You make **possible**

Segment Routing: Technology deep-dive and advanced use cases

Clarence Filsfils

BRKRST-3122

cisco Live!

Barcelona | January 27-31, 2020

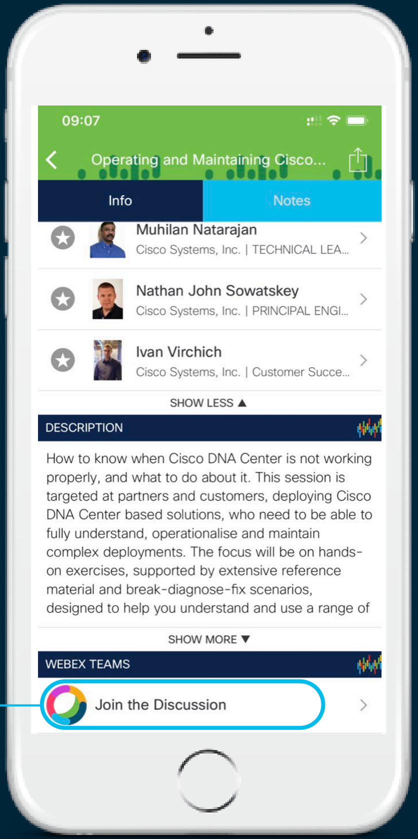
Cisco Webex Teams

Questions?

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

How

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



09:07

Operating and Maintaining Cisco...

Info | Notes

Muhilan Natarajan
Cisco Systems, Inc. | TECHNICAL LEA...

Nathan John Sowatskey
Cisco Systems, Inc. | PRINCIPAL ENGI...

Ivan Virchich
Cisco Systems, Inc. | Customer Succe...

SHOW LESS ▲

DESCRIPTION

How to know when Cisco DNA Center is not working properly, and what to do about it. This session is targeted at partners and customers, deploying Cisco DNA Center based solutions, who need to be able to fully understand, operationalise and maintain complex deployments. The focus will be on hands-on exercises, supported by extensive reference material and break-diagnose-fix scenarios, designed to help you understand and use a range of

SHOW MORE ▼

WEBEX TEAMS

Join the Discussion

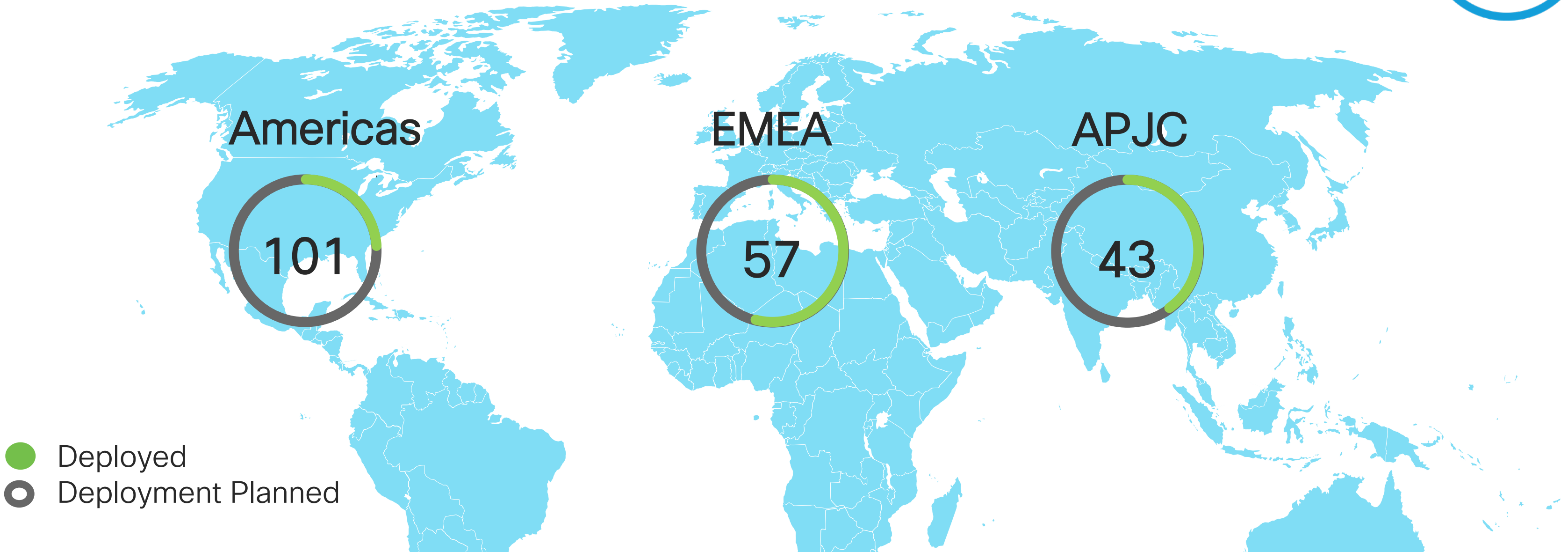
cisco Live!

SR-MPLS

SR MPLS
Industry Update

From thought to deployment leadership

SR
Segment Routing



Bell SoftBank vodafone Alibaba.com Walmart colt South Slope orange Microsoft Aussie Broadband

China unicom中国联通 SHENTEL Always connected to you Telefonica airtel kt eolo etisalat EQUINIX Google TELUS

cisco Live!

SR is IETF Proposed Standard

- RFC 8402 “SR Architecture” – Proposed Standard
  - Defines SR-MPLS with MPLS dataplane and Label SID’s
  - Defines SRv6 with SRH and SRv6 SID’s

SR-MPLS

- RFC Proposed Standard for most (21) documents
  - MPLS data plane
  - SR/LDP interworking
  - ISIS, OSPF, BGP, and PCEP extensions
  - OAM
  - PM
- 8 Informational RFCs
  - Use-cases

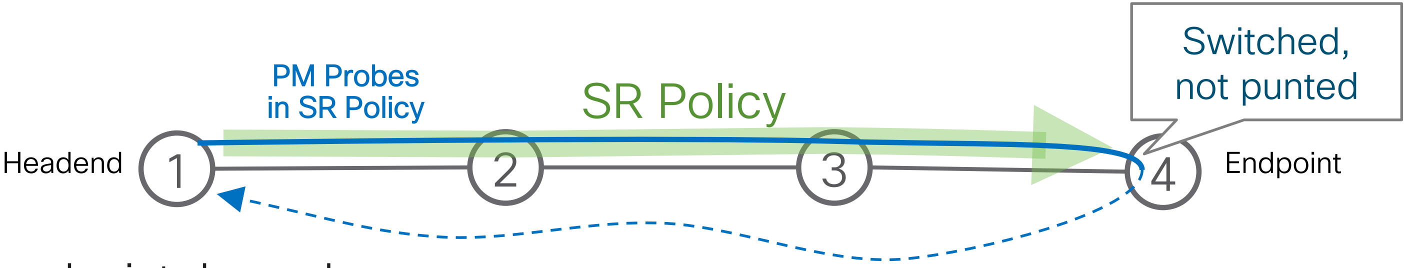
Innovations Highlights

Innovations we shared at Cisco Live 2019:

• ISIS Flex Algo	Shipping
• OSPF Flex Algo	Shipping
• SR-PCE / SRTE: Anycast-SID aware path computation	Shipping
• MPLS-PM: per-link delay measurement	Shipping
• MPLS-PM: end-to-end SR Policy delay measurement	Shipping
• SR Data Plane Monitoring (SR-DPM)	Shipping

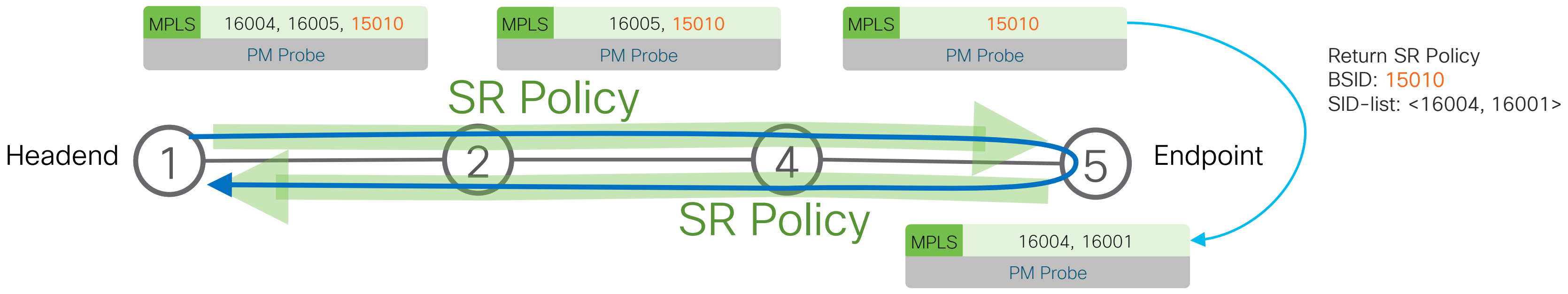
SR Policy Liveness Monitoring

SR Policy Liveness monitoring



- No endpoint dependency
  - Probes sent through endpoint and back to headend
  - Simpler deployment, higher scale
- Hardware offload provides 3.3ms tx interval
  - Liveness failure after loss of 3 consecutive probes
  - Failure detection in 10ms + RTT
- Can tear down active candidate path upon liveness failure

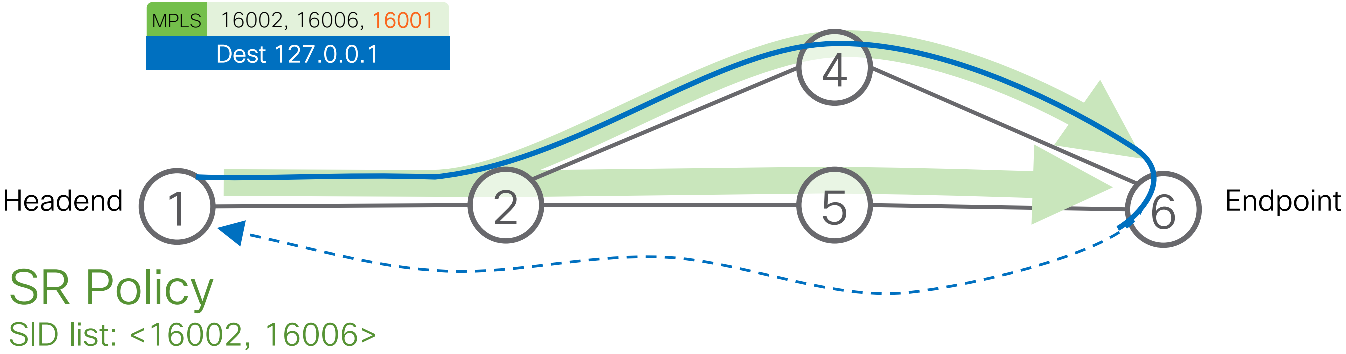
Variant – Bidirectional SR Policy



Return SR Policy
BSID: 15010
SID-list: <16004, 16001>

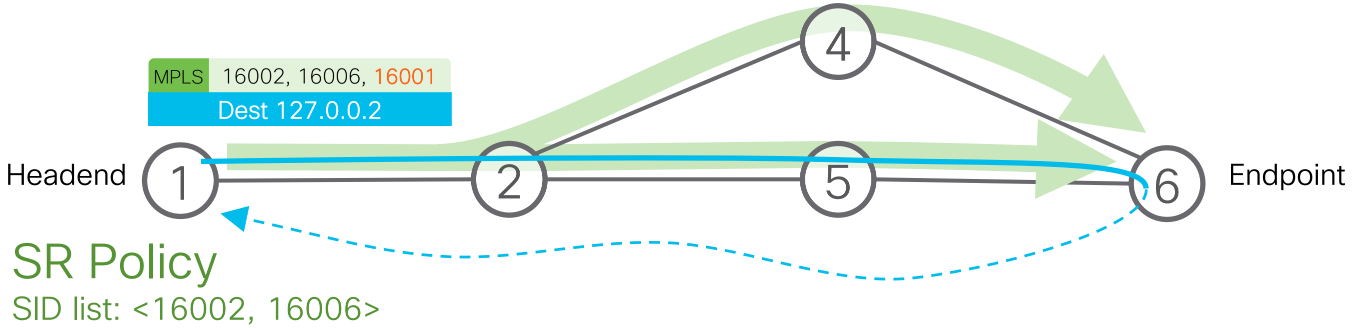
- BSID of a return SR Policy can be encoded in the probe label stack
- Prevent false negatives

Variant – ECMP sweeping



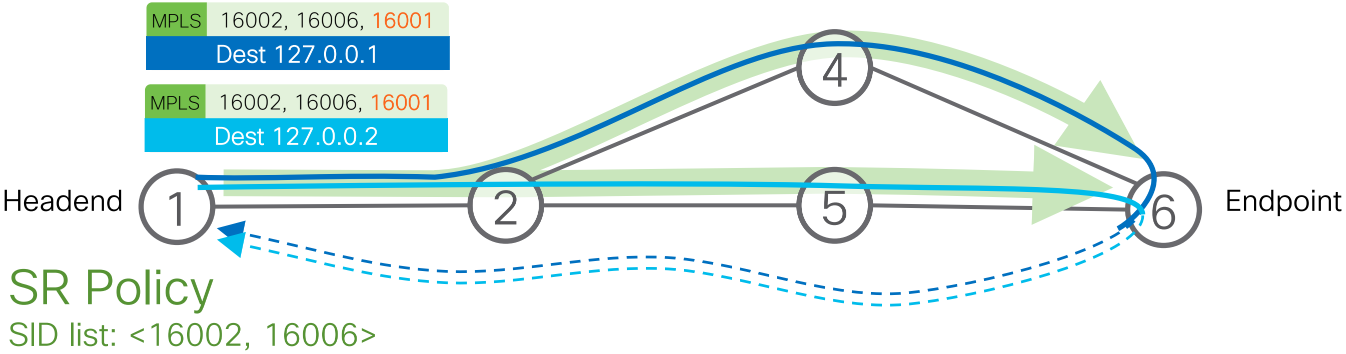
- Use probes with different destination addresses to hash on different paths
- Probabilistic coverage of SR Policy ECMP paths

Variant – ECMP sweeping



- Use probes with different destination addresses to hash on different paths
- Probabilistic coverage of SR Policy ECMP paths

Variant – ECMP sweeping



- Use probes with different destination addresses to hash on different paths
- Probabilistic coverage of SR Policy ECMP paths

Circuit-Style SR Policy

One SR deployment – Different service types

SR allows a single network to accommodate flows of different service types

- IP-centric services with ECMP and TI-LFA
- TDM-centric services with circuits and path protection

Components of the solutions

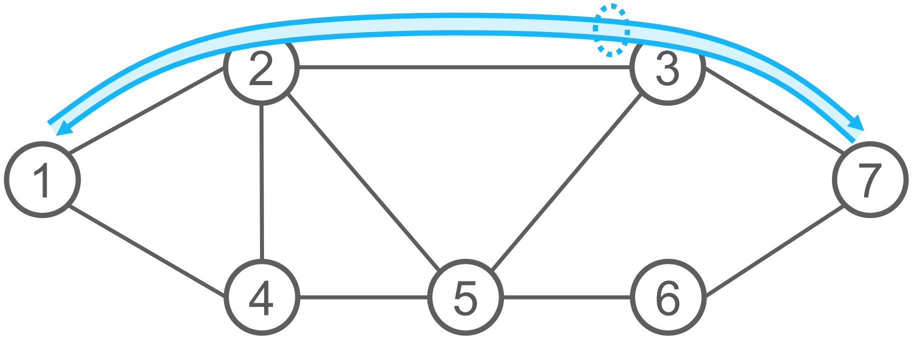
- Centralized Controller
  - computes the path
  - Encodes the path in list of Adj-SIDs
  - Bandwdith book-keeping for SLA guarantee
- QoS configuration on every link to isolate guaranteed traffic

Properties of SR Circuit-Style for TDM Services

- ✓ Co-routed bidirectional
- ✓ Persistence
- ✓ Guaranteed Latency
- ✓ End-to-end path protection
- ✓ Guaranteed Bandwidth

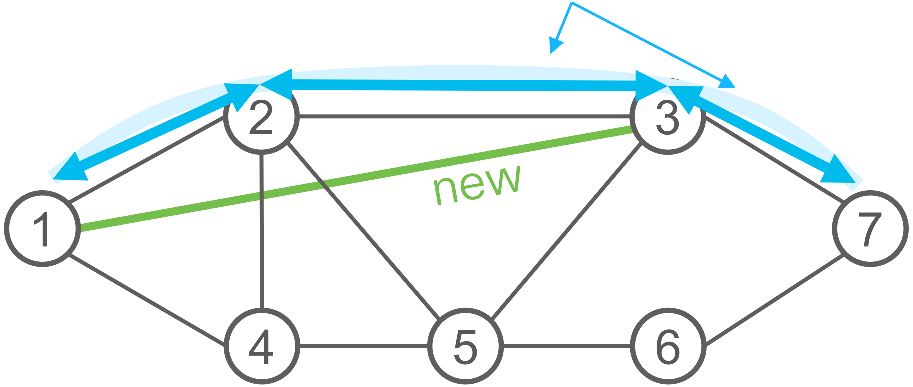
Co-routed bidirectional path

Forward and return SR Policies are routed along the same path



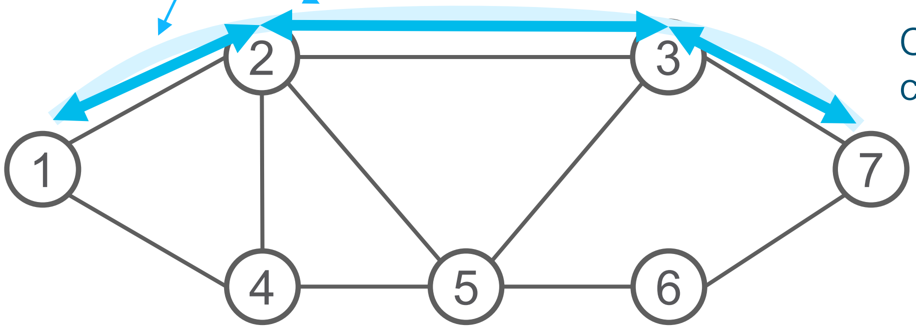
Control-plane independent persistence

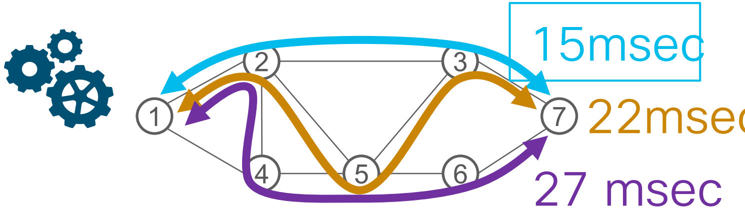
Adj-SIDs provide persistent path independent from network events, control-plane status and IP transport



Non-ECMP path with guaranteed latency

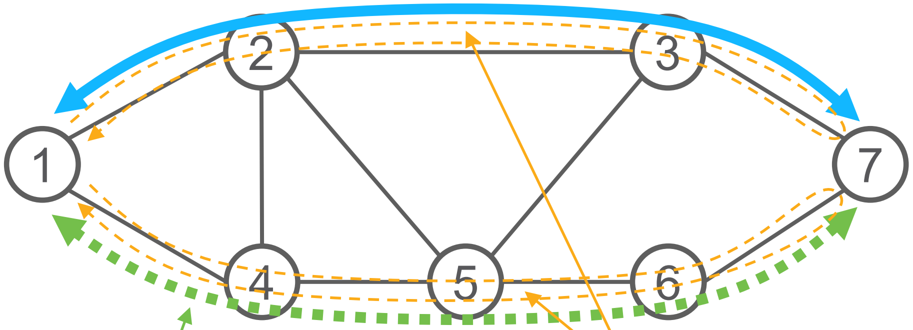
Adj-SIDs provide a non-ECMP path





Centralized controller computes the min delay path

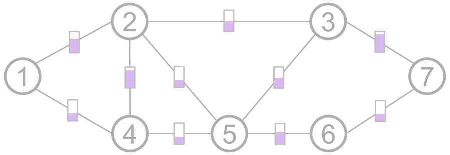
Integrity monitoring with path protection switching



Pre-programmed SRLG-disjoint protect path

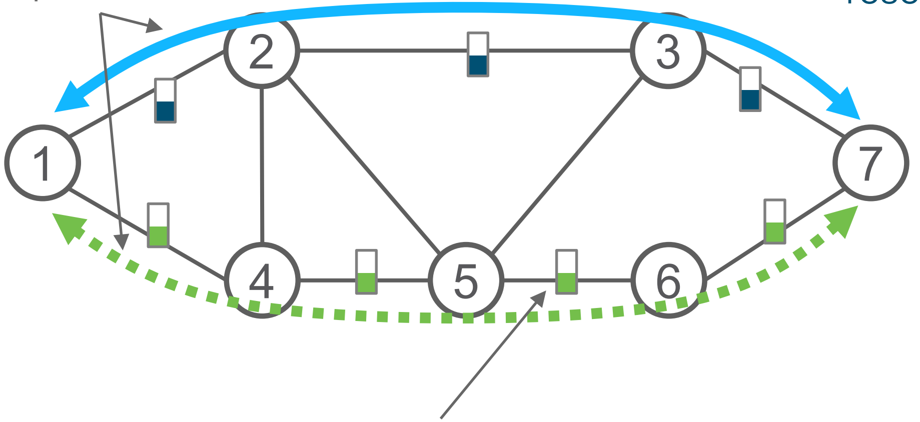
Bidirectional liveness monitoring on working and protect paths to ensure fast and consistent switchover

Guaranteed bandwidth



Centralized controller computes the paths and maintains bandwidth reservation bookkeeping

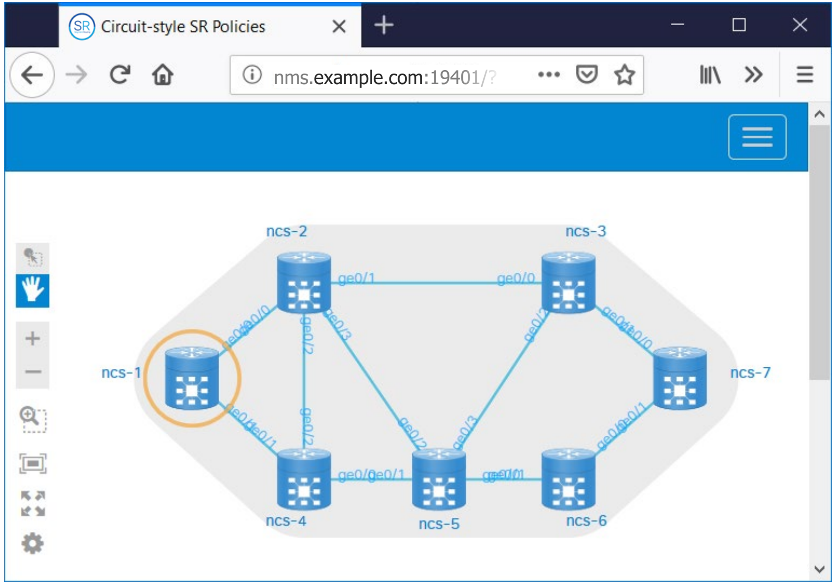
Bandwidth reserved on both working and protect paths



MQC configuration isolates circuit traffic from best-effort

Completely integrated with NMS

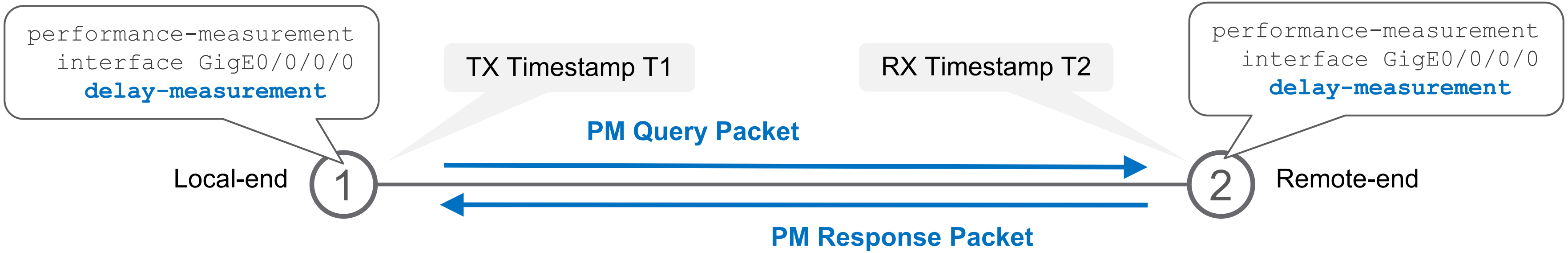
- The network management system takes care of the overall management of the TDM service circuits
  - End-to-end service provisioning
  - Network management assurance



cisco Live!

Per-Link Delay Measurement Reminder

Link Delay – Probe Measurement



```
performance-measurement
  interface GigE0/0/0/0
    delay-measurement
```

```
performance-measurement
  interface GigE0/0/0/0
    delay-measurement
```

- One Way Delay = (T2 – T1)
- Timestamps added in hardware
- PM Query format: RFC 5357 (IP/UDP/TWAMP) or RFC 6374 (MPLS/GAL)

Default: every 3 sec

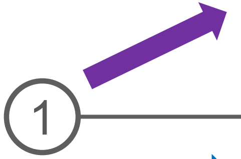
SR-TE handles Minimum delay (propagation delay)

- Minimum delay provides the propagation delay
  - fiber length / speed of light
- A property of the topology
  - with awareness of DWDM circuit change
- SR-TE (SR Policy or Flex-Algo) can optimize on min delay

Routing stability – Telemetry accuracy

Every 30sec
(10 queries)

Every 120sec
IF significant min change
THEN trigger an ISIS/OSPF flood



EDT Telemetry push

ISIS/OSPF/BGP-LS update

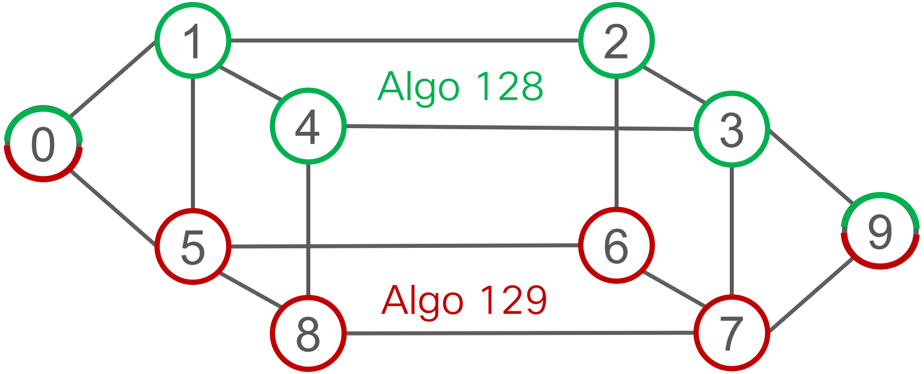
SR IGP Flex-Algo Reminder

SR IGP Flexible Algorithms

- Complements the SR-TE solution with customizable IGP Algorithms

- We call “Flex-Algo”
  - The algorithm is defined by the operator, on a per-deployment basis
- Flex-Algo K is defined as
  - The minimization of a specified metric: IGP, TE or delay
  - The exclusion of certain link properties: link-affinity, SRLG, …

Dual Plane



16009 Algo 0

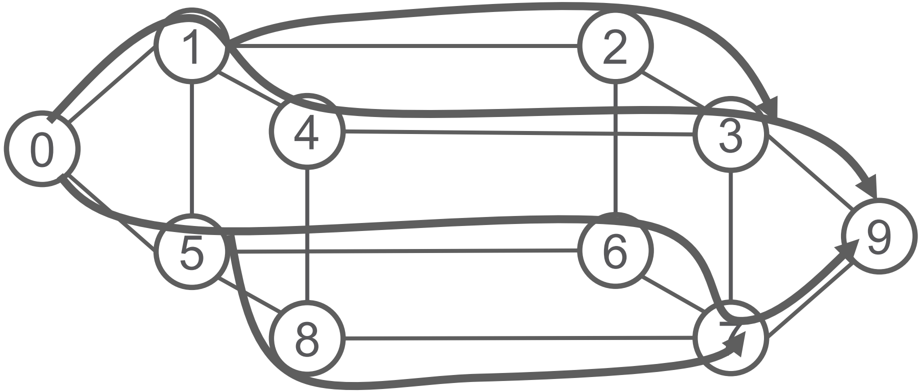
16809 Flex-Algo 128

16909 Flex-Algo 129

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

✓
- Operator customizes its IGP to deliver multiple transport services on same infra
- Automated by the IGP and leveraging TI-LFA and uLoop

Dual Plane



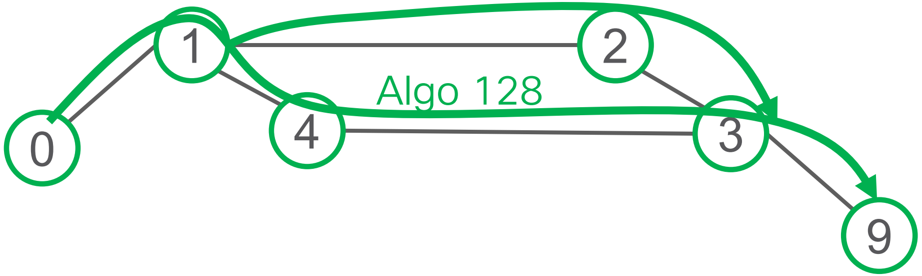
16009 Algo 0

- All the nodes support Algo 0: minimize IGP metric

✓

- Operator customizes its IGP to deliver multiple transport services on same infra
- Automated by the IGP and leveraging TI-LFA and uLoop

Dual Plane

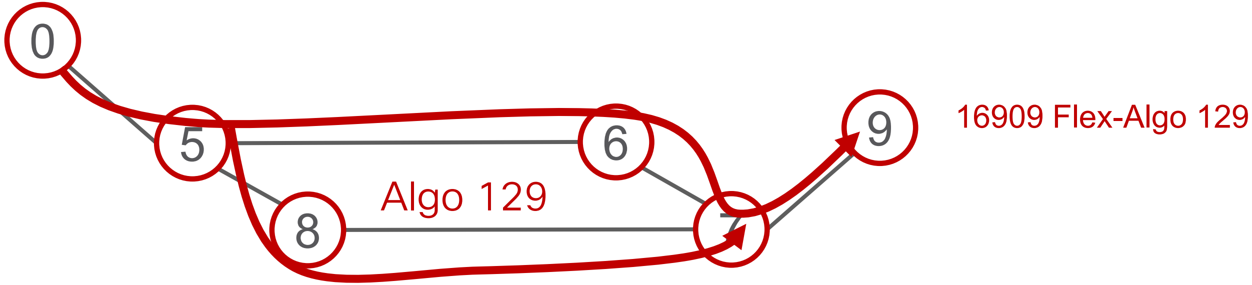


16809 Flex-Algo 128

- Green nodes also support 128: minimize IGP metric

- Operator customizes its IGP to deliver multiple transport services on same infra
- Automated by the IGP and leveraging TI-LFA and uLoop

Dual Plane



- Red nodes also support 129: minimize IGP metric

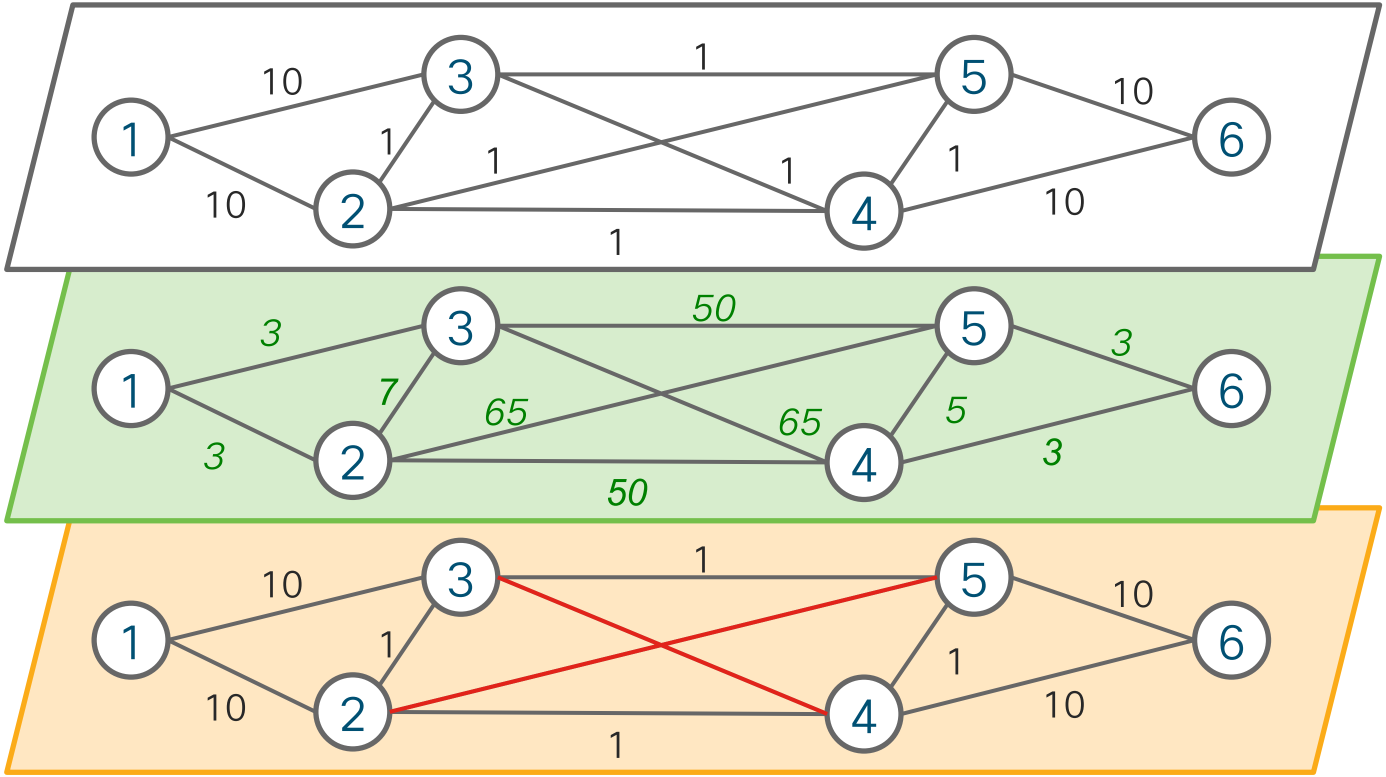
- ✓ Operator customizes its IGP to deliver multiple transport services on same infra
- Automated by the IGP and leveraging TI-LFA and uLoop

Network slicing with SR IGP Flexible Algorithms

Default slice
Algo 0

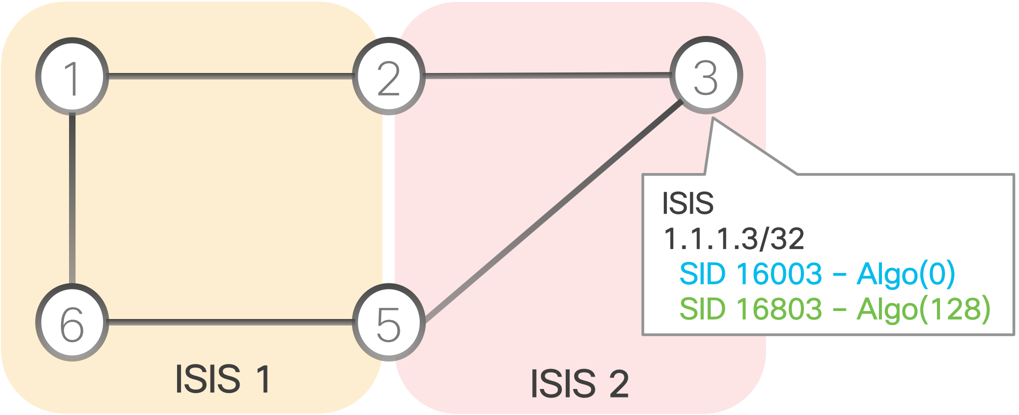
Low delay slice
Algo 128
(minimize delay metric)

Secured slice
Algo 129
(minimize IGP, exclude non-encrypted links)



Optimal End-to-End Inter-Domain Paths with Flex-Algo

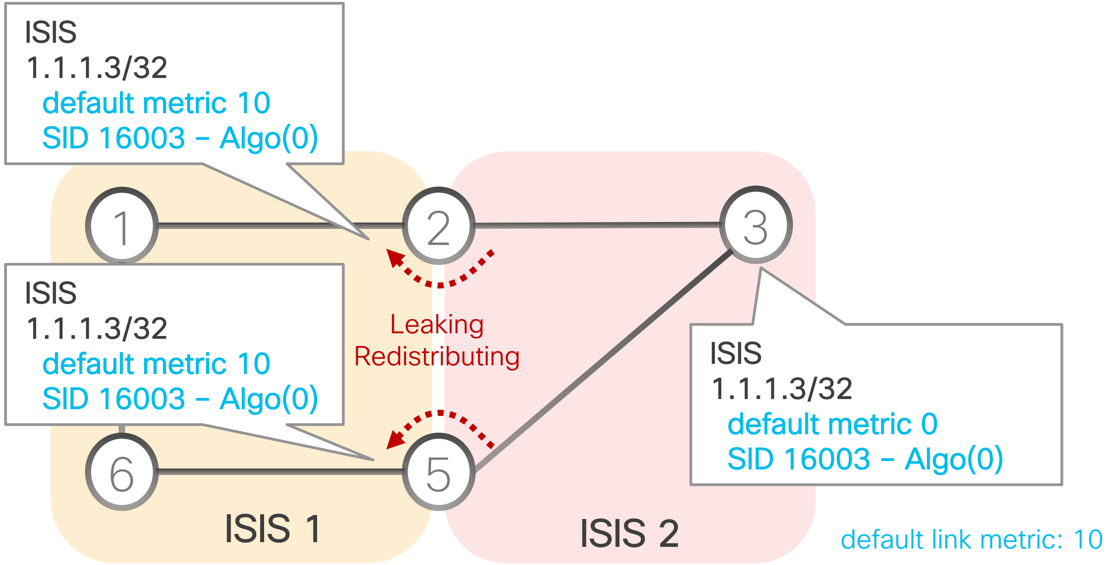
Flex-Algo multi-domain network



ISIS
1.1.1.3/32
SID 16003 – Algo(0)
SID 16803 – Algo(128)

- All nodes in both domains participate in Algo(0) and Algo(128)
- Algo(128) is defined as min-delay

Redistribute with default metric and Algo(0) SID

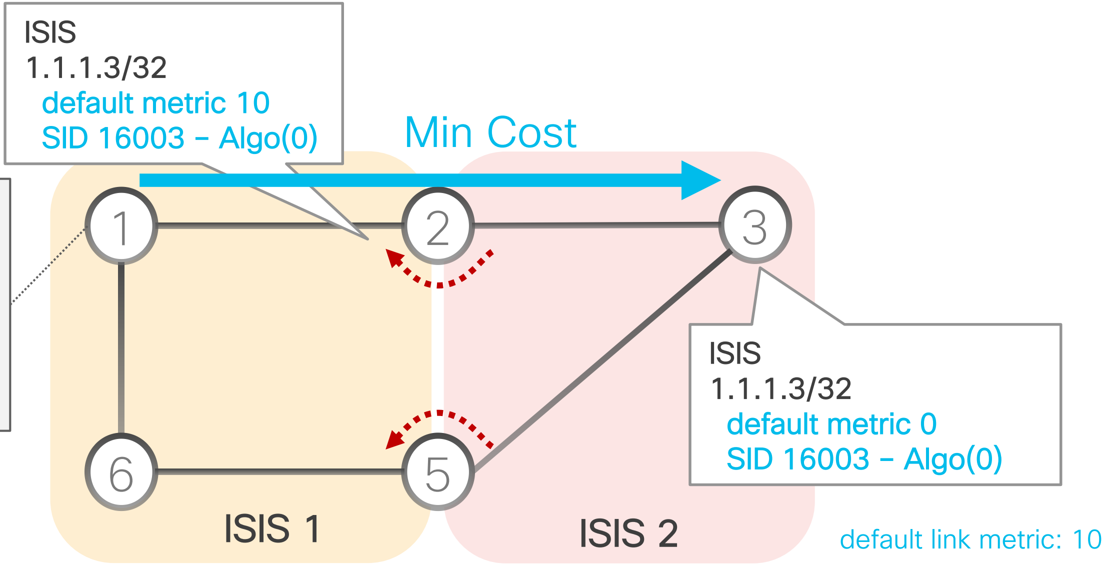


- Node 2 and 5 redistribute 1.1.1.3/32 with default metric and Algo(0) SID 16003

Optimal end-to-end Min-Cost path

```
FIB

16003 via 2
  metric 20(=10+10)
```



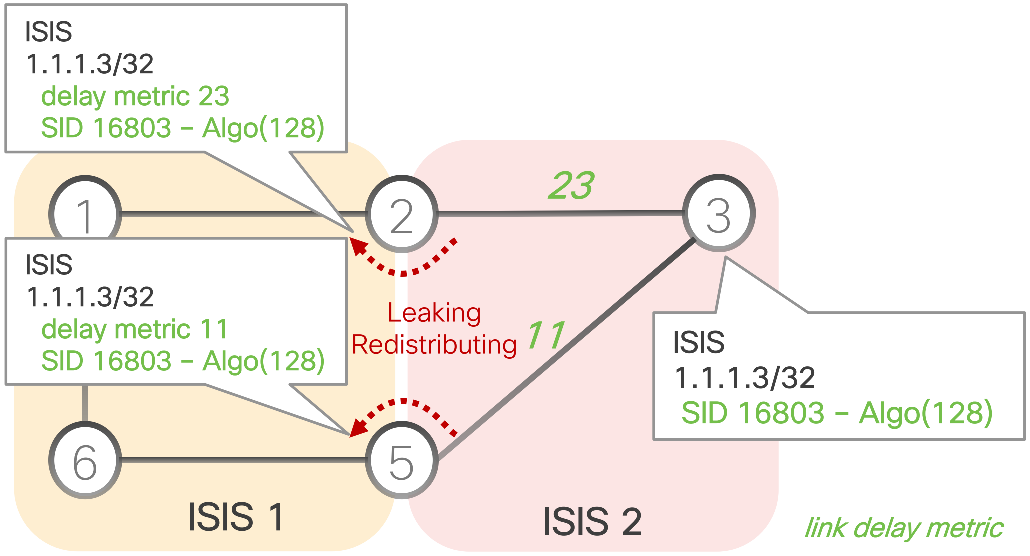
ISIS
1.1.1.3/32
  default metric 10
  SID 16003 – Algo(0)

ISIS
1.1.1.3/32
  default metric 0
  SID 16003 – Algo(0)

default link metric: 10

- IGP on Node 1 installs the Algo(0) SID 16003 via the optimal end-to-end min-cost path

Redistribute with delay metric and Algo(128) SID

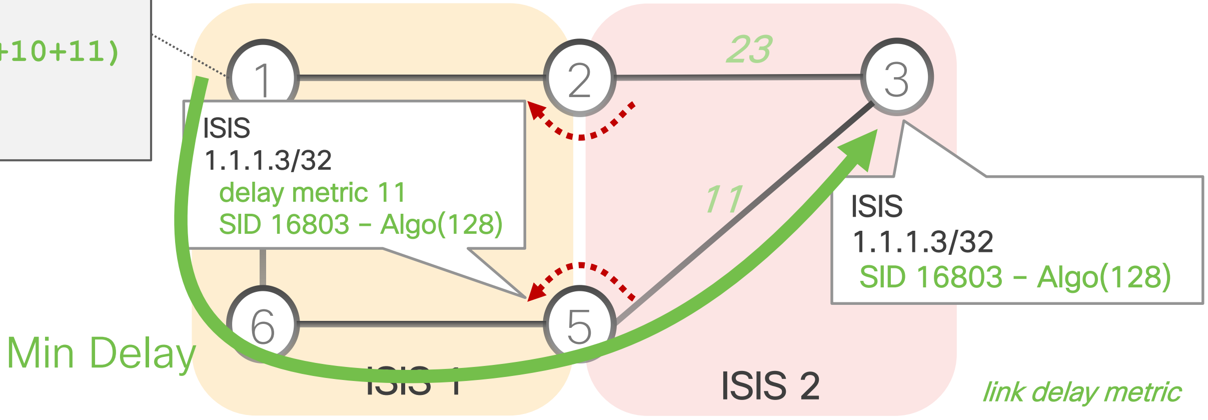


- Node 2 and 5 redistribute 1.1.1.3/32 with delay metric and Algo(128) SID 16803

Optimal end-to-end Min-Delay path

```
FIB

16803 via 6
  metric 31(=10+10+11)
```



ISIS
1.1.1.3/32
  delay metric 11
  SID 16803 – Algo(128)

ISIS
1.1.1.3/32
 SID 16803 – Algo(128)

- IGP on Node 1 installs the Algo(128) SID 16803 via the optimal end-to-end min-delay path

Per-Destination ODN/AS

Per-Destination Automated Steering

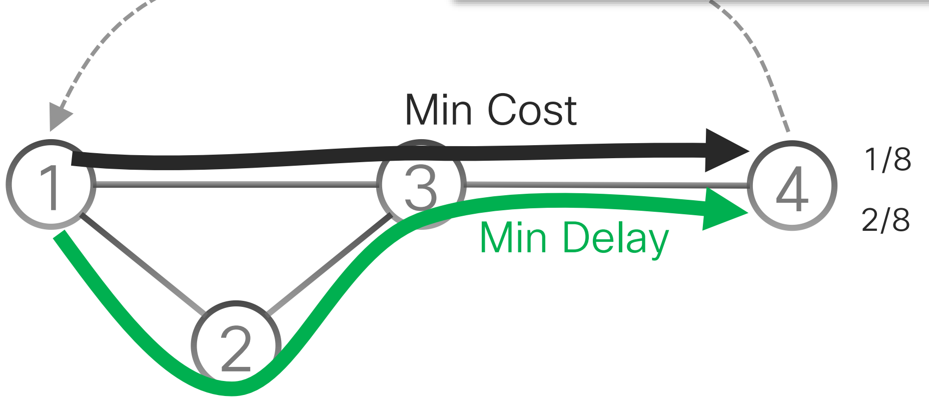
- Automated Steering steers service routes on their matching (intent + endpoint) SR Policy

BGP update:
1/8 via Node4 with **default path**

2/8 via Node4 with **min delay**

1/8 → default path to 4 (min cost)
Prefix-SID 16004

2/8 → SR Policy(min-delay, 4)
<16002, 16004>



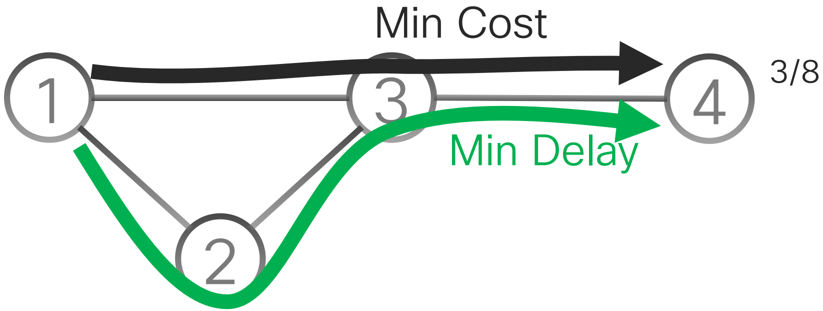
Need for Per-Flow ODN/AS

Same Destination

F1: 3/8 (DSCP 0) → default path to 4 (min cost) Prefix-SID 16004

F2: 3/8 (DSCP 46) → SR Policy(min-delay, 4) <16002, 16004>

Different Flows



Per-Flow ODN/AS

Per-Flow SR Policy

- A Per-Flow SR Policy provides up to 8 “ways” to the endpoint
- The Forward-Class setting of the packet selects the “way”
- This “way” can be a
  - Traffic-Engineered SR path: the low-delay path to the endpoint
  - Classic RIB path: the default shortest path to the endpoint

Forward-Class

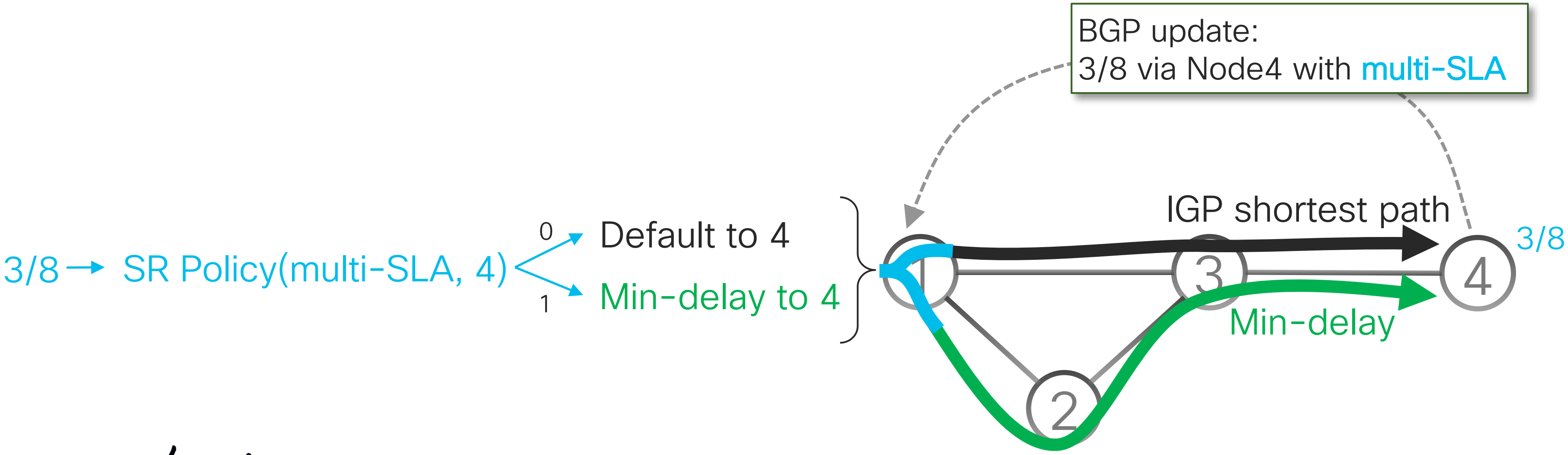
- FC: a local value attached to a packet within a router
  - Range from 0 to 7
- Set on the ingress interface on the basis of 5-tuple ACL or DSCP

```
class-map type traffic match-any MinDelay
 match dscp 46
end-class-map
!
class-map type traffic match-any PremiumHosts
 match access-group ipv4 PrioHosts
end-class-map
!
```

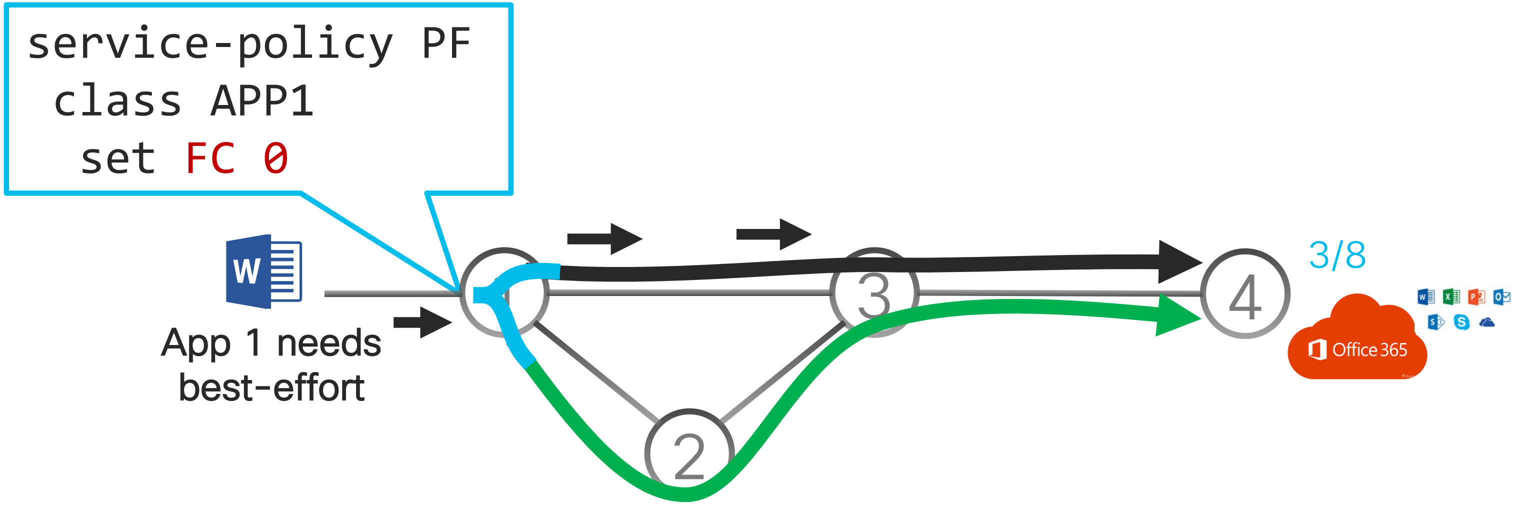
```
policy-map type pbr MyPerFlowPolicy
 class type traffic MinDelay
  set forward-class 1
 !
 class type traffic PremiumHosts
  set forward-class 2
 !
 class type traffic class-default
  set forward-class 0
 !
end-policy-map
```

Per-Flow Automated Steering (AS)

- AS automatically steers a service route on the PFP to E



App needs best-effort

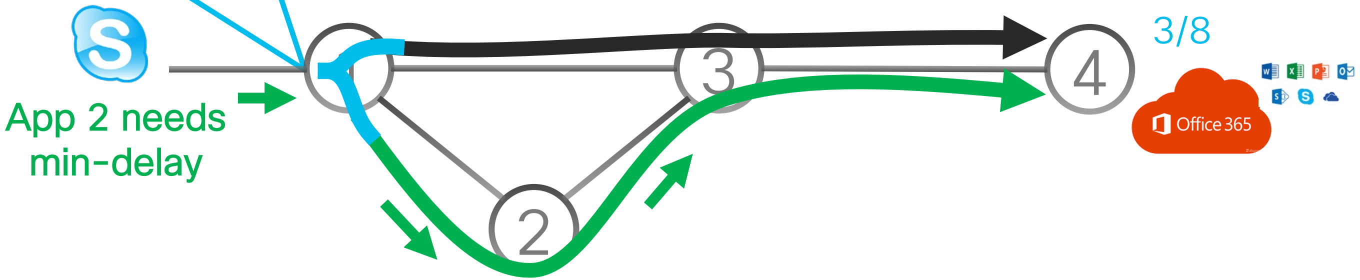


```
service-policy PF
 class APP1
  set FC 0
```

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

App needs min-delay

```
service-policy PF
 class APP2
  set FC 1
```



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

If we would have more time …

- ISIS Conditional Prefix Advertisement
- Flex-Algo OAM
- SR-PCE: Flex-Algo-aware Path Computation
- SR-PCE: SR-TE to BGP-LU Interworking
- SR ODN with EVPN
- BGP Peer-set EPE SID / Manual EPE
- Tree-SID

Conclusion

Industry at large backs up SR

Strong customer adoption
WEB, SP, DC,
Metro, Enterprise

De-facto SDN Architecture

Standardization
IETF

Multi-vendor Consensus

Open Source
Linux, VPP

SRv6

Simplicity Always Prevails

Furthermore with more scale and functionality

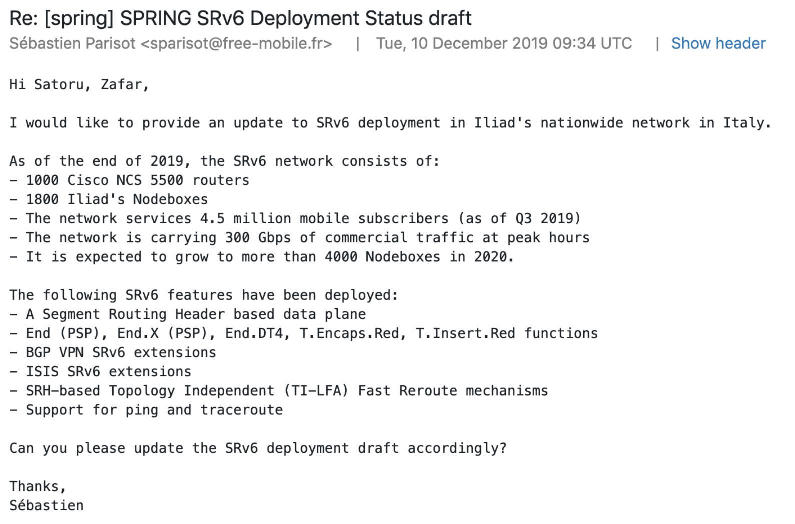
SRv6 Eco-System

At record speed

- 8 large-scale commercial deployments
  - Softbank, Iliad, China Telecom, LINE corporation, China Unicom, CERNET2, China Bank and Uganda MTN
- 18 HW linerate implementations
  - Cisco Systems, Huawei
  - Broadcom, Barefoot, Intel, Marvell, Mellanox
  - Multiple Interop Reports
- 9 open-source platforms/ Applications
  - Linux, FD.io VPP, P4, Wireshark, tcpdump, iptables, nftables, snort

Cisco Supports SoftBank on First Segment Routing IPv6 Deployment in Prep for 5G

Link to PR - https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=1969030



Re: [spring] SPRING SRv6 Deployment Status draft

Sébastien Parisot <sparisot@free-mobile.fr> | Tue, 10 December 2019 09:34 UTC | Show header

Hi Satoru, Zafar,

I would like to provide an update to SRv6 deployment in Iliad's nationwide network in Italy.

As of the end of 2019, the SRv6 network consists of:
- 1000 Cisco NCS 5500 routers
- 1800 Iliad's Nodeboxes
- The network services 4.5 million mobile subscribers (as of Q3 2019)
- The network is carrying 300 Gbps of commercial traffic at peak hours
- It is expected to grow to more than 4000 Nodeboxes in 2020.

The following SRv6 features have been deployed:
- A Segment Routing Header based data plane
- End (PSP), End.X (PSP), End.DT4, T.Encaps.Red, T.Insert.Red functions
- BGP VPN SRv6 extensions
- ISIS SRv6 extensions
- SRH-based Topology Independent (TI-LFA) Fast Reroute mechanisms
- Support for ping and traceroute

Can you please update the SRv6 deployment draft accordingly?

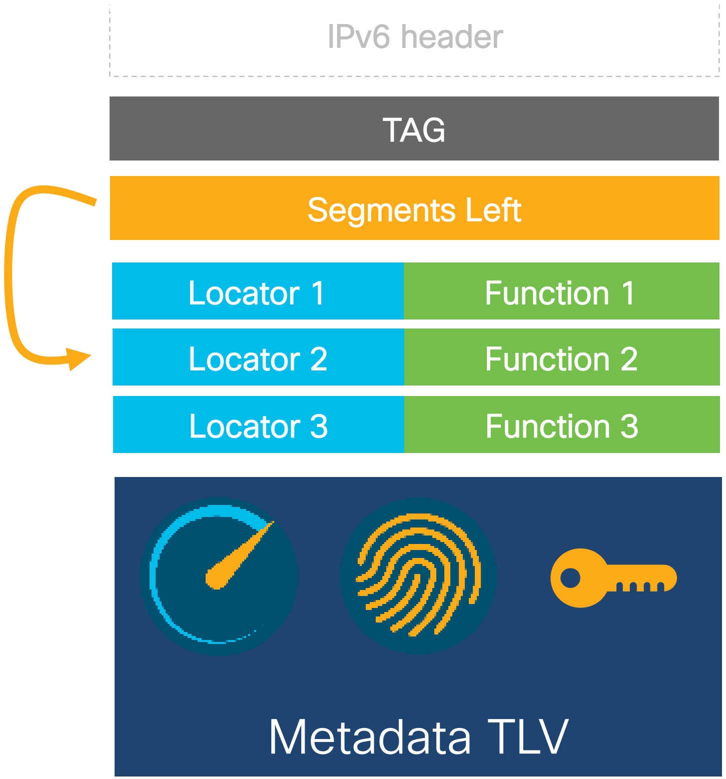
Thanks,
Sébastien

iliad cisco

- Nationwide deployment in Italy
- 1000 Cisco NCS 5500
- 1800 Iliad Nodeboxes

SRv6 - Reminder

SRv6 Header



IPv6 header

TAG

Segments Left

Locator 1 | Function 1

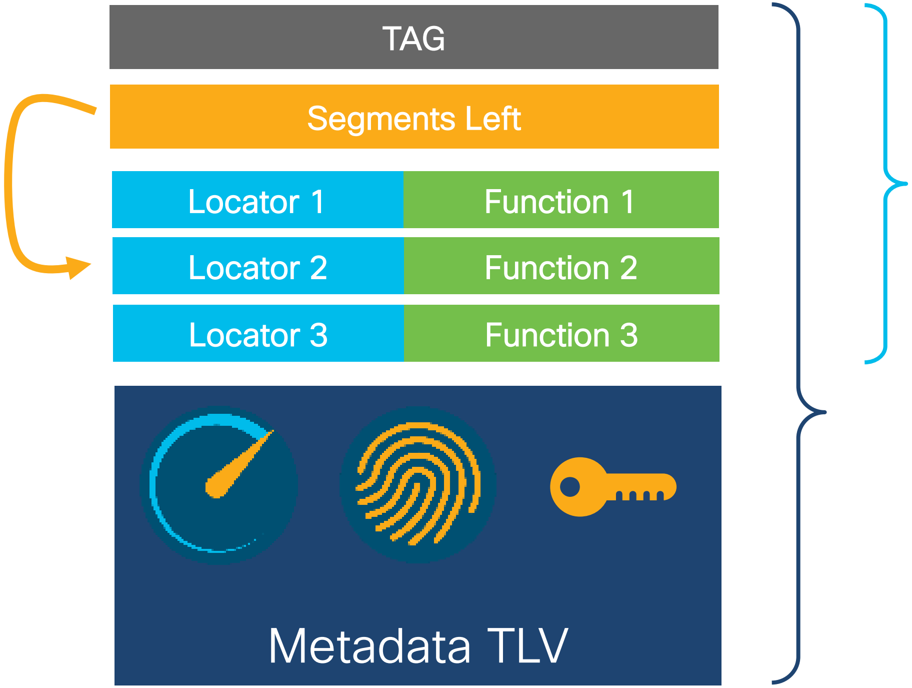
Locator 2 | Function 2

Locator 3 | Function 3

Metadata TLV

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Hdr Ext Len  | Routing Type  | Segments Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Last Entry   |     Flags     |              Tag              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[0] (128 bits IPv6 address)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
                              ...
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[n] (128 bits IPv6 address)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                                                             //
//         Optional Type Length Value objects (variable)       //
//                                                             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SRv6 for anything

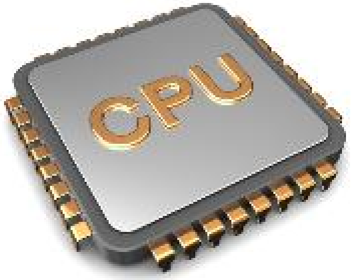


Optimized for HW processing

e.g. Underlay & Tenant use-cases

Optimized for SW processing

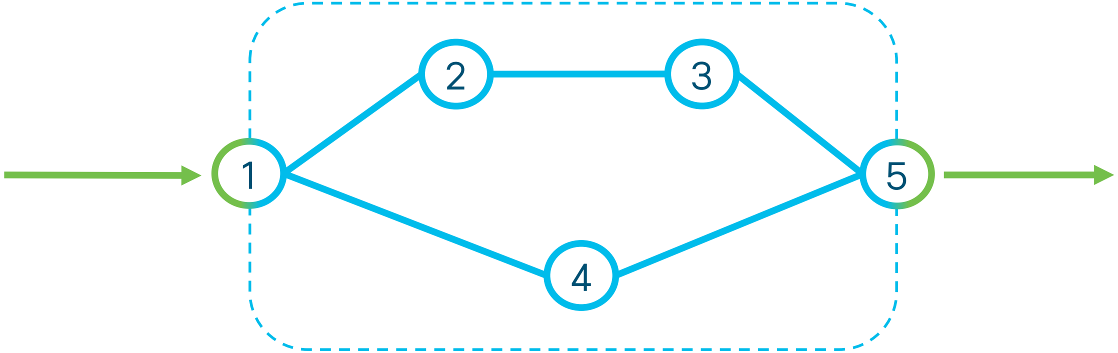
e.g. NFV, Container, Micro-Service



SRv6 Domain

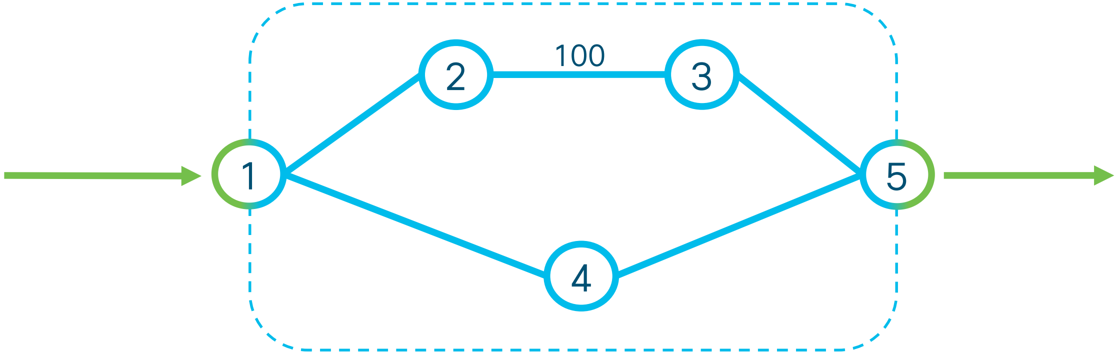
IPv6 enabled provider infrastructure

SR Domain



Encapsulation at the Domain ingress

- IPv4, IPv6 or L2 frame is encapsulated within the SR Domain
- Outer IPv6 header includes an SRH with the list of segments

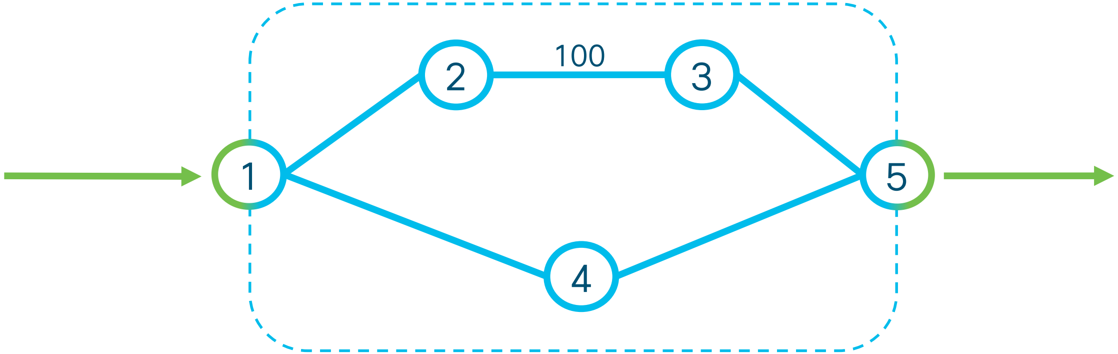


IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv6	SA = A:1::, DA = B:2:C3::
SR	(B:5:DB::) SL=1
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

SRH of the outer IPv6 encapsulation

- Domain acts as a giant computer
- The network program in the outer SRH is executed



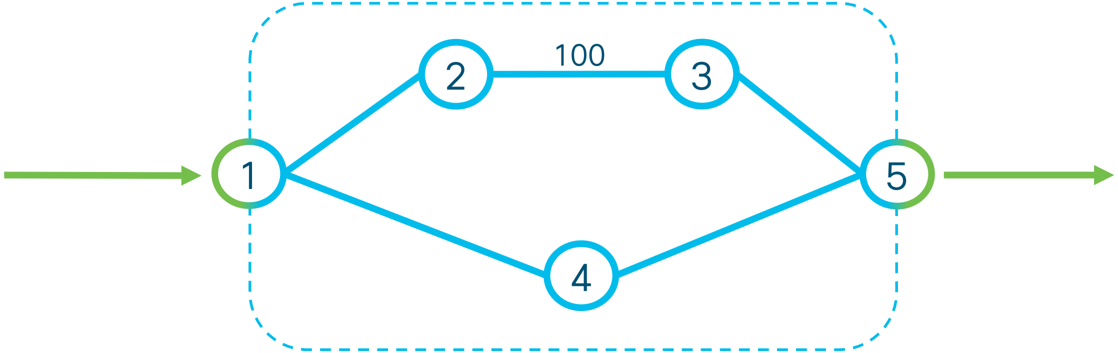
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv6	SA = A:1::, DA = B:2:C3::
SR	(B:5:DB::) SL=1
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv6	SA = A:1::, DA = B:5:DB::
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

Decapsulation at Domain Egress

- Egress PE removes the outer IPv6 header as the packet leaves the SR domain



IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

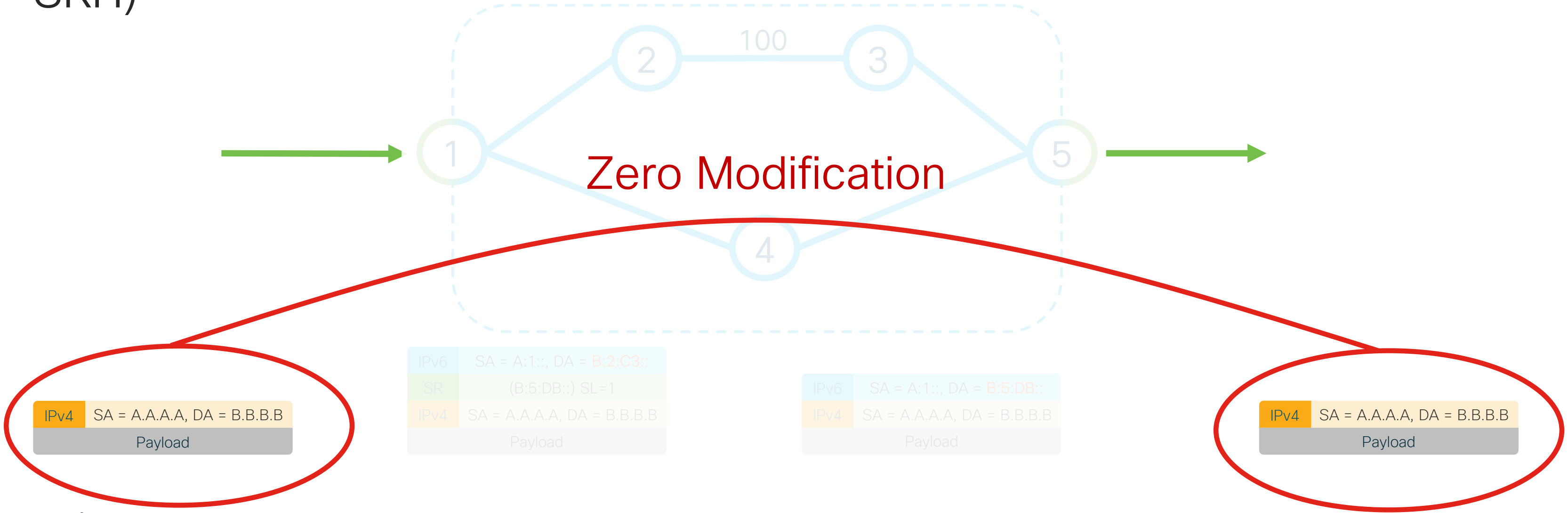
IPv6	SA = A:1::, DA = B:2:C3::
SR	(B:5:DB::) SL=1
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv6	SA = A:1::, DA = B:5:DB::
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

End-to-End Integrity

- End-to-end integrity principle is strictly guaranteed
  - Inner packet is unmodified
  - Same as SR-MPLS (MPLS stack is replaced by IPv6 outer header and SRH)



100
2
3
1
5
4
Zero Modification
IPv4
SA = A.A.A.A, DA = B.B.B.B
Payload
IPv6
SA = A:1::, DA = B:2:C3::
SR
(B:5:DB::) SL=1
IPv4
SA = A.A.A.A, DA = B.B.B.B
Payload
IPv6
SA = A:1::, DA = B:5:DB::
IPv4
SA = A.A.A.A, DA = B.B.B.B
Payload
IPv4
SA = A.A.A.A, DA = B.B.B.B
Payload

IETF

Assumed leadership

- Important investment to lead the IETF for the eco-system
- Lots of work
- Please help
  - Co-authoring concrete and useful work
  - Dismissing pure political plays

Image: texelart © 123RF.com

SR Architecture

- RFC 8402 – Proposed Standard
  - Defines SR-MPLS with MPLS dataplane and Label SID’s
  - Defines SRv6 with SRH and SRv6 SID’s

SRv6

- RFC Proposed Standard
  - SRv6 DataPlane: SRH and SRv6 SID
- Last-Call
  - Network Programming (END, END.X, END.DX/DT, T.Encaps)
  - Control Plane (ISIS, BGP-LS)
  - Policy
  - OAM
- One IETF away to Last-Call
  - BGP

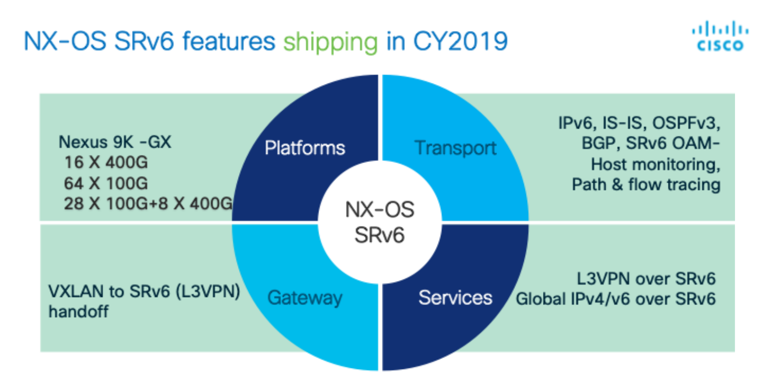
SRv6 - Roadmap

Shipping: NCS5500, NCS560, NCS540, ASR9k

- ISIS
  - TILFA and uLoop
  - Flex-Algo (Low-Delay Slice) with TILFA
- BGP
  - PIC Core/Edge
  - L3VPN (IPv4)
  - Internet (IPv4)
  - eVPN VPWS
- SRv6-SR-MPLS Gateway
- OAM
  - Ping
  - Trace
  - SID Verification

Also in the DC – with linerate SRv6 @ 400G

- Amazing set of SRv6 network instructions @ 400G !



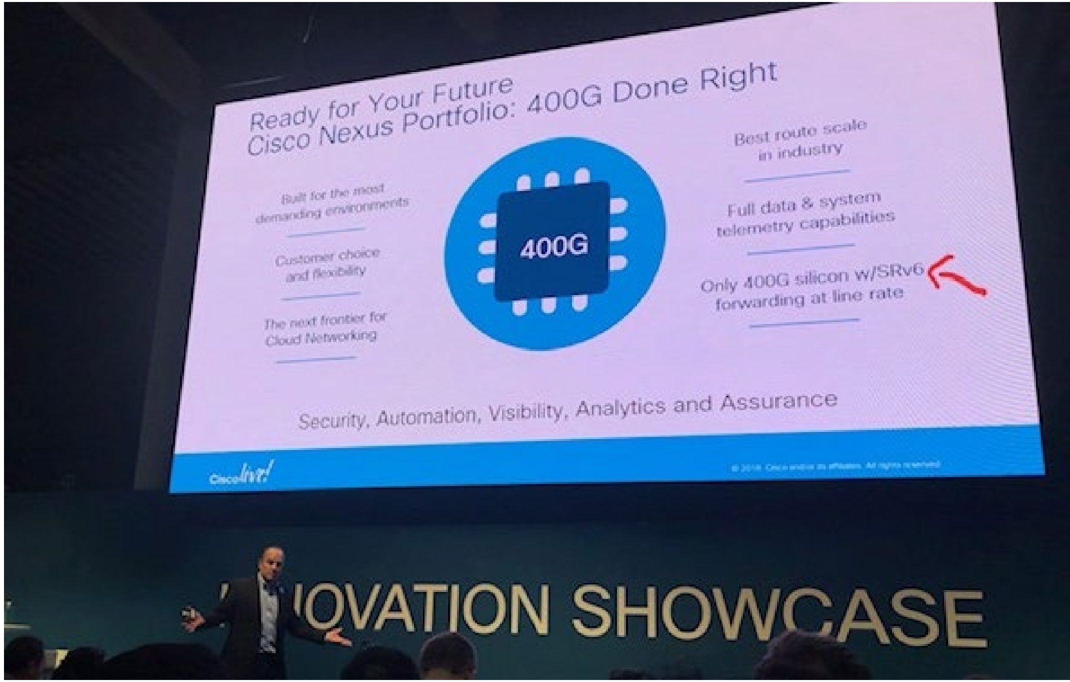
NX-OS SRv6 features shipping in CY2019

Nexus 9K -GX
16 X 400G
64 X 100G
28 X 100G+8 X 400G

IPv6, IS-IS, OSPFv3, BGP, SRv6 OAM-Host monitoring, Path & flow tracing

VXLAN to SRv6 (L3VPN) handoff

L3VPN over SRv6
Global IPv4/v6 over SRv6



Ready for Your Future
Cisco Nexus Portfolio: 400G Done Right

Built for the most demanding environments

Customer choice and flexibility

The next frontier for Cloud Networking

Best route scale in industry

Full data & system telemetry capabilities

Only 400G silicon w/SRv6 forwarding at line rate

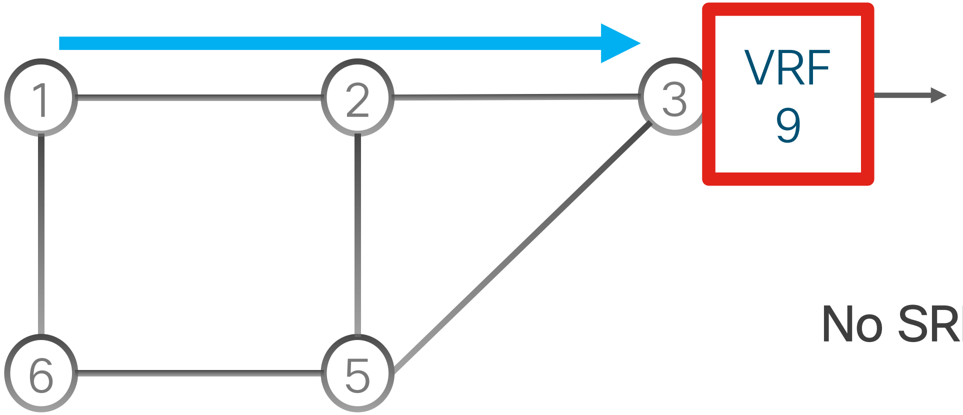
Security, Automation, Visibility, Analytics and Assurance

SRv6 Deployed Use-Cases

VPN over Best-Effort 5G Slice

Network Program: B:3:V(9)

B: locator block is associated with ISIS base algo (Low Cost, Best Effort)

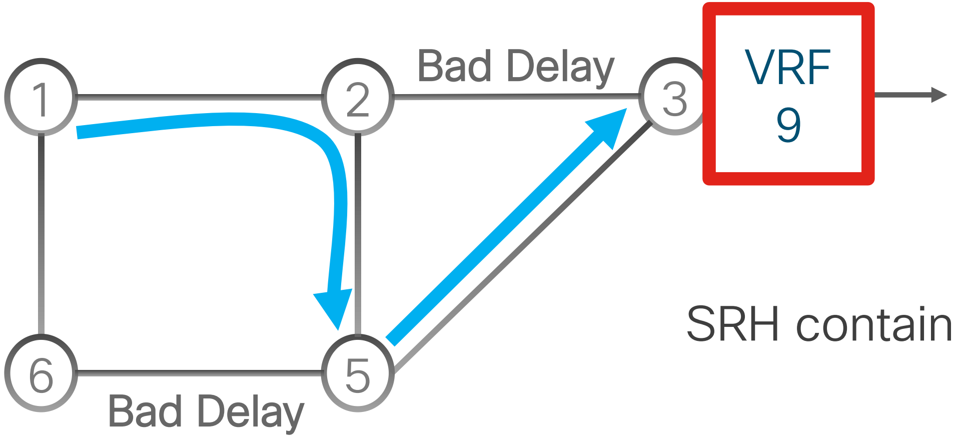


No SRH!

VPN with Low-Delay 5G Slice - SR-TE Option

Network Program: B:2:C5 then B:3:V(9)

B: locator block is associated with ISIS base algo (Low Cost)

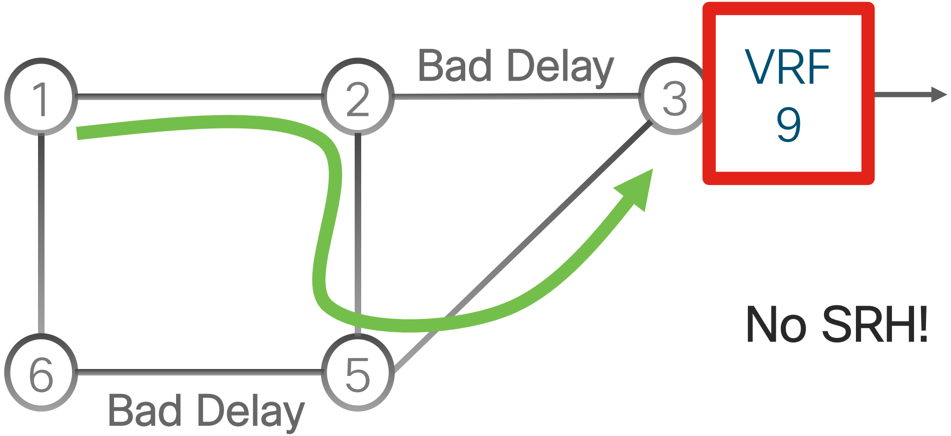


SRH contains 1 single SID

VPN with Low-Delay 5G Slice – Flex-Algo Option

Network Program: D:3:V(9)

D: locator block is associated with Low Delay Flex-Algo

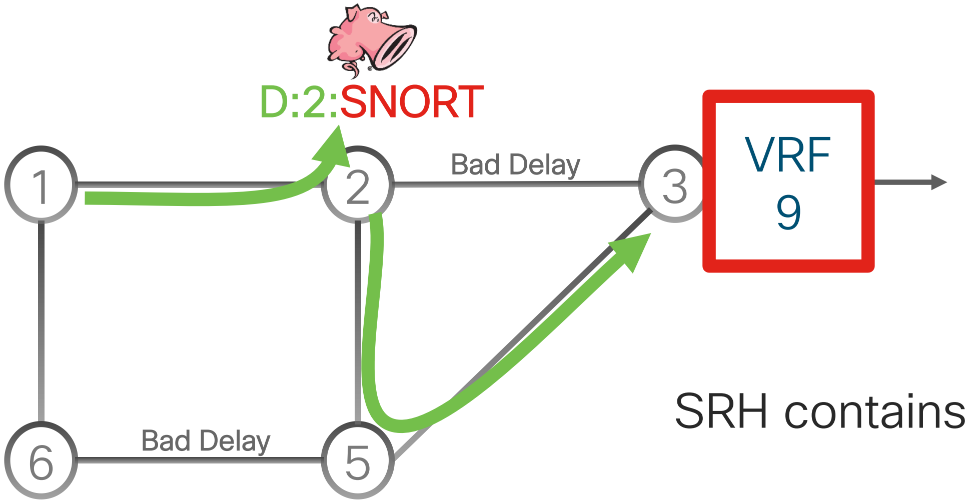


No SRH!

Snort Firewall, VPN & Low-Delay Slice

Network Program: D:2:SNORT then D:3:V(9)

D: locator block is associated with Low Delay Flex-Algo

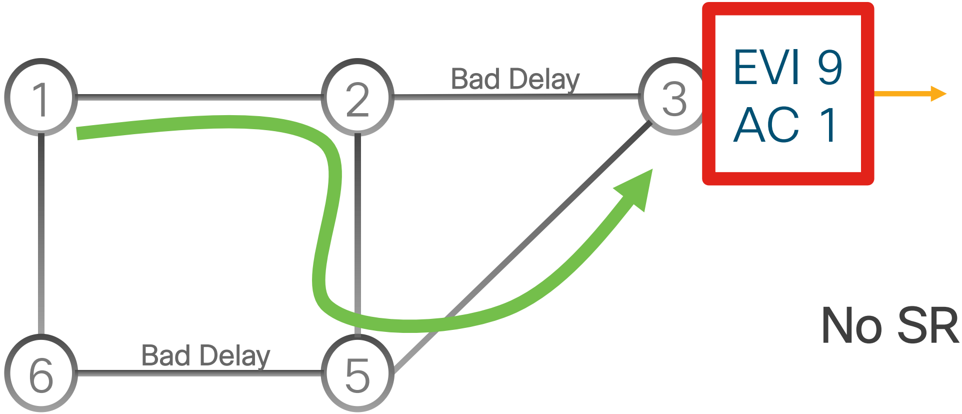


SRH contains 1 Single SID

EVPN VPWS Single-Home & Low-Delay 5G Slice

Network Program: D:3:X(1)

D: locator block is associated with Low Delay Flex-Algo

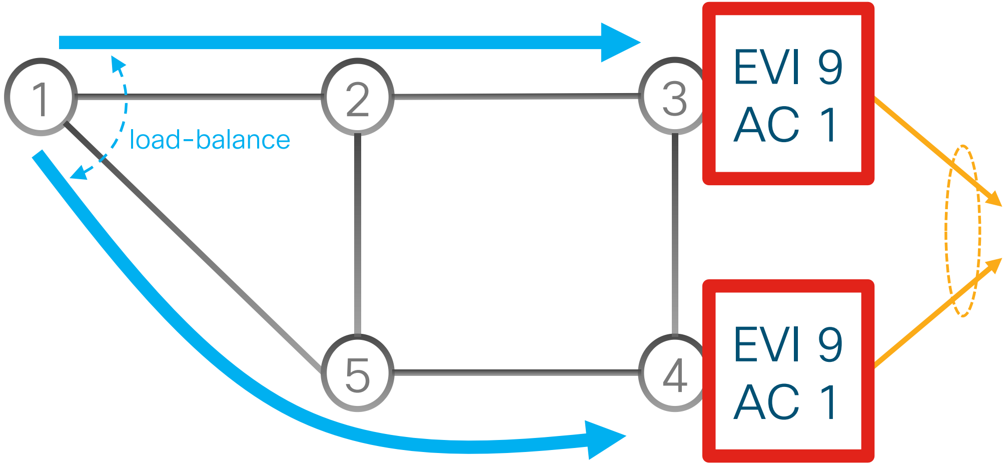


No SRH!

EVPN VPWS MH All-Active & Best-Effort 5G Slice

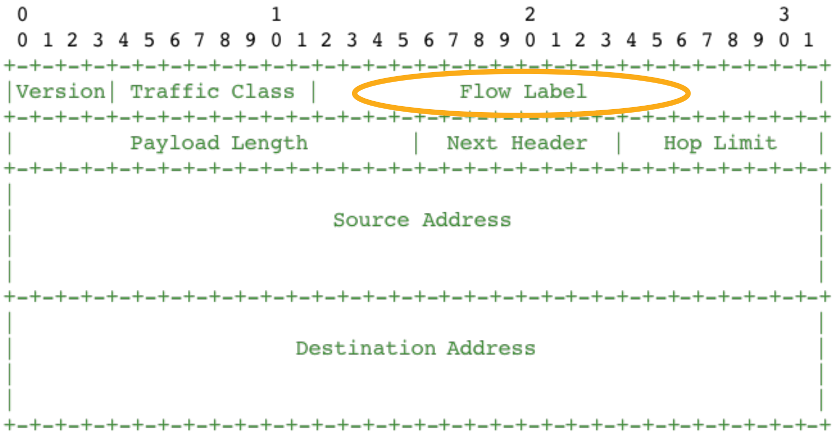
Network Program: B:3:X(1) or B:4:X(1)

B: locator block is associated with ISIS base algo (Low Cost)



No SRH!

Load-balancing



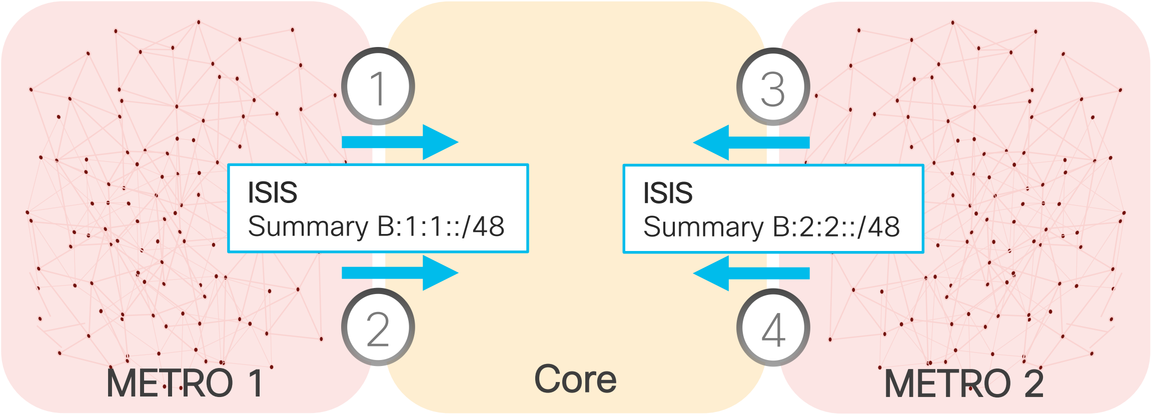
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |  Next Header  |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Source Address                        |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      Destination Address                      |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- 20-bit entropy
- No additional protocol
  - infamous mpls entropy label

Seamless Incremental Deployment

- As soon as the network supports plain IPv6 forwarding
  - A new SRv6-VPN service only requires PE upgrade
  - TE objective can be achieved with a few well selected TE waypoints
  - FRR is deployed incrementally

Prefix Summarization



- Back to basic IP routing and summarization
- No BGP inter-AS Option A/B/C

SRv6 has excellent native Scale

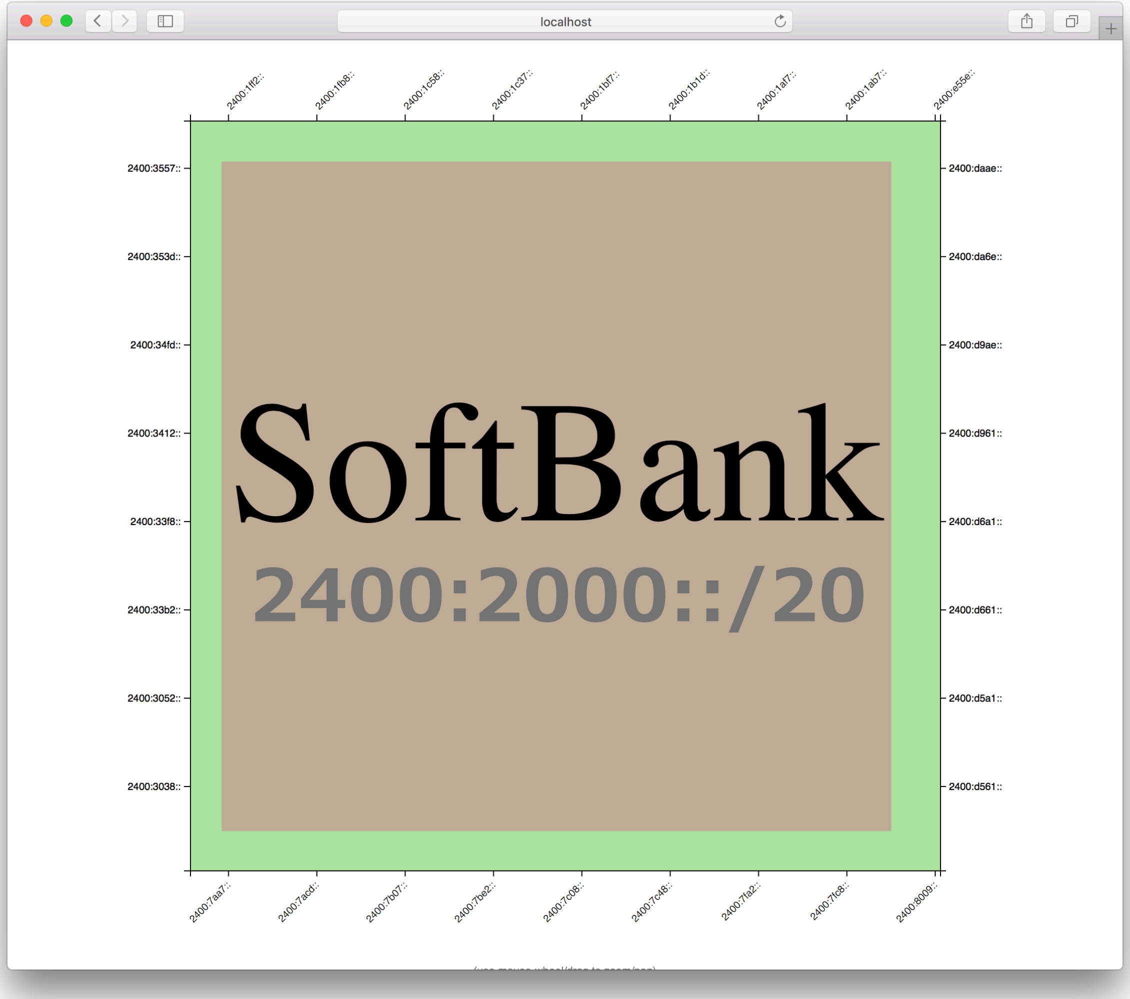
- Many use-cases do not even use an SRH ☺
  - Any VPN (L3VPN, PW, eVPN)
  - Egress Peering Engineering
  - Low-Latency or Disjoint Slicing
  - Optimal Load-Balancing
- If SRH is needed, most cases will use 1 or 2 SID's
- Prefix Summarization gain
- Talk to the operators who deployed, they are happy to share experience

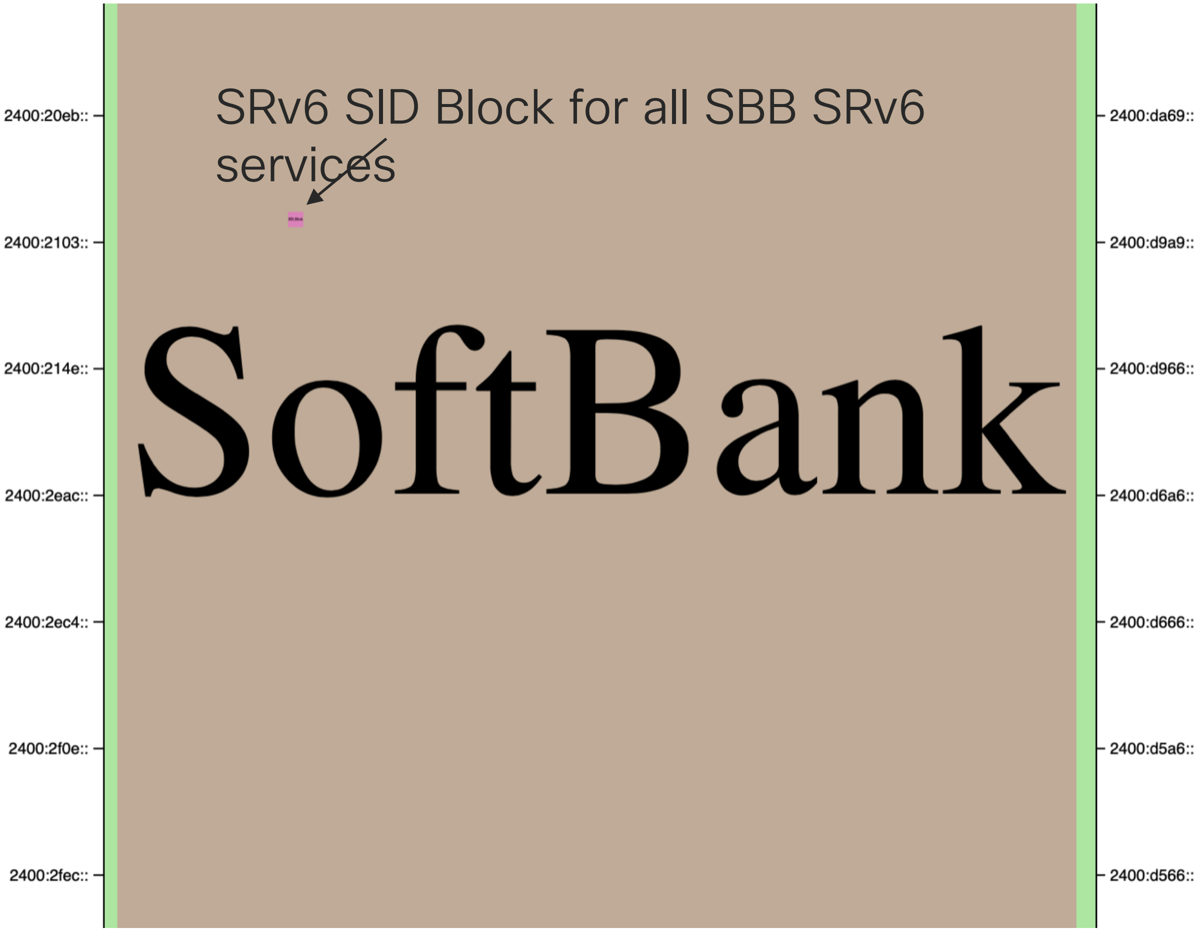
SRv6/MPLS L3 Service Interworking Gateway

Insignificant IPv6 Address Usage

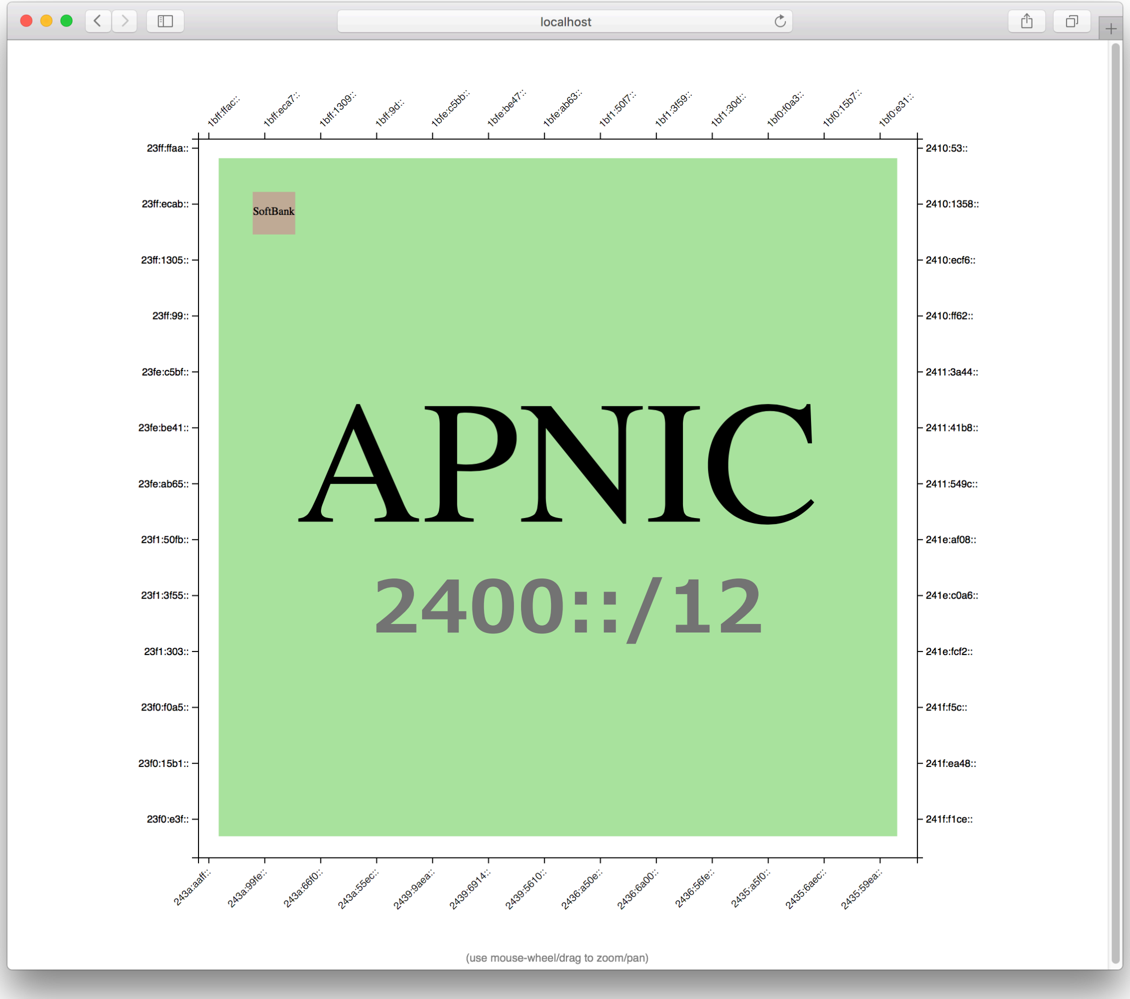
SBB example

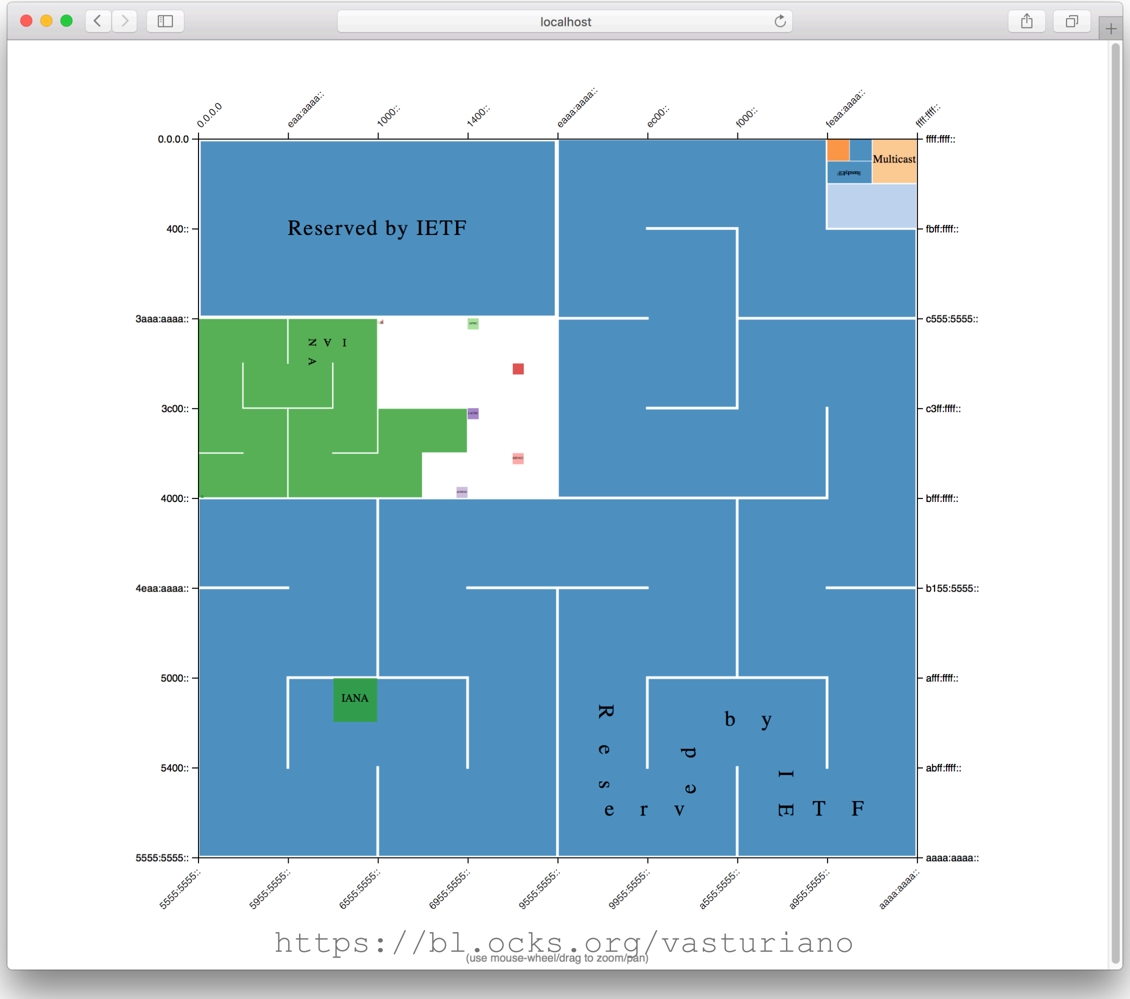
Credit to Satoru Matsushima – Softbank
who credits Vasco Astriano and Dave Plonka (Akamai)


localhost
2400:1ff2::
2400:1fb8::
2400:1c58::
2400:1c37::
2400:1bf7::
2400:1b1d::
2400:1af7::
2400:1ab7::
2400:e55e::
2400:3557::
2400:353d::
2400:34fd::
2400:3412::
2400:33f8::
2400:33b2::
2400:3052::
2400:3038::
2400:daae::
2400:da6e::
2400:d9ae::
2400:d961::
2400:d6a1::
2400:d661::
2400:d5a1::
2400:d561::
SoftBank
2400:2000::/20
2400:7aa7::
2400:7acd::
2400:7b07::
2400:7be2::
2400:7c08::
2400:7c48::
2400:7fa2::
2400:7fc8::
2400:8009::
(use mouse wheel/drag to zoom/pan)



SRv6 SID Block for all SBB SRv6 services


localhost
1bff:ffac::
1bff:eca7::
1bff:1309::
1bff:9d::
1bfe:c5bb::
1bfe:be47::
1bfe:ab63::
1bf1:50f7::
1bf1:3f59::
1bf1:30d::
1bf0:f0a3::
1bf0:15b7::
1bf0:e31::
23ff:ffaa::
23ff:ecab::
23ff:1305::
23ff:99::
23fe:c5bf::
23fe:be41::
23fe:ab65::
23f1:50fb::
23f1:3f55::
23f1:303::
23f0:f0a5::
23f0:15b1::
23f0:e3f::
2410:53::
2410:1358::
2410:ecf6::
2410:ff62::
2411:3a44::
2411:41b8::
2411:549c::
241e:af08::
241e:c0a6::
241e:fcf2::
241f:f5c::
241f:ea48::
241f:f1ce::
SoftBank
APNIC
2400::/12
243a:aaff::
243a:99fe::
243a:66f0::
243a:55ec::
2439:9aea::
2439:6914::
2439:5610::
2436:a50e::
2436:6a00::
2436:56fe::
2435:a5f0::
2435:6aec::
2435:59ea::
(use mouse-wheel/drag to zoom/pan)



(use mouse-wheel/drag to zoom/pan)

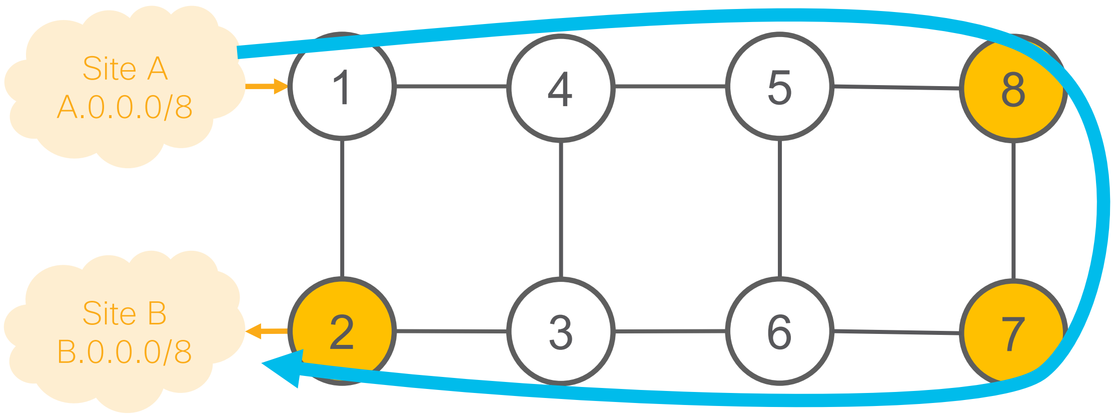
https://bl.ocks.org/vasturiano

Micro-Program

Intuitive SRv6 Network Program

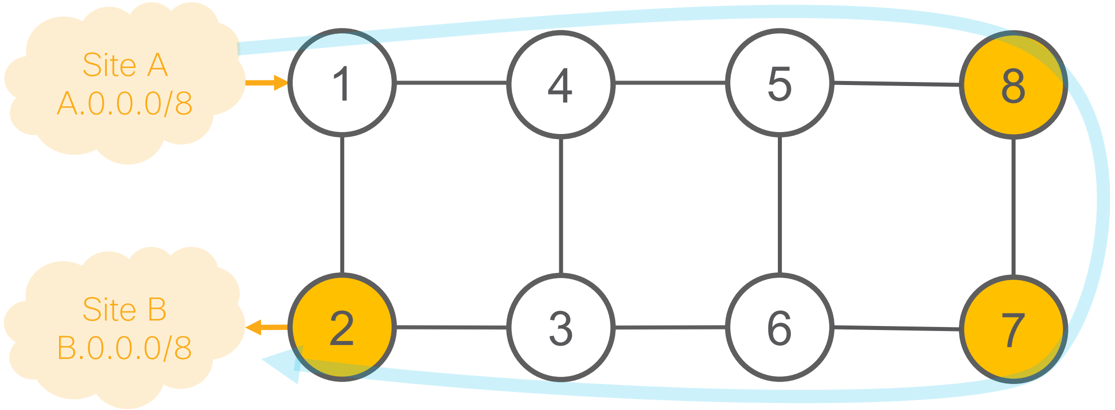
- Program
  - list of instructions contained in DA/SRH
- Instruction
  - SRv6 SID
- Micro-program
  - SRv6 SID (called carrier) that contains a list of micro-instructions
- Micro-Instruction
  - SRv6 uSID, can represent any behavior: TE, VPN, Service

SRv6 uSID illustration



- Traffic engineered path via 8 and 7 with a single 128-bit SRv6 SID
- Node 1 encapsulates IPv4 packet from Site A and sends an IPv6 packet with DA = bbbb:bbbb:0800:0700:0200:0000:0000:0000

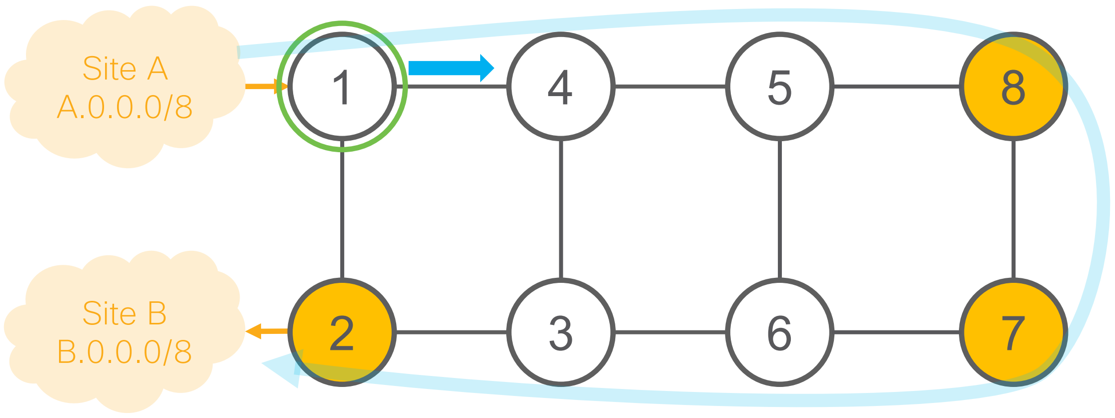
Routing



- Node 8 advertises the IGP route bbbb:bbbb:0800::/48
- Node 7 advertises the IGP route bbbb:bbbb:0700::/48
- Node 2 advertises the IGP route bbbb:bbbb:0200::/48

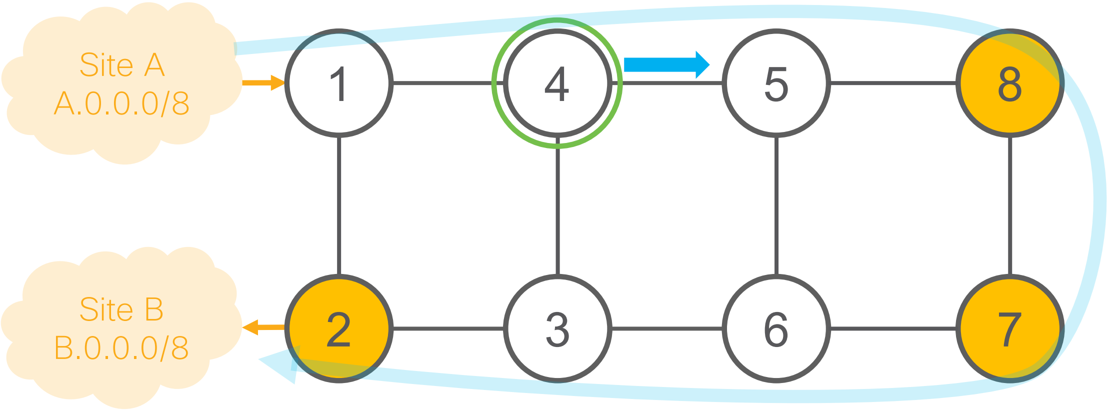
No new IGP extension required!

@1



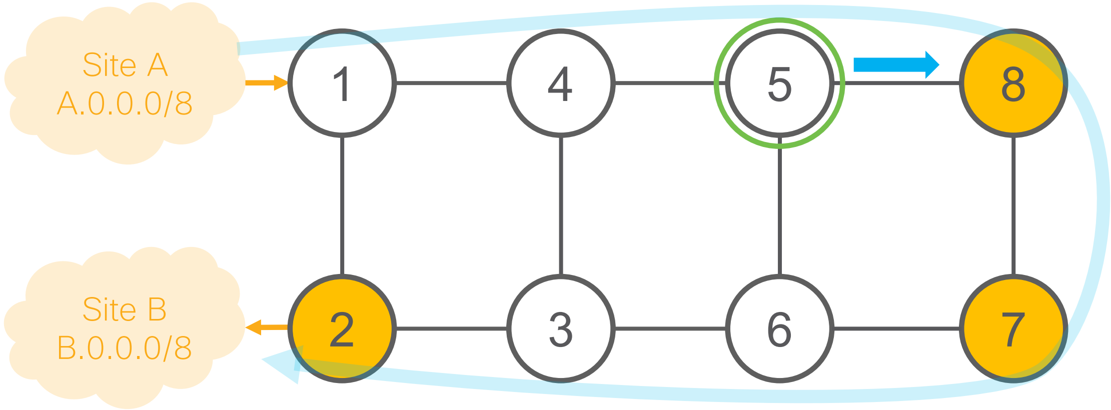
- DA = bbbb:bbbb:0800:0700:0200:0000:0000:0000
- Node 1 forwards to 4 (shortest-path to 8 (bbbb:bbbb:0800::/48))
- Seamless deployment through classic IPv6 nodes

@4



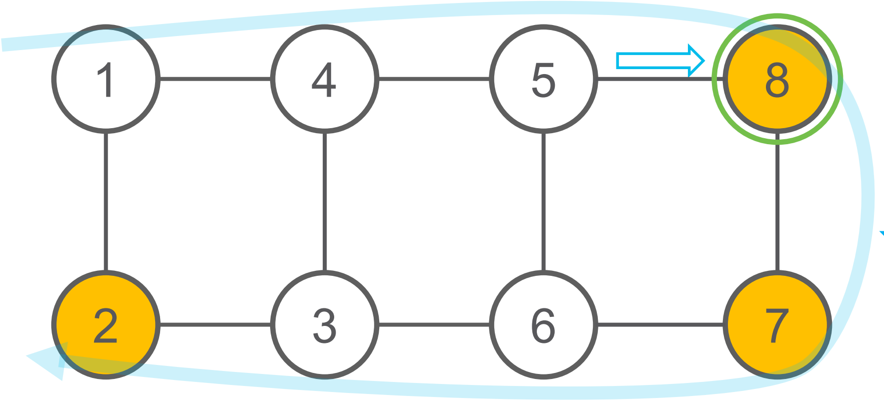
- DA = bbbb:bbbb:0800:0700:0200:0000:0000:0000
- Node 4 forwards to 5 (shortest-path to 8 (bbbb:bbbb:0800::/48))
- Seamless deployment through classic IPv6 nodes

@5



- DA = bbbb:bbbb:0800:0700:0200:0000:0000:0000
- Node 5 forwards to 8
- Seamless deployment through classic IPv6 nodes

@8: Shift and Forward



Rx'd DA: bbbb:bbbb:0800:0700:0200:0000:0000:0000

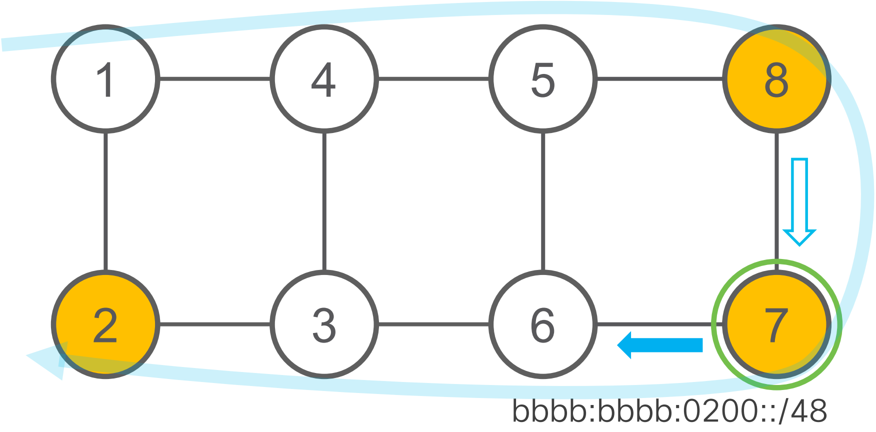
shift << 16

Tx'd DAbbbb:bbbb:0700:0200:0000:0000:0000

bbbb:bbbb:0700::/48

```
FIB Longest-Match bbbb:bbbb:0800::/48 → Pseudo-code:
  Copy bits [48:127] into position [32:111]
  Set bits at position [112:127] to 0
  Lookup the updated DA and forward
```

@7: Shift and Forward



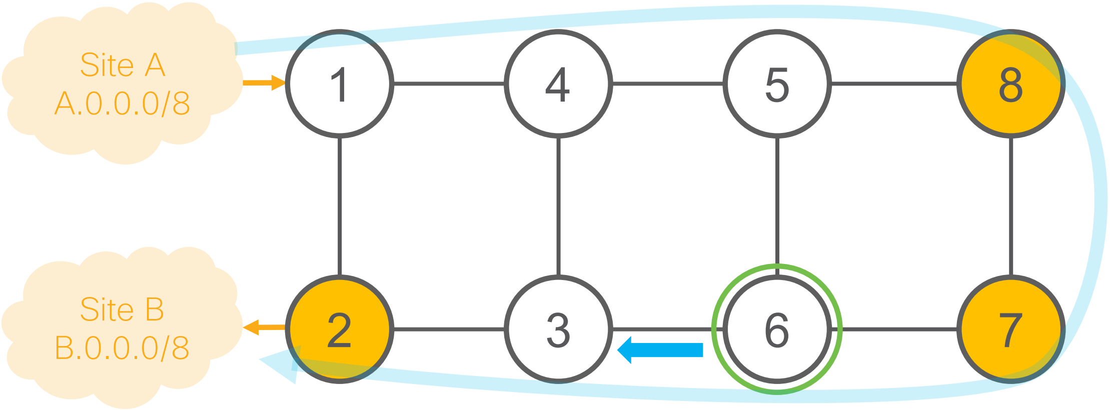
Rx’d DA: bbbb:bbbb:0700:0200:0000:0000:0000:0000

shift << 16

Tx’d DA:bbbb:bbbb:0200:0000:0000:0000:0000

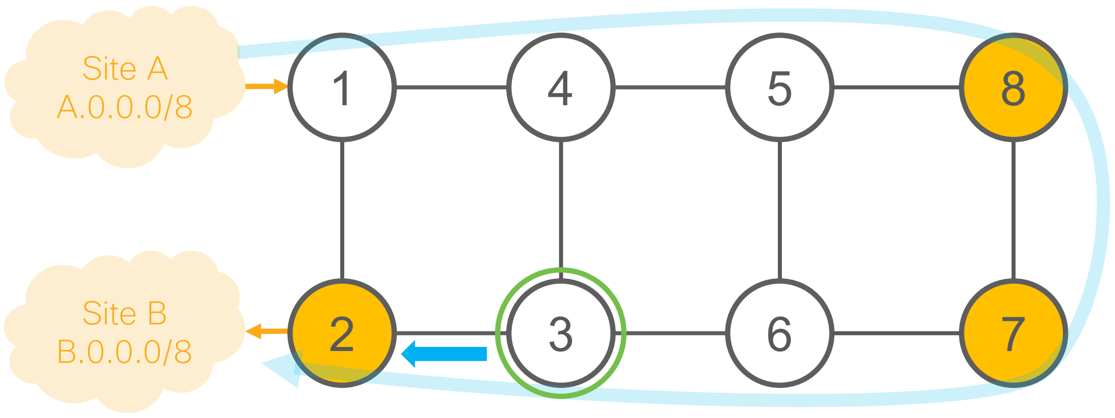
```
FIB Longest-Match bbbb:bbbb:0700::/48 → Pseudo-code:
  Copy bits [48:127] into position [32:111]
  Set bits at position [112:127] to 0
  Lookup the updated DA and forward
```

@6



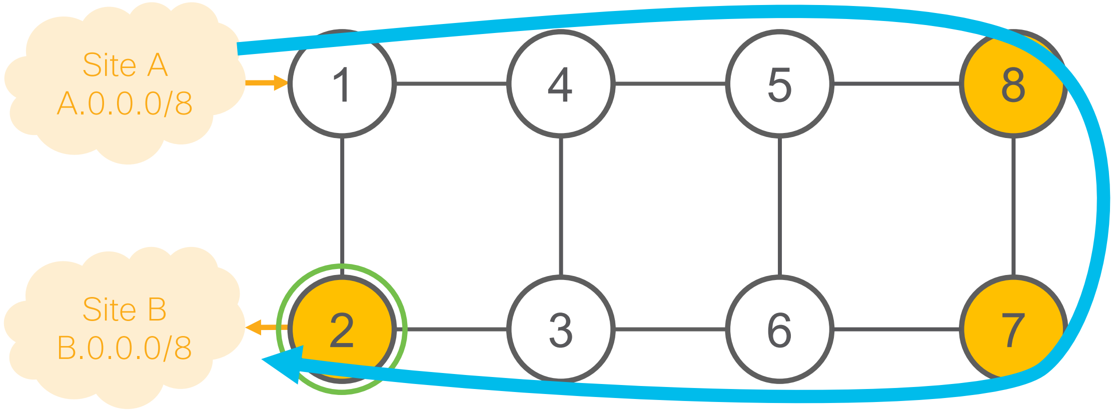
- DA = bbbb:bbbb:0200:0000:0000:0000:0000:0000
- Node 6 forwards to 3 (bbbb:bbbb:0300::/48)
- Seamless deployment through classic IPv6 nodes

@3



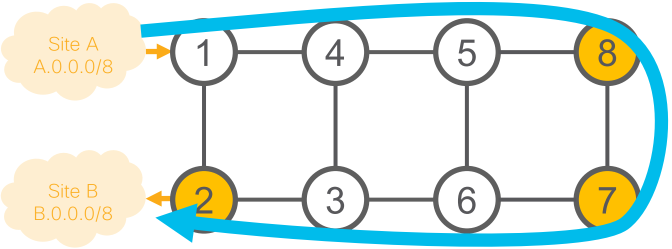
- DA = bbbb:bbbb:0200:0000:0000:0000:0000:0000
- Node 3 forwards to 2 (bbbb:bbbb:0200::/48)
- Seamless deployment through classic IPv6 nodes

@2: SRv6 End.DX4 behavior



- Match bbbb:bbbb:0200:0000::/64
- SRv6 Network Programming "End with decaps and IPv4 xconnect" behavior
  → Decapsulate and cross-connect inner IPv4 packet to Site B

Recap



- @1: inner packet P encapsulated with outer DA bbbb:bbbb:0800:0700:0200:0000:0000:0000
- @4 & @5: classic IP forwarding, outer DA unchanged
- @8: SRv6 uN behavior: shift and forward, outer DA becomes bbbb:bbbb:0700:0200:0000:0000:0000:0000
- @7: SRv6 uN behavior: shift and forward, outer DA becomes bbbb:bbbb:0200:0000:0000:0000:0000:0000
- @6 & @3: classic IP forwarding, outer DA unchanged
- @2: SRv6 End.DX4: Decapsulate and cross-connect inner packet

Compliant with SRv6, Net Prog and IPv6

100% SRv6 and Net Prog compliant

- ✓ Just another SID, just another pseudocode
- ✓ Any SID in SRH or DA can be a uSID Carrier
- ✓ uSIDs can be combined with any other SID

IPv6 compliant

- ✓ Leverage classic IP longest-match lookup
- ✓ Leverage classic IP-in-IP
- ✓ Use any IPv6 block available to the operator

uSID Benefits

Data Plane	✓ Best MTU efficiency (6 uSIDs without SRH) ✓ Hyper-Scalable SR-TE (18 uSIDs with 40 bytes overhead) ✓ Hardware-friendly (linerate on merchant silicon)
Control Plane	✓ Scalable number of globally unique uSIDs per domain ✓ No new protocol extensions
IP Power	✓ IP summarization and longest match is POWERFUL ✓ FIB efficiency 2 to 3 times gain vs MPLS ✓ Optimal IPv6 load-balancing (flow label)

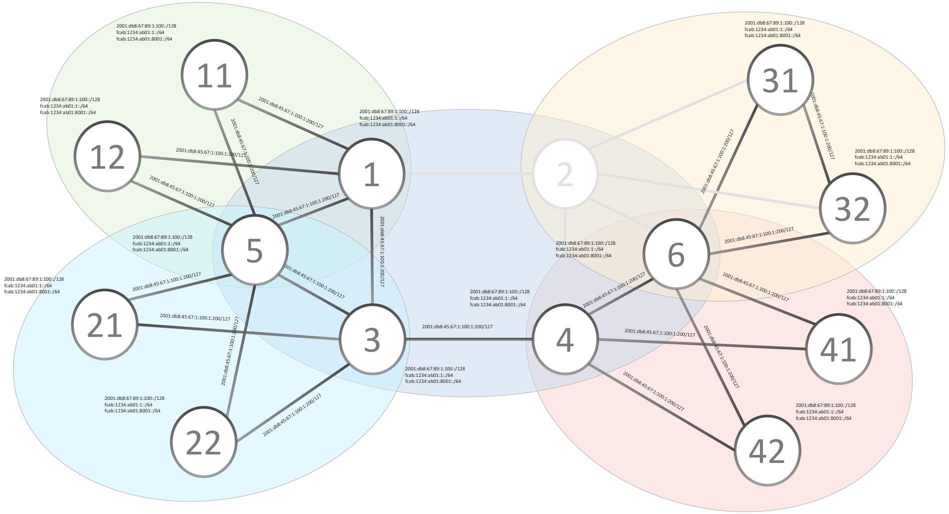
SRv6 - Automation

SRv6 Automation

NSO

NSO “click” and the following happens

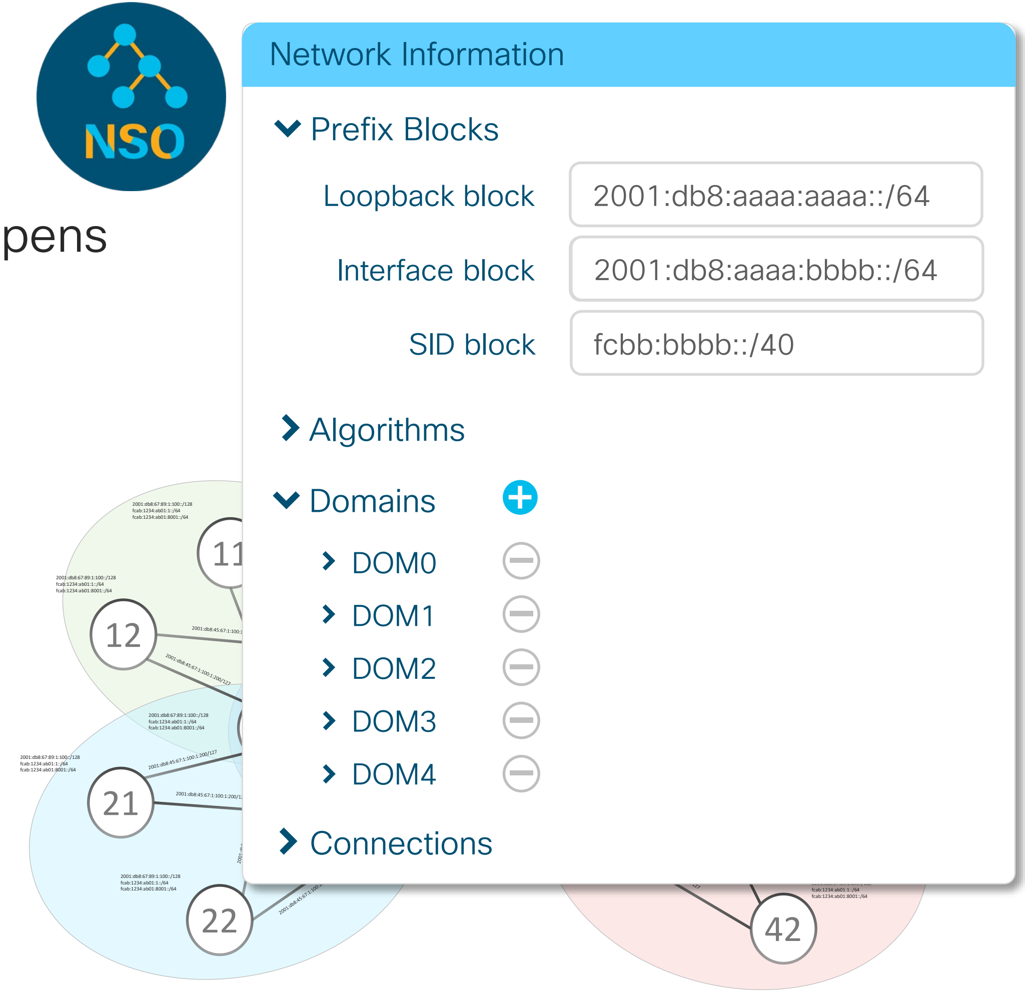
- Address allocation
  - Loopback and interfaces
- SID allocation
  - Algo 0 and Flex-Algos
- Multi-Domain
- ISIS summarization and redistribution between domains
- TI-LFA
- BFD


2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
11
2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
31
2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
12
2001:db8:45:67:1:100:1:200/127
2001:db8:45:67:1:100:1:200/127
2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
1
2
2001:db8:45:67:1:100:1:200/127
2001:db8:45:67:1:100:1:200/127
2001:db8:45:67:1:100:1:200/127
2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
32
2001:db8:45:67:1:100:1:200/127
2001:db8:45:67:1:100:1:200/127
2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
5
2001:db8:45:67:1:100:1:200/127
2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
6
2001:db8:45:67:1:100:1:200/127
2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
2001:db8:45:67:1:100:1:200/127
2001:db8:45:67:1:100:1:200/127
2001:db8:45:67:1:100:1:200/127
2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
2001:db8:45:67:1:100:1:200/127
2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
21
2001:db8:45:67:1:100:1:200/127
2001:db8:45:67:1:100:1:200/127
3
2001:db8:45:67:1:100:1:200/127
4
2001:db8:45:67:1:100:1:200/127
41
2001:db8:45:67:1:100:1:200/127
2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
2001:db8:45:67:1:100:1:200/127
2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
2001:db8:45:67:1:100:1:200/127
2001:db8:67:89:1:100::/128
fcab:1234:ab01:1::/64
fcab:1234:ab01:8001::/64
22
42

SRv6 Automation

NSO "click" and the following happens

- Address allocation
  - Loopback and interfaces
- SID allocation
  - Algo 0 and Flex-Algos
- Multi-Domain
- ISIS summarization and redistribution between domains
- TI-LFA
- BFD



Network Information

Prefix Blocks

Loopback block: 2001:db8:aaaa:aaaa::/64

Interface block: 2001:db8:aaaa:bbbb::/64

SID block: fcbb:bbbb::/40

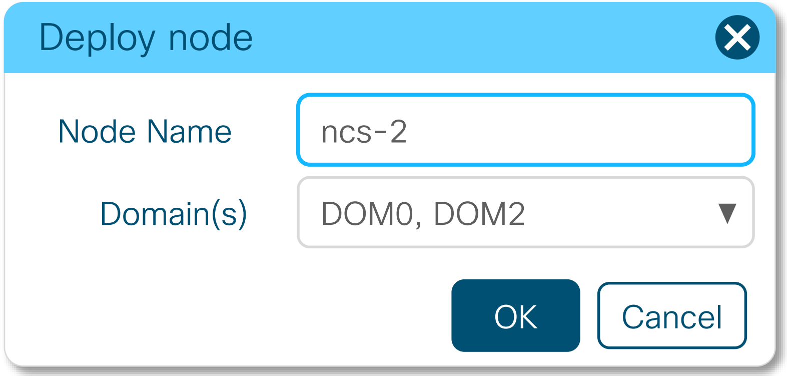
Algorithms

Domains

- DOM0
- DOM1
- DOM2
- DOM3
- DOM4

Connections

SRv6 Automation



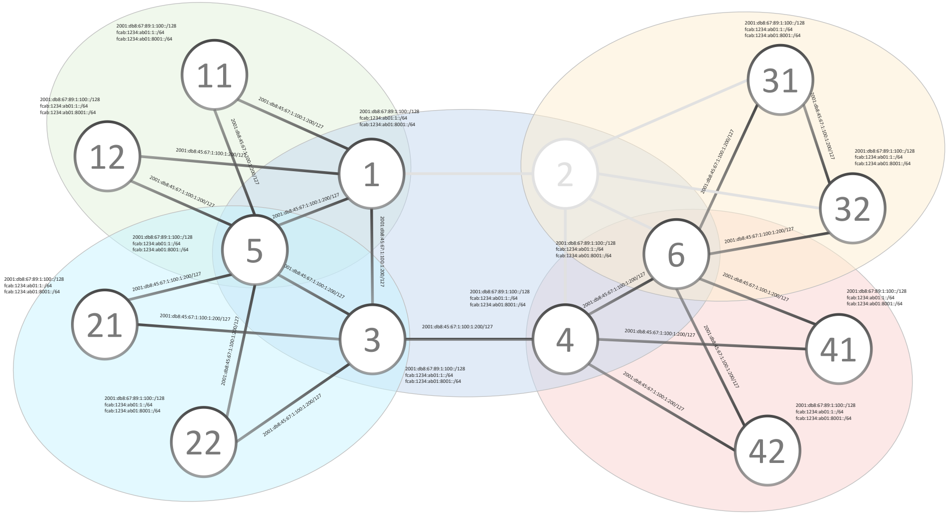
Deploy node

Node Name: ncs-2

Domain(s): DOM0, DOM2 ▼

NSO “click” and the following happens

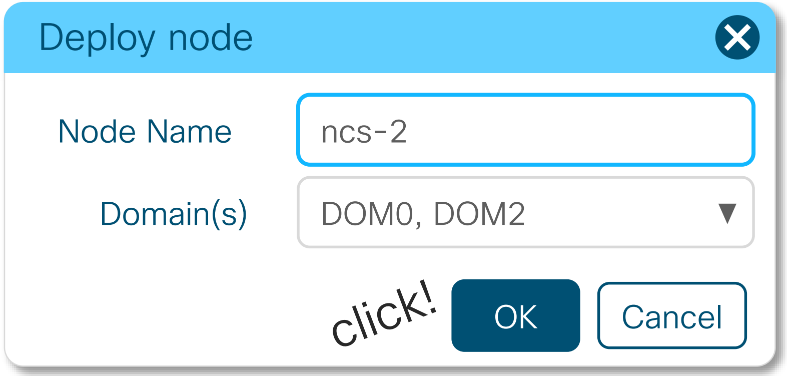
- Address allocation
  - Loopback and interfaces
- SID allocation
  - Algo 0 and Flex-Algos
- Multi-Domain
- ISIS summarization and redistribution between domains
- TI-LFA
- BFD



SRv6 Automation

NSO "click" and the following happens

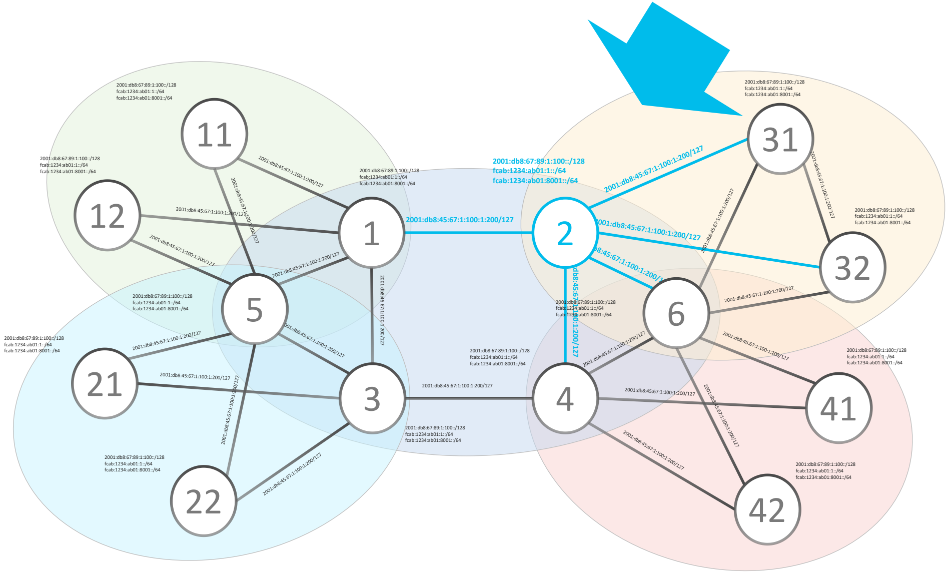
- Address allocation
  - Loopback and interfaces
- SID allocation
  - Algo 0 and Flex-Algos
- Multi-Domain
- ISIS summarization and redistribution between domains
- TI-LFA
- BFD



Deploy node

Node Name: ncs-2

Domain(s): DOM0, DOM2 ▼



cisco Live!

Configuration Automation next-step

- ISIS Flex-Algo Slicing
- BGP Services
  - Internet
  - L3VPN
  - eVPN PW
- Linux Servers

Troubleshooting Automation

- Brainstorming
- Please ping if interested

Conclusion

Simplicity Always Prevails

~~LDP~~

~~RSVP-TE~~

~~Inter-AS Option A/B/C~~

~~MPLS~~

~~UDP/VxLAN~~

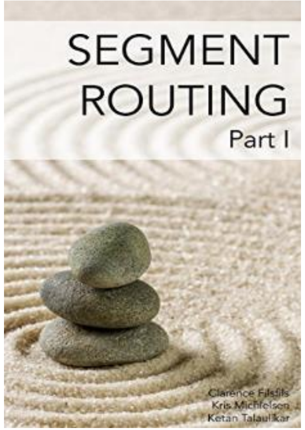
~~NSH~~

Furthermore with more scale and functionality

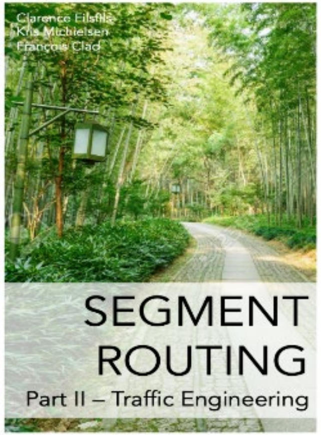
At record speed

- 8 large-scale commercial deployments
  - Softbank, Iliad, China Telecom, LINE corporation,
    China Unicom, CERNET2, China Bank and Uganda MTN.
- 18 HW linerate implementations
  - Cisco Systems, Huawei
  - Broadcom, Barefoot, Intel, Marvell, Mellanox
  - Multiple Interop Reports
- 9 open-source platforms/ Applications
  - Linux, FD.io VPP, P4, Wireshark, tcpdump, iptables, nftables, snort

Stay up-to-date



amzn.com/B01I58LSUO



amazon.com/dp/B07N13RDM9

twitter.com/SegmentRouting

segment-routing.net

facebook.com/SegmentRouting/

linkedin.com/groups/8266623

You make **possible**

References

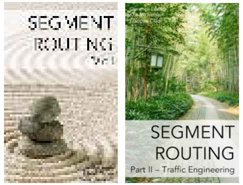
Resources / Stay Up-To-Date

http://www.segment-routing.net/

https://www.linkedin.com/groups/8266623

https://twitter.com/SegmentRouting

https://www.facebook.com/SegmentRouting/



Segment Routing, Part I / II - Textbooks

Demo

Let’s see the Demonstration …

Complete your online session survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

Continue your education

Demos in the Cisco campus

Walk-in self-paced labs

Meet the engineer 1:1 meetings

Related sessions

Thank you

cisco

You make **possible**

Appendices

Appendix

Industry Update

SR is IETF Proposed Standard

Architecture

- Segment Routing Architecture RFC 8402
- Source Packet Routing in Networking (SPRING) Problem Statement and Requirements RFC 7855
- Segment Routing with MPLS data plane RFC 8660

Use-cases

- SR-MPLS over IP RFC 8663
- Resiliency Use Cases in SPRING Networks RFC 8355
- Use Cases for IPv6 Source Packet Routing in Networking (SPRING) RFC 8354
- BGP Prefix Segment in Large-Scale Data Centers RFC 8670
- Interconnecting Millions Of Endpoints With Segment Routing RFC 8604
- Segment Routing interworking with LDP RFC 8661
- Recommendations for RSVP-TE and Segment Routing LSP co-existance RFC 8426

Protocol Extensions

ISIS

- IS-IS Extensions for Segment Routing RFC 8667
- Signaling MSD (Maximum SID Depth) using IS-IS RFC 8491
- Advertising L2 Bundle Member Link Attributes in IS-IS RFC 8668
- IS-IS Traffic Engineering (TE) Metric Extensions RFC 7810

BGP

- Segment Routing Prefix SID extensions for BGP RFC 8669
- BGP-LS Advertisement of IGP Traffic Engineering Performance Metric Extensions RFC 8571

OSPF

- OSPF Extensions for Segment Routing RFC 8665
- OSPFv3 Extensions for Segment Routing RFC 8666
- Signaling MSD (Maximum SID Depth) using OSPF RFC 8476
- OSPF Traffic Engineering (TE) Metric Extensions RFC 7471

PCEP

- PCEP Extensions for Segment Routing RFC 8664

OAM

- A Scalable and Topology-Aware MPLS Dataplane Monitoring System RFC 8403
- Label Switched Path (LSP) Ping/Trace for Segment Routing Networks Using MPLS Dataplane RFC 8287

Performance Measurement

- Packet Loss and Delay Measurement for MPLS Networks RFC 6374
- UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks RFC 7876

SR is IETF Proposed Standard

Architecture

- Segment Routing Architecture RFC 8402
- Source Packet Routing in Networking (SPRING) Problem Statement and Requirements RFC 7855
- Segment Routing with MP

Use-cases

- SR-MPLS over IP RFC 8663
- Resiliency Use Cases in SPRING Networks RFC 8355
- Use Cases for IPv6 Source Packet Routing in Networking (SPRING) RFC 8354
- RFC 8670
- Routing RFC 8604
- ng LSP co-existance RFC 8426

Protocol Exte

ISIS

- IS-IS Extensions for Seg
- Signaling MSD (Maximu
- Advertising L2 Bundle M
- IS-IS Traffic Engineering

FC 8476

C 7471

BGP

- Segment Routing Prefix
- BGP-LS Advertisement
  Performance Metric Exte

Cisco Leads Standards Bodies

Editor of 96% IETF RFCs

Co-author of 100% IETF RFCs

Editor of 77% IETF WG Drafts

Co-author of 84% IETF WG Drafts

OAM

- A Scalable and Topology-Aware MPLS Dataplane Monitoring System RFC 8403
- Label Switched Path (LSP) Ping/Trace for Segment Routing Networks Using MPLS Dataplane RFC 8287

Performance Measurement

- Packet Loss and Delay Measurement for MPLS Networks RFC 6374
- UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks RFC 7876

Appendix

SR Policy liveness monitoring

SR Policy liveness – Appendix (1)

- “SR Policy liveness”: The end-to-end (from headend to tailend) usability of an SR Policy endpoint and candidate-path on the forwarding plane
- liveness monitoring:
  - Monitor end-to-end liveness of an SR Policy candidate-path by periodically sending PM probes along the SR Policy candidate-path from headend through the tailend and back, without dependency on tailend
    - tailend switches probe packets – no punting, no awareness
    - no tailend dependency
      - easier to deploy
      - more scalable

SR Policy liveness – Appendix (2)

- Probe format
    - Same as link-delay measurement (TWAMP)
- PM sessions
    - An internal PM (sub-)session is created for each segment-list of the active candidate-path
- liveness failure detection
    - Liveness failure is detected when last N (default: 3) consecutive probe packets are lost
        - PM sends probe messages in pipeline mode i.e. PM does not wait for the probe response to arrive before sending the next probe query message
    - SR Policy PM liveness session declared down if any of the per-segment-list PM sub-sessions is down
    - Failure action:
        - default: notification only
        - tear down active candidate path
    - Warning if using IP return path:
        - false positives if return path fails while forward path stays up
        - path protection fails if the common return path of both primary and backup candidate-paths fails

SR Policy liveness – Appendix (3)

- Variants (user-configurable)
  - constrain return path by encoding this return in the probe's label stack
    - Prevent false negatives (return path fails while forward path stays up)
    - User can specify a label to return the packet (e.g. headend Prefix-SID, reverse SR Policy BSID)
    - default: IP return path (best-effort)
  - ECMP sweeping
    - change IP destination address (in 127/8 range) to hash on different ECMP paths of SR Policy
    - probabilistic coverage of ECMP paths
    - When using ECMP sweeping, one must encode return path in the probe's label stack (not possible to use default IP return path if probe's dest address is 127/8)
    - Implementation: when sweeping destination address, for each destination address an additional (internal) PM session is created. There is also always a PM (sub-)session to the endpoint address
    - SR Policy PM liveness session declared down if any of the per-destination address PM sub-sessions is down

Configuration

```
segment-routing
 traffic-eng
  policy FOO
   performance-measurement
    delay-measurement
     liveness-detection
      invalidation-action down !! default: none
!
performance-measurement
 delay-profile sr-policy
  probe
   measurement-mode loopback
```

Reverse path – Configuration

```
segment-routing
 traffic-eng
  policy FOO
   performance-measurement
    delay-measurement
     liveness-detection
    reverse-path label <lbl> ! E.g. BSID, Prefix-SID
!
performance-measurement
 delay-profile sr-policy
  probe
   measurement-mode loopback
```

ECMP Sweeping – Configuration

```
segment-routing
 traffic-eng
  policy FOO
   color 20 end-point ipv4 1.1.1.5
   performance-measurement
    delay-measurement
     liveness-detection
     reverse-path label <lbl> ! E.g. BSID, Prefix-SID
   candidate-paths
    preference 100
     dynamic
      metric
       type delay
!
performance-measurement
 delay-profile sr-policy
  probe
   measurement-mode loopback
   sweep
    destination ipv4 127.0.0.0 range 10
```

Appendix

Flex-Algo

Flexible Algorithm

- Flex-Algo
  - FA provides customized IGP algorithms defined by operator for intent-based instantiation of TE
    - great for 5G slicing
  - FA provides simplicity and automation by providing IGP-computed TE paths from anywhere to anywhere, automatically protected by TI-LFA backup paths that are optimized per FA slice (plane)
  - FA provides scalability by enforcing a TE path using a single SID and supporting participation in many FAs using a single loopback prefix
- FA Accumulated metric pitch
  - FA Prefix-SIDs are redistributed with their accumulated metric which allows IGP to compute optimal FA end-to-end paths for inter-area and inter-domain prefixes

ISIS Flex-Algorithm Prefix Metric Sub-TLV

- Flex-Algorithm Prefix Metric (FAPM) sub-TLV is attached to IP reachability TLV (TLVs 135, 235, 236, and 237) of propagated (redistributed/leaked) prefixes
- One FAPM sub-TLV per FA

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |Flex-Algorithm |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Metric                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- FAPM value = metric to reach the prefix for a given FA in a source area or domain
  - Cfr. how the metric is set when prefixes are advertised between areas or domains for default algorithm

M-flag in FAD

- The M-flag in the Flex-Algo Definition (FAD) indicates that ABRs/ASBRs MUST advertise the FAPM with the propagated prefixes and all FA-participating nodes MUST use the FAPM for FA computation
  - M-flag must be set
- For any FA: If FAD has M-flag set, then any propagated prefix without FAPM is considered unreachable
- Configuration:

```
router isis X
  flex-algo 128
    metric-type delay
    prefix-metric
```

Appendix

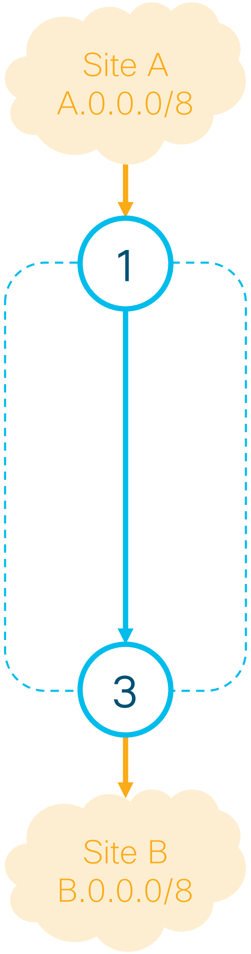
Per-flow ODN/AS

Appendix

SRv6 Use-cases

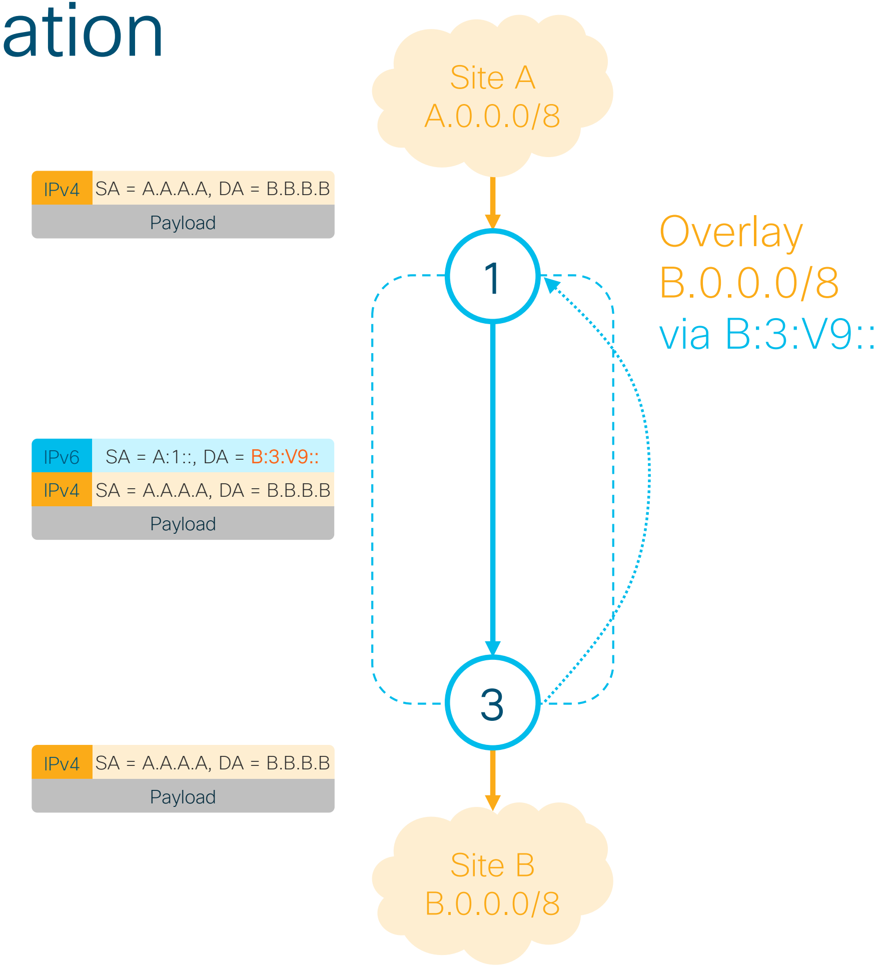
SRv6 – Configuration

```
router isis <ID>
 flex-algo 128
  metric-type delay
  advertise-definition
 !
 address-family ipv6 unicast
  segment-routing srv6
   locator DOM0_ALG0
   locator DOM0_ALG128
!
segment-routing
 srv6
  encapsulation
   source-address a:3::
  !
  locators
   locator DOM0_ALG0
    prefix b:3::/64
   !
   locator DOM0_ALG128
    prefix b:3:8::/64
    algorithm 128
```



VPNv4, VPNv6 – Configuration

```
router bgp <ASN>
  address-family vpnv4|vpnv6 unicast
    vrf all
      segment-routing srv6
        locator <name>
    !
  !
  neighbor <ipv6-addr>
    address-family vpnv4|vpnv6 unicast
    !
  !
  vrf <name>
    address-family ipv4|ipv6 unicast
      segment-routing srv6
        alloc mode {per-vrf | per-ce}
      !
!
```


Site A
A.0.0.0/8
IPv4 SA = A.A.A.A, DA = B.B.B.B
Payload
1
Overlay
B.0.0.0/8
via B:3:V9::
IPv6 SA = A:1::, DA = B:3:V9::
IPv4 SA = A.A.A.A, DA = B.B.B.B
Payload
3
IPv4 SA = A.A.A.A, DA = B.B.B.B
Payload
Site B
B.0.0.0/8

cisco Live!

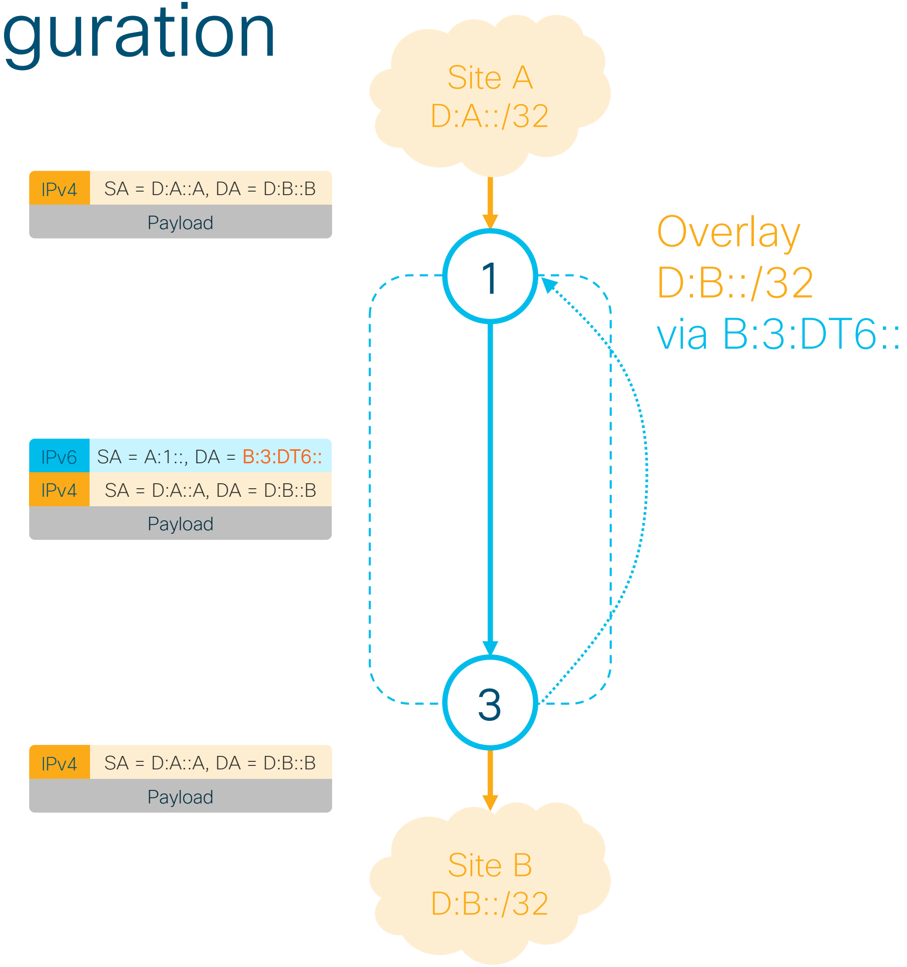
Locators in BGP – global, all vrfs, per-vrf

- Multiple SID Locators can be specified in BGP
  - Locators themselves configured under `segment-routing srv6`
- Global SID locator is used by default
- For L3VPN services, a SID Locator can be specified for all VRFs or per individual VRF
- For internet services a SID Locator can be specified per address-family
- SID allocation mode (per-vrf, per-ce) can be configured for all VRFs and per individual VRF

```
router bgp <ASN>
  segment-routing srv6
    locator <name>
  !
  address-family ipv4|ipv6 unicast
    segment-routing srv6
      locator <name>
  !
  address-family vpnv4|vpnv6 unicast
    vrf all
      segment-routing srv6
        locator <name>
        alloc mode {per-vrf | per-ce}
  !
  vrf <name>
    address-family ipv4|ipv6 unicast
      segment-routing srv6
        locator <name>
        alloc mode {per-vrf | per-ce}
```

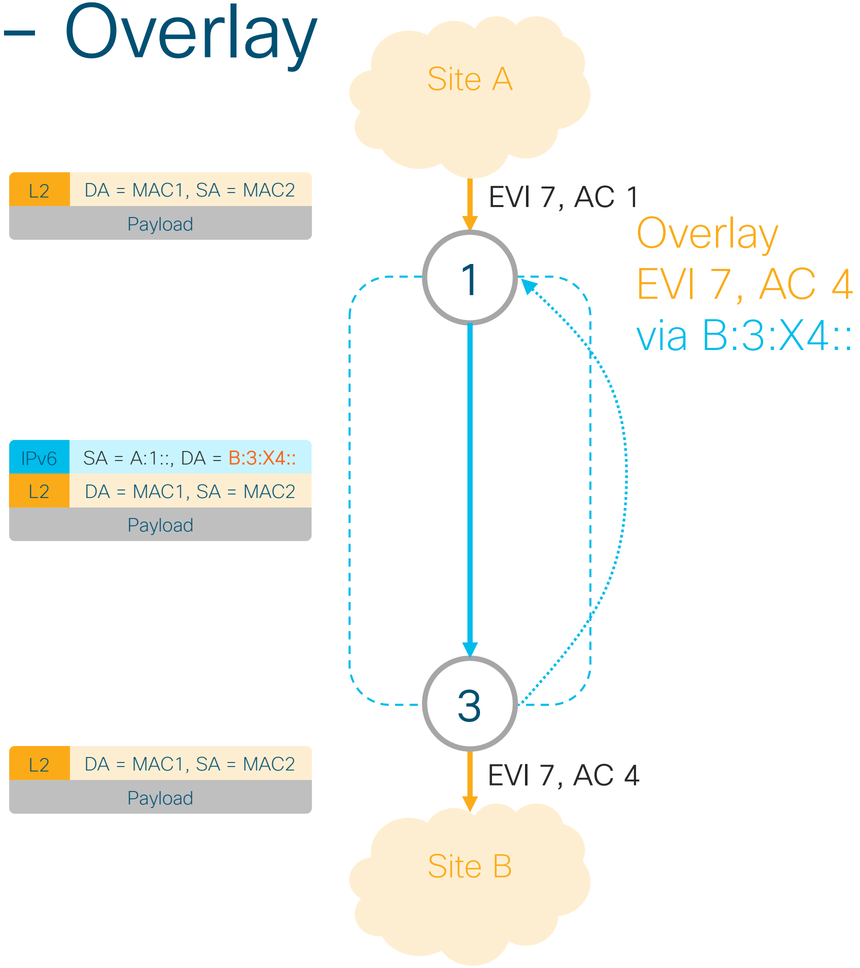
Internet IPv4, IPv6 – Configuration

```
router bgp <ASN>
 address-family ipv4|ipv6 unicast
  segment-routing srv6
   alloc mode {per-vrf | per-ce}
   locator <name>
  !
 !
 neighbor <ipv6-addr>
  address-family ipv4|ipv6 unicast
   encapsulation-type srv6
  !
 !
```



EVPN VPWS single-home – Overlay

- One single SID is needed (End.DX2)
  - B:3:X4
  - “go to 3, decaps and forward on AC 4”
- No new protocol (just BGP)
  - No new SAFI
  - Light ext. to BGP Prefix-SID attribute
- Automated
  - No tunnel to configure
- Efficient
  - SRv6 for everything
  - No other protocol, just IPv6 with SRv6
    - In fact, SRH not even needed (one single SID fits DA)



Site A

EVI 7, AC 1

L2 | DA = MAC1, SA = MAC2
Payload

Overlay
EVI 7, AC 4
via B:3:X4::

1

IPv6 | SA = A:1::, DA = B:3:X4::
L2 | DA = MAC1, SA = MAC2
Payload

3

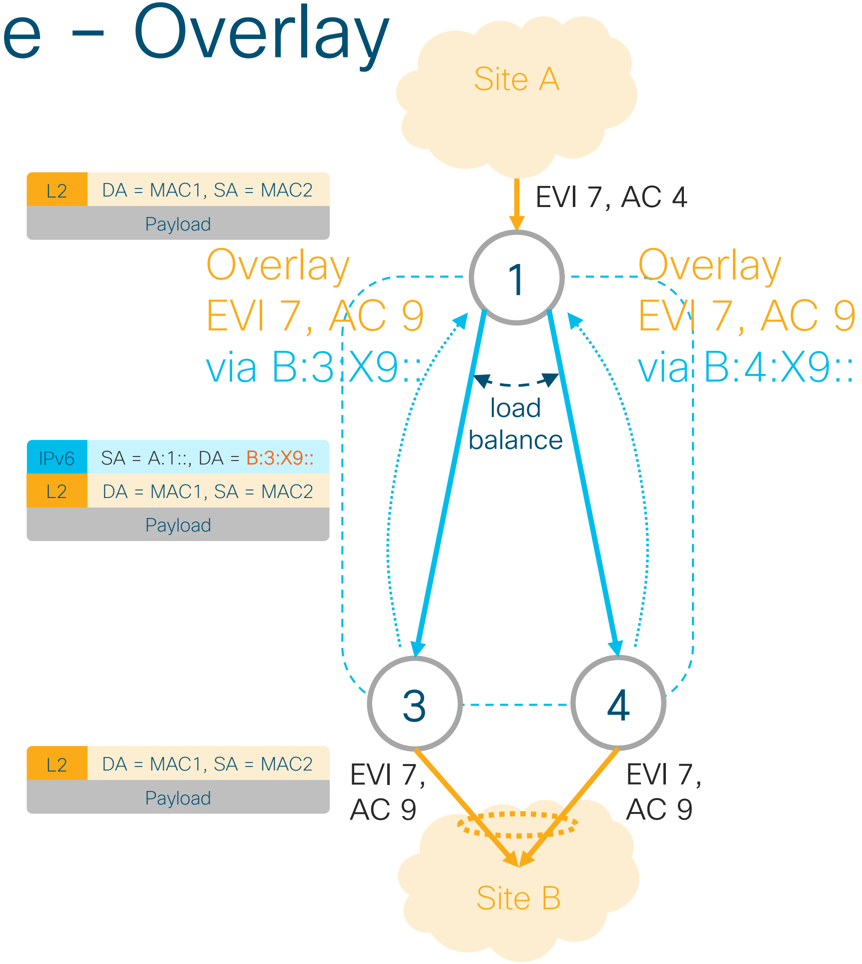
EVI 7, AC 4

L2 | DA = MAC1, SA = MAC2
Payload

Site B

EVPN VPWS MH all-active – Overlay

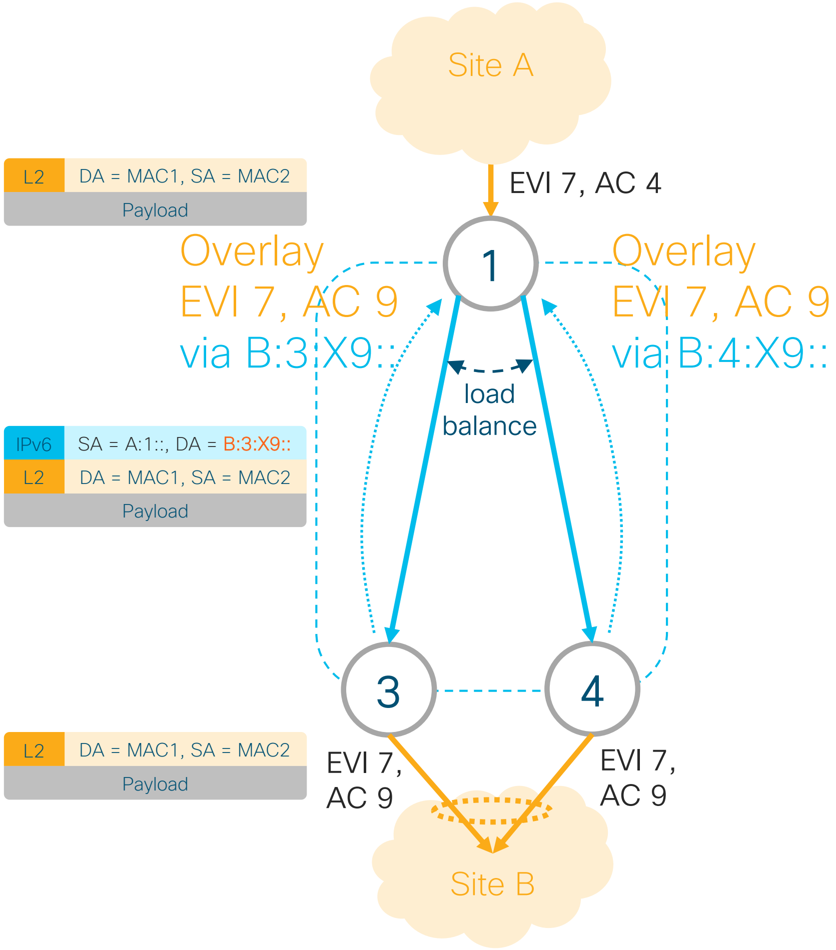
- One single SID is needed (End.DX2)
  - ECMP over B:3:X9 and B:4:X9
  - “LB to 3 and 4, decaps and forward on AC 9”
- EVPN VPWS multi-homing load-sharing and redundancy functionalities apply



Overlay configuration

On 3 and 4:

```
l2vpn
  xconnect group evpn-vpws
    p2p EVI7-AC9
      interface Bundle-Ether10.2
      neighbor evpn evi 7
                     target 4 source 9
        segment-routing srv6
          [locator <name>]
!
evpn
 segment-routing srv6
  locator LOC1
 !
 interface Bundle-Ether10
  ethernet-segment
   identifier type 0
               00.01.00.ac.ce.55.00.0a.00
```



Appendix

SRv6/MPLS L3 Service Interworking Gateway

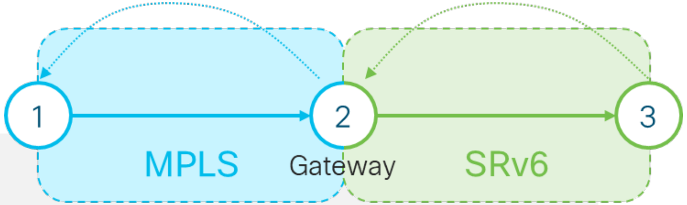
SRv6/MPLS L3 Service Interworking Gateway

- The L3 service SRv6/MPLS gateway enables customers to extend their L3 services between MPLS and SRv6 domains by providing service continuity on the control plane and data plane
- Gateway acts as intermediary for L3 services on control plane and data plane

L3 service stitching

- Gateway acts as intermediary for interworked L3 services
- GW has VRFs configured that need interworking with 2 sets of RTs
  - MPLS L3VPN RTs
  - SRv6 L3VPN RTs (called “stitching RTs”)
- GW imports service routes received from one domain (MPLS | SRv6)
- GW re-advertises exported service routes to the other domain (next-hop-self)
- GW stitches the service on the data plane (End.D*/T.Encaps.Red ⇔ service label)

Gateway configuration



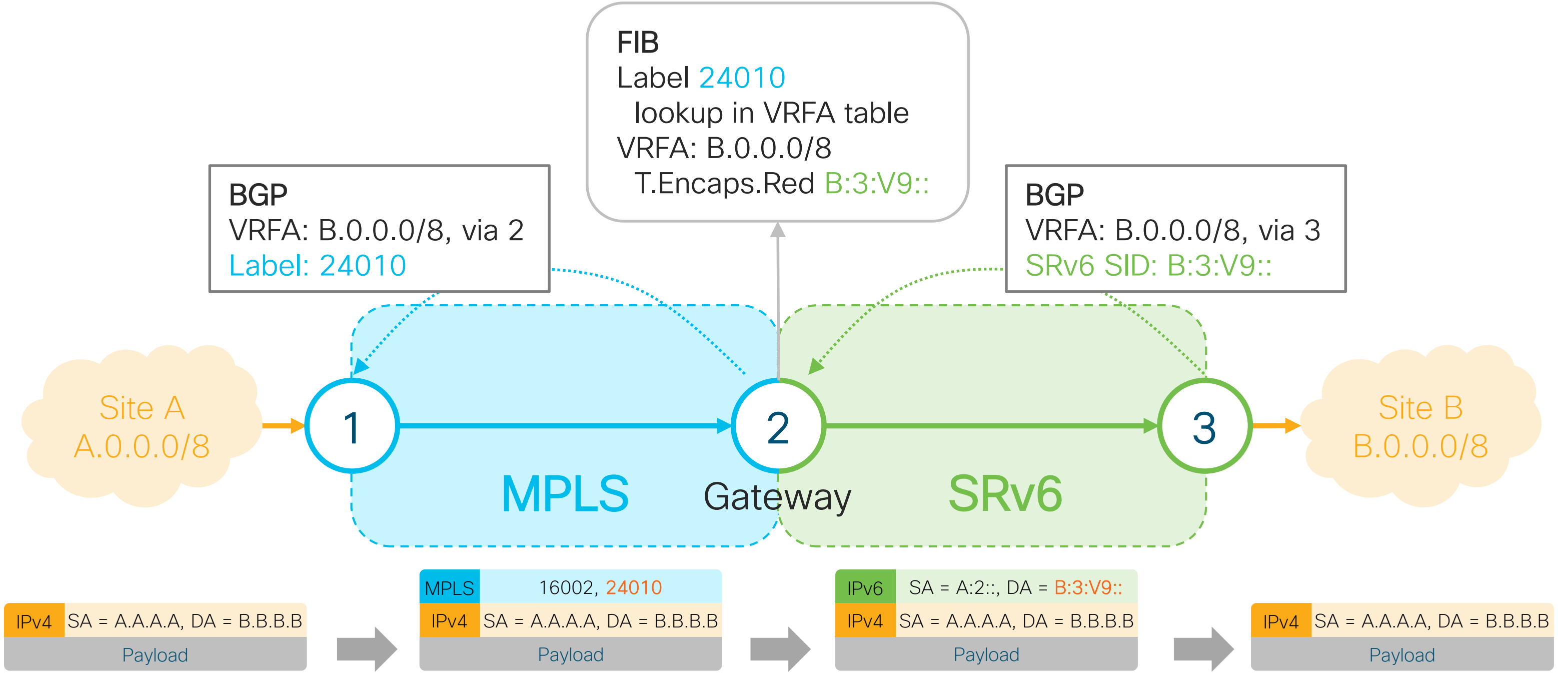
```
vrf ACME
 address-family ipv4 unicast
  import route-target
   1111:1              ; MPLS
   2222:1 stitching    ; SRv6
  !
  export route-target
   1111:1              ; MPLS
   2222:1 stitching    ; SRv6
```

Stitch MPLS domain RTs to SRv6 domain RTs

```
router bgp 100
 segment-routing srv6
  locator LOC1
 !
 neighbor 1.1.1.1
  address-family vpnv4 unicast
   import re-originate stitching-rt
   route-reflector-client
   advertise vpnv4 unicast re-originated
 !
 neighbor a::3
  address-family vpnv4 unicast
   import stitching-rt re-originate
   route-reflector-client
   encapsulation-type srv6
   advertise vpnv4 unicast re-originated stitching-rt
 !
 vrf ACME
  address-family ipv4 unicast
   enable label-mode
   segment-routing srv6
```

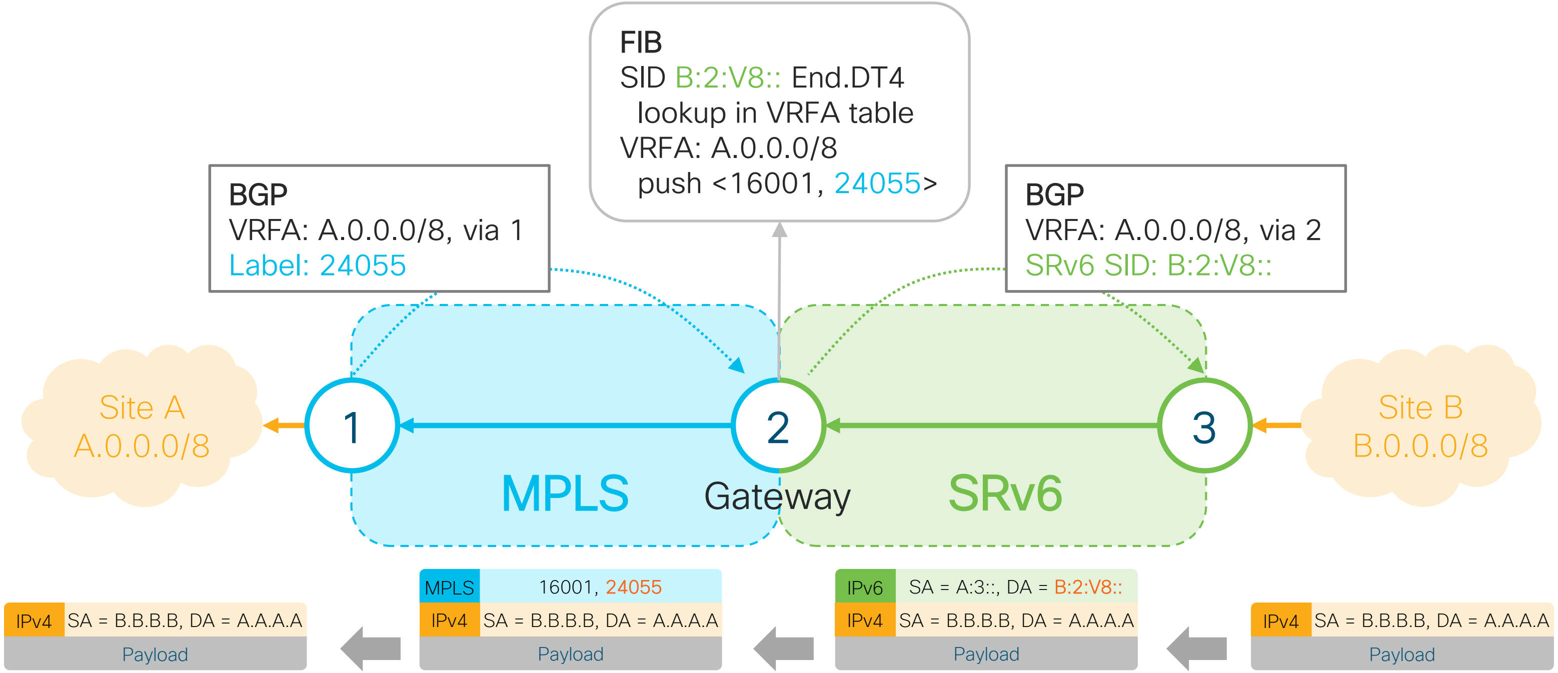
Allocate VPN label and SRv6 SID

MPLS to SRv6



FIB
Label 24010
lookup in VRFA table
VRFA: B.0.0.0/8
T.Encaps.Red B:3:V9::
BGP
VRFA: B.0.0.0/8, via 2
Label: 24010
BGP
VRFA: B.0.0.0/8, via 3
SRv6 SID: B:3:V9::
Site A
A.0.0.0/8
1
2
3
Site B
B.0.0.0/8
MPLS
Gateway
SRv6
MPLS 16002, 24010
IPv6 SA = A:2::, DA = B:3:V9::
IPv4 SA = A.A.A.A, DA = B.B.B.B
Payload

SRv6 to MPLS



FIB
SID B:2:V8:: End.DT4
lookup in VRFA table
VRFA: A.0.0.0/8
push <16001, 24055>

BGP
VRFA: A.0.0.0/8, via 1
Label: 24055

BGP
VRFA: A.0.0.0/8, via 2
SRv6 SID: B:2:V8::

IPv4	SA = B.B.B.B, DA = A.A.A.A
Payload	

MPLS	16001, 24055
IPv4	SA = B.B.B.B, DA = A.A.A.A
Payload	

IPv6	SA = A:3::, DA = B:2:V8::
IPv4	SA = B.B.B.B, DA = A.A.A.A
Payload	

IPv4	SA = B.B.B.B, DA = A.A.A.A
Payload	

Appendix

SRv6 Massive-Scale End-to-End Reachability With SLA

Locator Summarization

- Since SRv6 leverages longest-prefix-match IP forwarding, massive-scale reachability can be achieved by simply summarizing SID Locators at ABRs and ASBRs
  - No summarization possible in MPLS

Locator Summarization configuration

```
segment-routing
 srv6
  locators
   locator ALGO0
    prefix b:0:0:1::/64
   !
   locator ALGO128
    prefix b:0:8:1::/64
    algorithm 128
!
router isis SRv6
 address-family ipv6 unicast
  summary-prefix b:0:0::/48 explicit
  summary-prefix b:0:8::/48 algorithm 128 explicit
  !
  segment-routing srv6
   locator ALGO0
   !
   locator ALGO128
```

"explicit" → only locators from the specified algorithm contribute to the summary