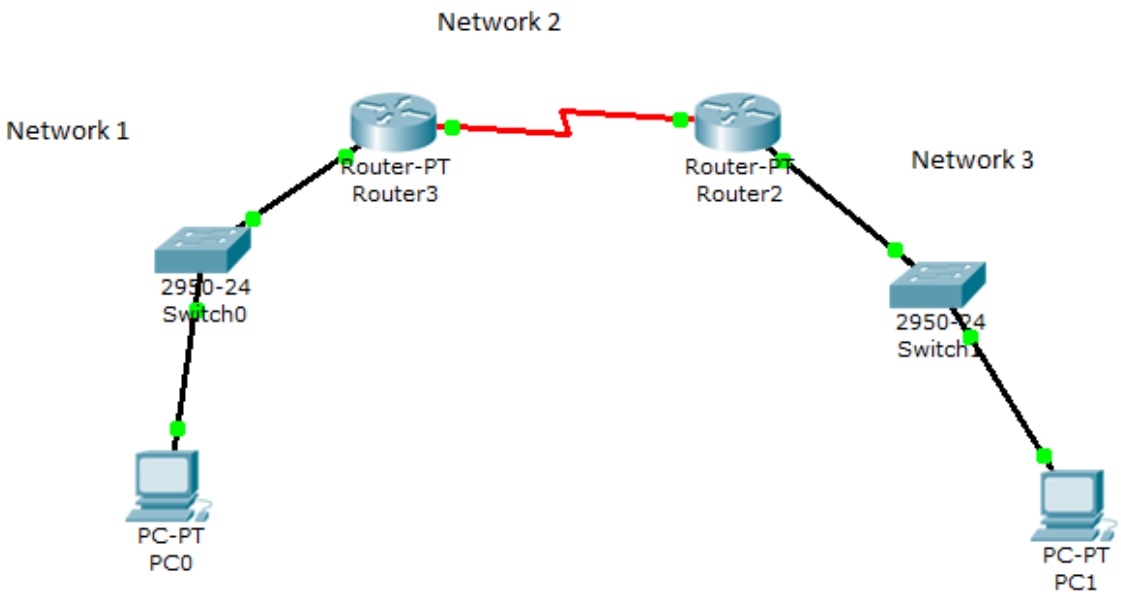
Static Routing

Static Routing involves manually configuring routers to tell them where a destination network is.



The objective of this lab is to successfully configure the topology shown, with a static route being set to inform network 1 of network 3 and vice versa.

The first objective is to set the IP addresses on each PC and enable them to speak to the routers on their respective networks. This is done by setting an IP on the fa 0/0 ports of the routers. After doing so we can see that PC 0 (IP 192.168.1.1) can communicate with Router3 (IP 192.168.1.2) and also that PC1 (IP 192.168.3.1) can communicate with Router2 (IP 192.168.3.3

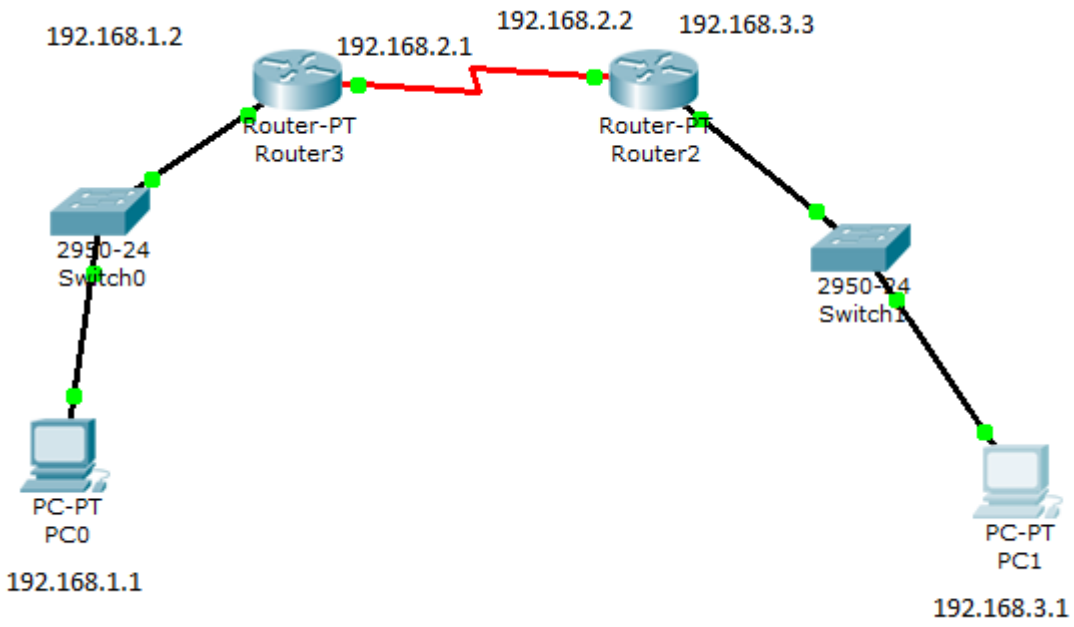
```
PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=19ms TTL=255
Reply from 192.168.1.2: bytes=32 time=11ms TTL=255
Reply from 192.168.1.2: bytes=32 time=8ms TTL=255
Reply from 192.168.1.2: bytes=32 time=8ms TTL=255

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 8ms, Maximum = 19ms, Average = 11ms
```

)

```
PC>ping 192.168.3.3

Pinging 192.168.3.3 with 32 bytes of data:

Reply from 192.168.3.3: bytes=32 time=19ms TTL=255
Reply from 192.168.3.3: bytes=32 time=10ms TTL=255
Reply from 192.168.3.3: bytes=32 time=6ms TTL=255
Reply from 192.168.3.3: bytes=32 time=2ms TTL=255

Ping statistics for 192.168.3.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 19ms, Average = 9ms
```

Now a static route must be created so the router connected to Network 1 knows where network 3 is, and the router connected to Network 3 knows where Network 1 is. The purpose of setting up this static route is so when devices on different networks try communicate, the router checks its routing table and knows which hop to send a packet to in order to get it to the desired network. By manually configuring the static routes (using the ip route command), we tell the router to send and packets destined for network X, to the hop with ip address Y, instead of allowing the router to drop the packet.

After assigning ip addresses to the ports, the topology now looks like this:



A DCE cable now connects router 3 to router2. A clock rate of 64000 has been set. The next step is to make sure that Router 3 (connecting networks 1-2) is informed of network 3. To do this, I accessed the router and manually configured it to send any packets destined for network 3, to the next hop on router2 (IP 192.168.2.2).

```
Enter configuration commands, one per line.  End with CNTL/Z
R1(config)#ip route 192.168.3.0 255.255.255.0 192.168.2.2
R1(config)#no shutdown
```

In the screenshot above, it basically tells the router, if you receive any packets destined for the network “192.168.3.0” send them to the next hop, 192.168.2.2.

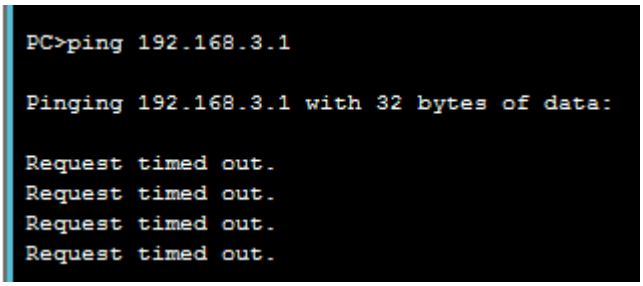
I then did the same for the other router.

Now, the two routers are able to speak to each other. A static route has been connected, and when PC0 tries to ping PC1, the connection is successful.

```
PC>ping 192.168.3.1

Pinging 192.168.3.1 with 32 bytes of data:

Reply from 192.168.3.1: bytes=32 time=20ms TTL=126
Reply from 192.168.3.1: bytes=32 time=21ms TTL=126
Reply from 192.168.3.1: bytes=32 time=26ms TTL=126
Reply from 192.168.3.1: bytes=32 time=14ms TTL=126

Ping statistics for 192.168.3.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 26ms, Average = 20ms
```

I did encounter problems doing this, although they were miniscule, and mostly minor mistakes. Originally after I had completed the topology, the ping from PC0 to PC1 was unsuccessful as I had

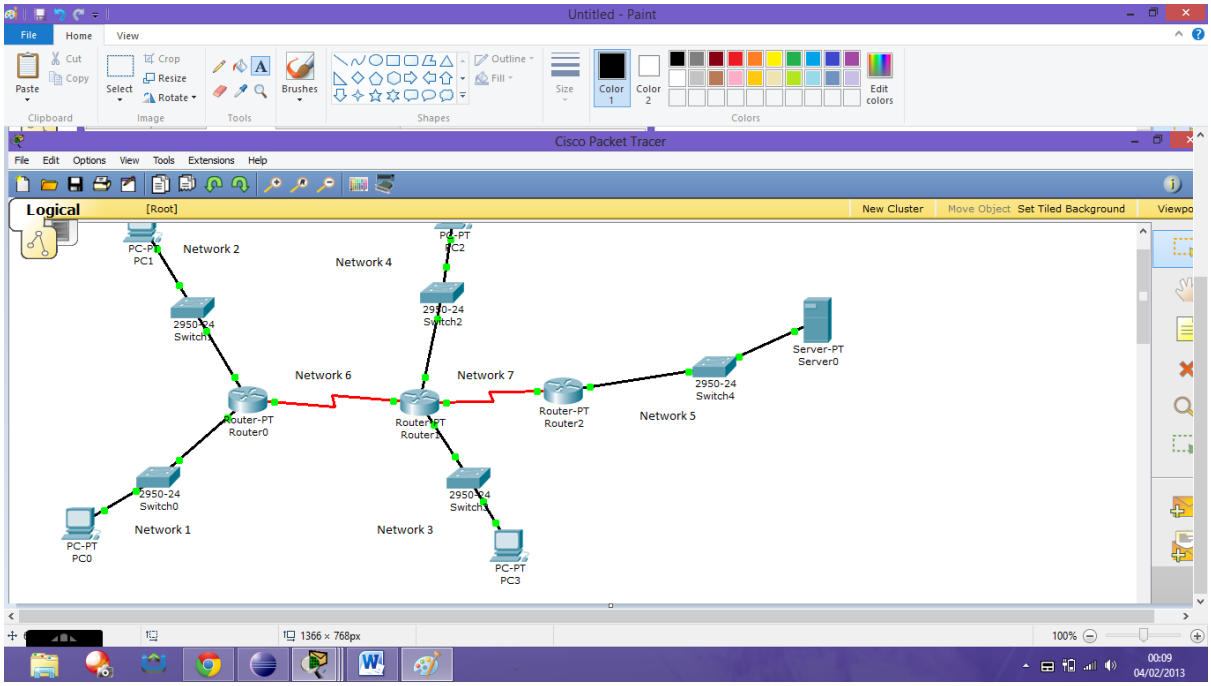
actually assigned the fa 0/0 port on router2(connecting networks 2-3) the same IP address as PC1. Obviously this meant the packets never reached PC1 on network 3.



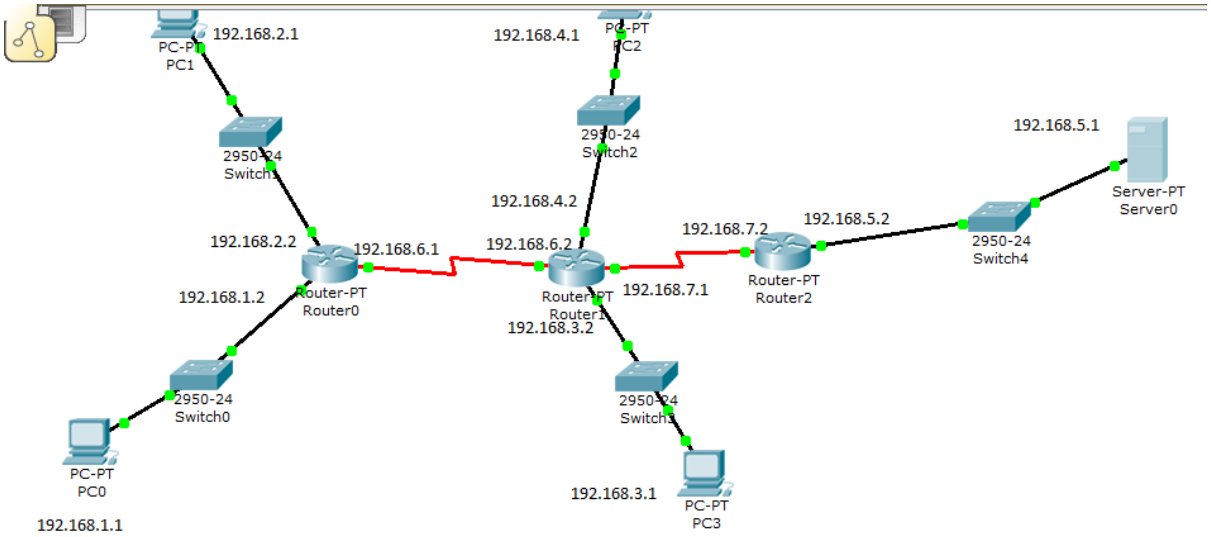
I also made the mistake of attempting to ping PC1 from PCO without assigning the default gateways for both PC's resulting in a similar problem.

Other than that, no real problems were encountered.

Larger network involving static routing



In order to further demonstrate my understanding of how static routes function, I decided to construct a topology consisting of 7 different networks, including 3 routers and a web server.



As before, I configured each end device with an IP address and remembered to assign a default gateway to all. As you can see from the diagram, in order for PC's on networks 1+2 to communicate with end devices on networks 4, 5, and 6, static routes had to be configured so router 2 would forward the packets on to the next hop which was on network 6 and had an IP address of 192.168.6.2. Static routes would also have to be configured on the other routers (1+2) in order to communicate back to networks 1+2.

After configuring all the static routes on router0 (connected to networks 1+2), its routing table is as follows:

```
C    192.168.1.0/24 is directly connected, FastEthernet1/0
C    192.168.2.0/24 is directly connected, FastEthernet0/0
S    192.168.3.0/24 [1/0] via 192.168.6.2
S    192.168.4.0/24 [1/0] via 192.168.6.2
S    192.168.5.0/24 [1/0] via 192.168.6.2
C    192.168.6.0/24 is directly connected, Serial2/0
R1#
```

We can see from the routing table that all there are 3 static routes after being manually configured, and so the PC's on networks 1+2 will in fact be successfully able to ping the other end devices, assuming that other routers along the path are also statically configured to know where networks 1+2 are.

Here are the routing tables for Routers 1+2 respectively:

```
S    192.168.1.0/24 [1/0] via 192.168.6.1
S    192.168.2.0/24 [1/0] via 192.168.6.1
C    192.168.3.0/24 is directly connected, FastEthernet1/0
C    192.168.4.0/24 is directly connected, FastEthernet0/0
C    192.168.6.0/24 is directly connected, Serial2/0
C    192.168.7.0/24 is directly connected, Serial3/0
Router#
```

```
S    192.168.1.0/24 [1/0] via 192.168.7.1
S    192.168.2.0/24 [1/0] via 192.168.7.1
S    192.168.3.0/24 [1/0] via 192.168.7.1
S    192.168.4.0/24 [1/0] via 192.168.7.1
C    192.168.5.0/24 is directly connected, FastEthernet0/0
C    192.168.7.0/24 is directly connected, Serial2/0
r5#
```

I can understand why it is best to only use static routes on small networks. As you can see from the screenshot below, I attempted to incorrectly assign a static route, by asking the router to forward packets to itself.

```
r5(config)#
r5(config)#ip route 192.168.1.0 255.255.255.0 192.168.7.2
%Invalid next hop address (it's this router)
r5(config)#ip route 192.168.1.0 255.255.255.0 192.168.7.1
r5(config)#ip route 192.168.2.0 255.255.255.0 192.168.7.1
r5(config)#ip route 192.168.3.0 255.255.255.0 192.168.7.1
r5(config)#ip route 192.168.4.0 255.255.255.0 192.168.7.1
r5(config)#exit
```

So to ultimately test the construction of the topology, I pinged the web server (IP 192.168.5.1) from PC1, and the ping was successful.

```
PC>ping 192.168.5.1

Pinging 192.168.5.1 with 32 bytes of data:

Reply from 192.168.5.1: bytes=32 time=25ms TTL=125
Reply from 192.168.5.1: bytes=32 time=16ms TTL=125
Reply from 192.168.5.1: bytes=32 time=23ms TTL=125
Reply from 192.168.5.1: bytes=32 time=23ms TTL=125

Ping statistics for 192.168.5.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

The server can also successfully ping any other PC on the network. Here the server successfully pings PC2 on network 2.

```
SERVER>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time=27ms TTL=125
Reply from 192.168.2.1: bytes=32 time=23ms TTL=125
Reply from 192.168.2.1: bytes=32 time=23ms TTL=125
Reply from 192.168.2.1: bytes=32 time=27ms TTL=125

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 23ms, Maximum = 27ms, Average = 25ms
```

So the topology shown is successful due to the configuration of static routes between routers.

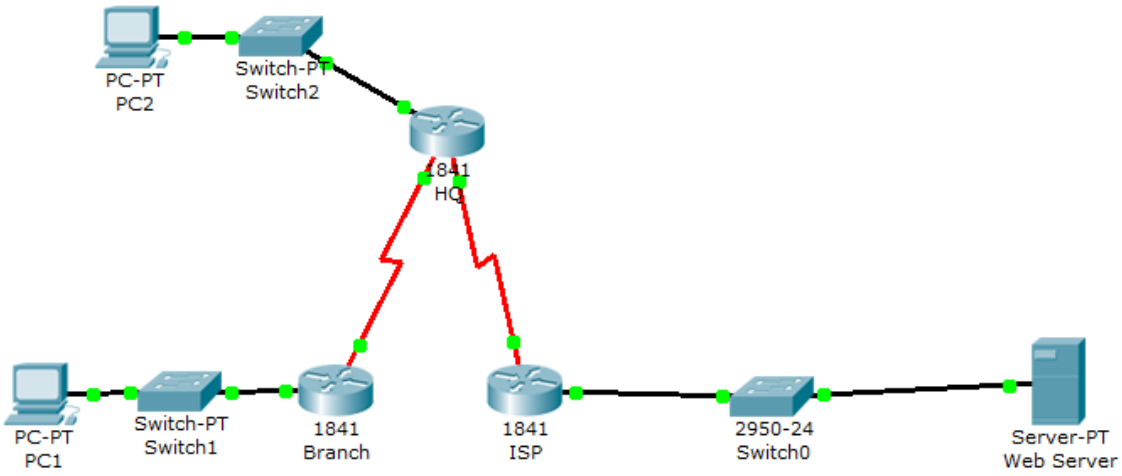
Additional notes: I noticed that when I pinged a PC from the server for the first time, even though the topology was successfully configured, the ping would time out, and then automatically receive replies, but often one packet was dropped and I am still not entirely sure why this happens. I have included a screenshot of the first time I attempted to ping PC2 from the server.

```
SERVER>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.1: bytes=32 time=17ms TTL=125
Reply from 192.168.2.1: bytes=32 time=21ms TTL=125
Reply from 192.168.2.1: bytes=32 time=20ms TTL=125

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 17ms, Maximum = 21ms, Average = 19ms
```

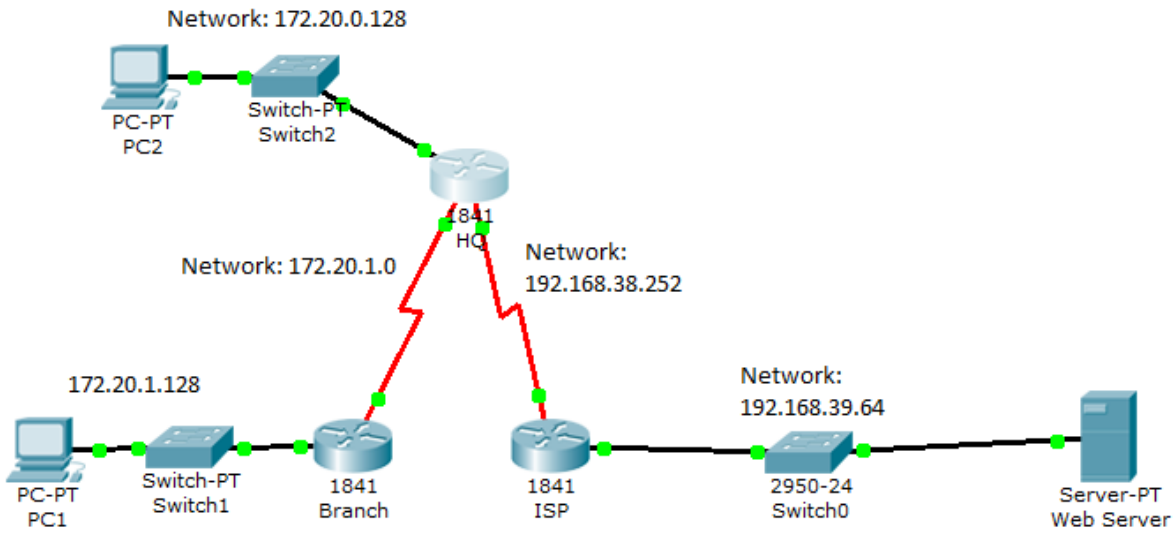
Lab 2: Static Route Configuration.

This lab consisted of a number of errors we had to identify and then fix in order to get the network to function.

I had a number of problems on this lab, but ultimately solved them to create the functioning network seen below.



The first thing I did was determine the network addresses of each network in the topology, so when it was time to statically configure the routes, the addresses would be ready and the it would also be easier to cross reference the spec provided. This also made incorrect route network addresses easier to spot in the spec.



First, I configured the PC's, setting the correct IP's, subnets, and default gateways

In the spec for the router named “Branch”, the incorrect next hop IP is given. It is listed as 172.20.0.129 however as this relates to the serial 0/0/0 port on HQ, the next hop IP should be 172.20.1.2.

```
     172.20.0.0/25 is subnetted, 2 subnets
C       172.20.1.0 is directly connected, Serial0/0/0
C       172.20.1.128 is directly connected, FastEthernet0/0
     192.168.38.0/30 is subnetted, 1 subnets
S       192.168.38.252 [1/0] via 172.20.1.2
     192.168.39.0/26 is subnetted, 1 subnets
S       192.168.39.64 [1/0] via 172.20.1.2
S*   0.0.0.0/0 [1/0] via 172.20.1.2
```

The figure above shows the static routes I created for the “Branch” router. As it is only connected to the HQ router, all the static routes configured inform it to pass packets to HQ’s serial 0/0/0 port.

There were a number of errors encountered while configuring the HQ router. I had to change the IP route as it was not set to the correct address. PC1 is on the network 172.20.1.128, and there was no IP route to inform the HQ router of which hop to send packets to if they were destined for that network. In the spec, the network address was incorrectly listed as 172.20.0.0. Also in the spec, no clock rate was set on HQ for the serial connection 0/0/1.

```
!
interface Serial0/0/1
 ip address 192.168.38.254 255.255.255.252
 no shutdown
!
```

This is incorrect as this port has a DCE connection, and the clock rate must be set at this end. I added a clock rate of 64000. The correct static route configuration for HQ is below:

```
     172.20.0.0/25 is subnetted, 3 subnets
C       172.20.0.128 is directly connected, FastEthernet0/0
C       172.20.1.0 is directly connected, Serial0/0/0
S       172.20.1.128 [1/0] via 172.20.1.1
     192.168.38.0/30 is subnetted, 1 subnets
C       192.168.38.252 is directly connected, Serial0/0/1
     192.168.39.0/26 is subnetted, 1 subnets
S       192.168.39.64 [1/0] via 192.168.38.253
HQ#
```

From the image above, we can see that HQ can communicate with any network. It is already directly connected to 3 networks, and if it wants to communicate with the network containing PC1, it knows to forward packets to serial 0/0/0 172.20.1.0, or if it wants to communicate with the server, it sends packets to 192.168.38.253 which is the serial 0/0/1 port on ISP.

On the ISP router, the IP route was once again incorrectly listed. There is a “no shutdown” command missing on the fa0/0 port. The spec also specifies to assign 192.168.38.253 255.255.255.252 to serial port 0/0/0 when it should be serial port 0/0/1, as that is the only serial port in use (connected to HQ). After static routes are configured for ISP, the route table is as follows:

```
     172.20.0.0/16 is variably subnetted, 2 subnets, 2 masks
S       172.20.0.128/25 [1/0] via 192.168.38.254
S       172.20.1.128/26 [1/0] via 192.168.38.254
     192.168.38.0/30 is subnetted, 1 subnets
C       192.168.38.252 is directly connected, Serial0/0/1
     192.168.39.0/26 is subnetted, 1 subnets
C       192.168.39.64 is directly connected, FastEthernet0/0
```

From this, we can see it is directly connected to the network with the server(192.168.39.64), and the network linking ISP to HQ (192.168.38.252).If any packet destined for PC1or PC2 reaches ISP, it it statically configured to send the packets to the serial 0/0/1 port on HQ.

The topology functions correctly, and it is possible to ping any device from any other device. Here a ping is successfully send from PC1 to the Server, taking advantage of two statically configured routes.

```
PC>ping 192.168.39.70

Pinging 192.168.39.70 with 32 bytes of data:

Reply from 192.168.39.70: bytes=32 time=18ms TTL=126
Reply from 192.168.39.70: bytes=32 time=17ms TTL=126
Reply from 192.168.39.70: bytes=32 time=18ms TTL=126
Reply from 192.168.39.70: bytes=32 time=21ms TTL=126

Ping statistics for 192.168.39.70:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 17ms, Maximum = 21ms, Average = 18ms
```

Conclusion

I learned the purpose of static routing, and also understand that it should only be used for small networks, as manually configuring static routes for a large network would be very difficult and time consuming. I also have an understanding of a DCE cable, and its purpose. I now know that the most common clock rate used is 64000 and that it is only necessary to configure the clock rate on the router that has the DCE end. I was reminded of the process of configuring a router, and also noticed that I forgot small details like adding banners to the routers, which for the purpose of this exercise is not important, but in general would be something not to be overlooked. I was reminded not to rush the lab and overlook small details like setting an incorrect IP address, or forgetting to assign a default gateway to an end device.

My experience doing this lab was not without frustration. I did encounter problems. Not every ping was successful, and I was often left questioning why.

```
PC>ping 191.168.38.253

Pinging 191.168.38.253 with 32 bytes of data:

Reply from 172.20.0.129: Destination host unreachable.
Reply from 172.20.0.129: Destination host unreachable.
Reply from 172.20.0.129: Destination host unreachable.
Reply from 172.20.0.129: Destination host unreachable.

Ping statistics for 191.168.38.253:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

From my experience doing this lab, I would not rely on copying and pasting code as it is far too easy to overlook small details, which because they are so miniscule (serial 0/0/1 instead of 0/0/0) I found it was very time consuming and frustrating trying to pinpoint these problems. I also think that had we not been given a spec full of bugs, this lab would have been much easier to complete, although I understand that a bug filled spec is perhaps a lesson in its own.

Some stupid errors I made are as follows:

Set an IP to 172.20.1.11 instead of 172.20.1.1

Tried to change an IP route without first using the “no IP” command to remove the unwanted route.

Assigned the wrong default gateway for the server.

I pinged PC1 from PC2 early on in the configuration process, and assumed the ping was successful because I glanced and say four replies, when in fact they were four “Destination host unreachable” responses.

There were many more.

Feedback: In general I enjoyed doing this lab. I found that the experience of doing this exercise in the labs was more confusing, than in packet tracer, perhaps because I am unfamiliar with the different wires, and visually it seems very off putting. After reading back on this report, I noticed that I did not include many screenshots of me assigning IP's to fa & serial ports, or of me setting clock rates. The reason I did this is because I did not want to regurgitate step by step labs. I would appreciate feedback of whether you would like me to include these steps in future.