VLANs

Introduction:

Virtual LANs, or VLANs, are virtual separations within a switch that provides distinct logical LANs that each behaves as if they were configured on a separate physical switch.

VLANs are a mechanism to allow network administrators to create logical broadcast domains that can span across a single switch or multiple switches, regardless of physical proximity. This function is useful to reduce the size of broadcast domains or to allow groups or users to be logically grouped without the need to be physically located in the same place.

Connecting VLANs:

Figure 1 shows a switch with multiple VLANs. The VLANs have been numbered 10, 20, and 30. In general, VLANs can be named or numbered. The default VLAN is numbered 1. If you plug a number of devices into a switch without assigning its ports to specific VLANs, all the devices will reside in VLAN 1.

Frames cannot leave the VLANs from which they originate. This means that in the example configuration, Rami can communicate with Eyad, and Ali can communicate with Ahmad, but Mohammad cannot communicate with Eyad or Rami or Ahmad or Ali in any way.

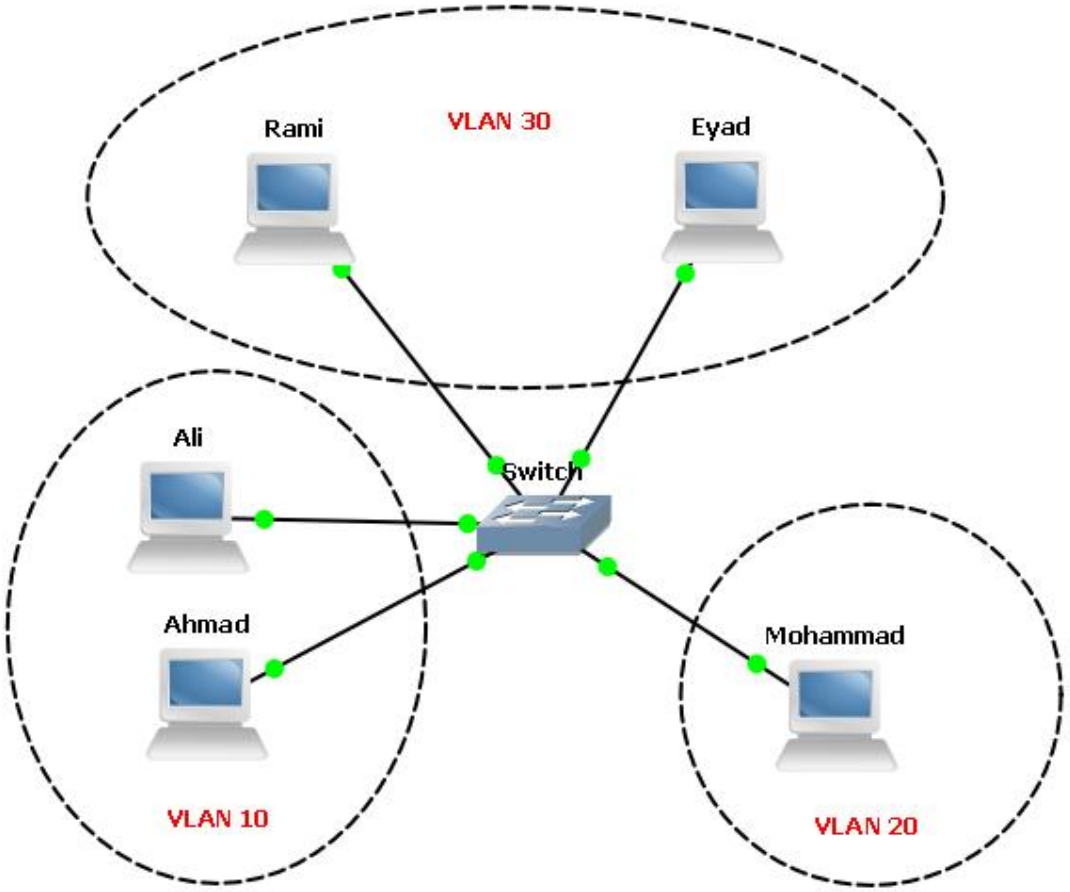


Figure 1

For a packet on a Layer-2 switch to cross from one VLAN to another, an outside router must be attached to each of the VLANs to be routed. For example if you add an external router connecting VLAN 20 with VLAN 30. Assuming a proper configuration on the router, Mohammad will now be able to communicate with Rami and Eyad, but no workstation will show any indication that they reside on the same physical switch.

When expanding a network using VLANs, you face the same limitations. If you connect another switch to a port that is configured for VLAN 20, the new switch will be able to forward frames only to or from VLAN 20. If you wanted to connect two switches, each containing four VLANs, you would need four links between the switches: one for each VLAN. A solution to this problem is to deploy *trunks* between switches. Trunks are links that carry frames for more than one VLAN.

Figure 2 shows two switches connected with a trunk. Rami is connected to VLAN 30 on Switch 1, and Eyad is connected to VLAN 30 on Switch 2. Because there is a trunk connecting these two switches together, assuming the trunk is allowed to carry traffic for all configured VLANs, Rami will be able to communicate with Eyad. Notice that the ports to which the trunk is connected are not assigned VLANs. These ports are *trunk ports* and, as such, do not belong to a single VLAN.

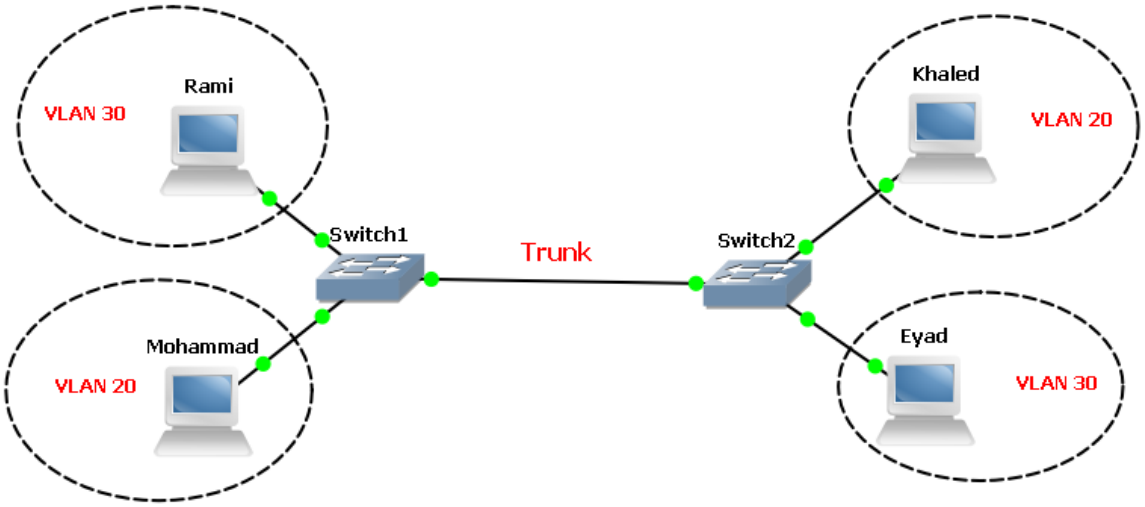


Figure 2

Trunks also allow another possibility with switches. Imagine you want to route between *all* of the VLANs on the switch. How would you go about implementing such a design? Traditionally, the answer would be to provide a single connection from the router to each network to be routed. On this switch, each network is a VLAN, so you'd need a physical connection between the router and each VLAN. As you can see, with this setup, 3 interfaces are being used both on the switch and on the router. Smaller routers rarely have three Ethernet interfaces, though, and Ethernet interfaces on routers can be costly. Additionally, users buy switches with a certain port density in mind. In this configuration, 3 ports of the entire switch have been used up just for routing between VLANs.

Another way to route between VLANs is commonly known as the *router-on-a-stick* configuration. Instead of running a link from each VLAN to a

router interface, you can run a single trunk from the switch to the router. All the VLANs will then pass over a single link.

Deploying a router on a stick saves a lot of interfaces on both the switch and the router. The downside is that the trunk is only one link, and the total bandwidth available on that link is only 10 Mbps. In contrast, when each VLAN has its own link, each VLAN has 10 Mbps to itself. Also, don't forget that the router is passing traffic between VLANs, so chances are each frame will be seen twice on the same link—once to get to the router, and once to get back to the destination VLAN.

Configuring VLANs Using IOS:

Adding VLANs in IOS is relatively straightforward when all of the defaults are acceptable, which is usually the case.
First, enter configuration mode. From there, issue the `vlan` command with the identifier for the VLAN you're adding or changing. Next, specify a name for the VLAN with the `name` subcommand (a default name of VLAN*xxxx* is used if you do not supply one):

```
Switch>enable

Switch#configure terminal

Switch(config)#vlan 10

Switch(config-vlan)#name IT
```

Exit configuration mode and then issue the `show vlan` command to see the VLANs present:

```
Switch#show 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, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
10   IT                               active
```

You assign ports to VLANs in IOS in interface configuration mode. Each interface must be configured individually with the `switchport access` command:

```
Switch(config)#interface fastEthernet 0/1

Switch(config-if)#switchport access vlan 10

Switch(config-if)#interface fastEthernet 0/2

Switch(config-if)#switchport access vlan 10
```

Modern versions of IOS allow you to apply commands to multiple interfaces with the `interface range` command. Using this command, you can accomplish the same result as before while saving some precious keystrokes:

```
Switch(config)#interface range fastEthernet 0/1 - 2

Switch(config-if-range)# switchport access vlan 10
```

Now, when you execute the `show vlan` command, you'll see that the ports have been assigned to the proper VLAN:

```
Switch#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/3, Fa0/4, Fa0/5, Fa0/6
                                                Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                                Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                                Fa0/23, Fa0/24
10   IT                               active    Fa0/1, Fa0/2
```

How Trunks Work:

To mark, or *tag*, frames to be sent over a trunk, both sides must agree to a protocol. Currently, the protocol for trunking is the IEEE standard 802.1Q.

802.1Q takes a different approach to VLAN tagging. Instead of adding more headers to a frame, 802.1Q inserts data into existing headers. An additional four-byte tag field is inserted between the Source Address and Type/Length

fields. Because 802.1Q has altered the frame, the FCS of the frame is altered to reflect the change.

Configuring Trunks using IOS:

To configure one of the Fast Ethernet ports to be an 802.1Q trunk using a 2950 switch:

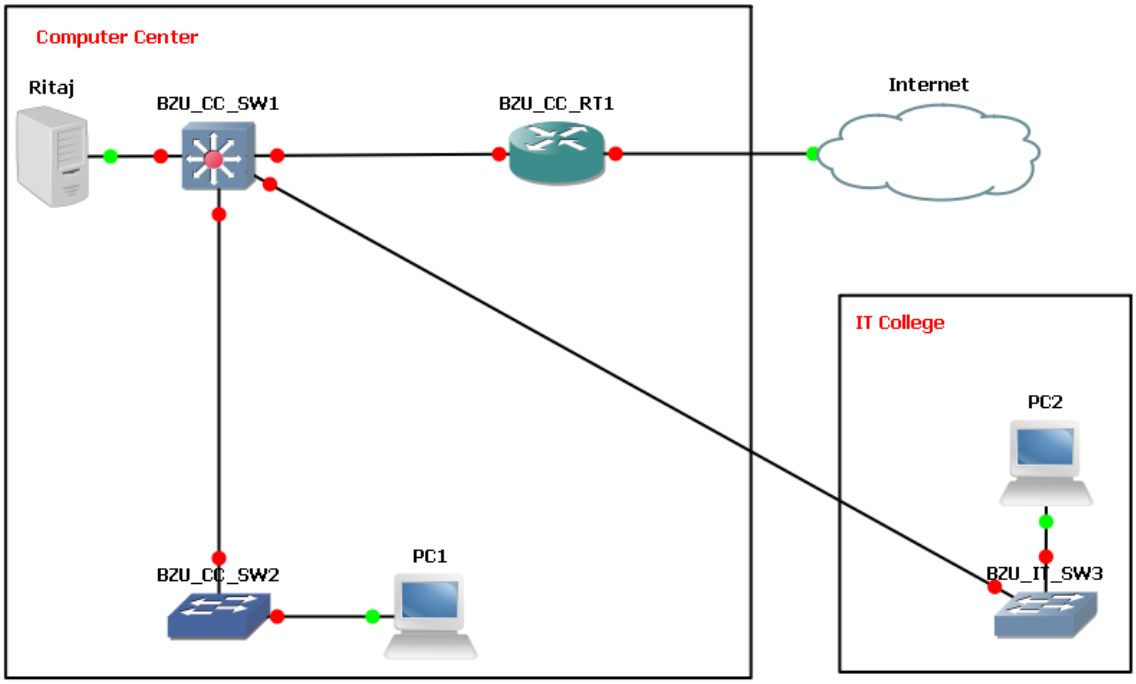
```
Switch(config)#interface fastEthernet 0/24

Switch(config-if)#switchport mode trunk
```

Equipment:

1. 2 Layer-2 Cisco Switches (i.e. 2950).
2. 1 Layer-3 Cisco Switch (i.e. 3560).
3. 1 Router (i.e. 2811).
4. Cables.

Procedure:

Scenario: Build and configure the following topology based on the following requirements.



Requirements:

To help guide this initial configuration, you've assembled a list of requirements.

- Configure the necessary VLANs on BZU-CC-SW1, BZU-CC-SW2, and BZU-IT-SW3.
  VLAN10: Student -- Network 192.168.10.0/24
  VLAN20: Instructor -- Network 192.168.20.0/24
  VLAN30: Server -- Network 192.168.30.0/24
- Assign hosts to the proper VLAN.
  PC1: VLAN20
  PC2: VLAN10
  Ritaj: VLAN30
- Configure Ritaj Server and PCs with the following configuration:

	Ritaj Server	**PC1**	**PC2**
Interface:	NIC	NIC	NIC
IP Address:	192.168.30.2	192.168.20.2	192.168.10.2
Gateway:	192.168.30.1	192.168.20.1	192.168.10.1

- Configure trunks connections between (BZU-CC-SW1 and BZU-CC-SW2) and (BZU-CC-SW1 and BZU- IT-SW3).
- Enable routing on the BZU-CC-SW1 by using the `ip routing` command.
  ```
  BZU-CC-SW1(config)#ip routing
  ```
- Create a routed interface on BZU-CC-SW1 for each of the VLANs. This interface should be assigned the first IP address from each of the VLAN subnets.
  ```
  BZU-CC-SW1#configure terminal
  BZU-CC-SW1(config)#interface Vlan10
  BZU-CC-SW1(config-if)#ip address 192.168.10.1 255.255.255.0
  BZU-CC-SW1(config-if)#no shutdown
  ```
  Repeat this process to VLAN20 and VLAN30

- Configure the interface of BZU-CC-SW1 connected to the default router (BZU-CC-RT1). In this scenario you have a Layer 3 FastEthernet port.

```
BZU-CC-SW1(config)#interface FastEthernet 0/24
BZU-CC-SW1(config-if)#no switchport
BZU-CC-SW1(config-if)#ip address 200.1.1.2 255.255.255.0
BZU-CC-SW1(config-if)#no shutdown
```

  The `no switchport` command makes the interface Layer 3 capable. The IP address is in the same subnet as the default router.

- Configure a static default route on BZU-CC-SW1 using the inside IP address of BZU_CC_RT1 (i.e. 200.1.1.1)

```
BZU-CC-SW1#configure terminal
BZU-CC-SW1(config)#ip route 0.0.0.0 0.0.0.0 200.1.1.1
```

- Testing
  1. PC1 should be able to ping PC2.
  2. PC1 and PC2 should both be able to perform a ping and traceroute to the Ritaj Server.
  3. BZU-CC-SW1 should be able to ping the default Router R1.
  4. Use the `show ip route` on BZU-CC-SW1.

```
BZU-CC-SW1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 200.1.1.1 to network 0.0.0.0

C    192.168.10.0/24 is directly connected, Vlan10
C    192.168.20.0/24 is directly connected, Vlan20
C    192.168.30.0/24 is directly connected, Vlan30
C    200.1.1.0/24 is directly connected, FastEthernet0/24
S*   0.0.0.0/0 [1/0] via 200.1.1.1
BZU-CC-SW1#
```