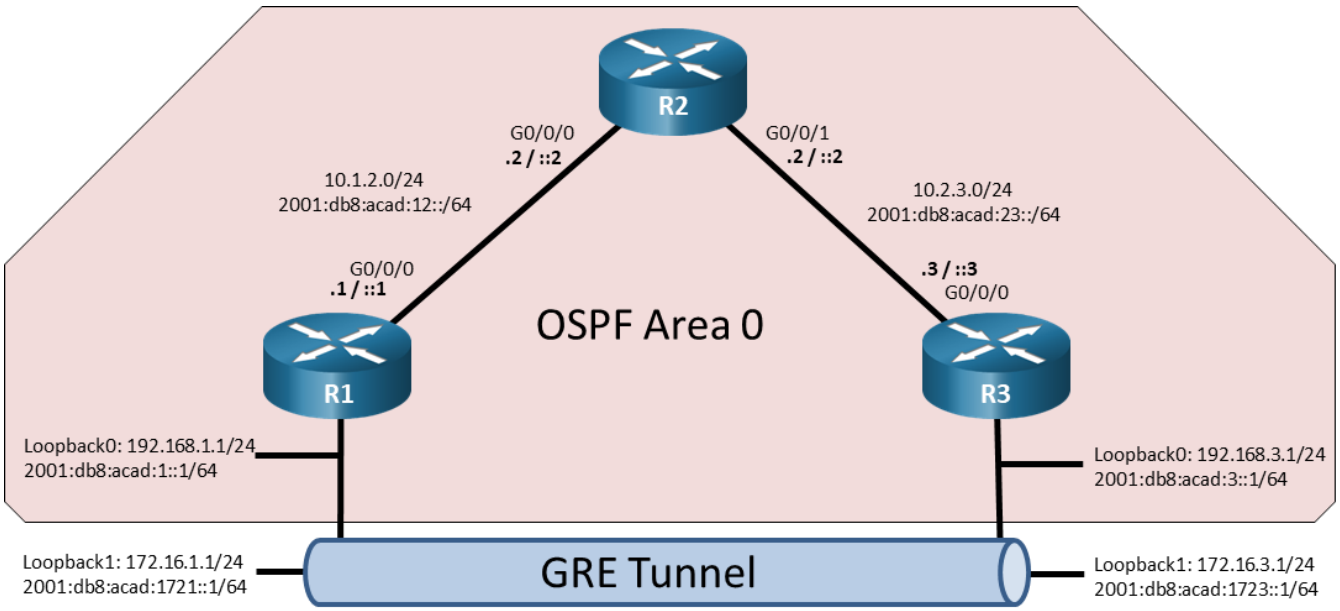
Lab - Implement a GRE Tunnel

Topology



Addressing Table

Device	Interface	IPv4 Address	IPv6 Address	IPv6 Link-Local
R1	G0/0/0	10.1.2.1/24	2001:db8:acad:12::1/64	fe80::1:1
	Loopback 0	192.168.1.1/24	2001:db8:acad:1::1/64	fe80::1:2
	Loopback 1	172.16.1.1/24	2001:db8:acad:1721::1/64	fe80::1:3
R2	G0/0/0	10.1.2.2/24	2001:db8:acad:12::2/64	fe80::2:1
	G0/0/1	10.2.3.2/24	2001:db8:acad:23::2/64	fe80::2:1
R3	G0/0/0	10.2.3.3/24	2001:db8:acad:23::3/64	fe80::3:1
	Loopback 0	192.168.3.1/24	2001:db8:acad:3::1/64	fe80::3:2
	Loopback 1	172.16.3.1/24	2001:db8:acad:1723::1/64	fe80::3:3

Objectives

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Configure and Verify GRE Tunnels with Static Routing

Part 3: Configure and Verify GRE Tunnels by Using a Routing Protocol

Part 4: Examine the Recursive Routing Problem with GRE

Background / Scenario

Overlay networks allow you to insert flexibility into existing topologies, which are then referred to as underlay networks. Cisco's Generic Routing Encapsulation (GRE) protocol is a very useful tool that allows you to create overlay networks to support many different purposes. It is very flexible and works with IPv4 or IPv6 as an underlay network. In this lab you will deploy basic GRE tunnels over both IPv4 and IPv6 underlay networks.

Note: This lab is an exercise in configuring and verifying various implementations of GRE tunnels and does not reflect networking best practices.

Note: The routers used with CCNP hands-on labs are Cisco 4221 with Cisco IOS XE Release 16.9.4 (universalk9 image). Other routers and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs.

Note: Ensure that the routers have been erased and have no startup configurations. If you are unsure contact your instructor.

Required Resources

- 3 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 1 PC (Choice of operating system with a terminal emulation program installed)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

Instructions

Part 1: Build the Network and Configure Basic Device Settings

In Part 1, you will set up the network topology and configure basic settings.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for each switch.

a. Console into each router, enter global configuration mode, and apply the basic settings. A command list for each device is listed below for initial configurations.

Router R1

```
hostname R1
no ip domain lookup
ipv6 unicast-routing
banner motd # R1, Implement a GRE Tunnel #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 4
 privilege level 15
 password cisco123
 exec-timeout 0 0
```

```
 logging synchronous
 login
 exit
router ospf 4
 router-id 1.1.1.4
 exit
ipv6 router ospf 6
 router-id 1.1.1.6
 exit
interface g0/0/0
 ip address 10.1.2.1 255.255.255.0
 ipv6 address fe80::1:1 link-local
 ipv6 address 2001:db8:acad:12::1/64
 no shutdown
 ip ospf 4 area 0
 ipv6 ospf 6 area 0
 exit
interface loopback 0
 ip address 192.168.1.1 255.255.255.0
 ipv6 address fe80::1:2 link-local
 ipv6 address 2001:db8:acad:1::1/64
 no shutdown
 ip ospf 4 area 0
 ipv6 ospf 6 area 0
 exit
interface loopback 1
 ip address 172.16.1.1 255.255.255.0
 ipv6 address fe80::1:3 link-local
 ipv6 address 2001:db8:acad:1721::1/64
 no shutdown
 exit
```

Router R2

```
hostname R2
no ip domain lookup
ipv6 unicast-routing
banner motd # R2, Implement a GRE Tunnel #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 4
 privilege level 15
 password cisco123
 exec-timeout 0 0
 logging synchronous
```

```
 login
 exit
router ospf 4
 router-id 2.2.2.4
 exit
ipv6 router ospf 6
 router-id 2.2.2.6
 exit
interface g0/0/0
 ip address 10.1.2.2 255.255.255.0
 ipv6 address fe80::2:1 link-local
 ipv6 address 2001:db8:acad:12::2/64
 no shutdown
 ip ospf 4 area 0
 ipv6 ospf 6 area 0
 exit
interface g0/0/1
 ip address 10.2.3.2 255.255.255.0
 ipv6 address fe80::2:2 link-local
 ipv6 address 2001:db8:acad:23::2/64
 no shutdown
 ip ospf 4 area 0
 ipv6 ospf 6 area 0
 exit
```

Router R3

```
hostname R3
no ip domain lookup
ipv6 unicast-routing
banner motd # R3, Implement a GRE Tunnel #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 4
 privilege level 15
 password cisco123
 exec-timeout 0 0
 logging synchronous
 login
 exit
router ospf 4
 router-id 3.3.3.4
 exit
ipv6 router ospf 6
 router-id 3.3.3.6
```

```
 exit
interface g0/0/0
 ip address 10.2.3.3 255.255.255.0
 ipv6 address fe80::3:1 link-local
 ipv6 address 2001:db8:acad:23::3/64
 no shutdown
 ip ospf 4 area 0
 ipv6 ospf 6 area 0
 exit
interface loopback 0
 ip address 192.168.3.1 255.255.255.0
 ipv6 address fe80::3:2 link-local
 ipv6 address 2001:db8:acad:3::1/64
 no shutdown
 ip ospf 4 area 0
 ipv6 ospf 6 area 0
 exit
interface loopback 1
 ip address 172.16.3.1 255.255.255.0
 ipv6 address fe80::3:3 link-local
 ipv6 address 2001:db8:acad:1723::1/64
 no shutdown
 exit
```

b. Set the clock on each device to UTC time.

c. Save the running configuration to startup-config.

Part 2: Configure and Verify GRE Tunnels with Static Routing

In Part 2, you will configure and verify GRE Tunnels between R1 and R3, and you will use static routes for overlay reachability and dynamic routing for underlay reachability. You will configure two tunnels, one for IPv4 traffic and one of IPv6 traffic. GRE tunnels are extremely flexible, and there are many options for implementation beyond what is being done in this lab.

Step 1: Verify reachability between R1 and R3.

a. From R1, ping R3 interface Loopback 0 using IPv4. All pings should be successful.

```
R1# ping 192.168.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

b. From R1, ping R3 interface Loopback 0 using IPv6. All pings should be successful.

```
R1# ping 2001:db8:acad:3::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:3::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/7 ms
```

Step 2: Create an IPv4-based GRE tunnel between R1 and R3.

a. On R1, create interface Tunnel 0, specifying the IP address 100.100.100.1/30, a tunnel source of Loopback0, and a tunnel destination of 192.168.3.1.

```
R1(config)# interface tunnel 0
R1(config-if)# ip address 100.100.100.1 255.255.255.252
R1(config-if)# tunnel source loopback 0
R1(config-if)# tunnel destination 192.168.3.1
R1(config-if)# exit
```

b. On R1, create a static route to 172.16.3.0/24 via interface Tunnel 0.

```
R1(config)# ip route 172.16.3.0 255.255.255.0 tunnel 0
```

c. On R3, create interface Tunnel 0, specifying the IP address 100.100.100.2/30, a tunnel source of Loopback0, and a tunnel destination of 192.168.1.1.

```
R3(config)# interface tunnel 0
R3(config-if)# ip address 100.100.100.2 255.255.255.252
R3(config-if)# tunnel source loopback 0
R3(config-if)# tunnel destination 192.168.1.1
R3(config-if)# exit
```

d. On R3, create a static route to 172.16.1.0/24 via interface Tunnel 0.

```
R3(config)# ip route 172.16.1.0 255.255.255.0 tunnel 0
```

e. On R1, issue the command **show interface tunnel 0** and examine the output.

```
R1# show interface tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 100.100.100.1/30
  MTU 9976 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 192.168.1.1 (Loopback0), destination 192.168.3.1
   Tunnel Subblocks:
      src-track:
         Tunnel0 source tracking subblock associated with Loopback0
          Set of tunnels with source Loopback0, 1 member (includes iterators), on
interface <OK>
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255, Fast tunneling enabled
  Tunnel transport MTU 1476 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters 00:02:45
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
```

```
Queueing strategy: fifo
Output queue: 0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
   0 packets input, 0 bytes, 0 no buffer
   Received 0 broadcasts (0 IP multicasts)
   0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   0 packets output, 0 bytes, 0 underruns
   0 output errors, 0 collisions, 0 interface resets
   0 unknown protocol drops
   0 output buffer failures, 0 output buffers swapped out
```

f. From R1, **ping 172.16.3.1**. The pings should be successful.

Step 3: Create an IPv6-based GRE tunnel between R1 and R3.

a. On R1, create interface Tunnel 1, specifying the IPv6 address 2001:db8:ffff::1/64, a tunnel source of Loopback0, a tunnel destination of 2001:db8:acad:3::1, and the tunnel mode GRE IPv6.

```
R1(config)# interface tunnel 1
R1(config-if)# ipv6 address 2001:db8:ffff::1/64
R1(config-if)# tunnel source loopback 0
R1(config-if)# tunnel destination 2001:db8:acad:3::1
R1(config-if)# tunnel mode gre ipv6
R1(config-if)# exit
```

b. On R1, create a static route to 2001:db8:acad:1723::/64 via interface Tunnel 1.

```
R1(config)# ipv6 route 2001:db8:acad:1723::/64 tunnel 1
```

c. On R3, create interface Tunnel 1, specifying the IPv6 address 1002:db8:ffff::2/64, a tunnel source of Loopback0, and a tunnel destination of 2001:db8:acad:1::1.

```
R3(config)# interface tunnel 1
R3(config-if)# ipv6 address 2001:db8:ffff::2/64
R3(config-if)# tunnel source loopback 0
R3(config-if)# tunnel destination 2001:db8:acad:1::1
R3(config-if)# tunnel mode gre ipv6
R3(config-if)# exit
```

d. On R3, create a static route to 2001:db8:acad:1721::/64 via interface Tunnel 1.

```
R3(config)# ipv6 route 2001:db8:acad:1721::/64 tunnel 1
```

e. On R1, issue the command **show interface tunnel 1** and examine the output.

```
R1# show interface tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  MTU 1456 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 255/255, rxload 255/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 2001:DB8:ACAD:1::1 (Loopback0), destination 2001:DB8:ACAD:3::1
```

```
   Tunnel Subblocks:
      src-track:
         Tunnel1 source tracking subblock associated with Loopback0
          Set of tunnels with source Loopback0, 2 members (includes iterators),on
interface <OK>
  Tunnel protocol/transport GRE/IPv6
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Path MTU Discovery, ager 10 mins, min MTU 1280
  Tunnel transport MTU 1456 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input 00:00:31, output 00:01:01, output hang never
  Last clearing of "show interface" counters 00:06:58
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 367000 bits/sec, 395 packets/sec
  5 minute output rate 367000 bits/sec, 395 packets/sec
     246335 packets input, 28574884 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     246336 packets output, 28575000 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
```

f. From R1, **ping 2001:db8:acad:1723::1**. The pings should be successful.

Part 3: Configure and verify GRE Tunnels with Dynamic Routing

In Part 3, you will configure and verify GRE tunnels between R1 and R3, and you will use dynamic routing for overlay reachability and static routing for underlay reachability. You will configure two tunnels, one for IPv4 traffic and one of IPv6 traffic.

Step 1: Remove the Tunnel 0 and Tunnel 1 interfaces on R1 and R3.

Issue the command **no interface tunnel 0** and **no interface tunnel 1** on R1 and R3.

Step 2: Replace the OSPF configuration on R1, R2, and R3 with static routing.

a. On R1, R2, and R3, remove OSPF with the **no router ospf 4** and **no ipv6 router ospf 6** commands.

b. On R1 and R3, create IPv4 and IPv6 static default routes that point to R2.

c. On R2, create IPv4 and IPv6 static routes that point to R1 and R3 loopback 0 networks.

```
R2(config)# ip route 192.168.1.0 255.255.255.0 10.1.2.1
R2(config)# ip route 192.168.3.0 255.255.255.0 10.2.3.3
R2(config)# ipv6 route 2001:db8:acad:1::/64 2001:db8:acad:12::1
R2(config)# ipv6 route 2001:db8:acad:3::/64 2001:db8:acad:23::3
```

d. Verify that R1 can reach Loopback 0 on R3 with pings using a source address of the R1 Loopback 0 address.

```
R1# ping 192.168.3.1 source loopback 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms

R1# ping 2001:db8:acad:3::1 source loopback 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:3::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:ACAD:1::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Step 3: Create an IPv4-based GRE tunnel between R1 and R3.

a. On R1, create interface Tunnel 0, specifying the IP address 100.100.100.1/30, bandwidth of 4000 kbps, a tunnel source of Loopback0, and a tunnel destination of 192.168.3.1.

```
R1(config)# interface tunnel 0
R1(config-if)# ip address 100.100.100.1 255.255.255.252
R1(config-if)# bandwidth 4000
R1(config-if)# ip mtu 1400
R1(config-if)# tunnel source loopback 0
R1(config-if)# tunnel destination 192.168.3.1
R1(config-if)# exit
```

b. On R1, configure OSPFv2 process-id 4 with router-id 1.1.1.4, and use network statements or interface configuration commands to include interface Tunnel 0 in Area 0 and interface Loopback 1 in Area 1.

```
R1(config)# router ospf 4
R1(config-router)# router-id 1.1.1.4
R1(config-router)# network 100.100.100.0 0.0.0.3 area 0
R1(config-router)# network 172.16.1.0 0.0.0.255 area 1
R1(config-router)# exit
```

c. On R3, create interface Tunnel 0, specifying the IP address 100.100.100.2/30, bandwidth of 4000 kbps, a tunnel source of Loopback0, and a tunnel destination of 192.168.1.1.

```
R3(config)# interface tunnel 0
R3(config-if)# ip address 100.100.100.2 255.255.255.252
R3(config-if)# bandwidth 4000
R3(config-if)# ip mtu 1400
R3(config-if)# tunnel source loopback 0
R3(config-if)# tunnel destination 192.168.1.1
R3(config-if)# exit
```

d. On R3, configure OSPFv2 process-id 4 with router-id 3.3.3.4, and use network statements or interface configuration commands to include interface Tunnel 0 in Area 0 and interface Loopback 1 in Area 1.

```
R3(config)# router ospf 4
```

```
R3(config-router)# router-id 3.3.3.4
R3(config-router)# network 100.100.100.0 0.0.0.3 area 0
R3(config-router)# network 172.16.3.0 0.0.0.255 area 1
R3(config-router)# exit
```

e. On R1, issue the command **show interface tunnel 0** and examine the output.

```
R1# show interface tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 100.100.100.1/30
  MTU 9976 bytes, BW 4000 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 192.168.1.1 (Loopback0), destination 192.168.3.1
   Tunnel Subblocks:
      src-track:
         Tunnel0 source tracking subblock associated with Loopback0
          Set of tunnels with source Loopback0, 1 member (includes iterators), on
interface <OK>
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255, Fast tunneling enabled
  Tunnel transport MTU 1476 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input 00:00:01, output 00:00:04, output hang never
  Last clearing of "show interface" counters 00:06:11
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     23 packets input, 2064 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     58 packets output, 6784 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
```

f. On R1, issue the command **show ip route ospf** and verify that 172.16.3.0/24 appears in the routing table as an OSPF route.

```
R1# show ip route ospf | begin Gateway
Gateway of last resort is 10.1.2.2 to network 0.0.0.0

      172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
```

```
O IA     172.16.3.1/32 [110/26] via 100.100.100.2, 00:02:53, Tunnel0
```

g. From R1, ping 172.16.3.1. The pings should be successful.

Step 4: Create an IPv6-based GRE tunnel between R1 and R3.

a. On R1, create interface Tunnel 1, specifying the IPv6 address 2001:db8:ffff::1/64, bandwidth of 4000kbps, a tunnel source of Loopback0, and a tunnel destination of 2001:db8:acad:3::1.

```
R1(config)# interface tunnel 1
R1(config-if)# ipv6 address 2001:db8:ffff::1/64
R1(config-if)# bandwidth 4000
R1(config-if)# tunnel source loopback 0
R1(config-if)# tunnel destination 2001:db8:acad:3::1
R1(config-if)# tunnel mode gre ipv6
R1(config-if)# exit
```

b. On R1, configure OSPFv3 process-id 6 with router-id 1.1.1.6, and interface configuration commands to include interface Tunnel 0 in Area 0 and interface Loopback 1 in Area 1.

```
R1(config)# ipv6 router ospf 6
R1(config-rtr)# router-id 1.1.1.6
R1(config-rtr)# exit
R1(config)# interface tunnel 1
R1(config-if)# ipv6 ospf 6 area 0
R1(config-if)# exit
R1(config)# interface loopback 1
R1(config-if)# ipv6 ospf 6 area 1
R1(config-if)# exit
```

c. On R3, create interface Tunnel 1, specifying the IPv6 address 1002:db8:ffff::2/64, bandwidth of 4000kbps, a tunnel source of Loopback0, and a tunnel destination of 2001:db8:acad:1::1.

```
R3(config)# interface tunnel 1
R3(config-if)# ipv6 address 2001:db8:ffff::2/64
R3(config-if)# bandwidth 4000
R3(config-if)# tunnel source loopback 0
R3(config-if)# tunnel destination 2001:db8:acad:1::1
R3(config-if)# tunnel mode gre ipv6
R3(config-if)# exit
```

d. On R3, configure OSPFv3 process-id 6 with router-id 3.3.3.6, and use network statements or interface configuration commands to include interface Tunnel 0 in Area 0 and interface Loopback 1 in Area 1.

```
R3(config)# ipv6 router ospf 6
R3(config-rtr)# router-id 3.3.3.6
R3(config-rtr)# exit
R3(config)# interface tunnel 1
R3(config-if)# ipv6 ospf 6 area 0
R3(config-if)# exit
R3(config)# interface loopback 1
R3(config-if)# ipv6 ospf 6 area 1
R3(config-if)# exit
```

e. On R1, issue the command **show interface tunnel 1** and examine the output.

```
R1# show interface tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  MTU 1456 bytes, BW 4000 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 2001:DB8:ACAD:1::1 (Loopback0), destination 2001:DB8:ACAD:3::1
   Tunnel Subblocks:
      src-track:
         Tunnel1 source tracking subblock associated with Loopback0
          Set of tunnels with source Loopback0, 2 members (includes iterators),on
interface <OK>
  Tunnel protocol/transport GRE/IPv6
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Path MTU Discovery, ager 10 mins, min MTU 1280
  Tunnel transport MTU 1456 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input 00:00:09, output 00:00:04, output hang never
  Last clearing of "show interface" counters 00:04:20
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     31 packets input, 4048 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     46 packets output, 5864 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
```

f. On R1, issue the command **show ipv6 route ospf** and verify that 2001:db8:acad:1723::/64 appears in the routing table as an OSPF route.

```
R1# show ipv6 route ospf
IPv6 Routing Table - default - 11 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
       NDr - Redirect, RL - RPL, O - OSPF Intra, OI - OSPF Inter
       OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1
```

```
        ON2 - OSPF NSSA ext 2, a - Application
OI  2001:DB8:ACAD:1723::1/128 [110/25]
     via FE80::12B3:D6FF:FE04:ED10, Tunnel1
```

g. From R1, ping 2001:db8:acad:1723::1. The pings should be successful.

Part 4: Examine the Recursive Routing Problem with GRE

Recursive routing in overlay networks occurs when the router decides that the best interface to use to cross the underlay network is an interface that is a part of the overlay network. For example, if R1 decided, based on the routing table, that the best way to get to the tunnel destination is via the tunnel itself. Care must be taken during configuration of routing protocols to prevent this from occurring, as it will cause the overlay network to fail.

a. To demonstrate how easily this could occur, add network 192.168.1.0 to the OSPF configuration of R1.

```
R1(config)# router ospf 4
R1(config-router)# network 192.168.1.0 0.0.0.255 area 0
R1(config-router)# end
R1#
*Jan 24 18:48:41.437: %SYS-5-CONFIG_I: Configured from console by console
R1#
*Jan 24 18:49:17.345: %OSPF-5-ADJCHG: Process 4, Nbr 3.3.3.4 on Tunnel0 from FULL to
DOWN, Neighbor Down: Dead timer expired
R1#
*Jan 24 18:49:45.422: %OSPF-5-ADJCHG: Process 4, Nbr 3.3.3.4 on Tunnel0 from LOADING
to FULL, Loading Done
R1#
*Jan 24 18:50:25.620: %OSPF-5-ADJCHG: Process 4, Nbr 3.3.3.4 on Tunnel0 from FULL to
DOWN, Neighbor Down: Dead timer expired
R1#
```

b. R1 shows that the dead timer expires and then the adjacency tries to reset. Now look at what is being logged at R3.

```
Jan 27 00:02:56.654: %ADJ-5-PARENT: Midchain parent maintenance for IP midchain out of
Tunnel0 - looped chain attempting to stack
R3(config)#
*Jan 27 00:03:00.485: %TUN-5-RECURDOWN: Tunnel0 temporarily disabled due to recursive
routing
*Jan 27 00:03:01.485: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed
state to down
*Jan 27 00:03:01.486: %OSPF-5-ADJCHG: Process 4, Nbr 1.1.1.4 on Tunnel0 from FULL to
DOWN, Neighbor Down: Interface down or detached
```

c. As you can see, R3 recognizes the issue and even tells you there is a recursive routing problem. Fix this by removing the network statement on R1 and the tunnel will come back up.

```
R1(config)# router ospf 4
R1(config-router)# no network 192.168.1.0 0.0.0.255 area 0
R1(config-router)# end
R1#
*Jan 24 18:54:22.496: %SYS-5-CONFIG_I: Configured from console by console
R1#
*Jan 24 18:54:29.439: %OSPF-5-ADJCHG: Process 4, Nbr 3.3.3.4 on Tunnel0 from LOADING
to FULL, Loading Done
```

Router Interface Summary Table

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
4221	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
4300	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.