

LAB 1:

Understanding of Network Equipment, Wiring in Details (CAT6 UTP EIA/TIA 568A/B Straight and Cross-Over Wiring and Testing)

Objective(s):

- To understand networking equipment, including repeaters, hubs, bridges, switches, routers, crimpers, UTP, fiber cables, connectors, patch panels, cable managers, racks, CAT6 straight and crossover wiring standards, LAN meters/testers, and RJ-45 connectors.
 - To understand the color-coding standard of UTP cables.
 - To create both straight and crossover cables and verify their connectivity using a LAN tester.
-

a. Understanding Networking Equipment:

1. Repeater:

- Devices operating at the physical layer of the OSI model.
- Their main function is to regenerate and amplify signals to extend the distance over which the data can travel without degradation.
- Commonly used in conjunction with hubs.

2. Hubs:

- A hub is a basic networking device used to connect multiple computers in a LAN.
- Hubs operate at the physical layer (Layer 1) of the OSI model and simply forward data received on one port to all other ports.
- **Types:**
 - **Passive Hubs:** Forward data without modifying it, require no power supply.
 - **Active Hubs:** Amplify and regenerate the data signal, acting as multiport repeaters.
- **Drawbacks:** Hubs do not filter traffic, which means all connected devices receive the same data, leading to inefficient bandwidth usage.

3. Switches:

- More intelligent devices than hubs, switches operate at the data link layer (Layer 2).
- They forward data only to the intended destination based on the MAC address, thereby reducing network traffic and segmenting the collision domain.
- Switches maintain a MAC address table to determine the destination of each data packet.
- **Benefits:** This selective forwarding reduces collisions and makes LANs more efficient by ensuring that only the destination machine receives the data.

4. Bridges:

- Used to extend and segment a network, bridges also work at the data link layer.
- They can filter traffic between network segments by looking at MAC addresses but are generally slower than switches because they perform software-based switching.
- **Comparison with switches:** While both devices filter traffic and reduce collision domains, switches perform hardware-based switching, making them faster and more efficient than bridges.

5. Routers:

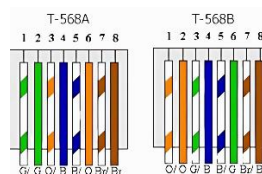
- Routers operate at the network layer (Layer 3) and are used to connect different networks, such as LANs to WANs or LAN to the internet.
- They route data based on IP addresses and determine the best path for data packets to travel between networks.
- Routers control both collision and broadcast domains and employ Network Address Translation (NAT) to allow multiple devices on a LAN to share a single public IP address.

6. Gateways:

- Gateways connect networks with different architectures and protocols, often operating at the application layer.
- They are more complex than routers, performing protocol conversions to enable communication between systems using different networking protocols, such as TCP/IP and AppleTalk.

b. Understanding the Color-Coding Standard of UTP Cable:

- UTP cables consist of four pairs of twisted wires, making a total of eight wires. The wires are color-coded and must be arranged in a specific order based on the EIA/TIA 568A or 568B wiring standards.
- **Color-Coding Rules:**
 1. **Odd-numbered positions** always hold the partially colored (striped) wire, while **even-numbered positions** hold the solid-colored wire.
 2. The main difference between T568A and T568B standards is the arrangement of the **green** and **orange** pairs.
 3. In T568A, the first pair (positions 1 and 2) is green, while in T568B, it is orange.
 4. The positions of wires 4, 5, 7, and 8 are fixed (blue and brown pairs) in both standards.
- **Wiring Patterns:**
 1. **T568A Standard:** Starts with green (pin 1: green-white, pin 2: green).
 2. **T568B Standard:** Starts with orange (pin 1: orange-white, pin 2: orange).



c. Creating Straight and Crossover Cables and Verifying Connectivity:

Apparatus:

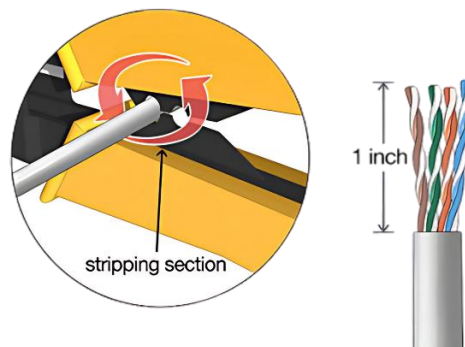
- UTP CAT6 cable (1 meter or more), RJ-45 connectors, crimper tool, LAN tester.



Steps for Creating a Straight Cable:

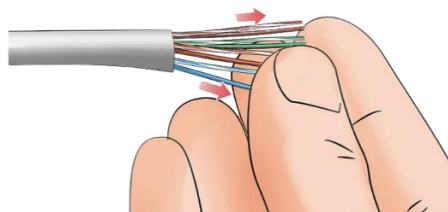
1. Strip the Cable Jacket:

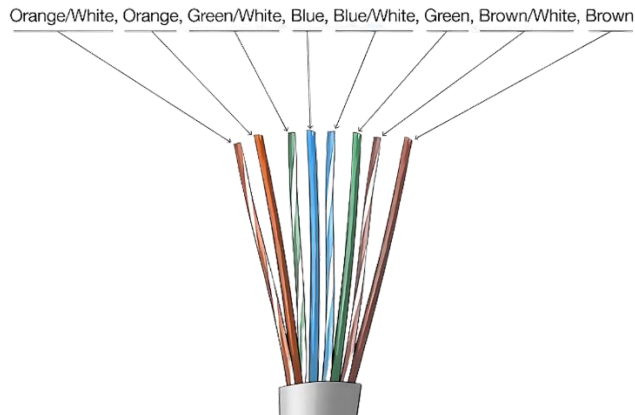
- Use a cable stripper to remove about ~1.2 inches of the outer jacket, exposing the twisted pairs of wires inside.
- If working with CAT6 cable, you may encounter a spine inside the cable which should be trimmed.



2. Untwist and Align the Wires:

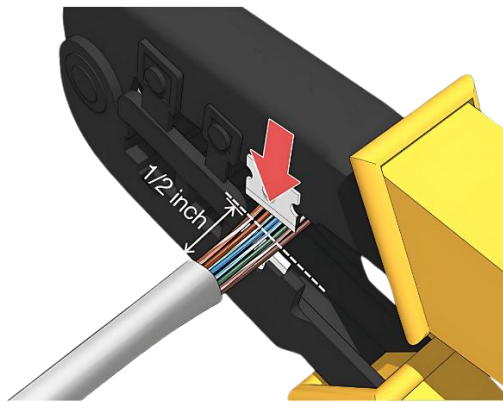
- Untwist the four pairs of wires and spread them out.
- Align them according to the T568B or T568A standard (whichever is required for the task).





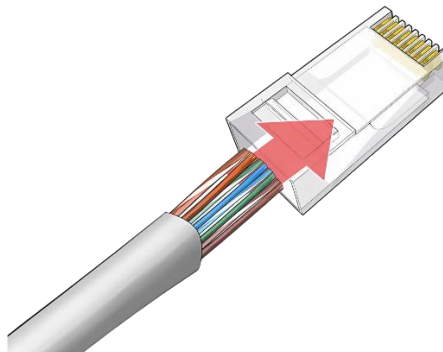
3. Trim the Wires:

- After aligning the wires, cut them evenly so they are about 0.5 inch above the jacket.



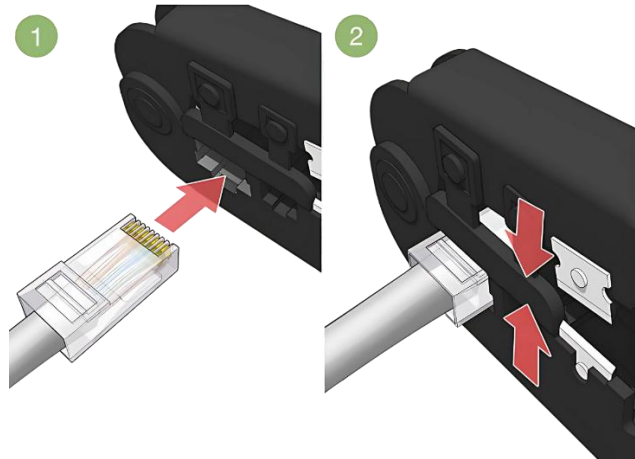
4. Insert Wires into the RJ-45 Connector:

- Push the wires into the connector, ensuring each wire follows its assigned slot inside the connector.
- Make sure the wires reach the end of the connector for proper contact.



5. Crimp the Connector:

- Insert the connector into the crimping tool and squeeze tightly to secure the wires in place.



6. Repeat for the Other End:

- Perform the same process on the other end of the cable to complete the straight-through cable.

7. Test the Cable:

- Use a LAN cable tester to check each pin's connectivity. The tester will verify whether the cable has been properly terminated at both ends.

Steps for Creating a Crossover Cable:

- Follow the same steps as for the straight cable, except:
 - **One end** of the cable should follow the T568A wiring standard.
 - **The other end** should follow the T568B wiring standard.

Verification:

- After both ends are crimped, use the LAN tester to verify the connections.
- The LAN tester should show that each pin corresponds correctly with its counterpart (e.g., pin 1 on one end connects to pin 3 on the other end for crossover cables).

OS Installation in a Virtual Machine using VMWare Workstation

Objective(s):

- To download and install a Linux distribution in a virtual machine using **VMWare Workstation**.
 - I will be installing **Fedora** as my chosen Linux distribution.
 - To understand the steps involved in setting up a virtual machine, installing an operating system, and configuring it for use.
 - To explore the concept of virtualization and how it enables multiple operating systems to run concurrently on a single physical machine.
-

Introduction:

In this lab, we will install **Fedora**, a popular Linux distribution, in a virtual machine using **VMWare Workstation**. Fedora is chosen for several reasons:

1. **Cutting-Edge Features:** Known for incorporating the latest technologies and software innovations, Fedora provides users access to cutting-edge tools and applications, ideal for developers and enthusiasts.
2. **Community Support:** As a community-driven project, Fedora boasts a vibrant user community, ensuring a wealth of resources, forums, and documentation are available for troubleshooting and learning.
3. **Stability and Performance:** Fedora is recognized for its stability and performance, utilizing the latest Linux kernel for improved functionality and hardware support.
4. **Development Environment:** It offers a robust environment for development, with support for various programming languages, frameworks, and tools, making it suitable for application development.
5. **Default Desktop Environment:** Fedora Workstation features the user-friendly **GNOME** desktop environment, enhancing the overall user experience.
6. **Frequent Updates:** With a regular release cycle, Fedora ensures users receive the latest software and security updates, prioritizing system freshness.
7. **Educational Value:** It serves as an excellent platform for learning about Linux, including package management, system configurations, and shell scripting.

By installing Fedora in a virtual environment, we can explore these features without affecting our host system, allowing for safe experimentation and learning. This report will detail the step-by-step process of creating the virtual machine, installing Fedora, and configuring it for practical use.

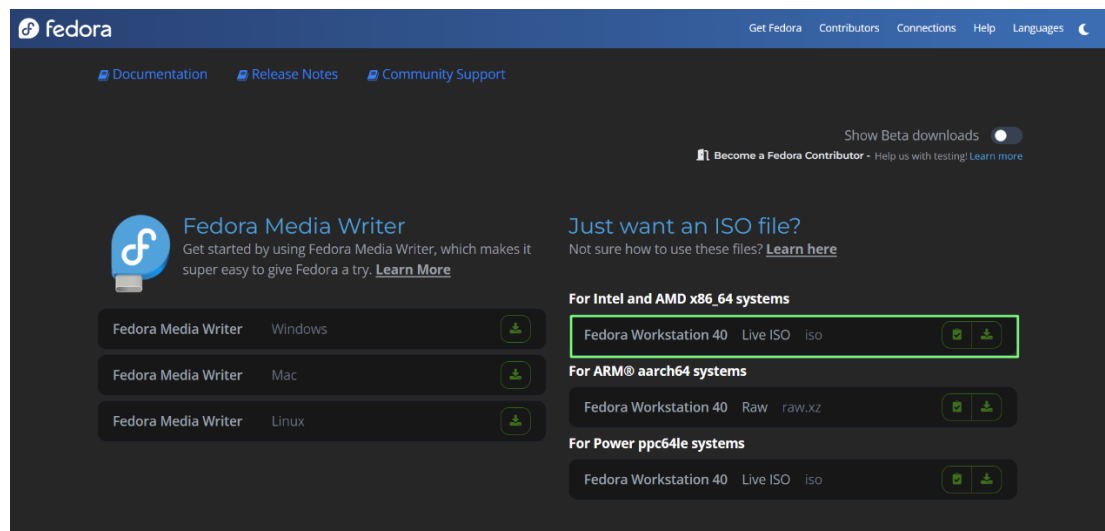
Apparatus:

1. **VMWare Workstation** installed on your host machine.
 2. **Fedora ISO** file downloaded from the official Fedora website.
 3. A **PC/Laptop** running any OS (Windows, Ubuntu, or CentOS) as the host system.
 4. **Minimum System Requirements** for VMWare Workstation:
 - 64-bit processor
 - 2 GB RAM (Recommended: 4 GB or more)
 - 2 GB disk space for the application
 - Enough disk space for Fedora installation (~20 GB or more)
-

Steps:

1. Download Fedora ISO:

- Go to **Fedora Official Website**: <https://fedoraproject.org/workstation/download> and download the latest version of Fedora (Workstation or Server Edition).
- Choose the architecture matching your system (64-bit for most PCs).



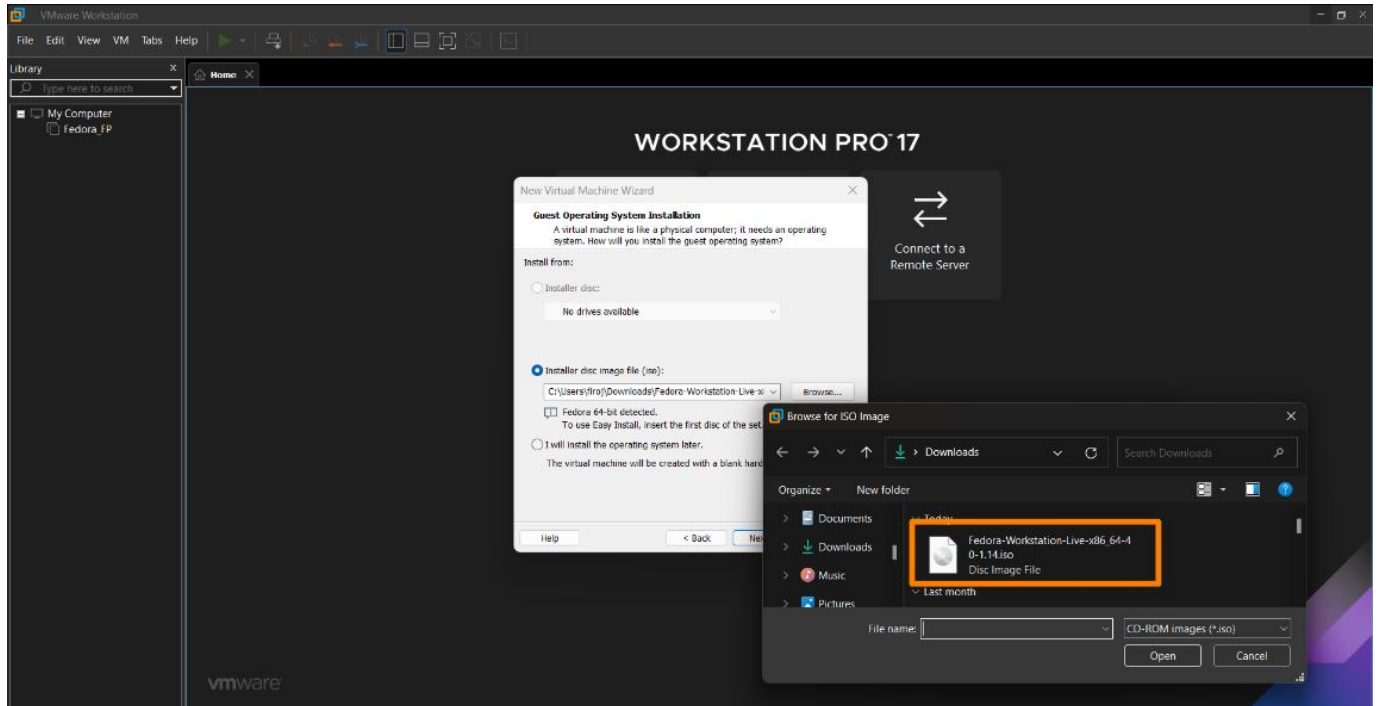
2. Install VMWare Workstation:

- Download and install **VMWare Workstation** from the official VMWare website if it's not already installed.
- Follow the installation prompts and launch the software.

3. Create a New Virtual Machine in VMWare:

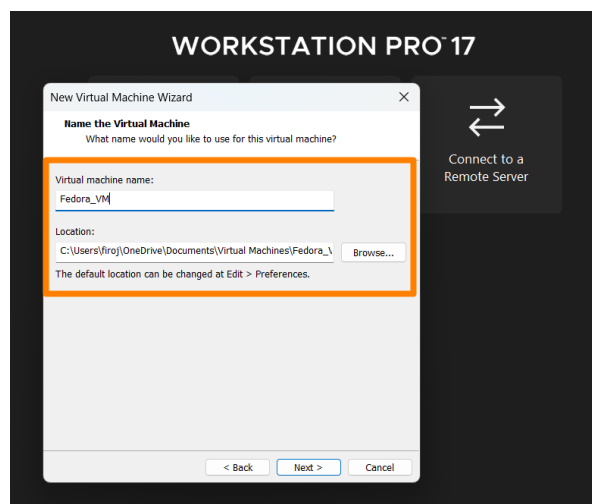
- **Step 1:** Open VMWare Workstation and click on **Create a New Virtual Machine**.
- **Step 2:** Choose **Custom** (advanced) setup to get more control over the configuration or **Typical** for a standard configuration.

- **Step 3:** Choose **Installer disc image file (ISO)** and browse to the downloaded **Fedora ISO** file.



4. Configuring the Virtual Machine:

- **Step 1:** Name the virtual machine (e.g., Fedora_VM) and choose the location to store the virtual machine files.
- **Step 2: Specify the Disk Capacity:** Create a new virtual hard disk of at least 20 GB (recommended: 30 GB or more).
- **Step 3:** Click on **Customize Hardware** to customize the Memory (Selected 2GB by default), Network Adapter and other devices as needed
- **Step 4:** Then click **Finish** to create the virtual machine.



5. Booting and Installing Fedora:

- **Step 1:** Start the newly created virtual machine from the **VMWare dashboard**—Select the Newly created Virtual Machine and click on “Power on this Virtual Machine”
- **Step 2:** The virtual machine will boot from the Fedora ISO image, and you will see the Fedora installer.
- **Step 3:** Select **Start Fedora** from the boot menu.

6. Fedora Installation Process:

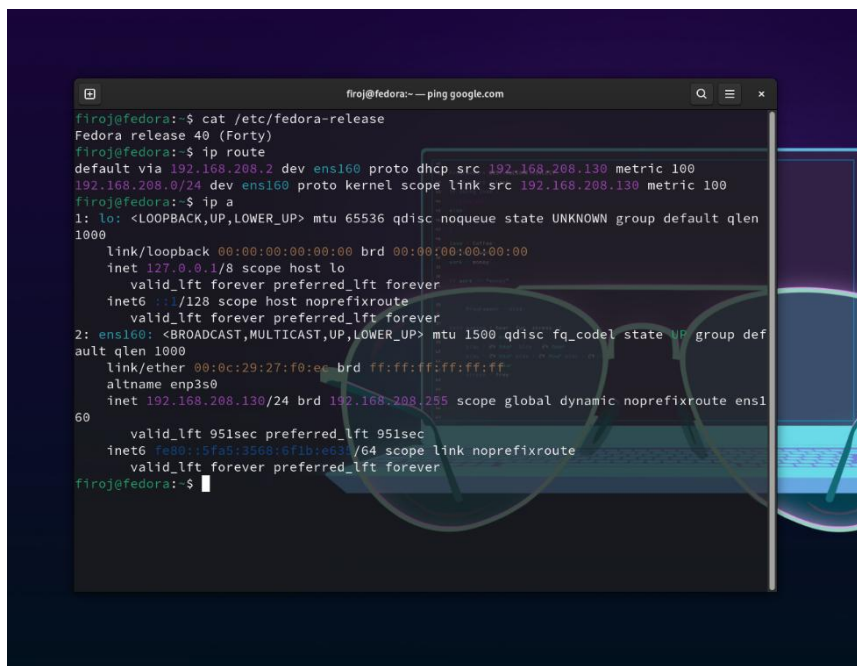
- **Step 1:** Choose your **language** and click **Continue**.
- **Step 2:** Select your **installation destination** (the virtual disk you created earlier). Use the automatic partitioning option unless you need a custom setup.
- **Step 3:** Choose your **time zone** and **keyboard layout** as per your region. (Automatic detection done in Fedora)
- **Step 4:** Set the **root password** and create a **user account** for yourself with administrative privileges.
- **Step 5:** Click **Begin Installation** and let the installation complete. This process might take several minutes depending on your system resources.

7. Testing the Installation:

- Verify that Fedora is installed and working correctly by performing some basic operations:
 - Open a terminal and run system commands.
 - Check network connectivity.
 - Update the system using the command:

```
sudo dnf update
```

- Install essential software packages (e.g., web browser, text editor).



```
firoj@fedora:~$ ping google.com
firoj@fedora:~$ cat /etc/fedora-release
Fedora release 40 (Forty)
firoj@fedora:~$ ip route
default via 192.168.208.2 dev ens160 proto dhcp src 192.168.208.130 metric 100
192.168.208.0/24 dev ens160 proto kernel scope link src 192.168.208.130 metric 100
firoj@fedora:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:27:f0:e0 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.208.130/24 brd 192.168.208.255 scope global dynamic noprefixroute ens160
        valid_lft 951sec preferred_lft 951sec
    inet6 fe80::15fa5:3568:6f1b:e63_/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
firoj@fedora:~$
```

Result:

- Fedora was successfully installed on the virtual machine using **VMWare Workstation**. The system is operational, and necessary configurations were completed.

Conclusion:

This lab allowed us to gain hands-on experience in setting up a virtual machine, installing Fedora, and configuring it for use. We also learned how VMWare Workstation facilitates the creation of isolated environments for operating system installations, making it an excellent tool for OS testing and development environments.

Precautions:

1. Ensure that your host system meets the minimum hardware requirements before proceeding with installation.
2. Always verify the downloaded ISO file for integrity to avoid corrupted installations.
3. Allocate sufficient resources (CPU, RAM, and disk space) to the virtual machine for optimal performance.

Overview of IP Addressing and Subnetting

Objective(s):

- To gain theoretical knowledge of IPv4 addressing and subnetting.
-

Background

An IP address (Internet Protocol address) is a numerical representation that uniquely identifies a specific interface on a network. In IPv4, addresses are 32 bits long, allowing for a maximum of 4,294,967,296 unique addresses. In contrast, IPv6 addresses are 128 bits, accommodating 3.4×10^{38} unique addresses. While IP addresses are binary numbers, they are typically expressed in decimal form (for IPv4) or hexadecimal form (for IPv6) to facilitate easier human readability.

Terminologies

- **IPv4 Address:** A 32-bit number, usually written in dotted decimal form, that uniquely identifies a network interface.
- **Host Address:** Another term for an end device's IP address.
- **Network:** A group of hosts with identical starting positions in their IP addresses.
- **Broadcast Address:** A 32-bit number used to address all hosts in a network; it cannot be assigned to a host.
- **Subnet:** A subdivision of a network, where all hosts share an identical portion of their IP addresses.
- **Subnetting:** The process of dividing networks into smaller subnets.
- **Subnet Mask:** A 32-bit combination that indicates which portion of an address refers to the subnet and which part refers to the host.

IPv4 Address Representations

IPv4 addresses are 32-bit binary numbers, generally represented as four octets (0-255) in decimal form. The address consists of two main parts: the **Network Number** and the **Host Number**. The network number identifies the network segment, while the host number identifies individual devices within that segment.

Subnet Masks

A subnet mask delineates where the network part of an IP address ends and the host portion begins. In binary, bits set to one signify the network address, while bits set to zero signify the host address. Common default subnet masks include:

- Class A: 255.0.0.0

- Class B: 255.255.0.0
- Class C: 255.255.255.0

IP Address Classes

The IPv4 address space is divided into five classes:

- **Class A:**
 - First bit: 0
 - Range: 1 - 127
 - Default Mask: 255.0.0.0
 - Networks: 126; Hosts: 16,777,214.
- **Class B:**
 - First two bits: 10
 - Range: 128 - 191
 - Default Mask: 255.255.0.0
 - Networks: 16,384; Hosts: 65,534.
- **Class C:**
 - First three bits: 110
 - Range: 192 - 223
 - Default Mask: 255.255.255.0
 - Networks: 2,097,152; Hosts: 254.
- **Class D:**
 - First four bits: 1110
 - Range: 224 - 239
 - Reserved for multicasting; no subnet mask.
- **Class E:**
 - First four bits: 1111
 - Range: 240 - 255
 - Reserved for experimental purposes; no subnet mask.

Private Addresses

Certain networks are reserved for private use and are not routable on the public internet:

- **Class A:** 10.0.0.0 to 10.255.255.255
- **Class B:** 172.16.0.0 to 172.31.255.255
- **Class C:** 192.168.0.0 to 192.168.255.255

Special Addresses

- **Loopback Address:** 127.0.0.0
- **Broadcast Address:** 255.255.255.255
- **Multicast Address:** 224.0.0.0

IPV4 Subnetting

Classful addressing does not offer flexibility in terms of the number of hosts or networks. CIDR (Classless Inter-Domain Routing) allows for borrowing bits from the host portion to create subnets, enabling better network management.

Exercise:

Address Class Identification:

1. 10.250.1.1 - Class A
2. 150.10.15.0 - Class B
3. 192.14.2.0 - Class C
4. 148.17.9.1 - Class B
5. 193.42.1.1 - Class C
6. 126.8.156.0 - Class A
7. 220.200.23.1 - Class C
8. 230.230.45.58 - Class D
9. 177.100.18.4 - Class B
10. 119.18.45.0 - Class A
11. 249.240.80.78 - Class E
12. 199.155.77.56 - Class C
13. 117.89.56.45 - Class A
14. 215.45.45.0 - Class C
15. 199.200.15.0 - Class C

Network & Host Identification:

Circle the network portion of these addresses:	Circle the host portion of these addresses:
177.100.18.4	10.15.123.50
119.18.45.0	171.2.199.31
209.240.80.78	198.125.87.177
199.155.77.56	223.250.200.222
117.89.56.45	17.45.222.45
215.45.45.0	126.201.54.231
192.200.15.0	191.41.35.112
10.250.1.1	155.25.169.227
150.10.15.0	192.15.155.2
192.14.2.0	123.102.45.254
148.17.9.1	148.17.9.155
193.42.1.1	100.25.1.1
126.8.156.0	195.0.21.98
220.200.23.1	25.250.135.46

Default Subnet Masks, Network Address and Broadcast Address

Write the correct default subnet mask, network address and broadcast address for each of the following addresses:

IP Address	Default Subnet Mask	Network Address	Broadcast Address
177.100.18.4	255.255.0.0	177.100.0.0	177.100.255.255
119.18.45.0	255.0.0.0	119.0.0.0	119.255.255.255
191.249.234.191	255.255.0.0	191.249.0.0	191.249.255.255
223.23.223.109	255.255.255.0	223.23.223.0	223.23.223.255
10.10.250.1	255.0.0.0	10.0.0.0	10.255.255.255
126.123.23.1	255.255.0.0	126.123.0.0	126.123.255.255
223.69.230.250	255.255.255.0	223.69.230.0	223.69.230.255
192.12.35.105	255.255.255.0	192.12.35.0	192.12.35.255
77.251.200.51	255.0.0.0	77.0.0.0	77.255.255.255
189.210.50.1	255.255.0.0	189.210.0.0	189.210.255.255
88.45.65.35	255.0.0.0	88.0.0.0	88.255.255.255
193.100.77.8	255.255.255.0	193.100.77.0	193.100.77.255
125.125.250.1	255.0.0.0	125.0.0.0	125.255.255.255
220.90.130.45	255.255.255.0	220.90.130.0	220.90.130.255
134.125.34.9	255.255.0.0	134.125.0.0	134.125.255.255

Conclusion

This lab provided a foundational understanding of IPv4 addressing and subnetting. It covers key terminologies, classifications of IP addresses, and the significance of subnetting for effective network management.

Connecting the computers in LAN

Objective(s):

- To connect multiple computers in a Local Area Network (LAN).
- To configure Internet Connection Sharing (ICS) and ensure proper communication between devices.

Apparatus (Software):

- Windows Operating System on all participating computers.
- Administrator privileges for configuration.

Objectives:

- To configure the host computer to share the Internet connection.
 - To set up client computers to access the Internet through the shared connection.
-

Procedure:

On the Host Computer:

The host computer serves as the gateway for sharing the Internet connection with other devices in the network. Follow these steps:

1. **Log in as Administrator:**
 - Log on to the host computer using an Administrator or Owner account.
2. **Open Network Settings:**
 - Click on the **Start** menu, then go to **Control Panel**.
 - Navigate to **Network and Internet Connections** and then **Network Connections**.
3. **Select the Internet Connection:**
 - Right-click on the Internet connection you want to share. This could be a dial-up connection or another available network.
 - Select **Properties** from the context menu.
4. **Enable Internet Connection Sharing (ICS):**
 - Click the **Advanced** tab in the connection properties dialog box.
 - Under the **Internet Connection Sharing** section, check the box for **Allow other network users to connect through this computer's Internet connection**.
 - If using a dial-up connection, optionally select **Establish a dial-up connection whenever a computer on my network attempts to access the Internet** to allow automatic Internet connection.
5. **Apply and Confirm Changes:**

- Click **OK**. A message may appear stating:

When Internet Connection Sharing is enabled, your LAN adapter will be set to use IP address 192.168.0.1. Your computer may lose connectivity with other computers on your network. If these other computers have static IP addresses, it is a good idea to set them to obtain their IP addresses automatically. Are you sure you want to enable Internet Connection Sharing?

- Click **Yes** to confirm.

6. Result:

- The Internet connection is now shared with other computers on the LAN.
- The network adapter connected to the LAN is configured with a static IP address of **192.168.0.1** and a subnet mask of **255.255.255.0**.

On the Client Computer(s):

Each client computer in the LAN needs to be configured to connect to the shared Internet connection provided by the host computer. Follow these steps:

1. Log in as Administrator:

- Log on to the client computer using an Administrator or Owner account.

2. Open Network Settings:

- Click **Start**, then navigate to **Control Panel**.
- Go to **Network and Internet Connections** and then **Network Connections**.

3. Access Local Area Connection Properties:

- Right-click on **Local Area Connection** and select **Properties** from the context menu.

4. Configure TCP/IP Settings:

- On the **General** tab, select **Internet Protocol (TCP/IP)** from the list of connection items and click **Properties**.
- Choose **Obtain an IP address automatically** (recommended) to enable dynamic IP assignment.

5. Optional: Assign a Static IP Address:

- If you prefer to use a static IP address, assign values within the range of **192.168.0.2** to **192.168.0.254**. For example:
 - **IP Address:** 192.168.31.202
 - **Subnet Mask:** 255.255.255.0
 - **Default Gateway:** 192.168.31.1
- Click **OK** to save changes.

6. Apply and Test:

- Close the **Local Area Connection Properties** dialog box by clicking **OK**.
- Quit the Control Panel.

Results and Observations:

1. Successfully established a Local Area Network where client computers can access the Internet through the host computer.
2. The host computer was configured to use the IP address **192.168.0.1** with a subnet mask of **255.255.255.0**.
3. Client computers were able to dynamically obtain IP addresses or use static IPs within the designated range to ensure proper connectivity.

Conclusion: This practical demonstrates the procedure for setting up a Local Area Network and sharing an Internet connection among multiple computers. The configuration involves enabling Internet Connection Sharing on the host computer and setting up the network adapter properties on the client computers. Proper understanding of IP addressing and network settings ensures seamless connectivity and optimal performance.

Study of basic network command and network configuration commands

Objective(s):

Study of basic network command and network configuration commands.

Apparatus (Software):

Command Prompt / Packet Tracer.

Procedure:

To perform this experiment, follow these steps:

1. Understanding Basic Networking Commands:

- The goal is to familiarize students with essential networking commands such as ping and tracert.
- Students will also learn all commands related to network configuration, including switching between privilege and normal modes, configuring router interfaces, and saving configurations to flash or permanent memory.

2. Categories of Commands to Study:

- Configuring the Router Commands.
- General Commands to Configure Network.
- Privileged Mode Commands of a Router.
- Router Processes & Statistics.
- IP Commands.
- Other IP Commands (e.g., show ip route).

Detailed Study of Commands:

1. Configuring the Router Commands:

- Access the router through CLI (Command-Line Interface).
- Use the enable command to enter privileged mode.
- To configure the interface, use configure terminal followed by interface and the specific interface name (e.g., interface FastEthernet0/0).
- Assign an IP address using ip address <IP_Address> <Subnet_Mask>.
- Activate the interface with the no shutdown command.

2. General Commands to Configure Network:

- Use the `hostname` command to set a router name.
- Set a password for privileged mode with `enable secret <password>`.
- Save configurations using `copy running-config startup-config`.

3. Privileged Mode Commands of a Router:

- `show running-config`: Displays the current configuration.
- `show startup-config`: Displays the saved configuration.
- `debug ip`: Enables debugging for IP protocols.

4. Router Processes & Statistics:

- Use the `show processes` command to view running processes.
- View CPU and memory usage with `show version`.

5. IP Commands:

- `ping <IP_Address>`: Verifies connectivity to a specific IP address.
- `tracert <IP_Address>`: Tracks the route a packet takes to reach a destination.
- `show ip interface brief`: Displays a summary of all interfaces and their status.

6. Other IP Commands:

- `show ip route`: Displays the routing table.
 - `show ip protocols`: Displays protocols and related information.
-

Command Descriptions:

1. ping:

- The `ping` command sends an ICMP `ECHO_REQUEST` packet to a specified host. If the host is reachable, an ICMP reply is received, confirming connectivity.
- Usage example: `ping 8.8.8.8` checks the connectivity to the Google DNS server.

```
C:\>ping 192.168.101.1

Pinging 192.168.101.1 with 32 bytes of data:
Reply from 192.168.101.1: bytes=32 time=1ms TTL=64
Reply from 192.168.101.1: bytes=32 time=1ms TTL=64
Reply from 192.168.101.1: bytes=32 time=1ms TTL=64
Reply from 192.168.101.1: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.101.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

2. tracert:

- The tracert command (short for trace route) shows the path packets take to reach a specific destination.
- It lists all intermediary routers (hops) and measures the time each hop takes.
- Usage example: tracert www.example.com tracks the route to the domain www.example.com.

```
C:\>tracert 192.168.101.1

Tracing route to 192.168.101.1 [192.168.101.1]
over a maximum of 30 hops:

  1      1 ms      1 ms      1 ms  192.168.101.1 [192.168.101.1]

Trace complete.
```

Conclusion: This experiment provided practical exposure to basic and advanced network configuration commands. By learning commands such as ping, tracert, and router configurations, students gained insights into verifying network connectivity, tracing routes, and managing routers effectively. Mastery of these commands is essential for networking and system administration tasks.

Introduction to Packet Tracer: Basic Router Configuration

Objective(s):

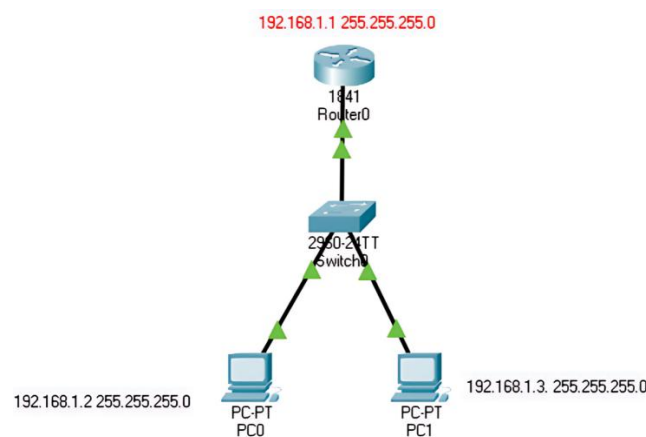
- To understand basic commands for router configuration.
-

Background:

Packet Tracer is a powerful network simulator created by Cisco Systems. It is utilized in training for network certification and learning by allowing students to create networks with numerous devices and experience troubleshooting without the need for real Cisco routers or switches. The purpose of Packet Tracer is to offer students a tool to learn the principles of networking.

Router Modes

Router>	User mode
Router#	Privileged mode
Router(config)#	Global configuration mode
Router(config-if)#	Interface mode
Router(config-subif)#	Subinterface mode
Router(config-line)#	Line mode
Router(config-router)#	Router configuration mode



Configuration Steps:

Configure the following in Router and Switch as illustrated in the figure:

1. Change Hostname (Cisco)
2. Configure passwords (password: cisco & secret: class)
3. Secure Console Port and Terminal lines (password: cisco)
4. Encrypt Passwords (service password-encryption)
5. Configure Clock (clock)
6. Configure Banners (banner motd)
7. Configure Interface (IP Address) on Router (interface fa0/0 or fa0/1)
8. Configure VLAN on Switch (interface vlan 1)
9. Save configurations (running-config to startup-config)
10. Use show commands:

```
show running-config
show startup-config
show ip interface brief
show interface vlan 1
```

11. Configure PCs
12. Verify Connectivity (ping)

Router Configuration:

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Cisco
Cisco(config)#enable password cisco
Cisco(config)#enable secret class
Cisco(config)#line console 0
Cisco(config-line)#password cisco
Cisco(config-line)#login
Cisco(config-line)#line vty 0 4
Cisco(config-line)#password cisco
Cisco(config-line)#login
Cisco(config-line)#exit
Cisco(config)#service password-encryption
Cisco(config)#exit
Cisco#
%SYS-5-CONFIG_I: Configured from console by console

Cisco#clock ?
set Set the time and date
Cisco#clock set ?
hh:mm:ss Current Time
Cisco#clock set 21:15:00 ?
<1-31> Day of the month
MONTH Month of the year
Cisco#clock set 21:15:00 May ?

<1-31> Day of the month
Cisco#clock set 21:15:00 May
% Incomplete command.
```

```

Cisco#clock set 21:15:00 May ?
<1-31> Day of the month
Cisco#clock set 21:15:00 May 15 ?
<1993-2035> Year
Cisco#clock set 21:15:00 May 15 2021
Cisco#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco(config)#banner motd $ UNAUTHORISED ACCESS RESTRICTED $
Cisco(config)#interface fastethernet 0/0
Cisco(config-if)#ip address 192.168.1.1 255.255.255.0
Cisco(config-if)#no shutdown

Cisco(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up

Cisco(config-if)#exit
Cisco(config)#exit
Cisco#
%SYS-5-CONFIG_I: Configured from console by console

Cisco#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Cisco#

```

Switch Configuration:

```

Switch>
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface vlan 1
Switch(config-if)#ip address 192.168.1.4 255.255.255.0
Switch(config-if)#no shutdown
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up
Switch(config-if)#

```

Conclusion:

This lab provided a foundational understanding of basic router and switch configuration commands using Packet Tracer. The successful completion of configurations helps in managing devices effectively and ensures secure access to network resources.

Configuring a Network topology using Packet Tracer

Objective(s):

To configure a network topology (*Star*) using Packet Tracer software and ensure successful communication between devices.

Apparatus (Software):

- Packet Tracer Software
-

Procedure:

Step 1: Setting Up the Workspace in Packet Tracer

1. Open the Packet Tracer software.
2. From the **Device Types Toolbar**, drag and drop the required network devices onto the workspace:
 - **End Devices:** Select PCs, laptops, or servers as needed.
 - **Networking Devices:** Choose switches and routers based on the topology design.
 - **Connections:** Use appropriate cables (copper straight-through or crossover) to connect the devices.

Step 2: Configuring the Devices

2.1 Assigning IP Addresses to End Devices:

1. Click on each **end device** (PC, laptop, etc.).
2. Go to the **Desktop tab** and select **IP Configuration**.
3. Assign the following:
 - **IP Address:** As per the subnet design.
 - **Subnet Mask:** Automatically set if IP address follows classful addressing.
 - **Default Gateway:** Assign the router's interface IP.

2.2 Configuring the Router:

1. Click on the router.
2. Navigate to the **CLI tab** (Command Line Interface).
3. Enter privileged EXEC mode by typing:


```
enable
```

4. Enter global configuration mode:

```
configure terminal
```

5. Assign IP addresses to the router interfaces:

```
interface [interface_id]  
ip address [IP] [Subnet Mask]  
no shutdown
```

6. Save the configuration:

```
copy running-config startup-config
```

2.3 Configuring the Switch (if applicable):

1. Click on the switch and open the **CLI tab**.
2. Assign an IP address to the management VLAN (optional):

```
enable  
configure terminal  
interface vlan 1  
ip address [IP] [Subnet Mask]  
no shutdown
```

3. Save the configuration.

Step 3: Testing the Connectivity

1. Use the **Add Simple PDU Tool** (envelope icon) from the toolbar to simulate sending a packet between devices.
2. Alternatively, open the **CLI tab** on one of the devices and use the ping command:

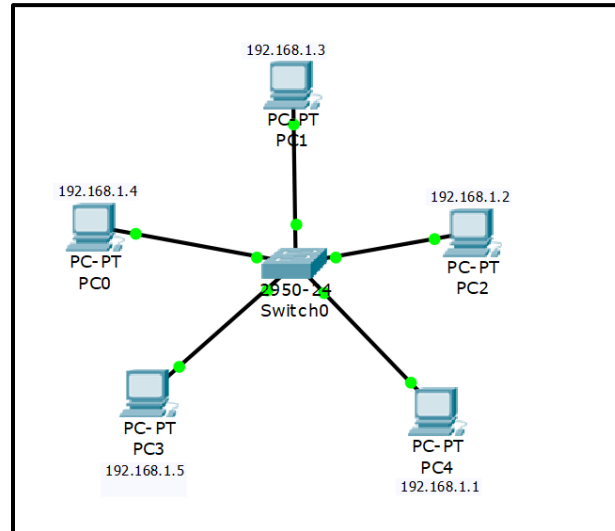
```
ping [Destination IP]
```

3. Verify that packets are successfully sent and received.

Verification:

1. Ensure all devices are connected using appropriate cables and ports.
2. Check for green lights on device interfaces, indicating active connections.
3. Use the simulation mode in Packet Tracer to visualize the packet transmission across the topology.

We will be using Star Topology for this lab report:



PC0
Physical Config Desktop Custom Interface

Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time=0ms TTL=128
Reply from 192.168.1.3: bytes=32 time=0ms TTL=128
Reply from 192.168.1.3: bytes=32 time=1ms TTL=128
Reply from 192.168.1.3: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```

Conclusion:

The network topology was successfully configured using Packet Tracer. Connectivity between devices was validated by sending packets and observing successful replies. The experiment demonstrates the fundamental steps to design and test a functional network.

Configuring a Network Using Distance Vector Routing Protocol (RIP)

Objective(s):

Configuring a network using the Distance Vector Routing Protocol (RIP).

Apparatus (Software):

Packet Tracer Software

Procedure:**1. Introduction to RIP:**

- RIP (Routing Information Protocol) is a Distance Vector Routing Protocol used to determine the best route for data packets within a network. It uses hop count as a metric to decide the shortest path, with a maximum hop count of 15.

2. Topology Design:

- Create a network topology using Packet Tracer.
- Add multiple routers, switches, and end devices (PCs) as required for the design.
- Interconnect devices using appropriate cables (crossover for router-to-router connections and straight-through for router-to-switch or switch-to-PC).

3. Configuring RIP via GUI:

- **Assign IP Addresses:**
 - Assign IP addresses to all router interfaces and end devices in their respective subnets.
 - Ensure each interface IP address is unique within the network.
- **Enable RIP on Routers:**
 1. Open each router's configuration window.
 2. Navigate to the *Routing* tab and select RIP.
 3. Add all connected network addresses to the RIP routing table.
 4. Save the configuration.
- **Verify Connectivity:**
 - Use the simulation tool in Packet Tracer to send a packet from one network segment to another.
 - Ping devices across subnets to ensure proper routing.

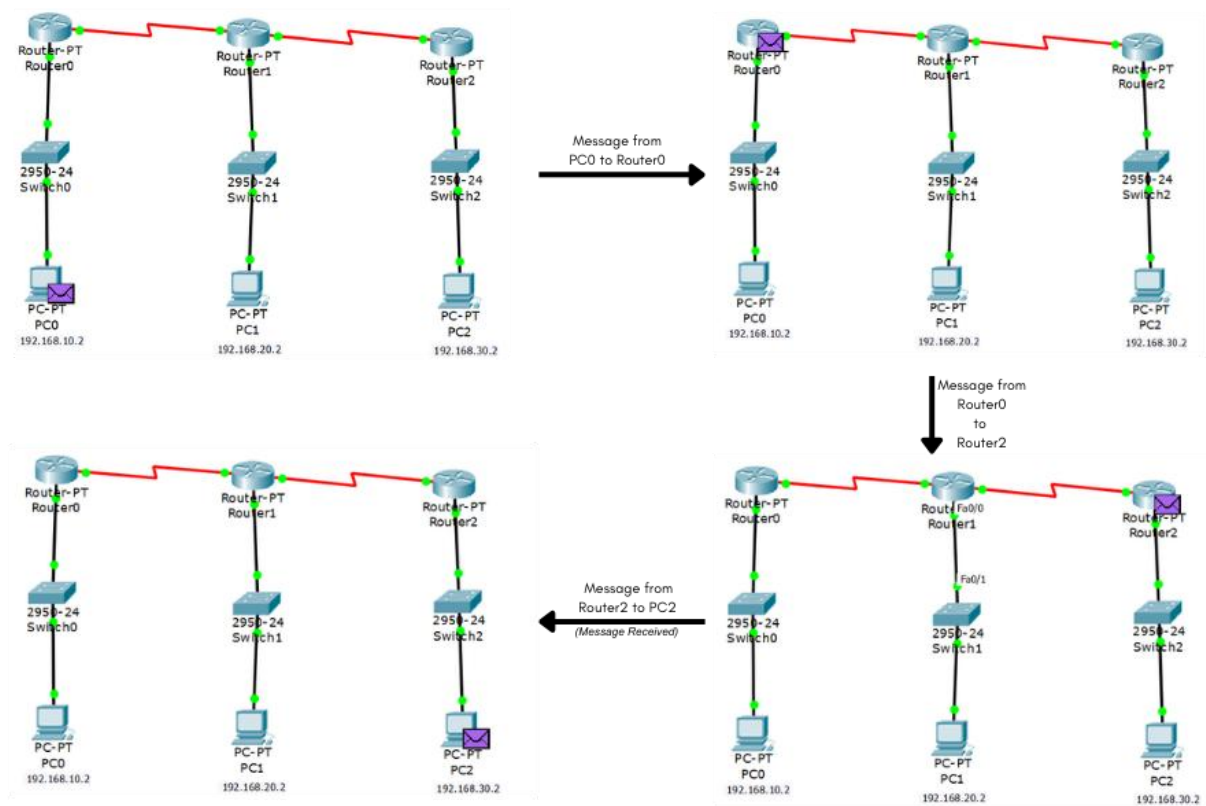
2. Testing the Configuration:

- Verify that routers are exchanging routing information.
- Use the command `show ip route` (optional, via CLI) to check RIP routes on each router.
- Confirm that all connected devices can communicate seamlessly.

Observations:

- All routers successfully exchanged RIP routing tables.
 - Devices within different subnets were able to communicate with each other.
 - Packets followed the expected shortest path based on the hop count.
-

Simulation Diagram (Stepwise):



Conclusion: The network was successfully configured using RIP as the Distance Vector Routing Protocol. Routing tables were populated, and inter-subnet communication was achieved. This practical demonstrated the fundamental concepts of Distance Vector Routing and its application in network design.

Configuring a Network Using Link-State Vector Routing Protocol (OSPF)

Objective(s):

- To understand and configure a network using the link-state routing protocol.
 - To implement Open Shortest Path First (OSPF) routing in a simulated environment.
 - To analyze packet transfers and ensure optimal data routing.
-

Apparatus/Software

- **Software:** Cisco Packet Tracer
-

Theory:

The Open Shortest Path First (OSPF) is a dynamic link-state routing protocol used in IP networks to determine the best path for data transmission. Unlike distance-vector protocols, OSPF uses Dijkstra's algorithm to calculate the shortest path, ensuring efficient routing decisions. It divides an autonomous system into areas and maintains a link-state database (LSDB) to map the network topology.

OSPF operates by:

1. **Establishing Neighbor Relationships:** Routers exchange Hello packets to identify neighbors.
 2. **Building Link-State Advertisements (LSAs):** Routers broadcast information about their directly connected links.
 3. **Calculating Shortest Paths:** Using the LSDB, routers compute the optimal routes.
 4. **Updating Dynamically:** Changes in the topology trigger updates, ensuring consistent network efficiency.
-

Procedure:

Step 1: Setting Up the Network Topology

- Open Cisco Packet Tracer and create a new project.
 - Arrange routers, switches, and end devices (e.g., PCs) in a structured topology.
 - Use appropriate cables (straight-through or crossover) for interconnections.
-

Step 2: Configuring Basic Router Settings

- Assign IP addresses to all router interfaces and connected devices.
- Plan subnets and ensure logical IP distribution.
- Set up basic configurations such as hostname and interface activation on each router.

Step 3: Enabling OSPF Routing Protocol

- Access the configuration terminal of each router.
- Enable OSPF and assign a unique process ID.
- Define OSPF areas and associate networks with specific areas.

Step 4: Verifying OSPF Configuration

- Verify neighbor relationships using commands to display OSPF neighbor tables.
- Inspect routing tables to confirm the inclusion of OSPF routes.

Step 5: Testing Connectivity

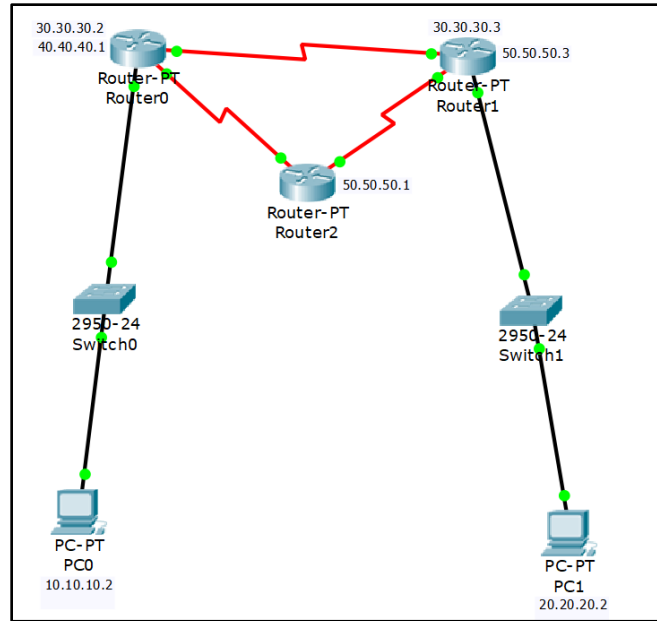
- Use tools such as ping and traceroute to validate end-to-end communication.
- Confirm that packets are routed through the shortest path determined by OSPF.

Step 6: Monitoring Packet Transfers

- Activate simulation mode in Cisco Packet Tracer.
- Observe the flow of OSPF-specific packets such as Hello, DBD, Link State Request, and Link State Update.

Observations

- Establishment of OSPF adjacencies and neighbor relationships was successful.
- Efficient path selection ensured minimal latency and robust connectivity.
- Key metrics such as route convergence time and packet delivery ratio were within acceptable limits.



In the above network layout, simulation was created and this was the output:

Simulation Panel					
Event List					
Vis.	Time(sec)	Last Dev	At Device	Type	Info
	0.914	Switch0	PC0	STP	
	1.430	--	Router1	OSPF	
	1.431	Router1	Router2	OSPF	
	1.530	--	Router1	OSPF	
	1.531	Router1	Switch1	OSPF	
	1.532	Switch1	PC1	OSPF	

Conclusion

This lab demonstrated the effective implementation of the OSPF routing protocol in a simulated environment. OSPF's link-state mechanism ensured optimized routing, showcasing its advantages for complex and dynamic networks.

Firewall Implementation, Router Access Control List (ACL)

Objective(s):

- To Understand the Router Firewall: Access Control Lists (ACLs).

Background:

Packet filtering at the network level can be achieved by applying the Access Control Lists (ACLs) at the router called router firewall. ACLs at the router filter the inbound traffic while permitting or denying packets based on source IP/network and destination IP/network, IP, TCP, UDP protocol information.

Generally, we use ACLs to provide a basic level of security for accessing our network. Access lists can allow one host to access a part of the network and prevent another host from accessing the same area. A standard ACL can be used for several purposes. In this lab, we will see how it can be used in controlling unwanted network traffic. With standard ACLs, we can define certain conditions for the network traffic passing through the router.

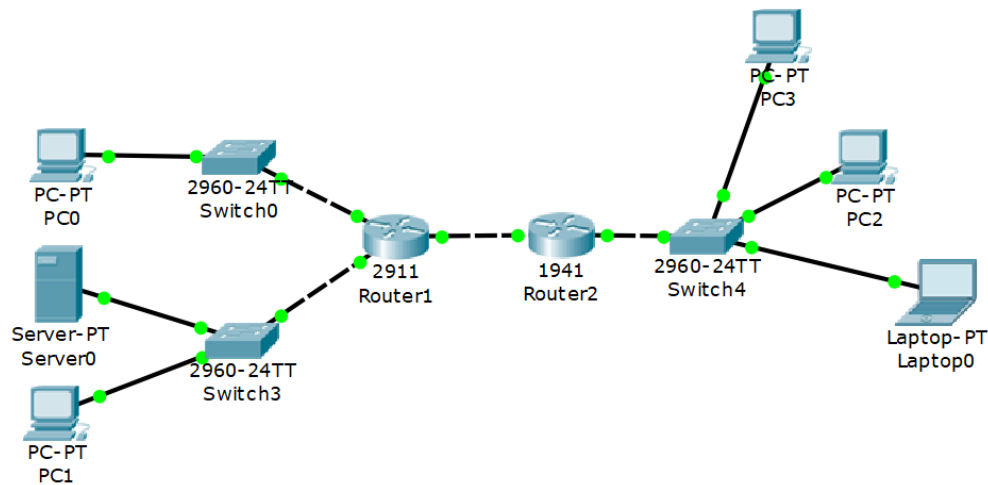
By default, routers do not filter any traffic unless we manually configure an ACL.

Types of ACLs:

1. **Standard ACL:** Permits or denies packets based on the source IP address.
 - **Valid ACL ID range:** 1 - 99.
 - **Applied closest to the destination.**
 - Denies or permits:
 - Source IP Address.
2. **Extended ACL:** Permits or denies packets based on source and destination IP address and IP protocol information.
 - **Valid ACL ID range:** 100 - 199.
 - **Applied closest to the source.**
 - Denies or permits:
 - Source IP Address.
 - Destination IP Address.
 - Port or Service.

Access lists of some protocols must be identified by a name, while others may be identified by a number. These conditions are used in filtering the traffic passing through the router. When creating an access list, we define criteria that are applied to each packet processed by the router, determining whether to forward or block it based on these criteria.

Configuration



Router 1 Configuration:

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)#interface gigabitEthernet 0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
Router(config-if)#interface gigabitEthernet 0/1
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
Router(config-if)#interface gigabitEthernet 0/2
Router(config-if)#ip address 192.168.3.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to up
Router(config-if)#exit
Router(config)#
Router(config)#ip route 192.168.4.0 255.255.255.0 192.168.1.2
Router(config)#
Router#
```

Router 2 Configuration:

```
Router>
Router>enable
Router#configure terminal
```

```

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface gigabitEthernet 0/0
Router(config-if)#ip address 192.168.1.2 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up
Router(config-if)#interface gigabitEthernet 0/1
Router(config-if)#ip address 192.168.4.1 255.255.255.0
Router(config-if)#no shut down
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
Router(config-if)#exit
Router(config)#
Router(config)#ip route 192.168.2.0 255.255.255.0 192.168.1.1
Router(config)#ip route 192.168.3.0 255.255.255.0 192.168.1.1
Router(config)#

```

1. Standard ACL Implementation

a. Blocking a Host (192.168.4.101) in Network 192.168.2.0:

Steps:

1. **Create the Access List (Standard: 1-99):**
 - Specify more specific statements on the top.
 - Specify more general statements at the bottom.
 - Note that at the end of every access-list there is an implicit deny (*e.g., Access-list 1 deny any*).
2. **Apply the Access List to an Interface (Outbound):**

Router 1 Configuration:

```

Router(config)#access-list 1 deny 192.168.4.100 0.0.0.0
Router(config)#access-list 1 permit any
Router(config)#exit
Router#show access-list

```



```

Router(config)#interface gigabitEthernet 0/1
Router(config-if)#ip access-group 1 out
Router(config-if)#exit
Router(config)#exit
Router#

```

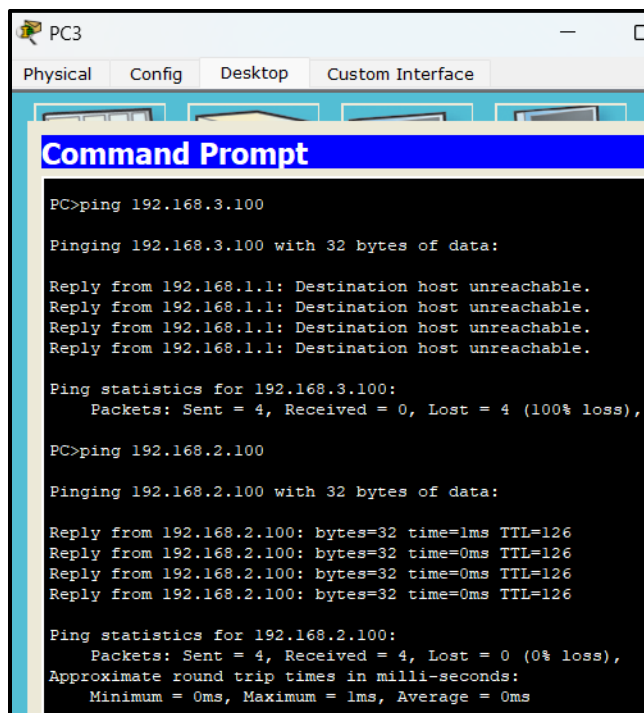
```
%SYS-5-CONFIG_I: Configured from console by console
Router#show run
```



```
Router1
Physical Config CLI
IOS Command Line Interface

!
interface GigabitEthernet0/1
ip address 192.168.3.1 255.255.255.0
ip access-group 1 out
duplex auto
speed auto
!
interface GigabitEthernet0/2
ip address 192.168.2.1 255.255.255.0
duplex auto
speed auto
!
```

Verify the Connectivity



```
PC3
Physical Config Desktop Custom Interface
Command Prompt

PC>ping 192.168.3.100

Pinging 192.168.3.100 with 32 bytes of data:

Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.

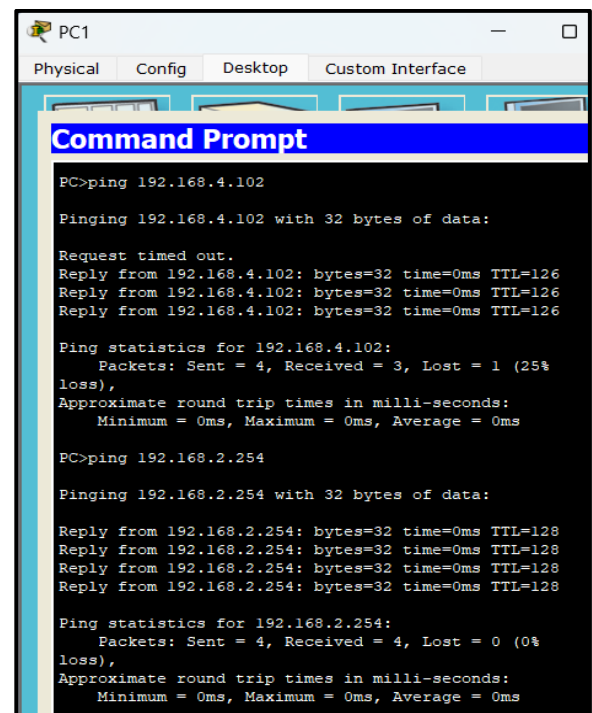
Ping statistics for 192.168.3.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 192.168.2.100

Pinging 192.168.2.100 with 32 bytes of data:

Reply from 192.168.2.100: bytes=32 time=1ms TTL=126
Reply from 192.168.2.100: bytes=32 time=0ms TTL=126
Reply from 192.168.2.100: bytes=32 time=0ms TTL=126
Reply from 192.168.2.100: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.2.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```



```
PC1
Physical Config Desktop Custom Interface
Command Prompt

PC>ping 192.168.4.102

Pinging 192.168.4.102 with 32 bytes of data:

Request timed out.
Reply from 192.168.4.102: bytes=32 time=0ms TTL=126
Reply from 192.168.4.102: bytes=32 time=0ms TTL=126
Reply from 192.168.4.102: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.4.102:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>ping 192.168.2.254

Pinging 192.168.2.254 with 32 bytes of data:

Reply from 192.168.2.254: bytes=32 time=0ms TTL=128
Reply from 192.168.2.254: bytes=32 time=0ms TTL=128
Reply from 192.168.2.254: bytes=32 time=0ms TTL=128
Reply from 192.168.2.254: bytes=32 time=0ms TTL=128

Ping statistics for 192.168.2.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

b. Blocking a Network (e.g., 192.168.4.0):

- Use a wildcard mask (0.0.0.255) for the Class C network when blocking the whole network.

Configuration:

```
Router(config)#no access-list 1
Router(config)#access-list 1 deny 192.168.4.0 0.0.0.255
Router(config)#access-list 1 permit any
Router(config)#exit
Router#show access-lists
```

```
Router1
Physical Config CLI
IOS Command Line Interface
Router#
%SYS-5-CONFIG_I: Configured from console by console
show access-lists
Standard IP access list 1
 10 deny 192.168.4.0 0.0.0.255
 20 permit any
```

```
PC2
Physical Config Desktop Custom Interface
Command Prompt
Pinging 192.168.2.100 with 32 bytes of data:
Reply from 192.168.2.100: bytes=32 time=2ms TTL=126
Reply from 192.168.2.100: bytes=32 time=0ms TTL=126
Reply from 192.168.2.100: bytes=32 time=0ms TTL=126
Reply from 192.168.2.100: bytes=32 time=1ms TTL=126
Ping statistics for 192.168.2.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms
PC>ping 192.168.3.100
Pinging 192.168.3.100 with 32 bytes of data:
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Ping statistics for 192.168.3.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

```
PC3
Physical Config Desktop Custom Interface
Command Prompt
Pinging 192.168.2.100 with 32 bytes of data:
Reply from 192.168.2.100: bytes=32 time=0ms TTL=126
Reply from 192.168.2.100: bytes=32 time=0ms TTL=126
Reply from 192.168.2.100: bytes=32 time=1ms TTL=126
Reply from 192.168.2.100: bytes=32 time=0ms TTL=126
Ping statistics for 192.168.2.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>ping 192.168.3.100
Pinging 192.168.3.100 with 32 bytes of data:
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Ping statistics for 192.168.3.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

```
Laptop0
Physical Config Desktop Custom Interface
Command Prompt
Pinging 192.168.3.100 with 32 bytes of data:
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Ping statistics for 192.168.3.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 192.168.2.100
Pinging 192.168.2.100 with 32 bytes of data:
Reply from 192.168.2.100: bytes=32 time=0ms TTL=126
Reply from 192.168.2.100: bytes=32 time=0ms TTL=126
Reply from 192.168.2.100: bytes=32 time=0ms TTL=126
Reply from 192.168.2.100: bytes=32 time=8ms TTL=126
Ping statistics for 192.168.2.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 8ms, Average = 2ms
```

2. Extended ACL Implementation

a. Create an Access List (100-199):

- Denies or permits port (service).
- Denies or permits Source IP Address.
- Denies or permits Destination IP Address.

b. Apply the Access List to an Interface (Inbound):

Steps:

1. Remove the Standard ACL from Router 1:

```
Router(config)#no access-list 1
Router(config)#interface gigabitEthernet 0/1
Router(config-if)#no ip access-group 1 out
Router(config-if)#exit
Router(config)#exit
```

2. Configure Extended ACL in Router 2:

Router 2 Configuration:

```
Router(config)#access-list 100 deny ip 192.168.4.101 0.0.0.0 192.168.4.0 0.0.0.255
Router(config)#access-list 100 permit ip any any
Router(config)#exit
Router(config)#interface gigabitEthernet 0/1
Router(config-if)#ip access-group 100 in
Router(config-if)#exit
```

Example: Allow HTTP Traffic but Block ICMP (Ping):

```
Router(config)#no access-list 100
Router(config)#access-list 100 deny icmp 192.168.4.101 0.0.0.0 192.168.2.254 0.0.0.0
Router(config)#access-list 100 permit ip any any
Router(config)#
```