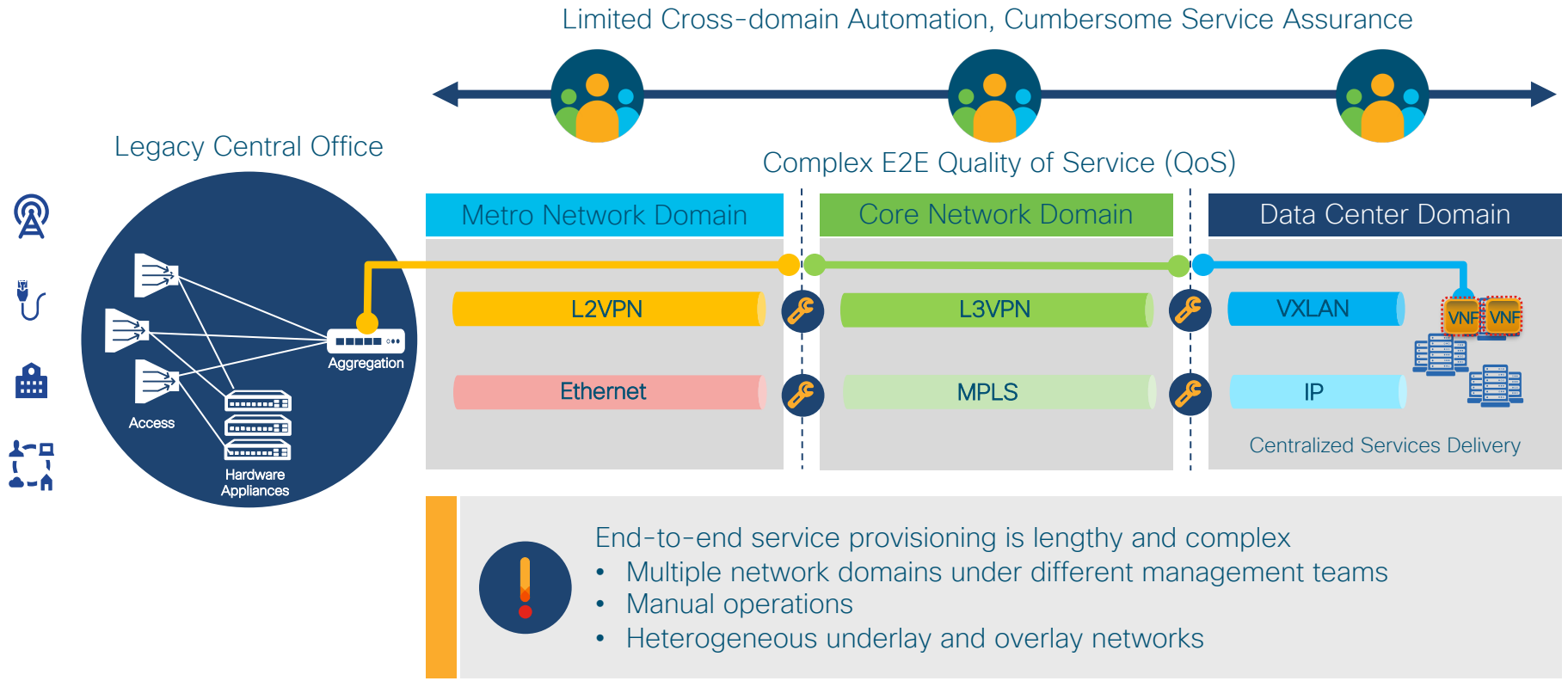
The bridge to possible

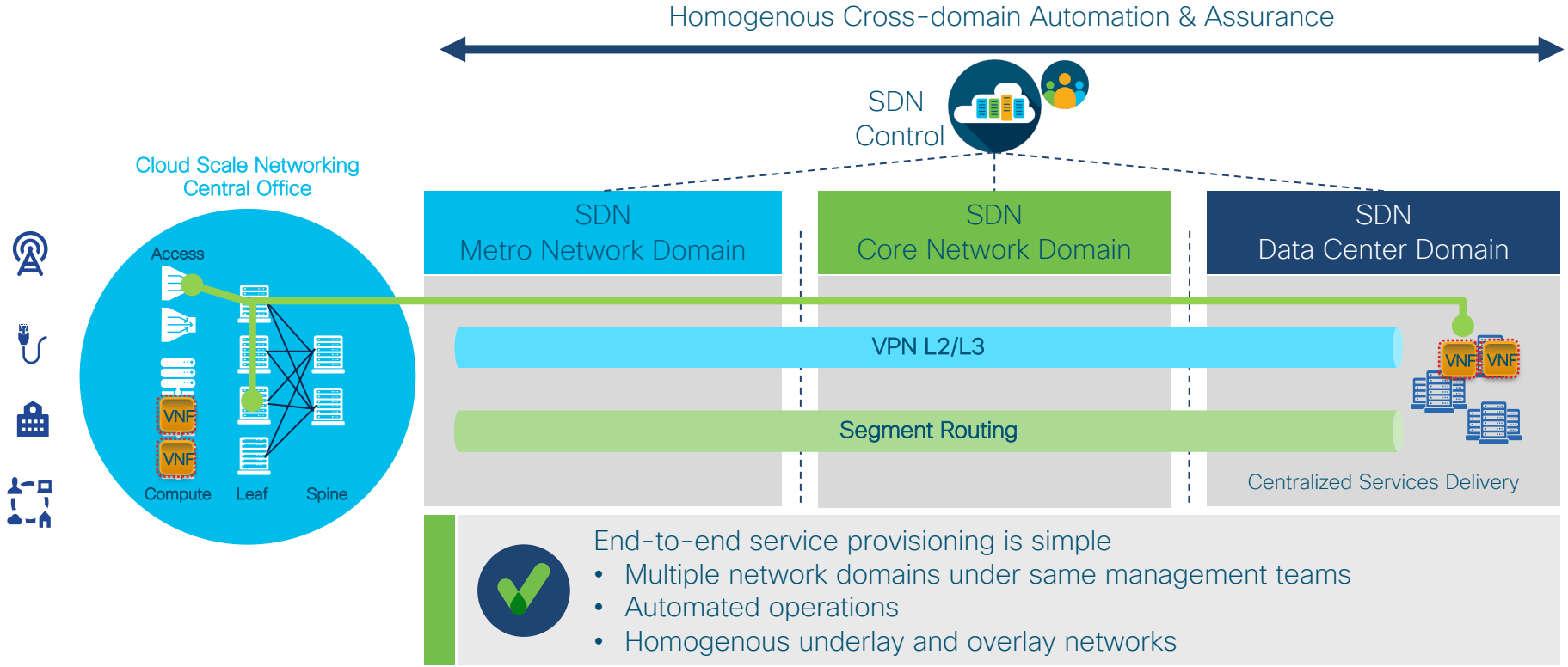
SRv6 uSID Introduction

Jakub Horn
Principal Technical Marketing Engineer
May 2023

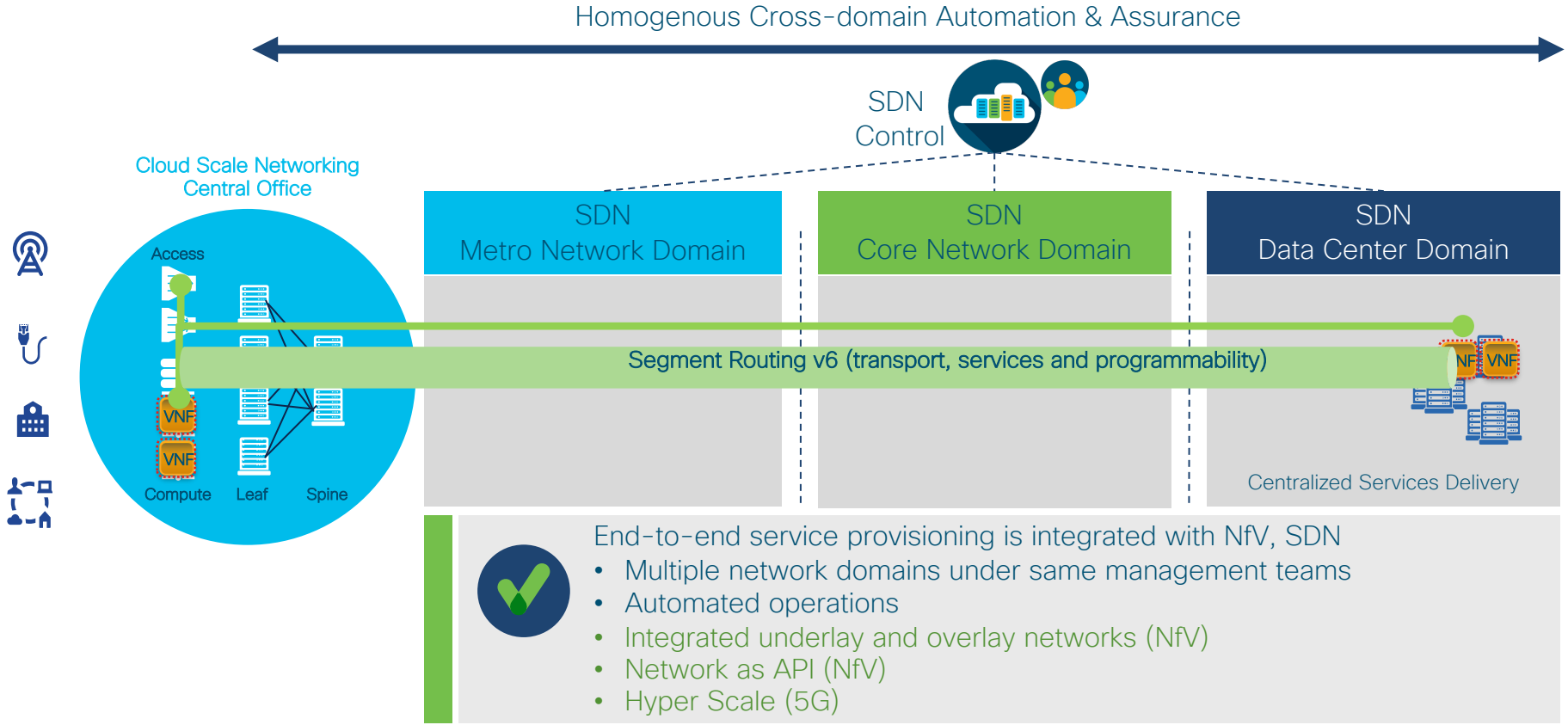
Understanding Today's Service Creation



SR-MPLS: SDN ready
“Network as a Fabric” for Service Creation



SRv6: SDN, NfV, 5G ready
“Network as an API” for Service Creation



End-to-end service provisioning is integrated with NfV, SDN

- Multiple network domains under same management teams
- Automated operations
- Integrated underlay and overlay networks (NfV)
- Network as API (NfV)
- Hyper Scale (5G)

Content:

- uSID Technology
  - SRv6 DataPlane
  - SRv6 Network Programming
  - SRv6 ISIS
  - SRv6 BGP
  - SRv6 Flexible Algorithm
- SRv6 Design
  - Addressing Plan
  - SRv6 Migration

SRv6
IPv6 Segment Routing Header (SRH)

Internet Engineering Task Force (IETF)
Request for Comments: 8754
Category: Standards Track
ISSN: 2070-1721

C. Filsfils, Ed.
D. Dukes, Ed.
Cisco Systems, Inc.
S. Previdi
Huawei
J. Leddy
Individual
S. Matsushima
SoftBank
D. Voyer
Bell Canada

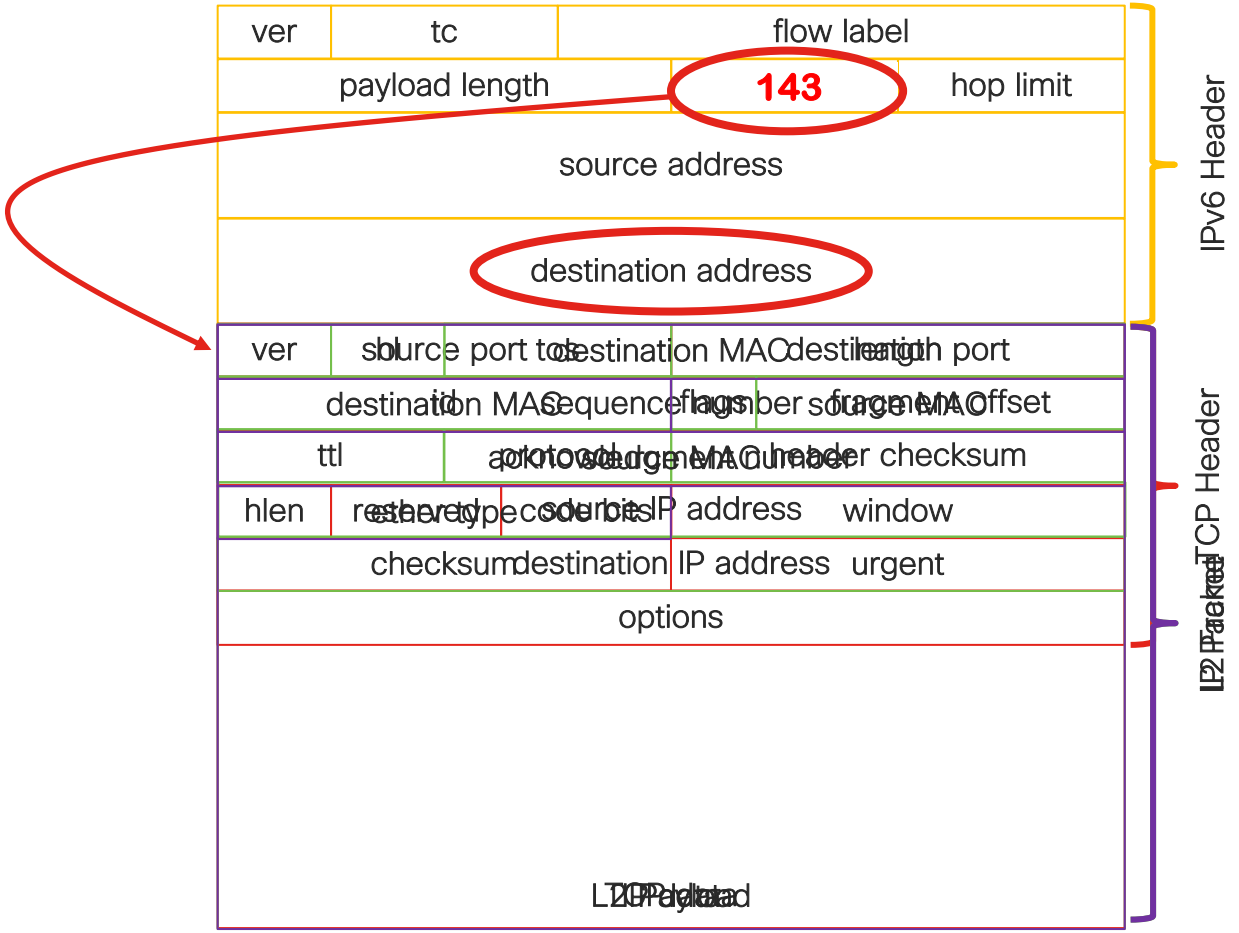
IPv6 Segment Routing Header (SRH)

Abstract

Segment Routing can be applied to the IPv6 data plane using a new type of Routing Extension Header called the Segment Routing Header (SRH). This document describes the SRH and how it is used by nodes that are Segment Routing (SR) capable.

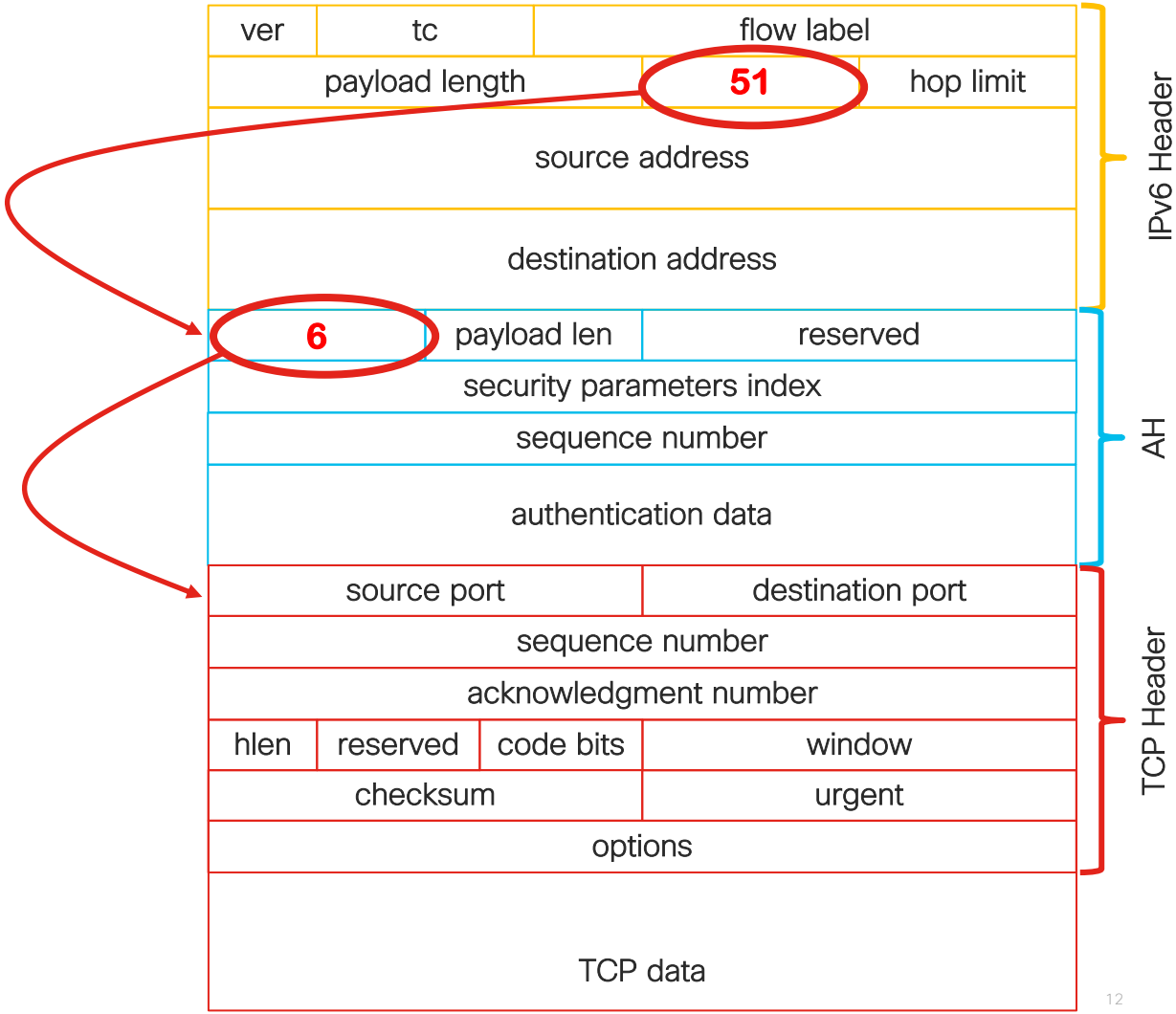
SRv6

- IPv6 Header
- Destination IP address
- Next header field:
  - TCP, UDP, ICMP....
  - IPv4, IPv6, L2



SRv6

- IPv6 Header
- Destination IP address
- Next header field:
  - TCP, UDP, ICMP….
  - IPv4, IPv6, L2
  - Hop by Hop, Dest. Options, Fragmentation, Authentication Header …



SRv6

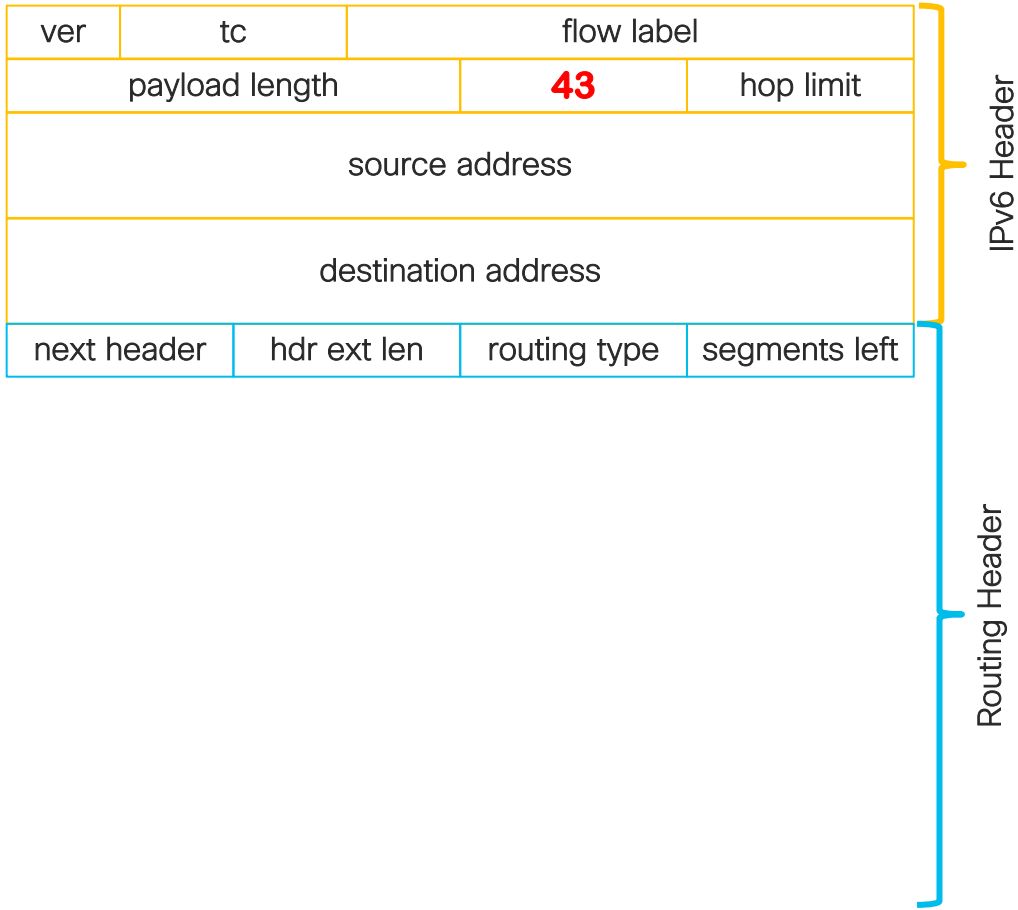
- IPv6 Header
- Destination IP address
- Next header field:
  - TCP, UDP, ICMP....
  - IPv4, IPv6, L2
  - Hop by Hop, Dest. Options, Fragmentation, Authentication Header ...
  - Routing Header

    0 Source Route (deprecated)

    1 Nimrod (deprecated)

    2 Type 2 (RFC 6275)

    3 RPL (RFC 6554)



SRv6

- IPv6 Header
- Destination IP address
- Next header field:
  - TCP, UDP, ICMP....
  - IPv4, IPv6, L2
  - Hop by Hop, Dest. Options, Fragmentation, Authentication Header ...
  - Routing Header

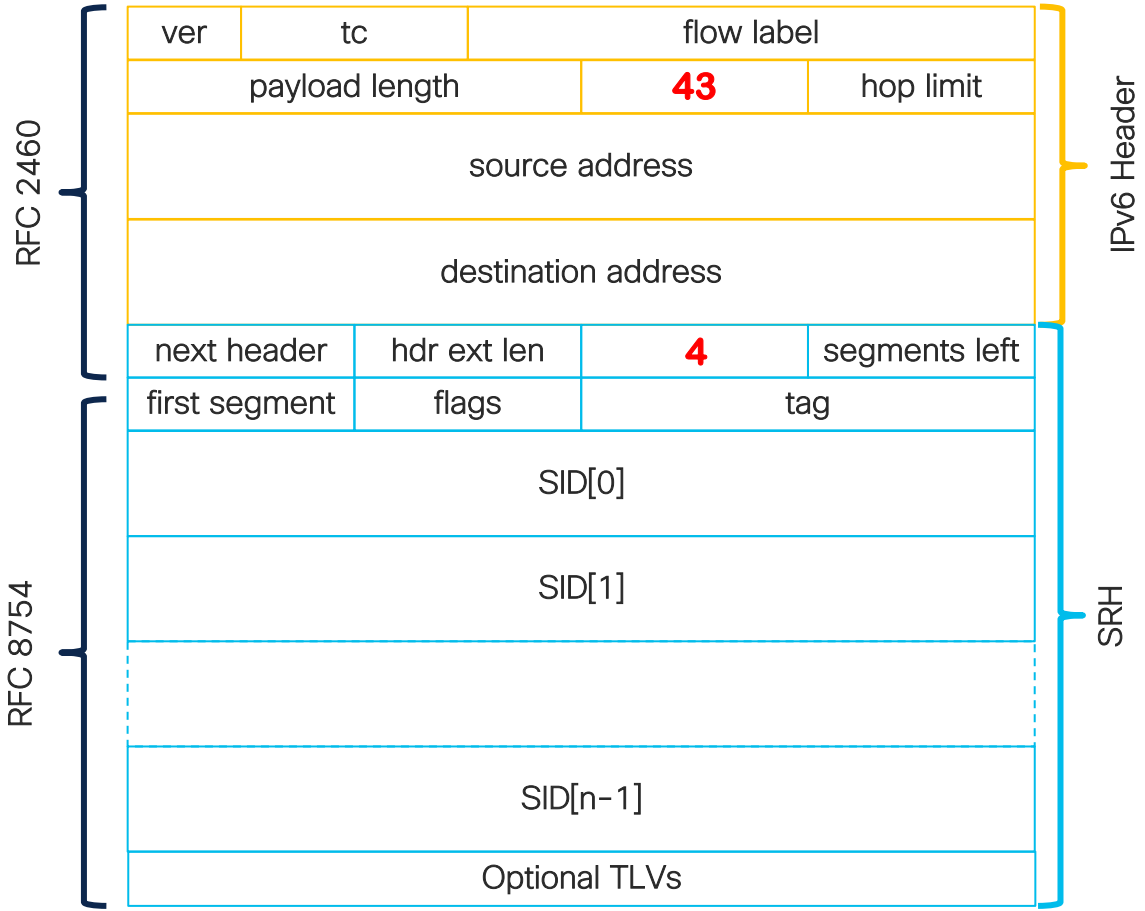
0 Source Route (deprecated)

1 Nimrod (deprecated)

2 Type 2 (RFC 6275)

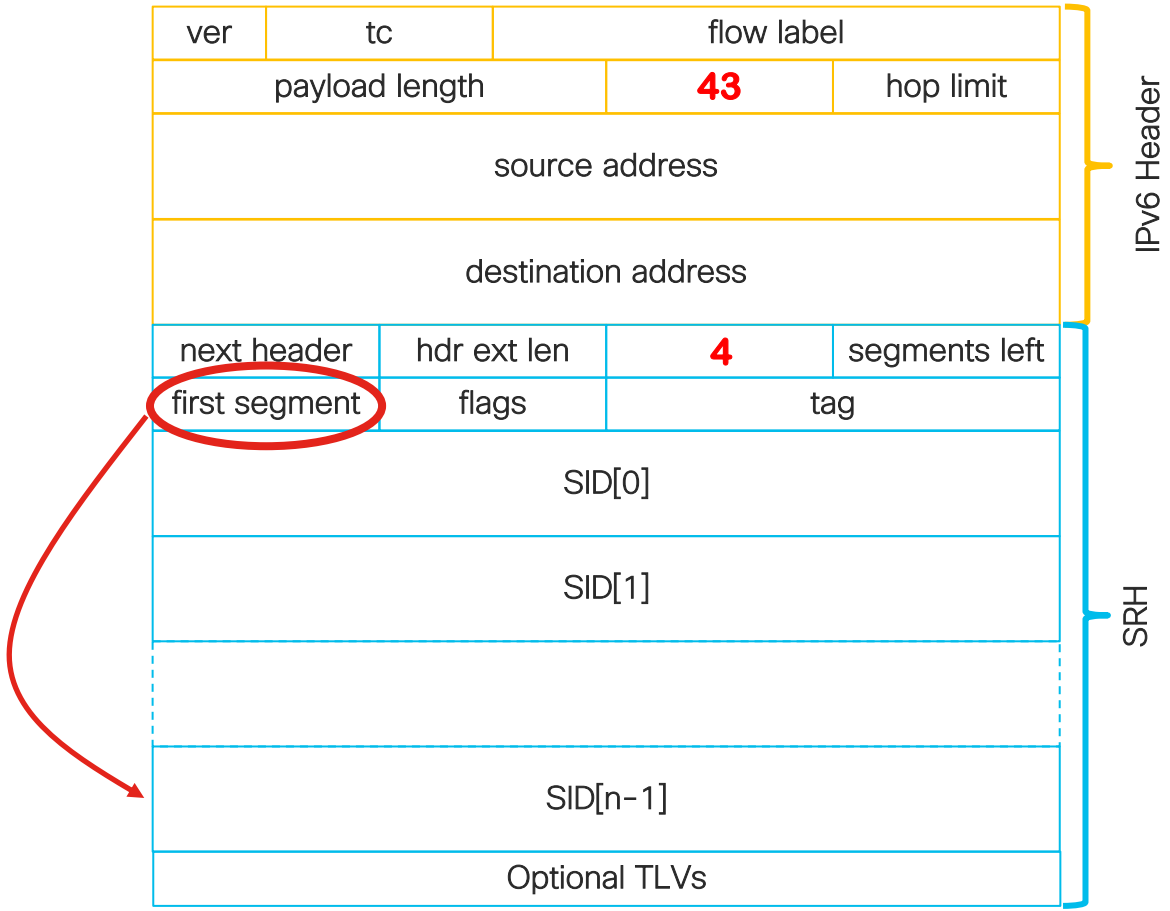
3 RPL (RFC 6554)

4 SRH (RFC 8754)



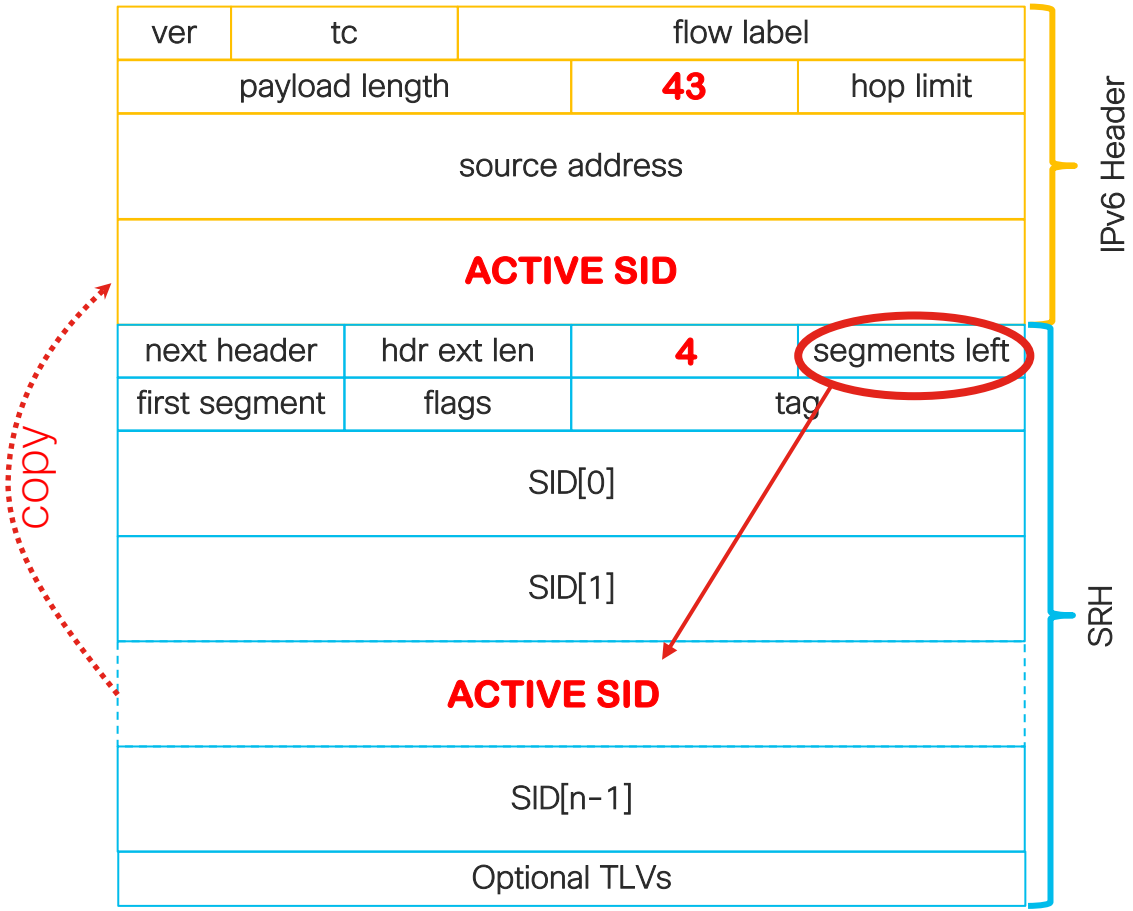
SRH

- Segment Routing Header
- First Segment
  - Pointer to very first SID



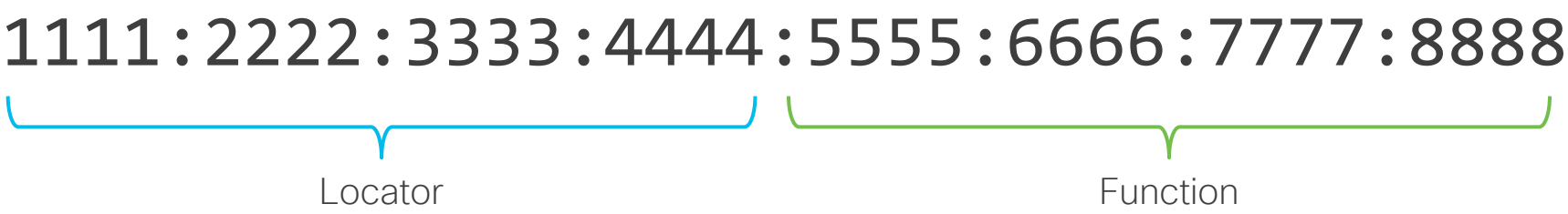
SRH

- Segment Routing Header
- First Segment
  - Pointer to very first SID
- Segments left
  - Pointer to Active SID
  - Active SID always in destination addr

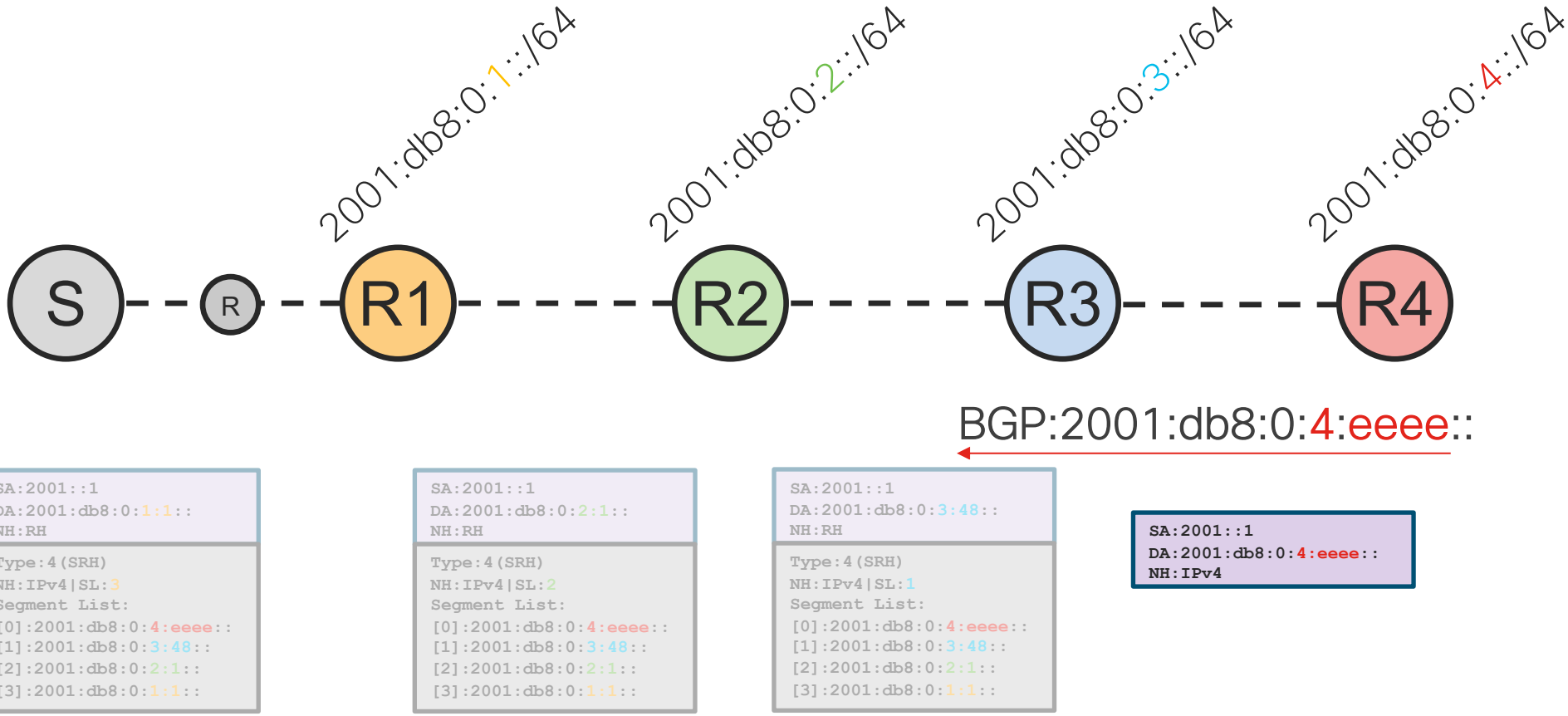


SID Structure -Locator

128 Bits Like IPv6 address but different semantics

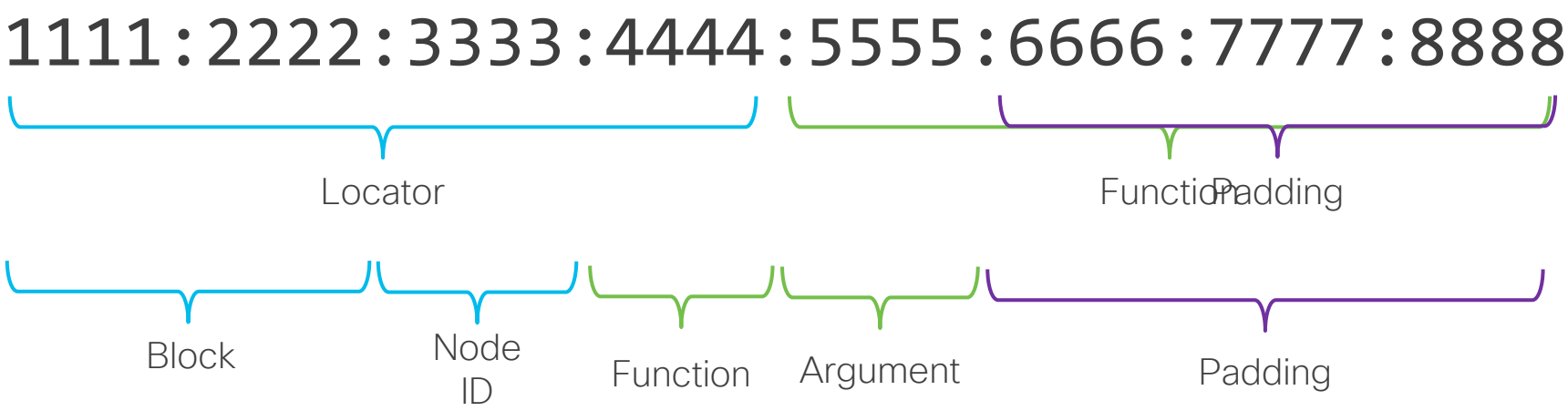


SRv6 Full SID



SID Structure

128 Bits Like IPv6 address but different semantics



SPRING
Internet-Draft
Intended status: Standards Track
Expires: 15 July 2023

W. Cheng, Ed.
China Mobile
C. Filsfils
Cisco Systems, Inc.
Z. Li
Huawei Technologies
B. Decraene
Orange
F. Clad, Ed.
Cisco Systems, Inc.
11 January 2023

SRv6 uSID

Compressed SRv6 Segment List Encoding in SRH
draft-ietf-spring-srv6-srh-compression-03

Abstract

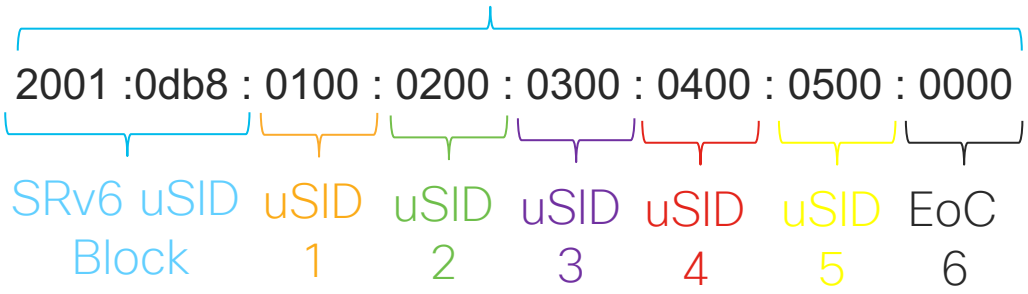
This document specifies new flavors for the SR endpoint behaviors defined in RFC 8986, which enable a compressed SRv6 Segment-List encoding in the Segment Routing Header (SRH).

SRv6 uSID format

: 0100 : =SRV6 uSID

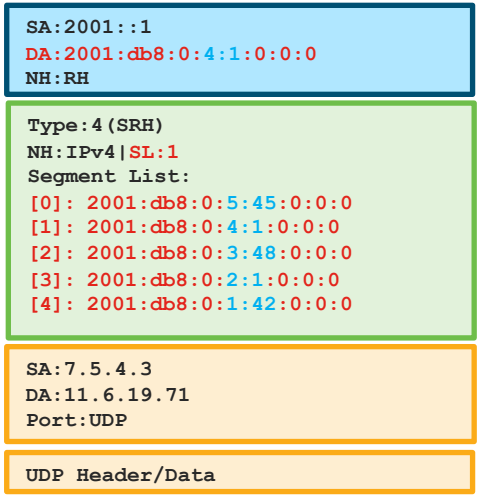
16 bits here, but can be anything

SRV6 uSID Container

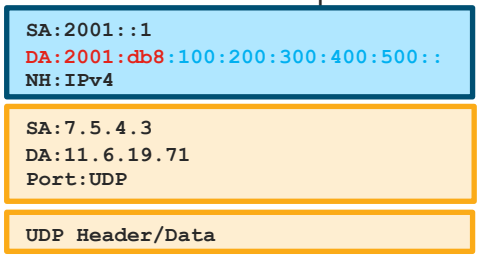


32 bits here,
but can be anything

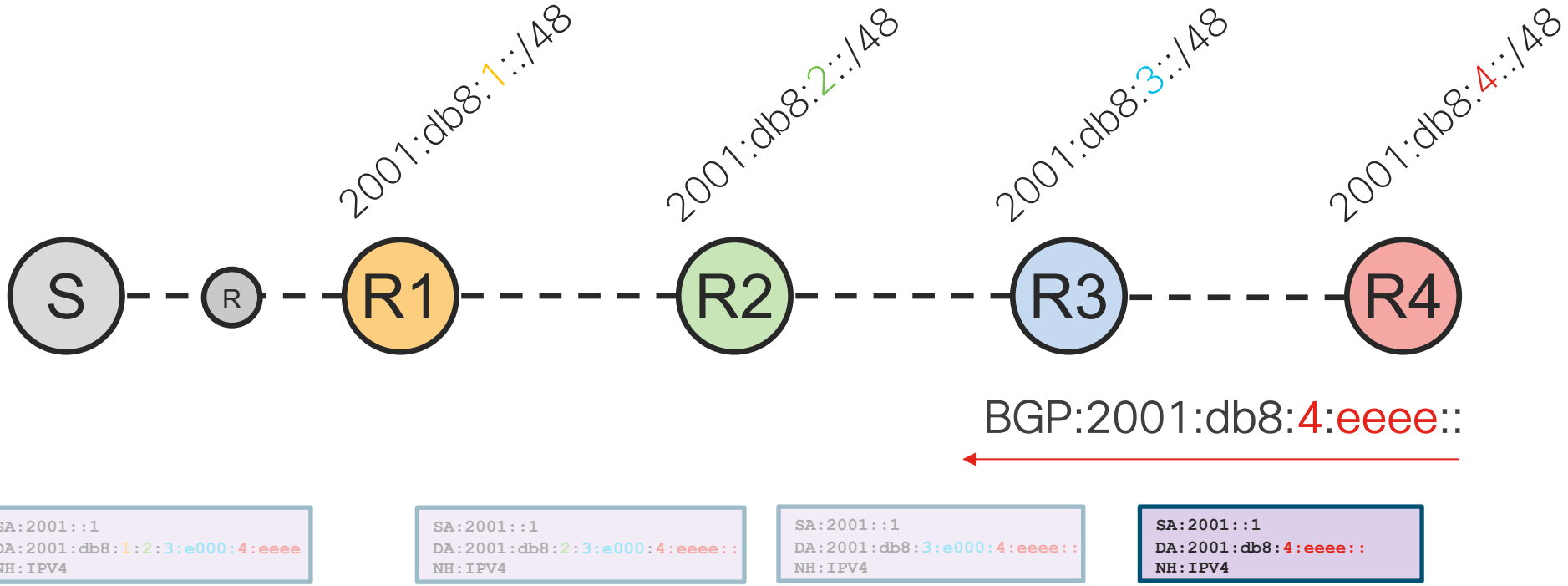
SRV6 Encapsulation



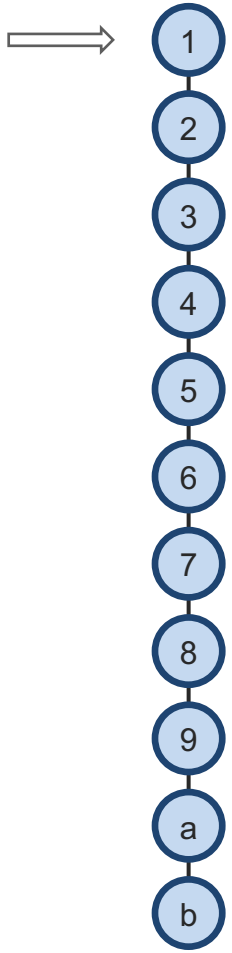
SRV6 uSID Encapsulation



SRv6 uSID F3216



SRv6 uSID More Than 6 SIDs?



100->200->300->400->500->600->700->800->900->a00->b00

Carrier 1 2001 : 0db8 : 0100 : 0200 : 0300 : 0400 : 0500 : 0600

Carrier 2 2001 : 0db8 : 0700 : 0800 : 0900 : 0a00 : 0b00 : 0000

```
SA:2001::1
DA:2001:db8:100:200:300:400:500:600
NH:IPv4
```

```
Type:4(SRH)
NH:IPv4|SL:0
Segment List:
[0]: 2001:db8:700:800:900:a00:b00::
```

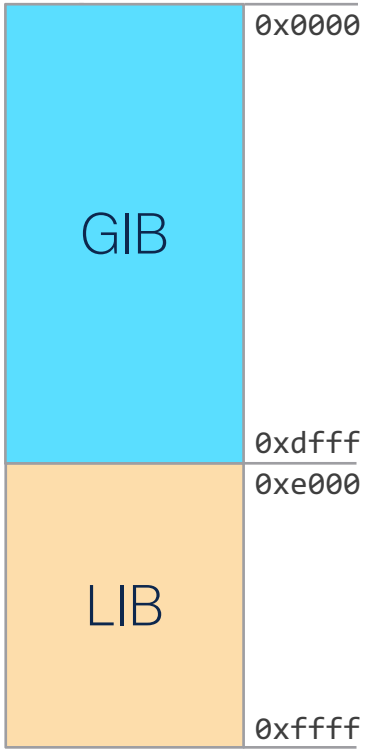
```
SA:7.5.4.3
DA:11.6.19.71
Port:UDP
```

```
UDP Header/Data
```

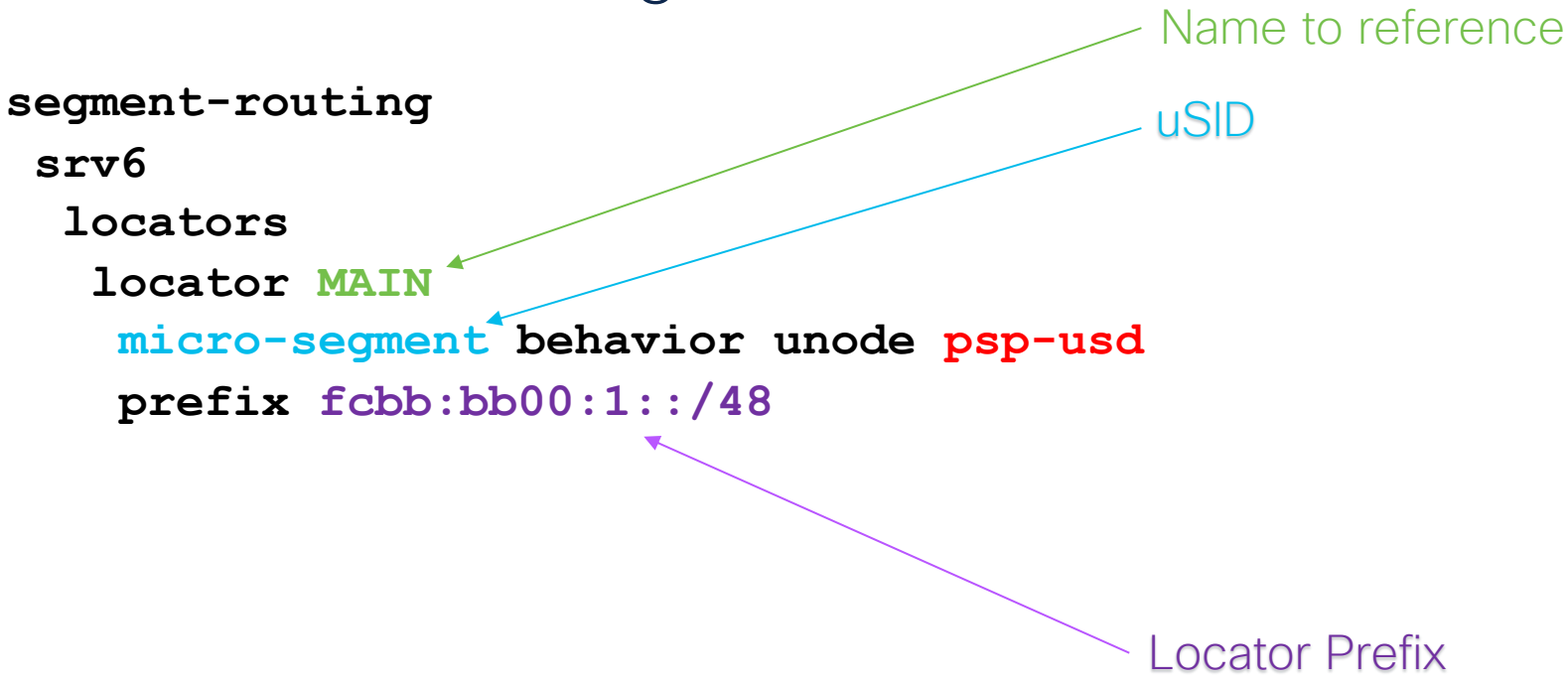
Shift & Forward
END of Carrier
-> is there SRH?
Decrement SL
Copy New SID (Carrier)
PSP

Sets, Global ID Block (GIB), Local ID Block (LIB)

- Within a Block, SIDs are allocated: **FCBB:BB00:SSII**
- SID can be:
  - Global: shortest path to a node – globally unique
  - Local: a local function – not globally unique
- 256 Sets in a Block, identified by "SS"
- Global Sets (GIB): first "S" values 0 to D
  - 224 global Sets → 224 * 256 = 56k global IDs
- Local Sets (LIB): first "S" values E to F
  - 32 local Sets → 32 * 256 = 8k local IDs

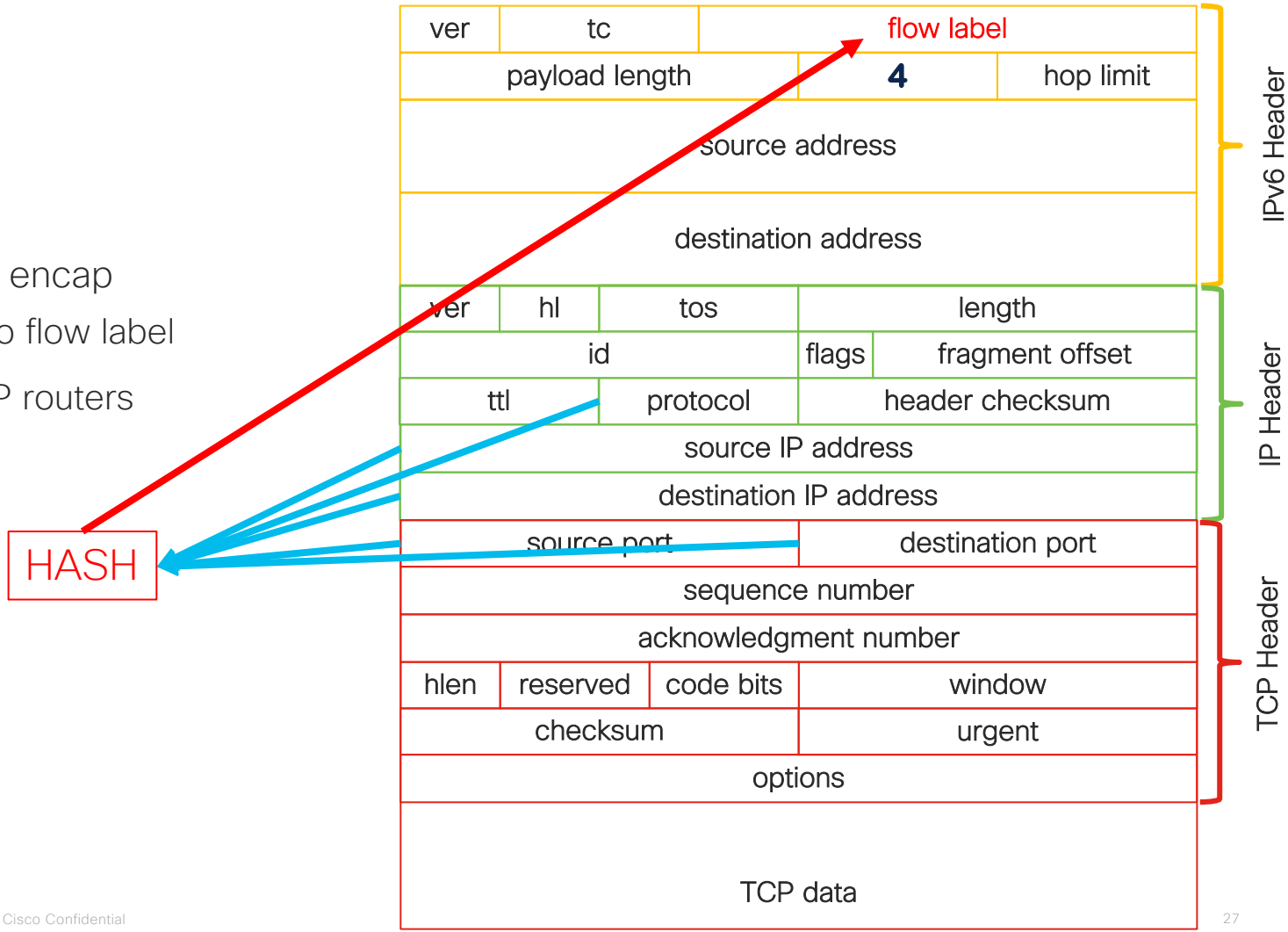


SRv6 uSID Configuration

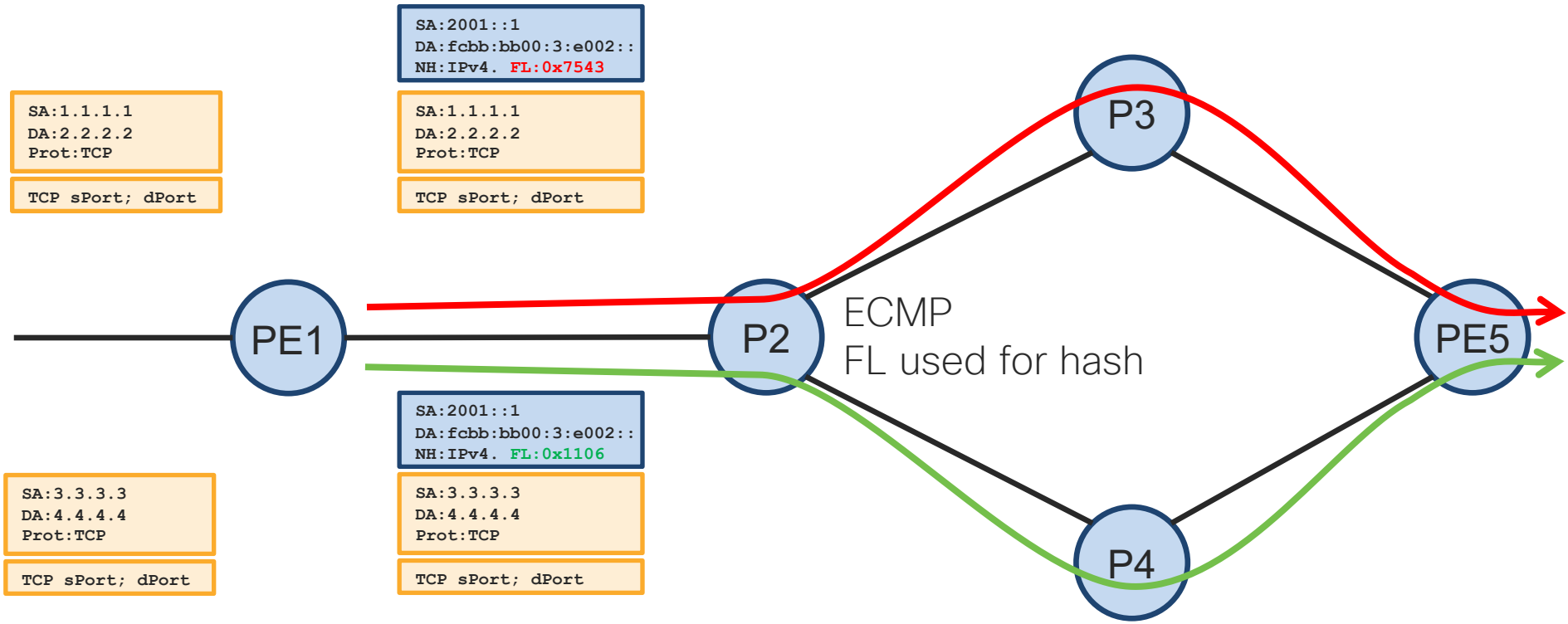


Flow Label

- What for?
- Entropy encoding encap
  - 5 Tuple Hash into flow label
- Used for Hash on P routers



Flow Label



Internet Engineering Task Force (IETF)
Request for Comments: 8986
Category: Standards Track
Published: February 2021
ISSN: 2070-1721

C. Filsfils, Ed.
Cisco Systems, Inc.
P. Camarillo, Ed.
Cisco Systems, Inc.
J. Leddy
Akamai Technologies
D. Voyer
Bell Canada
S. Matsushima
SoftBank
Z. Li
Huawei Technologies

SRv6 Network Programming

Segment Routing over IPv6 (SRv6) Network Programming

Abstract

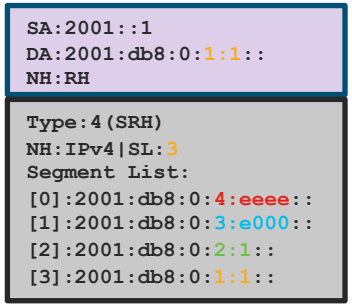
The Segment Routing over IPv6 (SRv6) Network Programming framework enables a network operator or an application to specify a packet processing program by encoding a sequence of instructions in the IPv6 packet header.

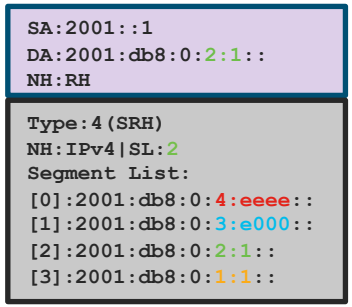
Each instruction is implemented on one or several nodes in the network and identified by an SRv6 Segment Identifier in the packet.

This document defines the SRv6 Network Programming concept and specifies the base set of SRv6 behaviors that enables the creation of interoperable overlays with underlay optimization.

END- Default endpoint (Node SID)

- *Decrement SL*
- *Copy Active SID*
- *Forward*



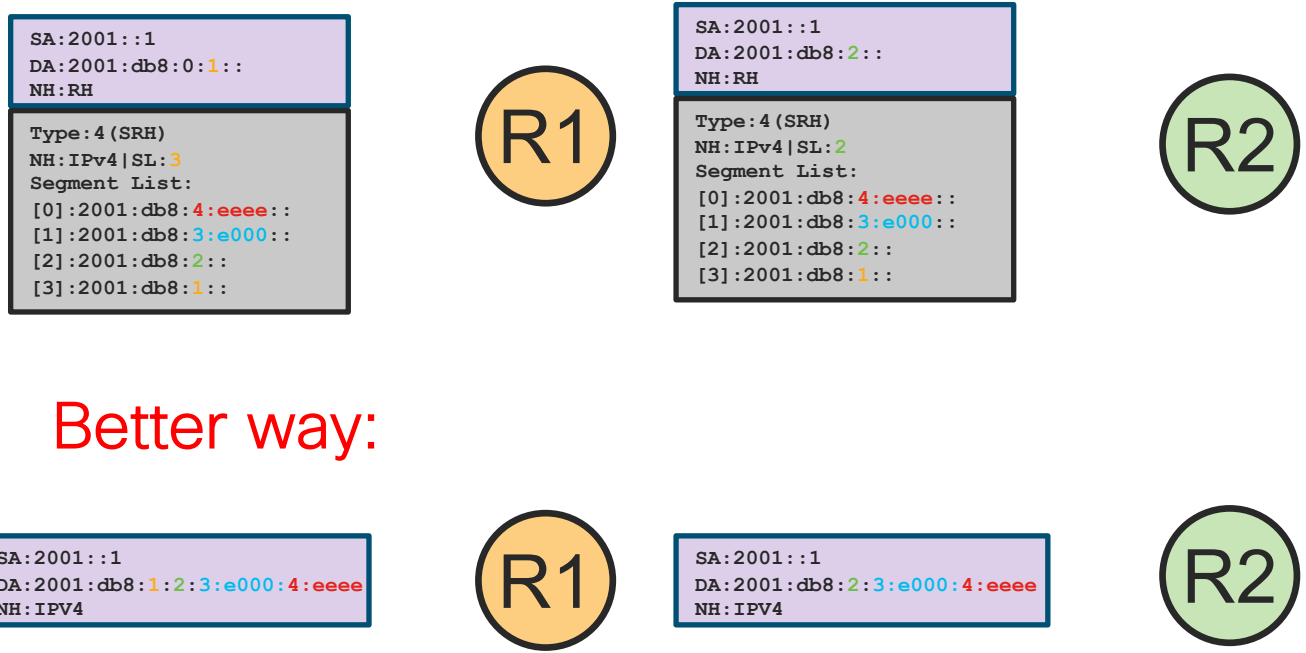


- Different Flavors:
  - End
  - End with PSP
  - End with USP
  - End with PSP & USP
  - End with USD
  - End with PSP & USD
  - End with USP & USD
  - End with PSP, USP & USD

- End with NEXT-ONLY-CSID
- End with NEXT-CSID
- End with NEXT-CSID & PSP
- End with NEXT-CSID & USP
- End with NEXT-CSID, PSP & USP
- End with NEXT-CSID & USD
- End with NEXT-CSID, PSP & USD
- End with NEXT-CSID, USP & USD
- End with NEXT-CSID, PSP, USP & USD

uN=END with Next – Default endpoint (Node SID)

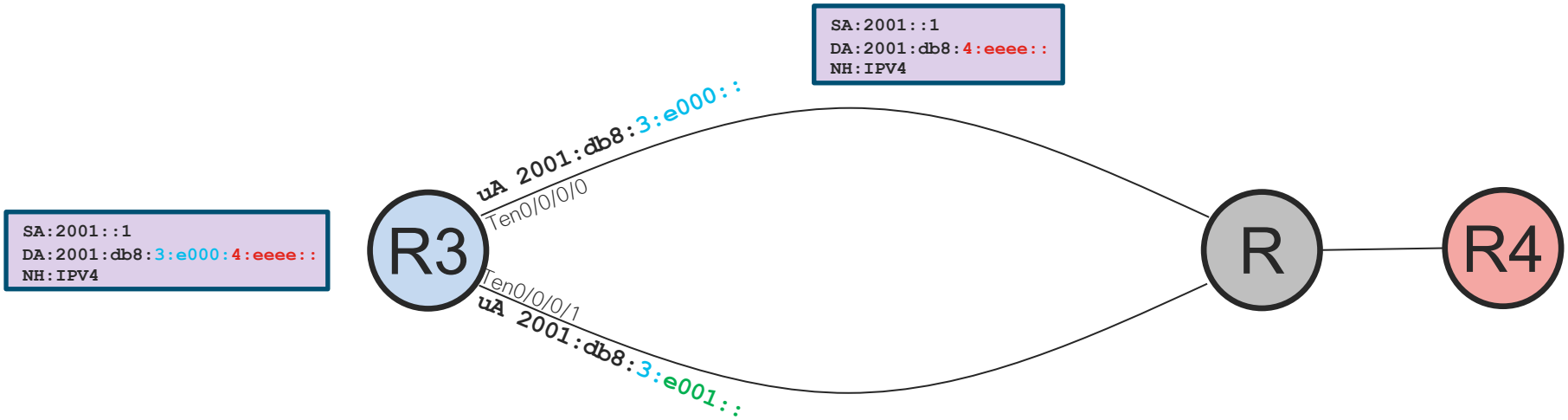
- *Decrement SL*
- *Copy Active SID*
- *Forward*



Better way:

- *Shift & Forward*

uA=END.X with Next – (Adjacency SID)



- *Shift & Forward to SPECIFIC INTERFACE*

uDX4=END.DX4, uDX6=END.DX6, uDX2=END.DX2 Endpoint with Decapsulation and Xconnect

- *Decapsulate and Forward to **SPECIFIC INTERFACE***
- ***Same as Per CE Label Allocation***
- *Must be last function in SID list*

uDT4=END.DT4, uDT6=END.DT6
Endpoint with Decapsulation and Table Lookup

- *Decapsulate and Table Lookup (VRF)*
- *Same as Per VRF Label Allocation (aggregate label)*
- *Must be last function in SID list*

SRv6 functions: Steering and Services

Codename		Behavior	
End	uN	Endpoint	[Node SID]
End.X	uA	Endpoint with Layer-3 cross-connect	[Adj SID]
End.B6.Insert	uB6.Insert	Endpoint bound to an SRv6 policy	[BSID]
End.B6.Encap	uB6.Encaps	Endpoint bound to an SRv6 encapsulation policy	[BSID]
End.DX6	uDX6	Endpoint with decapsulation and IPv6 cross-connect	[L3VPN Per-CE]
End.DX4	uDX4	Endpoint with decapsulation and IPv4 cross-connect	[L3VPN Per-CE]
End.DT6	uDT6	Endpoint with decapsulation and specific IPv6 table lookup	[L3VPN Per-VRF]
End.DT4	uDT4	Endpoint with decapsulation and specific IPv4 table lookup	[L3VPN Per-VRF]
End.DX2	uDX2	Endpoint with decapsulation and L2 cross-connect	[E-LINE]
End.DT2U/M	uDT2U/M	Endpoint with decapsulation and L2 unicast lookup / flooding	[E-LAN]
End.DTM	uDTM	Endpoint with decapsulation and MPLS table lookup	[Interworking]
H.Insert / H.Encaps		Headend with Insertion / Encapsulation of / into an SRv6 policy	[TiLFA]
H. Encaps.L2		H.Encaps Applied to Received L2 Frames	[L2 Port Mode]
H.Encaps.M		H.Encaps Applied to MPLS Label Stack	[Interworking]

Workgroup: Networking Working Group
Internet-Draft:
draft-ietf-lsr-isis-srv6-extensions-19
Updates: 7370 (if approved)
Published: 14 November 2022
Intended Status: Standards Track
Expires: 18 May 2023

P. Psenak, Ed.
Cisco Systems
C. Filsfils
Cisco Systems
A. Bashandy
Cisco Systems
B. Decraene
Orange
Z. Hu
Huawei Technologies

SRv6 ISIS Extensions

IS-IS Extensions to Support Segment Routing over IPv6 Dataplane

Abstract

The Segment Routing (SR) architecture allows flexible definition of the end-to-end path by encoding it as a sequence of topological elements called "segments". It can be implemented over the MPLS or the IPv6 data plane. This document describes the IS-IS extensions required to support Segment Routing over the IPv6 data plane.

This document updates RFC 7370 by modifying an existing registry.

Functions might be signaled differently

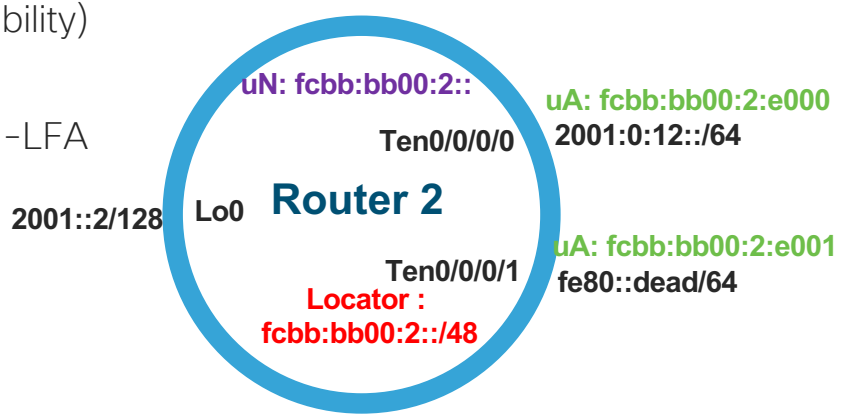
Signalling	IGP	BGP-LS	BGP-IP/VPN
End, uN	Yes	Yes	
End.X, uA	Yes	Yes	
End.T	Yes	Yes	
End.DX4,uDX4		Yes	Yes
End.DX6,uDX6	Yes	Yes	Yes
End.DX2,uDX2		Yes	Yes
END.DT4,uDT4		Yes	Yes
End.DT6,uDT6	Yes	Yes	Yes
End.B		Yes	

Signalling	IGP	BGP-LS	BGP-IP/VPN
T.insert		Yes	
T.Encap		Yes	

Locator – routing table

IGP for uSID

- Uses TLVs
- For Srv6:
  - Locator – for Reachability (twice for backward compatibility)
  - END function – TI-LFA and TE
  - END.X function for each interface in routing protocol TI-LFA and TE
  - Capabilities:
  - Max SID depth for different functions
- OSPF will follow



SRv6 ISIS Configuration

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

Name of the Locator

This will result in:

- Locator is advertised
- uN function is advertised
- uA for each ISIS interface is allocated and advertised

ISIS LSP Example

```
IS-IS 1 (Level-2) Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime/Rcvd  ATT/P/OL
r2.00-00              0x00000009   0x4f06        1145 /1200         0/0/0
  Area Address:   49
  NLPID:          0x8e
  Hostname:       r1
  IPv6 Address:   2001::2
  Metric: 10         MT (IPv6 Unicast) IPv6 2001::2/128
    Prefix Attribute Flags: X:0 R:0 N:1 E:0 A:0
  Metric: 1          MT (IPv6 Unicast) IPv6 fcbb:bb00:2::/48
    Prefix Attribute Flags: X:0 R:0 N:0 E:0 A:0
  MT:             IPv6 Unicast                                  0/0/0
  SRv6 Locator:   MT (IPv6 Unicast) fcbb:bb00:2::/48 D:0 Metric: 0 Algorithm: 0
    Prefix Attribute Flags: X:0 R:0 N:0 E:0 A:0
    END SID: fcbb:bb00:2:: uN (PSP/USD)
      SID Structure:
        Block Length: 32, Node-ID Length: 16, Func-Length: 0, Args-Length: 0
  Router Cap:     0.0.0.0 D:0 S:0
    IPv6 Router ID: 2001::2
    SR Algorithm:
      Algorithm: 0
      Algorithm: 1
    SRv6: O:0
    Node Maximum SID Depth:
      SRH Max SL:       3
      SRH Max End Pop:  3
      SRH Max T.insert: 3
      SRH Max T.encaps: 4
      SRH Max End D:    4
  Metric: 10         MT (IPv6 Unicast) IS-Extended r2.00
    Local Interface ID: 6, Remote Interface ID: 6
    Interface IPv6 Address: 2001:0:0:12::1
    Neighbor IPv6 Address: 2001:0:0:12::2
    END.X SID: fcbb:bb00:2:e000:: B:0 S:0 P:0 uA (PSP/USD) Alg:0
      SID Structure:
        Block Length: 32, Node-ID Length: 16, Func-Length: 16, Args-Length: 0
Total Level-2 LSP count: 1     Local Level-2 LSP count: 0
```

Locator
Capabilities
END
END.X
SID Structure

Content:

- uSID Technology
  - SRv6 DataPlane
  - SRv6 Network Programming
  - SRv6 ISIS
  - SRv6 BGP
  - SRv6 Flexible Algorithm
- SRv6 Design
  - Addressing Plan
  - SRv6 Migration

Internet Engineering Task Force (IETF)
Request for Comments: 9252
Category: Standards Track
ISSN: 2070-1721

G. Dawra, Ed.
LinkedIn
K. Talaulikar, Ed.
Cisco Systems
R. Raszuk
NTT Network Innovations
B. Decraene
Orange
S. Zhuang
Huawei Technologies
J. Rabadan
Nokia

July 2022

SRv6 BGP Overlay Services

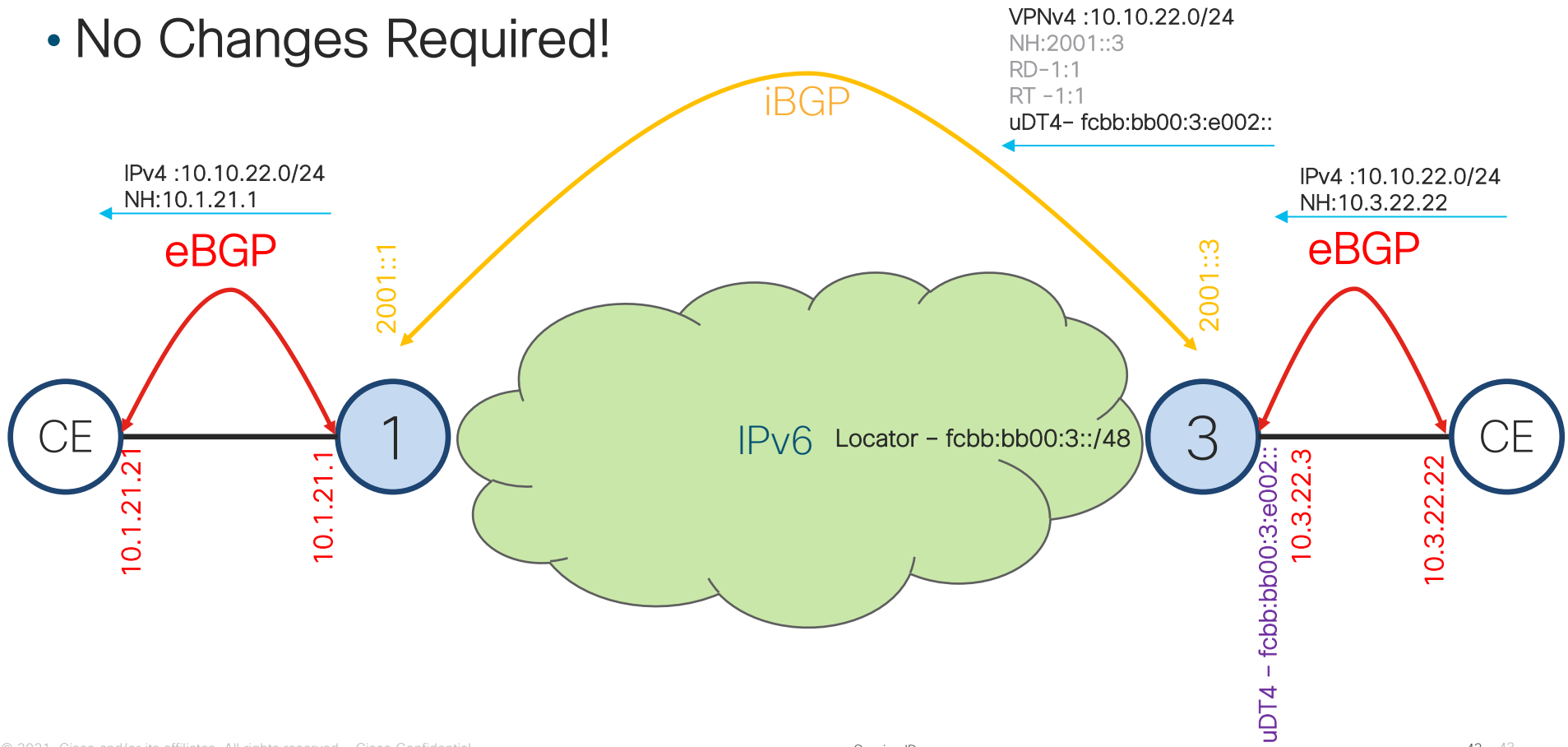
BGP Overlay Services Based on Segment Routing over IPv6 (SRv6)

Abstract

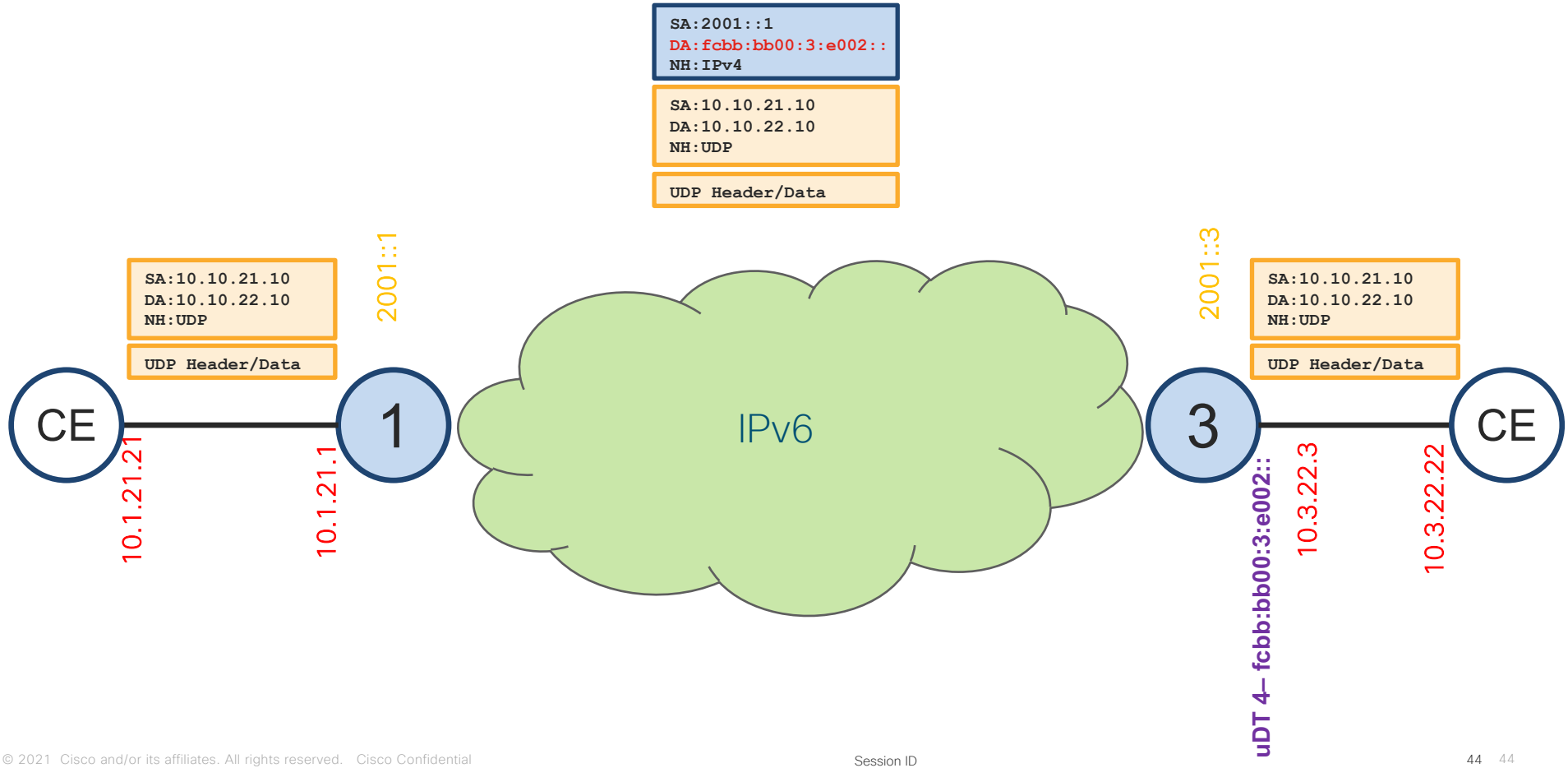
This document defines procedures and messages for SRv6-based BGP services, including Layer 3 Virtual Private Network (L3VPN), Ethernet VPN (EVPN), and Internet services. It builds on "BGP/MPLS IP Virtual Private Networks (VPNs)" (RFC 4364) and "BGP MPLS-Based Ethernet VPN" (RFC 7432).

BGP

- No Changes Required!



L3 VPN Dataplane



SRv6 L3 VPN Configuration

```
router bgp 1
 address-family vpnv4 unicast
 vrf BestEffort
  rd 1:1
  address-family ipv4 unicast
   segment-routing srv6
    locator MAIN
    alloc mode per-vrf
```

Name of the Locator

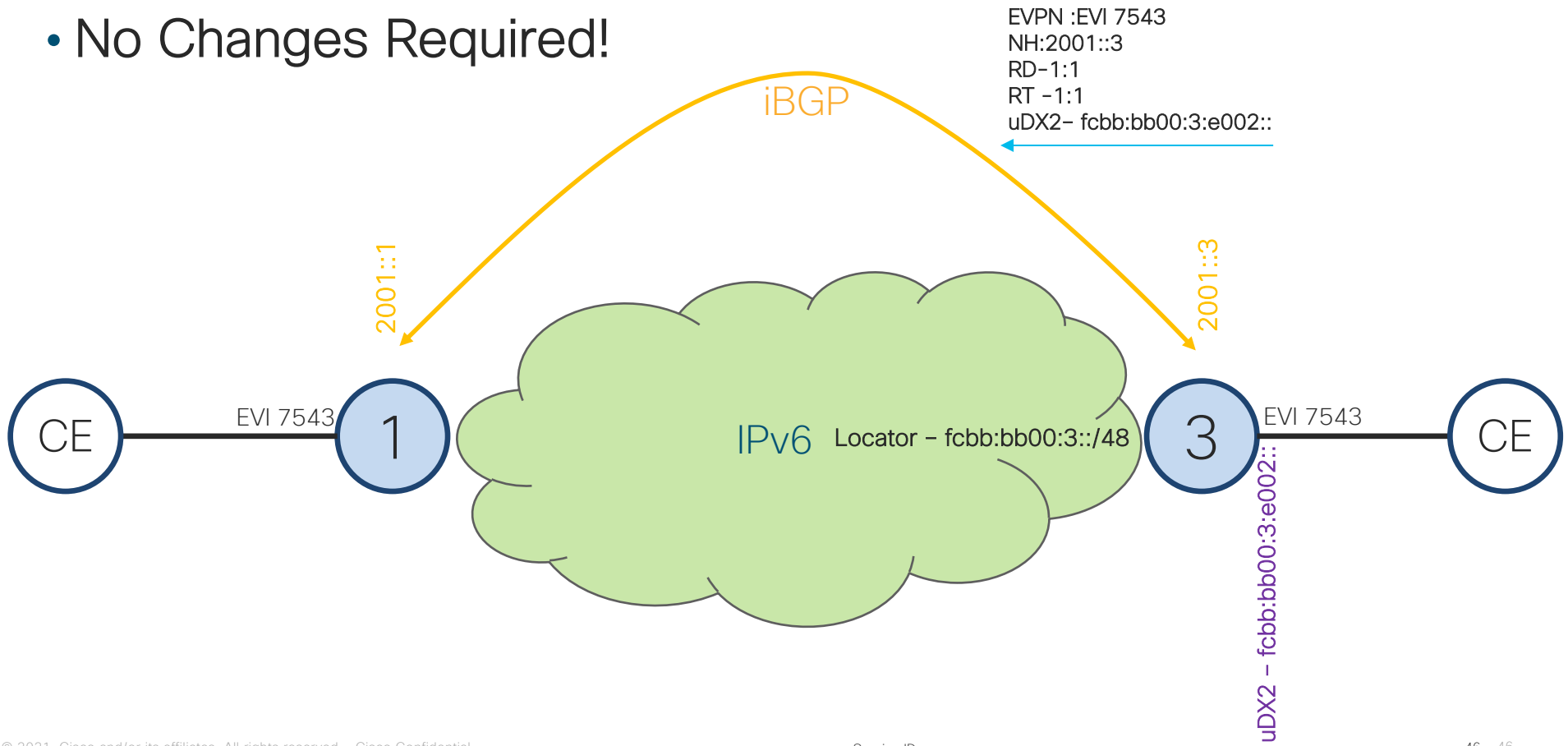
Single DT function is allocated per VRF and AF

This will result in:

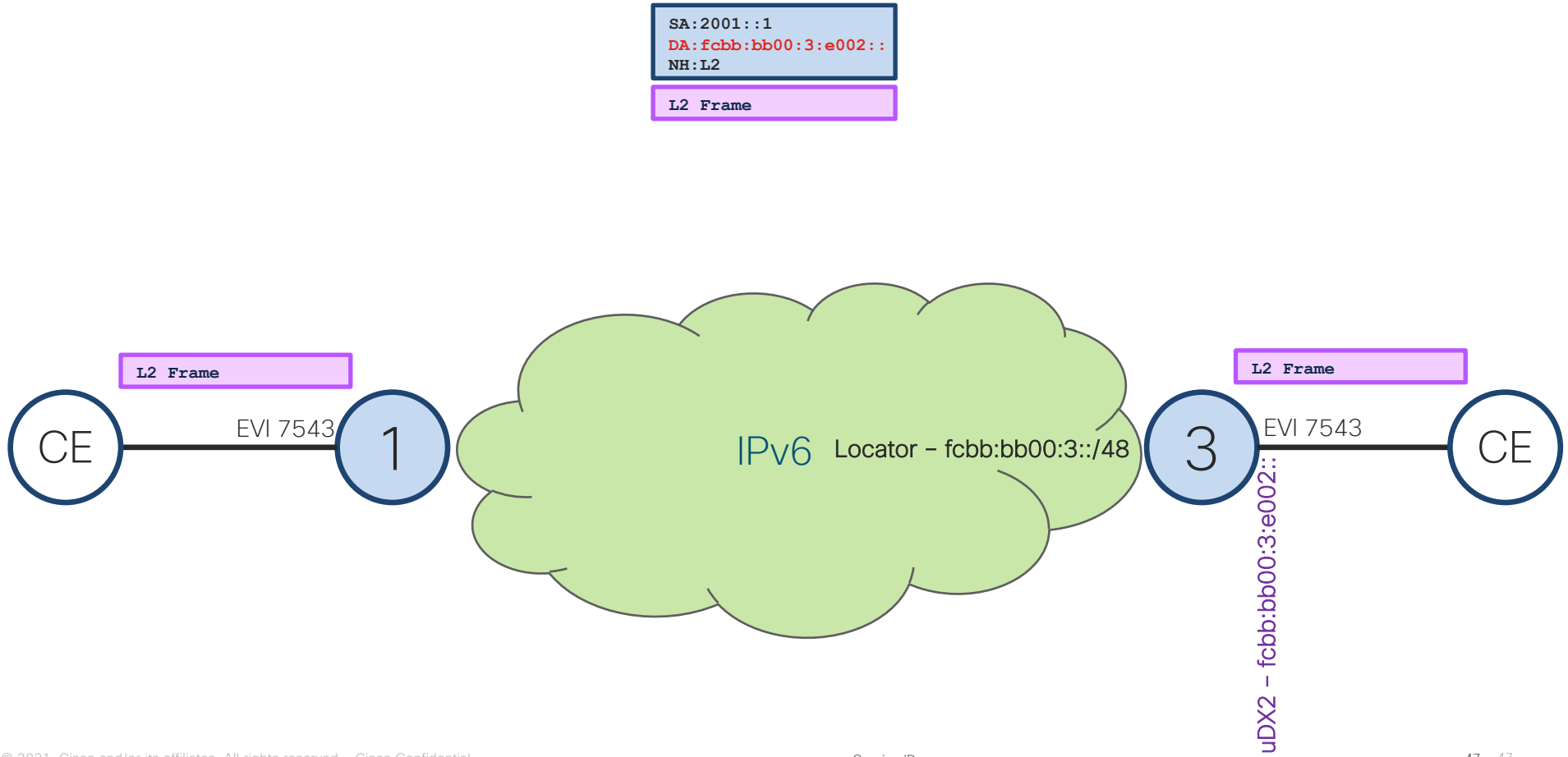
- uDT4 function is allocated
- All prefixes are advertised with uDT4 function

EVPN

- No Changes Required!



EVPN Dataplane



SRv6 L2 VPWS

```
interface TenGigE0/0/0/0.7543 l2transport
 encapsulation dot1q 7543
 rewrite ingress tag pop 1 symmetric

l2vpn
 xconnect group P2P
  p2p 13-14
   interface TenGigE0/0/0/0.7543
   neighbor evpn evi 7543 service 7543
segment-routing srv6

evpn
 evi 7543 segment-routing srv6
  locator MAIN
 segment-routing srv6
!
```

This will result in:

- uDX2 function is allocated per EVI
- EVI is advertised with uDX2 function

Name of the Locator

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 20 April 2023

P. Psenak, Ed.
Cisco Systems, Inc.
S. Hegde
Juniper Networks, Inc.
C. Filsfils
Cisco Systems, Inc.
K. Talaulikar
Cisco Systems, Inc
A. Gulko
Edward Jones
17 October 2022

SRv6 Flexible Algorithm

IGP Flexible Algorithm
draft-ietf-lsr-flex-algo-26

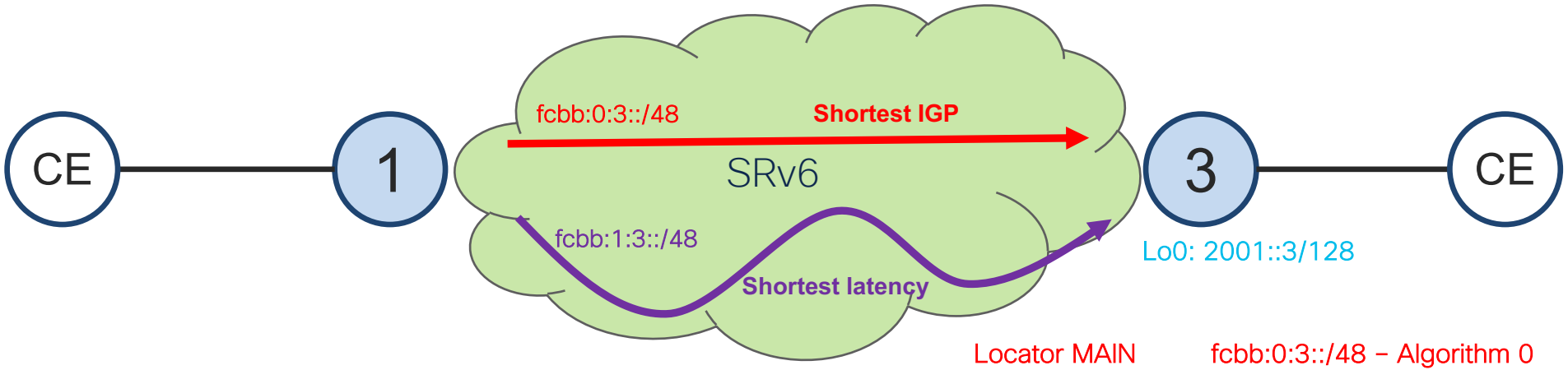
Abstract

IGP protocols historically compute best paths over the network based on the IGP metric assigned to the links. Many network deployments use RSVP-TE based or Segment Routing based Traffic Engineering to steer traffic over a path that is computed using different metrics or constraints than the shortest IGP path. This document specifies a solution that allows IGPs themselves to compute constraint-based paths over the network. This document also specifies a way of using Segment Routing (SR) Prefix-SIDs and SRv6 locators to steer packets along the constraint-based paths.

Flexible Algorithm

- We call “Flex-Algo”
  - The algorithm is defined by the operator, on a per-deployment basis
- Flex-Algo K is defined as
  - The minimization of a specified metric: IGP, delay, …
  - The exclusion of certain link properties: link-affinity, SRLG, …
- Example
  - Operator1 defines Flex-Algo 128 as “minimize IGP metric and avoid link-affinity “green”
  - Operator2 defines Flex-Algo 128 as “minimize delay metric and avoid link-affinity “blue”

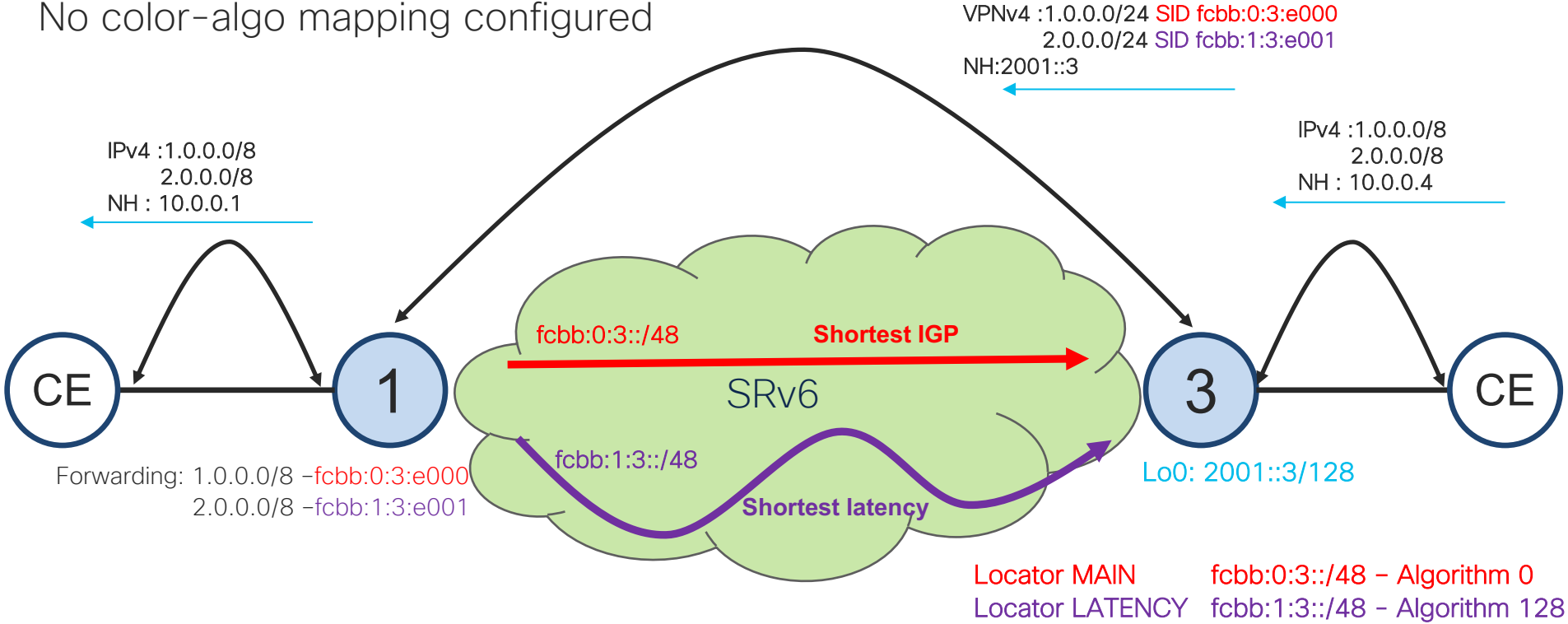
SRv6 Flex Algo IGP



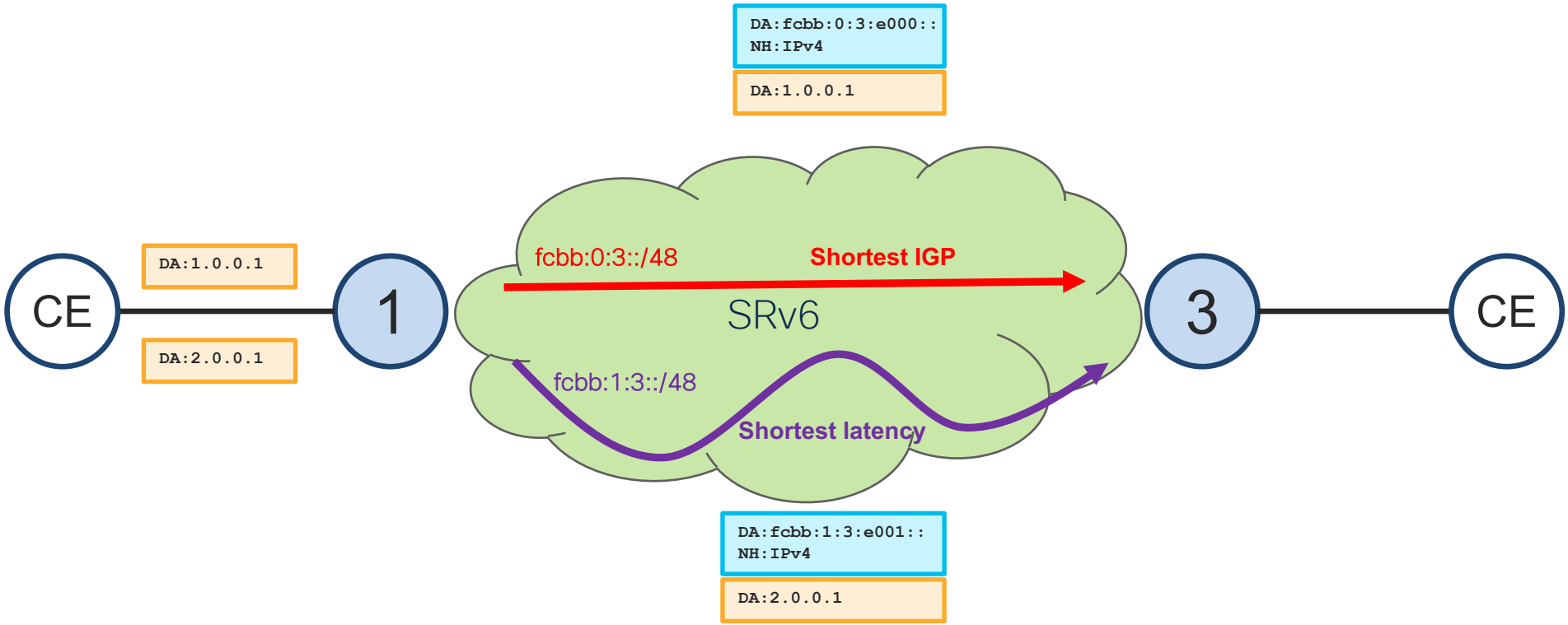
BGP – SRv6

No color advertised

No color-algo mapping configured



SRv6 DATAPLANE



SRv6 Flex Algo -IGP

```
segment-routing
 srv6
  locators
   locator LATENCY
    micro-segment behavior unode psp-usd
    prefix fcbb:bb01:1::/48
    algorithm 128
router isis 1
 flex-algo 128
  metric-type delay
  advertise-definition
 address-family ipv6 unicast
  segment-routing srv6
   locator LATENCY
```

New Locator Name

Locator Prefix (Different)

Flex Algo number 128-255

Definition of specific Flex Algo
Latency metric for 128

This Router will advertise
FA definition within the domain

This will result in:

- Locator is advertised +FA definition
- uN function is advertised - for FA
- uA for each ISIS interface is allocated and advertised for FA

SRv6 L3 VPN Flex Algo

```
router bgp 1
 address-family vpnv4 unicast
 vrf LowLatency
  rd 1:2
  address-family ipv4 unicast
   segment-routing srv6
    locator LATENCY
    alloc mode per-vrf
```

Name of the Locator

Single DT function is allocated per VRF and AF

This will result in:

- uDT4 function is allocated from LATENCY locator
- All prefixes in VRF are advertised with uDT4 function

SRv6 L3 VPN Multiple Algorithms in VRF

```
route-policy MIX
  if destination in (1.1.1.1/32) then
    set srv6-alloc-mode per-vrf locator LATENCY
  else
    set srv6-alloc-mode per-vrf locator MAIN
  endif
end-policy
router bgp 1
 vrf Both
  address-family ipv4 unicast
    segment-routing srv6
      alloc mode route-policy MIX
```

For prefix 1.1.1.1 we will allocate uDT from LATENCY

For all others uDT from MAIN

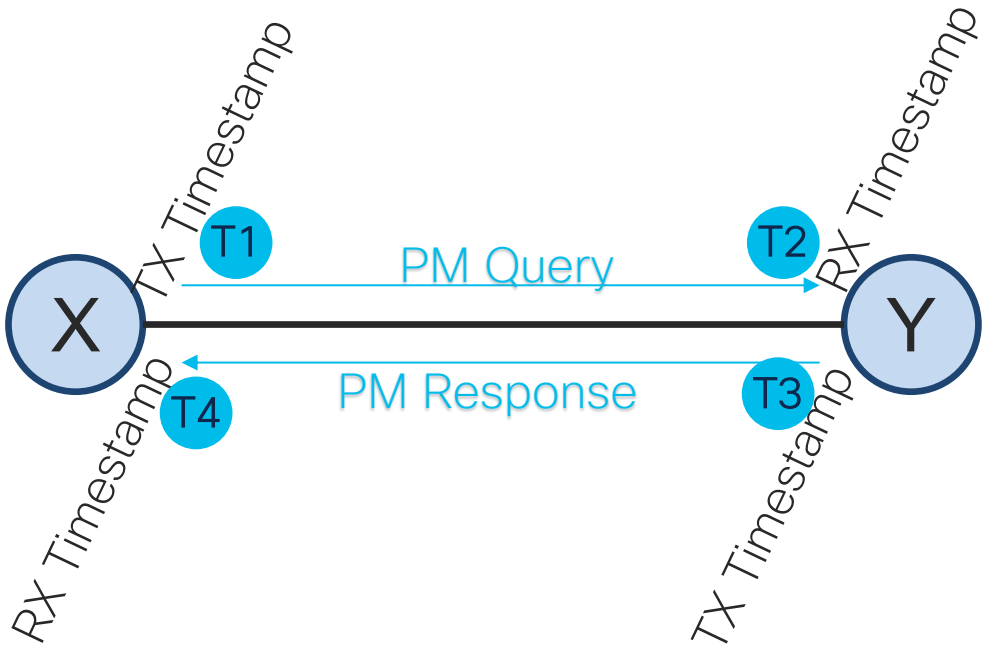
Route-Policy application

Latency Configuration

```
performance-measurement
 interface Gig0/0/0/0
  delay-measurement
   advertise-delay 7543
```

This will set latency of the link to 7543 microseconds

Performance Measurement



- TWAMP Ligth Protocol
- HW Level Timestamping
- ns precision!
- Link Latency Calculation:
  - One WAY Measurement

    $$Latency = T2 - T1$$
  - Two Way Measurement

    $$Latency = \frac{(T4 - T1) - (T3 - T2)}{2}$$

PM Configuration

```
performance-measurement
 interface Gig0/0/0/0
  delay-measurement
```

This will:

- Start PM probes on interface
- Provide Dynamic measurement values to IGP
- Both ends must be PM capable (provide HW based timestamping)

SRv6 Addressing

Separation between SIDs and addresses

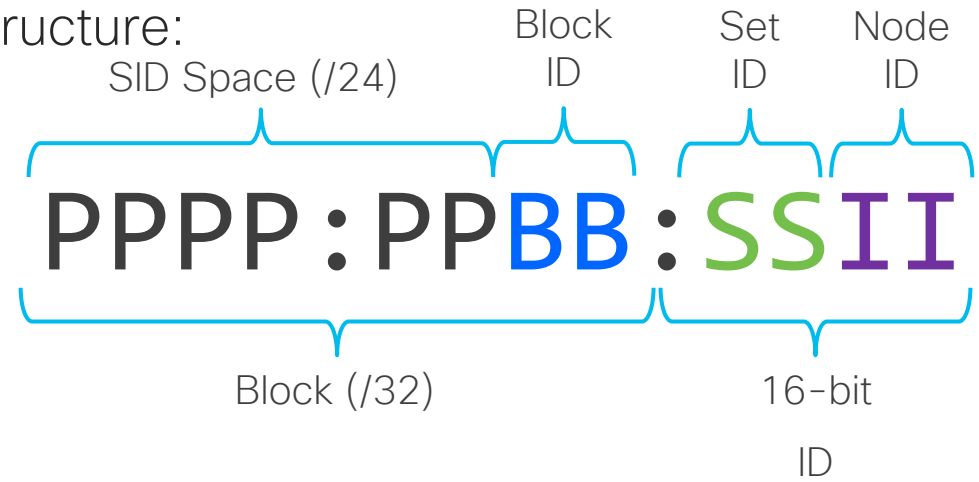
- Infrastructure addressing and SRv6 SID allocation belong to two different planes and are different
  - Infrastructure IP addresses (e.g., link interfaces, loopbacks) are allocated on the management plane
  - SRv6 SIDs are allocated on the service plane
- SRv6 SIDs are assigned to a node independently from the IP addressing of that node
- Even if they are both represented as IPv6 addresses, infrastructure addresses and SIDs cannot be merged and should be allocated off different blocks.

An existing IPv6 address plan is not a constraint for a future SRv6 SID allocation plan.

Terminology – uSID F3216

- uSID F3216: uSID format with
  - uSID Block size: 32 bits
  - ID size: 16 bits

- uSID F3216 structure:



SRv6 Space allocation recommendation

- Private range allocation ✓
  - Recommended allocation
  - Use /24 sub-range from ULA **FC00::/8** space
  - **FCBB:BB**00::/24, with **B** indicating a nibble value picked by operator

- Public range allocation ⚠
  - Supported, not advised
  - From allocated public GUA range

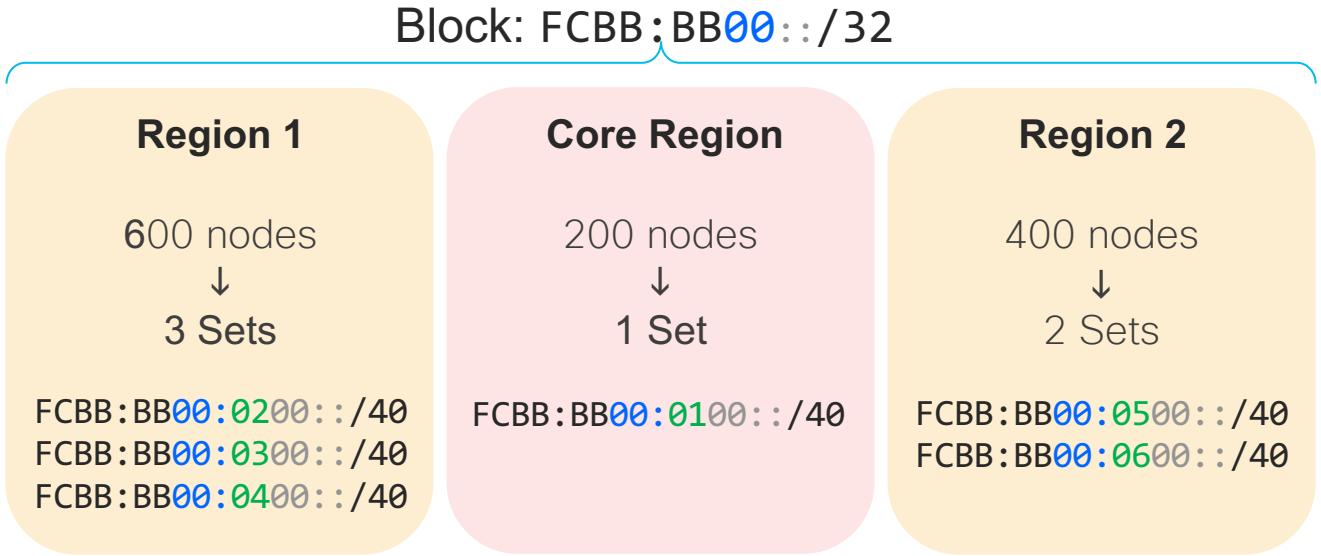
uSID Block per slice (Flex Algo) if possible

- 256 Blocks are available in the SRv6 Space:

  `FCBB:BBTT::/32`, with `TT` = slice ID

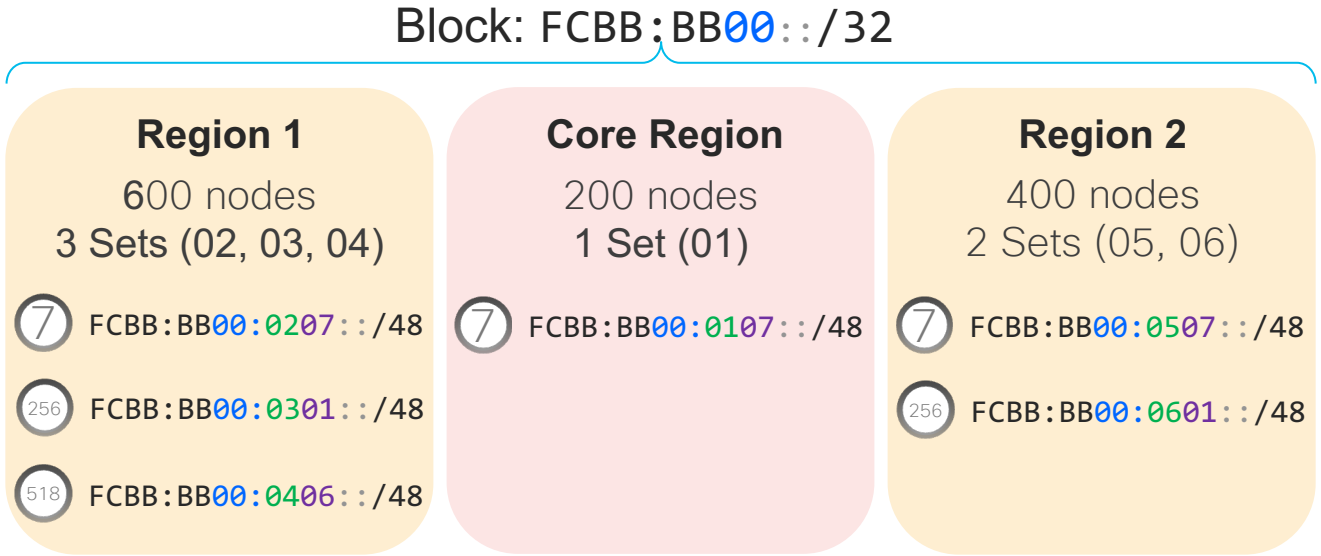
  - Multiple Blocks can be concurrently used on a node
  - 63 Blocks available on DNX1 platforms (`TT` = 00 to 3E)
- We assume 2 slices (Blocks), e.g.:
  - `FCBB:BB00::/32` Low-cost slice (algo 0) ← focus, other Blocks are similar
  - `FCBB:BB01::/32` Low-delay slice (algo 128)

Set Allocation Example



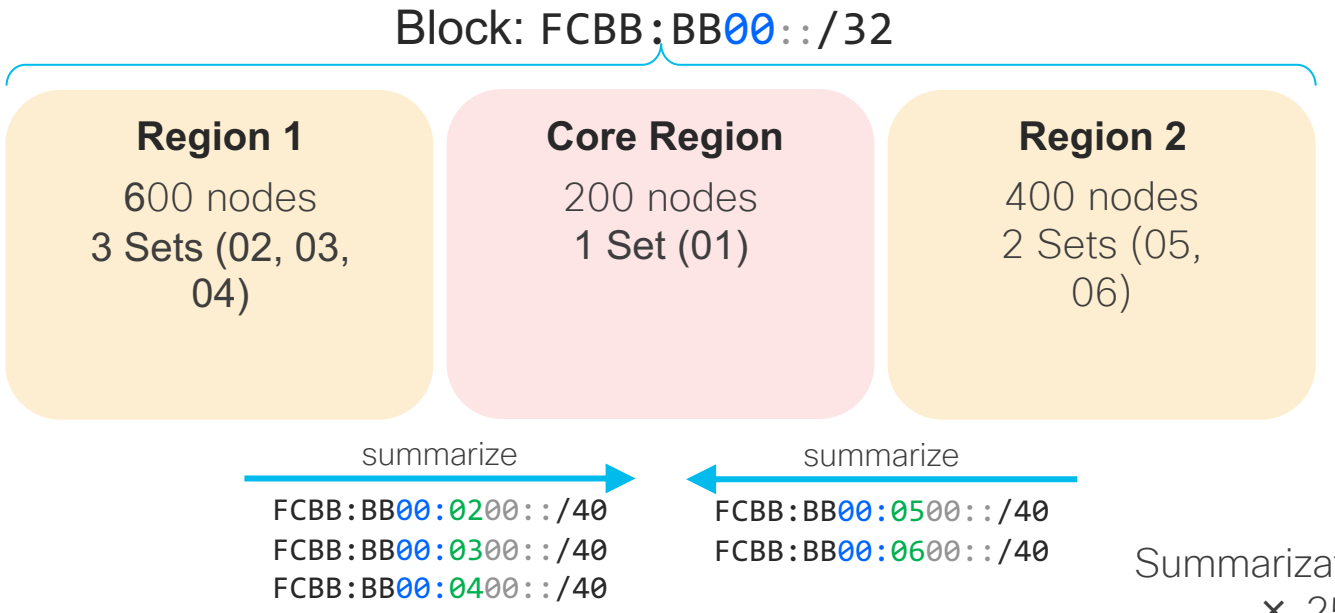
- If a region outgrows its allocated Sets, then allocate more Sets to this region

uSID Allocation Example



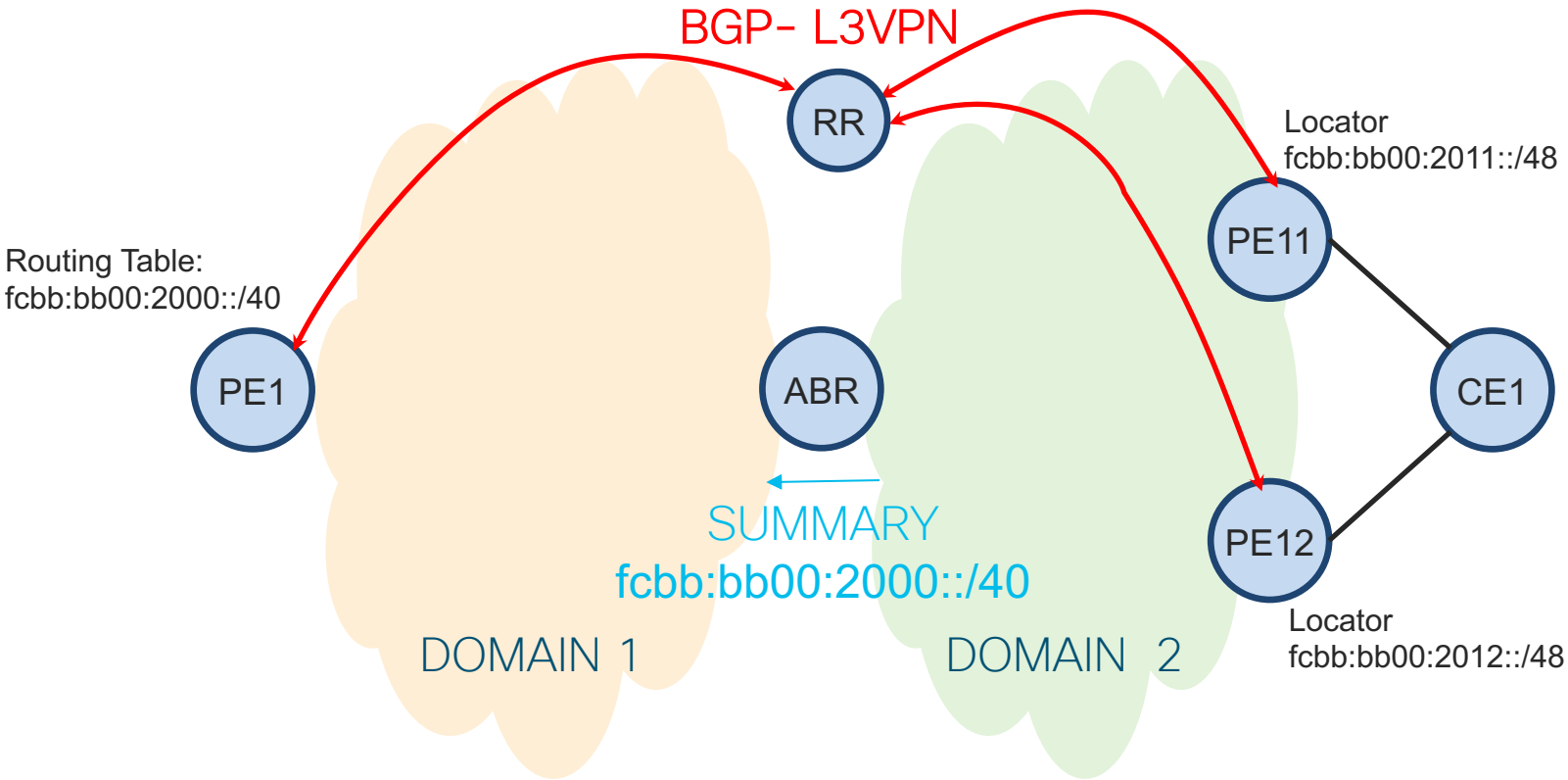
- Remaining unallocated uSIDs in Sets are for future growth

Summarization

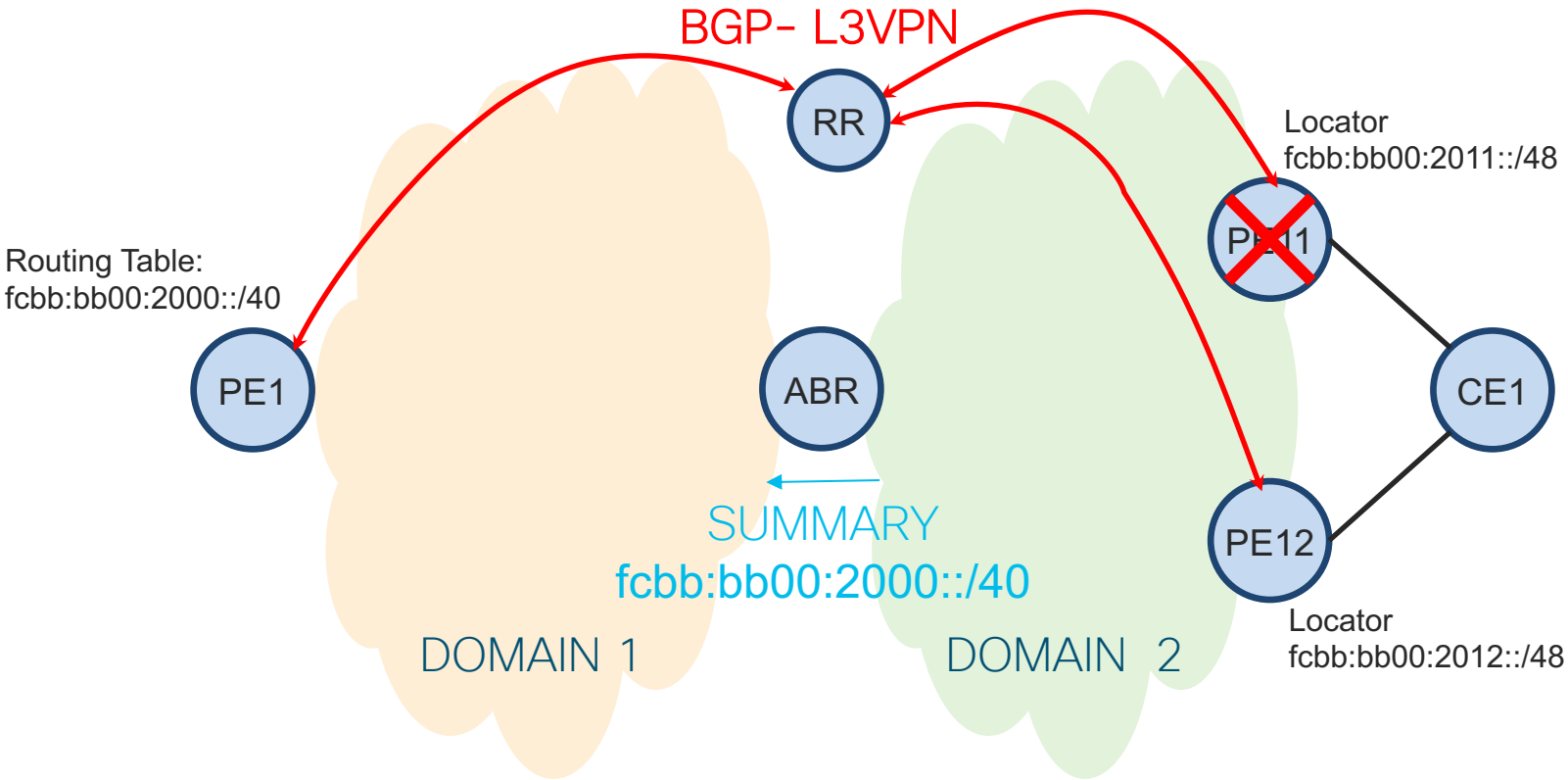


Summarization gain:
× 256

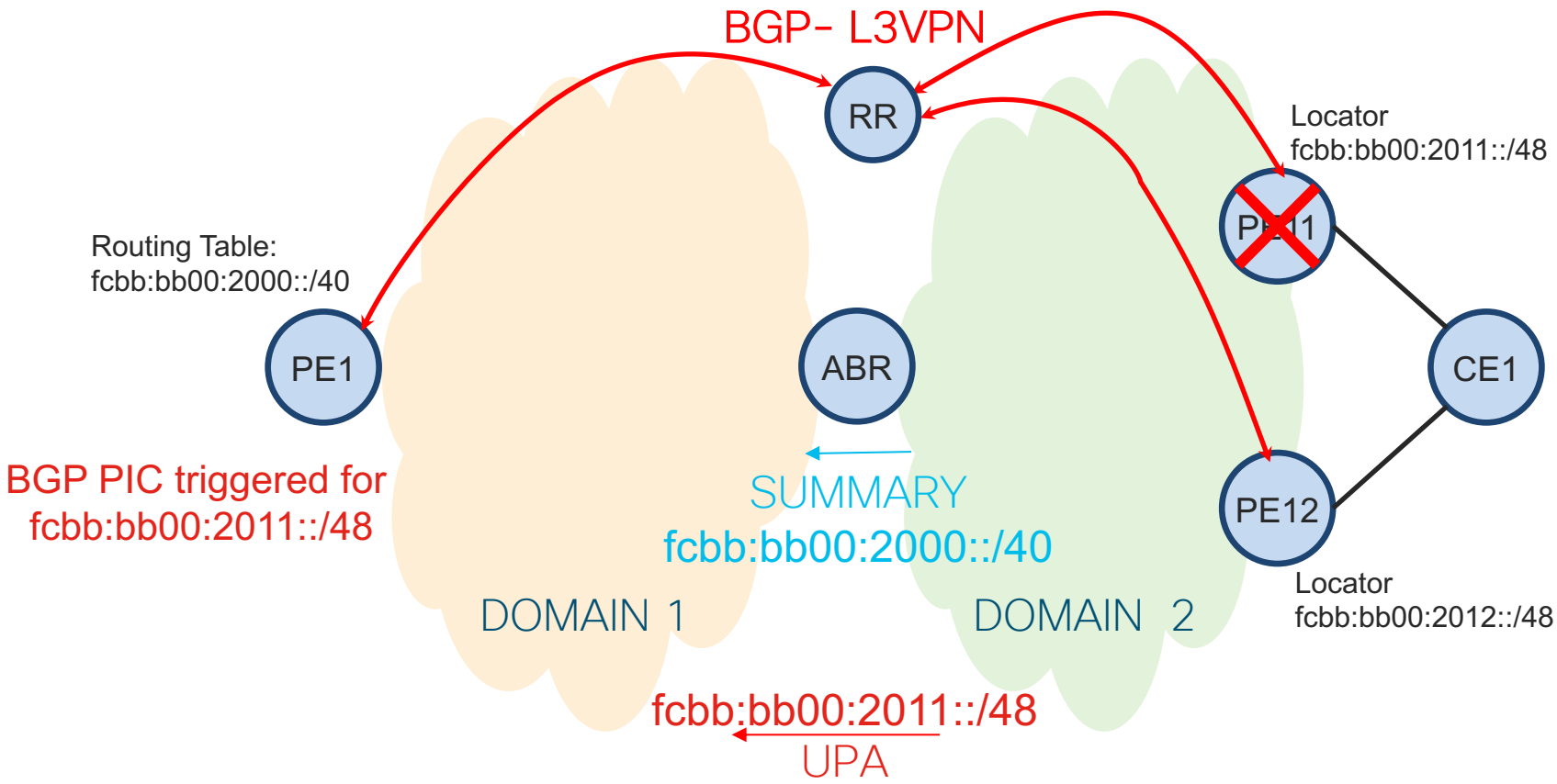
Unreachable Prefix Announcement



Unreachable Prefix Announcement



Unreachable Prefix Announcement



SRv6 Summarization +UPA configuration

ABR:

```
router isis 1
 address-family ipv6 unicast
  summary-prefix fcbb:bb00:2000::/40 adv-unreachable
  summary-prefix fcbb:bb01:2000::/40 algorithm 128 adv-unreachable
```

Summarization for Algorithm 0, per Set /40

UPA

Summarization for Algorithm 128, per Set /40

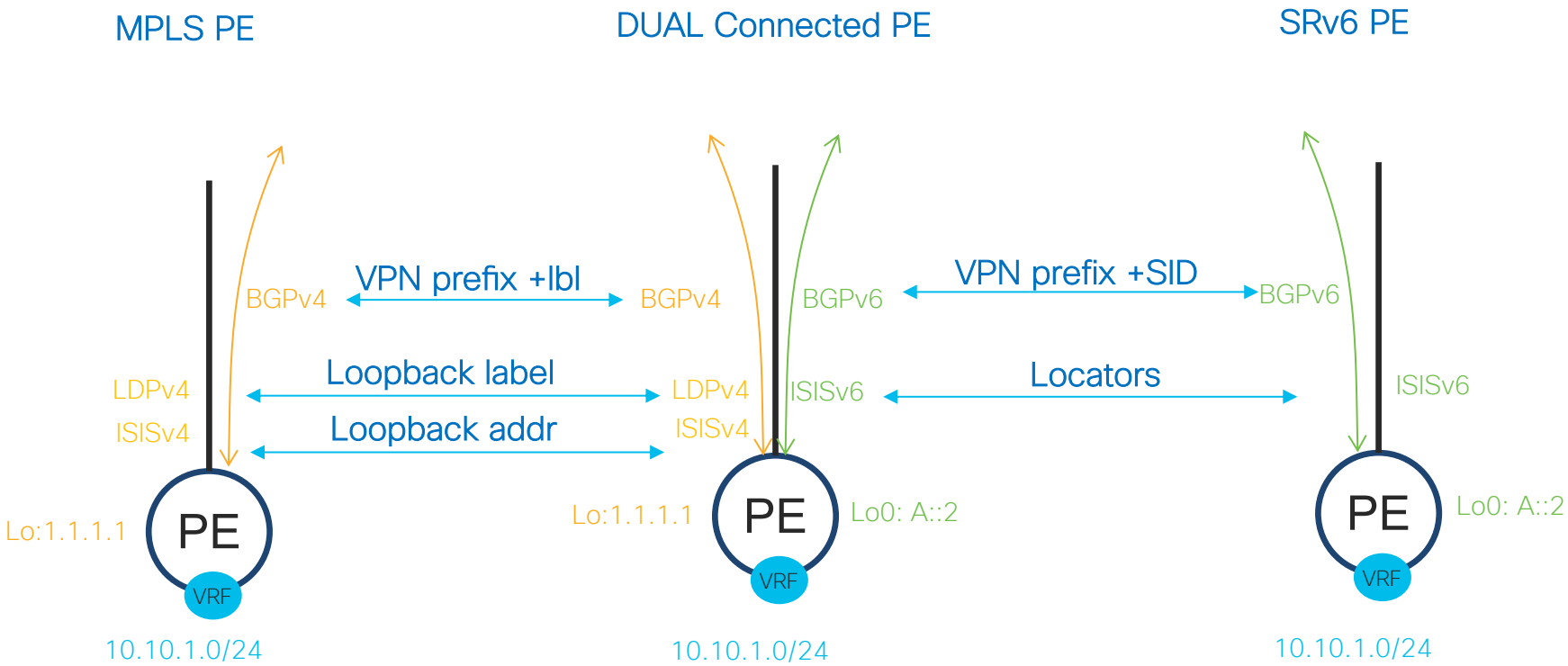
PE:

```
router isis 1
 address-family ipv6 unicast
  prefix-unreachable
   rx-process-enable
```

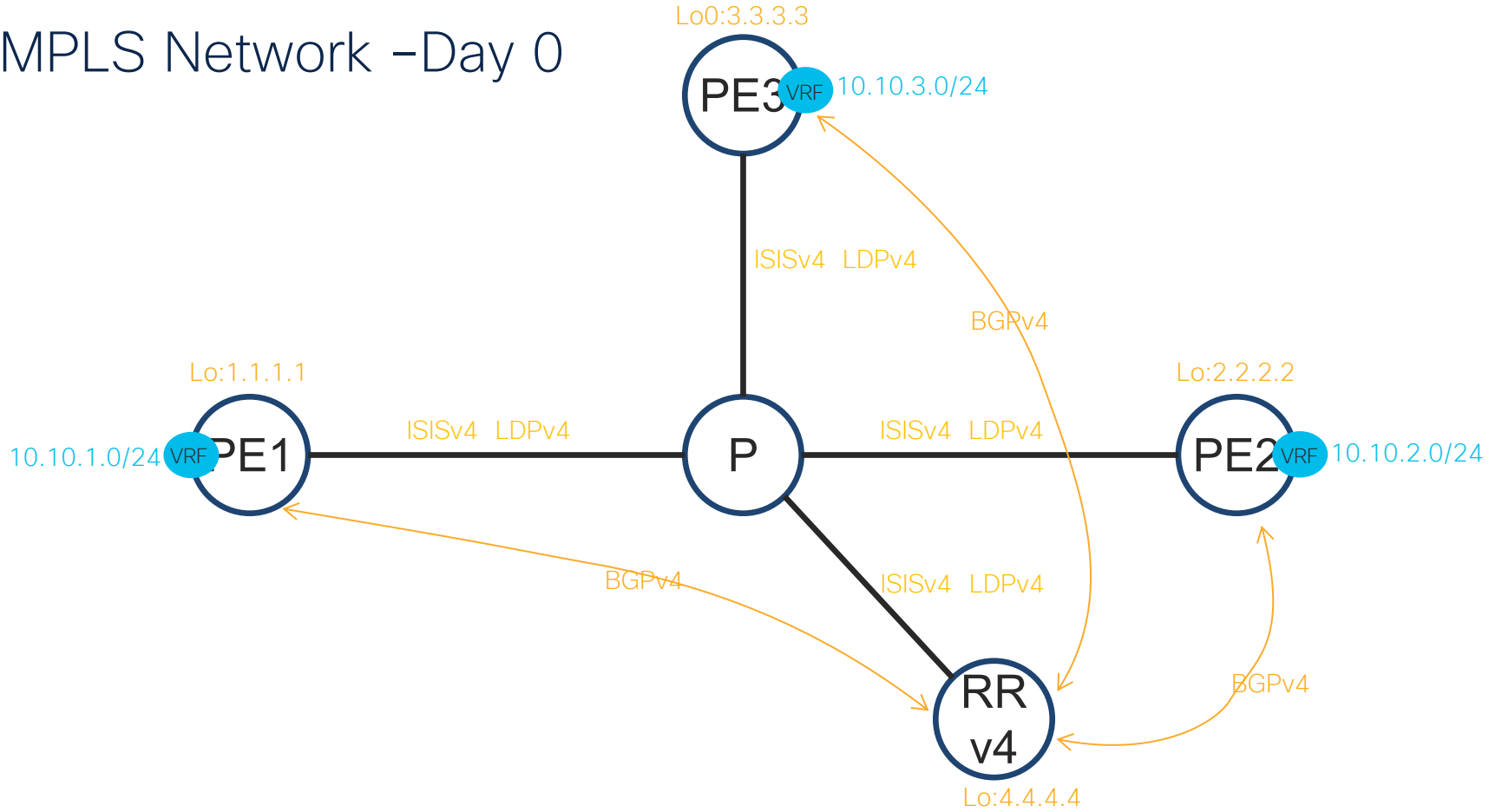
Triggers BGP PIC

SRv6 Migration

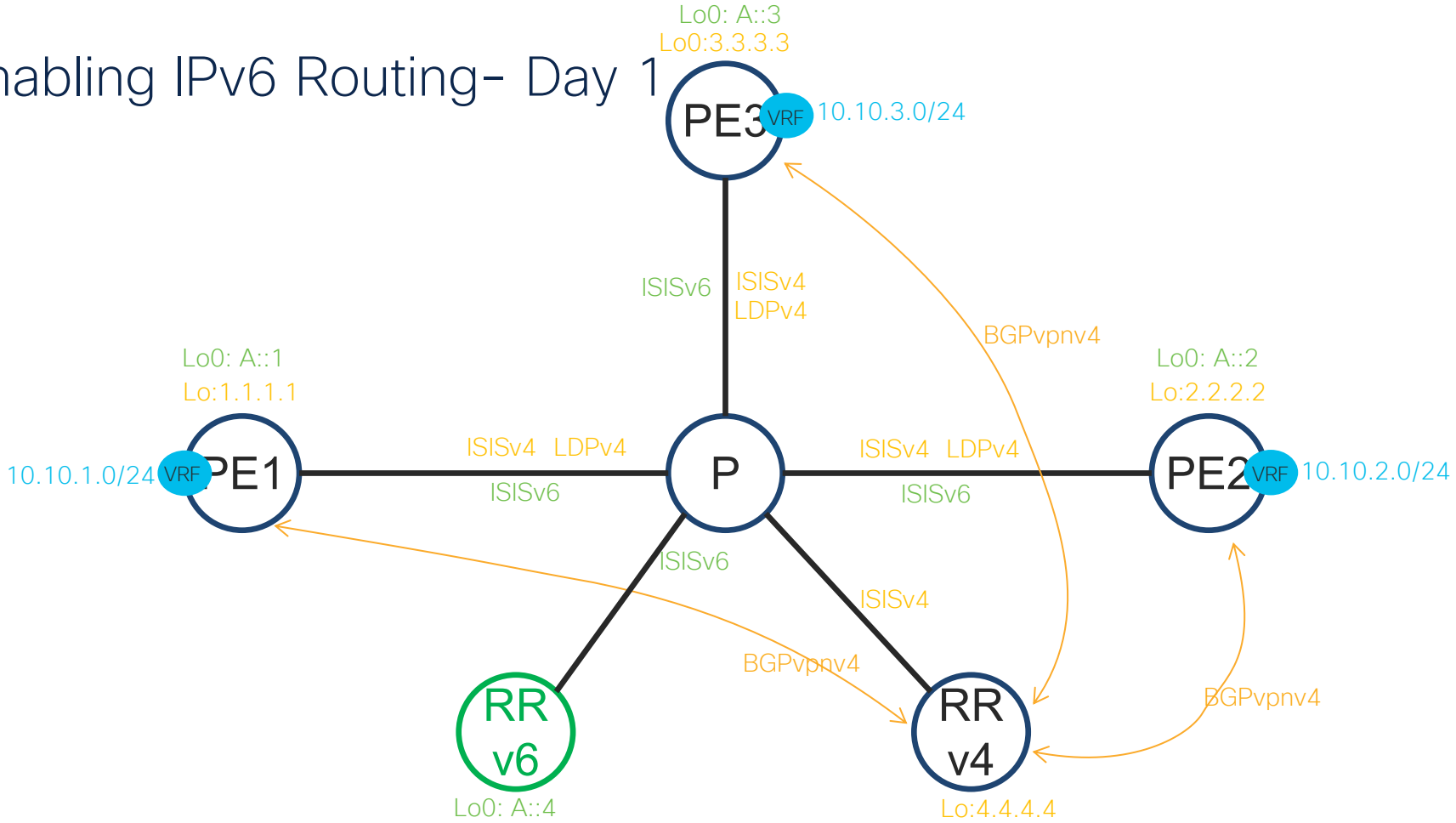
Dual Connected PE



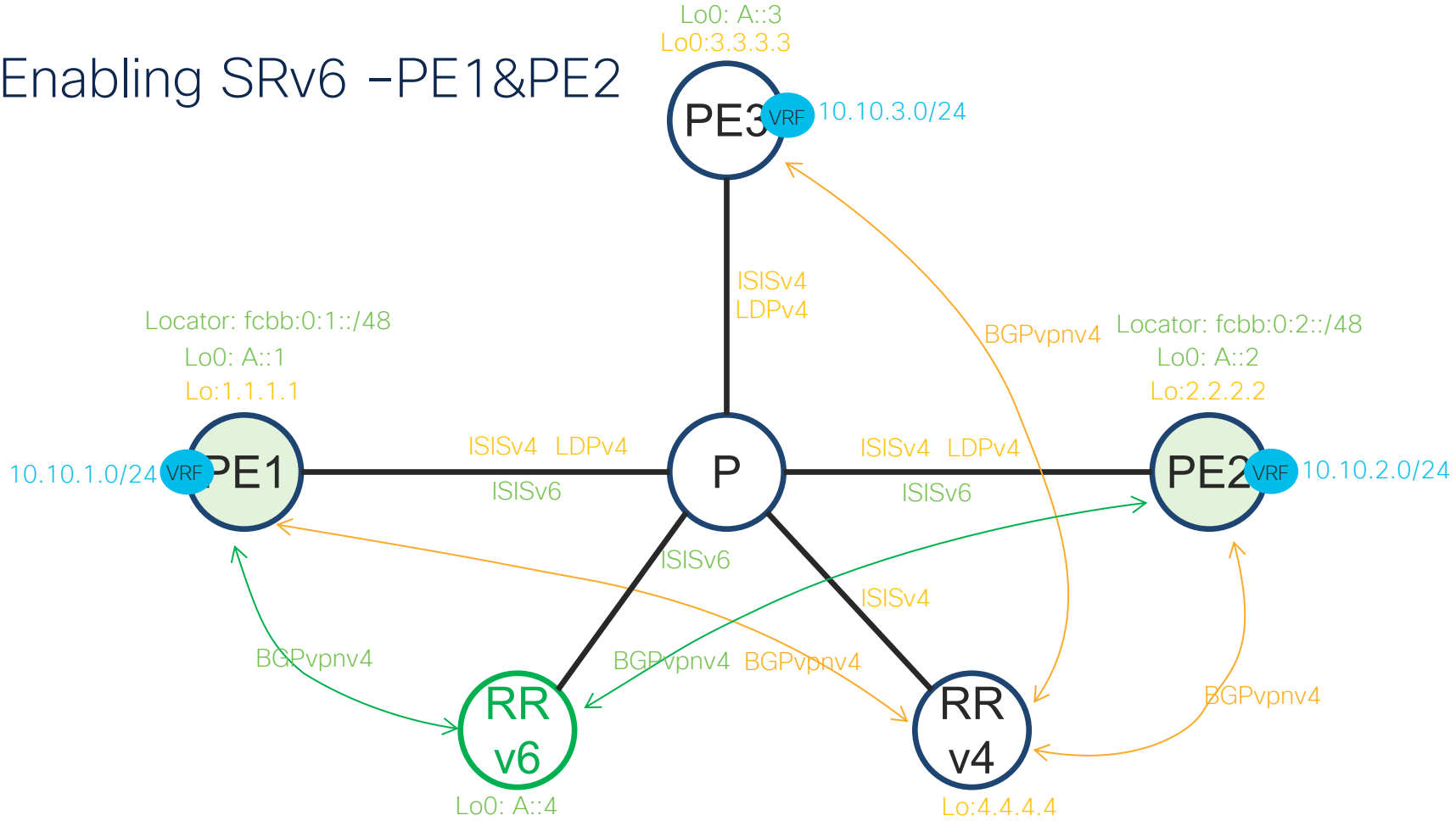
MPLS Network –Day 0



Enabling IPv6 Routing- Day 1

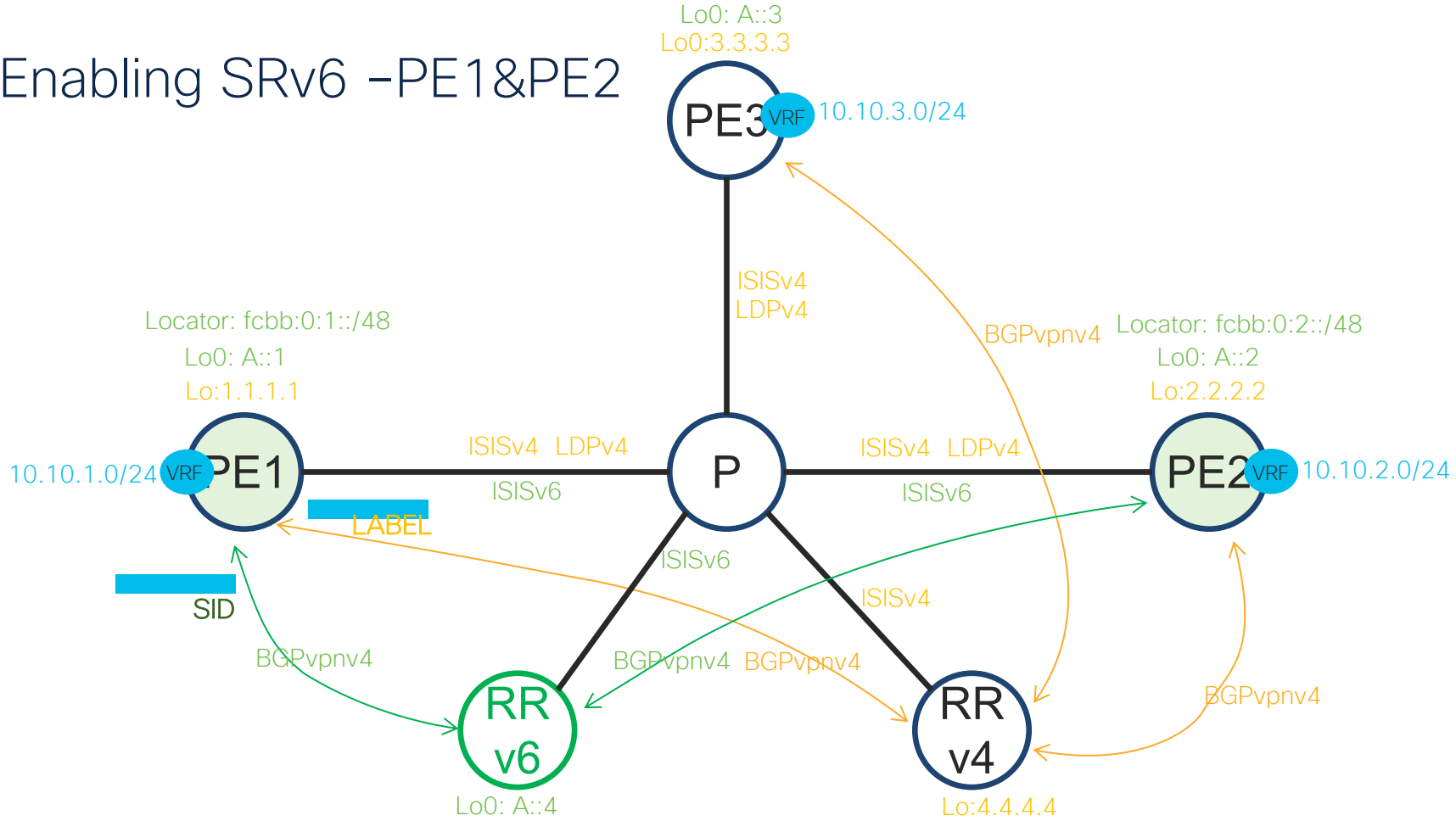


Enabling SRv6 –PE1&PE2



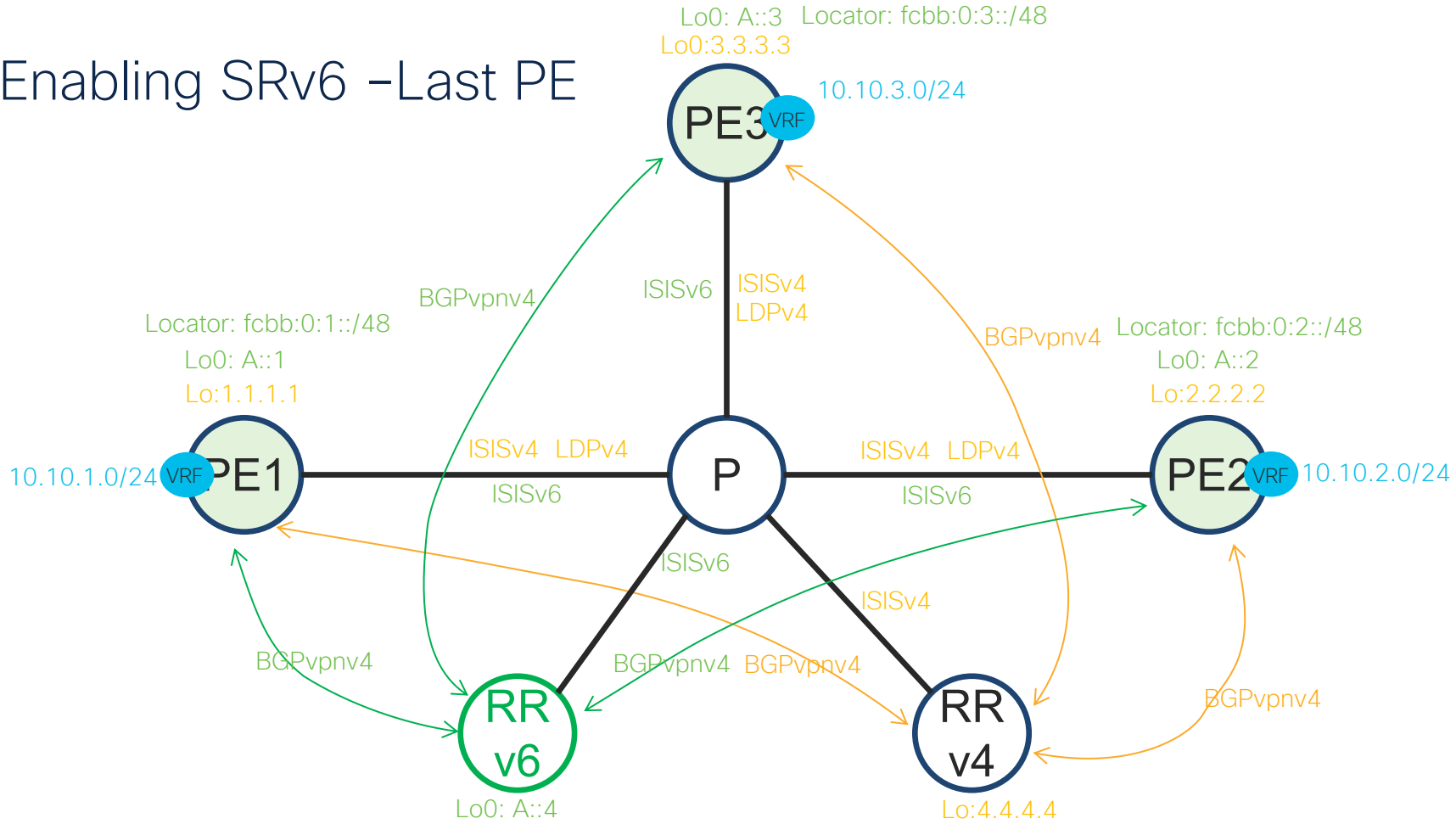
Everything is BGP best path selection driven!! (ie Local Preference)

Enabling SRv6 –PE1&PE2

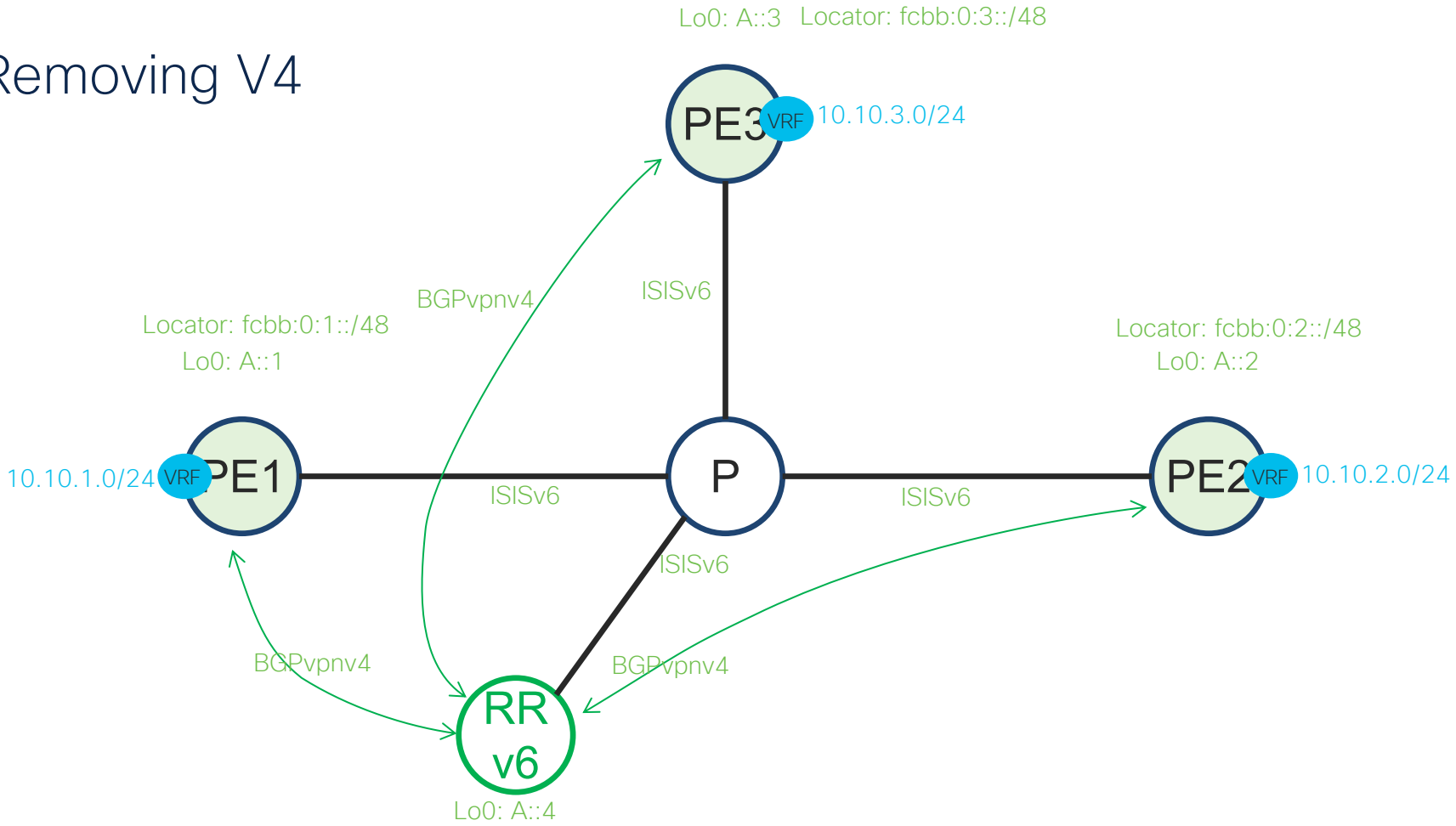


Everything is BGP best path selection driven!! (ie Local Preference)

Enabling SRv6 –Last PE



Removing V4



SRv6 Dual PE Configuration

```
router bgp 1
 neighbor A::4
  address-family vpnv4 unicast
   encapsulation-type srv6
   route-policy RRv6 out
 neighbor 4.4.4.4
  address-family vpnv4 unicast
   route-policy RRv4 out
vrf 1
  address-family ipv4 unicast
   mpls alloc enable
   segment-routing srv6
    locator MAIN
    alloc mode per-vrf
```

Via RPL we set specific BGP attributes to to prefixes ie Local Preference towards RRv6 and RRv4

Policy towards v6 RR

Policy towards v4 RR

Allocates Labels for all prefixes in VRF

Allocates SIDs for all prefixes in VRF from Locator MAIN

SRv6 uSID Conclusion

SRv6 Mature Standardization

- Proposed Standard
  - RFC 8402 SR Architecture
  - RFC 8754 SRv6 DataPlane
  - RFC 8986 SRv6 Network Programming
  - RFC 9256 SR Policy Architecture
  - RFC 9252 SRv6 BGP Extension
  - RFC 9259 SRv6 OAM
  - RFC tba SRv6 ISIS Extension

- WG Document: Proposed Standard
  - draft-ietf-spring-srv6-srh-compression

RFC 8986 — SRv6 Network Programming

RFC 8754 — IPv6 Segment Routing Header

100M *live subscribers* over SRv6

SIMPLICITY ALWAYS PREVAILS

Rich SRv6 uSID Ecosystem

Network Equipment Manufacturers

Cisco, Huawei, Nokia, Ciena, Juniper Networks, ZTE, Arrcus, Arista, H3C

Merchant Silicon

Broadcom, Marvell, Barefoot Networks (Intel)

SR v6
Segment Routing

Open-Source Networking Stacks

Linux, fd.io, P4, OPEN Compute Project, Kubernetes, SONiC, FRRouting

Smart NIC

Intel, Mellanox Technologies

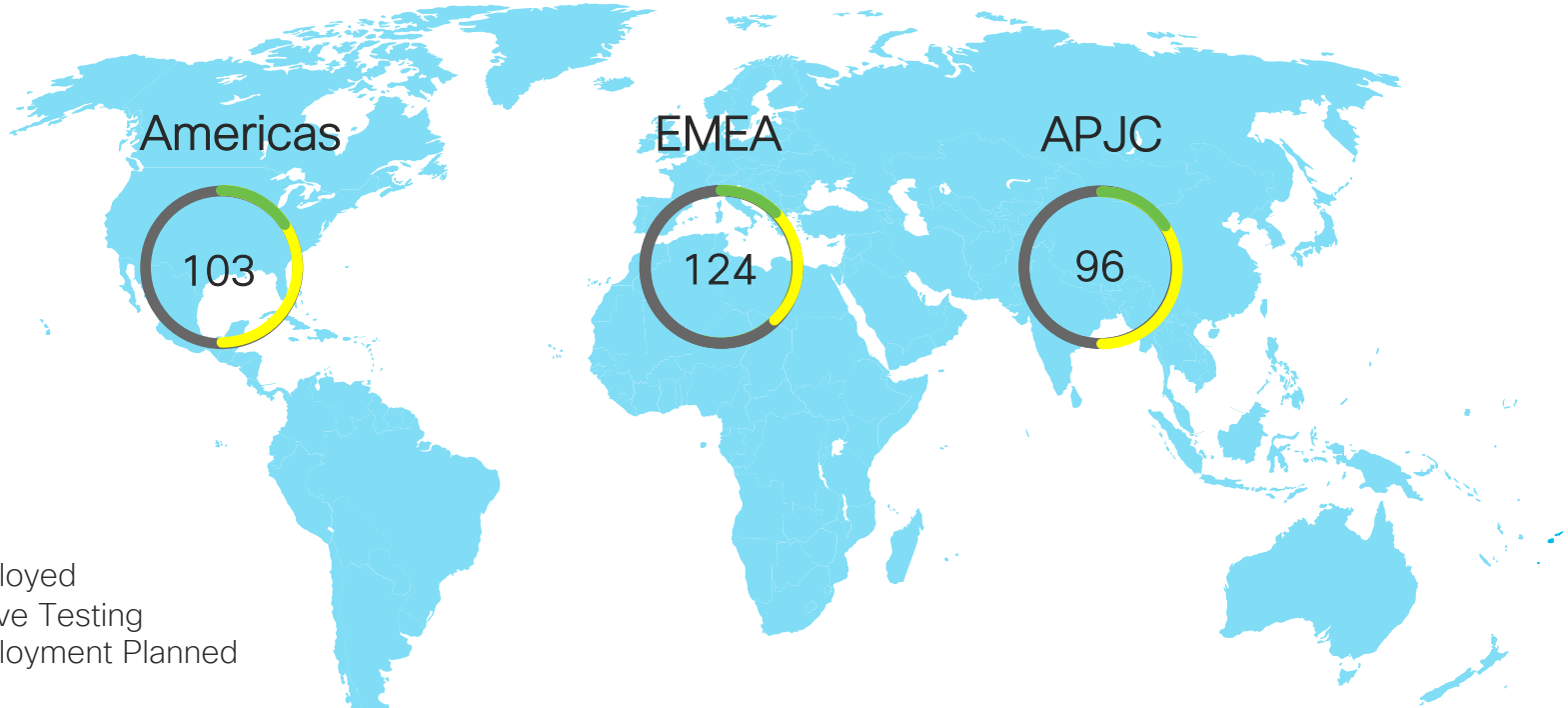
Open-Source Applications

Pyroute2, IPTABLES SERA, GoBGP, exaBGP, eBPF, cilium, TCPDUMP, Wireshark, Snort, GoBMP

Partners

kaloom, Trend Micro, ENEA Qosmos Division, NoviFlow, ixia A Keysight Business, Spirent Promise. Assured

SRv6 … at Record Speed



Simplicity Always Prevails

~~LDP~~

~~RSVP-TE~~

~~BGP 3108~~

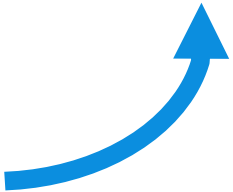
~~MPLS~~

~~UDP/VxLAN~~

~~NSH~~

Furthermore, with more scale

and functionality



The bridge to possible