KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A++" Grade
Anand Nagar, Krishnankoil, Srivilliputtur (Via), Virudhunagar (Dt) - 626126, Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

212INT3301 – DATA COMMUNICATIONS AND COMPUTER NETWORKS

LAB MANUAL

ACADEMIC YEAR (2024 -2025) ODD SEMESTER

Bachelor of Technology (Information Technology)

III Year / V Semester

UNIVERSITY VISION	UNIVERSITY MISSION
To be a University of Excellence of International Repute in Education and Research.	1) To provide a scholarly teaching-learning ambience which results in creating graduates equipped with skills and acumen to solve real-life problems. 2) To promote research and create knowledge for human welfare, rural and societal development. 3)To nurture entrepreneurial ambition, industrial and societal connect by creating an environment through which innovators and leaders emerge.

DEPARTMENT OF INFORMATION TECHNOLOGY	
VISION	**MISSION**
To be a department of repute offering programmes in frontier areas of IT through quality education, research and imbibing societal values.	1)To provide quality education through effective curriculum and innovative teaching. 2)To facilitate conducive learning environment for students and faculty to investigate, apply and transfer knowledge. 3)To instill the ethical behaviour and social responsibilities to provide sustainable information technology solutions.

B.TECH. IT PROGRAMME EDUCATIONAL OBJECTIVES(PEO'S)

Within a few years of obtaining an undergraduate degree in Information Technology, the students will be able to:

PEO-1: The graduates will be successful IT professionals in their chosen area and / or pursue higher studies.

PEO-2: The graduates will comprehend, analyze, design and create novel products and technologies that provide sustainable solutions.

PEO-3: The graduates will demonstrate multidisciplinary knowledge, personal and interpersonal skills and work as an effective team member with ethical standards.

B.TECH. IT. ABET OUTCOMES

At the end of the programme, the students will be able to:

AO1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

AO2: Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

AO3: Communicate effectively in a variety of professional contexts.

AO4: Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.

AO5: Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.

AO6: Identify and analyse user needs and to take them into account in the selection, creation, integration, evaluation, and administration of computing-based systems.

B.TECH. IT PROGRAMME SPECIFIC OUTCOMES (PSO'S)

PSO-1: Ability to identify, design and develop processes and systems for enterprises.

PSO-2: Ability to identify, deploy and maintain the IT infrastructure based on the needs of the businesses.

PSO-3: Practice and promote information technologies for societal needs.

B.TECH. IT PROGRAMME OUTCOMES

At the end of the programme, the students will be able to:

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

SYLLABUS

212INT3301	DATA COMMUNICATION AND COMPUTER NETWORKS	L	T	P	C
		0	0	2	4

COURSE OUTCOMES:

CO 1: Inspect the basics of data communication and various categories of networks and its securities
CO 2: Identify the technologies for error free secure transmission of data in data link layer
CO 3: Apply various routing protocols to select optimal path and relate addressing entities in Network layer
CO 4: Analyze the various security protocols at different layers of OSI architecture
CO 5: Analyze the various protocols in application layer
CO 6: Understand and apply different network commands using packet tracer
CO 7: Analyze and apply the different networking concepts for implementing network solution through working as a team and communicate effectively with technical community in both the written and oral forms

LSIT OF EXPERIMENTS

1)Study of different types of network cables

2)Study of various networking devices

3) Study of Basic Network commands and Network Configuration Commands.

4)Checking Layer 2 functionality using packet tracer

Topologies - Ring Topology, Mesh Topology, Bus Topology, Star Topology

5)Checking Layer 2 functionality using packet tracer.

a) Configure Spanning Tree Protocol.

b) Configure ARP and MAC Table.

6) Checking Layer 3 functionality using packet tracer.

7) Network Protocol Analysis:

a. Capture and Analyze ICMP Packets

b. Capture and Analyze ARP frame

c. Capture and Analyze TCP Segment

d. Capture and Analyze UDP Datagram

e. Capture and Analyze IP Packets

8) Domain Name Service

WEIGHTAGE:

Mid Semester Practical	Regular Laboratory Performance	End Semester Examination
10%	5%	Pratical-15%

Exp No:

Checking Layer 2 Functionality using Packet Tracer 7.3.0

Aim:

To create ring topology and checking layer 2 functionality by using Packet Tracer 7.3.0.

Requirement:

1.Packet Tracer 7.3.0 Tool

2.PC

3.Switch

4.Connecting Wires

Theory:

About Packet Tracer

Packet Tracer is a network simulation tool designed by Cisco Systems. It is widely used in educational and training settings to help network professionals to develop their skills and knowledge. It allows users to create network topologies by dragging and dropping routers, switches, and other network devices and imitate modern computer networks. It also enables users to experiment with network behaviour, configuration, and troubleshooting.

About Layer 2 of OSI Model

Layer 2 of Open System Interconnection Model is data link layer. The primary responsibility of data link layer is node to node (hop to hop) delivery of data and also ensures error free transmission of data.

Other Responsibilities of data link layer

1) Framing- The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

2)Physical Addressing- If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame.

3)Flow Control- If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

4) Error Control- The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

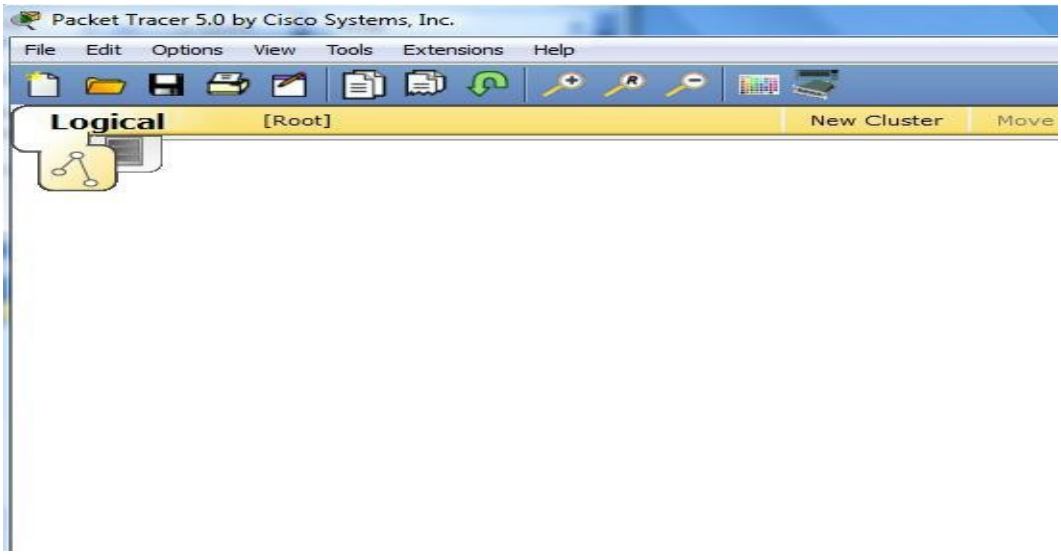
5)Access Control- When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Procedure:

Ring Topology

Is a kind of arrangement of the network in which every device is linked with two other devices. This makes a circular ring of interconnected devices. Data is usually transmitted in one direction along the ring, known as a unidirectional ring. The data is delivered from one device to the next until it reaches the decided destination. In a bidirectional ring, data can travel in either direction.

Step 1: Start Packet Tracer



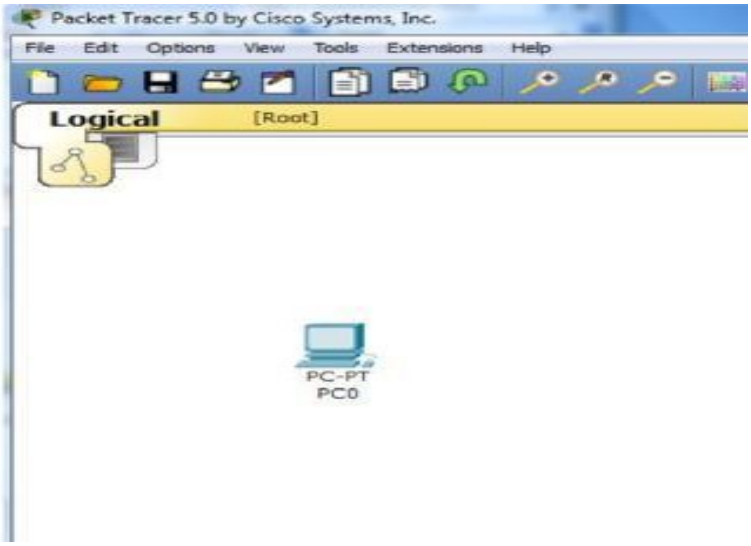
Step 2: Choosing Devices and Connections

Ring topology is created by selecting end devices, network devices and the media in which to connect them.

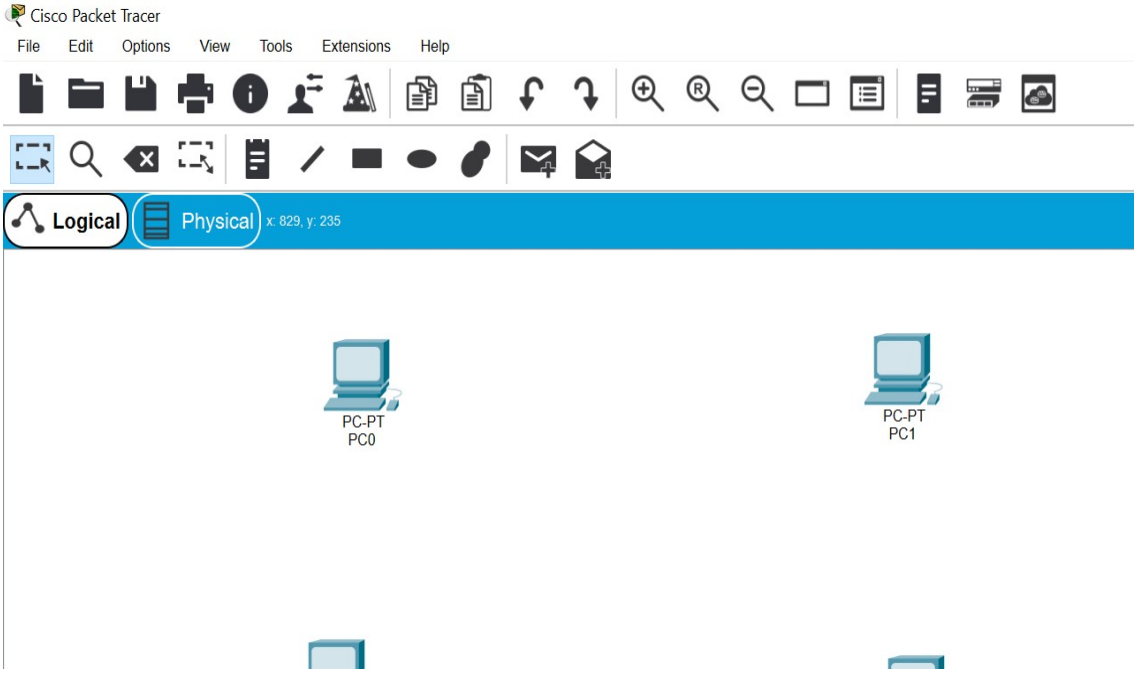
Single click on the **End Devices** and Single click on **Generic End Devices**.

Move the cursor into topology area. We will notice it turns into a plus "+" sign.

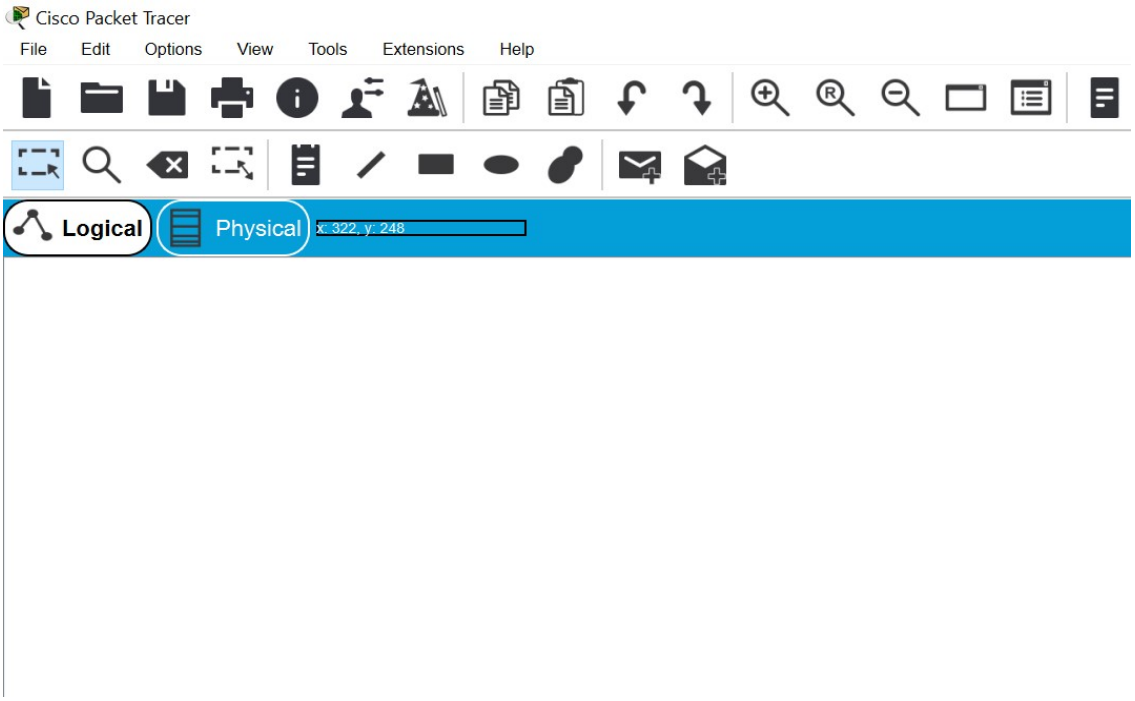
Single click in the topology area and it copies the device.



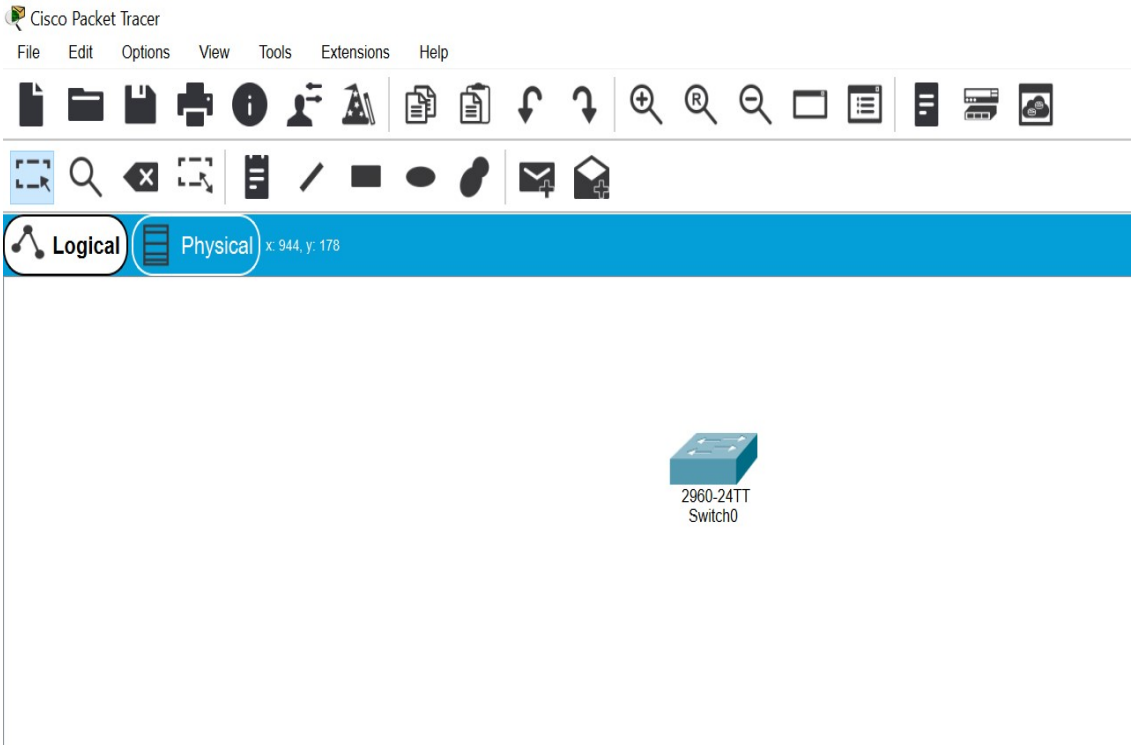
If required, add two or more devices to topology area.



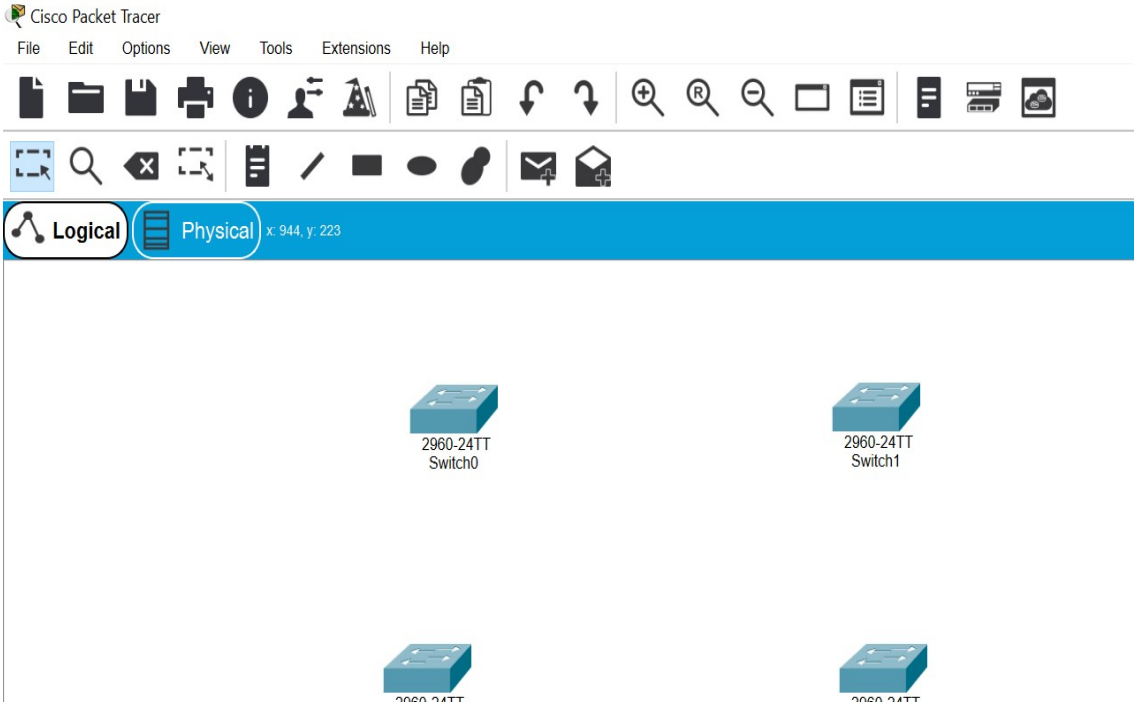
Single click on the **Network Devices(Switches)** and Single click on generic switch.



Move the cursor into topology area. You will notice it turns into a plus “+” sign. Single click in the topology area and it copies the device.

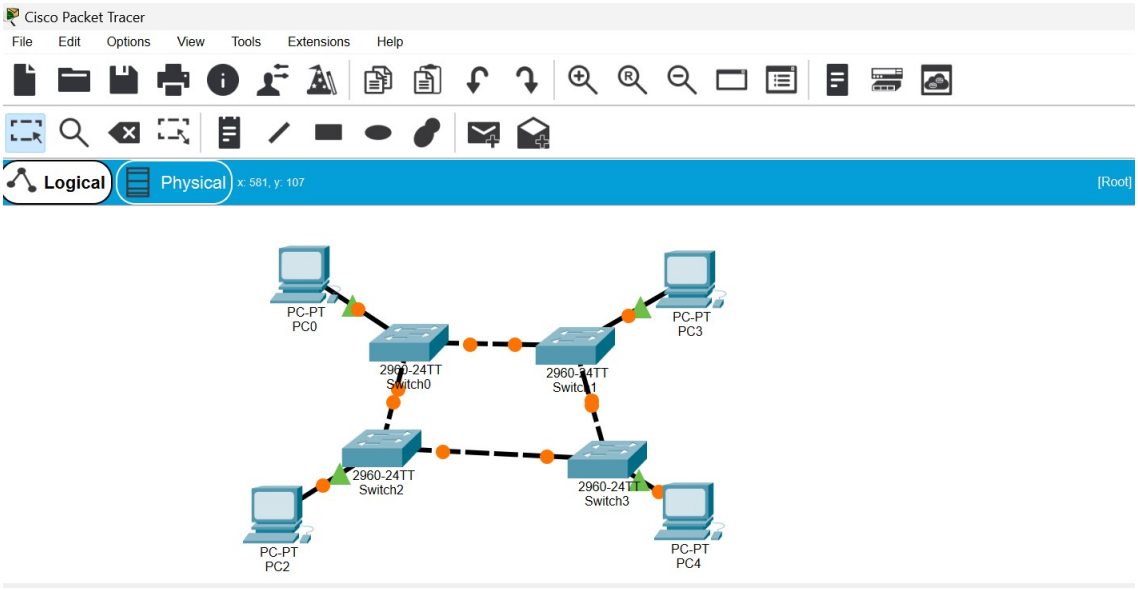


If required, add two or more switches to topology area.



Step 3: Connecting the Hosts to Switches

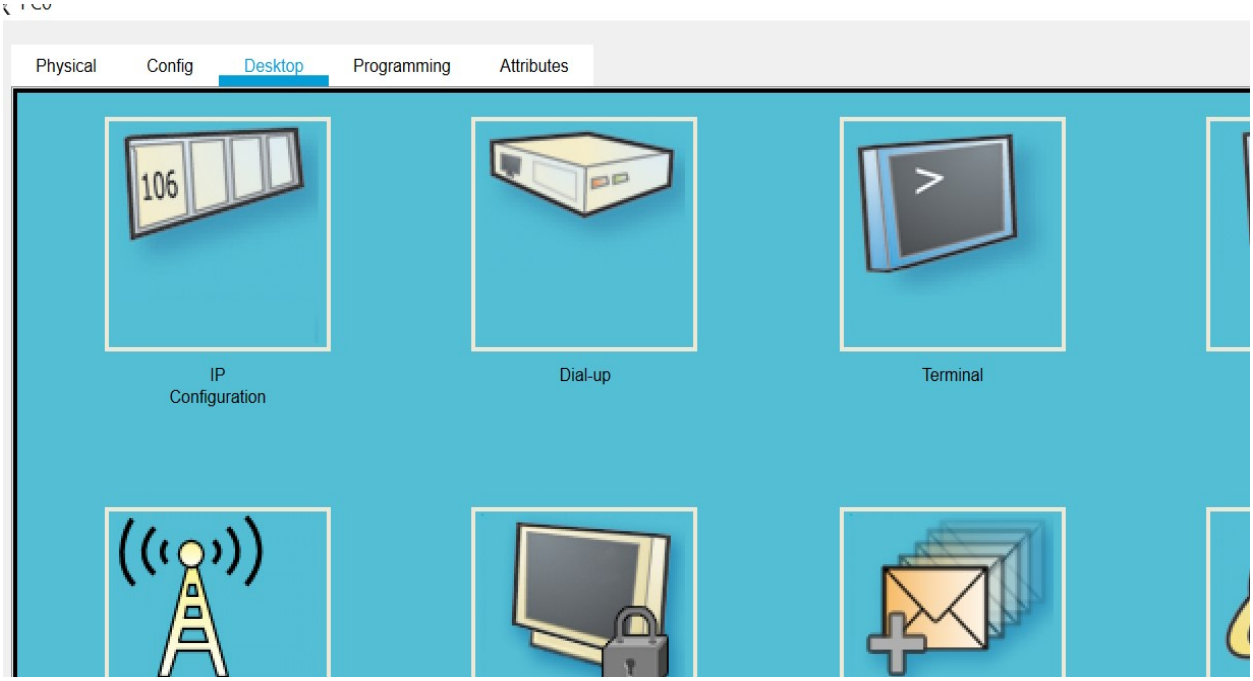
Connect End Devices to Switch by first choosing **Connections (Automatically Choose Connection Type).**

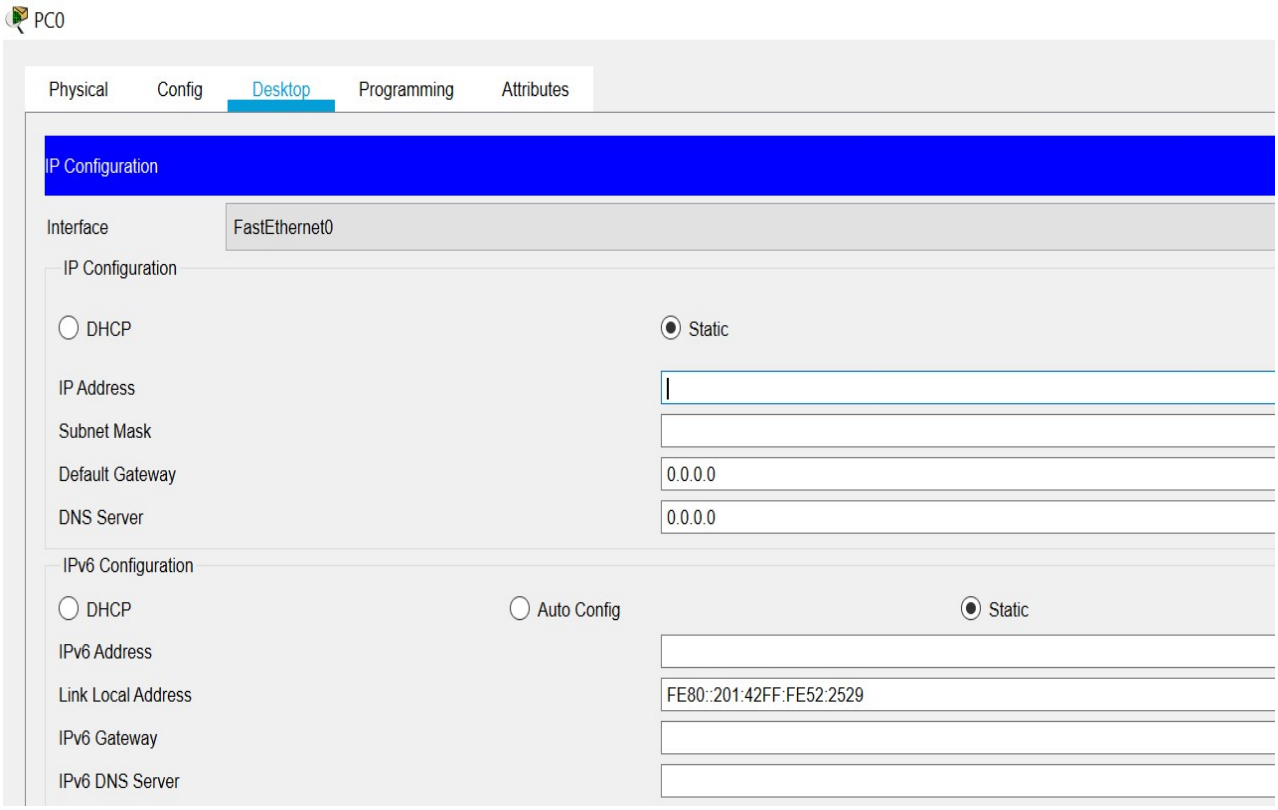


Step 4: Configuring IP Address for all end devices in the topology area

Before we can communicate between the hosts, we need to configure IP Addresses on the devices.

Click on End Device and go to desktop menu, then go to IP configuration and set the IP Adress.

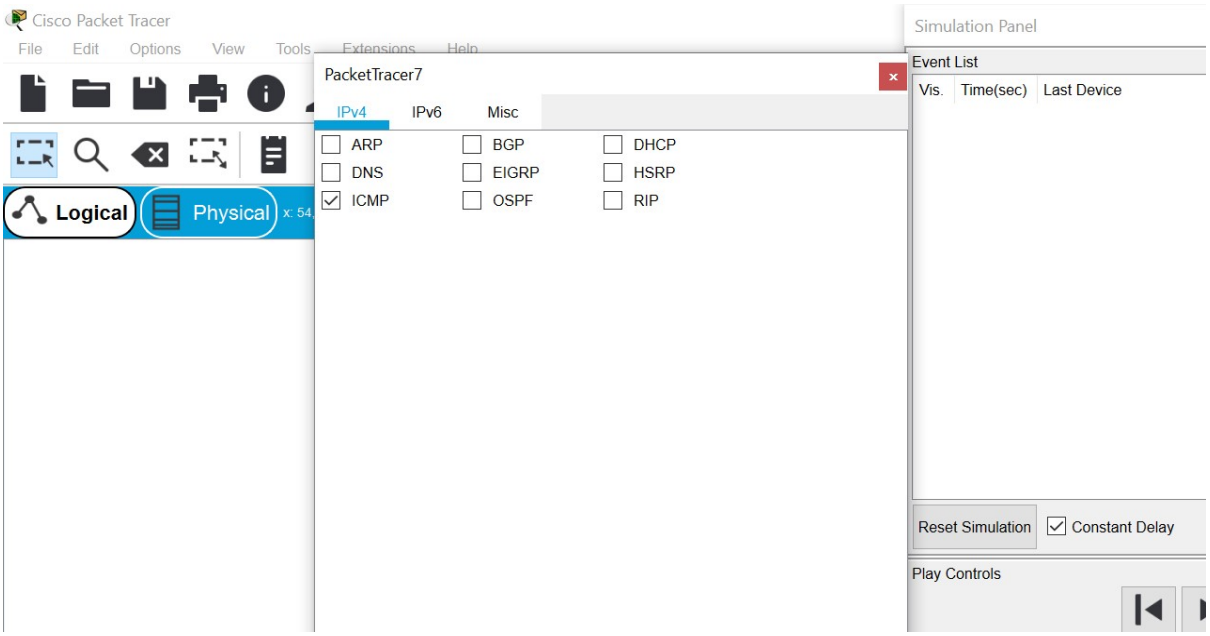




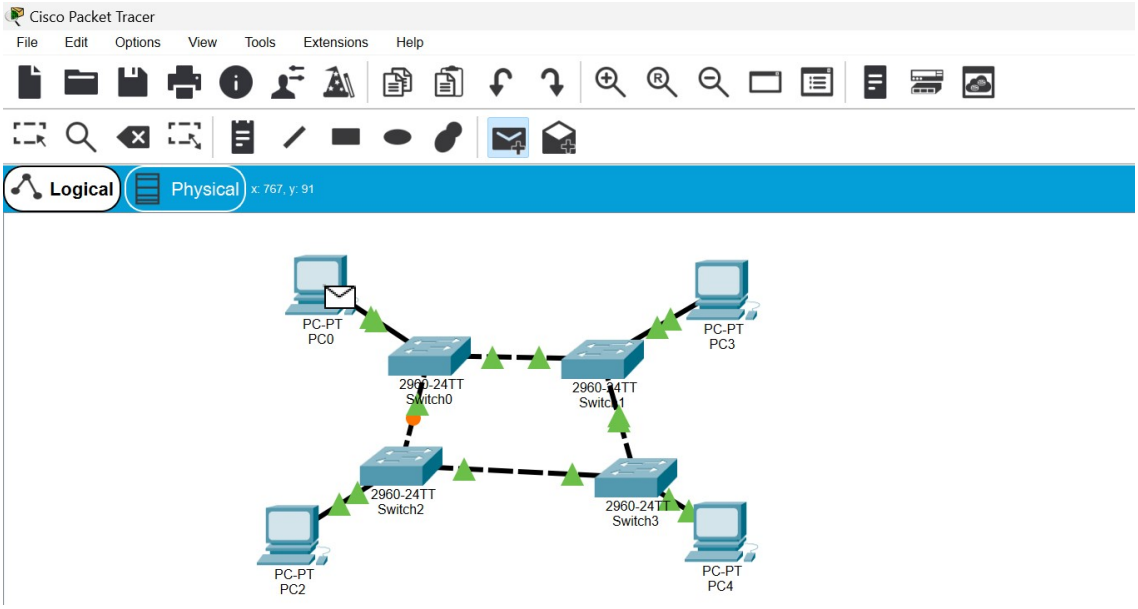
Repeat these steps for the remaining end devices on the network to create IP Address.

Step 5: Verifying Connectivity in Simulation Mode

Be sure you are in **Simulation** mode. To ensure this, go to view tab in packet tracer and press simulation mode.Deselect all filters (All/None) and select only **ICMP**.



Select the **Add Simple PDU** tool to ping devices. Click once on one end device, then once on another end device.



Continue clicking **Capture/Forward** button until the ICMP ping is completed. You should see the ICMP messages move between the hosts, hub and switch. The PDU **Last Status** should show as **Successful**.

Result:

Thus the ring topology was created using packet tracer and simulated and studied.

Exp No:

Checking Layer 2 Functionality using Packet Tracer 7.3.0

Aim:

To create mesh topology and checking layer 2 functionality by using Packet Tracer 7.3.0.

Requirement:

1.Packet Tracer 7.3.0 Tool

2.PC

3.Switch

4.Connecting Wires

Theory:

About Packet Tracer

Packet Tracer is a network simulation tool designed by Cisco Systems. It is widely used in educational and training settings to help network professionals to develop their skills and knowledge. It allows users to create network topologies by dragging and dropping routers, switches, and other network devices and imitate modern computer networks. It also enables users to experiment with network behaviour, configuration, and troubleshooting.

About Layer 2 of OSI Model

Layer 2 of Open System Interconnection Model is data link layer. The primary responsibility of data link layer is node to node (hop to hop) delivery of data and also ensures error free transmission of data.

Other Responsibilities of data link layer

1) Framing- The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

2)Physical Addressing- If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame.

3)Flow Control- If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

4) Error Control- The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

5)Access Control- When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Procedure:

Mesh Topology

In this topology, every device has a dedicated point-to-point link to every other device. The term dedicated means that the link carries traffic only between the two devices it connects. The Internet is an example of the mesh topology. Mesh topology is mainly used for wireless networks.

Types of Mesh Topology

1. Full Mesh topology
2. Partial mesh topology

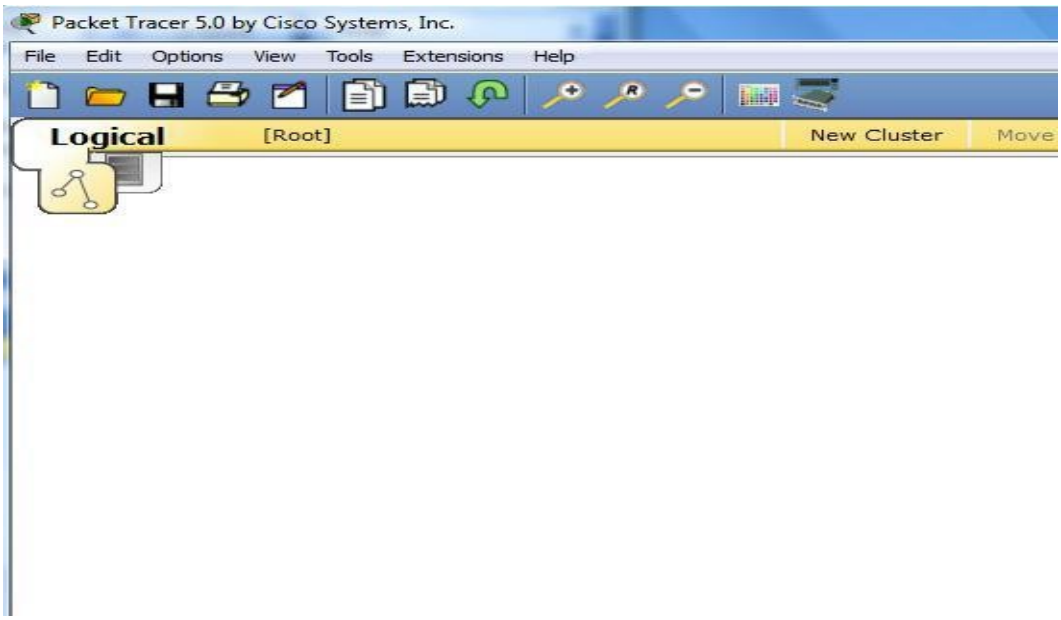
1. Full Mesh Topology

In a full mesh topology, every computer in the network has a connection to each of the other computers via cable.

2. Partial Mesh Topology

In partial mesh topology, all computers are not connected to each other.

Step 1: Start Packet Tracer



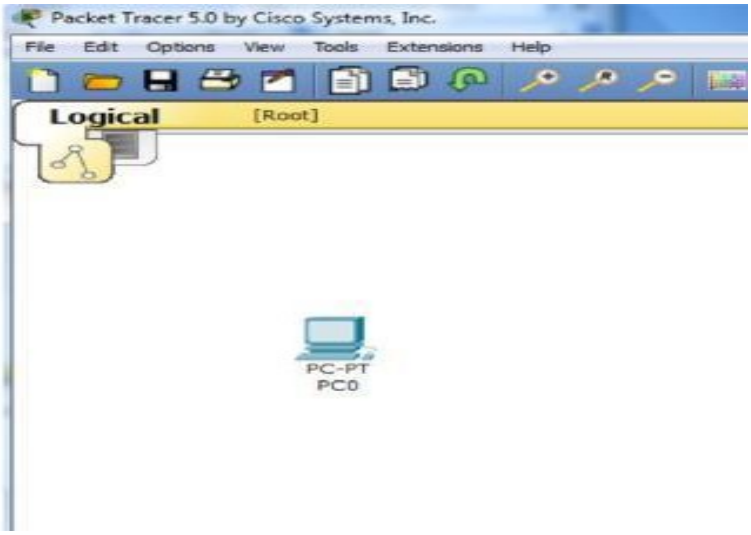
Step 2: Choosing Devices and Connections

Mesh topology is created by selecting end devices, network devices and the media in which to connect them.

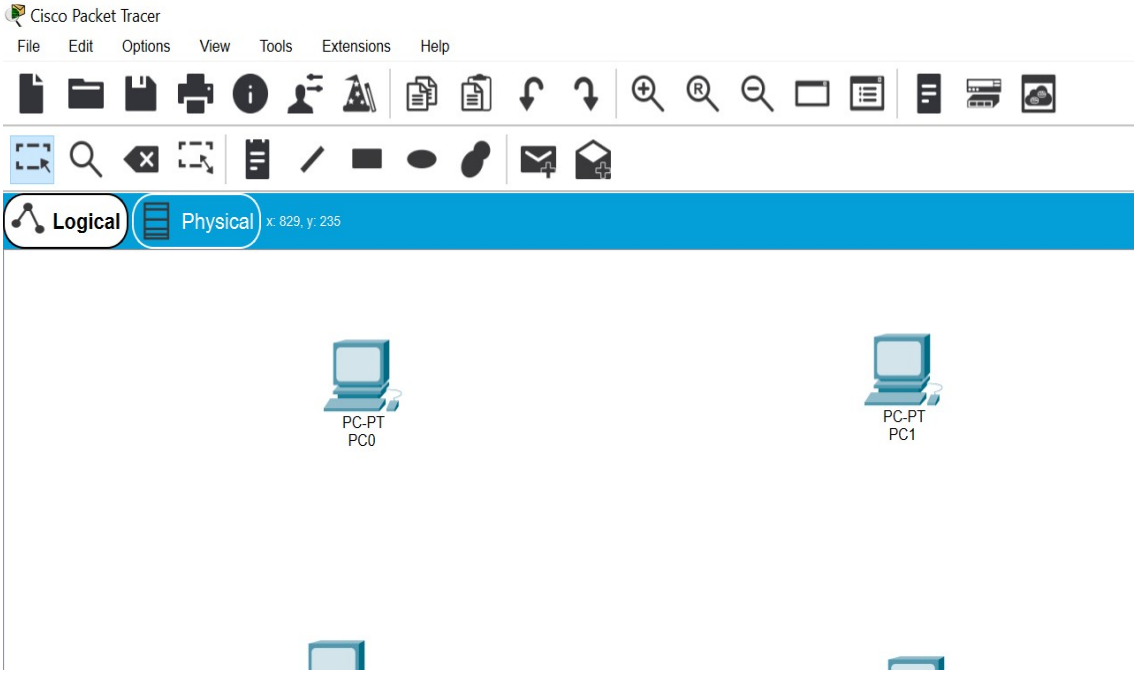
Single click on the **End Devices** and Single click on **Generic End Devices**.

Move the cursor into topology area. We will notice it turns into a plus “+” sign.

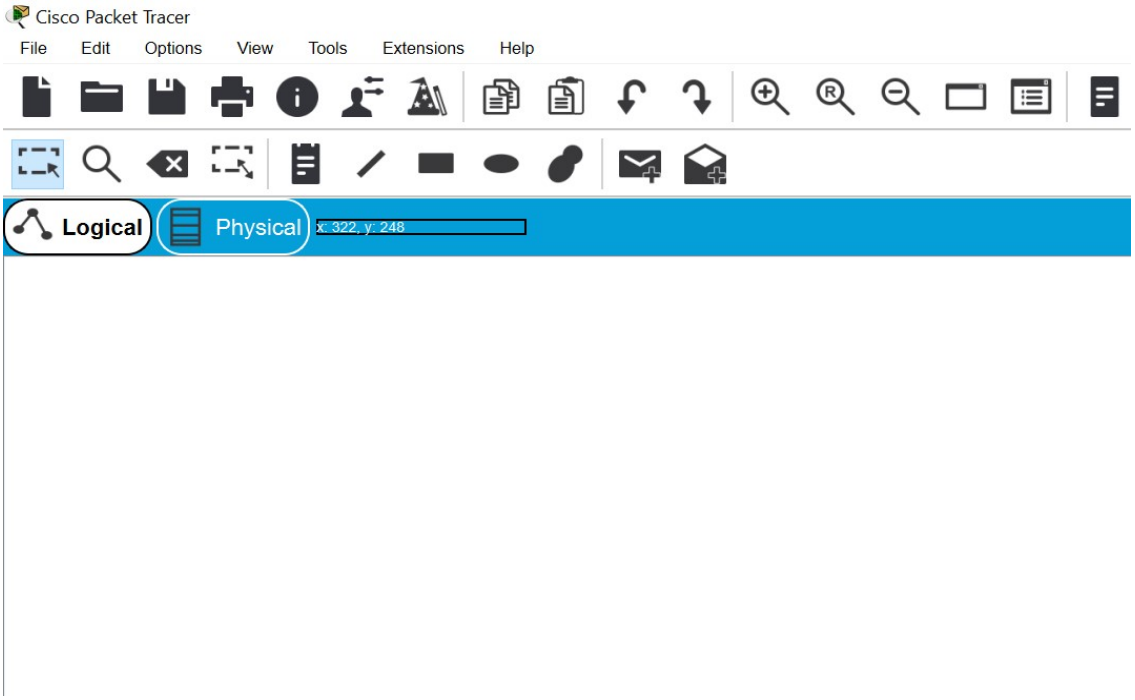
Single click in the topology area and it copies the device.



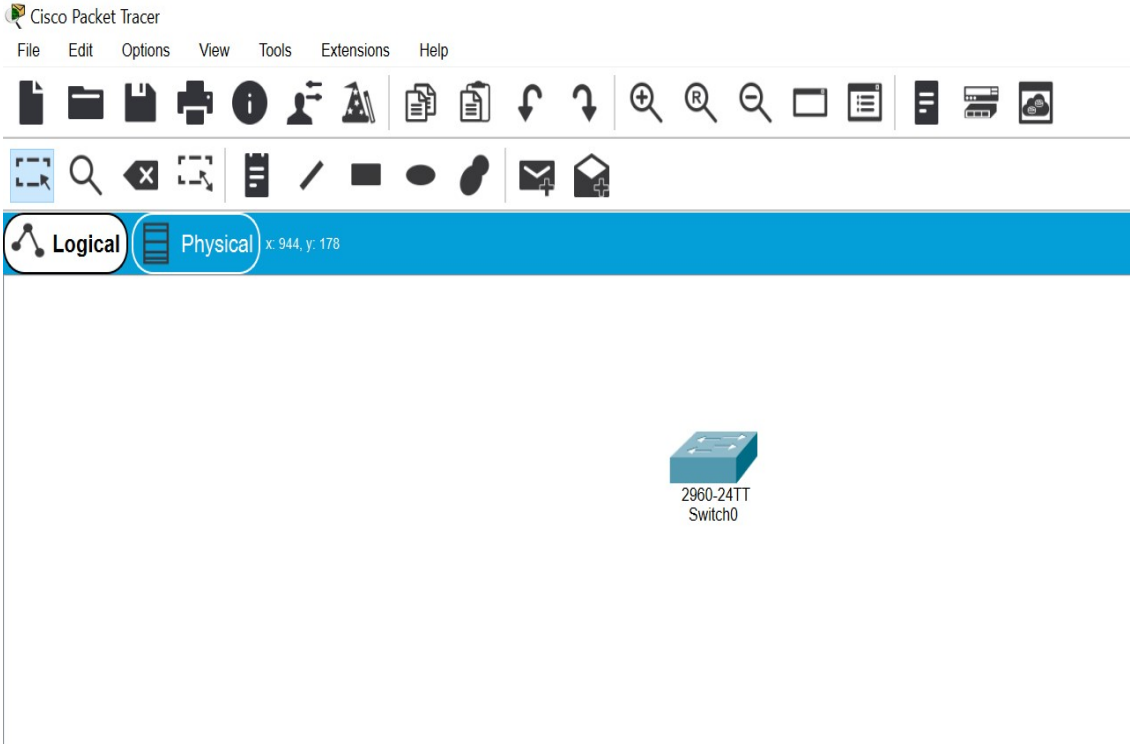
If required, add two or more devices to topology area.



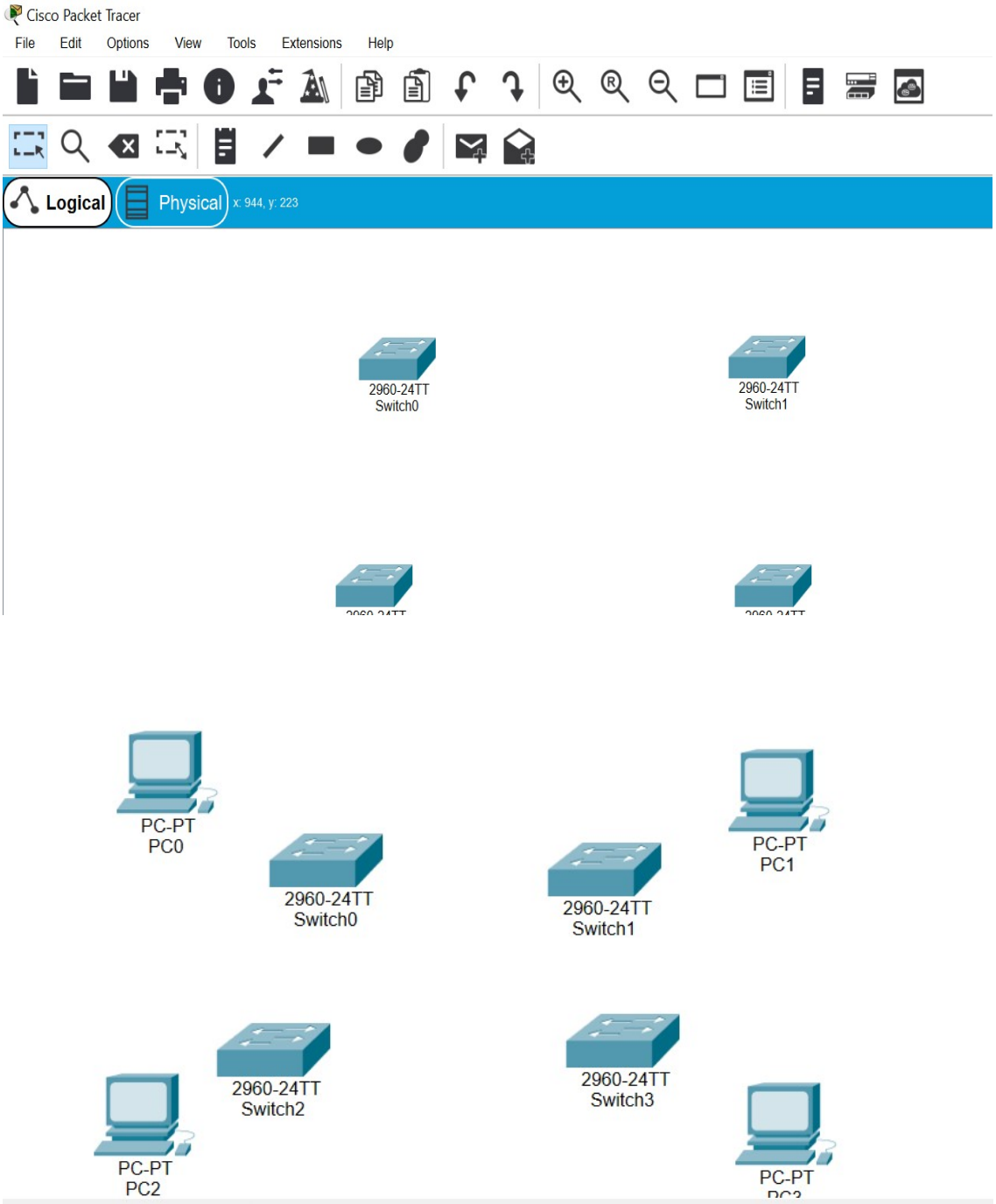
Single click on the **Network Devices(Switches)** and Single click on generic switch.



Move the cursor into topology area. You will notice it turns into a plus “+” sign. Single click in the topology area and it copies the device.

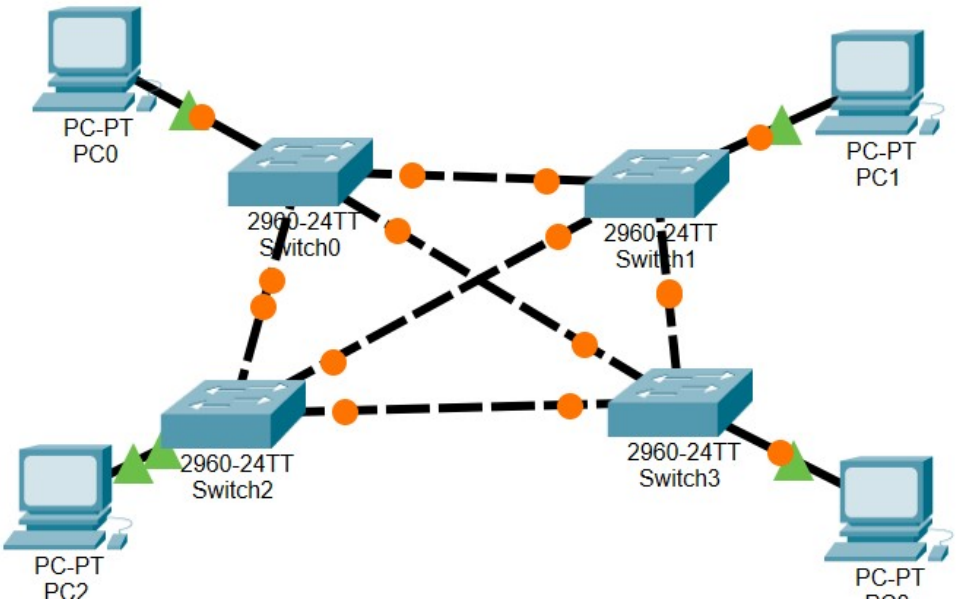


If required, add two or more switches to topology area.



Step 3: Connecting the Hosts to Switches

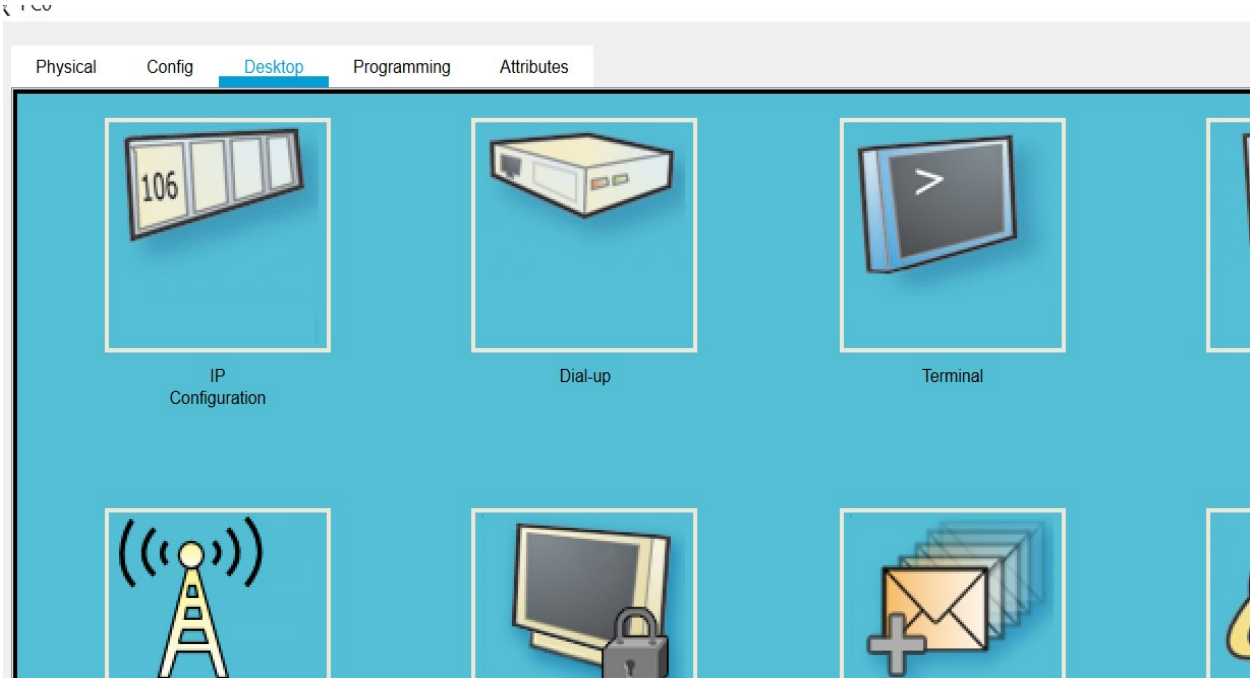
Connect End Devices to Switch by first choosing **Connections (Automatically Choose Connection Type).**

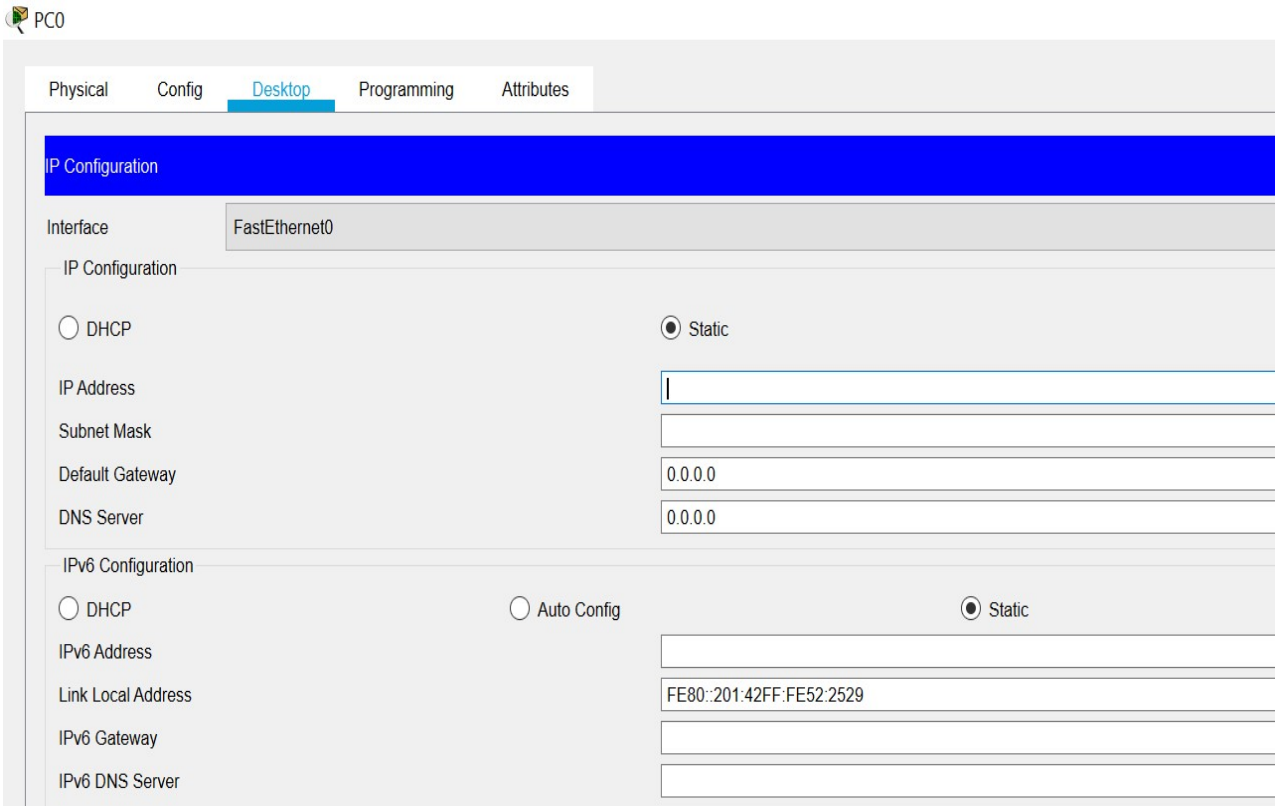


Step 4: Configuring IP Address for all end devices in the topology area

Before we can communicate between the hosts, we need to configure IP Addresses on the devices.

Click on End Device and go to desktop menu, then go to IP Configuration and set the IP Adress.



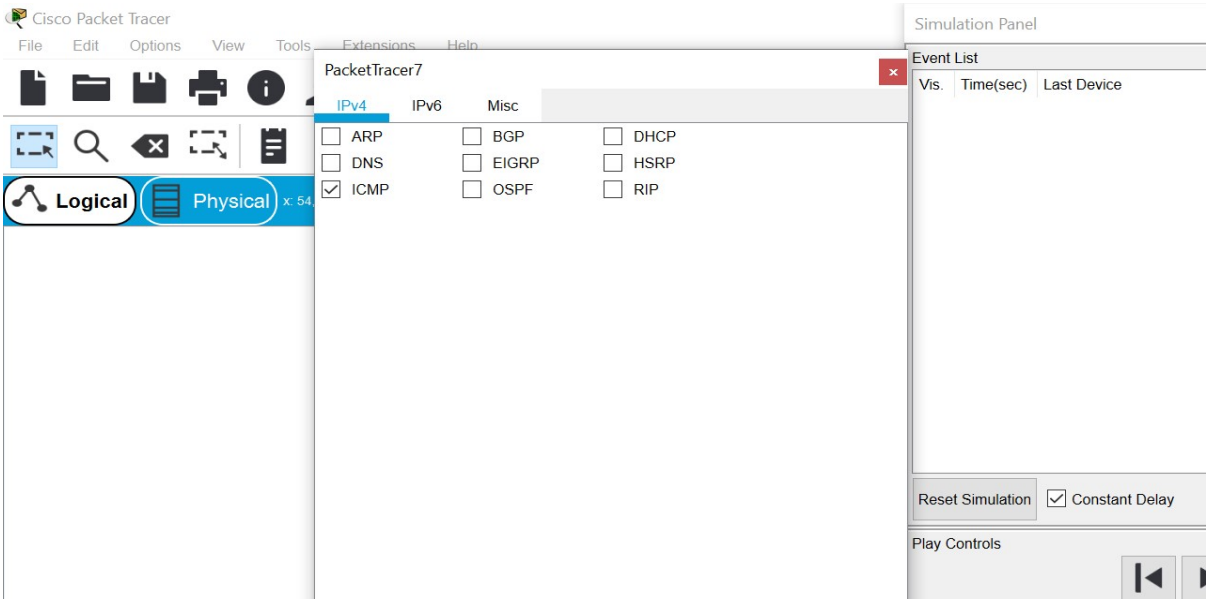


Repeat these steps for the remaining end devices on the network to create IP Address.

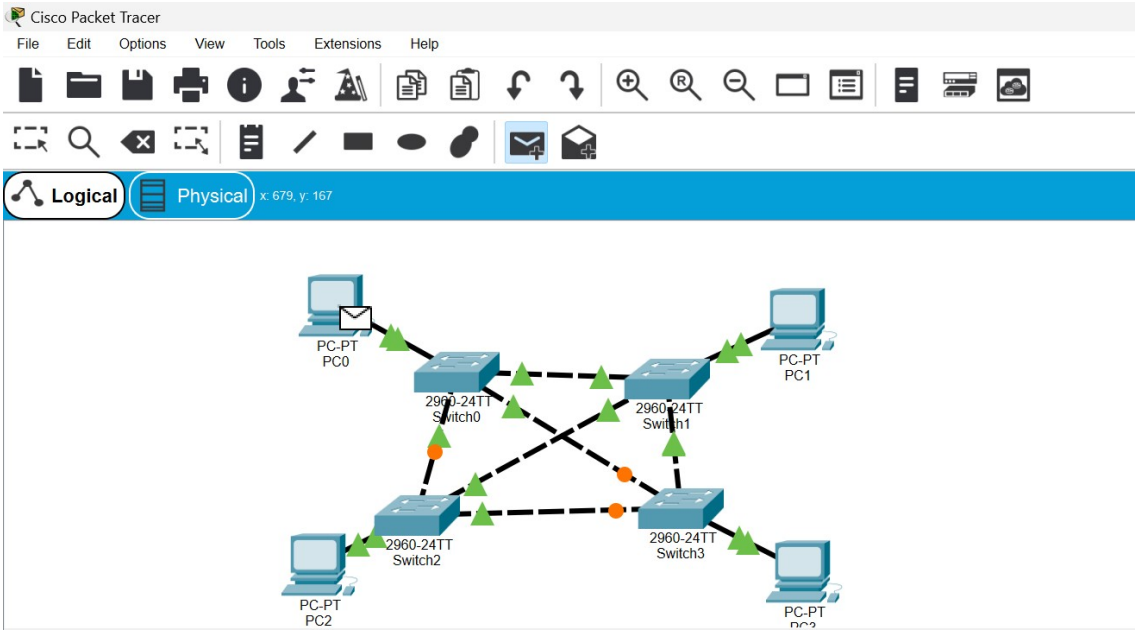
Step 5: Verifying Connectivity in Simulation Mode

Be sure you are in **Simulation** mode. To ensure this, go to view tab in packet tracer and press simulation mode.

Deselect all filters (All/None) and select only **ICMP**.



Select the **Add Simple PDU** tool to ping devices. Click once on one end device, then once on another end device.



Continue clicking **Capture/Forward** button until the ICMP ping is completed. You should see the ICMP messages move between the hosts, hub and switch. The PDU **Last Status** should show as **Successful**.

Result:

Thus the mesh topology was created using packet tracer and simulated and studied.

Exp No:

Checking Layer 2 Functionality using Packet Tracer 7.3.0

Aim:

To create bus topology and checking layer 2 functionality by using Packet Tracer 7.3.0.

Requirement:

1.Packet Tracer 7.3.0 Tool

2.PC

3.Switch

4.Connecting Wires

Theory:

About Packet Tracer

Packet Tracer is a network simulation tool designed by Cisco Systems. It is widely used in educational and training settings to help network professionals to develop their skills and knowledge. It allows users to create network topologies by dragging and dropping routers, switches, and other network devices and imitate modern computer networks. It also enables users to experiment with network behaviour, configuration, and troubleshooting.

About Layer 2 of OSI Model

Layer 2 of Open System Interconnection Model is data link layer. The primary responsibility of data link layer is node to node (hop to hop) delivery of data and also ensures error free transmission of data.

Other Responsibilities of data link layer

1) Framing- The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

2)Physical Addressing- If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame.

3)Flow Control- If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

4) Error Control- The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

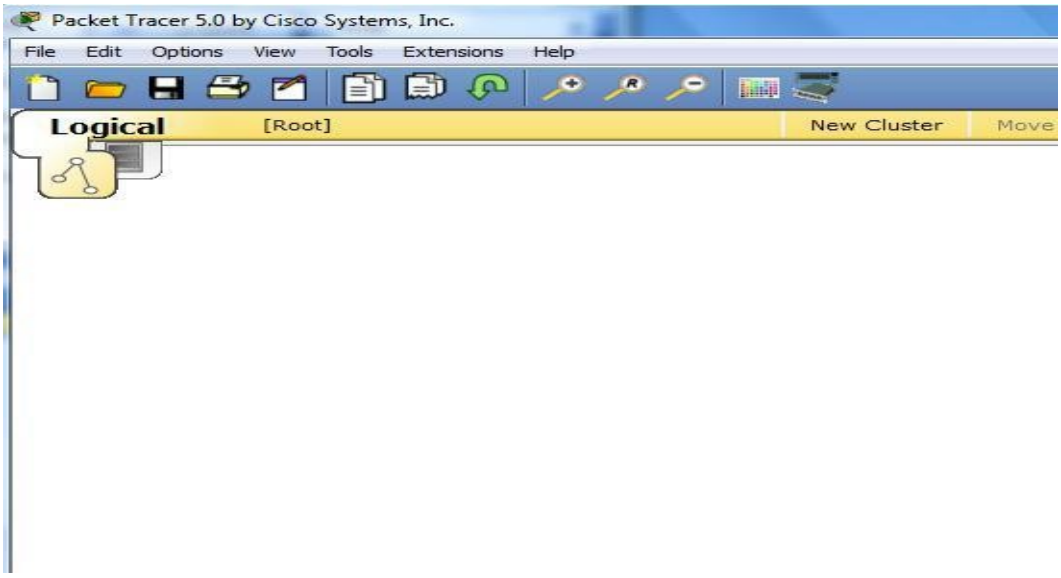
5)Access Control- When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Procedure:

Bus Topology

In this topology, each device has a dedicated point to point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along.

Step 1: Start Packet Tracer



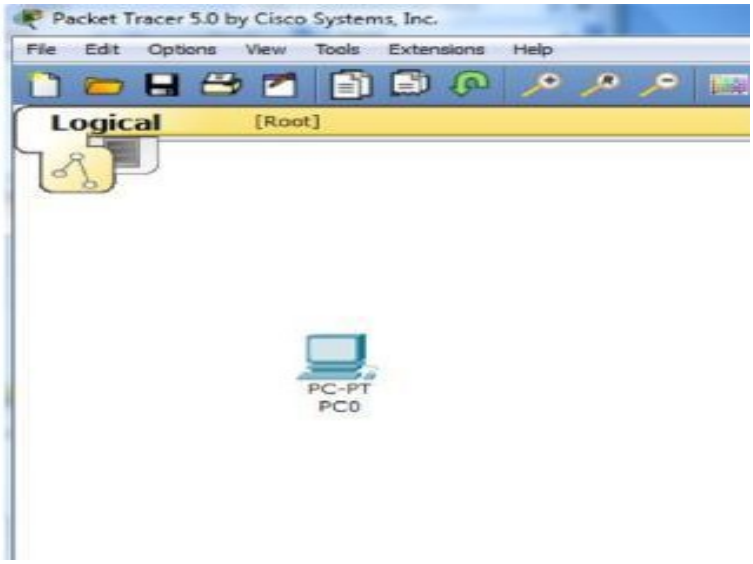
Step 2: Choosing Devices and Connections

Bus topology is created by selecting end devices, network devices and the media in which to connect them.

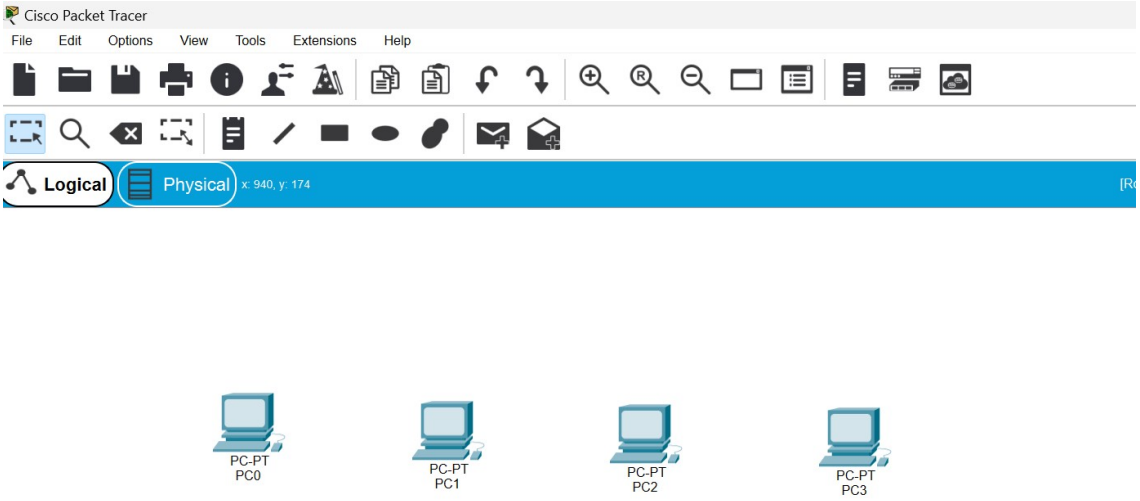
Single click on the **End Devices** and Single click on **Generic End Devices**.

Move the cursor into topology area.We will notice it turns into a plus “+” sign.

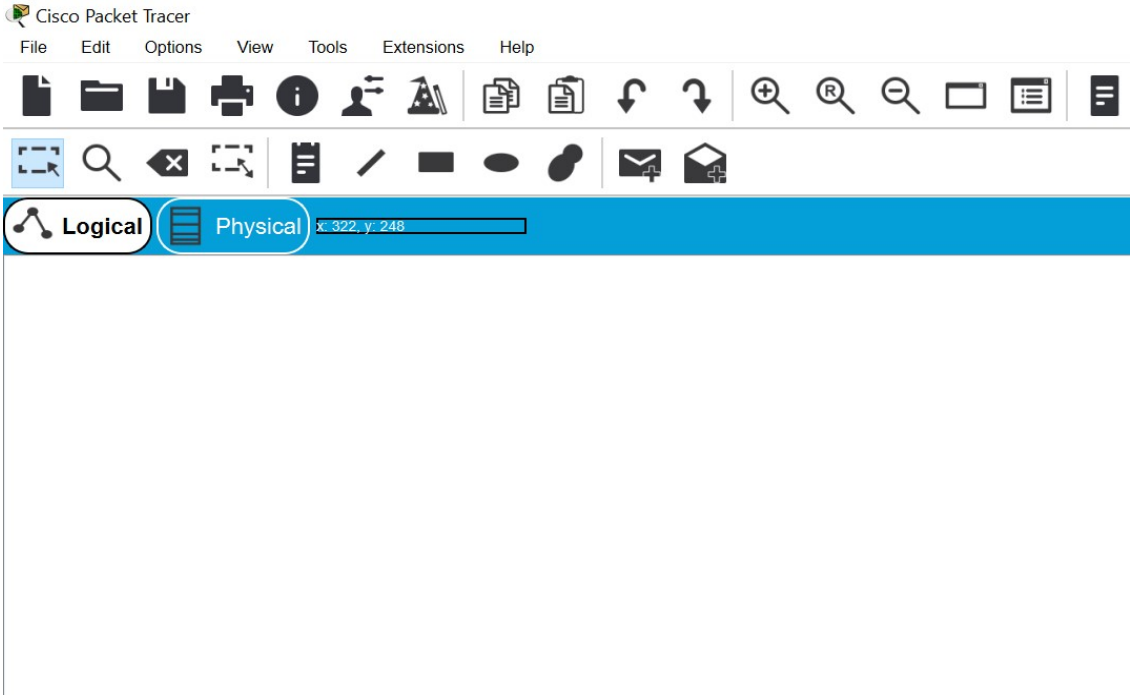
Single click in the topology area and it copies the device.



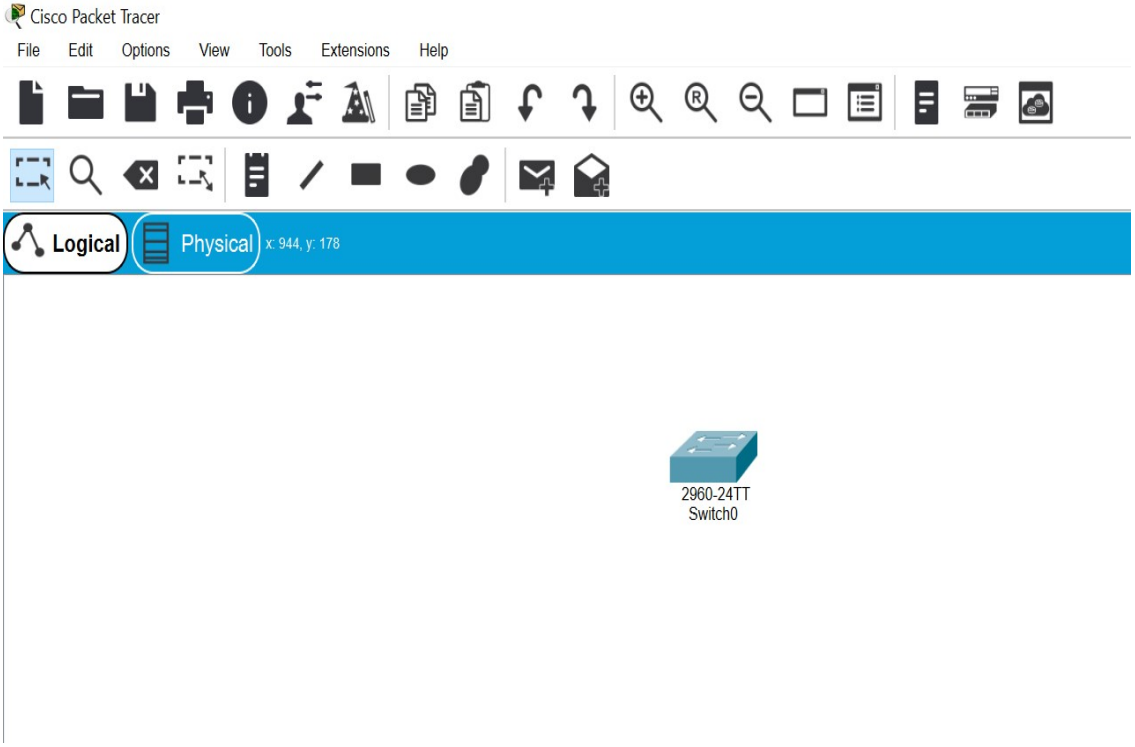
If required, add two or more devices to topology area.



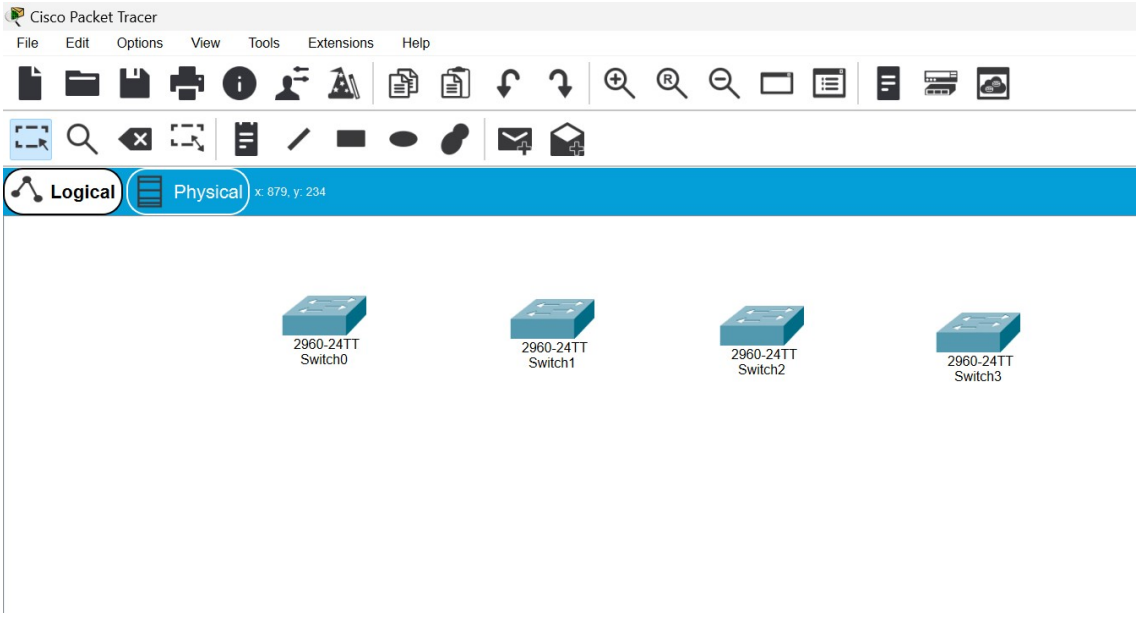
Single click on the **Network Devices(Switches)** and Single click on generic switch.



Move the cursor into topology area. You will notice it turns into a plus "+" sign. Single click in the topology area and it copies the device.

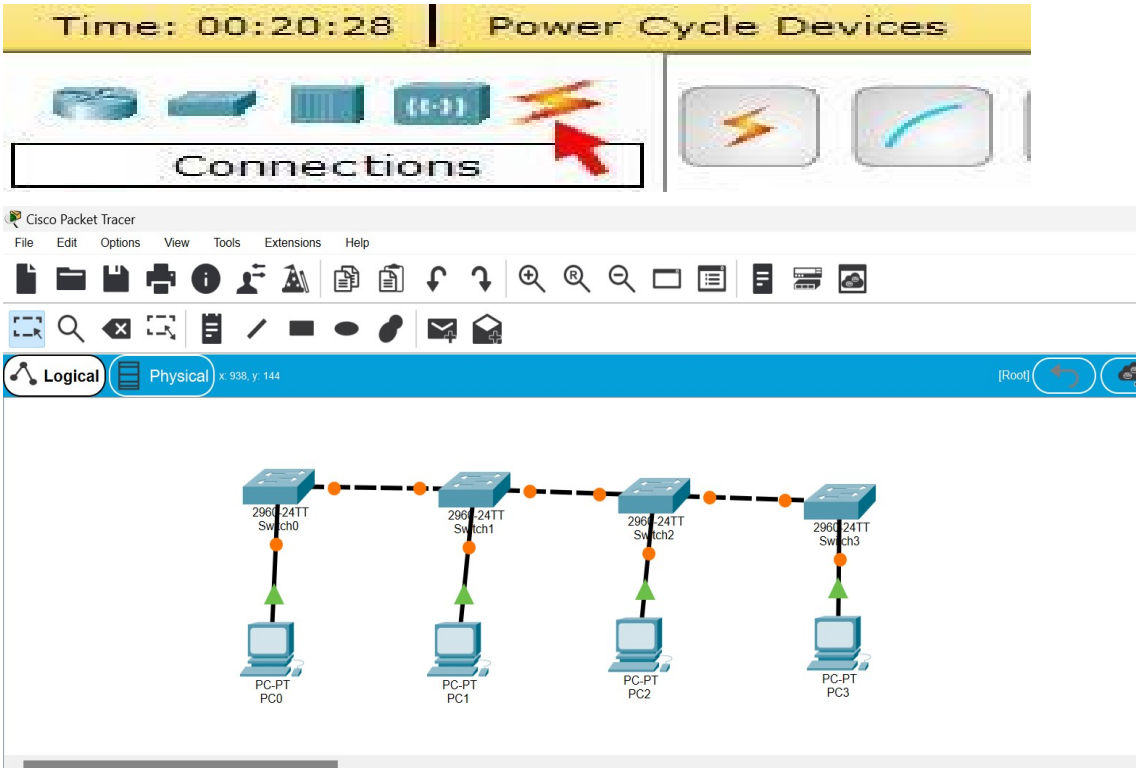


If required, add two or more devices to topology area.



Step 3: Connecting the Hosts to Switches

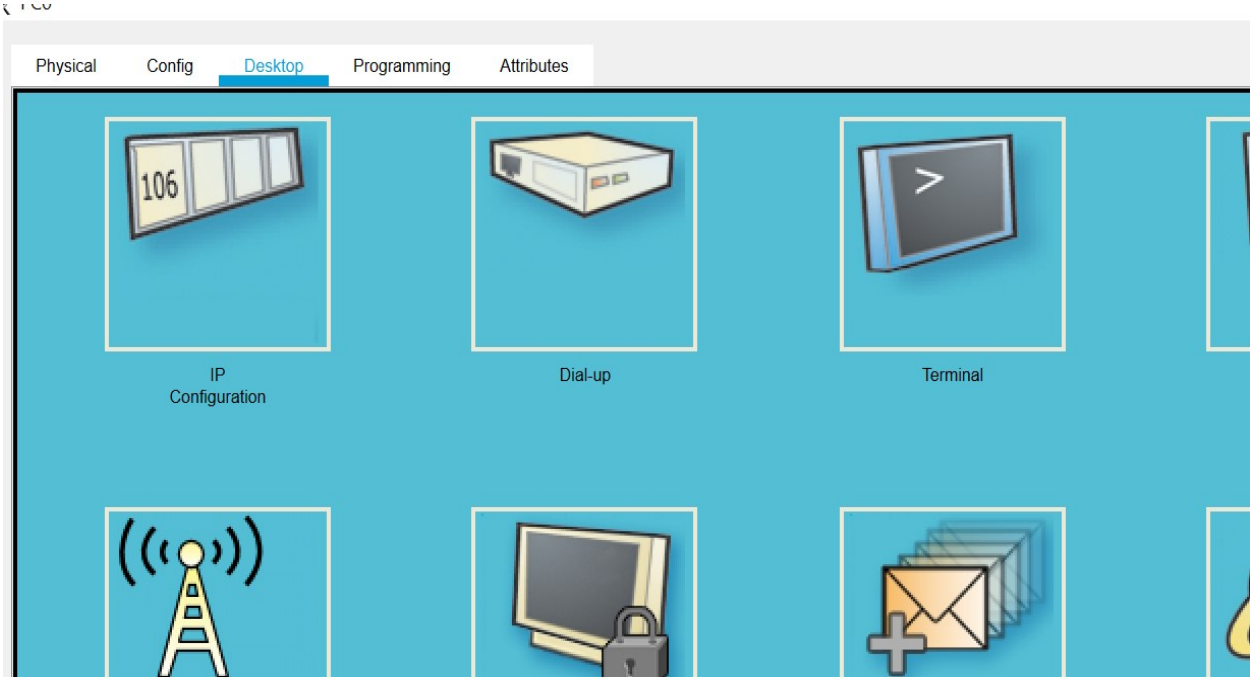
Connect End Devices to Switch by first choosing **Connections (Automatically Choose Connection Type).**

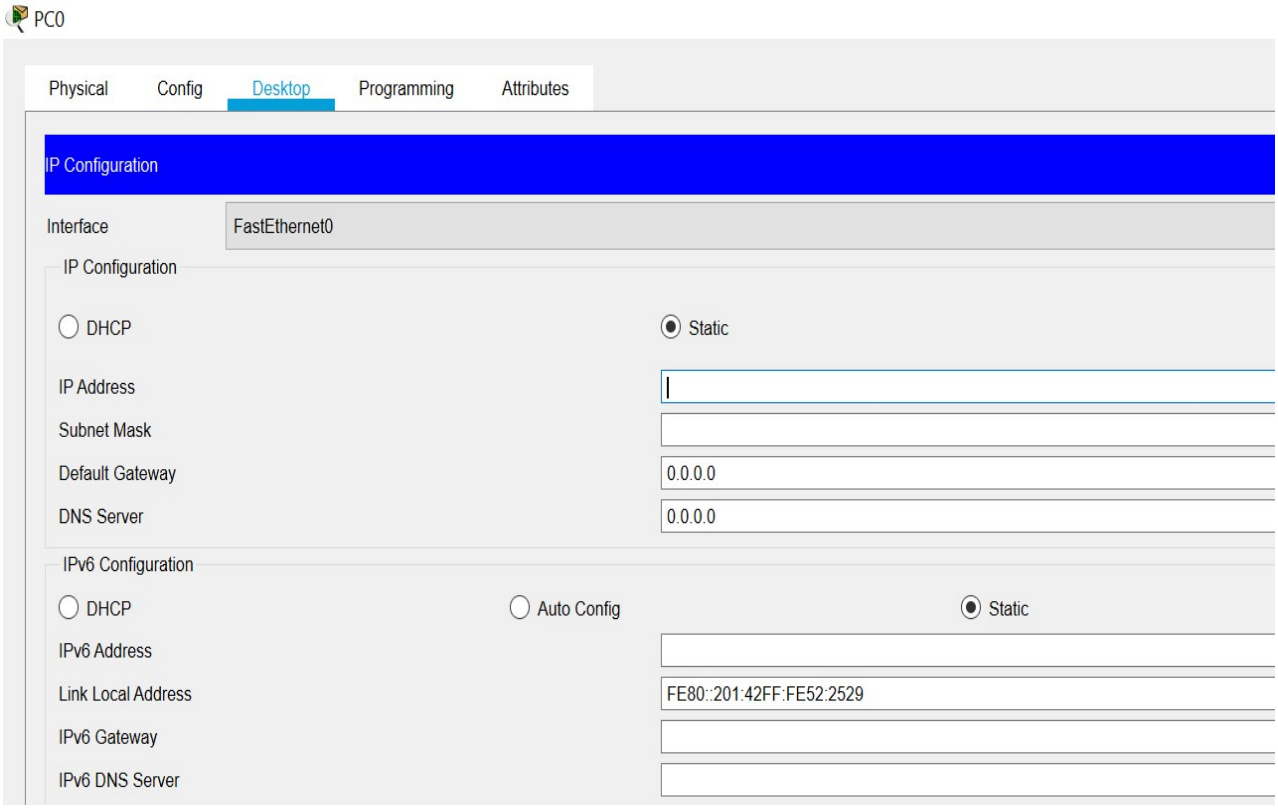


Step 4: Configuring IP Address for all end devices in the topology area

Before we can communicate between the hosts, we need to configure IP Addresses on the devices.

Click on End Device and go to desktop menu, then go to IP Configuration and set the IP Adress.



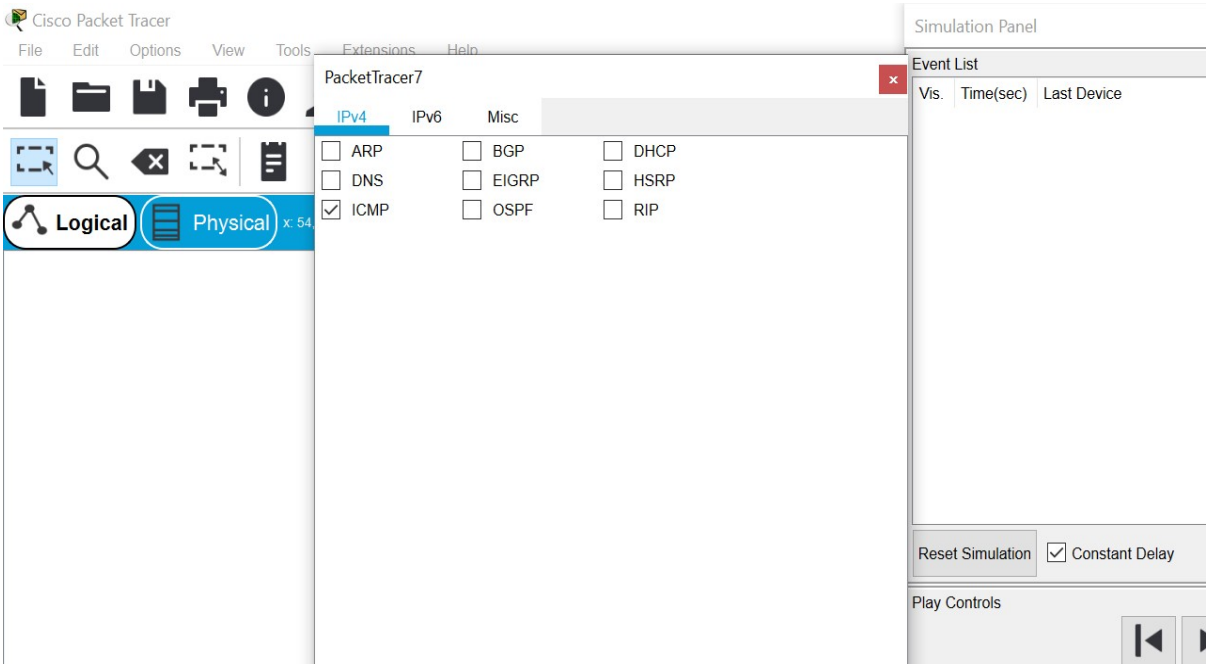


Repeat these steps for the remaining end devices on the network to create IP Address.

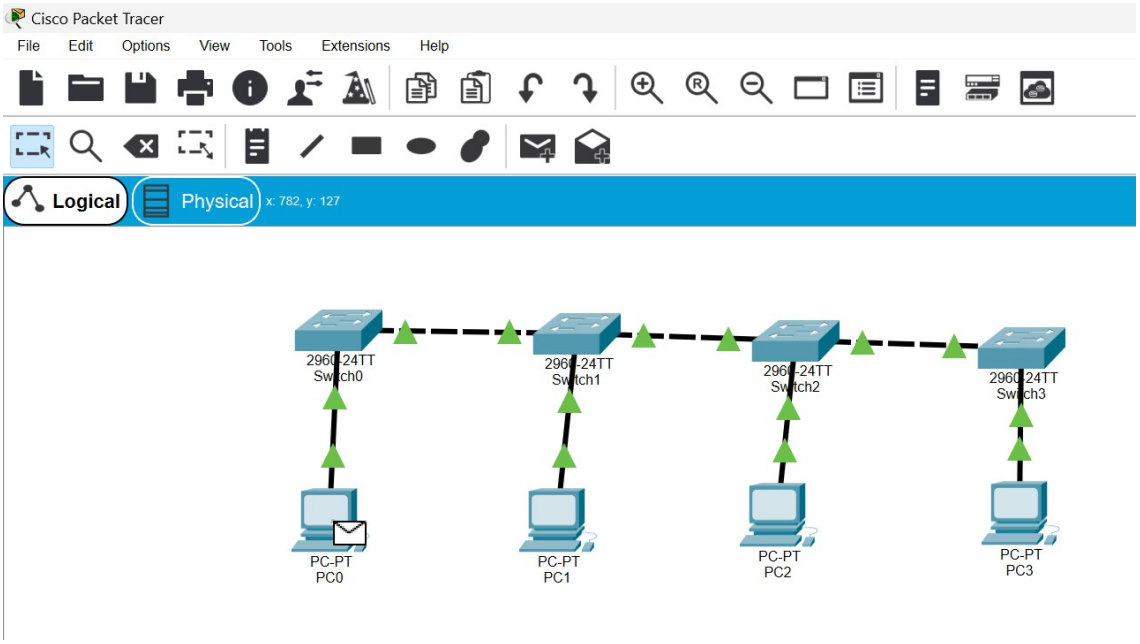
Step 5: Verifying Connectivity in Simulation Mode

Be sure you are in **Simulation** mode. To ensure this, go to view tab in packet tracer and press simulation mode.

Deselect all filters (All/None) and select only **ICMP**.



Select the **Add Simple PDU** tool to ping devices. Click once on one end device, then once on another end device.



Continue clicking **Capture/Forward** button until the ICMP ping is completed. You should see the ICMP messages move between the hosts, hub and switch. The PDU **Last Status** should show as **Successful**.

Result:

Thus the bus topology was created using packet tracer and simulated and studied.

Exp No:

Checking Layer 2 Functionality using Packet Tracer 7.3.0

Aim:

To create star topology and checking layer 2 functionality by using Packet Tracer 7.3.0.

Requirement:

1.Packet Tracer 7.3.0 Tool

2.PC

3.Switch

4.Connecting Wires

Theory:

About Packet Tracer

Packet Tracer is a network simulation tool designed by Cisco Systems. It is widely used in educational and training settings to help network professionals to develop their skills and knowledge. It allows users to create network topologies by dragging and dropping routers, switches, and other network devices and imitate modern computer networks. It also enables users to experiment with network behaviour, configuration, and troubleshooting.

About Layer 2 of OSI Model

Layer 2 of Open System Interconnection Model is data link layer. The primary responsibility of data link layer is node to node (hop to hop) delivery of data and also ensures error free transmission of data.

Other Responsibilities of data link layer

1) Framing- The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

2)Physical Addressing- If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame.

3)Flow Control- If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

4) Error Control- The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

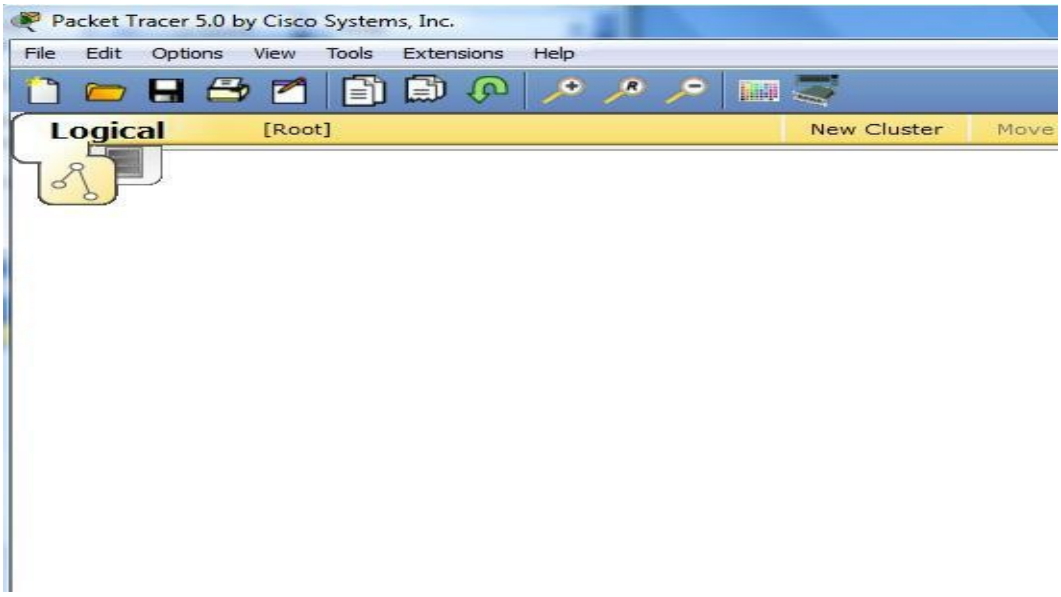
5)Access Control- When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Procedure:

Star Topology

In this topology, each device has a dedicated point to point link only to a central controller, usually called a **hub**. The devices are not directly linked to one another. The controller acts as an exchange: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device. This topology is used in LAN.

Step 1: Start Packet Tracer



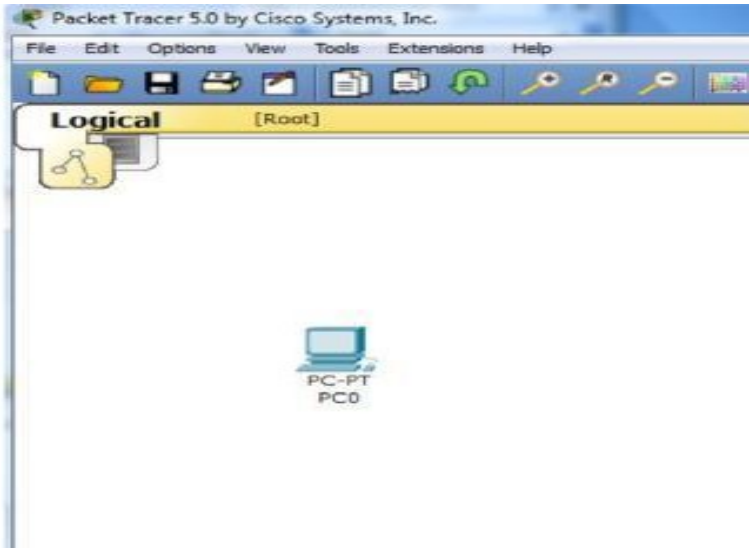
Step 2: Choosing Devices and Connections

Bus topology is created by selecting end devices, network devices and the media in which to connect them.

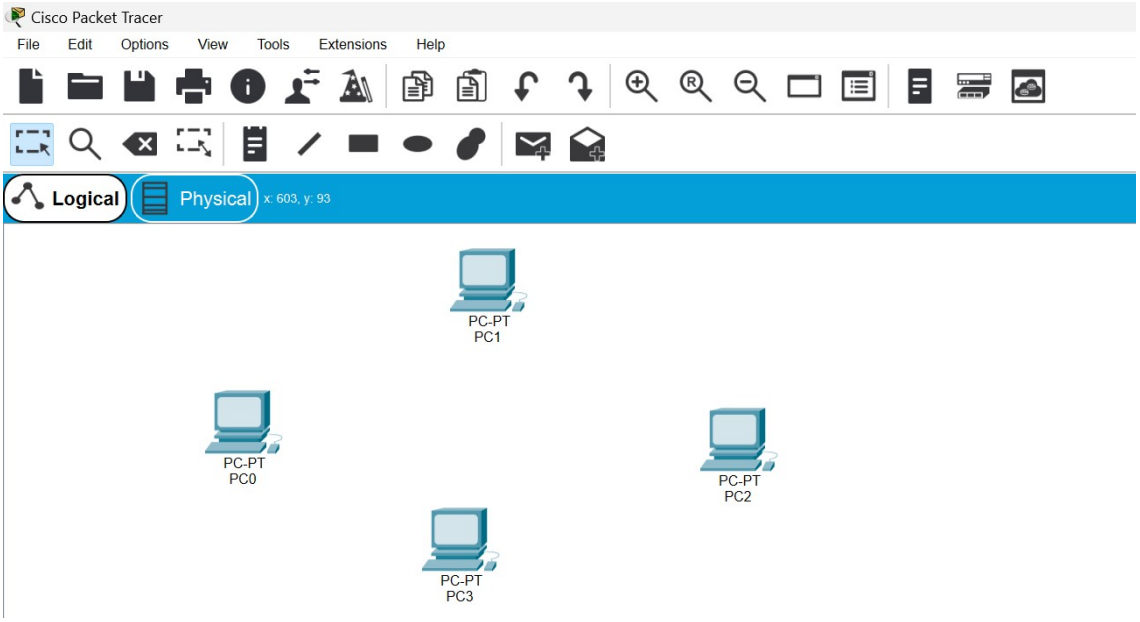
Single click on the **End Devices** and Single click on **Generic End Devices**.

Move the cursor into topology area. We will notice it turns into a plus "+" sign.

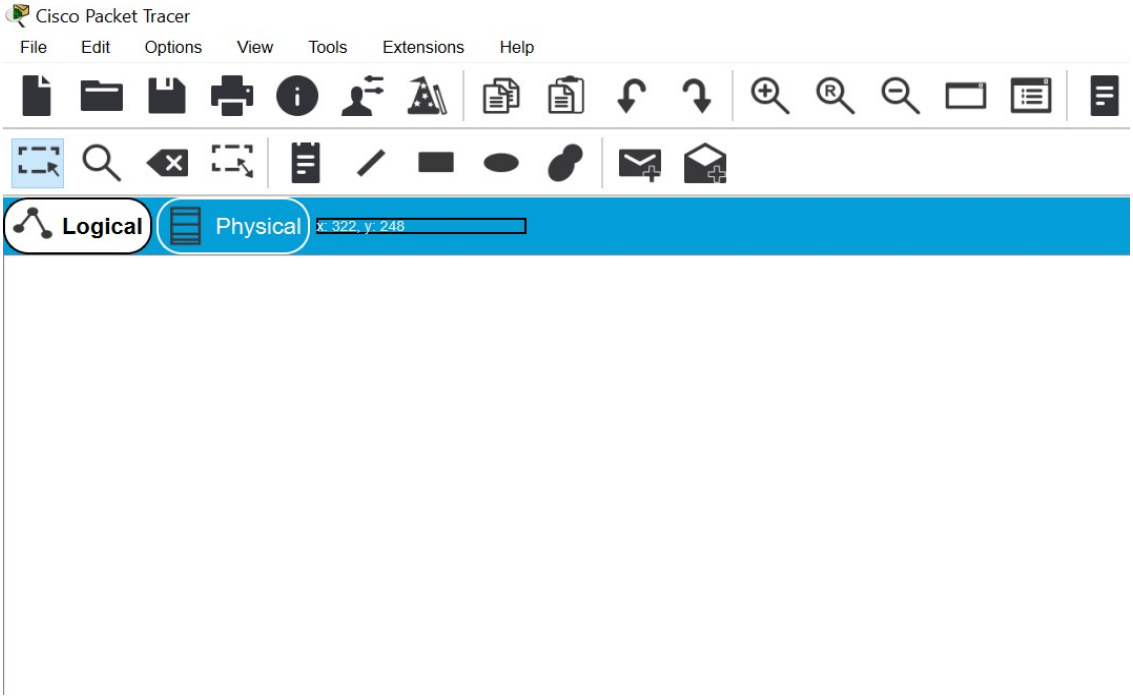
Single click in the topology area and it copies the device.



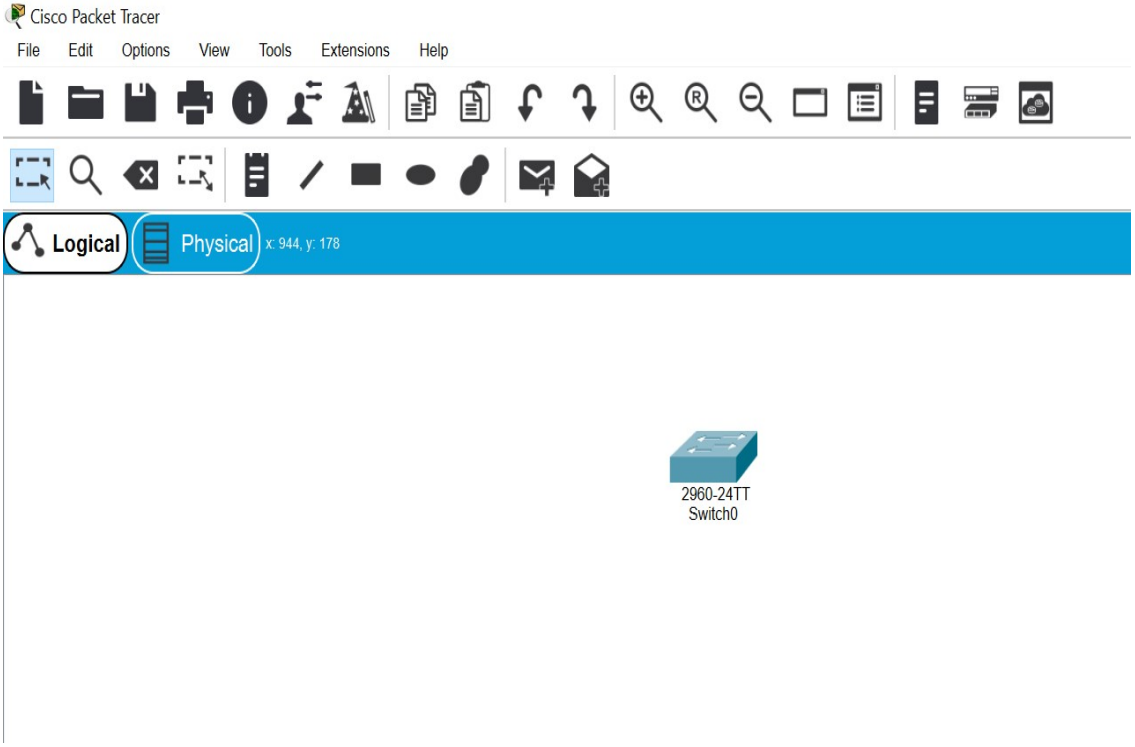
If required, add two or more devices to topology area.



Single click on the **Network Devices(Switches)** and Single click on generic switch.

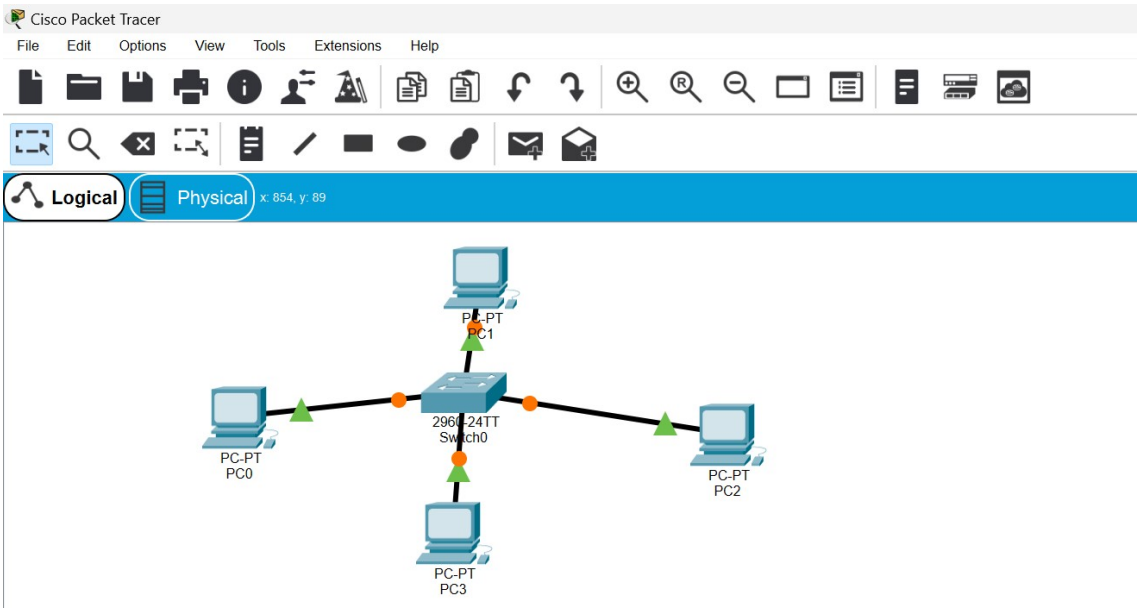


Move the cursor into topology area. You will notice it turns into a plus "+" sign. Single click in the topology area and it copies the device.



Step 3: Connecting the Hosts to Switches

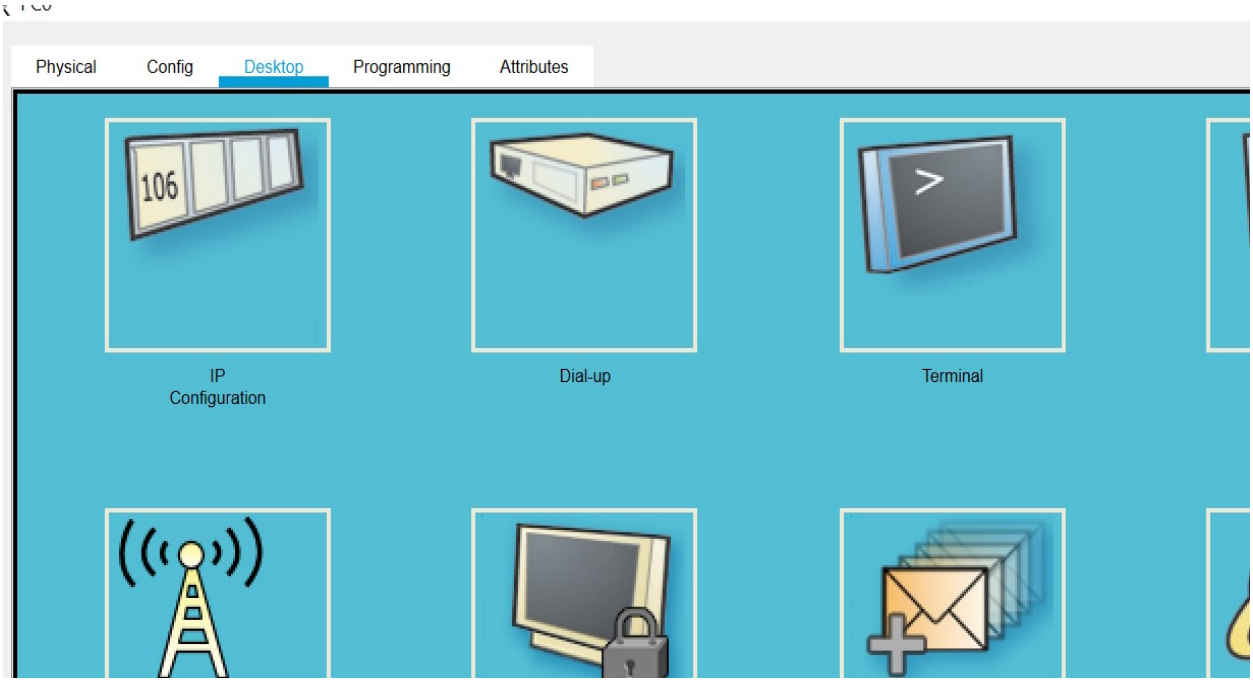
Connect End Devices to Switch by first choosing **Connections (Automatically Choose Connection Type).**

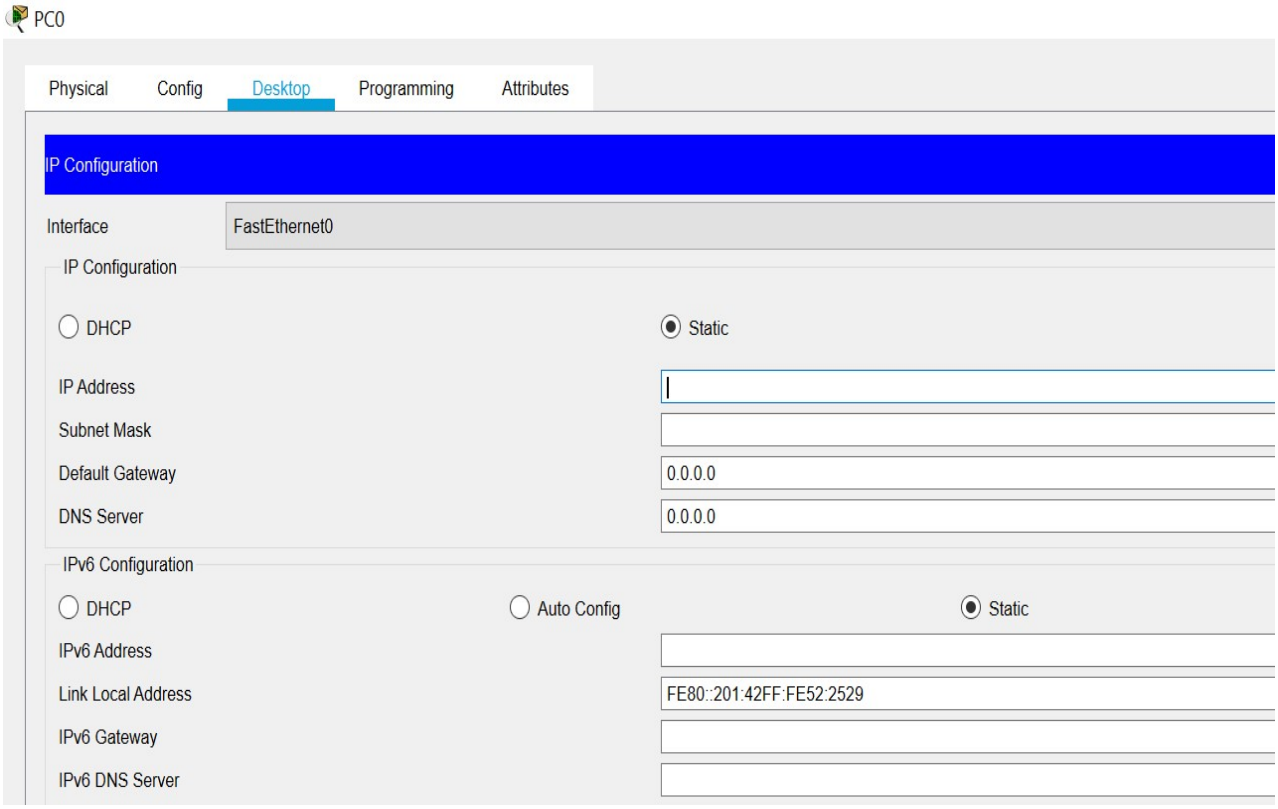


Step 4: Configuring IP Address for all end devices in the topology area

Before we can communicate between the hosts, we need to configure IP Addresses on the devices.

Click on End Device and go to desktop menu, then go to IP Configuration and set the IP Adress.



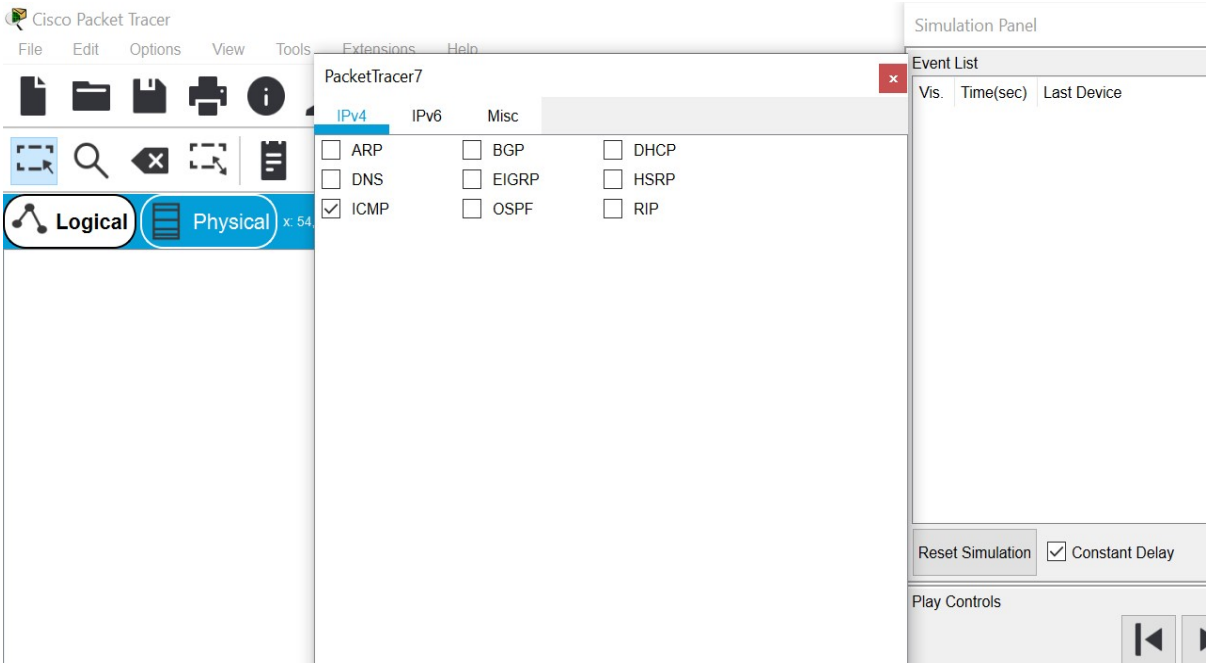


Repeat these steps for the remaining end devices on the network to create IP Address.

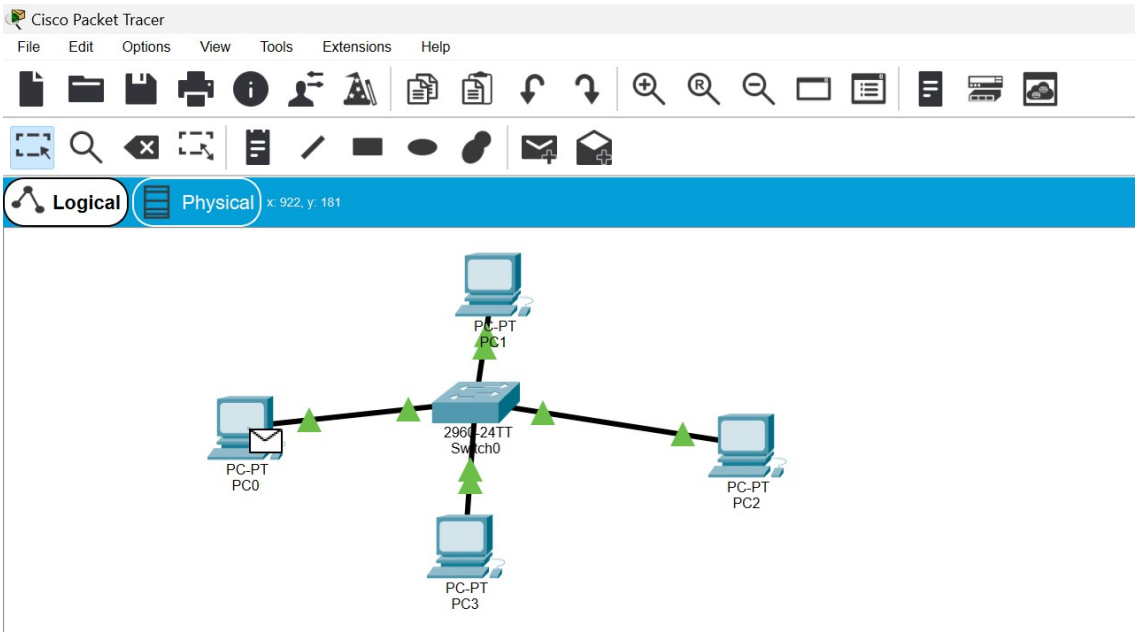
Step 5: Verifying Connectivity in Simulation Mode

Be sure you are in **Simulation** mode. To ensure this, go to view tab in packet tracer and press simulation mode.

Deselect all filters (All/None) and select only **ICMP**.



Select the **Add Simple PDU** tool to ping devices. Click once on one end device, then once on another end device.



Continue clicking **Capture/Forward** button until the ICMP ping is completed. You should see the ICMP messages move between the hosts, hub and switch. The PDU **Last Status** should show as **Successful**.

Result:

Thus the star topology was created using packet tracer and simulated and studied.

Exp No:

Configure ARP and MAC Table

Aim:

To configure ARP and MAC Table in star topology by using Packet Tracer 7.3.0.

Requirement:

1.Packet Tracer 7.3.0 Tool

2.PC

3.Switch

4.Connecting Wires

Theory:

About ARP Table:

ARP stands for **Address Resolution Protocol** which is one of the most important protocols of the network layer in the OSI model. ARP is a communication mechanism that is used to translate a logical address, such as an IP address, to a physical (MAC) address on a local network.

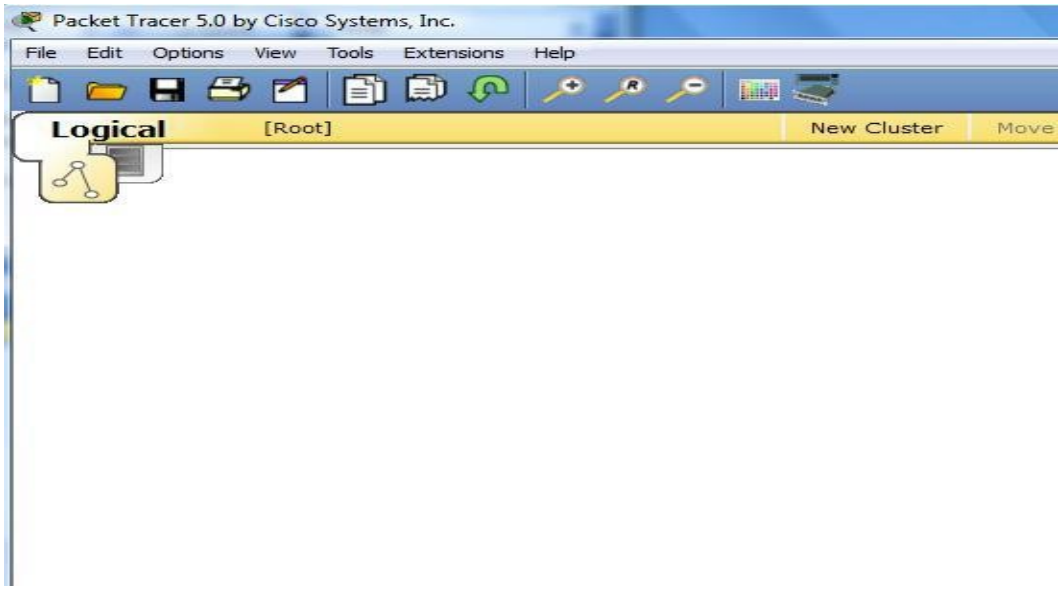
Procedure:

To create Star Topology and Configure ARP Table

Star Topology:

In this topology, each device has a dedicated point to point link only to a central controller. The devices are not directly linked to one another. The controller acts as an exchange: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device. Star Topology is used in Local Area Network.

Step 1: Start Packet Tracer



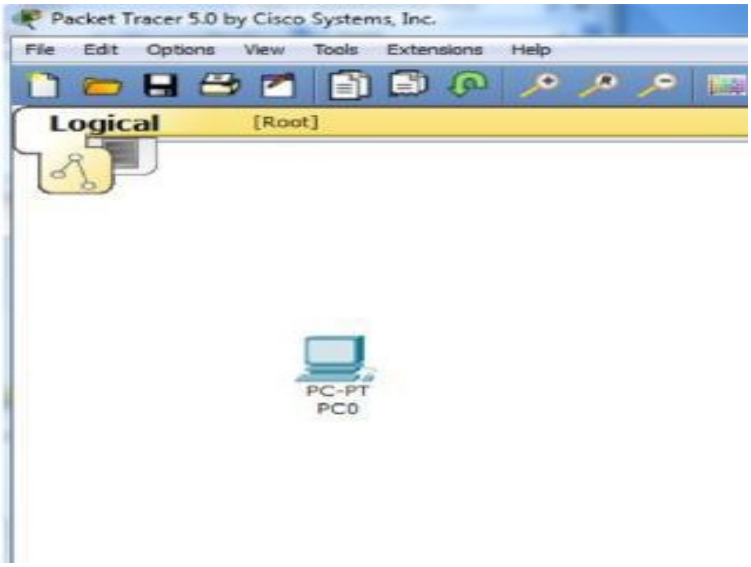
Step 2: Choosing Devices and Connections

Star topology is created by selecting end devices, network devices and the media in which to connect them.

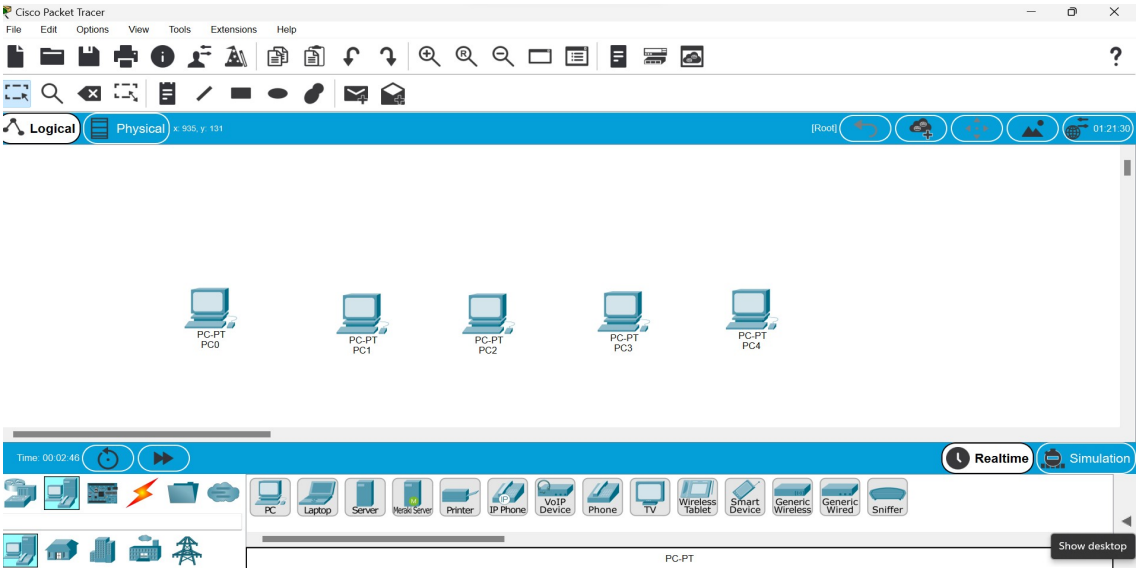
Single click on the **End Devices** and Single click on **Generic End Devices**.

Move the cursor into topology area.We will notice it turns into a plus “+” sign.

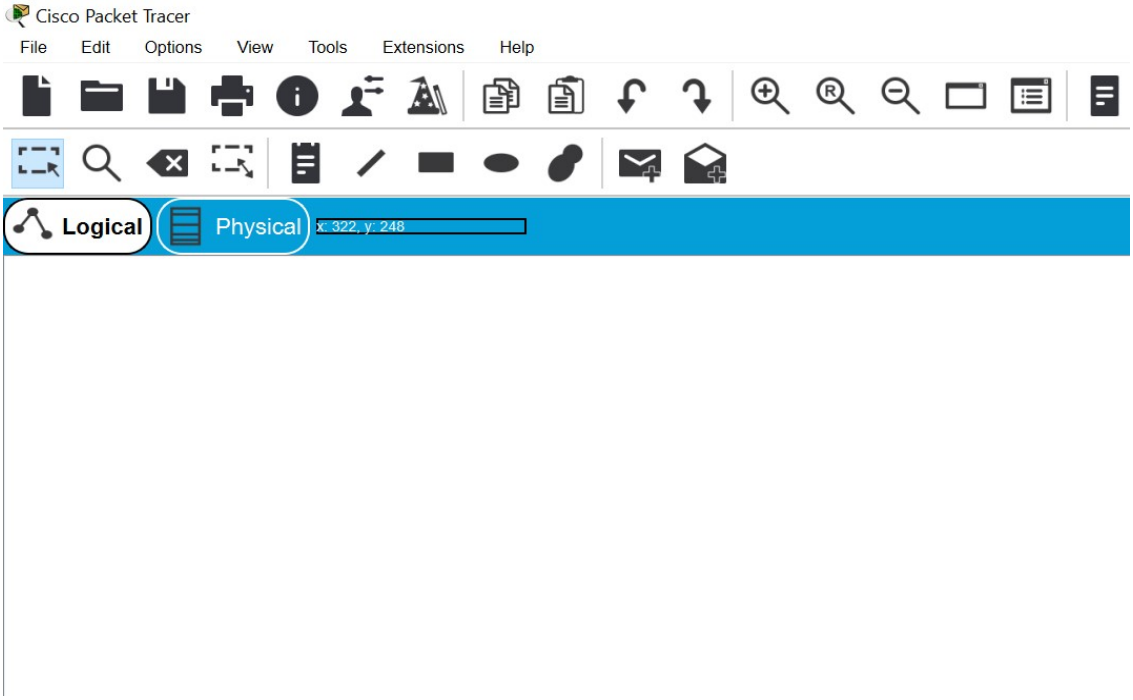
Single click in the topology area and it copies the device.



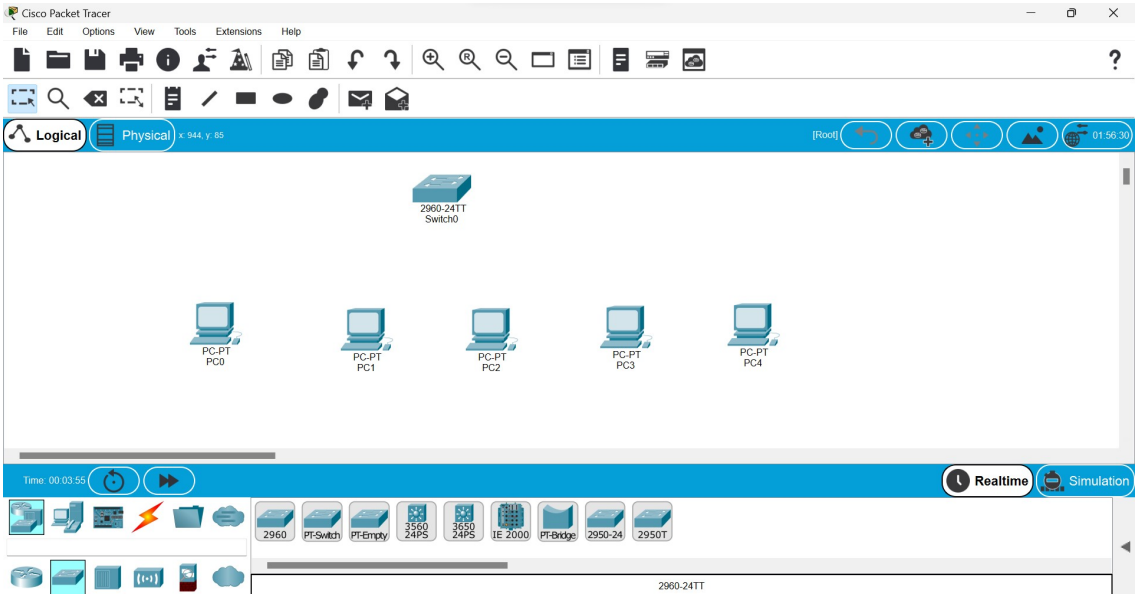
If required, add two or more devices to topology area.



Single click on the **Network Devices (Switches)** and Single click on generic switch.

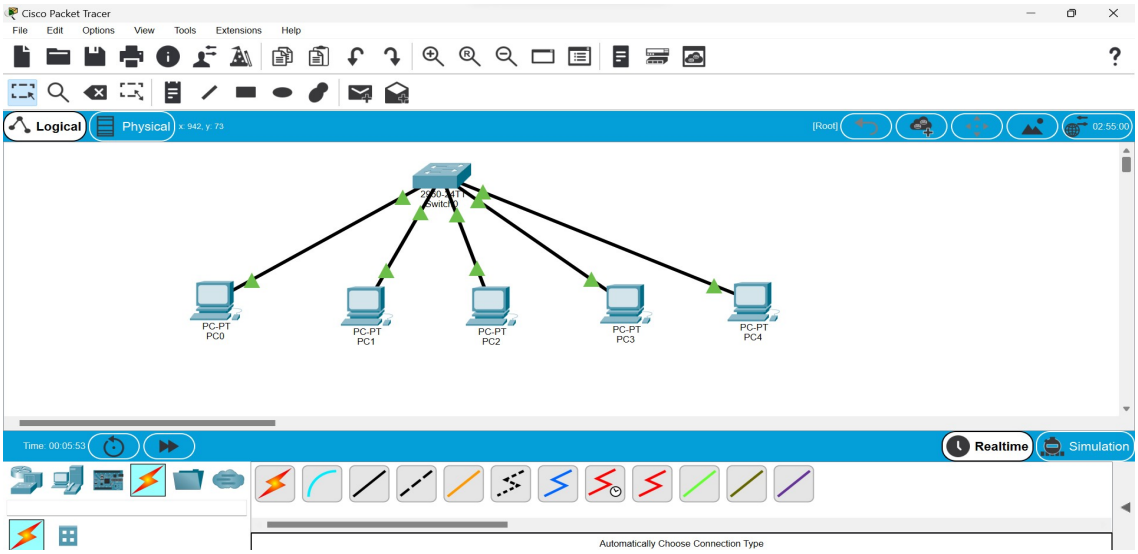


Move the cursor into topology area. You will notice it turns into a plus “+” sign. Single click in the topology area and it copies the device.



Step 3: Connecting the Hosts to Switches

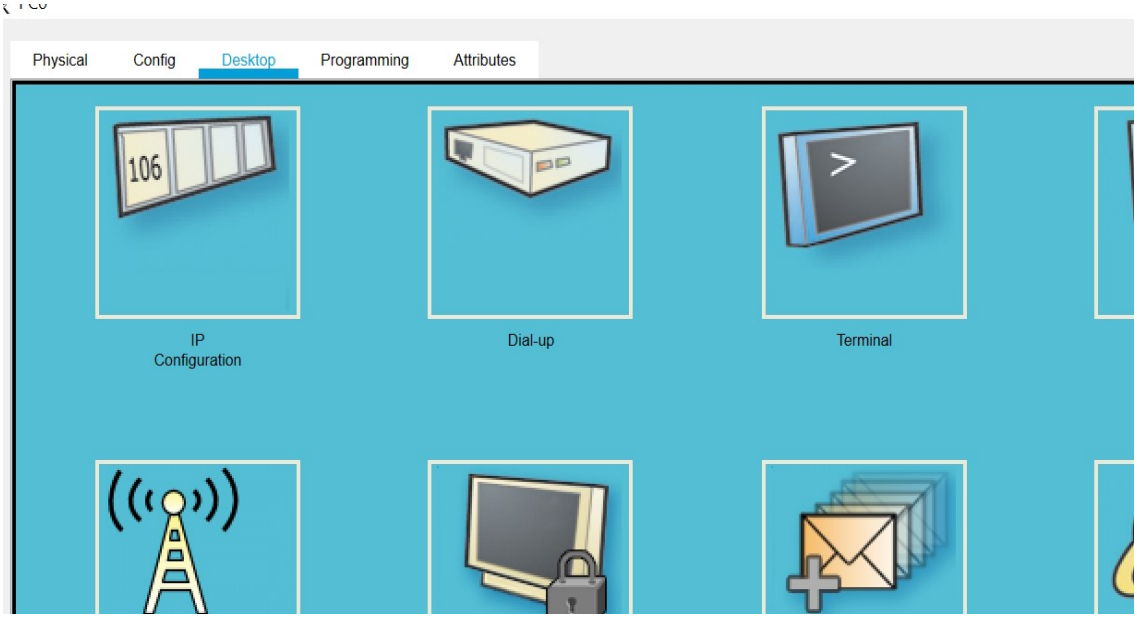
Connect End Devices to Switch by first choosing **Connections (Automatically Choose Connection Type).**

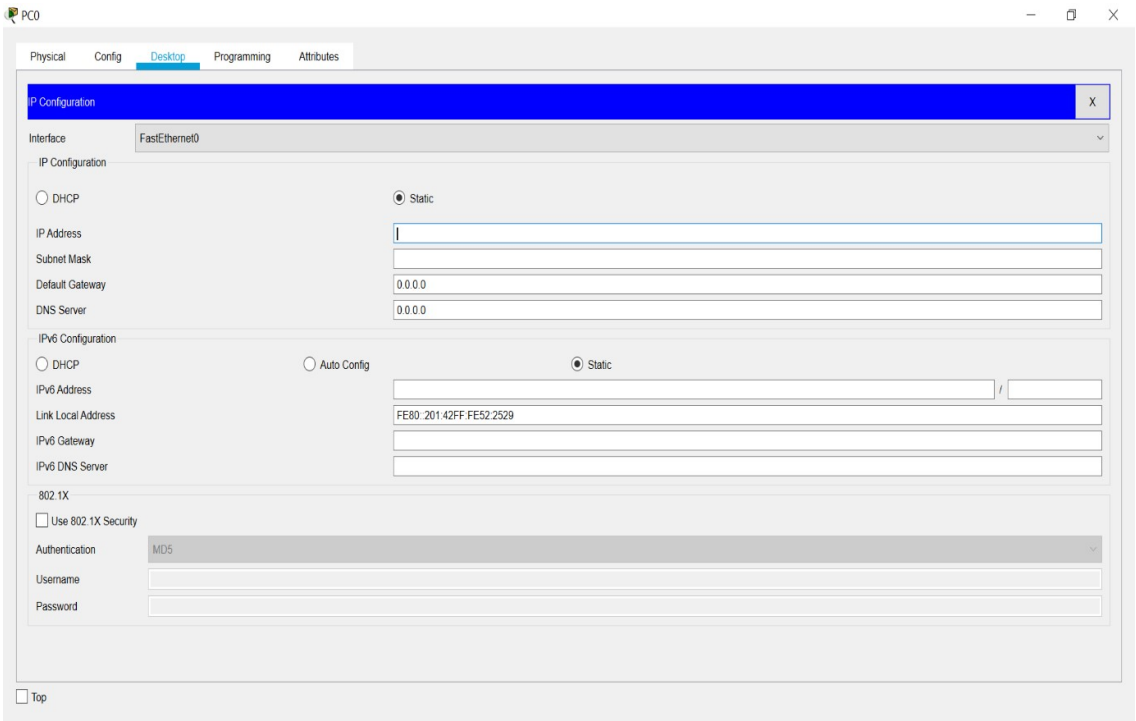


Step 4: Configuring IP Address for all end devices in the topology area

Before we can communicate between the hosts, we need to configure IP Addresses on the devices.

Click on End Device and go to desktop menu, then go to IP Configuration and set the IP Adress.



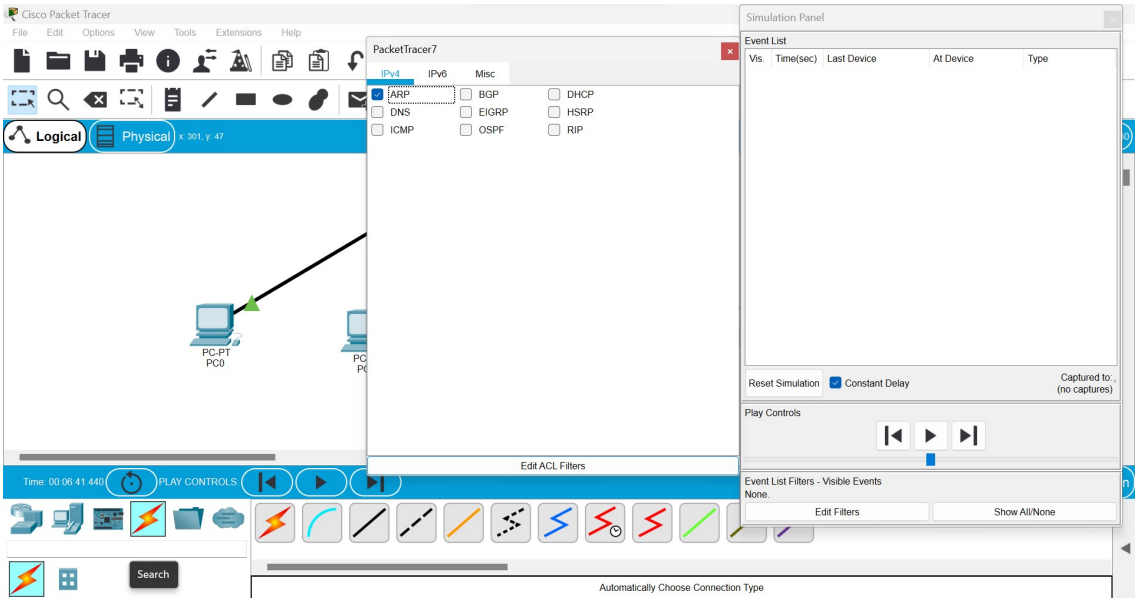


Repeat these steps for the remaining end devices on the network to create IP Address.

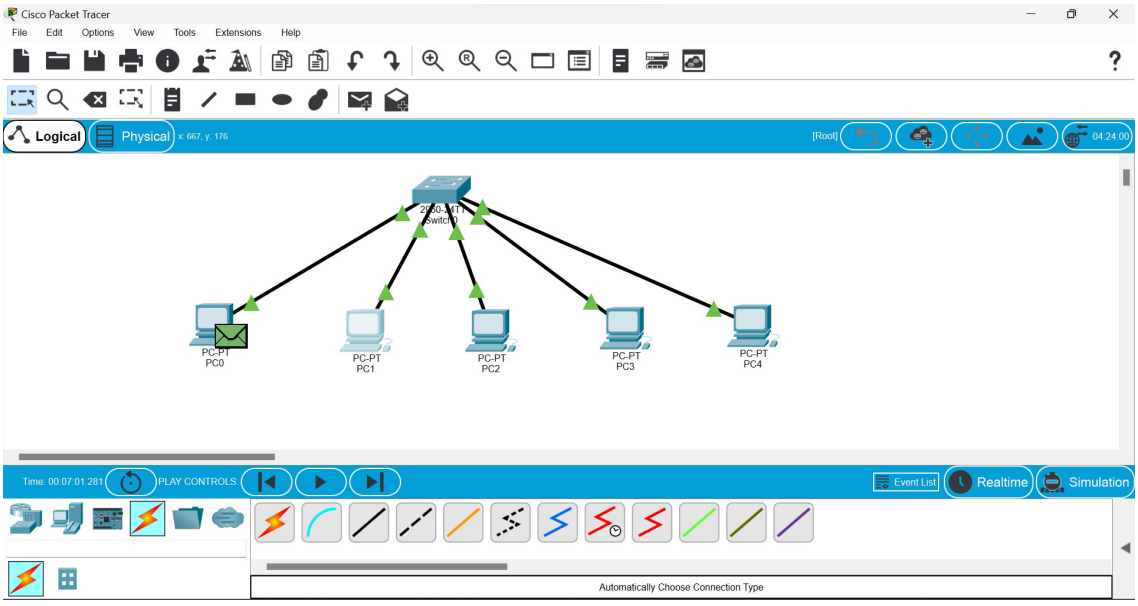
Step 5: Verifying Connectivity in Simulation Mode

Be sure you are in **Simulation** mode. To ensure this, go to view tab in packet tracer and press simulation mode.

Deselect all filters (All/None) and select only **ARP**.



Select the **Add Simple PDU** tool to ping devices. Click once on one end device, then once on another end device.



About MAC Table

MAC stands for Media Access Control which is a physical address that uniquely identifies a device. The switch maintains an address table called the MAC address table in order to efficiently switch frames between interfaces. So basically, a switch stores information about the other (Ethernet interfaces) to which it is connected on a network. When a switch receives a frame, it associates the MAC address of the sending device with the switch port on which it was received.

Procedure to configure MAC Table

Step 1: Create Star Topology

Step 2: Enable the switch

Step 3: Ping the device from another device.

Step 4: Type show mac-address-table in CLI.

Result:

Thus ARP and MAC table was configured successfully using packet tracer tool.

Exp No:

Configure Spanning Tree Protocol

Aim:

To configure spanning tree protocol by using Packet Tracer 7.3.0.

Requirement:

1.Packet Tracer 7.3.0 Tool

2.Switch

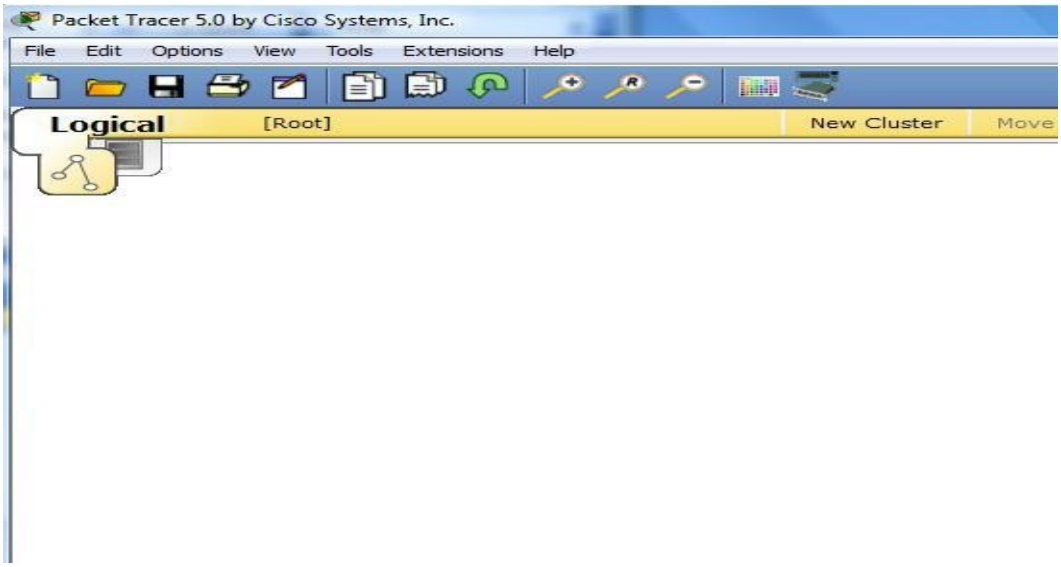
3.Connecting Wires

Theory:

Spanning Tree Protocol (STP) is a layer 2 protocol. STP automatically removes layer 2 switching loops by shutting down the redundant links(ie Is an additional link between two switches). If the flow of traffic is not carefully monitored and controlled, the data can be caught in a loop that circles around network segments, affecting performance and bringing traffic to a near halt.

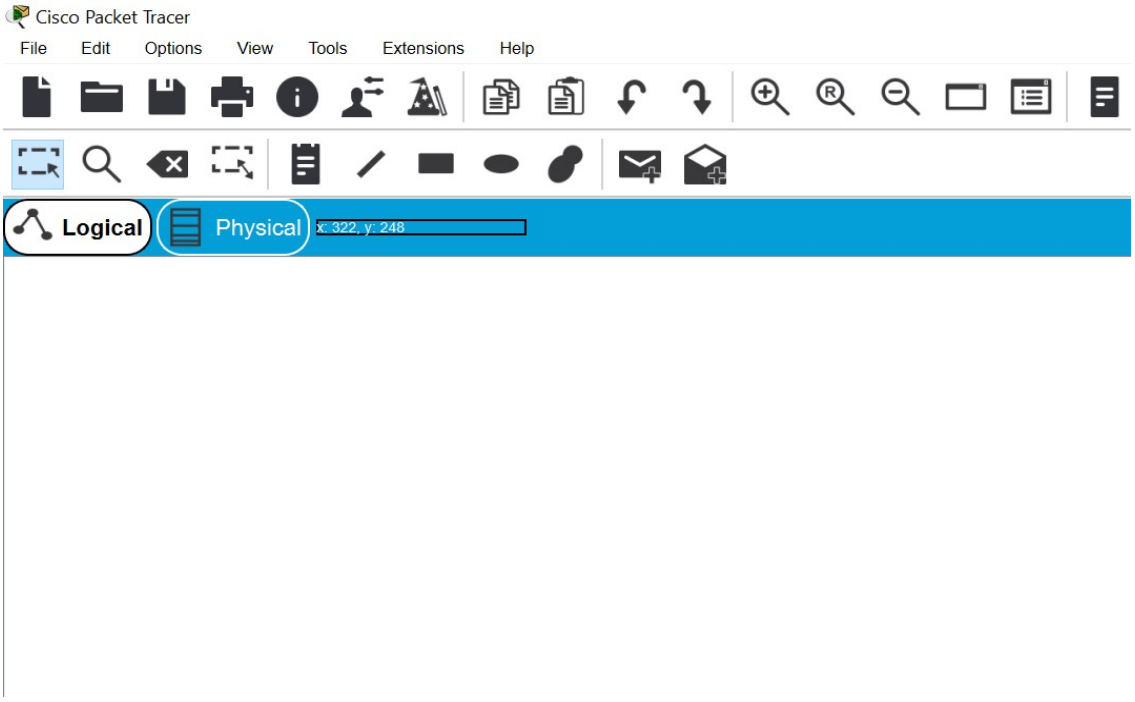
Procedure:

Step 1: Start Packet Tracer

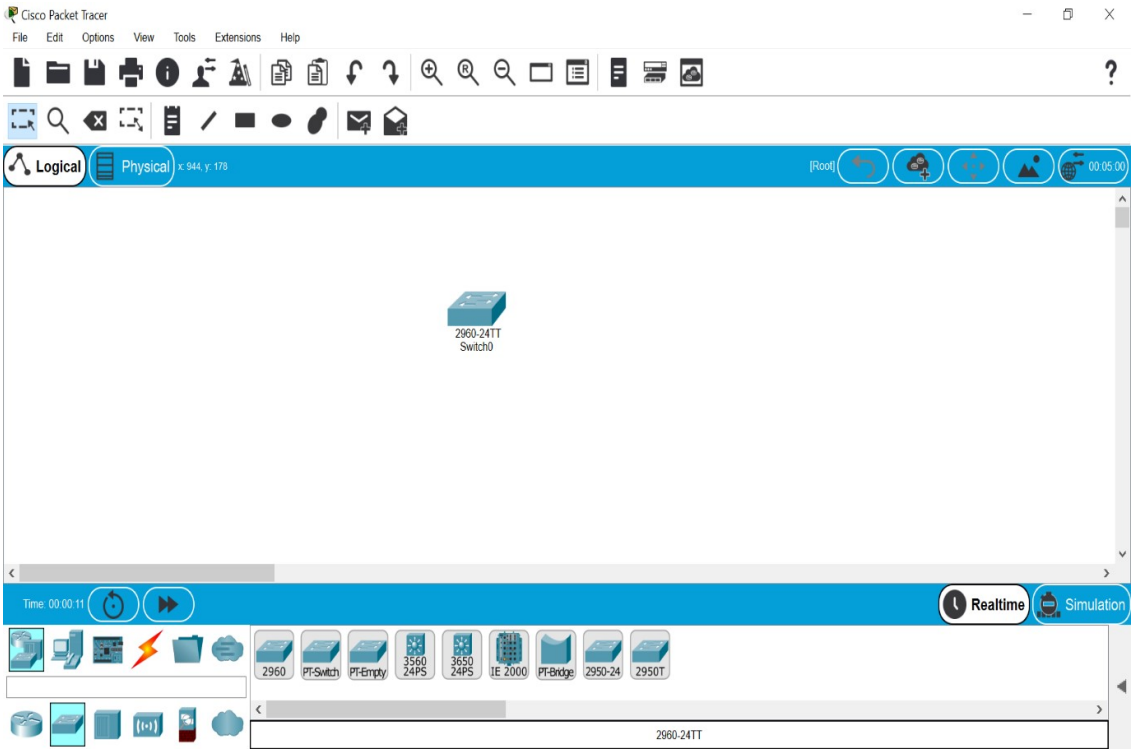


Step 2: Choosing Devices and Connections

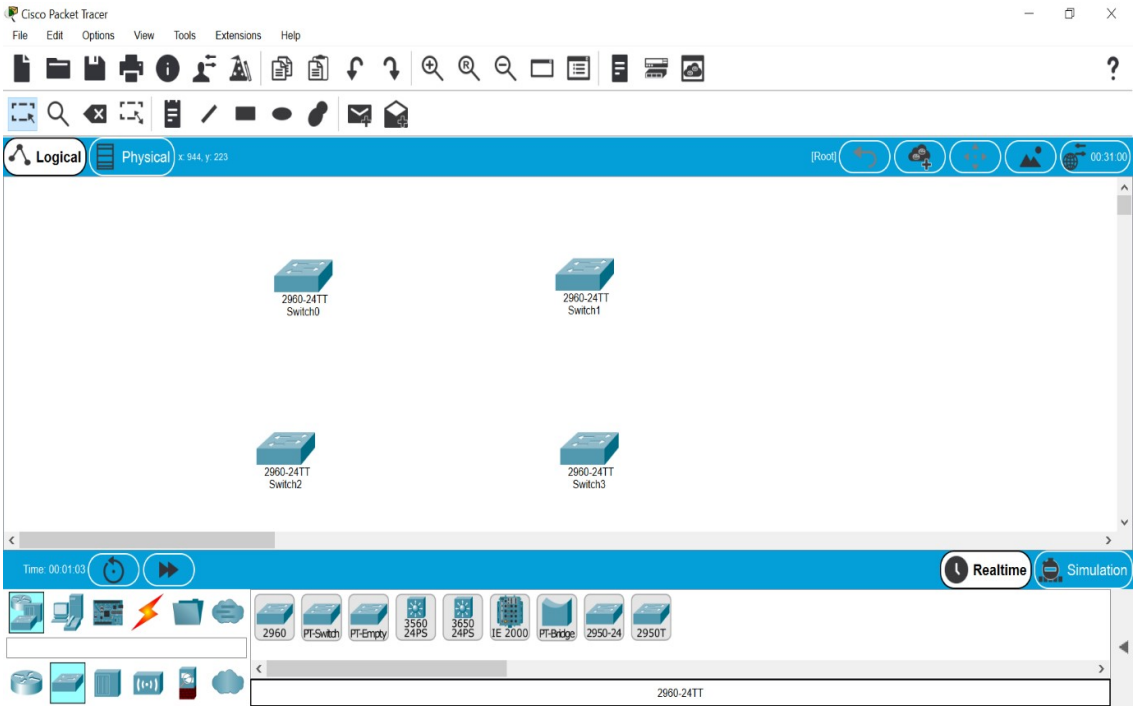
Single click on the **Network Devices(Switches)** and Single click on generic switch.



Move the cursor into topology area. You will notice it turns into a plus "+" sign. Single click in the topology area and it copies the device.

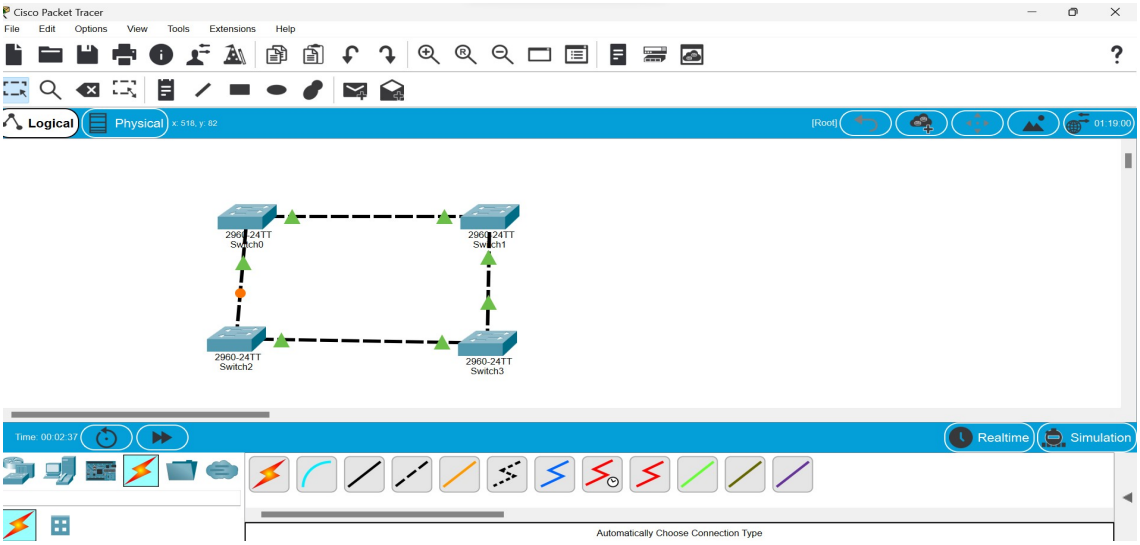


If required, add two or more switches to topology area.



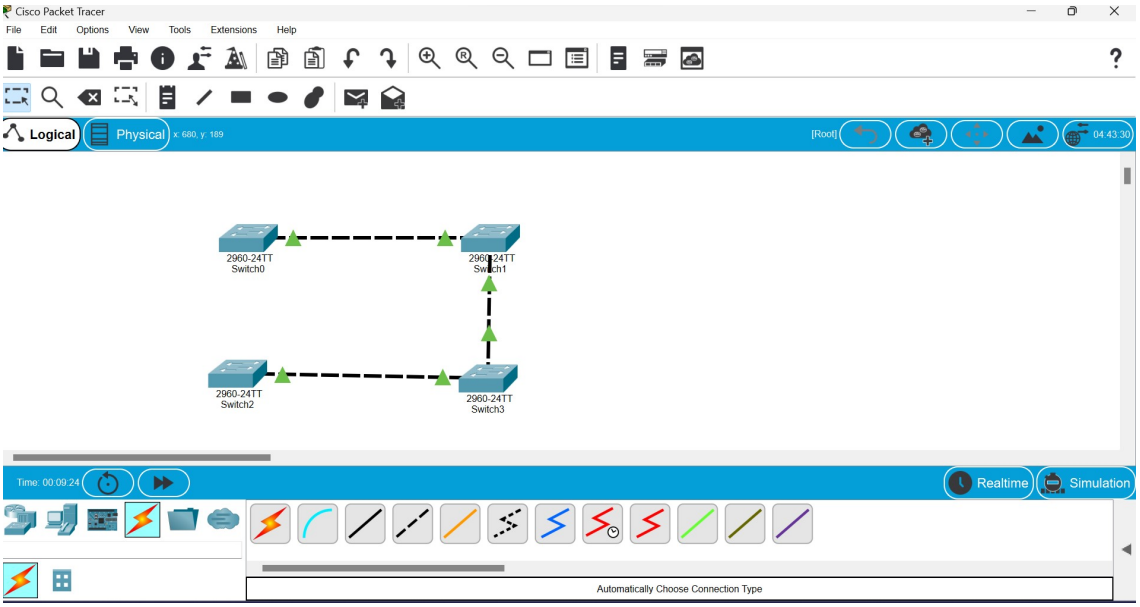
Step 3: Connecting the Hosts to Switches

Connect End Devices to Switch by first choosing **Connections (Automatically Choose Connection Type).**



After connecting the switches together in the loop position, one of the ports become **blocking**. Because by default STP is **enabled** and it is avoiding any **switching loop.**

The STP blocks one of the port of switch2. This election is done according to the **cost to the root**. The **designated ports** are selected and the remaining **non-designated port** on a segment is blocked. Therefore, **only one** designated port can exist in a segment.



Result:

Thus spanning tree protocol was configured successfully using packet tracer tool.

Exp No:

Checking Layer 3 Functionality using Packet Tracer 7.3.0

Aim:

To check layer 3(Network Layer) of OSI model by using Packet Tracer 7.3.0.

Requirement:

1.Packet Tracer 7.3.0 Tool

2.Switch

3.Router

4.PC's

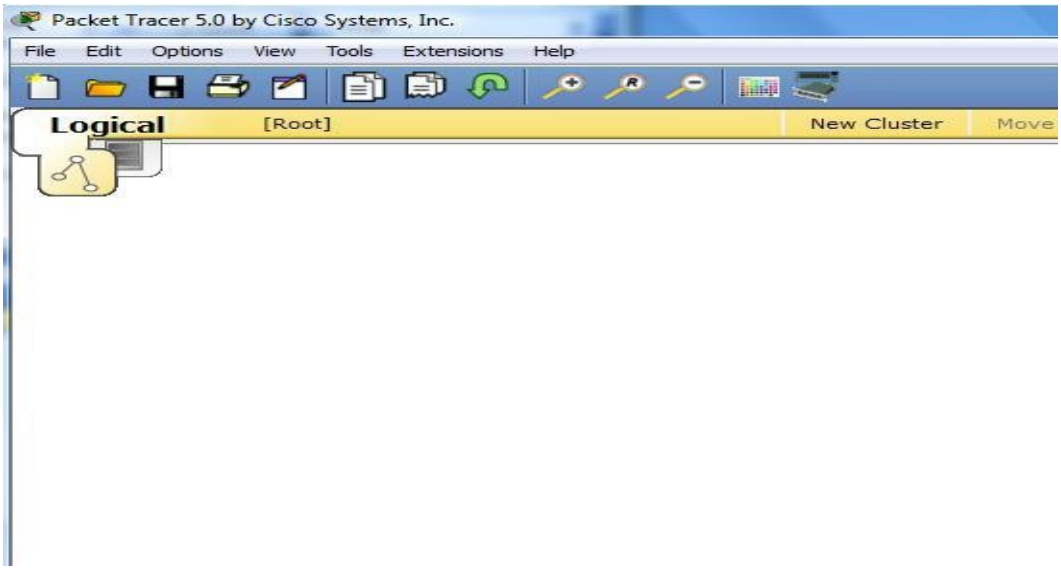
5.Connecting Wires

Theory:

Layer 3 refers to the third layer of the Open Systems Interconnection (OSI) Model, which is the network layer. Layer 3 is positioned between the transport layer and the data link layer. It responds to requests from the transport layer and subsequently issues requests to the data link layer. **Network Layer** provides the functional and procedural means of transferring variable length data sequences from a source host on one network to a destination host on a different network. Routers operate at the third layer as it handles the routing of data. In addition to this, network layer also perform fragmentation and reassembly, and report delivery errors. The major protocols included in the Network layer are Internet Protocol (IPv4 or IPv6), Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP) and Internet Group Management Protocol (IGMP).

Procedure:

Step 1: Start Packet Tracer

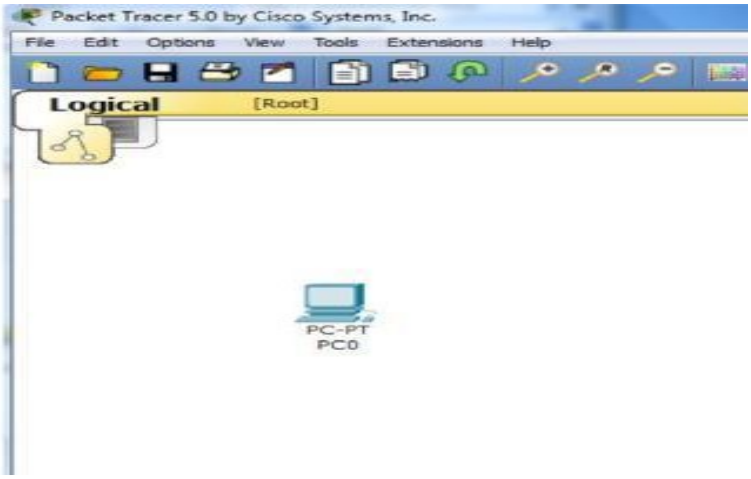


Step 2: Choosing Devices and Connections

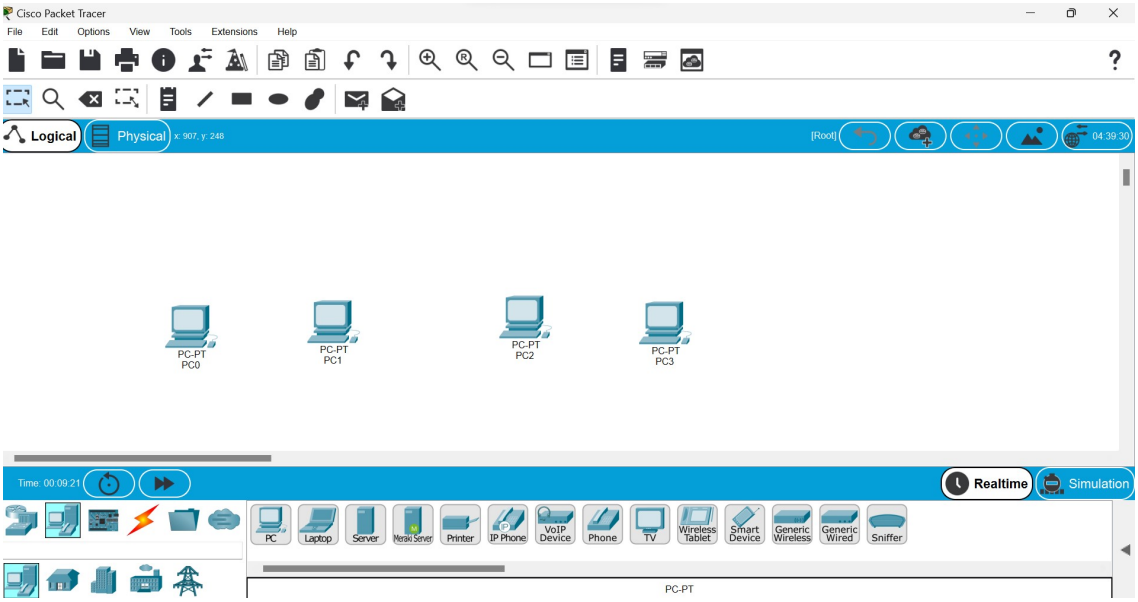
Single click on the **End Devices** and Single click on **Generic End Devices**.

Move the cursor into topology area. We will notice it turns into a plus "+" sign.

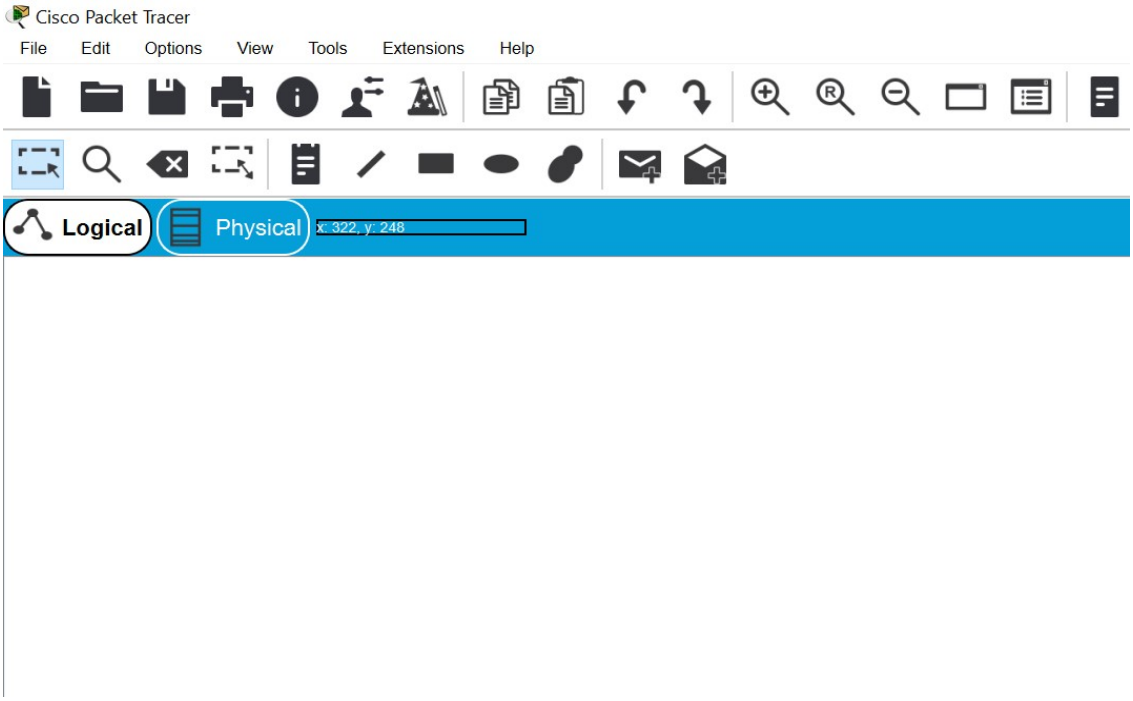
Single click in the topology area and it copies the device.



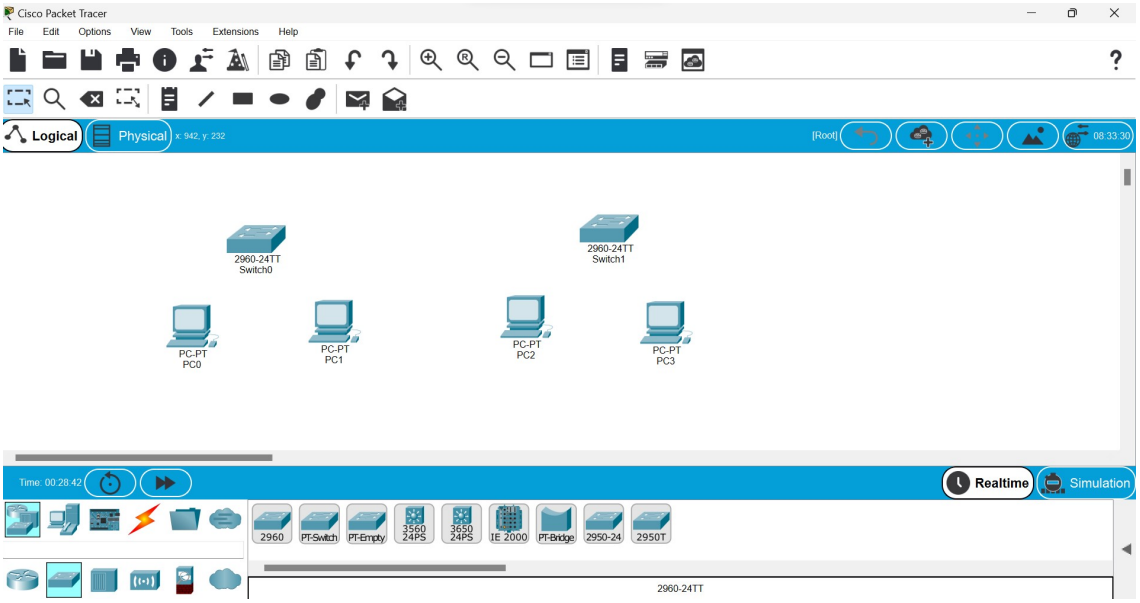
If required, add two or more devices to topology area.



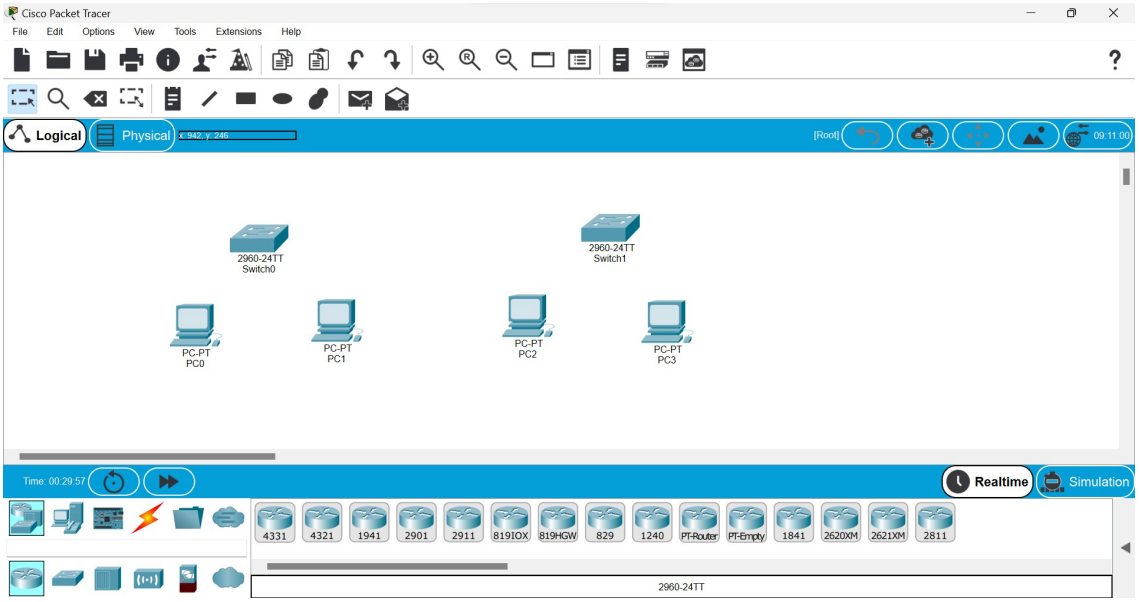
Single click on the **Network Devices(Switches)** and Single click on generic switch.



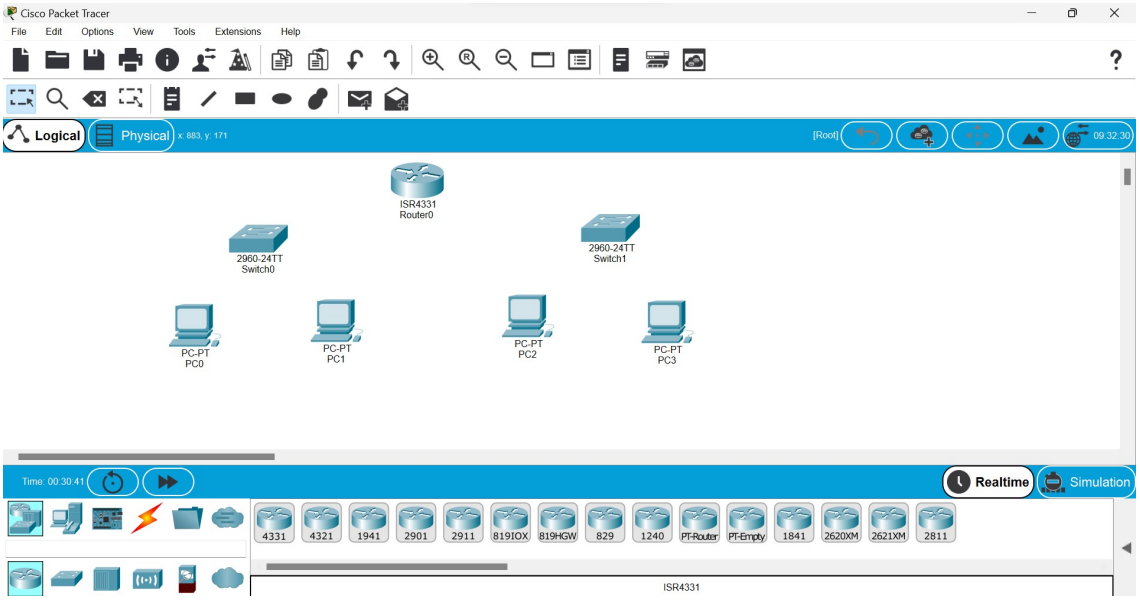
Move the cursor into topology area. You will notice it turns into a plus “+” sign. Single click in the topology area and it copies the device.



Single click on the **Network Devices(Routers)** and Single click on generic router.

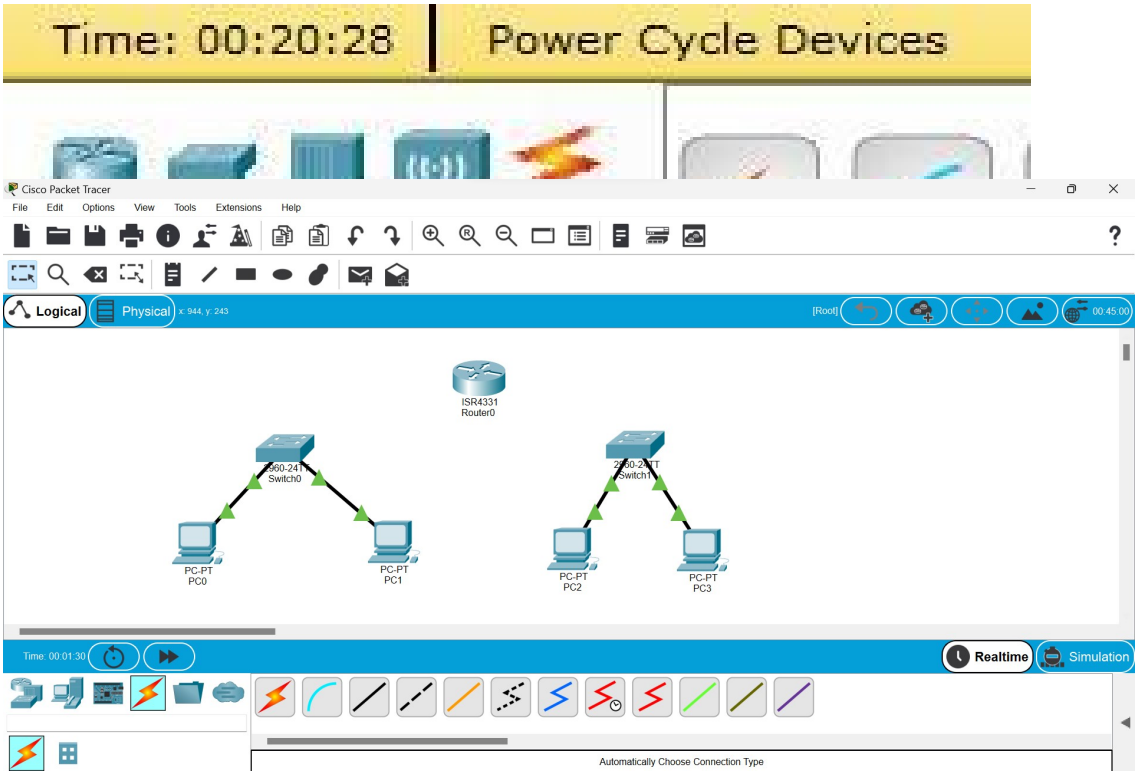


Move the cursor into topology area. You will notice it turns into a plus “+” sign. Single click in the topology area and it copies the device.



Step 3: Connecting the Hosts to Switches and Routers

Connect End Devices to switch by first choosing **Connections (Automatically Choose Connection Type).**



Perform the following steps to connect switches to router:

1. Choose copper cross over wire and click once on switch
2. Choose gigabit ethernet
3. Drag the cursor to router

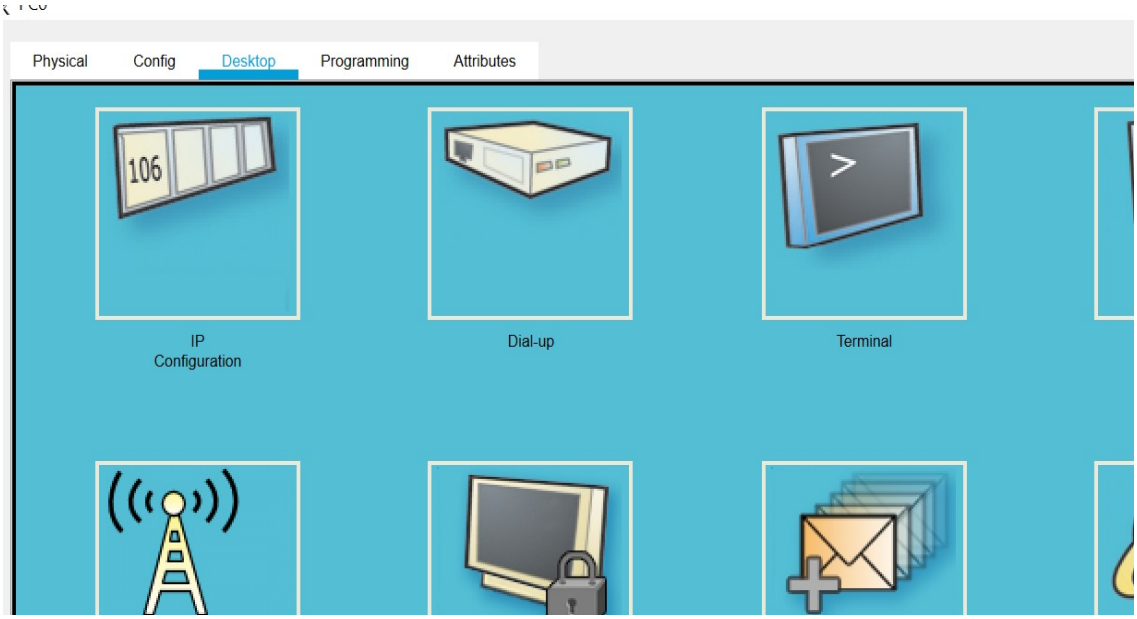
4. Click once on router and choose gigabit ethernet port

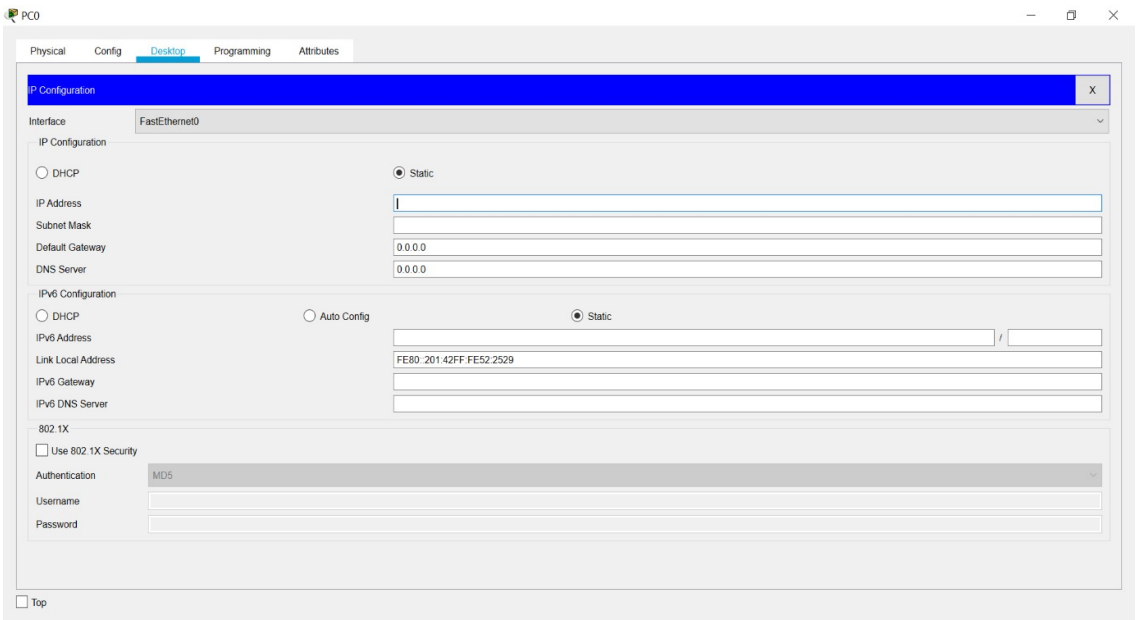
Step 4: Configuring IP Address for end devices and router in the topology area

Setting IP Address for End Devices

Before we can communicate between the hosts, we need to configure IP addresses on the devices.

Click on end device and go to desktop menu, then go to IP configuration and set the IP Adress.





Repeat these steps for the remaining end devices on the network to create IP address.
Choose different series of IP address for two different networks (for ex for network 1,Set IP address like 10.10.1.0 and for network 2 set IP address like 192.16.1.2)

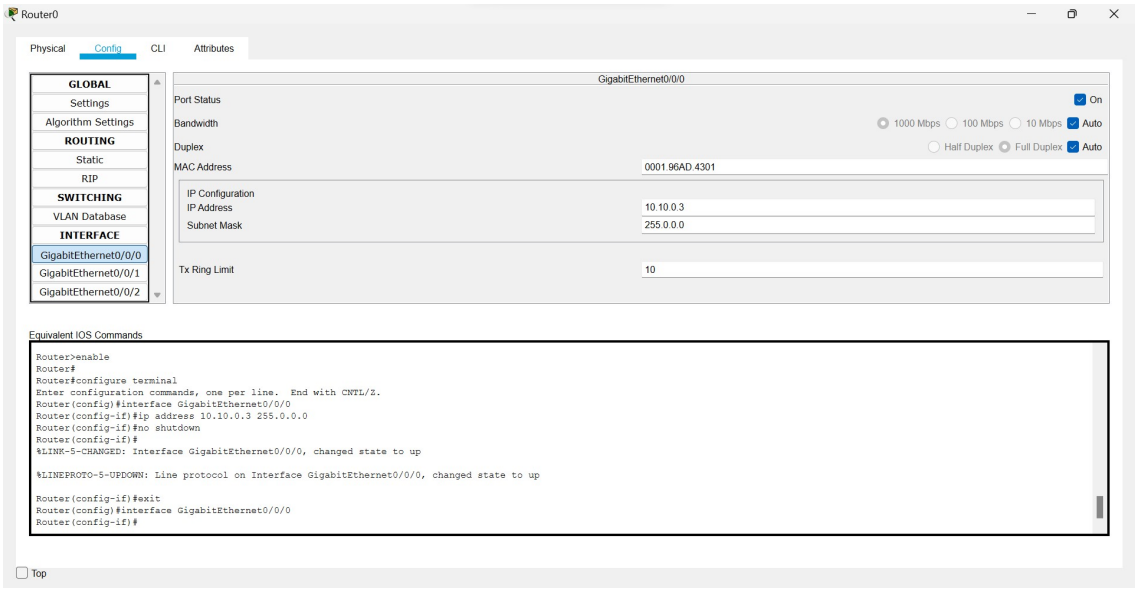
Setting IP address for router

Perform the following steps to configure router:

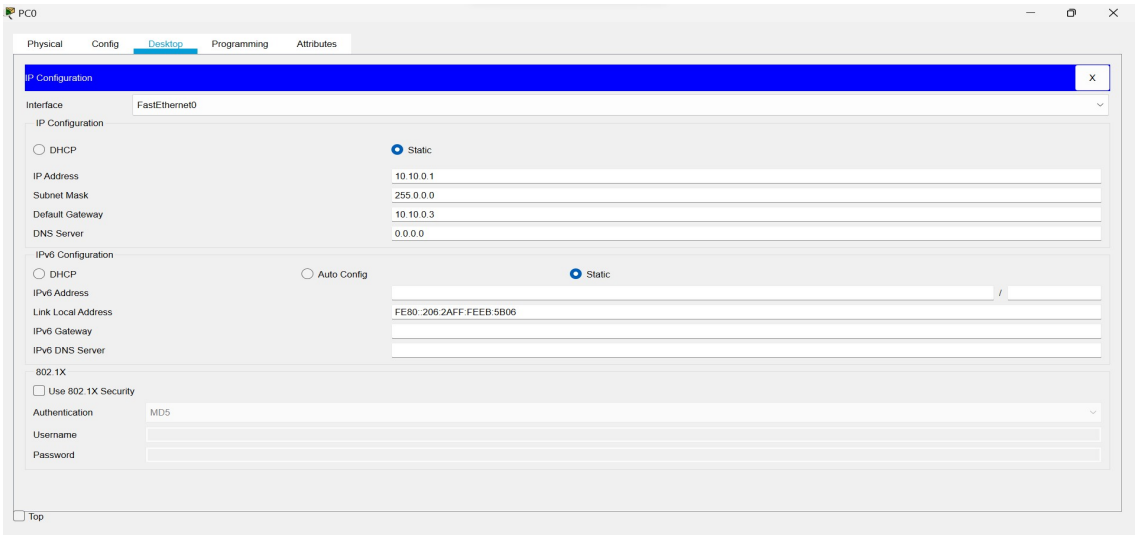
1)Click once on router and go to config

2)Select gigabit ethernet and set different IP address(for ex for network 1,Set IP address like 10.10.1.3 and for network 2 set IP address like 192.16.1.3)

3)Set the port status(ie on).



Then click once on end device and select IP configuration and then set default gateway address(IP address for network 1 is 10.10.1.3 and for network 2 , IP address is 192.16.1.3).

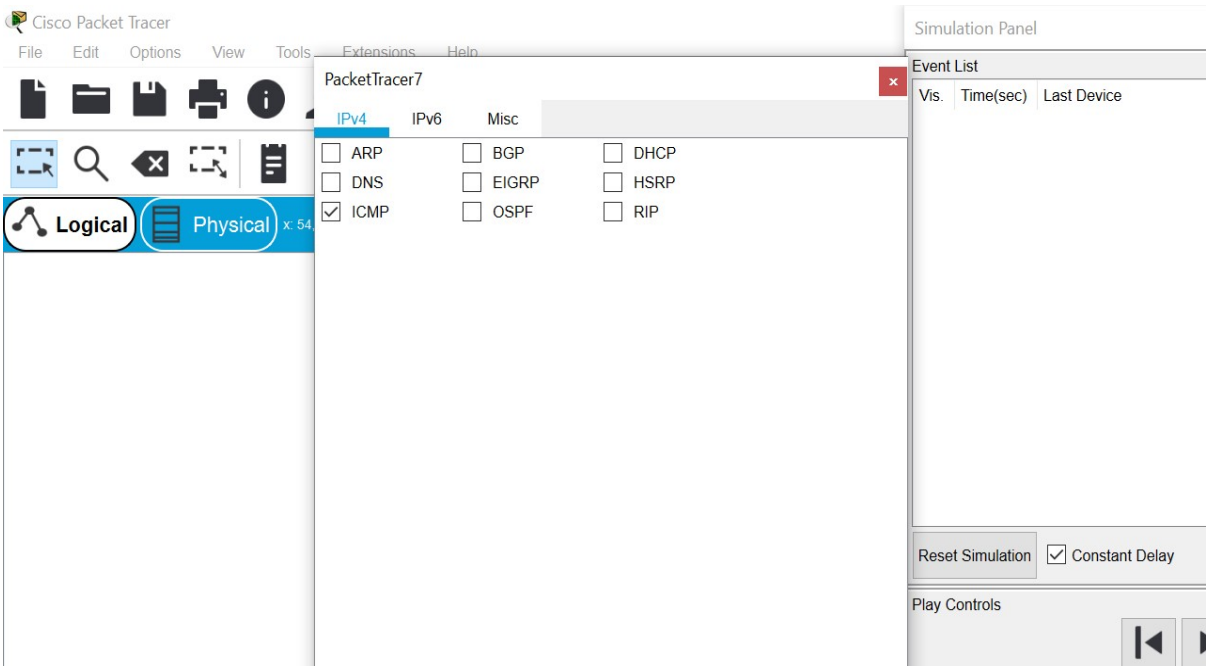


Repeat the following steps for all end devices.

Step 5: Verifying Connectivity in Simulation Mode

Be sure you are in **Simulation** mode. To ensure this, go to view tab in packet tracer and press simulation mode.

Deselect all filters (All/None) and select only **ICMP**.



Select the **Add Simple PDU** tool to ping devices. Click once on one end device, then once on another end device.

Continue clicking **Capture/Forward** button until the ICMP ping is completed. You should see the ICMP messages move between the hosts , switch and router. The PDU **Last Status** should show as **Successful**.

Result:

Thus the layer 3 functionality was verified successfully using packet tracer tool.

Exp No:

Capture and Analyse ICMPv6 Messages using Wireshark

Aim:

To capture and analyse ICMP messages using Wireshark Network Analyzer Tool.

Apparatus Required:

Wireshark Network Analyzer Tool

Theory:

IP stands for **internet protocol** which is defined in the TCP/IP model used for sending the packets from source to destination(i.e IP protocol is a best-effort delivery service that delivers a datagram from its original source to its final destination). However, it has two deficiencies: lack of error control and lack of assistance mechanisms. The IP protocol has no error-reporting or error-correcting mechanism. What happens if something goes wrong? What happens if a router must discard a datagram because it cannot find a router to the final destination, or because the time-to-live field has a zero value? What happens if the final destination host must discard all fragments of a datagram because it has not received all fragments within a predetermined time limit? These are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host. The IP protocol also lacks a mechanism for host and management queries. A host sometimes needs to determine if a router or another host is alive. And sometimes a network administrator needs information from another host or router. Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies. It is a companion to the IP protoco1.ICMP is a network layer protocol used to determine whether or not data is reaching its intended destination in a timely manner.

General Format of ICMP Messages

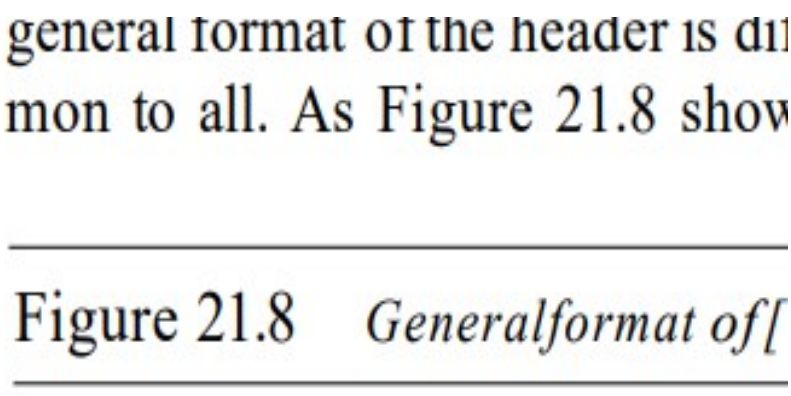

general format of the header is di
mon to all. As Figure 21.8 show

Figure 21.8 *Generalformat of[*

Type: (8 Bits)- This field defines the type of messages.ICMP messages are divided into two broad categories: error-reporting messages and information messages.

The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet. ICMPv6 error messages are Destination Unreachable, Source Quench, Time Exceeded, Parameter Problem and Redirection.

Destination Unreachable ICMPv6 error message: When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded and the router or the host sends a destination-unreachable message back to the source host that initiated the datagram.

Source Quench ICMPv6 Error Message: Whenever a device is sending too much data for the destination host to process, the recipient can send an ICMP Source Quench error message back to the sender, suggesting that the sender throttle back on the rate at which it is sending data.

Time Exceeded ICMPv6 Error Message: The time-exceeded message is generated in two cases: If routers use routing tables to find the next hop (next router) that must receive the packet. If there are errors in one or more routing tables, a packet can travel in a loop or a cycle, going from one router to the next or visiting a series of routers endlessly. Each datagram contains a field called time to live that controls this situation. When a datagram visits a router, the value of this field is decremented by 1. When the time-to-live value reaches 0, after decrementing, the router discards the datagram. However, when the datagram is discarded, a time-exceeded message must be sent by the router to the original source. Second, a time-exceeded message is also generated when not all fragments that make up a message arrive at the destination host within a certain time limit.

Parameter Problem ICMPv6 Error Message: Any ambiguity in the header part of a datagram can Create serious problems as the datagram travels through the Internet. If a router or the destination host discovers an ambiguous or missing value in any field of the datagram, it discards the datagram and sends a parameter-problem message back to the source.

Redirection ICMPv6 Error Message: A redirection error message is used when a router needs to tell a sender that it should use a different path for a specific destination.

ICMPv6 Query Messages:

The query messages help a host or a network manager to get specific information from a router or another host. ICMPv6 query messages are Echo Request and Reply, Timestamp Request and Reply, Address Mask Request and Reply,Router solicitation and advertisement, Neighbor Solicitation and Advertisement, Multicast Listener Report Message.

Echo Request and Reply ICMPv6 Query Messages:

The echo-request and echo-reply messages are designed for diagnostic purposes. The combination of echo-request and echo-reply messages determines whether two systems (hosts or routers) can communicate with each other. The echo-request and echo-reply messages can be used to determine if there is communication at the IP level. Because ICMP messages are encapsulated in IP datagrams, the receipt of an echo-reply message by the machine that sent the echo request is proof that the IP protocols in the sender and receiver are communicating with each other using the IP datagram. Also, it is proof that the intermediate routers are receiving, processing, and forwarding IP datagrams.

Timestamp Request and ReplyICMPv6 Query Messages: Two machines (hosts or routers) can use the timestamp request and timestamp reply messages to determine the round-trip time needed for an IP datagram to travel between them. It can also be used to synchronize the clocks in two machines.

Address-Mask Request and Reply ICMPv6 Query Messages:

A host may know its IP address, but it may not know the corresponding mask. For example, a host may know its IP address as 159.31.17.24, but it may not know that the corresponding mask is /24. To obtain its mask, a host sends an address-mask-request message to a router on the LAN. If the host knows the address of the router, it sends the request directly to the router. If it does not know, it

broadcasts the message. The router receiving the address-mask-request message responds with an address-mask-reply message, providing the necessary mask for the host. This can be applied to its full IP address to get its subnet address.

Router Solicitation and Advertisement ICMPv6 Query Messages:

A host that wants to send data to a host on another network needs to know the address of routers connected to its own network. Also, the host must know if the routers are alive and functioning. The router-solicitation and router-advertisement messages can help in this situation. A host can broadcast (or multicast) a router-solicitation message. The router or routers that receive the solicitation message broadcast their routing information using the router-advertisement message. A router can also periodically send router-advertisement messages even if no host has solicited.

Neighbor Solicitation and Advertisement ICMPv6 Query Messages:

The Neighbor Solicitation message allows a device to check that a neighbor exists and is reachable, and to initiate address resolution. The Neighbor Advertisement message confirms the existence of a host or router.

Multicast Listener Report Message ICMPv6 Query Messages:

Multicast Listener Report Message is an ICMPv6 message sent by a host when joining or leaving a multicast group, or in response to a Multicast Listener Query message sent by a router.

Code:-(8 Bits)-This field specifies the reason for the particular message type.

Checksum:(16 Bits)- It is a 16-bit field to detect whether the error exists in the message or not.

Procedure:

Step 1: Open Wireshark Network Analyzer Tool.

Step 2: Capture any one traffic.

Step 3: Capture any one data transmission using ICMP and analyse it.

Result:

Thus the ICMP messages was captured and analysed using wireshark network analyzer.

Exp No:

Capture and Analyse ARP Segment using Wireshark

Aim:

To capture and analyse ARP messages using Wireshark Network Analyzer Tool.

Apparatus Required:

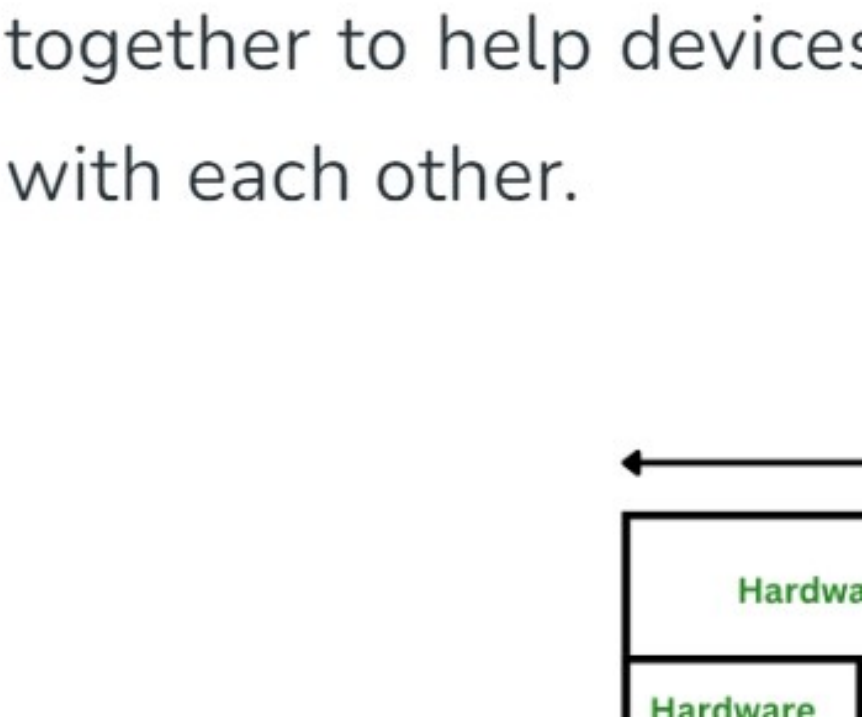
Wireshark Network Analyzer Tool

Theory:

The acronym ARP stands for **Address Resolution Protocol** which is one of the most important protocols of the network layer in the OSI model. ARP is a communication mechanism that is used to translate a logical address, such as an IP address, to a physical (MAC) address on a local network. ARP packets are transmitted and received on a network to achieve this mapping.

ARP Packet Format

The ARP packet format is used for ARP requests and replies and consists of multiple fields including hardware type, protocol type, hardware and protocol size, operation, sender and target hardware, and IP addresses. These fields work together to help devices on a network find and communicate with each other.



Hardware type: This is 16 bits field defining the type of the network on which ARP is running. Ethernet is given type 1.

Protocol type: This is 16 bits field defining the protocol. The value of this field for the IPv4 protocol is 0800H.

Hardware length: This is an 8 bits field defining the length of the physical address in bytes. Ethernet is the value 6.

Protocol length: This is an 8 bits field defining the length of the logical address in bytes. For the IPv4 protocol, the value is 4.

Operation (request or reply): This is a 16 bits field defining the type of packet. Packet types are ARP request (1), and ARP reply (2).

Sender hardware address: This is a variable length field defining the physical address of the sender. For example, for Ethernet, this field is 6 bytes long.

Sender protocol address: This is also a variable length field defining the logical address of the sender. For the IP protocol, this field is 4 bytes long.

Target hardware address: This is a variable length field defining the physical address of the target. For Ethernet, this field is 6 bytes long. For the ARP request messages, this field is all Os because the sender does not know the physical address of the target.

Target protocol address: This is also a variable length field defining the logical address of the target. For the IPv4 protocol, this field is 4 bytes long.

Procedure:

Step 1: Open Wireshark Network Analyzer Tool.

Step 2: Capture any one traffic.

Step 3: Capture any one data transmission using ARP and analyse it.

Result:

Thus the ARP messages was captured and analysed using wireshark network analyzer.

Exp No:

Capture and Analyse TCP Segment using Wireshark

Aim:

To capture and analyse TCP messages using Wireshark Network Analyzer Tool.

Apparatus Required:

Wireshark Network Analyzer Tool

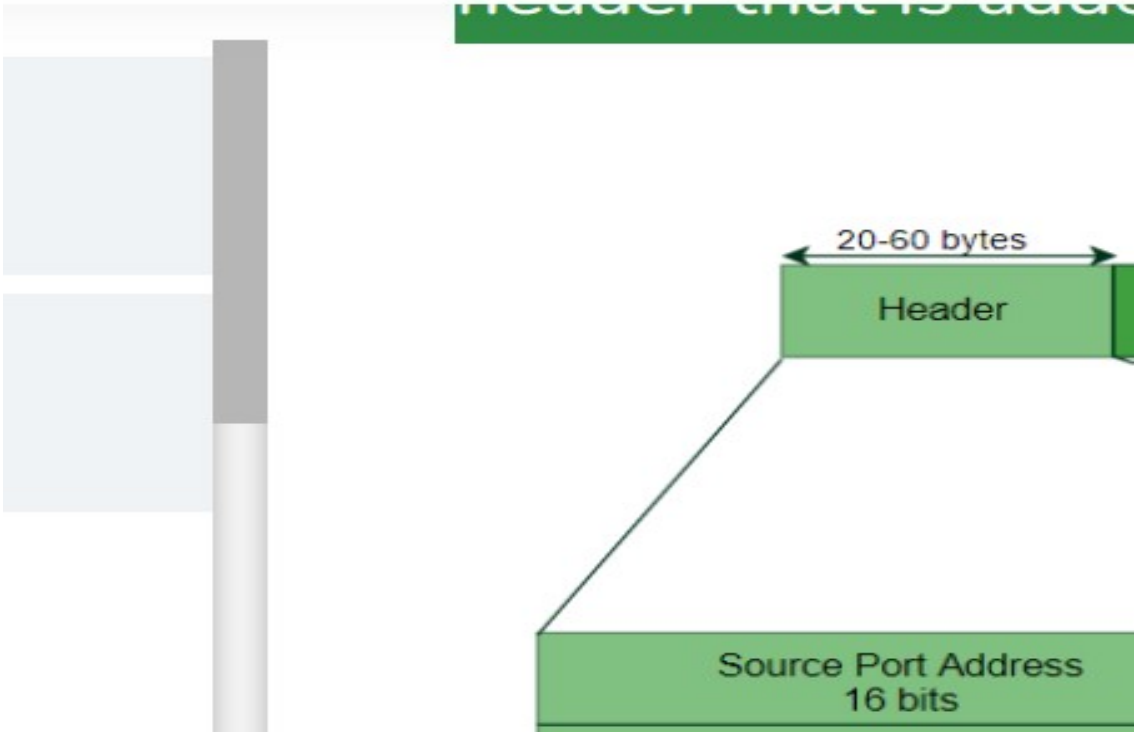
Theory:

TCP means transmission control protocol. It is commonly used protocol in transport layer in fourth layer of OSI Model. Transport Layer is the **heart** of Open System Interconnection Model. TCP is a connection-oriented protocol which means that connection needs to be established before the data is transmitted over the network. The transport layer takes services from the Application layer and provides services to the Network layer.

At the sender's side: The transport layer receives data (message) from the Application layer and then performs segmentation, divides the actual message into **segments,** adds the **source and destination's port numbers** into the header of the segment, and transfers the message to the Network layer.

At the receiver's side: The transport layer receives data from the Network layer, reassembles the segmented data, reads its header, identifies the port number, and forwards the message to the appropriate port in the Application layer.

A TCP segment consists of data bytes to be sent and a header that is added to the data by TCP as shown below



Source Port

A 16-bit field that holds the port address of the application that is sending the data segment.

Destination Port

A 16-bit field that holds the port address of the application in the host that is receiving the data segment.

Sequence Number

Sequence number is a 32 bit field. TCP assigns a unique sequence number to each byte of data contained in the TCP segment. This field contains the sequence number of the first data byte.

Acknowledgement Number

Acknowledgment number is a 32-bit field. It is an acknowledgement for the previous bytes being received successfully.

Header Length

Header length is a 4 bit field. It contains the length of TCP header.It helps in knowing from where the actual data begins.

The initial 5 rows of the TCP header are always used.So, minimum length of TCP header = 5 x 4 bytes = 20 bytes.

The size of the 6th row representing the Options field vary. The size of Options field can go up to 40 bytes.So, maximum length of TCP header = 20 bytes + 40 bytes = 60 bytes.

- If header length field contains decimal value 5 (represented as 0101), then

Header length = 5 x 4 = 20 bytes

Reserved Bits

The 6 bits are reserved. These bits are not used.

URG Bit

URG bit is used to treat certain data on an urgent basis.

When URG bit is set to 1,it indicates the receiver that **certain amount of data within the current segment** is urgent.

ACK Bit

ACK bit indicates whether acknowledgement number field is valid or not.

When ACK bit is set to 1, it indicates that acknowledgement number contained in the **TCP header** is valid.

PSH Bit

PSH bit is used to push the entire buffer immediately to the receiving application.

When PSH bit is set to 1,all the segments in the buffer are immediately pushed to the receiving application.

RST Bit

RST bit is used to reset the TCP connection.

When RST bit is set to 1,it indicates the receiver to terminate the connection immediately.

SYN Bit

SYN bit is used to synchronize the sequence numbers.

When SYN bit is set to 1, it indicates the receiver that the sequence number contained in the TCP header is the initial sequence number.

FIN Bit

FIN bit is used to terminate the TCP connection.

When FIN bit is set to 1,it indicates the receiver that the sender wants to terminate the connection.

Window Size

Window size is a 16 bit field. **Window size** the **most important part** in the TCP header. This field is used by the receiver to indicate to the sender, the amount of data that it can accept.

Checksum

Checksum is a 16 bit field used for error control. It verifies the integrity of data in the TCP payload.

Urgent Pointer

When the URG bit is set to 1, the Urgent Pointer is also set to 1. The URG pointer tell how many bytes of the data is urgent in the segment that has arrived.

Procedure:

Step 1: Open Wireshark Network Analyzer Tool.

Step 2: Capture any one traffic.

Step 3: Capture any one data transmission using TCP and analyse it.

Result:

Thus the TCP messages was captured and analysed using wireshark network analyzer.

Exp No:

Capture and Analyse UDP Segment using Wireshark

Aim:

To capture and analyse UDP messages using Wireshark Network Analyzer Tool.

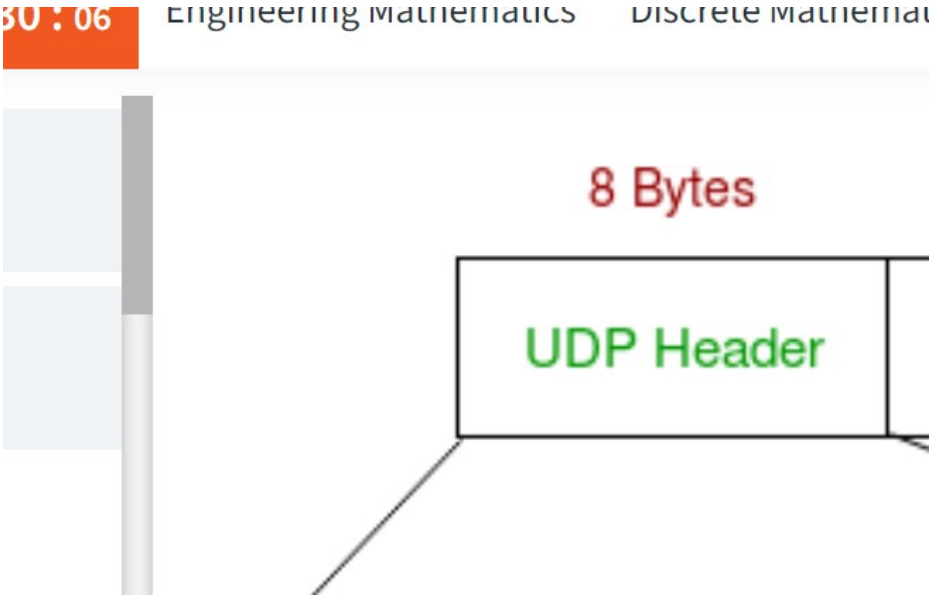
Apparatus Required:

Wireshark Network Analyzer Tool

Theory:

User Datagram Protocol (UDP) is a Transport Layer protocol. Unlike TCP, it is an **unreliable and connectionless protocol.** So, there is no need to establish a connection prior to data transfer.

UDP header is an **8-bytes** fixed and simple header. The first 8 Bytes contains all necessary header information and the remaining part consist of data. UDP port number fields are each 16 bits long, therefore the range for port numbers is defined from 0 to 65535; port number 0 is reserved. Port numbers help to distinguish different user requests or processes.



1. **Source Port:** Source Port is a 2 Byte long field used to identify the port number of the source.
2. **Destination Port:** It is a 2 Byte long field, used to identify the port of the destined packet.
3. **Length:** It is a 16-bits field. This field indicates the length of UDP including the header and the data.
4. **Checksum:** Checksum is 2 Bytes long field. The final two bytes of the UDP header is the checksum, a field that's used by the sender and receiver to check for data corruption.

Before sending off the segment, the sender:

1. Computes the checksum based on the data in the segment.
2. Stores the computed checksum in the field.

Upon receiving the segment, the recipient:

1. Computes the checksum based on the received segment.
2. Compares the checksums to each other. If the checksums aren't equal, it knows the data was corrupted.

Procedure:

Step 1: Open Wireshark Network Analyzer Tool.

Step 2: Capture any one traffic.

Step 3: Capture any one data transmission using UDP and analyse it.

Result:

Thus the UDP messages was captured and analysed using wireshark network analyzer.

Exp No:

DOMAIN NAME SERVICES (DNS)

Aim:

To domain name service using UDP protocol.

System and Software Requirements:

- PC with UNIX/ Linux Operating systems.
- C compiler in Linux Environment

Theory:

Every host that runs TCP/IP must have a unique IP address that's used when communicating with other computers in a network. Computers operate easily with IP addresses, but people don't; users would rather identify systems by a name. To facilitate effective and efficient communication, users need to be able to refer to computers by name, and still have their computer use IP addresses transparently.

The Domain Name System (DNS) is a hierarchical naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participants. Most importantly, it translates domain names meaningful to humans into the numerical (binary) identifiers associated with networking equipment for the purpose of locating and addressing these devices worldwide. An often used analogy to explain the Domain Name System is that it serves as the "phone book" for the Internet by translating human-friendly computer hostnames into IP addresses. For example, www.example.com translates to 192.0.32.10.

In the early days of the ARPANET, the forerunner to today's Internet, there were only a small number of computers attached to the network. The Network Information Center (NIC), located at Stanford Research Institute (SRI), was responsible for compiling a single file, HOSTS.TXT, which contained the names and addresses of every computer. Administrators would email SRI, which would then update the HOSTS.TXT file. Next, ARPANET users would download the new version of HOSTS.TXT using File Transfer Protocol (FTP).

As the ARPANET grew, it became obvious that this approach wouldn't scale, for the following three key reasons:

The bandwidth consumed in transmitting updated versions of an ARPANET-wide host file was proportional to the square of the number of hosts in the ARPANET. With the number of hosts growing at an exponential rate, the long-term impact was likely to be a load that no one host was going to be able to sustain.

The static flat host file also meant that no two computers on the ARPANET could have the same name. As the number of hosts grew, the risk of adding a duplicate name grew, as did the difficulty of trying to control this centrally.

The nature of the underlying network was changing—the large, timesharing computers that had once made up the ARPANET were being superseded by networks of workstation—seach of which needed to have a unique host name. This would be difficult, if not impossible, to control centrally.

As the ARPANET continued to grow, it became clear that ARPANET needed a better solution. Several proposals were generated based on the concept of a distributed naming service, which was based on a hierarchical name space. RFCs 882 and 883 emerged, which described the design for a domain name system, based on a distributed database containing generalized resource information. This design evolved, and RFCs 1034 and 1035 were issued to describe the Domain Name System (DNS) service used in today's Internet. This design continues to evolve, and a number of proposed updates and refinements are being discussed as this chapter is being written.

What Is DNS?

The DNS is an IETF-standard name service. The DNS service enables client computers on your network to register and resolve DNS domain names. These names are used to find and access resources offered by other computers on your network or other networks, such as the Internet. The following are the three main components of DNS:

- Domain name space and associated resource records (RRs) A distributed database of name-related information.
- DNS Name Servers Servers that hold the domain name space and RRs, and that answer queries from DNS clients.
- DNS Resolvers The facility within a DNS client that contacts DNS name servers and issues name queries to obtain resource record information.

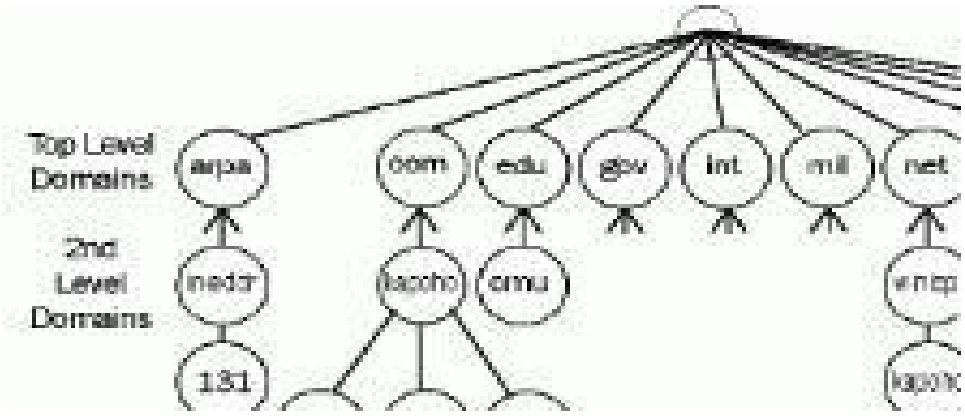
Key DNS Terms

This section describes the key components of the DNS and defines key DNS terms.

Domain Name Space

The domain name space is a hierarchical, tree-structured name space, starting at an unnamed root used for all DNS operations. In the DNS name space, each node and leaf in the domain name space tree represents a named domain. Each domain can have additional child domains. Figure illustrates the structure of Internet domain name space.

Figure: Domain name space for the Internet.



Domain Names

Each node in the DNS tree, as Figure illustrates, has a separate name, referred to in RFC 1034 as a label. Each DNS label can be from 1 through 63 characters in length, with the root domain having a length of zero characters.

A specific node's domain name is the list of the labels in the path from the node being named to the DNS Tree root. DNS convention is that the labels that compose a domain name are read left to right—from the most specific to the root (for example, www.kapoho.com). This full name is also known as the fully qualified domain name (FQDN).

Domain names can be stored as upper case or lower case, but all domain comparisons and domain functions are defined, by RFC 1034, to be case insensitive. Thus, www.kapoho.com is identical to WWW.KAPOHO.COM for domain naming operations.

Top-Level Domains

A top-level domain is a DNS domain directly below the root. As the above mentioned figure illustrates, a number of top-level domains have been defined. Additional names (at least for the Internet) are difficult to create. The following are the three categories of top-level domains:

- "ARPA" This is a special domain—used today for reverse-name lookups.
- 3-letter domains There are six 3-character, top-level domains noted below.

- 2-letter country-based domain names These country code domains are based on the International Organization for Standardization (ISO) country name, and are used principally by companies and organizations outside the US. The exception is the UK, which uses .uk as the top-level domain, even though the ISO country code is GB. Table shows the six top-level domains in use today, as defined by RFC 1591.

3-Character	
Domain Name	**Use**
Com	Commercial organizations, such as microsoft.com for the Microsoft Corporation
Edu	Educational institutions, now mainly four-year colleges and universities, such as cmu.edu for Carnegie Mellon University
Gov	Agencies of the US Federal Government, such as fbi.gov for the US Federal Bureau of Investigation
Int	Organizations established by international treaties, such as nato.int for NATO
Mil	US military, such as af.mil for the US Air Force
Net	Computers of network providers, organizations dedicated to the
	Internet, Internet Service Providers (ISPs), and so forth, such as internic.net for the Internet Network Information Center (InterNIC)
Org	A top-level domain for groups that don't fit anywhere else, such as non-government or non-profit organizations (for example, reiki.org for information about Reiki)

Note: While these are the only 3-letter domains available today, there is pressure to expand this number; we may well end up with more in the future.

Resource Records (RR)

A resource record is a record containing information relating to a domain that the DNS database can hold and that a DNS client can retrieve and use. For example, the host RR for a specific

domain holds the IP address of that domain (host); a DNS client will use this RR to obtain the IP address for the domain.

Each DNS server contains the RRs relating to those portions of the DNS namespace for which it's authoritative (or for which it can answer queries sent by a host). When a DNS server is authoritative for a portion of the DNS name space, those systems' administrators are responsible for ensuring that the information about that DNS name space portion is correct. To increase efficiency, a given DNS server can cache the RRs relating to a domain in any part of the domain tree.

There are numerous RR types defined in RFCs 1035 and 1036, and in later RFCs. Most of the RR types are no longer needed or used.

Table lists the key RRs that might be used in a Windows 2000 network. (For more detail on the contents of specific RRs, see the "DNS Resource Records" section later in this chapter.)

Table Key Resource Records as Used by a Windows 2000 Network.

Type	Contents	Use
A	Host Address	Used to hold a specific host's IP address.
CNAME	Canonical Name (alias)	Used to make an alias name for a host.
MX	Mail Exchanger	Provides message routing to a mail server, plus backup server(s) in case the target server isn't active.
NS	Name Server	Provides a list of authoritative servers for a domain or indicates authoritative DNS servers for any delegated sub-domains.
PTR	Pointer	Used for reverse lookup—resolving an IP address into a domain name using the IN-ADDR.ARPA domain.
SOA	Start of Authority	Used to determine the DNS server that's the primary server for a DNS zone and to store other zone property information.

RRs can be attached to any node in the DNS tree, although RRs won't be provided in some domains (for example, Pointer (PTR) RRs are found only in domains below the in-addr.arpa domain). Thus, higher-level domains, such as microsoft.com, can have individual RRs (for example, Mail Exchange (MX) record for mail to be sent to the Microsoft Corporation) as well as having sub-domains that also might have individual RRs (for instance, eu.microsoft.com, which has a host record www.eu.microsoft.com).

Canonical Names

The Canonical Name (CNAME) RR enables the administrator to create an alias to another domain name. The use of CNAME RRs are recommended for use in the following scenarios:

- When a host specified in an (A) RR in the same zone needs to be renamed. For example, if you need to rename kona.kapoho.com to hilo.kapoho.com, you could create a CNAME entry for kona.kapoho.com to point to hilo.kapoho.com.
- When a generic name for a well-known service, such as ftp or www, needs to resolve to a group of individual computers (each with an individual (A) RR). For example, you might want www.kapoho.com to be an alias for kona.kapoho.com and hilo.kapoho.com. A user will access www.kapoho.com and generally won't be aware of which computer is actually servicing this request.

DNS Query Operation

A DNS client issues a query operation against a DNS server to obtain some or all of the RR information relating to a specific domain, for instance, to determine which host (A) record or records are held for the domain named kapoho.com. If the domain exists and the requested RRs exist, the DNS server will return the requested information in a query reply message. The query reply message will return both the initial query and a reply containing the relevant records, assuming the DNS server can obtain the required RRs.

A DNS query, referred to in RFC 1034 as a standard query, contains a target domain name, a query type, and a query class. The query will contain a request for the specific RR(s) that the resolver wished to obtain (or a request to return all RRs relating to the domain).

DNS Update Operation

A DNS update operation is issued by a DNS client against a DNS server to update, add, or delete some or all of the RR information relating to a specific domain, for instance, to update the host record for the computer named kona.kapoho.com to point to 10.10.1.100. The update operation is also referred to as a dynamic update.

DNS Zones

A DNS server that has complete information for part of the DNS name space is said to be the authority for that part of the name space. This authoritative information is organized into units called zones, which are the main units of replication in DNS. A zone contains one or more RRs for one or more related DNS domains.

The following are the three DNS zone types implemented in Windows 2000:

- Standard Primary Holds the master copy of a zone and can replicate it to secondary zones. All changes to a zone are made on the standard primary.
- Standard Secondary Contains a read-only copy of zone information that can provide increased performance and resilience. Information in a primary zone is replicated to the secondary by use of the zone transfer mechanism.
- Active Directory-integrated A Microsoft proprietary zone type, where the zone information is held in the Windows 2000 Active Directory (AD) and replicated using AD replication.

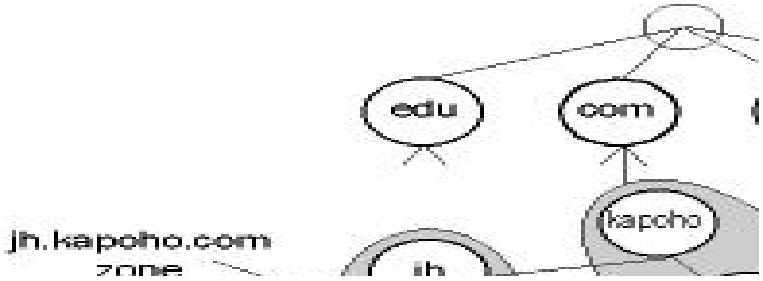
Traditionally, the master copy of each zone is held in a primary zone on a single DNS server. On that server, the zone has a Start Of Authority (SOA) record that specifies it to be the primary zone. To improve performance and redundancy, a primary zone can be automatically distributed to one or more secondary zones held on other DNS servers. When changes are made to the zone, for instance, to add an (A) record, the changes are made to the primary zone and are transferred to the secondary zone. The transfer of zone information is handled by the zone replication process, which is described later in the "Zone Transfer" section.

The zone will only hold information about a single DNS domain name, for example, kapoho.com. After the zone is created, the administrator can then add RRs to the zone, or can set the domain to be dynamically updated. For example, the administrator could add (A) records (host records) for hosts in the domain, such as kona.kapoho.com. If dynamic updates are enabled for the zone, a Windows 2000 computer can then directly update the A and PTR records on the DNS server (if the DNS client is also a DHCP client, the administrator can configure a DHCP server to send the updates).

Once the administrator has created the zone, he can add additional sub-domains to the zone (for example, jh.kapoho.com). These might be added to provide DNS services to a new building that

is managed separately from the parent domain. This sub-domain, which might reside in a separate zone, would have RRs added (for example, a host record for jasmine.jh.kapoho.com).

As the following figure illustrates, if other domains are added below the domain used initially to create the zone, these domains can either be part of the same zone or belong to another. For example, the sub-domain jh.kapoho.com, which is subordinate to kapoho.com, could be held in the same zone as kapoho.com, or in a separate zone. This allows the sub-domain to be managed and included as part of the original zone records, or to be delegated away to another zone created to support that sub-domain.



In this example, the domain kapoho.com has a sub-domain of jh.kapoho.com. Additionally, both domains contain a single host record. In this example, the domains jh.kapoho.com and kapoho.com are held in separate zones on different DNS servers. The kapoho.com zone holds one host record for kona.kapoho.com. The jh.kapoho.com domain holds the host record for the host jasmine.jh.kapoho.com.

Active Directory-Integrated Zones

A major new feature in the Windows 2000 DNS service is the ability to store DNS zones within the AD. An Active Directory-integrated zone is a primary DNS zone that's held within the AD and replicated to other AD primary zones, using AD replication (and not traditional zone transfer). Although this method of holding zones is a Microsoft proprietary approach, it can provide some useful benefits.

The main advantage of AD-integrated zones is that the zones become, in effect, multi-master, with the capability of updates being made to any DNS server. This can increase the fault tolerance of the DNS service. In addition, replication of zone information occurs using AD replication, which can be more efficient across slow links, because of the way that AD compresses replication data between sites.

Reverse-Lookup Zones

Most queries sent to a DNS server involve a search based on the DNS name of another computer as stored in an address (A) RR. This type of query expects an IP address as the resource data for the answered response. This type of query is generally referred to as a forward query. DNS also provides a reverse-lookup process, which enables a host to determine another host's name based on its IP address. For example, "What is the DNS domain name of the host at IP address 10.10.1.100?"

To allow for reverse queries, a special domain, in-addr.arpa, was defined and reserved in the Internet DNS name space. Sub-domains within the in-addr.arpa domain are named using the reverse ordering of the numbers in the dotted-decimal notation of IP addresses. The reverse ordering of the

domain name is needed because, unlike DNS names, IP addresses are read from left to right, but are interpreted in the opposite manner

(that is, the left-most part is more generalized than the right-most part). For this reason, the order of IP address octets is reversed when building the in-addr.arpa domain tree; for example, the reverse-lookup zone for the subnet 192.168.100.0 is 100.168.192.in-addr.arpa.

This approach enables the administration of lower limbs of the DNS in-addr.arpa tree to be delegated to an organization when it obtains a set of IP addresses from an IP registry.

The in-addr.arpa domain tree makes use of the PTR RR. The PTR RR is used to associate the IP address to the owning domain name. This lookup should correspond to an Address RR for the host in a forward-lookup zone. The success of a PTR RR used in reverse query depends on the validity of its pointer data, the (A) RR, which must exist.

Note: The in-addr.arpa domain is used only for Internet Protocol version 4 (IPv4)-based networks. In the Windows 2000 DNS Microsoft Management Console (MMC) snap-in, the DNS server's New Zone wizard will use this domain when it creates a new reverse-lookup zone. Internet Protocol version 6 (IPv6)-based reverse-lookup zones are based on the domain ip6.arpa.

Reverse Queries

A reverse query is one in which the DNS server is requested to return the DNS domain name for a host at a particular IP address. Reverse-Lookup Query messages are, in effect, standard queries, but relating to the reverse-lookup zone. The reverse-lookup zone is based on the in-addr.arpa domain name and mainly holds PTR RRs.

Inverse Queries

Inverse queries originally were described in RFC 1032, but now are outdated. Inverse queries were meant to look up a host name based on its IP address and use a nonstandard DNS query operation. The use of inverse queries is limited to some of the earlier versions of NSLOOKUP.EXE, a utility used to test and troubleshoot a DNS service.

DNS Query Classes

DNS queries fall into one of two classes: recursive queries and iterative queries. A recursive query is a DNS query sent to a DNS server in which the querying host asks the DNS server to provide a complete answer to the query, even if that means contacting other servers to provide the answer. When sent a recursive query, the DNS server will use separate iterative queries to other DNS servers on behalf of the querying host to obtain an answer for the query.

An iterative query is a DNS query sent to a DNS server in which the querying host requests it to return the best answer the DNS server can provide without seeking further assistance from other DNS servers.

In general, host computers issue recursive queries against DNS servers. The host assumes that the DNS server either knows the answer to the query, or can find the answer. On the other hand, a DNS server will generally issue iterative queries against other DNS servers if it is unable to answer a recursive query from cached information.

System Function Used in Program
Description

The gethostbyname() function returns a structure of type hostent for the given host name. Here name is either a host name, or an IPv4 address in standard dot notation, or an IPv6 address in colon (and possibly dot) notation. (See RFC 1884 for the description of IPv6 addresses.) If name is an IPv4 or IPv6 address, no lookup is performed and gethostbyname() simply copies name into the h_name field and its struct in_addr equivalent into the h_addr_list[0] field of the returned hostent structure. If name doesn't end in a dot and the environment variable HOSTALIASES is set, the alias file pointed to by HOSTALIASES will first be searched for name (see hostname(7) for the file format). The current domain and its parents are searched unless name ends in a dot.

The gethostbyaddr() function returns a structure of type hostent for the given host address addr of length len and address type type. Valid address types are AF_INET and AF_INET6. The host address argument is a pointer to a struct of a type depending on the address type, for example a struct in_addr * (probably obtained via a call to inet_addr()) for address type AF_INET.

The sethostent() function specifies, if stayopen is true (1), that a connected TCP socket should be used for the name server queries and that the connection should remain open during successive queries. Otherwise, name server queries will use UDP datagrams. The endhostent() function ends the use of a TCP connection for name server queries.

The (obsolete) herror() function prints the error message associated with the current value of h_errno on stderr.

The (obsolete) hstrerror() function takes an error number (typically h_errno) and returns the corresponding message string.

The domain name queries carried out by gethostbyname() and gethostbyaddr() use a combination of any or all of the name server named(8), a broken out line from /etc/hosts, and the Network Information Service (NIS or YP), depending upon the contents of the order line in /etc/host.conf. The default action is to query named(8), followed by /etc/hosts.

The hostent structure is defined in <netdb.h> as follows:

```
struct hostent { char  h_name;

/* official name of host */ char

**h_aliases; /* alias list */ int
h_addrtype;

/* host address type */ int

h_length;

/* length of address */ char

**h_addr_list;
/* list of addresses */
}

#define h_addr h_addr_list[0]
/* for backward compatibility */
```

The members of the hostent structure are: h_name

The official name of the host. h_aliases

An array of alternative names for the host, terminated by a NULL pointer. h_addrtype

The type of address; always AF_INET or AF_INET6 at present. h_length

The length of the address in bytes. h_addr_list

An array of pointers to network addresses for the host (in network byte order), terminated by a NULL pointer.
h_addr

The first address in h_addr_list for backward compatibility.

Return Value
The gethostbyname() and gethostbyaddr() functions return the hostent structure or a NULL pointer if an error occurs. On error, the h_errno variable holds an error number. When non-NULL, the return value may point at static data, see the notes below.
Errors

The variable h_errno can have the following values: HOST_NOT_FOUND

The specified host is unknown. NO_ADDRESS or NO_DATA

The requested name is valid but does not have an IP address. NO_RECOVERY

A non-recoverable name server error occurred. TRY_AGAIN

A temporary error occurred on an authoritative name server. Try again later. Files
/etc/host.conf

resolver configuration file
/etc/hosts

host database file
/etc/nsswitch.conf

name service switch configuration

Algorithm:

- Start the program
- Include necessary header files such as sys/socket.h, sys/types.h, neinet/in.h
- Create the variable for predefined structure hostent

- Get the members of the structure as Name, Address, Address type, Alias and address list with the help of the structure variable.
- Print all the output
- End the program.

Program:

```
#include<stdio.h>
#include<netdb.h>
#include<sys/socket.h>
#include<netinet/in.h>
int main(int argc,char *argv[])

{

struct hostent *h; h=gethostbyname(argv[1]); if(h==NULL)
{
herror("ERROR\n");
}
else
{
printf("\n Name  :%s ",h->h_name);
printf("\n Address:%s ",inet_ntoa((struct in_addr*)h->h_addr)); printf("\n Addresstype is
%d",h->h_addrtype);

printf("\n Alias:%s",h->h_aliases[0]);

printf("\n Address list:%s\n",inet_ntoa((struct in_addr *)h->h_addr_list[0]));

}
return 0;
}
```

How to Run the Server Program

```
[root @ root] $ cc dns.c
[root @ root] $ ./a.out local host
```

Output of the Program

```
Name  : local host. local domain
Address: 16.151.4.8
Addresstype is 2
Alias    : 1108550716.
Address list: 32.151.4.8
```

Result:

Thus the domain names was obtained.