

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

# **КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА**

**на тему:**

**«Графічний інтерфейс інтерактивного налаштування  
роутерів CISCO з підтримкою протоколу IPv6»**

**Завідувач  
випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Великодний Д.В.**

**Студентка групи ІК.м-91**

**Вітер А.А.**

**СУМИ 2020**

Сумський державний університет  
(назва вузу)

Факультет ЕЛІП Кафедра Комп'ютерних наук  
Спеціальність 122 «Комп'ютерні науки»

Затверджую:  
зав. кафедри \_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Вітер Альбіни Анатоліївни  
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Графічний інтерфейс інтерактивного налаштування роутерів CISCO  
з підтримкою протоколу 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) Огляд існуючих рішень. 2) Налаштування мережі.  
3) Проектування та реалізація графічного інтерфейсу

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
	<i>Огляд існуючих рішень</i>		
	<i>Налаштування мережі</i>		
	<i>Проектування та реалізація графічного інтерфейсу</i>		
	<i>Оформлення кваліфікаційної магістерської роботи</i>		

Студент – дипломник \_\_\_\_\_

(підпис)

Керівник проекту \_\_\_\_\_

(підпис)

## РЕФЕРАТ

**Записка:** 47 стор., 13 рис., 1 додаток, 18 джерел.

**Об'єкт дослідження** — процес проектування та реалізації схем мережі на основі протоколу IPv6

**Мета роботи** — розробка графічного інтерфейсу .

**Методи дослідження** — методи налаштування мережі на основі протоколу IPv6, методи створення графічного інтерфейсу.

**Результати** — був створений графічний інтерфейс для інтерактивного налаштування роутерів CISCO з підтримкою протоколу IPv6. В ході роботи було проведено аналіз різних методів налаштувань мережі, у тому числі тих, що можуть підтримувати протоколи IPv6 та IPv4 одночасно. Також, розглянуто та проаналізовано інструменти створення графічного інтерфейсу.

IPV6, NAT64, МАРШРУТИЗАЦІЯ,  
ГРАФІЧНИЙ ІНТЕРФЕЙС, HTML, CSS, JAVA SCRIPT

## ЗМІСТ

ВСТУП .....	5
1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	6
1.1.Протокол IPv4.....	6
1.2.Протокол IPv6.....	8
1.3.Порівняння IPv6 та IPv4 .....	10
1.4.Способи роботи з протоколами .....	12
1.5.Огляд методів проектування та реалізації графічного інтерфейсу .....	15
1.6. Постановка задачі .....	16
2 НАЛАШТУВАННЯ МЕРЕЖІ .....	18
2.1. Налаштування вузлів на основі протоколу IPv6.....	18
2.2. Подвійний стек .....	18
2.3. Тунелювання.....	21
2.4. NAT64.....	23
3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ГРАФІЧНОГО ІНТЕРФЕЙСУ ..	26
3.1. Проектування.....	26
3.2. Створення дизайну інтерфейсу .....	28
3.3. Налаштування логіки інтерфейсу.....	29
3.4. Перевірка роботи інтерфейсу .....	30
ВИСНОВОК .....	32
СПИСОК ЛІТЕРАТУРИ .....	33
ДОДАТОК .....	35

## ВСТУП

На сьогоднішній день розвитку технологій присвячено особливу увагу. Люди вже не можуть уявити своє життя без використання телефонів, комп'ютерів та інших технологій. Але чим далі йде розвиток, тим більше з'являється вимог, яких ми повинні дотримуватися. Усім нам відома така річ, як IP-адреса. Та мало хто цікавився, скільки їх повинно існувати, щоб кожному пристрою дісталася своя унікальна адреса.

З появою протоколу IPv6 виникла потреба в переході від IPv4 до IPv6. Можливо, швидше та практичніше, було б усім одночасно перейти від формату IPv4 до IPv6, але цей варіант потребує значних коштів і тому не є найбільш ефективним. Найбільш логічний варіант у цьому випадку, змусити працювати обидва протоколи в одній мережі та поступово переходити до нового формату. Такий підхід вимагає багато одноманітної роботи від системних адміністраторів.

Вирішенням цієї проблеми та пришвидшенням роботи адміністраторів можна вважати створення графічного інтерфейсу. Він дасть змогу при введенні певних індивідуальних даних автоматично генерувати код налаштування на роутерах. Таким чином, навіть мало досвідченні адміністратори матимуть змогу швидко та якісно налаштовувати роутери. При такому підході, великій компанії, що потребує переходу від протоколу IPv4 до IPv6, або налаштування цих протоколів одночасно, не знадобиться багато людей для цієї роботи. Для реалізації графічного інтерфейсу спочатку потрібно ознайомитися з самою технологією IPv6 та порівняти її з попередніми версіями. Потрібно опрацювати методи роботи з даною технологією та зрозуміти які з них найбільш ефективні. Необхідно впевнитися, чи дійсно ця технологія необхідна, чи ми можемо обійтися і без неї? Наступним кроком потрібно ознайомитися з інструментами, що допоможуть створити інтерфейс. Такими, як HTML, CSS та JavaScript. Саме

їх буде достатньо, для створення функціонального та інформативного інтерфейсу та розробки досить приємного дизайну.

## 1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

### 1.1. Протокол IPv4

IPv4 являє собою запис у вигляді набору чотирьох десяткових чисел від 0 до 255. Даний протокол використовує 32-бітні адреси.

**172 . 16 . 254 . 1**

Рисунок 1.1 - IPv4-адреса

На Рисунку 1.1 представлений зовнішній вигляд IPv4-адреси. Це чотири групи десяткових чисел від 0 до 255, розділених між собою крапками.

Адреси IPv4 поділяються на декілька класів:

- А - мережі з 1.0.0.0 до 127.0.0.0. Номер мережі знаходиться у першому октеті. Це забезпечує 24-ох розрядну частину для позначення хостів. Дозволяє використання приблизно 16 мільйонів хостів у мережі. Зазвичай даний клас використовують для дуже великих мереж.
- В - вміщає мережі з 128.0.0.0 до 191.255.0.0; номер мережі знаходиться у перших двох октетах. Нараховує 16320 мереж з 65024 хостами у кожній. В основному використовують для роботи з великим та середнім розміром мережі.
- С - діапазон мереж від 192.0.0.0 до 223.255.255.0; номер мережі — три перших октети. Нараховує близько 2 мільйонів мереж з 254 хостами у кожній. Призначена для роботи з малими мережами.
- D, E, та F - адреси, які налічують діапазон з 224.0.0.0 до 254.0.0.0 є або дослідними, або збережені для використання у майбутньому і не описують будь-якої мережі. Клас D був створений для налаштування механізму

багатоадресної розсилки. Клас E хоч і виділили в окремий блок, але він був зареєстрований, як проблемна група для проектування.

Протокол IPv4 підтримує три режими адресації:

- Одноадресний. При використанні цього режиму дані передаються тільки на один мережевий вузол, причому кожен з них може бути як відправником, так і одержувачем. Поле адреси призначення містить 32-розрядний IP-адресу пристрою-одержувача. Одноадресний режим використовується найчастіше при зверненні до інтернет-протоколу.

- Широкомовний. При його використанні всі пристрої, підключені до мережі з множинним доступом, мають можливість отримання і обробки даних, переданих по протоколу TCP / IPv4. Для цього поле IP-адреси призначення включає в себе спеціальний широкомовний код ідентифікації.

- Багатоадресний. Згідно з правилами обробки даних по протоколу IPv4, сюди входять адреси в діапазоні від 224.0.0.0 до 239.255.255.255. Режим об'єднує два попередніх, визначається найбільш значущою моделлю 1110. У цьому пакеті адреса призначення містить спеціальний код, який починається з 224.x.x.x і може використовуватися більш ніж одним вузлом.

Для мереж розміром / 24 або більше, широкомовна адреса завжди закінчується 255. Проте, це не означає, що кожен адреса, що закінчується на 0 або 255, не може бути використаний в якості адреси хоста.

Наприклад, в / 16 підмережі 192.168.0.0/255.255.0.0, що еквівалентно діапазону адрес 192.168.0.0-192.168.255.255, широкомовна адреса - 192.168.255.255. Можна використовувати такі адреси для хостів, навіть якщо вони закінчуються 255: 192.168.1.255, 192.168.2.255 і т.д. Крім того, 192.168.0.0 є ідентифікатором мережі і не повинен бути віднесений до інтерфейсу. Адреси 192.168.1.0, 192.168.2.0 і т.д., можуть бути призначені, незважаючи на закінчення з 0.



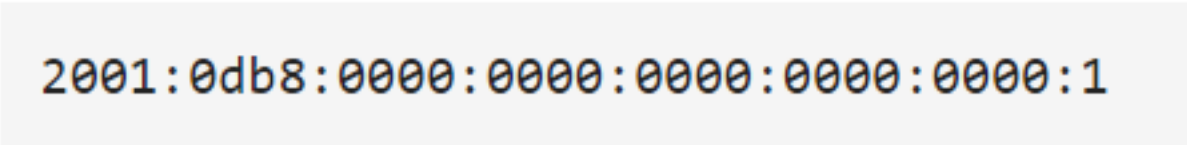
У мережах менших, ніж / 24, широкомовні адреси не обов'язково закінчуються 255. Наприклад, CIDR підмережа 203.0.113.16/28 має широкомовну адресу 203.0.113.31.

При всіх своїх перевагах протокол інтернету IPv4 має один критичний недолік. Кількість адрес, створених з його допомогою, не може перевищити цифру  $4\ 294\ 967\ 296$  (мінімальний адреса - 0.0.0.0, максимальний - 255.255.255.255).

З урахуванням того, що населення земної кулі становить понад семи мільярдів чоловік, а кількість всіляких мережевих пристроїв зростає щодня, граничний поріг досить близький.

## 1.2. Протокол IPv6

IPv6 являє собою запис у вигляді набору восьми груп шістнадцяткових чисел, розділених двукрапками. Даний протокол використовує 128-бітні адреси. Адресний простір цього протоколу досить великий, а локальна підмережа використовує 64 біти для формування частини, що приймає адрес. Інші 64 біти використовують як префікс для маршрутизаторів.



2001:0db8:0000:0000:0000:0000:0000:1

Рисунок 1.2 - IPv6-адреса

На Рисунку 1.1, приведено приклад IPv6-адреси. На перший погляд вона має доволі громіздкий вигляд і здається, що з нею складно працювати. Саме тому є декілька правил скорочення:

- Якщо блок складається з чотирьох нулів, то можемо записати його у вигляді одного нуля(0000 →0)

- Якщо декілька блоків підряд складаються з нулів, можемо випустити їх, поставивши дві двокрапки, але тільки в одному місці.
- Слід віддавати перевагу прописним літерам перед заголовними(DB8→db8), але запис заголовними літерами помилкою не являється
- Якщо блок починається з нуля, а далі йдуть літери або цифри, то нуть можна не писати(00db→db)

В протоколі існує декілька типів адрес, вони допомагають визначити спосіб адресації пакетів:

- Одноадресна – адреси, які прив'язані до одного мережевого інтерфейсу. У цьому випадку відразу відомо, хто отримає даний пакет за цією адресою
- Групові – адреси, що представляють собою групу мережевих інтерфейсів. При відправленні, пакет потрапить до тієї адреси з групи, що стоїть найближче в таблиці маршрутизації. Такий підхід можна використовувати, коли необхідно симулювати роботу одного комп'ютера, але розподілити навантаження між декількома
- Багатоадресні – адреси, що дозволяють групове мовлення. При відправленні, пакети досягнуть всіх мережевих інтерфейсів, прив'язаних до даної адреси

За правилами протоколу IPv6 призначення мережевих адрес відбувається автоматично та є унікальним, за рахунок ідентифікації на рівні MAC-адреси певної одиниці обладнання, яка потребує виходу в публічну мережу. Тобто, кожен з приладів, що має функцію підключення до зовнішніх пристроїв, отримує власну IP-адресу для з'єднання з іншими хостами через інтернет.

Для того, щоб більш детально ознайомитися з типами адрес, слід приділити увагу таким поняттям, як префікс або ж мережева частина адреси.

Протокол IPv6 не використовує маски підмережі у вигляді десяткових чисел, розділених крапками.

Довжина префікса позначає мережеву частину IPv6-адресу за допомогою адреси або довжини префікса. Довжина префікса має бути в діапазоні від 0 до 128. Для локальних та інших типів мереж традиційно використовують префікс - /64.

### 1.3. Порівняння IPv6 та IPv4

Протокол IPv6 можна вважати величезним кроком в історії розвитку IP. Зараз відбувається поступовий перехід від четвертої версії до шостої. Але чому саме так відбувається та в чому їх різниця?

Основною різницею та перевагою протоколу IPv6 є розширення адресного простору. Так, як розмір адреси збільшився від 32 біт до 128, з'явилася більша можливість комбінацій, що формують унікальні IP-адреси. Розширення адрес стало передумовою зміни їх зовнішнього вигляду, тобто відбувся перехід від десяткової системи до шістнадцяткової.

У протокола IPv4 відсутня безпека. Він не був створений з метою забезпечування безпеки. Спочатку він був створений для ізольованої військової мережі, а вже потім адаптований для суспільства. На противагу своєму попереднику, IPv6 має влаштовану систему захисту, таку, як шифрування та аутентифікація.

Багато інтернет-провайдерів мають підключення до IPv4, а до IPv6 у багатьох із них підключення немає.

У протокола IPv6 існує ряд переваг перед IPv4:

- IPv6 значно спростив маршрутизацію у порівнянні з IPv4
- IPv6 більш сумісний з мобільними мережами, ніж IPv4
- IPv6 дозволяє збільшити корисне навантаження, у порівнянні з тим,

що дозволено в IPv4

- IPv6 використовується менше ніж 1% мереж, тоді коли IPv4 ще досі використовує залишивці 99%

Не можна стверджувати, що протокол IPv6 є безпечнішим та швидшим, за протокол IPv4, але він має ряд переваг, у вигляді більш ефективної маршрутизації без фрагментації пакетів, влаштованій підтримки IPsec та автоконфігурації адрес. А саме через необхідність розширення адресного простору, перехід на протокол IPv6 є необхідним.

Зараз, технологія IPv6 знаходиться у стадії активного розвитку. Ведеться багато дискусій, у тому числі і на тему безпеки даних у випадку гібридного застосування обох протоколів. Провайдери налаштовують IPv6-тунелі, для того щоб надати доступ користувачам IPv4 доступ до високорівневого контенту. Нажаль, використання цієї технології, збільшує ризики хакерських атак. Функція автоконфігурації, коли пристрої самостійно генерують IP-адресу на основі MAC-адреси, може бути використана для незаконного відстеження конфіденціальних даних користувачів.

Саме через неможливість швидкого переходу від IPv4 до IPv6, існує необхідність стримувати проблему недостачі унікальних мережевих ідентифікаторів, за рахунок використання динамічних IP-адрес протоколу IPv4.

Тобто, проблеми адміністрування локальних мереж переходять на кінцевих користувачів, які повинні налаштовувати складні схеми маршрутизації підмереж та купувати додаткові IP-адреси.

У той же час, приріст кількості мережевих пристроїв відбувається дуже швидко. Впровадження технологій прямої конфігурації потребує нових підходів в налаштуванні архітектури їх взаємодії. У зв'язку з цим, перехід на використання протоколу IPv6 неминучий.

## 1.4. Способи роботи з протоколами

Взагалі існує три основні способи праці з IPv6 та IPv4 – це подвійний стек, NAT64 та тунелювання.

NAT64 являється своєрідним перекладачем між протоколами IPv4 та IPv6. Данна технологія полегшує зв'язок між хостами, використовуючи форму трансляції мережевих адрес.



Рисунок 1.3 - Метод NAT64

Клієнт IPv6 вбудовує адресу IPv4, з якою бажає взаємодіяти, використовуючи хост-частину сегмента мережі IPv6, в результаті чого вбудовані IPv4-адреси IPv6, і відправляє пакети до отримана адреса. Шлюз NAT64 створює зіставлення між адресами IPv6 та IPv4, які можуть бути налаштовані вручну або визначені автоматично.

Проста установка NAT64 може складатися із шлюзу з двома інтерфейсами, підключеними до мережі IPv4 та мережі IPv6, відповідно. Трафік з мережі IPv6 передається через шлюз, який виконує всі необхідні трансляції для передачі пакетів між двома мережами.

Однак переклад не є симетричним, оскільки адресний простір IPv6 набагато більший за простір адреси IPv4; таким чином, однозначне відображення адрес неможливо.

Шлюз підтримує відображення адрес IPv6-в-IPv4, яке може бути встановлене за допомогою автоматичного алгоритму (відображення без стану) або за допомогою спеціальних та ручних перекладів (відображення стану), коли перший пакет з мережі IPv6 досягає шлюзу NAT64.

Переклад без статусу доречний, коли перекладач NAT64 використовується перед серверами, що мають лише IPv4, щоб дозволити їм отримати доступ віддаленим клієнтам, що мають лише IPv6. Переклад з підтримкою стану підходить для розгортання на стороні клієнта або у постачальника послуг, дозволяючи клієнтським хостам лише IPv6 діставатись до віддалених вузлів лише IPv4.

Загалом, NAT64 призначений для використання, коли зв'язок ініціюється хостами IPv6. Існують деякі механізми, включаючи статичне відображення адрес, щоб дозволити зворотний сценарій.

Не кожен тип ресурсу доступний за допомогою NAT64. Протоколи, що вбудовують адреси літералів IPv4, такі як SIP і SDP, FTP, WebSocket, Skype, MSN та будь-який інший вміст з літералами IPv4, виключаються, але подвійний стек веб-проксі дозволяє клієнтам лише для IPv6 отримувати доступ навіть до веб-сторінок з IPv4 літерали в URL-адресах.

Подвійний стек - це механізм забезпечення сумісності між IPv6 та IPv4. Він вважається найпростішим способом налагодження взаємодії. Принцип його дії полягає у тому, щоб на кожному вузлі мережі, який працює з IPv6 і якому потрібна взаємодія з IPv4-хостами, встановлюємо стек протоколу IPv4, після цього, йому виділяється IPv4-адреса. Тепер даний вузол може взаємодіяти з іншими хостами, що працюють з різними версіями протоколу IP.

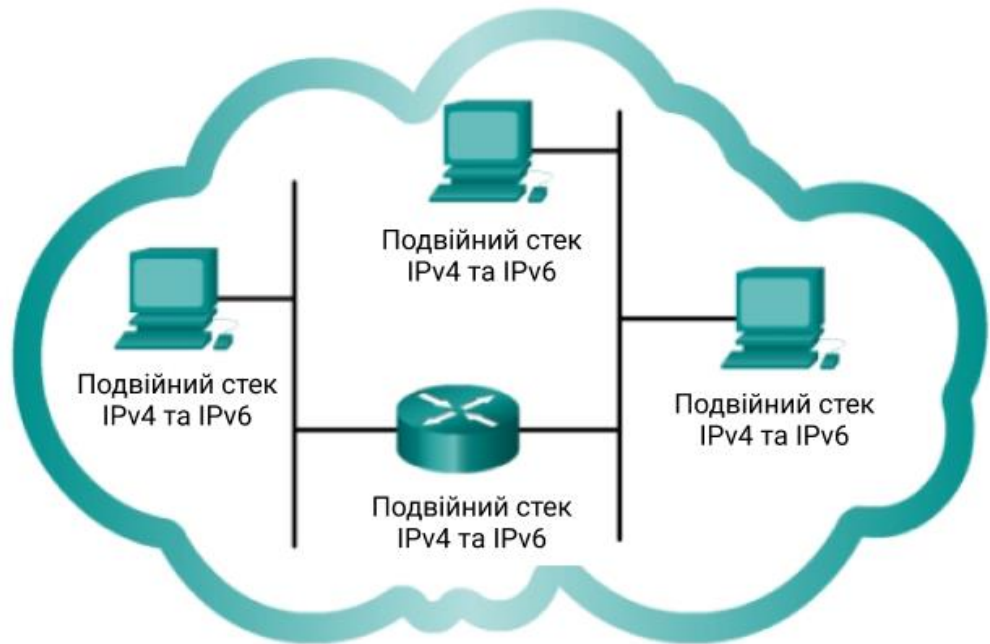


Рисунок 1.4 - Метод подвійного стеку

Але в налаштуванні подвійного стеку є й недоліки. Наприклад, для того, щоб подвійний стек повноцінно працював, необхідно, щоб більшість проміжних маршрутизаторів глобальної мережі працювали, як з IPv6, так і з IPv4.

Також, для використання подвійного стеку системні адміністратори повинні встановити та налаштувати спеціальне програмне забезпечення на кожному з вузлів. Це потребує додаткового часу для спеціалістів та певних грошових витрат на закупку програмного забезпечення.

Ще одним недоліком використання подвійного стеку є те, що він підвищує використання системних ресурсів вузлів мережі і може сповільнити їх роботу.

Метод тунелювання полягає у тому, що будується мережа накладання, яка тунелює один протокол над іншим. При цьому пакети IPv6 інкапсулюються у IPv4, а пакети IPv4 у пакети IPv6. Перевагою такого підходу можна вважати, що

новий протокол буде працювати, не порушуючи старий протокол, забезпечуючи зв'язок між користувачами нового протоколу.

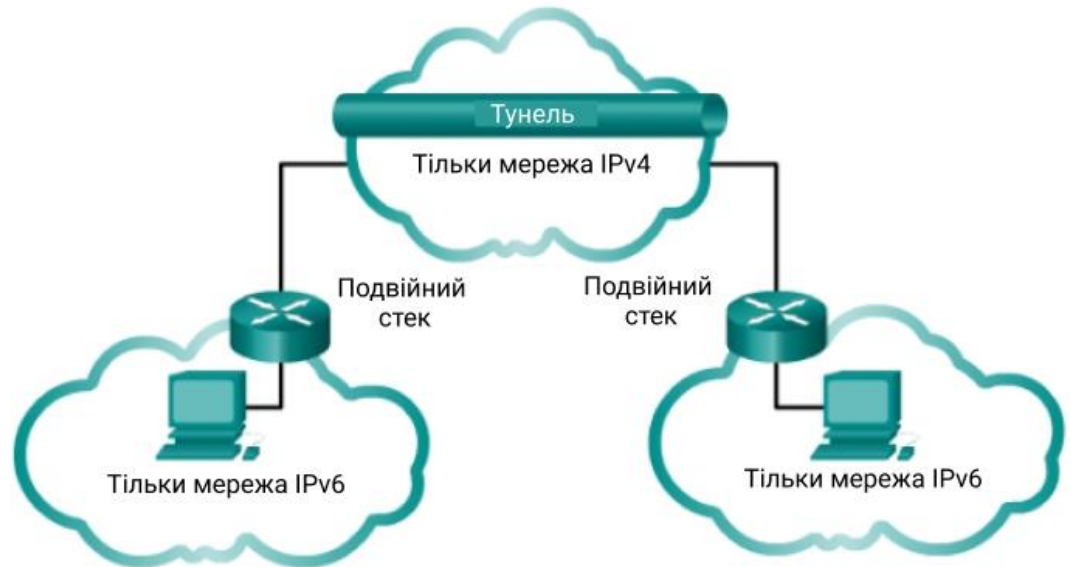


Рисунок 1.5 - Метод тунелювання

Але тунелювання також має недоліки. Користувачі нової архітектури не можуть користуватися послугами базової інфраструктури. А також, тунелювання не дозволяє користувачам нового протоколу спілкуватися з користувачами старого протоколу без подвійних стекових хостів, що забезпечує взаємодію.

## **1.5. Огляд методів проектування та реалізації графічного інтерфейсу**

Для створення графічного інтерфейсу необхідно ознайомитися з такими інструментами, як HTML, CSS, JavaScript. Їх буде достатньо для створення зручного інтерфейсу, що буде практичним у використанні та приємним для користувачів. Розглянемо більш детально кожен з інструментів.



HTML – це стандартизована мова розмітки веб-сторінок. HTML код обробляється браузером, а результат обробки відображається на екрані пристрою користувача.

Для створення графічного інтерфейсу використовувалася п'ята версія HTML. Вона необхідна для розміщення необхідних елементів на веб-сторінці. За допомогою HTML можна створити форми, таблиці, додати малюнки та розмістити текстові блоки.

CSS – це формальна мова, що необхідна для налаштувань зовнішнього виду веб-сторінки, також називається мовою розмітки.

При розробці інтерфейсу була використана третя версія CSS. Вона необхідна для задання кольорів, стилів, шрифтів та розміщення блоків на сторінці. Простіше кажучи, CSS необхідний для придання сайту приємного вигляду.

JavaScript – об'єктно-орієнтована мова програмування. Вона являється однією з найбільш популярних мов програмування. В інтерфейсі використовувалася для задання логіки сторінки.

## **1.6. Постановка задачі**

Метою наукової роботи є створення графічного інтерфейсу, для інтерактивного налаштування роутерів CISCO. Функціонал даного інтерфейсу спрямований на полегшення та пришвидшення роботи з протоколом IPv6.

При введенні початкових даних користувачем, повинен автоматично згенеруватися код налаштування для роутерів.

Постановка задачі:

- Ознайомлення з існуючими проблемами та методами їх вирішення
- Налаштування маршрутизації з використанням роутерів CISCO
- Створення інтерфейсу для роботи з протоколом IPv6
- Тестування створеного інтерфейсу



## 2 НАЛАШТУВАННЯ МЕРЕЖІ

### 2.1. Налаштування вузлів на основі протоколу IPv6

Налаштування IPv6-адрес на вузлах можна вважати навіть простішим, ніж IPv4.

IPv4 Address	10.10.1.100
Subnet Mask	255.255.255.224
Default Gateway	10.10.1.97
DNS Server	0.0.0.0
IPv6 Configuration	
<input type="radio"/> Automatic	<input checked="" type="radio"/> Static
IPv6 Address	2001:DB8:1:1::A / 64
Link Local Address	FE80::202:17FF:FEAA:BA82
Default Gateway	FE80::1
DNS Server	

Рисунок 2.1 - Налаштування IPv4 та IPv6 на комп'ютерах

На рисунку представленні задання IPv4 та IPv6 адрес. Можна спостерігати, що в обох випадках зроблена майже однакова кількість роботи. Ми однаково задаємо два параметри IP-адресу та Gateway, але в різних форматах, і у другому випадку нам вже не потрібно задавати маску, а лише вказати префікс - /64, саме стільки бітів відведено під адресу в мережі.

### 2.2. Подвійний стек

EIGRP – це один з протоколів маршрутизації, що був оптимізований після змін в мережі для більш економного та ефективного використання потужностей маршрутизатора. За допомогою цього протоколу можна легко та швидко налаштувати маршрутизацію з підтримкою протоколу IPv6.

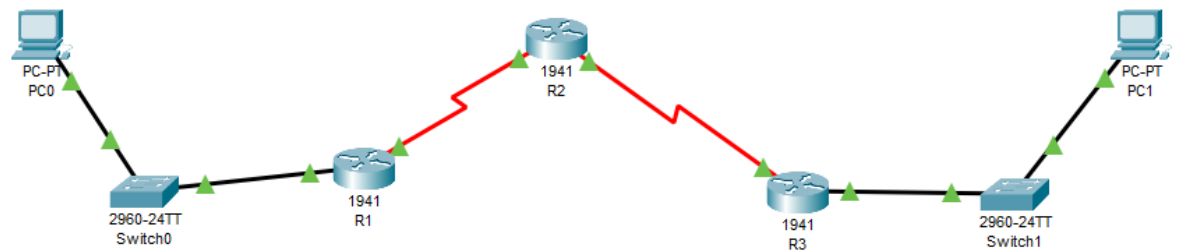


Рисунок 2.2 - Схема для налаштування подвійного стеку

Спочатку, необхідно задати IP-адреси на роутерах (так, як налаштування дуже схожі, для прикладу візьмемо лише R2):

```
Router>en
Router#conf t
R2(config)#interface Serial0/0/0
R2(config-if)#no sh
R2(config-if)# ipv6 address FE80::2 link-local
R2(config-if)# ipv6 address 2001:DB8:1:2::1/64
R2(config-if)#ex
R2(config)#interface Serial0/0/1
R2(config-if)#no sh
R2(config-if)# ipv6 address FE80::2 link-local
R2(config-if)# ipv6 address 2001:DB8:1:3::1/64
```

У командах вище, приведено приклад, як перейти в режим розробника, увійти до інтерфейсу та задати IP-адресу. Слід не забувати вмикати кожен з інтерфейсів, та задавати локальну адресу.

Що стосується самої маршрутизації з використанням протоколу IPv6, то тут все значно простіше. Якщо при використанні протоколу IPv4 було необхідно

вводити номер мережі та маску, то у цьому випадку потрібно лише увімкнути протокол EIGRP та задати його на всіх інтерфейсах.

```
R2(config)#ipv6 router eigrp 1
```

```
R2(config-rtr)#no sh
```

```
R2(config)#int int se 0/0/0
```

```
R2(config-if)#ipv6 eigrp 1
```

```
R2(config-if)#int se 0/0/1
```

```
R2(config-if)#ipv6 eigrp 1
```

Для перевірки результатів налаштувань, необхідно відправити запит з одного комп'ютера на інший, як це показано на рисунку 2.3.

```
C:\>ping 2001:DB8:1:4::A

Pinging 2001:DB8:1:4::A with 32 bytes of data:

Reply from 2001:DB8:1:4::A: bytes=32 time=10ms TTL=125
Reply from 2001:DB8:1:4::A: bytes=32 time=9ms TTL=125
Reply from 2001:DB8:1:4::A: bytes=32 time=24ms TTL=125
Reply from 2001:DB8:1:4::A: bytes=32 time=2ms TTL=125
```

Рисунок 2.3 - Результат налаштувань

У результаті можемо спостерігати, що пакет був доставлений без втрат, тобто схема налаштована вірно.

Розглянемо ще одне налаштування маршрутизації. Для того, щоб налаштувати статичну маршрутизацію, візьмемо ту саму схему, що й у першому випадку, з налаштованими лише IP-адресами. Далі необхідно ввести налаштування на роутерах.

```
R1(config)#ipv6 route ::/0 se 0/0/1
```

```
R3(config)#ipv6 route ::/0 se 0/0/1
```

```
R2(config)#ipv6 route 2001:DB8:1:1::0/64 se 0/0/0
```

```
R2(config)#ipv6 route 2001:DB8:1:4::0/64 se 0/0/1
```

У цьому випадку ми чітко прописуємо маршрут, по якому повинні рухатися пакети. Тепер перевіримо налаштування.

```
C:\>ping 2001:DB8:1:4::A

Pinging 2001:DB8:1:4::A with 32 bytes of data:

Reply from 2001:DB8:1:4::A: bytes=32 time=10ms TTL=125
Reply from 2001:DB8:1:4::A: bytes=32 time=9ms TTL=125
Reply from 2001:DB8:1:4::A: bytes=32 time=24ms TTL=125
Reply from 2001:DB8:1:4::A: bytes=32 time=2ms TTL=125
```

Рисунок 2.4 - Результат налаштувань статичної маршрутизації

На Рисунку 2.4 бачимо, що налаштування були вірними, тому що пакет доставлено без втрат.

Повний об'єм команд приведено у додатку.

### 2.3. Тунелювання

Для налаштування методу тунелювання візьмемо схему, представлену на Рисунку 2.5. Вона складається з п'яти роутерів, на нижніх інтерфейсах IP-адреси протоколу IPv6, а на верхніх адреси протоколу IPv4, відповідно до позначень на рисунку. А між роутерами 2 та 4 потрібно прокласти тунель, через який будуть проходити запити протоколу IPv6.

Такий підхід вважається доречним, коли не всі роутери підтримують протокол IPv6. На Рисунку 2.5, представлений шлях тунелю від другого до четвертого роутера. Таким чином, третій роутер, на якому немає налаштувань з підтримкою протоколу IPv6, не стане перешкодою у передачі повідомлень.

Переходячи до реалізації, постають такі задачі:

- Налаштувати IP-адреси на всіх інтерфейсах
- Прокласти тунель між необхідними роутерами
- Налаштувати маршрутизацію
- Перевірити результати роботи

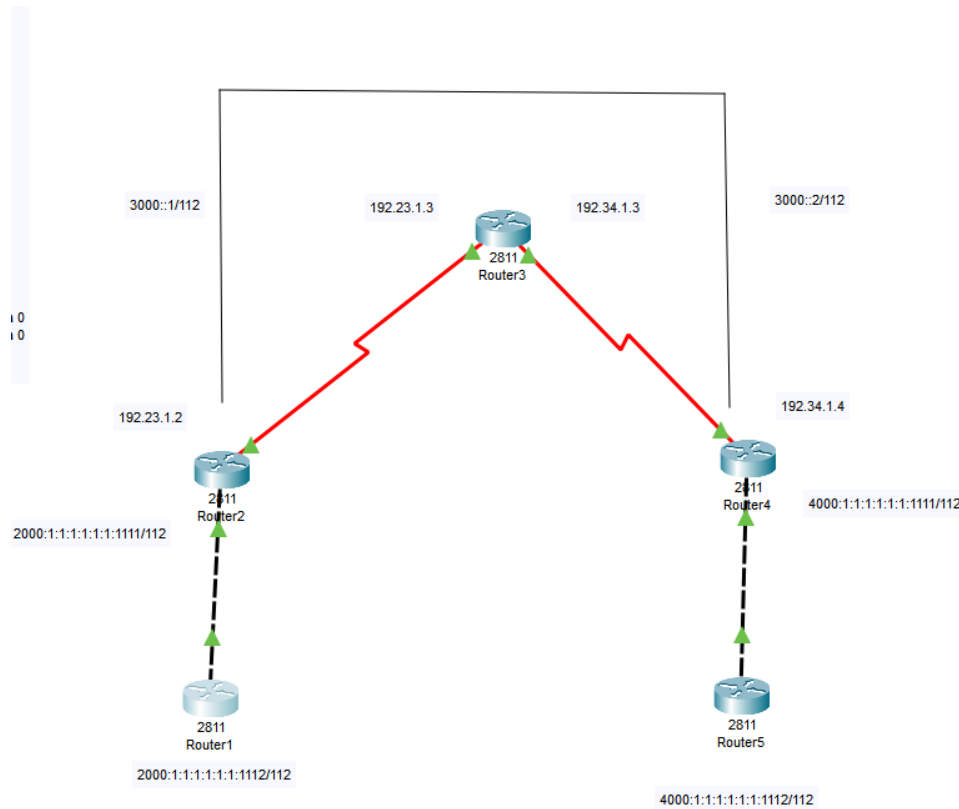


Рисунок 2.5 - Схема для налаштування тунелювання

Для початку задамо IP-адреси на всіх інтерфейсах, для всіх роутерів, відповідно до позначень на Рисунок 2.5. таким чином на другому роутері будуть такі налаштування:

```
Router(config)#int serial 0/3/0
Router(config-if)#ip add 192.23.1.2 255.255.255.0
Router(config-if)#no sh
Router(config-if)#int fa0/0
Router(config-if)#ipv6 add 2000:1:1:1:1:1:1:1111/112
Router(config-if)#no sh
```

На першому та п'ятому роутерах досить прості та ідентичні налаштування. На другому та четвертому роутерах потрібно налаштувати тунель. Для цього вводимо такі команди на другому роутері:

```

Router(config)#int tunnel0
Router(config-if)#ipv6 add 3000::1/112
Router(config-if)#ipv6 rip бbone enable
Router(config-if)#tunnel source serial 0/3/0
Router(config-if)#tunnel destination 192.34.1.4
Router(config-if)#tunnel mode ipv6ip
Router(config-if)#ex

```

Для налаштувань ми вказуємо умовну адресу та який номер інтерфейсу буде використовуватися для виходу в тунель, в даному випадку, це адреса 3000::1/112 та номер інтерфейсу 0/3/0.

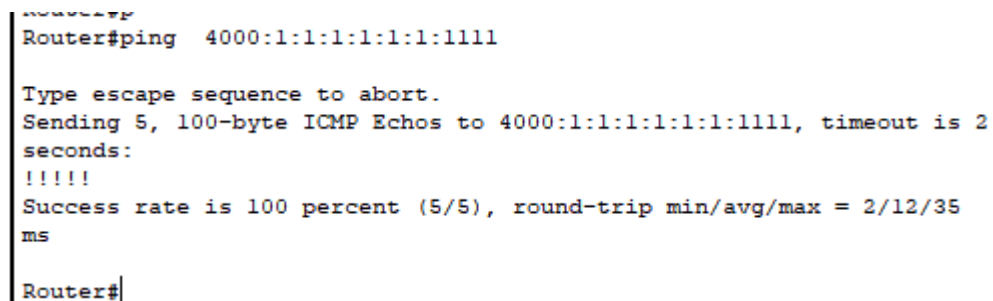
Також, налаштуємо ospf, для третього роутера це будуть команди:

```

Router(config)#router ospf 1
Router(config-router)#network 192.23.1.0 0.0.0.255 area 0
Router(config-router)#network 192.34.1.0 0.0.0.255 area 0

```

Результат відправлення запиту з першого роутера на п'ятий можемо спостерігати на Рисунок 2.6.



```

Router#ping 4000:1:1:1:1:1:1:1111
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4000:1:1:1:1:1:1:1111, timeout is 2
seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/12/35
ms
Router#|

```

Рисунок 2.6 - Результат роботи тунелювання

Повний об'єм команд приведено у додатку.

## 2.4. NAT64

Для налаштування схеми з використанням NAT64 було обрано схему, що представлена на Рисунок 2.7.



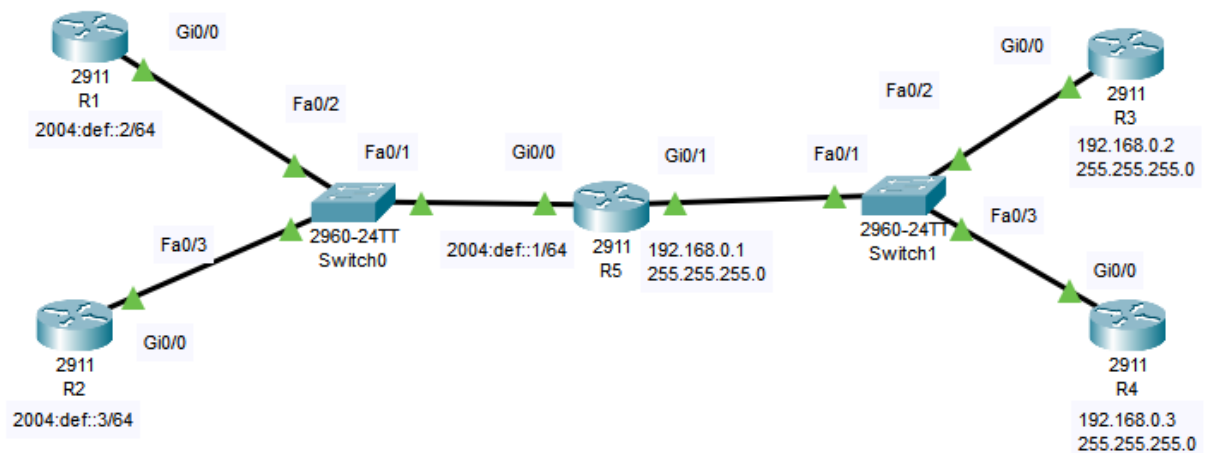


Рисунок 2.7 - Схема для налаштування NAT64

Перше, що необхідно зробити, це налаштувати IP-адреси на інтерфейсах. Зліва будуть IPv6-адреси, справа IPv4-адреси, а в центрі буде роутер з налаштуваннями NAT64.

Для налаштування IP-адрес необхідно ввести:

```
Router>en
```

```
Router#conf t
```

```
Router(config)#int gi0/0
```

```
Router(config-if)#ipv6 add 2004:def::2/64
```

```
Router(config-if)#no sh
```

Аналогічні команди необхідно вводити на всіх роутерах, замінюючи IP-адреси відповідно до необхідних. Також, слід пам'ятати, що на роутерах, де використовується IPv6 слід вводити команду:

```
Router(config)#ipv6 unicast-routing
```

Також на перших чотирьох роутерах слід налаштувати маршрутизацію:

```
Router(config)#ipv6 route 2003:12::/96 2004:def::1, для IPv6 та:
```

Router(config)#ip route 172.16.0.0 255.255.255.0 192.168.0.1, для IPv4

Далі, потрібно переходити до налаштувань п'ятого роутера, а саме до NAT64.

R5(config-if)#ipv6 nat

R5(config-if)#ex

R5(config)#ipv6 nat v4v6 source 192.168.0.2 2003:12::2

R5(config)#ipv6 nat v4v6 source 192.168.0.3 2003:12::3

R5(config)#ipv6 nat v6v4 source 2004:def::2 172.16.0.2

R5(config)#ipv6 nat v6v4 source 2004:def::3 172.16.0.3

R5(config)#ipv6 nat prefix 2003:12::/96

При налаштуванні NAT64 ми беремо адресу кожного з роутерів та задаємо йому адресу іншого типу. Наприклад, візьмемо перший роутер. Його IP-адреса 2004:def::2, це адреса протоколу IPv6, тому ми повинні задати йому адресу IPv4 - 172.16.0.2.

Тепер слід перевірити роботу схеми. Для цього потрібно відправити запит з одного роутера на інший. На Рисунку 2.8 представлений результат успішного з'єднання з першого роутера.

```
R1#ping 2003:12::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2003:12::2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms
```

Рисунок 2.8 - Результат роботи NAT64

Повний об'єм команд приведено у додатку.

## 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ГРАФІЧНОГО ІНТЕРФЕЙСУ

### 3.1. Проектування

До початку створення самого графічного інтерфейсу необхідно зрозуміти на кого він орієнтований. Так, як даний інтерфейс несе в собі ціль полегшення роботи системним адміністраторам, він не повинен бути перевантажений зайвою інформацією. Сторінка має містити всю необхідну інформацію, але не бути змістовно перенавантаженою.

Багато хто нехтує етапом прототипування, але він є не менш важливим ніж етап розробки. Саме на цьому етапі розроблюється не лише дизайн, а й продумується логіка сторінки, що буде зрозуміла та проста у використанні користувачеві. Якщо не приділити достатньо уваги прототипу, то можна допуститися багатьох помилок на подальших етапах.

Для розробки прототипу та дизайну інтерфейсу була використана Figma. Це зручний та безкоштовний спосіб розробки будь-якого прототипу.

Як показано на Рисунку 3.1, інтерфейс розроблено достатньо зрозуміло для користувачів. Він має простий, але функціональний вміст та не перенавантажений зайвою навігацією.

На даному інтерфейсі буде розміщений лише один рисунок. Адже, інтерфейс розрахований на те щоб використовуватися у роботі, а не мати якийсь незвичайний та мальовничий дизайн. Тому, якщо додати анімацію, це може вплинути на швидкість роботи сторінки, а це є однією з основних принципів, для зручного користування.

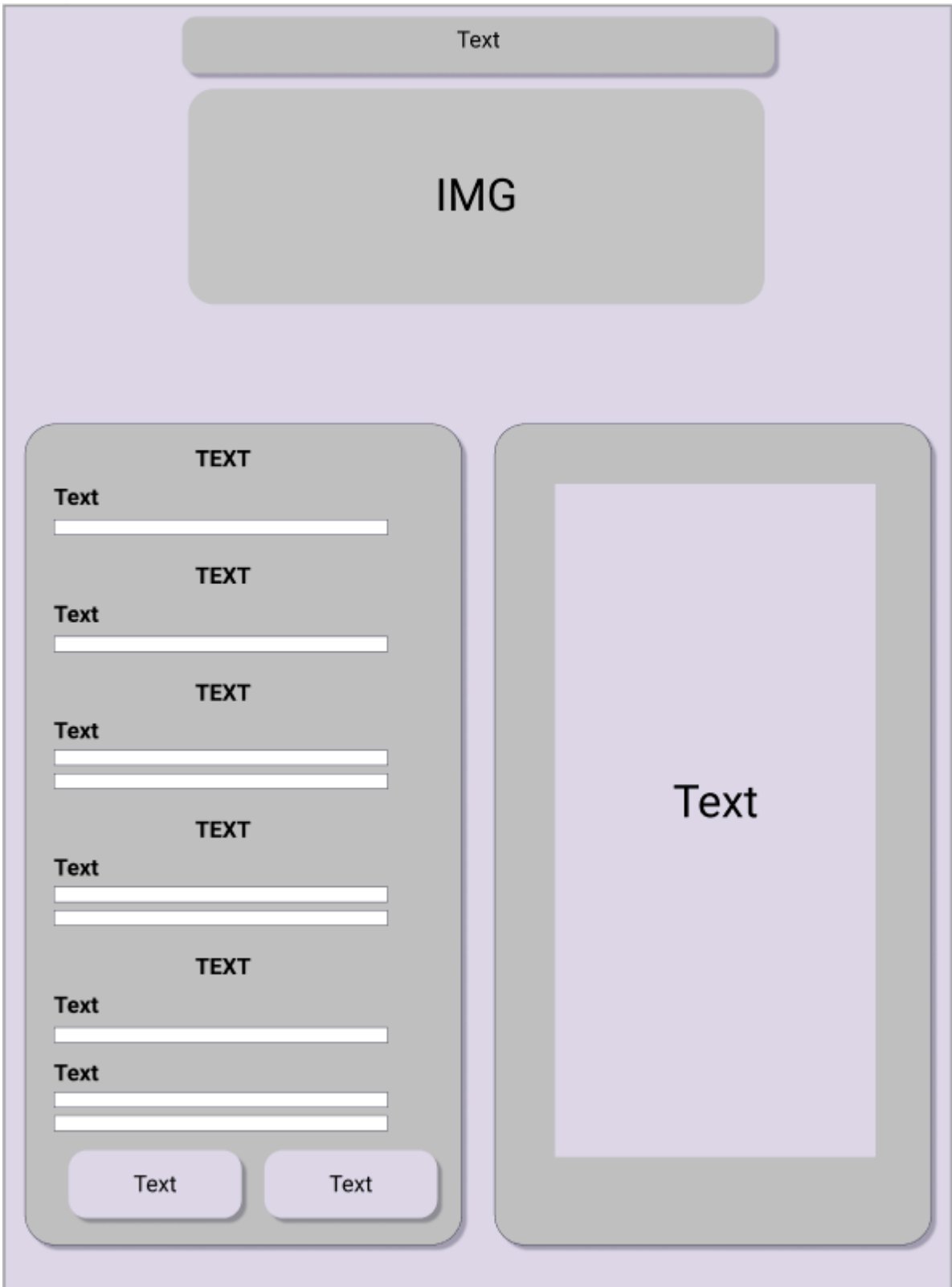


Рисунок 3.1 – Прототип графічного інтерфейсу

### 3.2. Створення дизайну інтерфейсу

Для розробки дизайну інтерфейсу було використано HTML та CSS. Так, як інтерфейс орієнтований для вузького кола користувачів, а саме для системних адміністраторів, не має сенсу перенавантажувати його зайвою анімацією.

Реалізований графічний інтерфейс поєднує в собі не багато кольорів, які добре поєднуються між собою. Це не ускладнює сприйняття інформації користувачем, та не має негативного впливу на сенсори його сприйняття. Також використано лише один тип шрифту, різних розмірів. Це значно покращує та пришвидшує читання тексту.

В шапці сторінки розташована назва та схема для налаштування.

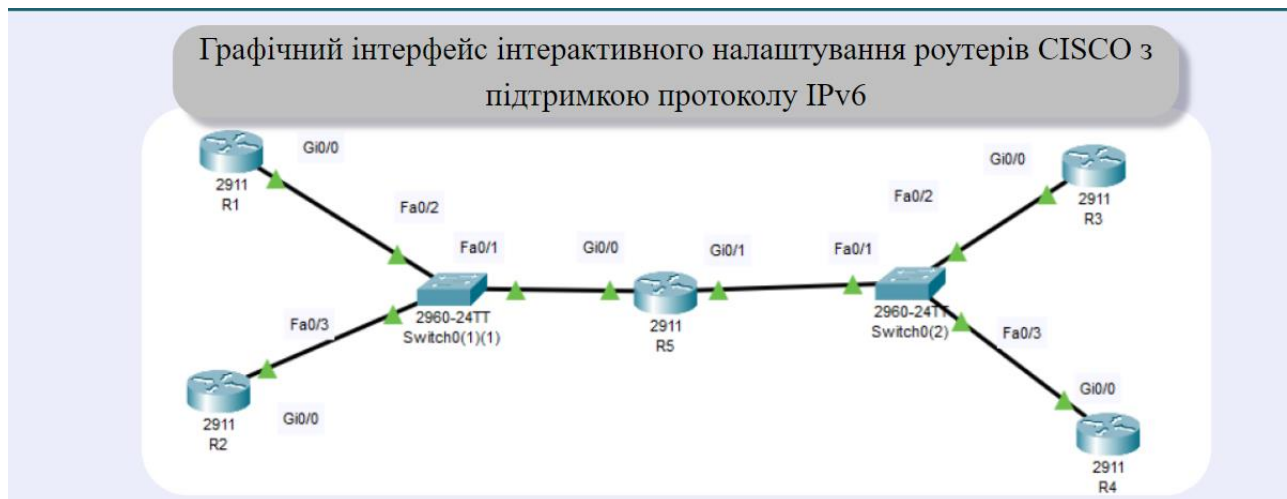


Рисунок 3.2 – Шапка сторінки

На сторінці знаходиться форма, що представлена на Рисунку 3.3. через підписи в інпутах, одразу зрозуміло, IP-адресу якого типу туди потрібно вводити. Також в кінці форми є дві кнопки. Перша – для вставлення базових налаштувань, а друга для генерації коду.

The image shows a web-based configuration interface for five routers, labeled R1 through R5. Each router configuration section includes the following fields:

- R1:** Interface Gi0/0 with an IPv6 address input field.
- R2:** Interface Gi0/0 with an IPv6 address input field.
- R3:** Interface Gi0/1 with IPv4 address and mask input fields.
- R4:** Interface Gi0/1 with IPv4 address and mask input fields.
- R5:** Interface Gi0/0 with an IPv6 address input field, and Interface Gi0/1 with IPv4 address and mask input fields.

At the bottom of the form, there are two buttons: "Базові налаштування" (Basic settings) and "Згенерувати" (Generate).

Рисунок 3.3 – Форма

Ще на сторінці знаходиться блок, у який будуть записуватися згенеровані команди.

### 3.3. Налаштування логіки інтерфейсу

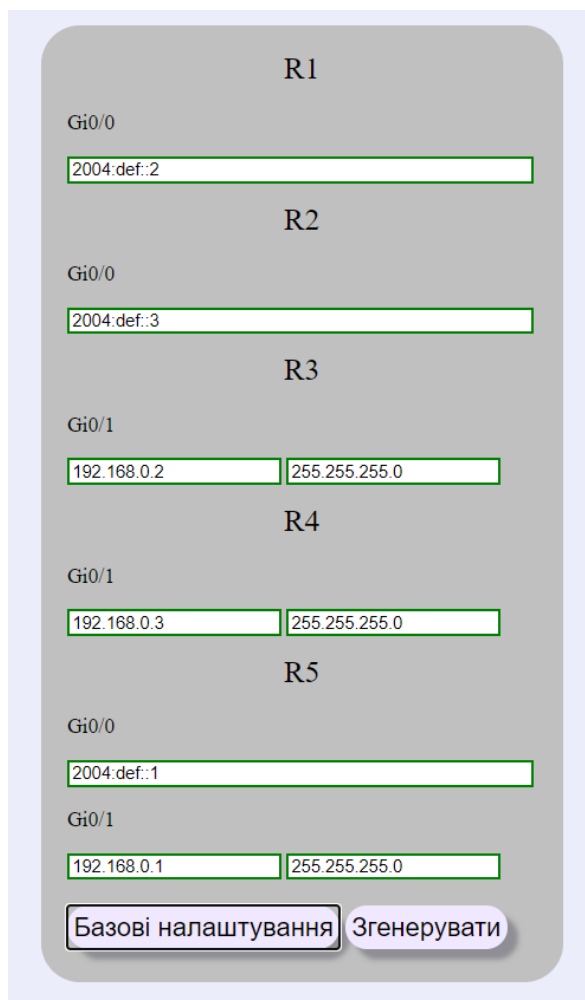
Для налаштування роботи інтерфейсу було обрано об'єктно-орієнтовану мову програмування – JavaScript. Схему було реалізовано за методом NAT64.

Так, як команди для налаштування за цим методом однотипні для різних мереж, то логіку можна реалізувати таким чином. Необхідно створити певний шаблон, у який в певні місця будуть підставлені значення, що введе користувач.

Особливу увагу було приділено валідації. Для того, щоб користувач міг відразу помітити, якщо він буде вводити не коректні значення. Початково поле має червоний контур, але при введенні коректного значення він змінює колір на зелений. Таким чином відразу помітно, якщо в якомусь з полів значення має не коректний формат.

### 3.4. Перевірка роботи інтерфейсу

Для перевірки роботи слід заповнити форму, та натиснути кнопку згенерувати.



The image shows a web-based configuration form for five routers, labeled R1 through R5. Each router has one or two input fields for IP addresses. The fields are currently filled with valid IP addresses, and their borders are green, indicating they are correct. At the bottom of the form, there are two buttons: 'Базові налаштування' (Basic settings) and 'Згенерувати' (Generate).

Router	Interface	IP Address
R1	Gi0/0	2004.def:2
R2	Gi0/0	2004.def:3
R3	Gi0/1	192.168.0.2
R3	Gi0/1	255.255.255.0
R4	Gi0/1	192.168.0.3
R4	Gi0/1	255.255.255.0
R5	Gi0/0	2004.def:1
R5	Gi0/1	192.168.0.1
R5	Gi0/1	255.255.255.0

Рисунок 3.4 – Заповнена форма для тестування

Для прикладу візьмемо базові налаштування. На Рисунку 3.5, бачимо згенерований код для кожного з роутерів. Тепер можна його копіювати та вставляти в вікно налаштування роутера, без подальших змін.

```
R1
en
conf t
ipv6 unicast-routing
int gi0/0
ipv6 add 2004:def::2/64
no sh
ex
ipv6 route 2003:12::/96 2004:def::1

R2
en
conf t
ipv6 unicast-routing
int gi0/0
ipv6 add 2004:def::3/64
no sh
ipv6 route 2003:12::/96 2004:def::1

R3
en
conf t
int gi0/0
ip add 192.168.0.2 255.255.255.0
no sh
ex
ip route 172.16.0.0 255.255.255.0 192.168.0.1

R4
en
conf t
int gi0/0
ip add 192.168.0.3 255.255.255.0
no sh
ip route 172.16.0.0 255.255.255.0 192.168.0.1

R5
en
conf t
ipv6 unicast-routing
int gi0/0
ipv6 add 2004:def::1/64
no sh
ipv6 nat
ex
int gi0/1
ip add 192.168.0.1 255.255.255.0
no sh
ipv6 nat
ex
ipv6 nat v4v6 source 192.168.0.2 2003:12::2
ipv6 nat v4v6 source 192.168.0.3 2003:12::3
ipv6 nat v6v4 source 2004:def::2 172.16.0.2
ipv6 nat v6v4 source 2004:def::3 172.16.0.3
ipv6 nat prefix 2003:12::/96
```

Рисунок 3.5 – Результат роботи інтерфейсу



## ВИСНОВОК

Прихід протоколу IPv6 був лише питанням часу. Через недостатність адресного простору, протокол IPv4 має неминуче відійти у минуле. Тому, результат даної роботи у наш час є досить актуальним.

У ході виконання роботи було розглянуто протоколи IPv6 та IPv4, порівняно їх між собою та виявлено переваги та недоліки. Розкрито тему актуальності даної роботи.

Були розглянуті різні методи роботи з IPv6 та IPv4 одночасно. Проаналізовано дані методи та виявлено переваги та недоліки кожного з них.

Практично реалізовано налаштування схеми на основі використання протоколу IPv6. Розглянуто налаштування маршрутизації за допомогою протоколу EIGRP та налаштовано статичну маршрутизацію.

Також, налаштовано схеми на основі NAT64 та тунелювання. Практично перевірено роботу кожного з методів. Обрано метод, який має більш практичне застосування, та на основі нього створено графічний інтерфейс.

Розроблено графічний інтерфейс, з урахуванням зручності роботи з ним, для налаштування роутерів на основі протоколу IPv6. Враховано, для кого створений інтерфейс. Практично перевірено правильність роботи сторінки.

## СПИСОК ЛІТЕРАТУРИ

1. Аксак, В.А. Новейшая энциклопедия Интернет 2008; Эксмо, 2008. - 912 с.
2. Создание сайта, web-дизайн. – Режим доступа: <http://www.artus.ru/>
3. Басюк Т.М. Принципи побудови системи аналізу та просування інтернетресурсів. Вісник Нац. ун-ту “Львівська політехніка” “Комп’ютерні науки та інформаційні технології”- Львів :2012.– № 784. – С.43–48.
4. Ajax Hacks By Bruce W. Perry, Publisher: O'Reilly, Pub Date: March 2006, Print ISBN-10: 0-596-10169-4, Print ISBN-13: 978-0-59-610169-5, Pages: 438
5. Орлов, А.А. и др. Полная энциклопедия Интернета; АСТ, 2008. - 896 с.
6. Шмитт, К. CSS. Рецепты программирования; М.: Русская редакция; Издание 2-е, 2007. - 592 с.
7. Дунаев В. HTML, скрипты и стили. - Петербург: БХВ, 2011. – 816 с
8. [https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/enterprise-ipv6-solution/white\\_paper\\_c11-676278.html](https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/enterprise-ipv6-solution/white_paper_c11-676278.html)
9. Золотов С. Протоколы Internet – Спб.: ВHV – С-питер, 1998. – 304с.
10. Годд Леммл, Моника Леммл, Джеймс Челлис. Учебное руководство для специалистов MCSE: TCP/IP – М.: ЛОРИ, 1997. -280 с
11. Сэм Хелеби и Денни Мак–Ферсон Принципы маршрутизации в Internet. 2 издание. – М. СПб. Киев: Вильямс, 2001 г;
12. RFC 793. Transmission Control Protocol. J. Postel. Sep-01-1981.
13. RFC 0826. Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware. D.C. Plummer. Nov-01-1982.
14. RFC 959. File Transfer Protocol. J. Postel, J.K. Reynolds. Oct-01-1985.

15. Программа сетевой академии. Cisco CCNA 1 и 2. Вспомогательное руководство, 3-е изд., с испр.: Пер. с англ. – М.: Издательский дом “Вильямс”, 2005. – 1168 с.: ил. – Парал. тит. англ.
16. Эспозито, Д. Разработка приложений для Windows 8 на HTML5 и JavaScript / Д. Эспозито, Ф. Эспозито. - Москва: РГГУ, 2014. - 384 с
17. Мержевич, Влад HTML и CSS на примерах / Влад Мержевич. - М.: "БХВ-Петербург", 2005. - 448
18. Дуванов, А. Web-конструирование. DHTML / А. Дуванов. - М.: БХВ-Петербург, 2003. - 504 с.
19. Хуан, Диего Гоше HTML5. Для профессионалов / Хуан Диего Гоше. - М.: "Издательство "Питер", 2011. - 496 с.

## ДОДАТОК

### HTML

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="author" content="Magister">
    <meta name="description" content="Магістерська робота">
    <meta name="keywords" content="Магістерська, найкраща, робота, 2020">
    <title>IPv6</title>
    <link rel="stylesheet" href="css/style.css">

    <script type="text/javascript" src="js/script.js"></script>
  </head>
  <body>

    <!-- 1 -->

    <header class="topblok">
      <h1>Графічний інтерфейс інтерактивного налаштування
        роутерів CISCO з підтримкою протоколу IPv6</h1>
    </header>

    <!-- 2 -->

    <div class="container containerr">
      
    </div>

    <!-- 3,1 Блоки инпут и текст -->

    <div class="main">

    <div class="block_input">
      <form id="file_form" action="">
        <h1>R1</h1>
        <p>Gi0/0</p>
        <input class="input_length" id="R1" type="text" name="" value="" placeholder="IPv6" required pattern="^((([0-9A-
Fa-f]{1,4}:){7}[0-9A-Fa-f]{1,4})|(([0-9A-Fa-f]{1,4}:){6}:[0-9A-Fa-f]{1,4})|(([0-9A-Fa-f]{1,4}:){5}:([0-9A-Fa-
f]{1,4}:)?[0-9A-Fa-f]{1,4})|(([0-9A-Fa-f]{1,4}:){4}:([0-9A-Fa-f]{1,4}:){0,2}[0-9A-Fa-f]{1,4})|(([0-9A-Fa-
f]{1,4}:){3}:([0-9A-Fa-f]{1,4}:){0,3}[0-9A-Fa-f]{1,4})|(([0-9A-Fa-f]{1,4}:){2}:([0-9A-Fa-f]{1,4}:){0,4}[0-9A-Fa-
f]{1,4})|([0-9A-Fa-f]{1,4}:){6}((\b((25[0-5])|(1\d{2}))|(2[0-4]\d)|(\d{1,2})))\b\.\.){3}(\b((25[0-5])|(1\d{2}))|(2[0-
4]\d)|(\d{1,2})))\b)|([0-9A-Fa-f]{1,4}:){0,5}:((\b((25[0-5])|(1\d{2}))|(2[0-4]\d)|(\d{1,2})))\b\.\.){3}(\b((25[0-
5])|(1\d{2}))|(2[0-4]\d)|(\d{1,2})))\b)|(::([0-9A-Fa-f]{1,4}:){0,5}((\b((25[0-5])|(1\d{2}))|(2[0-
4]\d)|(\d{1,2})))\b\.\.){3}(\b((25[0-5])|(1\d{2}))|(2[0-4]\d)|(\d{1,2})))\b)|([0-9A-Fa-f]{1,4}::([0-9A-Fa-f]{1,4}:){0,5}[0-
9A-Fa-f]{1,4})|(::([0-9A-Fa-f]{1,4}:){0,6}[0-9A-Fa-f]{1,4})|([0-9A-Fa-f]{1,4}:){1,7}:))$">

        <h1>R2</h1>
        <p>Gi0/0</p>
        <input class="input_length" id="R2" type="text" name="" value="" placeholder="IPv6" required pattern="^((([0-9A-
Fa-f]{1,4}:){7}[0-9A-Fa-f]{1,4})|(([0-9A-Fa-f]{1,4}:){6}:[0-9A-Fa-f]{1,4})|(([0-9A-Fa-f]{1,4}:){5}:([0-9A-Fa-
f]{1,4}:)?[0-9A-Fa-f]{1,4})|(([0-9A-Fa-f]{1,4}:){4}:([0-9A-Fa-f]{1,4}:){0,2}[0-9A-Fa-f]{1,4})|(([0-9A-Fa-
f]{1,4}:){3}:([0-9A-Fa-f]{1,4}:){0,3}[0-9A-Fa-f]{1,4})|(([0-9A-Fa-f]{1,4}:){2}:([0-9A-Fa-f]{1,4}:){0,4}[0-9A-Fa-
f]{1,4})|([0-9A-Fa-f]{1,4}:){6}((\b((25[0-5])|(1\d{2}))|(2[0-4]\d)|(\d{1,2})))\b\.\.){3}(\b((25[0-5])|(1\d{2}))|(2[0-
4]\d)|(\d{1,2})))\b)|([0-9A-Fa-f]{1,4}:){0,5}:((\b((25[0-5])|(1\d{2}))|(2[0-4]\d)|(\d{1,2})))\b\.\.){3}(\b((25[0-

```

```
5])|(1\d{2})|(2[0-4]\d|(\d{1,2}))b))(:([0-9A-Fa-f]{1,4}):(0,5)((\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b)\.){3}(\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b))|([0-9A-Fa-f]{1,4}::([0-9A-Fa-f]{1,4}):(0,5)[0-9A-Fa-f]{1,4})|(:([0-9A-Fa-f]{1,4}):(0,6)[0-9A-Fa-f]{1,4})|((([0-9A-Fa-f]{1,4}):(1,7):))$)">
```

```
<h1>R3</h1>
```

```
<p>Gi0/1</p>
```

```
<input id="R3_1" type="text" name="" value="" placeholder="IPv4" required pattern="^(25[0-5]|2[0-4][0-9]|1[0-9][0-9][0-9]{1,2})\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9][0-9]{1,2}){3}$">
```

```
<input id="R3_2" type="text" name="" value="" placeholder="mask" required pattern="^(?:[01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.(?:[01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.(?:[01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])$">
```

```
<h1>R4</h1>
```

```
<p>Gi0/1</p>
```

```
<input id="R4_1" type="text" name="" value="" placeholder="IPv4" required pattern="^(25[0-5]|2[0-4][0-9]|1[0-9][0-9][0-9]{1,2})\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9][0-9]{1,2}){3}$">
```

```
<input id="R4_2" type="text" name="" value="" placeholder="mask" required pattern="^(?:[01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.(?:[01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.(?:[01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])$">
```

```
<h1>R5</h1>
```

```
<p>Gi0/0</p>
```

```
<input class="input_length" id="R5_1" type="text" name="" value="" placeholder="IPv6" required pattern="^(((0-9A-Fa-f){1,4}):(7)[0-9A-Fa-f]{1,4})|((([0-9A-Fa-f]{1,4}):(6):[0-9A-Fa-f]{1,4})|((([0-9A-Fa-f]{1,4}):(5):([0-9A-Fa-f]{1,4}):(0-9A-Fa-f){1,4})|((([0-9A-Fa-f]{1,4}):(4):([0-9A-Fa-f]{1,4}):(0,2)[0-9A-Fa-f]{1,4})|((([0-9A-Fa-f]{1,4}):(3):([0-9A-Fa-f]{1,4}):(0,3)[0-9A-Fa-f]{1,4})|((([0-9A-Fa-f]{1,4}):(2):([0-9A-Fa-f]{1,4}):(0,4)[0-9A-Fa-f]{1,4})|((([0-9A-Fa-f]{1,4}):(6)((\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b)\.){3}(\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b))|((([0-9A-Fa-f]{1,4}):(0,5):(\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b)\.){3}(\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b))|(:([0-9A-Fa-f]{1,4}):(0,5)((\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b)\.){3}(\b((25[0-5])|(1\d{2})|(2[0-4]\d)|(\d{1,2}))\b))|([0-9A-Fa-f]{1,4}::([0-9A-Fa-f]{1,4}):(0,5)[0-9A-Fa-f]{1,4})|(:([0-9A-Fa-f]{1,4}):(0,6)[0-9A-Fa-f]{1,4})|((([0-9A-Fa-f]{1,4}):(1,7):))$)">
```

```
<p>Gi0/1</p>
```

```
<input id="R5_2" type="text" name="" value="" placeholder="IPv4" required pattern="^(25[0-5]|2[0-4][0-9]|1[0-9][0-9][0-9]{1,2})\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9][0-9]{1,2}){3}$">
```

```
<input id="R5_3" type="text" name="" value="" placeholder="mask" required pattern="^(?:[01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.(?:[01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.(?:[01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])$">
```

```
<div class="but_input">
```

```
<button class="but_BN btn" type="button" name="button" onclick="clik1()">Базові налаштування</button>
```

```
<button id="tete" class="but_gen btn" type="button" name="button" onmousedown="viewDiv2(), values()">3генерувати</button>
```

```
</div>
```

```
<div id="messenger"></div>
```

```
</form>
```

```
</div>
```

```
<!-- Блок текст -->
```

```
<div class="block_text">
```

```
<pre id="text_saves">
```

```
R1
```

```
en
```

```
conf t
```

```
ipv6 unicast-routing
```

```
int gi0/0
```

```
ipv6 add <p class="ip_text" id="R1_id"></p>/64
```

```
no sh
```

```
ex
```

```
ipv6 route 2003:12::96 <p class="ip_text" id="R5_1_2id"></p>
```

```

R2
en
conf t
ipv6 unicast-routing
int gi0/0
ipv6 add <p class="ip_text" id="R2_id"></p>/64
no sh
ipv6 route 2003:12::/96 <p class="ip_text" id="R5_1_3id"></p>

```

```

R3
en
conf t
int gi0/0
ip add <p class="ip_text" id="R3_1id"></p> <p class="ip_text" id="R3_2id"></p>
no sh
ex
ip route 172.16.0.0 255.255.255.0 <p class="ip_text" id="R5_2_2id"></p>

```

```

R4
en
conf t
int gi0/0
ip add <p class="ip_text" id="R4_1id"></p> <p class="ip_text" id="R4_2id"></p>
no sh
ip route 172.16.0.0 255.255.255.0 <p class="ip_text" id="R5_2_3id"></p>

```

```

R5
en
conf t
ipv6 unicast-routing
int gi0/0
ipv6 add <p class="ip_text" id="R5_1id"></p>/64
no sh
ipv6 nat
ex
int gi0/1
ip add <p class="ip_text" id="R5_2id"></p> <p class="ip_text" id="R5_3id"></br>
<p class="ip_text1">no sh</p>
<p class="ip_text">ipv6 nat </p>
<p class="ip_text">ex</p>
ipv6 nat v4v6 source <p class="ip_text" id="R3_1_2id"></p> 2003:12::2
ipv6 nat v4v6 source <p class="ip_text" id="R4_1_2id"></p> 2003:12::3
ipv6 nat v6v4 source <p class="ip_text" id="R1_2id"></p> 172.16.0.2
ipv6 nat v6v4 source <p class="ip_text" id="R2_2id"></p> 172.16.0.3
ipv6 nat prefix 2003:12::/96
</pre>
</div>
</div>
</body>
</html>

```

## CSS

```

html, body{
    background-color: #ebedfb;

```

```
}  
.topblok{  
position: relative;  
  background: #C0C0C0;  
box-shadow: 8px 8px 4px rgba(39, 38, 59, 0.33);  
border-radius: 23px;  
margin: 10px 25% 0 25%;  
z-index: 10000;  
/* margin-left: auto;  
margin-right: auto;  
width: 60%; */  
  
}  
h1{  
  text-align: center;  
  font-weight: 400;  
  font-size: 150%;  
  line-height: 35px;  
  margin: 10px;  
  
}  
  
/* 2 */  
  
.containerr{  
  height: 275px;  
}  
  
.imgtop{  
  text-align: center;  
  max-width: 800px;  
  margin-left: auto;  
  margin-right: auto;  
  display: block;  
  margin-top: -17px;  
  z-index: 1;  
  
}
```

```
/* ///////////3///////// */

.main{
  position: relative;
}

.block_input{
  background-color: #C0C0C0;
  max-width: 361px;
  margin-left: 20%;
  margin-top: 2.2%;
  padding: 5px 20px 5px 20px;
  border-radius: 30px;
}

.block_text{
  background-color: #C0C0C0;
  max-width: 350px;
  padding: 5px 20px 5px 20px;
  border-radius: 30px;
  right: 0;
  position: absolute;
  top: 0;
  margin-right: 20%;
  margin-bottom: 40px;
  min-width: 322px;
  min-height: 894px;
}

.but_input{
  margin-top: 20px;
  margin-bottom: 20px;
}

.btn{
  background-color: #f0e8ff;
  border: 0;
  border-radius: 25px;
  padding: 5px;
```



```
    box-shadow: 8px 8px 4px rgba(39, 38, 59, 0.33);
    font-size: 20px;
}
.but_gen{

}
.but_BN{

}
.btn_save{
margin-top: 15px;
display: none;
}
#text_saves{
    display: none;
    margin-bottom: 10px;
}

.ip_text{
    display: contents;
}
.ip_text1{
    display: block;
    margin-left: 15px;
    margin-top: 0;
    margin-bottom: -15px;
}

input:valid{
    border: 2px solid green;
}

input:invalid{
    border: 2px solid red;
}
.input_length{
    width: 97%;
}
```

## JavaScript

```
function klik1(){
    document.getElementById("R1").value = "2004:def::2";
    document.getElementById("R2").value = "2004:def::3";
    document.getElementById("R3_1").value = "192.168.0.2";
    document.getElementById("R3_2").value = "255.255.255.0";
    document.getElementById("R4_1").value = "192.168.0.3";
    document.getElementById("R4_2").value = "255.255.255.0";
    document.getElementById("R5_1").value = "2004:def::1";
    document.getElementById("R5_2").value = "192.168.0.1";
    document.getElementById("R5_3").value = "255.255.255.0";
};
function values(){
    var val = document.getElementById("R1").value;
    document.getElementById("R1_id").textContent= val;
    document.getElementById("R1_2id").textContent= val;
    var val = document.getElementById("R2").value;
    document.getElementById("R2_id").textContent= val;
    document.getElementById("R2_2id").textContent= val;
    var val = document.getElementById("R3_1").value;
    document.getElementById("R3_1id").textContent= val;
    document.getElementById("R3_1_2id").textContent= val;
    var val = document.getElementById("R3_2").value;
    document.getElementById("R3_2id").textContent= val;
    var val = document.getElementById("R4_1").value;
    document.getElementById("R4_1id").textContent= val;
    document.getElementById("R4_1_2id").textContent= val;
    var val = document.getElementById("R4_2").value;
    document.getElementById("R4_2id").textContent= val;
    var val = document.getElementById("R5_1").value;
    document.getElementById("R5_1id").textContent= val;
    document.getElementById("R5_1_2id").textContent= val;
    document.getElementById("R5_1_3id").textContent= val;
    var val = document.getElementById("R5_2").value;
    document.getElementById("R5_2id").textContent= val;
    document.getElementById("R5_2_2id").textContent= val;
    document.getElementById("R5_2_3id").textContent= val;
    var valz = document.getElementById("R5_3").value;
```

```

document.getElementById("R5_3id").textContent = valz
};
function viewDiv2(){
  var field = document.getElementById("R1", "R2", "R3_1", "R3_2", "R4_1", "R4_2", "R5_1", "R5_2", "R5_3").value;
  var textese = document.getElementById("text_saves")
  for (var i = 0; i < field.length; i++) {
    if (i==0 ) {
      textese.style = "display: none";
    }
    else {
      textese.style = "display: block";
    }
  }
};

```

## NAT64

```

R1
en
conf t
ipv6 unicast-routing
int gi0/0
ipv6 add 2004:def::2/64
no sh
ex
ipv6 route 2003:12::/96 2004:def::1
R2
en
conf t
ipv6 unicast-routing
int gi0/0
ipv6 add 2004:def::3/64
no sh
ipv6 route 2003:12::/96 2004:def::1
R3
en
conf t
int gi0/0
ip add 192.168.0.2 255.255.255.0
no sh

```

```

ex
ip route 172.16.0.0 255.255.255.0 192.168.0.1
R4
en
conf t
int gi0/0
ip add 192.168.0.3 255.255.255.0
no sh
ip route 172.16.0.0 255.255.255.0 162.168.0.1
R5
en
conf t
ipv6 unicast-routing
int gi0/0
ipv6 add 2004:def::1/64
no sh
ipv6 nat
ex
int gi0/1
ip add 192.168.0.1 255.255.255.0
no sh
ipv6 nat
ex
ipv6 nat v4v6 source 192.168.0.2 2003:12::2
ipv6 nat v4v6 source 192.168.0.3 2003:12::3
ipv6 nat v6v4 source 2004:def::2 172.16.0.2
ipv6 nat v6v4 source 2004:def::3 172.16.0.3
ipv6 nat pr
ipv6 nat prefix 2003:12::/96

```

## Подвійний стек

```

R1
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#interface GigabitEthernet0/0
Router(config-if)#ip address 10.10.1.97 255.255.255.224
Router(config-if)#ipv6 address FE80::1 link-local
Router(config-if)# ipv6 address 2001:DB8:1:1::1/64
Router(config-if)#ex

```

```

Router(config)#interface Serial0/0/1
Router(config-if)#ip address 10.10.1.6 255.255.255.252
Router(config-if)# ipv6 address FE80::1 link-local
Router(config-if)# ipv6 address 2001:DB8:1:2::2/64
Router(config-if)#ex
R1(config)#router eigrp 1
R1(config-router)# network 10.10.1.4 0.0.0.3
R1(config-router)# network 10.10.1.96 0.0.0.31
R1(config)#ipv6 unicast-routing
R1(config)#int gi 0/0
R1(config-if)#ipv6 eigrp 1

R1(config-if)#int se 0/0/1
R1(config-if)#ipv6 eigrp 1

```

R2

```

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#host
Router(config)#hostname R2
R2(config)#interface Serial0/0/0
R2(config-if)# ip address 10.10.1.5 255.255.255.252
R2(config-if)# ipv6 address FE80::2 link-local
R2(config-if)# ipv6 address 2001:DB8:1:2::1/64
R2(config-if)#ex
R2(config)#interface Serial0/0/1
R2(config-if)# ip address 10.10.1.9 255.255.255.252
R2(config-if)# ipv6 address FE80::2 link-local
R2(config-if)# ipv6 address 2001:DB8:1:3::1/64
R2(config)#router eigrp 1
R2(config-router)# network 10.10.1.8 0.0.0.3
R2(config-router)# network 10.10.1.4 0.0.0.3

```

R3

```

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hos
Router(config)#hostname R3
R3(config)#interface GigabitEthernet0/0
R3(config-if)#ip address 10.10.1.17 255.255.255.240
R3(config-if)#ipv6 address FE80::3 link-local
R3(config-if)# ipv6 address 2001:DB8:1:4::1/64
R3(config-if)#
R3(config-if)#ex
R3(config)#interface Serial0/0/1
R3(config-if)#ip address 10.10.1.10 255.255.255.252
R3(config-if)# ipv6 address FE80::3 link-local
R3(config-if)# ipv6 address 2001:DB8:1:3::2/64
R3(config)#router eigrp 1
R3(config-router)# network 10.10.1.8 0.0.0.3
R3(config-router)# network 10.10.1.16 0.0.0.15
R3(config)#ipv6 router eigrp 1
R3(config-rtr)#no sh

```

## Тунелювання

R1

```
Router#conf t
Router(config)#ipv6 unicast-routing
Router(config)#int fa0/0
Router(config-if)#no shutdown
Router(config-if)#ipv6 add 2000:1:1:1:1:1:1:1:1112/112
Router(config-if)#ipv6 rip 6bone en
Router(config-if)#ipv6 rip 6bone enable
Router(config-if)#ex
```

R2

```
Router>en
Router#conf t
Router(config)#ipv6 unicast-routing
Router(config)#int serial 0/3/0
Router(config-if)#ip add 192.23.1.2 255.255.255.0
Router(config-if)#no sh
Router(config-if)#int fa0/0
Router(config-if)#ipv6 add 2000:1:1:1:1:1:1:1:1111/112
Router(config-if)#no sh
Router(config-if)#ipv6 rip 6bone enable
Router(config-if)#ex
Router(config)#int tunnel0
Router(config-if)#ipv6 add 3000::1/112
Router(config-if)#ipv6 rip 6bone enable
Router(config-if)#tunnel source serial 0/3/0
Router(config-if)#tunnel destination 192.34.1.4
Router(config-if)#tunnel mode ipv6ip
Router(config-if)#ex
Router(config)#router ospf 1
Router(config-router)#network 192.23.1.0 0.0.0.255 area 0
Router(config-router)#ex
```

R3

```
Router>en
Router#conf t
Router(config)#int serial 0/3/0
Router(config-if)#ip add 192.23.1.3 255.255.255.0
Router(config-if)#no sh
Router(config-if)#ex
Router(config)#int s
Router(config)#int serial 0/3/1
Router(config-if)#int serial 0/3/1
Router(config-if)#ip add 192.34.1.3 255.255.255.0
Router(config-if)#no sh
Router(config-if)#ex
Router(config)#router ospf 1
Router(config-router)#network 192.23.1.0 0.0.0.255 area 0
Router(config-router)#network 192.34.1.0 0.0.0.255 area 0
Router(config-router)#ex
```

R4

```
Router>en
Router#conf t
```

```
Router(config)#ipv6 unicast-routing
Router(config)#int serial 0/3/1
Router(config-if)#ip add 192.34.1.4 255.255.255.0
Router(config-if)#no sh
Router(config-if)#ex
Router(config)#int fa0/1
Router(config-if)#ipv6 add 4000:1:1:1:1:1:1111/112
Router(config-if)#ipv6 rip 6bone enable
Router(config-if)#no sh
Router(config-if)#ex
Router(config)#int tunnel0
Router(config-if)#ipv6 add 3000::2/122
Router(config-if)#ipv6 rip 6bone enable
Router(config-if)#tunnel source serial 0/3/1
Router(config-if)#tunnel destination 192.23.1.2
Router(config-if)#tunnel mode ipv6ip
Router(config-if)#ex
Router(config)#router ospf 1
Router(config-router)#network 192.34.1.0 0.0.0.255 area 0
Router(config-router)#ex
R5
Router>en
Router#conf t
Router(config)#ipv6 unicast-routing
Router(config)#int fa 0/1
Router(config-if)#no sh
Router(config-if)#ipv6 add 4000:1:1:1:1:1:1112/112
Router(config-if)#ipv6 rip 6bone enable
Router(config-if)#ex
```