Smit Sutariya **21BCP142**

Practical 8

Part A

Class B Tech CSE 3rd Year **Sub : Computer Networks**

Aim:
Write a program for interactive application using UDP socket.

Prerequisite: Nil

Outcome: To impart knowledge of Socket Programming

Theory:
Socket programming is a fundamental technique used to establish communication between two processes, whether they are on the same machine or different machines across a network. It relies on utilizing the APIs (Application Programming Interfaces) provided by the operating system to create, configure, and utilize network sockets. These sockets act as the endpoints for transmitting and receiving data.

In the context of network communication, two essential transport layer protocols are commonly used: TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). In this theory, we focus on UDP.

UDP is a core component of the Internet Protocol Suite and serves the purpose of enabling communication between applications that require efficient and low-latency data exchange. Unlike TCP, which is connection-oriented and ensures reliable data delivery, UDP is connectionless. This means that it does not establish and maintain a persistent connection between the communication endpoints. Instead, it operates by sending discrete messages, known as datagrams. These datagrams are designed to be transmitted without the need for implicit handshaking dialogues, making UDP suitable for applications where speed and low overhead are prioritized over guaranteed delivery and order of data.

Procedure:

1. Write Simple Client Server Program using Java/Python Programming Language
2. Execute the program using appropriate compiler.
3. Verify the working of the program.

Part B

Steps:
UDP_Client.py

```python
import socket

if __name__ == "__main__":
    host = "127.0.0.1"
    port = 4455
    addr = (host, port)

    client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
    while True:
        data = input("Enter a word: ")

        if data == "EXIT":
            data = data.encode()
            client.sendto(data, addr)

            print("Disconneted from the server.")
            break

        data = data.encode()
        client.sendto(data, addr)

        data, addr = client.recvfrom(1024)
        data = data.decode()
        print(f"Server: {data}")
```

UDP_Server.py

```
import socket

if __name__ == "__main__":
    host = "127.0.0.1"
    port = 4455

    server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    server.bind((host, port))

    print(f"UDP Server listening on {host}: {port}")
```

```python
    while True:
        data, addr = server.recvfrom(1024)
        data = data.decode()
        # print(data)

        if data == "EXIT":
            print("Client disconnected.")
            break

        print(f"Client: {data}")

        data = data.upper()
        data = data.encode()
        server.sendto(data, addr)
```

Output:

```
PS D:\Sem-5\Network Lab\Lab_89&10> python -u "d:\Sem-5\Network Lab\Lab
89&10\Udp_client.py"
Enter a word: hello
Server: HELLO
Enter a word: i am smit sutariya
Server: I AM SMIT SUTARIYA
Enter a word: this is Network Lab
Server: THIS IS NETWORK LAB
Enter a word: done
Server: DONE
Enter a word: EXIT
Disconneted from the server.
PS D:\Sem-5\Network Lab\Lab_89&10>
```

```
PS D:\Sem-5\Network Lab\Lab_89&10> python -u UDP_Server.py
UDP Server listening on 127.0.0.1: 5500
Client: hello
Client: i am smit sutariya
Client: this is Network Lab
Client: done
Client disconnected.
PS D:\Sem-5\Network Lab\Lab_89&10>
```

Observation & Learning:

We have successfully created client and server components for network applications, which are fundamental in network programming.

We learned how to establish socket connections between the client and server, enabling communication.

The data exchange between the client and server allows us to understand the flow of information in a network application.

Conclusion:

This experiment provides a crucial foundation in socket programming and network communication. It equips us with the practical skills needed to design and implement client-server components for various network applications. Such skills are invaluable in the realms of computer networking and software development. As we progress through this experiment, we are gaining a deeper understanding of complex network applications and the underlying principles of network protocols.

Smit Sutariya **21BCP142**

Practical 9

Part A	
Class B Tech CSE 3rd Year	**Sub : Computer Networks**
Aim: Configure DHCP and SMTP in a small LAN	
Prerequisite: Nil	
Outcome: To impart knowledge of Application Layer Protocol	

Theory:

DHCP (Dynamic Host Configuration Protocol):
DHCP is a network protocol used to automatically assign IP addresses and other network configuration settings to devices within a network. It simplifies IP address management by dynamically allocating and renewing IP addresses to devices as they connect to the network. Here are the key aspects of DHCP:

- **Dynamic IP Assignment:** DHCP allows devices to receive IP addresses dynamically, eliminating the need for manual IP configuration.
- **IP Address Pool:** DHCP servers maintain a pool of available IP addresses. When a device connects to the network, the server assigns an IP address from this pool.
- **Lease Duration:** Devices are assigned IP addresses for a specific lease duration. After the lease expires, devices must renew their lease or request a new IP address.
- **Centralized Management:** DHCP offers centralized control over IP address allocation and configuration settings, making it easier to manage and scale networks.
- **Reduced IP Conflicts:** DHCP helps reduce IP address conflicts, ensuring that devices receive unique IP addresses.

SMTP (Simple Mail Transfer Protocol):
SMTP is a protocol used for sending and relaying email messages between email clients and email servers. It plays a crucial role in the transmission of electronic mail across the internet and within a local area network. Key features of SMTP include:

- **Email Transmission:** SMTP is responsible for transmitting outgoing email messages from the sender's email client to the recipient's email server.
- **Relaying:** SMTP servers can relay email messages to other SMTP servers, facilitating email delivery across networks.
- **Protocols:** SMTP can use both unencrypted (port 25) and encrypted (e.g., port 587 with TLS/SSL) communication channels.
- **Authentication:** SMTP servers may require authentication to prevent unauthorized email sending.
- **Message Routing**: SMTP servers use DNS (Domain Name System) to route messages to the recipient's email server based on the recipient's email address.
- **Message Format:** SMTP defines the format of email messages and provides rules for addressing, headers, and message content.

Procedure:

1. Simulate the DHCP and SMTP using Cisco Packet Tracer.
2. Analyse the traffic using Wireshark.

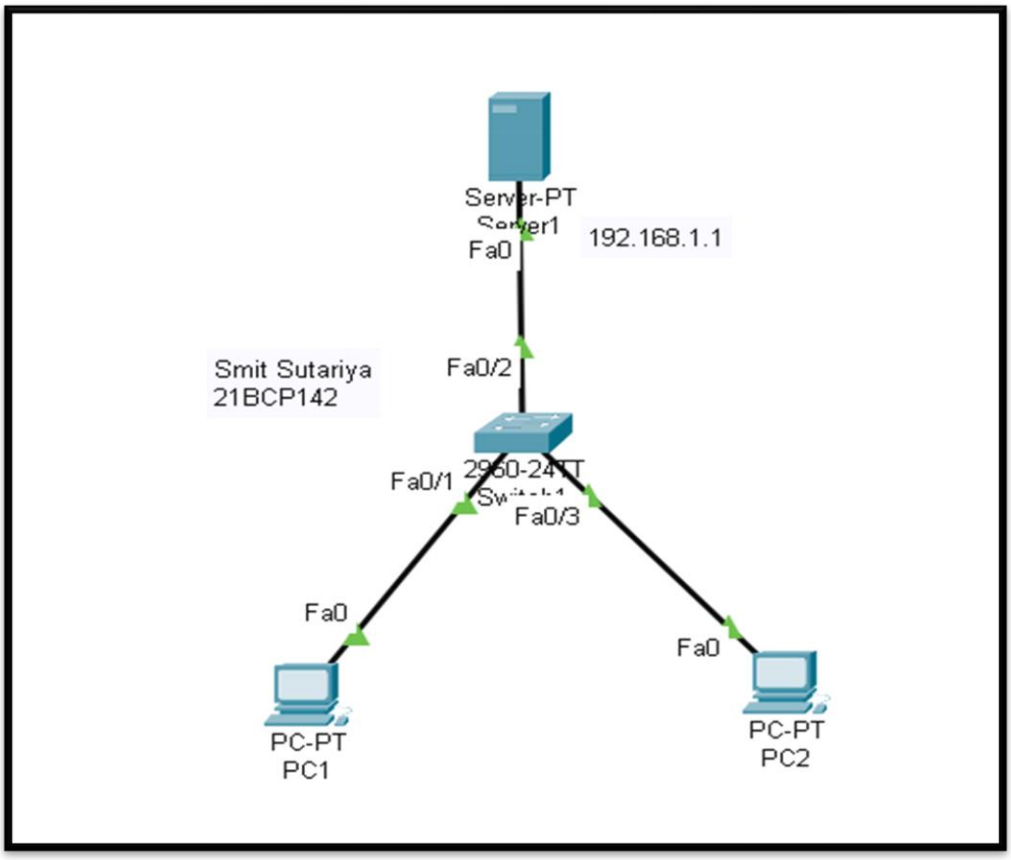
PART - B

Steps:

DHCP Configuration:

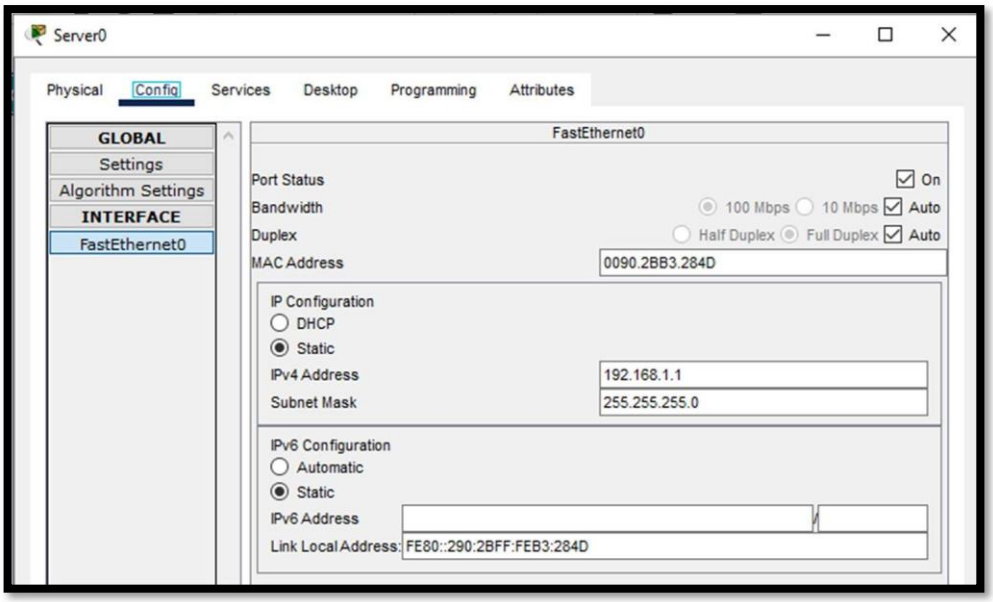
Step 1: Create Your Network

- Open Cisco Packet Tracer and launch it.
- Drag and drop a Server onto the workspace.
- Add a switch to the workspace.
- Place end devices (e.g., PCs) on the workspace.
- Connect each PC to a switch port using the 'Connections' tool.
- Connect the switch to the server Ethernet port.



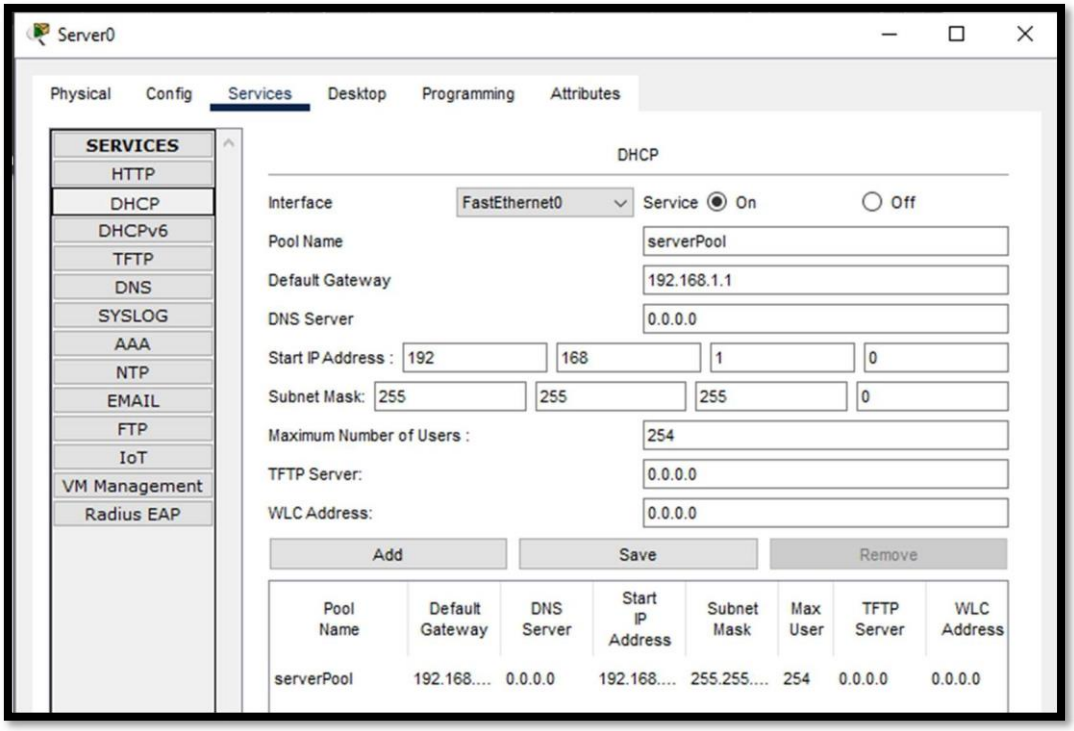
Step 2: Configure the Server

- Set the IP address (e.g., 192.168.1.1) and subnet mask (e.g., 255.255.255.0).



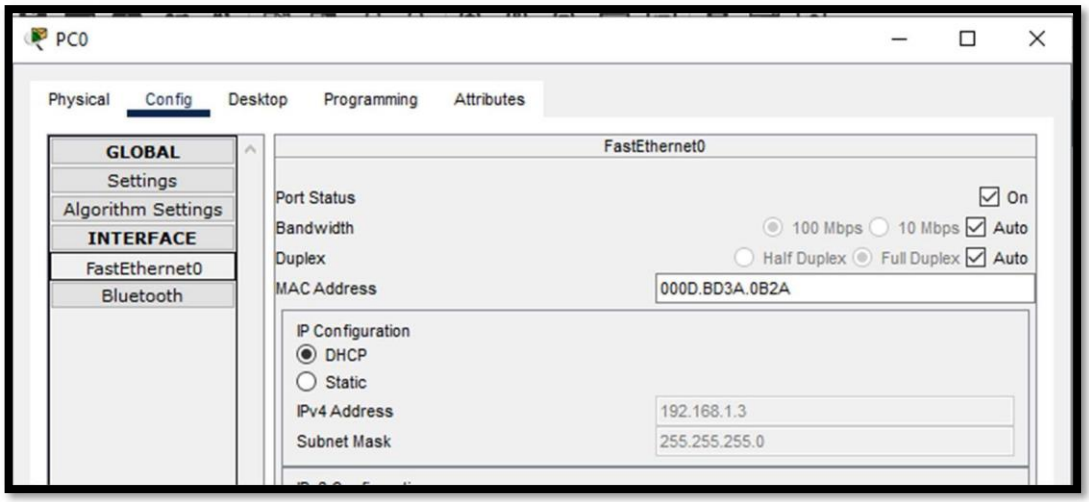
Step 3: Configure DHCP Server

- In the GUI, go to the 'Services' tab.
- Select 'DHCP' from the service list.
- Create a new DHCP pool by clicking 'Add' or editing an existing one.
- Configure the DHCP pool with the following settings:
- Pool Name: Enter a name for your DHCP pool.
- Default Gateway: Set to the router's IP address (e.g., 192.168.1.1).
- Start IP Address: Specify the beginning of the IP range (e.g., 192.168.1.100).
- Subnet Mask: Enter the appropriate subnet mask (e.g., 255.255.255.0).
- Maximum Number of Users: Define the number of devices that can get an IP from this pool.
- Optionally, set TFTP Server, DNS Server, and Lease Time.
- Save the configuration



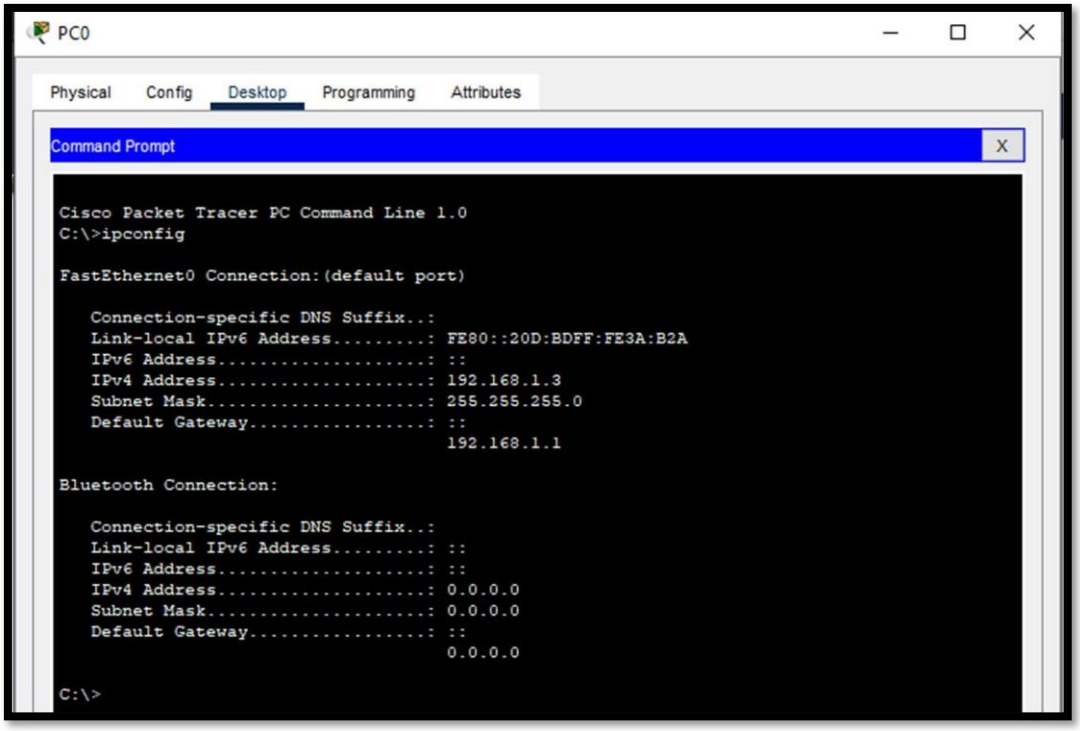
Step 4: Configure End Devices

- Click on each PC or end device.
- Go to 'Desktop' > 'IP Configuration'.
- Set the device to 'DHCP' to allow automatic IP assignment.



Step 5: Verify the DHCP Configuration

- After setting all devices to DHCP, check if they have received IP addresses from the DHCP pool:
- On each end device, go to 'Desktop' > 'Command Prompt' and type 'ipconfig' to verify the assigned IP addresses.



SMTP Configuration:

Step 1: Open Cisco Packet Tracer

- Launch the application to create your network

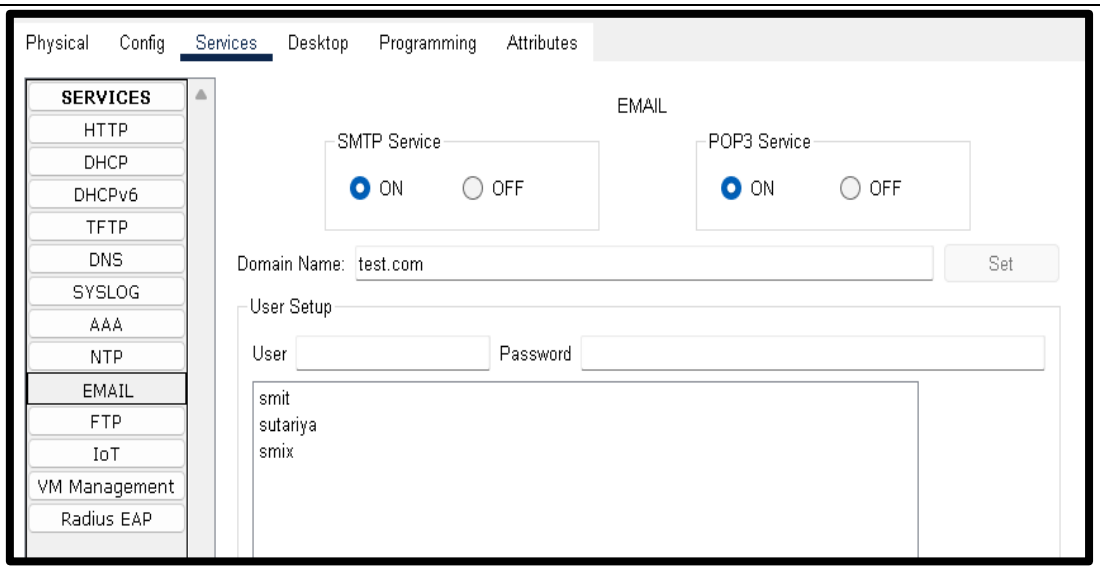
.

Step 2: Create the Network Topology

- Add a Server: Select 'End Devices' and drag and drop a generic server onto the workspace.
- Add Two Host Devices: Select 'End Devices' again and place two PCs or laptops onto the workspace.
- Add a Switch: Select 'Network Devices' > 'Switches' and drag and drop a switch (e.g., 2960) onto the workspace.
- Connect the Devices: Use the 'Connections' tool to connect each PC and the server to the switch.

Step 3: Configure the Server with SMTP

- Click on the server.
- Navigate to the 'Config' tab.
- Configure the server with a static IP address (e.g., 192.168.1.2) and subnet mask (e.g., 255.255.255.0) for 'FastEthernet0'.
- Turn the interface on.
- Go to the 'Services' tab, scroll down, and select 'Email' to configure SMTP:
- Ensure that the 'Service' is turned on.
- Add email users.



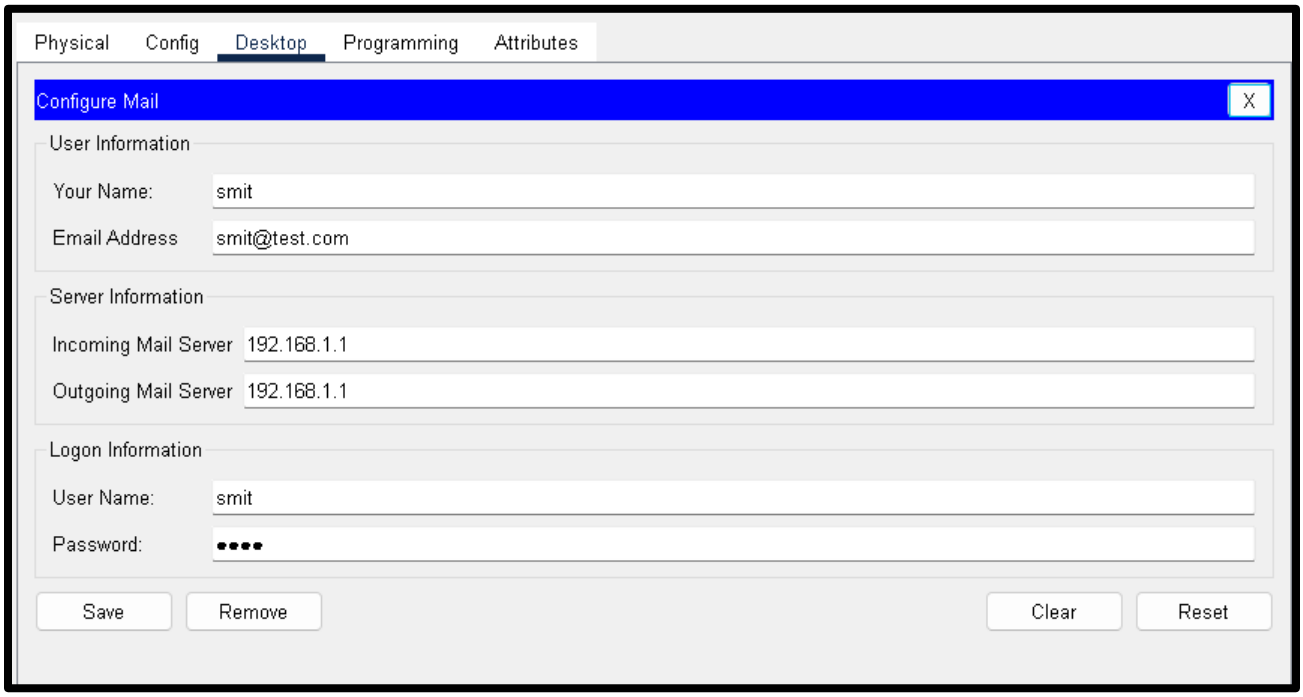
Step 4: Configure the Host Devices

- Click on each host device (PC).
- Navigate to 'Desktop' > 'IP Configuration'.
- Set each PC to 'DHCP' or manually assign static IP addresses in the same subnet as the server (e.g., 192.168.1.3 for PC1 and 192.168.1.4 for PC2).

Step 5: Test SMTP Configuration

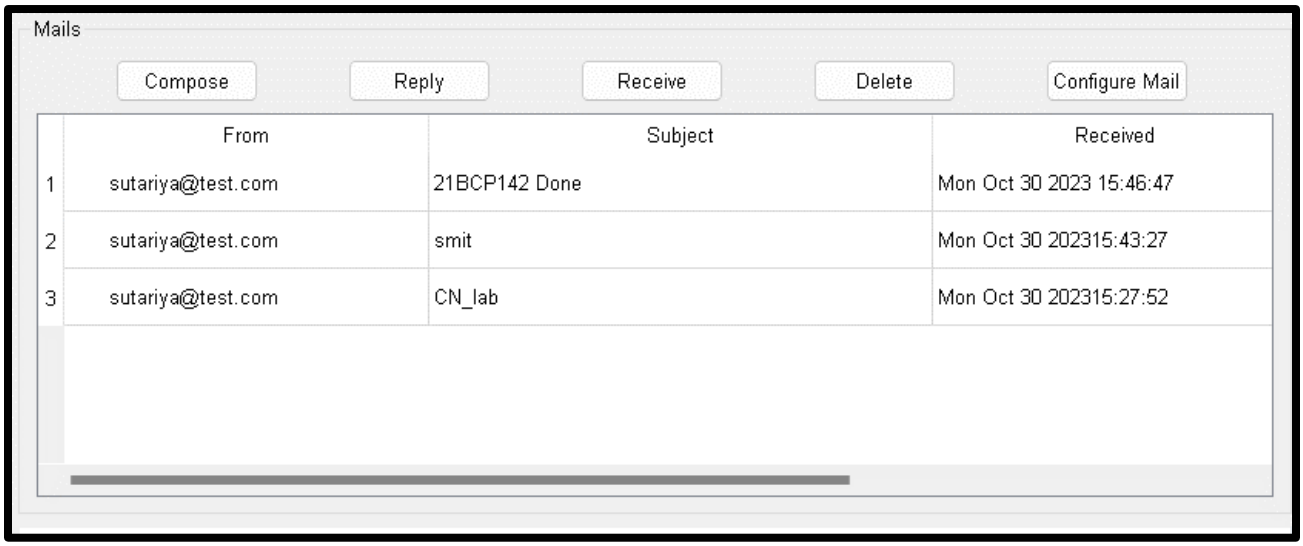
- Open a web browser on one of the PCs.
- Access the server's web service using its IP address.
- Log in with an email account created on the server.
- Compose an email and send it to the other user's email address created on the server.
- Verify the email by logging in to the recipient's email account through the server's web service.

For PC-1:

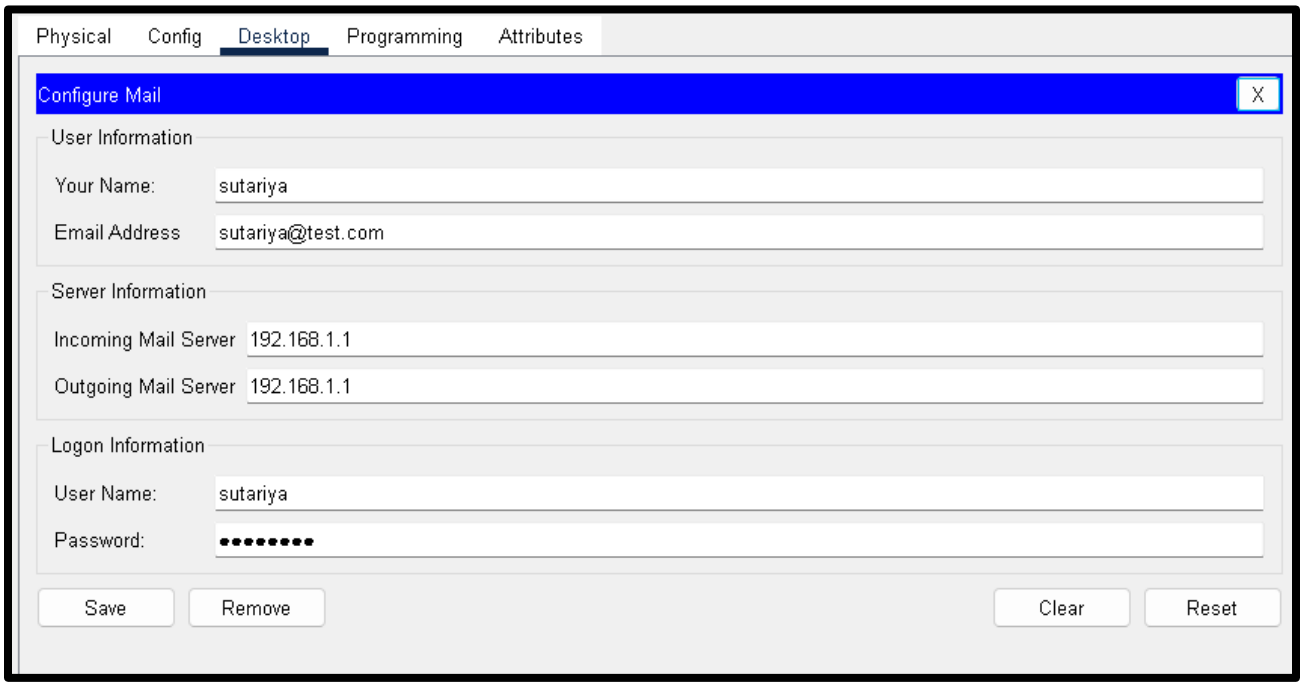


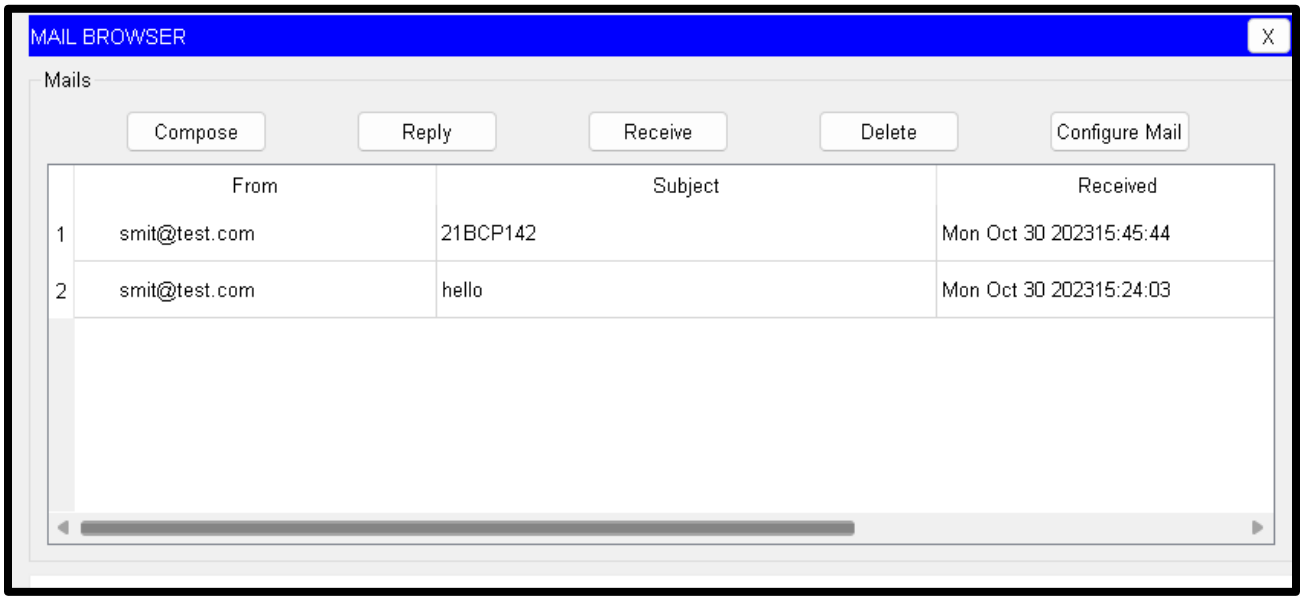
Step 6: Verify Email Receipt

- Switch to the other PC and access the email service via the server's IP address.
- Log in with the recipient's account details and check the inbox for the received email.



For PC-2:


Physical
Config
Desktop
Programming
Attributes
Configure Mail
X
User Information
Your Name:
sutariya
Email Address
sutariya@test.com
Server Information
Incoming Mail Server
192.168.1.1
Outgoing Mail Server
192.168.1.1
Logon Information
User Name:
sutariya
Password:
Save
Remove
Clear
Reset


MAIL BROWSER
X
Mails
Compose
Reply
Receive
Delete
Configure Mail
From
Subject
Received
1
smit@test.com
21BCP142
Mon Oct 30 202315:45:44
2
smit@test.com
hello
Mon Oct 30 202315:24:03

Observation & Learning:

This experiment provides practical experience in configuring both DHCP and SMTP in a network using Cisco Packet Tracer. Key observations and learning include understanding howto set up DHCP for automatic IP address assignment and configuring an email server with

SMTP for sending and receiving emails. It also involves troubleshooting network issues that may arise.

Conclusion:

This experiment has helped us gain hands-on knowledge of setting up crucial network services like DHCP and SMTP. DHCP simplifies IP address management, while SMTP facilitates email communication within a network. Both services are essential for the smooth operation of networks, and configuring them is a fundamental skill in computer networking.

Questions:

1. **What are the different ports used by the DHCP and SMTP?**
   Ans:
   DHCP typically uses UDP port 67 for the server and UDP port 68 for the client.
   SMTP uses TCP port 25 for unencrypted communication and TCP port 587 for encrypted communication.

2. **What are the benefits of using DHCP services?**
   Ans:
   DHCP automates the process of IP address assignment, reducing the risk of address conflicts.
   It simplifies network management and maintenance, making it easier to add or remove devices.
   DHCP allows for centralized control of IP address allocation, lease duration, and other network parameters.

3. **What is the role of Active Directory in the DHCP context?**
   Ans:
   Active Directory is a directory service used in Windows environments. In the context of DHCP, Active Directory integration can provide enhanced security and management capabilities.
   It allows DHCP servers to authenticate clients, ensuring that only authorized devices receive IP addresses.
   Active Directory can store DHCP configurations and lease information, providing centralized management and fault tolerance.

Smit Sutariya **21BCP142**

Practical 10

Part A

Class B Tech CSE 3rd Year **Sub : Computer Networks**

Aim:
Study of Wireshark and understand its functionality

Prerequisite: Nil

Outcome: To impart knowledge of Application Layer Protocol

Theory:
Wireshark is a widely used open-source network protocol analyser. It is a powerful tool for capturing, analysing, and inspecting network traffic in real-time. Wireshark allows network administrators, security professionals, and developers to examine the data exchanged between devices on a network. Here are some key aspects of Wireshark:

1. **Packet Capture:** Wireshark captures data packets as they travel over a network. It can capture packets from a variety of network interfaces, including wired and wireless connections.
2. **Protocol Analysis:** It decodes and analyses network protocols at various layers of the OSI model, from the physical layer to the application layer. This includes protocols like Ethernet, TCP/IP, HTTP, DNS, and many more.
3. **Real-Time Monitoring:** Wireshark provides real-time monitoring of network traffic, allowing users to see how data is transmitted, received, and processed.
4. **Filtering and Search**: Users can filter and search for specific packets based on various criteria, such as source/destination IP addresses, protocols, and data content. This helps in pinpointing specific network issues or security threats.
5. **Packet Inspection:** Wireshark allows users to inspect packet details, including headers, payloads, and various protocol-specific fields. This is invaluable for troubleshooting and analysing network behaviour.
6. **Export and Save**: Captured data can be saved and exported for further analysis, documentation, or sharing with colleagues or experts.
7. **Security Analysis:** Wireshark can be used to identify security vulnerabilities, anomalies, and malicious network activity by examining patterns and behaviours in network traffic.
8. **Education and Training:** Wireshark is commonly used for teaching network protocols and network analysis in educational settings. It helps students understand how data flows within a network.

Procedure:

1. Install the Wireshark and integrate it with network simulator
2. Analyse the traffic using Wireshark.

Part B

Steps:

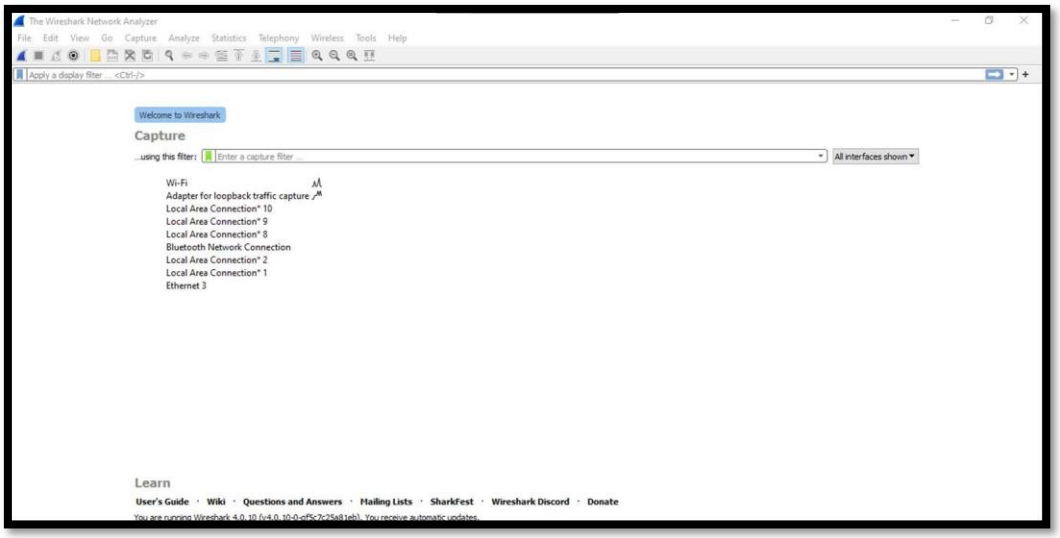
Installation of Wireshark:

```
Command Prompt
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>winget search wireshark
Name      Id                            Version  Source
-----------------------------------------------------------
Wireshark WiresharkFoundation.Wireshark 4.0.10.0 winget

C:\Users\Admin>
```

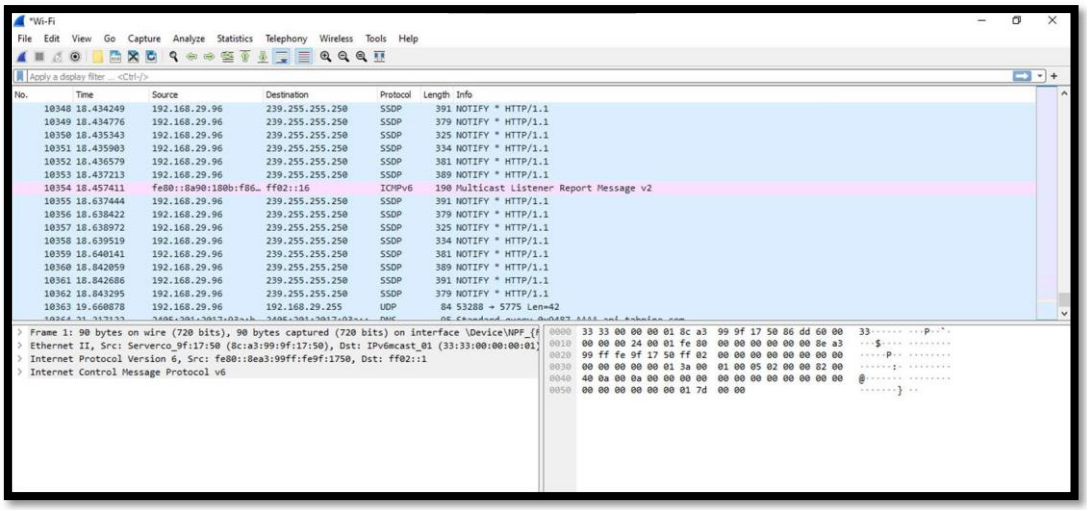
```
C:\Users\Admin>winget install WiresharkFoundation.Wireshark
Found an existing package already installed. Trying to upgrade the installed package...
Found Wireshark [WiresharkFoundation.Wireshark] Version 4.0.10.0
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://www.wireshark.org/download/win64/all-versions/Wireshark-win64-4.0.10.exe
  75.4 MB / 75.4 MB
Successfully verified installer hash
Starting package install...
Successfully installed
```

Traffic Analysis:

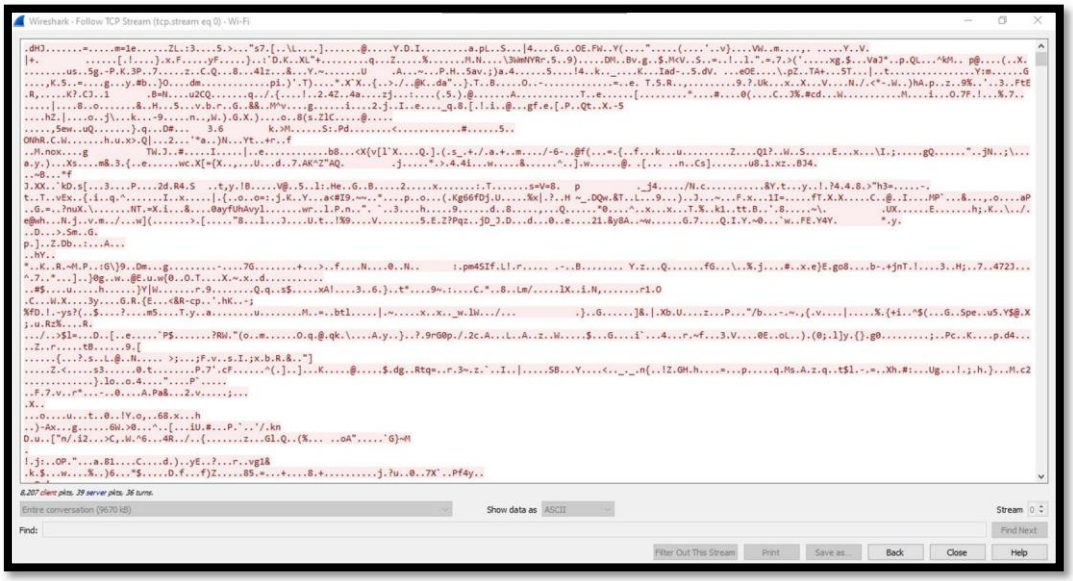
1. **Open Wireshark:** Launch Wireshark to view available network interfaces.



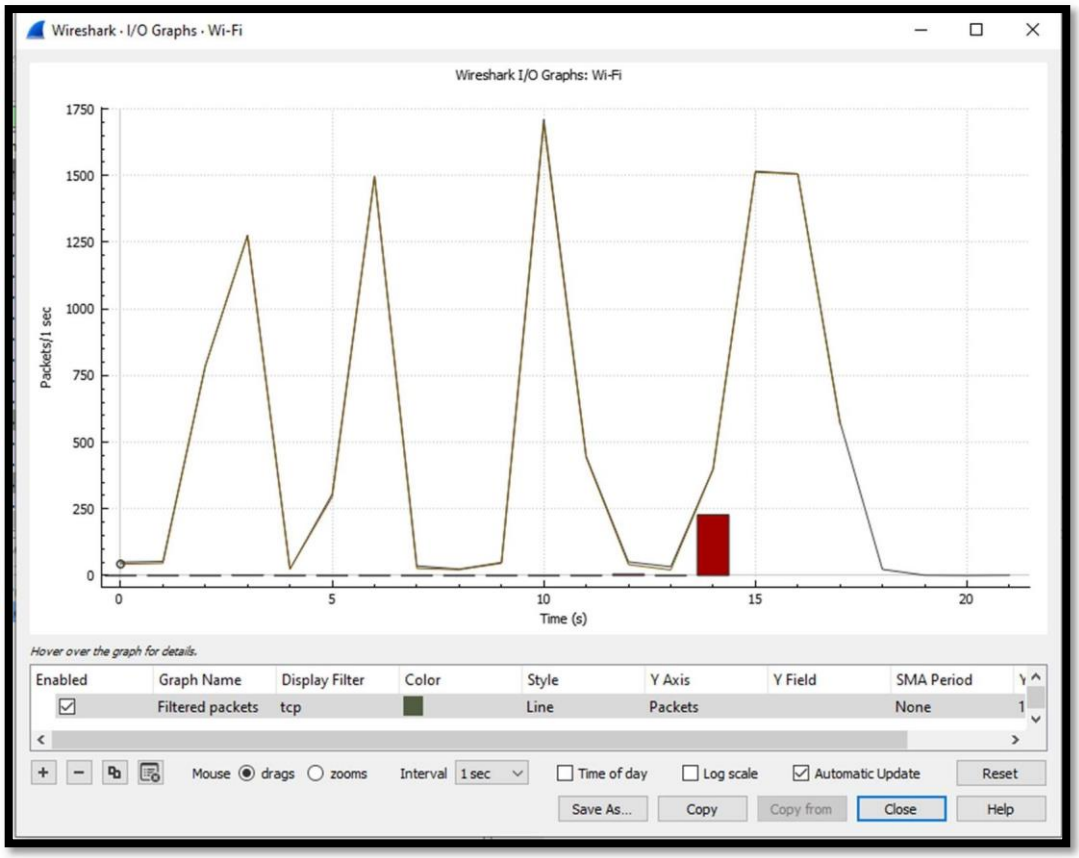
2. **Select Network Interface:** Choose the network interface you want to monitor, such as Ethernet or wireless.
3. **Start Capturing:** Click the shark fin icon to start capturing live traffic.
4. **Capture Traffic:** Allow Wireshark to capture traffic for the desired duration (seconds, minutes, etc.).
5. **Stop Capturing:** Click the red "Stop" button to halt the packet capture when you have enough data.



6. **Apply Display Filters:** Use the Display Filter bar to isolate specific traffic types, e.g., typing "http" shows only HTTP traffic.
7. **Analyse Packets:** Click on a packet to view detailed information, with packet details in the middle pane and raw data in the bottom pane.
8. **Use Colour Coding:** Color-coded packets represent different protocols or traffic types.
9. **Follow Streams:** Right-click a packet and select "Follow" (TCP, UDP) to see the entire conversation between endpoints.

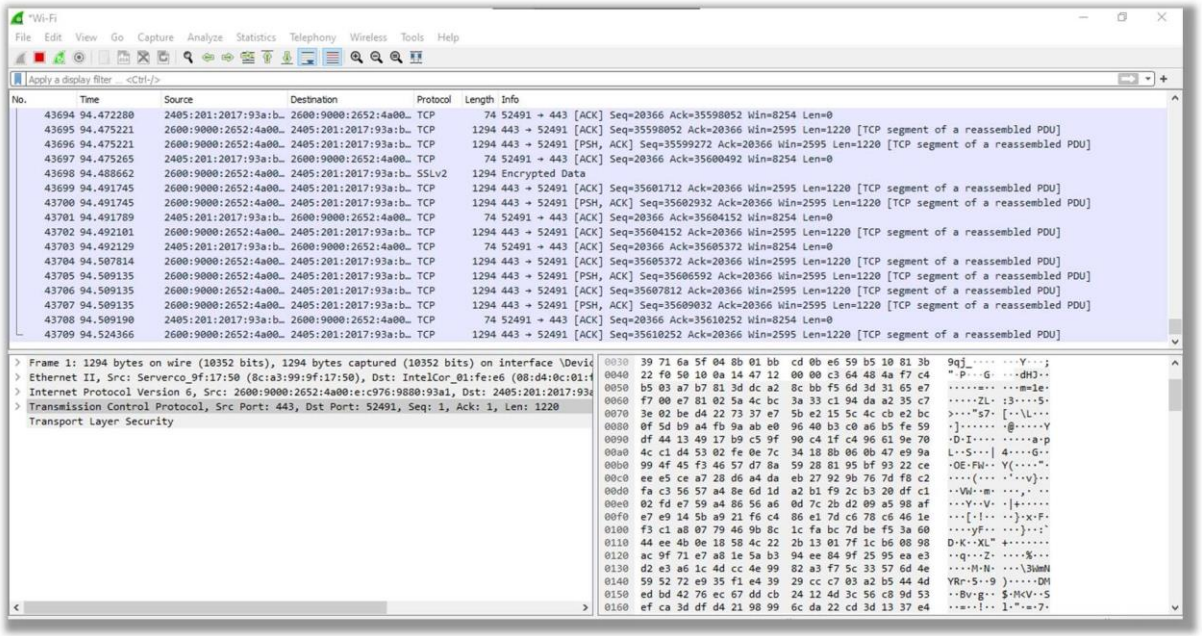


10. **Inspect Packet Details:** Examine the packet details pane to learn about each layer of network protocols within a selected packet.
11. **Save or Export Capture**: Save your packet capture for later analysis or documentation through "File" > "Save As."
12. **Use Statistics and Analysis Tools:** Explore the "Statistics" menu for graphical views, endpoint summaries, protocol hierarchy, and advanced analyses.



13. **Export Specific Packets**: Select packets or conversations for export using options under the "File" menu.
14. **Close Wireshark:** Close Wireshark when finished, with the option to save your session for future review.

Output:



Observation & Learning:

1. **Packet Capture:** Wireshark allows capturing and analysing network traffic from various sources.
2. **Real-time Analysis:** It offers real-time analysis, making it easier to identify and troubleshoot network issues as they occur.
3. **Extensive Protocol Support:** Wireshark supports a wide range of network protocols, allowing in-depth inspection of network traffic.
4. **Packet Decoding:** Wireshark dissects captured packets, providing human-readable information about the protocols and their fields.
5. **Packet Filtering:** Users can apply filters to narrow down the analysis to specific criteria, simplifying the focus on relevant traffic.

Conclusion:

Wireshark is a versatile and powerful tool for network analysis, valuable for network administrators, security professionals, and developers. It offers:

1. **Efficient Troubleshooting:** Real-time analysis and filtering help in identifying and resolving network issues promptly.
2. **In-Depth Protocol Understanding:** It supports various network protocols and provides deep insights into their behaviours.
3. **Network Performance Monitoring:** Wireshark's statistics and display features aid in monitoring network performance.
4. **Security Incident Investigation:** Security professionals can use Wireshark to investigate incidents and assess vulnerabilities.
5. **Customization and Automation:** Extensible functionality through scripting enhances versatility for specific tasks.
6. **Multi-Platform Availability:** It's accessible on various operating systems, making it widely available to users.