The bridge to possible

Preparing for a Successful SR Deployment

Jose Liste, Distinguished TME

BRKMPL-2135

cisco Live!

#CiscoLive

Cisco Webex App

https://ciscolive.ciscoevents.com/ciscolivebot/#BRKMPL-2135

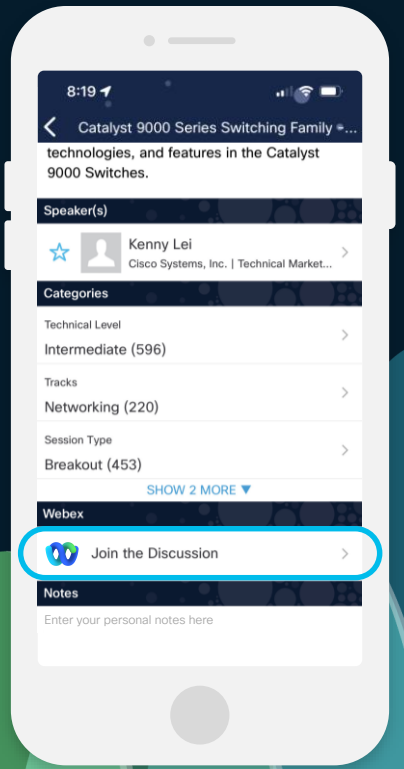
Questions?

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 June 7, 2024.



Getting a hold of me

jliste@cisco.com

Objectives

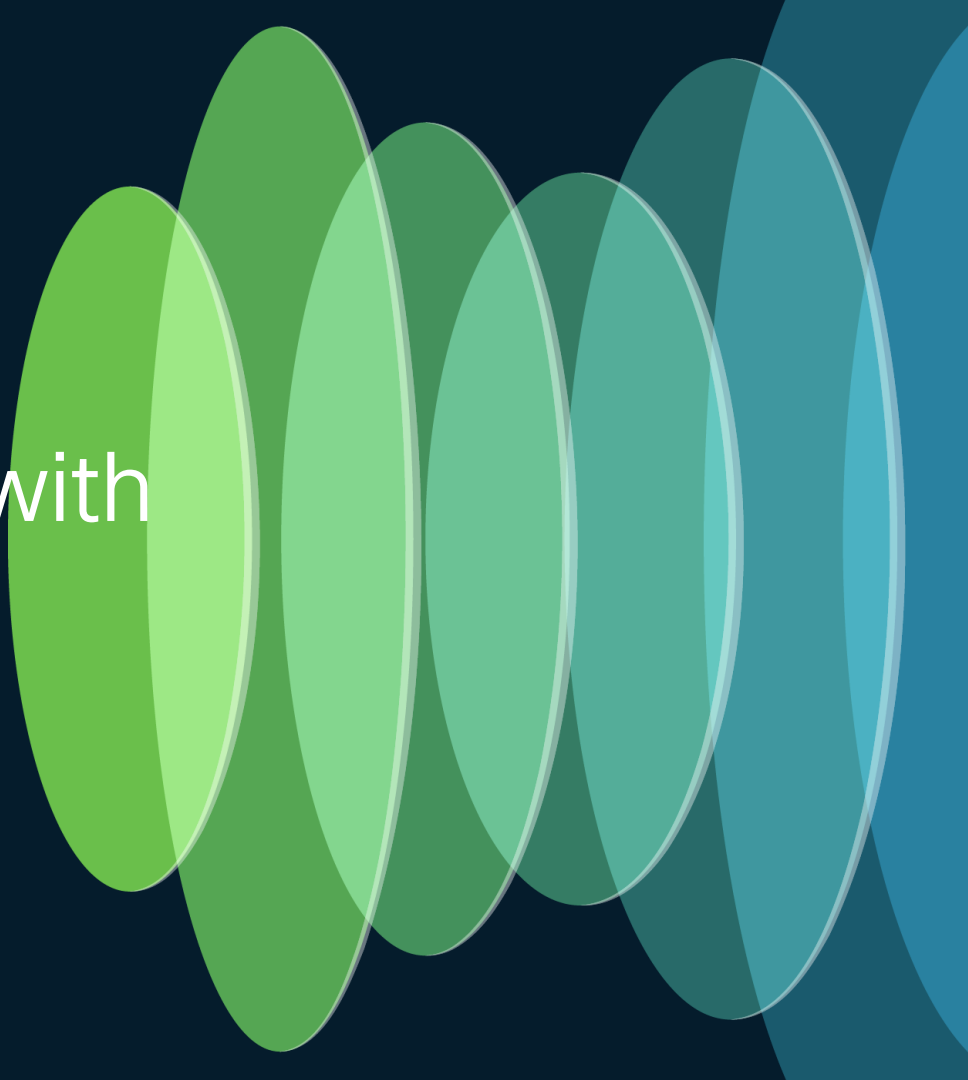
Presentation covers key SR technical building blocks, feature highlights and best practices for a successful rollout of SR in your network

Presentation focuses on Cisco IOS-XR implementation and examples across SR-MPLS and SRv6

Agenda

- Introduction
- SR-MPLS Label Blocks
- Resiliency with BGP-SR and Weighted Anycast SID
- SRTE / SR-PCE
- Flexible Algorithm
- Transport Assurance with SR-PM
- Conclusion

Network Evolution with Segment Routing



cisco Live!

One Architecture / Two Data-Plane instantiations

Segment Routing

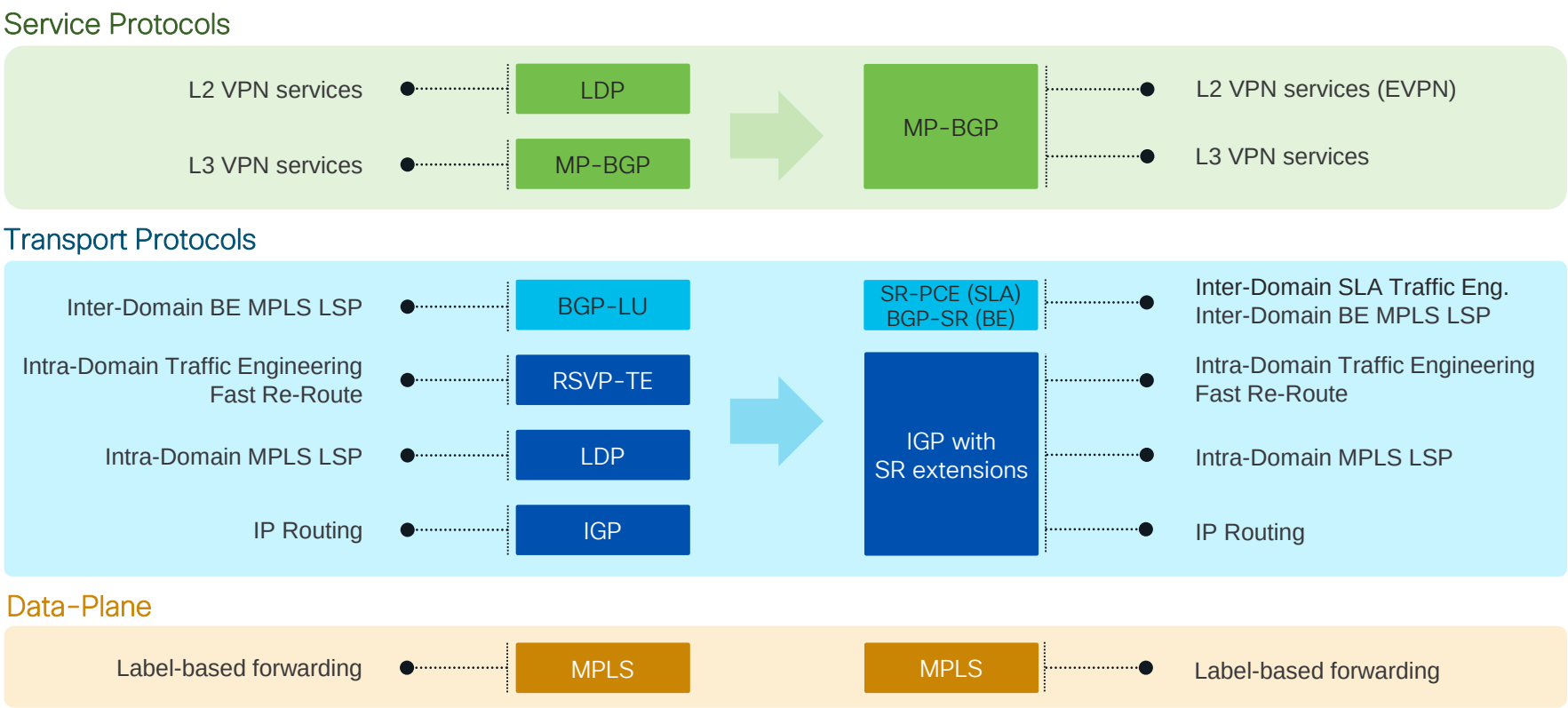
SR-MPLS

- Instantiation of SR on the MPLS data-plane
- Segment ID (SID) is an MPLS label associated with the segment

SRv6

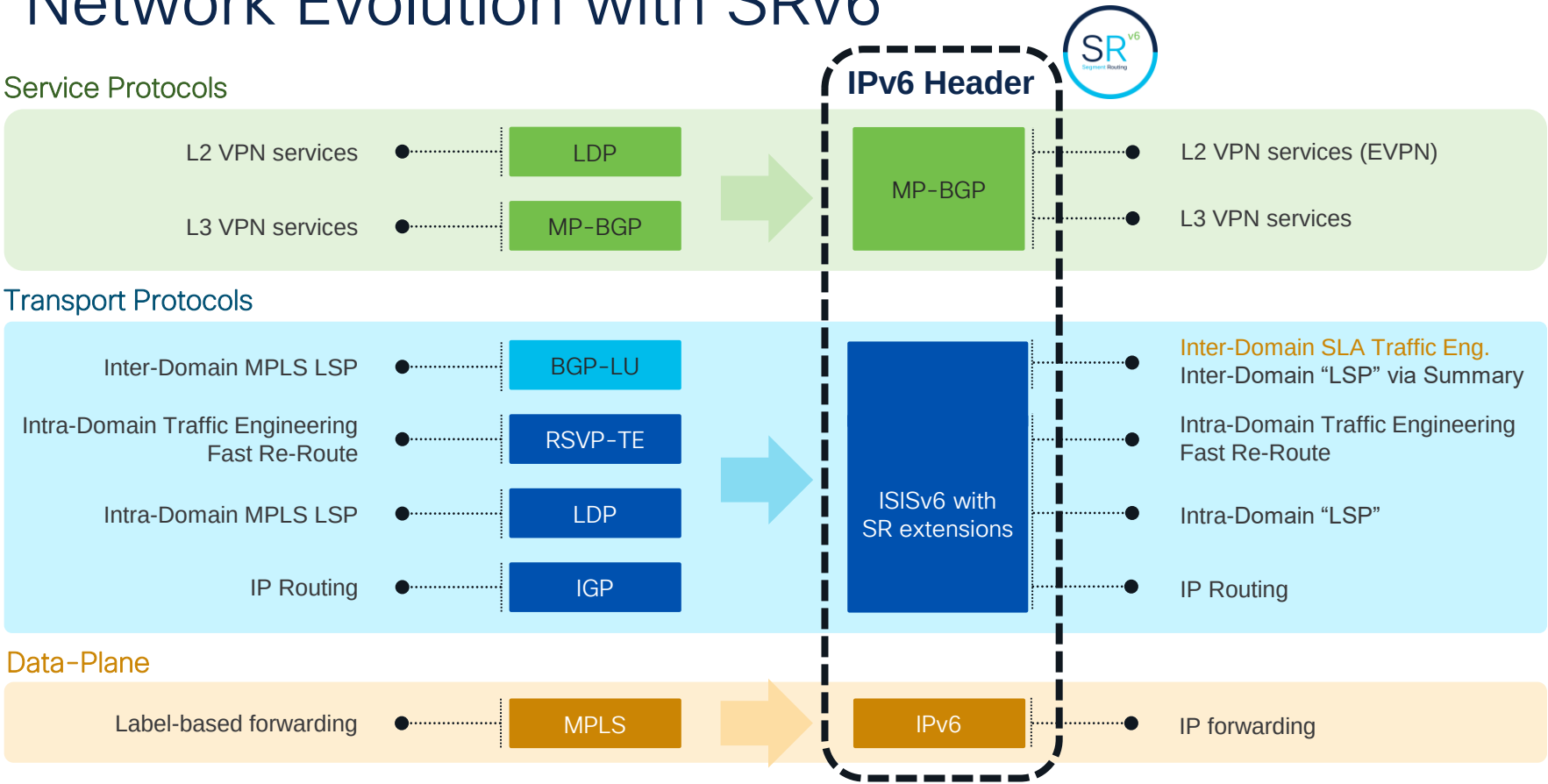
- Instantiation of SR on the IPv6 data-plane
- SID is a IPv6 prefix associated with the segment

Network Evolution with SR-MPLS



LDP: Label Distribution Protocol, **MP-BGP**: Multi-protocol BGP, **BGP-LU**: BGP Labeled-Unicast, **PCE**: Path Computation Element, **RSVP-TE**: Reservation Protocol Traffic Engineering

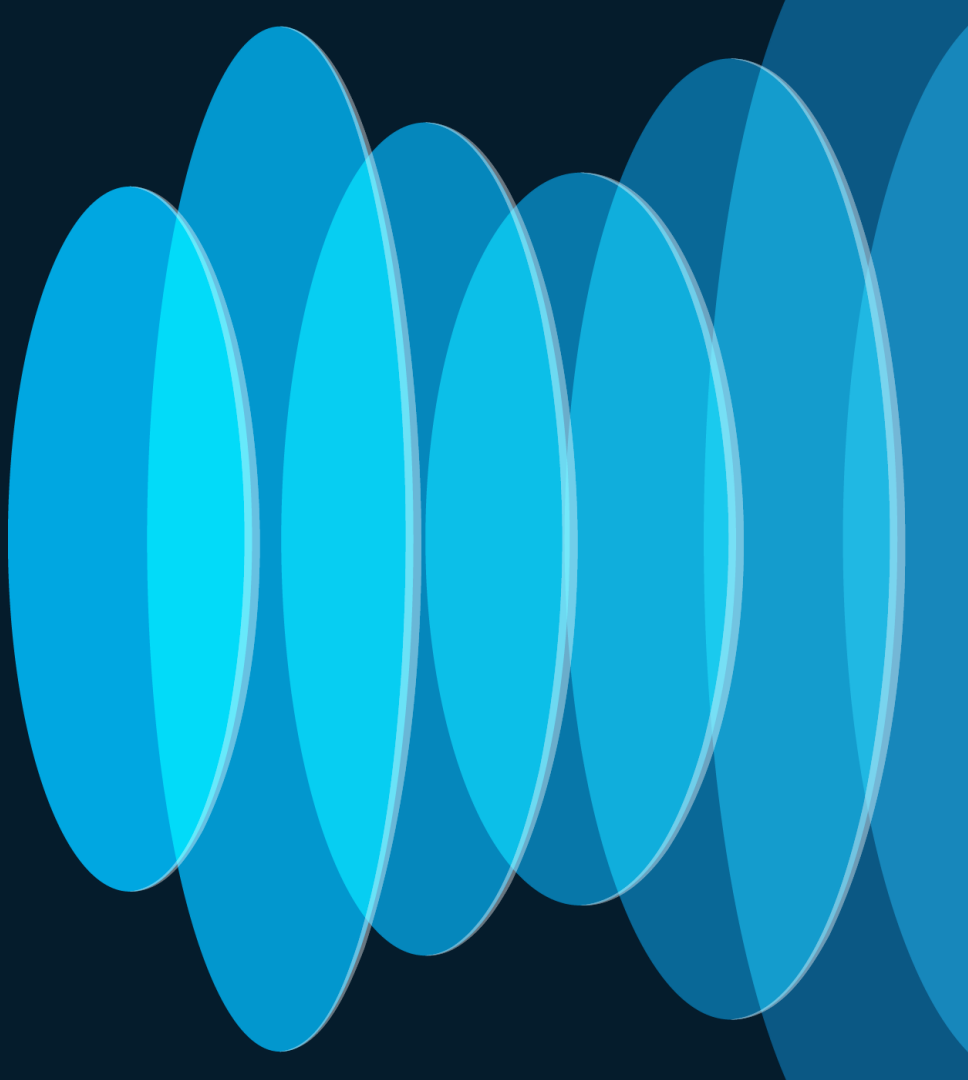
Network Evolution with SRv6



LDP: Label Distribution Protocol, **MP-BGP**: Multi-protocol BGP, **BGP-LU**: BGP Labeled-Unicast, **RSVP-TE**: Reservation Protocol Traffic Engineering

SR-MPLS Label Blocks

Design Recommendations



cisco Live!

SR Global Block (SRGB)

- Segment Routing Global Block
  - Range of labels reserved for Global Segments – Prefix-SID
- A prefix-SID is advertised as a domain-wide unique index
  - 1 to 1 mapping between prefix and prefix-sid index
- IOS-XR implementation
  - Default SRGB is 16,000 – 23,999 (size 8,000)
  - User can configure a different SRGB between (16,000 and 1,048,575)
  - Does not impose any constraints on the max size
    - One needs to ensure that enough labels are available in the dynamic label pool for service labels, local SIDs, and other dynamic labels

SR Global Block (SRGB)

- The Prefix-SID index points to a unique label within the SRGB
  - Index is zero based, i.e., first index = 0
  - Label = SRGB base + Prefix-SID index

Example:
- Loopbak 0 = 10.10.10.1/32, prefix SID index 1 gest label 16001
- SRGB – 16,000 – 23,999

OR

```
router isis 1
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 1
```

```
router isis 1
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid absolute 16001
```

SRGB Configuration

- The SRGB can be configured:
  - Globally (Recommended)
    - By default, all IGP instances and BGP use this global SRGB
  - Under IGP (Not Recommended)

```
segment-routing
 global-block 18000 19999
```

Recommended

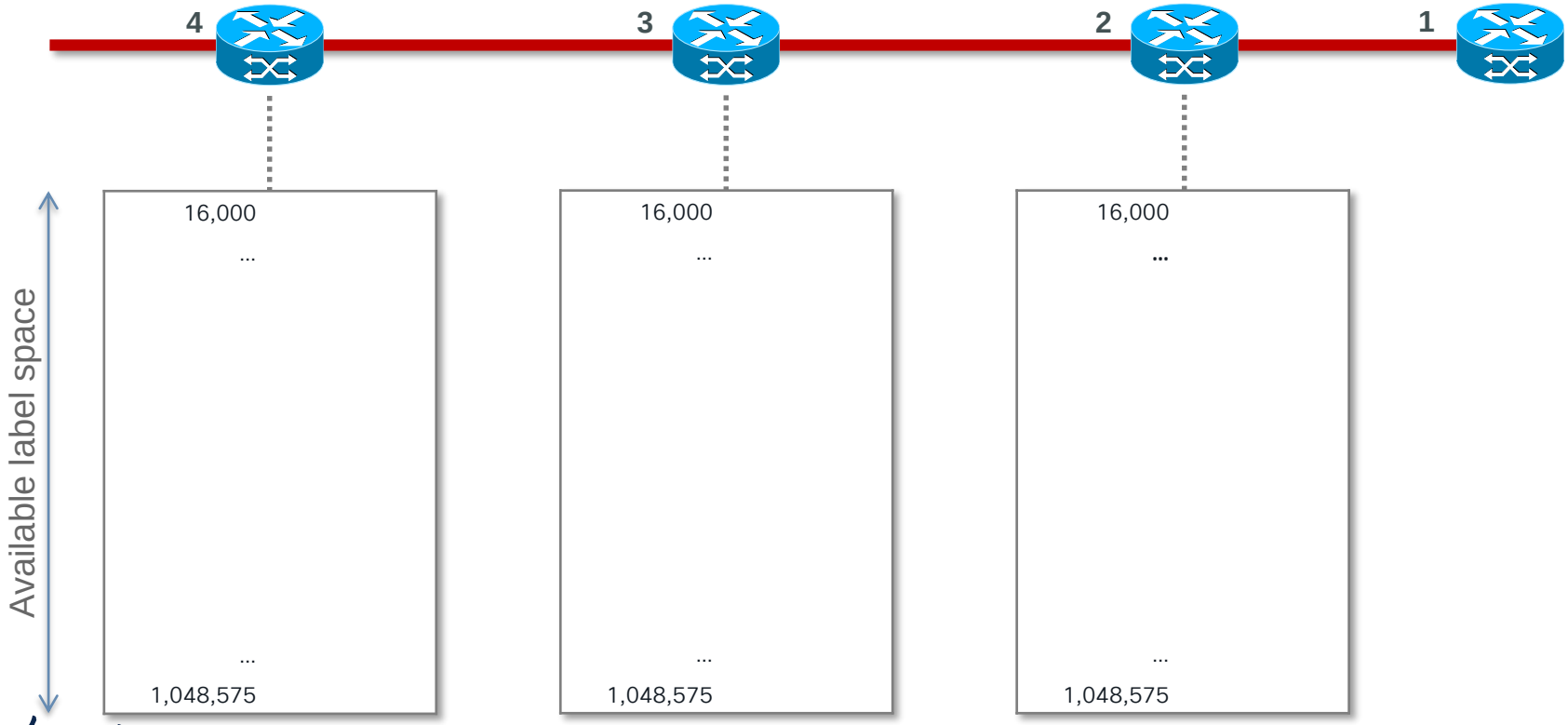
```
router isis | ospf <tag>
 segment-routing global-block 18000 19999
```

Not Recommended

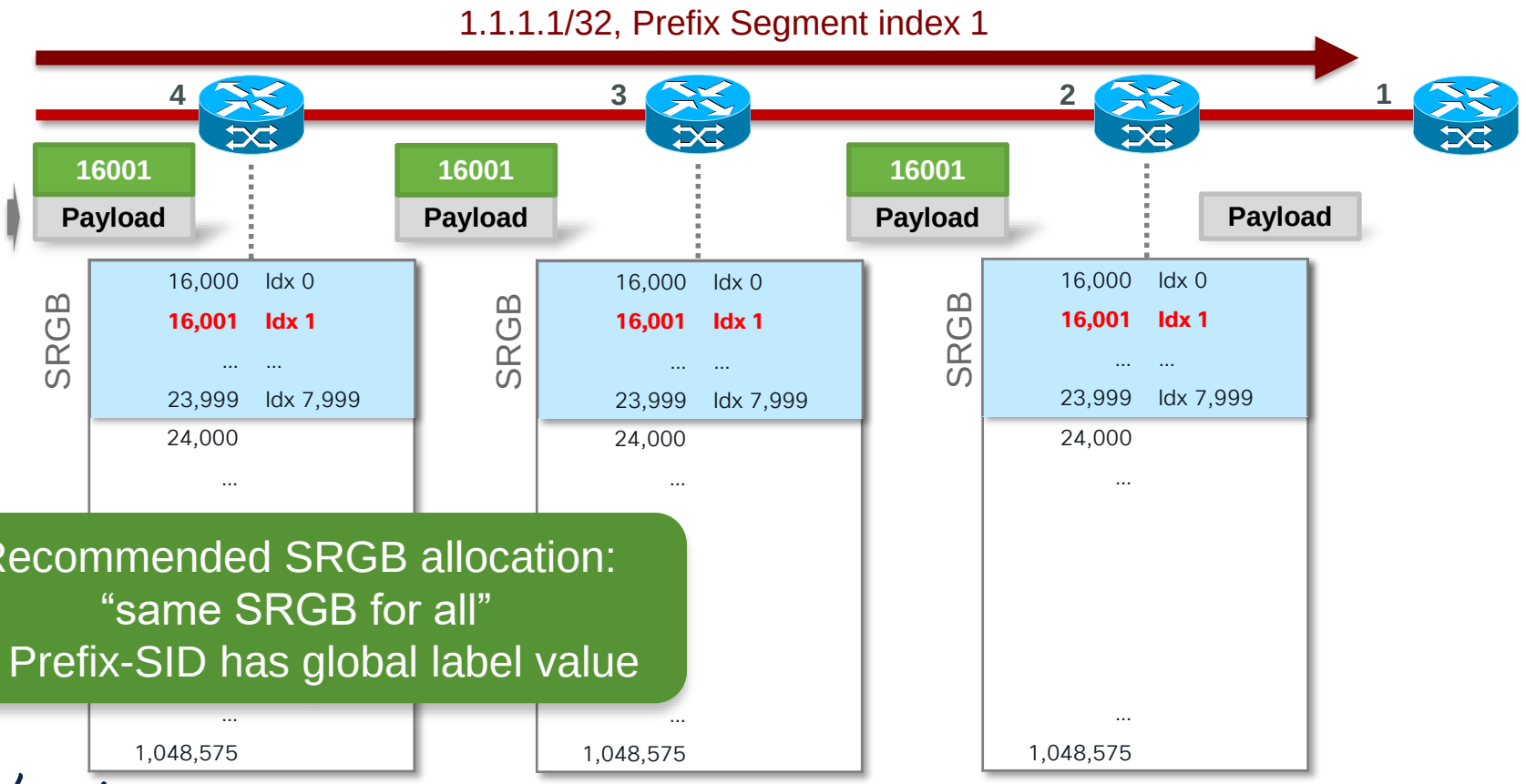
SRGB design

- SRGB planning should aim for the following goals:
  - Goal 1: Homogenous SRGB
  - Goal 2: Unique SID-to-prefix mappings
    - SRGB size > # required SIDs
    - Each SID can be allocated to a single prefix. No SID re-use among prefixes
- Large majority of deployments should be able to meet these goals
  - in rare cases, these goals cannot be met due to network scale an/or platform limitations

Segment Routing Global Block (SRGB)

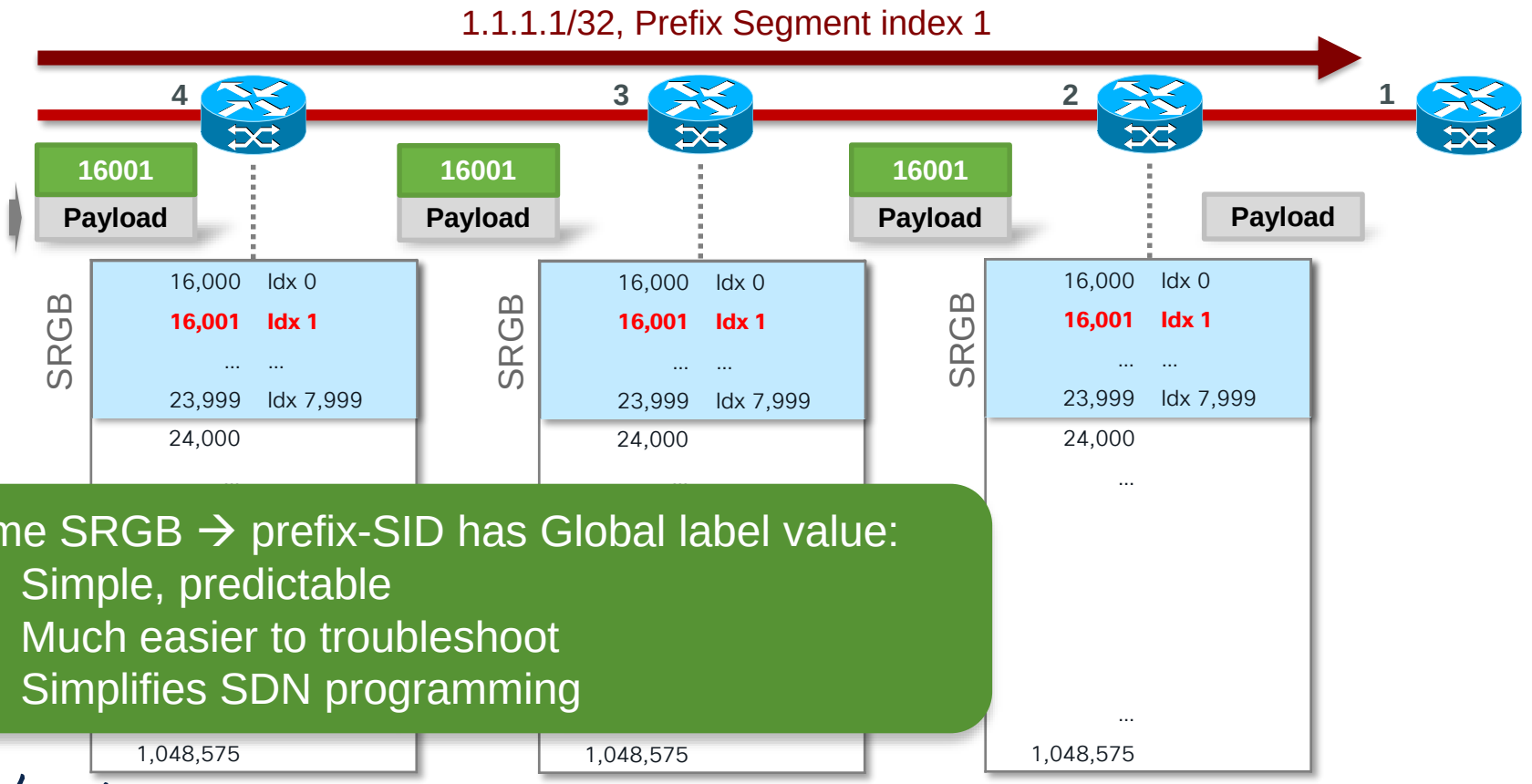


Homogeneous SRGB allocation (Recommended)



Recommended SRGB allocation:
"same SRGB for all"
→ Prefix-SID has global label value

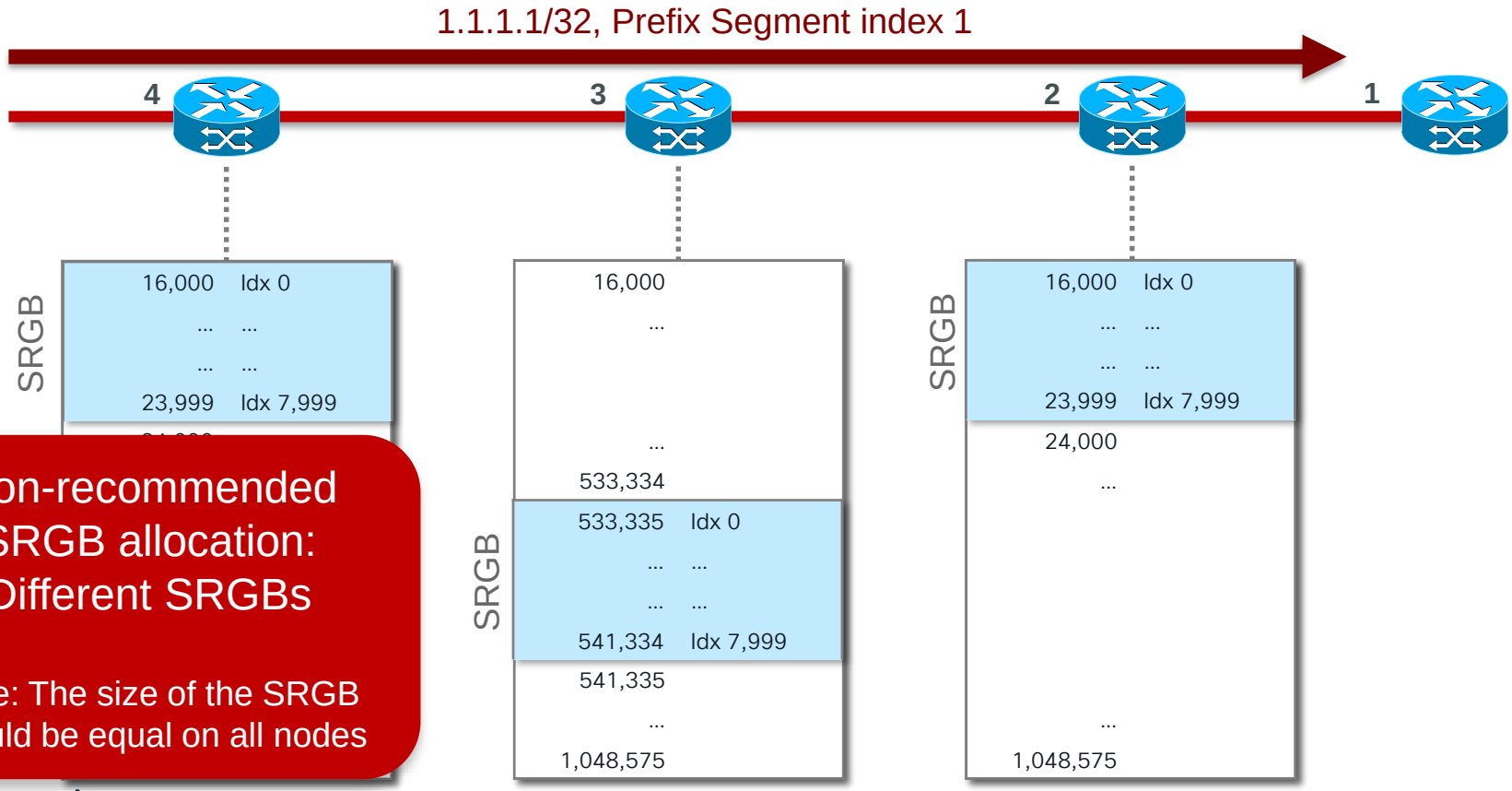
Homogeneous SRGB allocation (Recommended)



Same SRGB → prefix-SID has Global label value:

- Simple, predictable
- Much easier to troubleshoot
- Simplifies SDN programming

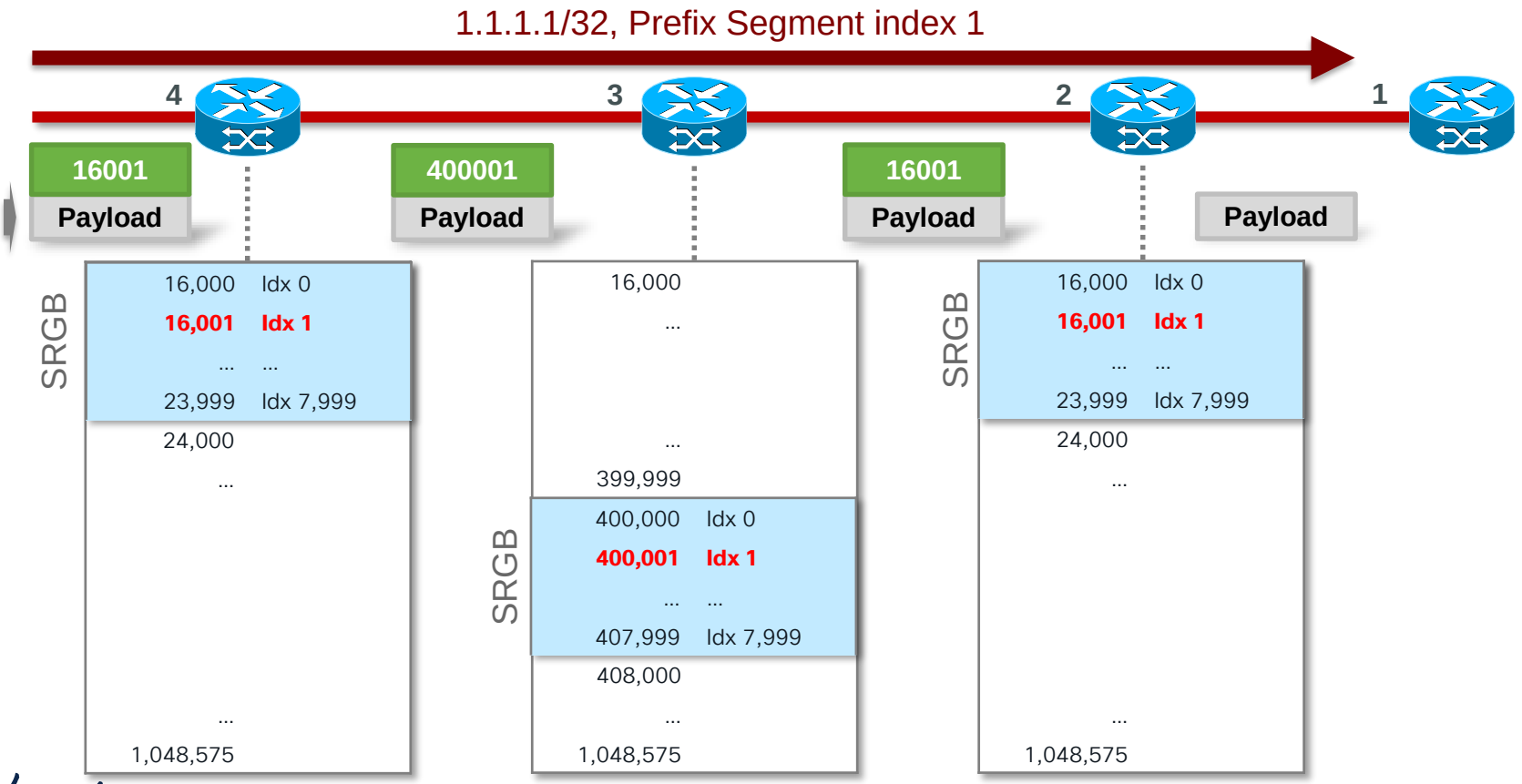
Heterogeneous SRGB allocation (Not Recommended)



Non-recommended SRGB allocation: Different SRGBs

Note: The size of the SRGB should be equal on all nodes

Heterogeneous SRGB allocation (Not Recommended)



SRGB label range preservation

- IOS-XR preserves the default SRGB label range [16,000-23,999]
  - In any Segment Routing capable software release
  - Even if Segment Routing is not enabled
  - Except if the configured mpls label range includes this default range
- LSD (IOS-XR label manager) allocates dynamic labels starting from 24,000

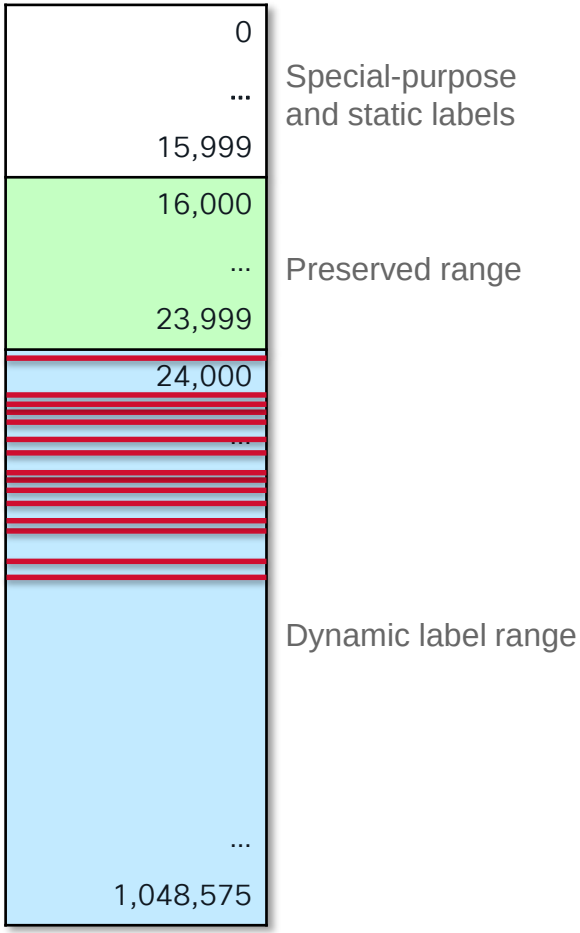
SRGB label range preservation

- Preservation of the default SRGB label range makes future Segment Routing activation possible without reboot
  - No labels are allocated from that preserved range. When enabling Segment Routing with default SRGB some time in the future, that label range is available and ready for use

LSD SRGB preservation – Example

- An example sequence of Segment Routing activation:

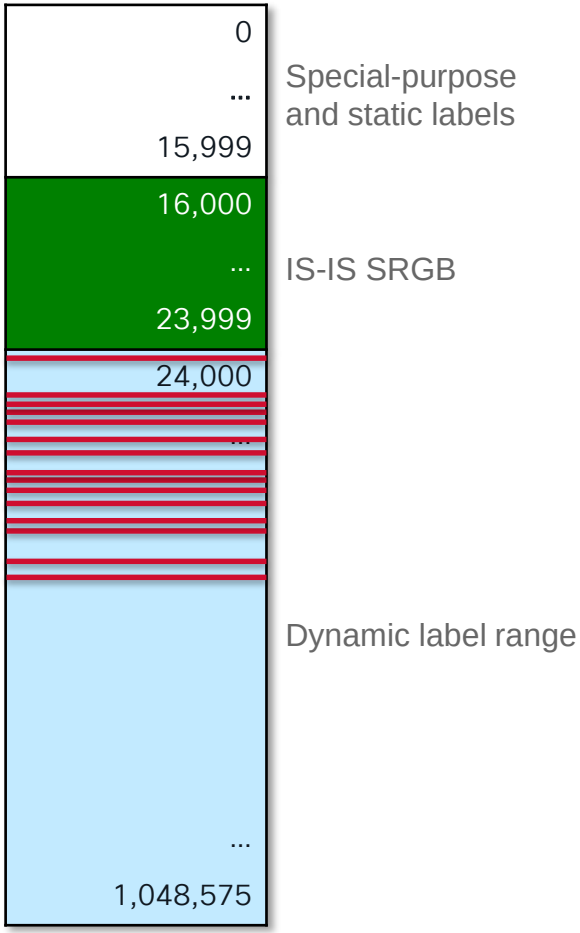
1. No Segment Routing enabled, no SRGB allocated
   - LSD preserves default SRGB label range
   - Dynamic labels are allocated by various MPLS applications (— in diagram)



LSD SRGB preservation – Example

- An example sequence of Segment Routing activation:

1. No Segment Routing enabled, no SRGB allocated
   - LSD preserves default SRGB label range
   - Dynamic labels are allocated by various MPLS applications (— in diagram)
2. Sometime later, Segment Routing IS-IS is enabled with default SRGB
   - SRGB label range is free (preserved), start using Segment Routing without reboot!



Segment Routing Global Block (SRGB) Notes

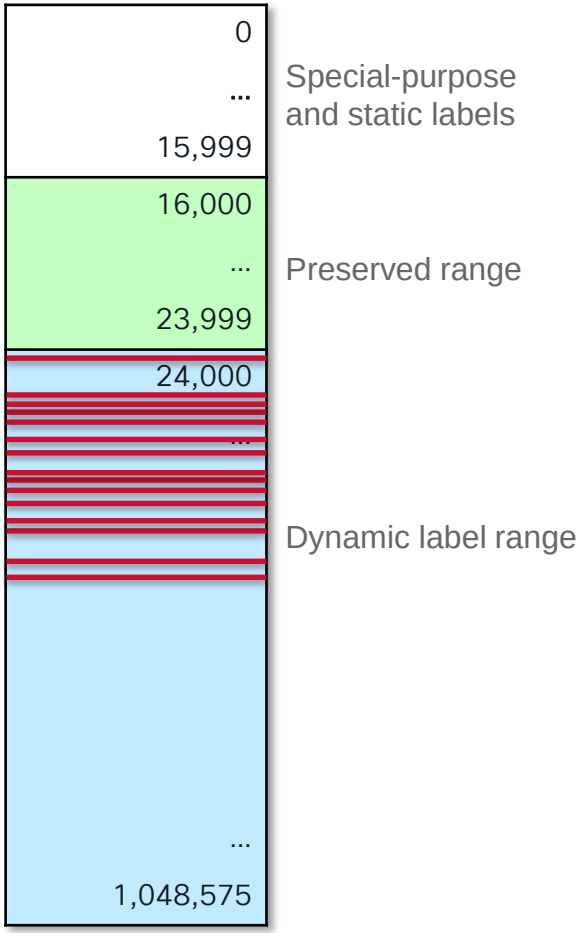
- Modifying a SRGB configuration is disruptive for traffic
  - And may require a reboot if the new SRGB is not (entirely) available
- Best practice: allocate a non-default SRGB in the upper part of the MPLS label space increases the chance that the labels would be free (validate first)

```
segment-routing
 global-block 400000 431999
```

LSD SRGB allocation – Example

- An example sequence of Segment Routing activation:

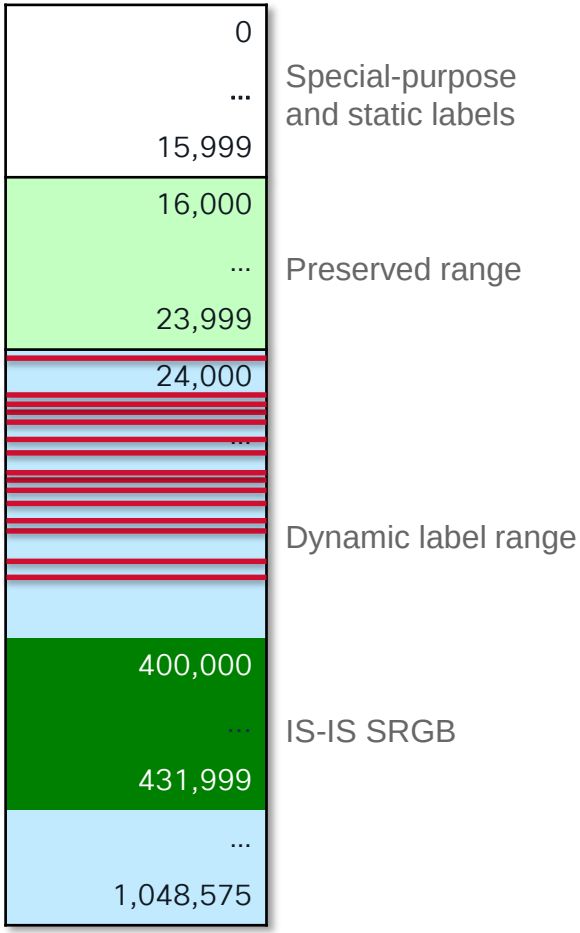
1. No Segment Routing enabled, no SRGB allocated
   - LSD preserves default SRGB label range
   - Dynamic labels are allocated by various MPLS applications (—in diagram)



LSD SRGB allocation - Example

- An example sequence of Segment Routing activation:

1. No Segment Routing enabled, no SRGB allocated
   - LSD preserves default SRGB label range
   - Dynamic labels are allocated by various MPLS applications (▬ in diagram)
2. Sometime later, SR IS-IS is enabled with non-default SRGB in the upper label range (hence likely unused)
   - SRGB label range is free, start using SR without reboot!



Segment Routing Global Block (SRGB)

Non-default SRGB Example

```
segment-routing
 global-block 400000 431999
!
router isis 1
 address-family ipv4 unicast
  segment-routing mpls
```

Configure a non-default SRGB

Non-default SRGB label block allocation for ISIS
[400,000 – 431,999]

```
RP/0/0/CPU0:xrvr-1#show mpls label table detail
Table Label   Owner                              State  Rewrite
----- ------- ---------------------------------- ------ -------
<...snip...>

0     400000   ISIS(A):1                          InUse  No
  (Lbl-blk SRGB, vers:0, (start_label=400000, size=32000)
<...snip...>
```

SRGB

Start_label = 400,000

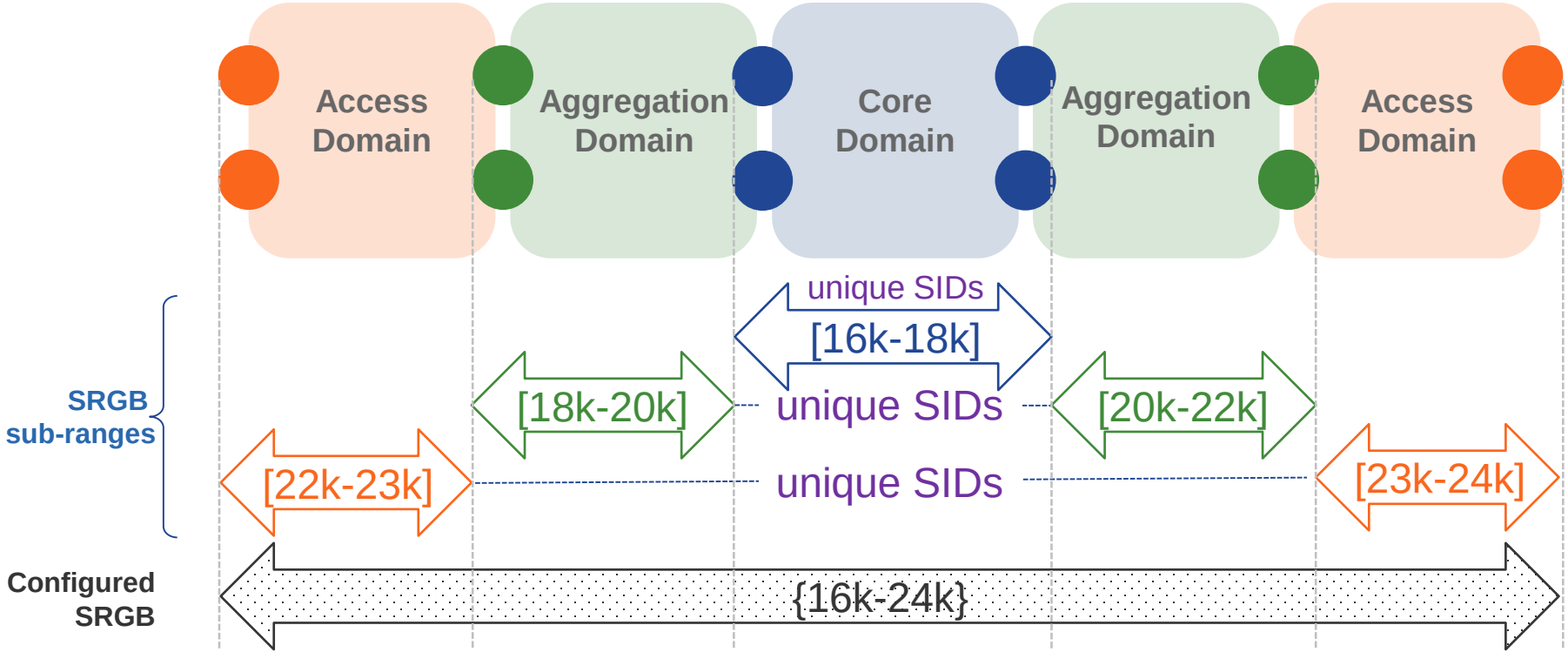
Size = 32,000

Configured SRGB and SRGB sub-ranges

- For ease of administration and operations, the configured SRGB is carved in administrative sub-ranges
  - Allocate a sub-range to each domain
  - The configured SRGB is still the entire SRGB, not the SRGB sub-range
- Alternatively, an operator could treat the configured SRGB as a global pool of SIDs
  - A global pool might lead to a more optimal use of the SRGB
  - More complicated administration

SRGB and SRGB sub-ranges

Notation convention:
SRGB {XXX-YYY}
sub-range [QQQ-RRR]



Note: [16k-17k] really means [16000-16999]

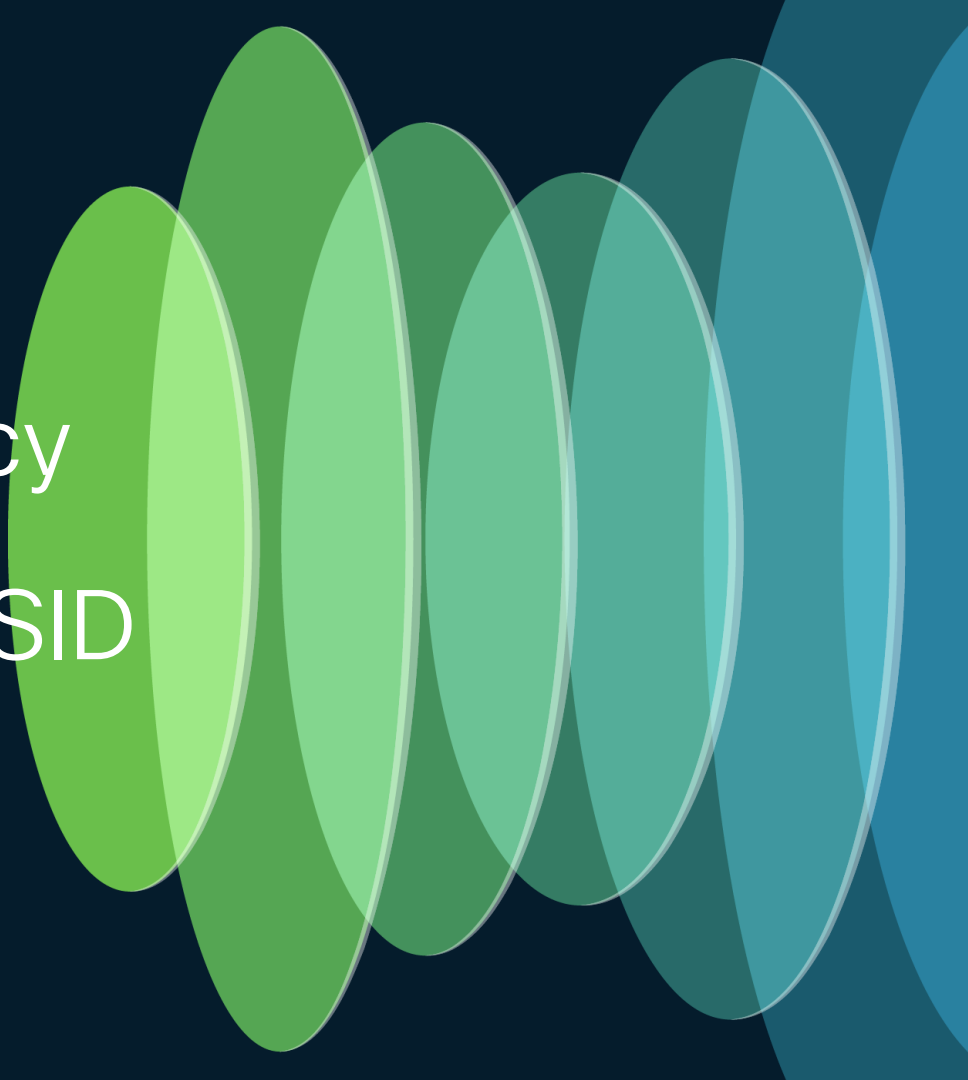
What should be the size of my SRGB? How many Prefix SIDs would I need?

- Number of prefix sids SIDs MAY be larger than number of nodes
- i.e., more than 1 prefix SIDs may be needed per node
  - Algo(0) Prefix-SID
  - With Flex-Algo, multiple SIDs are mapped to a prefix
  - How many Flex-Algo SIDs?
    - Delay metric: 1 Flex-Algo for low-delay service
    - TE metric: 1 Flex-Algo for e.g. premium service
    - IGP metric: 2 Flex-Algos for dual-plane
    - FA with affinity constraints? E.g., use encrypted links, avoid low BW links

How many Flex-Algos should I estimate?

- An estimate for number of Flex-Algos required is between 4-8
  - IGP metric: 2 Flex-Algos for dual-plane
    - Maybe more for additional slices?
    - More using affinities? E.g. use encrypted links, avoid low BW links
  - TE metric: 1 Flex-Algo for e.g. premium service
  - Delay metric: 1 Flex-Algo for low-delay service
  - Multiply by 2 for future expansion
    → (2 + 1 + 1) * 2 = 8 Flex-Algos
- This fits the rule-of-thumb to limit # Flex-Algos on a node should be single-digit number

Increasing Resiliency with BGP-SR and Weighted Anycast SID



cisco Live!

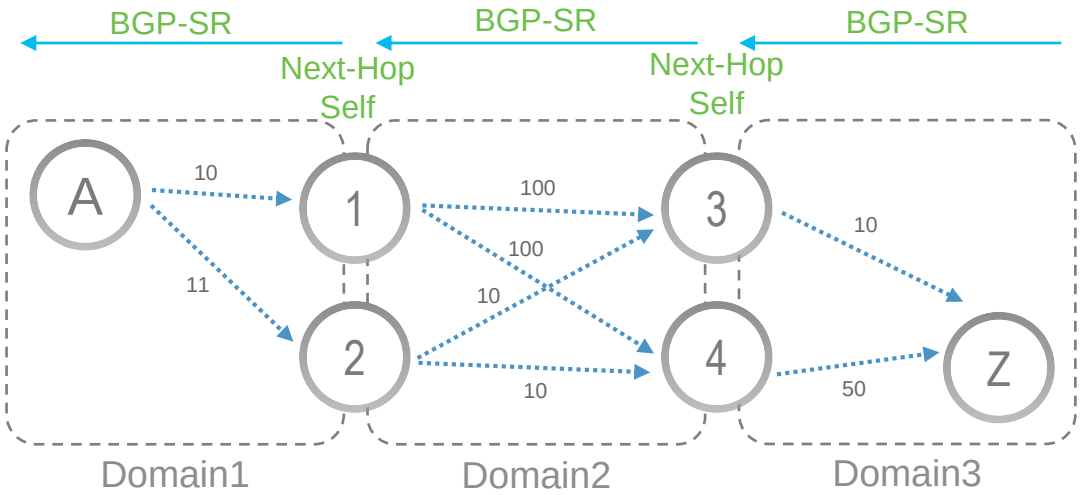
Increasing Resiliency with BGP-SR

- Better resiliency with BGP-SR !!!
- Thanks for SR Prefix SID global labels, BGP-LU local labels are the same across ASBRs

- As a result, ASBR (inline BGP-LU RRs) can use anycast loopback as NH for BGP-LU prefixes

Multi-Domain – BGP-SR

- With BGP-SR, LU local labels are the same across the network
  - E.g., BGP-LU label for loopback of node Z at ASBR 1, 2, 3, 4 = 1600Z

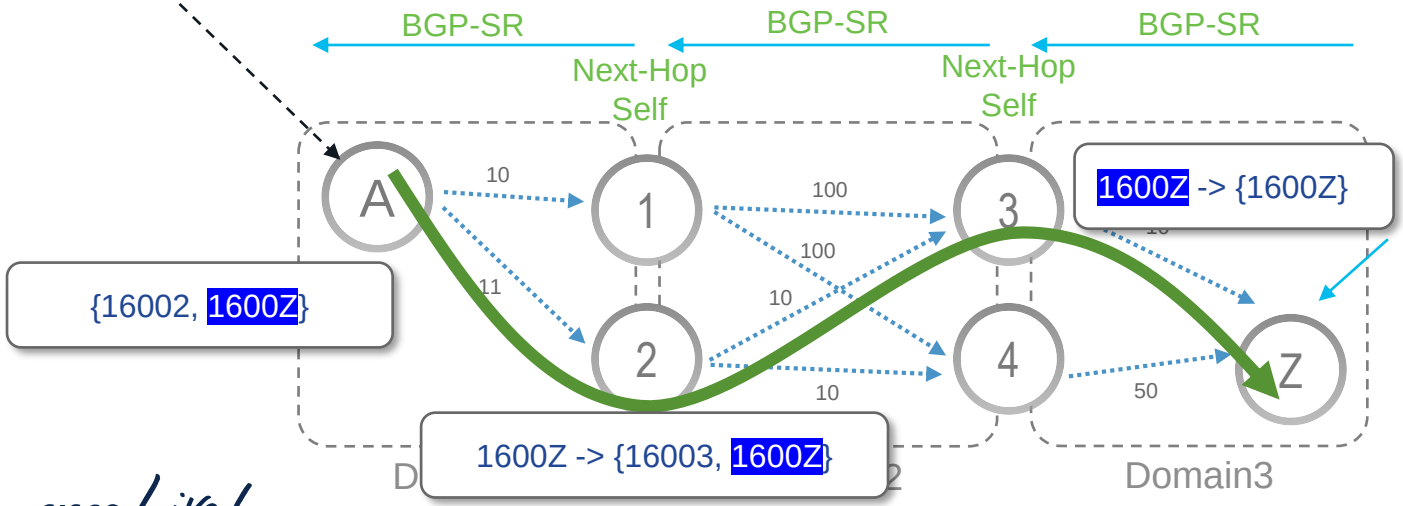


Link IGP metric shown

Multi-Domain – BGP-SR

- Shortest path from A to Z?
- Run BGP best-path including AIGP

BGP RIB at Node A
Z/32 via 1.1.1.1, 1600Z
1.1.1.2, 1600Z (best path)



Optimum path using Node SIDs

Link IGP metric shown

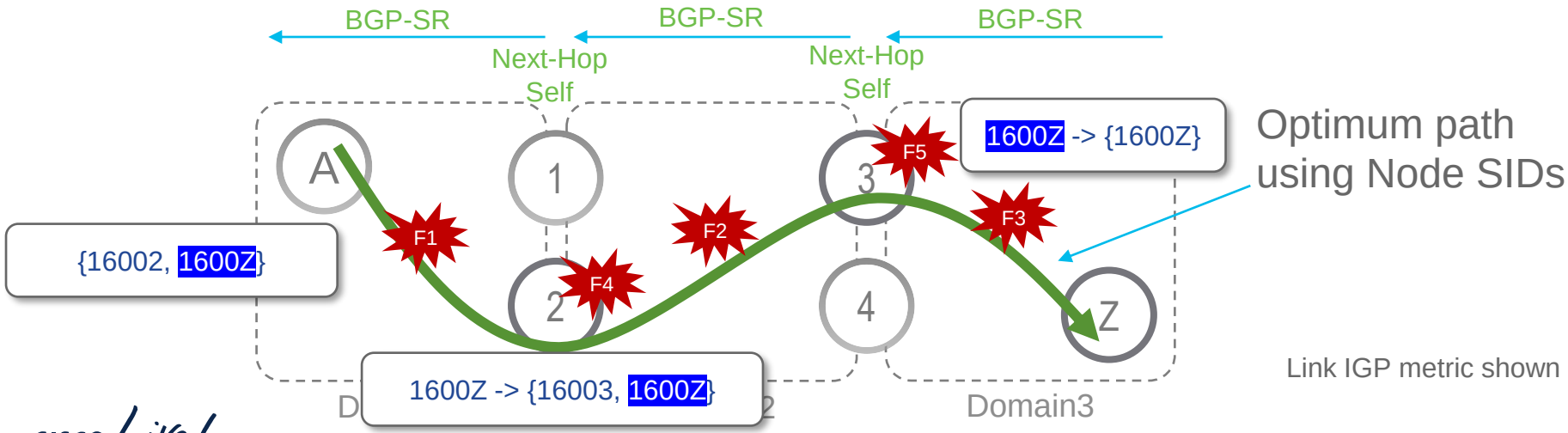
Multi-Domain – BGP-SR

- Failures F1, F2, F3
  - TI-LFA FRR

Fast convergence

- Failures F4, F5
  - IGP + BGP PIC convergence

Slower convergence



Optimum path using Node SIDs

Link IGP metric shown

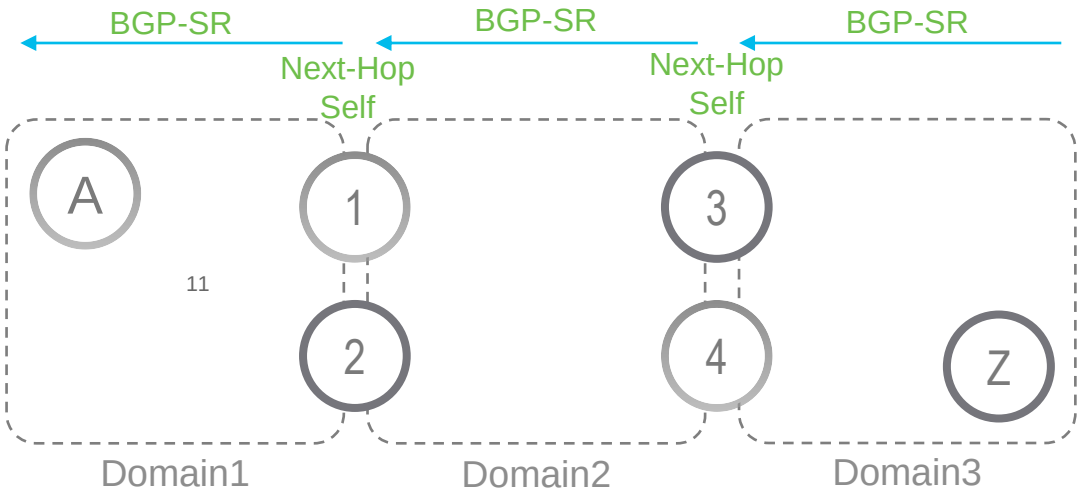
With BGP-SR and Weighted Anycast NH

1.1.1.101 - 16101 = Anycast SID in R1 and R2
low igp prefix metric on 1
high igp prefix metric on 2

1.1.1.102 - 16102 = Anycast SID in R1 and R2
low igp prefix metric on 2
high igp prefix metric on 1

1.1.1.203 - 16203 = Anycast SID in R3 and R4
low igp prefix metric on 3
high igp prefix metric on 4

1.1.1.204 - 16204 = Anycast SID in R3 and R4
low igp prefix metric on 3
high igp prefix metric on 4



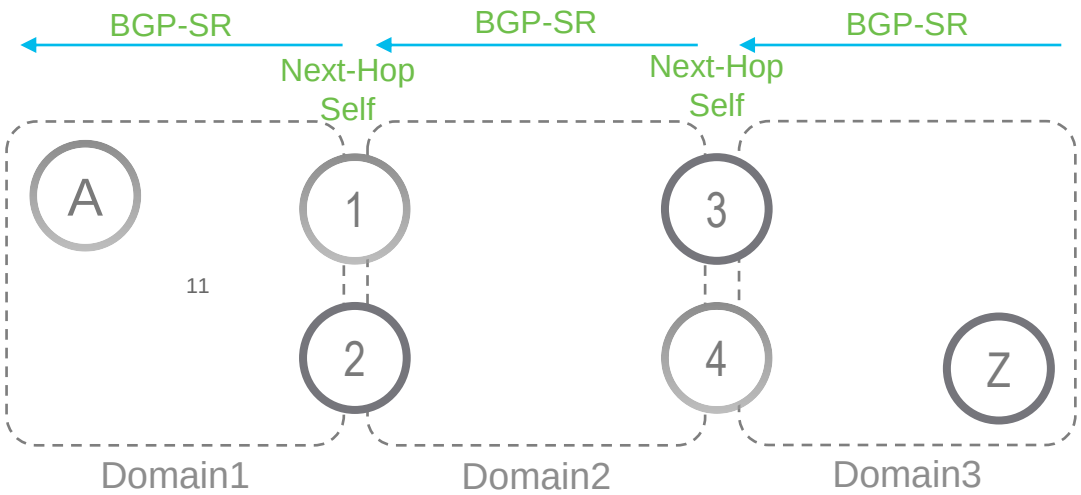
Link IGP metric shown

With BGP-SR and Weighted Anycast NH

1.1.1.101 - 16101 = Anycast SID in R1 and R2
low igp prefix metric on 1
high igp prefix metric on 2

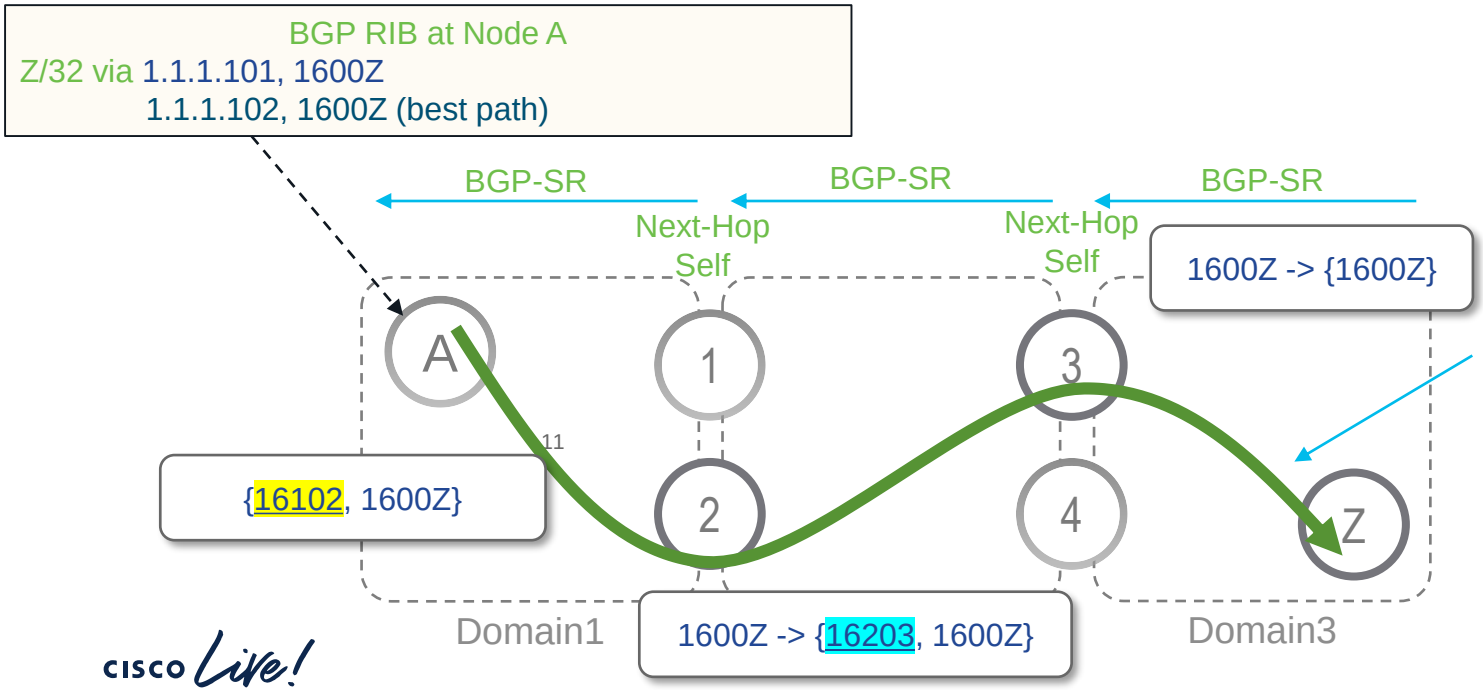
1.1.1.203 - 16203 = Anycast SID in R3 and R4
low igp prefix metric on 3
high igp prefix metric on 4

Weighted Anycast SIDs are used as BGP NH for BGP-SR



Link IGP metric shown

With BGP-SR and Weighted Anycast NH



Optimum and Resilient path using weighted Anycast SIDs

Link IGP metric shown

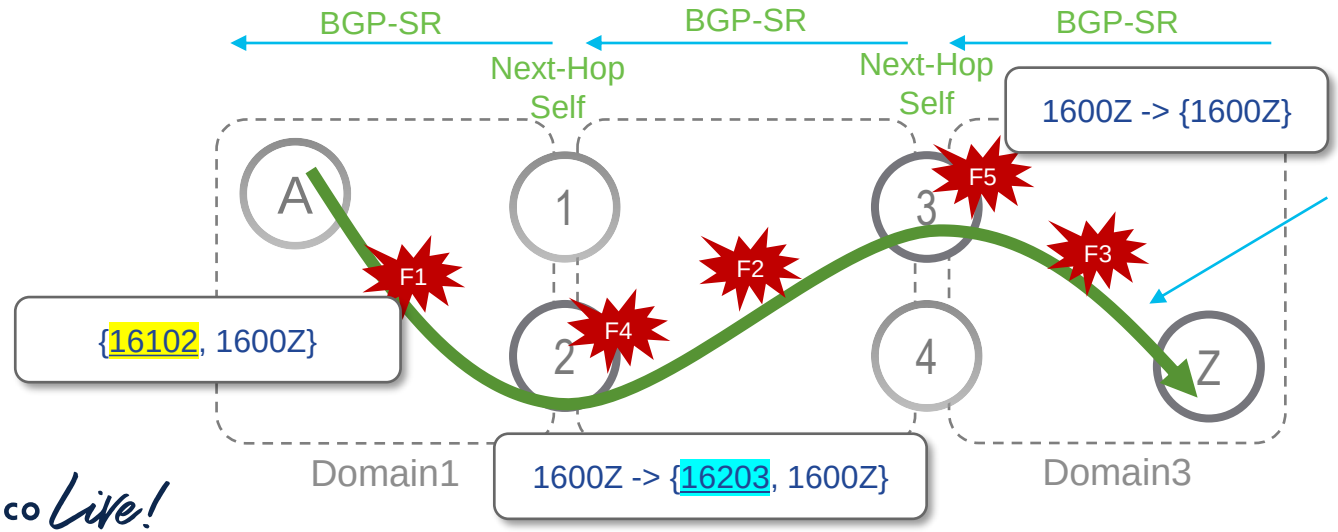
With BGP-SR and Weighted Anycast NH

- Failures F1, F2, F3
  - TI-LFA FRR

Fast convergence

- Failures F4, F5
  - TI-LFA FRR

Fast convergence

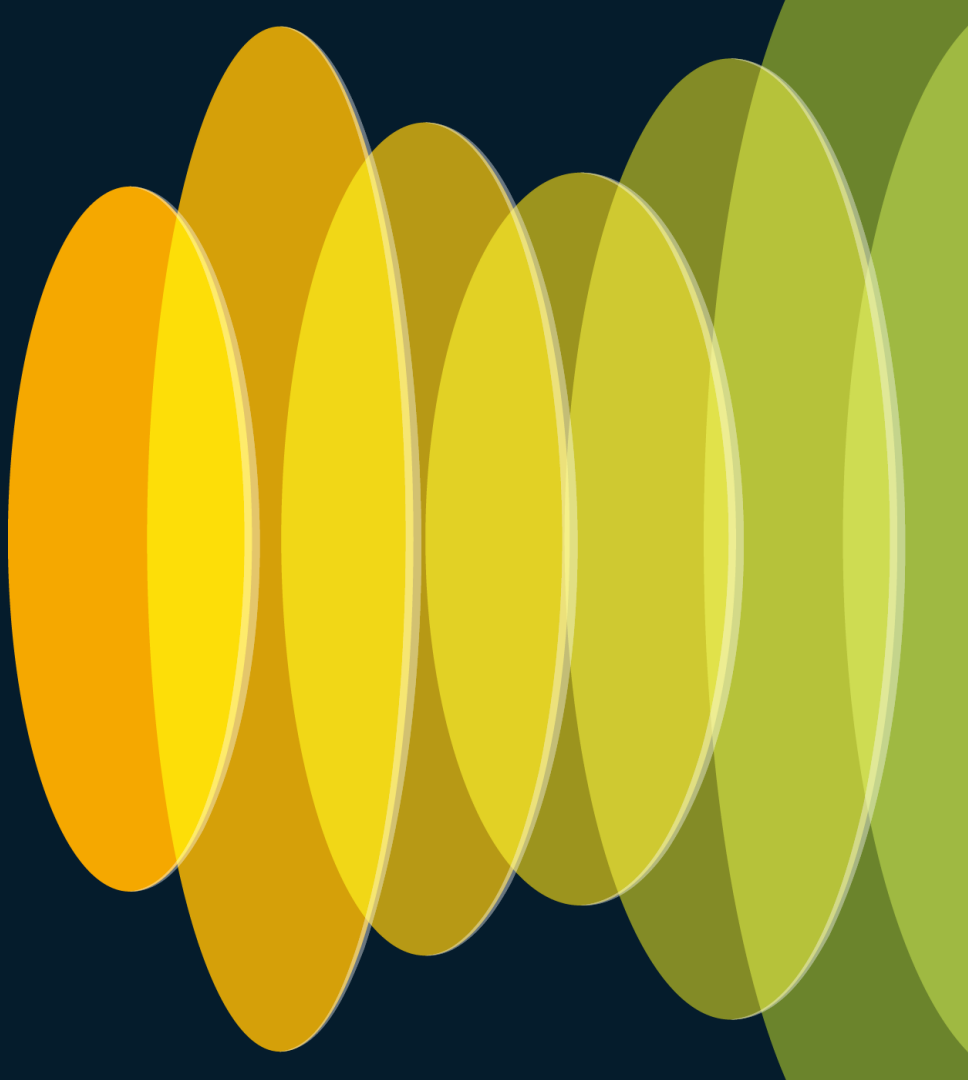


Optimum and Resilient path using weighted Anycast SIDs

Link IGP metric shown

SR Traffic Engineering

Intent-based Transport



cisco Live!

What is Transport intent?

- Topology path selection
  - Minimize cost per bit
  - Minimize expected delay
  - Minimize delay
  - Avoid resource
  - Disjoint paths
  - Disjoint planes
  - Data Sovereignty
  - Minimize carbon footprint
- Others
  - Steer traffic along a service chain
- Any combination of the above

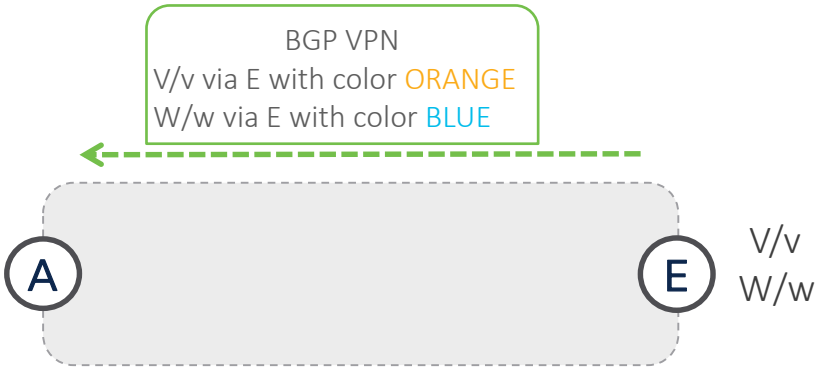
Intent encoded as a color

- Color is a standard way to signal intent
  - A 32-bit number

- Mapping an intent to a color:
  - Low-latency: BLUE
  - Low-cost: ORANGE

- Colored Service Routes – requesting a particular intent
- Color-aware Transport Routes – satisfying a particular intent
- A colored service route is steered over a color-aware route of same intent

Colored service route

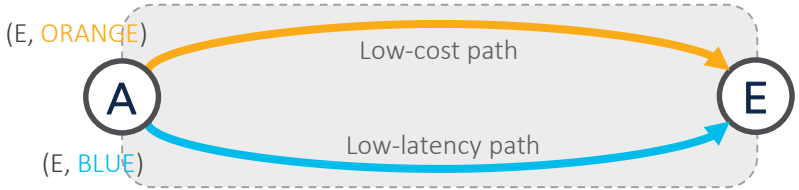


- E sends colored service routes, each requesting a particular intent
- Route coloring is done by using the BGP Color Extended-Community
  - Standard (RFC5512 / RFC9012), supported by all major BGP implementations
- Any service route can be colored (L3VPN, EVPN, Internet routes)

Color-aware transport routes

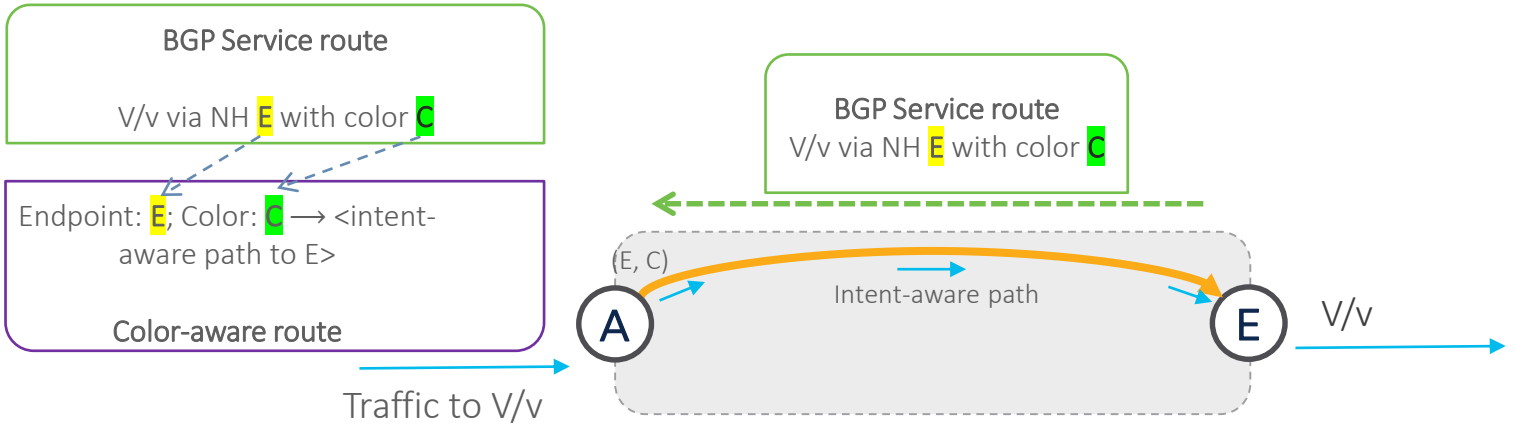
Color-aware routes @ A:

Endpoint	Color	Path
E	ORANGE	<low-cost path A to E>
E	BLUE	<low-latency path A to E>



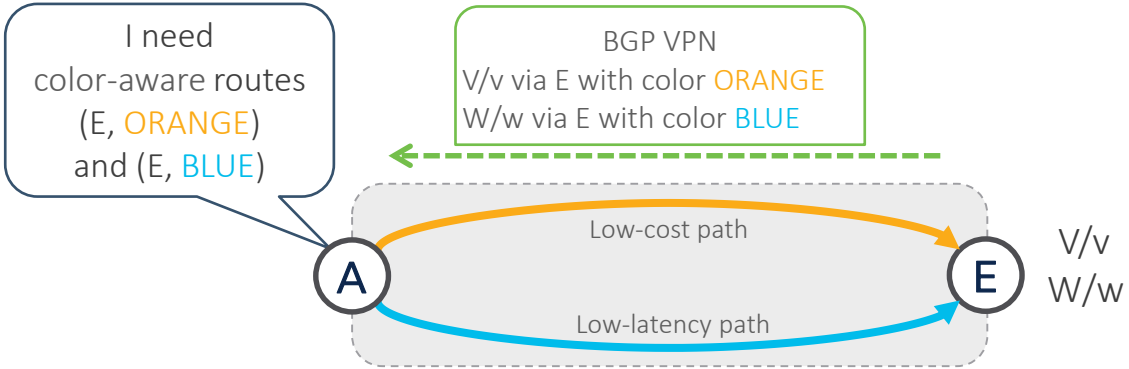
- A color-aware route satisfies a particular intent
- A color-aware route is identified by the tuple (Endpoint and Color)
  - in short (E,C)
- A color-aware route can be signaled/instantiated by different mechanisms

Service routes steered on color-aware route



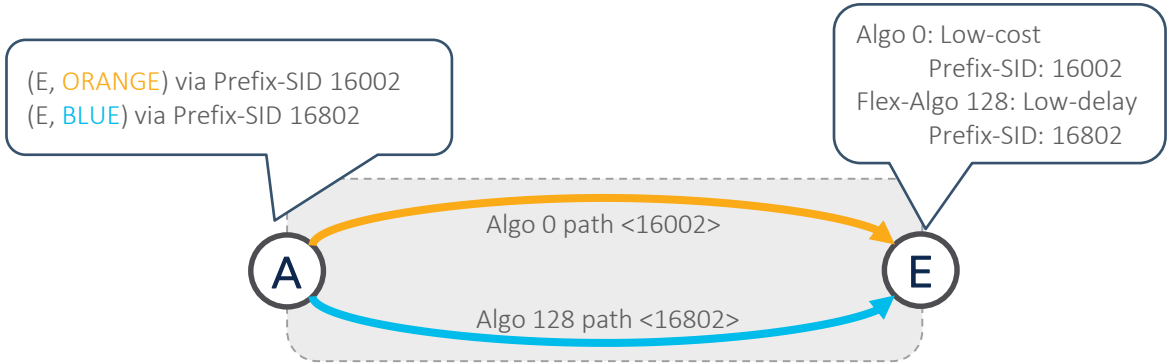
- A colored service route is steered over a color-aware route of same intent
  - Traffic destined to prefix V/v via **E** with color **C** is steered over color-aware route (**E**, **C**)
  - This is known as Automated Steering (AS)

Service routes steered on color-aware route



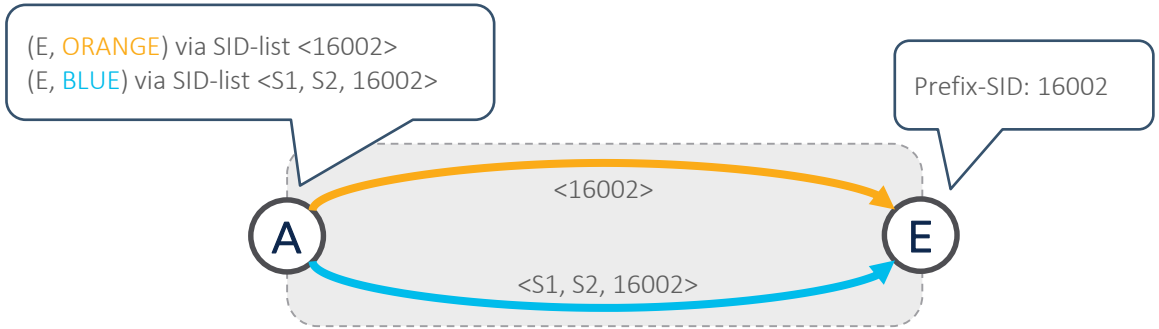
- A colored service route is steered over a color-aware route of same intent
  - V/v via E with color ORANGE is steered over (E, ORANGE)
  - W/w via E with color BLUE is steered over (E, BLUE)

Color-aware route (E, C) provided by IGP Flex Algo



- IGP Flex Algo (RFC9350)
- A maps color to an IGP algorithm
  - Orange ⟶ Algo 0
  - Blue ⟶ Flex-Algo 128

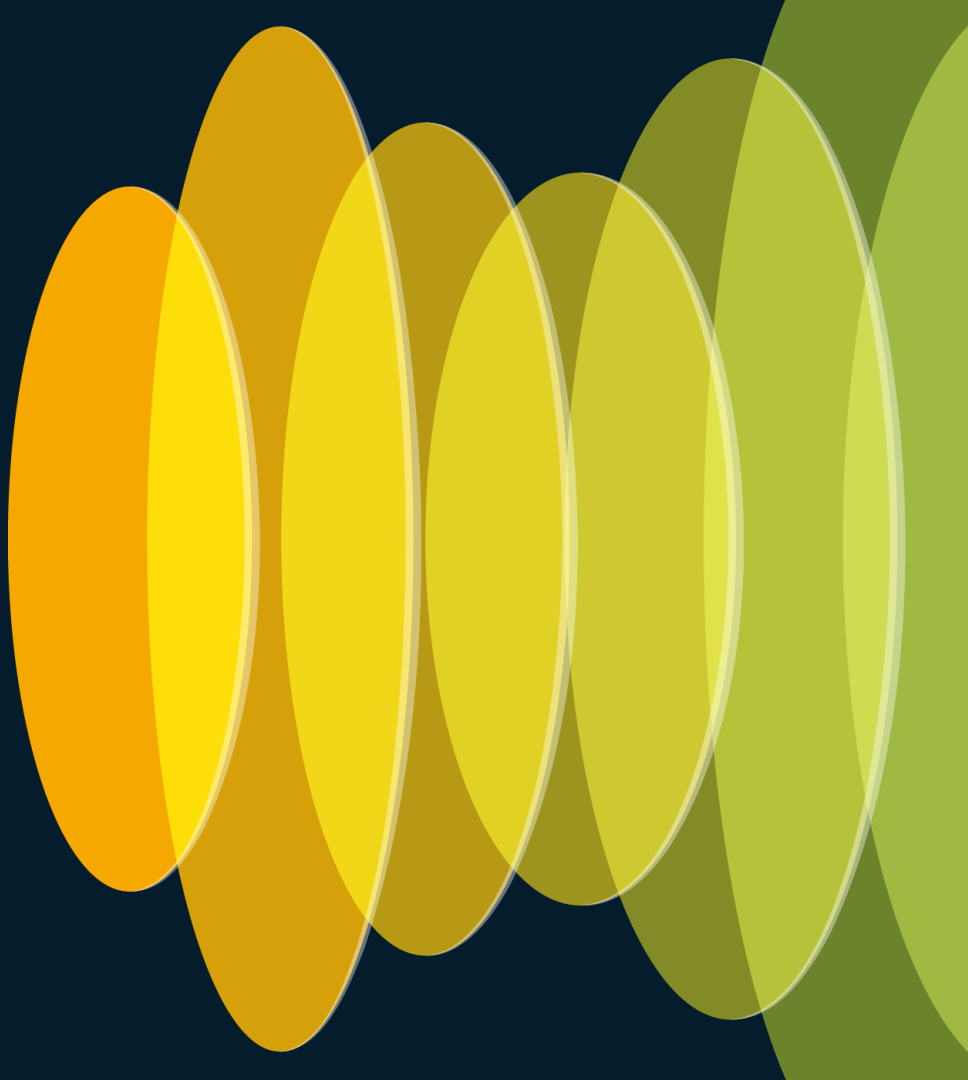
Color-aware route (E, C) provided by SR Policy



- SR Traffic Engineering Policy – in short SR Policy (RFC9256)
- A has two SR policies
  - (E, ORANGE): Dynamic, low cost with SID-list <16002>
  - (E, BLUE): Dynamic, low delay with SID-list <S1, S2, 16002>

Intent-based Transport

SR Policy Fundamentals



cisco Live!

SR Traffic Engineering

- Simple, Automated and Scalable
  - No state in the core – state in the packet header
  - No tunnel interface – SR Policy
    - Uniquely identified by a tuple (head-end, color, end-point)
    - Where the Color provides the hint of Transport Intent
  - No head-end a-priori configuration – on-demand SR policy instantiation
  - No head-end a-priori steering – automated steering
- Multi-Domain
  - SR PCE for compute
  - Binding-SID (BSID) for scale
- Rich Functionality designed with lead operators along their use-cases

Key IETF document for SR Policy

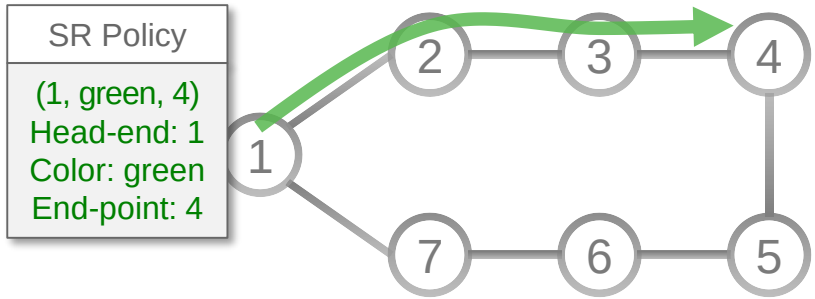
Internet Engineering Task Force (IETF)
Request for Comments: 9256
Updates: 8402
Category: Standards Track
ISSN: 2070-1721

C. Filsfils
K. Talaulikar, Ed.
Cisco Systems, Inc.
D. Voyer
Bell Canada
A. Bogdanov
British Telecom
P. Mattes
Microsoft
July 2022

Segment Routing Policy Architecture

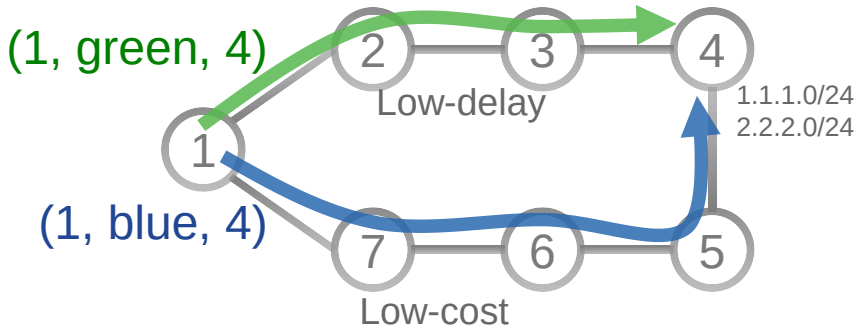
SR Policy Identification

- An SR Policy is uniquely identified by a tuple (head-end, color, end-point)
  - Head-end: where the SR Policy is instantiated (*implemented*)
  - Color: a numerical value to differentiate multiple SRTE Policies between the same pair of nodes
  - End-point: the destination of the SR Policy
- At a given head-end, an SR Policy is uniquely identified by a tuple (color, end-point)



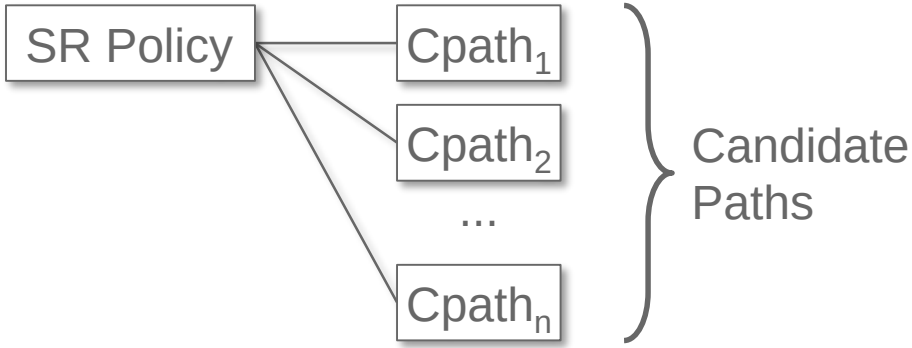
SR Policy Color

- 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)



SR Policy – Candidate Paths

- An SR Policy consists of one or more candidate paths (Cpaths)

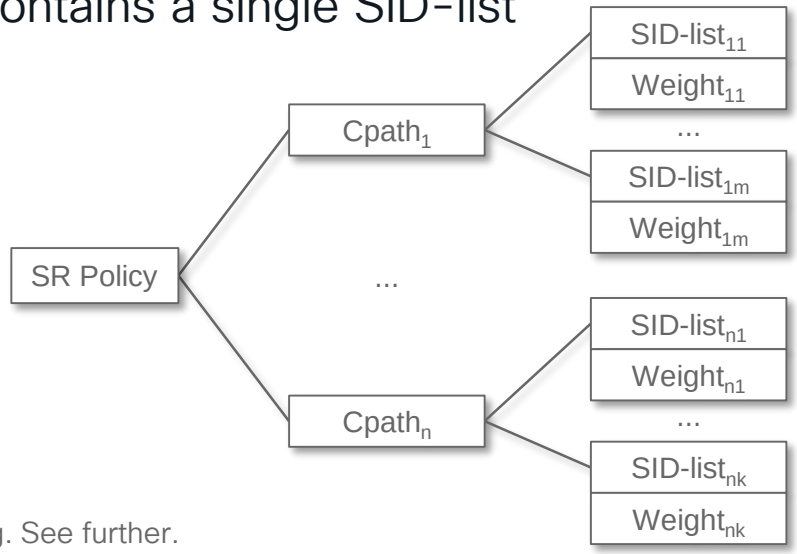


- An SR Policy instantiates one single path in RIB/FIB
  - the selected* path, which is the preferred valid candidate path
- A candidate path is either dynamic or explicit

* See further.

SR Policy – Candidate Path

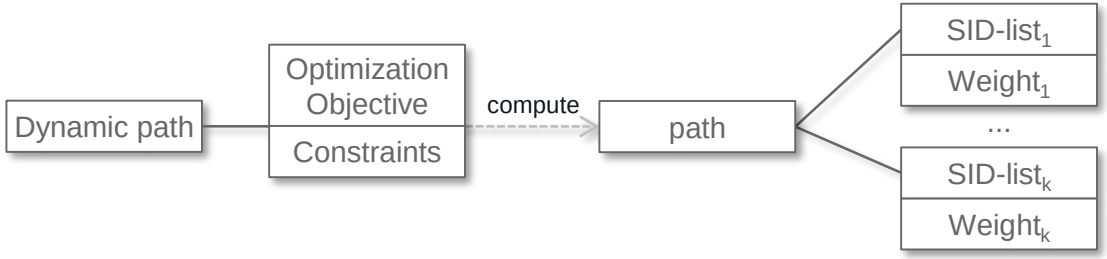
- A candidate path is a single segment list (SID-list) or a set of weighted* SID-lists
  - Typically, an SR Policy path only contains a single SID-list
- Traffic steered into an SR Policy path is load-shared over all SID-lists of the path



SID = Segment ID

*For Weighted Equal Cost Multi-Path (WECMP) load-sharing. See further.

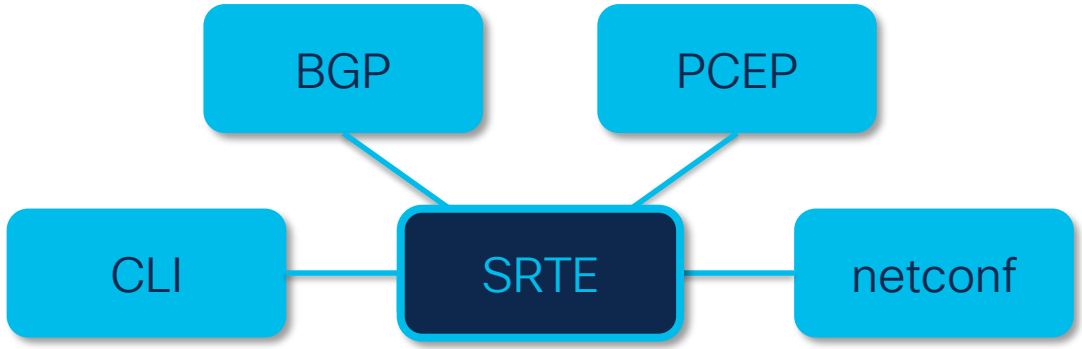
Dynamic Path



- A dynamic path expresses an optimization objective and a set of constraints
- The head-end computes a solution to the optimization problem as a SID-list or a set of SID-lists
- When the head-end does not have enough topological information (e.g. multi-domain problem), the head-end may delegate the computation to a PCE
- Whenever the network situation changes, the path is recomputed

Dynamic Paths (cont.)

- A head-end may be informed about candidate paths for an SR Policy by various means including
  - local configuration (CLI), netconf, PCEP, or BGP-TE

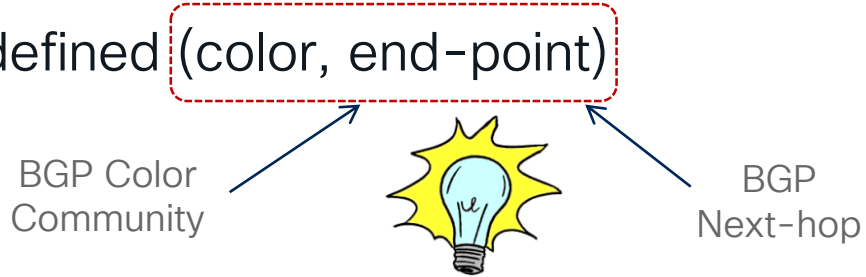


Dynamic Paths (cont.)

- A head-end may also automatically trigger a candidate path of an SR Policy based on the arrival of a colored service route
- We call this On-Demand Next-Hop (ODN)

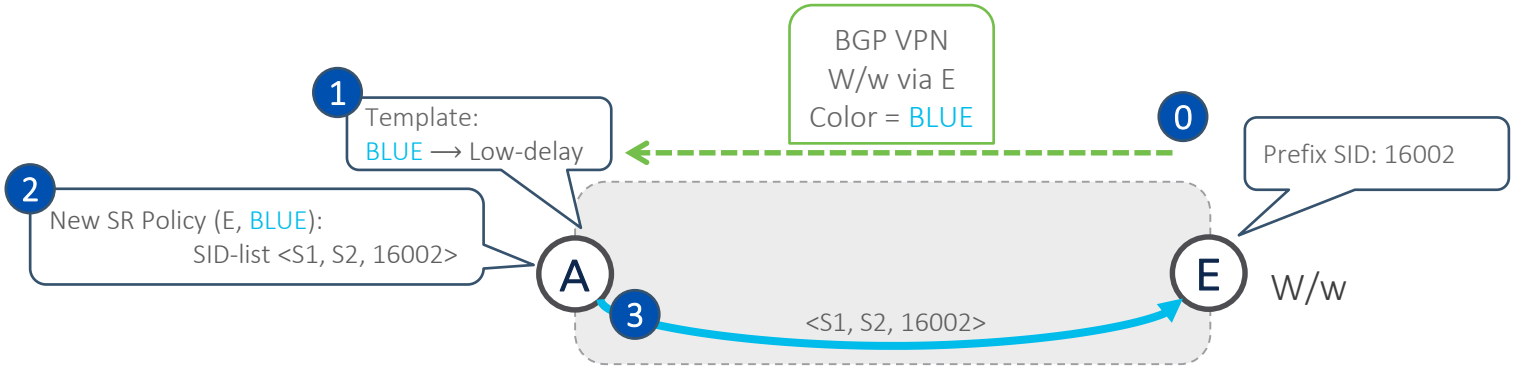
On-Demand Next-hop (ODN)

- A service head-end automatically instantiates an SR Policy to a BGP Next-hop when required (on-demand)
- Color community is used as Transport SLA intent / indicator
- Reminder: an SR Policy is defined (color, end-point)



- Automated Steering (AS) automatically steers the BGP traffic into this SR Policy, also based on Next-hop and color

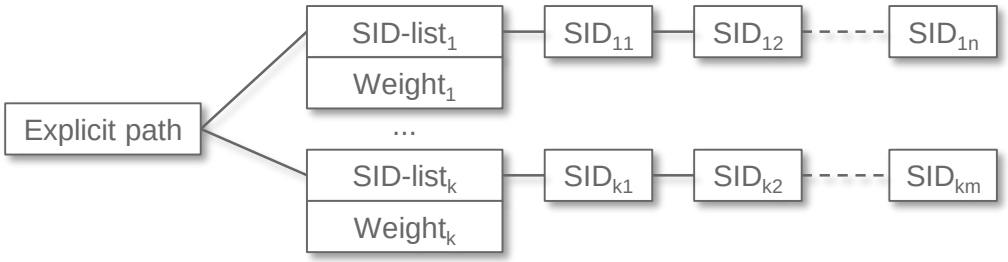
On-Demand Next-hop (ODN)



1. A maps color BLUE to the low-delay intent
2. Upon receiving a service route via E with color BLUE, A automatically instantiates the SR Policy (E, BLUE)
   - This is called On-Demand Next-hop (ODN)
   - Each PE installs only the SR Policies that it needs
3. A automatically steers the traffic for prefix W/w onto SR Policy (E, BLUE)

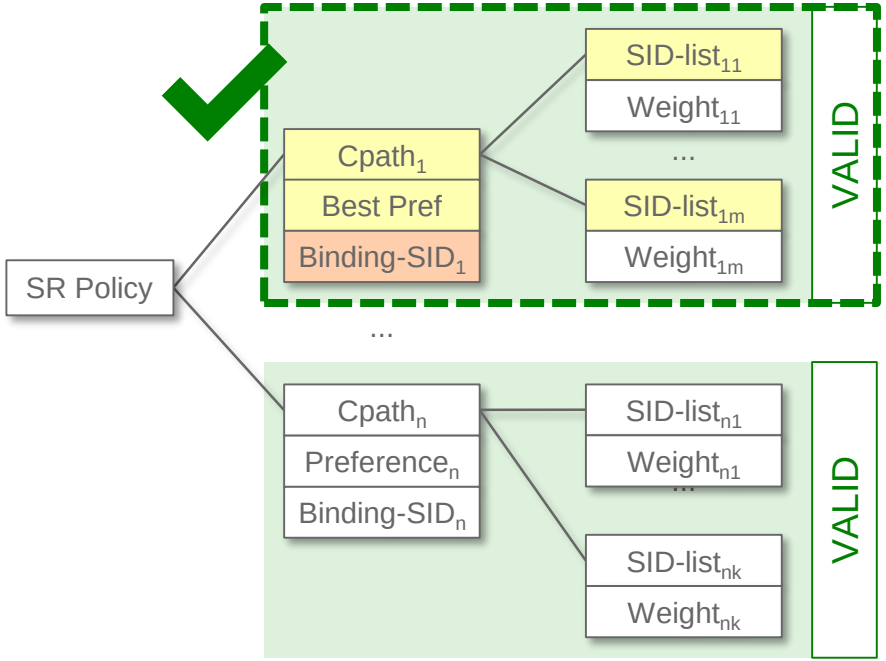
Explicit Path

- An explicit path is an explicitly specified SID-list or set of SID-lists



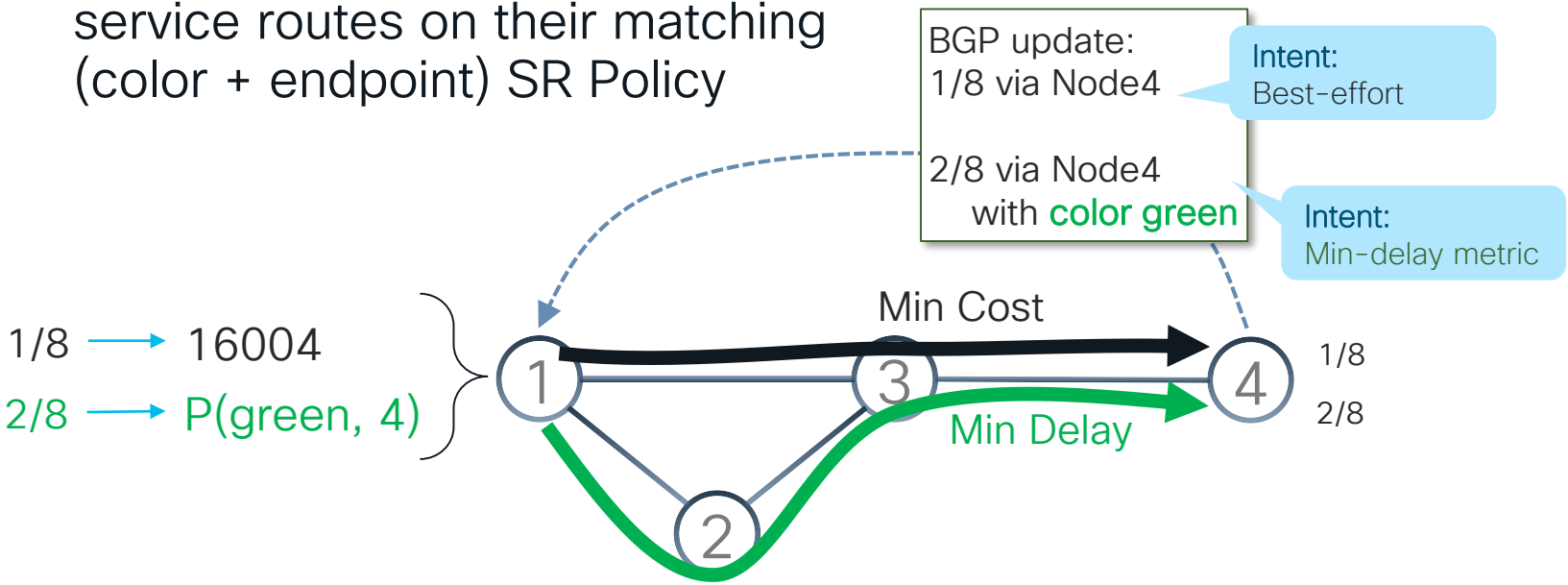
Binding-SID (BSID) of an SR Policy

- The BSID of an SR Policy is the BSID of the selected candidate path



Per-Destination Automated Steering

- Automated Steering steers service routes on their matching (color + endpoint) SR Policy



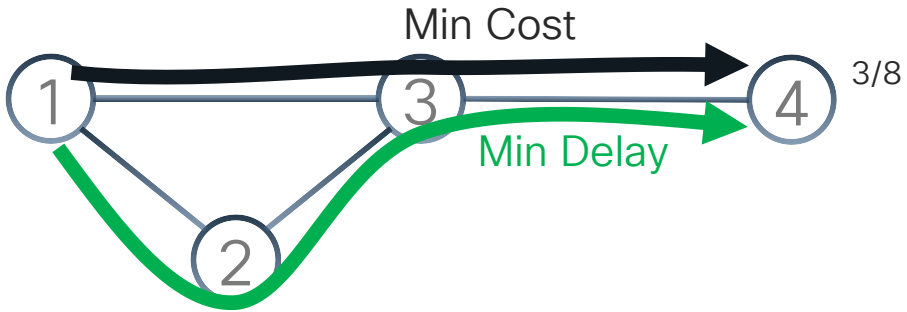
Need for Per-Flow Automated Steering

Same Destination

F1: 3/8 (DSCP 0) → IGP to 4

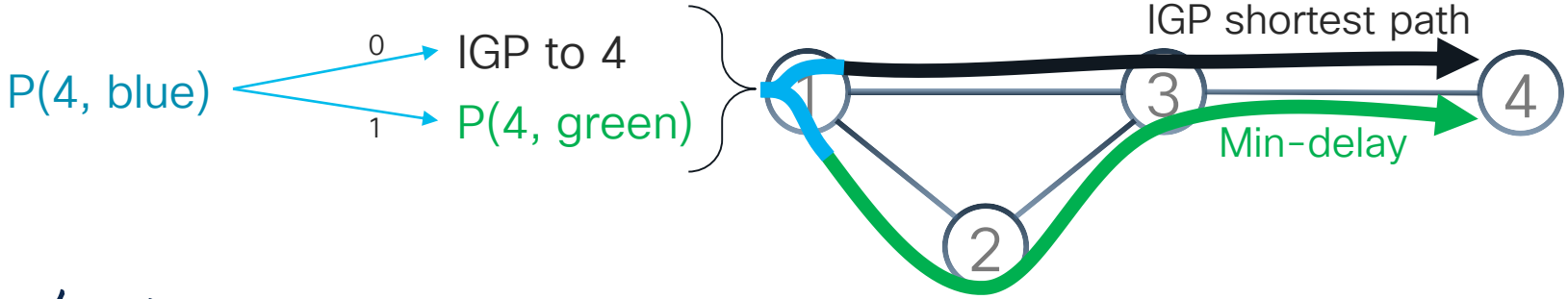
F2: 3/8 (DSCP 46) → P(green, 4)

Different Flows (flow 1, flow 2)



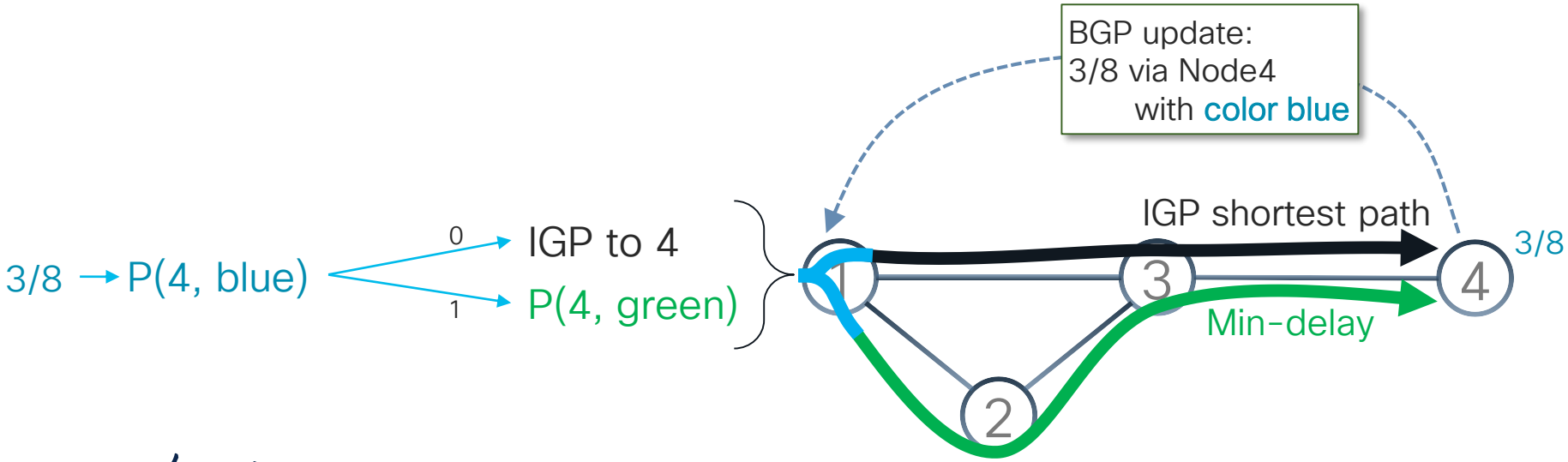
Per-Flow SR Policy (PFP)

- Per-Flow Policy (Node4, blue) @ Node1
  - FC=0 ➔ IGP shortest path == 16004
  - FC=1 ➔ Per-Destination SR Policy (Node4, green)
- Per-Destination Policy (Node4, green)
  - Defined as Min Delay ➔ <16002, 16004>

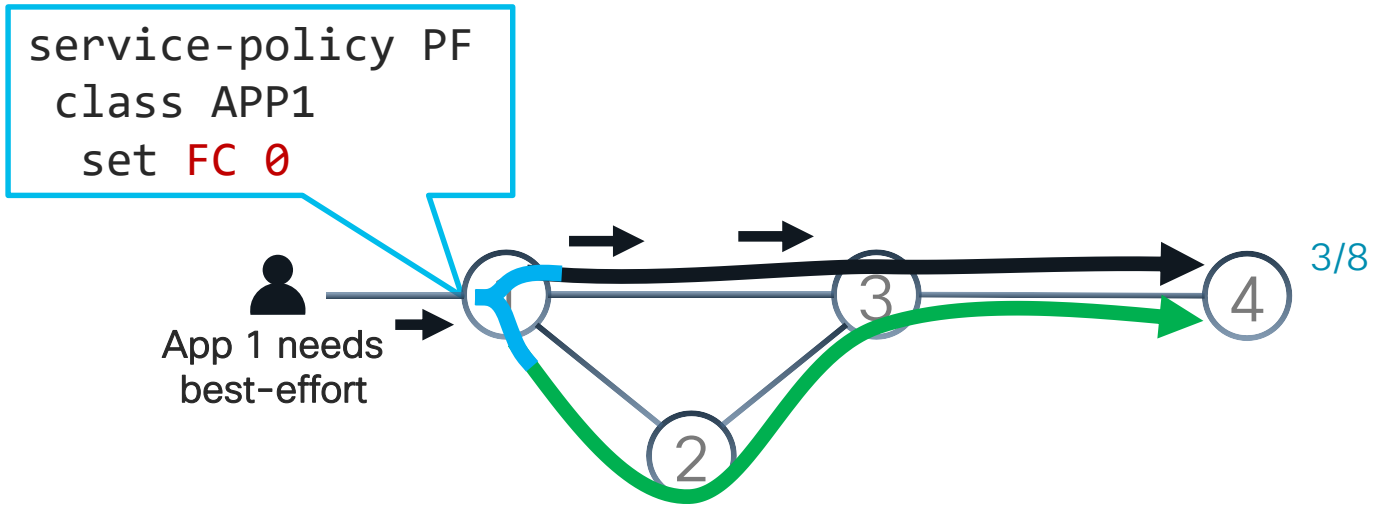


Per-Flow Automated Steering (AS)

- AS automatically steers a service route on the SR policy (E, C):
  - E == Next-hop
  - C == color of the service route

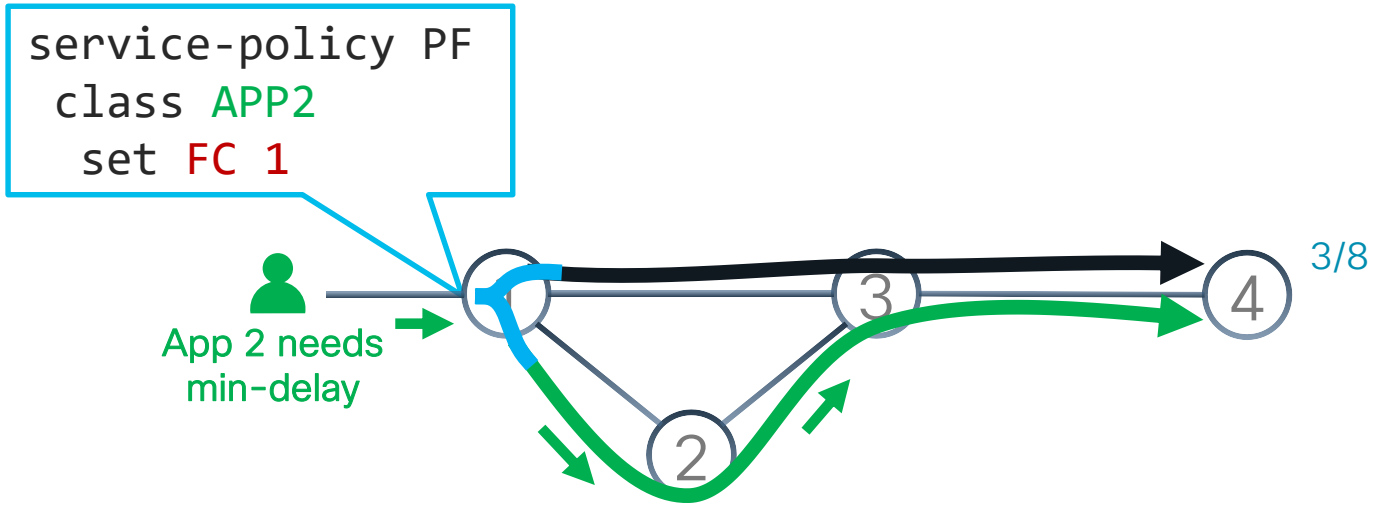


App needs best-effort



- PE Node1 classifies App 1 flow packets in FC 0
- Automated Steering steers 3/8 in SR Policy P(Node4, blue)
- Flow is switched on P(Node4, blue)'s FC 0 path:
  IGP shortest path to Node4

App needs min-delay

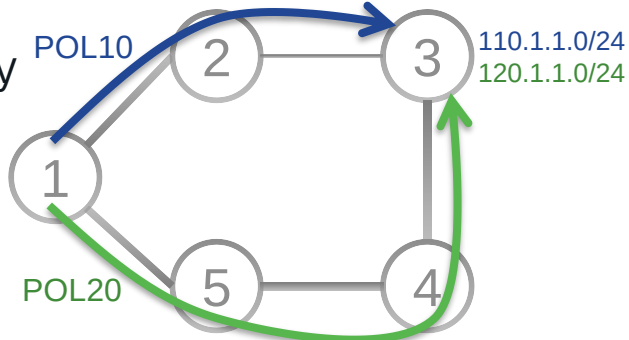


- PE Node1 classifies App 2 flow packets in FC 1
- Automated Steering steers 3/8 in SR Policy P(Node4, blue)
- Flow is switched on P(Node4, blue)'s FC 1 path: min-delay path to Node4

Automated steering

- BGP can automatically steer traffic into an SR Policy based on BGP Next-hop and color of a route
  - color of a route is specified by its color extended community attribute
- By default:
  If the BGP Next-hop and color 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
  - end-point and color uniquely identify an SR Policy on a given head-end

110.1.1.0/24 (color 10, NH 1.1.1.3)
via SR Policy POL10 (10, 1.1.1.3)
120.1.1.0/24 (color 20, NH 1.1.1.3)
via SR Policy POL20 (10, 1.1.1.3)



Steering – Color-only (CO) bits

- Assume route R with Next-hop N has a single color C
- The Color-Only (CO) bits in the color extended community attribute flags of R are 00, 01, or 10 (11 is treated as 00)
- BGP steers R according to this preference order:

CO=00 (or CO=11)

Preference:
1. SR Policy(N, C)
2. IGP to N

CO=01

Preference:
1. SR Policy(N, C)
2. SR Policy(null(AF_N), C)
3. SR Policy(null(any), C)
4. IGP to N

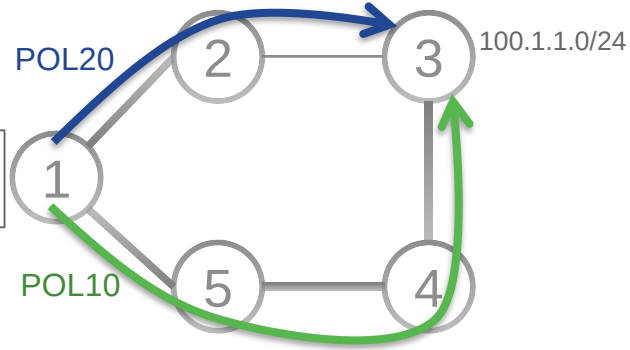
CO=10

Preference:
1. SR Policy(N, C)
2. SR Policy(null(AF_N), C)
3. SR Policy(null(any), C)
4. SR Policy(<any(AF_N)>, C)
5. SR Policy(<any(any)>, C)
6. IGP to N

Route has multiple colors

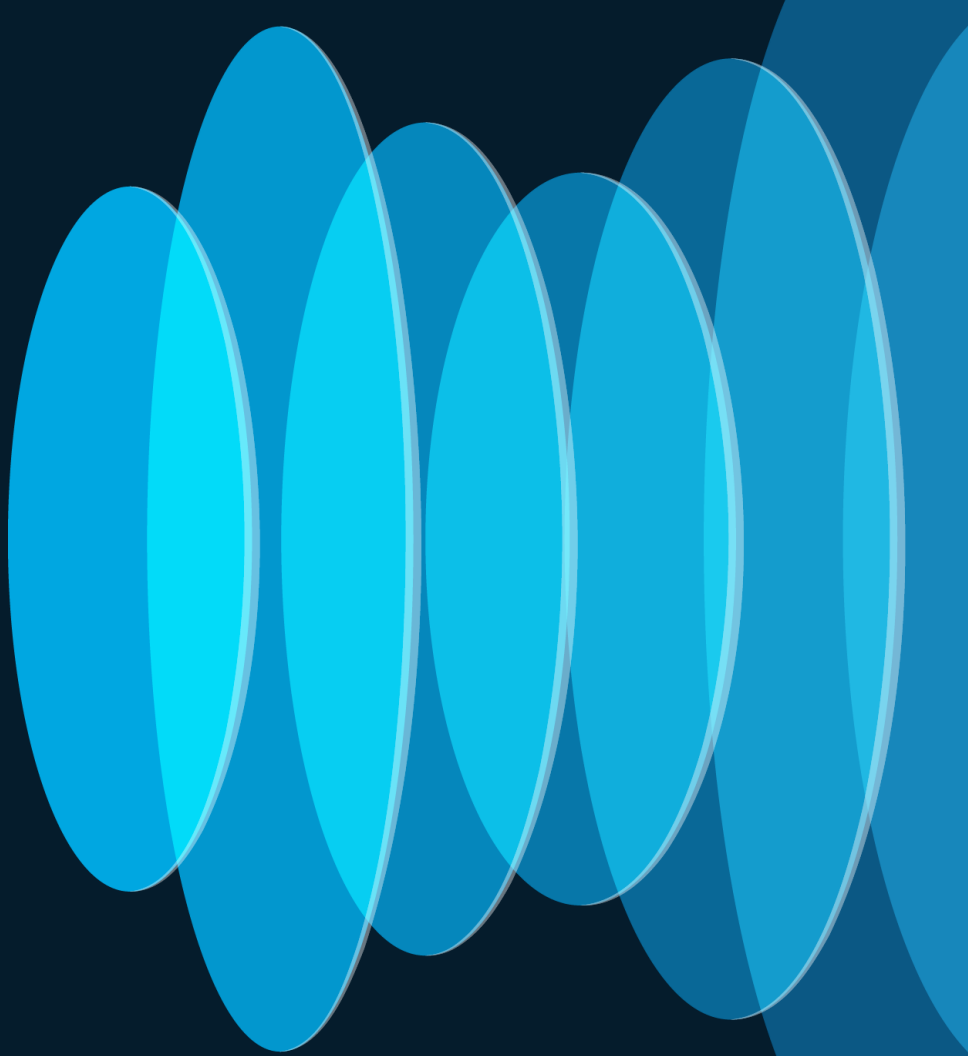
- If a route R with Next-hop N has multiple colors C_1 ... C_k then BGP steers R into the SR Policy with the numerically highest color
  - Considering only valid and authorized-to-steer SR Policies (C_i,N) with i=1...k
- Example:
  - Node1 receives 100.1.1.0/24 with NH 1.1.1.3 and colors 10 and 20
  - BGP resolves 100.1.1.0/24 on BSID of POL20 (has numerically highest color 20)

100.1.1.0/24 (NH 1.1.1.3; color 20, color 10)
via SR Policy POL20 (20, 1.1.1.3)



Intent-based Transport

SR Policy with SR-MPLS



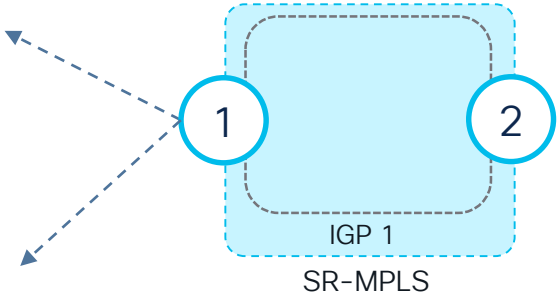
cisco Live!

SRTE – TED @ Head-end

SR-TE HE (node 1)

```
router isis 100
 distribute link-state instance-id 100
```

```
router ospf 100
 distribute link-state instance-id 100
```



SRTE – ODN Template

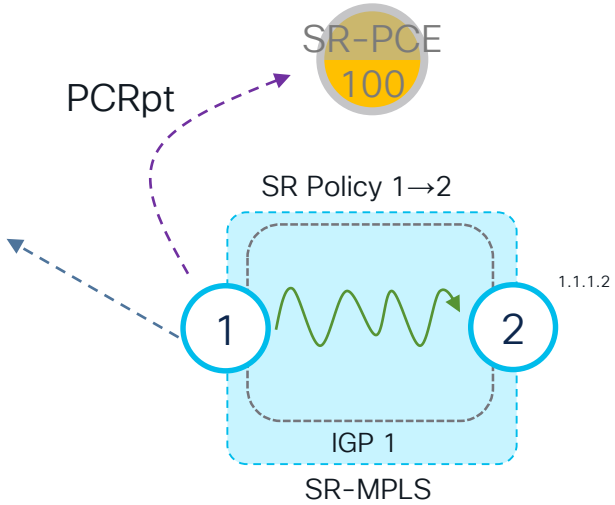
SR-TE HE (node 1)

```
segment-routing
 traffic-eng
  on-demand color 10
   dynamic
    metric
     type [igp | te | latency | hopcount]
```

SRTE - HE computed example

SR-TE HE (node 1)

```
segment-routing
 traffic-eng
  policy pol-C_1-EP_2
   color 1 end-point ipv4 1.1.1.2
   candidate-paths
    preference 100
     dynamic
      metric
       type [igp | te | latency | hopcount]
```

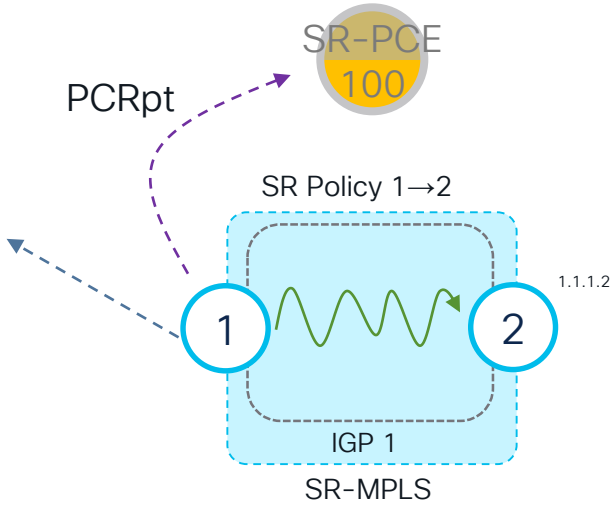


PCRpt = PCEP Report message

SRTE - HE computed example

SR-TE HE (node 1)

```
segment-routing
 traffic-eng
  policy pol-C_1-EP_2
   color 1 end-point ipv4 1.1.1.2
   candidate-paths
    preference 100
     dynamic
      metric
       type [igp | te | latency | hopcount]
      !
     !
     constraints
      affinity
       exclude-any
        name aff-brown
```



PCRpt = PCEP Report message

SRTE - Constraints

```
RP/0/RP0/CPU0:R1(config-sr-te-policy-path-pref)#constraints ?
  affinity       Assign affinities to path
  disjoint-path  Request disjoint path computation
  resources      Specify resources constraints for path computation
  segments       Path segments constraints

RP/0/RP0/CPU0:R1(config-sr-te-policy-path-pref)#constraints affinity ?
  exclude-any    Affinity attributes to exclude - presence of at least one
excludes link
  include-all    Affinity attributes - all must be included
  include-any    Affinity attributes - at least one must be included

RP/0/RP0/CPU0:R1(config-sr-te-policy-path-pref)#constraints resources ?
  exclude        Exclude resources from path computation

RP/0/RP0/CPU0:R1(config-sr-te-policy-path-pref)#constraints resources exclude ?
  resource-list  Exclude a resource-list from path computation
```

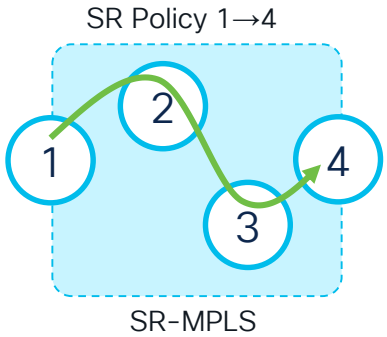
SRTE – Constraints (cont.)

```
RP/0/RP0/CPU0:R1(config-sr-te-policy-path-pref)#constraints segments ?
  adjacency-sid-only  Use only adjacency segment IDs
  protection          Protection type
  sid-algorithm       Prefix-SID algorithm

RP/0/RP0/CPU0:R1(config-sr-te-policy-path-pref)#constraints segments protection ?
  protected-only         Protected adj-SID only
  protected-preferred    Protected adj-SID preferred (default)
  unprotected-only       Unprotected adj-SID only
  unprotected-preferred  Unprotected adj-SID preferred
```

SRTE – SR Policy with Explicit Path

SR-TE HE (node 1)

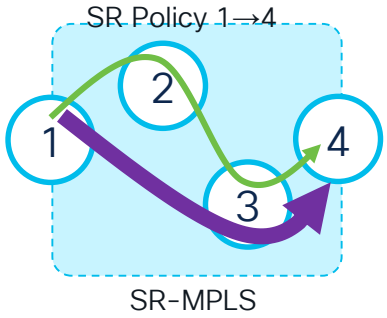
```
segment-routing
 traffic-eng
  segment-list sample-SL
   index 10 mpls adjacency 1.1.1.2
   index 20 mpls adjacency 1.1.1.3
   index 30 mpls adjacency 1.1.1.4
  !
  policy pol-C_10-EP_4
   color 10 end-point ipv4 1.1.1.4
   candidate-paths
    preference 100
     explicit segment-list sample-SL
```



SRTE – SR Policy with Explicit Path (multi-SL)

SR-TE HE (node 1)

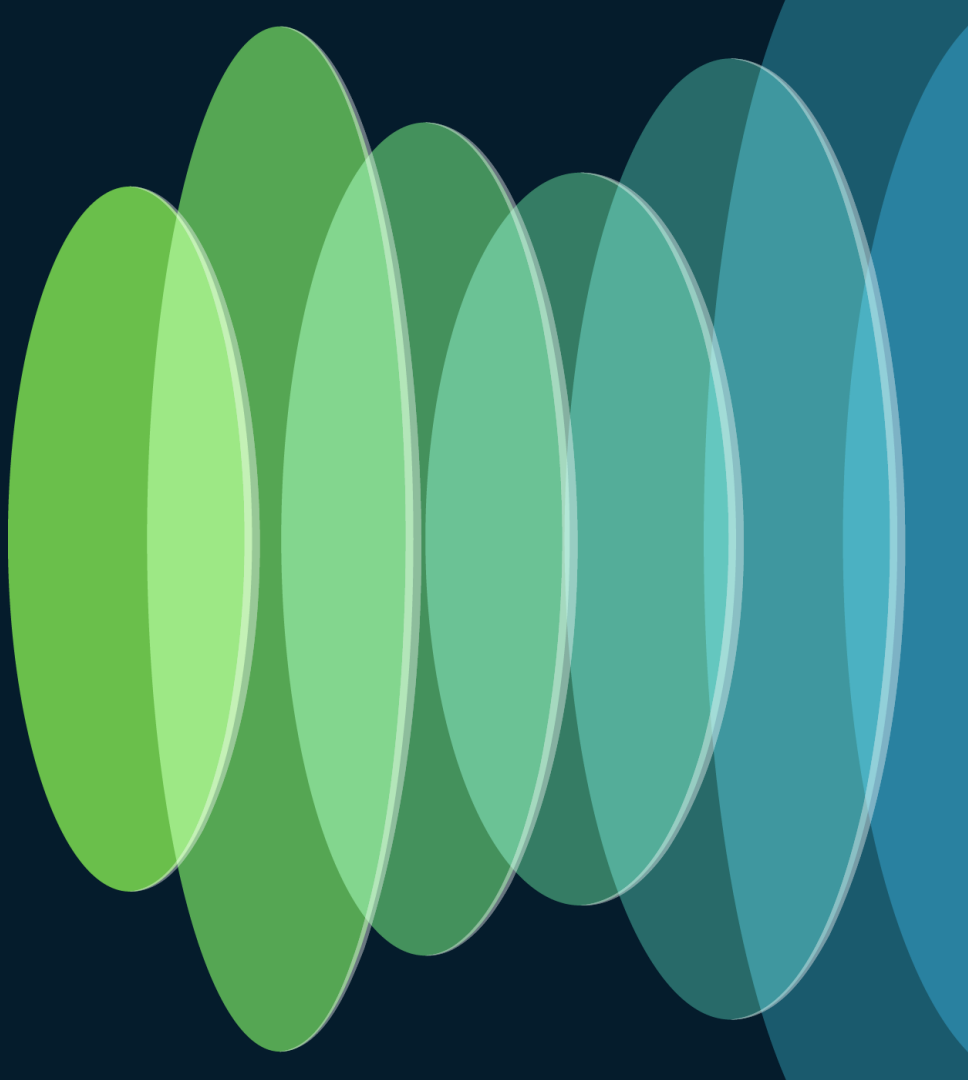
```
segment-routing
 traffic-eng
  segment-list sample-SL1
   index 10 mpls adjacency 1.1.1.2
   index 20 mpls adjacency 1.1.1.3
   index 30 mpls adjacency 1.1.1.4
  !
  segment-list sample-SL2
   index 20 mpls adjacency 1.1.1.3
   index 30 mpls adjacency 1.1.1.4
```




```
segment-routing
 traffic-eng
  policy pol-C_20-EP_4
   color 20 end-point ipv4 1.1.1.4
   candidate-paths
    preference 100
     explicit segment-list sample-SL1
      !
     explicit segment-list sample-SL2
      weight 3
```

Intent-based Transport

SR Policy with SRv6



cisco Live!

SRv6 SR Policy with explicit path

```
segment-routing
 traffic-eng
  segment-lists
   srv6
   segment-list STATIC_SID_LIST
    srv6
     index 20 sid fcbb:bb00:2::
     index 30 sid fcbb:bb00:3::
     index 40 sid fcbb:bb00:4:e001::

policy POLICY
  srv6
   locator MAIN binding-sid dynamic behavior ub6-insert-reduced
  color 10 end-point ipv6 fcbb:bb00:5::1
  candidate-paths
   preference 100
    explicit STATIC_SID_LIST
```

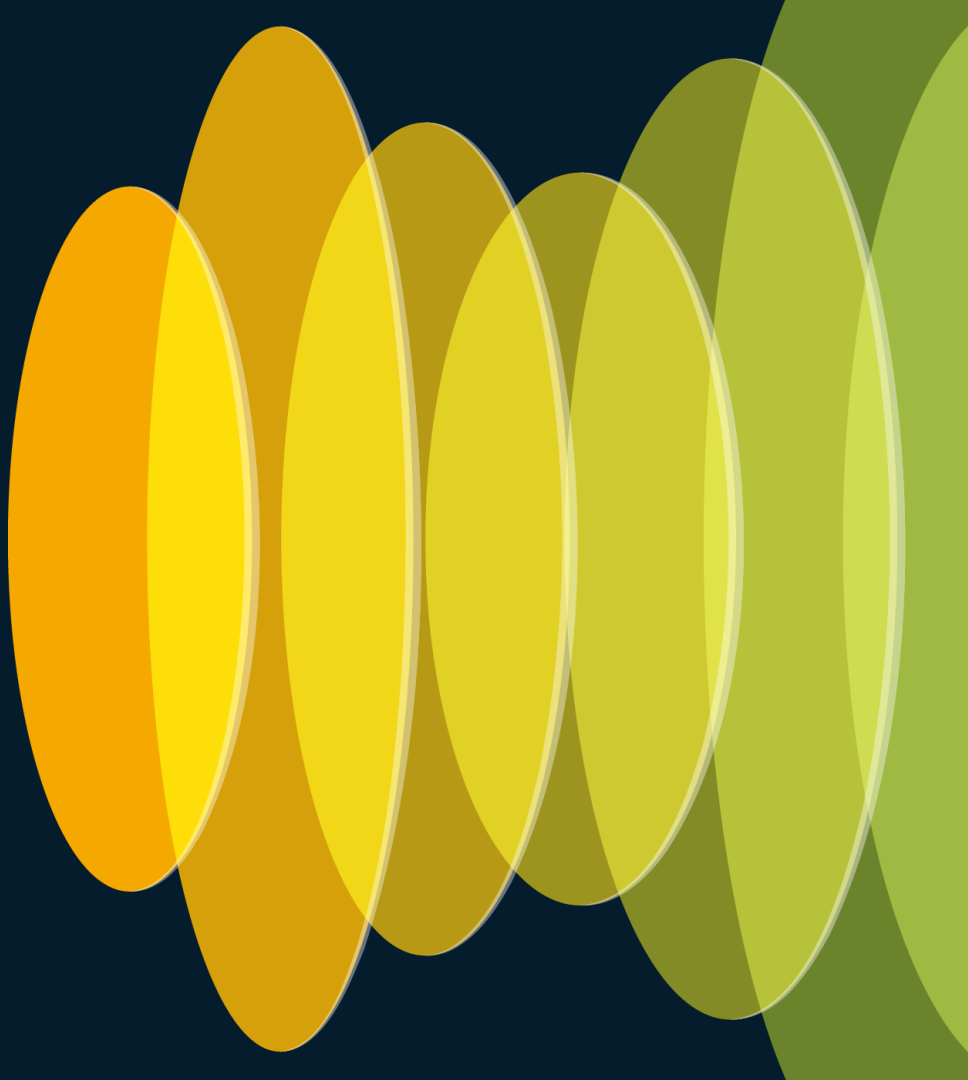
SID List definition

Locator for BSID

BGP Next Hop

Intent-based Transport

SR Path Computation Element (SR-PCE)



cisco Live!

SR Path Computation Element (SR-PCE)

SRTE Head-End

Distributed Mode – SR-TE Head-End
Visibility is limited to its own IGP domain

Solution

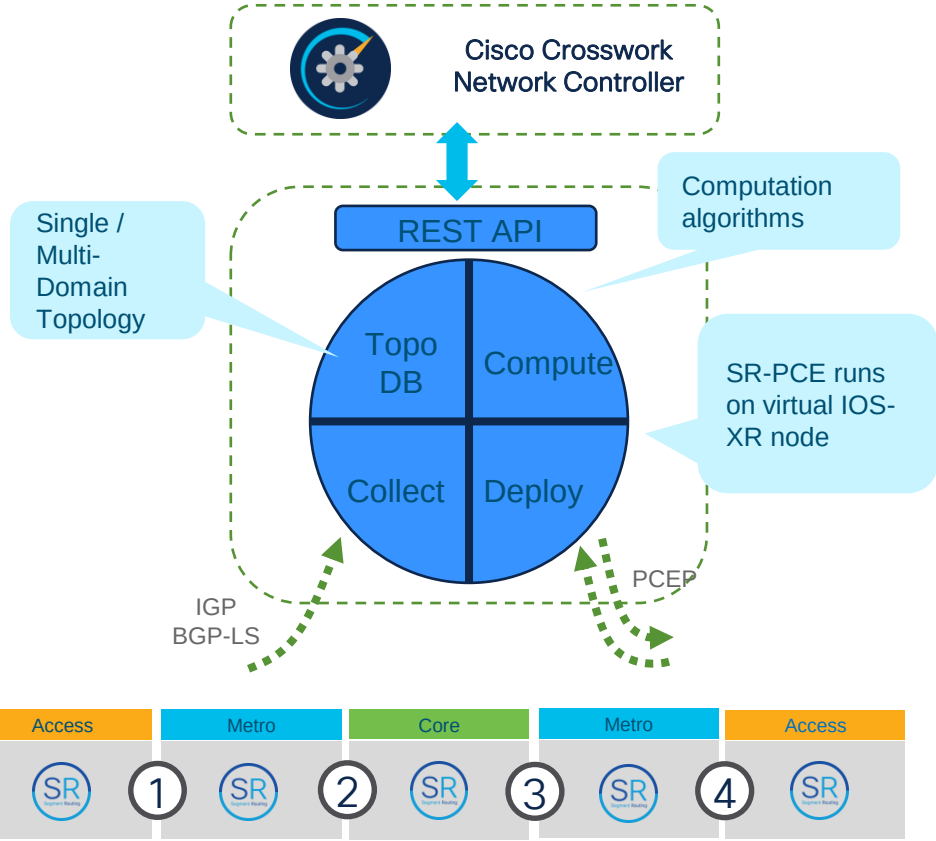
Multi-Domain SRTE Visibility
Centralized SR-PCE for Multi-Domain Topology view

Integration with Applications
North-bound APIs for topology/deployment

Delivers **across the unified SR Fabric** the SLA requested by the service

Benefits

Simplicity and Automation
End-to-End network topology awareness
SLA-aware path computation across network domains



SRTE Use Cases

Distributed or Centralized Path Computation?

Blue = computed by SR-PCE
Green = computed by head-end

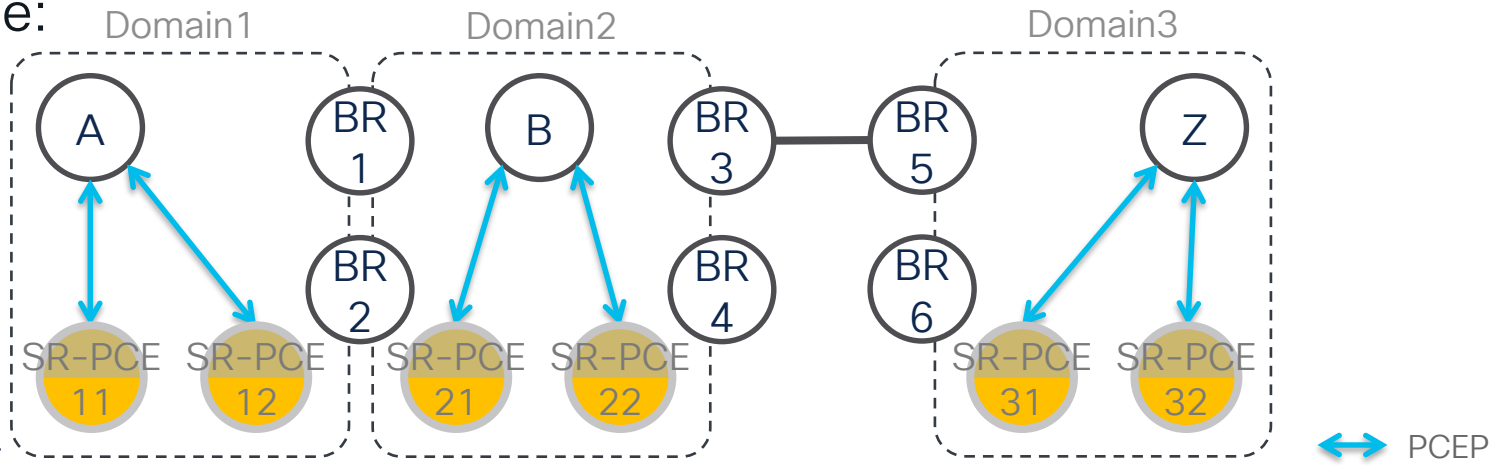
Use Case	Optimization objective / constraints	Single-Domain	Multi-Domain
Reachability	IGP metric + constraints	Distributed or Centralized	Centralized
Low Delay (TE metric)	TE metric + constraints	Distributed or Centralized	Centralized
Low Delay (measured)	Delay + constraints	Distributed or Centralized	Centralized
Path Disjointness	IGP / TE metric + PCEP association group	Centralized	Centralized
Tree-SID	P2MP	Centralized	Centralized

SR-PCE – Fundamentally Distributed

- SR-PCE not to be considered as a single all-overseeing device
- SR-PCE deployment is closer to BGP RR deployment model
- Different service end-points (PEs) can use different pairs of SR-PCEs
- Choice of SR-PCE can either be based on proximity or service-type

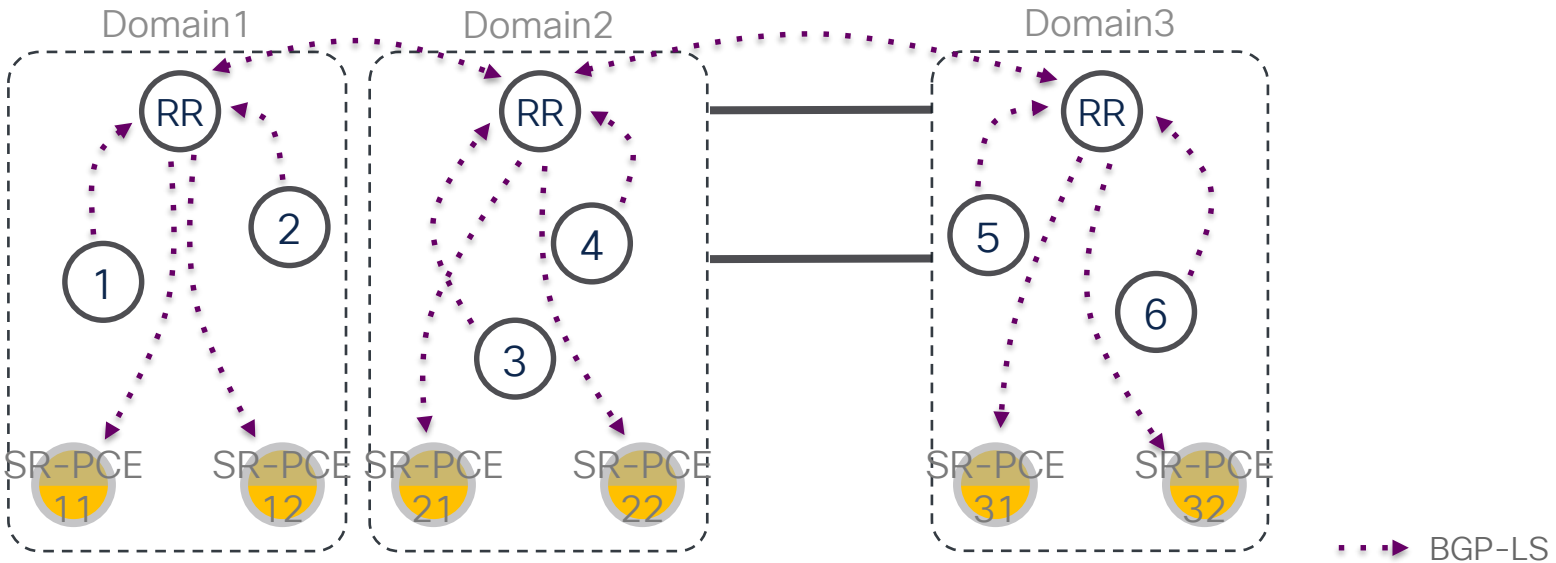
SR-PCE – Fundamentally Distributed

- Add SR-PCE nodes where needed; per geographic region, per service, ...
  - SR-PCE needs to get the required topology information for its task
    - i.e., to compute inter-domain paths SR-PCE needs the multi-domain topology
- Example:



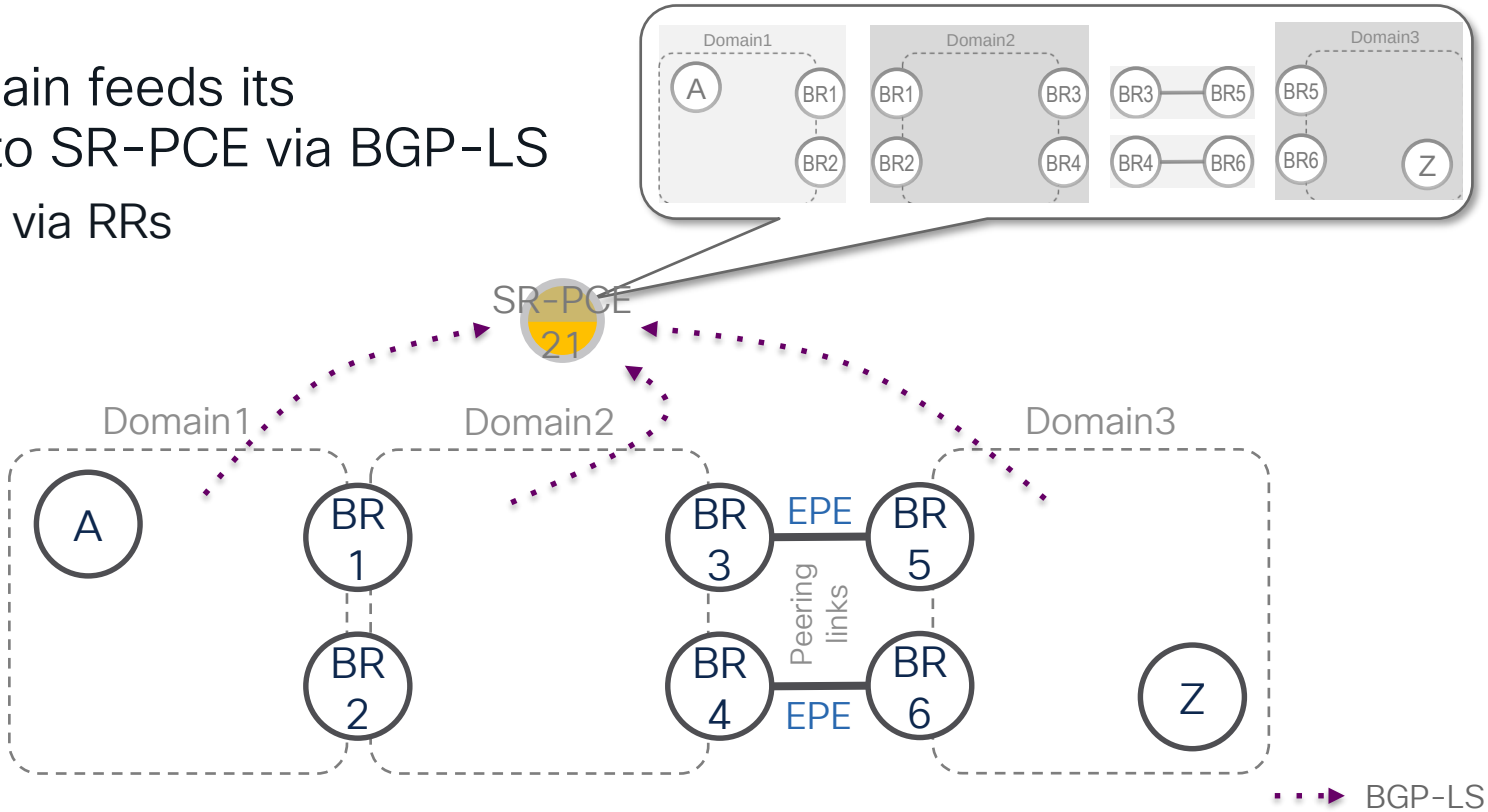
SR-PCE – Fundamentally Distributed

- Using RRs to scale the BGP-LS topology distribution
- Any node can have a BGP-LS session to the RR



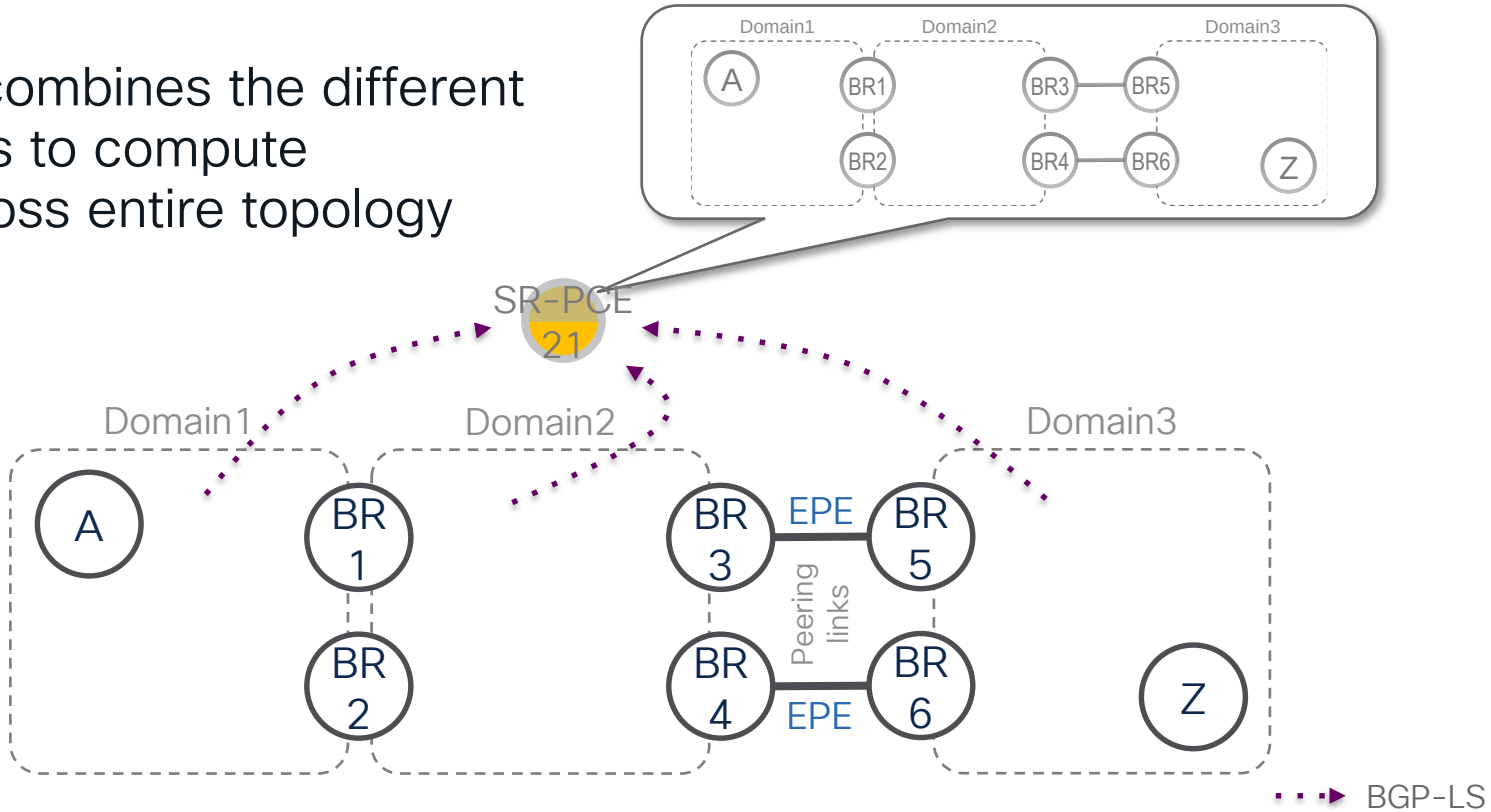
SR-PCE receives topology of all domains

- Each domain feeds its topology to SR-PCE via BGP-LS
  - Typically, via RRs



SR-PCE consolidates the topologies

- SR-PCE combines the different topologies to compute paths across entire topology



SR-PCE – High Availability (HA)

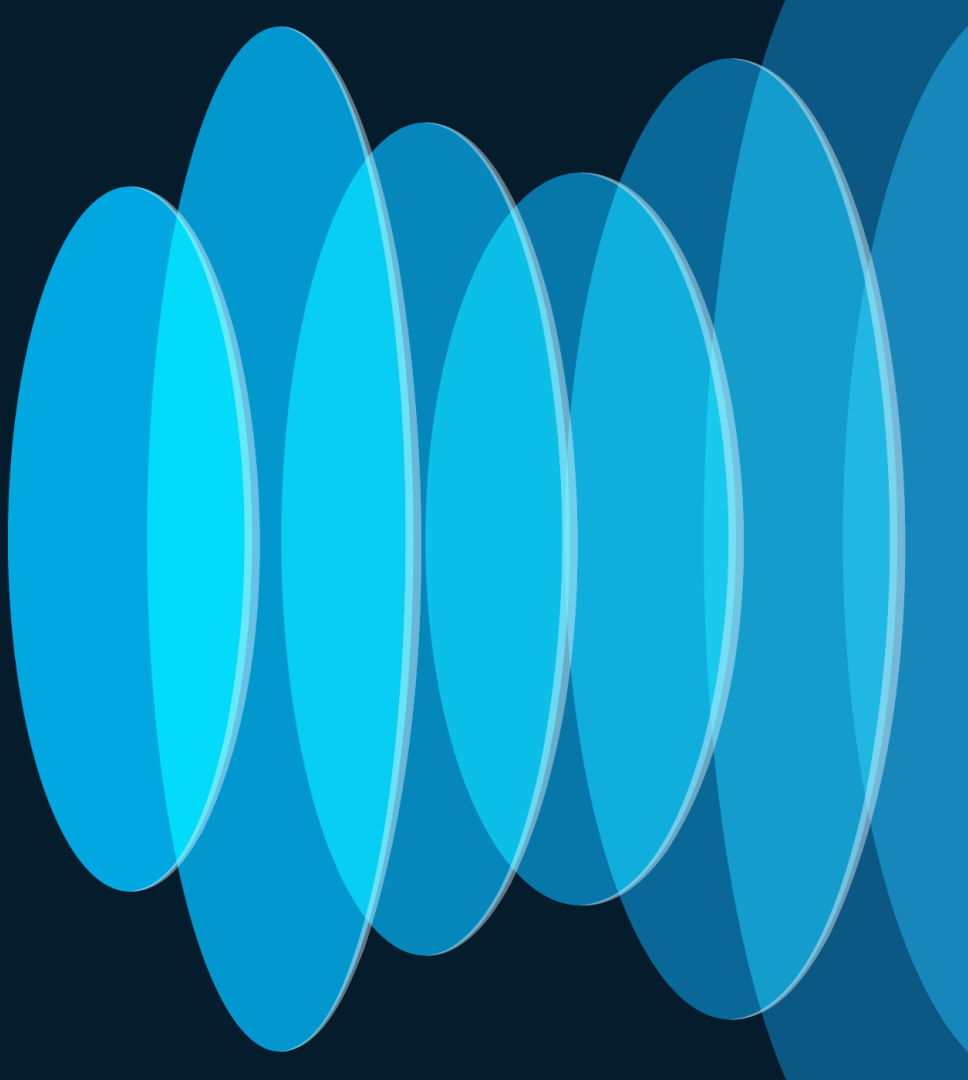
- SR-PCE leverages the well-known standardized PCEP HA
- Head-end sends PCEP Report for its SR Policies to all connected SR-PCE nodes
- Head-end delegates control to its primary SR-PCE
  - Delegate flag (D) is set in PCRept to primary SR-PCE
- Upon failure of the primary SR-PCE, head-end re-delegates control to another SR-PCE

SR-PCE and SR data-planes

- SR-PCE functions are applicable to SR-MPLS and SRv6

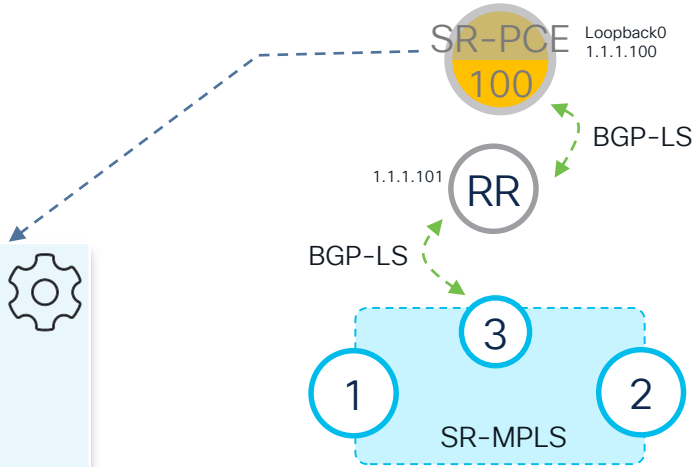
Intent-based Transport

SR-PCE with SR-MPLS



cisco Live!

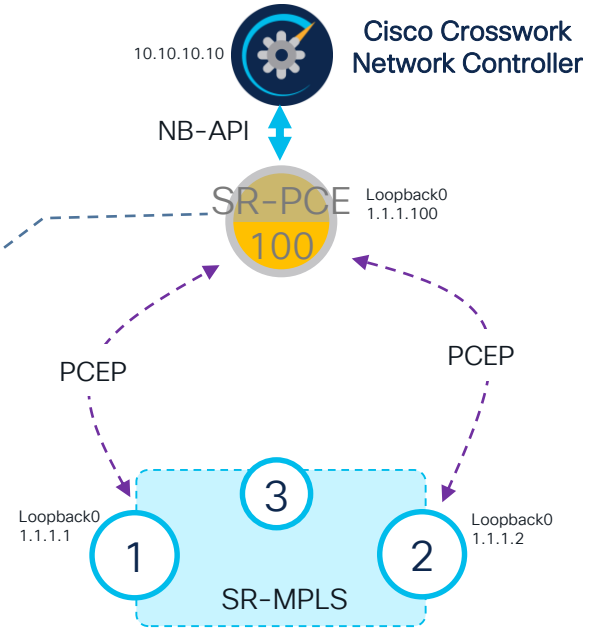
SR-PCE – Topology Collection



SR-PCE

```
router bgp 64000
 bgp router-id 1.1.1.100
 address-family link-state link-state
 !
 neighbor-group BGPLS
  remote-as 64000
  update-source Loopback0
  address-family link-state link-state
  !
 !
 neighbor 1.1.1.101
  use neighbor-group BGPLS
```

SR-PCE – NB-API and PCEP



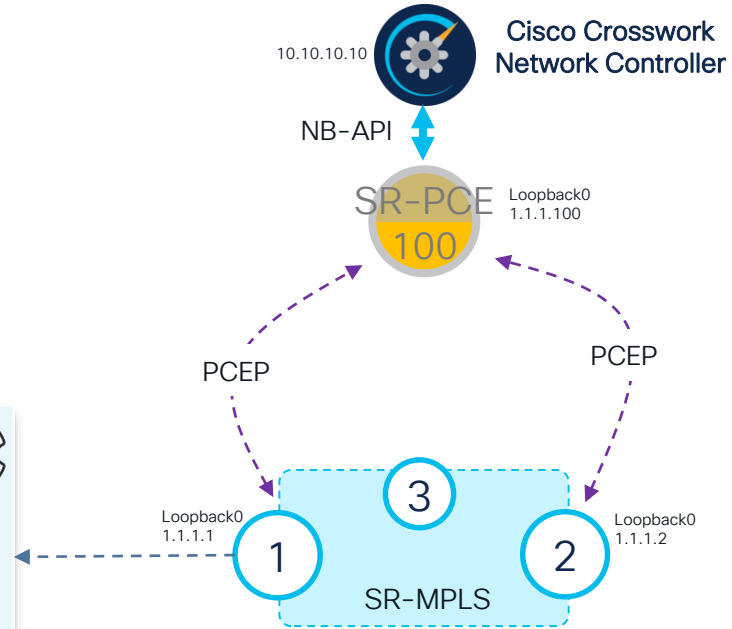
SR-PCE

```
pce
 address ipv4 1.1.1.100
 api
  user cisco
   password encrypted <api_auth_pwd>
  !
  authentication digest
  ipv4
   address 10.10.10.10
  !
 !
 password encrypted <pcep_md5_auth_pwd>
```

SR-PCE – PCEP at HE

SR-TE HE (e.g., node 1)

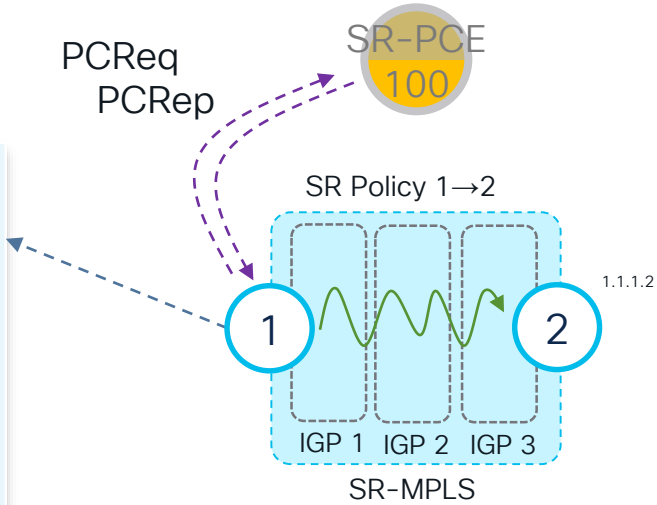
```
segment-routing
 traffic-eng
  pcc
   source-address ipv4 1.1.1.1
   pce address ipv4 1.1.1.100
    precedence 100
    password encrypted <pcep_md5_auth_pwd>
   !
   report-all
```



Multi-domain SR-Policy

SR-TE HE (node 1)

```
segment-routing
 traffic-eng
  policy pol-C_1-EP_2
   color 1 end-point ipv4 1.1.1.2
   candidate-paths
    preference 100
     dynamic
     pcep
      !
      metric
       type [igp | te | latency | hopcount]
```

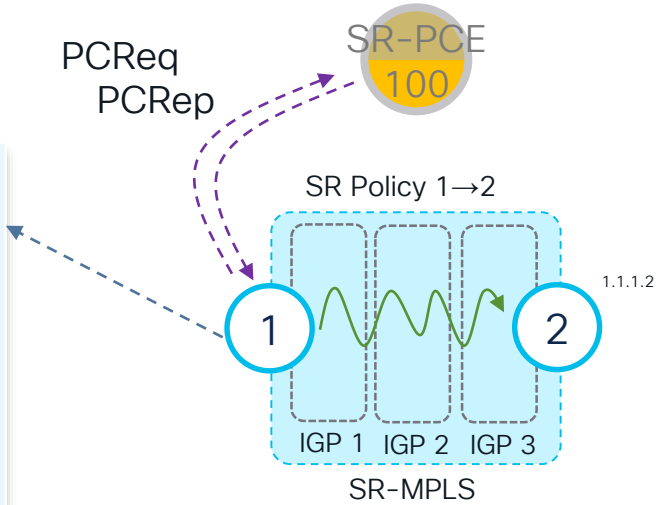


PCReq = PCEP Request message
PCRep = PCEP Reply message

Multi-domain SR-Policy

SR-TE HE (node 1)

```
segment-routing
 traffic-eng
  policy pol-C_1-EP_2
   color 1 end-point ipv4 1.1.1.2
   candidate-paths
    preference 100
     dynamic
     pcep
      !
      metric
       type [igp | te | latency | hopcount]
       !
      !
     constraints
      affinity
       exclude-any
        name aff-brown
```



PCReq = PCEP Request message
PCRep = PCEP Reply message

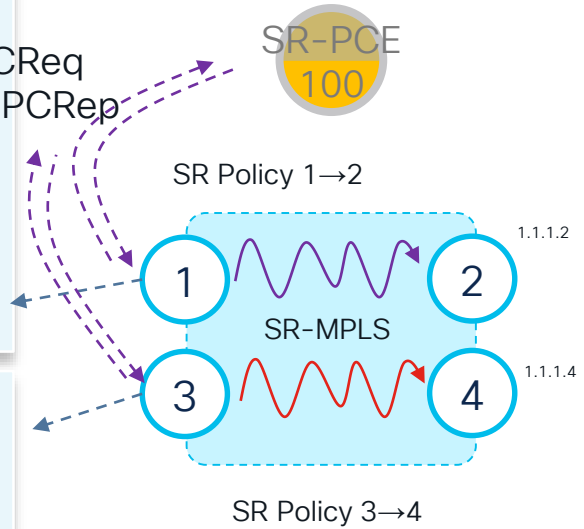
Disjoint SR-Policies

SR-TE HE (node 1)

```
segment-routing
 traffic-eng
  policy pol-C_1-EP_2
   color 10 end-point ipv4 1.1.1.2
   candidate-paths
    preference 100
     dynamic
      pcep
       metric
        type [igp | te | latency | hopcount]
      constraints
       disjoint-path group-id 77 type [link|node|srlg|srlg-node]
```

SR-TE HE (node 3)

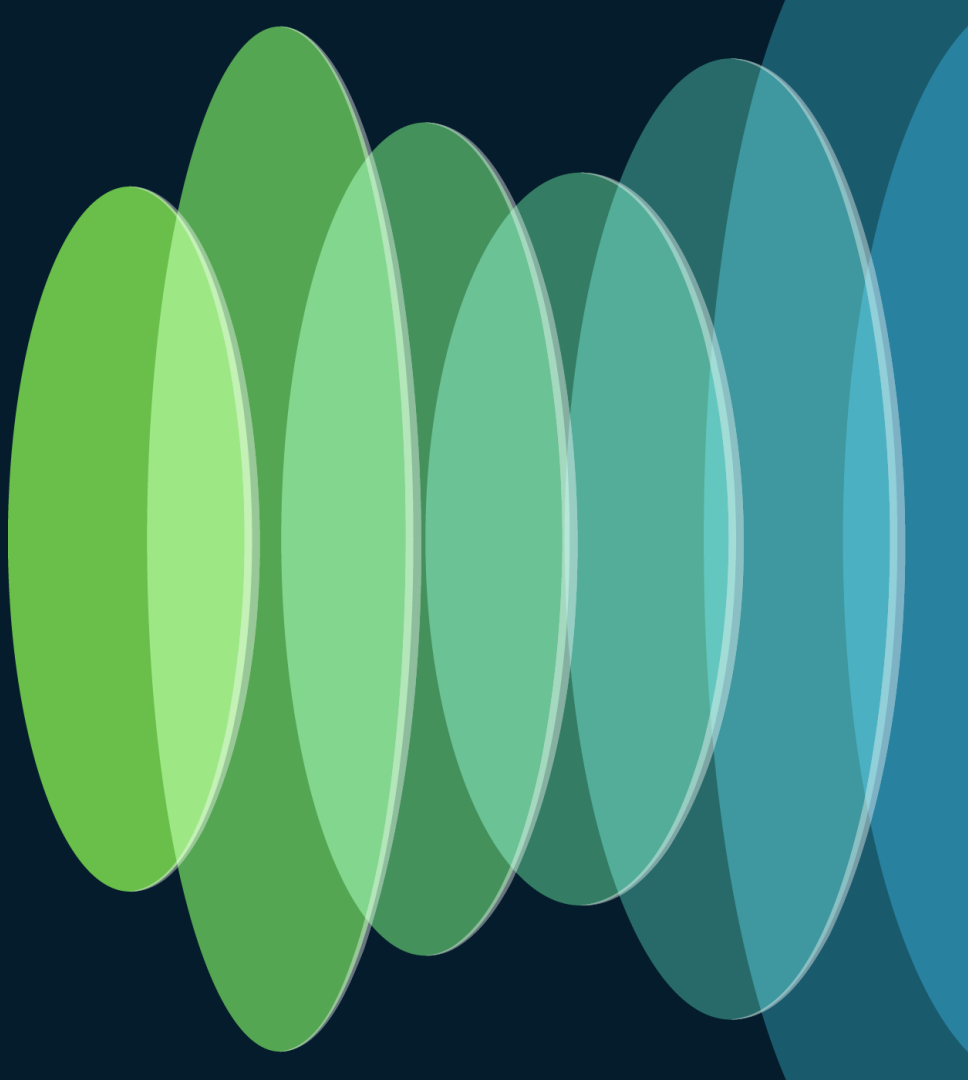
```
segment-routing
 traffic-eng
  policy pol-C_3-EP_4
   color 10 end-point ipv4 1.1.1.4
   candidate-paths
    preference 100
     dynamic
      pcep
       metric
        type [igp | te | latency | hopcount]
      constraints
       disjoint-path group-id 77 type [link|node|srlg|srlg-node]
```



PCReq = PCEP Request message
PCRep = PCEP Reply message

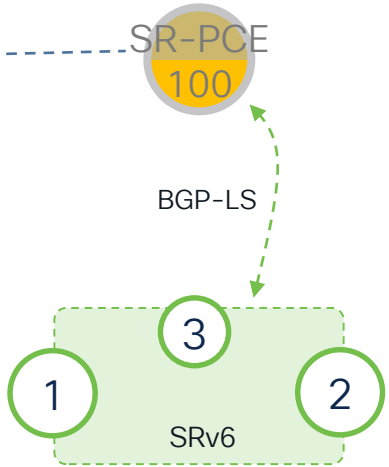
Intent-based Transport

SR-PCE with SRv6



cisco Live!

SRv6-PCE – Topology Collection



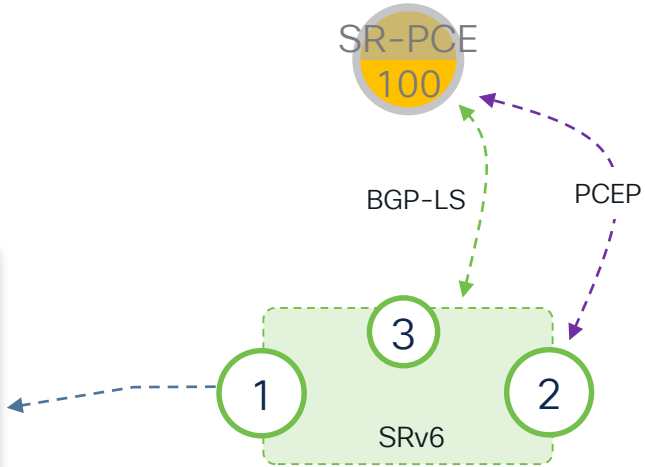
SR-PCE

```
router bgp 64000
 bgp router-id 1.1.1.1
 address-family link-state link-state
 !
 neighbor-group BGPLS
  remote-as 64000
  update-source Loopback0
  address-family link-state link-state
  !
 !
 neighbor fcbb:bb00:3::1
  use neighbor-group BGPLS
```

SRv6-PCE – Headend

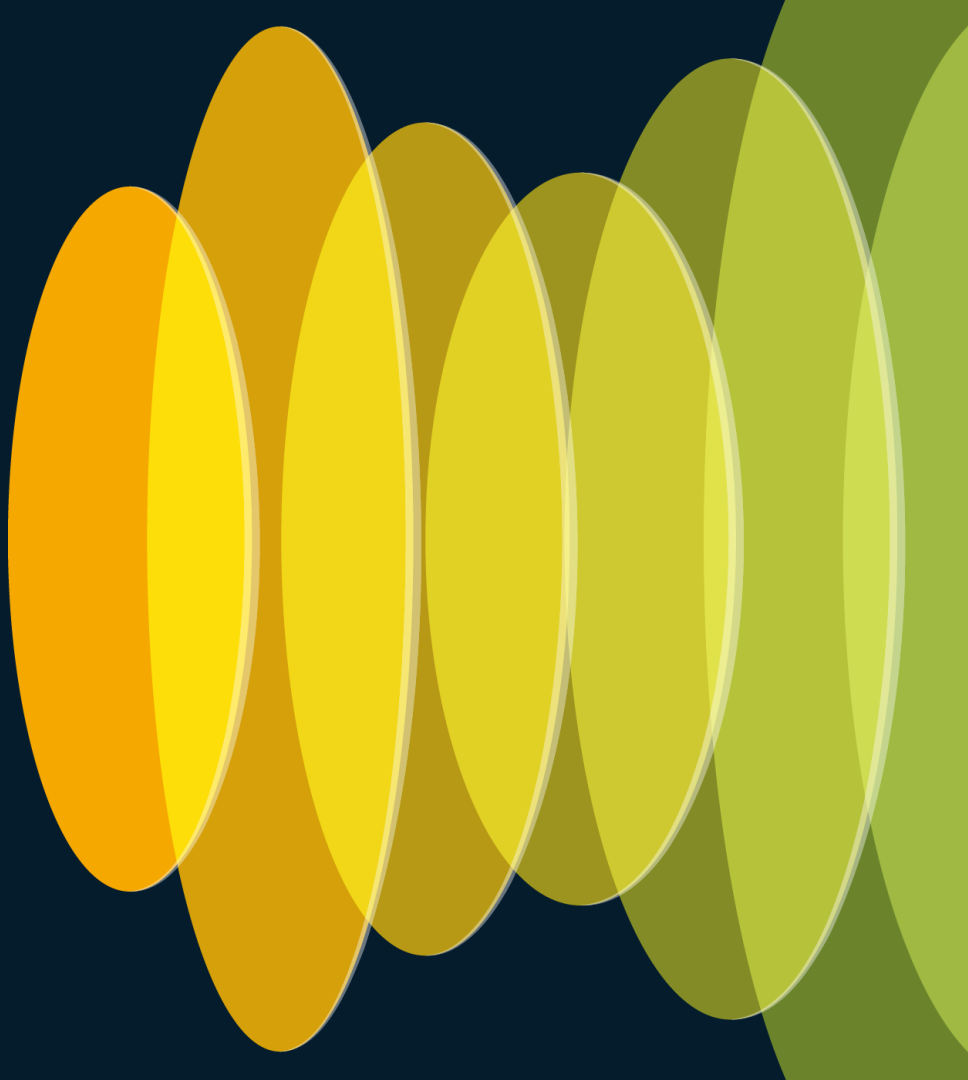
SR-PCE

```
segment-routing
 traffic-eng
  pcc
   source-address ipv4 fcbb:bb00:1::1
   pce address ipv6 1::1
    precedence 100
    password encrypted <pcep_md5_auth_pwd>
   !
   report-all
```



Intent-based Transport

IGP Flexible Algorithm



cisco Live!

IGP Flexible Algorithm (FA)

- A solution that allows IGPs themselves to compute constraint-based paths over the network
- It specifies a way of using Segment Routing (SR) Prefix-SIDs and SRv6 locators to steer packets along the constraint-based paths
- Standardized in RFC9350

IGP Flexible Algorithm (FA)

- Flex-Algo or FA in short
- Flex == an algorithm instance is defined by the operator, on a per-deployment basis

IGP Flexible Algorithm (FA)

- A Flex-Algo instance definition includes a metric and constraints

IGP Flexible Algorithm (FA)

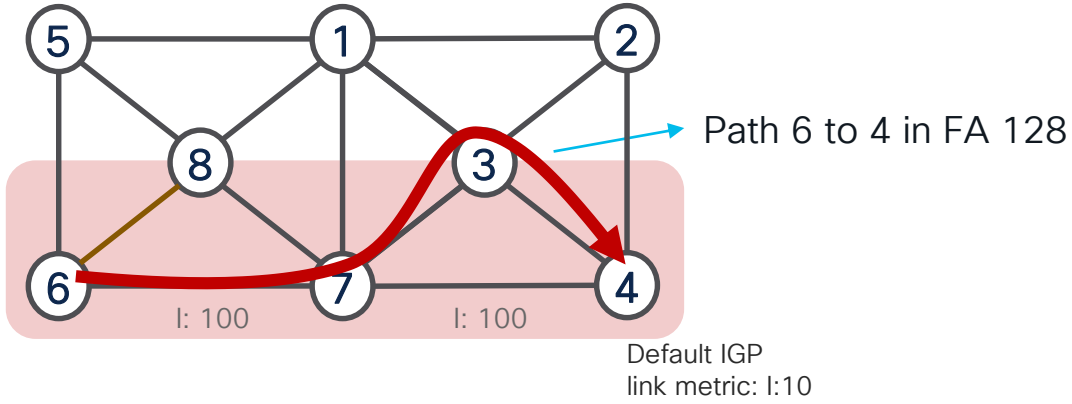
- A Flex-Algo instance definition includes a metric and constraints
- Metric – minimization of a given link metric
  - IGP metric
  - Traffic Engineering (TE) metric
  - Delay metric – measured (min-delay value) / static
  - Generic metric – other user-defined

IGP Flexible Algorithm (FA)

- A Flex-Algo instance definition includes a metric and constraints
- Constraints – based on link attributes
  - Affinity – inclusion / exclusion of links with a given admin-group(s)
  - Reverse Affinity
  - Shared Risk Link Groups – exclusion of links with a given SRLG
  - Min-BW – inclusion of links with an interface speed equal or greater than a given value
  - Max-Delay – inclusion of links with a min-delay equal or less than a given value

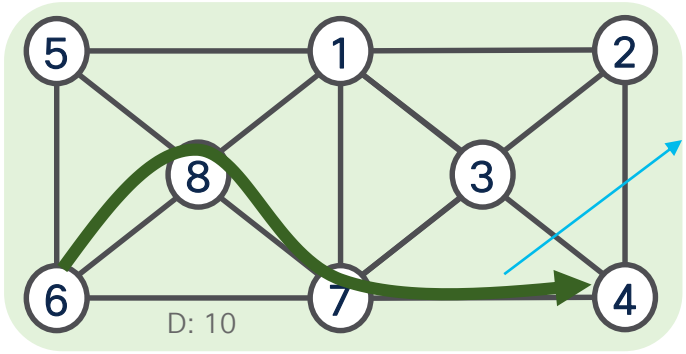
IGP Flexible Algorithm

- Example
  - Operator defines Flex-Algo instance 128 in a sub-set of nodes (6-8-7-3-4) as “minimize IGP metric while avoiding links with link-affinity brown”

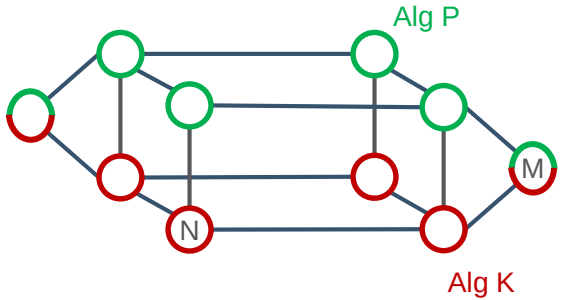


IGP Flexible Algorithm

- Example
  - Operator defines Flex-Algo instance 129 in all nodes as “minimize delay metric”



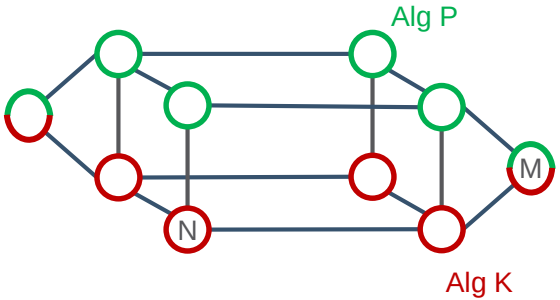
Flexible Algorithm Operation



1 • Flex-Algo Membership
  - Each node MUST advertise Flex-Algo(s) that it is participating in
  - A Flex-Algo instance can be enabled on all or a subset of nodes
  - Each node can participate in multiple Flex-Algo(s)

  - Example:
    - Node N is enabled to participate in Flex-Algo instance K
    - Node M is enabled to participate in Flex-Algo instances K and P

Flexible Algorithm Operation



2 • Flex-Algo Prefix SID / Locator

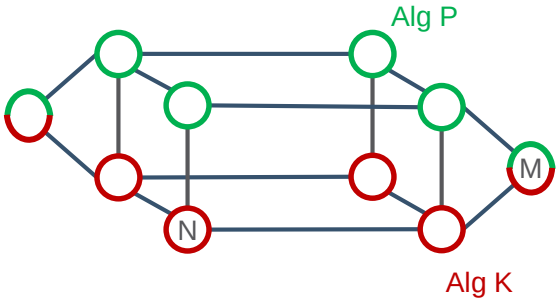
- If a node advertises participation in a Flex-Algo likely it also advertises a prefix SID (SR-MPLS) or locator (SRv6) for that Flex-Algo

- Examples:

  - SR-MPLS – Node M advertises multiple Prefix SIDs for its Loopback interface
    - 16010 for ALGO 0
    - 17010 for ALGO P
    - 18010 for ALGO K

  - SRv6 – Node M advertises multiple Locators
    - fcbb:bb00:0010:: for ALGO 0
    - fcbb:bb01:0010:: for ALGO P
    - fcbb:bb02:0010:: for ALGO K

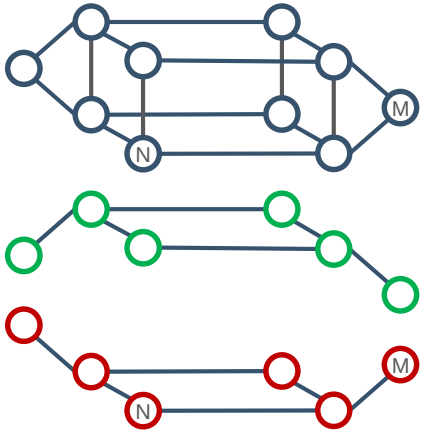
Flexible Algorithm Operation



3

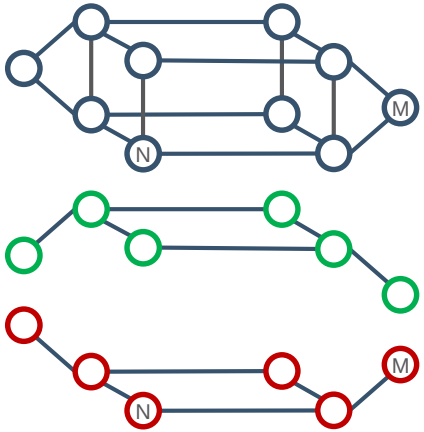
- Flex-Algo Definition
  - Each node MUST have a consistent definition of the Flex-Algo(s) that it is participating in
  - Local configuration
    - Day-0 provisioning
  - Learned via IGP flooding
    - New top TLV defined for Flex-Algo definition advertisement
  - Example:
    - Flex-Algo instance K == minimize delay metric

Flexible Algorithm Operation



4 • Flex-Algo Computation
  - N prunes any node not a member of instance K
  - N prunes any link that is exclude by instance K
  - Resulting topology is called Topo(K)
  - N computes shortest-path tree on Topo(K) with metric defined by K

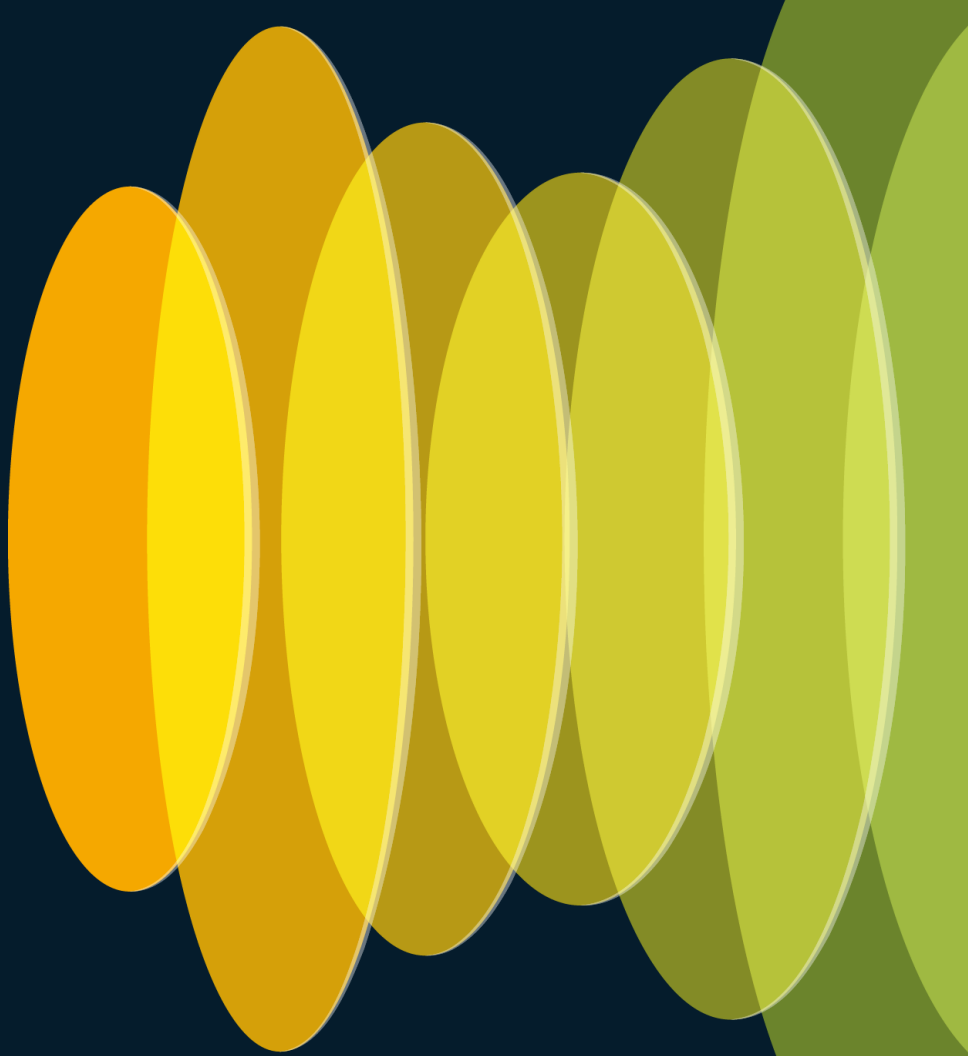
Flexible Algorithm Operation



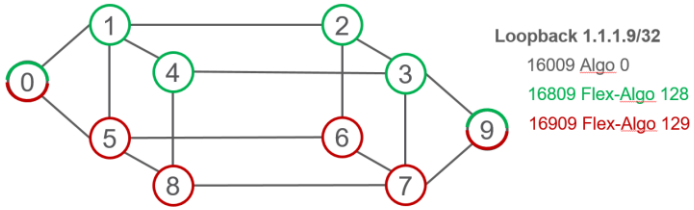
5

- Flex-Algo Prefix SID / Locator FIB installation
  - N installs any reachable Prefix-SID / Locator of instance K in the forwarding table along the computed shortest-path on Topo(K)

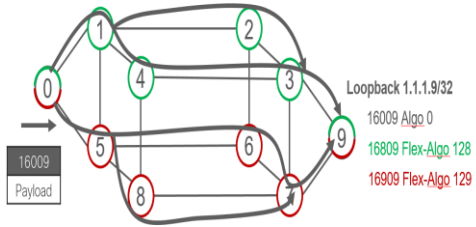
FA Use Cases



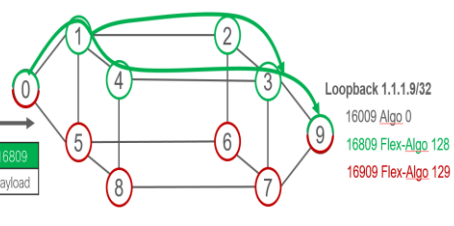
Use-Case – Multi-Plane Networks



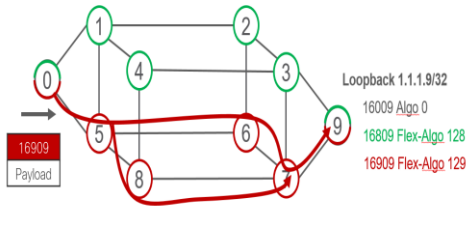
- All the nodes support Algo 0: minimize IGP metric
- Green nodes also support 128: minimize IGP metric
- Red nodes also support 129: minimize Delay



- Path to Node 9 across Algo 0



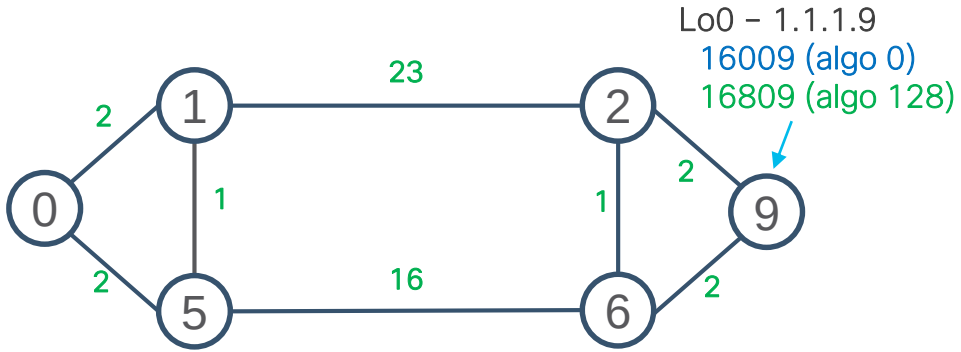
- Path to Node 9 across Flex-Algo 128

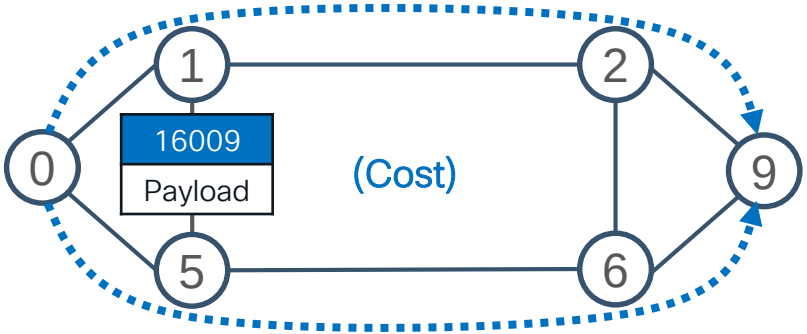


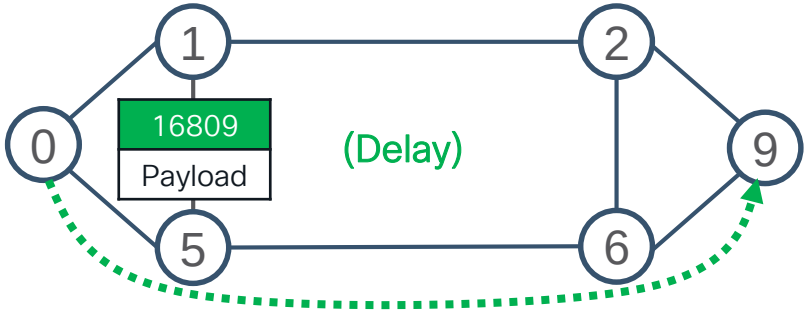
- Path to Node 9 across Flex-Algo 129

Use-Case – Low Delay Transport

- All nodes support Algo 0 & 128
- ISIS link metric 10
- Algo 128: minimize delay metric
- Per-link measurement of delay and advertisement as delay metric via ISIS
- Delay metric at that time shown in green

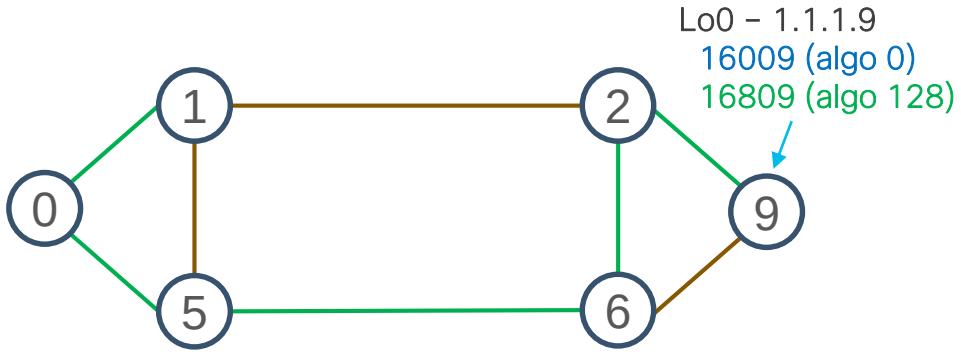


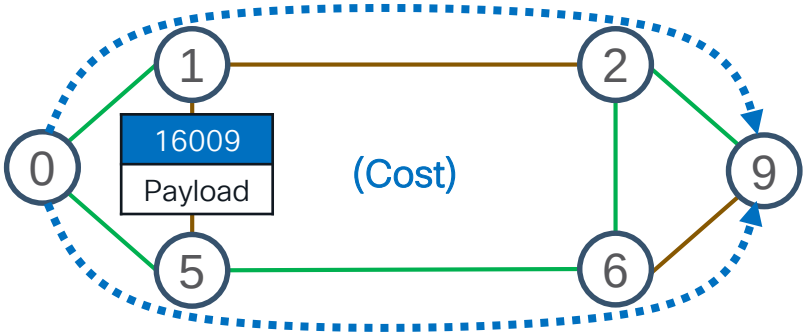


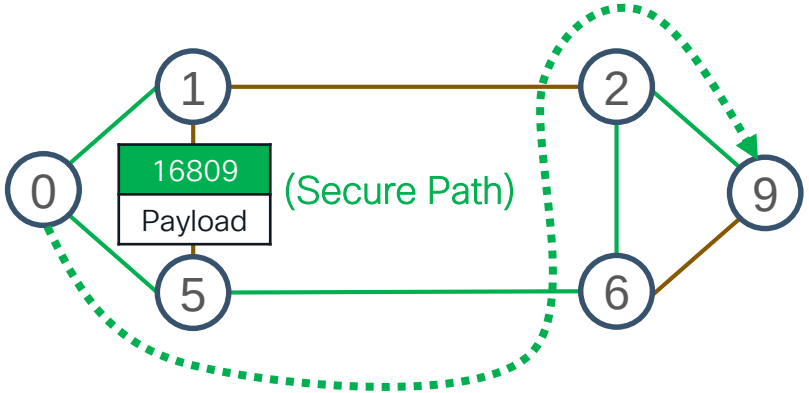


Use-Case – Secure Transport

- ISIS link metric 10
- Link colors shown Unencrypted / Encrypted
- All nodes support Algo 0 & 128
- Algo 128: minimize IGP while traversing links with encryption enabled (exclude brown)
- Per-link colors flooded in IGP

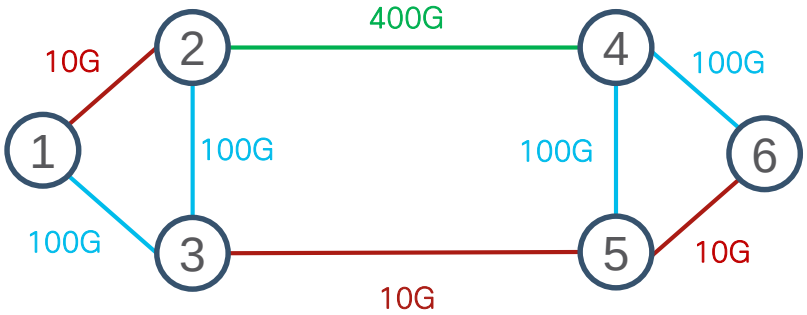






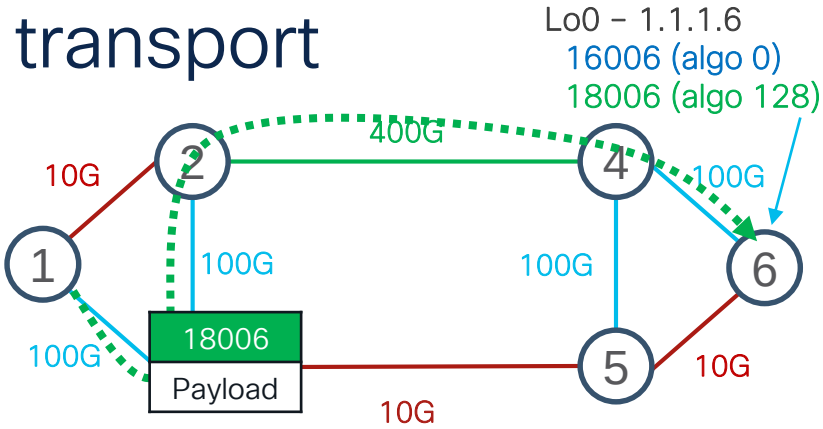
Use-Case – High-capacity transport

- Flex-Algo instance consisting of links with a bandwidth exceeding a user-configured minimum value



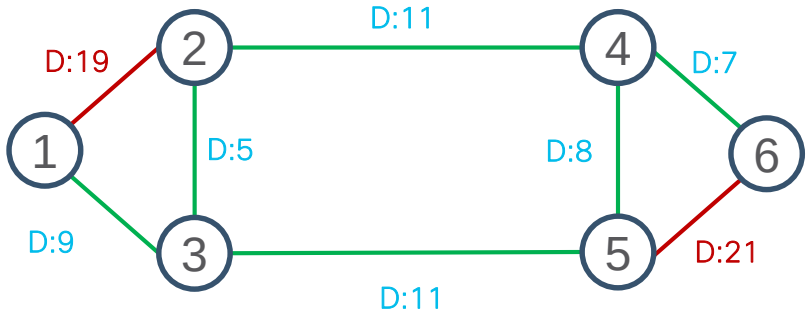
Use-Case – High-capacity transport

- Flex-Algo instance consisting of links with a bandwidth exceeding a user-configured minimum value



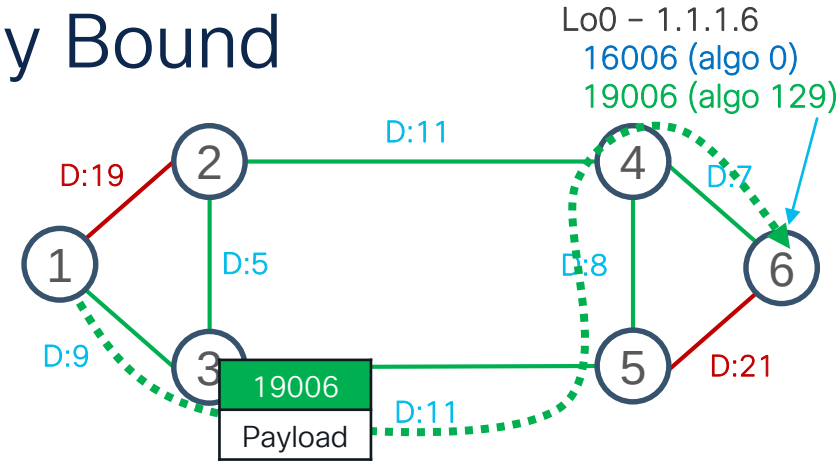
Use-Case – Interface Delay Bound

- Flex-Algo instance consisting of links with a manual / measured propagation delay less than a user-configured maximum value
- Note that this is different than FA with delay metric. Instead of minimizing e-2-e delay, it prunes links with a high-delay



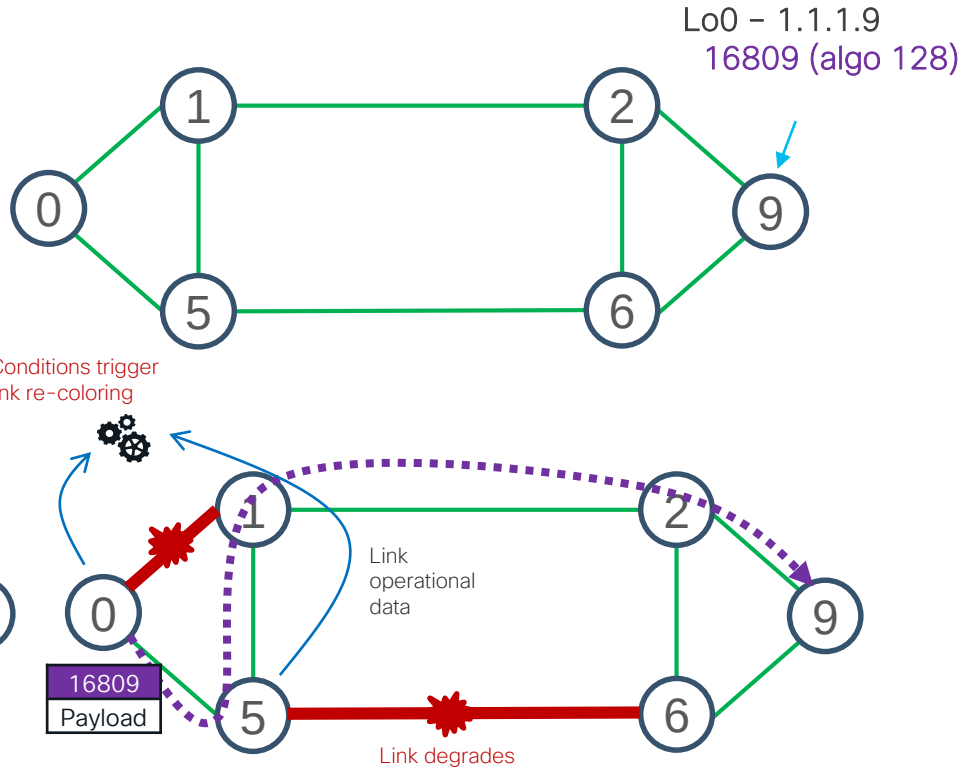
Use-Case – Interface Delay Bound

- Flex-Algo instance consisting of links with a manual / measured propagation delay less than a user-configured maximum value
- Note that this is different than FA with delay metric. Instead of minimizing e-2-e delay, it prunes links with a high-delay



Use-Case – “Clean” Transport

- ISIS link metric 10
- Link colors shown reliable / unreliable
  - Reliability of a link based on operator-defined factors
- All nodes support Algo 0 & 128
- Algo 128: minimize IGP while traversing reliable links (e.g. exclude unreliable)
- Per-link colors flooded in IGP



Topology Independent LFA (TI-LFA)

- TI-LFA algorithm is performed within Topo(K)
- Backup path is expressed with Prefix-SIDs of Algo K

- Benefits: the backup path is optimized per Flex-Algo !!!

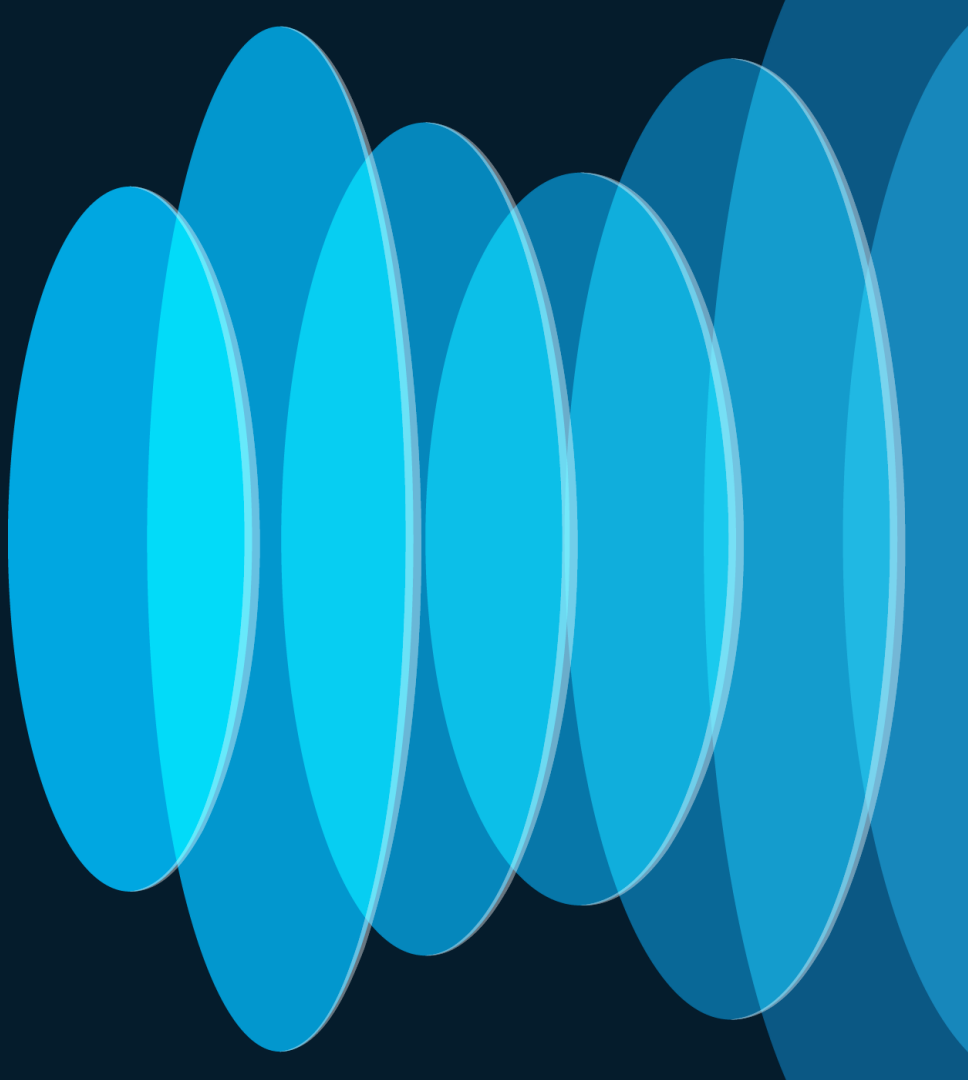
Flex-Algo and SR data-planes

- Flex-Algo is applicable to SR-MPLS and SRv6

- SR-MPLS
  - A flex-algo node SID (prefix label) is assigned to the loopback interface
- SRv6
  - A flex-algo locator is assigned to the node

Intent-based Transport

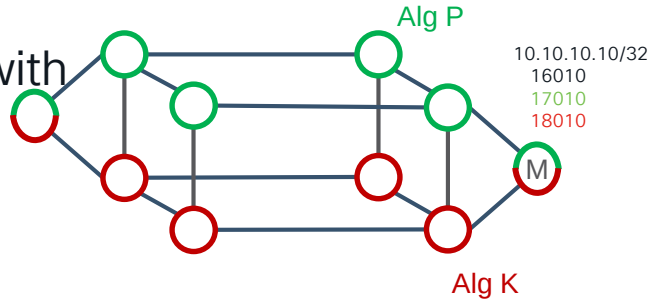
Flex-Algo over SR-MPLS



cisco Live!

IGP Flexible Algorithm (FA)

- FA provides custom Prefix-SIDs representing different transport intents
  - Single IGP process
  - Single loopback (no additional loopback interfaces required)
  - Flex-Algo Prefix SIDs are advertised as additional SIDs of the existing loopback address
- Example:
  - Node M advertises loopback0 10.10.10.10/32 with
    - Prefix SID 16010 for ALGO 0
    - Prefix SID 17010 for ALGO P
    - Prefix SID 18010 for ALGO K



Flex-Algo over SR-MPLS

Router 1:

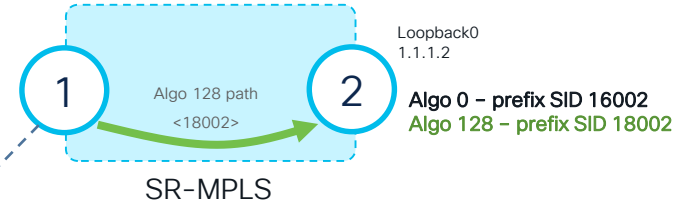
```
router isis core
 flex-algo 128
  metric-type delay
  advertise-definition
 !
 address-family ipv4 unicast
  segment-routing mpls
 !
 interface Loopback0
  passive
  address-family ipv4 unicast
   prefix-sid absolute 16001
   prefix-sid algorithm 128 absolute 18001
```

Flex-Algo over SR-MPLS – Steering

- Leverages an SRTE policy (on-demand / manual) with a flex-algo constraint
  - sid-list == single label (flex algo prefix-sid of intended destination)
- Automated steering is used to steer service routes with color over matching SR policies

Flex-Algo over SR-MPLS – Steering

- Example: On-demand SR Policy



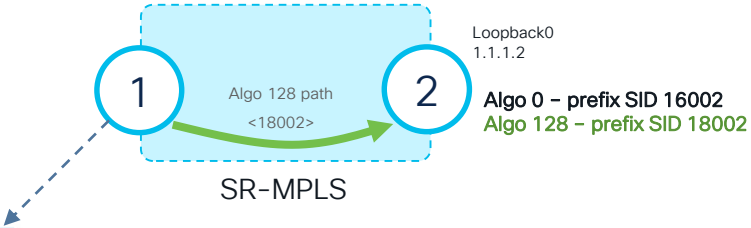
Router 1:

```
segment-routing
 traffic-eng
  on-demand color 128
   dynamic
   !
   constraints
    segments
     sid-algorithm 128
```

"Any 128-colored BGP route should be steered via the prefix-SID(ALGO 128) of the BGP nhop"

Flex-Algo over SR-MPLS – Steering

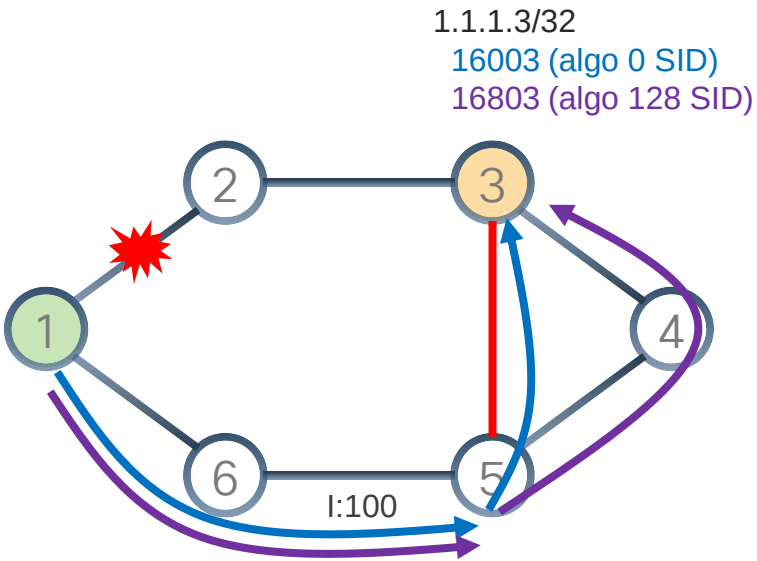
- Example: Manual SR policy



Router 1:

```
segment-routing
 traffic-eng
  policy sample-POLICY
   color 128 end-point ipv4 1.1.1.2
   candidate-paths
    preference 100
     dynamic
     !
     constraints
      segments
       sid-algorithm 128
```

FA over SR-MPLS - TI-LFA Backup path per Algo



At node 1 for destination 3

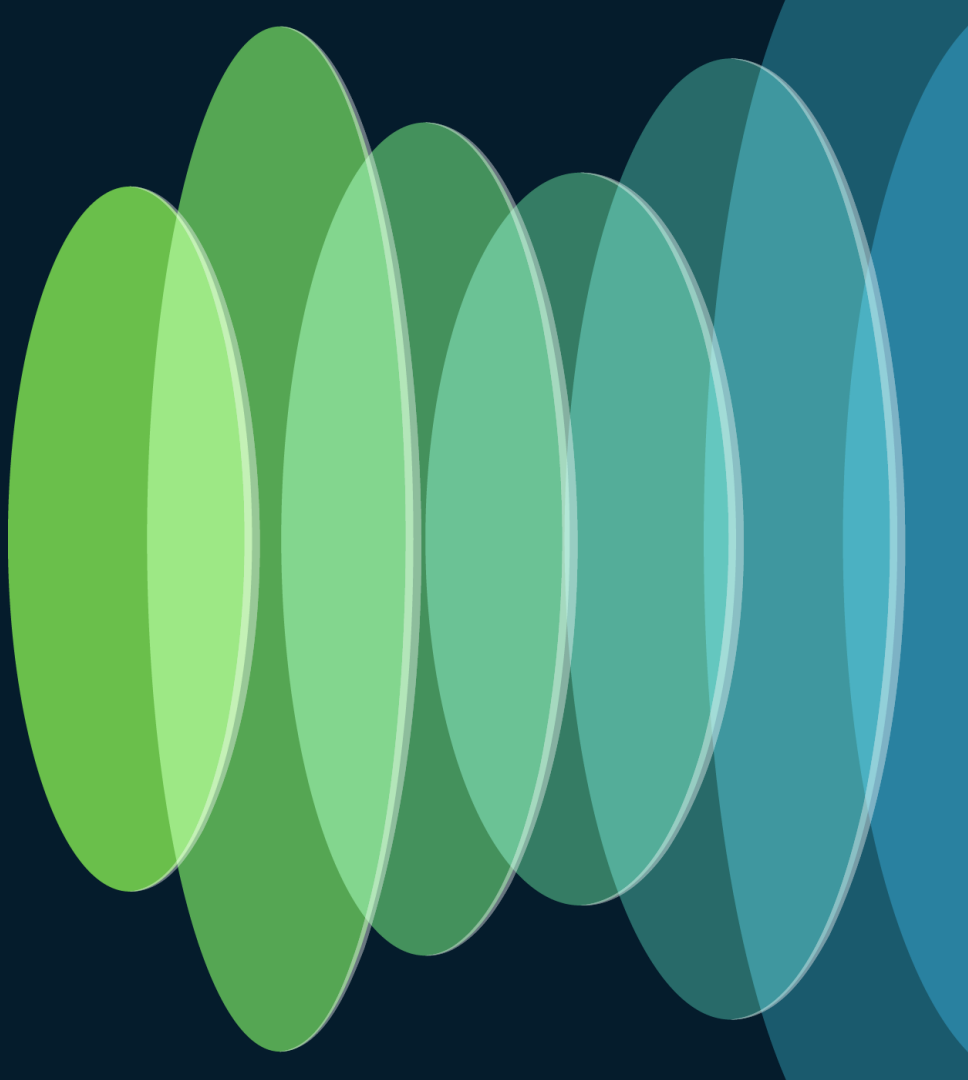
```
16003 => 16003 via 2
  backup: <24065, 16003> via 6

16803 => 16803 via 2
  backup: <24065, 16803> via 6
```

Usage of Algo-128 Prefix-SID 16803 ensures that the Algo 128 backup path also avoids the red link

IGP link metric: I:10 (default)
Adj SID convention: 240XY Adj SID from node X to node Y

Intent-based Transport

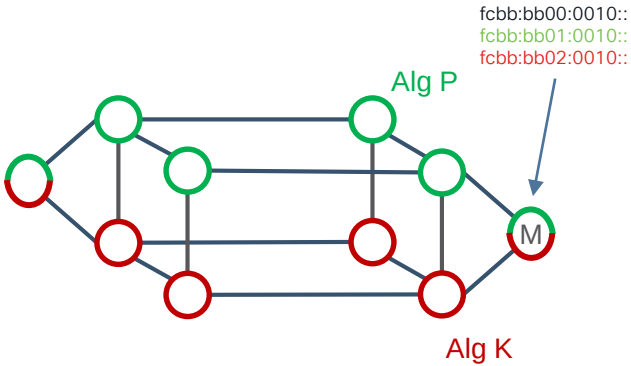
Flex-Algo over SRv6



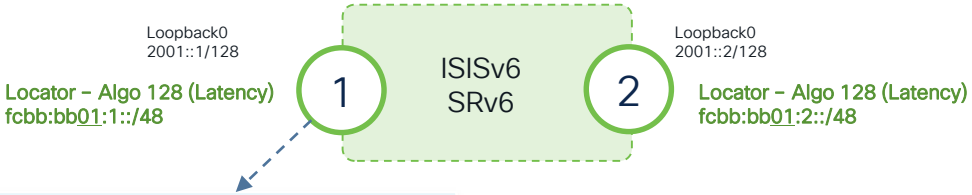
cisco Live!

IGP Flexible Algorithm (FA)

- FA provides custom Locators representing different transport intents
  - Single IGP process
  - Flex-algo locators are advertised in addition to the best-effort one
- Example:
  - Node M advertises:
    - fcbb:bb00:0010:: for ALGO 0
    - fcbb:bb01:0010:: for ALGO P
    - fcbb:bb02:0010:: for ALGO K



Flex-Algo over SRv6




```
segment-routing
 srv6
  locators
   locator LATENCY
    micro-segment behavior unode psp-usd
    prefix fcbb:bb01:1::/48
    algorithm 128
```

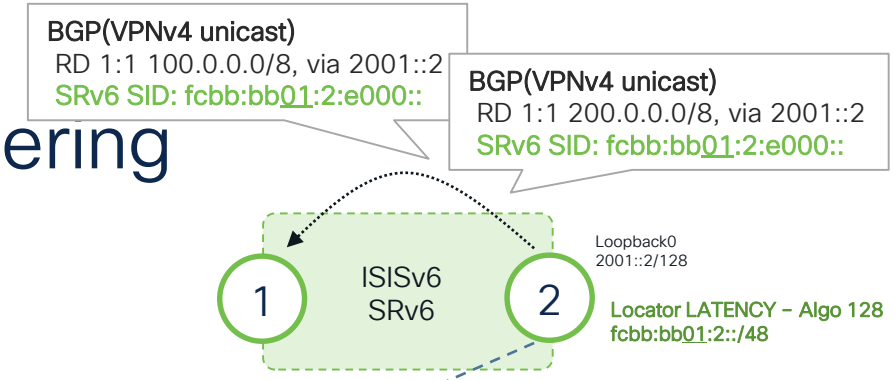
```
router isis 1
 flex-algo 128
  metric-type delay
  advertise-definition
 !
 address-family ipv6 unicast
  segment-routing srv6
   locator LATENCY
```

Flex-Algo over SRv6 – Steering

- Ingress PE directly encapsulates traffic based on the Service route's SRv6 SID advertised in BGP
  - Service SID = algo locator + decap function
  - Service SID directly encodes the transport intent (algo locator)
- Ultimate simplification – direct steering into Algo path
  - Does not require an SR-TE policy
  - Does not require to color service routes

Flex-Algo over SRv6 – Steering

Per-VRF granularity



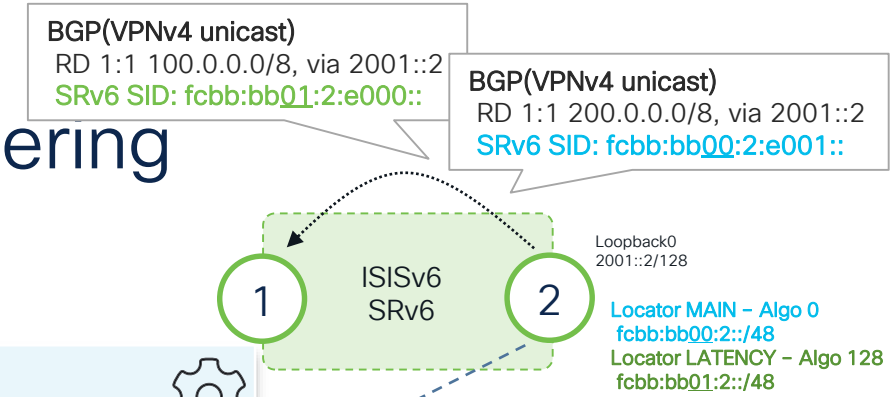
Router 2:

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

Egress PE (router 2) allocates End.DT4 function from locator in FA and advertises it with all prefixes

Flex-Algo over SRv6 – Steering

Per-Prefix granularity

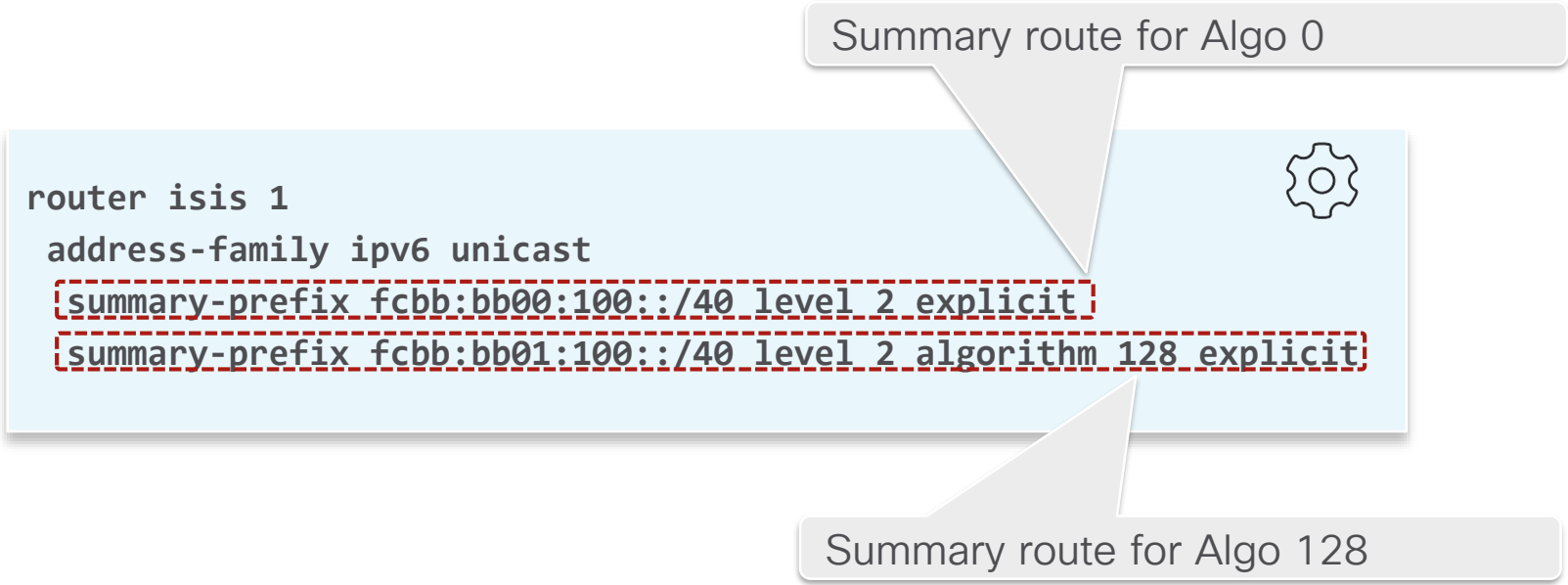



```
route-policy MIX
  if destination in (100.0.0.0/8) then
    set srv6-alloc-mode per-vrf locator LATENCY
  else
    set srv6-alloc-mode per-vrf locator MAIN
  endif
end-policy


router bgp 1
 vrf fa_per_prefix
  address-family ipv4 unicast
   segment-routing srv6
    alloc mode route-policy MIX
```

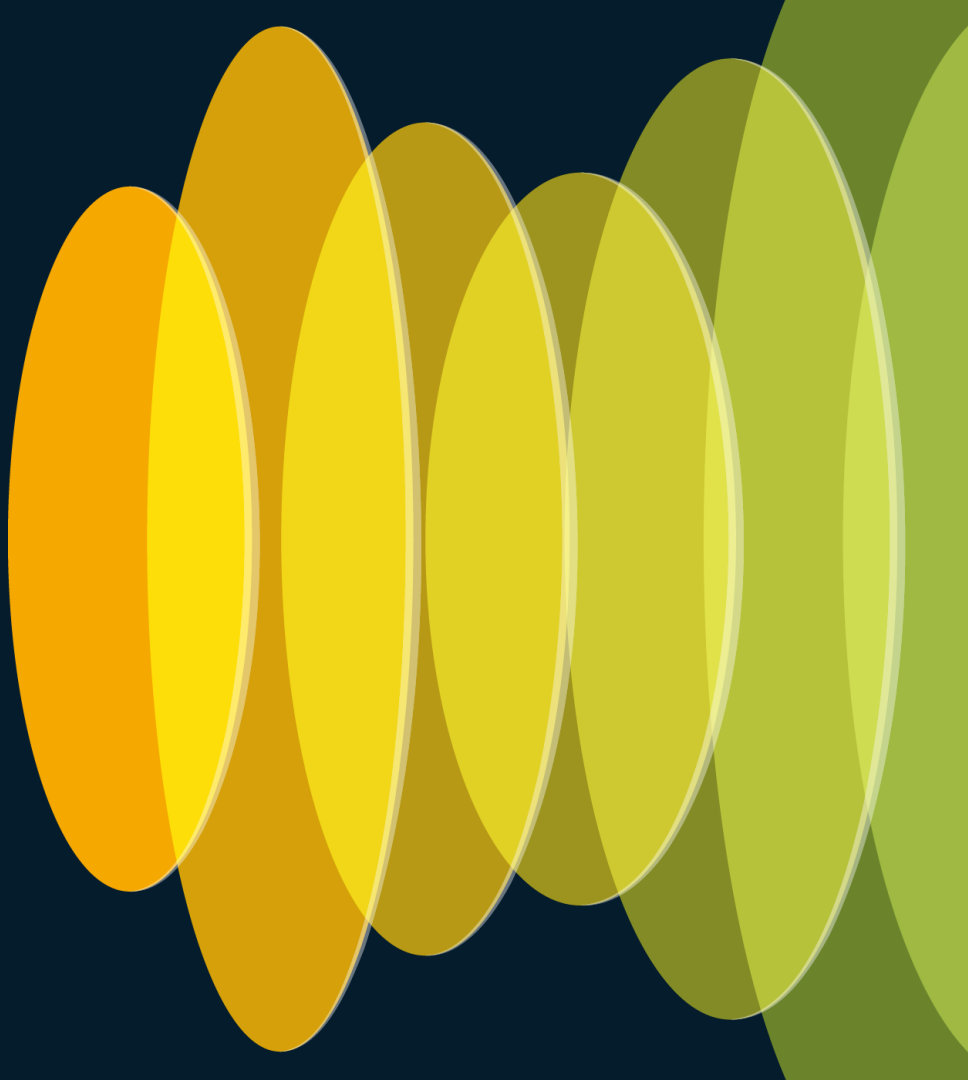
Set of prefixes assigned function from LATENCY locator

Reference to route-policy

Flex-Algo Summarization



Flex-Algo Details



Advertisements of Link Attributes for FA

- Link attributes that are to be used during Flex-Algorithm calculation MUST use the Application-Specific Link Attribute (ASLA) advertisements defined in [RFC8919] or [RFC8920]
- The mandatory use of ASLA advertisements applies to link attributes; including:
  - Min Unidirectional Link Delay
  - TE Default Metric
  - Administrative Group / Extended Administrative Group
  - Shared Risk Link Group
  - And any other link attributes that may be used in the future

Flex-Algo – Metric-types

```
RP/0/RP0/CPU0:R1(config-isis)# flex-algo 140
RP/0/RP0/CPU0:R1(config-isis-flex-algo)#?
  <...>
  metric-type           Metric-type used by flex-algo calculation
  <...>

RP/0/RP0/CPU0:R1(config-isis-flex-algo)#metric-type ?
  delay  Use delay as metric
  te     Use Traffic Engineering metric
```

IGP == default metric-type

Flex-Algo – TE metric link attribute

```
RP/0/RP0/CPU0:R1(config)#router isis 100
RP/0/RP0/CPU0:R1(config-isis)#interface tenGigE 0/0/0/0
RP/0/RP0/CPU0:R1(config-isis-if)#address-family ipv4 unicast
RP/0/RP0/CPU0:R1(config-isis-if-af)#?
  <...>
  te-metric    Configure an application specific TE metric for the interface
  <...>

RP/0/RP0/CPU0:R1(config-isis-if-af)#te-metric ?
  flex-algo  Configure a Flex-algo TE metric for the interface

RP/0/RP0/CPU0:R1(config-isis-if-af)#te-metric flex-algo ?
  <1-16777214>  Flex-algo traffic-engineering metric
```

Flex-Algo – Example: metric TE

```
router isis 100
 flex-algo 140
  metric-type te
   advertise-definition
 !
 interface TenGigE0/0/0/0
  address-family ipv4 unicast
   te-metric flex-algo 100
 !
 interface TenGigE0/0/0/0
  address-family ipv4 unicast
   te-metric flex-algo 50
```

Flex-Algo – Example: metric Delay

```
router isis 100
 flex-algo 140
  metric-type delay
   advertise-definition
 !
 interface TenGigE0/0/0/0
  address-family ipv4 unicast
 !
 interface TenGigE0/0/0/1
  address-family ipv4 unicast
 !
!
```

```
performance-measurement
 interface TenGigE0/0/0/0
  delay-measurement
   advertise-delay 12
  !
 !

 interface TenGigE0/0/0/1
  delay-measurement
  !
 !
!
```

Manually configured min unidirectional link delay

Measured min unidirectional link delay

ISIS max-metric options for TE / Delay

```
RP/0/RP0/CPU0:R1(config-isis)#?
  <...>
  max-metric    Signal other routers to use us as transit option of last resort
  <...>

RP/0/RP0/CPU0:R1(config-isis)#max-metric ?
  <...>
  delay          Apply max-metric to delay metric
  te             Apply max-metric to TE metric
  <...>
```

Max-metric == 16,777,214

ISIS max-metric options for TE / Delay

```
router isis 100
 max-metric te
```

```
router isis 100
 max-metric delay
```

```
router isis 100
 max-metric te delay
```

Flex-Algo – Affinity link attribute

```
RP/0/RP0/CPU0:R1(config)#router isis 100
RP/0/RP0/CPU0:R1(config-isis)#?
  <...>
  affinity-map          Affinity map configuration
  <...>

RP/0/RP0/CPU0:R1(config-isis)#affinity-map ?
  WORD  Affinity attribute name
RP/0/RP0/CPU0:R1(config-isis)#affinity-map foo ?
  bit-position  Bit position for affinity attribute value

RP/0/RP0/CPU0:R1(config-isis)#affinity-map foo bit-position ?
  <0-255>  Bit position
```

Flex-Algo – Affinity link attribute

```
RP/0/RP0/CPU0:R1(config-isis)#interface tenGigE 0/0/0/0
RP/0/RP0/CPU0:R1(config-isis-if)#?
  <...>
  affinity          Application specific interface affinities
  <...>

RP/0/RP0/CPU0:R1(config-isis-if)#affinity ?
  flex-algo  Affinities for Flexible Algorithm application

RP/0/RP0/CPU0:R1(config-isis-if)#affinity flex-algo ?
  anomaly  Affinities to advertise when there is a link anomaly
  WORD     Affinity names
```

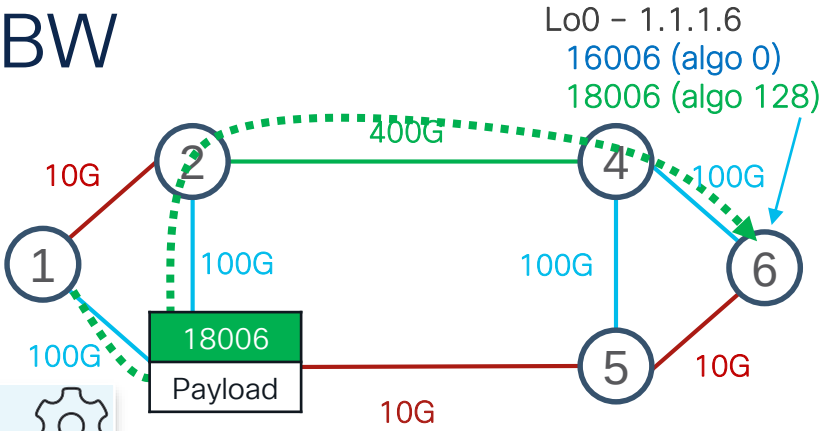
Flex-Algo – Affinity link attribute

```
RP/0/RP0/CPU0:R1(config-isis)#flex-algo 140
RP/0/RP0/CPU0:R1(config-isis-flex-algo)#?
  <...>
  affinity              Specify affinity names
  <...>

RP/0/RP0/CPU0:R1(config-isis-flex-algo)#affinity ?
  exclude-any  Exclude objects in flex-algo calculation
  include-all  Include objects in flex-algo calculation
  include-any  Include objects in flex-algo calculation
  reverse      Apply to links in the reverse direction
```

Flex-Algo – Example: metric IGP and Affinity constraints

```
router isis 100
 affinity-map foo bit-position 0
 affinity-map baa bit-position 2
 !
 flex-algo 140
  advertise-definition
  affinity include-any foo baa
 !
 interface TenGigE0/0/0/0
  affinity flex-algo foo baa
 !
 interface TenGigE0/0/0/1
  affinity flex-algo baa
 !
```

FA constraints – minimum-BW



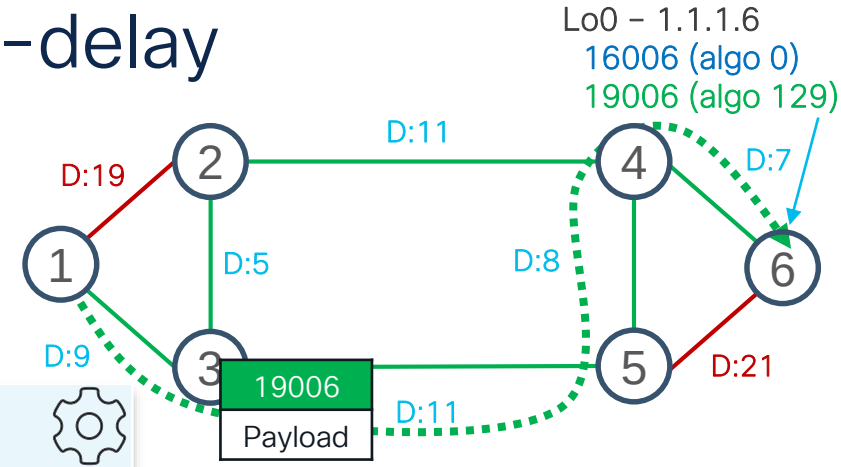
Router 6:

```
router isis core
 flex-algo 128
  advertise-definition
  minimum-bandwidth 100000000
 !
 interface Loopback0
  passive
  address-family ipv4 unicast
   prefix-sid algorithm 128 absolute 18006
```

Metric = IGP (default)

Minimum link bandwidth (kbits/sec); e.g. 100 Gbps

FA constraints - maximum-delay



Router 6:

```
router isis core
 flex-algo 129
  advertise-definition
  maximum-delay 12
 !
 interface Loopback0
  passive
  address-family ipv4 unicast
   prefix-sid algorithm 129 absolute 19006
```

Metric = IGP (default)

Maximum link delay (usec); e.g. 12 usecs

Flex-Algo – Disabling TI-LFA

- Usecase
- A Flex-algo instance for low-tier services without any fast-reroute guarantees

Flex-Algo – Disabling TI-LFA

```
RP/0/RP0/CPU0:R1(config)#router isis 100
RP/0/RP0/CPU0:R1(config-isis)# flex-algo 128
RP/0/RP0/CPU0:R1(config-isis-flex-algo)#?
  <...>
  fast-reroute          Configure Fast ReRoute
  <...>
RP/0/RP0/CPU0:R1(config-isis-flex-algo)#fast-reroute disable
```

```
router isis 100
 flex-algo 128
  fast-reroute disable
  metric-type delay
  advertise-definition
```

ISIS – Max-Paths enhanced granularity

- Usecase:

- Tight control of ASIC resources used for ECMP programming
  - Max-path = 1 => eliminates ECMP and programs unipath
- Various granularity:
  - max-paths per-algo + per-address-family
  - max-paths per-algo + per-address-family and per-prefix
- Applicable to SR-MPLS and SRv6

ISIS – Max-Paths enhanced granularity

```
router isis tag
 algorithm 0
  address-family {ipv4 | ipv6} unicast
   maximum-paths {max_path_val | route-policy rpl_name}
```

max_path_val == 1 programs single path (unipath)

```
router isis tag
 flex-algo algo_num
  address-family {ipv4 | ipv6} unicast
   maximum-paths {max_path_val | route-policy rpl_name}
```

ISIS – Max-Paths enhanced granularity

Example – per-prefix granularity (SR-MPLS)

```
prefix-set sample-pfx-set
  1.1.1.100/32
end-set
!
route-policy sample-rpl
  if destination in sample-pfx-set then
    set maximum-paths 1
  endif
end-policy
!
router isis 100
 flex-algo 128
  address-family ipv4 unicast
   maximum-paths route-policy sample-rpl
```

ISIS – Max-Paths enhanced granularity

Example – per-prefix granularity (SRv6)

```
prefix-set sample-pfx-set
  fcbb:bb01:108::/48
end-set
!
route-policy sample-rpl
  if destination in sample-pfx-set then
    set maximum-paths 1
  endif
end-policy
!
router isis 100
 flex-algo 128
  address-family ipv6 unicast
   maximum-paths route-policy sample-rpl
```

Flex-Algo – Example: No ECMP, no LFA/TI-LFA

- Usecase:
- Provide low-tier services with minimal transport SLA guarantees
  - No ECMP
  - No LFA / TI-LFA

```
router isis 100
 flex-algo 128
  fast-reroute disable
  advertise-definition
  address-family ipv4 unicast
   maximum-paths 1
```

SR-PM/IGP Delay Anomaly

- Use-Case
  - Degraded links (brown failures) can be problematic for applications. Increased latency or packet loss on a single link can be difficult to troubleshoot and can affect applications (e.g. using the ECMP paths)
- Requirement
  - When the delay or packet loss on a link goes over a certain threshold, the link should be avoided
- Solution
  - Node connecting to a degraded link measures link performance and automatically increases the IGP/TE metric of the link when degradation exceeds desired thresholds

SR-PM/IGP Delay Anomaly

- SR-PM
  - Anomaly-check threshold configuration is checked every computation-interval to trigger immediate flooding of PM metrics:
  - PM sets the Anomaly bit (A-bit) when the min-delay exceeds the upper-bound.
  - PM resets the A-bit when the min-delay falls below the lower-bound.
  - Lower-bound < Upper-bound to add hysteresis
- IGP
  - IGP adds a penalty to IGP / FA TE metrics when A-bit is set
  - Penalized IGP metric is advertised in the network
  - Do not need to make any change in FA to act on A-bit but instead rely on the updated IGP/FA TE metric value
  - When node is reloaded, the default / configured IGP metric (without penalty) is advertised until new measurement is available

SR-PM/IGP Delay Anomaly

- ISIS
  - "maximum" keyword will set the maximum link metric (0xFFFFFF) when an anomaly occurs. In ISIS this makes the link unusable during SPF computations. This is like configuring "metric maximum" for a link.
  - The "increment" and "multiplier" options will have the resulting metric capped at 0xFFFFFE which is the least desirable, but still usable.
  - All of ISIS metric configurations are per-topology (AFI/SAFI/topology-name).
- OSPF
  - OSPF config knobs are available at the OSPF interface, area and instance levels, and will be applied on all interfaces under that area/instance if configured at those levels.

SR-PM/ISIS/OSPF Delay Anomaly

```
performance-measurement
 delay-profile interfaces [name < STRING >]
  advertisement
   anomaly-check upper-bound <1-200000 usec> lower-bound <1-200000 usec>
```

```
router isis <tag>
 interface <name>
  address-family {ipv4 | ipv6} unicast
   metric fallback anomaly [delay] [te-metric] {increment <n> | multiplier <n> | maximum}
```

```
router ospf <tag>
 cost-fallback anomaly [delay] {igp-metric | te-metric} {increment <n> | multiplier <n> |
maximum <n>}
```

Flex-Algo Affinity on PM Anomaly

- Usecase
- Automatically set an affinity on an interface when the delay or packet loss goes over a certain threshold (Link Anomaly)
- Links with this affinity are excluded from a Flex-Algo instance

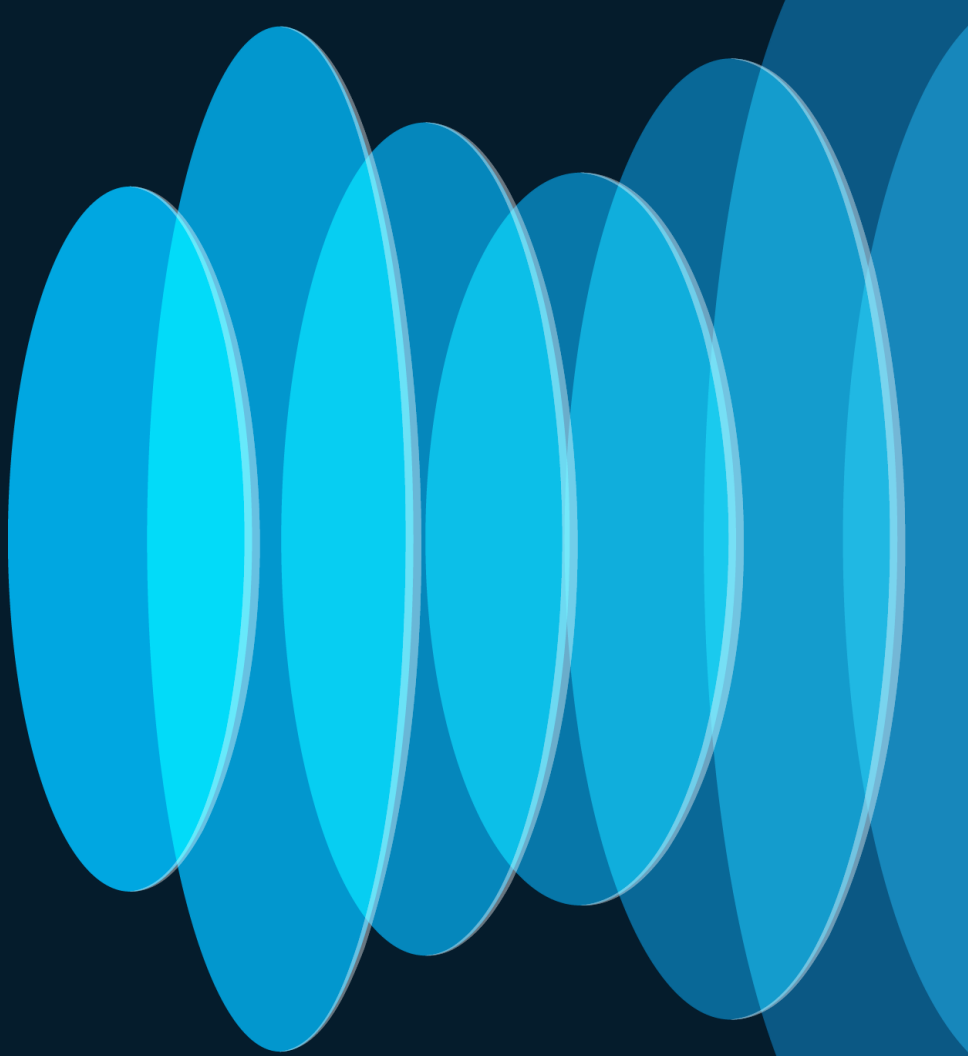
Flex-Algo Affinity on PM Anomaly

```
RP/0/RP0/CPU0:R1(config)#router isis 100
RP/0/RP0/CPU0:R1(config-isis)#int tenGigE 0/0/0/0
RP/0/RP0/CPU0:R1(config-isis-if)#?
  affinity          Application specific interface affinities

RP/0/RP0/CPU0:R1(config-isis-if)#affinity ?
  flex-algo  Affinities for Flexible Algorithm application

RP/0/RP0/CPU0:R1(config-isis-if)#affinity flex-algo ?
  anomaly  Affinities to advertise when there is a link anomaly
  WORD     Affinity names

RP/0/RP0/CPU0:R1(config-isis-if)#affinity flex-algo anomaly foo-anomaly
```

Flex-Algo – Example: Affinity on PM Anomaly

```
router isis 100
 affinity-map foo-anomaly bit-position 255
 !
 flex-algo 140
  advertise-definition
  affinity exclude-any foo-anomaly
 !
 interface TenGigE0/0/0/0
  affinity flex-algo anomaly foo-anomaly
 !
 interface TenGigE0/0/0/1
  affinity flex-algo anomaly foo-anomaly
 !
```

Flex-Algo Operational Considerations



Flex-Algo – Operational Considerations

- Number of Flex-Algo instances in the network
  - Treat a FA instance as a network-wide transport SLA intent
  - All services associated with this transport intent share a FA instance
  - Do not treat a FA as specific for one service

- Number of Flex-Algo instances in a node
  - Recommended to be in the single-digit range

Flex-Algo – Operational Considerations

- SRGB considerations (SR-MPLS)
  - A node participating in a FA instance has a Node SID allocated for the FA
  - Consider the expected number of Flex-Algo instances in the network when planning the SRGB size

```
router isis core
 interface Loopback0
  passive
  address-family ipv4 unicast
   prefix-sid absolute 16006
   prefix-sid algorithm 128 absolute 18006
```

Flex-Algo – Operational Considerations

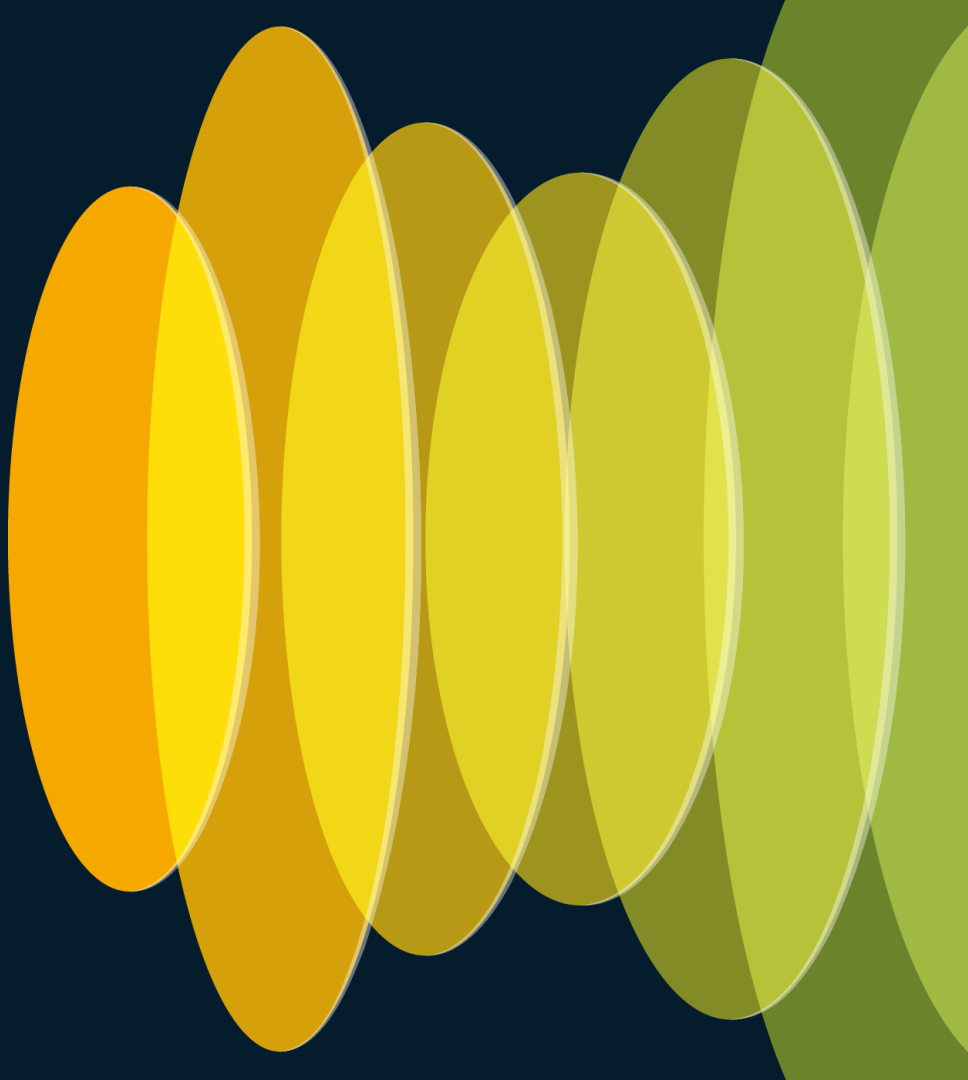
- FA Definition Consistency
- Each node MUST have a consistent definition of the Flex-Algo(s) that it is participating in
- Best practice:
  - Local configuration of FAD on a few selected nodes that would advertise it in an area
    - Configure best FAD advertisement preference on these node(s)
  - Configure remaining routers in area to learn the FAD via IGP flooding

Flex-Algo – Operational Considerations

- Flex-Algo with metric-type == TE
  - Explicitly configure the FA TE metric link attribute under the participating interfaces, independently of the legacy TE metric
  - If not done, IOS-XR implementation, by default, would advertise the legacy TE metric (configured under SRTE) as a FA ASLA link attribute

Transport Assurance

SR Performance Monitoring



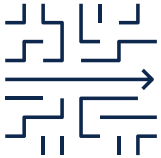
cisco Live!

Performance Measurement Toolkit Enables

AI + ML
Predictive Networks

+

Intent-Based
Network Slices



+

Service Assurance
and Health

Performance Measurement Toolkit

Measurement Analytics

Visibility to enable analytics using MDT and EDT KPI data (histogram/ min/ max/ avg/ variance delay, loss)

Network as a Horizontal Controller

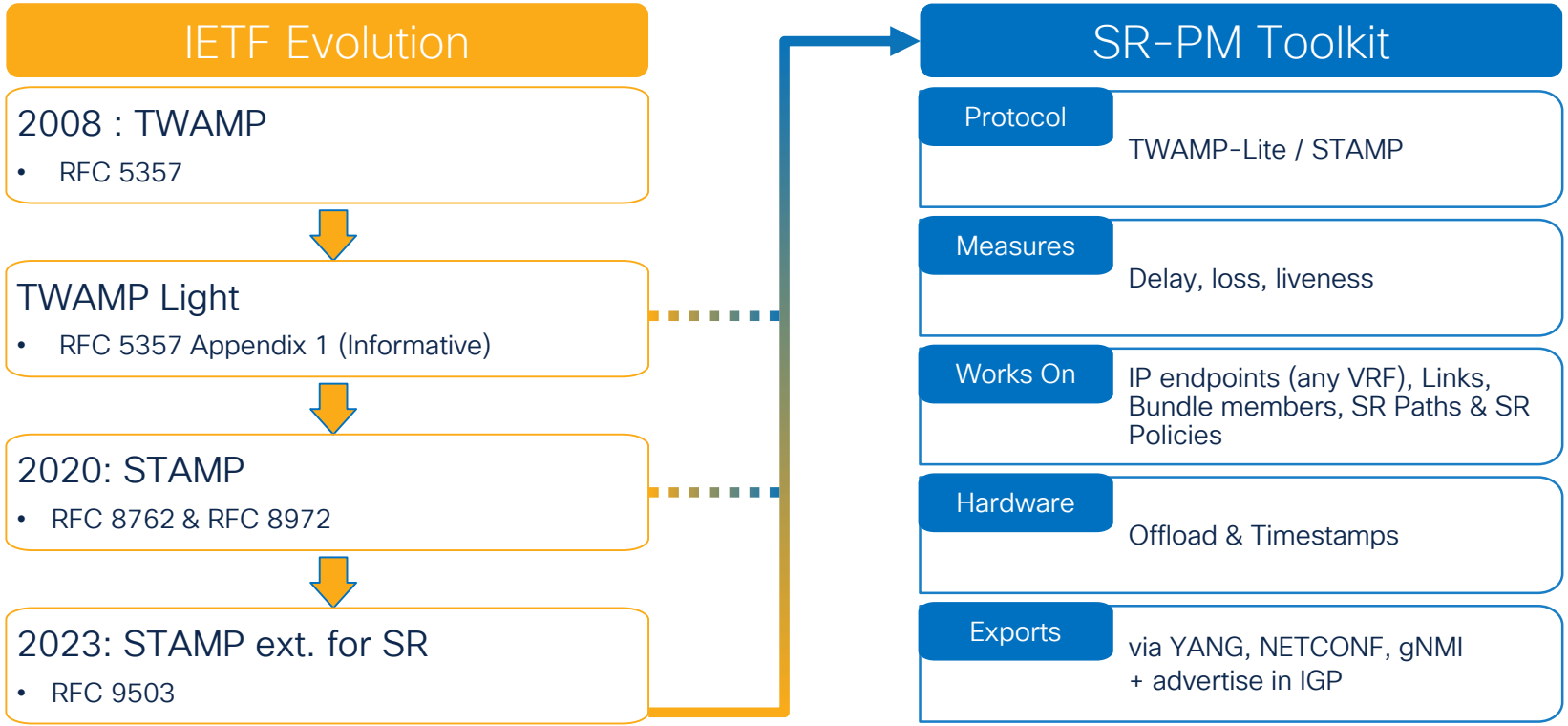
Measured link delay & loss flooded in IGP for dynamic TE computation

End to End Transport and Service Monitoring

Monitoring SRTE Policy and IP/SR Endpoint (in VRF) and EVPN

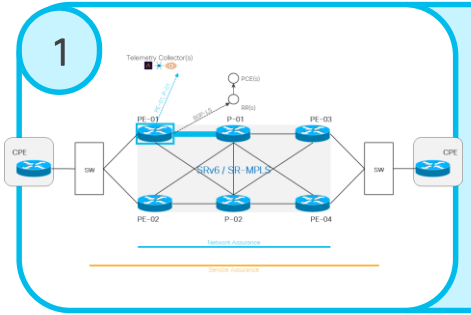
E2E/Per-Hop Delay | Loss | Liveness | Path Tracing

Cisco SR-PM is a TWAMP-based Toolkit



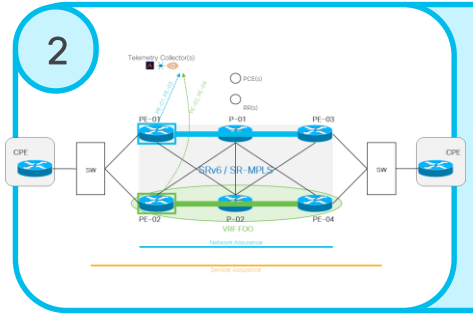
Transport Assurance Options

1



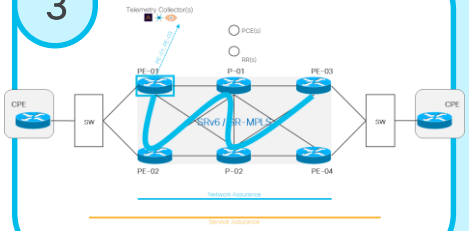
Link Measurement
Use Case: Measure the distance between PE-01 and P-01 (optical distance via HW timestamping)
Behaviors:
1. Provide min, max, avg, variance, loss via telemetry
2. ***Update IGP*** with min for routing decisions (SRTE, Flex-Algo)

2



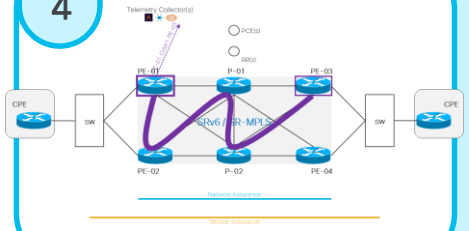
IP Endpoint Measurement
Use Case: Measurement to any endpoint (using best IGP path) can be within VRF. Note that measurement for multiple VRF will be the same
Behaviors:
1. Provide min, max, avg, variance, loss via telemetry
2. Endpoint is any TWAMP reflector

3



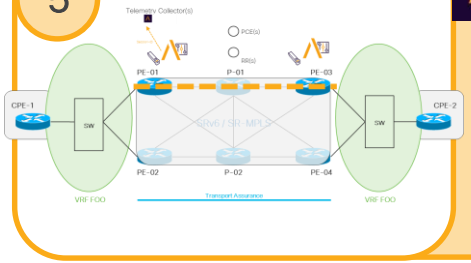
IP Endpoint Meas. via SID list
Use Case: Measurement to any endpoint <via> "some" SID list (mimic an SR policy behavior)
Behaviors:
1. Provide min, max, avg, variance, loss via telemetry
2. Endpoint is any TWAMP reflector

4



SR-TE Endpoint Measurement
Use Case: Measurement for end-to-end SR-TE policy
Behaviors:
1. Provide min, max, avg, variance, loss via telemetry
2. Endpoints any TWAMP reflector

5



SFP (Nano) Model
Use Case: Measurement to specific endpoint (using best IGP path)
Behaviors: Provide 50+ KPIs including Y.1731, Y.1564, QoE end to end
Opportunity: Collect SR-PM data via telemetry collector

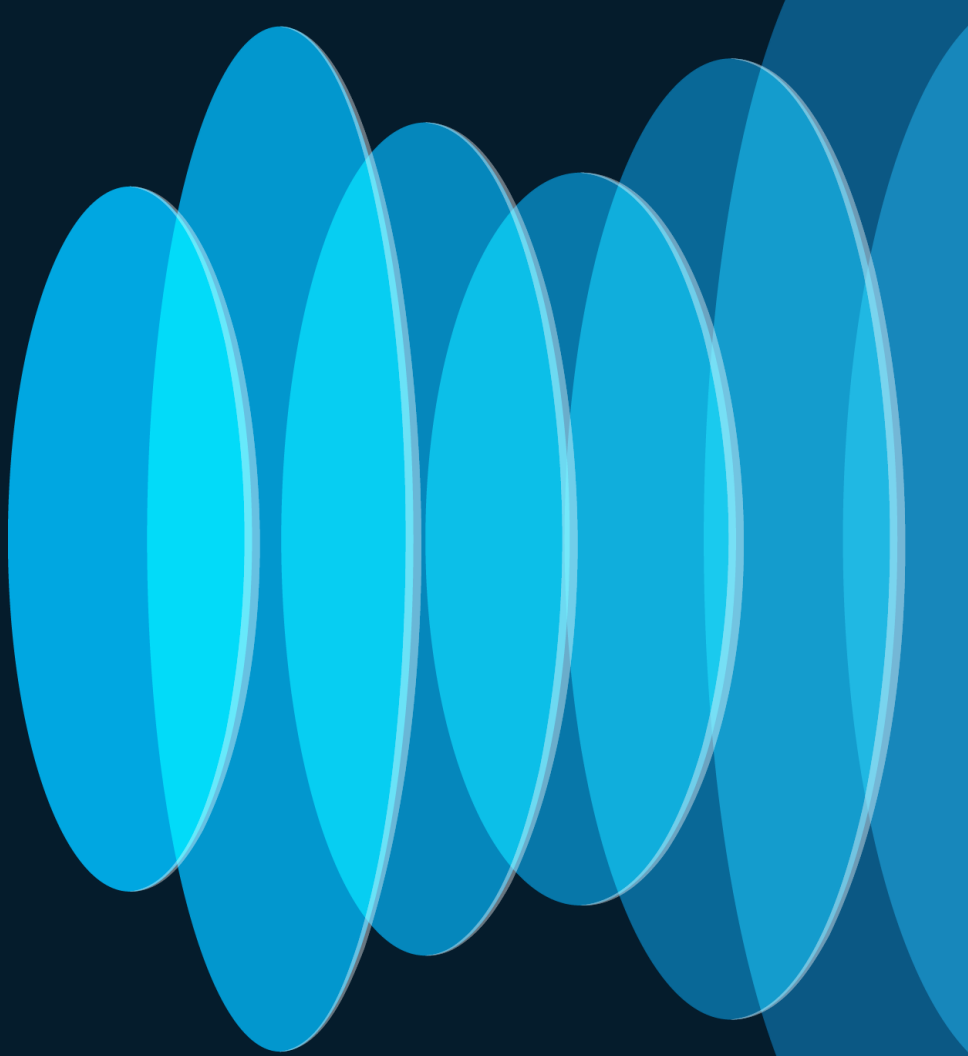
cisco Live!

SR-PM and SR data-planes

- SR-PM functionality is applicable to SR-MPLS and SRv6

Transport Assurance

SR-PM with SR-MPLS

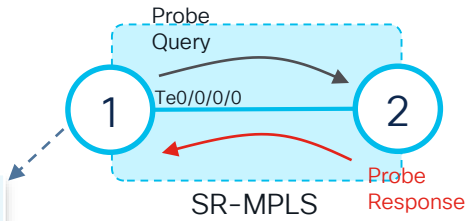


cisco Live!

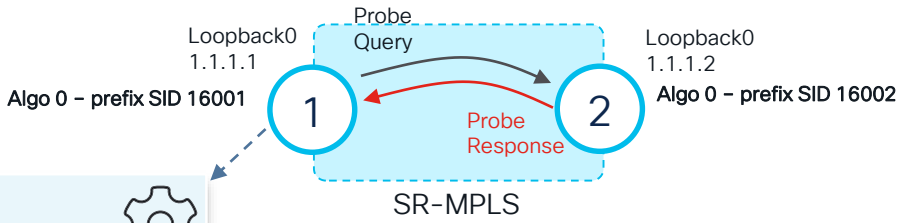
SR-PM – Interface Delay Measurement

Router 1:

```
performance-measurement
 interface TenGigE0/0/0/0
  delay-measurement
   !
 !
 delay-profile interfaces default
  probe
   measurement-mode [one-way | two-way]
```



SR-PM – IPv4 Endpoint Delay Measurement

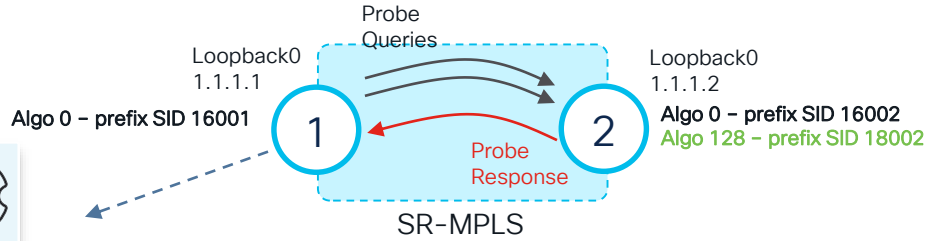


Router 1:

```
performance-measurement
 endpoint ipv4 1.1.1.2
  source-address ipv4 1.1.1.1
  delay-measurement
   delay-profile name profile-2way
  !
 !
delay-profile name profile-2way
 probe
  tx-interval 1000000
  measurement-mode [one-way | two-way]
```

SR-PM – IPv4 Endpoint Delay Measurement

Custom Segment-Lists



Router 1:

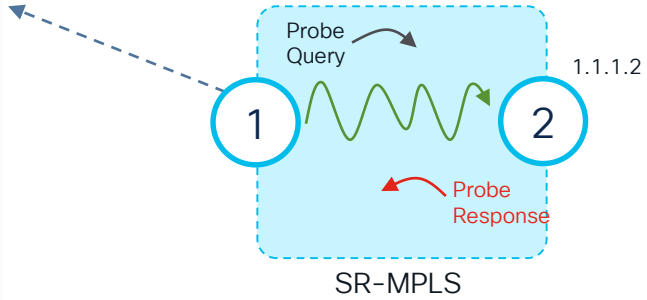
```
performance-measurement
 endpoint ipv4 1.1.1.2
  segment-list name node-2-algo-0
  !
  segment-list name node-2-algo-128
  !
  source-address ipv4 1.1.1.1
  delay-measurement
   delay-profile name profile-2way
  !
 !
 delay-profile name profile-2way
  probe
   tx-interval 1000000
   measurement-mode two-way
```

```
segment-routing
 traffic-eng
  segment-list node-2-algo-0
   index 10 mpls adjacency 1.1.1.2
  !
  segment-list node-2-algo-128
   index 10 mpls label 18002
```

SR-PM – SR Policy Delay Measurement

SR-TE HE (node 1)

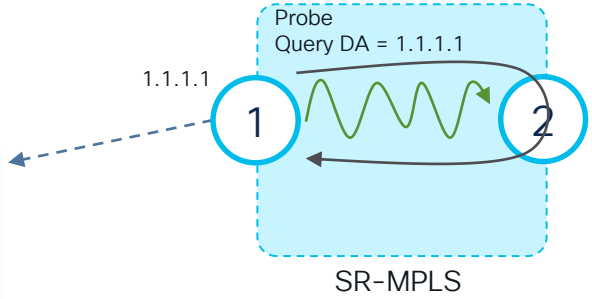
```
segment-routing
 traffic-eng
  policy pol-C_1-EP_2
   color 1 end-point ipv4 1.1.1.2
   candidate-paths
    preference 100
     dynamic
      metric
       type hopcount
      !
     !
    !
   !
   performance-measurement
    delay-measurement
     delay-profile name sample-delay-profile
```



SR-PM – SR Policy Liveness Detection

SR-TE HE (node 1)

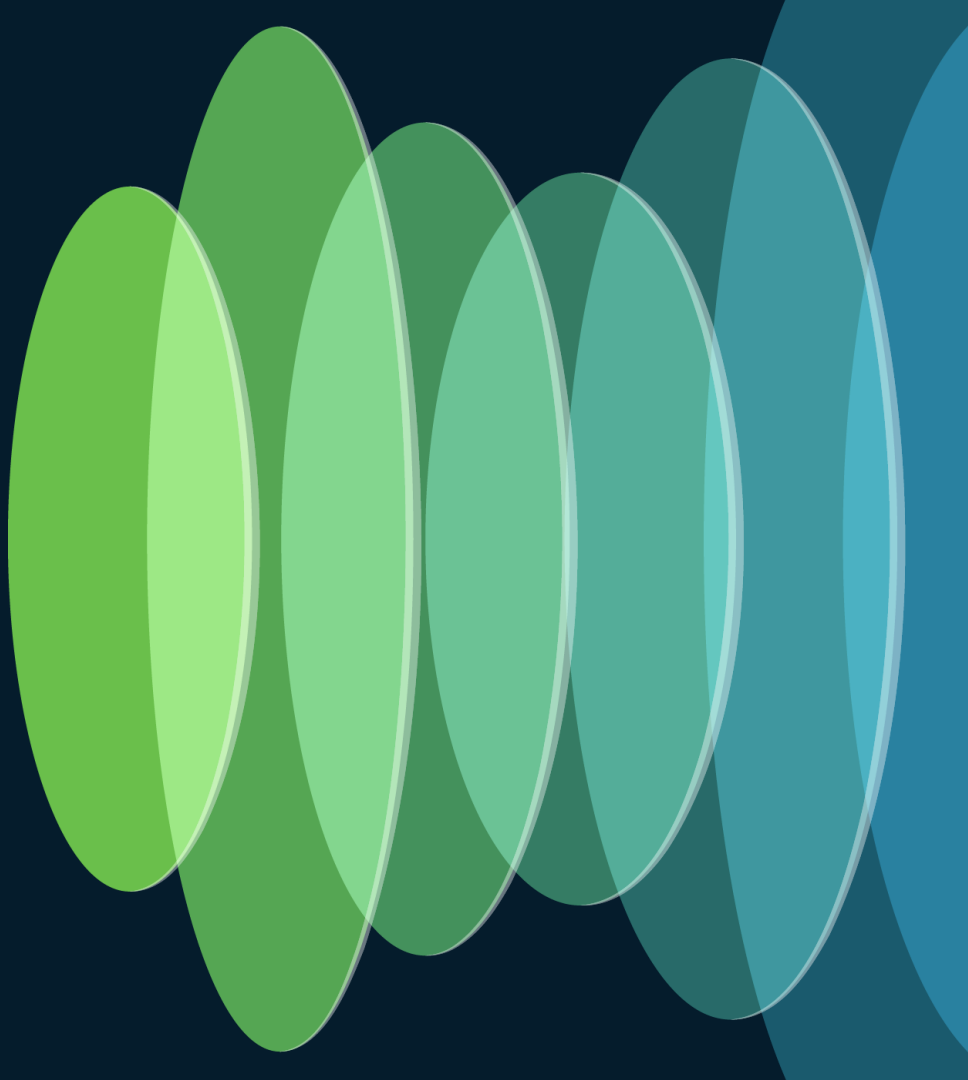
```
segment-routing
 traffic-eng
  policy pol-C_1-EP_2
   color 1 end-point ipv4 1.1.1.2
   candidate-paths
    preference 100
     dynamic
      metric
       type hopcount
      !
     !
    !
   !
   performance-measurement
    liveness-detection
     liveness-profile name sample-live-profile
```




```
performance-measurement
 liveness-profile name sample-live-profile
  liveness-detection
   multiplier 3
  !
  probe
   tx-interval 10000
```

Transport Assurance

SR-PM with SRv6

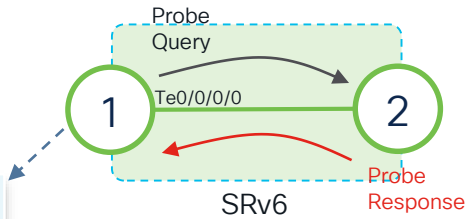


cisco Live!

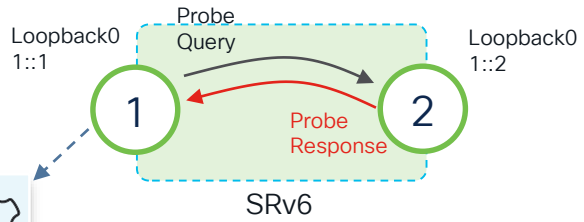
SRv6-PM – Interface Delay Measurement

Router 1:

```
performance-measurement
 interface TenGigE0/0/0/0
  delay-measurement
   !
 !
 delay-profile interfaces default
  probe
   measurement-mode [one-way | two-way]
```



SR-PM – IPv6 Endpoint Delay Measurement



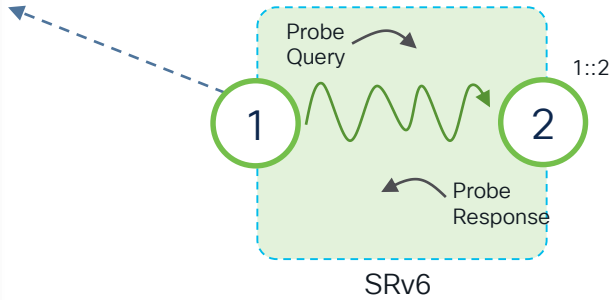
Router 1:

```
performance-measurement
 endpoint ipv6 1::2
  source-address ipv6 1::1
  delay-measurement
   delay-profile name PROFILE
  !
 !
delay-profile name PROFILE
 probe
  tx-interval 1000000
  measurement-mode [one-way | two-way]
```

SR-PM – SR Policy Delay Measurement

SR-TE HE (node 1)

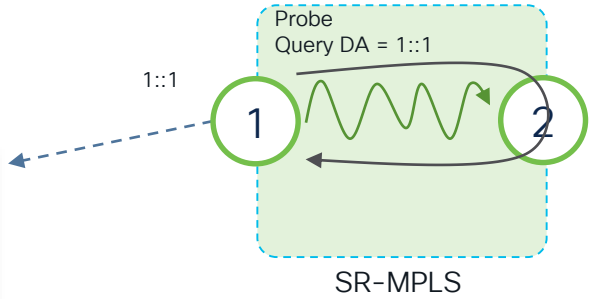
```
segment-routing
 traffic-eng
  policy pol-C_1-EP_2
   srv6
   color 1 end-point ipv6 1::2
   candidate-paths
    preference 100
     dynamic
      metric
       type hopcount
      !
     !
    !
   !
   performance-measurement
    delay-measurement
     delay-profile name sample-delay-profile
```



SR-PM – SRv6 Policy Liveness Detection

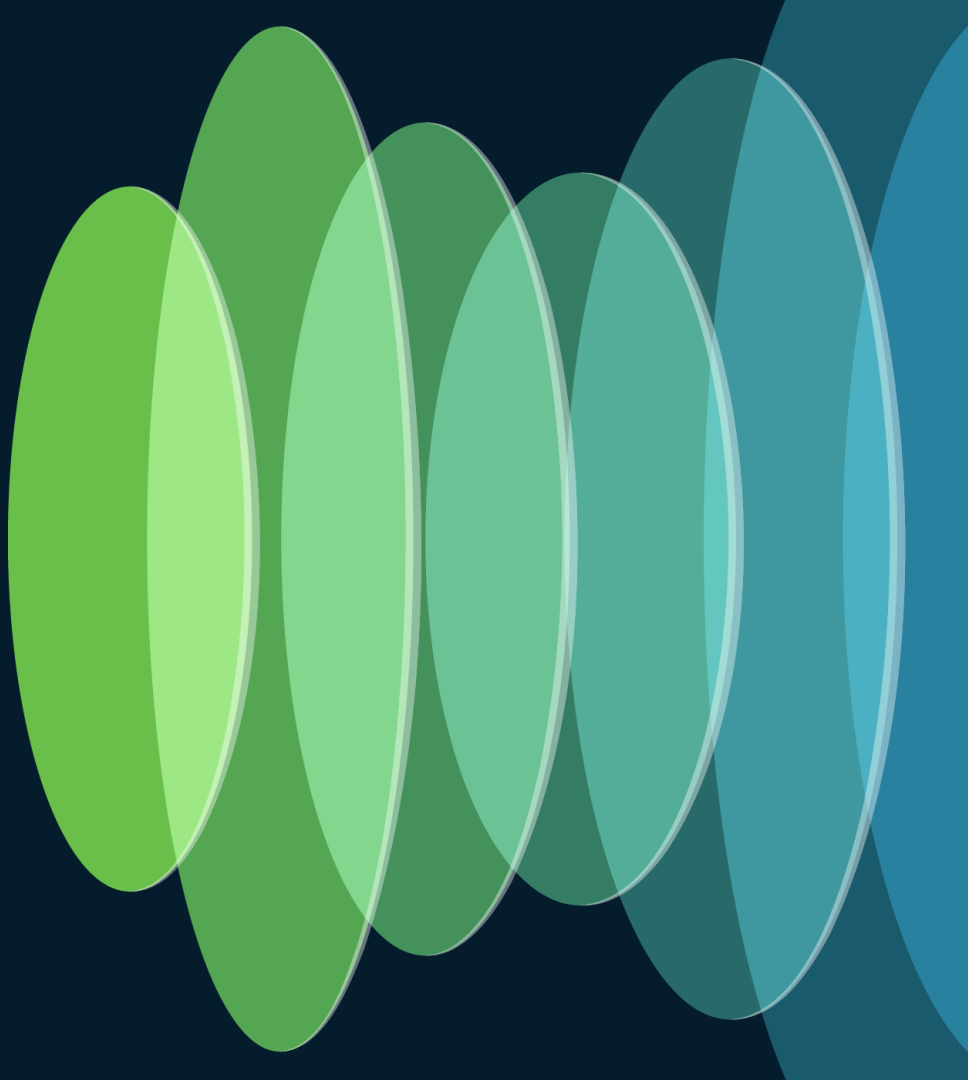
SR-TE HE (node 1)

```
segment-routing
 traffic-eng
  policy pol-C_1-EP_2
   srv6
   color 1 end-point ipv6 1::1
   candidate-paths
    preference 100
     dynamic
      metric
       type hopcount
      !
     !
    !
   !
   performance-measurement
    liveness-detection
     liveness-profile name sample-live-profile
```




```
performance-measurement
 liveness-profile name sample-live-profile
  liveness-detection
   multiplier 3
  !
  probe
   tx-interval 10000
```

Conclusion



Conclusion

- SR deployments are widespread
  - Mature standardization
  - SRv6 is hardened by common innovations and deployments from SR-MPLS
  - SRv6 brings the ultimate simplicity – just IP
- SRv6 uSID is real and de-facto-industry standard
- YOU are in control
  - Select the richer, cheaper, simpler and more reliable solution
- Simplification is Rewarding
  - Avoid distracting investments which increase complexity

Complete Your Session Evaluations

Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to **win 1 of 5 full conference passes** to Cisco Live 2025.

Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn **exclusive prizes!**

Complete your surveys in the **Cisco Live mobile app.**

SR Learning Path

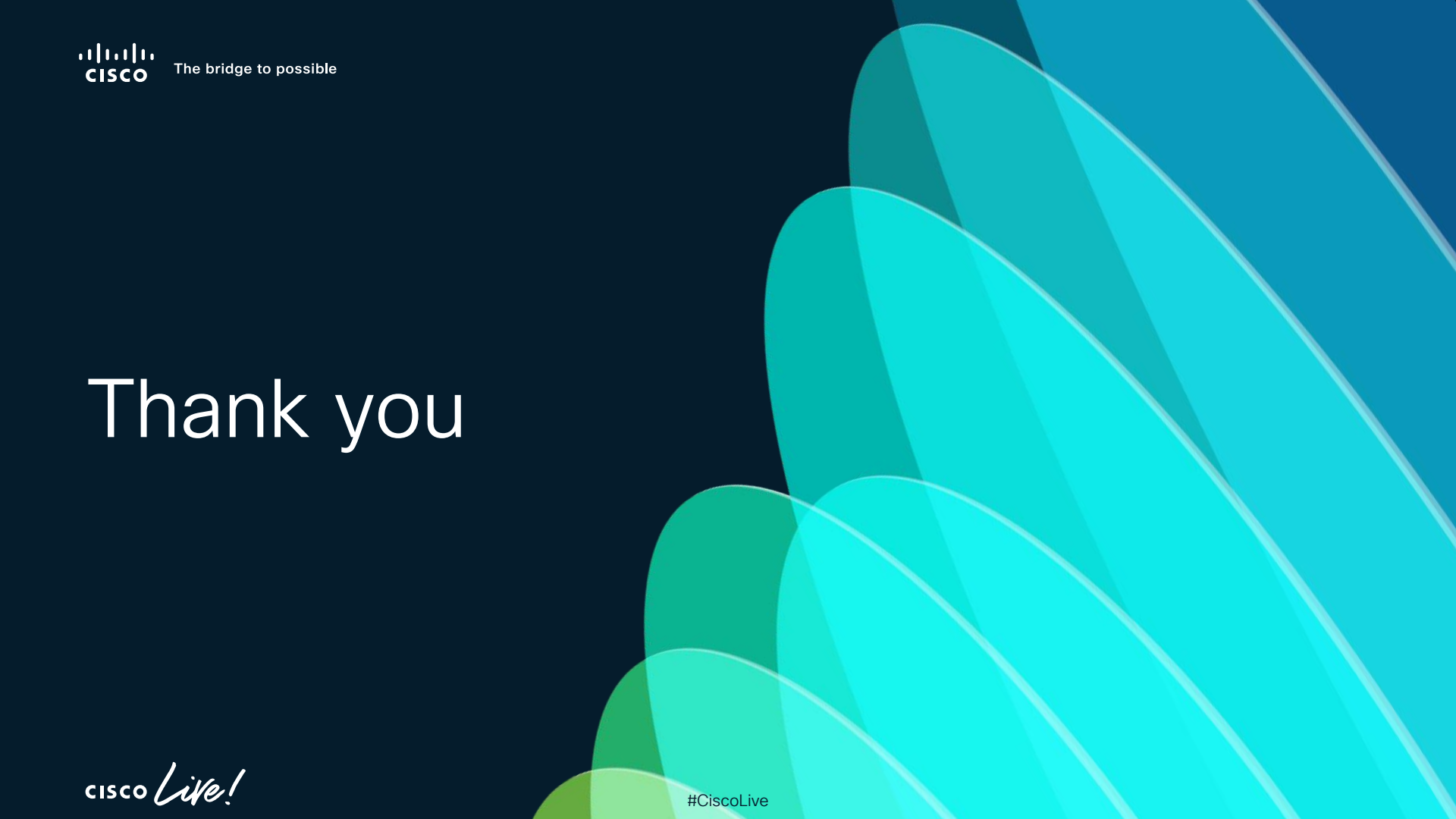
Session ID	Title	Session Type	Speakers	Schedule and location
TECSPG-1000	Segment Routing Masterclass	Technical Seminar	Jose Liste Jakub Horn	Jun 2 | 9:00 am - 1:00 pm L2, Breakers BH
BRKMPL-2203	Introduction to SRv6 uSID Technology	Breakout	Jakub Horn	Jun 3 | 10:30 am - 12:00 pm L3, South Seas B
BRKMPL-2135	Preparing for a Successful Segment Routing Deployment -	Breakout	Jose Liste	Jun 3 |10:30 am - 12:00 pm L2, Surf EF
BRKENT-1520	Segment Routing Innovations in IOS XE	Breakout	Jason Yang Sumant Mali	Jun 3 | 9:30 am - 10:30 am L3, Palm D
BRKMPL-2131	Deploying VPNs over Segment Routed Networks Made Easy	Breakout	Krishnan Thirukonda	Jun 3 | 01:00 PM / LL, Tradewinds DEF
BRKMPL-2043	Simplify Your Journey to SR and SRv6 with Cisco Crosswork Automation	Breakout	Sujay Murthy Eric Ortheau	Jun 4 | 04:00 PM / LL, Tradewinds ABC

SR Learning Path

Session ID	Title	Session Type	Speakers	Schedule and location
BRKSPG-2474	Reduced Resolution Time with Svc-centric Approach to Troubleshooting	Breakout	Paola Arosio	
LTRSPG-2006	Explore the Power of SRv6: Unleashing the Potential of Next-Generation Networking	Instructor-led Lab	Jakub Horn Marius Stoica Alex Kiritchenko	Jun 5 | 8:00 am - 12:00 pm Luxor - L1, Lotus 3
BRKMPL-2133	Circuit-Style Segment Routing and Service Emulation -	Breakout	Thomas Wang	Jun 5 | 4:00 pm - 5:00 pm L2, Surf CD
BRKSPG-2263	Design, Deploy and Manage Transport Slices using SDN Controller and Assurance	Breakout	Sujay Murthy	Jun 6 | 09:30 AM / LL, Tradewinds ABC
BRKSPG-2870	Automate Transport Service Provisioning, Optimization, and Assurance with SDN Controller	Breakout	Deepak Bhargava	Jun 6 | 01:00 PM / L3, South Seas J
LABMPL-1201	SRv6 Basics	Walk-in Lab	Luc De Ghein	
LABSP-3393	Implementing Segment Routing v6 (SRv6) Transport on NCS 55xx/5xx and Cisco 8000: Advanced -	Walk-in Lab	Paban Sarma Gautam Renjen Alexey Babaytsev	
LABSPG-3000	Configure and Implement BGP-EVPN with Segment Routing using NCS 55xx/5xx Platforms	Walk-in Lab	Tejas Lad Paban Sarma	

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand


The bridge to possible
Thank you
cisco Live!
#CiscoLive