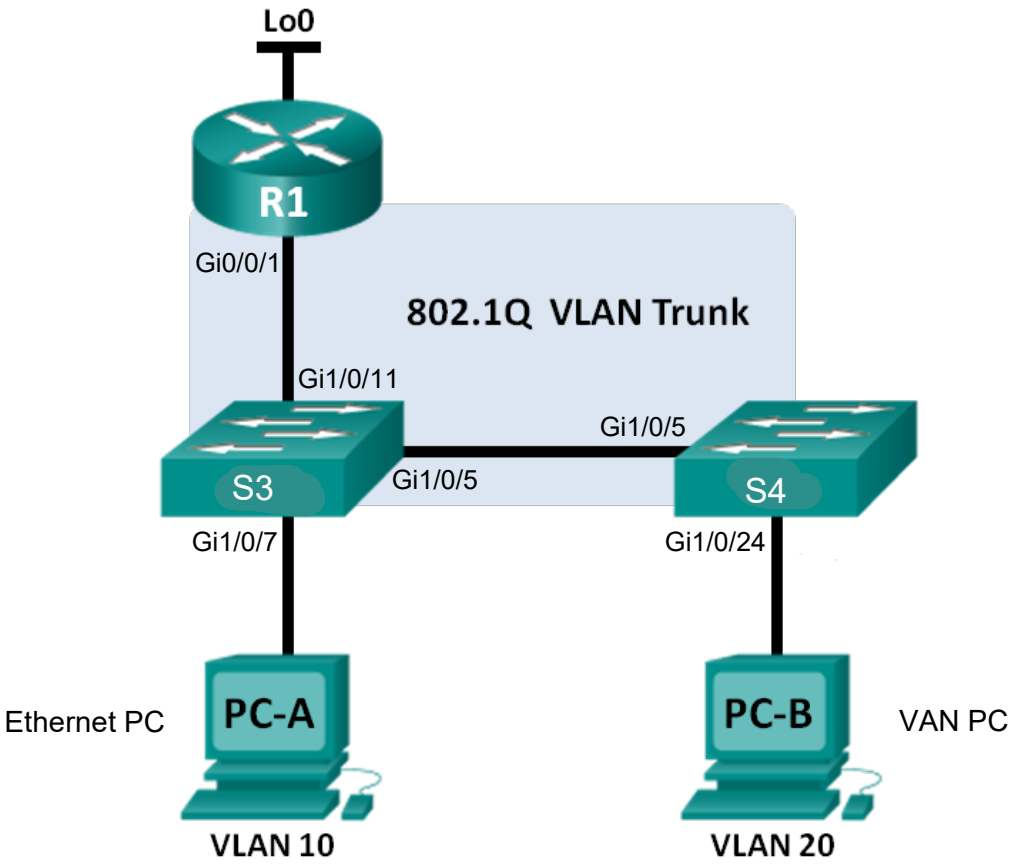
Lab SU-5b – Configuring 802.1Q Trunk-Based Inter-VLAN Routing

Topology Diagram



Modifications to Network Drawing

If you are working via remote access, the PCs in the diagram are just for reference and will not be connected to your lab topology. If you are working on-campus, you will set up virtual PCs as PC-A and PC-B.

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	Gi0/0/1.99	192.168.1.1	255.255.255.0	N/A
	Gi0/0/1.10	192.168.10.1	255.255.255.0	N/A
	Gi0/0/1.20	192.168.20.1	255.255.255.0	N/A
	Lo0	209.165.200.225	255.255.255.224	N/A
S3	VLAN 99	192.168.1.11	255.255.255.0	192.168.1.1
S4	VLAN 99	192.168.1.12	255.255.255.0	192.168.1.1
PC-A	NIC	192.168.10.3	255.255.255.0	192.168.10.1
PC-B	NIC	192.168.20.3	255.255.255.0	192.168.20.1

Switch Port Assignment Specifications

Ports	Assignment	Network
S3 Gi1/0/5	802.1Q Trunk	N/A
S4 Gi1/0/5	802.1Q Trunk	N/A
S3 Gi1/0/11	802.1Q Trunk	N/A
S3 Gi1/0/7	VLAN 10 – Students	192.168.10.0/24
S4 Gi1/0/24	VLAN 20 – Faculty	192.168.20.0/24

Objectives

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Configure Switches with VLANs and Trunking

Part 3: Configure Trunk-Based Inter-VLAN Routing

Background / Scenario

A second method of providing routing and connectivity for multiple VLANs is through the use of an 802.1Q trunk between one or more switches and a single router interface. This method is also known as **router-on-a-stick** inter-VLAN routing. In this method, the physical router interface is divided into multiple subinterfaces that provide logical pathways to all VLANs connected.

In this lab, you will configure trunk-based inter-VLAN routing and verify connectivity to hosts on different VLANs as well as with a loopback on the router.

Note: This lab provides minimal assistance with the actual commands necessary to configure the router and switches. You should refer to your lab journal and previous lab handouts if you require assistance.

Note: This lab is a variation of the previous lab, you should start with the previous configuration and make the necessary changes.

Required Resources

- 1 Router
- 2 Switches
- Ethernet cables as shown in the topology
- 2 PCs (when working on-campus)

Part 1: Build the Network and Configure Basic Device Settings

In Part 1, you will set up the network topology and configure basic settings on the PC hosts, switches, and router.

Step 1: Cable the network as shown in the topology.

The cabling configuration is very similar to the configuration in Lab SU-5a lab.

Disable the Ethernet connection between S4 and R1 by disabling interface Gi0/0/0 on the router.

Use the **show ip interface brief** command to validate your devices are interconnected as shown in the Topology Diagram.

Step 2: Switch configurations.

If you have previously completed Lab SU 5-a, load your saved configurations on S3 and S4 following the below steps for each switch:

- Open the text file containing the running configuration you saved at the end of Lab SU-5a.
- Copy the entire contents of the file by doing **ctrl + a** and then **ctrl + c**.
- On the switch CLI, go to Global configuration mode and paste the configuration by **right-clicking**.
- Do **show run** and compare its output to the configuration on the text file. They should be the same.
- Re-configure Lab SU-5a VLANs on the switches (as they are not saved in the running configuration).

Step 3: (On-campus task) PC hosts configurations.

If you are working on-campus and have previously completed Lab SU 5-a, do not change any configuration of your two virtual PCs. Use the configuration from the end of Lab SU-5a.

Step 4: Initialize and reload the router.

We will be reconfiguring the router, reload the router to get a clean configuration. If you saved the configuration during the previous lab, you will need to remove the startup-config using the **write erase** command before reloading the router.

Step 5: Reconfigure the Management VLAN for each switch.

In Lab SU-5a, we shared the Management VLAN with the Student VLAN as our router only had two Ethernet ports. Having a separate Management VLAN would require a router with three ports to provide full connectivity. This approach scales poorly, if we have 50 VLANs, we need a router with 50 ports to connect them together. The **router-on-a-stick** approach overcomes this limitation.

As we can now support more VLANs, we will move the management VLAN to a new VLAN

a. Move the switch management VLAN from **VLAN10** to **VLAN99**

b. Remove the interface **VLAN10** address configurations on the switch

c. Remove the interface VLAN 10 from the switch using the **no interface vlan** command

d. Enable the interface **VLAN99** address configuration as per the addressing table

e. Confirm that the two switches can successfully ping each other via the configured trunk line. Do not proceed if this does not function.

Step 6: Configure basic settings for the router.

a. Disable DNS lookup.

b. Configure device names as shown in the topology.

c. Don not configure the Ethernet interfaces at this time

Step 7: Configure the loopback interface on the router.

On the router we can create a number of virtual interfaces (called **lo0**, **lo1**, **lo2**, etc).Virtual interfaces are used in lab environments to simulate a subnet or a remote device. In this lab it will provide another network address directly reachable by the router for connectivity testing purposes.

A loopback interface is automatically created when you start configuring it. A loopback interface is automatically enabled and not **shutdown** upon creation.

a. Configure the **lo0** interface:

```
R1(config)# interface lo0
R1(config-if)# description Loopback test interface
R1(config-if)# ip address 209.165.200.225 255.255.255.224
R1(config-if)# end
R1#
```

b. Confirm the loopback interface with the correct subnet mask is configured using the **sh ip route** command.

Part 2: Configure Trunk-Based Inter-VLAN Routing

You will configure R1 to route to multiple VLANs by creating subinterfaces for each VLAN. This method of inter-VLAN routing is called **router-on-a-stick**.

Router-on-a-stick is implemented by configuring 802.1Q trunking on the router and on the switchport connected to the router. All packets for all VLANs are now sent to the router, each packet is tagged so the router can correctly identify which VLAN it belongs to. The router is also able to send packets to specific VLANs by appropriately tagging them, the switches will then forward those packets to the correct devices and ports based on its VLAN configuration.

Step 1: Configure S3 connection to R1 to be a trunk.

a. Change the configuration of Gi1/0/11 on the switch from being an access port on VLAN 10 to a trunk port. Write the command you used in the space provided.

S3(config)#interface gigabitEthernet 1/0/11
S3(config-if)#switchport mode trunk

Step 2: Enable the Ethernet Interface on R1.

When configuring VLAN trunking on a router, the IP addresses are allocated to sub-interfaces of the physical interface. As such, it is important that **NO** IP address is allocated to the Ethernet interface itself. At the interface level we need to enable the interface but **not** configure it.

a. Enable the **Gi0/0/1** interface:

```
R1(config)# interface g0/0/1
R1(config-if)# no shutdown
R1(config-if)# end
R1#
```

b. Confirm that the interface is up on both S3 and the router

Step 3: Configure a sub-interface for the management VLAN.

To properly configure trunking on a router, we need to create one sub-interface for each VLAN the router is connected to over the trunk. For each sub-interface, we need to configure which VLAN it is connected to, and which IP address the router will use on this sub-interface.

a. Create a subinterface on R1 **Gi0/0/1** for VLAN 99 using **99** as the sub-interface ID and assign an interface description.

```
R1(config)# interface g0/0/1.99
R1(config-subif)# description Connection to management VLAN
R1(config-subif)# encapsulation dot1Q 99
R1(config-subif)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# end
```

b. The first command will create (or edit if the sub-interface has already been created) the sub-interface Gi0/0/1.99. Gi0/0/1.99 is a sub-interface of Gi0/0/1. The sub-interface number does **NOT HAVE** to match the VLAN ID, however it is considered good practice to make them the same as it makes configuration easier to read.

c. The second command assigns a sub-interface description

d. The third command configures this sub-interface to be attached to VLAN 99. It is important that the correct number is specified here as this relates to all device configurations.

e. The fourth command assigns the IP address to the sub-interface.

f. Consider the output of **`sh ip route`** and **`sh ip interface brief`**. What do these commands tell you about your configuration

g. Test connectivity. You previously could ping S3 from S4, you should now be able to ping both S3 and S4 from R1, you should also be able to ping 209.165.200.225 from either switch

Step 4: Configure the VLAN10 and VLAN20 sub-interfaces.

Create and correctly configure the sub-interfaces for VLANS 10 and 20 based on the addressing table. Assign interface descriptions to all sub-interfaces. Write the commands you used in the space provided.

VLAN 10

R1(config)#interface gigabitEthernet 0/0/1.10
R1(config-subif)#description Connection to student VLAN
R1(config-subif)#encapsulation dot1Q 10
R1(config-subif)#ip address 192.168.10.1 255.255.255.0

VLAN 20

```
R1(config)#interface gigabitEthernet 0/0/1.20
R1(config-subif)#description Connection to faculty VLAN
R1(config-subif)#encapsulation dot1Q 20
R1(config-subif)#ip address 192.168.20.1 255.255.255.0
```

Verify connectivity.

a. Enter the command to view the routing table on R1. What networks are listed?

Network of interface G0/0/1.99, G0/0/1.10, G0/0/1.20 and network of loopback

b. Connectivity scenarios:

From PC-A, would it possible to ping the default gateway for VLAN 10? Yes

From PC-A, would it possible to ping PC-B? Yes

From PC-A, would it possible to ping Lo0? Yes

From PC-A, would it possible to ping S4? Yes

c. Connectivity tests from R1:

Use the extended options of the ping command to test connectivity between the different networks connected to the router.

Ping S3 in VLAN 99 sourcing the ping from the router interface connected to VLAN 10

```
R1# ping 192.168.1.11 source g0/0/1.10
```

Ping S3 in VLAN 99 sourcing the ping from the router interface connected to VLAN 20

```
R1# ping 192.168.1.11 source g0/0/1.20
```

Ping S3 in VLAN 99 sourcing the ping from the Loopback0 interface in the router

```
R1# ping 192.168.1.11 source lo0
```

Ping S4 from interfaces **Gi0/0/1.10**, **Gi0/0/1.20** and **Lo0** on the router using similar commands.

You can also run ping tests in the other direction, i.e. ping all IP addresses configured on active router interfaces and sub-interfaces from the switches.

If the above ping tests are successful, this is an indication that inter-VLAN routing is working properly in your network. If they are unsuccessful, troubleshoot the configurations and correct any errors.

d. On-campus tests:

From PC-A, is it possible to ping the default gateway for VLAN 10? Yes

From PC-A, is it possible to ping PC-B? Yes

From PC-A, is it possible to ping Lo0? Yes

From PC-A, is it possible to ping S4? Yes

If the answer is **no** to any of these questions, troubleshoot the configurations and correct any errors.

Reflection

What are the advantages of trunk-based or router-on-a-stick inter-VLAN routing?

The main advantages of router-on-a-stick inter-VLAN routing are its affordability and ease of use, especially for smaller networks. Router-on-a-stick lowers hardware costs and streamlines network operations by using a single physical link between the router and the switch. It also provides flexibility in terms of VLAN configurations and effectively makes use of the router's excess resources for activities like access control and QoS enforcement.

Part 3: Clean up

Step 1: Clear device configurations

a. You **MUST** clear the VLAN database using the **delete vlan.dat** command on both your switches

b. If you saved the startup-config, use the **write erase** command to delete it prior to turning off your devices