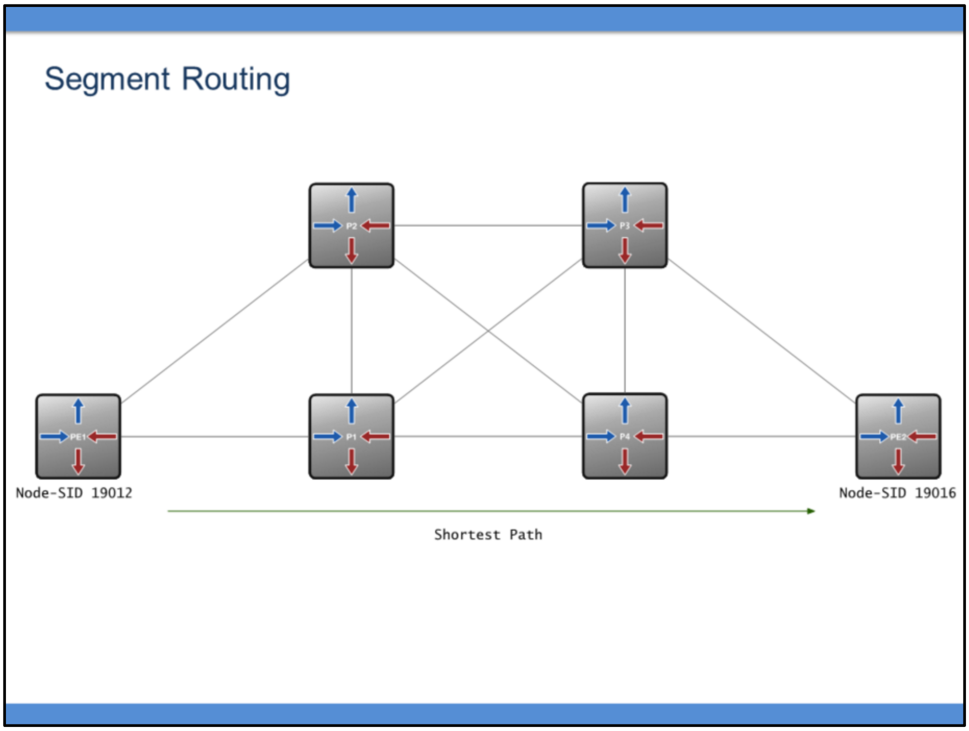
Scamall Networks

MPLS Ain't Dead

Training. Operations. Architecture.



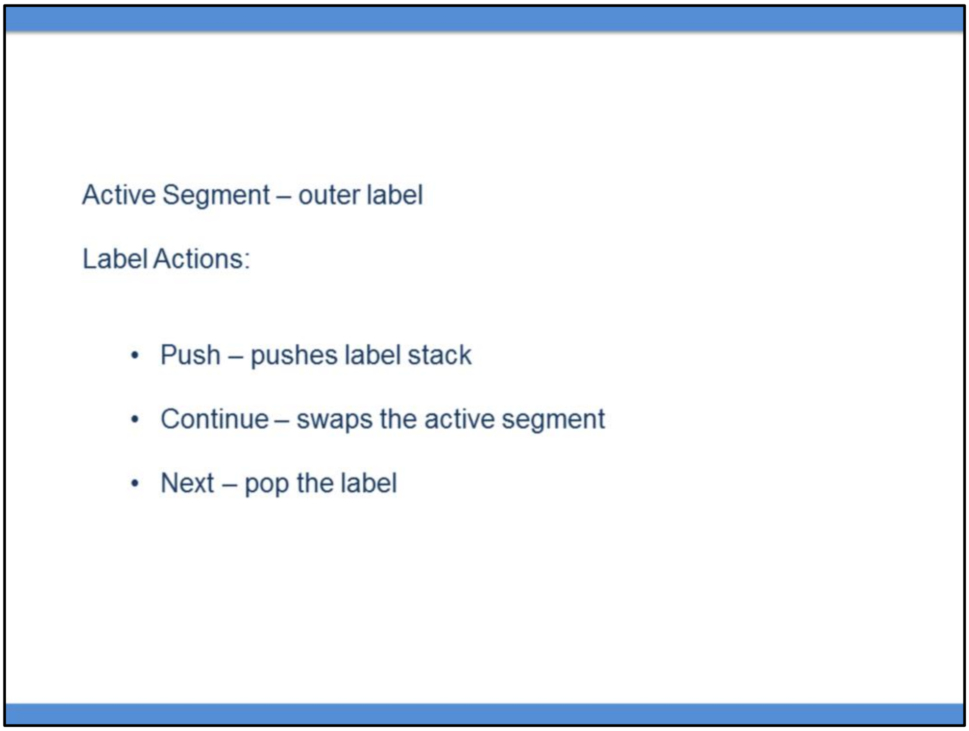
Segment routing allows for dynamic label allocation in the same manner as LDP does while also providing static path assignment in the vain of RSVP-TE, without the associated state in the network.

```
A:pe1# show router mpls-labels label-range

===============================================================================
Label Ranges
===============================================================================
Label Type      Start Label End Label   Aging       Available   Total
-------------------------------------------------------------------------------
Static          32          18431       -           18400       18400
Dynamic         18432       131071      0           112581      112640
    Seg-Route   19000       19050       -           0           51

*A:pe1# show router isis database detail pe1.00-00 | match expression
"GB|Prefi"
       SRGB Base:19000, Range:51
    Prefix   : 192.0.2.12
      Prefix-SID Index:12, Algo:0, Flags:NnP

*A:pe1# show router isis database detail pe2.00-00 | match expression
"GB|Prefi"
       SRGB Base:19000, Range:51
    Prefix   : 192.0.2.16
      Prefix-SID Index:16, Algo:0, Flags:NnP
```

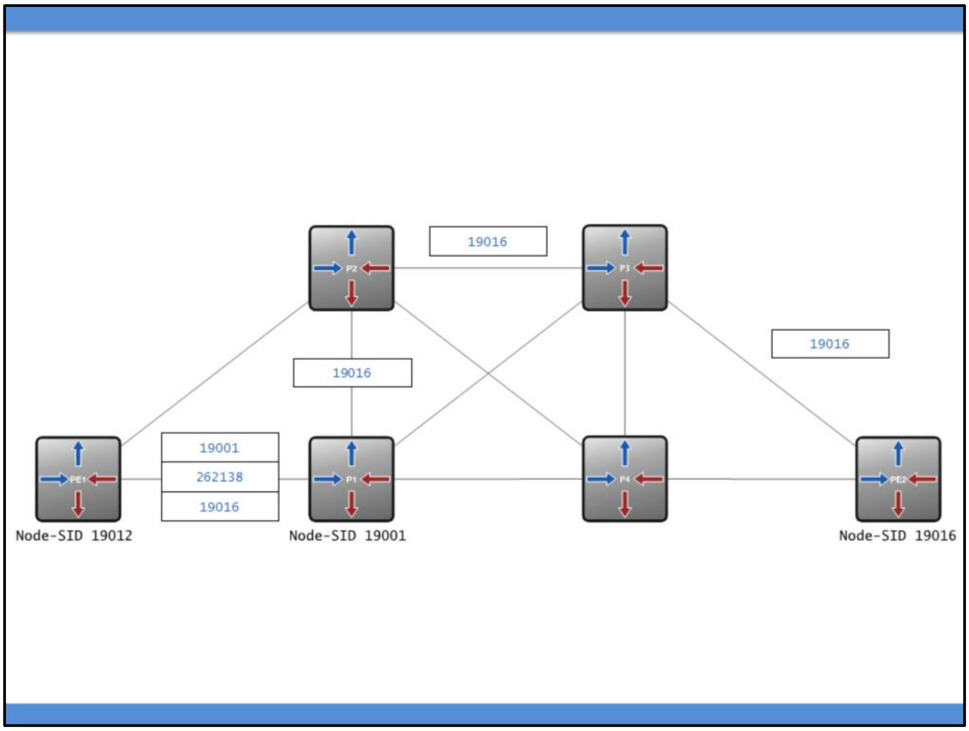
One of the base elements of Segment Routing is the concept of the SRGB, or global block. As different vendors have allocated various label ranges to protocols there must be a way to ensure every SR node in the network understands the base allocation. Here PE1 is using SRGB 19000 to 19050 (range of 51) and its local Prefix SID (Node SID) is index 12. PE2 uses the same SRGB and advertises this in IS-IS. Note its prefix-SID index is 16 so PE1 deduces its Node SID label is 19016.

```
*A:pe1# show router isis prefix-sids sid 16

Prefix                            SID        Lvl/Typ    SRMS   AdvRtr
                                                         MT     Flags
-----------------------------------------------------------------------------
--
192.0.2.16/32                     16         2/Int.       N    pe2
                                                             0       NnP
<snip>

*A:pe1# show router bgp routes vpn-ipv4 198.51.100.16/32 hunt | match "VPN L"
Route Dist.    : 64497:16              VPN Label      : 262142


MultiProtocol Label Switching Header, Label: 19016, Exp: 7, S: 0, TTL: 254
  0000 0100 1010 0100 1000 .... .... .... = MPLS Label: 19016
  .... .... .... .... .... 111. .... .... = MPLS Experimental Bits: 7
  .... .... .... .... .... ...0 .... .... = MPLS Bottom Of Label Stack: 0
  .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
MultiProtocol Label Switching Header, Label: 262142, Exp: 7, S: 1, TTL: 64
  0011 1111 1111 1111 1110 .... .... .... = MPLS Label: 262142
  .... .... .... .... .... 111. .... .... = MPLS Experimental Bits: 7
  .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
  .... .... .... .... .... .... 0100 0000 = MPLS TTL: 64
Internet Protocol Version 4, Src: 198.51.100.12 (198.51.100.12), Dst: 198.51.100.16
Internet Control Message Protocol
```

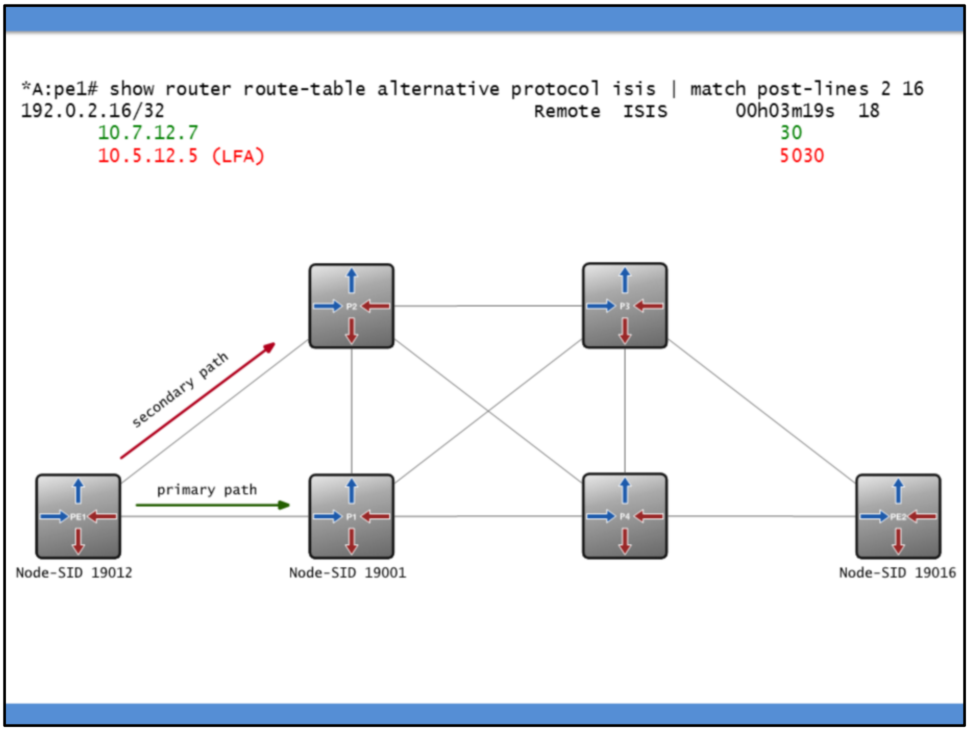
This is a packet capture taken between P1 and P4 on the shortest path to PE2 for an echo reply, originating on PE1.



Push is the same function as with generic MPLS. Continue is similar to the swap but if the SRGB is in use the value will be the same. Next pops the label and follows the next instruction.

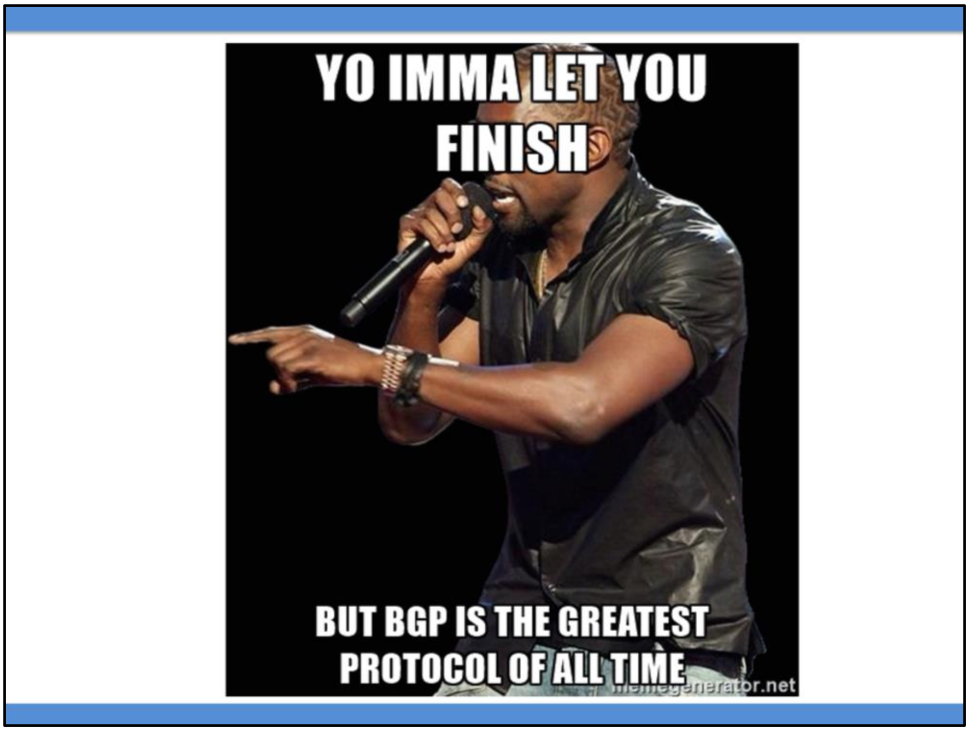


If we wish to control the path more explicitly a la RSVP-TE we can push a label stack at the iLER. Here we are using a controller to push a three label stack. The outer label of 19001 identifies P1 in the network. Once P1 receives this value it knows to pop it and forwards the packet towards P2 based on the 262138 Adj-SID for that link. P2 continues with a continue of label 19016, followed by P3 all the way to the eLER PE2.

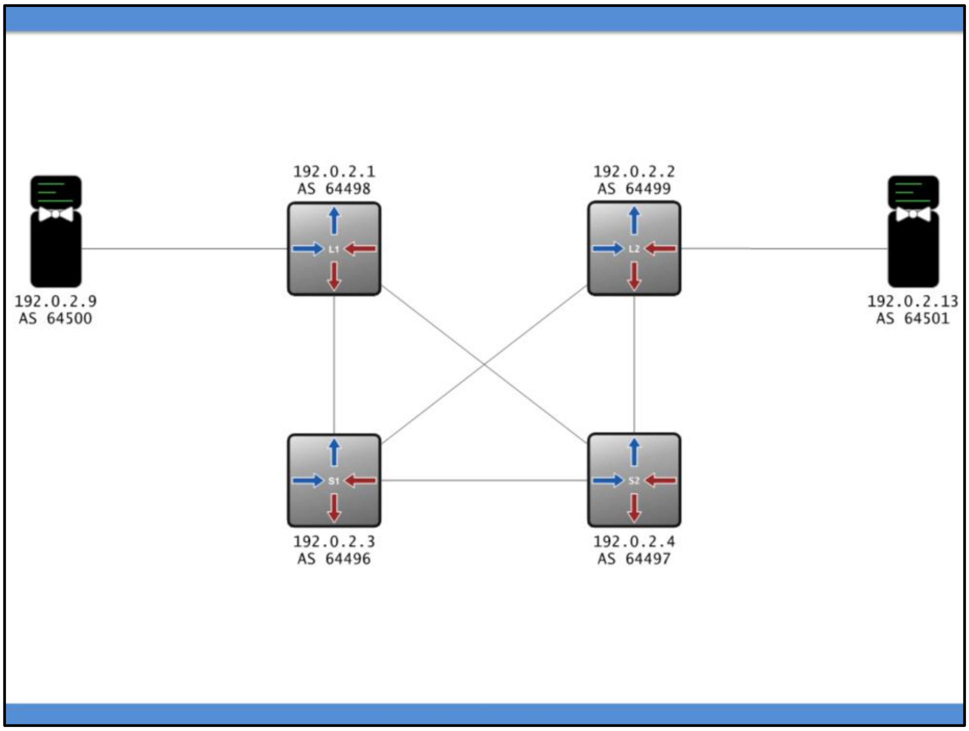


SR gives us the ability to have TI-LFA, or Topology Independent Loop Free Alternates. There is no requirement to use targeted LDP sessions as we do with Remote LFA. Loop free paths can be installed ready for service upon a link or nodal failure. Should the link between PE1 and P1 fail, LFA will allows us to reroute around the problem without the risk of a microloop

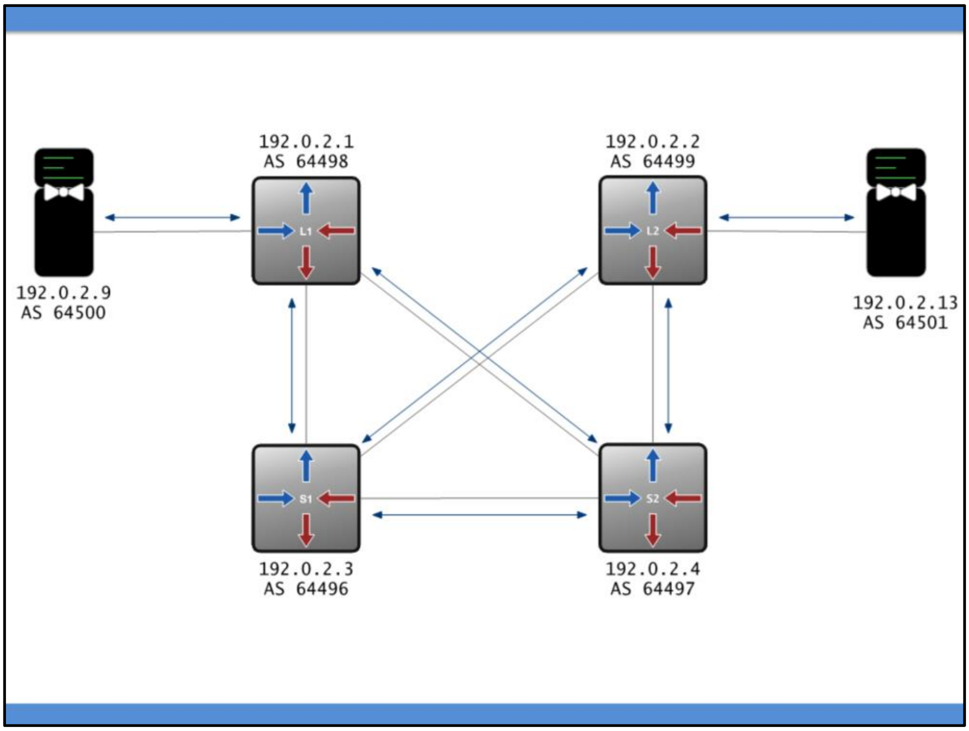
deb router ip route-table 192.0.2.16/32
show router route-t alternative protocol isis
show router bfd session
show router isis lfa



Did you know BGP can signal transport labels? Let’s take a look!



Consider an all BGP data center. We have two spine routers, two leaves and two hosts. Each of the devices are running eBGP and only eBGP. By using the advertise label functionality for IPv4 unicast, we can build a complete MPLS transport network with BGP. Why eBGP? I dont have to worry about split horizon, messing with iBGP route reflectors and I can control policy with much greater ease.



We run eBGP between link local addresses and export our system address in to BGP.

```
 group "lu_peer"
                type external
                export "TO_BGP"
                peer-as 64497
                neighbor 10.3.4.4
                    advertise-label ipv4
                exit
            exit
group "lu_downstream"
                type external
                export "TO_BGP"
                neighbor 10.1.3.1
                    peer-as 64498
                    advertise-label ipv4
                exit
                neighbor 10.2.3.2
                    peer-as 64499
                    advertise-label ipv4
                exit
            exit
```

Here is the configuration on R3 for example. This ensured we propagate labels to our directly connected routers. We can see our neighbours label status using show router bgp neigh $address | match Label

```
A:all_the_routers# show router policy "TO_BGP"
    entry 10
        from
            prefix-list "LOCAL"
        exit
        to
            protocol bgp
        exit
        action accept
        exit
    exit
A:all_the_routers# show router policy prefix-list "LOCAL"
prefix 192.0.2.x/32 exact
```

```
A:host1# show router bgp routes
===============================================================================
==
 BGP Router ID:192.0.2.9        AS:64500        Local AS:64500

<snip>

Flag  Network                                            LocalPref   MED
      Nexthop (Router)                                   Path-Id     Label
      As-Path
-------------------------------------------------------------------------------
--
u*>i  192.0.2.2/32                                       None        None
      10.1.9.1                                           None        131064
      64498 64496 64499
u*>i  192.0.2.3/32                                       None        None
      10.1.9.1                                           None        131066
      64498 64496
u*>i  192.0.2.4/32                                       None        None
      10.1.9.1                                           None        131069
      64498 64497
i     192.0.2.9/32                                       None        None
      10.1.9.1                                           None        131071
      64498 64500
u*>i  192.0.2.13/32                                      None        None
      10.1.9.1                                           None        131063
      64498 64496 64499 64501
```

We have label allocations for the two most important routers from host1s perspective: its future all-but-vanilla router reflector and the path to 192.0.2.13/32

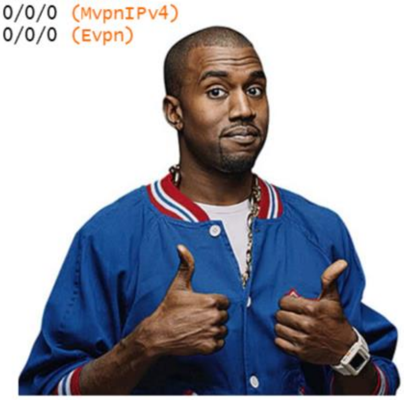
```
group "rr"
            family vpn-ipv4 vpn-ipv6 l2-vpn mvpn-ipv4 evpn
            type external
            export "TO_BGP"
            peer-as 64497
            neighbor 192.0.2.4
            exit
        exit
group "vpn_clients"
            family vpn-ipv4 vpn-ipv6 l2-vpn mvpn-ipv4 evpn
            type external
            multihop 10
            cluster 0.0.0.3
            neighbor 192.0.2.9
                peer-as 64500
            exit
            neighbor 192.0.2.13
                peer-as 64501
            exit
        exit
```

Once we have an LSP from each device to every other we can create our services BGP peerings. S1 and S2 are route reflectors and they are running inter AS VPNs.

```
A:s1>config>router>bgp# show router bgp routes 192.0.2.9
<snip>
u*>i  192.0.2.9/32                                              None        None
      10.1.3.1                                                 None        131071
      64498 64500                                                 #transport label

A:host2# show router bgp routes vpn-ipv4 198.51.100.9/32
<snip>
u*>i  64500:20:198.51.100.9/32                                  None        None
      192.0.2.9                                                None        262143
      64496 64500                                                   #service label


MultiProtocol Label Switching Header, Label: 131071, Exp: 7, S: 0, TTL: 253
  0001 1111 1111 1111 1111 .... .... .... = MPLS Label: 131071
  .... .... .... .... .... 111. .... .... = MPLS Experimental Bits: 7
  .... .... .... .... .... ...0 .... .... = MPLS Bottom Of Label Stack: 0
  .... .... .... .... .... .... 1111 1101 = MPLS TTL: 253
MultiProtocol Label Switching Header, Label: 262143, Exp: 7, S: 1, TTL: 64
  0011 1111 1111 1111 1111 .... .... .... = MPLS Label: 262143
  .... .... .... .... .... 111. .... .... = MPLS Experimental Bits: 7
  .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
  .... .... .... .... .... .... 0100 0000 = MPLS TTL: 64
Internet Protocol Version 4, Src: 198.51.100.13 (198.51.100.13), Dst: 198.51.100.9
Internet Control Message Protocol
```

Here we see the transport label, as signalled by eBGP IPv4 and the service label signalled by MPBGP VPNv4

```
192.0.2.3
                 64496      472      0 00h42m15s 2/1/2 (VpnIPv4)
                             99      0           2/1/2 (VpnIPv6)
                                                 2/1/2 (L2VPN)
                                                 0/0/0 (MvpnIPv4)
                                                 0/0/0 (Evpn)
192.0.2.4
                 64497      473      0 03h51m56s 2/0/2 (VpnIPv4)
                            478      0           2/0/2 (VpnIPv6)
                                                 2/0/2 (L2VPN)
                                                 0/0/0 (MvpnIPv4)
                                                 0/0/0 (Evpn)
```



Even Kanye didn't see that coming, so many address families.

https://snu.training

keith@scamall-networks.com

Training. Operations. Architecture.