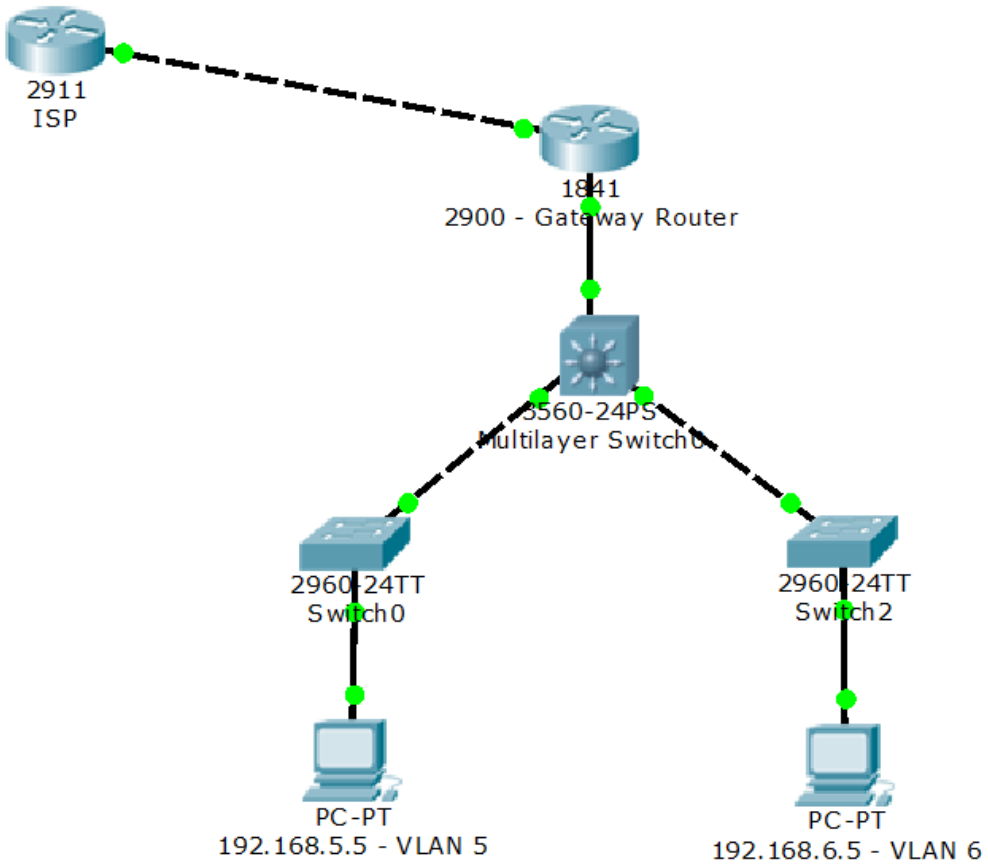
Lab 2 – Basic Network – Small Office



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:

- 3 Internal Subnets (VLANs)
    - VLAN 5 - 192.168.5.0/24
    - VLAN 6 - 192.168.6.0/24
    - VLAN 10 – 192.168.10.0/24 – used for switch and router management.
- 3 External Subnets
    - WAN subnet – 200.200.200.0/30
    - Internet IPs leased from provider – 201.201.201.0/30

    - Internet IPs assigned to ACME.com – 202.202.202.1/32 – test IP

- 4 Routers (2 used for hosts, 1 gateway router, 1 ISP router)
    - ***You will need to use routers 4 and 5 for the ISP to Gateway connection, using a serial interface***
- 3 Layer 2 switches
- Configurations from Lab 1

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.

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

Important! You may need to adjust your Lab 1 configurations to accommodate your new topology.

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

Lab 2 preparation:

Lab 1 is a pre-requisite to this lab. You will need to start with the basic configs from lab 1. Edit and re-paste the configurations you used, or start with Lab 1 from the beginning and continue to Lab 2 when finished.

Lab 2 instructions for our new setup:

LAN Section:

Expansion of the access and distribution layers in the switching environment

Initially, we want to expand our network to add a distribution layer into our layer 2 environment. This will allow us to expand our network with flexibility, reliability, and scalability. Access layer switches will connect to the distribution layer switch, and in later labs we will move our routing between our VLANs to this distribution layer.

Using the knowledge you obtained in lab 1, **connect your 2 access layer switches to the new distribution layer switch** using trunks. These trunks should only carry traffic for your 2 data VLANs, plus the new management VLAN, VLAN 10.

Add configurations to your VLANs to name them. VLAN 5 will be used for Systems Administrators, VLAN 6 will be used for Network Administrators, and VLAN 10 will be used for Network Management.

Your switches and routers will also need an IP address for your new Network Management VLAN, which we will add later. For now, define the VLAN and add it to the trunks until needed.

You will also need to move your original trunk from lab 1 to the new distribution layer switch. The router and the switch will connect as a "router of a stick" like you configured in Lab 1. Remember, this trunk will need to be modified to allow for the new VLAN.

Use the *"switchport trunk allowed VLAN add"* command to accomplish this. The keyword "add" will allow for VLAN 10 to be added, without removing VLANs 5 and 6. Without the add part of the command, only the new VLAN would be allowed, and 5 and 6 would be removed.

At this point, perform a *"show int trunk"* command to verify that all trunks are operational on the 3 switches, and we are only allowing the 3 VLANs as designed. Below is a sample of the output of this command. Your output will vary:

```
VCC-CORE#show int trunk
Port        Mode         Encapsulation  Status        Native vlan
Fa0/24      on           802.1q         trunking      1
Gig0/1      on           802.1q         trunking      1
Gig0/2      on           802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/24      10,20
Gig0/1      10
Gig0/2      20

Port        Vlans allowed and active in management domain
Fa0/24      10,20
Gig0/1      10
Gig0/2      20

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/24      10,20
Gig0/1      10
Gig0/2      20
VCC-CORE#
```

After verification of layer 2, we are ready to add some management IP addresses to our network.

Management VLAN IP addressing:

Now we will need to add IP addressing to our management network.

First, we will add the new virtual interface to the router. **As in Lab 1, add a new sub-interface to the gateway router on the Ethernet interface attached to the trunk.** Remember, the sub-interface will need an encapsulation type, the VLAN to be routed, and the IP address for the VLAN. Use the .1 of the

subnet for this interface. This new interface will serve as the management IP of the router, as well as the default gateway for the switch management interfaces your about to configure.

Now that we have defined the router interface and IP address, let's add IP addresses for management to our switches.

By default, VLAN 1 is used for management for the switches. This could be seen as a security issue, and so we will not use VLAN 1 for that purpose. Instead, we will shutdown the VLAN 1 interface of the switches, and create a new virtual interface for VLAN 10 to use for management.

First, shutdown VLAN 1:

```
LALLY(config)#int vlan 1
LALLY(config-if)#shut
LALLY(config-if)#end
LALLY#
```

Now create a new VLAN interface on the switches, give your switches IP addresses of .2, .3, and .4:

```
LALLY(config)#int vlan 10
LALLY(config-if)#
%LINK-5-CHANGED: Interface Vlan10, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to up

LALLY(config-if)#ip add 192.168.10.2 255.255.255.0
```

Note that the virtual interface comes up/up as soon as it is created. This is due to it being a virtual interface.

Verify that you can ping from your router to the IP addresses of your switches.

```
LALLY#ping 192.168.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/4/19 ms

LALLY#
```

At this point on your switches you will not be able to ping past the Gateway IP address. Can you guess why?

Add a default gateway to your switches. This will allow you to have connectivity past the management subnet. You will want to get from the hosts to the switches later in the lab.

Verify that all interfaces you have configured at this point can ping each other. If you are able to ping from both hosts to the new switch IP addresses, you should be able to manage those devices later on.

Once you have finished the access/distribution section, continue to the DCHP section.

DHCP:

In Lab 1 we setup our PCs with static IP addresses, in which we put an IP address and a default gateway on our "hosts". At this point in designing our network we want to assign any hosts a dynamic IP address via DHCP. In order to do this, we will need to setup DHCP pools on our routers, and set our hosts to look to the DHCP server for assignment.

The following configuration is an example of a DHCP pool. We will need one of these for both of our subnets, VLAN 5 and 6.

Note the exclude addresses. These are IPs that are "Excluded" from the pool, usually for static assignment, possibly for servers, network gear, etc. Your IP assignments for hosts should start at the .5 IP of the subnet.

Here you see we have a DHCP pool, named VLAN10-POOL, with a /24 size subnet. We are excluding the .1-.20 IPs from the scope. We setup the default gateway for the VLANs to the .1 of the subnet. We are setting the DNS server as the gateway as well, since the router can be used to serve DNS in some cases.

```
ip dhcp excluded-address 192.168.10.1 192.168.10.20
!
ip dhcp pool VLAN10-POOL
 network 192.168.10.0 255.255.255.0
 default-router 192.168.10.1
 dns-server 192.168.10.1
```

Configure 2 pools on the router that connects to your PC environment to support DHCP for the 2 subnets.

Now that we have configured the pools, we need to setup our hosts to get a DHCP address assigned to them. In Lab 1, we configured these as static addresses.

To accomplish this task, we will use the "ip address DHCP" command on the Ethernet interfaces of our 2 hosts. The following is an example:

```
HOST1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
HOST1(config)#int gi0/0
HOST1(config-if)#no ip add
HOST1(config-if)#ip add dhcp
HOST1(config-if)#end
```

Perform this command on both hosts. Verify it worked by using your "show interface" command:

HOST1#
Log to screen and logging buffer:

*Sep 9 00:03:30.025: %DHCP-6-ADDRESS_ASSIGN: Interface GigabitEthernet0/0 assigned DHCP address 192.168.5.11, mask 255.255.255.0, hostname HOST1

HOST1#**show int gi 0/0**
GigabitEthernet0/0 is up, line protocol is up
Hardware is CN Gigabit Ethernet, address is 588d.0904.bbb0 (bia 588d.0904.bbb0)
Internet address is 192.168.5.11/24
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 00:00:03, output 00:00:05, 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
1627393 packets input, 462294727 bytes, 0 no buffer
Received 1627377 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog, 27320 multicast, 0 pause input
0 input packets with dribble condition detected
210608 packets output, 45951188 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets
310490 unknown protocol drops
0 babbles, 0 late collision, 0 deferred
6 lost carrier, 0 no carrier, 0 pause output
0 output buffer failures, 0 output buffers swapped out
HOST1#

You can now check the binding on the gateway router using the *"show ip dhcp bindings"* command:

GATEWAY#show ip dhcp binding

IP address Client-ID/ Lease expiration Type

Hardware address

192.168.5.10 0007.EC94.5BC3 -- Automatic

GATEWAY#

WAN Section:

ISP Connection and IP Routes:

Now that we have established LAN connections, we will shift our focus to configuring our connections to our ISP.

First, we have to connect our Gateway router to the ISP router. This should be a direct connection to the provider VIA a serial interface (Simulation of a T1). In order to do this we will need to use routers 4 and 5 as stated above, however, if you are up for a challenge, you can also accomplish this via frame relay. If you like very complex puzzles, use the command reference guide and take up a new challenge! (If you don't choose to config this now, no sweat! Plenty of frame relay to come!)

Configure the serial interface with the following:

- Ip addressing
- Clock Rate on the proper interface
- No shutdown

Example:

```
Gateway #conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Gateway (config)#int s0/0/0
Gateway (config-if)#ip add 200.200.200.2 255.255.255.252
Gateway (config-if)#clock rate 8000000
Gateway (config-if)#no shut
Gateway (config-if)#
```

You will need to configure both sides of the connection. **Verify connectivity with a simple ping.**

```
Gateway#ping 200.200.200.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.200.200.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/4/19 ms

Gateway #
```

Now we will need to setup some static routing.

The majority of the time, connections from a small to medium size organization to their ISP consists of 1 connection; one way into the network, one way out of the network.

This is easy for us to accomplish using static routes. One of which we will call a "default" static route. This route will belong to the customer side of the connection. It's basic, and states, "anything I don't know, send it over the WAN connection."

The command we will need is ***"ip route 0.0.0.0 0.0.0.0** ISP IP"* where ISP IP is the IP address of the OTHER side of our WAN connection.

Example:

ip route 0.0.0.0 0.0.0.0 200.200.200.1

Configure the previous command on your gateway router.

Note that we have been assigned a set of "real world" IP addresses from our provider. This will allow users and servers on our LAN to get connectivity to the internet. Of course, we are using private addressing for our LAN, so we will need to use NAT (Network Address Translation) to use these "real world", internet routable, IPs. We will configure NAT later in the lab.

For now, we will need to configure the network to be routed to our Gateway router on the ISP router. In order to accomplish this task, we will use a static route. Not default of course, but still a static route.

A static route on the ISP router for our leased network will look like this:

ip route 201.201.201.0 255.255.255.252 200.200.200.2

NOTE that the "next hop" IP this time **is our WAN interface**. This statement states "if you want to get to 201.201.201.201.0/30 subnet, go over to 200.200.200.2."

Verify our new commands via the "show ip route" command:

```
GATEWAY#show ip route
Codes: C - connected, S - static, I - IGRP, 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, E - EGP
       i - IS-IS, 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

Gateway of last resort is 200.200.200.1 to network 0.0.0.0

C    192.168.5.0/24 is directly connected, FastEthernet0/0.5
C    192.168.6.0/24 is directly connected, FastEthernet0/0.6
     200.200.200.0/30 is subnetted, 1 subnets
C       200.200.200.0 is directly connected, FastEthernet0/1
```

S* 0.0.0.0/0 [1/0] via 200.200.200.1
GATEWAY#

ISP#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, E - EGP
i - IS-IS, 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

Gateway of last resort is not set

S 201.201.201.0/27 [1/0] via 200.200.200.2
ISP#

While we are configuring the "real world" IPs, we need to create some interfaces on the Gateway and ISP routers for 2 reasons:

- NAT – this interface will be used on the Gateway router for our internal hosts to use for NAT
- Internet simulation – we will configure an IP on the ISP router to simulate "acme.com"

We will accomplish these tasks using loopback interfaces.

Configure a loopback on our gateway router for our "real world" routable IPs using the *"interface loopback"* command as follows:

Example:

interface Loopback0
ip address 201.201.201.1 255.255.255.224

Note again that the interface is listed as up/up as soon as it is created. This is due to the fact that this loopback interface is a virtual interface, and is always up.

Configure a loopback on our ISP router for our "real world" routable IP for "acme.com (testing site) using the *"interface loopback"* command. This loopback will have an IP address of 202.202.202.1/32.

Example:

interface Loopback0
ip address 202.202.202.1 255.255.255.255

Verify our configurations using the *"show ip int brief"* command.

At this point, our Gateway router should know how to get to the "acme.com" IP. This is due to the default static route we have created on our Gateway router. **Perform a ping from our Gateway router to the IP address of "acme.com", 202.202.202.1.**

GATEWAY#ping 202.202.202.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 202.202.202.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms

GATEWAY#

How about our hosts? They do not have a way to get to ACME. That is where NAT comes in.

NAT Section:

In this final section we will configure NAT on the Gateway router.

Nat is established in 4 steps:

- Define the Internal Addresses
- Define the External Address(es)
- Define the translation(s)
- Define the interfaces (inside/outside)

That's it. Simple, right?

Keep in mind we will configure all these steps on the Gateway router.

Step 1 – define the internal space

We will use an access list to define the internal IP space. If you read the access list for what we are trying to accomplish, it will read, permit the LAN IPs used by our users to translate to real world IPs. Our gateway router will keep track of these "translations".

To configure a standard access list to accomplish this task, we will use the following commands:

access-list 1 permit 192.168.5.0 0.0.0.255
access-list 1 permit 192.168.6.0 0.0.0.255

Note the use of what is called a "wild card" mask. This mask reads "match the first 3 parts of the subnet, the last part can be anything…", make sense?

We will use this access list in the translation.

Step 2 – define the external space
Remember above when we created our loopback interface for 201.201.201.1 255.255.255.252? We will use this interface for NAT translations. We have one additional IP in that subnet, but we will reserve it for servers if need be. For this definition, we will use an IP Pool. This pool will consist of 1 IP address, but may contain more in other configurations:

ip nat pool POOL 201.201.201.1 201.201.201.1 netmask 255.255.255.252

Reading this line, we are stating, "use IPs in the range of 201.201.201.1 to 201.201.201.1 (1 IP), to translate internal addresses. Our pool name has a very simple name, "POOL". We could also simply define the interface to use, for instance, we could just say "use loopback 0" for the translation. You will need to define a pool on most exams.

Step 3 – define the translation

In this step, we will combine the internal and external spaces with a translation. There are a few types of translations we can do. They can be defined in two categories, static and dynamic. This will be a dynamic translation.

The following configuration will allow our access list IPs to use the external addresses to gain access to the internet:

ip nat inside source list 1 pool POOL overload

Note the use of the "overload" keyword. This allows us to use a "one to many" translation. We are using 1 outside IP address to NAT multiple internal addresses. The router will use port numbers to keep the translations separate.

Step 4 – define the interfaces

We will need to define what interfaces we will use for NAT. We will have "INSIDE" and "OUTSIDE" interfaces. Inside interfaces are those that contain the internal IPs needed to be translated, our hosts. The external interface is the one used by the external IPs, in this case, the Loopback interface.

Use the following commands to define the internal and external interfaces:

```
interface gi0/0.5
 ip nat inside
!
interface gi0/0.6
 ip nat inside
!
interface serial 0/0/1
 ip nat outside
```

These commands will allow for our internal IPs (VLANs 5 and 6) to use NAT. We will define the serial interface as the interface for the outside traffic.

Be sure to use the interfaces that you need for your physical infrastructure, as they may be different than those configured above. (FastEthernet vs. Gigabit., etc)

Verification:

At this point, we should verify that our hosts have connectivity to the outside world, specifically 202.202.202.1, acme's IP address.

Perform a ping from both host machines to 202.202.202.1. If successful, you will also see translations for DHCP in the Gateway router, with the *"show IP NAT translations"* command. Translation tables will look like the following:

```
GATEWAY#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 201.201.201.2:5   192.168.5.5:5      201.201.201.2:5    201.201.201.2:5
icmp 201.201.201.2:6   192.168.5.5:6      201.201.201.2:6    201.201.201.2:6
icmp 201.201.201.2:7   192.168.5.5:7      201.201.201.2:7    201.201.201.2:7
icmp 201.201.201.2:8   192.168.5.5:8      201.201.201.2:8    201.201.201.2:8

GATEWAY#
```

Not seeing expected results? Use a troubleshooting methodology discussed in class, or asks a TA or instructor for assistance!

At this point you have completed lab 2, save your configurations for Lab 3!