

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА
ДИЗАЙНУ
ФАКУЛЬТЕТ МЕХАТРОНІКИ ТА КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

Дипломна бакалаврська робота

на тему:
**«СЕРВЕРИ ТА ОСОБЛИВОСТІ ЇХ ЗАСТОСУВАННЯ В КОМП'ЮТЕРНИХ
МЕРЕЖАХ»**

Виконав студент 4 курсу, групи БІТск1-21
спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

Роман Мокрий

(прізвище та ініціали)

Керівник

Тетяна ДЕМКІВСЬКА

(прізвище та ініціали)

Рецензент

Віктор ЧУПРИНКА

(прізвище та ініціали)

КИЇВ – 2023

АНОТАЦІЯ

Мокрий Роман Олексійович. Сервери та особливості їх застосування в комп'ютерних мережах.

Дипломна бакалаврська робота за спеціальністю 122 — “Комп'ютерні науки” — Київський національний університет технологій та дизайну, Київ, 2023 рік.

У процесі виконання завдання підприємства з побудови мережі було проведено проектування та налаштування топології мережі відповідно до вимог. З метою якісного й комплексного тестування працездатності топології, систему було запрограмовано за допомогою бекенду, який працює на сервері та здійснює обробку запитів клієнтів і надає необхідних даних або послуг. Фронтенд виступає в ролі клієнта і взаємодіє з сервером, відправляючи й отримуючи запити та відповіді. Для реалізації цього використовуються транспортного протоколу передачі даних та сокети.

Створена топологія мережі призначена для етапного впровадження реальної мережі офісу підприємства.

Ключові слова: сервери, мережі, топологія, кластер.

ABSTRACT

Mokry Roman Oleksiyovych. Servers and features of their use in computer networks.

Bachelor's thesis in specialty 122 — "Computer science" — Kyiv National University of Technology and Design, Kyiv, 2023.

In the course of the enterprise's task of building a network, the design and configuration of the network topology was carried out in accordance with the requirements. For the purpose of high-quality and comprehensive testing of the functionality of the topology, a network component was developed that allows checking the server settings.

The created network topology is intended for phased implementation of the real network of the enterprise's office.

Keywords: servers, networks, topology, cluster.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Факультет мехатроніки та комп'ютерних технологій
Кафедра комп'ютерних наук
Спеціальність 122 Комп'ютерні науки
Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри
комп'ютерних наук та технологій
_____ Володимир

ЩЕРБАНЬ

(підпис)

«_____» _____ 2023р.

ЗАВДАННЯ

НА ДИПЛОМНУ БАКАЛАВРСЬКУ РОБОТУ

студента

Мокрому Роману Олексійовичу

1. Тема роботи: Сервери та особливості їх застосування у комп'ютерних мережах.
Науковий керівник роботи Демківська Тетяна Іванівна
затверджені наказом КНУТД від “ 08 ” 11__ 2022 року № 224-уч.
2. Строк подання студентом дипломної роботи 10.06.2023 р.
3. Вихідні дані до дипломної роботи (проекту) Розробки кафедри комп'ютерних наук, рекомендована література.
4. Зміст дипломної бакалаврської роботи: 1) Аналіз предметної області, 2) Проектування мережі, 3) Реалізація мережі.
5. Дата видачі завдання 15.03.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної бакалаврської роботи	Терміни виконання етапів		Примітка про виконання
1	Вступ	20.04.2023		
2	Розділ 1. Аналіз предметної області	25.04.2023		
3	Розділ 2. Проектування мережі	06.05.2023		
4	Розділ 3. Реалізація мережі	09.05.2023		
5	Висновки	10.05.2023		
6	Оформлення дипломної бакалаврської роботи (чистовий варіант)	15.05.2023		
7	Здача дипломної бакалаврської роботи на кафедру для рецензування (за 14 днів до захисту)	25.05.2023		
8	Перевірка дипломної бакалаврської роботи на наявність ознак плагіату (за 10 днів до захисту)			
9	Подання дипломної бакалаврської роботи на затвердження завідувачу кафедри (за 7 днів до захисту)			

Студент

(підпис)

Роман МОКРИЙ

Науковий керівник

(підпис)

Тетяна ДЕМКІВСЬКА

Рецензент

(підпис)

Віктор ЧУПРИНКА

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

FTP (File Transfer Protocol) — мережевий протокол, призначений для пересилання файлів між клієнтом та сервером в комп'ютерній мережі.

WINS (Windows Internet Name Service) — Служба інтернет імен Windows

TCP (Transmission Control Protocol) — протокол зі встановленням з'єднання та з гарантованою доставкою пакетів.

IP (Internet Protocol address) — унікальна адреса пристрою, підключеного до Інтернету.

DHCP (Dynamic Host Configuration Protocol) — протокол динамічної конфігурації вузла.

API (Application programming interface) – Програмний інтерфейс додатків.

HTTP (Hypertext Transfer Protocol) — Протокол передачі даних.

DNS (Domain Name System) — Система доменних імен.

MAC (Media Access Control) — Управління доступом до посередників.

IoT (Internet of Things) — Інтернет речей.

ПЗ - Програмне забезпечення.

Зміст

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1. Сервери та їх типи	10
1.2. Функціональні можливості серверів.....	17
1.3. Організація роботи серверів	25
РОЗДІЛ 2. ПРОЕКТУВАННЯ МЕРЕЖІ	28
2.1. Мета топології.....	28
2.2. Технічні вимоги до мережі	31
2.3. Мережеве проектування.....	32
2.4. Програма для реалізації.....	33
РОЗДІЛ 3. РЕАЛІЗАЦІЯ МЕРЕЖІ	37
3.1. Побудова топології.....	37
3.2. Налаштування сервісів	39
3.3. Програмування клієнт-серверного компонента для тестування..	42
3.4. Перевірка справності мережі	43
ВИСНОВКИ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49
ДОДАТОК А. ЛІСТИНГ КОДУ	52
ДОДАТОК Б. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	55

ВСТУП

Актуальність теми дослідження

Застосування комп'ютерних систем та мереж є надзвичайно ефективним інструментарієм у розвитку бізнесу, який спрямований на обробку величезних обсягів інформації, оперативне надання великої кількості послуг клієнтам. Нині ці функції можуть виконувати сервери. Саме дослідженню питань використання серверів і присвячена дана дипломна робота, що підтверджує її актуальність.

Мета та завдання

Метою є вивчення та аналіз застосування серверів, а також проектування та налаштування мережі офісу компанії, розташованого в трьох кімнатах.

У відповідності з метою, в роботі поставлені та вирішені наступні завдання:

- дана характеристика серверів та їх типи;
- розкриті функціональні можливості серверів;
- розглянута організація роботи серверів;
- згідно вимог до побудови топології офісу компанії, зпроектовано розміщення та обрано мережеве обладнання;
- побудована топологія, проведено налаштування маршрутизаторів, серверів і IoT пристроїв, а також перевірена працездатність мережі офісу в системі Cisco Packet Tracer.

Предмет дослідження

Предметом даного дослідження є особливості застосування серверів в комп'ютерних мережах. Дослідження буде включати аналіз ролі серверів у забезпеченні централізованого управління, обробки та зберігання даних.

Об'єкт дослідження

Об'єктом даного дослідження є сервери та їх роль у комп'ютерних мережах. Дослідження буде спрямоване на розгляд різних типів серверів, їх функціональних можливостей та використання в комп'ютерних мережах.

Методи розробки

Для вирішення практичного завдання було використано мову програмування Python та інструменту симуляції мереж Cisco Packet Tracer.

Структура, зміст та обсяг роботи

Робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатку. Загальний обсяг роботи – 51 сторінка друкованого тексту, включаючи 11 рисунків та 30 джерел.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Сервери та їх типи

В будь-якій мережі вузли мають можливість отримувати або надавати ресурси. Один з головних видів вузлів - це сервери. Сервер - це комп'ютер (рис. 1.1), який надає інформаційні, програмні та технічні ресурси користувачам інших комп'ютерів в мережі.

У порівнянні з звичайними комп'ютерами, сервери зазвичай мають значно більшу потужність, зокрема потужні процесори, великий обсяг оперативної пам'яті, великий об'єм жорстких дисків, хорошу систему охолодження та інше обладнання, необхідне для цілодобової роботи. Отже, сервер можна назвати потужним комп'ютером.



Рисунок 1.1. Приклад надання сервером послуг користувачу

Важливо розрізняти обладнання для серверів та звичайних комп'ютерів. Устаткування для серверів зазвичай має високу потужність та відмовостійкість, оскільки призначене для безперебійної роботи протягом довгого періоду часу. Це обладнання зазвичай коштує дорожче.

Більшість серверів має можливість підключення або відключення додаткового обладнання, масштабування та автоматизації роботи, що забезпечує максимальну безпеку інформаційних ресурсів.

Сервери можуть бути розміщені в спеціальних приміщеннях, таких як серверні або дата-центри, у стандартних шафах. Також існують сервери

менших розмірів, які можуть бути встановлені в звичайних кімнатах, наприклад, в офісах. Сервери можуть бути виконані у форм-факторі підлогових, настільних, в стійку, стельових та інших типів.

Кожний сервер складається з:

- материнської плати;
- процесора;
- оперативної пам'яті;
- жорсткого диску.

Основні типи серверів.

Сервери мережевої інфраструктури можна розділити на кілька типів, які забезпечують роботу в мережі TCP/IP. Серед них DHCP-, DNS- та WINS-сервери. Починаючи налаштування роботи великої мережі, зазвичай спочатку налаштовують саме ці сервери:

1. DHCP-сервери відповідають за видання параметрів клієнту, таких як унікальна IP-адреса та маска підмережі. Клієнти можуть також отримати від DHCP-сервера додаткові параметри, такі як адреси DNS- та WINS-серверів, адресу основного шлюзу, назву домену та інші;

2. DNS-сервери забезпечують перетворення імен вузлів на відповідні їм IP-адреси і є дуже важливими в мережах. Повне доменне ім'я машини може бути розбите на дві частини - ім'я області-домена та власне ім'я машини, і зазвичай використовується для ідентифікації машин в мережах Microsoft;

3. WINS-сервери - це служба, яка використовується для розподілу імен комп'ютерів та їх відповідних IP-адрес в мережах Windows. Вони є альтернативою DNS-серверам, тому що дозволяють користувачам звертатися до комп'ютерів за їх іменами, а не за IP-адресами.

Одним з важливих аспектів управління доменами є відповідальність за порядок в них. Зазвичай таку відповідальність бере на себе комп'ютер, на якому працюють користувачі-адміністратори, і вони стежать за тим, щоб у мережі не виникало проблем з однаковими IP-адресами різних машин.

Служба імен DNS (Domain Name System) — це розподілена база даних доволі простої структури. Для початкового знайомства можна вважати, що це кілька таблиць, у яких записано:

- яку IP-адресу має машина з певним іменем;
- яке ім'я має машина з визначеною адресою;
- що це за комп'ютер і яка операційна система встановлена на ньому;
- куди потрібно направляти електронну пошту для користувачів цієї машини;
- які псевдоніми є у даної машини.

Для прикладу, якщо користувач з комп'ютера `myscomputer.example.com` намагається надіслати електронний лист на адресу `user@example.net`, то системний адміністратор встановлює на комп'ютері протоколи TCP/IP та налаштовує IP-адресу ближчого сервера імен. Далі поштова програма на комп'ютері `myscomputer.example.com` надсилає запит до ближчого сервера імен: "Куди надіслати листа для `example.net`?" Якщо ближчий сервер не може надати відповідь, то він передає запит до більш старшого сервера імен. У кінцевому підсумку стає зрозумілим, що усі листи для `example.net` мають бути надіслані на сервери `mail.example.net` або `relay.example.com`. Разом з цією відповіддю надсилається адреса цих серверів. Поштова програма на комп'ютері `myscomputer.example.com` зв'язується з сервером `mail.example.net` (використовуючи IP-адресу, а не ім'я) та передає листа. Ці переговори та передача листа, як правило, відбувається протягом кількох секунд і користувач не помічає цього. Якщо сервер `mail.example.net` недоступний, то лист чекає на пересилку на сервері `relay.example.com`, поки сервер `mail.example.net` знову стане доступним.

Це типова поведінка програм, що працюють в Інтернеті. Зазвичай, поштові програми створюють багато запитів до DNS-сервера, і ці запити можуть бути досить складними. У більшості випадків, користувачі цих програм хочуть знати лише IP-адресу сервера з відповідним іменем. Очевидно,

що всередині цієї системи імен існує багато правил та тонкощів, які можна дізнатися з описів стандартів Інтернету або спеціальних книг.

Українська компанія "Хостмастер" разом з ICANN запровадила локальний кореневий DNS-сервер, що містить інформацію про домени верхнього рівня. Цей сервер є "дзеркалом" одного з 13 глобальних корневих серверів ICANN, який відомий як "L-root".

1. Сервери WINS зберігають в мережі імена комп'ютерів та відповідні їм IP-адреси за допомогою протоколу NetBIOS, а потім перетворюють ці імена на IP-адреси за запитом клієнтів WINS. Назва WINS правильно перекладається як "Служба мережевих імен Windows"; ця служба була розроблена для підтримки роботи додатків, що використовують протокол TCP/IP в мережах, що використовують NetBIOS. Вона все ще використовується, щоб забезпечити коректну роботу застарілих операційних систем, таких як Windows 9x або Windows NT, в мережах.

Комп'ютери, щоб взаємодіяти між собою, використовують числові IP-адреси. Однак людині їх важко запам'ятати, тому у мережах зазвичай використовуються словесні імена комп'ютерів. Проте, багато додатків дозволяють працювати безпосередньо з IP-адресами, іноді це навіть є зручнішим способом для перевірки працездатності додатку, особливо якщо системи розпізнавання імен в даній мережі не працюють.

У взаємодії комп'ютерів використовуються імена вузлів або NetBIOS-імена. Імена вузлів складаються з комбінацій букв, цифр та знаку "дефіс", розділених крапками і можуть бути використані як в Інтернеті (наприклад, `www.microsoft.com`), так і в локальній мережі (наприклад, `server1.domain.local`). Натомість NetBIOS-імена - це "власні" імена комп'ютерів, які містять не більше 15 будь-яких символів, за винятком точок (наприклад, `SERVER1`). Люди часто використовують імена вузлів для зручності, адже вони легше запам'ятати, ніж числові IP-адреси. Однак, в деяких випадках можна застосовувати і безпосередньо IP-адреси, особливо якщо системи розпізнавання імен не працюють.

2. Сервери файлів (файл-сервери) використовуються для зберігання великих обсягів даних та надання доступу до них користувачам. Файл-сервер може обслуговувати сотні і навіть тисячі користувачів одночасно. Щоб забезпечити збереження інформації, файл-сервери зазвичай оснащені відмовостійкими масивами жорстких дисків та системами резервного копіювання на магнітну стрічку або інші носії.

3. Сервери файлів (файл-сервери) використовуються для зберігання великих обсягів даних та надання доступу до них користувачам. Файл-сервер може обслуговувати сотні і навіть тисячі користувачів одночасно. Щоб забезпечити збереження інформації, файл-сервери зазвичай оснащені відмовостійкими масивами жорстких дисків та системами резервного копіювання на магнітну стрічку або інші носії.

4. Факс-сервери, які забезпечують відправку факсів від клієнтів, також відповідають за отримання та доставку факсів користувачам, що є однією з їхніх функцій.

5. Найпоширенішими типами серверів є файл-сервери та сервери друку.

6. Сервери додатків служать для обслуговування запитів користувачів на вибірку або обробку інформації, і часто їх об'єднують з серверами баз даних. Важливо зазначити, що ці сервери можуть обслуговувати багато користувачів одночасно, і виконання запитів на спеціалізованому багатопроцесорному сервері здійснюється значно швидше, ніж на комп'ютерах користувачів.

7. Сервери віддаленого доступу та сервери VPN забезпечують можливість здійснення віддаленого підключення до локальної мережі підприємства, офісу або навчального закладу через модем або Інтернет. Це дозволяє користувачам працювати з ресурсами мережі з будь-якого місця, де є доступ до Інтернету, наприклад, з дому або Інтернет-кафе.

8. Термінальні сервери дозволяють адміністраторам та користувачам віддалено працювати з додатками, які встановлені на сервері, за допомогою спеціальних програм - термінальних клієнтів. Ці програми дозволяють

адміністраторам отримувати доступ до консолі сервера, навіть якщо вони перебувають далеко від локальної мережі, та повністю керувати ним.

9. Брандмауери (міжмережеві екрани) використовуються для захисту корпоративних серверів від проникнення або атаки зловмисників при підключенні до Інтернету. Проксі-сервери (сервери-посередники) виконують функції контролю доступу користувачів до Інтернету та кешування часто запитуваних веб-сторінок. Оскільки обидва цих сервери призначені для установки на комп'ютер, що зв'язує локальну мережу з Інтернетом, їх можна об'єднати у єдину програмно-апаратну систему.

10. Сервери електронної пошти (поштові сервери, mail-сервери) обслуговують поштові скриньки користувачів в організації, забезпечують підключення до них поштових клієнтів та оброблюють вхідні і вихідні повідомлення. Вони також можуть використовуватись для ведення адресних книг, загальних папок та систем електронного документообігу.

11. Веб- та FTP-сервери дають змогу зовнішнім та внутрішнім користувачам отримати доступ до веб- та FTP-ресурсів, розміщених в даній мережі. Веб-сервери служать для виконання HTTP-запитів та передачі веб-сторінок, а FTP-сервери забезпечують доступ до файлів, що знаходяться на сервері.

12. Контролери домену використовуються в мережах Microsoft для управління службами Активного каталогу (Active Directory) та зберігання бази даних з інформацією про зареєстрованих у домені користувачів, комп'ютери, групи та ресурси. Це дозволяє адміністраторам централізовано керувати всіма мережевими об'єктами та ресурсами, а користувачам - входити в мережу з будь-якого комп'ютера, що належить до домену, та отримувати доступ до ресурсів без введення імені та пароля.

Залежно від рівня володіння користувача комп'ютером та його потреб, він обирає підходящий для себе комп'ютер. Наприклад, якщо користувач планує використовувати комп'ютер у офісній роботі, йому потрібен комп'ютер з оптимальним співвідношенням ціни та функціональності, який зможе

працювати з офісними додатками. Якщо комп'ютер буде використовуватись у сімейному оточенні, де дорослі дивляться фільми, а діти грають у комп'ютерні ігри, то важливо обрати потужний комп'ютер, здатний відтворювати різні медіа-дані. Варто зазначити, що будь-який персональний комп'ютер має одного користувача.

Завдяки наявності сервера підприємство може успішно виконувати більш складні завдання, які не можна виконати за допомогою звичайного комп'ютера.

Ефективність роботи всієї мережі підприємства та його здатність досягати поставлених цілей залежить від якості сервера, який використовується. Вибір сервера повинен відповідати завданням, що стоять перед компанією.

Однією з найважливіших характеристик сервера є його продуктивність, яка залежить від кількох ключових параметрів:

- тип і продуктивність процесорів
- обсяг і тип оперативної пам'яті;
- продуктивність дискової підсистеми.

Наприклад, чим більше процесорів міститься в сервері і чим більше ядер у кожному з них, тим більше потужності надається всій мережі. При виборі конфігурації сервера, важливо забезпечити можливість розширення його функціоналу в майбутньому, якщо з'явиться потреба. Для цього необхідно планувати наявність додаткових процесорів, пам'яті та інших пристроїв, які можуть бути сумісні з наявним обладнанням.

Ще одним важливим аспектом сервера є його керованість. Це означає, що сервер повинен мати функції віддаленого моніторингу та діагностики. Доцільно мати змогу керувати сервером здалеку, включати його, перезавантажувати, діагностувати та виправляти проблеми навіть у разі вимкнення (якщо він підключений до електромережі).

Продуктивність та керованість - це дві важливі характеристики, які мають значний вплив на надійність сервера. Надійність не обмежується

фізичною стійкістю і якістю складання сервера, а також включає програмну надійність, яка забезпечує стабільну роботу всіх програм.

Ще однією важливою характеристикою сервера є його масштабованість. Ця характеристика означає, що сервер може збільшувати свою продуктивність залежно від збільшення робочого навантаження, і при цьому зберігати високу надійність та відмовостійкість. Така можливість дозволяє забезпечувати безперебійну роботу мережі навіть при зростанні обсягу обробки даних та збільшенні кількості користувачів. Тому при виборі сервера варто звернути увагу на його масштабованість та здатність пристосовуватися до змін у робочому середовищі.

1.2. Функціональні можливості серверів

– Існує велика кількість функцій у серверів, основними, із яких є функції:

- пошукових систем;
- зберігання та передачі файлів;
- електронної пошти;
- організації та доступу до WEB – сторінки;
- бази даних;
- відеоспостереження;
- шлюзу Інтернету;
- віддаленого доступу.

Функції пошукових систем виконує сервер, який спеціалізується на пошуку інформації. Будь-яку інформацію можна знайти в Інтернеті, в цьому допомагають пошукові системи. Ось найпопулярніші з них:

- Google;
- Bing;
- Baidu;
- Yahoo!.

Пошукова система (ПС) представляє собою складний комплекс програм, які використовують спеціальні алгоритми для аналізу вмісту веб-сайтів у всьому Інтернеті. Існують спеціалізовані служби, які називаються пошуковими серверами, що дозволяють здійснювати пошук в Інтернеті. Вони представлені веб-сайтами, де можна ввести ключові слова в спеціальне поле і отримати багато посилань на ресурси з необхідною інформацією. Наприклад, при запиті "домашні тварини" пошуковий сервер "Google" надає понад 16 000 000 посилань на веб-сторінки менше ніж за пів секунди, які містять відповідну інформацію. Однак не завжди все так просто, оскільки при відвідуванні деяких знайдених веб-сторінок можна помітити, що не вистачає або зовсім відсутня шукана інформація.

ПС постійно сканують Інтернет для оновлення своїх баз даних. Зазвичай цей процес відбувається без участі людей. ПС складається з трьох основних компонентів: робота, індекс та обробки запитів. Робот (Crawler, Bot, Robot) - це програма, яка відвідує веб-сторінки, читає їх вміст (індексує) і переходить за посиланнями, які вона знаходить на цих сторінках. Робот періодично повертається (наприклад, щомісяця) і знову індексує сторінки. Все, що робот знаходить і прочитує, потрапляє до індексу пошукової системи.

Індекси представляють собою велике сховище інформації, де зберігаються копії текстового вмісту всіх відвіданих та проіндексованих веб-сторінок. Обробник запитів - це програма, яка, відповідно до поданого запиту, перебирає індекси пошукової системи, шукаючи інформацію, що цікавить користувача, та видає знайдені документи в порядку зменшення їх релевантності (відповідності запиту). Щоб зрозуміти, як пошуковий сервер може повернути таку велику кількість сторінок на введений запит менше, ніж за пів секунди, потрібно знати принципи роботи пошукових серверів. Одним з принципів є використання спеціальної програми-павука, яка завантажує вміст веб-сайтів. Задача цього павука полягає у передачі вмісту веб-сторінок іншій програмі-мандрівному павуку. Мандрівний павук виявляє посилання на інші веб-сторінки у завантаженому контенті і передає їх назад павуку. Цей цикл

повторюється безперервно. Проте робота не закінчується лише на цьому. Далі вступає в дію програма-індексатор, яка застосовує певні правила для аналізу веб-сторінок, які були отримані павуками, і створює складну базу даних пошукового сервера. Ця база даних потім використовується для надання результатів пошуку після обробки введеного запиту.

Для більш детального опису принципу роботи індексатора, слід зазначити, що від нього залежить те, що включатиметься у результати пошуку. В більшості сучасних пошукових серверів основою є індекс цитування, який обчислюється індексатором шляхом аналізу посилань на поточну веб-сторінку з інших веб-сторінок Інтернету. Чим більше посилань вказують на цю веб-сторінку, тим вище буде індекс цитування, що сприяє показу цієї сторінки в результатах пошуку. Крім індексу цитування, індексатор також враховує індекс цитування веб-сторінок, що посилаються на аналізовану сторінку. Це означає, що якщо високоякісні та авторитетні веб-сторінки посилаються на певну сторінку, її індекс цитування збільшується, що також впливає на її рейтинг у результатах пошуку. Крім індексу цитування, пошукові системи враховують інші параметри, такі як:

1. Після обробки запиту користувача за справу береться система видачі результатів. Шляхом аналізу та порівняння різних показників, система визначає ступінь відповідності вмісту веб-сторінок умовам запиту. Чим більше ця ступінь, тим вище веб-сторінка буде розміщена у списку знайдених результатів. Однак, цей метод має очевидний недолік: вміст веб-сторінок може змінюватися з часом, а "павук" не завжди зможе встигати знаходити і обробляти оновлення, що може призводити до неточностей у результатах пошуку.

2. Час, необхідний пошуковому серверу для проіндексування всього Інтернету, залежить від різних факторів, таких як алгоритми обробки інформації та апаратне забезпечення. Зазвичай цей процес займає від декількох днів до кількох неділь. Тому недавно створені веб-сайти можуть бути відсутніми у результатах пошуку. Розробники пошукових серверів

активно працюють над вирішенням цієї проблеми, проте з різними успіхами. Наприклад, багато сучасних пошукових серверів надають сервіс пошуку у новинних стрічках, які оновлюються кожні декілька хвилин, що дозволяє їх більш часту індексацію пошуковими серверами.

Спеціалізований комп'ютер виконує функцію **зберігання та передачі даних файлів**. Його основна роль полягає у зберіганні значної кількості файлів. Користувачі мають доступ до цього комп'ютера зі своїх пристроїв, таких як комп'ютери, ноутбуки або мобільні пристрої, через локальну мережу або Інтернет. Вони можуть завантажувати файли з сервера, які їм потрібні, а також завантажувати файли на сервер для зберігання. Використання цього рішення дозволяє користувачам уникнути необхідності зберігання файлів локально на їх пристроях, що допомагає заощадити пам'ять на накопичувачах.

Основна мета цього файлового сервера - збереження файлів, до яких користувачі мають спільний віддалений доступ. В організаційних умовах це можуть бути документи, а в домашніх умовах - музика, фільми, фотографії, дистрибутиви програм та багато іншого.

Основні цілі встановлення файлового сервера - економія пам'яті на дисках користувачів та полегшення роботи з інформацією.

При передачі файлів використовується протокол передачі файлів FTP, тому файлові сервери ще називають FTP-серверами.

Відповідно до протоколу FTP два користувачі не можуть обмінятися файлами напряму; обмін здійснюється тільки через FTP-сервер.

При зверненні клієнта до сервера, сервер вимагає введення імені користувача та пароля. На більшості серверів ця процедура є формальною, оскільки достатньо ввести слово "anonymous" (анонімний) як ім'я без вказування пароля або повідомити свою електронну пошту. Проте, на деяких серверах доступ до папок є платним, тобто користувач отримує ім'я та пароль за певну плату, які діють на сервері протягом певного періоду часу. Більшість загальнодоступних файлових архівів зберігаються в каталозі /pub, а файли, які завантажує клієнт, зберігаються в каталозі /incoming.

Для роботи з FTP-серверами можна використовувати веб-браузери, проте вони мають обмежені можливості та використовуються рідко, коли дії з архівами виконуються нечасто.

Існують спеціальні програми, що призначені для роботи з такими серверами. Їх називають FTP-клієнтами.

- FTP Explorer;
- AutoFTP.

Програма, встановлена на поштовому сервері, виконує функцію електронної пошти. Поштовий провайдер - це сервіс, який має багато поштових серверів і надає можливість створити поштову скриньку.

Для підключення до поштового сервера Інтернету існують різні можливості:

1. За допомогою локальної обчислювальної мережі (ЛОМ).
2. Використовуючи звичайний модем і засоби віддаленого доступу.
3. За допомогою іншої програми зв'язку та модема.

За допомогою ЛОМ: Для підключення до поштового сервера через локальну мережу потрібна наявність Інтернет-виходу через Шлюз - пристрій, який забезпечує комунікацію між різними поштовими системами та обмін повідомленнями. Поштовий сервер або Інтернет-сервер повинні бути розташовані в межах локальної мережі. У цьому випадку, поштовий сервер автоматично доставляє нову пошту з певною періодичністю. Також може бути налаштоване віддалене з'єднання, яке використовується у випадку недоступності локальної мережі.

За допомогою модема: Провайдер Інтернет-послуг повинен надати необхідні дані про використовувані телефонні номери, налаштування модема та мережеві протоколи. Постачальник Інтернет-послуг є компанією, яка забезпечує доступ до Інтернету для використання електронної пошти та інших Інтернет-сервісів. Деякі постачальники є міжнародними та пропонують доступ у різних країнах, в той час як інші обмежені певним регіоном.

За допомогою іншої програми зв'язку: Для підключення за допомогою іншої програми зв'язку та вибору часу доставки вхідних та вихідних повідомлень можна встановлювати з'єднання з сервером вручну. Цей варіант має сенс, якщо швидкість з'єднання за допомогою віддаленого доступу невелика, а ви не хочете чекати автоматичного підключення сервера.

Функцію організації та доступу до веб-сторінки виконує комп'ютер, який зберігає файли сайту, такі як HTML-документи, CSS-стилі, JavaScript-файли, зображення та інші ресурси, і надає їх на пристрій кінцевого користувача через веб-браузер. Цей комп'ютер підключений до Інтернету і може бути доступним за допомогою доменного імені.

Коли веб-браузер потребує файл, який знаходиться на веб-сервері, він надсилає запит за допомогою протоколу HTTP. Коли запит досягає веб-сервера, сервер HTTP отримує запит, знаходить запрошений документ (якщо такий існує, інакше повертає помилку 404) і надсилає його назад за допомогою протоколу HTTP.

Основним завданням веб-сервера є обробка запиту від клієнта, розуміння, до якого файлу відноситься запит, обробка цього файлу та надання відповіді клієнту.

Варто зрозуміти, що HTTP-запити можна виконувати не лише з веб-браузера, а й з командного рядка або за допомогою спеціальних серверних додатків. Крім того, веб-сервер має такі завдання, як створення журналу помилок і відстеження запитів до файлів.

Наступним кроком веб-сервера є аутентифікація та авторизація користувачів. Веб-сервер може визначити, хто звертається до нього і чи має цей користувач доступ до файлів. Якщо доступ заборонено, веб-сервер блокує доступ до цих файлів. Веб-сервер також містить налаштування щодо обробки файлів та керування папками. Усі ці правила встановлюються на веб-сервері.

Оскільки веб-сервер є програмою, його можуть розробляти різні компанії за своїми алгоритмами. На сьогоднішній день найпопулярнішими

веб-серверами є Apache, IIS та Nginx. Кожен з них має свої особливості та способи роботи.

Функцію бази даних виконує спеціалізоване програмне забезпечення, яке розміщене на сервері. Крім зберігання, сервер баз даних також виконує завдання з архівації. За допомогою клієнт-серверної архітектури, сервер баз даних здатен обробляти та аналізувати дані. Важливо зазначити, що сервер баз даних не залежить від конкретної архітектури бази даних, тобто він може підтримувати реляційні і нереляційні бази даних, а також бази даних на плоских файлах.

Функція відеоспостереження полягає у записі, зберіганні та передачі контенту з камер (рис. 1.2). Для організації віддаленого відеоспостереження, відеосервер може передавати відео через Інтернет. Записаний контент можна переглядати через систему управління VMS (Video Management System) або віддалено за допомогою звичайного інтернет-браузера. Для ефективної роботи, сервер відеоспостереження може починати запис по спрацьовуванню датчика руху і надсилати оповіщення про події, наприклад, вторгнення в закриту зону, за допомогою SMS або електронної пошти.

Відеосервери дозволяють швидко інтегрувати існуючі аналогові системи відеоспостереження. Спочатку відеосервери були створені як доповнення до вже існуючих аналогових систем відеоспостереження, але незабаром вони розширили можливості цих систем.

Головна функція відеосервера полягає в тому, щоб користувачі системи відеоспостереження могли переглядати відеозображення з аналогових камер у цифровому форматі.

Відеосервери працюють як з аналоговими, так і з цифровими камерами відеоспостереження. Вони також дозволяють одночасний перегляд відео з кількома відеокамерами для кількох користувачів.

Сучасні IP-відеосервери основне завдання полягає в оцифруванні аналогових відеосигналів з камер і передачі цифрових зображень по локальній мережі. IP-відеосервери перетворюють старі аналогові системи

відеоспостереження на сучасні системи мережевого відеоспостереження. Раніше вважалося, що користувач може переглядати записані зображення з камер лише на одному центральному комп'ютері, поблизу якого потрібно було перебувати.

З появою відеосерверів це змінилося: зараз кілька користувачів з різних місць можуть одночасно переглядати в реальному часі відео з камер за допомогою свого власного комп'ютера або смартфона. Раніше ніхто навіть не уявляв собі, що можна одночасно переглядати записи з однієї аналогової камери декількома авторизованими користувачами.

Сучасні відеосервери дозволяють підключати різні типи камер без обмежень. Це охоплює як мініатюрні, так і приховані камери для відеоспостереження. Крім того, якщо використовувати поворотні камери спостереження, відеосервер дозволяє віддалено керувати їх положенням, нахилом об'єктиву, зумом та іншими параметрами.

Функцію віддаленого доступу виконує веб-сервер, до якого можна отримати доступ через Інтернет. Веб-сервер може належати самій компанії або орендуватися у хостинг-провайдера в хмарі або на віртуальних серверах. Інші комп'ютери можуть підключатися до нього за допомогою служб віддаленого доступу.

Найпоширеніший сценарій використання полягає в тому, що база даних розміщена на сервері, до якого звертається велика кількість користувачів з будь-якої точки світу, будь то вдома або в локальній корпоративній мережі. Доступ до бази даних здійснюється через Інтернет, і швидкість доступу залежить від швидкості підключення, а не від фізичного розташування сервера, що робить роботу з розміщеною там інформацією зручною незалежно від місця розташування сервера.

Сервер віддаленого доступу підтримує два режими роботи:

1. Віддалений доступ до мережі. Класичний випадок, описаних в прикладах вище - коли віддалені клієнти підключаються до сервера так, немов

він перебуває з ними в одній локальній мережі. Тут сервер працює так само, як і будь-який інший комп'ютер в корпоративній мережі.

2. Віддалене управління. Клієнти дистанційно керують роботою сервера (його ще називають сервером терміналів).

Стандартний спосіб з'єднання з віддаленим сервером - використання протоколу RDP (Remote Desktop Protocol). Компанія Microsoft внесла значний внесок у його популяризацію і рекомендує його для організації віддаленої роботи між сервером і користувачами, що підключаються з термінальних машин.

Інтернет-шлюз - це апаратне або програмне рішення, яке призначене для конвертації пакетів даних у різних фізичних середовищах передачі сигналу. Іншими словами, мережеві шлюзи здатні перетворити сигнали в мережах різних форматів або їх сегментах, виступаючи своєрідним "мостом" між, наприклад, локальною та глобальною мережами.

Мережеві шлюзи необхідні, оскільки існуючі пристрої для маршрутизації, такі як маршрутизатори та роутери, зазвичай функціонують у межах одного фізичного середовища передачі сигналу. Це призводить до обмеження їх діяльності рамками однієї мережі. Тому мережеві шлюзи використовуються для можливості конвертації сигналу з одного середовища в інше.

1.3. Організація роботи серверів

Якісна та безперебійна робота локальної мережі забезпечується за допомогою бездоганної роботи сервера. Але для його хорошого функціонування необхідні правильні настройки та кваліфіковане обслуговування.

В сучасному світі високих технологій неможливо уявити навіть саму маленьку компанію без сервера, а діяльність великої корпорації взагалі складно здійсненна без нього. При розширенні або ж розвитку будь-яка організація встановлює сервер. Його функції виходять від діяльності організації, масштабів і поставленої мети. За таким принципом визначається

рівень навантаження серверів. Важливим моментом є якісне та систематичне обслуговування серверів, адміністрування серверів вже після їх установки і налаштування. Адже такий супровід гарантує безперебійну та стабільну роботу сервера, оскільки навіть найменший збій може призвести до втрати величезної кількості грошових активів компанії. А своєчасне і швидке обслуговування забезпечить гарантії швидкої ліквідації неполадок та мінімізації втрат.

Процедура налагодження серверів - досить складне завдання, яка залежить від конкретного призначення. При запуску нового серверного обладнання потрібно вибрати спеціальне програмне забезпечення.

Якщо сервер розрахований на роботу в мережі інтернет, бажано встановлювати Unix платформи, але це залежить від уподобань адміністратора і поставлених завдань. Після того, як програмне забезпечення встановлено, можна приступати до основних налаштувань.

Кожен сервер налаштовується по-своєму. Але, в основному, дана процедура зводиться до налаштування:

- користувачів;
- прав доступу;
- установки і настройки основного ПЗ (сервісів);
- мережевих інтерфейсів;
- віддаленого доступу до сервера.

Для цього можна створювати суперкористувачів, які можуть заходити в систему під певним паролем і змінювати основні характеристики.

Також дуже важливо правильно налаштувати всі основні права на доступ, щоб не кожен користувач міг керувати файлами. Це дуже важливо для безпеки, а також дозволить зберегти всі дані.

Ця установка проводиться тільки відповідно до рекомендацій виробника того чи іншого ПЗ, а також фахівцями - адміністраторами, які мають досвід виконання таких робіт.

Адміністратор повинен добре розбиратися у всіх основних параметрах системи і конфігурувати її тільки під конкретні завдання.

Налаштування в загальному можна розділити на дві частини:

1. Підбір та налаштування апаратної частини: виходячи з потреб та вимог замовника відбувається підбір та узгодження оптимального апаратного забезпечення для побудови стабільної та надійної системи, яка буде в повній мірі виконувати поставлені завдання. При проектуванні та виборі материнської плати, процесора, оперативної пам'яті, потрібно враховувати також специфіку саме серверних систем, забезпечення системи резервного живлення, дублювання мережевих ресурсів, система резервного копіювання, система контролю стабільності роботи компонентів сервера;

2. Програмне налаштування сервера: після налаштування апаратної частини проводиться встановлення та налаштування потрібного програмного забезпечення яке буде виконувати завдання необхідні замовнику, а також для стабільної та безпечної роботи сервера. Встановлення операційної системи та допоміжного програмного забезпечення, тестування та перевірка функціоналу перед здачею в експлуатацію.

РОЗДІЛ 2. ПРОЕКТУВАННЯ МЕРЕЖІ

2.1. Мета топології

Основна мета топології полягає у створенні мережі офісу для компанії “ТОВ Сільпо-Фуд” кількістю до 30-ти чоловік.

У рамках практичної частини буде проведено дослідження використання серверів у комп'ютерних мережах. Для цього буде розроблена топологія, що має включати в себе:

- моделювання мережі;
- налаштування мережевих девайсів та серверів;
- програмування компонентів IoT.

Для початку дослідження, потрібно усвідомити значення топології.

Топологія мережі - це організація та структура вузлів і з'єднань у мережі, що охоплює як фізичні, так і логічні аспекти. Вузли зазвичай складаються з таких пристроїв, як комутатори, маршрутизатори та програмне забезпечення зі схожими функціональними можливостями. Для представлення топології мережі зазвичай використовують графіки.

Поняття мережевої топології описує, як влаштована мережа і взаємозв'язки між різними елементами, включаючи потік трафіку. Мережеві адміністратори використовують діаграми топології мережі для визначення оптимального розміщення вузлів і шляхів трафіку. Створивши чітко визначену і ретельно сплановану топологію мережі, організації можуть ефективно виявляти і усувати несправності, що призводить до підвищення ефективності передачі даних.

Мережеві топології можна розділити на два основних типи: фізичні та логічні топології. Фізична топологія (наведена на рис. 2.1) відноситься до фізичного розташування і взаємозв'язку пристроїв в мережі. Сюди входять власне кабелі, дроти та з'єднання, що використовуються для встановлення зв'язку. Прикладами фізичних топологій є зірка, шина, кільце і комірчаста топологія.

З іншого боку, логічна топологія (наведена на рис. 2.2) фокусується на концептуальному дизайні та потоці даних в мережі. Вона описує, як вузли спілкуються та обмінюються інформацією. Логічні топології можуть бути представлені різними патернами, такими як зірка, кільце, шина і дерево. Логічні топології не залежать від фізичної структури і можуть включати віртуальні та хмарні ресурси.

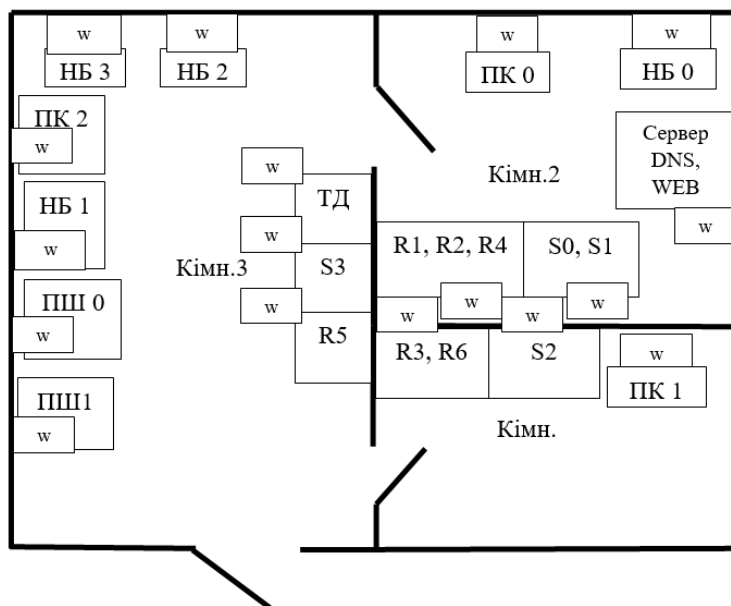


Рис. 2.1. Приклад фізичної топології мережі

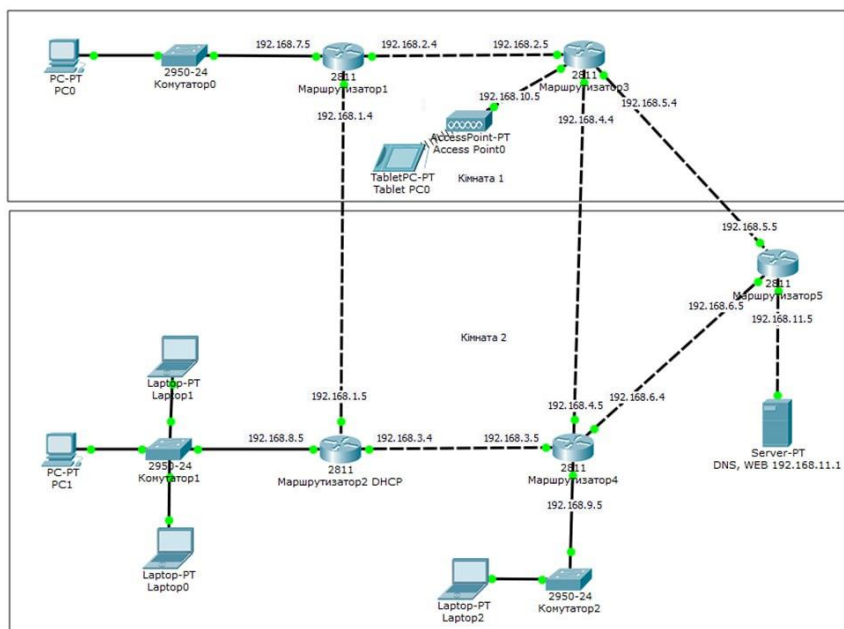


Рис. 2.2. Приклад логічної топології мережі

2.1.1. Важливість топології

Важливість топології мережі неможливо переоцінити. Вона безпосередньо впливає на функціональність і продуктивність мережі. Ретельно плануючи та впроваджуючи правильну топологію мережі, організації можуть досягти підвищення енергоефективності, пришвидшити передачу даних та підвищити загальну продуктивність.

Топологія мережі дає уявлення про структуру мережі, дозволяючи адміністраторам зрозуміти її ключові компоненти та їх взаємозв'язки. Ці знання уможливають ефективне управління продуктивністю, незалежно від розміру та складності мережі.

Існує кілька переваг і причин у використанні топологій мереж.

Економічна ефективність. Чітко визначена топологія мережі допомагає виявити неефективні операції та непотрібні компоненти, що призводить до економії коштів. Це підвищує продуктивність праці, забезпечуючи вищу якість продукції за менший час, що в кінцевому підсумку призводить до збільшення прибутку і масштабованості.

Поглиблене розуміння мережевих концепцій. Топологія мережі полегшує вивчення та розуміння принципів побудови мереж. Вона спрощує розуміння того, як функціонують комп'ютерні мережі, ІТ-системи та мережі 5G. Топології допомагають у налаштуванні домашніх або бізнес-мереж, забезпечуючи чітке розуміння фундаментальних мережевих концепцій, таких як концентратори, маршрутизатори та трансляції.

Ефективна комунікація за допомогою топологічних схем. Добре розроблені топологічні діаграми слугують дороговказами для побудови функціональних систем. Вони покращують комунікацію між командами, працівниками та клієнтами, гарантуючи, що всі дотримуються однакового розуміння. Топологічні діаграми можна використовувати для створення інструкцій, посібників і шаблонів, підвищуючи продуктивність і підзвітність.

Легкість навчання. Мережеві топології слугують посібниками для введення в курс справи нових співробітників та інженерів. Вони надають

комплексне уявлення про мережеві з'єднання, сприяючи кращій співпраці та взаєморозумінню між членами команди. Інженери, незнайомі з мережею, можуть швидко зрозуміти її структуру і потенційні наслідки.

Діагностика стану мережі. Топології мережі допомагають виявляти та діагностувати мережеві проблеми. Аналізуючи топологію, можна визначити слабкі ланки і потенційні проблемні області, що дозволяє цілеспрямовано виправляти і оптимізувати їх. Розпізнавання топології мережі також дозволяє точно співвіднести помилки і ключові показники ефективності, що допомагає швидко і точно визначити першопричину проблем. Це, в свою чергу, покращує моніторинг, управління та оптимізацію мережі. діаграм дає цінну інформацію як про фізичну, так і про логічну структуру, допомагаючи в пошуку і вирішенні проблем.

2.2. Технічні вимоги до мережі

2.2.1. Відомості про компанію

ТОВ «Сільпо-Фуд» або Silpo.ua - компанія, що є однією з найбільших національних мереж продовольчих супермаркетів України. Вона займається продажем продовольчих товарів, продуктів власного виробництва та імпорту.

Одна з важливих діяльностей компанії, це велика пропускна спроможність для обробки та аналізу даних. Це є необхідним для ефективної роботи з великими обсягами інформації та забезпечення швидкого доступу до неї.

Офіс компанії складається з **4** основних кімнат, розташованих на одному поверсі:

1. кімнати відпочинку;
2. головної робочої кімнати;
3. серверної 1;
4. серверної 2.

2.2.2. Завдання

Побудувати ефективну топологію мережі, з практичним використанням **груп серверів.**

Наступні технології мають бути застосовані:

- Серверні кластери;
- Клієнт-сервер;
- Wi-Fi;
- IoT.

Наступні сервіси мають бути зконфігуровані:

- DNS:
 - silpo.ua - головний сайт компанії;
 - iot.silpo.ua – “адмінка” для IoT-ів;
- HTTP;
- DHCP.

Наступні критерії мають бути виконані:

- мережа організації має представляти одну реальну мережу - єдиний маршрутизатор;
- Побудувати 2 серверних кластера;
- Контроль температури за рахунок IoT.

2.3. Мережеве проектування

Для того щоб задовільнити вимоги використаємо технологію серверних кластерів.

2.3.1. Серверна 1

Для вирішення проблеми великого навантаження створимо серверний кластер. З цього кластеру виділимо певну кількість серверів для налаштування таких сервісів як HTTP та DNS. Також в цьому кластері буде розміщений сервер для Бекенду – детальний опис програмування буде наведений в третьому розділі.

2.3.2. Серверна 2

Тут розмістимо другий серверний кластер для того щоб розподілити навантаження та не концентрувати весь трафік на першому кластері. Він може стати в нагоді для майбутніх розробок компанії.

2.3.3. Головна кімната для роботи

В головному робочому приміщенні розташуємо 13 робочих станцій, такі як персональні комп'ютери і ноутбуки. Вони під'єднані до головного комутатора і точки доступу Wi-Fi. Також в кімнаті знаходиться термостат який призначений для підтримання та регулювання температури в робочій кімнаті – відбувається це за допомогою під'єднаних до нього батареї та кондиціонера. Для друку документів в кімнаті присутній принтер, який підключений бездротовим способом для підвищення мобільності та зручності використання.

2.3.4. Кімната відпочинку

Для відпочинку під час перерв у співробітників є можливість скористуватися безкоштовним Wi-Fi. Також, для комфортного перебування співробітників у кімнаті відпочинку, розташований стельовий вентилятор, який влітку слугує для охолодження кімнати. Для підвищення комфорту і урізноманітнення дозвілля працівників встановленні інтерактивні планшети.

2.4. Програма для реалізації

Cisco Packet Tracer - це програмне забезпечення для моделювання топологій від компанії Cisco, що пропонує платформу для створення складних мережевих топологій та імітації абстрактних мережевих концепцій. Це забезпечує практичний досвід, близький до реальних комп'ютерних мереж.

Cisco Packet Tracer пропонує різноманітні функції, які покращують його функціональність та зручність використання. Нижче наведені ключові можливості Cisco Packet Tracer:

1. імітація мережевих пристроїв і протоколів. Cisco Packet Tracer дозволяє користувачам імітувати широкий спектр мережевих пристроїв, включаючи маршрутизатори, комутатори, сервери і кінцеві точки. Він також підтримує різні мережеві протоколи, такі як TCP/IP, OSPF, EIGRP, BGP та інші. Це дозволяє користувачам створювати реалістичні мережеві сценарії та експериментувати з різними конфігураціями.

2. підтримка декількох протоколів і технологій. На додаток до симуляції мережевих пристроїв і протоколів, Cisco Packet Tracer підтримує безліч технологій, таких як бездротові мережі, Інтернет речей (IoT), хмарні обчислення і мережева безпека. Це розширює сферу симуляцій і дозволяє користувачам досліджувати різні мережеві концепції і технології.

3. комплексний набір мережевих інструментів. Cisco Packet Tracer надає повний набір мережевих інструментів, які допомагають в усуненні несправностей і оптимізації мережі. Ці інструменти включають ping, traceroute, debug і Wireshark, які допомагають діагностувати мережеві проблеми, аналізувати мережевий трафік і забезпечувати оптимальну продуктивність мережі.

4. різноманітність мережевих пристроїв і топологій. за допомогою Cisco Packet Tracer користувачі мають доступ до широкого спектру мережевих пристроїв, включаючи маршрутизатори, комутатори, концентратори, сервери та кінцеві точки. Ці пристрої можна використовувати для створення різноманітних мережевих топологій, що дозволяє користувачам проектувати і налаштовувати мережі відповідно до конкретних вимог. Така універсальність розширює можливості для навчання та експериментів.

5. багатокористувацька співпраця. Cisco Packet Tracer підтримує багатокористувацьку співпрацю, дозволяючи декільком користувачам одночасно працювати над одним мережевим проектом. Ця функція особливо корисна для мережевих викладачів і студентів, які можуть співпрацювати над мережевими завданнями і проектами. Вона сприяє командній роботі та обміну знаннями у віртуальному мережевому середовищі.

В цілому, Cisco Packet Tracer пропонує потужний набір функцій, які полегшують моделювання, конфігурацію та усунення несправностей мережі. Це цінний інструмент для мережевих спеціалістів, студентів та викладачів.

Використання Cisco Packet Tracer має кілька переваг, які сприяють його широкому розповсюдженню. Ось переваги використання Cisco Packet Tracer:

1.забезпечує безпечне і віртуальне середовище для мережевих експериментів. Cisco Packet Tracer пропонує змодельоване мережеве середовище, яке дозволяє користувачам експериментувати з різними мережевими топологіями, протоколами і конфігураціями. Це усуває ризик пошкодження фізичного обладнання або порушення роботи мережі, забезпечуючи безпечний простір для навчання і досліджень.

2.зменшує залежність від фізичного обладнання. Імітуючи мережеве обладнання та з'єднання, Cisco Packet Tracer усуває необхідність інвестувати в дороге фізичне мережеве обладнання. Користувачі можуть створювати і налаштовувати віртуальні пристрої, заощаджуючи витрати, пов'язані з придбанням або орендою фізичного мережевого обладнання.

3.сприяє ефективному налаштуванню мережі та усуненню несправностей. Cisco Packet Tracer дозволяє користувачам швидко і ефективно налаштовувати мережеві пристрої та імітувати мережевий трафік. Ця функція допомагає в тестуванні та усуненні несправностей мережевих конфігурацій, виявленні потенційних проблем і оптимізації продуктивності мережі. Оптимізуючи ці процеси, Cisco Packet Tracer економить час, ресурси і мінімізує ризик простою мережі.

4.сприяє розвитку необхідних мережевих навичок і знань. Завдяки практичному досвіду роботи з мережевими концепціями і технологіями, Cisco Packet Tracer допомагає користувачам розвивати основні навички і знання, необхідні для кар'єри в сфері ІТ та мережевих технологій. Працюючи зі змодельованими мережевими сценаріями, користувачі можуть зрозуміти фундаментальні принципи, попрактикуватися у вирішенні проблем і набути впевненості у своїх силах.

5.дозволяє створювати та обмінюватися мережевими проектами та сценаріями. Cisco Packet Tracer дозволяє користувачам створювати і ділитися мережевими проектами і сценаріями з іншими, полегшуючи спільну роботу над проектами, тренінгами та освітніми заходами. Ця можливість сприяє

обміну знаннями, забезпечує зворотний зв'язок і покращує навчальний процес у віртуальному мережевому середовищі.

Загалом, переваги Cisco Packet Tracer включають безпечне віртуальне навчальне середовище, економічну ефективність, ефективну конфігурацію мережі та усунення несправностей, розвиток навичок, а також можливість співпраці та обміну мережевими проектами і сценаріями. Ці переваги роблять його цінним інструментом для мережевих фахівців, студентів і викладачів.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ МЕРЕЖІ

3.1. Побудова топології

Перший етап побудови мережі полягає в розташуванні пристроїв на логічній топології. Для цього потрібно виконати план мережі, який був описаний у другій частині:

1. Визначити приблизну схему кожної кімнати.
2. Розмістити відповідні пристрої в кожній кімнаті.
3. Провести з'єднання між пристроями, використовуючи різні типи з'єднань.
4. Налаштувати сервіси та мережеві протоколи.

Таким чином, ми створимо початкову структуру мережі, яка відповідає нашому плану.

Особлива увага повинна бути приділена створенню серверних кластерів для забезпечення високої доступності та надійності системи. Серверний кластер - це група з'єднаних серверів, які працюють разом для розподілення навантаження та забезпечення резервування. Це дозволяє забезпечити безперебійну роботу системи навіть у випадку відмови окремих серверів.

Створення топології серверних кластерів у середовищі Cisco Packet Tracer може включати наступні етапи:

1. Вибрати та розташувати сервери: вибрати необхідну кількість серверів для кластера та розмістити їх у топології. Використати пристрої "Сервер" з бібліотеки Cisco Packet Tracer.

2. Налаштування мережі: Підключити сервери до мережі, встановивши необхідні з'єднання між ними. Використати комутатори або маршрутизатори для створення мережевої інфраструктури. Налаштувати IP-адреси та мережеві параметри для кожного сервера.

3. Конфігурація серверів: Налаштувати операційну систему на кожному сервері. Встановити необхідні програми, сервіси та компоненти, що використовуються в кластері.

4. Тестування та перевірка: Виконати тестування роботи кластеру, переконатися, що всі сервери належним чином взаємодіють та розподіляють навантаження. Перевірити, чи працюють механізми резервування та відновлення при відмовах серверів.

Приклад налаштування однієї з точок доступу бездротового зв'язку наведено на рис. 3.1.

Для динамічної роздачі **IP** адрес зконфігуруємо **DHCP** на маршрутизаторі. Зарезервуємо певний діапазон адрес, щоб статично призначити їх кожному серверу. Лістинг наведений у додатку.

Побудована схема зображена на рис. 3.2.

Ефективність та справність мережі перевіримо після усіх налаштувань в одній з наступних глав.

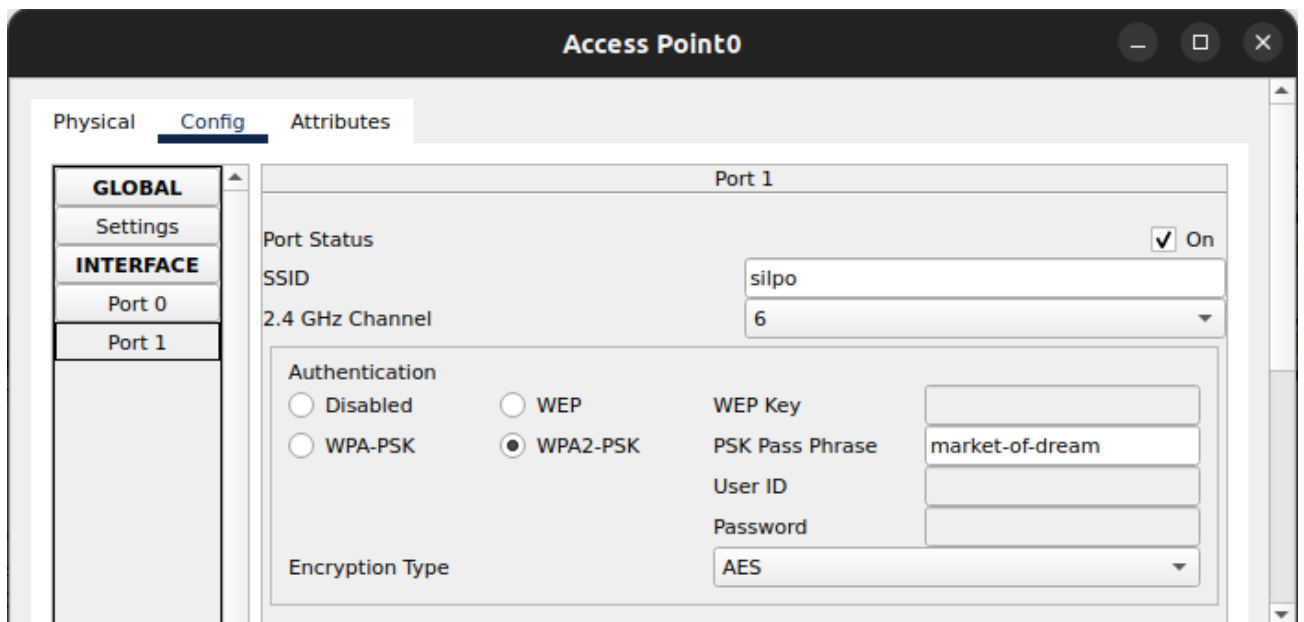


Рис. 3.1. Налаштування точки доступу

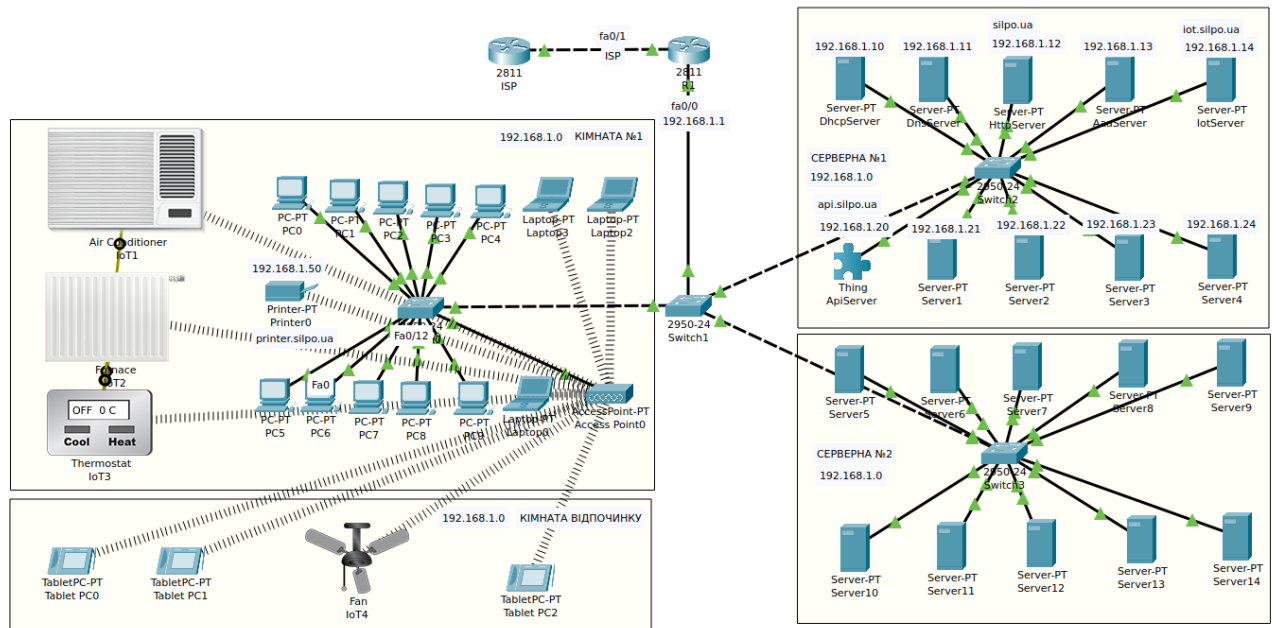


Рис. 3.2. Топологія офісу з відображенням бездротового з'єднання

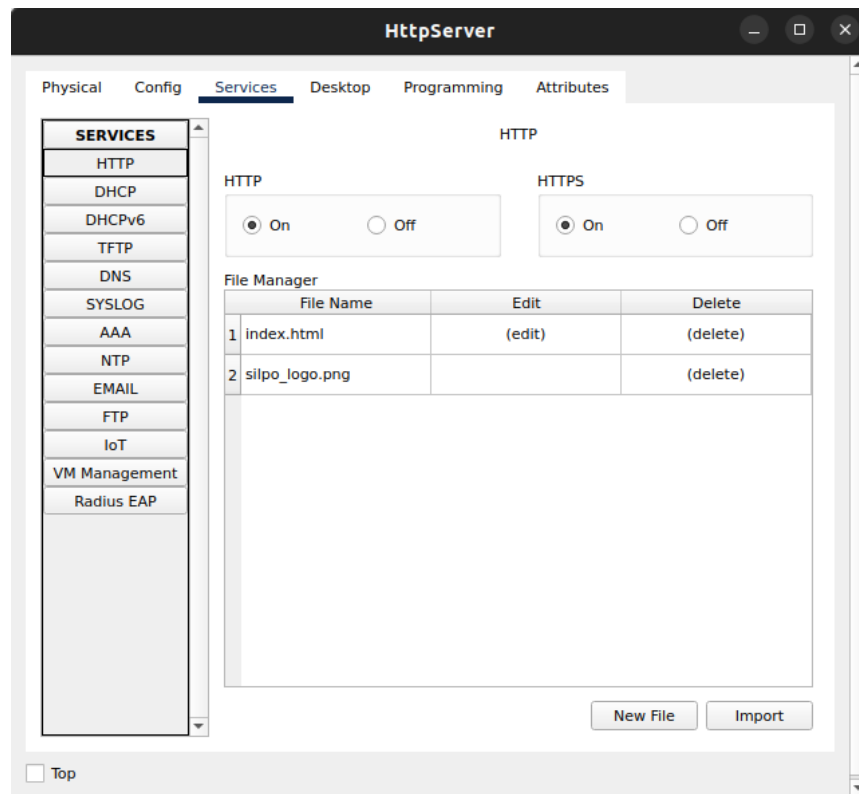


Рис. 3.3. Налаштування HTTP серверу

3.2. Налаштування сервісів

Вимоги до виконання включають налаштування сервісів, таких як HTTP та DNS. Налаштування цих сервісів зображено на рисунку 3.3 і рисунку 3.4 відповідно. Завдяки DNS ми можемо спростити конфігурацію IoT-пристроїв, встановивши домен (замість просто IP-адреси) в якості адміністративного

сервера. Сервіс HTTP буде використовуватися для внутрішнього сайту компанії. На головній сторінці буде розміщене посилання на "адмінку" IoT. Останнім сервісом є адміністративний сервер для IoT-пристроїв. Cisco Packet Tracer надає можливість використовувати свою вбудовану версію, яка налаштовується дуже просто. Приклад наведений на рисунку 3.5.

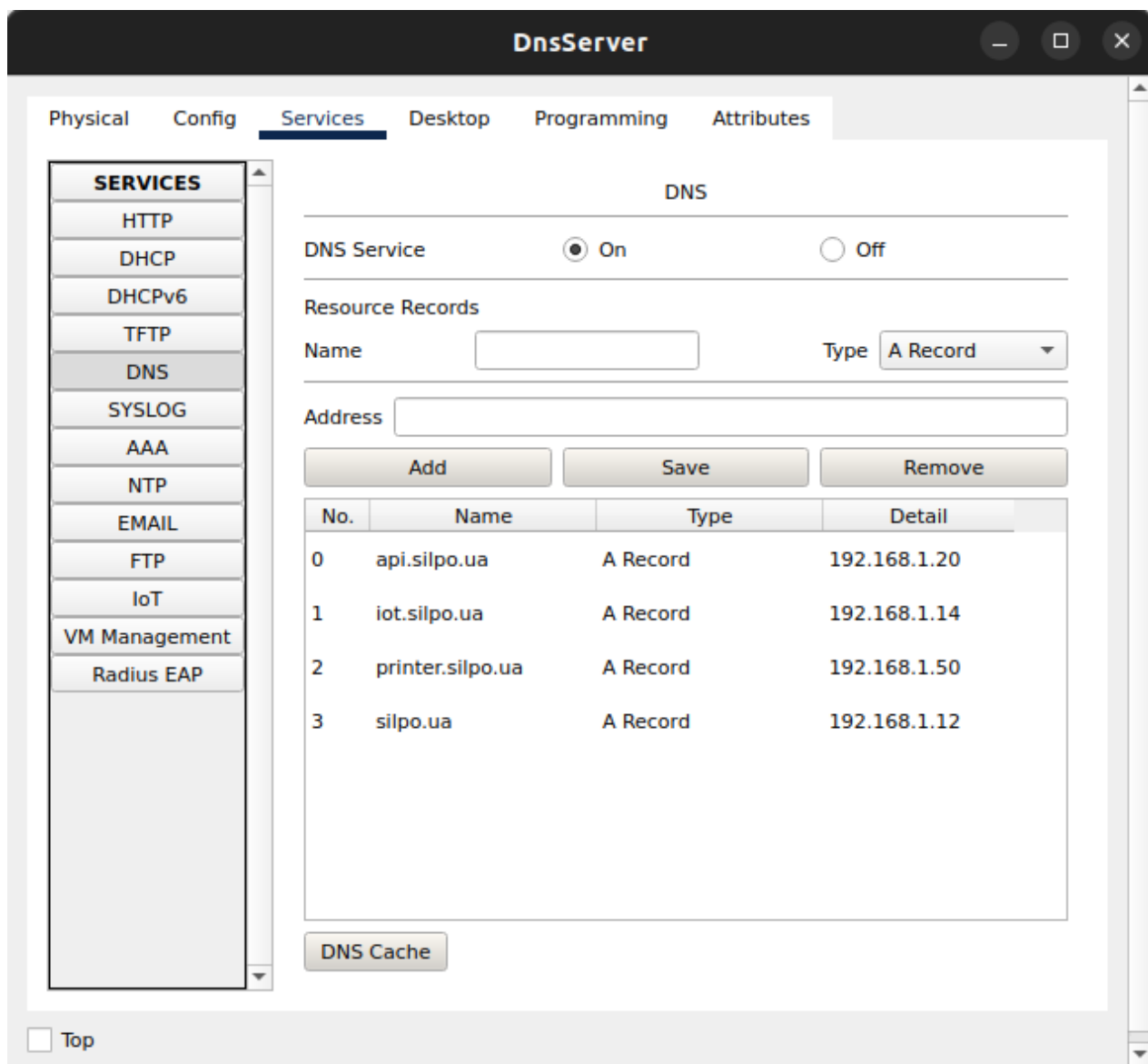


Рис. 3.4. Налаштування DNS серверу

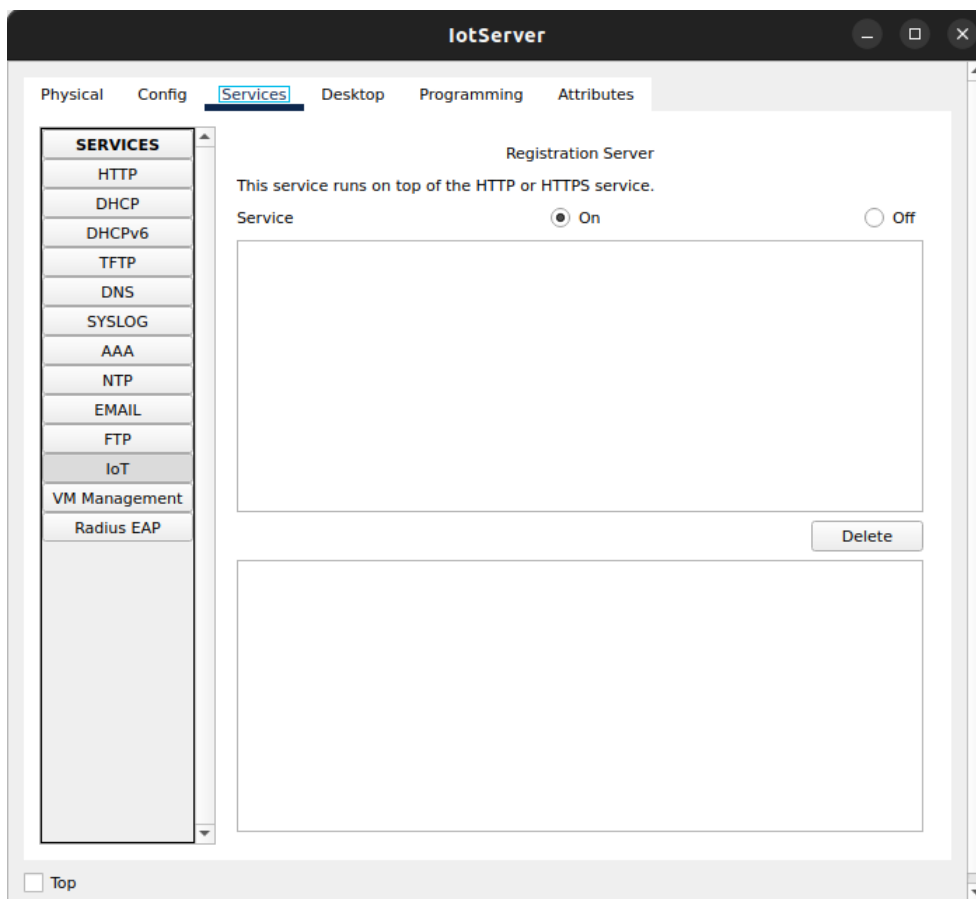


Рис. 3.5. Увімкнення IoT серверу

Всі пристрої IoT підключені за допомогою бездротового зв'язку. Кожен пристрій IoT повинен бути підключений до центрального адміністративного сервера, щоб мати повний доступ до пристроїв з будь-якого місця в мережі. Це здійснюється за допомогою вбудованого адміністративного сервера Інтернету речей, який було налаштовано раніше. Для конфігурації IoT попередньо було використано налаштований домен. Приклад наведений на рис. 3.6.

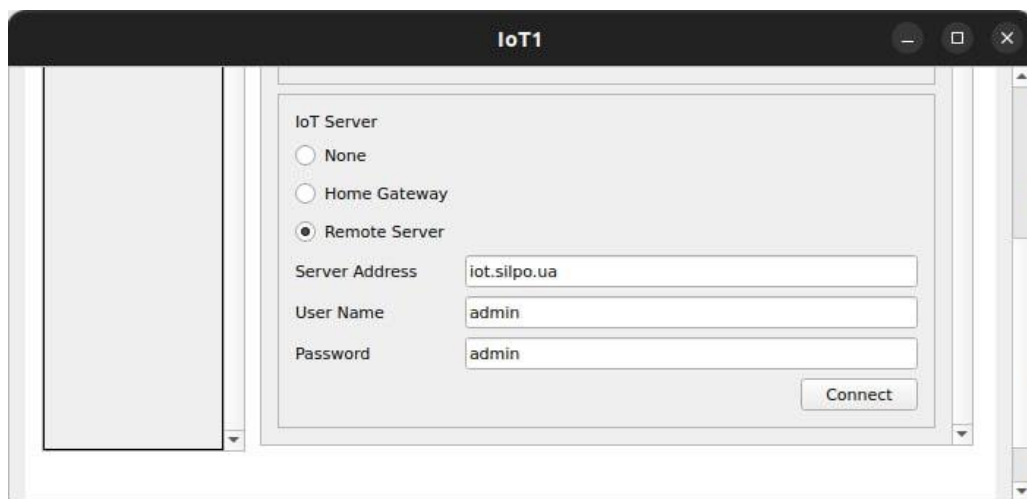


Рис. 3.6. Приклад налаштування адмін серверу для IoT

3.3. Програмування клієнт-серверного компоненту для тестування

3.3.1. Можливості програмування в Cisco Packet Tracer

Cisco Packet Tracer пропонує різні способи програмування девайсів, серед яких:

- Blockly.
- JavaScript;
- Python;

JavaScript та Python є чудовими інструментами для цього випадку, оскільки вони є скриптовими мовами програмування, які дозволяють легко і швидко реалізувати ідеї.

Для програмування IoT в Cisco Packet Tracer буде використовуватись мова Python.

3.3.2. Програмування

Запрограмуємо компонент топології на основі клієнт-серверної архітектури.

Для цього створимо API бекенд, який буде працювати в серверному кластері. Цей бекенд буде виконувати роль сервера. Він буде відповідати за обробку запитів від клієнтів та надання їм необхідних даних або послуг - в даному випадку для автентифікації (звірка логіна й паролю) та авторизації (видачі access токену).

Також розробимо універсальний фронтенд, що будет здатен працювати на будь-якому вузлі мережі. Фронтенд буде виконувати роль клієнта, який взаємодіє з сервером, надсилаючи й отримуючи запити та відповіді — в даному випадку для отримання access токену.

Використаємо вбудовану в Cisco Packet Tracer бібліотеку — запрограмуємо передачу даних, використовуючи протокол TCP. Для цього скористаємося стандартними сокетми, а саме, обгортками у вигляді TCP-серверу та TCP-клієнту, що можна імпортувати з вбудованої TCP бібліотеки.

Процес успішного виконання роботи можна описати наступним чином:

- бекенд (tcp-сервер) починає прослуховувати порт та очікувати на з'єднання клієнтів;
- фронтенд (tcp-клієнт) підключається до бекенду;
- бекенд приймає запит на підключення та готується виконувати команди, що будуть надіслані згодом;
- фронтенд посилає команду для отримання access токена та супроводжує її обліковими даними для цього, а саме, логіном та паролем;
- бекенд оброблює команду та перевіряє дані в локальній базі:
 - випадку успішної перевірки повертається access токен;
 - інакше повертається пуста відповідь.

Перевірка роботи даної системи забезпечить нам гарантію, що мережа функціонує належним чином та має цілісність, всі використані технології та сервіси налаштовані коректно, а топологія відповідає своїм цілям.

Повний лістинг коду наведено у додатку.

3.4. Перевірка справності мережі

Далі, ми перевіримо роботу протоколів HTTP та DNS. Для цього ми перейдемо на домен компанії з будь-якого пристрою в мережі.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)
	Successful	PC CEO	Kitchen-Coffee-1	ICMP	Green	0.000
	Successful	Work-Room-Laptop-1	Conference-Room-Tablet-1	ICMP	Purple	0.000
	Successful	Conference-Room-Tablet-1	Talk-Room-Music-Player	ICMP	Light Green	0.000
	Successful	Work-Room-PC-1	Server5	ICMP	Yellow	0.000
	Successful	Work-Room-Laptop-4	Kitchen-Tablet-1	ICMP	Red	0.000

Рис. 3.7. Перевірка справності мережі в режимі PDU.

Для перевірки справності комунікації між пристроями використаємо пінгування, результат якого наведено на рис. 3.7.

Щоб переглянути повний список підключених IoT-пристроїв, можна перейти на "адмінку", яка працює на сабдомені компанії. Результат показано на рисунку 3.12. Таким чином, ми перевіряємо статус роботи сервера IoT.

Наступним кроком перевірки буде тестування роботи протоколів HTTP та DNS. Для цього ми перейдемо з будь-якого пристрою в мережі на домен компанії. Скріншот сторінки у браузері можна побачити на рисунку 3.8.

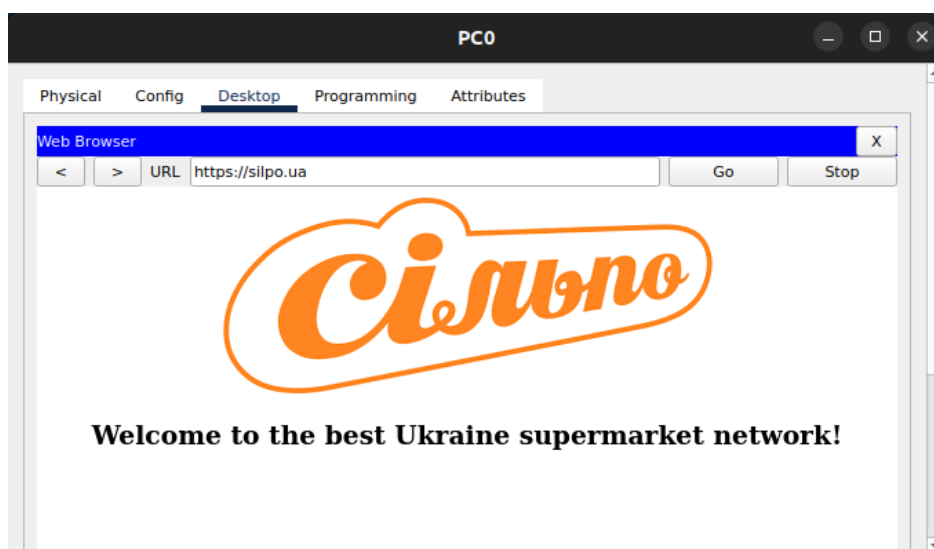


Рис. 3.8. Перевірка HTTP сервісу

Для перегляду повного списку підключених IoT-пристроїв можна перейти на "адмінку", яка працює на сабдомені компанії. Результат показаний на рисунку 3.9. Таким чином, ми перевіряємо статус роботи сервера IoT. Перевірка роботи API серверу показана на рис 3.10, а перевірка роботи API серверу на стороні користувача наведена на рис 3.11.

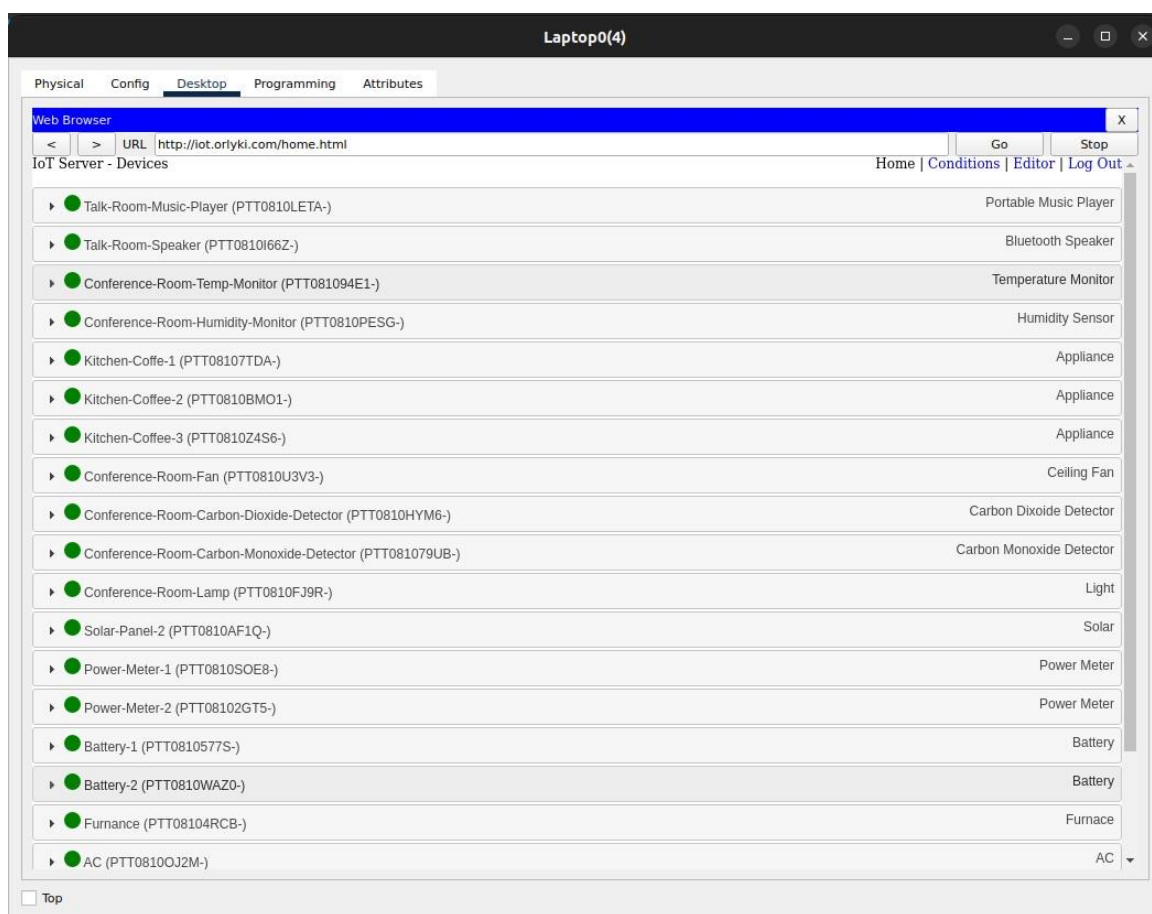
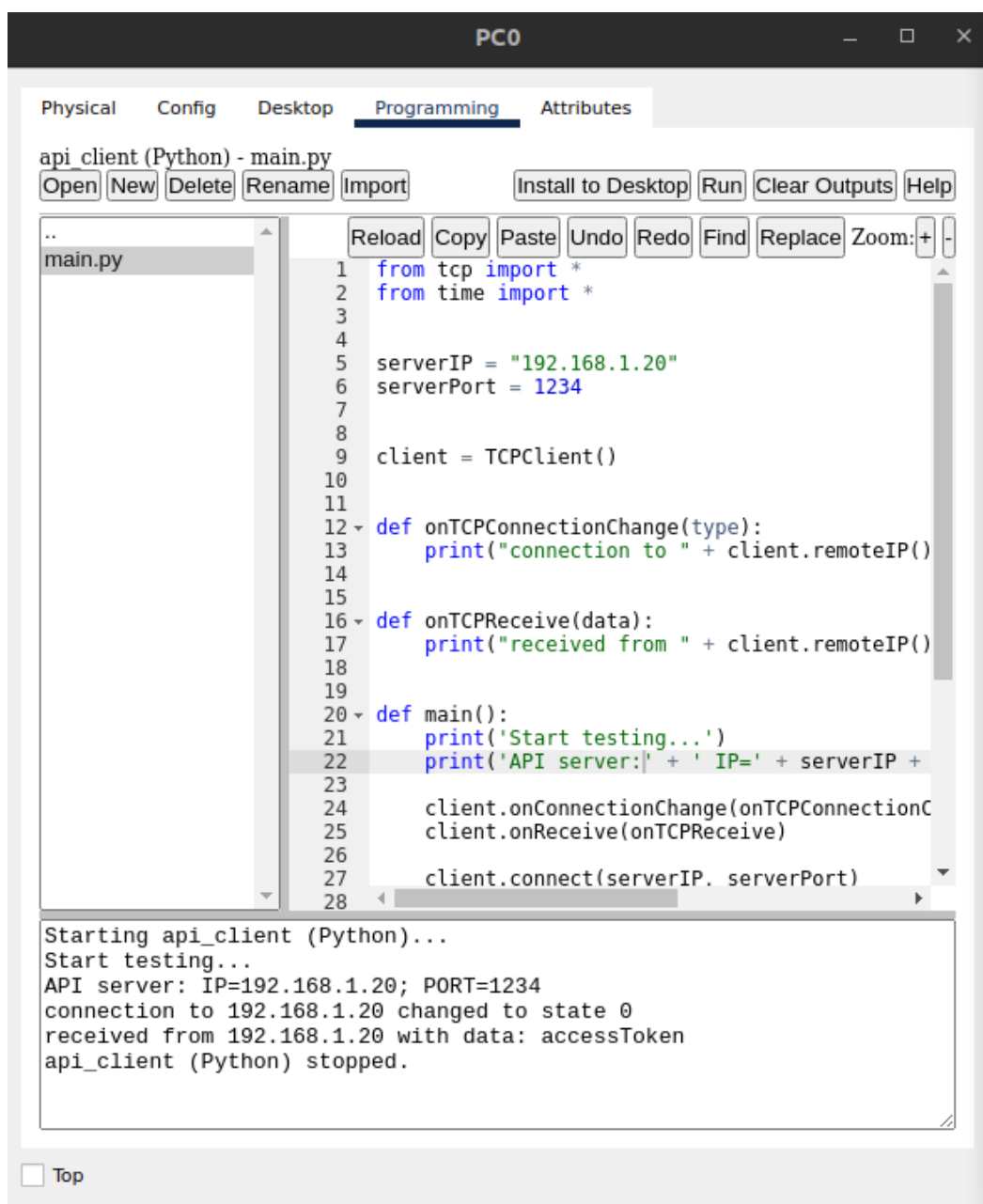


Рис. 3.9. Адмін сайт IoT.



Рис. 3.10. Перевірка роботи API серверу



The image shows a terminal window titled "PC0" with a menu bar containing "Physical", "Config", "Desktop", "Programming", and "Attributes". The "Programming" tab is active, showing a Python script named "main.py" in a text editor. The script is as follows:

```
1 from tcp import *
2 from time import *
3
4
5 serverIP = "192.168.1.20"
6 serverPort = 1234
7
8
9 client = TCPClient()
10
11
12 def onTCPConnectionChange(type):
13     print("connection to " + client.remoteIP())
14
15
16 def onTCPReceive(data):
17     print("received from " + client.remoteIP())
18
19
20 def main():
21     print('Start testing...')
22     print('API server:|' + ' IP=' + serverIP +
23
24     client.onConnectionChange(onTCPConnectionC
25     client.onReceive(onTCPReceive)
26
27     client.connect(serverIP, serverPort)
28
```

Below the code editor is a terminal output window showing the execution of the script:

```
Starting api_client (Python)...
Start testing...
API server: IP=192.168.1.20; PORT=1234
connection to 192.168.1.20 changed to state 0
received from 192.168.1.20 with data: accessToken
api_client (Python) stopped.
```

At the bottom left of the terminal window, there is a "Top" button with a small square icon next to it.

Рис. 3.11. Перевірка роботи API серверу на стороні користувача

ВИСНОВКИ

В даній роботі вирішені наступні завдання:

- дана характеристика серверів та їх типи;
- розкриті функціональні можливості серверів;
- розглянута організація роботи серверів;
- згідно вимог до побудови топології офісу компанії, зпроектовано розміщення та обрано мережеве обладнання;
- побудована топологія, проведено налаштування маршрутизаторів, серверів і IoT пристроїв, а також перевірена працездатність мережі офісу в системі Cisco Packet Tracer.

У процесі виконання завдання підприємства з побудови мережі було проведено проектування та налаштування топології мережі відповідно до вимог. З метою якісного й комплексного тестування працездатності топології, систему було запрограмовано за допомогою бекенду, який працює на сервері та здійснює обробку запитів клієнтів і надає необхідних даних або послуг. Фронтенд виступає в ролі клієнта і взаємодіє з сервером, відправляючи й отримуючи запити та відповіді. Для реалізації цього використовуються транспортний протокол передачі даних та сокети.

Базуючись на результатах виконання дипломної роботи можна зробити наступні висновки та рекомендації.

Сервери виконують роль сервісних засобів, які надають дані і ресурси для спільного використання та обслуговують різні клієнтські запити. У великих мережах сервери стають необхідністю, оскільки вони задовольняють зростаючі потреби корпоративних користувачів. Зазвичай, для задоволення різних потреб, сервери можуть бути спеціалізованими для підтримки конкретних служб або додатків, або ж об'єднані (консолідовані) в один потужний апаратний сервер.

Подальший розвиток серверів полягає в їхньому комплексному використанні у хмарних технологіях, які надають масштабовану інфраструктуру та програмні засоби безпосередньої залежності від фізичних

машин. Хмарні технології дозволяють використовувати безліч фізичних серверів як єдине обчислювальне середовище. У хмарних обчисленнях користувачі отримують доступ до додатків через Інтернет, використовуючи браузер або інші мережеві застосування, при цьому не завантажуючи ресурси свого комп'ютера або локального сервера. Користувач має повний доступ до своїх даних і може працювати з ними з будь-якої точки світу і на будь-якому пристрої, не займаючись управлінням операційною системою, програмною базою та обчислювальними потужностями. Зберігання даних і додатків у хмарі змінює традиційну клієнт-серверну модель і спрямовує нас в бік нової обчислювальної парадигми, де на стороні користувача зберігається мінімально необхідна функціональність.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Види серверів і їх ролі [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://www.myvin.com.ua/news/12371-vydy-serveriv-i-ikh-rolj>.
2. Що таке сервер і для чого він потрібний [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://blogchain.com.ua/shcho-take-server-i-dlia-choho-vin-potribnyj/>.
3. Сервер – що це таке, як працює і навіщо потрібен. Види, функції та приклади. [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://termin.in.ua/server/>.
4. Робочі станції, сервери та обчислювальні центри [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://elib.lntu.edu.ua/sites/default/files/elib_upload/Savarin_Pavlo_ENP/teoretic/lec2.html.
5. Сервер [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://wikifizmat.udpu.edu.ua/index.php?title=%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80>.
6. Скопень М.М, Сукач М.К, Будя О.П, Артеменко О.І, Хрущ Л.А. Інформаційні системи і технології в готельно-ресторанному та туристичному бізнесі: підручник. Київ: Ліра-К, 2016. 764 с.
7. Сервери і системи зберігання даних [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <http://www.itnt.net.ua/lang/ua/servers.html>.
8. Windows Server 2019 Administration Fundamentals: A Beginner's Guide to Managing an
9. Таненбаум, Е., Ван Стін, М. Комп'ютерні мережі. Принципи, технології, протоколи. - М.: ДМК Прес, 2011. - 960 с.
10. Курковская, Г. І., Рябко, Б. Я., Силаєва, О. С. Основи мережевих технологій. - Київ: Видавничий дім «Слово», 2014. - 368 с.
11. Колесников, О. С., Войтенко, О. Є., Харченко, І. В. Комп'ютерні мережі і телекомунікації. - Київ: Київський університет, 2016. - 288 с.

12. Топологія комп'ютерних мереж [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/Топологія_комп%27ютерних_мереж (Дата звернення: 2023, 22 травня)
13. Peterson, L. L., Davie, B. S. Computer Networks: A Systems Approach. - San Francisco: Morgan Kaufmann, 2011. - 920 p.
14. Comer, D. E. Computer Networks and Internets. - Upper Saddle River, NJ: Pearson, 2014. - 672 p.
15. Stallings, W. Data and Computer Communications. - Upper Saddle River, NJ: Pearson, 2013. - 912 p.
16. Kurose, J. F., Ross, K. W. Computer Networking: A Top-Down Approach. - Boston: Pearson, 2016. - 864 p.
17. Cisco Networking Academy. CCNA Routing and Switching: Introduction to Networks. - Cisco Press, 2013. - 720 p.
18. Al-Shaer, E., Hamed, H. S. Security Design of Network Topologies: Theory and Practice. - CRC Press, 2015. - 200 p.
19. Lammle, T. CCNA Routing and Switching Study Guide: Exams 100-105, 200-105, and 200-125. - Wiley, 2016. - 1152 p.
20. Odom, W. CCENT/CCNA ICND1 100-105 Official Cert Guide. - Cisco Press, 2016. - 1024 p.
21. Hartpence, B. Network Topology and Its Engineering Considerations. - CreateSpace Independent Publishing Platform, 2016. - 174 p.
22. Chakrabarti, P., Mishra, D. Network Topology in Command and Control: Organization, Operation, and Evolution.
23. Олинiк, В. І., Дробот, О. О., Кушнір, І. М. Комп'ютерні мережі. Основи та практика. - Київ: Кондор, 2013. - 384 с.
24. Глушченко, А. А., Задорожний, В. А., Платонов, А. М. Основи комп'ютерної інженерії. - Київ: МАУП, 2016. - 384 с.
25. Колесников, А. В., Гавриш, В. О., Руденко, В. В. Комп'ютерні мережі: практикум. - Київ: НТУУ "КПІ", 2016. - 208 с.

26. Іванчук, В. В., Литвиненко, О. І., Мельничук, В. О. Мережева топологія. - Київ: Ліра-К, 2015. - 384 с.
27. Глущенко, А. А., Бондаренко, О. О., Корнієнко, О. О. Комп'ютерні мережі: підручник. - Київ: Академвидав, 2016. - 320 с.
28. Дейтел, Г., Дейтел, П. Компьютерные сети, интернет и технологии. - Москва: Питер, 2012. - 1136 с.
29. Форузанн, Б. А. TCP/IP Протоколы, архитектура, реализация. - Москва: Диалектика, 2015. - 1184 с.
30. Гарсія-Моля, М., Велтен, Г. Топологія мереж: підхід з інженерії систем. - Київ: Ліра-К, 2017. - 208 с.

ДОДАТОК А. ЛІСТИНГ КОДУ**api_client.py**

```
from tcp import *
from time import *
serverIP = "192.168.1.20"
serverPort = 1234
client = TCPClient()
def onTCPConnectionChange(type):
    print("connection to " + client.remoteIP() + " changed to state " + str(type))
def onTCPReceive(data):
    print("received from " + client.remoteIP() + " with data: " + data)
def main():
    print('Start testing...')
    print('API server:' + ' IP=' + serverIP + '; PORT=' + str(serverPort))
    client.onConnectionChange(onTCPConnectionChange)
    client.onReceive(onTCPReceive)
    client.connect(serverIP, serverPort)
    token_command = "token|roman:mokryi"
    while True:
        data = token_command
        client.send(data)
        sleep(10)
if __name__ == "__main__":
    main()
```

api_server.py

```
from tcp import *
from time import *
```

```

port = 1234
server = TCPServer()
#####
# DB setup
db = {}
logins = {
    'roman': 'mokryi',
}db['logins'] = logins
token = 'accessToken'
#####
def handle_token_command(username, password):
    if username not in db['logins']:
        print(username + ': not in db')
        return ""
    db_password = db['logins'][username]
    if password != db_password:
        print(username + ': incorrect password')
        return ""
    print(username + ': matched')
    return token
def get_some_data(user_token):
    if user_token != token:
        print('incorrect token')
        return None
    return 'Here is some important data!'
def handle(data):
    splitted = data.split('|')
    command = splitted[0]
    if command == 'token':
        username_password = splitted[1].split(':')

```

```

        return handle_token_command(username_password[0],
username_password[1])
    if command == 'data':
        return get_some_data(splitted[1])
    return 'command not found'
def onTCPNewClient(client):
    def onTCPConnectionChange(type):
        print("connection to " + client.remoteIP() + " changed to state " +
str(type))
        def onTCPReceive(data):
            print("received from " + client.remoteIP() + " with data: " + data)
            response = handle(data)
            # send back same data
            client.send(response)
        client.onConnectionChange(onTCPConnectionChange)
        client.onReceive(onTCPReceive)
def main():
    server.onNewClient(onTCPNewClient)
    server.listen(port)
    print('listen on: ' + str(port))
    # don't let it finish
    while True:
        sleep(3600)
if __name__ == "__main__":
    main()

```

ДОДАТОК Б. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

Сервери та їх застосування в комп'ютерних мережах

Виконавець – Роман Мокрий

Науковий керівник – к.т.н., доцент Тетяна Демківська

Мета роботи

Вивчення та аналіз застосування серверів, а також проектування та налаштування мережі офісу компанії, розташованого в трьох кімнатах.



Основні типи серверів



- DHCP-сервери - сервери відповідають за видання параметрів клієнту, таких як унікальна IP-адреса та маска підмережі.
- DNS-сервери-сервери забезпечують перетворення імен вузлів на відповідні їм IP-адреси і є дуже важливими в мережах.
- WINS-сервери - це служба, яка використовується для розподілу імен комп'ютерів та їх відповідних IP-адрес в мережах Windows.

3

Існує велика кількість функцій у серверів, основними, із яких є функції:

- | | |
|---|------------------------|
| – пошукових систем; | – бази даних; |
| – зберігання та передачі файлів; | – відеоспостереження; |
| – електронної пошти; | – шлюзу Інтернету; |
| – організації та доступу до WEB – сторінки; | – віддаленого доступу. |

4

Переваги і причини у використанні топологій мережі

- економічна ефективність;
- поглиблене розуміння мережевих концепцій;
- ефективна комунікація за допомогою топологічних схем;
- легкість навчання;
- діагностика стану мережі;

5

Ключові можливості Cisco Packet Tracer

- імітація мережевих пристроїв і протоколів;
- підтримка декількох протоколів і технологій;
- комплексний набір мережевих інструментів;
- різноманітність мережевих пристроїв і топологій;
- багатокористувацька співпраця.



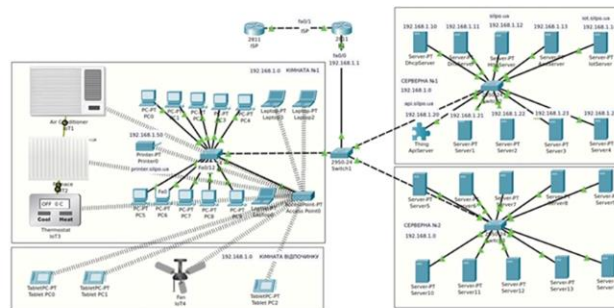
6

Створення топології серверних кластерів у середовищі Cisco Packet Tracer включає наступні етапи:

- вибір та розташування серверів;
- налаштування мережі;
- конфігурація серверів;
- тестування та перевірка.

7

Топологія офісу з відображенням бездротового з'єднання



8

Висновки

В даній роботі вирішені наступні завдання:

- дана характеристика серверів та їх типи;
- розкриті функціональні можливості серверів;
- розглянута організація роботи серверів;
- згідно вимог до побудови топології офісу компанії, зпроектовано розміщення та обрано мережеве обладнання;
- побудована топологія, проведено налаштування маршрутизаторів, серверів і IoT пристроїв, а також перевірена працездатність мережі офісу в системі Cisco Packet Tracer.