Lab 4: Introduction to Packet Tracer

Theory

Cisco Packet Tracer is a network simulation software developed by Cisco Systems. It allows users to simulate complex networks by configuring various network devices and testing different scenarios in a virtual environment. The software is particularly useful for educational purposes, providing a platform for Ashlesha to experiment with network designs and configurations without the need for physical hardware.

Key Concepts of Cisco Packet Tracer

Packet Tracer offers a range of features that support network design, configuration,and simulation. Some key concepts include:

Network Simulation: The ability to simulate real-world network scenarios with a wide range of devices and protocols.

Real-Time and Simulation Modes: These modes allow users to view the network's behavior in real-time or to simulate the propagation of packets step by step.

Activity Wizard: A feature that enables the creation of guided learning activities within the software.

Interface of Cisco Packet Tracer

The interface of Cisco Packet Tracer is designed to be user-friendly, with various panels and tools to facilitate network design and simulation.

Work Space Details

The workspace is the primary area where you design and visualize your network topology. You can drag and drop devices such as routers, switches, and end devices onto the workspace and connect them using different types of cables. The workspace is equipped with grid lines for precise alignment and organization of components, allowing for a clear and structured network diagram.

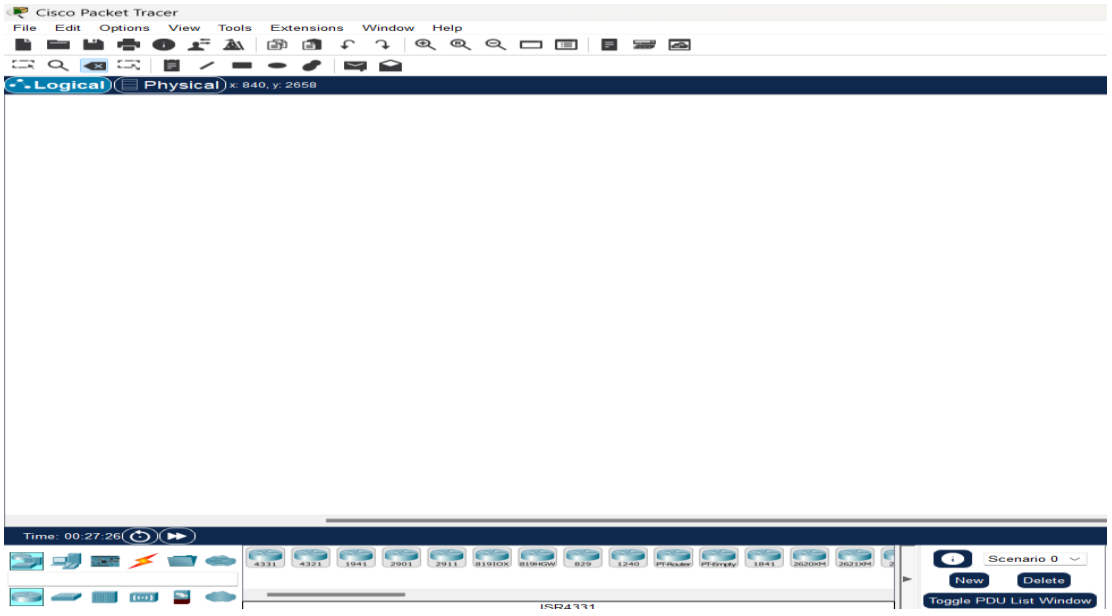


Fig 1.0: Cisco Packet Tracer Workspace

Toolbar

The toolbar provides quick access to essential functions like saving and opening files, zooming in and out, and controlling the simulation process. It also allows you to switch between different views, such as physical and logical views, of your network. The toolbar streamlines your workflow by offering direct access to frequently used tools and settings.

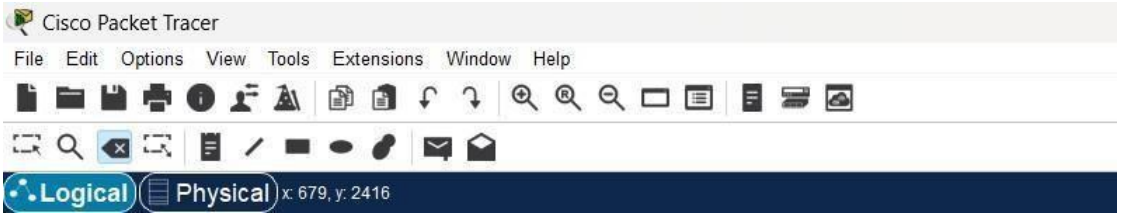


Fig1.1: Toolbar in Cisco packet tracer

Device-Type Selection Panel

This panel, located on the left side of the interface, lists all available network devices organized by category, such as routers, switches, and end devices. You can easily select a device from this panel and drag it onto the workspace to include it in your network design. This panel simplifies the process of finding and placing the necessary devices for your network topology.

Fig1.2: Device selection panel

Device Configurations

After placing a device on the workspace, you can configure its settings through a configuration window. This includes assigning IP addresses, setting up routing protocols, and enabling specific features like DHCP or NAT. The configuration can be done using both a graphical interface and a command-line interface (CLI), providing flexibility for both beginners and advanced users.

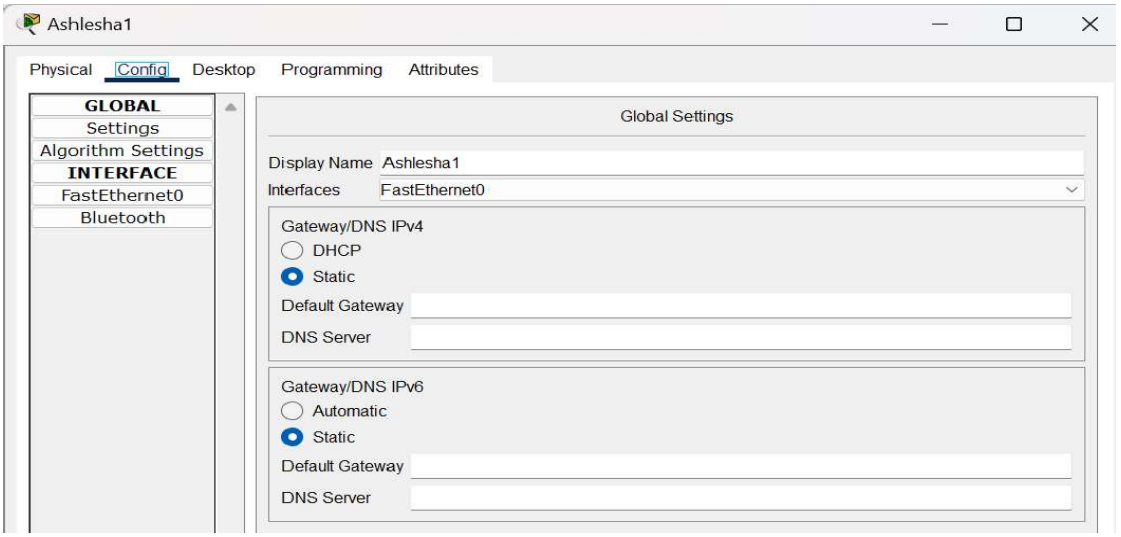


Fig 1.3: Device configuration settings

Real-Time and Simulation Mode

Packets Tracer operates in two modes: Real-Time Mode, where network actions occur instantly as they would in a real network, and Simulation Mode, which allows you to pause and step through the network's operations. Simulation Mode is particularly useful for analyzing packet flow, understanding protocol behavior, and troubleshooting network issues by observing data movement across the network.

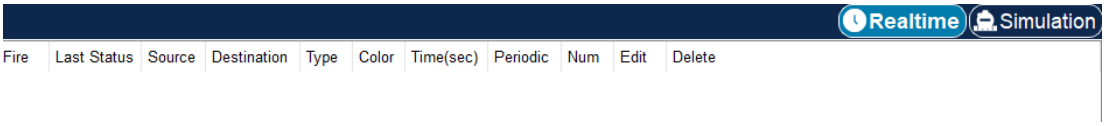


Fig1.4: Different modes in Cisco packet tracer

Network Component Icons and Labels

Devices and connections in the workspace are represented by specific icons, making it easy to identify different network components. Labels can be added to these icons to provide additional details, such as device names, IP addresses, or VLAN information. These labels help organize the network diagram and improve clarity, making it easier to understand and manage complex network designs.

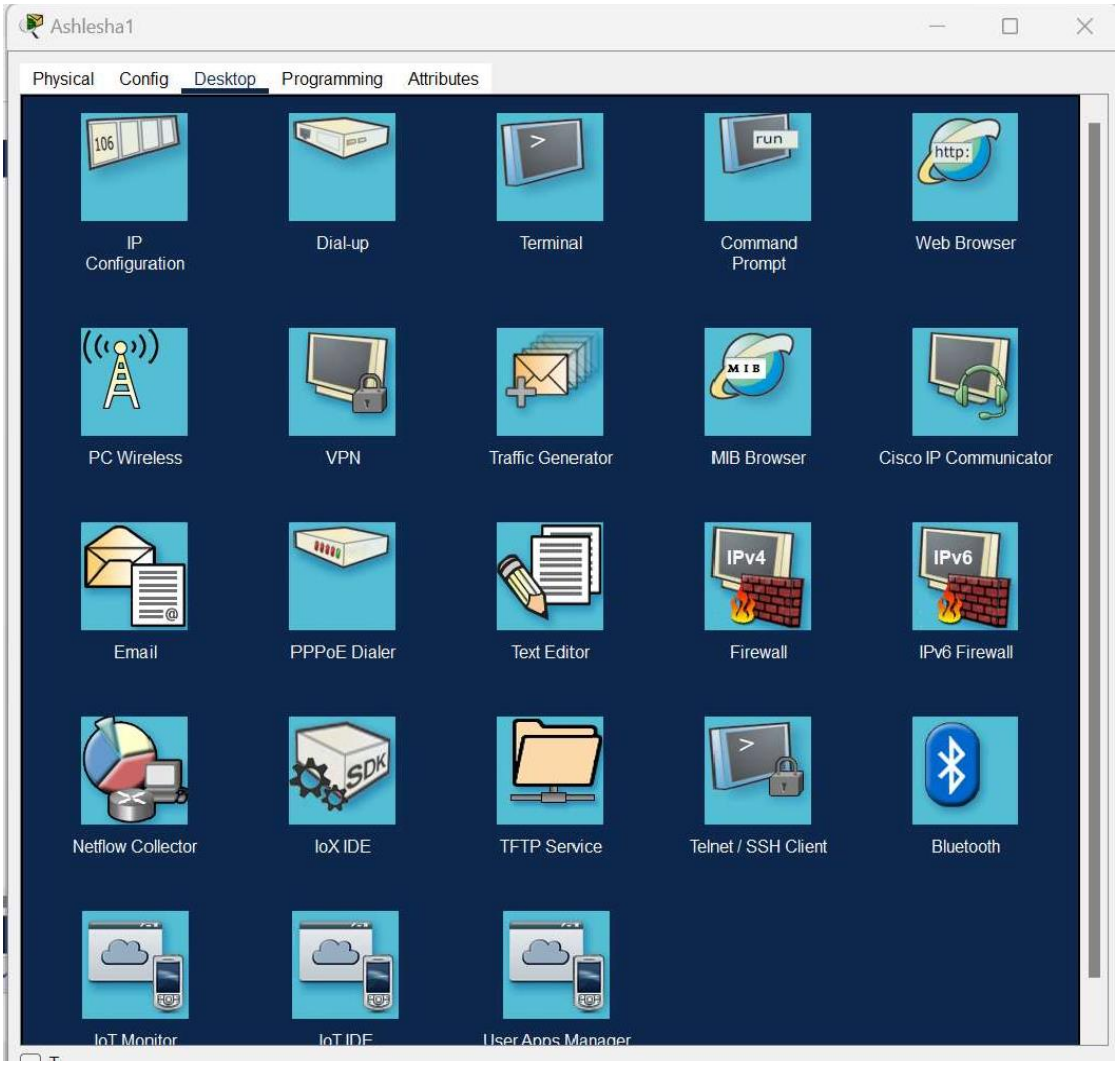


Fig 1.5: Network component icons and labels

Options and Preferences

The Options and Preferences menu allows you to customize your Packet Tracer environment. You can adjust visual settings like background color and font size, set default values for device configurations, and manage simulation speeds. This customization helps tailor the software to your personal preferences and working style, enhancing your overall experience.

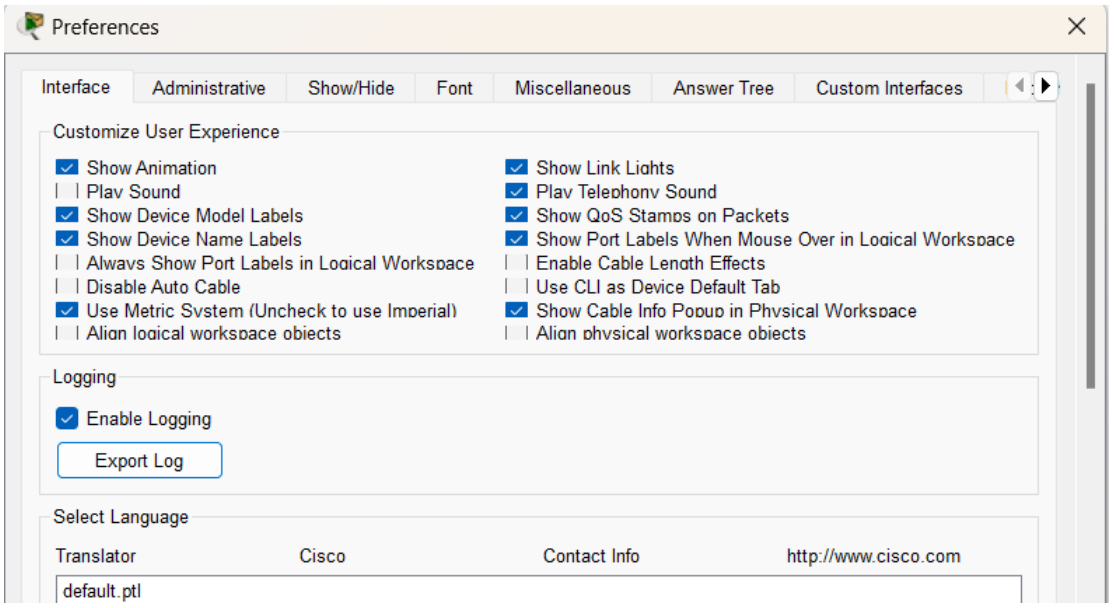


Fig 1.6: Option and preferences menu

Activity Wizard

The Activity Wizard is a feature designed for creating interactive learning activities within Packet Tracer. Instructors can use it to design tasks, provide step-by-step instructions, and set up assessments that Ashleshas can follow within the software. This tool is particularly useful for educational purposes, allowing Ashleshas to practice and test their networking skills in a guided and structured environment.

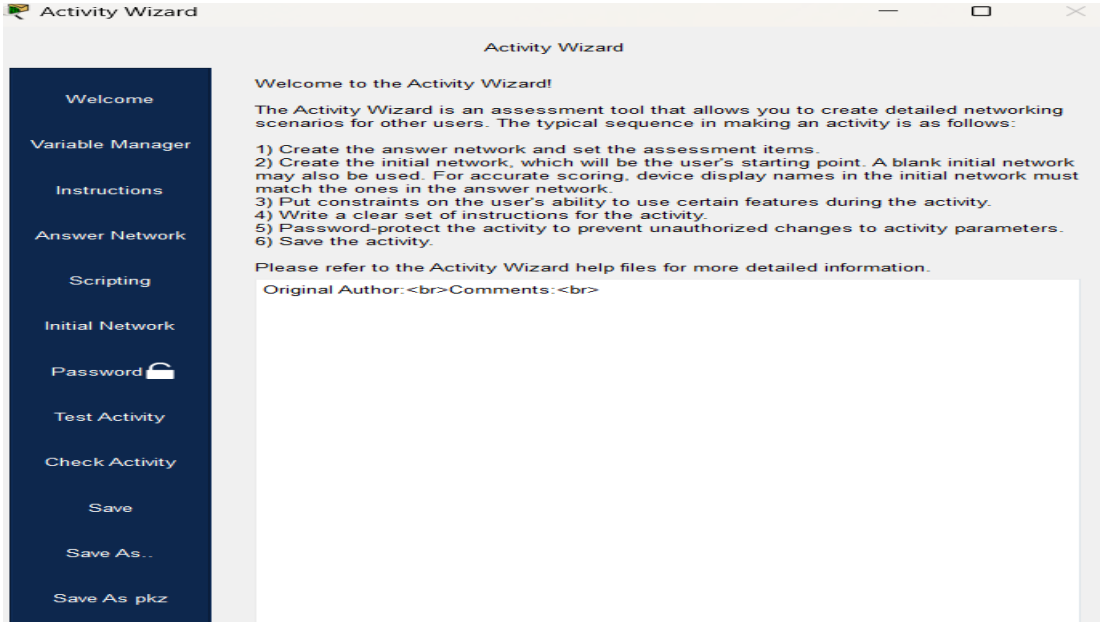


Fig 1.7: Activity wizard feature

Conclusion

In this lab, we utilized Cisco Packet Tracer, a crucial tool for students studying computer networks. It offers a secure and flexible platform for experimenting with different network setups. By familiarizing ourselves with its interface and features, we gained the ability to design and simulate a variety of network topologies. This hands-on experience allowed us to explore network configurations in a controlled environment, helping us build a solid foundation for understanding real-world networking concepts.

Lab 5: Creating a LAN and testing the connectivity using Packet Tracer

Theory

A Local Area Network (LAN) is a network that connects computers and devices within a limited geographical area, such as a home, office, or a building. LAN's are used to share resources like files, printers, and internet connections among multiple devices. Because LAN's cover smaller areas, they usually have higher data transfer speeds and lower latency compared to larger networks like WAN's (Wide Area Networks).

Key Features of LAN

1. It is a private network, so an outside regulatory body never controls it.

2. LAN operates at a relatively higher speed compared to other WAN systems.

3. There are various kinds of media access control methods like token ring and Ethernet.

4. It connects computers in a single building, block or campus (i.e. a restricted geographical area).

LAN Architecture

LAN architecture defines the structure, components, and communication protocols of a LAN. It involves several key elements:

Key Components

Switches and Hubs: Switches intelligently forward data to specific devices, while hubs broadcast data to all devices.

Ethernet Cables: Commonly used cables include Cat5e, Cat6, and fiber optics.

Network Interface Cards (NICs): NICs are hardware components that allow devices to connect to the Ethernet.

Topologies

The physical arrangement of devices in a LAN is called topology. Common topologies include:

1. Star topology

2. Bus topology,

3. Ring topology,

4. Mesh topology etc.

Protocols

LAN's use protocols to ensure smooth communication. Some common protocols include:
Ethernet: The most common protocol for wired LAN's.

Wi-Fi: Protocol for wireless LAN's.

TCP/IP: Suite of protocols for data communication.

Component Used

Hardware: Switches (1), Ethernet cables, End devices (4).

Software: Cisco Packet Tracer Network Diagram.

Network Diagram

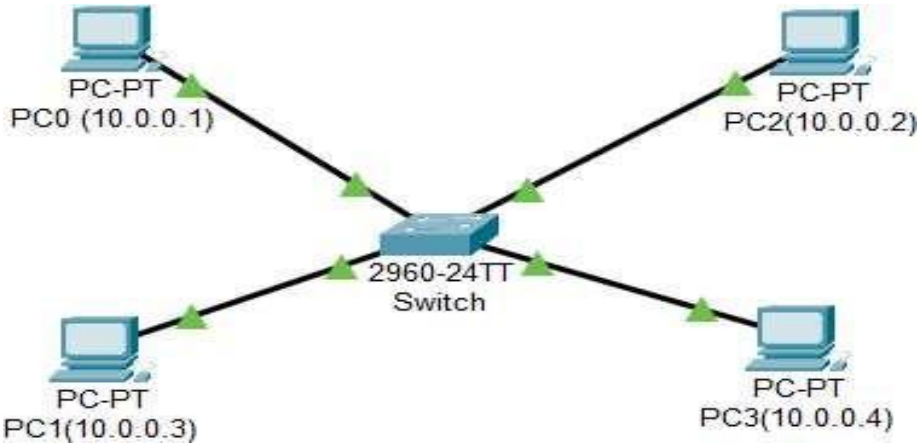


Fig: Network Diagram of LAN

Procedure:

Here is the procedure for creating the LAN network shown in the image using Cisco Packet Tracer:

Step 1: Launch Cisco Packet Tracer

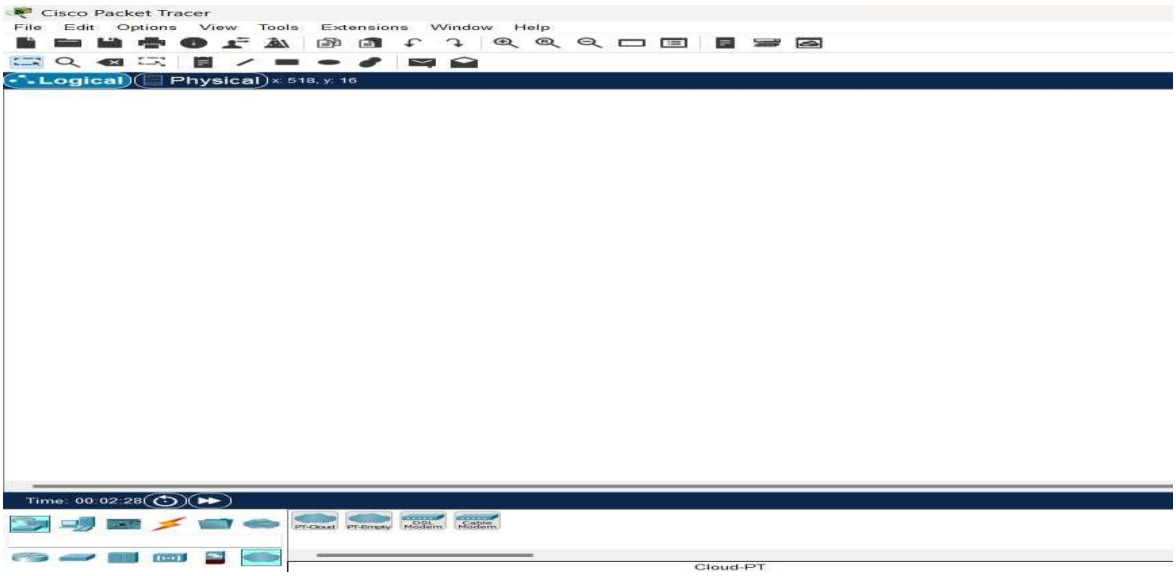


Fig: Workspace for network design

Step 2: Add the network devices to the workspace

2.1 From the Device-Type Selection box, choose the following devices and add them to the workspace:
2.2 One 2960-24TT Switch
2.3 Four PCs (labeled Ashlesha0, Ashlesha1, Ashlesha2, and Ashlesha3)

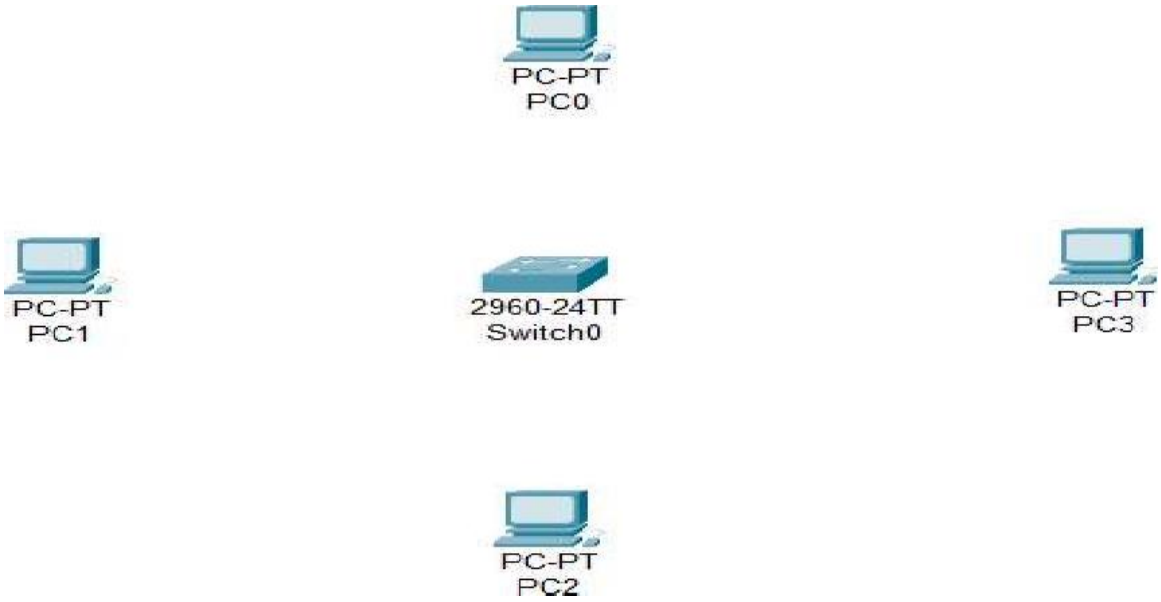


Fig: Switches and PC's for LAN Creation

Step 3: Connect the devices

3.1 Use the copper straight-through cable to connect each PC to one of the available ports on the switch.

3.2 Ensure that each connection is made properly.

3.3 Also renamed the PC's as Ashlesha0(10.0.0.1), Ashlesha1(10.0.0.3), Ashlesha2(10.0.0.2), and Ashlesha3(10.0.0.4)

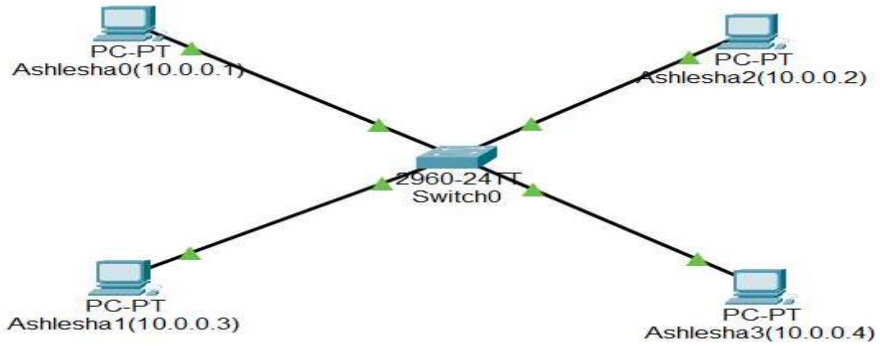


Fig: Connection between Switch and PC

Step 4: Configure IP addresses

4.1 Right-click on each PC and select "IP Configuration."

4.2 In the IP Configuration window, enter the IP address as (10.0.0.1 to 10.0.0.4) subnet mask, and default gateway for each PC.

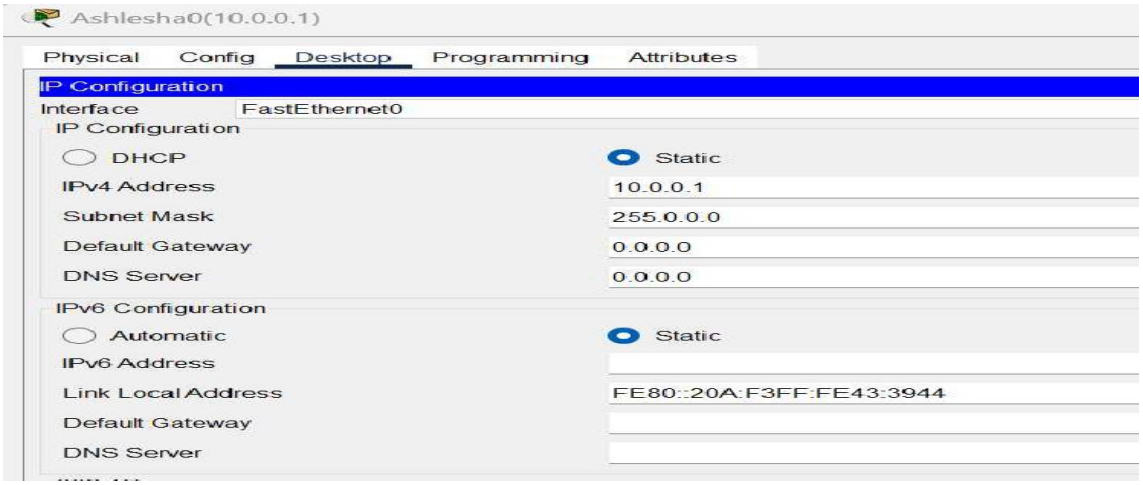


Fig: IP Configuration

Step 5: Testing and Validation

5.1 To test whether the network is working, you can ping other devices on the network from each PC.

5.2 Now ping Ashlesha0(10.0.0.1) from Ashlesha3(10.0.0.4) and vice-versa.

5.3 If the ping is successful, you should see replies from the other device.

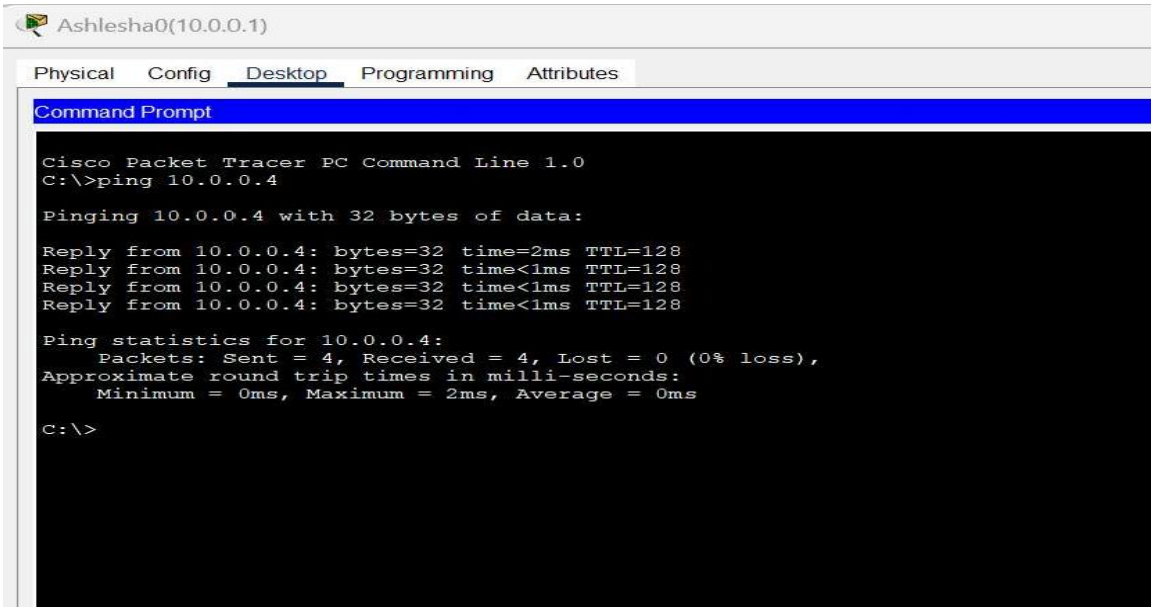


Fig: Connectivity test from Ashlesha0(10.0.0.1) to Ashlesha3(10.0.0.4)

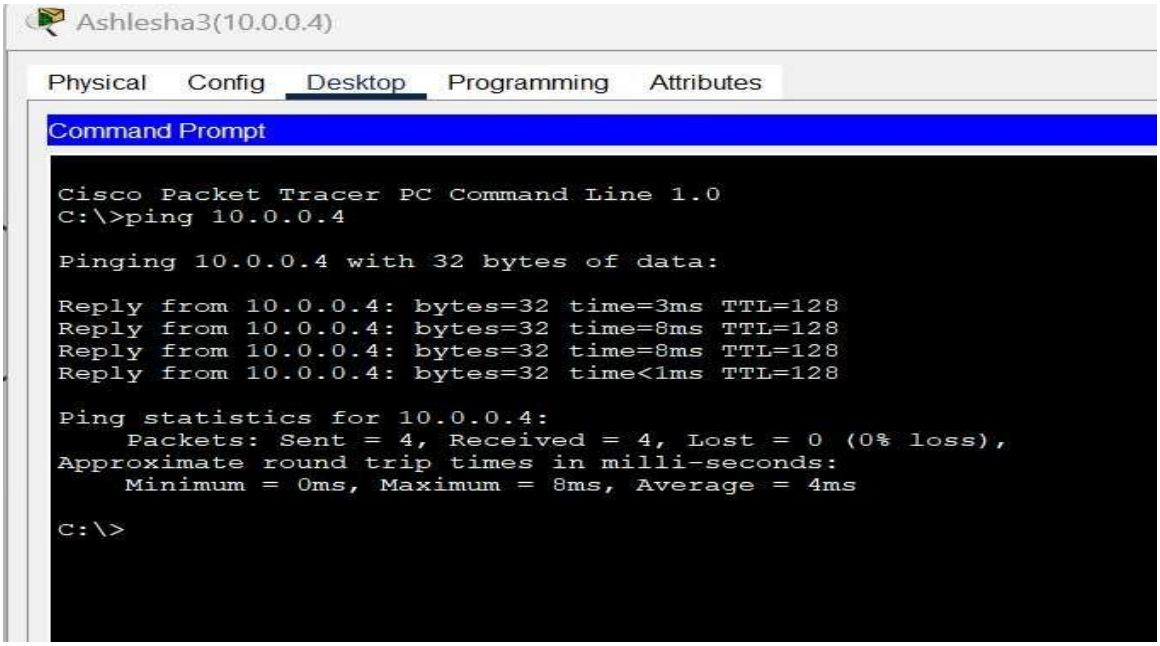


Fig: Connectivity test from Ashlesha3(10.0.0.4) to Ashlesha0(10.0.0.1)

Conclusion

In this lab, we built a Local Area Network (LAN) using Cisco Packet Tracer, providing valuable insights into how network devices interconnect and communicate. We simulated the design and setup of a LAN, incorporating key components such as routers, switches, and end devices. This allowed us to visualize and configure the network's structure effectively. By testing connectivity in the simulated environment, we ensure that network was functioning properly, allowing seamless data transfer between the devices.

Lab 6: Creating network topologies using Packet Tracer

Theory

Network topology describes the arrangement of devices and communication paths within a network, outlining both physical and logical structures. It plays a vital role in determining how data flows between devices. A strong grasp of various topologies is essential for effective network design, optimization, and troubleshooting.

Different types of Network Topologies:

1. Bus Topology
2. Ring Topology
3. Star Topology
4. Mesh Topology
5. Tree Topology
6. Hybrid Topology

Ring Topology

Ring topology is a network configuration where devices are connected in a circular manner, forming a closed loop. Each device is connected to exactly two other devices, creating a continuous pathway for data transmission.

Component Used

Hardware: Switches (4), Ethernet cables, End devices (4).
Software: Cisco Packet Tracer

Network Diagram

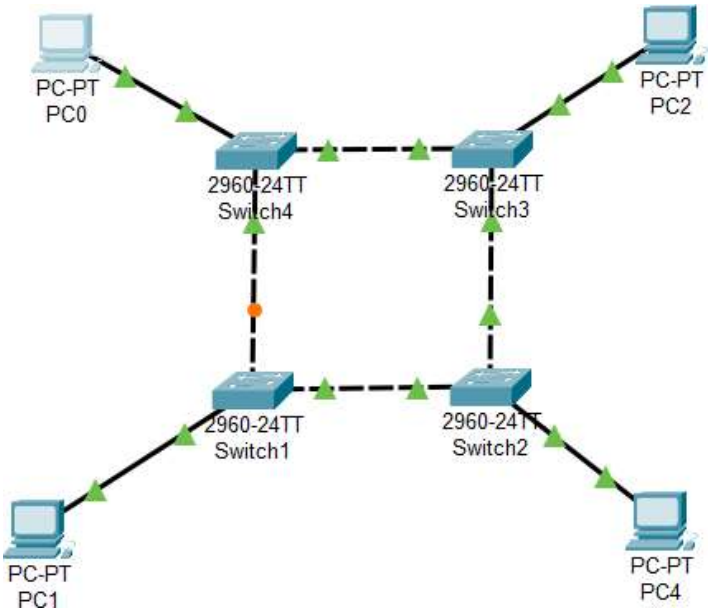


Fig: Network map for Ring Topology

Procedure

Here is the procedure for creating the Ring Topology shown in the image using Cisco Packet Tracer:

Step 1: Launch Cisco Packet Tracer

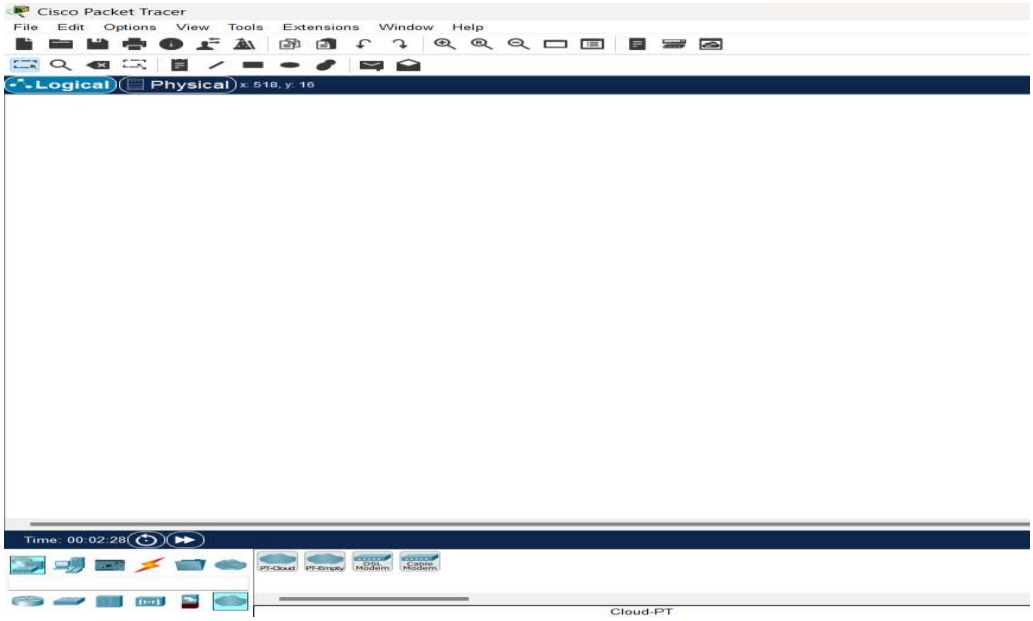


Fig: Workspace for network design

Step 2: Add the network devices to the workspace

2.1 From the Device-Type Selection box, choose the following devices and add them to the workspace:

2.2 Four 2960-24TT Switch

2.3 Four PCs (labeled PC0, PC1, PC2, and PC3)

Fig: Switches and PC's for Ring Topology

Step 3: Connect the devices

3.1 Use the copper straight-through cable to connect each PC to one of the available ports on the each switch and copper cross-over cable to connect between each adjacent switches.

3.2 Ensure that each connection is made properly.

3.3 Also renamed the PC's as PC0(10.0.0.1), PC1(10.0.0.2), PC2(10.0.0.4), and PC3(10.0.0.3)

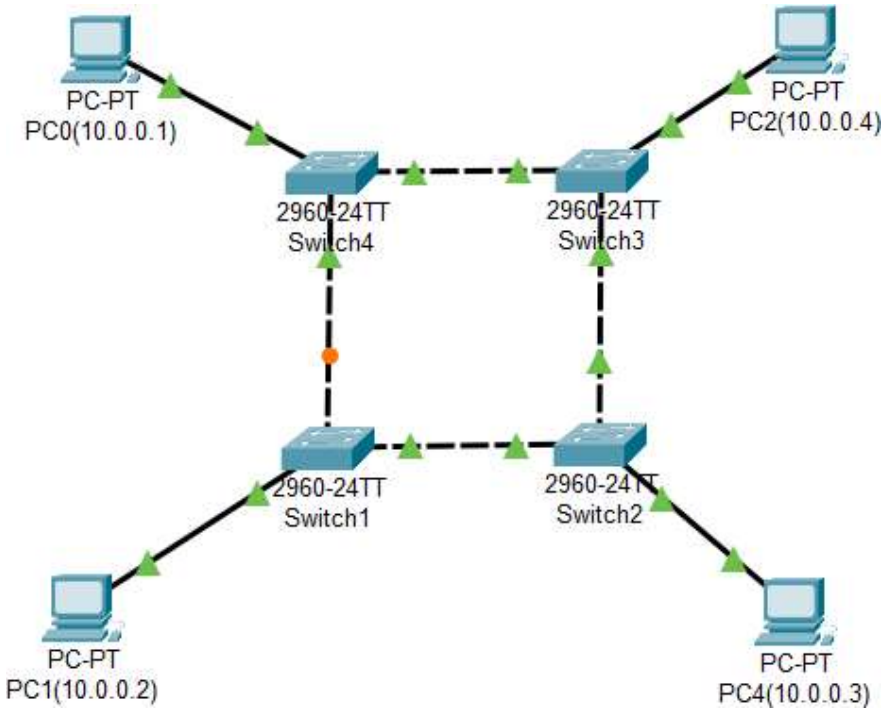


Fig: Connection between Switch and PC's

Step 4: Configure IP addresses

4.1 Right-click on each PC and select "IP Configuration."

4.2 In the IP Configuration window, enter the IP address as (10.0.0.1 to 10.0.0.4), subnet mask, and default gateway for each PC.

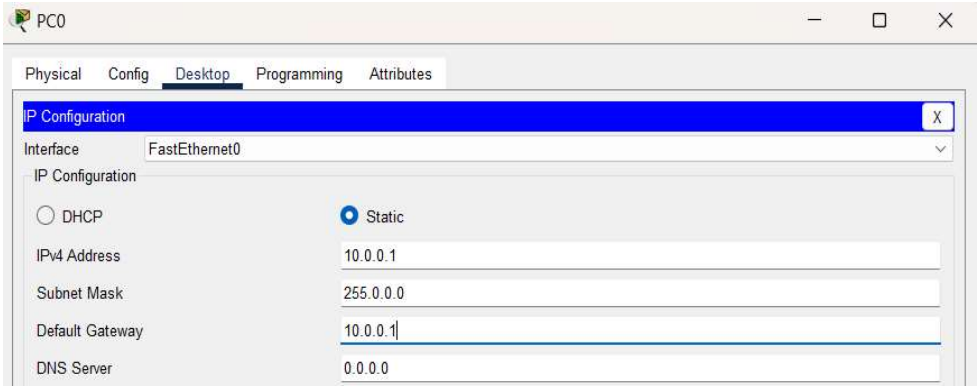


Fig: IP Configuration

Step 5: Verify connectivity:

5.1 To test whether the network is working, you can ping other devices on the network from each PC.

5.2 Now ping PC1(10.0.0.2) from PC2(10.0.0.4) and vice-versa.

5.3 If the ping is successful, you should see replies from the other device.

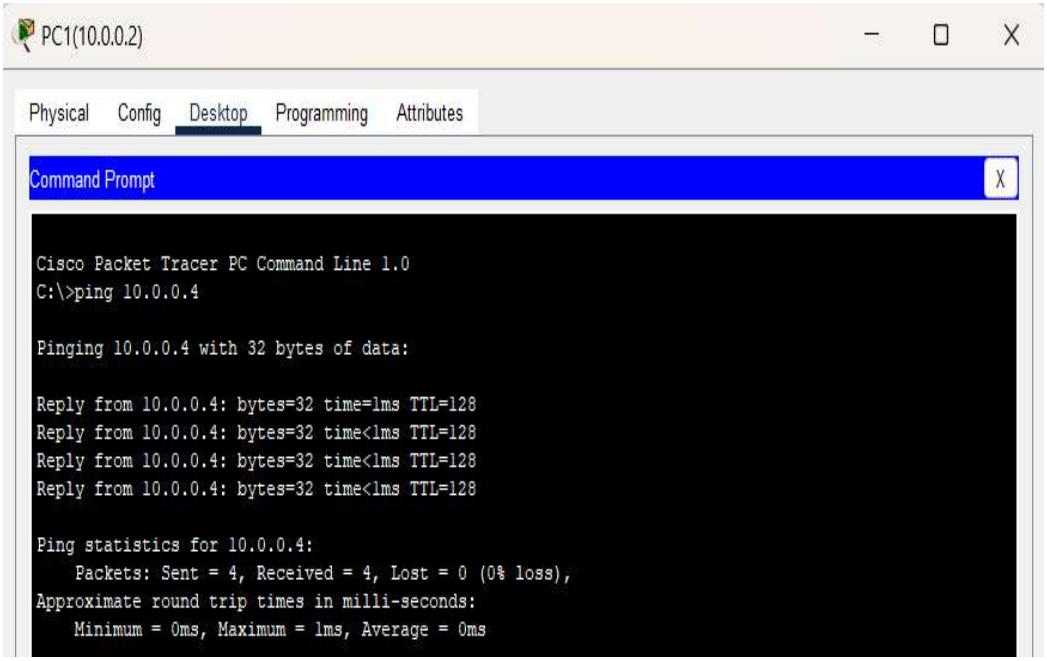


Fig: Connectivity test from PC1(10.0.0.2) to PC2(10.0.0.4)

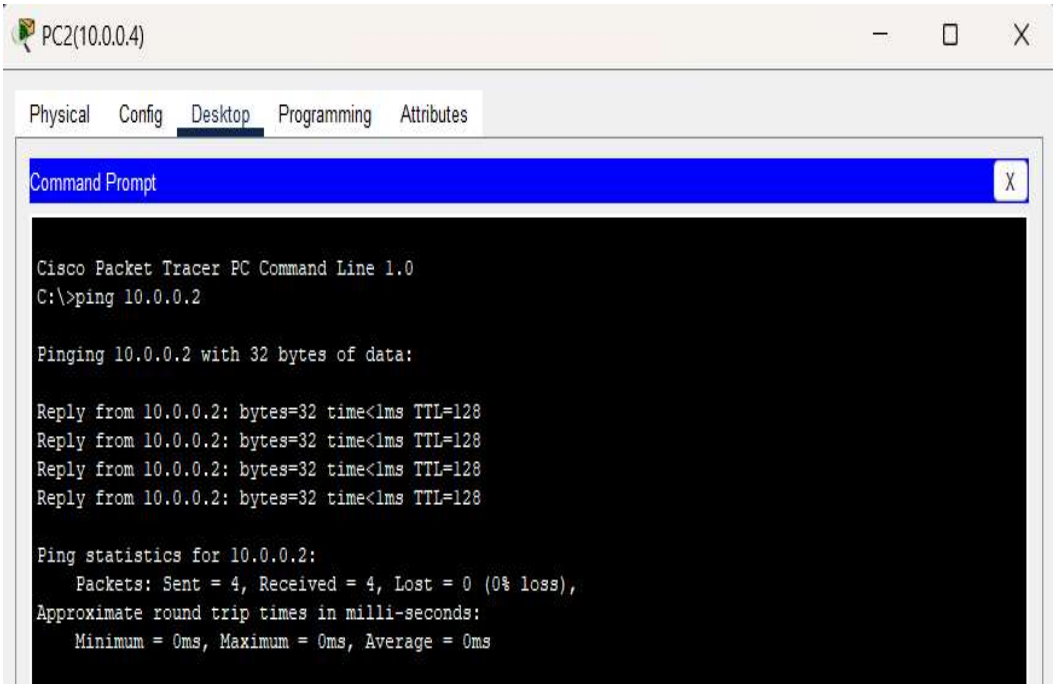


Fig: Connectivity test from PC2(10.0.0.4) to PC1(10.0.0.2)

Star Topology

Star topology is a network configuration where all devices are connected to a central hub or switch. This central device acts as a communication hub for all connected devices.

Component Used

Hardware: Switches (1), Ethernet cables, End devices (5).
Software: Cisco Packet Tracer

Network Diagram

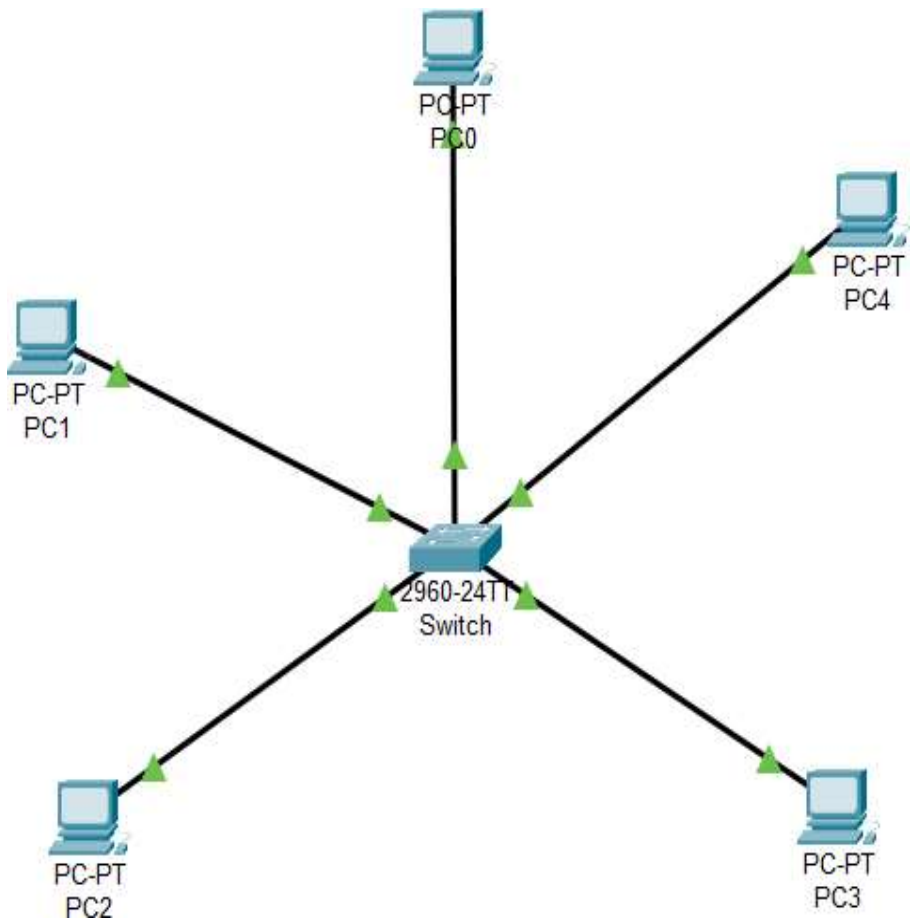


Fig: Network Map for Star Topology

Procedure

Here is the procedure for creating the Star Topology shown in the image using Cisco Packet Tracer:

Step 1: Launch Cisco Packet Tracer

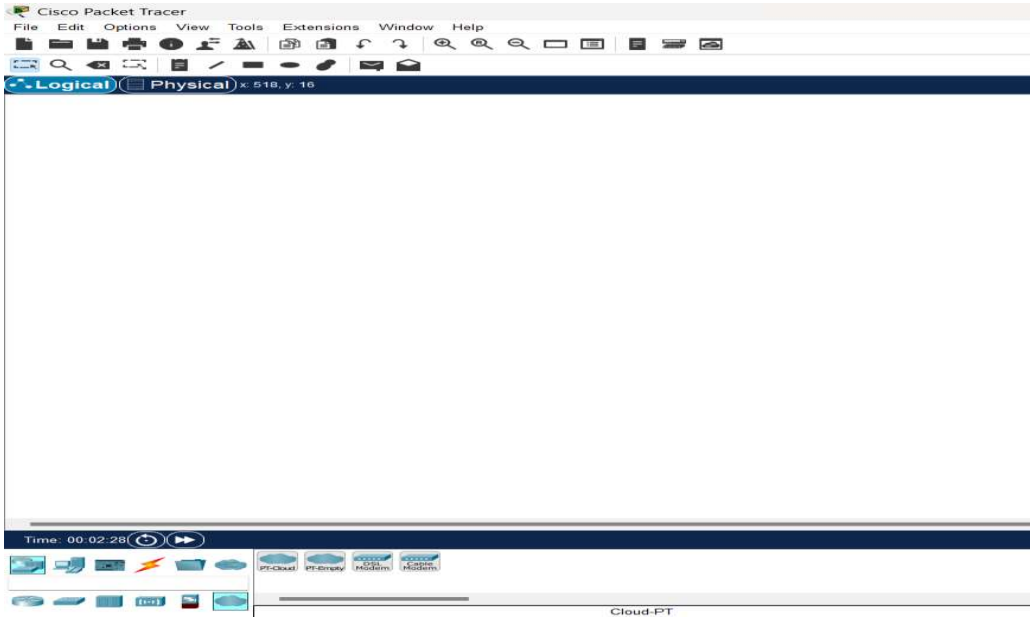


Fig: Workspace for network design

Step 2: Add the network devices to the workspace

2.1 From the Device-Type Selection box, choose the following devices and add them to the workspace:

2.2 One 2960-24TT Switch

2.3 Five PCs (labeled PC0, PC1, PC2, PC3and PC4)

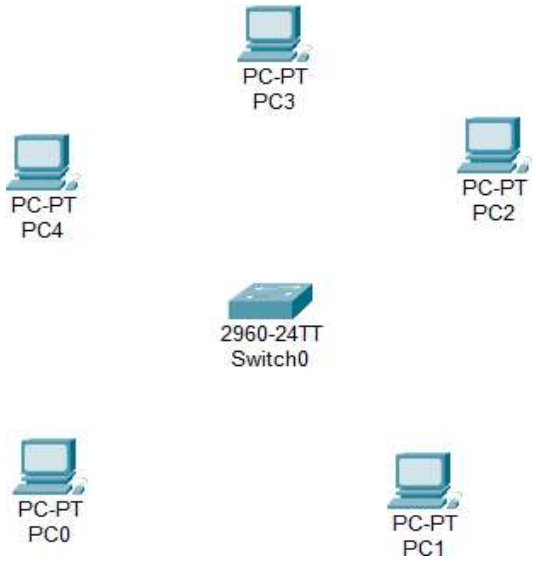


Fig: Switches and PC's for Star Topology

Step 3: Connect the devices

3.1 Use the copper straight-through cable to connect each PC to one of the available ports on the switch.

3.2 Ensure that each connection is made properly.

3.3 Also renamed the PC's as PC0(10.0.0.1), PC1(10.0.0.2), PC2(10.0.0.3), PC3(10.0.0.4) and PC4(10.0.0.5).

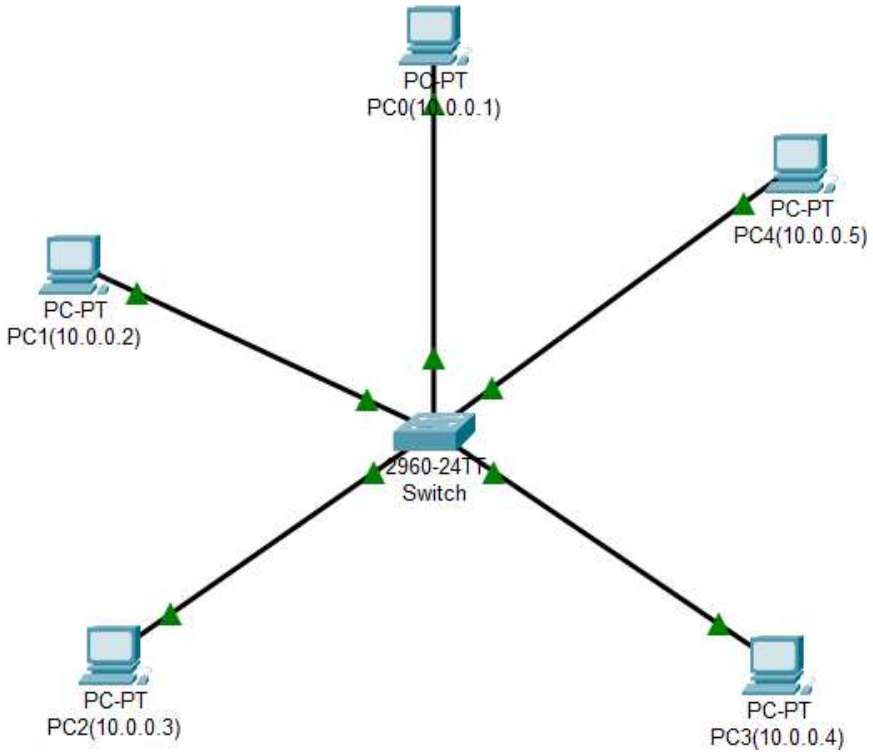


Fig: Connection between Switch and PC's

Step 4: Configure IP addresses

4.1 Right-click on each PC and select "IP Configuration."

4.2 In the IP Configuration window, enter the IP address as (10.0.0.1 to 10.0.0.5), subnet mask, and default gateway for each PC .

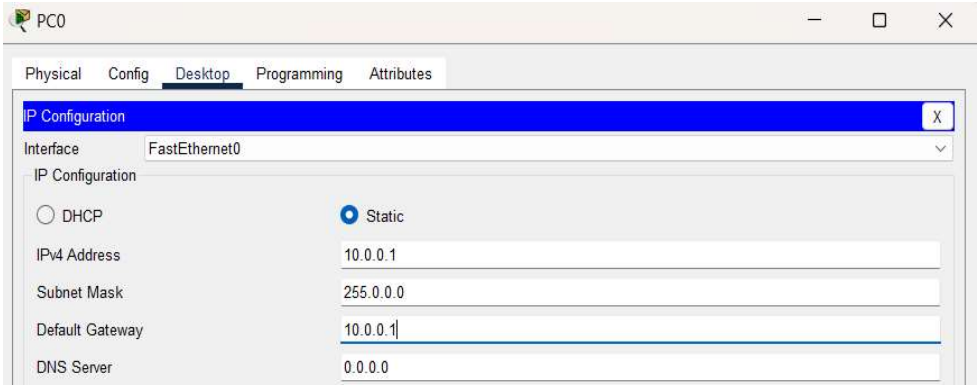


Fig: IP Configuration

Step 5: Verify connectivity:

5.1 To test whether the network is working, you can ping other devices on the network from each PC.

5.2 Now ping PC0(10.0.0.1) from PC2(10.0.0.3) and vice-versa.

5.3 If the ping is successful, you should see replies from the other device.

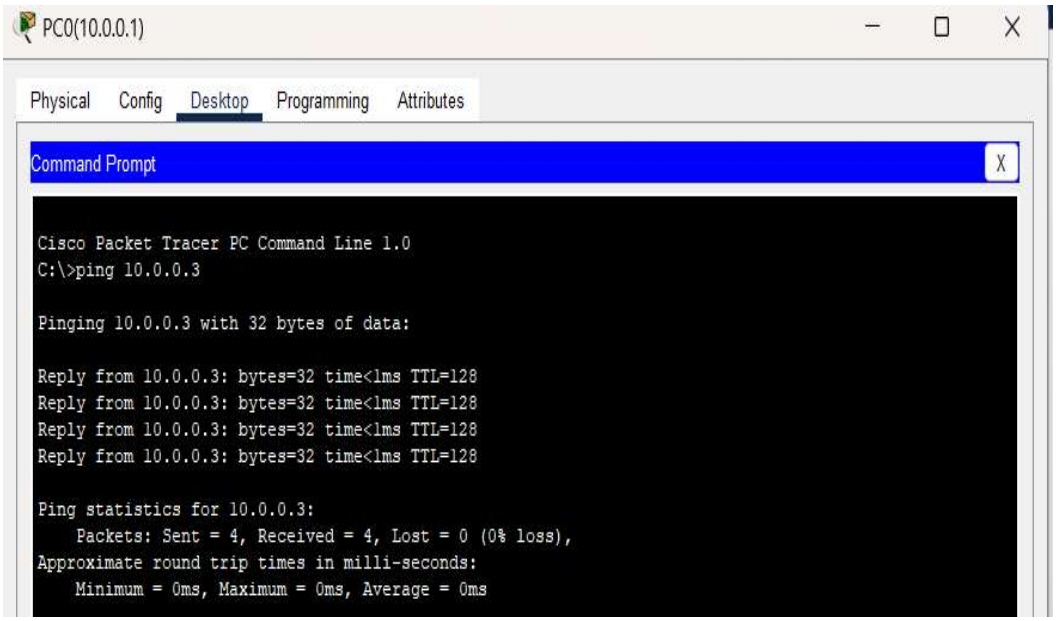


Fig: Connectivity test from PC0(10.0.0.1) to PC2(10.0.0.3)

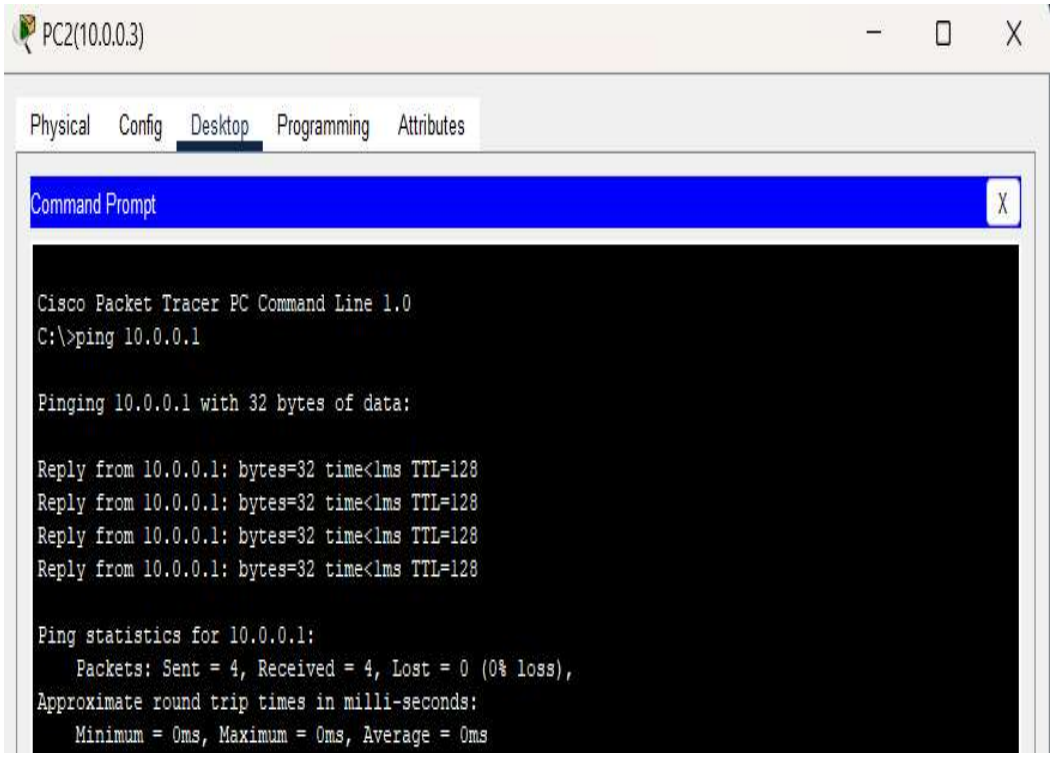


Fig: Connectivity test from PC2(10.0.0.3) to PC0(10.0.0.1)

Mesh Topology

Mesh topology is a network configuration where every device is connected to every other device. This creates a highly interconnected network with multiple paths for data transmission.

Component Used

Hardware: Switches (4), Ethernet cables, End devices (4).
Software: Cisco Packet Tracer

Network Diagram

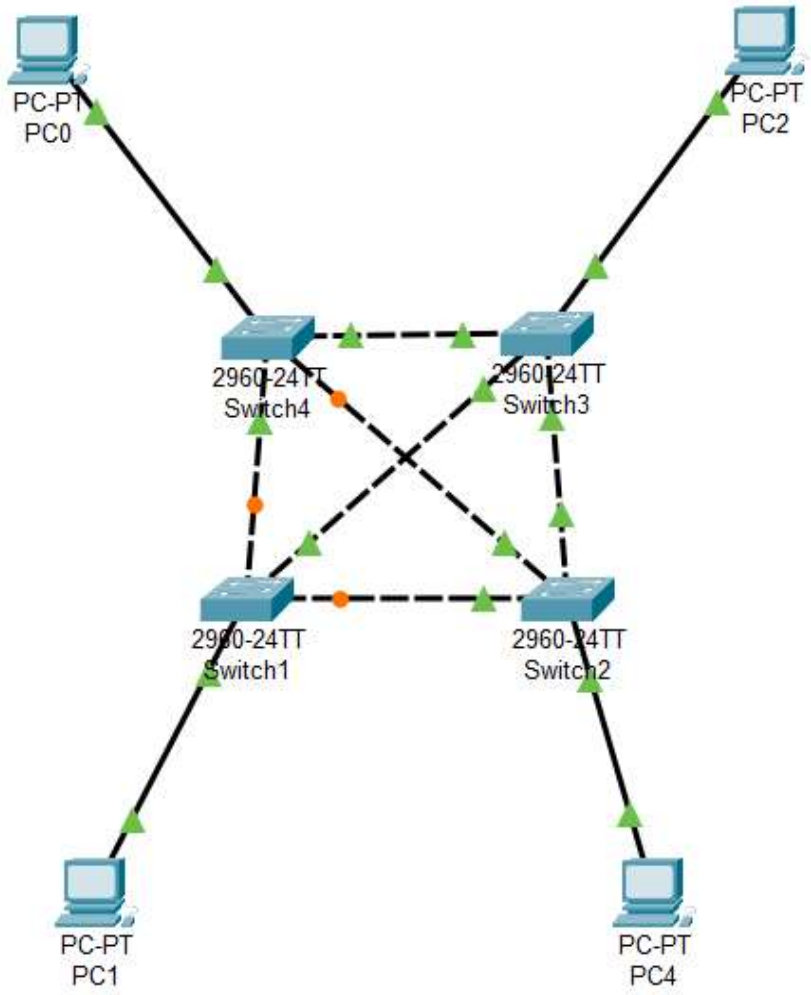


Fig: Network Map for Mesh Topology

Procedure

Here is the procedure for creating the Mesh Topology shown in the image using Cisco Packet Tracer:

Step 1: Launch Cisco Packet Tracer

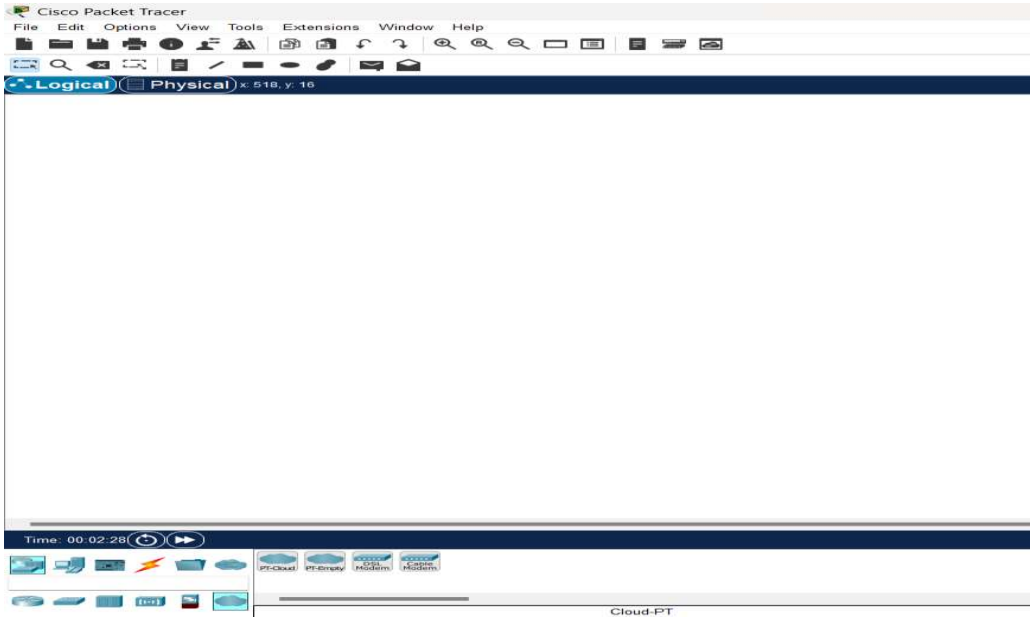


Fig: Workspace for network design

Step 2: Add the network devices to the workspace

2.1 From the Device-Type Selection box, choose the following devices and add them to the workspace:

2.2 Four 2960-24TT Switch

2.3 Four PCs (labeled PC0, PC1, PC2, and PC4)

Fig: Switches and PC's for Mesh Topology

Step 3: Connect the devices

3.1 Use the copper straight-through cable to connect each PC to one of the available ports on the each switch and copper cross-over cable to connect between each adjacent and diagonal switches.

3.2 Ensure that each connection is made properly.

3.3 Also renamed the PC's as PC0(10.0.0.1), PC1(10.0.0.2), PC2(10.0.0.3), PC4(10.0.0.4).

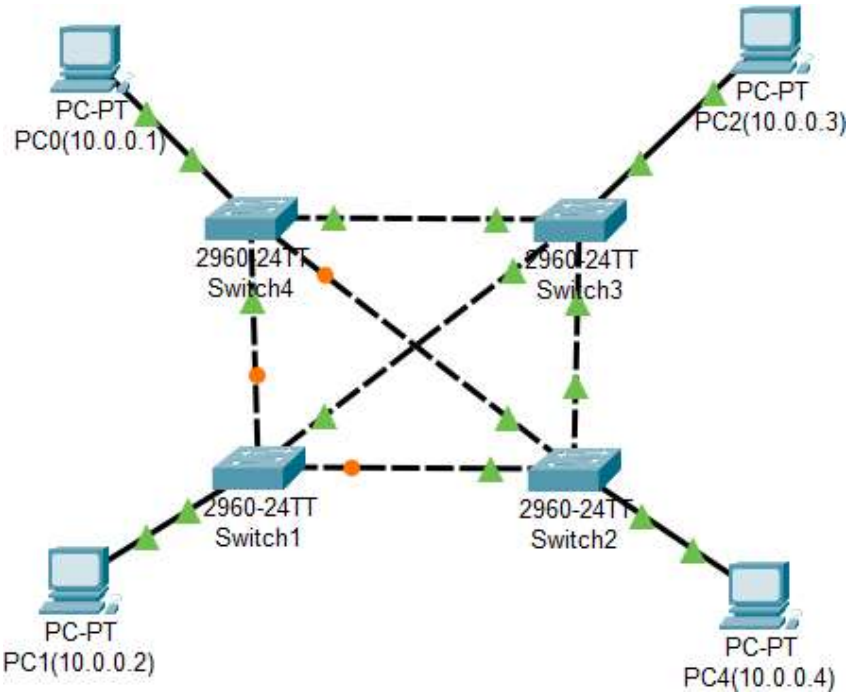


Fig: Connection between Switch and PC

Step 4: Configure IP addresses

4.1 Right-click on each PC and select "IP Configuration."

4.2 In the IP Configuration window, enter the IP address as (10.0.0.1 to 10.0.0.4), subnet mask, and default gateway for each PC .

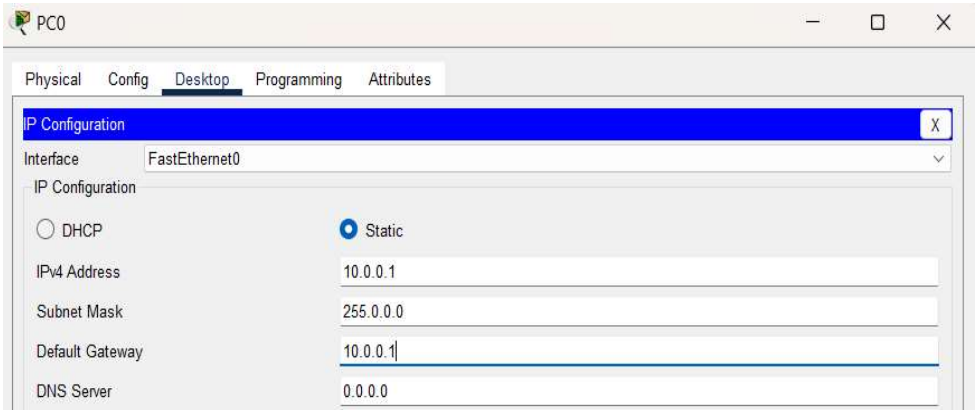


Fig: IP Configuration

Step 5: Verify connectivity:

5.1 To test whether the network is working, you can ping other devices on the network from each PC.

5.2 Now ping PC0(10.0.0.1) from PC2(10.0.0.3) and vice-versa.

5.3 If the ping is successful, you should see replies from the other device.

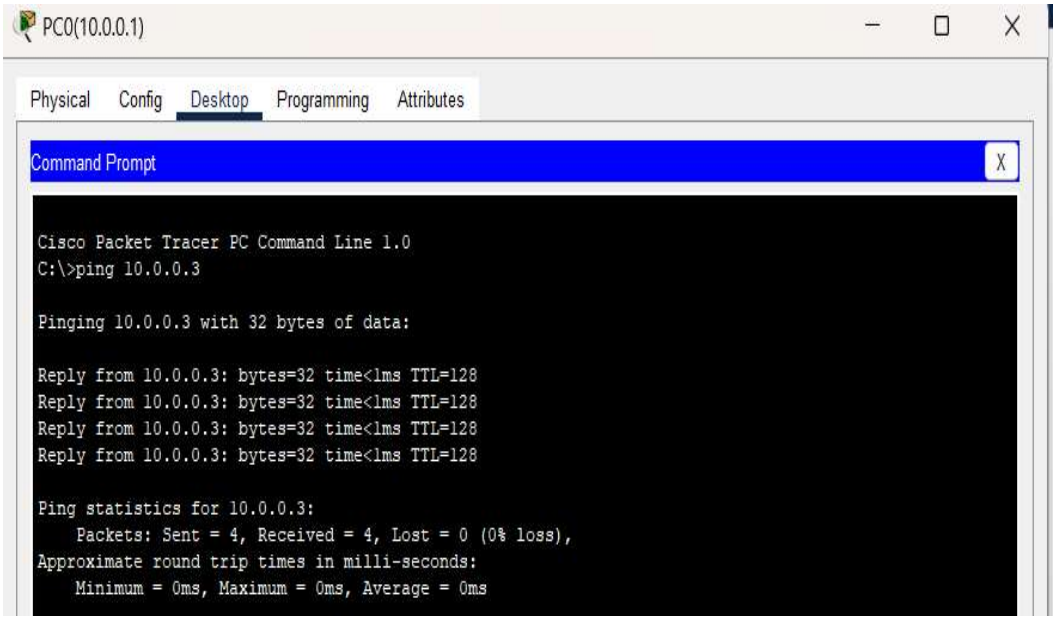


Fig: Connectivity test from PC0(10.0.0.1) to PC2(10.0.0.3)

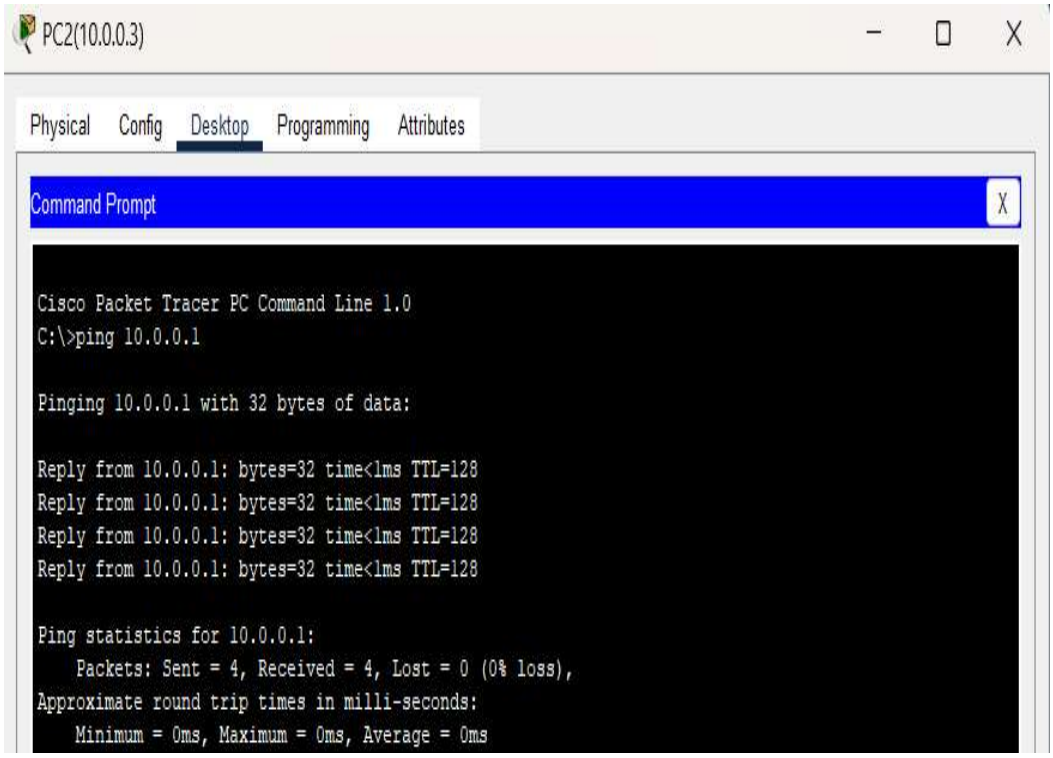


Fig: Connectivity test from PC2(10.0.0.3) to PC0(10.0.0.1)

Conclusion

In this lab, setting up various network topologies in Cisco Packet Tracer provided invaluable, hands-on experience in designing and configuring network structures. By simulating the placement and interconnection of essential components such as switches, routers, and end devices, we gained insights into how these elements interact within a network. This exercise allowed us to evaluate connectivity, troubleshoot issues, and refine configurations to ensure seamless data transmission and optimal network performance.

Lab 7: Creating VLAN and VLAN Trunking using Packet Tracer

Theory

A VLAN (Virtual Local Area Network) is a network configuration that segments a single physical network into multiple logical networks. Each VLAN acts like an independent network, even though multiple VLANs may share the same physical network infrastructure. VLANs improve network security, reduce broadcast traffic, and allow network administrators to segment traffic logically based on factors like department or function within an organization.

VLAN Trunking

VLAN trunking is a method used to allow traffic from multiple VLANs to traverse a single network link between switches or other network devices. This is achieved by tagging Ethernet frames with a VLAN identifier, commonly through IEEE 802.1Q tagging. Trunking enables the extension of VLANs across network devices, supporting greater flexibility in network design and allowing VLANs to span across different physical locations.

VLAN Architecture

VLAN architecture is designed to logically group devices across different network segments, creating multiple broadcast domains on a single network infrastructure. Each VLAN typically corresponds to a different logical network, isolating traffic between VLANs unless explicitly allowed through routing or firewall rules. The architecture includes components like access ports (where devices are connected to the VLAN), trunk ports (which carry traffic for multiple VLANs), and VLAN-aware network devices that manage traffic across various segments.

Component Used

Hardware: Switches (2), Ethernet cables, End devices (4).

Software: Cisco Packet Tracer

Network Diagram:

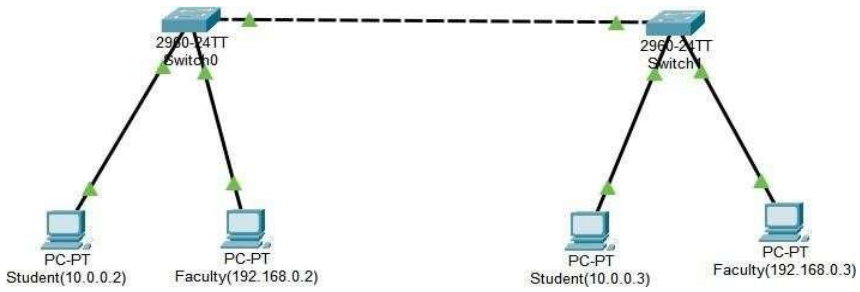


Fig: Network map for VLAN

Procedure:

Here is the procedure for creating the LAN network shown in the image using Cisco Packet Tracer:

Step 1: Launch Cisco Packet Tracer

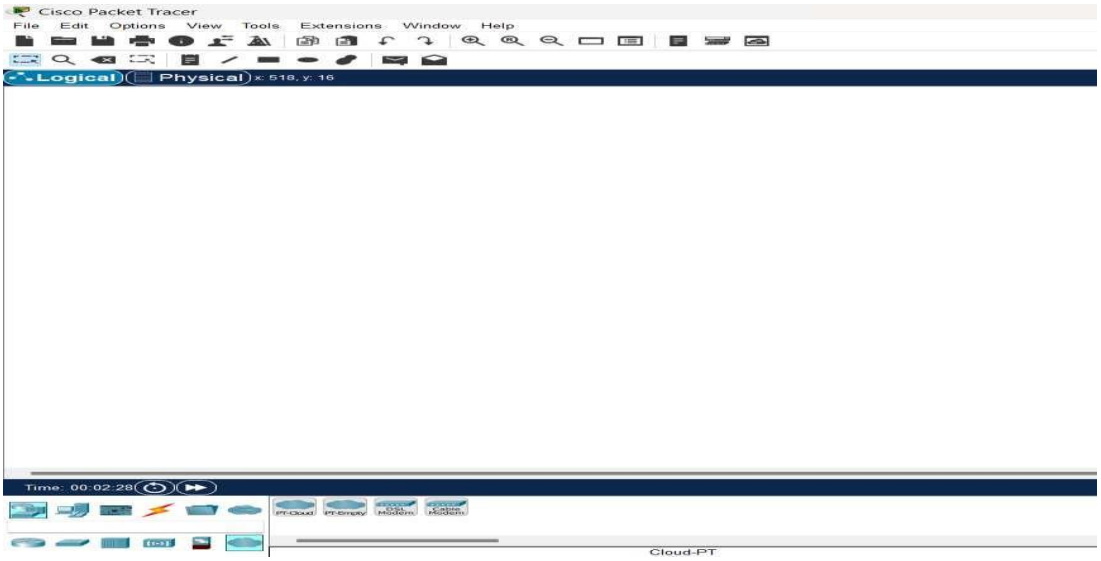


Fig: Workspace for network design

Step 2: Add the network devices to the workspace and connecting devices:

2.1 From the Device-Type Selection box, choose the following devices and add them to the workspace.

2.2 One 2960-24TT Switch and four PC's

2.3 Use the copper straight-through cable to connect each PC to one of the available ports on the switch.

2.4 Ensure that each connection is made properly.

2.5 Also renamed the PC's as Ashlesha (10.0.0.2), Faculty (192.168.0.2), Ashlesha (10.0.0.3), and Faculty (192.168.0.3).

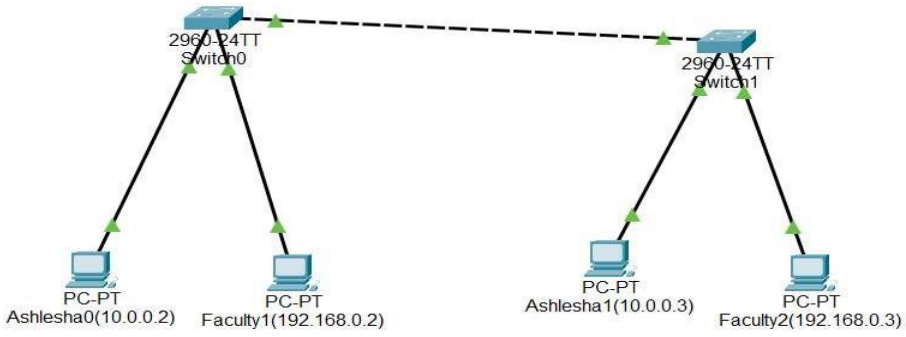


Fig: Connection between all devices in VLAN

Step 3: Configure IP addresses:

3.1 Right-click on each PC and select "IP Configuration."

3.2 In the IP Configuration window, assign IP addresses as follows: Ashlesha - devices 10.0.0.2 and 10.0.0.3; Faculty devices - 192.168.0.2 and 192.168.0.3. Ashlesha devices will connect via port Fa 0/1, while Faculty devices will connect via port Fa 0/2.

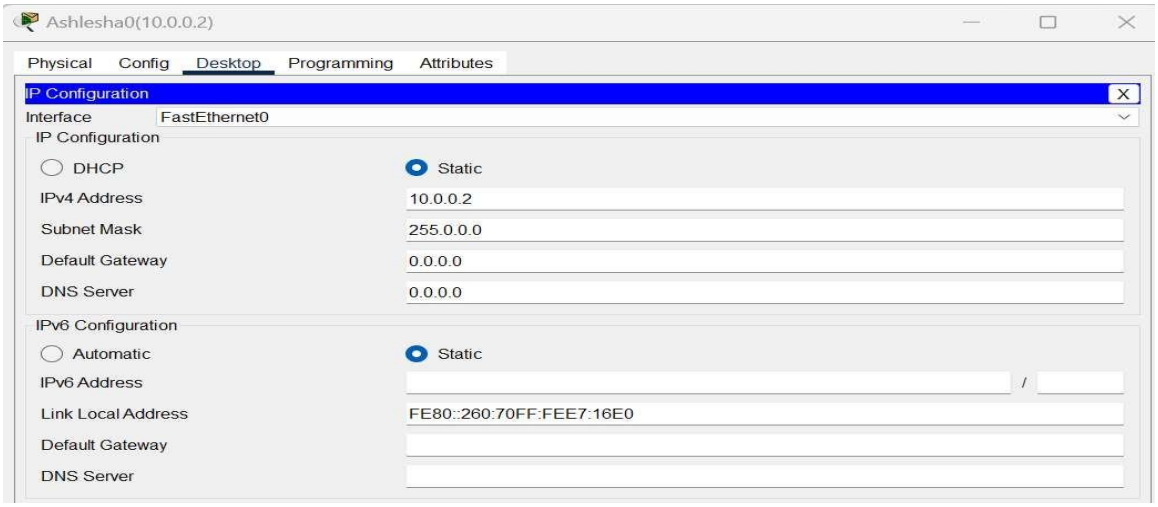


Fig: IP Configuration

Step 4: Configuring VLANs:

4.1 Create VLAN on Both Switches & Assign Port to Both Switches:

To create VLANs on both switches, enter the configuration mode on each switch and use the vlan command to create separate VLANs for Ashlesha and Faculty devices. After creating the VLANs, assign ports to the VLANs by selecting the specific interfaces (e.g., Fa 0/1 for Ashlesha and Fa 0/2 for Faculty) and using the switchport access vlan command to associate the ports with the correct VLAN.

4.2Create Trunking on Both Switches:

To enable trunking, configure the interfaces between the two switches using the switchport mode trunk command. This allows multiple VLANs to pass through the same link, making communication between devices in the same VLAN but connected to different switches. Trunking ensures that tagged traffic is carried across the switches while maintaining the VLAN distinctions.

Code for VLAN configurations:

```
Switch(config)#vlan 10
```

Switch(config-vlan)#name Ashlesha

Switch(config-vlan)#vlan 20

Switch(config-vlan)#name faculty

Switch(config-vlan)#exit

Switch(config)#exit

Code for Assigning ports:

Switch#config t

Switch(config)#int fa 0/1

Switch(config-if)#switchport mode access

Switch(config-if)#switchport access vlan 10

Switch(config-if)#int fa 0/2

Switch(config-if)#switchport mode access

Switch(config-if)#switchport access vlan 20

Switch(config-if)#exit

Switch(config)#exit

Code for Trunking Switches:

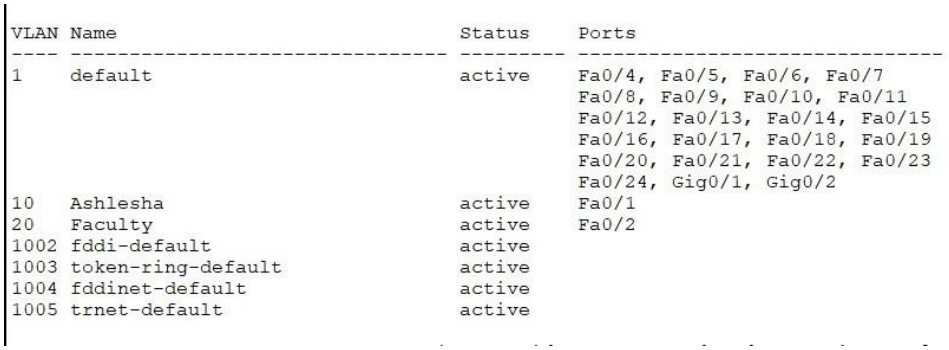
Switch#config t

Switch(config)#int fa 0/3

Switch(config-if)#switchport mode trunk

Switch(config-if)#exit

Switch(config)#exit



VLAN	Name	Status	Ports
1	default	active	Fa0/4, Fa0/5, Fa0/6, Fa0/7 Fa0/8, Fa0/9, Fa0/10, Fa0/11 Fa0/12, Fa0/13, Fa0/14, Fa0/15 Fa0/16, Fa0/17, Fa0/18, Fa0/19 Fa0/20, Fa0/21, Fa0/22, Fa0/23 Fa0/24, Gig0/1, Gig0/2
10	Ashlesha	active	Fa0/1
20	Faculty	active	Fa0/2
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

Fig: Assigning ports to VLAN

```
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int fa 0/3
Switch(config-if)#switchport mode trunk

Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed state to up

Switch(config-if)#exit
Switch(config)#exit
Switch#
```

Fig: Configuring trunking between switches

Step 5: Verify connectivity

5.1 To test whether the network is working, you can ping other devices on the network from each PC.

5.2 Now ping Ashlesha (10.0.0.2) from student (10.0.0.3) and vice-versa.

5.3 Also ping Ashlesha (10.0.0.2) from faculty (198.68.0.2) to check there is no connection Ashlesha and faculty.

5.3 If the ping is successful, you should see replies from the other device.

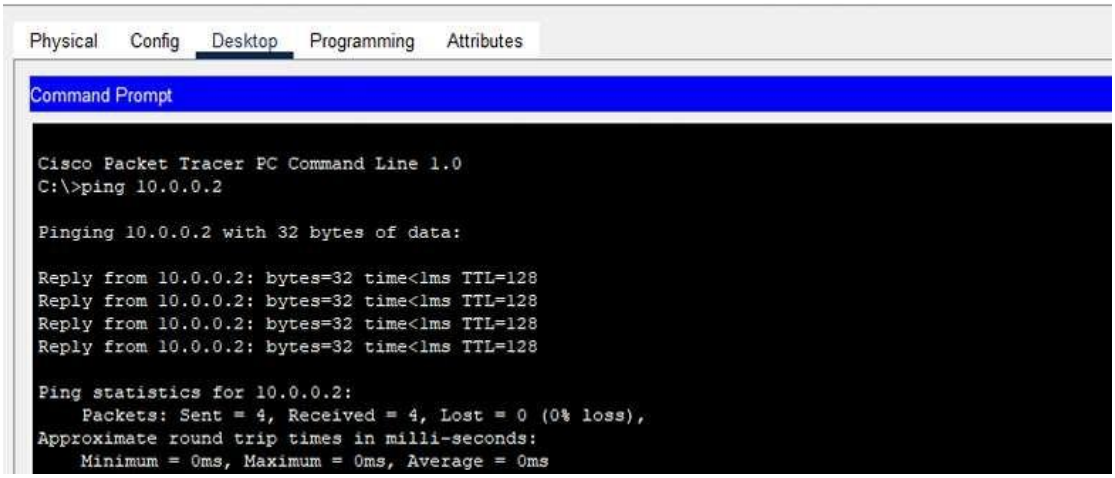


Fig: Connectivity test from Ashlesha (10.0.0.3) to Ashlesha (10.0.0.2)

```
C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=19ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 19ms, Average = 5ms
```

Fig: Connectivity test from Ashlesha (10.0.0.2) to Ashlesha (10.0.0.3)

```
C:\>ping 192.168.0.2

Pinging 192.168.0.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Fig: Connectivity test from Ashlesha (10.0.0.2) to faculty (192.168.0.2)

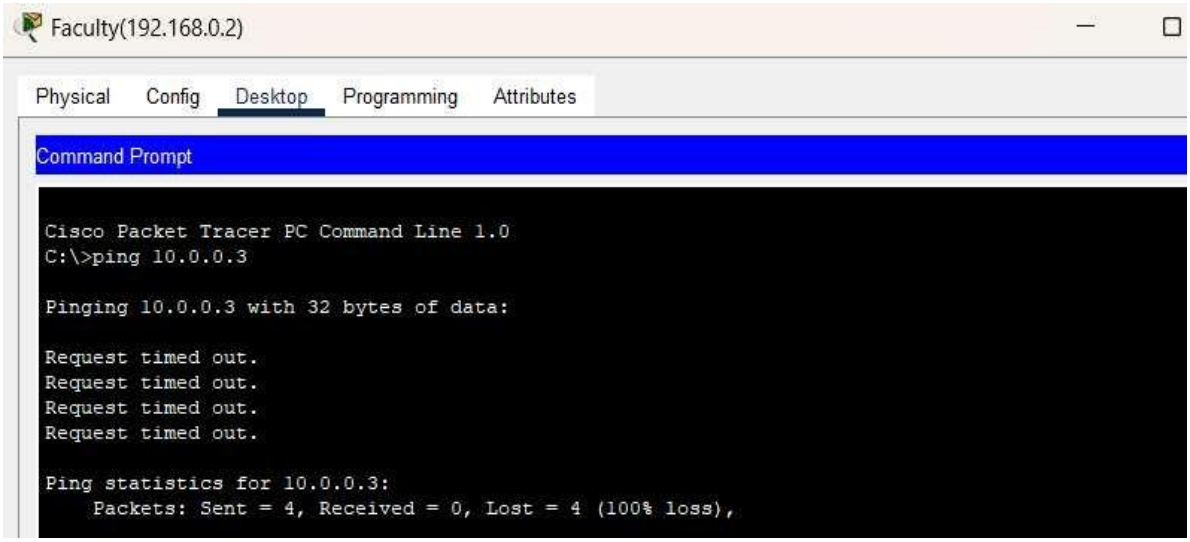


Fig: Connectivity test from faculty (192.168.0.2) to Ashlesha (10.0.0.3)

Conclusion

In summary, setting up VLANs and utilizing VLAN trunking in Cisco Packet Tracer improves network segmentation and management. Grouping devices into separate VLANs reduces broadcast domains, enhances security, and boosts network performance. VLAN trunking, which enables the transfer of multiple VLANs over a single link, ensures smooth communication between VLANs across different switches. This method emphasizes the value of organized network design, minimizing broadcast traffic and simplifying management, while promoting scalability and efficiency in modern networks.

Lab 8: Basic router configuration and static routing in Packet Tracer

Theory

A router is a networking device that forwards data packets between computer networks, performing traffic directing on the Internet. Data sent across the Internet, such as a web page or email, is in the form of data packets. A router is connected to at least two networks and decides which way to send each information packet based on its understanding of the networks it's connected to.

Network Diagram

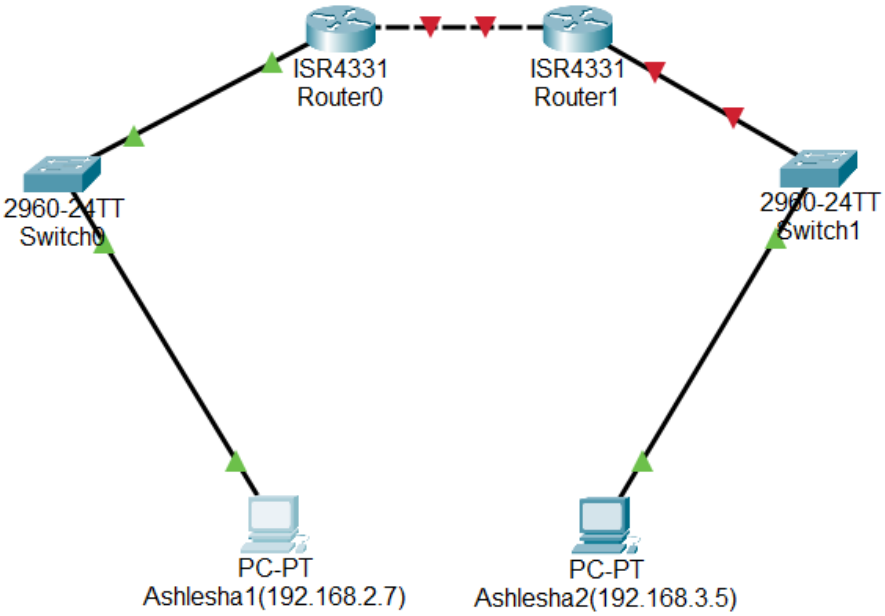


Fig: Network diagram

Basic Router Configuration

Configuring Global Parameters

The initial configuration of the router involves setting global parameters such as hostname, passwords, and interface descriptions.

Steps:

1.Enter privileged EXEC mode.
2.Configure the hostname using the hostname command.
3.Set passwords for privileged EXEC mode and console access.
4.Configure banners if necessary.
5.Repeat same for another router as well.

```
         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: n


Press RETURN to get started!



Router>enable
Router#config ter
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Ashlesha
Ashlesha(config)#enable secret xyz
Ashlesha(config)#no ip domain-lookup
Ashlesha(config)#exit
Ashlesha#
%SYS-5-CONFIG_I: Configured from console by console
```

Fig: Router configuration

Configuring Gigabit Ethernet

Once global parameters are set, configure the Gigabit Ethernet interfaces of the router to enable communication between different networks.

Steps:

1.Access the interface using the interface gig0/0 command.

2.Set the IP address and subnet mask for the interface.

3.Enable the interface using the no shutdown command.

4.Repeat same for another router as well.

```
Ashlesha# config ter
Enter configuration commands, one per line.  End with CNTL/Z.
Ashlesha(config)#interface GigabitEthernet0/0/0
Ashlesha(config-if)#no shutdown

Ashlesha(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to up
 ip address 192.168.1.2 255.255.255.0
Ashlesha(config-if)# ip address 192.168.1.2 255.255.255.0
Ashlesha(config-if)#
Ashlesha(config-if)#exit
Ashlesha(config)#
```

Fig: Gigabit Ethernet configuration

Connection Testing Before Static Routing Configuration

Steps:

1.Pinging PC(Ashlesha2(192.168.3.5)) from PC(Ashlesha1(192.168.2.7)) to verify connection if exists.

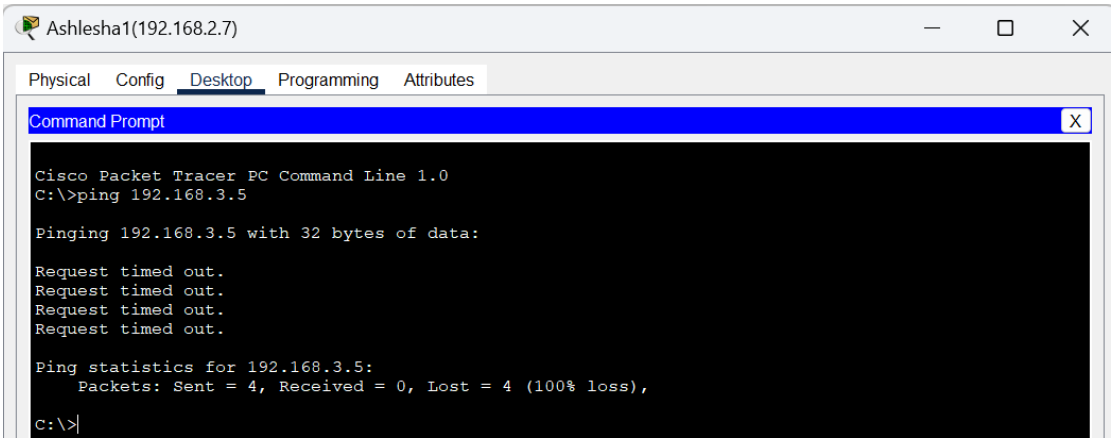


Fig: Connectivity test from PC(Ashlesha1(192.168.2.7)) to PC(Ashlesha2(192.168.3.5))

Here we can see there is not any connection in the network. So to establish connection in the network ,we need to statically configure the router through CLI.

Static Routing Configuration

Configuring Network (PCs and Routers)
Set up static routes to allow the routers to communicate with networks beyond their directly connected networks.
Steps:
1.Configure the IP addresses of PCs connected to each network.

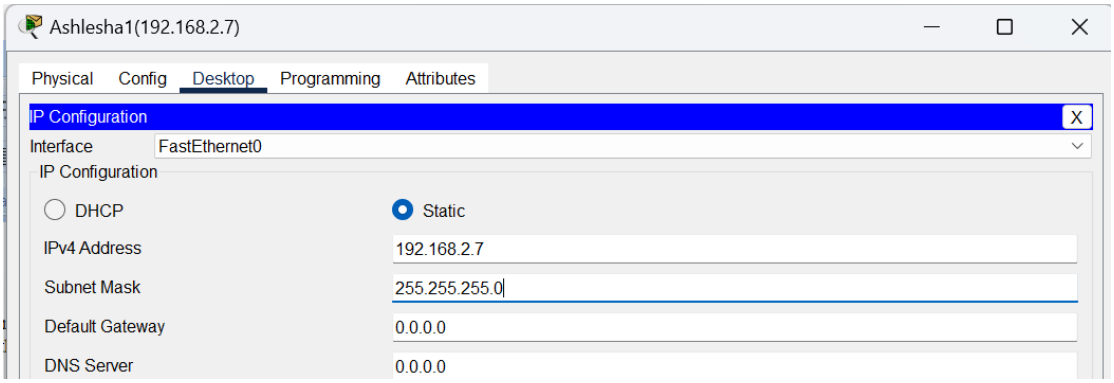


Fig: IP configuration PC(Ashlesha1(192.168.2.7))

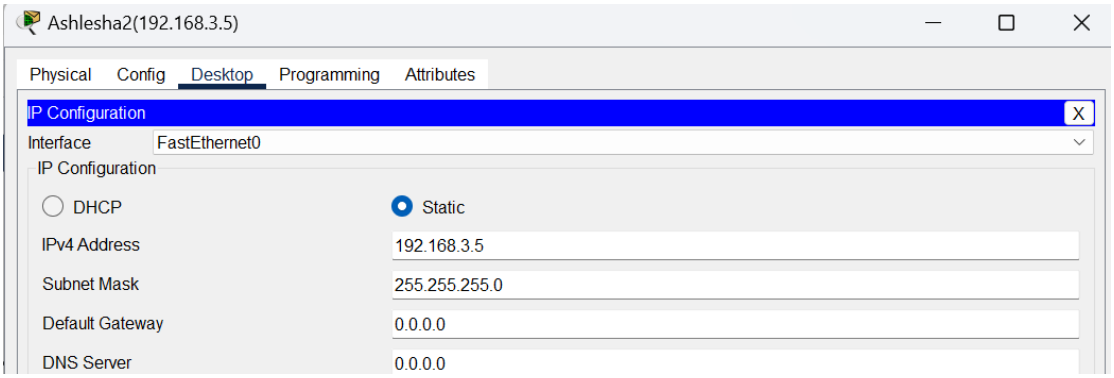


Fig: IP configuration on PC(Ashlesha2(192.168.3.5))

2.On each router, configure static routes using the ip route command to manually specify the next hop for network traffic.

```
Ashlesha(config)#ip route 192.168.2.0 255.255.255.0 192.168.1.2
```

Fig: Configuring IP route on router (Ashlesha1)

```
Ashlesha(config)#ip route 192.168.1.2 255.255.255.0 192.168.1.4
```

Fig: Configuring IP route on router (Ashlesha2)

Testing and Validation

To test whether the network is working, you can ping other devices on the network from each PC.

Steps:

1. Ping PC(Ashlesha2(192.168.3.5)) from PC(Ashlesha1(192.168.2.7)).
2. If the ping is successful, you should see replies from the other device.

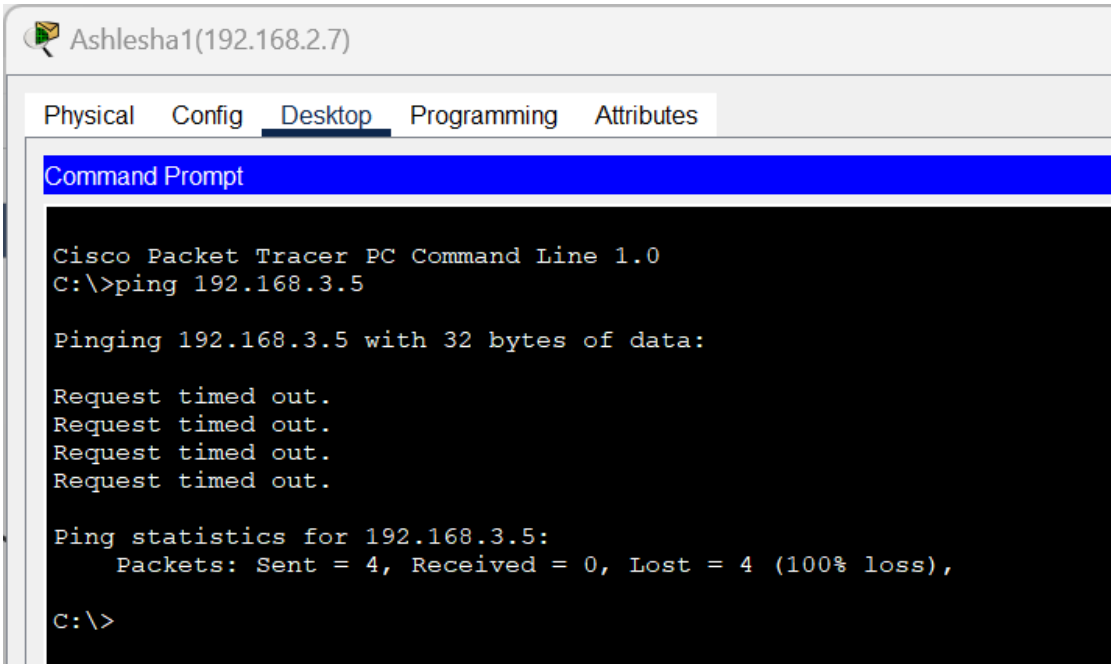


Fig: Connectivity test from PC(Ashlesha1(192.168.2.7)) to PC(Ashlesha2(192.168.3.5))

Conclusion

In this lab, we successfully configured basic router settings and implemented static routing using Cisco Packet Tracer. The exercise provided hands-on experience with setting up global parameters, configuring Ethernet interfaces, and establishing static routes between routers. By configuring static routes, we manually directed network traffic, ensuring that different network segments were able to communicate effectively.

Lab 9: Implementation of Dynamic interior/ Exterior Routing (RIP, OSPF, BGP)

Theory

Dynamic Interior/Exterior Routing

Interior Routing: Dynamic Interior Routing refers to routing protocols used within a single autonomous system (AS). An autonomous system is a collection of IP networks and routers under the control of a single organization. Dynamic Interior Gateway Protocols (IGPs) automatically update the routing table in response to changes in network topology, making it easier to maintain large networks. Interior routing protocols help ensure that data packets find the most efficient path within the AS.

The most common Interior Gateway Protocols (IGPs) include:
1. RIP (Routing Information Protocol)
2. OSPF (Open Shortest Path First)
3. EIGRP (Enhanced Interior Gateway Routing Protocol)

Exterior Routing: Dynamic Exterior Routing is used for routing between different autonomous systems, typically over the internet. Exterior Gateway Protocols (EGPs) are designed to exchange routing information between different organizations, ISPs, or large networks. The most common protocol in this category is BGP (Border Gateway Protocol). BGP ensures that data packets can travel across the internet by finding the best path between ASes, taking into consideration policies, path attributes, and network stability. Unlike interior protocols, BGP focuses on scalability, security, and policy-based routing to manage traffic between different autonomous systems.

RIP, OSPF, BGP

RIP (Routing Information Protocol): RIP is a distance-vector routing protocol that uses hop count as the metric to determine the best route. It has a maximum hop limit of 15, making it suitable for small networks. While easy to configure, it has slow convergence and is not ideal for large or complex networks due to its simplicity and limited scalability.

OSPF (Open Shortest Path First): OSPF is a link-state routing protocol that calculates the shortest path using the Dijkstra algorithm. It's highly scalable and supports large networks by dividing them into areas. OSPF converges quickly and allows for advanced features like route summarization and variable-length subnet masks, but it is more complex to configure than RIP.

BGP (Border Gateway Protocol): BGP is a path-vector protocol used for routing between different autonomous systems, primarily on the internet. It prioritizes policy- based routing, making it essential for controlling traffic between ISPs and large networks. BGP is highly scalable but requires careful configuration due to its complexity and slower convergence compared to interior protocols like OSPF.

Network Diagram

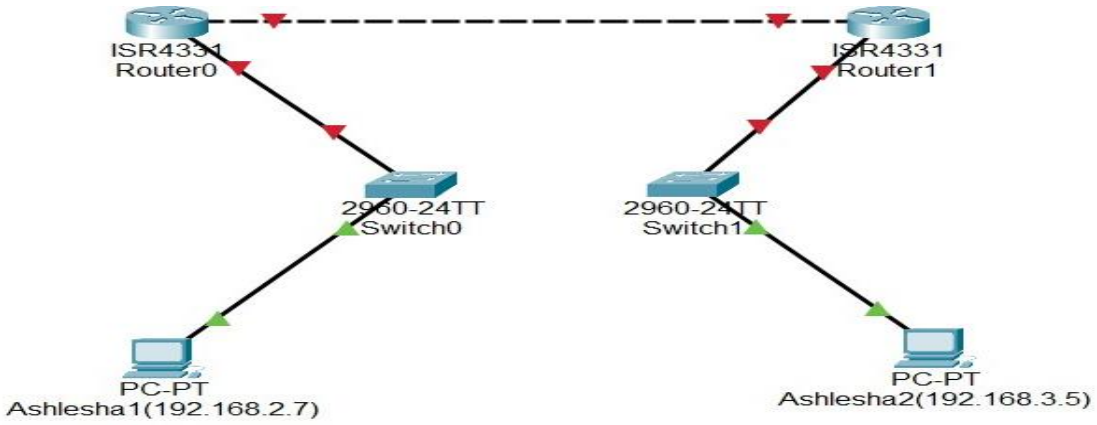


Fig: Network Diagram

Configuring Network

Configure network for PCs and Routers

Steps:

Configure the IP addresses of PCs connected to each network.

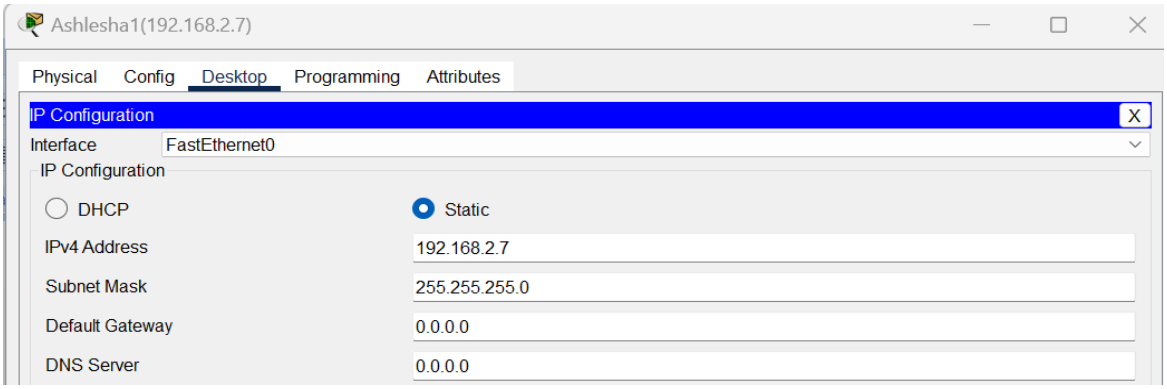


Fig: IP configuration

1. Configure the Gigabit Ethernet interfaces of the router to enable communication between different networks.

```
Ashlesha(config)#interface GigabitEthernet0/0/0
Ashlesha(config-if)#no shutdown

Ashlesha(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to up
ip address 192.168.1.2 255.255.255.0
Ashlesha(config-if)#
Ashlesha(config-if)#exit
Ashlesha(config)#
```

Fig: Gigabit Ethernet configuration

2. Repeat same for other PC's and router as well.

Implementation & Need for Dynamic Routing

Implementation

Dynamic routing protocols like RIP, OSPF, and BGP are implemented to allow routers to automatically adjust and exchange routing information in response to network changes. We configure routers to use these protocols to dynamically update their routing table4

Network Configuration: Set up the routers and PCs, assigning IP addresses to each device.

Dynamic Routing Setup: Activate RIP, OSPF, or BGP on each router, depending on the network's size and complexity, to automatically share routing information.

Testing: Validate the configurations using tools like ping to ensure network connectivity across the routers.

Need for Dynamic Routing

Dynamic routing is essential in modern networks due to its ability to automatically adjust to changes in the network topology without requiring manual intervention. The key reasons for needing dynamic routing are:

Automatic Route Updates: When a network changes, such as when a link fails or new devices are added, dynamic routing protocols automatically update routing tables across routers. This ensures continuous connectivity without manual reconfiguration.4

Scalability: In large or frequently changing networks, manually configuring static routes becomes unmanageable. Dynamic routing protocols efficiently handle the routing of traffic as the network grows, making it scalable.

Efficient Path Selection: Dynamic protocols continuously monitor network conditions and select the best possible path for data transmission. This helps optimize network performance and reduces delays.

Redundancy and Fault Tolerance: Dynamic routing enhances network reliability by quickly adapting to network failures, rerouting traffic through alternate paths, which minimizes downtime.

Overall, dynamic routing is necessary for reducing administrative overhead and ensuring the network remains efficient and responsive to changes.

Dynamic Routing Configuration

Using RIP Command

Network Diagram

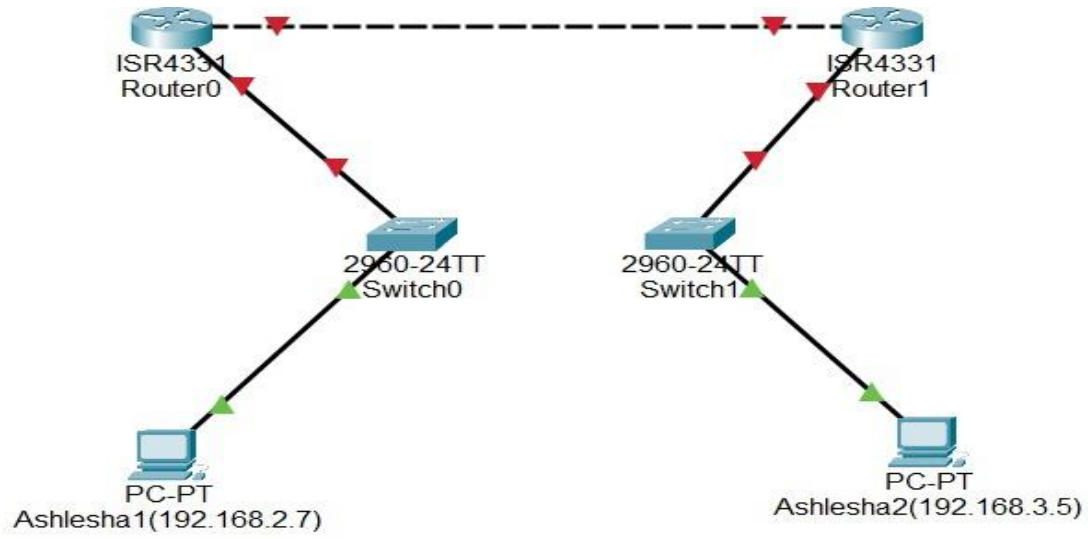


Fig: Network Diagram

Connection Testing Before Dynamic Routing Configuration using RIP command

Steps:

1.Pinging PC(Ashlesha2(192.168.3.5)) from PC(Ashlesha1(192.168.2.7)) to verify connection if exists.

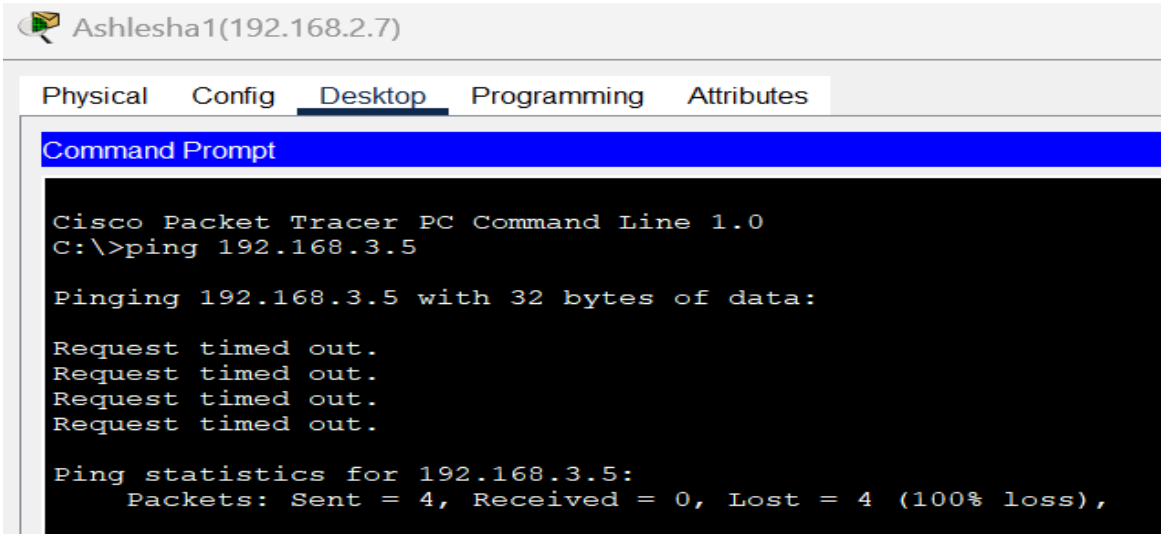


Fig: Connectivity test from PC(Ashlesha1(192.168.2.7)) to PC(Ashlesha2(192.168.3.5))

Here we can see there is not any connection in the network. So, to establish connection in the network, we need to dynamically configure the router through CLI using RIP command.

Code For Dynamic Routing Configuration Using RIP Command

For Router 0:

```
Router0> enable
Router0# configure terminal
Router0(config)# router rip
Router0(config-router)# version 2
Router0(config-router)# network 192.168.1.0
Router0(config-router)# network 192.168.2.0
Router0(config-router)# exit
```

For Router 1:

```
Router1> enable
Router1# configure terminal
Router1(config)# router rip
Router1(config-router)# version 2
Router1(config-router)# network 192.168.1.0
Router1(config-router)# network 192.168.3.0
Router1(config-router)# exit
```

Steps For Dynamic Routing Configuration Using RIP Command

1. Access Router.
2. Enable RIP on Router.
3. Specify RIP version 2 (for more efficiency and subnet support).
4. Advertise the networks connected to Router1 (LAN and WAN).
5. Exit RIP configuration.
6. Repeat same steps for another Router.

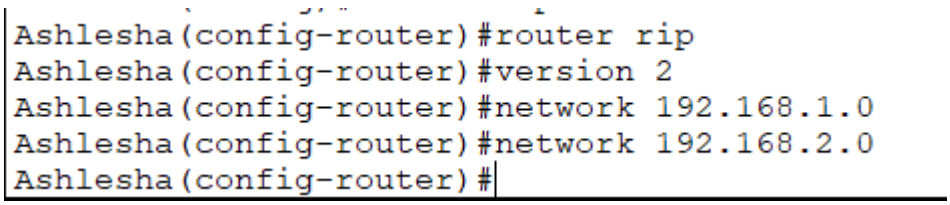


Fig: Router configuration on router (Ashlesha) using RIP command.

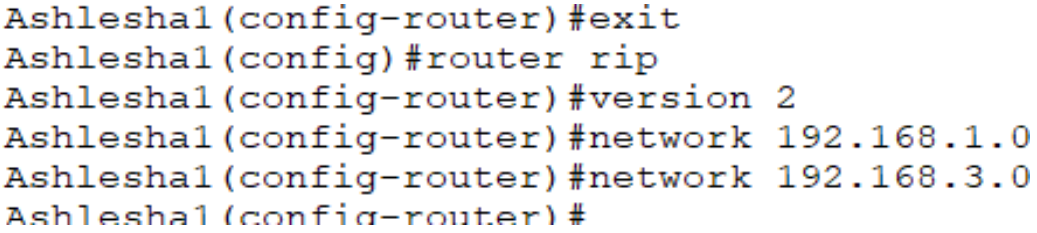


Fig: Router configuration router (Ashlesha1) using RIP command.

Testing and Validation

Totest whether the network is working, you can ping other devices on the network from each PC.

Steps:

1. Ping PC(Ashlesha2(192.168.3.5)) from PC(Ashlesha1(192.168.2.7)) .
2. If the ping is successful, you should see replies from the other device.

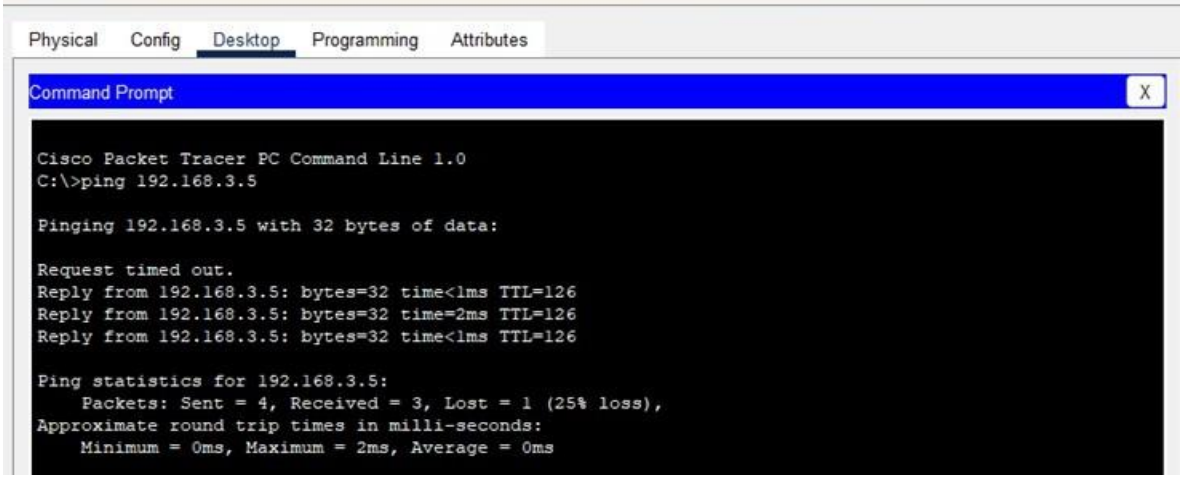


Fig: Connectivity test from PC(Ashlesha1(192.168.2.7)) to PC(Ashlesha2(192.168.3.5)

Using OSPF Command Network

Diagram

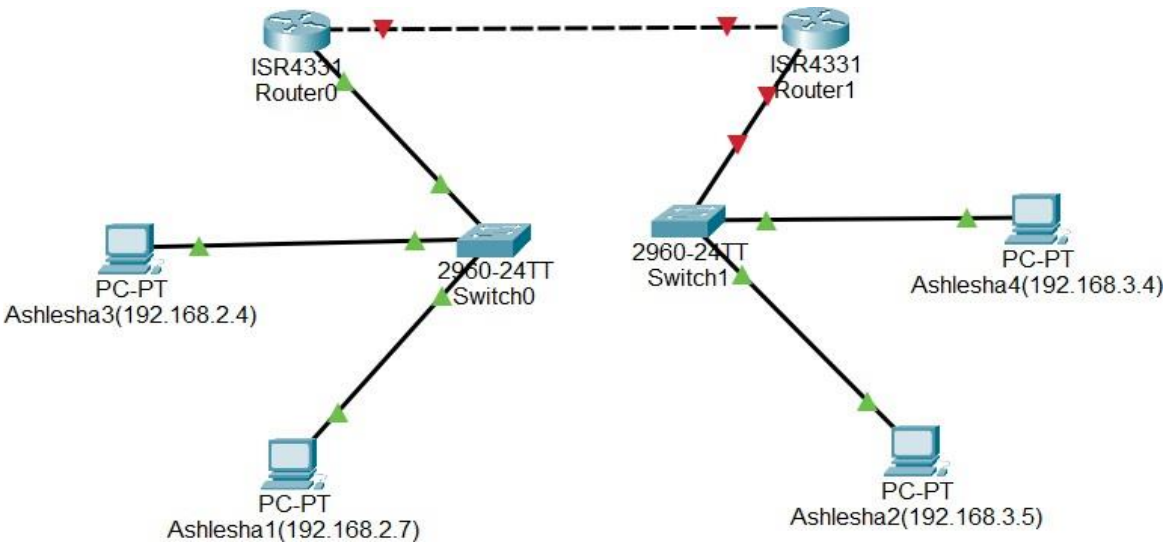


Fig: Network Diagram

Connection Testing Before Dynamic Routing Configuration using OSPF command

Steps:

1.Pinging PC(Ashlesha2(192.168.3.5)) from PC(Ashlesha1(192.168.2.7)) to verify connection if exists.

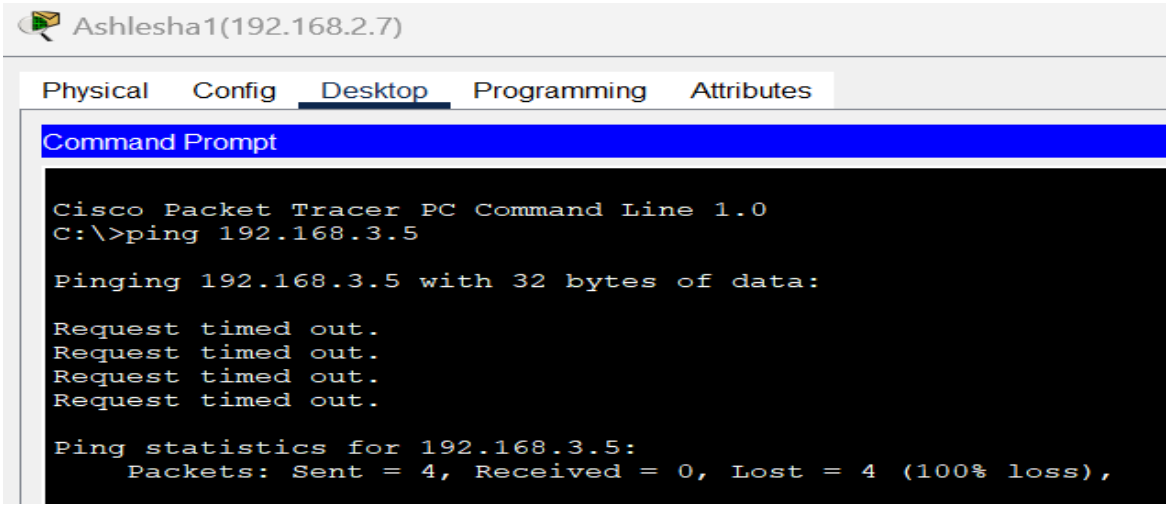


Fig: Connectivity test from PC(Ashlesha1(192.168.2.7)) to PC(Ashlesha2(192.168.3.5))

Here we can see there is not any connection in the network. So, to establish connection in the network, we need to dynamically configure the router through CLI using OSPF command.

Code For Dynamic Routing Configuration Using OSPF Command For

Router 0:

```
Router0> enable
Router0# configure terminal
Router0(config)# router ospf 1
Router0(config-router)# router-id 1.1.1.1
Router0(config-router)# network 192.168.1.0 0.0.0.255 area 0
Router0(config-router)# network 192.168.2.0 0.0 0.255 area 0
Router0(config-router)# exit
```

For Router 1:

```
Router1> enable
Router1# configure terminal
Router1(config)# router ospf 1
Router1(config-router)# router-id 2.2.2.2
Router1(config-router)# network 192.168.1.0 0.0.0.255 area 0
Router1(config-router)# network 192.168.3.0 0.0 0.255 area 0
Router1(config-router)# exit
```

Steps For Dynamic Routing Configuration Using OSPF Command

1. Access Router.
2. Start the OSPF process and assign it a process ID (use 1 in this case)
3. Assign a router ID (OSPF will choose automatically if omitted)
4. Specify the networks connected to Router1, and define the areas.
5. Exit OSPF configuration.
6. Repeat same steps for another Router

```
Ashlesha#config ter
Enter configuration commands, one per line.  End with CNTL/Z.
Ashlesha(config)#router ospf 1
Ashlesha(config-router)#router-id 1.1.1.1
Ashlesha(config-router)#network 192.168.1.0 0.0.0.255 area 0
Ashlesha(config-router)#network 192.168.2.0 0.0.0.255 area 0
Ashlesha(config-router)#exit
Ashlesha(config)#
```

Fig: Router configuration on router (Ashlesha) using OSPF command.

```
Ashlesha1>enable
Ashlesha1#config ter
Enter configuration commands, one per line.  End with CNTL/Z.
Ashlesha1(config)#router ospf 1
OSPF process 1 cannot start. There must be at least one "up" IP interface
Ashlesha1(config-router)#router-id 2.2.2.2
Ashlesha1(config-router)#network 192.168.1.0 0.0.0.255 area 0
Ashlesha1(config-router)#network 192.168.3.0 0.0.0.255 area 0
Ashlesha1(config-router)#exit
Ashlesha1(config)#
```

Fig: Router configuration on router (Ashlesha1) using OSPF command.

Testing and Validation

To test whether the network is working, you can ping other devices on the network from each PC.

Steps:

1. Ping PC(Ashlesha2(192.168.3.5)) from PC(Ashlesha1(192.168.2.7).
2. If the ping is successful, you should see replies from the other device.

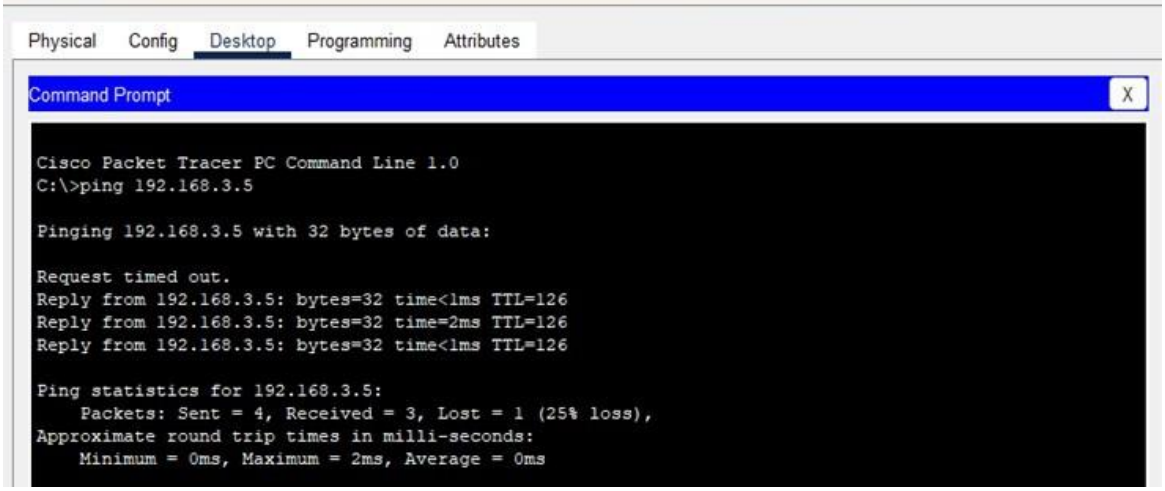


Fig: Connectivity test from PC(Ashlesha1(192.168.2.7)) to PC(Ashlesha2(192.168.3.5))

Using BGP Command Network

Diagram

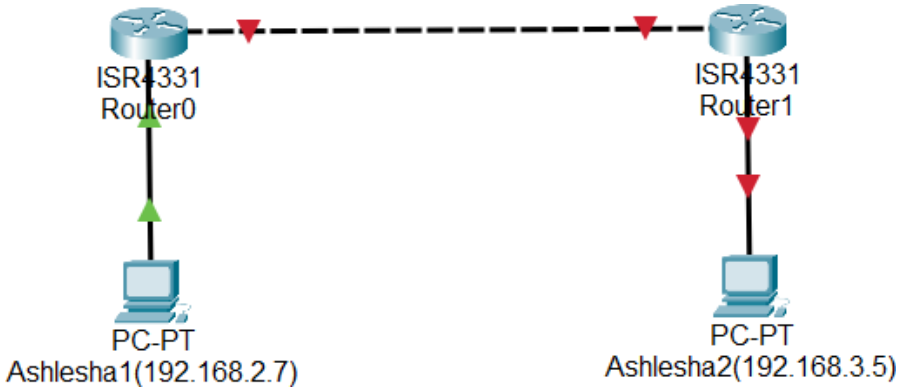


Fig: Network Diagram

Connection Testing Before Dynamic Routing Configuration using BGP command

Steps:

1.Pinging PC(Ashlesha2(192.168.3.5)) from PC(Ashlesha1(192.168.2.7)) to verify connection if exists.

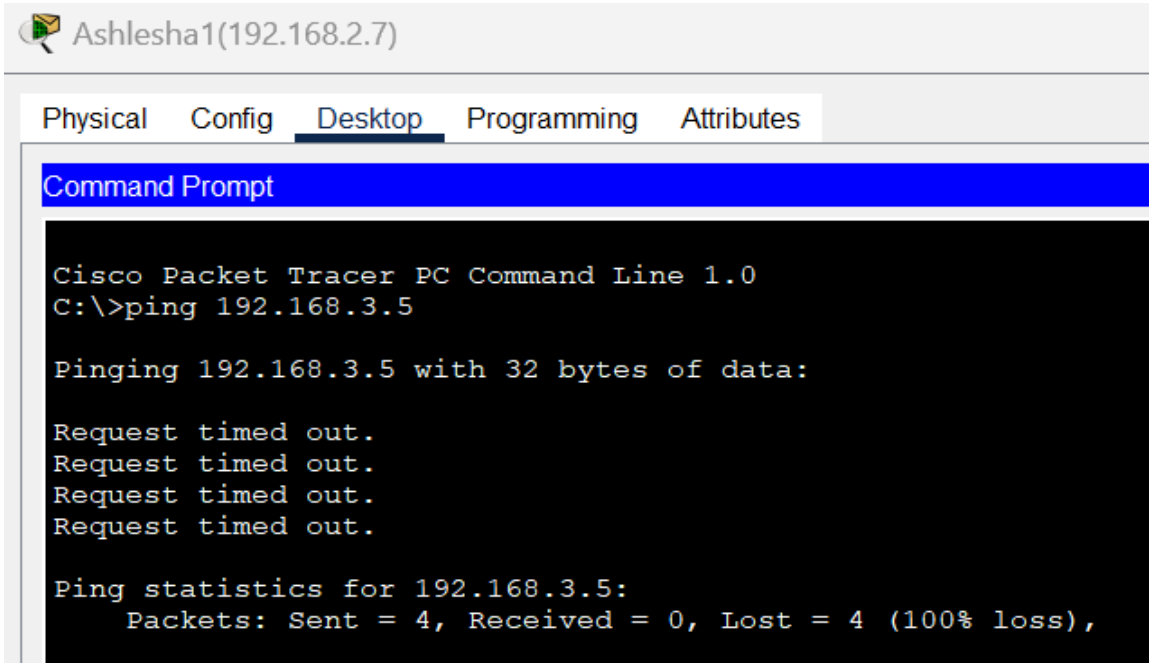


Fig: Connectivity test from PC(Ashlesha1(192.168.2.7)) to PC(Ashlesha2(192.168.3.5))

Here we can see there is not any connection in the network. So, to establish connection in the network, we need to dynamically configure the router through CLI using BGP command.

Code For Dynamic Routing Configuration Using BGP Command

For Router 0:

```
Router0> enable
Router0# configure terminal Router0(config)#
router bgp 65001
Router0(config-router)# neighbor 192.168.1.4  remote-as 65002
Router0(config-router)# network 192.168.1.0 mask 255.255.255.0
Router0(config-router)# exit
```

For Router 1:

```
Router1> enable
Router1# configure terminal Router1(config)#
router bgp 65002
Router1(config-router)# neighbor 192.168.1.2 remote-as 65001
Router1(config-router)# network 192.168.2.0 mask 255.255.255.0
Router1(config-router)# exit
```

Steps For Dynamic Routing Configuration Using BGP Command

1. Access Router.
2. Start the BGP process and specify the AS number (65001)
3. Specify Router1 as a neighbor and provide its AS number (65002)
4. Advertise the LAN network behind Router1
5. Exit BGP configuration.
6. Repeat same steps for another Router.

```
Ashlesha1(config-router)#router bgp 65002
Ashlesha1(config-router)#neighbor 192.168.1.2 remote-as 65001
Ashlesha1(config-router)#network 192.168.1.4 mask 255.255.255.0
Ashlesha1(config-router)#exit
Ashlesha1(config)#
```

Fig: Router configuration on router (Ashlesha1) using BGP command.

```
Ashlesha(config)#router bgp 65001
Ashlesha(config-router)#neighbor 192.168.1.4 remote-as 65002
Ashlesha(config-router)#network 192.168.1.0 mask 255.255.255.0
Ashlesha(config-router)#exit
Ashlesha(config)#
```

Fig: Router configuration on router (Ashlesha) using BGP command.

Testing and Validation

To test whether the network is working, you can ping other devices on the network from each PC.

Steps:

1. Ping PC(Ashlesha1(192.168.2.7)) from PC(Ashlesha2(192.168.3.5)).
2. If the ping is successful, you should see replies from the other device.

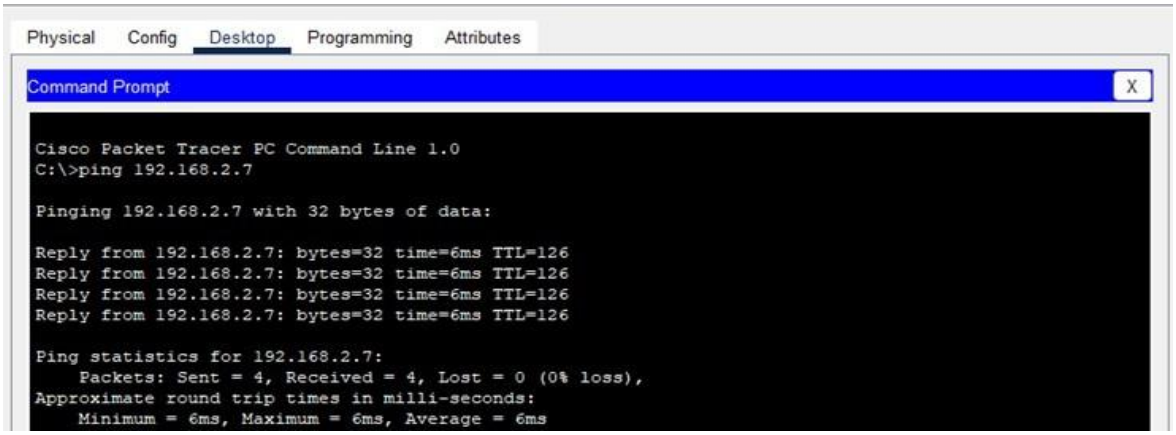


Fig: Connectivity test from PC(Ashlesha2(192.168.3.5))
to PC(Ashlesha1(192.168.2.7)

Conclusion

In this lab, we successfully implemented three key dynamic routing protocols—RIP, OSPF, and BGP demonstrating their functionality in both interior and exterior routing contexts. Each protocol serves distinct purposes based on the network's size, complexity, and requirements. Through testing and verification, we observed how dynamic routing protocols automatically adjusted to network changes and maintained efficient data routing.

Lab 10: Implementing ACL in Packet Tracer

Theory

An Access Control List (ACL) is a set of rules applied to router interfaces to control the traffic that can flow through the network. ACLs can be configured to allow or deny traffic based on various parameters, such as source/destination IP addresses, protocols (TCP, UDP, ICMP), and port numbers.

ACLs are classified into two types:
Standard ACL: Filters traffic based only on the source IP address.
Extended ACL: Filters traffic based on both source and destination IPs, protocols, and ports.

The primary functions of ACLs are:
1. Enhancing network security by limiting access to certain resources.
2. Controlling traffic flow by permitting or denying packets based on specific criteria.
3. Improving network performance by reducing unwanted traffic.

Network Diagram

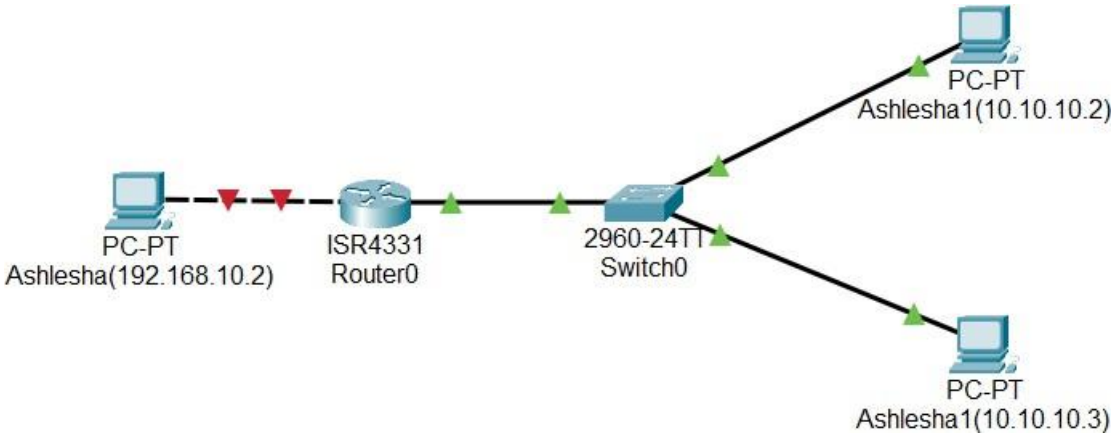


Fig: Network Diagram

Configure network for PCs and Routers

Configure PCs

1. Assign IP addresses to the PCs according to the network plan like PC(Ashlesha3(192.168.10.2)), PC(Ashlesha2(10.10.10.2))

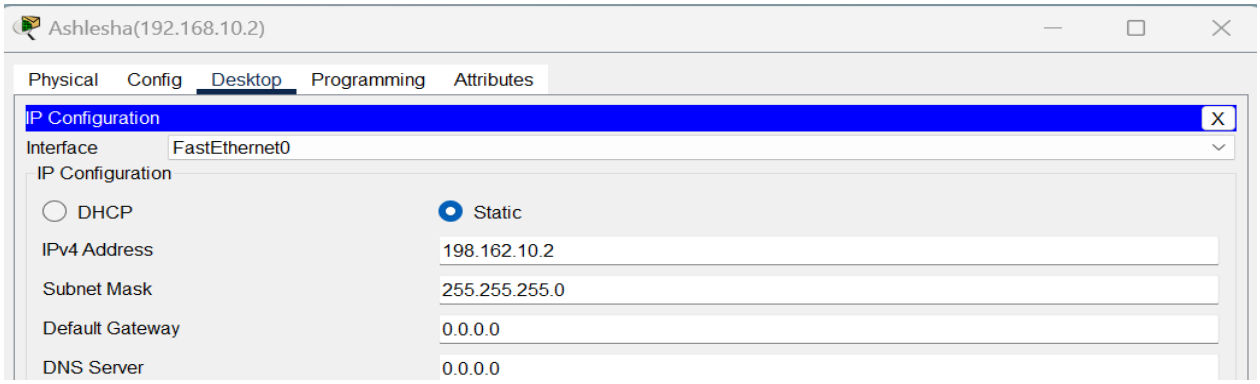


Fig: IP configuration on PC's

Configure Routers

1. Access the router's command-line interface.
2. Assign IP addresses to router interfaces that connect to the PCs.

```
Ashlesha(config)#interface GigabitEthernet0/0/0
Ashlesha(config-if)#ip address 192.168.10.1 255.255.255.0
Ashlesha(config-if)#no shutdown
Ashlesha(config-if)#
```

Fig: Router configuration

Configuring Access List

Configure DENY and PERMIT list

1. Access global configuration mode
2. Apply the ACL to an interface (e.g., blocking PC1's access to the network):

```
Ashlesha(config)#access-list 1 deny host 10.10.10.2
Ashlesha(config)#access-list 1 permit host 10.10.10.3
Ashlesha(config)#int gig0/0/0
Ashlesha(config-if)#ip access-group 1 in
Ashlesha(config-if)#exit
Ashlesha(config)#
```

Fig: Configuring DENY and PERMIT list

Code For Configuring DENY and PERMIT list

Router(config)# access-list 1 deny host 10.10.10.2
Router(config)# access-list 1 permit host 10.10.10.3 Router(config)#
interface gig0/0
Router(config-if)# ip access-group 1 in
Router(config-if)# exit

Implementation and Testing

To test whether the network is working, you can ping other devices on the network from each PC.

Steps:

1. Ping PC(Ashlesha1(10.10.10.2)) from PC(Ashlesha3(192.168.10.2)) to verify that the connection is denied.
2. Ping PC(Ashlesha1(10.10.10.3)) from PC(Ashlesha3(192.168.10.2)) to verify that the connection is permitted.
3. If the ping is successful, you should see replies from the other device.

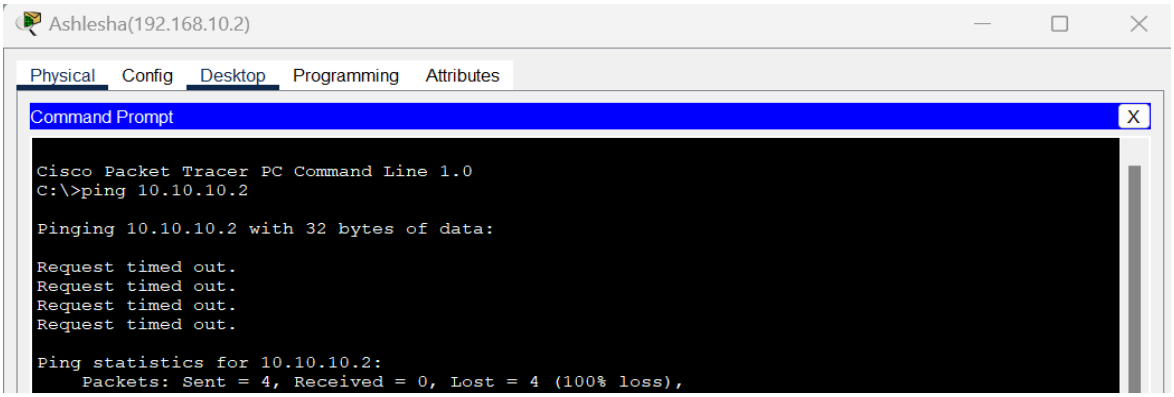


Fig: Connectivity test from PC(Ashlesha3(192.168.10.2)) to PC(Ashlesha1(10.10.10.2))

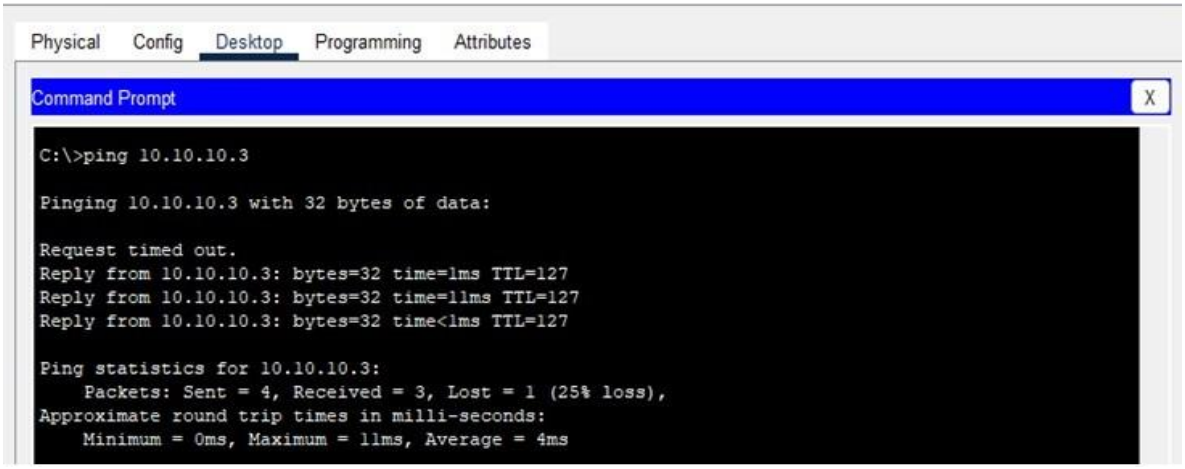


Fig: Connectivity test from PC(Ashlesha3(192.168.10.2)) to PC(Ashlesha1(10.10.10.3)

Conclusion

In this lab, we implemented Access Control Lists (ACLs) using Cisco Packet Tracer to control network traffic. Both standard and extended ACLs were used to permit or deny traffic based on IP addresses, protocols, and ports. This exercise highlighted the importance of ACLs in enhancing network security by managing access and restricting unauthorized traffic. Testing confirmed the proper functionality and effectiveness of the ACL configurations.

Lab 12: FTP Configuration and Implementation using Packet Tracer

Theory

FTP (File Transfer Protocol) is a standard network protocol used to transfer files between a client and a server over a TCP/IP network. It uses port 21 for control commands and port 20 for data transfer in active mode. Clients can access servers either anonymously or with authentication. FTP operates in two modes: active and passive, depending on whether the server or client initiates the data connection. For secure transfers, FTPS or SFTP can be used to encrypt data. FTP supports various file operations such as uploading, downloading, and managing files and directories through specific commands like RETR, STOR, and DELE. In ASCII mode, it handles text files by converting line endings between different operating systems, while in binary mode, it preserves the exact byte sequence of files. Despite its functionality, FTP is considered less secure compared to modern protocols due to its lack of built-in encryption, making secure alternatives like FTPS and SFTP preferable for sensitive data transfers.

Key Concepts of FTP

1. **Client-Server Model**: FTP follows a client-server architecture where the client initiates requests for file operations and the server responds. The client communicates with the server to request files, upload files, or perform other file management tasks.

2. **Ports:** FTP uses port 21 for the control connection, where commands and responses are exchanged. Port 20 is used for the data connection in active mode, while passive mode uses a dynamically assigned port by the server for data transfer.

3. **Active and Passive Modes:** In active mode, the client opens a port for data transfer and the server connects to it. In passive mode, the server opens a port and the client connects to it, which helps navigate firewalls and NAT issues.

4. **Authentication:** FTP can operate in anonymous mode, allowing users to access files without a password, or in authenticated mode, requiring a username and password for access. This distinction helps manage access and security.

5. **FTP Commands:** Common FTP commands include LIST to view files, RETR to download files, STOR to upload files, DELE to delete files, and MKD to create directories. These commands manage file operations on the server.

6. **Data Types:** FTP supports ASCII mode for text files, converting line endings between systems, and binary mode for non-text files, preserving exact byte sequences. Choosing the correct mode ensures proper file integrity.

7. **FTP Security:** FTPS and SFTP are used to secure FTP connections. FTPS adds SSL/TLS encryption to FTP, while SFTP uses SSH for secure file transfer, protecting data from interception and unauthorized access.

Network Diagram

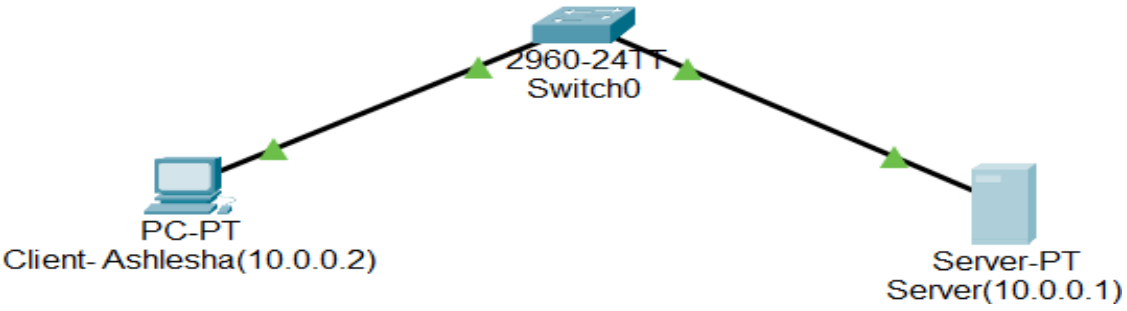


Fig: Network diagram

Configuring FTP Server and FTP Client

FTP Server Configuration

Step1: Click on server and go to ip configuration and set ip address and subnet mask.

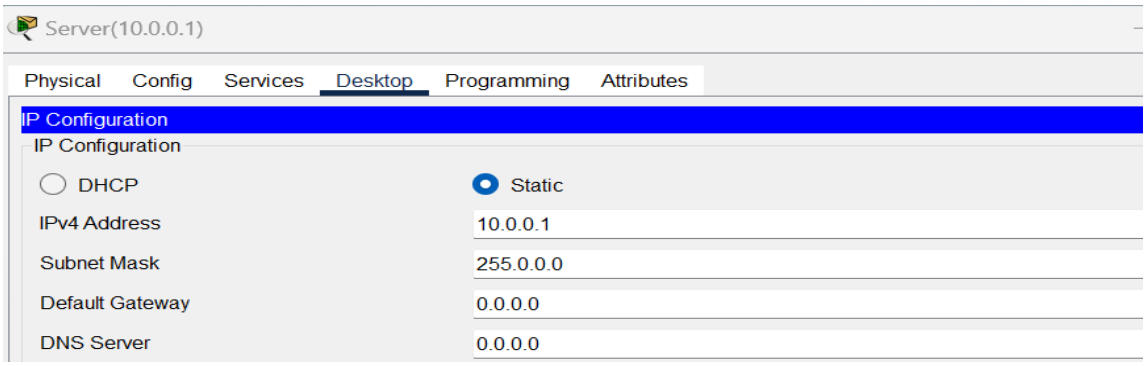


Fig: IP configuration on Sever

Step 2: Click on server, go to service, click ftp and click on ON button.

Step 3: Set username, password, and tick write, read, delete, rename, and list. Click add.

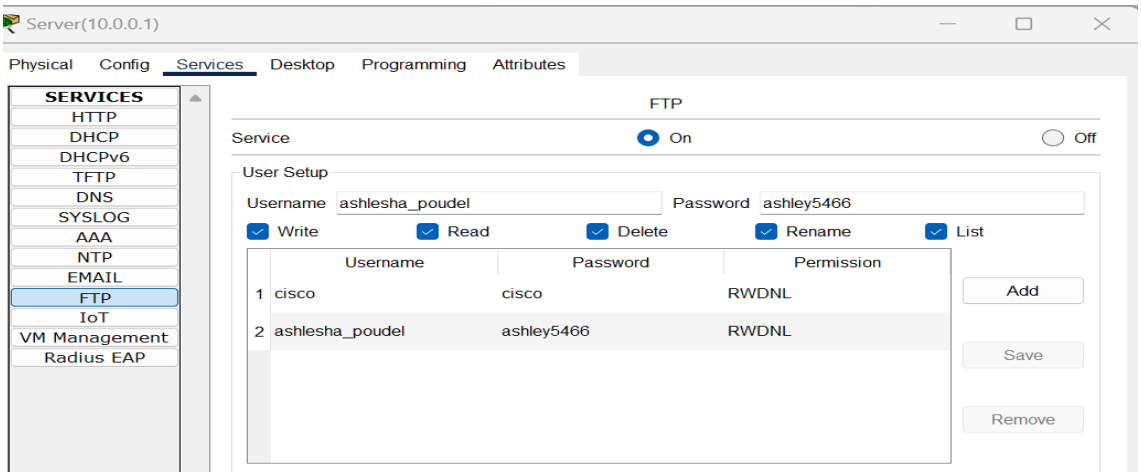


Fig: Server Configuration

Step 4: Go to desktop, click text editor write something and save the file as Ashlesha.txt.

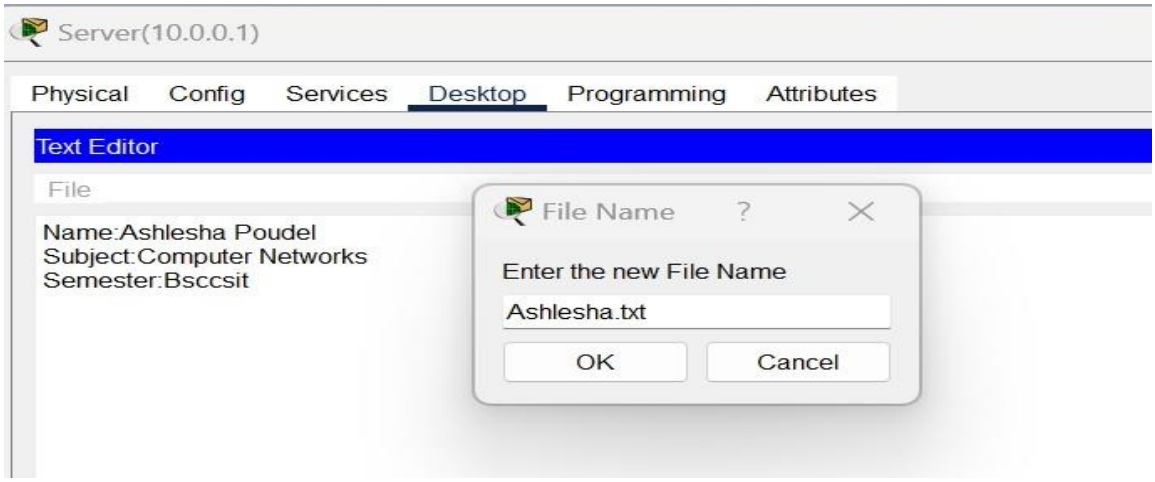


Fig: Creating a file name Ashlesha.txt

Step 5: In desktop, open command prompt and type dir command, we can see the file.

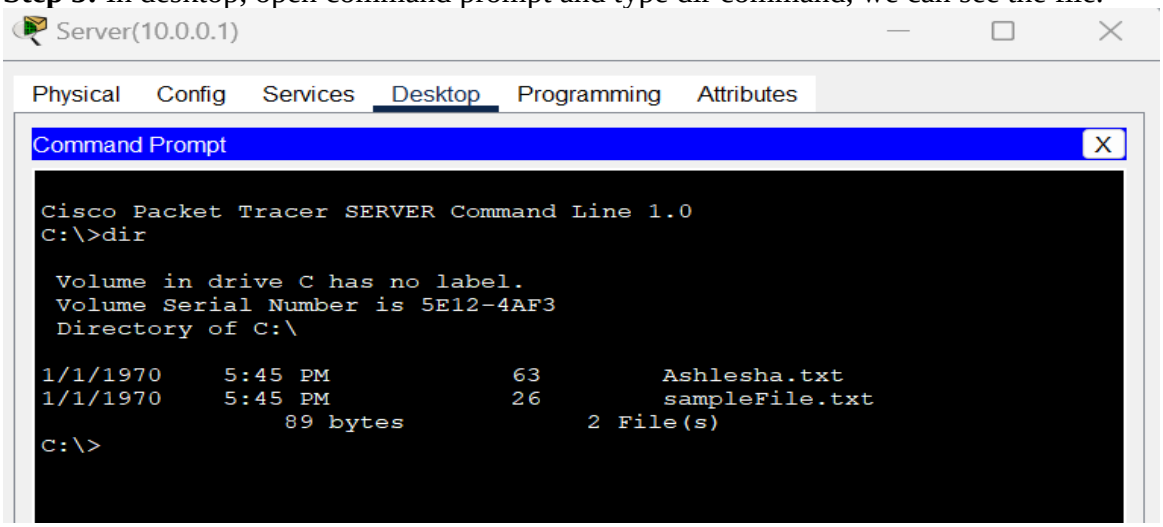


Fig: Using dir command to see file

FTP Client Configuration

Step 1: Click on pc and goto ip configuration and set ip address and subnet mask.

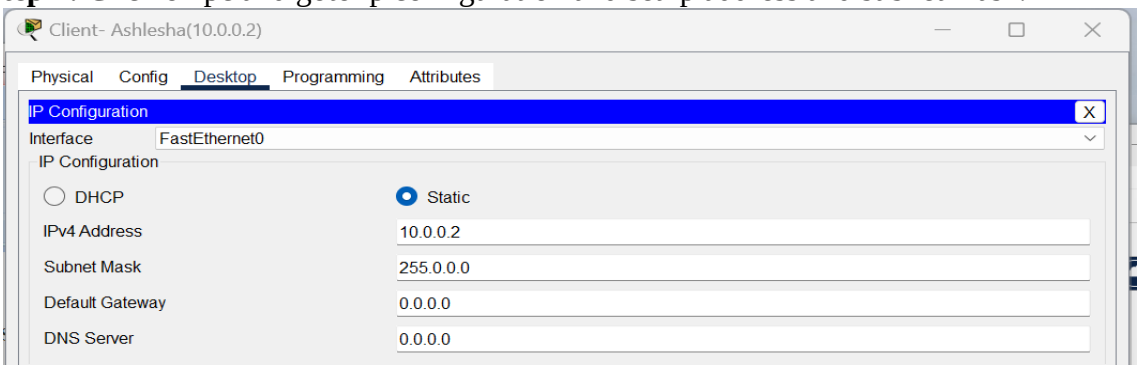


Fig: IP Configuration in Client

In command prompt type command ‘ftp 10.0.0.1’then insert username and password, we will be connected to ftp server.

```
1/1/1970    5:45 PM             63          Ashlesha.txt
1/1/1970    5:45 PM             26          sampleFile.txt
               89 bytes               2 File(s)
C:\>ftp 10.0.0.1
Trying to connect...10.0.0.1
Connected to 10.0.0.1
220- Welcome to PT Ftp server
Username:ashlesha_poudel
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

Fig: FTP Server Connection

TRANSFERRING FILE USING PUT COMMAND

Command: Ftp > put Ashlesha.txt

```
ftp>put Ashlesha.txt

Writing file Ashlesha.txt to 10.0.0.1:
File transfer in progress...

[Transfer complete - 63 bytes]

63 bytes copied in 0.042 secs (1500 bytes/sec)
ftp>
```

Fig: Transferring file using PUT command

RENAME FILE

Command: Ftp > rename Ashlesha.txt temp.txt

```
ftp>rename Ashlesha.txt tempt.txt

Renaming Ashlesha.txt
ftp>
[OK Renamed file successfully from Ashlesha.txt to tempt.txt]
ftp>
```

Fig: Renaming file

GET THE FILE AND SAVE THE COPY ON OUR MACHINE

Command: Ftp > get temp.txt

FTP Server Connection

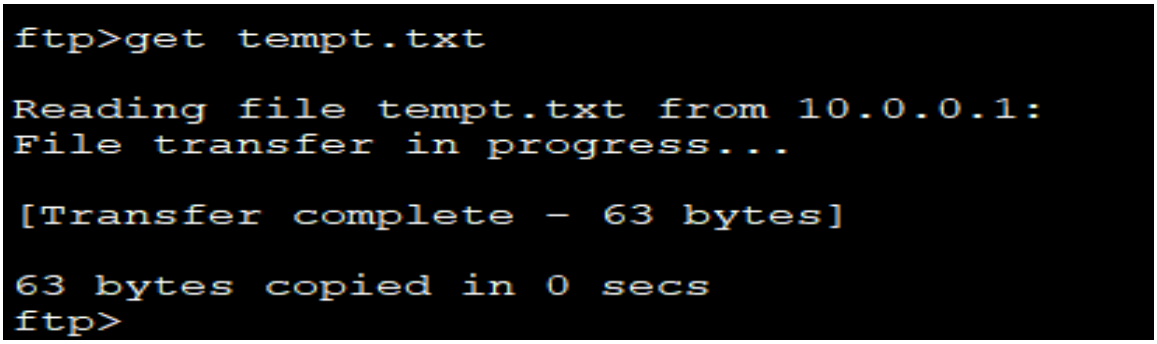


Fig: Saving copy of file in PC

GO TO PC

Command: Ftp> quit ftp

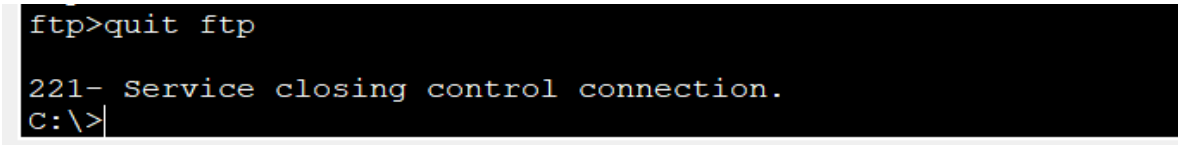


Fig: Quitting FTP

DISPLAYING THE FILES

Command: PC > dir

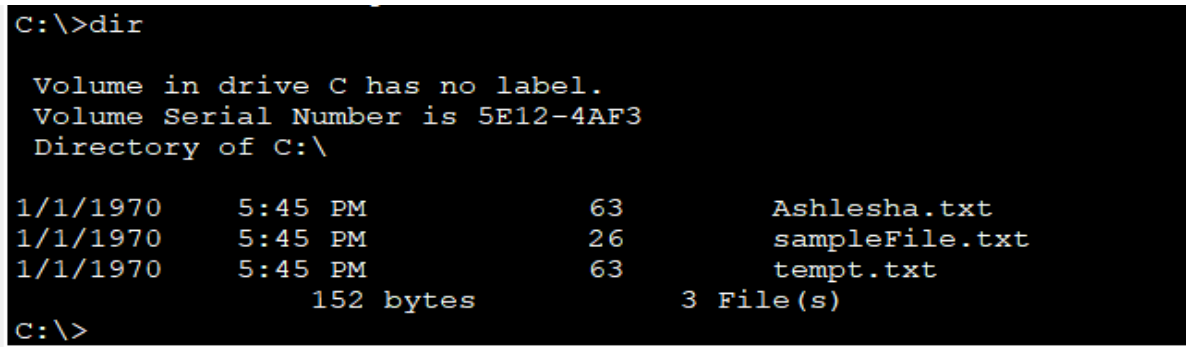


Fig: Displaying the files

Conclusion

In this lab, we successfully configured an FTP server and client using Packet Tracer. By following the step-by-step process, we understood how to set up an IP address, enable the FTP service, and create user accounts with appropriate permissions. We also created a file, verified its existence through command-line operations, and demonstrated file management using FTP commands. This practical implementation helps in understanding of FTP's client-server model, the role of IP addressing, and the importance of secure file transfer operations in network environments.

Lab-13: Introduction to Network Traffic Analysis using Wireshark

Theory

Wireshark is a popular open-source tool for analyzing network protocols. It captures and examines network traffic in real time, enabling users to inspect data packets for troubleshooting, monitoring performance, and detecting security threats. Featuring a graphical interface, it supports numerous protocols and offers detailed insights into network communications. Wireshark is a crucial tool for both network administrators and cybersecurity specialists.

Key Concepts of Wireshark

Packet Capture: Wireshark captures network data packets in real time as they move through a network. Each packet includes headers and data, which can be analyzed to assess network performance or identify security problems.

Protocols: Wireshark supports thousands of protocols (e.g., TCP, UDP, HTTP, DNS), allowing you to analyze traffic from different network layers.

Filters: Filters in Wireshark allow users to refine data based on specific criteria, such as IP addresses, port numbers, or protocol types. Display filters show only the packets of interest, while capture filters restrict the data being recorded during capture.

Frames and Layers: Wireshark displays packet data in different layers (e.g., Ethernet, IP, TCP) following the OSI model, which helps break down packet content at various levels of networking.

Hexadecimal and ASCII Views: The packet data is shown in both hexadecimal and ASCII formats, allowing you to inspect the raw content of a packet and understand its structure.

Packet Dissection: Wireshark breaks down packet contents into readable components, showing fields, flags, and values related to the specific protocol in use.

Statistics and Graphs: Wireshark provides statistical tools (e.g., flow graphs, IO Wireshark can capture packets from different interfaces, including Ethernet, Wi-Fi, and loopback, based on the system's available interfaces.

Real-Time and Offline Analysis: Wireshark can analyze both live network traffic and saved capture files (PCAP), making it versatile for both real-time troubleshooting and post-event analysis.

Security: Wireshark helps detect network security issues like malware, unauthorized access, and other vulnerabilities by analyzing unusual patterns in packet flows.

Interface of Wireshark

Main Toolbar: The toolbar provides quick access to common functions such as opening, saving, or closing captures. It also includes buttons for starting and stopping packet captures, restarting them, and filtering the packet display. Key features include a field for applying display filters, navigating through packets, and zooming in on packet views.

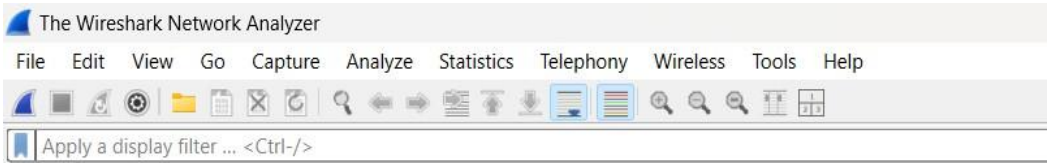


Fig: Toolbar

Packet List Pane: This pane shows all the captured packets in real time. Each row represents a packet, and columns display key details like packet number, timestamp, source, destination, protocol, length, and info. Clicking on a packet here allows you to view its detailed information in the other panes.

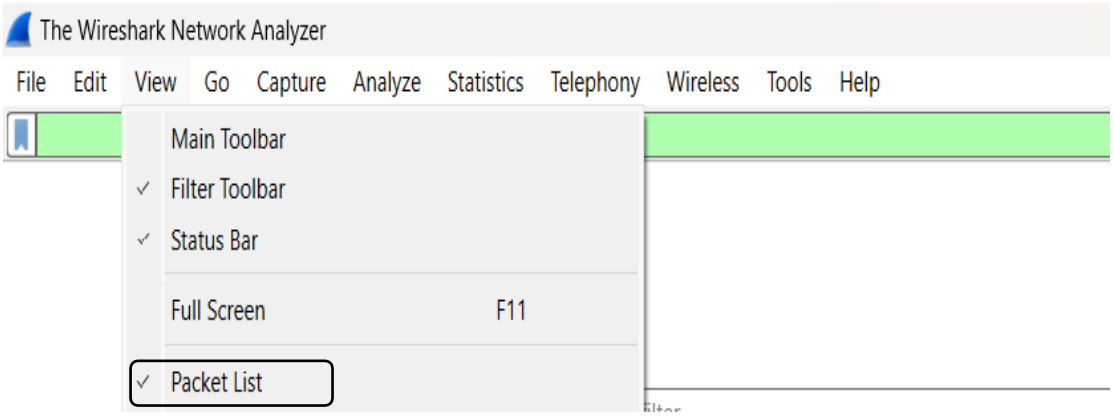


Fig: Packet list pane

Packet Details Pane: When you select a packet from the Packet List, this pane provides an expandable tree view of the packet's structure. It breaks down the packet into its various protocol layers, such as Ethernet, IP, TCP/UDP, etc., offering detailed protocolinformation for analysis.

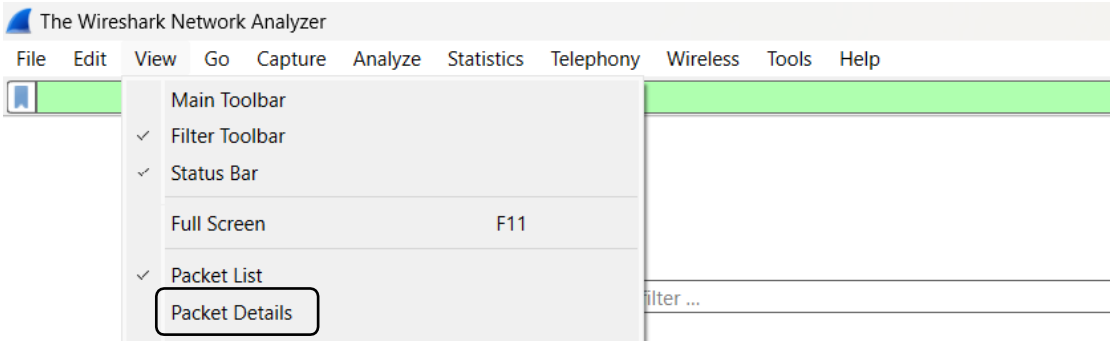


Fig: Packet Details pane

Packet Bytes Pane: This pane displays the raw data of the selected packet in hexadecimal and ASCII format. It allows users to view the actual bytes transmitted, which can be useful for low-level protocol analysis or detecting anomalies at the byte level.

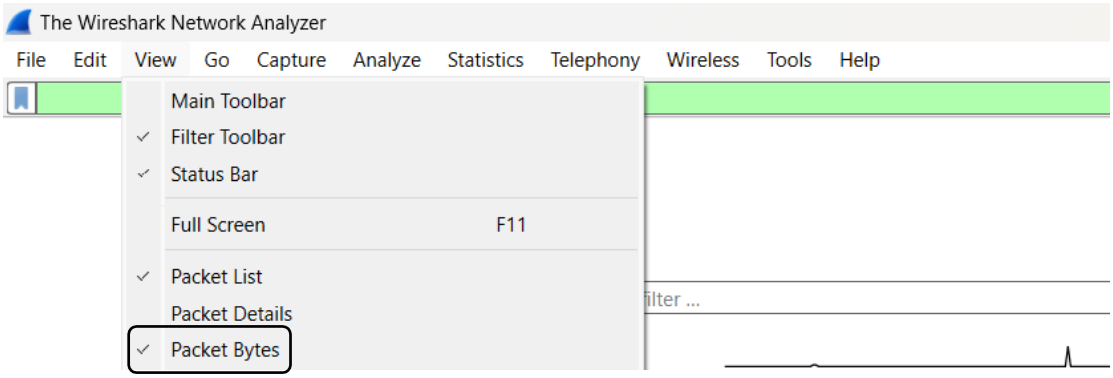


Fig: Packet Bytes Pane

Basic Network Capture and Analysis

Selecting a Network Interface

Steps:

1. Open Wireshark.
2. You'll see a list of available network interfaces (Wi-Fi, Ethernet, etc.).
3. Look for the interface capturing active traffic (typically with a moving graph or traffic count).
4. Select the desired interface by clicking on it.

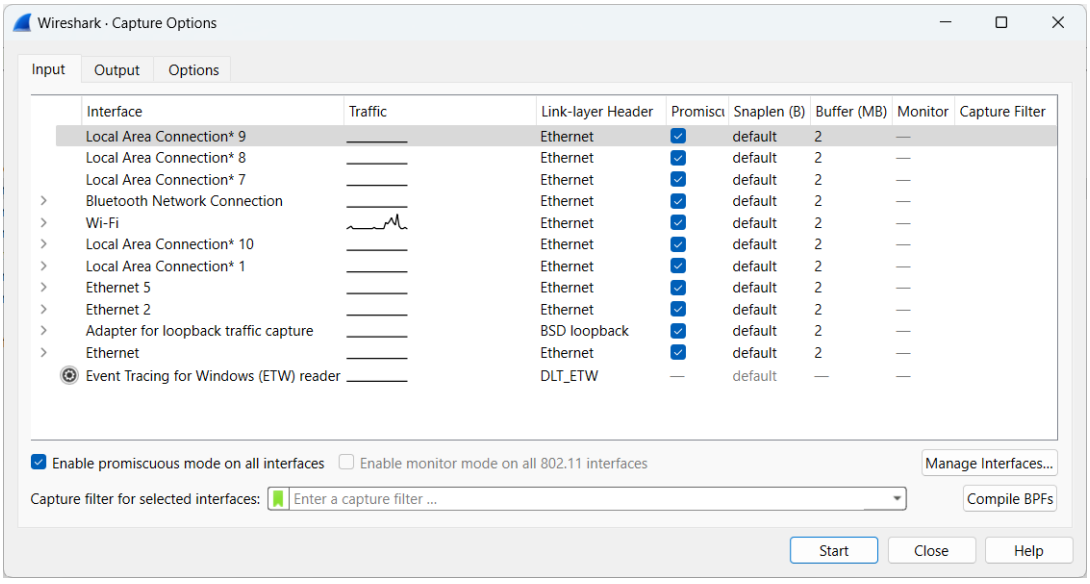


Fig: Selecting network interface

Starting Packet Capture

Steps:

1. After selecting the network interface, click the blue shark fin icon in the toolbar to start the packet capture.
2. Open your browser and navigate to any like (example.com)
3. Wireshark will start capturing all network traffic on the selected interface.
4. To filter the capture for packets related to example.com, you can use a display filter.
5. In the filter bar, type ip.addr == < ip address of src >.
6. Press Enter to apply the filter.

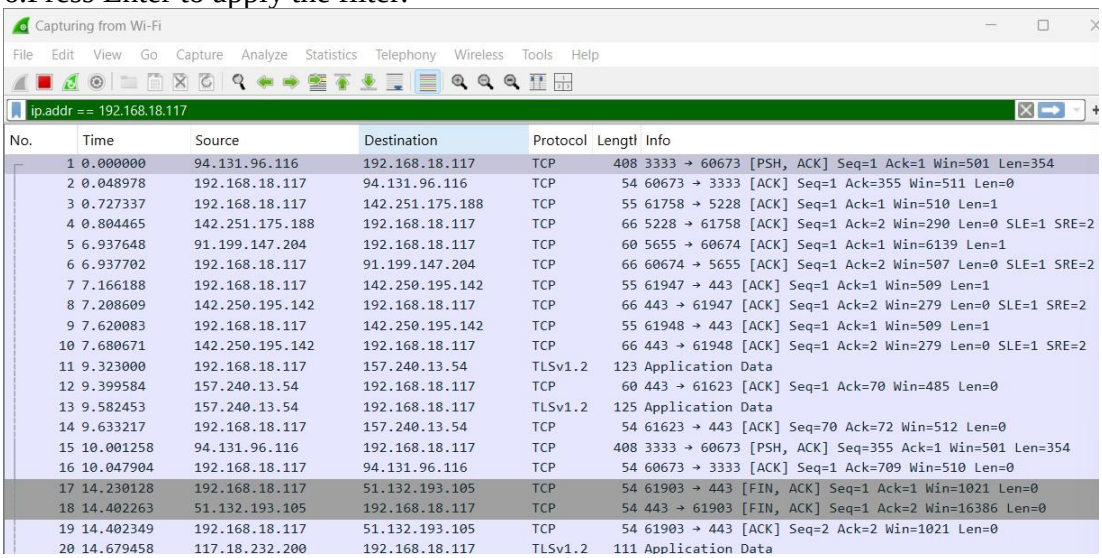


Fig: Packet Capture

Stopping and Saving Captures

Steps:

1. Once you've captured the desired traffic, click the red square icon in the toolbar to stop the capture.
2. To save the capture: Go to File > Save As.
3. Choose a file name and location.
4. Select a format (default is. pcapng), then click Save.

Fig: Saving a file

Exporting the Captured Data

Steps:

1. To export specific packets or data: Go to File > Export Specified Packets.
2. Choose the range or filter for the packets you want to export.

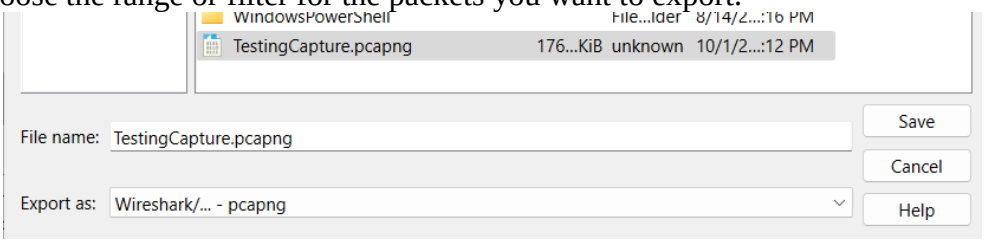


Fig: Exporting Capture Data

Conclusion

In this lab, we explored the basics of network traffic analysis with Wireshark. By capturing and analyzing live network traffic, we gained hands-on experience in identifying network protocols, examining packet details, and filtering traffic for specific hosts or services. This practical approach enhanced our understanding of how data moves through networks and equipped us with the skills needed to effectively troubleshoot and secure network communications.

Lab 14: Packet Capture and Header Analysis by Wireshark (TCP, UDP, IP)

Theory

Wireshark is a commonly used open-source tool for analyzing network traffic. It captures and examines data in real time, allowing users to inspect packets for troubleshooting, monitoring network performance, and detecting security threats. With its user-friendly interface, it supports various protocols and gives detailed insights into network communication. Wireshark is a valuable tool for network administrators and cybersecurity professionals.

TCP (Transmission Control Protocol) is a connection-oriented protocol designed to ensure reliable data transfer between devices. It first establishes a connection before transmitting data, verifies for errors, and makes sure that data packets are received in the correct sequence without any duplicates. TCP is commonly used in applications like web browsing (HTTP/HTTPS) and email to guarantee stable and accurate communication.

UDP (User Datagram Protocol) is a connectionless protocol that focuses on speed rather than reliability. It transmits data packets without setting up a connection or guaranteeing their delivery, making it well-suited for applications like video streaming, online gaming, and VoIP, where low latency is important and minor data loss can be tolerated.

IP (Internet Protocol) is the main protocol used for routing data packets between networks. Operating at the network layer, it assigns IP addresses to devices and ensures that packets are sent to the right destination through routing. IP collaborates with both TCP and UDP to enable end-to-end communication across the internet.

Network Interface Selection and Traffic Filtering

Steps:

1. Open Wireshark and select the network interface (Wi-Fi or Ethernet) where traffic is to be captured.
2. Click the start button to begin capturing live traffic.
3. Apply a filter to focus on specific traffic, such as tcp for TCP traffic, udp for UDP, or ip for general IP traffic.

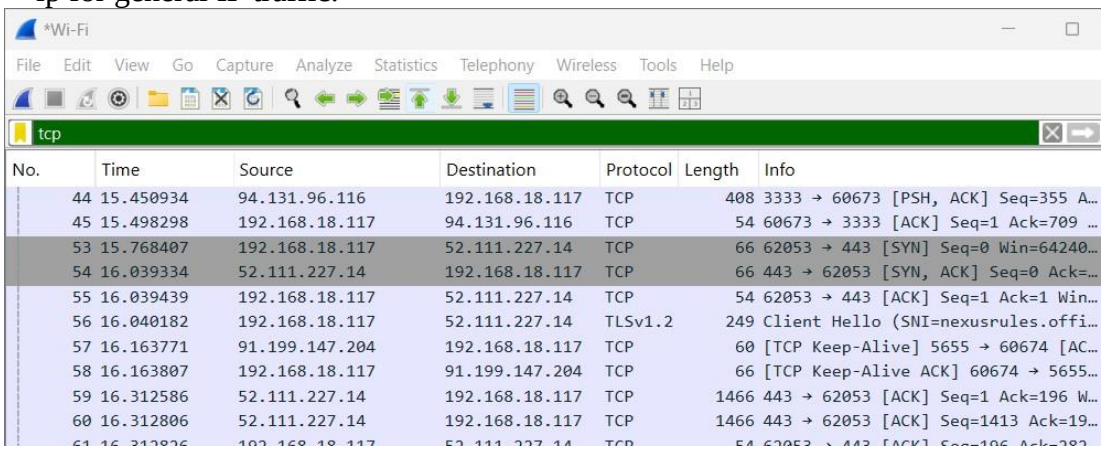


Fig: Traffic filtering

TCP Header Analysis

After capturing TCP traffic, select a TCP packet to view its header details, which include:

Source Port: Identifies the port on the sender's machine (e.g., port 443 for HTTPS).
Destination Port: Specifies the port on the recipient's machine.
Sequence Number: Keeps track of the packet's position in the communication stream.
Acknowledgment Number: Confirms the receipt of previous packets.
Flags: Control bits (e.g., SYN, ACK, FIN) used to manage the connection's state.

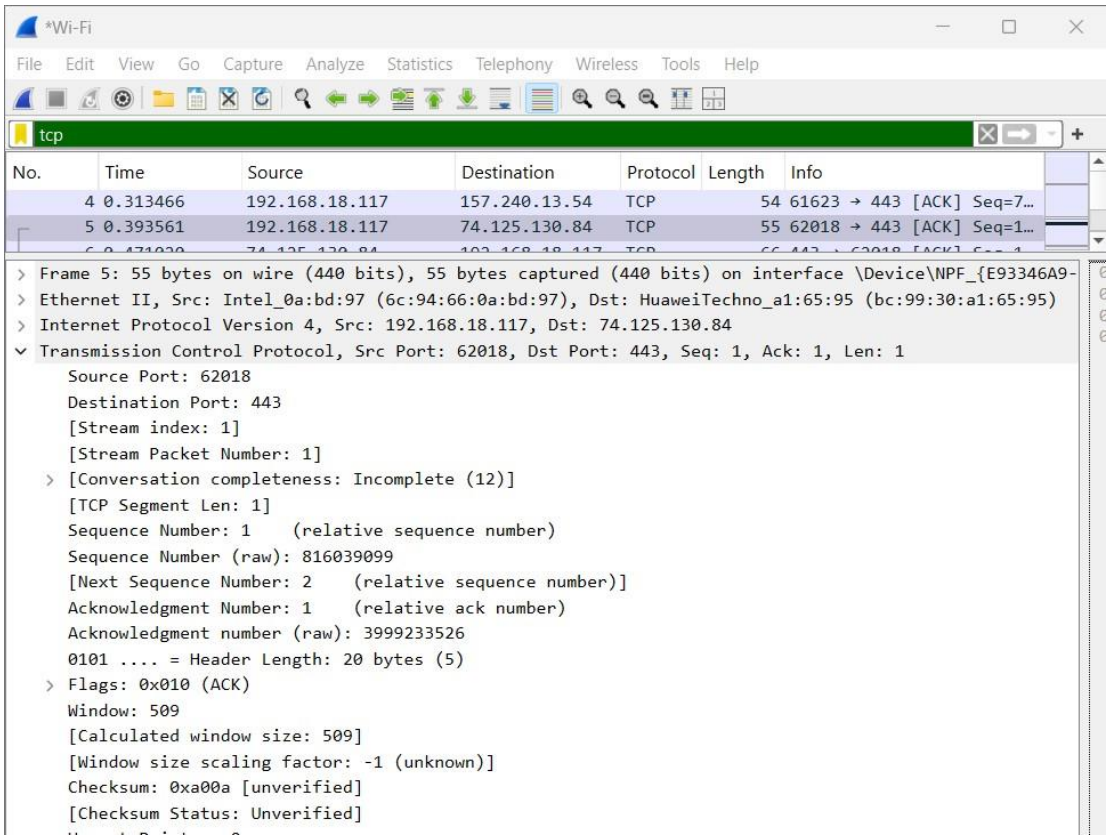


Fig: TCP header analysis of selected packet

TCP Header Analysis Result:
From above figure of TCP header analysis, we can deduce the following details for thewebsite youtube.com

SN	Parameters	Details
1	Source Port	62018
2	Destination Port	443
3	Sequence Number	1
4	Acknowledgment Number	1
5	Flags	ACK

Fig: TCP header analysis details table

UDP Header Analysis

Select a UDP packet and analyze its header:

Source Port: The port on the sender's side.

Destination Port: The port on the receiver's side.

Length: Indicates the size of the UDP packet, including the header and data.

Checksum: A verification field for ensuring data integer

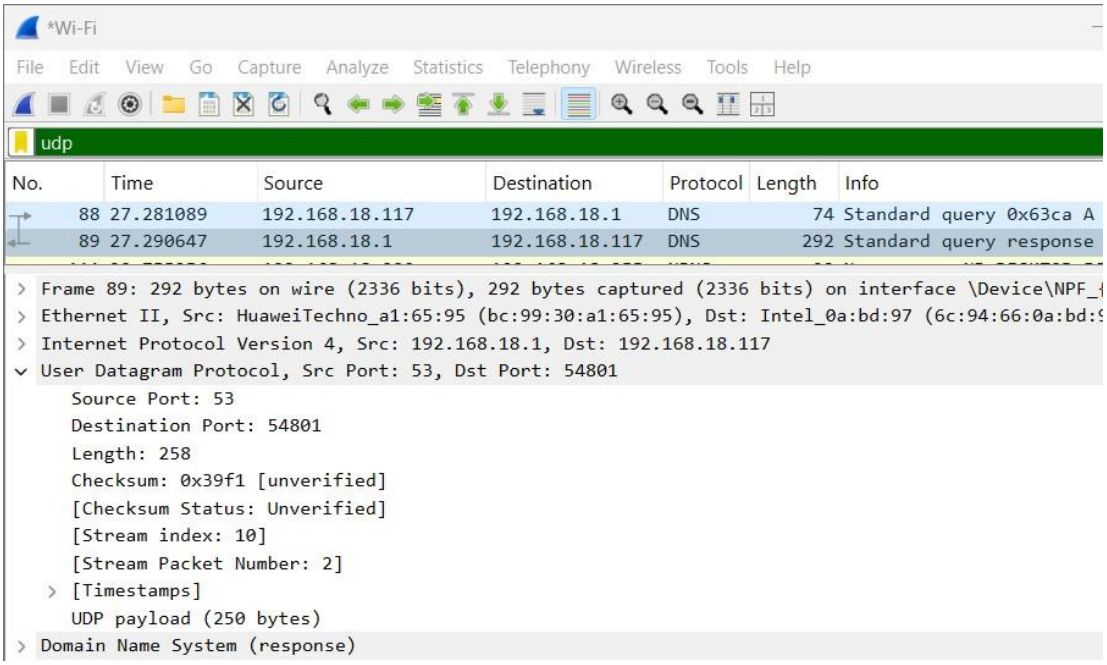


Fig: UDP header analysis of selected packet

UDP Header Analysis Result:
From above figure of UDP header analysis, we can deduce the following details for thewebsite youtube.com

SN	Parameters	Details
1	Source Port	53
2	Destination Port	54801
3	Length	258
4	Checksum	0x39f1
5	Stream Index	10
6	Stream Packet Number	2

Fig: UDP header analysis details table

IP Header Analysis

For IP packet analysis, the following fields are important:

Source IP: The sender's IP address.

Destination IP: The receiver's IP address.

Header Length: Indicates the size of the IP header.

TTL (Time to Live): Limits the lifespan of the packet, decremented by each router.

Protocol: Specifies whether TCP, UDP, or another protocol is being used.

Fragmentation: If the packet is fragmented, this field shows relevant information.

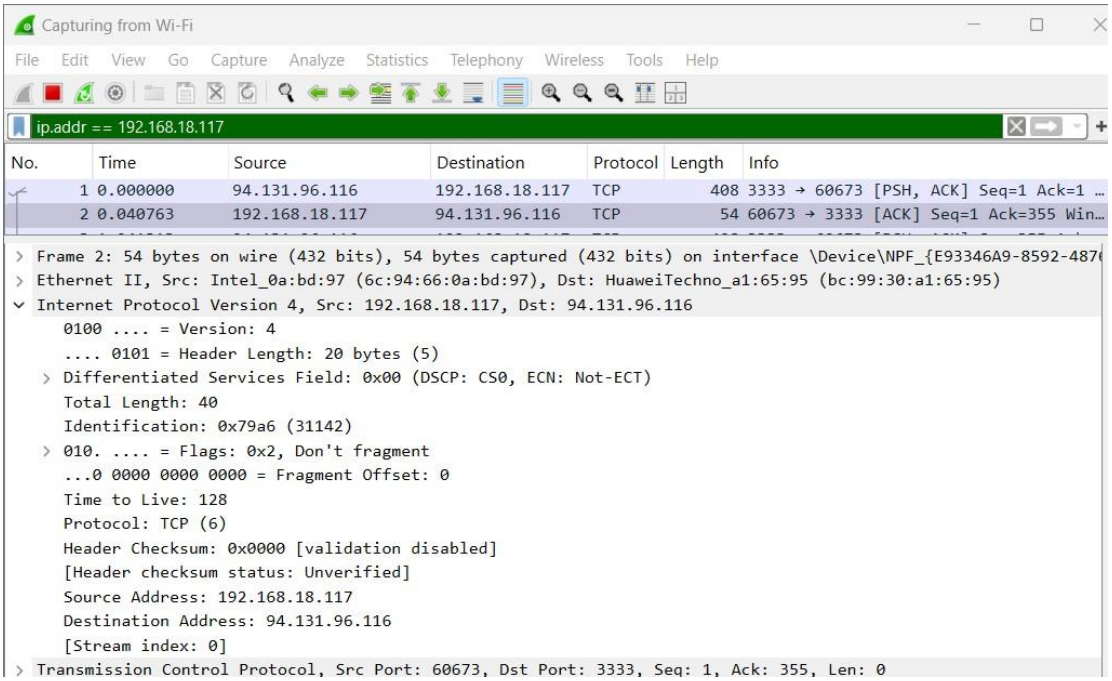


Fig: IP header analysis of selected packet

IP Header Analysis Result:
From above figure of IP header analysis, we can deduce the following details for thewebsite youtube.com

SN	Parameters	Details
1	Source address	192.168.18.117
2	Destination address	93.131.96.116
3	Length	40
4	Time To Live	128
5	Header Checksum	0x0000
6	Stream Index	0
7	Protocol	TCP (6)

Fig: IP header analysis details table

Conclusion

In this lab, we examined Packet Capture and Header Analysis using Wireshark, concentrating on the fundamental protocols TCP, UDP, and IP. By capturing real-time network traffic and analyzing packet headers, we gained valuable insights into how data is transmitted and handled across networks. This practical experience is crucial for understanding the functioning of different protocols, helping us troubleshoot network problems, improve performance, and strengthen security by closely inspecting packet-level details like ports, IP addresses, and control flags.