

Lab 19b1. L3 Switch SVIs

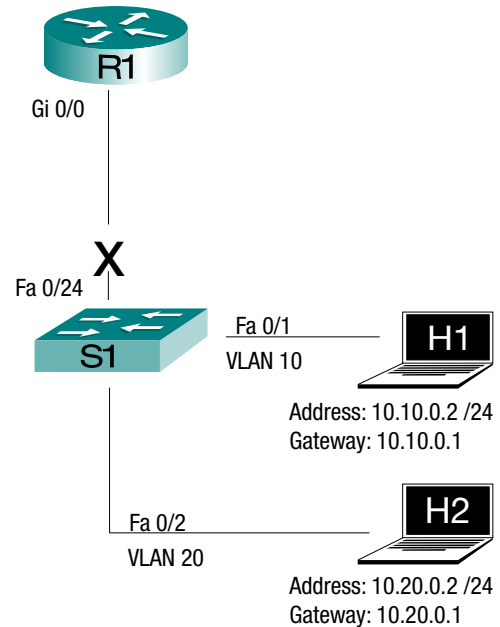
G O A L

Configure and verify layer 3 routing between VLANs on S1.

Topology

For simplicity, you can reuse the physical topology of the previous lab, 19a, and shutdown the port on S1 that leads to the router. You'll be handling that routing within S1. If you're starting from scratch, R1 doesn't even need to exist, and you'll need:

- A switch (S1) of almost any vintage that can handle L3 switching. A 2960 is a fairly normal L2 switch that can have L3 routing enabled. I'm using a 3550 layer-3 switch because it's cheap
- Two hosts (H1 & H2) to test connectivity. I'm actually using a couple of old Cisco routers with default gateways and the "no ip routing" command configured
- 2 Ethernet cables—since we've shutdown the switchport leading to R1, we don't really need that cable
- A way to issue commands over the console ports



C O N F I G U R A T I O N S T E P S

If you haven't already (Lab 19a), wire the topology and give your switch, S1, its boilerplate configurations

- Hostnames per the diagram
- No IP domain-lookup
- Synchronous logging and no exec-timeout on the console

If you haven't already, configure the L2 access aspects of the switch, S1

- Create VLANs 10 and 20
- Define interfaces Fa0/1 and Fa0/2 as access ports, placing Fa0/1 in VLAN 10 and Fa0/2 in 20

Configure the hosts with IP addresses and default gateways and test connectivity between them.

Finally, the point of the whole exercise...

- Configure L3 SVI interfaces for VLAN 10 and VLAN 20
- Enable L3 switching (routing) on S1

Verify connectivity between H1 and H2.

CONFIGURATION WALKTHROUGH

Configure S1 as in Lab 19a—Skip if already done

Give the switch its basic "convenience" configurations.

```
S1
-----
hostname S1
no ip domain-lookup
line con 0
  exec-timeout 0 0
  logging synchronous
-----
```

On the switch, create VLANs 10 and 20, and assign them to switchports Fa0/1 and Fa0/2, respectively. Ensure that those switchports are in access mode, not trunking.

```
S1
-----
1 S1(config)# interf fa0/1
2 S1(config-if)# switchport mode access
3 S1(config-if)# switchport access vlan 10
4 % Access VLAN does not exist. Creating vlan 10
5 S1(config-if)# interf fa0/2
6 S1(config-if)# switchport mode access
7 S1(config-if)# switchport access vlan 20
8 % Access VLAN does not exist. Creating vlan 20
-----
```

Since we don't care about naming the VLANs, we can simply allow the switch to automatically create them when we enable them on switchports [Lines 3 and 7].

Now we shutdown the interface leading to the router (in Lab 19a, this was our trunk)

```
S1
-----
1 S1(config)# interf fa0/24
2 S1(config-if)# shutdown
-----
```

Before leaving the switch, make sure that your access interfaces are in the right VLANs and not shutdown and that the link to the router *is* shutdown.

```
S1
-----
1 S1# show interfaces status
2
3 Port      Name              Status      Vlan      Duplex  Speed Type
4 Fa0/1     Fa0/1             connected   10        a-full  a-100 10/100BaseTX
5 Fa0/2     Fa0/2             notconnect  20        auto    auto  10/100BaseTX
6 ...
7 Fa0/24    Trunk to R1       disabled    1         auto    auto  10/100BaseTX
-----
```

If one of the hosts has a status of "notconnect" [Line 5], that means that it's off or disconnected at the other end of the wire. In this case, I shut down the port of the router that I'm using as a host. A status of "disabled" [Line 7] means that the switch's own interface is administratively shut down. In the case of Fa 0/24, that's what we wanted.

Configure the Hosts—Skip if already done.

This will depend on your host operating system. If you happen to be (mis)using ancient Cisco routers as I am, that configuration looks like this:

H1	H2
<pre> 1 H1(config)# no ip routing 2 H1(config)# ip default-gateway 10.10.0.1 3 H1(config)# interface fa0/0 4 H1(config-if)# ip address 10.10.0.2 255.255.255.0 5 H1(config-if)# no shutdown </pre>	<pre> no ip routing ip default-gateway 10.20.0.1 interface FastEthernet0/0 ip address 10.20.0.2 255.255.255.0 </pre>

And verification looks like this:

H1					
1	H1# show ip interf br				
2	Interface	IP-Address	OK?	Method	Status
3	Protocol				
4	FastEthernet0/0	10.10.0.2	YES	manual	up
5	FastEthernet0/1	unassigned	YES	unset	administratively down
6					
7	H1# show ip route				
8	Default gateway is 10.10.0.1				
9					
10	Host	Gateway	Last Use	Total Uses	Interface
11	ICMP redirect cache is empty				

A ping from H1 to H2 should fail because they're in separate VLANs and subnets, with no routing in between.

Now, you're ready to begin the heart of this lab, creating L3 SVIs (Switched Virtual Interfaces) and using them to forward packets at wire-speed between VLANs 10 and 20 without the packets ever leaving the switch. Essentially, we're giving the switch two things the router had:

- IP addresses on interfaces that the hosts can use as default gateways—in this case the interfaces will be "virtual." we'll put IP addresses on "VLAN" interfaces whose numbers correspond to the VLANs our hosts are using. You've already seen this concept when you placed an IP address on VLAN 1 so you could telnet into a switch.
- The ability to route between subnets—we'll explicitly type the command "ip routing," which was already on by default on the router. Remember, I had to explicitly turn it off on H1 and H2 to make them act like hosts instead of routers

First, we'll create and verify the svIs

S1						
1	S1(config)# interface vlan 10					
2	S1(config-if)# ip address 10.10.0.1 255.255.255.0					
3	S1(config-if)# interf vlan 20					
4	S1(config-if)# ip address 10.20.0.1 255.255.255.0					
5	S1(config-if)# do show ip interface brief					
6	Interface	IP-Address	OK?	Method	Status	Protocol
7	Vlan1	unassigned	YES	NVRAM	administratively down	down
8	Vlan10	10.10.0.1	YES	manual	up	up
9	Vlan20	10.20.0.1	YES	manual	up	up
10	FastEthernet0/1	unassigned	YES	unset	up	up
11	FastEthernet0/2	unassigned	YES	unset	up	up

This shows an interesting distinction. Our svIs [Lines 8 and 9] are L3 virtual interfaces, while our Ethernet interfaces [Lines 10 and 11] are L2 switchports.

Now we can enable L3 routing. But first I'll prove that our switch isn't a router (yet), just so we know what that looks like in case we ever forget to enable routing on an L3 switch.

```
S1
1 S1(config)# do show ip route
2 Default gateway is not set
3
4 Host          Gateway          Last Use      Total Uses   Interface
5 ICMP redirect cache is empty
6               We're still just a Layer 2 switch
7 S1(config)# ip routing
8               Now we're a router
9 S1(config)# do show ip route
10 Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
11          D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
12          N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
13          E1 - OSPF external type 1, E2 - OSPF external type 2
14          i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
15          ia - IS-IS inter area, * - candidate default, U - per-user static route
16          o - ODR, P - periodic downloaded static route
17
18 Gateway of last resort is not set
19
20      10.0.0.0/24 is subnetted, 2 subnets
21 C        10.10.0.0 is directly connected, vlan10
22 C        10.20.0.0 is directly connected, vlan20
```

Notice that our exit interfaces [Lines 21 and 22] are the svifs. We would also need routes to the rest of the world, but I'll leave that for the next lab.

VERIFICATION WALKTHROUGH

Finally, a quick ping shows that I've been making all of this much more complex than the five lines of configuration it really is.

```
R1
1 H1> ping 10.20.0.1
2
3 Type escape sequence to abort.
4 Sending 5, 100-byte ICMP Echos to 10.20.0.1, timeout is 2 seconds:
5 !!!!!
6 Success rate is 100 percent (5/5), round-trip min/avg/max = 1/201/1000 ms
```

To recap, all we did was

- Create two virtual interfaces
- Turn on IP routing

To be clear, a few specifics had to be observed

- Our svif interface names (numbers), "vlan 10" and "vlan 20", had to match the VLANs assigned to the interfaces our hosts were attached to. This is because the way that VLANs subdivide a switch into multiple virtual switches really does work. The ladder has to be in the correct hole for the packet to climb out.
- The IP address subnets assigned to the svifs needed to match the host configurations in the corresponding VLAN, but you knew that.