Practical - 1

Aim: To experiment on Simulation Tools: (CISCO PACKET TRACER).

About CISCO Packet Tracer:

What is CISCO Packet Tracer?

Cisco Packet Tracer is a powerful network simulation tool developed by Cisco Systems. It allows users to design, configure, and troubleshoot virtual networks, making it ideal for learning networking concepts. The software supports a wide range of devices, including routers, switches, PCs, and IoT devices, enabling users to simulate real-world networking scenarios. Packet Tracer is widely used in educational environments for CCNA and other networking certifications, as it helps students practice skills without requiring physical hardware. Its user-friendly interface and drag-and-drop functionality make it accessible to beginners. Additionally, it supports multi-user collaboration, allowing users to work on the same network simulation in real-time. Overall, Cisco Packet Tracer is an essential tool for building practical networking knowledge and skills.

Features of CISCO Packet Tracer:

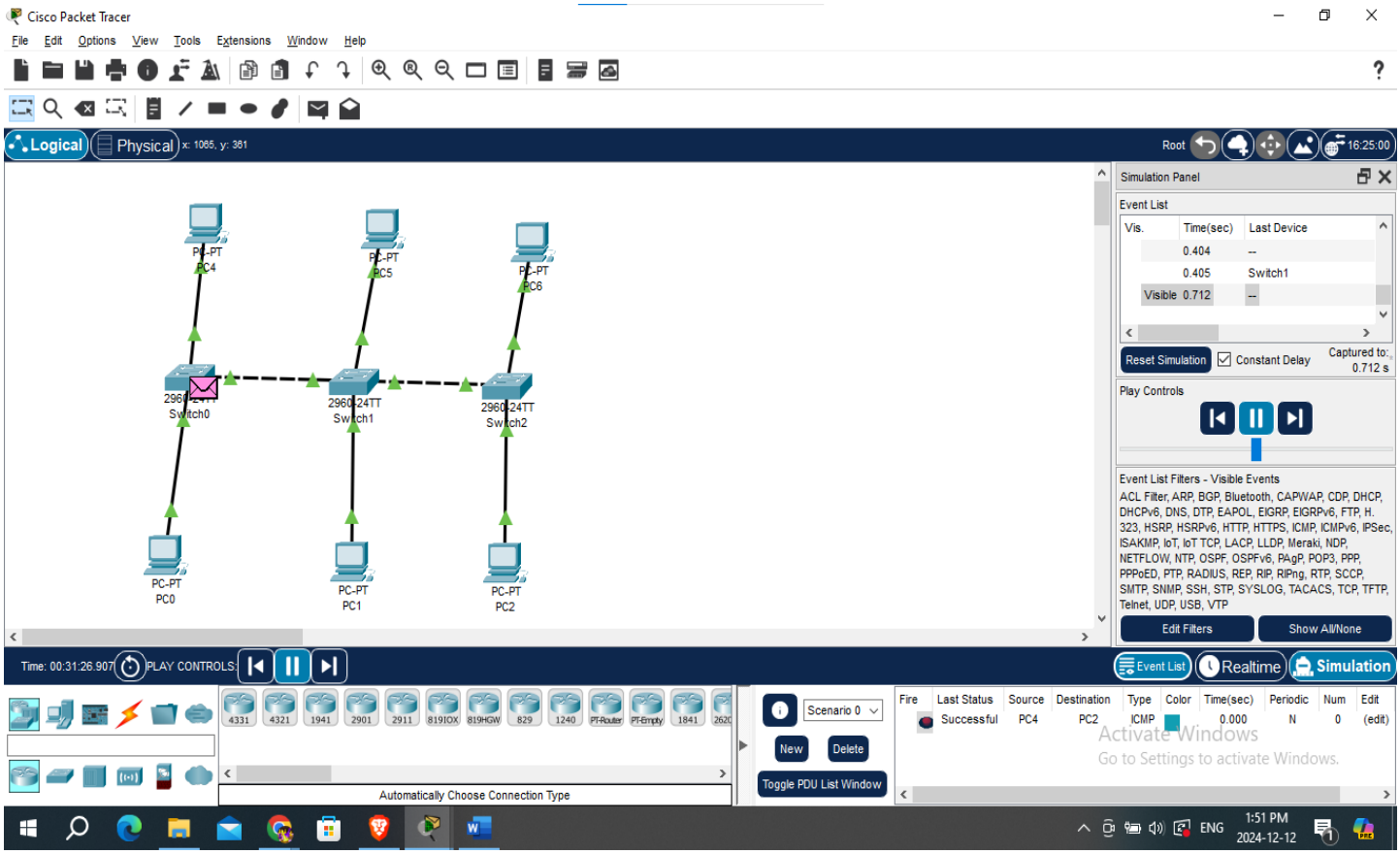
- **Network Topology Creation**: Supports designing complex networks with devices like routers, switches, PCs, and IoT components.
- **Realistic Simulation**: Provides a virtual environment to practice network configuration, troubleshooting, and protocol implementation.
- **User-Friendly Interface**: Features a drag-and-drop interface, making it easy to use for beginners.
- **Real-Time and Simulation Modes**: Allows users to observe network behavior and analyze packet flow.
- **Multi-User Collaboration**: Enables teams to work together on the same network project simultaneously.
- **Integration with Cisco Courses**: Works seamlessly with Cisco Networking Academy for hands-on learning and certifications like CCNA.
- **Lightweight Design**: Runs efficiently on various systems without heavy resource requirements.
- **Diverse Device Support**: Includes support for a variety of devices, from traditional network components to IoT devices.

Enrollment No. : 2403031307008
Div : 4 CC1

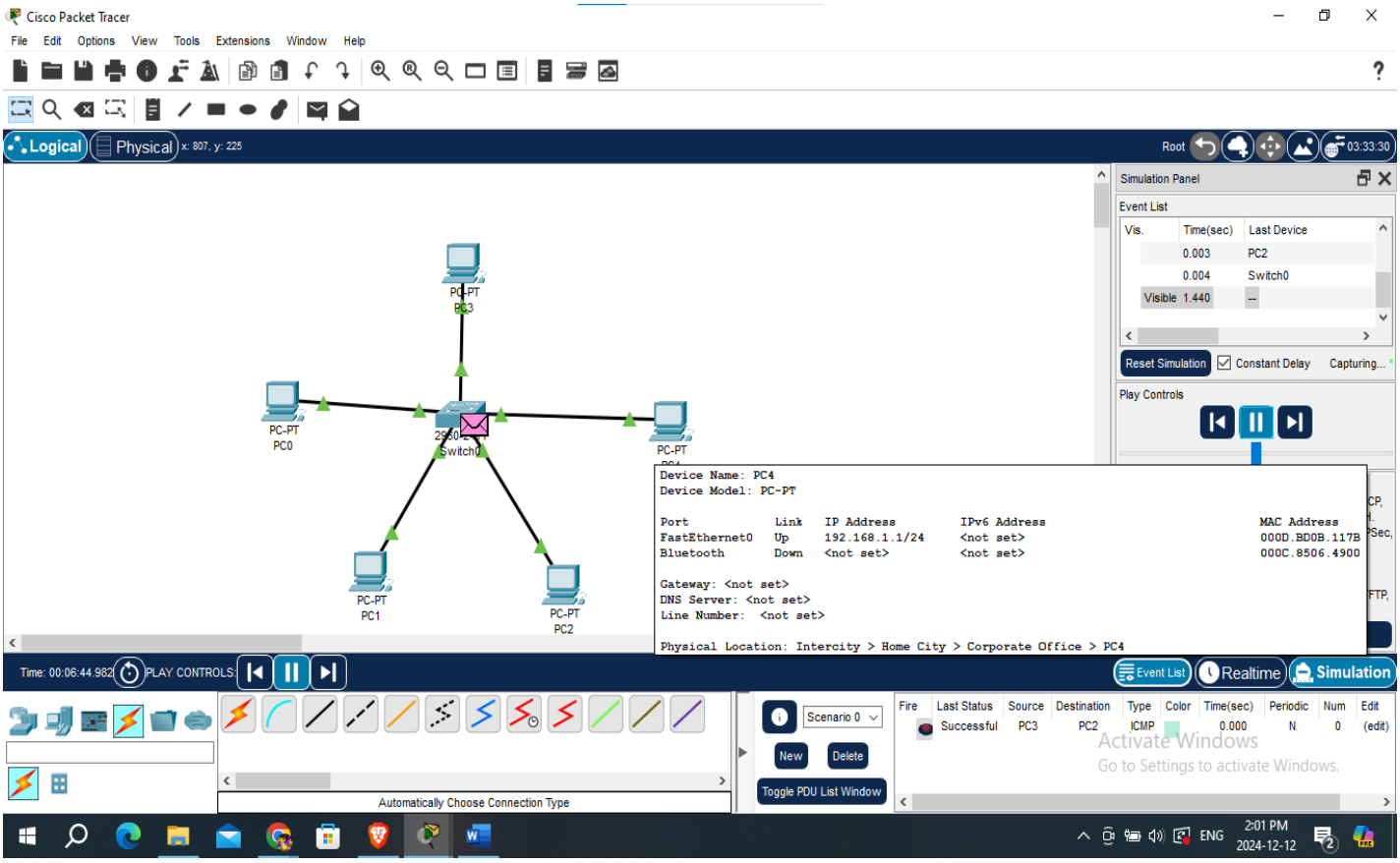
What is topology?

A **topology** refers to the layout or arrangement of devices and connections in a network. It defines how computers, switches, routers, and other devices are physically or logically connected to communicate with one another. Topologies can be **physical** (actual hardware connections) or **logical** (how data flows within the network). Common types include **bus, star, ring, mesh**, and **hybrid** topologies, each with unique advantages and use cases. The choice of topology impacts network performance, scalability, and reliability.

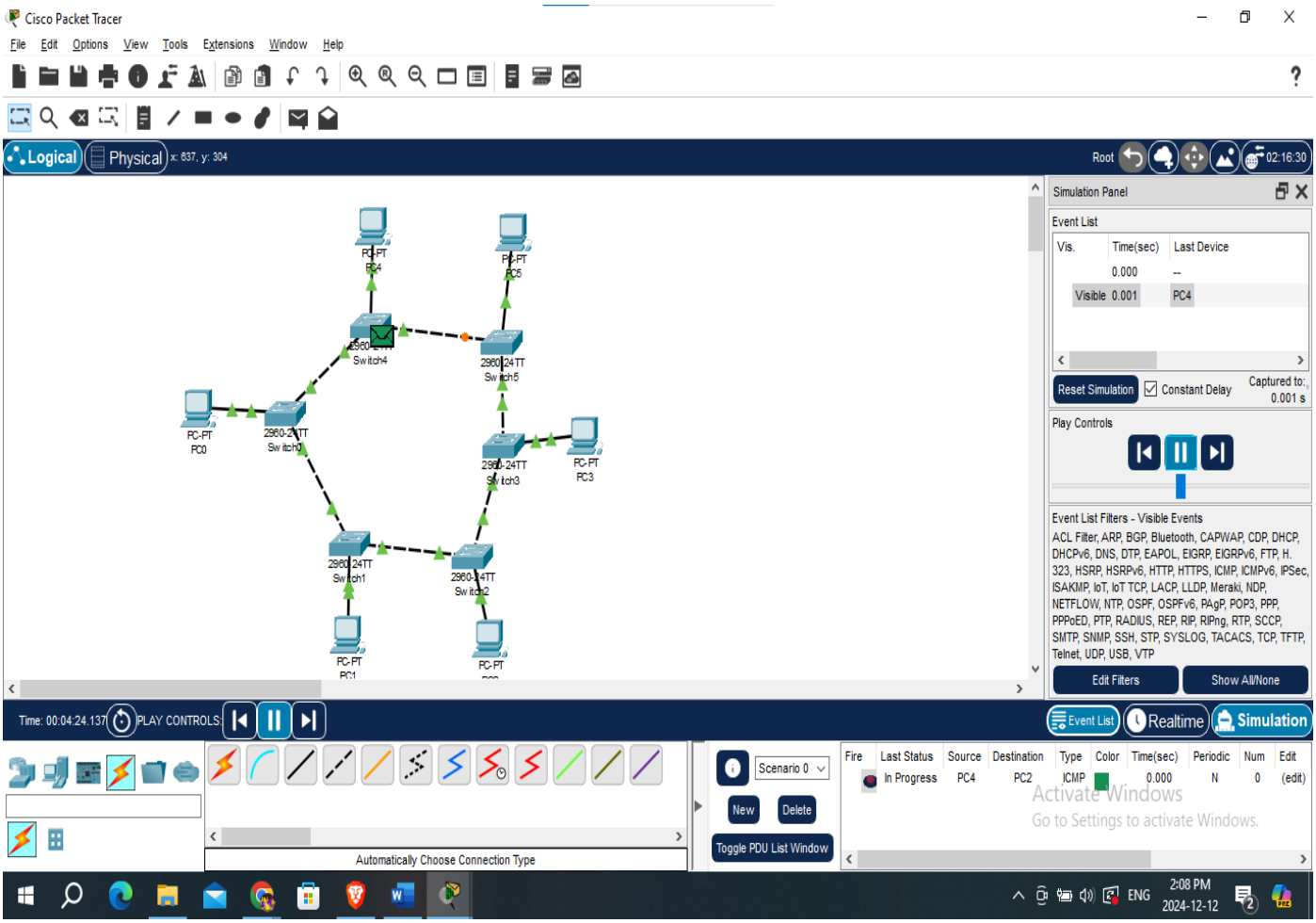
1. **Bus Topology**:
    - All devices are connected to a single central cable (bus) with terminators at both ends.
    - Data travels in both directions along the bus, and all devices share the same communication line.
    - It is simple and cost-effective but can experience data collisions and network failure if the main cable is damaged.



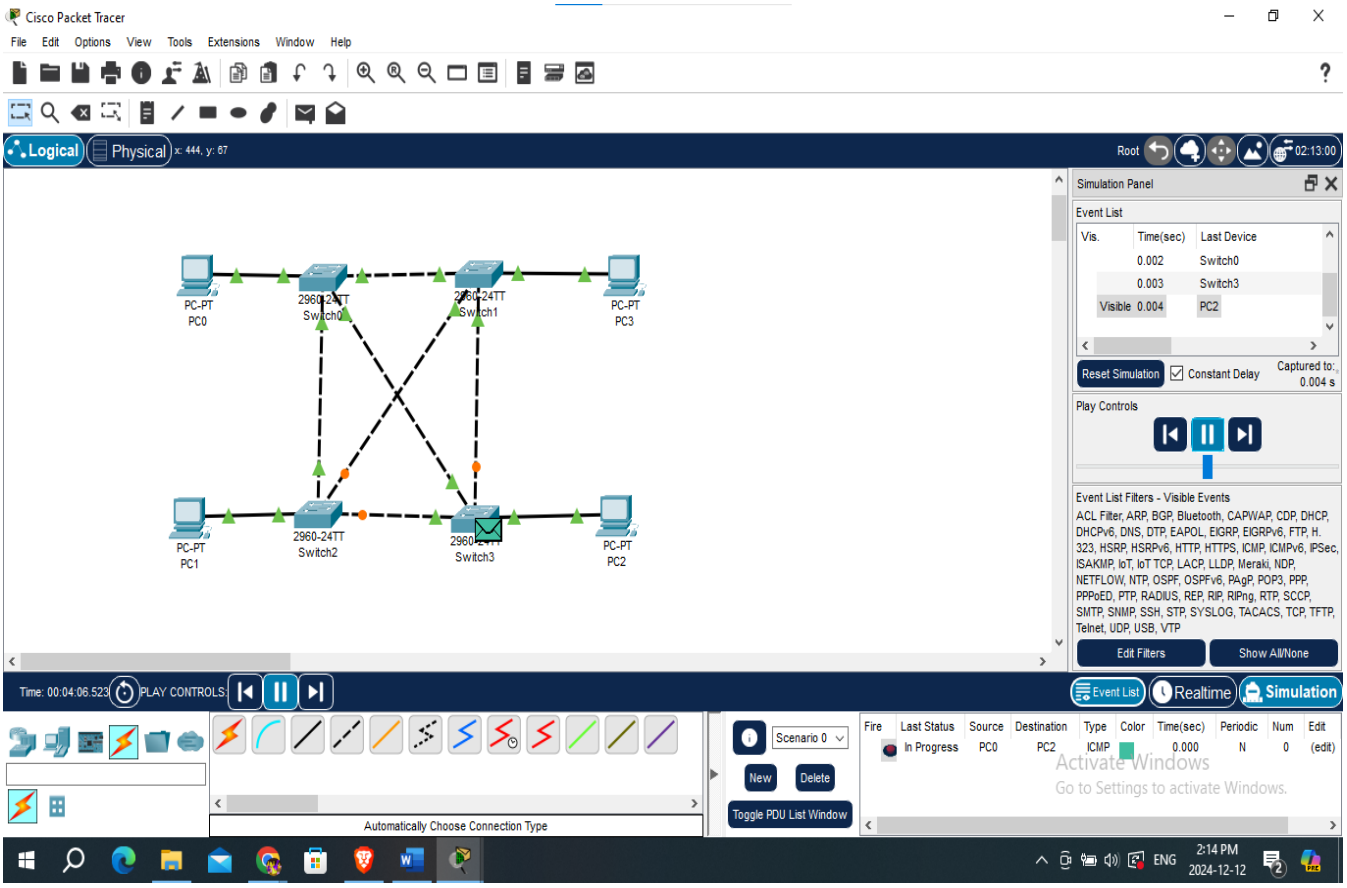
2. **Star Topology**:
    - All devices are connected to a central hub or switch, acting as a point of communication.
    - It is easy to install, manage, and troubleshoot, as a single device failure doesn't affect others.
    - However, the central hub is a single point of failure, and its failure can bring down the entire network.



3. **Ring Topology**:
   - Devices are connected in a circular loop, where data travels in one direction (unidirectional) or both directions (bidirectional).
   - It reduces the risk of collisions, but a single device failure can disrupt the entire network unless redundancy is in place.
   - Common in older networks and token-based communication systems.

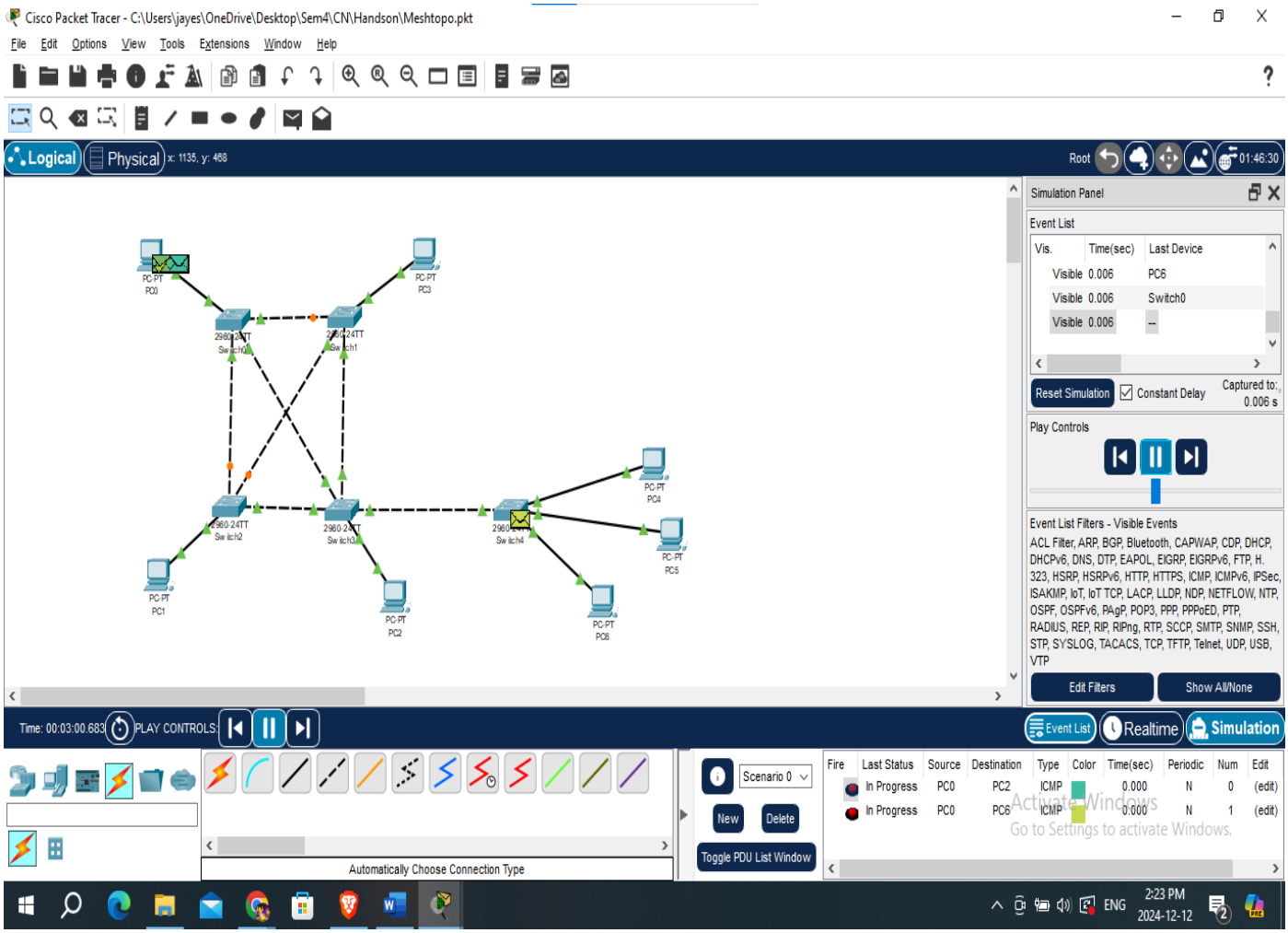


4. **Mesh Topology**:
    - Every device connects directly to every other device, creating multiple paths for data transmission.
    - It offers high reliability and fault tolerance, as data can take alternative paths if a connection fails.
    - However, it is expensive and complex to implement due to the large number of connections.



5. **Hybrid Topology**:
    - Combines two or more types of topologies (e.g., star-bus or star-ring) to meet specific network requirements.
    - It is highly flexible, scalable, and reliable, making it suitable for large and complex networks.

- However, it can be expensive and challenging to design and manage.



Conclusion:

In this practical, we explored the capabilities of **Cisco Packet Tracer** as a powerful network simulation tool. We learned how to create different network topologies, configure various networking devices, and simulate network behavior. Packet Tracer's user-friendly interface and wide range of supported devices allowed us to design and test networks without the need for physical hardware. This practical demonstrated how effective Cisco Packet Tracer is for learning networking concepts, troubleshooting, and practicing real-world network configurations. Overall, Packet Tracer proved to be an invaluable tool for gaining hands-on experience in networking and preparing for certifications like **CCNA**.

Enrollment No. : 2403031307008
Div : 4 CC1

Practical – 2

Aim : To experiment on Packet capture tool: Wireshark.

About Wireshark:

What is Wireshark?

Wireshark is a free and open-source network packet analyzer used to monitor and analyze network traffic in real time. It captures data packets traveling across a network and displays them in a detailed and readable format, helping users troubleshoot network issues, monitor performance, and detect security vulnerabilities. Wireshark supports multiple protocols, making it useful for analyzing different types of networks, including Ethernet, Wi-Fi, and TCP/IP. It provides filtering, visualization, and export features for deeper packet inspection. Widely used by network administrators, security experts, and developers, Wireshark is a powerful tool for understanding network behavior.

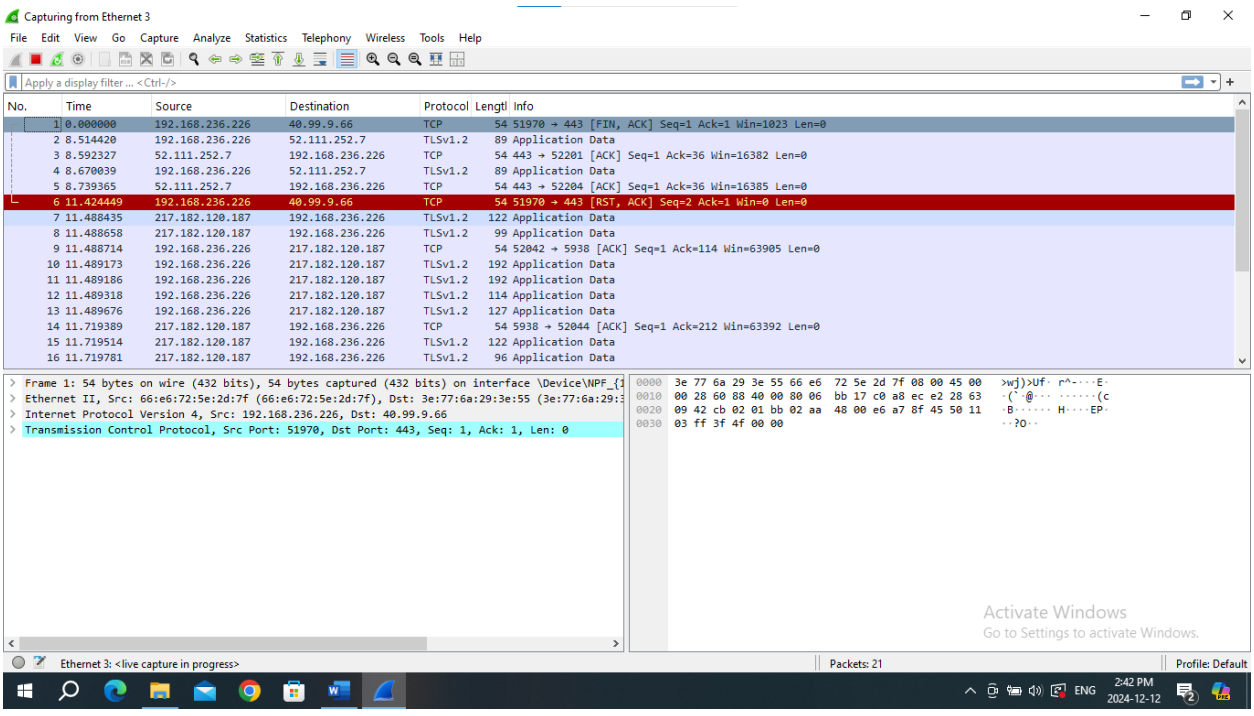
Uses of Wireshark:

1. **Network Troubleshooting**: Helps identify and resolve network issues by analyzing packet-level details.
2. **Performance Monitoring**: Monitors network performance, bandwidth usage, and traffic patterns.
3. **Protocol Analysis**: Examines how different protocols (e.g., TCP, UDP, HTTP) operate and interact on the network.
4. **Security Analysis**: Detects suspicious activities, potential attacks, or vulnerabilities in network traffic.
5. **Packet Filtering**: Filters specific packets based on criteria like IP address, port, or protocol for focused analysis.
6. **Learning and Development**: Used for educational purposes to understand networking concepts and protocols.

Enrollment No. : 2403031307008
Div : 4 CC1

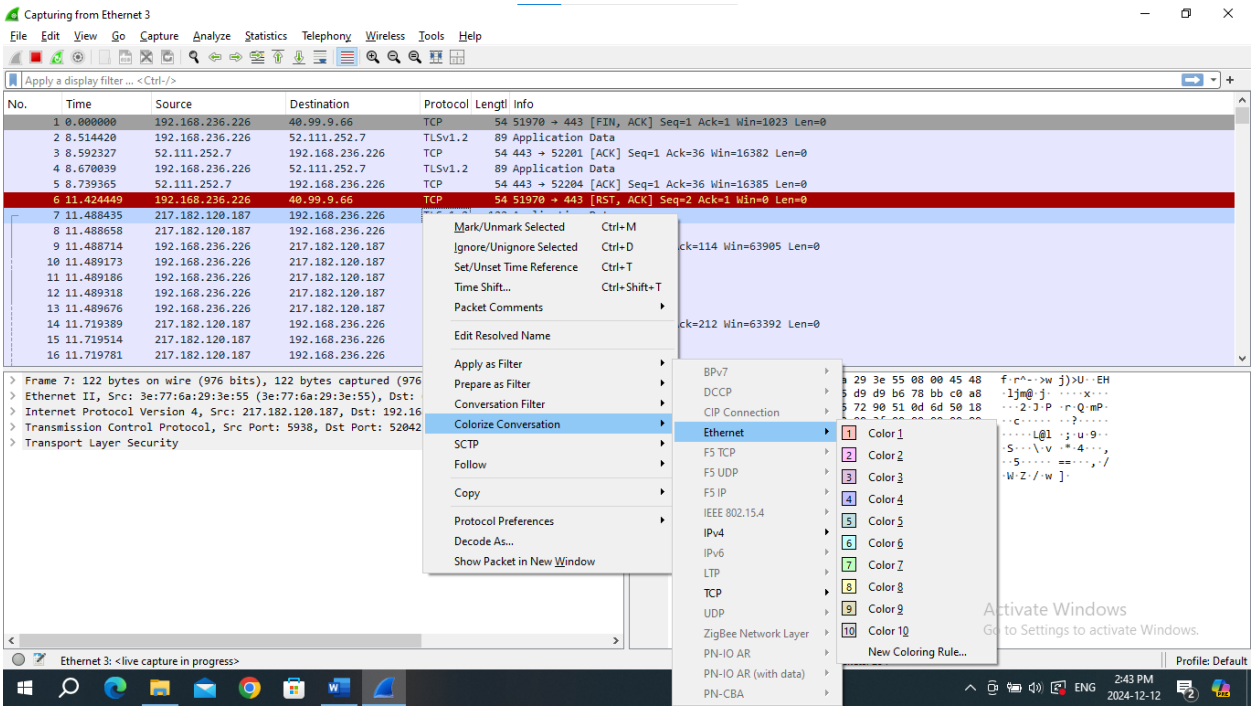
Functionality of Wireshark:

1. **Packet Capture**: Captures live network traffic and displays detailed packet information.
2. **Packet Filtering**: Filters captured data based on protocols, IP addresses, ports, or keywords for focused analysis.
3. **Protocol Analysis**: Decodes and analyzes a wide range of network protocols like TCP, UDP, HTTP, and more.
4. **Traffic Inspection**: Allows in-depth inspection of packet data to detect errors, delays, or anomalies.
5. **Data Visualization**: Provides graphical representations such as flow charts and statistics to analyze traffic trends.
6. **Export and Save**: Saves captured data in various formats for future analysis or sharing with others.

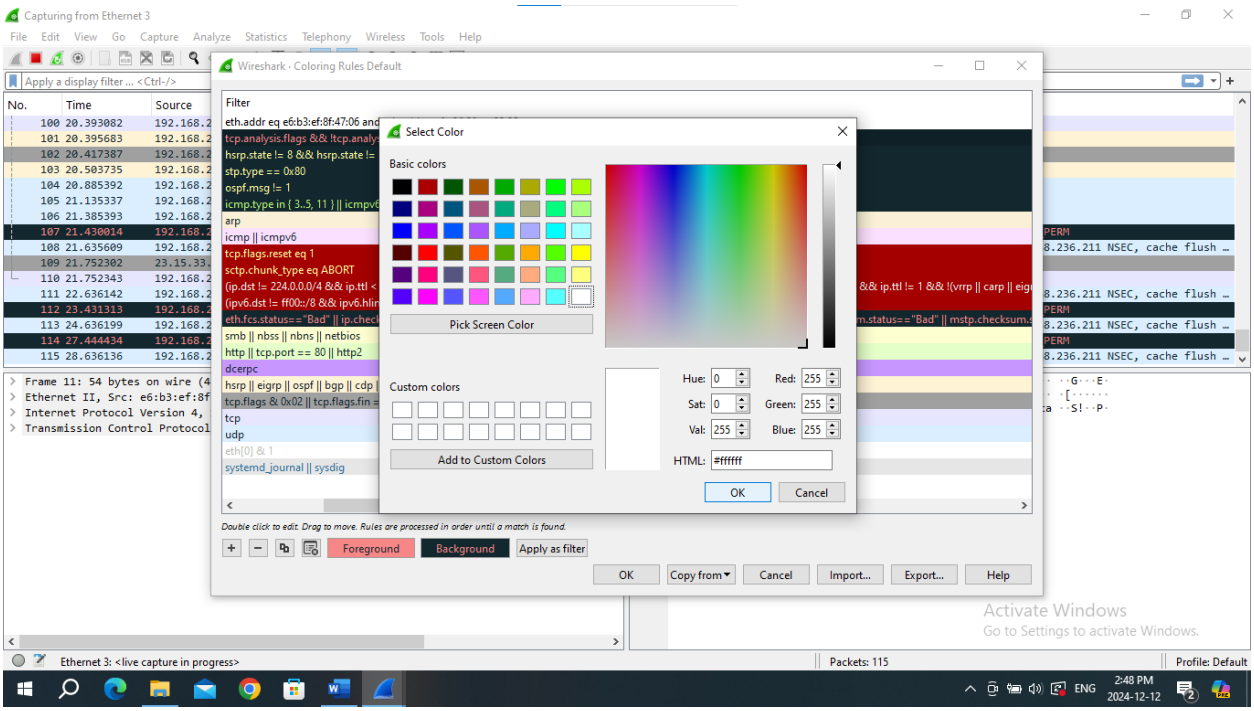


Color Coding in Wireshark:

1. **Default Coloring Rules**: Wireshark uses color coding to highlight different types of traffic for easier analysis.
2. **Light Purple**: Represents TCP traffic.
3. **Light Blue**: Denotes UDP traffic.
4. **Black**: Highlights packets with errors or issues, such as checksum errors.
5. **Green**: Indicates successful and established TCP connections.
6. **Custom Rules**: Users can create custom coloring rules based on filters to identify specific packets quickly.



Enrollment No. : 2403031307008
Div : 4 CC1

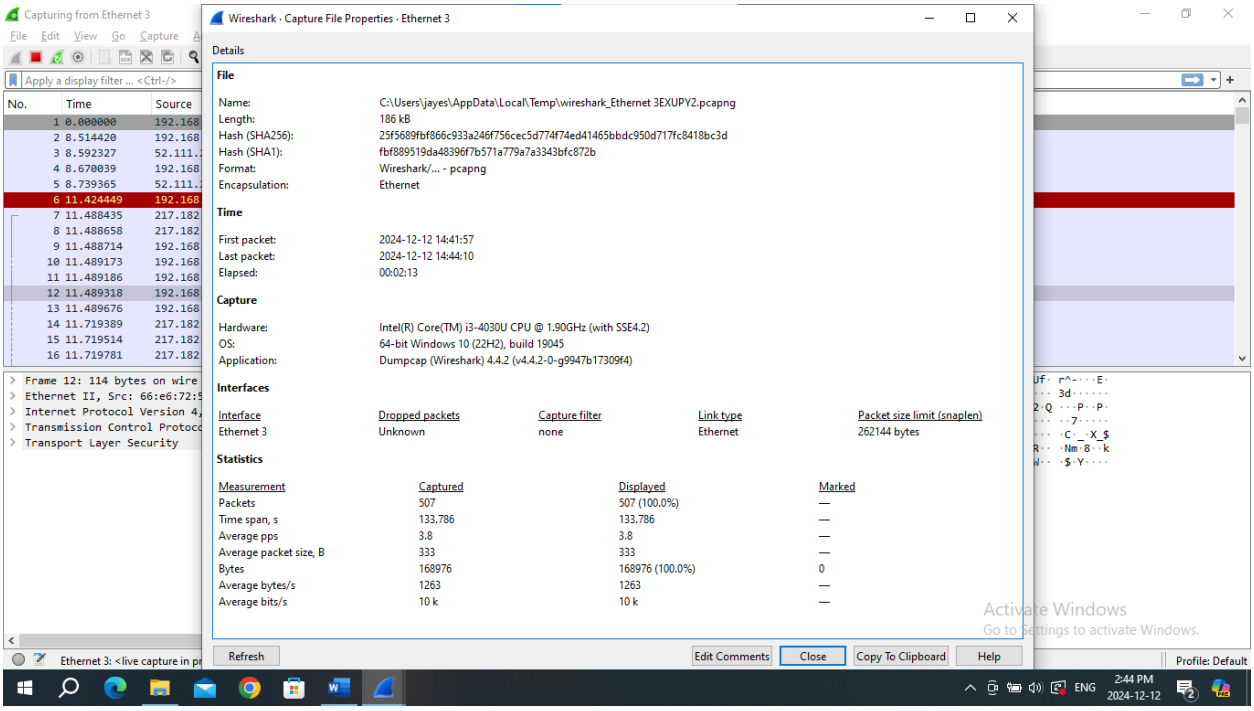


Features of Wireshark:

1. **Deep Packet Inspection**: Allows detailed analysis of individual packets, including headers and payloads.
2. **Real-Time Traffic Capture**: Captures network traffic live from various network interfaces (Ethernet, Wi-Fi, etc.).
3. **Protocol Support**: Supports a wide range of protocols (over 1000), including HTTP, DNS, TCP/IP, and more.
4. **Advanced Filtering**: Provides powerful display filters to focus on specific packets based on conditions like IP address, protocol, or port.
5. **Packet Reassembly**: Reassembles fragmented packets and stream data for a complete analysis of communication sessions.
6. **Statistics and Graphs**: Generates statistics and graphs like protocol hierarchy, packet length distribution, and throughput analysis.
7. **Export Options**: Saves captured data in various formats (PCAP, CSV, etc.) for further analysis or sharing.

Enrollment No. : 2403031307008
Div : 4 CC1

8. **Cross-Platform**: Available on multiple operating systems, including Windows, macOS, and linux.



Conclusion:

In this practical, we explored the functionality and uses of Wireshark, a powerful network packet capture and analysis tool. We learned how Wireshark captures live network traffic, decodes various protocols, and provides deep insights into network behavior. The tool helps in monitoring and troubleshooting network issues by analyzing individual packets, detecting performance bottlenecks, and identifying security vulnerabilities.

We observed the color-coding system used in Wireshark to easily differentiate between types of traffic, such as TCP, UDP, and errors, enhancing the overall user experience. The ability to filter captured packets based on IP addresses, protocols, or ports allowed us to focus on specific network interactions and conduct more efficient analysis.

This practical has provided valuable hands-on experience with Wireshark, demonstrating its importance as an essential tool for network analysis and security monitoring.

Practical – 3

Aim : To study behavior of generic devices used for networking: (CISCO PACKET TRACER)

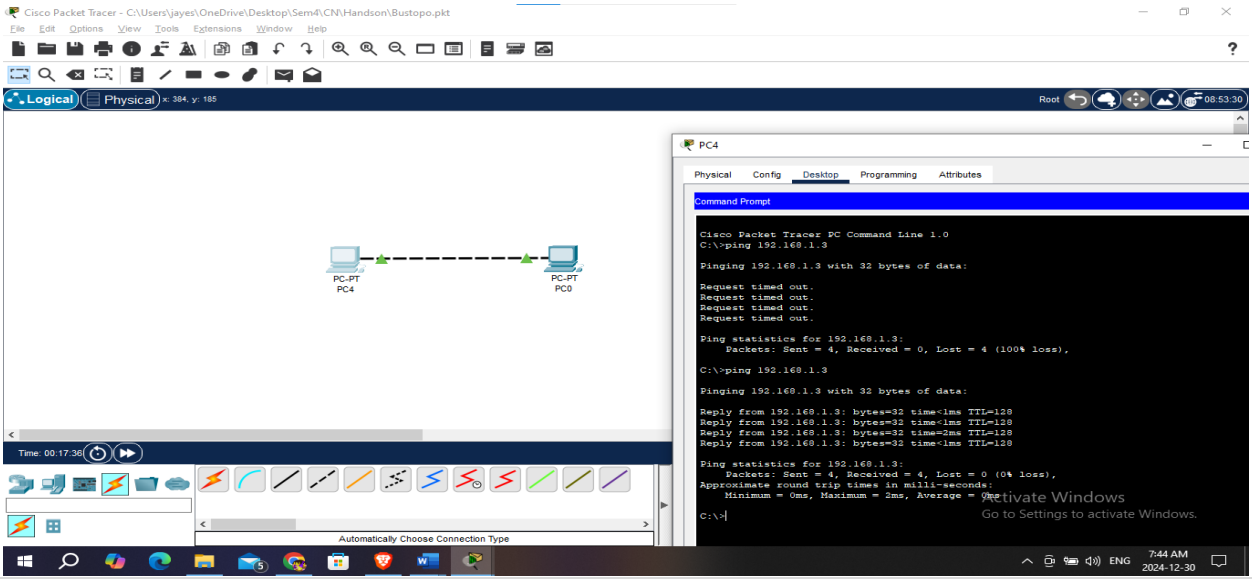
PC (Personal Computer):

Description:

- End-user devices used to send and receive data in a network.
- Operates at the Application Layer of the OSI model.
- Requires IP address configuration for communication.
- Can simulate tasks like ping and data exchange.

Practical Task:

1. Drag and drop two PCs into the Cisco Packet Tracer workspace.
2. Assign IP addresses:
    - **PC1:** IP - 192.168.1.2, Subnet Mask - 255.255.255.0.
    - **PC2:** IP - 192.168.1.3, Subnet Mask - 255.255.255.0.
3. Test communication between PCs by using the **ping** command.
    - On PC1, open **Command Prompt** and type:
    - ping 192.168.1.3



Enrollment No. : 2403031307008
Div : 4 CC1

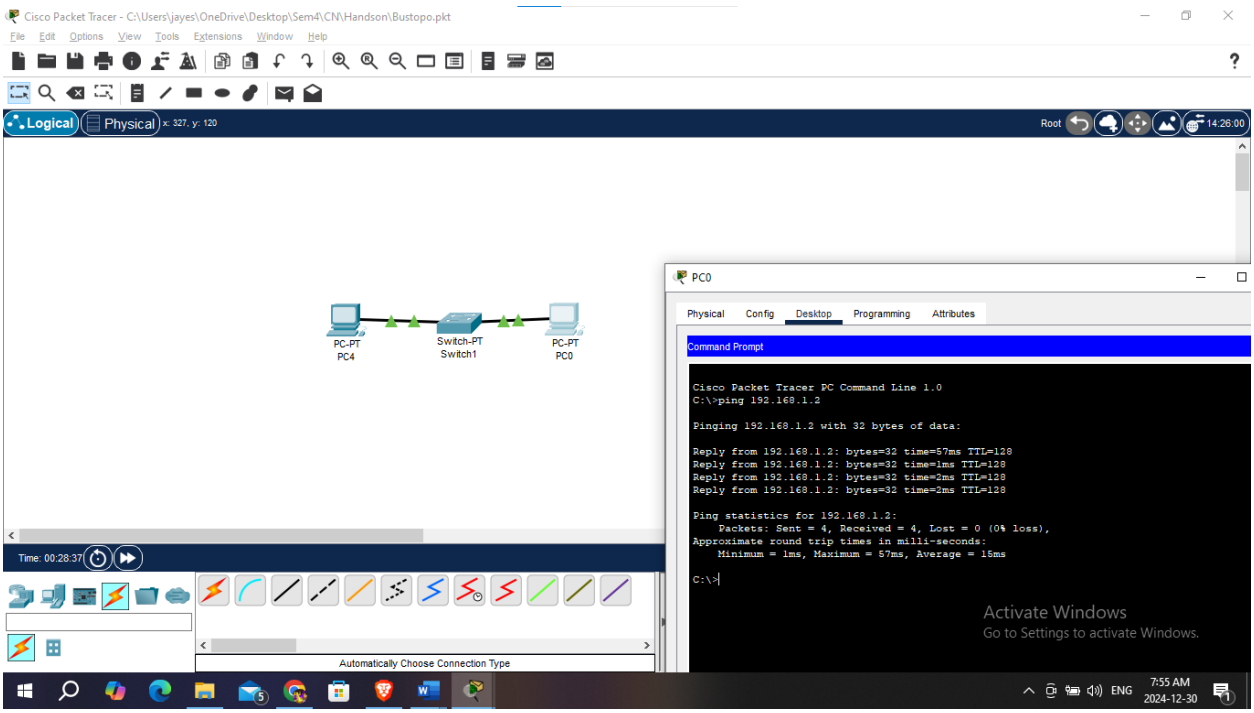
Switch:

Description:

- Layer 2 device in the OSI model for connecting devices within a LAN.
- Forwards data based on MAC addresses, reducing network collisions.
- Supports VLANs and other configurations in managed switches.
- Used for efficient communication within the same subnet.

Practical Task:

1. Drag and drop a Switch into the workspace.
2. Connect the PCs to the Switch using **straight-through cables**:
    - PC1 → Switch (FastEthernet0/1).
    - PC2 → Switch (FastEthernet0/2).
3. Verify connectivity between the PCs by testing with the **ping** command as above.



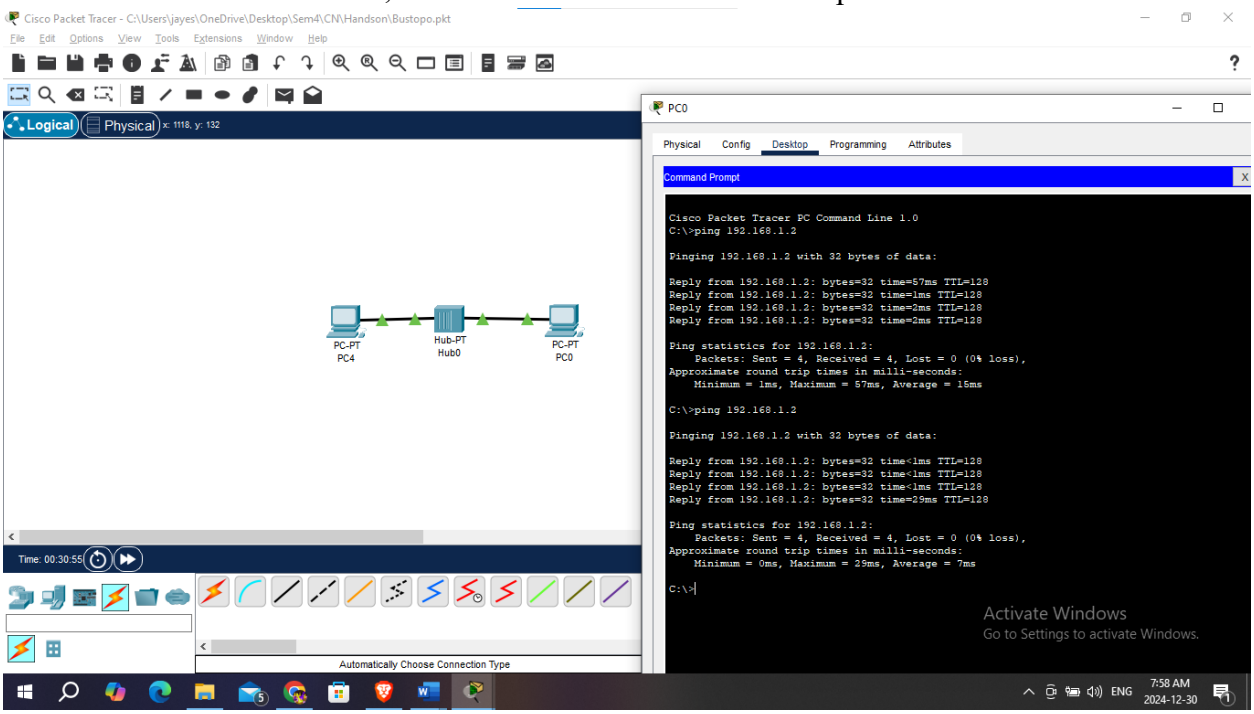
Hub:

Description:

- Layer 1 device that broadcasts data to all connected devices.
- Creates a single collision domain, leading to less efficient performance compared to switches.
- Best suited for small and simple networks.
- Does not require configuration.

Practical Task:

1. Drag and drop a Hub into the workspace.
2. Connect the PCs to the Hub using **straight-through cables**:
    - PC1 → Hub (Port 1).
    - PC2 → Hub (Port 2).
3. Test communication between the PCs using the **ping** command. Observe that communication works, but the Hub broadcasts data to all ports.



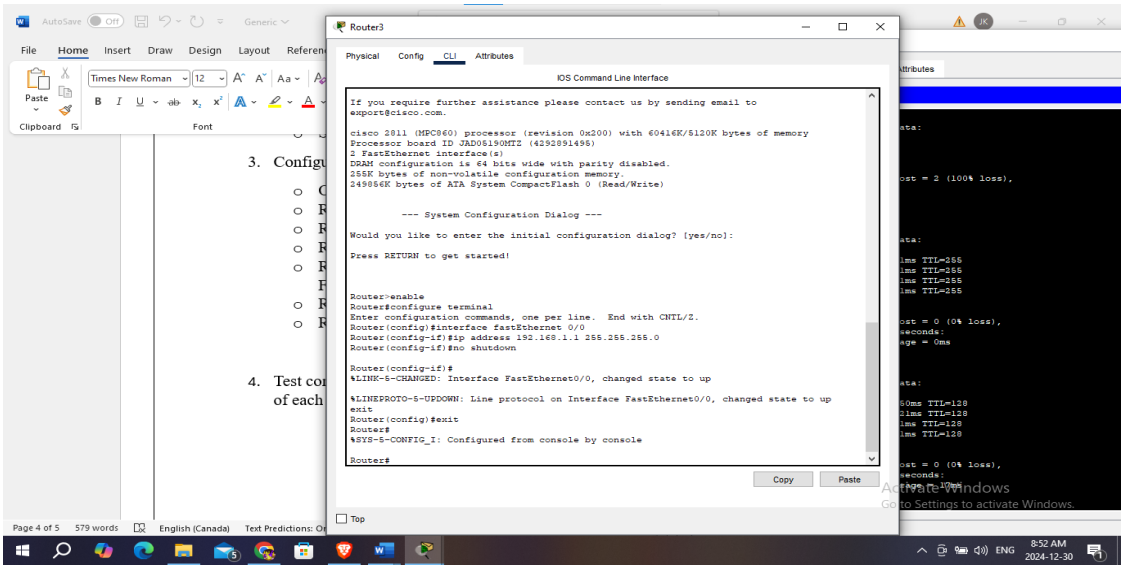
Router:

Description:

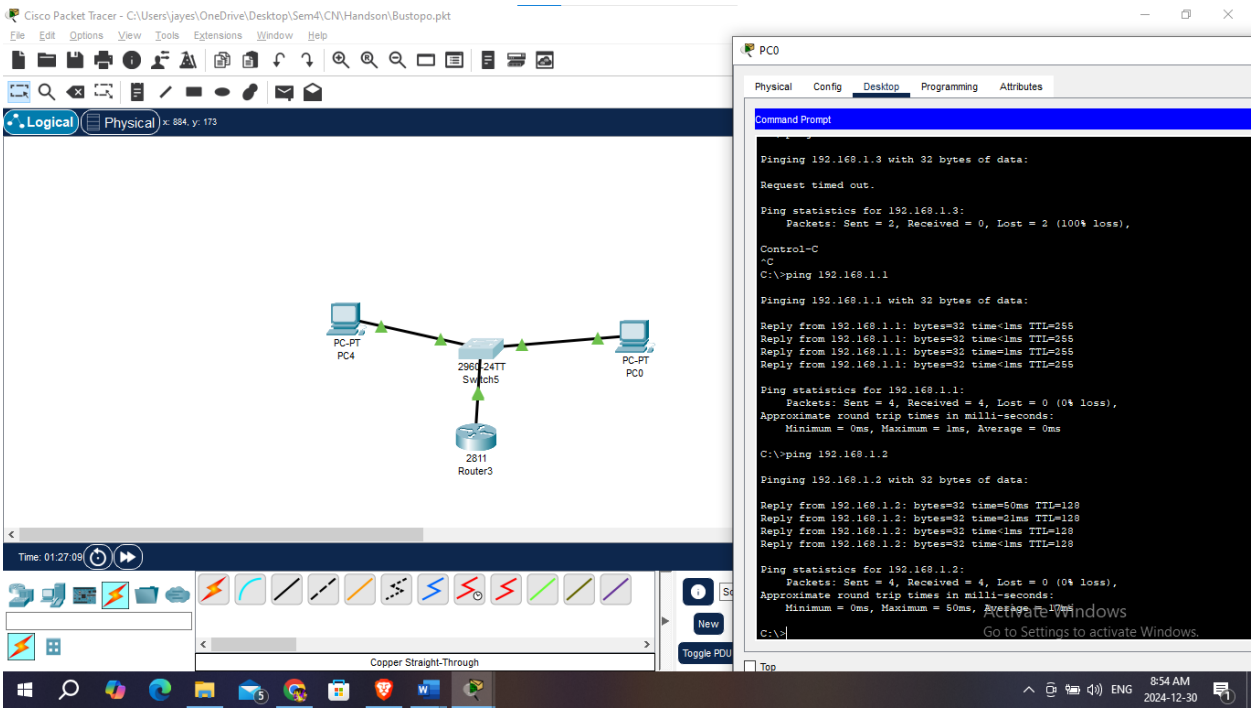
- Layer 3 device that connects multiple networks and routes packets based on IP addresses.
- Determines the best path for data transmission.
- Requires configuration of interfaces and routing protocols.
- Essential for inter-network communication.

Practical Task:

1. Drag and drop a Router into the workspace.
2. Connect the Switch to the Router using a **straight-through cable**:
   - Switch (FastEthernet0/0) → Router.
3. Configure the Router:
   - Open the **CLI** tab on the Router and execute the following commands:
   - Router> enable (Enter privileged exec mode)
   - Router# configure terminal (Enter global configuration mode)
   - Router(config)# interface fastEthernet 0/0 (Select interface connected to PC1)
   - Router(config-if)# ip address 192.168.1.1 255.255.255.0 (Assign IP address to FastEthernet0/0)
   - Router(config-if)# no shutdown (Enable the interface)
   - Router(config-if)# exit (Exit interface configuration mode)



4. Test communication between the PCs through the Router by setting the default gateway of each PC to 192.168.1.1.



Conclusion:

In this practical, we successfully studied and configured the behavior of various networking devices using Cisco Packet Tracer. By simulating the operation of **PCs**, **Switches**, **Hubs**, and **Routers**, we gained a better understanding of their roles in a network.

- **PCs** allowed us to simulate end-user communication by assigning IP addresses and testing connectivity using the ping command.
- **Switches** facilitated efficient data transfer between connected devices, showcasing their role in reducing network collisions.
- **Hubs**, while still functional, demonstrated less efficient data handling by broadcasting data to all devices in the network.
- **Routers** provided the crucial role of inter-network communication, routing data between different networks and managing traffic with IP addressing.

Overall, this practical exercise reinforced key networking concepts and provided hands-on experience in configuring and understanding the behavior of networking devices in a simulated environment. This knowledge is fundamental for building and troubleshooting real-world networks.