School of Computers and Information Engineering

Spring Semester 2025

Computer Networks (CIE 3110)

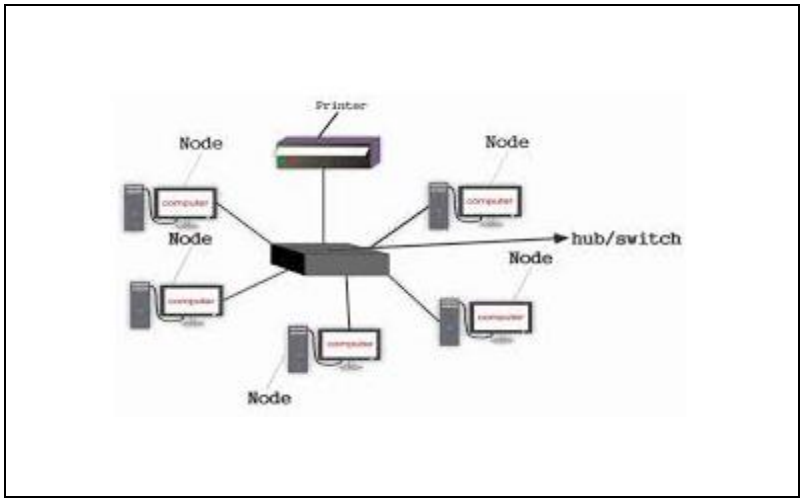
STUDENT CENTERED LEARNING ACTIVITY

Major Assignment

Network Simulation

using

Cisco Packet Tracer



Submitted to:
Prof. Ashish Seth

Submitted by

U2210141, Maxmudov Komron
(Section 001)

TOSHKENT SHAHRIDAGI INHA UNIVERSITETI
INHA UNIVERSITY IN TASHKENT

Contents

0 Introduction

0.1 Overview

In today's digital era, where information is abundant and constantly evolving, it becomes essential to transform theoretical knowledge into practical skills. This report focuses on applying foundational knowledge of **Computer Networks** through practical simulations. Utilizing Cisco Packet Tracer, I explore virtual network design and configuration. In the final part, I consider a real-world scenario involving a company that requires network planning by a qualified professional.

0.2 Cisco Packet Tracer

Cisco Packet Tracer is a powerful network simulation tool developed by Cisco. It enables users—students and professionals alike—to design, configure, and troubleshoot networks without needing physical devices.

Key Features:

- **Network Simulation** – Emulates routers, switches, end devices, and IoT components.
- **Realistic Configuration** – Offers CLI-based setup similar to real Cisco hardware.
- **Packet Analysis** – Allows users to visualize packet flow within a network.

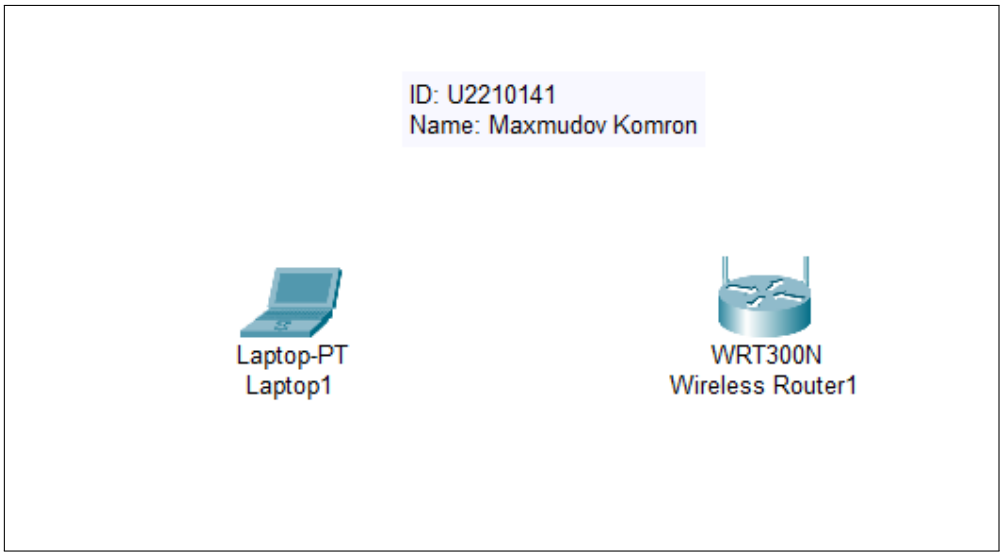
1 Activity 1 – Configuring a Wireless Router

1.1 Concept Overview

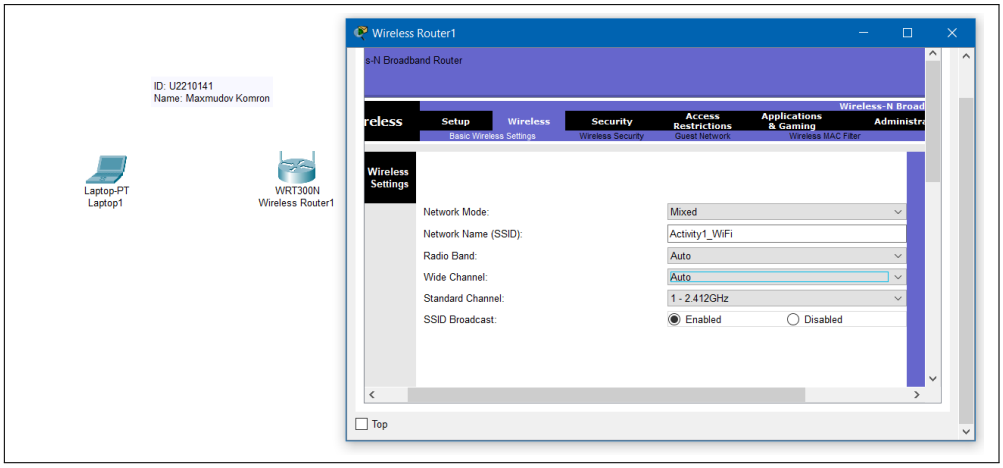
A **wireless router** allows devices to connect to a network without using cables. It functions as both an access point and a gateway to the internet or local resources. In this simulation, I configure a wireless router in Cisco Packet Tracer, set up Wi-Fi with WPA2 security, enable DHCP, and connect a laptop wirelessly to verify network functionality.

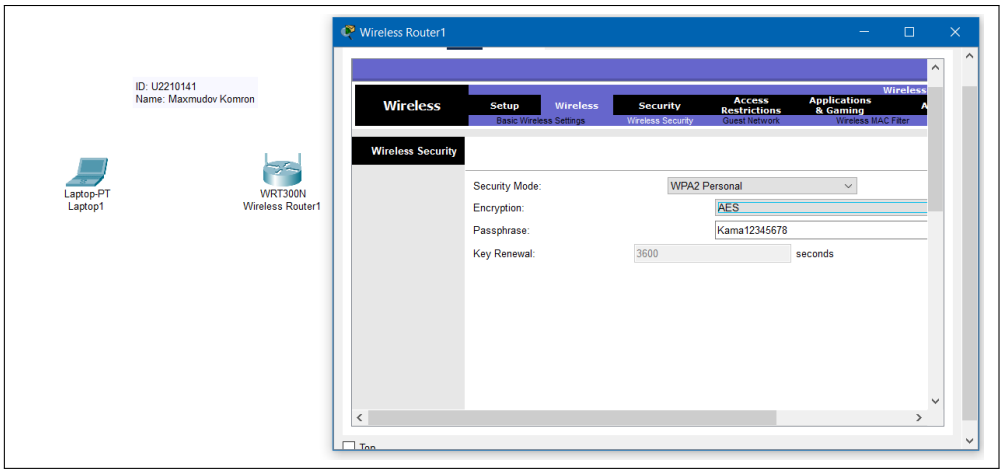
1.2 Simulation Steps

1. **Added Devices:** I placed a wireless router and a laptop in the workspace.

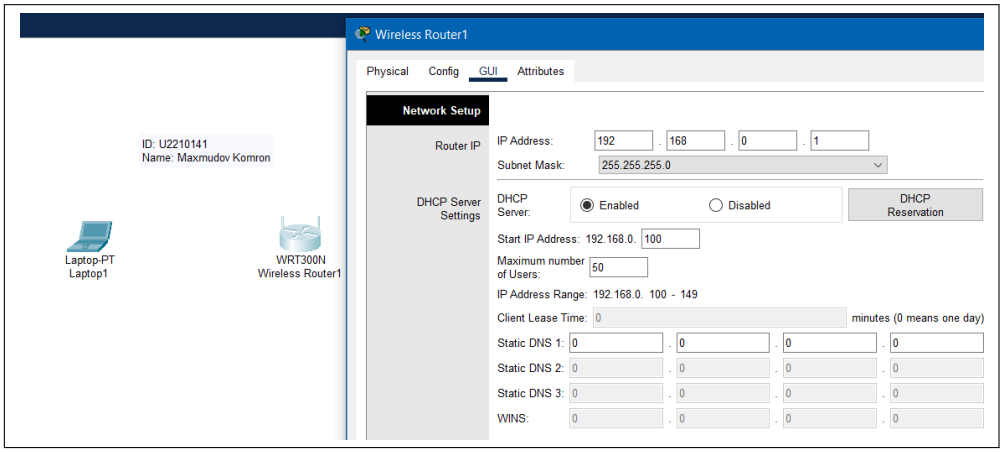


2. **Configured SSID and Security:** I assigned the SSID "Activity1-WiFi" and set WPA2-PSK with a strong password.

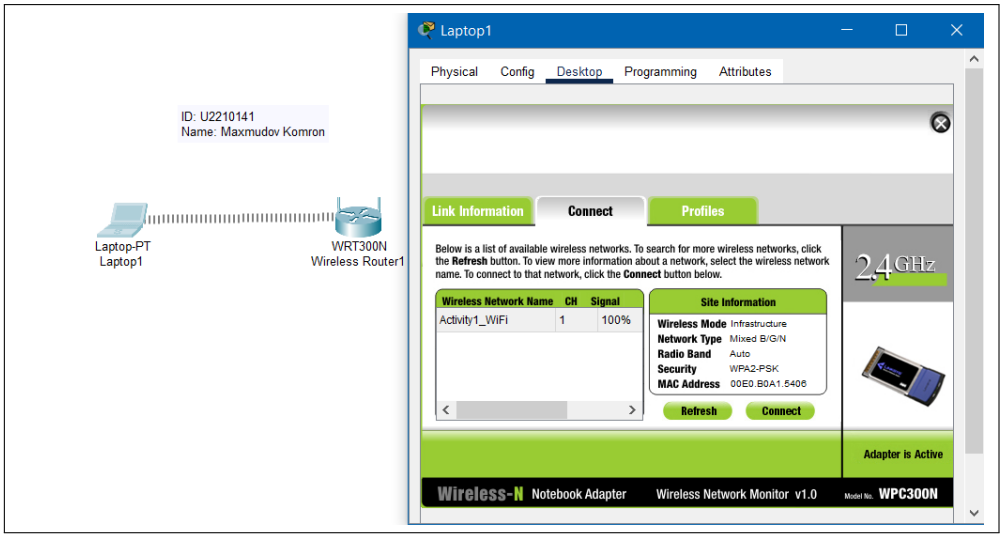




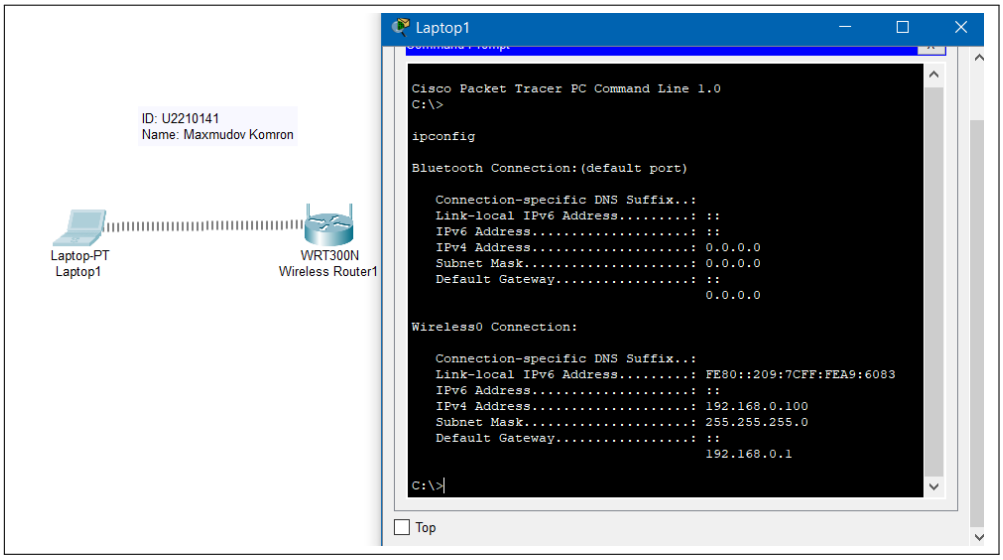
3. **Enabled DHCP:** DHCP was enabled on the router to assign IP addresses dynamically.



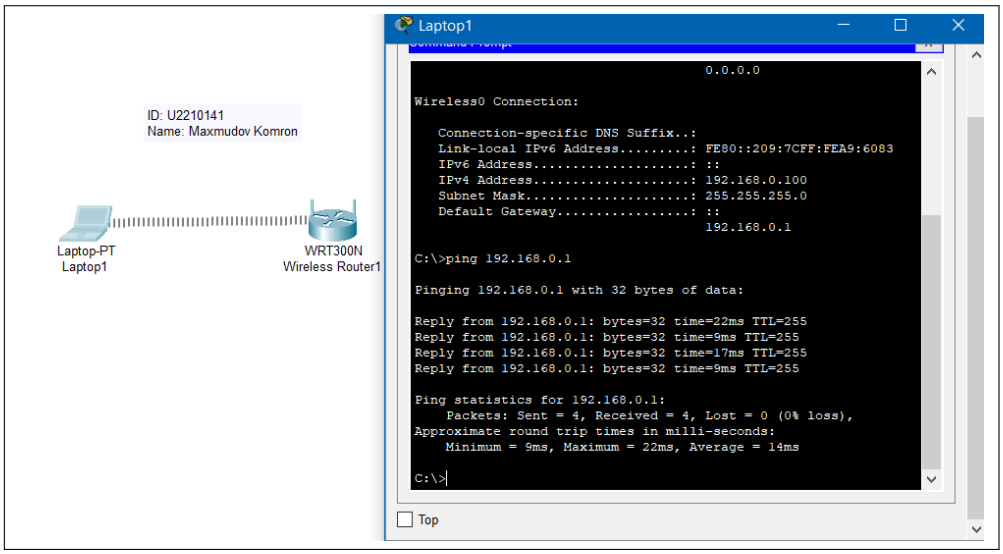
4. **Connected Laptop to Wi-Fi:** The laptop was connected to the network by selecting the SSID and entering the password.



5. **Verified IP Address:** The laptop received an IP address from the router, confirming a successful connection.



6. **Tested Connectivity:** I pinged the router from the laptop, and received replies — confirming a working wireless link.



1.3 Summary

I successfully configured a wireless router with secure settings and DHCP, connected a laptop via Wi-Fi, and verified both IP allocation and connectivity. This simulation models a standard home or small-office wireless setup.

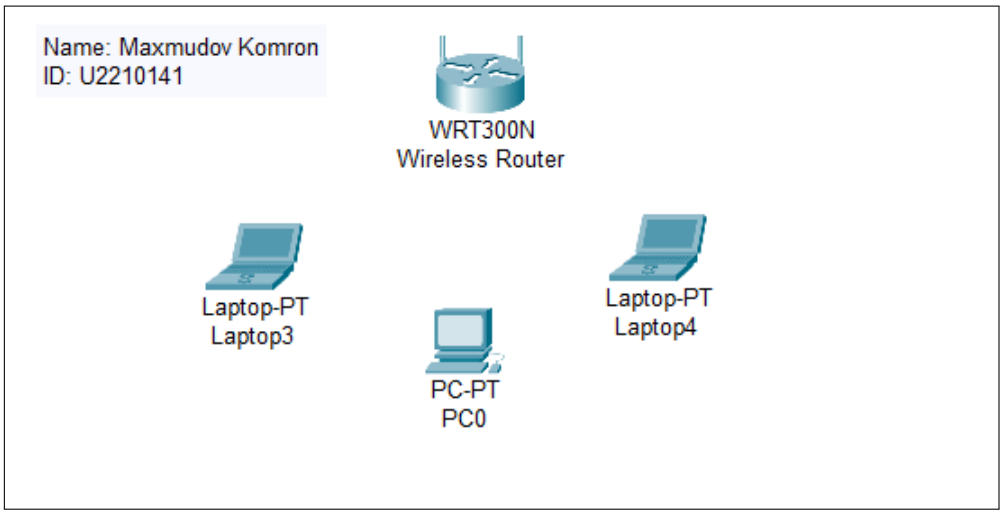
2 Activity 2 – Configuring WLAN using Packet Tracer

2.1 Concept Overview

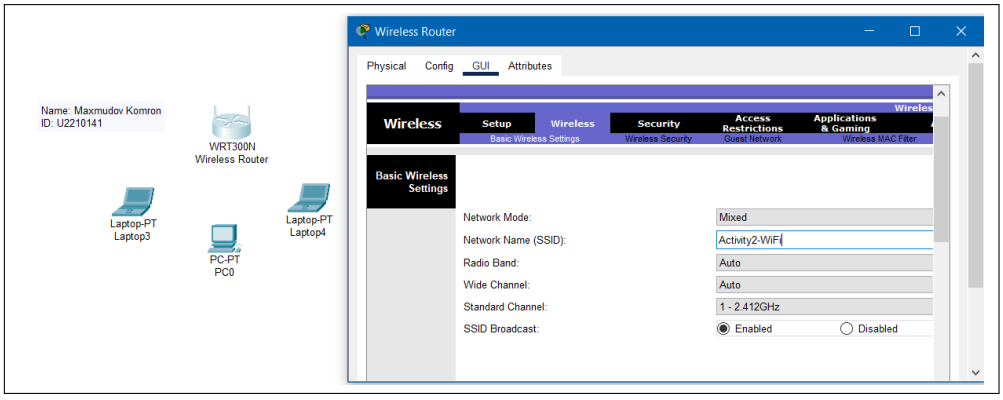
A **Wireless Local Area Network (WLAN)** allows devices to connect wirelessly within a specific range, using Wi-Fi standards. In this simulation, I will configure a WLAN using Cisco Packet Tracer, set up a wireless router as the access point, define the wireless network (SSID), enable security features, and connect multiple devices to verify network functionality.

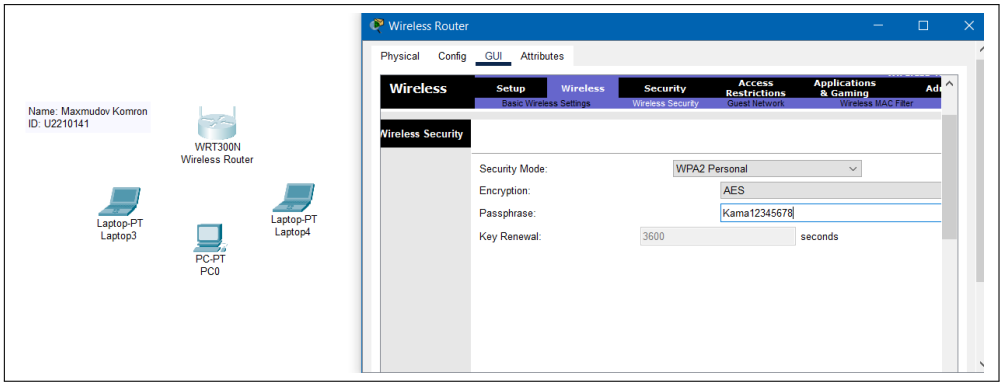
2.2 Simulation Steps

1. **Added Devices:** I placed the wireless router, two laptops, and a desktop in the Packet Tracer workspace.

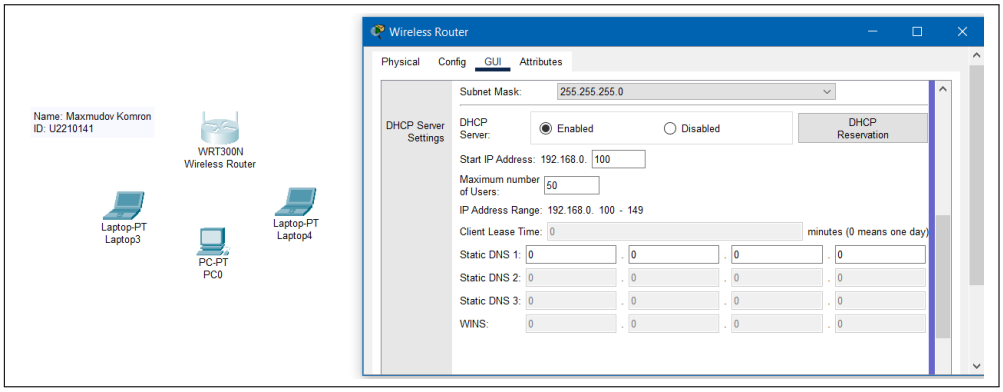


2. **Set SSID and Wireless Security:** I created the SSID ”Activity2-WiFi” and applied WPA2-PSK security with a secure password to prevent unauthorized access.

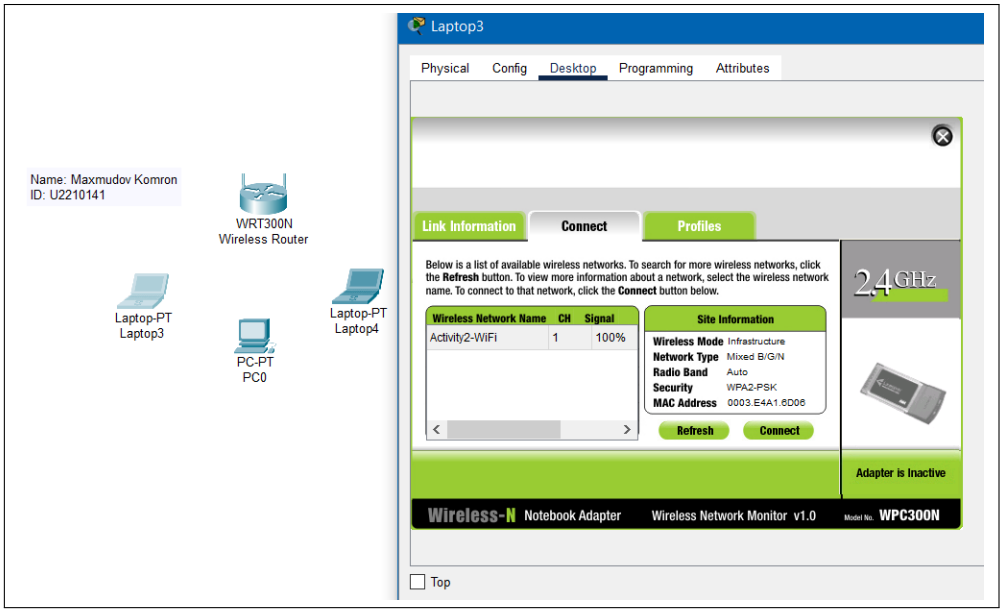


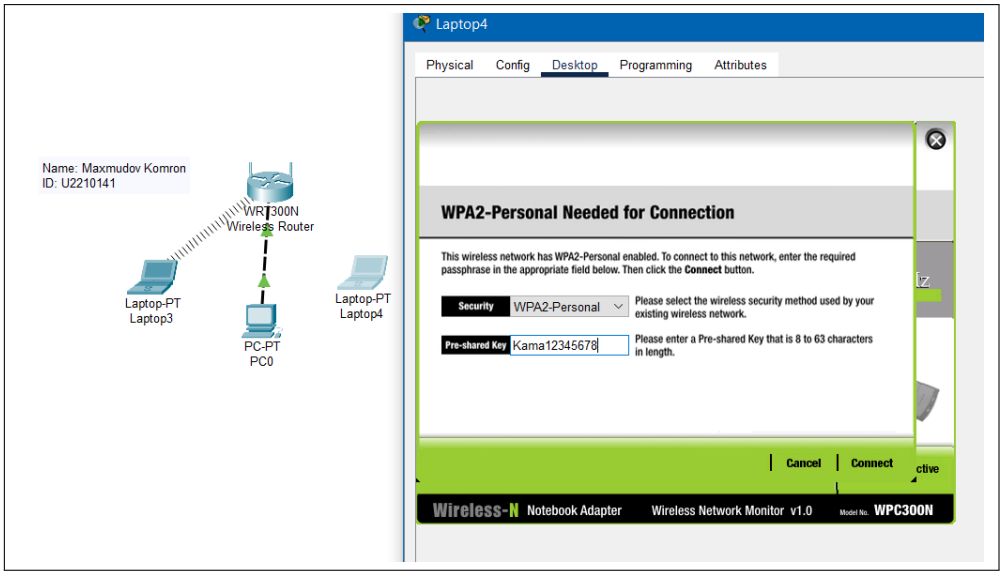


3. **Enabled DHCP on Router:** I enabled DHCP on the wireless router to automatically assign IP addresses to the connected devices.

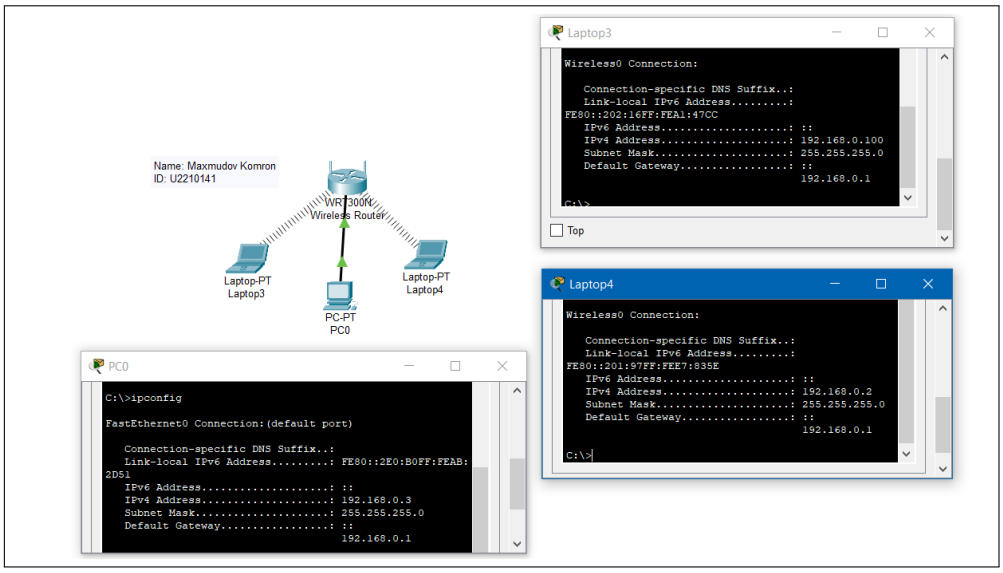


4. **Connected Devices to WLAN:** I connected the two laptops to the wireless network by selecting the SSID and entering the WPA2 password. The desktop was connected via a wired Ethernet connection because, in Packet Tracer, PCs do not have wireless network adapters, and they can only connect via wired connections.





5. **Verified IP Address Assignment:** I checked that all connected devices received an IP address from the router, confirming they were successfully connected to the WLAN (for laptops) and wired network (for the desktop).



2.3 Summary

In this activity, I successfully configured a WLAN by setting up a wireless router as an access point, securing the network with WPA2 encryption, enabling DHCP for automatic IP address assignment, and connecting multiple devices to the network. The laptops were connected via Wi-Fi, while the desktop was connected via a wired Ethernet connection. The desktop was connected through a wire because Packet Tracer does not support wireless network adapters for PCs, so the only option for connecting PCs is through a wired connection.

3 Activity 3 – Configuring DHCP on Router using Cisco Packet Tracer

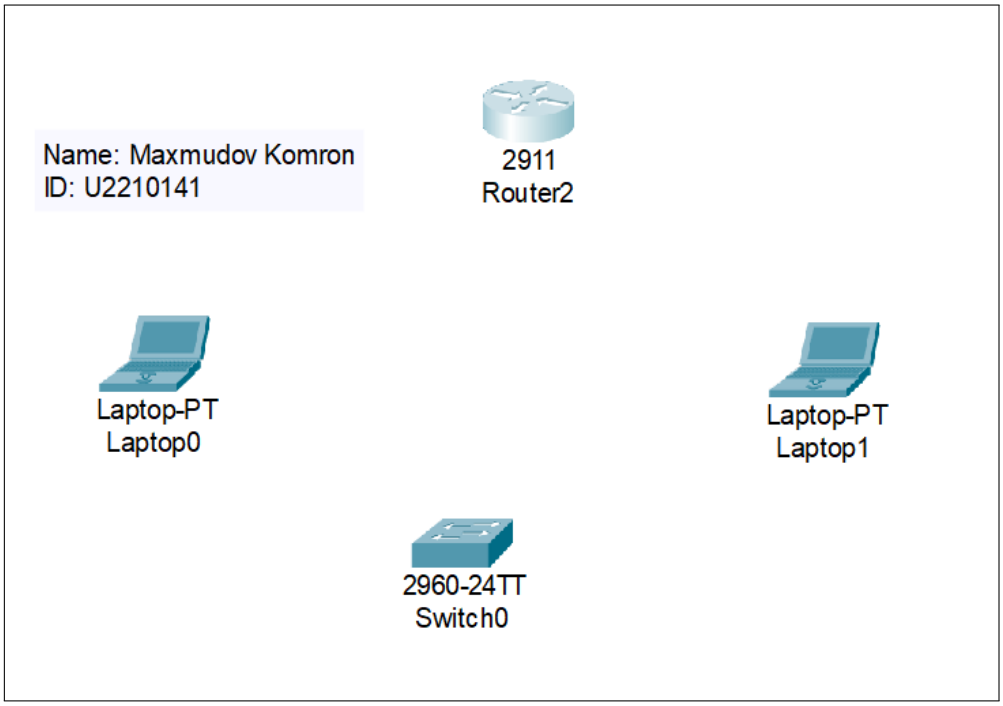
3.1 Concept Overview

A **Dynamic Host Configuration Protocol (DHCP)** server allows devices on a network to automatically obtain IP addresses and other network configuration settings. In this activity, I will configure a router to

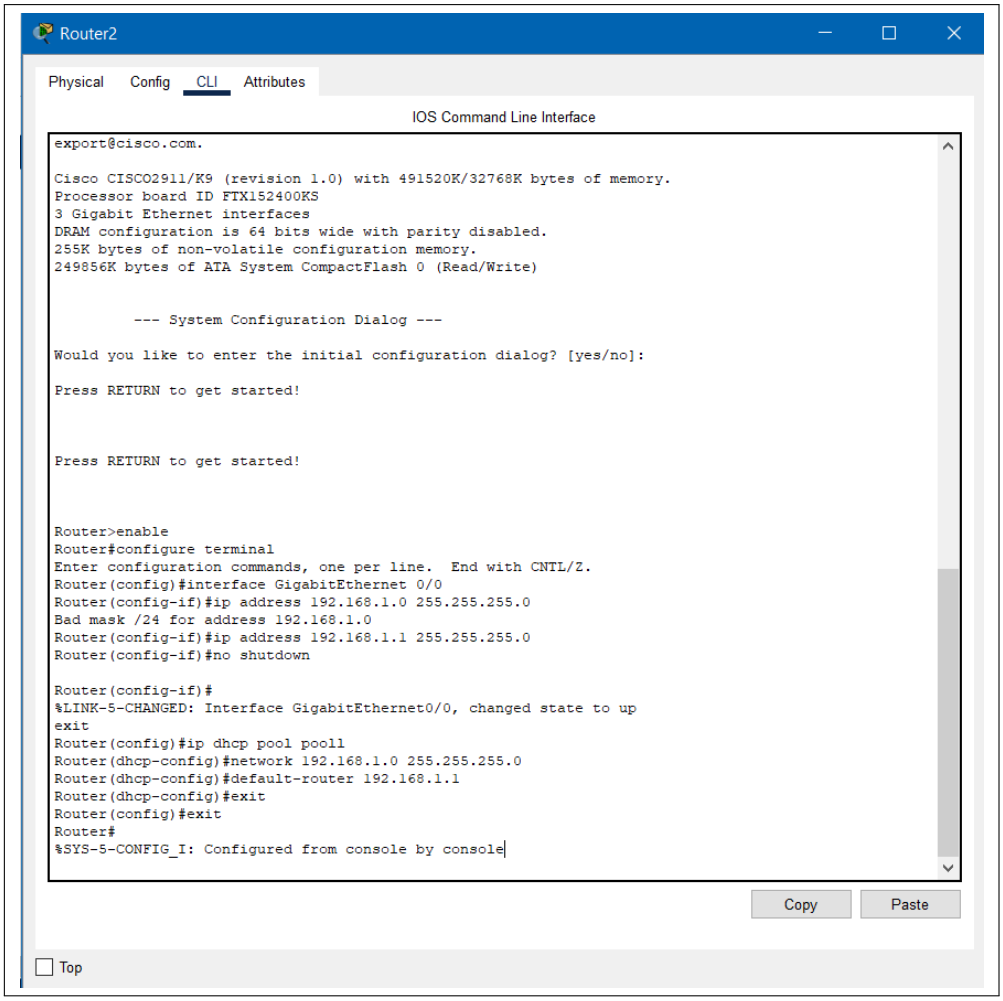
function as a DHCP server in Cisco Packet Tracer. The router will dynamically assign IP addresses to devices on the network, eliminating the need for manual IP address configuration.

3.2 Simulation Steps

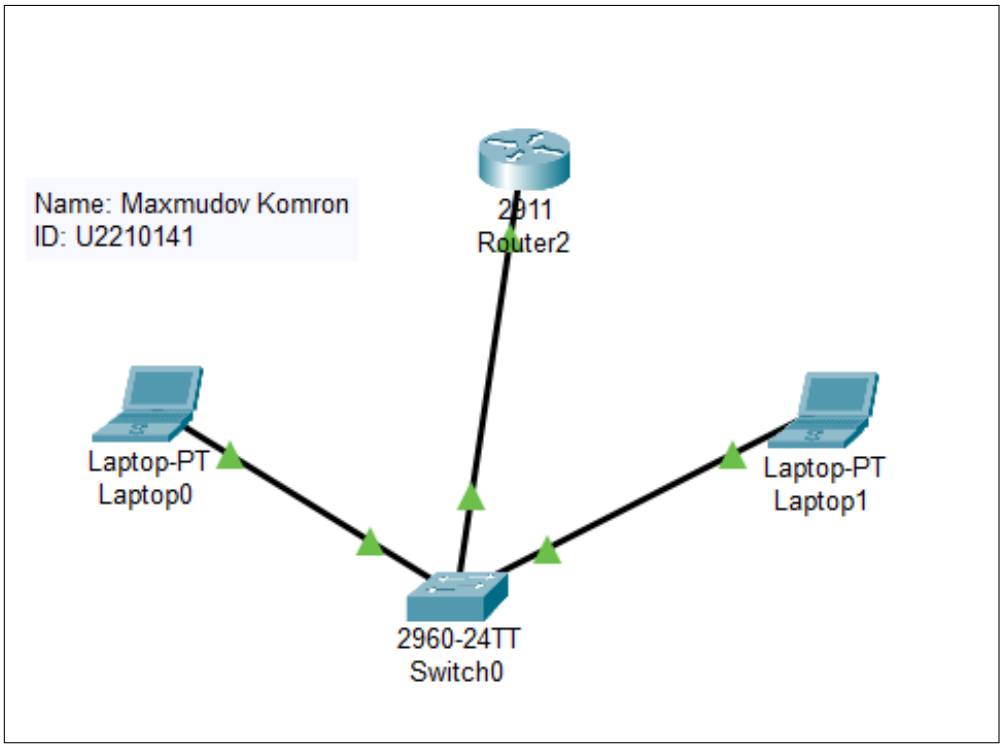
1. **Added Devices and Configured Router:** I placed a router, a switch, and several end devices in the Packet Tracer workspace.



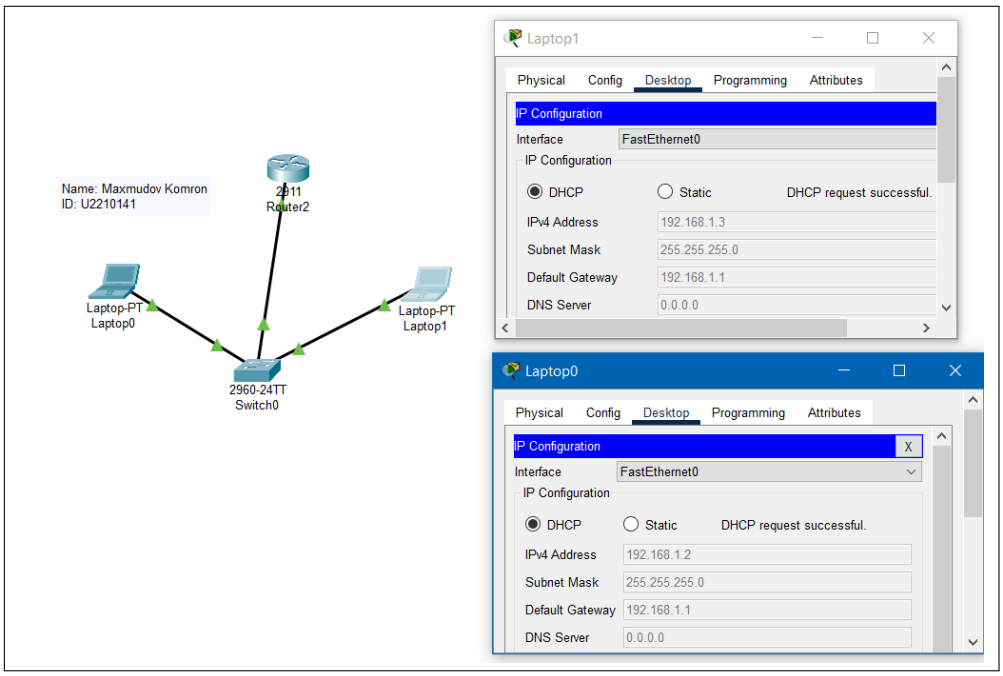
2. **Configured DHCP Pool and Enabled DHCP on Interface:** I defined the DHCP pool with a network range (e.g., `192.168.1.0` to `192.168.1.255`), set the router's IP address (`192.168.1.1`) as the default gateway, and configured DHCP to assign IP addresses within this range. DHCP was enabled on the router's `GigabitEthernet 0/0` interface to allow devices to automatically receive their IP addresses and network settings.



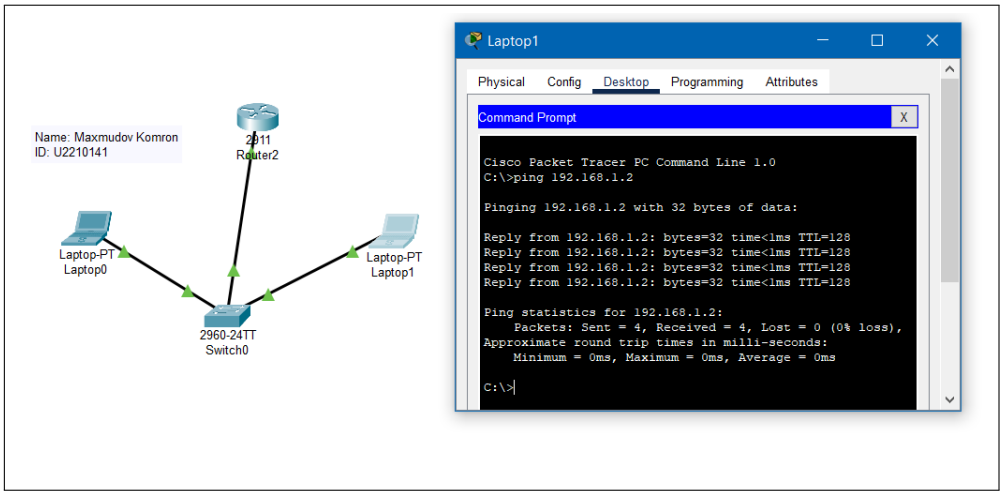
3. **Connected Devices to the Network:** I connected the devices to the switch, ensuring they were in the same network as the router. The devices were set to receive their IP addresses automatically from the router.



4. **Verified IP Address Assignment:** I opened the IP configuration settings on each device and set them to obtain an IP address automatically via DHCP. This allowed the devices to request IP addresses from the router's DHCP server.



5. **Tested Connectivity:** I performed a ping test between devices to verify that all devices were successfully connected to the network and were receiving proper IP addresses.



3.3 Summary

In this activity, I successfully configured a router as a DHCP server, which dynamically assigned IP addresses to devices on the network. Devices were connected to the network and received IP addresses via DHCP. Connectivity was verified using ping tests between devices.

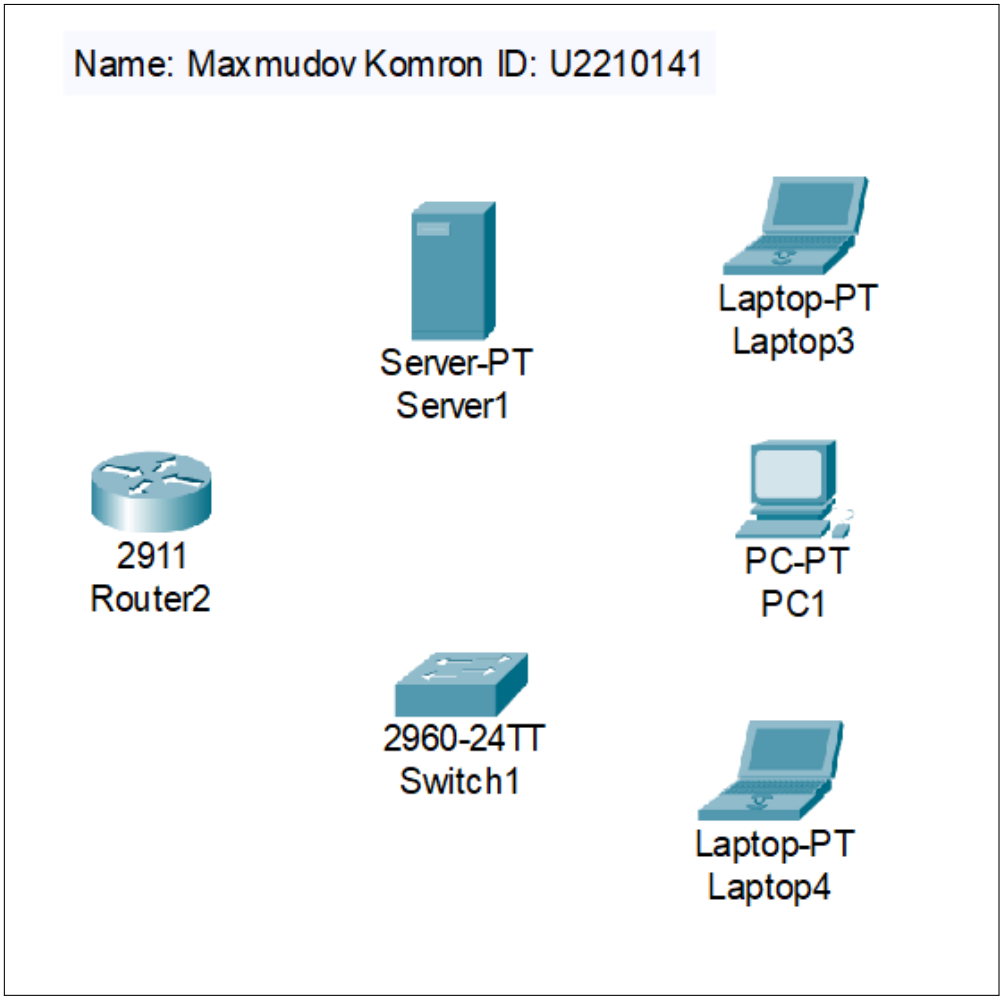
4 Activity 4 – Configuring a Dedicated DHCP Server using Cisco Packet Tracer

4.1 Concept Overview

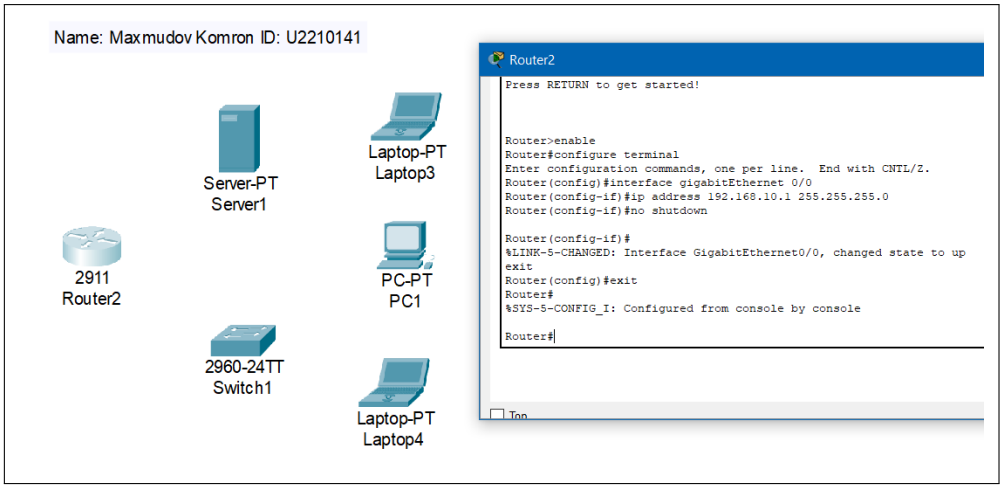
A **Dedicated DHCP (Dynamic Host Configuration Protocol) Server** is a standalone device or service responsible for automatically assigning IP addresses and other network settings to client devices. Unlike a router-based DHCP configuration, using a dedicated DHCP server allows for more centralized management and scalability. In this activity, I configured a DHCP server device in Cisco Packet Tracer to handle IP address distribution for devices connected to a network.

4.2 Simulation Steps

1. **Added Devices and Network Topology:** I placed a DHCP server, a router, a switch, and several end devices (PCs and laptops) in the Cisco Packet Tracer workspace.

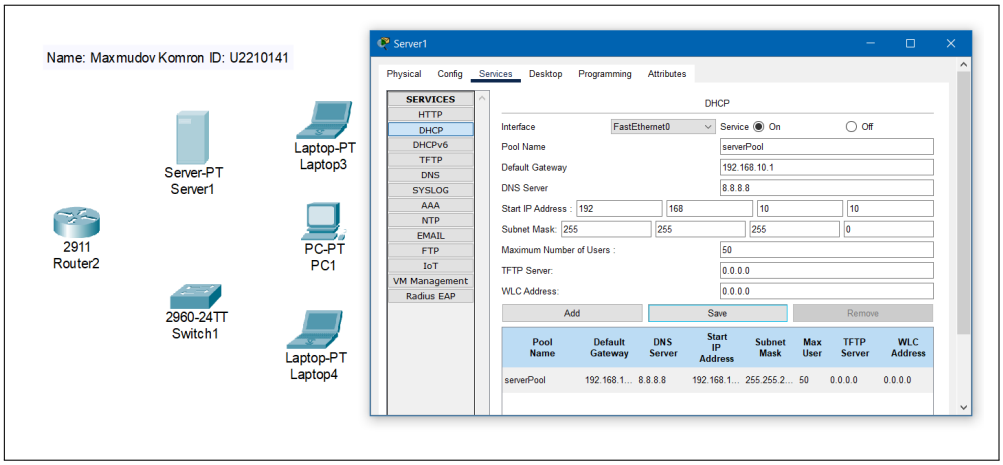


2. **Configured Router Interface:** I assigned the IP address `192.168.10.1` with a subnet mask of `255.255.255.0` to the router's `GigabitEthernet 0/0` interface and enabled it to act as the default gateway for the network.

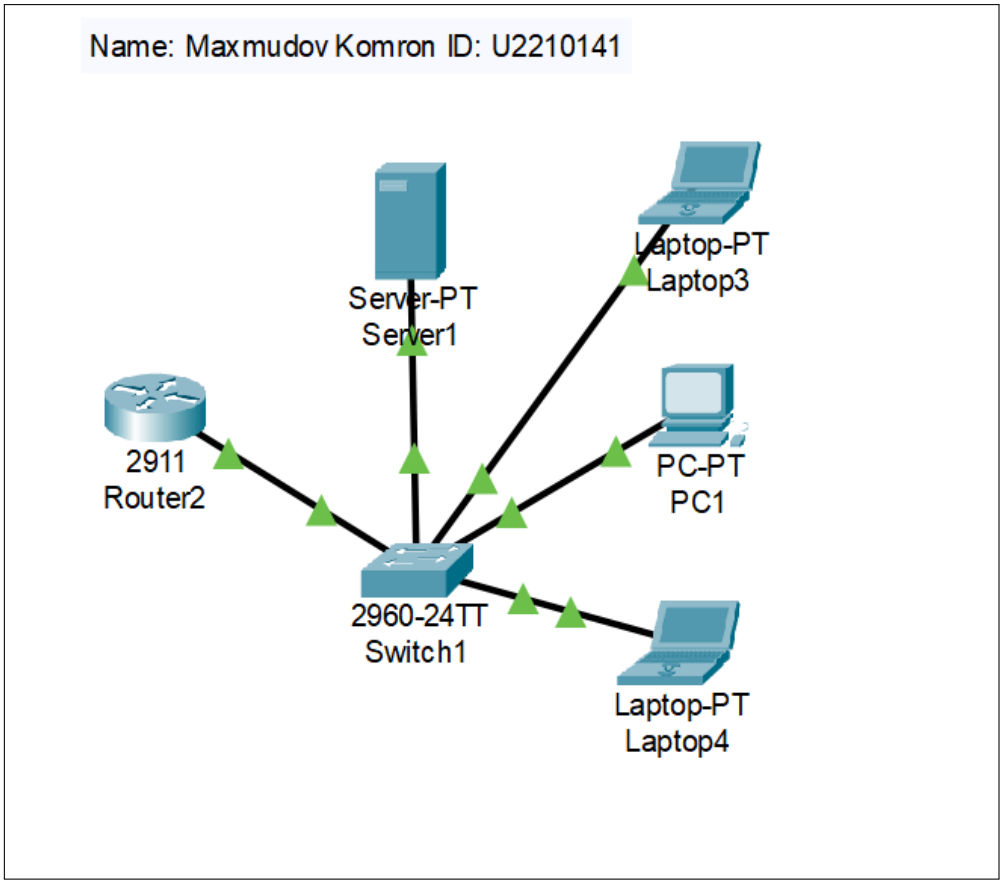


3. **Configured DHCP Server:** On the DHCP server, I opened the `Services` tab, selected `DHCP`, and created a DHCP pool with the following parameters:

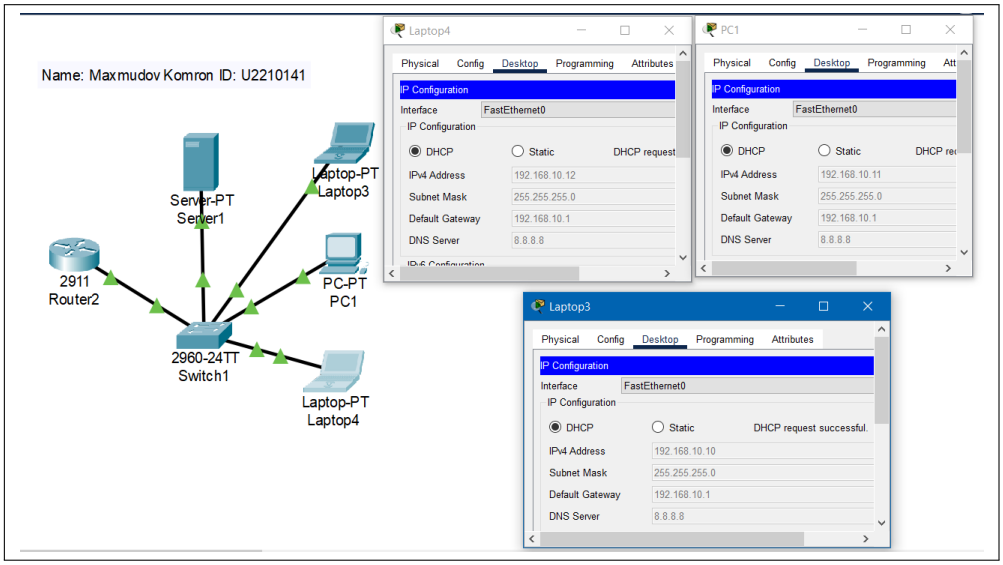
   - Default Gateway: `192.168.10.1`
   - Subnet Mask: `255.255.255.0`
   - DNS Server: `8.8.8.8`
   - Starting IP Address: `192.168.10.10`
   - Maximum Number of Users: 50



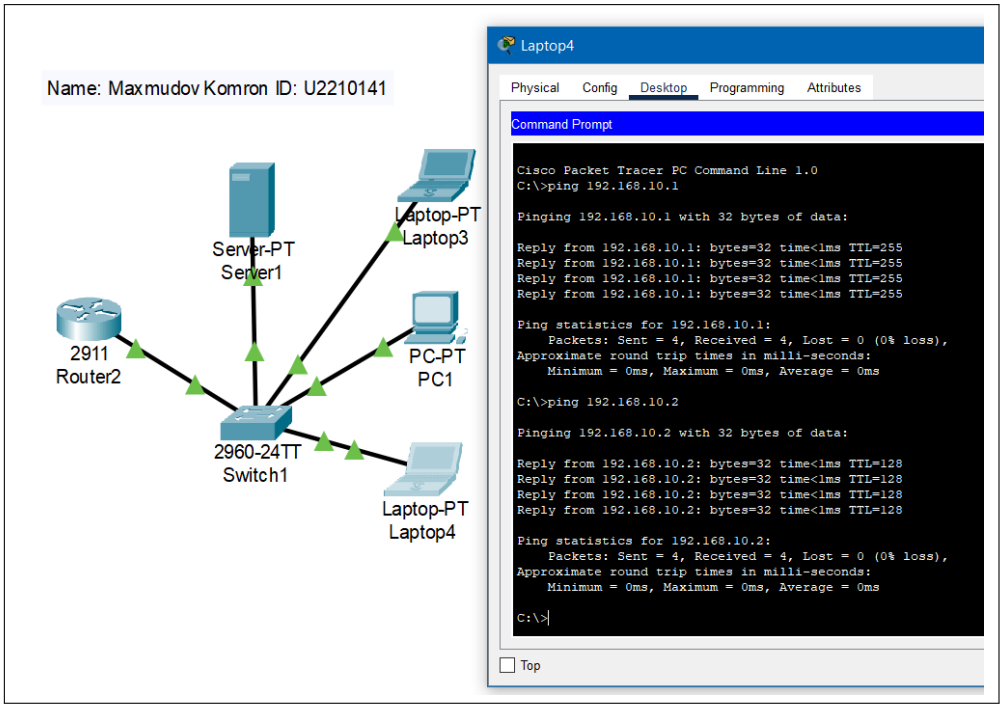
4. **Connected All Devices:** I connected the router, DHCP server, and end devices to the switch using copper straight-through cables. This ensured all devices, including the DHCP server, were on the same subnet and broadcast domain, allowing DHCP requests to reach the server directly without routing.



5. **Enabled DHCP on End Devices:** I opened the IP configuration on each PC and laptop, set them to obtain an IP address automatically, and verified that each device received an IP address from the DHCP server within the specified range.



6. **Tested Network Connectivity:** I used the `ping` command between end devices and to the default gateway to confirm successful IP assignment and ensure that all devices were able to communicate over the network.



4.3 Summary

In this activity, I successfully configured a dedicated DHCP server to automatically assign IP addresses to multiple client devices on a network. The DHCP server was connected to the switch along with all end devices, ensuring they shared the same local network. The router was configured as the default gateway. All devices were able to obtain their addresses dynamically and communicate with each other, confirming correct DHCP setup and full network connectivity.

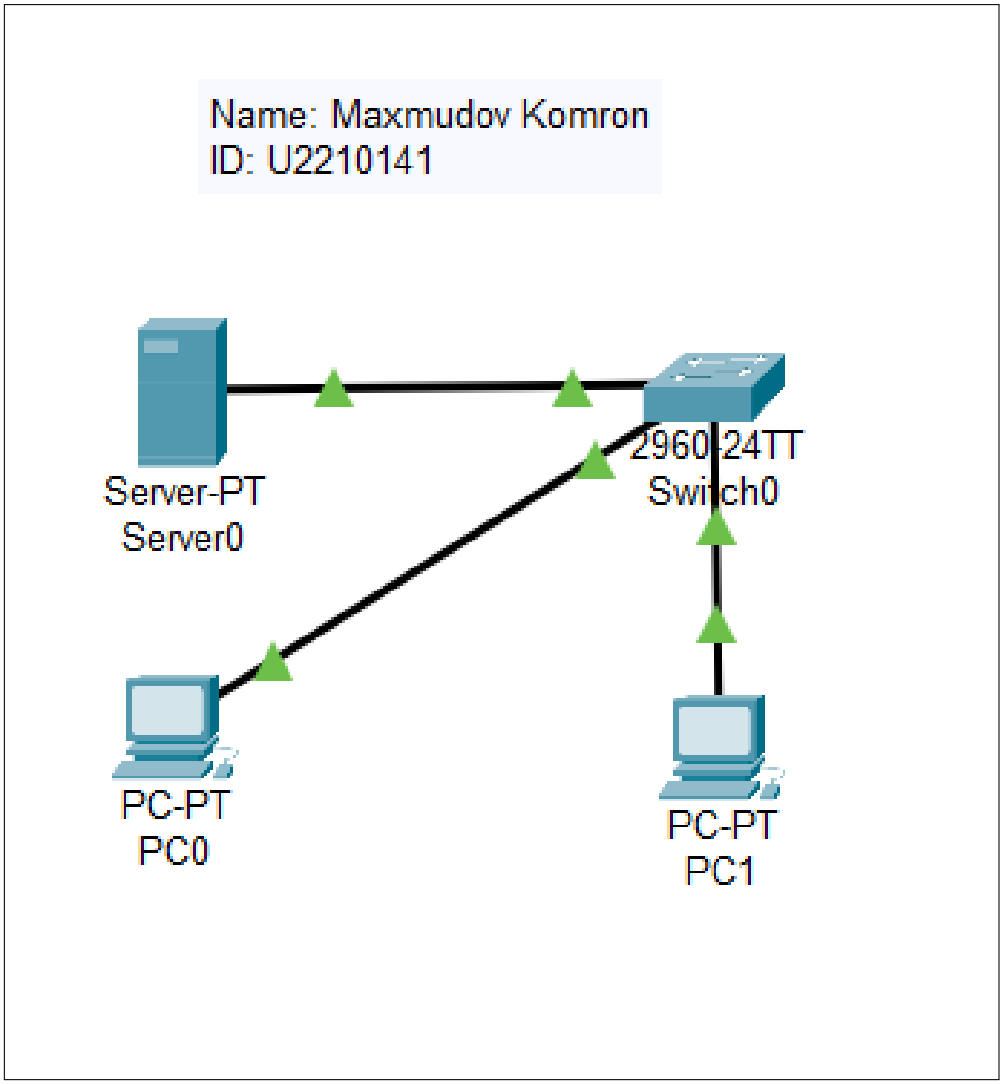
5 Activity 5 – Simulate FTP Server using Cisco Packet Tracer

5.1 Concept Overview

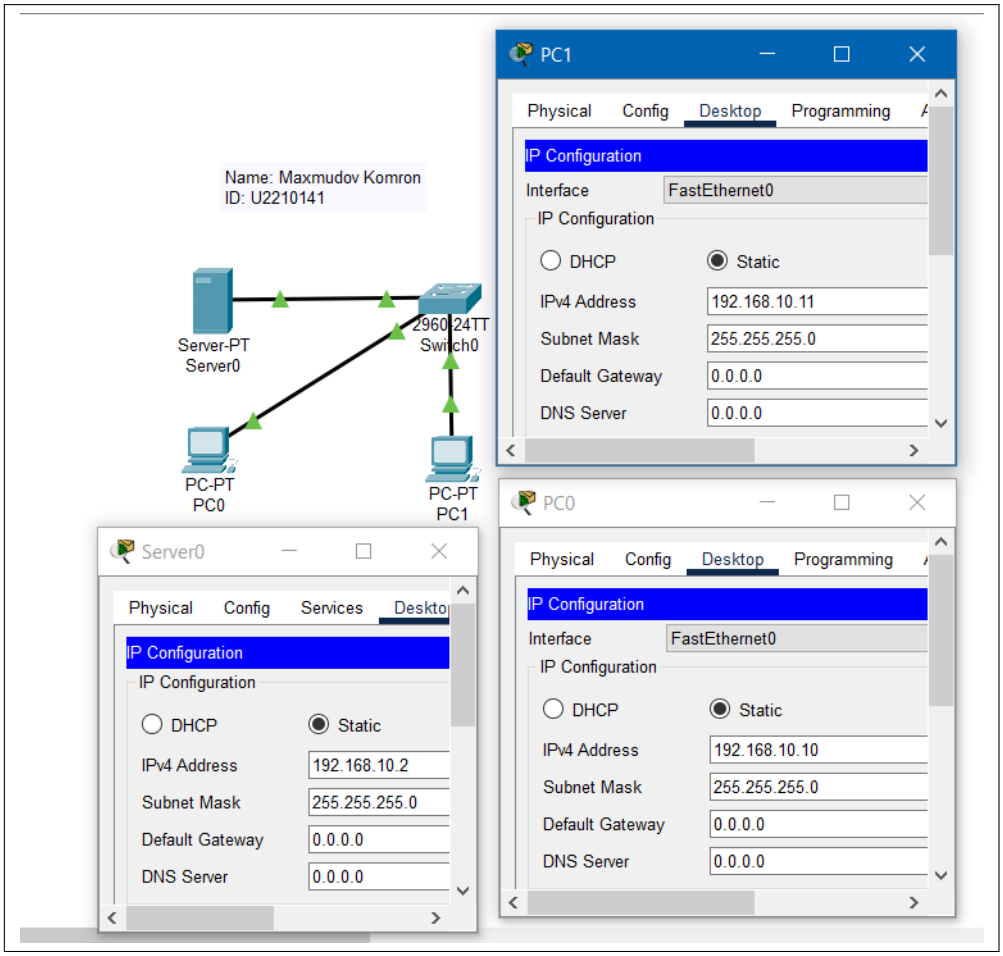
A **File Transfer Protocol (FTP) Server** is used for sharing and transferring files over a network. It allows client devices to connect to a centralized server and perform file operations such as uploading and downloading using FTP commands. In this activity, I simulated an FTP server in Cisco Packet Tracer, where PCs were configured to access and interact with the server.

5.2 Simulation Steps

1. **Created Network Topology and Connected Devices:** I placed a server, a switch, and two PCs in the Cisco Packet Tracer workspace to simulate a simple LAN-based FTP network. I used copper straight-through cables to connect all devices to the switch as follows:
    - Server → Switch
    - PC0 → Switch
    - PC1 → Switch

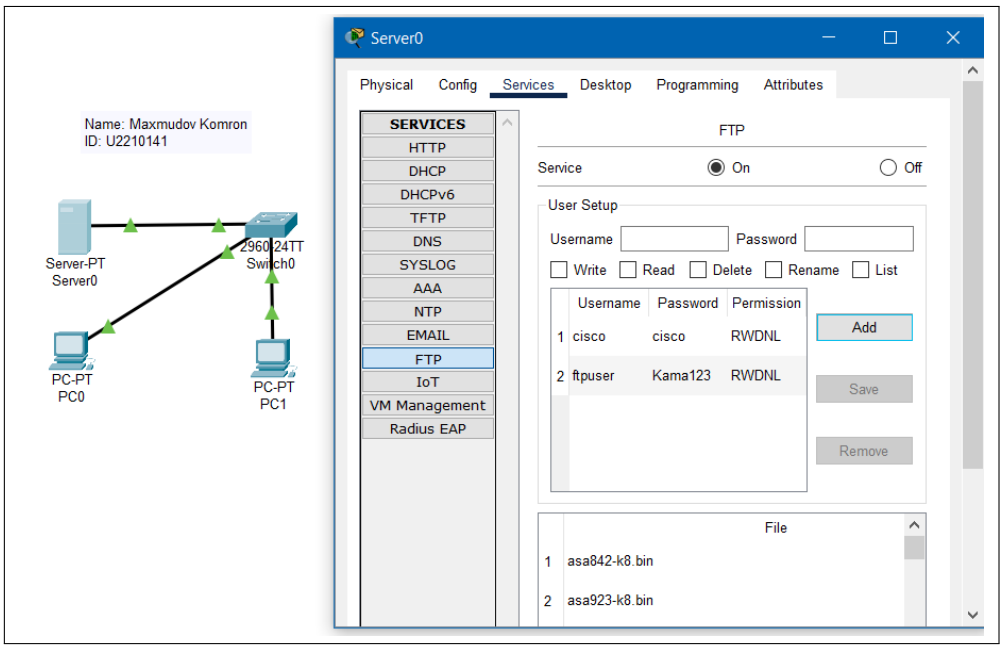


2. **Assigned IP Addresses:** I manually configured IP addresses for all devices in the same subnet:
   - Server: `192.168.10.2`, Subnet Mask: `255.255.255.0`
   - PC0: `192.168.10.10`, Subnet Mask: `255.255.255.0`
   - PC1: `192.168.10.11`, Subnet Mask: `255.255.255.0`



3. **Configured FTP Service on Server:** I opened the `Services` tab on the server, selected `FTP`, turned the service ON, and created a user with login credentials:

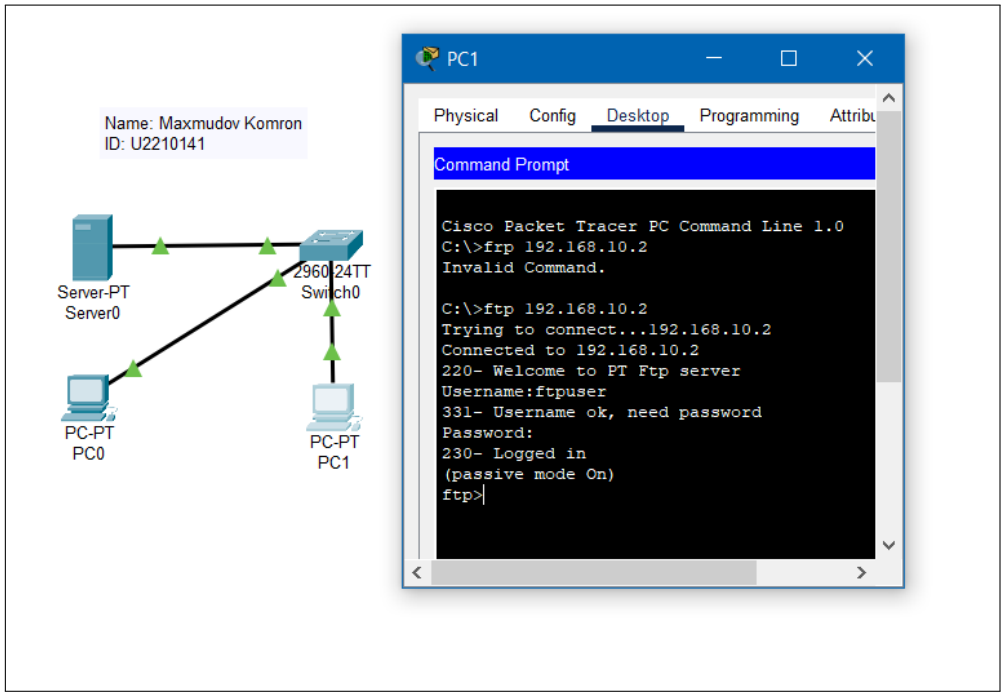
   - Username: `ftpuser`
   - Password: `Kama123`



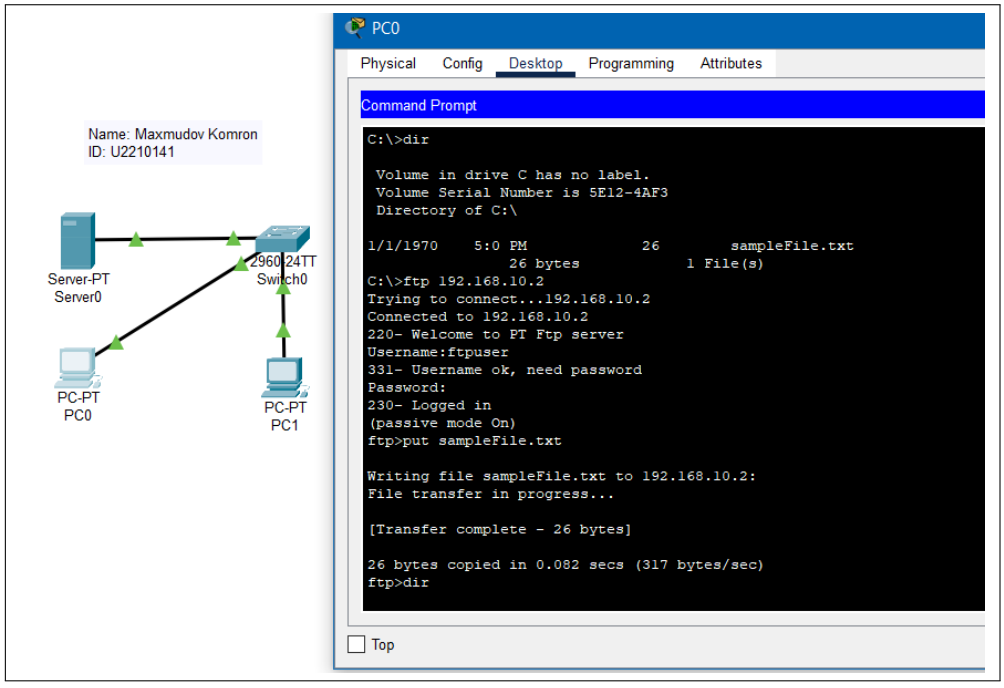
4. **Tested FTP from PC:** I opened the `Command Prompt` on PC0 and typed:

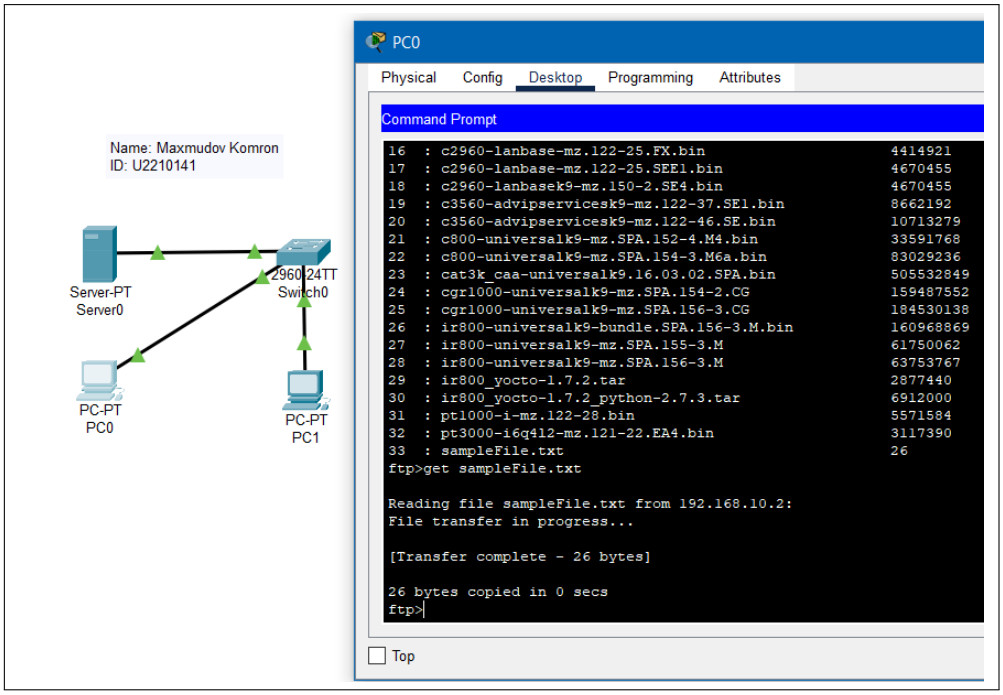
`ftp 192.168.10.2`

Then I logged in using the created username and password to access the server.



5. **Transferred Files Using FTP Commands:** After successful login, I used FTP commands such as:
   - `dir` – to list server files
   - `get filename` – to download files
   - `put filename` – to upload files
   - `bye` – to close the session





5.3 Summary

In this activity, I simulated a basic FTP network in Cisco Packet Tracer using a server and two PCs connected through a switch. After enabling the FTP service and creating user credentials, I tested connectivity and successfully uploaded and downloaded files using FTP commands from the PCs. This activity demonstrated how FTP works in a local network environment for file sharing.

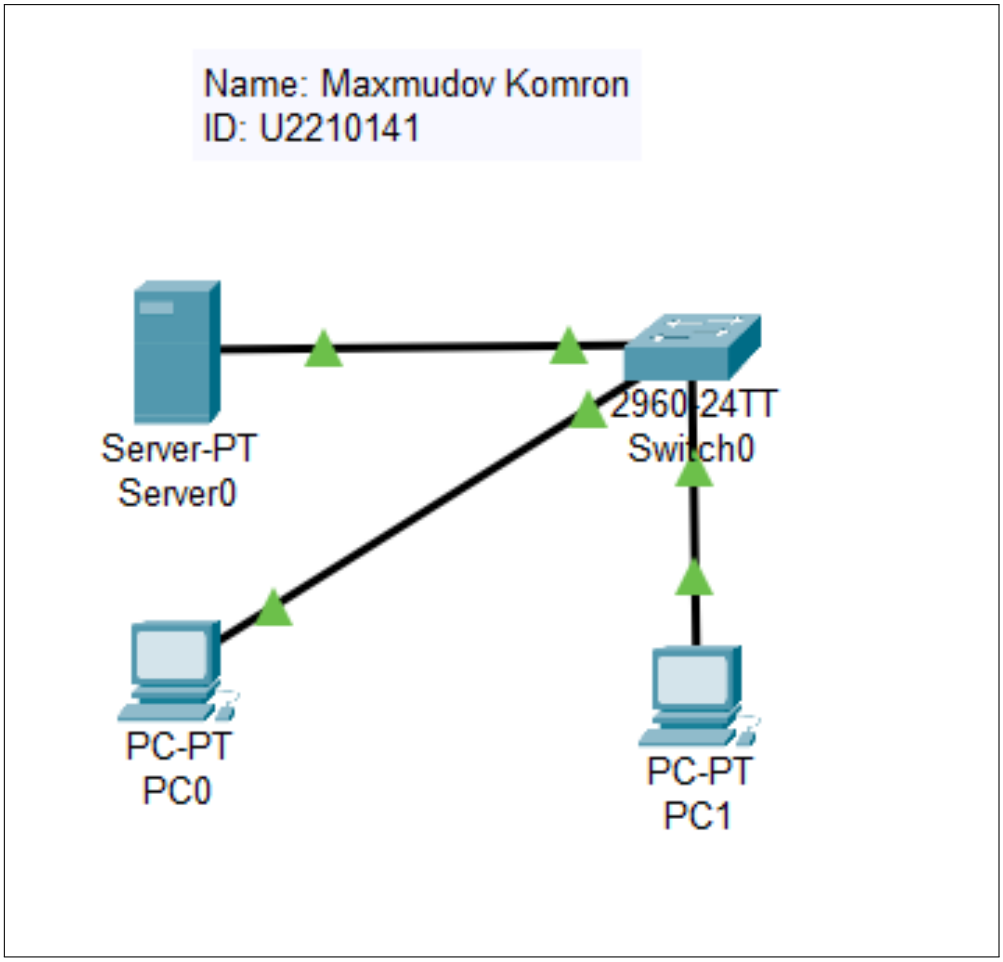
6 Activity 6 – HTTP Web Server Configuration using Cisco Packet Tracer

6.1 Concept Overview

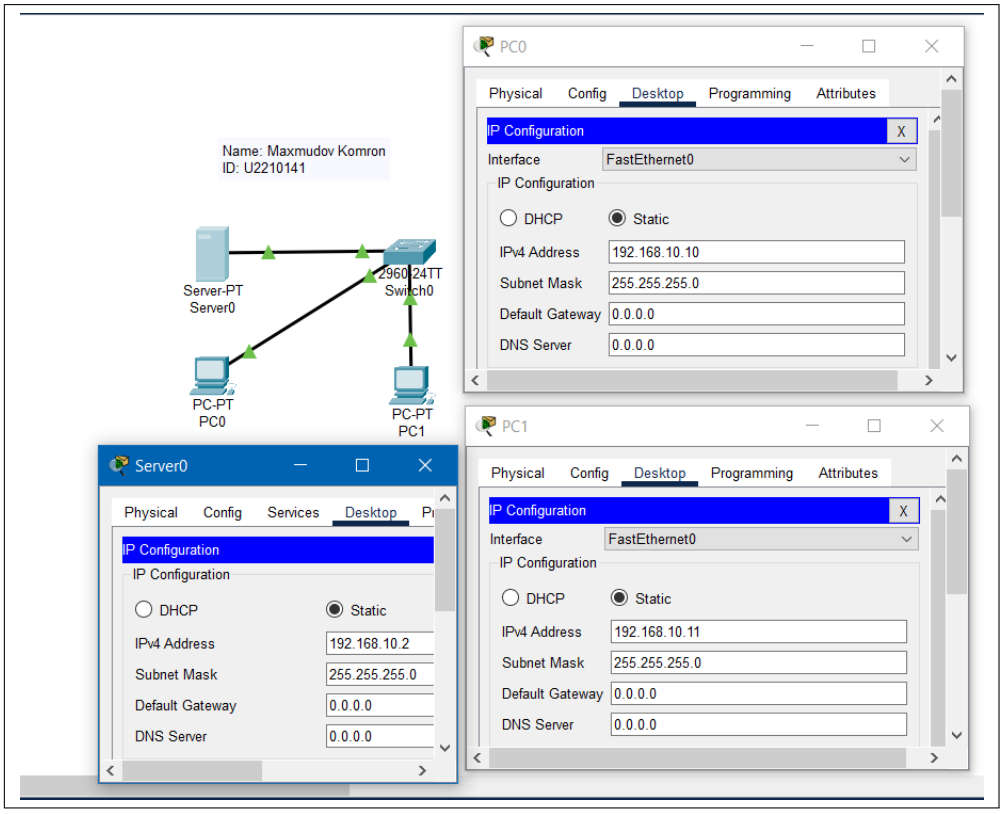
A **Hypertext Transfer Protocol (HTTP) Web Server** is used to host and serve web pages to clients over a network. Web browsers (clients) request web pages from the server using HTTP, and the server responds by delivering the requested content. In this activity, I configured an HTTP web server in Cisco Packet Tracer and accessed it using a web browser from client PCs.

6.2 Simulation Steps

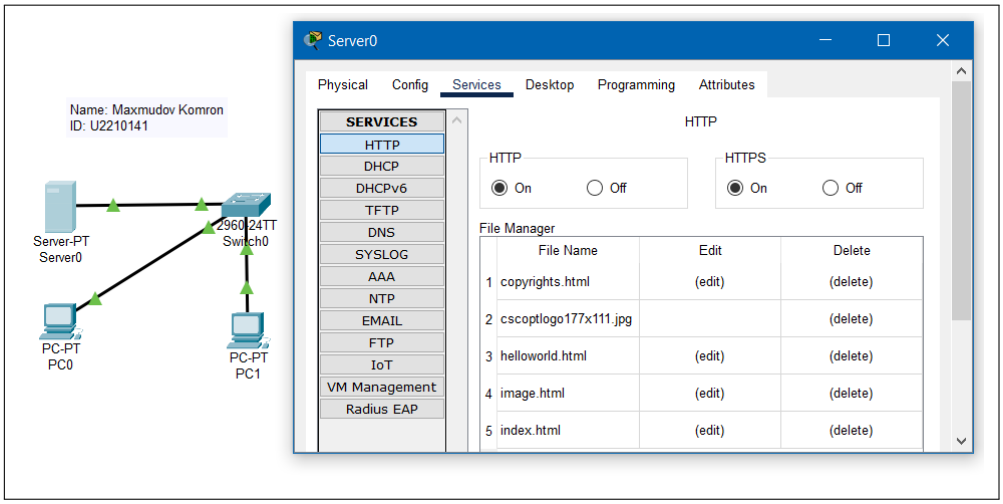
1. **Created Network Topology and Connected Devices:** I placed a server, a switch, and two PCs in the Cisco Packet Tracer workspace to simulate a simple LAN-based web network. Copper straight-through cables were used to connect all devices to the switch:
   - Server → Switch
   - PC0 → Switch
   - PC1 → Switch



2. **Assigned IP Addresses:** Each device was manually assigned an IP address in the same subnet:
   - Server: `192.168.10.2`, Subnet Mask: `255.255.255.0`
   - PC0: `192.168.10.10`, Subnet Mask: `255.255.255.0`
   - PC1: `192.168.10.11`, Subnet Mask: `255.255.255.0`



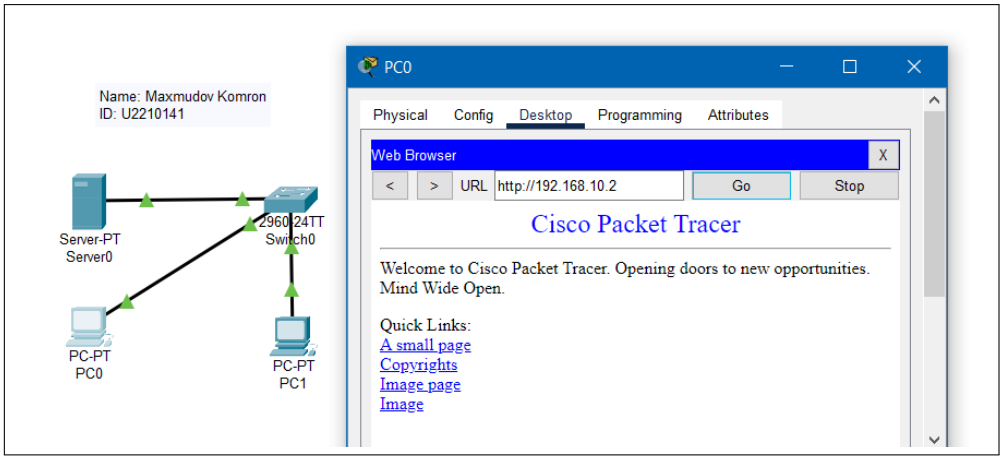
3. **Configured HTTP Service on the Server:** I opened the **Services** tab on the server, selected **HTTP**, and turned the service ON.



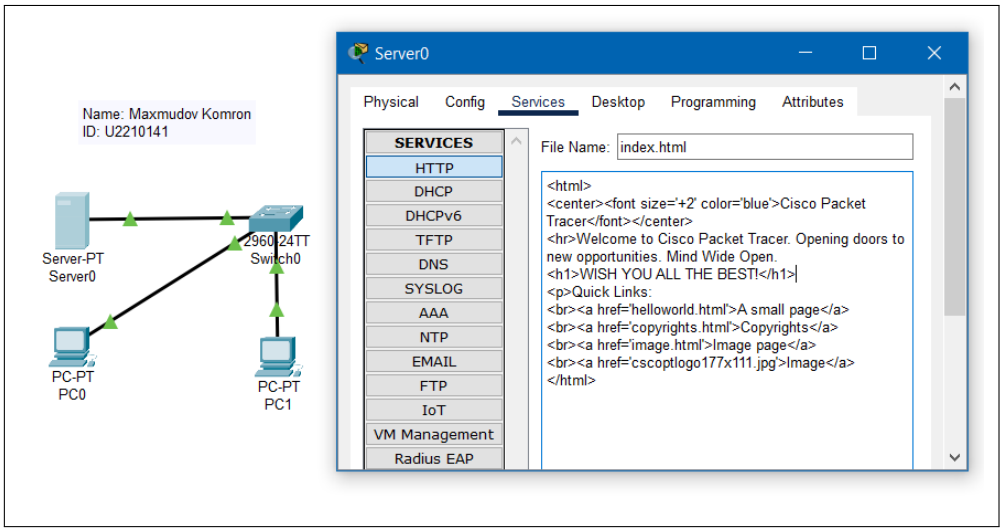
4. **Accessed the Web Page from PC0:** On PC0, I opened the web browser and entered the server's IP address in the URL bar:

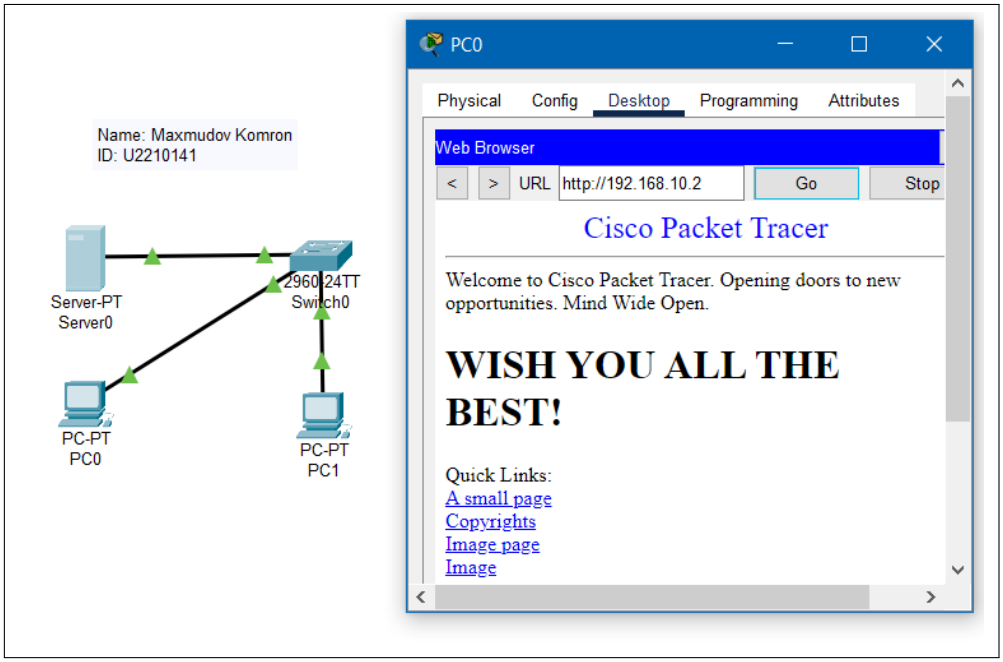
   `http://192.168.10.2`

   The default web page hosted on the server was successfully displayed.



5. **Customized the Web Page:** I modified the default web page by navigating to the **HTTP** service tab on the server and editing the default `index.html` file. After saving the changes, I refreshed the browser on PC0 to verify the update.





6.3 Summary

In this activity, I simulated an HTTP web server in Cisco Packet Tracer and connected two PCs through a switch. After enabling the HTTP service on the server, I accessed the hosted web page from a web browser on PC0. Additionally, I modified the web page's content and verified the changes in real-time. This activity demonstrated how a basic web server operates within a local network.

7 Activity 7 – Subnetting and Network Design in Cisco Packet Tracer

7.1 Concept Overview

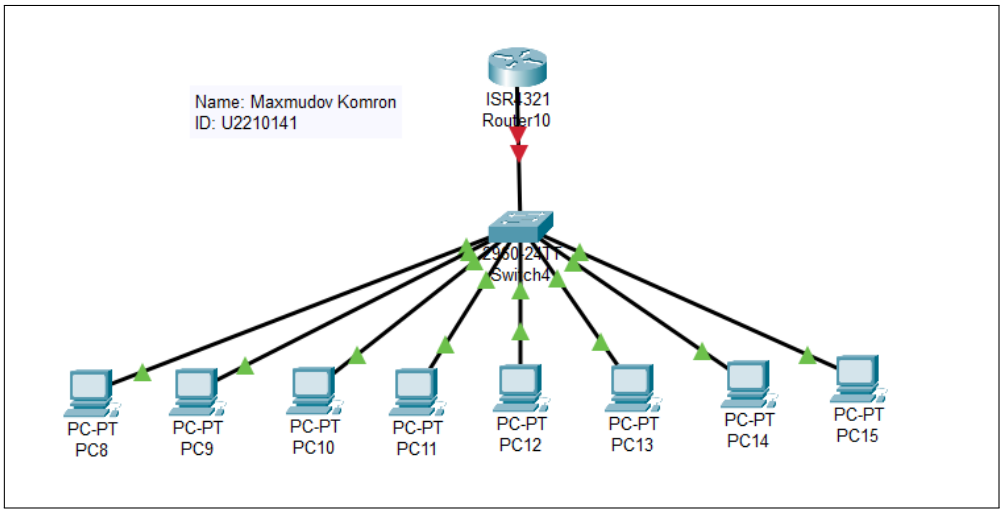
In this activity, I was given the IP block `130.34.12.64/26` and required to divide it into four equal subnetworks. To efficiently implement this design, I used VLAN segmentation and a single-router interface configured with subinterfaces (Router-on-a-Stick). This approach is both scalable and resource-efficient, offering full inter-subnet communication while minimizing physical complexity.

Subnet	Network Address	Range	Broadcast Address
Subnet 1	130.34.12.64/28	130.34.12.65 – 130.34.12.78	130.34.12.79
Subnet 2	130.34.12.80/28	130.34.12.81 – 130.34.12.94	130.34.12.95
Subnet 3	130.34.12.96/28	130.34.12.97 – 130.34.12.110	130.34.12.111
Subnet 4	130.34.12.112/28	130.34.12.113 – 130.34.12.126	130.34.12.127

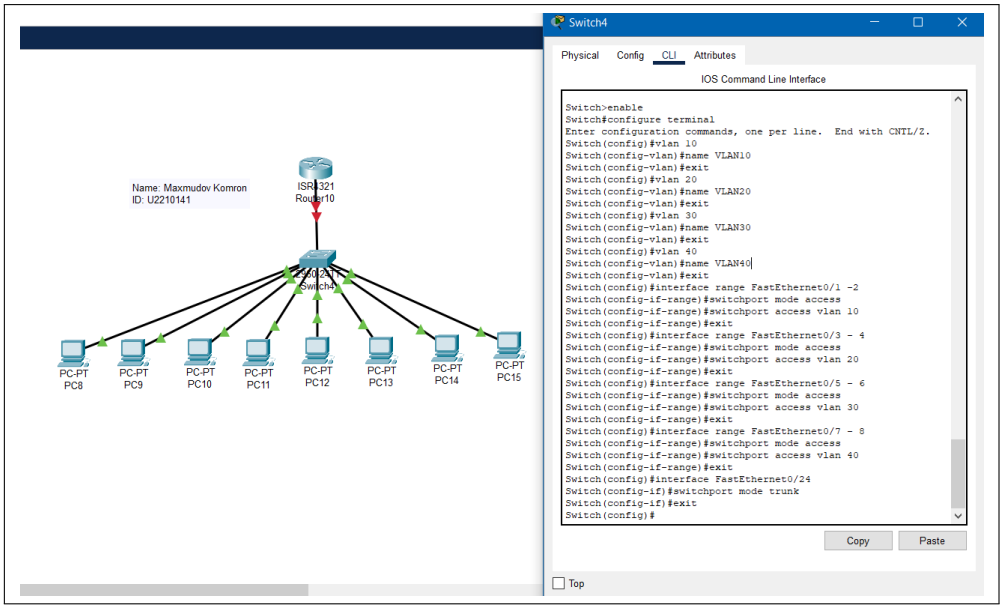
Table 1: Subnet Division of `130.34.12.64/26` into Four `/28` Networks

7.2 Simulation Steps

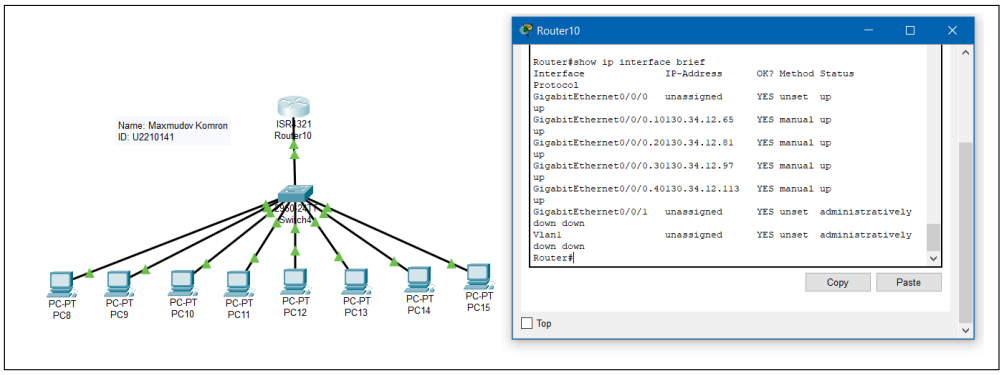
1. **Built the Network Topology:** I placed one router, one switch, and eight PCs (two per subnet). All PCs were connected to the same switch. The switch was then connected to the router's `GigabitEthernet0` interface using a trunk link.



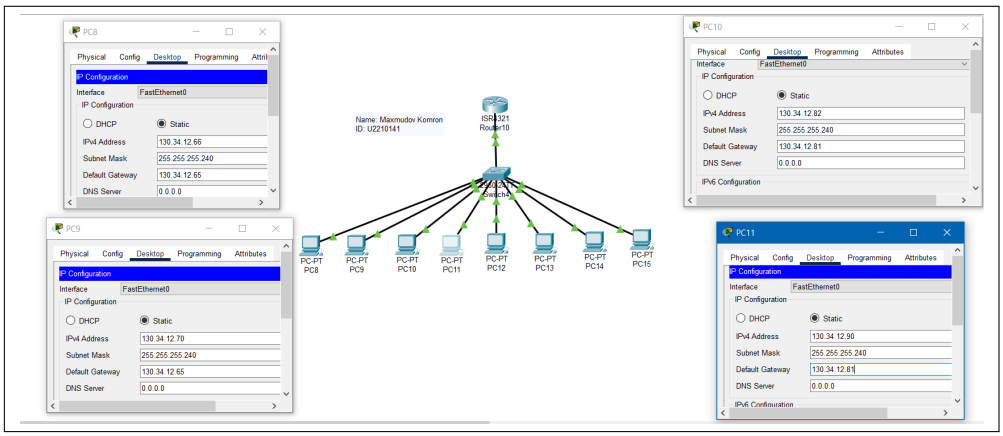
2. **Configured VLANs on the Switch:** I created four VLANs (10, 20, 30, 40), one for each subnet. Each VLAN contained two PCs. Access ports were assigned based on VLAN membership, and the switch port connected to the router was configured as a trunk port to allow VLAN-tagged traffic.



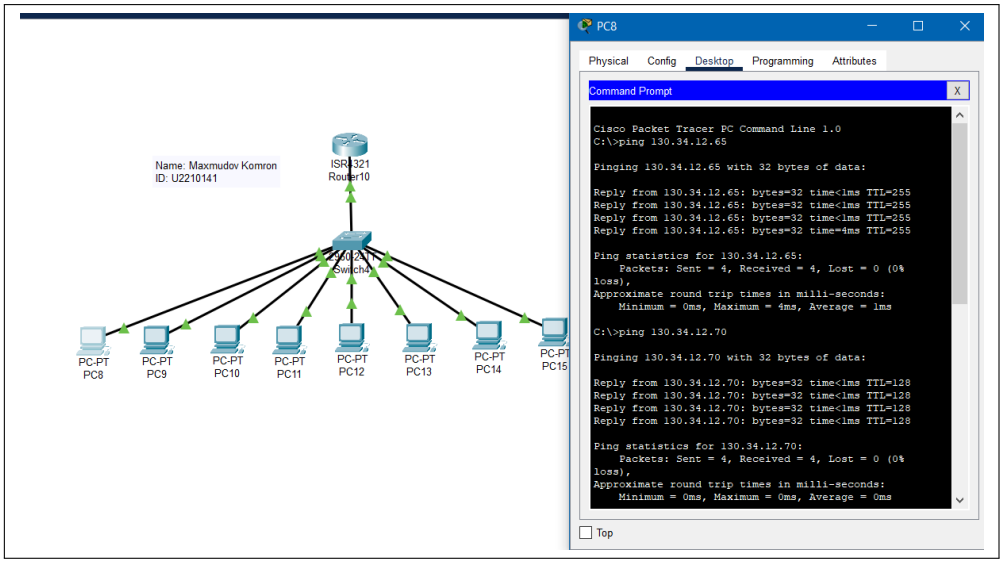
3. **Configured Router Subinterfaces:** I created four subinterfaces under `GigabitEthernet0`, each configured with 802.1Q encapsulation for its corresponding VLAN. Each subinterface was assigned an IP address from its respective subnet, serving as the default gateway for that VLAN.

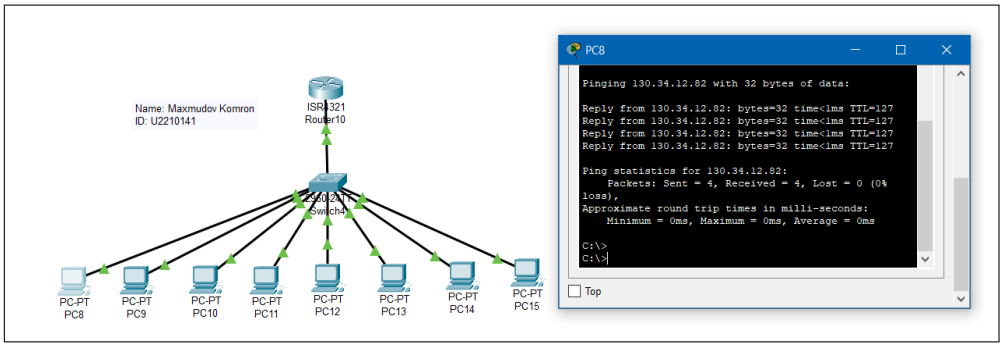


4. **Assigned IP Addresses to PCs:** Each PC was manually configured with an IP address, subnet mask, and default gateway according to its VLAN's subnet. This ensured accurate IP addressing and gateway communication.



5. **Tested Connectivity:** I verified communication within VLANs by pinging between PCs in the same subnet. Then, I tested inter-VLAN routing by pinging across different VLANs. All connectivity tests were successful, demonstrating proper routing and subnetting.





7.3 Summary

In this activity, I subnetted the `/26` network into four equal `/28` subnetworks and implemented a VLAN-based topology using Router-on-a-Stick. This design reduced hardware requirements, centralized routing, and maintained clear separation between subnetworks. The approach allowed efficient IP address management, reliable routing, and ensured all end devices could communicate across subnet boundaries.

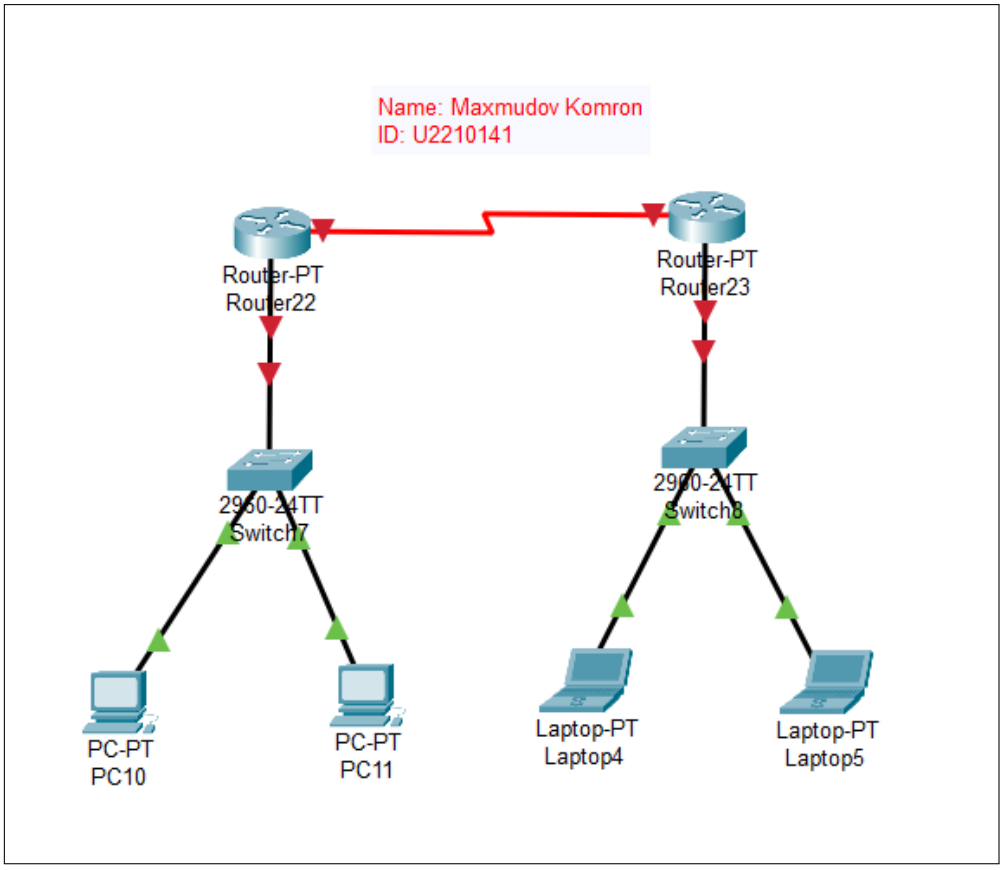
8 Activity 8 – Distance Vector Routing using RIP Protocol

8.1 Concept Overview

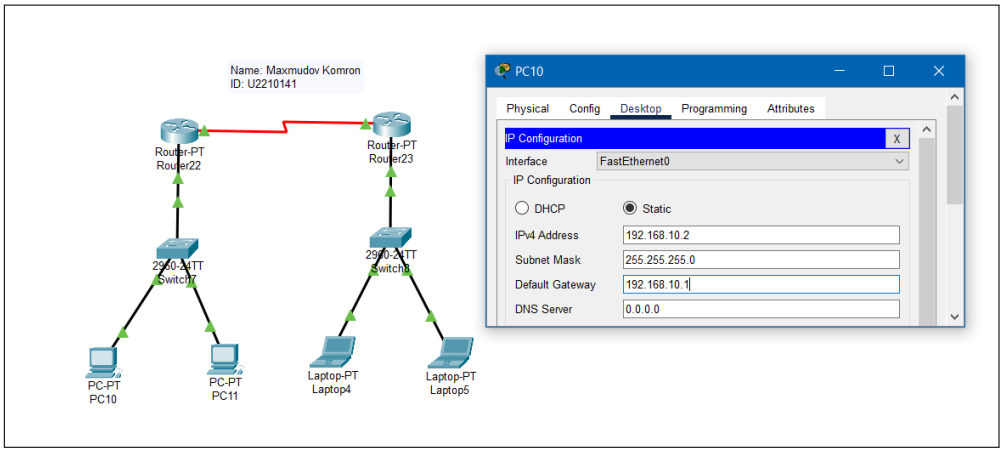
In this activity, I demonstrated the principles of Distance Vector Routing by implementing the Routing Information Protocol (RIP) in a simulated network with two routers. RIP is a dynamic routing protocol that uses hop count as its metric and operates through periodic updates and routing table advertisements. This activity helped reinforce concepts such as dynamic route learning, hop count metrics, and route convergence.

8.2 Simulation Steps

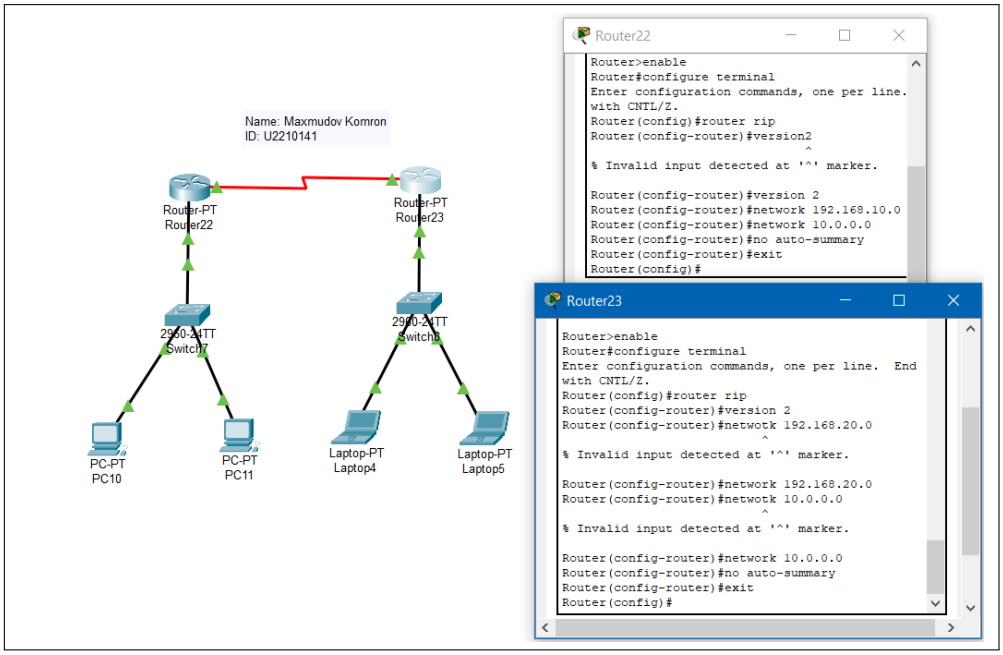
1. **Built the Network Topology:** I created a network with two routers (Router22 and Router23), each connected to its own local network of end devices via a switch. Router22 is connected to PC10 and PC11; Router23 is connected to Laptop4 and Laptop5. The routers are connected through a serial cable.



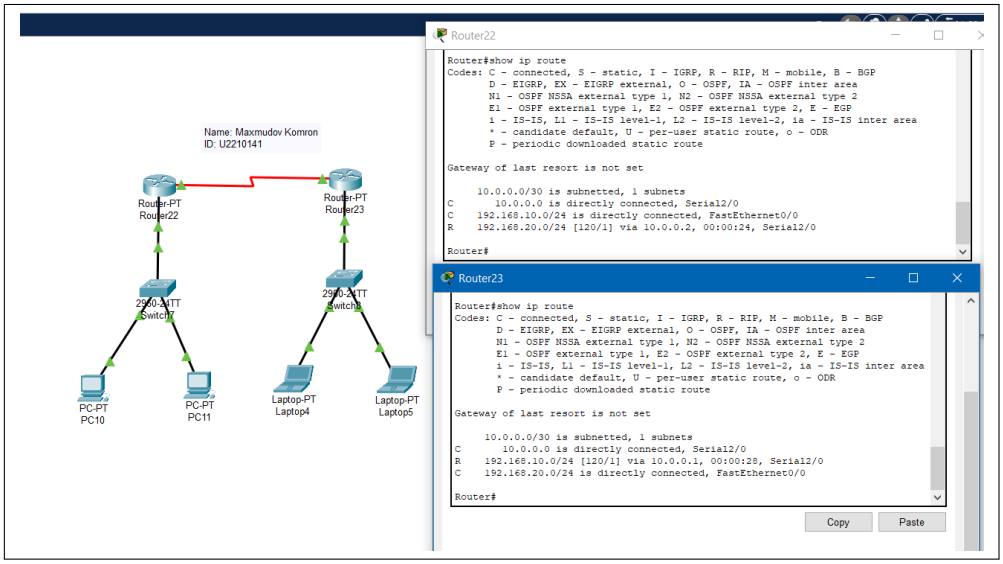
2. **Assigned IP Addresses:** I manually assigned IP addresses to each PC, laptop, and router interface. Each subnet was given a unique IP block. For example, Router22's LAN used 192.168.10.0/24, Router23's LAN used 192.168.20.0/24, and the serial link used 10.0.0.0/30.



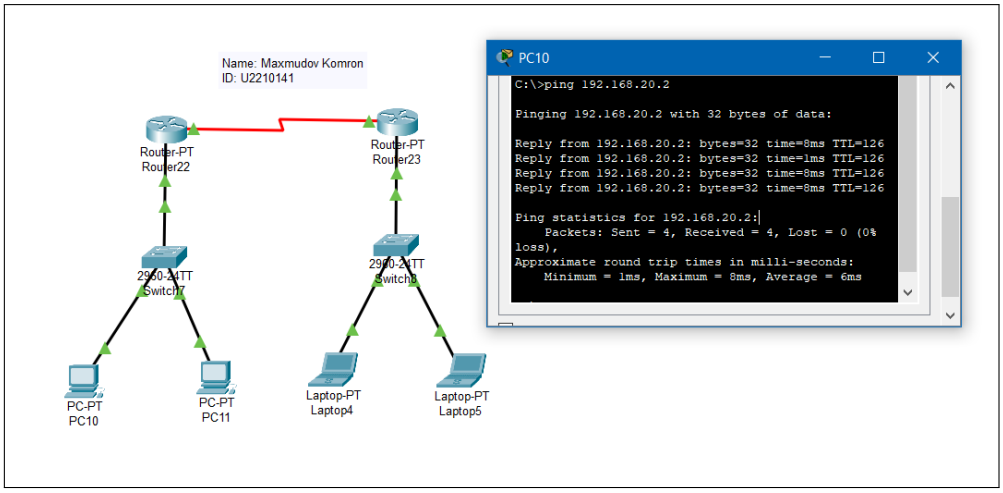
3. **Enabled RIP on Routers:** On both routers, I enabled RIP version 2 and advertised all directly connected networks. This allowed the routers to dynamically learn about remote subnets through periodic updates.



4. **Verified Routing Tables:** I used the `show ip route` command on each router to check if routes to remote networks had been learned via RIP. The presence of RIP-learned routes confirmed correct configuration.



5. **Tested Connectivity:** I tested connectivity by pinging from PC10 (in Router22's network) to Laptop4 (in Router23's network). Successful pings confirmed that the RIP protocol propagated the necessary routes.



8.3 Summary

In this activity, I implemented RIP routing between two routers to allow communication between two LANs. RIP enabled dynamic route exchange, removing the need for manual static routes. I verified the successful configuration by checking routing tables and testing end-to-end connectivity. This hands-on task demonstrated the functionality of RIP's distance vector mechanism, hop count limitation, and automatic route updates across routers.

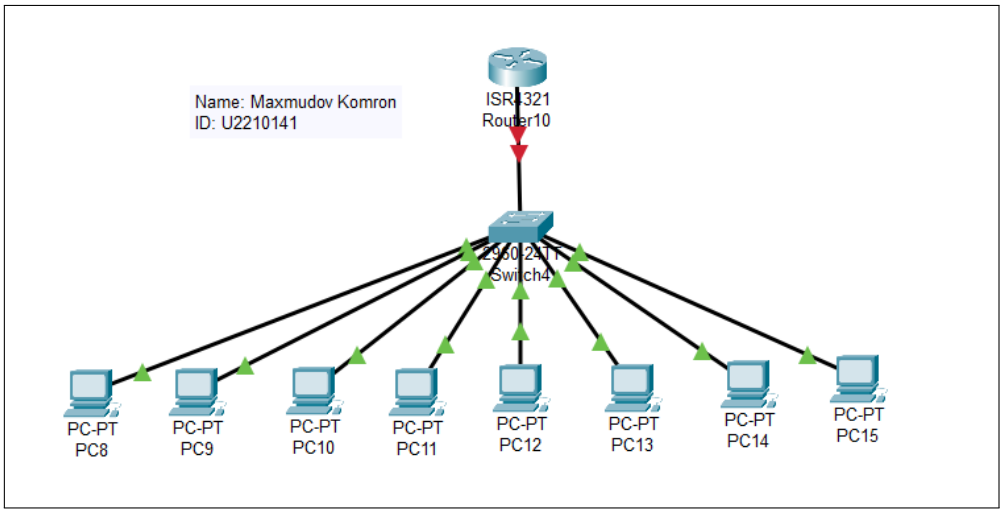
9 Activity 9 – Configuring VLAN and Inter Routing VLAN setup

9.1 Concept Overview

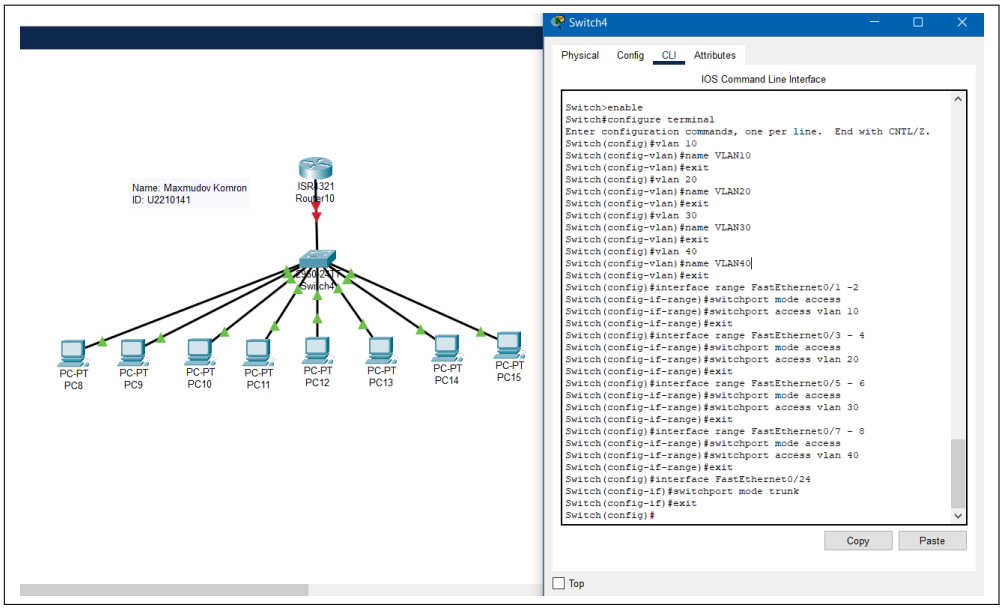
In this activity, I implemented VLAN segmentation and Inter-VLAN routing in Cisco Packet Tracer. VLANs provide logical separation of devices within a switch, improving security and traffic control. Since devices in different VLANs cannot communicate by default, I used a Router-on-a-Stick configuration to enable inter-VLAN communication through router subinterfaces. This design supports centralized routing while maintaining VLAN boundaries.

9.2 Simulation Steps

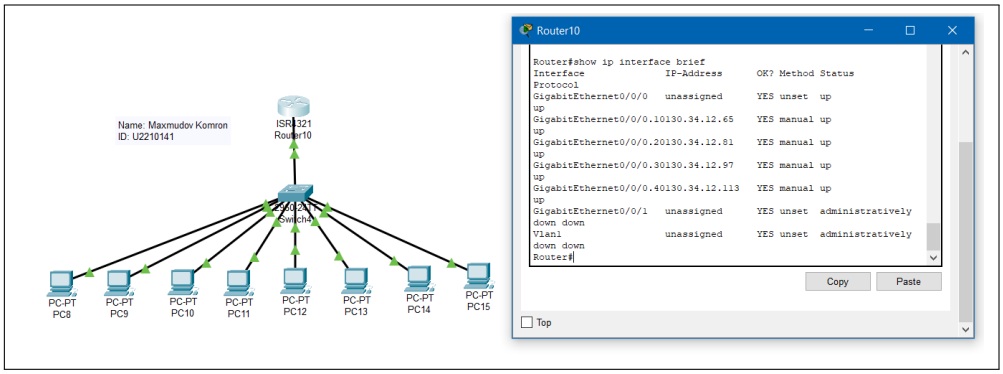
1. **Built the Network Topology:** I used one router, one switch, and eight PCs (two per VLAN). All PCs were connected to the same switch, which was linked to the router's `GigabitEthernet0` interface via a trunk connection.



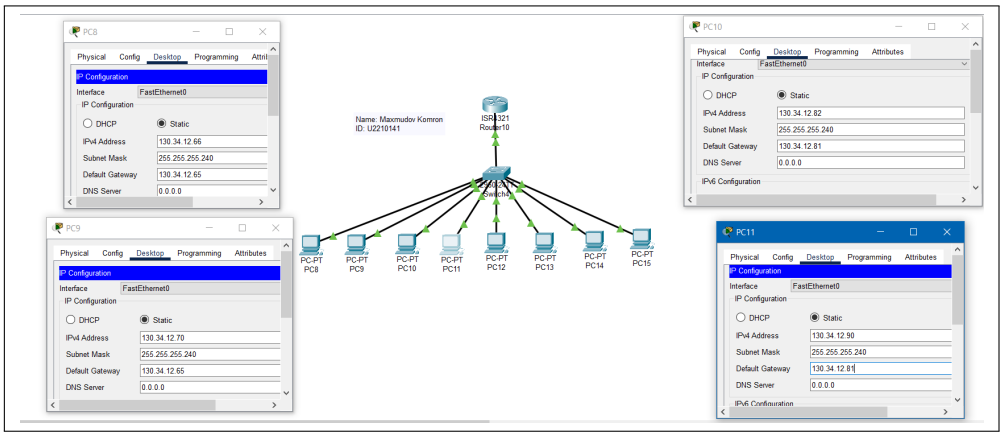
2. **Configured VLANs on the Switch:** I created four VLANs (10, 20, 30, 40) and assigned each pair of PCs to a separate VLAN. I also configured access ports for end devices and set the router-facing port as a trunk to carry VLAN-tagged traffic.



3. **Configured Router Subinterfaces:** I created four subinterfaces under `GigabitEthernet0`, each mapped to a VLAN using 802.1Q encapsulation. Each subinterface was assigned a gateway IP address for its VLAN.

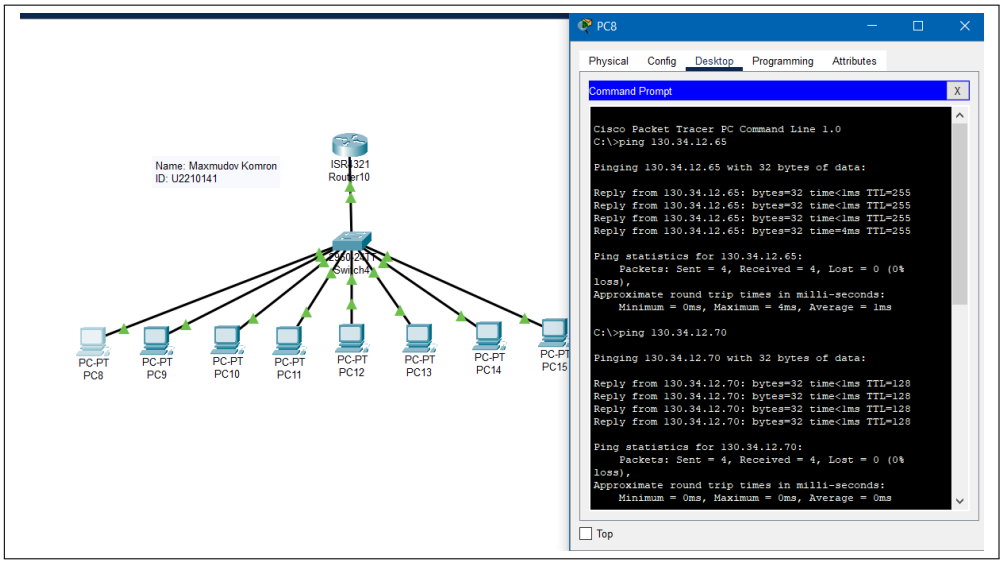


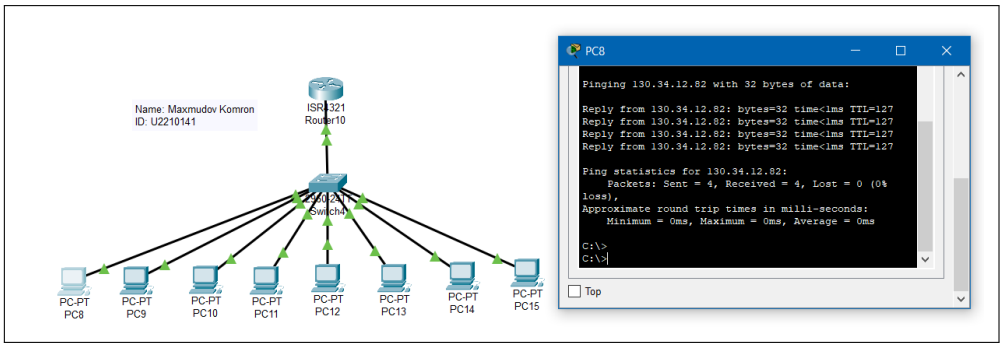
4. **Assigned IP Addresses to PCs:** Each PC was configured with a static IP address, subnet mask, and default gateway that matched its VLAN. This ensured proper addressing and gateway communication.



5. **Tested Inter-VLAN Connectivity:** I verified communication within each VLAN by pinging be-

tween same-VLAN PCs. Then, I tested inter-VLAN routing by pinging across different VLANs. All tests were successful.





9.3 Summary

In this activity, I used VLANs to segment the network and configured a Router-on-a-Stick setup to allow inter-VLAN communication. This approach centralized routing while maintaining logical separation between devices. The final setup demonstrated successful VLAN isolation and routing, suitable for scalable network designs.

10 Activity 10 – Comprehensive Network Design for a Two-Floor Office Building

10.1 Project Overview

This activity involved the design, simulation, and documentation of a network for a two-story office building. The objective was to ensure secure, segmented, and efficient connectivity across various functional areas, considering both wired and wireless access. I approached this task by analyzing floor-wise needs, determining access policies, and selecting suitable network devices and topology. The final design emphasizes practical implementation, security, scalability, and cost-efficiency.

10.2 Site Layout and Requirements

First Floor

- **Reception Area:** Requires wired internet access for handling queries and general business operations.
- **Coffee Shop:** Contains two isolated computers with no internet access. Intended for offline note preparation or internal reference.
- **Waiting Room:** Offers wireless internet access to guests and visitors via a dedicated access point.

Second Floor

- **Library:** Includes three desktop computers and a network printer. This zone is restricted from internet access and intended for internal resource sharing.
- **Office Rooms:** Equipped with desktops for employees. Devices are allowed internet access and inter-office communication. Wi-Fi is also provided for personal or mobile use.

10.3 Design Strategy and Decisions

I adopted a **hybrid hierarchical design** with both wired and wireless segments. VLANs were implemented for logical separation of departments and access policies. A single router serves as the central gateway and applies ACLs to enforce internet restrictions. The design ensures that:

- Internal zones (e.g., Library, Coffee Shop) are fully isolated.
- Office zones have internet and internal access.
- Visitor access is separated via a guest VLAN with Wi-Fi.
- Devices are grouped by function and location for management ease.

10.4 VLAN Plan and IP Addressing

VLAN ID	Zone	Subnet	Purpose
10	Office Network	192.168.10.0/24	Internet + Intranet access
20	Library	192.168.20.0/24	Intranet only, printer sharing
30	Reception	192.168.30.0/24	Business internet access
40	Coffee Shop	192.168.40.0/24	Isolated, offline access only
50	Guest Wi-Fi	192.168.50.0/24	Internet access only, no LAN reach

Table 2: VLAN and Subnet Design

10.5 Access Control and Security Rules

Access Control Lists (ACLs) were configured on the router to restrict or permit traffic based on zone policies:

- VLAN 10 (Office) – Full internet and internal access
- VLAN 20 (Library) – No internet; access to printer only
- VLAN 30 (Reception) – Internet only; no internal access
- VLAN 40 (Coffee Shop) – No external or internal access
- VLAN 50 (Guest Wi-Fi) – Internet only; isolated from internal VLANs

10.6 Physical and Logical Design Summary

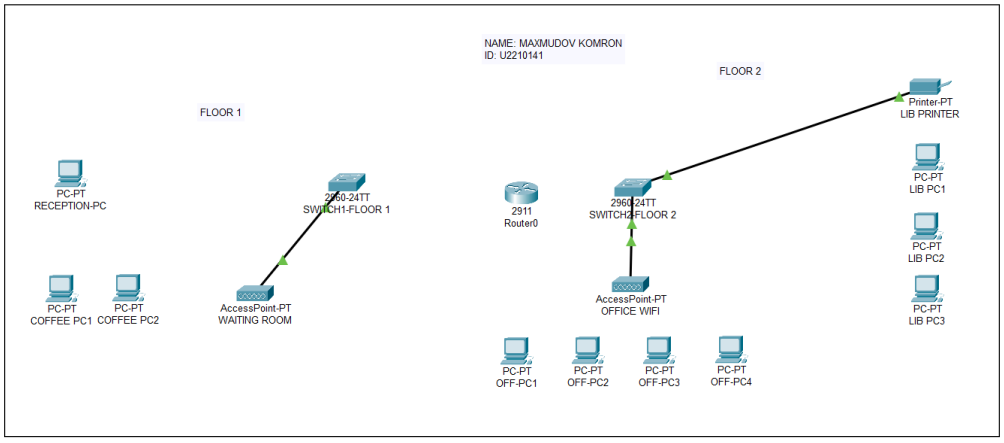
- **Router:** Acts as the default gateway for all VLANs and manages ACL enforcement.
- **Switches:** Two 24-port managed switches provide connectivity across floors.
- **Access Points:** One per floor, connected via trunk port for VLAN tagging.
- **Printer:** Connected to library VLAN and shared with library PCs.

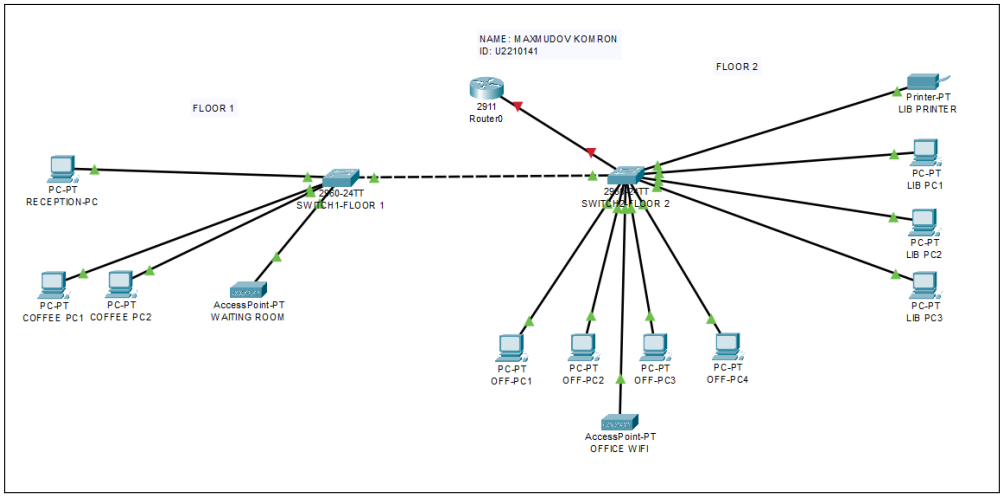
10.7 Cost Estimation and Device Breakdown

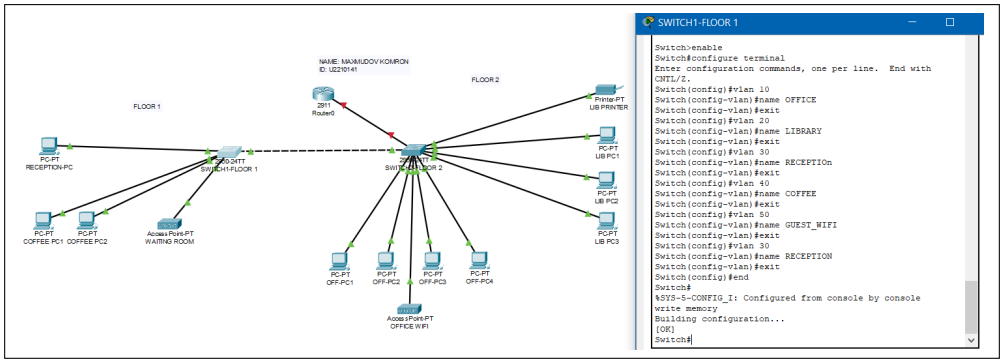
Item	Quantity	Cost (USD)
Cisco Router (ISR Series)	1	350
24-port Managed Switch	2	400
Access Points (Dual-Band)	2	200
Desktop Computers	10	6000
Network Printer	1	150
Ethernet Cables and Patch Panels	1 set	100
UPS Units	2	300
Rack Mount	1	120
Total Estimated Cost		**6,620 USD**

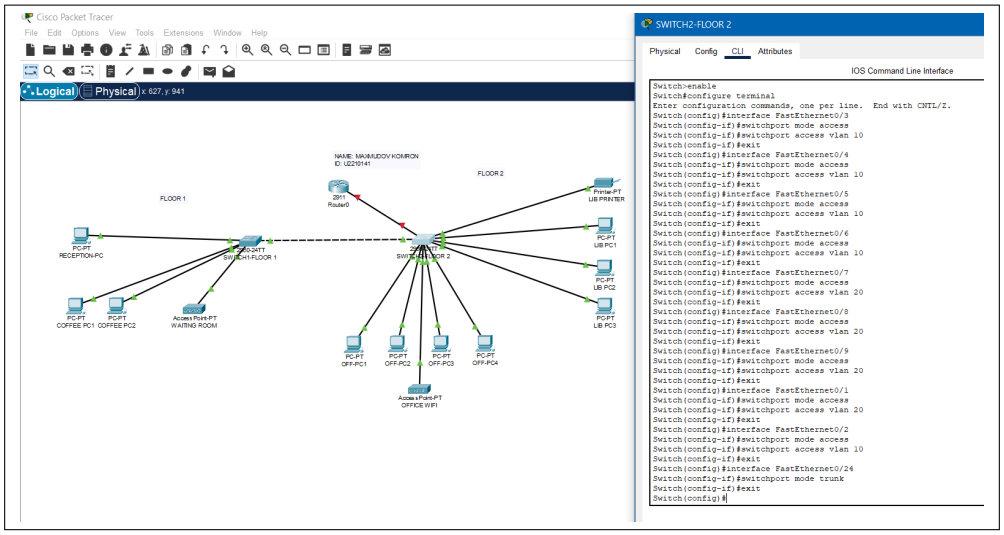
Table 3: Table 3: Tentative Cost Estimation

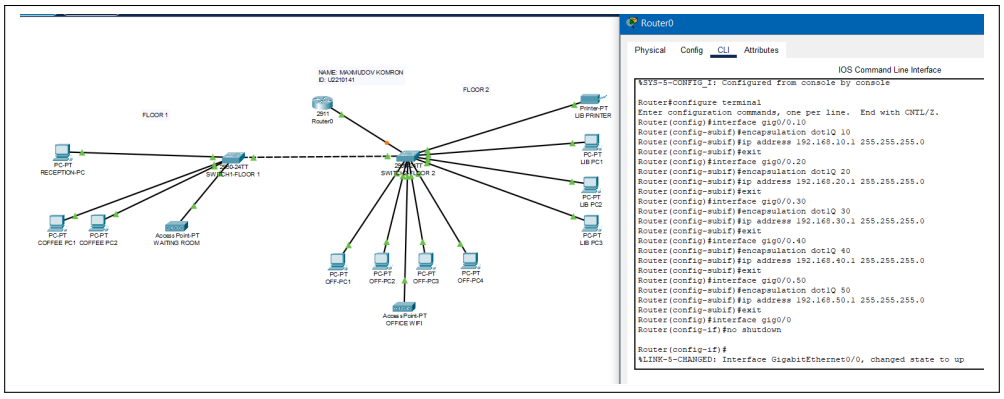
10.8 Simulation

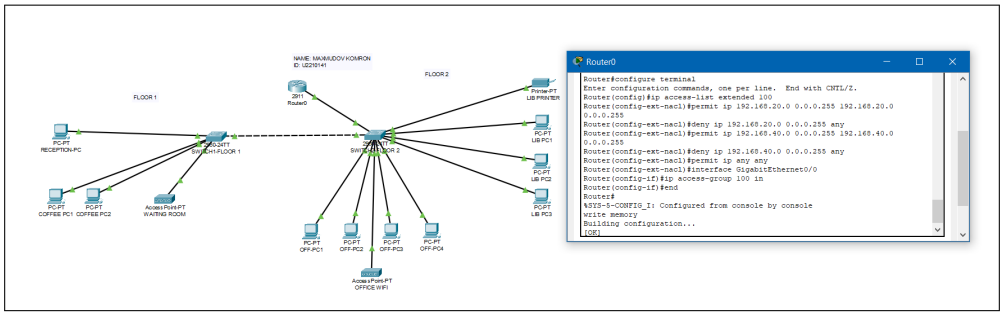


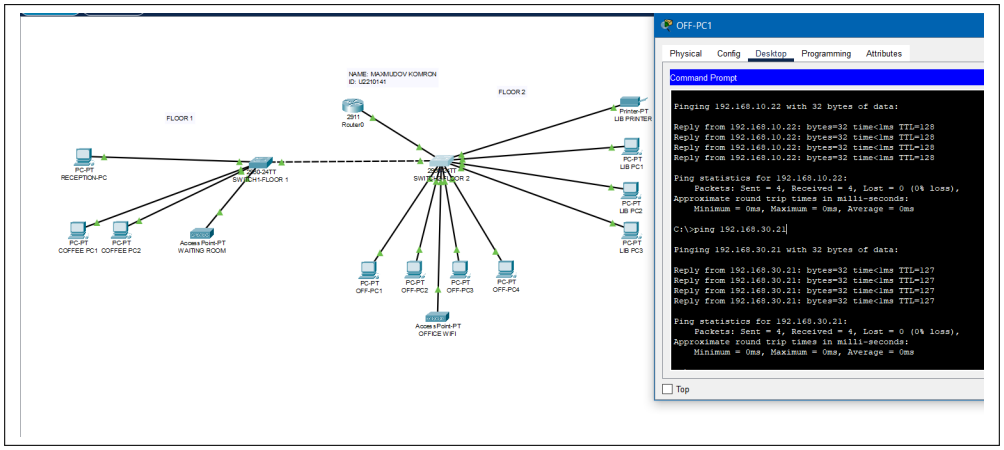



SWITCH1-FLOOR 1
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#name OFFICE
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name LIBRARY
Switch(config-vlan)#exit
Switch(config)#vlan 30
Switch(config-vlan)#name RECEPTIOn
Switch(config-vlan)#exit
Switch(config)#vlan 40
Switch(config-vlan)#name COFFEE
Switch(config-vlan)#exit
Switch(config)#vlan 50
Switch(config-vlan)#name GUEST_WIFI
Switch(config-vlan)#exit
Switch(config)#vlan 30
Switch(config-vlan)#name RECEPTION
Switch(config-vlan)#exit
Switch(config)#end
Switch#
%SYS-5-CONFIG_I: Configured from console by console
write memory
Building configuration...
[OK]
Switch#


Cisco Packet Tracer
SWITCH2-FLOOR 2
Physical Config CLI Attributes
IOS Command Line Interface
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface FastEthernet0/3
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#interface FastEthernet0/4
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#interface FastEthernet0/5
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#interface FastEthernet0/6
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#interface FastEthernet0/7
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#interface FastEthernet0/8
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#interface FastEthernet0/9
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#interface FastEthernet0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#interface FastEthernet0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#interface FastEthernet0/24
Switch(config-if)#switchport mode trunk
Switch(config-if)#exit
Switch(config)#


Router0
Physical Config CLI Attributes
IOS Command Line Interface
%SYS-5-CONFIG_I: Configured from console by console
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface gig0/0.10
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface gig0/0.20
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface gig0/0.30
Router(config-subif)#encapsulation dot1Q 30
Router(config-subif)#ip address 192.168.30.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface gig0/0.40
Router(config-subif)#encapsulation dot1Q 40
Router(config-subif)#ip address 192.168.40.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface gig0/0.50
Router(config-subif)#encapsulation dot1Q 50
Router(config-subif)#ip address 192.168.50.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface gig0/0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up


Router0
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip access-list extended 100
Router(config-ext-nacl)#permit ip 192.168.20.0 0.0.0.255 192.168.20.0 0.0.0.255
Router(config-ext-nacl)#deny ip 192.168.20.0 0.0.0.255 any
Router(config-ext-nacl)#permit ip 192.168.40.0 0.0.0.255 192.168.40.0 0.0.0.255
Router(config-ext-nacl)#deny ip 192.168.40.0 0.0.0.255 any
Router(config-ext-nacl)#permit ip any any
Router(config-ext-nacl)#interface GigabitEthernet0/0
Router(config-if)#ip access-group 100 in
Router(config-if)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console
write memory
Building configuration...
[OK]



10.9 Summary and Reflections

Requirement	Implementation	Fulfilled
Employees on second floor can access internet via Wi-Fi and communicate internally	Office VLAN (10) allows both internet and intra-office communication; dedicated AP provided for employees	Yes
Reception desk has internet access for managing business queries	Reception PC connected to VLAN 30 with internet access permitted via ACL	Yes
Library computers can share files and access a printer; no internet access	VLAN 20 configured for Library, ACL blocks internet access, allows internal communication and printer use	Yes
Waiting room has wireless internet for visitors	Guest AP configured on VLAN 50; isolated from internal network using ACLs	Yes
Coffee shop has two isolated computers with no internet or internal access	VLAN 40 applied; ACL denies all traffic outside VLAN 40, isolating the devices completely	Yes

Table 4: Requirement Fulfillment Matrix

This project provided a real-world simulation of designing a structured, secure, and functional office network across multiple zones. By applying VLAN segmentation, ACLs, and a layered design, I was able to meet all requirements efficiently. The use of a hybrid model (wired + wireless) ensured flexibility, and the simulation verified full connectivity, isolation, and access control. This activity helped solidify concepts in network planning, implementation, and policy-based security.