Experiment 1: Basic Networking Commands in Cisco Packet Tracer

Objective:

To learn and use basic networking commands to verify network configuration and connectivity between devices.

Step-by-Step Procedure:

Step 1: Open Cisco Packet Tracer

1. Launch **Cisco Packet Tracer** on your system.

Step 2: Create a Simple Network Topology

1. **Drag and drop devices** from the device panel:
    - **1 Switch** (e.g., 2960 Series)
    - **2 PCs**
2. **Connect devices using cables**:
    - Use **Copper Straight-Through Cable** to connect:
        - **PC0 → Switch (FastEthernet 0/1)**
        - **PC1 → Switch (FastEthernet 0/2)**

Step 3: Assign IP Addresses to PCs

1. Click on **PC0 → Desktop → IP Configuration**:
    - **IP Address:** `192.168.1.1`
    - **Subnet Mask:** `255.255.255.0`
    - **Gateway:** Leave it blank.
2. Click on **PC1 → Desktop → IP Configuration**:
    - **IP Address:** `192.168.1.2`
    - **Subnet Mask:** `255.255.255.0`

- **Gateway:** Leave it blank.

Step 4: Test Network Connectivity Using Basic Commands

1. Check Assigned IP Address

- Open **Command Prompt** on **PC0**:

Type:
ipconfig

-
- It will display the **IP address** and **subnet mask**.

2. Check Connectivity Using Ping

- Open **Command Prompt** on **PC0**:

Type:
ping 192.168.1.2

-
- If successful, it will show **Reply from 192.168.1.2**.

3. Trace the Route to Destination

- Open **Command Prompt** on **PC0**:

Type:
tracert 192.168.1.2

-
- Since it's a simple LAN, the route will show a **direct connection**.

4. View ARP Table (MAC Address Learning)

- Open **Command Prompt** on **PC0**:

Type:
arp -a

-
- This will display the **MAC address** of **PC1**.

5. View Active Network Connections

- Open **Command Prompt** on **PC0**:

Type:
netstat

  -
  - This will show active network connections.

6. Resolve Domain Name to IP Address (if DNS is configured)

- Open **Command Prompt** on **PC0**:

Type:
nslookup google.com

  -
  - It will return the **IP address of Google's DNS server**.

Step 5: Verify Results

- Ensure **PC0 and PC1** can communicate using `ping`.
- Check if the **ARP table contains MAC addresses**.
- Confirm **IP configuration** using `ipconfig`.

Step 6: Save the Packet Tracer File

1. Click **File** → **Save As**.
2. Save your experiment as **Basic_Networking_Commands.pkt**.

Conclusion

By performing this experiment, you learned how to:

- Assign **IP addresses** in a LAN.
- Use **basic networking commands** (`ping`, `tracert`, `arp -a`, `netstat`, `nslookup`).
- Verify **connectivity and network settings**.

This completes **Experiment 1: Basic Networking Commands**! 🚀 Let me know if you need more details.

Experiment 2: Simple LAN Configuration in Cisco Packet Tracer

Objective:

To create a **Local Area Network (LAN)** using a **switch and PCs**, assign **IP addresses**, and verify connectivity between devices.

Step-by-Step Procedure:

Step 1: Open Cisco Packet Tracer

1. **Launch Cisco Packet Tracer** on your system.

Step 2: Create a Simple LAN Topology

1. **Drag and drop the following devices** from the device panel:
    - **1 Switch** (e.g., 2960 Series)
    - **3 PCs**
2. **Connect the devices** using **copper straight-through cables**:
    - **PC0 → Switch (FastEthernet 0/1)**
    - **PC1 → Switch (FastEthernet 0/2)**
    - **PC2 → Switch (FastEthernet 0/3)**

Step 3: Assign IP Addresses to PCs

Each PC must have a **unique IP address** within the same subnet.

1. **Click on PC0** → Go to **Desktop** → **IP Configuration**:
    - **IP Address:** `192.168.1.1`
    - **Subnet Mask:** `255.255.255.0`
    - **Default Gateway:** Leave it blank.
2. **Click on PC1** → Go to **Desktop** → **IP Configuration**:
    - **IP Address:** `192.168.1.2`
    - **Subnet Mask:** `255.255.255.0`

    - **Default Gateway:** Leave it blank.
3. **Click on PC2** → Go to **Desktop** → **IP Configuration**:
    - **IP Address:** `192.168.1.3`
    - **Subnet Mask:** `255.255.255.0`
    - **Default Gateway:** Leave it blank.

Step 4: Test Network Connectivity

To verify that all devices are properly connected and can communicate with each other, use the **ping** command.

1. Check the assigned IP address

- Open **Command Prompt** on **PC0**:

Type:
ipconfig

    -
    - It should display **192.168.1.1** as the assigned IP address.

2. Test connectivity between PCs

- Open **Command Prompt** on **PC0**:

Type:
ping 192.168.1.2

    -

If the connection is successful, you should see:
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

    -

- Similarly, **test connectivity from PC1 to PC2**:

Open **Command Prompt** on **PC1** and type:
ping 192.168.1.3

    -
    - If successful, you should get a **reply**.

Step 5: Verify MAC Address Learning (Optional)

Each PC's **MAC address** should be stored in the switch's MAC address table.

1. Click on the **Switch** → Go to **CLI**.

Type the following command to check the MAC address table:
show mac address-table

2.
3. It should display a list of MAC addresses of the connected PCs.

Step 6: Save the Packet Tracer File

1. Click **File** → **Save As**.
2. Save your experiment as **Simple_LAN.pkt**.

Conclusion

By performing this experiment, you have:

- Created a **simple LAN** using a switch.
- Assigned **IP addresses** to PCs.
- Verified **network connectivity** using **ping**.
- Observed **MAC address learning** on the switch.

This completes **Experiment 2: Simple LAN**! 🚀 Let me know if you need further clarifications.

Experiment 3: Different Network Topologies in Cisco Packet Tracer

Objective:

To design and implement different network topologies (**Bus, Star, Ring, Mesh, and Hybrid**) in **Cisco Packet Tracer** and test connectivity between devices.

Step-by-Step Procedure for Each Topology:

1️⃣ Star Topology

📌 In **Star Topology**, all devices are connected to a central switch.

Step 1: Create the Topology

1. Open **Cisco Packet Tracer**.
2. **Drag and drop**:
    - **1 Switch (2960 Series)**
    - **4 PCs**
3. **Connect devices** using **Copper Straight-Through Cables**:
    - **PC0 → Switch (FastEthernet 0/1)**
    - **PC1 → Switch (FastEthernet 0/2)**
    - **PC2 → Switch (FastEthernet 0/3)**
    - **PC3 → Switch (FastEthernet 0/4)**

Step 2: Assign IP Addresses

1. Click on **PC0** → Go to **Desktop** → **IP Configuration**:
    - `192.168.1.1 / 255.255.255.0`
2. Click on **PC1** → `192.168.1.2 / 255.255.255.0`
3. Click on **PC2** → `192.168.1.3 / 255.255.255.0`
4. Click on **PC3** → `192.168.1.4 / 255.255.255.0`

Step 3: Verify Connectivity

Open **Command Prompt** on **PC0** and type:
ping 192.168.1.2
ping 192.168.1.3
ping 192.168.1.4

-
- If successful, you will receive **reply messages**.

2️⃣ Bus Topology

📌 In **Bus Topology**, all devices share a single communication medium.

Step 1: Create the Topology

1. **Drag and drop**:
    - **1 Hub**
    - **4 PCs**
2. **Connect devices** using **Copper Straight-Through Cables**:
    - **PC0 → Hub (Port 1)**
    - **PC1 → Hub (Port 2)**
    - **PC2 → Hub (Port 3)**
    - **PC3 → Hub (Port 4)**

Step 2: Assign IP Addresses

(Same as Star Topology)

Step 3: Verify Connectivity

- **Use the ping command** between PCs.
- Note: **Hubs broadcast traffic**, so communication may be slower.

3️⃣ Ring Topology

📌 In **Ring Topology**, each device is connected to two other devices in a loop.

Step 1: Create the Topology

1. **Drag and drop**:
    - **4 PCs**
    - **4 Switches**
2. **Connect devices using Copper Straight-Through Cables**:
    - **PC0 → Switch0**
    - **Switch0 → Switch1**
    - **Switch1 → PC1**
    - **PC1 → Switch2**
    - **Switch2 → Switch3**

    - **Switch3 → PC2**
    - **PC2 → Switch0** (Complete the ring)

Step 2: Assign IP Addresses

(Same as Star Topology)

Step 3: Verify Connectivity

- **Use ping commands** to check connectivity.

4️⃣ Mesh Topology

📌 In **Mesh Topology**, every device is connected to every other device.

Step 1: Create the Topology

1. **Drag and drop**:
    - **4 PCs**
2. **Connect each PC to all other PCs** using **Copper Cross-Over Cables**.

Step 2: Assign IP Addresses

(Same as Star Topology)

Step 3: Verify Connectivity

- **Ping every device from every other device**.

5️⃣ Hybrid Topology

📌 **Hybrid Topology** combines multiple topologies.

Step 1: Create the Topology

1. **Combine Star, Bus, and Ring topologies** using switches and hubs.
2. **Example:**
    - **PC0 & PC1 connected in Bus Topology using Hub**
    - **PC2 & PC3 connected in Star Topology using Switch**
    - **Ring Topology connects all switches together**

Step 2: Assign IP Addresses

(Same as Star Topology)

Step 3: Verify Connectivity

- Use **ping commands** to check connectivity.

Step 6: Save the Packet Tracer File

1. Click **File** → **Save As**.
2. Save your experiment as **Network_Topologies.pkt**.

Conclusion

By performing this experiment, you have: ✔ Created **Star, Bus, Ring, Mesh, and Hybrid topologies**.
✔ Assigned **IP addresses** and tested connectivity using **ping**.
✔ Understood how different topologies **affect communication**.

This completes **Experiment 3: Different Network Topologies**! 🚀 Let me know if you need clarifications. 😊

Experiment 4: Router Configuration in Cisco Packet Tracer

Objective:

To configure a **router** to enable communication between two different networks.

Step-by-Step Procedure:

Step 1: Open Cisco Packet Tracer

1. **Launch Cisco Packet Tracer**.

Step 2: Create the Network Topology

1. **Drag and drop** the following devices:
    - **1 Router (e.g., 2911 Series)**
    - **2 Switches**
    - **4 PCs** (2 for each network)
2. **Connect the devices using cables**:
    - Use **Copper Straight-Through Cable** to connect:
        - **PC0 → Switch0 (FastEthernet 0/1)**
        - **PC1 → Switch0 (FastEthernet 0/2)**
        - **PC2 → Switch1 (FastEthernet 0/1)**
        - **PC3 → Switch1 (FastEthernet 0/2)**
    - Use **Copper Cross-Over Cable** to connect:
        - **Switch0 → Router (GigabitEthernet 0/0)**
        - **Switch1 → Router (GigabitEthernet 0/1)**

Step 3: Assign IP Addresses to PCs

Each PC needs a unique IP address in its respective network.

1. **Click on PC0** → Go to **Desktop → IP Configuration**:
    - **IP Address:** `192.168.1.2`

- **Subnet Mask:** `255.255.255.0`
- **Gateway:** `192.168.1.1`

2. **Click on PC1** → Set:
   - **IP Address:** `192.168.1.3`
   - **Subnet Mask:** `255.255.255.0`
   - **Gateway:** `192.168.1.1`
3. **Click on PC2** → Set:
   - **IP Address:** `192.168.2.2`
   - **Subnet Mask:** `255.255.255.0`
   - **Gateway:** `192.168.2.1`
4. **Click on PC3** → Set:
   - **IP Address:** `192.168.2.3`
   - **Subnet Mask:** `255.255.255.0`
   - **Gateway:** `192.168.2.1`

Step 4: Configure the Router

1. **Click on the Router** → Go to the **CLI (Command Line Interface)**.

Enter Configuration Mode:
enable
configure terminal

2.

Configure GigabitEthernet 0/0 (Network 1)
interface GigabitEthernet 0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit

3.

Configure GigabitEthernet 0/1 (Network 2)
interface GigabitEthernet 0/1
ip address 192.168.2.1 255.255.255.0
no shutdown
exit

4.

Enable Routing
ip routing
exit

5.

Step 5: Test Connectivity

Open **Command Prompt** on **PC0** and type:
ping 192.168.1.1 (Router's Gateway for PC0's Network)
ping 192.168.2.2 (PC2 in the other network)

1.
2. If successful, you will receive **Reply messages**.

Step 6: Save the Packet Tracer File

1. Click **File** → **Save As**.
2. Save as **Router_Configuration.pkt**.

Conclusion

By performing this experiment, you have: ✔ Configured a **router** to enable communication between two networks.
✔ Assigned **IP addresses and gateways**.
✔ Tested network connectivity using **ping**.

This completes **Experiment 4: Router Configuration**! 🚀 Let me know if you need clarifications. 😊

Experiment 5: VLAN Configuration in Cisco Packet Tracer

Objective:

To configure **VLANs (Virtual LANs)** on a switch to segment network traffic.

Step-by-Step Procedure:

Step 1: Open Cisco Packet Tracer

1. **Launch Cisco Packet Tracer**.

Step 2: Create the Network Topology

1. **Drag and drop the following devices**:
   - **1 Switch (e.g., 2960 Series)**
   - **4 PCs**
2. **Connect the devices using Copper Straight-Through Cables**:
   - **PC0 → Switch (FastEthernet 0/1)**
   - **PC1 → Switch (FastEthernet 0/2)**
   - **PC2 → Switch (FastEthernet 0/3)**
   - **PC3 → Switch (FastEthernet 0/4)**

Step 3: Assign IP Addresses to PCs

Each VLAN should have its own network.

1. **Click on PC0** → Go to **Desktop** → **IP Configuration**:
   - **IP Address:** `192.168.10.2`
   - **Subnet Mask:** `255.255.255.0`
   - **Gateway:** `192.168.10.1`
2. **Click on PC1** → Set:
   - **IP Address:** `192.168.10.3`
   - **Subnet Mask:** `255.255.255.0`

- **Gateway:** `192.168.10.1`

3. **Click on PC2** → Set:
   - **IP Address:** `192.168.20.2`
   - **Subnet Mask:** `255.255.255.0`
   - **Gateway:** `192.168.20.1`
4. **Click on PC3** → Set:
   - **IP Address:** `192.168.20.3`
   - **Subnet Mask:** `255.255.255.0`
   - **Gateway:** `192.168.20.1`

Step 4: Configure VLANs on the Switch

1. **Click on the Switch** → Go to **CLI (Command Line Interface)**.

Enter Configuration Mode:
enable
configure terminal

2.

Create VLANs:
vlan 10
name Sales
vlan 20
name HR
exit

3.
4. **Assign Ports to VLANs**:

Assign FastEthernet 0/1 & 0/2 to VLAN 10 (Sales):
interface FastEthernet 0/1
switchport mode access
switchport access vlan 10
exit

interface FastEthernet 0/2
switchport mode access
switchport access vlan 10
exit

-

Assign FastEthernet 0/3 & 0/4 to VLAN 20 (HR):

```
interface FastEthernet 0/3
switchport mode access
switchport access vlan 20
exit

interface FastEthernet 0/4
switchport mode access
switchport access vlan 20
exit
```

-

Step 5: Verify VLAN Configuration

Type the following command to verify VLAN assignments:

```
show vlan brief
```

1.
   - **Ports Fa0/1 & Fa0/2 should be in VLAN 10**.
   - **Ports Fa0/3 & Fa0/4 should be in VLAN 20**.

Step 6: Test Connectivity

PC0 (Sales VLAN) should be able to ping PC1 (Sales VLAN):

```
ping 192.168.10.3
```

-

PC2 (HR VLAN) should be able to ping PC3 (HR VLAN):

```
ping 192.168.20.3
```

-
- **PC0 (Sales VLAN) should NOT be able to ping PC2 (HR VLAN)** because VLANs **isolate traffic**.

Step 7: Save the Packet Tracer File

1. Click **File** → **Save As**.
2. Save as **VLAN_Configuration.pkt**.

Conclusion

By performing this experiment, you have: ✔ Created **VLANs** to segment network traffic.
- ✔ Assigned **specific ports** to different VLANs.
- ✔ Verified VLAN configuration using **CLI commands**.
- ✔ Tested VLAN **communication and isolation**.

This completes **Experiment 5: VLAN Configuration**! 🚀 Let me know if you need clarifications. 😊

Experiment 6: WLAN & DHCP Configuration in Cisco Packet Tracer

Objective:

To configure **Wireless LAN (WLAN)** and **Dynamic Host Configuration Protocol (DHCP)** in Cisco Packet Tracer to provide automatic IP addressing to wireless and wired devices.

Step-by-Step Procedure:

Step 1: Open Cisco Packet Tracer

1. **Launch Cisco Packet Tracer**.

Part 1: WLAN Configuration (Wireless Network Setup)

Step 2: Create the Wireless Network Topology

1. **Drag and drop the following devices**:
    - **1 Wireless Router (e.g., Linksys)**
    - **2 PCs (wired connection)**
    - **2 Laptops (wireless connection)**
    - **1 Smartphone (wireless connection)**
2. **Connect the wired devices using Copper Straight-Through Cables**:
    - **PC0 → Wireless Router (Ethernet 1)**
    - **PC1 → Wireless Router (Ethernet 2)**

Step 3: Configure the Wireless Router

1. **Click on the Wireless Router** → Go to the **GUI (Graphical User Interface) tab**.
2. **Go to Wireless Settings**:
    - **Network Name (SSID):** `MyWiFi`
    - **Wireless Mode:** Mixed
    - **Channel:** Auto

    - **SSID Broadcast:** Enabled
3. **Enable Security**:
    - **Security Mode:** WPA2 Personal
    - **Passphrase:** `12345678`
    - Click **Save Settings**.

Step 4: Connect Wireless Devices to WiFi

1. **Click on Laptop0** → Go to **Desktop** → **PC Wireless**:
    - **Select the WiFi network (`MyWiFi`)**.
    - Enter the **password (`12345678`)**.
    - Click **Connect**.
2. **Repeat the same steps for Laptop1 and Smartphone**.

Part 2: DHCP Configuration (Automatic IP Address Assignment)

Step 5: Enable DHCP on the Wireless Router

1. **Click on the Wireless Router** → Go to **GUI** → **Setup** → **Network Setup**.
2. **Under LAN Settings**:
    - **Enable DHCP Server:** Yes
    - **Start IP Address:** `192.168.1.100`
    - **Maximum Number of Users:** `50`
    - **Subnet Mask:** `255.255.255.0`
    - **Gateway (Router's IP Address):** `192.168.1.1`
    - Click **Save Settings**.

Step 6: Configure the PCs to Obtain IP Automatically

1. **Click on PC0** → Go to **Desktop** → **IP Configuration**.
    - Select **DHCP**.
    - The PC should receive an IP **(e.g., 192.168.1.100)**.
2. **Repeat the same steps for PC1, Laptop0, Laptop1, and Smartphone**.

Step 7: Verify DHCP and Wireless Connectivity

1. **Check assigned IP addresses**:

Click on any device → **Desktop** → **Command Prompt** → Type:
ipconfig

    -
    - The device should display an IP from the **DHCP range (192.168.1.100 - 192.168.1.150)**.
2. **Test connectivity using ping**:

From **PC0**, open **Command Prompt** and type:
ping 192.168.1.101 (Another device's IP)

    -
    - If successful, you will receive **reply messages**.

Step 8: Save the Packet Tracer File

1. Click **File** → **Save As**.
2. Save as **WLAN_DHCP_Configuration.pkt**.

Conclusion

By performing this experiment, you have: ✔ Configured a **Wireless LAN (WLAN) with security**.
✔ Enabled a **DHCP Server** to provide automatic IP addresses.
✔ Connected both **wired and wireless** devices to the network.
✔ Verified connectivity using **ping and ipconfig**.

This completes **Experiment 6: WLAN & DHCP Configuration**! 🚀 Let me know if you need clarifications. 😊

Experiment 7: Spanning Tree Protocol (STP) Configuration in Cisco Packet Tracer

Objective:

To configure **Spanning Tree Protocol (STP)** in Cisco Packet Tracer to prevent network loops in a redundant switched network.

Step-by-Step Procedure:

Step 1: Open Cisco Packet Tracer

1. **Launch Cisco Packet Tracer**.

Step 2: Create the Network Topology

1. **Drag and drop the following devices**:
    - **3 Switches (e.g., 2960 Series)**
    - **4 PCs** (for testing connectivity)
2. **Connect the devices using Copper Straight-Through Cables**:
    - **PC0 → Switch0 (FastEthernet 0/1)**
    - **PC1 → Switch1 (FastEthernet 0/1)**
    - **PC2 → Switch2 (FastEthernet 0/1)**
    - **PC3 → Switch2 (FastEthernet 0/2)**
    - **Switch0 → Switch1 (FastEthernet 0/2 to FastEthernet 0/2)**
    - **Switch1 → Switch2 (FastEthernet 0/3 to FastEthernet 0/3)**
    - **Switch2 → Switch0 (FastEthernet 0/4 to FastEthernet 0/4)**
3. This creates a **looped network topology**, which **STP will resolve**.

Step 3: Configure STP on Switches

1. **Click on Switch0** → Go to **CLI (Command Line Interface)**.

Enter Configuration Mode:
```
enable
configure terminal
```

2.

Set STP Mode to PVST (Per VLAN Spanning Tree):
```
spanning-tree mode pvst
```

3.

Set Switch0 as the Root Bridge (Lower priority makes it the Root Bridge):
```
spanning-tree vlan 1 priority 4096
exit
```

4.
5. **Repeat the same on Switch1 and Switch2 but with higher priorities**:

On Switch1:
```
enable
configure terminal
spanning-tree mode pvst
spanning-tree vlan 1 priority 8192
exit
```

-

On Switch2:
```
enable
configure terminal
spanning-tree mode pvst
spanning-tree vlan 1 priority 12288
exit
```

-

Step 4: Verify STP Operation

On each switch, type:
```
show spanning-tree
```

1.
   - **Switch0 should be the Root Bridge**.
   - **One of the redundant links will be in the "Blocking" state**.

Step 5: Test Connectivity

1. **Click on PC0** → Open **Command Prompt**.

Ping another PC (e.g., from PC0 to PC3):
ping 192.168.1.3

2.
    - If **successful**, STP is working correctly by preventing loops.

Step 6: Save the Packet Tracer File

1. Click **File** → **Save As**.
2. Save as **Spanning_Tree_Configuration.pkt**.

Conclusion

By performing this experiment, you have:

- ✔ Configured **Spanning Tree Protocol (STP)** to prevent loops.
- ✔ Set a **Root Bridge** to control traffic flow.
- ✔ Verified STP operation with **show spanning-tree**.
- ✔ Tested network connectivity after STP blocking.

This completes **Experiment 7: Spanning Tree Protocol Configuration**! 🚀 Let me know if you need clarifications. 😊

Experiment 8: MAC Learning Using ARP Table in Cisco Packet Tracer

Objective:

To observe how **MAC address learning** occurs in switches and how devices use the **ARP (Address Resolution Protocol) table** to communicate.

Step-by-Step Procedure:

Step 1: Open Cisco Packet Tracer

1. **Launch Cisco Packet Tracer**.

Step 2: Create the Network Topology

1. **Drag and drop the following devices**:
    - **1 Switch (e.g., 2960 Series)**
    - **3 PCs (PC0, PC1, and PC2)**
2. **Connect the devices using Copper Straight-Through Cables**:
    - **PC0 → Switch (FastEthernet 0/1)**
    - **PC1 → Switch (FastEthernet 0/2)**
    - **PC2 → Switch (FastEthernet 0/3)**

Step 3: Assign IP Addresses to PCs

1. **Click on PC0** → Go to **Desktop** → **IP Configuration**:
    - **IP Address:** `192.168.1.1`
    - **Subnet Mask:** `255.255.255.0`
2. **Click on PC1** → Set:
    - **IP Address:** `192.168.1.2`
    - **Subnet Mask:** `255.255.255.0`
3. **Click on PC2** → Set:
    - **IP Address:** `192.168.1.3`

- **Subnet Mask:** `255.255.255.0`

Step 4: Check the ARP Table Before Communication

1. **Click on PC0** → Open **Command Prompt**.

Type:
arp -a

2.
   - Since **no communication has occurred**, the **ARP table should be empty**.

Step 5: Test Communication Between Devices

On PC0, type:
ping 192.168.1.2

1.
   - PC0 will send an **ARP request** to learn the MAC address of PC1.
   - After the ARP process, PC0 will successfully **ping PC1**.

Check the ARP Table Again:
arp -a

2.
   - The **MAC address of PC1** should now be recorded in the table.

Repeat the same steps from PC1 to PC2:
ping 192.168.1.3

3.
   - This will add PC2's MAC address to PC1's ARP table.

Step 6: Verify MAC Learning on the Switch

1. **Click on the Switch** → Go to **CLI (Command Line Interface)**.

Type the following command to view the MAC address table:
enable
show mac address-table

2.
    - This will display the **learned MAC addresses** and the corresponding **ports**.

Step 7: Save the Packet Tracer File

1. Click **File** → **Save As**.
2. Save as **MAC_Learning_ARP_Experiment.pkt**.

Conclusion

By performing this experiment, you have:

- ✔ Observed how a switch **learns MAC addresses dynamically**.
- ✔ Used the **ARP table** to store MAC addresses for communication.
- ✔ Verified MAC learning using **show mac address-table**.
- ✔ Confirmed ARP resolution using **arp -a** on PCs.

This completes **Experiment 8: MAC Learning Using ARP Table**! 🚀 Let me know if you need any clarifications. 😊

Experiment 9: Subnetting in Cisco Packet Tracer

Objective:

To implement **Subnetting** in Cisco Packet Tracer and verify communication between different subnets using a **Router**.

Step-by-Step Procedure:

Step 1: Open Cisco Packet Tracer

1. **Launch Cisco Packet Tracer**.

Step 2: Plan the Subnetting

Assume we have a **Class C network:** `192.168.1.0/24` and we need **four subnets**.

1. **Subnet Calculation** (Using **Subnet Mask** `255.255.255.192 (/26)`):
    - **Subnet 1:** `192.168.1.0/26` (1st subnet, range: `192.168.1.1` to `192.168.1.62`, Broadcast: `192.168.1.63`)
    - **Subnet 2:** `192.168.1.64/26` (2nd subnet, range: `192.168.1.65` to `192.168.1.126`, Broadcast: `192.168.1.127`)
    - **Subnet 3:** `192.168.1.128/26` (3rd subnet, range: `192.168.1.129` to `192.168.1.190`, Broadcast: `192.168.1.191`)
    - **Subnet 4:** `192.168.1.192/26` (4th subnet, range: `192.168.1.193` to `192.168.1.254`, Broadcast: `192.168.1.255`)

Step 3: Create the Network Topology

1. **Drag and drop the following devices**:
    - **1 Router (Router0)**
    - **2 Switches (Switch0 & Switch1)**

- **4 PCs (PC0, PC1, PC2, PC3)**

2. **Connect the devices using Copper Straight-Through Cables**:

   - **PC0 → Switch0 (FastEthernet 0/1)**
   - **PC1 → Switch0 (FastEthernet 0/2)**
   - **PC2 → Switch1 (FastEthernet 0/1)**
   - **PC3 → Switch1 (FastEthernet 0/2)**
   - **Switch0 → Router0 (GigabitEthernet 0/0)**
   - **Switch1 → Router0 (GigabitEthernet 0/1)**

Step 4: Configure IP Addresses on PCs

Subnet 1 (Switch0 - PC0, PC1)

1. **Click on PC0** → Go to **Desktop** → **IP Configuration**:
   - **IP Address:** `192.168.1.10`
   - **Subnet Mask:** `255.255.255.192`
   - **Gateway:** `192.168.1.1`
2. **Click on PC1** → Go to **Desktop** → **IP Configuration**:
   - **IP Address:** `192.168.1.20`
   - **Subnet Mask:** `255.255.255.192`
   - **Gateway:** `192.168.1.1`

Subnet 2 (Switch1 - PC2, PC3)

3. **Click on PC2** → Go to **Desktop** → **IP Configuration**:
   - **IP Address:** `192.168.1.70`
   - **Subnet Mask:** `255.255.255.192`
   - **Gateway:** `192.168.1.65`
4. **Click on PC3** → Go to **Desktop** → **IP Configuration**:
   - **IP Address:** `192.168.1.80`
   - **Subnet Mask:** `255.255.255.192`
   - **Gateway:** `192.168.1.65`

Step 5: Configure the Router (Inter-VLAN Routing)

1. **Click on Router0** → Go to **CLI (Command Line Interface)**.

Enter Configuration Mode:

```
enable
configure terminal
```

2.
3. **Assign IP Addresses to Router Interfaces**:

For Subnet 1 (Connected to Switch0):

```
interface GigabitEthernet0/0
ip address 192.168.1.1 255.255.255.192
no shutdown
exit
```

-

For Subnet 2 (Connected to Switch1):

```
interface GigabitEthernet0/1
ip address 192.168.1.65 255.255.255.192
no shutdown
exit
```

-

Enable Routing:

```
exit
write memory
```

4.

Step 6: Test Connectivity

1. **Go to PC0** → Open **Command Prompt**.

Ping PC2 in a different subnet:

```
ping 192.168.1.70
```

2.
   - If **successful**, the router is correctly routing packets between subnets.

Ping the Router Gateway from PC1:
ping 192.168.1.1

3.
    - If **successful**, PC1 can communicate with the router.

Step 7: Save the Packet Tracer File

1. Click **File** → **Save As**.
2. Save as **Subnetting_Experiment.pkt**.

Conclusion

By performing this experiment, you have:

- ✔ **Subnetted a Class C network into multiple subnets**.
- ✔ **Configured IP addresses and subnet masks on PCs**.
- ✔ **Configured a Router for inter-subnet communication**.
- ✔ **Verified communication using the ping command**.

This completes **Experiment 9: Subnetting in Cisco Packet Tracer**! 🚀 Let me know if you need any clarifications. 😊

Experiment 10: Manual Routing in Cisco Packet Tracer

Objective:

To configure **Manual (Static) Routing** in Cisco Packet Tracer to enable communication between multiple networks without using dynamic routing protocols.

Step-by-Step Procedure:

Step 1: Open Cisco Packet Tracer

1. **Launch Cisco Packet Tracer**.

Step 2: Create the Network Topology

1. **Drag and drop the following devices**:
    - **2 Routers (Router0 & Router1)**
    - **2 Switches (Switch0 & Switch1)**
    - **4 PCs (PC0, PC1, PC2, PC3)**
2. **Connect the devices using Copper Straight-Through Cables**:
    - **PC0 → Switch0 (FastEthernet 0/1)**
    - **PC1 → Switch0 (FastEthernet 0/2)**
    - **PC2 → Switch1 (FastEthernet 0/1)**
    - **PC3 → Switch1 (FastEthernet 0/2)**
    - **Switch0 → Router0 (GigabitEthernet 0/0)**
    - **Switch1 → Router1 (GigabitEthernet 0/0)**
    - **Router0 → Router1 (via Serial Connection on Serial0/0/0)**

Step 3: Assign IP Addresses and Subnet Masks

We will use the following network subnets:

1. **Network 1 (PC0, PC1, Switch0, Router0):** `192.168.1.0/24`
2. **Network 2 (PC2, PC3, Switch1, Router1):** `192.168.2.0/24`
3. **Network 3 (Router0 to Router1 link):** `192.168.3.0/30`

Step 4: Configure IP Addresses on PCs

Subnet 1 (Switch0 - PC0, PC1)

1. **Click on PC0** → Go to **Desktop** → **IP Configuration**:
   - **IP Address:** `192.168.1.10`
   - **Subnet Mask:** `255.255.255.0`
   - **Gateway:** `192.168.1.1`
2. **Click on PC1** → Go to **Desktop** → **IP Configuration**:
   - **IP Address:** `192.168.1.20`
   - **Subnet Mask:** `255.255.255.0`
   - **Gateway:** `192.168.1.1`

Subnet 2 (Switch1 - PC2, PC3)

3. **Click on PC2** → Go to **Desktop** → **IP Configuration**:
   - **IP Address:** `192.168.2.10`
   - **Subnet Mask:** `255.255.255.0`
   - **Gateway:** `192.168.2.1`
4. **Click on PC3** → Go to **Desktop** → **IP Configuration**:
   - **IP Address:** `192.168.2.20`
   - **Subnet Mask:** `255.255.255.0`
   - **Gateway:** `192.168.2.1`

Step 5: Configure Routers

Router0 Configuration

1. **Click on Router0** → Go to **CLI**.

Enter Configuration Mode:
enable
configure terminal

2.

Configure the GigabitEthernet Interface (Connecting to Switch0):

```
 interface GigabitEthernet0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
```

3.

Configure the Serial Interface (Connecting to Router1):

```
 interface Serial0/0/0
ip address 192.168.3.1 255.255.255.252
clock rate 64000
no shutdown
exit
```

4.

Set Static Route to Network 2 (192.168.2.0/24):

```
 ip route 192.168.2.0 255.255.255.0 192.168.3.2
exit
```

5.

Router1 Configuration

1. **Click on Router1** → Go to **CLI**.

Enter Configuration Mode:

```
 enable
configure terminal
```

2.

Configure the GigabitEthernet Interface (Connecting to Switch1):

```
 interface GigabitEthernet0/0
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
```

3.

Configure the Serial Interface (Connecting to Router0):

```
interface Serial0/0/0
ip address 192.168.3.2 255.255.255.252
no shutdown
exit
```

4.

Set Static Route to Network 1 (192.168.1.0/24):

```
ip route 192.168.1.0 255.255.255.0 192.168.3.1
exit
```

5.

Save Configuration:

```
write memory
```

6.

Step 6: Test Connectivity

1. **Go to PC0** → Open **Command Prompt**.

Ping PC2 (from different subnet):

```
ping 192.168.2.10
```

2.
    - If **successful**, the routers are correctly routing packets.

Ping the Router Gateway from PC1:

```
ping 192.168.1.1
```

3.
    - If **successful**, PC1 can communicate with Router0.

Step 7: Save the Packet Tracer File

1. Click **File** → **Save As**.
2. Save as **Manual_Routing_Experiment.pkt**.

Conclusion

By performing this experiment, you have:

- ✔ **Manually configured static routing** on routers.
- ✔ **Assigned IP addresses to PCs and routers** for proper subnetting.
- ✔ **Used `ip route` command to manually define routing paths**.
- ✔ **Verified communication between networks using the `ping` command**.

This completes **Experiment 10: Manual Routing in Cisco Packet Tracer**! 🚀 Let me know if you need any clarifications. 😊