Lab 19a. Router On A Stick

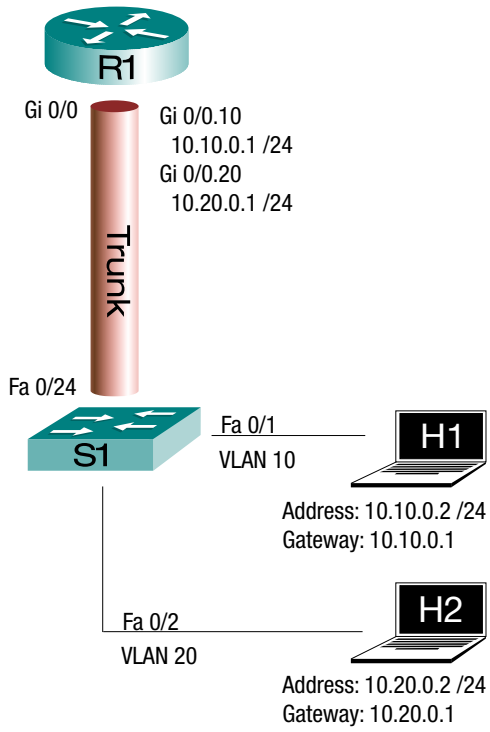
ccnacookbook.com

GOAL

Configure routing on R1, enabling host H1 to communicate with host H2, then make VLAN 10 native on the trunk.

Topology—You'll need

- A router (R1) of almost any vintage. A 2621xm is cheap. I'm using a 2821. Interface names may vary between FastEthernet and GigabitEthernet, depending on your hardware choice
- A switch (S1) of almost any vintage. A 2960 is a fairly normal layer-2 switch. I'm using a 3550 layer-3 switch because it's dirt-cheap and can handle the next two labs without rewiring. The 3550 also requires me to explicitly choose 802.1Q for trunking encapsulation—a good habit
- Two hosts (H1 & H2) to test connectivity. I'm actually using a couple of old Cisco routers because they're conveniently in the same rack for wiring and I can remotely configure them with the same terminal server as everything else
- 3 Ethernet cables—the trunk is a normal cable
- A way to issue commands over the console ports



CONFIGURATION STEPS

Wire the topology and give your devices, R1 and S1, their boilerplate configurations

- Hostnames per the diagram
- Tell the IOS that when it doesn't recognize a command (for example a typo), it shouldn't attempt to contact a DNS server to prepare to telnet to that hostname
- Tell the console port not to log you out after a period of inactivity
- Tell the IOS that if it sends syslog messages to the console port while you're typing there, it should reprint the prompt and whatever you had already typed so that you know where you were

Configure 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
- Create a trunk on interface Fa0/24

Configure the router, R1

- Configure the trunk interface with 802.1Q encapsulation
- Configure subinterfaces for each of the VLANs expected on the trunk

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

Make VLAN 10 the native VLAN

- Modify the configuration of S1 to make VLAN 10 the native VLAN across the trunk to R1
- Modify R1's configuration to cope with the L2 encapsulation change you just made on S1
- Re-test connectivity

VERIFICATION

What routes did you need to add to R1's routing table and why? _____

_____

What information does 802.1Q encapsulation add to traffic over the trunk? _____

_____

How does the encapsulation of a native VLAN differ? _____

_____

Why might you use a native VLAN? _____

_____

_____

CONFIGURATION WALKTHROUGH

Zero Out Your Devices

Make sure that the configurations of both devices really are wiped, including any existing VLANs on the switch. If they boot up asking to run the initial configuration dialog, you know that the startup-config file had been erased from NVRAM. However, there might still be VLANs lurking in the file "vlan.dat" in flash memory. The command "show vlan [brief]" will tell you for sure.

In a more complex topology (one with other switches), shut down the switch ports before deleting vlan.dat to prevent the switch from instantly relearning the VLANs from a neighbor via VTP. Since there's no startup-config, the ports will automatically not be shutdown after the reload.

```
Switch(config)# interface range fa0/1 -24 ,gi0/1 -2
Switch(config-if-range)# shutdown
Switch(config-if-range)# end
Switch# erase startup-config
Switch# delete vlan.dat
```

I keep the commands "Erase" and "Delete" straight by remembering that there's a "D" in both "vlan.Dat" and "Delete"

```
Switch# reload
```

Boilerplate

Give both devices their basic "convenience" configurations.

R1	S1
`Router(config)# ho R1`	`hostname S1`
`R1(config)# no ip domain-lookup`	`no ip domain-lookup`
`R1(config)# line con 0`	`line con 0`
`R1(config-line)# exec-time 0 0`	` exec-timeout 0 0`
`R1(config-line)# logg sync`	` logging synchronous`

Switch Configuration

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 can make the trunk to the router

S1

```
1 S1(config)# interf fa0/24
2 S1(config-if)# description Trunk to R1
3 S1(config-if)# switchport trunk encapsulation dot1q
4                       This is only required on some switches, but is harmless on others.
5                       It's not part of the CCNA, but it's a good habit I try to keep.
6 S1(config-if)# switchport mode trunk
```

Switch Verification

Before leaving the switch, make sure that your interfaces are in the right VLANs and not shutdown.

S1

```
1 S1# show interfaces status
2
3 Port      Name               Status       Vlan       Duplex  Speed Type
4 Fa0/1                        connected    10         a-full  a-100 10/100BaseTX
5 Fa0/2                        connected    20         a-full  a-100 10/100BaseTX
6 …
7 Fa0/24    Trunk to R1        notconnect   1            auto   auto 10/100BaseTX
8                       Don't worry about this "notconect" status. The router port on the other end of the wire is
9                       still shut down. If it said "disabled," then we'd need to do a "no shutdown" here.
```

The notation [Line 7] about fa0/24 being in VLAN 1 looks odd, but since we know it's a trunk, we can get better answers with the command "show interfaces fa0/24 trunk."

S1

```
S1# show interfaces fa0/24 trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/24      on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/24      1-4094

Port        Vlans allowed and active in management domain
Fa0/24      1,10,20

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/24      1,10,20
```

We can also check our access/trunk modes and VLAN assignments for each interface in detail.

S1

```
S1# show interfaces fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 10 (VLAN0010)
Trunking Native Mode VLAN: 1 (default)
…
S1# show interfaces fa0/2 switchport
Name: Fa0/2
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 20 (VLAN0020)
…
S1# show interfaces fa0/24 switchport
Name: Fa0/24
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)
```

This command is filled with half answers that you have to put together yourself. For example:

- Interface fa0/1 is "administratively" configured to be an access port [Line 4] *and* it actually is one, "operationally" [Line 5]
- It's assigned to be in VLAN 10 when it's an access port [Line 9] and it is one [Line 5]
- Its native VLAN will be 1 when it's a trunk [Line 10], but it's not a trunk [Line 5]
- Interface Fa0/24 is configured to be a trunk [Line 25] and it is one [Line 26]

Router Configuration

We'll use subinterfaces to separate frames by VLAN, according to their 802.1Q tags. That way, we can route the enclosed packets between the IP subnets of those subinterfaces.

R1

```
1  R1(config)# interf gi0/0
2  R1(config-if)# description Trunk to S1
3  R1(config-if)# no ip address
4                       Pointless on a router with no pre-existing configuration, but harmless. A good habit to
5                       avoid obscure potential problems with weird symptoms that only show up rarely, if ever.
6  R1(config-if)# interf gi0/0.10
7  R1(config-subif)# encapsulation dot1q 10
8  R1(config-subif)# ip address 10.10.0.1 255.255.255.0
9  R1(config-subif)# no shutdown
10                      A "no shut" on a subinterface is only useful if there's a pre-existing config and it won't
11                      bring up the underlying interface; I'll still have to do a "no shut" on Gi0/0, see line 17
12 R1(config-subif)# interf gi0/0.20
13 R1(config-subif)# encapsulation dot1q 20
14 R1(config-subif)# ip address 10.20.0.1 255.255.255.0
15 R1(config-subif)# no shutdown
16 R1(config-subif)# interf gi0/0
17 R1(config-if)# no shutdown
18                      This was saved for last as a personal preference. I like to finish configuring things before
19                      bringing them online.
```

The interesting thing about trunk subinterface configurations is that the encapsulation statements are on the subinterfaces, rather than declaring the whole interface to use that encapsulation. It helps to remember that the repeated commands also set *which* vlan is on each subinterface.

By the way, I happened to make the subinterface numbers match each VLAN number only for the sake of readability and predictability. They are completely unrelated.

Router Verification

Before leaving the router, check that your (sub)interfaces are all up and in the correct VLANs.

R1

```
1  R1# show ip interf br
2  Interface                  IP-Address      OK? Method Status
3  Protocol
4  GigabitEthernet0/0         unassigned      YES unset  up                    up
5  GigabitEthernet0/0.10      10.10.0.1       YES manual up                    up
6  GigabitEthernet0/0.20      10.20.0.1       YES manual up                    up
7  …
8  R1# show interfaces gi0/0.10
9  GigabitEthernet0/0.10 is up, line protocol is up
10   Hardware is MV96340 Ethernet, address is 001e.1321.e3a8 (bia 001e.1321.e3a8)
11   Internet address is 10.10.0.1/24
12   MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
13      reliability 255/255, txload 1/255, rxload 1/255
14   Encapsulation 802.1Q Virtual LAN, Vlan ID  10.
15   ARP type: ARPA, ARP Timeout 04:00:00
16   Keepalive set (10 sec)
17   Last clearing of "show interface" counters never
```

We use the "show interfaces" command instead of "show ip interface" to see 802.1Q encapsulation and VLAN choices because those are OSI L2 concepts, not L3. I used the "show ip interface brief" command to check everything else simply because it's so much easier to quickly read.

Configure the Hosts

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

```
H1(config)# no ip routing
H1(config)# ip default-gateway 10.10.0.1
H1(config)# interface fa0/0
H1(config-if)# ip address 10.10.0.2 255.255.255.0
H1(config-if)# no shutdown
```

H2

```
no ip routing
ip default-gateway 10.20.0.1
interface FastEthernet0/0
 ip address 10.20.0.2 255.255.255.0
```

And verification looks like this:

H1

```
H1# show ip interf br
Interface                  IP-Address      OK? Method Status
Protocol
FastEthernet0/0            10.10.0.2       YES manual up                    up
FastEthernet0/1            unassigned      YES unset  administratively down down

H1# show ip route
Default gateway is 10.10.0.1

Host               Gateway           Last Use    Total Uses  Interface
ICMP redirect cache is empty
```

VERIFICATION WALKTHROUGH

First, let's see what routes are in R1's routing table

R1

```
R1# show ip route
Codes: L - local, C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, 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, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.10.0.0/24 is directly connected, GigabitEthernet0/0.10
L        10.10.0.1/32 is directly connected, GigabitEthernet0/0.10
C        10.20.0.0/24 is directly connected, GigabitEthernet0/0.20
L        10.20.0.1/32 is directly connected, GigabitEthernet0/0.20
```

All the routes we need are directly connected (C) subinterfaces, so we won't need to add any static routes or run a routing protocol. Our switch is invisible to OSI L3 IP, so it doesn't change the "directly connected" status of those subnets and their hosts.

To really mess with your mind, consider this:

- At OSI L1, our switch is one hunk of steel and silicon
- At L2, it's two switches, one for each of our VLANs (and a third for the VLAN 1, which Cisco's IOS started up as its default)
- At L3, it's a transparent wire

OK, enough. Time for some pinging.

H1

```
H1# ping 10.10.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/201/1000 ms
H1# ping 10.20.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.20.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
H1# ping 10.20.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.20.0.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/4 ms
```

By now, you know to expect a router to drop the first packet while it waits for ARP

Congratulations, you're done unless you want to mess with native VLANs.

NATIVE VLAN ON TRUNK

Native VLANs travel across a trunk untagged—they don't have their L2 Ethernet header expanded to hold a VLAN ID. In order to maintain conceptual consistency and ensure that every frame is in *some* VLAN, a "none of the above" configuration is placed on the device at each end of the trunk. This is called a "native VLAN," and any frame that arrives untagged is placed in that VLAN. Also, frames in that VLAN aren't tagged as they're sent out to cross the trunk.

The goal isn't to avoid tagging some frames during their trip across the trunk, but rather to cope with untagged frames going to and from somewhere else. To understand that, you have to remember that Ethernet is a broadcast medium and the trunk connection between your switch and router might be more than just an Ethernet cable. That Ethernet segment might include a hub with printers, computers, and webcams. Those other devices aren't going to tag their frames and could choke on any tagged frames that are addressed to them.

Note: leaving native VLAN frames untagged is great for theory and Cisco exams, but don't do it in real life; there are security issues.

You Already Have a Native VLAN

One thing that I've ignored in the preceding diagnostic "show" commands is that we actually already have a native VLAN on our trunks. By default, Cisco uses VLAN 1 as the native VLAN on its trunks and CDP has been sending untagged VLAN 1 frames across our trunk ever since we brought it online. Here's a reminder from the switch.

S1

```
S1# show interfaces fa0/24 trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/24      on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/24      1-4094

Port        Vlans allowed and active in management domain
Fa0/24      1,10,20

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/24      1,10,20
```

As for the other end, the router has been receiving and sending VLAN 1 frames using the underlying interface, Gi0/0, while VLANs 10 and 20 have been using the subinterfaces that we created specifically for them.

R1

```
R1# show interfaces Gi 0/0
GigabitEthernet0/0 is up, line protocol is up
  Hardware is MV96340 Ethernet, address is 001e.1321.e3a8 (bia 001e.1321.e3a8)
  Description: Trunk to S1
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID  1., loopback not set
…
R1# show interfaces Gi 0/0.10
GigabitEthernet0/0.10 is up, line protocol is up
  Hardware is MV96340 Ethernet, address is 001e.1321.e3a8 (bia 001e.1321.e3a8)
  Internet address is 10.10.0.1/24
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID  10.
…
```

To prove that VLAN 1 is "native" on the router end, we can use the command "show vlans." This command produces voluminous output organized by VLAN rather than by interface.

R1

```
R1# show vlans 1
                        Leave off the "1" to show all VLANs
Virtual LAN ID:  1 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interface:   GigabitEthernet0/0

 This is configured as native Vlan for the following interface(s) :
GigabitEthernet0/0
                        Traffic counts omitted
```

Handling the native VLAN on the underlying interface is normal Cisco behavior and having VLAN 1 be native is the Cisco default. There's no IP address on the underlying interface, so VLAN 1 traffic can't be routed, but for things like CDP, which operate solely at L2, that's OK.

You may still run across older references that handle the native VLAN at L3 by simply putting an IP address on the underlying router interface, Gi 0/0, but modern best practice is to be more explicit by giving the native VLAN its own subinterface and using the "native" keyword. We'll declare VLAN 10 to be native on the switch:

```
S1(config)# interface fa0/24
S1(config-if)# switchport trunk native vlan 10
```

And on the router:

```
R1(config)# interf Gi0/0.10
R1(config-subif)# encapsulation dot1q 10 native
```

For context, here's a recap of our configs with the additions highlighted

R1

```
interface GigabitEthernet0/0
 description Trunk to S1
 no ip address
!
interface GigabitEthernet0/0.10
 encapsulation dot1Q 10 native
 ip address 10.10.0.1 255.255.255.0
!
interface GigabitEthernet0/0.20
 encapsulation dot1Q 20
 ip address 10.20.0.1 255.255.255.0
```

S1

```
interface FastEthernet0/24
 description Trunk to R1
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 10
 switchport mode trunk
```

We can check that our pings still work

R1

```
H1# ping 10.20.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.20.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/12/20 ms
```

We can check the configuration results on the switch

S1

```
S1# show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/24      on               802.1q         trunking      10

Port        Vlans allowed on trunk
Fa0/24      1-4094

Port        Vlans allowed and active in management domain
Fa0/24      1,10,20

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/24      1,10,20
```

And we can recheck the router.

R1

```
R1# show vlans

Virtual LAN ID:  1 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interface:   GigabitEthernet0/0

   Protocols Configured:   Address:              Received:        Transmitted:
        Other                                          0                439

   131 packets, 26508 bytes input
   439 packets, 33484 bytes output

Virtual LAN ID:  10 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interface:   GigabitEthernet0/0.10

 This is configured as native Vlan for the following interface(s) :
GigabitEthernet0/0

   Protocols Configured:   Address:              Received:        Transmitted:
           IP              10.10.0.1                602244              603639
        Other                                          0                   4

   602294 packets, 841736496 bytes input
   603643 packets, 843760181 bytes output

Virtual LAN ID:  20 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interface:   GigabitEthernet0/0.20

   Protocols Configured:   Address:              Received:        Transmitted:
           IP              10.20.0.1                603820              602067
        Other                                          0                   4

   603820 packets, 845530567 bytes input
   602071 packets, 843005881 bytes output
```

We still have VLAN 1 on the underlying interface (Gi0/0), but it's no longer the trunk's native VLAN.

Mismatched Native VLANs

Since the native VLAN needs to be separately configured on both the router and the switch, it's possible to get one end wrong. That could place frames in the wrong VLAN as they cross the trunk. It could also drop the packets they contain into a subinterface that's in the wrong subnet, breaking your ability to route properly.