CCNA Exam Prep: Back to Networking Basics with Hank Preston and Patrick Gargano

Mastering VLAN Configuration

Exploring why VLANs exist in network engineering and how to configure them on Cisco IOS switches.

- Creating VLANs
- Configuring Access Ports
- Configuring 802.1q trunk interfaces
- Inter VLAN Routing options

Setup and Scenario

In this set of lab-based demonstrations, you are the network engineer for a growing organization tasked with updating the network to support new network needs. The network traditionally has had 2 small network segments. One for "Employees" and a second one for "Guests".

You are getting two requests from the business:

1. The facilities team is asking for network connectivity for IOT Devices. These devices need to be segmented from employee and guest traffic.
2. The organization is growing, and there is need to add network capacity to support additional users. These new users will be located in new office space being prepared in a different part of the building.

The following lab-based demonstrations will look at how VLANs provide solutions for these requests.

> Note: In a real network, there would be security included to control traffic to and from these segments through the use of Access Control Lists (ACLs) and/or firewalls appliances. The security configuration is out of scope for this lab that focuses solely on VLANs.

*Be sure to **START** the lab before continuing to the demo labs.*

Demo 1: A Network Without VLANs

Before we dive into creating and managing VLANs, let's remind ourselves what life would be like without them.

Our starting network is composed of 2 simple unmanaged switches. `EmpSw` for employees, and `GstSw` for guests. There is also an IOS router called `Router` that provides routing between network segments.

Goal: To support the new IOT devices, we will be adding a new `IOTSw` to the network.

> Notes:
>
> - `Router` interface Ethernet0/3 is already configured for the new "IOT Devices" network segment.
> - `IOTHost` is already configured with an IP address on the "IOT Devices` network segment.

Testing current state

1. Open the console for `EmpHost` (credentials are `cisco / cisco`).
2. Ping `GstHost` with command `ping 192.168.2.11`.
    - Verify this is successful - `0% packet loss`
    - End the ping by pressing `Cntrl-C`
3. Attempt to ping `IOTHost` with command `ping 192.168.3.11`.
    - This should be unsuccessful - `100% packet loss`

Adding a switch/network for IOT devices

1. Add a new "Unmanaged Switch" to the "IOT Devices" network block within "Demo 1".
2. Name this new switch `IOTSw`.
3. As this is an "unmanaged switch", there isn't any configuration needed.
4. **Start** `IOTSw`. (Right click the new switch)
5. Add 2 links to the new `IOTSw`.
    1. From `Router` interface `Eth0/3` to `port0`
    2. From `IOTHost` interface `eth0` to `port1`

Testing the network

1. Return to the console for `EmpHost`

2. Retry pinging `IOTHost` with command `ping 192.168.3.11`.
    - Verify this is ***now*** successful - `0% packet loss`
    - If it is ***NOT*** successful, ensure you connected interface `eth0/3` from the router.

Conclusion

We've been able to support the new network segment by adding an additional switch, however, consider what happens as the number of network segments increase from 3 to 10, 20, or 100. The number of physical switches required will quickly become challenging. Also worth considering is the router requirement to support this number of switches.

Demo 2: Configuring VLANs and Access Ports

In this demonstration we are looking at how we can replace multiple "unmanged switches" with a single "managed switch" and maintain the segmentation with VLANs.

The starting network already has an IOS based "switch" named `Switch` with links to each of the hosts in the network. There are the following hosts:

- `EmpHost01` - an employee host
- `GstHost01` - a guest host
- `IOTHost01` & `IOTHost02` - 2 IOT host devices

In this demonstration we are looking at how VLANs enforce layer 2 network segmentation. To explore that, each of the hosts have been assigned an IP address on the same `10.0.0.0/24` network. We will use these addresses to test and verify the segmentation.

Goal: To create 3 VLANs and configure each host in the proper network segment.

Testing starting state

As we start, the only VLAN that exists on the switch is `Vlan 1`, the "default vlan". All hosts are on this vlan, and therefore should be able to communicate with each other.

1. Open the console for `Switch` and look at the starting VLAN configuration with the command `show vlan brief`.

    In addition to `Vlan 1`, you will see vlans `1002 - 1005`. These are special purpose VLAN numbers used for older non-ethernet based networks and out of scope for this lab and the CCNA.

2. On IOS switches, the configuration of "normal range VLANs" (numbers `1 - 1001`) is not stored in the configuration file, but in a special `vlan.dat` file. Because we don't have any VLANs created yet, this file does NOT exist. You can verify with the command `dir`. You should NOT see the `vlan.dat` file.
3. Open the console for `IOTHost01`. Attempt to ping each of the other hosts.
    - `ping 10.0.0.1`
    - `ping 10.0.0.2`
    - `ping 10.0.0.4`
4. All of these should work. This is because each host is on the same VLAN. Other ways to say this include:
    - All hosts are on the same network segment
    - All hosts are on the same broadcast domain

Creating VLANs

Now we will create three VLANs:

VLAN ID	VLAN Name
10	Employees
20	Guests
30	IOT-Devices

1. Open the console for `Switch`.
2. Enter privileged exec mode (ie enable mode) and then configuration mode with the commands `enable` and `config t`.
3. Create the first vlan with the following commands:

```
vlan 10
     name Employees
     exit
```

4. Repeat for the other two VLANs, changing the numbers and names.
5. Leave configuration mode.

6. Verify that the VLANs were created with the `show vlan brief` command.

7. Verify that the `vlan.dat` file was created with the `dir` command.

    Note: The file may be named something like `vlan.dat-00003`. The addition of the number is due to the simulated nature of the switch used in CML.

Moving hosts into the correct VLANs

We have created the VLANs, but all our hosts are still in VLAN 1. To move the hosts into the correct VLANs, we need to change the `access vlan` for the related interfaces on `Switch` for each host.

Host	Interface	VLAN ID
EmpHost01	Eth0/1	10
GstHost01	Eth0/2	20
IOTHost01	Eth0/3	30
IOTHost02	Eth1/0	30

1. Open the console for `Switch`.
2. Look at the current `switchport` configuration for interface `Eth0/1` - the interface connected to `EmpHost01`. Use the command `show interface eth0/1 switchport`.
    - Look for the lines `Operational Mode: static access` and `Access Mode VLAN: 1`
    - These show that hosts "accessing" the network through this port are assigned to VLAN 1.

3. Move to configuration mode and change the "access vlan" for port `EmpHost01` to VLAN 10 with the following commands:

```
interface ethernet0/1
     switchport mode access
     switchport access vlan 10
     exit
```

    - `switchport mode access` - The default mode is `dynamic auto`, as could be seen in the output from `show interface switchport` above. This mode works for hosts, but results in the switch attempting to negotiate a "vlan trunk" with a connected device. For ports that are known to connect to a single hosts, `mode access` is preferred.
    - `switchport access vlan 10` - This changes the VLAN that the interface will use when operating in mode `static access`

4. Repeat the configuration steps for the other three hosts. Pay special attention to the VLAN ID used for each interface, and notice that both IOTHosts are in the same VLAN.

5. Exit configuration mode.
6. Verify the updated VLAN configuration for each interface with the command `show vlan brief`. Compare the output to the table above.
7. Use the command `show interface ethernet 0/1 switchport` to check the current interface switchport configuration. Compare it to the output from before the changes.

Testing the resulting network segmentation

Now that we've segmented our network into three VLANs, let's see the impact to connectivity.

1. Open the console for `IOTHost01`.
2. Attempt to ping `EmpHost01` with the command `ping 10.0.0.1`.
    - This should be unsuccessful - `100% packet loss`
3. Attempt to ping `GstHost01` with the command `ping 10.0.0.2`.
    - This should be unsuccessful - `100% packet loss`
4. Attempt to ping `IOTHost02` with the command `ping 10.0.0.4`.
    - This should be ***successful*** - `0% packet loss`

The first 2 pings fail because the `IOTHost01`, `EmpHost01`, and `GstHost01` are all in *different* VLANs. And the last ping works because the two IOTHosts are in the **same** VLAN.

Conclusion

We've now been able to take a single switch and use VLANs to create three "virtual switches" (ie Virtual LANs).

Demo 3: Multi-Switch Networks and VLAN Trunks

Now that we have created VLANs, we are going to look at how we can handle capacity in a network by adding additional switches to our VLAN segmented network. Reasons for an additional switch may be needed include:

- When the existing switch at a location runs out of available ports
- When adding a new location (such as a new building or floor in a building)

In the starting network there are two switches, each with a host in each of the VLANs. The hosts have been segmented into VLANs 10, 20, and 30 already. The switches have been linked together on ports `Eth1/1`, `Eth1/2`, and `Eth1/3` - one for each of the network segments.

In this demonstration, we will first configure these "inter-switch links" as access ports, just like we did in Demo 2 for the host systems. Then we'll explore the better option for connecting switches together, VLAN trunks.

Goal: To create network segments that span multiple switches by using VLAN trunks.

Testing starting state

Before we begin connecting the VLANs across switches together, let's verify that hosts on different switches can NOT communicate.

1. Open the console for `EmpHost11` that is connected to `Switch01`.
2. Attempt to ping `EmpHost12` with the command `ping 192.168.1.12`.
    - This should be unsuccessful.
3. Repeat this test with `GstHost11` and `IOTHost11` by trying to ping the respective `Host02` on `Switch02`.
    - These should both be unsuccessful.

Configuring inter-switch links as access ports

Similar to how in Demo 1 we started by adding a new *dedicated* switch for each segment and then explored using VLANs and a ***single*** switch, we will first see how we can use *dedicated interfaces* for each segment.

Purpose	Interface	VLAN ID
Employees	`Eth1/1`	`10`
Guests	`Eth1/2`	`20`
IOT Devices	`Eth1/3`	`30`

Note: The same interface is used on both switches

1. Open the console on `Switch01`.
2. Check the current VLAN configuration for interfaces `Ethernet1/1 - 3` with the command `show vlan brief`.
    - All three interfaces should be listed as ports for VLAN 1.
3. Move into configuration mode and configure the access VLAN for each interface as shown in the table.
    - Use the same commands from Demo 2.
4. Exit configuration mode.
5. Use `show vlan brief` to verify the changes were made successfully.
6. Open the console on `Switch02` and repeat the configuration and verification.

Testing inter-switch connectivity

With the inter-switch links configured, we can now test that communications are successful.

1. Open the console for `EmpHost11` that is connected to `Switch01`.
2. Attempt to ping `EmpHost12` with the command `ping 192.168.1.12`.
    - This should now be ***successful***.
3. Repeat this test with `GstHost11` and `IOTHost11` by trying to ping the respective `Host12` on `Switch02`.
    - These should now both be ***successful***.

Improving inter-switch links with a single VLAN Trunk

Just like it doesn't scale to add a new unmanaged switch for each new network segment, it doesn't make sense to need separate interfaces for each VLAN when connecting two switches together. Instead, network engineers use VLAN trunks, single interfaces that can service multiple VLANs while maintaining network isolation and segmentation.

VLAN trunks are created with the 802.1q standard trunking protocol. This protocol adds a *VLAN tag* to each frame sent on the trunk interface. The tag includes the VLAN ID number. The receiving device can then read this tag to determine what VLAN (segment) the traffic belongs on.

Let's reconfigure our network to use a single 802.1q VLAN trunk interface instead of three separate links.

1. First we will `shutdown` interfaces `Ethernet1/2` and `Ethernet1/3` on both switches. ***On each switch:***

    1. Open the console

2. Disable `interface Ethernet1/2` with these commands:

```
interface ethernet1/2
     shutdown
     exit
```

3. Repeat for `interface Ethernet1/3` with the same commands.

2. Test connectivity for the hosts on the segments using `ping` commands.
   - `EmpHost11` should be able to successfully ping `EmpHost12` because we left `Ethernet1/1` enabled
   - `GstHost11` and `IOTHost11` should **not** be able to ping their peer hosts (Host12) because we shutdown the interfaces they used for communications.

3. Change the configuration on `Ethernet1/1` from `static access` to `static trunk`. ***On each switch:***

   1. Open the console

   2. Change the `mode` of interface `Ethernet1/1` with the following commands.

```
interface ethernet1/1
     switchport trunk encapsulation dot1q
     swithchport mode trunk
     exit
```

      Note: `dot1q` (802.1q) is an industry standard trunking protocol. `isl` is a Cisco trunking protocol that has been deprecated in favor of the standard 802.1q.

   3. Exit configuration mode

4. Verify the interface is now trunking. ***On each switch:***

   1. Check the trunking configuration with the command `show interface ethernet1/1 trunk`.
      - Notice the encapsulation and trunking state
      - Notice the "Vlans in spanning tree forwarding state and not pruned"
   2. Check the `switchport status` with the command `show interface ethernet1/1 switchport`.
      - Notice the values for `Administrative Mode`, `Operational Mode` and `Trunking Encapsulation`.

Testing inter-switch connectivity with the VLAN trunk

With the trunk configured, once more test the ability for each host to communicate with its peer on the other switch.

1. Open the console for `EmpHost11` that is connected to `Switch01`.
2. Attempt to ping `EmpHost12` with the command `ping 192.168.1.12`.
   - This should now be ***successful***.
3. Repeat this test with `GstHost11` and `IOTHost11` by trying to ping the respective `Host12` on `Switch02`.
   - These should now both be ***successful***.

Conclusion

Now that we've configured a VLAN trunk between the switches, hosts connected to new, future VLANS we create on our switches will be able to communicate without needing to add new links between our switches.

Demo 4: Inter VLAN Routing

Our last step to implementing VLANs in our network is to setup routing so that hosts on one VLAN can communicate with hosts on other VLANs.

There are two approaches to inter VLAN routing that we want to explore.

1. *Router-on-a-Stick*: This option leverages a dedicated router connected to the switches with a VLAN trunk. The router has ***subinterfaces*** for each VLAN and can forward traffic between the networks. This option is best for networks with switches that are "layer 2 only" and can't handle routing.
2. *Switched Virtual Interfaces (SVI)*: This option requires "layer 3 switches", sometimes known as "multi-layer switches". These switches have the ability to act as a router in addition to being a switch. You'll create "vlan interfaces" for each VLAN that will forward traffic between the networks.

   Note: Only one of the two options should be configured and enabled on a network at one time.

Goal: Configure routing between the networks using both Router on a Stick and SVIs.

Configuring Router-on-a-Stick

1. Locate `Router01` in the "Demo 4" box.
2. Add a link from `Router01` interface `Eth0/0` to `Switch02` interface `Eth0/0`

3. Now we'll configure the switch interface to be an 802.1q trunk, just like we did between switches.

    1. Open the console for `Switch02`.

    2. Enter configuration mode for `interface Eth0/0` and configure the interface as a trunk with the following commands.

    ```
    interface ethernet 0/0
         no shutdown
         switchport trunk encapsulation dot1q
         switchport mode trunk
         exit
    ```

4. Now we'll configure the "router on a stick".

    1. Open the console for `Router01`

    2. Enable the physical interface connected to the switch.

    ```
    interface ethernet 0/0
         no shutdown
         exit
    ```

        No IP address or other configuration is applied to the physical interface. We will be creating "subinterfaces" for each VLAN. But if the physical interface is `shutdown`, none of the subinterfaces will work.

    3. Create subinterface `ethernet0/0.10` for VLAN 10 with the following commands.

    ```
    interface ethernet0/0.10
         encapsulation dot1q 10
         ip address 192.168.1.1 255.255.255.0
         exit
    ```

        - Subinterfaces are created by adding `.NUMBER` to an interface name. The number used typically matches the VLAN the subinterface will be placed on, but it is *not* mandatory to match.
        - The command `encapsulation dot1q 10` associates this subinterface with VLAN 10

    4. Repeat the process to create subinterfaces `ethernet0/0.20` and `ethernet0/0.30` for VLANs 20 and 30 respectively using IP addresses `192.168.2.1` and `192.168.3.1`.

    5. You can verify the interface configuration on the router with the commands `show ip interface brief` and `show ip interface ethernet0/0.10`.

Testing Router-on-a-Stick

1. Open the console for `EmpHost11`.
2. Try to ping `GstHost12` with command `ping 192.168.2.12`
    - It should be successful.
3. Try to ping `IOTHost12` with command `ping 192.168.3.12`
    - It should be successful.
4. If the pings are NOT working, check the following:
    - Wait 60 seconds and try again. It takes time for spanning-tree to converge after configuring the subinterfaces and it may not be done yet.
    - Did you connect `Router01` to `Switch02`?
    - Did you connect interfaces `Ethernet0/0` on both devices?
    - Did you configure `Switch02` interface `Ethernet0/0` as a trunk?
    - Did you use the correct VLAN number on the `encapsulation dot1q` commands for the subinterfaces?
    - Did you use the correct IP addresses for each VLAN subinterface?

Great job! You've configured inter VLAN routing with router on a stick!

Disabling Router-on-a-Stick

Before we can configure Switched Virtual Interface (SVI) routing on the switches, we need to disconnect our "router on a stick". This could be done by removing the router configuration, shutting down the interfaces, or deleting the link. We will do this by simply removing

the link connecting the router to the switch.

1. Select the link connecting `Router01` to `Switch02`.
2. **Delete** the link.
3. Verify that you can ***no longer*** ping `GstHost02` from `EmpHost01`.
    - If you can still ping between networks, make sure you deleted the correct link.

Configuring Switched Virtual Interface (SVI) Routing

1. Open the console for `Switch01`.

2. Enter configuration mode, and enable the layer 3 features of the switch with the following command:

```
ip routing
```

    Note: Some switches might have this command enabled by default. But it doesn't hurt to get into the habit of configuring it to be sure.

3. Create `interface vlan 10` for the Employees network with the following commands:

```
interface vlan 10
     no shutdown
     ip address 192.168.1.1 255.255.255.0
     exit
```

4. Repeat the steps to create `interface vlan 20` and `interface vlan 30` for the guests and IOT devices networks respectively.

5. You can verify the interface configurations with the commands `show ip interface brief` and `show ip interface vlan 10`.

Testing Switched Virtual Interface (SVI) Routing

1. Open the console for `EmpHost11`.
2. Try to ping `GstHost12` with command `ping 192.168.2.12`
    - It should be successful.
3. Try to ping `IOTHost12` with command `ping 192.168.3.12`
    - It should be successful.
4. If the pings are NOT working, check the following:
    - Did you use the correct VLAN numbers when creating `interface vlan`?
    - Did you use the correct IP addresses for each VLAN interface?

Great job! You've configured inter VLAN routing with Switched Virtual Interfaces (SVIs)!