



**St. John College of Engineering and Management**

**Autonomous Institute**

**(A Christian Religious Minority Institution)**

Approved by AICTE and DTE, Affiliated to University of Mumbai / MSBTE

DTE Code : 3218 AICTE Permanent ID : 1-4790201

NAAC Accredited with Grade 'A+', Three Programs NBA Accredited



**A.Y. 2025-26**

## **Report on Utilization of Simulation Tool Wireshark for Packet Sniffer Tools**

**Subject: Computer Network Security**

**Subject Code: 24ITPCC502**

### **1. Introduction**

In the domain of Computer Network Security, understanding network traffic is a crucial skill. Packet sniffing tools allow security professionals to capture, analyze, and troubleshoot packets transmitted over a network. Wireshark, one of the most popular open-source packet analyzers, is widely used for this purpose. This report provides a detailed exploration of Wireshark, its installation, features, and utilization in network security.

### **2. Objectives**

- To understand the role of packet sniffing tools in network security.
- To explore Wireshark as a simulation tool for packet capture and analysis.
- To demonstrate the usage of Wireshark with examples.
- To evaluate the benefits of Wireshark in detecting network vulnerabilities.

### **3. Overview of Wireshark**

Wireshark is a network protocol analyzer that enables users to capture and interactively browse the traffic running on a computer network. Wireshark is available for Windows, Linux, and macOS platforms.

### **4. Features of Wireshark**

- Real-time capture and offline analysis.
- Rich display filters for detailed packet examination.
- Support for hundreds of network protocols

### **5. Installation of Wireshark**

Wireshark can be downloaded from its official website (<https://www.wireshark.org/>).

### **6. Utilization of Wireshark**

Wireshark can be used for the following tasks:

- Capturing live network traffic.
- Analyzing protocol behavior.
- Identifying malicious packets.
- Debugging network issues.
- Learning network protocols.



## St. John College of Engineering and Management

Autonomous Institute

(A Christian Religious Minority Institution)

Approved by AICTE and DTE, Affiliated to University of Mumbai / MSBTE

DTE Code : 3218 AICTE Permanent ID : 1-4790201

NAAC Accredited with Grade 'A+', Three Programs NBA Accredited



### 7. Demonstration Steps

Step 1: Launch Wireshark and select the active network interface.

Step 2: Start capturing packets in real-time.

Step 3: Use filters (e.g., 'http', 'ip.addr==192.168.1.1') to analyze specific traffic.

Step 4: Stop the capture and examine detailed packet information.

Step 5: Export captured data for reporting.

**FOR ANY HELP AND QUERIES VISIT: [https://www.wireshark.org/docs/wsug\\_html/](https://www.wireshark.org/docs/wsug_html/)**

### 8. Sample Screenshots

Below are some sample screenshots representing Wireshark utilization:

### 9. Benefits in Network Security

- Helps detect suspicious activity and intrusions.
- Assists in monitoring bandwidth usage.
- Useful in protocol analysis and education.
- Provides insights into vulnerabilities and misconfigurations.

### 10. Conclusion

Wireshark is an essential tool for network security professionals and learners. Its ability to capture and analyze network traffic in real-time makes it invaluable for troubleshooting, protocol learning, and detecting potential threats. The utilization of Wireshark in Computer Network Security (24ITPCC502) equips students with practical skills necessary for the modern cybersecurity landscape.



**St. John College of Engineering and Management**

**Autonomous Institute**

**(A Christian Religious Minority Institution)**

Approved by AICTE and DTE, Affiliated to University of Mumbai / MSBTE

DTE Code : 3218 AICTE Permanent ID : 1-4790201

NAAC Accredited with Grade 'A+', Three Programs NBA Accredited



**A.Y. 2025-26**

## **Report on Utilization of Simulation Tool -Cooja Simulator**

**Subject: Internet of Everything**

**Subject Code: 24ITPCC702**

### **1. Introduction**

The Internet of Everything (IoE) is an extension of the Internet of Things (IoT) that integrates people, data, processes, and things to create an intelligent and interconnected system. To study, test, and validate IoE concepts, network simulators play a crucial role. COOJA, a popular simulator bundled with the Contiki Operating System, is widely used for simulating wireless sensor networks (WSNs) and IoT-based applications.

### **2. COOJA Simulator Overview**

COOJA is a Java-based network simulator that allows the simulation of networks of Contiki nodes. It provides a flexible platform for testing low-power wireless protocols such as IEEE 802.15

### **3. Key Features of COOJA**

- Simulation of various node types (Sky, Z1, Cooja nodes)
- Radio message logging and interference modeling
- Real-time and emulation-based simulation modes

### **4. Utilization of COOJA in Internet of Everything (IoE)**

COOJA plays a vital role in IoE simulations as it allows experimentation with different communication protocols, topologies, and device configurations. Typical IoE use cases simulated in COOJA include:

1. Smart Cities: Testing communication among traffic sensors, streetlights, and emergency systems.
2. Healthcare: Simulating body sensor networks for remote patient monitoring.
3. Industrial IoE: Evaluating reliability and latency of wireless communication in industrial automation.
4. Smart Homes: Simulating IoT devices like thermostats, alarms, and appliances communicating via RPL.

Using COOJA, researchers can measure network parameters such as packet delivery ratio (PDR), energy consumption, throughput, and latency to validate IoE solutions before real-world deployment.

### **5. Steps to Use COOJA Simulator**

1. Install Contiki OS and launch COOJA.
2. Create a new simulation environment.





# St. John College of Engineering and Management

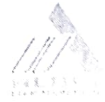
Autonomous Institute

(A Christian Religious Minority Institution)

Approved by AICTE and DTE, Affiliated to University of Mumbai / MSBTE

DTE Code : 3218 AICTE Permanent ID : 1-4790201

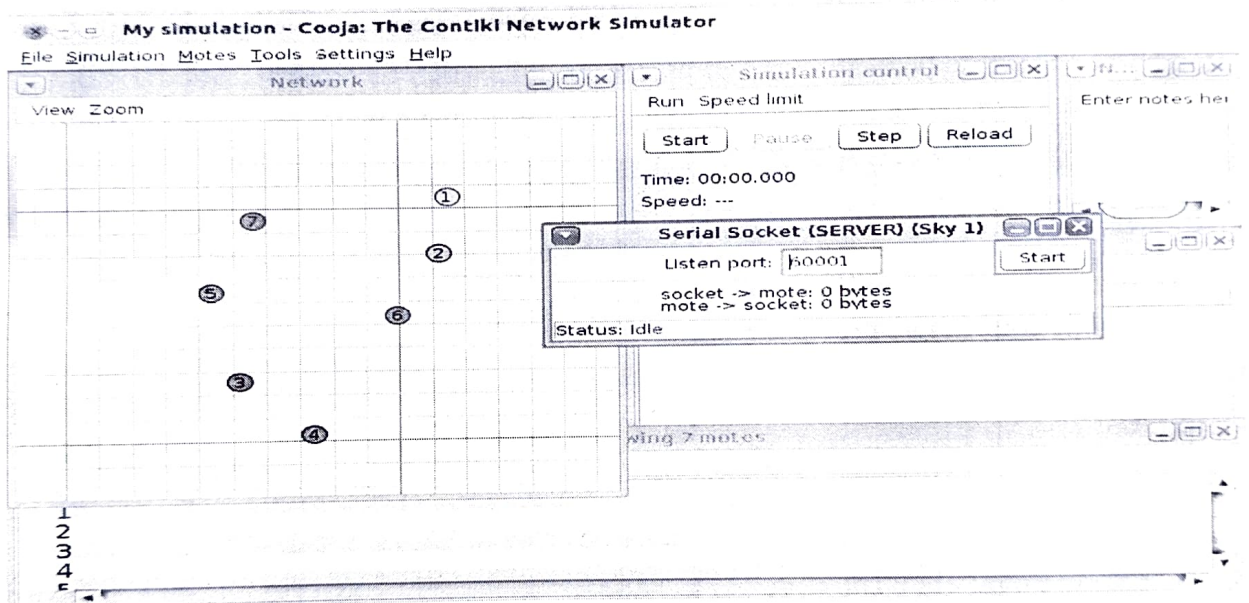
NAAC Accredited with Grade 'A+', Three Programs NBA Accredited



3. Add motes (Sky, Z1, or Cooja motes).
4. Configure wireless communication protocols.
5. Start the simulation and observe communication through the timeline and radio messages window.
6. Analyze results using built-in logging tools.

## 6. Screenshots of COOJA Simulator

Below are sample screenshots of COOJA Simulator in use:



## 7. Conclusion

The COOJA simulator is a powerful tool for simulating and analyzing IoE applications in a controlled environment. Its ability to emulate real hardware, test communication protocols, and provide detailed insights makes it highly valuable for both academic learning and research. By using COOJA, students and researchers can validate IoE solutions before large-scale deployment, thus saving time, cost, and resources.



## Experiment No:6

Aim: Use simulator (e.g., NS2) to understand functioning of ALOHA, CSMA/CD.

Network simulators like **NS2 (Network Simulator 2)** help analyze and understand network protocols such as **ALOHA** and **CSMA/CD** by simulating real-world networking scenarios. These simulations allow researchers and students to study protocol behavior, collision handling, and efficiency without requiring a physical network setup.

### 1. Understanding ALOHA and CSMA/CD

#### A. ALOHA (Pure and Slotted ALOHA)

ALOHA is a simple **random access protocol** where nodes transmit data without checking the medium status. If a **collision** occurs, the sender waits for a random time before retransmitting.

- **Pure ALOHA:** Nodes transmit anytime, leading to more collisions.
- **Slotted ALOHA:** Nodes transmit in fixed time slots, reducing collisions.
- **Disadvantage:** Low efficiency due to frequent collisions.
- **Efficiency:** 18.4% for Pure ALOHA, 36.8% for Slotted ALOHA.

#### B. CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

CSMA/CD is a **contention-based Ethernet protocol** that minimizes collisions by sensing the channel before transmission.

- If the medium is **idle**, a node transmits data.
- If two nodes transmit simultaneously, a **collision** occurs.
- The nodes stop transmission, send a **jam signal**, and wait for a **random backoff time** before retransmitting.
- **Efficiency:** Higher than ALOHA due to carrier sensing and collision detection.

### 2. Why Use NS2 for Simulation?

- **Cost-effective:** No need for physical hardware.
- **Performance analysis:** Helps in studying **throughput, delay, and collision rate**.
- **Customizable:** Users can define **network topologies, protocols, and traffic models**.

- **Visualization:** Tools like **NAM (Network Animator)** help in graphical analysis of network behavior.

### 3. Simulating ALOHA in NS2

We can use **TCL scripting** in NS2 to simulate ALOHA. The script will:

1. Create **two nodes** and establish a **wireless link**.
2. Assign **UDP traffic** to one node.
3. Send packets **without checking for collisions** (simulating ALOHA behavior).
4. Log results in a **trace file**.

#### 1. Setting Up NS2

Before running the simulation, ensure you have **NS2 installed** on your Linux system. If you haven't installed NS2 yet, you can do so using the following commands:

```
sudo apt update
```

```
sudo apt install ns2
```

```
ns
```

If NS2 is installed, it will display the NS2 version.

#### 2. Simulating ALOHA in NS2

ALOHA is a simple **random access protocol** used for wireless networks. In NS2, we can simulate ALOHA using a simple TCL script.

##### TCL Script for ALOHA Simulation

Create a new file named **aloha.tcl** and add the following code:

```
# Create a simulator instance
```

```
set ns [new Simulator]
```

```
# Define a trace file to capture output
```

```
set tracefile [open aloha.tr w]
```

```
$ns trace-all $tracefile
```

```
# Create two nodes
```

```
set n1 [$ns node]
```

```
set n2 [$ns node]
```

```
# Create a wireless link between nodes
```

```
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
```

```
# Define an ALOHA-based traffic source
```

```
set udp [new Agent/UDP]
```

```
$ns attach-agent $n1 $udp
```

```
set cbr [new Application/Traffic/CBR]
```

```
$cbr attach-agent $udp
```

```
$cbr set packetSize_ 512
```

```
$cbr set rate_ 100Kb
```

```
# Define a sink agent at the receiving node
```

```
set null [new Agent/Null]
```

```
$ns attach-agent $n2 $null
```

```
$ns connect $udp $null
```

```
# Schedule events
```

```
$ns at 0.5 "$cbr start"
```

```
$ns at 5.0 "finish"
```

```
# Define the finish procedure
```

```
proc finish {} {
```

```
    global ns tracefile
```

```
    $ns flush-trace
```



```
close $tracefile  
exit 0  
}
```

# Run the simulation

```
$ns run
```

Running the ALOHA Simulation

```
ns aloha.tcl
```

This will generate a trace file **aloha.tr**, which you can analyze to understand **packet collisions** and **throughput**.

### 3. Simulating CSMA/CD in NS2

Carrier Sense Multiple Access with Collision Detection (**CSMA/CD**) is used in Ethernet networks to manage access to the transmission medium.

#### TCL Script for CSMA/CD Simulation

Create a file named **csma\_cd.tcl** and add the following code:

```
# Create a new simulator instance
```

```
set ns [new Simulator]
```

```
# Define a trace file
```

```
set tracefile [open csma_cd.tr w]
```

```
$ns trace-all $tracefile
```

```
# Define nodes
```

```
set n0 [$ns node]
```

```
set n1 [$ns node]
```

```
set n2 [$ns node]
```

```
# Define links (Ethernet-based)
```

```
$ns duplex-link $n0 $n1 10Mb 5ms DropTail
```

```
$ns duplex-link $n1 $n2 10Mb 5ms DropTail
```

```
# Attach traffic sources
```

```
set udp0 [new Agent/UDP]
```

```
$ns attach-agent $n0 $udp0
```

```
set cbr0 [new Application/Traffic/CBR]
```

```
$cbr0 attach-agent $udp0
```

```
$cbr0 set packetSize_ 512
```

```
$cbr0 set rate_ 500Kb
```

```
# Attach sink agents
```

```
set null0 [new Agent/Null]
```

```
$ns attach-agent $n2 $null0
```

```
$ns connect $udp0 $null0
```

```
# Schedule traffic
```

```
$ns at 1.0 "$cbr0 start"
```

```
$ns at 6.0 "finish"
```

```
# Define finish procedure
```

```
proc finish {} {
```

```
    global ns tracefile
```

```
    $ns flush-trace
```

```
    close $tracefile
```

```
    exit 0
```

```
}
```

# Run the simulation

\$ns run

### Running the CSMA/CD Simulation

Save the file and run:

```
ns csma_cd.tcl
```

This will generate a trace file **csma\_cd.tr**, where you can analyze **packet transmission, collisions, and retransmissions**.

### 4. Analyzing the Results

To analyze the trace file and extract useful metrics, you can use **awk** or Python scripts.

For example, to count the number of **packet collisions**, use:

```
grep -c "COLLISION" csma_cd.tr
```

To visualize network behavior, you can use **NAM (Network Animator)**:

```
nam csma_cd.nam
```

This will open a GUI showing the **packet flow, collisions, and retransmissions**.

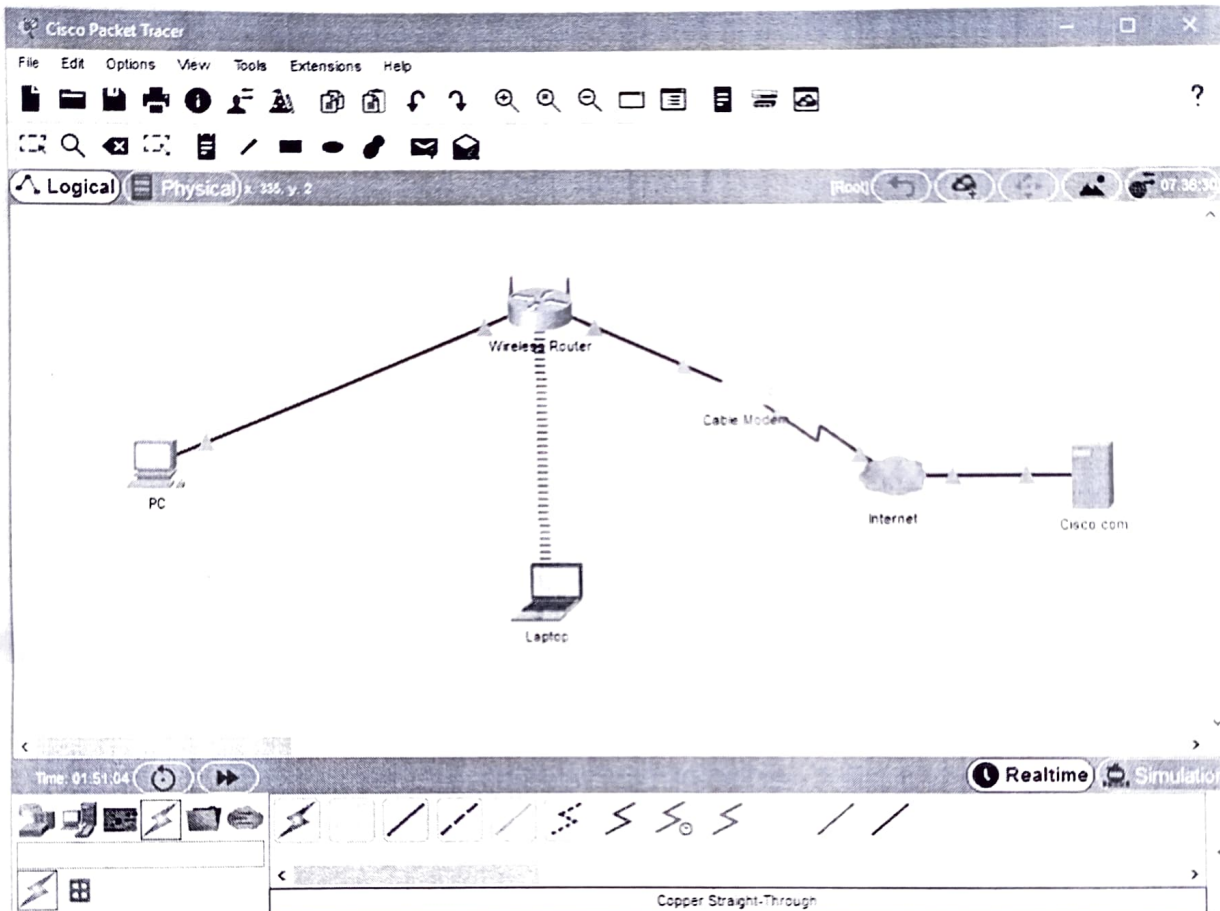
### Conclusion

- **ALOHA** allows random access but has **higher collisions** and lower efficiency.
- **CSMA/CD** senses the medium before transmission and detects collisions, improving **Ethernet network efficiency**.
- Using **NS2**, you can **simulate, analyze, and visualize** the network behavior for both protocols.



## Packet Tracer – Create a Simple Network Using Packet Tracer

### Topology



### Addressing Table

| Device           | Interface | IP Address     | Subnet Mask   | Default Gateway |
|------------------|-----------|----------------|---------------|-----------------|
| PC               | Ethernet0 | DHCP           |               | 192.168.0.1     |
| Wireless Router  | LAN       | 192.168.0.1    | 255.255.255.0 |                 |
| Wireless Router  | Internet  | DHCP           |               |                 |
| Cisco.com Server | Ethernet0 | 208.67.220.220 | 255.255.255.0 |                 |
| Laptop           | Wireless0 | DHCP           |               |                 |

## Objectives

Part 1: Build a Simple Network in the Logical Topology Workspace

Part 2: Configure the Network Devices

Part 3: Test Connectivity between Network Devices

Part 4: Save the File and Close Packet Tracer

## Background / Scenario

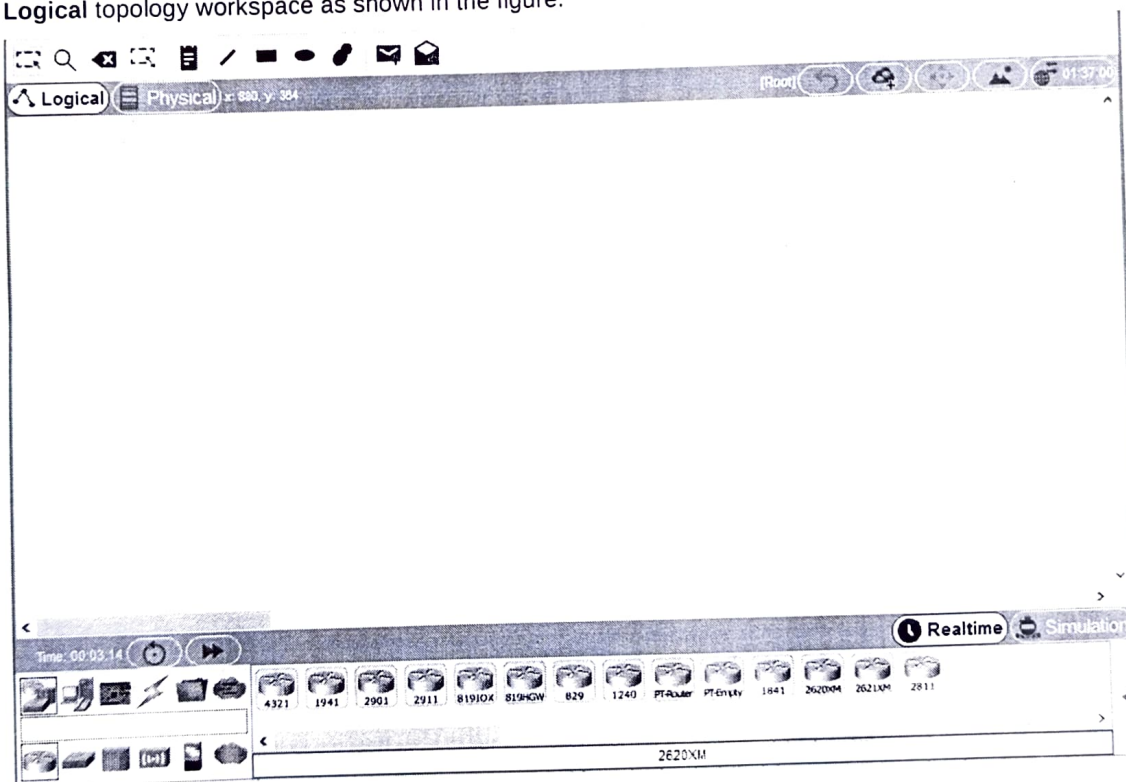
In this activity you will build a simple network in Packet Tracer from scratch and then save the network as a Packet Tracer Activity File (.pkt).

## Part 1: Build a Simple Network in the Logical Topology Workspace

### Step 1: Launch Packet Tracer.

- a. Launch Packet Tracer on your PC or laptop computer

Double click on the **Packet Tracer** icon on your desktop or navigate to the directory that contains the Packet Tracer executable file and launch Packet Tracer. Packet Tracer should open with a blank default Logical topology workspace as shown in the figure.



### Step 2: Build the topology

- a. Add network devices to the workspace.

Using the device selection box, add the network devices to the workspace as shown in the topology diagram.

To place a device onto the workspace, first choose a device type from the **Device-Type Selection** box. Then, click on the desired device model from the **Device-Specific Selection** box. Finally, click on a location in the workspace to put your device in that location. If you want to cancel your selection, click the **Cancel** icon for that device. Alternatively, you can click and drag a device from the **Device-Specific Selection** box onto the workspace.

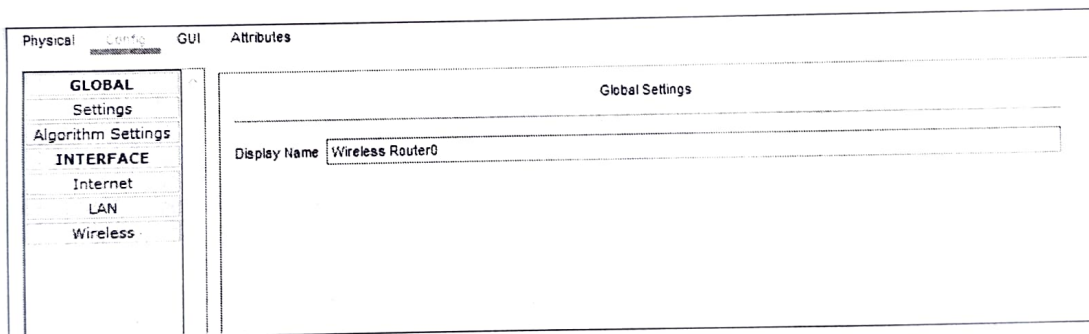
- b. Add network devices to the workspace.

Using the device selection box, add the network devices to the workspace as shown in the topology diagram

To place a device onto the workspace, first choose a device type from the **Device-Type Selection** box. Then, click on the desired device model from the **Device-Specific Selection** box. Finally, click on a location in the workspace to put your device in that location. If you want to cancel your selection, click the **Cancel** icon for that device. Alternatively, you can click and drag a device from the **Device-Specific Selection** box onto the workspace.

- c. Change display names of the network devices.

To change the display names of the network devices click on the device icon on the Packet Tracer **Logical** workspace, then click on the **Config** tab in the device configuration window. Type the new name of the device into the **Display Name** box as show in the figure below.



- d. Add the physical cabling between devices on the workspace

Using the device selection box, add the physical cabling between devices on the workspace as shown in the topology diagram.

The PC will need a copper straight-through cable to connect to the wireless router. Select the copper straight-through cable in the device selection box and attach it to the FastEthernet0 interface of the PC and the Ethernet 1 interface of the wireless router.



The wireless router will need a copper straight-through cable to connect to the cable modem. Select the copper straight-through cable in the device-selection box and attach it to the Internet interface of the wireless router and the Port 1 interface of the cable modem.

The cable modem will need a coaxial cable to connect to the Internet cloud. Select the coaxial cable in the device-selection box and attach it to the Port 0 interface of the cable modem and the coaxial interface of the Internet cloud.

The Internet cloud will need copper straight-through cable to connect to the Cisco.com server. Select the copper straight-through cable in the device-selection box and attach it to the Ethernet interface of the Internet cloud and the FastEthernet0 interface of the Cisco.com server.

## Part 2: Configure the Network Devices

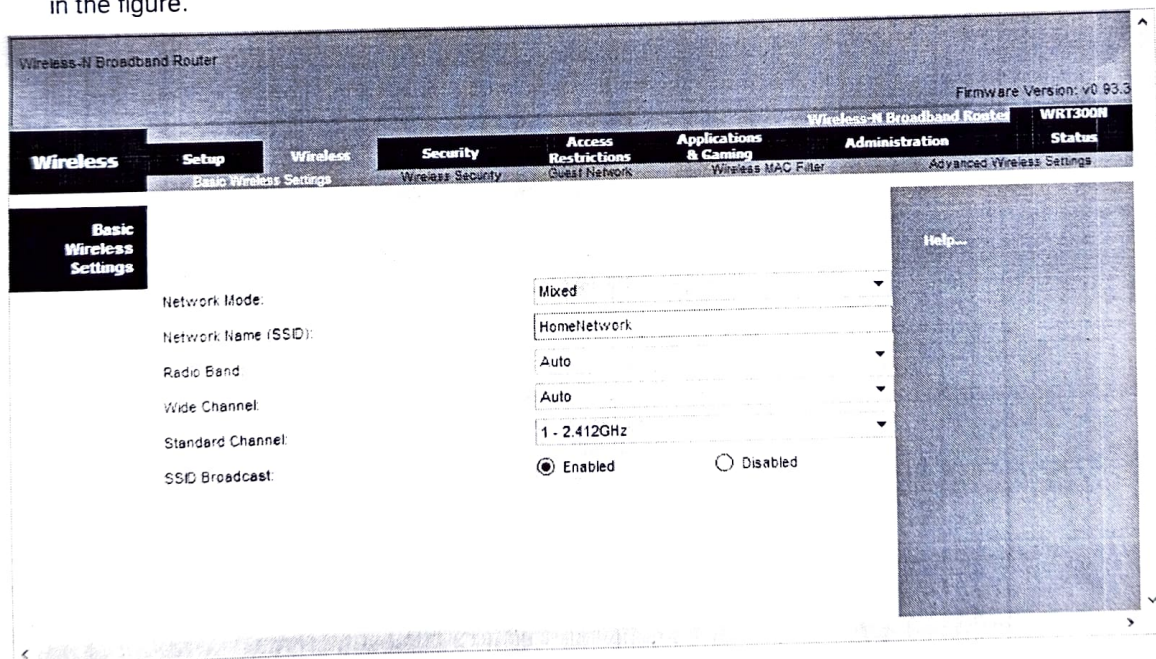
### Step 1: Configure the wireless router

- Create the wireless network on the wireless router

Click on the **Wireless Router** icon on the Packet Tracer **Logical** workspace to open the device configuration window.

In the wireless router configuration window, click on the **GUI** tab to view configuration options for the wireless router.

Next, click on the **Wireless** tab in the GUI to view the wireless settings. The only setting that needs to be changed from the defaults is the **Network Name (SSID)**. Here, type the name "HomeNetwork" as shown in the figure.



Configure the Internet connection on the wireless router

Click on the **Setup** tab in the wireless router GUI.

In the **DHCP Server** settings verify that the **Enabled** button is selected and configure the static IP address of the DNS server as 208.67.220.220 as shown in the figure.

- b. Click on the **Save Settings** tab.

The screenshot shows the configuration interface for a Wireless-N Broadband Router (Firmware Version: v0.93.3). The **Setup** tab is selected, and the **Internet Setup** section is active. The Internet Connection type is set to **Automatic Configuration - DHCP**. Under **Optional Settings**, the Host Name and Domain Name are empty, and the MTU is set to 1500. The **Network Setup** section shows the Router IP as 192.168.0.1 with a Subnet Mask of 255.255.255.0. The **DHCP Server Settings** are configured with the **DHCP Server** **Enabled** (radio button selected), **DHCP Reservation** disabled, **Start IP Address** 192.168.0.100, **Maximum number of Users** 50, and **IP Address Range** 192.168.0.100 - 149. The **Client Lease Time** is 0 minutes. The **Static DNS** settings are configured with **Static DNS 1** as 208.67.220.220, and **Static DNS 2**, **Static DNS 3**, and **WINS** are all set to 0.

## Step 2: Configure the laptop

- a. Configure the Laptop to access the wireless network

Click on the Laptop icon on the Packet Tracer **Logical** workspace and in the laptop configuration windows select the **Physical** tab.

In the **Physical** tab you will need to remove the Ethernet copper module and replace it with the Wireless WPC300N module.

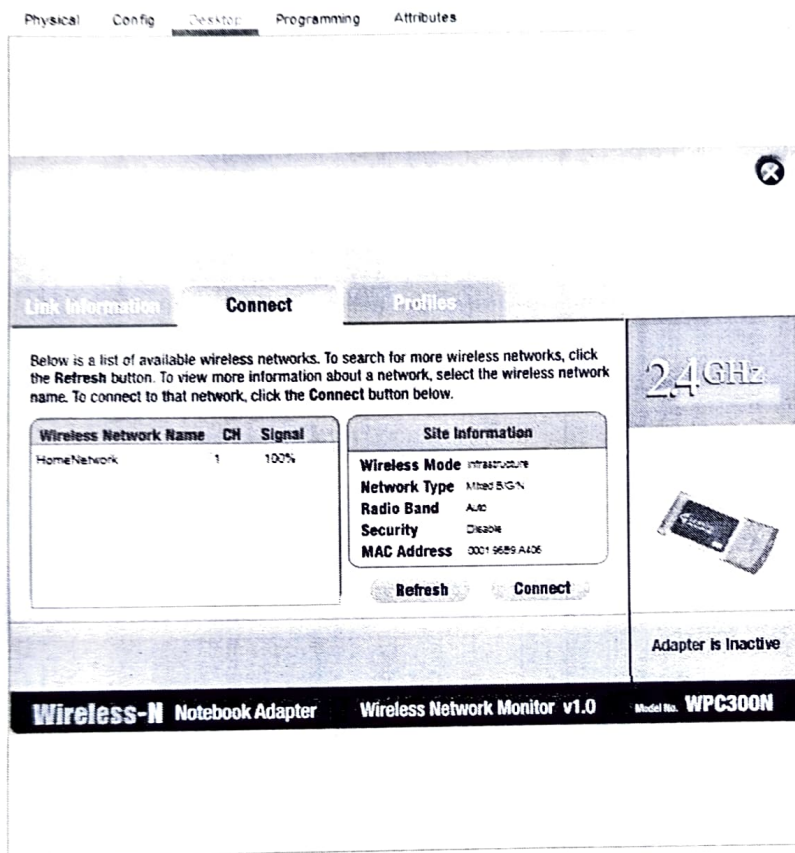
To do this, you first power the Laptop off by clicking the power button on the side of the laptop. Then remove the currently installed Ethernet copper module by clicking on the module on the side of the laptop and dragging it to the **MODULES** pane on the left of the laptop window. Then install the Wireless WPC300N module by clicking on it in the **MODULES** pane and dragging it to the empty module port on the side of the laptop. Power the laptop back on by clicking on the Laptop power button again.



With the wireless module installed, the next task is to connect the laptop to the wireless network.

Click on the **Desktop** tab at the top of the Laptop configuration window and select the **PC Wireless** icon. Once the Wireless-N Notebook Adapter settings are visible, select the **Connect** tab. The wireless network "HomeNetwork" should be visible in the list of wireless networks as shown in the figure.

Select the network, and click on the **Connect** tab found below the **Site Information** pane.



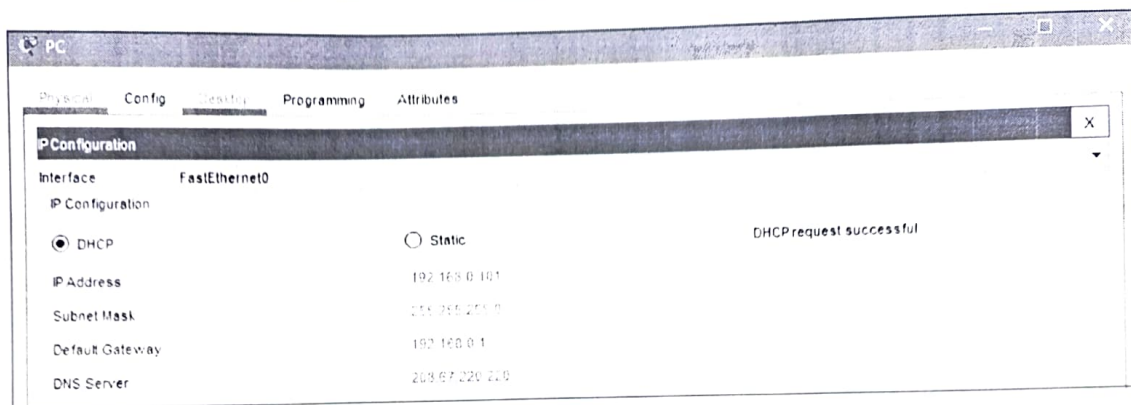
## Step 3: Configure the PC

- Configure the PC for the wired network

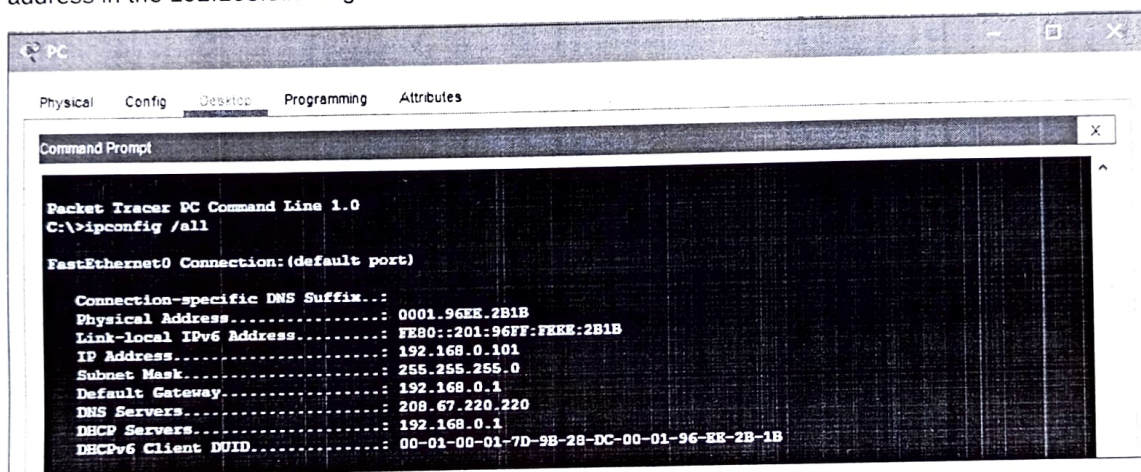
Click on the **PC** icon on the Packet Tracer **Logical** workspace and select the **Desktop** tab and then the **IP Configuration** icon.

In the IP Configuration window, select the **DCHP** radio button as shown in the figure so that the PC will use DHCP to receive an IPv4 address from the wireless router. Close the IP Configuration window.





Click on the Command Prompt icon. Verify that the PC has received an IPv4 address by issuing the **ipconfig /all** command from the command prompt as shown in the figure. The PC should receive an IPv4 address in the 192.168.0.x range.



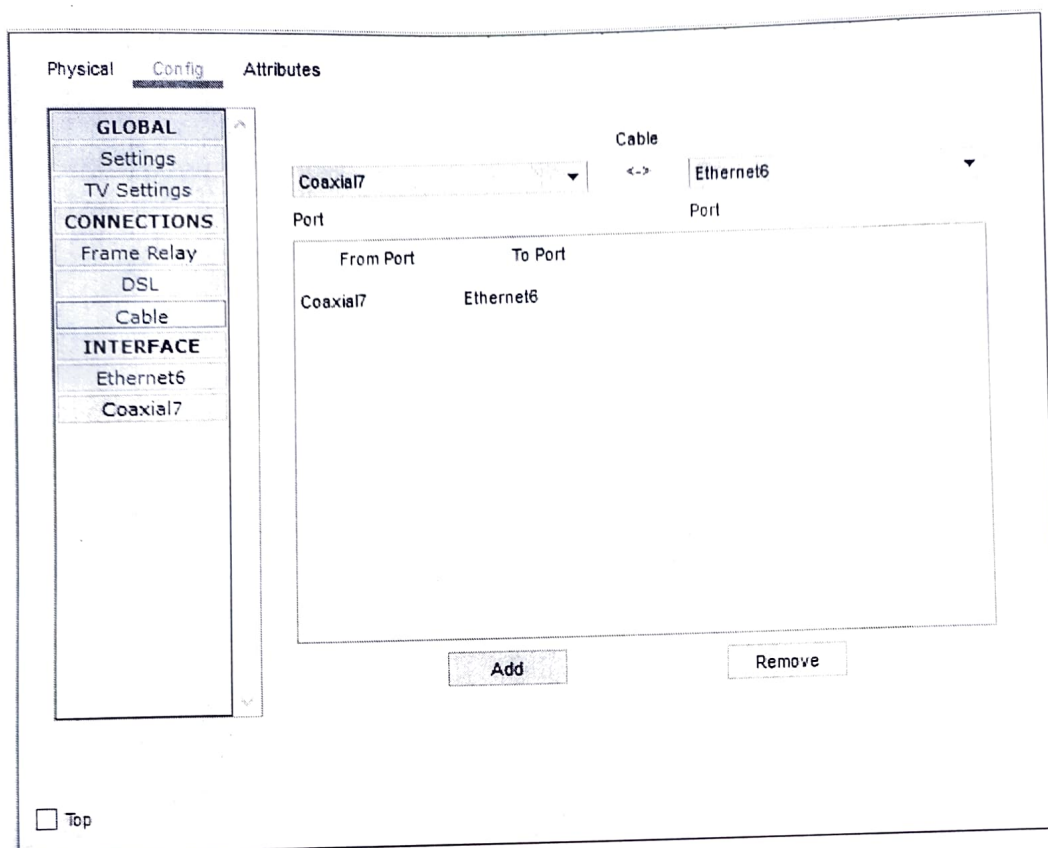
## Step 4: Configure the Internet cloud

- a. Install network modules if necessary

Click on the **Internet Cloud** icon on the Packet Tracer **Logical** workspace and then click on the **Physical** tab. The cloud device will need two modules if they are not already installed. The PT-CLOUD-NM-1CX which is for the cable modem service connection and the PT-CLOUD-NM-1CFE which is for a copper Ethernet cable connection. If these modules are missing, power off the physical cloud devices by clicking on the power button and drag each module to an empty module port on the device and then power the device back on.

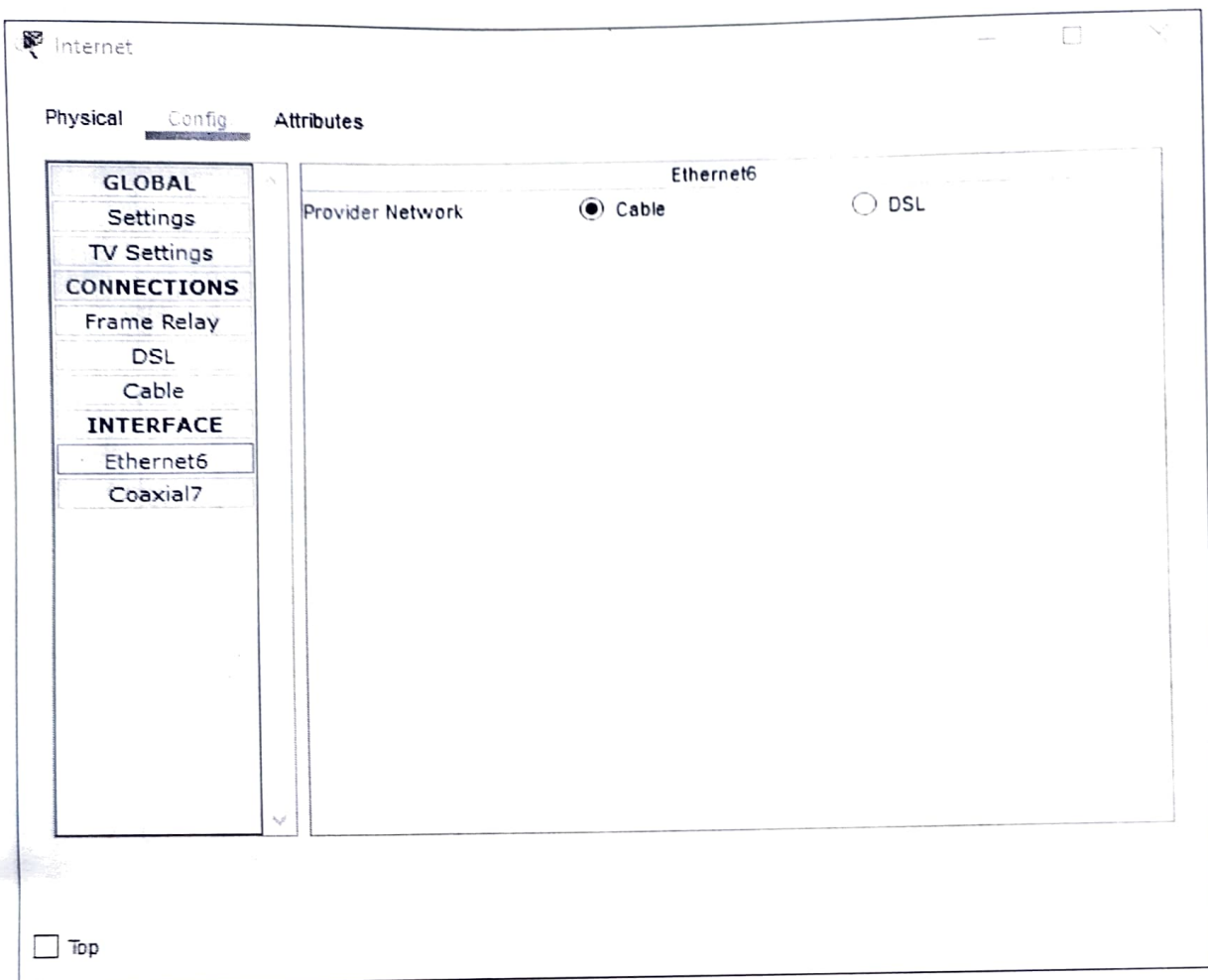
- b. Identify the From and To Ports

Click on the **Config** tab in the Cloud device window. In the left pane click on **Cable** under **CONNECTIONS**. In the first drop down box choose Coaxial and in the second drop down box choose Ethernet then click the **Add** button to add these as the **From Port** and **To Port** as shown in the figure.



- c. Identify the type of provider

While still in the **Config** tab click Ethernet under **INTERFACE** in the left pane. In the Ethernet configuration window select **Cable** as the Provider Network as shown in the figure.



### Step 5: Configure the Cisco.com server

- a. Configure the Cisco.com server as a DHCP server

Click on the Cisco.com server icon on the Packet Tracer **Logical** workspace and select the **Services** tab. Select **DHCP** from the **SERVICES** list in the left pane.

In the DHCP configuration window, configure a DHCP as shown in the figure with the following settings.

- Click **On** to turn the DHCP service on
- Pool name: DHCPpool
- Default Gateway: 208.67.220.220
- DNS Server: 208.67.220.220
- Starting IP Address: 208.67.220.1
- Subnet Mask 255.255.255.0
- Maximum number of Users: 50

Click **Add** to add the pool

The screenshot shows the 'Services' tab in the Cisco Packet Tracer configuration window. On the left, a list of services includes HTTP, DHCP, DHCPv6, TFTP, DNS, SYSLOG, AAA, NTP, EMAIL, FTP, IoT, VM Management, and Radius EAP. The 'DHCP' service is selected. The main configuration area is titled 'DHCP' and shows the following settings:

- Interface: FastEthernet0
- Service: ☒ On
- Pool Name: DHCPpool
- Default Gateway: 208.67.220.220
- DNS Server: 208.67.220.220
- Start IP Address: 208.67.220.1
- Subnet Mask: 255.255.255.0
- Maximum Number of Users: 50
- TFTP Server: 0.0.0.0
- WLC Address: 0.0.0.0

At the bottom, there are 'Add', 'Save', and 'Remove' buttons. Below these buttons is a table showing the configured DHCP pool:

| Pool Name | Default Gateway | DNS Server     | Start IP Address | Subnet Mask   | Max User | TFTP Server | WLC Address |
|-----------|-----------------|----------------|------------------|---------------|----------|-------------|-------------|
| DHCPpool  | 208.67.220.220  | 208.67.220.220 | 208.67.220.1     | 255.255.255.0 | 50       | 0.0.0.0     | 0.0.0.0     |

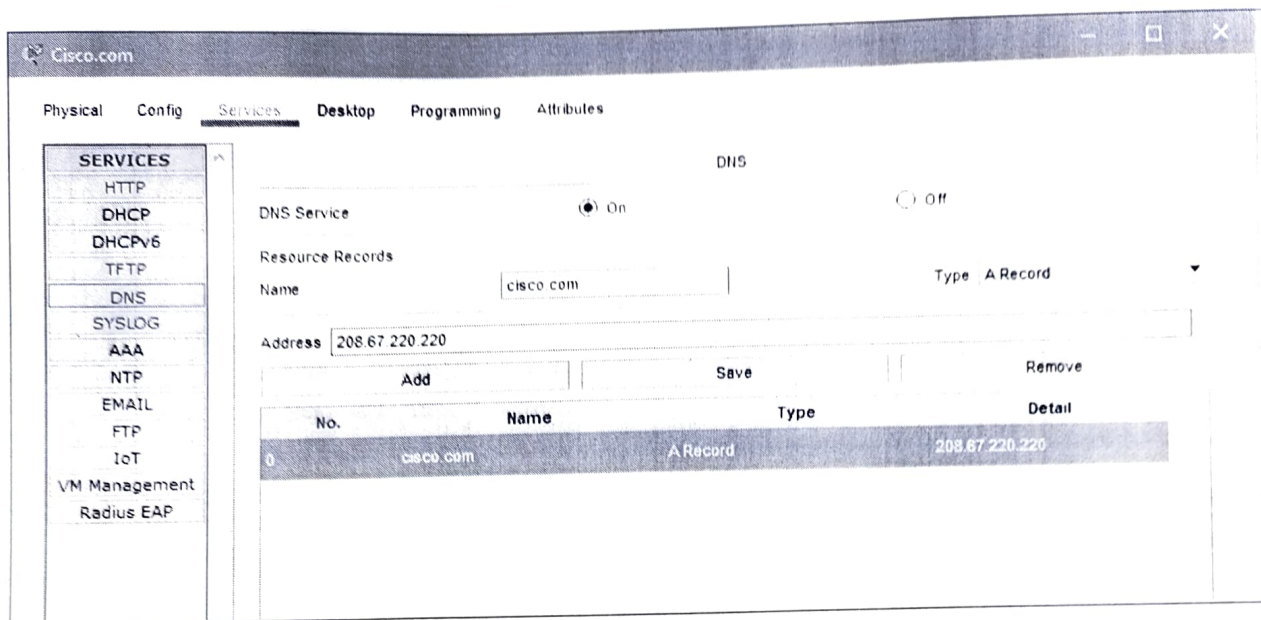
- b. Configure the Cisco.com server as a DNS server to provide domain name to IPv4 address resolution. While still in the **Services** tab, select **DNS** from the **SERVICES** listed in the left pane.

Configure the DNS service using the following settings as shown in the figure.

- Click **On** to turn the DNS service on
- Name: Cisco.com
- Type: A Record
- Address: 208.67.220.220

Click **Add** to add the DNS service settings





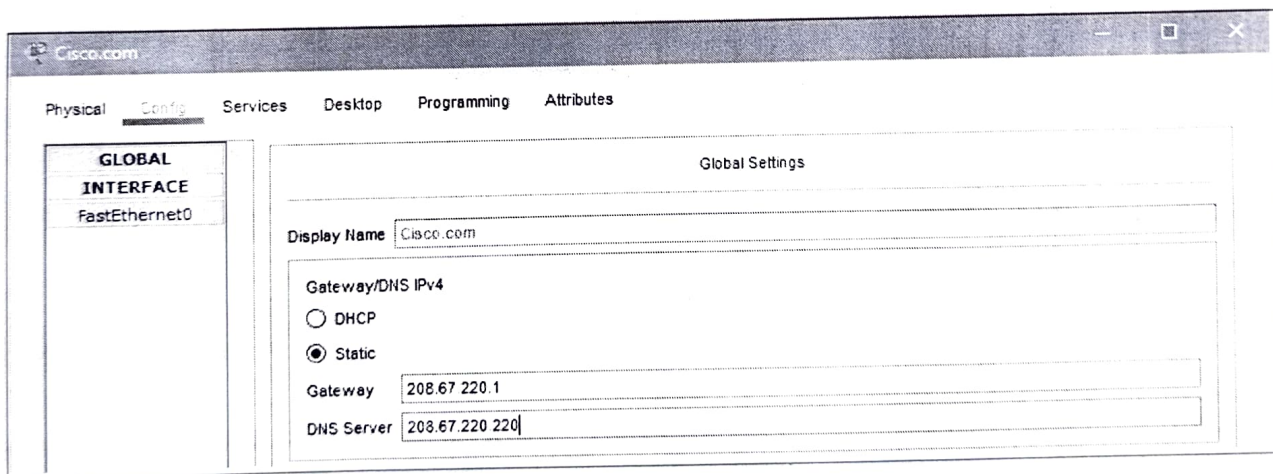
c. Configure the Cisco.com server Global settings.

Select the **Config** tab.

Click on **Settings** in left pane.

Configure the Global settings of the server as follows:

- Select **Static**
- Gateway: 208.67.220.1
- DNS Server: 208.67.220.220



d. Configure the Cisco.com server FastEthernet0 Interface settings.

Click on **FastEthernet** in left pane of the **Config** tab

Configure the FastEthernet Interface settings of the server as follows:



- Select **Static** under IP Configuration
- IP Address: 208.67.220.220
- Subnet Mask: 255.255.255.0

The screenshot shows the Cisco Packet Tracer configuration window for a FastEthernet0 interface. The window has a top navigation bar with tabs: Physical, Config (selected), Services, Desktop, Programming, and Attributes. On the left, there is a sidebar with 'GLOBAL' and 'INTERFACE' sections, where 'FastEthernet0' is selected under 'INTERFACE'. The main configuration area is titled 'FastEthernet0' and contains the following settings:

- Port Status:** ☒ On
- Bandwidth:** 100 Mbps, ☒ Auto
- Duplex:** Half Duplex, ☒ Auto
- MAC Address:** 000C.85CC.0116
- IP Configuration:**
  - ☐ DHCP
  - ☒ Static
- IP Address:** 208.67.220.220
- Subnet Mask:** 255.255.255.0
- IPv6 Configuration:**
  - ☐ DHCP
  - ☐ Auto Config
  - ☒ Static
- IPv6 Address:** (empty field)
- Link Local Address:** FE80::20C:85FF:FECC:116

At the bottom left of the window, there is a 'Top' button.

## Part 3: Verify Connectivity

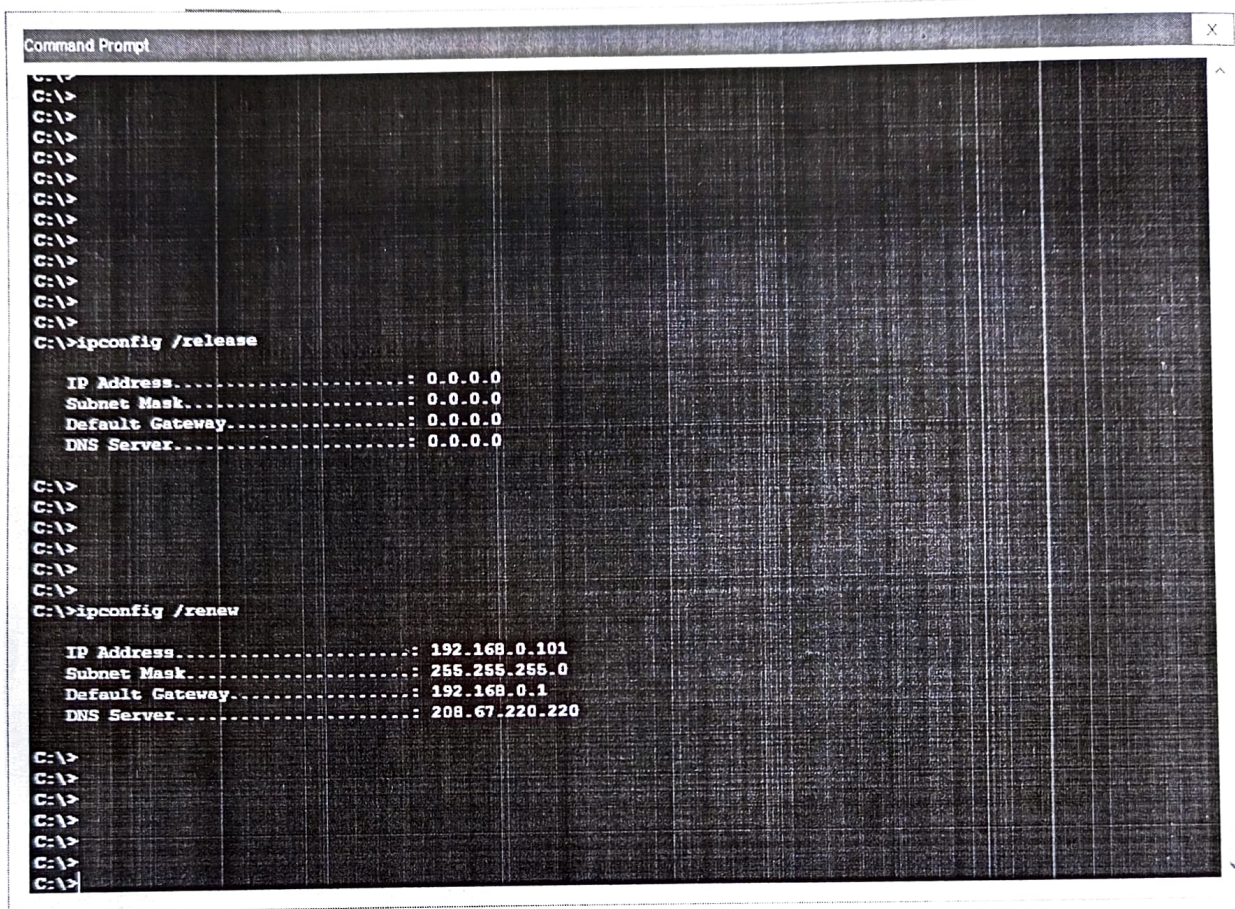
### Step 1: Refresh the IPv4 settings on the PC

- Verify that the PC is receiving IPv4 configuration information from DHCP.

Click on the **PC** on the Packet Tracer **Logical** workspace and then select the **Desktop** tab of the PC configuration window.

Click on the **Command Prompt** icon

In the command prompt refresh the IP settings by issuing the commands **ipconfig /release** and then **ipconfig /renew**. The output should show that the PC has an IP address in the 192.168.0.x range, a subnet mask, a default gateway, and DNS server address as shown in the figure.



```
Command Prompt
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>ipconfig /release

IP Address. . . . .: 0.0.0.0
Subnet Mask. . . . .: 0.0.0.0
Default Gateway. . . . .: 0.0.0.0
DNS Server. . . . .: 0.0.0.0

C:\>
C:\>
C:\>
C:\>
C:\>
C:\>ipconfig /renew

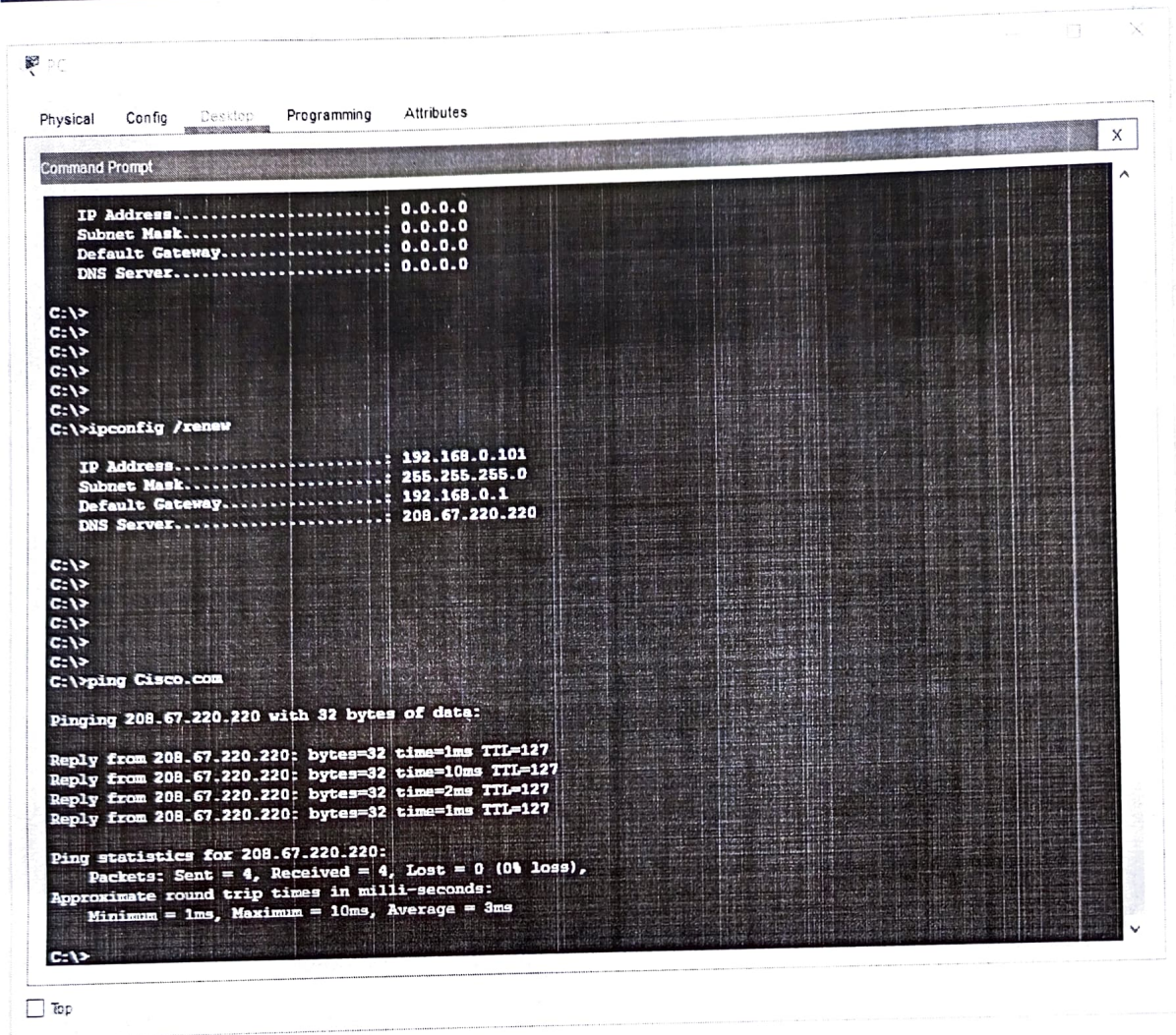
IP Address. . . . .: 192.168.0.101
Subnet Mask. . . . .: 255.255.255.0
Default Gateway. . . . .: 192.168.0.1
DNS Server. . . . .: 208.67.220.220

C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
```

- b) Test connectivity to the Cisco.com server from the PC

From the command prompt, issue the command **ping Cisco.com**. It may take a few seconds for the ping to return. Four replies should be received as shown in the figure.

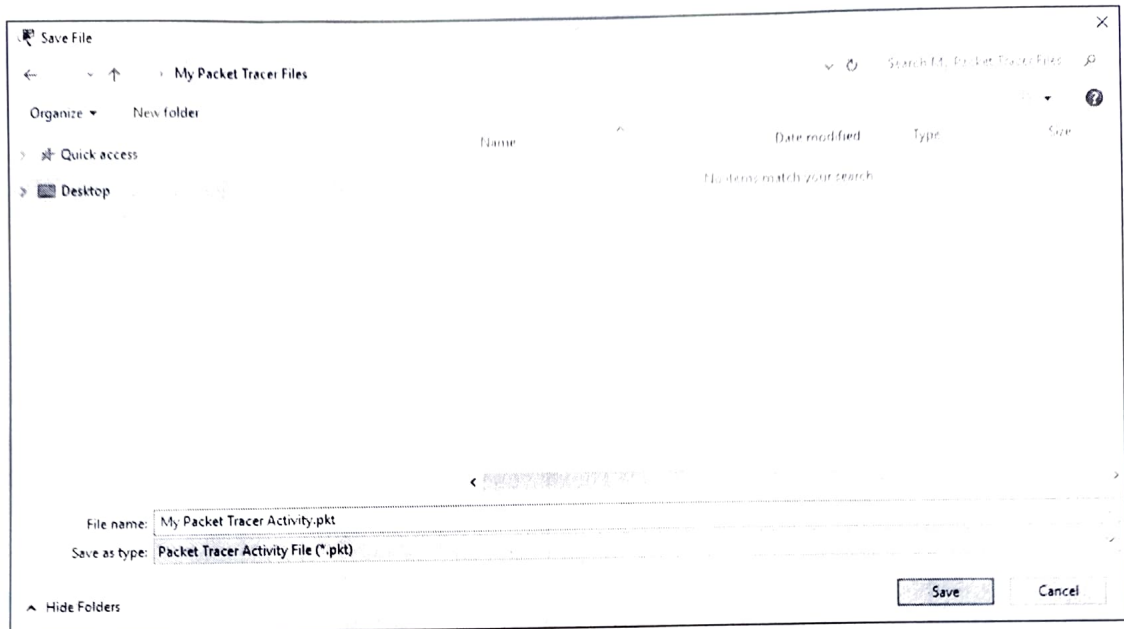




## Part 4: Save the File and Close Packet Tracer

### Step 1: Save the File as a Packet Tracer Activity File (\*.pkt)

To save the completed network, click on **File** in the Packet Tracer menu bar and then select **Save A** from the dropdown menu. In the **Save File** window choose a directory to save the file to and give the file an appropriate file name. The Save as type defaults to Packet Tracer Activity File (\*.pkt). Click **Save** to save the file.



### Step 2: Close Packet Tracer

To close Packet Tracer you can either click the "X" in the top right corner of the Packet Tracer window, or click on **Exit** in the file drop down menu.



**A.Y. 2025-26**

## **Report on simulation tool**

**Subject: Database & Management System**

**Sem: III**

### **1. Introduction**

A Database Management System (DBMS) is software that allows users to create, manage, and manipulate databases. SQLite is a lightweight, self-contained, and serverless SQL database engine. It is widely used in mobile devices, embedded systems, and applications requiring a compact database solution.

SQLite is often used as a simulation tool for learning SQL commands and practicing database concepts because it requires no installation of a full database server.

### **2. Objectives**

- To simulate database operations using SQLite.
- To practice SQL commands such as CREATE, INSERT, UPDATE, DELETE, and SELECT.
- To analyze query execution results.
- To understand the role of DBMS in data storage and retrieval.

### **3. Tools Used**

- Software: SQLite (sqlite3 command-line tool or DB Browser for SQLite)
- Platform: Windows/Linux
- Features Used:
  - SQL command execution
  - Query output window
  - Table creation and data manipulation





#### 4. Simulation Steps

1. Launch SQLite (via command-line or DB Browser).
2. Create a new database file (student.db).
3. Create a table using SQL command.
4. Insert records into the table.
5. Perform operations like SELECT, UPDATE, DELETE.
6. View results of executed queries.

#### 5. Example Program & Output

##### SQL Commands:

```
1 CREATE TABLE Student (  
2   RollNo INTEGER PRIMARY KEY,  
3   Name TEXT,  
4   Marks INTEGER  
5 );  
6  
7 INSERT INTO Student VALUES (1, 'Amit', 85);  
8 INSERT INTO Student VALUES (2, 'Pooja', 90);  
9 INSERT INTO Student VALUES (3, 'Ravi', 75);  
10  
11  
12 SELECT * FROM Student;
```

| RollNo | Name  | Marks |
|--------|-------|-------|
| 1      | Amit  | 85    |
| 2      | Pooja | 90    |
| 3      | Ravi  | 75    |





The screenshot shows the SQLite IDE interface. At the top, there are buttons for 'Import' and 'Export'. Below them are 'Run' and 'SQLite' buttons. The main area contains a list of SQL commands:

```
1 -- Update Record
2 UPDATE Student SET Marks = 95 WHERE RollNo = 2;
3
4 -- Delete Record
5 DELETE FROM Student WHERE RollNo = 3;
6
7 SELECT * FROM Student;
```

Below the commands, a table view is displayed with the following data:

| RollNo | Name  | Marks |
|--------|-------|-------|
| 1      | Amit  | 85    |
| 2      | Pooja | 95    |

## 6. Results & Analysis

- A database student.db was successfully created.
- SQL commands were executed correctly.
- Results matched expectations (e.g., record updated, record deleted).
- SQLite proved to be an effective tool for simulating DBMS operations.

## 7. Conclusion

Simulation of DBMS using SQLite provided hands-on experience with SQL commands. It demonstrated how data can be stored, retrieved, and manipulated efficiently. Being lightweight and portable, SQLite is a powerful tool for learning and practicing database concepts.



**St. John College of Engineering and Management**

**Autonomous Institute**

**(A Christian Religious Minority Institution)**

Approved by AICTE and DTE, Affiliated to University of Mumbai / MSBTE

DTE Code : 3218 AICTE Permanent ID : 1-4790201

NAAC Accredited with Grade 'A+', Three Programs NBA Accredited



**A.Y. 2025-26**

## **Report on simulation tool**

**Subject: Microprocessor & Computer Organization**

**Sem: III**

### **1. Introduction**

The Intel 8086 microprocessor is a 16-bit processor that serves as the basis of the x86 architecture. To understand and practice assembly programming, the 8086 Emulator (8086emu) is widely used. It provides an environment where users can write, assemble, and execute programs without the need for actual hardware.

This report presents the simulation of sample programs using the 8086emu.

### **2. Objectives**

- To understand the working of 8086 microprocessor.
- To simulate assembly language programs.
- To analyze the results using register values and memory status.

### **3. Tools Used**

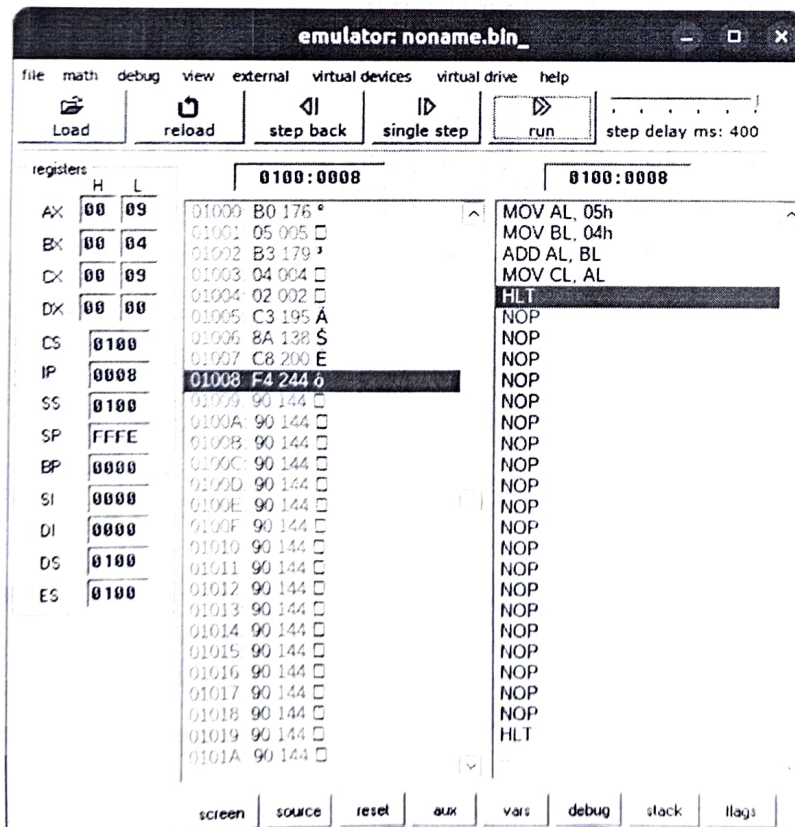
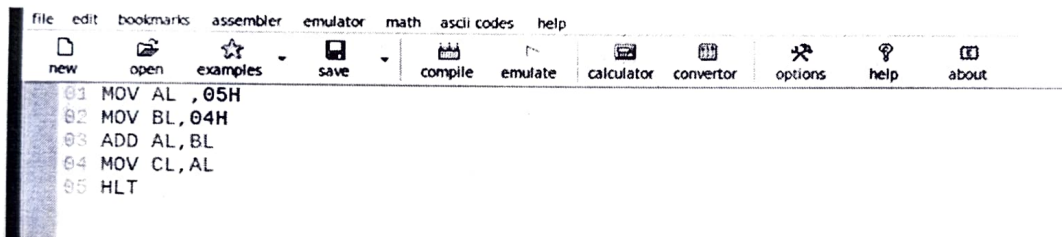
- Software: 8086 Microprocessor Emulator (8086emu)
- Platform: Windows/Linux
- Features Used:
  - Code Editor
  - Assembler
  - Registers & Flags Window
  - Memory Dump



#### 4. Simulation Steps

1. Open 8086emu.
2. Create a new program and enter assembly code.
3. Assemble the program to check for errors.
4. Run the program step-by-step or continuously.
5. Observe the changes in registers, flags, and memory.

### 5. Example (Screenshot)







**St. John College of Engineering and Management**

**Autonomous Institute**

**(A Christian Religious Minority Institution)**

Approved by AICTE and DTE, Affiliated to University of Mumbai / MSBTE

DTE Code : 3218 AICTE Permanent ID : 1-4790201

NAAC Accredited with Grade 'A+', Three Programs NBA Accredited



## **6. Results & Analysis**

- The program executed successfully.
- Registers showed the expected results.
- Flags were updated according to the operation.
- The emulator proved useful in debugging and understanding 8086 assembly programs.

## **7. Conclusion**

The simulation using **8086emu** helped in understanding the basic operations of the 8086 microprocessor. It provides a simple and effective way to execute and debug assembly programs, making it an essential tool for students and beginners in microprocessor programming.