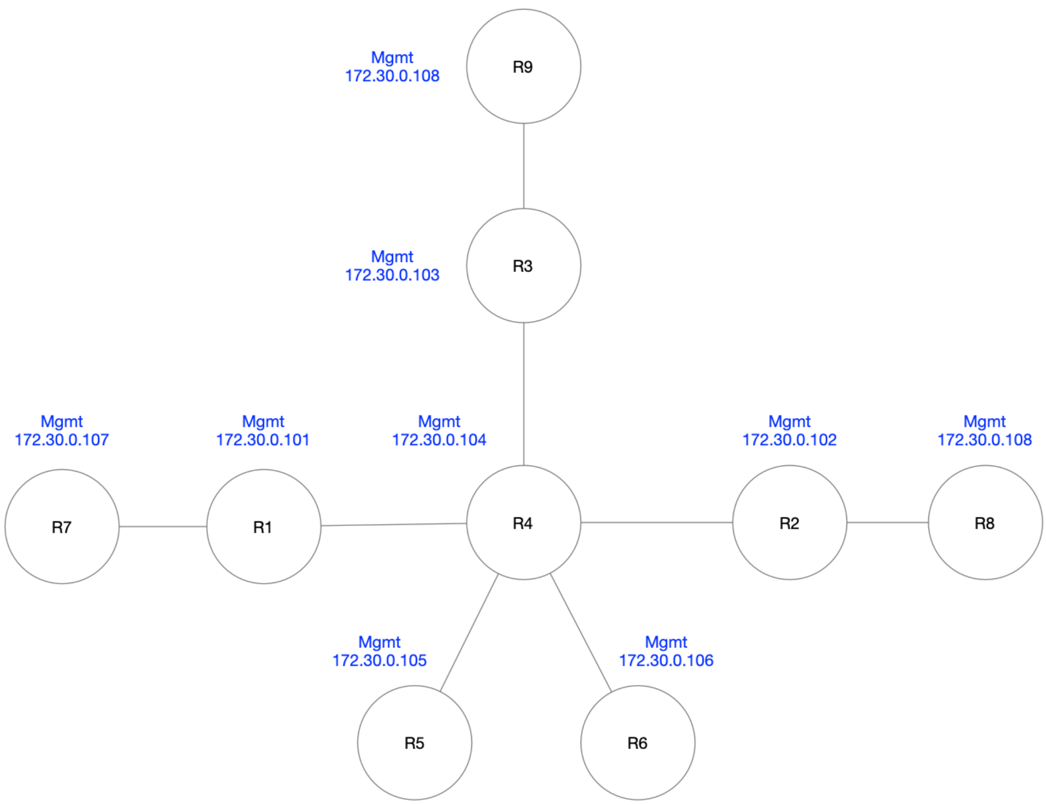
1 LAB Access

This content includes preconfigured users and components to illustrate the scripted scenarios and features of the solution. All components are fully configurable with predefined administrative user accounts. Logical topology is created on single server using XRd.

Logical Topology

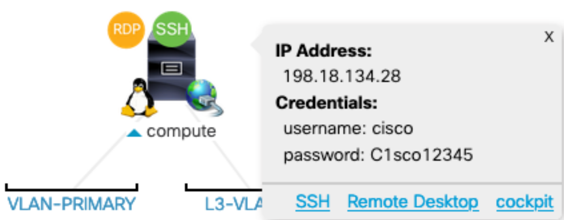


- Routers xr1,xr2 and xr3 represent PE routers in MPLS, xr4 is pure P device
- Routers xr5 and xr6 will act as route reflectors
- Routers xr7,xr8 and xr9 are CE routers, and they will be running the IPv4 protocol only.

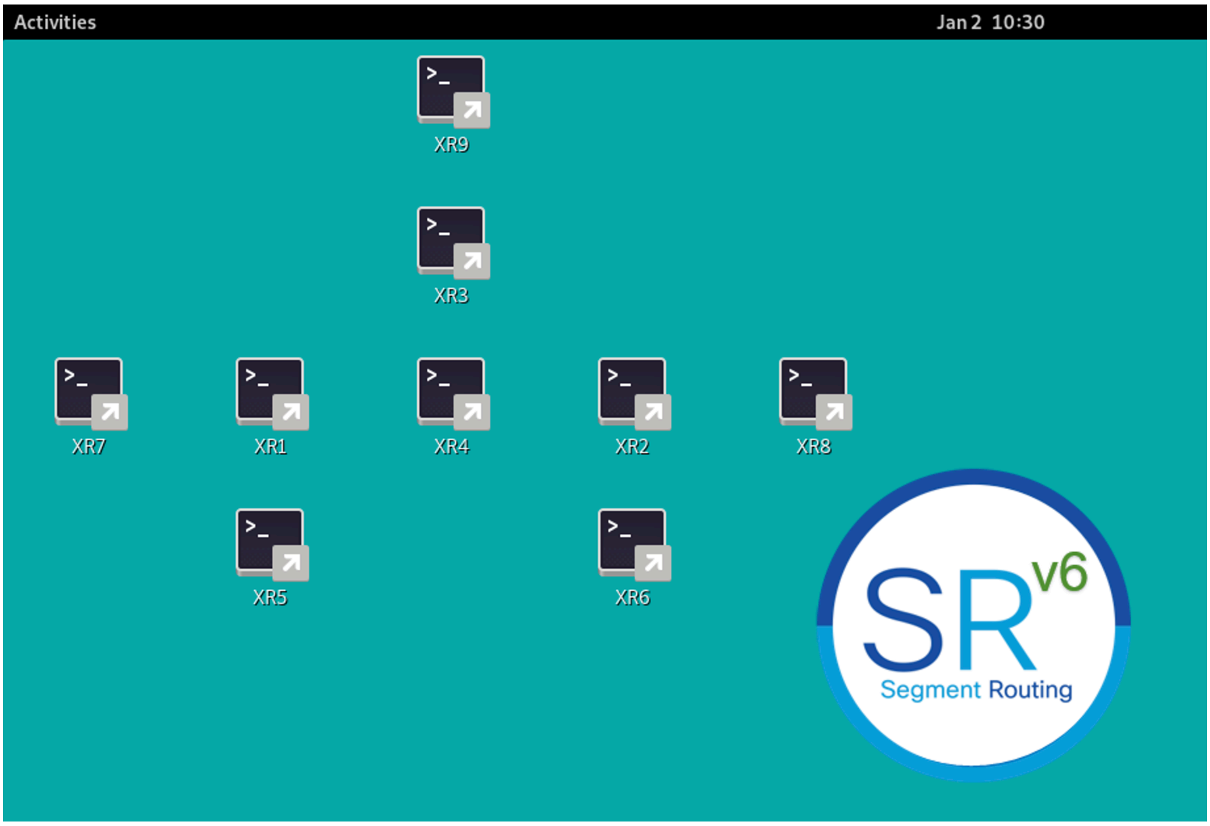
Equipment Access Details

Router	Username	Password	Mgmt IP
xr1	lab	lab	172.30.0.101
xr2	lab	lab	172.30.0.102
xr3	lab	lab	172.30.0.103
xr4	lab	lab	172.30.0.104
xr5	lab	lab	172.30.0.105
xr6	lab	lab	172.30.0.106
xr7	lab	lab	172.30.0.107
xr8	lab	lab	172.30.0.108
xr9	lab	lab	172.30.0.109

- Your Workstation **198.18.134.28**, has following credentials- Username: **cisco**, Password: **C1sco12345**
-
- There are several other ways to conect to the routers
  1. Web based

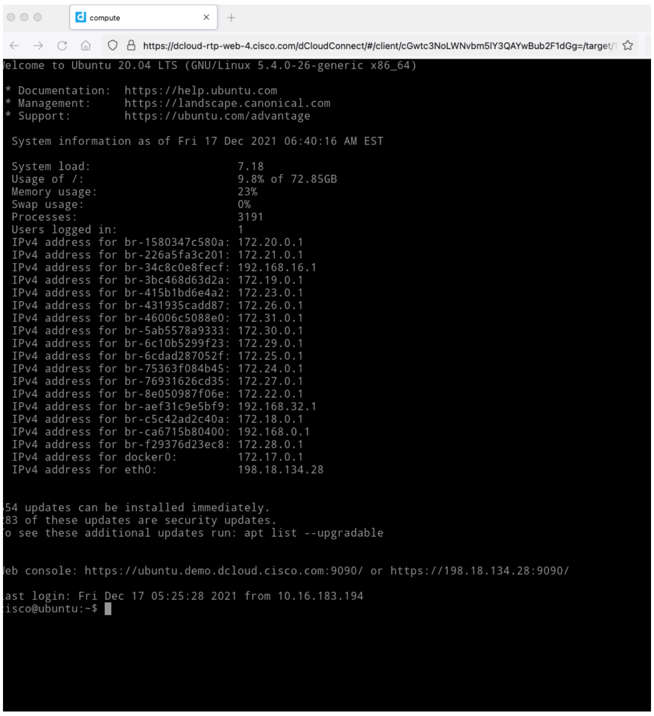


-
    a. RDP
- Click to your compute icon and select Remote Desktop
- Type password (C1sco12345)

- 

- Now you can double click to any xr icon and widow with particular router in exec mode will appear

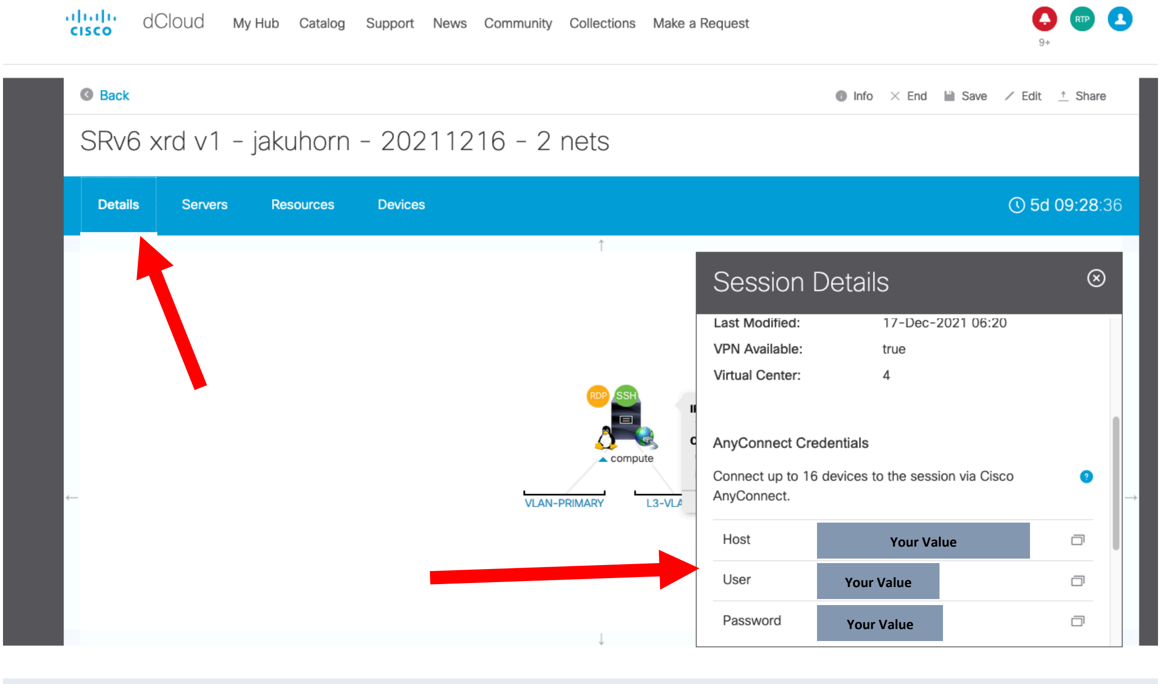
    b. SSH

- Click to your compute icon and select SSH
- 

- Now you can type ssh xr1 to access router xr1
- No password is needed.

2. VPN based

- Connect via Cisco Anyconect client
- Host to connect and VPN credential can be found in Details of you lab session



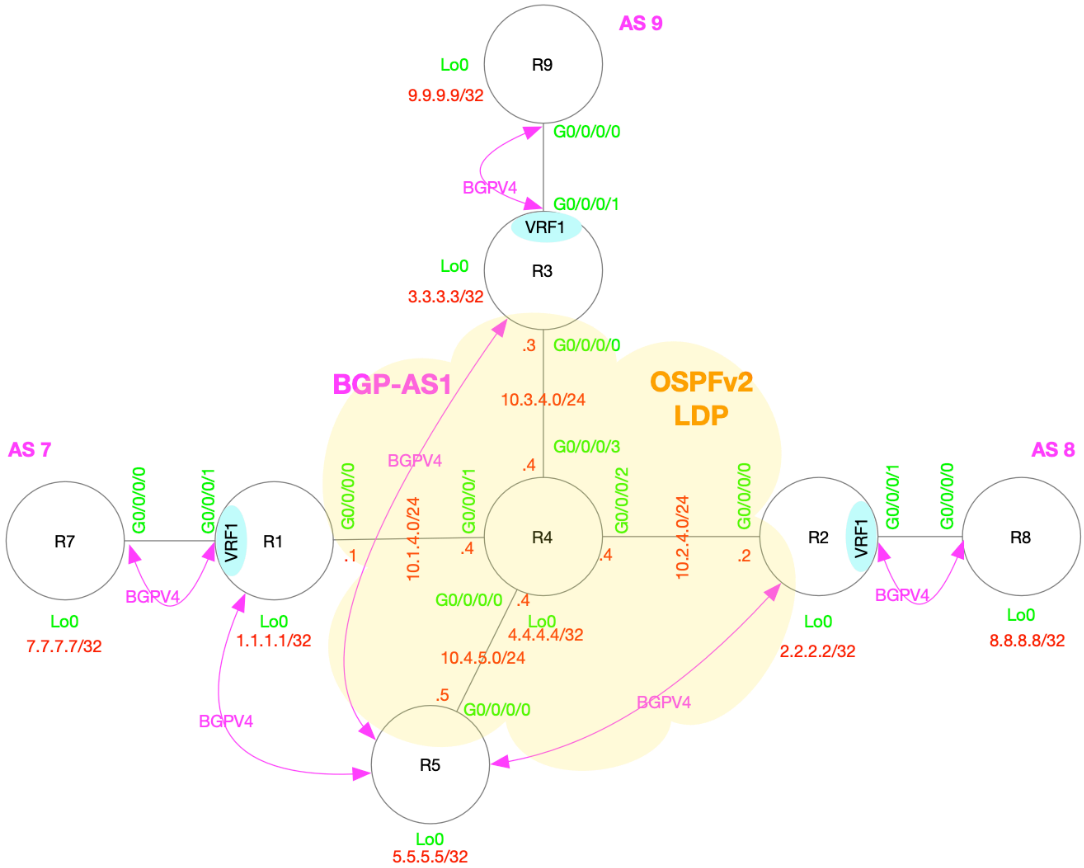
a. RDP

- Then you can use RDP client of your choice.
- Username is cisco, password C1sco12345

b. SSH

- Now you can use your prefered ssh client and connect to the server
- Username cisco, Password C1sco12345
- And from the server you can ssh to all routers – just type ssh xr1

2 Initial MPLS Network



Network is fully configured MPLS network

- IGP is OSPFv2 for IPv4 reachability
- LDP to distribute labels
- R1, R2 and R3 are PE routers with single vrf named “1”
- R4 is pure P router
- R5 act as Route Reflector for VPNv4 address family
- Each interface has IPv4 addressing
- Each router has loopback IPv4 address derived from id of the router so R4 has 4.4.4.4/32
- BGP AS is 1
- R7, R8 and R9 are CE routers each in different AS – 7,8 and 9 respectively
- PE-CE protocol is BGPv4

2.1 Task 0 Traffic

As a first step start 3 terminal windows connect to all 3 CE routers. R7, R8, R9.
Initiate continuous ping from loopback to loopback. Ensure that ping will last at least two hours and send icmp every 100ms

R7 to R8

```
RP/0/RP0/CPU0:xr7#ping 8.8.8.8 source 7.7.7.7 interval 100 repeat 40000
```

R8 to R9

```
RP/0/RP0/CPU0:xr8#ping 9.9.9.9 source 8.8.8.8 interval 100 repeat 40000
```

R9 to R7

```
RP/0/RP0/CPU0:xr9#ping 7.7.7.7 source 9.9.9.9 interval 100 repeat 40000
```

We will use those traffic streams to check connectivity and encapsulation during the migration.

2.2 Task 1- Check Label Allocation

Find out what is the label stack used on R1 to reach prefix 8.8.8.8/32

```
RP/0/RP0/CPU0:xr1#sh cef vrf 1 8.8.8.8
Tue Oct  4 13:00:07.692 UTC
8.8.8.8/32, version 8, internal 0x5000001 0x30 (ptr 0x8710ce98) [1], 0x0 (0x0),
0x208 (0x891841e8)
 Updated Oct  4 06:48:52.343
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
  gateway array (0x87e276b8) reference count 2, flags 0x2038, source rib (7), 0
backups
                [1 type 1 flags 0x48441 (0x891c75e8) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
  gateway array update type-time 1 Oct  4 06:48:52.343
 LDI Update time Oct  4 06:48:52.343
   via 2.2.2.2/32, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x87191960 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 2.2.2.2/32 via 24005/0/21
     next hop 10.1.4.4/32 Gi0/0/0/0    labels imposed {24001 24007}

    Load distribution: 0 (refcount 1)

    Hash  OK  Interface                 Address
    0     Y   recursive                 24005/0
```

Remember outer (transport label) we will use it for packet capture later. Here that would be 24001.

2.3 Task 1- Verify Label Stack with Packet Capture

In Docker environment you can do packet capture on any link. Links are represented by linux bridges. But you need to find out name of the bridge.

On server issue command “sudo docker network ls”

```
cisco@ubuntu:~$ sudo docker network ls
NETWORK ID     NAME                 DRIVER    SCOPE
f199bb97218b   bridge               bridge    local
e06ad94f1869   host                 host      local
1c1ef889b355   lab_mgmt             bridge    local
dc039a7076f7   none                 null      local
9c81bc2c3000   xr-1-gi0-xr-4-gi1    bridge    local
c6e02eea69e3   xr-1-gi0-xr-10-gi0   bridge    local
101702dafb87   xr-1-gi1-xr-2-gi0    bridge    local
7bf5044489c0   xr-1-gi1-xr-7-gi0    bridge    local
a73835587be7   xr-1-gi2-xr-3-gi0    bridge    local
c5c10b779190   xr-2-gi0-xr-4-gi2    bridge    local
8b239062174b   xr-2-gi1-xr-4-gi0    bridge    local
ccaa1f440f4a   xr-2-gi1-xr-8-gi0    bridge    local
b83d608c3a5b   xr-2-gi2-xr-5-gi1    bridge    local
3b006b2edbaf   xr-3-gi0-xr-4-gi3    bridge    local
bcd5ec0b4a9a   xr-3-gi1-xr-5-gi0    bridge    local
```

```
a7f8ac00dce2   xr-3-gi1-xr-9-gi0    bridge    local
0d2d14c80418   xr-3-gi2-xr-4-gi1    bridge    local
ec0ef79a855d   xr-4-gi2-xr-6-gi0    bridge    local
deb0feb3f46d   xr-4-gi3-xr-7-gi1    bridge    local
793b104c2ac1   xr-4-gi4-xr-9-gi0    bridge    local
20c6e92e8014   xr-5-gi0-xr-4-gi0    bridge    local
10a69a170445   xr-5-gi2-xr-7-gi0    bridge    local
735fd7ebe15c   xr-5-gi3-xr-6-gi1    bridge    local
0de71a15bcea   xr-6-gi0-xr-4-gi4    bridge    local
a3e4204610b3   xr-6-gi2-xr-8-gi1    bridge    local
b027f7a86cba   xr-6-gi3             bridge    local
14fac5f9ba94   xr-7-gi2-xr-8-gi2    bridge    local
a1f2423d6095   xr-8-gi0-xr-11-gi0   bridge    local
cisco@ubuntu:~$
```

You need to identify bridge id between router 1 and 4, in this case its ID is **9c81bc2c3000**
So we need to run tshark to capture packet on that bridge interface. We want to capture ICMP packet only and we want verbose output to see whole encapsulation. We also need transport label towards router 2 from previous task 24001
So our command will be "sudo tshark -i br-**9c81bc2c3000** -f "mpls **24001**" -V -c 1" (you need to replace **9c81bc2c3000** with your value if different, and label with your value if different) .

```
cisco@ubuntu:~$ sudo tshark -i br-9c81bc2c3000 -f "mpls 24001" -V -c 1
Running as user "root" and group "root". This could be dangerous.
Capturing on 'br-6355c3112450'
Frame 1: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on
interface br-6355c3112450, id 0
    Interface id: 0 (br-6355c3112450)
        Interface name: br-6355c3112450
    Encapsulation type: Ethernet (1)
    Arrival Time: Jan  3, 2023 04:25:36.217500627 EST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1672737936.217500627 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 122 bytes (976 bits)
    Capture Length: 122 bytes (976 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:mpls:ip:icmp:data]
Ethernet II, Src: 02:42:ac:18:00:02 (02:42:ac:18:00:02), Dst: 02:42:ac:18:00:03
(02:42:ac:18:00:03)
    Destination: 02:42:ac:18:00:03 (02:42:ac:18:00:03)
        Address: 02:42:ac:18:00:03 (02:42:ac:18:00:03)
        .... ..1. .... .... .... .... = LG bit: Locally administered address
(this is NOT the factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: 02:42:ac:18:00:02 (02:42:ac:18:00:02)
        Address: 02:42:ac:18:00:02 (02:42:ac:18:00:02)
        .... ..1. .... .... .... .... = LG bit: Locally administered address
(this is NOT the factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: MPLS label switched packet (0x8847)
```

```
MultiProtocol Label Switching Header, Label: 24001, Exp: 0, S: 0, TTL: 254
    0000 0101 1101 1100 0001 .... .... .... = MPLS Label: 24001
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...0 .... .... = MPLS Bottom Of Label Stack: 0
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
MultiProtocol Label Switching Header, Label: 24007, Exp: 0, S: 1, TTL: 254
    0000 0101 1101 1100 0111 .... .... .... = MPLS Label: 24007
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
Internet Protocol Version 4, Src: 7.7.7.7, Dst: 8.8.8.8
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        0000 00.. = Differentiated Services Codepoint: Default (0)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport
(0)
    Total Length: 100
    Identification: 0x3292 (12946)
    Flags: 0x0000
        0... .... .... .... = Reserved bit: Not set
        .0.. .... .... .... = Don't fragment: Not set
        ..0. .... .... .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 254
    Protocol: ICMP (1)
    Header checksum: 0x6be9 [validation disabled]
    [Header checksum status: Unverified]
    Source: 7.7.7.7
    Destination: 8.8.8.8
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x8546 [correct]
    [Checksum Status: Good]
    Identifier (BE): 5947 (0x173b)
    Identifier (LE): 15127 (0x3b17)
    Sequence number (BE): 12946 (0x3292)
    Sequence number (LE): 37426 (0x9232)
    Data (72 bytes)

0000  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0010  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0020  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0030  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0040  ab cd ab cd ab cd ab cd                           ........
        Data: abcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcd…
        [Length: 72]
```

So we see that packet here is MPLS encapsulated with transport and service labels according to sh cef command.

3 IPv6 Routing

3.1 Interface Addressing

For simplicity we just enable IPv6 on all core interfaces R1, R2, R3 on Gig0/0/0/0

R1, R2, R3 configuration:

```
configure

int gigabitEthernet 0/0/0/0
ipv6 enable

commit
end
```

We also need to enable IPv6 on R4 all interfaces towards PE routers So Gig0/0/0/1, Gig0/0/0/2, Gig0/0/0/3.

R4 configuration

```
configure

int gigabitEthernet 0/0/0/1
ipv6 enable
int gigabitEthernet 0/0/0/2
ipv6 enable
int gigabitEthernet 0/0/0/3
ipv6 enable

commit
end
```

For SRv6 services we will use independent Route Reflector, hence R5 – current RR does not need any IPv6 configuration.

3.2 Loopbacks

For Control plane traffic we will need IPv6 addresses on the loopbacks. We will configure IPv6 address 2001::X/128 on R1, R2, R3 and R4. Where X is router ID 1 for R1 etc. Address will use same loopback0 as IPv4 address.

Example for R4:

```
configure

int loopback0
ipv6 address 2001::4/128

commit
end
```

3.3 ISIS

We will use ISIS as routing protocol for SRv6.

NOTE: The **net** for each router will be 49.0000.0000.000**X**.00, where **X** is the number of particular router. For example, on Router 1, the net will be 49.0000.0000.000**1**.00, for Router 2 the net will be 49.0000.0000.000**2**.00, and so on.

This is example for router 4

```
configure
```

```
router isis 1

 net 49.0000.0000.0004.00
 address-family ipv6 unicast
  router-id lo0
  metric-style wide

 interface Loopback0
  address-family ipv6 unicast

 interface GigabitEthernet0/0/0/1
  point-to-point
  address-family ipv6 unicast

 interface GigabitEthernet0/0/0/2
  point-to-point
  address-family ipv6 unicast

interface GigabitEthernet0/0/0/3
  point-to-point
  address-family ipv6 unicast

commit
end
```

Routers R1, R2 and R3 will enable ISIS only on interfaces Gig0/0/0/0 and Loopback0
Example for R1:

```
configure

router isis 1

 net 49.0000.0000.0001.00
 address-family ipv6 unicast
  router-id lo0
  metric-style wide

 interface Loopback0
  address-family ipv6 unicast

 interface GigabitEthernet0/0/0/0
  point-to-point
  address-family ipv6 unicast

commit
end
```

Verification:
Now you should have full IPv6 reachability within core network.
Display IPv6 routing table on R4:

```
RP/0/RP0/CPU0:xr4# sh route ipv6
Tue Oct  4 14:08:03.351 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       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
```

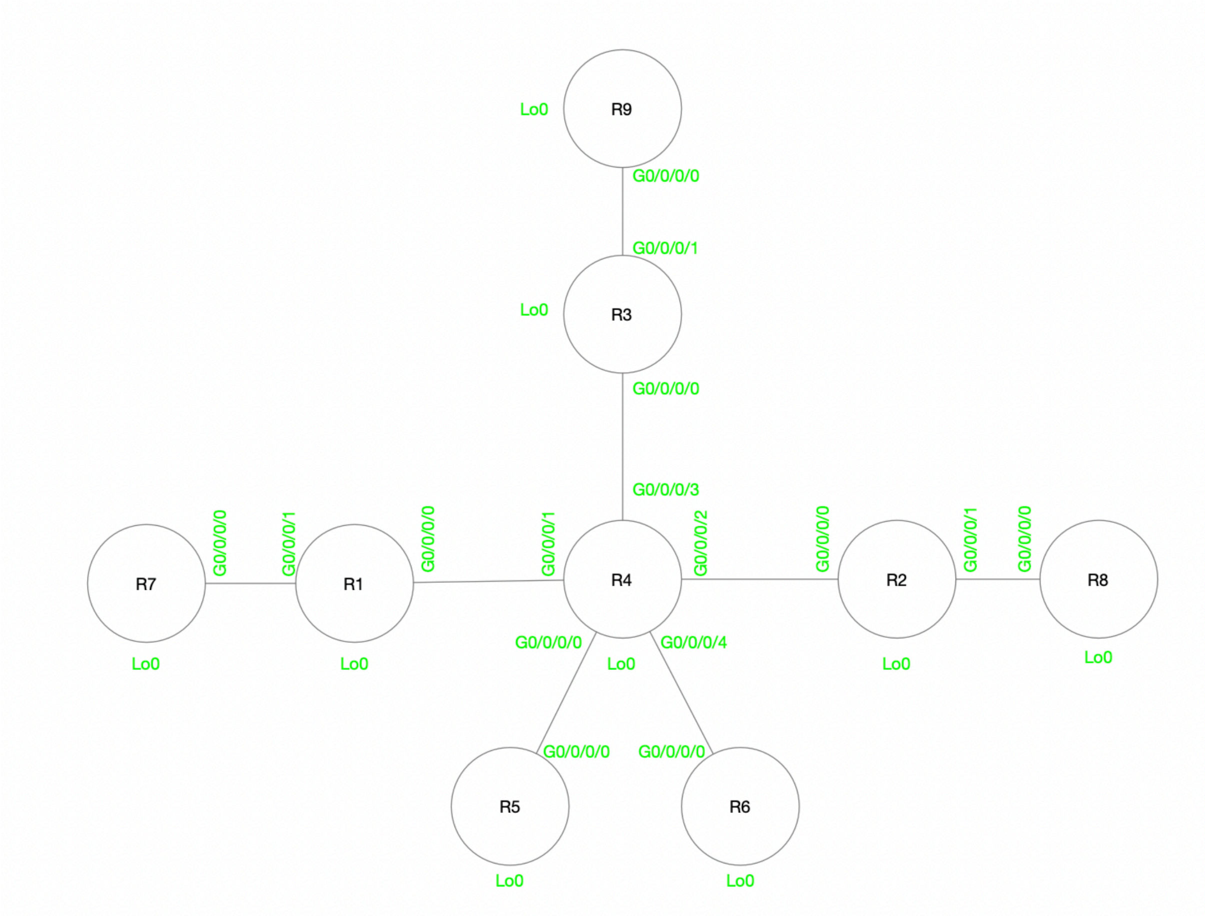
```
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, t - Traffic Engineering, (!) - FRR Backup path

Gateway of last resort is not set

i L1 2001::1/128
      [115/20] via fe80::42:c0ff:fea8:1002, 00:03:16, GigabitEthernet0/0/0/1
i L1 2001::2/128
      [115/20] via fe80::42:c0ff:fea8:3002, 00:03:16, GigabitEthernet0/0/0/2
i L1 2001::3/128
      [115/20] via fe80::42:c0ff:fea8:5002, 00:00:12, GigabitEthernet0/0/0/3
L    2001::4/128 is directly connected,
      00:11:40, Loopback0
RP/0/RP0/CPU0:xr4 #
```

All 4 loopbacks should be in routing table.
And if you issue ping from R1 you should be able to reach loopback of R2 and R3:

```
RP/0/RP0/CPU0:xr1# ping 2001::2
Tue Oct  4 14:13:24.645 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/6 ms
RP/0/RP0/CPU0:xr1# ping 2001::3
Tue Oct  4 14:13:28.219 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001::3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/3 ms
RP/0/RP0/CPU0:xr1#      [115/20] via fe80::42:c0ff:fea8:5002, 00:00:12,
```

3.4 Add v6 Route Reflector

For SRv6 services we need to enable new Route Reflector which will be handling all SRv6 based services. We will add to the network new router R6 It will be attached according to following diagram:



To make R6 part of the network we need to unshut and configure g0/0/0/4 with IPv6 enabled including ISIS on R4:

```
configure

interface GigabitEthernet0/0/0/4
 ipv6 enable
 no shutdown
 cdp
!
router isis 1
 interface GigabitEthernet0/0/0/4
  point-to-point
  address-family ipv6 unicast
  !
commit
end
```

And also, we need to fully configure R6 including ISIS (BGP will be configured later):

```
configure

interface Loopback0
 ipv6 address 2001::6/128

interface GigabitEthernet0/0/0/0
 ipv6 enable
 no shutdown
```

```
router isis 1
 net 49.0000.0000.0006.00
 address-family ipv6 unicast
  metric-style wide
  router-id Loopback0

 interface Loopback0
  address-family ipv6 unicast

 interface GigabitEthernet0/0/0/0
  point-to-point
  address-family ipv6 unicast

commit
end
```

You can verify routing table on R6:

```
RP/0/RP0/CPU0:xr6#sh route ipv6
Tue Oct  4 14:34:43.083 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       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 - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, t - Traffic Engineering, (!) - FRR Backup path

Gateway of last resort is not set

i L1 2001::1/128
      [115/30] via fe80::42:acff:fe1a:2, 00:00:46, GigabitEthernet0/0/0/0
i L1 2001::2/128
      [115/30] via fe80::42:acff:fe1a:2, 00:00:46, GigabitEthernet0/0/0/0
i L1 2001::3/128
      [115/30] via fe80::42:acff:fe1a:2, 00:00:46, GigabitEthernet0/0/0/0
i L1 2001::4/128
      [115/20] via fe80::42:acff:fe1a:2, 00:00:46, GigabitEthernet0/0/0/0
L    2001::6/128 is directly connected,
      00:02:04, Loopback0
RP/0/RP0/CPU0:xr6
```

You are supposed to see loopbacks of all other IPv6 routers over ISIS protocol.

4 Enabling SRv6

4.1 Create Locators

You need to configure locators on all PE routers

Enable SRv6 on all routers in the network by creating Locator. We will use ULA (unique local address) form block fcbb:bb00::/32.

Each locator will have following structure: fcbb:bb00:X::/48

Example for R1:

```
configure

segment-routing
 srv6
  locators
   locator MAIN
    micro-segment behavior unode psp-usd
    prefix fcbb:bb00:1::/48

commit
end
```

You do not need to configure Locator on R6, because it acts as RR and does not do any SRv6 function.
Locator on P router and RR is optional. Pure IPv6 routing is sufficient to allow end to end services. But in real network SRv6 functionalities are more than advisable for TI-LFA and traffic engineering to work.

4.2 Advertise Locators in ISIS

Locators must be advertised into IGP, hence we need to configure it on R1, R2 and R3:

```
configure

router isis 1
 address-family ipv6 unicast
  segment-routing srv6
   locator MAIN

commit
end
```

Verification:
All locators should be in routing table across SRv6 network so we can verify it on R4:

```
RP/0/RP0/CPU0:xr4#sh route ipv6
Tue Oct  4 14:50:30.781 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       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 - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, t - Traffic Engineering, (!) - FRR Backup path

Gateway of last resort is not set

i L1 2001::1/128
      [115/20] via fe80::42:c0ff:fea8:1002, 00:45:44, GigabitEthernet0/0/0/1
i L1 2001::2/128
      [115/20] via fe80::42:c0ff:fea8:3002, 00:45:44, GigabitEthernet0/0/0/2
i L1 2001::3/128
```

```
      [115/20] via fe80::42:c0ff:fea8:5002, 00:42:39, GigabitEthernet0/0/0/3
L    2001::4/128 is directly connected,
      00:54:08, Loopback0
i L1 2001::6/128
      [115/20] via fe80::42:acff:fe1a:3, 00:16:34, GigabitEthernet0/0/0/4
i L1 fcbb:bb00:1::/48
      [115/11] via fe80::42:c0ff:fea8:1002, 00:00:06, GigabitEthernet0/0/0/1
i L1 fcbb:bb00:2::/48
      [115/11] via fe80::42:c0ff:fea8:3002, 00:00:03, GigabitEthernet0/0/0/2
i L1 fcbb:bb00:3::/48
      [115/11] via fe80::42:c0ff:fea8:5002, 00:00:04, GigabitEthernet0/0/0/3
RP/0/RP0/CPU0:xr4#
```

5 Migration

For the migration towards SRv6, all PE routers must be connected t v6 RR and all services must be advertised via v6 RR. Then it is just matter of BGP best path selection to use MPLS or SRv6 received route.

5.1 BGP Sessions for SRv6

Here we will configure BGP session for first two routers we want to migrate R1 and R2:

```
configure

route-policy RRv6
  set local-preference 50
end-policy
!
router bgp 1
 neighbor 2001::6
  remote-as 1
  update-source Loopback0
  address-family vpnv4 unicast
   encapsulation-type srv6
   route-policy RRv6 out

commit
end
```

Also we need to configure BGP neighbors on v6 RR:

```
configure

router bgp 1
 bgp router-id 6.6.6.6
 address-family vpnv4 unicast
 !
 neighbor-group RRC
  remote-as 1
  update-source Loopback0
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
```

```
 neighbor 2001::1
  use neighbor-group RRC
 !
 neighbor 2001::2
  use neighbor-group RRC
commit
end
```

Enable SRv6 for VRF

R1 and R2:

```
configure

router bgp 1
 vrf 1
  address-family ipv4 unicast
   mpls alloc enable
   segment-routing srv6
    locator MAIN
    alloc mode per-vrf
   !
commit
end
```

Now we can observe that each prefix in VRF 1 between R1 and R2 is received twice. BGP then select best path based on Local Preference attribute.

R1:

```
RP/0/RP0/CPU0:xr1#sh bgp vpnv4 uni
Mon Jan  2 17:06:24.175 UTC
BGP router identifier 1.1.1.1, local AS number 1
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0
BGP main routing table version 24
BGP NSR Initial initsync version 5 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf 1)
*> 7.7.7.7/32         10.1.7.7                 0             0 7 i
*>i8.8.8.8/32         2.2.2.2                  0    100      0 8 i
* i                   2001::2                  0     50      0 8 i
*>i9.9.9.9/32         3.3.3.3                  0    100      0 9 i
*> 10.1.7.0/24        0.0.0.0                  0         32768 ?
*>i10.2.8.0/24        2.2.2.2                  0    100      0 ?
* i                   2001::2                  0     50      0 ?
*>i10.3.9.0/24        3.3.3.3                  0    100      0 ?

Processed 6 prefixes, 8 paths
RP/0/RP0/CPU0:xr1#
```

5.2 First Prefix

Now everything is ready to start moving individual VPN prefixes from MPLS to SRv6
First prefix to move will be loopback of CE7 hence 7.7.7.7/32
We will just change Local preference for that prefix in route-policy towards v6 RR on PE1:

```
configure

route-policy RRv6
  if destination in (7.7.7.7/32) then
    set local-preference 101
  else
    set local-preference 50
  endif
end-policy
!
commit
end
```

This configuration actually moved all traffic from PE2 towards prefix 7.7.7.7/32 in VRF1 to SRv6. But you shouldn't observe any packet drop in any CE terminal window (ping you started in the beginning):

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Immediately after commit we can verify forwarding on PE2:

```
RP/0/RP0/CPU0:xr2#sh bgp vpnv4 unicast
Mon Jan  2 17:18:28.949 UTC
BGP router identifier 2.2.2.2, local AS number 1
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0
BGP main routing table version 27
BGP NSR Initial initsync version 7 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
```

```
               i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf 1)
* i7.7.7.7/32         1.1.1.1                  0    100      0 7 i
*>i                   2001::1                  0    101      0 7 i
*> 8.8.8.8/32         10.2.8.8                 0             0 8 i
*>i9.9.9.9/32         3.3.3.3                  0    100      0 9 i
*>i10.1.7.0/24        1.1.1.1                  0    100      0 ?
* i                   2001::1                  0     50      0 ?
*> 10.2.8.0/24        0.0.0.0                  0         32768 ?
*>i10.3.9.0/24        3.3.3.3                  0    100      0 ?

Processed 6 prefixes, 8 paths
RP/0/RP0/CPU0:xr2#
```

And obviously we can check also cef on PE2:

```
RP/0/RP0/CPU0:xr2#sh cef vrf 1 7.7.7.7/32 detail
Mon Jan  2 17:23:05.931 UTC
7.7.7.7/32, version 21, SRv6 Headend, internal 0x5000001 0x30 (ptr 0x870fdf90)
[1], 0x0 (0x0), 0x0 (0x8983e2a0)
 Updated Jan  2 17:18:04.918
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
  gateway array (0x899040a8) reference count 1, flags 0x2010, source rib (7), 0
backups
                [1 type 3 flags 0x48441 (0x87d9ef48) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
  gateway array update type-time 1 Jan  2 17:18:04.919
 LDI Update time Jan  2 17:18:04.930

  Level 1 - Load distribution: 0
  [0] via fcbb:bb00:1::/128, recursive

   via fcbb:bb00:1::/128, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x87137648 0x0]
    next hop VRF - 'default', table - 0xe0800000
    next hop fcbb:bb00:1::/128 via fcbb:bb00:1::/48
    SRv6 H.Encaps.Red SID-list {fcbb:bb00:1:e001::}

    Load distribution: 0 (refcount 1)

    Hash  OK  Interface                 Address
    0     Y   GigabitEthernet0/0/0/0    fe80::42:c0ff:fea8:3003
```

Now, we should see different encapsulation for packets in each direction using tshark. So start tshark first and then send single ping from r7

Starting tshark:

```
cisco@ubuntu:~/LAB$ sudo tshark -i br-9c81bc2c3000 -Y ip.addr==7.7.7.7 -V
```

Stop capture after few seconds.

You should be able to see 4 types packets with different encapsulation to and from 7.7.7.7.

8.8.8.8 to 7.7.7.7 SRv6

7.7.7.7 to 8.8.8.8 MPLS

9.9.9.9 to 7.7.7.7 MPLS

7.7.7.7 to 9.9.9.9 MPLS

```
cisco@ubuntu:~/LAB$ sudo tshark -i br-9c81bc2c3000 -Y ip.addr==7.7.7.7 -V
```

```
Running as user "root" and group "root". This could be dangerous.
Capturing on 'br-9c81bc2c3000'
Frame 129: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits) on
interface br-9c81bc2c3000, id 0
    Interface id: 0 (br-9c81bc2c3000)
        Interface name: br-9c81bc2c3000
    Encapsulation type: Ethernet (1)
    Arrival Time: Jan  3, 2023 05:20:51.057026798 EST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1672741251.057026798 seconds
    [Time delta from previous captured frame: 0.000390847 seconds]
    [Time delta from previous displayed frame: 0.000390847 seconds]
    [Time since reference or first frame: 3.019817878 seconds]
    Frame Number: 129
    Frame Length: 154 bytes (1232 bits)
    Capture Length: 154 bytes (1232 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ipv6:ip:icmp:data]
Ethernet II, Src: 02:42:ac:18:00:03 (02:42:ac:18:00:03), Dst: 02:42:ac:18:00:02
(02:42:ac:18:00:02)
    Destination: 02:42:ac:18:00:02 (02:42:ac:18:00:02)
        Address: 02:42:ac:18:00:02 (02:42:ac:18:00:02)
        .... ..1. .... .... .... .... = LG bit: Locally administered address
(this is NOT the factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: 02:42:ac:18:00:03 (02:42:ac:18:00:03)
        Address: 02:42:ac:18:00:03 (02:42:ac:18:00:03)
        .... ..1. .... .... .... .... = LG bit: Locally administered address
(this is NOT the factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: 2001::2, Dst: fcbb:bb00:1:e001::
    0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0,
ECN: Not-ECT)
        .... 0000 00.. .... .... .... .... .... = Differentiated Services
Codepoint: Default (0)
        .... .... ..00 .... .... .... .... .... = Explicit Congestion
Notification: Not ECN-Capable Transport (0)
    .... .... .... 0000 0000 0000 0000 0000 = Flow Label: 0x00000
    Payload Length: 100
    Next Header: IPIP (4)
    Hop Limit: 63
    Source: 2001::2
    Destination: fcbb:bb00:1:e001::
    [Source Teredo Server IPv4: 0.0.0.0]
    [Source Teredo Port: 65535]
    [Source Teredo Client IPv4: 255.255.255.253]
Internet Protocol Version 4, Src: 8.8.8.8, Dst: 7.7.7.7
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        0000 00.. = Differentiated Services Codepoint: Default (0)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport
(0)
    Total Length: 100
```

```
    Identification: 0x0bef (3055)
    Flags: 0x0000
        0... .... .... .... = Reserved bit: Not set
        .0.. .... .... .... = Don't fragment: Not set
        ..0. .... .... .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 254
    Protocol: ICMP (1)
    Header checksum: 0x928c [validation disabled]
    [Header checksum status: Unverified]
    Source: 8.8.8.8
    Destination: 7.7.7.7
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xa256 [correct]
    [Checksum Status: Good]
    Identifier (BE): 10446 (0x28ce)
    Identifier (LE): 52776 (0xce28)
    Sequence number (BE): 3055 (0x0bef)
    Sequence number (LE): 61195 (0xef0b)
    [Request frame: 126]
    [Response time: 2.585 ms]
    Data (72 bytes)

0000  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0010  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0020  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0030  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0040  ab cd ab cd ab cd ab cd                           ........
        Data: abcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcd…
        [Length: 72]
```

Frame 130: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on

```
interface br-9c81bc2c3000, id 0
    Interface id: 0 (br-9c81bc2c3000)
        Interface name: br-9c81bc2c3000450
    Encapsulation type: Ethernet (1)
    Arrival Time: Jan  3, 2023 05:20:51.155021446 EST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1672741251.155021446 seconds
    [Time delta from previous captured frame: 0.097994648 seconds]
    [Time delta from previous displayed frame: 0.097994648 seconds]
    [Time since reference or first frame: 3.117812526 seconds]
    Frame Number: 130
    Frame Length: 118 bytes (944 bits)
    Capture Length: 118 bytes (944 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:mpls:ip:icmp:data]
Ethernet II, Src: 02:42:ac:18:00:03 (02:42:ac:18:00:03), Dst: 02:42:ac:18:00:02
(02:42:ac:18:00:02)
    Destination: 02:42:ac:18:00:02 (02:42:ac:18:00:02)
        Address: 02:42:ac:18:00:02 (02:42:ac:18:00:02)
        .... ..1. .... .... .... .... = LG bit: Locally administered address
(this is NOT the factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
```

```
    Source: 02:42:ac:18:00:03 (02:42:ac:18:00:03)
        Address: 02:42:ac:18:00:03 (02:42:ac:18:00:03)
        .... ..1. .... .... .... .... = LG bit: Locally administered address
(this is NOT the factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: MPLS label switched packet (0x8847)
MultiProtocol Label Switching Header, Label: 24007, Exp: 0, S: 1, TTL: 253
    0000 0101 1101 1100 0111 .... .... .... = MPLS Label: 24007
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1101 = MPLS TTL: 253
Internet Protocol Version 4, Src: 9.9.9.9, Dst: 7.7.7.7
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        0000 00.. = Differentiated Services Codepoint: Default (0)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport
(0)
    Total Length: 100
    Identification: 0x0c1c (3100)
    Flags: 0x0000
        0... .... .... .... = Reserved bit: Not set
        .0.. .... .... .... = Don't fragment: Not set
        ..0. .... .... .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 254
    Protocol: ICMP (1)
    Header checksum: 0x905d [validation disabled]
    [Header checksum status: Unverified]
    Source: 9.9.9.9
    Destination: 7.7.7.7
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x9a49 [correct]
    [Checksum Status: Good]
    Identifier (BE): 10414 (0x28ae)
    Identifier (LE): 44584 (0xae28)
    Sequence number (BE): 3100 (0x0c1c)
    Sequence number (LE): 7180 (0x1c0c)
    Data (72 bytes)

0000  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0010  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0020  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0030  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0040  ab cd ab cd ab cd ab cd                           ........
        Data: abcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcd…
        [Length: 72]

Frame 131: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on
interface br-6355c3112450, id 0
    Interface id: 0 (br-9c81bc2c3000)
        Interface name: br-9c81bc2c3000
    Encapsulation type: Ethernet (1)
    Arrival Time: Jan  3, 2023 05:20:51.155432658 EST
    [Time shift for this packet: 0.000000000 seconds]
```

```
    Epoch Time: 1672741251.155432658 seconds
    [Time delta from previous captured frame: 0.000411212 seconds]
    [Time delta from previous displayed frame: 0.000411212 seconds]
    [Time since reference or first frame: 3.118223738 seconds]
    Frame Number: 131
    Frame Length: 122 bytes (976 bits)
    Capture Length: 122 bytes (976 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:mpls:ip:icmp:data]
Ethernet II, Src: 02:42:ac:18:00:02 (02:42:ac:18:00:02), Dst: 02:42:ac:18:00:03
(02:42:ac:18:00:03)
    Destination: 02:42:ac:18:00:03 (02:42:ac:18:00:03)
        Address: 02:42:ac:18:00:03 (02:42:ac:18:00:03)
        .... ..1. .... .... .... .... = LG bit: Locally administered address
(this is NOT the factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: 02:42:ac:18:00:02 (02:42:ac:18:00:02)
        Address: 02:42:ac:18:00:02 (02:42:ac:18:00:02)
        .... ..1. .... .... .... .... = LG bit: Locally administered address
(this is NOT the factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: MPLS label switched packet (0x8847)
MultiProtocol Label Switching Header, Label: 24001, Exp: 0, S: 0, TTL: 254
    0000 0101 1101 1100 0001 .... .... .... = MPLS Label: 24001
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...0 .... .... = MPLS Bottom Of Label Stack: 0
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
MultiProtocol Label Switching Header, Label: 24007, Exp: 0, S: 1, TTL: 254
    0000 0101 1101 1100 0111 .... .... .... = MPLS Label: 24007
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
Internet Protocol Version 4, Src: 7.7.7.7, Dst: 8.8.8.8
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        0000 00.. = Differentiated Services Codepoint: Default (0)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport
(0)
    Total Length: 100
    Identification: 0x0bf0 (3056)
    Flags: 0x0000
        0... .... .... .... = Reserved bit: Not set
        .0.. .... .... .... = Don't fragment: Not set
        ..0. .... .... .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 254
    Protocol: ICMP (1)
    Header checksum: 0x928b [validation disabled]
    [Header checksum status: Unverified]
    Source: 7.7.7.7
    Destination: 8.8.8.8
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x9a55 [correct]
```

```
    [Checksum Status: Good]
    Identifier (BE): 10446 (0x28ce)
    Identifier (LE): 52776 (0xce28)
    Sequence number (BE): 3056 (0x0bf0)
    Sequence number (LE): 61451 (0xf00b)
    Data (72 bytes)

0000  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0010  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0020  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0030  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0040  ab cd ab cd ab cd ab cd                           ........
        Data: abcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcd…
        [Length: 72]
```

Frame 132: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on

```
interface br-9c81bc2c3000, id 0
    Interface id: 0 (br-9c81bc2c3000)
        Interface name: br-9c81bc2c3000
    Encapsulation type: Ethernet (1)
    Arrival Time: Jan  3, 2023 05:20:51.156495965 EST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1672741251.156495965 seconds
    [Time delta from previous captured frame: 0.001063307 seconds]
    [Time delta from previous displayed frame: 0.001063307 seconds]
    [Time since reference or first frame: 3.119287045 seconds]
    Frame Number: 132
    Frame Length: 122 bytes (976 bits)
    Capture Length: 122 bytes (976 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:mpls:ip:icmp:data]
Ethernet II, Src: 02:42:ac:18:00:02 (02:42:ac:18:00:02), Dst: 02:42:ac:18:00:03
(02:42:ac:18:00:03)
    Destination: 02:42:ac:18:00:03 (02:42:ac:18:00:03)
        Address: 02:42:ac:18:00:03 (02:42:ac:18:00:03)
        .... ..1. .... .... .... .... = LG bit: Locally administered address
(this is NOT the factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: 02:42:ac:18:00:02 (02:42:ac:18:00:02)
        Address: 02:42:ac:18:00:02 (02:42:ac:18:00:02)
        .... ..1. .... .... .... .... = LG bit: Locally administered address
(this is NOT the factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: MPLS label switched packet (0x8847)
MultiProtocol Label Switching Header, Label: 24003, Exp: 0, S: 0, TTL: 254
    0000 0101 1101 1100 0011 .... .... .... = MPLS Label: 24003
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...0 .... .... = MPLS Bottom Of Label Stack: 0
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
MultiProtocol Label Switching Header, Label: 24007, Exp: 0, S: 1, TTL: 254
    0000 0101 1101 1100 0111 .... .... .... = MPLS Label: 24007
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
Internet Protocol Version 4, Src: 7.7.7.7, Dst: 9.9.9.9
    0100 .... = Version: 4
```

```
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        0000 00.. = Differentiated Services Codepoint: Default (0)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport
(0)
    Total Length: 100
    Identification: 0x0c1c (3100)
    Flags: 0x0000
        0... .... .... .... = Reserved bit: Not set
        .0.. .... .... .... = Don't fragment: Not set
        ..0. .... .... .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 254
    Protocol: ICMP (1)
    Header checksum: 0x905d [validation disabled]
    [Header checksum status: Unverified]
    Source: 7.7.7.7
    Destination: 9.9.9.9
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xa249 [correct]
    [Checksum Status: Good]
    Identifier (BE): 10414 (0x28ae)
    Identifier (LE): 44584 (0xae28)
    Sequence number (BE): 3100 (0x0c1c)
    Sequence number (LE): 7180 (0x1c0c)
    [Request frame: 130]
    [Response time: 1.475 ms]
    Data (72 bytes)

0000  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0010  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0020  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0030  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0040  ab cd ab cd ab cd ab cd                           ........
        Data: abcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcd…
        [Length: 72]
```

5.3 First PE

once we verified everything is working, we can proceed with other prefixes or VPNs or PEs or address families based on overall migration strategy. Here for simplicity, we will do migration per PE

next, we will migrate whole PE1:

```
configure

route-policy RRv6
  set local-preference 101
end-policy
!
commit
end
```

Now all traffic VPNv4 from PE2(because it is SRv6 enabled) towards PE1 will be SRv6 encapsulated. Check on PE2:

```
RP/0/RP0/CPU0:xr2#sh bgp vpnv4 unicast
Mon Jan  2 17:58:55.491 UTC
BGP router identifier 2.2.2.2, local AS number 1
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0
BGP main routing table version 28
BGP NSR Initial initsync version 7 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf 1)
* i7.7.7.7/32         1.1.1.1                  0    100      0 7 i
*>i                   2001::1                  0    101      0 7 i
*> 8.8.8.8/32         10.2.8.8                 0             0 8 i
*>i9.9.9.9/32         3.3.3.3                  0    100      0 9 i
* i10.1.7.0/24        1.1.1.1                  0    100      0 ?
*>i                   2001::1                  0    101      0 ?
*> 10.2.8.0/24        0.0.0.0                  0         32768 ?
*>i10.3.9.0/24        3.3.3.3                  0    100      0 ?

Processed 6 prefixes, 8 paths
RP/0/RP0/CPU0:xr2#
```

But traffic from and to PE3 will be still MPLS, because PE3 is not SRv6 enabled. Check on PE3:

```
RP/0/RP0/CPU0:xr3#sh bgp vpnv4 unicast
Mon Jan  2 18:00:59.708 UTC
BGP router identifier 3.3.3.3, local AS number 1
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0
BGP main routing table version 16
BGP NSR Initial initsync version 7 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf 1)
*>i7.7.7.7/32         1.1.1.1                  0    100      0 7 i
*>i8.8.8.8/32         2.2.2.2                  0    100      0 8 i
*> 9.9.9.9/32         10.3.9.9                 0             0 9 i
*>i10.1.7.0/24        1.1.1.1                  0    100      0 ?
*>i10.2.8.0/24        2.2.2.2                  0    100      0 ?
```

```
*> 10.3.9.0/24        0.0.0.0                  0         32768 ?

Processed 6 prefixes, 6 paths
RP/0/RP0/CPU0:xr3#
```

Still 0 packet loss expected in CE ping!!

5.4 Second PE

Now we do exactly same procedure on PE2:

```
configure

route-policy RRv6
  set local-preference 101
end-policy
!
commit
end
```

now everything between PE1 and PE2 is SRv6 only.
PE3 is still MPLS only
0 packet loss expected!

5.5 Last PE

PE3 is currently MPLS only. All traffic from and to PE3 is MPLS only because we didn't enable any SRv6 service there and it is not connected to v6. route reflector
So we prepare R6:

```
configure

router bgp 1
 !
 neighbor 2001::3
  use neighbor-group RRC
commit
end
```

And now we configure R3 to enable SRv6 functionality for VPN, but keeping MPLS dataplane prefered :

```
configure

route-policy RRv6
  set local-preference 50
end-policy
!
router bgp 1
 neighbor 2001::6
  remote-as 1
  update-source Loopback0
  address-family vpnv4 unicast
   encapsulation-type srv6
   route-policy RRv6 out

 vrf 1
  address-family ipv4 unicast
   mpls alloc enable
   segment-routing srv6
    locator MAIN
    alloc mode per-vrf
```

```
   !
commit
end
```

Now you can see that on R1 it receives prefixes from R3 with SRv6 and MPLS encapsulation but MPLS is still. Prefixes received from R2 use SRv6:

```
RP/0/RP0/CPU0:xr1#sh bgp vpnv4 unicast
Mon Jan  2 20:10:40.079 UTC
BGP router identifier 1.1.1.1, local AS number 1
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0
BGP main routing table version 26
BGP NSR Initial initsync version 5 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf 1)
*> 7.7.7.7/32         10.1.7.7                 0             0 7 i
*>i8.8.8.8/32         2.2.2.2                  0    100      0 8 i
* i                   2001::2                  0     50      0 8 i
*>i9.9.9.9/32         3.3.3.3                  0    100      0 9 i
* i                   2001::3                  0     50      0 9 i
*> 10.1.7.0/24        0.0.0.0                  0         32768 ?
*>i10.2.8.0/24        2.2.2.2                  0    100      0 ?
* i                   2001::2                  0     50      0 ?
*>i10.3.9.0/24        3.3.3.3                  0    100      0 ?
* i                   2001::3                  0     50      0 ?

Processed 6 prefixes, 10 paths
RP/0/RP0/CPU0:xr1#
```

All traffic between R3 and R1 is running MPLS.
We can start wireshark:

```
sudo tshark -i br-9c81bc2c3000 -Y mpls
```

Start continuous ping from R9 towards R7 with 100ms between each icmp:
Ping between CE should be still running so we should see some packets here:

```
cisco@ubuntu:~$ sudo tshark -i br-9c81bc2c3000 -f "mpls"
Running as user "root" and group "root". This could be dangerous.
Capturing on 'br-9c81bc2c3000'
    2 0.001366884      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=283/6913, ttl=254 (request in 1)
    4 0.118824046      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=284/7169, ttl=254 (request in 3)
    6 0.202651466      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=285/7425, ttl=254 (request in 5)
    8 0.302608222      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=286/7681, ttl=254 (request in 7)
   10 0.403814203      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=287/7937, ttl=254 (request in 9)
   12 0.504236720      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=288/8193, ttl=254 (request in 11)
```

```
   14 0.604608703      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=289/8449, ttl=254 (request in 13)
   16 0.704480599      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=290/8705, ttl=254 (request in 15)
   18 0.805782504      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=291/8961, ttl=254 (request in 17)
   20 0.906456439      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=292/9217, ttl=254 (request in 19)
   22 1.006865807      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=293/9473, ttl=254 (request in 21)
   24 1.107435685      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=294/9729, ttl=254 (request in 23)
```

Now it is just matter of local preference configuration again.
R3:

```
configure

route-policy RRv6
  set local-preference 101
end-policy
!
commit
end
```

at this moment all VPN traffic in the network should be using SRv6. Only traffic running with MPLS encapsulation should be BGP messages towards r5 - v4 Route Reflector
At the moment you commit command above tshak window stops showing ICMP packets and you should see just BGP keepalive every 30 seconds:

```
2231 106.450131909      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=1968/45063, ttl=254 (request in 2230)
 2233 106.543951347      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=1969/45319, ttl=254 (request in 2232)
 2236 106.645118918      7.7.7.7 → 10.3.9.9     ICMP 122 Echo (ping) reply
id=0xcea7, seq=1970/45575, ttl=254 (request in 2235)
 2928 139.456916830      1.1.1.1 → 5.5.5.5      TCP 58 56417 → 179 [ACK] Seq=39
Ack=58 Win=32521 Len=0
 2984 142.229815035      1.1.1.1 → 5.5.5.5      BGP 77 KEEPALIVE Message
 4194 199.457213798      1.1.1.1 → 5.5.5.5      TCP 58 56417 → 179 [ACK] Seq=58
Ack=77 Win=32502 Len=0
 4253 202.229880919      1.1.1.1 → 5.5.5.5      BGP 77 KEEPALIVE Message
```

6 IPv4 Removal

Now we can remove entire IPv4 and MPLS configuration.

6.1 Disable v4 BGP

First step is to disconnect v4 Route Reflector, so we shut down interface towards v4 RR on r4:

```
configure

interface GigabitEthernet0/0/0/0
 shutdown
!
```

```
commit
end
```

Now we will remove v4 related BGP configuration from all PE routers – r1, r2,r3:

```
configure

router bgp 1
 no neighbor 5.5.5.5
 vrf 1
  address-family ipv4 unicast
  no mpls alloc enable
commit
end
```

This will result in removal of all MPLS entries in BGP table.
Example for R3:

```
RP/0/RP0/CPU0:xr3#sh bgp vpnv4 unicast
Mon Jan  2 20:48:06.482 UTC
BGP router identifier 3.3.3.3, local AS number 1
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0
BGP main routing table version 38
BGP NSR Initial initsync version 7 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf 1)
*>i7.7.7.7/32         2001::1                  0    101      0 7 i
*>i8.8.8.8/32         2001::2                  0    101      0 8 i
*> 9.9.9.9/32         10.3.9.9                 0             0 9 i
*>i10.1.7.0/24        2001::1                  0    101      0 ?
*>i10.2.8.0/24        2001::2                  0    101      0 ?
*> 10.3.9.0/24        0.0.0.0                  0         32768 ?

Processed 6 prefixes, 6 paths
RP/0/RP0/CPU0:xr3#
```

6.2 Disable LDP and OSPF

Now we can remove entire IPv4 underlay routing on r1, r2, r3, r4:

```
configure

no router ospf 1
no mpls ldp

commit
end
```

6.3 Remove IPv4 addressing

And we do not need v4 addressing in underlay r1,r2,r3:

```
configure

interface Loopback0
 no ipv4 address
!
interface GigabitEthernet0/0/0/0
 no ipv4 address

commit
end
```

r4:

```
configure

interface Loopback0
 no ipv4 address
!
interface GigabitEthernet0/0/0/0
 no ipv4 address

interface GigabitEthernet0/0/0/1
 no ipv4 address

interface GigabitEthernet0/0/0/2
 no ipv4 address

interface GigabitEthernet0/0/0/3
 no ipv4 address

commit
end
```

Still 0 packet loss in any CE ping window so you stop ping and verify it:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!^C
Success rate is 100 percent (15001/15001), round-trip min/avg/max = 2/4/260 ms
RP/0/RP0/CPU0:xr9#
```

7 Conclusion

You made full migration from MPLS .to SRv6. Original MPLS network was based on OSPFv2 and LDP. But concept stays identical for any MPLS based network. Migration can be very granular using BGP path selection algorithm in our case we used local preference, but in real network any attribute influencing best path selection can be used. For this lab we used VPNv4 address family, but same principles apply to VPNv6, EVPN or global address families.