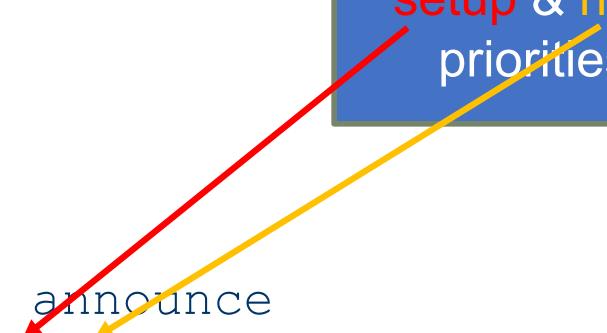


show mpls traffic-eng topology brief

MPLS TE – Dynamic Tunnels

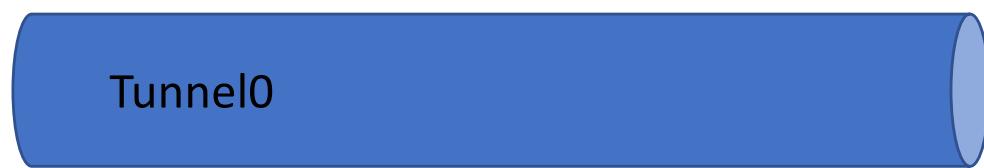
- Define a tunnel (headend) with dynamic path:
 - interface Tunnel1
 - ip unnumbered Loopback1
 - tunnel destination 172.16.1.5
 - tunnel mode mpls traffic-eng
 - tunnel mpls traffic-eng autoroute announce
 - tunnel mpls traffic-eng priority 5 5
 - tunnel mpls traffic-eng bandwidth 400
 - tunnel mpls traffic-eng path-option 1 dynamic
- Check tunnel status
 - show mpls traffic-eng tunnels brief

setup & hold
priorities



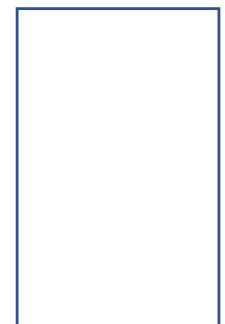
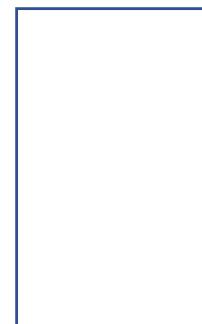
MPLS TE – Dynamic Tunnels (2)

shutdown interface s2/1 on R1, then play with the following priorities



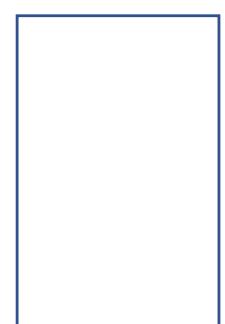
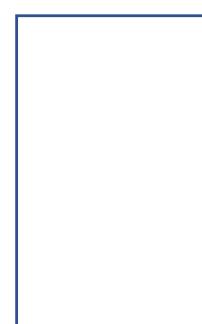
Setup

Hold

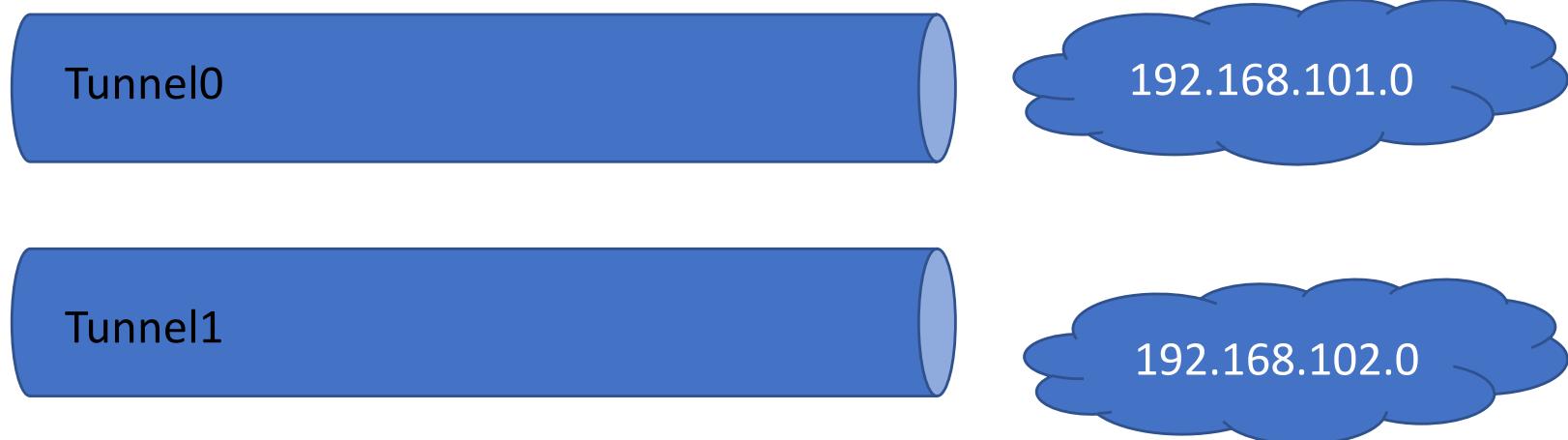


Setup

Hold



Differentiate routing



- To differentiate routing you can set static routes (e.g. for load balancing):
 - `ip route 192.168.101.0 255.255.255.0 Tunnel0`
 - `ip route 192.168.102.0 255.255.255.0 Tunnel1`

Test 1

- Generate tunnel0 (priority 2 2)
- Generate tunnel1 (priority 3 3)
- Turn on serial 2/1 on R1
- Increase priority of tunnel1 to 1 1
- Check tunnel status at each step
- Observe traffic

MPLS TE – Explicit-path Tunnels

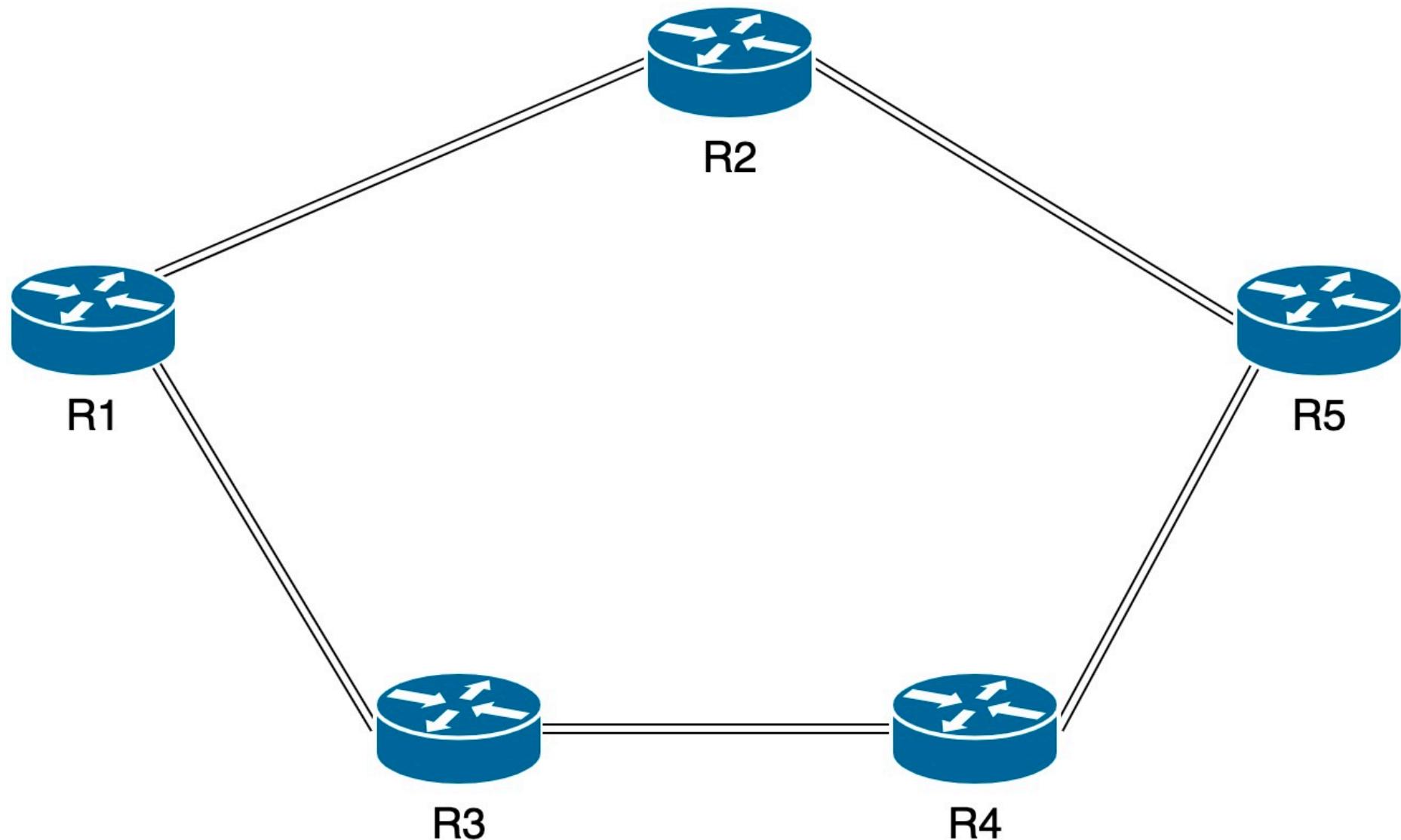
- Define a tunnel (headend) with explicit path:

```
interface Tunnel2
    ip unnumbered Loopback0
    tunnel destination 172.16.1.5
    tunnel mode mpls traffic-eng
    tunnel mpls traffic-eng autoroute announce
    tunnel mpls traffic-eng priority 2 2
    tunnel mpls traffic-eng bandwidth 400
    tunnel mpls traffic-eng path-option 1 explicit name longpath
    tunnel mpls traffic-eng path-option 2 dynamic bandwidth 400
    tunnel mpls traffic-eng path-option 3 dynamic bandwidth 0
```

- Define the path:

```
ip explicit-path name longpath enable
    next-address 10.2.2.2
    next-address 10.3.3.2
    next-address 10.4.4.2
```

```
tunnel mpls traffic-eng path-option 1 explicit name longpath  
tunnel mpls traffic-eng path-option 2 dynamic bandwidth 400  
tunnel mpls traffic-eng path-option 3 dynamic bandwidth 0
```



Test 2

- Remove either tunnel0 or tunnel1
- Create tunnel2
- Check where traffic is going

Test 2

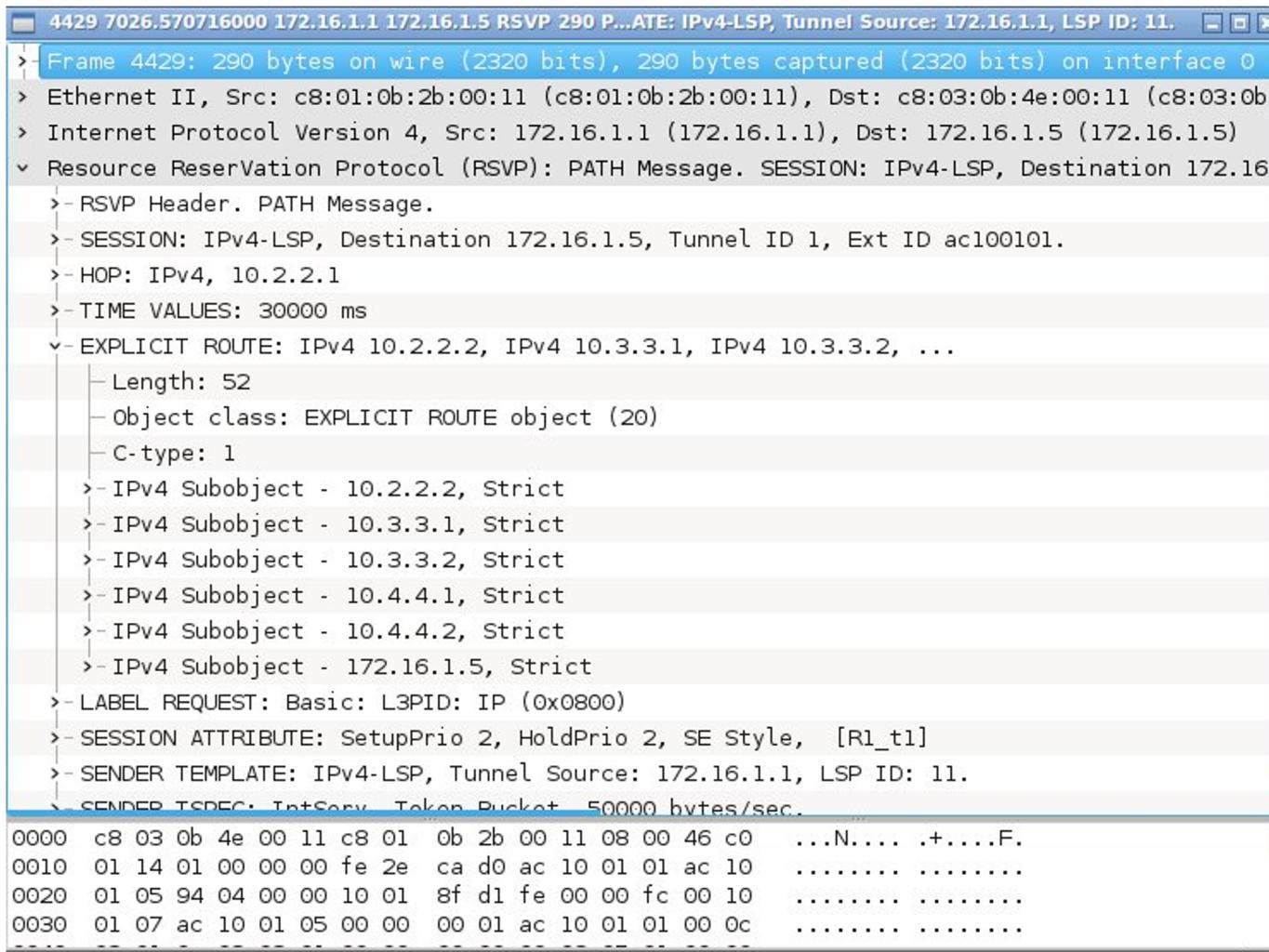
- show ip route

```
R1
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set

    172.16.0.0/32 is subnetted, 5 subnets
o    172.16.1.5 [110/21] via 0.0.0.0, 00:08:13, Tunnel1
o    172.16.1.4 [110/21] via 10.2.2.2, 00:08:13, Ethernet1/4
C    172.16.1.1 is directly connected, Loopback0
o    172.16.1.3 [110/11] via 10.2.2.2, 00:08:13, Ethernet1/1
o    172.16.1.2 [110/11] via 10.1.1.2, 00:08:13, Ethernet1/0
    10.0.0.0/30 is subnetted, 5 subnets
o      10.4.4.0 [110/30] via 0.0.0.0, 00:08:13, Tunnel1
          [110/30] via 10.2.2.2, 00:08:13, Ethernet1/1
o      10.5.5.0 [110/20] via 10.1.1.2, 00:08:14, Ethernet1/0
o      10.3.3.0 [110/20] via 10.2.2.2, 00:08:14, Ethernet1/1
C      10.1.1.0 is directly connected, Ethernet1/0
C      10.2.2.0 is directly connected, Ethernet1/1
o      192.168.102.0/24 [110/30] via 0.0.0.0, 00:08:15, Tunnel1
C      192.168.1.0/24 is directly connected, Ethernet0/0
o      192.168.103.0/24 [110/30] via 0.0.0.0, 00:08:15, Tunnel1
C      192.168.2.0/24 is directly connected, Ethernet1/2
C      192.168.3.0/24 is directly connected, Ethernet1/3
o      192.168.101.0/24 [110/30] via 0.0.0.0, 00:08:15, Tunnel1
R1#
```

Test 2

- RSVP messages can be seen



The screenshot shows a Wireshark capture of an RSVP message. The packet details pane at the bottom displays the raw hex and ASCII data of the message.

Frame 4429: 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits) on interface 0

> Ethernet II, Src: c8:01:0b:2b:00:11 (c8:01:0b:2b:00:11), Dst: c8:03:0b:4e:00:11 (c8:03:0b:4e:00:11)

> Internet Protocol Version 4, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.1.5 (172.16.1.5)

Resource Reservation Protocol (RSVP): PATH Message. SESSION: IPv4-LSP, Destination 172.16.1.5

-> RSVP Header. PATH Message.

-> SESSION: IPv4-LSP, Destination 172.16.1.5, Tunnel ID 1, Ext ID acl00101.

-> HOP: IPv4, 10.2.2.1

-> TIME VALUES: 30000 ms

-> EXPLICIT ROUTE: IPv4 10.2.2.2, IPv4 10.3.3.1, IPv4 10.3.3.2, ...

- Length: 52
- Object class: EXPLICIT ROUTE object (20)
- C-type: 1
- > IPv4 Subobject - 10.2.2.2, Strict
- > IPv4 Subobject - 10.3.3.1, Strict
- > IPv4 Subobject - 10.3.3.2, Strict
- > IPv4 Subobject - 10.4.4.1, Strict
- > IPv4 Subobject - 10.4.4.2, Strict
- > IPv4 Subobject - 172.16.1.5, Strict

-> LABEL REQUEST: Basic: L3PID: IP (0x0800)

-> SESSION ATTRIBUTE: SetupPrio 2, HoldPrio 2, SE Style, [R1_t1]

-> SENDER TEMPLATE: IPv4-LSP, Tunnel Source: 172.16.1.1, LSP ID: 11.

-> SENDER TSPEC: IntSrv_Token Bucket 50000 bytes/sec.

Hex	Dec	ASCII
0000	c8 03 0b 4e 00 11 c8 01 0b 2b 00 11 08 00 46 c0	...N.... .+....F.
0010	01 14 01 00 00 00 fe 2e ca d0 ac 10 01 01 ac 10
0020	01 05 94 04 00 00 10 01 8f d1 fe 00 00 fc 00 10
0030	01 07 ac 10 01 05 00 00 00 01 ac 10 01 01 00 0c

Test 2

- Open Wireshark on one of the links in the long path and try to ping a Server from a Client

4435 7029.188930000 192.168.1.2 192.168.101.2 ICMP 102 Echo 1 request id=0x1207, seq=1/256, ttl=63 (no response found!)

> Frame 4435: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface 0

> Ethernet II, Src: c8:01:0b:2b:00:11 (c8:01:0b:2b:00:11), Dst: c8:03:0b:4e:00:11 (c8:03:0b:4e:00:11)

> MultiProtocol Label Switching Header, Label: 16, Exp: 0, S: 1, TTL: 63

 0000 0000 0000 0001 0000 = MPLS Label: 16

 000. = MPLS Experimental Bits: 0

 1 = MPLS Bottom Of Label Stack: 1

 0011 1111 = MPLS TTL: 63

> Internet Protocol Version 4, Src: 192.168.1.2 (192.168.1.2), Dst: 192.168.101.2 (192.168.101.2)

> Internet Control Message Protocol

 Type: 8 (Echo (ping) request)

 Code: 0

 Checksum: 0x278b [correct]

 Identifier (BE): 4615 (0x1207)

 Identifier (LE): 1810 (0x0712)

 Sequence number (BE): 1 (0x0001)

 Sequence number (LE): 256 (0x0100)

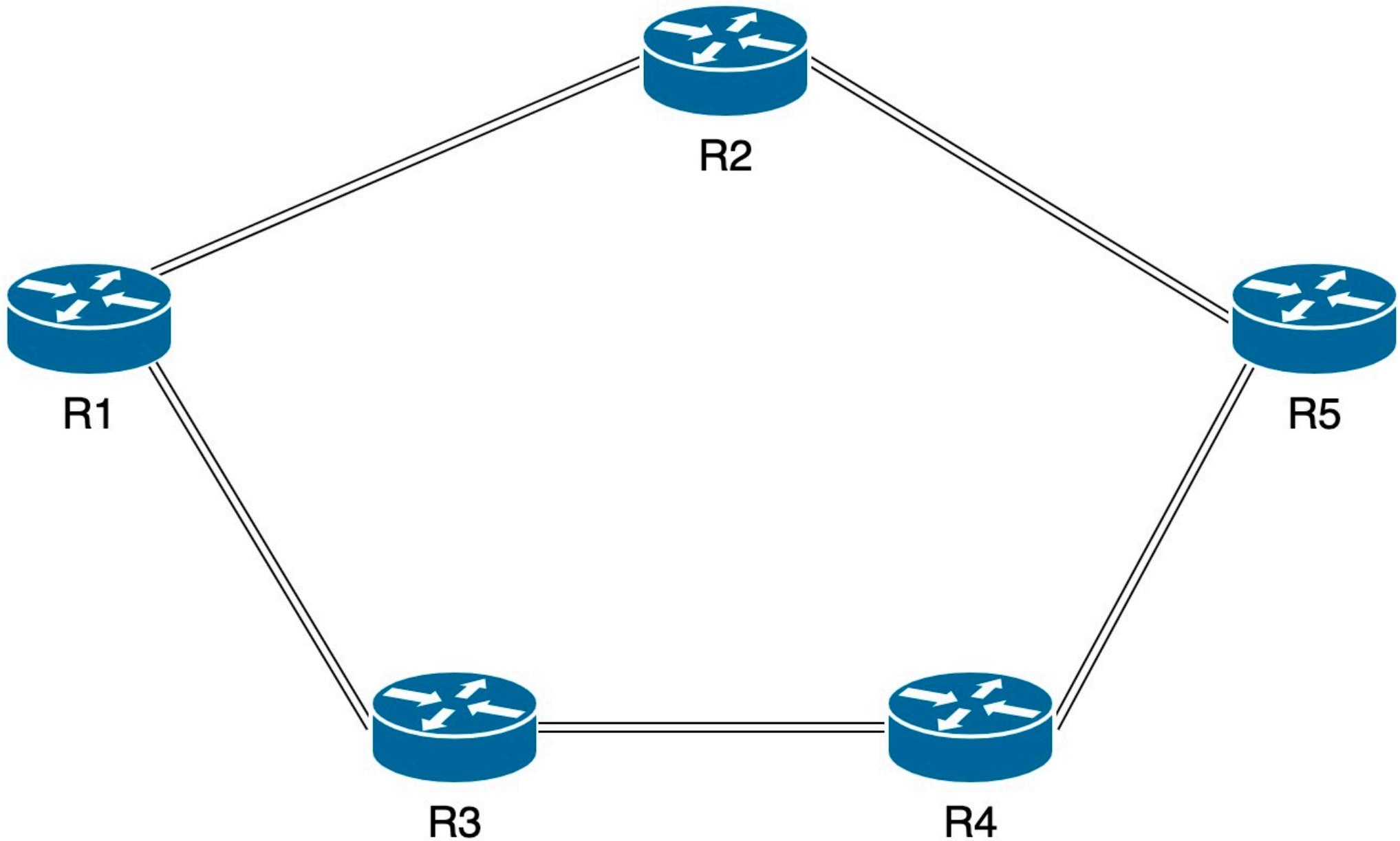
 > [No response seen]

 > Data (56 bytes)

0000 c8 03 0b 4e 00 11 c8 01 0b 2b 00 11 88 47 00 01 ...N.... .+...G..
0010 01 3f 45 00 00 54 00 00 40 00 3f 01 54 54 c0 a8 .?E..T.. @.?..TT..
0020 01 02 c0 a8 65 02 08 00 27 8b 12 07 00 01 81 cbe.... '.....
0030 3c a1 00 00 00 00 00 00 00 00 00 00 00 00 00 00 <.....

Response is
sent on the
short path!

Path protection



Path protection (2)

```
interface tunnel2
...
tunnel mpls traffic-eng path-option 1 dynamic bandwidth 400
tunnel mpls traffic-eng path-option protect 10 explicit name longpath
```

Test 3

- Create a tunnel between R1 and R5
- Use a dynamic path as primary
- Use an explicit path R1-R3-R4-R5 as secondary
- create a static route through the tunnel
- Observe traffic
- Shutdown any interface of R2
- Observe traffic again