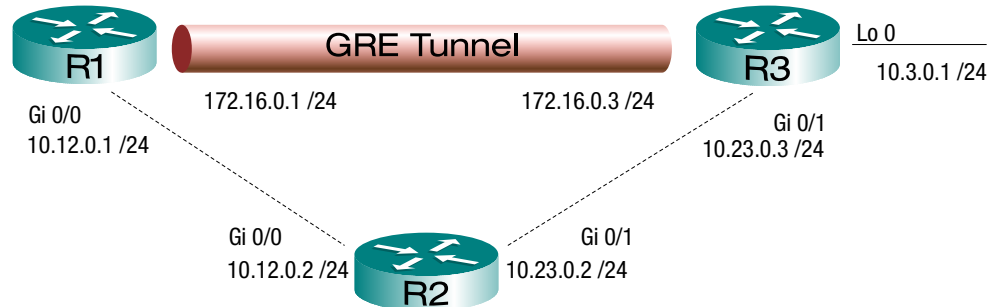


G O A L

Establish and use a GRE tunnel between two routers. Imagine that R1 and R3 are your two sites and R2 is the internet. The tunnel will appear to be a direct connection between your sites and will use your internal addressing, even though the data will physically travel through R2. You can view the tunnel as if the data never left your network.



Topology—You'll need

- 3 routers of almost any vintage (A 2621xm is cheap. I'm using 2851s and a 2821. Interface names may vary between FastEthernet and GigabitEthernet, depending on hardware choice.)
- 2 Ethernet crossover cable to connect the two routers
- Some way to issue commands over the router console ports
- The tunnel will be virtual; there is no wiring directly connecting R1 and R3.

C O N F I G U R A T I O N S T E P S

Wire the topology and give your routers their basic configurations:

- Configure hostnames per the diagram
- Tell the router that when it doesn't recognize a command (for example a typo), it shouldn't attempt to contact a DNS server to prepare to telnet to that hostname.
- Tell the console port not to log you out after a period of inactivity
- Tell the router that if it sends syslog messages to the console port while you're typing there, it should reprint the prompt and whatever you had already typed so that you know where you were.

Establish end-to-end routing with static routes and test before attempting the tunnel

- Configure interface IP addresses per the diagram for the physical interfaces and the loopback interface on R3 only (not the tunnel yet)
- Create static routes on R1 and R3 to ensure that R1 can ping R3 and the response can find its way back
- *Don't* create a static route from R1 to the loopback interface on R3; we'll use the tunnel for that route.

Create a GRE tunnel from R1 to R3 and confirm with a ping from one end of the tunnel to the address on the other end of the tunnel.

Create a static route on R1 that uses the tunnel to reach the subnet on R3's Loopback interface and test it with a ping.

Did the ping work? Why or why not? _____

Observe how your routing tables use the tunnel and fix anything that broke the ping you just tried.

C O N F I G U R A T I O N W A L K T H R O U G H

Give the routers their basic configurations, physical interface IP addresses, and static routes

R1	R2	R3
hostname R1 no ip domain lookup line con 0 exec-timeout 0 0 logging synchronous	hostname R2 no ip domain lookup line con 0 exec-timeout 0 0 logging synchronous	hostname R3 no ip domain lookup line con 0 exec-timeout 0 0 logging synchronous
interface GigabitEthernet0/0 ip address 10.12.0.1 255.255.255.0	interface GigabitEthernet0/0 ip address 10.12.0.2 255.255.255.0 ! interface GigabitEthernet0/1 ip address 10.23.0.2 255.255.255.0	interface GigabitEthernet0/1 ip address 10.23.0.3 255.255.255.0 ! interface Loopback0 ip address 10.3.0.1 255.255.255.0
ip route 10.23.0.0 255.255.255.0 10.12.0.2		ip route 10.12.0.0 255.255.255.0 10.23.0.2

Check end-to-end routing. You only need to ping one direction because the ICMP replies automatically test the other direction as they come back.

```
R1
R1# ping 10.23.0.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.23.0.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Now, you're ready to create the tunnel. On each end, you'll

- Create a virtual "tunnel" interface [Line 1 of configuration on next page]—just name it and it'll be created, just like a loopback. The tunnel number, e.g. Tunnel 0, is local to the router and doesn't need to match the other end
- Give your tunnel interface an IP address [Line 2]—These addresses are internal and both ends of the tunnel need to be in the same subnet
- Specify a source for the tunnel [Line 3]—The IP address of the "real" exit interface that leads in the direction your data will physically take after it's encapsulated with tunnel headers. You can accomplish the same thing by naming the interface rather than its IP address [see the R2 side of the example]
- Specify a destination for the tunnel [Line 4]—The IP address you used at the other end of the tunnel as its source

R1	R2
<pre> 1 interface Tunnel0 2 ip address 172.16.0.1 255.255.255.0 3 tunnel source 10.12.0.1 4 tunnel destination 10.23.0.3 </pre>	<pre> interface Tunnel58 ip address 172.16.0.3 255.255.255.0 tunnel source GigabitEthernet0/1 tunnel destination 10.12.0.1 </pre>

Test your tunnel with a ping.

```

R3
R3# ping 172.16.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

```

Double check that R1 doesn't already have a route to R3's loopback before you add one.

```

R1
R1# ping 10.3.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.0.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

R1# show ip route 10.3.0.1
% Subnet not in table

```

Set up a static route on R1 that uses the tunnel to reach R3's loopback.

```

R1
R1(config)# ip route 10.3.0.0 255.255.255.0 172.16.0.3
R1(config)# do ping 10.3.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

```

Notice that the ping worked even without a matching static route on R3. That's because the ping originated on your router and used the exit interface (the tunnel) as its source address. That network is directly connected on both R1 and R3, unlike the loopback on the far side of R3.

View the routing table to see how the tunnel is being used. Notice that its network is listed as directly connected even though the other end of the tunnel could be many "real" hops away.

```

R1
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      Rest of legend omitted
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
S       10.3.0.0/24 [1/0] via 172.16.0.3
C       10.12.0.0/24 is directly connected, GigabitEthernet0/0
L       10.12.0.1/32 is directly connected, GigabitEthernet0/0
S       10.23.0.0/24 [1/0] via 10.12.0.2
    172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/24 is directly connected, Tunnel0
L       172.16.0.1/32 is directly connected, Tunnel0

```

