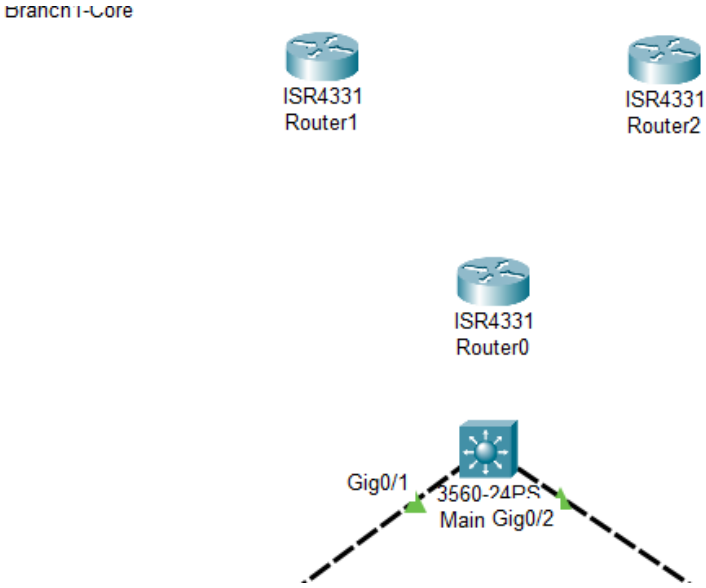
IPv4 Routing

In this lab, we are going to create static IPv4 routes between the thee sites. Creating a static route is essentially giving a router instructions on where it needs to send a specific packet to in order for it to reach its destination.

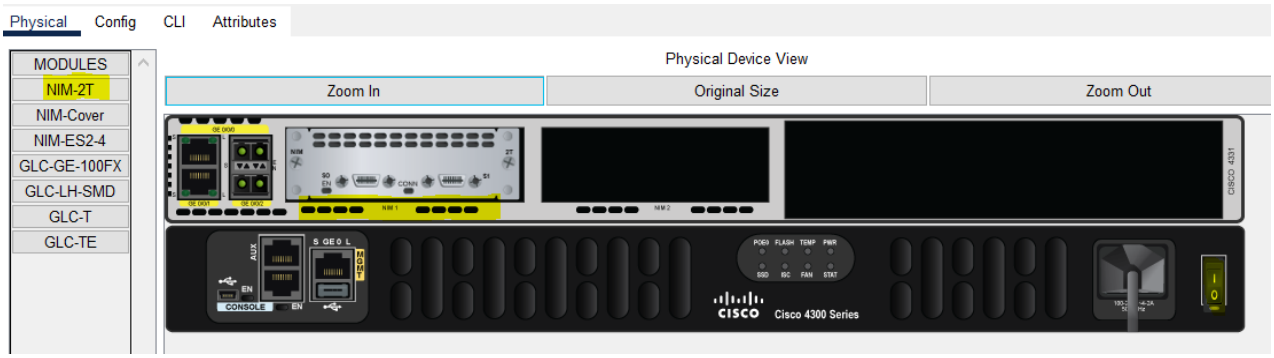
Create a copy of the completed Packet Tracer file for Lab 3 as you will use it for this lab.

First, we are going to add a router at each of our sites. Drag a ISR4331 router to each of the three locations.



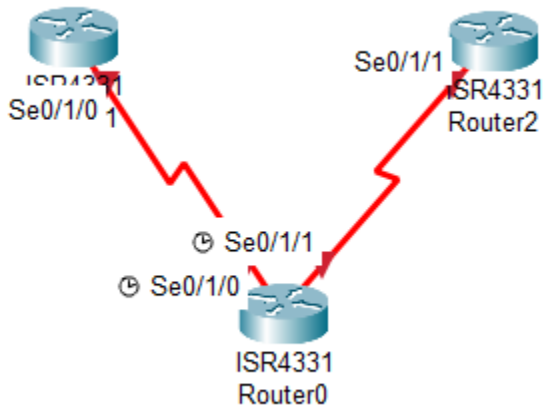
Next, we need to add a serial interface to each of our routers to connect them to one another.

- Click on the core router and click on the ‘physical’ tab.
- Power off the router by clicking on the power switch on the router.
- On the left of the device menu, click and drag the NIM-2T device to the leftmost slot on the router



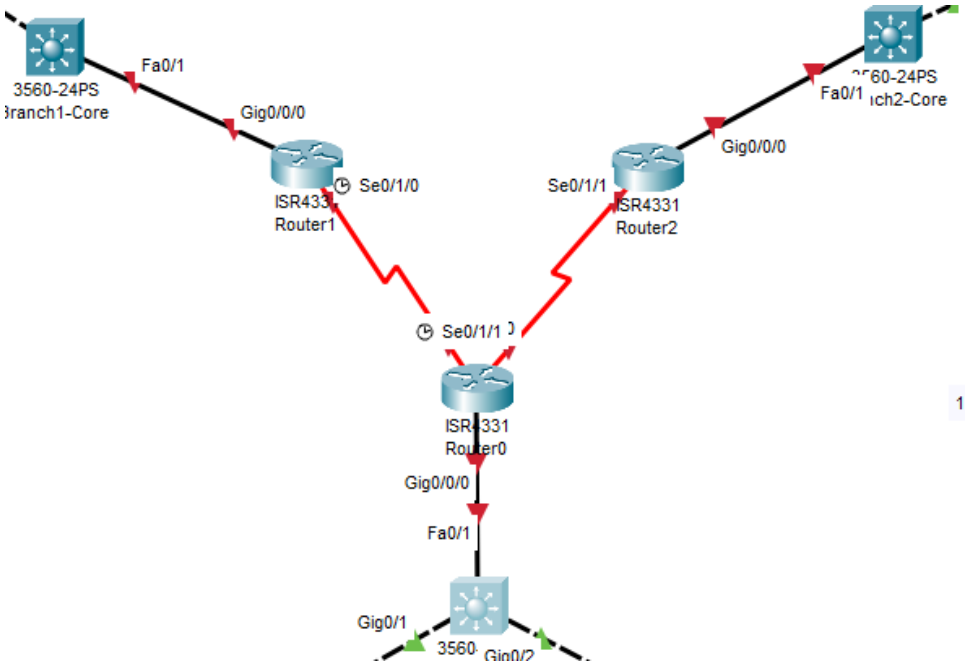
- After the serial interface is added, click on the power switch again to boot up the router.
- Repeat this process with the core switches at the other two branches.

Now, join all 3 routers with Serial DCE cables in the cabling menu.



For simplicity, Se0/1/0 was used on both sides to for branch 1, and Se0/1/1 was used on both sides for branch 2.

1. Next, Connect the routers to the three core switches on the network using copper straight-through cables.



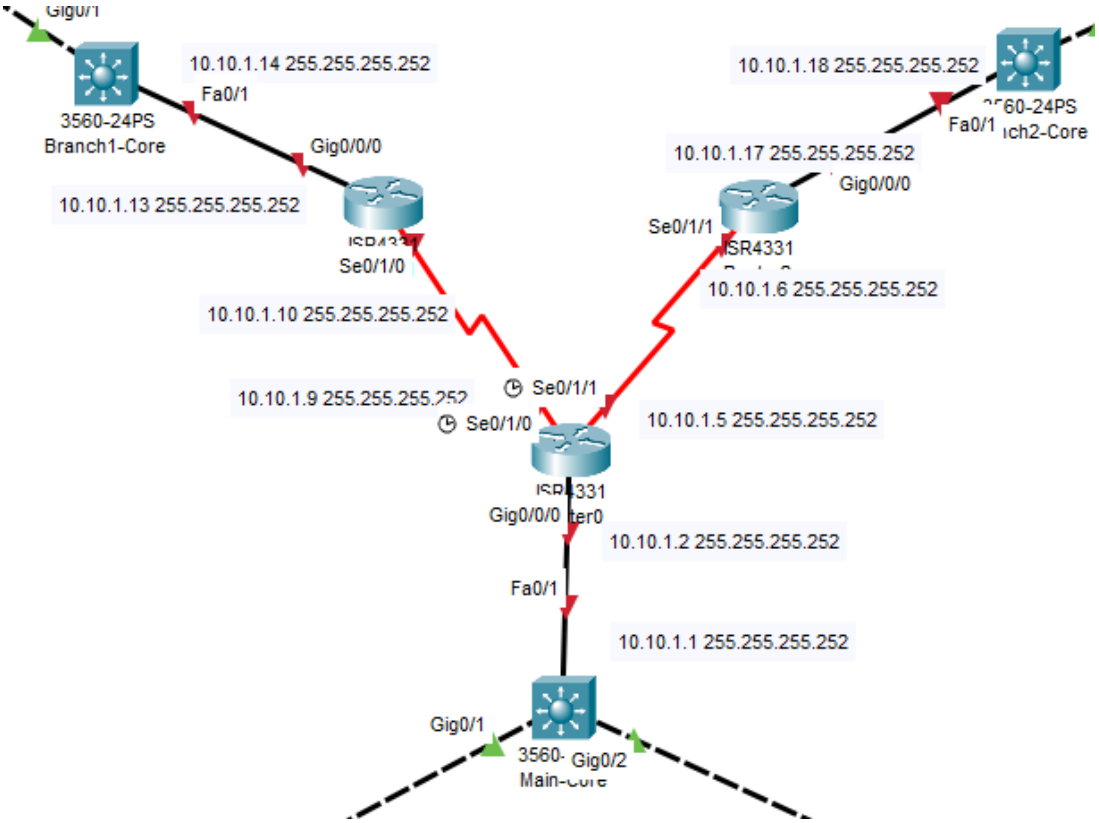
Note that the interface lights are currently showing ‘down’, as indicated by the red triangles. This is because all router interfaces are typically shutdown until explicitly enabled.

Now, we need to assign IP addresses to all of the interfaces. Because these are all point-to-point links, we are going to use a subnet mask that provides us with just two useable addresses. This is done using a subnet mask of /30 or 255.255.255.252. This carves the subnet into blocks of four addresses, with the first address being the network address and the last address the broadcast address, leaving us with 2 usable addresses.

Here is the IP assignment we are going to use for our 5 point-to-point links, each with a netmask of 255.255.255.252

- Main-Core to router0
    - Main-core F0/1 - 10.10.1.1
    - Router 0 G0/0/0 - 10.10.1.2
- Router0 to Router2
    - Router 0 Se0/1/1/1 - 10.10.1.5
    - Router 2 Se0/1/1 - 10.10.1.6
- Router0 to router1
    - Router0 S0/1/0 - 10.10.1.9
    - Router1 Se0/1/0 10.10.1.10
- Router 1 to Branch1-Core

    - Router1 G0/0/0 10.10.1.13
    - Branch1-Core F0/1 10.10.1.14
- Router 2 to Branch2-Core
    - Router2 G0/0/0 10.10.1.17
    - Branch2-Core F0/1 10.10.1.18



Adding labels inside of Packet Tracer can be a helpful tool for keeping track of IP assignments.

Now, we will need to log into each of these devices and assign the listed IP addresses.

First, we will configure the core switches.

- Log into the CLI of the main core switch and enter interface config mode for F0/1
- Enter the ‘no switchport’ command.
    - This tells the switch that that port should be treating as a routable interface instead of a swtichport.
- Assign the appropriate IP address to the interface.
    - In this case ‘ip address 10.10.1.1 255.255.255.252’

```
Main-Core>enable
Main-Core#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Main-Core(config)#interface f0/1
Main-Core(config-if)#no switchport
Main-Core(config-if)#ip address 10.10.1.1 255.255.255.252
Main-Core(config-if)#no shutdown
Main-Core(config-if)#
```

- Repeat this process with the other 2 core switches.

```
Branch1-Core>enable
Branch1-Core#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Branch1-Core(config)#interface f0/1
Branch1-Core(config-if)#no switchport
Branch1-Core(config-if)#ip address 10.10.1.14 255.255.255.252
Branch1-Core(config-if)#no shut
Branch1-Core(config-if)#
```

```
Branch2-Core>enable
Branch2-Core#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Branch2-Core(config)#interface f0/1
Branch2-Core(config-if)#no switchport
Branch2-Core(config-if)#ip address 10.10.1.18 255.255.255.252
Branch2-Core(config-if)#no shut
Branch2-Core(config-if)#
```

- Now, we will Add IP addresses to the routers.
- First, we will log into router0 at our core network.
- Just like the switches, we will configure the gigabit ethernet and serial interfaces with IP addresses. Because these are already routing ports, there is no need to enter ‘no switchport’.
- Remember to enter the ‘no shutdown’ command on each of these interfaces as they are shutdown by default.

On Router0

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface g0/0/0
Router(config-if)#ip address 10.10.1.2 255.255.255.252
Router(config-if)#int s0/1/0
Router(config-if)#ip add 10.10.1.9 255.255.255.252
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/1/0, changed state to down
Router(config-if)#interface s0/1/1
Router(config-if)#ip add 10.10.1.5 255.255.255.252
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/1/1, changed state to down
Router(config-if)#
```

On Router1

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface g0/0/0
Router(config-if)#ip address 10.10.1.13 255.255.255.252
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to up

Router(config-if)#interface s0/1/0
Router(config-if)#ip address 10.10.1.10 255.255.255.252
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up

Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, changed state to up
```

On Router2

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface g0/0/0
Router(config-if)#ip add 10.10.1.17 255.255.255.252
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to up

Router(config-if)#interface s0/1/1
Router(config-if)#ip add 10.10.1.6 255.255.255.252
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/1/1, changed state to up

Router(config-if)#
```

Now, every device on the network should be able to PING one of its neighboring devices.

```
Router#ping 10.10.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/3/17 ms
```

However, our routers only have knowledge of their own locally connected subnets. They have no knowledge of the other networks. It will be up to us to provide that information to them.

- Static routes are created in configure mode. You are essentially giving the router a network and telling it how to get to that network.
    - Router# ip route 10.100.1.0 255.255.255.0 10.90.1.1
        - This tells the router to send any traffic that matches 10.100.1 to 10.90.1.1.
- There is a special kind of static route known as a default route. When created, this type of route functions like an 'else' statement. If none of the previous static routes are in the routing table, then it will default to this one.
    - Router# ip route 0.0.0.0 0.0.0.0 10.80.1.1
        - As you can see, the destination network and netmask are all zeros.

First, we are going to add static routes to our core switches. Because the core switches only have one link leaving the network, we can assume that any traffic that is addressed to one of its local networks will need to be forwarded out that single link. This makes configuring the route easier as we can use a default route.

- On the main-core add a default route to the corresponding interface on router0 10.10.1.2

```
Main-Core>enable
Main-Core#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Main-Core(config)#ip route 0.0.0.0 0.0.0.0 10.10.1.2
Main-Core(config)#
```

- Repeat this process with the other two core switches

```
Branch1-Core>enable
Branch1-Core#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Branch1-Core(config)#ip route 0.0.0.0 0.0.0.0 10.10.1.13
Branch1-Core(config)#
```

```
Branch2-Core>enable
Branch2-Core#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Branch2-Core(config)#ip route 0.0.0.0 0.0.0.0 10.10.1.17
Branch2-Core(config)#
```

Now, we need to add routes to the routers to tell them how to route to the various networks we have. However, because our IP assignment was carefully planned out. We will not have to provide a route to every subnet that exists.

- Remember, Every IP address in our main branch follows the scheme 10.1.X.X
- Likewise with the two side branches with 10.2.X.X and 10.3.X.X
- So, instead of adding a route to every VLAN, we can use a /16 or 255.255.0.0 subnet mask to refer to an entire branch network and all of its subnetworks.

- This makes the routing table smaller and makes configuration much simpler. We only need to add three routes to each of our routers.

On Router0 at the main branch.

- Create a route to the main branch network 10.1.0.0 255.255.0.0 through the core switch
- Create routes to 10.2.0.0 255.255.0.0 and 10.3.0.0 255.255.0.0 through their appropriate serial interface

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 10.1.0.0 255.255.0.0 10.10.1.1
Router(config)#ip route 10.2.0.0 255.255.0.0 10.10.1.10
Router(config)#ip route 10.3.0.0 255.255.0.0 10.10.1.6
Router(config)#
```

Repeat the process for the other routers

Router1 at Branch1

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 10.2.0.0 255.255.0.0 10.10.1.14
Router(config)#ip route 10.1.0.0 255.255.0.0 10.10.1.9
Router(config)#ip route 10.3.0.0 255.255.0.0 10.10.1.9
Router(config)#
```

Router2 at Branch2

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 10.3.0.0 255.255.0.0 10.10.1.18
Router(config)#ip route 10.1.0.0 255.255.0.0 10.10.1.5
Router(config)#ip route 10.2.0.0 255.255.0.0 10.10.1.5
Router(config)#
```

Now, you can use the ‘show ip route’ command to view the full routing table for the devices.

```
Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
S       10.1.0.0/16 [1/0] via 10.10.1.1
S       10.2.0.0/16 [1/0] via 10.10.1.10
S       10.3.0.0/16 [1/0] via 10.10.1.6
C       10.10.1.0/30 is directly connected, GigabitEthernet0/0/0
L       10.10.1.2/32 is directly connected, GigabitEthernet0/0/0
C       10.10.1.4/30 is directly connected, Serial0/1/1
L       10.10.1.5/32 is directly connected, Serial0/1/1
C       10.10.1.8/30 is directly connected, Serial0/1/0
L       10.10.1.9/32 is directly connected, Serial0/1/0

Router#
```

Now, every device on the network should be accessible via its IPv4 address. Try pinging a device in either of the two side branches from ad PC in the core.

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.3.30.5

Pinging 10.3.30.5 with 32 bytes of data:

Request timed out.
Reply from 10.3.30.5: bytes=32 time=14ms TTL=124
Reply from 10.3.30.5: bytes=32 time=42ms TTL=124
Reply from 10.3.30.5: bytes=32 time=1ms TTL=124

Ping statistics for 10.3.30.5:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 42ms, Average = 19ms

C:\>
```

In the next lab, you will expand upon this by adding IPv6 addresses and IPv6 routing to the network.