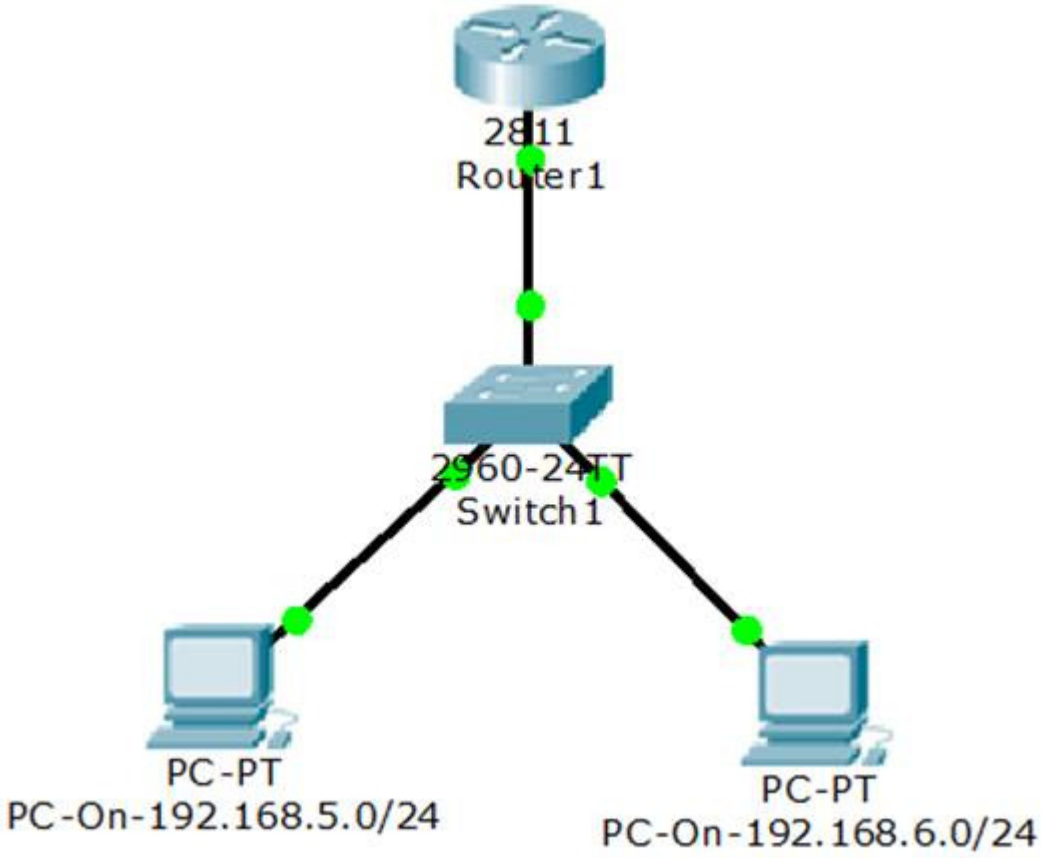
Lab 1 – Router on a stick



BEFORE YOU BEGIN – Clear all router/switch configurations and VLANs, using the Introduction Lab as a guide if needed.

For this lab you will need:

- 2 Subnets (VLANs)
    - VLAN 5 - 192.168.5.0/24
    - VLAN 6 - 192.168.6.0/24
- 3 Routers (2 used for hosts)
- 1 Layer 2 switch

Start with a diagram/drawing. In doing this task, you will find what you need for the lab topology based on your pod diagram by figuring out what equipment and ports are needed.

For this lab, you will need 3 routers that connect directly to the same layer 2 switch. Take the drawing above and write in hostnames and ports you will use to accomplish the task.

Stuck? Ask your instructor or TA for assistance! This part is a crucial skill that you will need throughout the course.

Now that you have your diagram and ports, we can start to configure the devices.

Let's start with the PCs:

First, use 2 routers that connect to 1 switch in your pod. These 2 routers will be used as hosts, and will not do any layer 3 routing**, so the first thing we can do is shut off the routing function with the *"no ip routing"* command.**

Next, we need to make sure we give our host a default gateway, since we have just turned off ip routing. This will allow our "host" to get off its local network/subnet when it wants to. **We will use the *"ip default gateway 192.168.x.1"* command to accomplish this.** As always, simply replace the "x" with the proper subnet number.

Next, let's give our host a name with the *"hostname"* command. Keep it simple. VLAN5HOST will do.

Next, configure the correct interface that connects to the switch. This port will probably be a gigabit0/0, or gigabit 0/1 on the router.

You will want to make sure the following commands are configured:

- *"ip address 192.168.x.5 255.255.255.0"* replace the X with whatever Subnet/VLAN you are configuring.
- A *"no shutdown"* command to place the interface in an active state.

Next, we will need to verify our interface is setup properly.

Do a *"show run interface gigabit x/x"* where the x/x is the correct number for the interface you used.

This command will allow you to verify what you configured for that particular interface. If you would like to see the entire router configuration, simply do a *"show run"*.

Now let's look at the operational state of that particular interface. We can do this with the *"show interface gigabit x/x"* command. Again, substitute the correct interface numbers.

Here is an example output, What we want to focus on is in Red:

HOST2#show int gi 0/0

GigabitEthernet0/0 is up, line protocol is up

Hardware is CN Gigabit Ethernet, address is 1cdf.0fe1.ccb8 (bia 1cdf.0fe1.ccb8)

Internet address is 192.168.6.5/24

MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, loopback not set

Keepalive set (10 sec)

Full-duplex, 1000Mb/s, media type is RJ45

output flow-control is unsupported, input flow-control is unsupported

ARP type: ARPA, ARP Timeout 04:00:00

Last input 00:00:22, output 00:00:00, output hang never

Last clearing of "show interface" counters never

Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0

Queueing strategy: fifo

Output queue: 0/40 (size/max)

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

OUTPUT CONTINUED>>>>Omitted

HOST2#

Note that we see the interface is up/up at layers 1 and 2, has an IP at layer 3, and has negotiated speed and duplex in a proper manor.

Next we move on to the switch configuration:

Configure a hostname for the switch, using the *"hostname"* command.

Now we will configure our VLANs that we will use for our subnets. Remember a VLAN is used to logically separate Subnets from each other without the use of additional hardware and interfaces. We will logically separate our 2 VLANs, 5 and 6, and their associated IP traffic, and allow them to communicate via a single interface on our local router.

Use the global configuration *"vlan"* command to accomplish this. Use the *"show vlan"* command to verify. Here is an example:

```
SWITCH1(config)#vlan 5

SWITCH1(config-vlan)#name VLAN5

SWITCH1(config-vlan)#exit

SWITCH1(config)#vlan 6

SWITCH1(config-vlan)#name VLAN6

SWITCH1(config-vlan)#exit


SWITCH1(config)#do show vlan

VLAN Name                       Status    Ports

---- -------------------------------- --------- -------------------------------

1    default                    active    Gi0/1, Gi0/2, Gi0/3, Gi0/4
                                          Gi0/5, Gi0/6, Gi0/7, Gi0/8
                                          Gi0/9, Gi0/10, Gi0/11, Gi0/12
                                          Gi0/13, Gi0/14, Gi0/15, Gi0/16
                                          Gi0/17, Gi0/18, Gi0/19, Gi0/20
                                          Gi0/21, Gi0/22, Gi0/23, Gi0/24
5    VLAN5                      active
6    VLAN6                      active
1002 fddi-default               act/unsup
1003 token-ring-default         act/unsup
1004 fddinet-default            act/unsup
```

1005 trnet-default act/unsup

OUTPUT CONTINUED>>>>Omitted

Now we need to associate our host ports with the appropriate VLANs. The port that we have connected to the host on VLAN 5, will need to be associated with VLAN 5. The port for the host on VLAN 6, of course, will need to be associated with VLAN 6. We should also verify that our ports are up/up, and are associated properly after configurations with the proper show commands.

Use the interface command "switchport access VLAN x" to associate the port with the VLAN. Substitute the proper VLAN # for x.

Example:

SWITCH1(config)#int gi 0/3

SWITCH1(config-if)#switchport access vlan 5

SWITCH1(config-if)#exit

SWITCH1(config)#

Verify with the "*show interface switchport*" and "*show interface*" commands shown below:

SWITCH1#show interface gigabitEthernet 0/3 switchport

Name: Gi0/3

Switchport: Enabled

Administrative Mode: dynamic auto

Operational Mode: static access

Administrative Trunking Encapsulation: negotiate

Operational Trunking Encapsulation: native

Negotiation of Trunking: On

Access Mode VLAN: 5 (VLAN5)

OUTPUT CONTINUED>>>>Omitted

SWITCH1#show int gi 0/3

GigabitEthernet0/3 is up, line protocol is up (connected)

Hardware is Gigabit Ethernet, address is f866.f2f8.6203 (bia f866.f2f8.6203)

MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, loopback not set

Keepalive set (10 sec)

Full-duplex, 1000Mb/s, media type is 10/100/1000BaseTX

OUTPUT CONTINUED>>>>Omitted

Now that we have our host ports associated with the proper VLANs, we now need to allow this VLAN traffic to be brought up to the router to be "routed".

We will accomplish this by setting up a link called a **trunk**. A trunk will allow for multiple VLANs to traverse the connection to the other device so that it can be processed. In this case, we are allowing VLANs 5 and 6 to traverse the trunk so our router can allow the 2 VLANS to communicate with each other. The frames traversing the link are "tagged" with their VLAN ID, which allows for the traffic to stay separate. It will be the job of the router to make the decisions on whether the VLANs can communicate with each other. This process can involve the use of routing protocols, Access Control Lists, Quality of Services, etc. In this lab, we will use basic "connected routes" to allow these VLANs to communicate.

In order to accomplish this task, we will need to identify the interface that you have chosen to connect to your router, and we will "statically" configure this link as a trunk. We will also only allow for VLANs 5 and 6 to traverse the link. This is done for bandwidth and security reasons, as you will learn later.

We will also set the layer 2 trunking encapsulation type, there are 2 for these switches, ISL (Cisco) and Dot1Q, Dot1Q being an open standard. We will use Dot1Q.

Use the *"switchport mode trunk" and "switchport trunk"* commands to accomplish this task. An example is below:

SWITCH1(config)#int gi 0/1

SWITCH1(config-if)#switchport trunk encapsulation dot1q

```
SWITCH1(config-if)#switchport mode trunk

SWITCH1(config-if)#switchport trunk allowed vlan 5,6

SWITCH1(config-if)#exit
```

Verify our trunk configurations using your *"show interface" and "show interface switchport"* commands as you did above for the access ports (PCs).

Also, verify your trunk using the *"show interfaces trunk"* command. Example:

```
SWITCH1#show int trunk

Port      Mode         Encapsulation  Status       Native vlan
Gi0/1     on           802.1q         trunking     1

Port      Vlans allowed on trunk
Gi0/1     5-6

Port      Vlans allowed and active in management domain
Gi0/1     5-6

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/1     none
SWITCH1#
```

Don't see a trunk? Check you router to make sure the interface is at least up/up. Do a no shut if needed, and verify again. Also, check your interface configuration using the *"show run interface"* command.

At this point we will move on to the router. The switch configuration is complete.

Router configuration:

We will need to configure our router to accept frames over our trunk for both VLANs 5 and 6. We will also need to configure the Gigabit interface for both VLANs, which will allow for us to route between these VLANs. Essentially, we are configuring the default gateways for both of our subnets.

Let's begin:

First, configure a **hostname** for the router, see above for commands to accomplish this.

Next, identify the interface on the router you used for the trunk to the switch. The first command you should do is a **no shutdown,** if you have not done so already, allowing for layer 1 and 2 connectivity to the switch.

Verify this with your ***"show interface"*** command that you used above, remember, you are looking for an up/up state on the interface.

Now let's configure the trunk and layer 3 IP commands:

We will need to use **"'Sub-Interfaces"** in order to use one physical interface to represent two virtual interfaces. Here is an example of how to configure a "sub-interface":

```
ROUTER1(config-if)#int gi 0/0.5
```

Note the use of the .5 on the end of the interface gig0/0 command. This is a **logical** separation of the interface, and we are choosing .5 because this will be used for VLAN 5. This is an arbitrary number…..however we will use 5 for documentation and clarity.

Next we will need to **set the encapsulation type**, and the VLAN 5 used for this sub-interface. This number is **NOT arbitrary**, and needs to match the exact VLAN used on the trunk.

Example:

```
ROUTER1(config-subif)#encapsulation dot1Q 5
```

Next we will need to set the default **gateway IP address** for our hosts. This will be the assigned IP for this sub-interface.

Here is an example for VLAN 5.

```
ROUTER1(config-subif)#ip address 192.168.5.1 255.255.255.0
```

Perform this same configuration for both VLANs. Use the .6 sub-interface for VLAN 6.

Verify your configurations using the show interface commands above, but let's also add a new command, the *"show interface brief"* command:

```
ROUTER1#show ip interface brief
```

```
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     unassigned      YES unset  up                    up
GigabitEthernet0/0.5   192.168.5.1     YES manual up                    up
GigabitEthernet0/0.6   192.168.6.1     YES manual up                    up
GigabitEthernet0/1     unassigned      YES unset  administratively down down
Serial0/0/0            unassigned      YES unset  administratively down down
Serial0/0/1            unassigned      YES unset  administratively down down
ROUTER1#
```

Note with this command we see that the main Gig interface is up/up, but has no IP address assigned. We also see that our 2 sub-interfaces have the IPs, and they are up/up as well.

At this point, we can start to verify our connectivity!

Lab connectivity verification:

We can now begin the process of verifying our configurations. First, do a double check of your configurations and make sure everything appears as you configured it using your *"show run"* commands.

Next, let's check our hosts. First, ping their gateway to see if they can reach the router.

```
HOST2#ping 192.168.6.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.6.1, timeout is 2 seconds:
..!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
HOST2#
```

Did you lose a couple of pings at first? This is usually due to the time it takes to do the first ARP lookups.

Can't ping your gateway??? Start you troubleshooting by **choosing a method**. (*Top down, bottom up, divide and conquer, etc*) **Next follow the packet!!!!** Remember we will leave the NIC on our host, then switch through to the router, and then we follow the same path back to the PC. Check things like port status, trunk status, VLAN assignments, etc.

If the ping to the gateway is successful, check to see if you can ping the other host on the different subnet. VLAN 5 to VLAN 6. This will verify that routing is working as well. We should now be able to

take a packet from VLAN 5, destined for VLAN 6, and the router will use its internal routing table to forward the packet.

```
HOST1#ping 192.168.6.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.6.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

HOST1#
```

Also, we can verify the router has the right routes with a new command, a *"show ip route"*. Perform this command on your router. Note these are listed as **CONNECTED** routes.

```
ROUTER1#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, + - replicated route


Gateway of last resort is not set


   192.168.5.0/24 is variably subnetted, 2 subnets, 2 masks

C      192.168.5.0/24 is directly connected, GigabitEthernet0/0.5

L      192.168.5.1/32 is directly connected, GigabitEthernet0/0.5

   192.168.6.0/24 is variably subnetted, 2 subnets, 2 masks

C      192.168.6.0/24 is directly connected, GigabitEthernet0/0.6
```

L 192.168.6.1/32 is directly connected, GigabitEthernet0/0.6

ROUTER1#

At this point you have successfully completed Lab1. Save all configurations as you will need them for upcoming labs!