



Faculty of Engineering and Technology
Electrical and Computer Engineering Department
ENCS4130 – Computer Network Laboratory

Experiment No. 06

DHCP, DNS, Email, and Web Server Configuration

■ Objectives

- (a) Learn how to configure a DHCP server.
- (b) Learn how to configure a DNS server.
- (c) Learn how to configure a Web server.
- (d) Learn how to configure an Email server.

■ Requirements

- (a) Cisco Packet Tracer software.
- (b) Two routers.
- (c) Four PCs.
- (d) Three switches.
- (e) Four servers.
- (f) One network sniffer.

6.1 Introduction

In modern networks, several core services work together to enable seamless communication and access to resources. One of the key components is the *Dynamic Host Configuration Protocol (DHCP)*, which automates the process of assigning *IP addresses* and network configuration settings to devices. This allows hosts to connect to the network and use essential services without manual setup. *DHCP* also enables access to services that rely on *IP-based communication*, including the *Domain Name System (DNS)*, the *Network Time Protocol (NTP)*, and applications based on the *User Datagram Protocol (UDP)* or *Transmission Control Protocol (TCP)*.

The *DNS* acts as the Internet's phonebook by translating human-friendly domain names (such as *birzeit.edu* or *nytimes.com*) into machine-readable *IP addresses*, enabling web browsers and other services to locate and connect to servers efficiently.

A *web server* is a combination of software and hardware designed to deliver web content over the Internet using the *Hypertext Transfer Protocol (HTTP)* or its secure version, *HTTPS*. Similarly, an *email server* is responsible for sending, receiving, and storing email messages. It typically uses the *Simple Mail Transfer Protocol (SMTP)* for sending emails and the *Post Office Protocol (POP3)* or the *Internet Message Access Protocol (IMAP)* for retrieving them. Secure communication is often ensured through encryption protocols such as *SSL/TLS*.

In this experiment, we will configure and simulate *DHCP*, *DNS*, *web*, and *email servers* using *Cisco Packet Tracer*. The following sections will explain each service and demonstrate how to set them up effectively within a *virtual network environment*.

6.2 Dynamic Host Configuration Protocol (DHCP)

DHCP is a system for assigning *IP addresses* to each network device (known as a *host*) on an organization's network. A *host* may be a desktop computer, a laptop, a tablet, a mobile device, a thin client, or other types of device. Each host must have an *IP address* to communicate with other devices over the Internet. The *DHCP* network protocol assigns addresses automatically, rather than requiring network administrators to make manual assignments. *DHCP* is also responsible for automatically assigning new *IP addresses* when devices move to new locations on the network. In addition to *IP addresses*, the *DHCP* service assigns configuration parameters such as *DNS addresses*, *subnet masks*, and *default gateways* that are essential to network communications.

6.2.1 How does DHCP work?

DHCP operates at the application layer and uses the UDP transport protocol. Its primary function is to dynamically assign IP addresses and provide essential TCP/IP configuration information to clients. DHCP is a client-server protocol; it uses port 67 for the server and port 68 for the client. When a new host joins the network, the DHCP process follows a four-step sequence, as shown in Figure 1:

1. **DHCP Discover:** The DHCP client broadcasts a DHCP Discover message to locate available DHCP servers. This message is sent with a destination IP address of 255.255.255.255 (broadcast) and a source IP address of 0.0.0.0 (since the

client has no IP address yet). The message is encapsulated in a datagram and sent over the link layer to reach all nodes on the subnet.

2. **DHCP Offer:** A DHCP server that receives the discover message responds with a DHCP Offer message. This message is also broadcast using 255.255.255.255 and contains the transaction ID from the discover message, a proposed IP address for the client, subnet mask, lease duration (the amount of time for which the IP address will be valid), and other configuration parameters.
3. **DHCP Request:** The client responds to the selected offer with a DHCP Request message, indicating acceptance of the proposed configuration. This message is also broadcast to inform other DHCP servers that their offers were declined.
4. **DHCP Acknowledgment (ACK):** The selected DHCP server replies with a DHCP ACK message, confirming the IP address assignment and finalizing the configuration. Once this message is received, the client can begin using the assigned IP address for the duration of the lease. If needed, the client can later request a lease renewal to extend the IP address usage.

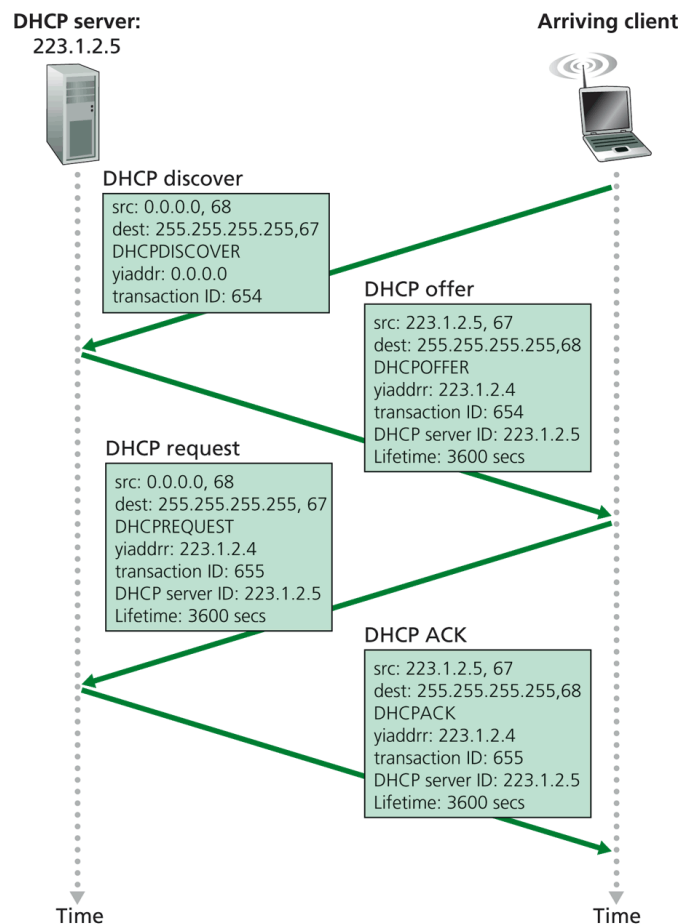


Figure 1: DHCP client-server interaction.

6.2.2 Benefits of Using DHCP

- DHCP automates and simplifies the assignment of IP addresses, reducing the administrative burden on network managers.

- By dynamically allocating and reusing IP addresses, DHCP helps optimize address usage, reducing the total number of addresses required.
- Organizations can modify IP address schemes seamlessly without disrupting end users, making network transitions smoother.
- DHCP minimizes human errors by centralizing and automating IP address assignments, preventing duplicate addresses and misconfigurations.

6.3 Domain Name System (DNS)

DNS translates human-readable *domain names* into *IP addresses*, which browsers use to load web pages. Every device connected to the Internet has a unique *IP address*, which other devices use to communicate with it. *DNS* eliminates the need for users to memorize complex numerical *IP addresses* by allowing them to enter easy-to-remember *domain names* instead.

6.3.1 DNS Server

A *DNS server* is a computer system that maintains a database of public *IP addresses* and their associated *domain names*. When a user enters a *domain name*—such as `google.com`—into a web browser, the *DNS server* resolves that name to its corresponding *IP address* (i.e., `142.250.72.206`). This *IP address* directs the user's device to the correct server hosting the website. *DNS* functions much like a phonebook for the Internet, enabling efficient and user-friendly access to websites.

6.3.2 Distributed, Hierarchical DNS

The *DNS infrastructure* is distributed and hierarchical, consisting of many servers worldwide. No single server holds all *domain name-to-IP address* mappings. Instead, the information is divided and managed across different types of *DNS servers*, as shown in Figure 2. The three main classes of *DNS servers* are:

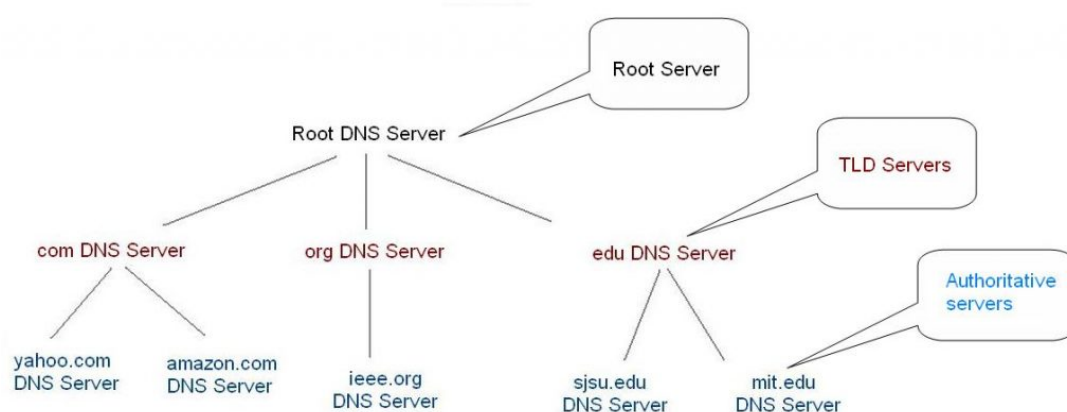


Figure 2: DNS server hierarchy.

- **Root DNS Servers:** These are the highest level in the DNS hierarchy. The root server system consists of 1916 instances operated by 12 independent organizations,

illustrated in Figure 3. Root servers respond to queries by providing the IP addresses of Top-Level Domain (TLD) servers.

- **Top-Level Domain (TLD) Servers:** Each top-level domain—such as *.com*, *.org*, *.net*, *.edu*, or country-specific domains like *.uk*, *.fr*, *.ca*, and *.jp*—has a dedicated TLD server (or server cluster). TLD servers respond with the IP addresses of authoritative DNS servers responsible for specific domain names.
- **Authoritative DNS Servers:** These servers store and serve the actual DNS records for domain names. Any organization that operates publicly accessible services, such as web or email servers, must maintain authoritative DNS records mapping hostnames to IP addresses. Most large companies and universities operate their own primary and secondary (backup) authoritative DNS servers to ensure reliability and control.

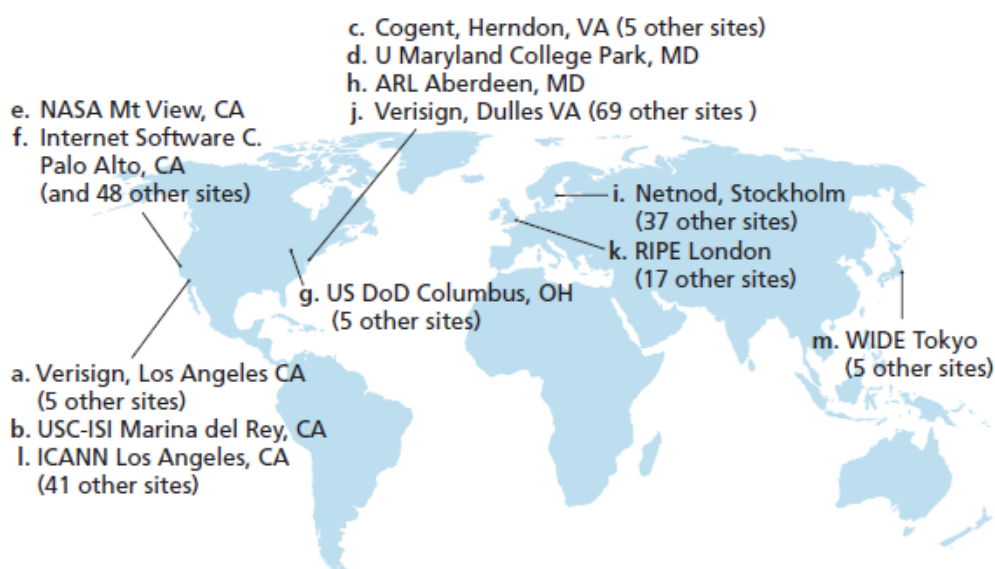


Figure 3: DNS root servers worldwide.

6.3.3 Overview of How DNS Works

DNS primarily uses *UDP Port 53*. However, as the internet evolves and security and functionality demands grow, *DNS* is increasingly relying on *TCP Port 53*. From its inception, *DNS* was designed to use both *UDP* and *TCP*: *UDP* is the default due to its lower overhead, while *TCP* is used as a fallback when a response is too large to fit in a single *UDP* packet or when reliability is required.

To understand how *DNS* resolution works in practice, consider a scenario in which a host at `engineering.nyu.edu` needs to resolve the domain name `gaia.cs.umass.edu` into an IP address. This process can be carried out using either *iterative* or *recursive* queries, as illustrated in Figure 4. In ***recursive queries***, the *DNS* server takes full responsibility for resolving the domain on behalf of the client, performing all necessary lookups and returning either a final answer or an error. In ***iterative queries***, the *DNS*

server returns the best information it has—typically a referral to another server—and leaves it to the client or resolver to continue the process.

In practice, a local DNS resolver typically performs recursive queries on behalf of end users, while itself using iterative queries to walk through the DNS hierarchy. This hybrid resolution process is shown in Figure 4(a) and proceeds as follows:

1. The user's computer (in this case, the host at `engineering.nyu.edu`) sends a DNS query to the local DNS server (`dns.nyu.edu`), often provided by the ISP.
2. If the local DNS server has the requested IP address cached, it replies directly to the user. If not, it initiates a query to a root DNS server.
3. The root server responds with a referral to a TLD DNS server responsible for `.edu` domains.
4. The local DNS server then queries the TLD server.
5. The TLD server responds with a referral to the authoritative DNS server for `cs.umass.edu`.
6. The local DNS server queries the authoritative server (`dns.umass.edu`).
7. The authoritative server returns the IP address for `gaia.cs.umass.edu`.
8. The local DNS server forwards the resolved IP address to the requesting host.

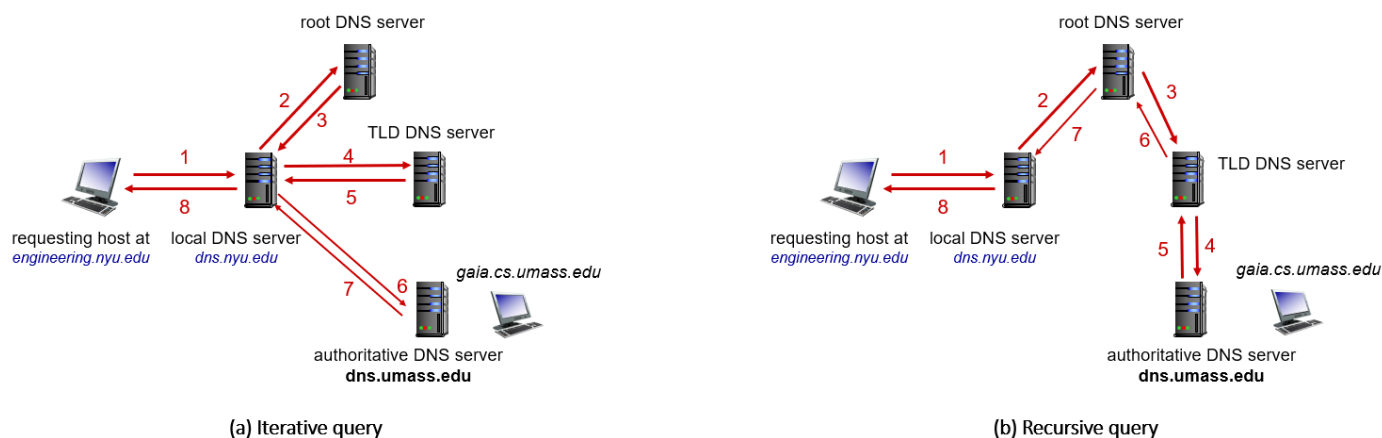


Figure 4: Types of DNS query resolution.

6.4 Web Server

A *web server* refers to both the software and hardware that handle client requests over the *World Wide Web* using the *HTTP*, which operates on *TCP port 80*, and its secure version, *HTTPS*, which operates on *TCP port 443*. The primary function of a web server is to store, process, and deliver *web pages* to users, enabling the display of website content in a browser. In addition to *HTTP*, web servers may also support other protocols such as *SMTP*, used for email communication over *TCP port 25*, and *File Transfer Protocol* (*FTP*), used for file transfers over *TCP port 21*. Web servers play a critical role in web hosting, providing the infrastructure needed to host websites and *web-based applications*.

6.4.1 *Web Client-Server Interaction*

- When a user enters a web address such as `www.example.com` into a browser and presses Enter, the browser begins the process of retrieving the web page.
- First, the domain name is resolved to an IP address using the DNS.
- Once the IP address is obtained, the browser initiates a **TCP connection** with the web server using the **TCP three-way handshake**:
 - The client sends a **SYN** (synchronize) packet to the server.
 - The server responds with a **SYN-ACK** (synchronize-acknowledge) packet.
 - The client sends an **ACK** (acknowledge) packet.
- After the reliable connection is established, the browser constructs an **HTTP request** (such as a **GET** request) and sends it to the server over the TCP connection.
- The client's ISP routes this request through the Internet to the server hosting `www.example.com`.
- The server receives the request, processes it, and generates an **HTTP response** (e.g., containing an HTML page).
- The response is sent back through the server's ISP, over the Internet, and eventually reaches the client's machine.
- The browser receives the HTTP response, parses the content (such as HTML, CSS, and JavaScript), and renders the web page for the user to view.

6.5 Email Server

An email server is a system responsible for the sending, receiving, storing, and management of email messages. It ensures the reliable delivery of messages between senders and recipients and operates continuously to allow email communication at any time. Email servers are widely used by individuals, businesses, and organizations to facilitate effective communication over the internet or within private networks. Some organizations choose to deploy their own email servers to maintain better security and control, while others opt for cloud-based services provided by companies such as Google or Microsoft.

To send emails, servers use the *SMTP*. When a user sends an email, the originating server processes the message and forwards it to the recipient's email server using SMTP. Upon arrival, the recipient's server stores the email until the user retrieves it. For email retrieval, servers typically employ either *POP3* or *IMAP*. POP3 downloads the email to the user's device and subsequently removes it from the server, making it suitable for devices with limited storage capacity. In contrast, IMAP retains messages on the server, allowing users to access and manage their emails from multiple devices while maintaining synchronization. Figure 5 illustrates the general flow of messages between email servers.

To ensure that emails are correctly routed to their destinations, email servers rely on the *DNS* and *Mail Exchange* (MX) records. These records determine the appropriate destination server based on the domain name in the recipient's email address (e.g., `@gmail.com`). In addition to routing, email servers incorporate various authentication

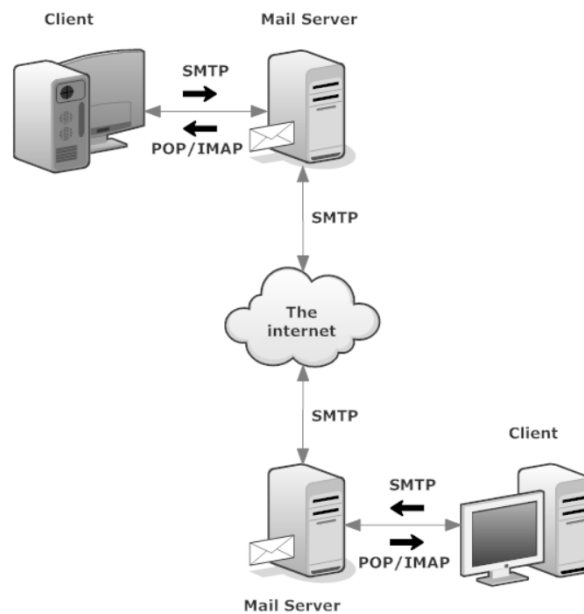


Figure 5: Mail server message flow.

and security mechanisms to protect against spam, phishing, and email spoofing. Modern email services implement encryption to ensure secure transmission of messages. The standard (non-secure) port numbers are as follows: IMAP uses port 143, POP3 uses port 110, and SMTP uses port 25. For secure communication, IMAP over SSL/TLS operates on port 993, POP3 over SSL/TLS uses port 995, and SMTP over SSL/TLS typically uses port 465 or port 587 when employing STARTTLS.

6.6 Procedure

In this lab, we will connect several PCs, routers, and multiple servers on different networks. This will require configuring key network services: DHCP, DNS, Email, and a Web server. These services will provide dynamic IP addressing, domain name resolution, and web service accessibility to devices on the network. The email server will handle sending and receiving emails within the network. Additionally, we will configure *Open Shortest Path First* (OSPF) to enable routing between routers and allow communication between all subnets.

6.6.1 Building the Topology

Construct the network topology shown in Figure 6, which consists of the following devices:

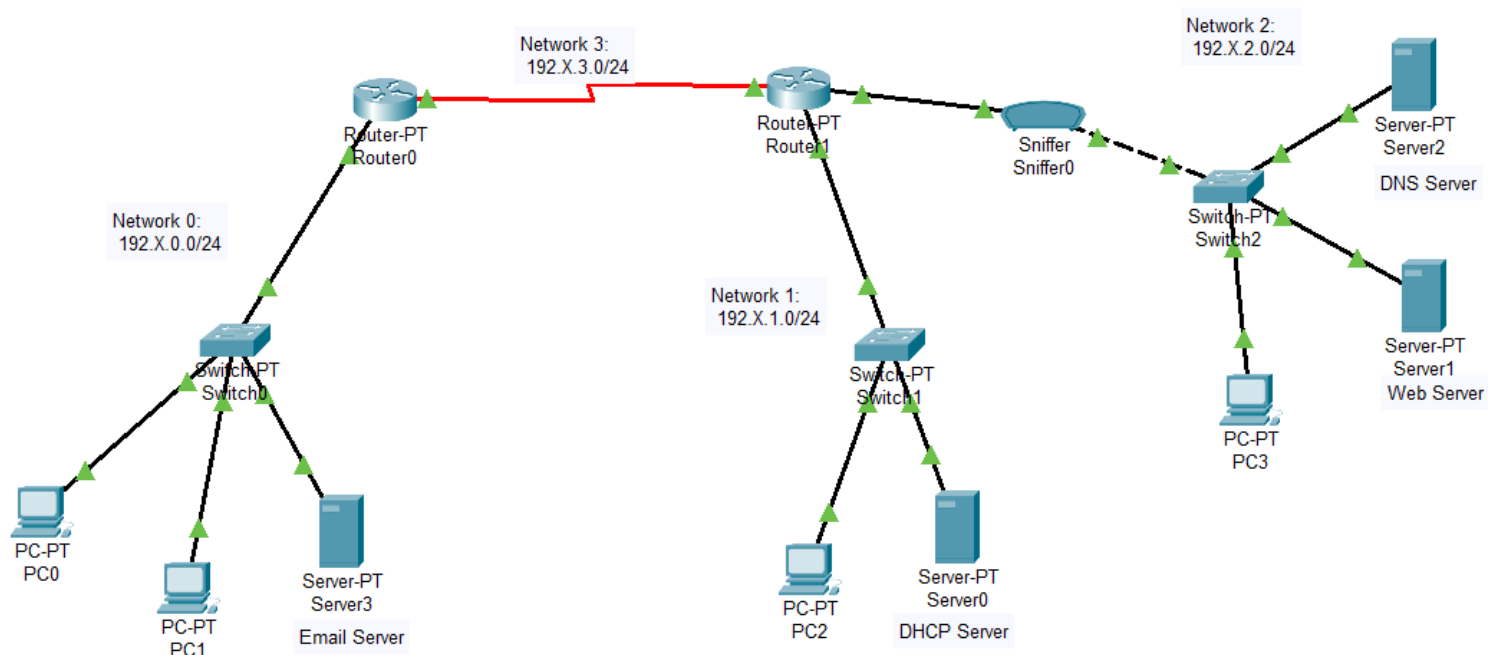


Figure 6: Server configuration topology.

- Routers: Use **Router-PT**.
 - Router0: The router will connect the two subnets (192.X.0.0/24 and 192.X.3.0/24) and participate in OSPF routing.
 - Router1: The router will act as the DHCP server for the two subnets (192.X.1.0/24 and 192.X.2.0/24) and participate in OSPF routing.
- Switches: Use **Switch-PT**.
- PCs: Use **PC-PT**.
 - PC0 and PC1 will receive dynamic IP addresses from the DHCP Server.
 - PC2 and PC3 will receive dynamic IP addresses from Router1 that act as the DHCP Server.

- Packet Capture: Place a **Sniffer** between Router 1 and Switch 2 to effectively monitor incoming packets.
- Servers:
 - Server0: The DHCP server that will assign IP addresses to the network 192.X.0.0/24.
 - Server1: The Web server that will host a sample webpage.
 - Server2: The DNS server that will resolve domain names to IP addresses.
 - Server3: The Email server.
- Connections: Use the “**Automatically Choose Connection Type**” option to link PCs, switches, and routers.

6.6.2 Network Setup and Configuration

A) Configuring Static IPs for Routers Interfaces

First, configure static IP addresses on the routers' interfaces. The value of **X** in the network address corresponds to the *last two digits* of your student ID. For example, if the network address is **192.X.0.0/24** and your student ID is **1230105**, then **X = 05**, resulting in the network address **192.5.0.0/24**.

- For Router0, configure static IP addresses on:

- Fa0/0 interface (connected to Switch0, 192.X.0.0/24 network):

```
Router(config)# interface fa0/0
Router(config-if)# ip address 192.X.0.1 255.255.255.0
Router0(config-if)# no shutdown
```

- Se2/0 interface (connected to Router1, 192.X.3.0/24 network):

```
Router0(config)# interface se2/0
Router0(config-if)# ip address 192.X.3.1 255.255.255.0
Router0(config-if)# no shutdown
```

- For Router1, configure static IP addresses on the relevant interfaces using the appropriate network addresses, following the same steps as Router0.

B) IP Configuration for the Servers

Figures 7 to 10 illustrate the IP configurations for the DNS, DHCP, web, and email servers. These configurations include the static IP addresses, subnet masks, default gateways, and DNS settings assigned to each server.

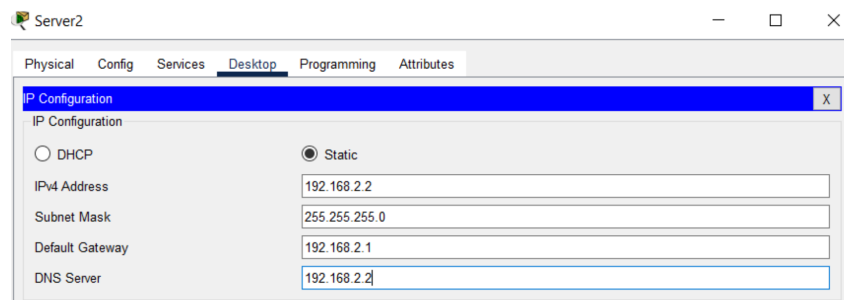


Figure 7: IP configuration for DNS server.

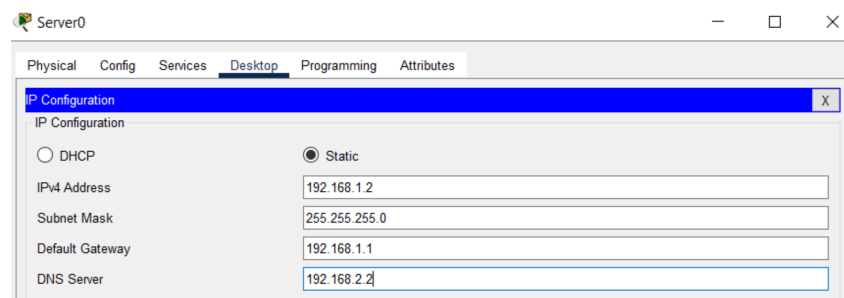


Figure 8: IP configuration for DHCP server.

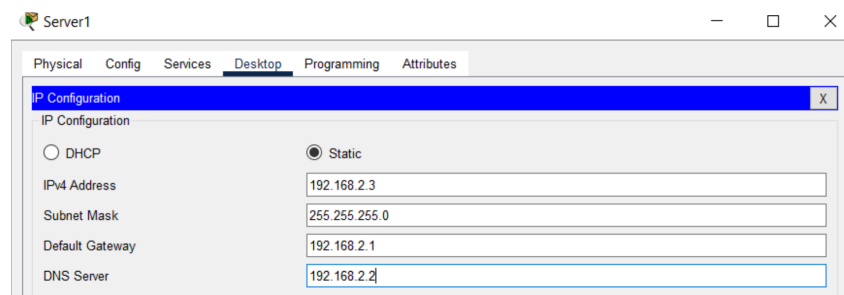


Figure 9: IP configuration for web server.

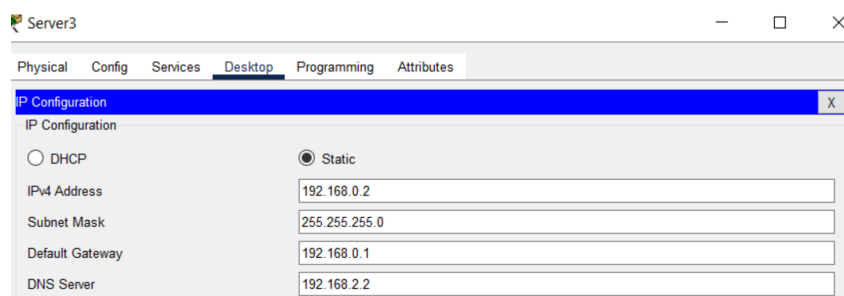


Figure 10: IP configuration for email server.

C) Enabling OSPF Routing

Enable OSPF routing on both Router0 and Router1, as mentioned in previous experiments. For example, to configure OSPF on **Router0**, use the following commands:

```
Router(config)# router ospf 1
Router(config-router)# network 192.X.0.0 0.0.0.255 area 0
Router(config-router)# network 192.X.3.0 0.0.0.255 area 0
```

D) Configuring DHCP on a Router

Now, we will configure **Router1** to serve as the DHCP server for both subnets (192.X.1.0/24 and 192.X.2.0/24).

- **Configure DHCP for Network 192.X.1.0/24**

1. **Exclude Reserved IP Addresses in Network 192.X.1.0/24:** Exclude the range 192.X.1.1 to 192.X.1.10 to reserve these addresses.

```
Router(config)# ip dhcp excluded-address 192.X.1.1
                  192.X.1.10
```

2. **Create DHCP Pool for Network 192.168.1.0/24:** Set up the DHCP pool with the default gateway and DNS server for devices on this subnet.

```
Router(config)# ip dhcp pool LAN1
Router(dhcp-config)# network 192.X.1.0 255.255.255.0
Router(dhcp-config)# default-router 192.X.1.1
Router(dhcp-config)# dns-server 192.X.2.2
Router(dhcp-config)# exit
```

- **Configure DHCP for Network 192.X.2.0/24**

1. **Exclude Reserved IP Addresses in Network 192.X.2.0/24:** Exclude the range 192.X.2.1 to 192.X.2.10 to reserve these addresses.

```
Router(config)# ip dhcp excluded-address 192.X.2.1
                  192.X.2.10
```

2. **Create DHCP Pool for Network 192.X.2.0/24:** Set up the DHCP pool with the default gateway and DNS server for devices on this subnet.

```
Router(config)# ip dhcp pool LAN0
Router(dhcp-config)# network 192.X.2.0 255.255.255.0
Router(dhcp-config)# default-router 192.X.2.1
Router(dhcp-config)# dns-server 192.X.2.2
Router(dhcp-config)# exit
```

3. **Activate DHCP on the router:**

```
Router(config)# service dhcp
```

E) Configuring DHCP Server

We will now configure **Server0** to dynamically assign IP addresses to the 192.X.0.0/24 network, as illustrated in Figures 11 and 12. The configuration will be performed using the following steps:

1. Open Server0 and navigate to the Services tab.
2. Select DHCP from the left-hand menu.
3. Select the appropriate interface (FastEthernet0 in this case).
4. Ensure the Service is turned On.
5. In the **Pool Name** field, enter a name (e.g., Pool1).
6. Set the **Default Gateway** to 192.X.0.1, which will act as the gateway for the assigned addresses.
7. Enter the **DNS Server** address as 192.X.2.2, which will handle domain name resolution.
8. Define the **Start IP Address** (192.X.0.2) for the first available address in the range.
9. Set the **Subnet Mask** to 255.255.255.0 to match the network configuration.
10. Click Add to add the DHCP pool to the server's settings.
11. Click Save to finalize the configuration.

Once configured, DHCP clients (PCs) in network 192.X.0.0/24 will receive an IP address dynamically from the defined pool, along with the appropriate gateway and DNS settings.

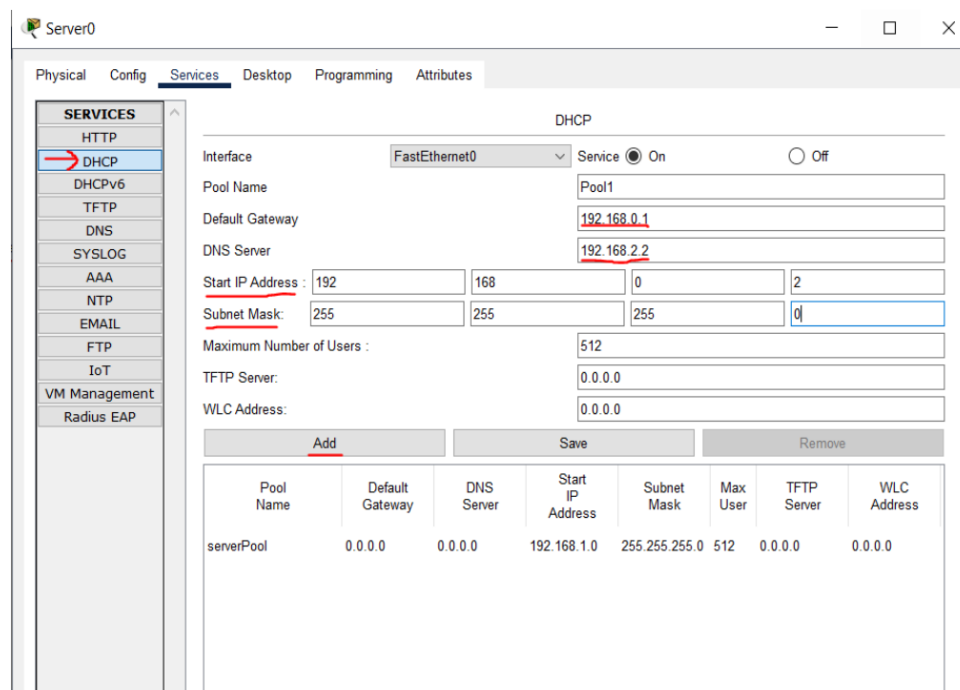


Figure 11: Assigning IP addresses dynamically using the DHCP server.

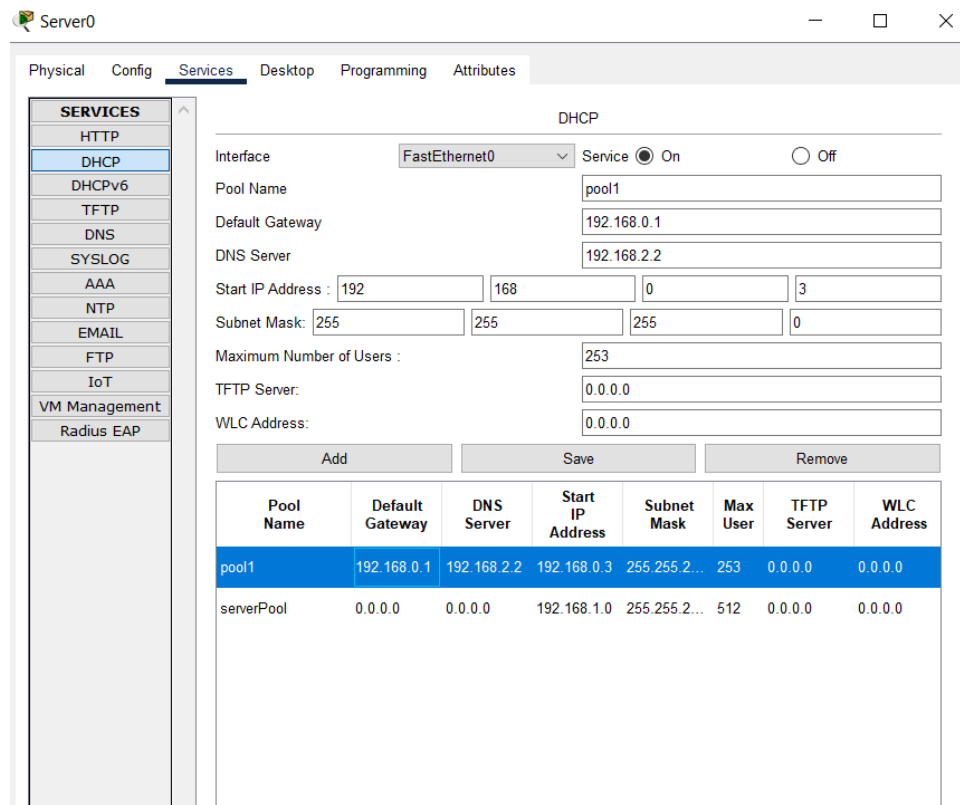


Figure 12: Adding the specified pool to the table.

F) Configuring DHCP Relay on Router0

Since the DHCP server (Server0) is not directly connected to network 192.X.0.0/24, we must configure Router0 to forward DHCP broadcast messages received on the gateway interface for this subnet to Server0 using the ip helper-address command.

Navigate to the FastEthernet0/0 interface (connected to the 192.X.0.0/24 network):

```
Router(config)# interface fa0/0
Router(config-if)# ip helper-address 192.X.1.2
Router(config-if)# exit
```

G) Assigning Dynamic IP Addresses

To assign dynamic IP addresses to all PCs, follow these steps:

- On each PC, go to Desktop.
- Set the IP Configuration to DHCP.

On PC0 and PC1, verify that they receive the correct IP configuration (IP address, subnet mask, default gateway, and DNS server) from the DHCP server. Figure 13 shows the DHCP IP configuration for PC0. On PC2 and PC3, verify that they receive the correct IP configuration from the router's DHCP service. Figure 14 shows the DHCP IP configuration for PC3.

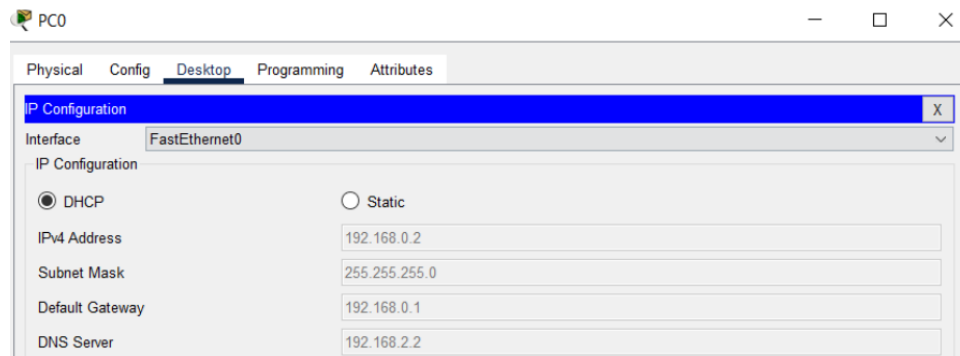


Figure 13: IP configuration for PC0.

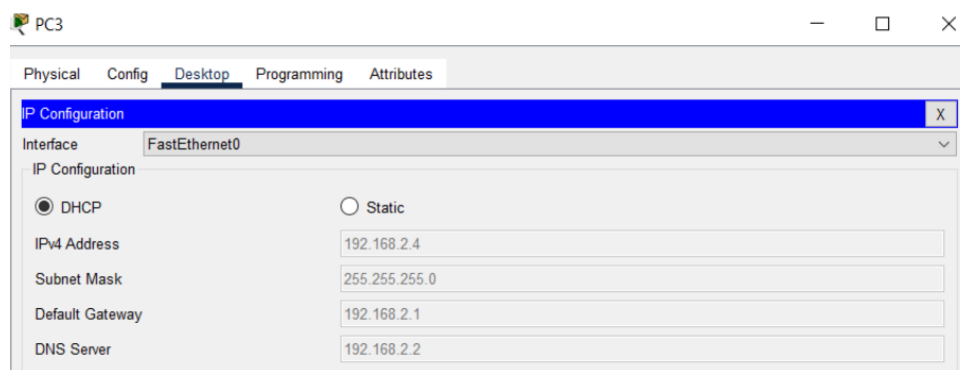


Figure 14: IP configuration for PC3.

H) Configuring Web Server

Now, we will configure the Web Server (Server1) to host a sample webpage.

- **Enable HTTP Service on Server1:**

1. Go to Server1 and select the Services tab.
2. Under Services, select HTTP.
3. Turn the HTTP service ON.

- **Upload or Create a Webpage:**

1. In the HTTP settings, upload a sample HTML file or use the default webpage.
2. If desired, add content for a custom webpage in the *index.html* field.

Figure 15 shows how to configure Web Server, you can edit the *index.html* page to modify the webpage that you want to show. Now, we want to access www.birzeit.edu, so we have to modify the *index.html* file in order to make it open the page. First, go to Birzeit University official page in the browser, right click and “Save As”. Then copy the HTML file and paste it into *index.html* after clicking on “Edit” and save.

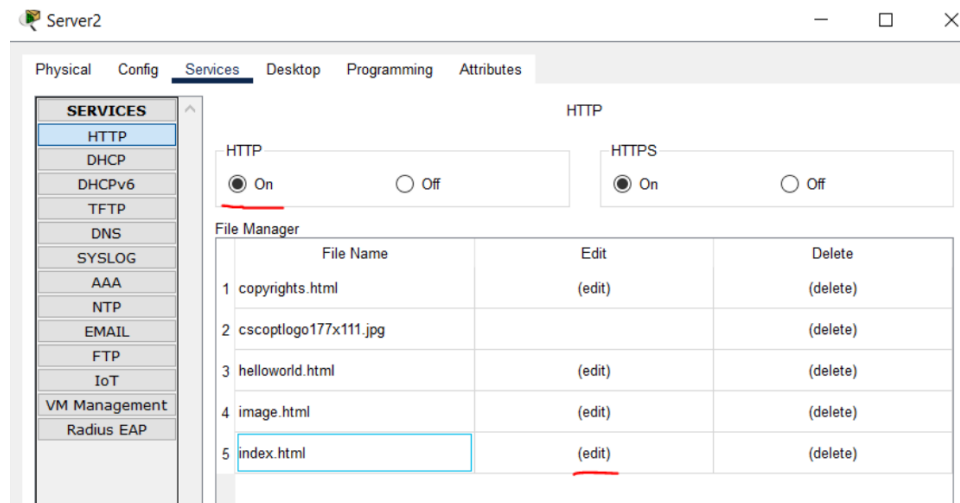


Figure 15: Enabling HTTP service.

I) Configuring DNS Server

Next, we will configure the DNS server (Server2) to resolve domain names to IP addresses.

- **Enable DNS on Server2:**

1. Go to Server2 and select the Services tab.
2. Under Services, select DNS.
3. Turn the DNS service ON.

- **Add DNS Entries:**

1. In the Name field, enter the domain name (e.g., www.birzeit.edu) or the PC name.
2. In the Address field, enter the IP address of all devices.
3. Add the entry to the DNS table.
4. Repeat this step for any additional domain names needed.

After saving and adding the DNS record, it has to be added with the record number, name, type and details, including its IP address, as shown in Figure 16.

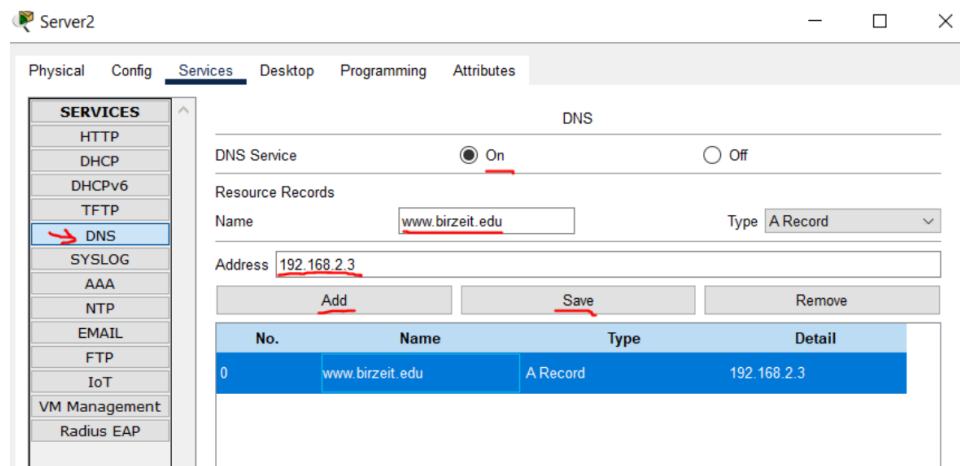


Figure 16: Adding the entry to the DNS table.

Now, we want to add a DNS to all PCs and servers in the topology, as shown in Figure 17, so that we can send packets to them by their names rather than their IP addresses.

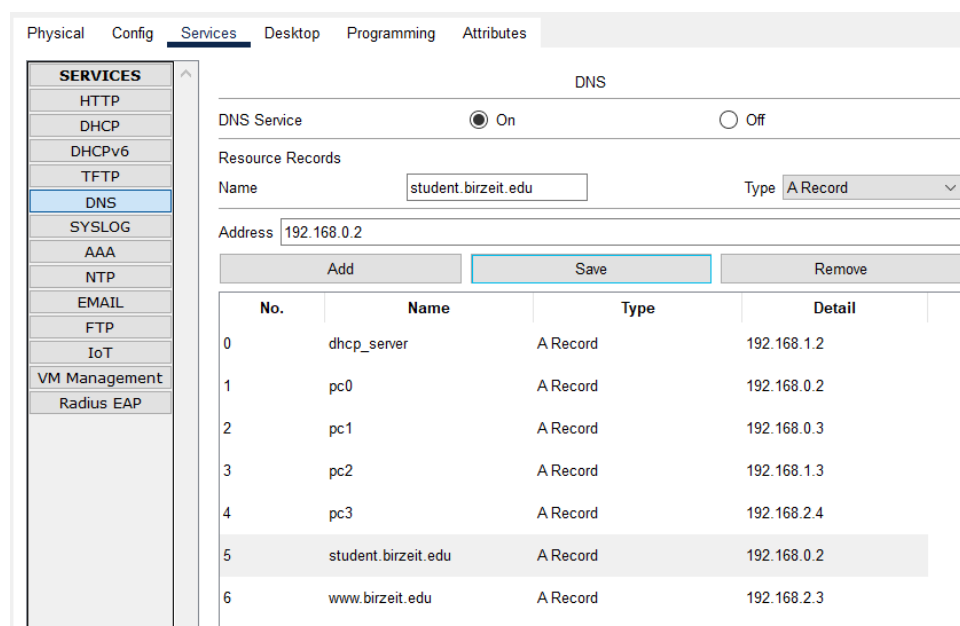


Figure 17: Adding records to all PCs to the DNS table.

J) Configuring Email Server

In this section, we will configure the Email Server (Server3) to enable sending and receiving emails within the network, as shown in Figures 18 and 19.

• Enable Email Service on Server3:

1. Go to Server3 and select the Services tab.
2. Under Services, select Email.
3. Turn the SMTP and POP3 services ON.

- **Add User Accounts:**

1. Navigate to the Users section.
2. Set the Domain Name (mail server alias hostname) as ***student.birzeit.edu***.
3. Create a user account by specifying the User name and Password.
4. Save the user entries.

The email address will follow this format: ***User@DomainName***.

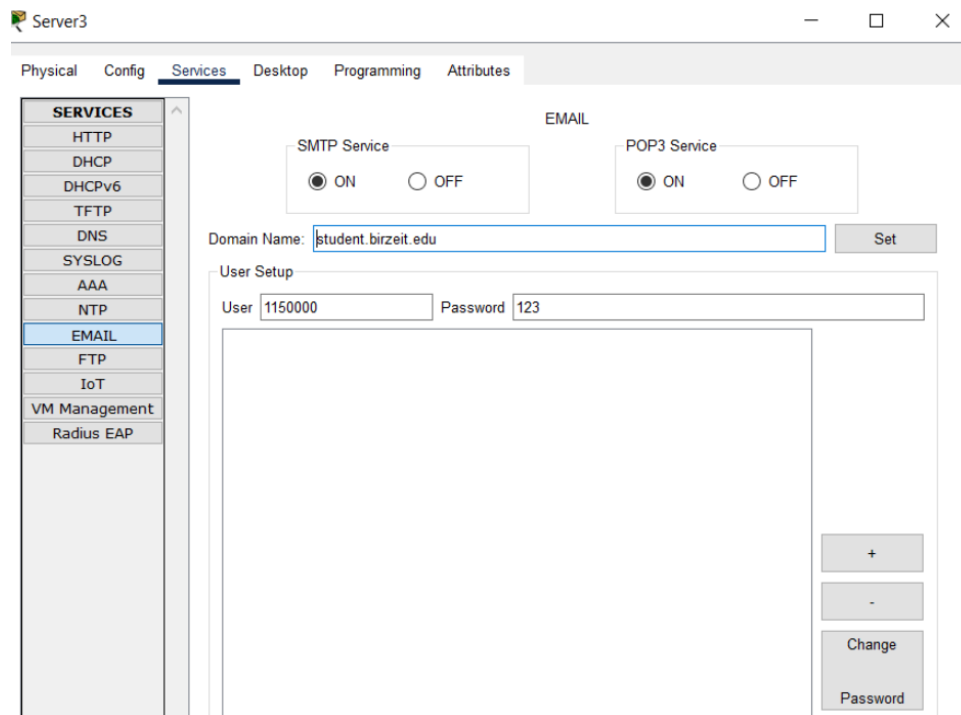


Figure 18: Setting domain name to the Email server.

Now, we need to create user accounts to all PCs, each with a specific account.

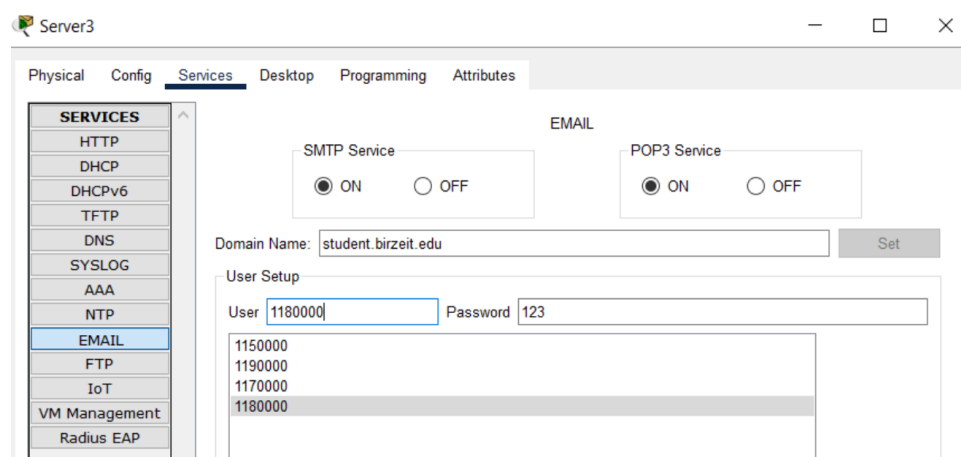


Figure 19: Creating user accounts.

- **Configure Email Clients on PCs:**

1. On each PC, open Email app. (Desktop → Email) and select Configure Mail.
2. Configure the PC's email client using the following settings:
 - (a) Your Name: FirstName
 - (b) Email Address: User@DomainName
 - (c) Incoming Mail Server: student.birzeit.edu
 - (d) Outgoing Mail Server: student.birzeit.edu
 - (e) User Name: User
 - (f) Password: Password

Figures 20 and 21 show how to configure the email client in PC0 and PC1 for the first and second accounts, respectively. **Configure mail to all other PCs in the topology.**

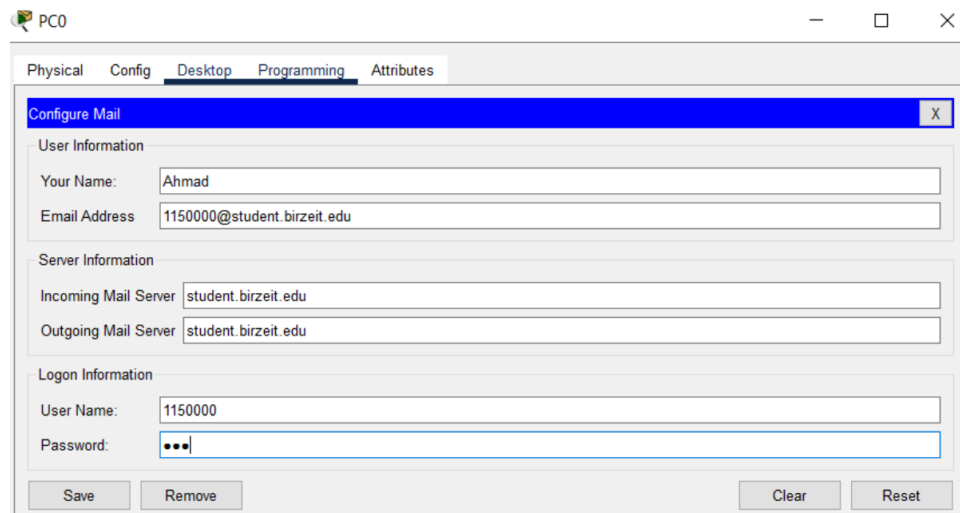


Figure 20: Configuring Email Clients on PC0.

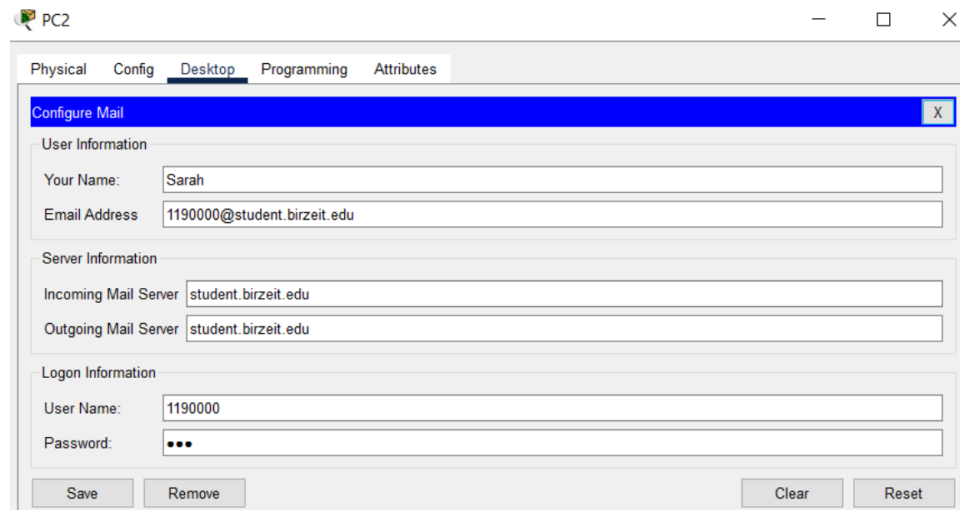


Figure 21: Configuring Email Clients on PC2.

- **Sending an Email between Users:**

Now, we want to send a mail from Ahmad, User: '1150000' to Sarah, User: '1190000'

1. Open the email client on PC-Ahmad.
2. Compose a new email, set the recipient as Sarah (User: 1190000), and enter a subject and message.
3. Click Send.

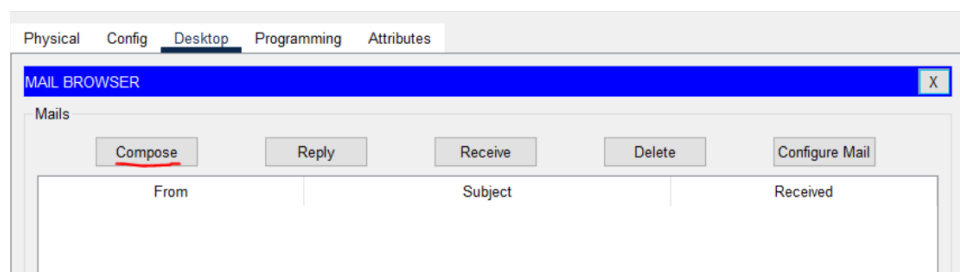


Figure 22: Compose a new email from Ahmad to Sarah.

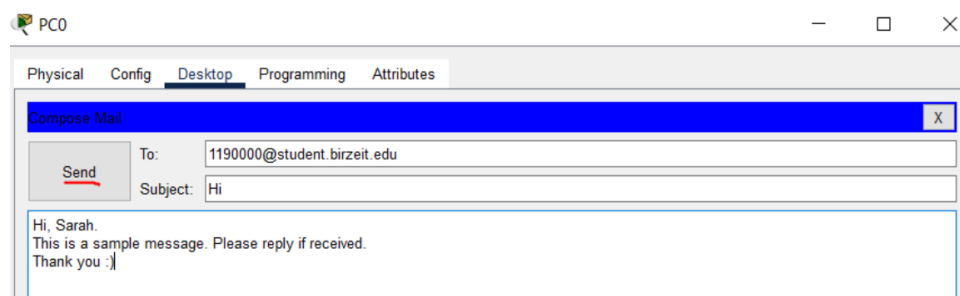


Figure 23: Sending an Email.

4. Verify that Sarah receives the email on her email client by clicking on ‘Receive’ to show all received Emails.

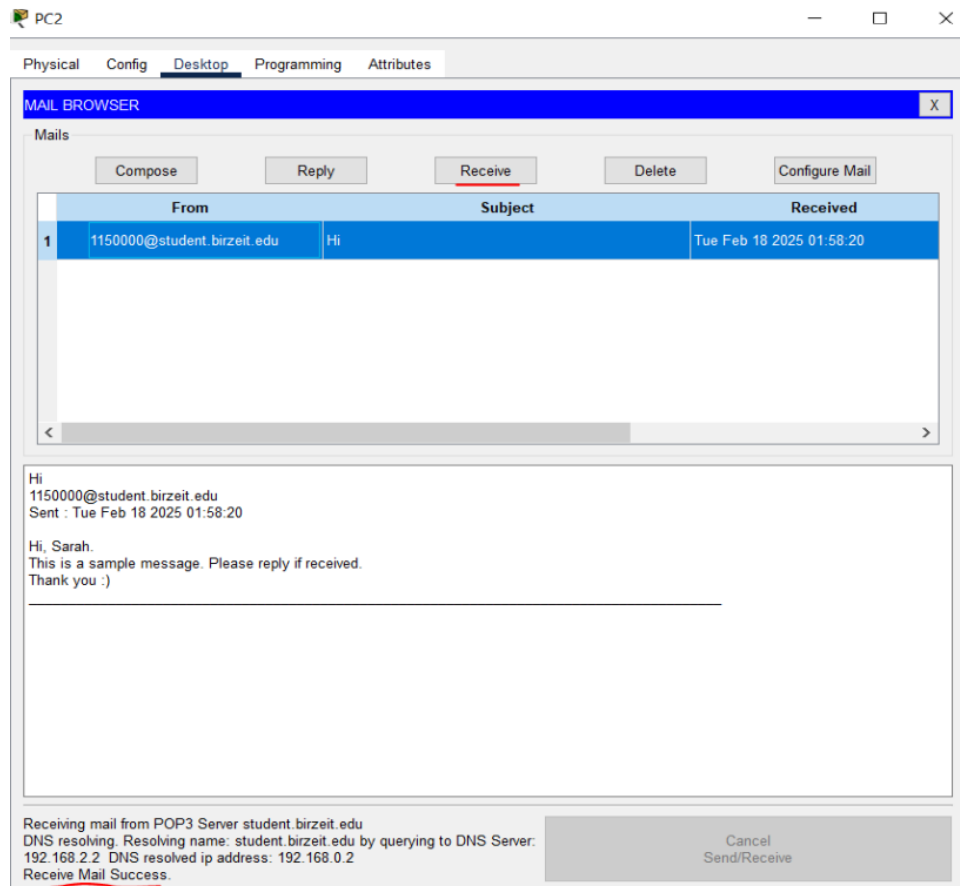


Figure 24: Show received Emails.

5. If Sarah want to reply to Ahmad's Email, this can be done by selecting the email that is received and clicking on 'Reply'.

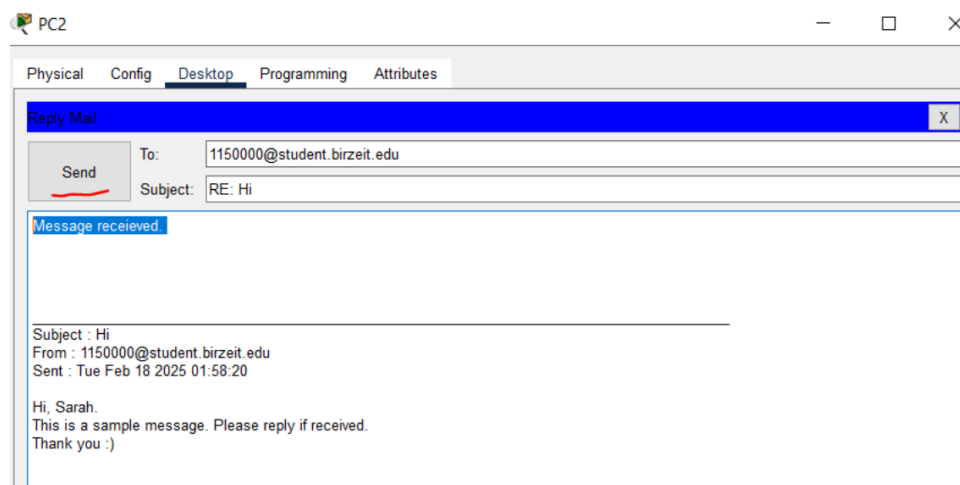


Figure 25: Replying to Email.

6. Ahmad can check the reply to his message by clicking on 'Receive' to show the received mails.

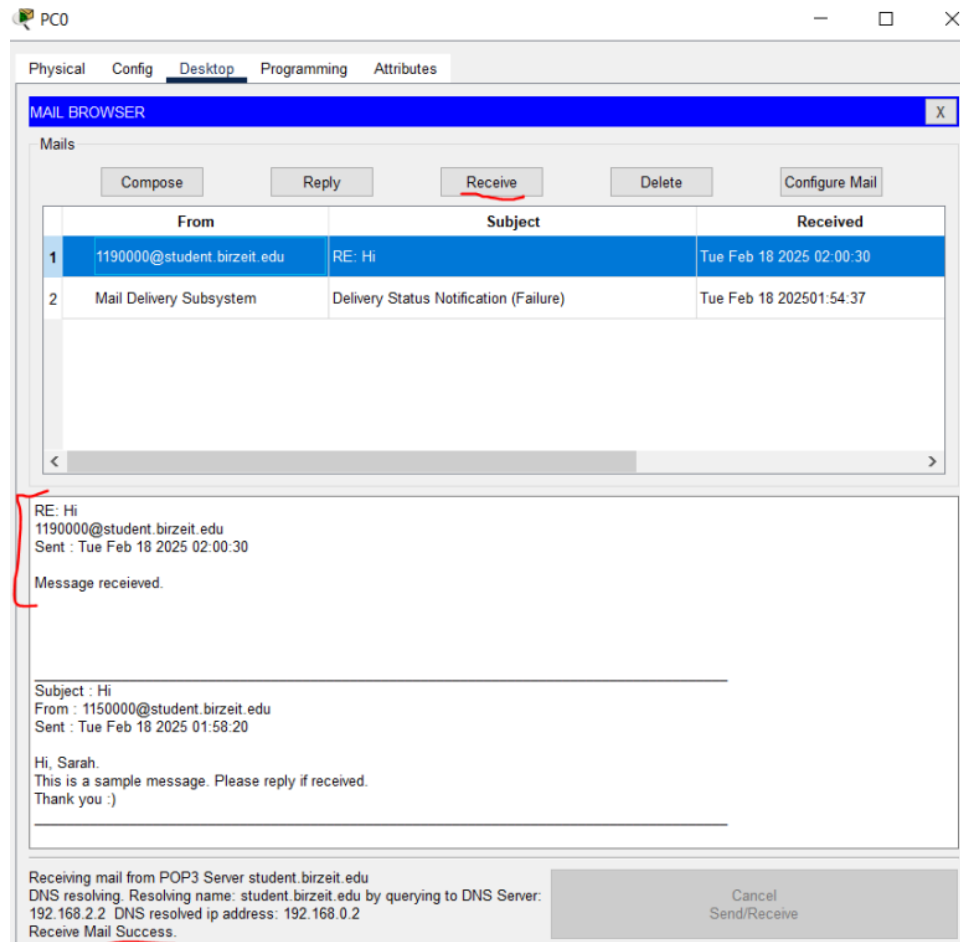


Figure 26: Show received Emails (2).

6.6.3 Testing and Verification

DNS Testing

- From any PC, open a Command Prompt and use the ping command to ping the domain name (e.g., ping www.birzeit.edu).
- Verify that the domain name resolves to the correct IP addresses.

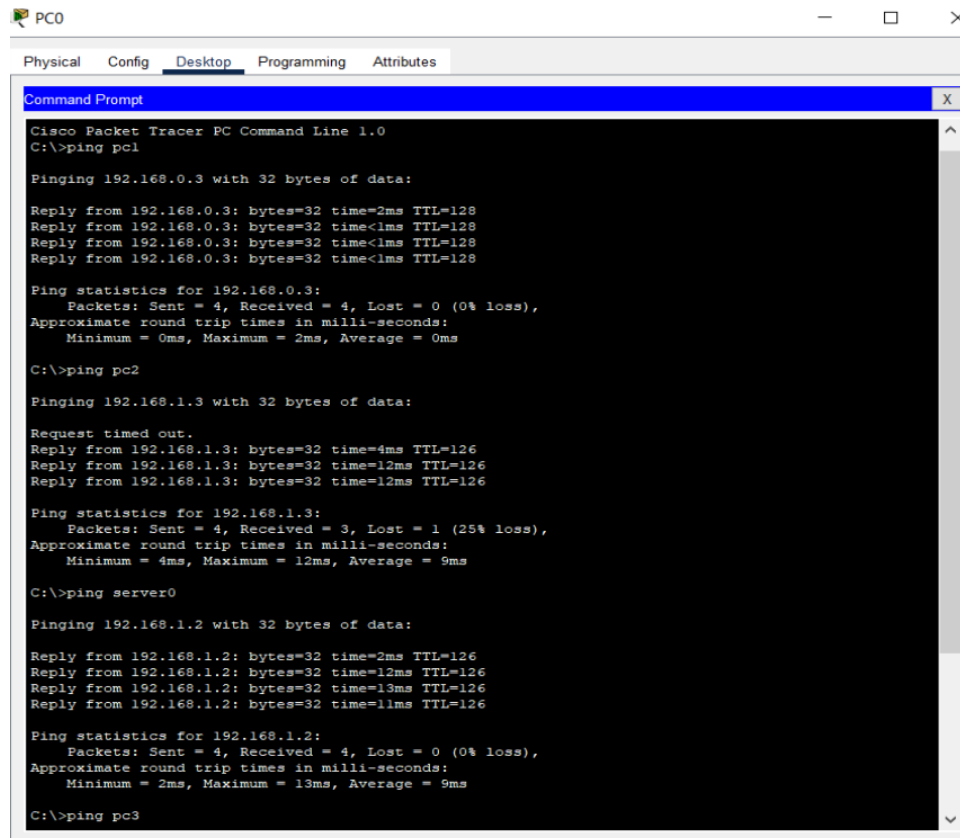


Figure 27: Pinging PCs by their Domain Names.

Web Server Testing

- On any PC, open a web browser.
- Type the IP address of Server1 (e.g., `http://192.X.2.3`) or the domain name (e.g., `http://www.birzeit.edu`).
- Verify that the webpage hosted on Server1 loads correctly.



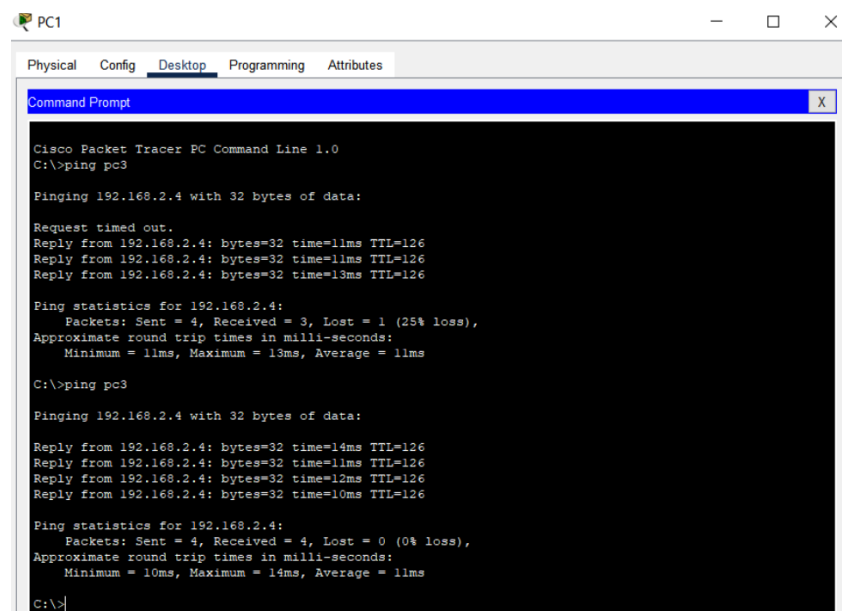
Figure 28: Web Server Testing by domain name.

6.6.4 Packet Sniffing

We used the **Sniffer** device to visualize packet movement and understand how data flows through the network. Go to **Sniffer** → **GUI** → **Service**, and toggle **Check On** to enable packet capturing. Then, select **Port 0** to determine the interface will be used to capture incoming packets. By default, the sniffer captures **all protocols**, but you can apply filters by;

- Clearing all incoming packets by clicking on ‘Clear’.
- Clicking on ‘Edit Filters’ to focus on specific types of network traffic for more precise analysis.

Perform a ping from **PC1** to **PC3** using the domain name (ping pc3). The domain name **pc3** is resolved by the **DNS server**, meaning the first packets sent will be **DNS requests and responses** to obtain PC3’s IP address. Once the IP address is resolved, **ICMP** packets will be sent for the actual ping operation to test connectivity.



```
PC1
Physical Config Desktop Programming Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping pc3

Pinging 192.168.2.4 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.4: bytes=32 time=11ms TTL=126
Reply from 192.168.2.4: bytes=32 time=11ms TTL=126
Reply from 192.168.2.4: bytes=32 time=13ms TTL=126

Ping statistics for 192.168.2.4:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 11ms, Maximum = 13ms, Average = 11ms

C:\>ping pc3

Pinging 192.168.2.4 with 32 bytes of data:

Reply from 192.168.2.4: bytes=32 time=14ms TTL=126
Reply from 192.168.2.4: bytes=32 time=11ms TTL=126
Reply from 192.168.2.4: bytes=32 time=12ms TTL=126
Reply from 192.168.2.4: bytes=32 time=10ms TTL=126

Ping statistics for 192.168.2.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 10ms, Maximum = 14ms, Average = 11ms

C:\>
```

Figure 29: Pinging PC3 by its Domain Name.

To analyze this process, we need to capture **DNS** and **ICMP** packets:

- **DNS** packets will show the name resolution process as PC1 queries the DNS server for PC3’s IP address.
- **ICMP** packets will verify the connectivity between PC1 and PC3 once the domain name is resolved.

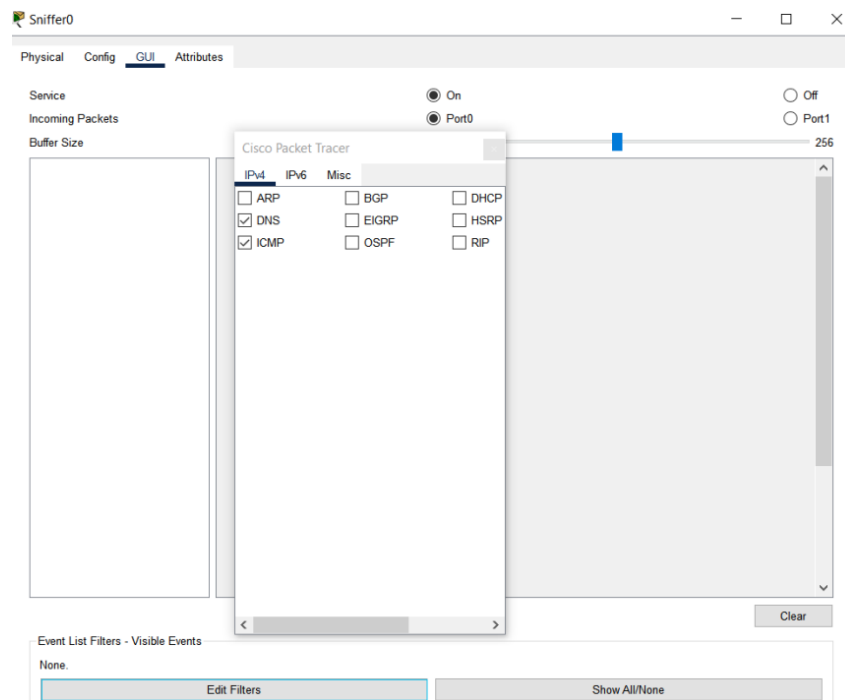


Figure 30: Applying Filter to Capture a Specific Protocol.

Now, after specifying only ICMP and DNS to show and sending a packet from PC1 to PC3, we can click on the packet in the sniffer tool to examine its details.

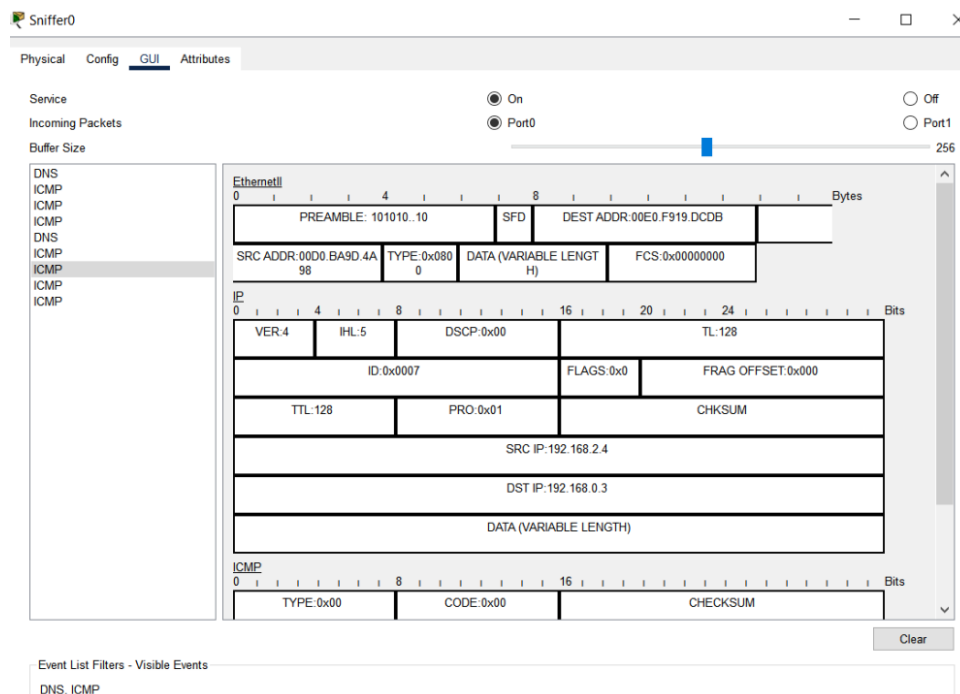


Figure 31: Packet Details for ICMP Protocol.

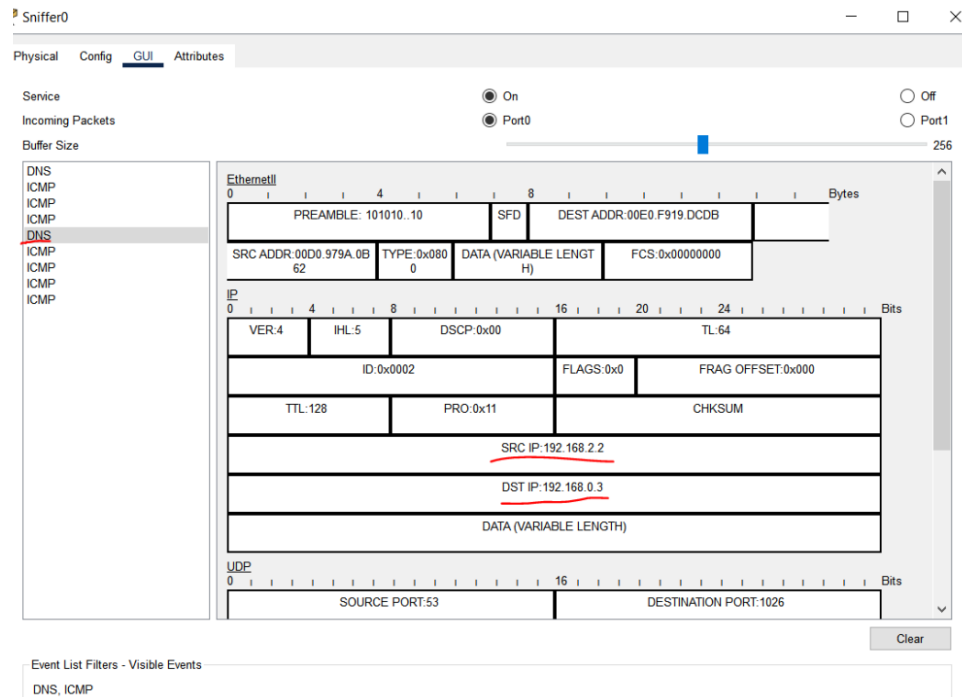


Figure 32: Packet Details for DNS Protocol.

6.7 ToDo

This section will be provided by the instructor.