NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCE
Computer Network Lab (CL-307)
Lab Session 10

Awais Ahmed || Faizan Yousuf || Munim Ali
awais.ahmed@nu.edu.pk || faizan.yousuf@nu.edu.pk || munim.ali@nu.edu.pk

UNDERSTANDING & CONFIGURING VLANs AND INTER-VLANs

Overview of VLANs

A VLAN is a group of devices on one or more LANs that are configured to communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, they are extremely flexible. VLANs define broadcast domains in a Layer 2 network. A broadcast domain is the set of all devices that will receive broadcast frames originating from any device within the set. Broadcast domains are typically bounded by routers because routers do not forward broadcast frames. Layer 2 switches create broadcast domains based on the configuration of the switch. Switches are multiport bridges that allow you to create multiple broadcast domains. Each broadcast domain is like a distinct virtual bridge within a switch. You can define one or many virtual bridges within a switch. Each virtual bridge you create in the switch defines a new broadcast domain (VLAN). Traffic cannot pass directly to another VLAN (between broadcast domains) within the switch or between two switches. To interconnect two different VLANs, you must use routers or Layer 3 switches.

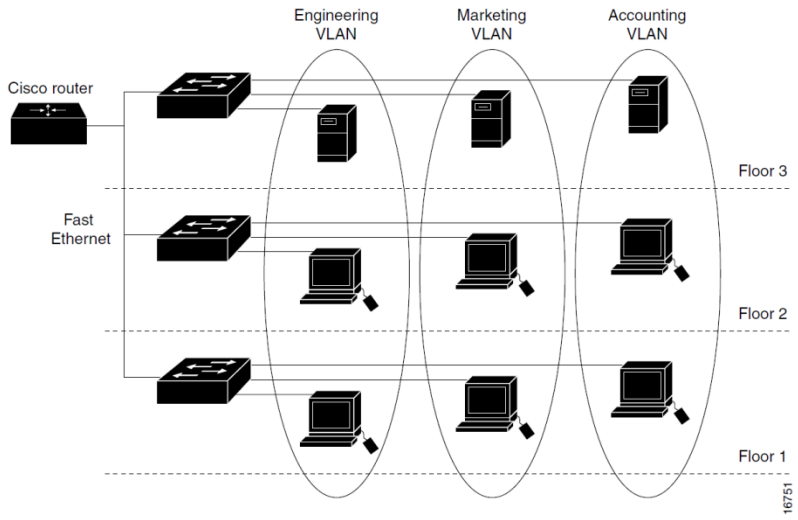


Figure 1 shows an example of three VLANs that create logically defined networks.

VLANs are often associated with IP sub-networks. For example, all of the end stations in a particular IP subnet belong to the same VLAN. Traffic between VLANs must be routed. You must assign LAN

interface VLAN membership on an interface-by-interface basis (this is known as interface-based or static VLAN membership).
You can set the following parameters when you create a VLAN in the management domain:

- VLAN number
- VLAN name
- VLAN type
- VLAN state (active or suspended)
- Maximum transmission unit (MTU) for the VLAN
- Security Association Identifier (SAID)
- VLAN number to use when translating from one VLAN type to another

VLANs work like this: Figure 2 shows all hosts in this very small company connected to one switch, meaning all hosts will receive all frames, which is the default behavior of all switches.

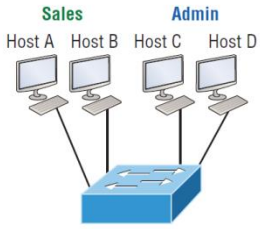


Figure 2 - One switch, one LAN: Before VLANs, there were no separations between hosts.

If we want to separate the host's data, we could either buy another switch or create virtual LANs, as shown in Figure 3.

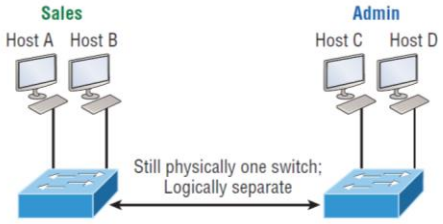


Figure 3 One switch, two virtual LANs (*logical* separation between hosts): Still physically one switch, but this switch acts as many separate devices.

In Figure 3, I configured the switch to be two separate LANs, two subnets, two broadcast domains, two VLANs—they all mean the same thing—without buying another switch. We can do this 1,000 times on most Cisco switches, which saves thousands of dollars and more!

Notice that even though the separation is virtual and the hosts are all still connected to the same switch, the LANs can't send data to each other by default. This is because they are still separate networks, but no worries—we'll get into inter-VLAN communication later.
Here's a short list of ways VLANs simplify network management:

- Network adds, moves, and changes are achieved with ease by just configuring a port into the appropriate VLAN.
- A group of users that need an unusually high level of security can be put into its own VLAN so that users outside of that VLAN can't communicate with it.
- As a logical grouping of users by function, VLANs can be considered independent from their physical or geographic locations.
- VLANs greatly enhance network security if implemented correctly.
- VLANs increase the number of broadcast domains while decreasing their size.

Coming up, we'll thoroughly explore the world of switching, and you learn exactly how and why switches provide us with much better network services than hubs can in our networks today.

Identifying VLANs

Switch ports are layer 2–only interfaces that are associated with a physical port that can belong to only one VLAN if it's an access port or all VLANs if it's a trunk port. Switches are definitely pretty busy devices. As myriad frames are switched throughout the network, switches have to be able to keep track of all of them, plus understand what to do with them depending on their associated hardware addresses. And remember—frames are handled differently according to the type of link they're traversing.

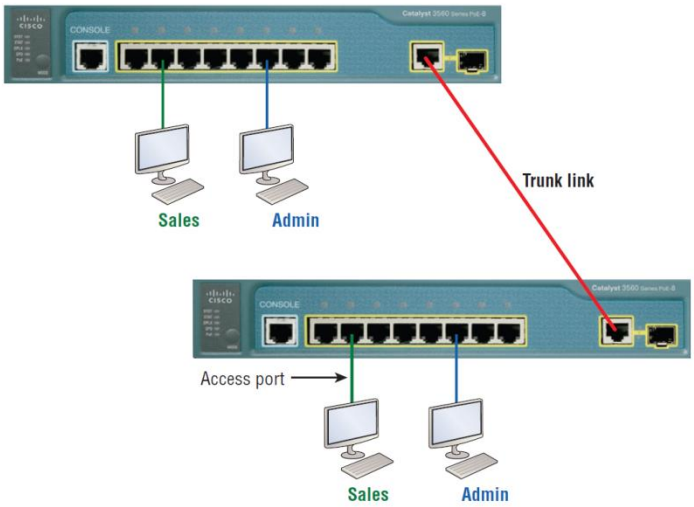
Access ports An access port belongs to and carries the traffic of only one VLAN. Traffic is both received and sent in native formats with no VLAN information (tagging) whatsoever. Anything arriving on an access port is simply assumed to belong to the VLAN assigned to the port. Because an access port doesn't look at the source address, tagged traffic—a frame with added VLAN information—can be correctly forwarded and received only on trunk ports. With an access link, this can be referred to as the configured VLAN of the port. Any device attached to an access link is unaware of a VLAN membership—the device just assumes it's part of some broadcast domain. But it doesn't have the big picture, so it doesn't understand the physical network topology at all.

Another good bit of information to know is that switches remove any VLAN information from the frame before it's forwarded out to an access-link device. Remember that access-link devices can't communicate with devices outside their VLAN unless the packet is routed. Also, you can only create a switch port to be either an access port or a trunk port—not both. So you've got to choose one or the other and know that if you make it an access port, that port can be assigned to one VLAN only. This is the same with Admin VLAN, and they can both communicate to hosts on the other switch because of an access link for each VLAN configured between switches.

Trunk ports Believe it or not, the term trunk port was inspired by the telephone system trunks, which carry multiple telephone conversations at a time. So it follows that trunk ports can similarly carry multiple VLANs at a time as well. A trunk link is a 100, 1,000, or 10,000 Mbps point-to-point link between two switches, between a switch and router, or even between a switch and server, and it carries the traffic of multiple VLANs—from 1 to 4,094 VLANs at a time. But the amount is really only up to 1,001 unless you're going with something called extended VLANs. Instead of an access link for each VLAN between switches, we'll create a trunk link demonstrated in Figure 4.
Trunking can be a real advantage because with it, you get to make a single port part of a whole bunch of different VLANs at the same time. This is a great feature because you can actually set ports up to have a server in two separate broadcast domains simultaneously so your users won't have to cross a layer 3 device (router) to log in and access it. Another benefit to trunking comes into play when you're

connecting switches. Trunk links can carry the frames of various VLANs across them, but by default, if the links between your switches aren't trunked, only information from the configured access VLAN will be switched across that link.

Figure 4 - VLANs can span across multiple switches by using trunk links, which carry traffic for multiple VLANs.



VLAN Identification Methods

VLAN identification is what switches use to keep track of all those frames as they're traversing a switch fabric. It's how switches identify which frames belong to which VLANs, and there's more than one trunking method.

Inter-Switch Link (ISL)

Inter-Switch Link (ISL) is a way of explicitly tagging VLAN information onto an Ethernet frame. This tagging information allows VLANs to be multiplexed over a trunk link through an external encapsulation method. This allows the switch to identify the VLAN membership of a frame received over the trunked link. By running ISL, you can interconnect multiple switches and still maintain VLAN information as traffic travels between switches on trunk links. ISL functions at layer 2 by encapsulating a data frame with a new header and by performing a new cyclic redundancy check (CRC). Of note is that ISL is proprietary to Cisco switches and it's used for Fast Ethernet and Gigabit Ethernet links only. ISL routing is pretty versatile and can be used on a switch port, router interfaces, and server interface cards to trunk a server. Although some Cisco switches still support ISL frame tagging, Cisco is moving toward using only 802.1q.

IEEE 802.1q

Created by the IEEE as a standard method of frame tagging, IEEE 802.1q actually inserts a field into the frame to identify the VLAN. If you're trunking between a Cisco switched link and a different brand of switch, you've got to use 802.1q for the trunk to work. Unlike ISL, which encapsulates the frame with control information, 802.1q inserts an 802.1q field along with tag control information, as shown in Figure 5. For the Cisco exam objectives, it's only the 12-bit VLAN ID that matters. This field identifies the VLAN and can be 212, minus 2 for the 0 and 4,095 reserved VLANs, which means an 802.1q tagged frame can carry information for 4,094 VLANs. It works like this: You first designate each port that's

going to be a trunk with 802.1q encapsulation. The other ports must be assigned a specific VLAN ID in order for them to communicate. VLAN 1 is the default native VLAN, and when using 802.1q, all traffic for a native VLAN is untagged. The ports that populate the same trunk create a group with this native VLAN and each port gets tagged with an identification number reflecting that. Again the default is VLAN 1. The native VLAN allows the trunks to accept information that was received without any VLAN identification or frame tag.

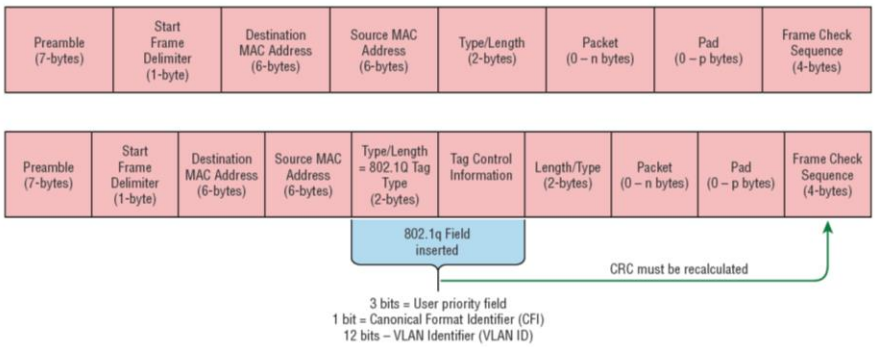


Figure 5 - IEEE 802.1q encapsulation with and without the 802.1q tag

Routing between VLANs

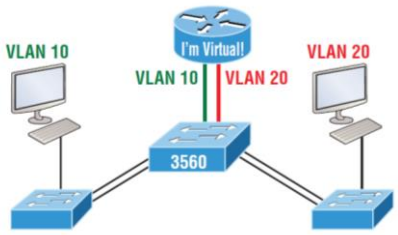


Figure 6 - With IVR, routing runs on the backplane of the switch, and it appears to the hosts that a router is present.

In Figure 6, it appears there's a router present, but there is no physical router present as there was when we used router on a stick. The IVR process takes little effort and is easy to implement, which makes it very cool! Plus, it's a lot more efficient for inter-VLAN routing than an external router is. To implement IVR on a multilayer switch, we just need to create logical interfaces in the switch configuration for each VLAN. We'll configure this method in a minute, but first let's take our existing switched network and add some VLANs, then configure VLAN memberships and trunk links between our switches.

Configuring VLANs

To configure VLANs on a Cisco Catalyst switch, use the global config vlan command.
In the following example, I'm going to demonstrate how to configure VLANs on the S1 switch by creating three VLANs for three different departments—again, remember that VLAN 1 is the native and management VLAN by default:

```
S1(config)#vlan ?
  WORD        ISL VLAN IDs 1-4094
  access-map  Create vlan access-map or enter vlan access-map command mode
  dot1q       dot1q parameters
  filter      Apply a VLAN Map
  group       Create a vlan group
  internal    internal VLAN
S1(config)#vlan 2
S1(config-vlan)#name Sales
S1(config-vlan)#vlan 3
S1(config-vlan)#name Marketing
S1(config-vlan)#vlan 4
S1(config-vlan)#name Accounting
S1(config-vlan)#^Z
S1#
```

```
S1#sh vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Gi0/1
                                                Gi0/2
2    Sales                                 active
3    Marketing                             active
4    Accounting                            active
[output cut]
```

In the preceding S1 output, you can see that ports Fa0/1 through *Fa0/14*, *Fa0/19* through *23*, and *Gi0/1* and *Gi0/2* uplinks are all in VLAN 1. But where are ports *15* through *18*? First, understand that the command show vlan only displays access ports, so now that you know what you're looking at with the show vlan command, where do you think ports Fa15–18 are? That's right! They are trunked ports. Cisco switches run a proprietary protocol called ***Dynamic Trunk Protocol (DTP),*** and if there is a compatible switch connected, they will start trunking automatically, which is precisely where my four ports are. You have to use the show interfaces trunk command to see your trunked ports like this:

```
S1# show interfaces trunk
Port        Mode             Encapsulation  Status        Native vlan
Fa0/15      desirable        n-isl          trunking      1
Fa0/16      desirable        n-isl          trunking      1
Fa0/17      desirable        n-isl          trunking      1
Fa0/18      desirable        n-isl          trunking      1

Port        Vlans allowed on trunk
Fa0/15      1-4094
Fa0/16      1-4094
Fa0/17      1-4094
Fa0/18      1-4094
```

This output reveals that the VLANs from 1 to 4094 are allowed across the trunk by default. Another helpful command is the ***show interfaces interface switchport*** command:

```
S1#sh interfaces fastEthernet 0/15 switchport
Name: Fa0/15
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: isl
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
[output cut]
```

The highlighted output shows us the administrative mode of dynamic desirable, that the port is a trunk port, and that DTP was used to negotiate the frame tagging method of ISL. It also predictably shows that the native VLAN is the default of 1. Now that we can see the VLANs created, we can assign switch ports to specific ones. Each port can be part of only one VLAN, with the exception of voice access ports. Using trunking, you can make a port available to traffic from all VLANs.

Assigning Switch Ports to VLANs

You configure a port to belong to a VLAN by assigning a membership mode that specifies the kind of traffic the port carries plus the number of VLANs it can belong to. You can also configure each port on a switch to be in a specific VLAN (access port) by using the interface switchport command. You can even configure multiple ports at the same time with the interface range command.
In the next example, I'll configure interface Fa0/3 to VLAN 3. This is the connection from the S3 switch to the host device:

```
S3#config t
S3(config)#int fa0/3
S3(config-if)#switchport mode access
S3(config-if)#switchport access vlan 3
```

Let's take a look at our VLANs now:

```
S3#show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/4, Fa0/5, Fa0/6, Fa0/7
                                                Fa0/8, Fa0/9, Fa0/10, Fa0/11,
                                                Fa0/12, Fa0/13, Fa0/14, Fa0/19,
                                                Fa0/20, Fa0/21, Fa0/22, Fa0/23,
                                                Gi0/1 ,Gi0/2

2    Sales                            active
3    Marketing                        active    Fa0/3
```

Configuring Trunk Ports

The 2960 switch only runs the IEEE 802.1q encapsulation method. To configure trunking on a FastEthernet port, use the interface command switchport mode trunk. It's a tad different on the 3560 switch. The following switch output shows the trunk configuration on interfaces Fa0/15–18 as set to trunk:

```
S1(config)#int range f0/15-18
S1(config-if-range)#switchport trunk encapsulation dot1q
S1(config-if-range)#switchport mode trunk
```

If you have a switch that only runs the 802.1q encapsulation method, then you wouldn't use the encapsulation command as I did in the preceding output. Let's check out our trunk ports now:

```
S1(config-if-range)#do sh int f0/15 swi
Name: Fa0/15
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
```

Notice that port Fa0/15 is a trunk and running 802.1q. Let's take another look:

```
S1(config-if-range)#do sh int trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/15      on               802.1q         trunking      1
Fa0/16      on               802.1q         trunking      1
Fa0/17      on               802.1q         trunking      1
Fa0/18      on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/15      1-4094
Fa0/16      1-4094
Fa0/17      1-4094
Fa0/18      1-4094
```

Take note of the fact that ports 15–18 are now in the trunk mode of on and the encapsulation is now 802.1q instead of the negotiated ISL. Here's a description of the different options available when configuring a switch interface:

switchport mode access I discussed this in the previous section, but this puts the interface (access port) into permanent non-trunking mode and negotiates to convert the link into a non-trunk link. The interface becomes a non-trunk interface regardless of whether the neighboring interface is a trunk interface. The port would be a dedicated layer 2 access port.

switchport mode dynamic auto This mode makes the interface able to convert the link to a trunk link. The interface becomes a trunk interface if the neighboring interface is set to trunk or desirable mode. The default is dynamic auto on a lot of Cisco switches, but that default trunk method is changing to dynamic desirable on most new models.

switchport mode dynamic desirable This one makes the interface actively attempt to convert the link to a trunk link. The interface becomes a trunk interface if the neighboring interface is set to trunk, desirable, or auto mode. I used to see this mode as the default on some switches, but not any longer. This is now the default switch port mode for all Ethernet interfaces on all new Cisco switches.

switchport mode trunk Puts the interface into permanent trunking mode and negotiates to convert the neighboring link into a trunk link. The interface becomes a trunk interface even if the neighboring interface isn't a trunk interface.

switchport nonegotiate Prevents the interface from generating DTP frames. You can use this command only when the interface switchport mode is access or trunk. You must manually configure the neighboring interface as a trunk interface to establish a trunk link.

Defining the Allowed VLANs on a Trunk

```
S1#sh int trunk
[output cut]

Port        Vlans allowed on trunk
Fa0/15      1-4094
Fa0/16      1-4094
Fa0/17      1-4094
Fa0/18      1-4094

S1(config)#int f0/15
S1(config-if)#switchport trunk allowed vlan 4,6,12,15
S1(config-if)#do show int trunk
[output cut]

Port        Vlans allowed on trunk
Fa0/15      4,6,12,15
Fa0/16      1-4094
Fa0/17      1-4094
Fa0/18      1-4094
```

To remove a range of VLANs, just use the hyphen:

```
S1(config-if)#switchport trunk allowed vlan remove 4-8
```

If by chance someone has removed some VLANs from a trunk link and you want to set the trunk back to default, just use this command:

```
S1(config-if)#switchport trunk allowed vlan all
```

Configuring Inter-VLAN Routing

By default, only hosts that are members of the same VLAN can communicate. To change this and allow inter-VLAN communication, you need a router or a layer 3 switch.

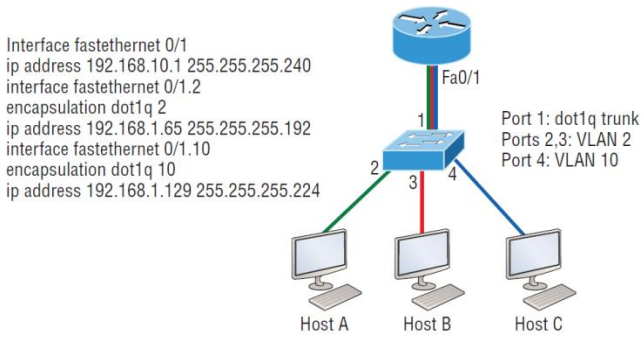


Figure 7 - Configuring inter-VLAN

The next step is to figure out which subnets are being used. By looking at the router configuration in the figure, you can see that we're using 192.168.1.64/26 with VLAN 2, 192.168.1.128/27, and VLAN 10. By looking at the switch configuration, you can see that ports 2 and 3 are in VLAN 2 and port 4 is in VLAN 10. This means that Host A and Host B are in VLAN 2, and Host C is in VLAN 10.
Here's what the hosts' IP addresses should be:

Host A: 192.168.1.66, 255.255.255.192, default gateway 192.168.1.65

Host B: 192.168.1.67, 255.255.255.192, default gateway 192.168.1.65

Host C: 192.168.1.130, 255.255.255.224, default gateway 192.168.1.129

Now, again using Figure 11.12, let's go through the commands necessary to configure switch port 1 so it will establish a link with the router and provide inter-VLAN communication using the IEEE version for encapsulation. Keep in mind that the commands can vary slightly depending on what type of switch you're dealing with. For a 2960 switch, use the following:

```
2960#config t
2960(config)#interface fa0/1
2960(config-if)#switchport mode trunk
```

That's it! As you already know, the 2960 switch can only run the 802.1q encapsulation, so there's no need to specify it. You can't anyway. For a 3560, it's basically the same, but because it can run ISL and 802.1q, you have to specify the trunking encapsulation protocol you're going to use.

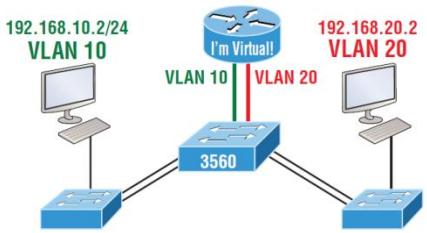


Figure 8 - Inter-VLAN routing with a multilayer switch

The hosts are already configured with the IP address, subnet mask, and default gateway address using the first address in the range. Now I just need to configure the routing on the switch, which is pretty simple actually:

```
S1(config)#ip routing
S1(config)#int vlan 10
S1(config-if)#ip address 192.168.10.1 255.255.255.0
S1(config-if)#int vlan 20
S1(config-if)#ip address 192.168.20.1 255.255.255.0
```

And that's it! Enable IP routing and create one logical interface for each VLAN using the interface vlan number command and voila! You've now accomplished making inter-VLAN routing work on the backplane of the switch!