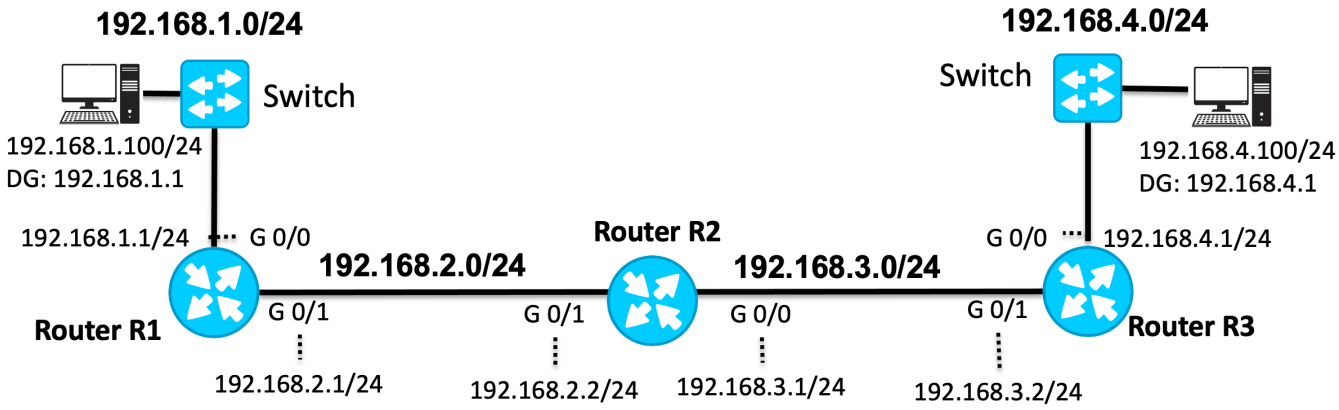
Static Routing Lab – PT Version

Network Topology



Packet Tracer Startup Topology

Part 1: Understanding the Objectives

Before you begin, take a moment to examine the network topology provided. In this lab, you will:

- **Configure Cisco Routers**: Set up the routers using Cisco IOS, including assigning IPv4 addresses to the interfaces. If you're new to Cisco IOS, it's recommended to start with one of the basic Cisco IOS labs first.
- **Cable the Network**: Physically connect the routers and switches as shown in the topology.
- **Set Up Static Routes**: Configure static routes on the routers to enable full end-to-end communication across the network.

This lab is designed to guide you through each of these tasks step by step. Throughout the process, you will encounter different scenarios, including what happens when tasks are incomplete or incorrectly configured. Don't worry—each step is clearly outlined to help you achieve successful network configuration.

Part 2: Basic IOS Commands

In this section, you will review how to configure Cisco routers using basic IOS (Internetwork Operating System) commands. These commands are fundamental to managing and configuring Cisco devices. You will start by accessing the router's command-line interface (CLI) and then proceed to set up essential configurations, such as disabling DNS lookups, setting passwords, and configuring hostnames. Mastering these basic IOS commands is crucial, as they form the foundation for other network configurations you'll encounter later in this lab, in other labs and in your career.

Note: The ONLY two passwords we will use in our labs are "cisco" or "class". Of course, this would not be the case in production networks.

Configure router R1

Method 1: Type in the commands using CLI

```
! Change from user mode to privileged exec mode
Router> enable

! Enter global configuration mode
Router# conf t

! Disable DNS lookup in case you mistype a command
Router(config)# no ip domain-lookup

! Configure the hostname
Router(config)# hostname R1

! Configure console port
R1(config)# line console 0
R1(config-line)# logging synchronous  ! Keeps output from interrupting input
R1(config-line)# exec-time 0 0   ! Keep device from timing out (lab only)
R1(config-line)# password cisco  ! Password required when connecting to console port
R1(config-line)# login           ! Login prompt
R1(config-line)# exit            ! Exit current mode (line mode)
R1(config)#
R1(config)# banner motd $        ! Banner message of the day, using $ as delimiter
Enter TEXT message.  End with the character '$'.
##############################
Authorized Access Only!
##############################
$

! Password required for entering privileged exec mode
R1(config)# enable secret class
R1(config)#
R1(config)# exit     ! Exit current mode (global config)
```

```
! Save running-config (RAM) to startup-config (NVRAM)
R1# copy running-config startup-config

Verification commands

R1# show running-config
R1# show startup-config
```

Method 2: Copy and Paste

- Add the following text to a file
- Select the text and copy it to the clipboard
- Paste the text to the Putty console window – Be sure you are at the proper prompt: Router(config)#

```
ena
conf t
no ip domain-lookup
hostname R1
line console 0
logging synchronous
exec-time 0 0
password cisco
login
exit

banner motd $
##############################
Authorized Access Only!
##############################
$
enable secret class
exit
```

Configure router R2

Configure similar commands on router R2 that you did on R1.

Method 1: Type in the commands using CLI

```
Router>
Router> enable
Router#
Router# conf t
Router(config)# no ip domain-lookup
Router(config)# hostname R2
R2(config)# line console 0
R2(config-line)# logging synchronous
```

```
R2(config-line)# exec-time 0 0
R2(config-line)# password cisco
R2(config-line)# login
R2(config-line)# exit
R2(config)#
R2(config)# banner motd $
Enter TEXT message.  End with the character '$'.
##############################
Authorized Access Only!
##############################
$
R2(config)#
R2(config)# enable secret class
R2(config)#
R2(config)# exit
R2# copy running-config startup-config

Verification commands

R2# show running-config
R2# show startup-config
```

Method 2: Copy and Paste

- Add the following text to a file
- Select the text and copy it to the clipboard
- Paste the text to the Putty console window – Be sure you are at the proper prompt: Router(config)#

```
ena
conf t
no ip domain-lookup
hostname R2
line console 0
logging synchronous
exec-time 0 0
password cisco
login
exit

banner motd $
##############################
Authorized Access Only!
##############################
$
enable secret class
exit
```

Configure router R3

Configure similar commands on R3 that you did on routers R1 and R2.

Method 1: Type in the commands using CLI

```
Router>
Router> enable
Router#
Router# conf t
Router(config)# no ip domain-lookup
Router(config)# hostname R2
R3(config)# line console 0
R3(config-line)# logging synchronous
R3(config-line)# exec-time 0 0
R3(config-line)# password cisco
R3(config-line)# login
R3(config-line)# exit
R3(config)#
R3(config)# banner motd $
Enter TEXT message.  End with the character '$'.
##############################
Authorized Access Only!
##############################
$
R3(config)#
R3(config)# enable secret class
R3(config)#
R3(config)# exit
R3# copy running-config startup-config

Verification commands

R3# show running-config
R3# show startup-config
```

Method 2: Copy and Paste

- Add the following text to a file
- Select the text and copy it to the clipboard
- Paste the text to the Putty console window – Be sure you are at the proper prompt: Router(config)#

```
ena
conf t
no ip domain-lookup
hostname R3
line console 0
logging synchronous
```

```
exec-time 0 0
password cisco
login
exit

banner motd $
############################
Authorized Access Only!
############################
$
enable secret class
exit
```

What if I make a mistake in IOS?

Don't worry if you make a mistake while configuring a Cisco device; IOS provides a simple way to correct it. Most commands can be undone by using the "**no**" keyword followed by the original command at the appropriate prompt.

For example, if you accidentally set an enable secret password and need to remove it:
```
R3(config)# enable secret class
R3(config)# no enable secret class
```

Similarly, if you assign an incorrect IP address to an interface, you can remove it like this:
```
R1(config-if)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# no ip address 192.168.1.1 255.255.255.0
```

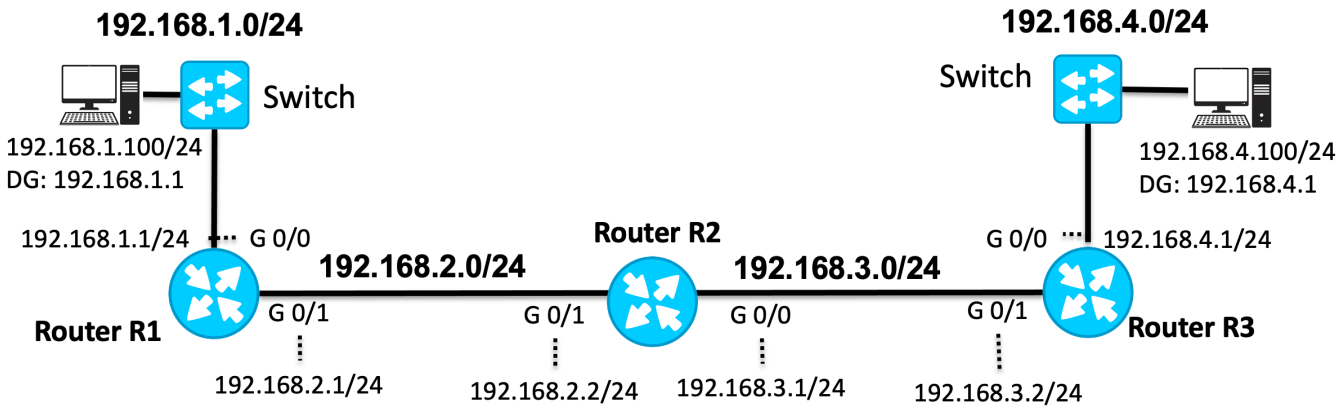
Some commands are specifically used to reverse default settings. For instance:

- To prevent the router from attempting DNS lookups when you mistype a command:
  `R1(config)# no ip domain-lookup`

- To bring an interface out of the default "shutdown" state and enable it
  `R1(config-if)# no shutdown`

By understanding how to reverse commands, you'll be able to troubleshoot and correct configurations efficiently.

Part 3: Configure Interfaces on R1, R2, and R3



Packet Tracer Startup Topology

Note: This lab was created using physical routers, switches and host computers. Some of the output may be different on Packet Tracer, such as showing serial interface. Also, there are a couple of "bugs", such as **show ip interface brief** displaying the status of an interface as **up** when there is no cable attached.

Step 1. Examine the Interfaces

To begin, examine the current status of the router's interfaces using the **show ip interface brief** command. This command provides a summary of the router's interfaces, including their IP addresses, status, and protocol state. For example, for router R1:

```
R1# show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
Embedded-Service-Engine0/0 unassigned      YES unset  administratively down down
GigabitEthernet0/0         unassigned      YES unset  administratively down down
GigabitEthernet0/1         unassigned      YES unset  administratively down down
R1#
```

By default, router interfaces are **administratively down**, meaning they are disabled and not ready to forward traffic. To activate an interface, you will need to issue the **no shutdown** command in interface configuration mode. This command enables the interface, allowing it to transmit and receive data.

Step 2. Configure R1 G0/0 LAN Interface

Next, you will configure R1's LAN interface, which connects to an Ethernet switch. This switch, along with any interconnected switches, forms the LAN where you would connect your end devices (such as computers). The IP address you configure on this interface will also serve as the default gateway for devices on this LAN.

Begin by entering the configuration mode and setting the IP address for the interface:

```
R1# conf t
R1(config)# interface gig 0/0
R1(config-if)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# end
```

After applying the configuration, you may notice that the interface's status and protocol are still down. This is expected, as the interface is not yet connected to an Ethernet switch or another device, and therefore is not receiving a carrier signal (no link light on the interface).

You can verify this by running the show ip interface brief command:

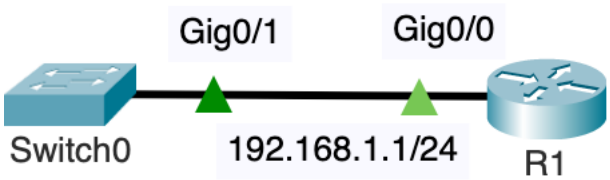
```
R1# show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
Embedded-Service-Engine0/0 unassigned      YES unset  administratively down down
GigabitEthernet0/0         192.168.1.1     YES manual down                  down
GigabitEthernet0/1         unassigned      YES unset  administratively down down
R1#
```

Here's what the output indicates:

- **Status**: This reflects the physical layer status of the interface. Since there is no cable connected, the physical layer cannot detect any connection, so the status is **down**.
- **Protocol**: This reflects the data link layer status. Because the physical layer is down (due to the lack of a carrier signal), the protocol is also **down**. Note that the protocol may also remain down if the interface is not properly configured with an IP address.

Step 3. Connect R1's LAN G 0/0 Interface to Switchport

Link lights during STP convergence.



Link lights after STP after STP convergence.

In this step, you will physically connect R1's LAN interface (G0/0) to an Ethernet switch, enabling network connectivity.

1. **Cable R1 to the LAN Switch**: Use an Ethernet straight-through cable to connect the GigabitEthernet0/0 (G0/0) interface on R1 to any available port on the Ethernet switch. You will notice an amber link light on the switch. After about 50 seconds the link light on the switch

turns from amber to green. This is due to the STP (Spanning Tree Protocol) on the switch ensuring that there is no loop in the Ethernet LAN before allowing the switch to receive and forward frames.

2. **Auto-Negotiation of Speed and Duplex**: The router and switch will automatically negotiate the best possible duplex and speed settings. Typically:
    - If both the router's port and the switch's port are GigabitEthernet, the connection will operate at 1 Gb/s in full-duplex mode.
    - If the router's port is GigabitEthernet and the switch's port is FastEthernet, the connection will operate at 100 Mb/s in full-duplex mode.
3. **Verify the Connection**: Once the cable is connected, you should see the interface status change to "up" on both the physical layer (Status) and the data link layer (Protocol). This change will be reflected in the router's console output, indicating a successful connection:

```
*Apr 21 13:38:55.387: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Apr 21 13:38:56.387: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0,
changed state to up
R1#
```

This step establishes the physical connection necessary for R1 to communicate with other devices on the LAN.

Step 4. Verify the interface is now up

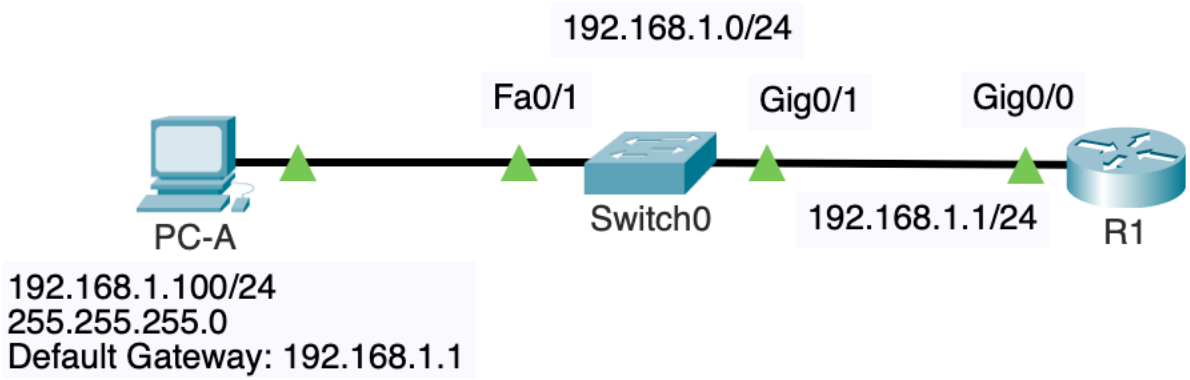
After connecting R1's G0/0 interface to the switch, it's crucial to verify that the interface is operational. Use the **show ip interface brief** command to check the status of all interfaces on the router:

```
R1# show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
Embedded-Service-Engine0/0 unassigned      YES unset  administratively down down
GigabitEthernet0/0         192.168.1.1     YES manual up                    up
GigabitEthernet0/1         unassigned      YES unset  administratively down down
R1#
```

In this output, you should observe that the **Status** and **Protocol** for the GigabitEthernet0/0 interface are both listed as "up." This indicates that the interface is not only physically connected (Status: up) but also operational at the data link layer (Protocol: up).

If both fields are "**up**," the interface is fully active and ready to transmit and receive data, confirming a successful connection between R1 and the switch.

Step 5: Connect and Configure Client Computer PC-A



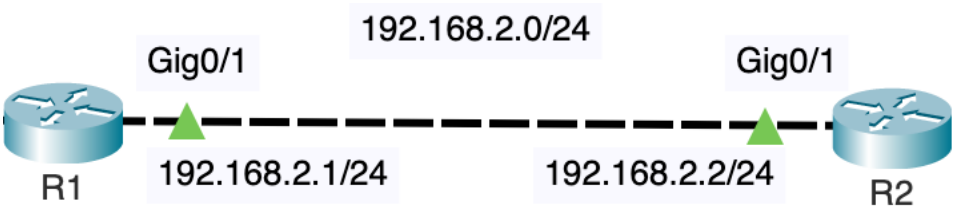
Link lights after STP after STP convergence.

In this step, you will connect PC-A to the network and configure its IP settings to enable communication with other devices on the LAN.

1. **Cable PC-A to Switch**: Use an Ethernet straight-through cable to connect PC-A to Switch0's Fa0/1 port. This physical connection will link PC-A to the same LAN as Router R1. Once again, you will notice an amber link light on the switch. After about 50 seconds the link light on the switch turns from amber to green, meaning STP has determined there are no loops.
2. **Configure IP Settings on PC-A**:
    - **IPv4 Address**: Set the IP address to 192.168.1.100. This address should be within the same subnet as R1's G0/0 interface.
    - **Subnet Mask**: Use the subnet mask 255.255.255.0, which matches the subnet configuration for the 192.168.1.0/24 network.
    - **Default Gateway**: Set the default gateway to 192.168.1.1, which is the IP address of R1's G0/0 interface. This gateway allows PC-A to communicate with devices outside its local subnet.

With these configurations, PC-A will be able to communicate with R1 and other devices on the 192.168.1.0/24 network, as well as route traffic to remote networks through R1.

Step 6. Configure and Enable R1's G0/1 Inter-router Interface with R2



Link lights after both R1 and R2 have been configured in step 6 and step 7.

In this step, you'll interconnect routers R1 and R2 to enable communication between them via their Gig0/1 interfaces.

1. **Cable R1 to R2**:
    - **Option 1**: Directly connect the GigabitEthernet0/1 (G0/1) interfaces of R1 and R2 using an Ethernet cross-over cable or straight-through cable. The routers support auto-MDIX, which means they can automatically adjust to the correct cable type.
    - **Option 2**: Alternatively, you can connect both routers to the same Ethernet switch using Ethernet straight-through cables, with each router connecting to a different port on the switch.
2. **Configure R1's G0/1 Interface**:
    - Enter interface configuration mode for G0/1 on R1.
    - Assign the IP address 192.168.2.1 with a subnet mask of 255.255.255.0. This IP address will be in the same subnet as the corresponding interface on R2, enabling direct communication between the two routers.
    - Enable the interface using the no shutdown command to bring it up.

The configuration commands are as follows:


```
R1(config)# inter g 0/1
R1(config-if)# ip add 192.168.2.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# exit
```

Purpose: By configuring and enabling this interface, R1 and R2 will be able to forward packets between each other over the 192.168.2.0/24 network, which is essential for routing traffic between different networks connected to these routers.

Step 7: Configure R2's G 0/1 inter-router interface with R1

In this step, you will configure the Gig0/1 interface on R2 to enable communication with R1 over the 192.168.2.0/24 network.

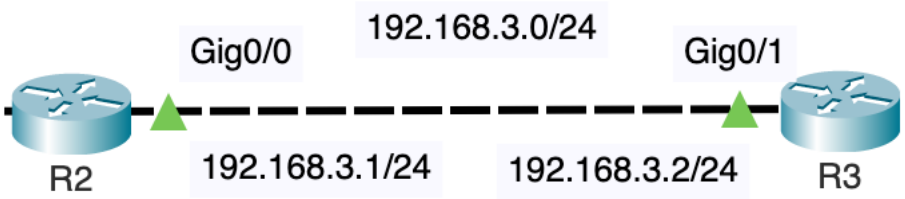
1. **Configure R2's G0/1 Interface**:
    - Enter interface configuration mode for G0/1 on R2.
    - Assign the IP address 192.168.2.2 with a subnet mask of 255.255.255.0. This address is in the same subnet as R1's G0/1 interface, ensuring that the two routers can communicate directly.
    - Enable the interface by issuing the no shutdown command to bring it up.

The configuration commands are as follows:

```
R2(config)# inter g 0/1
R2(config-if)# ip add 192.168.2.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# exit
```

Purpose: This configuration establishes the inter-router connection between R1 and R2 on the 192.168.2.0/24 network. With both routers now able to forward packets to each other over this shared network, they can route traffic between their respective connected networks.

Step 8: Configure R2's G 0/0 inter-router interface with R3 G 0/1



Link lights after both R1 and R2 have been configured in step 8 and step 9.

In this step, you will configure the connection between routers R2 and R3 to enable communication over the 192.168.3.0/24 network.

1. **Cable R2 to R3**:
    - Use an Ethernet straight-through or cross-over cable to connect the GigabitEthernet0/0 (G0/0) interface on R2 to the GigabitEthernet0/1 (G0/1) interface on R3. This physical connection will establish a network link between the two routers.
2. **Configure R2's G0/0 Interface**:
    - Enter interface configuration mode for G0/0 on R2.
    - Assign the IP address 192.168.3.1 with a subnet mask of 255.255.255.0. This address will be in the same subnet as R3's G0/1 interface, allowing direct communication between the two routers.
    - Enable the interface by issuing the no shutdown command to bring it up.

The configuration commands are as follows:


```
R2(config)# inter g 0/0
R2(config-if)# ip add 192.168.3.1 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# exit
```

Purpose: Configuring R2's G0/0 interface to share the same network as R3's G0/1 interface allows R2 and R3 to forward packets to each other. This setup is critical for routing traffic between the networks connected to each router.

Step 9: Configure R3's G 0/1 inter-router interface with R2

In this step, you'll configure the Gig0/1 interface on R3 to enable communication with R2 over the 192.168.3.0/24 network.

1. **Configure R3's G0/1 Interface**:
    - Enter interface configuration mode for G0/1 on R3.
    - Assign the IP address 192.168.3.2 with a subnet mask of 255.255.255.0. This IP address should be in the same subnet as R2's G0/0 interface, enabling direct communication between R3 and R2.
    - Enable the interface by issuing the no shutdown command to bring it up.

The configuration commands are as follows:


```
R3(config)# inter g 0/1
R3(config-if)# ip add 192.168.3.2 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# exit
```

Purpose: This configuration sets up the inter-router connection between R3 and R2 on the 192.168.3.0/24 network. With both routers now able to forward packets to each other, they can route traffic between the networks connected to them.

Step 10. Configure R3's G0/0 LAN Interface

In this step, you will configure R3's LAN interface, which will be used to connect to an Ethernet switch. This switch, along with any interconnected switches, forms the LAN where end devices, such as computers, will connect. The IP address you configure on this interface will also serve as the default gateway for devices on this LAN.

1. **Configure R3's G0/0 Interface**:
    - Enter interface configuration mode for G0/0 on R3.
    - Assign the IP address 192.168.4.1 with a subnet mask of 255.255.255.0. This IP address will be used as the default gateway for devices on the 192.168.4.0/24 network.
    - Enable the interface using the no shutdown command to bring it up.

The configuration commands are as follows:

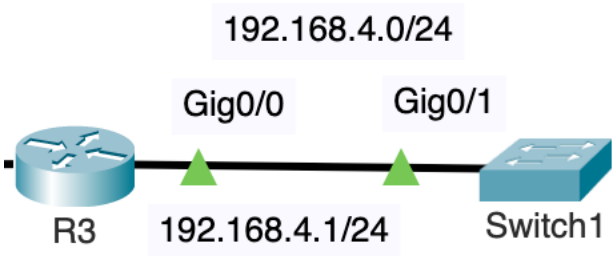

```
R3# conf t
R3(config)# interface gig 0/0
R3(config-if)# ip address 192.168.4.1 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# end
```

Note on Interface Status:

- After configuring the interface, you might notice that the interface status is still down. This is because the interface is not yet connected to an Ethernet switch or any other device, which means it is not receiving a carrier signal (indicated by the absence of a link light). Once the interface is connected to a switch or another device, the status should change to "up," indicating that the interface is active and ready to forward traffic.

This step ensures that R3 is properly configured to act as the default gateway for devices on its connected LAN, allowing for communication within the local network and with other networks through R3.

Step 11. Connect R3's LAN G 0/0 Interface to Switchport



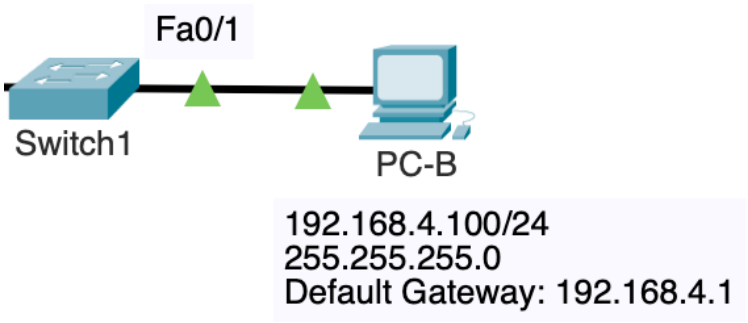
Link lights after step 10 and step 11.

In this step, you'll establish the physical connection between R3 and the LAN switch, enabling communication between R3 and other devices on the network.

1. **Cable R3 to the LAN Switch**:
    - Use an Ethernet straight-through cable to connect R3's GigabitEthernet0/0 (G0/0) interface to any available port on the Ethernet switch. This physical connection will link R3 to the LAN, allowing it to communicate with any end devices (such as computers) connected to the switch. Again, due to STP the link light on the switch will be amber and then turn green in about 50 seconds.
2. **Ensure the Switch is Powered On**:
    - Before connecting the cable, make sure that the switch is powered on and operational. This ensures that the interface on R3 will receive a carrier signal, which will bring the interface "up" and enable data transmission.
3. **Purpose**:

- This step completes the setup of R3's LAN connection, allowing it to act as the default gateway for devices on the network and facilitating communication between these devices and other networks connected through R3.

Step 12: Configure and Connect Client Computer Option

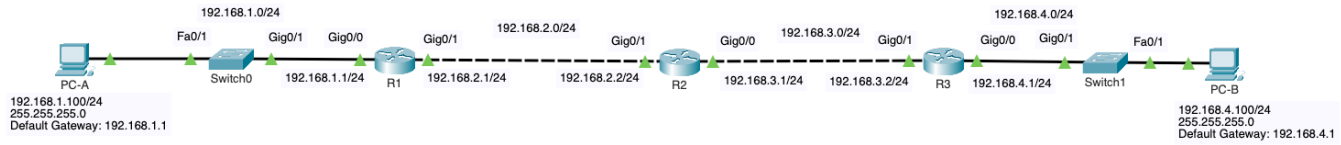


Link lights after STP after STP convergence.

In this step, you'll connect and configure PC-B to the network, enabling it to communicate with other devices on the LAN.

1. **Cable PC-B to Switch1**:
   - Use an Ethernet straight-through cable to connect PC-B to the Fa0/1 port on Switch1. This connection will integrate PC-B into the LAN associated with Router R3.
2. **Configure IP Settings on PC-B**:
   - **IPv4 Address**: Set the IP address to 192.168.4.100. This address should be within the same subnet as R3's G0/0 interface.
   - **Subnet Mask**: Use the subnet mask 255.255.255.0, matching the subnet configuration for the 192.168.4.0/24 network.
   - **Default Gateway**: Set the default gateway to 192.168.4.1, which is the IP address of R3's G0/0 interface. This gateway allows PC-B to route traffic through R3 to other networks.
3. **Purpose**:
   - Configuring PC-B in this manner ensures that it can communicate within the 192.168.4.0/24 network and with other networks via R3. This setup is essential for enabling PC-B to participate in the broader network topology.

Completed topology



Your topology is now completed and should look like the one above.

Step 13. Verify on all three routers that their GigabitEthernet interfaces are now up

In this step, you will verify that the GigabitEthernet interfaces on all three routers are operational, confirming that the physical connections and configurations are correct.

1. **Check R1's Interface Status**:
    - Use the **show ip interface brief** command to display the status of all interfaces on R1. Verify that both the G0/0 and G0/1 interfaces are "up" for both the Status and Protocol columns, indicating that the interfaces are active and ready to forward traffic.


```
R1# show ip inter brief
Interface                  IP-Address      OK? Method Status                Protocol
Embedded-Service-Engine0/0 unassigned      YES unset  administratively down down
GigabitEthernet0/0         192.168.1.1     YES manual up                    up
GigabitEthernet0/1         192.168.2.1     YES manual up                    up
R2#
```

2. **Check R2's Interface Status**:
    - Similarly, run the **show ip interface brief** command on R2 and confirm that the G0/0 and G0/1 interfaces are "up" in both the Status and Protocol columns.


```
R2# show ip inter brief
Interface                  IP-Address      OK? Method Status                Protocol
Embedded-Service-Engine0/0 unassigned      YES unset  administratively down down
GigabitEthernet0/0         192.168.3.1     YES manual up                    up
GigabitEthernet0/1         192.168.2.2     YES manual up                    up
R2#
```

3. **Check R3's Interface Status**:
    - Finally, use the **show ip interface brief** command on R3 to ensure that both G0/0 and G0/1 interfaces are "up" for Status and Protocol, confirming that R3 is fully connected and operational.


```
R3# show ip inter brief
Interface                  IP-Address      OK? Method Status                Protocol
Embedded-Service-Engine0/0 unassigned      YES unset  administratively down down
GigabitEthernet0/0         192.168.4.1     YES manual up                    up
GigabitEthernet0/1         192.168.3.2     YES manual up                    up
R2#
```

Part 4: Testing Reachability

In this section, you'll verify the network connectivity by examining the routing tables and testing the reachability between the routers.

Step 1: Examine the routing tables of R1, R2 and R3.

Start by reviewing the routing tables on each router to understand what networks they are aware of. Currently, each router only knows about its directly connected networks, as reflected in their routing tables. They do not have information about the remote networks connected to the other routers.

1. **Check R1's Routing Table**:
    - Use the **show ip route** command to display the routing table on R1. Notice that R1 only has entries for the networks directly connected to its interfaces (192.168.1.0/24 and 192.168.2.0/24).


```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
<output omitted>

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, GigabitEthernet0/1
L        192.168.2.1/32 is directly connected, GigabitEthernet0/1
R1#
```

2. **Check R2's Routing Table**:
    - Similarly, run the **show ip route** command on R2. You'll see that R2 only has routes for its directly connected networks (192.168.2.0/24 and 192.168.3.0/24).


```
R2# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
<output omitted>

Gateway of last resort is not set
```

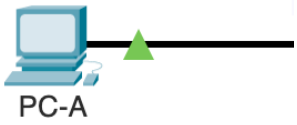
```
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, GigabitEthernet0/1
L        192.168.2.2/32 is directly connected, GigabitEthernet0/1
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/24 is directly connected, GigabitEthernet0/0
L        192.168.3.1/32 is directly connected, GigabitEthernet0/0
R2#
```

3. **Check R3's Routing Table**:
   - Finally, inspect the routing table on R3 with the **show ip route** command. R3 should only have routes for its directly connected networks (192.168.3.0/24 and 192.168.4.0/24).


```
R3# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
<output omitted>

Gateway of last resort is not set

      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/24 is directly connected, GigabitEthernet0/1
L        192.168.3.2/32 is directly connected, GigabitEthernet0/1
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/24 is directly connected, GigabitEthernet0/0
L        192.168.4.1/32 is directly connected, GigabitEthernet0/0
R2#
```

Step 2. Test Reachability from End Devices

From PC-A



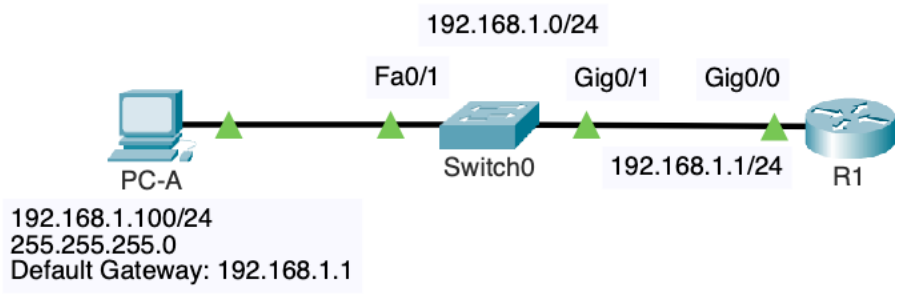
192.168.1.100/24
255.255.255.0
Default Gateway: 192.168.1.1

When a PC, such as PC-A, needs to send a packet to a device on a different network, it uses its default gateway to forward the packet. The default gateway is the IP address of the router interface connected to the same subnet as the PC (in this case, 192.168.1.1 on Router R1). When PC-A tries to send a packet to a remote network, such as 192.168.4.0/24, it recognizes that the destination IP is not within its local 192.168.1.0/24 subnet. The PC then forwards the packet to its default gateway, Router R1, which is responsible for routing the packet to the correct remote network, using static routes or other routing

protocols configured on the router. This process allows communication between different networks through the router.

From PC-A ping the following IP addresses. Were these pings successful? Why or why not?

Can PC-A ping its default gateway, router R1?

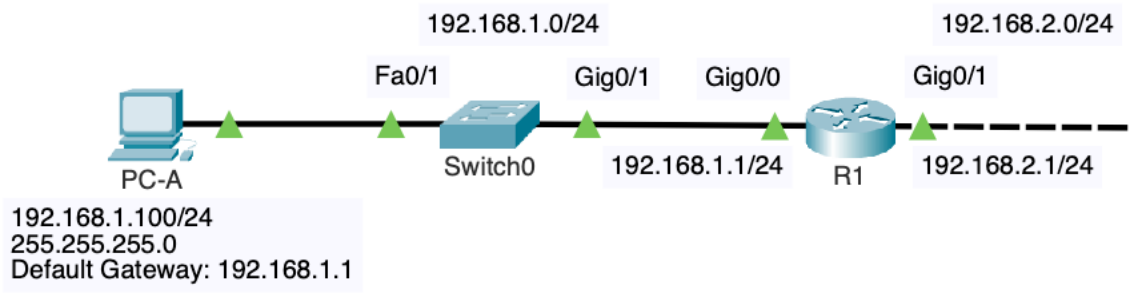



```
C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.1.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

Success? Yes. PC-A can ping 192.168.1.1, its default gateway, because both PC-A and the gateway interface are on the same local subnet (192.168.1.0/24).

PCs and routers can ping other devices within the same network because they share the same IP subnet, which allows them to communicate directly without needing a router. For example, PC-A with IP address 192.168.1.100 can communicate with Router R1's G0/0 interface at 192.168.1.1 because both devices are in the same 192.168.1.0/24 subnet. This direct communication is facilitated by the switch, which operates at Layer 2 of the OSI model, forwarding packets based on MAC addresses within the same network. Similarly, devices on the 192.168.2.0/24 and 192.168.3.0/24 networks can communicate with each other directly because they are also within the same subnet, allowing local traffic to pass through without routing.

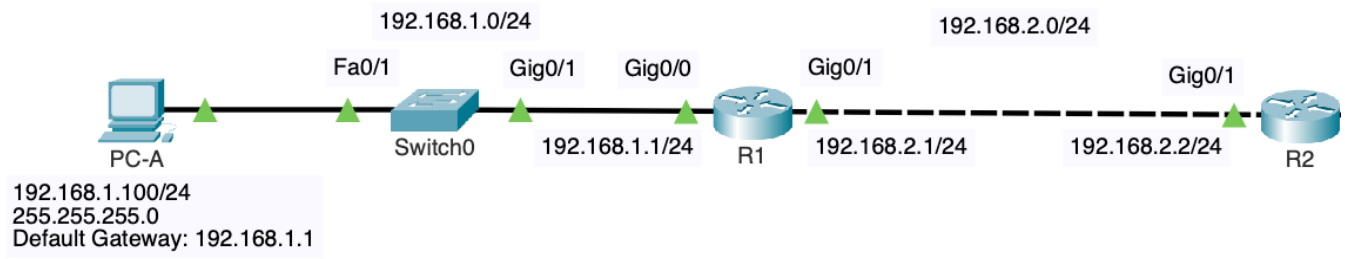
Can PC-A ping the other interface on router R1?




```
C:\>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time<1ms TTL=255
Reply from 192.168.2.1: bytes=32 time<1ms TTL=255
<output omitted>
```

Success? Yes. PC-A can ping 192.168.2.1, the Gig0/1 interface on Router R1, because the router is directly connected and has routes for both the 192.168.1.0/24 and 192.168.2.0/24 networks, allowing it to forward packets between these two subnets. Additionally, no static routes are required within Router R1 itself to reach directly connected networks.

Can PC-A ping the closest interface on router R2?




```
C:\> ping 192.168.2.2

Pinging 192.168.4.100 with 32 bytes of data:

Request timed out.
Request timed out.
<output omitted>
```

Success? No: PC-A cannot ping 192.168.2.2, the Gig0/1 interface on Router R2, because while Router R1 can forward the packet to Router R2 (since both routers share the 192.168.2.0/24 network), Router R2 does not have a route back to the 192.168.1.0/24 network where PC-A resides. When PC-A sends a ping to 192.168.2.2, Router R1 successfully forwards the packet to R2 because they are on the same 192.168.2.0/24 subnet. However, when R2 receives the packet, it does not have a route to return the reply to PC-A's 192.168.1.0/24 network, causing the ping to fail. To enable successful communication, a static route would need to be configured on R2, instructing it on how to reach the 192.168.1.0/24 network via Router R1.

Step 3: Verify that R1 and R2 can ping each other on same interconnected network

In this step, you'll verify that R1 and R2 can successfully ping each other over their directly connected link, confirming that the physical connection and IP configuration are correct.

1. **Ping from R1 to R2**:
    - On R1, use the ping command to test connectivity to R2's IP address on the interconnected network (192.168.2.2).


```
R1# ping 192.168.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
R1#
```

Note: The first ping may fail due to the need for an ARP (Address Resolution Protocol) Request and Reply to resolve the MAC address associated with the IP address. Once ARP resolution is complete, subsequent pings should succeed.

2. **View the ARP Table on R1**:
    - After the ping, you can view the ARP table on R1 to see the resolved MAC addresses for the connected devices.

```
R1# show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.1.1              -   64f6.9d1c.8970  ARPA   GigabitEthernet0/0
Internet  192.168.2.1              -   64f6.9d1c.8971  ARPA   GigabitEthernet0/1
Internet  192.168.2.2              1   64f6.9d1c.8c41  ARPA   GigabitEthernet0/1
R1#
```

This table shows the MAC addresses that R1 has learned through ARP, allowing it to forward packets to the correct device on the network.

3. **Ping from R2 to R1**:
    - Similarly, on R2, use the ping command to test connectivity to R1's IP address on the interconnected network (192.168.2.1).


```
R2# ping 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
```

```
!!!!!
Success rate is 100 percent (4/5), round-trip min/avg/max = 1/1/1 ms
R1#
```

PCs and routers can ping other devices within the same network because they share the same IP subnet. This allows them to communicate directly without the need for routing. The successful pings confirm that R1 and R2 are correctly configured and can communicate over the 192.168.2.0/24 network.

Step 4: Verify that R2 and R3 can ping each other on same interconnected network

In this step, you will verify that R2 and R3 can successfully ping each other over their directly connected link, ensuring that their physical connection and IP configuration are correct.

1. **Ping from R2 to R3**:
    - On R2, use the ping command to test connectivity to R3's IP address on the interconnected network (192.168.3.2).


```
R2# ping 192.168.3.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
R1#
```

Note: Similar to the previous step, the first ping may fail due to the ARP process, which resolves the MAC address associated with the IP address. Once ARP resolution is complete, subsequent pings should succeed.

2. **Ping from R3 to R2**:
    - On R3, use the ping command to test connectivity to R2's IP address on the interconnected network (192.168.3.1).


```
R3# ping 192.168.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (4/5), round-trip min/avg/max = 1/1/1 ms
R1#
```

The successful pings between R2 and R3 confirm that both routers are correctly configured and can communicate directly over the 192.168.3.0/24 network. This step ensures that the network connection between these two routers is fully operational.

Step 5: Verify routers cannot reach Remote networks

From R1

```
R1# ping 192.168.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R1#
```

Previously, Router R1 could ping 192.168.2.2 on Router R2 because both interfaces are part of the same directly connected 192.168.2.0/24 network, allowing R1 to forward the ping packet directly to R2. However, when R1 attempts to ping 192.168.3.1, which is the Gig0/0 interface on R2, the ping fails because R1 does not have a route to the 192.168.3.0/24 network. Since the 192.168.3.0/24 network is not directly connected to R1 and no static routes or routing protocols have been configured to inform R1 how to reach this network, R1 is unable to forward the ping to the correct destination. To successfully reach 192.168.3.1, R1 would need to have a route that tells it to send traffic destined for the 192.168.3.0/24 network to R2, which can then forward the packet to the appropriate interface. This is the case for other remote networks as well.

```
R1# ping 192.168.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R1#
```

From R2: Unable to reach remote networks


```
R2# ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R2#
```

```
R2# ping 192.168.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R2#
```

From R3: Unable to reach remote networks


```
R3# ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R2#

R3# ping 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R2#
```

Part 5: Configuring Static Routes

In this section, you'll configure static routes on each router to ensure that packets can be successfully routed to remote networks. Static routes are essential for directing traffic to its final destination when the destination network is not directly connected to the router.

Step 1. Configure Static Routes on R1 and R2 to reach remote networks

Each router will be configured with static routes using the next-hop IP address of its neighboring router to reach any remote networks. To ensure successful packet delivery, each router must have a static route that directs the packet towards its final destination.

Keep in mind that routers share a directly connected network with their neighboring routers, allowing them to communicate directly. They utilize this shared network to forward packets to each other for remote network destinations.

Remember Alex Zinin's Routing Table Principles:

- **Principle 1:** Every router makes its decision alone, based on the information it has in its own routing table.
- **Principle 2:** The fact that one router has certain information in its routing table does not mean that other routers have the same information.
- **Principle 3:** Routing information about a path from one network to another does not provide routing information about the reverse, or return, path.

Configuring Static Routes on R1

1. **To Reach the 192.168.3.0/24 Network**:
   - R1 must send packets destined for the 192.168.3.0/24 network to R2's next-hop IP address (192.168.2.2).


```
R1# conf t
R1(config)# ip route 192.168.3.0 255.255.255.0 192.168.2.2
R1(config)#
```

2. **To Reach the 192.168.4.0/24 Network**:
   - Similarly, R1 must send packets destined for the 192.168.4.0/24 network to R2's next-hop IP address (192.168.2.2).

To reach the 192.168.4.0/24 network, R1 must send the packets to neighbor R2's next-hop address 192.168.2.2

```
R1# conf t
R1(config)# ip route 192.168.4.0 255.255.255.0 192.168.2.2
R1(config)#
```

Configuring Static Routes on R2

1. **To Reach the 192.168.1.0/24 Network**:
   - R2 must send packets destined for the 192.168.1.0/24 network to R1's next-hop IP address (192.168.2.1).


```
R2# conf t
R2(config)# ip route 192.168.1.0 255.255.255.0 192.168.2.1
R2(config)#
```

2. **To Reach the 192.168.4.0/24 Network**:
   - R2 must send packets destined for the 192.168.4.0/24 network to R3's next-hop IP address (192.168.3.2).

```
R2# conf t
R2(config)# ip route 192.168.4.0 255.255.255.0 192.168.3.2
R2(config)#
```

Troubleshooting Connectivity from PC-A to PC-B

Attempt to ping PC-B from PC-A.

192.168.1.100/24
255.255.255.0
Default Gateway: 192.168.1.1

```
C:\> ping 192.168.4.100

Pinging 192.168.4.100 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Reply from 192.168.1.1: Destination host unreachable.

Ping statistics for 192.168.4.100:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

The ping from PC-A to PC-B fails. Why? Consider the following:

1. **Is the packet reaching its destination?**
    - To determine if the packet is reaching PC-B, you can perform a **traceroute** from PC-A to PC-B. This will show each hop the packet takes towards its destination. If the traceroute reaches R3 but doesn't complete to PC-B, it indicates that the packet is reaching the network connected to R3, but there might be an issue with the final delivery to PC-B.

2. **Is there an issue with the return path of the packets?**
    - Even if the packet reaches PC-B, the ping can still fail if there's no route back to PC-A. Since R3 does not have any static routes configured to provide a return path to the 192.168.1.0/24 network (where PC-A resides), PC-B's response cannot reach PC-A. Without a static route on R3 that directs packets back to PC-A's network, the return packet will be dropped because R3 doesn't know how to reach the 192.168.1.0/24 network.

Answer: The ping is failing due to a lack of a return route on R3. The packet might reach PC-B, but without a configured static route on R3 to send the reply back to PC-A's network, the response packet cannot be delivered.

Step 2. Configure Static Routes on R3 to reach remote networks

In this step, you will configure static routes on R3 to ensure it can properly route traffic to remote networks, specifically the networks connected to R1.

R3's Static Routes Configuration

1. **To Reach the 192.168.2.0/24 Network**:
    - R3 needs to send packets destined for the 192.168.2.0/24 network to R2's next-hop IP address (192.168.3.1). This will allow R3 to forward traffic towards R2, which then routes the packets to the 192.168.2.0/24 network.


```
R3# conf t
R3(config)# ip route 192.168.2.0 255.255.255.0 192.168.3.1
R3(config)#
```

2. **To Reach the 192.168.1.0/24 Network**:
    - Similarly, R3 must send packets destined for the 192.168.1.0/24 network to R2's next-hop IP address (192.168.3.1). This allows R3 to route traffic towards R1's network, using R2 as an intermediary.

```
R3# conf t
R3(config)# ip route 192.168.1.0 255.255.255.0 192.168.3.1
R3(config)#
```

By configuring these static routes, R3 will know how to forward packets to networks that are not directly connected to it. This is essential for enabling full communication between devices on different networks across the routers. Once these routes are in place, R3 will be able to send packets to both the 192.168.2.0/24 and 192.168.1.0/24 networks via R2.

Step 3: Verify Static Routes in IP Routing Tables

In this step, you will verify that the static routes you configured have been successfully added to the routing tables of each router. This ensures that the routers not only know about their directly connected networks but can also route traffic to remote networks via the static routes.

1. **Check R1's Routing Table**:
   - Use the **show ip route** command to view R1's routing table. You should see entries for the directly connected networks (192.168.1.0/24 and 192.168.2.0/24) as well as the static routes you configured to reach the 192.168.3.0/24 and 192.168.4.0/24 networks via R2.

R1

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
<output omitted>

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, GigabitEthernet0/1
L        192.168.2.1/32 is directly connected, GigabitEthernet0/1
S     192.168.3.0/24 [1/0] via 192.168.2.2
S     192.168.4.0/24 [1/0] via 192.168.2.2
R1#
```

2. **Check R2's Routing Table**:
   - On R2, run the **show ip route** command to verify that the static routes you configured for the 192.168.1.0/24 and 192.168.4.0/24 networks have been added, alongside the directly connected networks (192.168.2.0/24 and 192.168.3.0/24).

R2

```
R2# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
<output omitted>

Gateway of last resort is not set

S     192.168.1.0/24 [1/0] via 192.168.2.1
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, GigabitEthernet0/1
L        192.168.2.2/32 is directly connected, GigabitEthernet0/1
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/24 is directly connected, GigabitEthernet0/0
L        192.168.3.1/32 is directly connected, GigabitEthernet0/0
S     192.168.4.0/24 [1/0] via 192.168.3.2

R2#
```

3. **Check R3's Routing Table**:
   - Finally, on R3, use the **show ip route** command to ensure that the static routes to the 192.168.1.0/24 and 192.168.2.0/24 networks are in place, along with the directly connected networks (192.168.3.0/24 and 192.168.4.0/24).

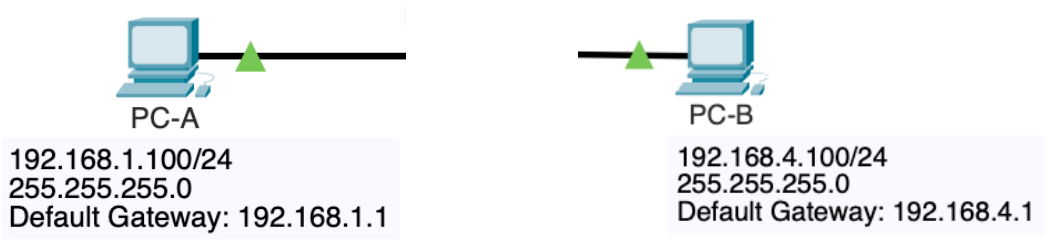

```
R3# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
<output omitted>

Gateway of last resort is not set

S     192.168.1.0/24 [1/0] via 192.168.3.1
S     192.168.2.0/24 [1/0] via 192.168.3.1
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/24 is directly connected, GigabitEthernet0/1
L        192.168.3.2/32 is directly connected, GigabitEthernet0/1
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/24 is directly connected, GigabitEthernet0/0
L        192.168.4.1/32 is directly connected, GigabitEthernet0/0
R2#
```

The routing tables on each router should now include not only the directly connected networks but also the static routes you configured for reaching remote networks. This means that all routers can route traffic to networks that are not directly connected to them, enabling full communication across the network.

Step 4. Verify reachability from PC-A

In this step, you'll verify that the network configuration is working correctly by testing connectivity between PC-A and PC-B.

1. **Ping from PC-A to PC-B**:
    - On PC-A, use the ping command to test connectivity to PC-B's IP address (192.168.4.100).




```
C:\>ping 192.168.4.100

Pinging 192.168.4.100 with 32 bytes of data:

Reply from 192.168.4.100: bytes=32 time<1ms TTL=125
Reply from 192.168.4.100: bytes=32 time<1ms TTL=125
Reply from 192.168.4.100: bytes=32 time<1ms TTL=125
Reply from 192.168.4.100: bytes=32 time<1ms TTL=125

Ping statistics for 192.168.4.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

Expected Outcome:

- If the network is correctly configured, you should receive replies from PC-B, indicating that PC-A can successfully communicate with PC-B. The ping should show 0% packet loss, and the round-trip times should be minimal, indicating a direct and efficient path between the two devices.

Explanation:

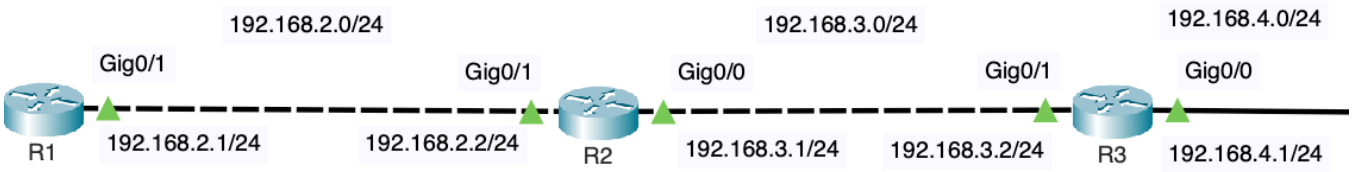
- A successful ping from PC-A to PC-B confirms that the static routes on the routers are properly configured, allowing traffic to flow between the networks connected to R1, R2, and R3. It demonstrates that packets can travel from PC-A's network (192.168.1.0/24) to PC-B's network (192.168.4.0/24) and back, verifying end-to-end connectivity across the entire network.

Step 5. Verify reachability to remote networks from router R1

Now that PC-A can successfully ping PC-B, we have confirmed end-to-end reachability across the network. However, it is also important to verify reachability directly from the routers to ensure that they can communicate with remote networks.

1. **Ping Remote Networks from R1**:
    - On R1, use the ping command to test connectivity to the IP addresses of R3's interfaces, 192.168.3.1 and 192.168.4.1.


```
R1# ping 192.168.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
R1#
```




```
R1# ping 192.168.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
R1#
```

Expected Outcome: The pings to both 192.168.3.1 and 192.168.4.1 should succeed with a 100% success rate, indicating that R1 can reach R3's interfaces. This confirms that the static routes are functioning correctly and that R1 can communicate with networks that are not directly connected to it.

Conclusion: Successfully pinging remote networks from R1 demonstrates that all routers and networks within the topology are fully reachable from one another. This ensures that the routing configurations are correct and that the network is operating as intended.

At this point, every router (and connected device) should be able to reach (ping) every other router and device across the entire network.

Step 6. Sending a ping from a different router interface

When a router sends an ICMP Echo Request (ping), it typically uses the IP address of the interface through which the packet is forwarded as the source IP address. However, you can specify a different source IP address from any of the router's interfaces to test whether the remote device can send an ICMP Echo Reply back to that specific IP address.

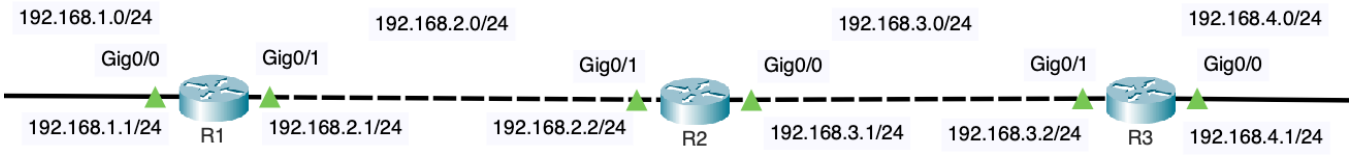
This is especially useful when you want to test connectivity from a particular interface or when you don't have end devices (like computers) on the LANs.

Sending a Ping with a Specified Source Address

1. **Specify the Source Interface**:
    - You can use the ping command with the source option to specify which interface's IP address should be used as the source IP in the ICMP Echo Request.

Example:

- To send a ping from R1's GigabitEthernet0/0 interface (IP address 192.168.1.1) to the IP address 192.168.4.1:




```
R1# ping 192.168.4.1 ?
  data      specify data pattern
  df-bit    enable do not fragment bit in IP header
  repeat    specify repeat count
  size      specify datagram size
  source    specify source address or name
  timeout   specify timeout interval
  tos       specify type of service value
  validate  validate reply data
  <cr>

R1# ping 192.168.4.1 source gig 0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
R1#
```

Explanation: In this example, the ping packet is sent from R1, using the IP address of the G0/0 interface (192.168.1.1) as the source. This allows you to verify that the remote device (in this case, 192.168.4.1) can successfully receive the ping and send a reply back to the specified source IP address.

Use Cases: This technique is useful for testing the reachability of specific interfaces, especially in scenarios where you want to ensure that traffic can be routed back to a particular interface on the router. It also helps confirm that routing is correctly configured across different parts of the network.

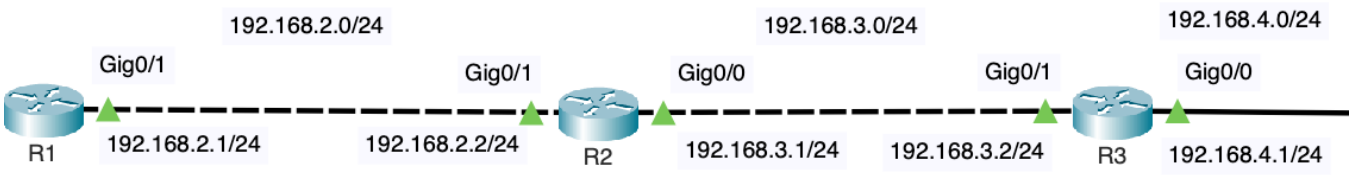
By using the source option, you can gain more control over your network tests, ensuring that every interface on the router can successfully send and receive traffic across the network.

Step 7. Use Traceroute to see the path the packets take

The traceroute command is a valuable tool for visualizing the path that packets take through a network. By using traceroute, you can see each hop along the route from the source to the destination, which helps in understanding the routing behavior and identifying any potential issues in the network.

Traceroute Examples

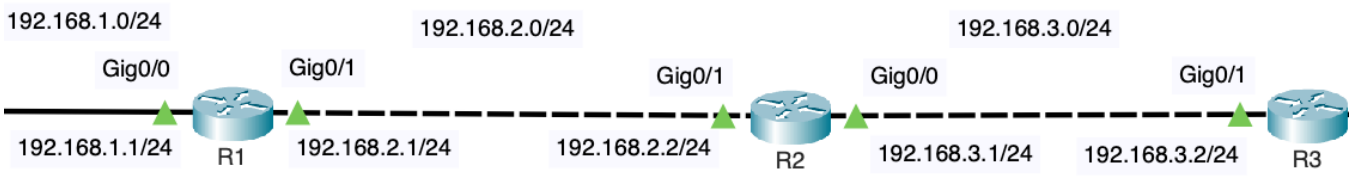
1. **Traceroute from R1 to 192.168.4.1**:
    - Use the traceroute command on R1 to trace the path packets take to reach the IP address 192.168.4.1 (which is on R3's LAN).




```
R1# traceroute 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.4.1

  1 192.168.2.2  0 msec  0 msec  0 msec
  2 192.168.3.2  4 msec 4 msec 4 msec
```

Explanation: This output shows that packets from R1 first go to the next-hop router R2 (192.168.2.2), and then to R3 (192.168.3.2) before reaching the final destination (192.168.4.1). Each hop represents a different router or device along the path.

2. **Traceroute from R3 to 192.168.1.1**:
   - Similarly, you can use traceroute on R3 to trace the path to 192.168.1.1 (which is on R1's LAN).




```
R3# traceroute 192.168.1.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 192.168.3.1  0 msec  0msec  0 msec
  2 192.168.2.1  4 msec 4 msec 4 msec
```

Explanation: This output indicates that packets from R3 first go to the next-hop router R2 (192.168.3.1), and then to R1 (192.168.2.1) before reaching the final destination (192.168.1.1).

Conclusion:

The traceroute command helps verify that the network paths are functioning as expected and that packets are taking the correct routes through the network. This final step confirms that your network is properly configured and that all routers can communicate with one another effectively.

Congratulations! You've successfully completed the lab!