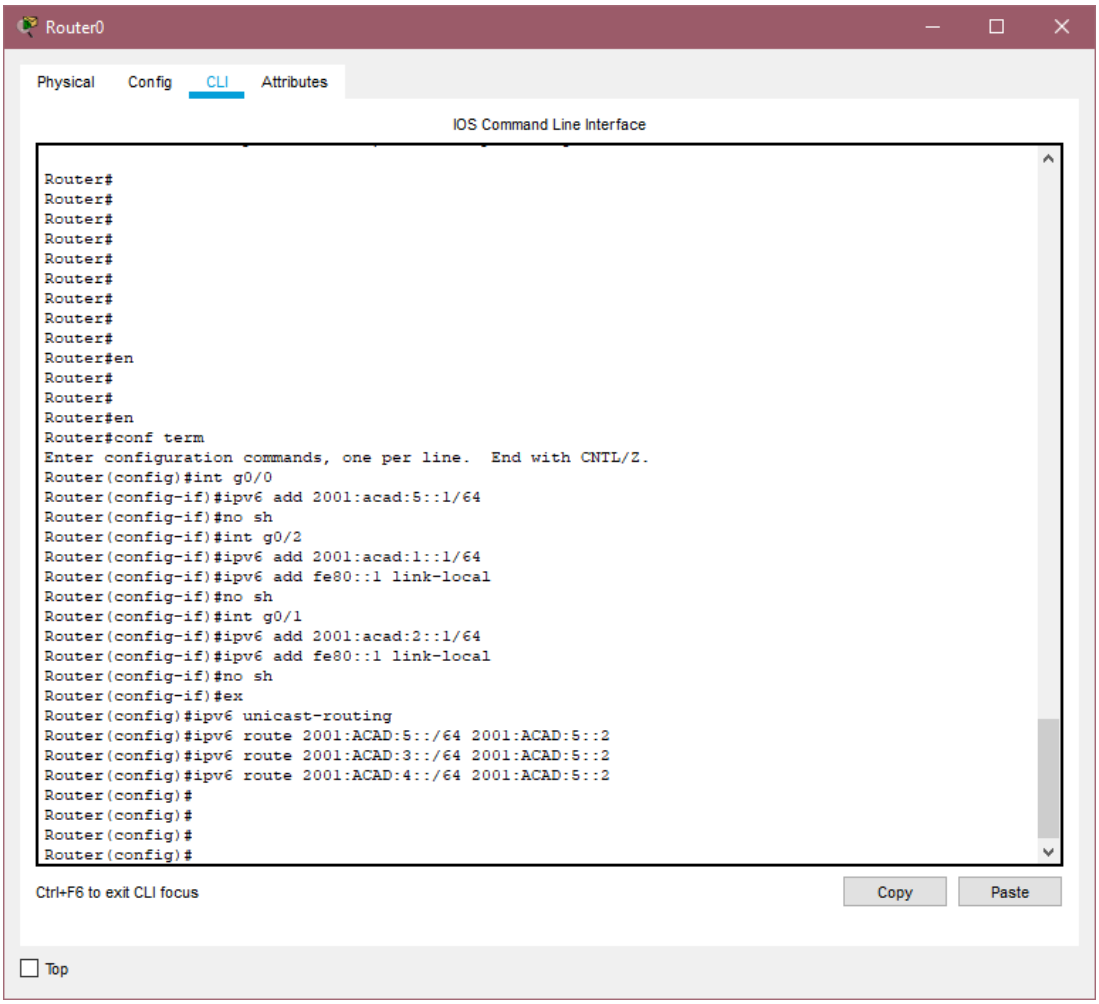

Рисунок 2.2 – налаштування протоколу протоколу IPv6 на роутерах

Комманды на рисунку 2.1 для налаштування протоколу IPv6 на роутерах:

	Команда	**Призначення**
Крок 1	**enable**	Вмикає привілейований режим EXEC.
Крок 2	**configure terminal**	Входить у режим глобальної конфігурації.
Крок 3	**interface** *type number*	Вказує тип та номер інтерфейсу та переводить пристрій у режим конфігурації інтерфейсу.
Крок 4	**ipv6 address ipv6-prefix/prefix-length**	Налаштовує IPv6 адресу інтерфейсу
Крок 5	**ipv6 address ipv6-prefix/prefix-length link-local**	Автоматично налаштовує локальну адресу IPv6 посилання в інтерфейсі, одночасно включаючи інтерфейс для обробки IPv6.
Крок 6	**exit**	Виходить із режиму конфігурації пулу DHCPv6 та повертає пристрій у режим глобальної конфігурації.
Крок 7	**ipv6 unicast-routing**	Вмикає переадресацію одноадресних дейтаграм IPv6.
Крок 8	**ipv6 route ipv6-prefix / prefix-length ipv6-address**	Статичний маршрут IPv6 за замовчуванням налаштовується на послідовний інтерфейс.

Таблиця 2.2 - налаштування протоколу IPv6 на роутерах

2.2 Застосування мови JavaScript для написання веб-додатків

Графічний інтерфейс налаштування протоколу DHCP в мережах з підтримкою IPv6 розроблявся за допомогою мови JavaScript.

JavaScript – динамічна, об`єктно-орієнтована та гнучка мова програмування, яка є найпопулярнішою на даний момент у веб-розробці.

В сучасних браузерах є інтерпретатор цієї мови, тому оброблятися JS буде на стороні клієнта.

Написані скрипти на JS можна використовувати для додавання елементів для керування веб додатком, зчитування введених даних з форм та їх обробки, можна керувати вікном браузера, в тому числі і виводити на екран вікно з повідомленням під час якого б призупинялось виконання скрипта. Скрипти можуть виконуватись відразу при завантаженні сторінки, або при певних діях користувача.

Js цілком задовольняє потребам для написання такого графічного інтерфейсу налаштування протоколу DHCP в мережах з підтримкою IPv6.

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ГРАФІЧНИОГО ІНТЕРФЕЙСУ НАЛАШТУВАННЯ ПРОТОКОЛУ DHCP В МЕРЕЖАХ З ПІДТРИМКОЮ IPV6

3.1 Розробка графічного інтерфейсу налаштування протоколу DHCP в мережах з підтримкою IPv6

Конфігурування мережі Ethernet проводилось в емуляторі Cisco Packet Tracer. Роутери були налаштовані за протоколом DHCP в мережах з підтримкою IPv6. Було виявлено, що Cisco Packet Tracer має не тільки консольний інтерфейс для налаштування динамічної маршрутизації. Введення команд вручну до консолі це дуже довготривалий та складний шлях, саме тому був реалізований проект по розробці графічного інтерфейсу налаштування протоколу DHCP в мережах з підтримкою IPv6, яка б збільшила зручніть для користувача та зекономила його час.

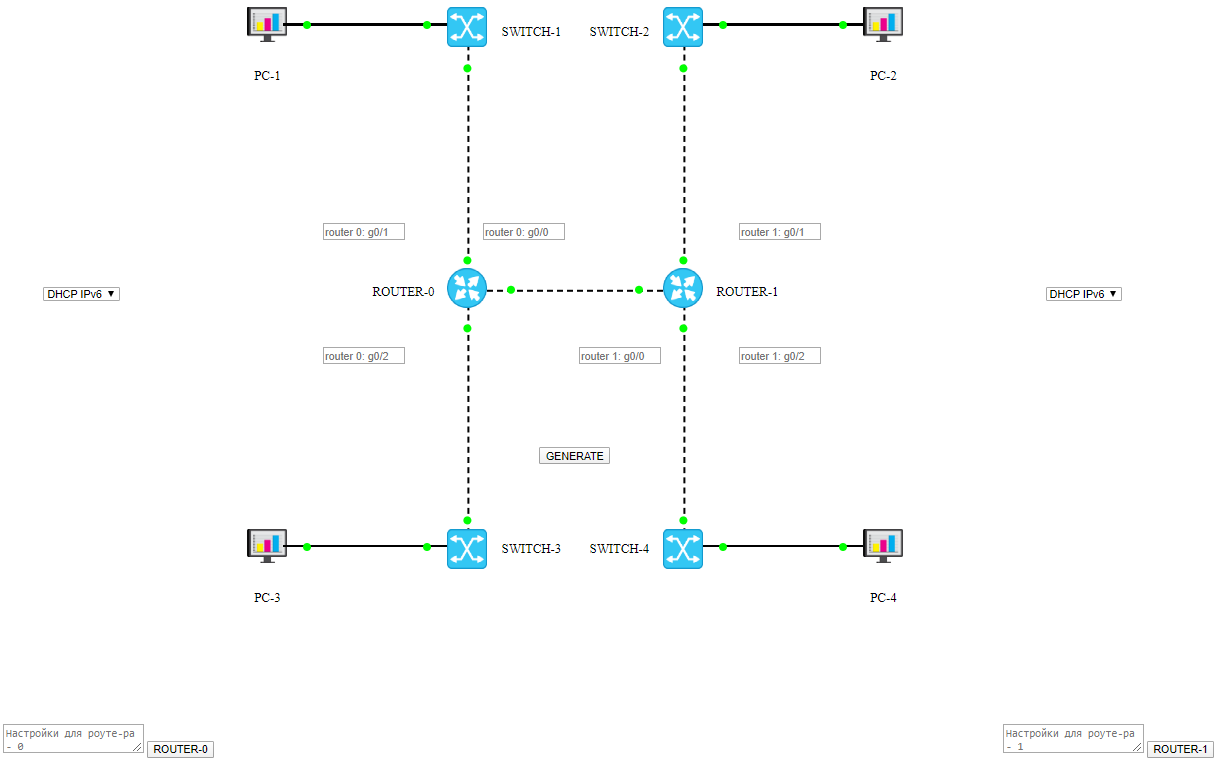

Рисунок 3.1 – Інтерфейс графічного інтерфейсу налаштування протоколу DHCP в мережах з підтримкою IPv6

Перейшовши на сторінку, користувач побаче схему мережі Ethernet, три кнопки «Generate» та «Copy» форми для кінцевих налаштувань для роутерів.

Користувач повинен ввести до всіх полів дані та натиснути на кнопку «Generate». Якщо IP введено вірно, вспливаючого вікна не буде і згенеруется код для налаштування динамічної маршрутизації

Згенерований код для усіх роутерів зображено на рисунку 3.2 .

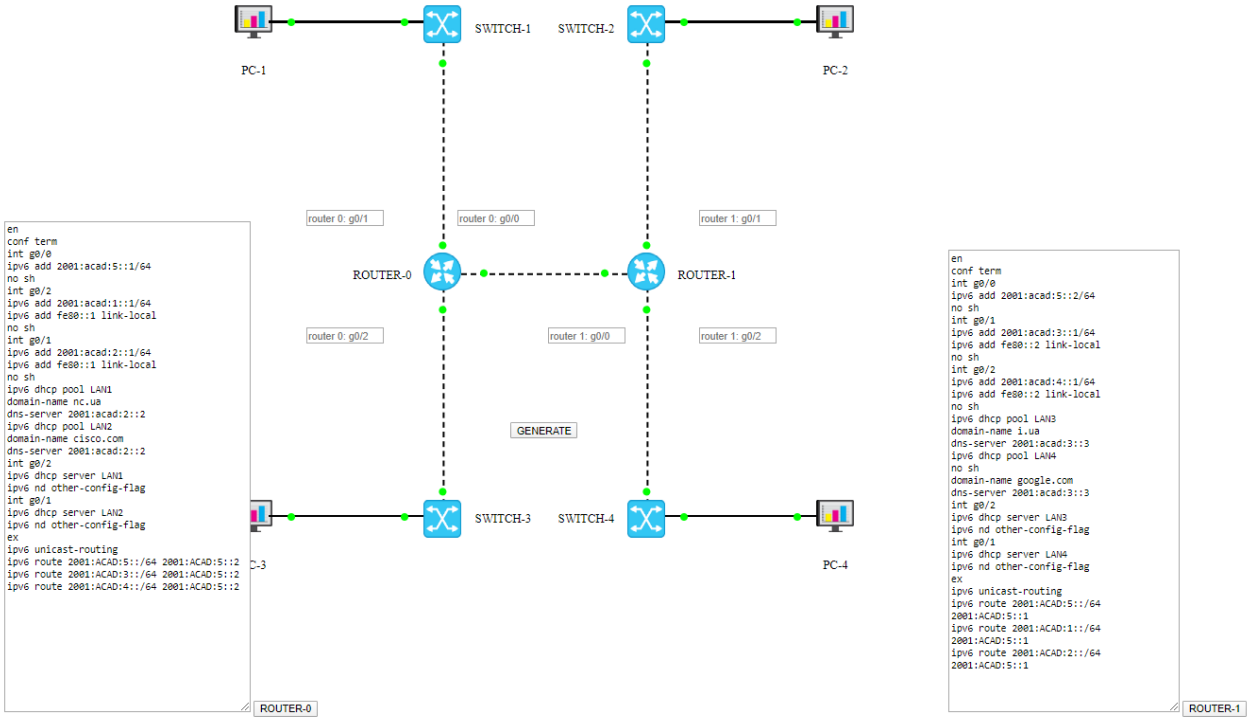

Рисунок 3.2 – Згенерований код за протоколом DHCP в мережах з підтримкою IPv6

Веб-додаток забезпечений валідацією для IP полів. Ввівши невірний формат та клацнувши на кнопку «Generate», браузер покаже віконце з текстом «Введен некорректний адрес IP», код для вибраного протоколу не згенерується.

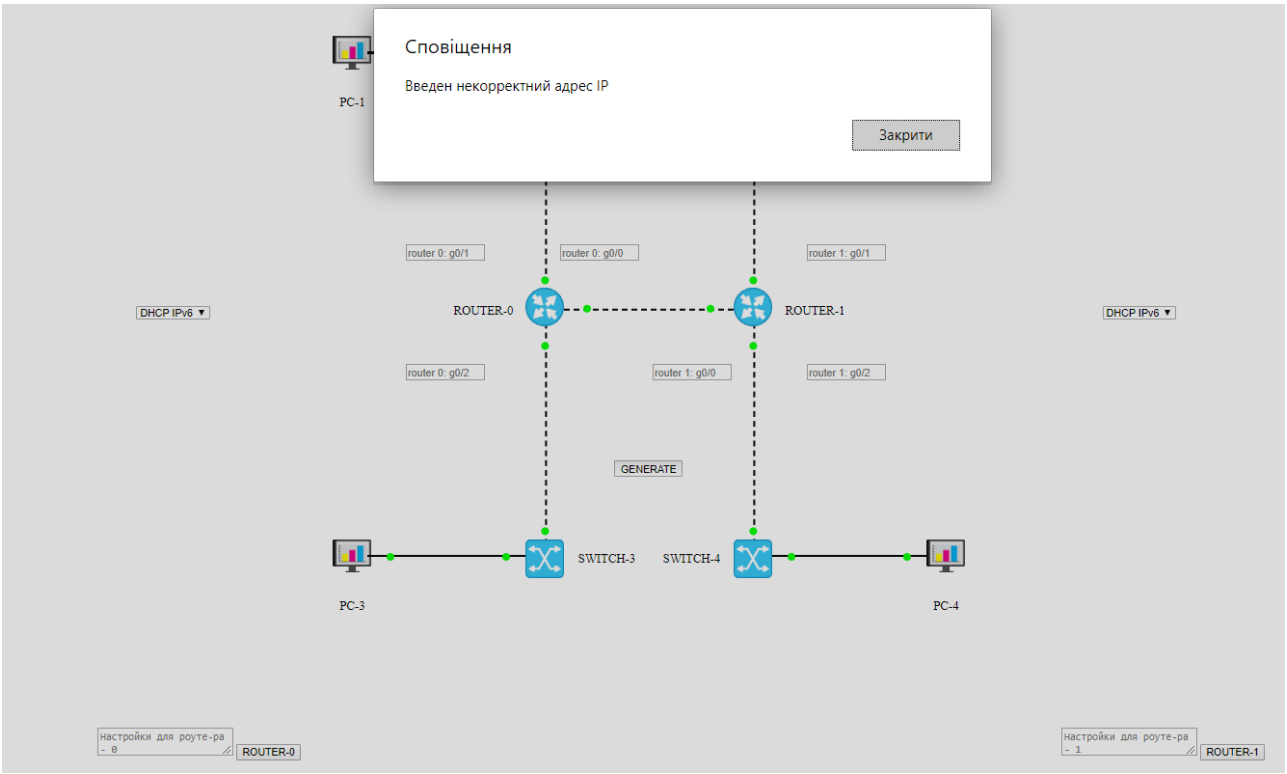

Рисунок 3.3 – Перевірка формату IP адреси

Для налаштування мережі користувачу потрібно натиснути кнопку з назвою роутера, яка копіює код налаштувань до буферу. Користувач повинен вставити готові налаштування до консолі відповідного роутера.

3.2 Тестування графічного інтерфейсу в Cisco Packet Tracer

Щоб перевірити графічний інтерфейс налаштування протоколу DHCP в мережах з підтримкою IPv6 на помилки потрібно протестувати її в Cisco Packet Tracer.

На основі згенерованих команд для роутерів перевіримо розроблену програму:

Спочатку сгенеруємо код для налаштування протоколу DHCP в мережах з підтримкою IPv6 та скопіюємо його.

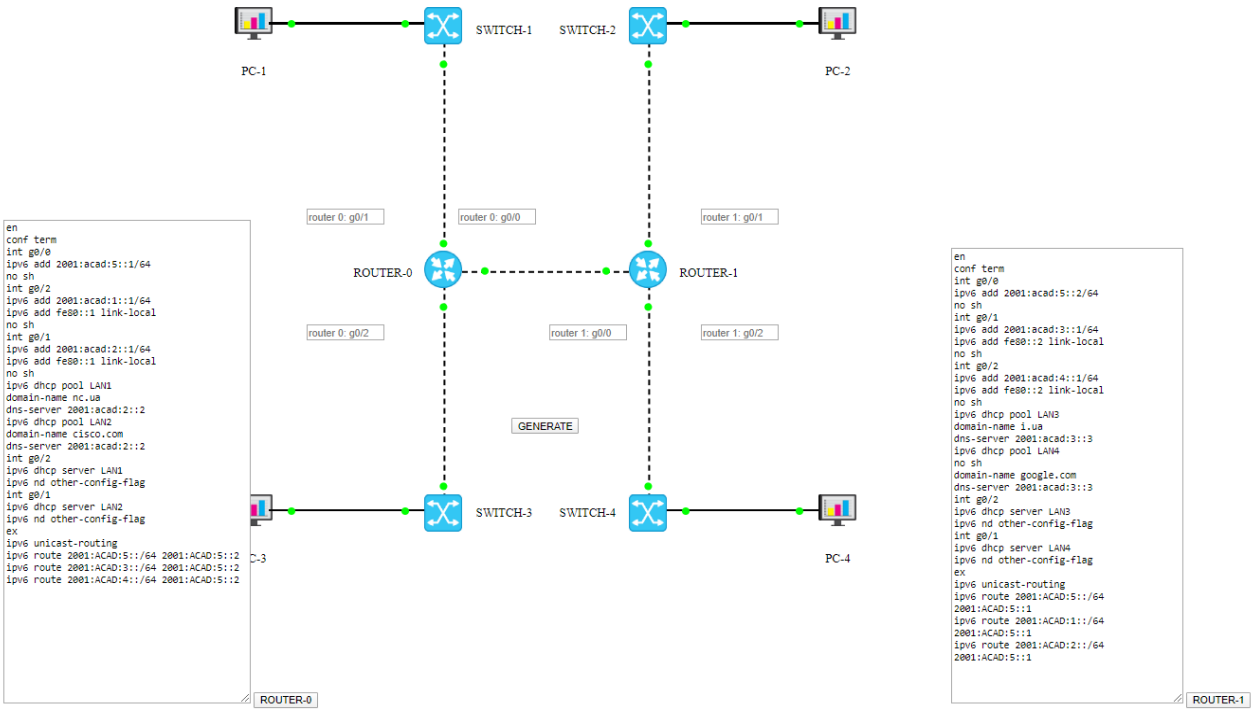

Рисунок 3.4 – Згенерований код за протоколом DHCP в мережах з підтримкою IPv6

Вставляємо налаштування у консольне вікно роутера в симуляторі Cisco Packet Tracer (рис 3.5).

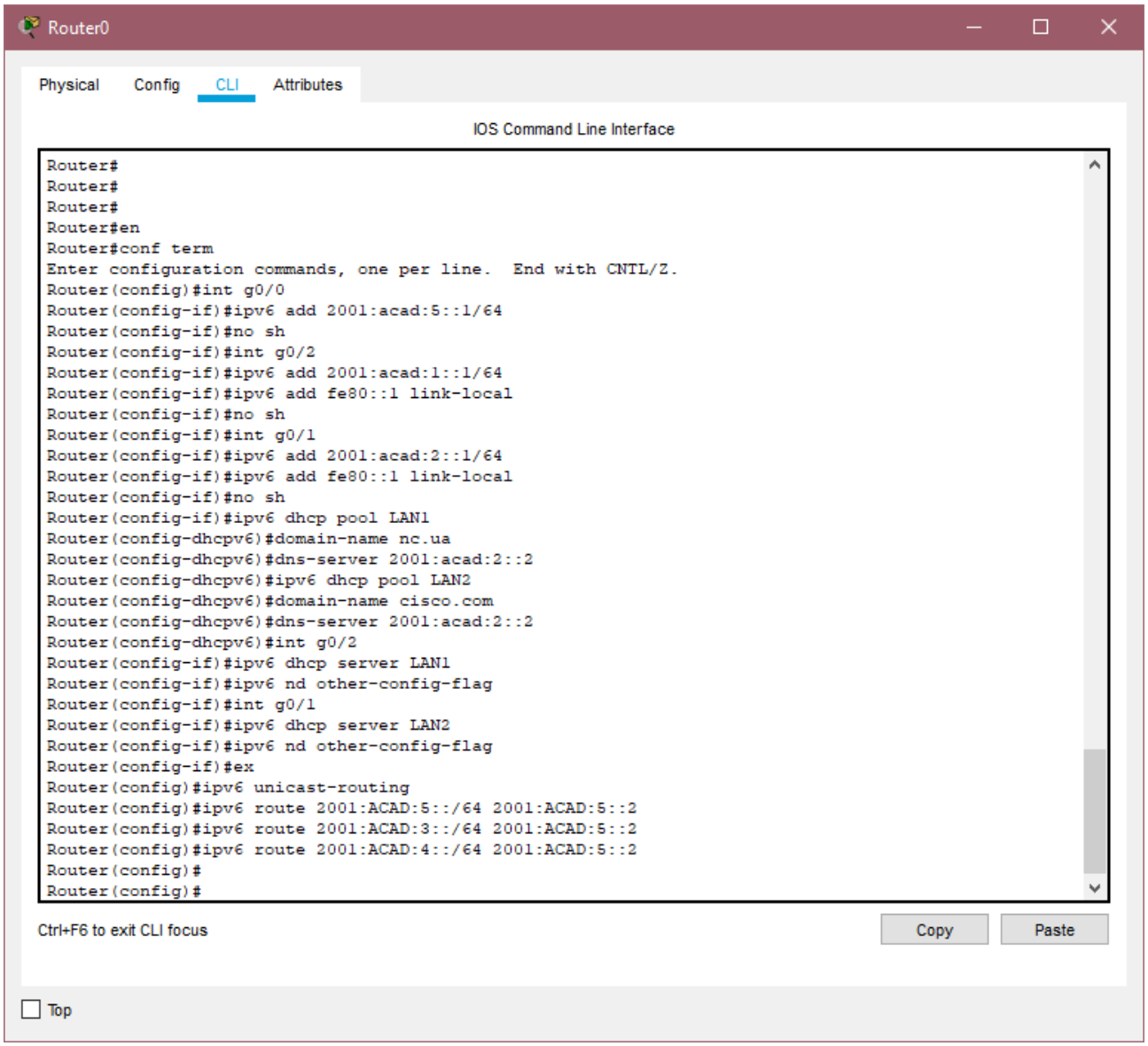

Рисунок 3.5 – Вікно налаштувань роутера – 0

Таким способом налаштуємо і інші роутери.

Щоб переконатисьв тому, що налаштування застосувались треба виконати команду «show run».

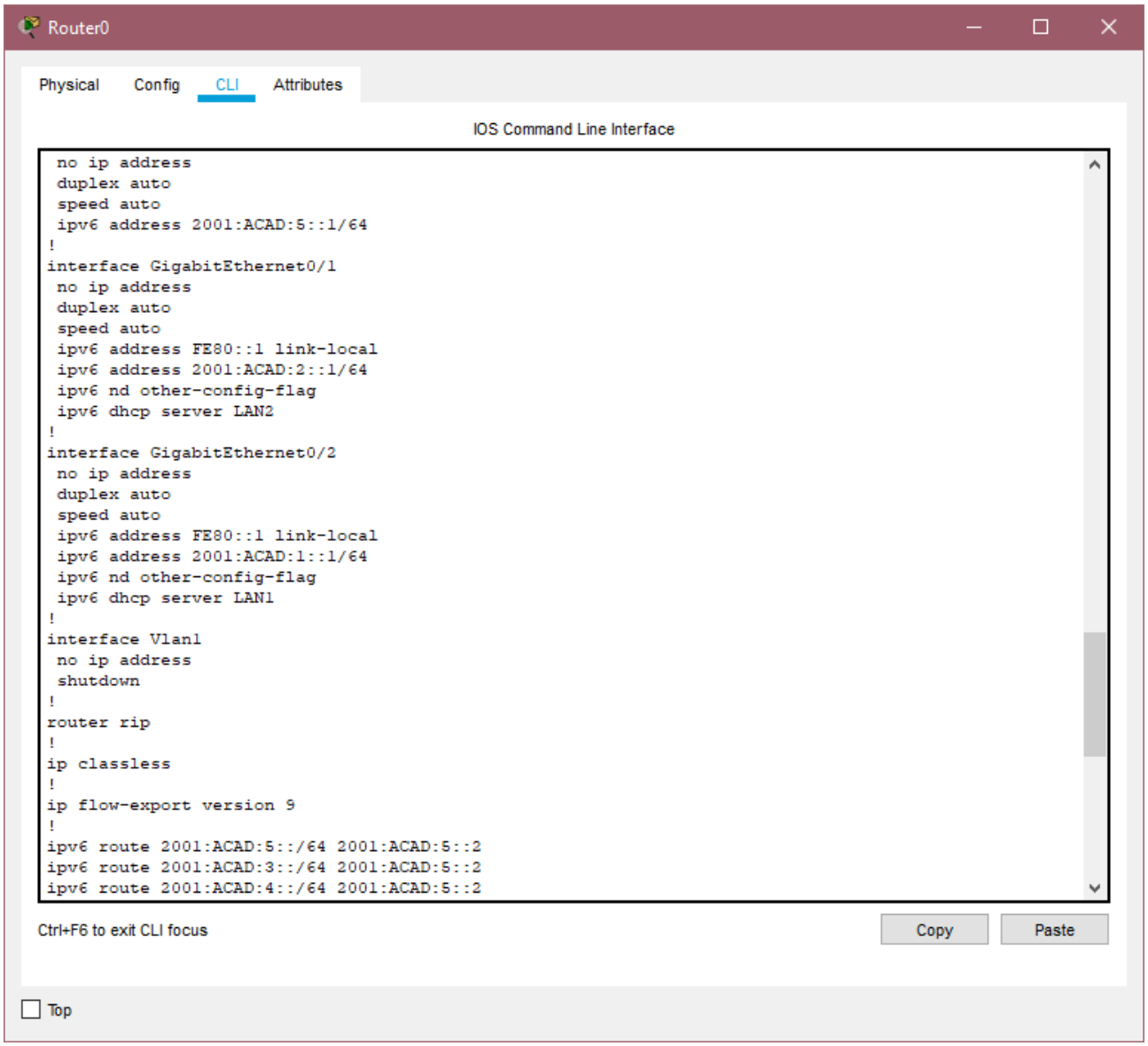

Рисунок 3.6 – Результат команди «show run»

Шляхом перевірки за допомогою команди «show run» запевнюємось в тому, шо налаштування сгенерувались вірно та застосувались для всіх роутерів.

Протестуємо мережу та впевнимось що вона робоча.

Спочатку за допогою команди «ping», а потім за допогою передачі симуляційного пакету.

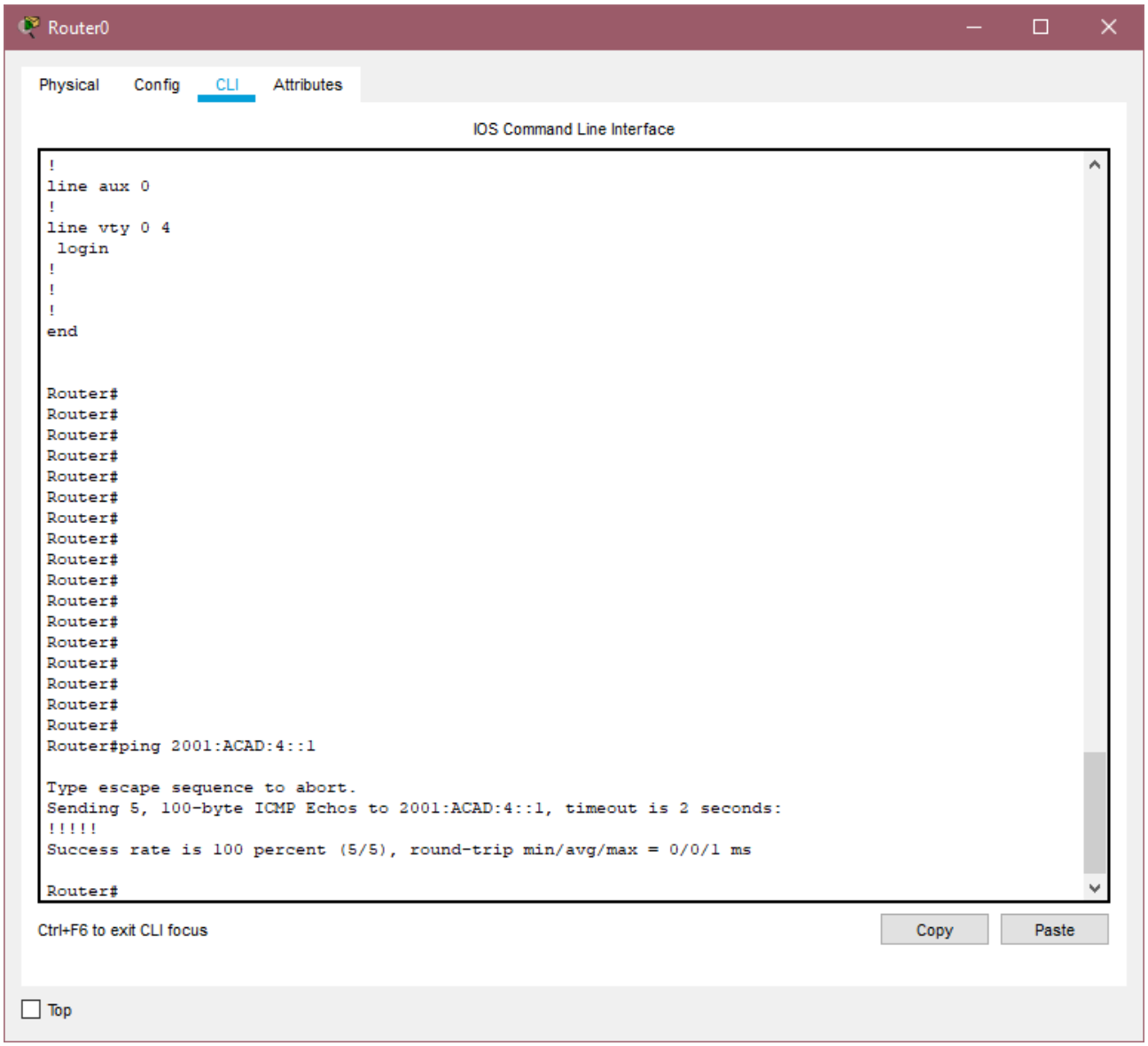

Рисунок 3.7 – Перевірка роботоздатності мережі за допогомю команди «ping»

Перевірка роботоздатності мережі пройшла успішно (рис 3.7).

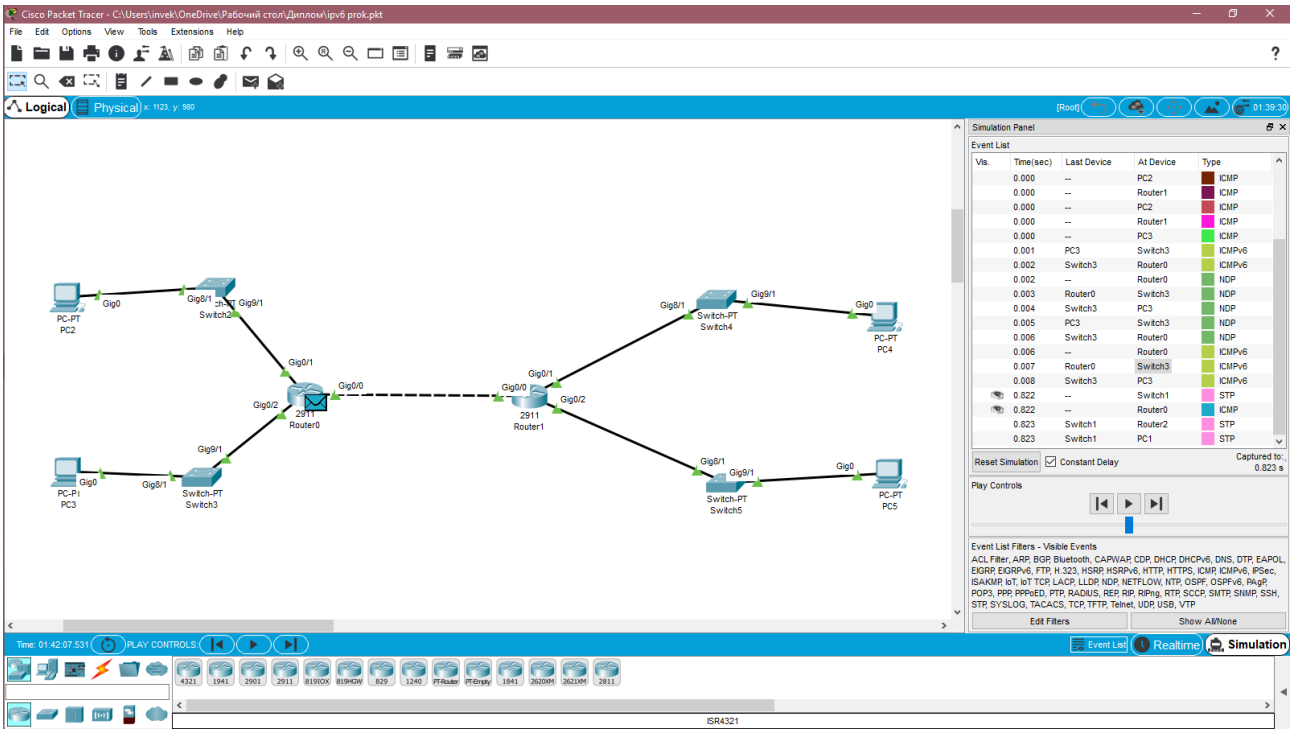

Рисунок 3.8 – Перевірка роботоздатності мережі за допогомю симуляційного пакету

Відправимо симуляційни пакет з роутера 0 до роутера 1. Перевірка роботоздатності мережі пройшла успішно (рис 3.8).

На налаштування було витрачено 5 хвилин, що значно коротше ніж якщо налаштовувати вручу.

Протестований графічний інтерфейс спрощує та робить налаштування протоколу DHCP в мережах IPv6 значно зручнішим.

ВИСНОВКИ

У ході виконання кваліфікаційної магістерської роботи було проведено критичний аналіз літературних джерел, виходячи з результатів якого можна стверджувати що, основною проблемою сучасних мережевих емуляторів, таких як Cisco Packet Tracer є довготривалість та складність процесу налаштування динамічної маршрутизації для мереж.

Складність насамперед проявляется в обов`язковому знанні команд налаштування протоколу DHCP в мережах з підтримкою IPv6 та їх послідовності, що ускладнює новим користувачам отримати комфортний продукт під час роботи.

Постійні відволікання на налаштування окремих роутерів та мереж, введення кожен раз великої кількості команд змушують досвідчених користувачів втрачати дорогоцінні хвилини свого життя, затримує їх розвиток, а, інколи, забирають, навіть, кошти. Тому, в рамках наукової роботи, простим та очевидним способом пришвидшення даного процесу є розробка графічного інтерфейсу.

Створений графічний інтерфейс в рамках наукової роботи має інтуїтивний та зрозумілий інтерфейс. Для отримання готової робочої мережі в Cisco Packet Tracer потрібно лише задати ip-адреси інтерфейсів роутера та отримати налаштування за допомогою універсальної кнопки, скопіювати виведений результат, вписати його в консоль керування роутерів та радіти життю.

СПИСОК ЛИТЕРАТУРЫ

1. Пайпер Б. - Администрирование сетей Cisco: освоение за месяц / пер. с анг. М. А. Райтмана. – М.: ДМК Пресс, 2018. – 316 с.: ил.
2. Лэммл Т. - Cisco Certified Network Associate. Учебное руководство – 2002
3. IP Addressing DHCP Configuration Guide, Cisco IOS Release 12.4 – 2011
4. Cisco IOS Cookbook™, Second Edition by Kevin Dooley and Ian J. Brown - 2016
5. Schudel G. - Router Security Strategies – 2018
6. Kocharinas N. - CCIE Routing And Switchng v5.0 vol 2 – 2015
7. Santos O. - CCNA Security 210-260 - 2015.
8. Амато, Вито. - Основы организации сетей Cisco, том 2 - 2018
9. Kocharinas N. - CCIE Souting and Switchng v5.0 vol 1 – 2015
10. Дж. Бони - Руководство по Cisco IOS для проффесионалов, 2018
11. Jain V., Edgeworth B., Furr R. - Troubleshooting Cisco Nexus Switches and NX-OS – 2017
12. CCENT/ CCNA ICND1 100-105 Official Cert Guide, 2016
13. W. Odom - CCENT CCNA ICND1 640-822 Official Cert Guide, 3rd Edition – 2015
14. Cisco Networking Academy Program CCNA® 1 and 2 Companion Guide Revised Third Edition – 2018
15. W. Odom - CCNA ICND2 640-816 Official Cert Guide, 3rd Edition – 2015

ДОДАТОК

Додаток А

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<!-- НАЧАЛО КОНСТРУКТОРА -->
<!--
.dot-red - красныче точки на красной линии
.dot-green - зеленые точки на красной линии
.dot-succes - зеленые точки
.dot-error - красные точки
-->
<!-- контейнер с сервером -->
<div class="server-wrapper" style="top:50px;left:0; display:none;">
<div class="container-server">
<img src="img/server.png" alt="">
<p class="server-name">DNS</p>
</div>
</div>

<!-- контейнер проводов зеленые-->
<div class="line-wrapper" style="top:20px;left:0px; display:none;">
<div class="line-container" style="width: 300px; transform: rotate(0deg);">
<div class="line">
<div class="dot-succes dot-2"></div>
<div class="dot-succes dot-1"></div>
</div>
</div>
</div>
<!-- контейнер проводов красные -->
<div class="line-wrapper" style="top:40px;left:0; display:none;">
<div class="line-container" style="width: 300px; transform: rotate(0deg);">
<div class="line">
<div class="dot-error dot-2"></div>
<div class="dot-error dot-1"></div>
```

```
</div>
</div>
</div>
<!-- контейнер проводов зеленые штрих-->
<div class="line-wrapper-shtr" style="top:20px;left:400px; display:none;">
<div class="line-container-shtr" style="width: 300px; transform: rotate(0deg);">
<div class="line">
<div class="dot-succes dot-2"></div>
<div class="dot-succes dot-1"></div>
</div>
</div>
</div>
<!-- контейнер проводов красные штрих-->
<div class="line-wrapper-shtr" style="top:40px;left:400; display:none;">
<div class="line-container-shtr" style="width: 300px; transform: rotate(0deg);">
<div class="line">
<div class="dot-error dot-2"></div>
<div class="dot-error dot-1"></div>
</div>
</div>
</div>
<!-- красная линия -->
<div class="red-line-wrapper" style="top:100px;left:100px; display:none;">
<div class="container-line">
<div class="line-1">
<!-- <div class="dot-red"></div> -->
<div class="dot-green"></div>
</div>
<div class="line-2">
</div>
<div class="line-3">
<!-- <div class="dot-red"></div> -->
<div class="dot-green"></div>
</div>
</div>
</div>
<!-- форма для отправки данных -->
<input maxlength="15" size="11" style="display:none;"/>
<!-- ФОРМА для принятия данных-->
<div class="form-wrapper" style="display:none;">
<form action="#">
```

```
<textarea id="copyTextArea999" name="form" placeholder="Настройки для роу-тера" readonly></textarea>
<button id="other999" action="submit" >Копировать</button>
</form>
</div>

<!-- КОНЕЦ КОНСТРКТОРА -->

<!-- 1 блок -->

<!-- контейнер с свитчем -->
<div class="pc-wrapper" style="top:50px;left:530px;">
<div class="container-pc">
<img src="img/pc.png" alt="">
<p class="server-name"></br></br> PC-1</p>
</div>
</div>
<!-- контейнер проводов зеленые-->
<div class="line-wrapper" style="top:70px;left: 600px;">
<div class="line-container" style="width: 300px; transform: rotate(0deg);">
<div class="line">
<div class="dot-succes dot-2"></div>
<div class="dot-succes dot-1"></div>
</div>
</div>
</div>
<!-- контейнер с роутером -->
<div class="switch-wrapper" style="top:50px;left:780px; z-index: 10;">
<div class="container-switch">
<img src="img/switch.png" alt="">
</div>
</div>
<div class="switch-wrapper" style="top:75px;left:860px; z-index: 10;">
<div class="container-switch">
<p class="server-name">SWITCH-1</p>
</div>
</div>

<!-- контейнер с роутером -->
<div class="switch-wrapper" style="top:50px;left:1050px; z-index: 10;">
<div class="container-switch">
<img src="img/switch.png" alt="">
</div>
```

```
</div>
<div class="switch-wrapper" style="top:75px;left:970px; z-index: 10;">
<div class="container-switch">
<p class="server-name">SWITCH-2</p>
</div>
</div>
<!-- контейнер проводов зеленые-->
<div class="line-wrapper" style="top:70px;left: 1120px;">
<div class="line-container" style="width: 300px; transform: rotate(0deg);">
<div class="line">
<div class="dot-succes dot-2"></div>
<div class="dot-succes dot-1"></div>
</div>
</div>
</div>
<!-- контейнер с свитчем -->
<div class="pc-wrapper" style="top:50px;left:1300px;">
<div class="container-pc">
<img src="img/pc.png" alt="">
<p class="server-name"></br></br> PC-2</p>
</div>
</div>

<!-- 2 блок -->

<!-- контейнер проводов зеленые штрих-->
<div class="line-wrapper-shtr" style="top:243px;left:680px;">
<div class="line-container-shtr" style="width: 300px; transform: rotate(90deg);">
<div class="line">
<div class="dot-succes dot-2"></div>
<div class="dot-succes dot-1"></div>
</div>
</div>
</div>
<!-- контейнер проводов зеленые штрих-->
<div class="line-wrapper-shtr" style="top:243px;left:950px;">
<div class="line-container-shtr" style="width: 300px; transform: rotate(90deg);">
<div class="line">
<div class="dot-succes dot-2"></div>
<div class="dot-succes dot-1"></div>
</div>
```

```
</div>
</div>

<!-- 3 блок -->
<select class="select" id="1select" style="top:400px;left:300px;">
<option> DHCP IPv6 </option>
</select>
<!-- контейнер с роутером -->
<div class="router-wrapper" style="top:376px;left:780px; z-index: 10;">
<div class="container-router">
<img src="img/router.png" alt="">
</div>
</div>
<div class="router-wrapper" style="top:400px;left:700px;">
<div class="container-router">
<p class="server-name">ROUTER-0</p>
</div>
</div>

<!-- контейнер с роутером -->
<div class="router-wrapper" style="top:376px;left:1050px; z-index: 10;" >
<div class="container-router">
<img src="img/router.png" alt="">
</div>
</div>
<div class="router-wrapper" style="top:400px;left:1130px;">
<div class="container-router">
<p class="server-name">ROUTER-1</p>
</div>
</div>

<!-- контейнер проводов зеленые штрих-->
<div class="line-wrapper-shtr" style="top:400px; left:855px;">
<div class="line-container-shtr" style="width: 220px; transform: rotate(180deg);">
<div class="line">
<div class="dot-succes dot-2"></div>
<div class="dot-succes dot-1"></div>
</div>
</div>
</div>
<select class="select" id="2select" style="top:400px;left:1553px;">
<option> DHCP IPv6 </option>
```

```
</select>

<!-- 4 блок -->

<!-- контейнер проводов зеленые штрих-->
<div class="line-wrapper-shtr" style="top:568px;left:680px;">
<div class="line-container-shtr" style="width: 300px; transform: rotate(90deg);">
<div class="line">
<div class="dot-succes dot-2"></div>
<div class="dot-succes dot-1"></div>
</div>
</div>
</div>
<!-- контейнер проводов зеленые штрих-->
<div class="line-wrapper-shtr" style="top:568px;left:950px;">
<div class="line-container-shtr" style="width: 300px; transform: rotate(90deg);">
<div class="line">
<div class="dot-succes dot-2"></div>
<div class="dot-succes dot-1"></div>
</div>
</div>
</div>

<!-- 5 блок -->

<!-- контейнер с свитчем -->
<div class="pc-wrapper" style="top:702px;left:530px;">
<div class="container-pc">
<img src="img/pc.png" alt="">
<p class="server-name"></br></br> PC-3</p>
</div>
</div>
<!-- контейнер проводов зеленые-->
<div class="line-wrapper" style="top:722px;left: 600px;">
<div class="line-container" style="width: 300px; transform: rotate(0deg);">
<div class="line">
<div class="dot-succes dot-2"></div>
<div class="dot-succes dot-1"></div>
</div>
</div>
</div>
<!-- контейнер с switch -->
```

```
<div class="switch-wrapper" style="top:702px;left:780px;">
<div class="container-switch">
<img src="img/switch.png" alt="">
</div>
</div>
<div class="switch-wrapper" style="top:702px;left:860px;">
<div class="container-switch">
<p class="server-name"></br>SWITCH-3</p>
</div>
</div>

<!-- контейнер с роутером -->
<div class="switch-wrapper" style="top:702px;left:1050px; z-index: 10;">
<div class="container-switch">
<img src="img/switch.png" alt="">
</div>
</div>
<div class="switch-wrapper" style="top:702px;left:970px; z-index: 10;">
<div class="container-switch">
<p class="server-name"></br>SWITCH-4</p>
</div>
</div>

<!-- контейнер проводов зеленые-->
<div class="line-wrapper" style="top:722px;left: 1120px;">
<div class="line-container" style="width: 300px; transform: rotate(0deg);">
<div class="line">
<div class="dot-succes dot-2"></div>
<div class="dot-succes dot-1"></div>
</div>
</div>
</div>

<!-- контейнер с свитчем -->
<div class="pc-wrapper" style="top:702px;left:1300px;">
<div class="container-pc">
<img src="img/pc.png" alt="">
<p class="server-name"></br></br> PC-4</p>
</div>
</div>

<div class="form-wrapper" style="top: 600px; left: 920px;">
<button id="generate" action="submit" >GENERATE</button>
</div>
```

```
<!-- БЛОК ДЛЯ ФОРМ -->

<!-- ВВОД -->

<!-- Router 0 -->
<!-- форма для отправки данных -->
<input class="input-wrapper" id="input0.2" maxlength="15" size="11"
style="top:320px;left:850px" placeholder="router 0: se 0/2" />

<!-- форма для отправки данных -->
<input class="input-wrapper" id="input0.1" maxlength="15" size="11"
style="top:320px;left:650px" placeholder="router 0: fa 0/1" />

<!-- форма для отправки данных -->
<input class="input-wrapper" id="input0.0" maxlength="15" size="11"
style="top:475px;left:650px" placeholder="router 0: fa 0/0" />


<!-- Router 1 -->
<!-- форма для отправки данных -->
<input class="input-wrapper" id="input1.2" maxlength="15" size="11"
style="top:475px;left:970px" placeholder="router 1: se 0/2" />

<!-- форма для отправки данных -->
<input class="input-wrapper" id="input1.1" maxlength="15" size="11"
style="top:320px;left:1170px" placeholder="router 1: fa 0/1" />

<!-- форма для отправки данных -->
<input class="input-wrapper" id="input1.0" maxlength="15" size="11"
style="top:475px;left:1170px" placeholder="router 1: fa 0/0" />


<!-- ВЫВОД -->
<!-- ФОРМА для принятия данных-->
<div class="form-wrapper" style="bottom: 10px; left: 250px;">
<form action="#">
<textarea id="copyTextArea0" name="form" placeholder="Настройки для роуте-ра - 0"
readonl></textarea>
<button id="other0" action="submit" >ROUTER-0</button>
</form>
</div>
<!-- ФОРМА для принятия данных-->
<div class="form-wrapper" style="bottom: 10px; left: 1500px;">
<form action="#">
<textarea id="copyTextArea1" name="form" placeholder="Настройки для роуте-ра - 1"
readonl></textarea>
<button id="other1" action="submit" >ROUTER-1</button>
```

```
</form>
</div>

<!-- STYLES-->
<style>
.select{
position: absolute;
}
.server-name{
margin-top: -3px;
z-index: 10;
}
.lineSuperRed {
position: absolute;
}
.input-wrapper{
position: absolute;
}
.form-wrapper{
position: absolute;
z-index: 10;
}
.red-line-wrapper {
position: absolute
width: 40%;
height: 10%;
}
.red-line-wrapper .line-1 {
display: inline-block;
height: 3px;
width: 300px;
margin-right: -46px;
margin-bottom: -5px;
background-color: red;
transform: rotate(5deg);
}
.red-line-wrapper .line-1 .dot-red {
position: absolute;
display: inline-block;
height: 10px;
width: 10px;
```

```
top: -3px;
left: 20px;
border-radius: 50%;
background-color: red;
}
.red-line-wrapper .line-3 .dot-red {
position: absolute;
display: inline-block;
height: 10px;
width: 10px;
top: -3px;
right: 20px;
border-radius: 50%;
background-color: red;
}
.red-line-wrapper .line-1 .dot-green {
position: absolute;
display: inline-block;
height: 10px;
width: 10px;
top: -3px;
left: 20px;
border-radius: 50%;
background-color: #00FF00;
}
.red-line-wrapper .line-3 .dot-green {
position: absolute;
display: inline-block;
height: 10px;
width: 10px;
top: -3px;
right: 20px;
border-radius: 50%;
background-color: #00FF00;
}
.red-line-wrapper .line-2 {
display: inline-block;
height: 3px;
width: 50px;
background-color: red;
transform: rotate(45deg);
```

```
}
.red-line-wrapper .line-3 {
display: inline-block;
height: 3px;
width: 300px;
margin-left: -47px;
margin-bottom: 4px;
background-color: red;
transform: rotate(5deg);
}
body {
position: relative;
}
.container-router img,
.container-switch img,
.container-server img,
.container-pc img {
width: 50px;
}
.container-router,
.container-switch,
.container-server,
.container-pc {
display: flex;
flex-direction: column;
align-items: center;
text-align: center;
width: 100px;
}
.line-wrapper,
.line-wrapper-shtr,
.server-wrapper,
.switch-wrapper,
.router-wrapper,
.pc-wrapper {
position: absolute;
}
.line-container .line {
position: relative;
display: block;
width: 70%;
```

```
height: 3px;
background-color: #000;
}
.line-container .line>.dot-error {
position: absolute;
display: inline-block;
height: 10px;
width: 10px;
top: -3px;
border-radius: 50%;
background-color: red;
}
.line-container .line>.dot-succes {
position: absolute;
display: inline-block;
height: 10px;
width: 10px;
top: -3px;
border-radius: 50%;
background-color: #00FF00;
}
.line-container .line>.dot-1 {
left: 25px;
}
.line-container .line>.dot-2 {
right: 25px;
}
/* Штрих */
.line-container-shtr .line {
position: relative;
display: block;
width: 100%;
height: 3px;
border-top: dashed 3px #000;
}
.line-container-shtr .line>.dot-error {
position: absolute;
display: inline-block;
height: 10px;
width: 10px;
top: -6px;
```

```
border-radius: 50%;
background-color: red;
}
.line-container-shtr .line>.dot-succes {
position: absolute;
display: inline-block;
height: 10px;
width: 10px;
top: -6px;
border-radius: 50%;
background-color: #00FF00;
}
.line-container-shtr .line>.dot-1 {
left: 25px;
}
.line-container-shtr .line>.dot-2 {
right: 25px;
}
</style>

<script src="jquery-latest.js"></script>
</body>
</html>
```