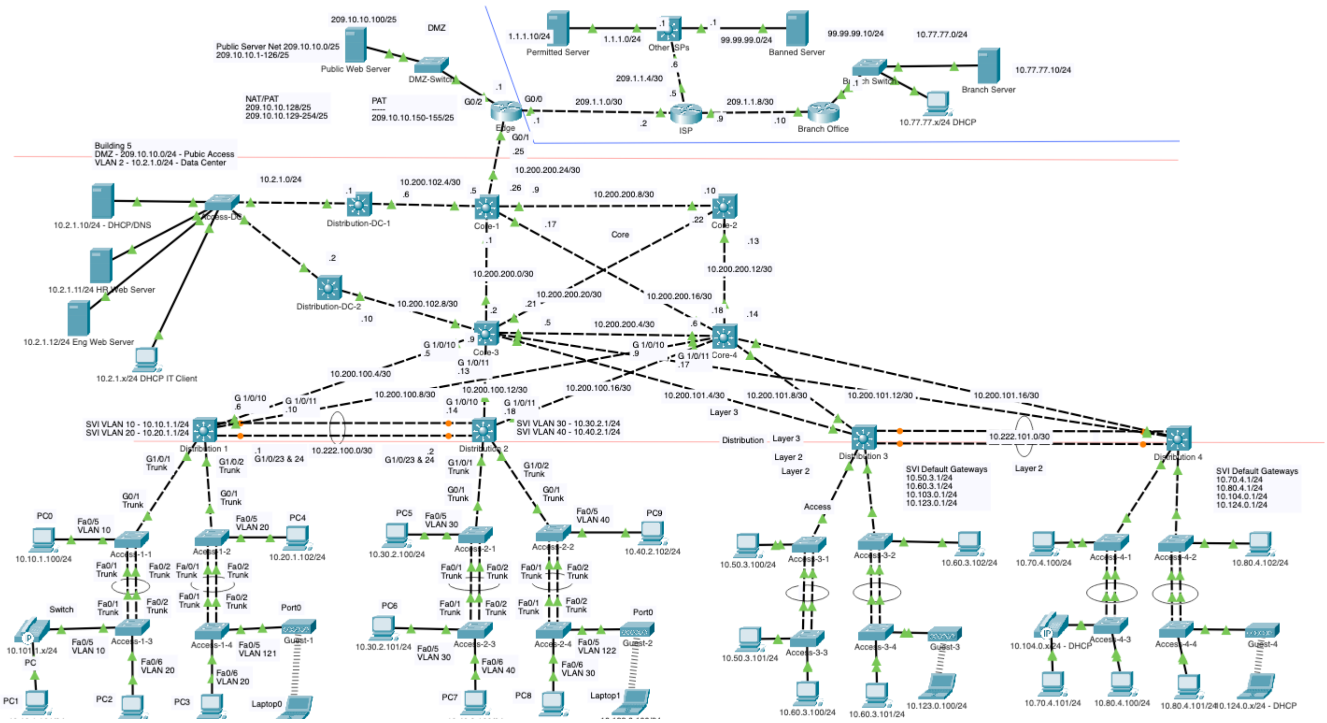
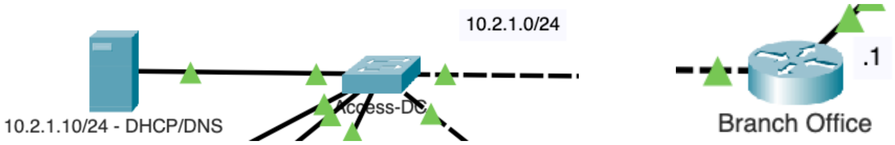
Enterprise Network

Lab 7: DHCPv4 Servers, DHCPv4 Relay, RSTP and PortFast





Lab Objectives

By the end of this lab, students will be able to:

- Explain the purpose and operation of DHCPv4, including the client-server lease process.
- Configure a DHCPv4 server on a dedicated server device to support multiple VLANs.
- Create and manage DHCP address pools for different user, VoIP, and WLAN VLANs.
- Identify and troubleshoot issues related to DHCP failure, including APIPA address assignment.
- Understand the need for DHCP relay in multi-VLAN environments.
- Configure **ip helper-address** on Layer 3 devices to enable DHCP relay.
- Enable and verify Rapid Spanning Tree Protocol (RSTP) to improve network convergence.
- Configure PortFast and BPDU Guard on access layer switch ports to support fast device startup and protect against loops.
- Observe BPDU Guard behavior when an unauthorized switch is connected to a PortFast-enabled port.
- Configure a Cisco router to act as a DHCPv4 server for a remote branch network.
- Verify successful DHCP address assignment on client devices across different VLANs and networks.

Configure SVIs for VoIP and WLAN Networks

We have already configured the SVIs (default gateway IP addresses) on Distribution 1 and Distribution 2 switches for VLANs 10, 20, 30 and 40. To support IP address assignment and routing for voice and wireless clients, we need to configure the SVIs on both distribution switches for the VoIP and WLAN VLANs.

On Distribution-1, we configured VLAN 101 for VoIP with IP address 10.101.0.1/24 and VLAN 121 for WLAN with IP address 10.121.0.1/24. Similarly, on Distribution-2, VLAN 102 was configured with IP address 10.102.0.1/24 for VoIP, and VLAN 122 with 10.122.0.1/24 for WLAN. These SVIs serve as the default gateways for devices in their respective VLANs and allow routing between the VLANs and the rest of the enterprise network.

```
distribution-1(config)# interface vlan 101
distribution-1(config-if)# ip add 10.101.0.1 255.255.255.0
distribution-1(config-if)# exit
distribution-1(config)# inter vlan 121
distribution-1(config-if)# ip add 10.121.0.1 255.255.255.0
distribution-1(config-if)# exit
```

```
distribution-2(config)# inter vlan 102
distribution-2(config-if)# ip add 10.102.0.1 255.255.255.0
distribution-2(config-if)# exit
distribution-2(config)# inter vlan 122
distribution-2(config-if)# ip add 10.122.0.1 255.255.255.0
distribution-2(config-if)# exit
```

DHCPv4 Service

Dynamic Host Configuration Protocol version 4 (DHCPv4) is a network protocol used to automatically assign IP addresses and other network configuration parameters to devices on a network. It enables devices to request and receive an IP address automatically from a DHCP server, eliminating the need for manual network configuration.

DHCPv4 operates on a client-server model where the DHCP server dynamically distributes network configuration parameters, such as IP addresses, subnet masks, default gateways, and DNS server addresses, to DHCP clients. This process involves a series of transactions including IP lease request, offer, acknowledgment, and renewal, ensuring that devices can seamlessly connect and communicate over the network without IP conflicts or manual setup.

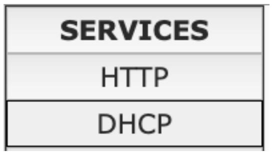
DHCPv4 Server

In this lab, we will configure the device with IP address **10.2.1.10/24** as the **DHCPv4 server**. It will provide dynamic IPv4 addressing for **all networks with client devices**, including VLANs 10, 20, 30, and 40.

Enabling and Configuring the DHCPv4 Service

Configure the 10.2.1.10/24 DHCPv4 server:

1. Select **Services** tab
2. Under Services select **DHCP**



Edit the existing serverPool

serverPool: Packet Tracer automatically created this DHCPv4 pool for its local network. This pool **cannot be edited**, so we will make put this pool in our **parking-lot VLAN**.

1. Change the **Default Gateway** to: 0.0.0.0
2. Change the **DNS Server** to: 0.0.0.0
3. Change the **Start IP Address** to: 10.255.0.0
4. Change the **Maximum Number of Users** to: 0
5. Select: **Save**

Adding pools for user IPv4 networks (VLANs)

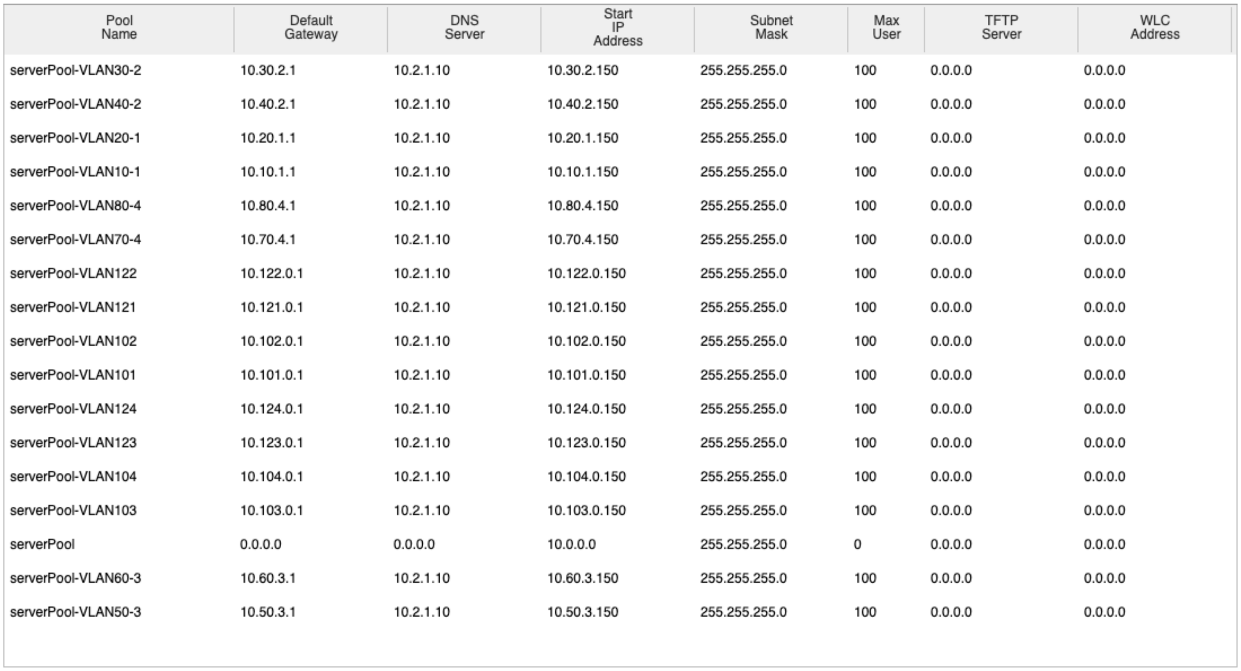
Make the following changes:

1. **Pool Name**: serverPool-VLAN10
2. **Default Gateway**: 10.10.1.1
3. **DNS Server**: 10.2.1.10
4. **Start IP Address**: 10.10.1.150
5. **Subnet Mask**: 255.255.255.0
6. **Maximum Number of Users**: 100
7. Select **Add** (Do **not** click “Save”, you will overwrite the previous entry, unless you need to make an edit.)

Repeat this process for all the networks in the table below. You will only need to modify the Pool Name, Default Gateway and Start IP Address for each. Some of the networks are for the other switch block.

Pool Name	Default Gateway	DNS Server	Start Address	Subnet Mask	Max Users
serverPool	0.0.0.0	0.0.0.0	10.10.0.150	255.255.255.0	100
serverPool-VLAN10	10.10.1.1	10.2.1.10	10.10.1.150	255.255.255.0	100
serverPool-VLAN20	10.20.1.1	10.2.1.10	10.20.1.150	255.255.255.0	100
serverPool-VLAN30	10.30.2.1	10.2.1.10	10.30.2.150	255.255.255.0	100
serverPool-VLAN40	10.40.2.1	10.2.1.10	10.40.2.150	255.255.255.0	100
serverPool-VLAN50	10.50.3.1	10.2.1.10	10.50.3.150	255.255.255.0	100
serverPool-VLAN60	10.60.3.1	10.2.1.10	10.60.3.150	255.255.255.0	100
serverPool-VLAN70	10.70.4.1	10.2.1.10	10.70.4.150	255.255.255.0	100
serverPool-VLAN80	10.80.4.1	10.2.1.10	10.80.4.150	255.255.255.0	100
serverPool-VLAN101	10.101.0.1	10.2.1.10	10.101.0.150	255.255.255.0	100
serverPool-VLAN102	10.102.0.1	10.2.1.10	10.102.0.150	255.255.255.0	100
serverPool-VLAN103	10.103.0.1	10.2.1.10	10.103.0.150	255.255.255.0	100
serverPool-VLAN104	10.104.0.1	10.2.1.10	10.104.0.150	255.255.255.0	100
serverPool-VLAN121	10.121.0.1	10.2.1.10	10.121.0.150	255.255.255.0	100
serverPool-VLAN122	10.122.0.1	10.2.1.10	10.122.0.150	255.255.255.0	100
serverPool-VLAN123	10.123.0.1	10.2.1.10	10.123.0.150	255.255.255.0	100
serverPool-VLAN124	10.124.0.1	10.2.1.10	10.124.0.150	255.255.255.0	100

When completed, your DHCPv4 server pool table should look as shown below. The order is the last entry is shown first.



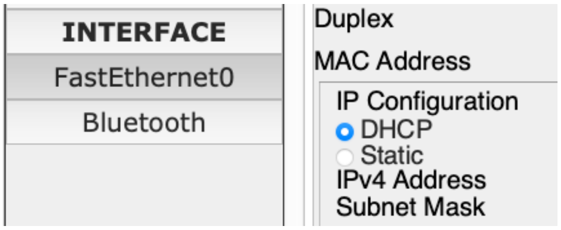
Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool-VLAN30-2	10.30.2.1	10.2.1.10	10.30.2.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN40-2	10.40.2.1	10.2.1.10	10.40.2.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN20-1	10.20.1.1	10.2.1.10	10.20.1.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN10-1	10.10.1.1	10.2.1.10	10.10.1.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN80-4	10.80.4.1	10.2.1.10	10.80.4.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN70-4	10.70.4.1	10.2.1.10	10.70.4.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN122	10.122.0.1	10.2.1.10	10.122.0.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN121	10.121.0.1	10.2.1.10	10.121.0.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN102	10.102.0.1	10.2.1.10	10.102.0.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN101	10.101.0.1	10.2.1.10	10.101.0.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN124	10.124.0.1	10.2.1.10	10.124.0.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN123	10.123.0.1	10.2.1.10	10.123.0.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN104	10.104.0.1	10.2.1.10	10.104.0.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN103	10.103.0.1	10.2.1.10	10.103.0.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool	0.0.0.0	0.0.0.0	10.0.0.0	255.255.255.0	0	0.0.0.0	0.0.0.0
serverPool-VLAN60-3	10.60.3.1	10.2.1.10	10.60.3.150	255.255.255.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN50-3	10.50.3.1	10.2.1.10	10.50.3.150	255.255.255.0	100	0.0.0.0	0.0.0.0

Enable DHCPv4 Service

Enable the DHCP service:

- **Service**: On

Preparing DHCPv4 Clients



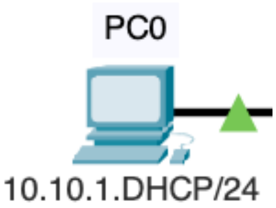
INTERFACE
FastEthernet0
Bluetooth

Duplex
MAC Address
IP Configuration
● DHCP
○ Static
IPv4 Address
Subnet Mask

For selected PCs of your choosing, connected to the bottom access layer switches, change the configuration for DHCPv4:

- Select the **Config** tab
- Select **Interface> FastEthernet0**
  - Select **DHCP**
- Note: You should check that Packet Tracer automatically made the change to **Global > Settings**

Note: You can configure all the clients PCs for DHCP if you wish and modify the display name.



Verify that these devices are configured for DHCPv4 but did not receive an IPv4 address. You will notice that the PC will have an IPv4 address starting with 169.254.x.x. The 169.254.x.x address, also known as an Automatic Private IP Addressing (APIPA) address, is assigned to a device when it fails to obtain an IP address from a DHCP server. This self-assigned IP range allows devices to communicate on the same local network, despite the absence of centralized DHCP configuration.

You can verify by either issuing the **ipconfig** command or ‘wanding’ over the device with the cursor.

Verify DHCPv4 Unsuccessful on other Networks/VLANs

An **APIPA address** (Automatic Private IP Addressing), in the range **169.254.0.0 to 169.254.255.255**, is automatically assigned to a device when it fails to obtain an IPv4 address from a DHCP server.

In our case, it is because their DHCP discover messages, which are Ethernet broadcasts, are staying within their broadcast domains. The DHCPv4 server is never receiving their DHCPv4 messages.

For example, the PC1, previously 10.10.1.101/24 is now shows as:

```
C:\> ipconfig

FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.........: FE80::260:47FF:FE33:61A1
   IPv6 Address....................: ::
   Autoconfiguration IPv4 Address..: 169.254.97.162
   Subnet Mask.....................: 255.255.0.0
   Default Gateway.................: ::
                                     0.0.0.0
C:\> ipconfig /release

   IP Address......................: 0.0.0.0
   Subnet Mask.....................: 0.0.0.0
   Default Gateway.................: 0.0.0.0
   DNS Server......................: 0.0.0.0

C:\> ipconfig /renew
DHCP request failed.

C:\> ipconfig /renew
DHCP request failed.

C:\>
```

DHCPv4 Relay

DHCP relay is needed when DHCP clients are on a different network or VLAN than the DHCP server. Since DHCP discovery messages are broadcasts, they do not cross router boundaries. By configuring a router or Layer 3 switch with the **ip helper-address** command, it can forward these DHCP requests as unicast messages to the DHCP server on another network, allowing centralized IP address management across multiple subnets.

The ip helper-address command

The **ip helper-address** command is used in routers to enable DHCPv4 relay, allowing DHCP broadcast requests from clients on one network segment to be forwarded to a DHCP server on a different segment. When a DHCP client sends a broadcast request for an IP address, routers typically do not forward these broadcasts to other networks.

By configuring an **ip helper-address**, the router converts the broadcast into a unicast packet and forwards it to the specified DHCP server's IP address. This mechanism enables devices on other networks or VLANs to communicate with a central DHCP server, facilitating centralized management of IP address allocation and reducing the need for multiple DHCP servers across different network segments.

The **ip helper-address** is typically configured on the Layer 3 switch or router that acts as the default gateway for the clients. This setup allows the gateway device to intercept and relay DHCP broadcast requests from clients to the specified DHCP server, even if it's located on a different network segment.

The **ip helper-address** function takes DHCP broadcast packets (255.255.255.255) from clients, which are meant to be local to the subnet, and converts them into unicast packets. It then forwards these unicast packets directly to the specified DHCP server's IP address, allowing for cross-network DHCP service communication.

Configuring ip helper-address on Distribution-2 Switch

For example, the Layer 3 switch Distribution-1 is the default gateway for device on 10.10.1.0/24 (VLAN 10) network. The SVI, interface VLAN 10, has previously been configured to act as the default gateway for devices in this network.

The **ip helper-address 10.2.1.10 255.255.0.0** command will forward DHCP broadcast packets and converts the destination IPv4 address from 255.255.255.255 to 10.30.0.1.

```
distribution-1# show running-config | section interface Vlan10
interface Vlan10
 mac-address 0060.5cd7.5101
 ip address 10.10.1.1 255.255.255.0
distribution-1#

distribution-1(config)# interface vlan 10
distribution-1(config-if)# ip helper-address 10.2.1.10
distribution-1(config-if)# end

distribution-1# show running-config | section interface Vlan10
interface Vlan10
 mac-address 0060.5cd7.5101
 ip address 10.10.1.1 255.255.255.0
 ip helper-address 10.2.1.10
```

FYI: DHCPv4 Relay IPv4 Addressing Details

If you want more detail on the operations of DHCPv4 relay, here is a break down the IPv4 addressing process in the DHCP relay scenario step by step:

1. **Client to Router**:
    - A **DHCP client** on the 10.10.1.0/24 network sends a **DHCP Discover** message to the network **broadcast** address, which is 255.255.255.255.
    - The **source IPv4 address is 0.0.0.0** (since the client doesn't have an IPv4 address yet), and the destination is the broadcast address.
    - The **destination MAC** address is a broadcast address **ff:ff:ff:ff:ff:ff**.
    - Importantly, the **client** includes its **own MAC address** in the **chaddr** field (Client Hardware Address) inside the DHCP Discover message. This field is later used by both the DHCP server and the relay agent to identify and respond to the correct client.
2. **Router to DHCP Server**:
    - The **router** (or **L3 switch**) with an SVI on interface **10.10.1.1 receives the DHCP Discover** broadcast.
    - Normally, a router would drop broadcasts, but with the **DHCP relay agent** feature (ip helper-address) enabled, it forwards the DHCP Discover as a **unicast** packet to the configured DHCP server.
    - The router sets the **giaddr (Gateway IP Address)** field to **10.10.1.1**, indicating to the DHCP server which **subnet the request came from**. This helps the DHCP server choose the correct address pool.
    - The **source IP address** in the new packet is **10.10.1.1**, and the **destination IP is the DHCP server** of **10.2.1.10**.
    - The **original chaddr field is preserved**, so the DHCP server can see the MAC address of the requesting client.
3. **DHCP Server to Router**:
    - The **DHCP server** receives the Discover message, examines the **giaddr** to determine the **appropriate IP pool**, and creates a **DHCP Offer** message.
    - This **Offer is unicast back** to the router's **giaddr** address — in this case, **10.10.1.1**.
    - The server **includes** the **proposed IP address** and keeps the client's MAC address in the **chaddr field, unchanged**.
4. **Router to Client**:
    - The router receives the DHCP Offer and now needs to forward it back to the correct client.
    - The **router** uses the **chaddr field** in the DHCP message to **determine the client's MAC address**. Since the client still does not have an assigned IP address, the router **cannot use IP unicast** to reach it.
    - Instead, the router encapsulates the IP unicast packet in a **Layer 2 frame**. The Ethernet frame may be:
        - a **broadcast** (ff:ff:ff:ff:ff:ff) if the router does not have an entry for the client in its ARP cache, or
        - a **unicast** to the client's MAC address if it was learned from the initial DHCP Discover by examining the **chaddr** field.
    - The **IP destination address** remains **255.255.255.255**, because the client has not yet accepted the offered IP address.
    - This ensures the DHCP Offer reaches the correct client, even though it does not yet have a usable IP address or fully initialized IP stack.

Verify DHCPv4 Clients

On the PCs you enabled for DHCP, you verify they can now obtain the IPv4 addressing from the DHCPv4 server.

Depending on the client operating system, the PC may need to resend its DHCPv4 discover message. This can be done on Windows using **ipconfig /release** and **ipconfig /renew**. You may receive a DHCP failed on Packet Tracer due to ARP delays and may need to issue the **ipconfig /renew** command more than once.

For example, PC1 (formerly 10.10.1.101/24):

```
C:\>ipconfig /release

C:\>ipconfig /renew
   IP Address......................: 10.10.1.150
   Subnet Mask.....................: 255.255.255.0
   Default Gateway.................: 10.10.1.1
   DNS Server......................: 10.2.1.10
```

Configuring ip helper-address on Distribution-1 and Distribution-2 Switches

Next, we will use a similar **ip helper-address** command for 10.2.1.10 on Distribution 1 and Distribution 2 switches for all user VLANs.

```
distribution-1(config)# interface vlan 10
distribution-1(config-if)# ip helper-address 10.2.1.10
distribution-1(config-if)# exit
distribution-1(config)# interface vlan 20
distribution-1(config-if)# ip helper-address 10.2.1.10
distribution-1(config-if)# exit
distribution-1(config)# interface vlan 101
distribution-1(config-if)# ip helper-address 10.2.1.10
distribution-1(config-if)# exit
distribution-1(config)# interface vlan 121
distribution-1(config-if)# ip helper-address 10.2.1.10
distribution-1(config-if)# exit
distribution-1(config)#
```

```
distribution-2(config)# interface vlan 30
distribution-2(config-if)# ip helper-address 10.2.1.10
distribution-2(config-if)# exit
distribution-2(config)# interface vlan 40
distribution-2(config-if)# ip helper-address 10.2.1.10
distribution-2(config-if)# exit
distribution-2(config)# interface vlan 102
distribution-2(config-if)# ip helper-address 10.2.1.10
distribution-2(config-if)# exit
distribution-2(config)# interface vlan 122
distribution-2(config-if)# ip helper-address 10.2.1.10
distribution-2(config-if)# exit
distribution-2(config)#
```

We can verify the end devices we configured for DHCP, including the VoIP phones and laptops are now receiving their IPv4 addressing from the DHCP server.

Note: You can wand over the VoIP phone to display its IPv4 address.

For example, Laptop0 (formerly 10.121.0.100/24):

```
C:\>ipconfig /release

C:\>ipconfig /renew
   IP Address......................: 10.121.1.150
   Subnet Mask.....................: 255.255.255.0
   Default Gateway.................: 10.121.1.1
   DNS Server......................: 10.2.1.10

C:\>
```

Rapid Spanning Tree, PortFast and BPDU Guard

Rapid Spanning Tree

By default, Cisco switches use **PVST+** (Per-VLAN Spanning Tree Plus), which runs a separate instance of the **original Spanning Tree Protocol (STP)** for each VLAN. While PVST+ offers per-VLAN load balancing and loop prevention, it uses the older **802.1D** standard, which has slower convergence times — often taking up to 30–50 seconds for ports to transition from blocking to forwarding.

To improve network responsiveness, we will configure the switches to use **Rapid Spanning Tree Protocol (RSTP)**, based on the **IEEE 802.1w** standard. RSTP dramatically reduces convergence time after a topology change, often down to just a few seconds. It introduces new port roles (like *alternate* and *backup*) and transitions ports to the forwarding state more quickly by eliminating the traditional listening and learning states.

We will use the command, **spanning-tree mode rapid-pvst**. This command changes the spanning tree mode from the default **PVST+** to **Rapid PVST+**, which retains per-VLAN spanning tree instances while using the faster convergence behavior of RSTP.

RSTP is compatible with legacy STP networks but provides a much faster and more efficient response to network changes. Before configuring features like **PortFast** and **BPDU Guard**, we will enable **RSTP mode** to ensure a faster-spanning tree environment that is better suited for modern access-layer devices like PCs, IP phones, and wireless access points.

RSTP follows the same core logic: elect root bridge → find best paths (RPs) → identify DPs → everything else is non-forwarding. But it **enhances the transition process**, removes dependency on long timers, and defines **new port roles** for faster loop prevention and recovery.

RSTP introduces **Alternate** and **Backup** port roles instead of just "Blocked." This speeds up convergence using **handshake-based** mechanism instead of relying on long timers (like max age and forward delay), especially during temporary loops during convergence. RSTP ports can **move to forwarding state immediately** in some cases (like edge ports or point-to-point links with proper agreement).

Configure all switches for RSTP

To enable **Rapid Spanning Tree Protocol (RSTP)** on a Cisco switch, use the global configuration command **spanning-tree mode rapid-pvst**. This command changes the spanning tree mode from the default **PVST+** to **Rapid PVST+**, which retains per-VLAN spanning tree instances while using the faster convergence behavior of RSTP. This mode is ideal for modern networks where quick recovery from link changes is important, especially at the access layer.

```
! Distribution switches
distribution-1(config)# spanning-tree mode rapid-pvst

distribution-2(config)# spanning-tree mode rapid-pvst

! Access switches
access1-1(config)# spanning-tree mode rapid-pvst

access1-2(config)# spanning-tree mode rapid-pvst

access1-3(config)# spanning-tree mode rapid-pvst

access1-4(config)# spanning-tree mode rapid-pvst

access2-1(config)# spanning-tree mode rapid-pvst

access2-2(config)# spanning-tree mode rapid-pvst

access2-3(config)# spanning-tree mode rapid-pvst

access2-4(config)# spanning-tree mode rapid-pvst
```

Configure the same commands for the other switch block.

```
! Distribution switches
distribution-3(config)# spanning-tree mode rapid-pvst

distribution-4(config)# spanning-tree mode rapid-pvst

! Access switches
Access3-1(config)# spanning-tree mode rapid-pvst

Access3-2(config)# spanning-tree mode rapid-pvst

Access3-3(config)# spanning-tree mode rapid-pvst

Access3-4(config)# spanning-tree mode rapid-pvst

Access4-1(config)# spanning-tree mode rapid-pvst

Access4-2(config)# spanning-tree mode rapid-pvst

Access4-3(config)# spanning-tree mode rapid-pvst

Access4-4(config)# spanning-tree mode rapid-pvst
```

Note: You will notice how much faster your topology converges with RSTP when starting Packet Tracer.

When RSTP is enabled using the **spanning-tree mode rapid-pvst** command, the switch will briefly **reconverge the spanning tree topology**. During this process, some ports may temporarily go **amber (orange)**, indicating they are transitioning through STP states such as *discarding* or *learning*. This is normal behavior and typically lasts only a few seconds, thanks to RSTP's fast convergence. Once reconvergence completes, ports will return to their forwarding state, and normal traffic flow will resume.

The **show spanning-tree summary** command can be used to verify RSTP is enabled on the switch.

```
access-1-1# show spanning-tree summary
Switch is in rapid-pvst mode
<output omitted>
```

Here's a comparison of **802.1D**, **PVST+**, and **RSTP**:

Feature	**802.1D (Classic STP)**	**PVST+ (Cisco)**	**RSTP (802.1w)**
Vendor	IEEE standard	Cisco proprietary	IEEE standard
VLAN Support	Single instance for all VLANs	One STP instance per VLAN	Single instance (but Cisco's Rapid-PVST+ adds per-VLAN support)
Convergence Speed	Slow (~30–50 seconds)	Slow (~30–50 seconds)	Fast (<10 seconds)
Port Roles	Root, Designated, Blocked	Same as 802.1D	Adds Alternate and Backup roles
Port States	Blocking, Listening, Learning, Forwarding	Same	Discards, Learning, Forwarding
Default on Cisco	No	**Yes (default)**	No (requires **spanning-tree mode rapid-pvst**)
Compatibility	Universal	Works with STP and CST	Backward-compatible with 802.1D

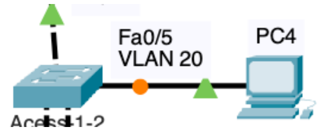
Summary

- **802.1D** = original standard STP, slow.
- **PVST+** = Cisco's per-VLAN version of 802.1D (also slow, but better load balancing).
- **RSTP (802.1w)** = modern, faster STP.
- **Rapid-PVST+** = Cisco's per-VLAN version of RSTP (fast **and** flexible).

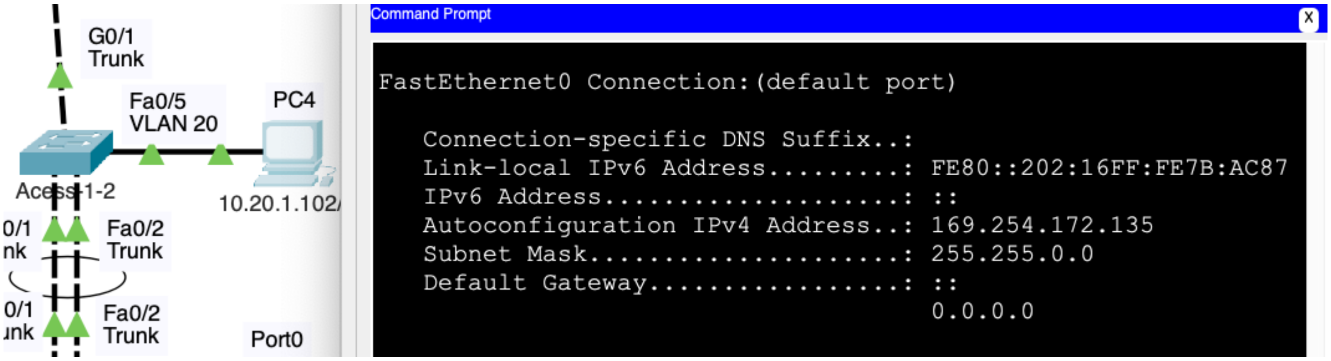
PortFast and BPDU Guard

Next, we will configure **PortFast** and **BPDU Guard** on switch ports that have client devices using DHCPv4. Both Cisco PVST+ and RSTP support these features.

Without **PortFast**, a switch port goes through the standard Spanning Tree Protocol (STP) states—*listening*, *learning*, and finally *forwarding*—which can take up to 30 seconds. During this time, the port does not forward traffic, including DHCP Discover messages sent by the client.



As a result, the client is unable to contact the DHCP server and assign itself a proper IP address. Instead, it will fall back to an **APIPA address** in the 169.254.x.x range, indicating it failed to obtain a network address through DHCP.



```
FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.........: FE80::202:16FF:FE7B:AC87
   IPv6 Address....................: ::
   Autoconfiguration IPv4 Address..: 169.254.172.135
   Subnet Mask.....................: 255.255.0.0
   Default Gateway.................: ::
                                     0.0.0.0
```

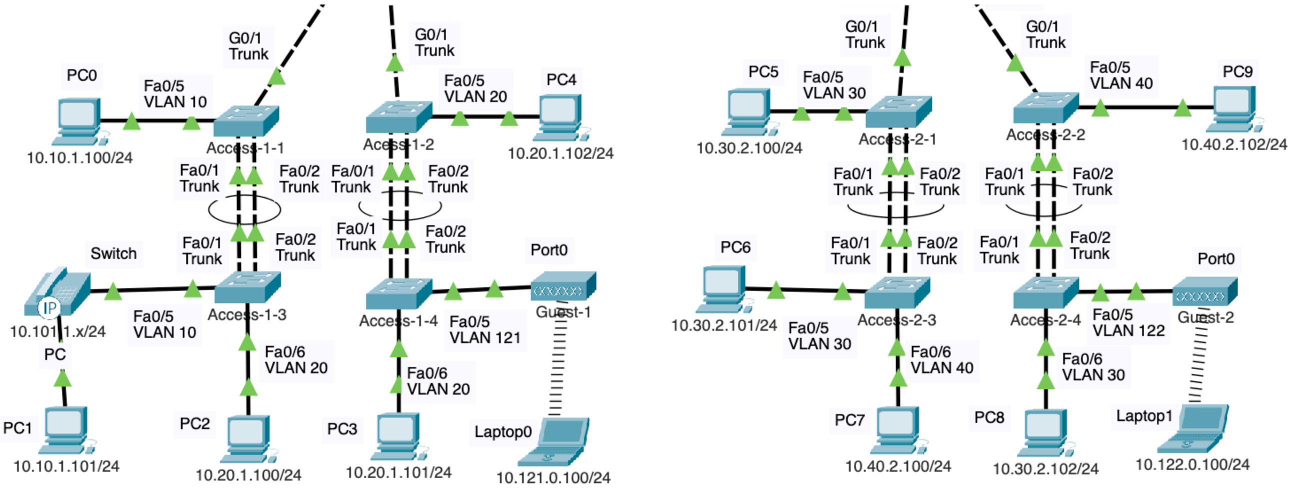
Depending on the client OS, it is possible that the device will try again or it may never receive IPv4 addressing information from the DHCPv4 server.

Using **portfast** is crucial in DHCP environments because it ensures that the switch port immediately transitions to the forwarding state, bypassing the usual Spanning Tree Protocol (STP) listening and learning phases. Without **portfast**, a device connected to the port might time out in its attempt to obtain an IP address via DHCP, as the port would not be in the forwarding state yet, delaying the transmission of DHCP requests and responses. By enabling **portfast**, we minimize the risk of DHCP timeouts, allowing devices to receive their IP addressing promptly as the port starts forwarding packets right away.

BPDU Guard is commonly used in conjunction with **spanning-tree portfast** to prevent accidental network disruptions that could occur if a switch is connected to a port configured for end devices like computers or IP phones.

When **portfast** is enabled, it allows the port to skip the usual STP deliberation phases and immediately start forwarding packets, which is great for quick network access but risky if a switch is mistakenly plugged into such a port. This could lead to changes in the spanning tree topology or even create network loops, potentially causing widespread network outages. **BPDU Guard** mitigates this risk by automatically shutting down the port if it receives a BPDU, indicating that a switch or bridge has been connected, thereby preserving the network's stability and preventing loop conditions.

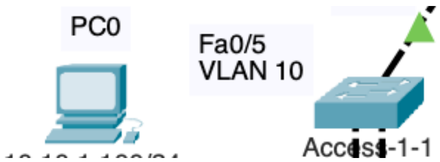
Configure the DHCP Client Switchports for PortFast and BPDU Guard

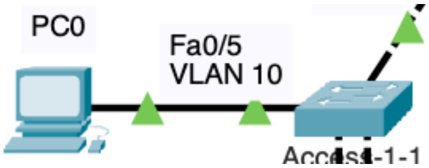


Enable **PortFast** on the switchports to allow the port to transition immediately to the forwarding state. Also, enable **BPDU Guard** to ensure that if a switch is attached to this port, the port will be disabled. Notice the warning message regarding attaching only a "single host."

```
access-1-1(config)# interface fa 0/5
access-1-1(config-if)# spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface  when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION

%Portfast has been configured on FastEthernet0/5 but will only
have effect when the interface is in a non-trunking mode.
access-1-1(config-if)# spanning-tree bpduguard enable
access-1-1(config-if)#
```

Next, remove the link between PC0 and Access-1-1 switch, and attach the same cable a second time. Notice that the port immediately transitions to forwarding state (green link light) as soon as the cable is plugged into the switchport.



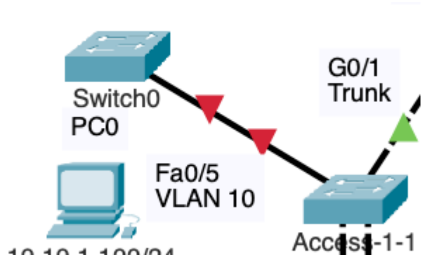


Add these same commands to all other access layer switches with end devices (PCs) attached, regardless if they are using DHCPv4 or statically configured IPv4 addresses.

You can use the interface range command to specify multiple ports at the same time. Here is a complete list of commands.

```
access-1-1(config)# interface fa 0/5
access-1-1(config-if)# spanning-tree portfast
access-1-1(config-if)# spanning-tree bpduguard enable

access-1-2(config)# interface fa 0/5
access-1-2(config-if)# spanning-tree portfast
access-1-2(config-if)# spanning-tree bpduguard enable

access-1-3(config)# interface range fa 0/5, fa 0/6
access-1-3(config-if)# spanning-tree portfast
access-1-3(config-if)# spanning-tree bpduguard enable

access-1-4(config)# interface range fa 0/5, fa 0/6
access-1-4(config-if)# spanning-tree portfast
access-1-4(config-if)# spanning-tree bpduguard enable

access-2-1(config)# interface fa 0/5
access-2-1(config-if)# spanning-tree portfast
access-2-1(config-if)# spanning-tree bpduguard enable

access-2-2(config)# interface fa 0/5
access-2-2(config-if)# spanning-tree portfast
access-2-2(config-if)# spanning-tree bpduguard enable

access-2-3(config)# interface range fa 0/5, fa 0/6
access-2-3(config-if)# spanning-tree portfast
access-2-3(config-if)# spanning-tree bpduguard enable

access-2-4(config)# interface range fa 0/5, fa 0/6
access-2-4(config-if)# spanning-tree portfast
access-2-4(config-if)# spanning-tree bpduguard enable
```

Testing BPDU Guard



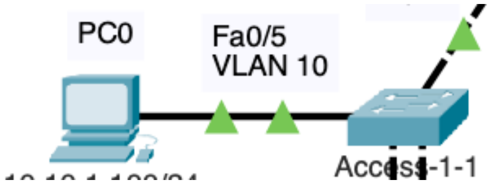
In this scenario, we removed the original direct connection between **PC0** and the access switch port **Fa0/5**, which had **PortFast** and **BPDU Guard** enabled. Instead of a PC, we connected a **Layer 2 switch** to this port. Because PortFast allows the port to immediately transition to the forwarding state, it assumes only a single end device (like a PC) is connected. However, the moment the newly connected switch sends a **Bridge Protocol Data Unit (BPDU)**, **BPDU Guard** is triggered. This is a safety mechanism that automatically puts the port into an **err-disabled** (error-disabled) state to protect the network from potential loops or topology changes caused by unauthorized or unintended switch connections.

```
access-1-1(config-if)#
%LINK-3-UPDOWN: Interface FastEthernet0/5, changed state to down

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

access-1-1(config-if)#%SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port
FastEthernet0/5 with BPDU Guard enabled. Disabling port.

%PM-4-ERR_DISABLE: bpduguard error detected on 0/5, putting 0/5 in err-disable
state

%LINK-5-CHANGED: Interface FastEthernet0/5, changed state to administratively down

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

access-1-1(config-if)#
```

This state can be verified using the **show interface status** command.

```
access-1-1# show interface status
Port      Name               Status       Vlan       Duplex  Speed Type
Po1                          connected    trunk      auto    auto
Fa0/1                        connected    trunk      auto    auto  10/100BaseTX
Fa0/2                        connected    trunk      auto    auto  10/100BaseTX
Fa0/3                        notconnect   1          auto    auto  10/100BaseTX
Fa0/4                        notconnect   1          auto    auto  10/100BaseTX
Fa0/5                        err-disabled 10         auto    auto  10/100BaseTX
Fa0/6                        notconnect   1          auto    auto  10/100BaseTX
<output omitted>
```

To fix the error-disabled port caused by BPDU Guard, first **remove the switch** and **reconnect the original PC** to the port. Then, on the switch, issue a shutdown followed by a no shutdown command on **interface Fa0/5** to reset the port and bring it back into service.



```
access-1-1(config)# interface fa 0/5
access-1-1(config-if)# shutdown

%LINK-5-CHANGED: Interface FastEthernet0/5, changed state to administratively down
access-1-1(config-if)# no shutdown

access-1-1(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/5, changed state to up

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

access-1-1(config-if)#
```

Switchports with IP Phones

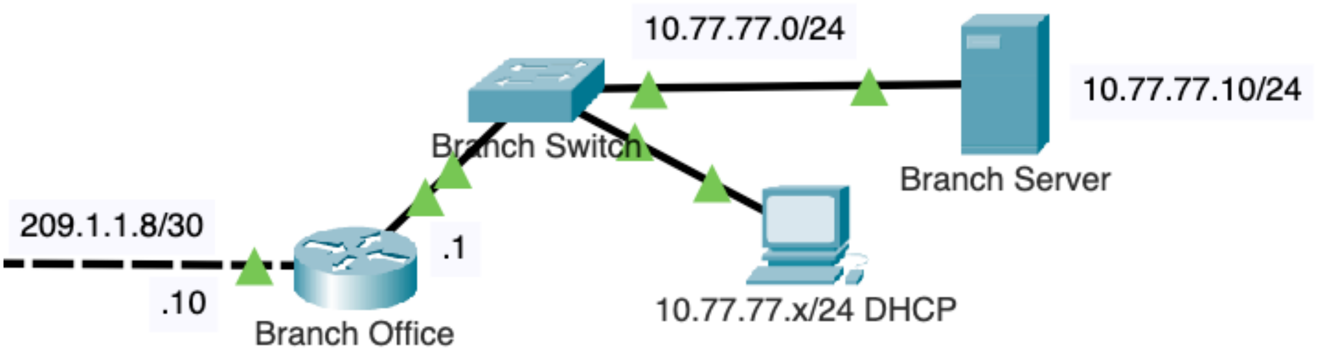
Since an IP Phone has a built-in switch, is it safe to configure PortFast on the access switch? Yes, **spanning-tree portfast** and **spanning-tree bpduguard** can be enabled on switchports that have IP phones connected.

Using **spanning-tree portfast** on a port connected to an IP phone with a built-in switch is generally considered safe because the switch in the IP phone is typically a simple Layer 2 device that does not participate in the Spanning Tree Protocol (STP). These built-in switches are designed to handle direct traffic between the phone and the connected PC and do not propagate BPDU packets that would participate in the STP process of the broader network.

As a result, the risk of creating a network loop through this connection is minimal. Additionally, enabling **spanning-tree bpduguard** on the port provides an extra layer of protection by ensuring that the port will be automatically disabled if it ever receives a BPDU, further mitigating the risk of accidental network topology changes.

```
access-1-3# show running-config | begin interface FastEthernet0/5
interface FastEthernet0/5
 switchport access vlan 10
 switchport mode access
 switchport voice vlan 101
 spanning-tree portfast
 spanning-tree bpduguard enable
 mls qos trust cos
!
```

Cisco Router as a DHCPv4 Server



In addition to routing traffic between networks, a **Cisco router can also function as a DHCPv4 server**. This is especially useful in small or remote office environments, like the **Branch Office** shown here, where deploying a separate DHCP server may not be practical. In this section, we will configure the Branch Office router to dynamically assign IPv4 addresses to devices on the **10.77.77.0/24** network, allowing client devices such as PCs to automatically receive their IP configuration.

```
Branch-Office(config)# ip dhcp excluded-address 10.77.77.1 10.77.77.50
Branch-Office(config)# ip dhcp pool DHCP-BRANCH-POOL
Branch-Office(dhcp-config)# network 10.77.77.0 255.255.255.0
Branch-Office(dhcp-config)# default-router 10.77.77.1
Branch-Office(dhcp-config)# dns-server 10.77.77.10
Branch-Office(dhcp-config)# end
Branch-Office#
```

The following commands configure the Branch Office router to act as a DHCP server by reserving static addresses, defining a DHCP pool, and specifying key settings such as the network, default gateway, and DNS server for client devices.

- **ip dhcp excluded-address 10.77.77.1 10.77.77.50** – Reserves addresses in this range so they are not assigned by DHCP (e.g., for routers, servers).
- **ip dhcp pool DHCP-BRANCH-POOL** – Creates and names a new DHCP address pool for configuration.
- **network 10.77.77.0 255.255.255.0** – Specifies the subnet for which the router will assign addresses.
- **default-router 10.77.77.1** – Sets the default gateway clients will use.
- **dns-server 10.77.77.10** – Assigns a DNS server address for client name resolution.
- **end** – Exits DHCP pool configuration mode and returns to privileged EXEC mode.

Configure the PC (10.77.77.x/24 DHCP) for DHCPv4:

- Select the **Config** tab
- Select **Interface> FastEthernet0**
  - Select **DHCP**

Verify the client PC has received its IPv4 addressing information using DHCP:

```
C:\>ipconfig /all

FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..: 
   Physical Address................: 00D0.BC20.77C7
   Link-local IPv6 Address.........: FE80::2D0:BCFF:FE20:77C7
   IPv6 Address....................: ::
   IPv4 Address....................: 10.77.77.51
   Subnet Mask.....................: 255.255.255.0
   Default Gateway.................: ::
                                     10.77.77.1
   DHCP Servers....................: 10.77.77.1
   DHCPv6 IAID.....................: 
   DHCPv6 Client DUID..............: 00-01-00-01-BD-39-D8-0D-00-D0-BC-20-77-C7
   DNS Servers.....................: ::
                                     10.77.77.10
```