CSE 4412 [Data Communication and Networking Lab]

Lab # 04

1. Objectives:

- Design and implement Inter-VLAN routing using Router on a stick
- Design and implement Inter-VLAN routing using Multilayer Switch
- Understand and implement Static Routing

2. Theory:

As with other labs, this lab will also build up on the concepts and techniques of previous labs. So, make sure you've properly understood the previous lab contents.

VLAN:

VLAN or *Virtual LAN* (Local Area Network) is a logical grouping of networking devices. When we create VLAN, we actually break large broadcast domain in smaller broadcast domains. Consider VLAN as a subnet. Same as two different subnets cannot communicate with each other without router, different VLANs also requires router to communicate.

Advantages of VLAN

VLAN provides following advantages:-

- Solve broadcast problem
- Reduce the size of broadcast domains
- Allow us to add additional layer of security
- Make device management easier
- Allow us to implement the logical grouping of devices by function instead of location

Solves broadcast problem

When we connect devices into the switch ports, switch creates single broadcast domain for all ports. Switch forwards a broadcast frame from all possible ports. In a large network having hundreds of computers, it could create performance issues. Of course, we could use routers to solve broadcast problem, but that would be costly solution since each broadcast domain requires its own port on router. Switch has a unique solution to broadcast issue known as VLAN. In practical environment, we use VLAN to solve broadcast issue instead of router.

Each VLAN has a separate broadcast domain. Logically VLANs are also subnets. Each VLAN requires a unique network number known as VLAN ID. Devices with same VLAN ID are the members of same broadcast domain and receive all broadcasts. These broadcasts are filtered from all ports on a switch that aren’t members of the same VLAN.

Reduces the size of broadcast domains

VLANs increase the numbers of broadcast domain while reducing their size. For example, lets consider we have a network of 100 devices. Without any VLAN implementation, we

have single broadcast domain that contain 100 devices. We create 2 VLANs and assign 50 devices in each VLAN. Now we have two broadcast domains with fifty devices in each. Thus, more VLAN means more broadcast domain with less devices.

Allows us to add additional layer of security

VLANs enhance the network security. In a typical layer 2 network, all users can see all devices by default. Any user can see network broadcast and responds to it. Users can access any network resources located on that specific network. Users could join a workgroup by just attaching their system in existing switch. This could create real trouble on security platform. Properly configured VLANs gives us total control over each port and users. With VLANs, you can control the users from gaining unwanted access over the resources. We can put the group of users that need high level security into their own VLAN so that users outside from VLAN can't communicate with them.

Makes device management easier

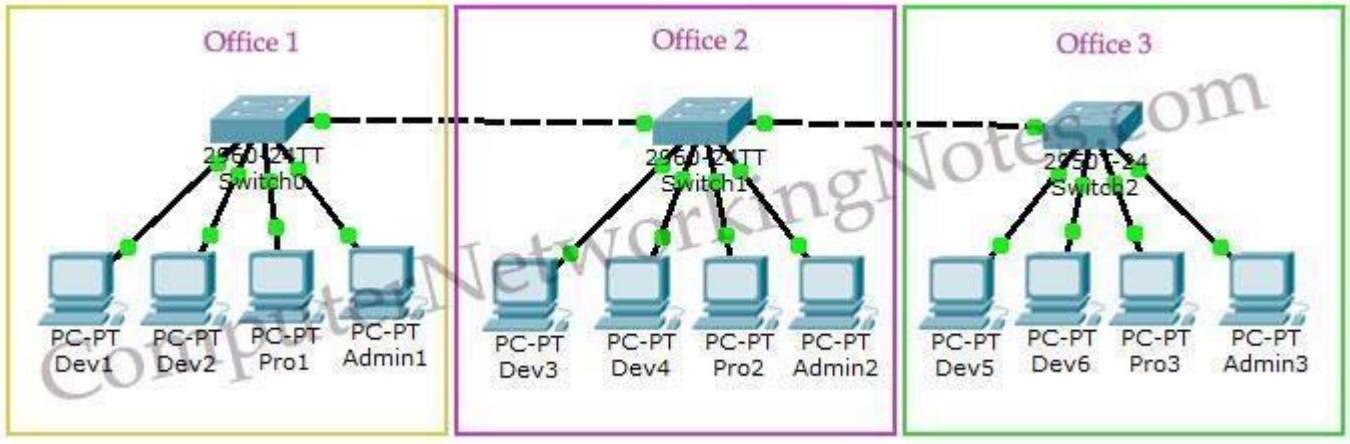
Device management is easier with VLANs. Since VLANs are a logical approach, a device can be located anywhere in the switched network and still belong to the same broadcast domain. We can move a user from one switch to another switch in same network while keeping his original VLAN. For example, a company has a five story building and a single layer two network. In this scenario, VLAN allows to move the users from one floor to another floor while keeping his original VLAN ID. The only limitation is that device when moved, must still be connected to the same layer 2 network.

Allows us to implement the logical grouping of devices by function instead of location

VLANs allow us to group the users by their function instead of their geographic locations. Switches maintain the integrity of your VLANs. Users will see only what they are supposed to see regardless what their physical locations are.

VLAN Examples

To understand VLAN more clearly let's take an example.



- Our company has three offices.
- All offices are connected with back links (links connecting switches).
- Company has three departments Development, Production and Administration.
- Development department has six computers.
- Production department has three computers.
- Administration department also has three computers.

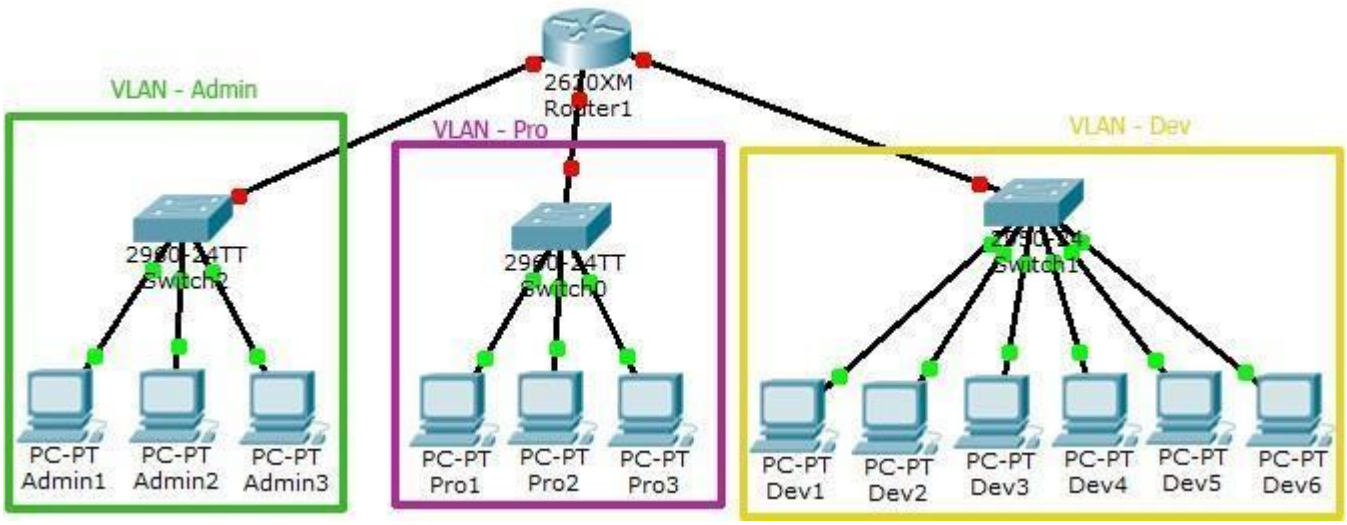
- Each office has two PCs from development department and one from both production and administration department.
- Administration and production department have sensitive information and need to be separate from development department.

With default configuration, all computers connected to the same switch share same broadcast domain. Development department can access the administration or production department resources.

With VLAN we could create logical boundaries over the physical network. Assume that we created three VLANs for our network and assigned them to the related computers.

- VLAN **Admin** for Administration department
- VLAN **Dev** for Development department
- VLAN **Pro** for Production department

Physically we changed nothing but logically we grouped devices according to their function. These groups [VLANs] need router to communicate with each other. Logically our network look likes following diagram.



With the help of VLAN, we have separated our single network in three small networks. These networks do not share broadcast with each other improving network performance and enhancing security. Now Development department cannot access the Administration and Production department directly.

VLAN Connections

During the configuration of VLAN on port, we need to know what type of connection it has. Switch supports two types of VLAN connection:

- Access link
- Trunk link

Access link

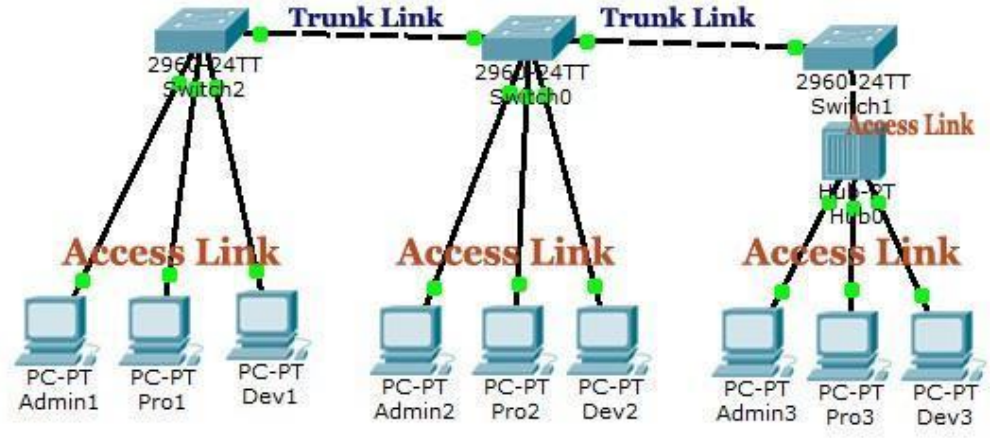
Access link connection is the connection where switch port is connected with a device that has a standardized Ethernet NIC. Standard NIC only understand IEEE 802.3 or Ethernet II

frames. Access link connection can only be assigned with *single* VLAN. That means all devices connected to this port will be in same broadcast domain.

For example, twenty users are connected to a hub, and we connect that hub with an access link port on switch, then all of these users belong to same VLAN. If we want to keep ten users in another VLAN, then we have to purchase another hub. We need to plug in those ten users in that hub and then connect it with another access link port on switch.

Trunk link

Trunk link connection is the connection where switch port is connected with a device that is capable of understanding multiple VLANs. Usually trunk link connection is used to connect two switches or switch to router. Remember earlier when we said that VLAN can span anywhere in network, that is basically due to trunk link connection. Trunking allows us to send or receive VLAN information across the network. To support trunking, original Ethernet frame is modified to carry VLAN information.



Lab Demo (with Inter-VLAN routing)

I. At first, configure 2 VLANs with VLAN ID 10, 20 inside the switch and assign appropriate names.

```
S1(config)# vlan 10

S1(config)# name [VLAN_name]

S1(config-vlan)# exit

S1(config)# vlan 20

S1(config)# name [VLAN_name]

S1(config-vlan)# exit

S1(config)# exit

S1# show vlan
```

II. Now, configure the Interfaces belonging to each VLAN:

```
S1(config)# interface Fast-Ethernet 0/1
S1(config-if)# switchport mode access
```

This command configures the interface as an access link (see theory section to understand whats an access link).

```
S1(config-if)# switchport access vlan 10
```

This command assigns VLAN 10 access ports.

```
S1(config-if)# no shutdown
```

Do this for all the ports!

The interface connected to the router will be the trunk port.

```
S1(config)# interface Fast-Ethernet 0/5
S1(config-if)# switchport mode trunk
```

This command configures the interface as a trunk link (see theory section to understand whats a trunk link).

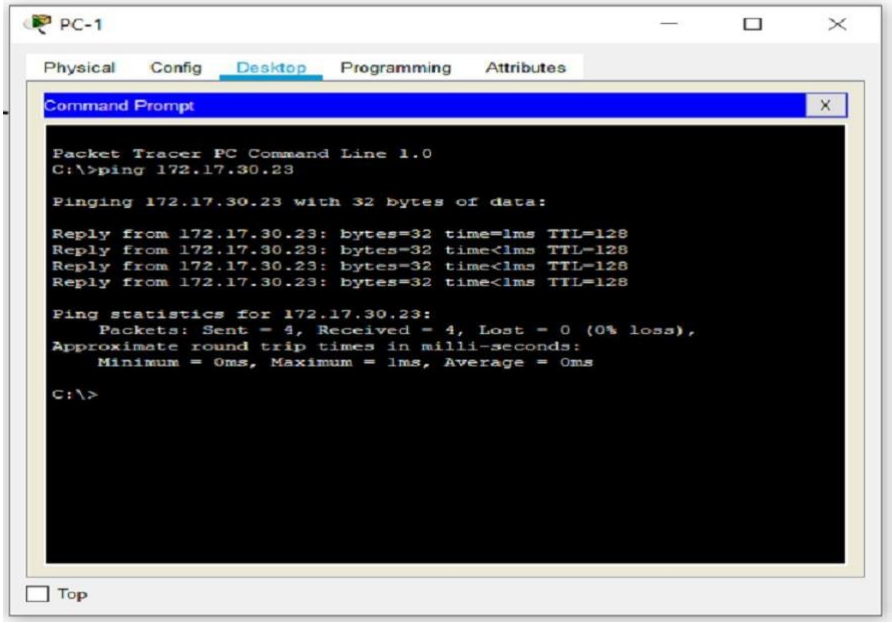
```
S1(config-if)# switchport trunk allowed vlan all
```

This command specifies the list of VLANs specified on the trunk port. In this case, we've allowed *all* the VLANs.

```
S1(config-if)# no shutdown
```

III. Setup the PCs with appropriate IP and subnet masks:

IV. Verify the routing is working properly by pinging *from different PCs*.



V. Now, we can reach the PCs of the VLAN but not the other VLANs. So, we need Inter-VLAN routing.

First we need to keep the router running:

```
Router(config)#int g0/0
Router(config-if)#no shutdown
```

Now, we need to assign virtual VLAN to the router interface:

```
Router(config)#int g0/0.10
Router(config-subif)#encapsulation dot1q 10
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)#int g0/0.20
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
Router(config-subif)#exit
```

VI. After all these are done, we can now ping from one PC to another.

In the case of MLS or Multi-Layer Switch, we need to do the following:

```
Switch(config)#vlan 10
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#ip routing
```

Now, we can make the trunk links and assign IP addresses for the VLANs.

```
Switch(config)#int range fa0/1-2
Switch(config-if-range)#switchport trunk encapsulation dot1q
Switch(config-if-range)#switchport mode trunk
Switch(config-if-range)#no shutdown
Switch(config-if-range)#int vlan 10
Switch(config-if)#ip address 192.168.10.1 255.255.255.0
Switch(config-if)#no shut
Switch(config-if)#exit
```

```
Switch(config)#int vlan 20

Switch(config-if)#ip address 192.168.20.1 255.255.255.0

Switch(config-if)#no shut
```

Static Routing:

By now, you all know that routers take the help of routing table to forward packets to the intended destinations. When a packet reaches the router, it looks up the routing table, finds the corresponding output interfaces for the destination network address and sends the packet through that interface. The question is how this routing table is formed in the first place. The answer is either manually configuring the routing entries or using routing protocols to configure the routes dynamically. The first approach is called static routing and this is what we'll talk about and learn in this lab. The second approach is the topic of next lab.

In static routing, the network administrator manually adds the routing entries to the routing table. The routing entries will not be changed automatically. All changes have to be done manually. If the network condition changes (for example, *some link goes down*), then the necessary changes in the routing table must be done manually whereas these info are updated automatically in dynamic routing.

In practical large networks, static routing is mostly used as a backup to dynamic routing. Unlike dynamic routing, static routing requires very less computational resource and bandwidth as no extra packet is required for routing table update process. But as the network administrator needs to know the whole network topology and network addresses to effectively configure the routing table, static routing is not used as the only routing mechanism in large scale networks.

There are some concepts related to static routing that you need to be familiar with before you get your hands dirty. You know that packets travel from one hop to the next to reach its final destination. In the routing table of a router, the next hop address is associated with a certain destination address. Its not realistic to assume that there would be next-hop entry for every possible destination network. That's why a routing entry known as the **default route** is present in the routing table. It defines a default exit interface for the packets that don't have any corresponding route in the routing table.

When working with CISCO devices and specifically for this lab, you'll encounter two types of static default routes. One is **directly connected** static default route and another is **next-hop** static default route. You'll have to configure these two types of routes in this lab.

The general format of the command to specify static routes is:

```
ip route destination_network_prefix destination_prefix_mask (next-
            hop_address | interface) [distance_metric]
ipv6 route ipv6_destination_network_prefix(with CIDR) (ipv6_next-hop_address
                    | interface) [distance_metric]
```

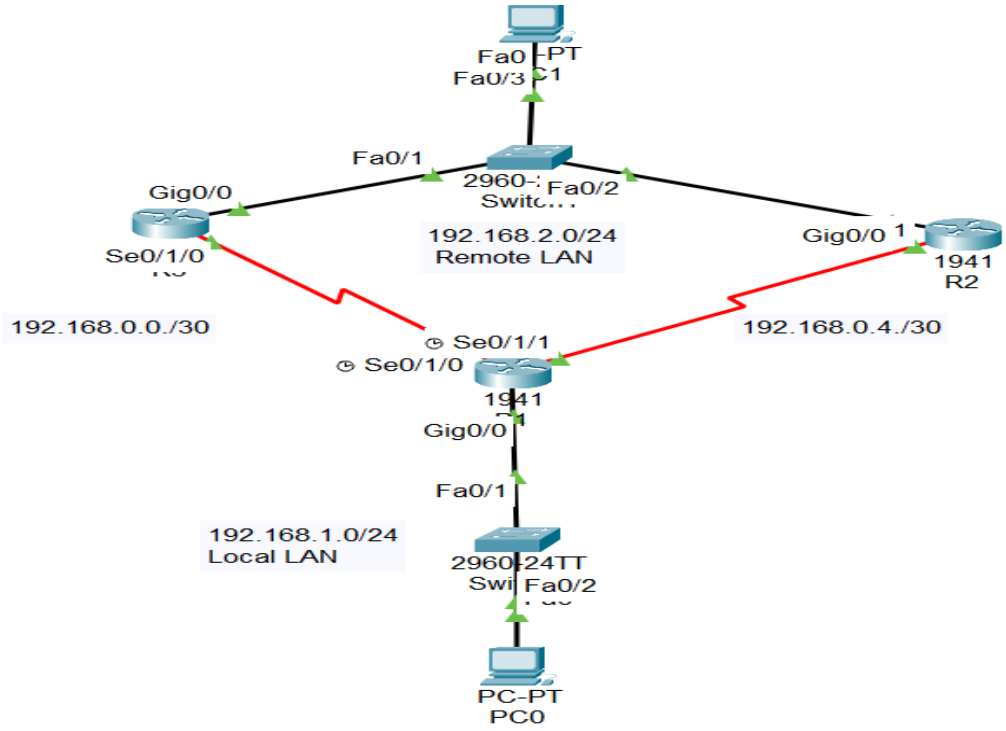
In case of directly connected static default routes, you'll specify the ***interface***. In case of next-hop static default routes, you'll specify the ***next_hop address***. One special use case of the above command is to configure a **primary static default route** where both the *destination_network_prefix* and *destination_prefix_mask* are *0.0.0.0*. The IPv4 and IPv6 command format for specifying primary static default route is given below:

```
ip route 0.0.0.0 0.0.0.0 (next-hop_address | interface) [distance_metric]
ipv6 route ::/0 (ipv6_next-hop_address | interface) [distance_metric]
```

The above commands basically mean that "*packets from any IP address with any subnet mask get sent to the specified next-hop address or interface*".

Another concept when configuring static routing is the **floating static route**. A floating route is nothing but the route that's used to forward a packet to a certain destination when main route is unavailable. The way floating routes are defined is by providing a higher ***distance_metric*** to a certain route. The default ***distance_metric*** when its not manually specified is **1**. The floating static routes are given higher numbers than 1. Routers always take the route with *lower* ***distance_metric*** when multiple routes to the same destination are available. That's why this floating static route will only be used when main route is down or unavailable.

Configure static routing:



I. **Configure R1 Interfaces**

```
R1(config)#int g0/0
R1(config-if)#ip add 192.168.1.1 255.255.255.0
R1(config-if)#desc connection-to-PC0
R1(config-if)#no shutdown
R1(config-if)#exit

R1(config)#int s0/1/0
R1(config-if)#ip add 192.168.0.2 255.255.255.252
R1(config-if)#desc connection-to-R3
R1(config-if)#clock rate 64000
R1(config-if)#no shutdown
R1(config-if)#exit
```

```
R1(config)#int s0/1/1
R1(config-if)#ip add 192.168.0.6 255.255.255.252
R1(config-if)#desc connection-to-R2
R1(config-if)#clock rate 64000
R1(config-if)#no shutdown
R1(config-if)#exit
```

II. **Configure R2 Interfaces**

```
R2(config)#int s0/1/1
R2(config-if)#ip add 192.168.0.5 255.255.255.252
R2(config-if)#desc connection-to-R1
R2(config-if)#no shutdown
R2(config-if)#exit

R2(config)#int g0/0
R2(config-if)#ip add 192.168.2.1 255.255.255.0
R2(config-if)#desc connection-to-RemoteLAN
R2(config-if)#no shutdown
R2(config-if)#exit
```

III. **Configure R3 Interfaces**

```
R3(config)#int s0/1/0
R3(config-if)#ip add 192.168.0.1 255.255.255.252
R3(config-if)#desc connection-to-R1
R3(config-if)#no shutdown
R3(config-if)#exit

R3(config)#int g0/0
R3(config-if)#ip add 192.168.2.2 255.255.255.0
R3(config-if)#desc connection-to-RemoteLAN
R3(config-if)#no shutdown
R3(config-if)#exit
```

IV. Configure PC0

```
IP: 192.168.1.10
Mask: 255.255.255.0
Gateway: 192.168.1.1
```

V. Configure PC1

```
IP: 192.168.2.10
Mask: 255.255.255.0
Gateway: 192.168.2.1
```

VI. Configure static routing to Remote LAN in R1

```
R1(config)#ip route 192.168.2.0 255.255.255.0 s0/1/1
```

It's a **directly connected** static default route.

```
R1(config)#ip route 192.168.2.0 255.255.255.0 192.168.0.1 5
```

It's a **next-hop floating** static default route.

VII. Configure static routing to Local LAN in R2

```
R2(config)#ip route 192.168.1.0 255.255.255.0 s0/1/1
```

It's a **directly connected** static default route.

```
R2(config)#ip route 192.168.1.0 255.255.255.0 192.168.0.6 5
```

It's a **next-hop floating** static default route

VIII. Configure static routing to Local LAN in R3

```
R2(config)#ip route 192.168.1.0 255.255.255.0 s0/1/0
```

It's a **directly connected** static default route.

```
R2(config)#ip route 192.168.1.0 255.255.255.0 192.168.0.2 5
```

It's a **next-hop floating** static default route

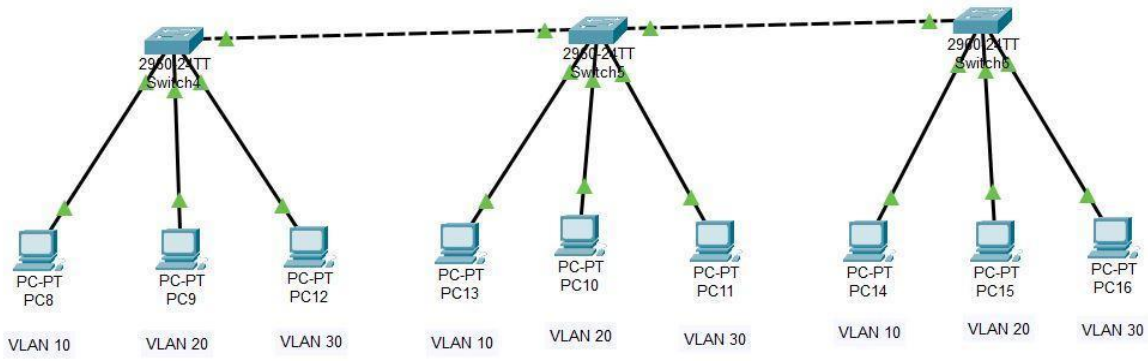
IX. Verify

```
Ping PC1 from PC0
```

3. Tasks:

Here, X = last 2 digits of your Student ID

I. You will implement Inter-VLAN routing. VLAN 10, 20 and 30 are Students, Teachers and Admin respectively. Assign IPs from **192.168.X+10.0**, **192.168.X+20.0** and **192.168.X+30.0** for the VLAN 10, 20 and 30 respectively. **Establish the Inter-VLAN routing by attaching a router on one of the switches (Router-on-a-stick).**



II. Create another file with 2 switches having 4 VLANs each. Assign IPs from **192.168.X+11.0**, **192.168.X+12.0, 192.168.X+13.0** and **192.168.X+14.0** for the VLAN 10, 20, 30 and 40 respectively. Now, connect a Multi-Layer Switch (Layer 3 switch) and implement Inter-VLAN communication properly.

III. Implement the Demo of Static Routing by yourself and write proper explanations using labels/textboxes. But for the IP addresses, instead of **192.168**, you should use **192.X+10.** The rest should remain the same.