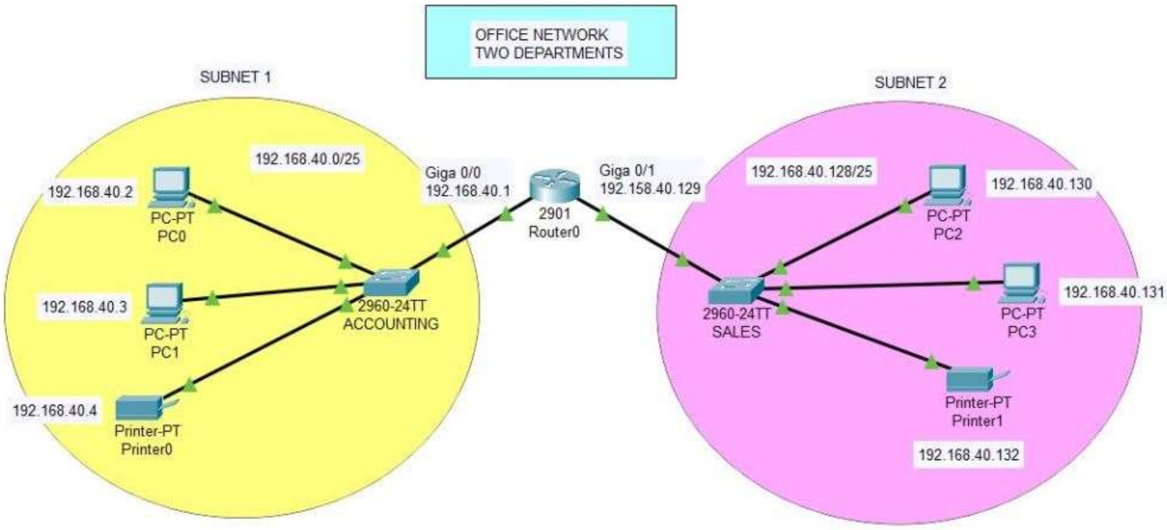
Subnetting a Small Office Network in Cisco Packet Tracer via Static IP Assignment



As an exercise, I decided to build a small office network with two departments – one for accounting and one for sales. This would be a small representation of a company so we will pretend it's merely one arm of a network inside a larger building. For my topology, each subnetwork had the following:

2 PC's
1 printer
1 switch

Connecting both subnets together would be:
1 router
Simple enough, right? Let's hope so.

Calculations

Calculation: number of bits for subnet
I started by assigning a network address for the entire network. I chose a Class C private address of 192.168.40.0.

The first question I had to answer was how many subnet bits we would have to borrow from the network to devote to the subnet. The formula to calculate this is:

2^n = Number of Subnets

The *n* value above represents the number of subnet bits – as in, the number of bits needed to create the subnet. We already know we want two subnets so we can set the right side of the equation to 2, so now the calculation would be:
2^n = 2

In other words, 2 raised to what power equals 2? In this case, *n* = 1.
2^1 (subnet bits) = 2 (number of subnets)

So now we know that we will need to borrow exactly 1 bit from the network to create the subnetwork.

Calculation: Subnet mask
Both IP addresses and subnet mask addresses are 32 bits total, consisting of four octets (4 segments of 8 bits).
For our class C address, the default subnet mask is 255.255.255.0. Now we convert this to its binary form in bits of 0 and 1.

255.255.255.0 converted to binary would be:
11111111.11111111.11111111.00000000

We already know that we need one of these bits to create the subnet. So we will convert the first bit of the last octet to 1. In binary form this would be:
11111111.11111111.11111111.10000000

Now we convert this binary form of the address to decimal format to get our subnet mask.
11111111 in binary form converts to decimal as 255.

10000000 in binary form converts to decimal as 128.

Therefore our subnet mask will be 255.255.255.128.

The subnet mask 255.255.255.128 corresponds to a CIDR notation of /25. This means that 25 bits of the IP address are used for the network, leaving 7 bits for host addresses (25 bits + 7 bits = 32 bits total for the address).

Calculation: block size
Block size, or the range of consecutive IP addresses in a subnet, is determined by the number of host bits. Since there are 7 host bits available in this subnet mask, the block size can be calculated as 2^7, which equals 128.

So, in a subnet with a subnet mask of 255.255.255.128 (/25), the block size is 128. This means there are 128 addresses in each subnet, with 126 addresses available for hosts (after accounting for the network and broadcast addresses).

Calculation: Range of valid host addresses & broadcast address
We have our network ID and our subnet mask. Now we need two more items – the range of valid hosts on the subnet and the broadcast address.

Subnet mask = 255.255.255.128
Network address = 192.168.40.0

To calculate the first host address, we add 1 to the network address above. This gives us:
First host address = 192.168.40.1

To get the last usable host address, we subtract 1 from the broadcast address. This gives us 192.168.40.127.

Last host address = 192.168.40.126

To calculate the Broadcast ID, we note that the block size is 128. Add this value to the network address and then subtract 1, which gives us 127. This is added onto the last part of the address giving us:

Broadcast ID = 192.168.40.127

Putting this all together, we have calculated for the first subnet:
Network address = 192.168.40.0
Subnet mask = 255.255.255.128
Range of valid hosts = 192.168.40.1 – 192.168.40.126
Broadcast ID = 192.168.40.127

Calculations for subnet 2:
Using the same process outlined above, we calculate the same values for the 2nd subnet.
To get the network address for the second subnet, we add block size of 128 to the first network address of 192.168.40.0 which means the next block starts at 192.168.40.128:
Network address = 192.168.40.128
The subnet mask is the same as the first subnet:
Subnet Mask = 255.255.255.128

To get the first host address, we add 1 again to the network address, giving us: 192.168.40.129.

First host address = 192.168.40.129

The last usable host on the subnet would be 1 less than the broadcast ID, giving us 192.168.50.254 as the last usable host address.

Last host address = 192.168.50.254

To get the broadcast ID for subnet 2, we note that the block size is 128. To find the broadcast address, add the block size minus 1 (127 in this case) to the network address, giving us:

Broadcast ID = 192.168.40.255

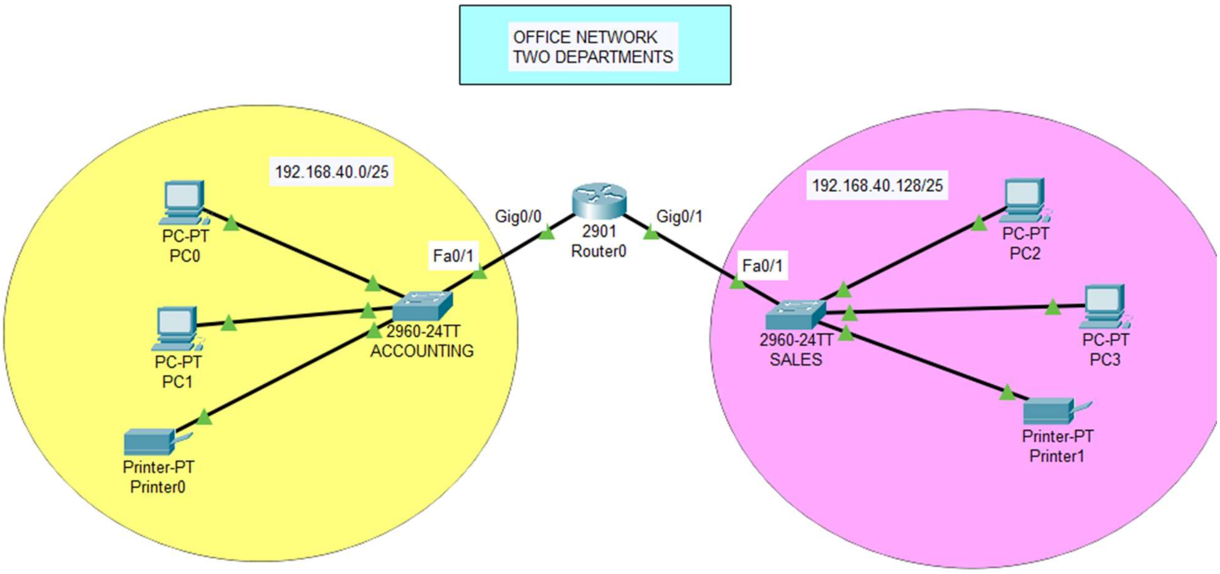
Putting all of this together again, we have calculated for the subnet 1 and 2:

Subnet 1:
Network address = 192.168.40.0
Subnet mask = 255.255.255.128 (/25 in CIDR notation)
Range of valid hosts = 192.168.40.1 – 192.168.40.126
Broadcast ID = 192.168.40.127

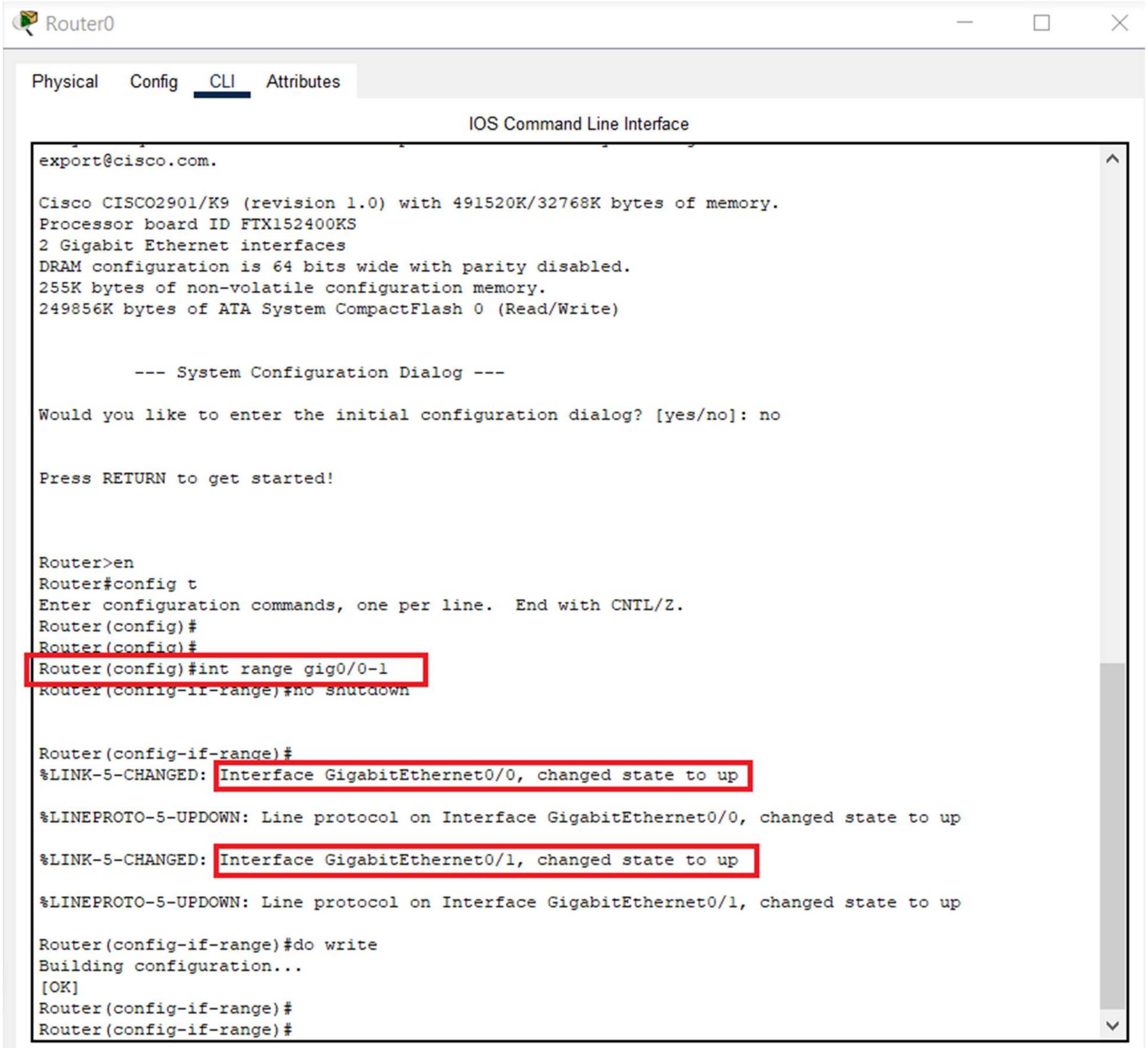
Subnet 2:
Network address = 192.168.40.128
Subnet mask = 255.255.255.128 (/25 in CIDR notation)
Range of valid hosts = 192.168.40.129 – 192.168.40.254
Broadcast ID = 192.168.40.255

Network device configuration

After subnetting, I went into Cisco Packet Tracer and configured the devices.
The router in the center has two interfaces attached to it: gig0/0 and gig0/1.



Clicking into the router and accessing the CLI for the router, I enabled both interfaces:

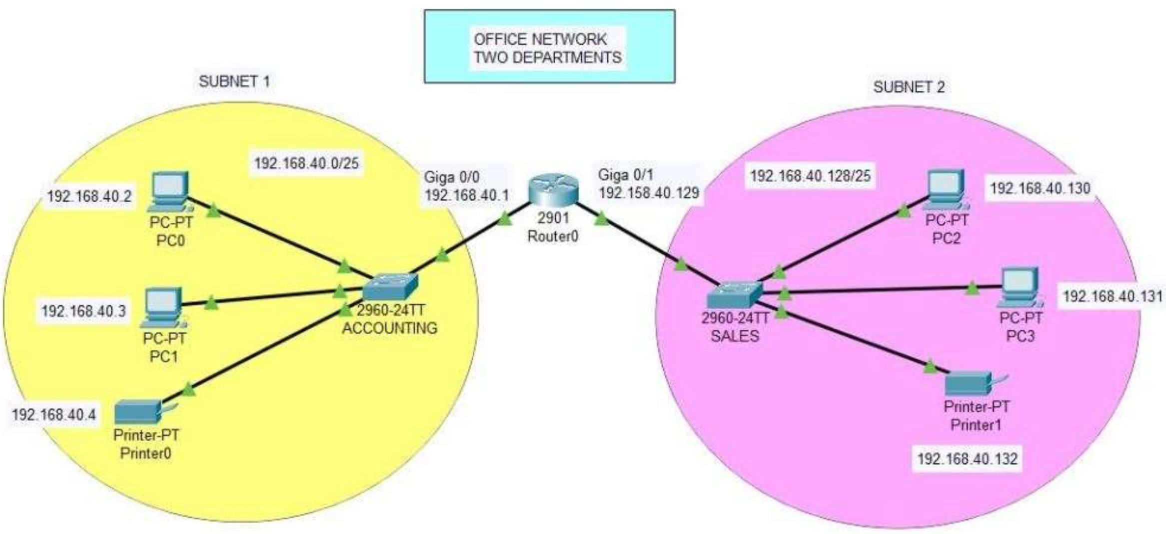


First, I assigned the IP address for the router. Remember the range of valid hosts for subnet 1 is 192.168.40.1 – 192.168.40.126. I assigned the first address in the series to interface GigabitEthernet0/0 – 192.168.40.1.

Then I assigned an IP address to interface GigabitEthernet0/1 – remember, the range of valid hosts on subnet 2 is 192.168.40.129 – 192.168.40.254. I assigned the first address in the series to interface GigabitEthernet0/1 – 192.168.40.129.

```
interface GigabitEthernet0/0
 ip address 192.168.40.1 255.255.255.128
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 192.168.40.129 255.255.255.128
 duplex auto
 speed auto
!
interface Vlan1
 no ip address
 shutdown
```

Configuration of IP addresses on Router interfaces 0/0 and 0/1.



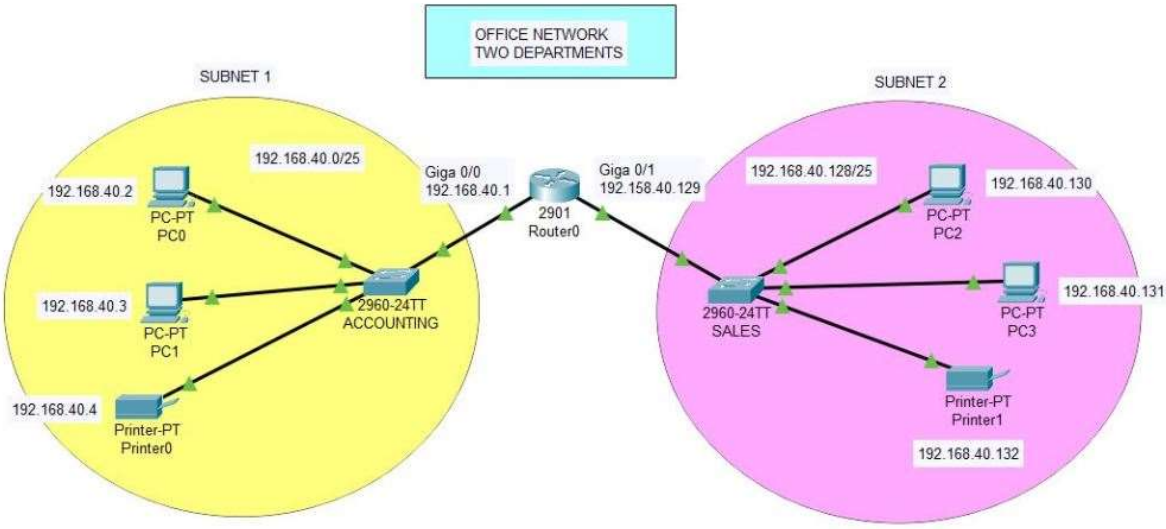
The router now acts as a default gateway for both subnets.

Next, it was time to assign IP's to the PCs and printers on both subnets.
The range of valid host addresses for subnet 1 is 192.168.40.1 – 192.168.40.126. The first in the range, 192.168.40.1, has now been assigned to the router interface (Gig 0/0) So I simply used the next two in the series to assign to the PCs:

192.168.40.2 for PC0
192.168.40.3 for PC1
192.168.40.4 for Printer0

Now onto the devices on subnet 2.

The range of valid host addresses for subnet 2 is 192.168.40.129 – 192.168.40.254. The first in the range, 192.168.40.129, has been assigned to the router's second interface (Gig 0/1). So we can use the next two in the series to assign to the PCs:
192.168.40.130 for PC2
192.168.40.131 for PC3
192.168.40.132 for Printer1.

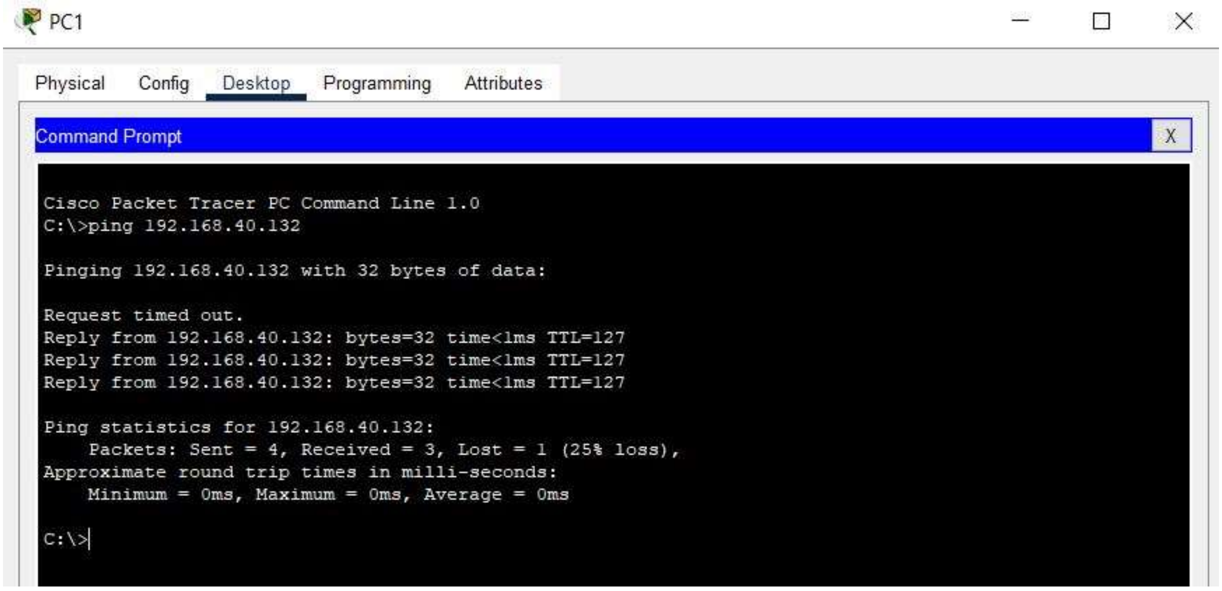


Now that everything was been configured, I tested communication between devices.

Going into Command Line on PC0 (on subnet 1), I pinged PC2 on subnet 2:

```
C:\>ping 192.168.40.130

Pinging 192.168.40.130 with 32 bytes of data:

Reply from 192.168.40.130: bytes=32 time<1ms TTL=127
Reply from 192.168.40.130: bytes=32 time<1ms TTL=127
Reply from 192.168.40.130: bytes=32 time<1ms TTL=127
Reply from 192.168.40.130: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.40.130:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

All packets returned, signaling successful communication and connection between devices.
I also pinged from PC1 (on subnet 1) to printer1 (on subnet 2):



Although a packet got lost, the communication overall was intact.

Although in most real-life set ups DHCP servers would automatically assign IP addresses and you would not assign static IPs, this practice in calculating the subnet mask, broadcast address, and usable host addresses was useful in understanding the mechanics behind subnetting. Will detail more as I continue learning the tool and in the future, simulate larger networks and protocols.