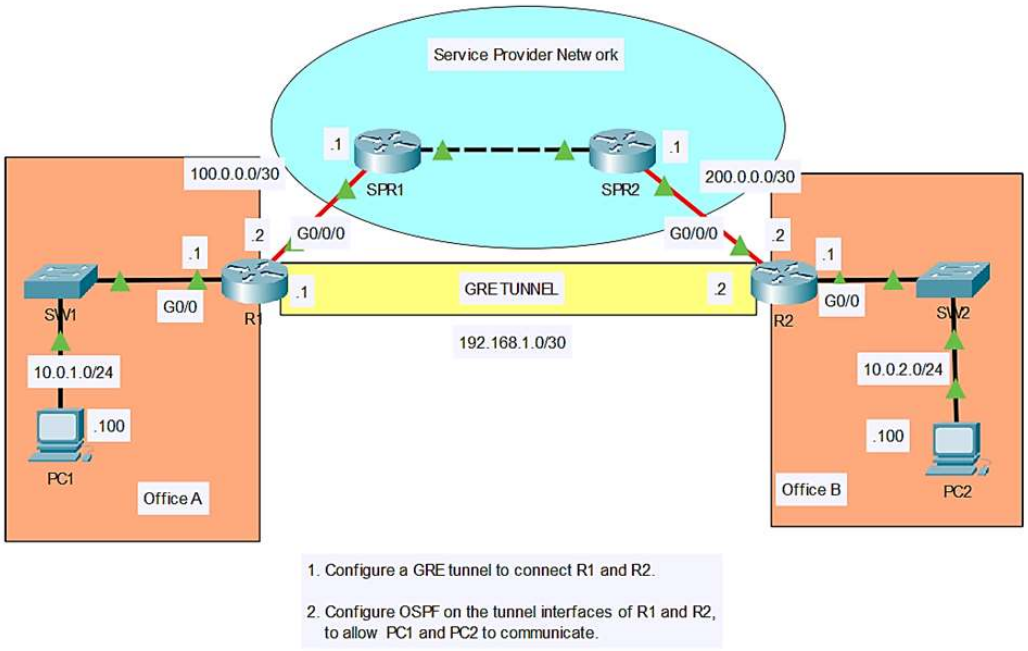
GRE Tunnel Configuration

In this lab, I'll show you how to configure a GRE tunnel between two routers. Then we will configure OSPF on the tunnel interfaces of R1 and R2, to allow PC1 in Office A to communicate with PC2 in Office B. You can follow along by downloading this GRE Tunnels Packet Tracer File and opening it in Cisco's Free Packet Tracer Simulator (*create a free account, enroll in one of the free courses and download the free software*).



R1

To configure a GRE tunnel, we have to make a tunnel interface. This is not a physical interface of course, but a virtual interface, like a loopback interface:

R1(config)#interface tunnel 0

```
R1>en
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface ?
  Dialer            Dialer interface
  Dot11Radio        Dot11 interface
  Ethernet          IEEE 802.3
  FastEthernet      FastEthernet IEEE 802.3
  GigabitEthernet   GigabitEthernet IEEE 802.3z
  Loopback          Loopback interface
  Port-channel      Ethernet Channel of interfaces
  Serial            Serial
  Tunnel            Tunnel interface
  Virtual-Template  Virtual Template interface
  Vlan              Catalyst Vlans
  range             interface range command
R1(config)#interface tunnel 0

R1(config-if)#
%LINK-5-CHANGED: Interface Tunnel0, changed state to up
```

Okay, we've have created the tunnel interface.

Now we have to specify 3 more items to complete the GRE configuration:

1. *Tunnel Source*
2. *Tunnel Destination*
3. *IP address of the virtual tunnel itself*

Tunnel Source: specify which physical interface on R1 will be used for the tunnel. We should use the interface connected to the service provider, which is g0/0/0 :

R1(config-if)#tunnel source g0/0/0

Tunnel Destination: specify the IP address of the other end of the tunnel, so R2. So, enter R2's WAN interface's IP, 200.0.0.2 :

R1(config-if)#tunnel destination 200.0.0.2

Virtual tunnel interface itself needs an IP address:

R1(config-if)#ip address 192.168.1.1 255.255.255.252

Okay, that's all the configuration needed on R1. Below is what we just configured above on R1:

```
R1(config-if)#tunnel source g0/0/0
R1(config-if)#tunnel destination 200.0.0.2
R1(config-if)#ip address 192.168.1.1 255.255.255.252
```

Let's verify the virtual tunnel interface has been created:

```
R1(config-if)#do show ip int br
Interface              IP-Address      OK? Method Status
Protocol
GigabitEthernet0/0     10.0.1.1        YES NVRAM  up                    up
GigabitEthernet0/1     unassigned      YES NVRAM  administratively down down
GigabitEthernet0/2     unassigned      YES NVRAM  administratively down down
GigabitEthernet0/0/0   100.0.0.2       YES manual up                    up
Tunnel0                192.168.1.1     YES manual up                    down
Vlan1                  unassigned      YES unset  administratively down down
```

Yes, there it is. It shows that the interface is in an up/down status. Why down? We'll investigate shortly.

Let's move on to R2 and configure the other side of the tunnel.

R2

Create the virtual tunnel Interface:

R2(config)#int tunnel 0

Tunnel source (just like on R1, it will be g0/0/0):

R2(config-if)#tunnel source g0/0/0

Tunnel destination (this time it will be the IP address of R1's WAN interface, 100.0.0.2):

R2(config-if)#tunnel destination 100.0.0.2

Finally, the IP address on the tunnel:

R2(config-if)#ip address 192.168.1.2 255.255.255.252

Let's verify the virtual tunnel interface has been created:

R2(config-if)#do show ip int br

```
R2>en
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#int tunnel 0

R2(config-if)#
%LINK-5-CHANGED: Interface Tunnel0, changed state to up

R2(config-if)#tunnel source g0/0/0
R2(config-if)#tunnel destination 100.0.0.2
R2(config-if)#ip address 192.168.1.2 255.255.255.252
R2(config-if)#
R2(config-if)#
R2(config-if)#do show ip int br
Interface              IP-Address      OK? Method Status
Protocol
GigabitEthernet0/0     10.0.2.1        YES NVRAM  up                    up
GigabitEthernet0/1     unassigned      YES NVRAM  administratively down down
GigabitEthernet0/2     unassigned      YES NVRAM  administratively down down
GigabitEthernet0/0/0   200.0.0.2       YES manual up                    up
Tunnel0                192.168.1.2     YES manual up                    down
Vlan1                  unassigned      YES unset  administratively down down
```

Above is what we just configured on R2

Above, we see the tunnel interface is also showing an up/down status even though we've configured both sides. Why is that? Let me show you by issuing:

R2(config-if)#do show ip route

```
R2(config-if)#do 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

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.2.0/24 is directly connected, GigabitEthernet0/0
L       10.0.2.1/32 is directly connected, GigabitEthernet0/0
     200.0.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       200.0.0.0/30 is directly connected, GigabitEthernet0/0/0
L       200.0.0.2/32 is directly connected, GigabitEthernet0/0/0
```

As you can see, So R2 doesn't have a connected route for its tunnel interface of course, because the interface is still down.
It has connected routes for its physical interfaces, but we're missing a critical route.
R2 doesn't know how to reach the IP address we specified as the tunnel destination, 100.0.0.2.
If R2 doesn't know how to get to 100.0.0.2, it can't build a GRE tunnel to 100.0.0.2.

Let's fix that:

R2(config-if)#exit

Then I'll just configure a default route:

R2(config)#ip route, 0.0.0.0 0.0.0.0 200.0.0.1

After we enter a default route which goes to 100.0.0.2, the tunnel interface comes up.

Let's verify with:

R2(config)#do show ip route

Now we have the connected route for the tunnel in the route table:

```
R2(config-if)#exit
R2(config)#
R2(config)#ip route 0.0.0.0 0.0.0.0 200.0.0.1
R2(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up

R2(config)#do 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 200.0.0.1 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.2.0/24 is directly connected, GigabitEthernet0/0
L       10.0.2.1/32 is directly connected, GigabitEthernet0/0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/30 is directly connected, Tunnel0
L       192.168.1.2/32 is directly connected, Tunnel0
     200.0.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       200.0.0.0/30 is directly connected, GigabitEthernet0/0/0
L       200.0.0.2/32 is directly connected, GigabitEthernet0/0/0
S*   0.0.0.0/0 [1/0] via 200.0.0.1
```

Let's try to ping the R1 side:

```
R2(config)#do ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

The ping still isn't working even though the interface is up and we have a route.

The reason for that is that we also have to configure a route on R1 to 200.0.0.2

R1

Let's check the routing table on R1:

```
R1(config-if)#do 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

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.1.0/24 is directly connected, GigabitEthernet0/0
L       10.0.1.1/32 is directly connected, GigabitEthernet0/0
     100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       100.0.0.0/30 is directly connected, GigabitEthernet0/0/0
L       100.0.0.2/32 is directly connected, GigabitEthernet0/0/0
```

As you can see, there's only connected routes for its physical interfaces, so it doesn't know how to get to the tunnel destination 200.0.0.2

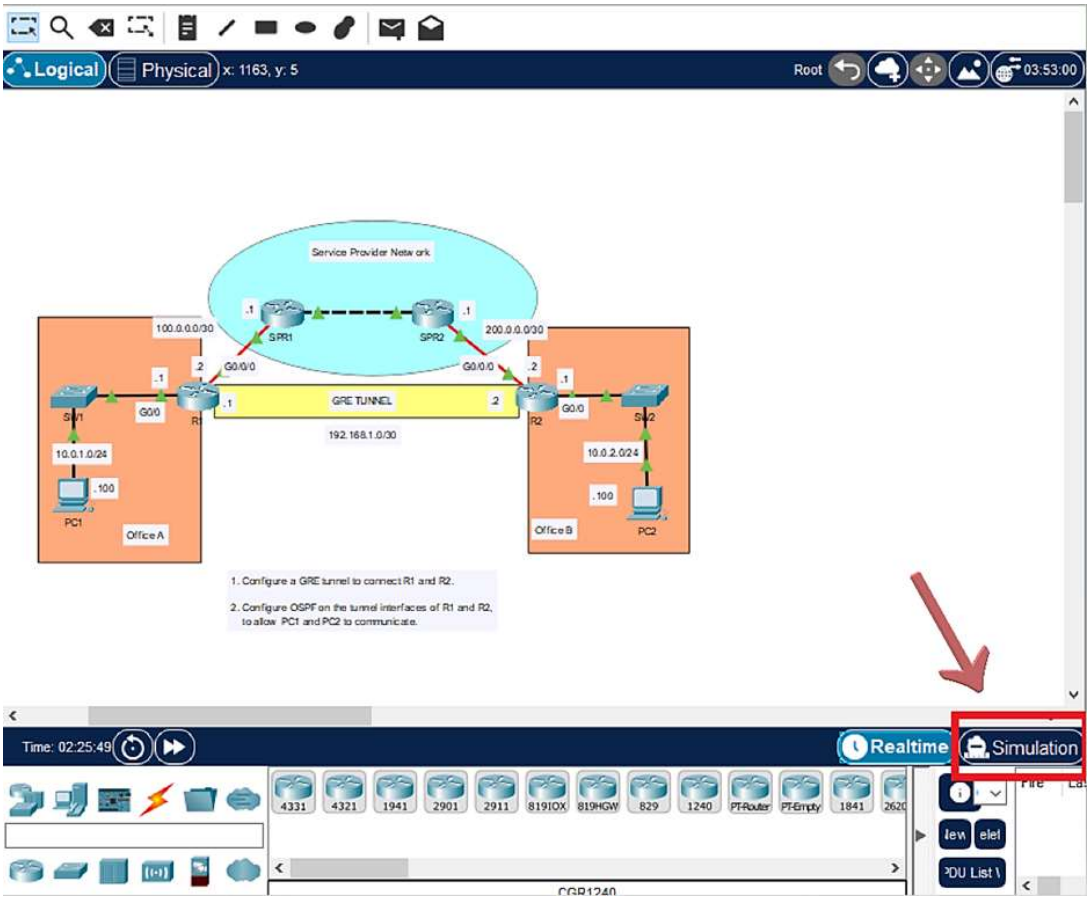
Let's give it a default route as well:

```
R1(config-if)#exit
R1(config)#ip route 0.0.0.0 0.0.0.0 100.0.0.1
R1(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up

R1(config)#
R1(config)#do ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/0 ms
```

As usual, the first few pings might fail because the ARP process is a bit slow in Packet Tracer. But after ARP completes, they will succeed.

So, although R1 and R2 aren't directly connected, they will behave as if they are directly connected through the GRE tunnel.

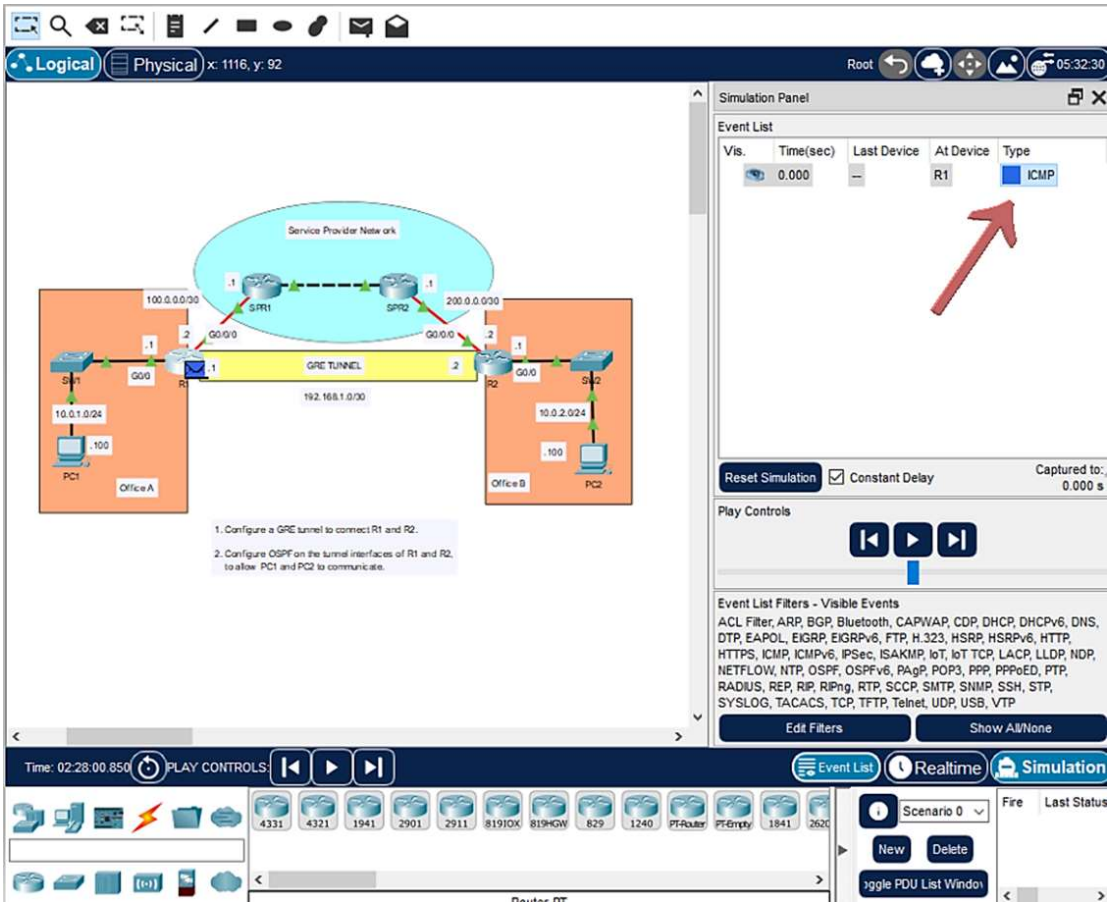
Let's first look at a ping from R1 to R2 over the GRE tunnel in Simulation mode:



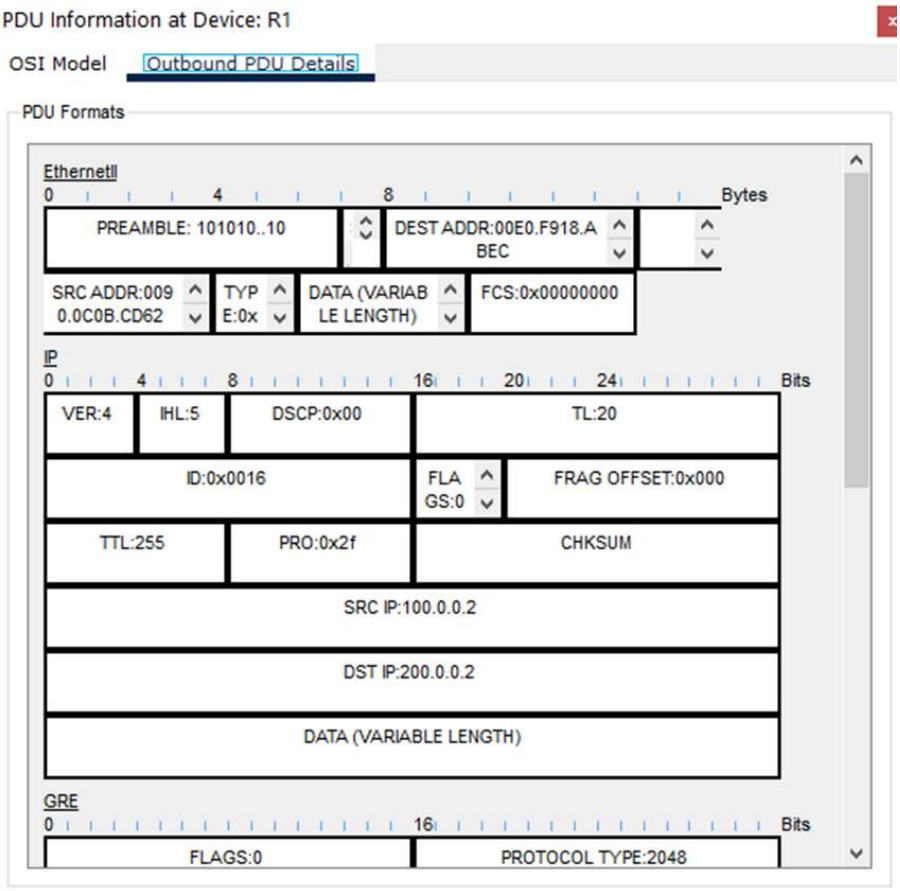
From R1 in Simulation issue the following ping:

R1(config)#do ping 192.168.1.2

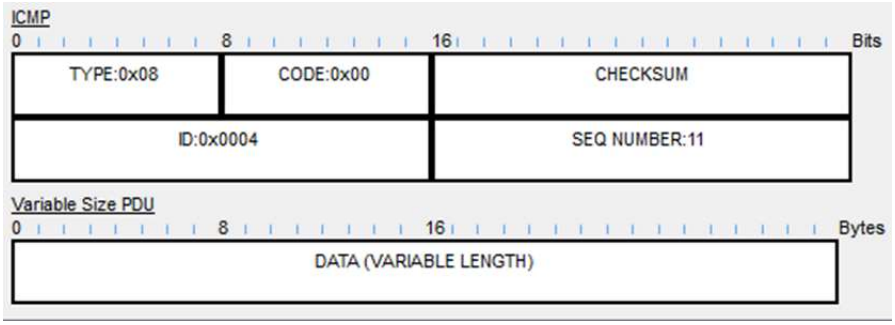
Then click on ICMP under Type in the right Simulation Panel to view the packet:



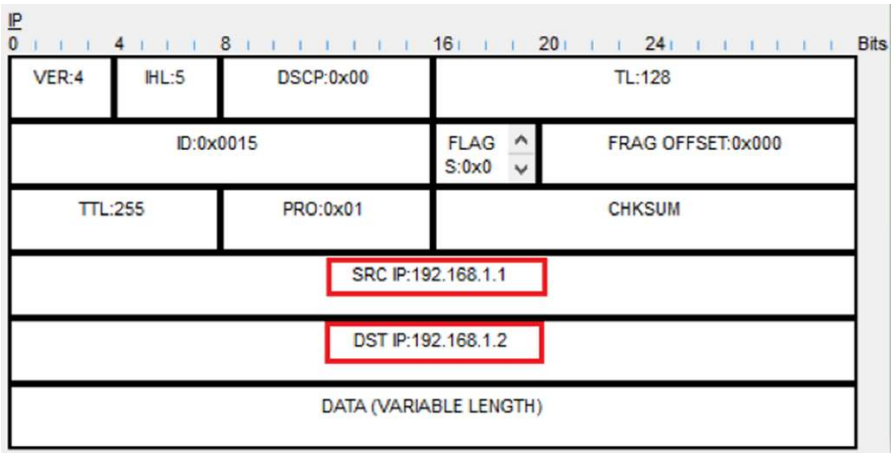
Then a popup window will appear labeled PDU Information at Device: R1. Select Outbound PDU Details tab:



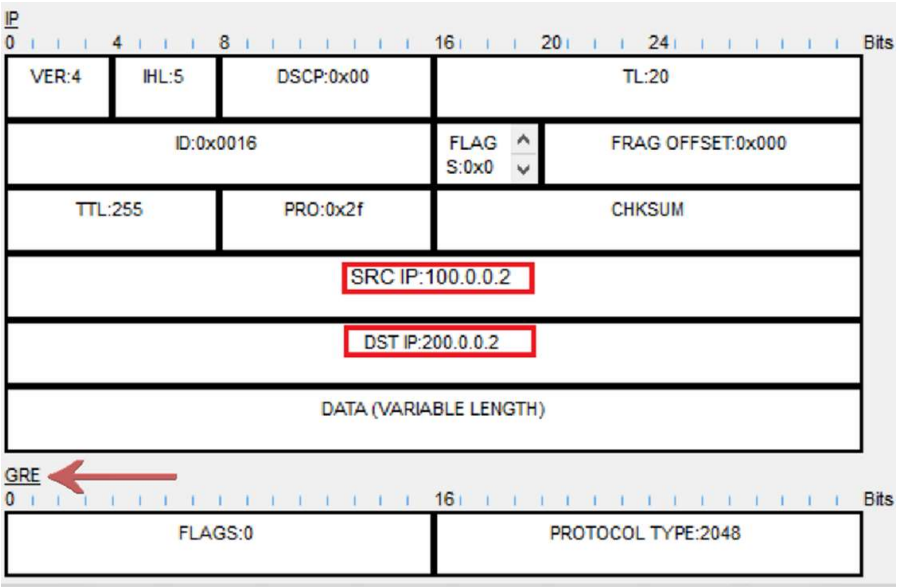
Scroll to bottom where we will see the ICMP message (the PING):



Then above that we see that the ICMP is encapsulated with an IP Header, which shows the source 192.168.1.1 and destination 192.168.1.2. These are the addresses of the tunnel interfaces:



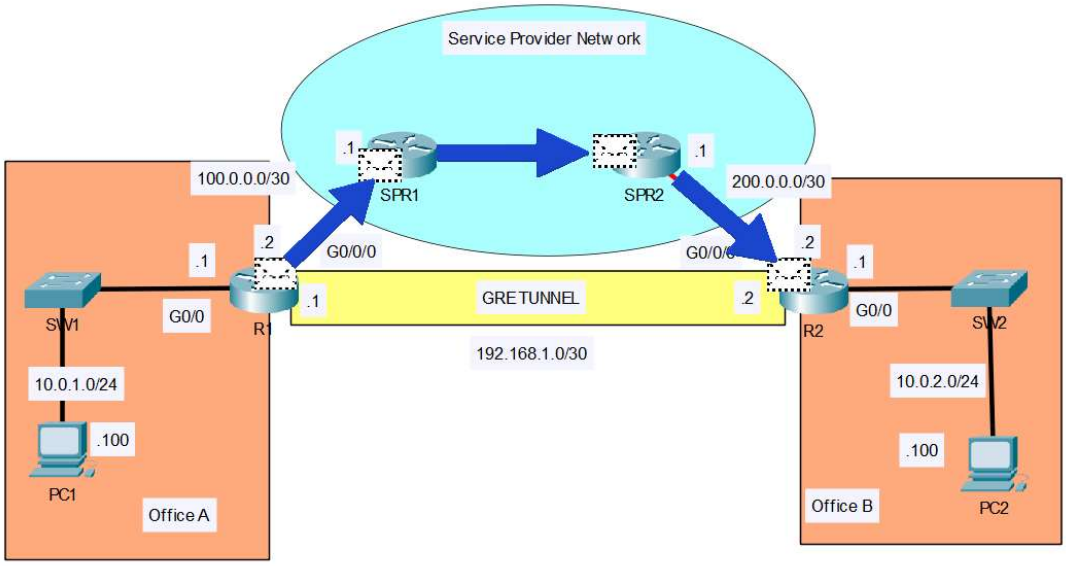
But still even on top of that, there is a GRE header, and on top of that, there is another IP header. In this outer IP header, the source IP is 100.0.0.2 (R1's g0/0/0 interface), and the destination IP is 200.0.0.2 (R2's g0/0/0 interface):



NOTE

The ping from R1 to R2 moves through the Service Provider Network as it would if the GRE tunnel was not configured. But as we have seen above, the communication is encapsulated several times over to create the GRE tunnel:

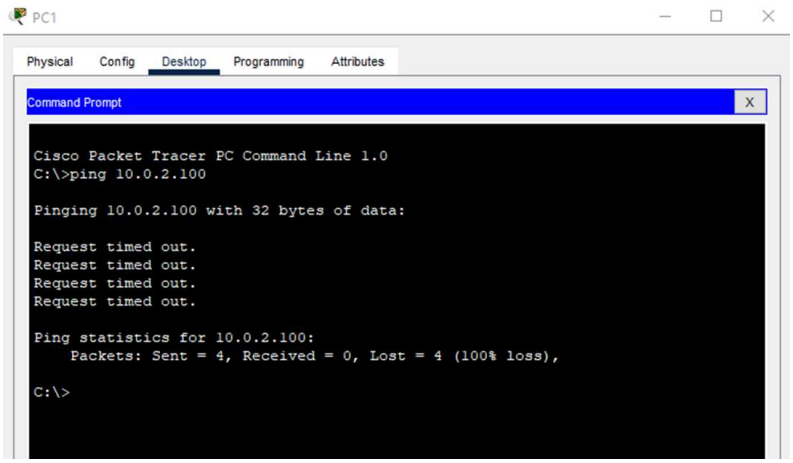
First, PING goes to the SPR1 router. Second, the PING goes to the SPR2 router. Thirdly, the PING finally goes to R2.



Then the PING will take the same path back a ping echo reply.

The second part of this lab is to "Configure OSPF on the tunnel interfaces of R1 and R2, to allow PC1 and PC2 to communicate.

First let's send a ping from PC1 to PC2 to show that they cannot communicate yet:



As you can see the ping's timed out and didn't work. But when R1 and R2 become OSPF neighbors, they will learn each other's routes, and PC1 and PC2 will be able to communicate over the GRE tunnel.

Let's do that by enabling OSPF on R1 first by entering OSPF mode:

R1(config)#router ospf 1

The following network command enables OSPF on the Tunnel0 interface:

R1(config-router)#network 192.168.1.1 0.0.0.0 area 0

The following network command enables OSPF on the g0/0/0 interface.

R1(config-router)#network 10.0.1.1 0.0.0.0 area 0

And we'll make g0/0 a passive interface since there are no neighbors connected to it:

R1(config-router)#passive-interface g0/0

Below are the OSPF commands we just configured:

```
R1(config)#router ospf 1
R1(config-router)#network 192.168.1.1 0.0.0.0 area 0
R1(config-router)#
R1(config-router)#network 10.0.1.1 0.0.0.0 area 0
R1(config-router)#
R1(config-router)#passive-interface g0/0
R1(config-router)#
```

Now let's do the same to R2:

```
R2(config)#router ospf 1
R2(config-router)#network 192.168.1.2 0.0.0.0 area 0
R2(config-router)#network
12:46:19: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on Tunnel0 from LOADING
to FULL, Loadin
% Ambiguous command: "n"
R2(config-router)#network 10.0.2.1 0.0.0.0 area 0
R2(config-router)#passive-interface g0/0
R2(config-router)#
```

As highlighted above, you can see R1 and R2 have become OSPF neighbors. So fast that it messed up my second network command I was typing lol. I should've have enabled the logging synchronous, which eliminates th typing errors.

Let's check the routes on R2 with the *do show ip route* command:

```
R2(config-router)#do 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 200.0.0.1 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O       10.0.1.0/24 [110/1001] via 192.168.1.1, 00:05:07, Tunnel0
C       10.0.2.0/24 is directly connected, GigabitEthernet0/0
L       10.0.2.1/32 is directly connected, GigabitEthernet0/0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/30 is directly connected, Tunnel0
L       192.168.1.2/32 is directly connected, Tunnel0
     200.0.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       200.0.0.0/30 is directly connected, GigabitEthernet0/0/0
L       200.0.0.2/32 is directly connected, GigabitEthernet0/0/0
S*   0.0.0.0/0 [1/0] via 200.0.0.1
```

Okay, it learned a route to 10.0.1.0/24, R1's LAN, via the Tunnel0 interface.

Let's check the routes on R1 as well:

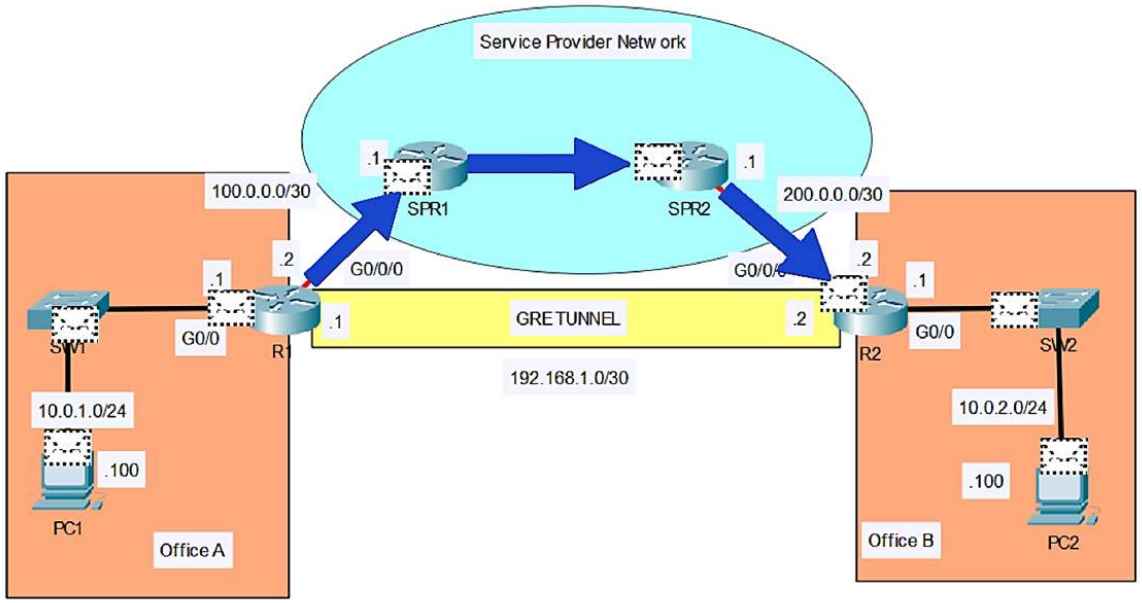
```
R1(config-router)#do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external 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 100.0.0.1 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.0.1.0/24 is directly connected, GigabitEthernet0/0
L       10.0.1.1/32 is directly connected, GigabitEthernet0/0
O       10.0.2.0/24 [110/1001] via 192.168.1.2, 00:06:50, Tunnel0
     100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       100.0.0.0/30 is directly connected, GigabitEthernet0/0/0
L       100.0.0.2/32 is directly connected, GigabitEthernet0/0/0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/30 is directly connected, Tunnel0
L       192.168.1.1/32 is directly connected, Tunnel0
S*   0.0.0.0/0 [1/0] via 100.0.0.1
```

Okay, it learned a route to 10.0.2.0/24 via the Tunnel0 interface.

Now let's try that same ping again from PC1 to PC2:

```
C:\>ping 10.0.2.100

Pinging 10.0.2.100 with 32 bytes of data:

Request timed out.
Reply from 10.0.2.100: bytes=32 time<1ms TTL=126
Reply from 10.0.2.100: bytes=32 time<1ms TTL=126
Reply from 10.0.2.100: bytes=32 time<1ms TTL=126

Ping statistics for 10.0.2.100:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

One or two pings might fail, but after that, they will work. R1 will encapsulate the packet from PC1 using GRE and send it over the tunnel to R2.



In this lab I've shown how to create a GRE tunnel between two routers and activate OSPF on the tunnel.