Segment Routing Bootcamp

Prerequisite

Please read www.segment-routing.net/tutorial

SR Concepts >
- SR Introduction
- SR MPLS data plane
- SR Global Block (SRGB)

SR IGP >
- SR IGP Control Plane
- Topology Independent LFA (TI-LFA)

SR and LDP >
- SR and LDP co-existence
- SR Mapping Server
- SR and LDP interworking

SR TE >
- SR TE

Please view www.segment-routing.net/demos

SR On-Demand Next Hops (ODN)
SR TI-LFA
SR Microloop Avoidance

Please view
https://www.youtube.com/watch?v=EUxm9EH1wyQ&t=636s

Segment Routing - Deployment Experience and Technology Update

Agenda of Day1 and Day2

Day-1

- Cisco SD IP Network Framework
- Segment Routing (SR) - TI-LFA
- SR Mapping Server
- Use Case P-PE (SR + LDP/RSVP-TE)
- Lab Hands-On

Lunch & Pray

- SR Migration
- Use-case P(SR) + PE(SR)
- SR Policy Concept
- Lab Hands-On

Day-2

- VPN SR TE based on Low Latency/ Constraint/WECMP
- SR PCE - BGP LS + PCEP
- Use-Case Seamless MPLS - BGP LU
- Lab Hands-on

Lunch & Pray

- SR On Demand NextHop (ODN)
- SR PCE Rest API
- Lab Hands-on
- Demo SR ODN + SR PCE + NSO Automation

Housekeeping

Please introduce yourself : Name – Company – Role

Please access and reserve http://dcloud.cisco.com
Cisco WAN Automation Engine 7.1.1.1 - Segment Routing and XTC Sandbox v1

Please access dcloud vpn and test Telnet Session to every nodes

Please have python installed in the laptop

And respect others, please be punctual and set HP to silent mode

SR Champion and SR Redelivery Session

Congratulations ! YOU are Partner SR Champion

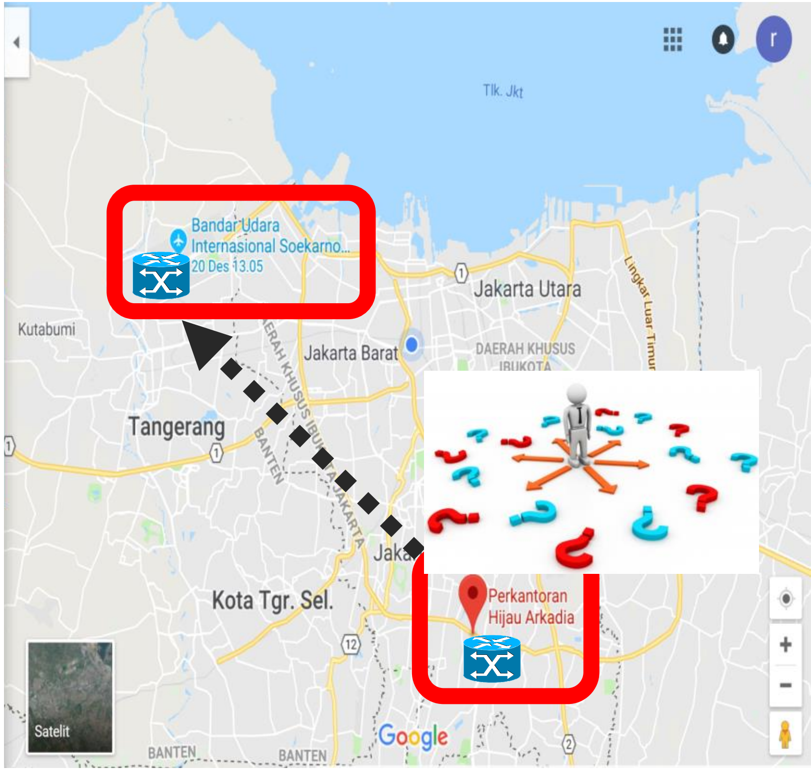
We will need your help to run SR Redelivery Session to your colleague engineer

Every 2 SR Champions will deliver 1 SR Redelivery Session

Please advice time and place

Cisco SD IP Network Framework

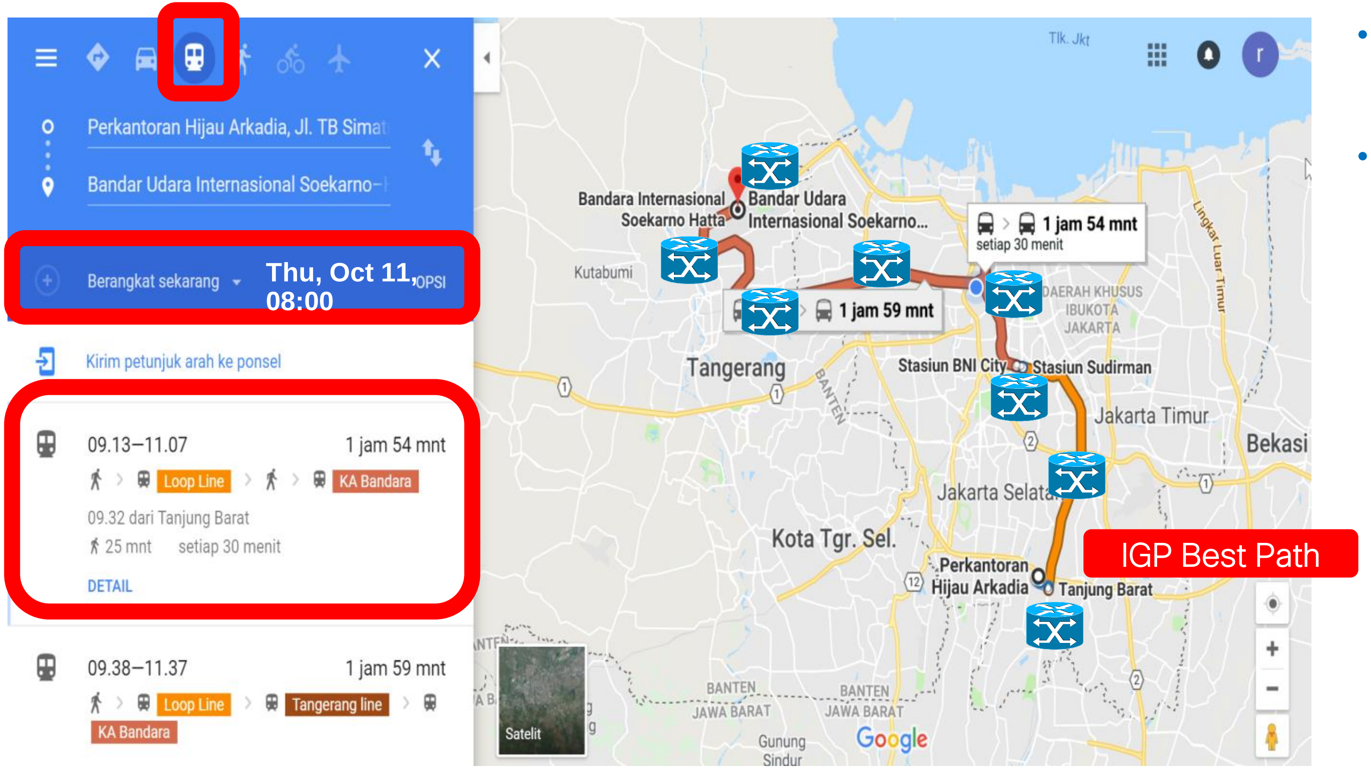
Objective : Go to the Airport ASAP (Business Requirement)



- What is the Best Path ?
- Alternative Path ?
- How is Situation along the Path ?
- What if the Path is Blocked ?
- How about Tomorrow Path ?

Objective : Go to the Airport ASAP (Business Requirement)

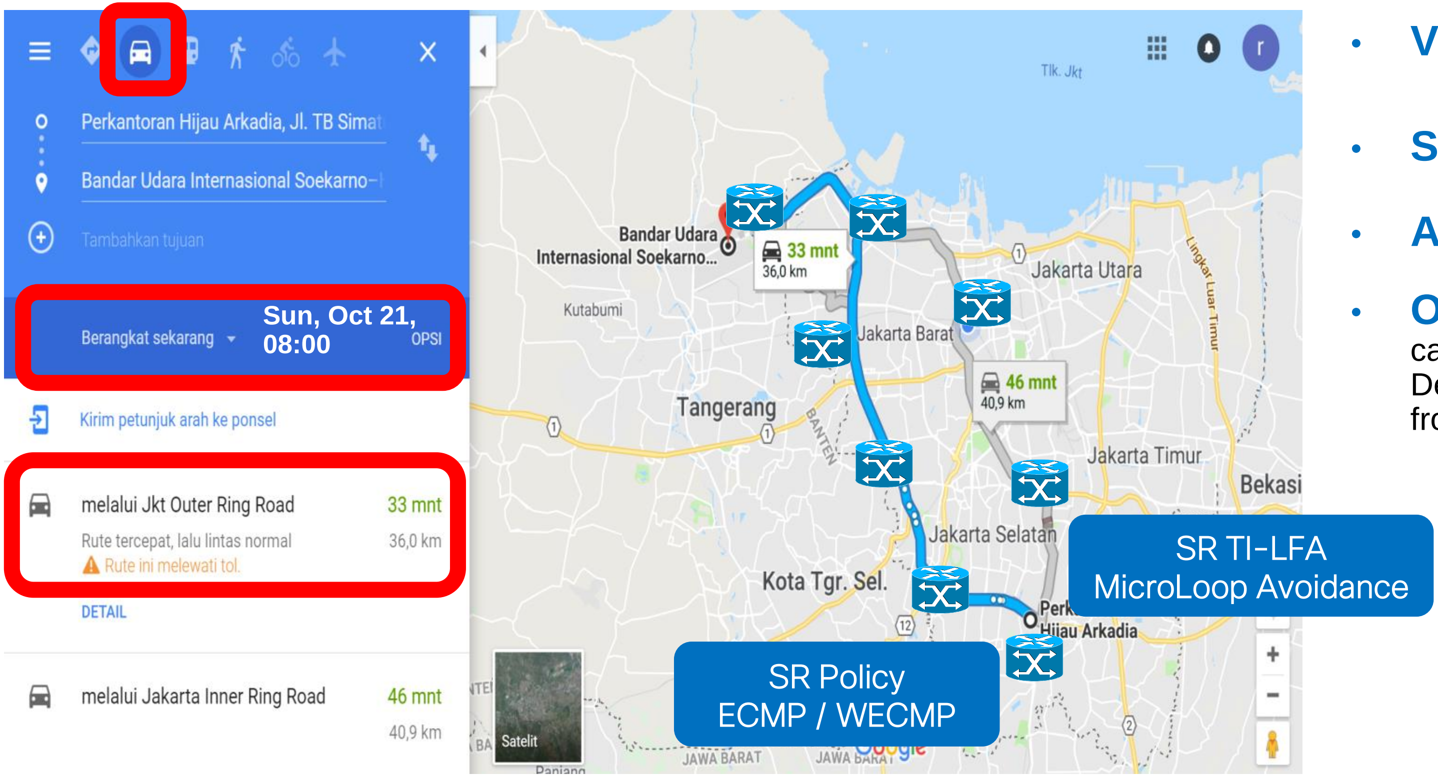
How is the Path From A (Telkom) to Z (Airport) NOW ? (Plan/Design)



- **Visualization** for overall network visibility
- **Simplification** for only highlight On-Demand Network Path from A to Z

But… it takes long time. (Latency Analysis)

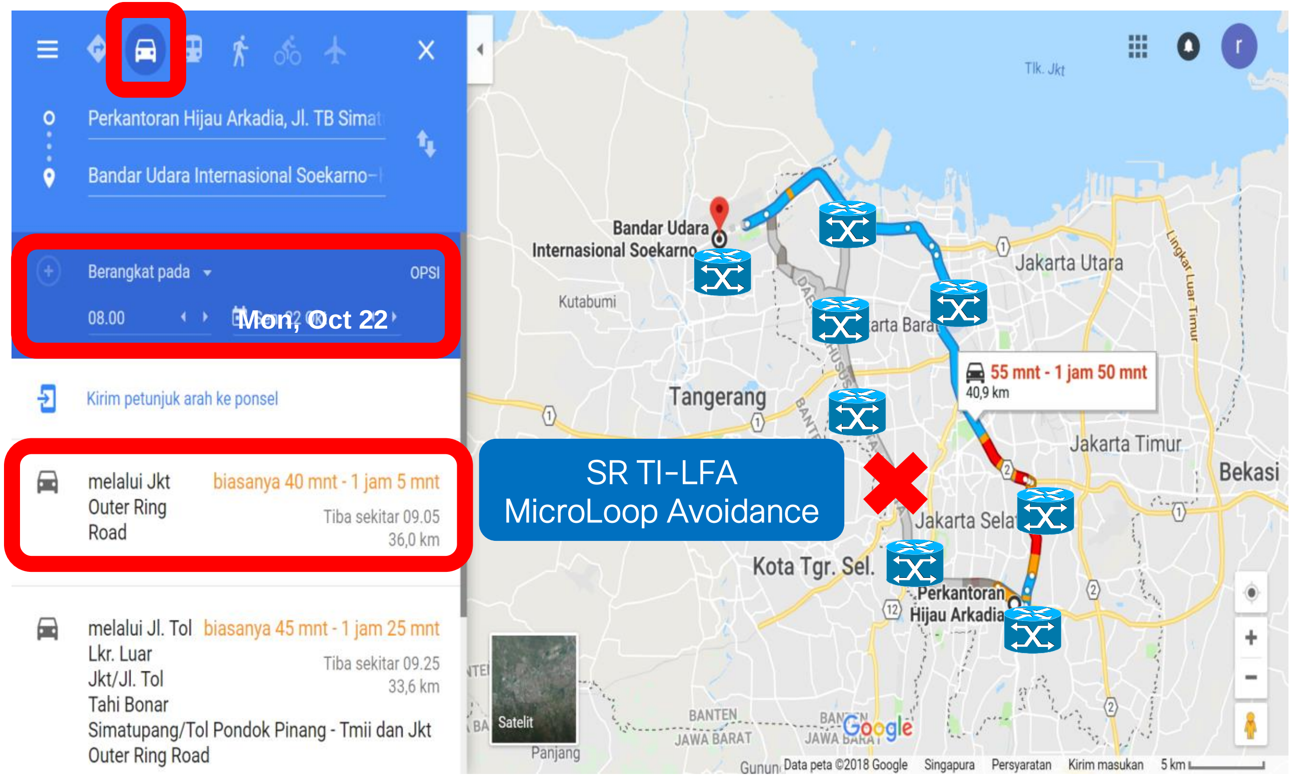
I need shorter time. How to get faster and better Path ? (Optimization)



- **Visualization**
- **Simplification**
- **Analysis**
- **Optimization** for calculating better On-Demand SLA Route Path from A to Z

Good Optimization, …. What if the Path is blocked ? ☹ (Simulation)

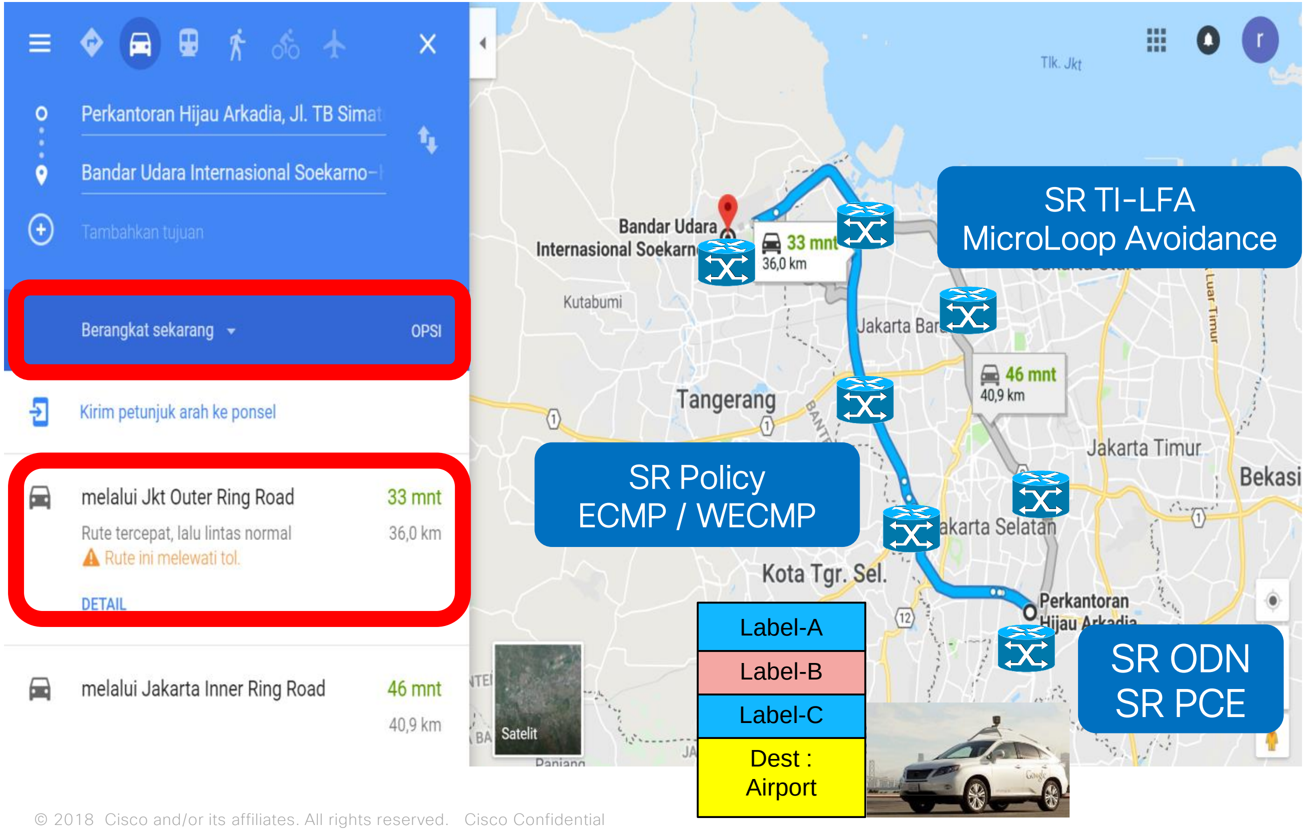
What will the Path Condition be NEXT DAY ? (Prediction)



- **Visualization**
- **Simplification**
- **Analysis**
- **Optimization** for best SLA Path Requirement
- **Simulation** for critical failure analysis and solutions
- **Prediction** for network traffic growth planning

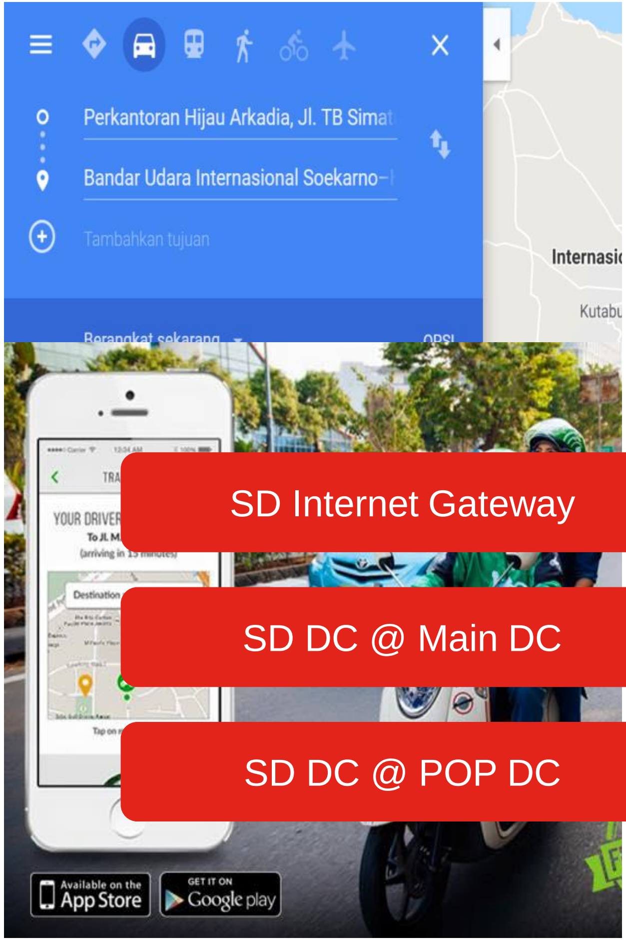
Next Level Transport : Automated Intelligent Transportation

(Segment Routing + SDN Automation with Intent-based Networking)



- **Visualization**
- **Simplification**
- **Analysis**
- **Optimization**
- **Simulation**
- **Prediction**
- **On-Demand** for business SLA requirement fulfillment

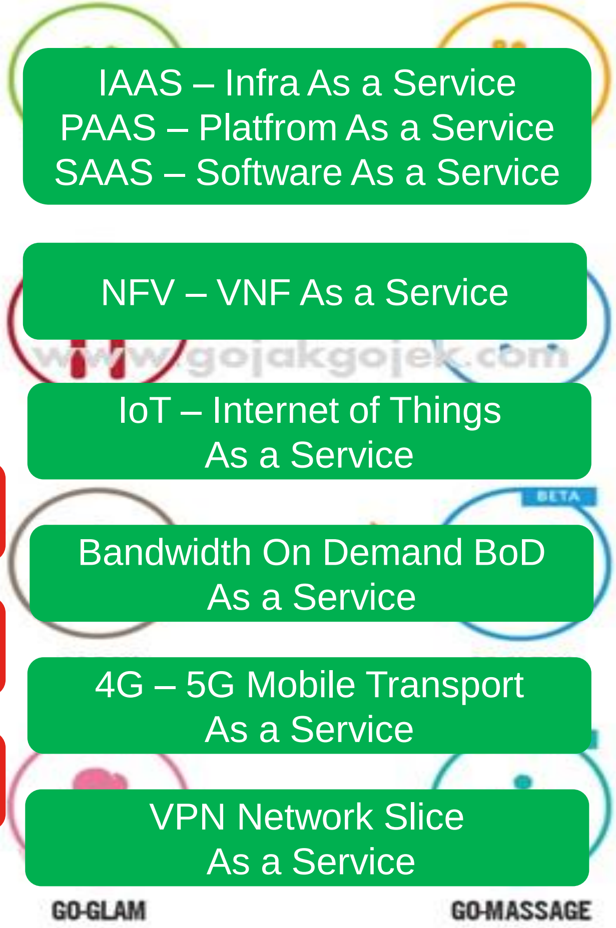
More Digital Value Add Service … More Revenue with Automation



SD Internet Gateway

SD DC @ Main DC

SD DC @ POP DC



IAAS – Infra As a Service
PAAS – Platfrom As a Service
SAAS – Software As a Service

NFV – VNF As a Service

IoT – Internet of Things As a Service

Bandwidth On Demand BoD As a Service

4G – 5G Mobile Transport As a Service

VPN Network Slice As a Service

- **Visualization** for overall network visibility
- **Simplification** for network complexity
- **Analysis & Simulation** for critical failure analysis and solutions
- **Prediction** for network traffic growth planning
- **Optimization** for better SLA network performance
- **On-Demand** for instant business SLA requirement fulfillment
- **Value Add Service** for more revenue generation

Cisco SD IP Network

Scalable Platform

- Scalable fabric designs to accommodate elastic network service expansion
- Granular building blocks using high-density and efficient Cisco routers

Network Simplification

- Segment Routing – EVPN into building blocks with simplified feature sets
- Increased visibility and path control with XTC SR PCE
- Consistent operational model

SDN Automation

- NSO enables modern Service Automation to transform network operations
- Common automatable tasks prescribed
- Standards-based YANG models used for configuration and state

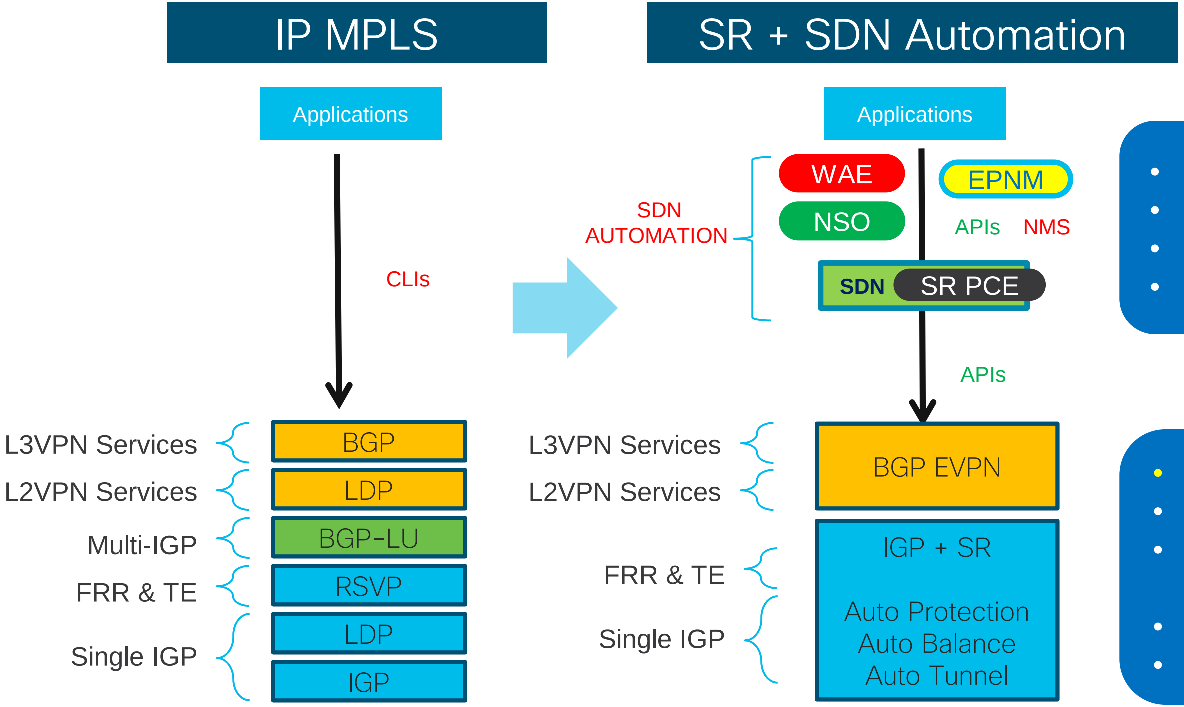
Analytic Telemetry

- WAE enables Proactive Analytic, Simulation and Predictive Network Modeling
- Rich streaming telemetry available for all protocols using standards-based YANG models
- Unlock proactive fault detection and remediation

Validated

- Network design and its component features tested individually and as a system
- design is continually evaluated as a component of end to end network

IP MPLS Evolution to Segment Routing + SDN Automation



Benefit

- End to End Visibility & Analysis
- Simulation & Prediction
- Optimization
- On-Demand Automation

- > 50% Simplification Protocol
- Simplied Route Table with SR-ODN
- Automated sub-50ms Fast Reroute Protection
- Automated Microloop Avoidance
- Automated ECMP Load Balance

Segment Routing: Key capabilities

Foundation for Network Operations Simplification and Automation,
and Service Agility via Programmability

Simplification of network protocols
Improved scalability (3 to ONLY 1)

Automated 50ms convergence

Simplification of Traffic Engineering

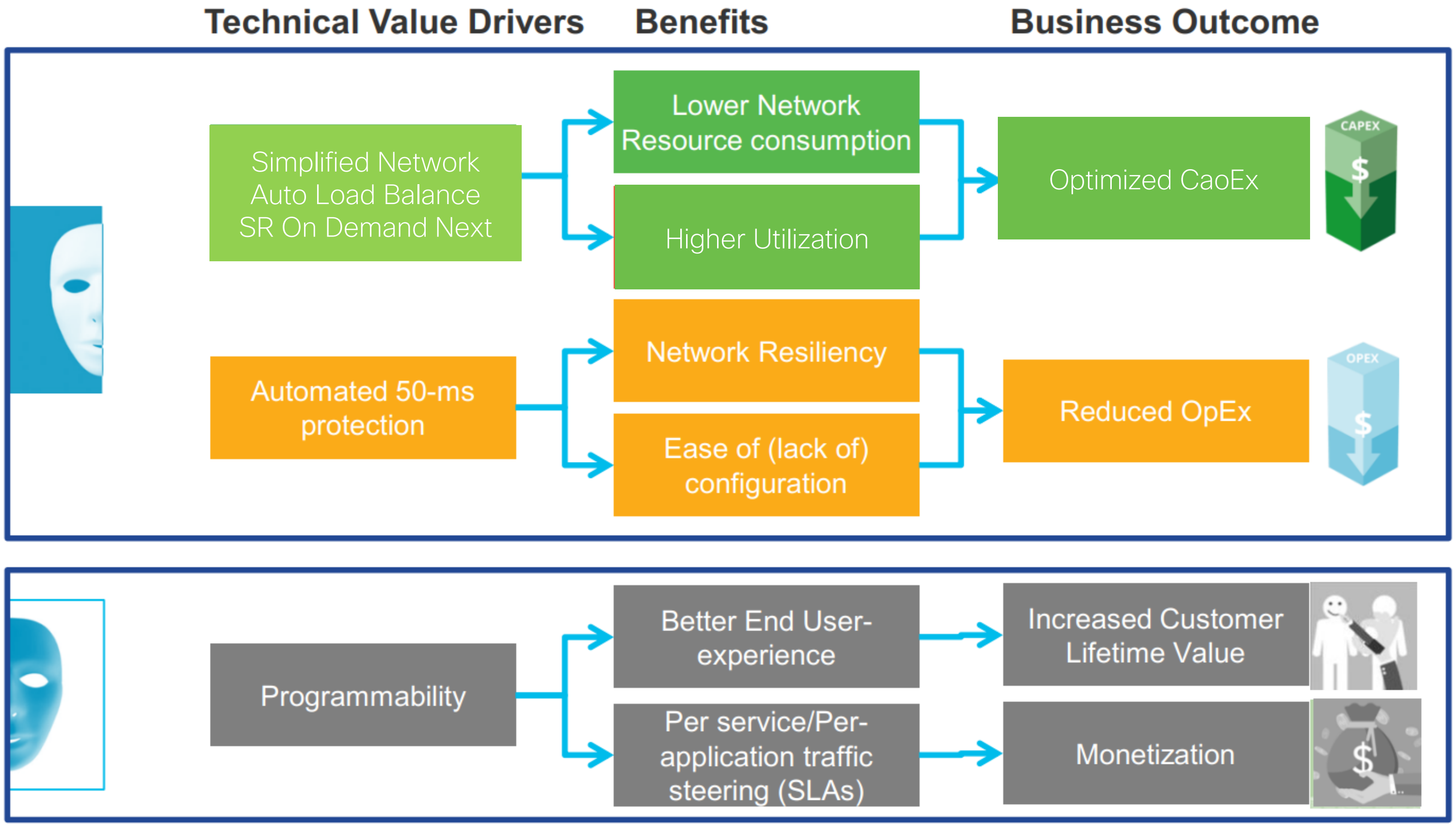
Built-in Redundancy & HA

Application enabled policy using SDN techniques

Support MPLS & v6 Forwarding

Universal Forwarding Plane
From Access to DC

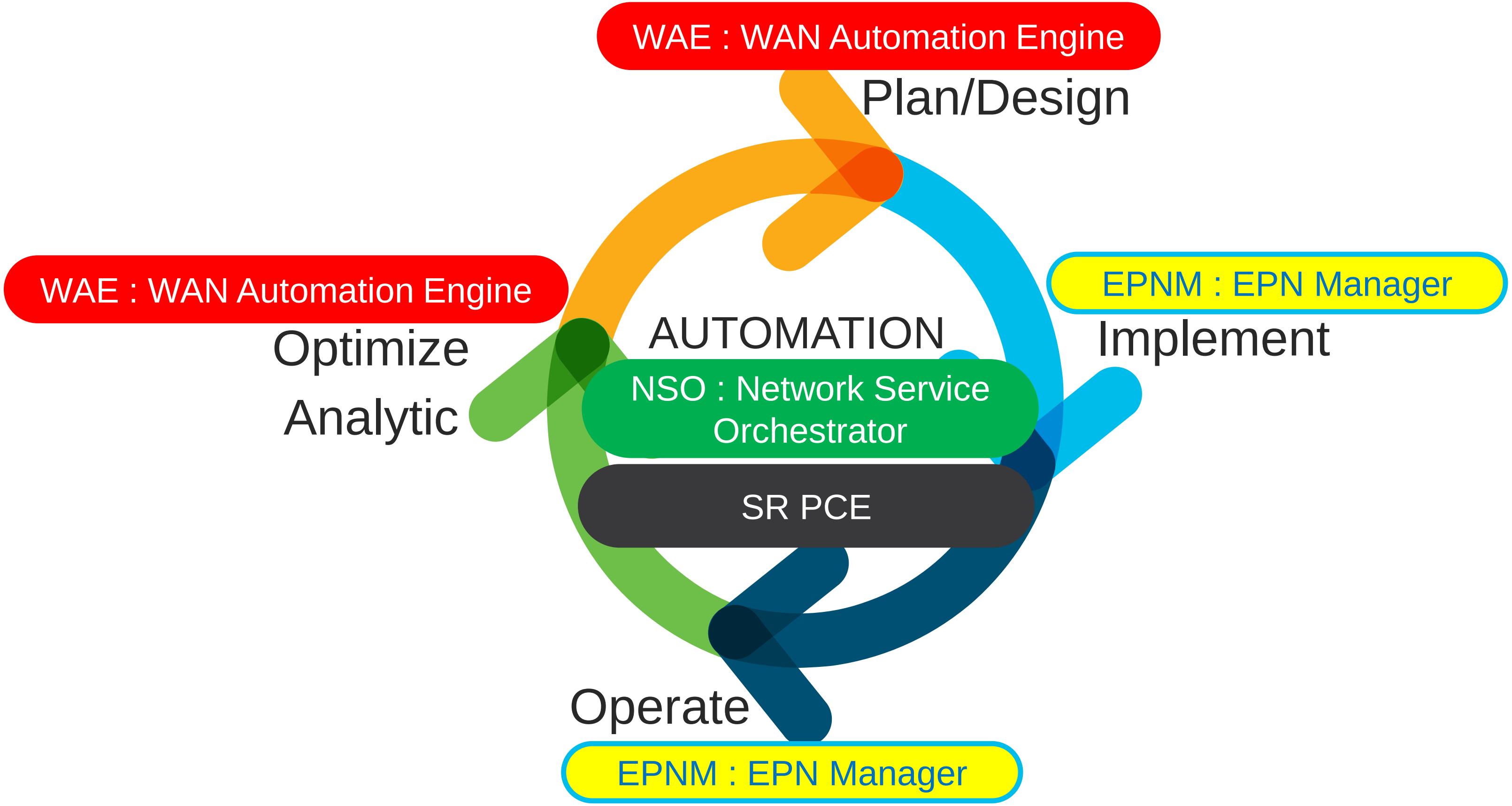
Segment Routing Value Proposition



Segment Routing vs RSVP-TE vs LDP

Use Case	SR	TE	LDP
Operational Simplicity	✓	✓	✓
Fast Reroute	✓	✓	✓
ECMP	✓	✗	✓
Traffic Engineering	✓	✓	✗
Multi Domain TE	✓	✓	✗
Network Slicing	✓	✓	✗
On-Demand-Next Hop	✓	✗	✗
Intent Based Automated Steering	✓	✗	✗

Automation in Network Lifecycle

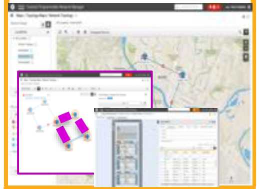


Automation in Network Lifecycle

EPNM

Evolved Programmable Network Manager

Detect
Correlate
Isolate



Dynamic Inventory

"Operate" service, network & device resources as the single point of truth for all operations

NMS

WAE

WAN Automation Engine

Optimize

BW & Topology Optimization

Traffic planning, design & optimization within the network

ANALYTIC SIMULATION

SR PCE

XR Transport Controller

Path Computation Element

Model Driven Orchestration

Real Time Path Compute and Topology

SDN CONTROLLER

NSO

Network Service Orchestrator

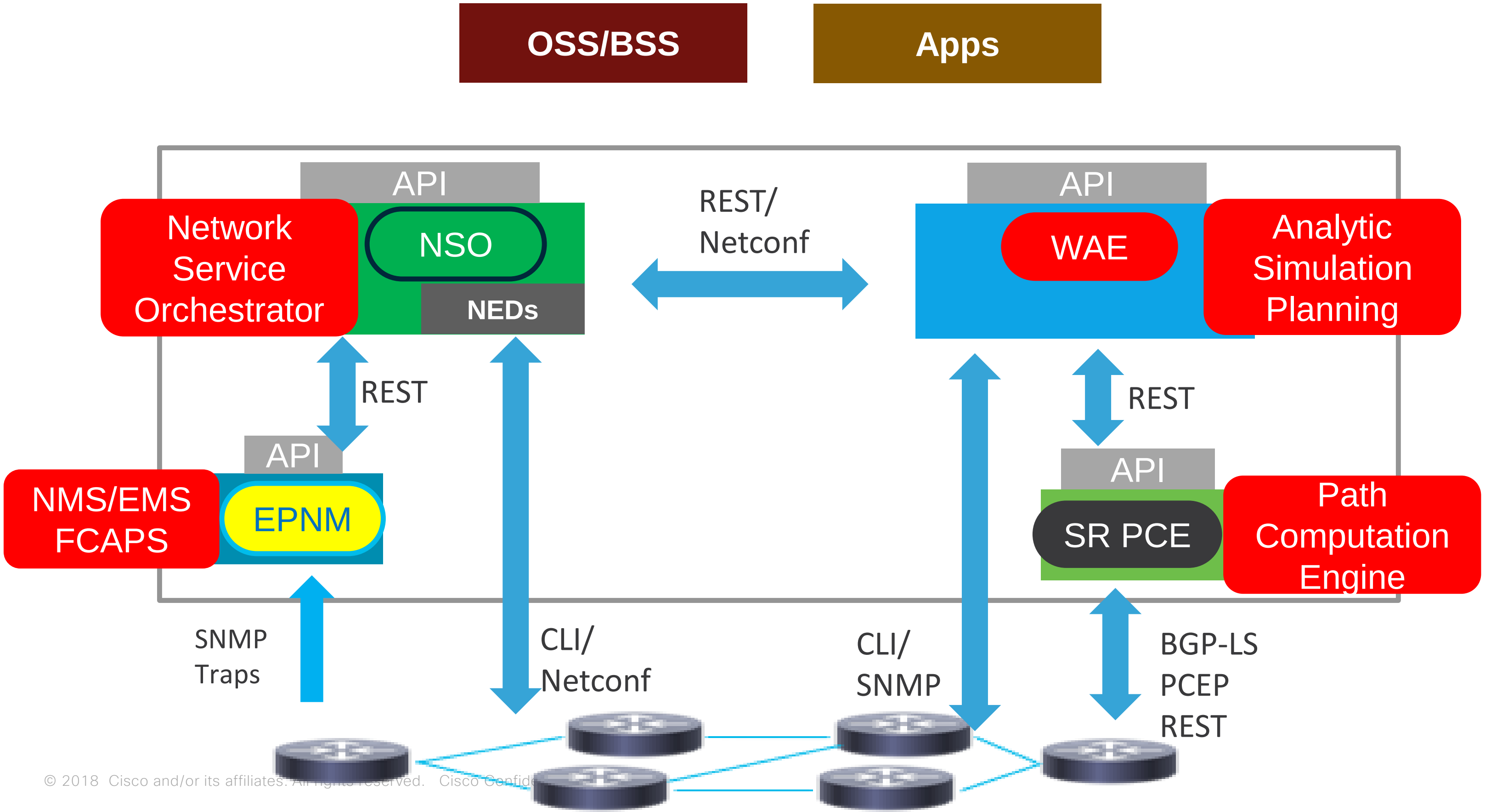
Deploy

Model Driven Orchestration

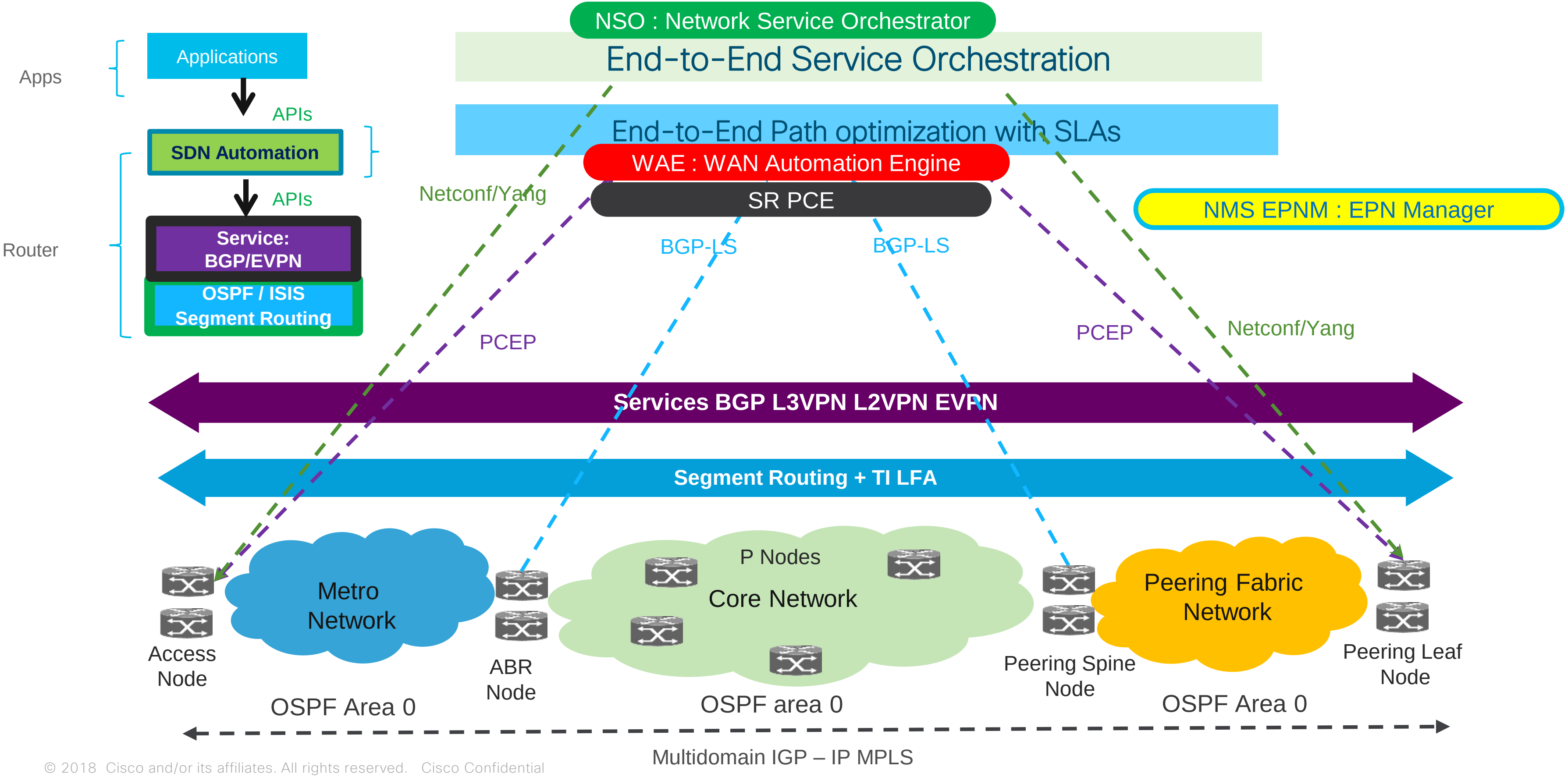
Implement Agile Service Orchestration for Across WAN, DC, NFVI

ORCHESTRATOR

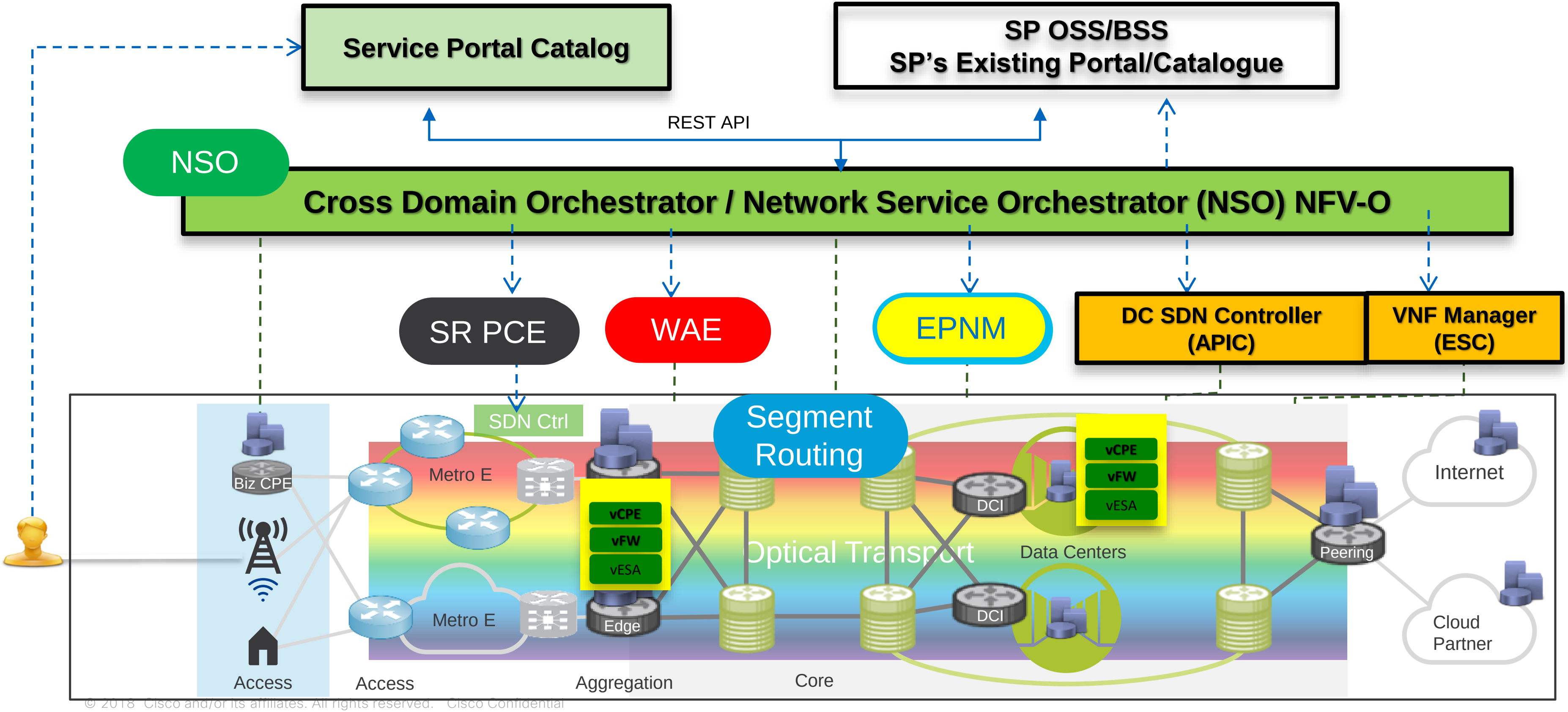
Integration Architecture



Segment Routing + SDN Automation



SDN Automation Framework



Segment Routing – SPRING RFC 7855
Clarence Filsfils@Cisco

Internet Engineering Task Force (IETF)
Request for Comments: 7855
Category: Informational
ISSN: 2070-1721

S. Previdi, Ed.
C. Filsfils, Ed.
Cisco Systems, Inc.
S. Litkowski
Orange
M. Horneffer
Deutsche Telekom
R. Shakir
Jive Communications, Inc.
May 2016

Source Packet Routing in Networking (SPRING)
Problem Statement and Requirements

Abstract

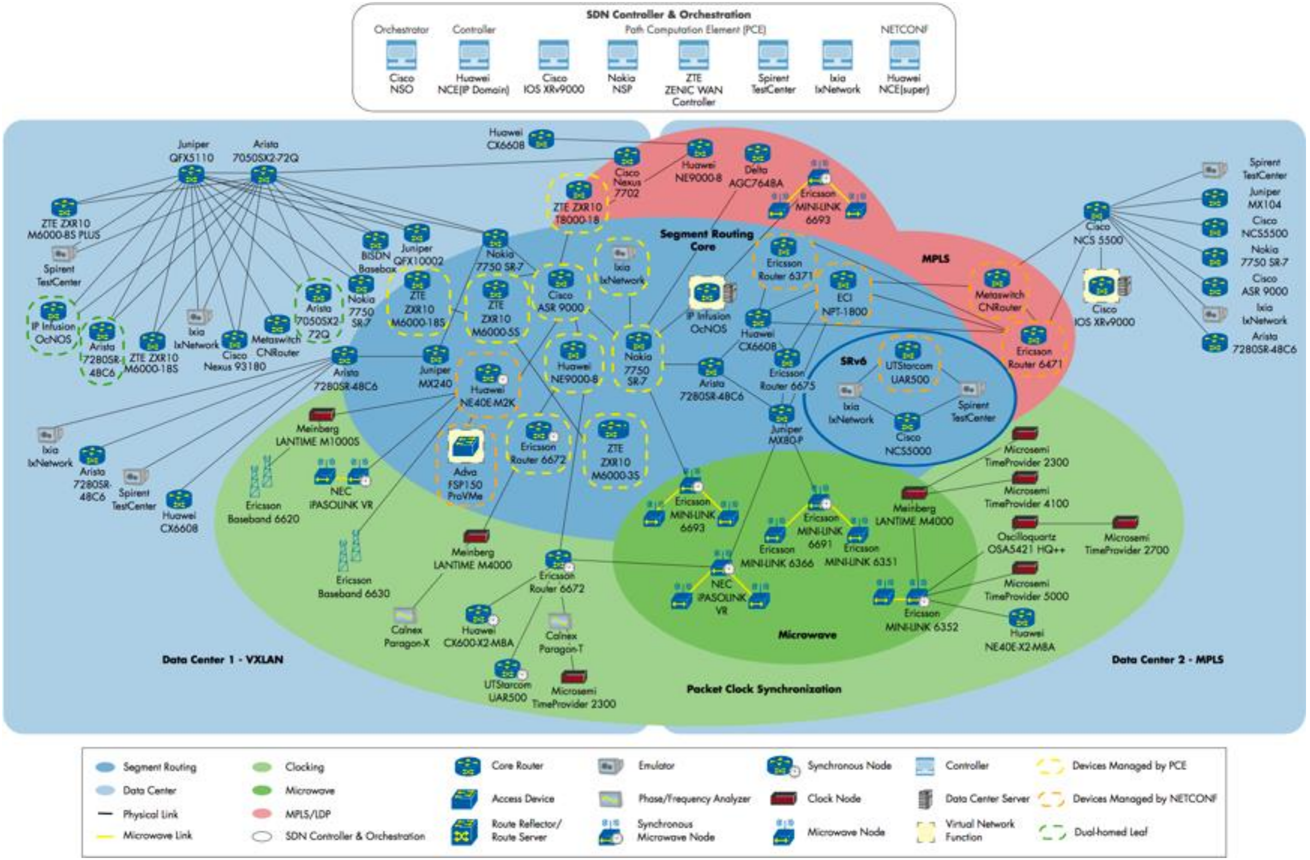
The ability for a node to specify a forwarding path, other than the normal shortest path, that a particular packet will traverse, benefits a number of network functions. Source-based routing mechanisms have previously been specified for network protocols but have not seen widespread adoption. In this context, the term "source" means "the point at which the explicit route is imposed"; therefore, it is not limited to the originator of the packet (i.e., the node imposing the explicit route may be the ingress node of an operator's network).

Segment Routing + SDN Interop Showcase

- Segment Routing
- Segment Routing Traffic Engineering
- SDN Controller XTC
- Topology Independent LFA
- Segment Routing and LDP Interworking
- BGP Prefix Independent Convergence (PIC)
- Stateful PCE Path Computation Element
- Network Provisioning with NETCONF

EANTC

EANTC MPLS + SDN + NFV World Congress
Public Multi-Vendor Interoperability Test 2018 Scope



- Segment Routing Test Cases ✓
- Ethernet VPN Test Cases
- SDN Test Cases ✓
- Microwave Test Cases
- Clock Synchronization Test Cases

www.segment-routing.net for tutorial & demos

Not secure | www.segment-routing.net/tutorials/

Not secure | www.segment-routing.net/demos/

SEGMENT ROUTING NEWS DEMOS TUTORIALS

Tutorials

› SR concepts

Learn about the foundational SR concepts.

› SR IGP

The SR IGP control plane and automated fast-reroute resiliency.

› SR and LDP

Introducing SR in an LDP network, how they can co-exist and interwork.

› SR Traffic Engineering

SR-TE provides you low-delay paths, disjoint paths,... and many more.

SR On-Demand Next Hops (ODN)

Watch Jose Liste's demonstration of SR ODN. Learn how S
metro to backbone at scale.

Jose Liste, Cisco – November 08, 2016

Topology Independent - Loop Free Alternate (TI-LFA)

Watch Kris Michielsen's demonstration of TI-LFA. TI-LFA
topology, along the optimum backup path, against link, node

Kris Michielsen, Cisco – November 07, 2016

Microloop Avoidance

Watch Kris Michielsen's demonstration of Microloop Avoida
microloops, a day-1 problem of IP networks.

Kris Michielsen, Cisco – November 06, 2016

https://xrdocs.github.io/design/

@XRDOCS DESIGN BLOGS TUTORIALS SEARCH TAGS ARCHIVE

Network Designs

Service Provider deployments have consistently evolved for the last few decades and keeping abreast of the changes can get hard. Let the best in the industry walk you through Network Designs and Architectures that will lay the foundation for the next-gen SP networks

Get Started! Github

Metro Fabric Design

Read about the goals of the Metro Fabric Design and familiarize yourself with all its components: the platforms, the software, the technologies and use cases

High Level Design Implementation Guide

EPN 5.0 Design Guide

https://www.cisco.com/c/en/us/solutions/enterprise/design-zone-service-provider/programmable-network.html#~tab-epn-5guides

https://www.cisco.com/c/en/us/solutions/enterprise/design-zone-service-provider/programmable-network.html#~tab-...

EPN 5 Validated Design Guides

EPN 5.0 Transport Design Guide (PDF - 5.6 MB)

EPN 5.0 Services Design Guide (PDF - 2.7 MB)

EPN 5.0 System Test Topology Reference Guide (PDF - 2 MB)

EPN 5.0 Network Service Orchestration User Guide (PDF - 2.3 MB)

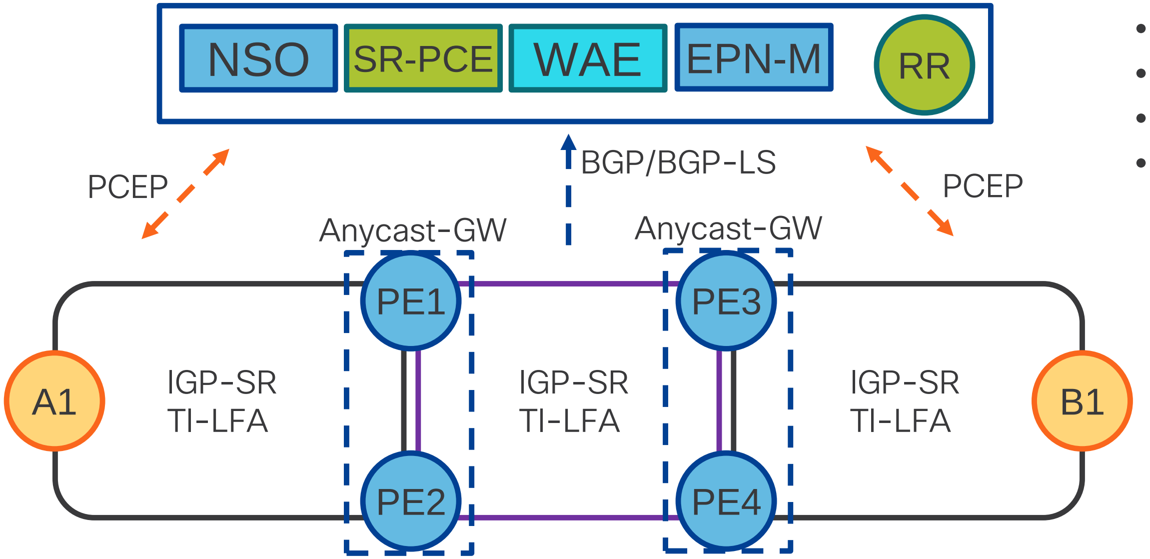
EPN 5.0 Implementation Guide for Large Network with End to End Segment Routing (PDF - 2.1 MB)

EPN 5.0 Implementation Guide for Small Network with End to End Segment Routing (PDF - 2 MB)

EPN 5.0 Implementation Guide for Large Network with End to End Programmable Segment Routing (PDF - 2 MB)

EPN 5.0 Implementation Guide for Large Network with Segment Routing and LDP Interworking (PDF - 2.1 MB)

EPN 5.0 Framework



- Segment Routing adoption
- EVPN - Common Control-Plane
- SR/LDP Seamless integration
- Automation

Controller/Orchestration:

- SR-PCE Provides inter-IGP domain path, WAE - WAN optimization
- NSO Service provisioning
- EPN-M Service provisioning/Service assurance/Network Visibility/Fault Management

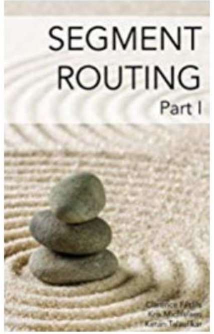
Services: BGP - L3VPN/L2VPN(EVPN), point-to-point services static PW

Transport: Segment-Routing with Topology Independent LFA (TI-LFA)

Segment Routing Recommendation Books

https://www.amazon.com/gp/product/B07N6H54VP/ref=series_rw_dp_sw

1

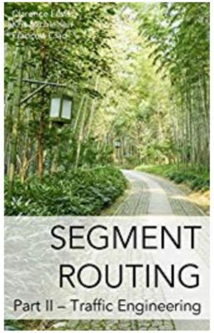


Segment Routing, Part I (Oct 6, 2016)
by Clarence Filsfils, Kris Michielsen, Ketan Talaulikar
(6)
$46.80
Sold by: Amazon Digital Services LLC

Over the past ten years, operators have heavily invested in network infrastructure because of exploding growth in IP traffic. Moreover, the lack of close interaction

Read More

2



Segment Routing Part II: Traffic Engineering (Apr 22, 2019)
by Clarence Filsfils, Kris Michielsen, Francois Clad
$46.80
Sold by: Amazon Digital Services LLC

As more and more Service Providers and Enterprises operate a single network infrastructure to support an ever-increasing number of services, the ability to custom

Read More

Segment Routing (SR) - TI-LFA
SR Mapping Server
Use Case P-PE (SR + LDP/RSVP-TE)
SR Migration
Use-case P (SR) + PE (SR)

SR References

www.segment-routing.net/tutorials/

SR Concepts >

- SR Introduction
- SR MPLS data plane
- SR Global Block (SRGB)

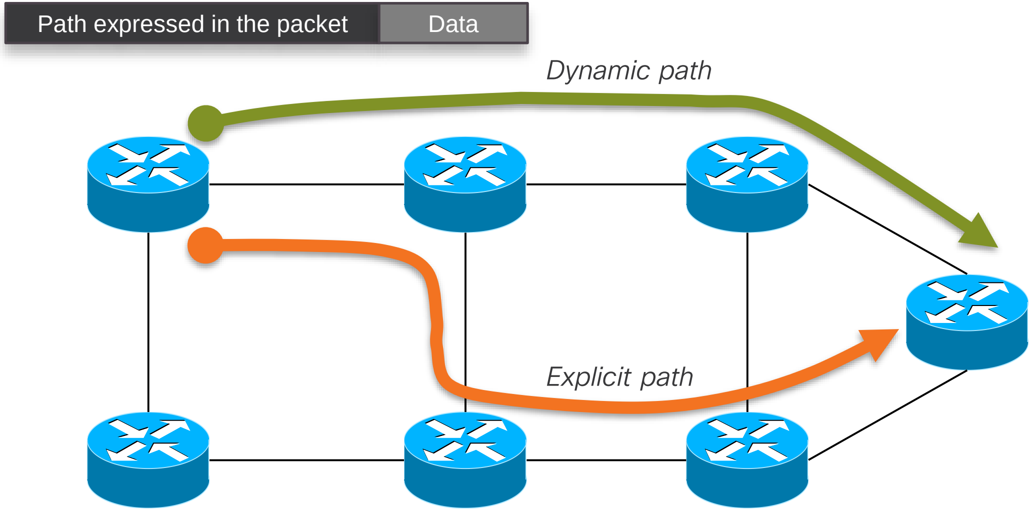
SR IGP >

- SR IGP Control Plane
- Topology Independent LFA (TI-LFA)

SR and LDP >

- SR and LDP co-existence
- SR Mapping Server
- SR and LDP interworking

Segment Routing - Overview



Data Plane	
MPLS (segment labels)	IPv6 (+ SR extension header)

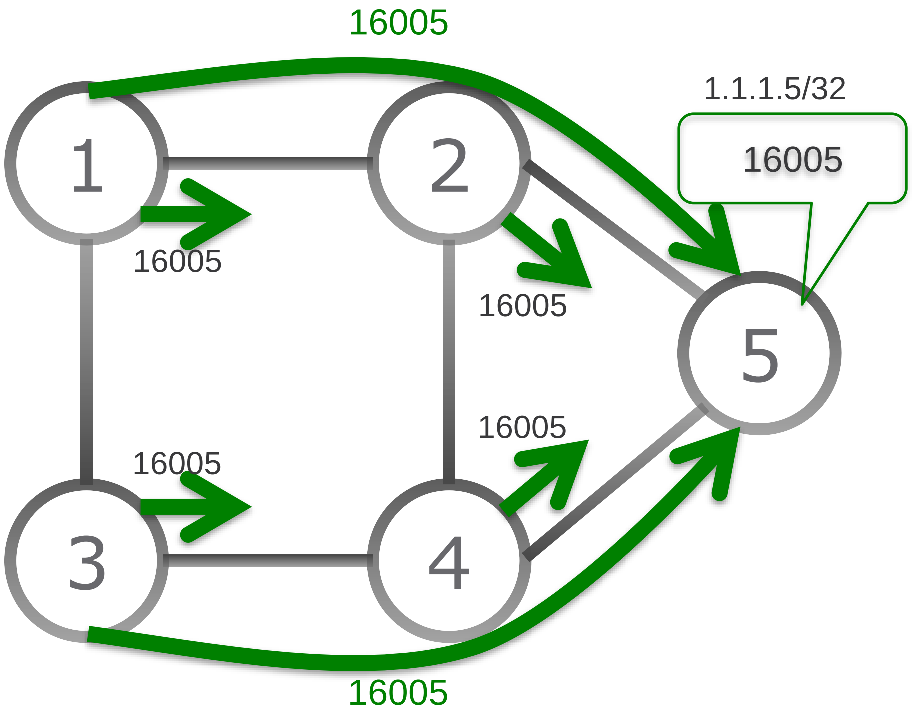
Control Plane	
Routing protocols with extensions (IS-IS,OSPF, BGP)	SDN controller (BGP LS, PCEP, NETCONF/YANG)

Paths options	
Dynamic (STP computation)	Explicit (expressed in the packet)

IGP Prefix Segment

- Signaled by ISIS/OSPF
  - Minor extensions to existing link-state routing protocols
- Shortest-path to IGP prefix
  - Equal Cost MultiPath (ECMP)-aware
- Global significance in SR domain
- Label = SRGB + Index
  - SRGB = Segment Routing Global Block
  - Default SRGB: 16,000 – 23,999
  - Advertised as index

Example 1: Best Path

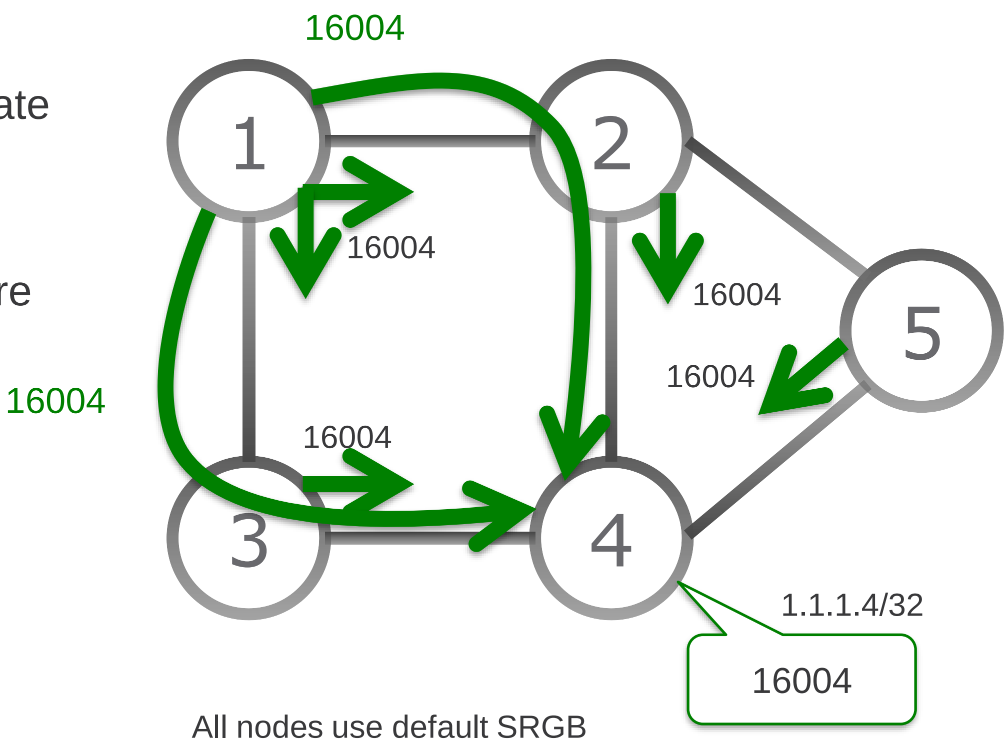


All nodes use default SRGB
16,000 – 23,999

IGP Prefix Segment

- Signaled by ISIS/OSPF
  - Minor extensions to existing link-state routing protocols
- Shortest-path to IGP prefix
  - Equal Cost MultiPath (ECMP)-aware
- Global significance in SR domain
- Label = SRGB + Index
  - SRGB = Segment Routing Global Block
  - Default SRGB: 16,000 – 23,999
  - Advertised as index

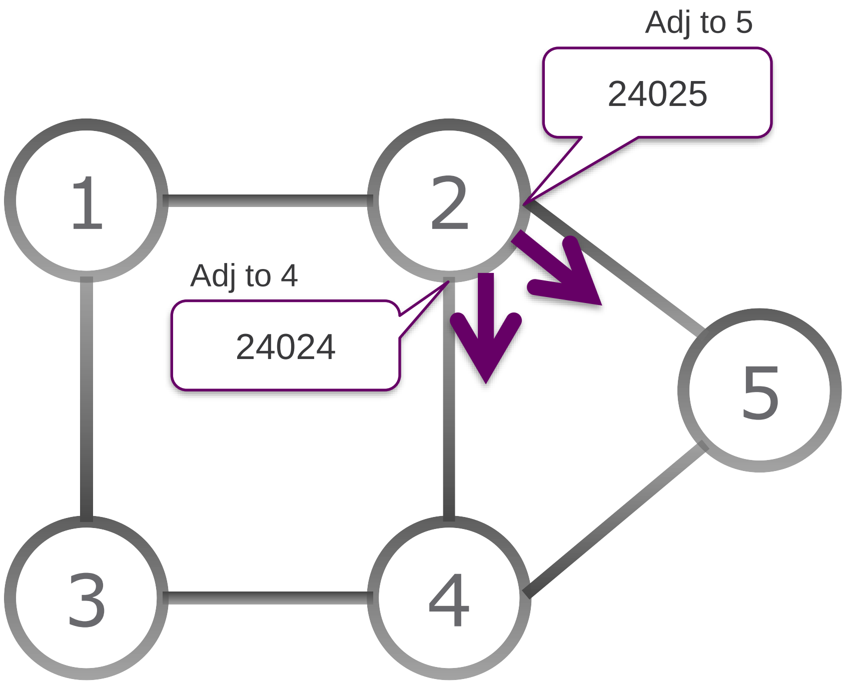
Example 2: ECMP



All nodes use default SRGB 16,000 – 23,999

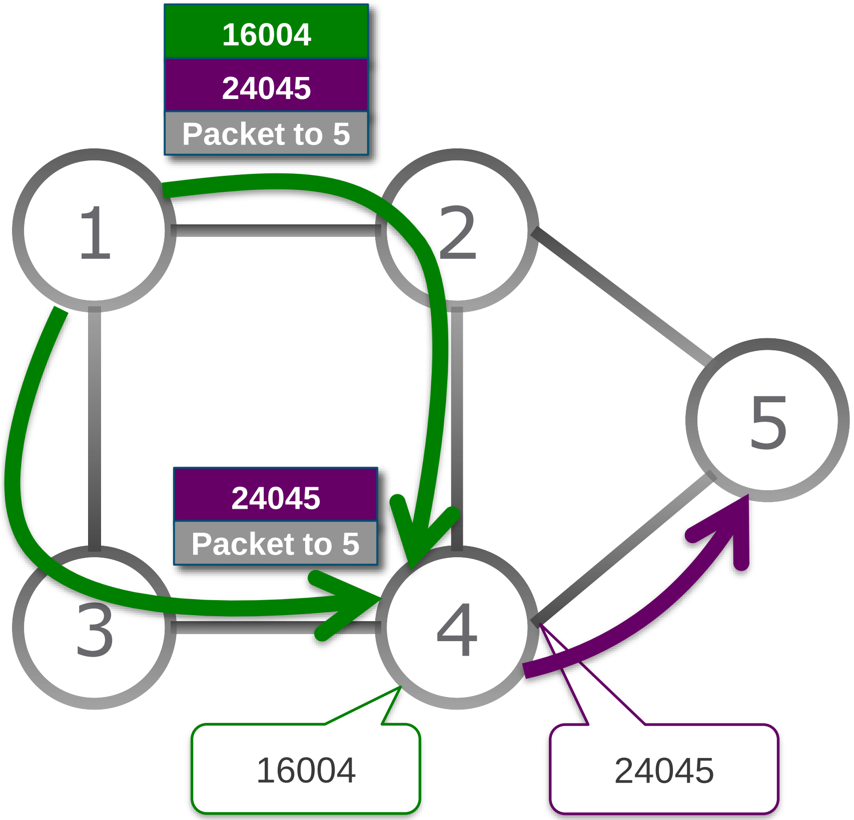
IGP Adjacency Segment

- Signaled by ISIS/OSPF
  - Minor extensions to existing link-state routing protocols
- Forward on IGP adjacency
- Local significance
- Automatically allocated by router

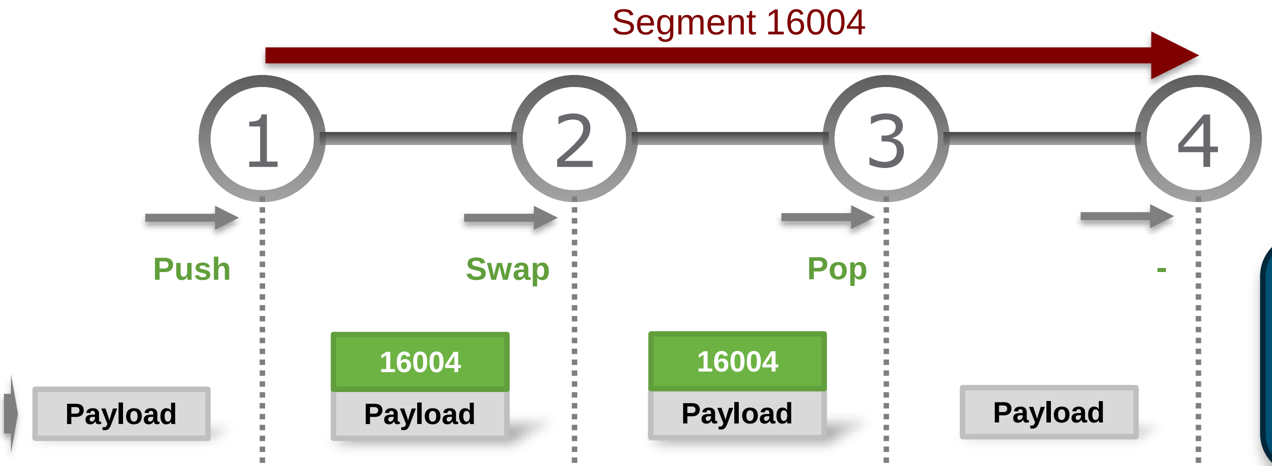


Combining IGP Segments

- Signaled by ISIS/OSPF
- Steer traffic on any path through the network
- Path is specified by list of segments in packet header, a stack of labels
- No path is signaled
- No per-flow state is created



MPLS Data Plane Operations



IPv4: 1.1.1.4/32
or
IPv6: 2001::0101:0104/128
Prefix-SID 16004

Assumptions:
SR enabled on all nodes
LDP not enabled or SR-preferred on Node1

- Node4 advertises its loopback v4 or v6 address with attached prefix-SID 16004
  - IPv4 address: 1.1.1.4/32
  - IPv6 prefix: 2001::0101:0104/128
- Node4 requests default PHP functionality

SR Label Forwarding Verification

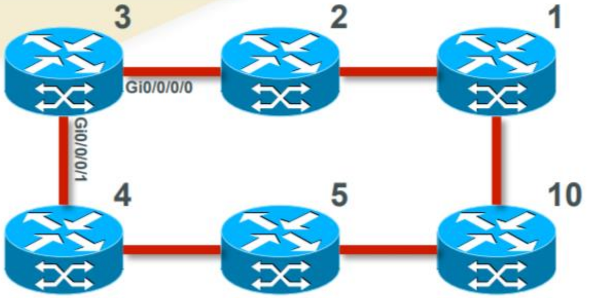
```
RP/0/0/CPU0:xrvr-3#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16001  16001       SR Pfx (idx 1)     Gi0/0/0/0    99.2.3.2        0
16002  Exp-Null-v4 SR Pfx (idx 2)     Gi0/0/0/0    99.2.3.2        0
16004  Pop         SR Pfx (idx 4)     Gi0/0/0/1    99.3.4.4        0
16005  16005       SR Pfx (idx 5)     Gi0/0/0/1    99.3.4.4        0
16010  16010       SR Pfx (idx 10)    Gi0/0/0/0    99.2.3.2        0
       16010       SR Pfx (idx 10)    Gi0/0/0/1    99.3.4.4        0
24032  Pop         SR Adj (idx 1)     Gi0/0/0/0    99.2.3.2        0
24034  Pop         SR Adj (idx 3)     Gi0/0/0/1    99.3.4.4        0
```

Remote prefix-SID

Neighbor prefix-SID Explicit-Null

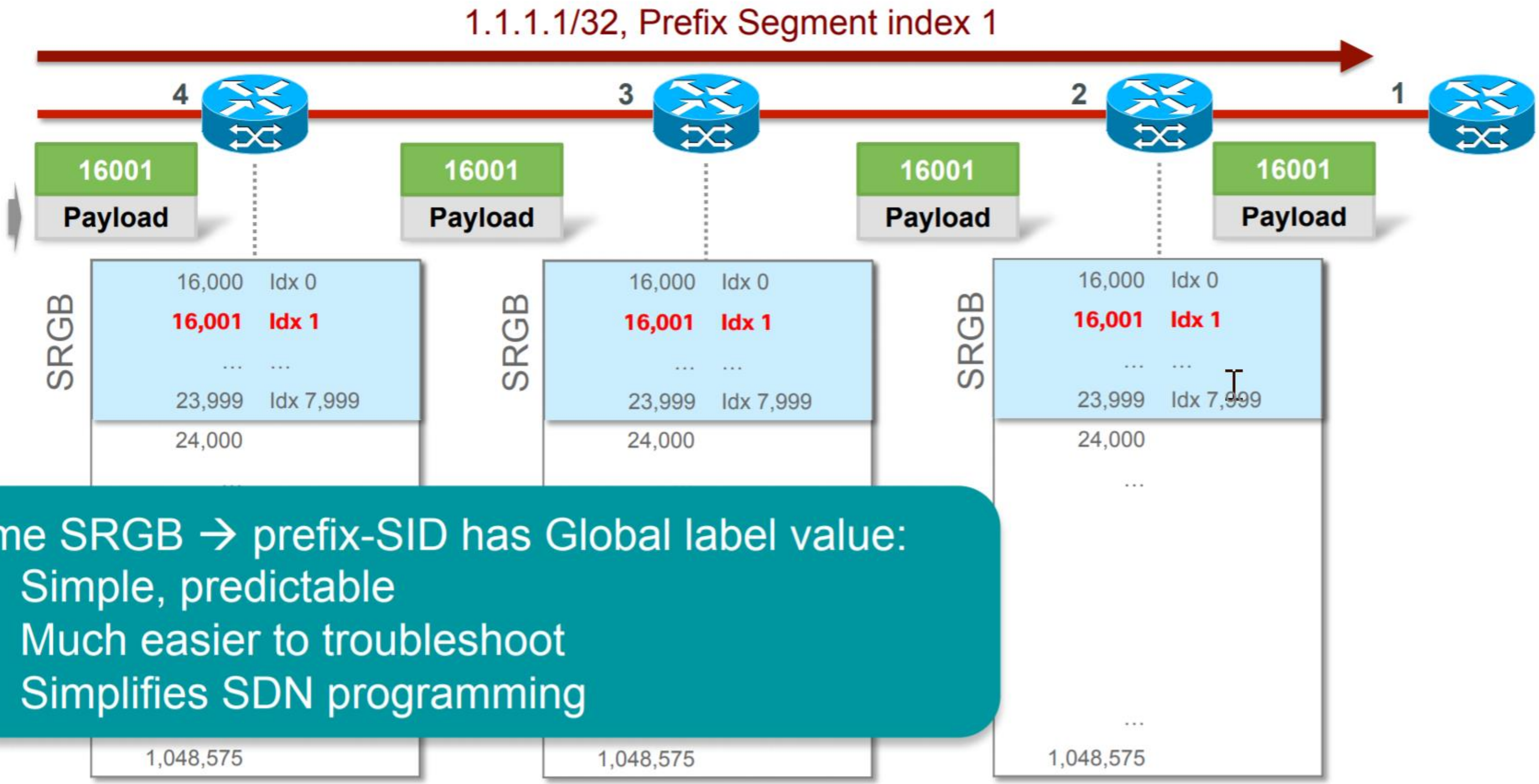
Neighbor prefix-SID PHP on

Remote prefix-SIDs ECMP



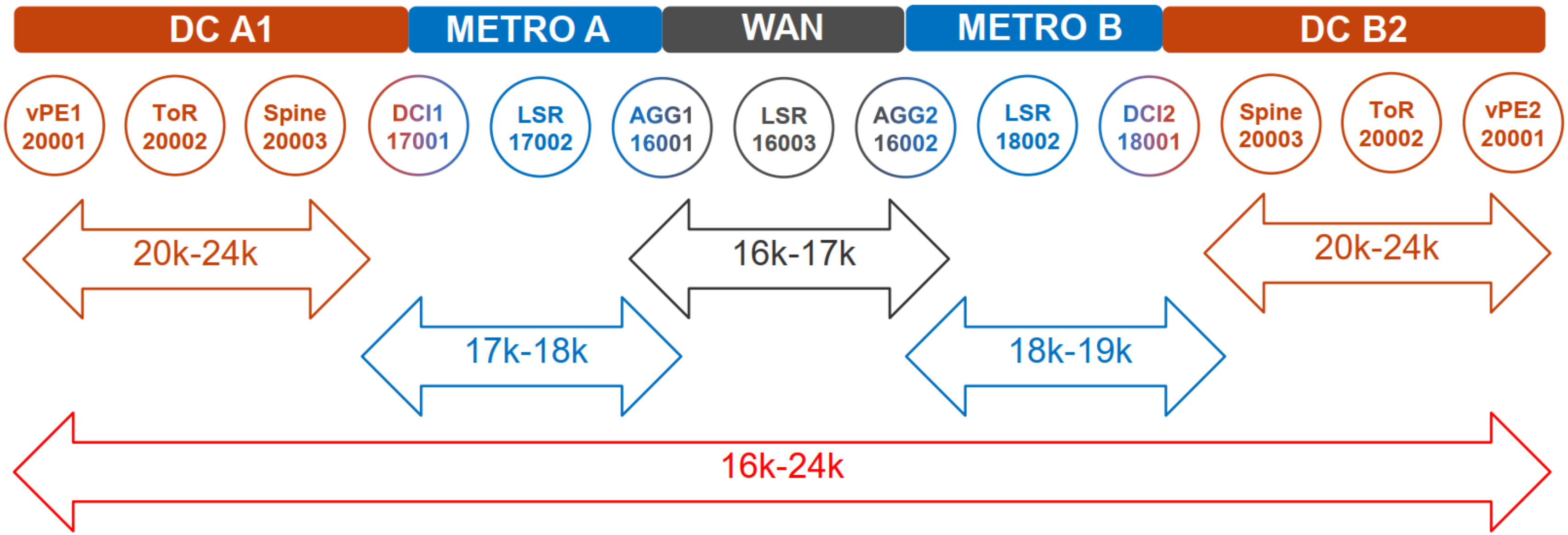
Node *n* advertises prefix-SID 16000+*n*

Recommended SRGB



Same SRGB → prefix-SID has Global label value:
- Simple, predictable
- Much easier to troubleshoot
- Simplifies SDN programming

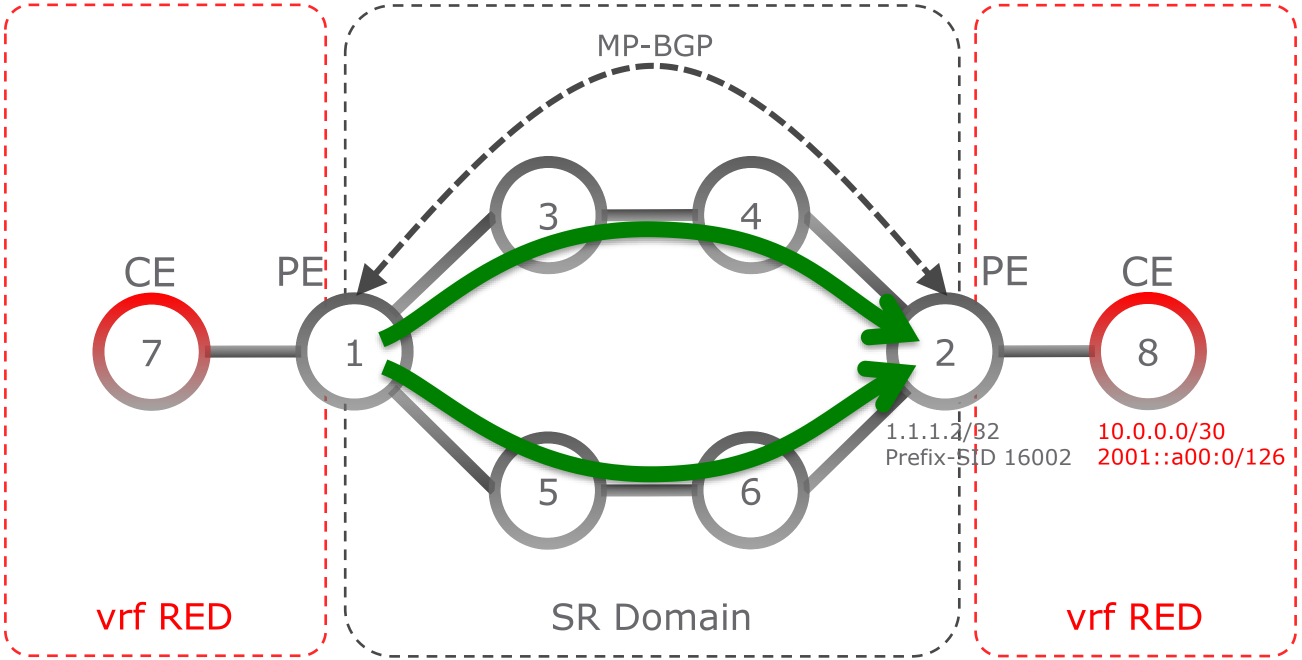
SRGB and SID Allocation



- Homogenous end-to-end SRGB for simplicity
- Globally Unique Prefix SIDs for devices WAN and Metro domains
- Locally Unique Prefix SIDs for Datacenters

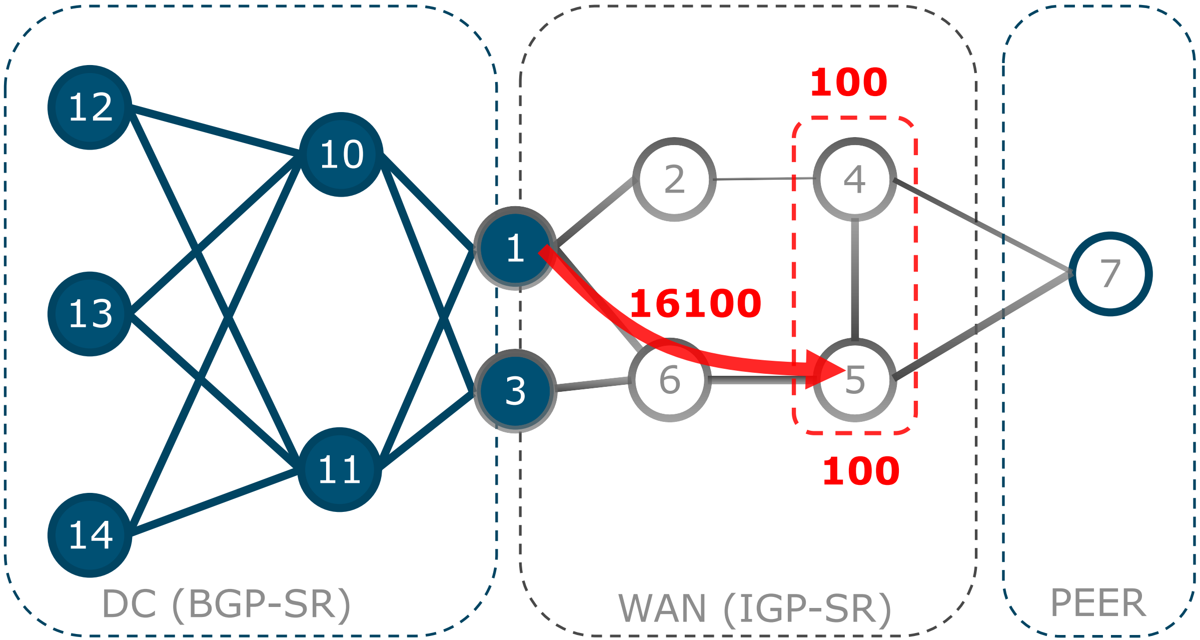
Simple and Efficient Transport of MPLS services

- MPLS services ride on prefix segments
- Simple, one less protocol to operate (LDP)



Anycast Prefix Segment ID (SID)

- Same prefix advertised by multiple nodes
- Traffic forwarded to one of Anycast prefix-SIDs based on best IGP path
- If primary node fails, traffic is auto re-routed to other node



IGP with Segment Routing

```
router isis 1
 address-family ipv4 unicast
  metric-style wide
  segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid prefix-sid index 1
```

```
router ospf 1
 router-id 1.1.1.1
 segment-routing mpls
 segment-routing forwarding mpls
 area 0
  interface Loopback0
   passive enable
   prefix-sid index 1
```

IGP-SR

↓

TI-LFA

↓

SR/LDP Co-existence

LDP->SR migration

IGP with Segment Routing + TI-LFA

```
router isis 1
 address-family ipv4 unicast
  metric-style wide
  segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid prefix-sid index 1
```

```
router ospf 1
 router-id 1.1.1.1
 segment-routing mpls
 segment-routing forwarding mpls
 area 0
  interface Loopback0
   passive enable
   prefix-sid index 1
```

IGP-SR

↓

TI-LFA

```
router isis 1
 interface GigabitEthernet0/0/0/2
  address-family ipv4 unicast
   fast-reroute per-prefix
   fast-reroute per-prefix ti-lfa
```

```
router ospf 1
 fast-reroute per-prefix
 fast-reroute per-prefix ti-lfa
```

↓

SR/LDP Co-existence

LDP->SR migration

IGP with Segment Routing + TI-LFA + LDP->SR Migration

IGP-SR

```
router isis 1
 address-family ipv4 unicast
  metric-style wide
  segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid prefix-sid index 1
```

```
router ospf 1
 router-id 1.1.1.1
 segment-routing mpls
 segment-routing forwarding mpls
 area 0
  interface Loopback0
   passive enable
   prefix-sid index 1
```

TI-LFA

```
router isis 1
 interface GigabitEthernet0/0/0/2
  address-family ipv4 unicast
   fast-reroute per-prefix
   fast-reroute per-prefix ti-lfa
```

```
router ospf 1
 fast-reroute per-prefix
 fast-reroute per-prefix ti-lfa
```

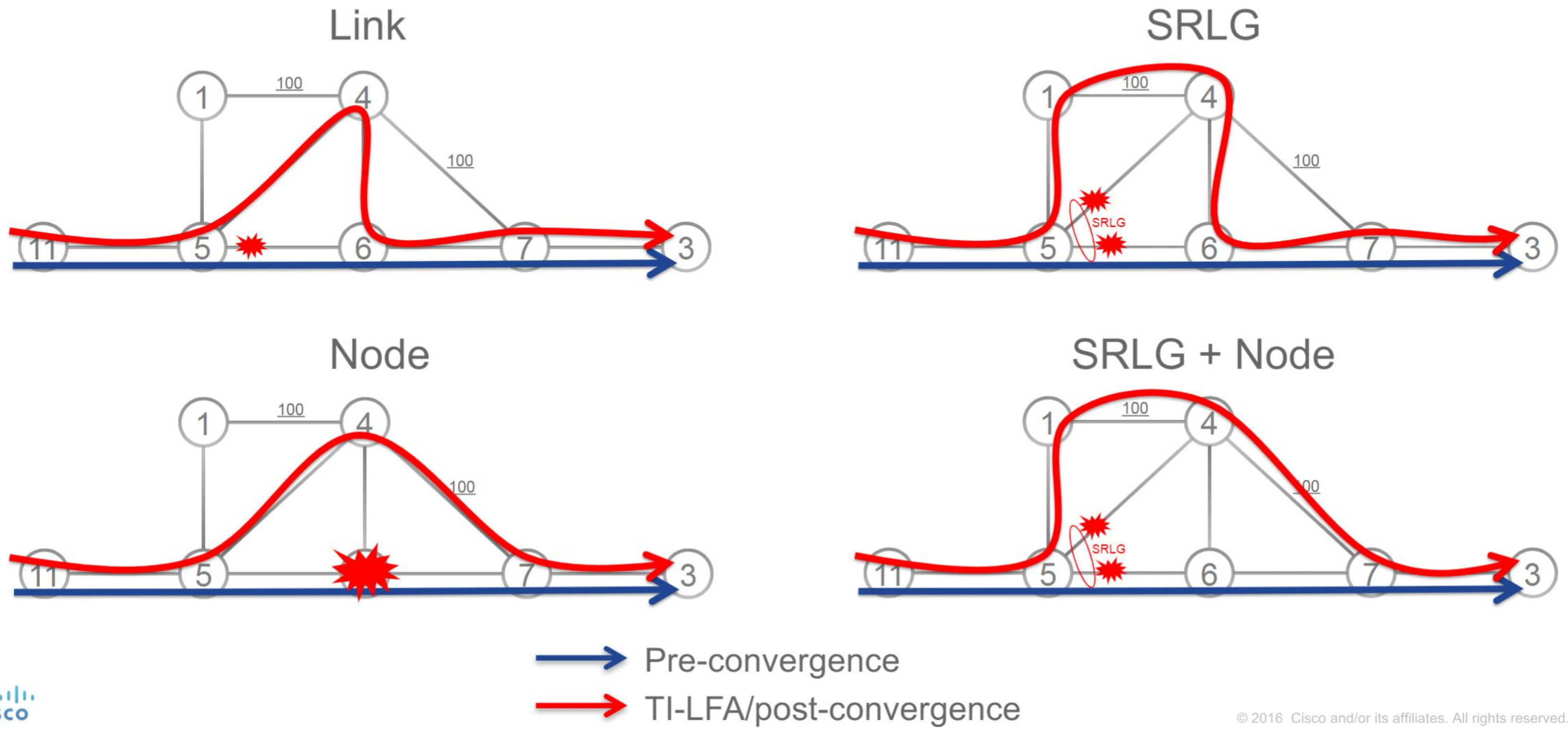
SR/LDP Co-existence

LDP->SR migration

```
router isis 1
 address-family ipv4 unicast
  segment-routing mpls sr-prefer
```

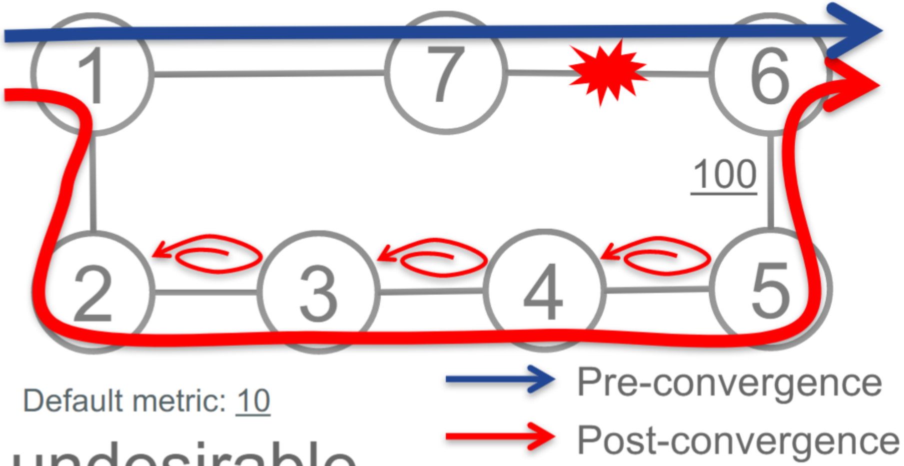
```
router ospf 1
 segment-routing mpls
 segment-routing sr-prefer
```

TI-LFA Protecting Link, Node, SRLG



What is MicroLoop ?

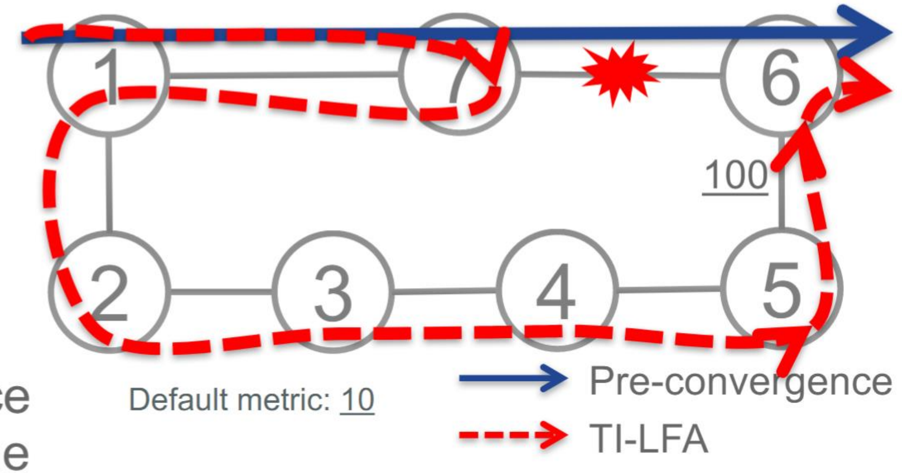
- Microloops are a day-one IP drawback
- IP hop-by-hop routing may induce microloop at any topology transition
  - Link up/down, metric up/down
- E.g. Microloops can occur after failure of link 6-7



- Microloops can increase packet loss, which is especially undesirable when FRR is used.

SR MicroLoop Avoidance workflow

- TI-LFA protection kicks in on Node7, repairing the traffic to Node6 via Node5 and link Node5-Node6
- All nodes are notified of the topology change due to the failure
- E.g. Node1 computes the post-convergence SPT and detects possible microloops on the post-convergence paths for any destination, such as Node6

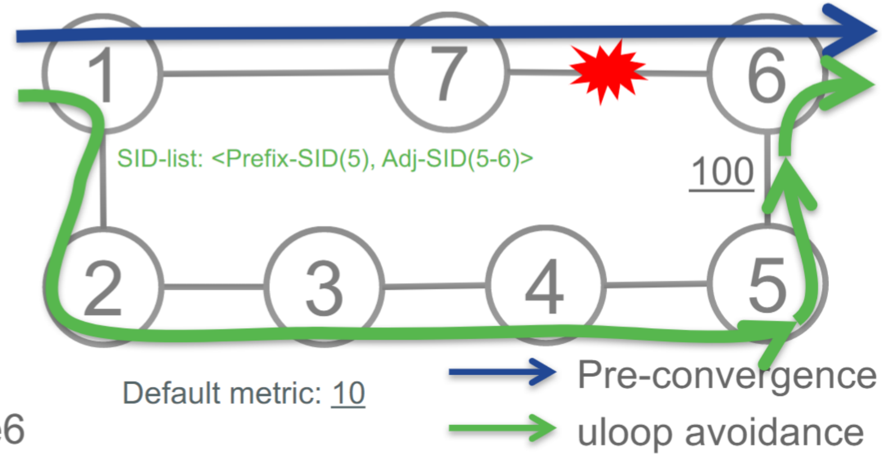


- If microloops are possible on the post-convergence path for a destination, then a SID-list is constructed to steer the traffic to that destination loop-free over the post-convergence path; in this example: <Prefix-SID(5), Adj-SID(5-6)> for destination Node6

SR Microloop Avoidance workflow

-

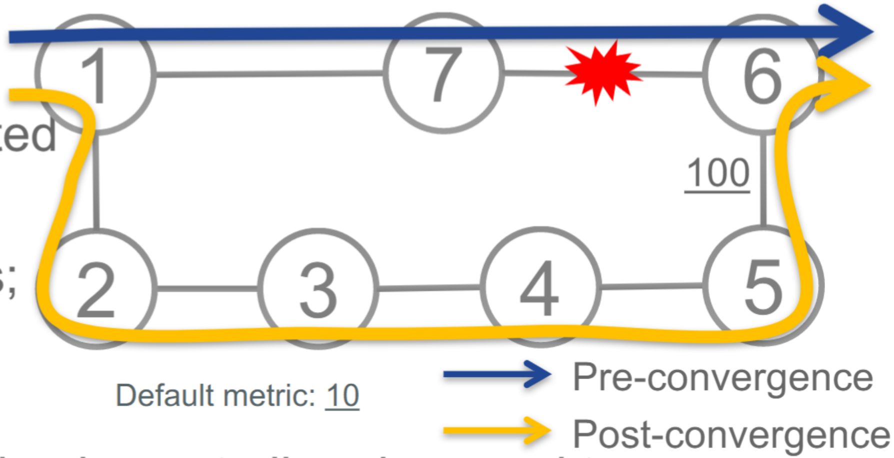
- IGP on Node1 updates the forwarding table and installs the SID-list imposition entries for those destinations with possible microloops, such as destination Node6
  - Node1 imposes SID-list <Prefix-SID(5), Adj-SID(5-6)> on packets to Node6



- All nodes converge and update their forwarding tables, using SID-lists where needed

SR Microloop Avoidance workflow

- Some time later, the new topology is applied and no more microloops are expected
- IGP updates the forwarding table, removing the microloop avoidance SID-lists; traffic now natively follows the post-convergence path



- Note: SR microloop avoidance is a local behavior, not all nodes need to implement it to get the benefits
- There is incremental benefit for each node that has it implemented
  - E.g. if only Node1 has SR microloop avoidance, then e.g. traffic entering Node2 (not from Node1) to Node6 would still see microloops
  - When enabling SR microloop avoidance on Node2, then e.g. traffic entering Node3 (not from Node2) to Node6 would still see microloops, etc.

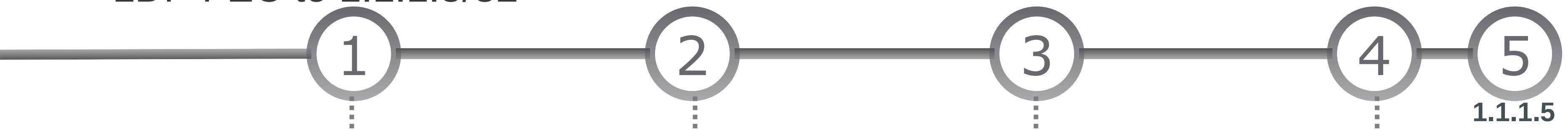
MPLS-to-MPLS and MPLS-to-IP

All nodes: SR + LDP

SR LDP

- LDP FEC to 1.1.1.5/32

Prefix-SID
index 5



SRGB

Node 1

local/in lbl	out lbl
16000	
...	
16005	**16005**
…	
23999	
24000	
…	
24002	**24001**
…	
…	
1048575	

Node 2

local/in lbl	out lbl
16000	
...	
16005	**16005**
…	
23999	
24000	
24001	**32011**
…	
…	
1048575	

Node 3

local/in lbl	out lbl
16000	
...	
16005	**16005**
…	
23999	
24000	
32011	**24003**
…	
…	
1048575	

Node 4

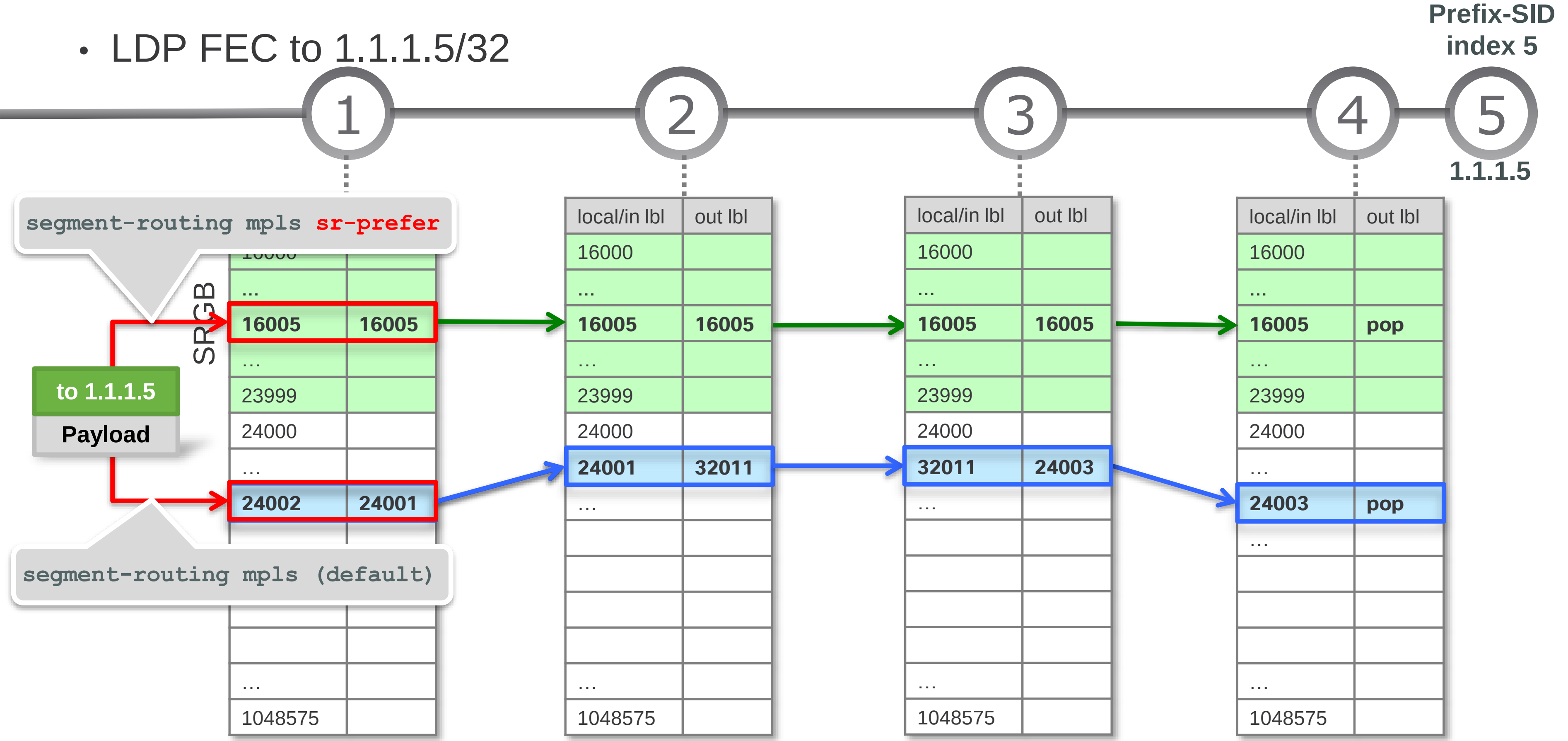
local/in lbl	out lbl
16000	
...	
16005	**pop**
…	
23999	
24000	
…	
24003	**pop**
…	
…	
1048575	

MPLS-to-MPLS and MPLS-to-IP

All nodes: SR + LDP

SR LDP

- LDP FEC to 1.1.1.5/32



Node 1:

local/in lbl	out lbl
16000	
...	
16005	16005
…	
23999	
24000	
…	
24002	24001
…	
…	
1048575	

Node 2:

local/in lbl	out lbl
16000	
...	
16005	16005
…	
23999	
24000	
24001	32011
…	
…	
1048575	

Node 3:

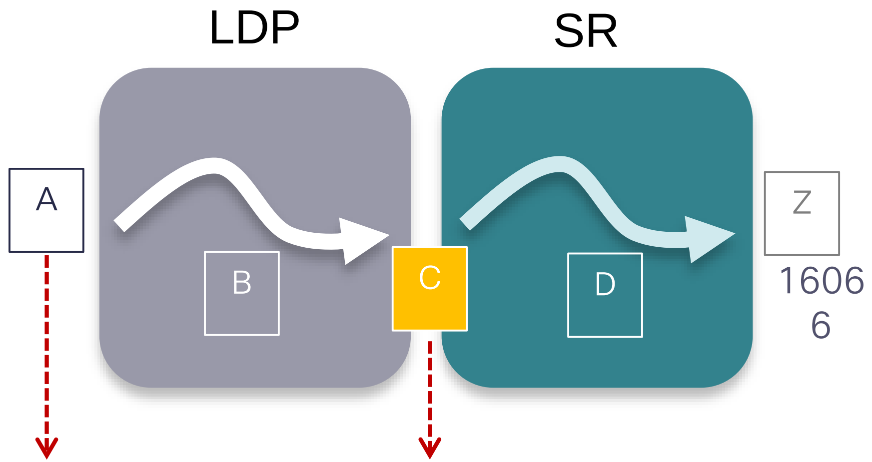
local/in lbl	out lbl
16000	
...	
16005	16005
…	
23999	
24000	
32011	24003
…	
…	
1048575	

Node 4:

local/in lbl	out lbl
16000	
...	
16005	pop
…	
23999	
24000	
…	
24003	pop
…	
…	
1048575	

LDP to SR

- When a node is LDP capable but its next-hop along the SPT to the destination is not LDP capable
  - no LDP outgoing label
- In this case, the LDP LSP is connected to the prefix segment
- C installs the following LDP-to-SR FIB entry:
  - incoming label: label bound by LDP for FEC Z
  - outgoing label: prefix segment bound to Z
  - outgoing interface: D
- This entry is derived automatically at the routing layer

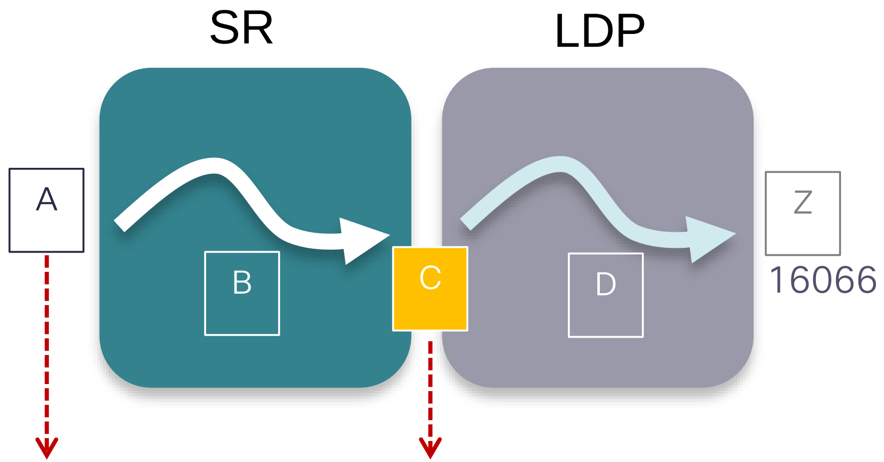


Prefix	Out Label (LDP), Interface
Z	16, 0

Input Label (LDP)	Out Label (SID), Interface
32	16066, 1

SR to LDP

- When a node is SR capable but its next-hop along the SPT to the destination is not SR capable
  - no SR outgoing label available
- In this case, the prefix segment is connected to the LDP LSP
  - Any node on the SR/LDP border installs SR-to-LDP FIB entry(ies)



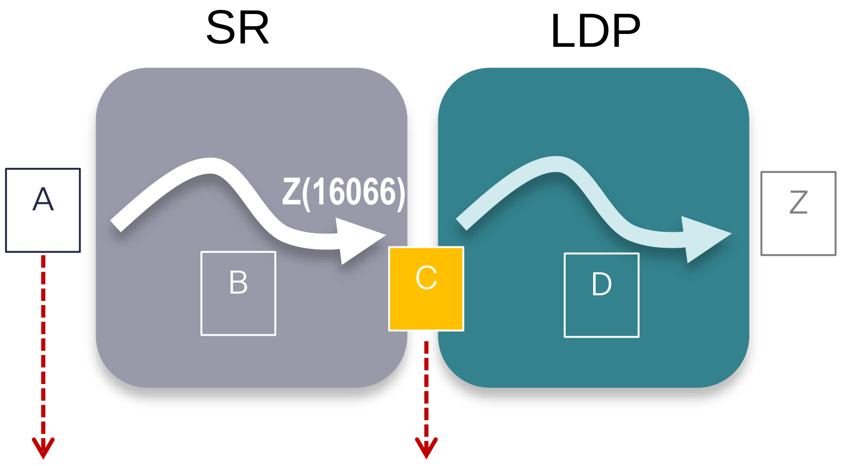
Prefix	Out Label (SID), Interface
Z	?, 0

Input Label (SID)	Out Label (LDP), Interface
?	16, 1

Mapping Server

- A wants to send traffic to Z, but
  - Z is not SR-capable, Z does not advertise any prefix-SID

  → which label does A have to use?
- The Mapping Server advertises the SID mappings for the non-SR routers
  - for example, it advertises that Z is 16068
- A and B install a normal SR prefix segment for 16066
- C realizes that its next hop along the SPT to Z is not SR capable hence C installs an SR-to-LDP FIB entry
  - incoming label: prefix-SID bound to Z (16066)
  - outgoing label: LDP binding from D for FEC Z
- A sends a frame to Z with a single label: 16066



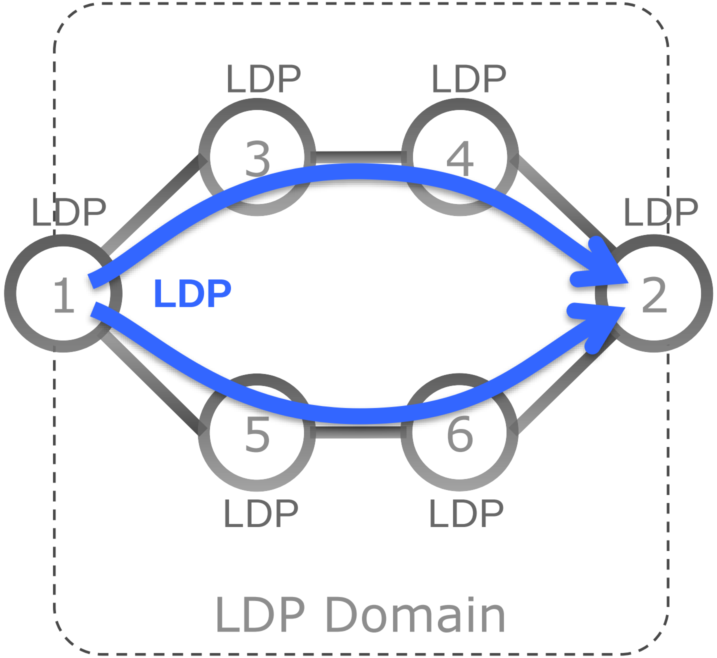
Prefix	Out Label (SID), Interface
Z	?, 0

Input Label (SID)	Out Label (LDP), Interface
?	16, 1

Simplest migration LDP to SR

Assumptions:
- all the nodes can be upgraded to SR
- all the services can be upgraded to SR

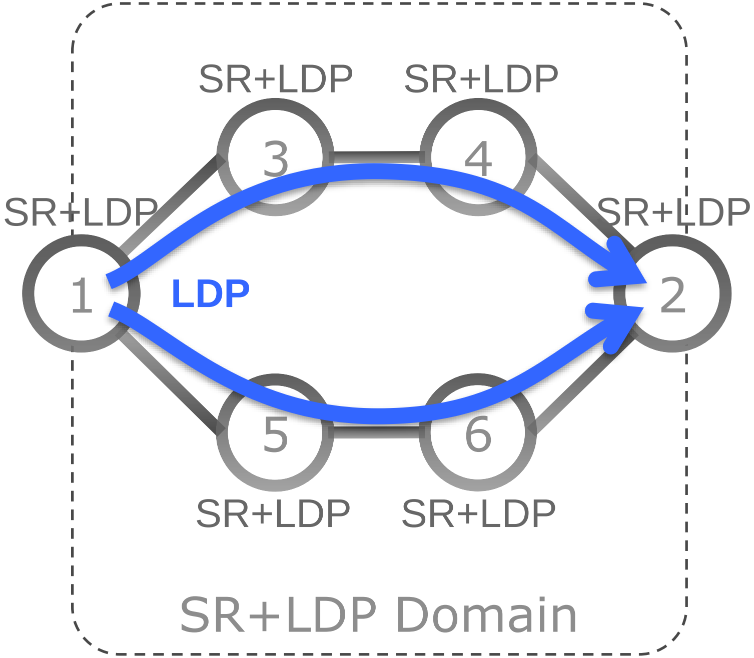
- Initial state: All nodes run LDP, not SR



Simplest migration LDP to SR

Assumptions:
- all the nodes can be upgraded to SR
- all the services can be upgraded to SR

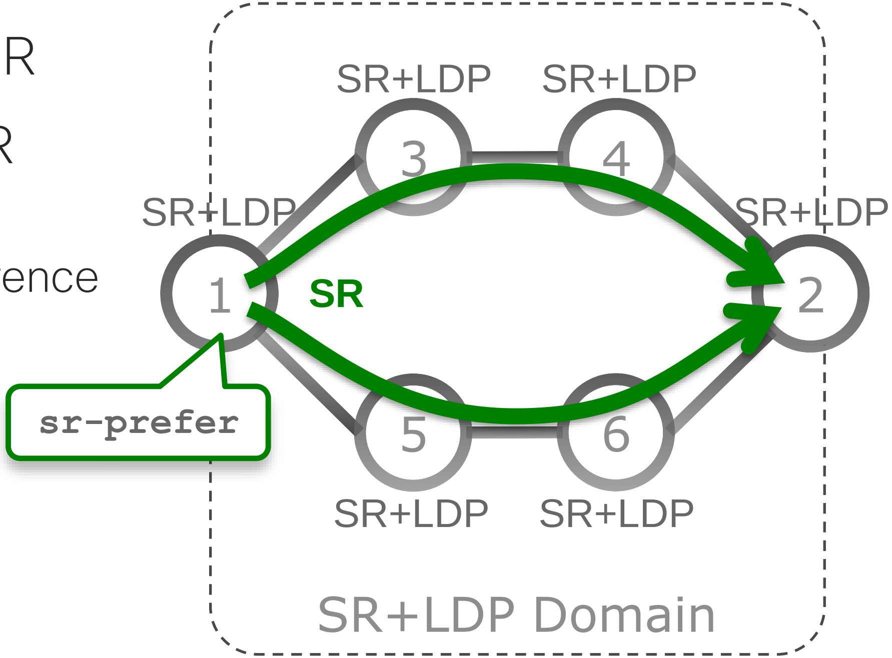
- Initial state: All nodes run LDP, not SR
- Step1: All nodes are upgraded to SR
  - In no particular order
  - leave default LDP label imposition preference



Simplest migration LDP to SR

Assumptions:
- all the nodes can be upgraded to SR
- all the services can be upgraded to SR

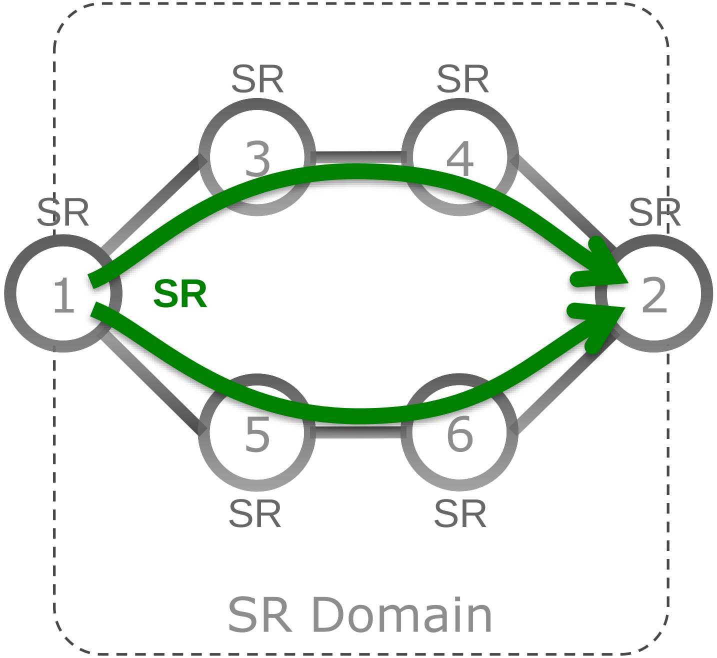
- Initial state: All nodes run LDP, not SR
- Step1: All nodes are upgraded to SR
  - In no particular order
  - leave default LDP label imposition preference
- Step2: All PEs are configured to prefer SR label imposition
  - In no particular order



Assumptions:
- all the nodes can be upgraded to SR
- all the services can be upgraded to SR

Simplest migration LDP to SR

- Initial state: All nodes run LDP, not SR
- Step1: All nodes are upgraded to SR
  - In no particular order
  - leave default LDP label imposition preference
- Step2: All PEs are configured to prefer SR label imposition
  - In no particular order
- Step3: LDP is removed from the nodes in the network
  - In no particular order
- Final state: All nodes run SR, not LDP



SR Policy Concept
VPN SR TE based on
Low Latency/Constraint/WECMP

SR References

www.segment-routing.net/tutorials/

SR Traffic Engineering >
SR TE

3-tiered latency service

- Minimum Latency

  ⇔ minimize the delay to the endpoint

- Minimum Cost

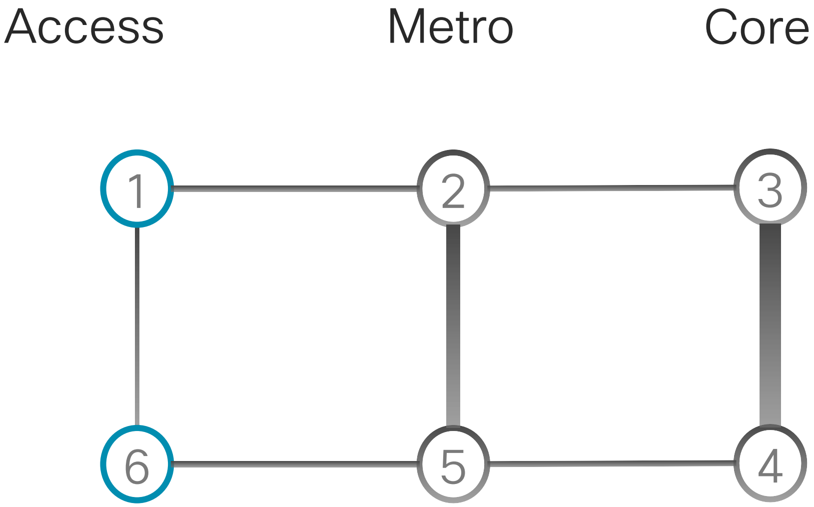
  ⇔ minimize the isis metric to the endpoint

- Minimum Cost but with a bound on the delay
  - SRTE Policy
  - SR Native Algorithm
  - Minimize IGP metric
  - Cumulative delay along the path <= delay bound

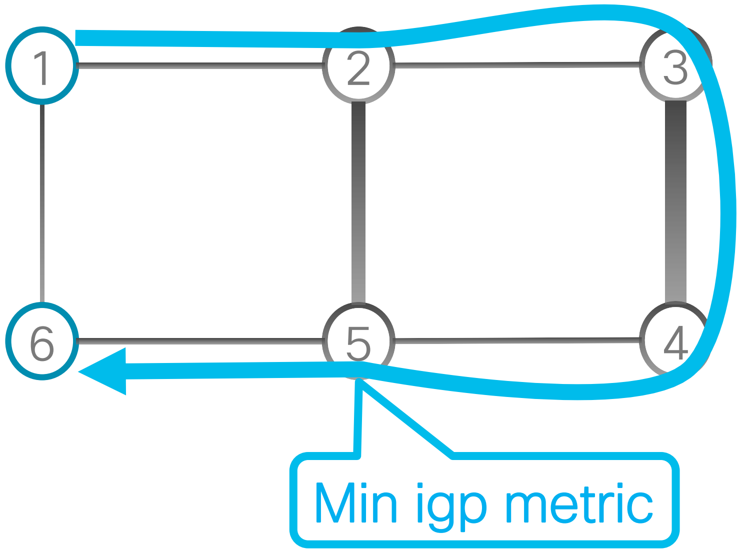
Another obvious business service that was never realized before SR

Business Relevance: Cost vs Latency

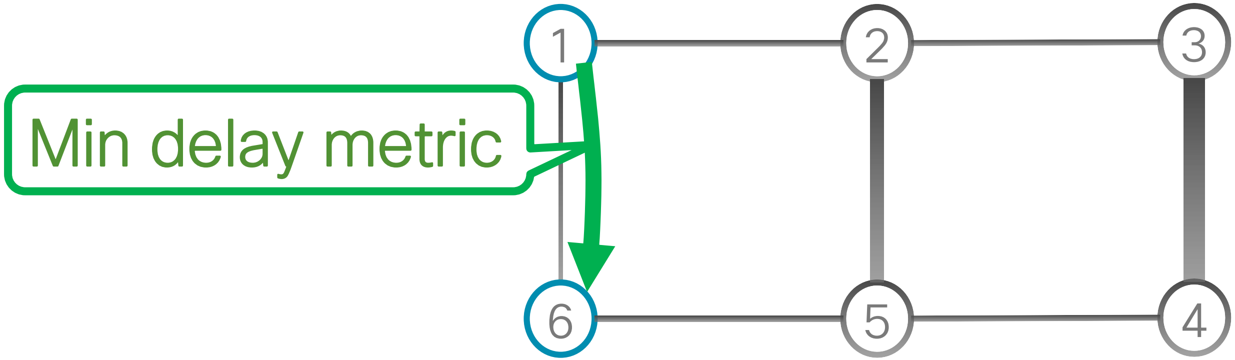
- Core
  - Cheaper
  - Longer delay
- Access Shortcut
  - Most expensive
  - Lowest latency
- Metro Shortcut
  - In-between



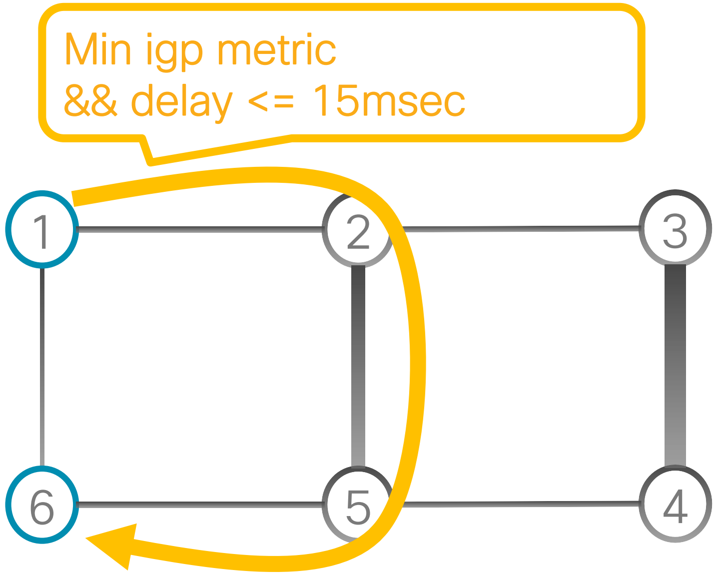
Bulk of the traffic - Best-Effort



Ultra Low Latency – 5G

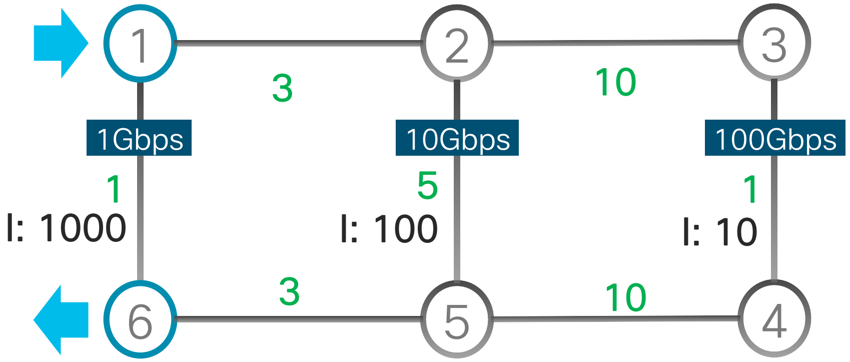


Business traffic with constraint



Use-case: min IGP metric with bound on delay

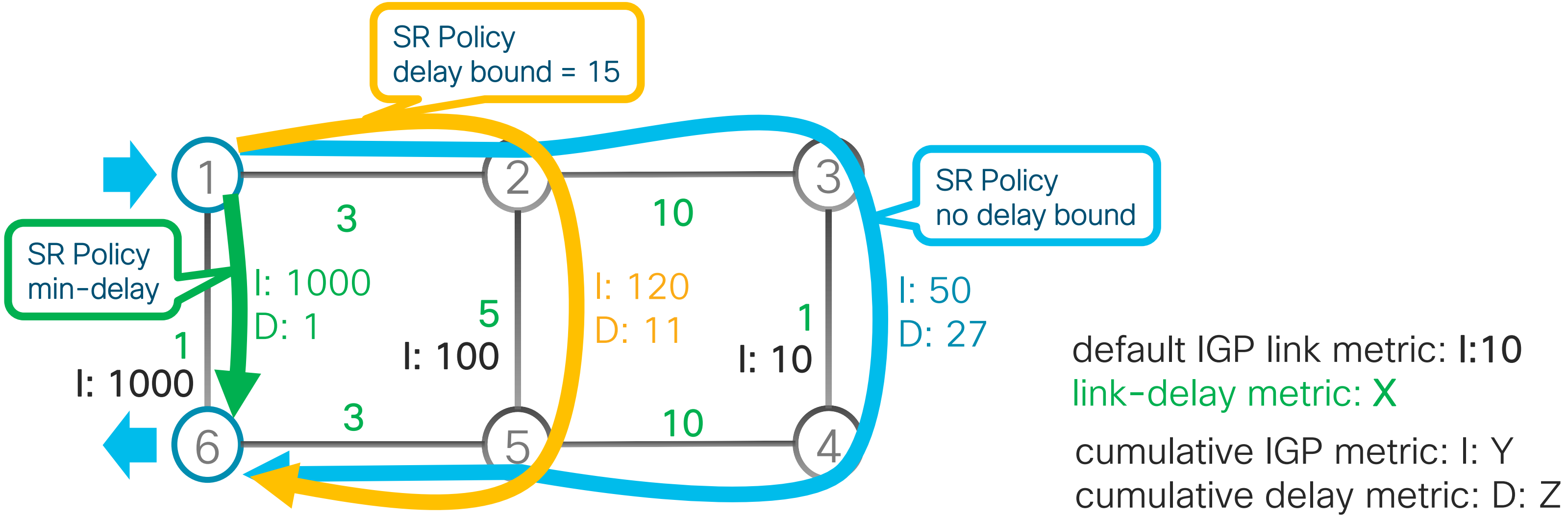
- SRTE returns the shortest IGP path that has a cumulative delay < bound
- Goal: improve network utilization by moving traffic closer to the core while maintaining a delay SLA



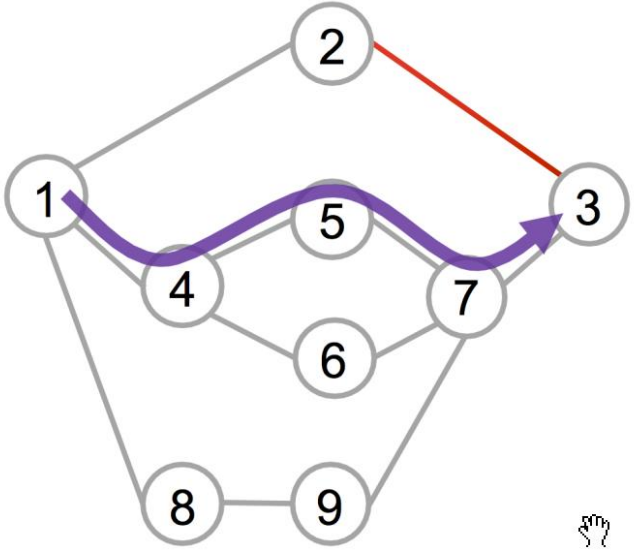
default IGP link metric: I:10
link-delay metric: X

Use-case: min IGP metric with bound on delay

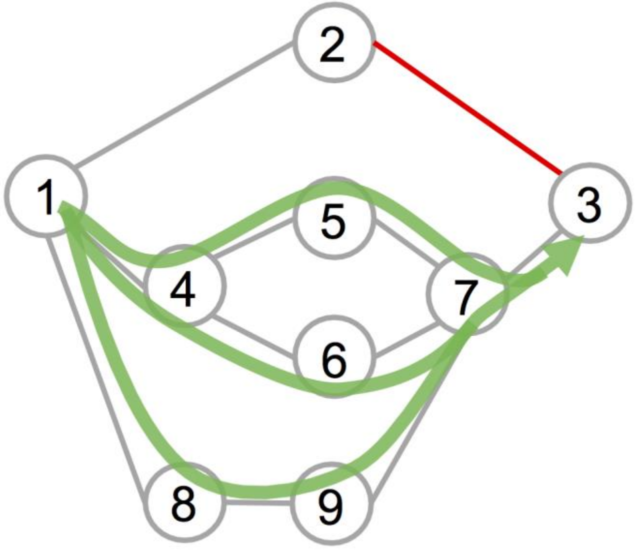
- SRTE returns the shortest IGP path that has a cumulative delay < bound
- Goal: improve network utilization by moving traffic closer to the core while maintaining a delay SLA



RSVP-TE Optimization vs SR Optimization



Pre-SRTE is circuit-based
CSPF => non-ECMP path
Re-using this for SRTE is not good
SID List: <24014, 24045, 24057, 24073>
Poor ECMP, big SID list, ATM-optimized

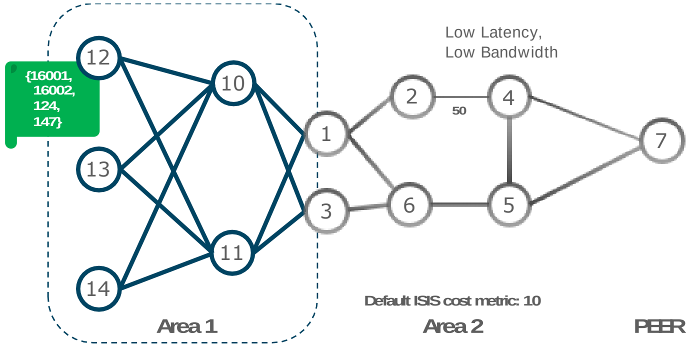


SR-native TE is needed
Recognized Innovation - SIGCOMM 2015

SID List: <16007, 16003>
ECMP, Small SID list, IP-optimized

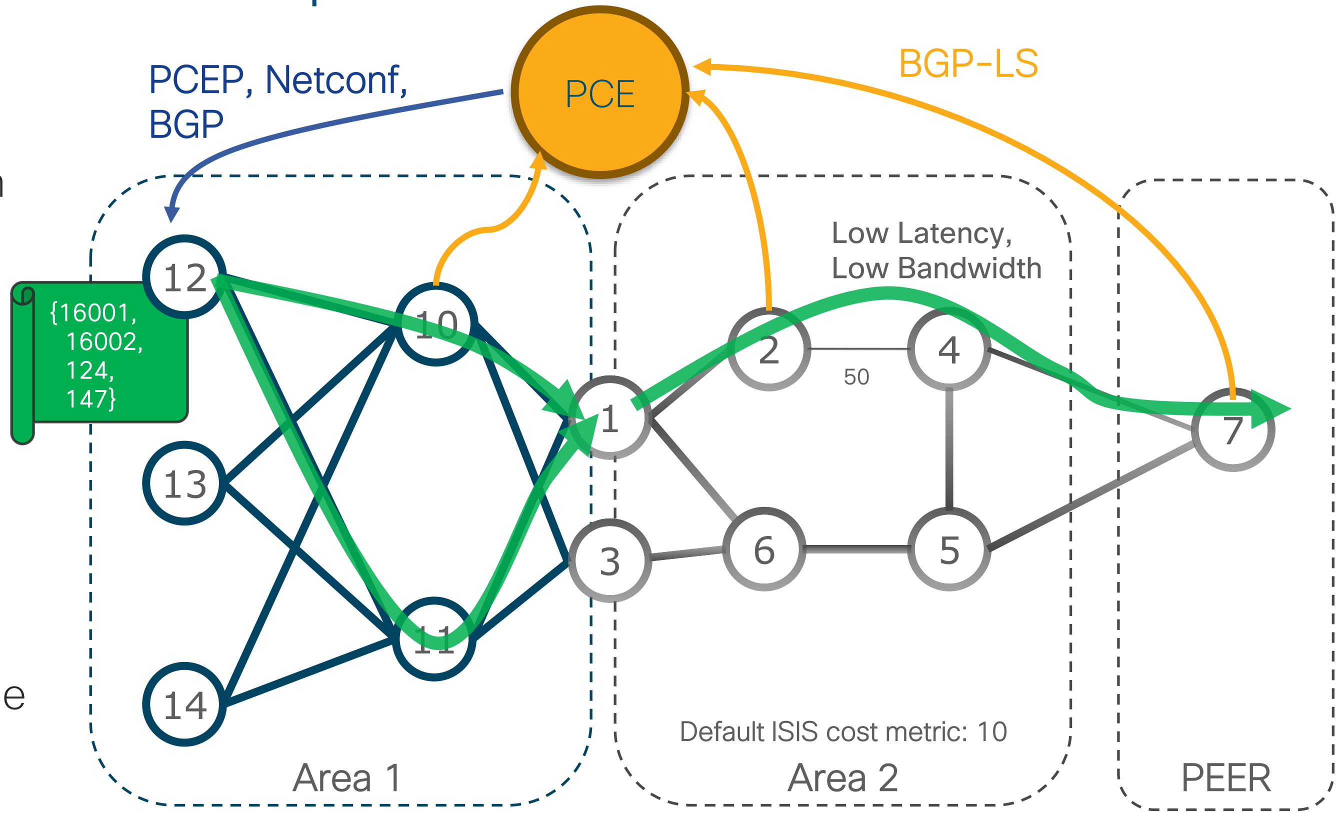
Distributed vs Centralised Path Computation

- There are two possibilities to compute the dynamic path for an SR Policy:
  - Head-end computes the path itself ("distributed")
  - Head-end requests SR PCE to compute the path ("centralised")
- By default, dynamic paths are computed by the head-end
- Head-end uses an external PCE when local computation is not possible
  - An external PCE is required if more information is needed than is available on a head-end; e.g. multi-area/domain paths, or disjoint paths from different head-ends



Centralized Path Computation Element

- The head-end is configured to use the PCE for dynamic path computation
- The PCE computes that the green path can be encoded as
  - 16001
  - 16002
  - 124
  - 147
- The PCE programs the path to the head-end

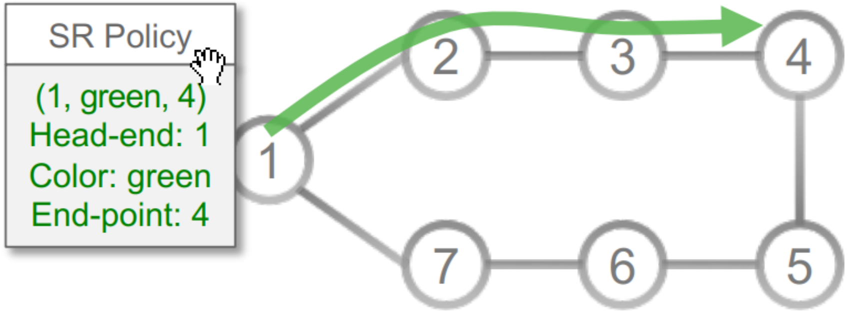


SRTE

- Simple, Automated and Scalable
  - No core state: state in the packet header
  - No tunnel interface: “SR Policy”
  - No head-end a-priori configuration: on-demand policy instantiation
  - No head-end a-priori steering: automated steering
- Multi-Domain
  - SR PCE for compute
  - Binding-SID (BSID) for scale
- Lots of Functionality
  - Designed with lead operators along their use-cases

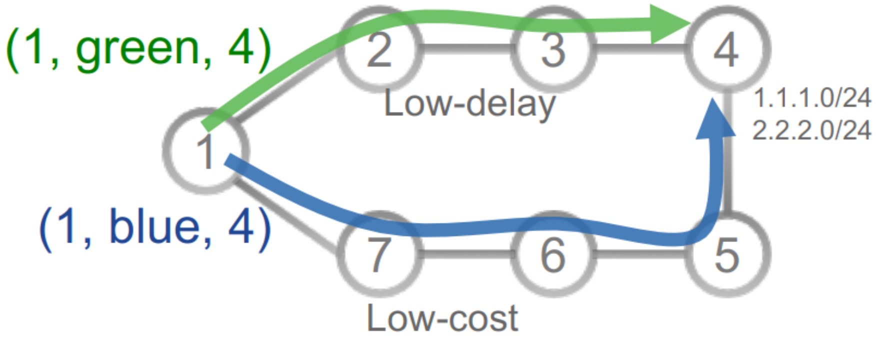
SR Policy Identification

- 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)



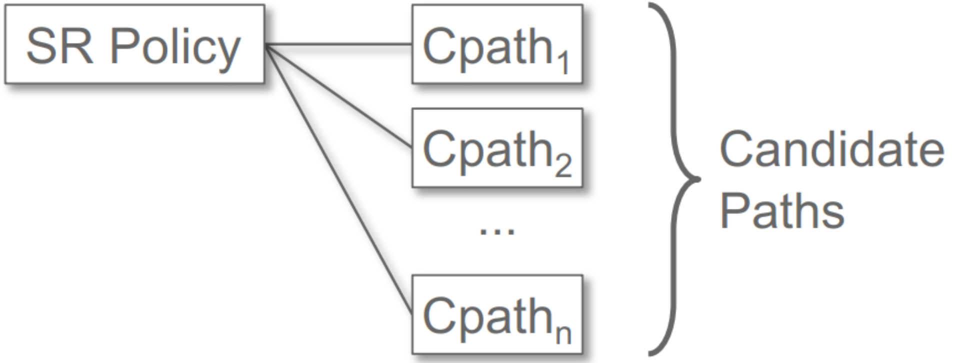
SR Policy Color

- 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

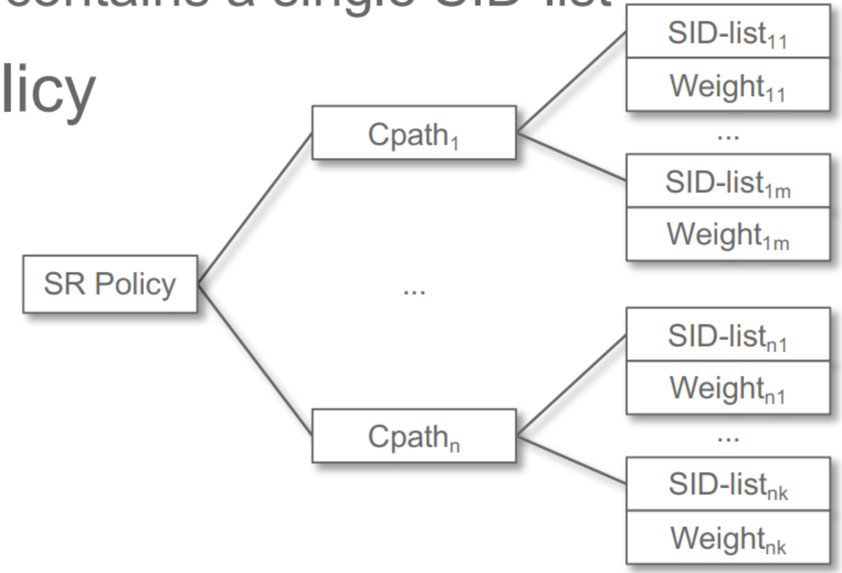
- An SR Policy consists of one or more candidate paths (Cpaths)



- 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

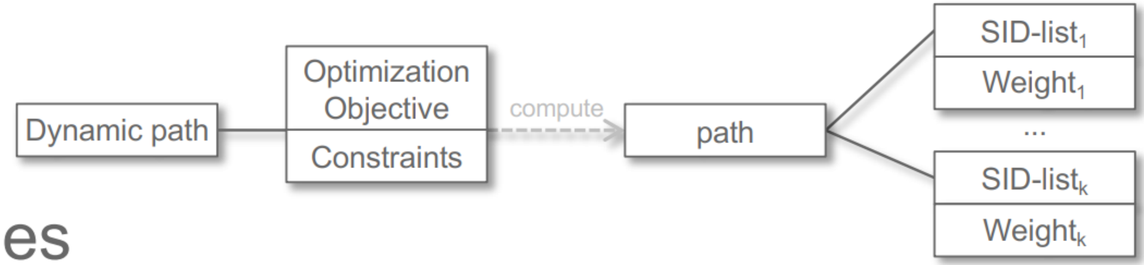
SR Policy – Candidate Path

- 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

Dynamic Path



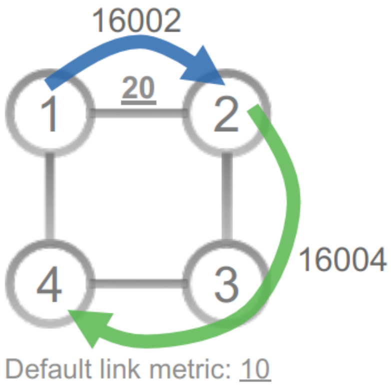
- A dynamic path expresses an optimization objective and a set of constraints
- The head-end computes a solution to the optimization problem as a SID-list or a set of SID-lists
- When the head-end does not have enough topological information (e.g. multi-domain problem), the head-end may delegate the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

Topological path to SID-list

- After the path is computed, the SID-list that expresses the desired path is derived
- High-level algorithm:
  1. Node = head-end
  2. Find an IGP prefix-SID that leads as far down the desired path as possible (without using any link not included in the desired path)
  3. If no such prefix-SID exists, use the Adj-SID to the first neighbor along the path
  4. Node = the farthest node that is reached; goto 2.

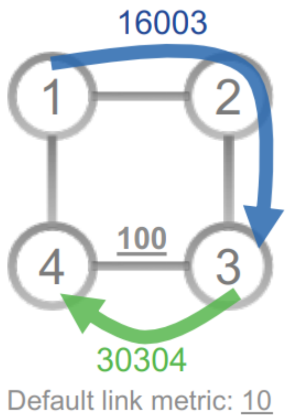
Topological path to SID-list – Example 1

- Desired topological path = 1→2→3→4
- SID-list = <16002, 16004>
  - 16002 brings the packet from 1 to 2 (shortest path from Node1 to Node2)
  - 16004 brings the packet from 2 to 4 via 3 (shortest path from Node2 to Node4)



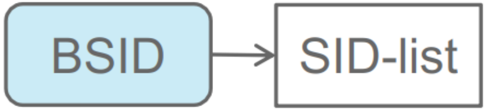
Topological path to SID-list – Example 2

- Desired topological path = 1→2→3→4
- SID-list = <16003, 30304>
  - 16003 brings the packet from 1 to 3 (shortest path from Node1 to Node3)
  - 30304 brings the packet from 3 to 4 using the Adjacency-SID

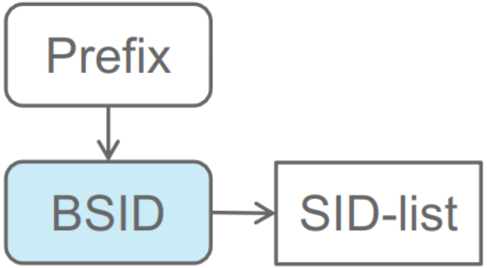


Binding-SID (BSID) is fundamental

- The BSID of the SR Policy selected path is installed in the forwarding table
- Remote steering
  - A packet arriving on the SR Policy head-end with the BSID as Active Segment (top of label stack) is steered into the SR Policy associated with the BSID



- Local steering
  - A packet that matches a forwarding entry that resolves on the BSID of an SR Policy is steered into that SR Policy



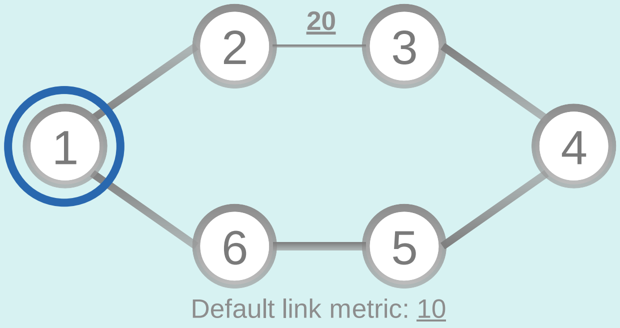
SR Policy – Configuration Example

On Node1:

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
    preference 100
     dynamic
      metric type te
     constraints
      affinity
       exclude-any color red
    !
    preference 200
     explicit segment-list SIDLIST1
  !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

SRTE

SR Policy



SR Policy – Configuration Example

On Node1:

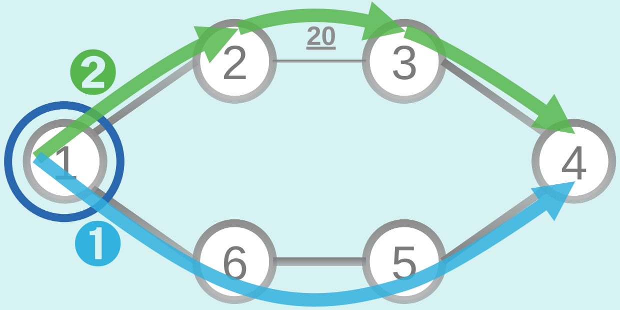
```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
1   preference 100
     dynamic
      metric type te
     constraints
      affinity
       exclude-any color red
    !
2   preference 200
     explicit segment-list SIDLIST1
  !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

User-defined name

Color and End-point

Binding-SID

Local Candidate Paths



SR Policy – configuration example

On Node1:

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
1   preference 100
     dynamic
      metric type te
     constraints
      affinity
       exclude-any color red
    !
2   preference 200
     explicit segment-list SIDLIST1
   !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

Path preference 100

Dynamic path

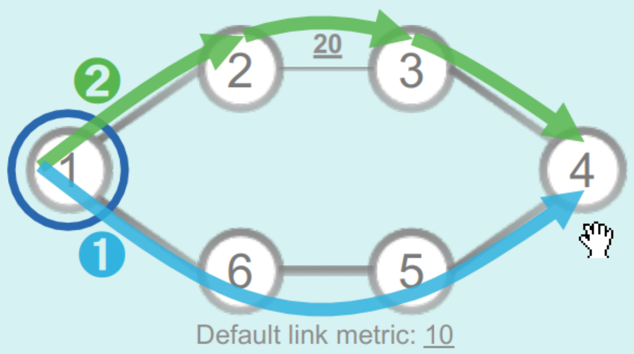
Opt. Obj.: TE metric

Constraint

Path preference 200

Explicit SID-list1

SID-list1



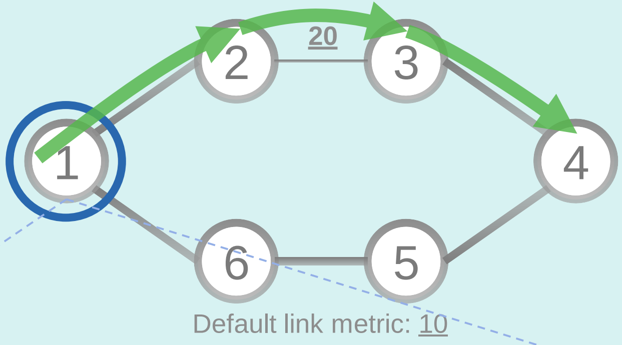

```
segment-routing
 traffic-eng
  affinity-map
   color red bit-position 0
```

SR Policy – Configuration Example

On Node1:

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
    preference 100
     dynamic
      metric type te
     constraints
      affinity
       exclude-any red
    !
    preference 200
     explicit segment-list SIDLIST1
  !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

Selected Path:
- Valid Path
- Highest Pref value



FIB @ head-end Node1
Incoming label: 1000
Action: pop and push <16002, 30203, 16004>

SR Policy – configuration example

On Node1:

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
    preference 100
     dynamic
      metric type te
     constraints
      affinity
       exclude-any color red
    !
    preference 200
     explicit segment-list SIDLIST1
  !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

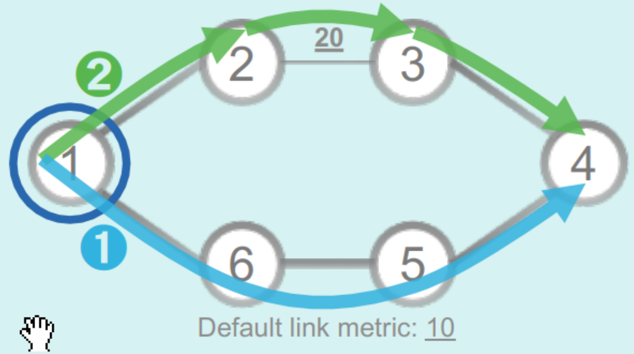
User-defined name

Color and End-point

SR Policy ID: (20,1.1.1.4)

Binding-SID

Local Candidate Paths




```
segment-routing
 traffic-eng
  affinity-map
   color red bit-position 0
```

SR Policy – configuration example

On Node1:

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
    preference 100
     dynamic mpls
      metric
       type te
      affinity
       exclude-any red
    !
    preference 200
     explicit segment-list SIDLIST1
  !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

Node1 may receive other candidate paths for SR Policy (20, 1.1.1.4) from other sources, some examples:

Source of path is not considered for path selection

Selected Path:
- Valid Path
- Highest Pref value

Other candidate paths received for SR Policy (20, 1.1.1.4)

Path received via BGP signaling
preference 150
binding-sid mpls 1000
weight 1, SID-list <16002, 16005>
weight 2, SID-list <16004, 16008>

Path received via PCEP signaling
preference 120
binding-sid mpls 1000
SID-list <16002, 16005>

Path received via NETCONF signaling
preference 50
binding-sid mpls 1000
SID-list <16002, 16005>

WECMP example

On Node1:

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
    preference 200
     explicit segment-list SIDLIST1
      weight 1
     !
     explicit segment-list SIDLIST2
      weight 4
     !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
  !
  segment-list name SIDLIST2
   index 10 address ipv4 1.1.1.4
```

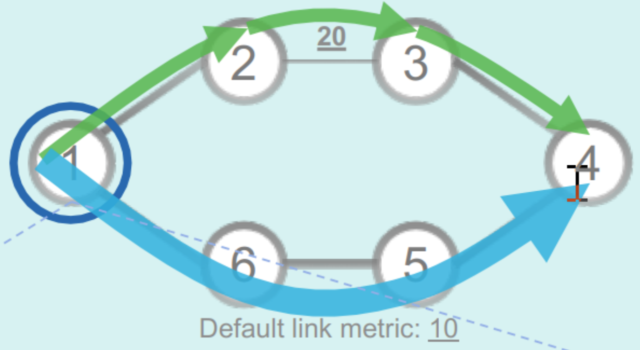
Path preference 200

Explicit SID-list1, Weight 1

Explicit SID-list2, Weight 4

SID-list1

SID-list2



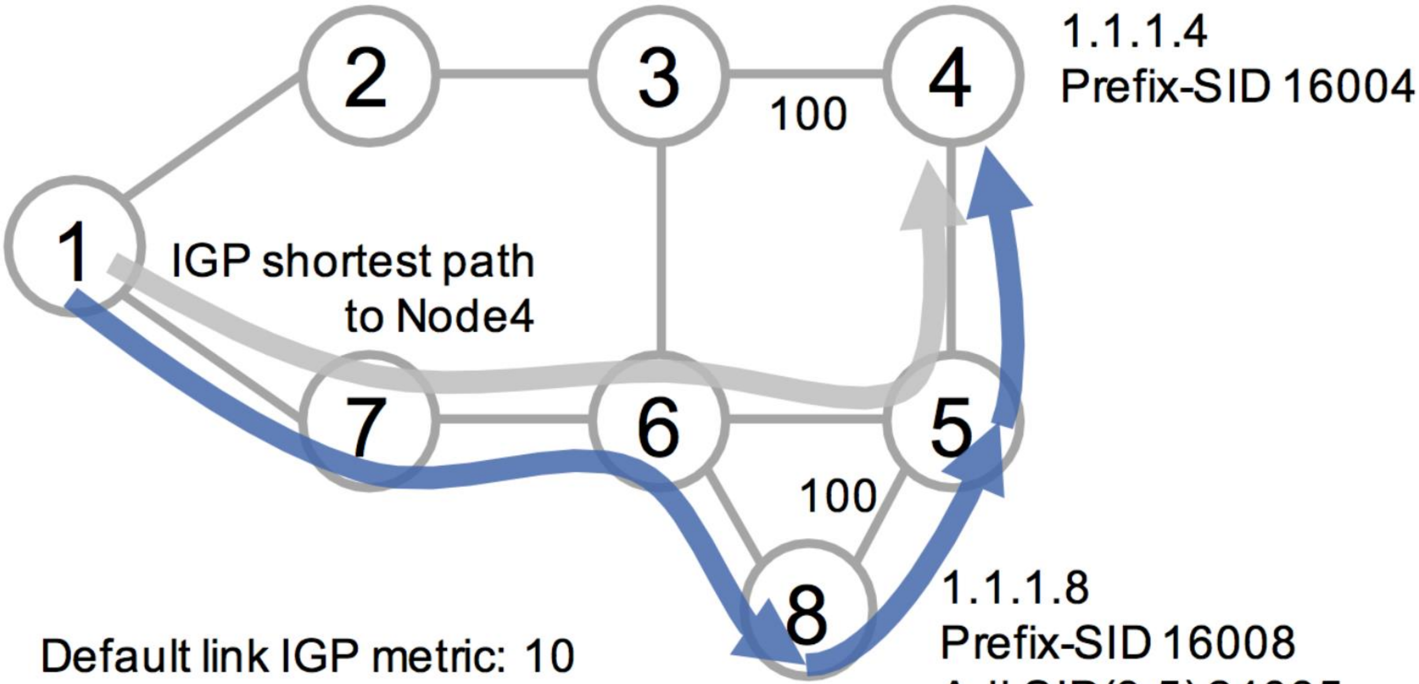
FIB @ head-end Node1

Incoming label: **1000**

Action:pop and push <16002, 30203, 16004> (20%)

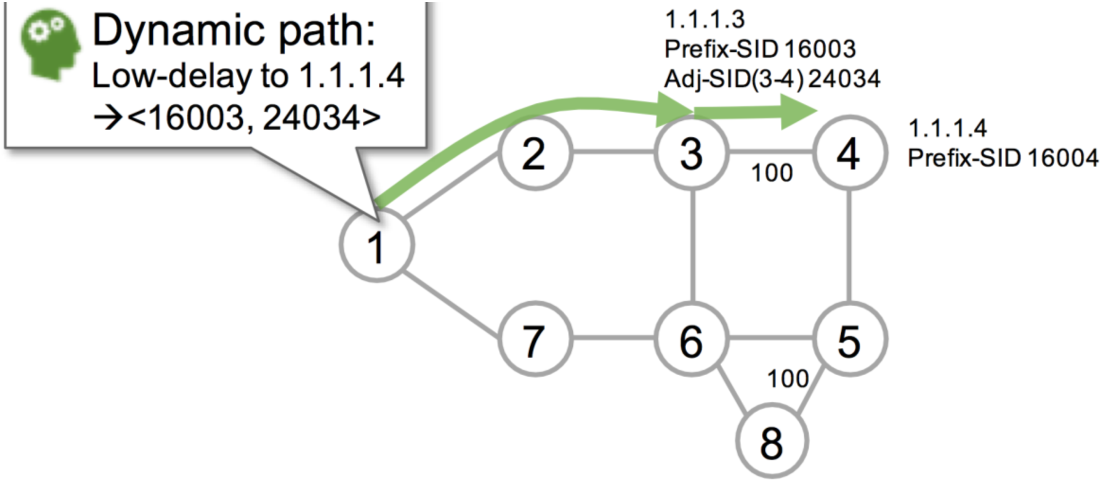
push <16004> (80%)

SR Policy : Dynamic Path & Explicit Path



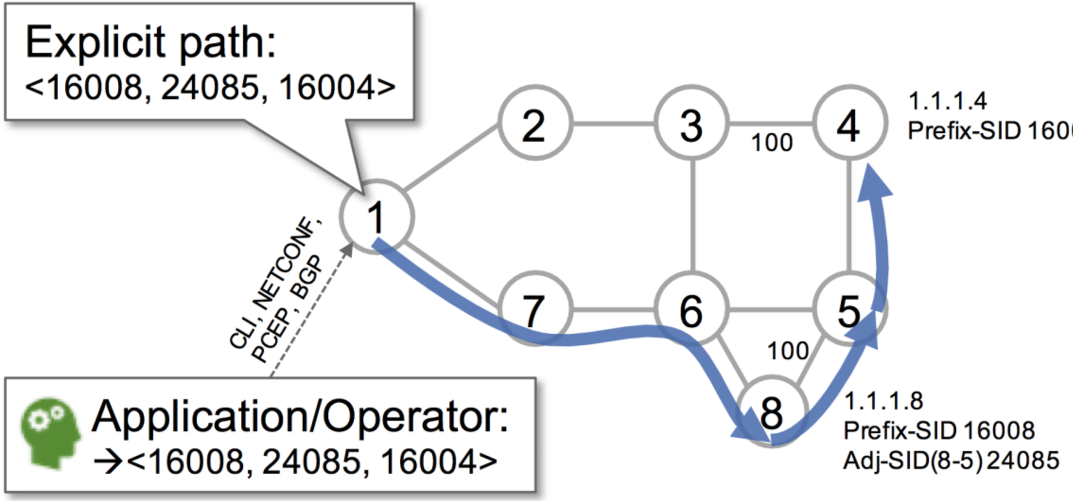
Dynamic Path

(a)



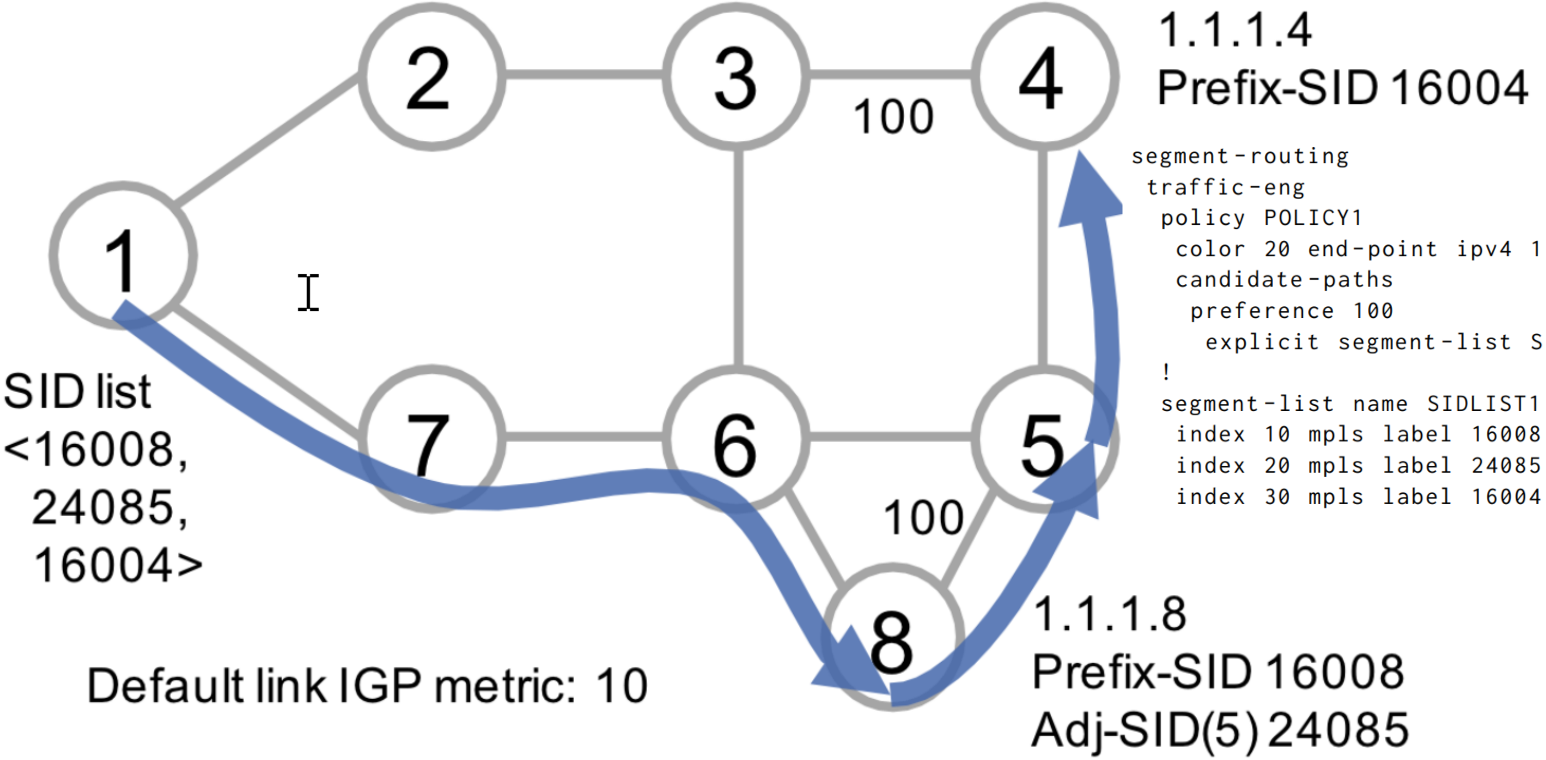
Explicit Path

(b)



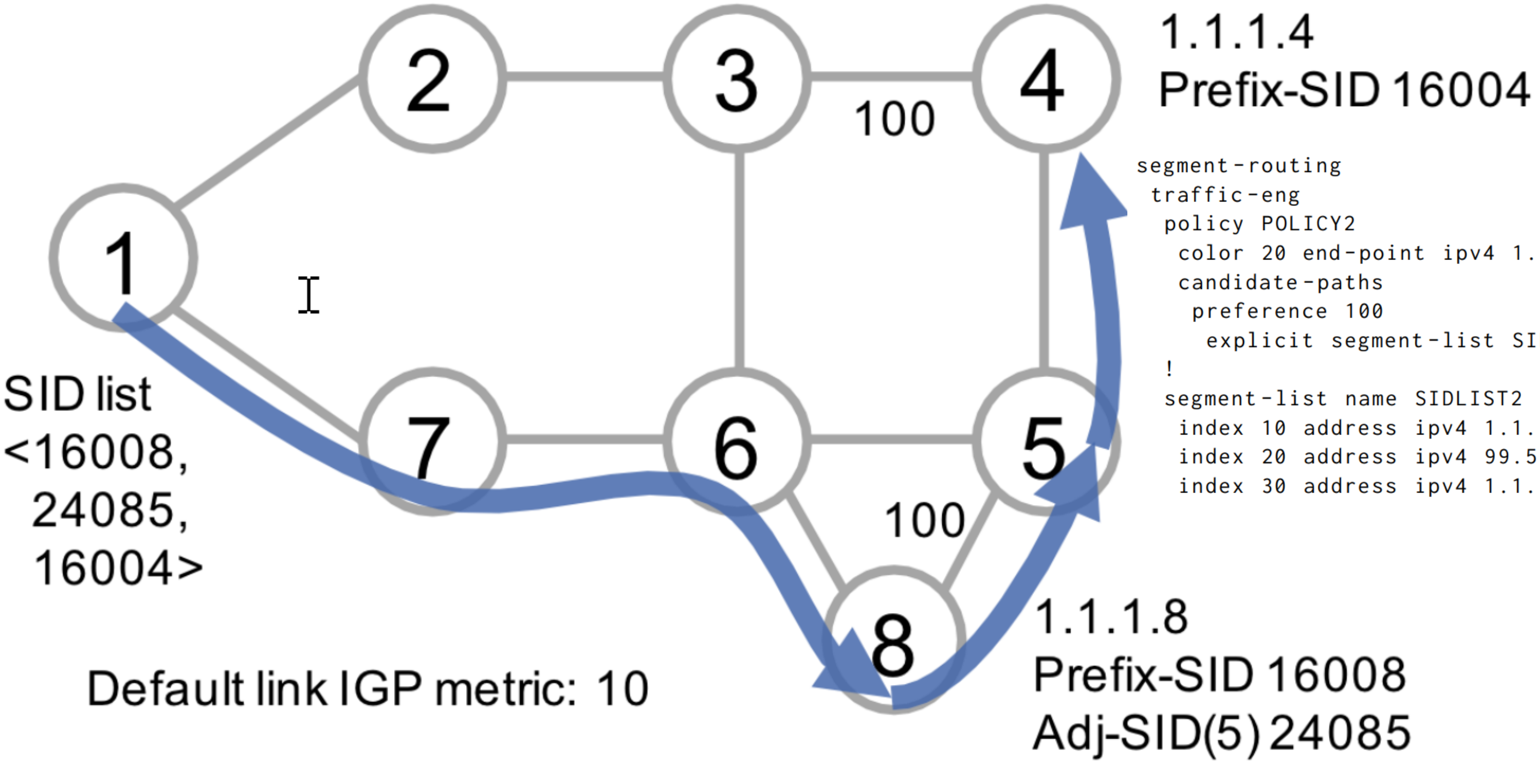
Default link IGP metric: 10
Default link-delay metric: ***10***

SR Policy : Explicit Path SID List with MPLS Label



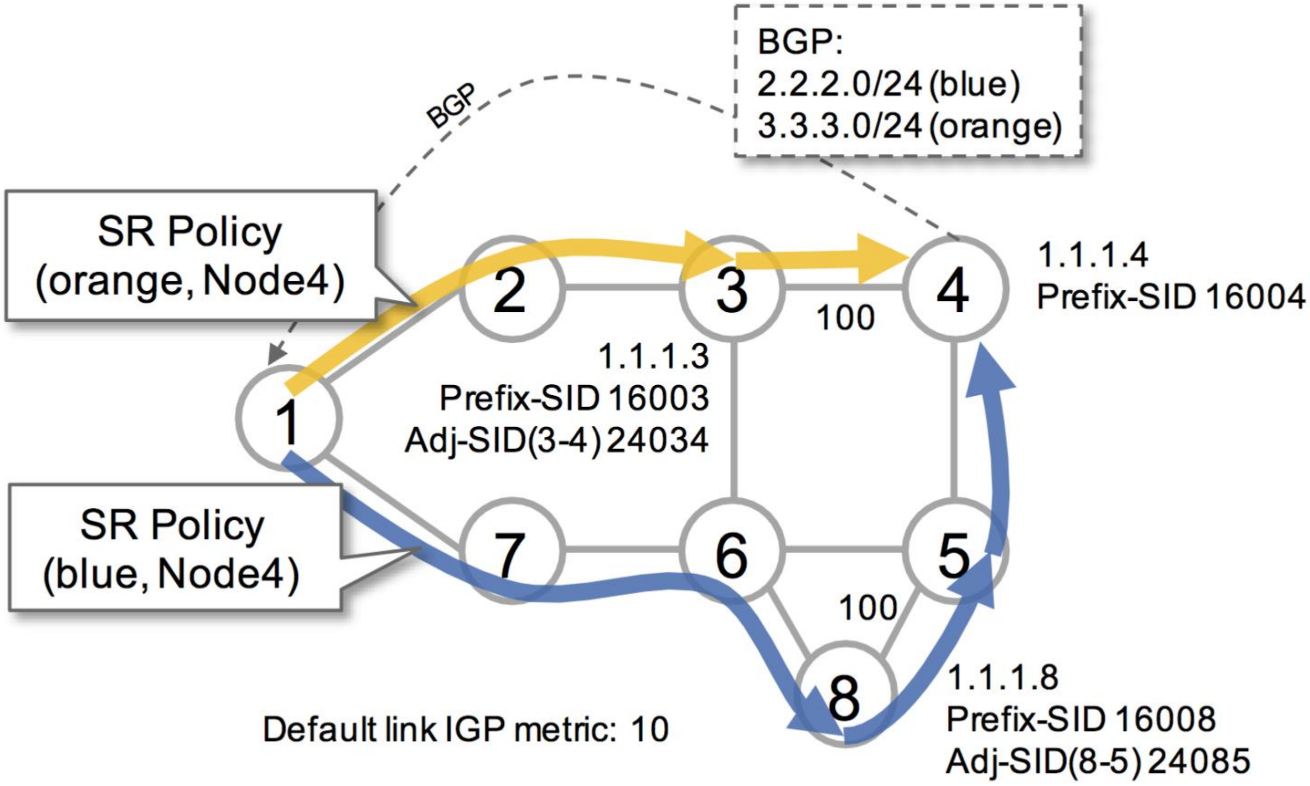

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   candidate-paths
    preference 100
     explicit segment-list SIDLIST1
  !
  segment-list name SIDLIST1
   index 10 mpls label 16008    !! Prefix-SID Node8
   index 20 mpls label 24085    !! Adj-SID link 8->5
   index 30 mpls label 16004    !! Prefix-SID Node4
```

SR Policy : Explicit Path SID List with IP Address

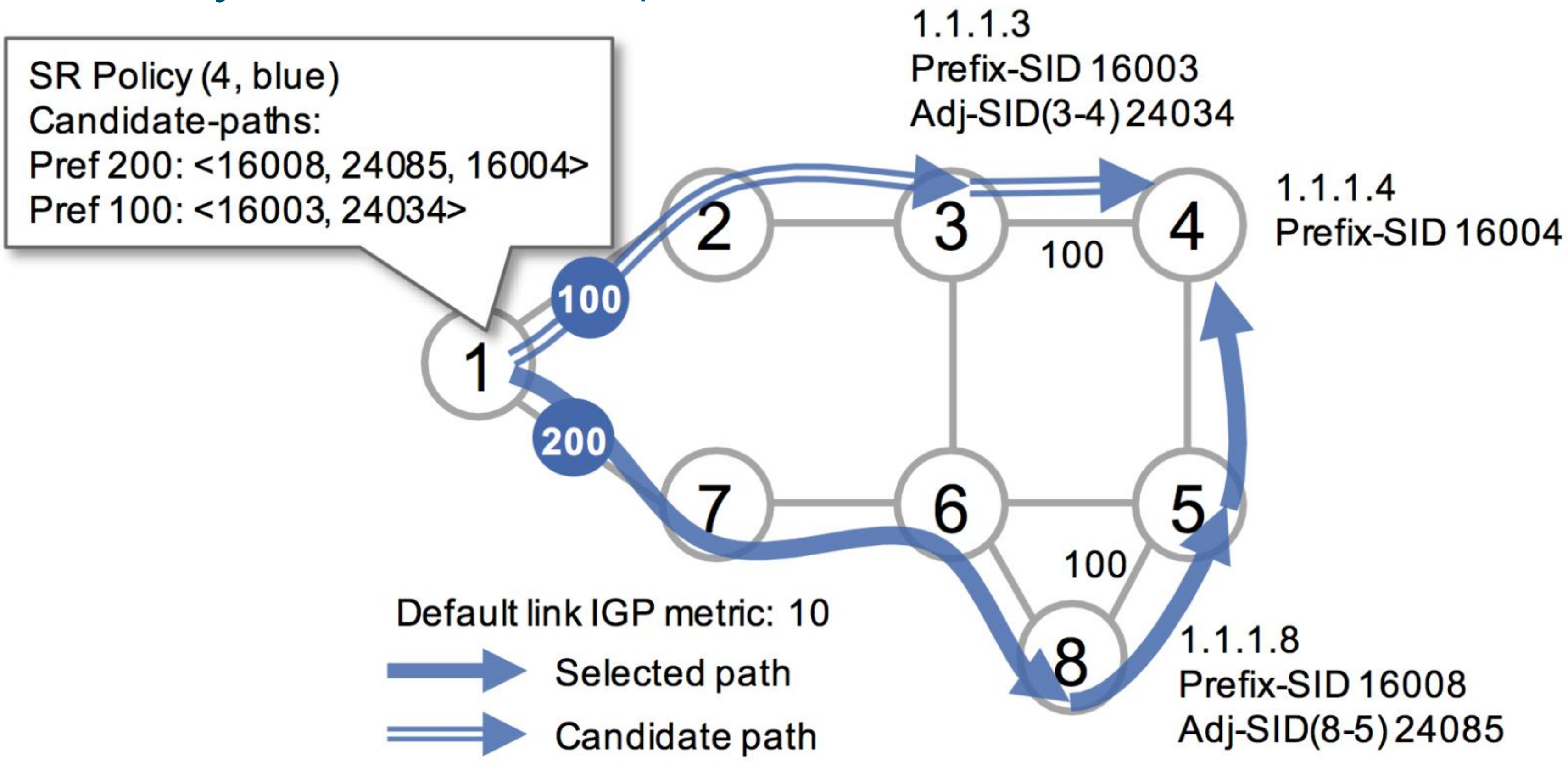



```
segment-routing
 traffic-eng
  policy POLICY2
   color 20 end-point ipv4 1.1.1.4
   candidate-paths
    preference 100
     explicit segment-list SIDLIST2
  !
  segment-list name SIDLIST2
   index 10 address ipv4 1.1.1.8    !! Prefix-SID Node8
   index 20 address ipv4 99.5.8.8   !! Adj-SID link 8->5
   index 30 address ipv4 1.1.1.4    !! Prefix-SID Node4
```

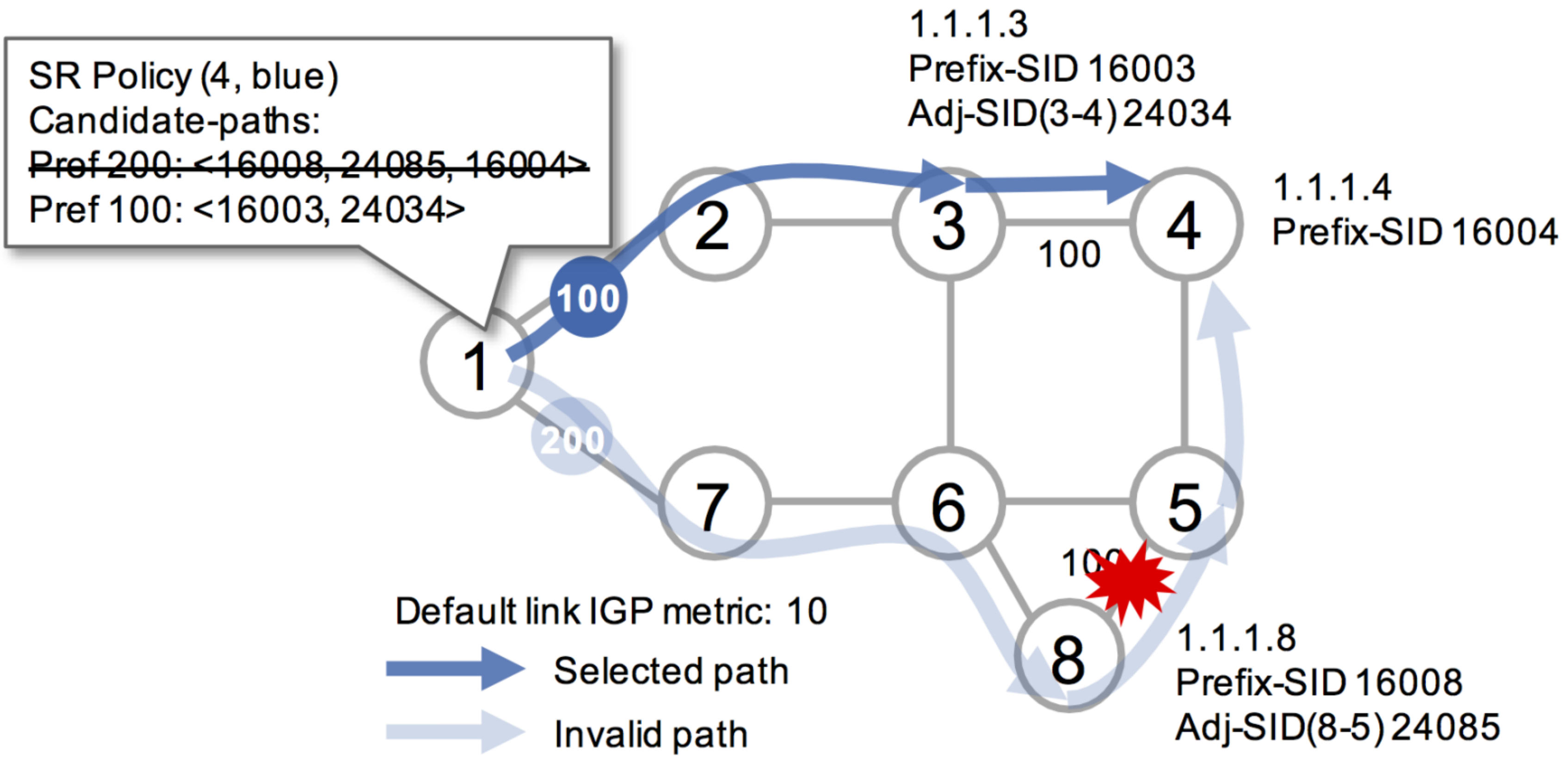
SR Policy with Single Explicit Candidate Path



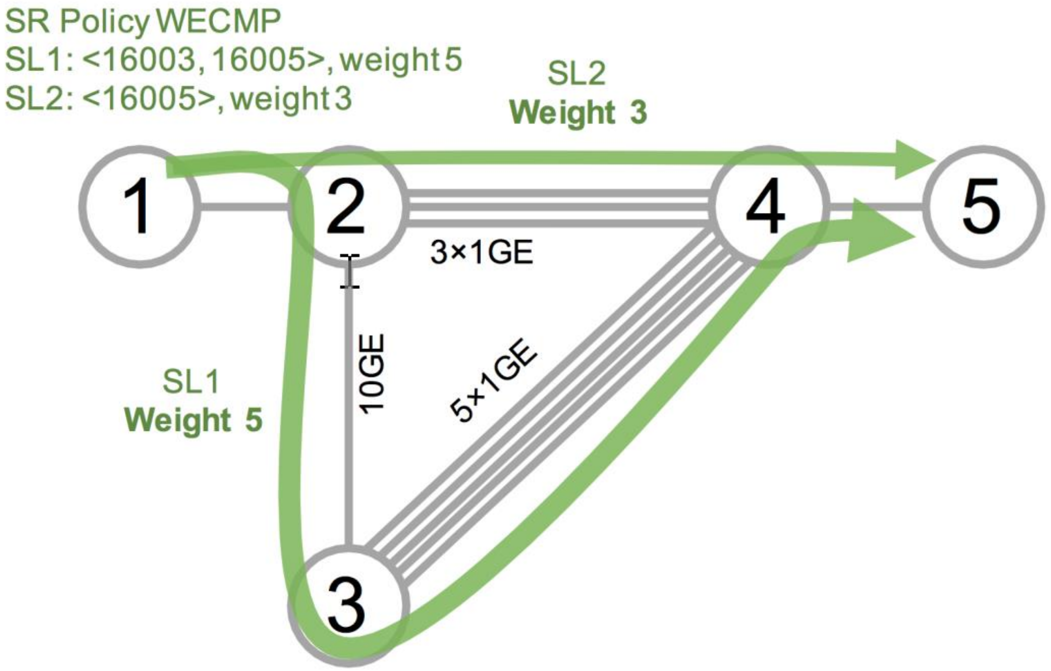
SR Policy with Two Explicit Candidate Paths



SR Policy with Two Explicit Candidate Paths

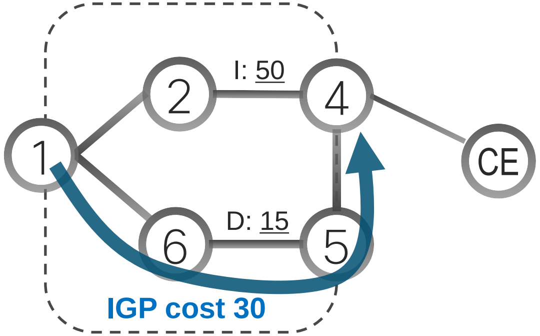


Use-case : Weighted ECMP SR Policy



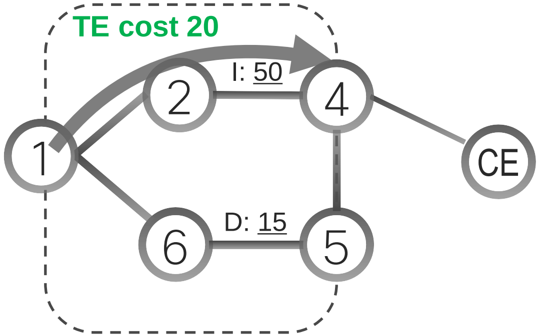
segment - routing
traffic -eng
segment - list SL1
index 10 mpls label 16003
index 20 mpls label 16005
!
segment - list SL2
index 10 mpls label 16005
!
policy WECMP
color 20 end - point ipv4 1.1.1.5
candidate - paths
preference 100
explicit segment - list SL1
weight 5
!
explicit segment - list SL2
weight 3

Different VPNs need different underlay SLA



Default IGP cost: I:10
Default Delay cost: D:10

Basic VPN should use lowest cost underlay path

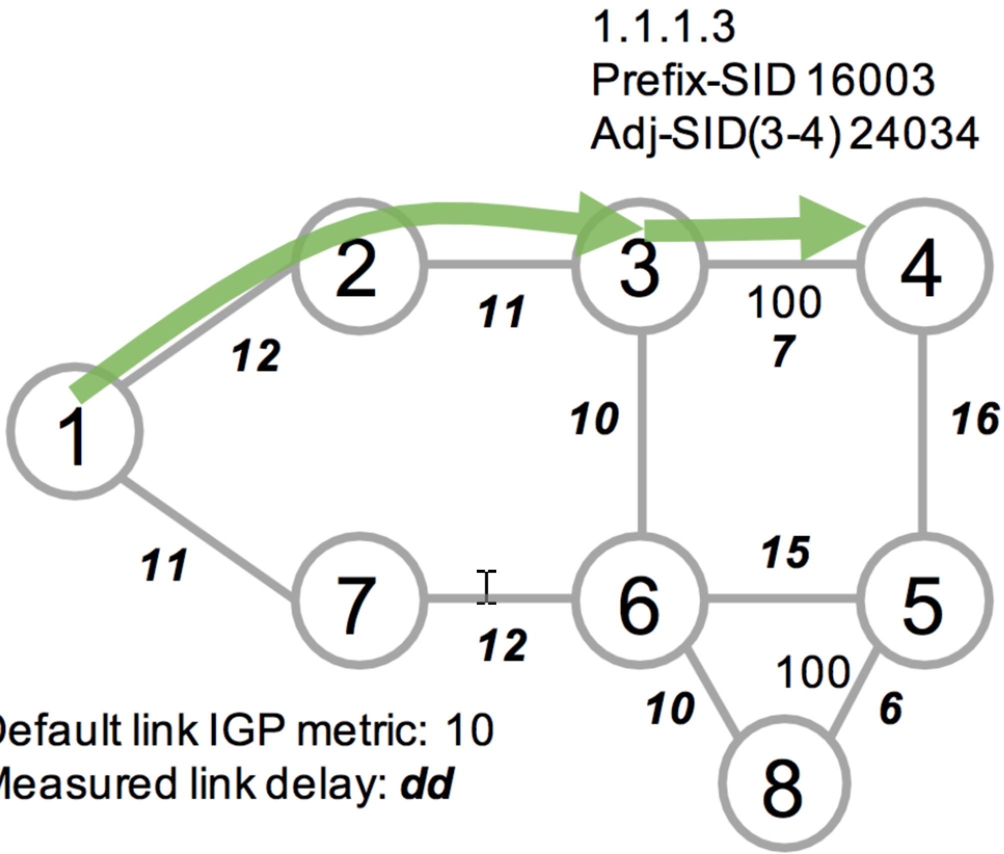


Default IGP cost: I:10
Default Delay cost: D:10

Premium VPN should use lowest delay path

Objective: operationalize this service for simplicity, scale and performance

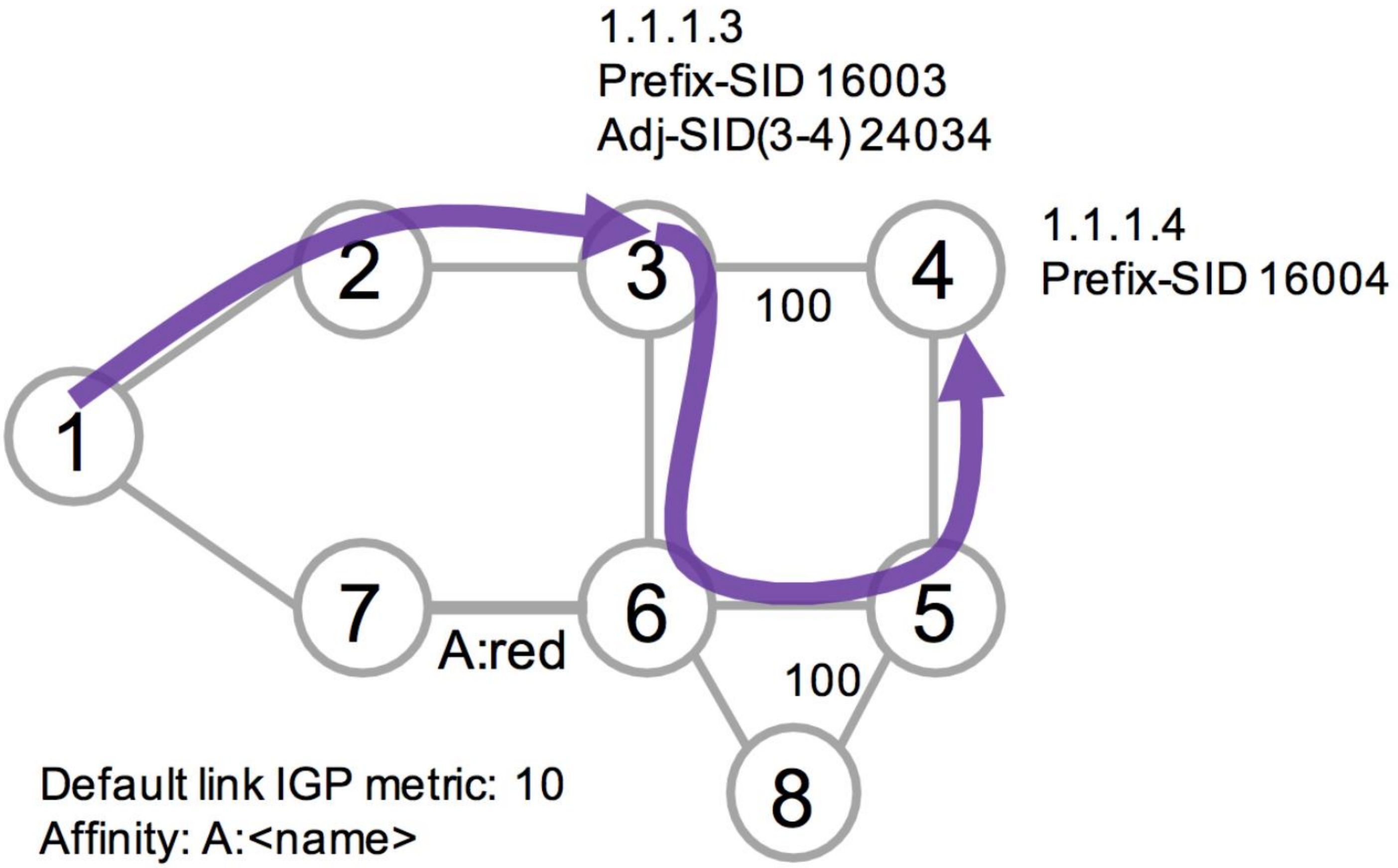
SR Policy with Dynamic Path + Low Delay Intent



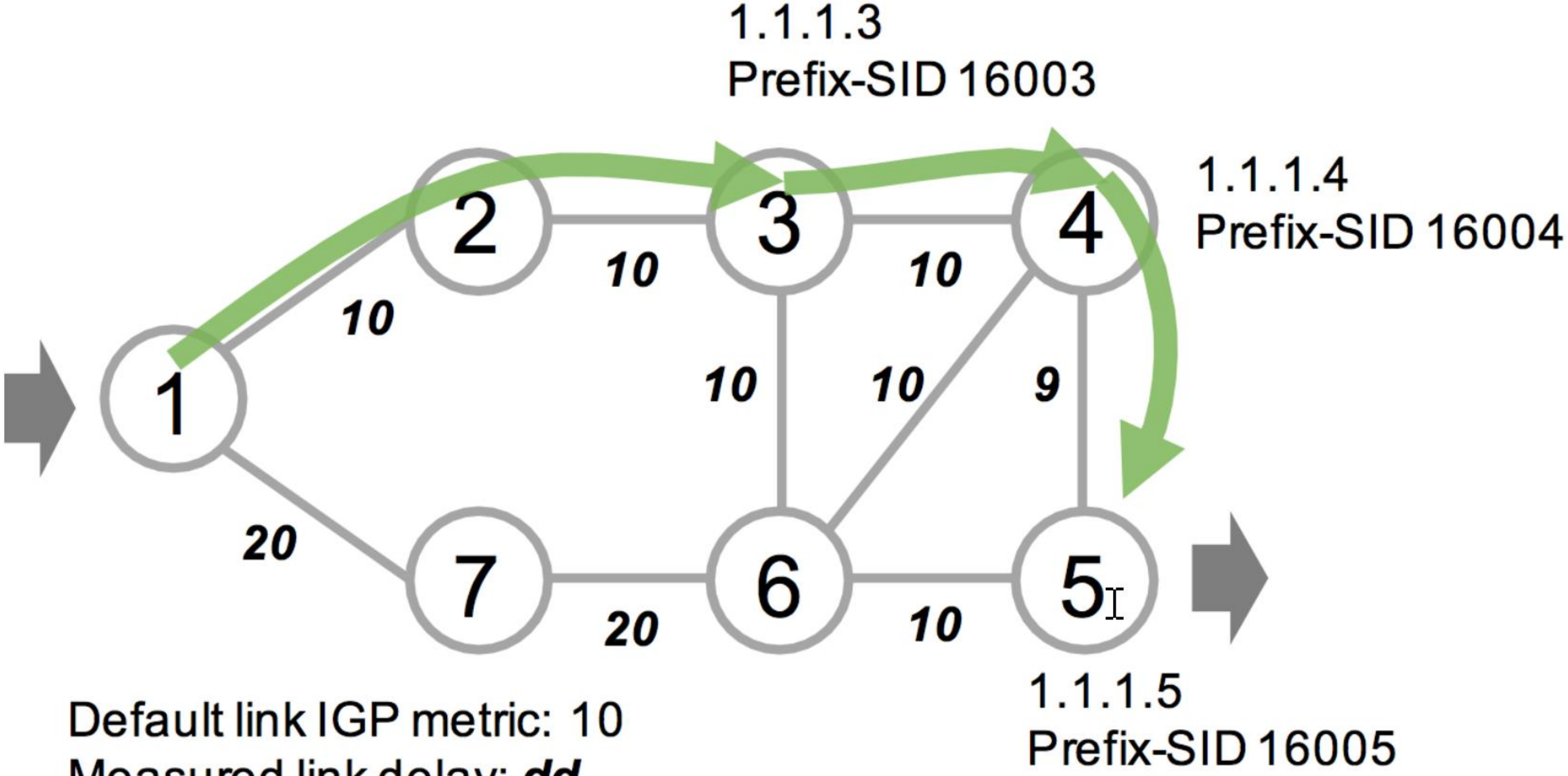

```
segment-routing
 traffic-eng
  policy POLICY2
   color 30 end-point ipv4 1.1.1.4
   candidate-paths
    preference 100
     dynamic mpls
      metric
       type delay
```

Default link IGP metric: 10
Measured link delay: ***dd***

SR Policy with Dynamic Path + Avoiding Specific Link



Use-Case : Low Delay Margin for ECMP



Default link IGP metric: 10
Measured link delay: ***dd***

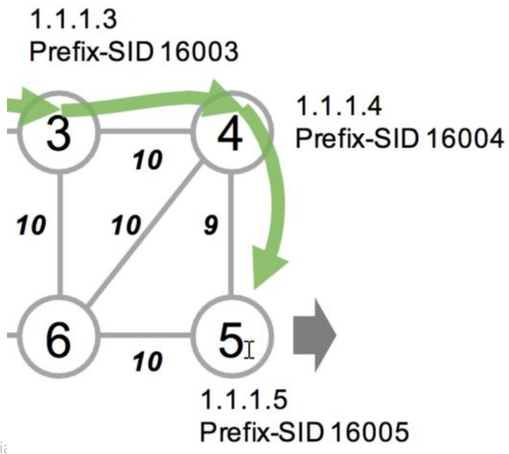
Use-Case : Low Delay Margin for ECMP

```
segment-routing
  traffic-eng
    policy LOW-DELAY
      color 20 end-point ipv4 1.1.1.5
      candidate-paths
        preference 100
          dynamic
            metric
              type delay
```

No ECMP

R3 – R4 – R5 (delay = 19)

Lowest Delay Metric = 19




```
segment-routing
  traffic-eng
    policy LOW-DELAY
      color 20 end-point ipv4 1.1.1.5
      candidate-paths
        preference 100
          dynamic
            metric
              type delay
              margin absolute 2000
```

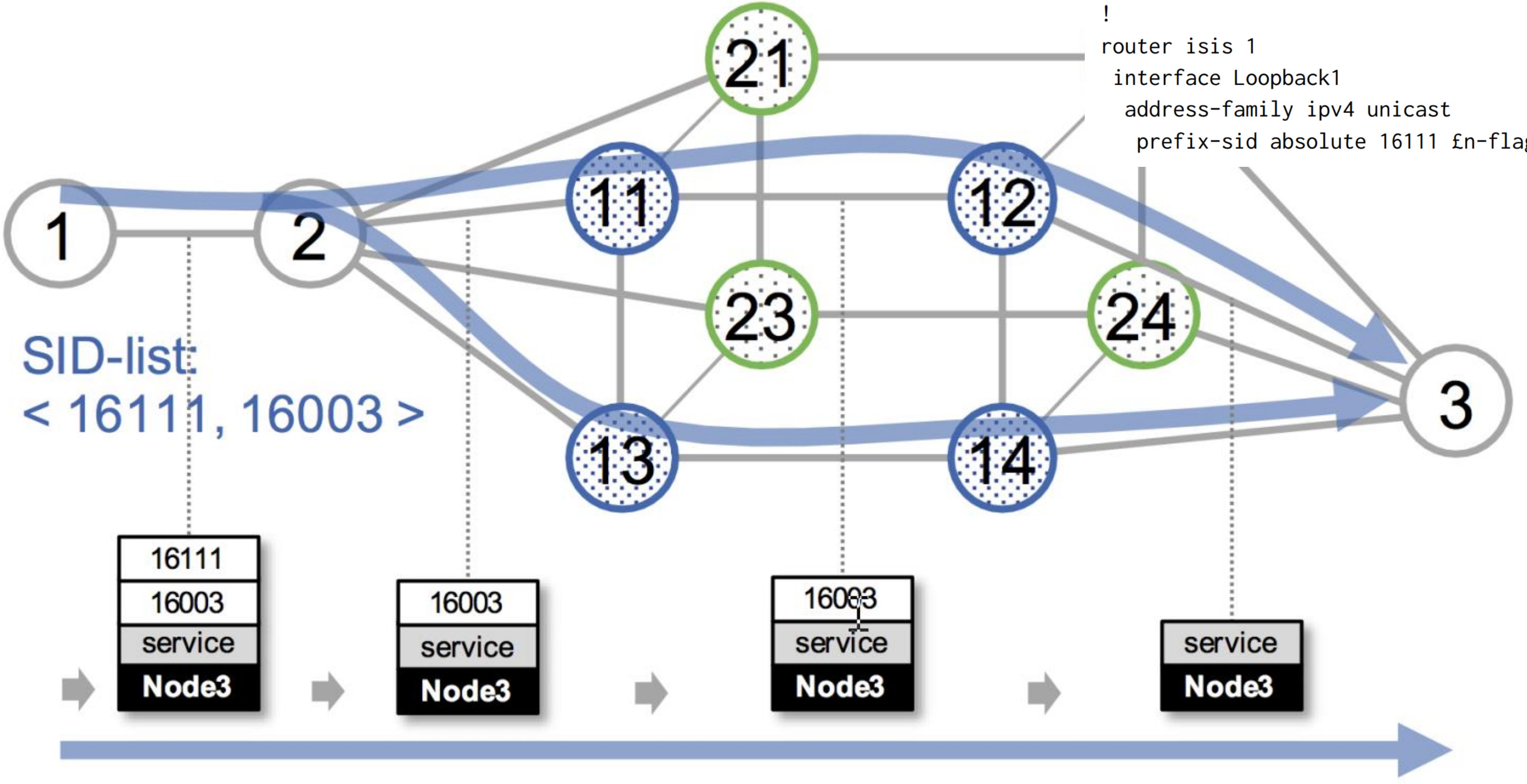
ECMP

R3 – R4 – R5 (delay = 19)
R3 – R6 – R5 (delay = 20)

Margin = 2 ms (2000 us)
Lowest + 2 = 19 + 2 = 21 ms

Dual Plane Disjoint with Anycast-SID

```
interface Loopback1
 description blue plane anycast address
 ipv4 address 1.1.1.111/32
!
router isis 1
 interface Loopback1
  address-family ipv4 unicast
   prefix-sid absolute 16111 £n-flag-clear£
```

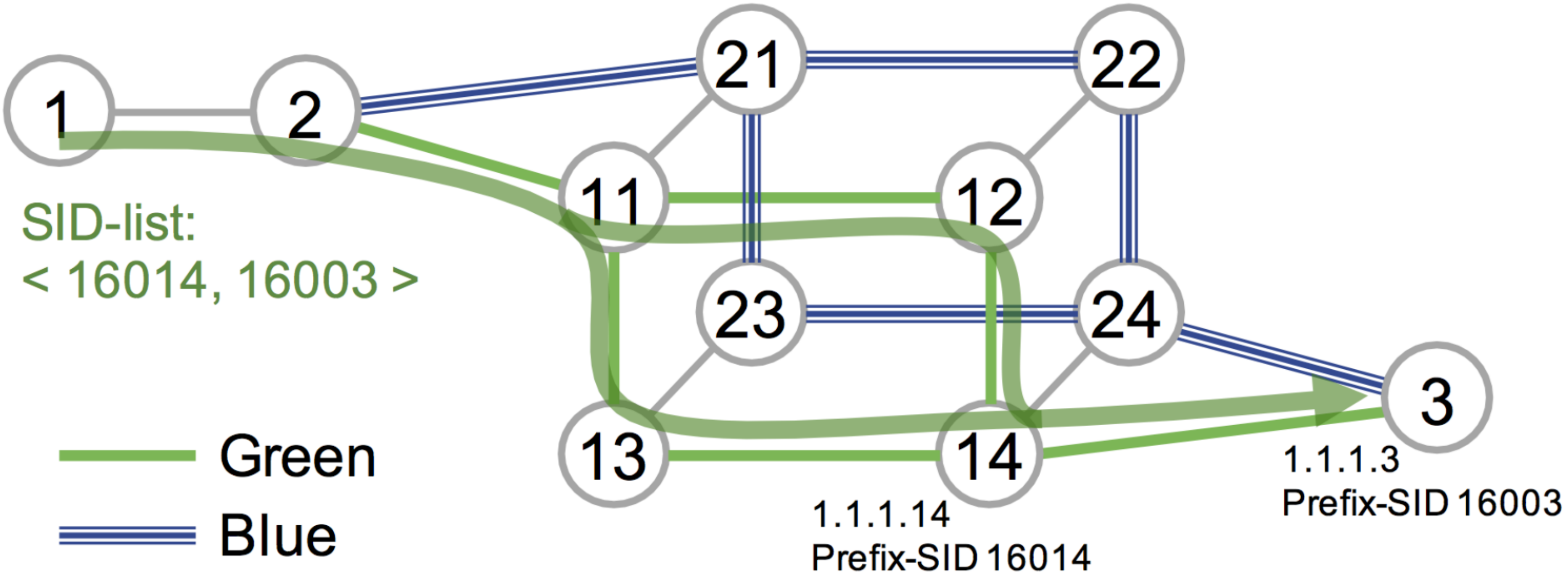


Dual Plane Disjoint with Anycast-SID

```
interface Loopback1
 description blue plane anycast address
 ipv4 address 1.1.1.111/32
!
router isis 1
 interface Loopback1
  address-family ipv4 unicast
   prefix-sid absolute 16111 £n-flag-clea
```

```
segment-routing
 traffic-eng
  policy BLUE
   color 10 end-point ipv4 1.1.1.3
   candidate-paths
    preference 100
     explicit segment-list SIDLIST1
  !
  policy GREEN
   color 20 end-point ipv4 1.1.1.3
   candidate-paths
    preference 100
     explicit segment-list SIDLIST2
  !
  segment-list name SIDLIST1
   index 10 address ipv4 1.1.1.111  !! blue plane anycast
   index 20 address ipv4 1.1.1.3
  !
  segment-list name SIDLIST2
   index 10 address ipv4 1.1.1.222  !! green plane anycast
   index 20 address ipv4 1.1.1.3
```

Use-Case : Dual Plane Affinity Resource Avoidance



Use-Case : Dual Plane Affinity Resource Avoidance

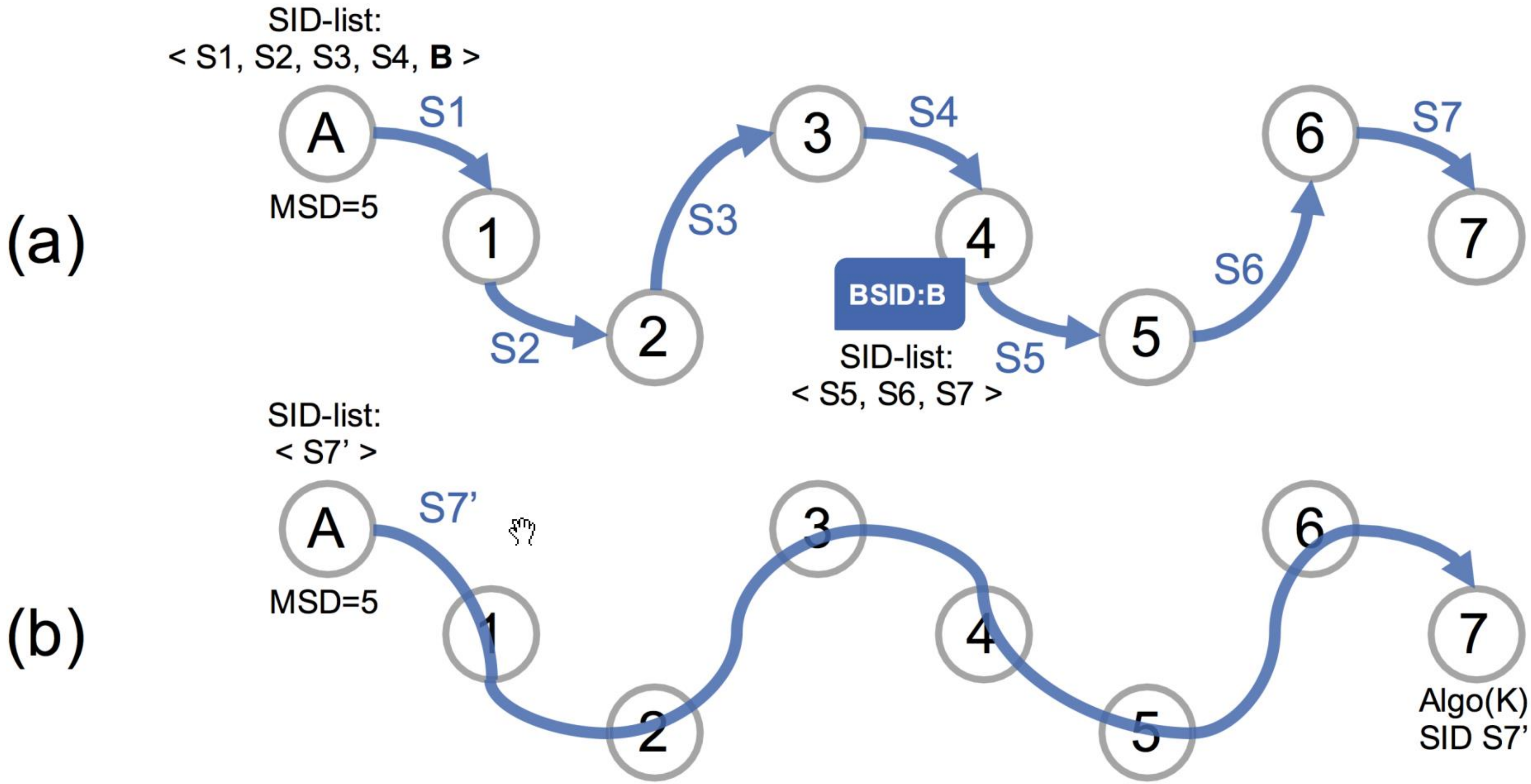
```
segment-routing
 traffic-eng
  affinity-map
   name GREEN bit-position 0
   name BLUE bit-position 2
  !
  interface Gi0/0/0/0
   !! link to Node11, in plane Green
   affinity name GREEN
  !
  interface Gi0/0/0/1
   !! link to Node21, in plane Blue
   affinity name BLUE
```

All Nodes & consistents

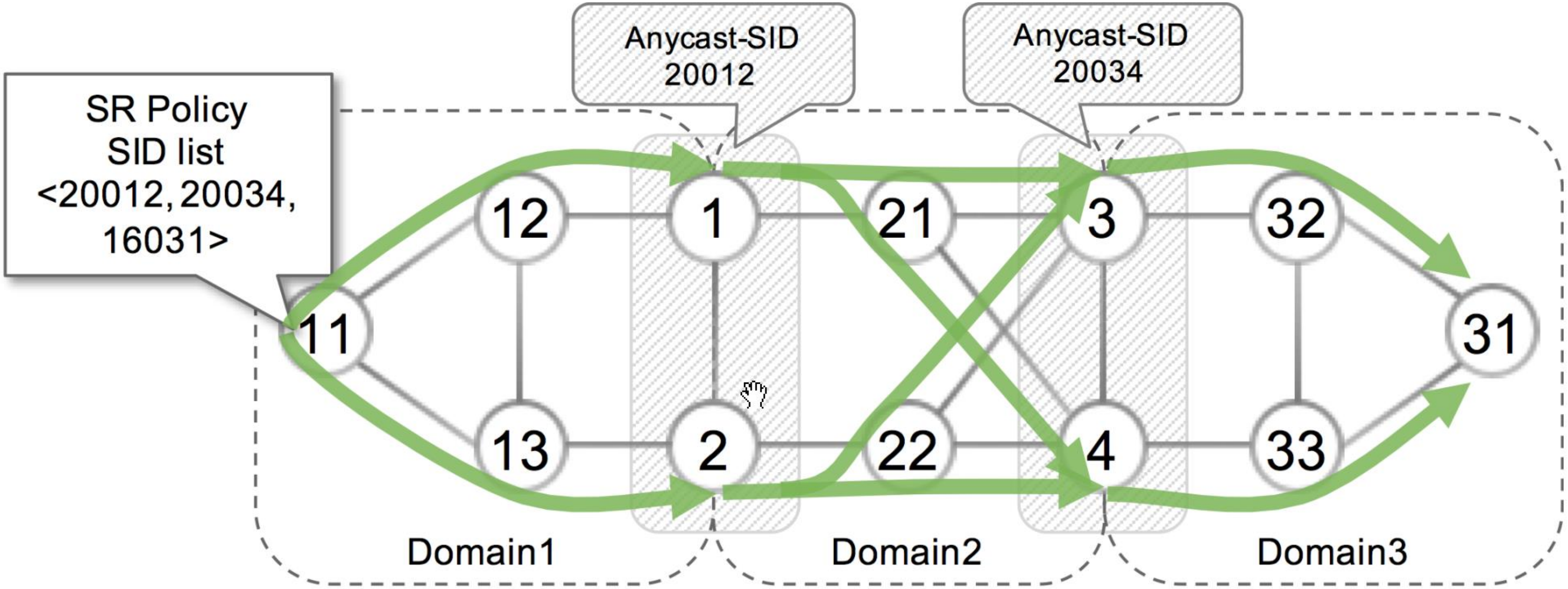
Head End – R2

```
segment-routing
 traffic-eng
  policy VIA-PLANE-GREEN
   color 20 end-point ipv4 1.1.1.3
   candidate-paths
    preference 100
     dynamic
      metric
       type igp
     !
     constraints
      affinity
       exclude-any
        name BLUE
  !
  policy VIA-PLANE-BLUE
   color 30 end-point ipv4 1.1.1.3
   candidate-paths
    preference 100
     dynamic
      metric
       type igp
     !
     constraints
      affinity
       exclude-any
        name GREEN
```

How many SIDs ?

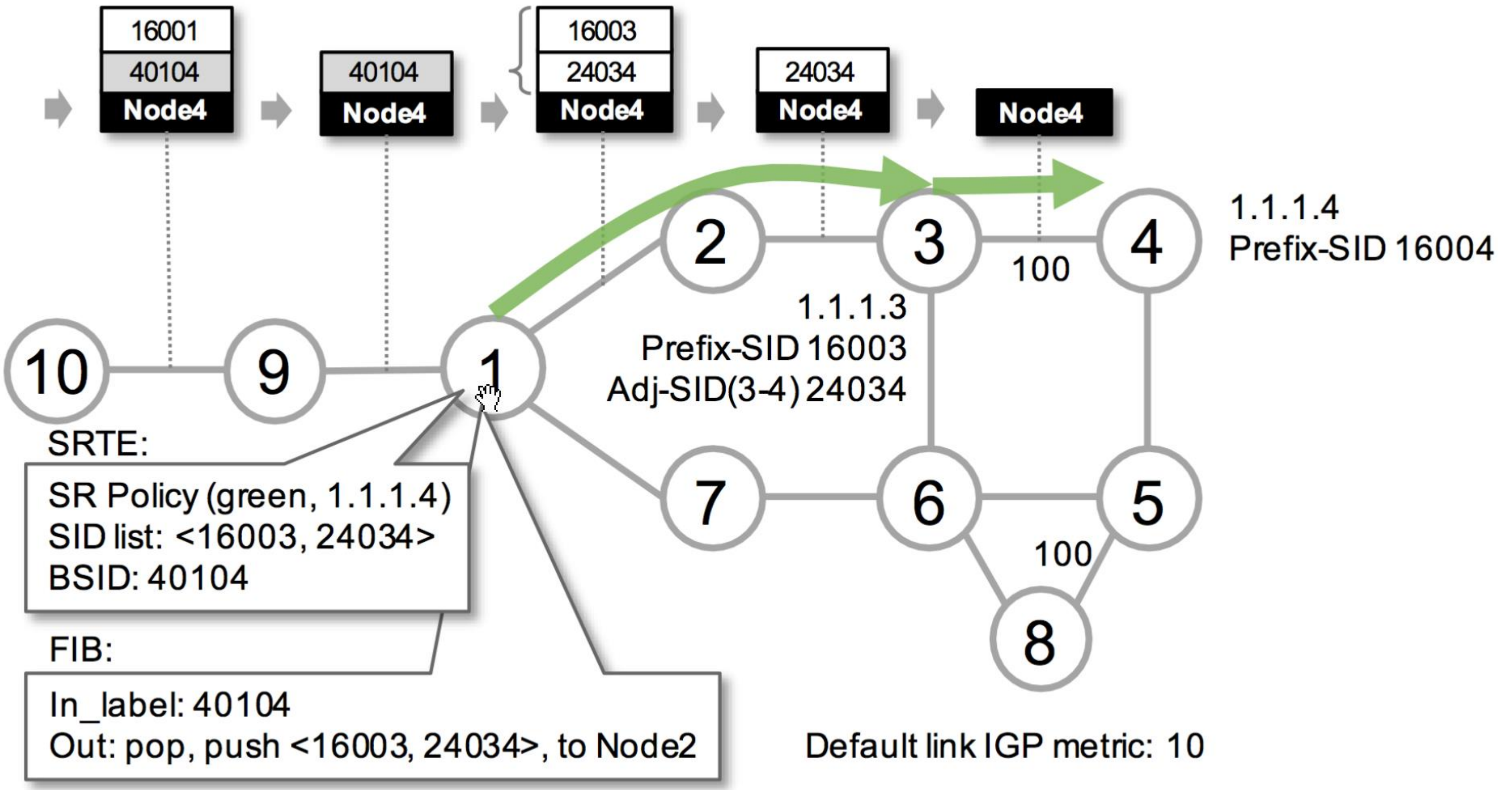


How many SIDs ?

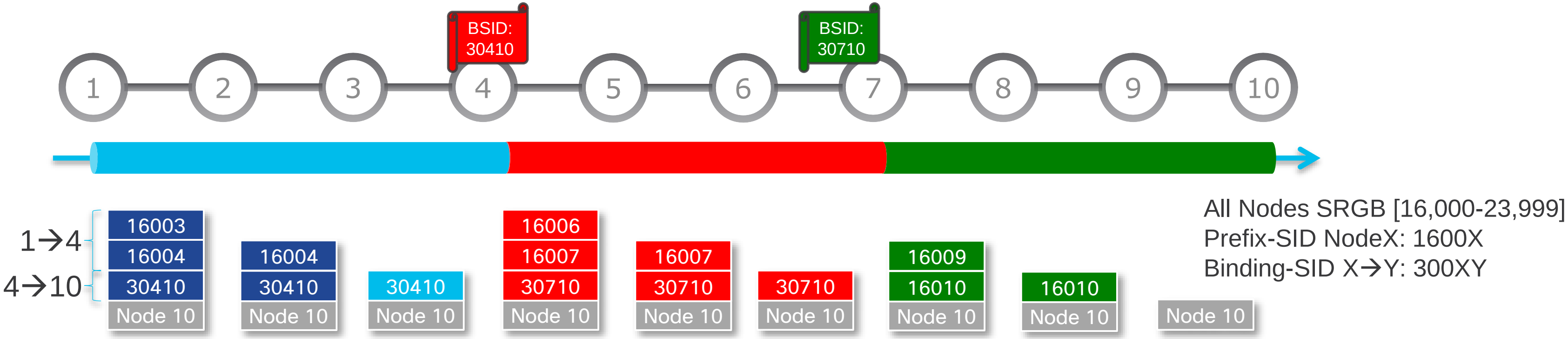


Default link IGP metric: 10

Binding SID

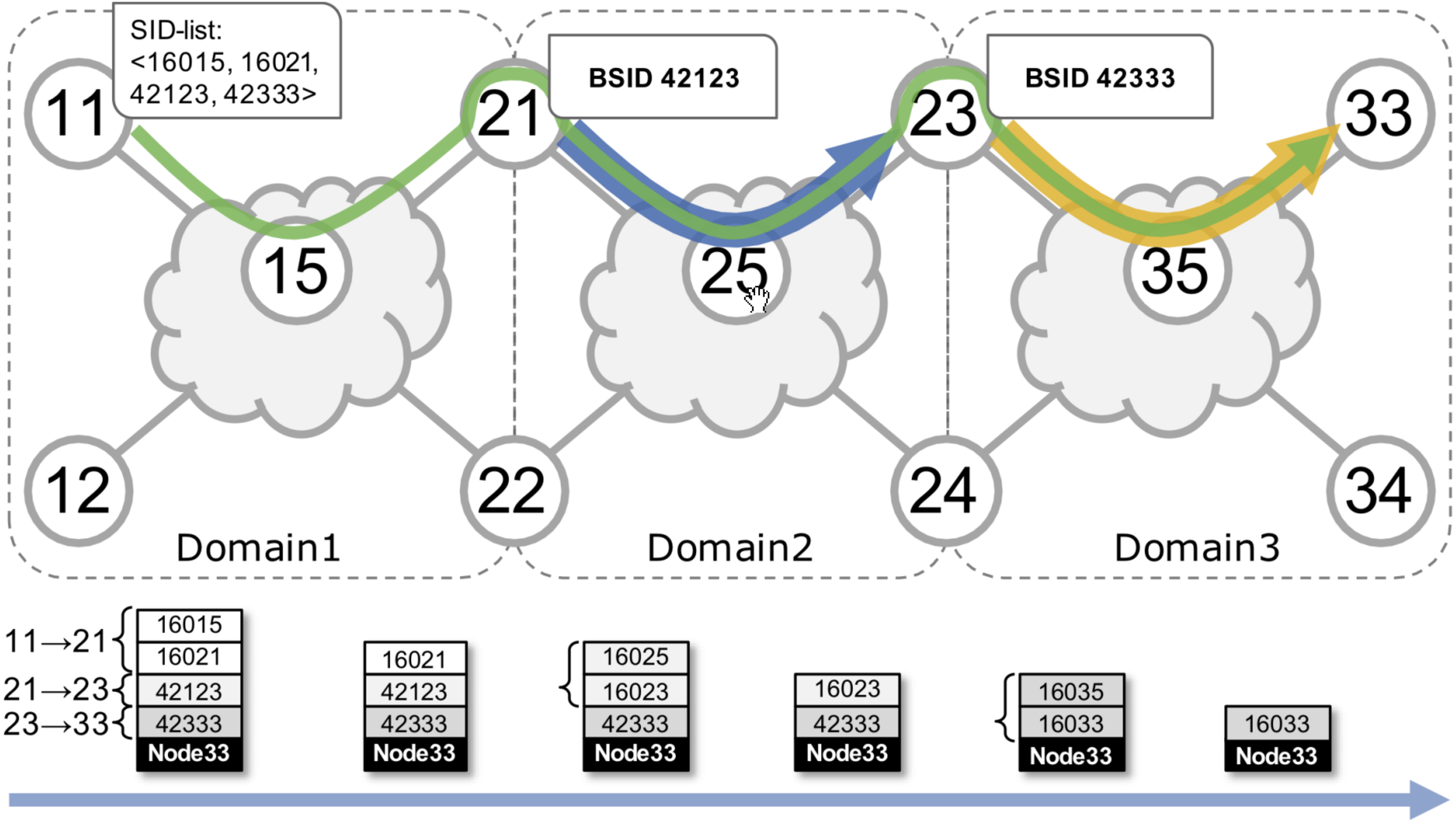

16001
40104
Node4
40104
Node4
16003
24034
Node4
24034
Node4
Node4
1.1.1.4
Prefix-SID 16004
100
1.1.1.3
Prefix-SID 16003
Adj-SID(3-4) 24034
10
9
1
2
3
4
7
6
5
8
100
SRTE:
SR Policy (green, 1.1.1.4)
SID list: <16003, 24034>
BSID: 40104
FIB:
In_label: 40104
Out: pop, push <16003, 24034>, to Node2
Default link IGP metric: 10

Binding-SID – Stitching – Illustration

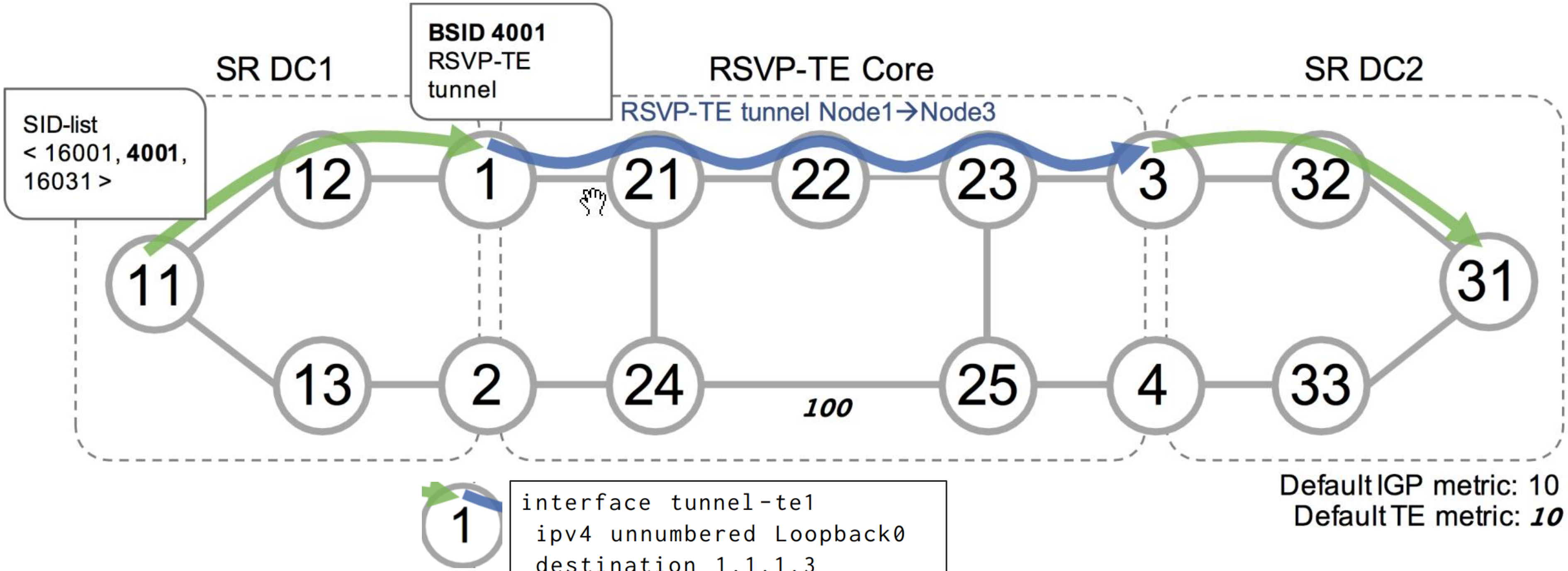


- Assume Node1 can't push 8 labels to go to Node10
  - "compress" label stack by stitching SRTE Policies:
    - Node1 pushes:
      - 2 labels to go to Node4
      - Binding-SID to go to Node10
    - Node4 pops Binding-SID and pushes:
      - 2 labels to go to Node7
      - Binding-SID to go to Node10
    - Node7 pops Binding-SID and pushes 2 labels to go to Node10

Binding SID


SID-list:
<16015, 16021,
42123, 42333>
BSID 42123
BSID 42333
11
21
23
33
15
25
35
12
22
24
34
Domain1
Domain2
Domain3
11→21
21→23
23→33
16015
16021
42123
42333
Node33
16021
42123
42333
Node33
16025
16023
42333
Node33
16023
42333
Node33
16035
16033
Node33
16033
Node33

Binding SID for RSVP-TE Tunnel




```
interface tunnel-te1
 ipv4 unnumbered Loopback0
 destination 1.1.1.3
 binding-sid mpls label 4001
 path-selection metric te
 path-option 1 dynamic
```

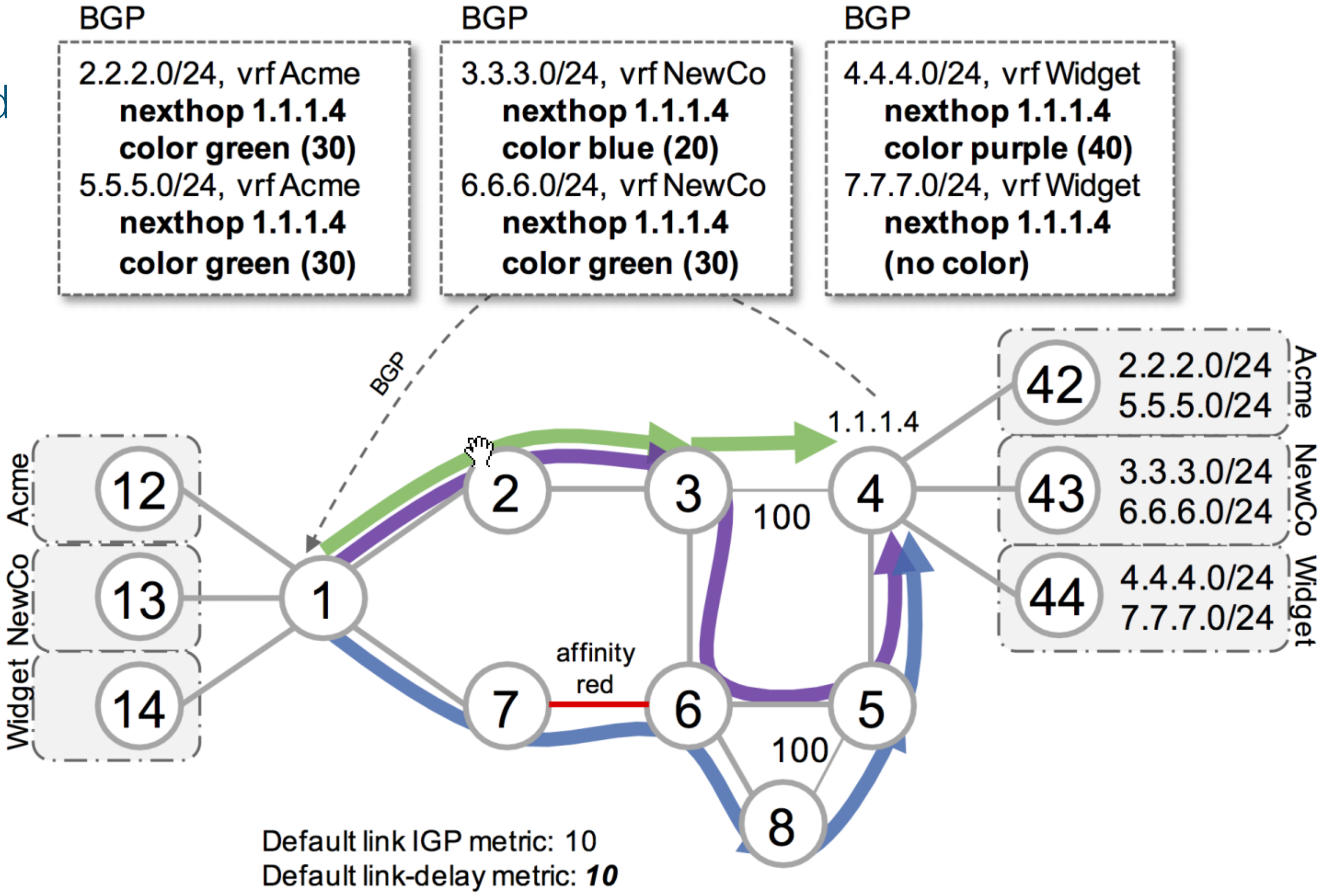
Segment Routing Automated Steering

Intent-based

Automated Steering

L3VPN

BGP-based



Intent-based Automated Steering L3VPN

1

```
segment - routing
traffic -eng
affinity -map
name red bit - position 0
!
segment - list name BLUE_PATH
index 10 mpls label 16008
index 20 mpls label 24085
index 30 mpls label 16004
!
policy BLUE
color 20 end - point ipv4 1.1.1.4
candidate - paths
preference 100
explicit segment - list BLUE_PATH
!
```

1

```
policy GREEN
color 30 end - point ipv4 1.1.1.4
candidate - paths
preference 100
dynamic
metric
type delay
!
policy PURPLE
color 40 end - point ipv4 1.1.1.4
candidate - paths
preference 100
dynamic
metric
type igp
constraints
affinity
exclude -any
name red
```

4

```
extcommunity -set opaque BLUE
20
end -set
!
extcommunity -set opaque GREEN
30
end -set
!
extcommunity -set opaque PURPLE
40
end –set

route - policy VRF - COLOR
 if destination in (3.3.3.0/24) then
 set extcommunity color BLUE
 endif
 if destination in (6.6.6.0/24) then
 set extcommunity color GREEN
 endif
 pass
end - policy
!
route - policy GLOBAL - COLOR
 if destination in (8.8.8.0/24) then
 set extcommunity color PURPLE
 endif
 pass
end - policy
```

4

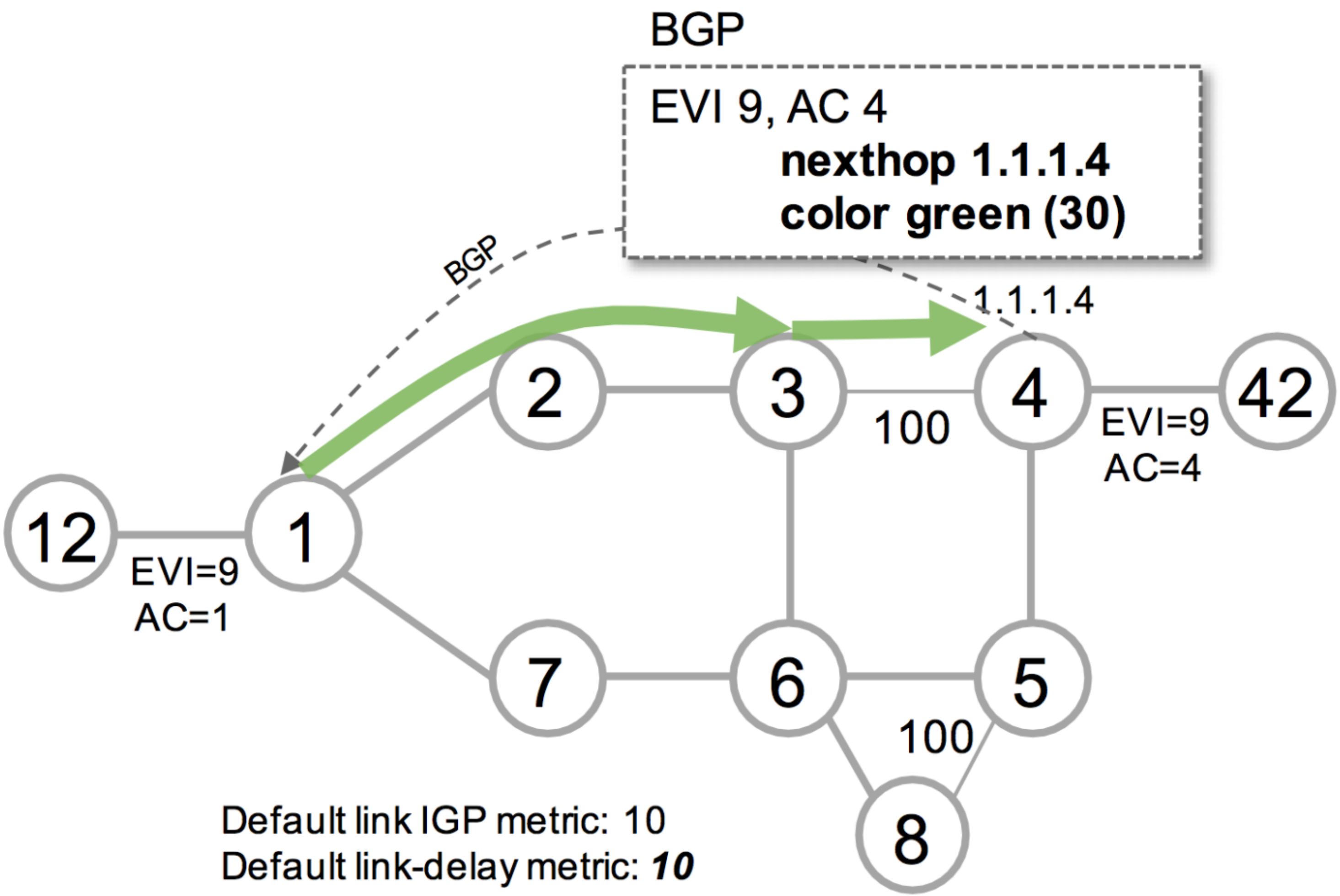
```
router bgp 1
 bgp router -id 1.1.1.4
 address - family ipv4 unicast
 address - family vpnv4 unicast
 !
 neighbor 1.1.1.1
 remote -as 1
 update - source Loopback0
 address - family ipv4 unicast
 route - policy GLOBAL - COLOR out
 !
 address - family vpnv4 unicast
 !
 vrf NewCo
 rd auto
 address - family ipv4 unicast
 !
 neighbor 99.4.43.43
 remote -as 2
 description to CE43
 address - family ipv4 unicast
 route - policy VRF - COLOR out
```

Intent-based

Automated Steering

EVPN

BGP-based



Intent-based Automated Steering EVPN VPWS

1

```
segment - routing
traffic -eng
policy GREEN
color 30 end - point ipv4 1.1.1.4
candidate - paths
preference 100
dynamic mpls
metric
type delay

interface GigabitEthernet0 /0/0/0.100
l2transport
encapsulation dot1q 100
!
l2vpn
xconnect group evpn - vpws
p2p EVI9
interface GigabitEthernet0 /0/0/0.100
neighbor evpn evi 9 target 4 source 1
```

1

```
router bgp 1
bgp router -id 1.1.1.1
address - family l2vpn evpn
!
neighbor 1.1.1.4
remote -as 1
update - source Loopback0
!
address - family l2vpn evpn
```

4

```
extcommunity -set opaque GREEN
# color green identifies low - delay
30
end -set
!
route - policy evpn_vpws_policy
if rd in (1.1.1.4:9) then
set extcommunity color GREEN
endif
end - policy
!
router bgp 1
bgp router -id 1.1.1.4
address - family l2vpn evpn
!
neighbor 1.1.1.1
remote -as 1
update - source Loopback0
!
address - family l2vpn evpn
route - policy evpn_vpws_policy out
```

4

```
interface GigabitEthernet0 /0/0/0.100
l2transport
encapsulation dot1q 100
!
l2vpn
xconnect group evpn - vpws
p2p EVI9
interface GigabitEthernet0 /0/0/0.100
neighbor evpn evi 9 target 1 source 4
```

Segment Routing On Demand NextHop

On-demand SR Policy

- Configure an SR Policy template for each color for which on-demand SR Policy instantiation is desired
- An example with two color templates configured:
  - color 10 for high bandwidth (optimize IGP metric)
  - color 20 for low-delay (optimize link-delay metric)

```
segment-routing
 traffic-eng
  on-demand color 10
   dynamic
    metric type igp
  !
  on-demand color 20
   dynamic
    metric type delay
```

SR Policy template
High-BW (color 10)

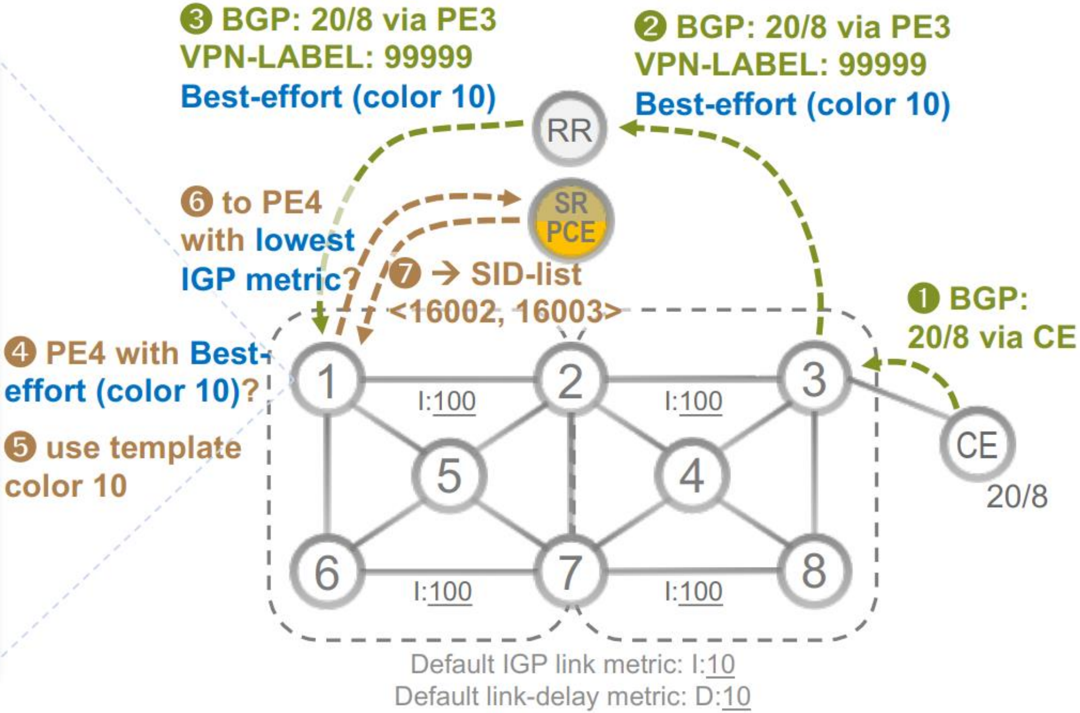
SR Policy template
low-delay (color 20)

On-Demand Nexthop reachability

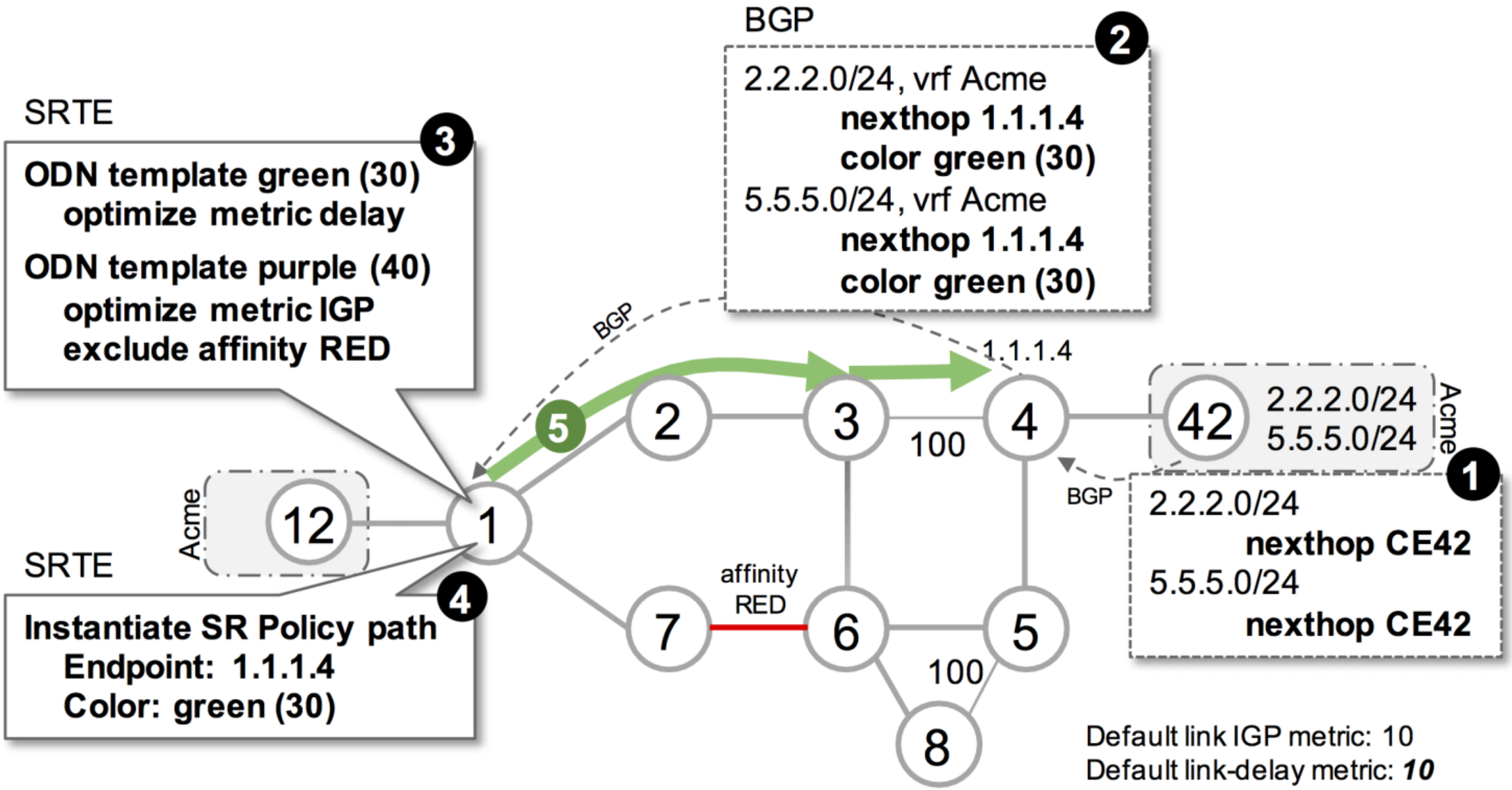
❺

```
router bgp 1
 neighbor 1.1.1.10
  address-family vpnv4 unicast
!
segment-routing
 traffic-eng
  on-demand color 10
   dynamic
    pcep
    metric
     type igp
  !
  on-demand color 20
   dynamic
    pcep
    metric
     type delay
```

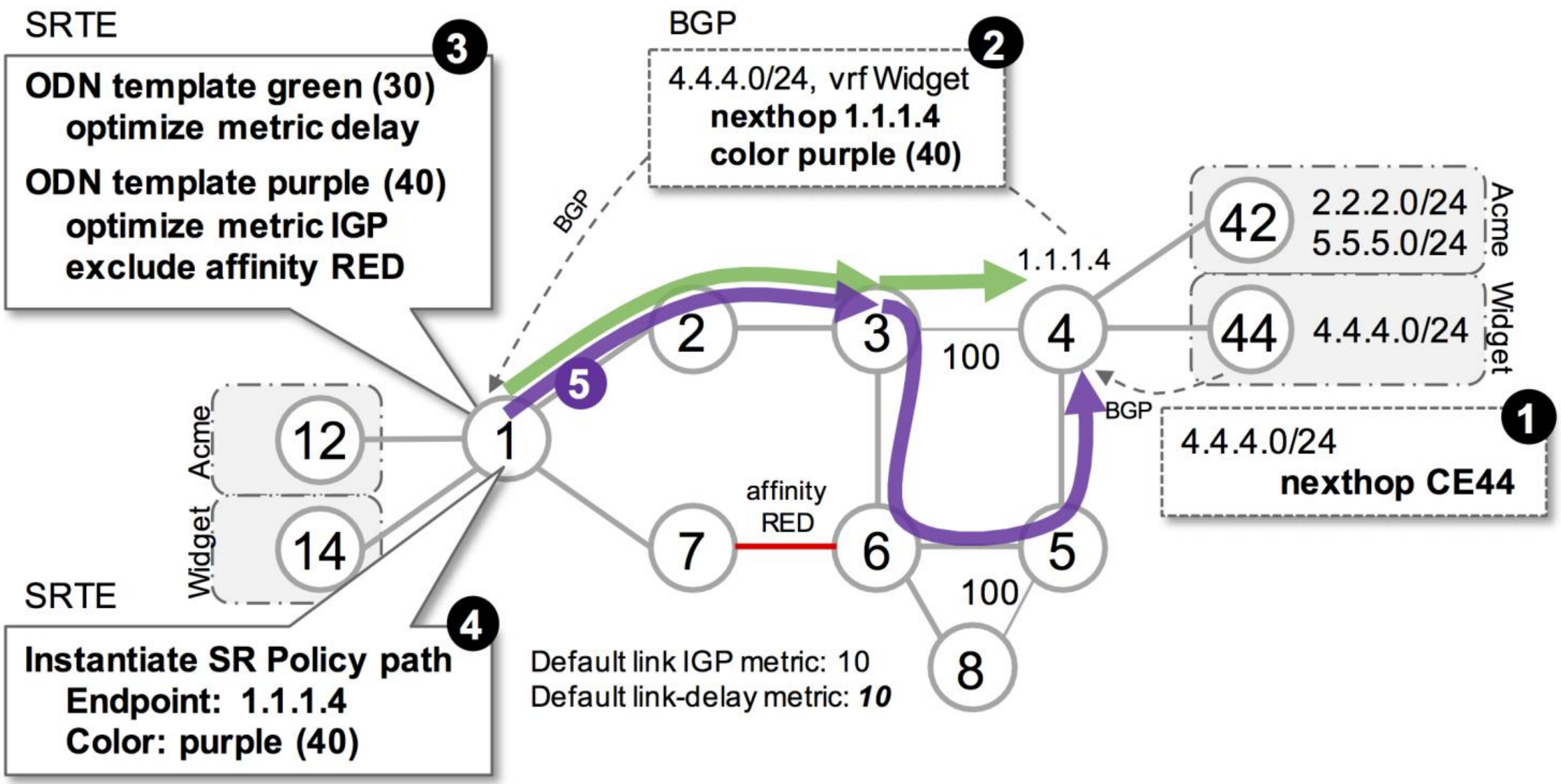
SR Policy template
Best-effort (color 10)



ODN On Demand NextHop



ODN On Demand NextHop



Intent-based ODN L3VPN

1

```
segment - routing
traffic -eng
affinity -map
        name RED bit - position 1
!
!! green ODN template
on - demand color 30
        candidate - paths
                preference 100
                dynamic
                metric
                type delay
!
!! purple ODN template
on - demand color 40
        candidate - paths
                preference 100
                constraints
                affinity
                !! RED is defined in affinity -map above
                        exclude -any
                        name RED
                dynamic
                metric
                type igp
```

4

```
extcommunity -set opaque BLUE
20
end -set
!
extcommunity -set opaque GREEN
30
end -set
!
extcommunity -set opaque PURPLE
40
end –set

route - policy VRF - COLOR
 if destination in (3.3.3.0/24) then
 set extcommunity color BLUE
 endif
 if destination in (6.6.6.0/24) then
 set extcommunity color GREEN
 endif
 pass
end - policy
!
route - policy GLOBAL - COLOR
 if destination in (8.8.8.0/24) then
 set extcommunity color PURPLE
 endif
 pass
end - policy
```

4

```
router bgp 1
 bgp router -id 1.1.1.4
 address - family ipv4 unicast
 address - family vpnv4 unicast
 !
 neighbor 1.1.1.1
 remote -as 1
 update - source Loopback0
 address - family ipv4 unicast
 route - policy GLOBAL - COLOR out
 !
 address - family vpnv4 unicast
 !
 vrf NewCo
 rd auto
 address - family ipv4 unicast
 !
 neighbor 99.4.43.43
 remote -as 2
 description to CE43
 address - family ipv4 unicast
 route - policy VRF - COLOR out
```

Automated Steering vs ODN

1

segment - routing
traffic -eng
affinity -map
name red bit - position 0
!
segment - list name BLUE_PATH
index 10 mpls label 16008
index 20 mpls label 24085
index 30 mpls label 16004
!
policy BLUE
color 20 end - point ipv4 1.1.1.4
candidate - paths
preference 100
explicit segment - list BLUE_PATH
!

1

policy GREEN
color 30 end - point ipv4 1.1.1.4
candidate - paths
preference 100
dynamic
metric
type delay
!
policy PURPLE
color 40 end - point ipv4 1.1.1.4
candidate - paths
preference 100
dynamic
metric
type igp
constraints
affinity
exclude -any
name red

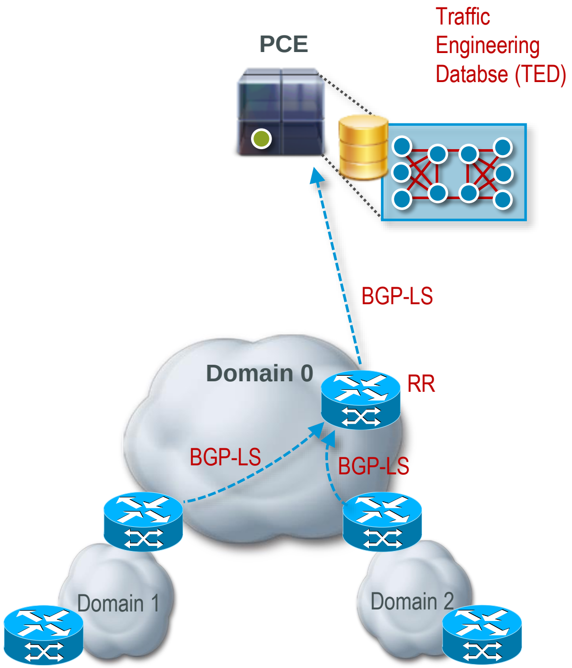
1

segment - routing
traffic -eng
affinity -map
name RED bit - position 1
!
!! green ODN template
on - demand color 30
candidate - paths
preference 100
dynamic
metric
type delay
!
!! purple ODN template
on - demand color 40
candidate - paths
preference 100
constraints
affinity
!! RED is defined in affinity -map above
exclude -any
name RED
dynamic
metric
type igp

SR PCE - BGP LS + PCEP
Use-Case Seamless MPLS - BGP LU

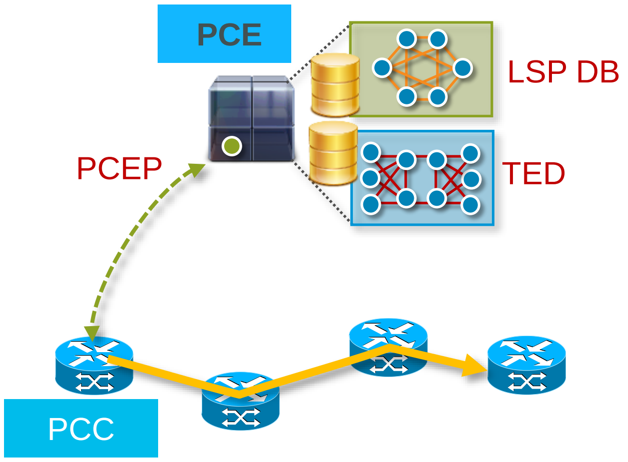
BGP-LS Overview

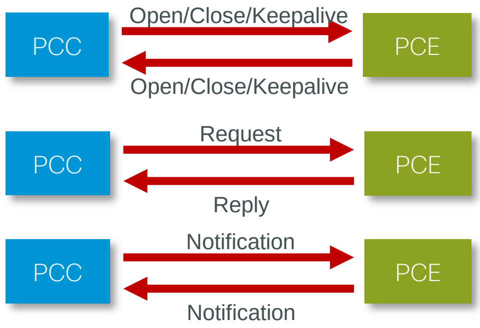
- Optimal Path Computation for Multi-area TE
- Solution is BGP, not IGP.
- BGP-LS is an address-family
  - afi=16388, safi=71
- Defined to carry IGP link-state database via BGP
  - Supports both IS-IS and OSPF
  - Delivers topology information to outside agents



PCEP Architectural Introduction

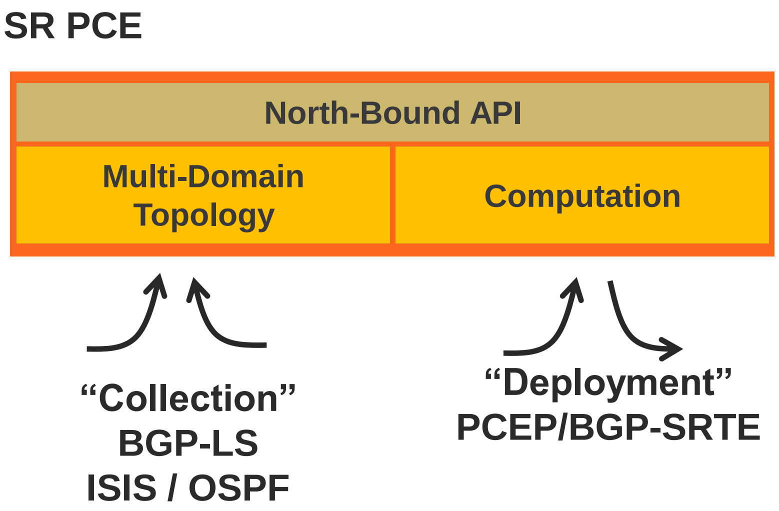
- Path computation
- Large, multi-domain and multi-layer networks
- Path computation element (**PCE**)
  - Computes network paths (topology, paths, etc.)
  - Stores TE topology database (synchronized with network)
  - May initiate path creation
  - Stateful - stores path database included resources used (synchronized with network)
- Path computation client (**PCC**)
  - May send path computation requests to PCE
  - May send path state updates to PCE
- Used between head-end router (PCC) and PCE to:
  - Request/receive path from PCE subject to constraints
  - State synchronization between PCE and router
  - Hybrid CSPF





Segment Routing - Path Computation Element (SR-PCE)

- An IOS XR-powered Stateful Path Computation Element (PCE)
- Multi-Domain topology Collection
  - Real-time reactive feed
- Computation
  - Native SR-TE algorithms backed by extensive scientific research

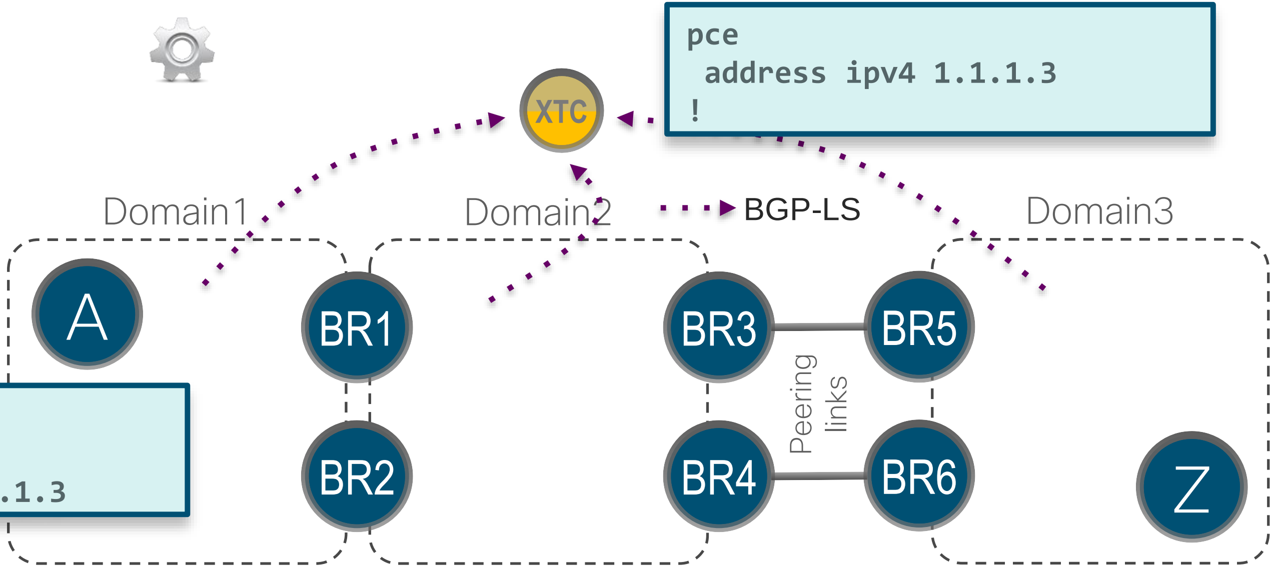


SR PCE Controller

- SR PCE is an IOS XR multi-domain stateful SR Path Computation Element (PCE)
  - Fundamentally Distributed (RR-like Deployment)
  - Supports RSVP-TE

On SR PCE:

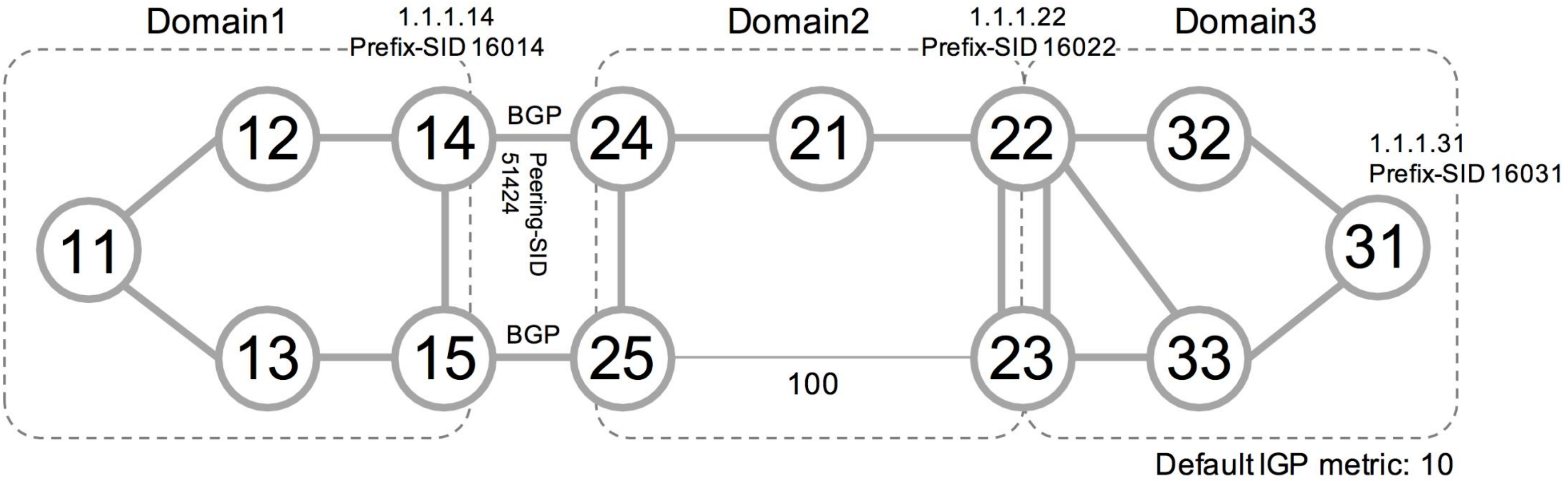
```
pce
 address ipv4 1.1.1.3
!
```



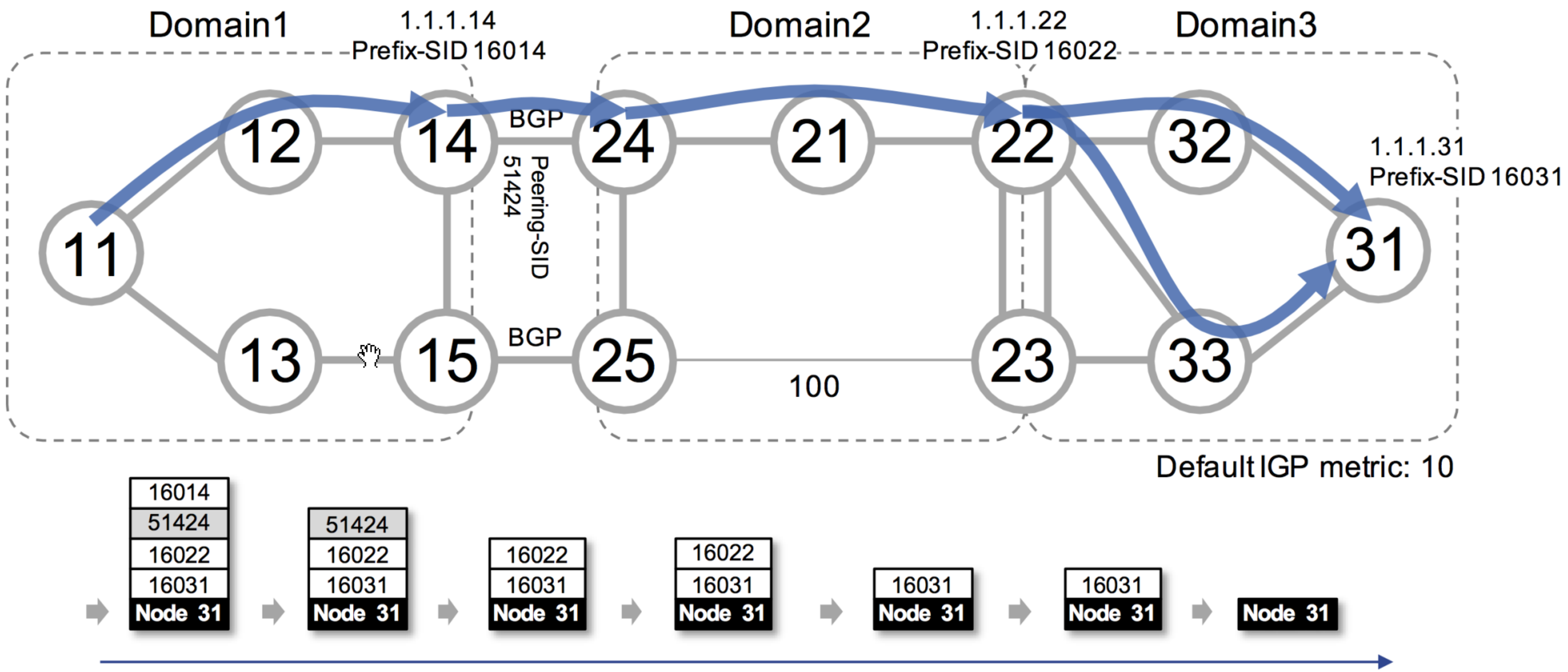
On PE:

```
pcc
  pce
   address ipv4 1.1.1.3
!
```

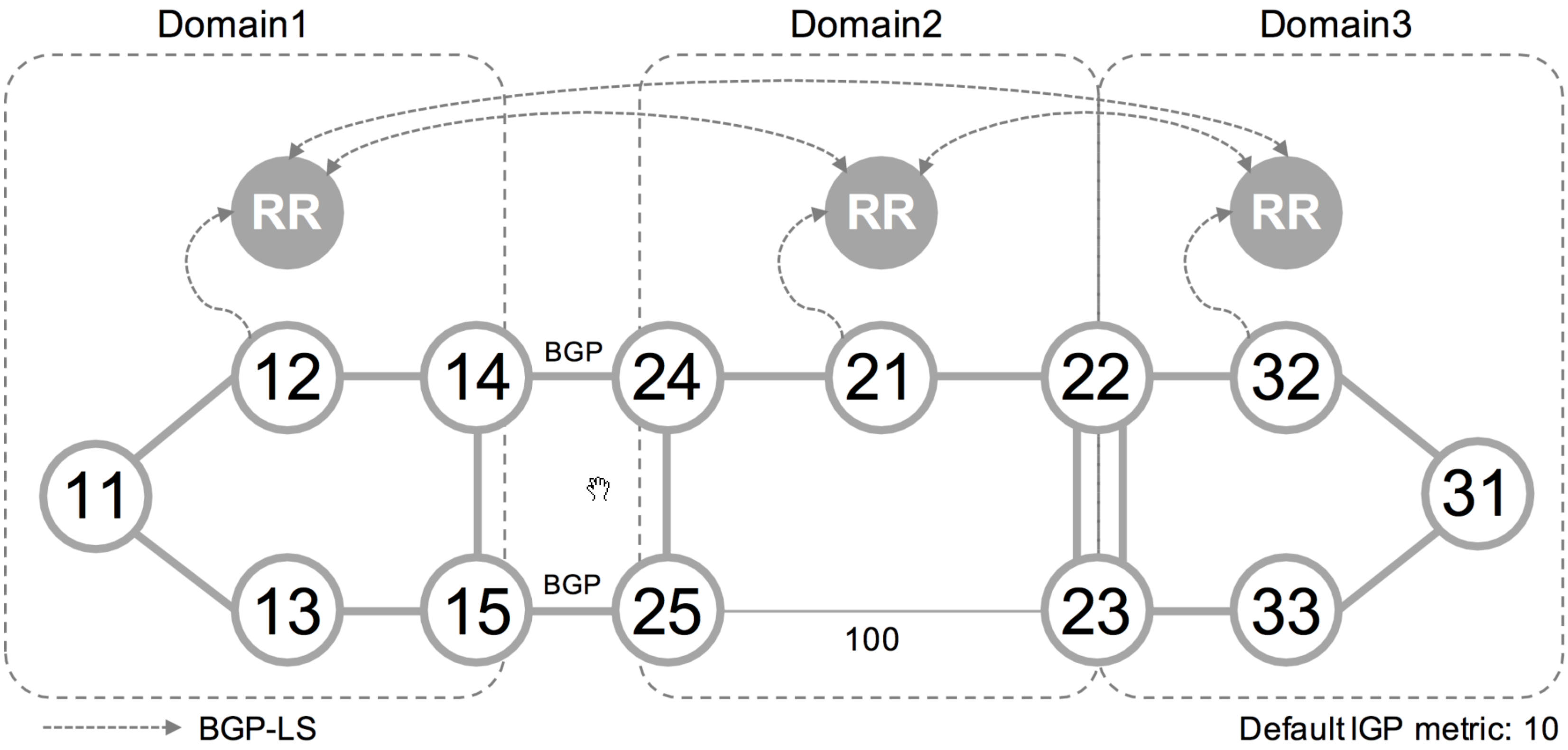
Inter-domain : Seamless MPLS



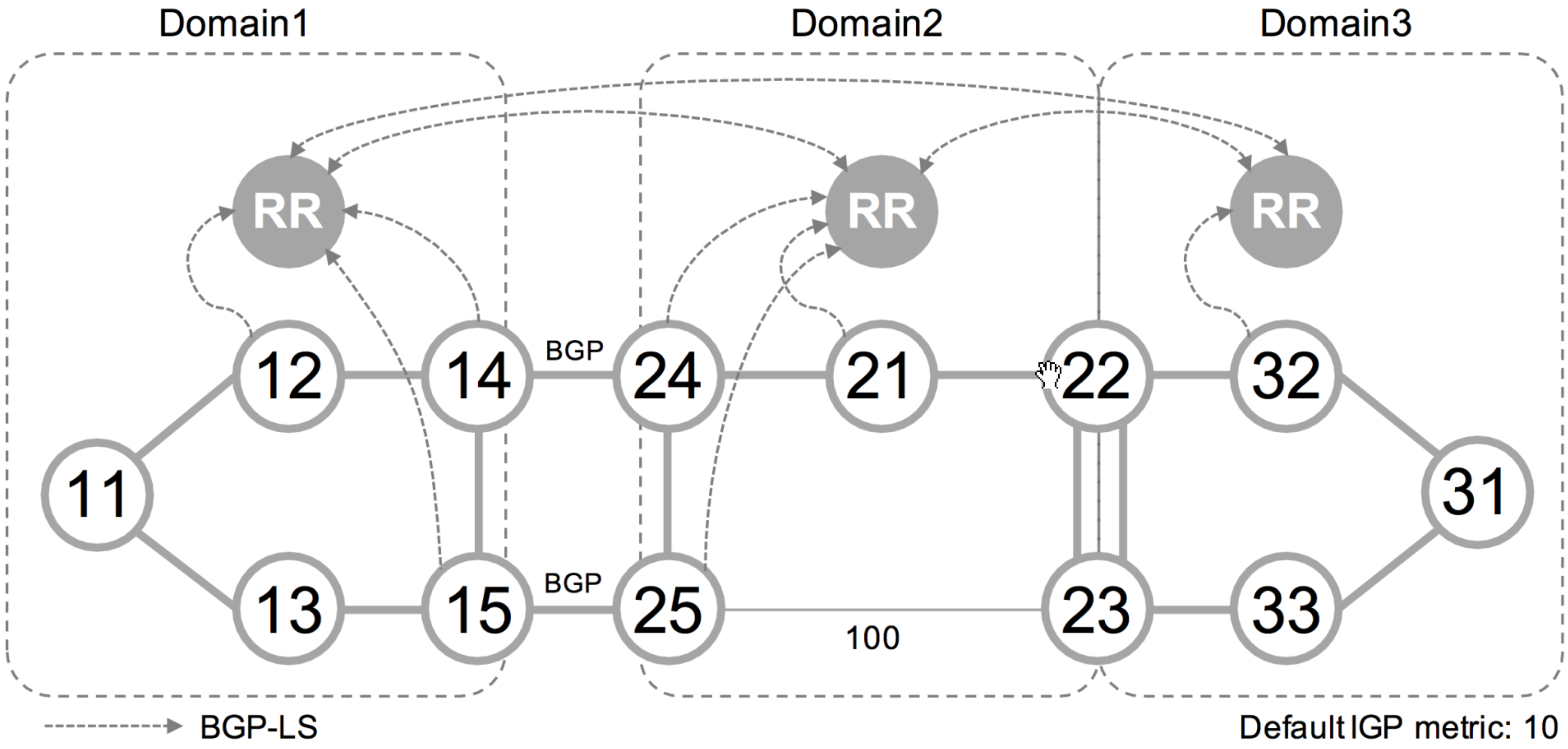
End-to-End Inter-domain Explicit Path



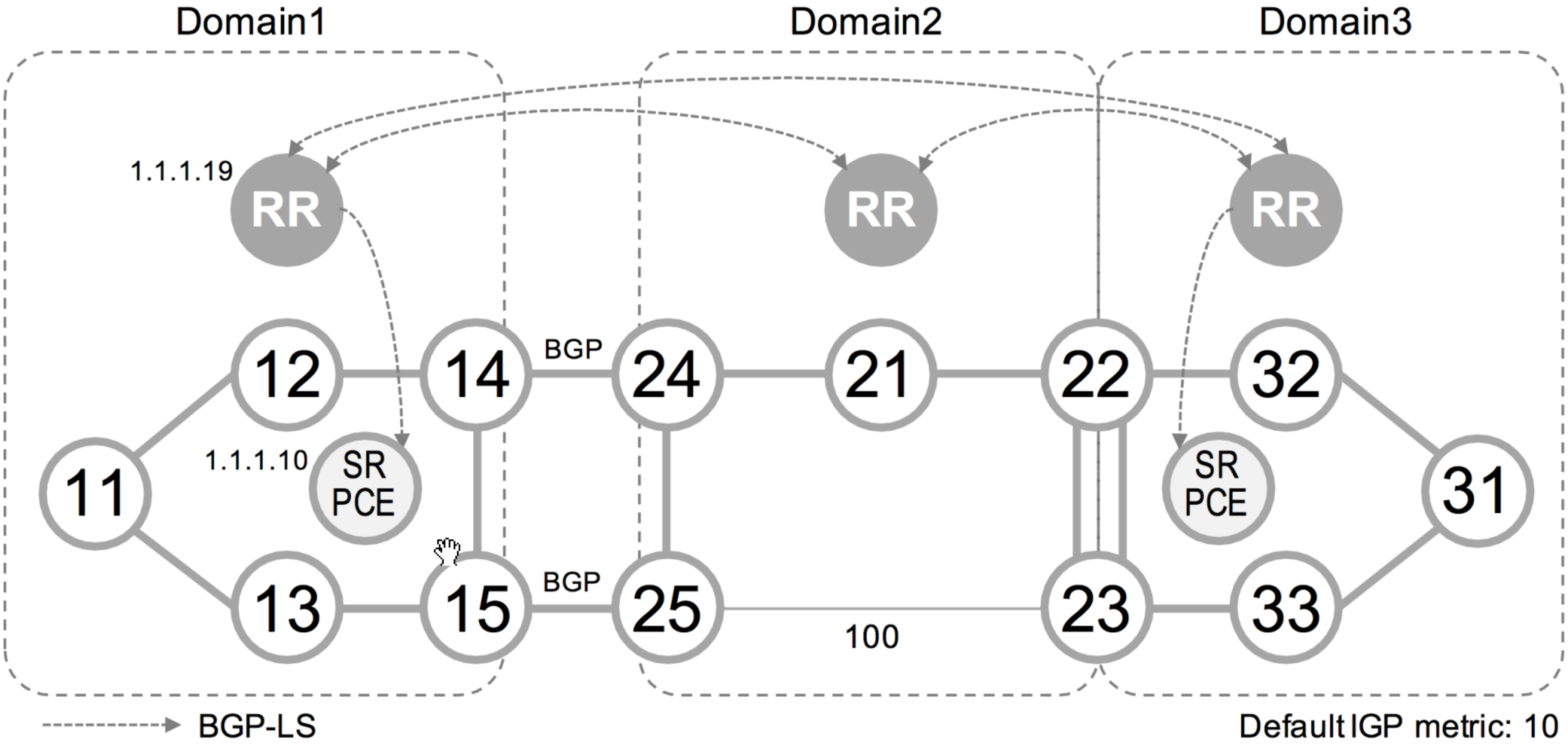
IGP LS-DB in BGP-LS to RR



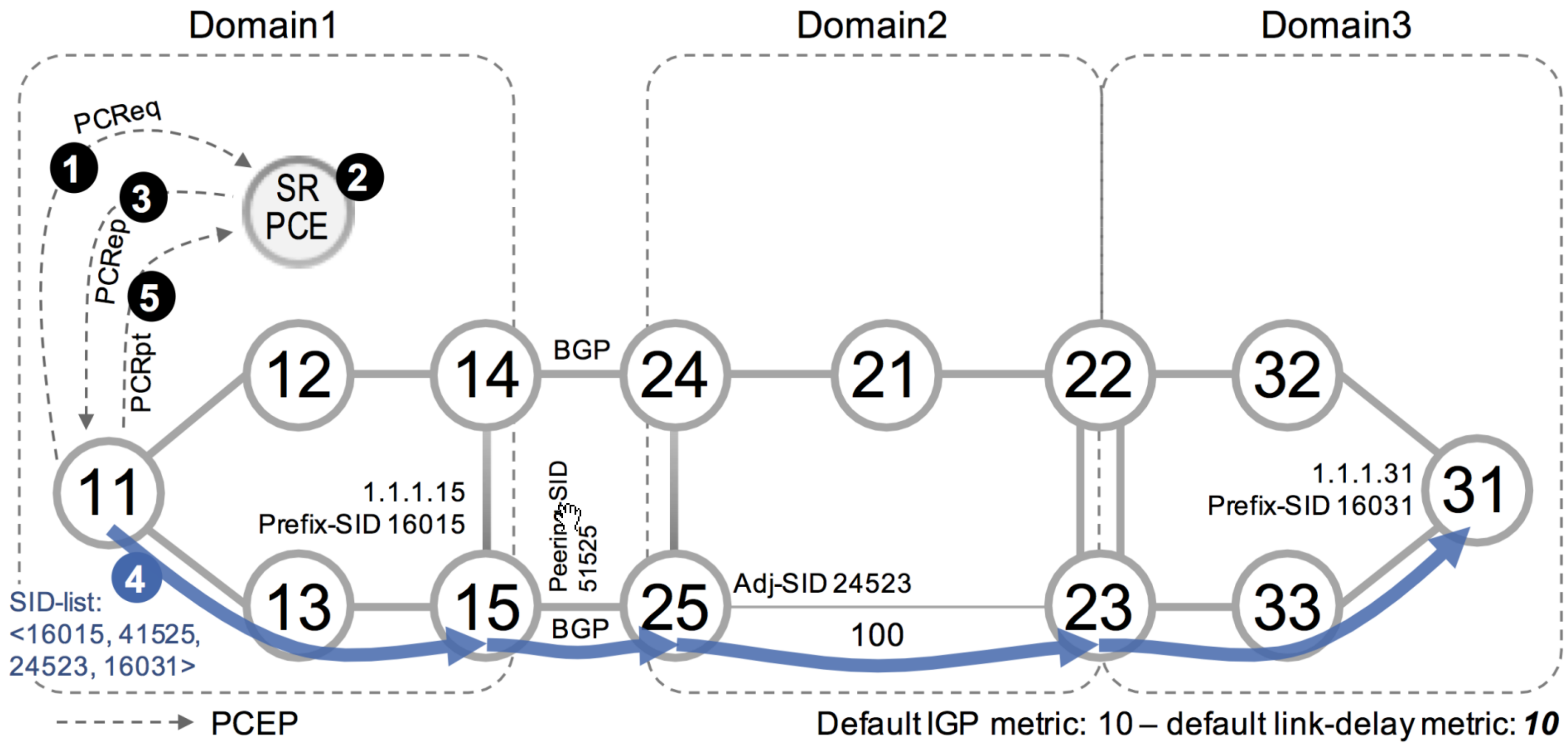
EPE in BGP-LS to RR



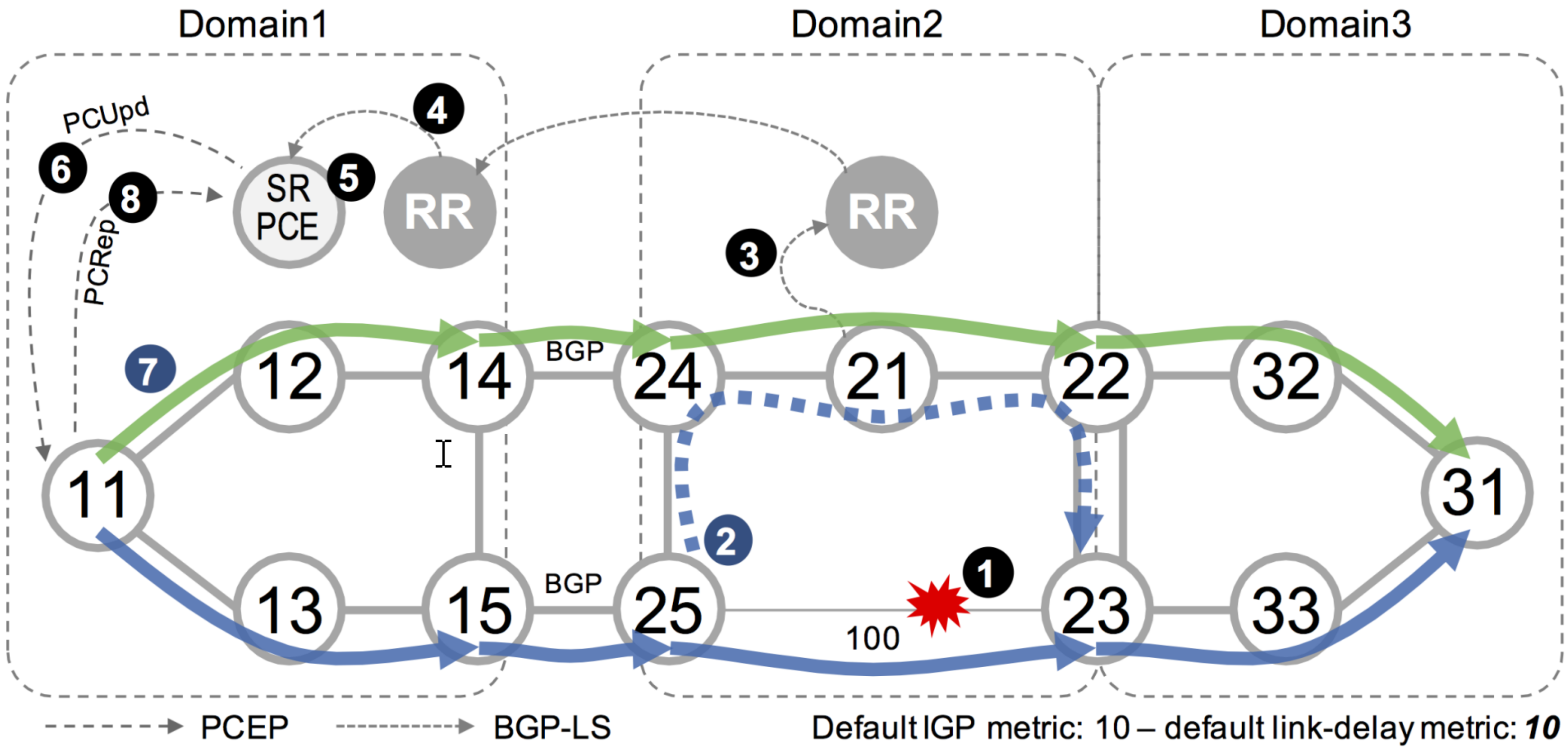
SR PCE + BGP-LS to receive Multi-domain Topology



Inter-domain Path PCEP

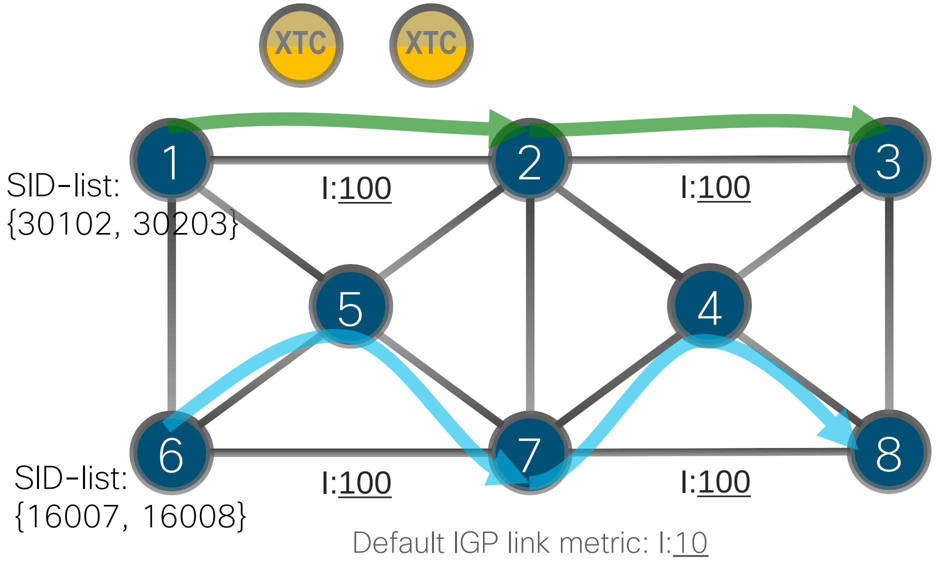


Stateful PCE updates inter-domain path after failure



Service Disjointness

Intra and inter domain



Node1

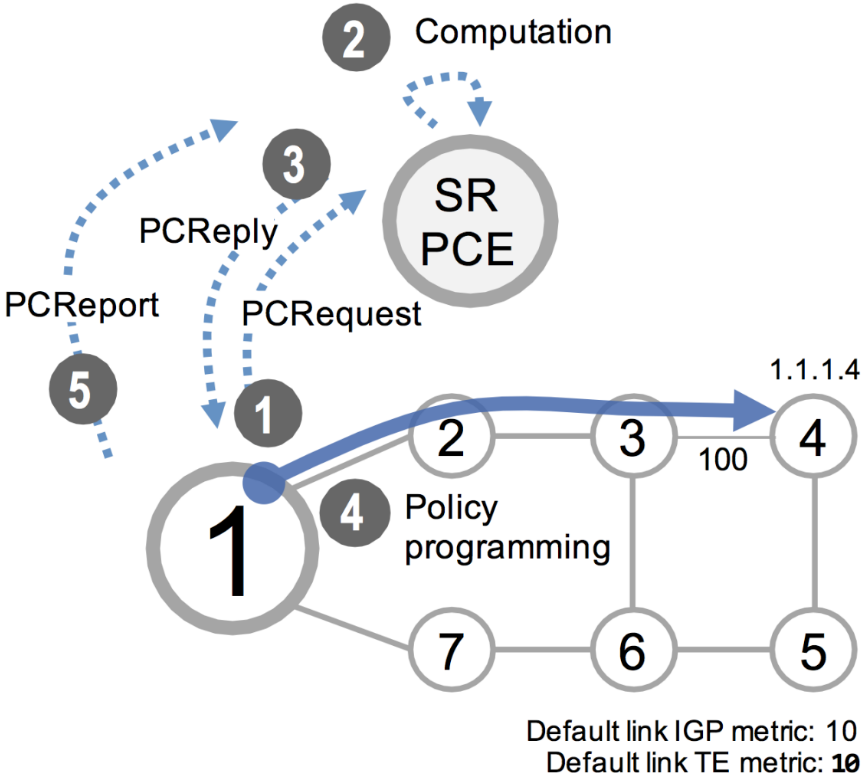
```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.3
    candidate-paths
     preference 100
      dynamic mpls pce
       metric
        type igp
       association group 1 type node
```

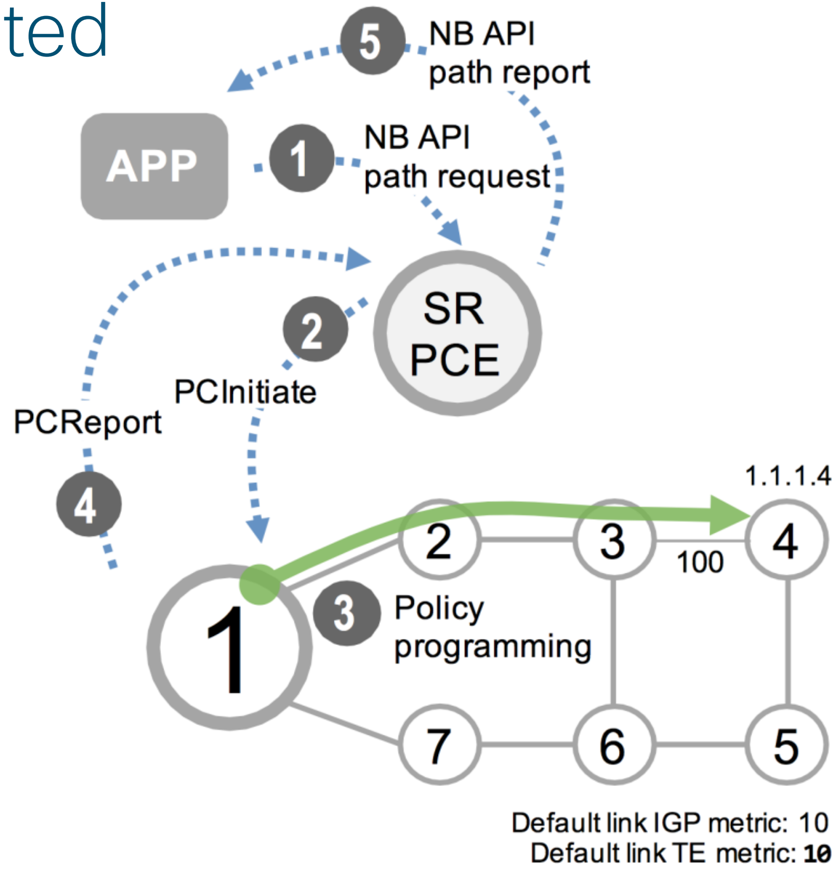
Node6

```
segment-routing
 traffic-eng
  policy POLICY2
   color 20 end-point ipv4 1.1.1.8
    candidate-paths
     preference 100
      dynamic mpls pce
       metric
        type igp
       association group 1 type node
```

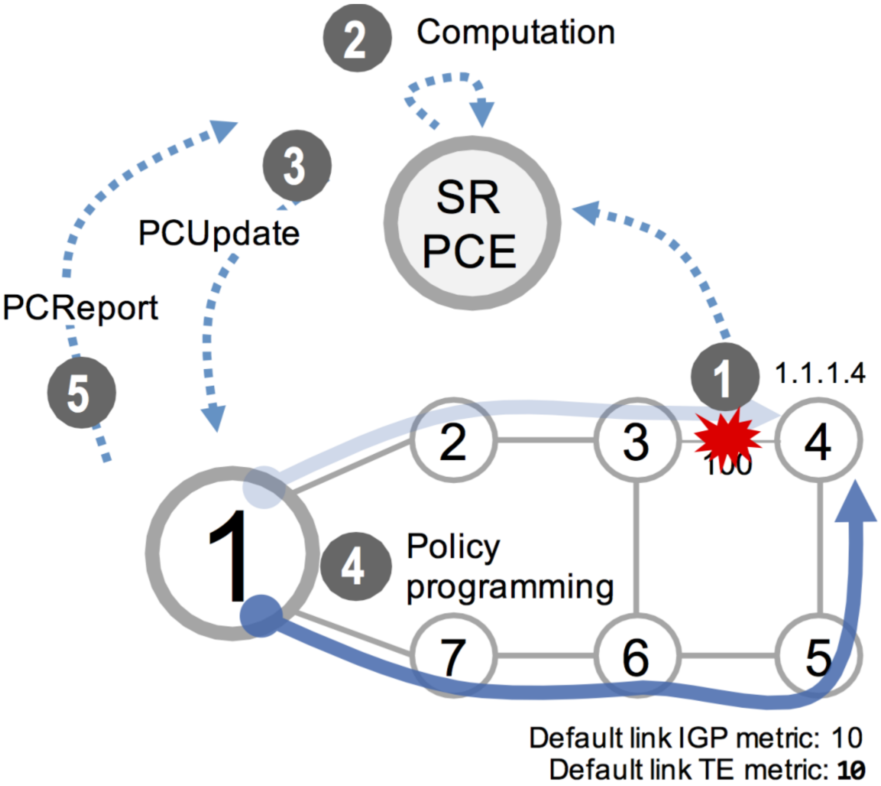
- Two dynamic paths between two different pairs of (head-end, end-point) must be disjoint from each other

PCEP SRTE : Headend initiated
SR PCE initiated

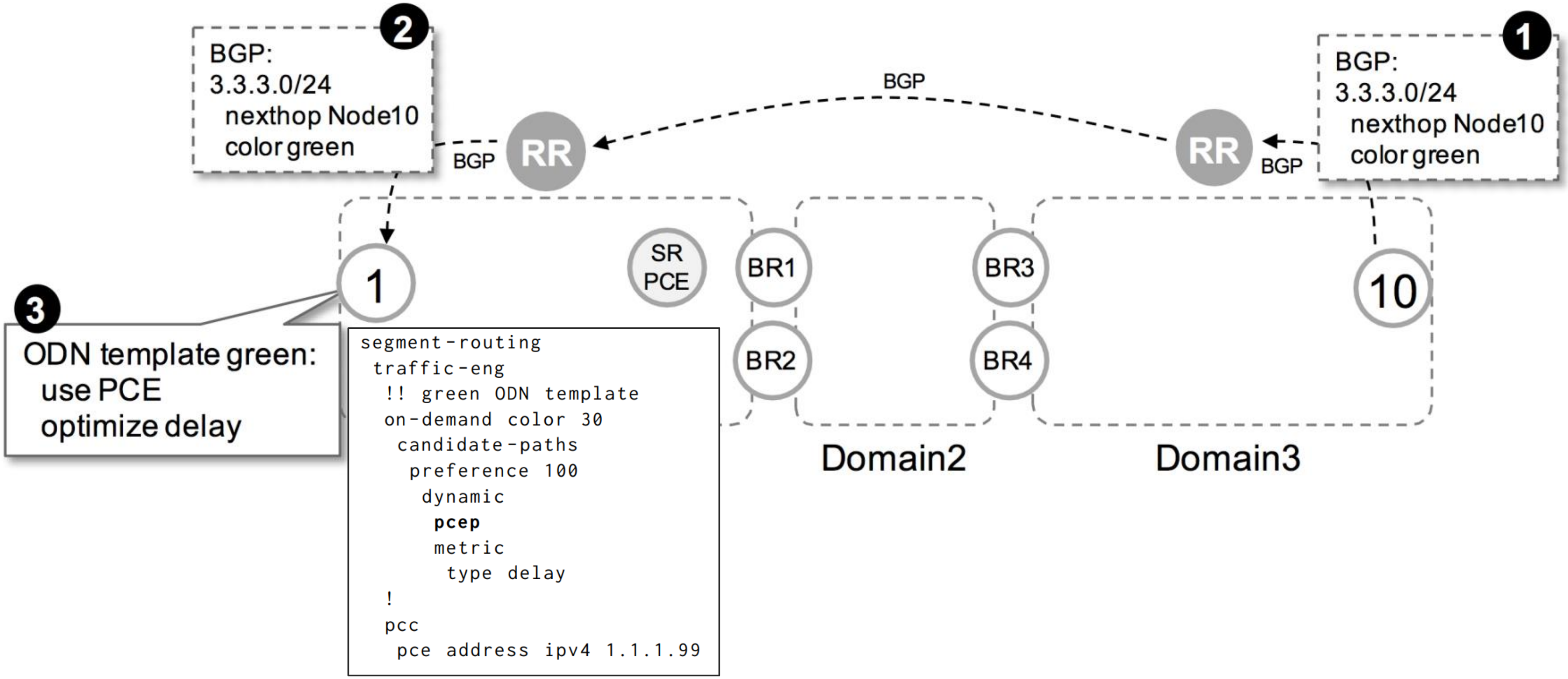




SR PCE Update



SR ODN Multi-domain Topology



Segment Routing PCE REST API

SR PCE REST

- RP/0/0/CPU0:XTC-1#sh run pce
- Sun Feb 3 02:36:53.445 UTC
- pce
- address ipv4 1.1.1.201
- rest
-   user cisco
-    password encrypted 00071A150754
-   !
-   authentication basic
- !
- logging
-   no-path
-   fallback
- !
- state-sync ipv4 1.1.1.202
- password encrypted 121A0C041104

SR PCE REST client – curl & browser

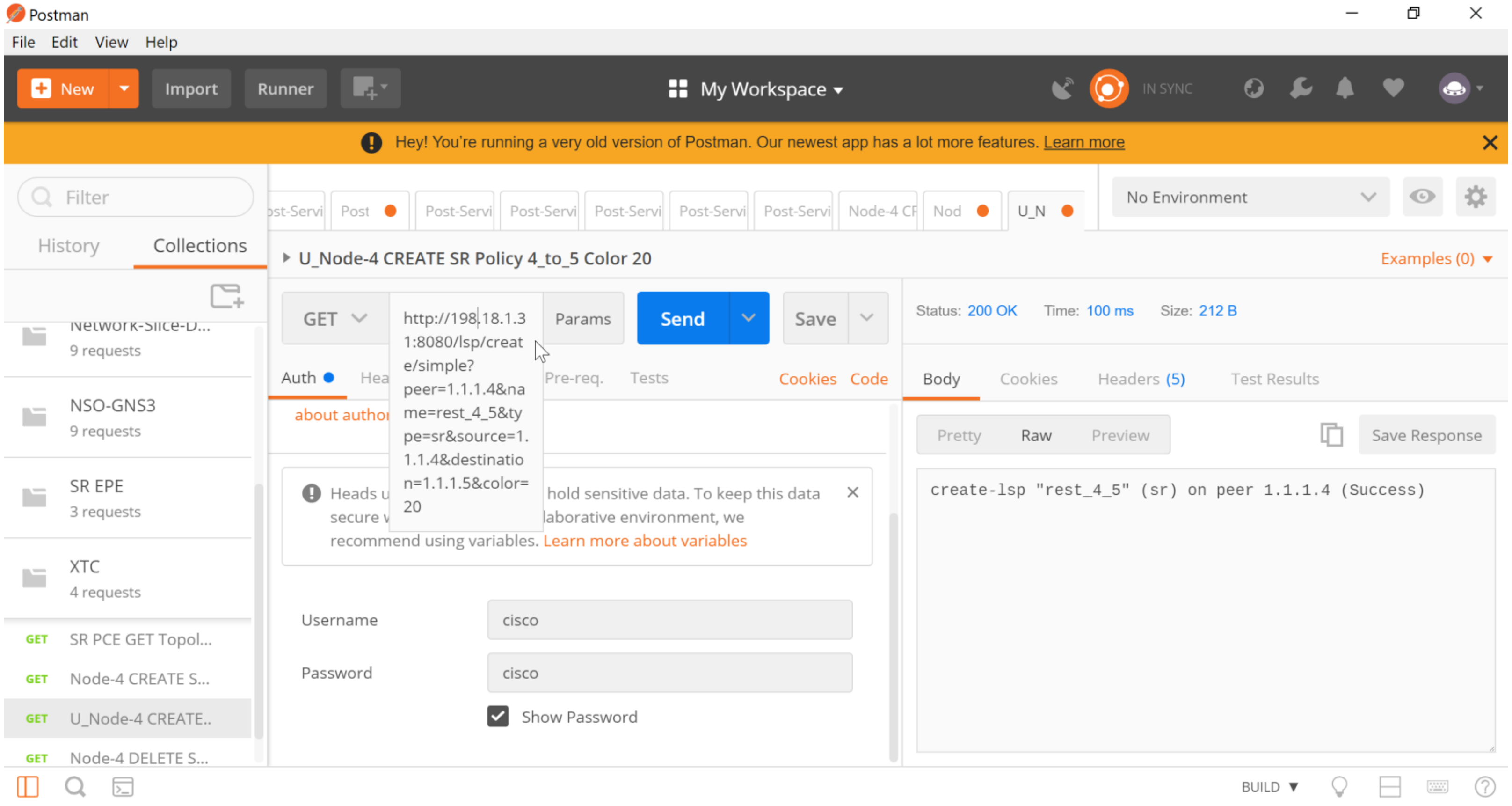
```
Not secure | 198.18.1.31:8080/lsp/delete/simple?peer=1.1.1.4&name=rest_4_5
```

```
delete-lsp "rest_4_5" (SR) on peer 1.1.1.4 (Success)
```

```
htjun@HTJUN-L90X6 MINGW64 ~
$ curl -u cisco:cisco --raw -vN "http://198.18.1.31:8080/lsp/create/simple?peer=1.1.1.5&name=rest_5_4&type=sr&source=1.1.1.5&destination=1.1.1.4&color=10&hop1-node-sid=16001&hop1-ip-local=1.1.1.1&hop2-node-sid=16002&hop2-ip-local=1.1.1.2&hop3-node-sid=16004&hop3-ip-local=1.1.1.4"
*   Trying 198.18.1.31...
* TCP_NODELAY set
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     035
create-lsp "rest_5_4" (sr) on peer 1.1.1.5 (Success)
0

create-lsp "rest_5_4" (sr) on peer 1.1.1.5 (Success) * Connected to 198.18.1.31 (198.18.1.31) port 8080 (#0)
* Server auth using Basic with user 'cisco'
> GET /lsp/create/simple?peer=1.1.1.5&name=rest_5_4&type=sr&source=1.1.1.5&destination=1.1.1.4&color=10&hop1-node-sid=16001&hop1-ip-local=1.1.1.1&hop2-node-sid=16002&hop2-ip-local=1.1.1.2&hop3-node-sid=16004&hop3-ip-local=1.1.1.4 HTTP/1.1
> Host: 198.18.1.31:8080
> Authorization: Basic Y2lzY286Y2lzY28=
> User-Agent: curl/7.57.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Cache-Control: no-cache, no-store
< Content-Type: text/json; charset=utf-8
< Expires: -1
< Transfer-Encoding: chunked
< Connection: keep-alive
<
{ [64 bytes data]
100    53    0    53    0     0     53      0 --:--:-- --:--:-- --:--:--   679
* Connection #0 to host 198.18.1.31 left intact
```

SR PCE REST client - postman



SR PCE REST create_lsp parameter -1

/lsp/create/simple – rest url
peer=1.1.1.2 – pcc (tunnel headend router) address

name=rest_2_4 – tunnel name, max name length is restricted to 128 characters. Name cannot contains spaces or special characters (allowed are '-', '_', '(', ')')

type=sr – setup type. If not specified SR is assumed. Allowed values are sr or rsvp

source=1.1.1.2 – tunnel source address. If not specified "peer" address will be used instead

allow-xtc-reoptimization=0 - REST client prevents XTC to update lsp with new path e.g. in case of link down and REST client has to update path itself. Default value is 1, in that case XTC is allowed to modify lsp.

destination=1.1.1.4 – tunnel destination address

color=10 – SR policy color, mandatory when creating SR policy (not needed for rsvp lsps)

preference=10 – SR policy path preference, default value is 100, (not needed for rsvp lsps)

SR PCE REST create_lsp parameter -2

metric-igp=10 - using igp metric as path computation constraint

metric-te=10 - using te metric as path computation constraint

metric-latency=10 - using latency metric as path computation constraint

hop1-adj-sid=24001&hop1-ip-local=99.1.2.2&hop1-ip-remote=99.1.2.1 – path's first hop defined by adj-sid

hop2-adj-sid=28099&hop2-ip-local=99.1.3.1&hop2-ip-remote=99.1.3.3 – path's second hop defined by adj-sid

hop3-adj-sid=24003&hop3-ip-local=99.3.4.3&hop3-ip-remote=99.3.4.4 – path's third hop defined by adj-sid

hopX-node-sid=30001&hopX-ip-local=1.1.1.1

Example: (no_space)

```
curl --raw -vN "http://<xtc-ip-addr>:8080/lsp/create/simple?peer=1.1.1.2&name=rest_2_4&type=sr
&source=1.1.1.2&destination=1.1.1.4&color=10
&hop1-adj-sid=24001&hop1-ip-local=99.1.2.2&hop1-ip-remote=99.1.2.1
&hop2-adj-sid=28099&hop2-ip-local=99.1.3.1&hop2-ip-remote=99.1.3.3
&hop3-adj-sid=24003&hop3-ip-local=99.3.4.3&hop3-ip-remote=99.3.4.4"
```

SR PCE REST update_lsp parameter

/lsp/update/simple – rest url

peer=1.1.1.2 – pcc (tunnel headend router) address

name=rest_2_4 – tunnel name, max name length is restricted to 128 characters. Name cannot contains spaces or special characters (allowed are '-', '_', '(', ')')

type=sr – setup type. If not specified SR is assumed

hop1-adj-sid=24001&hop1-ip-local=99.1.2.2&hop1-ip-remote=99.1.2.1 – path's first hop defined by adj-sid

hop2-adj-sid=28099&hop2-ip-local=99.1.3.1&hop2-ip-remote=99.1.3.3 – path's second hop defined by adj-sid

hop3-adj-sid=24003&hop3-ip-local=99.3.4.3&hop3-ip-remote=99.3.4.4 – path's third hop defined by adj-sid

Example: (no_space)
curl --raw -vN "http://<xtc-ip-addr>:8080/lsp/update/simple?peer=1.1.1.2&name=rest_2_4
&hop1-adj-sid=24001&hop1-ip-local=99.1.2.2&hop1-ip-remote=99.1.2.1
&hop2-adj-sid=28099&hop2-ip-local=99.1.3.1&hop2-ip-remote=99.1.3.3
&hop3-adj-sid=24003&hop3-ip-local=99.3.4.3&hop3-ip-remote=99.3.4.4"

SR PCE REST delete_lsp parameter

/lsp/delete/simple – rest url

peer=1.1.1.2 – pcc (tunnel headend router) address

name=rest_2_4 – tunnel name, max name length is restricted to 128 characters. Name cannot contains spaces or special characters (allowed are '-', '_', '(', ')')

Example:
curl --raw -vN "http://<xtc-ip-addr>:8080/lsp/delete/simple?peer=1.1.1.2&name=rest_2_4"

Steering Traffic over SR Policy

Steering Traffic over SR Policy

- Automated Steering
- On Demand Nexthop (ODN)
- Static route via local SR Policy
- SR Policy Autoroute
- VPWS, VPLS and PWHE Preferred-path over SR Policy
- EVPN-VPWS Preferred-path over SR Policy

Steering Traffic over SR Policy
Static Route via Local SR Policy

```
router static
 address-family ipv4 unicast
  202.202.1.0/24 sr-policy POLICY-31

segment-routing
 traffic-eng
  policy POLICY-31
   color 3001 end-point ipv4 100.100.4.20  < ----- local SR Policy has (color:3001, EP:100.100.4.20)
   candidate-paths
    preference 100
     explicit segment-list Path-31
```

Steering Traffic over SR Policy SR Policy Autoroute

```
segment-routing
 traffic-eng
  policy POLICY-31
   color 31 end-point ipv4 100.100.4.20
                 < ----- L3VPN or EVPN next-hop
   autoroute
    include all
   !
   candidate-paths
    preference 100
     explicit segment-list Path-31
     !
```

```
segment-routing
 traffic-eng
  policy POLICY-31
   color 31 end-point ipv4 100.100.4.20
autoroute
    include ipv4 98.1.1.0/24
    include ipv4 99.1.1.0/24
              < ----- IGP subnets behind 100.100.4.20 router   !
   candidate-paths
    preference 100
     explicit segment-list Path-31
     !
```

Steering Traffic over SR Policy
VPWS, VPLS and PWHE Preferred-path over SR Policy

```
segment-routing
 traffic-eng
  policy POLICY-100
   color 100 end-point ipv4 2.2.2.2
   candidate-paths
    preference 100
     explicit segment-list PE1-P1-PE2
     !
  policy POLICY-200
   color 200 end-point ipv4 2.2.2.2
   candidate-paths
    preference 100
     explicit segment-list PE1-P2-PE2
     !
  policy POLICY-300
   color 300 end-point ipv4 2.2.2.2
   candidate-paths
    preference 100
     explicit segment-list PE1-P3-PE2
     !
```

```
l2vpn
 pw-class PW-100
  encapsulation mphls
   preferred-path sr-te policy POLICY-100
  !
 pw-class PW-200
  encapsulation mpls
   preferred-path sr-te policy POLICY-200
  !
 pw-class PW-300
  encapsulation mpls
   preferred-path sr-te policy POLICY-300
  !
```

Steering Traffic over SR Policy
VPWS, VPLS and PWHE Preferred-path over SR Policy

```
xconnect group PWHE-100  < ----- PWHE preferred-path over SR Policy example
  p2p PWHE-100
   interface PW-Ether1
   neighbor ipv4 2.2.2.2 pw-id 1003
    pw-class PW-300
   !
 xconnect group VPWS-100  < ----- VPWS preferred-path over SR Policy exmaple
  p2p VPWS-100
   interface TenGigE0/9/0/1.1001
   neighbor ipv4 2.2.2.2 pw-id 1001
    pw-class PW-100
   !
 bridge group BD-100      < ----- VPLS preferred-path over SR Policy exmaple
  bridge-domain BG-100
   interface TenGigE0/9/0/1.1002
   !
   neighbor 2.2.2.2 pw-id 1002
    pw-class PW-200
   !
```

Steering Traffic over SR Policy
EVPN-VPWS Preferred-path over SR Policy

```
segment-routing
 traffic-eng
  policy POLICY-400
   color 400 end-point ipv4 2.2.2.2
   candidate-paths
    preference 100
     explicit segment-list PE1-P4-PE2
     !
l2vpn
 pw-class PW-400
  encapsulation mpls
   preferred-path sr-te policy POLICY-400
  !
 xconnect group EVPN-VPWS-100
  p2p EVPN-VPWS-100
   interface TenGigE0/9/0/1.1004
   neighbor evpn evi 10 target 5004 source 1004 < ----- EVPN-VPWS preferred-path over SR Policy exmaple
    pw-class PW-400
   !
  !
```