

The fundamentals of Segment Routing and SRv6

cisco Live !

Leonir Hoxha
Technical Solutions Architect

Session ID: BRKSPG-1214

Cisco Webex App

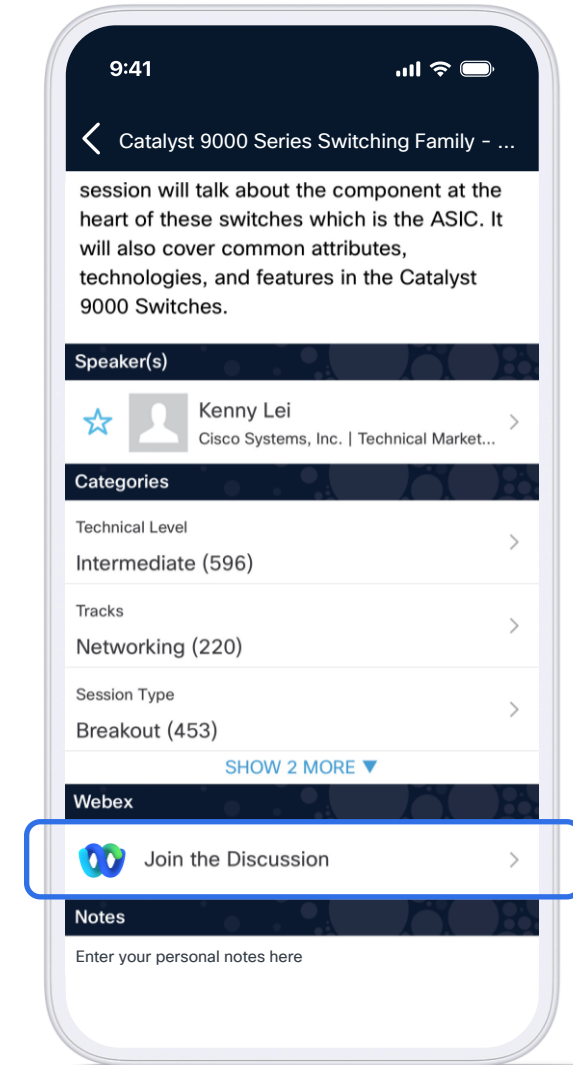
Questions?

Use Cisco Webex App to chat with the speaker after the session

How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until 14 November 2025.



<https://cislive.ciscoevents.com/cislivebot/#BRKSPG-1214>

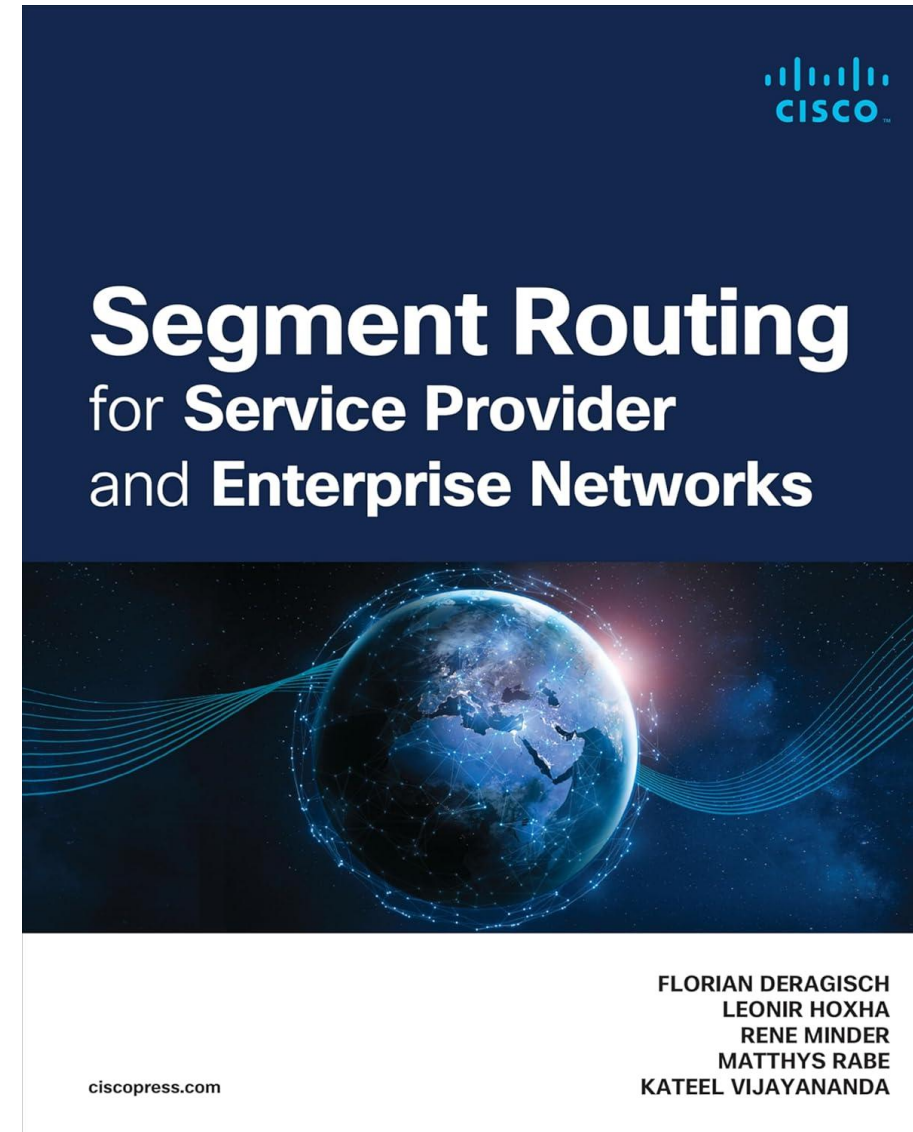
Agenda

- 01 MPLS
- 02 Segment Routing Fundamentals
- 03 Traffic Protection with TI-LFA
- 04 Virtual Networks - FlexAlgo
- 05 SR Traffic Engineering
- 06 SRv6
- 07 SRv6 uSID
- 08 Conclusion

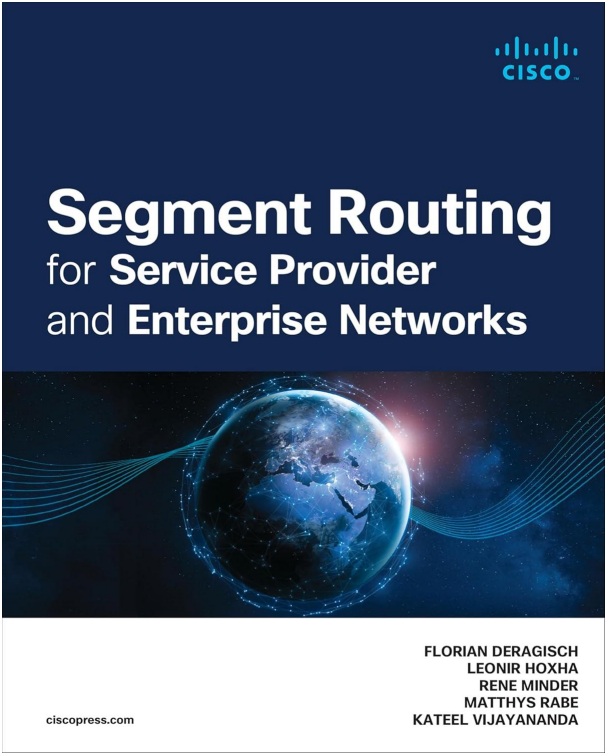
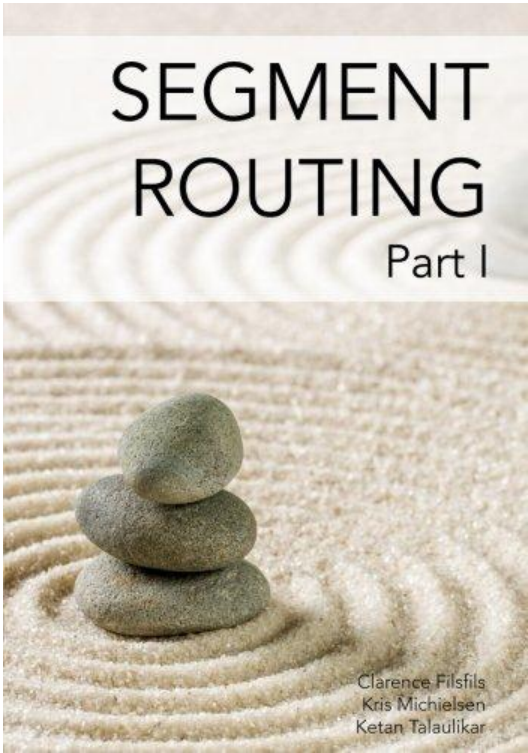
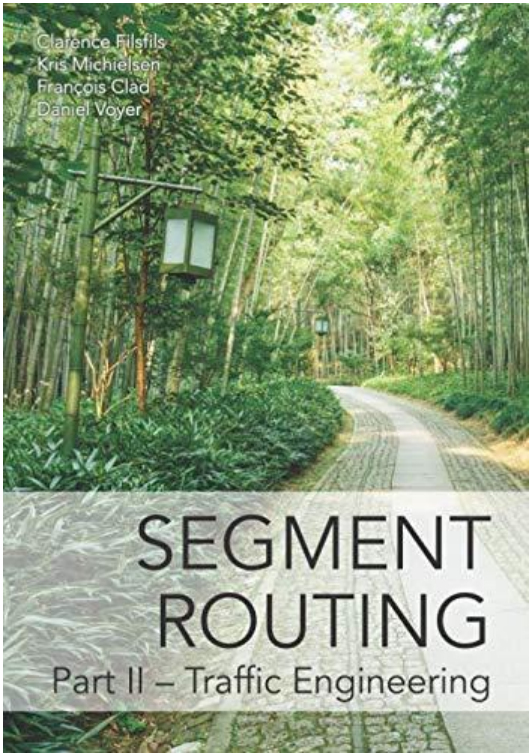
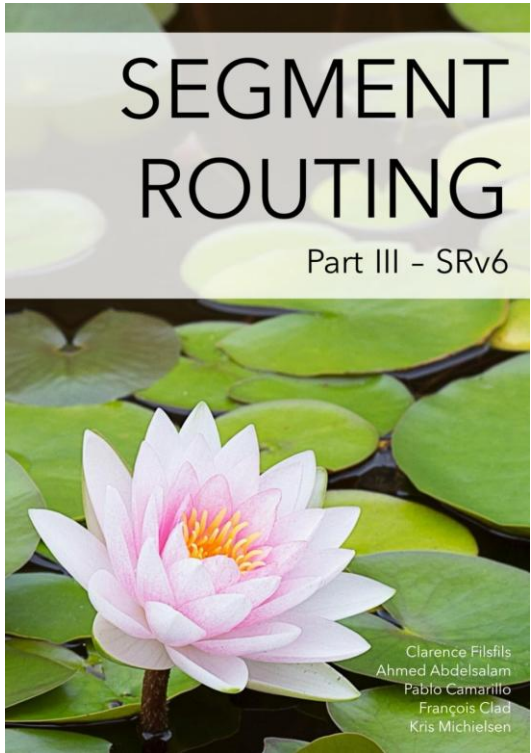
Who am I

Leonir Hoxha

- **Segment Routing Evangelist**
- **Solutions Architect**
- **Conference Speaker**
- **Co-author**



Segment Routing & SRv6 resources



MPLS

Multiprotocol Label Switching (MPLS)

- **1990s** – MPLS was developed in the late 1990s to improve packet forwarding performance and support traffic engineering, originally influenced by Cisco's Tag Switching and other vendor-specific label switching proposals
- **Early 2000s** – Standardization and Adoption: MPLS became standardized by the IETF and was widely adopted in service provider networks for VPNs (L3VPN, L2VPN), fast reroute (FRR), and traffic engineering with RSVP-TE, while remaining complex due to dependency on protocols like LDP, RSVP-TE, and BGP for different functions
- **2013–2014** – Introduction of Segment Routing: Segment Routing (SR) was proposed as a simplification of MPLS, eliminating the need for LDP and RSVP-TE by encoding path information as a list of segments

Segment Routing Fundamentals

Segment Routing Essentials

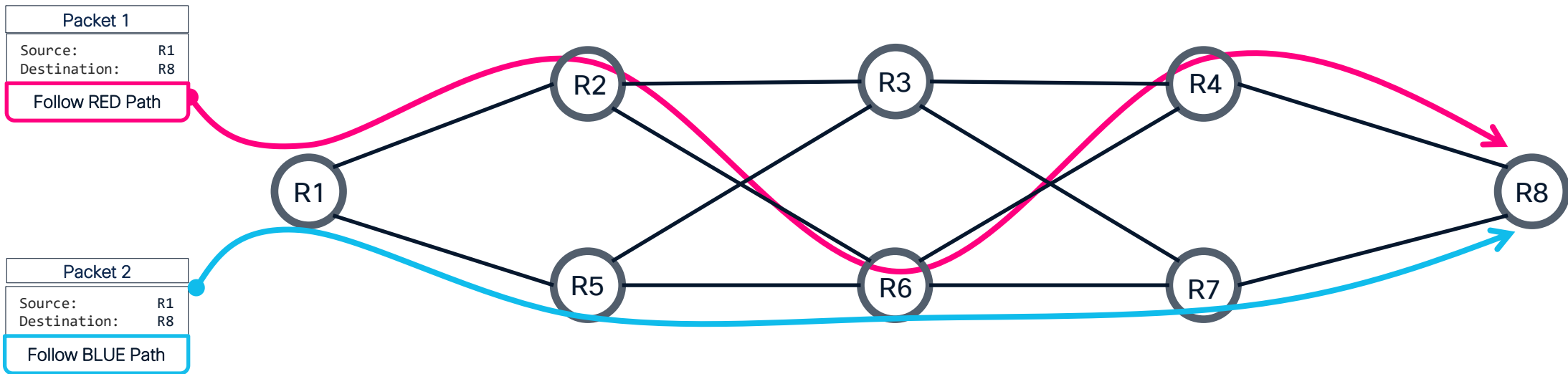
- Segment Routing (SR) is a modern routing architecture that simplifies how traffic is steered through a network by encoding the path as a list of segments, each representing an instruction like “go to this node” or “use this link” directly in the packet header.
- Protocol Simplification: Unlike traditional MPLS, SR removes the need for complex signalling protocols like LDP and RSVP-TE, reducing operational overhead while still enabling advanced use cases such as traffic engineering, fast reroute, and network slicing.
- SR works over:
 - **MPLS** and
 - IPv6 (as **SRv6**)

Segment Routing Essentials

What is a **Segment**

- Basic Definition

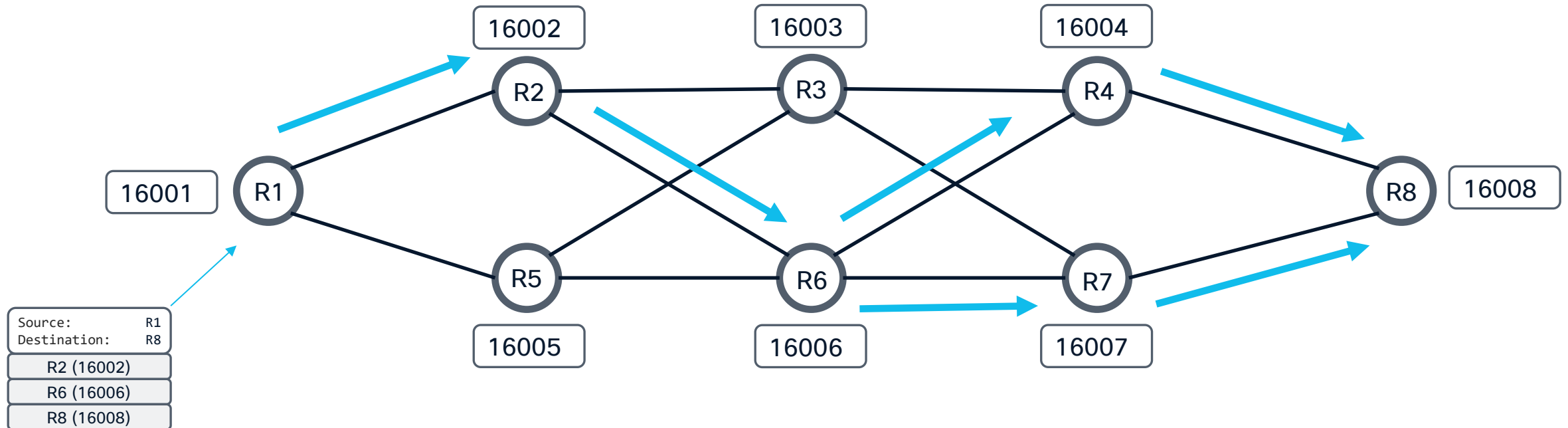
- A **segment** is a single instruction or “waypoint” embedded in a packet that tells the network how to forward it.
- Think of it as an instruction in the header guiding the packet along a specific path or through a specific service.



Segment Routing Essentials

Segment Types (*Node SID*)

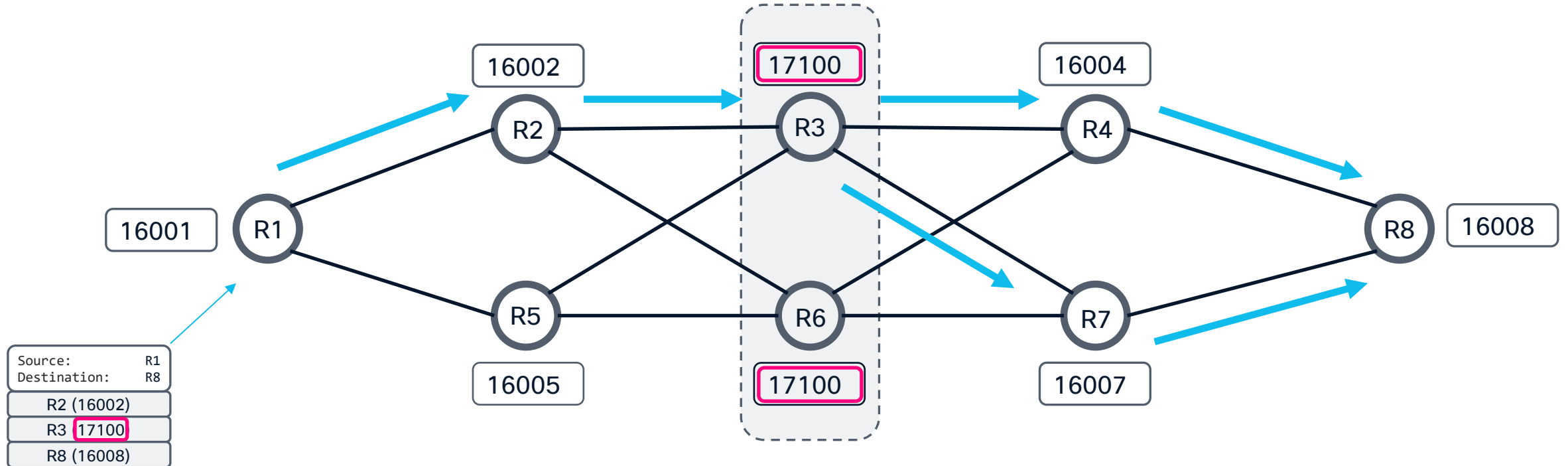
- **Prefix Segment ID** (Node SID):
 - Each node identifies itself in the network with a unique **Node-SID**
 - **ECMP** Load Balancing if multiple equal cost paths exist
 - Global significance within an IGP domain



Segment Routing Essentials

Segment Types (*Anycast SID*)

- **Prefix Segment ID** (Anycast SID):
 - A pair of Nodes share the same **Anycast SID**
 - The nearest Anycast SID is calculated (based on cost)
 - ECMP Load Balancing if multiple equal cost paths exist

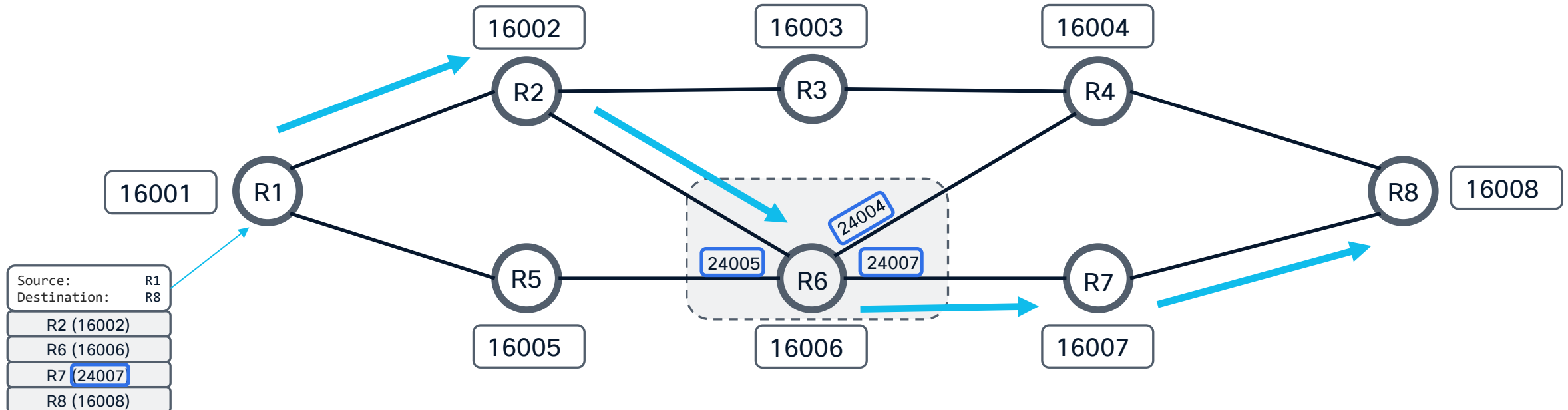


Segment Routing Essentials

Segment Types (*Adjacency SID*)

- **Adjacency Segment ID:**

- Each link of a node is assigned an **Adj-SID**
- Local significance within a node itself
- Advertised but not installed on remote nodes
- By combining **Node SID** and **Adj-SID** in a Label Stack, you are essentially doing SR Traffic Engineering



Where do the SIDs come from

- **Segment Routing Global Block (SRGB)**

- Globally Unique SIDs (Prefix SIDs)
- Range: 16.000 – 23.999

- **Segment Routing Local Block (SRLB)**

- Local Significance only (Adj SIDs)
- Range: 15.000 – 15.999

Segment Routing Essentials

Any other Segment Types

- BGP Prefix SID
 - Forward packet along best-path to reach the prefix associated with the segment
- BGP Peering SID
 - Forward packet over the interface where the neighbor is formed
- Binding SID
 - Forward packet along the path of the associated SR Policy
- Tree SID
 - Replicate / Forward multicast packet to all receivers of the multicast group

Traffic Protection with TI-LFA

Topology Independent LFA (TI-LFA)

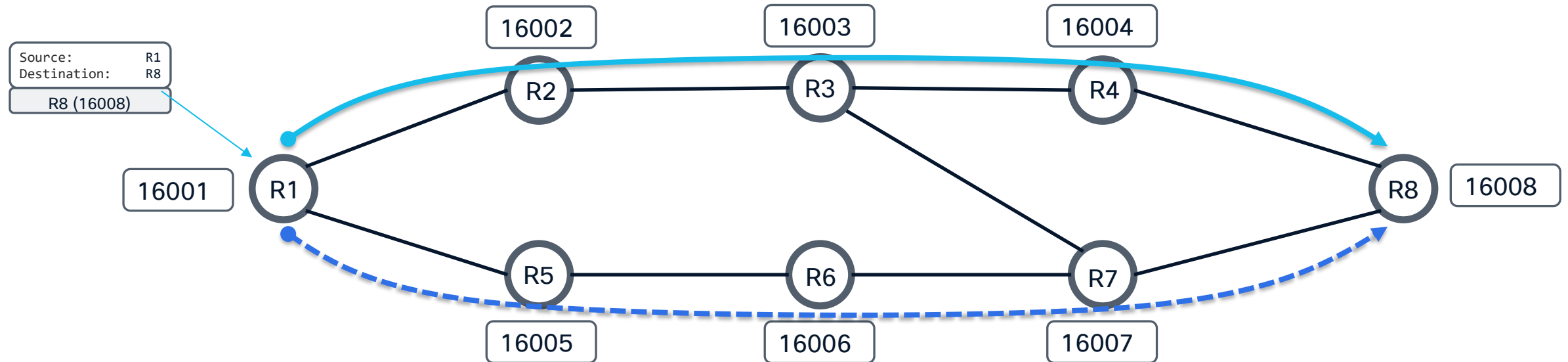
How is my traffic protected

- **100%-coverage** 50-msec link, node, and SRLG protection
- **Simple** to operate and understand
 - Automatically computed by the IGP
 - One configuration line only
- **Prevents** transient **congestion** and suboptimal routing
 - Leverages the post-convergence path, planned to carry the traffic

Topology Independent LFA (TI-LFA)

Traffic Protection in less than 50msec

- For every destination, the IGP is computing the **active** and the **backup** path
- The backup path is the Post Convergence path



Topology Independent LFA (TI-LFA)

Traffic Protection in less than 50msec

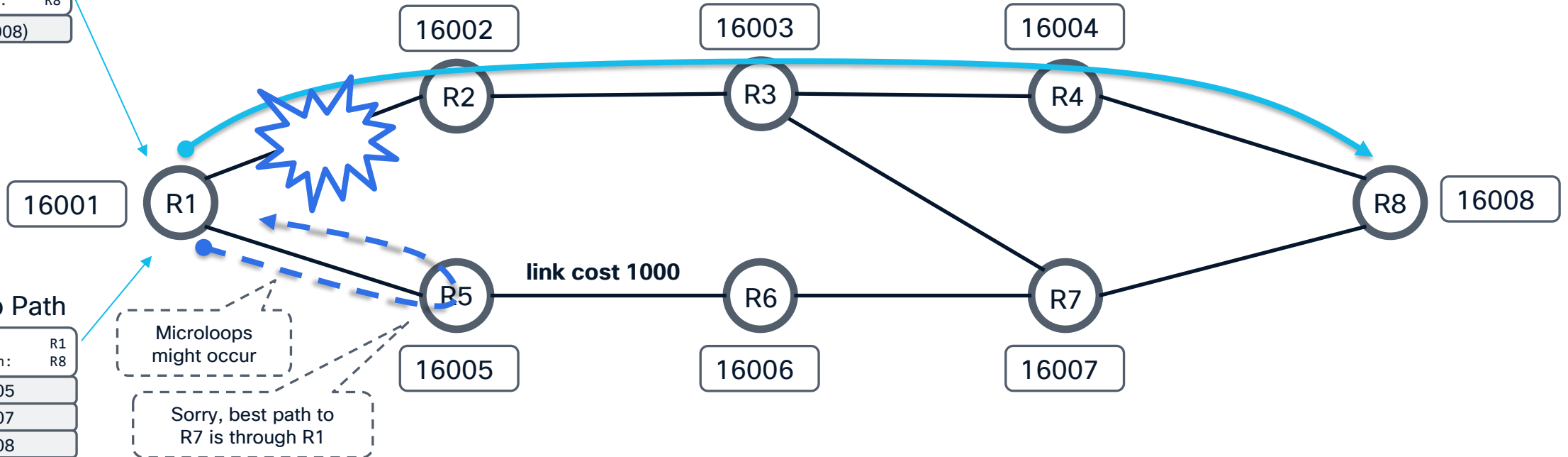
- For every destination, the IGP is computing the **active** and the **backup** path
- The backup path is the Post Convergence path
- Without TI-LFA, Microloops might occur as seen here

Active Path

Source:	R1
Destination:	R8
R8 (16008)	

Backup Path

Source:	R1
Destination:	R8
16005	
16007	
16008	



Topology Independent LFA (TI-LFA)

Traffic Protection in less than 50msec

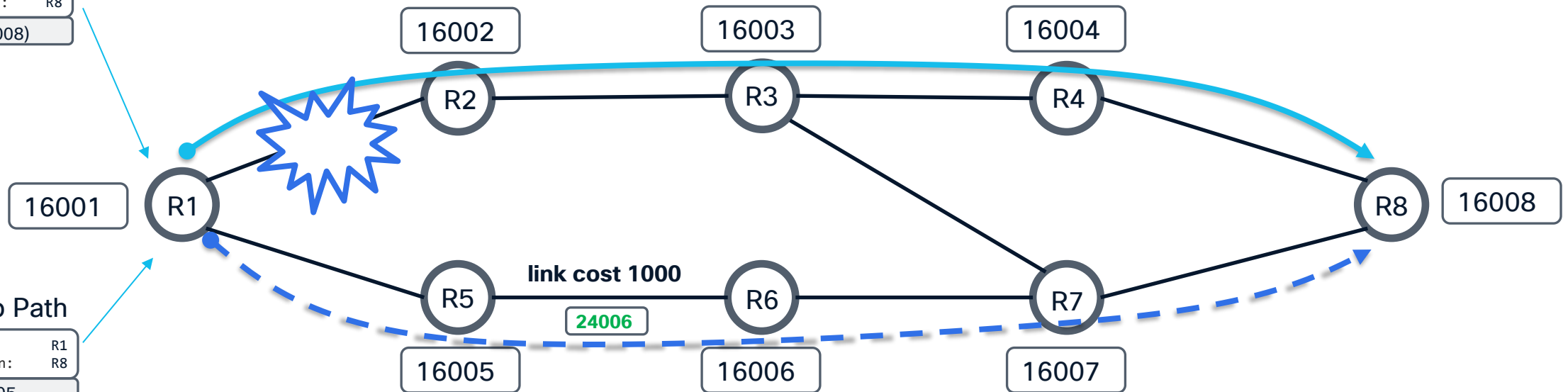
- For every destination, the IGP is computing the **active** and the **backup** path
- The backup path is the Post Convergence path
- TI-LFA and Microloop avoidance in effect (Adj-SID in the Label Stack) as seen here

Active Path

Source:	R1
Destination:	R8
R8 (16008)	

Backup Path

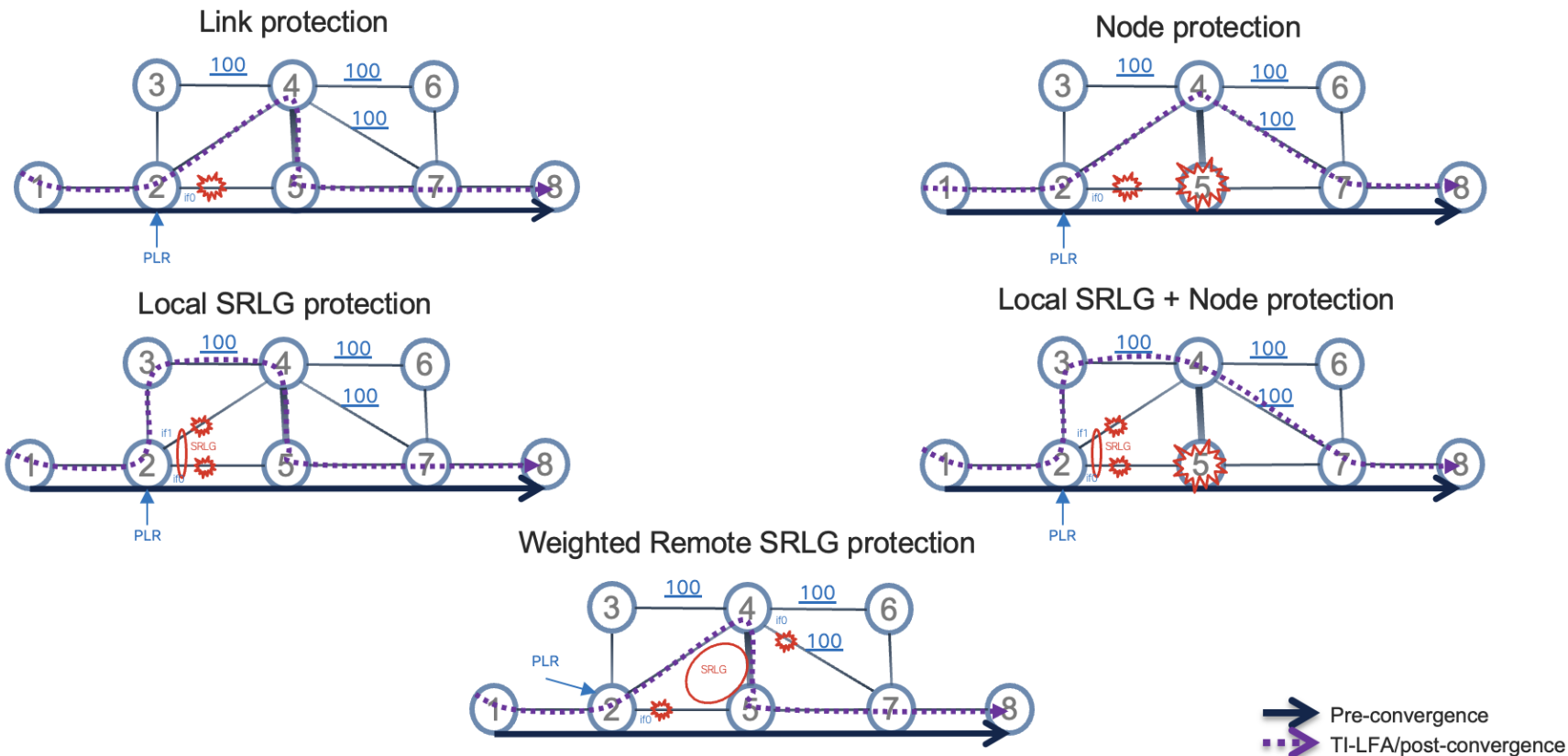
Source:	R1
Destination:	R8
16005	
24006	
16007	
16008	



Topology Independent LFA (TI-LFA)

Link, Node and SRLG Protection

- For every destination, the IGP is computing the **active** and the **backup** path
- Whenever possible the backup path is the post convergence path



Flexible Algorithm

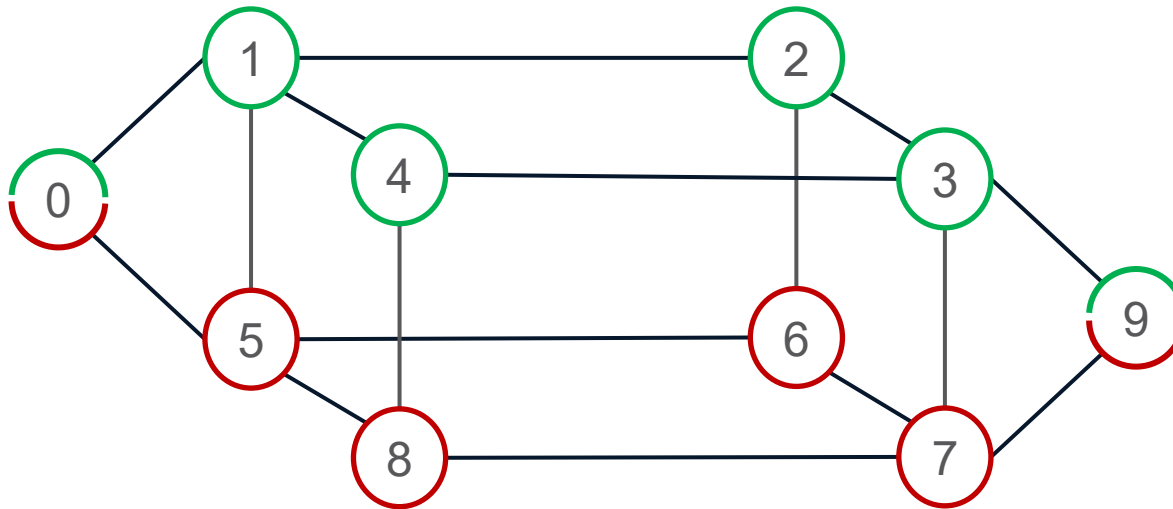
Virtual Topologies with Flexible Algorithm

- Also known as “Network Slicing”
 - 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**
 - Include only Secure (MACSec enabled) Links
- Example
 - Operator1 defines Flex-Algo 128 as “best IGP cost path and avoid link-affinity **green**”
 - Operator2 defines Flex-Algo 128 as “lowest delay path and avoid link-affinity **blue**”

Flexible Algorithm

Flex-Algo definition

- Flex-Algo values: 128 - 255
- All the nodes support Algo 0: minimize IGP metric
- Green nodes also support 128: Secure (MACSec enabled) links
- Red nodes also support 129: minimize Delay



Loopback: 1.1.1.9/32

16009 Algo 0

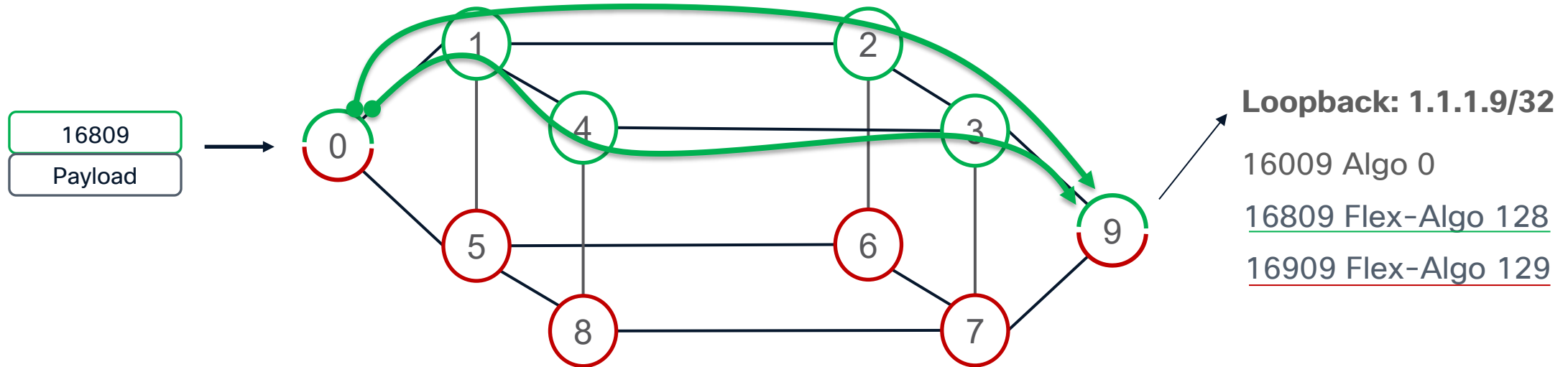
16809 Flex-Algo 128

16909 Flex-Algo 129

Flexible Algorithm

Green Plane

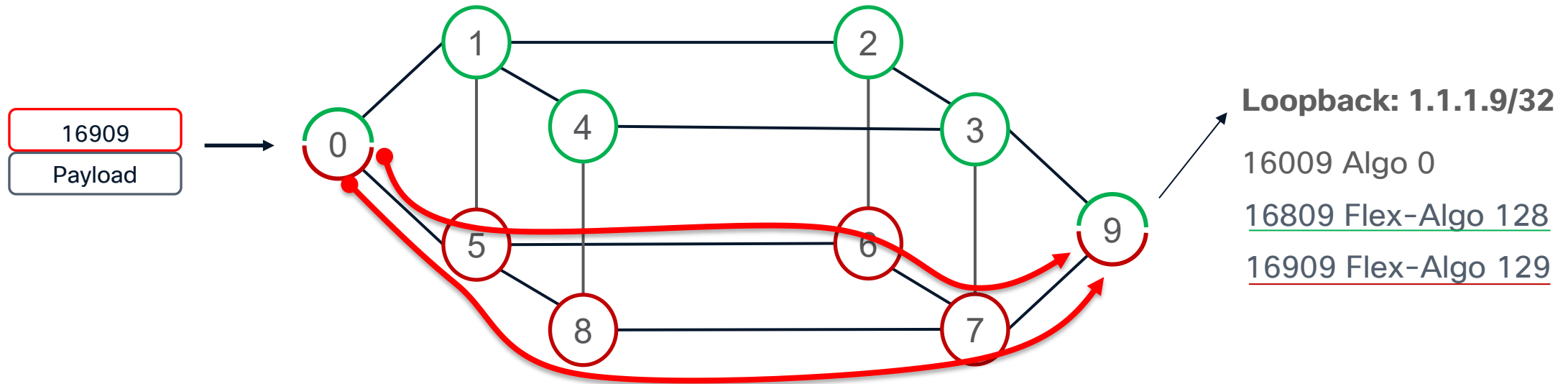
- Flex-Algo values: 128 - 255
- All the nodes support Algo 0: minimize IGP metric
- Green nodes also support 128: Secure (MACSec enabled) links
- Red nodes also support 129: minimize Delay



Flexible Algorithm

Red Plane

- Flex-Algo values: 128 - 255
- All the nodes support Algo 0: minimize IGP metric
- Green nodes also support 128: Secure (MACSec enabled) links
- Red nodes also support 129: minimize Delay



Flexible Algorithm

Minimize label stack, and maximize ECMP at midpoints:

- Single SID needed to enforce traffic towards a specific Path/Plane
- ECMP if possible, on that specific Path/Plane

Slicing of the network on a per:

- Latency
- Bandwidth
- Secure links – MACSec

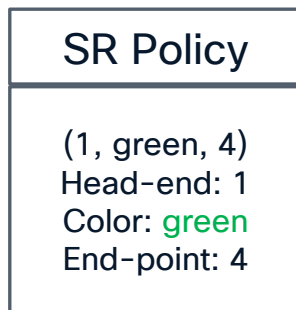
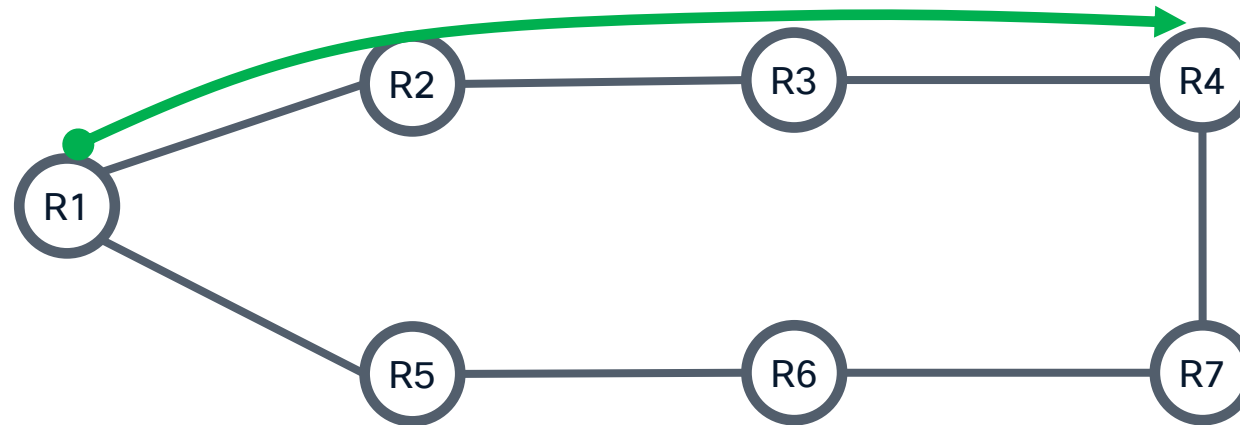
TI-LFA aware

- Protected path stays in Flex-Algo virtual topology

Segment Routing Traffic Engineering

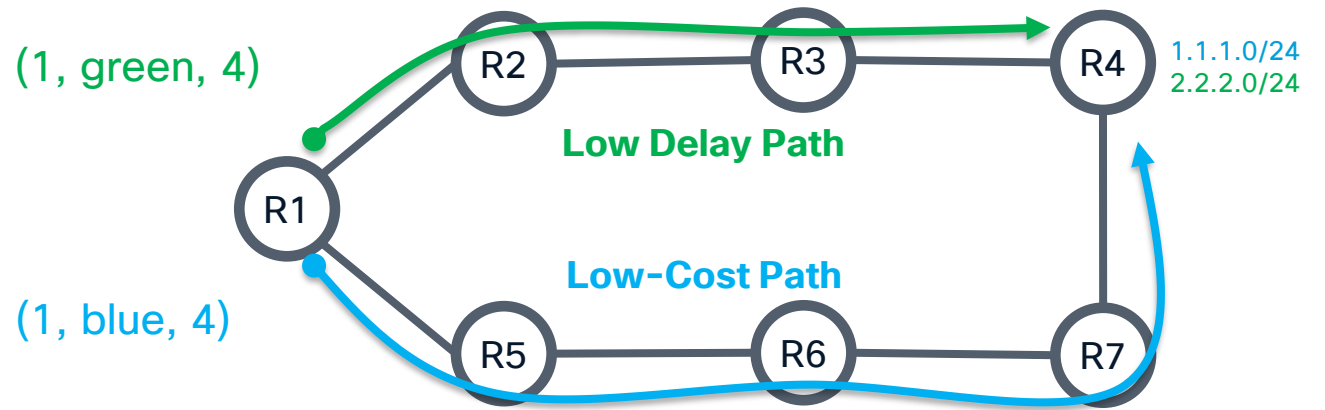
SR-TE (SR Policy)

- The RSVP-TE tunnel Interface construct has been replaced
- SR-Policy is a Traffic Engineering intent by means of a solution SID-list.
- An SR-Policy is uniquely identified by a tuple:
 - Head-end: where the SR Policy has been instantiated (configured)
 - End-point: the destination of the SR Policy
 - Color: a numerical value to differentiate multiple SRTE Policies between the same pair of nodes



SR Policy

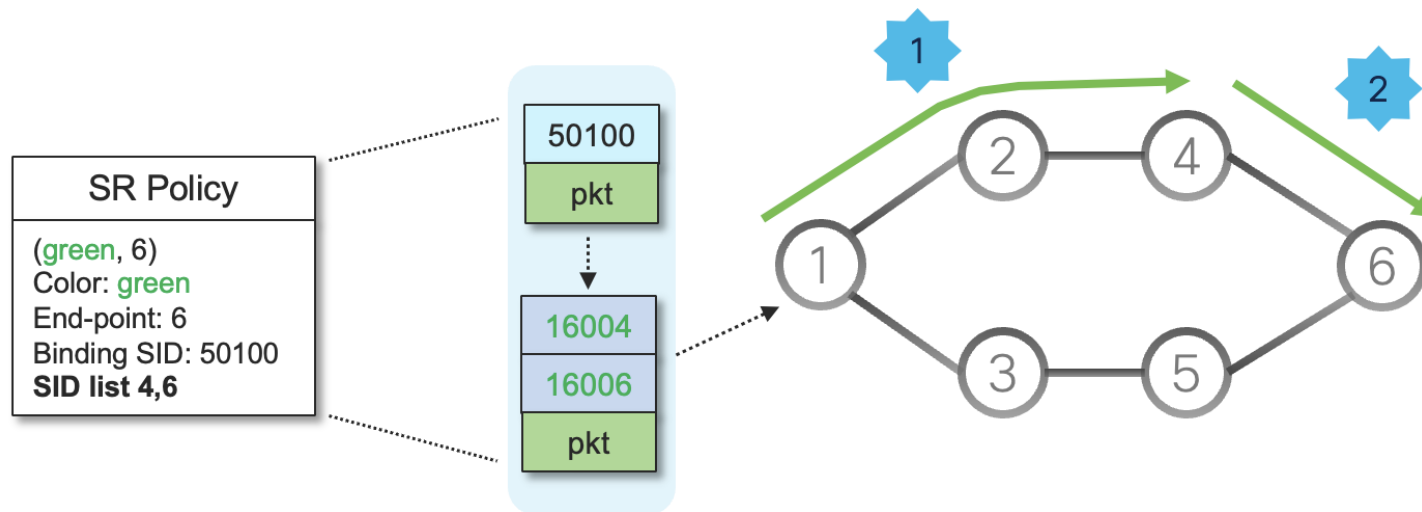
- Each SR Policy has a color
 - Color can be used to indicate a certain treatment (SLA, policy) provided by an SR Policy
- Only one SR Policy with a given color C can exist between a given node pair (head-end (H), end-point (E))
 - In other words: each SR Policy triplet (H, C, E) is unique
- Example:
 - Low-cost=“blue”, Low-delay=“green”
 - steer traffic to 1.1.1.0/24 via Node4 into Low-cost SR Policy (1, blue, 4)
 - steer traffic to 2.2.2.0/24 via Node4 into Low-delay SR Policy (1, green, 4)



Binding SID

What is a Binding SID

- The **Binding SID** is a fundamental building block of SR-TE solution
- Explicitly configured or dynamically allocated
- A Binding SID identifies a SRTE Policy
 - Packet received with Binding-SID as Top Label is steered into the SRTE Policy associated with the Binding-SID
 - Binding-SID label is popped, SRTE Policy's SID list is pushed



SR-TE Solution

Traffic Steering and Network Controller

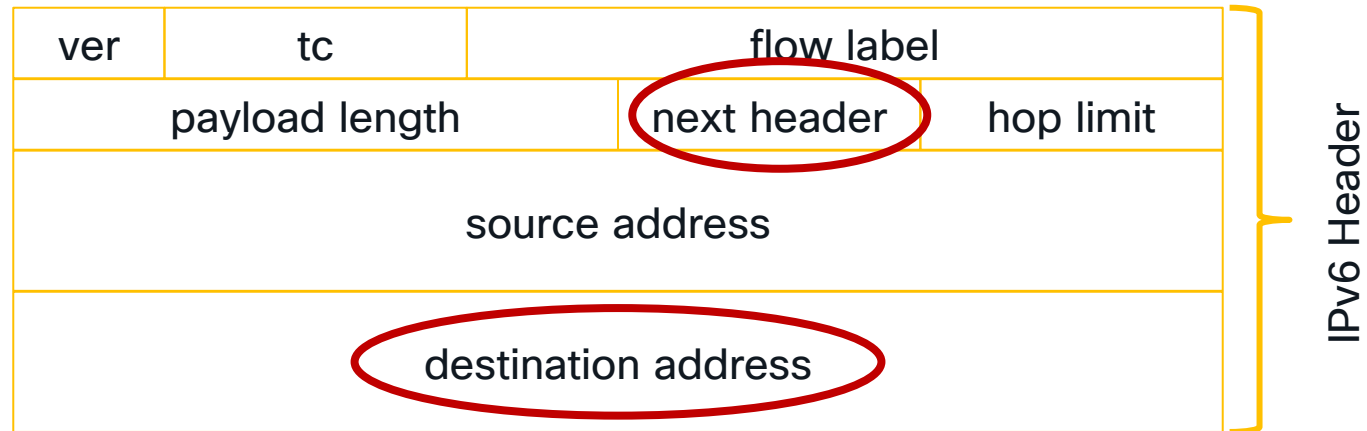
- Automated Steering
 - When a BGP **next-hop** and **color** (BGP community) of a route match the **end-point** and **color** of an SR Policy, then BGP installs the route resolving on the BSID of the SR Policy
- On-Demand Next-Hop (ODN)
 - A head-end router automatically instantiates an SR Policy to a BGP next-hop when required (on-demand)
- SR-PCE
 - Multi-domain stateful SR Path Computation Element (PCE), uses PCEP and BGP-LS to instruct the Head-end Routers about Path Instantiation

SRv6 in a Nutshell

IPv6 Header

Two important fields:

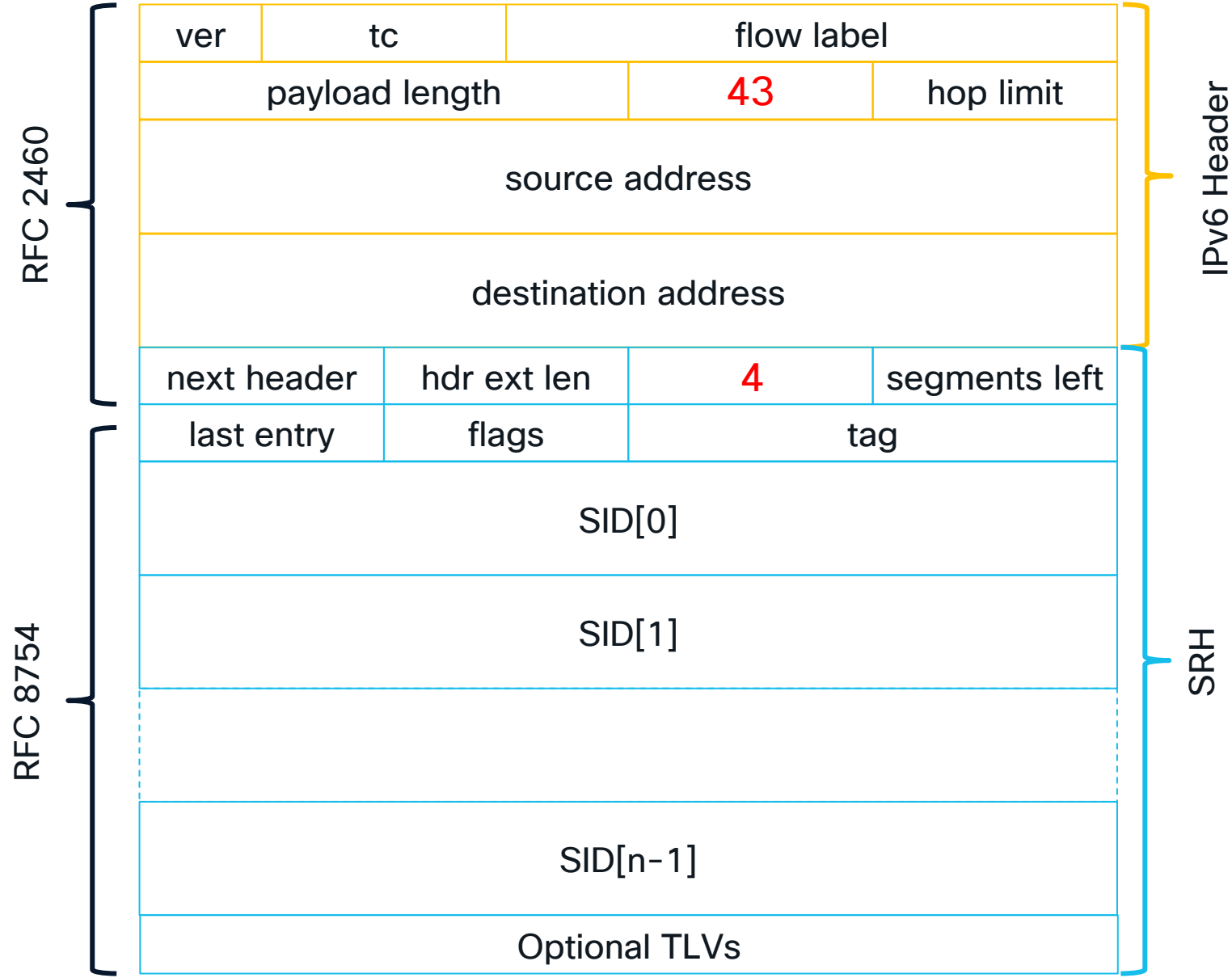
- Next header
- Destination address



Segment Routing Header

RFC 8754

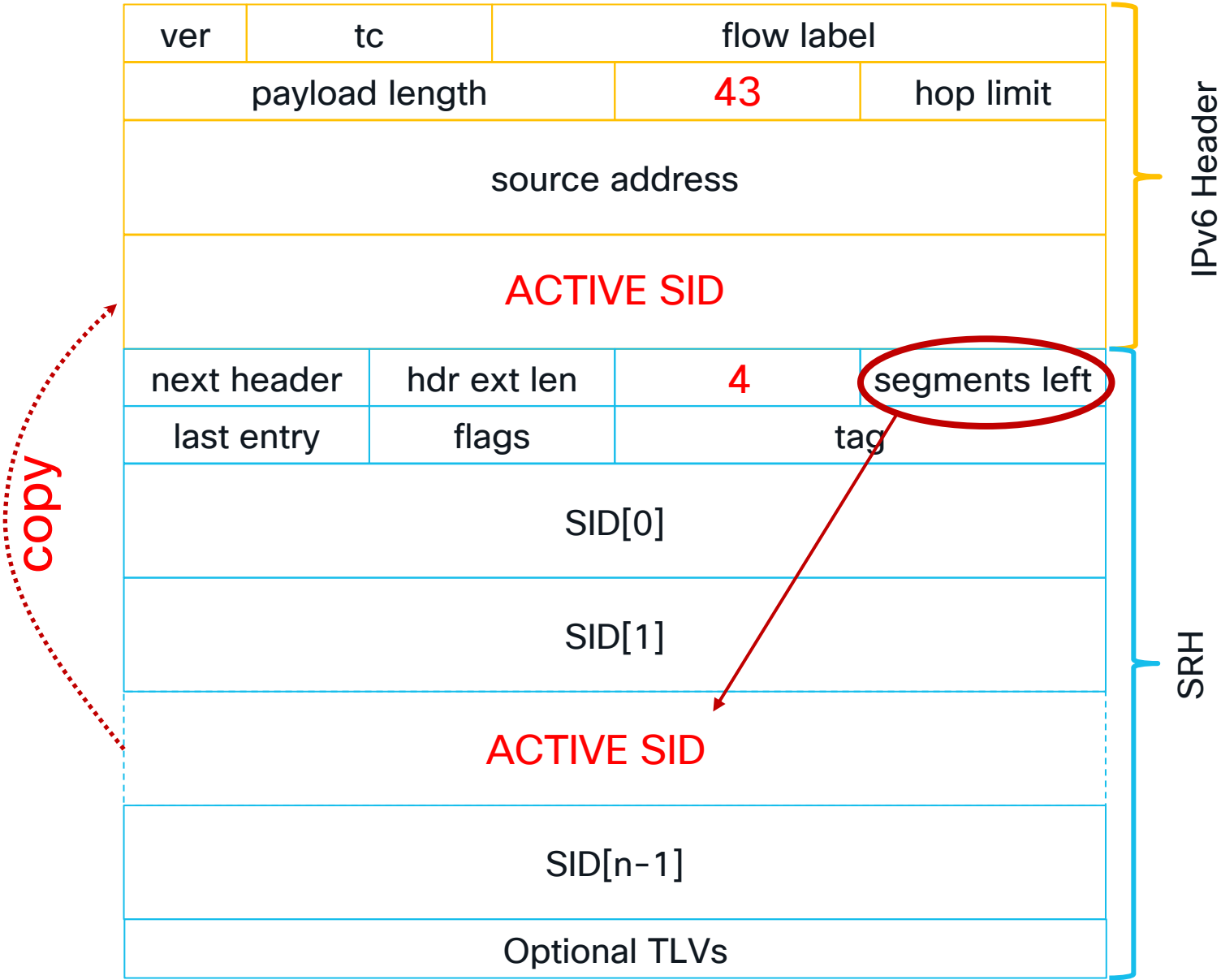
- IPv6 Header
- Destination IP address
- Next header field:
 - TCP, UDP, ICMP
 - IPv4, IPv6, L2
 - Hop by Hop, Dest. Options, Fragmentation, Authentication Header
- Routing Header
 - 0 Source Route (deprecated)
 - 1 Nimrod (deprecated)
 - 2 Type 2 (RFC 6275)
 - 3 RPL (RFC 6554)
 - 4 SRH (RFC 8754)**



Segment Routing Header

RFC 8754

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

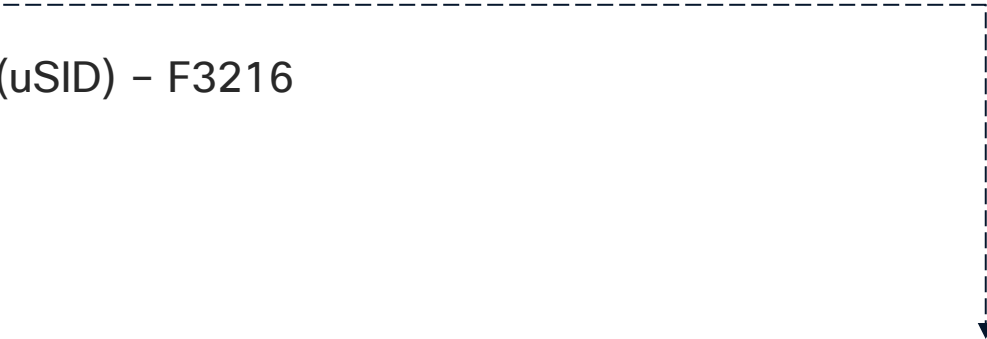


SID Structure

Locator and Function – 128 Bits Like IPv6 address but different semantics

Two SID Formats exist today:

- Base SID
- Micro SID (uSID) – F3216



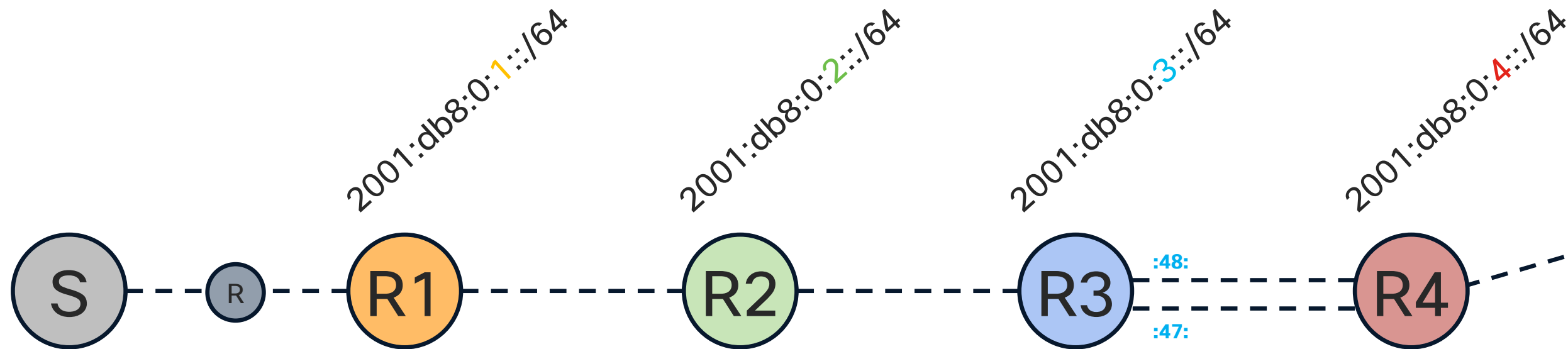
1111:2222:3333:4444:5555:6666:7777:8888



Locator

Function

SRv6 Base SID



SA:2001::1
DA:2001:db8:0:1:1::
NH:RH

Type:4(SRH)
NH:IPv4|SL:3
Segment List:
[0]:2001:db8:0:4:eeee::
[1]:2001:db8:0:3:48::
[2]:2001:db8:0:2:1::
[3]:2001:db8:0:1:1::

SA:2001::1
DA:2001:db8:0:2:1::
NH:RH

Type:4(SRH)
NH:IPv4|SL:2
Segment List:
[0]:2001:db8:0:4:eeee::
[1]:2001:db8:0:3:48::
[2]:2001:db8:0:2:1::
[3]:2001:db8:0:1:1::

SA:2001::1
DA:2001:db8:0:3:48::
NH:RH

Type:4(SRH)
NH:IPv4|SL:1
Segment List:
[0]:2001:db8:0:4:eeee::
[1]:2001:db8:0:3:48::
[2]:2001:db8:0:2:1::
[3]:2001:db8:0:1:1::

SA:2001::1
DA:2001:db8:0:4:eeee::
NH:IPv4

Type:4(SRH)
NH:IPv4|SL:0
Segment List:
[0]:2001:db8:0:4:eeee::
[1]:2001:db8:0:3:48::
[2]:2001:db8:0:2:1::
[3]:2001:db8:0:1:1::

SRv6 uSID Format

- Micro-SID consists of 16-bit fields: : 0100 : = SRv6 uSID

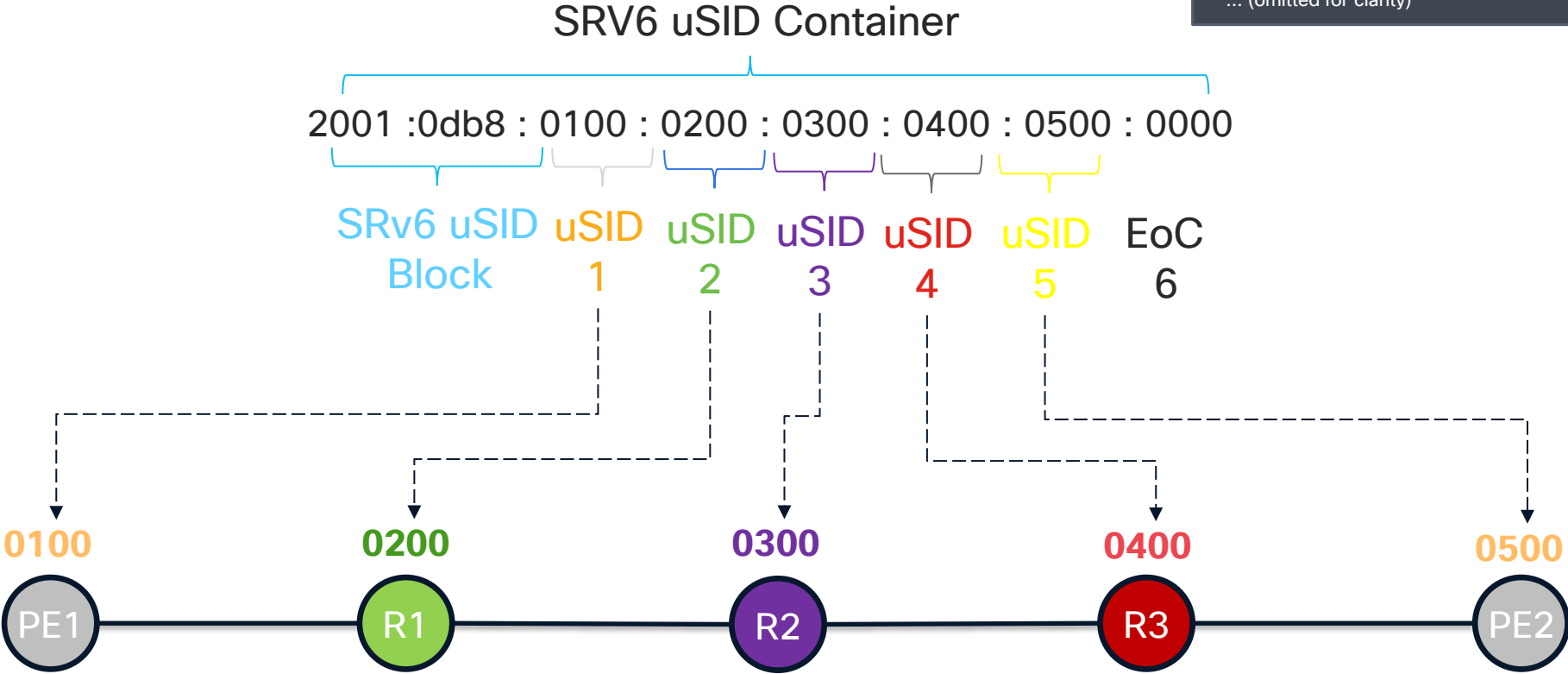
```
RP/0/RP0/CPU0:PE-1# show route ipv6 isis

i L1 2001:db8:100::/48
    [115/21] via fe80::42:acff:fe16:a303, 01:03:30, GigabitEthernet0/0/0/2

i L1 2001:db8:200::/48
    [115/21] via fe80::42:acff:fe16:a303, 01:03:22, GigabitEthernet0/0/0/2

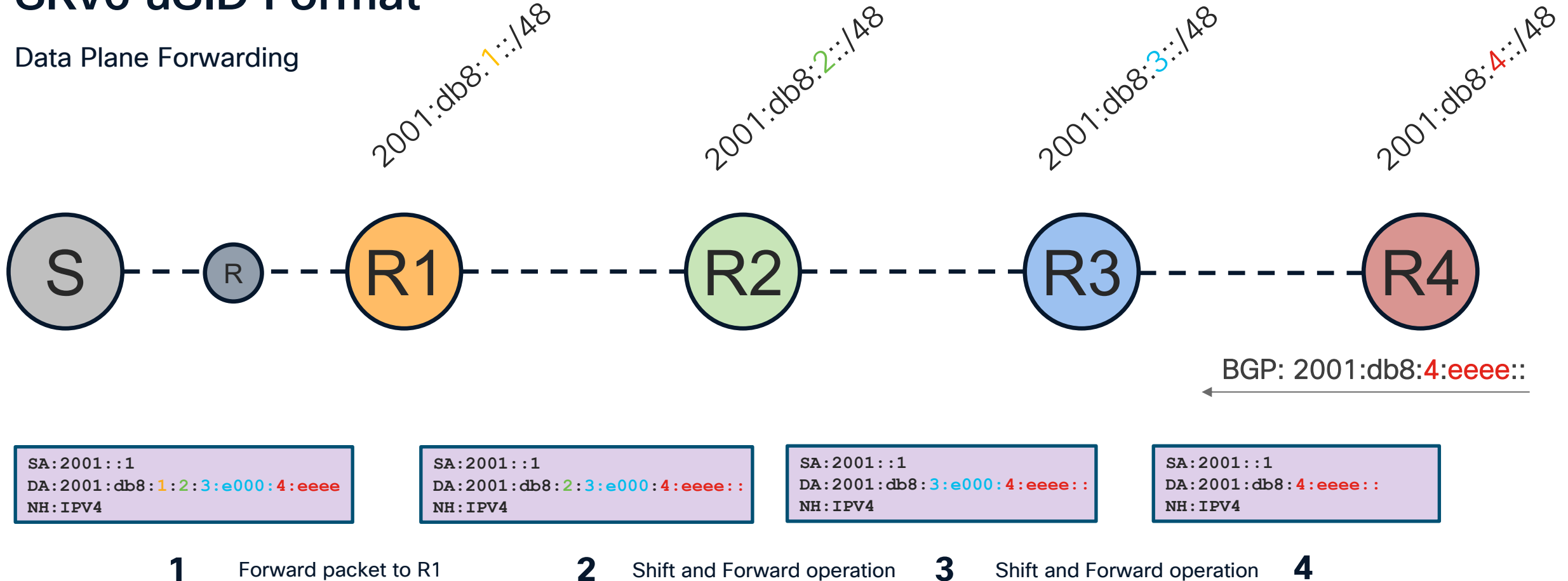
i L1 2001:db8:300::/48
    [115/11] via fe80::42:acff:fe16:a303, 01:03:30, GigabitEthernet0/0/0/2

... (omitted for clarity)
```



SRv6 uSID Format

Data Plane Forwarding



- Shift and Forward means:

1. Remove its own uSID from the container (from the Destination Address)
2. Shift the remaining bits (all remaining uSIDs) to the left
3. Forward the Packet to the next-hop (based on the new Destination Address)

Transit Node packet forwarding

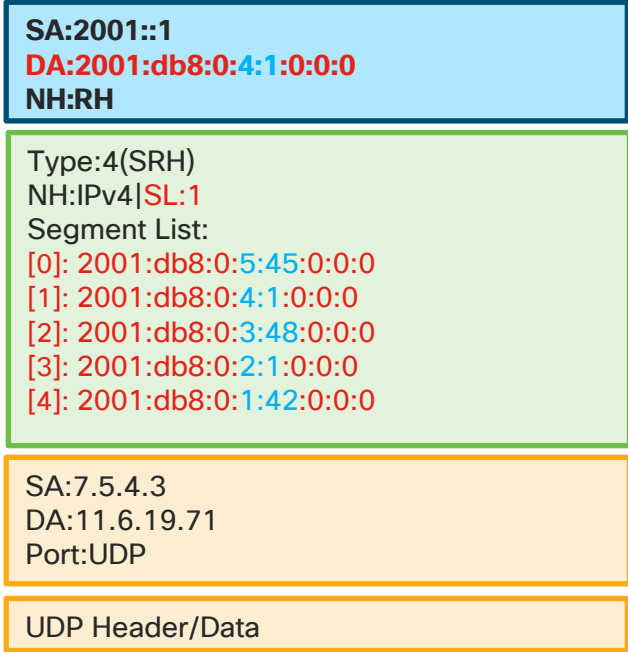
Longest Prefix Match

A transit Node simply forwards **2001:0db8:0100:0200:0300:0400:0500::** towards **0100**

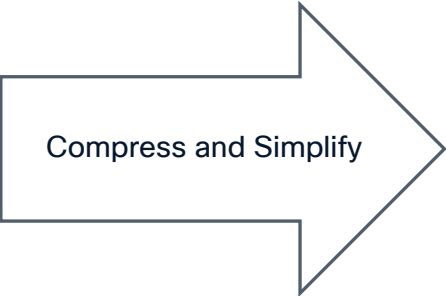
A transit node does not see further than longest match **2001:db8:0100**

SRv6 Base SID vs uSID

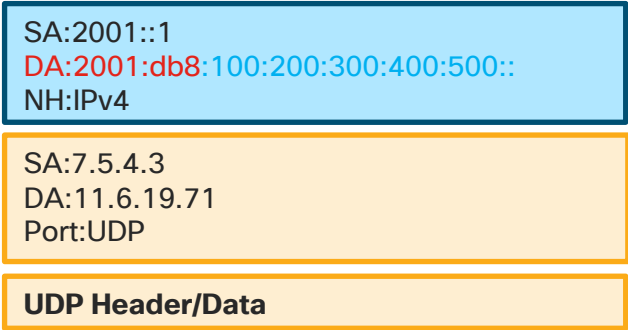
SRv6 Base SID Encapsulation



SRH: 5x 128 bits
IPv6 Source (SA): 1x 128
IPv6 Header (DA): 1x 128



SRv6 uSID Encapsulation



No SRH
IPv6 Header (SA): 1x 128
IPv6 Header (DA): 1x 128

SRv6 uSID

All SR concepts apply to SRv6

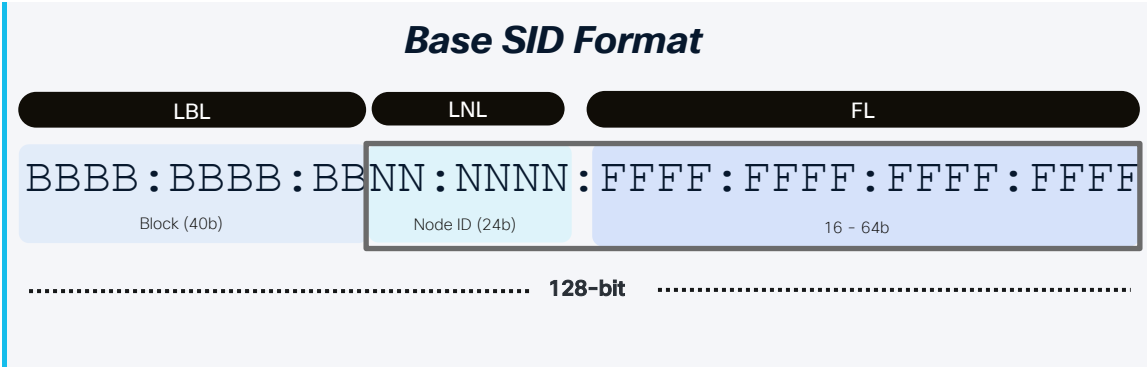
- **Flex-Algo**
- **Ti-LFA**
- **L2VPN / L3VPN**
- **SR Policy**

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]

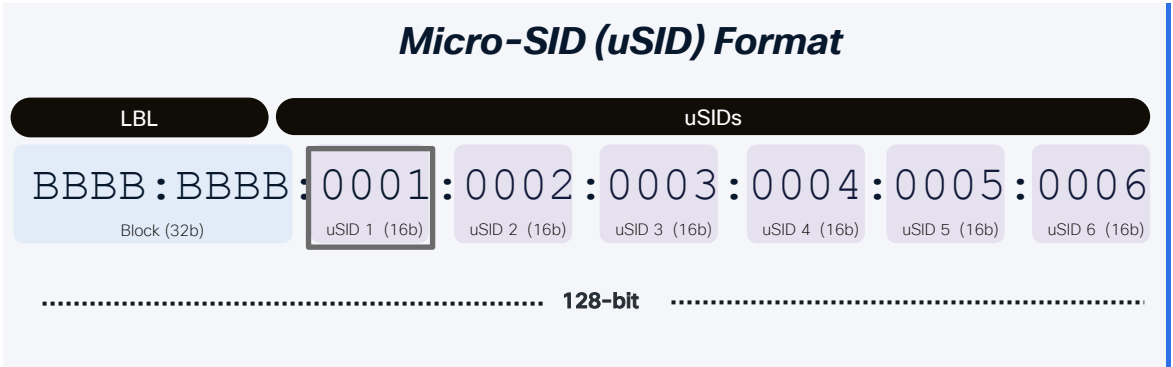
SRv6 in a Nutshell

BEFORE COMPRESSION



ACRONYM	DESCRIPTION
LBL	Locator-Block length
LNL	Locator-Node length
FL	Function length

AFTER COMPRESSION

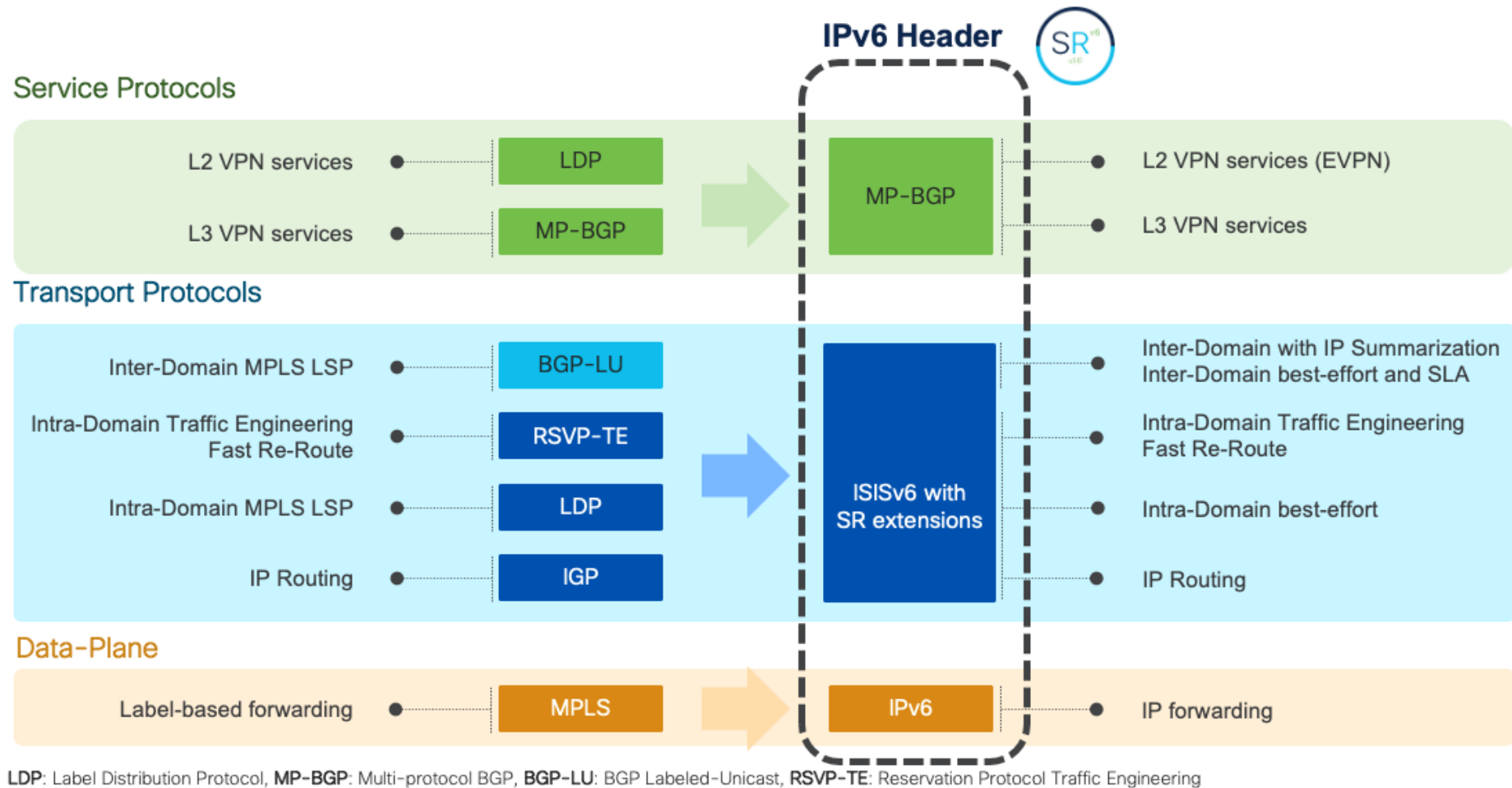


ACRONYM	DESCRIPTION
LBL	Locator-Block length
uSID	Micro SIDs

Protocol Simplification

Protocol Simplification

Network Evolution



Conclusion

Segment Routing MPLS and SRv6 are Fully Standardized

Architecture

- SR Architecture – **RFC 8402**
- SRTE Policy Architecture – **RFC 9256**
- Compressed SRv6 Segment List – **WG Draft**

Data Plane

- SRv6 Network Programming – **RFC 8986**
- IPv6 SR header – **RFC 8754**

Control Plane

- SRv6 BGP Services – **RFC 9252**
- SRv6 ISIS – **RFC 9352**
- SR Flex-Algo – **RFC 9350**

Operation & Management

- SRv6 OAM – **RFC 9259**
- Performance Management – **RFC 5357**

Strong Cisco Commitment and Leadership

Editor of 96% IETF RFCs
Co-author of 100% IETF RFCs

Rich SRv6 uSID Ecosystem

Network Equipment Manufacturers



Merchant Silicon



Open-Source Applications



Open-Source Networking Stacks



Smart NIC



Partners



Customer Traction



SRv6 Customer Adoption



Complete your session evaluations



Complete a minimum of 4 session surveys and the Overall Event Survey to claim a Cisco Live T-Shirt.



Earn up to 800 points by completing all surveys and climb the Cisco Live Challenge leaderboard.



Level up and earn exclusive prizes!



Complete your surveys in the Cisco Live Events app.

Continue your education



Visit the Cisco Stand for related demos



Book your one-on-one Meet the Expert meeting



Attend the interactive education with Capture the Flag, and Walk-in Labs



Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Contact me at: lhoxha@cisco.com

Thank you

CISCO Live !

