# SAVEETHA SCHOOL OF ENGINEERING

**CAPSTONE PROJECT**

Design and Implementation of a Secured VPN Connection

**NAME:** Harigowtham. A

**REGISTER NUMBER:** 192324198 **COURSE CODE:** CSA0747

**COURSE NAME:** Computer Network for IoT

## INTRODUCTION:

### In today's interconnected world, the need for secure communication channels has become more critical than ever. Virtual Private Networks (VPNs) offer a robust solution by encrypting data transmitted over the internet, ensuring privacy and security. This assignment focuses on the Design and Implementation of a Secured VPN Connection, where the objective is to create a comprehensive VPN solution that can safeguard user data, whether on desktop or mobile platforms. Through this project, students will explore the fundamentals of VPN technologies, encryption protocols, and secure communication methods, culminating in the development and deployment of a real-world VPN solution that prioritizes security, scalability, and usability.

### Objective:

* Design a network architecture that includes a secure VPN server.
* Configure the VPN server to provide encrypted communication services to its clients.
* Implement secure access for remote users, ensuring privacy and data integrity.
* Showcase the VPN solution by demonstrating secure web services and remote access functionality.
* Analyze the pros and cons of the VPN setup, focusing on security, scalability, and usability.
* Test the VPN solution in real-world scenarios, assessing its performance under various conditions.

## LITERATURE REVIEW

In this project, Cisco Packet Tracer is employed to simulate a Virtual Private Network (VPN), a crucial tool for securing communication over public networks by encrypting data exchanged between clients and servers. The focus of the project is on configuring and managing secure connections using various protocols, including IPsec (Internet Protocol Security) and SSL (Secure Sockets Layer).IPsec provides a suite of protocols designed to secure IP communications through encryption and authentication, while SSL/TLS (Transport Layer Security) secures the data transmitted over web-based applications. Cisco Packet Tracer enables the practical experimentation of these protocols within a controlled network environment. The project involves setting up and configuring routers, switches, and servers within Packet Tracer to create VPN tunnels. This hands-on approach allows for the exploration of network topologies and the implementation of security measures. Participants will gain a deeper understanding of encryption methods, authentication techniques, and the intricacies of maintaining both security and performance in VPN deployments. Through this simulation, users will encounter real-world challenges and scenarios, such as optimizing VPN performance, managing network traffic, and troubleshooting potential issues. This experiential learning process enhances comprehension of how VPNs function, their role in safeguarding data, and their impact on network operations.

# METHODOLOGY

**Software:**

* Cisco Packet Tracer

### Network Design:

Network consist of

* + 4 Routers
  + 2 Switches
  + 4 PC

**Methodology:**

**1. Network Design Overview**

* **PCs (PC0, PC1, PC2, PC3)** are connected to **Switches (2960-24TT)**, which are then connected to **Routers (2811, 2111)**.
* **Routers** are connected in a chain to allow inter-network communication across different subnets.
* The **goal** seems to ensure that PCs from different subnets (10.0.0.0, 20.0.0.0, 60.0.0.0, 70.0.0.0, 80.0.0.0) can communicate.

**2. Assign IP Addresses**

* Assign static IP addresses to the PCs.
  + PC0: 10.0.0.1
  + PC1: 10.0.0.2
  + PC2: 20.0.0.1
  + PC3: 20.0.0.2
* Assign IP addresses to the router interfaces connected to the respective networks:
  + Router0:
    - Interface connecting to Switch0: 10.0.0.3
    - Interface connecting to Router1: 60.0.0.1
  + Router1:
    - Interface connecting to Router0: 60.0.0.2
    - Interface connecting to Router2: 70.0.0.1
  + Router2:
    - Interface connecting to Router1: 70.0.0.2
    - Interface connecting to Router3: 80.0.0.1
  + Router3:
    - Interface connecting to Router2: 80.0.0.2
    - Interface connecting to Switch1: 20.0.0.3

**3. Configure Routing**

* Since the routers are on different networks, you need to configure **static routes** or enable a **dynamic routing protocol** (e.g., RIP, OSPF).
* For static routing:
  + Configure static routes on each router to know how to reach other networks.

Example of static route configuration on Router0:

arduino

Copy code

Router0(config)# ip route 20.0.0.0 255.255.255.0 60.0.0.2 Router0(config)# ip route 70.0.0.0 255.255.255.0 60.0.0.2 Router0(config)# ip route 80.0.0.0 255.255.255.0 60.0.0.2

**4. Routing Protocol Configuration (Optional)**

* If you choose dynamic routing like **RIP**, configure it across the routers:

arduino

Copy code

Router0(config)# router rip Router0(config-router)# version 2 Router0(config-router)# network 10.0.0.0 Router0(config-router)# network 60.0.0.0

**5. Test Connectivity**

* Use the **ping** command to test connectivity between PCs on different subnets.
* For instance, from PC0, ping PC2 to ensure that traffic routes correctly across the routers.
* Example command from PC0:

Copy code

ping 20.0.0.1

**6. Simulation and Debugging**

* In **simulation mode**, observe the flow of ICMP packets between devices to ensure routing is properly set up.
* If the ping fails:
  + Verify IP addresses and subnet masks.
  + Check that routing tables are correct.
  + Ensure there are no ACLs blocking the traffic.

**7. Final Adjustments**

* Configure any **additional services** like DHCP, DNS, or NAT if required.
* Test the network for stability by continuously pinging between multiple PCs.

**8. Document the Configuration**

* Keep a log of all configuration commands for each router.
* Ensure the configuration files are saved properly.

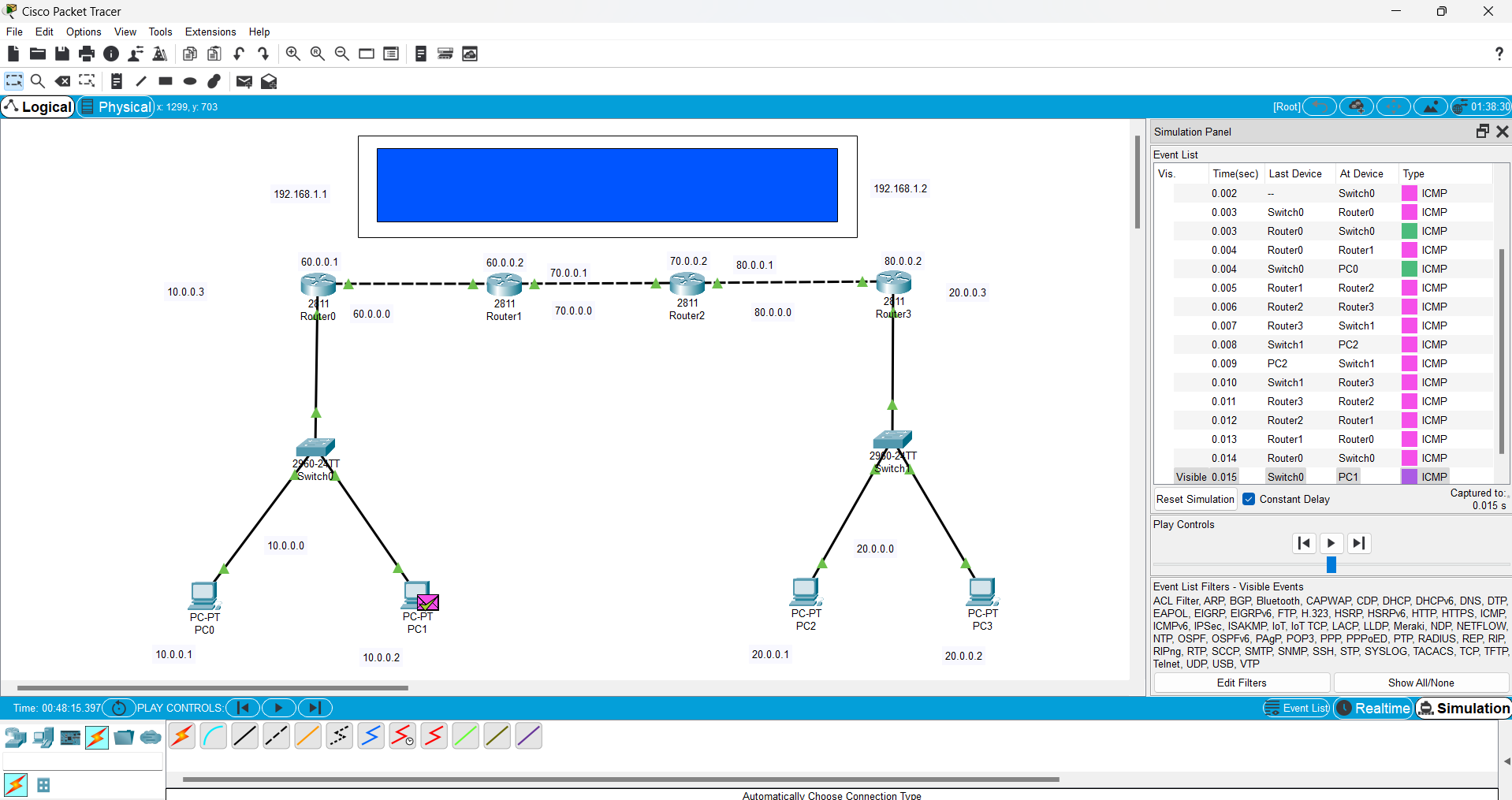
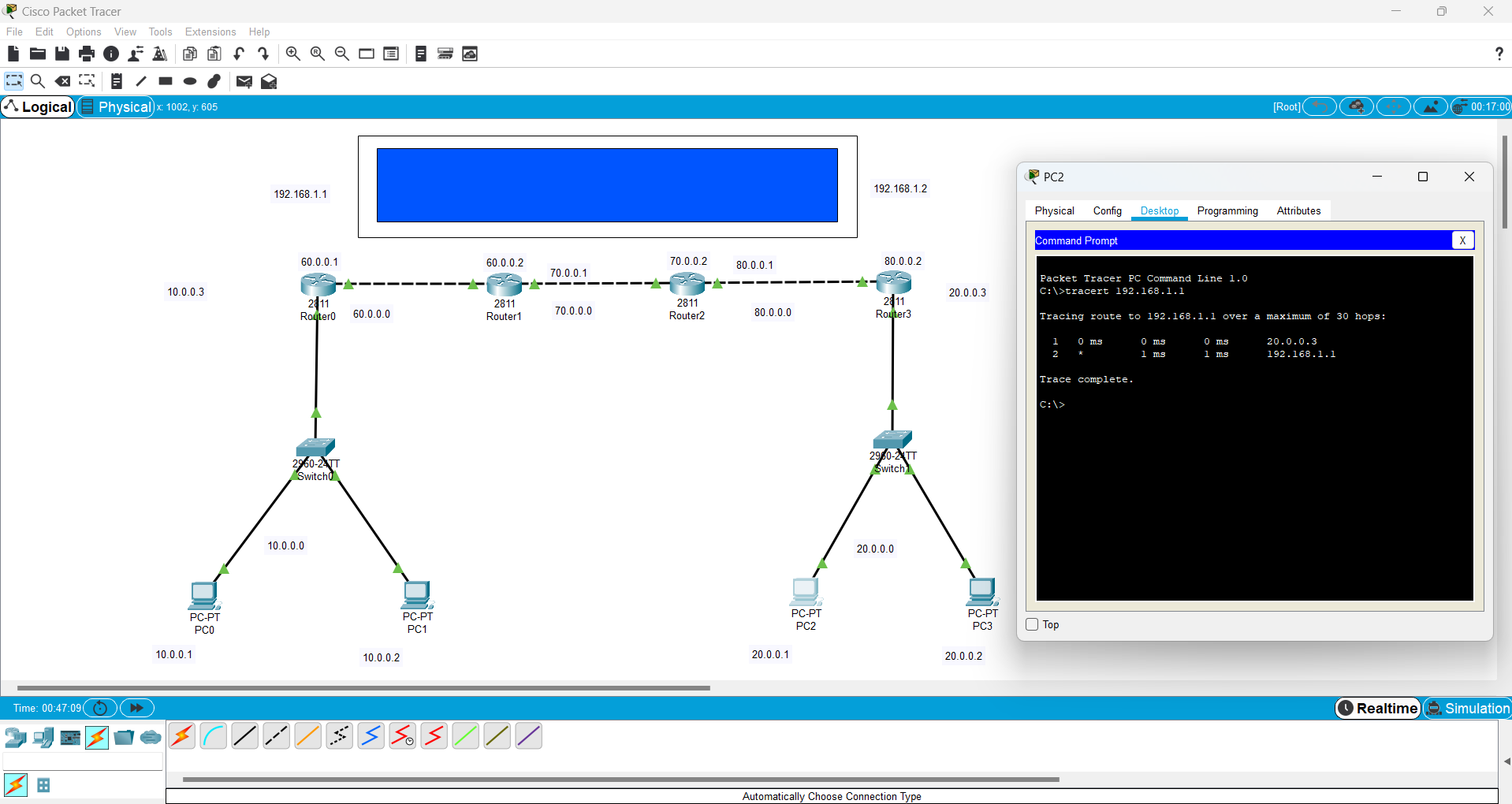
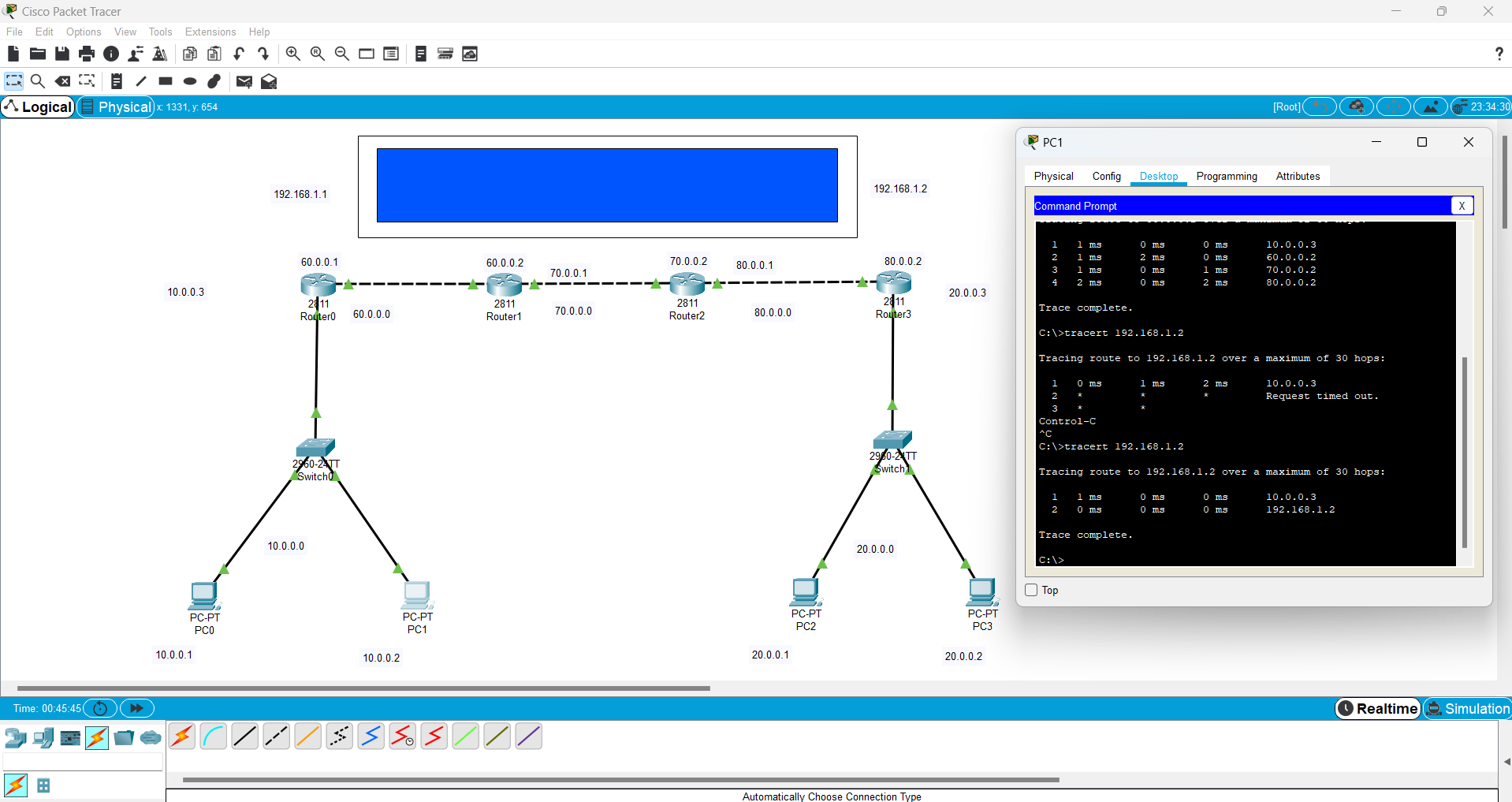
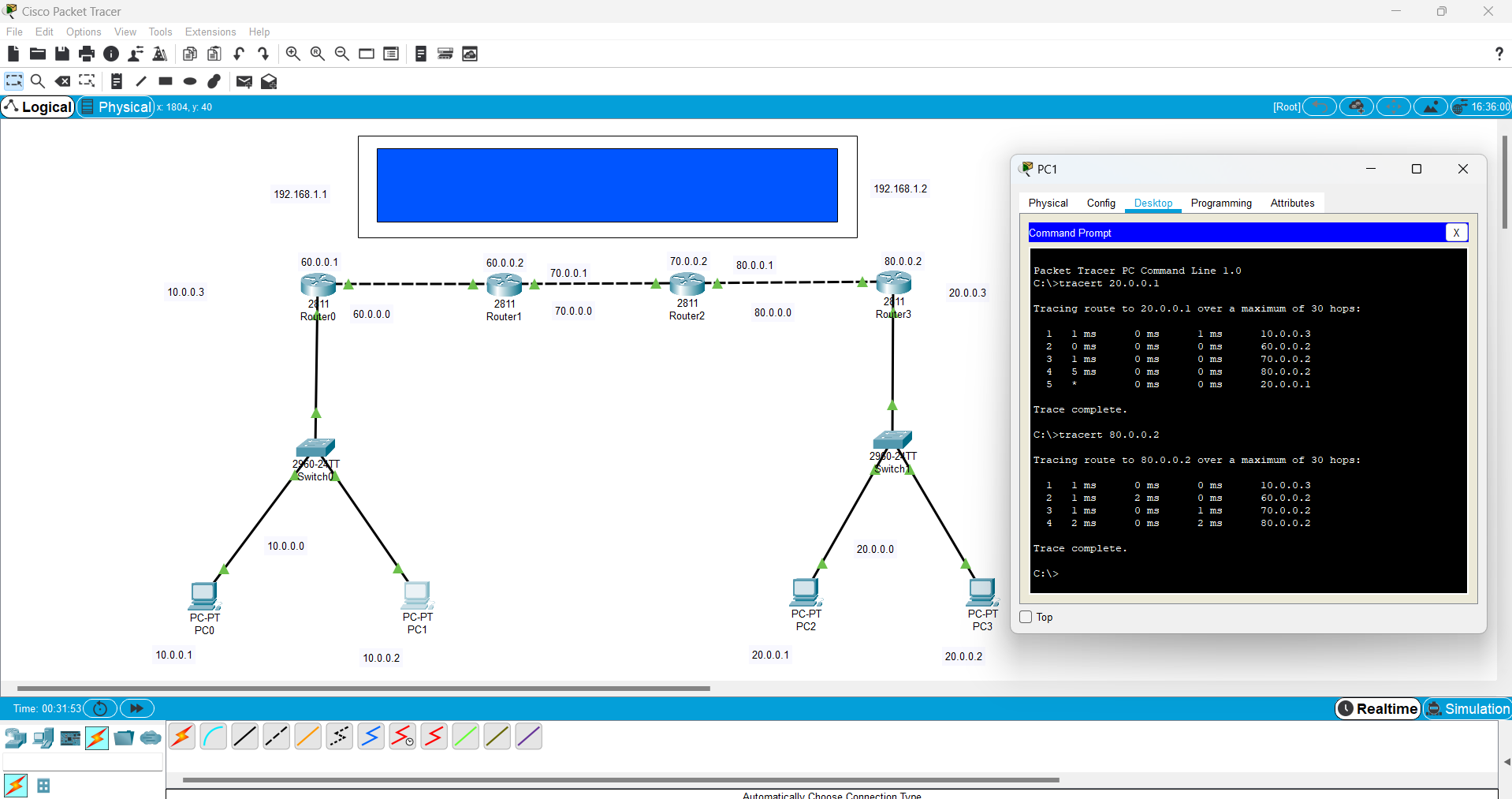
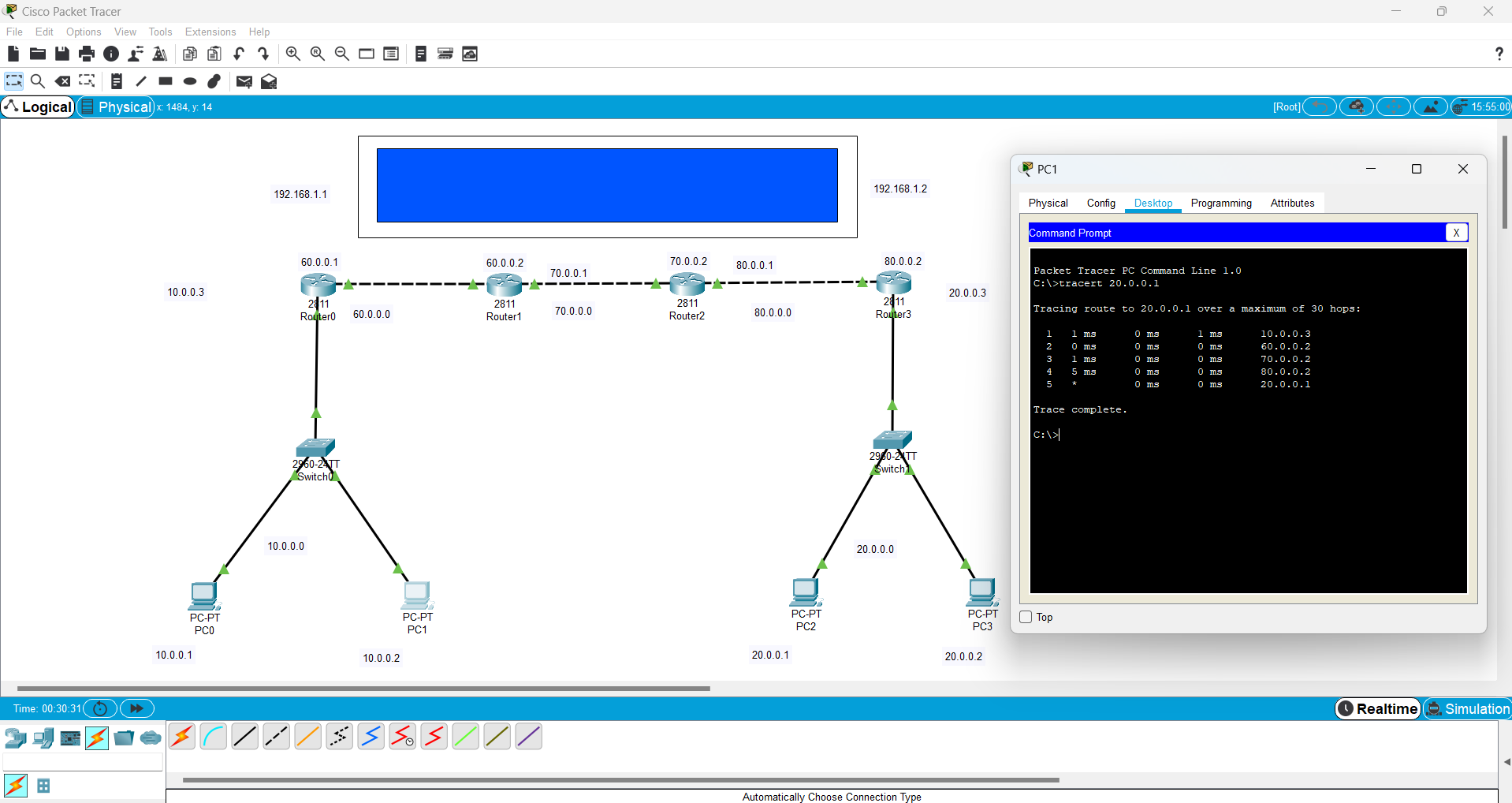
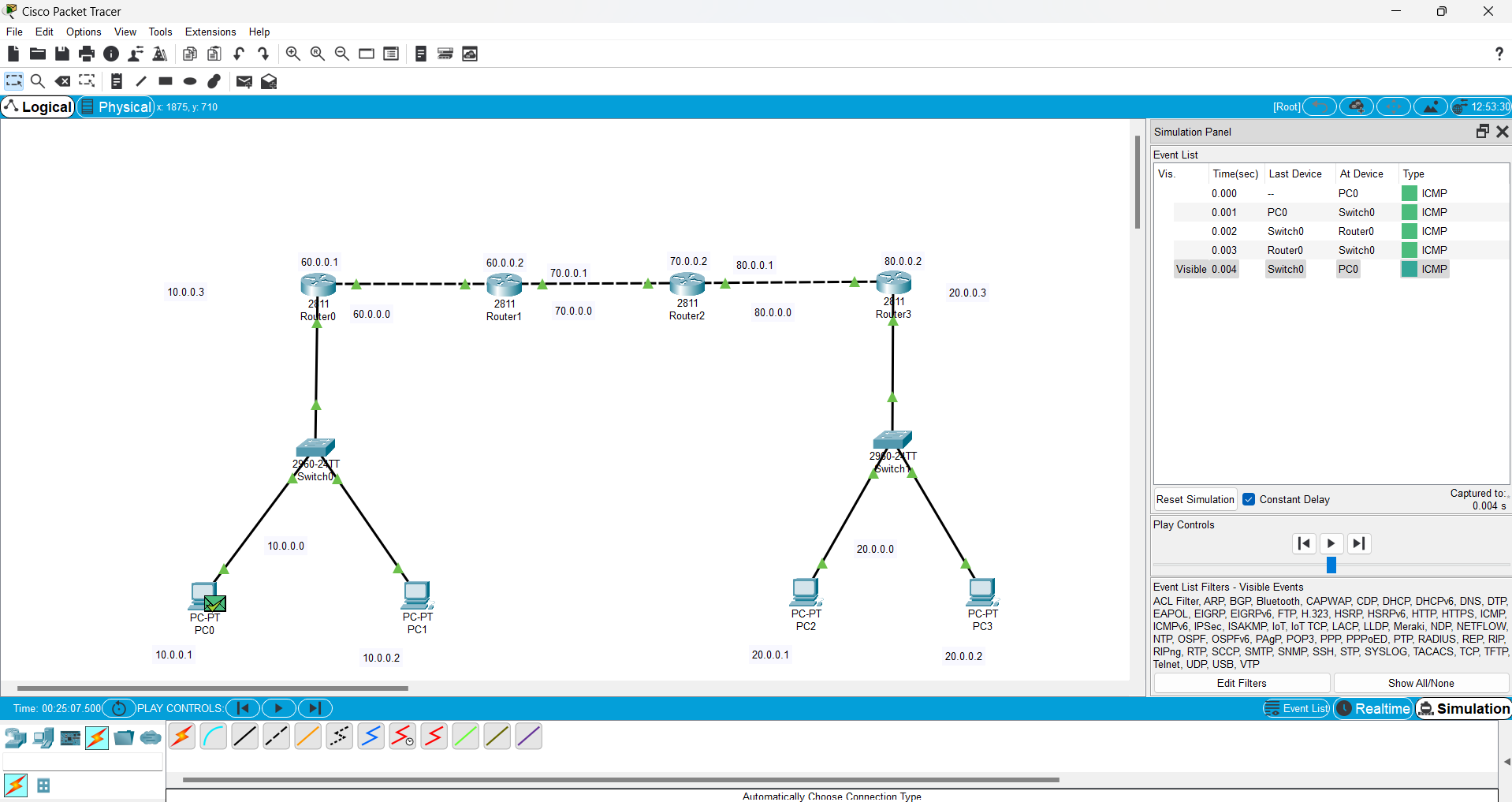
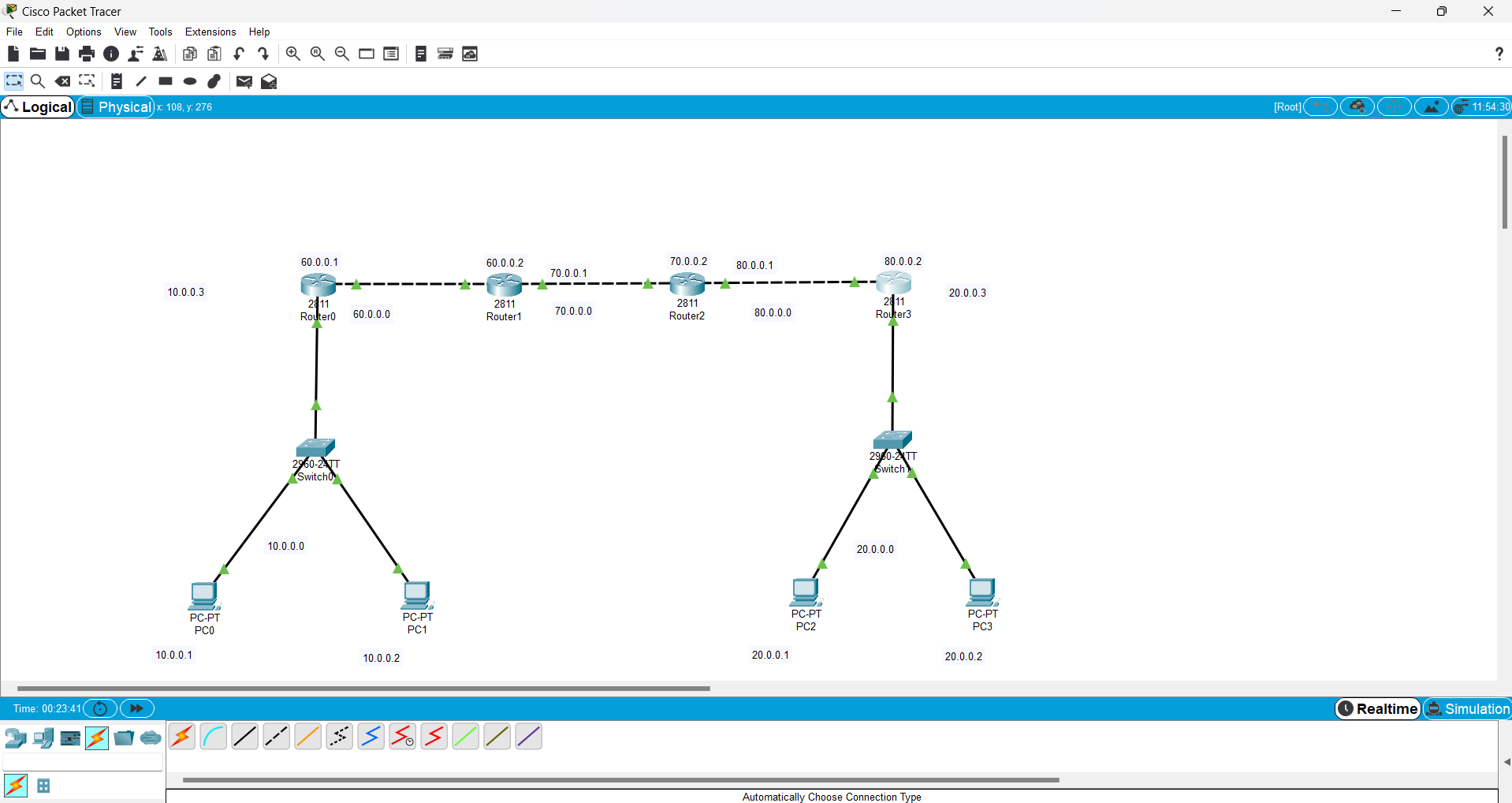
**Protocol:-ICMP**

**ICMP (Internet Control Message Protocol)**

* **Purpose:** Used for diagnostics and error reporting in network communication.
* **Usage in this project:** ICMP is used to send **ping** messages between PCs to verify connectivity across the routers and subnets.

**RESULTS:**

**Network Design:**

****

**CONCLUSION:**