

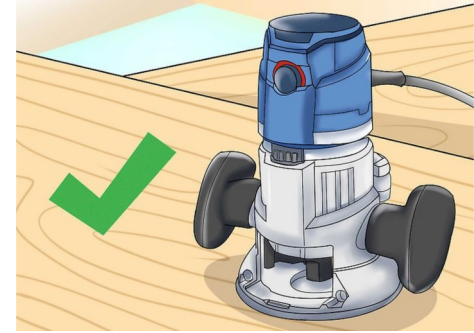


# The Label Switched Path Not Taken

An exploration of MPLS2IP Forward in brownfield Segment Routing Environments

# About Me

- **Steve Crutchley**
- **Network Engineer for Claranet**
- **Lived in London for 11 years**
- **From New Zealand**



[steve@netquirks.co.uk](mailto:steve@netquirks.co.uk)



[netquirks.co.uk](http://netquirks.co.uk)



[/stevecrutchleynz](https://www.linkedin.com/company/netquirks)



[@netquirks](https://twitter.com/netquirks)



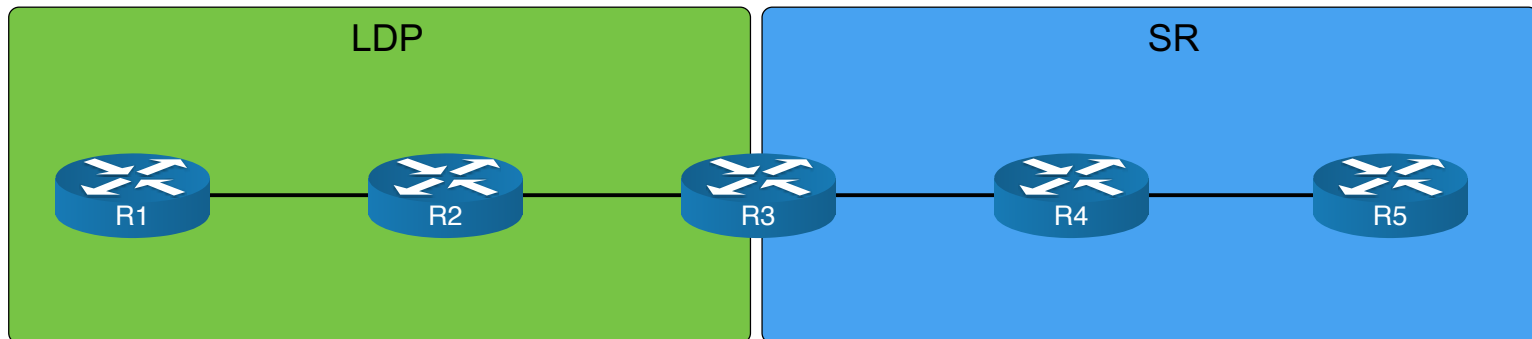
# Overview

- SR/LDP co-existence refresher
- Brownfield topology
- IPv4 forwarding
- IPv6 forwarding - MPLS2IP forwarding behaviour using different vendors
- Possible Solutions
- Broken LSP - who is correct?



# SR/LDP co-existence refresher

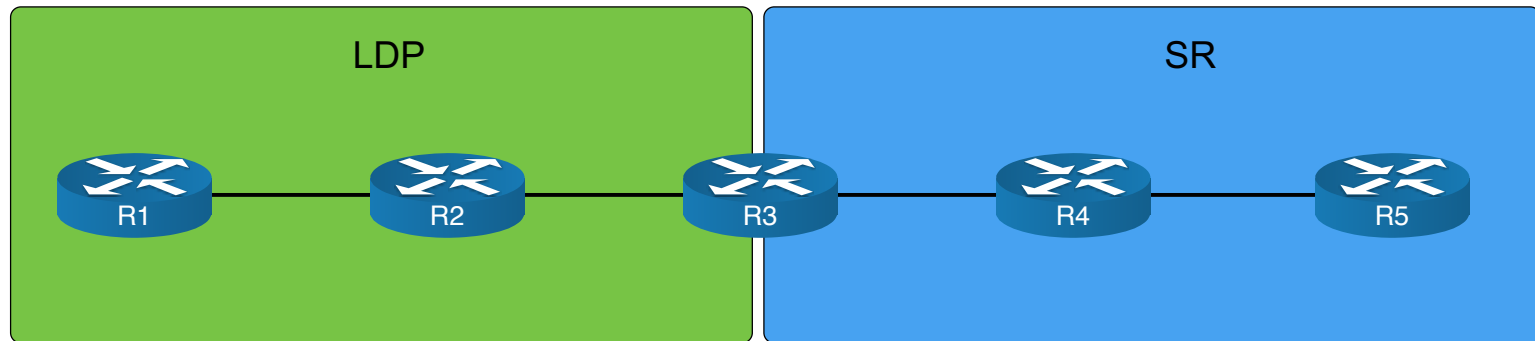
- Not everything will support SR – so LDP will still be with us.
- What happens when we cross from SR to LDP and vice versa?
- The basic idea is as follows:
  - Border nodes run LDP and SR
  - If the border node doesn't have a label for on protocol it will inherit one from the other



***Interworking is achieved by replacing an unknown outbound label from one protocol by a valid outgoing label from another protocol.***

# SR/LDP co-existence refresher

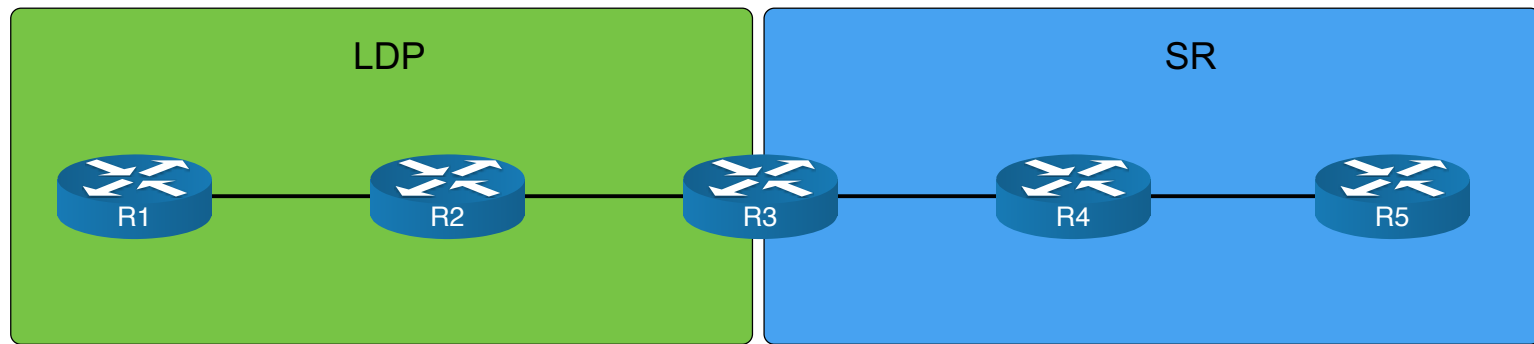
## LDP to SR



- R3 is the border node
- It's runs LDP with R2 and is configured with SR.
- R5 will advertise its node SID which R3 will install in it's LFIB
- All of the LDP nodes will allocate local labels for 1.1.1.5 when then see it in the IGP and advertise it their neighbors

# SR/LDP co-existence refresher

## LDP to SR



	In Label	Out Label
SR	17005	17005
		↓ inherited
LDP	23053	17005

# SR/LDP co-existence refresher

## LDP to SR

- The `show mpls ldp forwarding` command will show Unlabelled as the output.

```
RP/0/0/CPU0:xrvr-3#show mpls ldp forwarding 1.1.1.5/32
```

Codes:  
- = GR label recovering, (!) = LFA FRR pure backup path  
{ } = Label stack with multi-line output for a routing path  
G = GR, S = Stale, R = Remote LFA FRR backup

Prefix	Label In	Label(s) Out	Outgoing Interface	Next Hop	Flags G S R
1.1.1.5/32	23053	Unlabelled	Gi0/0/0/0	99.3.4.4	

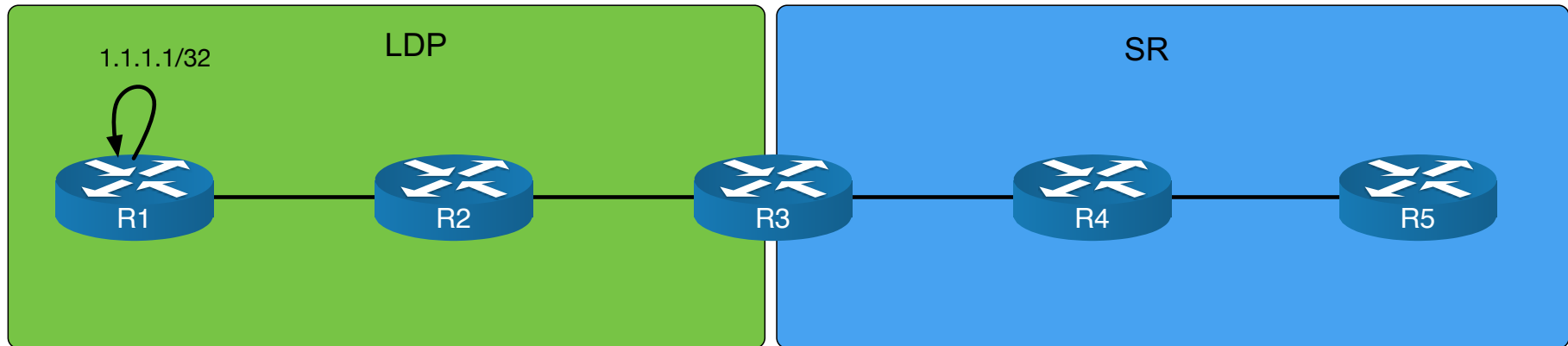
- Better to use `show mpls forwarding`

```
RP/0/0/CPU0:xrvr-3#show mpls forwarding prefix 1.1.1.5/32
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
23053	17005	1.1.1.5/32	Gi0/0/0/0	99.3.4.4	0

# SR/LDP co-existence refresher

## SR to LDP



- R3 is the border node
- It's runs LDP with R2 and is configured with SR.
- All of the LDP nodes will allocate local labels for 1.1.1.1 when then see it in the IGP and advertise it their neighbors
- But how do the SR nodes know what label to use?
  - MAPPING SERVERS!



# SR/LDP co-existence refresher

## SR to LDP

- Mapping servers advertise IP to Node-SID mapping in their IS-IS TLV updates.
- They basically advertise the Node-SIDs on half of those that can't!



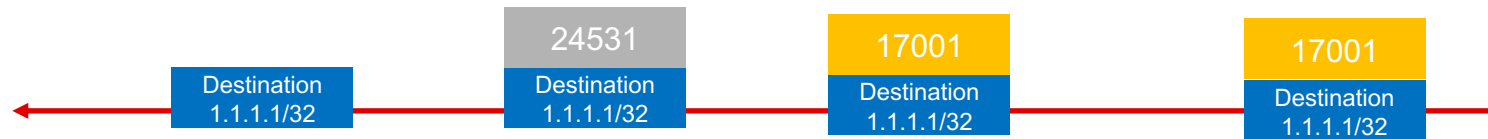
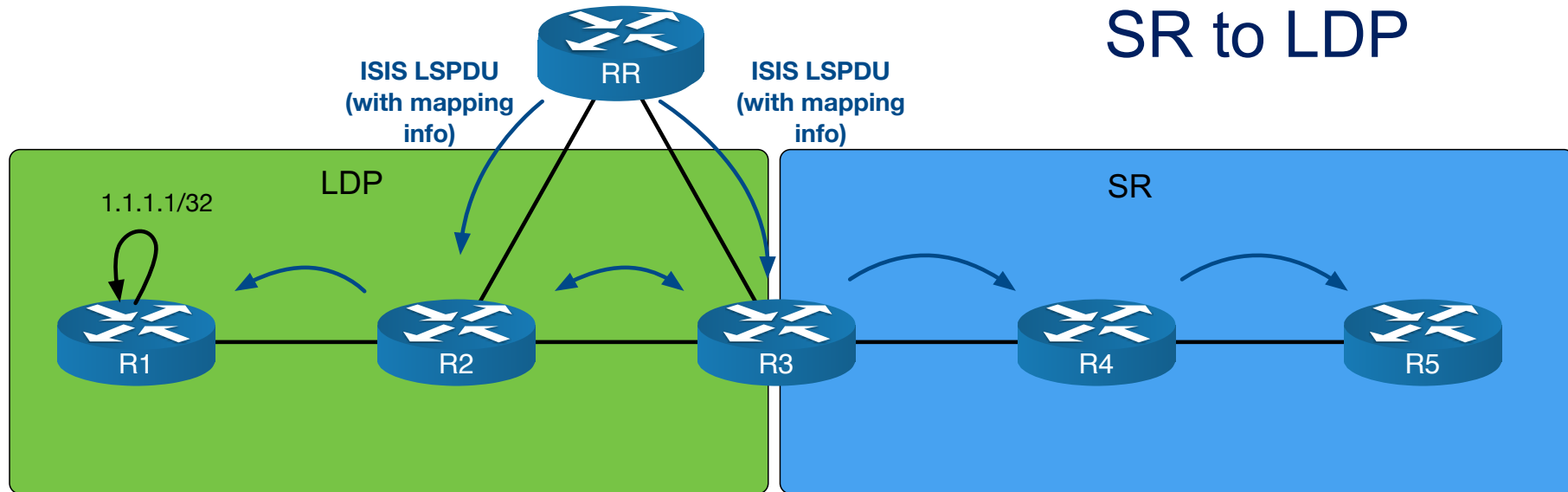
```
segment-routing
global-block 17000 18999
mapping-server
  prefix-sid-map
    address-family ipv4
      1.1.1.1/32 1 range 1
  !
  !
  !
  !
router isis LAB
  address-family ipv4 unicast
  segment-routing prefix-sid-map advertise-local
```

**SRBG**

**SID index**

**Command to act as  
mapping server**

# SR/LDP co-existence refresher



	Out Label	In Label
LDP	24531	23011
	↓ inherited	
SR	24531	17001

SR recognises that downstream is LDP only.

Non-SR nodes just ignore the TLV info

# SR/LDP co-existence refresher

## SR to LDP

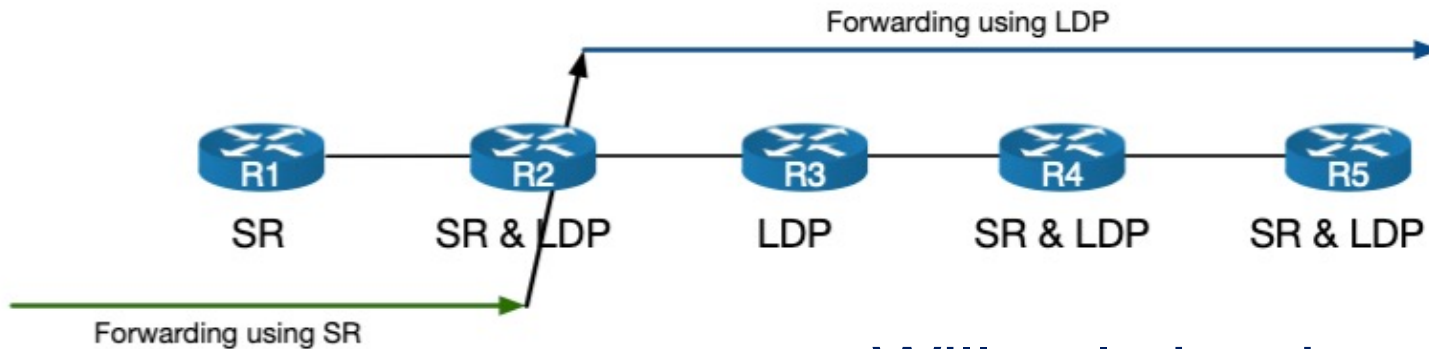
- The `show mpls forwarding labels <SID>` command

```
RP/0/0/CPU0:xrvr-3#show mpls forwarding labels 16001
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
17001	24531	SR Pfx (idx 1)	Gi0/0/0/1	99.2.3.2	0

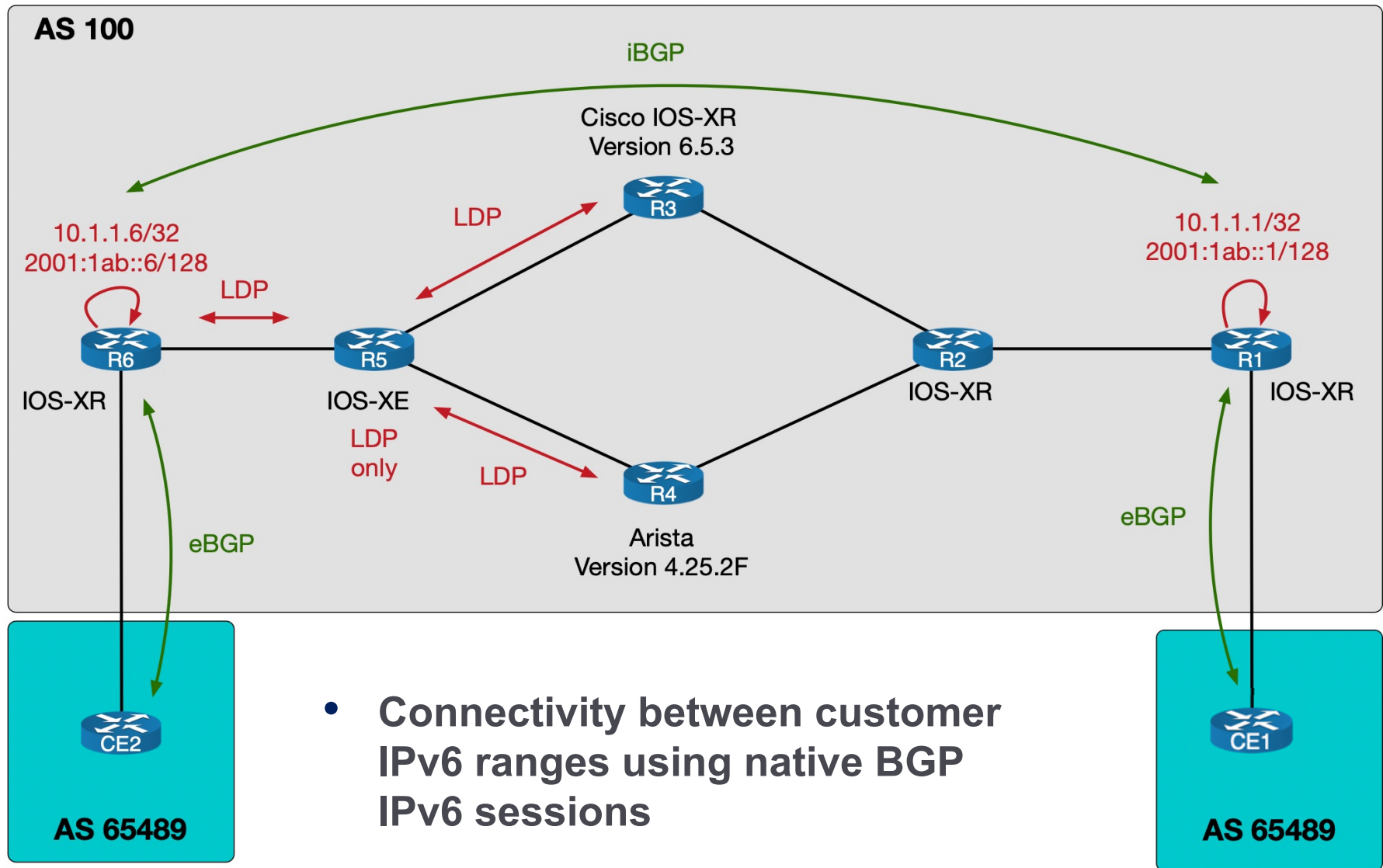
# SR/LDP co-existence refresher

Moving between

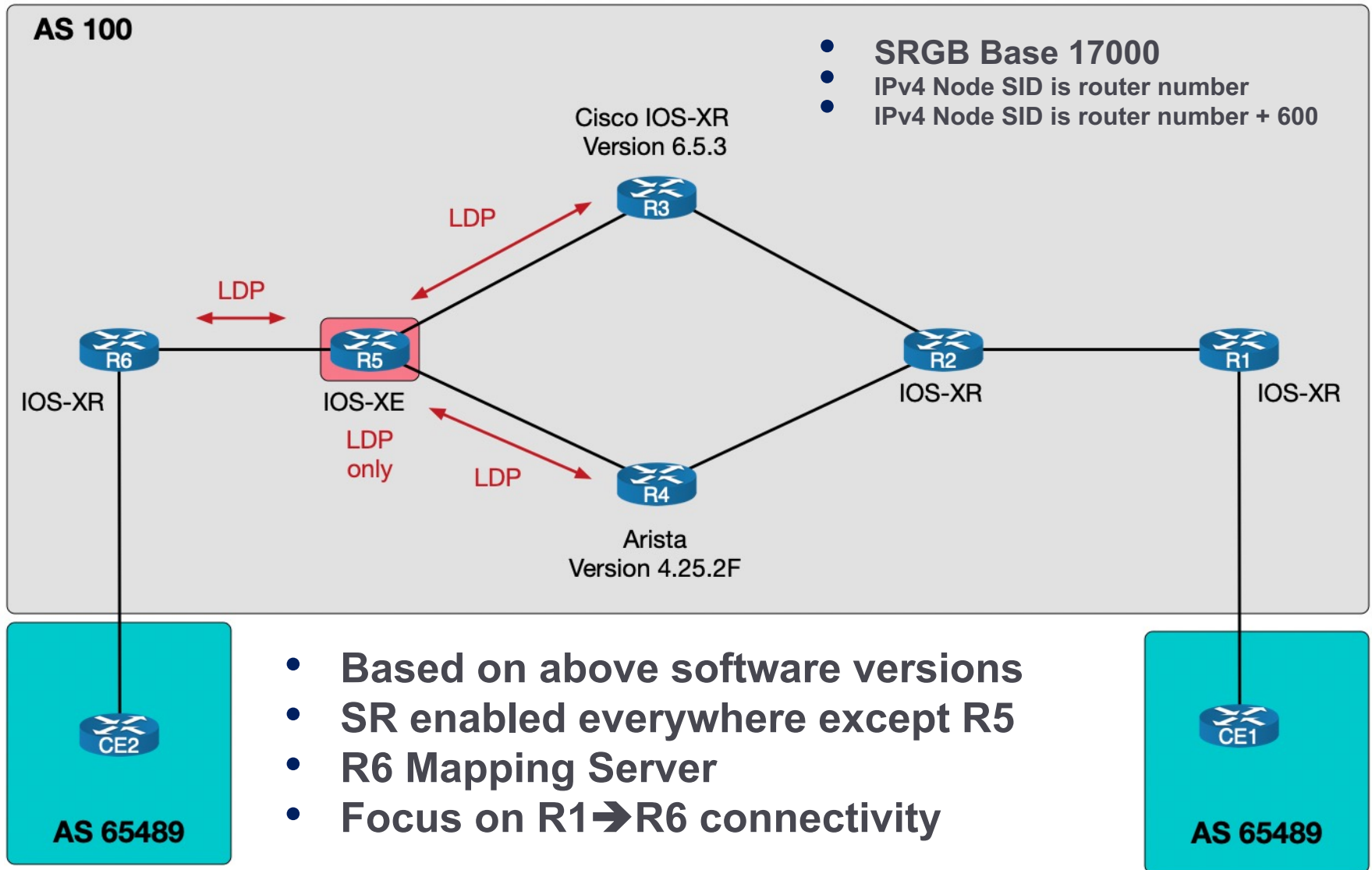


Will only be done if it  
needs to

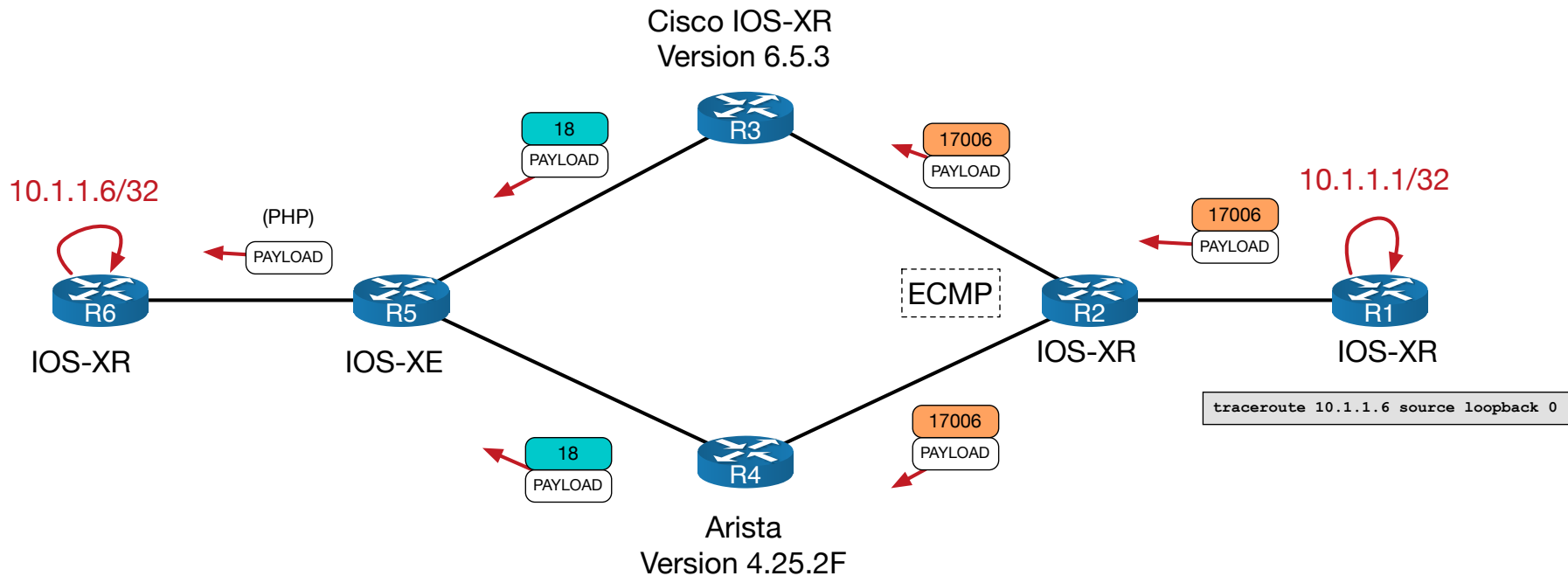
# Brownfield Topology



# Brownfield Topology

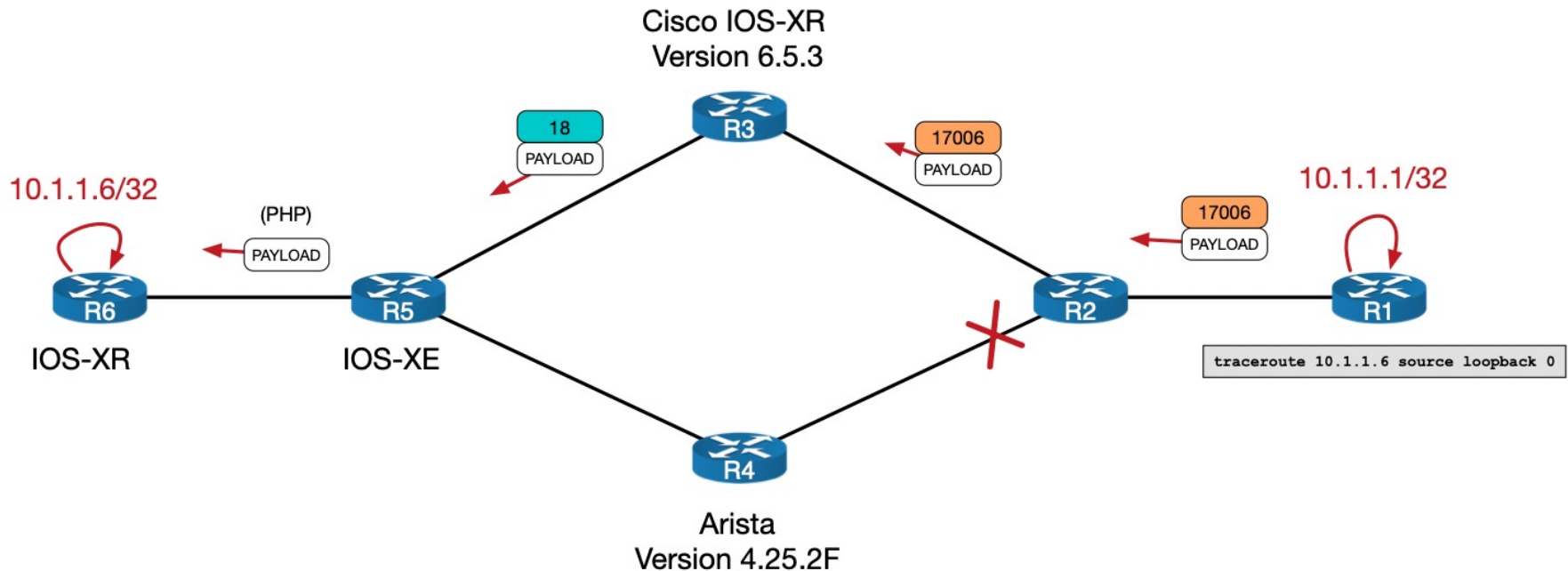


# IPv4 forwarding



- Connectivity works between R1 and R6 over IPv6 regardless of ECMP path.
- Label changes as it reaches the LDP only node.

# IPv4 forwarding - Cisco



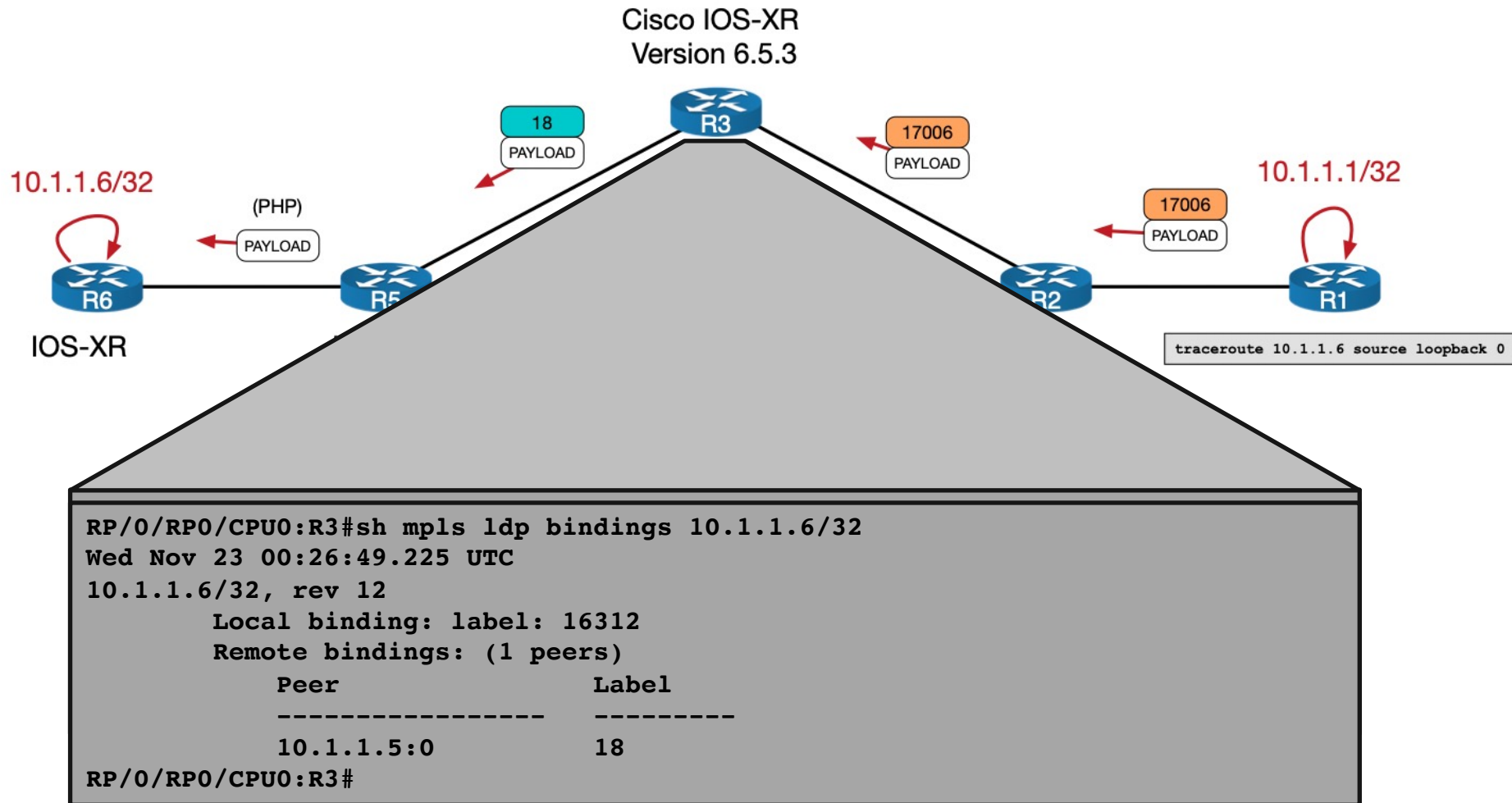
```
RP/0/RP0/CPU0:R1#traceroute 10.1.1.6 source lo0
Wed Nov 23 00:13:59.116 UTC
```

```
Type escape sequence to abort.
Tracing the route to 10.1.1.6
```

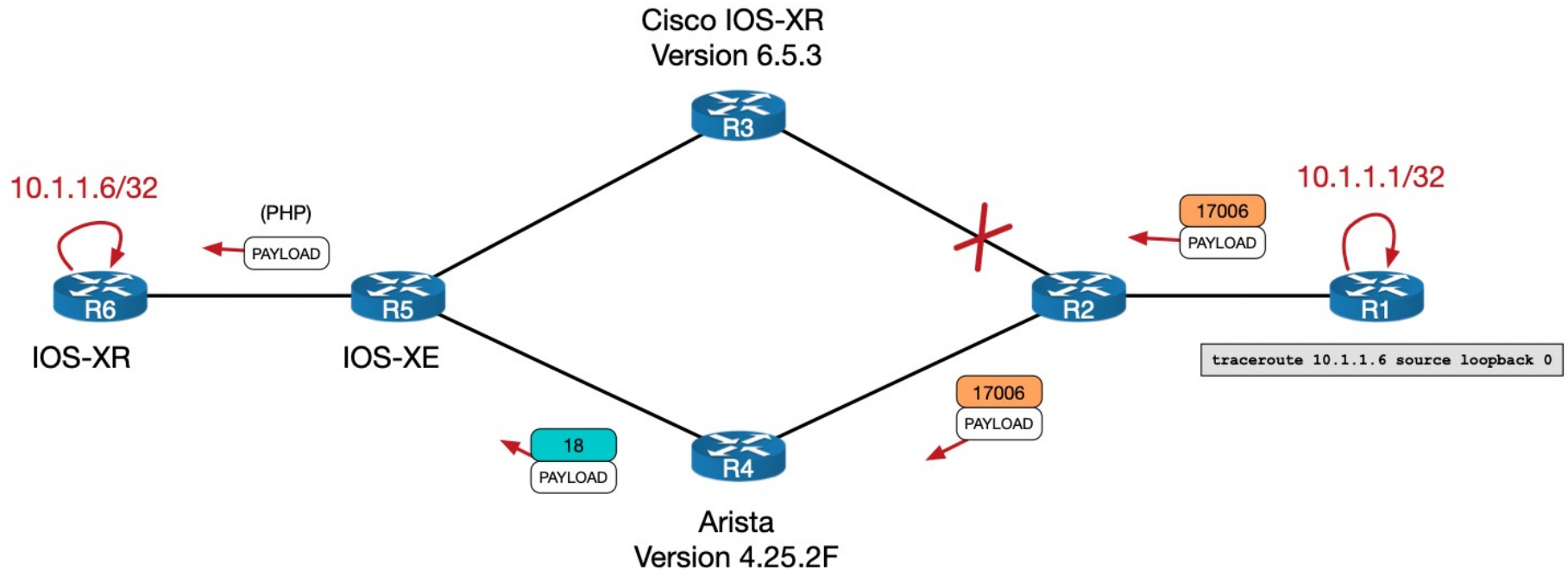
```
1 10.1.2.2 [MPLS: Label 17006 Exp 0] 152 msec 142 msec 135 msec
2 10.2.3.3 [MPLS: Label 17006 Exp 0] 142 msec 148 msec 146 msec
3 10.3.5.5 [MPLS: Label 18 Exp 0] 140 msec 137 msec 135 msec
4 10.5.6.6 148 msec * 126 msec
RP/0/RP0/CPU0:R1#
```



# IPv4 forwarding - Cisco



# IPv4 forwarding - Arista



```
RP/0/RP0/CPU0:R1#traceroute 10.1.1.6 source lo0
```

```
Wed Nov 23 00:41:12.857 UTC
```

```
Type escape sequence to abort.
```

```
Tracing the route to 10.1.1.6
```

```
1  10.1.2.2 [MPLS: Label 17006 Exp 0] 57 msec 54 msec 56 msec
```

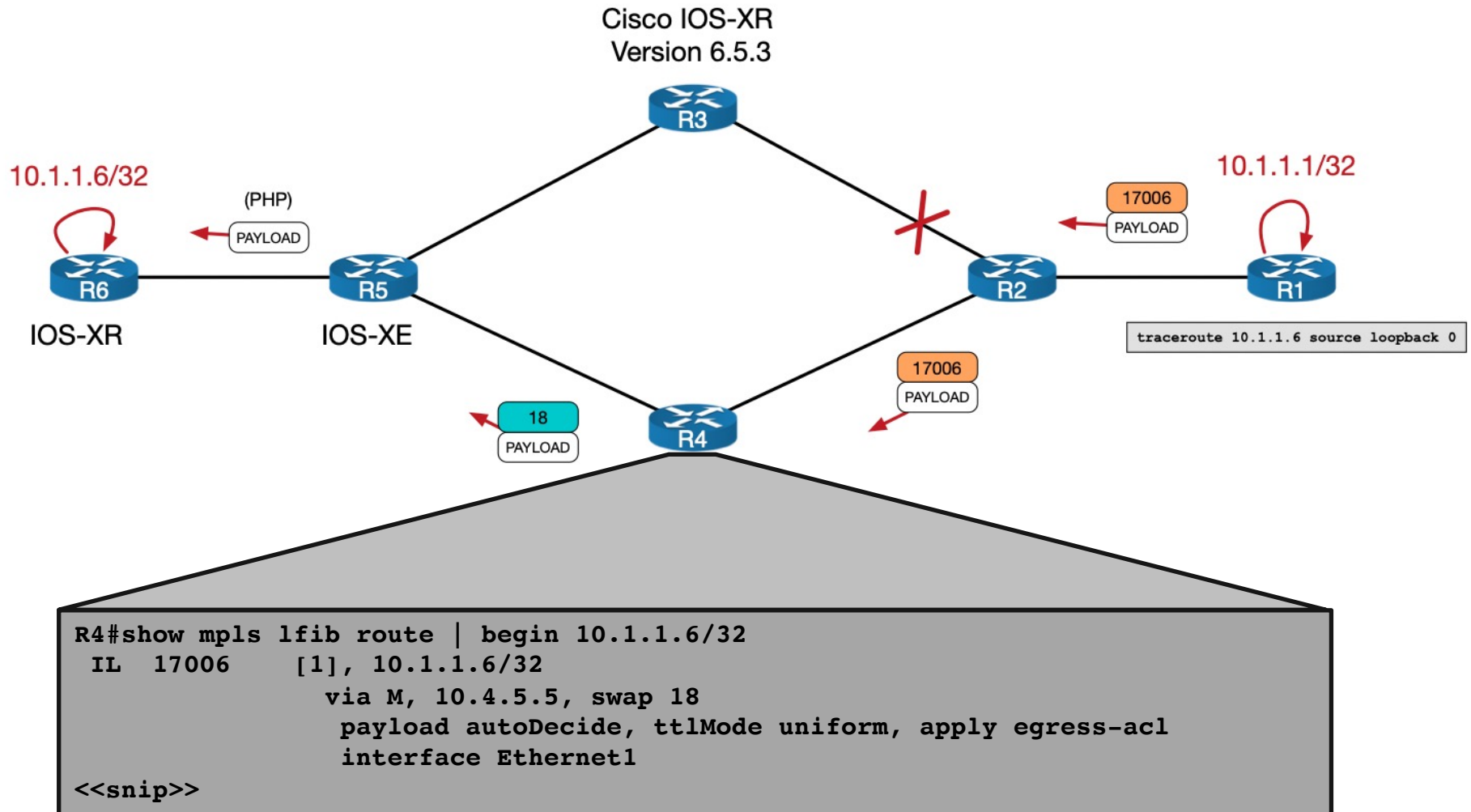
```
2  * * *
```

```
3  10.4.5.5 [MPLS: Label 18 Exp 0] 60 msec 53 msec 51 msec
```

```
4  10.5.6.6 60 msec * 62 msec
```

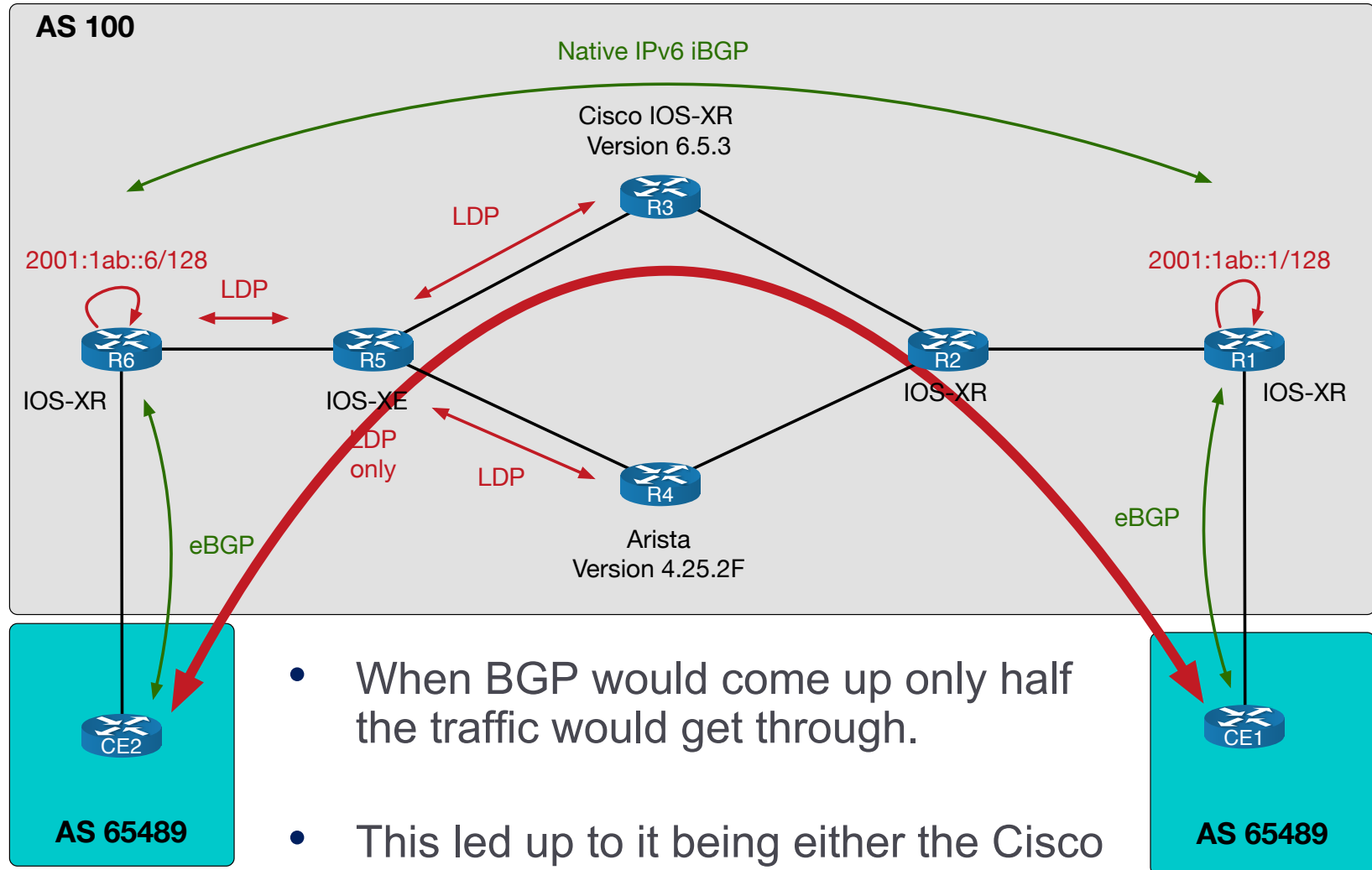
```
RP/0/RP0/CPU0:R1#
```

# IPv4 forwarding - Arista



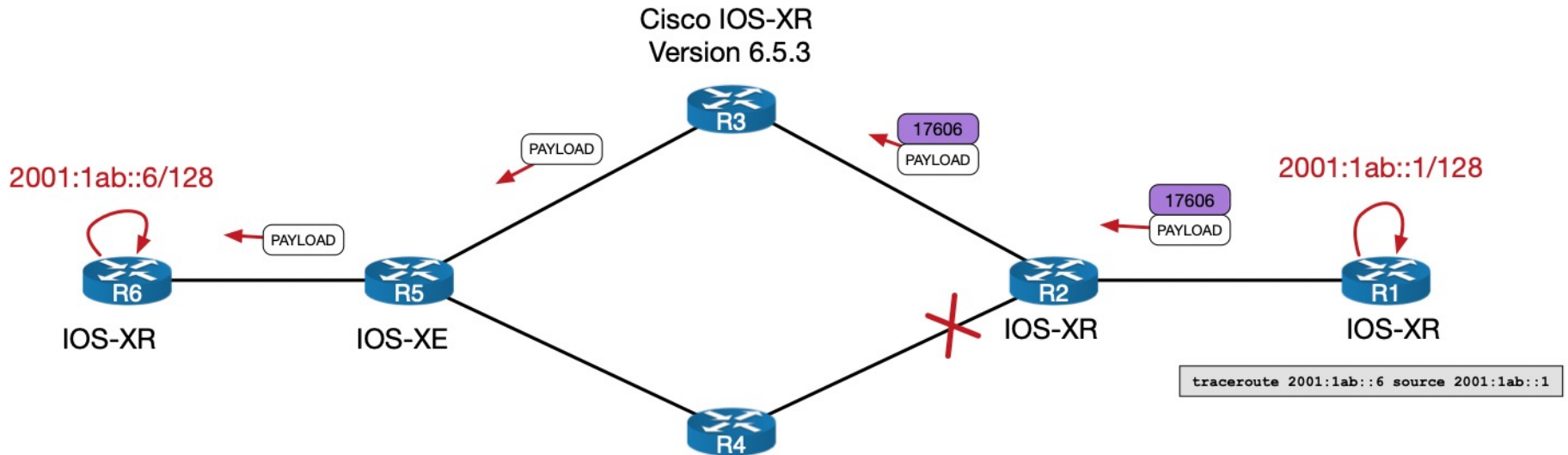
So IPv4 is fine. What about IPv6....?

# IPv6 forwarding



- When BGP would come up only half the traffic would get through.
- This led up to it being either the Cisco or the Arista...

# IPv6 forwarding - Cisco



```
RP/0/RP0/CPU0:R1#traceroute ipv6 2001:1ab::6 source lo0
Wed Nov 23 00:50:51.770 UTC
```

```
Type escape sequence to abort.
Tracing the route to 2001:1ab::6
```

- 1 2001:1ab:1:2::2 [MPLS: Label 17606 Exp 0] 261 msec 48 msec 47 msec
  - 2 2001:1ab:2:3::3 [MPLS: Label 17606 Exp 0] 52 msec 33 msec 46 msec
  - 3 2001:1ab:3:5::5 50 msec 49 msec 48 msec
  - 4 2001:1ab::6 105 msec 86 msec 88 msec
- ```
RP/0/RP0/CPU0:R1#
```

# IPv6 forwarding - Cisco


- Why does Cisco forward natively?



## SEGMENT

If the incoming packet has a single label ... (the label has the End of Stack (EOS) bit set to indicate it is the last label), then the label is removed and the packet is forwarded as an IP packet. If the incoming packet has more than one label ... then the packet is dropped and this would be the erroneous termination of the LSP that we referred to previously.

Segment Routing, Part 1 by by [Clarence Filsfils](#) , [Kris Michielsen](#) , et al.



Clarence Filsfils  
Kris Michielsen  
Ketan Talaulikar

# IPv6 forwarding - Cisco

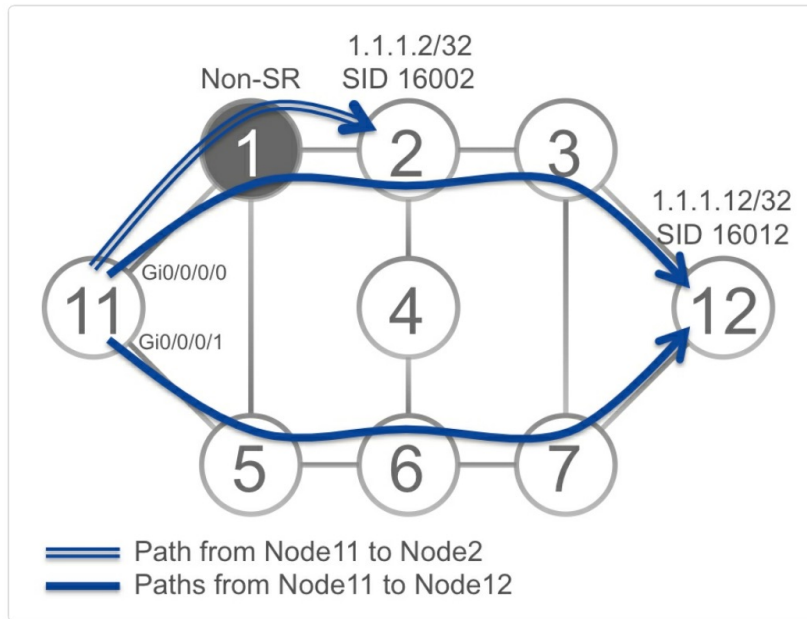
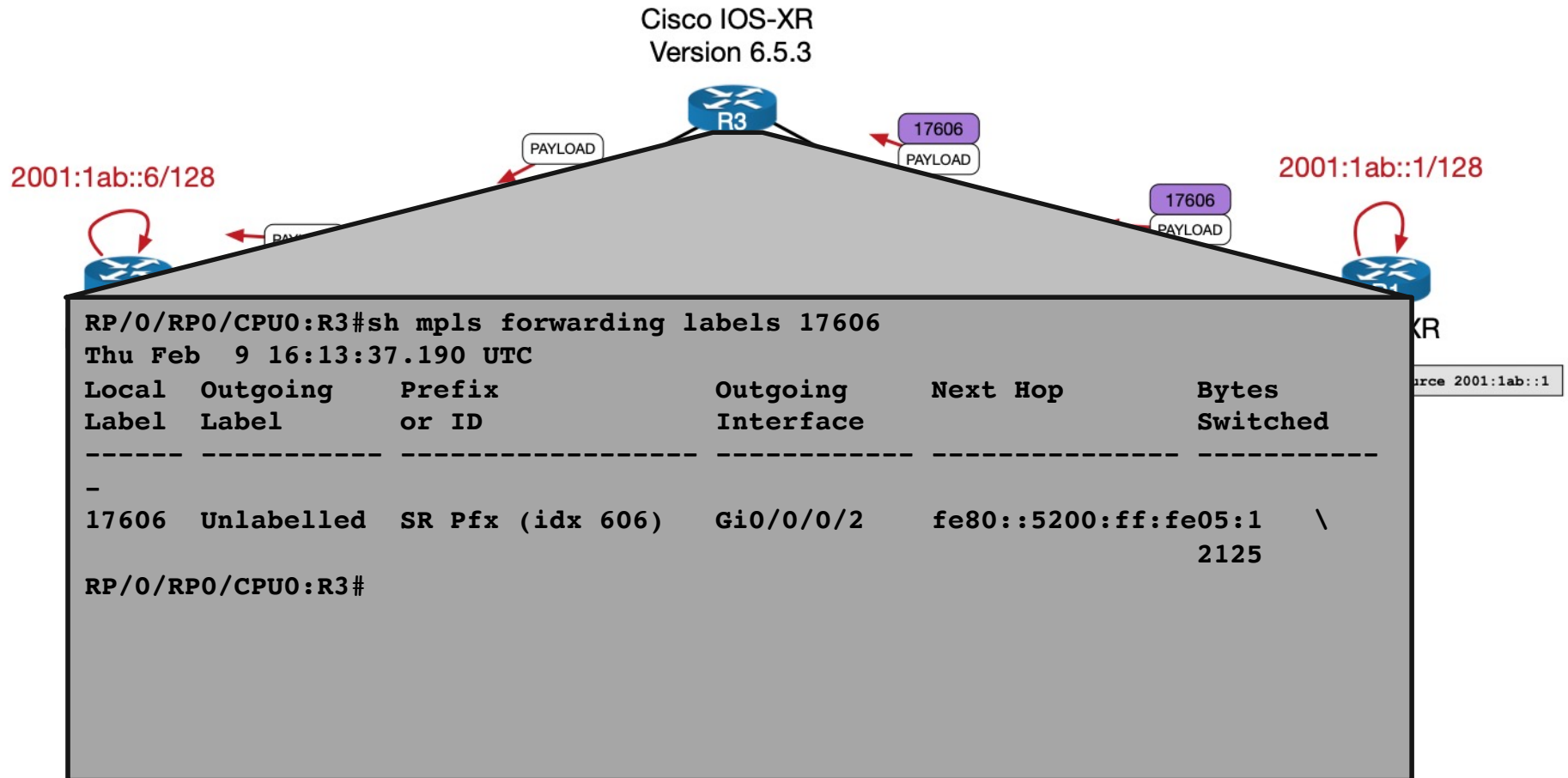


Figure 3-10: Example Unlabelled operation

- RULE AS FOLLOWS:
  - If there is one SR label with EoS bit set: Forward on natively
  - Else, treat as broken LSP and drop
- Specifically for brownfield migrations?

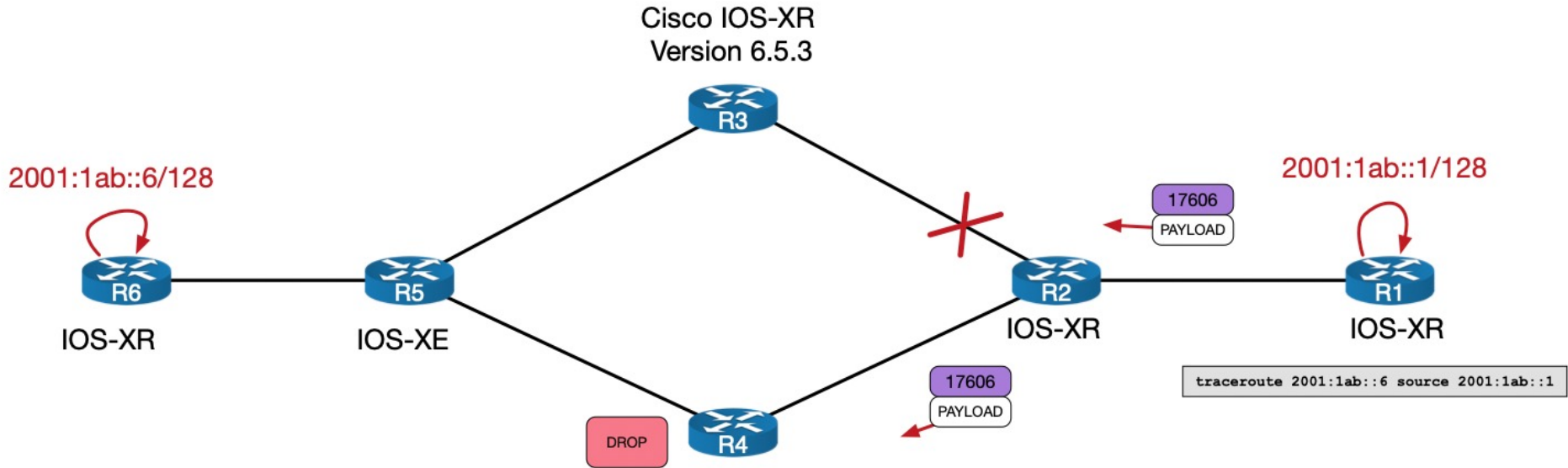
If Node 11 receives an *unlabelled* packet destined for 1.1.1.2/32 then the packet is forwarded as IP packet. If the incoming packet has a *single label* with value 16002 (the label has the End of Stack (EOS) bit set to indicate it is the last label), then the label is removed and the packet is forwarded as an IP packet. If the incoming packet has more than one label and the top label is 16002 (this label has the End of Stack (EOS) bit unset to indicate there is one or more labels underneath), then the packet is dropped and this would be the erroneous termination of the LSP that we referred to previously.

# IPv6 forwarding - Cisco





# IPv6 forwarding - Arista



```
RP/0/RP0/CPU0:R1#traceroute ipv6 2001:1ab::6 source lo0
Sun Jan 29 21:23:48.084 UTC
```

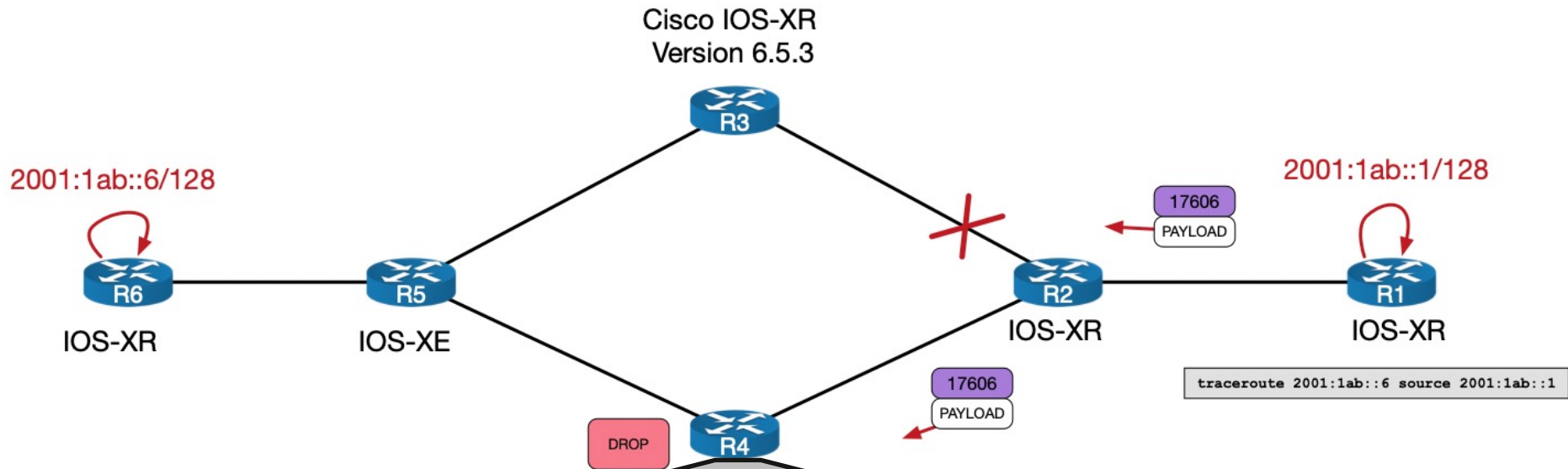
```
Type escape sequence to abort.
Tracing the route to 2001:1ab::6
```

|   |   |   |   |
|---|---|---|---|
| 1 | * | * | * |
| 2 | * | * | * |
| 3 | * | * | * |
| 4 | * | * | * |

 $\hat{C}$ 

**RP/0/RP0/CPU0:R1#**

# IPv6 forwarding - Arista



Pinging direct from R4 (IP2IP works fine!)

```
R4#traceroute ipv6 2001:1ab::6 source 2001:1ab::4
traceroute to 2001:1ab::6 (2001:1ab::6), 30 hops max, 80 byte packets
 1  2001:1ab:4:5::5 (2001:1ab:4:5::5)  5.145 ms  9.544 ms 10.500 ms
 2  2001:1ab::6 (2001:1ab::6)  95.139 ms  94.834 ms 96.574 ms
R4#
```

# IPv6 forwarding - Arista

- Checking SR from Arista point of view

Does it see SR Sub-TLV from R6?

```
R4#sh isis database R6.00-00 detail

IS-IS Instance: LAB VRF: default
IS-IS Level 2 Link State Database
  LSPID          Seq Num    Cksum    Life    IS Flags
  R6.00-00       3184       38204   1104    L2 <>
    Remaining lifetime received: 1198 s Modified to: 1200 s
    NLPID: 0xCC(IPv4) 0x8E(IPv6)
    Hostname: R6
    Area address: 49.0100
    Topology: 0 (IPv4)
    Topology: 2 (IPv6)
    Interface address: 10.1.1.6
    Interface address: 2001:1ab::6
    IS Neighbor      : R5.00              Metric: 10
      IPv4 Neighbor Address: 10.5.6.5
      IPv4 Interface Address: 10.5.6.6
      Adj-sid: 16310 flags: [ L V ] weight: 0x0
    IS Neighbor (MT-IPv6): R5.00          Metric: 10
      Adj-sid: 16312 flags: [ L V F ] weight: 0x0
    Reachability      : 10.1.1.6/32 Metric: 0 Type: 1 Up
      SR Prefix-SID: 6 Flags: [ N ] Algorithm: 0
    Reachability (MT-IPv6): 2001:1ab::6/128 Metric: 0 Type: 1 Up
      SR Prefix-SID: 606 Flags: [ N ] Algorithm: 0
    Router Capabilities: Router Id: 10.1.1.6 Flags: [ ]
      SR Local Block:
        SRLB Base: 15000 Range: 1000
        SR Capability: Flags: [ I V ]
        SRGB Base: 17000 Range: 7000
        Algorithm: 0
        Algorithm: 1
    Segment Binding: Flags: [ ] Weight: 0 Range: 1 Pfx 10.1.1.5/32
      SR Prefix-SID: 5 Flags: [ ] Algorithm: 0
```

R4#

Does it receive node-SID?

```
R4#show isis segment-routing prefix-segments vrf all
```

```
System ID: 1111.1111.0004                               Instance: 'LAB'
SR supported Data-plane: MPLS                            SR Router ID:

<snip>

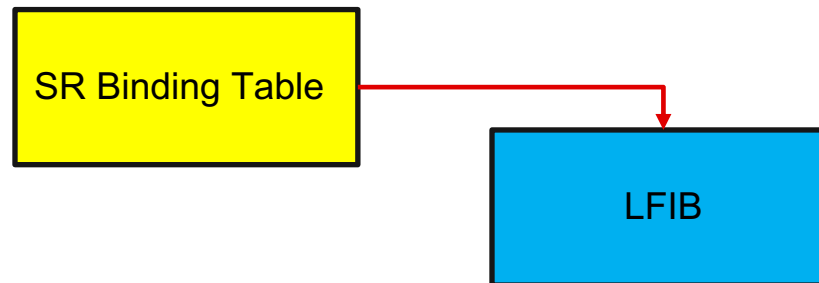
Prefix          SID Type    Flags          System ID
-----
10.1.1.1/32     1 Node      R:0 N:1 P:0 E:0 V:0 L:0 1111.1111.0004
10.1.1.2/32     2 Node      R:0 N:1 P:0 E:0 V:0 L:0 1111.1111.0004
10.1.1.3/32     3 Node      R:0 N:1 P:0 E:0 V:0 L:0 1111.1111.0004
* 10.1.1.4/32    4 Node      R:0 N:1 P:0 E:0 V:0 L:0 1111.1111.0004
10.1.1.5/32     5 Proxy-Node R:0 N:0 P:0 E:0 V:0 L:0 1111.1111.0004
10.1.1.6/32     6 Node      R:0 N:1 P:0 E:0 V:0 L:0 1111.1111.0004
2001:1ab::1/128 601 Node     R:0 N:1 P:0 E:0 V:0 L:0 1111.1111.0004
2001:1ab::2/128 602 Node     R:0 N:1 P:0 E:0 V:0 L:0 1111.1111.0004
2001:1ab::3/128 603 Node     R:0 N:1 P:0 E:0 V:0 L:0 1111.1111.0004
* 2001:1ab::4/128 604 Node     R:0 N:1 P:0 E:0 V:0 L:0 1111.1111.0004
2001:1ab::6/128 606 Node     R:0 N:1 P:0 E:0 V:0 L:0 1111.1111.0004
```

R4#

# IPv6 forwarding - Arista

- Why does Arista drop it?

Arista has advised us that when forwarding using SR the entry is first assigned in this SR-bindings table and then to the LFIB.



LFIB entry appears to be if this can be performed:

- **Local label:** local SRGB Base + node SID (**17000 + 607 = 17606**)
- **Outbound Label:** next-hops SRGB Base (learn via TLVs) + the SID

# IPv6 forwarding - Arista

```
R4#show mpls segment-routing bindings ipv6
2001:1ab::1/128
  Local binding: Label: 17601
  Remote binding: Peer ID: 1111.1111.0002, Label: 17601
2001:1ab::2/128
  Local binding: Label: 17602
  Remote binding: Peer ID: 1111.1111.0002, Label: imp-null
2001:1ab::3/128
  Local binding: Label: 17603
  Remote binding: Peer ID: 1111.1111.0002, Label: 17603
2001:1ab::4/128
  Local binding: Label: imp-null
  Remote binding: Peer ID: 1111.1111.0002, Label: 17604
2001:1ab::6/128
  Local binding: Label: 17606
  Remote binding: Peer ID: 1111.1111.0002, Label: 17606
R4#
```

SID is in the SR-Bindings table

**Local label:** local SRGB Base + node SID ( $17000 + 606 = 17606$ )

**Remote binding:** SRGB Base of next-hop + node SID

This fails as the next hop is not SR capable!  
Never enters LFIB

Note that there are remote bindings, but all of these point to R2 (1111.1111.0002) which is not the IGP next hop

# IPv6 forwarding - Arista

- Arista sees this as a broken LSP
- Offered us to put in a feature request



So how do we work around it?

# Possible Solutions

- Solutions are suitably dull from technical point of view.
- Remove the IPv6 as the next-hop, the MPLS label or both!

## Weight out the link

```
R4(config)#int eth2  
R4(config-if-Et2)#isis ipv6 metric 9999  
R4(config-if-Et2)#int eth4  
R4(config-if-Et4)#isis ipv6 metric 9999
```



# Possible Solutions

## Remove IPv6 Node SID

```
RP/0/RP0/CPU0:R6(config)#router isis LAB
RP/0/RP0/CPU0:R6(config-isis)# interface Loopback0
RP/0/RP0/CPU0:R6(config-isis-if)# address-family ipv6 unicast
RP/0/RP0/CPU0:R6(config-isis-if-af)#no prefix-sid absolute 17606
RP/0/RP0/CPU0:R6(config-isis-if-af)#commit
Wed Feb  8 15:22:21.414 UTC
RP/0/RP0/CPU0:Feb  8 15:22:22.083 UTC: config[67901]: %MGBL-CONFIG-6-
DB_COMMIT : Configuration committed by user 'user1'. Use 'show
configuration commit changes 1000000015' to view the changes.
RP/0/RP0/CPU0:R6(config-isis-if-af)#RP/0/RP0/CPU0:Feb  8 15:22:31.678 UTC:
bgp[1060]: %ROUTING-BGP-5-ADJCHANGE_DETAIL : neighbor 2001:lab::1 Up (VRF:
default; AFI/SAFI: 2/1) (
RP/0/RP0/CPU0:R6(config-i
```

```
RP/0/RP0/CPU0:R1#traceroute 2001:lab::6 source 2001:lab::1
Wed Feb  8 15:23:50.587 UTC
```

```
Type escape sequence to abort.
Tracing the route to 2001:lab::6
```

```
 1  2001:lab:1:2::2 79 msec 4 msec 3 msec
 2  2001:lab:2:4::4 6 msec 5 msec 5 msec
 3  2001:lab:4:5::5 8 msec 8 msec 7 msec
 4  2001:lab::6 12 msec 11 msec 9 msec
```

```
RP/0/RP0/CPU0:R1#
```

**Native forwarding**



# Possible Solutions

## Fall back to 6PE

```
RP/0/RP0/CPU0:R1(config)#router bgp 100
RP/0/RP0/CPU0:R1(config-bgp)#address-family ipv6 unicast
RP/0/RP0/CPU0:R1(config-bgp-af)#allocate-label all
RP/0/RP0/CPU0:R1(config-bgp-af)#exit
RP/0/RP0/CPU0:R1(config-bgp)#neighbor 10.1.1.6
RP/0/RP0/CPU0:R1(config-bgp-nbr)#address-family ipv6 labeled-unicast
```

```
RP/0/RP0/CPU0:R1#sh bgp ipv6 labeled-unicast
Wed Feb  8 15:12:37.771 UTC
BGP router identifier 10.1.1.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0800000 RD version: 4
BGP main routing table version 4
BGP NSR Initial initsync version 2 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs
```

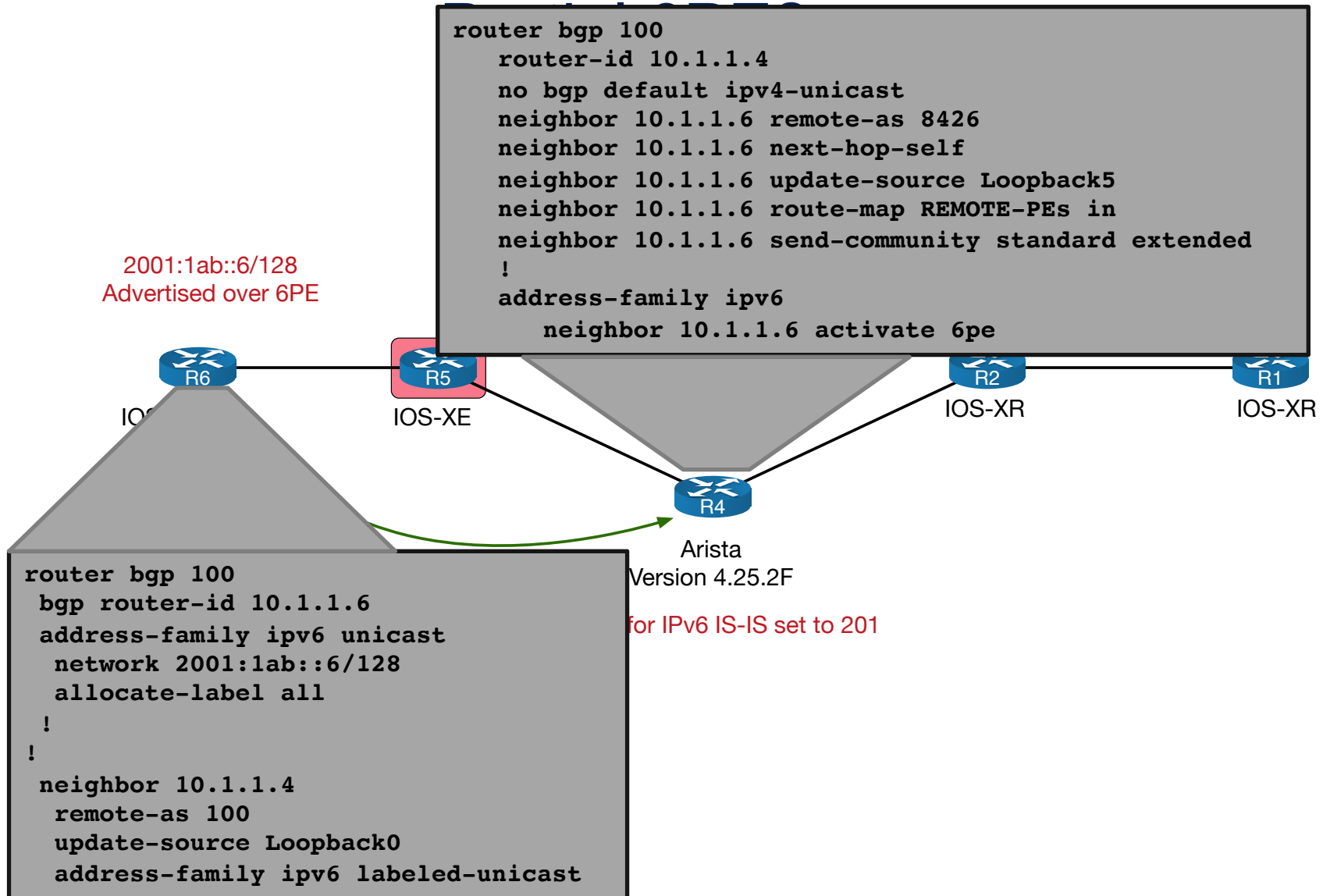
```
Status codes: s suppressed, d damped, h history, * valid, > best
                i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
```

| Network             | Next Hop      | Metric | LocPrf | Weight | Path    |
|---------------------|---------------|--------|--------|--------|---------|
| *> 2001:cafe:1::/64 | 2001:db8:1::1 | 0      |        | 0      | 65489 i |
| *>i2001:cafe:2::/64 | 10.1.1.6      | 0      | 100    | 0      | 65489 i |

```
Processed 2 prefixes, 2 paths
RP/0/RP0/CPU0:R1#
```

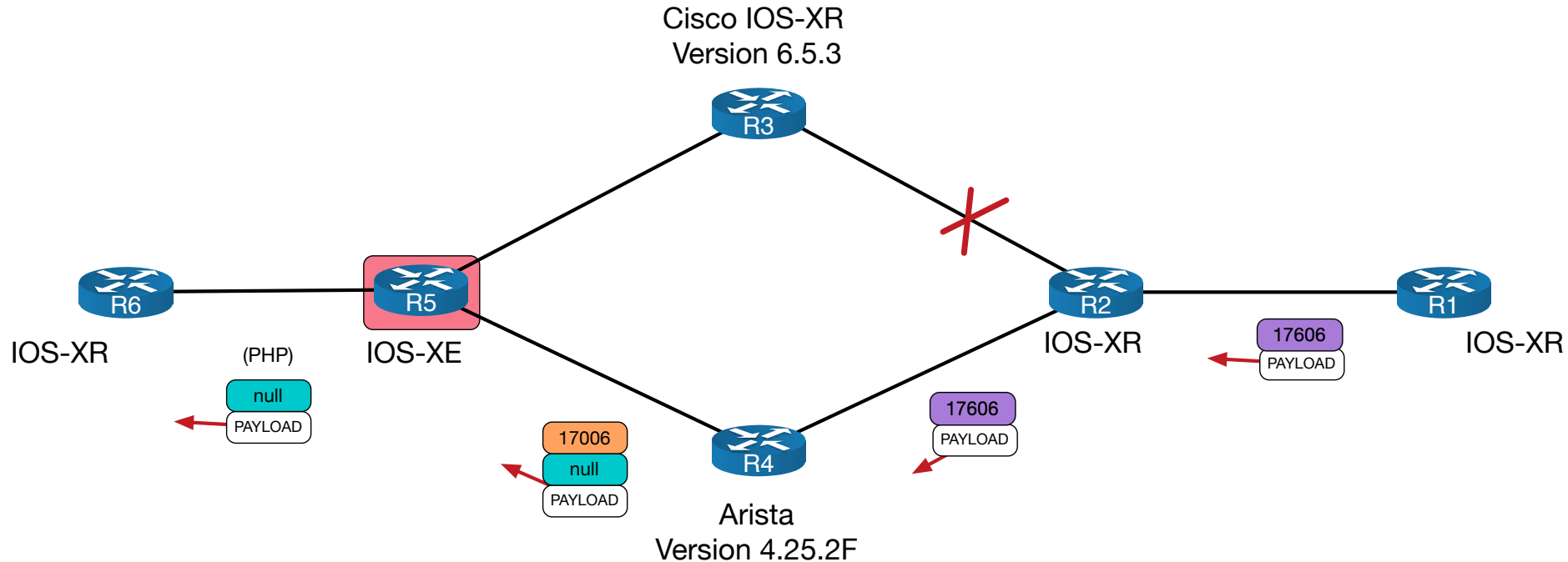
**Our original reason to moving towards native v6 was due to a 6PE edge router allocating one label for IPv6 prefix which caused label exhaustion issues**

# Possible Solutions



# Possible Solutions

## Partial 6PE?



**Could not test in lab  
but possibly scalable?**

```
R4#sh bgp ipv6 labeled-unicast summary
% Not supported
R4#
```

# Possible Solutions

## Partial 6PE?

```
router bgp 100
  bgp router-id 10.1.1.3
  bgp log neighbor changes detail
  address-family ipv6 unicast
    allocate-label all
  !
  neighbor 10.1.1.6
    remote-as 100
    update-source Loopback0
    address-family ipv6 labeled-unicast
      route-policy ONLY-PEs in
    !
  !
  !
  route-policy ONLY-PEs
    if destination in (2001:1ab::6/128)
    then
      pass
    else
      drop
    endif
  end-policy
```

Seems to work with Cisco...

# Possible Solutions

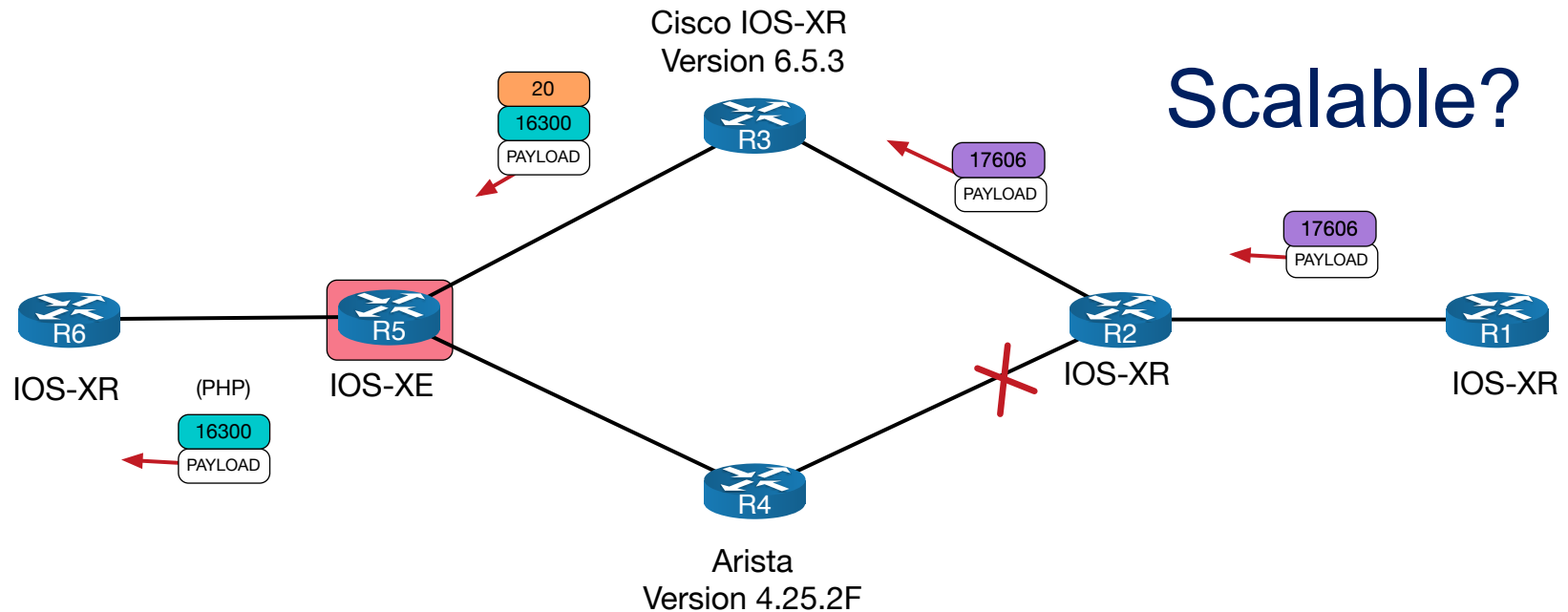
## Partial 6PE?

```
RP/0/RP0/CPU0:R3#sh route ipv6 2001:1ab::6
Thu Feb  9 17:32:36.947 UTC

Routing entry for 2001:1ab::6/128
  Known via "bgp 100", distance 200, metric 0, type internal
  Installed Feb  9 17:27:46.401 for 00:04:50
  Routing Descriptor Blocks
    ::ffff:10.1.1.6, from ::ffff:10.1.1.6
      Nexthop in Vrf: "default", Table: "default", IPv4 Unicast, Table Id:
0xe0000000
      Route metric is 0
      No advertising protos.
RP/0/RP0/CPU0:R3#sh bgp ipv6 labeled-unicast 2001:1ab::6
Thu Feb  9 17:32:46.946 UTC
BGP routing table entry for 2001:1ab::6/128
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          3         3
Last Modified: Feb  9 17:26:35.080 for 00:06:12
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    10.1.1.6 (metric 20) from 10.1.1.6 (10.1.1.6)
      Received Label 16300
      Origin IGP, metric 0, localpref 100, valid, internal, best, group-best,
labeled-unicast
      Received Path ID 0, Local Path ID 1, version 3
RP/0/RP0/CPU0:R3#
```

# Possible Solutions

## Partial 6PE?



Scalable?

```
RP/0/RP0/CPU0:R3#sh mpls forwarding prefix 10.1.1.6/32
```

```
Thu Feb 9 17:35:28.405 UTC
```

| Local Label | Outgoing Label | Prefix or ID   | Outgoing Interface | Next Hop | Bytes Switched |
|-------------|----------------|----------------|--------------------|----------|----------------|
| 17006       | 20             | SR Pfx (idx 6) | Gi0/0/0/2          | 10.3.5.5 | 12703          |

```
RP/0/RP0/CPU0:R3#
```

# Possible Solutions

Upgrade out of the issue



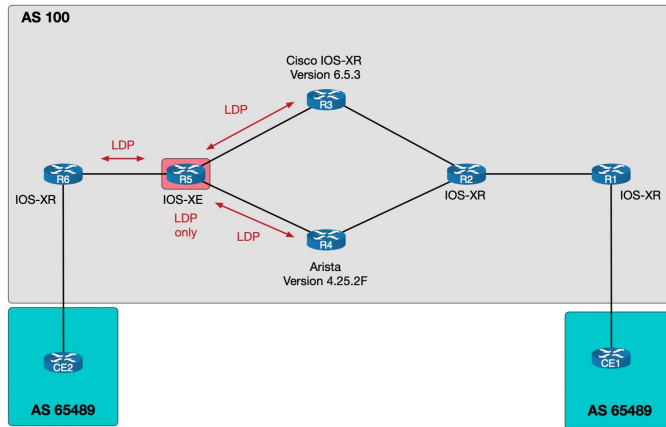
- Other options?
  - Multiple ISIS instances
  - Arista proxy-node-segment command
  - Other tunnelling methods
  - SRv6

# So who is correct?





# MPLS2IP forwarding – broken LSP?



- **Who is correct?**
  - It *is* technically a broken LSP but is there justification for Cisco's behaviour?
- **Are there RFCs to support this?**
  - Not really
  - RFC 8661 acknowledges it...

---

The same applies for the MPLS2IP forwarding entries. MPLS2IP is the forwarding behavior where a router receives a labeled IPv4/IPv6 packet with one label only, pops the label, and switches the packet out as IPv4/IPv6.

RFC 8661 Section 2.1

---

# Any questions?



Steve Crutchley



[steve@netquirks.co.uk](mailto:steve@netquirks.co.uk)



[netquirks.co.uk](http://netquirks.co.uk)



[/stevecrutchleynz](https://www.linkedin.com/company/netquirks/)



[@netquirks](https://twitter.com/netquirks)

[netquirks.co.uk](http://netquirks.co.uk)