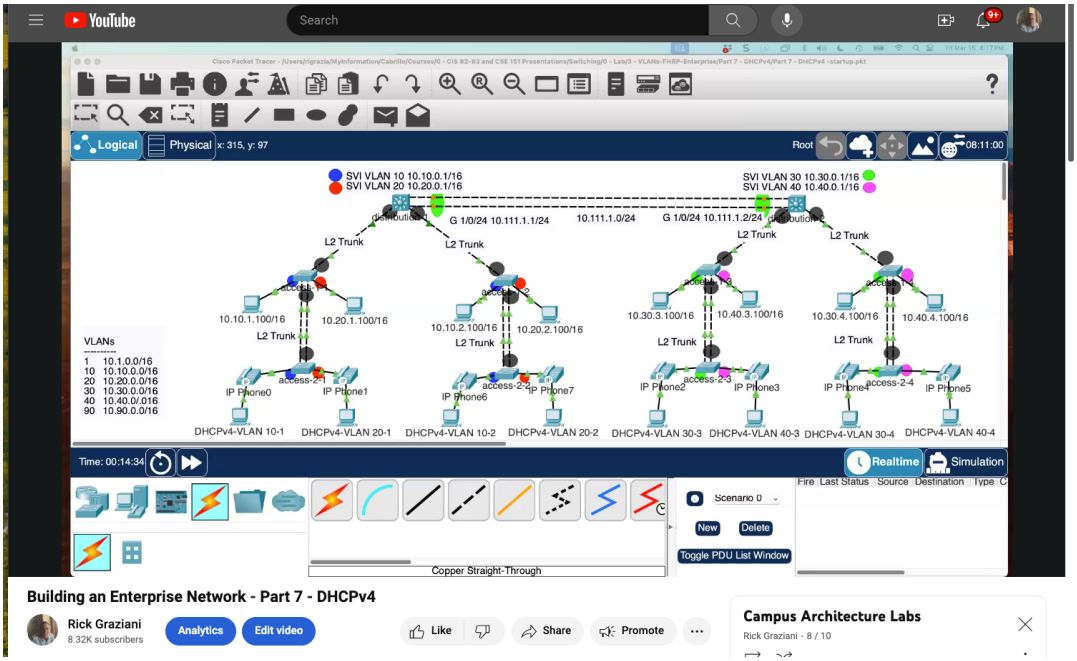
Building an Enterprise Network
Part 7: DHCPv4

Refer to the following YouTube Video:
https://www.youtube.com/watch?v=bN86F-hUQw8&list=PLMLm7-g0V0kdkzbfJS-naBrLMQNDubT8p&index=9



Topology from Previous Lab

Begin by using the topology you completed in the previous lab. We currently have configured VLANs, trunking, routed ports, routing and EtherChannel L2/L3 in our switch block, so we have complete reachability between all device.

DHCPv4

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.

Preparing DHCPv4 Clients

For the PCs connected to the access layer switches on layer 2, change the configuration for DHCPv4:

- Select the **Config** tab
- Either **Global > Settings** or **Interface> FastEthernet0** (Packet Tracer will change it for the other)
    - Select **DHCP**

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.

Adding the DHCPv4 Server

First add a DHCPv4 server to the 10.40.0.0/16 network, VLAN 40.

Note: Best practice may be to have a separate IP network/VLAN for servers, and in many cases be a separate switch block. But for demonstration purposes we will add the DHCPv4 server to one of our user VLANs.

Add the DHCPv4 server to the access switch in column 4, level 2 with the following IPv4 addressing information:

- IPv4 address: 10.40.0.99/16
- Default gateway: 10.40.0.1

Configure the DHCPv4 server:

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

Edit existing serverPool

Packet Tracer automatically created a 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 **Start IP Address** to: 10.255.0.0
2. Change the **Maximum Number of Users** to: 0
3. Select: **Save**

Adding pools for user IPv4 networks (VLANs)

Make the following changes:

1. **Pool Name**: serverPool-VLAN10
2. **Default Gateway**: 10.10.0.1
3. **DNS Server**: 10.88.88.88 (This server does not exist)
4. **Start IP Address**: 10.10.99.1
5. **Subnet Mask**: 255.255.0.0
6. **Maximum Number of Users**: 100
7. Select **Add** (If you click "Save", you will overwrite the previous entry.)

Repeat this process for the 10.20.0.0/16, 10.30.0.0/16 and 10.40.0.0/16 networks. You will only need to modify the Pool Name, Default Gateway and Start IP Address for each.

When completed, your DHCPv4 server pool table should look as follows:

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool-VLAN10	10.10.0.1	10.88.88.88	10.10.99.1	255.255.0.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN20	10.20.0.1	10.88.88.88	10.20.99.1	255.255.0.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN30	10.30.0.1	10.88.88.88	10.30.99.1	255.255.0.0	100	0.0.0.0	0.0.0.0
serverPool-VLAN40	10.40.0.1	10.88.88.88	10.40.99.1	255.255.0.0	100	0.0.0.0	0.0.0.0
serverPool	0.0.0.0	0.0.0.0	10.255.0.0	255.255.0.0	0	0.0.0.0	0.0.0.0

Enable DHCPv4 Service

Enable the DHCP service:

- **Service**: On

Connecting the DHCPv4 Server to the Network

Physically Connect the Server to the Switch

Connect the server to access layer switch in column 4, level 2 using GigabitEthernet 0/1 port on the switch.

Note: When using the section filter, case matters.

```
access-2-4# show running-config | section interface GigabitEthernet0/1
interface GigabitEthernet0/1
 switchport access vlan 255
 switchport mode access
 shutdown
access-2-4#
```

Configuring the Switchport Configuration

The switchport is currently in the parking lot VLAN 255 and shutdown. Configure the port for VLAN 40 and enable the port.

Note: Although not necessary, I prefer to explicitly include commands even though they are part of my default configuration, and remove commands even though IOS will overwrite it.

```
access-2-4(config)# interface g 0/1
access-2-4(config-if)# no switchport access vlan 255
access-2-4(config-if)# switchport mode access
access-2-4(config-if)# switchport access vlan 40
access-2-4(config-if)# no shutdown

%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state
to up
```

Verify IPv4 Address via DHCP on 10.40.0.0/16

On the PCs connected to VLAN 40 and enabled for DHCPv4 (level 2 switches on column 2 and 3) verify the PCs 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.

```
Cisco Packet Tracer PC Command Line 1.0
C:\> ipconfig

FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.....: FE80::260:70FF:FE6B:E32D
   IPv6 Address.....: ::
   Autoconfiguration IPv4 Address..: 169.254.227.48
   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

   IP Address......................: 10.40.99.1
   Subnet Mask.....................: 255.255.0.0
   Default Gateway.................: 10.40.0.1
   DNS Server......................: 10.88.88.88

C:\>
```

DHCPv4 Relay

Verify DHCPv4 Unsuccessful on other Networks/VLANs

If you examine the PCs on other IP networks/VLANs they are still not obtaining their IPv4 addressing from the DHCPv4 server. This 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 PC a VLAN 30:

```
C:\> ipconfig

FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.........: FE80::230:F2FF:FEB6:EA58
   IPv6 Address....................: ::
   Autoconfiguration IPv4 Address..: 169.254.234.88
   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:\>
```

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 subnetted 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 command

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

The Layer 3 switch Distribution-2 is the default gateway for device on 10.30.0.0/16 (VLAN 30) network. The SVI, interface VLAN 30, has previously been configured to act as the default gateway for devices in this network.

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

```
distribution-2# show running-config | section interface Vlan30
interface Vlan30
 mac-address 000c.8533.1201
 ip address 10.30.0.1 255.255.0.0
distribution-2#

distribution-2(config)# interface vlan 30
distribution-2(config-if)# ip helper-address 10.40.0.99
```

```
distribution-2(config-if)#
```

Note: Notice that we did not have to use the **ip helper-address** command for devices on VLAN 40 because they are on the same broadcast domain as the DHCPv4 server.

DHCPv4 Relay IPv4 Addressing Details

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 10.30.0.0/16 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.
2. **Router to DHCP Server**:
    - The router, configured with **ip helper-address** 10.40.0.99, receives the broadcast on its interface with IPv4 address 10.30.0.1.
    - The router forwards this packet as a unicast packet with the source IPv4 address 10.30.0.1 (the interface it received the broadcast on) and the destination IPv4 address 10.40.0.99 (the DHCP server).
3. **DHCP Server to Router**:
    - The DHCP server receives the Discover message and responds with a DHCP Offer message.
    - This Offer is sent as a unicast packet with the source IPv4 address 10.40.0.99 and the destination IPv4 address 10.30.0.1 (back to the router's interface that sent the Discover message).
4. **Router to Client**:
    - Upon receiving the DHCP Offer, the router broadcasts it to the 10.30.0.0/16 network.
    - The source IP address remains 10.40.0.99 (DHCP server's IP), but the router broadcasts it on the local network, so the packet uses the local network broadcast address (e.g., 10.30.255.255) as the destination to ensure the client receives it.

Verify DHCPv4 and 10.30.0.0/16 Clients

On the PCs connected to VLAN 30, 10.30.0.0/16 network and enabled for DHCPv4 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.

A PC on VLAN 30:

```
C:\> ipconfig

FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.....: FE80::20C:85FF:FE19:8754
   IPv6 Address.....: ::
   IPv4 Address.....: 10.30.99.1
   Subnet Mask.....: 255.255.0.0
   Default Gateway.....: ::
                                 10.30.0.1
```

Configuring ip helper-address on Distribution-1 Switch

Devices on 10.10.0.0/16 (VLAN 10) and 10.20.0.0/16 (VLAN 20) use distribution-1 layer 3 switch as their default gateway.

So, we will use a similar **ip helper-address** command for 10.40.0.99 on distribution-1 switch. We will configure this command on the interfaces the clients use as the default gateways, the SVIs for VLAN 10 and VLAN 20.

```
distribution-1(config)# interface vlan 10
distribution-1(config-if)# ip helper-address 10.40.0.99
distribution-1(config-if)# exit
distribution-1(config)# interface vlan 20
distribution-1(config-if)# ip helper-address 10.40.0.99
distribution-1(config-if)#
```

Spanning Tree PortFast and BPDU Guard

Reasons for PortFast and DHCP

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.

Add the PC to the Switch

Add a PC to column 1, level 1 switch. Configure the PC to obtain its addressing using DHCP.

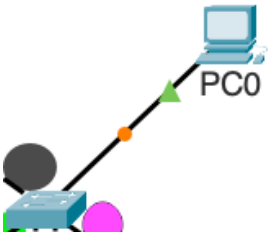
Configure the Switchport without PortFast and Attach the PC

Configure the column 1, level 1 switch as a VLAN 40 access port. Since there is no cable connected (or if the connected device is powered off), the switch port will not detect a carrier signal and will therefore be in a "down" state.

```
access-1-4(config)# interface fa 0/21
access-1-4(config-if)# no switchport access vlan 255
access-1-4(config-if)# switchport mode access
access-1-4(config-if)# switchport access vlan 40
access-1-4(config-if)# no shutdown

%LINK-5-CHANGED: Interface FastEthernet0/21, changed state to down
access-1-4(config-if)#
```

Now attach the PC to fa 0/21. Notice that the switchport is amber until about 50 seconds later when Spanning Tree transitions the port to forwarding state (green link light).



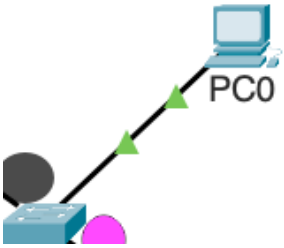
As described previously, depending on the client OS, it is possible that the device will never receive IPv4 addressing information from the DHCPv4 server.

Enable **PortFast** on this switchport 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-1-4(config)# interface fa 0/21
access-1-4(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 FastEthernet0/21 but will only
have effect when the interface is in a non-trunking mode.
access-1-4(config-if)# spanning-tree bpduguard enable
access-1-4(config-if)#
```

Next, remove the link 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.



Cleaning up: You can remove the PC that we just added and put that port back to the parking-lot VLAN and shutdown.

```
access-1-4(config)# interface fa 0/21
access-1-4(config-if)# no switchport access vlan 40
access-1-4(config-if)# switchport mode access
access-1-4(config-if)# switchport access vlan 255
access-1-4(config-if)# shutdown

%LINK-5-CHANGED: Interface FastEthernet0/21, changed state to administratively
down
access-1-4(config-if)#
```

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

Note: Since all 8 access layer use the same switchports that have IP phones connected, you can use the interface range command to specific both ports at once.

```
access-x-y(config)# inter range fa 0/10, fa 0/20
access-x-y(config-if)# spanning-tree portfast
access-x-y(config-if)# spanning-tree bpduguard enable
```

Switchports with IP Phones

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-2-4# show running-config | section interface GigabitEthernet0/1
interface GigabitEthernet0/1
 switchport access vlan 255
 switchport mode access
 shutdown
access-2-4#
```